

# Panduan: Simulasi Load Balancing

## Deskripsi

Panduan ini bertujuan untuk memahami implementasi client-server socket yang mensimulasikan “*load balancing*” secara sederhana melalui Bahasa Python. Sistem terdiri dari tiga komponen utama: *Client*, *Broker Server*, dan *Worker Servers*. Setiap komponen berfungsi menangani *request* menggunakan dua metode *load balancing*.

---

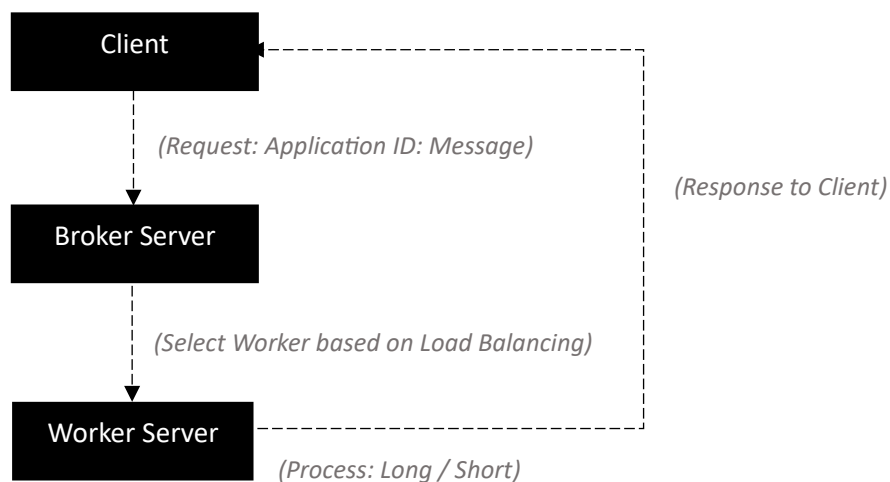
## Gambaran Umum Proses

- Server utama (**Broker Server**) menerima *request* dari **client**
- **Broker Server** mendistribusikan *request* ke 3 server lain (**Worker Servers**) dalam “cluster”, menggunakan dua pendekatan alokasi yang ditentukan.
  1. Pemerataan jumlah *request* yang sudah dilayani oleh masing-masing server pekerja (**Worker Server**)
  2. Berurutan sesuai server-id berdasar *counter* pada server utama
- Masing-masing **Worker Server** mempunyai dua jenis aplikasi yang dilayani
  1. Long => seperti perhitungan yang kompleks (dalam simulasi ini dimisalkan perhitungan factorial)
  2. Short => seperti echo atau perhitungan sederhana (dalam simulasi ini dimisalkan perhitungan penjumlahan)
- **Client** mengirimkan sintaks ke **Broker Server** dengan menyebutkan *Application ID*.
- **Broker Server** mencetak pesan (*print*) informasi ke layar. **Worker Servers** memproses *request* sesuai dengan jenis aplikasi (*Application ID*) yang diminta.

---

## Flowchart Alur Kerja

Berikut adalah flowchart yang menjelaskan alur kerja sistem load balancing ini:



## Langkah-langkah Menjalankan Sistem

### Persiapan

1. Pastikan telah menginstal Python di sistem. Python dapat diunduh dari [Python Official Website](#).
2. Pastikan semua file script (*client.py*, *broker\_server.py* dan *worker\_server.py*) berada dalam satu direktori.

### Menjalankan Worker Servers

1. Buka terminal baru untuk setiap Worker Server (total 3 terminal).
2. Di setiap terminal, jalankan Worker Server dengan perintah berikut (ganti nomor port sesuai dengan yang diinginkan):

Terminal 1:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\UNUNG\Works\2024 - YYYY - UNESA - S2 INFORMATIKA\02. KOMPUTASI JARINGAN\Socket Programming\Load Balancing
> python .\worker_server.py 8081
Worker server running on port 8081
█
```

Terminal 2:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\UNUNG\Works\2024 - YYYY - UNESA - S2 INFORMATIKA\02. KOMPUTASI JARINGAN\Socket Programming\Load Balancing
> python .\worker_server.py 8082
Worker server running on port 8082
█
```

Terminal 3:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\UNUNG\Works\2024 - YYYY - UNESA - S2 INFORMATIKA\02. KOMPUTASI JARINGAN\Socket Programming\Load Balancing
> python .\worker_server.py 8083
Worker server running on port 8083
█
```

### Menjalankan Broker Server

Di terminal baru, jalankan Broker Server, dengan memilih metode, menggunakan perintah berikut:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\UNUNG\Works\2024 - YYYY - UNESA - S2 INFORMATIKA\02. KOMPUTASI JARINGAN\Socket Programming\Load Balancing
> python .\broker_server.py 8080 1
Broker server running on port 8080 using method 1
█
```

## Mengirim Request dari Client

Di terminal baru, jalankan Client untuk mengirim request:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\UNUNG\Works\2024 - YYYY - UNESA - S2 INFORMATIKA\02. KOMPUTASI JARINGAN\Socket Programming\Load Balancing
> python .\client.py
```

Client akan mengirim dua request dengan Application ID yang berbeda (Long dan Short) ke Broker Server.

## Melihat Hasil

- Setiap **Worker Server** akan menampilkan hasil pemrosesan *request* yang diterima dari **Broker Server**.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\UNUNG\Works\2024 - YYYY - UNESA - S2 INFORMATIKA\02. KOMPUTASI JARINGAN\Socket Programming\Load Balancing
> python .\worker_server.py 8081
Worker server running on port 8081
Worker on port 8081 received request: Long:This is a test request for long computation. Result: 120
█
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\UNUNG\Works\2024 - YYYY - UNESA - S2 INFORMATIKA\02. KOMPUTASI JARINGAN\Socket Programming\Load Balancing
> python .\worker_server.py 8082
Worker server running on port 8082
Worker on port 8082 received request: Short:This is a test request for short computation. Result: 7
█
```

- **Broker Server** akan *print* pesan informasi mengenai *request* yang diterima dan **worker** yang dipilih untuk memproses *request*.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\UNUNG\Works\2024 - YYYY - UNESA - S2 INFORMATIKA\02. KOMPUTASI JARINGAN\Socket Programming\Load Balancing
> python .\broker_server.py 8080 1
Broker server running on port 8080 using method 1
Broker forwarded request 'Long:This is a test request for long computation. Result: 120' to worker on port 8081
Broker forwarded request 'Short:This is a test request for short computation. Result: 7' to worker on port 8082
█
```

---

**Nama Mahasiswa:** Unung Istopo Hartanto

**NIM:** 24051905001