

Modul Praktikum Microservices

Tujuan Praktikum:

1. Mampu mengimplementasikan layanan microservices sederhana.
2. Menguasai tools dan teknologi pendukung microservices (Docker, Kubernetes, API Gateway, dll.).
3. Mampu mengintegrasikan dan mengelola sistem berbasis microservices.

Rincian Modul Praktikum

Pertemuan	Topik Praktikum	Tugas Praktikum	Tools/Technologies
1	Pengenalan Tools dan Setup Environment	- Install Docker, Postman, dan IDE (Visual Studio Code/IntelliJ). - Setup environment development.	Docker, Postman, Visual Studio Code
2	Membangun REST API Sederhana	- Buat REST API untuk CRUD (Create, Read, Update, Delete) data produk.	Node.js/Spring Boot, Postman
3	Implementasi Service Discovery	- Implementasi service discovery menggunakan Eureka atau Consul.	Spring Cloud Eureka, Consul
4	Membangun API Gateway	- Buat API Gateway untuk mengarahkan request ke layanan yang sesuai.	Spring Cloud Gateway, Kong
5	Komunikasi Antar Microservices (Synchronous)	- Implementasi komunikasi antar layanan menggunakan REST API.	Spring Boot, REST Template/Feign Client
6	Komunikasi Antar Microservices (Asynchronous)	- Implementasi komunikasi asynchronous menggunakan message queue (RabbitMQ/Kafka).	RabbitMQ, Apache Kafka
7	Manajemen Data dengan Database Per Service	- Buat layanan terpisah dengan database independen (MySQL/PostgreSQL).	MySQL, PostgreSQL
8	Implementasi Event Sourcing dan CQRS	- Buat sistem event-driven menggunakan event sourcing dan CQRS.	Axon Framework, Kafka
9	Containerization dengan Docker	- Buat Dockerfile untuk setiap layanan dan jalankan dalam container Docker.	Docker, Docker Compose
10	Orchestration dengan Kubernetes	- Deploy microservices ke cluster Kubernetes.	Kubernetes, Minikube
11	Monitoring dan Logging	- Implementasi monitoring menggunakan Prometheus dan Grafana.	Prometheus, Grafana

Pertemuan Topik Praktikum		Tugas Praktikum	Tools/Technologies
12	Centralized Logging dengan ELK Stack	- Setup ELK Stack (Elasticsearch, Logstash, Kibana) untuk logging terpusat.	ELK Stack
13	Testing Microservices	- Lakukan unit testing, integration testing, dan contract testing.	JUnit, Postman, Pact
14	Implementasi CI/CD Pipeline	- Buat pipeline CI/CD menggunakan Jenkins atau GitLab CI.	Jenkins, GitLab CI
15	Scalability dan Resilience	- Implementasi pola resilience (circuit breaker, retry, fallback).	Spring Cloud Circuit Breaker, Hystrix
16	Proyek Akhir: Implementasi Sistem Microservices	- Bangun sistem microservices lengkap (minimal 3 layanan) dengan fitur CI/CD, monitoring, dan logging.	Semua tools yang telah dipelajari

Detail Tugas Praktikum

Pertemuan 1: Pengenalan Tools dan Setup Environment

- **Tugas:**
 1. Install Docker, Postman, dan IDE (Visual Studio Code/IntelliJ).
 2. Setup environment development (Java/Node.js, Maven/NPM).
- **Output:**
 - Docker terinstall dan bisa menjalankan container sederhana.
 - Postman terinstall dan siap digunakan untuk testing API.

Pertemuan 2: Membangun REST API Sederhana

- **Tugas:**
 1. Buat REST API untuk CRUD data produk (Create, Read, Update, Delete).
 2. Gunakan Spring Boot (Java) atau Express.js (Node.js).
- **Output:**
 - API dapat diakses via Postman dan melakukan operasi CRUD.

Pertemuan 3: Implementasi Service Discovery

- **Tugas:**
 1. Setup service discovery menggunakan Eureka atau Consul.
 2. Registrasikan minimal 2 layanan ke service discovery.
- **Output:**

- Layanan terdaftar di Eureka/Consul dan dapat ditemukan oleh layanan lain.

Pertemuan 4: Membangun API Gateway

- **Tugas:**
 1. Buat API Gateway menggunakan Spring Cloud Gateway atau Kong.
 2. Arahkan request ke layanan yang sesuai berdasarkan endpoint.
- **Output:**
 - API Gateway dapat meneruskan request ke layanan yang benar.

Pertemuan 5: Komunikasi Antar Microservices (Synchronous)

- **Tugas:**
 1. Buat 2 layanan yang berkomunikasi secara synchronous menggunakan REST API.
 2. Gunakan REST Template atau Feign Client.
- **Output:**
 - Layanan A dapat memanggil Layanan B dan mendapatkan response.

Pertemuan 6: Komunikasi Antar Microservices (Asynchronous)

- **Tugas:**
 1. Implementasi komunikasi asynchronous menggunakan RabbitMQ atau Kafka.
 2. Buat producer dan consumer untuk mengirim dan menerima pesan.
- **Output:**
 - Pesan dapat dikirim dan diterima antar layanan.

Pertemuan 7: Manajemen Data dengan Database Per Service

- **Tugas:**
 1. Buat layanan terpisah dengan database independen (MySQL/PostgreSQL).
 2. Implementasi CRUD pada layanan tersebut.
- **Output:**
 - Layanan dapat menyimpan dan mengambil data dari database sendiri.

Pertemuan 8: Implementasi Event Sourcing dan CQRS

- **Tugas:**
 1. Buat sistem event-driven menggunakan event sourcing dan CQRS.
 2. Gunakan Axon Framework atau Kafka untuk event sourcing.
- **Output:**
 - Sistem dapat menghasilkan event dan memprosesnya secara terpisah.

Pertemuan 9: Containerization dengan Docker

- **Tugas:**
 1. Buat Dockerfile untuk setiap layanan.
 2. Jalankan layanan dalam container Docker menggunakan Docker Compose.
- **Output:**
 - Layanan dapat dijalankan dalam container Docker.

Pertemuan 10: Orchestration dengan Kubernetes

- **Tugas:**
 1. Deploy microservices ke cluster Kubernetes.
 2. Buat file YAML untuk deployment dan service.
- **Output:**
 - Layanan dapat diakses melalui Kubernetes cluster.

Pertemuan 11: Monitoring dan Logging

- **Tugas:**
 1. Setup monitoring menggunakan Prometheus dan Grafana.
 2. Tampilkan metrik seperti CPU usage, memory usage, dan request rate.
- **Output:**
 - Dashboard Grafana menampilkan metrik dari layanan.

Pertemuan 12: Centralized Logging dengan ELK Stack

- **Tugas:**
 1. Setup ELK Stack (Elasticsearch, Logstash, Kibana).
 2. Kirim log dari layanan ke ELK Stack.
- **Output:**
 - Log dapat dilihat dan dianalisis di Kibana.

Pertemuan 13: Testing Microservices

- **Tugas:**
 1. Lakukan unit testing dan integration testing.
 2. Implementasi contract testing menggunakan Pact.
- **Output:**
 - Layanan memiliki test coverage yang baik.

Pertemuan 14: Implementasi CI/CD Pipeline

- **Tugas:**
 1. Buat pipeline CI/CD menggunakan Jenkins atau GitLab CI.
 2. Otomatisasi build, test, dan deploy.
- **Output:**
 - Pipeline dapat menjalankan build, test, dan deploy secara otomatis.

Pertemuan 15: Scalability dan Resilience

- **Tugas:**
 1. Implementasi pola resilience seperti circuit breaker, retry, dan fallback.
 2. Gunakan Spring Cloud Circuit Breaker atau Hystrix.
- **Output:**
 - Layanan dapat menangani kegagalan dengan baik.

Pertemuan 16: Proyek Akhir

- **Tugas:**
 1. Bangun sistem microservices lengkap (minimal 3 layanan).
 2. Implementasi fitur CI/CD, monitoring, dan logging.
- **Output:**
 - Sistem microservices yang siap di-deploy dan di-monitor.

Referensi Tools dan Teknologi:

1. **Docker:** Untuk containerization.
2. **Kubernetes:** Untuk orchestration.
3. **Spring Boot:** Untuk membangun layanan microservices.
4. **RabbitMQ/Kafka:** Untuk komunikasi asynchronous.
5. **Prometheus/Grafana:** Untuk monitoring.
6. **ELK Stack:** Untuk centralized logging.
7. **Jenkins/GitLab CI:** Untuk CI/CD pipeline.