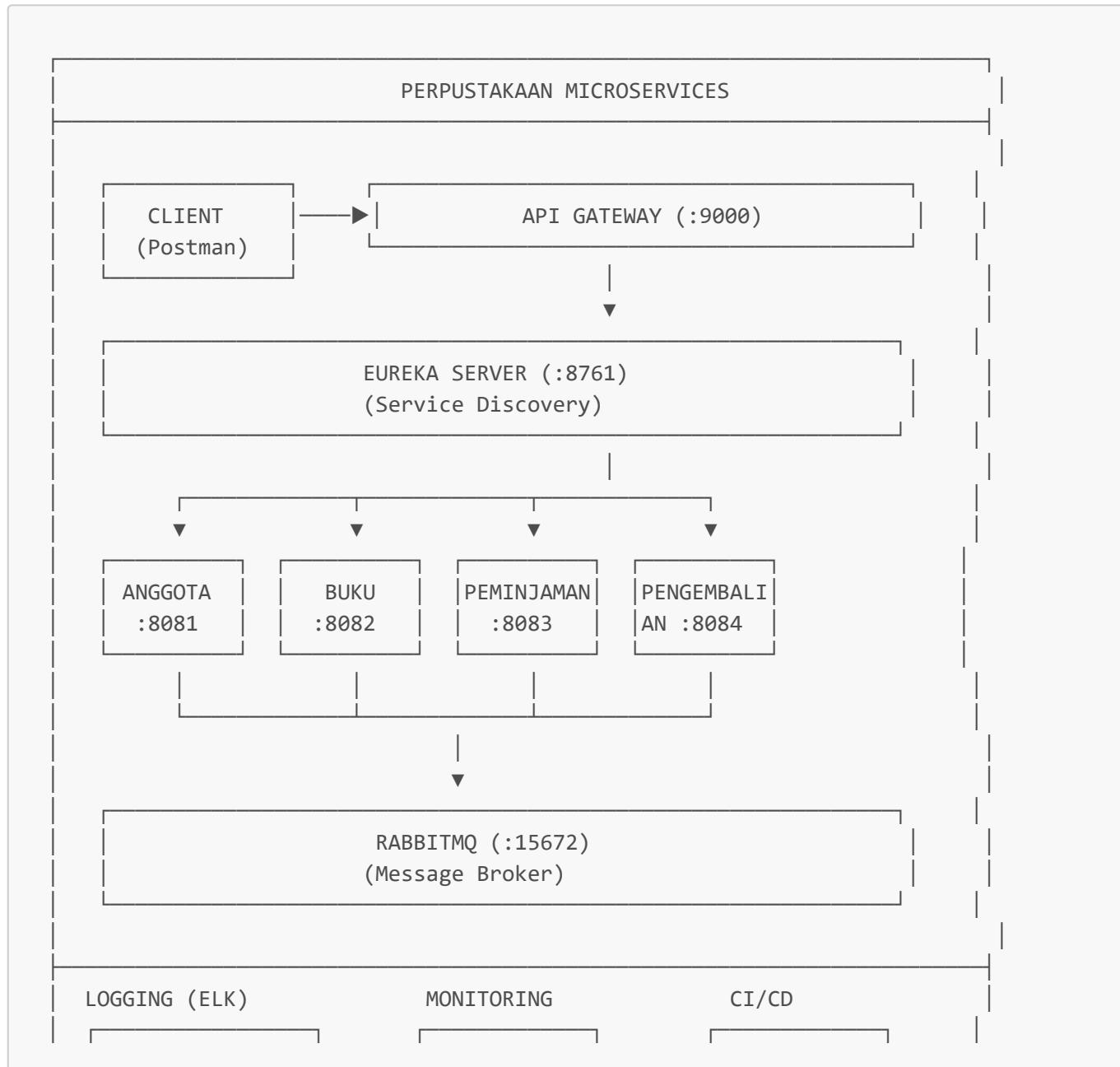


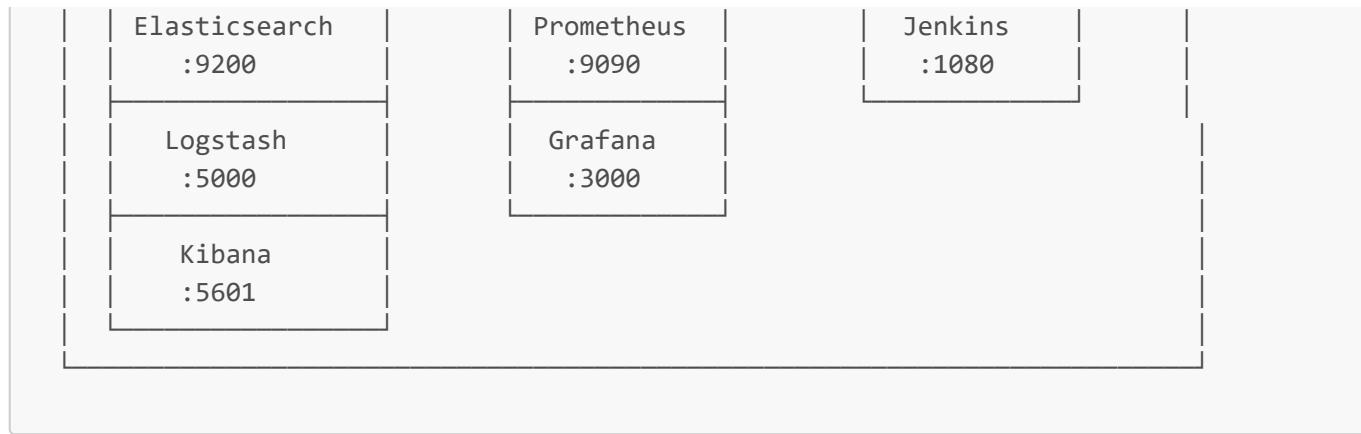
PANDUAN LENGKAP SISTEM PERPUSTAKAAN MICROSERVICES

Daftar Isi

1. [Tahap 1: Menjalankan Semua Service](#)
2. [Tahap 2: Verifikasi Infrastructure](#)
3. [Tahap 3: Verifikasi Microservices](#)
4. [Tahap 4: Setup Logging \(Kibana\)](#)
5. [Tahap 5: Setup Monitoring \(Grafana\)](#)
6. [Tahap 6: Setup CI/CD \(Jenkins\)](#)
7. [Tahap 7: Testing API](#)

ARSITEKTUR SISTEM





TAHAP 1: MENJALANKAN SEMUA SERVICE

Apa yang dilakukan?

Menjalankan semua container Docker yang berisi microservices dan tools pendukung.

Langkah:

1. Buka **PowerShell** atau **Terminal**
2. Masuk ke folder project:

```
cd D:\PERPUSTAKAAN_key
```

3. Jalankan Docker Compose:

```
docker-compose up -d
```

4. Tunggu sampai semua container running (\pm 2-3 menit)

Cek Status Container:

```
docker-compose ps
```

Hasil yang diharapkan:

Semua container berstatus **Up** atau **Running**

Container	Port	Fungsi
server-eureka	8761	Service Discovery
rabbitmq	15672	Message Broker
anggota-service	8081	Kelola data anggota

Container	Port	Fungsi
buku-service	8082	Kelola data buku
peminjaman-service	8083	Kelola peminjaman
pengembalian-service	8084	Kelola pengembalian
api-gateway	9000	Gateway utama
elasticsearch	9200	Penyimpanan log
logstash	5000	Pemrosesan log
kibana	5601	Visualisasi log
prometheus	9090	Pengumpulan metrics
grafana	3000	Dashboard monitoring
jenkins	1080	CI/CD Pipeline

TAHAP 2: VERIFIKASI INFRASTRUCTURE

2.1 Cek Eureka Server (Service Discovery)

Apa itu Eureka? Eureka adalah "buku telepon" untuk microservices. Semua service mendaftarkan diri ke Eureka agar bisa saling menemukan.

Langkah:

1. Buka browser
2. Akses: <http://localhost:8761>
3. Scroll ke bagian "**Instances currently registered with Eureka**"

Yang harus terlihat:

- ANGGOTA-SERVICE
- BUKU-SERVICE
- PEMINJAMAN-SERVICE
- PENGEMBALIAN-SERVICE
- API-GATEWAY

Jika tidak muncul? Tunggu 1-2 menit, lalu refresh. Service butuh waktu untuk register.

2.2 Cek RabbitMQ (Message Broker)

Apa itu RabbitMQ? RabbitMQ adalah "kantor pos" untuk microservices. Service mengirim pesan ke RabbitMQ, lalu RabbitMQ meneruskan ke service tujuan.

Langkah:

1. Buka browser

2. Akses: <http://localhost:15672>

3. Login:

- Username: **guest**
- Password: **guest**

Yang harus terlihat:

- Dashboard RabbitMQ
 - Connections dan Channels aktif
 - Queues yang digunakan service
-

TAHAP 3: VERIFIKASI MICROSERVICES

3.1 Cek Health Setiap Service

Buka browser dan akses setiap URL berikut:

Service	URL Health Check	Status Normal
Anggota	http://localhost:8081/actuator/health	{"status": "UP"}
Buku	http://localhost:8082/actuator/health	{"status": "UP"}
Peminjaman	http://localhost:8083/actuator/health	{"status": "UP"}
Pengembalian	http://localhost:8084/actuator/health	{"status": "UP"}
API Gateway	http://localhost:9000/actuator/health	{"status": "UP"}

3.2 Cek via API Gateway

Semua request bisa melalui API Gateway (port 9000):

Service	URL via Gateway
Anggota	http://localhost:9000/api/anggota
Buku	http://localhost:9000/api/buku
Peminjaman	http://localhost:9000/api/peminjaman
Pengembalian	http://localhost:9000/api/pengembalian

TAHAP 4: SETUP LOGGING (KIBANA)

Apa itu ELK Stack?

- **Elasticsearch**: Database untuk menyimpan log
- **Logstash**: Mengumpulkan dan memproses log
- **Kibana**: Dashboard untuk melihat log

4.1 Cek Elasticsearch

1. Buka: <http://localhost:9200>
2. Harus muncul JSON dengan info cluster

4.2 Setup Kibana

1. Buka: <http://localhost:5601>
2. Tunggu loading selesai (bisa 1-2 menit pertama kali)

4.3 Buat Data View (Index Pattern)

1. Klik **☰ Menu** (kiri atas)
2. Klik **Stack Management** (di bagian Management)
3. Klik **Data Views** (di sidebar kiri)
4. Klik tombol **Create data view**
5. Isi form:
 - **Name:** Perpustakaan Logs
 - **Index pattern:** logstash-perpustakaan-*
 - **Timestamp field:** Pilih @timestamp
6. Klik **Save data view to Kibana**

4.4 Lihat Log

1. Klik **☰ Menu**
2. Klik **Discover**
3. Pilih data view **Perpustakaan Logs**
4. Atur time range di kanan atas: **Last 24 hours** atau **Today**
5. Log dari semua service akan muncul!

4.5 Filter Log (Opsional)

Di search bar, ketik:

```
service_name: "anggota-service"
```

atau

```
level: "ERROR"
```

TAHAP 5: SETUP MONITORING (GRAFANA)

Apa itu Prometheus & Grafana?

- **Prometheus:** Mengumpulkan metrics (CPU, memory, request count, dll)
- **Grafana:** Dashboard untuk visualisasi metrics

5.1 Cek Prometheus

1. Buka: <http://localhost:9090>
2. Klik **Status** → **Targets**
3. Pastikan semua target berstatus **UP** (hijau)

5.2 Login Grafana

1. Buka: <http://localhost:3000>
2. Login:
 - Username: **admin**
 - Password: **admin**
3. Klik **Skip** atau ganti password

5.3 Tambah Data Source

1. Klik **☰ Menu** → **Connections** → **Data sources**
2. Klik **Add data source**
3. Pilih **Prometheus**
4. Isi URL: <http://prometheus:9090>
5. Klik **Save & test**
6. Harus muncul "Successfully queried the Prometheus API"

5.4 Buat Dashboard Sederhana

1. Klik **☰ Menu** → **Dashboards**
2. Klik **New** → **New Dashboard**
3. Klik **Add visualization**
4. Pilih data source **Prometheus**

Panel 1 - Memory Usage:

```
sum(jvm_memory_used_bytes{area="heap"}) by (application)
```

- Title: **Heap Memory Used**
- Klik **Apply**

Panel 2 - HTTP Requests:

1. Klik **Add** → **Visualization**
2. Query:

```
rate(http_server_requests_seconds_count[5m])
```

- Title: **HTTP Request Rate**
- Klik **Apply**

Panel 3 - Uptime:

1. Klik **Add** → **Visualization**

2. Query:

```
process_uptime_seconds
```

- Title: **Service Uptime**
- Klik **Apply**

5.5 Simpan Dashboard

1. Klik  **Save** (kanan atas)
 2. Nama: **Perpustakaan Dashboard**
 3. Klik **Save**
-

TAHAP 6: SETUP CI/CD (JENKINS)

Apa itu Jenkins?

Jenkins adalah tool untuk otomatisasi build dan deploy. Setiap kali code di-push ke GitHub, Jenkins akan otomatis build & test.

6.1 Dapatkan Password Awal

Buka PowerShell, jalankan:

```
docker exec jenkins cat /var/jenkins_home/secrets/initialAdminPassword
```

Catat password yang muncul!

6.2 Setup Jenkins

1. Buka: <http://localhost:1080>
2. Paste password dari langkah 6.1
3. Klik **Install suggested plugins**
4. Tunggu instalasi selesai (5-10 menit)
5. Buat admin user:
 - Username: **admin**
 - Password: **admin123**
 - Full name: **Administrator**
 - Email: **admin@perpustakaan.com**
6. Klik **Save and Continue**
7. Jenkins URL: biarkan default
8. Klik **Start using Jenkins**

6.3 Setup Maven

1. Klik **Manage Jenkins** (sidebar kiri)
2. Klik **Tools**
3. Scroll ke **Maven installations**
4. Klik **Add Maven**
5. Isi:
 - Name: **MAVEN**
 - Centang **Install automatically**
 - Version: pilih **3.9.6** (atau terbaru)
6. Klik **Save**

6.4 Buat Pipeline

1. Klik **New Item** (sidebar kiri)
2. Nama: **perpustakaan-pipeline**
3. Pilih **Pipeline**
4. Klik **OK**

6.5 Konfigurasi Pipeline

1. Scroll ke bagian **Pipeline**
2. Definition: pilih **Pipeline script from SCM**
3. SCM: pilih **Git**
4. Repository URL:

```
https://github.com/LatifaKeysha/perpustakaan_key.git
```

5. Branch: *** /main**
6. Script Path: **Jenkinsfile**
7. Klik **Save**

6.6 Jalankan Pipeline

1. Klik **Build Now** (sidebar kiri)
2. Lihat progress di **Build History**
3. Klik nomor build (#1)
4. Klik **Console Output** untuk lihat log detail

TAHAP 7: TESTING API

7.1 Gunakan Postman

Download Postman: <https://www.postman.com/downloads/>

7.2 Test CRUD Anggota

CREATE - Tambah Anggota Baru:

```
POST http://localhost:9000/api/anggota
Content-Type: application/json

{
    "nama": "John Doe",
    "email": "john@email.com",
    "alamat": "Jakarta",
    "noTelepon": "08123456789"
}
```

READ - Lihat Semua Anggota:

```
GET http://localhost:9000/api/anggota
```

READ - Lihat Anggota by ID:

```
GET http://localhost:9000/api/anggota/1
```

UPDATE - Edit Anggota:

```
PUT http://localhost:9000/api/anggota/1
Content-Type: application/json

{
    "nama": "John Updated",
    "email": "john.updated@email.com",
    "alamat": "Bandung",
    "noTelepon": "08111111111"
}
```

DELETE - Hapus Anggota:

```
DELETE http://localhost:9000/api/anggota/1
```

7.3 Test CRUD Buku

CREATE - Tambah Buku:

```
POST http://localhost:9000/api/buku
Content-Type: application/json
```

```
{  
    "judul": "Belajar Java",  
    "penulis": "Programmer Indonesia",  
    "penerbit": "Tech Publisher",  
    "tahunTerbit": 2024,  
    "stok": 10  
}
```

READ - Lihat Semua Buku:

```
GET http://localhost:9000/api/buku
```

7.4 Test Peminjaman**CREATE - Buat Peminjaman:**

```
POST http://localhost:9000/api/peminjaman  
Content-Type: application/json  
  
{  
    "anggotaId": 1,  
    "bukuId": 1,  
    "tanggalPinjam": "2026-01-19",  
    "tanggalKembali": "2026-01-26"  
}
```

READ - Lihat Semua Peminjaman:

```
GET http://localhost:9000/api/peminjaman
```

📋 QUICK REFERENCE - SEMUA URL

Service/Tool	URL	Credentials
Eureka	http://localhost:8761	-
RabbitMQ	http://localhost:15672	guest / guest
API Gateway	http://localhost:9000	-
Anggota	http://localhost:8081	-
Buku	http://localhost:8082	-
Peminjaman	http://localhost:8083	-

Service/Tool	URL	Credentials
Pengembalian	http://localhost:8084	-
Elasticsearch	http://localhost:9200	-
Kibana	http://localhost:5601	-
Prometheus	http://localhost:9090	-
Grafana	http://localhost:3000	admin / admin
Jenkins	http://localhost:1080	admin / admin123

🔧 TROUBLESHOOTING

Container tidak mau start?

```
docker-compose down
docker-compose up -d
```

Service tidak muncul di Eureka?

Tunggu 1-2 menit, lalu refresh browser.

Kibana tidak bisa buat index pattern?

Pastikan ada log yang masuk. Coba hit beberapa API dulu, lalu coba lagi.

Grafana tidak ada data?

1. Cek Prometheus targets: http://localhost:9090/targets
2. Pastikan semua target UP
3. Tunggu 1-2 menit untuk data terkumpul

Jenkins build gagal?

1. Pastikan Maven sudah di-setup di Tools
2. Cek Console Output untuk error detail

☑ CHECKLIST URUTAN PENGERJAAN

- 1. Jalankan `docker-compose up -d`
- 2. Cek Eureka: <http://localhost:8761>
- 3. Cek RabbitMQ: <http://localhost:15672>
- 4. Cek health semua service
- 5. Setup Kibana Data View
- 6. Lihat log di Kibana Discover
- 7. Setup Grafana Data Source

- 8. Buat Grafana Dashboard
 - 9. Setup Jenkins + Maven
 - 10. Buat dan jalankan Pipeline
 - 11. Test API dengan Postman
-

 **SELAMAT! Sistem Perpustakaan Microservices Anda sudah siap!**