

**LAPORAN TUGAS 2: Load Testing Restful API
SISTEM TERDISTRIBUSI**



Nama : **Naufal Hisyam Muzakki**
NIM : **L0123106**
Hari, Tanggal : **05 Oktober 2025**
Dosen : **Fajar Muslim S.T., M.T.**

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA
UNIVERSITAS SEBELAS MARET
2025**

BAB I

PERALATAN YANG DIGUNAKAN

Perangkat yang digunakan dalam praktikum ini antara lain:

- 1) Laptop dengan Windows 11
- 2) *Command prompt* pada Windows 11
- 3) Server dengan bahasa golang 1.25
- 4) Python 3.12
- 5) Locust

BAB II

LANGKAH-LANGKAH PRAKTIKUM

1. Jalankan server yang akan di test

```
❖ ls
README.md  dockerfile  generate_html.py  go.mod  go.sum  html  main.go
❖ go run main.go
[GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery middleware already attached.

[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env:   export GIN_MODE=release
- using code:  gin.SetMode(gin.ReleaseMode)

[GIN-debug] GET    /static/*filepath --> github.com/gin-gonic/gin.(*RouterGroup).createStaticHandler.func1 (3 handlers)
[GIN-debug] HEAD  /static/*filepath --> github.com/gin-gonic/gin.(*RouterGroup).createStaticHandler.func1 (3 handlers)
[GIN-debug] GET    /                --> main.main.func1 (3 handlers)
[GIN-debug] GET    /sepuluhkb       --> main.main.func2 (3 handlers)
[GIN-debug] GET    /seratuskb       --> main.main.func3 (3 handlers)
[GIN-debug] GET    /satumb          --> main.main.func4 (3 handlers)
[GIN-debug] GET    /limamb          --> main.main.func5 (3 handlers)
[GIN-debug] GET    /sepuluhmb       --> main.main.func6 (3 handlers)
[GIN-debug] [WARNING] You trusted all proxies, this is NOT safe. We recommend you to set a value.
Please check https://pkg.go.dev/github.com/gin-gonic/gin#readme-don-t-trust-all-proxies for details.
[GIN-debug] Listening and serving HTTP on :8080
```

Contohnya ini adalah server yang di buat pada tugas 1 sistem terdistribusi

2. Membuat python script into load test menggunakan Locust

Code :

```
from locust import HttpUser, task, between

class WebsiteUser(HttpUser):
    wait_time = between(0.1, 2)

    def on_start(self):
        self.request_timeout = 10

    @task(5)
    def load_home(self):
        self.client.get("/", timeout=self.request_timeout)

    @task(5)
    def load_10kb(self):
        self.client.get("/sepuluhkb", timeout=self.request_timeout)

    @task(5)
    def load_100kb(self):
        self.client.get("/seratuskb", timeout=self.request_timeout)

    @task(5)
    def load_1mb(self):
        self.client.get("/satumb", timeout=self.request_timeout)

    @task(5)
    def load_5mb(self):
        self.client.get("/limamb", timeout=self.request_timeout)

    @task(5)
    def load_10mb(self):
        self.client.get("/sepuluhmb", timeout=self.request_timeout)
```

Script locust python ini melakukan load testing dengan cara melakukan request ke semua endpoint di sebuah server yang dibuat pada tugas 1. Setiap endpoint diberikan bobot task 5, artinya setiap endpoint akan dipanggil dengan frekuensi yang sama, sehingga beban didistribusikan merata ke semua endpoint. Di dalam on_start, setiap request diberikan timeout 10 detik menggunakan self.request_timeout = 10, artinya jika server tidak merespons dalam 10 detik, request tersebut akan dianggap gagal. ini dimaksudkan agar simulasi mendekati kondisi nyata di mana server memiliki batas waktu untuk merespons, sehingga jika server terlalu lambat atau terlalu penuh, request akan timeout dan dihitung sebagai failure.

3. Mulai Locust

Untuk memulai locust dan menjalankan script, jalankan perintah ini di dalam command prompt :

```
locust -f locus.py --headless -u 2000 -r 100 -t 1m --host https://tugas.halonopal.space -  
-csv result
```

note:

- -f locus.py : Menentukan file Python yang berisi skrip load testing (dalam hal ini locus.py).
- --headless : Menjalankan Locust tanpa GUI, biasanya untuk running di terminal atau server.
- -u 2000 : Jumlah maksimum virtual user (VU) yang akan dijalankan, di sini 2000 user.
- -r 100 : Kecepatan spawn virtual user, 100 user baru per detik sampai mencapai jumlah maksimum.
- -t 1m : Durasi testing, yaitu 1 menit.
- --host https://tugas.halonopal.space : Alamat server yang akan diuji.
- --csv result : Menyimpan semua statistik dan hasil testing ke file CSV dengan prefix result.

4. Analisa Hasil yang dikeluarkan oleh locust

result_exceptions.csv
result_failures.csv
result_stats_history.csv
result_stats.csv

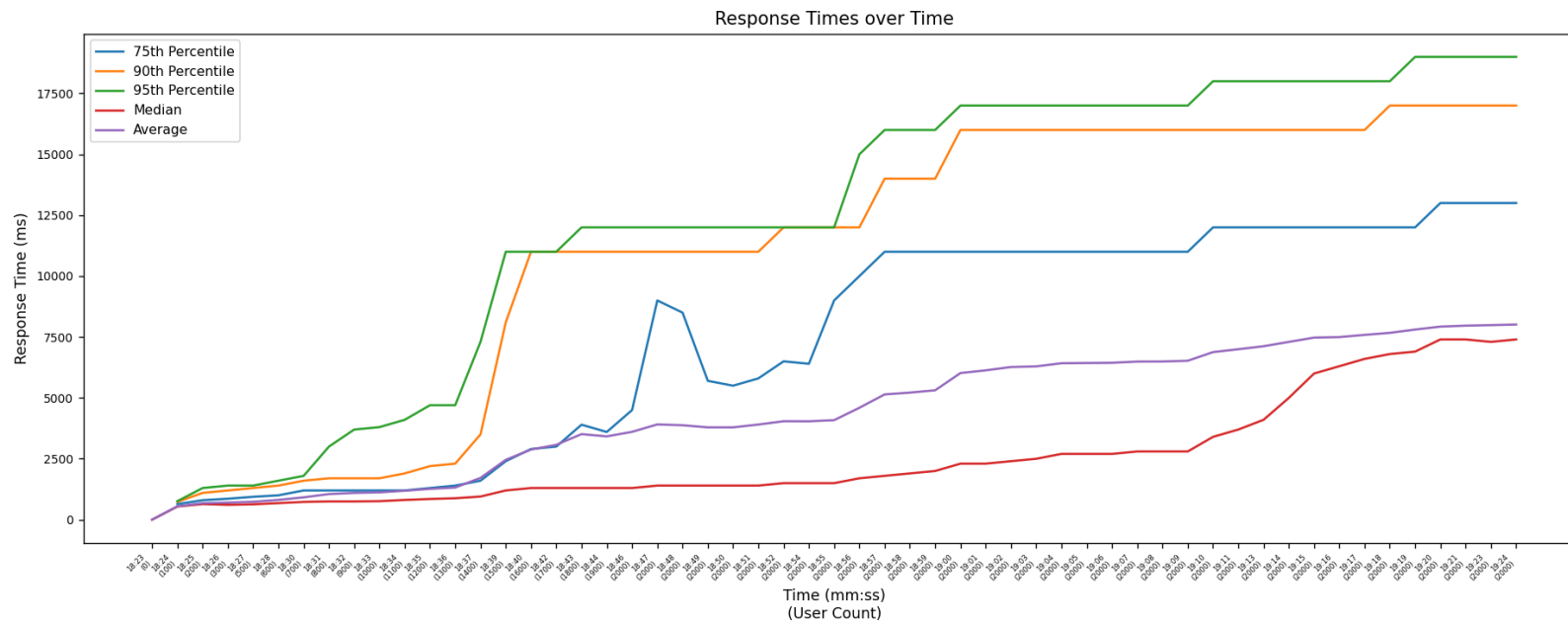
[2025-10-05 11:19:24,159] MSI/INFO/Locust.main: Shutting down (exit code 1)

Type	Name	# reqs	# fails	Avg	Min	Max	Med	req/s	failures/s
GET	/	1619	689(42.56%)	7293	143	38336	6100	26.88	11.44
GET	/limamb	1433	644(44.94%)	9119	407	45407	10000	23.79	10.69
GET	/satumb	1531	614(40.10%)	7552	259	30928	6700	25.42	10.19
GET	/sepuluhkb	1565	667(42.62%)	7761	186	31208	7000	25.98	11.07
GET	/sepuluhmb	1441	688(47.74%)	9236	411	38616	10000	23.92	11.42
GET	/seratuskb	1518	631(41.57%)	7292	188	30497	6000	25.20	10.47
Aggregated		9107	3933(43.19%)	8012	143	45407	7400	151.18	65.29

BAB III

HASIL PRAKTIKUM

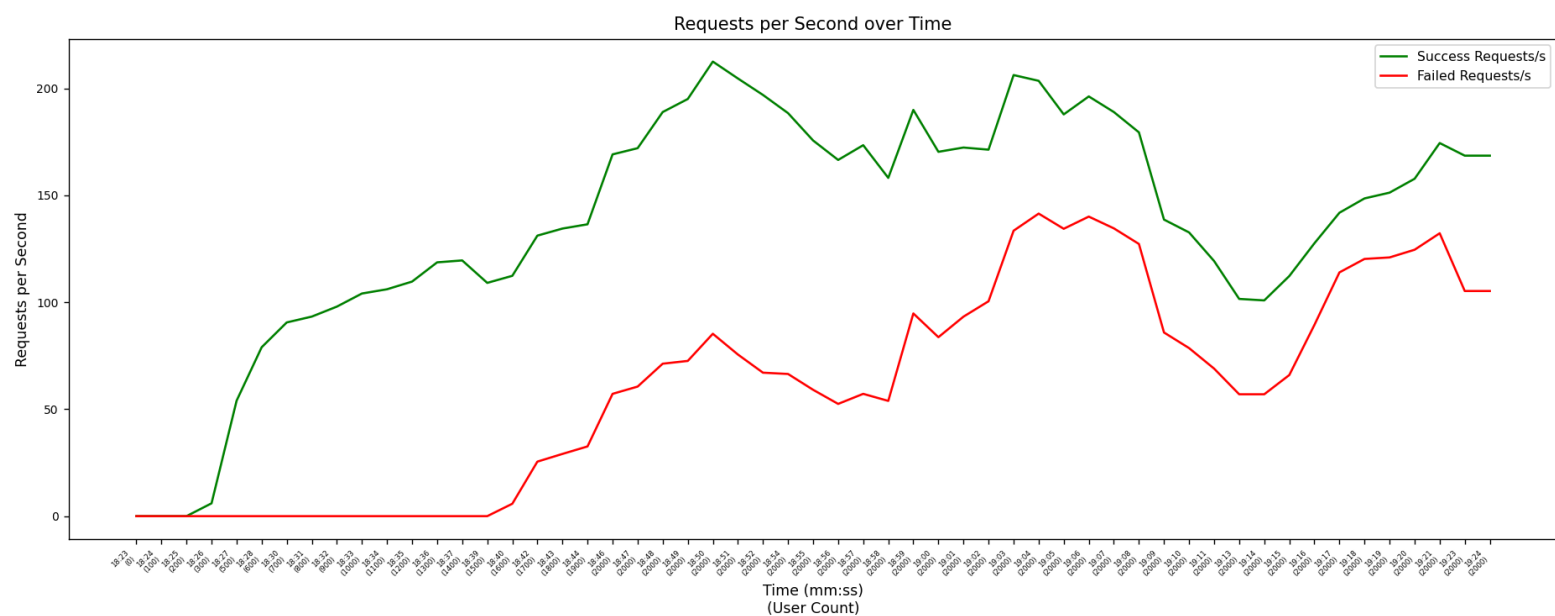
Hasil Response Time



Pada pengujian ini, server menunjukkan perilaku normal hingga sekitar 18:34, di mana latensi median dan percentile atas (50%-95%) masih relatif stabil dan meningkat secara bertahap seiring bertambahnya jumlah pengguna. Namun, mulai sekitar 18:36, terlihat lonjakan dramatis pada percentile 90% dan 95%, dari kisaran 4–5 detik menjadi 7–12 detik, dan terus meningkat hingga lebih dari 20 detik pada sekitar 18:42. Hal ini menandakan server mulai kesulitan memproses request, terutama untuk request-request di percentile atas, yang menunggu lebih lama sebelum mendapatkan respons. Seiring bertambahnya jumlah user mendekati 2000, terlihat lonjakan signifikan pada average dan max response time, terutama untuk endpoint /sepuluhmb dan /limamb, di mana max response mencapai lebih dari 10.000 ms (10 s). Median response juga mulai naik tajam, mencapai 6–7 s pada puncak load, sementara rata-rata response untuk semua endpoint mendekati 8 s. Hal ini menunjukkan server mulai kesulitan memproses request secara bersamaan, terutama untuk request yang lebih berat atau kompleks, sehingga sebagian request mengalami penundaan panjang. Lonjakan ini menunjukkan bottleneck, di mana kapasitas server sudah tidak mampu mengimbangi jumlah request simultan dari 2000 pengguna aktif. Namun,

server mulai mengalami kesulitan menerima request sekitar 18:36, dengan puncak latensi tercatat antara 18:36 hingga 18:42, terutama di 90%-95% percentile, dengan failure percentage 50%

Hasil Success/Failure rate



Hasil Akhir

Analisa Failure Rate :

Type	Name	# reqs	# fails	Avg	Min	Max	Med	req/s	failures/s
GET	/	1619	689(42.56%)	7293	143	38336	6100	26.88	11.44
GET	/limamb	1433	644(44.94%)	9119	407	45407	10000	23.79	10.69
GET	/satumb	1531	614(40.10%)	7552	259	30928	6700	25.42	10.19
GET	/sepuluhkb	1565	667(42.62%)	7761	186	31208	7000	25.98	11.07
GET	/sepuluhmb	1441	688(47.74%)	9236	411	38616	10000	23.92	11.42
GET	/seratuskb	1518	631(41.57%)	7292	188	30497	6000	25.20	10.47
Aggregated		9107	3933(43.19%)	8012	143	45407	7400	151.18	65.29

Total failure rate kumulatif selama pengujian mencapai 43,19% (3933 dari 9107 request). Ini menunjukkan server tidak mampu memproses request 2000 user secara bersamaan, sehingga banyak request yang gagal atau timeout.

Analisa Response Time :

Response time percentiles (approximated)													# reqs
Type	Name	50%	66%	75%	80%	90%	95%	98%	99%	99.9%	99.99%	100%	
GET	/	6100	11000	12000	13000	16000	17000	19000	21000	26000	38000	38000	1619
GET	/limamb	10000	12000	14000	16000	19000	21000	26000	29000	41000	45000	45000	1433
GET	/satumb	6700	11000	12000	13000	17000	18000	20000	23000	30000	31000	31000	1531
GET	/sepuluhkb	7000	11000	13000	14000	17000	19000	20000	21000	31000	31000	31000	1565
GET	/sepuluhmb	10000	12000	14000	16000	19000	21000	26000	27000	39000	39000	39000	1441
GET	/seratuskb	6100	11000	12000	13000	16000	18000	20000	23000	26000	30000	30000	1518
Aggregated		7400	11000	13000	14000	17000	19000	23000	25000	36000	45000	45000	9107

Rata-rata response time untuk seluruh endpoint selama pengujian mencapai sekitar 8 detik (8012 ms), dengan median berada di 7,4 detik (7400 ms). Sebagian besar request sudah mengalami latensi tinggi. terdapat outlier ekstrem dengan response time maksimum hingga 45 detik. Persentil 50–99,99% menunjukkan lonjakan signifikan di level 90–100%, ini menunjukkan bahwa request berat akan diproses terlambat secara signifikan. server mulai mengalami overload ketika jumlah user mendekati 2000. Endpoint dengan beban lebih berat, seperti /limamb dan /sepuluhmb, menunjukkan rata-rata dan median response time lebih tinggi dibandingkan endpoint ringan.

Analisa Error:

Error report # occurrences	Error
421	GET /seratuskb: ReadTimeout(ReadTimeoutError("HTTPSConnectionPool(host='tugas.halonopal.space', port=443): Read timed out. (read timeout=10)"))
425	GET /: ReadTimeout(ReadTimeoutError("HTTPSConnectionPool(host='tugas.halonopal.space', port=443): Read timed out. (read timeout=10)"))
426	GET /sepoluhhb: ReadTimeout(ReadTimeoutError("HTTPSConnectionPool(host='tugas.halonopal.space', port=443): Read timed out. (read timeout=10)"))
434	GET /sepoluhmb: ReadTimeout(ReadTimeoutError("HTTPSConnectionPool(host='tugas.halonopal.space', port=443): Read timed out. (read timeout=10)"))
422	GET /satumb: ReadTimeout(ReadTimeoutError("HTTPSConnectionPool(host='tugas.halonopal.space', port=443): Read timed out. (read timeout=10)"))
482	GET /limamb: ReadTimeout(ReadTimeoutError("HTTPSConnectionPool(host='tugas.halonopal.space', port=443): Read timed out. (read timeout=10)"))
189	GET /sepoluhhb: ConnectTimeoutError(<urllib3.connection.HTTPSConnection object at 0x000001640D180C48>, 'Connection to tugas.halonopal.space timed out. (connect timeout=10)')
285	GET /: ConnectTimeoutError(<urllib3.connection.HTTPSConnection object at 0x000001640D180C48>, 'Connection to tugas.halonopal.space timed out. (connect timeout=10)')
35	GET /limamb: ConnectionResetError(10054, 'An existing connection was forcibly closed by the remote host', None, 10054, None)
46	GET /sepoluhhb: ConnectionResetError(10054, 'An existing connection was forcibly closed by the remote host', None, 10054, None)
42	GET /seratuskb: ConnectionResetError(10054, 'An existing connection was forcibly closed by the remote host', None, 10054, None)
36	GET /: ConnectionResetError(10054, 'An existing connection was forcibly closed by the remote host', None, 10054, None)
38	GET /sepoluhmb: ConnectionResetError(10054, 'An existing connection was forcibly closed by the remote host', None, 10054, None)
39	GET /satumb: ConnectionResetError(10054, 'An existing connection was forcibly closed by the remote host', None, 10054, None)
152	GET /seratuskb: ConnectTimeoutError(<urllib3.connection.HTTPSConnection object at 0x000001640CCE1160>, 'Connection to tugas.halonopal.space timed out. (connect timeout=10)')
173	GET /sepoluhmb: ConnectTimeoutError(<urllib3.connection.HTTPSConnection object at 0x000001640CCE248B0>, 'Connection to tugas.halonopal.space timed out. (connect timeout=10)')
141	GET /satumb: ConnectTimeoutError(<urllib3.connection.HTTPSConnection object at 0x000001640CCE248B0>, 'Connection to tugas.halonopal.space timed out. (connect timeout=10)')
153	GET /limamb: ConnectTimeoutError(<urllib3.connection.HTTPSConnection object at 0x000001640CCE256A0>, 'Connection to tugas.halonopal.space timed out. (connect timeout=10)')
37	GET /limamb: ReadTimeoutError("HTTPSConnectionPool(host='tugas.halonopal.space', port=443): Read timed out.")
32	GET /sepoluhmb: ReadTimeoutError("HTTPSConnectionPool(host='tugas.halonopal.space', port=443): Read timed out.")
23	GET /: RemoteDisconnected('Remote end closed connection without response')
12	GET /satumb: RemoteDisconnected('Remote end closed connection without response')
15	GET /seratuskb: RemoteDisconnected('Remote end closed connection without response')
1	GET /seratuskb: SSLError(SSLError(8, '[SSL: UNEXPECTED_EOF_WHILE_READING] EOF occurred in violation of protocol (_ssl.c:1000)'))
3	GET /limamb: SSLError(SSLError(8, '[SSL: UNEXPECTED_EOF_WHILE_READING] EOF occurred in violation of protocol (_ssl.c:1000)'))
1	GET /sepoluhhb: SSLError(SSLError(8, '[SSL: UNEXPECTED_EOF_WHILE_READING] EOF occurred in violation of protocol (_ssl.c:1000)'))
1	GET /sepoluhmb: SSLError(SSLError(8, '[SSL: UNEXPECTED_EOF_WHILE_READING] EOF occurred in violation of protocol (_ssl.c:1000)'))
14	GET /limamb: RemoteDisconnected('Remote end closed connection without response')
10	GET /sepoluhmb: RemoteDisconnected('Remote end closed connection without response')
11	GET /sepoluhhb: RemoteDisconnected('Remote end closed connection without response')

Jenis Error	Jumlah	Indikasi
ReadTimeoutError	421–434 per endpoint	Server terlalu lambat merespon request → timeout baca → indikasi overload atau query lambat
ConnectTimeoutError	152–173 per endpoint	Server tidak bisa menerima koneksi baru tepat waktu → indikasi socket queue penuh / server kehabisan resource
ConnectionResetError	35–46 per endpoint	Koneksi ditutup paksa oleh server → indikasi server memutus koneksi karena resource limit
RemoteDisconnected	12–37 per endpoint	Server menutup koneksi sebelum merespon → indikasi overload / crash sementara
SSLError	1–3 per endpoint	Masalah TLS/SSL, minor tapi muncul ketika load tinggi, kemungkinan akibat connection reset / timeout

Mayoritas error disebabkan oleh ReadTimeout, ConnectTimeout, dan ConnectionReset menunjukkan server mulai tidak mampu menangani jumlah user tinggi (~2000+), Response time tinggi dan failure rate hampir 50% pada beberapa endpoint menegaskan server mencapai titik batas kapasitasnya.