

SAMSUNG INNOVATION CAMPUS BATCH 6

MATERIAL

Pretest – 1

Revolusi Industri terdiri dari empat fase utama:

1. **Industri 1.0** (akhir abad ke-18) – Ditandai dengan mesin uap dan mekanisasi produksi.
2. **Industri 2.0** (akhir abad ke-19) – Muncul listrik, jalur perakitan, dan produksi massal.
3. **Industri 3.0** (pertengahan abad ke-20) – Era komputer, otomasi, dan internet.
4. **Industri 4.0** (awal abad ke-21) – Menggunakan AI, IoT, dan Big Data untuk otomatisasi cerdas.

(**Sensor, Aktuator, AI, dan Controller**) adalah bagian dari **komponen IoT**. Berikut penjelasannya:

1. **Sensor** – Mengumpulkan data dari lingkungan, seperti suhu, kelembaban, atau gerakan.
2. **Aktuator** – Menjalankan perintah berdasarkan data, misalnya motor yang bergerak atau lampu yang menyala.
3. **Controller** – Mengontrol dan mengelola komunikasi antara sensor dan aktuator, sering berupa **Mikrokontroler (Arduino, ESP32) atau Mikroprosesor (Raspberry Pi)**.
4. **AI (Artificial Intelligence)** – Menganalisis data IoT untuk membuat keputusan otomatis, misalnya prediksi cuaca atau deteksi pola perilaku. (Di Pretest ini dikecualikan.)

Berikut beberapa contoh sensor dalam **IoT** dan fungsinya:

- **Sensor Suhu (Temperature Sensor)** – Mengukur suhu, digunakan di **AC pintar & sistem pemantauan cuaca**.
- **Sensor Kelembaban (Humidity Sensor)** – Mendeteksi kelembaban udara, dipakai dalam **pertanian cerdas & HVAC**.

- **Sensor Gerak (Motion Sensor/PIR)** – Mendeteksi pergerakan, sering digunakan dalam **CCTV & sistem keamanan pintar**.
- **Sensor Cahaya (Light Sensor/LDR)** – Mengukur intensitas cahaya, digunakan dalam **lampu otomatis & smart blinds**.
- **Sensor Gas (Gas Sensor/MQ Series)** – Mendeteksi gas berbahaya, dipakai dalam **alarm kebocoran gas & monitoring kualitas udara**.
- **Sensor Jarak (Ultrasonic Sensor)** – Menggunakan gelombang ultrasonik untuk mengukur jarak atau mendeteksi objek, digunakan dalam **mobil otonom & parkir pintar**.
- **Sensor Tekanan (Pressure Sensor)** – Mengukur tekanan udara atau cairan, digunakan di **monitor ban & industri hidrolik**.
- **Sensor Infrared (IR Sensor)** – Menggunakan sinar inframerah untuk mendeteksi objek atau panas, dipakai dalam **remote TV, sistem keamanan, dan robot line follower**.
- **Sensor Gyro (Gyroscope Sensor)** – Mengukur orientasi, rotasi, dan kemiringan objek, digunakan dalam **smartphone, drone, dan VR headset**.

Berikut beberapa **IoT Platform** yang populer beserta fungsinya:

1. **AWS IoT Core** – Platform dari Amazon untuk menghubungkan perangkat IoT ke cloud dengan fitur AI dan analitik.
2. **Google Cloud IoT** – Layanan IoT dari Google yang menyediakan analitik, machine learning, dan koneksi ke layanan Google Cloud lainnya.
3. **Microsoft Azure IoT Hub** – Platform IoT dari Microsoft yang mendukung komunikasi antar perangkat dengan keamanan tinggi.
4. **IBM Watson IoT** – Memungkinkan pengolahan data IoT dengan AI dan analitik berbasis IBM Watson.
5. **ThingSpeak** – Platform IoT open-source yang digunakan untuk mengolah dan menampilkan data sensor secara real-time.
6. **Kaa IoT** – Platform open-source yang fleksibel untuk membangun solusi IoT skala besar.
7. **Blynk** – Platform IoT berbasis cloud yang banyak digunakan untuk proyek DIY dan smart home.
8. **Arduino IoT Cloud** – Didesain untuk perangkat berbasis Arduino dengan kemudahan integrasi sensor dan aktuator.

Kecerdasan manusia adalah kemampuan berpikir, memahami, belajar, bernalar, memecahkan masalah, beradaptasi dengan lingkungan, serta menggunakan pengetahuan dan pengalaman untuk mengambil keputusan. Kecerdasan ini mencakup berbagai aspek, seperti **logika, kreativitas, emosi, dan keterampilan sosial**, yang membedakan manusia dari makhluk lain.

Kecerdasan manusia dapat dikategorikan ke dalam beberapa jenis, seperti:

- **Kecerdasan logis-matematis** (analisis & pemecahan masalah)
- **Kecerdasan linguistik** (kemampuan berbahasa & komunikasi)
- **Kecerdasan emosional** (memahami & mengelola emosi)
- **Kecerdasan kinestetik** (koordinasi tubuh & gerakan)
- **Kecerdasan sosial** (interaksi & memahami orang lain)

Mesin bisa belajar (Machine Learning) adalah cabang dari kecerdasan buatan (AI) yang memungkinkan mesin untuk belajar dari data tanpa pemrograman eksplisit. Mesin belajar dengan menganalisis pola dalam data dan membuat prediksi atau keputusan berdasarkan data tersebut.

Ada tiga jenis utama machine learning:

1. **Supervised Learning:** Mesin dilatih dengan data yang sudah memiliki label (jawaban yang benar). Mesin belajar untuk memetakan input ke output.
 - Contoh: Prediksi harga rumah berdasarkan data harga sebelumnya.
2. **Unsupervised Learning:** Mesin dilatih dengan data tanpa label, dan bertugas menemukan pola atau struktur dalam data tersebut.
 - Contoh: Pengelompokan pelanggan berdasarkan perilaku belanja mereka.
3. **Reinforcement Learning:** Mesin belajar dengan mencoba berbagai tindakan dan menerima umpan balik dalam bentuk reward atau punishment untuk belajar strategi yang optimal.
 - Contoh: Program game yang belajar cara memenangkan permainan melalui percakapan.

Dengan machine learning, mesin dapat meningkatkan kinerjanya seiring dengan bertambahnya data tanpa perlu diprogram ulang secara manual.

Machine Learning vs Traditional Programming:

1. Machine Learning:

- Mesin belajar dari data untuk membuat prediksi atau keputusan.
- Tidak perlu perintah eksplisit untuk setiap langkah, karena algoritma dapat "belajar" pola dari data.
- Cocok untuk tugas yang kompleks atau tidak bisa diprogram secara langsung, seperti pengenalan wajah, prediksi cuaca, atau rekomendasi produk.

Contoh: Menggunakan data penjualan masa lalu untuk memprediksi penjualan di masa depan.

2. Traditional Programming:

- Programmer menulis instruksi atau kode secara manual untuk setiap langkah.
- Mesin mengikuti aturan yang ditulis secara eksplisit oleh programmer.
- Cocok untuk masalah yang lebih sederhana dan dapat dijelaskan secara jelas dengan logika atau aturan tetap.

Contoh: Program untuk menghitung total harga belanja dengan rumus yang sudah ditentukan.

Perbedaan utama:

- **Machine Learning:** Mesin belajar dari data dan pola.
- **Traditional Programming:** Programmer memberikan instruksi langkah demi langkah.

Machine Learning Cycle adalah proses berulang yang digunakan untuk mengembangkan model machine learning. Berikut langkah-langkahnya secara singkat:

1. **Definisikan Masalah:** Tentukan masalah yang ingin diselesaikan, misalnya prediksi, klasifikasi, atau rekomendasi.
2. **Kumpulkan Data:** Kumpulkan data yang relevan untuk melatih model.
3. **Persiapkan Data:** Lakukan pembersihan dan pra-pemrosesan data (misalnya menghapus nilai hilang, normalisasi, dll).
4. **Pilih Model:** Pilih algoritma machine learning yang sesuai (misalnya regresi, decision tree, neural networks).
5. **Latih Model:** Latih model dengan data yang sudah dipersiapkan untuk mengidentifikasi pola.

6. **Evaluasi Model:** Uji model dengan data yang belum terlihat dan evaluasi kinerjanya menggunakan metrik seperti akurasi, presisi, atau F1-score.
7. **Perbaiki Model:** Sesuaikan model berdasarkan hasil evaluasi untuk meningkatkan kinerja (misalnya dengan tuning parameter atau menggunakan data tambahan).
8. **Deploy Model:** Terapkan model dalam lingkungan nyata untuk membuat prediksi atau keputusan.
9. **Monitor & Update:** Terus pantau kinerja model dan perbarui dengan data baru bila diperlukan.

Proses ini dilakukan secara berulang untuk meningkatkan akurasi dan kinerja model seiring waktu.

Pengaplikasian AI digunakan dalam berbagai bidang untuk meningkatkan efisiensi dan automasi. Berikut beberapa contoh singkat:

1. **Pengolahan Gambar dan Pengenalan Wajah:**
 - Digunakan dalam aplikasi seperti Face ID, sistem keamanan, dan identifikasi objek.
2. **Asisten Virtual (Chatbots):**
 - Seperti Siri, Alexa, atau Google Assistant yang membantu menjawab pertanyaan dan mengatur tugas.
3. **Sistem Rekomendasi:**
 - Digunakan di platform seperti Netflix, Amazon, dan Spotify untuk memberikan rekomendasi produk atau konten berdasarkan preferensi pengguna.
4. **Mobil Otonom:**
 - Kendaraan yang dapat mengemudi sendiri dengan menggunakan AI untuk memproses data dari sensor dan kamera.
5. **Penyaringan Data:**
 - AI digunakan untuk memproses dan menganalisis data besar (big data) untuk menemukan pola atau wawasan yang berguna dalam bidang keuangan, medis, dan pemasaran.
6. **Deteksi Anomali:**
 - Digunakan untuk mendeteksi penipuan dalam transaksi keuangan atau anomali dalam jaringan komputer.

AI meningkatkan produktivitas, automasi tugas, dan memberikan solusi pintar dalam kehidupan sehari-hari dan industri.

AI Do's (Hal yang Harus Dilakukan):

1. **Pastikan Data Berkualitas:** Gunakan data yang relevan, bersih, dan terstruktur untuk pelatihan model AI.
2. **Transparansi dan Kejelasan:** Jelaskan bagaimana model AI bekerja dan keputusan yang diambil.
3. **Pertimbangkan Etika:** Pastikan AI digunakan dengan cara yang etis dan tidak merugikan individu atau kelompok tertentu.
4. **Perbarui dan Pemeliharaan:** Terus monitor dan update model AI agar tetap relevan dan akurat seiring waktu.
5. **Keamanan:** Lindungi model AI dari potensi penyalahgunaan dan pastikan data aman.

AI Don'ts (Hal yang Tidak Boleh Dilakukan):

1. **Menggunakan Data Bias:** Hindari penggunaan data yang bias atau diskriminatif yang bisa memperburuk ketidakadilan.
2. **Mengabaikan Privasi:** Jangan gunakan data pribadi tanpa izin atau melanggar privasi individu.
3. **Mempercayai Model Tanpa Verifikasi:** Jangan hanya mengandalkan hasil model AI tanpa evaluasi atau verifikasi dari manusia.
4. **Mengabaikan Potensi Dampak Sosial:** Jangan abaikan dampak sosial yang bisa ditimbulkan dari penggunaan AI, seperti pengangguran atau ketimpangan.
5. **Menggunakan AI untuk Tujuan Jahat:** Hindari penggunaan AI untuk penipuan, manipulasi, atau aktivitas ilegal.

AI harus digunakan secara bijak dan bertanggung jawab agar memberikan manfaat yang optimal tanpa menimbulkan risiko atau masalah.

Definisi Kecerdasan Buatan (Artificial Intelligence - AI)

Kecerdasan Buatan (AI) adalah cabang ilmu komputer yang memungkinkan mesin untuk **meniru kecerdasan manusia**, seperti berpikir, belajar, dan mengambil keputusan secara otomatis. AI bekerja dengan **algoritma, data, dan model pembelajaran** untuk mengolah informasi dan menyelesaikan masalah tanpa intervensi manusia langsung.

Jenis Kecerdasan Buatan

1. **AI Lemah (Narrow AI)** – Dirancang untuk tugas spesifik, seperti **asisten virtual (Siri, Alexa)**, **rekomendasi film (Netflix)**, dan **pengenalan wajah**.
2. **AI Kuat (General AI)** – Memiliki kemampuan berpikir seperti manusia dan dapat memecahkan berbagai masalah. (Masih dalam tahap penelitian).
3. **Super AI** – AI yang lebih cerdas dari manusia dalam segala aspek (konsep futuristik yang belum terwujud).

Komponen Utama AI

- **Machine Learning (ML)** – AI belajar dari data untuk meningkatkan akurasi.
- **Deep Learning (DL)** – Model AI yang meniru cara kerja otak dengan jaringan saraf tiruan.
- **Natural Language Processing (NLP)** – Kemampuan AI memahami dan merespons bahasa manusia (contoh: ChatGPT).
- **Computer Vision** – Memungkinkan AI mengenali objek dan gambar, seperti dalam mobil otonom.

Clustering adalah teknik pembelajaran tanpa pengawasan yang mengelompokkan data berdasarkan kesamaan, seperti segmentasi pelanggan.

Regression adalah teknik pembelajaran terawasi yang digunakan untuk memprediksi nilai kontinu, seperti harga rumah atau suhu.

Classification juga merupakan pembelajaran terawasi yang mengklasifikasikan data ke dalam kategori, misalnya untuk deteksi spam atau klasifikasi penyakit.

Reinforcement Learning melibatkan pembelajaran melalui trial and error, di mana agen belajar memilih tindakan terbaik berdasarkan reward, contohnya dalam AI bermain game atau robot otonom.

Dua jenis utama **machine learning** adalah:

1. **Supervised Learning**

- Dalam supervised learning, model dilatih menggunakan data yang sudah memiliki **label atau jawaban** yang benar. Model belajar untuk memetakan input ke output yang benar berdasarkan data yang sudah diberi label.
- **Contoh:** Prediksi harga rumah berdasarkan data yang sudah diketahui (seperti ukuran dan lokasi rumah).

2. **Unsupervised Learning**

- Dalam unsupervised learning, model dilatih dengan data yang **belum dilabeli**. Tujuannya adalah untuk menemukan pola atau struktur dalam data tanpa adanya jawaban yang sudah ditentukan.
- **Contoh:** Pengelompokan pelanggan berdasarkan perilaku belanja mereka tanpa informasi lebih lanjut tentang setiap pelanggan.

IoT (Internet of Things) adalah konsep di mana objek fisik, seperti perangkat rumah tangga, kendaraan, atau sensor, terhubung dan dapat saling berkomunikasi melalui internet. Tujuannya adalah untuk mengumpulkan, berbagi, dan menganalisis data untuk meningkatkan efisiensi, kenyamanan, dan pengambilan keputusan.

Contoh:

- Smart home devices seperti lampu otomatis atau thermostat pintar yang dapat dikendalikan melalui smartphone.
- Sensor pada mobil otonom untuk mendeteksi lingkungan sekitar dan membuat keputusan.

IoT memungkinkan perangkat untuk "berbicara" satu sama lain dan bekerja secara otomatis berdasarkan data yang diperoleh.

IoT Use Case adalah contoh penerapan IoT untuk menyelesaikan masalah atau meningkatkan efisiensi dalam berbagai sektor. Berikut beberapa **contoh IoT use case**:

1. **Smart Home:**

- **Contoh:** Sistem rumah pintar yang menghubungkan perangkat seperti lampu, termostat, dan kamera keamanan. Pengguna dapat mengontrol dan memonitor perangkat melalui aplikasi smartphone.
- **Manfaat:** Kenyamanan, penghematan energi, dan keamanan.

2. **Smart Agriculture:**

- **Contoh:** Sensor yang memonitor kelembaban tanah, suhu, dan kondisi tanaman. Data ini digunakan untuk mengatur sistem irigasi otomatis.
- **Manfaat:** Meningkatkan hasil pertanian, mengurangi penggunaan air, dan efisiensi operasional.

3. **Smart Cities:**

- **Contoh:** Penggunaan sensor untuk memantau kualitas udara, kemacetan lalu lintas, dan pengelolaan sampah. Data yang dikumpulkan digunakan untuk perencanaan kota yang lebih efisien.
- **Manfaat:** Meningkatkan kualitas hidup, mengurangi polusi, dan manajemen lalu lintas yang lebih baik.

4. Healthcare (Telemedicine):

- **Contoh:** Perangkat wearable yang memantau tanda vital seperti detak jantung atau kadar gula darah dan mengirimkan data ke dokter secara real-time.
- **Manfaat:** Pemantauan kesehatan jarak jauh, deteksi dini masalah kesehatan, dan pengurangan biaya perawatan.

5. Industrial IoT (IIoT):

- **Contoh:** Sensor di mesin industri untuk memantau kondisi dan kinerja alat, serta mendeteksi potensi kerusakan sebelum terjadi (predictive maintenance).
- **Manfaat:** Mengurangi downtime, mengoptimalkan efisiensi produksi, dan memperpanjang umur mesin.

IoT membantu berbagai sektor untuk meningkatkan efisiensi, produktivitas, dan kualitas layanan melalui otomatisasi dan pemantauan berbasis data.

MQTT (Message Queuing Telemetry Transport) adalah protokol komunikasi ringan yang digunakan untuk mentransfer pesan antar perangkat dalam aplikasi IoT. MQTT dirancang untuk komunikasi yang efisien, dengan bandwidth rendah, dan dapat bekerja dengan baik pada jaringan yang tidak stabil.

Fitur utama MQTT:

1. **Publisher-Subscriber Model:** Perangkat (publisher) mengirimkan pesan ke topik tertentu, dan perangkat lain (subscriber) menerima pesan yang diterbitkan pada topik tersebut.
2. **QoS (Quality of Service):** Menyediakan tiga tingkat kualitas layanan untuk memastikan pengiriman pesan yang andal, yaitu QoS 0 (terkirim sekali), QoS 1 (terkirim setidaknya sekali), dan QoS 2 (terkirim hanya sekali).
3. **Ringan dan Efisien:** Protokol ini menggunakan header pesan kecil, cocok untuk perangkat dengan sumber daya terbatas dan koneksi jaringan yang tidak stabil.

Contoh penggunaan MQTT:

- Sensor suhu yang mengirimkan data secara periodik ke server untuk diproses dan diteruskan ke aplikasi pengguna.

MQTT sangat populer dalam aplikasi IoT karena kemudahan implementasi dan keandalannya dalam kondisi jaringan terbatas.

Python adalah bahasa pemrograman tingkat tinggi yang mudah dipahami, fleksibel, dan sangat populer di kalangan pengembang perangkat lunak. Python sering digunakan dalam berbagai bidang, seperti pengembangan web, analisis data, kecerdasan buatan (AI), machine learning, dan otomatisasi.

Variabel adalah tempat penyimpanan data yang memiliki nama dan dapat menyimpan nilai. Variabel digunakan untuk menyimpan informasi yang dapat digunakan dan diubah selama program berjalan.

Ciri-ciri Variabel:

- **Nama:** Harus unik dan mengikuti aturan tertentu (misalnya, tidak diawali angka).
- **Tipe Data:** Dapat menyimpan berbagai jenis data, seperti **angka (int, float)**, **teks (string)**, **boolean**, atau **objek**.
- **Nilai:** Variabel menyimpan nilai yang dapat diubah atau dimodifikasi sepanjang eksekusi program.

Contoh:

```
age = 25          # Variabel 'age' menyimpan nilai integer
```

```
name = "John"    # Variabel 'name' menyimpan string
```

```
is_student = True # Variabel 'is_student' menyimpan boolean
```

Variabel memudahkan pengelolaan data dan manipulasi nilai dalam program.

Tipe data merujuk pada jenis nilai yang dapat disimpan dalam variabel. Berikut adalah beberapa tipe data dasar dalam pemrograman:

1. **Integer (int)** – Menyimpan angka bulat tanpa desimal.
 - Contoh: `x = 5`

2. **Float** – Menyimpan angka desimal.
 - Contoh: `y = 3.14`
3. **String (str)** – Menyimpan teks atau karakter.
 - Contoh: `name = "John"`
4. **Boolean (bool)** – Menyimpan nilai **True** atau **False**.
 - Contoh: `is_active = True`
5. **List** – Menyimpan koleksi data yang dapat diubah (mutable) dalam urutan.
 - Contoh: `fruits = ["apple", "banana", "cherry"]`
6. **Tuple** – Menyimpan koleksi data yang tidak dapat diubah (immutable) dalam urutan.
 - Contoh: `coordinates = (10, 20)`
7. **Dictionary (dict)** – Menyimpan pasangan kunci-nilai (key-value).
 - Contoh: `person = {"name": "John", "age": 30}`

Tipe data digunakan untuk menentukan jenis informasi yang akan disimpan dalam variabel, yang membantu dalam pengolahan data.

Casting tipe data adalah proses mengubah nilai dari satu tipe data ke tipe data lain. Ini dilakukan dengan menggunakan fungsi built-in di bahasa pemrograman. Casting bisa **implisit** (otomatis) atau **eksplisit** (manual).

Jenis Casting:

1. Implicit Casting (Type Coercion)

- Proses otomatis oleh Python ketika mengubah tipe data yang lebih kecil ke tipe data yang lebih besar.
- Contoh: Mengubah **int** menjadi **float**.

```
x = 5    # int
y = 2.5  # float
z = x + y # Python secara otomatis mengubah x menjadi float
print(z) # Output: 7.5
```

2. Explicit Casting (Type Conversion)

- Proses manual dengan menggunakan fungsi seperti `int()`, `float()`, `str()`, dll., untuk mengubah satu tipe data ke tipe data lain.

- Contoh: Mengubah **float** menjadi **int** atau **string**.

```
x = 10.5  
  
y = int(x) # mengubah float menjadi int  
  
print(y) # Output: 10
```

Contoh Explicit Casting:

- **float()**: Mengubah ke tipe data float.

```
x = "5.7"  
  
y = float(x)  
  
print(y) # Output: 5.7
```

- **int()**: Mengubah ke tipe data integer.

```
x = "10"  
  
y = int(x)  
  
print(y) # Output: 10
```

- **str()**: Mengubah ke tipe data string.

```
x = 100  
  
y = str(x)  
  
print(y) # Output: "100"
```

Casting membantu mengelola tipe data yang berbeda dalam operasi atau fungsi tertentu.

Input adalah cara untuk menerima data dari pengguna. Di Python, fungsi `input()` digunakan untuk meminta input dari pengguna, yang selalu dikembalikan dalam bentuk **string**. Untuk tipe data lain, seperti **integer** atau **float**, perlu dilakukan **casting**.

Contoh:

```
name = input("Masukkan nama: ")  
  
age = int(input("Masukkan usia: "))
```

Fungsi `input` memungkinkan program berinteraksi dengan pengguna untuk menerima informasi yang diperlukan.

Operasi aritmatika adalah operasi dasar dalam matematika untuk memanipulasi angka. Dalam Python, operasi ini mencakup:

1. **Penjumlahan (+):** Menambahkan dua angka.
Contoh: $5 + 3 \rightarrow 8$
2. **Pengurangan (-):** Mengurangi angka pertama dengan angka kedua.
Contoh: $5 - 3 \rightarrow 2$
3. **Perkalian (*):** Mengalikan dua angka.
Contoh: $5 * 3 \rightarrow 15$
4. **Pembagian (/):** Membagi angka pertama dengan angka kedua (hasilnya float).
Contoh: $5 / 2 \rightarrow 2.5$
5. **Pembagian Bulat (//):** Membagi angka dan menghasilkan hasil bulat (integer).
Contoh: $5 // 2 \rightarrow 2$
6. **Modulus (%):** Menghasilkan sisa pembagian.
Contoh: $5 \% 2 \rightarrow 1$
7. **Pemangkatan (**):** Menghitung angka pertama dipangkatkan dengan angka kedua.
Contoh: $5 ** 3 \rightarrow 125$

Operasi aritmatika ini memungkinkan kita untuk melakukan perhitungan dasar dalam program.

Operasi komparasi digunakan untuk membandingkan dua nilai dan menghasilkan **True** atau **False** berdasarkan hasil perbandingan. Berikut adalah operasi komparasi dasar:

1. **Sama dengan (==):** Membandingkan apakah dua nilai sama.
Contoh: $5 == 5 \rightarrow \text{True}$
2. **Tidak sama dengan (!=):** Membandingkan apakah dua nilai tidak sama.
Contoh: $5 != 3 \rightarrow \text{True}$
3. **Lebih besar dari (>):** Membandingkan apakah nilai pertama lebih besar dari nilai kedua.
Contoh: $5 > 3 \rightarrow \text{True}$
4. **Lebih kecil dari (<):** Membandingkan apakah nilai pertama lebih kecil dari nilai kedua.
Contoh: $3 < 5 \rightarrow \text{True}$
5. **Lebih besar atau sama dengan (>=):** Membandingkan apakah nilai pertama lebih besar atau sama dengan nilai kedua.
Contoh: $5 >= 5 \rightarrow \text{True}$

6. **Lebih kecil atau sama dengan (<=)**: Membandingkan apakah nilai pertama lebih kecil atau sama dengan nilai kedua.

Contoh: $3 \leq 5 \rightarrow \text{True}$

Operasi komparasi ini sering digunakan dalam **percabangan** dan **perulangan** untuk mengendalikan alur program.

Operasi logika digunakan untuk menggabungkan atau membandingkan nilai boolean (**True** atau **False**). Berikut adalah operasi logika dasar:

1. **AND (and)**: Mengembalikan **True** jika kedua nilai yang dibandingkan adalah **True**.
Contoh: $\text{True and False} \rightarrow \text{False}$
2. **OR (or)**: Mengembalikan **True** jika salah satu nilai yang dibandingkan adalah **True**.
Contoh: $\text{True or False} \rightarrow \text{True}$
3. **NOT (not)**: Membalikkan nilai boolean.
Contoh: $\text{not True} \rightarrow \text{False}$

Operasi logika digunakan untuk membuat keputusan dalam program berdasarkan kondisi boolean.

Operator assignment digunakan untuk memberikan atau memperbarui nilai variabel. Berikut adalah operator assignment dasar:

1. **=**: Menetapkan nilai ke variabel.
Contoh: $x = 5$
2. **+=**: Menambahkan nilai ke variabel dan menetapkan hasilnya ke variabel.
Contoh: $x += 3$ (setara dengan $x = x + 3$)
3. **-=**: Mengurangi nilai dari variabel dan menetapkan hasilnya ke variabel.
Contoh: $x -= 2$ (setara dengan $x = x - 2$)
4. ***=**: Mengalikan variabel dengan nilai dan menetapkan hasilnya ke variabel.
Contoh: $x *= 4$ (setara dengan $x = x * 4$)
5. **/=**: Membagi variabel dengan nilai dan menetapkan hasilnya ke variabel.
Contoh: $x /= 2$ (setara dengan $x = x / 2$)
6. **%=**: Mengambil sisa pembagian antara variabel dan nilai, lalu menetapkan hasilnya ke variabel.
Contoh: $x \% = 3$ (setara dengan $x = x \% 3$)

Operator assignment ini menyederhanakan penulisan kode dengan mempersingkat operasi yang melibatkan variabel.

Operasi manipulasi string digunakan untuk mengubah atau memanipulasi nilai string. Berikut adalah beberapa operasi dasar pada string:

1. **Concatenation (+)**: Menggabungkan dua string.
Contoh: "Hello" + " World" → "Hello World"
2. **Repetition (*)**: Mengulang string beberapa kali.
Contoh: "Hi" * 3 → "HiHiHi"
3. **Slicing ([])**: Mengambil bagian dari string.
Contoh: "Hello"[1:4] → "ello"
4. **Length (len())**: Menghitung panjang string.
Contoh: len("Hello") → 5
5. **Uppercase (upper())**: Mengubah semua huruf menjadi kapital.
Contoh: "hello".upper() → "HELLO"
6. **Lowercase (lower())**: Mengubah semua huruf menjadi kecil.
Contoh: "HELLO".lower() → "hello"
7. **Replace (replace())**: Mengganti bagian string dengan yang baru.
Contoh: "Hello".replace("e", "a") → "Hallo"
8. **Find (find())**: Mencari posisi substring dalam string.
Contoh: "Hello".find("e") → 1

Operasi ini memungkinkan manipulasi dan pengolahan data teks dengan mudah dalam program.

Format string digunakan untuk menyisipkan nilai variabel ke dalam string. Ada beberapa cara untuk melakukannya:

1. **F-string (Python 3.6+)**: Menggunakan tanda kurung {} di dalam string yang diawali dengan huruf f.
Contoh:

```
name = "Alice"
greeting = f"Hello, {name}!"
```

2. **str.format():** Menggunakan placeholder {} di dalam string dan menggantinya dengan variabel di dalam format().

Contoh:

```
name = "Bob"
```

```
greeting = "Hello, {}".format(name)
```

3. **Old-style (%):** Menggunakan % untuk menyisipkan nilai dalam string.

Contoh:

```
name = "Charlie"
```

```
greeting = "Hello, %s!" % name
```

Format string memudahkan dalam membuat teks dinamis dengan nilai variabel yang dapat berubah-ubah.

if, else, dan elif digunakan untuk membuat percabangan dalam program berdasarkan kondisi.

1. **if:** Mengecek kondisi pertama, jika **True**, blok kode di dalamnya akan dijalankan.

Contoh:

```
if x > 10:
```

```
    print("x lebih besar dari 10")
```

2. **else:** Menjalankan blok kode jika kondisi pada if **False**.

Contoh:

```
if x > 10:
```

```
    print("x lebih besar dari 10")
```

```
else:
```

```
    print("x tidak lebih besar dari 10")
```

3. **elif:** Menyediakan kondisi lain untuk diperiksa jika kondisi if sebelumnya **False**.

Contoh:

```
if x > 10:
```

```
    print("x lebih besar dari 10")
```



```
elif x == 10:
    print("x sama dengan 10")
else:
    print("x lebih kecil dari 10")
```

Ini memungkinkan program untuk membuat keputusan berdasarkan nilai atau kondisi yang diberikan.

Looping digunakan untuk menjalankan blok kode berulang kali berdasarkan kondisi tertentu. Berikut adalah penjelasan singkat tentang looping dan perintah terkait:

1. **for loop:** Digunakan untuk mengulang kode untuk setiap item dalam urutan (list, string, dll).

Contoh:

```
for i in range(3):
    print(i)
```

2. **while loop:** Digunakan untuk mengulang kode selama kondisi tertentu **True**.

Contoh:

```
i = 0
while i < 3:
    print(i)
    i += 1
```

3. **continue:** Melompati sisa kode dalam iterasi saat ini dan melanjutkan ke iterasi berikutnya dalam loop.

Contoh:

```
for i in range(5):
    if i == 2:
        continue # Melewati iterasi ketika i == 2
    print(i)
```

4. **break:** Menghentikan loop sepenuhnya, keluar dari loop meskipun kondisi belum selesai.

Contoh:

```
for i in range(5):  
    if i == 3:  
        break # Menghentikan loop ketika i == 3  
    print(i)
```

5. **pass:** Digunakan sebagai placeholder, tidak melakukan apa-apa, dan menghindari error pada blok kode yang kosong.

Contoh:

```
for i in range(5):  
    if i == 3:  
        pass # Tidak melakukan apa-apa ketika i == 3  
    print(i)
```

Perintah ini membantu mengontrol alur perulangan dalam program.

Fungsi adalah blok kode yang dirancang untuk melakukan tugas tertentu, yang dapat dipanggil berulang kali dalam program. Fungsi membantu membuat kode lebih terstruktur dan mudah dipelihara.

Membuat Fungsi:

Fungsi didefinisikan menggunakan kata kunci `def`, diikuti nama fungsi dan parameter (jika ada).

```
def nama_fungsi(parameter):  
    # kode yang dijalankan  
    return hasil
```

Contoh Fungsi:

```
def tambah(a, b):  
    return a + b
```

```
result = tambah(5, 3)
print(result) # Output: 8
```

return:

return digunakan untuk mengembalikan nilai dari fungsi ke tempat fungsi dipanggil. Tanpa return, fungsi akan mengembalikan nilai None secara default.

```
def kali(a, b):
```

```
    return a * b
```

```
result = kali(4, 2)
```

```
print(result) # Output: 8
```

Fungsi memungkinkan kita untuk mengorganisir dan mengulang kode secara efisien.

Class adalah template atau blueprint untuk membuat objek. Class mendefinisikan atribut (data) dan metode (fungsi) yang dimiliki oleh objek. Dengan class, kita bisa membuat banyak objek dengan sifat dan perilaku yang sama.

Contoh Class:

```
class Mobil:
```

```
    def __init__(self, merek, warna):
```

```
        self.merek = merek
```

```
        self.warna = warna
```

```
    def info(self):
```

```
        return f"Merek: {self.merek}, Warna: {self.warna}"
```

```
# Membuat objek dari class Mobil
```

```
mobil1 = Mobil("Toyota", "Merah")
```

```
print(mobil1.info()) # Output: Merek: Toyota, Warna: Merah
```

Penjelasan:

- **__init__()**: Konstruktor yang dipanggil saat objek dibuat, digunakan untuk menginisialisasi atribut.
- **self**: Merujuk ke objek itu sendiri, digunakan untuk mengakses atribut dan metode dalam class.

Class memungkinkan kita untuk membuat dan mengelola objek secara lebih terstruktur dan modular.

args** dan *kwargs** adalah cara untuk mengirimkan jumlah argumen yang tidak terbatas ke dalam fungsi.

1. ***args**:

- Digunakan untuk mengirimkan jumlah argumen yang tidak terbatas sebagai **tuple**.
- Biasanya digunakan untuk parameter yang jumlahnya tidak diketahui.

Contoh:

```
def tambah(*args):
    return sum(args)
```

```
print(tambah(1, 2, 3)) # Output: 6
```

2. ****kwargs**:

- Digunakan untuk mengirimkan jumlah argumen yang tidak terbatas sebagai **dictionary** (key-value pairs).
- Biasanya digunakan untuk parameter yang memiliki nama (keyword arguments).

Contoh:

```
def info(**kwargs):
    for key, value in kwargs.items():
        print(f"{key}: {value}")
```

```
info(nama="Alice", umur=25)
```

```
# Output:
```

```
# nama: Alice
```

```
# umur: 25
```

Kesimpulan:

- *args untuk argumen posisi (tanpa nama).
- **kwargs untuk argumen kata kunci (dengan nama).

Global scope dan **local scope** merujuk pada area di dalam program di mana variabel dapat diakses.

1. Global Scope:

- Variabel yang didefinisikan di luar fungsi, sehingga dapat diakses oleh semua fungsi dalam program.
- Variabel global berlaku untuk seluruh program.

Contoh:

```
x = 10 # Variabel global
```

```
def tampilkan():
```

```
    print(x) # Mengakses variabel global
```

```
tampilkan() # Output: 10
```

2. Local Scope:

- Variabel yang didefinisikan di dalam fungsi hanya bisa diakses di dalam fungsi tersebut.
- Variabel lokal bersifat sementara dan hanya berlaku dalam fungsi yang mendefinisikannya.

Contoh:

```
def contoh():
```

```
    y = 5 # Variabel lokal
```

```
    print(y) # Mengakses variabel lokal
```

```
contoh() # Output: 5
```

```
# print(y) # Error, karena y adalah variabel lokal
```

Kesimpulan:

- **Global:** Variabel bisa diakses dari mana saja dalam program.
- **Local:** Variabel hanya dapat diakses dalam fungsi tempat variabel tersebut didefinisikan.

IaaS (Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service), dan **On-premise** adalah model penyediaan layanan teknologi informasi yang berbeda, baik di cloud maupun di lokal.

1. IaaS (Infrastructure as a Service):

- Menyediakan infrastruktur IT dasar (seperti server, penyimpanan, jaringan) melalui cloud.
- Pengguna dapat mengelola dan mengonfigurasi sistem operasi, aplikasi, dan data, tapi tidak perlu mengelola hardware.
- Contoh: Amazon Web Services (AWS), Microsoft Azure.

2. PaaS (Platform as a Service):

- Menyediakan platform dan alat pengembangan aplikasi tanpa harus mengelola infrastruktur atau sistem operasi.
- Pengguna hanya fokus pada pengembangan dan pengelolaan aplikasi, sementara penyedia layanan mengurus platform.
- Contoh: Google App Engine, Heroku.

3. SaaS (Software as a Service):

- Menyediakan aplikasi perangkat lunak yang dapat diakses melalui internet tanpa perlu instalasi atau pengelolaan perangkat keras.
- Pengguna mengakses aplikasi yang sudah jadi, tanpa perlu memikirkan infrastruktur atau pembaruan.
- Contoh: Gmail, Dropbox, Microsoft 365.

4. On-premise:

- Infrastruktur atau aplikasi yang diinstal dan dijalankan di lokasi fisik (lokasi perusahaan).
- Perusahaan bertanggung jawab untuk pemeliharaan, pembaruan, dan pengelolaan infrastruktur.

- Contoh: Server lokal perusahaan yang menjalankan software dan penyimpanan data.

Perbedaan utama:

- **IaaS:** Pengelolaan infrastruktur dasar.
- **PaaS:** Pengelolaan platform pengembangan aplikasi.
- **SaaS:** Aplikasi siap pakai yang dapat diakses secara online.
- **On-premise:** Infrastruktur dan aplikasi dikelola sepenuhnya di lokasi perusahaan.

RFID, NFC, WiFi, dan Bluetooth adalah teknologi komunikasi nirkabel yang digunakan untuk berbagai keperluan seperti transfer data, pengenalan objek, dan konektivitas antar perangkat.

1. RFID (Radio Frequency Identification):

- Teknologi yang menggunakan gelombang radio untuk mengidentifikasi objek, hewan, atau orang secara otomatis.
- Dikenal dengan penggunaan tag yang berisi informasi yang dapat dipindai dengan pembaca RFID.
- Contoh: Sistem pembayaran tol otomatis, kartu identitas, manajemen inventaris.

2. NFC (Near Field Communication):

- Subset dari RFID dengan jarak transmisi yang sangat pendek (biasanya kurang dari 10 cm).
- Digunakan untuk komunikasi antar perangkat yang dekat satu sama lain, seperti pembayaran mobile atau pengiriman data.
- Contoh: Pembayaran menggunakan smartphone (Google Pay, Apple Pay), akses kartu kunci hotel.

3. WiFi:

- Teknologi komunikasi nirkabel yang memungkinkan perangkat untuk terhubung ke internet atau jaringan lokal (LAN) menggunakan gelombang radio.
- Biasanya digunakan di area yang lebih luas seperti rumah, kantor, atau publik.
- Contoh: Koneksi internet di rumah menggunakan router WiFi.

4. Bluetooth:

- Teknologi komunikasi nirkabel jarak pendek (biasanya hingga 100 meter) untuk menghubungkan perangkat elektronik satu sama lain.
- Digunakan untuk perangkat seperti earphone, keyboard, speaker, atau perangkat IoT lainnya.
- Contoh: Menghubungkan headset Bluetooth ke ponsel atau speaker Bluetooth ke laptop.

Perbedaan utama:

- **RFID & NFC:** Digunakan untuk identifikasi dan komunikasi jarak pendek (NFC lebih pendek dari RFID).
- **WiFi:** Digunakan untuk koneksi internet atau jaringan di area yang lebih luas.
- **Bluetooth:** Digunakan untuk koneksi antar perangkat elektronik dalam jarak dekat.

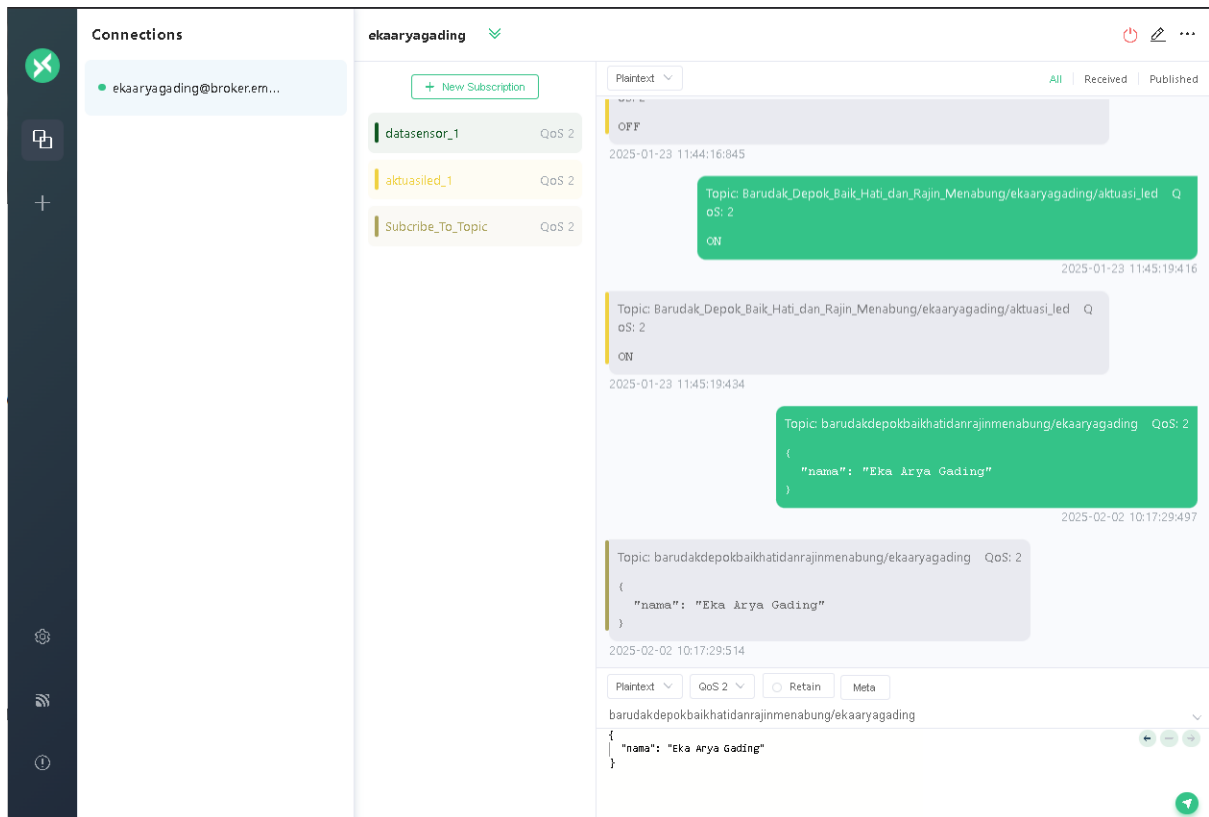
Challenge 1



Dengan menggunakan MQTT Client, cobalah publish dan subscribe mqtt data dengan kondisi sebagai berikut:

1. Publish sebuah pesan berisikan nama anda ke topic /NAMAGRUP/nama
2. Subscribe ke topic /NAMAGRUP/nama

Lakukanlah dengan tim kalian dan coba diganti rolenya antara publisher dan juga subscriber:

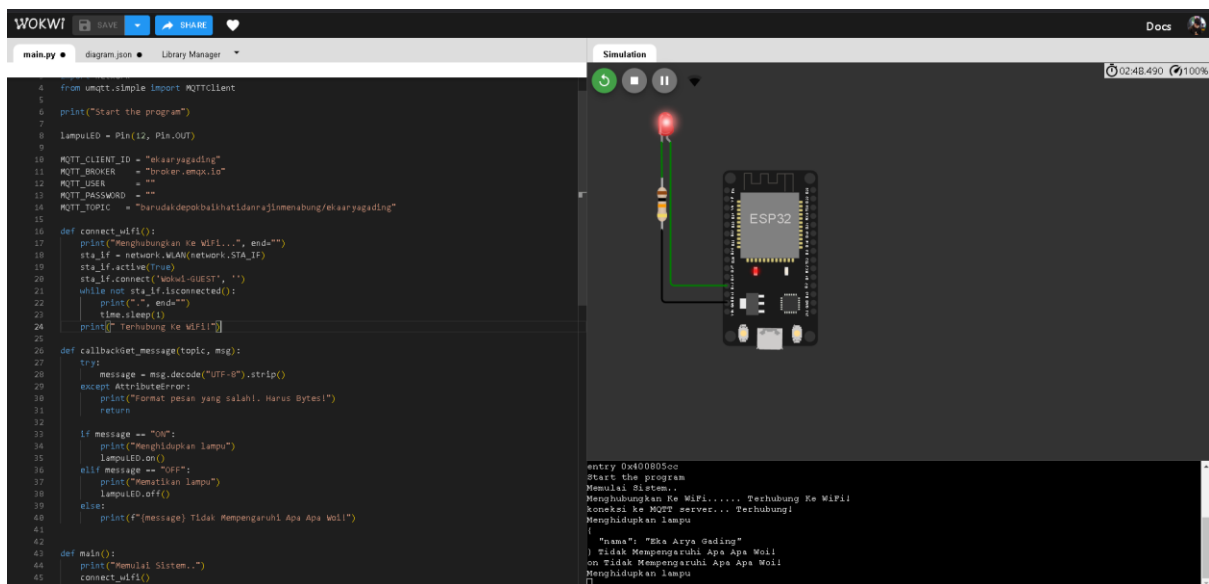
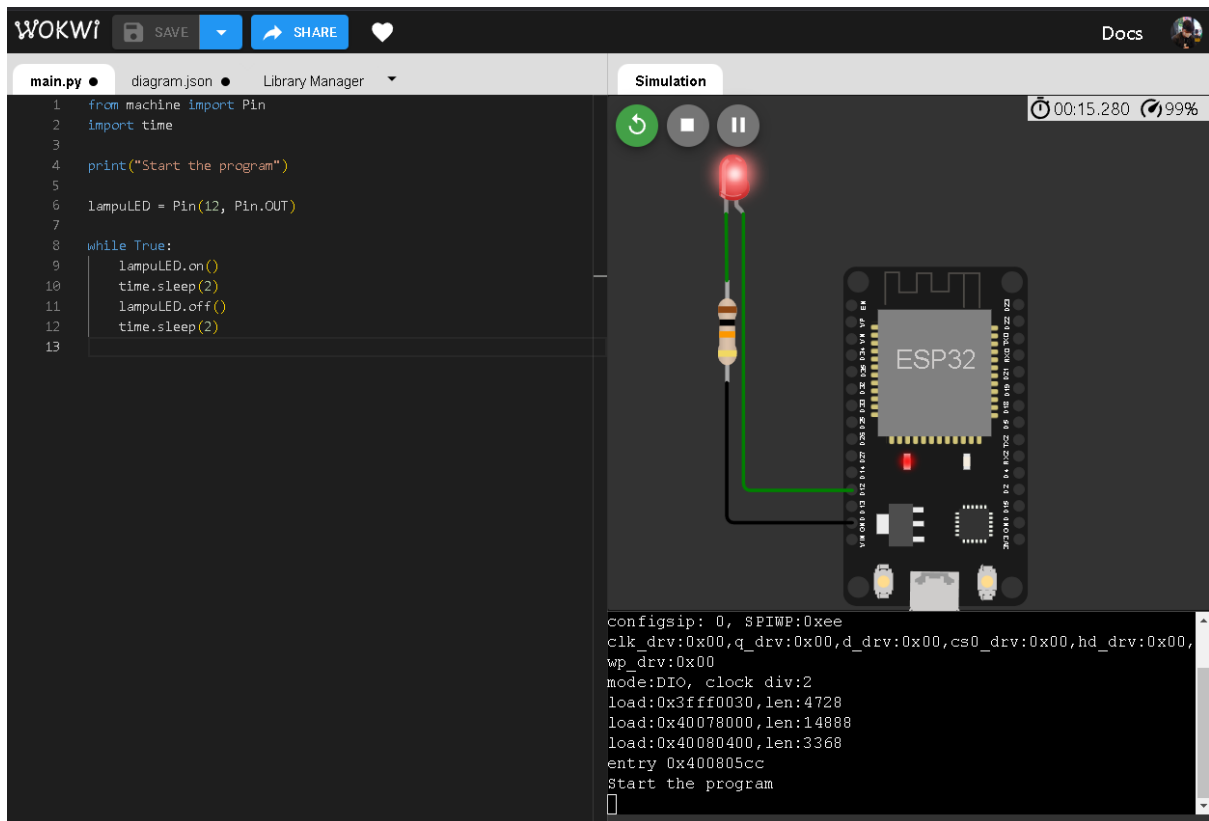


Challenge 2

Samsung
Innovation Campus
Batch 6 – 2024/2025

Dengan menggunakan simulator Wokwi (<https://wokwi.com/projects/new/esp32>)

1. Publish sebuah pesan berisikan “Turn Off” / “Turn On” ke topic /NAMAGRUP/actuate_lamp menggunakan MQTT Client
2. Subscribe ke topic /NAMAGRUP/nama di WokWi ESP32 untuk meng-aktuasi LED



```

from machine import Pin
import time
import network
from umqtt.simple import MQTTClient

print("Start the program")

lampuLED = Pin(12, Pin.OUT)

MQTT_CLIENT_ID = "ekaaryagading"

```

```

MQTT_BROKER    = "broker.emqx.io"
MQTT_USER      = ""
MQTT_PASSWORD  = ""
MQTT_TOPIC     = "barudakdepokbaikhatidanrajinmenabung/ekaaryagading"

def connect_wifi():
    print("Menghubungkan Ke WiFi...", end="")
    sta_if = network.WLAN(network.STA_IF)
    sta_if.active(True)
    sta_if.connect('Wokwi-GUEST', '')
    while not sta_if.isconnected():
        print(".", end="")
        time.sleep(1)
    print(" Terhubung Ke WiFi!")

def callbackGet_message(topic, msg):
    try:
        message = msg.decode("UTF-8").strip()
    except AttributeError:
        print("Format pesan yang salah!. Harus Bytes!")
        return

    if message == "ON":
        print("Menghidupkan lampu")
        lampuLED.on()
    elif message == "OFF":
        print("Mematikan lampu")
        lampuLED.off()
    else:
        print(f"{message} Tidak Mempengaruhi Apa Apa Woi!")

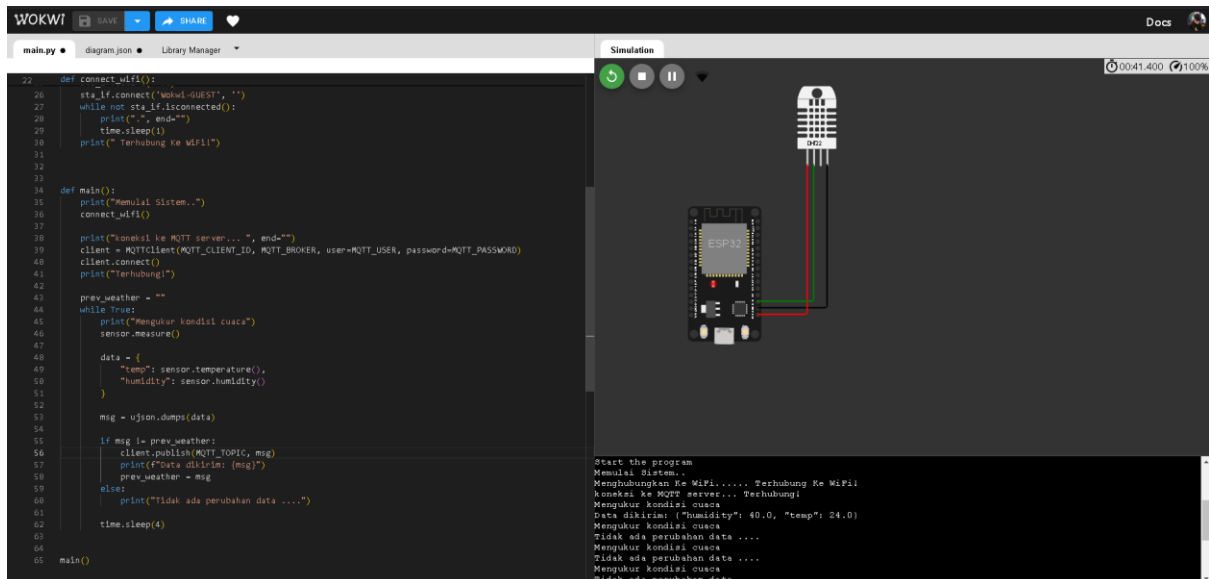
def main():
    print("Memulai Sistem..")
    connect_wifi()

    print("koneksi ke MQTT server... ", end="")
    client = MQTTClient(MQTT_CLIENT_ID, MQTT_BROKER, user=MQTT_USER,
password=MQTT_PASSWORD)
    client.set_callback(callbackGet_message)
    client.connect()
    client.subscribe(MQTT_TOPIC)
    print("Terhubung!")

    while True:
        client.check_msg()
        time.sleep(4)

```

```
main()
```



```
from machine import Pin
import time
import network
import dht
import ujson
from umqtt.simple import MQTTClient

print("Start the program")

lampuLED = Pin(12, Pin.OUT)

MQTT_CLIENT_ID = "ekaaryagading"
MQTT_BROKER     = "broker.emqx.io"
MQTT_USER       = ""
MQTT_PASSWORD   = ""
MQTT_TOPIC      = "barudakdepokbaikhatidanrajinmenabung/ekaaryagading"

sensor = dht.DHT22(Pin(15))

def connect_wifi():
    print("Menghubungkan Ke WiFi...", end="")
    sta_if = network.WLAN(network.STA_IF)
    sta_if.active(True)
    sta_if.connect('Wokwi-GUEST', '')
    while not sta_if.isconnected():
        print(".", end="")
```

```

        time.sleep(1)
    print(" Terhubung Ke WiFi!")

def main():
    print("Memulai Sistem..")
    connect_wifi()

    print("koneksi ke MQTT server... ", end="")
    client = MQTTClient(MQTT_CLIENT_ID, MQTT_BROKER, user=MQTT_USER,
password=MQTT_PASSWORD)
    client.connect()
    print("Terhubung!")

    prev_weather = ""
    while True:
        print("Mengukur kondisi cuaca")
        sensor.measure()

        data = {
            "temp": sensor.temperature(),
            "humidity": sensor.humidity()
        }

        msg = ujson.dumps(data)

        if msg != prev_weather:
            client.publish(MQTT_TOPIC, msg)
            print(f>Data dikirim: {msg}")
            prev_weather = msg
        else:
            print("Tidak ada perubahan data ....")

        time.sleep(4)

main()

```

ekaaryagading

+ New Subscription

datasensor_1

QoS 2

aktuasiled_1

QoS 2

Subscribe_To_Topic

QoS 2

Plaintext

Topic: barudakdepokbaikhatidanrajinmenabung/ekaaryagading

QoS: 2

ON

2025-02-02 10:35:03:369

Topic: barudakdepokbaikhatidanrajinmenabung/ekaaryagading

QoS: 2

OFF

2025-02-02 10:36:49:822

Topic: barudakdepokbaikhatidanrajinmenabung/ekaaryagading

QoS: 2

ON

2025-02-02 10:37:00:619

Topic: barudakdepokbaikhatidanrajinmenabung/ekaaryagading

QoS: 0

{"humidity": 40.0, "temp": 24.0}

2025-02-02 11:15:06:734

JSON

QoS 2

Retain

Meta

barudakdepokbaikhatidanrajinmenabung/ekaaryagading

ON

Week 2

Challenge!

Samsung
Innovation Campus
Batch 6 – 2024/2025

Buat sebuah program yang jika kita berikan sebuah input array seperti [1,2,3,4,5] maka akan menghasilkan output [1,4,9,16,25]!

```
# Buat sebuah program yang jika kita berikan sebuah
# input array seperti
# [1,2,3,4,5] maka akan menghasilkan output [1,4,9,16,25]!

array = [1,2,3,4,5]
```

```
for i in range(len(array)):
    array[i] = array[i] * array[i]

print(array) # [1, 4, 9, 16, 25]
```

Challenge!

Samsung
Innovation Campus

Batch 6 – 2024/2025

Buat sebuah program dimana jika input yang diberikan adalah
[1,2,3,4,5] maka outputnya adalah
["ganjil","genap","ganjil","genap","ganjil"]

```
array = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

for i in range(len(array)):
    if array[i] % 2 == 0:
        array[i] = "Genap"
    else:
        array[i] = "Ganjil"

print(array) # ['Genap', 'Ganjil', 'Genap', 'Ganjil', 'Genap',
               'Ganjil', 'Genap', 'Ganjil', 'Genap', 'Ganjil']
```

Challenge!

Samsung
Innovation Campus

Batch 6 – 2024/2025

Buat sebuah program dimana jika input yang diberikan adalah
[[1,2,3],[2,3,4],[3,4,5]] maka outputnya adalah [6,9,12]!

```
array = [[1,2,3],[2,3,4],[3,4,5]]  
  
result = map(sum, array)  
print(list(result))
```

Additional Challenge!

Samsung
Innovation Campus

Batch 6 – 2024/2025

Buat sebuah program agar outputnya seperti berikut:

Result:
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
19
Buzz

```
for i in range (1,50):  
    if i % 3 == 0 and i % 5 == 0:  
        print("fizzbuzz")
```



```
elif i % 3 == 0:
    print("fizz")
elif i % 5 == 0:
    print("buzz")
else:
    print(i)
```

Challenge 1

Samsung
Innovation Campus

Batch 6 – 2024/2025

Buat sebuah program yang jika kita berikan sebuah
Input = [100,200,300,400,500]
Compared_input = [500,200,400]
Maka akan menghasilkan output [100,0,300,0,0]!
Usahakan tidak menggunakan list baru!

Example2!

Samsung
Innovation Campus

Batch 6 – 2024/2025

Buat sebuah program dimana jika
input yang diberikan adalah
Input = [
 {'nama': 'Budi', 'nilai': 90},
 {'nama': 'Dwi', 'nilai': 85},
 {'nama': 'Tri', 'nilai': 75},
]
Output = {
 "nilai_tertinggi": "Budi",
 "nilai_terendah": "Tri"
}

```
input = [  
    {'nama' : 'budi', 'gaji' : 1000000},  
    {'nama' : 'andi', 'gaji' : 2000000},
```

```

        {'nama' : 'tono', 'gaji' : 3000000}
    ]

    gajiTertinggi = max(input, key=lambda x: x['gaji'])
    print(gajiTertinggi)

    gajiTerendah = min(input, key=lambda x: x['gaji'])
    print(gajiTerendah)

    totalGajiKaryawan = sum(x['gaji'] for x in input)
    print(f"Total gaji karyawan: {totalGajiKaryawan}")

```

Example3!

Input 1

Buat sebuah program
dimana jika input
yang diberikan
adalah

```

data_toko = {
    "Indoramet":{
        "Ayam": 30000,
        "Sayur": 15000,
        "Buah": 20000,
        "Ikan": 22000
    }
}

```

```

items_to_buy = {
    "Ayam":2,
    "Sayur":1,
    "Ikan":1
}

```

Output = 97000

Rincian
 Ayam : 2 * 30.000 = 60.000
 Sayur : 1 * 15.000 = 15.000
 Ikan : 1 * 22.000 = 22.000

```

dataStore = [
    {
        "indomaret": {
            "ayam" : 30000,
            "bebek" : 50000,
            "sapi" : 100000,
            "kambing" : 150000,
            "ikan" : 20000
        }
    }
]

items_to_buy = {

```

Samsung
Innovation Campus

Batch 6 – 2024/2025

```

    "ayam" : 2,
    "bebek" : 1,
    "sapi" : 1
}

total_price = 0
for item, quantity in items_to_buy.items():
    total_price += datastore[0]["indomaret"][item] * quantity

print(total_price)

```

Challenge 3!

Input 1

Buat sebuah program dimana jika input yang diberikan adalah

Hint: Cari harga termurah untuk setiap item antara 2 toko tersebut

```

input = {
    "Indoramet":{
        "Ayam": 30000,
        "Sayur": 15000,
        "Buah": 20000,
        "Ikan": 22000
    },
    "Alfaramet":{
        "Ayam": 25000,
        "Sayur": 12000,
        "Buah": 30000,
        "Ikan": 25000
    }
}

```

Samsung
Innovation Campus

Batch 6 – 2024/2025

Input 2

```

items_to_buy = {
    "Ayam":2,
    "Sayur":1,
    "Ikan":1
}

```

Output = 84000

Rincian

Ayam : 2 * 25.000 = 50.000

Sayur : 1 * 12.000 = 12.000

Ikan : 1 * 22.000 = 22.000

```

dataStore = [
    {
        "indomaret": {
            "ayam" : 30000,
            "bebek" : 30000,
            "sapi" : 100000,
            "kambing" : 200000,
            "ikan" : 20000
        },
        "alfamart": {
            "ayam" : 25000,
            "bebek" : 50000,
            "sapi" : 100000,

```

```

        "kambing" : 150000,
        "ikan" : 1000
    }
}

]

items_to_buy = {
    "ayam" : 2,
    "bebek" : 1,
    "sapi" : 1
}

cheapest_prices = {}
for item in items_to_buy.keys():
    indomaret_price = dataStore[0]["indomaret"][item]
    alfamart_price = dataStore[0]["alfamart"][item]
    cheapest_prices[item] = min(indomaret_price, alfamart_price)

print("Harga termurah untuk setiap item:")
for item, price in cheapest_prices.items():
    print(f"{item}: {price}")

total_price = sum(cheapest_prices.values())
print(f"Total harga: {total_price}")

```

Example Lambda

Buat program untuk mem-filter angka dari list berikut
 Input : [1, 2, 3, 4, 4, 5, 6, 7]
 Output:
 genap: [2, 4, 4, 6]
 ganjil: [1, 3, 5, 7]

```

input = [1,2,3,4,5,6,7,8,9,10]

genap = list(filter(lambda x: x % 2 == 0, input))

print(genap)

ganjil = list(filter(lambda x: x % 2 == 1, input))

print(ganjil)

```

Challenge Lambda

Samsung
Innovation Campus

Batch 6 – 2024/2025

Buat program untuk mem-filter dan menghitung buah-buahan dari list berikut

Output:

total buah manis : 2
total buah tidak manis: 3

```

Input = [
{
    "nama": "apple",
    "rasa": "manis"
},
{
    "nama": "jambu",
    "rasa": "manis"
},
{
    "nama": "lemon",
    "rasa": "asam"
},
{
    "nama": "strawberry",
    "rasa": "asam"
},
{
    "nama": "durian",
    "rasa": "pahit"
}
]

```

```

input = [
{
    "nama" : "ayam",
    "rasa" : "manis"
},
{
    "nama" : "bebek",
    "rasa" : "asam"
},
{
    "nama" : "sapi",
    "rasa" : "manis"
},
{

```

```
        "nama" : "kambing",
        "rasa" : "asin"
    },
    {
        "nama" : "ikan",
        "rasa" : "asam"
    }
]

rasaManis = list(filter(lambda x: x['rasa'] == "manis", input))

print(len(rasaManis))
```