Laporan Praktikum

IOT

Estina Pangaribuan

11323038

D3 Teknologi Informasi

INSTITUT TEKNOLOGI DEL

FAKULTAS VOKASI

TAHUN AJARAN 2024/20

1. Pada kode berikut mengimport modul json untuk membaca file konfigurasi simulator dan mengimplementasikan kelas Topic untuk membuat dan mengelola topik MQTT dan BrokerSettings dan ClientSettings untuk menangani pengaturan broker dan klien MQTT.

```
import json
from topic import Topic
from data_classes import BrokerSettings, ClientSettings
```

Kode import ini memastikan simulator memiliki akses ke semua komponen yang diperlukan untuk berfungsi, termasuk membaca konfigurasi, membuat topik, dan mengelola komunikasi MQTT.

```
class Simulator:
    def __init__(self, settings_file):
        self.default_client_settings = ClientSettings(
            clean=True,
            retain=False,
            qos=2,
            time_interval=10
        )
        self.topics = self.load_topics(settings_file)
```

Pada Kode ini mempersiapkan simulator dengan:

- Pengaturan klien MQTT default dimana mengatur bagaimana klien MQTT beroperasi jika file konfigurasi tidak menentukan pengaturan tertentu.
- Daftar topik MQTT dimana memuat dan membuat objek topik berdasarkan pengaturan dalam file JSON.

Konstruktor ini memastikan bahwa simulator memiliki semua pengaturan dasar (klien dan topik) untuk dijalankan nantinya.

```
    def read_client_settings(self, settings_dict: dict, default: ClientSettings):
        return ClientSettings(
            clean=settings_dict.get('CLEAN_SESSION', default.clean),
            retain=settings_dict.get('RETAIN', default.retain),
            qos=settings_dict.get('QOS', default.qos),
            time_interval= settings_dict.get('TIME_INTERVAL', default.time_interval)
        )
```

Ada beberapa Fungsi ini berfungsi untuk membaca dan menggabungkan pengaturan klien dengan fleksibilitas:

1. Nilai yang diambil dari settings_dict akan diprioritaskan.
2. Jika suatu nilai tidak ada dalam settings_dict, maka nilai default (default) akan digunakan.

Fungsi dari kode berikut adalah untuk load_topics ini membaca file konfigurasi JSON dan membuat objek-objek Topic berdasarkan informasi yang terdapat dalam file tersebut. Ada tiga tipe topik yang didukung: single (satu topik), multiple (beberapa topik dengan ID), dan list (beberapa topik dengan item dari daftar). Fungsi ini mengonfigurasi setiap topik dengan pengaturan broker dan klien yang sesuai.

```python
    def run(self):
        for topic in self.topics:
            print(f'Starting: {topic.topic_url} ...')
            topic.start()
        for topic in self.topics:
            # workaround for Python 3.12
            topic.join()
```

Fungsi run dimulai dengan menjalankan semua topik yang ada (setiap topik kemungkinan besar berjalan dalam thread terpisah), dan kemudian menunggu agar semua topik selesai dengan memanggil join() pada setiap thread. Dengan cara ini, kita memastikan bahwa program tidak berlanjut sampai semua proses topik selesai.

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

Fungsi stop ini digunakan untuk menghentikan semua topik yang ada di dalam simulator. Setiap topik yang sedang berjalan akan dihentikan, dan pesan tentang penghentian tersebut dicetak untuk memberi tahu pengguna bahwa prosesnya telah dihentikan.