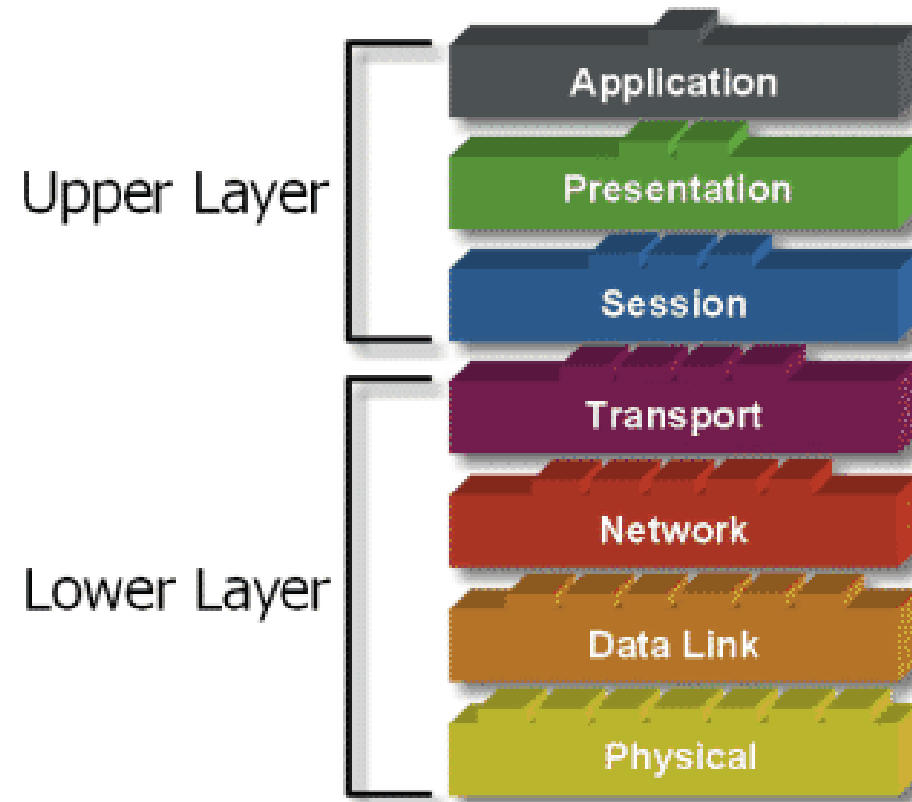


LAPISAN TRANSPORT

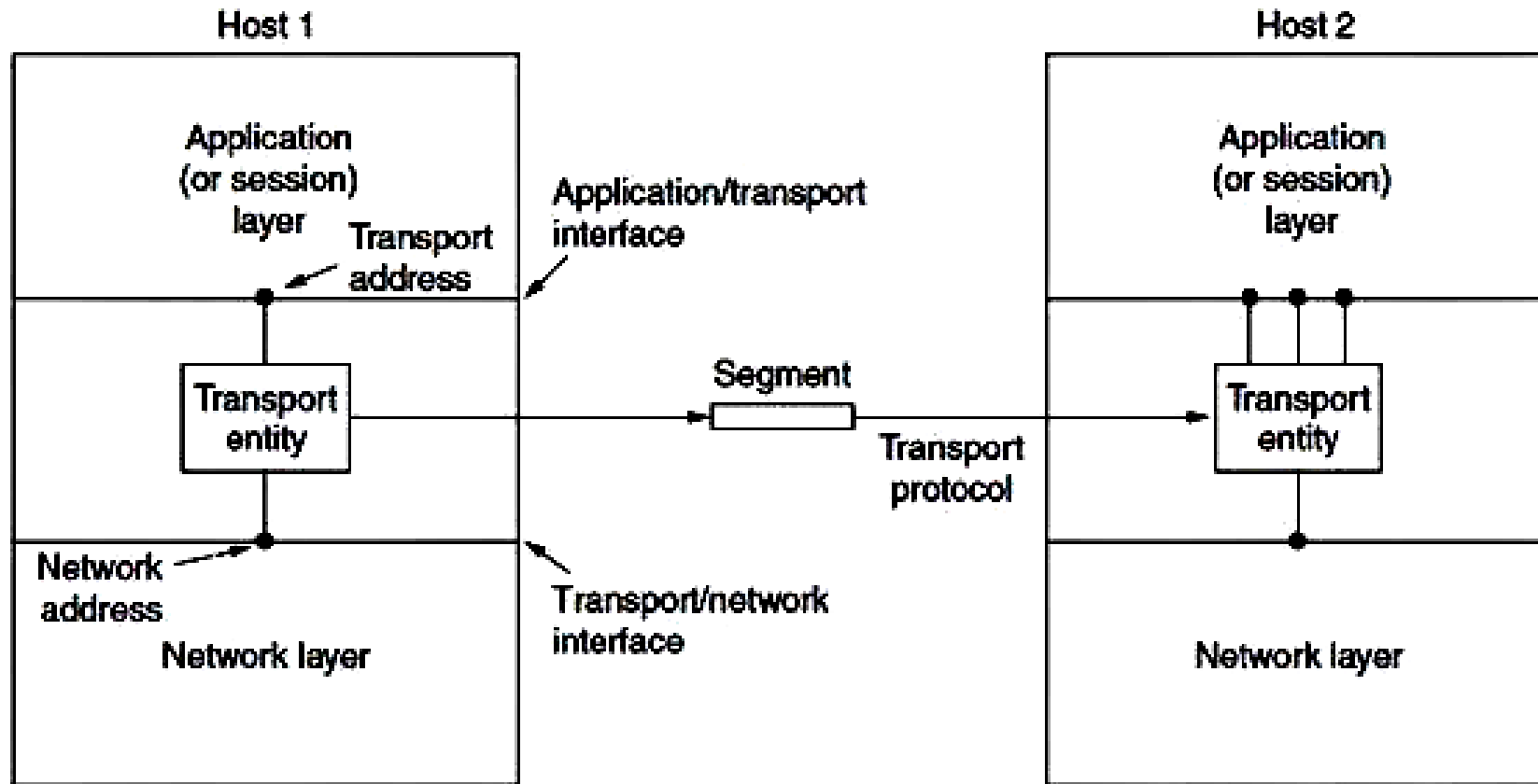
TRANSPORT LAYER

- Lapisan Transpor merupakan *lapisan ke-4* pada Model Referensi OSI yang secara umum lapisan transpor mempersiapkan data untuk di proses *pada lapisan ke-3 (Network Layer - dilihat di sisi pengirim)* dan menyiapkan data untuk diproses pada *lapisan ke-5 (Session Layer - dilihat di sisi penerima)*
- Lapisan transpor bertanggungjawab terhadap pengiriman data *proses to proses*.

TRANSPORT LAYER



SKEMA SKEMA KERJA LAYER TRANSPORT



FUNGSI LAYER TRANSPORT

1. Memberikan alamat untuk Service
2. Segmentasi dan Reassembly
3. Kontrol Koneksi
4. Kendali Aliran
5. Kendali Kesalahan
6. Multiplexing dan Demultiplexing

1. MEMBERIKAN ALAMAT UNTUK SERVICE

- Karena adanya layer transport ini maka dalam sebuah perangkat dan melalui alamat IP yang sama, sebuah client dapat menjalankan beberapa proses yang berbeda
- Misalkan pada saat yang bersamaan, user melakukan browsing suatu halaman web, mengunduh data atau file, membuka halaman sosial media dan lain-lain.



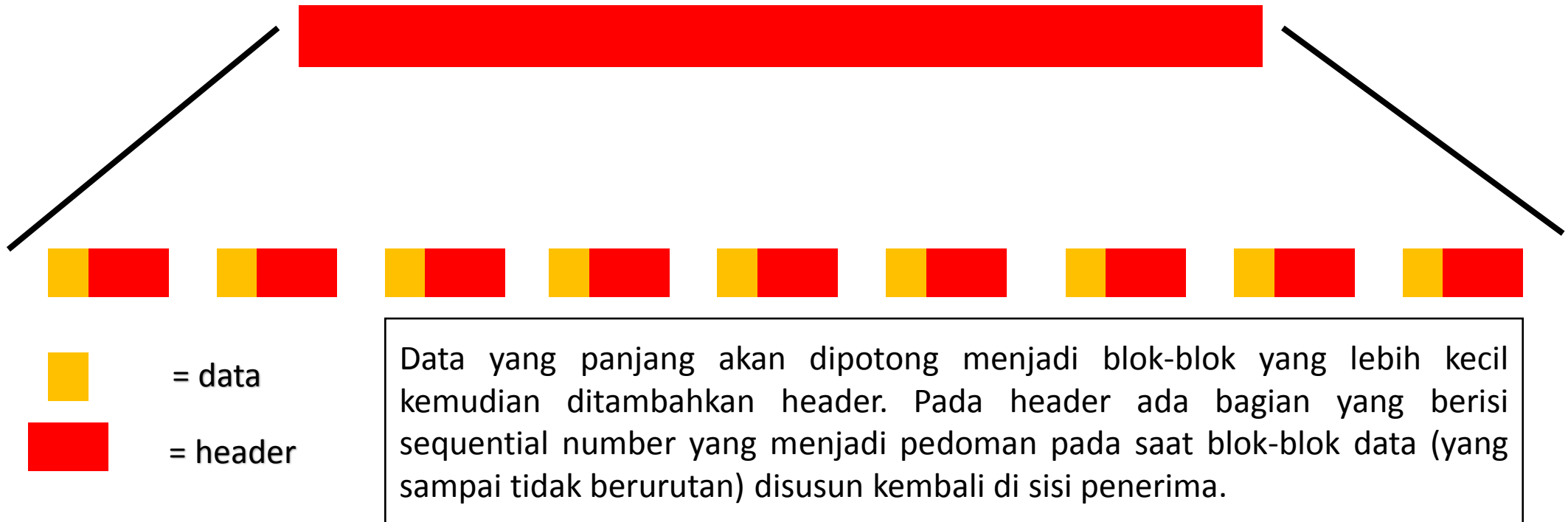
1. MEMBERIKAN ALAMAT UNTUK SERVICE

- Data yang dikirimkan untuk masing-masing proses ini tidak tertukar satu sama lain karena setiap proses memiliki alamat tambahan, selain IP yang dijalankan *dilayer network*.
- Alamat tambahan yang dimaksud adalah *alamat port*.
- Setiap proses dibedakan dengan *nomor port*, hal ini yang menyebabkan semua proses tersebut dapat berjalan secara bersamaan.

2. SEGMENTASI DAN REASSEMBLY

- Layer transport akan membagi data menjadi blok-blok yang lebih kecil, kemudian pada *masing-masing blok data akan ditambahkan sequential number yang menunjukkan nomor urut blok data* tersebut.
- Blok data ini akan dikirimkan secara independen di jaringan, sehingga bisa saja data yang sampai di penerima tidak berurutan.
- Dengan adanya *sequential number* pada *header data*, blok-blok data tadi dapat disusun kembali sesuai urutanya.

2. SEGMENTASI DAN REASSEMBLY



3. KONTROL KONEKSI

- Layer transport memungkinkan koneksi terjadi dengan cara *connection oriented* ataupun secara *Connectionless*.

4. KENDALI ALIRAN

- Berbeda dengan layer data link yang mengurus kendali aliran pada hubungan point to point, layer transport bertanggung jawab mengurus kendali aliran pada koneksi *end to end*.
- Kendali aliran ini merupakan proses untuk menyesuaikan pengiriman paket dengan kondisi penerima.
- Pada suatu waktu, penerima mungkin tidak dapat menerima paket secepat yang dikirim.

4. KENDALI ALIRAN

- Oleh karena itu dibutuhkan kemampuan kendali aliran untuk memastikan proses pengiriman data berjalan efisien, tanpa ada banyak data yang gagal terkirim karena terbatasnya kemampuan penerimaan data oleh *receiver*.

4. KENDALI ALIRAN



5. KENDALI KESALAHAN

- Data yang dikirim bisa saja mengalami perubahan selama transmisi.
- Layer transport akan memeriksa apakah data yang diterima itu mengalami ***error, loss*** atau duplikasi.
- Jika data mengalami error atau loss, maka pengirim dapat meminta data untuk dikirim ulang.
- Dan jika terjadi duplikasi maka data yang terduplikasi akan dihapus.
- Kendali kesalahan ini dilakukan ***process to process***

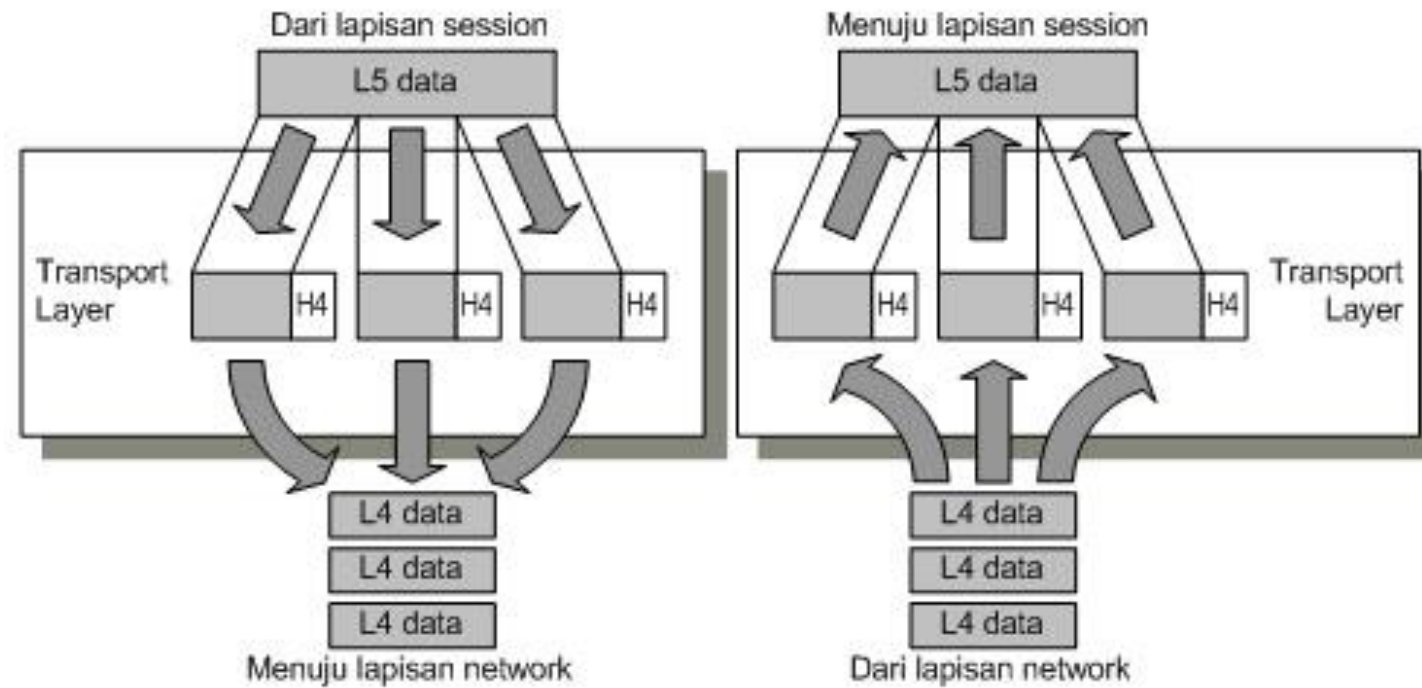
5. KENDALI KESALAHAN



6. MULTIPLEXING DAN DEMULTIPLEXING

- Jika user mengakses beberapa *service* dalam sebuah server, maka lapis transport pada server akan melakukan *multiplexing* atau penggabungan beberapa layanan untuk dikirim secara bersamaan ke user.
- Sampai di host, beberapa *service* yang sebelumnya telah di *multiplexing* akan dipisahkan kembali dan di *forward* ke alamat *port* masing-masing sesuai prosesnya.

6. MULTIPLEXING DAN DEMULTIPLEXING



TIPE TRANSFER DATA

1. Connectionless vs Connection Oriented
2. Reliable vs Unreliable
3. Sateful vs Stateless

1. CONNECTIONLESS VS CONNECTION ORIENTED

- Pada ***connectionless***, saat data hendak dikirimkan tidak diperlukan pembentukan hubungan terlebih dahulu.
- Selama hubungan komunikasi selesai tidak diperlukan penutupan atau terminasi.
- Sebaliknya pada ***connection oriented***, dibutuhkan pembentukan hubungan terlebih dahulu sebelum komunikasi dapat dijalankan.
- Selama komunikasi akan dilakukan maintenance, dan saat komunikasi selesai akan dilakukan penutupan.

2. RELIABLE VS UNRELIABLE

- Tipe hubungan dikatakan *reliable* jika data dapat dipastikan sampai ke semua penerima dengan benar.
- Jika ternyata data mengalami *error* saat pengiriman, maka data tersebut harus dapat dikirim ulang.
- Pada tipe *unreliable*, data tidak dapat dipastikan semuanya akan sampai ke tujuan dengan benar.
- Jika terdapat *error* atau *loss* saat pengiriman data, data ini tidak dapat diminta untuk dikirim ulang.

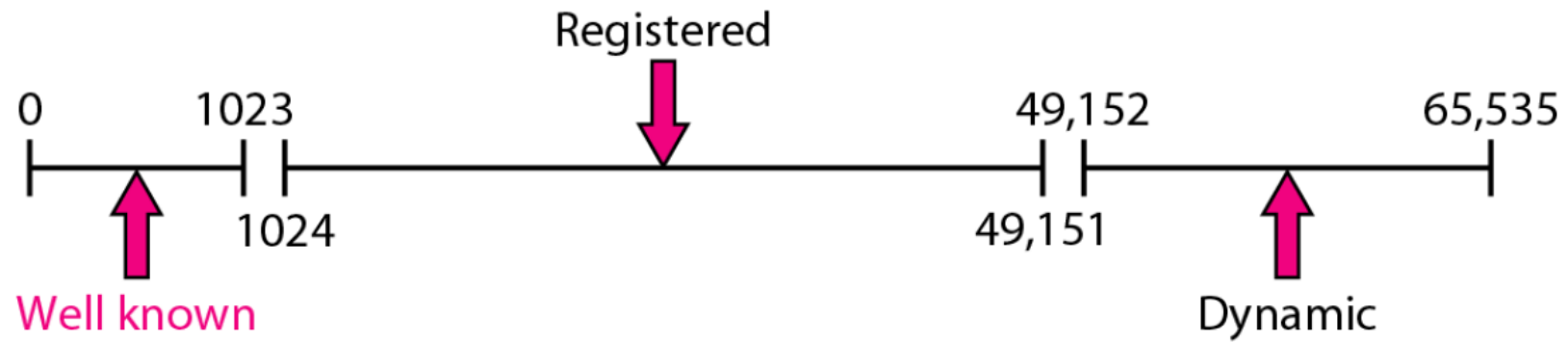
3. STATEFUL VS STATELESS

- Pengiriman data *stateful* berarti komunikasi yang dimasukan kedalam satu *request*, yang dikirimkan dari pengirim ke penerima, dapat dimodifikasi untuk *request* berikutnya.
- Sehingga tidak perlu konfirmasi ulang dari awal untuk setiap *request*.
- Pengiriman *stateless* berarti informasi dalam satu request tidak dapat dikaitkan dengan *request* lainnya, sehingga tidak adapat digunakan untuk *request* lainnya.

ALAMAT PORT

1. Well – Known Port
2. Registered Port
3. Dynamic/ Private Port

ALAMAT PORT



WELL-KNOWN PORT

- Yaitu nomor port **0-1023**, port-port ini direservasi untuk aplikasi dan service tertentu.
- Misalkan untuk HTTP menggunakan nomor port 80, POP3 port 110, Telnet port 23, SMTP port 25, IRC port 194, TFTP port 69, FTP port 21 dan lain-lain

REGISTERED PORT

- Yaitu nomor port **1024-49151**, port-port ini direservasi untuk proses di user, untuk menginstal dan sebagainya.
- Contohnya aplikasi MySQL menggunakan nomor port 1433, WAP menggunakan port 2948 dan lain-lain

DYNAMIC/ PRIVATE PORT

- Yaitu nomor port **49152-65535**, port-port ini digunakan untuk inisialisasi koneksi, koneksi peer to peer untuk sharing file dan lain sebagainya.
- Port-port ini dapat dipilih secara dinamis oleh client sebagai *source port*