

Sessions in a Glance

Samsung Innovation Campus 7



Samsung Innovation 7 Journey



Stage 1

Code Kickstart

- Getting Started with Python
- Variables & Data Types
- Control Flow: If-Else
- Loops & Patterns
- Functions & Reusability
- Lists & Dictionaries
- File Handling & CSV
- Hands-on Session: Python Project

Stage 2

Device & Hardware discovery

- Introduction to IoT
- ESP32 Setup
- Input Sensors
- Output Devices
- Serial Communication
- MQTT & HiveMQ
- Python Dashboard
- Hands-on Session: Mini IoT Project

Stage 3

AI

- Intro to AI/ML
- Data Preparation
- Building a Model
- Evaluating Model
- ML + Sensor Integration
- Visualization
- Edge AI
- Hands-on Session: Capstone Planning

Code Kickstart Recap

Variables & Data Types

```
# Real sensor reading
```

```
suhu = 28.5          # float
```

```
kelembapan = 65      # int
```

```
waktu = "08:00"      # str
```

```
aktif = True          # bool
```

```
# Operations
```

```
temp_f = (suhu * 9/5) + 32 # Convert to Fahrenheit
```

```
print(f"Ruangan {waktu}: {suhu}°C = {temp_f:.1f}°F")
```

Control Flow & Loops

```
# If/elif/else: Decision logic (seperti di ESP32)
```

```
if suhu > 30:  
    action = "AC ON"  
elif suhu < 18:  
    action = "Heater ON"  
else:  
    action = "Normal"
```

```
# For loop: Process multiple readings
```

```
readings = [25, 28, 32, 30, 26]  
for temp in readings:  
    if temp > 28:  
        print(f"⚠ {temp}°C - Hot!")
```

Functions & Reusability

```
def classify_temp(temperature):  
    """Classify temp for automation"""  
    if temperature > 30:  
        return "Panas"  
    elif temperature < 18:  
        return "Dingin"  
    else:  
        return "Normal"  
  
def avg_readings(data):  
    """Calculate average temperature"""  
    return sum(data) / len(data)  
  
# Usage  
temps = [25, 28, 30, 27]  
avg = avg_readings(temps) # 27.5  
status = classify_temp(avg) # "Normal"
```

File I/O & CSV

```
# Write sensor log to CSV
with open("sensor.csv", "w") as f:
    f.write("timestamp,temperature,humidity\n")
    f.write("08:00,25.5,60\n")
    f.write("08:10,26.0,62\n")

# Read & process CSV
with open("sensor.csv", "r") as f:
    for line in f:
        print(line.strip())
```

Three overlapping circles in orange, blue, and a larger yellow-orange color are positioned on the right side of the slide.

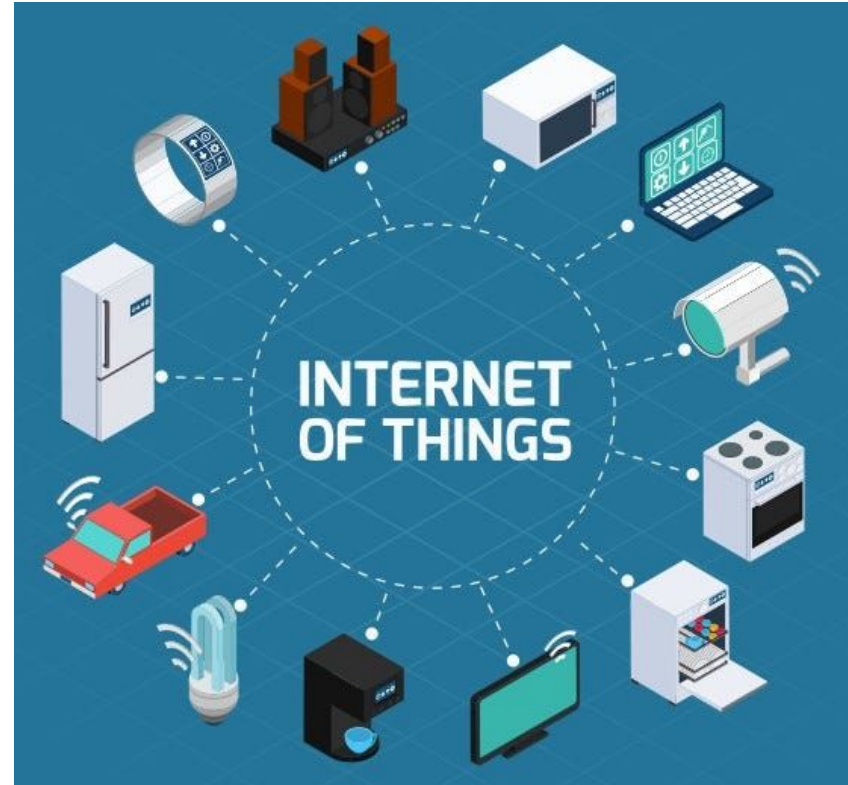
Device & Hardware discovery

Internet of Things

Internet of Things (IoT) adalah konsep jaringan perangkat/benda fisik yang terhubung ke internet (atau jaringan) sehingga bisa mengumpulkan, mengirim, dan bertukar data, lalu menjalankan aksi tertentu secara otomatis atau jarak jauh

Unsur penting dalam ekosistem IoT

- Perangkat Pintar (Smart Devices). Benda fisik dengan sensor dan konektivitas.
- Konektivitas. Jaringan untuk transfer data (internet, RFID, dll.).
- Platform IoT. Layanan cloud untuk menyimpan, mengelola, dan menganalisis data.
- Aplikasi. Antarmuka pengguna untuk mengontrol dan memantau perangkat.



IoT Architecture

Layer 4: User Application

Layer 3: Cloud / Backend Server

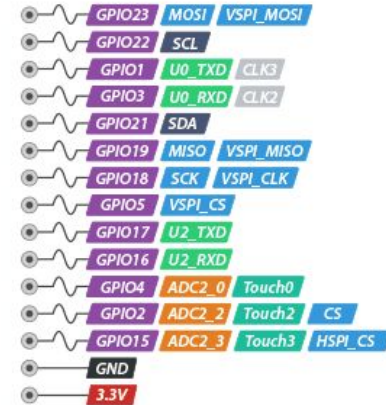
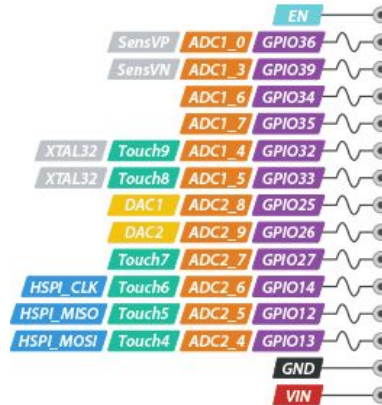
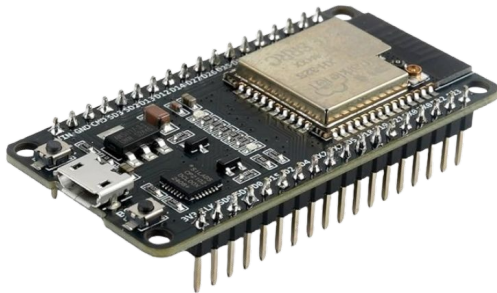
Layer 2: Edge Device (ESP32)

Layer 1: Sensors & Actuators

ESP32

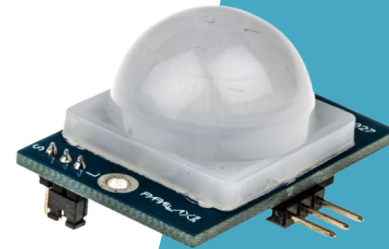
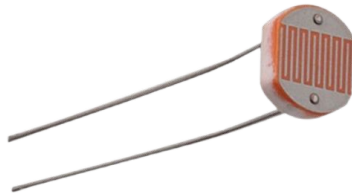
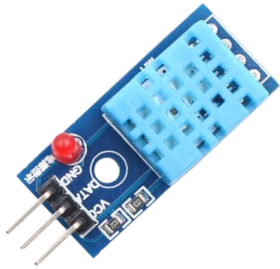
ESP32 adalah mikrokontroler berharga rendah dan hemat energi dengan wifi dan dual-mode bluetooth terintegrasi (bluetooth classic dan bluetooth low energy).

- Dual-core 240MHz processor
- 34+ GPIO pins
- Built-in WiFi + Bluetooth
- 520KB RAM, 4MB Flash memory
- USB for programming & serial
- 3.3V operating voltage

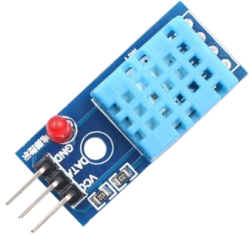


Sensors

Sensor adalah perangkat yang mendeteksi perubahan pada lingkungan fisik atau kimia (seperti cahaya, panas, gerakan, tekanan, atau gas) dan mengubahnya menjadi sinyal listrik atau data lain yang dapat dibaca, diproses, atau ditampilkan oleh sistem elektronik atau manusia



Common IoT Sensors



DHT11 (Temperature + Humidity)

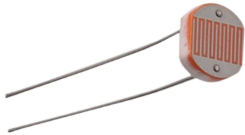
Type: Digital sensor

Output: Temp (0–50°C) + Humidity (20–90%)

Accuracy: $\pm 2^{\circ}\text{C}$, $\pm 5\% \text{ RH}$

Interface: Single-wire protocol

Use: Climate monitoring, smart home



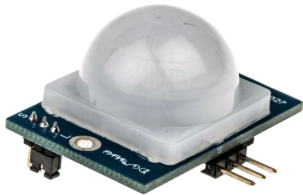
LDR (Light Dependent Resistor)

Type: Analog sensor

Output: Light intensity (0–4095 ADC)

Response: Decreases resistance in bright light

Use: Auto brightness, day/night detection



PIR HC-SR501 (Motion Sensor)

Type: Digital sensor

Output: HIGH when motion detected, LOW otherwise

Range: ~7 meters

Use: Security, auto lighting

Output/Actuators

Output device adalah komponen elektronik yang memberikan respon atau reaksi terhadap input atau kondisi sistem. Dalam ekosistem IoT, output device berperan sebagai jembatan antara data digital dan aksi fisik yang dapat diamati atau dirasakan oleh pengguna.

Output device mengubah sinyal listrik dari microcontroller menjadi berbagai bentuk output yang bermakna, seperti cahaya, suara, atau pergerakan mekanis.



Communication Protocols



HTTP (HyperText Transfer Protocol)

- Model: Request-Response (client-server)
- Connection: Temporary (per request)
- Bandwidth: Large headers, overhead
- Latency: Higher (100ms-1s)
- Power: Consumes more energy
- Use: Web services, REST APIs

HTTP Example:

ESP32 → POST /api/sensor → Server → Database
(send JSON: {"temp": 28.5, "humidity": 65})



MQTT (Message Queuing Telemetry Transport)

- Model: Publish-Subscribe (many-to-many)
- Connection: Persistent (always connected)
- Bandwidth: Very efficient (small payload)
- Latency: Low, real-time
- Power: Optimized for IoT (low power consumption)
- Use: Sensor streaming, real-time control

MQTT Example:

ESP32 → MQTT Broker → Cloud → Python subscriber
(publish "iot/room/temp" : 28.5)

IoT dataflow example

Real-time scenario (every 10 minutes):

1. ESP32 reads DHT11 sensor
 - temp = 28.5°C, humidity = 62%
2. ESP32 publishes to MQTT broker
 - topic: "home/livingroom/dht11"
 - payload: {"temp": 28.5, "humid": 62, "time": "08:00"}
3. Cloud server (Firebase) receives & stores in database
4. Python script fetches latest data
 - Runs inference on ML model
 - Predicts next 2 hours
5. Dashboard displays:
 - Current temp + humidity
 - Predicted temp (2h ahead)
 - Recommended action (AC on/off)
6. If auto-mode: ESP32 receives command
 - Activates relay (turns AC on/off)
7. Cycle repeats every 10 minutes

Artificial Intelligence Machine Learning



Why combine IoT + ML



Scenario 1 - IoT without ML (Reactive)

09:00 AM: Room temp is 30°C

Decision: "Temp > 30°C, turn AC ON now"

Problem: Room already hot, uncomfortable

Inefficient: AC works hard to cool down

Scenario 2: IoT + ML (Proactive)

History (30 days pattern):

- 08:00 AM = 25°C → usually rises to 32°C by 12:00 PM
- Peak at 2:00 PM (32–34°C)
- Cools down after 5:00 PM

ML Decision at 10:00 AM:

- Current: 28°C
- Model predicts: 32.5°C at 12:00 PM (2h ahead)
- Action: Turn AC ON NOW (pre-cooling)
- Result: Room stays 27–28°C throughout (comfortable)
- Benefit: ~25% less energy (AC runs fewer hours, less hard)

ML learns patterns → predict future → proactive action → save energy + comfort

ML Workflow in IoT

1 - Collect data

(30 days sensor readings = 4320 points)

2 - Prepare data

(clean, normalize, split 80/20)

3 - Train model

(algorithm learns pattern)

4 - Test model

(check accuracy on unseen data)

5 - Deploy model

(use daily for predictions)

6 - Monitor

(retrain weekly if accuracy drops)

ML tasks untuk IoT

Regression: Predict continuous number

- Task: Predict temperature (26.5°C)
- Algorithm: Linear Regression, Decision Tree
- Metric: MAE, RMSE
- IoT Example: Predict room temp for next 2 hours

Classification: Predict category

- Task: Predict "Hot" / "Normal" / "Cold"
- Algorithm: Logistic Regression, Random Forest
- Metric: Accuracy, Precision, Recall
- IoT Example: Classify day as hot/normal/cold

ML tasks untuk IoT

Anomaly Detection: Find unusual patterns

- Task: Detect sensor malfunction or unusual behavior
- Algorithm: Isolation Forest, Autoencoders
- Metric: Detection rate, False positive rate
- IoT Example: Alert if temp spikes unexpectedly

Time-Series Forecasting: Predict sequence

- Task: Predict next 24h temperature trend
- Algorithm: LSTM, Prophet
- Metric: MAPE, RMSE
- IoT Example: Forecast energy usage

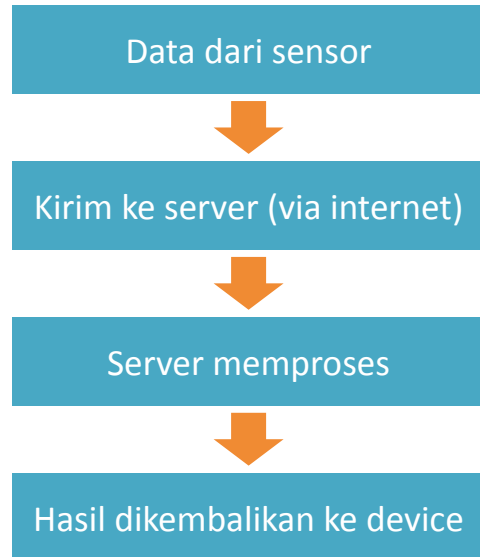
Evaluation matrix

		Predicted values		
		True	False	
Actual	True	True Positive (TP)	False Negative (FN) Type 1 Error	$Recall = Sensitivity = \frac{TP}{TP + FN}$
	False	False Positive (FP) Type 1 Error	True Negative (TN)	$Specificity = \frac{TN}{TN + FP}$
		$Precision = \frac{TP}{TP + FP}$		$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$ $F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$

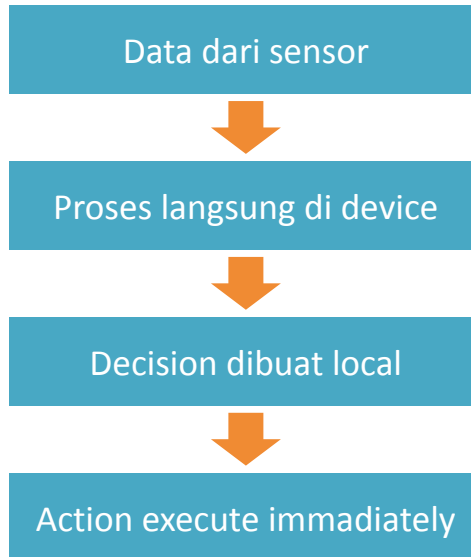
Edge AI

Edge AI adalah eksekusi Machine Learning model langsung di perangkat IoT edge device (ESP32, Jetson Nano, Raspberry Pi) tanpa harus mengirimkan semua data ke cloud server

Cloud AI



Edge AI



Edge AI

Aspek	Cloud AI	Edge AI
Location	AWS, Google Cloud, Azure	ESP32, Jetson Nano, Smartphone
Internet Dependency	WAJIB ada	TIDAK wajib (offline ok)
Latency	100ms - 1+ detik	10-50ms
Model Complexity	Kompleks (Deep Learning)	Simple-Medium
Privacy	Data ke server (risky)	Data tetap di device (aman)
Bandwidth	Tinggi (upload raw data)	Rendah (hanya inference result)
Power	Boros (upload stream)	Hemat (local processing)
Cost	Subscription cloud	Hardware one-time
Usecase	Training, big data analytics	Real-time alert, offline decision

Edge AI Pipeline



A large black parallelogram is positioned on the left side of the slide. Below it, two overlapping parallelograms in light orange and a darker orange shade are also positioned on the left, creating a layered effect.

Terima kasih