LAPORAN PRAKTIKUM

Internet of Things

MQTT Server

Putri Geraldine Alexsandra Sihombing

11323009

Diploma III Teknologi Informasi

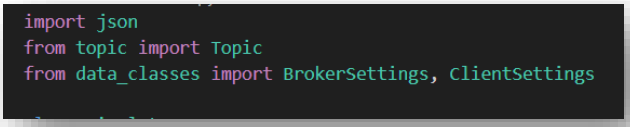
INSTITUT TEKNOLOGI DEL

FAKULTAS VOKASI

2024/2025

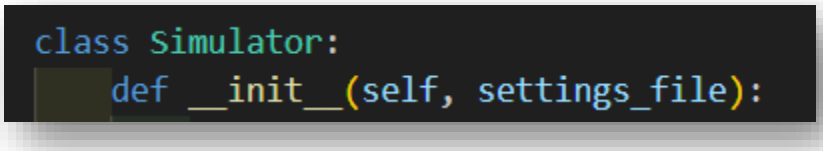
Simulator.py

1. Bagian import



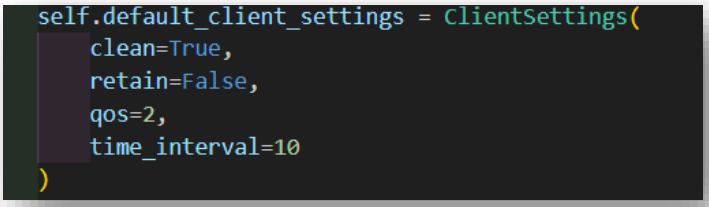
- json: Untuk membaca file JSON.
- Topic: Objek untuk mengatur topik yang nanti dibuat.
- BrokerSettings & ClientSettings: Kelas untuk menyimpan pengaturan seperti alamat server (*broker*) dan konfigurasi klien.

2. Membuat Kelas Simulator



- Simulator adalah kelas utama yang menjalankan simulasi.
- settings_file adalah file konfigurasi (format JSON) yang berisi pengaturan topik dan klien.

3. Pengaturan Klien Default



- default_client_settings: Pengaturan dasar kalau tidak ada pengaturan spesifik di file.
  - clean=True: Klien akan menghapus data lama saat tersambung.
  - retain=False: Pesan tidak disimpan oleh broker.
  - qos=2: Tingkat keandalan pesan (*quality of service*).
  - time_interval=10: Jeda waktu untuk kirim pesan (10 detik).

4. Memuat Topik

```
self.topics = self.load_topics(settings_file)
```

- self.topics: Daftar semua topik yang dibuat dari file JSON.

5. Fungsi Baca Pengaturan Klien

```
def read_client_settings(self, settings_dict: dict, default: ClientSettings):
```

- Fungsi ini membaca pengaturan dari file JSON. Kalau ada yang kosong, dia pakai pengaturan default.

6. Fungsi Memuat Topik dari File

```
def load_topics(self, settings_file):
    topics = []
```

- load_topics: Fungsi untuk membaca file JSON dan memuat semua topik.
- topics = []: Membuat daftar kosong untuk menyimpan topik.

7. Membuka File JSON

```
with open(settings_file) as json_file:
    config = json.load(json_file)
```

- with open: Membuka file JSON.
- json.load: Membaca isi file JSON jadi format Python.

8. Buat Pengaturan Broker

```
broker_settings = BrokerSettings(
    url=config.get('BROKER_URL', 'localhost'),
    port=config.get('BROKER_PORT', 1883),
    protocol=config.get('PROTOCOL_VERSION', 4) # mqtt.MQTTv311
)
```

- Membuat objek **broker_settings** yang menyimpan pengaturan server MQTT:
  - **url**: Alamat server, default-nya localhost.
  - **port**: Port server, default-nya 1883.
  - **protocol**: Versi protokol MQTT, default 4 (versi 3.1.1).

9. Membaca Pengaturan Topik

```
broker_client_settings = self.read_client_settings(config, default=self.default_client_settings)
```

- broker_client_settings: Membaca pengaturan klien untuk semua topik dari file JSON.

10. Mengelola Topik Berdasarkan Jenis

    Ada 3 jenis topik yang bisa dibuat:

    a. **Single**: Satu topik sederhana.

    Membuat satu topik dari PREFIX.

```
if topic['TYPE'] == 'single':
    # create single topic with format: /{PREFIX}
    topic_url = topic['PREFIX']
    topics.append(Topic(broker_settings, topic_url, topic_data, topic_payload_root, topic_client_settings))
```

    b. **Multiple**: Banyak topik dalam rentang angka (contoh: /sensor/1, /sensor/2).

    Membuat banyak topik berdasarkan angka mulai (RANGE_START) sampai akhir (RANGE_END).

```
elif topic['TYPE'] == 'multiple':
    # create multiple topics with format: /{PREFIX}/{id}
    for id in range(topic['RANGE_START'], topic['RANGE_END']+1):
        topic_url = topic['PREFIX'] + '/' + str(id)
        topics.append(Topic(broker_settings, topic_url, topic_data, topic_payload_root, topic_client_settings))
```

c. **List**: Banyak topik dengan nama tertentu (contoh: /sensor/temp, /sensor/humidity).
Membuat banyak topik dari daftar (`LIST`).

```
elif topic['TYPE'] == 'list':
    # create multiple topics with format: /{PREFIX}/{item}
    for item in topic['LIST']:
        topic_url = topic['PREFIX'] + '/' + str(item)
        topics.append(Topic(broker_settings, topic_url, topic_data, topic_payload_root, topic_client_settings))
```

11. Menjalankan Simulator

```
def run(self):
    for topic in self.topics:
        print(f'Starting: {topic.topic_url} ...')
        topic.start()
```

- run: Menjalankan semua topik.
  - ❖ topic.start(): Mulai operasi di setiap topik.

12. Menghentikan Simulator

```
def stop(self):
    for topic in self.topics:
        print(f'Stopping: {topic.topic_url} ...')
        topic.stop()
```

- stop: Menghentikan semua topik.
  - ❖ topic.stop(): Menghentikan operasi di setiap topik.