Buku Pegangan Teknik Jaringan Menggunakan MikroTik

Belajar Teknik Jaringan dari Nol dengan Ubuntu 26.04, VirtualBox, GNS3, VPC, dan RouterOS CLI

Onno W. Purboi
onno@indo.net.id
onno@itts.ac.id
Twitter onnowpurbo

Institut Teknologi Tangerang Selatan (ITTS)
Dinas KOMINFO Tangerang Selatan
OnnoCenter
2026

Lisensi: Karya ini dilisensikan di bawah **Creative Commons Attribution (CC BY) 4.0 International**. Anda bebas menggunakan, membagikan, dan mengadaptasi karya ini untuk tujuan apa pun, termasuk komersial, dengan tetap mencantumkan atribusi yang layak kepada penulis.

Disclaimer: Buku ini disusun untuk tujuan edukasi, dokumentasi, dan refleksi publik berdasarkan sumber-sumber yang dianggap kredibel pada saat penulisan. Penulis telah berupaya menjaga akurasi, namun sangat mungkin masih terdapat kekurangan atau kekeliruan; karena itu, buku ini tidak dimaksudkan sebagai satu-satunya rujukan yang final, dan penulis terbuka terhadap koreksi serta masukan yang membangun

Jaringan tidak cukup dipahami dengan membaca. Jaringan harus dibuat, diuji, di rusak, lalu diperbaiki.

Filosofi Buku

Buku ini tidak mengejar teori & sertifikasi secara berlebihan. Fokusnya adalah **kemampuan nyata**:

- Membuat lab jaringan sendiri.
- Menghitung IP dan netmask.
- Menghubungkan VPC GNS3 ke MikroTik.
- Menghubungkan Ubuntu 26.04 di VirtualBox ke GNS3.
- Mengkonfigurasi MikroTik lewat **CLI**.
- Membuat NAT, DHCP, firewall, HotSpot, VPN, IPv6, OSPF, dan BGP.
- Melakukan troubleshooting dengan cara teknisi jaringan.

Kata Pengantar

Alhamdulillah, buku *Buku Pegangan Teknik Jaringan Menggunakan MikroTik* ini disusun untuk membantu **pemula, mahasiswa, siswa SMK, teknisi jaringan, dan *network admin* awal** belajar jaringan secara praktis. Buku ini menekankan prinsip sederhana bahwa **belajar jaringan harus dimulai dari praktik**, bukan hanya membaca teori. Konsep seperti **IP address**, **netmask**, **gateway**, **DNS**, **NAT**, **firewall**, **VPN**, **OSPF**, dan **BGP** akan jauh lebih mudah dipahami jika langsung diuji dalam lab.

Dengan SATU PC berbasis **Ubuntu 26.04**, pembaca dapat membangun lab jaringan lengkap menggunakan **VirtualBox**, **GNS3**, **MikroTik CHR**, **VPC GNS3**, dan **Ubuntu VM**. Ubuntu 26.04 LTS layak menjadi basis lab buku ini karena mendapat dukungan keamanan dan perbaikan penting selama lima tahun sampai April 2031. (documentation.ubuntu.com)

VirtualBox dipakai untuk menjalankan router dan server virtual, sedangkan **GNS3** dipakai untuk menggambar dan menghubungkan topologi jaringan. **VPC GNS3** dipakai sebagai client ringan untuk latihan **DHCP, ping, trace, IPv4, dan IPv6**. Sementara itu, **MikroTik CHR** dipakai sebagai router virtual agar pembaca bisa berlatih RouterOS tanpa harus langsung membeli banyak perangkat fisik.

Buku ini memakai **CLI**, bukan Winbox, karena CLI membuat konfigurasi lebih mudah dipahami, dicatat, diulang, diperiksa, dan diekspor. Dengan CLI, pembaca belajar melihat hubungan antara **interface, IP address, route, NAT, firewall, VPN, OSPF, dan BGP** secara lebih jelas. Setiap perintah yang diketik harus dipahami, bukan sekadar disalin.

Cara menggunakan buku ini sederhana. Pembaca sebaiknya membaca konsep singkat terlebih dahulu, melihat topologi, mengisi tabel IP, mengetik CLI, lalu menguji hasilnya dengan **ping** dan **traceroute**. Jika terjadi error, jangan langsung menghapus semua konfigurasi. Catat error tersebut, periksa dari awal, lalu perbaiki satu per satu sampai lab berhasil.

Ada beberapa aturan utama yang perlu diingat. **Jangan menyalin konfigurasi tanpa paham**. **Jangan mengaktifkan DHCP sembarangan ke jaringan nyata**. **Jangan membuat firewall tanpa jalur akses cadangan**. Dan yang paling penting, **jangan lanjut ke bab berikutnya sebelum lab pada bab berjalan dengan benar**. Setiap bab dalam buku ini disusun bertahap, sehingga kegagalan pada bab awal akan membuat bab berikutnya lebih sulit dipahami.

Buku ini bukan sekadar kumpulan perintah. Buku ini melatih pola pikir teknisi jaringan: **cek interface, cek IP, cek route, cek DNS, cek NAT, cek firewall, lalu cek log**. Semoga buku ini membantu pembaca belajar jaringan MikroTik secara **mandiri, murah, aman, dan terstruktur**. Jangan takut error, karena **error yang dicatat dengan baik adalah guru terbaik dalam belajar jaringan**.

Jakarta, Mei 2026
Penulis

Daftar Isi

Kata Pengantar **4**
Daftar Isi **5**
Bab 1. Fondasi Teknik Jaringan untuk Pemula: TCP/IP, IP Address, dan Netmask **21**
1.1 Tujuan Bab 21
1.2 Masalah Nyata: IP Sudah Diisi, Tapi Tidak Bisa Ping 21
1.3 Apa Itu Jaringan Komputer? 22
1.4 Model TCP/IP Secara Sederhana 23
1.5 IPv4: Alamat Dasar dalam Lab 24
1.6 Network Address, Host Address, Broadcast Address 25
Network Address 25
Host Address 25
Broadcast Address 25
1.7 Membaca 192.168.10.1/24 26
1.8 Private IP vs Public IP 27
1.9 Gateway: Pintu Keluar Subnet 28
1.10 ARP, ICMP, dan DNS 29
ARP 29
ICMP 29
DNS 29
1.11 Kalkulasi Netmask Dasar 30
1.12 Tabel Kalkulasi Subnet 30
1.13 Contoh Hitung Subnet 31
1.14 Kapan Memakai /24, /30, dan /32 33
1.15 Hands-on Ubuntu 26.04 ..... **34**
1.16 Hands-on VPC GNS3 untuk Testing **37**
1.17 Preview CLI MikroTik RouterOS v7 dan v6 ..... **38**
1.18 Alur Paket dari Client ke Server **39**
1.19 Kesalahan Umum Pemula **40**
1.20 Latihan Cepat **40**
1.21 Checklist Akhir Bab **41**
Ringkasan Bab 42
Bab 2. Menyiapkan Ekosistem Lab Jaringan dengan Satu PC Ubuntu 26.04 ..... **43**
2.1 Tujuan Bab 43
2.2 Strategi Lab Murah: Satu PC Cukup 43
2.3 Instalasi Paket Pendukung di Ubuntu 26.04 45
2.4 Konsep Virtual Machine untuk Belajar Jaringan 46
2.5 Fungsi VirtualBox dalam Buku Ini 47
2.6 Mode Jaringan VirtualBox 48
NAT 48
NAT Network 49
Bridged Adapter 49
Host-only Adapter 50

Internal Network..... 50
2.7 Kapan Memakai Mode Jaringan.....51
2.8 Fungsi GNS3.....52
2.9 Fungsi VPC GNS3..... 53
2.10 Download MikroTik CHR..... 54
2.11 Import CHR ke VirtualBox..... 55
2.12 Import CHR ke GNS3.....56
2.13 Membuat Ubuntu Server VM sebagai Client/Server Uji..... 57
2.14 Menghubungkan Ubuntu Server VM ke GNS3..... 58
2.15 Desain Lab 1: 1 CHR + 1 VPC.....59
2.16 Desain Lab 2: 1 CHR + 2 VPC.....60
2.17 Desain Lab 3: 1 CHR + 1 Ubuntu Server VM.....61
2.18 Desain Lab 4: 3 CHR untuk Routing.....62
2.19 Desain Lab 5: 2 CHR + 2 VPC untuk VPN.....63
2.20 Desain Lab 6: 1 CHR + VPC IPv4 + VPC IPv6..... 64
2.21 Manajemen Nama.....65
2.22 Strategi Dokumentasi Lab.....66
2.23 Verifikasi Lab.....67
2.24 Kesalahan Umum dan Solusi.....68
2.25 Checklist Akhir Bab..... 69
Ringkasan Bab.....70
Bab 3. Masuk ke RouterOS CLI dan Konfigurasi Awal.....71
3.1 Tujuan Bab.....71
3.2 Masalah Nyata: Takut Salah Ketik di Router.....72
3.3 Mengenal RouterOS dan CLI.....72
3.4 Login Awal RouterOS CHR.....73
3.5 Mengganti Identity Router.....74
3.6 Mengganti Password dan Membuat User Admin Baru..... 75
3.7 Menonaktifkan Service yang Tidak Dipakai..... 76
3.8 Struktur Menu CLI RouterOS.....77
3.9 Perintah Dasar CLI.....78
3.10 Konfigurasi Awal Router.....80
3.11 Memberi IP Awal untuk Lab.....80
3.12 Akses SSH dari Ubuntu Terminal.....81
3.13 DNS Dasar..... 81
3.14 Backup vs Export..... 82
3.15 Cara Membaca Konfigurasi.....83
3.16 Cara Menghapus Konfigurasi Lab.....84
3.17 Perbedaan Umum RouterOS v7 dan v6.....85
3.18 Lab 3.1 — Konfigurasi Awal R1 di GNS3.....86
3.19 Lab 3.2 — Akses SSH dari Ubuntu.....87
3.20 Checklist Akhir Bab..... 88
Ringkasan Bab.....89
Bab 4. Router Dasar: IP Statik, DHCP Client, DHCP Server, DNS, dan Default Route...90

4.1 Tujuan Bab.....90
4.2 Masalah Nyata: Router Bisa Keluar, Tapi Client Tidak Bisa.....90
4.3 Fungsi Router.....91
4.4 WAN dan LAN.....92
4.5 IP Statik di Sisi LAN.....93
4.6 IP Dinamis dengan DHCP Client.....94
4.7 Default Route.....95
4.8 DNS Resolver dan DNS Cache.....96
4.9 DHCP Server untuk LAN.....97
4.10 IP Pool.....97
4.11 DHCP Server dan DHCP Network.....98
4.12 Lease DHCP.....99
4.13 Konfigurasi Router Dasar: Skenario Utama.....100
4.14 Verifikasi Router Dasar.....101
4.15 VPC GNS3 sebagai Client DHCP.....102
4.16 Ubuntu Server VM sebagai Client.....103
4.17 Cara Membaca Alur DHCP.....104
4.18 Lab 4.1 — CHR Tersambung ke GNS3, VPC Mendapat IP.....105
4.19 Lab 4.2 — Ubuntu VM sebagai Client Uji.....106
4.20 Kesalahan Umum dan Solusi.....107
4.21 Checklist Akhir Bab.....108
Ringkasan Bab.....109
Bab 5. NAT dan Masquerade: Menghubungkan VPC dan Ubuntu Client ke Internet...110
5.1 Tujuan Bab.....110
5.2 Masalah Nyata: Client Bisa Ping Router, Tapi Tidak Bisa Keluar.....110
5.3 Mengapa NAT Dibutuhkan.....111
5.4 NAT vs Routing.....112
5.5 Source NAT.....113
5.6 Masquerade.....113
5.7 NAT untuk Lab GNS3.....114
5.8 NAT untuk Ubuntu VM.....115
5.9 Konfigurasi Dasar Sebelum NAT.....116
5.10 Konfigurasi NAT Masquerade.....117
5.11 Verifikasi RouterOS v7/v6.....117
5.12 Verifikasi Koneksi VPC GNS3.....118
5.13 Verifikasi Koneksi Ubuntu VM.....119
5.14 Lab 5.1 — VPC Keluar Melalui NAT MikroTik.....120
5.15 Lab 5.2 — Ubuntu Client Keluar Melalui NAT MikroTik.....121
5.16 Membaca NAT dari Connection Tracking.....122
5.17 Troubleshooting NAT.....122
5.18 Checklist Akhir Bab.....125
Ringkasan Bab.....126
Bab 6. Firewall Dasar, NAT Lanjutan, dan L7 Firewall.....127
6.1 Tujuan Bab.....127

6.2 Mengapa Firewall Wajib..... 127
6.3 Chain: Input, Forward, Output..... 128
6.4 Connection State..... 129
6.5 Firewall untuk Melindungi Router..... 130
6.6 Firewall untuk Melindungi LAN..... 131
6.7 Address List..... 132
6.8 Logging Dasar..... 133
6.9 NAT Lanjutan: Dst-NAT dan Port Forwarding..... 134
6.10 Ubuntu Server untuk Uji Port Forwarding..... 135
6.11 Hairpin NAT sebagai Pengantar..... 136
6.12 L7 Firewall: Konsep Layer 7 Matcher..... 136
6.13 Contoh L7 RouterOS v7/v6..... 137
6.14 Lab 6.1 — Blok Akses dari WAN, Izinkan Admin dari LAN..... 138
6.15 Lab 6.2 — Firewall Forward untuk LAN ke WAN..... 139
6.16 Lab 6.3 — Port Forwarding ke Ubuntu Server..... 140
6.17 Lab 6.4 — Uji L7 Sederhana di Lab..... 141
6.18 Verifikasi Firewall..... 142
6.19 Kesalahan Umum Firewall..... 143
6.20 Checklist Akhir Bab..... 144
Ringkasan Bab..... 145
Bab 7. Bridge, Switching, VLAN Dasar, dan Segmentasi LAN..... 146
7.1 Tujuan Bab..... 146
7.2 Mengapa Bab Ini Penting..... 146
7.3 Bridge sebagai Switch Virtual..... 147
7.4 Port Bridge..... 148
7.5 Bridge vs Router..... 149
7.6 VLAN Dasar: Mengapa Dibutuhkan..... 150
7.7 VLAN ID, Access Port, dan Trunk Port..... 150
7.8 Segmentasi LAN: Admin, User, Tamu..... 151
7.9 Bridge di GNS3..... 152
7.10 Bridge di RouterOS..... 152
7.11 Lab 7.1 — 1 CHR + 2 VPC dalam Satu Bridge..... 153
7.12 VLAN Dasar di RouterOS..... 154
7.13 Lab 7.2 — VLAN Admin dan VLAN User..... 154
7.14 Access Port vs Trunk Port..... 156
7.15 Kapan Memakai Bridge, Kapan Memakai Routing..... 157
7.16 Kesalahan Umum..... 158
7.17 Verifikasi..... 159
7.18 Checklist Akhir Bab..... 160
Ringkasan Bab..... 160
Bab 8. Wireless MikroTik dan HotSpot..... 161
8.1 Tujuan Bab..... 161
8.2 Mengapa Wireless dan HotSpot Penting..... 161
8.3 Konsep Wireless LAN..... 162

8.4 Istilah Penting Wireless..... 162
SSID..... 162
Channel..... 162
Band..... 163
Country..... 163
Security Profile..... 163
8.5 Access Point vs Station..... 164
Access Point..... 164
Station..... 164
8.6 Wireless di Perangkat Fisik MikroTik..... 165
8.7 HotSpot sebagai Captive Portal..... 165
8.8 Komponen Dasar HotSpot..... 166
8.9 User HotSpot..... 166
8.10 Profile dan Limit..... 167
8.11 Walled Garden..... 168
8.12 RADIUS sebagai Pengantar..... 168
8.13 Arsitektur Lab Bab 8..... 169
8.14 Lab 8.1 — CHR untuk HotSpot di GNS3..... 170
8.15 Konfigurasi Dasar HotSpot..... 171
8.16 Verifikasi Konfigurasi HotSpot..... 172
8.17 VPC sebagai Client HotSpot..... 172
8.18 Ubuntu VM sebagai Client Uji..... 173
8.19 Wireless Fisik MikroTik sebagai Access Point..... 174
HotSpot bekerja sebagai pintu autentikasi sebelum akses penuh diberikan ke pengguna.. 174
8.20 Perbedaan HotSpot dan WPA/Wi-Fi Security..... 175
8.21 Kesalahan Umum..... 175
8.22 Checklist Akhir Bab..... 177
Ringkasan Bab..... 178
Bab 9. VPN untuk Admin Jaringan: WireGuard, L2TP/IPsec, dan Tunnel Aman..... 179
9.1 Tujuan Bab..... 179
9.2 Apa Itu VPN..... 179
9.3 Tunnel vs Routing Biasa..... 180
9.4 Remote Access vs Site-to-Site..... 181
9.5 WireGuard untuk RouterOS v7..... 182
9.6 L2TP/IPsec untuk RouterOS v6/v7..... 182
9.7 Arsitektur Lab VPN..... 183
9.8 Konfigurasi Dasar Site A dan Site B..... 184
9.9 Key Pair WireGuard..... 185
9.10 Konfigurasi Peer WireGuard..... 186
9.11 Firewall untuk VPN..... 187
9.12 Routing Lewat VPN..... 188
9.13 Lab 9.1 — VPN Antardua Site..... 189
Langkah 1 — Siapkan IP Dasar..... 189

Langkah 2 — Siapkan VPC....189
Langkah 3 — Buat WireGuard....189
Langkah 4 — Tambah Peer....190
Langkah 5 — Buka Firewall VPN....190
Langkah 6 — Uji Tunnel....190
9.14 Catatan RouterOS v6: L2TP/IPsec....191
9.15 Allowed Address....192
9.16 Troubleshooting VPN....193
9.17 Verifikasi Akhir....195
9.18 Checklist Akhir Bab....196
Ringkasan Bab....196
Bab 10. IPv6 dengan VPC GNS3 dan Ubuntu 26.04 di VirtualBox....197
10.1 Tujuan Bab....197
10.2 Mengapa IPv6 Penting....197
10.3 Format Alamat IPv6....198
10.4 Prefix IPv6....199
10.5 Link-local Address....200
10.6 Neighbor Discovery....200
10.7 SLAAC....201
10.8 DHCPv6 sebagai Pengantar....202
10.9 Dual-stack IPv4/IPv6....203
10.10 IPv6 di RouterOS....204
10.11 IPv6 di VPC GNS3....205
10.12 IPv6 di Ubuntu 26.04....206
10.13 Lab 10.1 — CHR IPv6 Router + VPC IPv6....207
10.14 Lab 10.2 — CHR IPv6 Router + Ubuntu VM....208
10.15 Lab 10.3 — Dual-stack IPv4/IPv6....209
10.16 Lab 10.4 — Routing IPv6 Sederhana....210
10.17 Firewall IPv6: Jangan Aktifkan Tanpa Proteksi....211
10.18 Verifikasi IPv6....212
10.19 Kesalahan Umum IPv6....212
10.20 Checklist Akhir Bab....213
Ringkasan Bab....214
Bab 11. Routing Lanjutan: Static Route, ECMP, Failover, dan OSPF....215
11.1 Tujuan Bab....215
11.2 Mengapa Routing Diperlukan....215
11.3 Static Route....216
11.4 Default Route vs Specific Route....217
11.5 Administrative Distance....218
11.6 Check-Gateway....218
11.7 ECMP....219
11.8 Failover Sederhana....219
11.9 OSPF: Mengapa Perlu Routing Dinamis....220
11.10 Istilah Penting OSPF....220

11.11 Topologi Lab Bab 11 221
11.12 Konfigurasi Dasar IP 222
11.13 Lab 11.1 — Static Route Antar-LAN 223
11.14 Lab 11.2 — OSPF v7 Single-Area 224
11.15 OSPF v6 sebagai Pembanding 225
11.16 ECMP Dua Gateway 226
11.17 Failover dengan Check-Gateway 226
11.18 Passive Interface 227
11.19 Single-Area dan Multi-Area 227
11.20 Troubleshooting Routing dan OSPF 228
11.21 Verifikasi Akhir 229
11.22 Checklist Akhir Bab 230
Ringkasan Bab 230
Bab 12. BGP untuk Pemula: Simulasi ISP Kecil dan Customer 231
12.1 Tujuan Bab 231
12.2 Apa Itu BGP 231
12.3 Kapan BGP Dibutuhkan 232
12.4 ASN 233
12.5 Prefix 233
12.6 Peer 234
12.7 eBGP vs iBGP 234
12.8 BGP Bukan Pengganti OSPF 235
12.9 Route Advertisement 236
12.10 Route Filtering 236
12.11 Default Route dari ISP ke Customer 237
12.12 Topologi Lab 1: Customer ↔ ISP-A 238
12.13 Konfigurasi Dasar IP 239
12.14 BGP RouterOS v7 — Customer 239
12.15 BGP RouterOS v7 — ISP-A 240
12.16 BGP RouterOS v6 sebagai Pembanding 240
12.17 Topologi Lab 2: Dua ISP dan Satu Customer 241
12.18 BGP v7 — Customer ke ISP-A dan ISP-B 241
12.19 Filter Mencegah Route Bocor 242
12.20 Default Route dari ISP 242
12.21 Verifikasi BGP 243
12.22 Troubleshooting BGP 243
12.23 Etika BGP 245
12.24 Checklist Akhir Bab 246
Ringkasan Bab 247
Bab 13. Troubleshooting dan Proyek Mini Akhir 248
13.1 Tujuan Bab 248
13.2 Prinsip Troubleshooting 249
13.3 Urutan Cek Dasar 250
13.4 Tools RouterOS 251

Ping..... 251
Traceroute..... 251
Torch.....251
Packet Sniffer..... 252
Log..... 252
Route.....252
Firewall Connection.....252
Monitor Traffic.....253
13.5 Tools Ubuntu..... 254
13.6 Tools VPC GNS3.....255
13.7 Kasus Nyata 1 — VPC Tidak Dapat IP..... 256
13.8 Kasus Nyata 2 — VPC Bisa Ping Gateway, Tetapi Tidak Bisa Keluar..... 257
13.9 Kasus Nyata 3 — Ubuntu VM Tidak Tersambung ke GNS3.....258
13.10 Kasus Nyata 4 — NAT Salah Interface..... 259
13.11 Kasus Nyata 5 — Firewall Terlalu Ketat..... 260
13.12 Kasus Nyata 6 — IPv6 Aktif Tetapi Tidak Aman..... 261
13.13 Kasus Nyata 7 — OSPF Neighbor Tidak Muncul.....262
13.14 Kasus Nyata 8 — BGP Session Tidak Established..... 263
13.15 Proyek Mini Akhir..... 264
13.16 IP Plan Proyek Mini.....265
13.17 Tugas Proyek Mini.....265
13.18 Verifikasi Proyek Mini..... 267
13.19 Template Laporan Proyek Mini.....268
13.20 Checklist Akhir Bab..... 269
Ringkasan Bab.....270
Lampiran A. Tabel CIDR, Netmask, Jumlah Host.....271
A.1 Tabel CIDR /24 sampai /32.....271
A.2 Contoh Pembagian Subnet dari 192.168.10.0/24.....272
A.3 Kapan Memakai Prefix Tertentu.....273
A.4 Contoh Praktis untuk Lab MikroTik.....273
A.5 Golden Rules CIDR..... 274
Lampiran B. Template IP Plan Lab.....275
B.1 Prinsip Membuat IP Plan..... 275
B.2 Template IP Plan Umum..... 275
B.3 Template Router.....276
B.4 Template Client VPC GNS3.....277
B.5 Template Ubuntu Server VM.....277
B.6 Template Link Antar-Router..... 278
B.7 Template VLAN / Segmentasi LAN.....278
B.8 Template IPv6 Lab..... 279
B.9 Template DNS dan Gateway.....279
B.10 Template Route Plan.....280
B.11 Contoh IP Plan Lengkap Lab 3 Router..... 281
B.12 Checklist IP Plan Sebelum Konfigurasi.....282

B.13 Golden Rules IP Plan....282
Lampiran C. Template Interface Plan....283
C.1 Prinsip Interface Plan....283
C.2 Pola Penamaan Interface yang Disarankan....283
C.3 Template Interface Plan Router....284
C.4 Template Link GNS3....284
C.5 Template VPC GNS3....285
C.6 Template Ubuntu VM....286
C.7 Template VirtualBox Adapter....287
C.8 Template Bridge dan VLAN Interface....287
C.9 Template Komentar Interface RouterOS....288
C.10 Contoh Interface Plan Lab 1 CHR + 1 VPC....288
C.11 Contoh Interface Plan Lab 1 CHR + 2 VPC....289
C.12 Contoh Interface Plan Lab 3 CHR Routing....290
C.13 Contoh Interface Plan Lab VPN....290
C.14 Kesalahan Umum Interface Plan....291
C.15 Checklist Interface Plan Sebelum Lab....291
C.16 Golden Rules Interface Plan....292
Lampiran D. Template Konfigurasi Router Dasar....293
D.1 Asumsi Topologi....293
D.2 Template Konfigurasi Cepat RouterOS v7/v6....294
D.3 Template dengan NAT Masquerade Opsional....295
D.4 Penjelasan Singkat Setiap Bagian....296
D.5 Verifikasi RouterOS....297
D.6 Verifikasi VPC GNS3....298
D.7 Verifikasi Ubuntu VM....299
D.8 Template IP Plan untuk Router Dasar....299
D.9 Template Interface Plan....299
D.10 Template Jika LAN Menggunakan IP Statik di Client....300
D.11 Template Konfigurasi DNS Client ke Router....300
D.12 Template Reset Lab Router Dasar....301
D.13 Kesalahan Umum....301
D.14 Checklist Akhir Template....302
D.15 Golden Rules Router Dasar....302
Lampiran E. Template NAT dan Masquerade....303
E.1 Topologi Dasar NAT....303
E.2 Template NAT Masquerade RouterOS v7/v6....304
E.3 Template Router Dasar + NAT Lengkap....304
E.4 Verifikasi di RouterOS....305
E.5 Verifikasi dari VPC GNS3....306
E.6 Verifikasi dari Ubuntu VM....307
E.7 Template NAT dengan Interface List....308
E.8 Template NAT untuk Beberapa LAN....309
E.9 Template NAT untuk Lab GNS3....310

E.10 Template NAT untuk Ubuntu VM....311
E.11 Troubleshooting NAT....312
E.12 Flow Verifikasi NAT....313
E.13 Checklist NAT Berhasil....313
E.14 Golden Rules NAT....314
Lampiran F. Template Firewall Aman....315
F.1 Prinsip Firewall Aman....315
F.2 Asumsi Interface....315
F.3 Template Address List LAN Admin....316
F.4 Template Firewall Input Aman....317
F.5 Template Firewall Forward Aman....318
F.6 Template Forward untuk Beberapa LAN....319
F.7 Template Firewall dengan Interface List....320
F.8 Template Izinkan DNS dari LAN ke Router....321
F.9 Template Izinkan SSH dari Admin LAN....321
F.10 Template Izinkan Port Forwarding ke Ubuntu Server....322
F.11 Template Firewall IPv6 Input Dasar....323
F.12 Urutan Rule yang Disarankan....323
F.13 Verifikasi Firewall....324
F.14 Template Logging Drop....324
F.15 Troubleshooting Firewall....325
F.16 Template Firewall Aman Lengkap untuk Lab....326
F.17 Checklist Firewall Aman....327
F.18 Golden Rules Firewall Aman....327
Lampiran G. Template L7 Firewall....328
G.1 Prinsip Dasar L7 Firewall....328
G.2 Kapan L7 Boleh Dipakai....329
G.3 Kapan L7 Sebaiknya Tidak Dipakai....329
G.4 Risiko CPU....330
G.5 Template Topologi Uji L7....331
G.6 Template L7 Paling Sederhana....331
G.7 Template L7 dengan Address List Client Uji....332
G.8 Template L7 dengan Logging Terbatas....332
G.9 Template Menonaktifkan L7....333
G.10 Template Uji dari VPC GNS3....333
G.11 Template Uji dari Ubuntu VM....334
G.12 Template Monitoring CPU Saat L7 Aktif....335
G.13 Template L7 sebagai Demonstrasi, Bukan Solusi Utama....336
G.14 Contoh Lab L7 Sederhana....337
G.15 Kesalahan Umum L7 Firewall....338
G.16 Checklist Sebelum Mengaktifkan L7....338
G.17 Checklist Setelah Mengaktifkan L7....339
G.18 Template Perintah Cepat L7....340
G.19 Golden Rules L7 Firewall....341

Lampiran H. Template HotSpot..... 342
H.1 Topologi Dasar HotSpot..... 342
H.2 Template HotSpot Dasar RouterOS v7/v6..... 343
H.3 Template HotSpot dengan WAN dan NAT..... 344
H.4 Penjelasan Komponen HotSpot..... 345
H.5 Template User HotSpot..... 346
H.6 Template Profile HotSpot..... 347
H.7 Template Walled Garden..... 348
H.8 Template Lab HotSpot + VPC GNS3..... 349
H.9 Template Lab HotSpot + Ubuntu VM..... 350
H.10 Template Verifikasi HotSpot di RouterOS..... 351
H.11 Template Firewall Minimum untuk HotSpot..... 352
H.12 Template Troubleshooting HotSpot..... 353
H.13 Template Reset HotSpot Lab..... 354
H.14 Template Dokumentasi HotSpot..... 355
H.15 Checklist HotSpot Berhasil..... 357
H.16 Golden Rules HotSpot..... 357
Lampiran I. Template VPN..... 358
WireGuard v7, L2TP/IPsec v6/v7..... 358
I.1 Prinsip Dasar VPN..... 358
Bagian A — WireGuard RouterOS v7..... 358
I.2 Topologi WireGuard Site-to-Site..... 359
I.3 Konfigurasi Dasar IP..... 360
I.4 Membuat Interface WireGuard..... 361
I.5 Membuat Peer WireGuard..... 362
I.6 Firewall WireGuard..... 363
I.7 Verifikasi WireGuard..... 364
I.8 Template WireGuard Lengkap — R1..... 365
I.9 Template WireGuard Lengkap — R2..... 366
Bagian B — L2TP/IPsec RouterOS v6/v7..... 367
I.10 Kapan Memakai L2TP/IPsec..... 367
I.11 Topologi L2TP/IPsec Site-to-Site Sederhana..... 367
I.12 L2TP/IPsec Server — R1..... 368
I.13 L2TP/IPsec Client — R2..... 368
I.14 Firewall L2TP/IPsec..... 369
I.15 Verifikasi L2TP/IPsec..... 370
Bagian C — Remote Access L2TP/IPsec..... 371
I.16 Template Remote Access L2TP/IPsec..... 371
Bagian D — Troubleshooting VPN..... 372
I.17 Troubleshooting WireGuard..... 372
I.18 Troubleshooting L2TP/IPsec..... 373
I.19 Perbandingan WireGuard dan L2TP/IPsec..... 373
I.20 Template Dokumentasi VPN..... 374
I.21 Checklist VPN Berhasil..... 375

I.22 Golden Rules VPN..... 375
Lampiran J. Template IPv6..... 376
J.1 Topologi Dasar IPv6..... 376
J.2 Template IPv6 RouterOS v7/v6..... 377
J.3 Template IPv6 VPC GNS3..... 378
J.4 Template IPv6 Ubuntu VM..... 379
J.5 Template Dual-Stack IPv4/IPv6..... 380
J.6 Template Routing IPv6 Sederhana..... 381
J.7 Template Firewall IPv6 Dasar..... 382
J.8 Template Forward Firewall IPv6..... 383
J.9 Verifikasi RouterOS IPv6..... 383
J.10 Verifikasi VPC GNS3 IPv6..... 384
J.11 Verifikasi Ubuntu IPv6..... 384
J.12 Template IPv6 dengan Bridge..... 385
J.13 Template IPv6 untuk VLAN..... 386
J.14 Template Dokumentasi IPv6..... 387
J.15 Troubleshooting IPv6..... 389
J.16 Checklist IPv6 Berhasil..... 389
J.17 Golden Rules IPv6..... 390
Lampiran K. Template OSPF 3 Router..... 391
RouterOS v7 dan v6..... 391
K.1 Topologi Lab..... 391
K.2 IP Plan..... 391
Bagian A — Konfigurasi IP Dasar..... 392
K.3 R1 — IP Dasar..... 392
K.4 R2 — IP Dasar..... 392
K.5 R3 — IP Dasar..... 392
K.6 VPC-A dan VPC-C..... 393
Bagian B — OSPF RouterOS v7..... 394
K.7 Konsep OSPF v7..... 394
K.8 OSPF v7 — R1..... 394
K.9 OSPF v7 — R2..... 395
K.10 OSPF v7 — R3..... 395
K.11 Verifikasi OSPF v7..... 396
Bagian C — OSPF RouterOS v6..... 397
K.12 Konsep OSPF v6..... 397
K.13 OSPF v6 — R1..... 397
K.14 OSPF v6 — R2..... 397
K.15 OSPF v6 — R3..... 398
K.16 Verifikasi OSPF v6..... 398
Bagian D — Static Route Pembanding..... 399
K.17 Static Route Manual Sebelum OSPF..... 399
Bagian E — Troubleshooting OSPF..... 400
K.18 OSPF Neighbor Tidak Muncul..... 400

K.19 Route OSPF Tidak Muncul..... 400
K.20 VPC Tidak Bisa Ping Jaringan Lawan..... 401
K.21 Firewall untuk OSPF Lab..... 401
Bagian F — Dokumentasi dan Export..... 402
K.22 Template Dokumentasi OSPF..... 402
K.23 Export Konfigurasi..... 402
K.24 Checklist OSPF Berhasil..... 403
K.25 Golden Rules OSPF 3 Router..... 403
Lampiran L. Template BGP Customer–ISP..... 404
RouterOS v7 dan v6..... 404
L.1 Topologi Dasar Customer–ISP..... 404
L.2 IP Plan..... 404
L.3 Konfigurasi IP Dasar..... 405
Bagian A — BGP RouterOS v7..... 406
L.4 Konsep BGP RouterOS v7..... 406
L.5 BGP v7 — Customer Mengumumkan Prefix ke ISP..... 407
L.6 BGP v7 — ISP Menerima Prefix Customer..... 408
L.7 BGP v7 — ISP Memberi Default Route ke Customer..... 409
L.8 BGP v7 — Filter Input di Customer..... 410
L.9 Template Lengkap RouterOS v7 — R-CUST..... 411
L.10 Template Lengkap RouterOS v7 — R-ISP..... 412
Bagian B — BGP RouterOS v6..... 413
L.11 Konsep BGP RouterOS v6..... 413
L.12 BGP v6 — Customer..... 413
L.13 BGP v6 — ISP..... 414
L.14 Default Route dari ISP ke Customer — RouterOS v6..... 414
L.15 Template Lengkap RouterOS v6 — R-CUST..... 415
L.16 Template Lengkap RouterOS v6 — R-ISP..... 415
Bagian C — Dua ISP dan Satu Customer..... 416
L.17 Topologi Multi-Homing Sederhana..... 416
L.18 RouterOS v7 — Customer ke Dua ISP..... 416
L.19 RouterOS v6 — Customer ke Dua ISP..... 417
Bagian D — Route Filtering dan Etika BGP..... 417
L.20 Filter Output Minimal RouterOS v7..... 417
L.21 Filter Input Minimal RouterOS v7..... 417
L.22 Etika BGP..... 418
Bagian E — Verifikasi BGP..... 419
L.23 Verifikasi RouterOS v7..... 419
L.24 Verifikasi RouterOS v6..... 419
L.25 Verifikasi dari VPC..... 420
Bagian F — Troubleshooting BGP..... 420
L.26 Session BGP Tidak Established..... 420
L.27 Session Up, Route Tidak Muncul..... 421
L.28 Prefix Salah Teradvertise..... 421

L.29 Firewall BGP..... 422
Bagian G — Dokumentasi dan Export.....423
L.30 Template Dokumentasi BGP..... 423
L.31 Export Konfigurasi..... 423
L.32 Checklist BGP Berhasil..... 424
L.33 Golden Rules BGP Customer–ISP.....424
Lampiran M. Checklist Troubleshooting.....425
M.1 Prinsip Troubleshooting.....425
M.2 Flow Cepat Troubleshooting..... 426
M.3 Checklist 1 — Interface dan Link.....427
M.4 Checklist 2 — IP Address.....428
M.5 Checklist 3 — Gateway dan Route..... 429
M.6 Checklist 4 — DNS..... 431
M.7 Checklist 5 — DHCP.....432
M.8 Checklist 6 — NAT dan Masquerade..... 433
M.9 Checklist 7 — Firewall Input.....434
M.10 Checklist 8 — Firewall Forward..... 435
M.11 Checklist 9 — Log..... 436
M.12 Checklist 10 — IPv6..... 437
M.13 Checklist 11 — OSPF..... 438
M.14 Checklist 12 — BGP.....439
M.15 Checklist Kasus Cepat.....440
M.16 Template Laporan Troubleshooting..... 442
M.17 Checklist Final Setelah Perbaikan.....443
M.18 Golden Rules Troubleshooting.....443
Glossary.....444
Referensi.....450
Tentang Penulis.....455

Bab 1. Fondasi Teknik Jaringan untuk Pemula: TCP/IP, IP Address, dan Netmask

1.1 Tujuan Bab

Setelah menyelesaikan bab ini, pembaca mampu:

- Membaca alamat seperti `192.168.10.1/24`.
- Memahami **network address, host address, broadcast address, subnet mask, dan CIDR**.
- Membedakan **private IP** dan **public IP**.
- Menjelaskan fungsi **gateway, ARP, ICMP, dan DNS**.
- Menghitung subnet dasar
- Memakai perintah dasar Ubuntu 26.04
- Memakai **VPC GNS3** untuk uji sederhana
- Melihat IP dan route dasar di MikroTik RouterOS v7/v6.

1.2 Masalah Nyata: IP Sudah Diisi, Tapi Tidak Bisa Ping

Masalah paling umum saat belajar jaringan bukan langsung OSPF atau BGP. Masalah paling sering justru sederhana:

- IP beda subnet.
- Netmask salah.
- Gateway salah.
- DNS kosong.
- Route tidak ada.
- Interface salah.
- Firewall atau NAT belum dibuat.

Contoh kasus:

```
PC Ubuntu / VPC GNS3 → MikroTik CHR → Internet
```

Client diberi IP:

```
IP Address : 192.168.10.10
Netmask    : 255.255.255.0
Gateway    : 192.168.10.1
DNS        : 1.1.1.1
```

Pertanyaan teknisi jaringan:

- Apakah client satu subnet dengan gateway?
- Apakah gateway bisa di-*ping*?
- Apakah route keluar ada?
- Apakah DNS bekerja?
- Apakah router sudah meneruskan paket?

Prinsip Bab 1: jangan menebak. **Cek IP, cek netmask, cek gateway, cek route, cek DNS.**

1.3 Apa Itu Jaringan Komputer?

Jaringan komputer adalah hubungan antara dua atau lebih perangkat agar bisa saling bertukar data.

Perangkat dalam jaringan bisa berupa:

- PC Ubuntu.
- VPC GNS3.
- Ubuntu Server VM.
- MikroTik CHR.
- Router fisik.
- Switch.
- Access point.
- Server.
- Perangkat IoT.

Bentuk paling sederhana:

```
Client → Router → Server / Internet
```

Dalam lab buku ini, bentuk dasarnya:

```
VPC GNS3 / Ubuntu VM → MikroTik CHR → Jaringan luar
```

Client adalah perangkat yang meminta layanan.
Server adalah perangkat yang menyediakan layanan.
Router adalah perangkat yang menghubungkan jaringan berbeda.
Switch menghubungkan perangkat dalam satu jaringan lokal.
Gateway adalah pintu keluar dari satu subnet ke subnet lain.
DNS menerjemahkan nama menjadi alamat IP; RFC 1035 menjelaskan DNS sebagai protokol dan format komunikasi client-server untuk sistem nama domain. (IETF Datatracker)

1.4 Model TCP/IP Secara Sederhana

Model TCP/IP dapat dipahami sebagai lapisan kerja jaringan. Untuk pemula, tidak perlu dihafal rumit. Yang penting: tahu fungsi setiap bagian.

Lapisan	Fungsi	Contoh
Application	Layanan yang dipakai pengguna	DNS, HTTP, SSH
Transport	Mengatur komunikasi aplikasi	TCP, UDP
Internet	Mengatur alamat dan routing	IP, ICMP
Network Access	Mengirim data di media jaringan	Ethernet, Wi-Fi

Contoh saat client membuka server:

1. Client mencari alamat IP tujuan.
2. Jika memakai nama, DNS menerjemahkan nama menjadi IP.
3. Client mengecek apakah tujuan satu subnet.
4. Jika beda subnet, paket dikirim ke gateway.
5. Router meneruskan paket ke jaringan berikutnya.
6. Balasan kembali melalui jalur routing.

Kunci pemahaman:

- **IP address** menjawab: alamat perangkat ini apa?
- **Netmask** menjawab: siapa yang masih satu jaringan?
- **Gateway** menjawab: kalau tujuan beda jaringan, lewat mana?
- **DNS** menjawab: nama ini IP-nya berapa?
- **Route** menjawab: jalurnya lewat mana?

1.5 IPv4: Alamat Dasar dalam Lab

IPv4 adalah alamat 32-bit yang biasa ditulis dalam empat angka desimal.

Contoh:

```
192.168.10.1
10.10.10.2
172.16.1.254
```

Dalam lab, biasanya ditulis dengan CIDR:

```
192.168.10.1/24
```

Artinya:

- IP address: **`192.168.10.1`**
- Prefix: **`/24`**
- Bagian network: **24 bit**
- Bagian host: **8 bit**

CIDR dipakai untuk menulis prefix jaringan secara ringkas; RFC 4632 membahas CIDR sebagai strategi pengalamatan dan routing IPv4 32-bit untuk efisiensi penggunaan alamat dan routing. (RFC Editor)

1.6 Network Address, Host Address, Broadcast Address

Dalam satu subnet IPv4, ada tiga jenis alamat yang harus dipahami.

Network Address

Network address adalah identitas subnet. Biasanya tidak dipakai oleh host.

Contoh:

```
192.168.10.0/24
```

Artinya subnet bernama:

```
192.168.10.0
```

Host Address

Host address adalah alamat yang dapat dipakai perangkat.

Contoh dalam subnet `192.168.10.0/24`:

```
192.168.10.1
192.168.10.10
192.168.10.20
192.168.10.254
```

Broadcast Address

Broadcast address adalah alamat untuk mengirim ke semua host dalam satu subnet.

Contoh untuk `192.168.10.0/24`:

```
192.168.10.255
```

RouterOS dapat menghitung network prefix dan broadcast address secara otomatis ketika IP ditambahkan ke interface; dokumentasi MikroTik juga menyebut `/ip address print detail` untuk melihat alamat dan interface terkait. (MikroTik Help)

1.7 Membaca 192.168.10.1/24

Alamat:

```
192.168.10.1/24
```

Dibaca sebagai:

Item	Nilai
IP perangkat	`192.168.10.1`
Prefix	`/24`
Netmask	`255.255.255.0`
Network	`192.168.10.0`
Host pertama	`192.168.10.1`
Host terakhir	`192.168.10.254`
Broadcast	`192.168.10.255`

Jika client:

```
192.168.10.10/24
```

dan gateway:

```
192.168.10.1
```

maka client dan gateway **satu subnet**.

Jika client:

```
192.168.20.10/24
```

dan gateway:

```
192.168.10.1
```

maka client dan gateway **bukan satu subnet**. Ini salah satu penyebab paling umum ping gagal.

1.8 Private IP vs Public IP

Untuk lab, gunakan **private IP**. Jangan memakai alamat publik sembarangan.

RFC 1918 menetapkan tiga blok private IPv4 untuk jaringan internal, yaitu `10.0.0.0/8`, `172.16.0.0/12`, dan `192.168.0.0/16`; alamat ini dipakai di jaringan privat dan tidak dirutekan langsung di Internet publik. (IETF Datatracker)

Blok Private	Rentang	Contoh Pemakaian
`10.0.0.0/8`	`10.0.0.0–10.255.255.255`	lab besar, routing, backbone internal
`172.16.0.0/12`	`172.16.0.0–172.31.255.255`	lab menengah, simulasi WAN
`192.168.0.0/16`	`192.168.0.0–192.168.255.255`	LAN kecil, lab pemula

Contoh alamat lab dalam buku ini:

```
192.168.10.0/24
192.168.20.0/24
10.10.12.0/30
10.10.23.0/30
172.16.99.0/24
```

Aturan praktis: untuk lab pemula, gunakan private IP agar aman dan tidak bentrok dengan alamat publik nyata.

1.9 Gateway: Pintu Keluar Subnet

Gateway adalah alamat router yang menjadi **pintu keluar** dari satu subnet.

Contoh:

```
Client  : 192.168.10.10/24
Gateway : 192.168.10.1
```

Jika client menghubungi:

```
192.168.10.20
```

maka paket dikirim langsung karena masih satu subnet.

Jika client menghubungi:

```
1.1.1.1
```

maka paket dikirim ke gateway:

```
192.168.10.1
```

RouterOS membuat *connected route* otomatis untuk jaringan yang langsung terpasang pada interface aktif; dokumentasi MikroTik menyebut connected route sebagai jaringan yang host-nya dapat dicapai langsung pada domain Layer 2 yang sama. (MikroTik Help)

Kesimpulan:

- Tujuan satu subnet → kirim langsung.
- Tujuan beda subnet → kirim ke gateway.
- Gateway salah → paket tidak tahu jalan keluar.

1.10 ARP, ICMP, dan DNS

ARP

ARP dipakai untuk mencari alamat MAC dari alamat IP dalam jaringan lokal.

Contoh pertanyaan ARP:

```
Siapa pemilik 192.168.10.1?
```

Gateway menjawab:

```
Saya, MAC address saya xx:xx:xx:xx:xx:xx
```

RFC 826 menjelaskan ARP sebagai protokol untuk menerjemahkan alamat protokol, misalnya IP address, menjadi alamat lokal/hardware seperti Ethernet address. (IETF Datatracker)

ICMP

ICMP dipakai untuk pesan kontrol jaringan. Perintah `ping` memakai ICMP untuk menguji apakah host bisa dijangkau.

Contoh:

```
ping 192.168.10.1
ping 1.1.1.1
```

RFC 792 menjelaskan bahwa ICMP message dikirim memakai IP header dasar dan dipakai untuk format pesan kontrol seperti error dan echo. (IETF Datatracker)

DNS

DNS menerjemahkan nama menjadi IP address.

Contoh:

```
dig example.net
```

Jika:

```
ping 1.1.1.1         berhasil
ping nama-domain     gagal
```

maka masalah kemungkinan besar ada di **DNS**, bukan IP atau gateway.

1.11 Kalkulasi Netmask Dasar

Rumus cepat:

```
Jumlah alamat = 2^(32 - prefix)
Host usable   = jumlah alamat - 2
```

Dikurangi 2 karena ada:

- **network address**
- **broadcast address**

Catatan:

- `/32` hanya satu alamat, biasanya untuk host route atau loopback.
- `/30` sangat umum untuk link antar-router.
- `/24` paling mudah untuk LAN pemula.

1.12 Tabel Kalkulasi Subnet

CIDR	Netmask	Jumlah Alamat	Host Usable	Umum Dipakai Untuk
`/24`	`255.255.255.0`	256	254	LAN
`/25`	`255.255.255.128`	128	126	LAN dibagi dua
`/26`	`255.255.255.192`	64	62	Segmentasi kecil
`/27`	`255.255.255.224`	32	30	Kelompok kecil
`/28`	`255.255.255.240`	16	14	Lab kecil
`/29`	`255.255.255.248`	8	6	Link kecil/perangkat terbatas
`/30`	`255.255.255.252`	4	2	Link antar-router
`/32`	`255.255.255.255`	1	1 alamat tunggal	Loopback/router ID

1.13 Contoh Hitung Subnet

Contoh 1: 192.168.10.0/24

Item	Nilai
Network	`192.168.10.0`
Host pertama	`192.168.10.1`
Host terakhir	`192.168.10.254`
Broadcast	`192.168.10.255`
Host usable	254

Cocok untuk LAN sederhana.

Contoh 2: 192.168.10.0/25

`/25` membagi `/24` menjadi dua subnet.

Subnet pertama:

Item	Nilai
Network	`192.168.10.0`
Host pertama	`192.168.10.1`
Host terakhir	`192.168.10.126`
Broadcast	`192.168.10.127`

Subnet kedua:

Item	Nilai
Network	`192.168.10.128`
Host pertama	`192.168.10.129`
Host terakhir	`192.168.10.254`
Broadcast	`192.168.10.255`

Contoh 3: 10.10.12.0/30

Item	Nilai
Network	`10.10.12.0`
Router 1	`10.10.12.1`
Router 2	`10.10.12.2`
Broadcast	`10.10.12.3`

Cocok untuk link:

```
R1 -------- R2
```

Contoh 4: /32

Contoh:

```
1.1.1.1/32
```

Artinya hanya satu alamat. Dalam routing, `/32` sering dipakai untuk:

- loopback,
- router ID,
- host route,
- alamat identitas router.

1.14 Kapan Memakai /24, /30, dan /32

/24 untuk LAN

Gunakan `/24` untuk jaringan lokal sederhana.

Contoh:

```
192.168.10.0/24
```

Cocok untuk:

- kelas,
- lab,
- kantor kecil,
- jaringan user,
- jaringan server sederhana.

/30 untuk Link Antar-Router

Gunakan `/30` untuk koneksi dua router.

Contoh:

```
10.10.12.0/30
```

Pemakaian:

```
R1 10.10.12.1/30 ---- 10.10.12.2/30 R2
```

/32 untuk Loopback atau Router ID

Gunakan `/32` untuk alamat tunggal.

Contoh:

```
1.1.1.1/32
```

Cocok untuk:

- router ID OSPF,
- loopback router,
- host route,
- identitas stabil dalam lab routing.

1.15 Hands-on Ubuntu 26.04

Bagian ini dilakukan di terminal Ubuntu 26.04. Tujuannya bukan menghafal perintah, tetapi **membaca kondisi jaringan**.

Lab 1.1 — Melihat IP Address

```
ip addr
```

Yang dicari:

- nama interface,
- IP address,
- prefix,
- status interface,
- MAC address.

Contoh output:

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP>
    inet 192.168.10.10/24 brd 192.168.10.255 scope
global enp0s3
```

Cara membaca:

Bagian	Arti
`enp0s3`	nama interface
`192.168.10.10/24`	IP dan prefix
`192.168.10.255`	broadcast
`UP`	interface aktif

Lab 1.2 — Melihat Routing Table

```
ip route
```

Contoh output:

```
default via 192.168.10.1 dev enp0s3
192.168.10.0/24 dev enp0s3 proto kernel scope link src
192.168.10.10
```

Cara membaca:

Baris	Arti
`default via 192.168.10.1`	semua tujuan luar dikirim ke gateway
`192.168.10.0/24 dev enp0s3`	subnet lokal ada di interface ini

Jika tidak ada `default via`, client tidak punya jalur keluar.

Lab 1.3 — Ping Gateway

```
ping 192.168.10.1
```

Jika berhasil:

```
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.5
ms
```

Jika gagal, cek:

- IP client.
- Netmask client.
- IP gateway.
- Interface.
- Koneksi VM/GNS3.
- Firewall.

Lab 1.4 — Ping IP Luar

```
ping 1.1.1.1
```

Jika ping gateway berhasil, tetapi ping IP luar gagal, kemungkinan:

- route keluar belum ada,
- NAT belum aktif,
- firewall memblokir,
- interface WAN belum benar.

Lab 1.5 — Uji DNS

```
dig example.net
```

Jika `dig` belum ada, instal paket DNS tools sesuai sistem Ubuntu yang dipakai.

Gejala umum:

Gejala	Kemungkinan
`ping 1.1.1.1` berhasil, `dig` gagal	DNS bermasalah
`ping gateway` gagal	IP/netmask/interface bermasalah
`ping gateway` berhasil, `ping luar` gagal	route/NAT/firewall bermasalah

Lab 1.6 — Melihat Port Aktif

```
ss -tulpn
```

Fungsi:

- melihat service yang aktif,
- melihat port TCP/UDP,
- membantu debugging server uji.

Contoh:

```
tcp LISTEN 0 128 0.0.0.0:22
```

Artinya ada service mendengarkan pada port 22.

1.16 Hands-on VPC GNS3 untuk Testing

VPC GNS3 dipakai sebagai client ringan. Ini cocok untuk latihan IP, gateway, DHCP, ping, dan trace. Dalam outline buku, VPC GNS3 memang menjadi bagian utama untuk uji ringan, DHCP, ping, traceroute, serta menghemat RAM pada lab pemula .

Lab 1.7 — Melihat IP VPC

Di console VPC:

```
PC1> show ip
```

Jika belum ada IP, hasilnya biasanya kosong atau belum lengkap.

Lab 1.8 — Memberi IP Manual di VPC

```
PC1> ip 192.168.10.10/24 192.168.10.1
```

Artinya:

Bagian	Arti
`192.168.10.10/24`	IP VPC
`192.168.10.1`	gateway

Cek kembali:

```
PC1> show ip
```

Lab 1.9 — Ping Gateway dari VPC

```
PC1> ping 192.168.10.1
```

Jika gagal, cek:

- apakah VPC terhubung ke link yang benar,
- apakah interface MikroTik sudah diberi IP,
- apakah subnet sama,
- apakah gateway benar.

Lab 1.10 — Trace Jalur dari VPC

```
PC1> trace 1.1.1.1
```

Gunakan `trace` untuk melihat jalur paket. Jika berhenti di gateway, masalah biasanya ada setelah router.

1.17 Preview CLI MikroTik RouterOS v7 dan v6

Bab ini belum masuk konfigurasi penuh MikroTik. Namun pembaca perlu melihat bahwa prinsipnya sama: **IP, interface, route, ping**.

Melihat Interface

RouterOS v7/v6:

```
/interface print
```

Melihat IP Address

RouterOS v7/v6:

```
/ip address print
/ip address print detail
```

Contoh:

```
ADDRESS            NETWORK         INTERFACE
192.168.10.1/24    192.168.10.0    ether2
```

Artinya:

- IP router: `192.168.10.1/24`
- Network: `192.168.10.0`
- Interface: `ether2`

Melihat Route

RouterOS v7/v6:

```
/ip route print
```

Yang dicari:

```
0.0.0.0/0
```

Itu adalah **default route**.

Ping dari RouterOS

RouterOS v7/v6:

```
/tool ping 192.168.10.10
/tool ping 1.1.1.1
```

Makna:

- Ping ke client menguji LAN.
- Ping ke IP luar menguji route keluar.
- Jika router bisa ping luar tetapi client tidak bisa, kemungkinan masalah ada di NAT, firewall, atau gateway client.

1.18 Alur Paket dari Client ke Server

Contoh:

```
Client : 192.168.10.10/24
Gateway: 192.168.10.1
Tujuan : 1.1.1.1
```

Alur paket:

1. Client mengecek apakah `1.1.1.1` satu subnet dengan `192.168.10.10/24`.
2. Karena beda subnet, client mengirim paket ke gateway `192.168.10.1`.
3. Client memakai ARP untuk mencari MAC address gateway.
4. Paket dikirim ke MikroTik.
5. MikroTik melihat routing table.
6. MikroTik meneruskan paket ke interface keluar.
7. Jika NAT dibutuhkan, alamat sumber diubah oleh router.
8. Balasan kembali melalui jalur routing.

Inti penting:

Client tidak perlu tahu seluruh jalur Internet.
Client hanya perlu tahu **gateway yang benar**.
Router yang menentukan jalur berikutnya.

1.19 Kesalahan Umum Pemula

Kesalahan	Gejala	Solusi
IP beda subnet dengan gateway	Tidak bisa ping gateway	Samakan subnet
Netmask salah	Host tertentu tidak bisa diakses	Hitung ulang CIDR
Gateway kosong	Tidak bisa keluar jaringan	Tambahkan default gateway
DNS salah	Ping IP bisa, nama gagal	Perbaiki DNS
Interface salah	IP benar tapi tidak jalan	Cek interface
VPC salah link	Ping gateway gagal	Cek topologi GNS3
Route tidak ada	Paket berhenti di router	Cek routing table
NAT belum aktif	Router bisa keluar, client tidak	Dibahas Bab 5
Firewall terlalu ketat	Ping/akses terblokir	Cek rule firewall

1.20 Latihan Cepat

Jawab sebelum lanjut ke bab berikutnya.

1. Apa network address dari `192.168.50.77/24`?
2. Apa broadcast address dari `192.168.50.0/24`?
3. Berapa host usable pada `/30`?
4. Apakah `192.168.10.10/24` satu subnet dengan `192.168.11.1/24`?
5. Kapan memakai `/32`?
6. Jika `ping 1.1.1.1` berhasil tetapi `dig example.net` gagal, masalah kemungkinan di mana?
7. Jika VPC tidak bisa ping gateway, apa tiga hal pertama yang harus dicek?

Jawaban:

1. `192.168.50.0`
2. `192.168.50.255`
3. 2 host
4. Tidak
5. Untuk alamat tunggal, loopback, router ID, atau host route
6. DNS
7. IP VPC, subnet/gateway, dan link/topologi GNS3

1.21 Checklist Akhir Bab

Pembaca dianggap siap lanjut ke Bab 2 jika mampu:

- Menjelaskan fungsi **client, server, router, switch, gateway, DNS**.
- Membaca IP seperti **`192.168.10.1/24`**.
- Menghitung subnet dasar `/24` sampai `/30`.
- Menjelaskan fungsi `/32`.
- Menentukan:
  - network address,
  - host pertama,
  - host terakhir,
  - broadcast address.
- Membedakan **private IP** dan **public IP**.
- Menjelaskan fungsi **gateway, ARP, ICMP, dan DNS**.
- Menggunakan Ubuntu:
  - `ip addr`
  - `ip route`
  - `ping`
  - `traceroute`
  - `dig`
  - `ss -tulpn`
- Menggunakan VPC GNS3:
  - `show ip`
  - `ip 192.168.10.10/24 192.168.10.1`
  - `ping`
  - `trace`
- Menggunakan RouterOS v7/v6:
  - `/interface print`
  - `/ip address print`
  - `/ip route print`
  - `/tool ping`

Ringkasan Bab

Bab ini mengajarkan pondasi jaringan: **IP address, netmask, CIDR, gateway, ARP, ICMP, DNS, route, dan kalkulasi subnet**. MikroTik hanyalah alat. Yang lebih penting adalah cara berpikir jaringan.

Kunci Bab 1:

> **Paket tidak berjalan karena harapan. Paket berjalan karena alamat, subnet, gateway, dan route yang benar.**

Bab 2. Menyiapkan Ekosistem Lab Jaringan dengan Satu PC Ubuntu 26.04

2.1 Tujuan Bab

Setelah menyelesaikan bab ini, pembaca mampu:

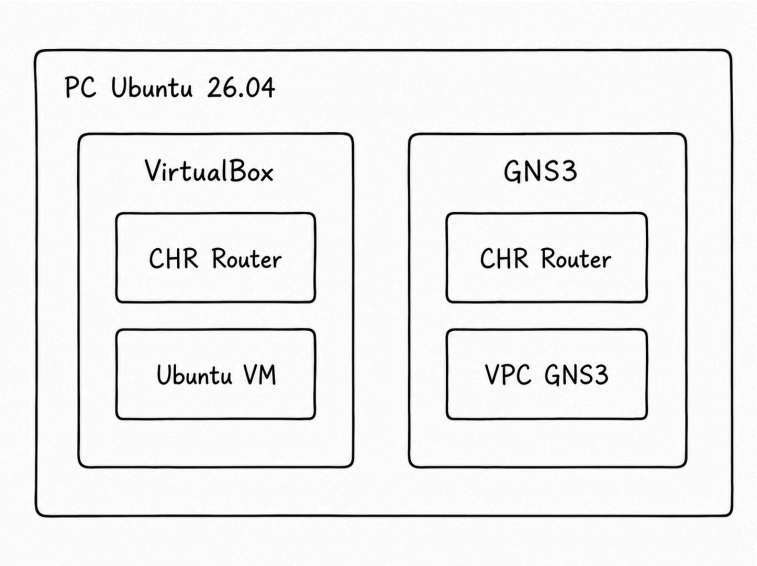
- Membangun **lab jaringan murah** dengan satu PC Ubuntu 26.04.
- Menjalankan **MikroTik CHR** sebagai router virtual.
- Menjalankan **Ubuntu Server VM** sebagai client/server uji.
- Memakai **VPC GNS3** sebagai client ringan untuk ping, DHCP, dan traceroute.
- Memahami fungsi **VirtualBox** dan **GNS3**.
- Memilih mode jaringan yang benar:
- Membuat desain lab:
    - 1 CHR + 1 VPC,
    - 1 CHR + 2 VPC,
    - 1 CHR + 1 Ubuntu Server VM,
    - 3 CHR untuk routing,
    - 2 CHR + 2 VPC untuk VPN,
    - 1 CHR + VPC IPv4 + VPC IPv6.
- Membuat dokumentasi

2.2 Strategi Lab Murah: Satu PC Cukup

Belajar jaringan sering dianggap mahal karena harus membeli router, switch, kabel, dan perangkat wireless. Untuk pemula, mahasiswa, *network admin*, dan siswa SMK, pendekatan yang lebih cerdas adalah memulai dari **lab virtual**.

Dengan satu PC Ubuntu 26.04, pembaca dapat membuat beberapa router dan client sekaligus:

```
PC Ubuntu 26.04
|
+-- VirtualBox
|   +-- MikroTik CHR
|   +-- Ubuntu Server VM
|
+-- GNS3
    +-- MikroTik CHR
    +-- VPC GNS3
    +-- Link antar-router
```



Gambar. Satu PC dapat menjadi laboratorium jaringan lengkap.

Keuntungan strategi ini:

- **Murah**: tidak perlu banyak perangkat fisik.
- **Aman**: salah konfigurasi tidak merusak jaringan nyata.
- **Bisa diulang**: lab dapat dihapus dan dibuat ulang.
- **Mudah diuji**: cukup pakai `ping`, `trace`, `ip route`, dan CLI RouterOS.
- **Bagus untuk belajar**: pemula bisa mencoba NAT, DHCP, firewall, VPN, IPv6, OSPF, dan BGP tanpa takut.

Catatan kritis:

> **Lab yang baik bukan yang paling rumit. Lab yang baik adalah yang rapi, bisa diulang, dan mudah diperbaiki.**

2.3 Instalasi Paket Pendukung di Ubuntu 26.04

Ubuntu 26.04 menjadi sistem utama untuk menjalankan terminal, VirtualBox, GNS3, dan tools jaringan. Ubuntu adalah sistem operasi berbasis Linux dan *open source*, sehingga cocok untuk lab pendidikan, riset, dan pelatihan jaringan. (Ubuntu)

Paket yang disarankan:

- Terminal.
- SSH client.
- *Text editor*.
- Tools jaringan.
- VirtualBox.
- GNS3.
- Wireshark jika ingin analisis paket.
- `traceroute`, `dnsutils`, `curl`, `tcpdump`.

Contoh instalasi tools dasar:

```
sudo apt update
sudo apt install -y openssh-client traceroute dnsutils curl
tcpdump net-tools
```

Cek perintah penting:

```
ip addr
ip route
ping 1.1.1.1
traceroute 1.1.1.1
dig example.net
```

Catatan:

- Gunakan Ubuntu sebagai **host lab utama**.
- Gunakan terminal untuk semua pekerjaan penting.
- Simpan file lab dalam folder khusus.

Contoh struktur folder:

```
~/Lab-MikroTik/
├── CHR/
├── Ubuntu-VM/
├── GNS3-Projects/
├── Export-RouterOS/
├── IP-Plan/
└── Catatan/
```

2.4 Konsep Virtual Machine untuk Belajar Jaringan

Virtual machine adalah komputer virtual yang berjalan di dalam komputer utama. Dalam buku ini, VM digunakan untuk membuat:

- **MikroTik CHR** sebagai router virtual.
- **Ubuntu Server VM** sebagai client/server uji.
- Beberapa router untuk routing.
- Beberapa client untuk DHCP, NAT, firewall, dan VPN.

VirtualBox adalah perangkat lunak virtualisasi *open source* yang memungkinkan beberapa sistem operasi berjalan pada satu perangkat fisik. Ini membuat satu PC dapat menjadi lab jaringan lengkap. (Oracle)

Prinsipnya sederhana:

```
Satu PC fisik
|
+-- VM Router 1
+-- VM Router 2
+-- VM Ubuntu Server
+-- VPC GNS3
```

Manfaat VM untuk belajar jaringan:

- Bisa membuat banyak perangkat virtual.
- Bisa mengulang konfigurasi dari nol.
- Bisa melakukan kesalahan tanpa risiko besar.
- Bisa membuat skenario jaringan kecil hingga menengah.

2.5 Fungsi VirtualBox dalam Buku Ini

VirtualBox dipakai untuk:

- Menjalankan **MikroTik CHR**.
- Menjalankan **Ubuntu Server VM**.
- Membuat beberapa adapter jaringan.
- Membuat lab WAN/LAN sederhana.
- Menghubungkan VM ke GNS3 jika dibutuhkan.

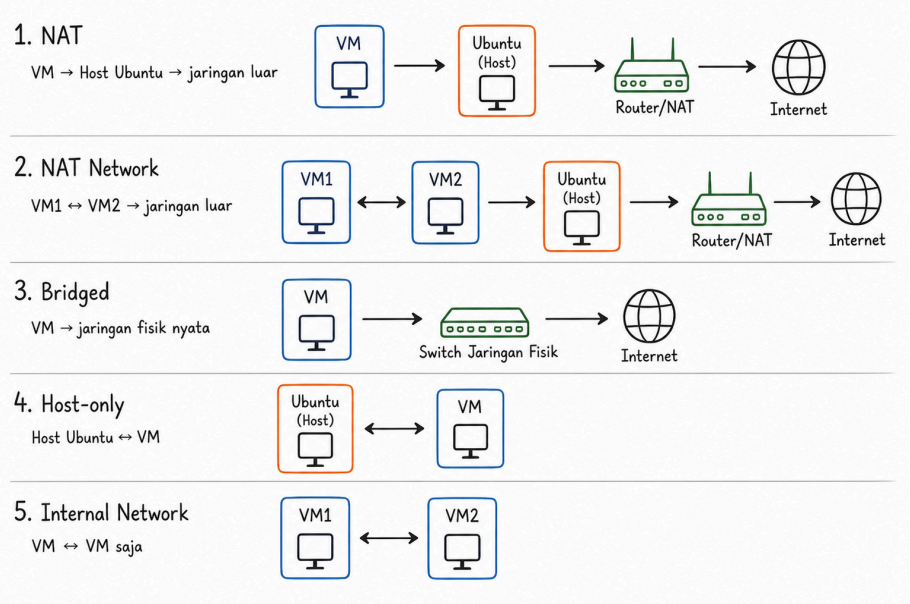
Contoh pemakaian:

Komponen	Fungsi
MikroTik CHR	Router virtual
Ubuntu Server VM	Client/server uji
Adapter NAT	Simulasi akses keluar
Host-only Adapter	Manajemen dari host
Internal Network	Jaringan privat antar-VM

VirtualBox menyediakan beberapa mode jaringan seperti NAT, bridged, host-only, dan internal network; setiap mode menentukan apakah VM bisa keluar ke jaringan luar, terlihat oleh jaringan fisik, hanya terhubung ke host, atau hanya terhubung antar-VM. (VirtualBox)

2.6 Mode Jaringan VirtualBox

Mode jaringan adalah bagian penting. Banyak lab gagal bukan karena MikroTik salah, tetapi karena **mode adapter salah**.



NAT

NAT adalah mode paling mudah agar VM bisa keluar ke jaringan luar.

Dipakai untuk:

- WAN simulasi MikroTik CHR.
- Ubuntu VM yang perlu akses paket.
- Lab awal yang butuh koneksi keluar.

Contoh:

```
CHR ether1 → NAT → jaringan luar
```

Kelebihan:

- Mudah.
- Aman untuk pemula.
- Cocok sebagai WAN simulasi.

Kekurangan:

- VM tidak mudah diakses dari jaringan fisik luar.
- Kurang cocok untuk lab antar-VM yang kompleks.

NAT Network

NAT Network membuat beberapa VM berada dalam satu jaringan NAT yang sama.

Dipakai untuk:

- Beberapa VM yang butuh koneksi keluar.
- Lab dengan lebih dari satu client.
- Simulasi jaringan kecil.

Kelebihan:

- Beberapa VM bisa satu jaringan.
- Tetap bisa keluar.

Kekurangan:

- Harus dibuat lebih dulu.
- Perlu dokumentasi IP agar tidak bingung.

Bridged Adapter

Bridged Adapter membuat VM seolah-olah menjadi perangkat langsung di jaringan fisik.

Dipakai jika:

- VM harus terlihat dari jaringan nyata.
- Lab perlu terhubung ke perangkat fisik.
- Pengujian membutuhkan akses dari perangkat lain.

Catatan kritis:

> **Jangan aktifkan DHCP Server sembarangan pada jaringan bridged.** Kesalahan DHCP bisa mengganggu jaringan nyata di rumah, sekolah, atau kantor.

Host-only Adapter

Host-only Adapter membuat jaringan khusus antara host Ubuntu dan VM.

Dipakai untuk:

- Akses SSH dari Ubuntu host ke VM.
- Manajemen CHR.
- Lab privat yang tidak mengganggu jaringan luar.

Kelebihan:

- Aman.
- Stabil.
- Cocok untuk manajemen.

Kekurangan:

- Tidak otomatis punya akses keluar.
- Perlu NAT atau route tambahan jika ingin keluar.

Internal Network

Internal Network membuat jaringan privat hanya antar-VM.

Dipakai untuk:

- LAN lab.
- Link antar-router.
- OSPF/BGP.
- VPN site-to-site.
- Simulasi multi-router.

Contoh nama internal network:

```
net-lan10
net-lan20
net-r1-r2
net-r2-r3
net-vpn-a
net-vpn-b
```

Kunci:

> **Internal Network adalah pilihan aman untuk lab routing karena tidak menyentuh jaringan nyata.**

2.7 Kapan Memakai Mode Jaringan

Kebutuhan	Mode yang Disarankan	Catatan
Router CHR butuh WAN simulasi	**NAT**	Paling aman untuk awal
Beberapa VM butuh satu jaringan keluar	**NAT Network**	Cocok untuk lab menengah
VM harus masuk jaringan nyata	**Bridged Adapter**	Hati-hati DHCP
Host Ubuntu perlu SSH ke VM	**Host-only Adapter**	Cocok untuk manajemen
Link antar-router	**Internal Network**	Cocok untuk OSPF/BGP
LAN client virtual	**Internal Network**	Aman dan bersih
Lab VPN	**Internal Network + NAT**	NAT sebagai Internet simulasi

Aturan cepat:

- **WAN simulasi** → NAT.
- **Manajemen** → Host-only.
- **LAN lab** → Internal Network.
- **Routing antar-router** → Internal Network.
- **Akses jaringan nyata** → Bridged, hanya jika benar-benar perlu.

2.8 Fungsi GNS3

GNS3 dipakai untuk menggambar dan menjalankan topologi jaringan. Dalam buku ini, GNS3 menjadi tempat utama untuk:

- Menghubungkan beberapa router CHR.
- Menambahkan VPC ringan.
- Membuat lab routing.
- Membuat lab VPN.
- Membuat simulasi IPv4 dan IPv6.
- Melihat hubungan antar-node dengan jelas.

GNS3 juga menyediakan cara menghubungkan topologi ke jaringan luar; dokumentasi GNS3 menjelaskan bahwa NAT node dapat dipakai agar node dalam topologi mengakses jaringan luar, dan metode ini lebih sederhana dibanding opsi tertentu yang lebih terbuka ke jaringan fisik. (GNS3 Documentation)

Kapan memakai GNS3?

- Jika topologi sudah lebih dari satu router.
- Jika ingin menggambar link antar-router.
- Jika ingin banyak VPC ringan.
- Jika ingin simulasi OSPF, BGP, VPN, dan IPv6.

2.9 Fungsi VPC GNS3

VPC GNS3 adalah client ringan. Ia tidak menggantikan Ubuntu Server, tetapi sangat berguna untuk testing cepat.

GNS3 menjelaskan VPCS sebagai *Virtual PC Simulator* yang ringan, mendukung DHCP dan ping, hanya memakai sekitar 2 MB RAM per instance, dan tidak membutuhkan image tambahan. (GNS3 Documentation)

VPC cocok untuk:

- Uji DHCP.
- Uji IP statik.
- Uji ping.
- Uji traceroute.
- Uji gateway.
- Uji IPv6 dasar.
- Menghemat RAM.

Perintah penting VPC:

```
show ip
ip dhcp
ip 192.168.10.10/24 192.168.10.1
ping 192.168.10.1
trace 1.1.1.1
```

Catatan:

> **Gunakan VPC untuk uji cepat. Gunakan Ubuntu Server VM untuk uji layanan nyata seperti web server, SSH, DNS, dan tcpdump.**

2.10 Download MikroTik CHR

MikroTik CHR adalah RouterOS yang berjalan sebagai mesin virtual. CHR cocok untuk VirtualBox dan GNS3 karena dapat menjadi router virtual dalam lab satu PC.

MikroTik menyebut CHR sebagai RouterOS untuk mesin virtual berbasis x86 64-bit dan dapat berjalan pada berbagai hypervisor; ini membuatnya cocok untuk lab virtual dan simulasi jaringan. (MikroTik)

Langkah umum:

1. Buka halaman download MikroTik.
2. Pilih **Cloud Hosted Router / CHR**.
3. Download image yang sesuai untuk VirtualBox atau GNS3.
4. Simpan di folder:

```
~/Lab-MikroTik/CHR/
```

Rekomendasi:

- Gunakan **RouterOS v7** sebagai lab utama.
- Gunakan **RouterOS v6** sebagai pembanding sintaks.
- Jangan mencampur terlalu banyak versi dalam satu lab pemula.

2.11 Import CHR ke VirtualBox

Langkah umum:

1. Buka VirtualBox.
2. Buat VM baru.
3. Nama VM:

```
CHR-R1
```

4. Pilih tipe sistem Linux 64-bit yang sesuai.
5. RAM awal:

```
256 MB – 512 MB
```

6. Gunakan disk CHR.
7. Tambahkan adapter:
   - Adapter 1: NAT.
   - Adapter 2: Internal Network `net-lan10`.
   - Adapter 3: Internal Network `net-r1-r2`, jika perlu.
8. Jalankan VM.
9. Login ke RouterOS melalui console.

Contoh tabel adapter:

Adapter	Mode	Fungsi
Adapter 1	NAT	WAN simulasi
Adapter 2	Internal Network `net-lan10`	LAN client
Adapter 3	Internal Network `net-r1-r2`	Link ke router lain
Adapter 4	Host-only	Manajemen opsional

Konfigurasi awal RouterOS v7/v6:

```
/system identity set name=R1-CHR
/interface print
/ip address add address=192.168.10.1/24 interface=ether2
comment="LAN Lab"
/ip address print
/export file=R1-awal
```

2.12 Import CHR ke GNS3

GNS3 cocok untuk topologi multi-router. Gunakan CHR sebagai router utama dalam project.

Langkah umum:

1. Buka GNS3.
2. Tambahkan appliance MikroTik CHR.
3. Pilih image CHR.
4. Buat template router.
5. Tambahkan CHR ke project.
6. Tambahkan VPC sebagai client.
7. Hubungkan CHR dan VPC.
8. Jalankan topologi.
9. Buka console router.
10. Konfigurasi lewat CLI.

Topologi awal:

```
VPC1 ---- ether1 CHR-R1
```

Contoh konfigurasi RouterOS v7/v6:

```
/system identity set name=R1-GNS3
/ip address add address=192.168.10.1/24 interface=ether1
comment="LAN ke VPC1"
/ip address print
```

Konfigurasi VPC:

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> show ip
PC1> ping 192.168.10.1
```

Output yang diharapkan:

```
84 bytes from 192.168.10.1 icmp_seq=1 ttl=64 time=...
```

Jika gagal, cek:

- link GNS3 sudah aktif,
- interface CHR benar,
- IP satu subnet,
- VPC memakai gateway yang benar.

2.13 Membuat Ubuntu Server VM sebagai Client/Server Uji

Ubuntu Server VM dipakai saat butuh client yang lebih lengkap daripada VPC.

Dipakai untuk:

- Server uji.
- SSH.
- Web server.
- `tcpdump`.
- `curl`.
- Uji port forwarding.
- Uji firewall.
- Uji IPv6.

Rekomendasi VM:

Item	Nilai
Nama VM	`U-SRV1`
RAM	1024–2048 MB
Disk	10–20 GB
Adapter 1	Internal Network atau koneksi ke GNS3
Fungsi	Client/server uji

Perintah dasar Ubuntu:

```
ip addr
ip route
ping 192.168.10.1
traceroute 192.168.10.1
ss -tulpn
```

Konfigurasi IP sementara:

```
sudo ip addr add 192.168.10.20/24 dev enp0s3
sudo ip route add default via 192.168.10.1
```

Catatan:

- Nama interface bisa berbeda.
- Cek nama interface dengan `ip addr`.
- Untuk server uji, dokumentasikan port yang aktif.

2.14 Menghubungkan Ubuntu Server VM ke GNS3

Ada dua pola umum.

Pola 1 — Ubuntu VM di VirtualBox, Terhubung ke Lab Internal

Gunakan Internal Network yang sama dengan router CHR.

Contoh:

Node	Interface	Network
CHR-R1	ether2	`net-lan10`
Ubuntu VM	enp0s3	`net-lan10`

Topologi:

```
CHR-R1 ether2 ---- net-lan10 ---- Ubuntu VM
```

Pola 2 — Ubuntu VM Dihubungkan ke GNS3

Gunakan node cloud atau mekanisme integrasi GNS3 sesuai kebutuhan. Gunakan hanya jika topologi membutuhkan Ubuntu VM masuk ke project GNS3.

Topologi:

```
VPC1 ---- CHR-R1 ---- Ubuntu Server VM
```

Prinsip:

- Gunakan VPC untuk test cepat.
- Gunakan Ubuntu VM untuk server nyata.
- Jangan membuat topologi terlalu rumit di awal.

2.15 Desain Lab 1: 1 CHR + 1 VPC

Lab paling awal di GNS3.

Topologi

```
+---------+        +----------------+
|  VPC1   | ------ | MikroTik CHR R1 |
+---------+        +----------------+
192.168.10.10      192.168.10.1
```

Tabel IP

Perangkat	Interface	IP	Gateway
R1	ether1	192.168.10.1/24	-
VPC1	eth0	192.168.10.10/24	192.168.10.1

RouterOS v7/v6

```
/system identity set name=R1
/ip address add address=192.168.10.1/24 interface=ether1
comment="LAN VPC1"
/ip address print
```

VPC

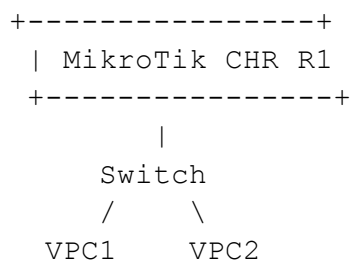
```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> show ip
PC1> ping 192.168.10.1
```

Target sukses: VPC bisa ping MikroTik.

2.16 Desain Lab 2: 1 CHR + 2 VPC

Lab untuk memahami satu LAN dengan dua client.

Topologi



Tabel IP

Perangkat	IP	Gateway
R1	192.168.10.1/24	-
VPC1	192.168.10.10/24	192.168.10.1
VPC2	192.168.10.20/24	192.168.10.1

RouterOS v7/v6

```
/ip address add address=192.168.10.1/24 interface=ether1
comment="LAN dua VPC"
```

VPC1

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> ping 192.168.10.1
PC1> ping 192.168.10.20
```

VPC2

```
PC2> ip 192.168.10.20/24 192.168.10.1
PC2> ping 192.168.10.1
PC2> ping 192.168.10.10
```

Target sukses: VPC1 dan VPC2 saling ping.

2.17 Desain Lab 3: 1 CHR + 1 Ubuntu Server VM

Lab ini dipakai untuk uji server, firewall, NAT, dan port forwarding pada bab berikutnya.

Topologi

```
+----------------+        +------------------+
| MikroTik CHR R1| ------ | Ubuntu Server VM |
+----------------+        +------------------+
192.168.20.1              192.168.20.10
```

Tabel IP

Perangkat	Interface	IP	Gateway
R1	ether1	192.168.20.1/24	-
Ubuntu VM	enp0s3	192.168.20.10/24	192.168.20.1

RouterOS v7/v6

```
/ip address add address=192.168.20.1/24 interface=ether1
comment="LAN Ubuntu Server"
```

Ubuntu

```
sudo ip addr add 192.168.20.10/24 dev enp0s3
sudo ip route add default via 192.168.20.1
ping 192.168.20.1
```

Target sukses: Ubuntu VM bisa ping MikroTik.

2.18 Desain Lab 4: 3 CHR untuk Routing

Lab ini menjadi dasar Bab 11.

Topologi

```
LAN-A            R1-R2              R2-R3             LAN-C
192.168.10.0/24  10.10.12.0/30      10.10.23.0/30     192.168.30.0/24

VPC-A ——————— R1 ————————— R2 ————————— R3 ———— VPC-C
```

Tabel IP

Perangkat	Interface	IP
R1	ether1 LAN-A	192.168.10.1/24
R1	ether2 R1-R2	10.10.12.1/30
R2	ether1 R1-R2	10.10.12.2/30
R2	ether2 R2-R3	10.10.23.1/30
R3	ether1 R2-R3	10.10.23.2/30
R3	ether2 LAN-C	192.168.30.1/24
VPC-A	eth0	192.168.10.10/24
VPC-C	eth0	192.168.30.10/24

RouterOS v7/v6 — R1

```
/system identity set name=R1
/ip address add address=192.168.10.1/24 interface=ether1 comment="LAN-A"
/ip address add address=10.10.12.1/30 interface=ether2 comment="R1-to-R2"
```

RouterOS v7/v6 — R2

```
/system identity set name=R2
/ip address add address=10.10.12.2/30 interface=ether1 comment="R2-to-R1"
/ip address add address=10.10.23.1/30 interface=ether2 comment="R2-to-R3"
```

RouterOS v7/v6 — R3

```
/system identity set name=R3
/ip address add address=10.10.23.2/30 interface=ether1 comment="R3-to-R2"
/ip address add address=192.168.30.1/24 interface=ether2 comment="LAN-C"
```

Routing belum dibahas penuh di bab ini. Tujuannya hanya menyiapkan topologi dan IP plan.

2.19 Desain Lab 5: 2 CHR + 2 VPC untuk VPN

Lab ini menjadi dasar Bab 9.

Topologi

```
LAN-A                    WAN Simulasi                LAN-B
192.168.10.0/24          172.16.99.0/24              192.168.20.0/24

VPC-A ------------ R1 ------------------- R2 ------- VPC-B
```

Tabel IP

Perangkat	Interface	IP
R1	LAN-A	`192.168.10.1/24`
R1	WAN simulasi	`172.16.99.1/24`
R2	WAN simulasi	`172.16.99.2/24`
R2	LAN-B	`192.168.20.1/24`
VPC-A	eth0	`192.168.10.10/24`
VPC-B	eth0	`192.168.20.10/24`

VPC-A

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> ping 192.168.10.1
```

VPC-B

```
PC2> ip 192.168.20.10/24 192.168.20.1
PC2> ping 192.168.20.1
```

Target awal:

- VPC-A bisa ping R1.
- VPC-B bisa ping R2.
- R1 dan R2 bisa saling ping di jaringan `172.16.99.0/24`.

VPN akan dikonfigurasi pada bab khusus.

2.20 Desain Lab 6: 1 CHR + VPC IPv4 + VPC IPv6

Lab ini menjadi dasar Bab 10.

Topologi

```
VPC1 IPv4/IPv6 ---- MikroTik CHR R1
```

Tabel IP

Perangkat	IPv4	IPv6	Gateway
R1	192.168.10.1/24	2001:db8:10::1/64	-
VPC1	192.168.10.10/24	2001:db8:10::10/64	R1

RouterOS v7/v6

```
/ip address add address=192.168.10.1/24 interface=ether1
/ipv6 address add address=2001:db8:10::1/64 interface=ether1
advertise=yes
```

VPC

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> ip 2001:db8:10::10/64 2001:db8:10::1
PC1> ping 192.168.10.1
PC1> ping 2001:db8:10::1
```

Catatan:

- Prefix 2001:db8::/32 dipakai untuk dokumentasi dan contoh, bukan jaringan produksi.
- Gunakan prefix ini untuk lab agar aman.

2.21 Manajemen Nama

Lab akan cepat kacau jika penamaan tidak disiplin. Gunakan nama yang konsisten.

Nama Project GNS3

```
mikrotik-bab02-lab01-1chr-1vpc
mikrotik-bab02-lab02-1chr-2vpc
mikrotik-bab02-lab03-3chr-routing
mikrotik-bab02-lab04-vpn
mikrotik-bab02-lab05-ipv6
```

Nama Router

```
R1
R2
R3
R-ISP
R-CUST
```

Nama VPC

```
PC1-LAN10
PC2-LAN10
PC-A
PC-B
PC-IPv6
```

Nama VM

```
U-SRV1
U-CLIENT1
CHR-R1
CHR-R2
CHR-R3
```

Komentar Interface RouterOS v7/v6

```
/interface comment ether1 comment="LAN to VPC1"
/interface comment ether2 comment="Link R1-R2"
/interface comment ether3 comment="WAN Simulasi"
```

Cek:

```
/interface print
```

Prinsip:

> **Nama yang rapi mengurangi 50% kebingungan saat troubleshooting.**

2.22 Strategi Dokumentasi Lab

Setiap lab wajib punya dokumentasi. Minimal empat hal:

- **Tabel IP**
- **Tabel interface**
- **Catatan konfigurasi**
- **Backup/export**

Template Tabel IP

Perangkat	Interface	IP/Prefix	Gateway	Keterangan
R1	ether1	`192.168.10.1/24`	-	LAN
PC1	eth0	`192.168.10.10/24`	`192.168.10.1`	VPC
U-SRV1	enp0s3	`192.168.20.10/24`	`192.168.20.1`	Server uji

Template Tabel Interface

Router	Interface	Terhubung ke	Network
R1	ether1	VPC1	`net-lan10`
R1	ether2	R2	`net-r1-r2`
R2	ether1	R1	`net-r1-r2`

Template Catatan Konfigurasi

```
Nama lab        : Bab02-Lab01-1CHR-1VPC
Tujuan          : VPC bisa ping MikroTik
Router          : R1
Client          : PC1
Network         : 192.168.10.0/24
Gateway         : 192.168.10.1
Status          : Berhasil / gagal
Catatan error   : -
```

Export RouterOS v7/v6

```
/export file=R1-bab02-lab01
/system backup save name=R1-bab02-lab01
```

Catatan:

- `export` mudah dibaca dan cocok untuk belajar.
- `backup` cocok untuk restore cepat.
- Simpan file export dengan nama yang jelas.

2.23 Verifikasi Lab

Setiap lab harus diuji. Jangan lanjut jika uji dasar belum berhasil.

RouterOS v7/v6

```
/interface print
/ip address print
/ip route print
/tool ping 192.168.10.10
```

VPC GNS3

```
PC1> show ip
PC1> ping 192.168.10.1
PC1> trace 192.168.10.1
```

Ubuntu VM

```
ip addr
ip route
ping 192.168.10.1
traceroute 192.168.10.1
```

Jika gagal, cek berurutan:

1. Apakah node hidup?
2. Apakah link aktif?
3. Apakah interface benar?
4. Apakah IP satu subnet?
5. Apakah gateway benar?
6. Apakah route ada?
7. Apakah firewall memblokir?
8. Apakah konfigurasi tersimpan?

2.24 Kesalahan Umum dan Solusi

Masalah	Penyebab Umum	Solusi
VPC tidak bisa ping CHR	Link salah/interface salah	Cek kabel/topologi GNS3
CHR tidak punya interface cukup	Adapter belum ditambah	Tambah interface sebelum boot
Ubuntu VM tidak bisa ping router	Beda network adapter	Samakan network
Router bisa ping client, client tidak bisa keluar	NAT belum dibuat	Dibahas Bab 5
VPC tidak dapat DHCP	DHCP belum dibuat	Dibahas Bab 4
Bingung ether1/ether2	Tidak ada komentar interface	Beri komentar interface
Project sulit diulang	Tidak ada IP plan	Buat tabel IP
Lab rusak setelah eksperimen	Tidak ada export	Jalankan `/export file=...`
Bridged mengganggu jaringan nyata	DHCP aktif sembarangan	Gunakan Internal Network untuk lab

2.25 Checklist Akhir Bab

Pembaca dianggap selesai Bab 2 jika mampu:

- Menjelaskan mengapa **satu PC cukup untuk lab jaringan**.
- Menjelaskan fungsi:
  - VirtualBox,
  - GNS3,
  - VPC,
  - MikroTik CHR,
  - Ubuntu Server VM.
- Membedakan mode VirtualBox:
  - NAT,
  - NAT Network,
  - Bridged Adapter,
  - Host-only Adapter,
  - Internal Network.
- Membuat lab:
  - 1 CHR + 1 VPC,
  - 1 CHR + 2 VPC,
  - 1 CHR + 1 Ubuntu Server VM,
  - 3 CHR untuk routing,
  - 2 CHR + 2 VPC untuk VPN,
  - 1 CHR + VPC IPv4 + VPC IPv6.
- Menggunakan VPC:
  - `show ip`
  - `ip dhcp`
  - `ip 192.168.10.10/24 192.168.10.1`
  - `ping`
  - `trace`
- Menggunakan RouterOS v7/v6:
  - `/interface print`
  - `/ip address print`
  - `/ip route print`
  - `/tool ping`
  - `/export file=...`
- Membuat:
  - tabel IP,
  - tabel interface,
  - catatan konfigurasi,
  - backup/export.

Ringkasan Bab

Bab ini membangun **ekosistem lab jaringan** dengan satu PC Ubuntu 26.04. Pembaca belajar memakai **VirtualBox, GNS3, VPC, MikroTik CHR, dan Ubuntu Server VM** secara terarah.

Kunci Bab 2:

> **Sebelum belajar konfigurasi rumit, bangun dulu lab yang rapi.**
> **Lab yang rapi membuat belajar MikroTik lebih cepat, aman, dan mudah diulang.**

Bab 3. Masuk ke RouterOS CLI dan Konfigurasi Awal

3.1 Tujuan Bab

Setelah menyelesaikan bab ini, pembaca mampu:

- Login awal ke **MikroTik CHR**.
- Mengganti password.
- Membuat user admin baru.
- Menonaktifkan service yang tidak dipakai.
- Mengenal struktur menu CLI:
    - `/interface`
    - `/ip address`
    - `/ip route`
    - `/ip dns`
    - `/ip firewall`
    - `/system`
    - `/tool`
    - `/routing`
- Menggunakan perintah dasar:
    - `print`
    - `add`
    - `set`
    - `remove`
    - `find`
    - `comment`
    - `export`
- Memahami beda **backup** dan **export**.
- Mengatur:
    - identity,
    - timezone,
    - DNS,
    - user,
    - SSH.
- Membaca konfigurasi.
- Menghapus konfigurasi lab.
- Memahami perbedaan umum RouterOS v7 dan v6.

3.2 Masalah Nyata: Takut Salah Ketik di Router

Banyak pemula takut memakai CLI karena merasa:

- takut salah ketik,
- takut router rusak,
- takut kehilangan akses,
- tidak tahu cara melihat konfigurasi,
- tidak tahu cara menghapus perintah yang salah.

Di lab virtual, ketakutan ini harus dibuang.

> **Router virtual bisa dibuat ulang.**
> **Konfigurasi bisa di-export.**
> **Kesalahan bisa menjadi bahan belajar.**

Prinsip bab ini:

> **Jangan takut CLI. Takutlah pada konfigurasi yang tidak didokumentasikan.**

3.3 Mengenal RouterOS dan CLI

RouterOS adalah sistem operasi perangkat MikroTik. Dokumentasi resmi RouterOS berlaku untuk versi stabil terbaru dan tersedia dalam format HTML/PDF untuk penggunaan offline. (MikroTik Help)

RouterOS CLI adalah cara langsung untuk mengelola router melalui terminal. Dokumentasi MikroTik menyebut perintah umum yang sering tersedia di banyak level menu, seperti `print`, `set`, `remove`, `add`, `find`, `export`, `enable`, `disable`, dan `comment`. (MikroTik Help)

Mengapa buku ini memakai CLI?

- **Lebih jelas**: konfigurasi terlihat sebagai perintah.
- **Lebih mudah didokumentasikan**: tinggal salin ke catatan lab.
- **Lebih mudah diulang**: perintah bisa dipakai ulang.
- **Lebih cocok untuk belajar konsep**: pembaca tahu menu dan parameter yang dipakai.
- **Lebih kuat untuk troubleshooting**: cepat melihat route, firewall, interface, dan log.

3.4 Login Awal RouterOS CHR

Pada lab VirtualBox atau GNS3, buka console MikroTik CHR.

Biasanya tampilan awal seperti:

```
MikroTik Login:
```

Masuk menggunakan user awal:

```
Login    : admin
Password : kosong / sesuai image yang dipakai
```

Setelah masuk, prompt biasanya berbentuk:

```
[admin@MikroTik] >
```

Segera lakukan konfigurasi dasar:

- ganti identity,
- buat user baru,
- atur password kuat,
- batasi service,
- export konfigurasi awal.

Catatan keamanan: MikroTik menyarankan password kuat, minimal 12 karakter, memakai angka, simbol, huruf besar, dan huruf kecil, serta bukan kata kamus. (MikroTik Help)

3.5 Mengganti Identity Router

Identity adalah nama router. Ini penting saat lab berisi banyak router.

Contoh nama yang rapi:

```
R1-LAB
R2-LAB
R3-LAB
R-ISP
R-CUST
```

RouterOS v7/v6

```
/system identity set name=R1-LAB
```

Cek hasil:

```
/system identity print
```

Prompt akan berubah menjadi:

```
[admin@R1-LAB] >
```

Prinsip:

> **Router tanpa nama yang jelas akan membingungkan saat topologi mulai besar.**

3.6 Mengganti Password dan Membuat User Admin Baru

Jangan biarkan router memakai user awal tanpa password kuat. Untuk lab, buat user admin baru.

RouterOS v7/v6 — Membuat User Baru

```
/user add name=admin-lab group=full
password=GantiPasswordKuat
```

Cek user:

```
/user print
```

Setelah user baru berhasil, login ulang dengan user baru.

RouterOS v7/v6 — Menonaktifkan User Lama

Jika user baru sudah bisa login:

```
/user disable admin
```

Catatan:

- Pastikan user baru sudah berhasil login sebelum menonaktifkan user lama.
- Jangan sampai semua user admin terkunci.
- Untuk lab awal, dokumentasikan username dan password di catatan lokal yang aman.

3.7 Menonaktifkan Service yang Tidak Dipakai

RouterOS memiliki beberapa service manajemen. Untuk lab dasar, aktifkan hanya yang dibutuhkan.

Lihat service:

```
/ip service print
```

Nonaktifkan service yang tidak dipakai:

```
/ip service disable telnet,ftp,www
```

Jika hanya ingin memakai SSH, batasi akses SSH dari jaringan lab:

```
/ip service set ssh address=192.168.10.0/24
```

RouterOS memiliki SSH server bawaan yang mendukung SSH v2 dan secara default mendengarkan pada TCP port 22; service ini dapat diubah atau dinonaktifkan melalui menu Services. (MikroTik Help)

Catatan penting:

> **Service manajemen adalah pintu masuk router.**
> **Semakin sedikit yang dibuka, semakin kecil risiko.**

3.8 Struktur Menu CLI RouterOS

RouterOS CLI berbentuk hirarki. Menu diawali tanda `/`.

Menu	Fungsi
`/interface`	Melihat dan mengatur interface
`/ip address`	Menambah/melihat IP address
`/ip route`	Melihat/mengatur route IPv4
`/ip dns`	Mengatur DNS
`/ip firewall`	Mengatur firewall, NAT, address list
`/system`	Identity, clock, user, backup, reset
`/tool`	Ping, traceroute, torch, sniffer
`/routing`	OSPF, BGP, routing lanjutan

Contoh:

```
/interface print
/ip address print
/ip route print
/ip dns print
/system clock print
```

Cara berpikir:

```
/menu sub-menu perintah parameter
```

Contoh:

```
/ip address add address=192.168.10.1/24 interface=ether1
```

Artinya:

- masuk menu `/ip address`,
- tambah konfigurasi dengan `add`,
- isi parameter `address`,
- tentukan `interface`.

3.9 Perintah Dasar CLI

Dokumentasi MikroTik menyebut perintah umum seperti `add`, `find`, `print`, `remove`, `set`, `export`, `enable`, `disable`, dan `comment` berlaku konsisten di banyak menu. (MikroTik Help)

print — Melihat Konfigurasi

```
/interface print
/ip address print
/ip route print
```

Gunakan `print` sebelum mengubah konfigurasi.

add — Menambahkan Konfigurasi

```
/ip address add address=192.168.10.1/24 interface=ether1
comment="LAN Lab"
```

set — Mengubah Konfigurasi

Contoh mengubah identity:

```
/system identity set name=R1-LAB
```

Contoh mengubah komentar interface:

```
/interface set ether1 comment="LAN ke VPC1"
```

remove — Menghapus Konfigurasi

Lihat dulu item:

```
/ip address print
```

Hapus item nomor tertentu:

```
/ip address remove 0
```

Catatan:

- Nomor item bisa berubah setelah `print`.
- Hati-hati menghapus rule firewall atau IP manajemen.
- Untuk lab, export dulu sebelum menghapus.

find — Mencari Item

`find` berguna untuk memilih item berdasarkan parameter.

Contoh:

```
/ip address print where interface=ether1
```

Contoh menghapus IP pada interface tertentu:

```
/ip address remove [find interface=ether1]
```

Gunakan hati-hati. Perintah ini bisa menghapus lebih dari satu item.

comment — Memberi Catatan

Komentar sangat penting agar lab tidak membingungkan.

```
/interface comment ether1 comment="LAN ke VPC1"
/interface comment ether2 comment="Link R1-R2"
/ip address add address=192.168.10.1/24 interface=ether1
comment="Gateway LAN10"
```

export — Menyimpan Konfigurasi Teks

```
/export file=R1-awal
```

Lihat file:

```
/file print
```

Export sangat berguna untuk belajar karena hasilnya bisa dibaca, dicatat, dan dibandingkan.

3.10 Konfigurasi Awal Router

Berikut konfigurasi awal yang aman untuk lab.

RouterOS v7/v6 — Konfigurasi Identitas, User, Service, Waktu, Export

```
/system identity set name=R1-LAB
/user add name=admin-lab group=full
password=GantiPasswordKuat
/ip service disable telnet,ftp,www
/ip service set ssh address=192.168.10.0/24
/system clock set time-zone-name=Asia/Jakarta
/export file=R1-awal
```

Jika SSH belum akan dipakai karena IP belum dipasang, baris pembatasan SSH dapat dilakukan setelah IP LAN dibuat.

RouterOS v7/v6 — Cek Hasil

```
/system identity print
/user print
/ip service print
/system clock print
/file print
```

3.11 Memberi IP Awal untuk Lab

Agar router bisa di-*ping* dari VPC GNS3, beri IP pada interface yang terhubung ke VPC.

Topologi:

```
VPC1 ---- ether1 MikroTik CHR
```

RouterOS v7/v6

```
/ip address add address=192.168.10.1/24 interface=ether1
comment="Gateway VPC1"
/ip address print
```

VPC GNS3

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> show ip
PC1> ping 192.168.10.1
```

Jika ping berhasil, berarti:

- link GNS3 aktif,
- IP VPC benar,
- IP MikroTik benar,
- interface benar.

3.12 Akses SSH dari Ubuntu Terminal

Setelah IP router siap, akses dari terminal Ubuntu.

Contoh:

```
ssh admin-lab@192.168.10.1
```

Jika gagal, cek:

- IP router sudah benar.
- SSH service aktif.
- User dan password benar.
- VPC/Ubuntu satu jaringan dengan router.
- `/ip service print` menunjukkan SSH aktif.

Di RouterOS:

```
/ip service print
/user print
/ip address print
```

3.13 DNS Dasar

DNS belum membuat Internet berjalan jika NAT dan route belum benar. Namun DNS perlu dikenalkan sejak awal.

RouterOS v7/v6

```
/ip dns set servers=1.1.1.1 allow-remote-requests=yes
/ip dns print
```

Catatan:

- `allow-remote-requests=yes` membuat router dapat menjawab query DNS dari client.
- Gunakan hanya pada jaringan yang dikontrol.
- Jangan membuka DNS resolver ke jaringan luar tanpa firewall.

3.14 Backup vs Export

Banyak pemula mencampuradukkan backup dan export. Keduanya berbeda.

Fitur	Backup	Export
Format	Binary	Teks
Mudah dibaca manusia	Tidak	Ya
Cocok untuk belajar	Kurang	Sangat cocok
Cocok untuk restore router yang sama	Ya	Bisa, tetapi perlu penyesuaian
Berisi detail spesifik perangkat	Ya	Lebih fleksibel
Cocok untuk dokumentasi buku	Tidak	Ya

MikroTik menjelaskan backup RouterOS sebagai file binary untuk mengkloning konfigurasi router dan direkomendasikan dipulihkan pada versi RouterOS yang sama. (MikroTik Help) Sebaliknya, export/import menghasilkan konfigurasi dalam bentuk teks yang dapat dijalankan dari CLI dan dapat dipakai untuk menyalin bagian konfigurasi antar perangkat. (MikroTik Help)

RouterOS v7/v6 — Export

```
/export file=R1-awal
```

RouterOS v7/v6 — Backup

```
/system backup save name=R1-awal
```

Lihat File

```
/file print
```

Prinsip buku ini:

> **Untuk belajar dan dokumentasi, gunakan export.**
> **Untuk pemulihan cepat pada router/VM yang sama, gunakan backup.**

3.15 Cara Membaca Konfigurasi

Gunakan `print` dan `export`.

Melihat Interface

```
/interface print
```

Yang perlu dibaca:

- nama interface,
- status,
- komentar,
- tipe interface.

Melihat IP

```
/ip address print
```

Yang perlu dibaca:

- IP address,
- network,
- interface.

Melihat Route

```
/ip route print
```

Yang perlu dicari:

- connected route,
- default route `0.0.0.0/0`,
- gateway.

Melihat DNS

```
/ip dns print
```

Melihat Firewall

```
/ip firewall filter print
/ip firewall nat print
```

Catatan:

> **Jangan mengubah konfigurasi sebelum membaca konfigurasi yang ada.**

3.16 Cara Menghapus Konfigurasi Lab

Saat lab salah, jangan panik. Ada beberapa pilihan.

Menghapus IP Address Tertentu

```
/ip address print
/ip address remove 0
```

Atau berdasarkan interface:

```
/ip address remove [find interface=ether1]
```

Menghapus Route Tertentu

```
/ip route print
/ip route remove 0
```

Menghapus File Export

```
/file print
/file remove R1-awal.rsc
```

Reset Konfigurasi

Jika lab benar-benar ingin diulang dari nol:

```
/system reset-configuration no-defaults=yes
skip-backup=yes
```

MikroTik mendokumentasikan bahwa reset konfigurasi dapat dilakukan dari CLI dengan perintah `/system reset-configuration`; proses ini mengembalikan konfigurasi dan harus digunakan dengan hati-hati. (MikroTik Help)

Catatan keras:

Jangan reset router produksi tanpa backup.
Di lab boleh, di jaringan nyata jangan sembarangan.

3.17 Perbedaan Umum RouterOS v7 dan v6

Untuk Bab 3, sebagian besar perintah dasar sama.

Topik	RouterOS v7	RouterOS v6
`/interface print`	Sama	Sama
`/ip address add`	Sama	Sama
`/ip route print`	Sama untuk dasar	Sama untuk dasar
`/ip dns set`	Sama untuk dasar	Sama untuk dasar
`/ip firewall`	Mirip untuk dasar	Mirip untuk dasar
`/system identity`	Sama	Sama
`/user`	Mirip	Mirip
`/routing`	Banyak berubah	Struktur lama
OSPF/BGP	Sintaks sangat berbeda	Sintaks lama
WireGuard	Tersedia native	Tidak tersedia native

Bab ini fokus pada perintah dasar yang relatif sama. Perbedaan besar akan dibahas pada bab routing, VPN, dan BGP.

Catatan penting:

- RouterOS v7 adalah fokus utama.
- RouterOS v6 dipakai sebagai pembanding.
- Jangan mencampur sintaks v6 dan v7 pada lab yang sama tanpa alasan jelas.

3.18 Lab 3.1 — Konfigurasi Awal R1 di GNS3

Topologi

```
VPC1 ---- ether1 R1-CHR
```

Tabel IP

Perangkat	Interface	IP	Gateway
R1	ether1	192.168.10.1/24	-
VPC1	eth0	192.168.10.10/24	192.168.10.1

Langkah 1 — Konfigurasi RouterOS v7/v6

```
/system identity set name=R1-LAB
/user add name=admin-lab group=full
password=GantiPasswordKuat
/ip service disable telnet,ftp,www
/system clock set time-zone-name=Asia/Jakarta
/interface comment ether1 comment="LAN ke VPC1"
/ip address add address=192.168.10.1/24 interface=ether1
comment="Gateway LAN10"
/ip dns set servers=1.1.1.1 allow-remote-requests=yes
/export file=R1-lab03-awal
```

Langkah 2 — Konfigurasi VPC

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> show ip
PC1> ping 192.168.10.1
```

Langkah 3 — Verifikasi RouterOS

```
/interface print
/ip address print
/ip route print
/ip dns print
/file print
/tool ping 192.168.10.10
```

Target Sukses

- Prompt router berubah menjadi `R1-LAB`.
- Interface ether1 punya komentar.
- IP `192.168.10.1/24` muncul di ether1.
- VPC bisa ping router.
- Router bisa ping VPC.

- File export muncul di `/file print`.

3.19 Lab 3.2 — Akses SSH dari Ubuntu

Topologi:

```
Ubuntu Host/VM ---- R1-CHR
```

Pastikan Ubuntu berada satu jaringan dengan router.

Dari Ubuntu

```
ssh admin-lab@192.168.10.1
```

Jika berhasil, pembaca akan masuk ke prompt RouterOS melalui terminal.

Jika Gagal

Cek di RouterOS:

```
/ip service print
/user print
/ip address print
```

Cek di Ubuntu:

```
ip addr
ip route
ping 192.168.10.1
```

Kesalahan umum:

Gejala	Kemungkinan
Ping gagal	IP/subnet/link salah
Ping berhasil, SSH gagal	SSH service mati atau user salah
Login ditolak	Password salah
Tidak ada route	Gateway/interface salah

3.20 Checklist Akhir Bab

Pembaca dianggap selesai Bab 3 jika mampu:

- Login ke RouterOS CHR.
- Mengganti identity router.
- Membuat user admin baru.
- Menonaktifkan service yang tidak dipakai.
- Melihat service dengan:

```
/ip service print
```

- Melihat interface dengan:

```
/interface print
```

- Menambahkan IP dengan:

```
/ip address add address=192.168.10.1/24 interface=ether1
```

- Menguji ping dari RouterOS:

```
/tool ping 192.168.10.10
```

- Menggunakan VPC GNS3:

```
show ip
ping 192.168.10.1
```

- Melakukan export:

```
/export file=R1-awal
```

- Membedakan **backup** dan **export**.
- Menghapus konfigurasi lab sederhana.
- Menjelaskan perbedaan umum RouterOS v7 dan v6.

Ringkasan Bab

Bab ini mengenalkan **RouterOS CLI** sebagai alat utama belajar MikroTik. Pembaca belajar login, membuat user, mengamankan service, membaca interface, memberi IP, menguji ping, menyimpan export, dan menghapus konfigurasi lab.

Kunci Bab 3:

> **CLI bukan sekadar cara mengetik perintah. CLI adalah cara memahami router secara jujur: apa yang dikonfigurasi, di mana dipasang, dan bagaimana hasilnya diverifikasi.**

Bab 4. Router Dasar: IP Statik, DHCP Client, DHCP Server, DNS, dan Default Route

4.1 Tujuan Bab

Setelah menyelesaikan bab ini, pembaca mampu:

- Menjelaskan fungsi dasar **router**.
- Membedakan sisi **WAN** dan **LAN**.
- Mengatur **IP statik** pada interface router.
- Mengaktifkan **DHCP Client** untuk menerima IP dinamis dari sisi atas.
- Memahami **default route**.
- Mengatur **DNS resolver/cache** di MikroTik.
- Membuat **DHCP Server** untuk client LAN.
- Membuat **IP Pool**.
- Membaca **DHCP Lease**.
- Menghubungkan **VPC GNS3** ke MikroTik dan mendapat IP otomatis.
- Menghubungkan **Ubuntu Server VM** ke MikroTik dan melakukan *ping*.
- Melakukan verifikasi dengan:
    - `/ip dhcp-client print`
    - `/ip address print`
    - `/ip route print`
    - `/ip dhcp-server lease print`
    - `/tool ping`

4.2 Masalah Nyata: Router Bisa Keluar, Tapi Client Tidak Bisa

Ini kasus yang sangat sering terjadi di lab:

- Router sudah mendapat IP di sisi WAN.
- Router bisa *ping* ke luar.
- Tetapi client di LAN:
    - tidak mendapat IP,
    - atau mendapat IP tetapi tidak bisa *ping* router,
    - atau bisa *ping* router tapi tidak bisa keluar.

Biasanya masalah ada di salah satu hal ini:

- interface salah,
- IP LAN belum dipasang,
- DHCP Server belum aktif,
- network DHCP belum diisi benar,
- DNS belum diatur,
- default route belum ada.

Bab ini mengajarkan cara membuat **router dasar yang lengkap**, bukan sekadar memberi IP ke satu interface.

4.3 Fungsi Router

Router adalah perangkat yang menghubungkan dua atau lebih jaringan berbeda. Dalam lab ini, peran router adalah:

- menerima koneksi dari sisi luar,
- menjadi *gateway* untuk LAN,
- memberikan IP ke client,
- meneruskan paket ke arah yang benar,
- membantu client menerjemahkan nama domain melalui DNS.

Dalam bentuk paling sederhana:

```
Jaringan luar / NAT
        |
      ether1
   MikroTik CHR
      ether2
        |
   VPC / Ubuntu Client
```

Router dasar di bab ini belum membahas firewall kompleks, VLAN, VPN, atau routing dinamis. Fokusnya adalah **fungsi inti router**:

- ada sisi luar,
- ada sisi dalam,
- ada IP,
- ada route,
- ada DHCP,
- ada DNS.

4.4 WAN dan LAN

Agar pembaca tidak bingung, gunakan istilah yang konsisten:

- **WAN** = sisi ke jaringan luar / upstream / Internet simulasi.
- **LAN** = sisi ke jaringan lokal / client / lab internal.

Contoh pembagian interface:

Interface	Fungsi	Contoh
`ether1`	WAN	DHCP Client dari NAT VirtualBox / node NAT GNS3
`ether2`	LAN	Gateway ke VPC / Ubuntu VM

Prinsip sederhana:

- **WAN** biasanya menerima IP dari atas.
- **LAN** biasanya memakai IP statik dari kita.
- **Client** di LAN biasanya menerima IP dari DHCP Server MikroTik.

4.5 IP Statik di Sisi LAN

Sisi LAN biasanya diatur manual agar stabil. Ini penting karena:

- router harus punya alamat tetap sebagai *gateway*,
- client harus tahu ke mana mengarah,
- DHCP Server perlu tahu network yang dilayani.

Contoh IP LAN:

```
192.168.10.1/24\
```

Artinya:

- network LAN: `192.168.10.0/24`
- gateway router: `192.168.10.1`

RouterOS v7/v6

```
/ip address add address=192.168.10.1/24 interface=ether2
comment="Gateway LAN"
```

Cek:

```
/ip address print
```

Yang harus terlihat:

- IP `192.168.10.1/24`
- terpasang di `ether2`

4.6 IP Dinamis dengan DHCP Client

Pada sisi WAN, cara termudah adalah memakai **DHCP Client**. Ini cocok untuk:

- NAT VirtualBox,
- NAT node di GNS3,
- jaringan atas yang memberi IP otomatis.

RouterOS menjelaskan bahwa DHCP Client dapat diaktifkan pada interface *Ethernet-like*, lalu akan menerima **IP address, netmask, default gateway, dan DNS server**. Default route yang diterima akan muncul sebagai entri **dynamic** di routing table. Jika lease hilang, route dinamis itu ikut hilang. (MikroTik Help)

RouterOS v7/v6

```
/ip dhcp-client add interface=ether1 disabled=no
```

Cek hasil:

```
/ip dhcp-client print
/ip address print
/ip route print
```

Yang harus diperhatikan:

- status DHCP Client = **bound**
- interface `ether1` mendapat IP
- muncul default route `0.0.0.0/0`

Contoh hasil yang diharapkan:

```
INTERFACE  STATUS  ADDRESS
ether1     bound   10.0.2.15/24
```

Dan di routing table:

```
0.0.0.0/0  gateway=10.0.2.2  dynamic
```

4.7 Default Route

Default route adalah jalur untuk semua tujuan yang tidak dikenal secara langsung.

Dalam bahasa sederhana:

> Jika router tidak tahu jaringan tujuan secara spesifik, kirim ke **default gateway**.

Di RouterOS, jika DHCP Client menerima *gateway* dari jaringan atas dan `add-default-route=yes` aktif, maka default route akan ditambahkan otomatis. Itu sebabnya DHCP Client sangat praktis untuk lab dasar. (MikroTik Help)

Cek default route:

```
/ip route print
```

Cari baris:

```
0.0.0.0/0
```

Jika tidak ada default route, router biasanya:

- bisa berkomunikasi ke jaringan lokal,
- tetapi tidak bisa keluar ke jaringan luar.

4.8 DNS Resolver dan DNS Cache

Client tidak cukup hanya punya IP dan gateway. Mereka juga perlu **DNS** agar bisa menerjemahkan nama menjadi alamat IP.

RouterOS mendokumentasikan bahwa router dapat menjadi **DNS cache** untuk client. Jika `allow-remote-requests=yes`, router dapat menerima permintaan DNS dari client pada port 53, dan router juga bisa ditentukan sebagai DNS server di pengaturan DHCP Server. (MikroTik Help)

RouterOS v7/v6

```
/ip dns set allow-remote-requests=yes servers=1.1.1.1
```

Cek:

```
/ip dns print
```

Makna konfigurasi:

- `servers=1.1.1.1` → router menggunakan resolver luar.
- `allow-remote-requests=yes` → client LAN boleh bertanya DNS ke router.

Prinsip praktis:

- Client cukup memakai IP router sebagai DNS.
- Router yang meneruskan query DNS ke resolver atas.
- Ini memudahkan pengelolaan lab.

4.9 DHCP Server untuk LAN

DHCP Server dipakai agar client tidak perlu diisi IP satu per satu secara manual.

RouterOS menyediakan komponen DHCP Server yang umumnya terdiri dari:

- **IP address** di interface LAN,
- **IP Pool**,
- **DHCP Server**,
- **DHCP Network**,
- **Lease**.

Urutan berpikirnya:

1. Tentukan IP gateway LAN.
2. Buat pool alamat untuk dibagikan.
3. Aktifkan DHCP Server pada interface LAN.
4. Tentukan network, gateway, dan DNS yang akan diberikan ke client.

4.10 IP Pool

IP Pool adalah rentang alamat IP yang boleh dibagikan ke client.

Contoh:

```
192.168.10.100 - 192.168.10.200
```

Artinya:

- client akan mendapat alamat dalam rentang itu,
- alamat di luar rentang bisa dipakai untuk perangkat statik.

RouterOS v7/v6

```
/ip pool add name=pool-lan
ranges=192.168.10.100-192.168.10.200
```

Cek:

```
/ip pool print
```

Strategi yang baik:

- `192.168.10.1` → router
- `192.168.10.2–99` → perangkat statik / cadangan
- `192.168.10.100–200` → DHCP client
- `192.168.10.201–254` → cadangan lanjutan

4.11 DHCP Server dan DHCP Network

Setelah pool siap, aktifkan DHCP Server di interface LAN.

RouterOS v7/v6

```
/ip dhcp-server add name=dhcp-lan interface=ether2
address-pool=pool-lan disabled=no
```

Lalu tentukan network yang dilayani:

```
/ip dhcp-server network add address=192.168.10.0/24
gateway=192.168.10.1 dns-server=192.168.10.1
```

Makna:

- `address=192.168.10.0/24` → network LAN
- `gateway=192.168.10.1` → *gateway* untuk client
- `dns-server=192.168.10.1` → client menggunakan router sebagai DNS

Cek hasil:

```
/ip dhcp-server print
/ip dhcp-server network print
```

Catatan penting:

> DHCP Server tidak cukup hanya dibuat.
> Ia harus tahu **network mana**, **gateway berapa**, dan **DNS apa** yang akan dibagikan.

4.12 Lease DHCP

Lease adalah daftar client yang telah mendapat IP dari DHCP Server.

Ini penting untuk:

- melihat client mana yang aktif,
- mengecek apakah client benar-benar mendapat IP,
- memverifikasi apakah DHCP berjalan.

Cek lease:

```
/ip dhcp-server lease print
```

Jika client berhasil mendapat IP, akan muncul data seperti:

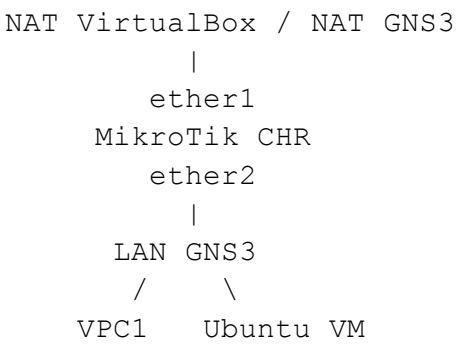
- address,
- MAC address,
- host name,
- status.

Jika daftar lease kosong, kemungkinan:

- client belum meminta IP,
- DHCP Server belum aktif,
- interface salah,
- network salah,
- topologi salah.

4.13 Konfigurasi Router Dasar: Skenario Utama

Topologi utama bab ini:



Tujuan lab:

- `ether1` mendapat IP otomatis dari DHCP Client.
- `ether2` memakai IP statik `192.168.10.1/24`.
- Router menjadi DHCP Server untuk LAN.
- VPC dan Ubuntu VM bisa mendapat IP dari DHCP.
- Client bisa *ping* router.

Konfigurasi RouterOS v7/v6 Lengkap

```
/ip dhcp-client add interface=ether1 disabled=no
/ip address add address=192.168.10.1/24 interface=ether2
/ip pool add name=pool-lan
ranges=192.168.10.100-192.168.10.200
/ip dhcp-server add name=dhcp-lan interface=ether2
address-pool=pool-lan disabled=no
/ip dhcp-server network add address=192.168.10.0/24
gateway=192.168.10.1 dns-server=192.168.10.1
/ip dns set allow-remote-requests=yes servers=1.1.1.1
```

Setelah itu, lakukan verifikasi.

4.14 Verifikasi Router Dasar

Gunakan lima perintah utama berikut.

1. Cek DHCP Client

```
/ip dhcp-client print
```

Yang dicari:

- interface = `ether1`
- status = **bound**

2. Cek IP Address

```
/ip address print
```

Yang dicari:

- IP WAN di `ether1`
- IP LAN `192.168.10.1/24` di `ether2`

3. Cek Route

```
/ip route print
```

Yang dicari:

- connected route untuk `192.168.10.0/24`
- default route `0.0.0.0/0`

4. Cek DHCP Lease

```
/ip dhcp-server lease print
```

Yang dicari:

- apakah VPC / Ubuntu muncul sebagai lease

5. Uji Ping

```
/tool ping 192.168.10.100
/tool ping 1.1.1.1
```

Yang diuji:

- ping ke client → LAN bekerja
- ping ke luar → jalur keluar bekerja

4.15 VPC GNS3 sebagai Client DHCP

VPCS mendukung DHCP dan ping, sehingga sangat cocok untuk menguji apakah DHCP Server MikroTik sudah berjalan. (GNS3 Documentation)

Di VPC GNS3

```
PC1> ip dhcp
PC1> show ip
PC1> ping 192.168.10.1
```

Jika berhasil, `show ip` akan menampilkan:

- IP dari rentang pool
- subnet mask
- gateway
- DNS bila tersedia

Makna hasil:

- `ip dhcp` berhasil → DHCP Server bekerja
- `ping 192.168.10.1` berhasil → router bisa dijangkau

Jika gagal:

- cek apakah VPC terhubung ke `ether2`,
- cek apakah DHCP Server aktif,
- cek apakah IP LAN ada di `ether2`.

4.16 Ubuntu Server VM sebagai Client

Ubuntu VM dipakai untuk uji yang lebih lengkap dibanding VPC.

Pilihan 1 — Ubuntu sebagai DHCP Client

Jika Ubuntu terhubung ke LAN yang sama, biarkan ia mendapat IP otomatis dari DHCP.

Setelah boot, cek:

```
ip addr
ip route
ping 192.168.10.1
```

Yang dicari:

- IP dari rentang `192.168.10.100–200`
- default route via `192.168.10.1`

Pilihan 2 — Ubuntu sebagai Client Statik

Jika ingin uji manual:

```
sudo ip addr add 192.168.10.20/24 dev enp0s3
sudo ip route add default via 192.168.10.1
ping 192.168.10.1
```

Bab ini tetap menyarankan DHCP untuk lab awal karena lebih cepat dan realistis.

4.17 Cara Membaca Alur DHCP

Saat client meminta IP, urutannya secara sederhana:

1. client mengirim permintaan DHCP,
2. router menawarkan IP dari pool,
3. client meminta IP tersebut,
4. router mengesahkan lease.

Untuk pembaca pemula, cukup pahami ini:

- **Pool** = sumber alamat
- **DHCP Server** = pembagi alamat
- **Lease** = bukti client sudah mendapat alamat
- **DHCP Network** = informasi network, gateway, DNS yang dibagikan

Gambar mentalnya:

```
Client minta IP → MikroTik pilih dari pool → MikroTik kirim
gateway + DNS → client siap dipakai
```

4.18 Lab 4.1 — CHR Tersambung ke GNS3, VPC Mendapat IP

Topologi

```
NAT GNS3 / VirtualBox NAT
          |
       ether1
   MikroTik CHR
       ether2
          |
        VPC1
```

Tabel IP

Perangkat	Interface	IP
R1 CHR	ether1	DHCP
R1 CHR	ether2	192.168.10.1/24
VPC1	eth0	DHCP

Langkah RouterOS v7/v6

```
/system identity set name=R1-BAB4
/ip dhcp-client add interface=ether1 disabled=no
/ip address add address=192.168.10.1/24 interface=ether2
comment="LAN"
/ip pool add name=pool-lan
ranges=192.168.10.100-192.168.10.200
/ip dhcp-server add name=dhcp-lan interface=ether2
address-pool=pool-lan disabled=no
/ip dhcp-server network add address=192.168.10.0/24
gateway=192.168.10.1 dns-server=192.168.10.1
/ip dns set allow-remote-requests=yes servers=1.1.1.1
/export file=R1-bab4-lab1
```

Langkah VPC

```
PC1> ip dhcp
PC1> show ip
PC1> ping 192.168.10.1
```

Target sukses

- VPC mendapat IP otomatis
- VPC bisa *ping* 192.168.10.1
- Lease muncul di router

4.19 Lab 4.2 — Ubuntu VM sebagai Client Uji

Topologi

```
NAT
 |
ether1
MikroTik CHR
ether2
 |
Ubuntu Server VM
```

Tabel IP

Perangkat	Interface	IP
R1 CHR	ether1	DHCP
R1 CHR	ether2	`192.168.10.1/24`
Ubuntu VM	enp0s3	DHCP atau statik

Verifikasi di Ubuntu

```
ip addr
ip route
ping 192.168.10.1
```

Verifikasi di RouterOS

```
/ip dhcp-server lease print
/tool ping 192.168.10.101
```

Target sukses

- Ubuntu punya IP di subnet `192.168.10.0/24`
- Ubuntu bisa *ping* router
- Router bisa *ping* Ubuntu

4.20 Kesalahan Umum dan Solusi

Masalah	Penyebab Umum	Solusi
DHCP Client tidak *bound*	`ether1` tidak terhubung ke upstream	Cek topologi WAN
VPC tidak mendapat IP	DHCP Server belum aktif / interface salah	Cek `/ip dhcp-server print`
Lease kosong	client belum minta IP	Jalankan `ip dhcp` di VPC
Client dapat IP tapi tidak bisa *ping* router	IP LAN salah / VPC salah link	Cek `ether2` dan topologi
Router bisa *ping* luar, client tidak	NAT belum dibuat	Akan dibahas di Bab 5
DNS gagal	`/ip dns` belum diatur	Cek `/ip dns print`
Ubuntu tidak dapat IP	adapter salah / DHCP client sistem tidak aktif	Cek `ip addr` dan network VM

Catatan penting:

> **Bab 4 belum menyelesaikan akses Internet penuh untuk client jika NAT belum dibuat.**
> Bab ini fokus pada **router dasar, DHCP, DNS, dan route dasar**.
> Akses keluar penuh untuk client akan diperkokoh di **Bab 5 tentang NAT**.

4.21 Checklist Akhir Bab

Pembaca dianggap selesai Bab 4 jika mampu:

- Menjelaskan fungsi dasar router.
- Membedakan **WAN** dan **LAN**.
- Memberi IP statik pada LAN.
- Mengaktifkan DHCP Client di WAN.
- Membaca default route.
- Mengatur DNS di MikroTik.
- Membuat IP Pool.
- Membuat DHCP Server.
- Membaca DHCP Lease.
- Menguji VPC dengan:

```
ip dhcp
show ip
ping 192.168.10.1
```

- Menguji Ubuntu VM dengan:

```
ip addr
ip route
ping 192.168.10.1
```

- Melakukan verifikasi RouterOS dengan:

```
/ip dhcp-client print
/ip address print
/ip route print
/ip dhcp-server lease print
/tool ping
```

Ringkasan Bab

Bab ini membentuk **router dasar yang utuh**:

- sisi WAN memakai **DHCP Client**,
- sisi LAN memakai **IP statik**,
- router membagikan IP lewat **DHCP Server**,
- client memakai router sebagai **gateway** dan **DNS**,
- VPC dan Ubuntu VM dipakai untuk pengujian.

Kunci Bab 4:

> **Router dasar yang baik bukan yang paling rumit, tetapi yang jelas:**
> **ada WAN, ada LAN, ada IP, ada route, ada DHCP, ada DNS, dan semua bisa diverifikasi.**

Bab 5. NAT dan Masquerade: Menghubungkan VPC dan Ubuntu Client ke Internet

5.1 Tujuan Bab

Setelah menyelesaikan bab ini, pembaca mampu:

- Menjelaskan **mengapa NAT dibutuhkan**.
- Membedakan **routing** dan **NAT**.
- Memahami **Source NAT**.
- Memahami **masquerade**.
- Membuat rule NAT dasar di RouterOS v7/v6.
- Menghubungkan **VPC GNS3** ke jaringan luar melalui MikroTik.
- Menghubungkan **Ubuntu Client/VM** ke jaringan luar melalui MikroTik.
- Melakukan verifikasi:
    - ping gateway,
    - ping IP luar,
    - traceroute,
    - cek NAT rule,
    - cek connection tracking.
- Melakukan troubleshooting jika NAT gagal.

5.2 Masalah Nyata: Client Bisa Ping Router, Tapi Tidak Bisa Keluar

Topologi:

```
NAT GNS3 / VirtualBox NAT
          |
        ether1
     MikroTik CHR
        ether2
          |
      VPC / Ubuntu Client
```

Client LAN mendapat IP:

```
IP      : 192.168.10.100
Gateway : 192.168.10.1
DNS     : 192.168.10.1
```

Client bisa ping router:

```
PC1> ping 192.168.10.1
```

Tetapi gagal ping IP luar:

```
PC1> ping 1.1.1.1
```

Kemungkinan masalah:

- NAT belum dibuat.
- Rule NAT salah interface.
- Router belum punya default route.
- DNS salah.
- Firewall memblokir *forward*.
- Client tidak punya gateway.

Bab ini fokus pada penyebab pertama: **NAT belum dibuat atau salah dibuat**.

5.3 Mengapa NAT Dibutuhkan

Alamat LAN biasanya memakai **private IP**, misalnya:

```
192.168.10.0/24
10.10.10.0/24
172.16.1.0/24
```

Alamat seperti ini aman untuk lab dan jaringan internal, tetapi tidak dirutekan langsung di Internet publik.

Masalahnya:

- VPC memakai IP private.
- Ubuntu Client memakai IP private.
- Server luar tidak tahu cara membalas langsung ke IP private itu.
- Router harus mengganti alamat sumber sebelum paket keluar.

Itulah fungsi **NAT**.

MikroTik menjelaskan pada dokumentasi konfigurasi awal bahwa alamat lokal tidak dapat dirutekan di Internet; solusinya adalah mengganti source address paket keluar menjadi IP router pada sisi publik/keluar, dan contoh rule yang diberikan adalah `chain=srcnat out-interface=ether1 action=masquerade`. (MikroTik Help)

5.4 NAT vs Routing

Ini bagian yang sering membingungkan pemula.

Routing menjawab:

> Paket ini harus lewat jalur mana?

NAT menjawab:

> Alamat sumber/tujuan paket perlu diubah atau tidak?

Tabel sederhana:

Konsep	Fungsi	Contoh
Routing	Menentukan jalur paket	`0.0.0.0/0 via ether1`
NAT	Mengubah alamat paket	`192.168.10.100 → IP ether1`
Masquerade	Source NAT otomatis berdasarkan IP interface keluar	cocok untuk WAN dinamis
Firewall filter	Mengizinkan atau menolak paket	allow/drop

Contoh:

- Router punya route keluar, tetapi NAT belum ada → router bisa keluar, client tidak.
- NAT ada, tetapi route keluar tidak ada → paket tetap tidak tahu jalan.
- NAT dan route benar, tetapi firewall memblokir → client tetap gagal.

Kunci:

> **NAT bukan pengganti routing. Routing menentukan jalan. NAT mengubah alamat.**

5.5 Source NAT

Source NAT mengubah alamat sumber paket.

Sebelum NAT:

```
Source      : 192.168.10.100
Destination : 1.1.1.1
```

Sesudah NAT:

```
Source      : IP ether1 MikroTik
Destination : 1.1.1.1
```

Balasan dari luar akan kembali ke IP MikroTik. Router kemudian mencocokkan koneksi dan mengirim balasan ke client LAN yang benar.

Alur sederhananya:

```
Client LAN → MikroTik ubah source IP → jaringan luar
jaringan luar → MikroTik cocokkan koneksi → Client LAN
```

5.6 Masquerade

Masquerade adalah bentuk Source NAT yang praktis untuk interface WAN dengan IP dinamis.

Cocok untuk:

- lab GNS3,
- VirtualBox NAT,
- koneksi WAN yang IP-nya berubah,
- DHCP Client pada `ether1`.

MikroTik menjelaskan bahwa *masquerade* memilih alamat sumber dari IP yang terpasang pada interface tempat paket keluar. Dokumentasi firewall lanjutan MikroTik juga menyebut bahwa untuk kasus WAN dinamis, *masquerade* menjadi pilihan yang sesuai. (MikroTik Help)

Contoh rule:

```
/ip firewall nat add chain=srcnat out-interface=ether1
action=masquerade comment="NAT LAN to WAN"
```

Makna:

- `chain=srcnat` → ubah alamat sumber.
- `out-interface=ether1` → hanya untuk paket yang keluar lewat ether1.
- `action=masquerade` → gunakan IP interface keluar sebagai alamat sumber.
- `comment` → dokumentasi agar mudah dibaca.

5.7 NAT untuk Lab GNS3

Dalam GNS3, lab umum:

```
NAT Node / jaringan luar
          |
        ether1
        CHR-R1
        ether2
          |
        VPC1
```

VPC GNS3 sangat cocok untuk uji cepat karena VPCS mendukung **DHCP** dan **ping**, serta hanya memakai sekitar 2 MB RAM per instance. (GNS3 Documentation)

Tabel IP

Perangkat	Interface	IP	Keterangan
R1	ether1	DHCP	WAN simulasi
R1	ether2	`192.168.10.1/24`	LAN
VPC1	eth0	DHCP	Client

Konsep

- VPC mendapat IP dari DHCP MikroTik.
- MikroTik mendapat IP WAN dari NAT/upstream.
- NAT masquerade membuat VPC bisa keluar.

5.8 NAT untuk Ubuntu VM

Ubuntu VM dipakai untuk uji lebih realistis:

- `ip route`
- `ping`
- `traceroute`
- `curl`
- `dig`
- `tcpdump`

Topologi:

```
NAT / jaringan luar
        |
      ether1
   MikroTik CHR
      ether2
        |
   Ubuntu Client
```

Ubuntu bisa memakai DHCP dari MikroTik atau IP statik.

Contoh IP statik:

```
sudo ip addr add 192.168.10.20/24 dev enp0s3
sudo ip route add default via 192.168.10.1
```

Uji:

```
ping 192.168.10.1
ping 1.1.1.1
traceroute 1.1.1.1
```

5.9 Konfigurasi Dasar Sebelum NAT

Sebelum NAT dibuat, pastikan Bab 4 sudah berjalan.

RouterOS v7/v6 — WAN DHCP Client

```
/ip dhcp-client add interface=ether1 disabled=no
```

RouterOS v7/v6 — LAN IP

```
/ip address add address=192.168.10.1/24 interface=ether2
comment="LAN"
```

RouterOS v7/v6 — DHCP Server LAN

```
/ip pool add name=pool-lan
ranges=192.168.10.100-192.168.10.200
/ip dhcp-server add name=dhcp-lan interface=ether2
address-pool=pool-lan disabled=no
/ip dhcp-server network add address=192.168.10.0/24
gateway=192.168.10.1 dns-server=192.168.10.1
/ip dns set allow-remote-requests=yes servers=1.1.1.1
```

Cek route

```
/ip route print
```

Pastikan ada:

```
0.0.0.0/0
```

Jika default route belum ada, NAT tidak akan menyelesaikan masalah.

5.10 Konfigurasi NAT Masquerade

RouterOS v7/v6

```
/ip firewall nat add chain=srcnat out-interface=ether1
action=masquerade comment="NAT LAN to WAN"
```

Cek rule:

```
/ip firewall nat print
```

Contoh hasil:

```
chain=srcnat action=masquerade out-interface=ether1
comment="NAT LAN to WAN"
```

Catatan:

Rule NAT harus mengarah ke interface keluar.
Jika interface WAN adalah `ether1`, pakai `out-interface=ether1`.
Jika WAN berbeda, sesuaikan.

5.11 Verifikasi RouterOS v7/v6

Gunakan perintah berikut.

Cek NAT Rule

```
/ip firewall nat print
```

Pastikan rule tidak salah interface.

Cek Connection Tracking

```
/ip firewall connection print
```

Jika client sedang mengakses jaringan luar, akan muncul koneksi aktif.
Connection tracking penting karena NAT perlu mengingat koneksi agar balasan dari luar bisa dikembalikan ke client yang benar.

Ping dari Router

```
/tool ping 1.1.1.1
```

Jika router sendiri tidak bisa ping luar, cek:

- DHCP Client WAN,
- default route,
- koneksi NAT/Internet simulasi,
- DNS tidak relevan untuk ping IP.

Traceroute dari Router

```
/tool traceroute 1.1.1.1
```

Traceroute membantu melihat jalur keluar dari router.

5.12 Verifikasi Koneksi VPC GNS3

Langkah 1 — Ambil IP DHCP

```
PC1> ip dhcp
```

Langkah 2 — Cek IP

```
PC1> show ip
```

Pastikan VPC mendapat:

- IP `192.168.10.x`
- gateway `192.168.10.1`

Langkah 3 — Ping Gateway

```
PC1> ping 192.168.10.1
```

Jika gagal:

- VPC belum satu link dengan MikroTik.
- Interface LAN salah.
- IP gateway belum dipasang.
- DHCP belum benar.

Langkah 4 — Ping IP Luar

```
PC1> ping 1.1.1.1
```

Jika gateway bisa diping tetapi IP luar gagal:

- cek NAT,
- cek default route,
- cek firewall forward.

Langkah 5 — Trace

```
PC1> trace 1.1.1.1
```

Jika trace berhenti di `192.168.10.1`, masalah ada di router atau jalur keluar.

5.13 Verifikasi Koneksi Ubuntu VM

Cek IP

```
ip addr
```

Pastikan Ubuntu mendapat IP LAN.

Cek Route

```
ip route
```

Harus ada:

```
default via 192.168.10.1
```

Ping Gateway

```
ping 192.168.10.1
```

Ping IP Luar

```
ping 1.1.1.1
```

Traceroute

```
traceroute 1.1.1.1
```

Jika traceroute belum tersedia:

```
sudo apt install -y traceroute
```

5.14 Lab 5.1 — VPC Keluar Melalui NAT MikroTik

Topologi

```
NAT GNS3 / VirtualBox NAT
          |
        ether1
        CHR-R1
        ether2
          |
        VPC1
```

Tabel IP

Perangkat	Interface	IP
R1	ether1	DHCP
R1	ether2	192.168.10.1/24
VPC1	eth0	DHCP

RouterOS v7/v6

```
/system identity set name=R1-BAB5
/ip dhcp-client add interface=ether1 disabled=no
/ip address add address=192.168.10.1/24 interface=ether2 comment="LAN"
/ip pool add name=pool-lan ranges=192.168.10.100-192.168.10.200
/ip dhcp-server add name=dhcp-lan interface=ether2
address-pool=pool-lan disabled=no
/ip dhcp-server network add address=192.168.10.0/24
gateway=192.168.10.1 dns-server=192.168.10.1
/ip dns set allow-remote-requests=yes servers=1.1.1.1
/ip firewall nat add chain=srcnat out-interface=ether1
action=masquerade comment="NAT LAN to WAN"
/export file=R1-bab5-nat
```

VPC GNS3

```
PC1> ip dhcp
PC1> show ip
PC1> ping 192.168.10.1
PC1> ping 1.1.1.1
PC1> trace 1.1.1.1
```

Target Sukses

- VPC mendapat IP dari DHCP.
- VPC bisa ping gateway.
- VPC bisa ping IP luar.
- NAT rule terlihat di RouterOS.
- Connection tracking menunjukkan koneksi aktif.

5.15 Lab 5.2 — Ubuntu Client Keluar Melalui NAT MikroTik

Topologi

```
NAT / jaringan luar
        |
      ether1
   MikroTik CHR
      ether2
        |
   Ubuntu Client VM
```

RouterOS v7/v6

Gunakan konfigurasi sama dengan Lab 5.1.

Ubuntu DHCP Client

```
ip addr
ip route
ping 192.168.10.1
ping 1.1.1.1
traceroute 1.1.1.1
```

Ubuntu IP Statik Jika Dibutuhkan

```
sudo ip addr add 192.168.10.20/24 dev enp0s3
sudo ip route add default via 192.168.10.1
```

Lalu uji:

```
ping 192.168.10.1
ping 1.1.1.1
traceroute 1.1.1.1
```

Target Sukses

- Ubuntu punya IP LAN.
- Ubuntu punya default route ke MikroTik.
- Ubuntu bisa ping gateway.
- Ubuntu bisa ping IP luar.

5.16 Membaca NAT dari Connection Tracking

Setelah VPC atau Ubuntu melakukan ping ke luar, cek router:

```
/ip firewall connection print
```

Yang dicari:

- koneksi dari `192.168.10.x`,
- tujuan luar,
- status koneksi.

Ini membantu pembaca memahami bahwa NAT bukan “sihir”, tetapi proses yang dilacak router.

Catatan:

> **Jika tidak ada connection entry, kemungkinan paket client belum sampai ke router atau firewall memblokir.**

5.17 Troubleshooting NAT

1. Salah out-interface

Gejala:

- NAT rule ada.
- Client tetap tidak bisa keluar.

Cek:

```
/ip firewall nat print
/interface print
```

Pastikan `out-interface` adalah interface ke jaringan luar.

Contoh salah:

```
out-interface=ether2
```

Padahal ether2 adalah LAN.

Contoh benar:

```
out-interface=ether1
```

2. Tidak Ada Default Route

Gejala:

- Router tidak bisa ping `1.1.1.1`.
- Client juga gagal.

Cek:

```
/ip route print
```

Cari:

```
0.0.0.0/0
```

Jika tidak ada, cek DHCP Client:

```
/ip dhcp-client print
```

3. DNS Salah

Gejala:

- Ping `1.1.1.1` berhasil.
- Akses nama domain gagal.

Cek RouterOS:

```
/ip dns print
```

Cek Ubuntu:

```
ip route
dig example.net
```

Catatan:

- Masalah DNS berbeda dengan masalah NAT.
- Uji NAT pertama sebaiknya pakai IP, bukan nama domain.

4. Firewall Memblokir Forward

Gejala:

- NAT benar.
- Route benar.
- Client tetap gagal keluar.

Cek firewall:

```
/ip firewall filter print
```

Jika ada rule drop pada chain `forward`, pastikan ada rule accept untuk LAN ke WAN.

Bab firewall penuh dibahas di Bab 6. Untuk lab awal, jangan membuat rule drop sebelum paham alur paket.

5. Client Tidak Punya Gateway

Gejala:

- Client punya IP.
- Client tidak bisa ping luar.
- Client mungkin bisa ping host satu subnet.

Cek VPC:

```
PC1> show ip
```

Cek Ubuntu:

```
ip route
```

Pastikan gateway:

```
192.168.10.1
```

5.18 Checklist Akhir Bab

Pembaca dianggap selesai Bab 5 jika mampu:

- Menjelaskan mengapa **NAT dibutuhkan**.
- Menjelaskan beda **routing** dan **NAT**.
- Menjelaskan fungsi **Source NAT**.
- Menjelaskan fungsi **masquerade**.
- Membuat rule NAT:

```
/ip firewall nat add chain=srcnat out-interface=ether1
action=masquerade comment="NAT LAN to WAN"
```

- Memverifikasi RouterOS dengan:

```
/ip firewall nat print
/ip firewall connection print
/tool ping 1.1.1.1
/tool traceroute 1.1.1.1
```

- Menguji VPC:

```
PC1> ping 192.168.10.1
PC1> ping 1.1.1.1
PC1> trace 1.1.1.1
```

- Menguji Ubuntu:

```
ip route
ping 192.168.10.1
ping 1.1.1.1
traceroute 1.1.1.1
```

- Menemukan penyebab NAT gagal:
    - salah `out-interface`,
    - tidak ada default route,
    - DNS salah,
    - firewall memblokir,
    - client tidak punya gateway.

Ringkasan Bab

Bab ini mengajarkan **NAT dan masquerade** sebagai jembatan antara jaringan private di lab dan jaringan luar. VPC dan Ubuntu Client tidak cukup hanya mendapat IP dan gateway. Mereka membutuhkan router yang memiliki **default route** dan **NAT rule** yang benar.

Kunci Bab 5:

> **Jika router bisa keluar tetapi client tidak bisa, cek NAT.**
> **Jika NAT sudah ada tetapi tetap gagal, cek route, interface, gateway, DNS, dan firewall.**

Bab 6. Firewall Dasar, NAT Lanjutan, dan L7 Firewall

6.1 Tujuan Bab

Setelah menyelesaikan bab ini, pembaca mampu:

- Menjelaskan mengapa **firewall wajib**.
- Membedakan chain:
  - `input`
  - `forward`
  - `output`
- Memahami *connection state*:
  - `established`
  - `related`
  - `invalid`
  - `new`
- Membuat firewall dasar untuk melindungi router.
- Membuat firewall dasar untuk melindungi LAN.
- Menggunakan **address list**.
- Mengaktifkan **logging dasar**.
- Membuat NAT lanjutan:
  - `dst-nat`
  - *port forwarding*
  - pengantar *hairpin NAT*
- Memahami konsep **Layer 7 matcher**.
- Mengetahui batasan L7 dan dampaknya ke CPU.
- Menguji firewall dari **VPC GNS3** dan **Ubuntu Server VM**.

6.2 Mengapa Firewall Wajib

Router adalah pintu lalu lintas. Tanpa firewall, router bisa menerima akses yang tidak perlu dari berbagai arah.

Firewall digunakan untuk:

- melindungi router dari akses tidak sah,
- membatasi akses manajemen,
- mengendalikan trafik dari LAN ke luar,
- mengatur trafik antarsegmen,
- membuat NAT lanjutan seperti *port forwarding*,
- membaca gejala serangan atau salah konfigurasi melalui log.

RouterOS firewall resmi dipakai untuk mengizinkan atau memblokir paket yang **diteruskan ke jaringan lokal**, **berasal dari router**, atau **ditujukan ke router**. Ada dua pendekatan umum: hanya izinkan trafik tertentu lalu drop sisanya, atau drop hanya trafik berbahaya dan izinkan sisanya. (MikroTik Help)

Prinsip buku ini memakai pendekatan aman untuk pemula:

> **Izinkan yang jelas dibutuhkan, lalu drop sisanya.**

6.3 Chain: Input, Forward, Output

Topologi:

```
Ping ke router:
VPC → MikroTik
      chain=input

Akses keluar:
VPC → MikroTik → jaringan luar
      chain=forward
```

Firewall RouterOS memakai konsep **chain**. Chain menentukan posisi paket.

Chain	Artinya	Contoh
input	paket menuju router	SSH ke MikroTik, ping ke router
forward	paket melewati router	VPC ke Internet, Ubuntu ke server luar
output	paket keluar dari router sendiri	router ping ke luar, router DNS query

Paling penting untuk pemula:

- **input** melindungi router.
- **forward** melindungi client dan jaringan yang melewati router.
- **output** jarang diubah di lab dasar.

Contoh sederhana:

```
Client ping router      → chain input
Client akses Internet   → chain forward
Router ping 1.1.1.1     → chain output
```

MikroTik menjelaskan bahwa firewall mengelola aliran data **to, from, and through the router** serta bekerja bersama NAT untuk mencegah akses tidak sah ke jaringan dan router. (MikroTik Help)

6.4 Connection State

Firewall modern tidak hanya melihat paket satu per satu. RouterOS dapat memakai **connection tracking**, sehingga router tahu status koneksi.

State	Arti Praktis
established	koneksi yang sudah berjalan
related	koneksi terkait koneksi lain yang sah
invalid	paket rusak/tidak cocok dengan koneksi valid
new	koneksi baru

Prinsip umum firewall dasar:

- Terima `established,related`.
- Drop `invalid`.
- Izinkan akses admin dari LAN.
- Izinkan ICMP seperlunya.
- Drop sisanya.

Connection tracking di RouterOS dapat dilihat melalui `/ip firewall connection` untuk IPv4 dan `/ipv6 firewall connection` untuk IPv6. Ini penting karena NAT dan firewall stateful membutuhkan pencatatan koneksi. (MikroTik Help)

6.5 Firewall untuk Melindungi Router

Router harus dilindungi dari akses yang tidak perlu, terutama dari sisi WAN.

Yang perlu dilindungi:

- SSH router.
- DNS router.
- service manajemen.
- ping dari sumber yang tidak dikenal.
- percobaan koneksi dari WAN.

Topologi dasar:

```
WAN / NAT GNS3
      |
   ether1
MikroTik CHR
   ether2
     |
    LAN
```

Asumsi:

- `ether1` = WAN.
- `ether2` = LAN/admin.
- Admin hanya boleh dari LAN.

RouterOS v7/v6 — Firewall Input Dasar

```
/ip firewall filter add chain=input connection-state=established,related
action=accept comment="Allow established related"
/ip firewall filter add chain=input connection-state=invalid action=drop
comment="Drop invalid"
/ip firewall filter add chain=input in-interface=ether2 action=accept
comment="Allow LAN admin"
/ip firewall filter add chain=input protocol=icmp action=accept
comment="Allow ping"
/ip firewall filter add chain=input action=drop comment="Drop other input"
```

Makna:

- koneksi yang sudah sah tetap diterima,
- paket invalid dibuang,
- admin dari LAN diizinkan,
- ping diizinkan untuk lab,
- sisanya ditolak.

Catatan kritis:

> **Urutan rule sangat penting.**
> Rule drop di atas rule allow bisa membuat router tidak bisa diakses.

6.6 Firewall untuk Melindungi LAN

Jika chain `input` melindungi router, chain `forward` melindungi trafik yang **melewati router**.

Contoh:

```
VPC / Ubuntu Client → MikroTik → jaringan luar
```

Trafik ini masuk chain `forward`, bukan `input`.

RouterOS v7/v6 — Firewall Forward Dasar

```
/ip firewall filter add chain=forward
connection-state=established,related action=accept
comment="Allow established related forward"
/ip firewall filter add chain=forward
connection-state=invalid action=drop comment="Drop invalid
forward"
/ip firewall filter add chain=forward in-interface=ether2
out-interface=ether1 action=accept comment="Allow LAN to WAN"
/ip firewall filter add chain=forward action=drop
comment="Drop other forward"
```

Makna:

- koneksi LAN ke WAN diizinkan,
- balasan dari koneksi sah diizinkan,
- paket invalid dibuang,
- selain itu ditolak.

Jika rule ini dipakai, pastikan NAT Bab 5 sudah benar.

6.7 Address List

Address list adalah daftar IP yang bisa dipakai berulang di firewall.

Contoh pemakaian:

- daftar IP admin,
- daftar jaringan LAN,
- daftar IP yang diblokir,
- daftar IP yang diizinkan remote.

RouterOS v7/v6 — Address List Admin

```
/ip firewall address-list add list=admin-lan
address=192.168.10.0/24 comment="Admin LAN"
```

Gunakan dalam firewall:

```
/ip firewall filter add chain=input
src-address-list=admin-lan in-interface=ether2 action=accept
comment="Allow admin LAN"
```

Kelebihan:

- rule lebih rapi,
- daftar IP mudah diubah,
- firewall lebih mudah dibaca.

Contoh prinsip:

> Daripada menulis banyak IP di banyak rule, masukkan IP ke **address list**, lalu panggil daftar itu di rule firewall.

6.8 Logging Dasar

Logging membantu pembaca melihat apa yang diblokir atau diterima firewall. Tetapi jangan logging semua trafik karena bisa membuat log penuh.

RouterOS v7/v6 — Log Paket Drop Input

```
/ip firewall filter add chain=input action=drop log=yes
log-prefix="DROP-INPUT " comment="Drop other input with log"
```

Untuk melihat log:

```
/log print
```

Untuk lab, logging berguna saat:

- ping gagal,
- SSH gagal,
- port forwarding tidak masuk,
- firewall terlalu ketat.

Catatan tajam:

> **Log adalah alat diagnosis, bukan hiasan.**
> Aktifkan secukupnya, baca hasilnya, lalu rapikan.

6.9 NAT Lanjutan: Dst-NAT dan Port Forwarding

Bab 5 membahas *source NAT* dan *masquerade*. Bab ini memperkenalkan **destination NAT** atau `dst-nat`.

MikroTik menjelaskan bahwa `dstnat` mengganti **destination IP address** paket saat paket bergerak menuju jaringan private. Ini umum dipakai agar host di jaringan private bisa diakses dari luar. (MikroTik Help)

Contoh kasus:

- Ubuntu Server VM berada di LAN.
- IP Ubuntu: `192.168.10.20`.
- Ubuntu menjalankan web server port `80`.
- Dari luar, akses ke port `8080` router diteruskan ke Ubuntu port `80`.

Topologi:

```
Client luar
    |
  ether1
MikroTik CHR
  ether2
    |
Ubuntu Server
192.168.10.20:80
```

RouterOS v7/v6 — Dst-NAT Port Forwarding

```
/ip firewall nat add chain=dstnat in-interface=ether1
protocol=tcp dst-port=8080 action=dst-nat
to-addresses=192.168.10.20 to-ports=80 comment="Forward 8080
to Ubuntu web"
```

Agar paket diteruskan, firewall forward juga harus mengizinkan.

```
/ip firewall filter add chain=forward protocol=tcp
dst-address=192.168.10.20 dst-port=80 action=accept
comment="Allow forwarded web to Ubuntu"
```

Catatan:

- `dst-nat` mengubah tujuan.
- `srcnat/masquerade` mengubah sumber.
- *Port forwarding* perlu NAT dan firewall forward yang benar.

6.10 Ubuntu Server untuk Uji Port Forwarding

Di Ubuntu Server VM, jalankan web server sederhana.

Contoh uji cepat:

```
sudo apt update
sudo apt install -y nginx
ss -tulpn
```

Cek dari Ubuntu:

```
curl http://127.0.0.1
```

Dari sisi luar/topologi lab, akses:

```
curl http://IP-WAN-MIKROTIK:8080
```

Jika gagal, cek:

- Ubuntu web server aktif atau tidak.
- IP Ubuntu benar atau tidak.
- Rule `dst-nat` benar atau tidak.
- Rule firewall forward mengizinkan atau tidak.
- Arah akses benar atau tidak.

6.11 Hairpin NAT sebagai Pengantar

Hairpin NAT adalah kasus saat client di LAN ingin mengakses server LAN memakai alamat WAN/router.

Contoh:

- Ubuntu Server: `192.168.10.20`.
- Client LAN: `192.168.10.100`.
- Client ingin akses `IP-WAN-Router:8080`.
- Paket masuk ke router lalu “berputar” kembali ke LAN.

Konsep ini penting di jaringan nyata, tetapi untuk pemula cukup dipahami sebagai pengantar.

Masalah umum:

- Dari luar port forwarding jalan.
- Dari LAN akses ke alamat WAN tidak jalan.
- Penyebabnya sering terkait *hairpin NAT* atau DNS internal.

Untuk bab ini, cukup beri pesan:

> **Hairpin NAT dibahas sebagai konsep. Jangan dipaksakan dulu jika lab dasar belum stabil.**

6.12 L7 Firewall: Konsep Layer 7 Matcher

Layer 7 matcher mencoba mencocokkan pola pada isi trafik ICMP/TCP/UDP.

```
Client → MikroTik Firewall → cek pola L7 → allow/drop
```

RouterOS mendokumentasikan `layer7-protocol` sebagai metode pencarian pola pada stream ICMP/TCP/UDP. Tetapi MikroTik juga menegaskan bahwa L7 matcher sangat berat untuk resource, hanya cocok untuk trafik sangat spesifik, dan tidak direkomendasikan untuk trafik generik seperti memblokir halaman web karena hampir tidak akan bekerja benar dan bisa menghabiskan resource perangkat. (MikroTik Help)

Jadi, pendekatan buku ini:

- L7 hanya dikenalkan.
- Dipakai untuk lab kecil.
- Tidak dipakai sebagai solusi utama blokir trafik besar.
- Tidak dipakai untuk semua trafik.
- Jangan dipakai sembarangan di router produksi.

6.13 Contoh L7 RouterOS v7/v6

Contoh sederhana untuk lab:

```
/ip firewall layer7-protocol add name=blok-contoh
regexp="^.+(contoh-domain).*$"
/ip firewall filter add chain=forward
layer7-protocol=blok-contoh action=drop comment="Contoh L7
lab"
```

Catatan penting:

- Rule ini hanya untuk demonstrasi konsep.
- Jangan jadikan L7 sebagai satu-satunya metode kontrol trafik.
- Uji CPU saat L7 aktif.

Cek resource:

```
/system resource print
```

Jika CPU naik tinggi, matikan rule L7:

```
/ip firewall filter print
/ip firewall filter disable [find comment="Contoh L7 lab"]
```

6.14 Lab 6.1 — Blok Akses dari WAN, Izinkan Admin dari LAN

Topologi

```
WAN / NAT GNS3
      |
    ether1
   MikroTik
    ether2
      |
   VPC Admin
192.168.10.100
```

RouterOS v7/v6

```
/ip firewall address-list add list=admin-lan
address=192.168.10.0/24 comment="Admin LAN"
/ip firewall filter add chain=input
connection-state=established,related action=accept
comment="Allow established related"
/ip firewall filter add chain=input connection-state=invalid
action=drop comment="Drop invalid"
/ip firewall filter add chain=input
src-address-list=admin-lan in-interface=ether2 action=accept
comment="Allow LAN admin"
/ip firewall filter add chain=input protocol=icmp
action=accept comment="Allow ping"
/ip firewall filter add chain=input action=drop comment="Drop
other input"
```

Uji VPC

```
PC1> ping 192.168.10.1
```

Target:

- VPC dari LAN bisa ping router.
- Akses dari WAN yang tidak diizinkan akan ditolak.

6.15 Lab 6.2 — Firewall Forward untuk LAN ke WAN

RouterOS v7/v6

```
/ip firewall filter add chain=forward
connection-state=established,related action=accept
comment="Allow established related forward"
/ip firewall filter add chain=forward
connection-state=invalid action=drop comment="Drop invalid
forward"
/ip firewall filter add chain=forward in-interface=ether2
out-interface=ether1 action=accept comment="Allow LAN to WAN"
/ip firewall filter add chain=forward action=drop
comment="Drop other forward"
```

Uji VPC

```
PC1> ping 192.168.10.1
PC1> ping 1.1.1.1
PC1> trace 1.1.1.1
```

Jika ping ke gateway berhasil tetapi ping luar gagal:

- cek NAT Bab 5,
- cek default route,
- cek urutan firewall forward.

6.16 Lab 6.3 — Port Forwarding ke Ubuntu Server

Topologi

```
Jaringan luar
     |
   ether1
MikroTik CHR
   ether2
     |
Ubuntu Server
192.168.10.20:80
```

Ubuntu Server

```
sudo apt update
sudo apt install -y nginx
ss -tulpn
```

RouterOS v7/v6

```
/ip firewall nat add chain=dstnat in-interface=ether1
protocol=tcp dst-port=8080 action=dst-nat
to-addresses=192.168.10.20 to-ports=80 comment="Forward 8080
to Ubuntu web"
/ip firewall filter add chain=forward protocol=tcp
dst-address=192.168.10.20 dst-port=80 action=accept
comment="Allow forwarded web to Ubuntu"
```

Verifikasi RouterOS

```
/ip firewall nat print
/ip firewall filter print
/ip firewall connection print
```

Uji dari client luar/topologi lab

```
curl http://IP-WAN-MIKROTIK:8080
```

Target:

- akses ke port `8080` router diteruskan ke Ubuntu port `80`.

6.17 Lab 6.4 — Uji L7 Sederhana di Lab

RouterOS v7/v6

```
/ip firewall layer7-protocol add name=blok-contoh
regexp="^.+(contoh-domain).*$"
/ip firewall filter add chain=forward
layer7-protocol=blok-contoh action=drop comment="Contoh L7
lab"
```

Cek CPU

```
/system resource print
```

Matikan jika tidak diperlukan

```
/ip firewall filter disable [find comment="Contoh L7 lab"]
```

Pesan utama:

> **L7 hanya untuk memahami konsep inspeksi pola.**
> **Jangan gunakan L7 untuk semua trafik.**

6.18 Verifikasi Firewall

Gunakan perintah berikut.

Cek Rule Filter

```
/ip firewall filter print
```

Cek NAT

```
/ip firewall nat print
```

Cek Connection Tracking

```
/ip firewall connection print
```

Cek Log

```
/log print
```

Cek Resource Jika L7 Aktif

```
/system resource print
```

Uji dari VPC

```
PC1> ping 192.168.10.1
PC1> ping 1.1.1.1
PC1> trace 1.1.1.1
```

Uji dari Ubuntu

```
ip route
ping 192.168.10.1
ping 1.1.1.1
traceroute 1.1.1.1
curl http://192.168.10.20
```

6.19 Kesalahan Umum Firewall

Masalah	Penyebab Umum	Solusi
Router tidak bisa diakses	Rule drop input terlalu awal	Tambahkan allow dari LAN sebelum drop
Client tidak bisa Internet	Forward chain memblokir LAN ke WAN	Tambahkan allow `in-interface=ether2 out-interface=ether1`
NAT ada tapi tetap gagal	Firewall forward drop	Cek `/ip firewall filter print`
Port forwarding gagal	Dst-nat benar, forward belum allow	Tambah rule forward ke server
Log terlalu ramai	Logging semua trafik	Logging hanya rule tertentu
CPU tinggi	L7 dipakai terlalu luas	Batasi/matikan L7
Salah interface	WAN/LAN tertukar	Cek `/interface print`
Rule benar tapi tidak kena	Urutan rule salah	Pindahkan rule ke posisi yang tepat

Catatan tajam:

> **Firewall salah urutan bisa lebih berbahaya daripada tidak ada firewall di lab.**
> Baca rule dari atas ke bawah.

6.20 Checklist Akhir Bab

Pembaca dianggap selesai Bab 6 jika mampu:

- Menjelaskan fungsi firewall.
- Membedakan chain:
  - `input`,
  - `forward`,
  - `output`.
- Menjelaskan *connection state*:
  - `established`,
  - `related`,
  - `invalid`,
  - `new`.
- Membuat firewall input dasar.
- Membuat firewall forward dasar.
- Menggunakan address list.
- Mengaktifkan log sederhana.
- Membuat `dst-nat`.
- Membuat *port forwarding* ke Ubuntu Server.
- Menjelaskan konsep *hairpin NAT*.
- Menjelaskan risiko L7 firewall.
- Melakukan verifikasi dengan:

```
/ip firewall filter print
/ip firewall nat print
/ip firewall connection print
/log print
/system resource print
```

- Menguji dari VPC:

```
PC1> ping 192.168.10.1
PC1> ping 1.1.1.1
PC1> trace 1.1.1.1
```

- Menguji dari Ubuntu:

```
ip route
ping 192.168.10.1
ping 1.1.1.1
traceroute 1.1.1.1
curl http://IP-TUJUAN
```

Ringkasan Bab

Bab ini mengajarkan bahwa firewall bukan sekadar “blokir atau izinkan”. Firewall adalah cara berpikir tentang **arah trafik**, **status koneksi**, **urutan rule**, dan **risiko salah konfigurasi**.

Kunci Bab 6:

> **Amankan router dulu, lalu kendalikan trafik yang melewati router.**
> **Rule firewall harus sedikit, jelas, terurut, dan selalu diuji.**

Bab 7. Bridge, Switching, VLAN Dasar, dan Segmentasi LAN

7.1 Tujuan Bab

Setelah menyelesaikan bab ini, pembaca mampu:

- Menjelaskan **bridge sebagai switch virtual**.
- Menjelaskan fungsi **port bridge**.
- Membedakan **bridge** dan **router**.
- Memahami VLAN dasar:
    - **VLAN ID**
    - **access port**
    - **trunk port**
- Memahami pentingnya segmentasi:
    - **LAN admin**
    - **LAN user**
    - **LAN tamu**
- Membuat bridge di **RouterOS**.
- Memakai bridge di **GNS3**.
- Membuat lab:
    - **1 CHR + 2 VPC dalam satu bridge**
    - **VLAN admin dan VLAN user**
- Menguji konektivitas:
    - *ping* ke gateway
    - *ping* antar client
- Menghindari kesalahan umum:
    - semua port dimasukkan ke satu bridge tanpa desain,
    - IP gateway dipasang di interface salah,
    - VLAN dibuat tanpa gateway.

7.2 Mengapa Bab Ini Penting

Banyak pemula membuat jaringan dengan pola sangat sederhana:

- semua port dijadikan satu jaringan,
- semua perangkat berada di subnet yang sama,
- semua user saling terlihat.

Awalnya ini mudah. Tetapi saat jaringan mulai membesar, masalah mulai muncul:

- trafik bercampur,
- manajemen tidak terpisah,
- user dan tamu berada dalam jaringan yang sama,
- troubleshooting menjadi sulit.

Karena itu, bab ini memperkenalkan dua ide besar:

- **Bridge** untuk membuat beberapa port seolah menjadi satu switch.
- **VLAN** untuk memisahkan jaringan walaupun melewati infrastruktur yang sama.

Prinsip penting:

> **Bridge memudahkan penyatuan port. VLAN memudahkan pemisahan trafik. Router menghubungkan jaringan yang berbeda.**

7.3 Bridge sebagai Switch Virtual

Di RouterOS, **bridge** adalah cara membuat beberapa interface bekerja seperti sebuah switch virtual. Jika beberapa port dimasukkan ke satu bridge, perangkat yang terhubung ke port-port itu bisa berada dalam satu domain *Layer 2* yang sama. Dokumentasi MikroTik tentang *Bridging and Switching* menjelaskan konsep bridge port, PVID, dan perilaku bridge sebagai pusat switching pada level *Layer 2*. (MikroTik Help)

Bayangan sederhananya:

```
ether2 \
        \
         +-- bridge-lan -- satu LAN
        /
ether3 /
```

Jika **ether2** dan **ether3** dimasukkan ke `bridge-lan`, maka:

- host di ether2 dan ether3 bisa saling berkomunikasi di level *Layer 2*,
- IP gateway cukup dipasang di interface **bridge**, bukan di masing-masing port anggota.

Itulah sebabnya bridge sering dipakai untuk:

- membuat LAN sederhana,
- menyatukan beberapa port ke satu segmen,
- dasar untuk VLAN switching,
- laboratorium switching di GNS3.

7.4 Port Bridge

Setelah bridge dibuat, langkah berikutnya adalah memasukkan port ke dalam bridge.

Contoh:

- `ether2` untuk VPC1
- `ether3` untuk VPC2
- keduanya dimasukkan ke `bridge-lan`

Konsepnya:

- **bridge** = wadah / domain switch virtual
- **bridge port** = port fisik atau virtual yang menjadi anggota bridge

CLI RouterOS v7/v6

```
/interface bridge add name=bridge-lan
/interface bridge port add bridge=bridge-lan interface=ether2
/interface bridge port add bridge=bridge-lan interface=ether3
/ip address add address=192.168.20.1/24 interface=bridge-lan
```

Makna konfigurasi:

- membuat bridge bernama `bridge-lan`,
- memasukkan `ether2` dan `ether3` ke bridge itu,
- memberi IP gateway `192.168.20.1/24` ke interface bridge.

Catatan penting:

> **Jika memakai bridge, IP gateway diletakkan di interface bridge, bukan di port anggota seperti ether2 atau ether3.**

7.5 Bridge vs Router

Ini perbedaan yang harus jelas.

Konsep	Fungsi Utama	Contoh
Bridge	menyatukan port dalam satu domain *Layer 2*	`ether2` dan `ether3` jadi satu LAN
Router	menghubungkan jaringan yang berbeda	VLAN admin ke VLAN user, LAN ke WAN
Switching	lalu lintas dalam satu segmen	VPC1 ke VPC2 dalam satu subnet
Routing	lalu lintas antarsegmen	`192.168.10.0/24` ke `192.168.20.0/24`

Contoh:

Kasus Bridge

- VPC1: `192.168.20.10/24`
- VPC2: `192.168.20.20/24`
- gateway: `192.168.20.1`
- ether2 dan ether3 ada dalam satu bridge

Hasil:

- VPC1 dan VPC2 satu LAN.
- Mereka bisa *ping* langsung.

Kasus Router

- VLAN admin: `192.168.10.0/24`
- VLAN user: `192.168.20.0/24`

Hasil:

- dua jaringan berbeda,
- butuh router untuk berkomunikasi.

Prinsip singkat:

> **Bridge menyatukan. Router memisahkan lalu menghubungkan kembali secara terkontrol.**

7.6 VLAN Dasar: Mengapa Dibutuhkan

Jika semua port dimasukkan ke satu bridge, maka semua perangkat berada di satu segmen. Ini sederhana, tetapi tidak selalu aman atau rapi.

VLAN dipakai untuk:

- memisahkan trafik admin,
- memisahkan user biasa,
- memisahkan tamu,
- membangun jaringan lebih terstruktur tanpa menambah banyak perangkat fisik.

Dokumentasi MikroTik *Basic VLAN switching* dan *Bridge VLAN Table* menjelaskan bahwa konfigurasi dasar VLAN dibangun di sekitar **VLAN ID**, **access port**, **trunk port**, dan **PVID**. PVID digunakan untuk menandai trafik *untagged* pada access port, sementara bridge VLAN table menentukan port mana yang menjadi anggota VLAN tertentu. (MikroTik Help)

7.7 VLAN ID, Access Port, dan Trunk Port

VLAN ID

VLAN ID adalah nomor identitas VLAN. Misalnya:

- VLAN 10 = admin
- VLAN 20 = user
- VLAN 30 = tamu

Access Port

Access port adalah port yang membawa trafik untuk **satu VLAN saja**, biasanya *untagged*.

Contoh:

- `ether2` sebagai access port VLAN 10
- `ether3` sebagai access port VLAN 20

Trunk Port

Trunk port adalah port yang membawa **lebih dari satu VLAN**, biasanya *tagged*.

Contoh:

- satu port menuju perangkat lain yang perlu membawa VLAN 10, 20, dan 30 sekaligus.

PVID

> **PVID** menentukan VLAN untuk trafik masuk yang belum bertag pada access port. MikroTik menjelaskan bahwa port dengan PVID tertentu akan menambahkan entry dinamis pada bridge VLAN table, dan port tersebut otomatis menjadi anggota *untagged* untuk VLAN itu. (MikroTik Help)

7.8 Segmentasi LAN: Admin, User, Tamu

Segmentasi adalah pemisahan jaringan berdasarkan fungsi.

Contoh segmentasi sederhana:

Segmen	VLAN	Subnet	Tujuan
LAN admin	10	`192.168.10.0/24`	manajemen, admin
LAN user	20	`192.168.20.0/24`	pengguna umum
LAN tamu	30	`192.168.30.0/24`	tamu / akses terbatas

Keuntungan segmentasi:

- jaringan lebih rapi,
- manajemen lebih mudah,
- risiko kebocoran atau salah akses berkurang,
- kebijakan firewall lebih mudah dibuat,
- troubleshooting lebih mudah.

Catatan kritis:

> **Segmentasi bukan sekadar “membuat VLAN”.**
> **Segmentasi yang benar harus punya: VLAN, gateway, IP plan, dan aturan akses yang jelas.**

7.9 Bridge di GNS3

Untuk lab buku ini, bridge dan VLAN dasar dapat diuji di GNS3 dengan **CHR** dan **VPC**.

Topologi sederhana:

```
VPC1 ---- ether2 CHR
VPC2 ---- ether3 CHR
```

Lalu `ether2` dan `ether3` dimasukkan ke satu bridge.

Kelebihan pendekatan ini:

- ringan,
- mudah diulang,
- mudah diuji dengan `ping`,
- cukup untuk memahami dasar switching.

GNS3 memang cocok untuk membangun topologi kecil maupun menengah, dan VPCS adalah pilihan ringan ketika hanya dibutuhkan perintah sederhana seperti *ping*. (GNS3 Documentation)

7.10 Bridge di RouterOS

Konfigurasi bridge dasar di RouterOS sangat lurus:

CLI RouterOS v7/v6

```
/interface bridge add name=bridge-lan
/interface bridge port add bridge=bridge-lan interface=ether2
/interface bridge port add bridge=bridge-lan interface=ether3
/ip address add address=192.168.20.1/24 interface=bridge-lan
```

Verifikasi

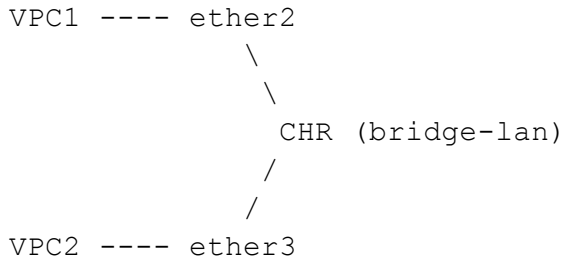
```
/interface bridge print
/interface bridge port print
/ip address print
```

Yang harus terlihat:

- ada `bridge-lan`,
- `ether2` dan `ether3` menjadi anggota,
- IP `192.168.20.1/24` ada di `bridge-lan`.

7.11 Lab 7.1 — 1 CHR + 2 VPC dalam Satu Bridge

Topologi



Tabel IP

Perangkat	Interface	IP	Gateway
CHR	bridge-lan	`192.168.20.1/24`	-
VPC1	eth0	`192.168.20.10/24`	`192.168.20.1`
VPC2	eth0	`192.168.20.20/24`	`192.168.20.1`

CLI RouterOS v7/v6

```
/interface bridge add name=bridge-lan
/interface bridge port add bridge=bridge-lan interface=ether2
/interface bridge port add bridge=bridge-lan interface=ether3
/ip address add address=192.168.20.1/24 interface=bridge-lan
comment="Gateway bridge LAN"
```

VPC1

```
PC1> ip 192.168.20.10/24 192.168.20.1
PC1> ping 192.168.20.1
PC1> ping 192.168.20.20
```

VPC2

```
PC2> ip 192.168.20.20/24 192.168.20.1
PC2> ping 192.168.20.1
PC2> ping 192.168.20.10
```

Target hasil

- VPC1 bisa *ping* gateway.
- VPC2 bisa *ping* gateway.
- VPC1 dan VPC2 bisa saling *ping*.

Makna lab ini:

- `ether2` dan `ether3` sekarang bertindak seperti port switch.
- IP gateway cukup satu di `bridge-lan`.

7.12 VLAN Dasar di RouterOS

Untuk VLAN dasar, kita gunakan satu bridge dengan **`vlan-filtering=yes`**. Ini penting, karena tanpa *vlan filtering*, PVID dan bridge VLAN table tidak bekerja sebagaimana mestinya. MikroTik menegaskan bahwa properti seperti PVID berlaku efektif ketika **`vlan-filtering`** diaktifkan. (MikroTik Help)

Struktur VLAN lab:

- `ether2` = access VLAN 10 (admin)
- `ether3` = access VLAN 20 (user)
- `bridge-vlan` = bridge utama
- dibuat interface VLAN di atas bridge untuk gateway

Konsep penting

- access port: *untagged*, satu VLAN
- trunk port: *tagged*, banyak VLAN
- gateway VLAN: diletakkan pada interface VLAN, bukan pada port fisik access

7.13 Lab 7.2 — VLAN Admin dan VLAN User

Topologi

```
VPC-ADMIN ---- ether2 (access VLAN 10)
                      \
                       CHR bridge-vlan
                      /
VPC-USER  ---- ether3 (access VLAN 20)
```

Tabel IP

Segmen	VLAN	Interface Gateway	IP Gateway
Admin	10	`vlan10-admin`	`192.168.10.1/24`
User	20	`vlan20-user`	`192.168.20.1/24`

CLI RouterOS v7/v6

```
/interface bridge add name=bridge-vlan vlan-filtering=yes

/interface bridge port add bridge=bridge-vlan interface=ether2 pvid=10
/interface bridge port add bridge=bridge-vlan interface=ether3 pvid=20

/interface vlan add interface=bridge-vlan name=vlan10-admin vlan-id=10
/interface vlan add interface=bridge-vlan name=vlan20-user vlan-id=20

/interface bridge vlan add bridge=bridge-vlan vlan-ids=10 tagged=bridge-vlan
untagged=ether2
/interface bridge vlan add bridge=bridge-vlan vlan-ids=20 tagged=bridge-vlan
untagged=ether3

/ip address add address=192.168.10.1/24 interface=vlan10-admin
comment="Gateway VLAN Admin"
/ip address add address=192.168.20.1/24 interface=vlan20-user
comment="Gateway VLAN User"
```

VPC-ADMIN

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> ping 192.168.10.1
```

VPC-USER

```
PC2> ip 192.168.20.10/24 192.168.20.1
PC2> ping 192.168.20.1
```

Uji Antar Client

Dari VPC-ADMIN:

```
PC1> ping 192.168.20.10
```

Dari VPC-USER:

```
PC2> ping 192.168.10.10
```

Makna hasil

- Jika kedua gateway dapat diping, berarti VLAN dan gateway sudah benar.
- Jika antarclient bisa saling *ping*, artinya **routing antar-VLAN** sedang berlangsung melalui router.
- Jika nanti ingin membatasi akses antar-VLAN, gunakan firewall pada bab berikutnya atau integrasikan dengan bab firewall.

Catatan kritis:

> **VLAN memisahkan segmen.**
> **Tetapi jika router merutekan antar-VLAN dan firewall tidak membatasi, antar-VLAN tetap bisa saling terhubung.**

7.14 Access Port vs Trunk Port

Agar pembaca tidak bingung, berikut ringkasannya.

Jenis Port	Membawa VLAN	Bentuk Trafik	Cocok untuk
Access	satu VLAN	biasanya *untagged*	PC, printer, VPC
Trunk	banyak VLAN	biasanya *tagged*	uplink ke perangkat lain

Contoh Access Port

- `ether2` untuk admin, PVID 10
- `ether3` untuk user, PVID 20

Contoh Trunk Port

- `ether4` menuju perangkat lain yang perlu membawa VLAN 10, 20, 30 sekaligus

Contoh tambahan trunk:

```
/interface bridge port add bridge=bridge-vlan
interface=ether4
/interface bridge vlan add bridge=bridge-vlan vlan-ids=10
tagged=bridge-vlan,ether4 untagged=ether2
/interface bridge vlan add bridge=bridge-vlan vlan-ids=20
tagged=bridge-vlan,ether4 untagged=ether3
```

Untuk bab ini, trunk cukup dikenalkan sebagai konsep dasar. Fokus praktik tetap pada access port dan gateway VLAN.

7.15 Kapan Memakai Bridge, Kapan Memakai Routing

Pakai Bridge jika:

- ingin beberapa port berada dalam satu LAN,
- host masih satu subnet,
- belum perlu segmentasi rumit,
- butuh switch virtual sederhana.

Pakai Routing jika:

- ada dua subnet atau lebih,
- ada VLAN admin dan user,
- ingin memisahkan jaringan,
- ingin kontrol akses lebih rapi.

Ringkasnya:

- **Bridge** = satukan port.
- **Routing** = hubungkan subnet.
- **VLAN** = pecah jaringan di level *Layer 2*.
- **Firewall** = atur akses antarsegmen.

7.16 Kesalahan Umum

1. Semua Port Dijadikan Satu Bridge Tanpa Desain

Gejala:

- semua host bercampur,
- tidak ada pemisahan admin/user/tamu.

Solusi:

- gunakan bridge untuk kebutuhan sederhana,
- gunakan VLAN jika mulai ada kebutuhan segmentasi.

2. IP Gateway Dipasang di Interface Salah

Contoh salah:

```
/ip address add address=192.168.20.1/24 interface=ether2
```

Padahal ether2 sudah menjadi anggota bridge.

Yang benar untuk bridge biasa:

```
/ip address add address=192.168.20.1/24
interface=bridge-lan
```

Yang benar untuk VLAN:

```
/ip address add address=192.168.10.1/24
interface=vlan10-admin
```

3. VLAN Dibuat Tanpa Gateway

Gejala:

- VLAN ada,
- tetapi client tidak bisa keluar dari segmennya atau tidak bisa *ping* gateway.

Solusi:

- buat interface VLAN,
- pasang IP gateway pada interface VLAN,
- isi IP client sesuai subnet VLAN.

4. Mengaktifkan VLAN Tanpa Paham PVID

Gejala:

- port terasa “mati”,
- client tidak masuk VLAN yang benar.

Solusi:

- pahami bahwa access port memakai **PVID**,
- cek `bridge vlan` table,
- cek apakah `vlan-filtering=yes` sudah benar.

7.17 Verifikasi

Gunakan perintah berikut di RouterOS:

```
/interface bridge print
/interface bridge port print
/interface bridge vlan print
/interface vlan print
/ip address print
```

Yang harus dicek:

- bridge sudah ada,
- port anggota sudah benar,
- VLAN table sudah sesuai,
- interface VLAN sudah dibuat,
- IP gateway sudah ada di interface yang benar.

Di VPC:

```
show ip
ping 192.168.20.1
ping 192.168.20.20
```

atau untuk VLAN:

```
ping 192.168.10.1
ping 192.168.20.10
```

7.18 Checklist Akhir Bab

Pembaca dianggap selesai Bab 7 jika mampu:

- Menjelaskan **bridge sebagai switch virtual**.
- Menjelaskan fungsi **bridge port**.
- Membedakan **bridge** dan **router**.
- Menjelaskan:
    - **VLAN ID**
    - **access port**
    - **trunk port**
- Menjelaskan mengapa segmentasi penting.
- Membuat bridge sederhana:

```
/interface bridge add name=bridge-lan
/interface bridge port add bridge=bridge-lan interface=ether2
/interface bridge port add bridge=bridge-lan interface=ether3
/ip address add address=192.168.20.1/24 interface=bridge-lan
```

- Membuat VLAN dasar pada bridge.
- Menguji:
    - *ping* ke gateway,
    - *ping* antarclient,
    - *ping* antar-VLAN bila routing ada.
- Menghindari kesalahan:
    - semua port jadi satu bridge tanpa desain,
    - IP gateway di interface salah,
    - VLAN tanpa gateway.

Ringkasan Bab

Bab ini mengajarkan bahwa jaringan yang rapi tidak harus langsung rumit. Mulailah dari **bridge** untuk memahami penyatuan port, lalu masuk ke **VLAN** untuk memahami pemisahan segmen.

Kunci Bab 7:

> **Jangan menyatukan semua port hanya karena mudah.**
> **Gunakan bridge untuk kebutuhan sederhana, dan gunakan VLAN saat jaringan perlu dipisah secara logis.**

Bab 8. Wireless MikroTik dan HotSpot

8.1 Tujuan Bab

Setelah menyelesaikan bab ini, pembaca mampu:

- Menjelaskan konsep dasar **wireless LAN**.
- Menjelaskan arti:
    - **SSID**
    - **channel**
    - **band**
    - **country**
    - **security profile**
- Membedakan:
    - **Access Point**
    - **Station**
- Memahami bahwa **wireless fisik** membutuhkan perangkat dengan radio Wi-Fi.
- Memahami bahwa **CHR** cocok untuk lab HotSpot dan routing, tetapi **tidak mensimulasikan radio wireless fisik**.
- Menjelaskan konsep **HotSpot sebagai captive portal**.
- Membuat **HotSpot dasar** di lab virtual.
- Membuat **user HotSpot**.
- Memahami **profile**, **limit**, dan **walled garden**.
- Mengenal **RADIUS** sebagai pengantar autentikasi terpusat.
- Menguji HotSpot memakai:
    - **VPC GNS3** untuk uji IP dasar,
    - **Ubuntu VM** untuk uji web/login.

8.2 Mengapa Wireless dan HotSpot Penting

Di banyak jaringan nyata, pengguna tidak peduli dengan NAT, route, atau bridge. Yang mereka lihat adalah:

- bisa tersambung ke Wi-Fi atau tidak,
- setelah tersambung apakah Internet jalan,
- apakah harus login,
- apakah akun mereka aktif,
- apakah kecepatan dibatasi.

Karena itu, **wireless** dan **HotSpot** adalah area yang sangat dekat dengan pengalaman pengguna.

Prinsip bab ini:

> **Wireless menghubungkan pengguna ke jaringan. HotSpot mengatur bagaimana pengguna boleh masuk dan memakai jaringan itu.**

8.3 Konsep Wireless LAN

Wireless LAN adalah jaringan lokal berbasis radio, bukan kabel. Pada MikroTik, wireless dipakai untuk:

- *access point*,
- koneksi klien,
- *backhaul*,
- jaringan dalam ruangan,
- koneksi luar ruangan tertentu.

Dokumentasi resmi RouterOS menjelaskan bahwa fitur *wireless* mendukung standar 802.11 dan berbagai fitur keamanan serta mode kerja nirkabel. Untuk perangkat dengan radio yang lebih baru, RouterOS v7 memakai menu **WiFi** untuk pengelolaan interface Wi-Fi 5 *wave2* dan yang lebih baru. (MikroTik Help)

Untuk pemula, cukup pahami lima istilah penting berikut.

8.4 Istilah Penting Wireless

SSID

SSID adalah nama jaringan Wi-Fi yang terlihat oleh pengguna.

Contoh:

- `LAB-MIKROTIK`
- `HOTSPOT-SEKOLAH`
- `WIFI-TAMU`

SSID yang baik:

- mudah dikenali,
- tidak membingungkan,
- sesuai fungsi jaringan.

Channel

Channel adalah kanal radio yang dipakai. Pemilihan channel penting agar interferensi tidak terlalu besar.

Prinsip sederhana:

- channel yang terlalu ramai akan menurunkan kualitas,
- channel harus disesuaikan dengan lingkungan,
- jangan asal memilih channel yang sama untuk semua AP.

Band

Band adalah kelompok frekuensi kerja, misalnya 2.4 GHz atau 5 GHz, tergantung perangkat.

Prinsip praktis:

- 2.4 GHz lebih mudah menjangkau, tetapi sering lebih padat,
- 5 GHz umumnya lebih lega, tetapi jangkauan bisa lebih terbatas.

Country

Parameter **country** penting karena berkaitan dengan aturan kanal dan daya pancar sesuai wilayah.

Prinsip:

> **Jangan sembarang memilih country.**
> Gunakan pengaturan yang sesuai agar kanal dan perilaku radio mengikuti aturan yang benar.

Security Profile

Security profile berisi pengaturan keamanan Wi-Fi, misalnya sandi dan jenis autentikasi.

Fungsi utamanya:

- melindungi akses Wi-Fi,
- membatasi klien yang tidak sah,
- memastikan SSID tidak terbuka sembarangan.

8.5 Access Point vs Station

Dua mode dasar yang harus dipahami:

Mode	Fungsi
Access Point	menyediakan layanan Wi-Fi untuk klien
Station	menjadi klien yang terkoneksi ke Access Point

Access Point

Mode ini dipakai ketika MikroTik bertindak sebagai penyedia jaringan Wi-Fi.

Contoh:

- router kantor,
- router sekolah,
- AP untuk tamu.

Station

Mode ini dipakai ketika perangkat terhubung ke AP lain sebagai klien.

Contoh:

- CPE klien,
- perangkat penghubung ke AP utama.

Prinsip dasar:

Access Point melayani. Station menerima layanan.

8.6 Wireless di Perangkat Fisik MikroTik

Bagian wireless fisik **harus diuji pada perangkat yang memiliki radio Wi-Fi**. Misalnya, router atau access point MikroTik yang memang punya interface wireless.

Ini penting karena:

- CHR adalah RouterOS untuk **virtual machine**,
- CHR bagus untuk routing, firewall, HotSpot, VLAN, dan lab logika jaringan,
- tetapi **CHR tidak menggantikan radio wireless fisik**.

CHR sendiri dijelaskan MikroTik sebagai RouterOS untuk lingkungan virtual. Artinya, CHR sangat cocok untuk lab virtual, tetapi untuk praktik radio Wi-Fi tetap dibutuhkan perangkat fisik yang memiliki interface wireless. (MikroTik)

Catatan penting:

> **Bab ini membagi dua dunia praktik:**
> **wireless fisik** → idealnya di router MikroTik fisik
> **HotSpot dan arsitektur akses** → bisa dilatih di CHR dalam GNS3

8.7 HotSpot sebagai Captive Portal

HotSpot adalah mekanisme *captive portal*. Artinya, pengguna yang terhubung ke jaringan akan diarahkan ke halaman login terlebih dahulu sebelum boleh memakai jaringan lebih jauh.

Dokumentasi MikroTik menyebut HotSpot Gateway sebagai sistem autentikasi klien sebelum akses ke jaringan publik. HotSpot bekerja sebagai *captive portal* dan memakai *web-proxy* dalam prosesnya. (MikroTik Help)

Secara sederhana, alurnya seperti ini:

1. client terhubung ke jaringan,
2. client mendapat IP,
3. client membuka web,
4. router mengarahkan ke halaman login,
5. user memasukkan akun,
6. jika valid, user boleh mengakses jaringan.

HotSpot cocok untuk:

- sekolah,
- kampus,
- lab,
- jaringan tamu,
- area publik kecil,
- pelatihan.

8.8 Komponen Dasar HotSpot

Agar HotSpot bekerja, biasanya dibutuhkan:

- **IP address** pada interface jaringan HotSpot,
- **DHCP Server** untuk klien,
- **IP Pool** untuk pembagian IP,
- **HotSpot Profile**,
- **HotSpot Server**,
- **User HotSpot**.

Dalam lab buku ini, semua komponen dibuat di satu CHR.

8.9 User HotSpot

User HotSpot adalah akun yang dipakai pengguna untuk login.

Contoh:

- `user1 / pass123`
- `siswa01 / belajar123`
- `tamu01 / akses123`

CLI RouterOS v7/v6

```
/ip hotspot user add name=user1 password=pass123
```

Fungsi user:

- mengidentifikasi pemakai,
- membatasi akses,
- menjadi dasar kontrol profil,
- bisa nanti dihubungkan ke RADIUS.

8.10 Profile dan Limit

Profile dipakai untuk menentukan perilaku layanan HotSpot. Dalam praktik, profile bisa mengatur:

- alamat HotSpot,
- nama DNS login,
- halaman login,
- parameter layanan.

Contoh:

```
/ip hotspot profile add name=hsprof
hotspot-address=192.168.30.1 dns-name=login.local
```

Selain itu, di implementasi lebih lanjut, user atau profil bisa dikaitkan dengan pembatasan tertentu, seperti:

- waktu aktif,
- durasi sesi,
- batas kecepatan,
- kebijakan akses.

Untuk pemula, cukup pahami bahwa:

> **Profile adalah template layanan. User adalah identitas pemakai.**

8.11 Walled Garden

Walled garden adalah daftar tujuan yang tetap boleh diakses walaupun user belum login.

Contoh pemakaian:

- halaman informasi,
- portal bantuan,
- situs tertentu milik institusi,
- halaman syarat penggunaan.

Makna praktisnya:

- pengguna belum login,
- tetapi masih boleh membuka beberapa tujuan tertentu.

Ini berguna jika administrator ingin:

- menampilkan informasi,
- mengizinkan akses ke halaman bantuan,
- mengizinkan halaman login eksternal tertentu.

Untuk bab ini, cukup sebagai pengantar konsep. Implementasi rinci bisa menjadi materi lanjutan.

8.12 RADIUS sebagai Pengantar

Untuk lab awal, user HotSpot bisa dibuat langsung di router. Tetapi pada jaringan yang lebih besar, autentikasi sering dipusatkan ke **RADIUS**.

Dokumentasi RouterOS menjelaskan bahwa RouterOS memiliki **RADIUS client** yang dapat dipakai untuk autentikasi HotSpot dan berbagai layanan lain. Parameter yang diterima dari server RADIUS dapat menimpa nilai pada *default profile*. (MikroTik Help)

Makna praktis:

- user tidak perlu disimpan satu per satu di router,
- autentikasi bisa dipusatkan,
- kontrol bisa lebih rapi untuk banyak perangkat.

Untuk bab ini, RADIUS hanya dikenalkan sebagai konsep. Fokus praktik tetap pada HotSpot lokal.

8.13 Arsitektur Lab Bab 8

Bab ini memakai dua pendekatan.

A. Lab HotSpot Virtual

Dipakai untuk:

- memahami alur HotSpot,
- membuat DHCP,
- membuat login portal,
- menguji autentikasi dasar.

Komponen:

- CHR di GNS3,
- VPC sebagai client IP dasar,
- Ubuntu VM sebagai client uji web/login.

B. Wireless Fisik Opsional

Dipakai untuk:

- memahami SSID,
- channel,
- band,
- country,
- security profile,
- mode Access Point.

Komponen:

- router MikroTik fisik dengan radio wireless.

8.14 Lab 8.1 — CHR untuk HotSpot di GNS3

Topologi

```
NAT / jaringan luar
        |
      ether1
   MikroTik CHR
      ether3
        |
   HotSpot LAN
     /     \
   VPC    Ubuntu VM
```

Tabel IP

Perangkat	Interface	IP
CHR	ether1	DHCP / WAN
CHR	ether3	`192.168.30.1/24`
VPC	eth0	DHCP
Ubuntu VM	enp0s3	DHCP

Tujuan lab

- CHR menjadi HotSpot server.
- Client mendapat IP dari DHCP.
- Client diarahkan ke portal login.
- User login memakai akun yang dibuat di router.

8.15 Konfigurasi Dasar HotSpot

RouterOS v7/v6

```
/ip pool add name=pool-hotspot
ranges=192.168.30.100-192.168.30.200
/ip address add address=192.168.30.1/24 interface=ether3
/ip dhcp-server add name=dhcp-hotspot interface=ether3
address-pool=pool-hotspot disabled=no
/ip dhcp-server network add address=192.168.30.0/24
gateway=192.168.30.1 dns-server=192.168.30.1
/ip hotspot profile add name=hsprof
hotspot-address=192.168.30.1 dns-name=login.local
/ip hotspot add name=hotspot1 interface=ether3
address-pool=pool-hotspot profile=hsprof disabled=no
/ip hotspot user add name=user1 password=pass123
```

Jika router juga dipakai untuk akses keluar, pastikan konfigurasi WAN, DNS, default route, dan NAT dari bab sebelumnya sudah aktif.

Tambahkan DNS router jika perlu:

```
/ip dns set allow-remote-requests=yes servers=1.1.1.1
```

Tambahkan NAT jika HotSpot harus keluar ke jaringan luar:

```
/ip firewall nat add chain=srcnat out-interface=ether1
action=masquerade comment="NAT HotSpot to WAN"
```

8.16 Verifikasi Konfigurasi HotSpot

Cek DHCP Server

```
/ip dhcp-server print
/ip dhcp-server lease print
```

Cek HotSpot

```
/ip hotspot print
/ip hotspot profile print
/ip hotspot user print
```

Cek IP

```
/ip address print
```

Yang dicari:

- `ether3` punya `192.168.30.1/24`
- DHCP aktif
- HotSpot server aktif
- user `user1` ada

8.17 VPC sebagai Client HotSpot

VPC sangat ringan dan cocok untuk menguji apakah client mendapat IP dari jaringan HotSpot.

VPC GNS3

```
PC1> ip dhcp
PC1> show ip
PC1> ping 192.168.30.1
```

Target hasil:

- VPC mendapat IP dari pool HotSpot,
- VPC bisa ping gateway HotSpot.

Catatan penting:

> **VPC cocok untuk uji DHCP, IP, dan ping.**
> **Tetapi untuk uji portal login berbasis web, Ubuntu VM jauh lebih cocok.**

Ini sejalan dengan dokumentasi GNS3 bahwa VPCS adalah PC simulator ringan untuk kebutuhan sederhana, bukan klien penuh untuk uji web yang kompleks. (GNS3 Documentation)

8.18 Ubuntu VM sebagai Client Uji

Ubuntu VM dipakai untuk menguji perilaku HotSpot secara lebih nyata.

Di Ubuntu

```
ip addr
ip route
ping 192.168.30.1
```

Setelah Ubuntu mendapat IP, buka browser di Ubuntu VM dan akses situs biasa atau alamat HTTP sederhana. HotSpot seharusnya mengarahkan pengguna ke halaman login.

Masuk memakai akun:

- **username**: `user1`
- **password**: `pass123`

Jika login berhasil:

- user terdaftar sebagai aktif,
- Ubuntu bisa mengakses jaringan lebih lanjut.

Cek user aktif di RouterOS

```
/ip hotspot active print
```

Cek host HotSpot

```
/ip hotspot host print
```

8.19 Wireless Fisik MikroTik sebagai Access Point

Jika pembaca memiliki router MikroTik fisik yang punya radio, maka latihan tambahan bisa dilakukan untuk memahami Wi-Fi nyata.

Fokus pengaturan dasar:

- nama SSID,
- band,
- channel,
- country,
- security profile,
- mode Access Point.

Karena perbedaan perangkat dan driver, sintaks detail dapat berbeda antara perangkat *wireless* lama, perangkat *wifiwave2*, dan perangkat dengan menu **WiFi** baru di RouterOS v7. Dokumentasi MikroTik memang memisahkan konfigurasi wireless klasik dan menu WiFi baru untuk radio yang lebih modern. (MikroTik Help)

Untuk bab ini, yang penting adalah **konsep**:

- perangkat harus punya radio,
- SSID harus jelas,
- country harus benar,
- keamanan harus aktif,
- jangan memakai jaringan wireless terbuka untuk kebutuhan serius.

Topologi Captive Portal dan Autentikasi User

```
Client connect
   ↓
Mendapat IP
   ↓
Buka web
   ↓
Dialihkan ke login HotSpot
   ↓
Masukkan user/password
   ↓
Akses diizinkan
```

HotSpot bekerja sebagai pintu autentikasi sebelum akses penuh diberikan ke pengguna.

8.20 Perbedaan HotSpot dan WPA/Wi-Fi Security

Pemula sering mencampur dua hal ini.

Fitur	Fungsi
Wi-Fi security	mengamankan akses ke jaringan wireless
HotSpot	mengautentikasi dan mengatur akses setelah pengguna masuk jaringan

Contoh:

- SSID memakai sandi Wi-Fi → pengguna boleh masuk ke jaringan nirkabel
- setelah itu HotSpot masih minta login user → pengguna baru boleh akses lebih lanjut

Artinya:

> **Wi-Fi security menjaga pintu masuk radio. HotSpot mengatur izin setelah pengguna masuk ke jaringan.**

8.21 Kesalahan Umum

1. Mengira CHR Bisa Menggantikan Radio Wireless

Gejala:

- ingin menguji SSID dan channel di CHR,
- tetapi tidak ada interface wireless fisik.

Solusi:

- gunakan CHR untuk logika HotSpot dan routing,
- gunakan perangkat fisik untuk uji wireless radio.

2. HotSpot Dibuat Tanpa DHCP

Gejala:

- client tidak mendapat IP,
- portal login tidak muncul.

Solusi:

- pastikan ada IP address, pool, DHCP Server, dan network DHCP.

3. User HotSpot Ada, Tapi Login Gagal

Gejala:

- halaman login muncul,
- user tidak bisa masuk.

Solusi:

- cek `/ip hotspot user print`
- cek ejaan user/password
- cek profile HotSpot

4. Client Bisa Ping Gateway, Tapi Tidak Bisa Keluar

Gejala:

- HotSpot aktif,
- login berhasil,
- tetapi Internet tidak jalan.

Solusi:

- cek default route
- cek DNS
- cek NAT
- cek firewall *forward*

5. Portal Login Tidak Muncul di VPC

Gejala:

- VPC hanya bisa ping, tidak ada halaman login.

Solusi:

- ini normal, karena VPC bukan klien web penuh.
- gunakan Ubuntu VM untuk menguji portal login.

8.22 Checklist Akhir Bab

Pembaca dianggap selesai Bab 8 jika mampu:

- Menjelaskan konsep **wireless LAN**.
- Menjelaskan arti:
    - **SSID**
    - **channel**
    - **band**
    - **country**
    - **security profile**
- Membedakan:
    - **Access Point**
    - **Station**
- Menjelaskan bahwa **CHR tidak mensimulasikan radio wireless fisik**.
- Menjelaskan bahwa **HotSpot** adalah *captive portal*.
- Membuat konfigurasi dasar HotSpot:

```
/ip pool add name=pool-hotspot
ranges=192.168.30.100-192.168.30.200
/ip address add address=192.168.30.1/24 interface=ether3
/ip dhcp-server add name=dhcp-hotspot interface=ether3
address-pool=pool-hotspot disabled=no
/ip dhcp-server network add address=192.168.30.0/24
gateway=192.168.30.1 dns-server=192.168.30.1
/ip hotspot profile add name=hsprof
hotspot-address=192.168.30.1 dns-name=login.local
/ip hotspot add name=hotspot1 interface=ether3
address-pool=pool-hotspot profile=hsprof disabled=no
/ip hotspot user add name=user1 password=pass123
```

- Menguji VPC:

```
PC1> ip dhcp
PC1> show ip
PC1> ping 192.168.30.1
```

- Menguji Ubuntu VM:
    - mendapat IP,
    - membuka portal login,
    - login dengan user HotSpot.
- Menjelaskan konsep:
    - **profile**
    - **limit**
    - **walled garden**
    - **RADIUS**

Ringkasan Bab

Bab ini menegaskan bahwa **wireless** dan **HotSpot** adalah dua lapisan yang saling berkaitan tetapi berbeda. Wireless berbicara tentang **akses radio**, sedangkan HotSpot berbicara tentang **kontrol masuk pengguna** setelah mereka berada di jaringan.

Kunci Bab 8:

> **Untuk belajar wireless, pahami SSID, channel, band, country, dan security. Untuk belajar HotSpot, pahami DHCP, portal login, user, profile, dan alur autentikasi.**

Bab 9. VPN untuk Admin Jaringan: WireGuard, L2TP/IPsec, dan Tunnel Aman

9.1 Tujuan Bab

Setelah menyelesaikan bab ini, pembaca mampu:

- Menjelaskan apa itu **VPN**.
- Membedakan **tunnel** dan routing biasa.
- Membedakan **remote access** dan **site-to-site**.
- Membuat lab **VPN site-to-site** dengan:
    - 2 CHR,
    - 2 VPC,
    - Site A,
    - Site B.
- Membuat WireGuard di **RouterOS v7**.
- Memahami L2TP/IPsec sebagai pilihan di **RouterOS v6/v7**.
- Menjelaskan:
    - **allowed-address**,
    - **key pair**,
    - routing lewat VPN,
    - firewall untuk VPN.
- Melakukan troubleshooting:
    - handshake tidak muncul,
    - route belum ada,
    - firewall memblokir,
    - allowed-address salah.

9.2 Apa Itu VPN

VPN atau *Virtual Private Network* adalah koneksi privat yang dibangun di atas jaringan lain yang tidak selalu kita kendalikan.

Contoh sederhana:

```
Site A ——— Internet / jaringan simulasi ——— Site B
```

Dengan VPN, dua lokasi berbeda bisa seolah-olah terhubung secara aman.

VPN dipakai untuk:

- menghubungkan kantor pusat dan cabang,
- akses admin dari luar lokasi,
- menghubungkan dua lab,
- menghubungkan jaringan antar sekolah/kampus,
- mengamankan lalu lintas antar-router.

Kalimat kunci:

> **VPN bukan sekadar “bisa ping”. VPN adalah jalur aman yang harus punya tunnel, route, firewall, dan kebijakan akses yang jelas.**

9.3 Tunnel vs Routing Biasa

Routing biasa meneruskan paket melalui jalur jaringan yang sudah ada.

Tunnel membuat jalur virtual di atas jaringan yang sudah ada.

Contoh routing biasa:

```
VPC-A → R1 → jaringan tengah → R2 → VPC-B
```

Contoh lewat VPN:

```
VPC-A → R1 → tunnel VPN → R2 → VPC-B
```

Perbedaannya:

Aspek	Routing Biasa	VPN Tunnel
Jalur	langsung lewat routing	lewat interface tunnel
Keamanan	tergantung jaringan	memakai mekanisme enkripsi/autentikasi
Identitas peer	tidak selalu kuat	memakai key/secret
Cocok untuk	jaringan internal	jaringan antar lokasi

Catatan penting:

> **VPN tetap membutuhkan routing.**
> Tunnel hanya membuka jalan. Route menentukan subnet mana yang lewat tunnel.

9.4 Remote Access vs Site-to-Site

Ada dua pola besar VPN.

Remote Access

Dipakai jika satu admin atau satu laptop masuk ke jaringan kantor/lab.

Contoh:

```
Laptop admin → VPN → Router kantor
```

Cocok untuk:

- admin bekerja dari luar lokasi,
- troubleshooting jarak jauh,
- akses perangkat internal.

Site-to-Site

Dipakai untuk menghubungkan dua jaringan.

Contoh:

```
LAN Site A → R1 → VPN → R2 → LAN Site B
```

Cocok untuk:

- kantor pusat dan cabang,
- lab antar-router,
- koneksi antarkampus,
- latihan routing antar-site.

Bab ini fokus pada **site-to-site** karena lebih mudah dipahami dengan GNS3 dan VPC.

9.5 WireGuard untuk RouterOS v7

WireGuard adalah pilihan utama di RouterOS v7 untuk lab VPN modern karena konfigurasinya relatif ringkas.

Konsep yang harus dipahami:

- **Interface WireGuard**
  Interface tunnel virtual di router.
- **Listen Port**
  Port UDP tempat router menerima koneksi WireGuard.
- **Private Key dan Public Key**
  Pasangan kunci untuk identitas dan autentikasi.
- **Peer**
  Router lawan.
- **Allowed Address**
  Alamat peer dan subnet yang boleh lewat tunnel.
- **Route**
  Jalur menuju subnet lawan melalui WireGuard.

Dokumentasi MikroTik menekankan penggunaan **public key**, **endpoint**, dan **allowed-address** dalam konfigurasi WireGuard. Dalam contoh remote, *allowed IPs* dapat dipakai untuk menentukan trafik mana yang dikirim melalui tunnel. (MikroTik Help)

9.6 L2TP/IPsec untuk RouterOS v6/v7

RouterOS v6 tidak memiliki WireGuard native. Karena itu, pilihan umum untuk VPN di RouterOS v6 adalah **L2TP/IPsec** atau tunnel lain yang didukung.

L2TP sendiri memperluas model PPP agar endpoint PPP dapat berada di perangkat berbeda melalui jaringan paket. L2TP memakai UDP untuk paket kontrol dan data. Untuk keamanan, L2TP biasanya dipasangkan dengan IPsec. MikroTik menjelaskan IPsec sebagai kumpulan protokol untuk mengamankan pertukaran paket di jaringan IP/IPv6 yang tidak terlindungi. (MikroTik Help)

Dalam buku ini:

- **RouterOS v7** → lab utama WireGuard.
- **RouterOS v6** → catatan pembanding memakai L2TP/IPsec.
- Sintaks v6 tidak dijadikan lab utama panjang agar buku tetap padat.

9.7 Arsitektur Lab VPN

Lab memakai 2 CHR dan 2 VPC.

```
LAN-A                     WAN Simulasi            LAN-B
192.168.10.0/24         172.16.99.0/24          192.168.20.0/24
VPC-A ------------ R1 ----------------- R2 --- VPC-B
```

Tabel IP

Perangkat	Interface	IP
R1	LAN-A	192.168.10.1/24
R1	WAN Simulasi	172.16.99.1/24
R2	WAN Simulasi	172.16.99.2/24
R2	LAN-B	192.168.20.1/24
VPC-A	eth0	192.168.10.10/24
VPC-B	eth0	192.168.20.10/24
WireGuard R1	wg-site	10.10.10.1/30
WireGuard R2	wg-site	10.10.10.2/30

Tujuan Lab

- R1 dan R2 bisa saling ping melalui jaringan WAN simulasi.
- WireGuard terbentuk antara R1 dan R2.
- VPC-A bisa ping VPC-B melalui tunnel.
- Route antar-LAN berjalan lewat WireGuard.

9.8 Konfigurasi Dasar Site A dan Site B

RouterOS v7/v6 — R1

```
/system identity set name=R1-SITE-A
/ip address add address=192.168.10.1/24 interface=ether1
comment="LAN-A"
/ip address add address=172.16.99.1/24 interface=ether2
comment="WAN-SIM"
```

RouterOS v7/v6 — R2

```
/system identity set name=R2-SITE-B
/ip address add address=172.16.99.2/24 interface=ether1
comment="WAN-SIM"
/ip address add address=192.168.20.1/24 interface=ether2
comment="LAN-B"
```

VPC-A

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> ping 192.168.10.1
```

VPC-B

```
PC2> ip 192.168.20.10/24 192.168.20.1
PC2> ping 192.168.20.1
```

Uji antar-router

Dari R1:

```
/tool ping 172.16.99.2
```

Dari R2:

```
/tool ping 172.16.99.1
```

Jika dua router belum bisa saling ping, jangan lanjut ke WireGuard.

9.9 Key Pair WireGuard

WireGuard memakai pasangan kunci:

- private key disimpan di router sendiri,
- public key diberikan ke router lawan.

Di RouterOS, saat interface WireGuard dibuat, key dapat dihasilkan otomatis.

RouterOS v7 — R1

```
/interface wireguard add name=wg-site listen-port=13231
/ip address add address=10.10.10.1/30 interface=wg-site
/interface wireguard print detail
```

Catat public-key R1.

```
RouterOS v7 — R2
/interface wireguard add name=wg-site listen-port=13231
/ip address add address=10.10.10.2/30 interface=wg-site
/interface wireguard print detail
```

Catat public-key R2.

Catatan:

- Public key R1 dimasukkan ke peer R2.
- Public key R2 dimasukkan ke peer R1.

Jangan tertukar.

9.10 Konfigurasi Peer WireGuard

Misal:

- Public key R1 = `PUBLIC_KEY_R1`
- Public key R2 = `PUBLIC_KEY_R2`

RouterOS v7 — R1

```
/interface wireguard peers add interface=wg-site
public-key="PUBLIC_KEY_R2" endpoint-address=172.16.99.2
endpoint-port=13231
allowed-address=10.10.10.2/32,192.168.20.0/24
persistent-keepalive=25

/ip route add dst-address=192.168.20.0/24 gateway=10.10.10.2
```

RouterOS v7 — R2

```
/interface wireguard peers add interface=wg-site
public-key="PUBLIC_KEY_R1" endpoint-address=172.16.99.1
endpoint-port=13231
allowed-address=10.10.10.1/32,192.168.10.0/24
persistent-keepalive=25

/ip route add dst-address=192.168.10.0/24 gateway=10.10.10.1
```

Makna penting:

- `endpoint-address` = alamat WAN router lawan.
- `endpoint-port` = port WireGuard lawan.
- `allowed-address` = IP tunnel lawan dan subnet LAN lawan.
- `gateway` route = IP WireGuard lawan.

9.11 Firewall untuk VPN

WireGuard memakai UDP. Jika firewall input aktif, port WireGuard harus diizinkan.

RouterOS v7 — R1 dan R2

```
/ip firewall filter add chain=input protocol=udp
dst-port=13231 action=accept comment="Allow WireGuard VPN"
```

Letakkan rule ini **sebelum rule drop input**.

Jika ada rule firewall forward yang ketat, izinkan trafik antar-LAN lewat WireGuard.

R1

```
/ip firewall filter add chain=forward
src-address=192.168.10.0/24 dst-address=192.168.20.0/24
action=accept comment="Allow LAN-A to LAN-B via VPN"

/ip firewall filter add chain=forward
src-address=192.168.20.0/24 dst-address=192.168.10.0/24
action=accept comment="Allow LAN-B to LAN-A via VPN"
```

R2

```
/ip firewall filter add chain=forward
src-address=192.168.20.0/24 dst-address=192.168.10.0/24
action=accept comment="Allow LAN-B to LAN-A via VPN"

/ip firewall filter add chain=forward
src-address=192.168.10.0/24 dst-address=192.168.20.0/24
action=accept comment="Allow LAN-A to LAN-B via VPN"
```

Catatan tajam:

> **VPN bisa terbentuk, tetapi trafik LAN tetap gagal jika firewall forward memblokir.**

9.12 Routing Lewat VPN

Tunnel tidak otomatis berarti semua subnet saling terhubung. Harus ada route.

Di R1:

```
/ip route add dst-address=192.168.20.0/24 gateway=10.10.10.2
```

Di R2:

```
/ip route add dst-address=192.168.10.0/24 gateway=10.10.10.1
```

Verifikasi:

```
/ip route print
```

Yang dicari:

- R1 punya route ke `192.168.20.0/24`.
- R2 punya route ke `192.168.10.0/24`.

Paket flow di RouterOS tetap mengikuti proses routing: paket masuk interface, router menentukan paket perlu dirutekan, lalu paket keluar interface tujuan. Ini penting karena VPN tetap harus diperlakukan sebagai jalur routing. (MikroTik Help)

9.13 Lab 9.1 — VPN Antardua Site

Langkah 1 — Siapkan IP Dasar

R1:

```
/system identity set name=R1-SITE-A
/ip address add address=192.168.10.1/24 interface=ether1
comment="LAN-A"
/ip address add address=172.16.99.1/24 interface=ether2
comment="WAN-SIM"
```

R2:

```
/system identity set name=R2-SITE-B
/ip address add address=172.16.99.2/24 interface=ether1
comment="WAN-SIM"
/ip address add address=192.168.20.1/24 interface=ether2
comment="LAN-B"
```

Langkah 2 — Siapkan VPC

VPC-A:

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> ping 192.168.10.1
```

VPC-B:

```
PC2> ip 192.168.20.10/24 192.168.20.1
PC2> ping 192.168.20.1
```

Langkah 3 — Buat WireGuard

R1:

```
/interface wireguard add name=wg-site listen-port=13231
/ip address add address=10.10.10.1/30 interface=wg-site
/interface wireguard print detail
```

R2:

```
/interface wireguard add name=wg-site listen-port=13231
/ip address add address=10.10.10.2/30 interface=wg-site
/interface wireguard print detail
```

Tukar public key antar-router.

Langkah 4 — Tambah Peer

R1:

```
/interface wireguard peers add interface=wg-site
public-key="PUBLIC_KEY_R2" endpoint-address=172.16.99.2
endpoint-port=13231
allowed-address=10.10.10.2/32,192.168.20.0/24
persistent-keepalive=25
/ip route add dst-address=192.168.20.0/24
gateway=10.10.10.2
```

R2:

```
/interface wireguard peers add interface=wg-site
public-key="PUBLIC_KEY_R1" endpoint-address=172.16.99.1
endpoint-port=13231
allowed-address=10.10.10.1/32,192.168.10.0/24
persistent-keepalive=25
/ip route add dst-address=192.168.10.0/24
gateway=10.10.10.1
```

Langkah 5 — Buka Firewall VPN

R1 dan R2:

```
/ip firewall filter add chain=input protocol=udp
dst-port=13231 action=accept comment="Allow WireGuard
VPN"
```

Langkah 6 — Uji Tunnel

Dari R1:

```
/tool ping 10.10.10.2
/tool ping 192.168.20.1
```

Dari R2:

```
/tool ping 10.10.10.1
/tool ping 192.168.10.1
```

Dari VPC-A:

```
PC1> ping 192.168.20.10
```

Dari VPC-B:

```
PC2> ping 192.168.10.10
```

Target akhir:

- R1 dan R2 saling ping melalui IP WireGuard.
- VPC-A bisa ping VPC-B.
- VPC-B bisa ping VPC-A.

9.14 Catatan RouterOS v6: L2TP/IPsec

Karena WireGuard native tidak tersedia di RouterOS v6, gunakan L2TP/IPsec sebagai pembanding.

Konsep minimal:

- aktifkan L2TP Server di salah satu router,
- buat PPP secret,
- aktifkan IPsec secret,
- router lawan menjadi L2TP Client,
- tambahkan route ke subnet lawan,
- buka firewall untuk layanan yang dibutuhkan.

Contoh sangat ringkas pada router server:

```
/interface l2tp-server server set enabled=yes use-ipsec=yes
ipsec-secret=SecretKuat authentication=mschap2

/ppp secret add name=site-b password=PasswordKuat
service=l2tp local-address=10.20.20.1
remote-address=10.20.20.2
```

Contoh client:

```
/interface l2tp-client add name=l2tp-to-site-a
connect-to=172.16.99.1 user=site-b password=PasswordKuat
use-ipsec=yes ipsec-secret=SecretKuat disabled=no
```

Catatan:

- Sintaks dan perilaku bisa berbeda tergantung versi RouterOS.
- Gunakan password dan secret kuat.
- Pastikan route dan firewall benar.
- L2TP/IPsec dibahas sebagai alternatif, bukan lab utama panjang.

9.15 Allowed Address

Allowed address adalah bagian yang sering salah dipahami.

Pada WireGuard, allowed-address berfungsi untuk menentukan:

- IP peer di tunnel,
- subnet yang dianggap berada di belakang peer,
- trafik yang boleh lewat peer tersebut.

Contoh di R1:

```
allowed-address=10.10.10.2/32,192.168.20.0/24
```

Artinya:

- peer memiliki IP tunnel `10.10.10.2`,
- subnet di belakang peer adalah `192.168.20.0/24`.

Contoh kesalahan:

```
allowed-address=10.10.10.2/32
```

Akibat:

- ping IP tunnel bisa berhasil,
- tetapi LAN Site B tidak bisa dicapai.

9.16 Troubleshooting VPN

1. Handshake Tidak Muncul

Cek:

```
/interface wireguard peers print detail
```

Kemungkinan:

- public key salah,
- endpoint address salah,
- port UDP diblokir,
- firewall input belum membuka port,
- R1 dan R2 belum bisa saling ping di jaringan WAN simulasi.

Cek awal:

```
/tool ping 172.16.99.2
/ip firewall filter print
```

2. Route Belum Ada

Gejala:

- tunnel terbentuk,
- IP WireGuard bisa diping,
- tetapi LAN lawan tidak bisa.

Cek:

```
/ip route print
```

Pastikan:

- R1 punya route ke `192.168.20.0/24`.
- R2 punya route ke `192.168.10.0/24`.

3. Firewall Memblokir

Gejala:

- route benar,
- WireGuard aktif,
- tetapi paket LAN gagal.

Cek:

```
/ip firewall filter print
/ip firewall connection print
```

Pastikan:

- UDP port WireGuard diizinkan di chain input.
- Forward antar-LAN diizinkan.

4. Allowed Address Salah

Gejala:

- handshake ada,
- tunnel IP bisa,
- subnet lawan gagal.

Cek:

```
/interface wireguard peers print detail
```

Pastikan allowed-address berisi:

- IP tunnel peer,
- subnet LAN peer.

5. Gateway Route Salah

Contoh salah:

```
/ip route add dst-address=192.168.20.0/24
gateway=172.16.99.2
```

Untuk lab WireGuard, sebaiknya route ke LAN lawan lewat IP tunnel:

```
/ip route add dst-address=192.168.20.0/24
gateway=10.10.10.2
```

9.17 Verifikasi Akhir

RouterOS v7

```
/interface wireguard print
/interface wireguard peers print detail
/ip address print
/ip route print
/ip firewall filter print
/tool ping 10.10.10.2
/tool ping 192.168.20.1
```

VPC-A

```
PC1> show ip
PC1> ping 192.168.10.1
PC1> ping 192.168.20.10
PC1> trace 192.168.20.10
```

VPC-B

```
PC2> show ip
PC2> ping 192.168.20.1
PC2> ping 192.168.10.10
PC2> trace 192.168.10.10
```

9.18 Checklist Akhir Bab

Pembaca dianggap selesai Bab 9 jika mampu:

- Menjelaskan apa itu **VPN**.
- Membedakan:
    - tunnel,
    - routing biasa,
    - remote access,
    - site-to-site.
- Membuat WireGuard di RouterOS v7:

```
/interface wireguard add name=wg-site listen-port=13231
/ip address add address=10.10.10.1/30 interface=wg-site
```

- Menambahkan peer WireGuard.
- Menjelaskan fungsi:
    - public key,
    - private key,
    - endpoint,
    - allowed-address,
    - persistent keepalive.
- Membuat route lewat VPN.
- Membuka firewall untuk port VPN.
- Menguji dengan VPC GNS3.
- Menjelaskan bahwa RouterOS v6 memakai alternatif seperti L2TP/IPsec.
- Melakukan troubleshooting:
    - handshake tidak muncul,
    - route belum ada,
    - firewall memblokir,
    - allowed-address salah.

Ringkasan Bab

Bab ini mengajarkan VPN sebagai **jalur aman antar lokasi**, bukan sekadar konfigurasi tambahan. WireGuard di RouterOS v7 memberi cara yang ringkas untuk membuat site-to-site VPN, sedangkan RouterOS v6 dapat memakai L2TP/IPsec sebagai alternatif.

Kunci Bab 9:

> **VPN berhasil jika empat hal benar: tunnel terbentuk, route ada, firewall mengizinkan, dan allowed-address tidak salah.**

Bab 10. IPv6 dengan VPC GNS3 dan Ubuntu 26.04 di VirtualBox

10.1 Tujuan Bab

Setelah menyelesaikan bab ini, pembaca mampu:

- Menjelaskan mengapa **IPv6 penting**.
- Membaca format alamat IPv6.
- Memahami **prefix IPv6**.
- Memahami fungsi **link-local address**.
- Menjelaskan konsep **Neighbor Discovery**.
- Memahami **SLAAC** dan pengantar **DHCPv6**.
- Membuat lab **dual-stack IPv4/IPv6**.
- Mengaktifkan IPv6 di **RouterOS**.
- Mengatur IPv6 di **VPC GNS3**.
- Mengatur dan menguji IPv6 di **Ubuntu 26.04**.
- Melakukan:
    - ping IPv6 dari VPC ke MikroTik,
    - ping IPv6 dari Ubuntu ke MikroTik,
    - routing IPv6 sederhana.
- Membuat firewall IPv6 dasar:
    - izinkan koneksi yang sah,
    - drop paket invalid,
    - izinkan ICMPv6,
    - drop akses yang tidak perlu.

10.2 Mengapa IPv6 Penting

IPv4 memakai alamat 32-bit. IPv6 memakai alamat 128-bit. Artinya, ruang alamat IPv6 jauh lebih besar.

Tetapi alasan belajar IPv6 bukan hanya “alamat lebih banyak”. IPv6 penting karena:

- jaringan modern makin banyak perangkat,
- banyak sistem baru mulai memakai dual-stack,
- IPv6 tidak memakai broadcast seperti IPv4,
- IPv6 memakai **Neighbor Discovery**,
- firewall IPv6 harus dipahami sejak awal,
- admin jaringan harus siap mengelola IPv4 dan IPv6 bersamaan.

MikroTik menjelaskan dalam materi fundamental IP bahwa IP dipakai untuk mengidentifikasi host dengan alamat logis unik, dan IPv4/IPv6 adalah bagian dari fondasi jaringan Layer 3. (MikroTik Help)

Catatan penting:

> **IPv6 bukan sekadar IPv4 dengan alamat panjang.**
> **IPv6 punya cara kerja sendiri, terutama pada Neighbor Discovery dan ICMPv6.**

10.3 Format Alamat IPv6

Alamat IPv6 ditulis dalam format heksadesimal dan dipisahkan dengan tanda titik dua.

Contoh:

```
2001:db8:10::1
```

Contoh lain:

```
2001:db8:10::10
fe80::1
```

Alamat IPv6 dapat disingkat:

```
2001:db8:10:0:0:0:0:1
```

menjadi:

```
2001:db8:10::1
```

Untuk lab, gunakan prefix dokumentasi:

```
2001:db8::/32
```

Contoh lab buku ini:

```
2001:db8:10::/64
2001:db8:20::/64
2001:db8:99::/64
```

Catatan:

> **Gunakan `2001:db8::/32` untuk dokumentasi dan lab.**
> **Jangan memakai alamat IPv6 publik sembarangan.**

10.4 Prefix IPv6

IPv6 memakai **prefix**, mirip konsep CIDR pada IPv4.

Contoh:

```
2001:db8:10::1/64
```

Artinya:

- alamat host: `2001:db8:10::1`
- prefix jaringan: `2001:db8:10::/64`
- panjang prefix: `/64`

Dalam praktik LAN IPv6, `/64` sangat umum dipakai.

Contoh tabel:

Kebutuhan	Prefix	Contoh
LAN IPv6	`/64`	`2001:db8:10::/64`
Link antar-router lab	`/64` atau lebih spesifik sesuai desain	`2001:db8:99::/64`
Alamat dokumentasi	`2001:db8::/32`	khusus contoh/lab

Prinsip:

> **Untuk lab pemula, gunakan `/64` agar sederhana dan sesuai pola umum IPv6 LAN.**

10.5 Link-local Address

Link-local address adalah alamat IPv6 lokal pada satu link. Biasanya diawali:

```
fe80::
```

Contoh:

```
fe80::1
fe80::abcd
```

Link-local dipakai untuk komunikasi lokal, termasuk **Neighbor Discovery**.

MikroTik menjelaskan bahwa **link-local address diperlukan untuk proses IPv6 Neighbor Discovery**. Jika interface menjadi port bridge, alamat link-local spesifik interface dapat dihapus dan tersisa link-local milik bridge. (MikroTik Help)

Makna praktis:

- jangan heran jika interface punya alamat `fe80::...`,
- link-local bukan alamat global,
- link-local penting untuk mekanisme lokal IPv6,
- jangan menghapus atau mengabaikannya sembarangan.

10.6 Neighbor Discovery

Pada IPv4, kita mengenal ARP. Pada IPv6, fungsi serupa ditangani oleh **Neighbor Discovery**.

Neighbor Discovery dipakai untuk:

- menemukan perangkat tetangga,
- menemukan router,
- resolusi alamat Layer 2,
- mendukung autokonfigurasi,
- mendeteksi tetangga yang masih aktif.

Di IPv6, **ICMPv6** sangat penting. Karena itu, firewall IPv6 tidak boleh asal memblokir semua ICMPv6.

Catatan tajam:

> **Di IPv4, banyak orang berani memblokir ICMP.**
> **Di IPv6, memblokir ICMPv6 sembarangan bisa merusak fungsi jaringan.**

10.7 SLAAC

SLAAC atau *Stateless Address Autoconfiguration* memungkinkan host membuat alamat IPv6 sendiri berdasarkan informasi prefix yang diumumkan router.

Gambaran sederhana:

Router mengumumkan prefix → client membentuk alamat IPv6 sendiri

Contoh:

- router punya prefix `2001:db8:10::/64`,
- client menerima informasi prefix,
- client membuat alamat IPv6 dalam prefix itu.

Di RouterOS, parameter `advertise=yes` pada IPv6 address dapat dipakai agar router mengiklankan prefix ke client.

Contoh:

```
/ipv6 address add address=2001:db8:10::1/64 interface=ether2
advertise=yes
```

Makna:

- MikroTik punya IPv6 `2001:db8:10::1/64`,
- prefix dapat diumumkan ke client pada interface tersebut.

10.8 DHCPv6 sebagai Pengantar

DHCPv6 adalah mekanisme untuk memberikan konfigurasi IPv6 ke client.

Namun, untuk bab ini, DHCPv6 hanya dikenalkan sebagai konsep karena lab utama memakai:

- IPv6 statik di VPC,
- IPv6 statik atau auto konfigurasi dasar di Ubuntu,
- RouterOS sebagai router IPv6 sederhana.

Perbedaan sederhana:

Mekanisme	**Fungsi**
SLAAC	client membentuk alamat sendiri dari prefix
DHCPv6	server memberikan konfigurasi IPv6 tertentu
Static IPv6	alamat diisi manual untuk lab

Untuk pemula:

> **Mulai dari IPv6 statik dulu. Setelah paham prefix dan routing, baru masuk DHCPv6/SLAAC lebih dalam.**

10.9 Dual-stack IPv4/IPv6

Dual-stack berarti satu perangkat menjalankan IPv4 dan IPv6 bersamaan.

Contoh VPC/Ubuntu:

```
IPv4 : 192.168.10.10/24
IPv6 : 2001:db8:10::10/64
```

Contoh MikroTik:

```
IPv4 : 192.168.10.1/24
IPv6 : 2001:db8:10::1/64
```

Manfaat dual-stack:

- jaringan IPv4 tetap berjalan,
- IPv6 bisa dipelajari tanpa memutus lab lama,
- troubleshooting lebih aman,
- transisi lebih realistis.

Catatan:

> **Jangan mematikan IPv4 hanya untuk belajar IPv6.**
> **Gunakan dual-stack agar lab tetap mudah diuji.**

10.10 IPv6 di RouterOS

Sebelum konfigurasi, cek apakah menu IPv6 tersedia.

```
/ipv6 address print
/ipv6 route print
```

Tambahkan IPv6 ke interface LAN.

RouterOS v7/v6

```
/ipv6 address add address=2001:db8:10::1/64 interface=ether2
advertise=yes
```

Cek hasil:

```
/ipv6 address print
/ipv6 route print
```

Yang harus terlihat:

- `2001:db8:10::1/64` pada `ether2`,
- route connected untuk `2001:db8:10::/64`.

Catatan:

- Jika interface memakai bridge, pasang IPv6 di interface bridge.
- Jangan pasang gateway IPv6 di port anggota bridge jika desain memakai bridge.

10.11 IPv6 di VPC GNS3

GNS3 mendukung penggunaan VPCS untuk lab sederhana. Untuk IPv6, sintaks umum VPC mengikuti pola:

```
ip <IPv6-address>/<prefix-length> <default-gateway>
```

Dalam outline attached, sintaks VPC GNS3 IPv6 yang dipakai adalah:

```
PC1> ip 2001:db8:10::10/64 2001:db8:10::1
PC1> show ipv6
PC1> ping 2001:db8:10::1
```

Ini sejalan dengan penggunaan VPC sebagai client ringan untuk uji lab IPv6 dalam buku .

VPC GNS3

```
PC1> ip 2001:db8:10::10/64 2001:db8:10::1
PC1> show ipv6
PC1> ping 2001:db8:10::1
```

Target:

- VPC punya IPv6 `2001:db8:10::10/64`,
- gateway IPv6 `2001:db8:10::1`,
- ping ke MikroTik berhasil.

10.12 IPv6 di Ubuntu 26.04

Ubuntu VM dipakai untuk uji yang lebih lengkap dibanding VPC.

Perintah dasar:

```
ip -6 addr
ip -6 route
ping -6 2001:db8:10::1
```

Jika ingin memberi IPv6 sementara:

```
sudo ip -6 addr add 2001:db8:10::20/64 dev enp0s3
sudo ip -6 route add default via 2001:db8:10::1
```

Cek:

```
ip -6 addr
ip -6 route
ping -6 2001:db8:10::1
```

Catatan:

- Nama interface bisa berbeda.
- Cek dengan `ip addr`.
- Jangan lupa memakai `ping -6` agar jelas memakai IPv6.

10.13 Lab 10.1 — CHR IPv6 Router + VPC IPv6

Topologi

```
VPC1 IPv6 ---- ether2 MikroTik CHR
```

Tabel IP

Perangkat	Interface	IPv6
R1 CHR	ether2	`2001:db8:10::1/64`
VPC1	eth0	`2001:db8:10::10/64`
Gateway VPC	-	`2001:db8:10::1`

RouterOS v7/v6

```
/system identity set name=R1-IPV6
/ipv6 address add address=2001:db8:10::1/64 interface=ether2
advertise=yes
/ipv6 address print
/ipv6 route print
```

VPC GNS3

```
PC1> ip 2001:db8:10::10/64 2001:db8:10::1
PC1> show ipv6
PC1> ping 2001:db8:10::1
```

Target hasil

- VPC memiliki alamat IPv6.
- VPC bisa ping IPv6 MikroTik.
- MikroTik punya route connected ke `2001:db8:10::/64`.

10.14 Lab 10.2 — CHR IPv6 Router + Ubuntu VM

Topologi

```
Ubuntu VM ---- ether2 MikroTik CHR
```

Tabel IP

Perangkat	Interface	IPv6
R1 CHR	ether2	2001:db8:10::1/64
Ubuntu VM	enp0s3	2001:db8:10::20/64
Gateway Ubuntu	-	2001:db8:10::1

RouterOS v7/v6

```
/ipv6 address add address=2001:db8:10::1/64 interface=ether2
advertise=yes
/ipv6 address print
```

Ubuntu 26.04

```
sudo ip -6 addr add 2001:db8:10::20/64 dev enp0s3
sudo ip -6 route add default via 2001:db8:10::1
ip -6 addr
ip -6 route
ping -6 2001:db8:10::1
```

Target hasil

- Ubuntu punya IPv6.
- Ubuntu punya default route IPv6.
- Ubuntu bisa ping MikroTik.

10.15 Lab 10.3 — Dual-stack IPv4/IPv6

Topologi

```
VPC1 / Ubuntu VM ---- MikroTik CHR
```

Tabel IP

Perangkat	IPv4	IPv6
MikroTik	192.168.10.1/24	2001:db8:10::1/64
VPC1	192.168.10.10/24	2001:db8:10::10/64
Ubuntu VM	192.168.10.20/24	2001:db8:10::20/64

RouterOS v7/v6

```
/ip address add address=192.168.10.1/24 interface=ether2
comment="IPv4 LAN"
/ipv6 address add address=2001:db8:10::1/64 interface=ether2
advertise=yes
```

VPC

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> ip 2001:db8:10::10/64 2001:db8:10::1
PC1> ping 192.168.10.1
PC1> ping 2001:db8:10::1
```

Makna lab

- IPv4 dan IPv6 bisa berjalan bersamaan.
- Jika IPv4 berhasil tetapi IPv6 gagal, cek konfigurasi IPv6.
- Jika IPv6 berhasil tetapi IPv4 gagal, cek konfigurasi IPv4.
- Dual-stack memudahkan isolasi masalah.

10.16 Lab 10.4 — Routing IPv6 Sederhana

Topologi

```
VPC-A ---- R1 ---- R2 ---- VPC-B
LAN-A IPv6: 2001:db8:10::/64
Link R1-R2: 2001:db8:99::/64
LAN-B IPv6: 2001:db8:20::/64
```

Tabel IP

Perangkat	Interface	IPv6
R1	LAN-A	2001:db8:10::1/64
R1	R1-R2	2001:db8:99::1/64
R2	R1-R2	2001:db8:99::2/64
R2	LAN-B	2001:db8:20::1/64
VPC-A	eth0	2001:db8:10::10/64
VPC-B	eth0	2001:db8:20::10/64

R1 — RouterOS v7/v6

```
/ipv6 address add address=2001:db8:10::1/64 interface=ether1 advertise=yes
/ipv6 address add address=2001:db8:99::1/64 interface=ether2
/ipv6 route add dst-address=2001:db8:20::/64 gateway=2001:db8:99::2
```

R2 — RouterOS v7/v6

```
/ipv6 address add address=2001:db8:99::2/64 interface=ether1
/ipv6 address add address=2001:db8:20::1/64 interface=ether2 advertise=yes
/ipv6 route add dst-address=2001:db8:10::/64 gateway=2001:db8:99::1
```

VPC-A

```
PC1> ip 2001:db8:10::10/64 2001:db8:10::1
PC1> ping 2001:db8:20::10
```

VPC-B

```
PC2> ip 2001:db8:20::10/64 2001:db8:20::1
PC2> ping 2001:db8:10::10
```

Target hasil

- R1 dan R2 saling ping pada link IPv6.
- VPC-A bisa ping VPC-B.
- Routing IPv6 statik berjalan.

10.17 Firewall IPv6: Jangan Aktifkan Tanpa Proteksi

IPv6 harus dilindungi dengan firewall. Jangan berasumsi bahwa NAT akan “melindungi” client seperti di IPv4.

Prinsip IPv6 Firewall Input Chain

```
IPv6 traffic ke router
        |
        v
established/related → accept
invalid             → drop
ICMPv6              → accept
LAN                 → accept
lainnya             → drop
```

firewall IPv6 harus mengizinkan ICMPv6 agar fungsi dasar IPv6 tidak rusak.

Prinsip dasar firewall IPv6:

- izinkan koneksi `established,related`,
- drop `invalid`,
- izinkan ICMPv6,
- izinkan akses dari LAN,
- drop akses lain yang tidak perlu.

MikroTik menyediakan fitur firewall IPv6, dan dokumentasi firewall RouterOS menekankan bahwa firewall mengelola trafik ke, dari, dan melalui router. (MikroTik Help)

RouterOS v7/v6 — IPv6 Firewall Input Dasar

```
/ipv6 firewall filter add chain=input
connection-state=established,related action=accept
comment="Allow established related"
/ipv6 firewall filter add chain=input
connection-state=invalid action=drop comment="Drop invalid"
/ipv6 firewall filter add chain=input protocol=icmpv6
action=accept comment="Allow ICMPv6"
/ipv6 firewall filter add chain=input in-interface=ether2
action=accept comment="Allow LAN IPv6"
/ipv6 firewall filter add chain=input action=drop
comment="Drop other input"
```

Catatan:

> **Jangan blok ICMPv6 sembarangan.**
> **IPv6 membutuhkan ICMPv6 untuk fungsi penting seperti Neighbor Discovery.**

10.18 Verifikasi IPv6

RouterOS v7/v6

```
/ipv6 address print
/ipv6 route print
/ipv6 neighbor print
/ipv6 firewall filter print
/tool ping 2001:db8:10::10
```

VPC GNS3

```
PC1> show ipv6
PC1> ping 2001:db8:10::1
```

Ubuntu 26.04

```
ip -6 addr
ip -6 route
ping -6 2001:db8:10::1
```

Yang harus dicek:

- alamat IPv6 benar,
- prefix benar,
- gateway benar,
- route ada,
- firewall tidak memblokir ICMPv6.

10.19 Kesalahan Umum IPv6

Masalah	Penyebab Umum	Solusi
VPC tidak bisa ping MikroTik	IPv6/prefix salah	Cek `show ipv6`
Ubuntu tidak bisa ping gateway	interface salah	Cek `ip -6 addr`
Route IPv6 tidak jalan	route statik belum dibuat	Cek `/ipv6 route print`
Neighbor tidak muncul	link/topologi salah	Cek `/ipv6 neighbor print`
ICMPv6 gagal	firewall memblokir	Izinkan `protocol=icmpv6`
Dual-stack membingungkan	tidak jelas uji IPv4/IPv6	Pakai `ping` dan `ping -6` secara terpisah
Prefix asal-asalan	bentrok/kurang rapi	Gunakan `2001:db8::/32` untuk lab

10.20 Checklist Akhir Bab

Pembaca dianggap selesai Bab 10 jika mampu:

- Menjelaskan mengapa **IPv6 penting**.
- Membaca alamat seperti:

  ```
  2001:db8:10::1/64
  ```

- Menjelaskan:
  - prefix IPv6,
  - link-local address,
  - Neighbor Discovery,
  - SLAAC,
  - DHCPv6 dasar,
  - dual-stack.
- Mengatur IPv6 di RouterOS:

  ```
  /ipv6 address add address=2001:db8:10::1/64 interface=ether2
  advertise=yes
  ```

- Mengatur IPv6 di VPC GNS3:

  ```
  PC1> ip 2001:db8:10::10/64 2001:db8:10::1
  PC1> show ipv6
  PC1> ping 2001:db8:10::1
  ```

- Menguji IPv6 di Ubuntu:

  ```
  ip -6 addr
  ip -6 route
  ping -6 2001:db8:10::1
  ```

- Membuat firewall IPv6 dasar:

  ```
  /ipv6 firewall filter add chain=input
  connection-state=established,related action=accept
  /ipv6 firewall filter add chain=input
  connection-state=invalid action=drop
  /ipv6 firewall filter add chain=input protocol=icmpv6
  action=accept
  /ipv6 firewall filter add chain=input in-interface=ether2
  action=accept
  /ipv6 firewall filter add chain=input action=drop
  ```

- Membuat routing IPv6 sederhana antar router.
- Menjelaskan mengapa **ICMPv6 tidak boleh diblokir sembarangan**.

Ringkasan Bab

Bab ini membangun fondasi IPv6 secara praktis. Pembaca belajar bahwa IPv6 bukan sekadar alamat panjang, tetapi memiliki konsep penting seperti **prefix, link-local address, Neighbor Discovery, SLAAC, dual-stack, routing IPv6, dan firewall IPv6**.

Kunci Bab 10:

> **IPv6 harus dipelajari dengan praktik.**
> **Mulai dari satu router, satu client, satu prefix, satu ping, lalu naik ke routing dan firewall.**

Bab 11. Routing Lanjutan: Static Route, ECMP, Failover, dan OSPF

11.1 Tujuan Bab

Setelah menyelesaikan bab ini, pembaca mampu:

- Menjelaskan **mengapa routing diperlukan**.
- Membuat **static route**.
- Memahami **default route**.
- Membedakan **specific route** dan **default route**.
- Memahami **administrative distance**.
- Memahami fungsi **check-gateway**.
- Memahami **ECMP**.
- Membuat **failover sederhana**.
- Memahami dasar **OSPF**:
    - **neighbor**
    - **area**
    - **cost**
    - **network type**
    - **passive interface**
    - **single-area**
    - **multi-area** sebagai pengantar
- Membangun lab **3 CHR di GNS3**.
- Menghubungkan **VPC-A di LAN R1** ke **VPC-C di LAN R3** dengan:
    - static route
    - OSPF v7
    - sintaks OSPF v6 sebagai pembanding

11.2 Mengapa Routing Diperlukan

Routing dibutuhkan saat ada **lebih dari satu jaringan**. Jika semua host masih berada di satu subnet, cukup pakai switching atau bridge. Tetapi saat jaringan sudah dipisah menjadi beberapa subnet atau beberapa site, router harus tahu:

- paket ke jaringan mana,
- harus dikirim lewat gateway siapa,
- apakah ada jalur utama dan jalur cadangan.

Contoh sederhana:

- LAN R1: `192.168.10.0/24`
- LAN R3: `192.168.30.0/24`

Host di LAN R1 tidak bisa menjangkau LAN R3 hanya dengan satu IP gateway lokal. Router di tengah harus memiliki informasi route ke jaringan tujuan. Inilah inti routing. (MikroTik Help)

11.3 Static Route

Static route adalah rute yang ditulis manual oleh administrator. Ini paling mudah dipahami dan sangat cocok untuk lab awal.

Kelebihan static route:

- sederhana,
- jelas,
- mudah dilihat,
- cocok untuk topologi kecil.

Kekurangan:

- jika topologi membesar, konfigurasi bertambah banyak,
- perubahan jalur harus diubah manual,
- tidak otomatis belajar rute baru.

Contoh Static Route RouterOS v7/v6

```
/ip route add dst-address=192.168.30.0/24 gateway=10.10.12.2
/ip route add dst-address=0.0.0.0/0 gateway=192.168.1.1
```

Makna:

- untuk mencapai `192.168.30.0/24`, kirim ke `10.10.12.2`,
- untuk semua tujuan lain, kirim ke gateway default `192.168.1.1`.

Static route tetap menjadi fondasi penting bahkan saat nanti memakai protokol dinamis, karena administrator harus tetap paham arah trafik dan *next-hop* yang benar. (MikroTik Help)

11.4 Default Route vs Specific Route

Default Route

Default route adalah rute umum untuk semua tujuan yang tidak cocok dengan rute lain yang lebih spesifik.

Contoh:

```
/ip route add dst-address=0.0.0.0/0 gateway=192.168.1.1
```

Specific Route

Specific route adalah rute yang menunjuk ke jaringan tertentu.

Contoh:

```
/ip route add dst-address=192.168.30.0/24
gateway=10.10.12.2
```

Prinsip Penting

Router akan memilih rute yang **lebih spesifik** dibanding rute default.

Contoh:

- ke `192.168.30.10` → pakai rute `192.168.30.0/24`
- ke `8.8.8.8` → pakai default route `0.0.0.0/0`

Kalimat kunci:

> **Default route adalah “jalan umum”. Specific route adalah “jalan khusus”. Jalan khusus menang jika ada.**

11.5 Administrative Distance

Administrative distance dipakai untuk menentukan prioritas antar-route saat ada lebih dari satu jalur menuju tujuan yang sama.

Secara praktis:

- nilai **distance** lebih kecil → lebih diprioritaskan,
- nilai lebih besar → menjadi cadangan atau prioritas lebih rendah.

Contoh:

```
/ip route add dst-address=0.0.0.0/0 gateway=192.168.1.1
distance=1
/ip route add dst-address=0.0.0.0/0 gateway=192.168.2.1
distance=2
```

Makna:

- gateway `192.168.1.1` dipakai lebih dulu,
- gateway `192.168.2.1` menjadi jalur cadangan.

Konsep ini sangat penting untuk **failover**. (MikroTik Help)

11.6 Check-Gateway

check-gateway dipakai untuk membantu router mendeteksi apakah gateway masih aktif.

Contoh:

```
/ip route add gateway=192.168.1.1 distance=1
check-gateway=ping
/ip route add gateway=192.168.2.1 distance=2
```

MikroTik memberi contoh sederhana failover seperti di atas: route utama diberi `distance=1` dan `check-gateway=ping`, lalu route cadangan diberi `distance=2`. Jika gateway utama gagal dijangkau, route cadangan dapat dipakai. (MikroTik Help)

Catatan penting:

> **Tanpa check-gateway, router bisa tetap menganggap jalur utama aktif walaupun sebenarnya putus.**

11.7 ECMP

ECMP (*Equal Cost Multi Path*) adalah teknik memakai beberapa jalur dengan biaya yang sama menuju tujuan yang sama.

Makna praktis:

- ada dua atau lebih gateway aktif,
- router dapat membagi trafik di antara jalur-jalur itu,
- cocok untuk *load sharing* sederhana.

Contoh ide ECMP:

```
/ip route add dst-address=0.0.0.0/0
gateway=192.168.1.1,192.168.2.1
```

Atau pendekatan setara dengan dua route berbiaya sama, tergantung desain.

Catatan penting:

- ECMP **bukan** failover murni.
- ECMP berusaha memakai lebih dari satu jalur aktif.
- Failover memakai satu jalur utama dan satu jalur cadangan.

Secara konsep, **routing** dan **load balancing/failover** di RouterOS dapat digabungkan untuk mengatur pemakaian gateway yang berbeda sesuai kebutuhan. (MikroTik Help)

11.8 Failover Sederhana

Failover berarti saat jalur utama gagal, jalur cadangan otomatis dipakai.

Contoh paling sederhana:

```
/ip route add gateway=192.168.1.1 distance=1
check-gateway=ping
/ip route add gateway=192.168.2.1 distance=2
```

Cara berpikirnya:

- selama gateway 1 aktif → pakai jalur utama,
- jika gateway 1 gagal → pindah ke jalur cadangan.

MikroTik juga punya pembahasan failover yang lebih lanjut, termasuk pendekatan **recursive routing**. Namun untuk pemula, model **distance + check-gateway** sudah cukup untuk memahami konsep dasar. (MikroTik Help)

11.9 OSPF: Mengapa Perlu Routing Dinamis

Saat jaringan punya banyak router, static route mulai merepotkan. Di sinilah **OSPF** berguna.

OSPF adalah protokol routing dinamis yang memungkinkan router:

- menemukan tetangga,
- bertukar informasi jaringan,
- menghitung jalur terbaik,
- menyesuaikan route saat topologi berubah.

MikroTik menjelaskan OSPF sebagai *IGP* berbasis *link-state* yang dirancang untuk mendistribusikan informasi routing di dalam satu AS. (MikroTik Help)

Kapan OSPF Cocok

- ada beberapa router,
- topologi mulai membesar,
- ingin rute dipelajari otomatis,
- ingin perubahan topologi lebih mudah diadaptasi.

11.10 Istilah Penting OSPF

Istilah	Arti / Penjelasan
Area	Pembagian logis dalam OSPF. Untuk lab awal, gunakan area 0 atau backbone.
Cost	Nilai biaya jalur. Jalur dengan cost lebih kecil lebih disukai.
Multi-Area	Jaringan OSPF dibagi ke beberapa area. Untuk bab ini hanya dikenalkan sebagai konsep awal.
Neighbor	Router OSPF yang saling mengenali dan bertukar informasi.
Network Type	Tipe media atau topologi yang mempengaruhi perilaku OSPF.
Passive Interface	Interface yang subnet-nya tetap diiklankan ke OSPF, tetapi tidak membentuk neighbor OSPF pada interface tersebut. Cocok untuk interface LAN yang hanya berisi host, bukan router lain.
Single-Area	Semua router berada di area 0.

11.11 Topologi Lab Bab 11

Lab utama memakai **3 router CHR** dan **2 VPC**.

```
VPC-A --- R1 --- R2 --- R3 --- VPC-C
```

Tabel IP

Perangkat	Interface	IP
VPC-A	eth0	192.168.10.10/24
R1	LAN	192.168.10.1/24
R1	ke R2	10.10.12.1/30
R2	ke R1	10.10.12.2/30
R2	ke R3	10.10.23.1/30
R3	ke R2	10.10.23.2/30
R3	LAN	192.168.30.1/24
VPC-C	eth0	192.168.30.10/24

Tujuan Lab

- VPC-A di belakang R1 bisa ping VPC-C di belakang R3.
- Pertama dengan **static route**.
- Lalu dengan **OSPF v7**.
- OSPF v6 ditulis sebagai pembanding sintaks.

11.12 Konfigurasi Dasar IP

R1

```
/system identity set name=R1
/ip address add address=192.168.10.1/24 interface=ether1
comment="LAN-R1"
/ip address add address=10.10.12.1/30 interface=ether2
comment="R1-R2"
```

R2

```
/system identity set name=R2
/ip address add address=10.10.12.2/30 interface=ether1
comment="R2-R1"
/ip address add address=10.10.23.1/30 interface=ether2
comment="R2-R3"
```

R3

```
/system identity set name=R3
/ip address add address=10.10.23.2/30 interface=ether1
comment="R3-R2"
/ip address add address=192.168.30.1/24 interface=ether2
comment="LAN-R3"
```

VPC-A

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> ping 192.168.10.1
```

VPC-C

```
PC2> ip 192.168.30.10/24 192.168.30.1
PC2> ping 192.168.30.1
```

Langkah awal wajib: pastikan router yang bertetangga langsung bisa saling ping.

11.13 Lab 11.1 — Static Route Antar-LAN

R1

```
/ip route add dst-address=192.168.30.0/24 gateway=10.10.12.2
```

R2

```
/ip route add dst-address=192.168.10.0/24 gateway=10.10.12.1
/ip route add dst-address=192.168.30.0/24 gateway=10.10.23.2
```

R3

```
/ip route add dst-address=192.168.10.0/24 gateway=10.10.23.1
```

Verifikasi

Di R1:

```
/ip route print
/tool ping 10.10.12.2
/tool ping 192.168.30.1
```

Di R3:

```
/ip route print
/tool ping 10.10.23.1
/tool ping 192.168.10.1
```

Di VPC-A:

```
PC1> ping 192.168.30.10
```

Di VPC-C:

```
PC2> ping 192.168.10.10
```

Makna Lab

Dengan static route, semua router tahu jalan ke LAN lawan secara manual.

11.14 Lab 11.2 — OSPF v7 Single-Area

Untuk lab awal, semua router dimasukkan ke **area 0**.

R1 — OSPF v7

```
/routing ospf instance add name=ospf-v2 version=2
router-id=1.1.1.1
/routing ospf area add name=backbone area-id=0.0.0.0
instance=ospf-v2
/routing ospf interface-template add networks=10.10.12.0/30
area=backbone
/routing ospf interface-template add networks=192.168.10.0/24
area=backbone passive
```

R2 — OSPF v7

```
/routing ospf instance add name=ospf-v2 version=2
router-id=2.2.2.2
/routing ospf area add name=backbone area-id=0.0.0.0
instance=ospf-v2
/routing ospf interface-template add networks=10.10.12.0/30
area=backbone
/routing ospf interface-template add networks=10.10.23.0/30
area=backbone
```

R3 — OSPF v7

```
/routing ospf instance add name=ospf-v2 version=2
router-id=3.3.3.3
/routing ospf area add name=backbone area-id=0.0.0.0
instance=ospf-v2
/routing ospf interface-template add networks=10.10.23.0/30
area=backbone
/routing ospf interface-template add networks=192.168.30.0/24
area=backbone passive
```

Verifikasi OSPF v7

```
/routing ospf neighbor print
/ip route print
```

Yang harus terlihat:

- R1 bertetangga dengan R2
- R2 bertetangga dengan R1 dan R3
- R3 bertetangga dengan R2
- route OSPF muncul otomatis

Setelah itu, uji dari VPC-A ke VPC-C.

OSPF v7 di RouterOS memang memakai pendekatan **instance**, **area**, dan **interface-template**. Ini berbeda dari gaya lama pada RouterOS v6. (MikroTik Help)

11.15 OSPF v6 sebagai Pembanding

Di RouterOS v6, sintaks OSPF lebih sederhana secara tampilan, biasanya memakai:

- set **router-id** pada instance default
- tambahkan **network** ke area tertentu

Contoh OSPF RouterOS v6 — R1

```
/routing ospf instance set default router-id=1.1.1.1
/routing ospf network add network=10.10.12.0/30 area=backbone
/routing ospf network add network=192.168.10.0/24
area=backbone
```

Contoh OSPF RouterOS v6 — R2

```
/routing ospf instance set default router-id=2.2.2.2
/routing ospf network add network=10.10.12.0/30 area=backbone
/routing ospf network add network=10.10.23.0/30 area=backbone
```

Contoh OSPF RouterOS v6 — R3

```
/routing ospf instance set default router-id=3.3.3.3
/routing ospf network add network=10.10.23.0/30 area=backbone
/routing ospf network add network=192.168.30.0/24
area=backbone
```

Catatan penting:

> **RouterOS v7 adalah fokus utama buku ini.**
> **Sintaks v6 ditulis sebagai pembanding**, karena banyak admin masih menemui perangkat v6 di lapangan.

11.16 ECMP Dua Gateway

Untuk memahami **ECMP**, bayangkan R1 punya dua jalur setara ke tujuan tertentu.

Contoh ide sederhana:

```
/ip route add dst-address=0.0.0.0/0
gateway=10.10.12.2,10.10.13.2
```

Atau dua route setara yang menghasilkan perilaku jalur ganda, sesuai desain.

Makna:

- ada dua gateway dengan biaya setara,
- trafik dapat dibagi di antara keduanya.

Catatan kritis:

- ECMP cocok untuk **lebih dari satu jalur aktif**.
- Jika target utama adalah **cadangan otomatis**, pakai **failover**, bukan ECMP.

11.17 Failover dengan Check-Gateway

Contoh sederhana:

```
/ip route add gateway=10.10.12.2 distance=1
check-gateway=ping
/ip route add gateway=10.10.13.2 distance=2
```

Makna:

- route utama lewat `10.10.12.2`,
- route cadangan lewat `10.10.13.2`,
- jika gateway utama gagal dijangkau, jalur kedua dipakai.

Ini adalah pola paling mudah untuk dipahami sebelum masuk ke failover yang lebih kompleks. (MikroTik Help)

11.18 Passive Interface

Di OSPF, interface LAN yang tidak menghadap router lain sebaiknya dijadikan **passive**.

Manfaatnya:

- subnet LAN tetap diiklankan,
- tetapi router tidak mencoba membentuk neighbor ke host biasa,
- lebih rapi dan lebih aman.

Contoh pada OSPF v7:

```
/routing ospf interface-template add networks=192.168.10.0/24
area=backbone passive
```

Contoh ini sangat tepat untuk LAN R1 dan LAN R3 pada lab. (MikroTik Help)

11.19 Single-Area dan Multi-Area

Single-Area

Semua router berada pada **area 0**. Ini model terbaik untuk pemula.

Kelebihan:

- mudah dipahami,
- mudah diuji,
- cocok untuk lab 3 router.

Multi-Area

Jaringan dibagi ke beberapa area.

Dipakai ketika:

- jaringan lebih besar,
- ingin struktur OSPF lebih rapi,
- ingin membatasi lingkup LSDB dan kompleksitas tertentu.

Untuk bab ini, **multi-area cukup dikenalkan sebagai pengantar**. Fokus praktik tetap **single-area area 0**.

11.20 Troubleshooting Routing dan OSPF

1. Neighbor OSPF Tidak Muncul

Cek:

```
/routing ospf neighbor print
/ip address print
```

Penyebab umum:

- IP link antar-router salah,
- interface-template atau network salah,
- router belum saling ping,
- area tidak cocok.

2. Route OSPF Tidak Muncul

Cek:

```
/ip route print
```

Penyebab umum:

- subnet tidak diiklankan,
- OSPF neighbor belum terbentuk,
- network LAN tidak dimasukkan,
- passive interface salah penempatan.

3. Static Route Salah Gateway

Contoh salah:

```
/ip route add dst-address=192.168.30.0/24
gateway=10.10.23.2
```

di R1, padahal R1 tidak langsung bertetangga dengan `10.10.23.2`.

Yang benar untuk R1:

```
/ip route add dst-address=192.168.30.0/24
gateway=10.10.12.2
```

4. VPC Tidak Bisa Ping Jaringan Lawan

Cek:

- IP VPC benar,
- gateway VPC benar,
- route di router ada,
- firewall forward tidak memblokir.

VPCS hanya mendukung pengujian sederhana seperti **DHCP** dan **ping**, sehingga sangat baik untuk validasi konektivitas dasar. (GNS3 Documentation)

11.21 Verifikasi Akhir

RouterOS

```
/ip address print
/ip route print
/routing ospf neighbor print
/tool ping 192.168.30.1
```

VPC-A

```
PC1> show ip
PC1> ping 192.168.30.10
PC1> trace 192.168.30.10
```

VPC-C

```
PC2> show ip
PC2> ping 192.168.10.10
PC2> trace 192.168.10.10
```

Yang harus tercapai:

- static route bekerja,
- OSPF neighbor terbentuk,
- route OSPF muncul,
- VPC-A bisa ping VPC-C,
- VPC-C bisa ping VPC-A.

11.22 Checklist Akhir Bab

Pembaca dianggap selesai Bab 11 jika mampu:

- Menjelaskan mengapa routing diperlukan.
- Membuat **static route**:

```
/ip route add dst-address=192.168.30.0/24 gateway=10.10.12.2
```

- Membuat **default route**:

```
/ip route add dst-address=0.0.0.0/0 gateway=192.168.1.1
```

- Menjelaskan beda **specific route** dan **default route**.
- Menjelaskan fungsi **administrative distance**.
- Menjelaskan fungsi **check-gateway**.
- Menjelaskan beda **ECMP** dan **failover**.
- Membuat OSPF v7 dasar:

```
/routing ospf instance add name=ospf-v2 version=2
router-id=1.1.1.1
/routing ospf area add name=backbone area-id=0.0.0.0
instance=ospf-v2
/routing ospf interface-template add networks=10.10.12.0/30
area=backbone
/routing ospf interface-template add networks=192.168.10.0/24
area=backbone passive
```

- Menjelaskan sintaks OSPF v6 sebagai pembanding.
- Menguji koneksi **VPC-A ↔ VPC-C** di GNS3.

Ringkasan Bab

Bab ini mengajarkan fondasi **routing lanjutan**: mulai dari **static route**, lanjut ke **default route**, **administrative distance**, **check-gateway**, **ECMP**, **failover**, lalu masuk ke **OSPF** sebagai protokol routing dinamis.

Kunci Bab 11:

> **Kalau static route melatih logika jalur, maka OSPF melatih logika jaringan. Pahami dulu jalur manual, baru percaya pada jalur otomatis.**

Bab 12. BGP untuk Pemula: Simulasi ISP Kecil dan Customer

12.1 Tujuan Bab

Setelah menyelesaikan bab ini, pembaca mampu:

- Menjelaskan apa itu **BGP**.
- Menjelaskan kapan BGP dibutuhkan.
- Memahami:
    - **ASN**
    - **prefix**
    - **peer**
    - **eBGP**
    - **iBGP**
- Menjelaskan mengapa **BGP bukan pengganti OSPF**.
- Membuat simulasi sederhana:
    - **Customer**
    - **ISP-A**
    - **ISP-B**
- Melakukan **route advertisement**.
- Membuat **route filtering** dasar.
- Memberi **default route dari ISP ke customer**.
- Memahami perbedaan konfigurasi BGP RouterOS v7 dan v6.
- Menjalankan etika dasar BGP:
    - jangan advertise prefix sembarangan,
    - filter wajib,
    - gunakan prefix lab/private untuk simulasi.

12.2 Apa Itu BGP

BGP atau *Border Gateway Protocol* adalah protokol routing untuk bertukar informasi rute antarjaringan besar yang berbeda administrasi.

Dalam bahasa sederhana:

> **OSPF dipakai di dalam jaringan sendiri.**
> **BGP dipakai untuk berbicara dengan jaringan lain.**

Contoh:

```
Customer ↔ ISP
ISP-A ↔ ISP-B
Kampus ↔ Provider
Perusahaan ↔ Dua Provider
```

BGP tidak hanya memilih “jalur tercepat”. BGP juga membawa kebijakan:

- prefix mana yang boleh diumumkan,
- prefix mana yang boleh diterima,
- peer mana yang dipercaya,
- jalur mana yang lebih dipilih,
- route mana yang harus ditolak.

Karena itu, BGP harus diajarkan bersama **filter** sejak awal. RouterOS v7 menyediakan *routing filter rule* dengan sintaks seperti skrip untuk mencocokkan prefix dan mengambil tindakan seperti `accept` atau `reject`. (MikroTik Help)

12.3 Kapan BGP Dibutuhkan

BGP dibutuhkan jika jaringan:

- memiliki **ASN sendiri**,
- memiliki koneksi ke satu atau lebih ISP,
- ingin mengumumkan prefix ke jaringan lain,
- perlu menerima route dari jaringan lain,
- perlu memilih jalur berdasarkan kebijakan,
- melakukan *multi-homing* ke dua provider.

BGP **tidak selalu dibutuhkan** untuk:

- rumah,
- lab kecil,
- kantor kecil satu ISP,
- jaringan yang cukup dengan default route,
- jaringan internal yang lebih cocok memakai OSPF atau static route.

Kalimat penting:

> **Jangan memakai BGP hanya karena terlihat keren.**
> **Pakai BGP jika memang ada kebutuhan bertukar route antar-AS.**

12.4 ASN

ASN atau *Autonomous System Number* adalah nomor identitas jaringan dalam BGP.

Dalam lab, kita bisa memakai ASN private untuk simulasi.

Contoh:

Peran	ASN
Customer	`65001`
ISP-A	`65002`
ISP-B	`65003`

ASN di atas dipakai untuk lab. Jangan dipakai sembarangan untuk jaringan publik nyata.

Prinsip:

- satu AS = satu domain administrasi routing,
- peer BGP memakai ASN untuk mengenali lawan,
- eBGP biasanya terjadi antar-AS berbeda.

12.5 Prefix

Prefix adalah jaringan yang diumumkan ke BGP.

Contoh:

```
192.168.10.0/24
```

Dalam lab ini, Customer akan mengumumkan prefix:

```
192.168.10.0/24
```

Namun ada catatan keras:

> **Di jaringan nyata, jangan advertise prefix yang bukan milik kita.**
> **Di lab, gunakan prefix private/dokumentasi agar aman.**

Untuk simulasi lokal GNS3, prefix private seperti `192.168.10.0/24` cukup aman karena tidak keluar ke Internet nyata.

12.6 Peer

Peer adalah router BGP lawan bicara.

Contoh:

```
Customer R1 ↔ ISP-A R2
Customer R1 ↔ ISP-B R3
```

Peer BGP harus tahu:

- alamat IP lawan,
- ASN lawan,
- ASN sendiri,
- role,
- route apa yang boleh dikirim,
- route apa yang boleh diterima.

Di RouterOS v7, konfigurasi BGP memakai konsep seperti **template**, **connection**, **role**, dan parameter remote. Dokumentasi RouterOS v7 BGP menjelaskan properti seperti router ID, routing table, VRF, dan koneksi BGP. (MikroTik Help)

12.7 eBGP vs iBGP

eBGP

eBGP terjadi antar-AS berbeda.

Contoh:

```
Customer AS65001 ↔ ISP-A AS65002
```

iBGP

iBGP terjadi di dalam AS yang sama.

Contoh:

```
Router-A AS65001 ↔ Router-B AS65001
```

Untuk bab pemula ini, fokus pada **eBGP**, karena lebih mudah dipahami dengan model Customer–ISP.

12.8 BGP Bukan Pengganti OSPF

OSPF Internal vs BGP Eksternal

```
Di dalam jaringan Customer:
R1 --- R2 --- R3  memakai OSPF

Keluar ke ISP:
Customer --- ISP  memakai BGP
```

OSPF dan BGP punya peran berbeda, bukan saling menggantikan.

Ini sangat penting.

> **OSPF** cocok untuk jaringan internal.
> **BGP** cocok untuk hubungan antar-AS.

Tabel ringkas:

Aspek	OSPF	BGP
Lingkup	internal	antar-AS
Jenis	IGP	EGP
Fokus	jalur internal	kebijakan dan pertukaran prefix
Cocok untuk	jaringan kampus/kantor	customer–ISP, antarprovider
Bab utama	Bab 11	Bab 12

Prinsip:

> **OSPF menyusun jalan di dalam rumah.**
> **BGP mengatur pintu keluar dan hubungan dengan tetangga besar.**

12.9 Route Advertisement

Route advertisement adalah proses mengumumkan prefix ke peer BGP.

Contoh:

- Customer mengumumkan `192.168.10.0/24` ke ISP-A.
- ISP-A menerima prefix itu.
- ISP-A tahu bahwa jaringan `192.168.10.0/24` berada di belakang Customer.

Namun prefix sebaiknya tidak diumumkan sembarangan. Di RouterOS v7, pendekatan yang aman adalah:

- buat route untuk prefix,
- buat filter output,
- hanya accept prefix yang benar,
- reject sisanya.

Contoh pendekatan:

```
/ip route add dst-address=192.168.10.0/24 blackhole
/routing filter rule add chain=bgp-out rule="if (dst ==
192.168.10.0/24) { accept }"
/routing filter rule add chain=bgp-out rule="reject"
```

Route **blackhole** dipakai agar prefix yang diumumkan benar-benar ada di routing table, sekaligus mencegah perilaku tidak jelas jika trafik menuju prefix tersebut masuk ke router tetapi tidak ada host spesifik.

12.10 Route Filtering

Route filtering wajib dalam BGP. Tanpa filter, router bisa:

- mengirim prefix yang salah,
- menerima route yang tidak diinginkan,
- membuat routing bocor,
- mengganggu jaringan lain.

MikroTik mendokumentasikan *routing filter rule* dengan sintaks yang dapat mencocokkan prefix dan menjalankan aksi seperti mengubah atribut, `accept`, atau `reject`. (MikroTik Help)

Prinsip pemula:

> **BGP tanpa filter adalah undangan masalah.**
> **Filter bukan tambahan. Filter adalah bagian inti konfigurasi BGP.**

12.11 Default Route dari ISP ke Customer

Dalam banyak kasus, customer kecil tidak perlu menerima seluruh route Internet. Cukup menerima **default route** dari ISP.

Contoh:

```
Customer tidak perlu tahu semua rute.
Customer cukup tahu:
0.0.0.0/0 lewat ISP
```

Di lab, ISP dapat mengirim default route ke Customer. Customer mengirim prefix LAN miliknya ke ISP.

Model sederhana:

- Customer → advertise `192.168.10.0/24`
- ISP → advertise `0.0.0.0/0`

Untuk bab ini, fokus utama tetap pada customer mengumumkan prefix ke ISP, lalu memahami konsep default route sebagai materi lanjutan sederhana.

12.12 Topologi Lab 1: Customer ↔ ISP-A

Topologi

```
Customer-AS65001 ---- ISP-A-AS65002
```

Tabel IP

Perangkat	Interface	IP	ASN
Customer R1	LAN	`192.168.10.1/24`	65001
Customer R1	ke ISP-A	`10.10.12.1/30`	65001
ISP-A R2	ke Customer	`10.10.12.2/30`	65002
VPC-CUST	eth0	`192.168.10.10/24`	-

VPC Customer

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> ping 192.168.10.1
```

Target awal:

- Customer bisa ping ISP-A.
- VPC bisa ping router Customer.
- Setelah BGP aktif, ISP-A belajar prefix Customer.

12.13 Konfigurasi Dasar IP

Customer R1

```
/system identity set name=R-CUST
/ip address add address=192.168.10.1/24 interface=ether1
comment="LAN Customer"
/ip address add address=10.10.12.1/30 interface=ether2
comment="to ISP-A"
```

ISP-A R2

```
/system identity set name=R-ISP-A
/ip address add address=10.10.12.2/30 interface=ether1
comment="to Customer"
```

Verifikasi
Di Customer:

```
/tool ping 10.10.12.2
```

Di ISP-A:

```
/tool ping 10.10.12.1
```

Jangan lanjut ke BGP jika link IP dasar belum berhasil.

12.14 BGP RouterOS v7 — Customer

```
/routing bgp template add name=tpl-as65001 as=65001
router-id=1.1.1.1
/routing bgp connection add name=to-isp
remote.address=10.10.12.2 remote.as=65002
templates=tpl-as65001 local.role=ebgp
/ip route add dst-address=192.168.10.0/24 blackhole
/routing filter rule add chain=bgp-out rule="if (dst ==
192.168.10.0/24) { accept }"
/routing filter rule add chain=bgp-out rule="reject"
/routing bgp connection set to-isp
output.filter-chain=bgp-out
```

Makna:

- template menyimpan ASN dan router-id Customer,
- connection membuat peer ke ISP-A,
- blackhole route memastikan prefix tersedia,
- filter hanya mengizinkan prefix Customer,
- selain itu ditolak.

RouterOS v7 memang membawa perubahan besar pada konfigurasi routing, termasuk pendekatan BGP template, connection, role, address families, dan filter input/output. (MikroTik Help)

12.15 BGP RouterOS v7 — ISP-A

```
/routing bgp template add name=tpl-as65002 as=65002
router-id=2.2.2.2
/routing bgp connection add name=to-customer
remote.address=10.10.12.1 remote.as=65001
templates=tpl-as65002 local.role=ebgp
```

Verifikasi di ISP-A:

```
/routing bgp session print
/ip route print
```

Yang dicari:

- session BGP terbentuk,
- route `192.168.10.0/24` diterima dari Customer.

12.16 BGP RouterOS v6 sebagai Pembanding

Di RouterOS v6, konfigurasi lebih bergaya lama:

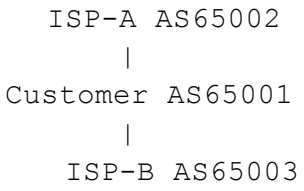
```
/routing bgp instance set default as=65001 router-id=1.1.1.1
/routing bgp peer add name=to-isp remote-address=10.10.12.2
remote-as=65002
/routing bgp network add network=192.168.10.0/24
```

Catatan:

- RouterOS v7 adalah fokus utama buku ini.
- RouterOS v6 ditulis sebagai pembanding karena masih banyak perangkat lama di lapangan.
- Cara filtering di v6 berbeda dari v7, sehingga jangan mencampur sintaks tanpa memahami versi.

12.17 Topologi Lab 2: Dua ISP dan Satu Customer

Topologi



Tabel IP

Link	IP Customer	IP ISP
Customer–ISP-A	10.10.12.1/30	10.10.12.2/30
Customer–ISP-B	10.10.13.1/30	10.10.13.2/30
LAN Customer	192.168.10.1/24	-

Tujuan

- Customer punya dua peer eBGP.
- Customer mengumumkan prefix yang sama ke ISP-A dan ISP-B.
- Customer belajar bahwa BGP bisa dipakai untuk *multi-homing* sederhana.

12.18 BGP v7 — Customer ke ISP-A dan ISP-B

```
/routing bgp template add name=tpl-as65001 as=65001 router-id=1.1.1.1

/routing bgp connection add name=to-isp-a remote.address=10.10.12.2
remote.as=65002 templates=tpl-as65001 local.role=ebgp
/routing bgp connection add name=to-isp-b remote.address=10.10.13.2
remote.as=65003 templates=tpl-as65001 local.role=ebgp

/ip route add dst-address=192.168.10.0/24 blackhole

/routing filter rule add chain=bgp-out rule="if (dst ==
192.168.10.0/24) { accept }"
/routing filter rule add chain=bgp-out rule="reject"

/routing bgp connection set to-isp-a output.filter-chain=bgp-out
/routing bgp connection set to-isp-b output.filter-chain=bgp-out
```

Verifikasi:

```
/routing bgp session print
/ip route print
```

12.19 Filter Mencegah Route Bocor

Route bocor terjadi ketika router mengumumkan prefix yang tidak semestinya.

Contoh berbahaya:

- customer mengumumkan semua route,
- customer meneruskan route dari satu ISP ke ISP lain tanpa kebijakan,
- customer mengiklankan prefix yang bukan miliknya.

Untuk pemula, filter paling sederhana:

```
/routing filter rule add chain=bgp-out rule="if (dst ==
192.168.10.0/24) { accept }"
/routing filter rule add chain=bgp-out rule="reject"
```

Makna:

- hanya prefix `192.168.10.0/24` yang boleh keluar,
- semua selain itu ditolak.

Kunci:

> **Filter BGP harus dibuat sebelum jaringan dianggap aman.**
> **Jangan menunggu route bocor baru belajar filter.**

12.20 Default Route dari ISP

Contoh konsep di ISP-A:

```
/ip route add dst-address=0.0.0.0/0 blackhole
/routing filter rule add chain=isp-out rule="if (dst ==
0.0.0.0/0) { accept }"
/routing filter rule add chain=isp-out rule="reject"
/routing bgp connection set to-customer
output.filter-chain=isp-out
```

Catatan:

- Ini contoh lab.
- Di jaringan nyata, default route harus disesuaikan dengan kebijakan upstream.
- Pastikan customer tidak menerima route yang tidak diperlukan.

Untuk pemula, cukup pahami:

> **Customer kecil biasanya cukup menerima default route, bukan seluruh tabel Internet.**

12.21 Verifikasi BGP

RouterOS v7

```
/routing bgp session print
/routing bgp connection print
/ip route print
/routing filter rule print
```

Customer

```
/tool ping 10.10.12.2
/tool ping 10.10.13.2
/ip route print
```

VPC Customer

```
PC1> show ip
PC1> ping 192.168.10.1
```

Jika ada server simulasi di sisi ISP, uji ping ke jaringan itu setelah route tersedia.

12.22 Troubleshooting BGP

1. Session Tidak Terbentuk

Cek:

```
/routing bgp session print
/tool ping 10.10.12.2
```

Penyebab umum:

- IP link belum bisa ping,
- remote.as salah,
- remote.address salah,
- firewall memblokir TCP 179,
- router-id bentrok.

BGP memakai TCP untuk membangun sesi, sehingga konektivitas IP dasar harus beres lebih dulu.

2. Session Up, Route Tidak Muncul

Cek:

```
/ip route print
/routing filter rule print
```

Penyebab umum:

- prefix belum ada di routing table,
- blackhole route belum dibuat,
- filter menolak prefix,
- output filter belum dipasang ke connection.

3. Prefix Salah Teradvertise

Cek:

```
/routing filter rule print
/routing bgp advertisements print
```

Solusi:

- buat filter ketat,
- accept hanya prefix yang benar,
- reject sisanya.

4. Salah ASN

Gejala:

- peer tidak terbentuk,
- log menunjukkan mismatch.

Cek:

```
/routing bgp connection print detail
```

Pastikan:

- ASN sendiri benar,
- ASN remote benar.

12.23 Etika BGP

BGP harus diajarkan dengan etika sejak awal.

Prinsip penting:

- **Jangan advertise prefix yang bukan milik kita.**
- **Jangan menerima semua route tanpa filter.**
- **Jangan mengirim semua route tanpa filter.**
- **Jangan melakukan eksperimen BGP ke jaringan nyata tanpa izin.**
- **Gunakan prefix private/lab untuk simulasi.**
- **Dokumentasikan ASN, peer, prefix, dan filter.**

Kalimat penting:

> **BGP adalah kepercayaan.**
> **Salah konfigurasi kecil bisa berdampak besar jika masuk jaringan nyata.**

12.24 Checklist Akhir Bab

Pembaca dianggap selesai Bab 12 jika mampu:

- Menjelaskan apa itu **BGP**.
- Menjelaskan kapan BGP dibutuhkan.
- Menjelaskan:
    - ASN,
    - prefix,
    - peer,
    - eBGP,
    - iBGP.
- Menjelaskan mengapa BGP bukan pengganti OSPF.
- Membuat koneksi eBGP sederhana Customer ↔ ISP.
- Membuat konfigurasi BGP RouterOS v7:

```
/routing bgp template add name=tpl-as65001 as=65001
router-id=1.1.1.1
/routing bgp connection add name=to-isp
remote.address=10.10.12.2 remote.as=65002
templates=tpl-as65001 local.role=ebgp
/ip route add dst-address=192.168.10.0/24 blackhole
/routing filter rule add chain=bgp-out rule="if (dst ==
192.168.10.0/24) { accept }"
/routing filter rule add chain=bgp-out rule="reject"
/routing bgp connection set to-isp
output.filter-chain=bgp-out
```

- Menjelaskan konfigurasi RouterOS v6 sebagai pembanding:

```
/routing bgp instance set default as=65001 router-id=1.1.1.1
/routing bgp peer add name=to-isp remote-address=10.10.12.2
remote-as=65002
/routing bgp network add network=192.168.10.0/24
```

- Memahami fungsi **route filtering**.
- Menjelaskan konsep **default route dari ISP ke customer**.
- Menjalankan simulasi dua ISP dan satu customer.
- Menjelaskan etika BGP.

Ringkasan Bab

Bab ini mengenalkan **BGP** dengan cara sederhana: mulai dari **ASN**, **prefix**, **peer**, **eBGP**, lalu masuk ke **route advertisement** dan **filtering**. BGP harus dipahami sebagai protokol pertukaran route antar-AS yang sangat kuat, tetapi juga berisiko jika salah konfigurasi.

Kunci Bab 12:

> **Di BGP, jangan hanya bertanya “route masuk atau tidak”.**
> **Tanyakan juga: route ini boleh masuk? route ini boleh keluar? prefix ini benar milik siapa?**

Bab 13. Troubleshooting dan Proyek Mini Akhir

13.1 Tujuan Bab

Setelah menyelesaikan bab ini, pembaca mampu:

- Memiliki pola pikir troubleshooting yang rapi.
- Tidak menebak penyebab masalah.
- Mengecek masalah dari layer bawah.
- Mengubah satu hal dalam satu waktu.
- Mencatat kondisi sebelum dan sesudah.
- Menggunakan tools RouterOS:
    - `/tool ping`
    - `/tool traceroute`
    - `/tool torch`
    - `/tool sniffer`
    - `/log print`
    - `/ip route print`
    - `/ip firewall connection print`
    - `/interface monitor-traffic`
- Menggunakan tools Ubuntu:
    - `ip addr`
    - `ip route`
    - `ping`
    - `traceroute`
    - `dig`
    - `curl`
    - `tcpdump`
- Menggunakan tools VPC GNS3:
    - `show ip`
    - `show ipv6`
    - `ping`
    - `trace`
    - `ip dhcp`
- Menyelesaikan proyek mini akhir sebagai **portofolio praktik**.

13.2 Prinsip Troubleshooting

Troubleshooting bukan menebak. Troubleshooting adalah proses mencari bukti.

Gunakan empat prinsip utama:

- **Jangan menebak.**
- **Cek dari layer bawah.**
- **Ubah satu hal dalam satu waktu.**
- **Catat sebelum dan sesudah.**

Pola sederhana:

```
Gejala → Dugaan → Uji → Bukti → Perbaikan → Uji ulang → Catat
```

Kesalahan pemula biasanya:

- langsung mengubah firewall tanpa cek IP,
- langsung menghapus NAT tanpa cek route,
- langsung reset router tanpa export,
- memperbaiki tiga hal sekaligus sehingga tidak tahu mana yang benar-benar menyelesaikan masalah.

Kalimat kunci:

> **Troubleshooting yang baik bukan yang cepat panik, tetapi yang disiplin membaca gejala.**

13.3 Urutan Cek Dasar

Gunakan urutan ini untuk hampir semua masalah jaringan:

1. **Cek interface**
    - apakah interface hidup?
    - apakah link benar?
    - apakah kabel/topologi GNS3 benar?
2. **Cek IP**
    - apakah IP ada?
    - apakah prefix benar?
    - apakah gateway benar?
3. **Cek route**
    - apakah route ke tujuan ada?
    - apakah default route ada?
4. **Cek DNS**
    - apakah ping IP berhasil tetapi nama gagal?
5. **Cek NAT**
    - apakah client private butuh NAT?
    - apakah NAT rule mengarah ke interface keluar?
6. **Cek firewall**
    - apakah input atau forward memblokir?
7. **Cek log**
    - apakah ada pesan drop, error, atau service gagal?

RouterOS packet flow menjelaskan bahwa paket masuk dari *in-interface*, lalu router menentukan apakah paket perlu dirutekan, melewati proses routing, dan keluar melalui *out-interface*. Pemahaman alur ini penting agar pembaca tahu apakah masalah berada di input, forward, routing, NAT, atau firewall. (MikroTik Help)

13.4 Tools RouterOS

Ping

```
/tool ping 192.168.10.10
/tool ping 1.1.1.1
/tool ping 2001:db8:10::10
```

Fungsi:

- uji konektivitas dasar,
- uji gateway,
- uji IP luar,
- uji IPv6.

Traceroute

```
/tool traceroute 1.1.1.1
/tool traceroute 192.168.30.10
```

Fungsi:

- melihat jalur paket,
- mengetahui di hop mana paket berhenti.

Torch

```
/tool torch interface=ether2
```

Fungsi:

- melihat trafik *real-time* pada interface,
- mengetahui apakah paket masuk/keluar,
- membantu saat NAT/firewall dicurigai bermasalah.

MikroTik menjelaskan bahwa Torch memonitor aliran trafik pada interface secara *real-time*. Trafik yang muncul di Torch terlihat sebelum difilter oleh firewall, sehingga membantu melihat paket yang mungkin akhirnya di-drop oleh rule firewall. (MikroTik Help)

Packet Sniffer

```
/tool sniffer set filter-interface=ether2
/tool sniffer start
/tool sniffer stop
/tool sniffer packet print
```

Fungsi:

- menangkap paket,
- melihat apakah paket benar-benar lewat,
- berguna untuk kasus sulit.

MikroTik menjelaskan *packet sniffer* sebagai alat untuk menangkap dan menganalisa paket yang menuju, keluar, atau melewati router. (MikroTik Help)

Log

```
/log print
```

Fungsi:

- membaca pesan sistem,
- melihat error,
- melihat drop jika firewall logging aktif,
- membantu diagnosis service.

Route

```
/ip route print
/ipv6 route print
```

Fungsi:

- melihat default route,
- melihat route static,
- melihat route OSPF/BGP,
- melihat route aktif/tidak aktif.

Firewall Connection

```
/ip firewall connection print
```

Fungsi:

- melihat koneksi aktif,
- memeriksa apakah NAT/connection tracking bekerja,
- melihat sumber dan tujuan koneksi.

Monitor Traffic

```
/interface monitor-traffic ether1
/interface monitor-traffic ether2
```

Fungsi:

- melihat trafik TX/RX,
- memastikan interface benar-benar dilalui paket.

13.5 Tools Ubuntu

Ubuntu VM dipakai untuk pengujian lebih lengkap dibanding VPC.

Cek IP

```
ip addr
ip -6 addr
```

Cek Route

```
ip route
ip -6 route
```

Ping

```
ping 192.168.10.1
ping 1.1.1.1
ping -6 2001:db8:10::1
```

Traceroute

```
traceroute 1.1.1.1
traceroute 192.168.30.10
```

DNS

```
dig example.net
```

HTTP Test

```
curl http://192.168.10.20
curl http://IP-WAN-ROUTER:8080
```

Packet Capture

```
sudo tcpdump -i enp0s3
sudo tcpdump -i enp0s3 icmp
```

Gunakan Ubuntu untuk kasus:

- uji web server,
- uji DNS,
- uji port forwarding,
- uji tcpdump,
- uji koneksi yang membutuhkan aplikasi nyata.

13.6 Tools VPC GNS3

VPC GNS3 sangat ringan dan cocok untuk uji cepat. VPCS adalah *Virtual PC Simulator* yang mendukung DHCP dan ping, memakai sekitar 2 MB RAM per instance, dan tidak membutuhkan image tambahan. (GNS3 Documentation)

Perintah utama:

```
show ip
show ipv6
ip dhcp
ping 192.168.10.1
trace 1.1.1.1
```

Untuk IPv6:

```
show ipv6
ping 2001:db8:10::1
```

Gunakan VPC untuk:

- uji IP,
- uji DHCP,
- uji gateway,
- uji routing,
- uji koneksi antar-LAN.

Catatan:

> **VPC cepat untuk uji konektivitas.**
> **Ubuntu VM lebih kuat untuk uji layanan nyata.**

13.7 Kasus Nyata 1 — VPC Tidak Dapat IP

Gejala:

```
PC1> ip dhcp
```

tetapi VPC tidak mendapat alamat.

Cek di RouterOS:

```
/ip dhcp-server print
/ip dhcp-server network print
/ip pool print
/ip address print
/interface print
```

Cek di GNS3:

- apakah VPC tersambung ke interface yang benar?
- apakah link aktif?
- apakah interface LAN MikroTik benar?

Penyebab umum:

- DHCP Server belum aktif.
- DHCP Server dipasang di interface salah.
- IP gateway belum ada pada interface LAN.
- Pool salah.
- VPC tersambung ke link yang salah.

Solusi cepat:

```
/ip address add address=192.168.10.1/24 interface=ether2
/ip pool add name=pool-lan
ranges=192.168.10.100-192.168.10.200
/ip dhcp-server add name=dhcp-lan interface=ether2
address-pool=pool-lan disabled=no
/ip dhcp-server network add address=192.168.10.0/24
gateway=192.168.10.1 dns-server=192.168.10.1
```

13.8 Kasus Nyata 2 — VPC Bisa Ping Gateway, Tetapi Tidak Bisa Keluar

Gejala:

```
PC1> ping 192.168.10.1
```

berhasil, tetapi:

```
PC1> ping 1.1.1.1
```

gagal.

Cek RouterOS:

```
/ip route print
/ip firewall nat print
/ip firewall filter print
/ip firewall connection print
/tool ping 1.1.1.1
```

Kemungkinan:

- router tidak punya default route,
- NAT belum dibuat,
- NAT salah `out-interface`,
- firewall forward memblokir.

Rule NAT dasar:

```
/ip firewall nat add chain=srcnat out-interface=ether1
action=masquerade comment="NAT LAN to WAN"
```

Catatan:

> **Jika router sendiri tidak bisa ping keluar, jangan salahkan NAT. Perbaiki route/WAN dulu.**

13.9 Kasus Nyata 3 — Ubuntu VM Tidak Tersambung ke GNS3

Gejala:

- Ubuntu tidak bisa ping router,
- router tidak bisa ping Ubuntu,
- VPC bisa jalan, tetapi Ubuntu tidak.

Cek Ubuntu:

```
ip addr
ip route
ping 192.168.10.1
```

Cek RouterOS:

```
/ip address print
/interface print
/tool ping 192.168.10.20
```

Cek desain:

- apakah adapter Ubuntu terhubung ke network yang benar?
- apakah Ubuntu berada di subnet yang sama?
- apakah gateway Ubuntu benar?
- apakah interface RouterOS yang menuju Ubuntu benar?

Jika Ubuntu memakai IP statik:

```
sudo ip addr add 192.168.10.20/24 dev enp0s3
sudo ip route add default via 192.168.10.1
```

Catatan:

> **Masalah Ubuntu ke GNS3 sering bukan masalah routing, tetapi salah adapter atau salah network.**

13.10 Kasus Nyata 4 — NAT Salah Interface

Gejala:

- NAT rule ada,
- client tetap tidak bisa keluar.

Cek:

```
/ip firewall nat print
/interface print
/ip route print
```

Contoh salah:

```
/ip firewall nat add chain=srcnat out-interface=ether2
action=masquerade
```

Padahal `ether2` adalah LAN.

Contoh benar jika WAN adalah `ether1`:

> /ip firewall nat add chain=srcnat out-interface=ether1 action=masquerade comment="NAT LAN to WAN"

Kunci:

> **NAT masquerade harus mengarah ke interface keluar.**

13.11 Kasus Nyata 5 — Firewall Terlalu Ketat

Gejala:

- IP benar,
- route benar,
- NAT benar,
- tetapi paket tetap gagal.

Cek:

```
/ip firewall filter print
/log print
/ip firewall connection print
```

Jika ada rule drop pada chain `input`, akses ke router bisa gagal.
Jika ada rule drop pada chain `forward`, client bisa gagal keluar.

Pola firewall dasar:

```
/ip firewall filter add chain=input
connection-state=established,related action=accept
/ip firewall filter add chain=input connection-state=invalid
action=drop
/ip firewall filter add chain=input in-interface=ether2
action=accept comment="Allow LAN admin"
/ip firewall filter add chain=input protocol=icmp
action=accept
/ip firewall filter add chain=input action=drop comment="Drop
other input"
```

Catatan:

> **Rule firewall dibaca dari atas ke bawah.**
> **Satu rule drop yang terlalu awal bisa mematikan lab.**

13.12 Kasus Nyata 6 — IPv6 Aktif Tetapi Tidak Aman

Gejala:

- IPv6 sudah aktif,
- client bisa ping IPv6,
- tetapi firewall IPv6 belum dibuat.

Cek:

```
/ipv6 address print
/ipv6 route print
/ipv6 firewall filter print
```

Rule dasar:

```
/ipv6 firewall filter add chain=input
connection-state=established,related action=accept
/ipv6 firewall filter add chain=input
connection-state=invalid action=drop
/ipv6 firewall filter add chain=input protocol=icmpv6
action=accept
/ipv6 firewall filter add chain=input in-interface=ether2
action=accept
/ipv6 firewall filter add chain=input action=drop
```

Catatan:

Jangan aktifkan IPv6 tanpa firewall.
Dan jangan blok ICMPv6 sembarangan.

13.13 Kasus Nyata 7 — OSPF Neighbor Tidak Muncul

Gejala:

```
/routing ospf neighbor print
```

kosong.

Cek:

```
/ip address print
/tool ping 10.10.12.2
/routing ospf instance print
/routing ospf area print
/routing ospf interface-template print
```

Kemungkinan:

- router belum bisa saling ping,
- network OSPF salah,
- area salah,
- interface-template salah,
- firewall memblokir.

Langkah disiplin:

1. pastikan IP link antar-router benar,
2. pastikan router bisa saling ping,
3. baru cek konfigurasi OSPF.

Kalimat kunci:

> **OSPF tidak akan menjadi tetangga jika IP dasar antar-router saja belum benar.**

13.14 Kasus Nyata 8 — BGP Session Tidak Established

Gejala:

```
/routing bgp session print
```

belum established.

Cek:

```
/tool ping 10.10.12.2
/routing bgp connection print detail
/ip firewall filter print
/log print
```

Kemungkinan:

- IP peer tidak bisa ping,
- ASN salah,
- remote address salah,
- router-id bermasalah,
- firewall memblokir TCP 179,
- filter terlalu ketat.

Langkah benar:

1. uji ping ke peer,
2. cek ASN sendiri dan ASN lawan,
3. cek koneksi BGP,
4. cek log,
5. cek firewall.

Catatan:

> **BGP tidak akan naik jika konektivitas IP dasar ke peer belum ada.**

13.15 Proyek Mini Akhir

Proyek ini menjadi **portofolio kecil** pembaca. Tujuannya bukan rumit, tetapi menggabungkan kemampuan dari seluruh buku.

Komponen

- **3 CHR**
- **2 VPC**
- **1 Ubuntu Server VM**
- DHCP
- NAT
- Firewall
- IPv6
- OSPF
- BGP simulasi
- Export konfigurasi

Topologi

```
                    ISP-SIM / R3
                         |
                      BGP Sim
                         |
VPC-A --- R1 --- R2 --- Ubuntu Server
           |
         VPC-B
```

Versi lebih terstruktur:

```
LAN-A              Routing Core           LAN-Server
192.168.10.0/24    10.10.12.0/30          192.168.30.0/24

VPC-A ---- R1 -------- R2 ---- Ubuntu Server
            \
             \ HotSpot/LAN-B
              \
              VPC-B
```

R2 ---- R3 sebagai ISP-SIM untuk BGP

13.16 IP Plan Proyek Mini

Perangkat	Interface	IPv4	IPv6
R1	LAN-A	`192.168.10.1/24`	`2001:db8:10::1/64`
R1	R1-R2	`10.10.12.1/30`	`2001:db8:12::1/64`
R2	R1-R2	`10.10.12.2/30`	`2001:db8:12::2/64`
R2	LAN-Server	`192.168.30.1/24`	`2001:db8:30::1/64`
R2	R2-R3	`10.10.23.1/30`	-
R3	R2-R3	`10.10.23.2/30`	-
VPC-A	eth0	DHCP / `192.168.10.10/24`	`2001:db8:10::10/64`
VPC-B	eth0	DHCP	-
Ubuntu Server	enp0s3	`192.168.30.10/24`	`2001:db8:30::10/64`

13.17 Tugas Proyek Mini

Pembaca harus menyelesaikan tugas berikut:

1. Routing Dasar

- R1 dan R2 bisa saling ping.
- R2 dan R3 bisa saling ping.
- VPC-A bisa ping Ubuntu Server.

2. DHCP

- VPC-A mendapat IP dari DHCP R1.
- VPC-B mendapat IP dari DHCP R1 atau segmen terpisah.

3. NAT

- Client LAN bisa keluar melalui R2/R3 simulasi.

4. Firewall

- Router hanya menerima akses admin dari LAN.
- Forward LAN ke server diizinkan.
- Akses tidak perlu diblokir.

5. IPv6

- VPC-A bisa ping IPv6 R1.
- Ubuntu bisa ping IPv6 R2.
- Routing IPv6 sederhana berjalan.

6. OSPF

- R1 dan R2 bertukar route internal.
- Route LAN muncul otomatis.

7. BGP Simulasi

- R2 menjadi customer.
- R3 menjadi ISP-SIM.
- R2 mengumumkan prefix lab ke R3.
- Filter BGP dipasang.

8. Export

Setiap router wajib diekspor:

```
/export file=R1-final
/export file=R2-final
/export file=R3-final
```

13.18 Verifikasi Proyek Mini

RouterOS

```
/ip address print
/ip route print
/ipv6 address print
/ipv6 route print
/ip firewall filter print
/ip firewall nat print
/routing ospf neighbor print
/routing bgp session print
/file print
```

VPC

```
show ip
show ipv6
ping 192.168.10.1
ping 192.168.30.10
trace 192.168.30.10
```

Ubuntu Server

```
ip addr
ip route
ip -6 addr
ip -6 route
ping 192.168.10.10
ping -6 2001:db8:10::10
curl http://127.0.0.1
```

13.19 Template Laporan Proyek Mini

Gunakan format singkat berikut.

```
Nama Proyek:
Tanggal:
Nama Pembuat:

Tujuan:
- ...

Topologi:
- ...

IP Plan:
- ...

Konfigurasi Utama:
- DHCP:
- NAT:
- Firewall:
- IPv6:
- OSPF:
- BGP:

Hasil Uji:
- VPC-A ping gateway: berhasil/gagal
- VPC-A ping Ubuntu: berhasil/gagal
- OSPF neighbor: ada/tidak
- BGP session: established/tidak
- IPv6 ping: berhasil/gagal

Masalah yang Ditemukan:
- ...

Solusi:
- ...

File Export:
- R1-final.rsc
- R2-final.rsc
- R3-final.rsc
```

13.20 Checklist Akhir Bab

Pembaca dianggap selesai Bab 13 jika mampu:

- Menjelaskan prinsip troubleshooting:
  - jangan menebak,
  - cek dari layer bawah,
  - ubah satu hal dalam satu waktu,
  - catat sebelum dan sesudah.
- Menggunakan tools RouterOS:

```
/tool ping
/tool traceroute
/tool torch
/tool sniffer
/log print
/ip route print
/ip firewall connection print
/interface monitor-traffic
```

- Menggunakan tools Ubuntu:

```
ip addr
ip route
ping
traceroute
dig
curl
tcpdump
```

- Menggunakan tools VPC GNS3:

```
show ip
show ipv6
ping
trace
ip dhcp
```

- Menyelesaikan kasus:
  - VPC tidak dapat IP,
  - VPC bisa ping gateway tetapi tidak bisa keluar,
  - Ubuntu VM tidak tersambung ke GNS3,
  - NAT salah interface,
  - firewall terlalu ketat,
  - IPv6 aktif tetapi tidak aman,
  - OSPF neighbor tidak muncul,
  - BGP session tidak established.
- Menyelesaikan proyek mini akhir.
- Mengekspor semua konfigurasi router.

Ringkasan Bab

Bab ini menutup buku dengan pola pikir paling penting: **jaringan harus diuji, bukan diasumsikan**. Pembaca belajar menggunakan tools RouterOS, Ubuntu, dan VPC GNS3 untuk membaca gejala, menemukan penyebab, memperbaiki masalah, dan mendokumentasikan hasil.

Kunci Bab 13:

> **Teknisi jaringan yang baik bukan yang tidak pernah salah konfigurasi. Teknisi jaringan yang baik adalah yang tahu cara menemukan kesalahan, memperbaikinya, dan mendokumentasikannya.**

Lampiran A. Tabel CIDR, Netmask, Jumlah Host

Lampiran ini dipakai sebagai **contekan cepat** saat membuat *IP plan* lab MikroTik, GNS3, VPC, dan Ubuntu VM.

Rumus dasar:

```
Jumlah alamat = 2^(32 - prefix)
Host usable   = jumlah alamat - 2
```

Catatan:

- **Network address** = alamat pertama dalam subnet.
- **Broadcast address** = alamat terakhir dalam subnet.
- **Host usable** = alamat yang dapat dipakai perangkat.
- Untuk `/31` dan `/32`, penggunaan praktis berbeda dari subnet LAN biasa. Di buku ini, `/32` dipakai untuk **loopback**, **router ID**, atau **host route**.

A.1 Tabel CIDR /24 sampai /32

CIDR	Netmask	Jumlah Alamat	Host Usable	Contoh Network	Contoh Broadcast	Umum Dipakai Untuk
`/24`	`255.255.255.0`	256	254	`192.168.10.0`	`192.168.10.255`	LAN kecil, lab kelas
`/25`	`255.255.255.128`	128	126	`192.168.10.0`	`192.168.10.127`	Membagi `/24` menjadi 2 subnet
`/26`	`255.255.255.192`	64	62	`192.168.10.0`	`192.168.10.63`	Segmentasi kecil
`/27`	`255.255.255.224`	32	30	`192.168.10.0`	`192.168.10.31`	Kelompok user kecil
`/28`	`255.255.255.240`	16	14	`192.168.10.0`	`192.168.10.15`	Lab kecil, server kecil
`/29`	`255.255.255.248`	8	6	`192.168.10.0`	`192.168.10.7`	Link kecil, perangkat terbatas
`/30`	`255.255.255.252`	4	2	`192.168.10.0`	`192.168.10.3`	Link antar-router

/31	255.255.255.254	2	2 untuk point-to-point	192.168.10.0	Tidak dipakai seperti broadcast biasa	Link point-to-point khusus
/32	255.255.255.255	1	1 alamat tunggal	192.168.10.1	Tidak ada broadcast	Loopback, router ID, host route

A.2 Contoh Pembagian Subnet dari 192.168.10.0/24

CIDR	Network	Host Pertama	Host Terakhir	Broadcast
/24	192.168.10.0	192.168.10.1	192.168.10.254	192.168.10.255
/25 subnet 1	192.168.10.0	192.168.10.1	192.168.10.126	192.168.10.127
/25 subnet 2	192.168.10.128	192.168.10.129	192.168.10.254	192.168.10.255
/26 subnet 1	192.168.10.0	192.168.10.1	192.168.10.62	192.168.10.63
/26 subnet 2	192.168.10.64	192.168.10.65	192.168.10.126	192.168.10.127
/26 subnet 3	192.168.10.128	192.168.10.129	192.168.10.190	192.168.10.191
/26 subnet 4	192.168.10.192	192.168.10.193	192.168.10.254	192.168.10.255
/30 subnet 1	192.168.10.0	192.168.10.1	192.168.10.2	192.168.10.3
/30 subnet 2	192.168.10.4	192.168.10.5	192.168.10.6	192.168.10.7
/30 subnet 3	192.168.10.8	192.168.10.9	192.168.10.10	192.168.10.11

A.3 Kapan Memakai Prefix Tertentu

Kebutuhan	Prefix yang Disarankan	Alasan
LAN lab sederhana	`/24`	Mudah dipahami, host banyak
Segmentasi user kecil	`/26` atau `/27`	Cukup untuk kelompok kecil
Server kecil / lab terbatas	`/28`	Hemat alamat
Link antar-router	`/30`	Hanya butuh dua IP
Link point-to-point khusus	`/31`	Efisien, tetapi perlu pemahaman lebih
Loopback / router ID	`/32`	Satu alamat tunggal

A.4 Contoh Praktis untuk Lab MikroTik

LAN Client

```
Network : 192.168.10.0/24
Gateway : 192.168.10.1
Client  : 192.168.10.10–192.168.10.254
```

Cocok untuk:

- VPC GNS3,
- Ubuntu Client,
- DHCP Server MikroTik,
- lab NAT dan firewall.

Link Antar-Router

```
Network : 10.10.12.0/30
R1      : 10.10.12.1
R2      : 10.10.12.2
Broadcast: 10.10.12.3
```

Cocok untuk:

- static route,
- OSPF,
- BGP simulasi,
- VPN lab.

Loopback / Router ID

```
R1 Loopback : 1.1.1.1/32
R2 Loopback : 2.2.2.2/32
R3 Loopback : 3.3.3.3/32
```

Cocok untuk:

- OSPF router ID,
- BGP router ID,
- identitas stabil router dalam lab.

A.5 Golden Rules CIDR

- **/24** paling mudah untuk LAN pemula.
- **/30** sangat cocok untuk link antar-router.
- **/32** cocok untuk alamat tunggal seperti *loopback* atau *router ID*.
- Jangan pasang gateway di luar subnet client.
- Jangan pakai alamat **network** untuk host.
- Jangan pakai alamat **broadcast** untuk host.
- Selalu tulis **IP plan** sebelum konfigurasi MikroTik.
- Saat troubleshooting, cek **IP, prefix, gateway, dan route** sebelum menyalahkan firewall.

Lampiran B. Template IP Plan Lab

Lampiran ini dipakai untuk membuat **rencana alamat IP** sebelum konfigurasi MikroTik, GNS3, VPC, dan Ubuntu VM. Tujuannya sederhana: **jangan konfigurasi dulu sebelum alamat, interface, gateway, dan DNS jelas**.

IP plan yang rapi akan membuat lab lebih mudah:

- dikonfigurasi,
- diuji,
- diperbaiki,
- diulang,
- dijadikan portofolio.

B.1 Prinsip Membuat IP Plan

Sebelum mengetik CLI, pastikan sudah jelas:

- **Nama router**.
- **Nama interface**.
- **Interface terhubung ke mana**.
- **IP address dan prefix**.
- **Gateway**.
- **DNS**.
- **Fungsi network**.
- **Catatan konfigurasi khusus**.

Kalimat kunci:

> **IP plan adalah peta. Tanpa peta, troubleshooting menjadi tebakan.**

B.2 Template IP Plan Umum

Perangkat	Interface	Terhubung ke	IP/Prefix	Gateway	DNS	Fungsi	Catatan
R1	ether1	WAN / NAT	DHCP	dari DHCP	dari DHCP / manual	WAN	DHCP Client
R1	ether2	LAN-A	`192.168.10.1/24`	-	`192.168.10.1`	Gateway LAN	DHCP Server
VPC1	eth0	LAN-A	DHCP / `192.168.10.10/24`	`192.168.10.1`	`192.168.10.1`	Client uji	GNS3 VPC
U-SRV1	enp0s3	LAN-A	`192.168.10.20/24`	`192.168.10.1`	`192.168.10.1`	Server uji	Ubuntu VM

B.3 Template Router

Gunakan tabel ini untuk setiap router MikroTik.

Router	Interface	Mode / Link	IP/Prefix	Gateway	DNS	Fungsi
R1	ether1	WAN				
R1	ether2	LAN				
R1	ether3	Link ke R2				
R1	ether4	Cadangan / trunk / lab				

Contoh terisi:

Router	Interface	Mode / Link	IP/Prefix	Gateway	DNS	Fungsi
R1	ether1	WAN NAT	DHCP	DHCP	DHCP	DHCP Client
R1	ether2	LAN-A	`192.168.10.1/24`	-	`192.168.10.1`	Gateway LAN
R1	ether3	R1-R2	`10.10.12.1/30`	-	-	Link routing
R1	ether4	HotSpot	`192.168.30.1/24`	-	`192.168.30.1`	Gateway HotSpot

B.4 Template Client VPC GNS3

Client	Interface	Terhubung ke	IP/Prefix	Gateway	DNS	Fungsi
VPC1	eth0					
VPC2	eth0					
VPC3	eth0					

Contoh terisi:

Client	Interface	Terhubung ke	IP/Prefix	Gateway	DNS	Fungsi
VPC1	eth0	LAN-A	`192.168.10.10/24`	`192.168.10.1`	`192.168.10.1`	Client admin
VPC2	eth0	LAN-B	`192.168.20.10/24`	`192.168.20.1`	`192.168.20.1`	Client user
VPC3	eth0	HotSpot	DHCP	`192.168.30.1`	`192.168.30.1`	Client HotSpot

B.5 Template Ubuntu Server VM

Server	Interface	Terhubung ke	IP/Prefix	Gateway	DNS	Layanan	Catatan
U-SRV1	enp0s3						
U-SRV2	enp0s3						

Contoh terisi:

Server	Interface	Terhubung ke	IP/Prefix	Gateway	DNS	Layanan	Catatan
U-SRV1	enp0s3	LAN-Server	`192.168.30.10/24`	`192.168.30.1`	`192.168.30.1`	Web server	Target port forwarding
U-SRV2	enp0s3	LAN-Admin	`192.168.10.20/24`	`192.168.10.1`	`192.168.10.1`	SSH, tcpdump	Client uji

B.6 Template Link Antar-Router

Gunakan `/30` agar link antar-router rapi dan mudah dibaca.

Link	Router A	Interface A	IP A	Router B	Interface B	IP B	Prefix
R1-R2	R1	ether3		R2	ether1		`/30`
R2-R3	R2	ether2		R3	ether1		`/30`
R1-R3	R1	ether4		R3	ether3		`/30`

Contoh terisi:

Link	Router A	Interface A	IP A	Router B	Interface B	IP B	Prefix
R1-R2	R1	ether3	`10.10.12.1`	R2	ether1	`10.10.12.2`	`/30`
R2-R3	R2	ether2	`10.10.23.1`	R3	ether1	`10.10.23.2`	`/30`
R1-R3	R1	ether4	`10.10.13.1`	R3	ether3	`10.10.13.2`	`/30`

B.7 Template VLAN / Segmentasi LAN

Segmen	VLAN ID	Subnet	Gateway	DNS	Access Port	Trunk Port	Fungsi
Admin	10						
User	20						
Tamu	30						
Server	40						

Contoh terisi:

Segmen	VLAN ID	Subnet	Gateway	DNS	Access Port	Trunk Port	Fungsi
Admin	10	`192.168.10.0/24`	`192.168.10.1`	`192.168.10.1`	ether2	ether5	Admin jaringan
User	20	`192.168.20.0/24`	`192.168.20.1`	`192.168.20.1`	ether3	ether5	Pengguna umum
Tamu	30	`192.168.30.0/24`	`192.168.30.1`	`192.168.30.1`	ether4	ether5	Tamu / HotSpot
Server	40	`192.168.40.0/24`	`192.168.40.1`	`192.168.40.1`	ether6	ether5	Server internal

B.8 Template IPv6 Lab

Gunakan prefix dokumentasi `2001:db8::/32` untuk lab.

Perangkat	Interface	IPv6/Prefix	Gateway IPv6	Fungsi
R1	ether2		-	
VPC1	eth0			
Ubuntu VM	enp0s3			

Contoh terisi:

Perangkat	Interface	IPv6/Prefix	Gateway IPv6	Fungsi
R1	ether2	`2001:db8:10::1/64`	-	Router IPv6 LAN
VPC1	eth0	`2001:db8:10::10/64`	`2001:db8:10::1`	Client IPv6
Ubuntu VM	enp0s3	`2001:db8:10::20/64`	`2001:db8:10::1`	Client/server IPv6

B.9 Template DNS dan Gateway

Network	Gateway	DNS Utama	DNS Cadangan	Keterangan
`192.168.10.0/24`				
`192.168.20.0/24`				
`192.168.30.0/24`				

Contoh terisi:

Network	Gateway	DNS Utama	DNS Cadangan	Keterangan
`192.168.10.0/24`	`192.168.10.1`	`192.168.10.1`	`1.1.1.1`	LAN Admin
`192.168.20.0/24`	`192.168.20.1`	`192.168.20.1`	`1.1.1.1`	LAN User
`192.168.30.0/24`	`192.168.30.1`	`192.168.30.1`	`1.1.1.1`	HotSpot / Tamu

B.10 Template Route Plan

Router	Tujuan / Prefix	Gateway / Next-hop	Jenis Route	Keterangan
R1			Static / Default / OSPF / BGP	
R2			Static / Default / OSPF / BGP	
R3			Static / Default / OSPF / BGP	

Contoh terisi:

Router	Tujuan / Prefix	Gateway / Next-hop	Jenis Route	Keterangan
R1	`192.168.30.0/24`	`10.10.12.2`	Static	LAN di belakang R3
R3	`192.168.10.0/24`	`10.10.23.1`	Static	LAN di belakang R1
R1	`0.0.0.0/0`	DHCP / WAN	Default	Akses keluar
R2	`192.168.10.0/24`	OSPF	Dynamic	Dipelajari dari OSPF

B.11 Contoh IP Plan Lengkap Lab 3 Router

Topologi:

```
VPC-A ---- R1 ---- R2 ---- R3 ---- VPC-C
```

Perangkat	Interface	Terhubung ke	IP/Prefix	Gateway	DNS	Fungsi
R1	ether1	VPC-A	192.168.10.1/24	-	192.168.10.1	Gateway LAN-A
R1	ether2	R2	10.10.12.1/30	-	-	Link R1-R2
R2	ether1	R1	10.10.12.2/30	-	-	Link R2-R1
R2	ether2	R3	10.10.23.1/30	-	-	Link R2-R3
R3	ether1	R2	10.10.23.2/30	-	-	Link R3-R2
R3	ether2	VPC-C	192.168.30.1/24	-	192.168.30.1	Gateway LAN-C
VPC-A	eth0	R1	192.168.10.10/24	192.168.10.1	192.168.10.1	Client A
VPC-C	eth0	R3	192.168.30.10/24	192.168.30.1	192.168.30.1	Client C

B.12 Checklist IP Plan Sebelum Konfigurasi

Sebelum mengetik CLI RouterOS, pastikan:

- Semua router sudah punya **nama**.
- Semua interface sudah jelas terhubung ke mana.
- Tidak ada IP duplikat.
- Gateway client berada dalam subnet yang sama.
- DNS client sudah jelas.
- Link antar-router memakai prefix yang sesuai, misalnya `/30`.
- LAN memakai prefix yang mudah dibaca, misalnya `/24`.
- VLAN punya:
    - VLAN ID,
    - subnet,
    - gateway,
    - port access/trunk.
- IPv6 lab memakai prefix dokumentasi `2001:db8::/32`.
- Route plan sudah jelas.
- File export akan disimpan setelah konfigurasi.

B.13 Golden Rules IP Plan

- **Tulis dulu, konfigurasi kemudian.**
- **Satu interface, satu fungsi yang jelas.**
- **Jangan pasang IP gateway di interface yang salah.**
- **Jangan menaruh semua port ke satu bridge tanpa alasan.**
- **Gunakan `/24` untuk LAN awal.**
- **Gunakan `/30` untuk link antar-router.**
- **Gunakan `/32` untuk loopback atau router ID.**
- **Gunakan DNS router untuk lab agar mudah dikendalikan.**
- **Simpan export setelah lab berhasil.**
- **Troubleshooting dimulai dari IP plan.**

Lampiran C. Template Interface Plan

Lampiran ini dipakai untuk merapikan **pemetaan interface** sebelum membangun lab MikroTik, VirtualBox, GNS3, VPC, dan Ubuntu VM.

Tujuannya sederhana: **jangan sampai `ether1`, `ether2`, link GNS3, VPC, dan Ubuntu VM tertukar**.

C.1 Prinsip Interface Plan

Sebelum konfigurasi, tentukan:

- **Interface mana untuk WAN**.
- **Interface mana untuk LAN**.
- **Interface mana untuk link antar-router**.
- **Interface mana menuju VPC GNS3**.
- **Interface mana menuju Ubuntu VM**.
- **Interface mana untuk manajemen**.
- **Interface mana untuk VLAN, HotSpot, VPN, atau server uji**.

Kalimat kunci:

> **IP yang benar tetap gagal jika dipasang di interface yang salah.**

C.2 Pola Penamaan Interface yang Disarankan

Interface	Fungsi Umum	Contoh Keterangan
`ether1`	WAN / NAT / jaringan luar	DHCP Client, NAT keluar
`ether2`	LAN utama	VPC, client, DHCP Server
`ether3`	LAN kedua / Ubuntu VM / HotSpot	server uji, HotSpot, user LAN
`ether4`	Link antar-router	R1-R2, R2-R3
`ether5`	Trunk / VLAN / cadangan	VLAN admin/user/tamu

Catatan:

- Pola ini tidak wajib, tetapi **konsisten**.
- Jika topologi berbeda, tetap buat tabel agar jelas.
- Hindari memakai `ether1` untuk LAN jika di bab lain `ether1` dipakai WAN, kecuali dicatat jelas.

C.3 Template Interface Plan Router

Router	Interface	Terhubung ke	Media / Link	Fungsi	IP/Prefix	Catatan
R1	ether1			WAN / NAT		
R1	ether2			LAN / VPC		
R1	ether3			Ubuntu VM / Server		
R1	ether4			Link router		
R1	ether5			VLAN / trunk / cadangan		

Contoh terisi:

Router	Interface	Terhubung ke	Media / Link	Fungsi	IP/Prefix	Catatan
R1	ether1	NAT GNS3 / VirtualBox NAT	WAN simulasi	DHCP Client	DHCP	Keluar jaringan luar
R1	ether2	VPC1	GNS3 link	LAN client	`192.168.10.1/24`	DHCP Server
R1	ether3	Ubuntu VM	Internal Network / GNS3 Cloud	Server uji	`192.168.20.1/24`	Web/SSH test
R1	ether4	R2 ether1	GNS3 link	Link antar-router	`10.10.12.1/30`	OSPF/static
R1	ether5	Switch/VLAN	GNS3 link	Trunk VLAN	-	VLAN 10,20,30

C.4 Template Link GNS3

Gunakan tabel ini untuk mencatat semua link di GNS3.

Link	Node A	Interface A	Node B	Interface B	Network	Fungsi
L1						
L2						
L3						
L4						

Contoh terisi:

Link	Node A	Interface A	Node B	Interface B	Network	Fungsi
L1	VPC1	eth0	R1	ether2	`192.168.10.0/24`	LAN-A
L2	R1	ether4	R2	ether1	`10.10.12.0/30`	Link R1-R2
L3	R2	ether2	R3	ether1	`10.10.23.0/30`	Link R2-R3
L4	R3	ether2	VPC2	eth0	`192.168.30.0/24`	LAN-C

C.5 Template VPC GNS3

VPC	Interface	Terhubung ke Router	Interface Router	IP/Prefix	Gateway	Fungsi
VPC1	eth0					
VPC2	eth0					
VPC3	eth0					

Contoh terisi:

VPC	Interface	Terhubung ke Router	Interface Router	IP/Prefix	Gateway	Fungsi
VPC1	eth0	R1	ether2	DHCP / `192.168.10.10/24`	`192.168.10.1`	Client LAN-A
VPC2	eth0	R3	ether2	`192.168.30.10/24`	`192.168.30.1`	Client LAN-C
VPC3	eth0	R1	ether3	DHCP	`192.168.30.1`	Client HotSpot

C.6 Template Ubuntu VM

VM	Interface Ubuntu	Terhubung ke	Interface Router	Network Mode	IP/Prefix	Gateway	Fungsi
U-SRV1	enp0s3						
U-CLIENT1	enp0s3						

Contoh terisi:

VM	Interface Ubuntu	Terhubung ke	Interface Router	Network Mode	IP/Prefix	Gateway	Fungsi
U-SRV1	enp0s3	R1	ether3	Internal Network `net-server`	`192.168.20.10/24`	`192.168.20.1`	Web server uji
U-CLIENT1	enp0s3	R1	ether2	Internal Network `net-lan10`	DHCP	`192.168.10.1`	Client Ubuntu
U-IPV6	enp0s3	R1	ether2	GNS3/Cloud link	`2001:db8:10::20/64`	`2001:db8:10::1`	Client IPv6

Catatan:

- Nama interface Ubuntu bisa berbeda, misalnya `enp0s3`, `ens3`, atau `eth0`.
- Selalu cek dengan:

```
ip addr
```

C.7 Template VirtualBox Adapter

VM	Adapter	Mode VirtualBox	Terhubung ke	Fungsi	Catatan
CHR-R1	Adapter 1				
CHR-R1	Adapter 2				
CHR-R1	Adapter 3				
U-SRV1	Adapter 1				

Contoh terisi:

VM	Adapter	Mode VirtualBox	Terhubung ke	Fungsi	Catatan
CHR-R 1	Adapter 1	NAT	Jaringan luar	WAN	menjadi `ether1`
CHR-R 1	Adapter 2	Internal Network `net-lan10`	VPC/Ubuntu LAN	LAN	menjadi `ether2`
CHR-R 1	Adapter 3	Internal Network `net-r1-r2`	R2	Link antar-router	menjadi `ether3`
U-SRV1	Adapter 1	Internal Network `net-server`	R1	Server uji	cek nama interface di Ubuntu

C.8 Template Bridge dan VLAN Interface

Router	Bridge	Port Anggota	VLAN ID	VLAN Interface	Fungsi	Gateway
R1						
R1						

Contoh terisi:

Router	Bridge	Port Anggota	VLAN ID	VLAN Interface	Fungsi	Gateway
R1	bridge-lan	ether2, ether3	-	-	LAN biasa	`192.168.20.1/24`
R1	bridge-vlan	ether2, ether3, ether4	10	vlan10-admin	Admin	`192.168.10.1/24`
R1	bridge-vlan	ether2, ether3, ether4	20	vlan20-user	User	`192.168.20.1/24`
R1	bridge-vlan	ether2, ether3, ether4	30	vlan30-guest	Tamu	`192.168.30.1/24`

C.9 Template Komentar Interface RouterOS

Gunakan komentar agar interface mudah dibaca.

```
/interface comment ether1 comment="WAN / NAT GNS3"
/interface comment ether2 comment="LAN-A to VPC1"
/interface comment ether3 comment="Ubuntu Server VM"
/interface comment ether4 comment="Link R1-R2"
/interface comment ether5 comment="Trunk VLAN"
```

Verifikasi:

```
/interface print
```

Contoh standar komentar:

Interface	Komentar
ether1	`WAN / NAT GNS3`
ether2	`LAN-A to VPC1`
ether3	`Ubuntu Server VM`
ether4	`Link R1-R2`
ether5	`Trunk VLAN`

C.10 Contoh Interface Plan Lab 1 CHR + 1 VPC

Topologi:

```
VPC1 ---- R1
```

Router	Interface	Terhubung ke	Fungsi	IP/Prefix
R1	ether1	VPC1 eth0	LAN	`192.168.10.1/24`

VPC	Interface	Terhubung ke	IP/Prefix	Gateway
VPC1	eth0	R1 ether1	`192.168.10.10/24`	`192.168.10.1`

CLI komentar:

```
/interface comment ether1 comment="LAN to VPC1"
```

C.11 Contoh Interface Plan Lab 1 CHR + 2 VPC

Topologi:

```
VPC1 ---- ether2
             \
              R1 bridge-lan
             /
VPC2 ---- ether3
```

Router	Interface	Terhubung ke	Fungsi	IP/Prefix
R1	bridge-lan	ether2, ether3	Gateway LAN	192.168.20.1/24
R1	ether2	VPC1	Bridge port	-
R1	ether3	VPC2	Bridge port	-

VPC	IP/Prefix	Gateway
VPC1	192.168.20.10/24	192.168.20.1
VPC2	192.168.20.20/24	192.168.20.1

CLI:

```
/interface bridge add name=bridge-lan
/interface bridge port add bridge=bridge-lan interface=ether2
/interface bridge port add bridge=bridge-lan interface=ether3
/ip address add address=192.168.20.1/24 interface=bridge-lan
```

C.12 Contoh Interface Plan Lab 3 CHR Routing

Topologi:

```
VPC-A ---- R1 ---- R2 ---- R3 ---- VPC-C
```

Router	Interface	Terhubung ke	Network	IP
R1	ether1	VPC-A	192.168.10.0/24	192.168.10.1
R1	ether2	R2 ether1	10.10.12.0/30	10.10.12.1
R2	ether1	R1 ether2	10.10.12.0/30	10.10.12.2
R2	ether2	R3 ether1	10.10.23.0/30	10.10.23.1
R3	ether1	R2 ether2	10.10.23.0/30	10.10.23.2
R3	ether2	VPC-C	192.168.30.0/24	192.168.30.1

VPC	Terhubung ke	IP/Prefix	Gateway
VPC-A	R1 ether1	192.168.10.10/24	192.168.10.1
VPC-C	R3 ether2	192.168.30.10/24	192.168.30.1

C.13 Contoh Interface Plan Lab VPN

Topologi:

```
VPC-A ---- R1 ---- WAN-SIM ---- R2 ---- VPC-B
```

Router	Interface	Terhubung ke	Network	IP
R1	ether1	VPC-A	192.168.10.0/24	192.168.10.1
R1	ether2	R2 ether1	172.16.99.0/24	172.16.99.1
R2	ether1	R1 ether2	172.16.99.0/24	172.16.99.2
R2	ether2	VPC-B	192.168.20.0/24	192.168.20.1
R1	wg-site	VPN tunnel	10.10.10.0/30	10.10.10.1
R2	wg-site	VPN tunnel	10.10.10.0/30	10.10.10.2

C.14 Kesalahan Umum Interface Plan

Kesalahan	Gejala	Solusi
`ether1` dan `ether2` tertukar	NAT/DHCP gagal	Cek `/interface print` dan komentar
IP gateway dipasang di port bridge	Client tidak stabil / bingung	Pasang IP di `bridge-lan`
VPC terhubung ke link salah	Ping gateway gagal	Cek kabel GNS3 dan tabel link
Ubuntu VM salah adapter	Tidak bisa ping router	Cek mode adapter dan `ip addr`
Interface tanpa komentar	Troubleshooting lambat	Tambahkan `/interface comment`
VLAN dibuat tanpa mencatat port	Access/trunk kacau	Buat tabel VLAN dan port
Link antar-router tidak pakai `/30`	IP plan membingungkan	Gunakan `/30` untuk lab dasar

C.15 Checklist Interface Plan Sebelum Lab

Sebelum menyalakan lab, pastikan:

- Semua router punya nama: R1, R2, R3.
- Semua interface sudah punya fungsi.
- Semua link GNS3 tercatat.
- Semua VPC tercatat gateway-nya.
- Ubuntu VM tahu interface yang dipakai.
- Mode VirtualBox sudah sesuai:
    - NAT untuk WAN,
    - Internal Network untuk lab,
    - Host-only untuk manajemen.
- Bridge dan VLAN dicatat terpisah.
- Interface yang menjadi bridge port tidak diberi IP langsung.
- Komentar interface sudah disiapkan.
- Setelah konfigurasi, jalankan:

```
/interface print
/ip address print
/interface bridge port print
```

C.16 Golden Rules Interface Plan

- **Satu interface harus punya satu fungsi yang jelas.**
- **Jangan mengingat-ingat interface; tulis di tabel.**
- **Komentari semua interface penting di RouterOS.**
- **Untuk bridge, IP gateway dipasang di bridge, bukan port anggota.**
- **Untuk VLAN, gateway dipasang di interface VLAN.**
- **Untuk NAT, pastikan `out-interface` adalah WAN.**
- **Untuk VPC, pastikan link GNS3 benar.**
- **Untuk Ubuntu VM, cek nama interface dengan `ip addr`.**
- **Troubleshooting pertama: cek interface, baru cek IP.**

Lampiran D. Template Konfigurasi Router Dasar

Lampiran ini berisi template konfigurasi **router dasar MikroTik** untuk lab VirtualBox, GNS3, VPC, dan Ubuntu VM.

Fokus template:

- **DHCP Client** di sisi WAN.
- **IP statik** di sisi LAN.
- **DHCP Server** untuk client.
- **DNS resolver/cache**.
- **Default route** dari DHCP Client.
- Verifikasi koneksi dasar.

Contoh ini memakai pola umum:

```
WAN / NAT GNS3 / NAT VirtualBox
        |
      ether1
   MikroTik CHR
      ether2
        |
   VPC / Ubuntu Client
```

D.1 Asumsi Topologi

Komponen	Interface	IP / Fungsi
MikroTik WAN	`ether1`	DHCP Client
MikroTik LAN	`ether2`	`192.168.10.1/24`
DHCP Pool LAN	-	`192.168.10.100-192.168.10.200`
VPC / Ubuntu Client	eth0 / enp0s3	DHCP
DNS Client	-	`192.168.10.1`
Gateway Client	-	`192.168.10.1`

D.2 Template Konfigurasi Cepat RouterOS v7/v6

Gunakan template ini untuk membangun router dasar dari nol.

```
/system identity set name=R1-BASIC

/ip dhcp-client add interface=ether1 disabled=no comment="WAN
DHCP Client"

/ip address add address=192.168.10.1/24 interface=ether2
comment="Gateway LAN"

/ip pool add name=pool-lan
ranges=192.168.10.100-192.168.10.200

/ip dhcp-server add name=dhcp-lan interface=ether2
address-pool=pool-lan disabled=no comment="DHCP Server LAN"

/ip dhcp-server network add address=192.168.10.0/24
gateway=192.168.10.1 dns-server=192.168.10.1 comment="LAN
Network"

/ip dns set allow-remote-requests=yes servers=1.1.1.1

/export file=R1-basic
```

Catatan:

- Template ini **belum membuat NAT**.
- NAT dibahas pada lampiran khusus dan Bab 5.
- Jika ingin client LAN bisa keluar ke jaringan luar, tambahkan rule NAT masquerade.

D.3 Template dengan NAT Masquerade Opsional

Gunakan jika client LAN perlu keluar ke jaringan luar.

```
/system identity set name=R1-BASIC-NAT

/ip dhcp-client add interface=ether1 disabled=no comment="WAN
DHCP Client"

/ip address add address=192.168.10.1/24 interface=ether2
comment="Gateway LAN"

/ip pool add name=pool-lan
ranges=192.168.10.100-192.168.10.200

/ip dhcp-server add name=dhcp-lan interface=ether2
address-pool=pool-lan disabled=no comment="DHCP Server LAN"

/ip dhcp-server network add address=192.168.10.0/24
gateway=192.168.10.1 dns-server=192.168.10.1 comment="LAN
Network"

/ip dns set allow-remote-requests=yes servers=1.1.1.1

/ip firewall nat add chain=srcnat out-interface=ether1
action=masquerade comment="NAT LAN to WAN"

/export file=R1-basic-nat
```

D.4 Penjelasan Singkat Setiap Bagian

Bagian	Fungsi
`/system identity`	Memberi nama router agar mudah dikenali
`/ip dhcp-client`	Mengambil IP otomatis dari sisi WAN
`/ip address`	Memberi IP gateway pada LAN
`/ip pool`	Menentukan rentang IP untuk client
`/ip dhcp-server`	Mengaktifkan DHCP Server pada LAN
`/ip dhcp-server network`	Menentukan gateway dan DNS untuk client
`/ip dns`	Mengatur DNS router dan mengizinkan client memakai router sebagai DNS
`/ip firewall nat`	Mengubah alamat private client agar bisa keluar
`/export`	Menyimpan konfigurasi sebagai teks

D.5 Verifikasi RouterOS

Setelah konfigurasi, jangan langsung lanjut. Verifikasi dulu.

Cek DHCP Client WAN

```
/ip dhcp-client print
```

Yang dicari:

```
status=bound
```

Jika belum `bound`, cek:

- interface WAN benar atau tidak,
- adapter VirtualBox/GNS3 benar atau tidak,
- NAT/upstream aktif atau tidak.

Cek IP Address

```
/ip address print
```

Yang dicari:

- `ether1` mendapat IP dari DHCP.
- `ether2` punya `192.168.10.1/24`.

Cek Default Route

```
/ip route print
```

Yang dicari:

```
0.0.0.0/0
```

Jika default route tidak muncul:

- cek DHCP Client,
- cek apakah `add-default-route` aktif,
- cek koneksi WAN.

Cek DNS

```
/ip dns print
```

Yang dicari:

- `servers=1.1.1.1`
- `allow-remote-requests=yes`

Cek DHCP Server

```
/ip dhcp-server print
/ip dhcp-server network print
/ip pool print
```

Yang dicari:

- DHCP Server aktif.
- Pool benar.
- Network benar.
- Gateway dan DNS benar.

Cek Lease DHCP

```
/ip dhcp-server lease print
```

Jika client sudah meminta IP, lease akan muncul.

D.6 Verifikasi VPC GNS3

Di VPC:

```
PC1> ip dhcp
PC1> show ip
PC1> ping 192.168.10.1
```

Jika NAT sudah ditambahkan, lanjutkan:

```
PC1> ping 1.1.1.1
PC1> trace 1.1.1.1
```

Target hasil:

- VPC mendapat IP dari pool.
- VPC bisa ping gateway.
- Jika NAT aktif, VPC bisa ping IP luar.

D.7 Verifikasi Ubuntu VM

Di Ubuntu VM:

```
ip addr
ip route
ping 192.168.10.1
```

Jika NAT sudah aktif:

```
ping 1.1.1.1
traceroute 1.1.1.1
dig example.net
```

Yang harus dicek:

- Ubuntu mendapat IP `192.168.10.x`.
- Default route via `192.168.10.1`.
- DNS bisa bekerja.

D.8 Template IP Plan untuk Router Dasar

Perangkat	Interface	IP/Prefix	Gateway	DNS	Fungsi
R1	ether1	DHCP	DHCP	DHCP	WAN
R1	ether2	`192.168.10.1/24`	-	`192.168.10.1`	LAN
VPC1	eth0	DHCP	`192.168.10.1`	`192.168.10.1`	Client uji
Ubuntu VM	enp0s3	DHCP	`192.168.10.1`	`192.168.10.1`	Client/server uji

D.9 Template Interface Plan

Router	Interface	Terhubung ke	Fungsi	Catatan
R1	ether1	NAT GNS3 / NAT VirtualBox	WAN	DHCP Client
R1	ether2	VPC / Ubuntu LAN	LAN	DHCP Server
VPC1	eth0	R1 ether2	Client	DHCP
Ubuntu VM	enp0s3	R1 ether2	Client/server	DHCP atau statik

Tambahkan komentar interface:

```
/interface comment ether1 comment="WAN DHCP Client"
```

```
/interface comment ether2 comment="LAN DHCP Server"
```

D.10 Template Jika LAN Menggunakan IP Statik di Client

Kadang VPC atau Ubuntu perlu IP manual.

RouterOS

```
/ip address add address=192.168.10.1/24 interface=ether2
comment="Gateway LAN"
```

VPC

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> ping 192.168.10.1
```

Ubuntu

```
sudo ip addr add 192.168.10.20/24 dev enp0s3
sudo ip route add default via 192.168.10.1
ping 192.168.10.1
```

Catatan:

- Gunakan DHCP untuk lab awal.
- Gunakan IP statik jika ingin server uji atau port forwarding.

D.11 Template Konfigurasi DNS Client ke Router

Jika client memakai router sebagai DNS, router harus mengizinkan query dari LAN.

RouterOS

```
/ip dns set allow-remote-requests=yes servers=1.1.1.1
```

DHCP Network

```
/ip dhcp-server network add address=192.168.10.0/24
gateway=192.168.10.1 dns-server=192.168.10.1
```

Makna:

- Client menerima DNS `192.168.10.1`.
- Router meneruskan query DNS ke server DNS di atas.

Catatan keamanan:

> **Jangan membuka DNS resolver ke jaringan luar tanpa firewall.**
> Untuk lab, izinkan dari LAN saja.

D.12 Template Reset Lab Router Dasar

Jika ingin mengulang lab dari nol, lakukan dengan hati-hati.

```
/export file=backup-sebelum-reset
/system reset-configuration no-defaults=yes skip-backup=yes
```

Catatan:

- Jalankan `export` sebelum reset.
- Reset hanya untuk lab.
- Jangan menjalankan reset pada router produksi tanpa backup dan izin.

D.13 Kesalahan Umum

Masalah	Penyebab Umum	Solusi
DHCP Client tidak `bound`	`ether1` salah link	Cek adapter WAN
Client tidak dapat IP	DHCP Server salah interface	Pastikan DHCP Server di `ether2`
Client dapat IP tapi tidak bisa ping router	IP gateway salah / link salah	Cek `show ip` dan `/ip address print`
Router bisa keluar, client tidak	NAT belum dibuat	Tambah masquerade
DNS gagal	DNS router belum diatur	Cek `/ip dns print`
Default route tidak ada	DHCP Client belum dapat gateway	Cek `/ip dhcp-client print`
Lease kosong	Client belum minta IP	Jalankan `ip dhcp` di VPC
IP dipasang di interface salah	ether1/ether2 tertukar	Gunakan interface plan

D.14 Checklist Akhir Template

Router dasar dianggap berhasil jika:

- DHCP Client WAN status `bound`.
- `ether2` punya IP `192.168.10.1/24`.
- DHCP Server LAN aktif.
- VPC mendapat IP dari DHCP.
- Ubuntu VM mendapat IP dari DHCP atau IP statik benar.
- Client bisa ping gateway.
- Router punya default route.
- DNS router aktif.
- Jika NAT ditambahkan, client bisa ping IP luar.
- Konfigurasi sudah diekspor:

```
/export file=R1-basic-final
```

D.15 Golden Rules Router Dasar

- **WAN menerima IP, LAN memberi IP.**
- **Gateway client harus IP LAN router.**
- **DHCP Server harus berada di interface LAN.**
- **DNS client boleh diarahkan ke router.**
- **Default route harus ada sebelum client bisa keluar.**
- **NAT dibutuhkan jika client private ingin keluar.**
- **Jangan lanjut ke firewall rumit sebelum router dasar berhasil.**
- **Setiap lab berhasil wajib di-export.**

Lampiran E. Template NAT dan Masquerade

Lampiran ini berisi template cepat untuk membuat **NAT LAN ke WAN** pada MikroTik RouterOS. Template ini dipakai setelah router dasar sudah berjalan:

- WAN sudah mendapat IP.
- LAN sudah punya IP gateway.
- Client sudah mendapat IP dari DHCP atau statik.
- Router sudah punya *default route*.

Tujuan utama: **client di LAN bisa keluar ke jaringan luar melalui MikroTik**.

E.1 Topologi Dasar NAT

```
Jaringan luar / NAT GNS3 / NAT VirtualBox
                |
            ether1
        MikroTik CHR
            ether2
              |
      VPC / Ubuntu Client
```

Asumsi interface:

Interface	Fungsi	Contoh
`ether1`	WAN / keluar	DHCP Client
`ether2`	LAN / client	`192.168.10.1/24`
VPC / Ubuntu	Client LAN	`192.168.10.x/24`

E.2 Template NAT Masquerade RouterOS v7/v6

Gunakan jika IP WAN bersifat dinamis, misalnya dari NAT VirtualBox, NAT GNS3, atau DHCP Client.

```
/ip firewall nat add chain=srcnat out-interface=ether1
action=masquerade comment="NAT LAN to WAN"
```

Makna:

- `chain=srcnat` → mengubah alamat sumber paket.
- `out-interface=ether1` → hanya berlaku untuk paket yang keluar lewat WAN.
- `action=masquerade` → memakai IP interface WAN sebagai alamat sumber.
- `comment` → memudahkan dokumentasi dan troubleshooting.

Catatan penting:

> **Pastikan `out-interface` adalah interface WAN.**
> Jika WAN adalah `ether1`, gunakan `ether1`.
> Jika WAN berbeda, sesuaikan.

E.3 Template Router Dasar + NAT Lengkap

Gunakan template ini untuk lab awal 1 router + 1 LAN.

```
/system identity set name=R1-NAT

/ip dhcp-client add interface=ether1 disabled=no comment="WAN
DHCP Client"
/ip address add address=192.168.10.1/24 interface=ether2
comment="Gateway LAN"

/ip pool add name=pool-lan
ranges=192.168.10.100-192.168.10.200
/ip dhcp-server add name=dhcp-lan interface=ether2
address-pool=pool-lan disabled=no comment="DHCP LAN"
/ip dhcp-server network add address=192.168.10.0/24
gateway=192.168.10.1 dns-server=192.168.10.1 comment="LAN
Network"

/ip dns set allow-remote-requests=yes servers=1.1.1.1

/ip firewall nat add chain=srcnat out-interface=ether1
action=masquerade comment="NAT LAN to WAN"

/export file=R1-nat-basic
```

E.4 Verifikasi di RouterOS

1. Cek DHCP Client WAN

```
/ip dhcp-client print
```

Yang dicari:

```
status=bound
```

Jika belum `bound`, cek:

- interface WAN benar,
- link WAN benar,
- NAT/upstream aktif.

2. Cek IP Address

```
/ip address print
```

Yang dicari:

- `ether1` mendapat IP dari DHCP.
- `ether2` punya `192.168.10.1/24`.

3. Cek Default Route

```
/ip route print
```

Yang dicari:

```
0.0.0.0/0
```

Jika tidak ada *default route*, client tidak akan bisa keluar walaupun NAT sudah dibuat.

4. Cek NAT Rule

```
/ip firewall nat print
```

Yang dicari:

```
chain=srcnat action=masquerade out-interface=ether1
```

5. Cek Connection Tracking

```
/ip firewall connection print
```

Jalankan setelah client melakukan `ping` atau akses keluar.

Jika koneksi muncul, berarti paket client melewati router dan dilacak oleh RouterOS.

6. Ping dari Router

```
/tool ping 1.1.1.1
/tool traceroute 1.1.1.1
```

Jika router sendiri tidak bisa ping keluar, masalah bukan di NAT client. Cek:

- DHCP Client WAN.
- Default route.
- Link WAN.
- Firewall input/output jika ada rule khusus.

E.5 Verifikasi dari VPC GNS3

Di VPC:

```
PC1> ip dhcp
PC1> show ip
PC1> ping 192.168.10.1
PC1> ping 1.1.1.1
PC1> trace 1.1.1.1
```

Urutan membaca hasil:

Uji	Makna
`show ip`	memastikan VPC mendapat IP, gateway, DNS
`ping 192.168.10.1`	memastikan LAN ke router berhasil
`ping 1.1.1.1`	memastikan route + NAT bekerja
`trace 1.1.1.1`	melihat jalur paket keluar

Jika `ping gateway` gagal, jangan cek NAT dulu. Cek:

- IP VPC,
- link GNS3,
- IP `ether2`,
- DHCP Server LAN.

E.6 Verifikasi dari Ubuntu VM

Di Ubuntu:

```
ip addr
ip route
ping 192.168.10.1
ping 1.1.1.1
traceroute 1.1.1.1
```

Jika ingin cek DNS:

```
dig example.net
curl http://example.net
```

Urutan troubleshooting:

Gejala	Kemungkinan
Tidak ada IP	DHCP gagal / adapter salah
Tidak ada default route	DHCP network salah / IP statik belum gateway
Ping gateway gagal	link atau subnet salah
Ping IP luar gagal	NAT/default route/firewall
Ping IP luar bisa, nama domain gagal	DNS bermasalah

E.7 Template NAT dengan Interface List

Untuk lab yang lebih rapi, bisa memakai **interface list**.

RouterOS v7/v6

```
/interface list add name=WAN
/interface list add name=LAN

/interface list member add list=WAN interface=ether1
/interface list member add list=LAN interface=ether2

/ip firewall nat add chain=srcnat out-interface-list=WAN
action=masquerade comment="NAT LAN to WAN via WAN list"
```

Kelebihan:

- lebih rapi,
- mudah jika WAN berubah,
- cocok untuk firewall yang lebih terstruktur.

Tetapi untuk pemula, `out-interface=ether1` lebih mudah dipahami.

E.8 Template NAT untuk Beberapa LAN

Jika router punya beberapa LAN, NAT tetap cukup satu rule jika semua keluar lewat WAN.

Contoh:

Interface	Network
ether2	192.168.10.0/24
ether3	192.168.20.0/24
ether4	192.168.30.0/24
ether1	WAN

Template:

```
/ip firewall nat add chain=srcnat out-interface=ether1
action=masquerade comment="NAT all LAN to WAN"
```

Jika ingin lebih ketat, pakai `src-address`:

```
/ip firewall nat add chain=srcnat src-address=192.168.10.0/24
out-interface=ether1 action=masquerade comment="NAT LAN10 to
WAN"
/ip firewall nat add chain=srcnat src-address=192.168.20.0/24
out-interface=ether1 action=masquerade comment="NAT LAN20 to
WAN"
/ip firewall nat add chain=srcnat src-address=192.168.30.0/24
out-interface=ether1 action=masquerade comment="NAT LAN30 to
WAN"
```

E.9 Template NAT untuk Lab GNS3

Topologi:

```
NAT Node GNS3
      |
    ether1
  MikroTik
    ether2
      |
     VPC1
```

RouterOS:

```
/ip dhcp-client add interface=ether1 disabled=no
/ip address add address=192.168.10.1/24 interface=ether2
/ip pool add name=pool-lan
ranges=192.168.10.100-192.168.10.200
/ip dhcp-server add name=dhcp-lan interface=ether2
address-pool=pool-lan disabled=no
/ip dhcp-server network add address=192.168.10.0/24
gateway=192.168.10.1 dns-server=192.168.10.1
/ip dns set allow-remote-requests=yes servers=1.1.1.1
/ip firewall nat add chain=srcnat out-interface=ether1
action=masquerade comment="NAT VPC to WAN"
```

VPC:

```
PC1> ip dhcp
PC1> show ip
PC1> ping 192.168.10.1
PC1> ping 1.1.1.1
PC1> trace 1.1.1.1
```

E.10 Template NAT untuk Ubuntu VM

Topologi:

```
NAT / jaringan luar
        |
      ether1
   MikroTik
      ether2
        |
   Ubuntu VM
```

RouterOS sama seperti template NAT dasar.

Ubuntu DHCP:

```
ip addr
ip route
ping 192.168.10.1
ping 1.1.1.1
traceroute 1.1.1.1
```

Ubuntu statik:

```
sudo ip addr add 192.168.10.20/24 dev enp0s3
sudo ip route add default via 192.168.10.1
ping 192.168.10.1
ping 1.1.1.1
```

Catatan:

- Nama interface Ubuntu bisa berbeda.
- Cek dengan `ip addr`.

E.11 Troubleshooting NAT

Masalah	Gejala	Cek	Solusi
Salah `out-interface`	NAT ada, client gagal keluar	`/ip firewall nat print`	Sesuaikan ke WAN
Tidak ada default route	Router tidak bisa keluar	`/ip route print`	Perbaiki DHCP Client/WAN
Client tidak punya gateway	Client tidak bisa keluar	`show ip` / `ip route`	Set gateway `192.168.10.1`
DNS salah	Ping IP bisa, nama gagal	`/ip dns print`, `dig`	Set DNS router/client
Firewall forward drop	NAT benar tapi tetap gagal	`/ip firewall filter print`	Izinkan LAN ke WAN
DHCP gagal	Client tidak dapat IP	`/ip dhcp-server lease print`	Cek DHCP Server dan interface
WAN belum dapat IP	DHCP Client belum bound	`/ip dhcp-client print`	Cek link WAN
NAT belum kena trafik	Connection kosong	`/ip firewall connection print`	Cek route/link client

E.12 Flow Verifikasi NAT

Gunakan alur ini agar tidak menebak:

```
Client punya IP?
      ↓
Client punya gateway?
      ↓
Client bisa ping router?
      ↓
Router punya default route?
      ↓
Router bisa ping luar?
      ↓
NAT rule benar?
      ↓
Firewall forward mengizinkan?
      ↓
Client bisa ping luar?
      ↓
DNS bekerja?
```

E.13 Checklist NAT Berhasil

NAT dianggap berhasil jika:

- WAN router mendapat IP.
- Router punya *default route*.
- LAN router punya IP gateway.
- Client mendapat IP LAN.
- Client punya gateway ke router.
- Client bisa ping router.
- Router bisa ping IP luar.
- Rule NAT masquerade ada.
- `out-interface` benar.
- Client bisa ping IP luar.
- DNS client bekerja jika diuji dengan nama domain.
- Konfigurasi sudah diekspor.

Export konfigurasi:

```
/export file=R1-nat-final
```

E.14 Golden Rules NAT

- **NAT bukan pengganti routing.**
- **Route keluar harus ada sebelum NAT berguna.**
- **Masquerade cocok untuk WAN dinamis.**
- **`out-interface` harus mengarah ke WAN.**
- **Jika router tidak bisa keluar, client pasti tidak bisa keluar.**
- **Uji dulu dengan IP, baru uji DNS.**
- **VPC cukup untuk ping dan trace.**
- **Ubuntu VM lebih baik untuk uji DNS, curl, dan aplikasi.**
- **Setelah NAT berhasil, langsung export konfigurasi.**

Lampiran F. Template Firewall Aman

Lampiran ini berisi template **firewall dasar yang aman** untuk lab MikroTik. Fokusnya adalah melindungi router dan mengatur trafik yang melewati router.

Template ini mencakup:

- **input chain** untuk melindungi router.
- **forward chain** untuk mengatur trafik antarjaringan.
- **invalid** untuk membuang paket bermasalah.
- **established, related** untuk mengizinkan koneksi yang sudah sah.
- **LAN admin** sebagai sumber akses manajemen yang diizinkan.

F.1 Prinsip Firewall Aman

Firewall harus dibuat dengan urutan yang jelas:

1. **Izinkan koneksi yang sudah sah**.
2. **Buang paket invalid**.
3. **Izinkan akses admin dari LAN**.
4. **Izinkan ICMP untuk troubleshooting**.
5. **Izinkan trafik LAN ke WAN jika diperlukan**.
6. **Drop sisanya**.

Kalimat kunci:

> **Firewall yang baik bukan yang banyak rule, tetapi yang sedikit, jelas, terurut, dan bisa diuji.**

F.2 Asumsi Interface

Interface	Fungsi	Contoh IP
`ether1`	WAN / jaringan luar	DHCP
`ether2`	LAN Admin / LAN utama	`192.168.10.1/24`
`ether3`	LAN User / server / lab tambahan	`192.168.20.1/24`

Asumsi jaringan admin:

```
192.168.10.0/24
```

F.3 Template Address List LAN Admin

Gunakan **address list** agar rule firewall lebih rapi.

```
/ip firewall address-list add list=admin-lan
address=192.168.10.0/24 comment="LAN admin allowed to manage
router"
```

Jika hanya satu PC admin yang boleh akses router:

```
/ip firewall address-list add list=admin-lan
address=192.168.10.10 comment="Admin PC only"
```

Catatan:

- Untuk lab awal, boleh pakai seluruh subnet admin.
- Untuk jaringan nyata, lebih baik batasi ke IP admin tertentu.

F.4 Template Firewall Input Aman

Input chain mengatur trafik yang **menuju router**.

Contoh trafik input:

- SSH ke MikroTik.
- Ping ke MikroTik.
- DNS request ke MikroTik.
- Akses service router.

RouterOS v7/v6

```
/ip firewall filter add chain=input
connection-state=established,related action=accept
comment="INPUT: allow established, related"
/ip firewall filter add chain=input connection-state=invalid
action=drop comment="INPUT: drop invalid"
/ip firewall filter add chain=input
src-address-list=admin-lan action=accept comment="INPUT:
allow admin LAN"
/ip firewall filter add chain=input protocol=icmp
action=accept comment="INPUT: allow ICMP for troubleshooting"
/ip firewall filter add chain=input action=drop
comment="INPUT: drop all other traffic"
```

Makna rule:

Rule	Fungsi
`established,related`	mengizinkan koneksi yang sudah sah
`invalid`	membuang paket rusak/tidak valid
`admin-lan`	mengizinkan admin mengakses router
`icmp`	mengizinkan ping untuk troubleshooting
`drop all other`	menutup akses yang tidak perlu

Catatan penting:

> **Pastikan rule admin LAN ada sebelum rule drop terakhir.**
> Kalau tidak, akses ke router bisa terkunci.

F.5 Template Firewall Forward Aman

Forward chain mengatur trafik yang **melewati router**.

Contoh trafik forward:

- VPC ke Internet.
- Ubuntu VM ke Internet.
- LAN admin ke LAN server.
- LAN user ke WAN.
- Trafik antar-VLAN.

RouterOS v7/v6

```
/ip firewall filter add chain=forward
connection-state=established,related action=accept
comment="FORWARD: allow established, related"
/ip firewall filter add chain=forward
connection-state=invalid action=drop comment="FORWARD: drop
invalid"
/ip firewall filter add chain=forward in-interface=ether2
out-interface=ether1 action=accept comment="FORWARD: allow
LAN admin to WAN"
/ip firewall filter add chain=forward action=drop
comment="FORWARD: drop all other traffic"
```

Makna rule:

Rule	Fungsi
`established,related`	mengizinkan balasan koneksi sah
`invalid`	membuang paket tidak valid
`in-interface=ether2 out-interface=ether1`	mengizinkan LAN admin keluar ke WAN
`drop all other`	menutup trafik forward lain

Catatan:

- Jika ada LAN user di `ether3`, tambahkan rule khusus.
- Jika ada server yang harus diakses, tambahkan rule khusus sebelum drop.
- Jika ada VPN, tambahkan rule VPN sebelum drop.

F.6 Template Forward untuk Beberapa LAN

Contoh:

Interface	Fungsi	Subnet
ether2	LAN Admin	192.168.10.0/24
ether3	LAN User	192.168.20.0/24
ether4	LAN Tamu	192.168.30.0/24
ether1	WAN	DHCP

Template:

```
/ip firewall filter add chain=forward
connection-state=established,related action=accept
comment="FORWARD: allow established, related"
/ip firewall filter add chain=forward
connection-state=invalid action=drop comment="FORWARD: drop
invalid"

/ip firewall filter add chain=forward in-interface=ether2
out-interface=ether1 action=accept comment="FORWARD: admin to
WAN"
/ip firewall filter add chain=forward in-interface=ether3
out-interface=ether1 action=accept comment="FORWARD: user to
WAN"
/ip firewall filter add chain=forward in-interface=ether4
out-interface=ether1 action=accept comment="FORWARD: guest to
WAN"

/ip firewall filter add chain=forward action=drop
comment="FORWARD: drop all other"
```

Prinsip:

Izinkan jalur yang jelas. Drop jalur yang tidak jelas.

F.7 Template Firewall dengan Interface List

Untuk lab yang lebih rapi, gunakan **interface list**.

Buat Interface List

```
/interface list add name=WAN
/interface list add name=LAN
/interface list add name=ADMIN

/interface list member add list=WAN interface=ether1
/interface list member add list=LAN interface=ether2
/interface list member add list=LAN interface=ether3
/interface list member add list=ADMIN interface=ether2
```

Firewall Input

```
/ip firewall filter add chain=input
connection-state=established,related action=accept
comment="INPUT: allow established, related"
/ip firewall filter add chain=input connection-state=invalid
action=drop comment="INPUT: drop invalid"
/ip firewall filter add chain=input in-interface-list=ADMIN
src-address-list=admin-lan action=accept comment="INPUT:
allow admin management"
/ip firewall filter add chain=input protocol=icmp
action=accept comment="INPUT: allow ICMP"
/ip firewall filter add chain=input action=drop
comment="INPUT: drop all other"
```

Firewall Forward

```
/ip firewall filter add chain=forward
connection-state=established,related action=accept
comment="FORWARD: allow established, related"
/ip firewall filter add chain=forward
connection-state=invalid action=drop comment="FORWARD: drop
invalid"
/ip firewall filter add chain=forward in-interface-list=LAN
out-interface-list=WAN action=accept comment="FORWARD: LAN to
WAN"
/ip firewall filter add chain=forward action=drop
comment="FORWARD: drop all other"
```

Kelebihan interface list:

- rule lebih pendek,
- lebih mudah dibaca,
- mudah menambah interface LAN/WAN baru.

F.8 Template Izinkan DNS dari LAN ke Router

Jika router dipakai sebagai DNS resolver/cache untuk client LAN:

```
/ip firewall filter add chain=input
src-address=192.168.10.0/24 protocol=udp dst-port=53
action=accept comment="INPUT: allow DNS UDP from LAN"
/ip firewall filter add chain=input
src-address=192.168.10.0/24 protocol=tcp dst-port=53
action=accept comment="INPUT: allow DNS TCP from LAN"
```

Letakkan rule ini sebelum drop input.

Catatan:

Jangan membuka DNS resolver ke WAN.
DNS router sebaiknya hanya untuk LAN yang dikontrol.

F.9 Template Izinkan SSH dari Admin LAN

Jika manajemen router memakai SSH:

```
/ip firewall filter add chain=input
src-address-list=admin-lan protocol=tcp dst-port=22
action=accept comment="INPUT: allow SSH from admin LAN"
```

Letakkan sebelum drop input.

Agar lebih aman, batasi service SSH juga:

```
/ip service set ssh address=192.168.10.0/24
/ip service disable telnet,ftp,www
```

Catatan:

- Untuk lab, boleh izinkan dari satu subnet admin.
- Untuk lingkungan lebih serius, batasi ke IP admin tertentu.

F.10 Template Izinkan Port Forwarding ke Ubuntu Server

Jika Bab NAT lanjutan memakai port forwarding ke Ubuntu Server:

Contoh:

Server	IP	Port
Ubuntu Server	192.168.20.10	TCP 80

NAT:

```
/ip firewall nat add chain=dstnat in-interface=ether1
protocol=tcp dst-port=8080 action=dst-nat
to-addresses=192.168.20.10 to-ports=80 comment="NAT: forward
8080 to Ubuntu web"
```

Firewall forward:

```
/ip firewall filter add chain=forward
connection-state=established,related action=accept
comment="FORWARD: allow established, related"
/ip firewall filter add chain=forward
connection-state=invalid action=drop comment="FORWARD: drop
invalid"
/ip firewall filter add chain=forward protocol=tcp
dst-address=192.168.20.10 dst-port=80 action=accept
comment="FORWARD: allow web to Ubuntu"
/ip firewall filter add chain=forward in-interface=ether2
out-interface=ether1 action=accept comment="FORWARD: admin to
WAN"
/ip firewall filter add chain=forward action=drop
comment="FORWARD: drop all other"
```

Catatan:

Dst-NAT saja tidak cukup jika forward chain memblokir.
NAT mengubah alamat. Firewall menentukan boleh lewat atau tidak.

F.11 Template Firewall IPv6 Input Dasar

Jika IPv6 aktif, buat firewall IPv6. Jangan mengaktifkan IPv6 tanpa proteksi.

```
/ipv6 firewall filter add chain=input
connection-state=established,related action=accept
comment="IPv6 INPUT: allow established, related"
/ipv6 firewall filter add chain=input
connection-state=invalid action=drop comment="IPv6 INPUT:
drop invalid"
/ipv6 firewall filter add chain=input protocol=icmpv6
action=accept comment="IPv6 INPUT: allow ICMPv6"
/ipv6 firewall filter add chain=input in-interface=ether2
action=accept comment="IPv6 INPUT: allow LAN"
/ipv6 firewall filter add chain=input action=drop
comment="IPv6 INPUT: drop all other"
```

Catatan penting:

> **Jangan blok ICMPv6 sembarangan.**
> IPv6 membutuhkan ICMPv6 untuk fungsi penting seperti *Neighbor Discovery*.

F.12 Urutan Rule yang Disarankan

Urutan rule sangat penting.

Input Chain

1. allow established, related
2. drop invalid
3. allow admin LAN
4. allow ICMP
5. allow DNS dari LAN jika router jadi DNS
6. allow SSH dari admin jika diperlukan
7. drop all other

Forward Chain

1. allow established, related
2. drop invalid
3. allow LAN ke WAN
4. allow port forwarding / VPN / antar-VLAN yang diperlukan
5. drop all other

Kalimat kunci:

> **Firewall dibaca dari atas ke bawah. Rule pertama yang cocok akan dipakai.**

F.13 Verifikasi Firewall

Cek Rule

```
/ip firewall filter print
/ip firewall nat print
```

Cek Koneksi

```
/ip firewall connection print
```

Cek Log

```
/log print
```

Cek dari Router

```
/tool ping 192.168.10.10
/tool ping 1.1.1.1
/tool traceroute 1.1.1.1
```

Cek dari VPC

```
PC1> show ip
PC1> ping 192.168.10.1
PC1> ping 1.1.1.1
PC1> trace 1.1.1.1
```

Cek dari Ubuntu

```
ip addr
ip route
ping 192.168.10.1
ping 1.1.1.1
traceroute 1.1.1.1
curl http://192.168.20.10
```

F.14 Template Logging Drop

Aktifkan logging hanya untuk troubleshooting.

Log Drop Input

```
/ip firewall filter add chain=input action=drop log=yes
log-prefix="DROP-INPUT " comment="INPUT: drop with log"
```

Log Drop Forward

```
/ip firewall filter add chain=forward action=drop log=yes
log-prefix="DROP-FORWARD " comment="FORWARD: drop with log"
```

Catatan:

- Jangan logging semua trafik dalam waktu lama.
- Log terlalu ramai bisa menyulitkan analisis.
- Gunakan logging saat debugging, lalu rapikan.

F.15 Troubleshooting Firewall

Gejala	Kemungkinan	Cek	Solusi
Tidak bisa SSH ke router	Input drop terlalu awal	`/ip firewall filter print`	Izinkan admin LAN sebelum drop
VPC tidak bisa ping router	ICMP/input diblokir	`/log print`	Izinkan ICMP dari LAN
Client bisa ping router, tidak bisa keluar	Forward drop / NAT salah	`/ip firewall filter print`, `/ip firewall nat print`	Allow LAN to WAN dan cek NAT
Port forwarding gagal	Forward belum allow	`/ip firewall nat print`	Tambah rule forward ke server
DNS gagal	DNS ke router diblokir	`/ip firewall filter print`, `/ip dns print`	Izinkan port 53 dari LAN
IPv6 tidak jalan	ICMPv6 diblokir	`/ipv6 firewall filter print`	Izinkan ICMPv6
Rule tidak bekerja	Urutan salah	`/ip firewall filter print`	Pindahkan rule sebelum drop
Router terkunci	Drop terlalu ketat	akses console	Perbaiki dari console GNS3/VirtualBox

F.16 Template Firewall Aman Lengkap untuk Lab

Template ini bisa dipakai untuk lab dasar dengan:

- `ether1` = WAN
- `ether2` = LAN admin
- `192.168.10.0/24` = admin LAN

```
/ip firewall address-list add list=admin-lan
address=192.168.10.0/24 comment="Admin LAN"

/ip firewall filter add chain=input
connection-state=established,related action=accept
comment="INPUT: allow established, related"
/ip firewall filter add chain=input connection-state=invalid
action=drop comment="INPUT: drop invalid"
/ip firewall filter add chain=input
src-address-list=admin-lan in-interface=ether2 action=accept
comment="INPUT: allow admin LAN"
/ip firewall filter add chain=input protocol=icmp
action=accept comment="INPUT: allow ICMP"
/ip firewall filter add chain=input action=drop
comment="INPUT: drop all other"

/ip firewall filter add chain=forward
connection-state=established,related action=accept
comment="FORWARD: allow established, related"
/ip firewall filter add chain=forward
connection-state=invalid action=drop comment="FORWARD: drop
invalid"
/ip firewall filter add chain=forward in-interface=ether2
out-interface=ether1 action=accept comment="FORWARD: allow
LAN admin to WAN"
/ip firewall filter add chain=forward action=drop
comment="FORWARD: drop all other"

/export file=R1-firewall-safe
```

Catatan:

- Tambahkan rule DNS/SSH khusus jika ingin lebih spesifik.
- Tambahkan rule port forwarding sebelum drop forward jika ada server.
- Tambahkan rule VPN sebelum drop input/forward jika VPN digunakan.

F.17 Checklist Firewall Aman

Firewall dianggap siap jika:

- Rule `established,related` ada di bagian atas.
- Rule `invalid` di-drop.
- Admin LAN bisa akses router.
- Akses dari WAN ke router tidak terbuka sembarangan.
- LAN yang dibutuhkan bisa keluar ke WAN.
- Rule drop terakhir ada.
- NAT tetap berjalan.
- DNS dari LAN bekerja jika router jadi DNS.
- IPv6 punya firewall sendiri jika IPv6 aktif.
- Semua rule punya komentar.
- Konfigurasi sudah di-*export*:

```
/export file=R1-firewall-final
```

F.18 Golden Rules Firewall Aman

- **Amankan input dulu, baru forward.**
- **Jangan drop sebelum allow admin LAN.**
- **Rule firewall dibaca dari atas ke bawah.**
- **Established dan related biasanya diizinkan lebih awal.**
- **Invalid sebaiknya dibuang.**
- **Akses manajemen router hanya dari LAN admin.**
- **Forward LAN ke WAN harus eksplisit jika memakai drop akhir.**
- **Port forwarding butuh NAT dan rule forward.**
- **Jangan aktifkan IPv6 tanpa firewall IPv6.**
- **Gunakan log secukupnya, bukan untuk semua trafik.**
- **Setiap perubahan firewall harus diuji dari VPC dan Ubuntu.**
- **Sebelum eksperimen firewall, lakukan export konfigurasi.**

Lampiran G. Template L7 Firewall

Lampiran ini berisi template **Layer 7 Firewall** di MikroTik RouterOS. Tujuannya bukan menjadikan L7 sebagai solusi utama filtering, tetapi memberi pemahaman praktis tentang:

- cara membuat **Layer 7 matcher**,
- contoh rule sederhana,
- risiko beban CPU,
- batasan penggunaan,
- cara verifikasi,
- cara menonaktifkan rule jika router mulai berat.

G.1 Prinsip Dasar L7 Firewall

Layer 7 firewall mencoba membaca pola pada trafik aplikasi. Di RouterOS, L7 biasanya memakai **regexp** untuk mencocokkan pola tertentu pada trafik.

Contoh sederhana:

```
/ip firewall layer7-protocol add name=blok-contoh
regexp="^.+(contoh-domain).*$"
```

Kemudian dipakai pada firewall filter:

```
/ip firewall filter add chain=forward
layer7-protocol=blok-contoh action=drop comment="L7: blok
contoh-domain"
```

Makna:

- `layer7-protocol` membuat pola pencocokan.
- `regexp` adalah pola teks yang dicari.
- `chain=forward` berarti trafik yang melewati router diperiksa.
- `action=drop` berarti trafik yang cocok akan diblokir.

Catatan penting:

> **L7 bukan alat ajaib.**
> Banyak trafik modern terenkripsi, sehingga L7 tidak selalu bisa membaca isi trafik dengan akurat.

G.2 Kapan L7 Boleh Dipakai

Gunakan L7 hanya untuk:

- **lab pembelajaran**,
- trafik sederhana,
- pola yang sangat spesifik,
- jumlah client kecil,
- pengujian konsep firewall aplikasi,
- demonstrasi dampak CPU.

Contoh penggunaan aman:

- lab GNS3,
- VPC sebagai client uji,
- Ubuntu VM sebagai server/client uji,
- trafik HTTP sederhana di jaringan lokal.

G.3 Kapan L7 Sebaiknya Tidak Dipakai

Hindari L7 untuk:

- jaringan produksi besar,
- banyak client,
- trafik berat,
- semua trafik internet,
- blokir aplikasi modern terenkripsi,
- pengganti DNS filtering,
- pengganti proxy/filter aplikasi yang lebih sesuai,
- router dengan CPU kecil.

Catatan kritis:

> **Jangan memakai L7 untuk semua trafik.**
> Router bisa menjadi lambat, CPU tinggi, dan koneksi terganggu.

G.4 Risiko CPU

L7 matcher lebih berat dibanding filter biasa karena router harus memeriksa pola pada trafik.

Risiko utama:

- **CPU naik tinggi**.
- Latensi meningkat.
- Throughput turun.
- Router terasa lambat.
- Koneksi client terputus-putus.
- Firewall menjadi sulit diprediksi.
- Log dan connection tracking bisa lebih ramai.

Cek resource:

```
/system resource print
```

Cek traffic interface:

```
/interface monitor-traffic ether1
/interface monitor-traffic ether2
```

Cek koneksi:

```
/ip firewall connection print
```

Jika CPU tinggi setelah rule L7 aktif, segera nonaktifkan rule.

G.5 Template Topologi Uji L7

Topologi lab sederhana:

```
Ubuntu Client / VPC
        |
      ether2
   MikroTik CHR
      ether1
        |
   Jaringan luar / server uji
```

Atau uji lokal:

```
Ubuntu Client ---- MikroTik CHR ---- Ubuntu Server
```

Tujuan lab:

- client mencoba akses layanan,
- MikroTik memeriksa pola L7,
- rule L7 melakukan drop jika pola cocok,
- CPU dipantau.

G.6 Template L7 Paling Sederhana

Buat L7 Protocol

```
/ip firewall layer7-protocol add name=blok-contoh
regexp="^.+(contoh-domain).*$"
```

Tambahkan Rule Firewall

```
/ip firewall filter add chain=forward
layer7-protocol=blok-contoh action=drop comment="L7: blok
contoh-domain"
```

Verifikasi

```
/ip firewall layer7-protocol print
/ip firewall filter print
/system resource print
```

Catatan:

- Ini hanya contoh lab.
- Ganti `contoh-domain` dengan pola yang benar-benar ingin diuji.
- Jangan gunakan untuk semua trafik di jaringan besar.

G.7 Template L7 dengan Address List Client Uji

Agar L7 tidak memeriksa semua client, batasi hanya pada client tertentu.

Tambahkan Client Uji

```
/ip firewall address-list add list=l7-test-client
address=192.168.10.10 comment="Client uji L7"
```

Buat L7 Protocol

```
/ip firewall layer7-protocol add name=blok-contoh
regexp="^.+(contoh-domain).*$"
```

Rule L7 Terbatas ke Client Uji

```
/ip firewall filter add chain=forward
src-address-list=l7-test-client layer7-protocol=blok-contoh
action=drop comment="L7: blok contoh-domain untuk client uji"
```

Kelebihan:

- lebih aman untuk lab,
- beban CPU lebih terkendali,
- tidak mengganggu semua client.

Prinsip:

> **Batasi L7 ke client uji, bukan seluruh jaringan.**

G.8 Template L7 dengan Logging Terbatas

Logging berguna untuk melihat rule kena atau tidak. Namun logging bisa membuat router ramai jika trafik banyak.

Rule L7 dengan Log

```
/ip firewall filter add chain=forward
src-address-list=l7-test-client layer7-protocol=blok-contoh
action=drop log=yes log-prefix="L7-DROP " comment="L7: drop
dengan log terbatas"
```

Lihat Log

```
/log print
```

Catatan:

- Gunakan logging saat debugging.
- Setelah selesai, matikan logging.
- Jangan logging semua trafik.

G.9 Template Menonaktifkan L7

Jika CPU naik atau trafik terganggu, nonaktifkan rule L7.

Cari Rule

```
/ip firewall filter print
```

Disable Berdasarkan Komentar

```
/ip firewall filter disable [find comment="L7: blok
contoh-domain untuk client uji"]
```

Atau jika memakai komentar lain:

```
/ip firewall filter disable [find comment="L7: drop dengan
log terbatas"]
```

Disable L7 Protocol Tidak Menghapus Rule

```
/ip firewall layer7-protocol print
```

Untuk menghapus protocol:

```
/ip firewall layer7-protocol remove [find name=blok-contoh]
```

Catatan:

> **Nonaktifkan rule filter dulu. Hapus L7 protocol belakangan jika sudah yakin tidak dipakai.**

G.10 Template Uji dari VPC GNS3

VPC hanya cocok untuk uji dasar seperti ping dan trace. Untuk L7 berbasis HTTP, VPC kurang ideal.

Tetap bisa dipakai untuk memastikan jalur dasar:

```
PC1> show ip
PC1> ping 192.168.10.1
PC1> trace 1.1.1.1
```

Jika jalur dasar belum berjalan, jangan uji L7 dulu.

Checklist sebelum L7:

- VPC punya IP.
- VPC bisa ping gateway.
- Router bisa ping jaringan tujuan.
- NAT dan route berjalan jika keluar ke jaringan luar.

G.11 Template Uji dari Ubuntu VM

Ubuntu VM lebih cocok untuk uji L7 karena bisa menjalankan `curl`.

Cek IP dan Route

```
ip addr
ip route
ping 192.168.10.1
```

Uji HTTP

```
curl http://contoh-domain
```

Atau uji ke server lokal:

```
curl http://192.168.20.10
```

Cek DNS

```
dig contoh-domain
```

Catatan:

- Jika DNS gagal, itu bukan masalah L7.
- Jika ping gateway gagal, itu bukan masalah L7.
- Uji L7 dilakukan setelah konektivitas dasar benar.

G.12 Template Monitoring CPU Saat L7 Aktif

Sebelum mengaktifkan L7:

```
/system resource print
```

Setelah mengaktifkan L7:

```
/system resource print
/interface monitor-traffic ether1
/interface monitor-traffic ether2
/ip firewall connection print
```

Yang perlu diamati:

- CPU load,
- memory,
- jumlah koneksi,
- trafik interface,
- apakah client mulai lambat.

Jika CPU naik tidak wajar:

```
/ip firewall filter disable [find comment~"L7"]
```

Catatan:

> **Tidak ada rule L7 yang layak dipertahankan jika membuat router tidak stabil.**

G.13 Template L7 sebagai Demonstrasi, Bukan Solusi Utama

Gunakan L7 untuk menunjukkan konsep:

```
Client mengakses pola tertentu
        |
        v
Router membaca pola L7
        |
        v
Jika cocok → drop
Jika tidak cocok → lanjut
```

Namun untuk desain jaringan yang lebih sehat, lebih baik gunakan kombinasi:

- DNS policy,
- firewall berdasarkan IP/subnet,
- address list,
- proxy khusus jika dibutuhkan,
- segmentasi VLAN,
- autentikasi HotSpot,
- policy routing,
- monitoring trafik.

L7 hanyalah salah satu alat, bukan fondasi utama.

G.14 Contoh Lab L7 Sederhana

Topologi

```
Ubuntu Client
192.168.10.10
      |
   MikroTik
      |
Ubuntu Server
192.168.20.10
```

RouterOS

```
/ip firewall address-list add list=l7-test-client
address=192.168.10.10 comment="Client uji L7"

/ip firewall layer7-protocol add name=blok-web-lab
regexp="^.+(web-lab).*$"

/ip firewall filter add chain=forward
src-address-list=l7-test-client layer7-protocol=blok-web-lab
action=drop log=yes log-prefix="L7-DROP " comment="L7: blok
web-lab untuk client uji"
```

Ubuntu Client

```
curl http://web-lab
```

RouterOS Monitoring

```
/log print
/system resource print
/ip firewall filter print
```

Matikan Setelah Uji

```
/ip firewall filter disable [find comment="L7: blok web-lab
untuk client uji"]
```

G.15 Kesalahan Umum L7 Firewall

Kesalahan	Gejala	Solusi
L7 dipakai untuk semua trafik	CPU tinggi, jaringan lambat	Batasi ke client uji
Regex terlalu umum	Banyak trafik ikut terblokir	Buat pola spesifik
Logging semua drop L7	Log penuh	Logging sementara saja
Uji L7 sebelum routing benar	Hasil membingungkan	Cek IP, route, DNS dulu
Mengira L7 bisa baca semua trafik terenkripsi	Rule tidak efektif	Gunakan metode lain
Rule L7 diletakkan setelah drop umum	Rule tidak pernah kena	Periksa urutan firewall
Tidak memonitor CPU	Router lambat tanpa disadari	Cek `/system resource print`
L7 dipakai sebagai pengganti segmentasi	Kontrol jaringan lemah	Gunakan VLAN/firewall biasa

G.16 Checklist Sebelum Mengaktifkan L7

Sebelum mengaktifkan L7, pastikan:

- IP client benar.
- Gateway client benar.
- Routing berjalan.
- DNS berjalan jika uji memakai nama.
- NAT berjalan jika keluar ke jaringan luar.
- Firewall dasar tidak memblokir uji.
- Rule L7 dibatasi ke client tertentu.
- Komentar rule jelas.
- CPU baseline sudah dicatat.
- Siap perintah disable jika router berat.

G.17 Checklist Setelah Mengaktifkan L7

Setelah rule L7 aktif, cek:

```
/ip firewall filter print
/ip firewall layer7-protocol print
/log print
/system resource print
/ip firewall connection print
```

Pastikan:

- rule berada di urutan yang benar,
- rule kena trafik yang dimaksud,
- CPU tidak naik berlebihan,
- client lain tidak terganggu,
- log tidak terlalu ramai.

G.18 Template Perintah Cepat L7

Buat Client Uji

```
/ip firewall address-list add list=l7-test-client
address=192.168.10.10 comment="Client uji L7"
```

Buat Pattern

```
/ip firewall layer7-protocol add name=blok-contoh
regexp="^.+(contoh-domain).*$"
```

Aktifkan Drop Terbatas

```
/ip firewall filter add chain=forward
src-address-list=l7-test-client layer7-protocol=blok-contoh
action=drop comment="L7: blok contoh-domain untuk client uji"
```

Aktifkan Drop + Log

```
/ip firewall filter add chain=forward
src-address-list=l7-test-client layer7-protocol=blok-contoh
action=drop log=yes log-prefix="L7-DROP " comment="L7: drop
dengan log terbatas"
```

Cek

```
/ip firewall layer7-protocol print
/ip firewall filter print
/log print
/system resource print
```

Matikan

```
/ip firewall filter disable [find comment="L7: blok
contoh-domain untuk client uji"]
```

Hapus

```
/ip firewall filter remove [find comment="L7: blok
contoh-domain untuk client uji"]
/ip firewall layer7-protocol remove [find name=blok-contoh]
```

G.19 Golden Rules L7 Firewall

- **Gunakan L7 hanya untuk lab atau trafik sangat spesifik.**
- **Jangan memakai L7 untuk semua trafik.**
- **Batasi L7 ke client uji dengan address list.**
- **Pantau CPU sebelum dan sesudah rule aktif.**
- **Jangan mengandalkan L7 untuk trafik terenkripsi.**
- **Jangan logging semua trafik terlalu lama.**
- **Pastikan IP, route, DNS, NAT, dan firewall dasar sudah benar sebelum uji L7.**
- **Gunakan komentar rule agar mudah dinonaktifkan.**
- **Jika router melambat, disable L7 lebih dulu.**
- **L7 adalah alat belajar konsep inspeksi aplikasi, bukan solusi utama desain jaringan.**

Lampiran H. Template HotSpot

Lampiran ini berisi template cepat untuk membuat **HotSpot dasar MikroTik** di lab VirtualBox/GNS3. Template ini mencakup:

- **IP Pool**
- **DHCP Server**
- **HotSpot Profile**
- **HotSpot Server**
- **User HotSpot**
- **Walled Garden**
- Verifikasi dari **VPC GNS3** dan **Ubuntu VM**

Catatan penting:

> **CHR cocok untuk melatih HotSpot, DHCP, NAT, dan alur autentikasi.**
> **CHR tidak mensimulasikan radio wireless fisik.**
> Untuk praktik SSID, channel, band, dan radio Wi-Fi, gunakan perangkat MikroTik fisik yang memiliki interface wireless.

H.1 Topologi Dasar HotSpot

```
NAT / WAN
   |
 ether1
MikroTik CHR
 ether3
   |
HotSpot LAN
 /       \
VPC     Ubuntu VM
```

Asumsi:

Komponen	Interface	IP / Fungsi
WAN	`ether1`	DHCP Client / jaringan luar
HotSpot LAN	`ether3`	`192.168.30.1/24`
Pool HotSpot	-	`192.168.30.100-192.168.30.200`
VPC	eth0	DHCP
Ubuntu VM	enp0s3	DHCP
DNS HotSpot	-	`192.168.30.1`

H.2 Template HotSpot Dasar RouterOS v7/v6

Gunakan template ini untuk membuat HotSpot dasar.

```
/system identity set name=R1-HOTSPOT

/ip address add address=192.168.30.1/24 interface=ether3
comment="Gateway HotSpot"

/ip pool add name=pool-hotspot
ranges=192.168.30.100-192.168.30.200

/ip dhcp-server add name=dhcp-hotspot interface=ether3
address-pool=pool-hotspot disabled=no comment="DHCP HotSpot"

/ip dhcp-server network add address=192.168.30.0/24
gateway=192.168.30.1 dns-server=192.168.30.1 comment="HotSpot
Network"

/ip dns set allow-remote-requests=yes servers=1.1.1.1

/ip hotspot profile add name=hsprof
hotspot-address=192.168.30.1 dns-name=login.local

/ip hotspot add name=hotspot1 interface=ether3
address-pool=pool-hotspot profile=hsprof disabled=no

/ip hotspot user add name=user1 password=pass123

/export file=R1-hotspot-basic
```

H.3 Template HotSpot dengan WAN dan NAT

Jika client HotSpot perlu keluar ke jaringan luar, tambahkan **DHCP Client WAN** dan **NAT masquerade**.

```
/system identity set name=R1-HOTSPOT-NAT

/ip dhcp-client add interface=ether1 disabled=no comment="WAN
DHCP Client"

/ip address add address=192.168.30.1/24 interface=ether3
comment="Gateway HotSpot"

/ip pool add name=pool-hotspot
ranges=192.168.30.100-192.168.30.200

/ip dhcp-server add name=dhcp-hotspot interface=ether3
address-pool=pool-hotspot disabled=no comment="DHCP HotSpot"

/ip dhcp-server network add address=192.168.30.0/24
gateway=192.168.30.1 dns-server=192.168.30.1 comment="HotSpot
Network"

/ip dns set allow-remote-requests=yes servers=1.1.1.1

/ip hotspot profile add name=hsprof
hotspot-address=192.168.30.1 dns-name=login.local

/ip hotspot add name=hotspot1 interface=ether3
address-pool=pool-hotspot profile=hsprof disabled=no

/ip hotspot user add name=user1 password=pass123

/ip firewall nat add chain=srcnat out-interface=ether1
action=masquerade comment="NAT HotSpot to WAN"

/export file=R1-hotspot-nat
```

Catatan:

- `ether1` = WAN.
- `ether3` = jaringan HotSpot.
- Sesuaikan jika interface berbeda.

H.4 Penjelasan Komponen HotSpot

Komponen	Fungsi
`/ip pool`	Rentang IP yang dibagikan ke client HotSpot
`/ip dhcp-server`	Memberi IP otomatis ke client
`/ip dhcp-server network`	Memberi gateway dan DNS ke client
`/ip dns`	Mengaktifkan router sebagai DNS untuk client
`/ip hotspot profile`	Template layanan HotSpot
`/ip hotspot`	HotSpot server pada interface tertentu
`/ip hotspot user`	Akun login HotSpot
`/ip firewall nat`	Mengizinkan client HotSpot keluar ke WAN

H.5 Template User HotSpot

User Dasar

```
/ip hotspot user add name=user1 password=pass123
```

Beberapa User

```
/ip hotspot user add name=siswa01 password=belajar01
/ip hotspot user add name=siswa02 password=belajar02
/ip hotspot user add name=tamu01 password=tamu123
```

User dengan Limit Waktu

```
/ip hotspot user add name=tamu1 password=tamu123
limit-uptime=1h
```

User dengan Profile Khusus

```
/ip hotspot user profile add name=profile-tamu
rate-limit=2M/2M shared-users=1
/ip hotspot user add name=tamu1 password=tamu123
profile=profile-tamu
```

Makna:

- `rate-limit=2M/2M` membatasi upload/download.
- `shared-users=1` membatasi satu akun hanya dipakai satu perangkat.
- `limit-uptime=1h` membatasi total waktu pakai.

Catatan:

> **Untuk lab awal, gunakan user sederhana dulu.**
> Setelah login berhasil, baru uji limit dan profile.

H.6 Template Profile HotSpot

Profile Dasar

```
/ip hotspot profile add name=hsprof
hotspot-address=192.168.30.1 dns-name=login.local
```

Profile dengan Nama DNS Login

```
/ip hotspot profile add name=hsprof-lab
hotspot-address=192.168.30.1 dns-name=hotspot.lab
```

Cek Profile

```
/ip hotspot profile print
```

Catatan:

- `dns-name` adalah nama lokal untuk portal login.
- Pastikan DNS router bekerja.
- Untuk lab, nama seperti `login.local` atau `hotspot.lab` cukup.

H.7 Template Walled Garden

Walled garden adalah daftar tujuan yang boleh diakses sebelum user login.

Contoh fungsi:

- halaman bantuan,
- portal informasi,
- alamat server lokal,
- halaman syarat penggunaan.

Izinkan IP Server Lokal Sebelum Login

Misal server informasi berada di:

```
192.168.30.10
```

Rule:

```
/ip hotspot walled-garden ip add dst-address=192.168.30.10
action=accept comment="Allow local info server before login"
```

Izinkan Web Lokal Sebelum Login

Jika server lokal memakai port HTTP:

```
/ip hotspot walled-garden ip add dst-address=192.168.30.10
protocol=tcp dst-port=80 action=accept comment="Allow local
web before login"
```

Cek Walled Garden

```
/ip hotspot walled-garden print
/ip hotspot walled-garden ip print
```

Catatan:

> **Walled garden harus dibatasi.**
> Jangan membuka terlalu banyak tujuan sebelum user login.

H.8 Template Lab HotSpot + VPC GNS3

VPC cocok untuk uji:

- DHCP,
- IP,
- gateway,
- ping.

VPC

```
PC1> ip dhcp
PC1> show ip
PC1> ping 192.168.30.1
```

Target:

- VPC mendapat IP dari `pool-hotspot`.
- Gateway VPC adalah `192.168.30.1`.
- VPC bisa ping gateway.

Catatan:

> **VPC tidak ideal untuk uji portal login berbasis web.**
> Gunakan Ubuntu VM untuk uji halaman login.

H.9 Template Lab HotSpot + Ubuntu VM

Ubuntu VM dipakai untuk uji portal login.

Cek IP

```
ip addr
ip route
```

Ping Gateway

```
ping 192.168.30.1
```

Uji DNS

```
dig login.local
```

Uji Portal Login

Buka browser di Ubuntu VM, lalu akses:

```
http://login.local
```

atau akses halaman web biasa agar diarahkan ke portal login.

Login memakai:

```
Username : user1
Password : pass123
```

Cek User Aktif di RouterOS

```
/ip hotspot active print
```

Cek Host HotSpot

```
/ip hotspot host print
```

H.10 Template Verifikasi HotSpot di RouterOS

Cek IP

```
/ip address print
```

Pastikan:

```
192.168.30.1/24 interface=ether3
```

Cek Pool

```
/ip pool print
```

Pastikan pool:

```
192.168.30.100-192.168.30.200
```

Cek DHCP Server

```
/ip dhcp-server print
/ip dhcp-server network print
/ip dhcp-server lease print
```

Cek DNS

```
/ip dns print
```

Pastikan:

```
allow-remote-requests=yes
```

Cek HotSpot

```
/ip hotspot print
/ip hotspot profile print
/ip hotspot user print
/ip hotspot active print
/ip hotspot host print
```

Cek NAT Jika Client Perlu Keluar

```
/ip firewall nat print
/ip route print
/tool ping 1.1.1.1
```

H.11 Template Firewall Minimum untuk HotSpot

Jika firewall input sudah ketat, izinkan akses dari jaringan HotSpot ke router untuk kebutuhan DHCP/DNS/HotSpot.

Contoh sederhana untuk lab:

```
/ip firewall filter add chain=input
connection-state=established,related action=accept
comment="INPUT: allow established related"
/ip firewall filter add chain=input connection-state=invalid
action=drop comment="INPUT: drop invalid"
/ip firewall filter add chain=input in-interface=ether3
action=accept comment="INPUT: allow HotSpot LAN to router"
/ip firewall filter add chain=input protocol=icmp
action=accept comment="INPUT: allow ICMP"
/ip firewall filter add chain=input action=drop
comment="INPUT: drop other"
```

Jika memakai forward chain ketat:

```
/ip firewall filter add chain=forward
connection-state=established,related action=accept
comment="FORWARD: allow established related"
/ip firewall filter add chain=forward
connection-state=invalid action=drop comment="FORWARD: drop
invalid"
/ip firewall filter add chain=forward in-interface=ether3
out-interface=ether1 action=accept comment="FORWARD: allow
HotSpot to WAN"
/ip firewall filter add chain=forward action=drop
comment="FORWARD: drop other"
```

Catatan:

- Rule ini untuk lab.
- Untuk produksi, akses HotSpot harus dibatasi lebih detail.

H.12 Template Troubleshooting HotSpot

Masalah	Kemungkinan	Cek	Solusi
Client tidak dapat IP	DHCP salah interface / pool salah	`/ip dhcp-server print`	Cek DHCP di `ether3`
Client dapat IP tapi tidak bisa ping router	IP/gateway salah	`show ip`, `/ip address print`	Cek gateway `192.168.30.1`
Portal login tidak muncul	DNS/HotSpot profile salah	`/ip hotspot profile print`	Cek `dns-name` dan DNS
User tidak bisa login	User/password salah	`/ip hotspot user print`	Buat ulang user
Login berhasil tapi tidak bisa keluar	NAT/default route belum ada	`/ip route print`, `/ip firewall nat print`	Tambah NAT masquerade
Walled garden tidak jalan	Rule salah IP/port	`/ip hotspot walled-garden ip print`	Sesuaikan tujuan
VPC tidak bisa buka portal	VPC bukan browser client	-	Gunakan Ubuntu VM
HotSpot aktif di interface salah	Client tidak diarahkan login	`/ip hotspot print`	Pastikan interface `ether3`

H.13 Template Reset HotSpot Lab

Jika ingin mengulang konfigurasi HotSpot dari nol, export dulu:

```
/export file=backup-sebelum-reset-hotspot
```

Hapus komponen HotSpot:

```
/ip hotspot remove [find]
/ip hotspot profile remove [find name=hsprof]
/ip hotspot user remove [find]
/ip hotspot walled-garden remove [find]
/ip hotspot walled-garden ip remove [find]
```

Hapus DHCP khusus HotSpot jika perlu:

```
/ip dhcp-server remove [find name=dhcp-hotspot]
/ip dhcp-server network remove [find address=192.168.30.0/24]
/ip pool remove [find name=pool-hotspot]
```

Hapus IP interface jika ingin mulai ulang:

```
/ip address remove [find address="192.168.30.1/24"]
```

Catatan:

> **Hati-hati memakai `remove [find]`.**
> Pastikan hanya lab yang sedang dihapus, bukan konfigurasi penting lain.

H.14 Template Dokumentasi HotSpot

Gunakan tabel ini untuk dokumentasi.

Item	Nilai
Nama router	
Interface HotSpot	
IP HotSpot	
Pool DHCP	
DNS login	
Nama HotSpot	
Profile	
User uji	
Password uji	
Walled garden	
Status uji VPC	
Status uji Ubuntu VM	
File export	

Contoh:

Item	Nilai
Nama router	`R1-HOTSPOT`
Interface HotSpot	`ether3`
IP HotSpot	`192.168.30.1/24`
Pool DHCP	`192.168.30.100-192.168.30.200`
DNS login	`login.local`
Nama HotSpot	`hotspot1`
Profile	`hsprof`
User uji	`user1`
Password uji	`pass123`
Walled garden	`192.168.30.10:80`
Status uji VPC	DHCP dan ping gateway berhasil
Status uji Ubuntu VM	Login portal berhasil
File export	`R1-hotspot-basic.rsc`

H.15 Checklist HotSpot Berhasil

HotSpot dianggap berhasil jika:

- Interface HotSpot punya IP.
- DHCP Server HotSpot aktif.
- Client mendapat IP dari pool.
- Client punya gateway `192.168.30.1`.
- DNS router aktif.
- HotSpot server aktif.
- User HotSpot tersedia.
- VPC bisa ping gateway.
- Ubuntu VM bisa membuka portal login.
- Login user berhasil.
- `/ip hotspot active print` menampilkan user aktif.
- Jika perlu akses keluar, NAT dan default route berjalan.
- Konfigurasi sudah diekspor.

  Export:

  ```
  /export file=R1-hotspot-final
  ```

H.16 Golden Rules HotSpot

- **HotSpot butuh IP, DHCP, DNS, profile, server, dan user.**
- **VPC cocok untuk uji DHCP dan ping.**
- **Ubuntu VM lebih cocok untuk uji portal login.**
- **CHR bisa melatih HotSpot, tetapi tidak menggantikan radio wireless fisik.**
- **Jangan buka walled garden terlalu luas.**
- **User sederhana dulu, limit belakangan.**
- **Jika login berhasil tetapi Internet gagal, cek NAT dan default route.**
- **Jika portal tidak muncul, cek DNS dan profile HotSpot.**
- **Setelah berhasil, simpan export konfigurasi.**

Lampiran I. Template VPN

WireGuard v7, L2TP/IPsec v6/v7

Lampiran ini berisi template cepat untuk membuat **VPN site-to-site** dan **remote access** sederhana pada MikroTik RouterOS.

Fokus utama:

- **WireGuard untuk RouterOS v7**
- **L2TP/IPsec untuk RouterOS v6/v7**
- Routing lewat VPN
- Firewall untuk VPN
- Troubleshooting dasar

WireGuard di RouterOS memakai konsep **interface**, **listen port**, **public/private key**, **peer**, **endpoint**, dan **allowed-address**. MikroTik menjelaskan bahwa *allowed address* menentukan alamat atau subnet yang dapat diarahkan melalui peer WireGuard. (MikroTik Help) Untuk RouterOS v6/v7, L2TP/IPsec tetap menjadi opsi umum; MikroTik menjelaskan bahwa L2TP dapat dipakai dengan IPsec untuk mengenkripsi trafik tunnel, sedangkan IPsec adalah kumpulan protokol untuk mengamankan pertukaran paket pada jaringan IP/IPv6 yang tidak terlindungi. (MikroTik Help)

I.1 Prinsip Dasar VPN

VPN dipakai untuk membuat **tunnel aman** melalui jaringan yang tidak selalu kita kendalikan.

Contoh:

```
Site A LAN ---- R1 ==== VPN Tunnel ==== R2 ---- Site B LAN
```

VPN yang baik harus punya:

- **tunnel** yang aktif,
- **routing** ke subnet lawan,
- **firewall** yang mengizinkan trafik VPN,
- **allowed-address** atau policy yang benar,
- dokumentasi key, port, IP, dan subnet.

Kalimat kunci:

> **VPN bukan hanya “tunnel up”. VPN berhasil jika subnet di kedua sisi bisa saling terhubung dengan aman.**

Bagian A — WireGuard RouterOS v7

I.2 Topologi WireGuard Site-to-Site

```
LAN-A                   WAN Simulasi            LAN-B
192.168.10.0/24         172.16.99.0/24          192.168.20.0/24

VPC-A ---------- R1 ----------------- R2 ------- VPC-B
```

IP Plan

Perangkat	Interface	IP
R1	LAN-A	192.168.10.1/24
R1	WAN-SIM	172.16.99.1/24
R2	WAN-SIM	172.16.99.2/24
R2	LAN-B	192.168.20.1/24
R1 WireGuard	wg-site	10.10.10.1/30
R2 WireGuard	wg-site	10.10.10.2/30
VPC-A	eth0	192.168.10.10/24
VPC-B	eth0	192.168.20.10/24

I.3 Konfigurasi Dasar IP

R1 — RouterOS v7/v6

```
/system identity set name=R1-SITE-A
/ip address add address=192.168.10.1/24 interface=ether1
comment="LAN-A"
/ip address add address=172.16.99.1/24 interface=ether2
comment="WAN-SIM"
```

R2 — RouterOS v7/v6

```
/system identity set name=R2-SITE-B
/ip address add address=172.16.99.2/24 interface=ether1
comment="WAN-SIM"
/ip address add address=192.168.20.1/24 interface=ether2
comment="LAN-B"
```

VPC-A

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> ping 192.168.10.1
```

VPC-B

```
PC2> ip 192.168.20.10/24 192.168.20.1
PC2> ping 192.168.20.1
```

Uji Antarrouter

Dari R1:

```
/tool ping 172.16.99.2
```

Dari R2:

```
/tool ping 172.16.99.1
```

Catatan:

> **Jangan lanjut ke VPN jika IP WAN antara dua router belum bisa saling ping.**

I.4 Membuat Interface WireGuard

R1 — RouterOS v7

```
/interface wireguard add name=wg-site listen-port=13231
/ip address add address=10.10.10.1/30 interface=wg-site
comment="WG tunnel to R2"
/interface wireguard print detail
```

Catat **public-key R1**.

R2 — RouterOS v7

```
/interface wireguard add name=wg-site listen-port=13231
/ip address add address=10.10.10.2/30 interface=wg-site
comment="WG tunnel to R1"
/interface wireguard print detail
```

Catat **public-key R2**.

Prinsip:

Public key R1 dimasukkan ke peer di R2.
Public key R2 dimasukkan ke peer di R1.

I.5 Membuat Peer WireGuard

Misal:

PUBLIC_KEY_R1 = public key dari R1
PUBLIC_KEY_R2 = public key dari R2

R1 — RouterOS v7

```
/interface wireguard peers add interface=wg-site
public-key="PUBLIC_KEY_R2" endpoint-address=172.16.99.2
endpoint-port=13231
allowed-address=10.10.10.2/32,192.168.20.0/24
persistent-keepalive=25 comment="Peer R2 Site-B"

/ip route add dst-address=192.168.20.0/24 gateway=10.10.10.2
comment="Route to LAN-B via WireGuard"
```

R2 — RouterOS v7

```
/interface wireguard peers add interface=wg-site
public-key="PUBLIC_KEY_R1" endpoint-address=172.16.99.1
endpoint-port=13231
allowed-address=10.10.10.1/32,192.168.10.0/24
persistent-keepalive=25 comment="Peer R1 Site-A"

/ip route add dst-address=192.168.10.0/24 gateway=10.10.10.1
comment="Route to LAN-A via WireGuard"
```

Makna:

Parameter	Fungsi
`interface=wg-site`	interface WireGuard yang dipakai
`public-key`	identitas peer lawan
`endpoint-address`	IP WAN peer lawan
`endpoint-port`	port WireGuard peer lawan
`allowed-address`	IP tunnel peer dan subnet di belakang peer
`persistent-keepalive`	menjaga koneksi tetap aktif pada beberapa skenario NAT
`ip route`	mengarahkan subnet lawan lewat tunnel

I.6 Firewall WireGuard

Jika firewall input aktif, buka port WireGuard.

R1 dan R2

```
/ip firewall filter add chain=input protocol=udp
dst-port=13231 action=accept comment="INPUT: allow WireGuard"
```

Letakkan rule ini **sebelum** rule drop input.

Jika forward chain ketat, izinkan trafik antar-LAN.

R1

```
/ip firewall filter add chain=forward
src-address=192.168.10.0/24 dst-address=192.168.20.0/24
action=accept comment="FORWARD: LAN-A to LAN-B via VPN"
/ip firewall filter add chain=forward
src-address=192.168.20.0/24 dst-address=192.168.10.0/24
action=accept comment="FORWARD: LAN-B to LAN-A via VPN"
```

R2

```
/ip firewall filter add chain=forward
src-address=192.168.20.0/24 dst-address=192.168.10.0/24
action=accept comment="FORWARD: LAN-B to LAN-A via VPN"
/ip firewall filter add chain=forward
src-address=192.168.10.0/24 dst-address=192.168.20.0/24
action=accept comment="FORWARD: LAN-A to LAN-B via VPN"
```

Catatan:

> **WireGuard bisa handshake, tetapi LAN tetap gagal jika route atau firewall forward salah.**

I.7 Verifikasi WireGuard

RouterOS v7

```
/interface wireguard print
/interface wireguard peers print detail
/ip address print
/ip route print
/ip firewall filter print
```

Uji dari R1

```
/tool ping 10.10.10.2
/tool ping 192.168.20.1
/tool ping 192.168.20.10
```

Uji dari R2

```
/tool ping 10.10.10.1
/tool ping 192.168.10.1
/tool ping 192.168.10.10
```

Uji dari VPC-A

```
PC1> ping 192.168.20.10
PC1> trace 192.168.20.10
```

Uji dari VPC-B

```
PC2> ping 192.168.10.10
PC2> trace 192.168.10.10
```

I.8 Template WireGuard Lengkap — R1

```
/system identity set name=R1-SITE-A

/ip address add address=192.168.10.1/24 interface=ether1
comment="LAN-A"
/ip address add address=172.16.99.1/24 interface=ether2
comment="WAN-SIM"

/interface wireguard add name=wg-site listen-port=13231
/ip address add address=10.10.10.1/30 interface=wg-site
comment="WG tunnel to R2"

/interface wireguard peers add interface=wg-site
public-key="PUBLIC_KEY_R2" endpoint-address=172.16.99.2
endpoint-port=13231
allowed-address=10.10.10.2/32,192.168.20.0/24
persistent-keepalive=25 comment="Peer R2"

/ip route add dst-address=192.168.20.0/24 gateway=10.10.10.2
comment="Route to LAN-B"

/ip firewall filter add chain=input protocol=udp
dst-port=13231 action=accept comment="INPUT: allow WireGuard"

/export file=R1-wireguard
```

I.9 Template WireGuard Lengkap — R2

```
/system identity set name=R2-SITE-B

/ip address add address=172.16.99.2/24 interface=ether1
comment="WAN-SIM"
/ip address add address=192.168.20.1/24 interface=ether2
comment="LAN-B"

/interface wireguard add name=wg-site listen-port=13231
/ip address add address=10.10.10.2/30 interface=wg-site
comment="WG tunnel to R1"

/interface wireguard peers add interface=wg-site
public-key="PUBLIC_KEY_R1" endpoint-address=172.16.99.1
endpoint-port=13231
allowed-address=10.10.10.1/32,192.168.10.0/24
persistent-keepalive=25 comment="Peer R1"

/ip route add dst-address=192.168.10.0/24 gateway=10.10.10.1
comment="Route to LAN-A"

/ip firewall filter add chain=input protocol=udp
dst-port=13231 action=accept comment="INPUT: allow WireGuard"

/export file=R2-wireguard
```

Bagian B — L2TP/IPsec RouterOS v6/v7

I.10 Kapan Memakai L2TP/IPsec

Gunakan L2TP/IPsec jika:

- memakai RouterOS v6,
- perangkat belum mendukung WireGuard native,
- butuh kompatibilitas dengan perangkat lama,
- ingin memahami VPN berbasis PPP + IPsec.

Catatan:

> **Untuk RouterOS v7, WireGuard lebih ringkas untuk lab.**
> **Untuk RouterOS v6, L2TP/IPsec adalah pilihan pembanding yang lebih umum.**

I.11 Topologi L2TP/IPsec Site-to-Site Sederhana

```
LAN-A                 WAN Simulasi              LAN-B
192.168.10.0/24     172.16.99.0/24              192.168.20.0/24

VPC-A ---- R1 L2TP Server -------- R2 L2TP Client ---- VPC-B
```

IP Plan

Perangkat	Interface	IP
R1	WAN-SIM	`172.16.99.1/24`
R1	LAN-A	`192.168.10.1/24`
R2	WAN-SIM	`172.16.99.2/24`
R2	LAN-B	`192.168.20.1/24`
L2TP R1	local-address	`10.20.20.1`
L2TP R2	remote-address	`10.20.20.2`

I.12 L2TP/IPsec Server — R1

RouterOS v6/v7

```
/system identity set name=R1-L2TP-SERVER

/ip address add address=192.168.10.1/24 interface=ether1
comment="LAN-A"
/ip address add address=172.16.99.1/24 interface=ether2
comment="WAN-SIM"

/interface l2tp-server server set enabled=yes use-ipsec=yes
ipsec-secret=SecretIPsecKuat authentication=mschap2

/ppp secret add name=site-b password=PasswordL2TPKuat
service=l2tp local-address=10.20.20.1
remote-address=10.20.20.2 comment="L2TP peer Site-B"

/ip route add dst-address=192.168.20.0/24 gateway=10.20.20.2
comment="Route to LAN-B via L2TP"

/export file=R1-l2tp-server
```

I.13 L2TP/IPsec Client — R2

RouterOS v6/v7

```
/system identity set name=R2-L2TP-CLIENT

/ip address add address=172.16.99.2/24 interface=ether1
comment="WAN-SIM"
/ip address add address=192.168.20.1/24 interface=ether2
comment="LAN-B"

/interface l2tp-client add name=l2tp-to-site-a
connect-to=172.16.99.1 user=site-b password=PasswordL2TPKuat
use-ipsec=yes ipsec-secret=SecretIPsecKuat disabled=no

/ip route add dst-address=192.168.10.0/24 gateway=10.20.20.1
comment="Route to LAN-A via L2TP"

/export file=R2-l2tp-client
```

Catatan keamanan:

- Ganti `SecretIPsecKuat`.
- Ganti `PasswordL2TPKuat`.
- Jangan memakai contoh password di jaringan nyata.
- Simpan credential dengan aman.

I.14 Firewall L2TP/IPsec

Jika firewall input aktif pada R1, buka port yang diperlukan untuk L2TP/IPsec.

R1 — RouterOS v6/v7

```
/ip firewall filter add chain=input protocol=udp dst-port=500
action=accept comment="INPUT: allow IPsec IKE"
/ip firewall filter add chain=input protocol=udp
dst-port=4500 action=accept comment="INPUT: allow IPsec
NAT-T"
/ip firewall filter add chain=input protocol=udp
dst-port=1701 action=accept comment="INPUT: allow L2TP"
/ip firewall filter add chain=input protocol=ipsec-esp
action=accept comment="INPUT: allow IPsec ESP"
```

Letakkan sebelum rule drop input.

Jika forward chain ketat:

```
/ip firewall filter add chain=forward
src-address=192.168.10.0/24 dst-address=192.168.20.0/24
action=accept comment="FORWARD: LAN-A to LAN-B via L2TP"
/ip firewall filter add chain=forward
src-address=192.168.20.0/24 dst-address=192.168.10.0/24
action=accept comment="FORWARD: LAN-B to LAN-A via L2TP"
```

Catatan:

> **VPN membutuhkan port input untuk membentuk tunnel dan forward rule untuk melewatkan trafik LAN.**

I.15 Verifikasi L2TP/IPsec

R1

```
/interface l2tp-server print
/ppp active print
/ip route print
/log print
```

R2

```
/interface l2tp-client print detail
/ip route print
/log print
```

Uji dari R1

```
/tool ping 10.20.20.2
/tool ping 192.168.20.1
/tool ping 192.168.20.10
```

Uji dari R2

```
/tool ping 10.20.20.1
/tool ping 192.168.10.1
/tool ping 192.168.10.10
```

Uji VPC-A

```
PC1> ping 192.168.20.10
PC1> trace 192.168.20.10
```

Uji VPC-B

```
PC2> ping 192.168.10.10
PC2> trace 192.168.10.10
```

Bagian C — Remote Access L2TP/IPsec

I.16 Template Remote Access L2TP/IPsec

Jika admin ingin masuk dari luar ke router memakai akun VPN.

RouterOS v6/v7 — Server

```
/ip pool add name=pool-vpn ranges=10.30.30.10-10.30.30.50

/interface l2tp-server server set enabled=yes use-ipsec=yes
ipsec-secret=SecretIPsecKuat authentication=mschap2
default-profile=default

/ppp profile add name=profile-vpn local-address=10.30.30.1
remote-address=pool-vpn dns-server=192.168.10.1
use-encryption=yes

/ppp secret add name=adminvpn password=PasswordVPNKuat
service=l2tp profile=profile-vpn

/ip firewall filter add chain=input protocol=udp dst-port=500
action=accept comment="INPUT: allow IPsec IKE"
/ip firewall filter add chain=input protocol=udp
dst-port=4500 action=accept comment="INPUT: allow IPsec
NAT-T"
/ip firewall filter add chain=input protocol=udp
dst-port=1701 action=accept comment="INPUT: allow L2TP"
/ip firewall filter add chain=input protocol=ipsec-esp
action=accept comment="INPUT: allow IPsec ESP"
```

Catatan:

- Remote access lebih sensitif.
- Batasi user.
- Gunakan password kuat.
- Catat siapa boleh masuk.

Bagian D — Troubleshooting VPN

I.17 Troubleshooting WireGuard

Gejala	Kemungkinan	Cek	Solusi
Handshake tidak muncul	Public key salah	`/interface wireguard peers print detail`	Tukar public key dengan benar
Handshake tidak muncul	Endpoint salah	cek `endpoint-address`	Sesuaikan IP WAN peer
Handshake tidak muncul	Port UDP diblokir	`/ip firewall filter print`	Buka UDP 13231
IP tunnel bisa ping, LAN gagal	Route belum ada	`/ip route print`	Tambah route subnet lawan
LAN gagal	`allowed-address` salah	peer detail	Masukkan subnet lawan
LAN gagal	Firewall forward drop	filter print	Izinkan trafik antar-LAN
VPC gagal ping	Gateway VPC salah	`show ip`	Set gateway ke router lokal

I.18 Troubleshooting L2TP/IPsec

Gejala	Kemungkinan	Cek	Solusi
L2TP client tidak connected	secret salah	`/log print`	Samakan IPsec secret
Auth gagal	username/password salah	`/ppp secret print`	Perbaiki credential
Tunnel up, LAN gagal	route belum ada	`/ip route print`	Tambah route subnet lawan
Tunnel tidak terbentuk	port diblokir	firewall print	Buka UDP 500, 4500, 1701, ESP
Client connected, tidak bisa akses LAN	firewall forward drop	filter print	Izinkan forward
IP tunnel bentrok	pool/alamat sama	ppp active	Ganti pool/alamat VPN

I.19 Perbandingan WireGuard dan L2TP/IPsec

Aspek	WireGuard RouterOS v7	L2TP/IPsec RouterOS v6/v7
Fokus buku	Lab utama	Pembanding / kompatibilitas
Ketersediaan	RouterOS v7	RouterOS v6 dan v7
Konsep utama	key, peer, allowed-address	PPP, user, password, IPsec secret
Konfigurasi	relatif ringkas	lebih banyak komponen
Cocok untuk	site-to-site modern	perangkat lama / kompatibilitas
Catatan	perlu public key benar	perlu secret, user, password benar

I.20 Template Dokumentasi VPN

Item	Site A	Site B
Nama router		
WAN IP		
LAN subnet		
VPN type	WireGuard / L2TP/IPsec	WireGuard / L2TP/IPsec
Tunnel IP		
Port		
Public key		
Peer public key		
Allowed address		
Route ke subnet lawan		
Firewall input		
Firewall forward		
File export		

Catatan:

- Jangan menulis private key di dokumen yang dibagikan.
- Jangan menulis password asli di dokumen publik.
- Untuk buku/lab, gunakan placeholder.

I.21 Checklist VPN Berhasil

VPN dianggap berhasil jika:

- WAN antar router bisa saling ping.
- Interface VPN aktif.
- WireGuard handshake muncul, atau L2TP client connected.
- IP tunnel bisa saling ping.
- Route ke subnet lawan ada.
- Firewall input mengizinkan port VPN.
- Firewall forward mengizinkan trafik antar-LAN.
- VPC-A bisa ping VPC-B.
- VPC-B bisa ping VPC-A.
- Konfigurasi sudah diekspor.

  Export:

```
/export file=R1-vpn-final
/export file=R2-vpn-final
```

I.22 Golden Rules VPN

- **Jangan mulai VPN sebelum IP dasar antar router berhasil.**
- **Tunnel aktif belum berarti LAN bisa saling akses.**
- **VPN selalu butuh route ke subnet lawan.**
- **Firewall input membuka tunnel; firewall forward melewatkan trafik LAN.**
- **WireGuard v7 membutuhkan public key dan allowed-address yang benar.**
- **RouterOS v6 tidak memakai WireGuard native; gunakan L2TP/IPsec atau tunnel lain yang didukung.**
- **L2TP/IPsec membutuhkan user, password, dan IPsec secret yang sama.**
- **Jangan memakai password contoh di jaringan nyata.**
- **Jangan dokumentasikan private key atau password asli di naskah publik.**
- **Setelah VPN berhasil, export konfigurasi.**

Lampiran J. Template IPv6

Lampiran ini berisi template cepat untuk membuat **lab IPv6 dasar** menggunakan:

- **MikroTik RouterOS**
- **VPC GNS3**
- **Ubuntu VM**
- **Dual-stack IPv4/IPv6**
- **Firewall IPv6 dasar**
- **Routing IPv6 sederhana**

Gunakan prefix dokumentasi untuk lab:

```
2001:db8::/32
```

Catatan penting:

> **Jangan memakai alamat IPv6 publik sembarangan untuk lab.**
> Gunakan `2001:db8::/32` agar aman untuk dokumentasi dan simulasi.

J.1 Topologi Dasar IPv6

```
VPC / Ubuntu VM ---- MikroTik CHR
```

Contoh IP:

Perangkat	Interface	IPv6
MikroTik	ether2	`2001:db8:10::1/64`
VPC GNS3	eth0	`2001:db8:10::10/64`
Ubuntu VM	enp0s3	`2001:db8:10::20/64`
Gateway IPv6	-	`2001:db8:10::1`

J.2 Template IPv6 RouterOS v7/v6

Pasang IPv6 pada interface LAN.

```
/system identity set name=R1-IPV6

/ipv6 address add address=2001:db8:10::1/64 interface=ether2
advertise=yes comment="IPv6 LAN"

/ipv6 address print
/ipv6 route print
```

Makna:

- `2001:db8:10::1/64` adalah IPv6 gateway LAN.
- `ether2` adalah interface ke VPC/Ubuntu.
- `advertise=yes` mengizinkan router mengiklankan prefix ke client.
- Route connected akan muncul otomatis untuk `2001:db8:10::/64`.

Catatan:

> Jika interface LAN memakai **bridge**, pasang IPv6 di **bridge**, bukan di port anggota.

Contoh bridge:

```
/ipv6 address add address=2001:db8:10::1/64
interface=bridge-lan advertise=yes comment="IPv6 LAN bridge"
```

J.3 Template IPv6 VPC GNS3

Di VPC GNS3:

```
PC1> ip 2001:db8:10::10/64 2001:db8:10::1
PC1> show ipv6
PC1> ping 2001:db8:10::1
```

Makna:

Perintah	Fungsi
`ip 2001:db8:10::10/64 2001:db8:10::1`	memberi IPv6 dan gateway
`show ipv6`	melihat alamat IPv6
`ping 2001:db8:10::1`	uji koneksi ke MikroTik

Target sukses:

- VPC punya IPv6.
- Gateway IPv6 benar.
- VPC bisa ping MikroTik.

J.4 Template IPv6 Ubuntu VM

Cek interface dulu:

```
ip addr
```

Misal interface Ubuntu adalah `enp0s3`.

Tambahkan IPv6 sementara:

```
sudo ip -6 addr add 2001:db8:10::20/64 dev enp0s3
sudo ip -6 route add default via 2001:db8:10::1
```

Verifikasi:

```
ip -6 addr
ip -6 route
ping -6 2001:db8:10::1
```

Target sukses:

- Ubuntu punya IPv6 `2001:db8:10::20/64`.
- Default route IPv6 menuju `2001:db8:10::1`.
- Ubuntu bisa ping IPv6 MikroTik.

Catatan:

> Nama interface Ubuntu bisa berbeda.
> Selalu cek dengan `ip addr`.

J.5 Template Dual-Stack IPv4/IPv6

Dual-stack berarti IPv4 dan IPv6 berjalan bersamaan.

RouterOS

```
/ip address add address=192.168.10.1/24 interface=ether2
comment="IPv4 LAN"
/ipv6 address add address=2001:db8:10::1/64 interface=ether2
advertise=yes comment="IPv6 LAN"
```

VPC GNS3

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> ip 2001:db8:10::10/64 2001:db8:10::1
PC1> ping 192.168.10.1
PC1> ping 2001:db8:10::1
```

Ubuntu VM

```
sudo ip addr add 192.168.10.20/24 dev enp0s3
sudo ip route add default via 192.168.10.1

sudo ip -6 addr add 2001:db8:10::20/64 dev enp0s3
sudo ip -6 route add default via 2001:db8:10::1

ping 192.168.10.1
ping -6 2001:db8:10::1
```

Makna:

- IPv4 tetap bisa dipakai.
- IPv6 bisa diuji tanpa merusak lab IPv4.
- Troubleshooting lebih mudah karena dua protokol bisa diuji terpisah.

J.6 Template Routing IPv6 Sederhana

Topologi:

```
VPC-A ---- R1 ---- R2 ---- VPC-B
```

IP Plan:

Perangkat	Interface	IPv6
R1	LAN-A	2001:db8:10::1/64
R1	R1-R2	2001:db8:12::1/64
R2	R1-R2	2001:db8:12::2/64
R2	LAN-B	2001:db8:20::1/64
VPC-A	eth0	2001:db8:10::10/64
VPC-B	eth0	2001:db8:20::10/64

R1 — RouterOS

```
/system identity set name=R1-IPV6

/ipv6 address add address=2001:db8:10::1/64 interface=ether1
advertise=yes comment="LAN-A"
/ipv6 address add address=2001:db8:12::1/64 interface=ether2
comment="Link R1-R2"

/ipv6 route add dst-address=2001:db8:20::/64
gateway=2001:db8:12::2 comment="Route to LAN-B"
```

R2 — RouterOS

```
/system identity set name=R2-IPV6

/ipv6 address add address=2001:db8:12::2/64 interface=ether1
comment="Link R2-R1"
/ipv6 address add address=2001:db8:20::1/64 interface=ether2
advertise=yes comment="LAN-B"

/ipv6 route add dst-address=2001:db8:10::/64
gateway=2001:db8:12::1 comment="Route to LAN-A"
```

VPC-A

```
PC1> ip 2001:db8:10::10/64 2001:db8:10::1
PC1> ping 2001:db8:20::10
```

VPC-B

```
PC2> ip 2001:db8:20::10/64 2001:db8:20::1
PC2> ping 2001:db8:10::10
```

Target sukses:

- R1 dan R2 saling ping melalui link IPv6.
- VPC-A bisa ping VPC-B.
- VPC-B bisa ping VPC-A.

J.7 Template Firewall IPv6 Dasar

Jangan aktifkan IPv6 tanpa firewall.

Template input dasar:

```
/ipv6 firewall filter add chain=input
connection-state=established,related action=accept
comment="IPv6 INPUT: allow established, related"
/ipv6 firewall filter add chain=input
connection-state=invalid action=drop comment="IPv6 INPUT:
drop invalid"
/ipv6 firewall filter add chain=input protocol=icmpv6
action=accept comment="IPv6 INPUT: allow ICMPv6"
/ipv6 firewall filter add chain=input in-interface=ether2
action=accept comment="IPv6 INPUT: allow LAN"
/ipv6 firewall filter add chain=input action=drop
comment="IPv6 INPUT: drop other"
```

Jika interface LAN adalah `bridge-lan`, sesuaikan:

```
/ipv6 firewall filter add chain=input in-interface=bridge-lan
action=accept comment="IPv6 INPUT: allow LAN bridge"
```

Catatan penting:

> **Jangan blok ICMPv6 sembarangan.**
> IPv6 membutuhkan ICMPv6 untuk fungsi seperti *Neighbor Discovery*.

J.8 Template Forward Firewall IPv6

Jika router meneruskan trafik IPv6 antar-LAN, tambahkan forward rule.

Contoh:

```
/ipv6 firewall filter add chain=forward
connection-state=established,related action=accept
comment="IPv6 FORWARD: allow established, related"
/ipv6 firewall filter add chain=forward
connection-state=invalid action=drop comment="IPv6 FORWARD:
drop invalid"
/ipv6 firewall filter add chain=forward
src-address=2001:db8:10::/64 dst-address=2001:db8:20::/64
action=accept comment="IPv6 FORWARD: LAN-A to LAN-B"
/ipv6 firewall filter add chain=forward
src-address=2001:db8:20::/64 dst-address=2001:db8:10::/64
action=accept comment="IPv6 FORWARD: LAN-B to LAN-A"
/ipv6 firewall filter add chain=forward action=drop
comment="IPv6 FORWARD: drop other"
```

Gunakan jika ingin kontrol lebih ketat antar-subnet IPv6.

J.9 Verifikasi RouterOS IPv6

```
/ipv6 address print
/ipv6 route print
/ipv6 neighbor print
/ipv6 firewall filter print
/tool ping 2001:db8:10::10
```

Yang harus dicek:

- alamat IPv6 ada di interface benar,
- route connected muncul,
- neighbor muncul setelah ada trafik,
- firewall tidak memblokir ICMPv6,
- ping ke client berhasil.

J.10 Verifikasi VPC GNS3 IPv6

```
PC1> show ipv6
PC1> ping 2001:db8:10::1
PC1> trace 2001:db8:20::10
```

Jika gagal:

- cek IPv6 VPC,
- cek gateway,
- cek link GNS3,
- cek IPv6 router,
- cek firewall IPv6.

J.11 Verifikasi Ubuntu IPv6

```
ip -6 addr
ip -6 route
ping -6 2001:db8:10::1
traceroute -6 2001:db8:10::1
```

Jika `traceroute -6` belum tersedia:

```
sudo apt install -y traceroute
```

Jika gagal:

- cek interface Ubuntu,
- cek IPv6/prefix,
- cek default route IPv6,
- cek firewall RouterOS,
- cek topologi VirtualBox/GNS3.

J.12 Template IPv6 dengan Bridge

Jika LAN memakai bridge:

```
/interface bridge add name=bridge-lan
/interface bridge port add bridge=bridge-lan interface=ether2
/interface bridge port add bridge=bridge-lan interface=ether3

/ipv6 address add address=2001:db8:10::1/64
interface=bridge-lan advertise=yes comment="IPv6 bridge LAN"
```

VPC1:

```
PC1> ip 2001:db8:10::10/64 2001:db8:10::1
```

VPC2:

```
PC2> ip 2001:db8:10::20/64 2001:db8:10::1
```

Uji:

```
PC1> ping 2001:db8:10::20
PC1> ping 2001:db8:10::1
```

Catatan:

Jika memakai bridge, **IP gateway IPv6 dipasang di bridge**, bukan di port anggota.

J.13 Template IPv6 untuk VLAN

Contoh:

VLAN	Fungsi	Prefix	Gateway
VLAN 10	Admin	2001:db8:10::/64	2001:db8:10::1
VLAN 20	User	2001:db8:20::/64	2001:db8:20::1

RouterOS:

```
/interface vlan add interface=bridge-vlan name=vlan10-admin
vlan-id=10
/interface vlan add interface=bridge-vlan name=vlan20-user
vlan-id=20

/ipv6 address add address=2001:db8:10::1/64
interface=vlan10-admin advertise=yes comment="IPv6 VLAN
Admin"
/ipv6 address add address=2001:db8:20::1/64
interface=vlan20-user advertise=yes comment="IPv6 VLAN User"
```

Catatan:

Untuk VLAN, **gateway IPv6 dipasang di interface VLAN**, bukan di bridge utama atau port fisik.

J.14 Template Dokumentasi IPv6

Item	Nilai
Nama lab	
Prefix IPv6 utama	
Router	
Interface router	
IPv6 gateway	
Client VPC	
IPv6 VPC	
Client Ubuntu	
IPv6 Ubuntu	
Route IPv6	
Firewall IPv6	
Hasil ping	
File export	

Contoh:

Item	Nilai
Nama lab	`Lab IPv6 Dasar`
Prefix IPv6 utama	`2001:db8:10::/64`
Router	`R1-IPV6`
Interface router	`ether2`
IPv6 gateway	`2001:db8:10::1/64`
Client VPC	`PC1`
IPv6 VPC	`2001:db8:10::10/64`
Client Ubuntu	`U-IPV6`
IPv6 Ubuntu	`2001:db8:10::20/64`
Route IPv6	connected
Firewall IPv6	input dasar aktif
Hasil ping	berhasil
File export	`R1-ipv6-final.rsc`

J.15 Troubleshooting IPv6

Masalah	Kemungkinan	Cek	Solusi
VPC tidak bisa ping router	IPv6/gateway salah	`show ipv6`	Perbaiki IP dan gateway
Ubuntu tidak bisa ping router	interface salah	`ip -6 addr`	Cek nama interface
Route IPv6 tidak ada	belum ada route statik	`/ipv6 route print`	Tambah route
Neighbor kosong	tidak ada trafik/link salah	`/ipv6 neighbor print`	Cek link dan ping
ICMPv6 gagal	firewall memblokir	`/ipv6 firewall filter print`	Izinkan ICMPv6
Dual-stack membingungkan	uji IPv4/IPv6 tercampur	`ping` dan `ping -6`	Pisahkan pengujian
IPv6 aktif tapi tidak aman	firewall kosong	`/ipv6 firewall filter print`	Tambah firewall IPv6
Prefix salah	client beda subnet	cek IP plan	Samakan `/64`

J.16 Checklist IPv6 Berhasil

IPv6 dianggap berhasil jika:

- Router punya IPv6 pada interface yang benar.
- Client VPC punya IPv6 dan gateway.
- Ubuntu punya IPv6 dan route.
- VPC bisa ping IPv6 router.
- Ubuntu bisa ping IPv6 router.
- Route IPv6 muncul di RouterOS.
- Neighbor muncul setelah ada trafik.
- ICMPv6 tidak diblokir.
- Firewall IPv6 dasar aktif.
- Jika multi-router, route statik IPv6 bekerja.
- Konfigurasi sudah diekspor.

  Export:

  ```
  /export file=R1-ipv6-final
  ```

J.17 Golden Rules IPv6

- **Gunakan `2001:db8::/32` untuk lab dan dokumentasi.**
- **Untuk LAN pemula, gunakan `/64`.**
- **IPv6 bukan IPv4 dengan alamat panjang; cara kerjanya berbeda.**
- **Link-local `fe80::/10` penting untuk operasi lokal.**
- **Jangan blok ICMPv6 sembarangan.**
- **Jangan aktifkan IPv6 tanpa firewall IPv6.**
- **Gunakan `ping -6` di Ubuntu agar jelas memakai IPv6.**
- **Gunakan `show ipv6` di VPC untuk cek alamat.**
- **Jika memakai bridge, pasang IPv6 di bridge.**
- **Jika memakai VLAN, pasang IPv6 di interface VLAN.**
- **Dual-stack membantu belajar tanpa memutus lab IPv4.**
- **Setelah berhasil, export konfigurasi.**

Lampiran K. Template OSPF 3 Router

RouterOS v7 dan v6

Lampiran ini berisi template cepat untuk membuat **OSPF 3 router** di GNS3 memakai **MikroTik CHR** dan **VPC GNS3**.

Target lab:

- R1, R2, R3 membentuk **OSPF single-area area 0**.
- VPC-A di belakang R1 bisa ping VPC-C di belakang R3.
- Pembaca memahami perbedaan sintaks **RouterOS v7** dan **RouterOS v6**.

K.1 Topologi Lab

```
VPC-A ---- R1 ---- R2 ---- R3 ---- VPC-C
```

Detail jaringan:

```
LAN-A       : 192.168.10.0/24
Link R1-R2  : 10.10.12.0/30
Link R2-R3  : 10.10.23.0/30
LAN-C       : 192.168.30.0/24
```

K.2 IP Plan

Perangkat	Interface	IP/Prefix	Fungsi
R1	ether1	`192.168.10.1/24`	Gateway LAN-A
R1	ether2	`10.10.12.1/30`	Link R1-R2
R2	ether1	`10.10.12.2/30`	Link R2-R1
R2	ether2	`10.10.23.1/30`	Link R2-R3
R3	ether1	`10.10.23.2/30`	Link R3-R2
R3	ether2	`192.168.30.1/24`	Gateway LAN-C
VPC-A	eth0	`192.168.10.10/24`	Client LAN-A
VPC-C	eth0	`192.168.30.10/24`	Client LAN-C

Gateway client:

Client	Gateway
VPC-A	192.168.10.1
VPC-C	192.168.30.1

Bagian A — Konfigurasi IP Dasar

K.3 R1 — IP Dasar

```
/system identity set name=R1

/interface comment ether1 comment="LAN-A to VPC-A"
/interface comment ether2 comment="Link R1-R2"

/ip address add address=192.168.10.1/24 interface=ether1
comment="LAN-A"
/ip address add address=10.10.12.1/30 interface=ether2
comment="R1-to-R2"
```

K.4 R2 — IP Dasar

```
/system identity set name=R2

/interface comment ether1 comment="Link R2-R1"
/interface comment ether2 comment="Link R2-R3"

/ip address add address=10.10.12.2/30 interface=ether1
comment="R2-to-R1"
/ip address add address=10.10.23.1/30 interface=ether2
comment="R2-to-R3"
```

K.5 R3 — IP Dasar

```
/system identity set name=R3

/interface comment ether1 comment="Link R3-R2"
/interface comment ether2 comment="LAN-C to VPC-C"

/ip address add address=10.10.23.2/30 interface=ether1
comment="R3-to-R2"
/ip address add address=192.168.30.1/24 interface=ether2
comment="LAN-C"
```

K.6 VPC-A dan VPC-C

VPC-A

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> show ip
PC1> ping 192.168.10.1
```

VPC-C

```
PC2> ip 192.168.30.10/24 192.168.30.1
PC2> show ip
PC2> ping 192.168.30.1
```

Target awal:

- VPC-A bisa ping R1.
- VPC-C bisa ping R3.
- R1 bisa ping R2.
- R2 bisa ping R3.

Verifikasi antar-router:

```
/tool ping 10.10.12.2
/tool ping 10.10.23.2
```

Bagian B — OSPF RouterOS v7

K.7 Konsep OSPF v7

Pada RouterOS v7, konfigurasi OSPF memakai:

- **instance**
- **area**
- **interface-template**
- **router-id**
- **passive** untuk LAN yang tidak perlu membentuk neighbor

Area yang dipakai:

```
area 0.0.0.0 / backbone
```

Router ID:

Router	Router ID
R1	1.1.1.1
R2	2.2.2.2
R3	3.3.3.3

Catatan penting:

> **Interface LAN sebaiknya passive.**
> Subnet LAN tetap diiklankan, tetapi router tidak mencoba membentuk neighbor OSPF dengan VPC.

K.8 OSPF v7 — R1

```
/routing ospf instance add name=ospf-v2 version=2
router-id=1.1.1.1

/routing ospf area add name=backbone area-id=0.0.0.0
instance=ospf-v2

/routing ospf interface-template add networks=10.10.12.0/30
area=backbone
/routing ospf interface-template add networks=192.168.10.0/24
area=backbone passive
```

K.9 OSPF v7 — R2

```
/routing ospf instance add name=ospf-v2 version=2
router-id=2.2.2.2

/routing ospf area add name=backbone area-id=0.0.0.0
instance=ospf-v2

/routing ospf interface-template add networks=10.10.12.0/30
area=backbone
/routing ospf interface-template add networks=10.10.23.0/30
area=backbone
```

K.10 OSPF v7 — R3

```
/routing ospf instance add name=ospf-v2 version=2
router-id=3.3.3.3

/routing ospf area add name=backbone area-id=0.0.0.0
instance=ospf-v2

/routing ospf interface-template add networks=10.10.23.0/30
area=backbone
/routing ospf interface-template add networks=192.168.30.0/24
area=backbone passive
```

K.11 Verifikasi OSPF v7

Cek Neighbor

```
/routing ospf neighbor print
```

Target:

Router	Neighbor yang Harus Muncul
R1	R2
R2	R1 dan R3
R3	R2

Cek Route

```
/ip route print
```

Target:

- R1 belajar route ke `192.168.30.0/24`.
- R3 belajar route ke `192.168.10.0/24`.
- R2 mengetahui kedua LAN.

Uji Ping

Dari VPC-A:

```
PC1> ping 192.168.30.10
PC1> trace 192.168.30.10
```

Dari VPC-C:

```
PC2> ping 192.168.10.10
PC2> trace 192.168.10.10
```

Bagian C — OSPF RouterOS v6

K.12 Konsep OSPF v6

Pada RouterOS v6, konfigurasi OSPF memakai pola:

- set **router-id** pada instance default,
- tambahkan **network** ke area backbone.

Area default:

```
backbone
```

Router ID sama:

Router	Router ID
R1	1.1.1.1
R2	2.2.2.2
R3	3.3.3.3

K.13 OSPF v6 — R1

```
/routing ospf instance set default router-id=1.1.1.1

/routing ospf network add network=10.10.12.0/30 area=backbone
/routing ospf network add network=192.168.10.0/24
area=backbone


Jika ingin LAN tidak membentuk neighbor, gunakan pengaturan
passive pada interface terkait sesuai versi dan menu yang
tersedia di RouterOS v6.
```

K.14 OSPF v6 — R2

```
/routing ospf instance set default router-id=2.2.2.2

/routing ospf network add network=10.10.12.0/30 area=backbone
/routing ospf network add network=10.10.23.0/30 area=backbone
```

K.15 OSPF v6 — R3

```
/routing ospf instance set default router-id=3.3.3.3

/routing ospf network add network=10.10.23.0/30 area=backbone
/routing ospf network add network=192.168.30.0/24
area=backbone
```

K.16 Verifikasi OSPF v6

```
/routing ospf neighbor print
/ip route print
```

Target:

- neighbor terbentuk antar-router,
- route LAN lawan muncul,
- VPC-A bisa ping VPC-C,
- VPC-C bisa ping VPC-A.

Bagian D — Static Route Pembanding

K.17 Static Route Manual Sebelum OSPF

Untuk memahami bedanya, lab yang sama bisa dicoba dengan static route dulu.

R1

```
/ip route add dst-address=192.168.30.0/24 gateway=10.10.12.2
comment="Static to LAN-C"
```

R2

```
/ip route add dst-address=192.168.10.0/24 gateway=10.10.12.1
comment="Static to LAN-A"
/ip route add dst-address=192.168.30.0/24 gateway=10.10.23.2
comment="Static to LAN-C"
```

R3

```
/ip route add dst-address=192.168.10.0/24 gateway=10.10.23.1
comment="Static to LAN-A"
```

Makna:

- Static route ditulis manual.
- OSPF mempelajari route otomatis.
- Pembaca sebaiknya memahami static route sebelum percaya pada OSPF.

Bagian E — Troubleshooting OSPF

K.18 OSPF Neighbor Tidak Muncul

Cek di router:

```
/ip address print
/tool ping 10.10.12.2
/routing ospf neighbor print
```

Penyebab umum:

Penyebab	Solusi
IP link antar-router salah	Perbaiki IP link
Router belum bisa saling ping	Cek interface dan subnet
Area tidak sama	Samakan area backbone
Network/interface-template salah	Perbaiki network
Router ID bentrok	Gunakan router-id unik
Firewall memblokir OSPF	Izinkan trafik OSPF di lab

K.19 Route OSPF Tidak Muncul

Cek:

```
/ip route print
/routing ospf interface-template print
/routing ospf network print
```

Penyebab umum:

Penyebab	Solusi
Neighbor belum terbentuk	Selesaikan neighbor dulu
LAN tidak diiklankan	Tambahkan network LAN
Passive salah paham	Passive tetap mengiklankan LAN, tetapi tidak membentuk neighbor
Network salah prefix	Perbaiki network dan prefix
Route static lebih prioritas	Cek distance/route aktif

K.20 VPC Tidak Bisa Ping Jaringan Lawan

Cek VPC:

```
PC1> show ip
PC1> ping 192.168.10.1
PC1> trace 192.168.30.10
```

Cek RouterOS:

```
/ip route print
/tool ping 192.168.30.10
```

Penyebab umum:

Gejala	Kemungkinan
VPC tidak bisa ping gateway	IP/gateway VPC salah
R1 tidak punya route ke LAN-C	OSPF belum belajar route
R3 tidak punya route balik ke LAN-A	OSPF belum lengkap
Ping router bisa, client gagal	Firewall client/router memblokir
Trace berhenti di R2	R2/R3 route belum benar

K.21 Firewall untuk OSPF Lab

Jika firewall input sangat ketat, OSPF bisa gagal. Untuk lab, izinkan OSPF antar-router.

Contoh pada link antar-router:

```
/ip firewall filter add chain=input protocol=ospf
action=accept comment="INPUT: allow OSPF"
```

Letakkan sebelum rule drop input.

Catatan:

> **Untuk lab, izinkan OSPF di link antar-router.**
> Jangan membuka OSPF ke jaringan yang tidak perlu.

Bagian F — Dokumentasi dan Export

K.22 Template Dokumentasi OSPF

Item	R1	R2	R3
Router ID	`1.1.1.1`	`2.2.2.2`	`3.3.3.3`
LAN subnet	`192.168.10.0/24`	-	`192.168.30.0/24`
Link ke kiri	-	`10.10.12.0/30`	`10.10.23.0/30`
Link ke kanan	`10.10.12.0/30`	`10.10.23.0/30`	-
Area	backbone	backbone	backbone
Passive LAN	ya	-	ya
Neighbor	R2	R1, R3	R2
Export file	`R1-ospf.rsc`	`R2-ospf.rsc`	`R3-ospf.rsc`

K.23 Export Konfigurasi

Setelah lab berhasil:

R1

```
/export file=R1-ospf-final
```

R2

```
/export file=R2-ospf-final
```

R3

```
/export file=R3-ospf-final
```

K.24 Checklist OSPF Berhasil

OSPF dianggap berhasil jika:

- R1 bisa ping R2.
- R2 bisa ping R1 dan R3.
- R3 bisa ping R2.
- OSPF neighbor muncul.
- R1 belajar route ke `192.168.30.0/24`.
- R3 belajar route ke `192.168.10.0/24`.
- VPC-A bisa ping VPC-C.
- VPC-C bisa ping VPC-A.
- LAN interface dibuat passive pada OSPF v7.
- Router ID unik.
- Konfigurasi sudah di-*export*.

K.25 Golden Rules OSPF 3 Router

- **Jangan mulai OSPF sebelum IP antar-router bisa ping.**
- **Router ID harus unik.**
- **Area harus konsisten.**
- **Gunakan area 0 untuk lab awal.**
- **LAN sebaiknya passive.**
- **Link antar-router tidak passive.**
- **OSPF mempelajari route, tetapi IP plan tetap harus benar.**
- **Jika neighbor kosong, cek IP link dulu.**
- **Jika neighbor ada tapi route tidak muncul, cek network yang diiklankan.**
- **Static route bagus untuk belajar logika; OSPF bagus untuk jaringan yang mulai membesar.**
- **Setelah berhasil, export konfigurasi semua router.**

Lampiran L. Template BGP Customer–ISP

RouterOS v7 dan v6

Lampiran ini berisi template cepat untuk membuat simulasi **BGP Customer–ISP** menggunakan **MikroTik CHR di GNS3**.

Fokus utama:

- **Customer** memiliki ASN sendiri.
- **ISP** memiliki ASN berbeda.
- Customer mengumumkan prefix LAN ke ISP.
- ISP dapat memberi default route ke Customer.
- Filter BGP dipasang agar tidak terjadi **route leak**.
- Disediakan contoh **RouterOS v7** dan **RouterOS v6**.

L.1 Topologi Dasar Customer–ISP

```
VPC-CUST ---- R-CUST ---- R-ISP
```

Detail:

```
LAN Customer : 192.168.10.0/24
Link BGP     : 10.10.12.0/30

Customer ASN : 65001
ISP ASN      : 65002
```

L.2 IP Plan

Perangkat	Interface	IP/Prefix	ASN	Fungsi
R-CUST	ether1	`192.168.10.1/24`	65001	LAN Customer
R-CUST	ether2	`10.10.12.1/30`	65001	Link ke ISP
R-ISP	ether1	`10.10.12.2/30`	65002	Link ke Customer
VPC-CUST	eth0	`192.168.10.10/24`	-	Client Customer

Gateway VPC:

```
192.168.10.1
```

L.3 Konfigurasi IP Dasar

R-CUST

```
/system identity set name=R-CUST

/interface comment ether1 comment="LAN Customer"
/interface comment ether2 comment="Link to ISP"

/ip address add address=192.168.10.1/24 interface=ether1
comment="LAN Customer"
/ip address add address=10.10.12.1/30 interface=ether2
comment="BGP link to ISP"
```

R-ISP

```
/system identity set name=R-ISP

/interface comment ether1 comment="Link to Customer"

/ip address add address=10.10.12.2/30 interface=ether1
comment="BGP link to Customer"
```

VPC-CUST

```
PC1> ip 192.168.10.10/24 192.168.10.1
PC1> show ip
PC1> ping 192.168.10.1
```

Verifikasi Link Customer–ISP

Dari R-CUST:

```
/tool ping 10.10.12.2
```

Dari R-ISP:

```
/tool ping 10.10.12.1
```

Catatan penting:

> **Jangan lanjut ke BGP jika link IP dasar Customer–ISP belum bisa saling ping.**

Bagian A — BGP RouterOS v7

L.4 Konsep BGP RouterOS v7

Pada RouterOS v7, konfigurasi BGP memakai beberapa bagian utama:

- **template**
- **connection**
- **remote address**
- **remote AS**
- **local role**
- **routing filter**
- **output filter**

Dalam lab ini:

Router	ASN	Router ID
R-CUST	65001	`1.1.1.1`
R-ISP	65002	`2.2.2.2`

L.5 BGP v7 — Customer Mengumumkan Prefix ke ISP

Customer akan mengumumkan prefix:

```
192.168.10.0/24
```

R-CUST — RouterOS v7

```
/routing bgp template add name=tpl-as65001 as=65001
router-id=1.1.1.1

/routing bgp connection add name=to-isp
remote.address=10.10.12.2 remote.as=65002
templates=tpl-as65001 local.role=ebgp

/ip route add dst-address=192.168.10.0/24 blackhole
comment="Prefix Customer for BGP advertisement"

/routing filter rule add chain=bgp-out rule="if (dst ==
192.168.10.0/24) { accept }"
/routing filter rule add chain=bgp-out rule="reject"

/routing bgp connection set to-isp
output.filter-chain=bgp-out
```

Makna:

- `tpl-as65001` menyimpan ASN dan router-id Customer.
- `to-isp` membuat koneksi BGP ke ISP.
- `blackhole` memastikan prefix Customer ada di routing table.
- `bgp-out` hanya mengizinkan prefix `192.168.10.0/24`.
- Semua prefix lain ditolak.

Catatan penting:

> **Filter output wajib.**
> Customer hanya boleh mengumumkan prefix yang memang miliknya.

L.6 BGP v7 — ISP Menerima Prefix Customer

R-ISP — RouterOS v7

```
/routing bgp template add name=tpl-as65002 as=65002
router-id=2.2.2.2

/routing bgp connection add name=to-customer
remote.address=10.10.12.1 remote.as=65001
templates=tpl-as65002 local.role=ebgp
```

Verifikasi di R-ISP

```
/routing bgp session print
/routing bgp connection print
/ip route print
```

Target:

- BGP session **established**.
- ISP menerima route `192.168.10.0/24` dari Customer.

L.7 BGP v7 — ISP Memberi Default Route ke Customer

Dalam jaringan nyata, customer kecil biasanya tidak perlu menerima semua route Internet. Customer cukup menerima **default route**.

Di lab, ISP dapat mengirim:

```
0.0.0.0/0
```

R-ISP — Buat Default Route Lab

```
/ip route add dst-address=0.0.0.0/0 blackhole
comment="Default route lab for BGP advertisement"
```

R-ISP — Filter Output ke Customer

```
/routing filter rule add chain=isp-out rule="if (dst ==
0.0.0.0/0) { accept }"
/routing filter rule add chain=isp-out rule="reject"

/routing bgp connection set to-customer
output.filter-chain=isp-out
```

Makna:

- ISP hanya mengirim default route ke Customer.
- ISP tidak mengirim route lain.

Verifikasi di R-CUST

```
/routing bgp session print
/ip route print
```

Target:

- Customer menerima route 0.0.0.0/0 dari ISP.

L.8 BGP v7 — Filter Input di Customer

Agar lebih aman, Customer dapat hanya menerima default route dari ISP.

R-CUST

```
/routing filter rule add chain=bgp-in rule="if (dst ==
0.0.0.0/0) { accept }"
/routing filter rule add chain=bgp-in rule="reject"

/routing bgp connection set to-isp input.filter-chain=bgp-in
```

Makna:

- Customer menerima hanya default route.
- Route lain dari ISP ditolak.

Catatan:

Filter input dan output membuat BGP lebih terkendali.

L.9 Template Lengkap RouterOS v7 — R-CUST

```
/system identity set name=R-CUST

/ip address add address=192.168.10.1/24 interface=ether1
comment="LAN Customer"
/ip address add address=10.10.12.1/30 interface=ether2
comment="Link to ISP"

/routing bgp template add name=tpl-as65001 as=65001
router-id=1.1.1.1

/routing bgp connection add name=to-isp
remote.address=10.10.12.2 remote.as=65002
templates=tpl-as65001 local.role=ebgp

/ip route add dst-address=192.168.10.0/24 blackhole
comment="Prefix Customer for BGP"

/routing filter rule add chain=bgp-out rule="if (dst ==
192.168.10.0/24) { accept }"
/routing filter rule add chain=bgp-out rule="reject"

/routing filter rule add chain=bgp-in rule="if (dst ==
0.0.0.0/0) { accept }"
/routing filter rule add chain=bgp-in rule="reject"

/routing bgp connection set to-isp
output.filter-chain=bgp-out input.filter-chain=bgp-in

/export file=R-CUST-bgp-v7
```

L.10 Template Lengkap RouterOS v7 — R-ISP

```
/system identity set name=R-ISP

/ip address add address=10.10.12.2/30 interface=ether1
comment="Link to Customer"

/routing bgp template add name=tpl-as65002 as=65002
router-id=2.2.2.2

/routing bgp connection add name=to-customer
remote.address=10.10.12.1 remote.as=65001
templates=tpl-as65002 local.role=ebgp

/ip route add dst-address=0.0.0.0/0 blackhole
comment="Default route lab"

/routing filter rule add chain=isp-out rule="if (dst ==
0.0.0.0/0) { accept }"
/routing filter rule add chain=isp-out rule="reject"

/routing bgp connection set to-customer
output.filter-chain=isp-out

/export file=R-ISP-bgp-v7
```

Bagian B — BGP RouterOS v6

L.11 Konsep BGP RouterOS v6

Pada RouterOS v6, konfigurasi BGP memakai pola lebih lama:

- set ASN pada **default instance**,
- buat **peer**,
- tambahkan **network** yang ingin diumumkan.

Dalam lab ini:

Router	ASN	Router ID
R-CUST	65001	`1.1.1.1`
R-ISP	65002	`2.2.2.2`

L.12 BGP v6 — Customer

```
/routing bgp instance set default as=65001 router-id=1.1.1.1

/routing bgp peer add name=to-isp remote-address=10.10.12.2
remote-as=65002

/routing bgp network add network=192.168.10.0/24
```

Catatan:

- Pastikan `192.168.10.0/24` ada di routing table.
- Prefix LAN biasanya sudah ada karena IP `192.168.10.1/24` terpasang di interface LAN.

L.13 BGP v6 — ISP

```
/routing bgp instance set default as=65002 router-id=2.2.2.2

/routing bgp peer add name=to-customer
remote-address=10.10.12.1 remote-as=65001
```

Verifikasi:

```
/routing bgp peer print
/ip route print
```

Target:

- peer BGP established,
- ISP menerima route `192.168.10.0/24`.

L.14 Default Route dari ISP ke Customer — RouterOS v6

R-ISP

```
/ip route add dst-address=0.0.0.0/0 type=blackhole
comment="Default route lab"

/routing bgp network add network=0.0.0.0/0
```

Catatan:

- Ini contoh lab.
- Pada jaringan nyata, default route biasanya berasal dari upstream atau desain routing yang sah.
- Filtering di RouterOS v6 perlu disesuaikan dengan mekanisme routing filter versi tersebut.

L.15 Template Lengkap RouterOS v6 — R-CUST

```
/system identity set name=R-CUST

/ip address add address=192.168.10.1/24 interface=ether1
comment="LAN Customer"
/ip address add address=10.10.12.1/30 interface=ether2
comment="Link to ISP"

/routing bgp instance set default as=65001 router-id=1.1.1.1

/routing bgp peer add name=to-isp remote-address=10.10.12.2
remote-as=65002

/routing bgp network add network=192.168.10.0/24

/export file=R-CUST-bgp-v6
```

L.16 Template Lengkap RouterOS v6 — R-ISP

```
/system identity set name=R-ISP

/ip address add address=10.10.12.2/30 interface=ether1
comment="Link to Customer"

/routing bgp instance set default as=65002 router-id=2.2.2.2

/routing bgp peer add name=to-customer
remote-address=10.10.12.1 remote-as=65001

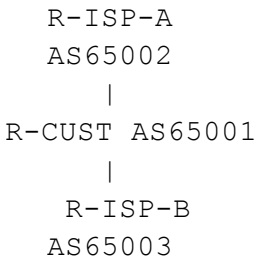
/ip route add dst-address=0.0.0.0/0 type=blackhole
comment="Default route lab"

/routing bgp network add network=0.0.0.0/0

/export file=R-ISP-bgp-v6
```

Bagian C — Dua ISP dan Satu Customer

L.17 Topologi Multi-Homing Sederhana



IP Plan:

Link	IP Customer	IP ISP	ASN ISP
Customer–ISP-A	10.10.12.1/30	10.10.12.2/30	65002
Customer–ISP-B	10.10.13.1/30	10.10.13.2/30	65003
LAN Customer	192.168.10.1/24	-	-

L.18 RouterOS v7 — Customer ke Dua ISP

```
/routing bgp template add name=tpl-as65001 as=65001 router-id=1.1.1.1

/routing bgp connection add name=to-isp-a remote.address=10.10.12.2
remote.as=65002 templates=tpl-as65001 local.role=ebgp
/routing bgp connection add name=to-isp-b remote.address=10.10.13.2
remote.as=65003 templates=tpl-as65001 local.role=ebgp

/ip route add dst-address=192.168.10.0/24 blackhole comment="Customer
prefix"

/routing filter rule add chain=bgp-out rule="if (dst ==
192.168.10.0/24) { accept }"
/routing filter rule add chain=bgp-out rule="reject"

/routing bgp connection set to-isp-a output.filter-chain=bgp-out
/routing bgp connection set to-isp-b output.filter-chain=bgp-out
```

Catatan:

> **Customer boleh mengumumkan prefix yang sama ke dua ISP dalam lab.**
> Namun filter tetap wajib agar hanya prefix Customer yang keluar.

L.19 RouterOS v6 — Customer ke Dua ISP

```
/routing bgp instance set default as=65001 router-id=1.1.1.1

/routing bgp peer add name=to-isp-a remote-address=10.10.12.2
remote-as=65002
/routing bgp peer add name=to-isp-b remote-address=10.10.13.2
remote-as=65003

/routing bgp network add network=192.168.10.0/24
```

Bagian D — Route Filtering dan Etika BGP

L.20 Filter Output Minimal RouterOS v7

Customer hanya boleh mengirim prefix miliknya.

```
/routing filter rule add chain=bgp-out rule="if (dst ==
192.168.10.0/24) { accept }"
/routing filter rule add chain=bgp-out rule="reject"
```

Pasang ke peer:

```
/routing bgp connection set to-isp
output.filter-chain=bgp-out
```

L.21 Filter Input Minimal RouterOS v7

Customer hanya menerima default route dari ISP.

```
/routing filter rule add chain=bgp-in rule="if (dst ==
0.0.0.0/0) { accept }"
/routing filter rule add chain=bgp-in rule="reject"
```

Pasang ke peer:

```
/routing bgp connection set to-isp input.filter-chain=bgp-in
```

L.22 Etika BGP

BGP harus diperlakukan serius, bahkan di lab.

Golden rules etika BGP:

- **Jangan advertise prefix yang bukan milik kita.**
- **Jangan menerima semua route tanpa filter.**
- **Jangan mengirim semua route tanpa filter.**
- **Jangan bereksperimen BGP ke jaringan nyata tanpa izin.**
- **Gunakan ASN private untuk lab.**
- **Gunakan prefix private/lab untuk simulasi.**
- **Dokumentasikan ASN, peer, prefix, filter, dan route.**

Kalimat kunci:

> **BGP adalah protokol kepercayaan. Salah filter bisa membuat route bocor.**

Bagian E — Verifikasi BGP

L.23 Verifikasi RouterOS v7

Cek Session

```
/routing bgp session print
```

Cek Connection

```
/routing bgp connection print
/routing bgp connection print detail
```

Cek Route

```
/ip route print
```

Cek Filter

```
/routing filter rule print
```

Target:

- BGP session established.
- ISP menerima prefix Customer.
- Customer menerima default route jika dikonfigurasi.
- Filter terpasang.

L.24 Verifikasi RouterOS v6

```
/routing bgp peer print
/routing bgp peer print detail
/ip route print
/routing bgp advertisements print
```

Target:

- Peer established.
- Prefix Customer muncul di ISP.
- Default route muncul di Customer jika dikirim ISP.

L.25 Verifikasi dari VPC

Di VPC Customer:

```
PC1> show ip
PC1> ping 192.168.10.1
PC1> trace 10.10.12.2
```

Jika ada jaringan simulasi di belakang ISP, uji:

```
PC1> ping <IP jaringan ISP>
PC1> trace <IP jaringan ISP>
```

Catatan:

VPC hanya menguji konektivitas dasar.
BGP diverifikasi terutama dari RouterOS.

Bagian F — Troubleshooting BGP

L.26 Session BGP Tidak Established

Cek:

```
/tool ping 10.10.12.2
/routing bgp session print
/routing bgp connection print detail
/log print
```

Penyebab umum:

Penyebab	Solusi
Link IP belum bisa ping	Perbaiki IP/link dulu
Remote address salah	Sesuaikan IP peer
Remote AS salah	Sesuaikan ASN lawan
Local AS salah	Sesuaikan ASN sendiri
Firewall memblokir TCP 179	Izinkan trafik BGP
Router ID bentrok	Gunakan router-id unik

L.27 Session Up, Route Tidak Muncul

Cek:

```
/ip route print
/routing filter rule print
/routing bgp connection print detail
```

Penyebab umum:

Penyebab	Solusi
Prefix tidak ada di routing table	Tambahkan route/blackhole
Filter output menolak prefix	Perbaiki filter
Filter input menolak route	Perbaiki filter
Output filter belum dipasang	Set `output.filter-chain`
Salah prefix	Cocokkan IP plan

L.28 Prefix Salah Teradvertise

Cek:

```
/routing filter rule print
/routing bgp advertisements print
```

Solusi:

- buat filter output ketat,
- accept hanya prefix yang benar,
- reject semua lainnya.

Contoh:

```
/routing filter rule add chain=bgp-out rule="if (dst ==
192.168.10.0/24) { accept }"
/routing filter rule add chain=bgp-out rule="reject"
```

L.29 Firewall BGP

Jika firewall input ketat, izinkan TCP 179 dari peer BGP.

Customer

```
/ip firewall filter add chain=input src-address=10.10.12.2
protocol=tcp dst-port=179 action=accept comment="INPUT: allow
BGP from ISP"
```

ISP

```
/ip firewall filter add chain=input src-address=10.10.12.1
protocol=tcp dst-port=179 action=accept comment="INPUT: allow
BGP from Customer"
```

Letakkan sebelum rule drop input.

Catatan:

Jangan membuka TCP 179 ke semua sumber. Batasi ke IP peer.

Bagian G — Dokumentasi dan Export

L.30 Template Dokumentasi BGP

Item	Customer	ISP
Nama router	R-CUST	R-ISP
ASN	65001	65002
Router ID	`1.1.1.1`	`2.2.2.2`
IP link	`10.10.12.1/30`	`10.10.12.2/30`
Peer address	`10.10.12.2`	`10.10.12.1`
Remote AS	65002	65001
Prefix advertised	`192.168.10.0/24`	`0.0.0.0/0` opsional
Output filter	`bgp-out`	`isp-out`
Input filter	`bgp-in`	opsional
Status session	established / belum	established / belum
Export file	`R-CUST-bgp.rsc`	`R-ISP-bgp.rsc`

L.31 Export Konfigurasi

R-CUST

```
/export file=R-CUST-bgp-final
```

R-ISP

```
/export file=R-ISP-bgp-final
```

L.32 Checklist BGP Berhasil

BGP dianggap berhasil jika:

- Customer bisa ping ISP pada link BGP.
- ISP bisa ping Customer pada link BGP.
- ASN Customer benar.
- ASN ISP benar.
- Router ID unik.
- BGP session established.
- Customer mengiklankan prefix yang benar.
- ISP menerima prefix Customer.
- Filter output aktif.
- Customer menerima default route jika dikonfigurasi.
- Tidak ada prefix tidak perlu yang keluar.
- Konfigurasi sudah di-*export*.

L.33 Golden Rules BGP Customer–ISP

- **Jangan mulai BGP sebelum IP link bisa ping.**
- **ASN sendiri dan ASN lawan harus benar.**
- **Router ID harus unik.**
- **BGP bukan pengganti OSPF.**
- **Customer hanya mengiklankan prefix miliknya.**
- **Gunakan filter output untuk mencegah route bocor.**
- **Gunakan filter input untuk membatasi route yang diterima.**
- **Customer kecil biasanya cukup menerima default route.**
- **Jangan membuka TCP 179 ke semua sumber.**
- **Gunakan ASN private dan prefix lab untuk simulasi.**
- **Setelah berhasil, export konfigurasi.**

Lampiran M. Checklist Troubleshooting

Lampiran ini dipakai sebagai **checklist cepat** saat lab MikroTik, GNS3, VPC, atau Ubuntu VM bermasalah.

Fokus troubleshooting:

- **IP**
- **route**
- **DNS**
- **NAT**
- **firewall**
- **log**

Kalimat kunci:

> **Jangan menebak. Cek berurutan: IP → route → DNS → NAT → firewall → log.**

M.1 Prinsip Troubleshooting

Gunakan prinsip berikut sebelum mengubah konfigurasi:

- **Cek dari layer bawah**: interface dulu, baru IP, route, DNS, NAT, firewall.
- **Ubah satu hal dalam satu waktu**.
- **Catat kondisi sebelum dan sesudah**.
- **Jangan reset sebelum export**.
- **Uji dari dua sisi**: router dan client.
- **Pisahkan masalah koneksi IP dan masalah nama domain**.

Sebelum eksperimen besar:

```
/export file=backup-sebelum-troubleshooting
```

M.2 Flow Cepat Troubleshooting

```
Masalah koneksi
   ↓
Cek interface/link
   ↓
Cek IP client dan router
   ↓
Cek gateway dan route
   ↓
Cek ping IP
   ↓
Cek DNS
   ↓
Cek NAT
   ↓
Cek firewall
   ↓
Cek log
```

M.3 Checklist 1 — Interface dan Link

RouterOS

```
/interface print
/interface monitor-traffic ether1
/interface monitor-traffic ether2
/ip address print
```

VPC GNS3

```
show ip
show ipv6
```

Ubuntu VM

```
ip addr
ip link
```

Yang Dicek

Pertanyaan	Ya/Tidak	Catatan
Interface yang dipakai sudah benar?		
Link GNS3 tersambung ke node yang benar?		
Interface WAN dan LAN tidak tertukar?		
Interface router punya komentar yang jelas?		
Ubuntu memakai interface yang benar?		
VPC tersambung ke interface router yang sesuai?		

Masalah umum:

- `ether1` dan `ether2` tertukar.
- VPC tersambung ke link GNS3 yang salah.
- Ubuntu VM memakai adapter yang salah.
- IP dipasang di interface yang bukan LAN.

M.4 Checklist 2 — IP Address

RouterOS

```
/ip address print
/ipv6 address print
```

VPC GNS3

```
show ip
show ipv6
```

Ubuntu VM

```
ip addr
ip -6 addr
```

Yang Dicek

Pertanyaan	Ya/Tidak	Catatan
Router punya IP di interface LAN?		
Client punya IP?		
Prefix benar?		
Client dan gateway berada dalam subnet yang sama?		
Tidak ada IP duplikat?		
IPv6 memakai prefix `/64` untuk LAN?		
IP gateway tidak memakai alamat network/broadcast?		

Contoh Salah

```
Client : 192.168.10.10/24
Gateway: 192.168.20.1

Salah karena client dan gateway berbeda subnet.
```

Contoh Benar

```
Client : 192.168.10.10/24
Gateway: 192.168.10.1
```

M.5 Checklist 3 — Gateway dan Route

RouterOS

```
/ip route print
/ipv6 route print
```

Ubuntu VM

```
ip route
ip -6 route
```

VPC GNS3

```
show ip
show ipv6
trace 1.1.1.1
```

Yang Dicek

Pertanyaan	Ya/Tidak	Catatan
Client punya gateway?		
Router punya route ke jaringan tujuan?		
Router punya default route jika butuh keluar?		
Route ke LAN lawan ada?		
Route balik dari tujuan ada?		
Gateway route bisa dijangkau langsung?		
Route OSPF/BGP muncul jika memakai dynamic routing?		

Perintah Uji

RouterOS

```
/tool ping 192.168.10.10
/tool ping 1.1.1.1
/tool traceroute 1.1.1.1
```

VPC

```
ping 192.168.10.1
trace 1.1.1.1
```

Ubuntu

```
ping 192.168.10.1
ping 1.1.1.1
traceroute 1.1.1.1
```

Catatan penting:

Jika router sendiri tidak bisa ping keluar, client juga tidak akan bisa keluar. Perbaiki route/WAN dulu sebelum menyalahkan NAT.

M.6 Checklist 4 — DNS

Masalah DNS biasanya terlihat seperti ini:

- Ping IP berhasil.
- Akses nama domain gagal.

RouterOS

```
/ip dns print
```

Ubuntu VM

```
dig example.net
cat /etc/resolv.conf
```

VPC GNS3

```
show ip
```

Yang Dicek

Pertanyaan	Ya/Tidak	Catatan
Router punya DNS server?		
`allow-remote-requests=yes` jika router jadi DNS client LAN?		
DHCP Server membagikan DNS yang benar?		
Client mendapat DNS?		
Ping IP luar berhasil?		
Nama domain gagal hanya karena DNS?		

RouterOS DNS Template

```
/ip dns set allow-remote-requests=yes servers=1.1.1.1
```

DHCP Network DNS

```
/ip dhcp-server network add address=192.168.10.0/24
gateway=192.168.10.1 dns-server=192.168.10.1
```

Catatan:

Uji koneksi keluar dengan IP dulu.
DNS diuji setelah ping IP berhasil,

M.7 Checklist 5 — DHCP

RouterOS

```
/ip dhcp-server print
/ip dhcp-server network print
/ip dhcp-server lease print
/ip pool print
```

VPC GNS3

```
ip dhcp
show ip
```

Ubuntu VM

```
ip addr
ip route
```

Yang Dicek

Pertanyaan	Ya/Tidak	Catatan
DHCP Server aktif?		
DHCP Server berada di interface LAN yang benar?		
IP pool benar?		
DHCP network sesuai subnet LAN?		
Gateway DHCP benar?		
DNS DHCP benar?		
Lease client muncul?		

Masalah umum:

- DHCP Server dibuat di interface yang salah.
- Pool tidak sesuai subnet.
- DHCP network belum dibuat.
- Client tidak menjalankan `ip dhcp`.
- Link GNS3 salah.

M.8 Checklist 6 — NAT dan Masquerade

NAT dicek jika:

- client bisa ping gateway,
- router bisa ping luar,
- tetapi client tidak bisa keluar.

RouterOS

```
/ip firewall nat print
/ip firewall connection print
/ip route print
/tool ping 1.1.1.1
```

VPC GNS3

```
ping 192.168.10.1
ping 1.1.1.1
trace 1.1.1.1
```

Ubuntu VM

```
ping 192.168.10.1
ping 1.1.1.1
traceroute 1.1.1.1
```

Yang Dicek

Pertanyaan	Ya/Tidak	Catatan
NAT masquerade ada?		
`out-interface` mengarah ke WAN?		
Router punya default route?		
Client punya gateway?		
Firewall forward mengizinkan LAN ke WAN?		
Connection tracking menunjukkan koneksi client?		

Template NAT Benar

/ip firewall nat add chain=srcnat out-interface=ether1 action=masquerade comment="NAT LAN to WAN"

Kesalahan Umum

```
/ip firewall nat add chain=srcnat out-interface=ether2
action=masquerade\
```

Salah jika ether2 adalah LAN.

Golden rule NAT:

Masquerade harus mengarah ke interface keluar, bukan interface LAN.

M.9 Checklist 7 — Firewall Input

Input chain mengatur trafik menuju router.

Contoh:

- SSH ke router.
- Ping ke router.
- DNS ke router.
- HotSpot login ke router.

RouterOS

```
/ip firewall filter print
/log print
```

Yang Dicek

Pertanyaan	Ya/Tidak	Catatan
Rule `established,related` ada?		
Rule `invalid drop` ada?		
Admin LAN diizinkan sebelum drop?		
ICMP diizinkan untuk troubleshooting?		
DNS dari LAN diizinkan jika router jadi DNS?		
Rule drop terakhir tidak memblokir admin?		
Akses dari WAN tidak terbuka sembarangan?		

Template Input Dasar

```
/ip firewall filter add chain=input
connection-state=established,related action=accept
comment="INPUT: allow established related"
/ip firewall filter add chain=input connection-state=invalid
action=drop comment="INPUT: drop invalid"
/ip firewall filter add chain=input
src-address=192.168.10.0/24 action=accept comment="INPUT:
allow LAN admin"
/ip firewall filter add chain=input protocol=icmp
action=accept comment="INPUT: allow ICMP"
/ip firewall filter add chain=input action=drop
comment="INPUT: drop other"
```

M.10 Checklist 8 — Firewall Forward

Forward chain mengatur trafik yang melewati router.

Contoh:

- VPC ke Internet.
- LAN admin ke LAN server.
- VPC-A ke VPC-C lewat OSPF.
- Client ke server hasil port forwarding.

RouterOS

```
/ip firewall filter print
/ip firewall connection print
```

Yang Dicek

Pertanyaan	Ya/Tidak	Catatan
`established,related` diizinkan?		
`invalid` di-drop?		
LAN ke WAN diizinkan?		
Trafik antar-LAN yang dibutuhkan diizinkan?		
Trafik VPN diizinkan?		
Port forwarding punya rule forward?		
Rule drop terakhir tidak terlalu awal?		

Template Forward Dasar

```
/ip firewall filter add chain=forward
connection-state=established,related action=accept
comment="FORWARD: allow established related"
/ip firewall filter add chain=forward
connection-state=invalid action=drop comment="FORWARD: drop
invalid"
/ip firewall filter add chain=forward in-interface=ether2
out-interface=ether1 action=accept comment="FORWARD: LAN to
WAN"
/ip firewall filter add chain=forward action=drop
comment="FORWARD: drop other"
```

M.11 Checklist 9 — Log

Log dipakai untuk membaca gejala, bukan sekadar melihat error.

RouterOS

```
/log print
```

Jika firewall memakai logging:

```
/ip firewall filter add chain=input action=drop log=yes
log-prefix="DROP-INPUT " comment="INPUT: drop with log"
/ip firewall filter add chain=forward action=drop log=yes
log-prefix="DROP-FORWARD " comment="FORWARD: drop with log"
```

Yang Dicek

Pertanyaan	Ya/Tidak	Catatan
Ada pesan DHCP gagal?		
Ada pesan login/SSH gagal?		
Ada drop firewall?		
Ada VPN error?		
Ada BGP/OSPF error?		
Log terlalu ramai?		

Catatan:

> **Logging drop berguna saat debugging, tetapi jangan dibiarkan terlalu ramai.**

M.12 Checklist 10 — IPv6

Jika IPv6 aktif, cek terpisah dari IPv4.

RouterOS

```
/ipv6 address print
/ipv6 route print
/ipv6 neighbor print
/ipv6 firewall filter print
```

VPC

```
show ipv6
ping 2001:db8:10::1
```

Ubuntu

```
ip -6 addr
ip -6 route
ping -6 2001:db8:10::1
```

Yang Dicek

Pertanyaan	Ya/Tidak	Catatan
IPv6 router ada di interface yang benar?		
Client punya IPv6?		
Gateway IPv6 benar?		
Prefix `/64` benar?		
ICMPv6 diizinkan?		
Firewall IPv6 aktif?		
Route IPv6 ke jaringan lawan ada?		

Golden rule IPv6:

> **Jangan aktifkan IPv6 tanpa firewall IPv6. Jangan blok ICMPv6 sembarangan.**

M.13 Checklist 11 — OSPF

RouterOS v7

```
/routing ospf neighbor print
/routing ospf interface-template print
/ip route print
```

RouterOS v6

```
/routing ospf neighbor print
/routing ospf network print
/ip route print
```

Yang Dicek

Pertanyaan	Ya/Tidak	Catatan
Router antarlink bisa saling ping?		
Router ID unik?		
Area sama?		
Network/interface-template benar?		
Neighbor muncul?		
Route OSPF muncul?		
LAN diiklankan?		
Passive interface tidak salah?		

Catatan:

Jika OSPF neighbor kosong, cek IP link antar-router dulu.

M.14 Checklist 12 — BGP

RouterOS v7

```
/routing bgp session print
/routing bgp connection print detail
/routing filter rule print
/ip route print
/log print
```

RouterOS v6

```
/routing bgp peer print
/routing bgp peer print detail
/ip route print
/log print
```

Yang Dicek

Pertanyaan	Ya/Tidak	Catatan
IP peer bisa ping?		
Local ASN benar?		
Remote ASN benar?		
Remote address benar?		
Router ID unik?		
TCP 179 tidak diblokir?		
Session established?		
Prefix ada di routing table?		
Filter output mengizinkan prefix benar?		
Filter menolak prefix lain?		

Catatan:

BGP tidak akan established jika konektivitas IP dasar ke peer belum benar.

M.15 Checklist Kasus Cepat

Kasus 1 — VPC Tidak Dapat IP

Cek:

```
/ip dhcp-server print
/ip dhcp-server network print
/ip pool print
/ip address print
```

VPC:

```
ip dhcp
show ip
```

Kemungkinan:

- DHCP salah interface.
- Pool salah.
- DHCP network belum ada.
- Link VPC salah.

Kasus 2 — Client Bisa Ping Gateway, Tidak Bisa Keluar

Cek:

```
/ip route print
/ip firewall nat print
/ip firewall filter print
/tool ping 1.1.1.1
```

Kemungkinan:

- Router tidak punya default route.
- NAT belum dibuat.
- NAT salah out-interface.
- Firewall forward drop.

Kasus 3 — Ping IP Bisa, Nama Domain Gagal

Cek:

```
/ip dns print
```

Ubuntu:

```
dig example.net
```

Kemungkinan:

- DNS router belum diatur.
- Client tidak mendapat DNS.
- Firewall memblokir DNS.

Kasus 4 — Router Tidak Bisa Diakses dari LAN

Cek dari console:

```
/ip firewall filter print
/ip address print
/ip service print
```

Kemungkinan:

- Rule input drop terlalu awal.
- Admin LAN belum diizinkan.
- Service SSH dibatasi ke subnet lain.
- IP admin berbeda subnet.

Kasus 5 — Port Forwarding Tidak Jalan

Cek:

```
/ip firewall nat print
/ip firewall filter print
/ip firewall connection print
```

Kemungkinan:

- dst-nat salah IP server.
- Forward chain belum allow.
- Server Ubuntu belum aktif.
- Port salah.

M.16 Template Laporan Troubleshooting

Gunakan format singkat ini untuk mencatat hasil troubleshooting.

```
Nama Lab:
Tanggal:
Router:
Client:

Gejala:
- ...

Topologi Singkat:
- ...

Cek Interface:
- hasil:

Cek IP:
- hasil:

Cek Route:
- hasil:

Cek DNS:
- hasil:

Cek NAT:
- hasil:

Cek Firewall:
- hasil:

Cek Log:
- hasil:

Penyebab:
- ...

Perbaikan:
- ...

Hasil Akhir:
- ...

File Export:
- ...
```

M.17 Checklist Final Setelah Perbaikan

Setelah masalah selesai:

- Uji dari router.
- Uji dari VPC.
- Uji dari Ubuntu.
- Simpan konfigurasi.
- Catat perubahan.
- Tambahkan komentar pada rule/interface.
- Export konfigurasi.

```
/export file=lab-setelah-troubleshooting
```

M.18 Golden Rules Troubleshooting

- **Jangan menebak.**
- **Cek interface sebelum IP.**
- **Cek IP sebelum route.**
- **Cek route sebelum NAT.**
- **Cek ping IP sebelum DNS.**
- **Cek NAT sebelum menyalahkan firewall.**
- **Cek firewall sebelum reset router.**
- **Cek log untuk bukti.**
- **Ubah satu hal dalam satu waktu.**
- **Setiap lab yang berhasil wajib di-export.**

Glossary

Istilah	Arti / Catatan Praktis
Access Point	Mode/perangkat yang menyediakan jaringan Wi-Fi untuk client. Dipakai pada bab **Wireless MikroTik**.
Address List	Daftar alamat IP yang dapat dipakai di rule firewall agar tidak menulis IP berulang.
Administrative Distance	Nilai prioritas route. Semakin kecil nilainya, semakin diprioritaskan. Dipakai untuk **failover**.
ARP	Mekanisme IPv4 untuk menemukan alamat MAC dari alamat IP di jaringan lokal.
ASN	*Autonomous System Number*, nomor identitas jaringan dalam BGP. Contoh lab: `65001`, `65002`.
BGP	*Border Gateway Protocol*, protokol routing antar-AS. Dipakai untuk simulasi **Customer–ISP**.
Bridge	Switch virtual di RouterOS yang menggabungkan beberapa port dalam satu jaringan Layer 2. IP gateway dipasang di **bridge**, bukan port anggota.
Broadcast Address	Alamat terakhir dalam subnet IPv4. Tidak dipakai sebagai alamat host biasa.
CHR	*Cloud Hosted Router*, RouterOS untuk mesin virtual. Dipakai di **VirtualBox** dan **GNS3**.
CIDR	Penulisan prefix jaringan, misalnya `/24`, `/30`, `/32`. Dasar membaca IP dan netmask.
Client	Perangkat pengguna yang memakai layanan jaringan. Contoh: VPC GNS3 atau Ubuntu VM.
Connection State	Status koneksi pada firewall. Contoh: `established`, `related`, `invalid`, `new`.
Connection Tracking	Mekanisme router melacak koneksi aktif. Penting untuk NAT dan firewall stateful.
Cost	Nilai biaya jalur pada OSPF. Jalur dengan cost lebih rendah biasanya dipilih.
Default Route	Route untuk semua tujuan yang tidak punya route lebih spesifik. Ditulis sebagai `0.0.0.0/0`.

DHCP Client	Fitur untuk menerima IP otomatis dari jaringan lain. Biasanya dipakai di interface WAN.
DHCP Lease	Catatan client yang mendapat IP dari DHCP Server. Dicek dengan `/ip dhcp-server lease print`.
DHCP Server	Layanan yang membagikan IP otomatis ke client. Dipakai untuk VPC dan Ubuntu Client.
DNS	Sistem penerjemah nama domain menjadi alamat IP. Router bisa menjadi DNS cache untuk client.
DNS Cache	Penyimpanan sementara hasil query DNS. Diaktifkan dengan `allow-remote-requests=yes`.
Dst-NAT	NAT yang mengubah alamat tujuan paket. Dipakai untuk *port forwarding*.
Dual-Stack	Jaringan yang menjalankan IPv4 dan IPv6 bersamaan. Berguna saat belajar IPv6 tanpa memutus IPv4.
eBGP	BGP antar-AS berbeda. Contoh: Customer AS65001 ke ISP AS65002.
ECMP	*Equal Cost Multi Path*, beberapa jalur setara menuju tujuan yang sama. Dipakai untuk pemahaman jalur ganda.
Established	Koneksi yang sudah sah dan berjalan. Biasanya diizinkan di awal rule firewall.
Export	Menyimpan konfigurasi RouterOS dalam bentuk teks. Cocok untuk dokumentasi dan belajar.
Failover	Perpindahan otomatis dari jalur utama ke jalur cadangan. Biasanya memakai `distance` dan `check-gateway`.
Firewall	Mekanisme mengizinkan atau memblokir trafik. Melindungi router dan jaringan LAN.
Forward Chain	Chain firewall untuk trafik yang melewati router. Contoh: VPC ke Internet.
Gateway	Alamat router tujuan pertama untuk keluar dari subnet. Client LAN biasanya memakai IP LAN MikroTik sebagai gateway.
GNS3	Aplikasi lab jaringan virtual. Dipakai untuk topologi CHR, VPC, dan multi-router.
Hairpin NAT	NAT saat client internal mengakses server internal memakai alamat WAN. Dikenalkan sebagai konsep lanjutan.

Host Address	Alamat IP yang dipakai perangkat dalam subnet. Tidak boleh memakai network atau broadcast address.
Host Usable	Jumlah alamat IP yang bisa dipakai host. Umumnya jumlah alamat dikurangi network dan broadcast.
HotSpot	*Captive portal* untuk autentikasi pengguna jaringan. Dipakai untuk login user sebelum akses penuh.
HotSpot Profile	Template pengaturan layanan HotSpot. Berisi alamat HotSpot dan nama DNS login.
iBGP	BGP dalam AS yang sama. Dikenalkan sebagai pembanding eBGP.
ICMP	Protokol untuk pesan kontrol jaringan IPv4. Digunakan oleh `ping`.
ICMPv6	Protokol kontrol penting di IPv6. Jangan diblokir sembarangan karena IPv6 membutuhkannya.
Input Chain	Chain firewall untuk trafik menuju router. Contoh: SSH atau ping ke MikroTik.
Interface	Port jaringan di RouterOS. Contoh: `ether1`, `ether2`, `bridge-lan`, `wg-site`.
Interface List	Daftar interface untuk menyederhanakan firewall/NAT. Contoh: `WAN`, `LAN`, `ADMIN`.
IP Address	Alamat logis perangkat di jaringan. Contoh: `192.168.10.1/24`.
IP Pool	Rentang alamat IP yang dibagikan DHCP Server atau HotSpot. Contoh: `192.168.10.100-192.168.10.200`.
IP Plan	Rencana alamat IP, gateway, DNS, dan interface. Wajib dibuat sebelum konfigurasi.
IPv4	Protokol IP 32-bit. Contoh alamat: `192.168.10.1`.
IPv6	Protokol IP 128-bit. Contoh alamat: `2001:db8:10::1`.
Layer 7 Firewall	Firewall yang mencoba membaca pola trafik aplikasi. Berat untuk CPU, hanya untuk lab atau trafik spesifik.
Link-local Address	Alamat IPv6 lokal pada satu link. Biasanya diawali `fe80::`.
L2TP/IPsec	VPN berbasis L2TP yang diamankan IPsec. Alternatif untuk RouterOS v6/v7.

LAN	*Local Area Network*, jaringan lokal. Contoh: jaringan VPC dan Ubuntu Client.
Masquerade	Source NAT otomatis memakai IP interface keluar. Cocok untuk WAN dinamis.
NAT	*Network Address Translation*, mengubah alamat IP paket. Dipakai agar IP private LAN bisa keluar.
Neighbor Discovery	Mekanisme IPv6 untuk menemukan tetangga/router lokal. Menggantikan banyak fungsi ARP pada IPv6.
Netmask	Penanda bagian network dan host pada IPv4. Contoh `/24` = `255.255.255.0`.
Network Address	Alamat pertama dalam subnet IPv4. Tidak dipakai sebagai alamat host biasa.
OSPF	Protokol routing dinamis internal berbasis *link-state*. Dipakai pada lab 3 router.
OSPF Area	Pembagian logis dalam OSPF. Lab memakai `area 0` atau backbone.
OSPF Neighbor	Router OSPF yang saling bertetangga. Dicek dengan `/routing ospf neighbor print`.
Output Chain	Chain firewall untuk trafik yang berasal dari router sendiri. Jarang diubah di lab dasar.
Passive Interface	Interface OSPF yang subnet-nya diumumkan tetapi tidak membentuk neighbor. Cocok untuk interface LAN.
Peer	Router lawan bicara dalam VPN atau BGP. Contoh: peer WireGuard atau peer BGP.
Ping	Perintah untuk menguji konektivitas. Dipakai di RouterOS, VPC, dan Ubuntu.
Port Forwarding	Meneruskan koneksi dari port router ke server internal. Menggunakan `dst-nat`.
Prefix	Panjang bagian jaringan pada IP. Contoh IPv4 `/24`, IPv6 `/64`.
Private IP	Alamat IP untuk jaringan lokal, bukan Internet publik langsung. Contoh: `192.168.0.0/16`, `10.0.0.0/8`.
Public IP	Alamat IP yang dapat dirutekan di Internet publik. Tidak digunakan sembarangan di lab.

RADIUS	Sistem autentikasi terpusat. Dikenalkan untuk HotSpot skala lebih besar.
Route Advertisement	Proses mengumumkan prefix ke protokol routing. Dipakai pada BGP dan OSPF.
Route Filtering	Penyaringan route yang masuk atau keluar. Wajib dalam BGP agar tidak terjadi route leak.
Router ID	Identitas unik router dalam OSPF/BGP. Contoh: `1.1.1.1`, `2.2.2.2`.
RouterOS	Sistem operasi MikroTik. Buku ini memakai CLI RouterOS v7 dan v6.
Routing	Proses memilih jalur paket menuju jaringan tujuan. Dasar static route, OSPF, dan BGP.
Routing Table	Tabel rute yang dipakai router untuk menentukan jalur paket. Dicek dengan `/ip route print`.
Security Profile	Profil keamanan wireless. Berisi pengaturan autentikasi dan sandi Wi-Fi.
SLAAC	*Stateless Address Autoconfiguration* pada IPv6. Client membentuk alamat IPv6 dari prefix router.
SSID	Nama jaringan Wi-Fi. Contoh: `LAB-MIKROTIK`.
Source NAT	NAT yang mengubah alamat sumber paket. Dipakai untuk LAN keluar ke WAN.
Static Route	Route manual yang ditulis administrator. Cocok untuk topologi kecil.
Subnet	Pecahan jaringan IP. Contoh: `192.168.10.0/24`.
Subnetting	Teknik membagi jaringan menjadi subnet lebih kecil. Dipakai untuk IP plan.
Switching	Pengiriman trafik dalam satu segmen Layer 2. Dipakai pada bridge dan VLAN.
Torch	Tool RouterOS untuk memantau trafik *real-time*. Dipakai saat troubleshooting.
Traceroute	Tool untuk melihat jalur paket menuju tujuan. Di VPC disebut `trace`.
Trunk Port	Port VLAN yang membawa banyak VLAN. Biasanya trafik bertag.
Tunnel	Jalur virtual di atas jaringan lain. Dipakai pada VPN.

Ubuntu VM	Mesin virtual Ubuntu untuk client/server uji. Dipakai untuk `ping`, `dig`, `curl`, `tcpdump`.
VLAN	Pemisahan jaringan secara logis pada Layer 2. Contoh VLAN admin, user, tamu.
VLAN ID	Nomor identitas VLAN. Contoh: VLAN 10, 20, 30.
VPC GNS3	*Virtual PC Simulator* di GNS3. Ringan untuk uji DHCP, ping, dan trace.
VPN	*Virtual Private Network*, tunnel aman antarjaringan. Contoh: WireGuard dan L2TP/IPsec.
WAN	*Wide Area Network*, sisi jaringan luar/upstream. Umumnya `ether1` pada lab.
Walled Garden	Daftar tujuan yang boleh diakses sebelum login HotSpot. Harus dibatasi agar tidak terlalu terbuka.
WireGuard	VPN modern yang tersedia native di RouterOS v7. Tidak tersedia native di RouterOS v6.
Wireless LAN	Jaringan lokal berbasis radio Wi-Fi. Perlu perangkat fisik dengan radio wireless.

Referensi

Canonical. (n.d.). *Install Ubuntu Server*. Retrieved May 2, 2026, from https://ubuntu.com/tutorials/install-ubuntu-server

Canonical. (n.d.). *Ubuntu Server documentation*. Retrieved May 2, 2026, from https://ubuntu.com/server/docs

Canonical. (n.d.). *Ubuntu Server*. Retrieved May 2, 2026, from https://ubuntu.com/server

Canonical. (n.d.). *Ubuntu release notes*. Retrieved May 2, 2026, from https://documentation.ubuntu.com/release-notes/

GNS3. (n.d.). *Connect GNS3 to the Internet*. Retrieved May 2, 2026, from https://docs.gns3.com/docs/using-gns3/advanced/connect-gns3-internet/

GNS3. (n.d.). *Getting started with GNS3*. Retrieved May 2, 2026, from https://docs.gns3.com/docs/

GNS3. (n.d.). *MikroTik Cloud Hosted Router appliance*. Retrieved May 2, 2026, from https://gns3.com/marketplace/appliances/mikrotik-cloud-hosted-router

GNS3. (n.d.). *VPCS*. Retrieved May 2, 2026, from https://docs.gns3.com/docs/emulators/vpcs/

GNS3. (n.d.). *Which emulator should I use?* Retrieved May 2, 2026, from https://docs.gns3.com/docs/emulators/which-emulators-should-i-use/

Internet Engineering Task Force. (1982). *Address Resolution Protocol (RFC 826)*. https://www.rfc-editor.org/rfc/rfc826

Internet Engineering Task Force. (1981). *Internet Control Message Protocol (RFC 792)*. https://www.rfc-editor.org/rfc/rfc792

Internet Engineering Task Force. (1981). *Internet Protocol (RFC 791)*. https://www.rfc-editor.org/rfc/rfc791

Internet Engineering Task Force. (1984). *Domain names—Implementation and specification (RFC 1035)*. https://www.rfc-editor.org/rfc/rfc1035

Internet Engineering Task Force. (1996). *Address allocation for private internets (RFC 1918)*. https://www.rfc-editor.org/rfc/rfc1918

Internet Engineering Task Force. (1998). *OSPF version 2 (RFC 2328)*. https://www.rfc-editor.org/rfc/rfc2328

Internet Engineering Task Force. (2006). *Classless inter-domain routing (CIDR): The Internet address assignment and aggregation plan (RFC 4632)*. https://www.rfc-editor.org/rfc/rfc4632

Internet Engineering Task Force. (2006). *IP version 6 addressing architecture (RFC 4291)*. https://www.rfc-editor.org/rfc/rfc4291

Internet Engineering Task Force. (2008). *OSPF for IPv6 (RFC 5340)*. https://www.rfc-editor.org/rfc/rfc5340

Internet Engineering Task Force. (2017). *Internet Protocol, Version 6 (IPv6) specification (RFC 8200)*. https://www.rfc-editor.org/rfc/rfc8200

Internet Engineering Task Force. (2022). *The WireGuard protocol (RFC 9228)*. https://www.rfc-editor.org/rfc/rfc9228

Internet Engineering Task Force. (2006). *A Border Gateway Protocol 4 (BGP-4) (RFC 4271)*. https://www.rfc-editor.org/rfc/rfc4271

Linux Foundation. (n.d.). *Linux networking documentation*. Retrieved May 2, 2026, from https://docs.kernel.org/networking/index.html

Linux man-pages project. (n.d.). *ip-address(8) — Linux manual page*. Retrieved May 2, 2026, from https://man7.org/linux/man-pages/man8/ip-address.8.html

Linux man-pages project. (n.d.). *ip-route(8) — Linux manual page*. Retrieved May 2, 2026, from https://man7.org/linux/man-pages/man8/ip-route.8.html

MikroTik. (n.d.). *BGP: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/328220/BGP

MikroTik. (n.d.). *Bridge VLAN table: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/328068/Bridging%2Band%2BSwitching

MikroTik. (n.d.). *Bridging and switching: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/328068/Bridging%2Band%2BSwitching

MikroTik. (n.d.). *CHR: Installing on VirtualBox*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/262864931/CHR%2Binstalling%2Bon%2BVirtualBox

MikroTik. (n.d.). *Cloud Hosted Router (CHR): RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/18350234/Cloud%2BHosted%2BRouter%2BCHR

MikroTik. (n.d.). *Command Line Interface: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/328134/Command%2BLine%2BInterface

MikroTik. (n.d.). *Configuration management: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/328155/Configuration%2BManagement

MikroTik. (n.d.). *Connection tracking: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/130220087/Connection%2Btracking

MikroTik. (n.d.). *Console: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/8978498/Console

MikroTik. (n.d.). *DHCP: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/24805500/DHCP

MikroTik. (n.d.). *DNS: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/37748767/DNS

MikroTik. (n.d.). *Failover (WAN backup): RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/26476608/Failover%2BWAN%2BBackup

MikroTik. (n.d.). *Filter: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/48660574/Filter

MikroTik. (n.d.). *Firewall: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/250708066/Firewall

MikroTik. (n.d.). *First Time Configuration: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/328151/First%2BTime%2BConfiguration

MikroTik. (n.d.). *Getting started: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/328119/Getting%2Bstarted

MikroTik. (n.d.). *HotSpot: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/56459266/HotSpot

MikroTik. (n.d.). *IP addressing: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/328247/IP%2BAddressing

MikroTik. (n.d.). *IP routing: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/328084/IP%2BRouting

MikroTik. (n.d.). *IPv4 and IPv6 fundamentals: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/119144661/IPv4%2Band%2BIPv6%2BFundamentals

MikroTik. (n.d.). *L2TP: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/2031631/L2TP

MikroTik. (n.d.). *Layer7: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/130220161/Layer7

MikroTik. (n.d.). *Load balancing: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/4390920/Load%2BBalancing

MikroTik. (n.d.). *Log: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/328094/Log

MikroTik. (n.d.). *Moving from ROSv6 to v7 with examples*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/30474256/Moving%2Bfrom%2BROSv6%2Bto%2Bv7%2Bwith%2Bexamples

MikroTik. (n.d.). *NAT: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/3211299/NAT

MikroTik. (n.d.). *OSPF: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/9863229/OSPF

MikroTik. (n.d.). *Packet Flow in RouterOS*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/328227/Packet%2BFlow%2Bin%2BRouterOS

MikroTik. (n.d.). *Packet Sniffer: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/8323088/Packet%2BSniffer

MikroTik. (n.d.). *Packages: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/40992872/Packages

MikroTik. (n.d.). *RADIUS: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/328097/RADIUS

MikroTik. (n.d.). *Route selection and filters: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/74678285/Route%2BSelection%2Band%2BFilters

MikroTik. (n.d.). *RouterOS configuration reset*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/24805498/RouterOS%2Bconfiguration%2Breset

MikroTik. (n.d.). *RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/328059/RouterOS

MikroTik. (n.d.). *Routing protocol overview: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/28606515/Routing%2BProtocol%2BOverview

MikroTik. (n.d.). *Securing your router: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/328353/Securing%2Byour%2Brouter

MikroTik. (n.d.). *Software specifications: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/19136707/Software%2BSpecifications

MikroTik. (n.d.). *SSH: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/132350014/SSH

MikroTik. (n.d.). *Torch: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/8323150/Torch

MikroTik. (n.d.). *Upgrading to RouterOS v7*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/115736772/Upgrading%2Bto%2Bv7

MikroTik. (n.d.). *VLAN: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/88014957/VLAN

MikroTik. (n.d.). *WiFi: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/224559120/WiFi

MikroTik. (n.d.). *WireGuard: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/69664792/WireGuard

MikroTik. (n.d.). *Wireless: RouterOS documentation*. Retrieved May 2, 2026, from https://help.mikrotik.com/docs/spaces/ROS/pages/8978446/Wireless

MikroTik. (n.d.). *RouterOS software*. Retrieved May 2, 2026, from https://mikrotik.com/software

Oracle. (n.d.). *Oracle VM VirtualBox*. Retrieved May 2, 2026, from https://www.virtualbox.org/

Oracle. (n.d.). *Oracle VM VirtualBox user manual*. Retrieved May 2, 2026, from https://www.virtualbox.org/manual/

Oracle. (n.d.). *Oracle VM VirtualBox user manual: Chapter 6. Virtual networking*. Retrieved May 2, 2026, from https://www.virtualbox.org/manual/ch06.html

Purbo, O. W. (2026). *Buku Pegangan Teknik Jaringan Menggunakan MikroTik, 2026* [Draft outline PDF]. Dokumen terlampir.

Ubuntu Community Help Wiki. (n.d.). *Network configuration*. Retrieved May 2, 2026, from https://help.ubuntu.com/community/NetworkConfiguration

WireGuard. (n.d.). *WireGuard: Fast, modern, secure VPN tunnel*. Retrieved May 2, 2026, from https://www.wireguard.com/

Tentang Penulis



Onno W. Purbo, saat ini bertugas sebagai rektor di Institut Teknologi Tangerang Selatan (ITTS). Onno memperoleh gelar Ph.D bidang Electrical Engineering dari University of Waterloo, Canada, adalah seorang copyleft, educator dan ICT evangelist. Dia sudah mempublikasikan 50++ buku, termasuk free ICT ebook untuk sekolah tahun 2008. Beberapa buku terakhirnya adalah "Internet-TCP/IP: Konsep Dan Implementasi", 2018; "Sistem Operasi, Konsep Dan Membuat Linux OpenWRT Dan ROM Android", 2019; "IPv6 Untuk Mendukung Operasi Jaringan Dan Domain Name System", 2019; "Kubernetes untuk Pemula", 2024 dan "Membuat Operator Seluler 5G Sendiri", 2024.

Dia memimpin sambungan pertama Internet di Institut Teknologi Bandung, tahun 1993-2000, dan menggunakannya untuk membuat jaringan Internet pendidikan yang pertama di Indonesia. Dia membebaskan frekuensi WiFi, memperkenalkan RT/RW-net, antenna Wajanbolic dan jaringan selular OpenBTS dan private 5G sendiri. Dia memimpin jaringan telepon pertama di atas Internet, VoIP Merdeka, yang kemudian hari dikenal sebagai VoIP Rakyat berbasis SIP dan menggunakan kode area +62520 dan +62521. Dia saat ini aktif memperkenalkan e-Learning, dan menjalankan server e-Learning di http://lms.onnocenter.or.id/moodle/ 66,000++ siswa / mahasiswa dan https://opencourse.itts.ac.id 39.000+ mahasiswa secara gratis. Email: onno@indo.net.id