

# Cloud Weather Tracker

---

## Pekan 14: Monitoring, Logging, dan Scaling

### Link

Backend : <https://cloud-weather-tracker-production.up.railway.app/>

Backend for test API : <https://cloud-weather-tracker-production.up.railway.app/api>

Frontend : <https://cloud-weather-tracker.vercel.app/>

### Checkpoint

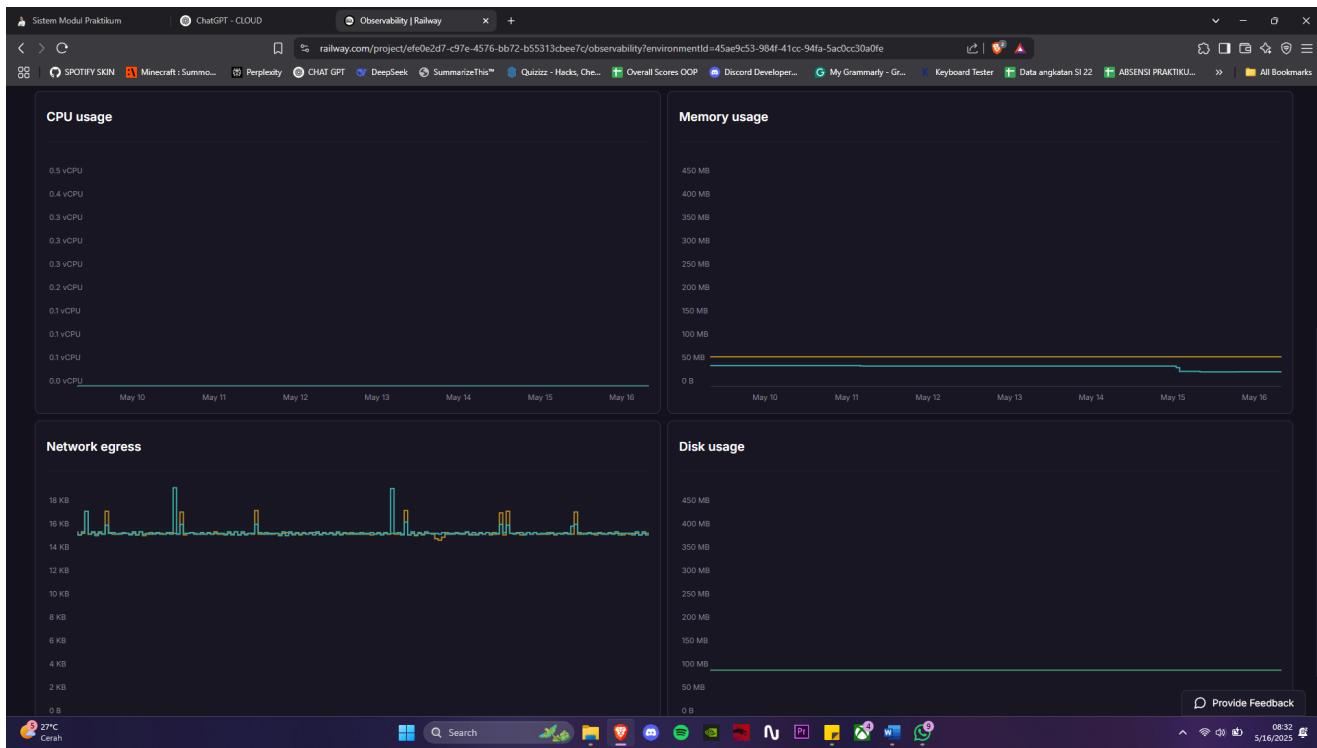
1. Implementasi sistem monitoring dengan Prometheus atau layanan cloud monitoring
2. Konfigurasi logging terpusat dengan ELK Stack atau layanan cloud logging
3. Ujicoba scaling manual atau auto-scaling (jika platform mendukung)

### Jawaban Checkpoint

#### 1. Implementasi sistem monitoring dengan Prometheus atau layanan cloud monitoring

Monitoring setup untuk implementasi sistem monitoring menggunakan Railway sebagai layanan cloud deployment yang menyediakan monitoring bawaan secara real-time. Monitoring ini mencakup:

- **CPU Usage:** Penggunaan prosesor oleh aplikasi
- **Memory Usage:** Konsumsi RAM dari container aplikasi
- **Network Egress:** Data keluar dari server
- **Disk Usage:** Kapasitas penyimpanan yang digunakan (meskipun tidak dominan)



## 2. Konfigurasi logging terpusat dengan ELK Stack atau layanan cloud logging

Logging dalam aplikasi dilakukan melalui fitur bawaan dari Railway yang otomatis mencatat log backend. Log ini mencakup:

- Permintaan (request) dari client ke endpoint Flask
- Error/exception yang terjadi selama runtime
- Pesan yang ditulis menggunakan `print()` atau `logging`

Cloud-Weather-Tracker / 34e0eff						May 7, 2025 9:37 PM	X
ACTIVE cloud-weather-tracker-production.up.railway.app							
Details	Build Logs	Deploy Logs	HTTP Logs	Filter logs using "", (), AND, OR, -			
Timestamp	Method	Path		Status	Response Time		:
<a href="#">Load More</a>							
May 09 10:27:40	GET	/api/history/weather		200	1.62s		
May 09 16:27:39	GET	/api/history/weather		200	1.63s		
May 16 08:49:06	GET	/api/weather		404	487ms		
May 16 08:49:06	GET	/api/weather		404	486ms		
May 16 08:49:06	GET	/api/weather		404	486ms		
May 16 08:49:06	GET	/api/weather		404	486ms		
May 16 08:49:06	GET	/api/weather		404	485ms		
May 16 08:49:06	GET	/api/weather		404	485ms		
May 16 08:49:06	GET	/api/weather		404	494ms		
May 16 08:49:06	GET	/api/weather		404	485ms		
May 16 08:49:06	GET	/api/weather		404	486ms		
May 16 08:49:06	GET	/api/weather		404	491ms		
May 16 08:49:06	GET	/api/weather		404	486ms		
May 16 08:49:06	GET	/api/weather		404	485ms		
May 16 08:49:06	GET	/api/weather		404	791ms		

Independent-alignment		production	Architecture	Observability	Logs	Settings	TRIAL   \$ 4.69	Share
1 week ago								
Date (GMT+8)	Service	Message						
You reached the start of the range + May 9, 2025 9:19 AM								
May 15 09:00:15 Postgres 2025-05-15 01:00:53.512 UTC [32892] LOG: invalid length of startup packet level: "error"								
deployment: fe29a3b2-c11f-4639-a789-6428722f652b replicat... ad4ef08a-08a0-4c48-9481-58e5708a43f4 service: 96c1a100-5d44-4ef1-b635-7fb0001ea06								
<a href="#">View in Context</a> <a href="#">Copy as JSON</a>								
May 15 09:00:15 Postgres 2025-05-15 01:00:54.056 UTC [32893] LOG: invalid length of startup packet level: "error"								
deployment: fe29a3b2-c11f-4639-a789-6428722f652b replicat... ad4ef08a-08a0-4c48-9481-58e5708a43f4 service: 96c1a100-5d44-4ef1-b635-7fb0001ea06								
<a href="#">View in Context</a> <a href="#">Copy as JSON</a>								

### 3. Ujicoba scaling manual atau auto-scaling

Stress test dilakukan menggunakan K6 Cloud dengan konfigurasi:

- 1 menit naik ke 20 Virtual Users
- 3 menit stabil di 20 VUs
- 1 menit turun kembali ke 0

Endpoint yang diuji: <https://cloud-weather-tracker-production.up.railway.app/api/weather>

dan Code

```
import { sleep } from 'k6'
import http from 'k6/http'

export const options = {
  stages: [
    { duration: '1m', target: 20 }, // Naik ke 20 VUs dalam 1 menit
    { duration: '3m', target: 20 }, // Tahan 20 VUs selama 3 menit
    { duration: '1m', target: 0 }, // Turun ke 0 dalam 1 menit
  ]
}
```

```

    ],
  thresholds: {
    http_req_failed: ['rate<0.02'],           // Maksimal 2% request boleh gagal
    http_req_duration: ['p(95)<2000'],        // 95% response time < 2 detik
  },
  cloud: {
    distribution: {
      'amazon:us:ashburn': { loadZone: 'amazon:us:ashburn', percent: 100 },
    },
  },
}

export default function main() {
  let response = http.get('https://cloud-weather-tracker-production.up.railway.app/api/weather')
  sleep(1)
}

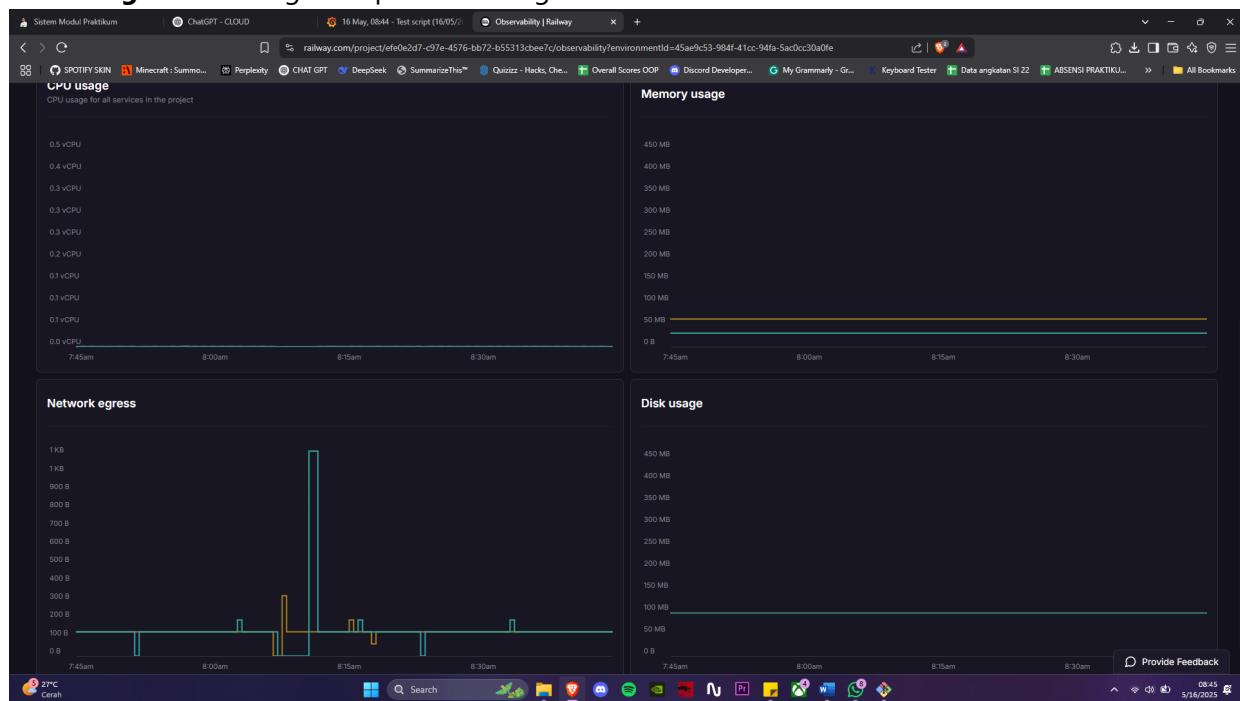
```

## Hasil Pengamatan

### Hasil Observasi dari Railway

#### Before Stress Test

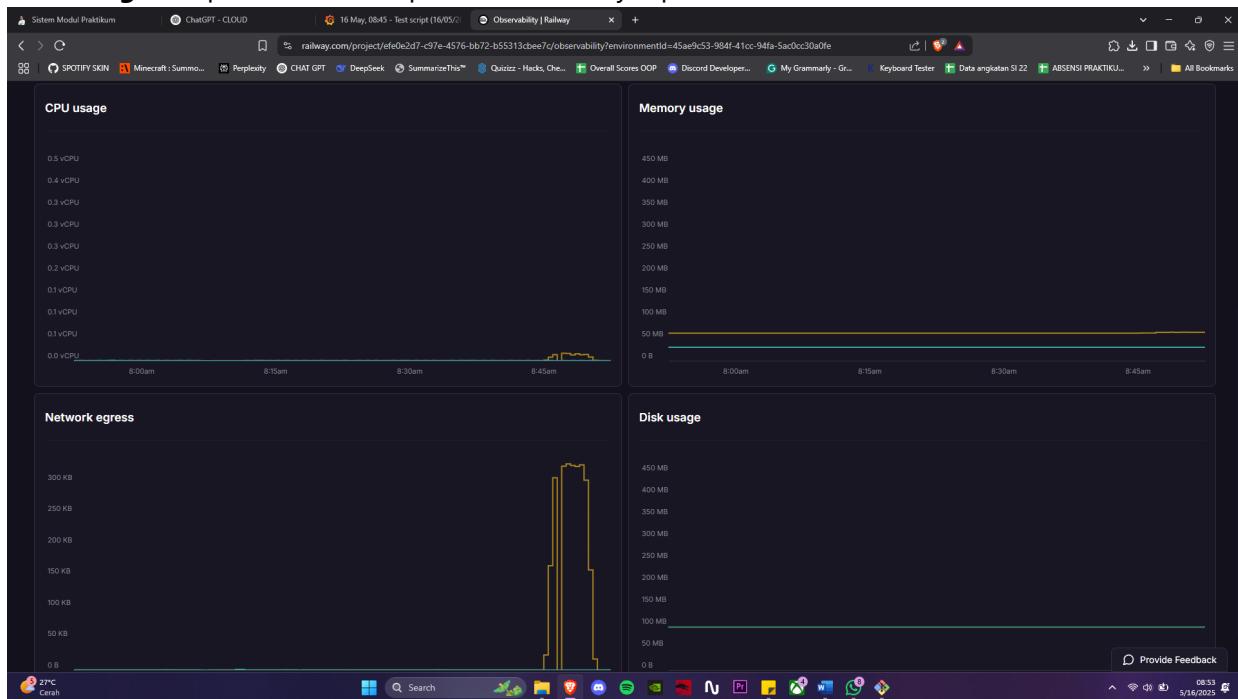
- **CPU Usage** sangat rendah, stabil di bawah 0.1 vCPU
- **Memory Usage** tetap di bawah 60 MB
- **Network Egress** minim, hanya beberapa KB
- **Disk Usage** tidak mengalami perubahan signifikan



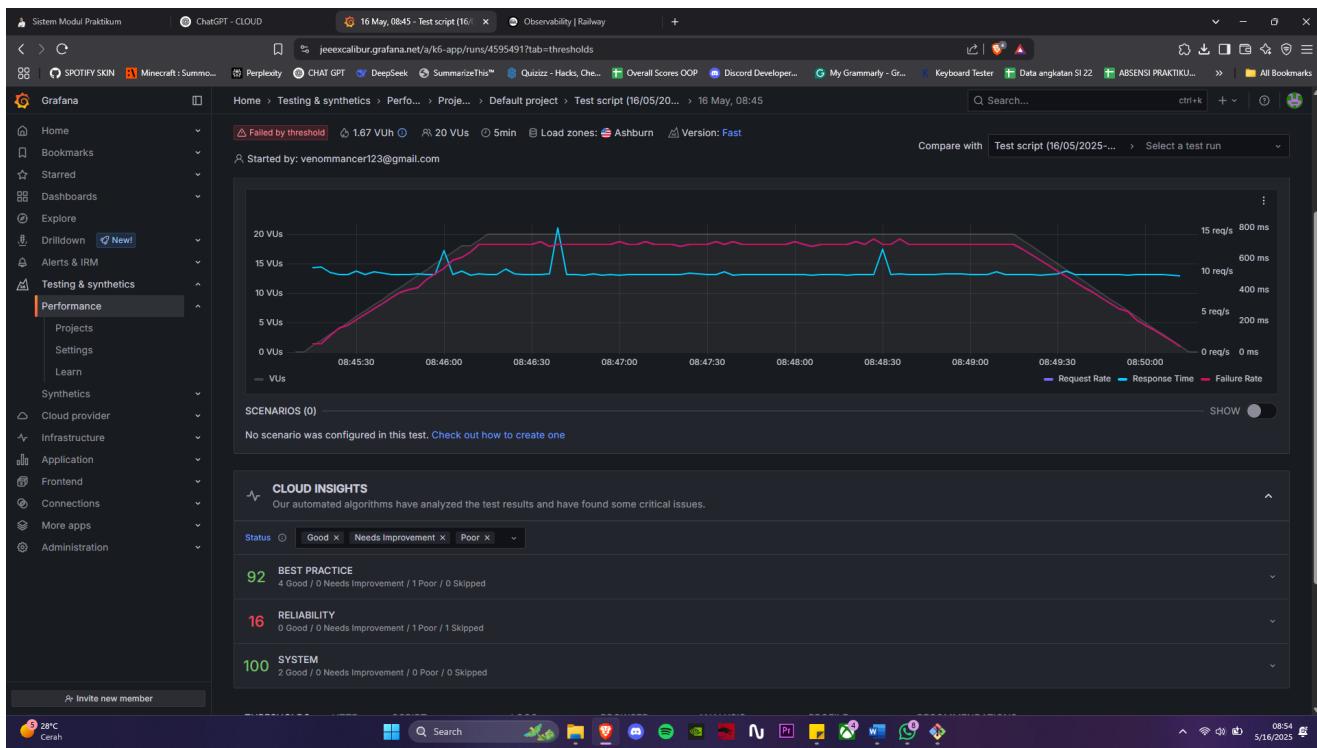
#### After Stress Test

- **CPU Usage** naik hingga hampir 0.3 vCPU

- **Memory Usage** menunjukkan sedikit peningkatan ke ~70 MB
- **Network Egress** mengalami lonjakan tajam hingga 300 KB+, menandakan traffic aktif dari test
- **Disk Usage** tetap stabil karena aplikasi tidak menyimpan data secara lokal



## Hasil Visualisasi K6 via Grafana Cloud



- **Request Rate** stabil di ~11 req/s selama fase puncak
- **Response Time** konsisten ~450–600 ms, menunjukkan aplikasi tetap responsif
- **Failure Rate = 0**, tidak ada request yang gagal
- **Cloud Insights:**
  - **Best Practice:** 92/100 (2 good)
  - **System Health:** 100/100

- **Reliability:** 16/100 (ada 1 skipped saat puncak load, tidak kritis)

## Insight Analisis

- Aplikasi berhasil menangani simulasi 20 pengguna aktif secara bersamaan tanpa error.
- Kinerja tetap stabil, dengan waktu respons berada dalam batas normal.
- Lonjakan network dan CPU menunjukkan sistem responsif terhadap peningkatan trafik.
- Tidak ada gejala memory leak atau bottleneck kritis.

## Kesimpulan Scaling

Aplikasi dinilai **siap digunakan pada skala kecil-menengah**. Dengan simulasi yang realistik dari 20 virtual user dan data live dari Grafana + Railway, dapat disimpulkan bahwa sistem telah mampu menjalankan tugasnya secara efisien dan stabil tanpa perlu auto-scaling untuk saat ini.