Simulasi Load Balancing

A. Load Balancing

Load balancing (pembagian beban) adalah teknik distribusi beban kerja secara merata ke beberapa server atau komponen sistem untuk memastikan bahwa tidak ada satu sumber daya yang kelebihan beban, sementara yang lain tidak digunakan secara maksimal. Tujuan utama load balancing adalah untuk meningkatkan kinerja, ketersediaan, dan reliabilitas sistem, serta memastikan waktu respons yang optimal bagi pengguna atau aplikasi.

B. Komponen Utama Load Balancing

a. **Broker:** komponen yang bertindak sebagai "pintu gerbang" yang mendistribusikan permintaan (request) ke beberapa server di belakangnya.

b. **Worker:** Server-server yang menerima permintaan dari load balancer. Pada simulasi ini menggunakan 3 worker yang menangani bagian dari beban kerja yang didistribusikan oleh Broker.

c. **Algoritma Load Balancing:** digunakan untuk menentukan cara mendistribusikan beban kerja. Pada simulasi ini menggunakan algoritma **Least Load** dimana Broker mengirimkan permintaan ke server yang memiliki jumlah koneksi aktif paling sedikit, cocok untuk permintaan yang membutuhkan waktu proses yang lama.

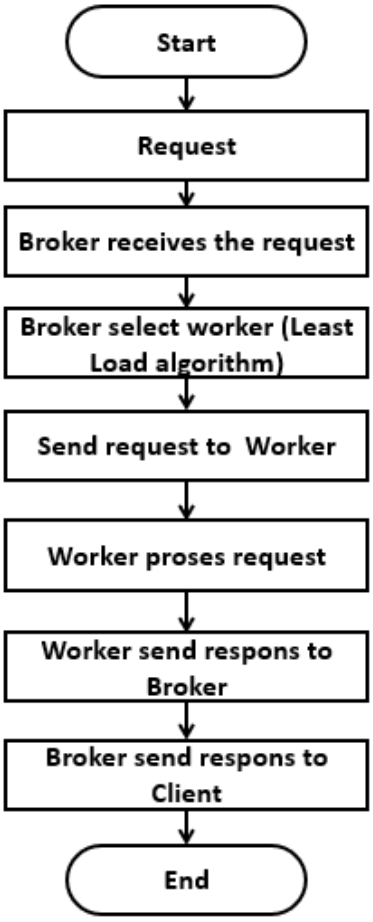
C. Alur Kerja Sistem Load Balancing

- Client membuat koneksi ke **BrokerServer** melalui socket, lalu mengirimkan tipe *request* **"Long"** atau **"Short".** Permintaan ini mewakili operasi yang berbeda dalam sistem, di mana "Long" memakan waktu lebih lama untuk diproses dibanding "Short".
- **BrokerServer** menerima koneksi dari **Client**. Ketika koneksi terhubung, broker menerima *request* dari client. **BrokerServer** kemudian mendistribusikan *request* ke 3 **WorkerServer.**

  BrokerServer menggunakan algoritma Least Load untuk memilih worker. **BrokerServer** memeriksa daftar load dari semua **WorkerServer** dan memilih **WorkerServer** dengan load paling rendah.

- **WorkerServer** menerima permintaan dari **BrokerServer** dan memprosesnya.
- Setelah memproses *request*, **WorkerServer** mengirimkan respons kembali ke **BrokerServer**, mengindikasikan bahwa permintaan telah selesai diproses.
- **BrokerServer** meneruskan respons tersebut ke **Client**.

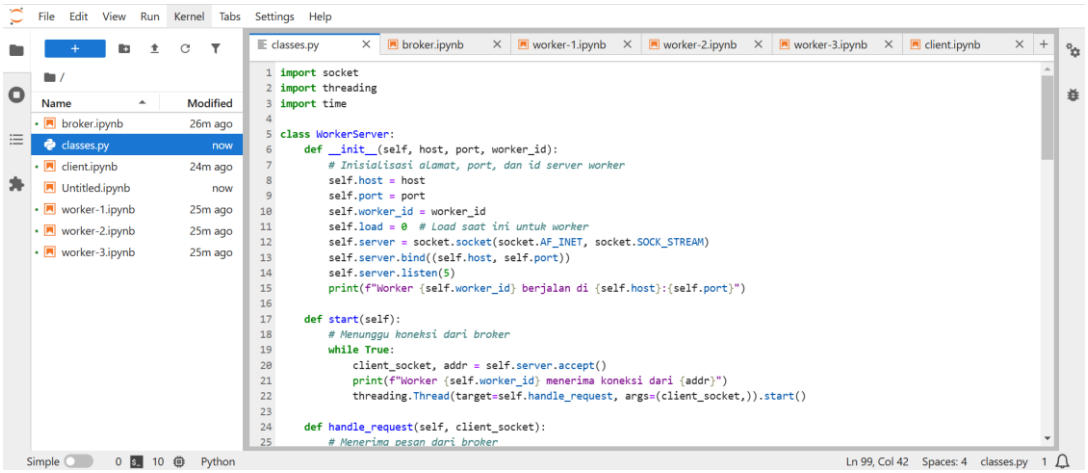
D. Flowchart Sistem Load Balancing



E. Langkah-langkah Simulasi Sistem

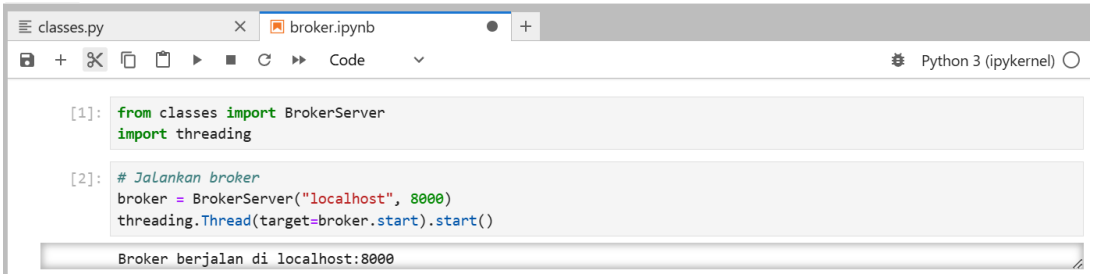
1. **Buka JupyterLab**

   Di terminal atau command prompt, navigasikan ke direktori tempat file Python berada, lalu jalankan JupyterLab dengan perintah **jupyter lab,** maka browser akan membuka file Python tersebut.



2. Buka dan jalankan File broker.ipynb

Broker diinisialisasi dengan alamat IP, port, dan daftar worker yang tersedia.

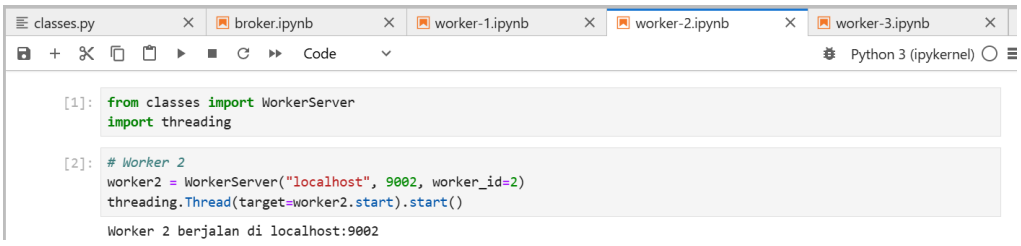


3. Buka dan jalankan File Worker

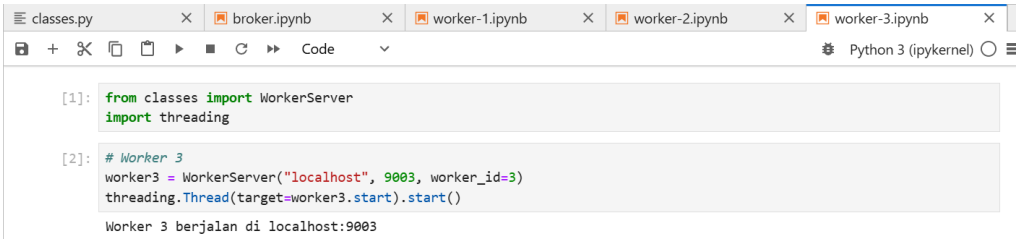
File Worker terdiri dari tiga jenis file yaitu:

a. **worker-1.ipynb** digunakan untuk menjalankan WorkerServer pertama dengan port 9001.

b. **worker-2.ipynb** digunakan untuk menjalankan WorkerServer kedua dengan port 9002.



c. **worker-3.ipynb** digunakan untuk menjalankan WorkerServer ketiga dengan port 9003.

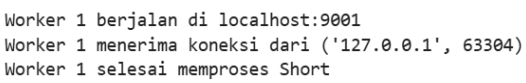


4. **Buka dan jalankan File Client.ipynb**

   Client akan mengirimkan request tipe *request* **"Long"** atau **"Short"** ke **Broker.**

5. **Broker** menerima *request* dari **Client** kemudian mendistribusikannya ke 3 **Worker.** **Broker** akan memeriksa daftar load dari semua **Worker** dan memilih **Worker** dengan load paling rendah. **Broker** akan mengupdate load setelah memilih **Worker** dengan ketentuan bobot **3** untuk tipe *request* **Long** dan bobot **1** untuk tipe *request* **Short.**

6. **Worker** akan mengirimkan *respons* setiap selesai melaksanakan *request* ke **Broker.**



7. **Broker** akan meneruskan *respons* dari **Worker** ke **Client.**

8. Proses dilanjutkan hingga semua *request* selesai dilaksanakan.

```
Broker berjalan di localhost:8000
Broker menerima koneksi dari ('127.0.0.1', 63301)
Broker menerima permintaan: Short
Update load worker: [1, 0, 0]
Broker menerima koneksi dari ('127.0.0.1', 63309)
Broker menerima permintaan: Long
Update load worker: [1, 3, 0]
Broker menerima koneksi dari ('127.0.0.1', 63319)
Broker menerima permintaan: Long
Update load worker: [1, 3, 3]
Broker menerima koneksi dari ('127.0.0.1', 63325)
Broker menerima permintaan: Short
Update load worker: [2, 3, 3]
Broker menerima koneksi dari ('127.0.0.1', 63331)
Broker menerima permintaan: Short
Update load worker: [3, 3, 3]
Broker menerima koneksi dari ('127.0.0.1', 63337)
Broker menerima permintaan: Short
Update load worker: [4, 3, 3]
Broker menerima koneksi dari ('127.0.0.1', 63342)
Broker menerima permintaan: Short
Update load worker: [4, 4, 3]
Broker menerima koneksi dari ('127.0.0.1', 63346)
Broker menerima permintaan: Short
Update load worker: [4, 4, 4]
Broker menerima koneksi dari ('127.0.0.1', 63350)
Broker menerima permintaan: Short
Update load worker: [5, 4, 4]
Broker menerima koneksi dari ('127.0.0.1', 63354)
Broker menerima permintaan: Long
Update load worker: [5, 7, 4]
```

```
Worker 1 berjalan di localhost:9001
Worker 1 menerima koneksi dari ('127.0.0.1', 63304)
Worker 1 selesai memproses Short
Worker 1 menerima koneksi dari ('127.0.0.1', 63326)
Worker 1 selesai memproses Short
Worker 1 menerima koneksi dari ('127.0.0.1', 63332)
Worker 1 selesai memproses Short
Worker 1 menerima koneksi dari ('127.0.0.1', 63338)
Worker 1 selesai memproses Short
Worker 1 menerima koneksi dari ('127.0.0.1', 63351)
Worker 1 selesai memproses Short
```

```
Worker 2 berjalan di localhost:9002
Worker 2 menerima koneksi dari ('127.0.0.1', 63310)
Worker 2 selesai memproses Long
Worker 2 menerima koneksi dari ('127.0.0.1', 63343)
Worker 2 selesai memproses Short
Worker 2 menerima koneksi dari ('127.0.0.1', 63355)
Worker 2 selesai memproses Long
```

```
Worker 3 berjalan di localhost:9003
Worker 3 menerima koneksi dari ('127.0.0.1', 63320)
Worker 3 selesai memproses Long
Worker 3 menerima koneksi dari ('127.0.0.1', 63347)
Worker 3 selesai memproses Short
```

```
Client menerima response: Worker 1 selesai memproses Short
Client menerima response: Worker 2 selesai memproses Long
Client menerima response: Worker 3 selesai memproses Long
Client menerima response: Worker 1 selesai memproses Short
Client menerima response: Worker 1 selesai memproses Short
Client menerima response: Worker 1 selesai memproses Short
Client menerima response: Worker 2 selesai memproses Short
Client menerima response: Worker 3 selesai memproses Short
Client menerima response: Worker 1 selesai memproses Short
Client menerima response: Worker 2 selesai memproses Long
```

Rezky Arisanti Putri (24051905008)
S2 Informatika