

Modul Praktikum 1

Pengantar Development Operation (DevOps)

1. Maksud

Modul praktikum ini dirancang untuk memberikan pemahaman dasar tentang konsep dan praktik Development Operation (DevOps). DevOps adalah pendekatan yang menggabungkan pengembangan perangkat lunak (Development) dan operasi IT (Operations) untuk meningkatkan kolaborasi, otomatisasi, dan efisiensi dalam siklus hidup pengembangan perangkat lunak. Melalui modul ini, mahasiswa diharapkan dapat memahami prinsip-prinsip DevOps, alat-alat yang digunakan, serta bagaimana menerapkannya dalam proyek nyata.

2. Tujuan

- a) Memahami konsep dasar DevOps dan manfaatnya dalam pengembangan perangkat lunak.
- b) Mempelajari alat-alat yang umum digunakan dalam praktik DevOps.
- c) Mampu mengimplementasikan pipeline CI/CD (Continuous Integration/Continuous Deployment) sederhana.
- d) Mengembangkan kemampuan kolaborasi antara tim pengembangan dan operasi.
- e) Meningkatkan efisiensi dan kualitas produk perangkat lunak melalui otomatisasi.

3. Dasar Teori

3.1. Pengertian DevOps

DevOps adalah budaya, filosofi, atau pendekatan yang menggabungkan tim pengembangan (Development) dan tim operasi (Operations) untuk bekerja sama secara erat dalam seluruh siklus hidup pengembangan perangkat lunak. Tujuannya adalah untuk mempercepat pengiriman produk, meningkatkan kualitas, dan memastikan keandalan sistem.

3.2. Prinsip-prinsip DevOps

- a) **Kolaborasi:** Menghilangkan silo antara tim pengembangan dan operasi.
- b) **Otomatisasi:** Mengotomatisasi proses pengujian, integrasi, dan deployment.
- c) **Continuous Integration (CI):** Mengintegrasikan kode ke dalam repositori bersama secara berkala.
- d) **Continuous Deployment (CD):** Menyebarkan kode ke lingkungan produksi secara otomatis setelah melewati tahap pengujian.
- e) **Monitoring dan Feedback:** Memantau aplikasi di lingkungan produksi dan memberikan umpan balik cepat ke tim pengembangan.

3.3. Alat-alat DevOps

- a) **Version Control System (VCS):** Git, GitHub, GitLab.
- b) **CI/CD Tools:** Jenkins, GitLab CI, CircleCI.
- c) **Configuration Management:** Ansible, Puppet, Chef.

- d) **Containerization:** Docker, Kubernetes.
- e) **Monitoring:** Prometheus, Grafana, Nagios.

4. Prosedur Praktik

4.1. Persiapan

- a) **Instalasi Git:** Pastikan Git terinstal di sistem Anda.
- b) **Instalasi Docker:** Instal Docker untuk containerization.
- c) **Instalasi Jenkins:** Siapkan Jenkins untuk pipeline CI/CD.
- d) **Buat Repositori:** Buat repositori di GitHub atau GitLab untuk proyek praktikum.

4.2. Langkah-langkah Praktik

1. Setup Version Control:

- Buat repositori Git untuk proyek Anda.
- Tambahkan file .gitignore untuk mengabaikan file yang tidak perlu di-track.
- Lakukan commit dan push ke repositori.

2. Setup CI/CD Pipeline:

- a) Buat pipeline Jenkins untuk otomatisasi build dan deployment.
- b) Konfigurasi Jenkins untuk terhubung dengan repositori Git.
- c) Tambahkan tahap build, test, dan deploy dalam pipeline.

3. Containerization dengan Docker:

- a) Buat Dockerfile untuk aplikasi Anda.
- b) Build dan jalankan container Docker.
- c) Push image Docker ke Docker Hub.

4. Monitoring:

- a) Setup Prometheus dan Grafana untuk memantau aplikasi.
- b) Tambahkan metrik dasar seperti CPU usage, memory usage, dan response time.

5. Latihan

1. Latihan 1: Menggunakan Github

Studi Kasus: Submit File ke Repository GitHub

Deskripsi Kasus

Anda adalah seorang developer yang baru bergabung dalam sebuah tim pengembangan perangkat lunak. Tim Anda menggunakan GitHub sebagai version control system untuk mengelola kode proyek. Tugas pertama Anda adalah mempelajari cara mengirim (submit)

file ke repositori GitHub menggunakan Git. Berikut adalah langkah-langkah yang perlu Anda ikuti untuk menyelesaikan tugas ini.

Langkah-langkah Latihan

1. Persiapan:

- Pastikan Git sudah terinstal di sistem Anda. Jika belum, unduh dan instal Git dari git-scm.com.
- Buat akun GitHub jika belum memiliki akun.
- Buat repositori baru di GitHub dengan nama latihan-devops.

2. Clone Repositori:

- Buka terminal atau command prompt.
- Clone repositori GitHub ke lokal menggunakan perintah berikut:

```
git clone https://github.com/username/latihan-devops.git
```

Ganti username dengan nama akun GitHub Anda.

3. Buat File Baru:

- Masuk ke direktori repositori yang telah di-clone:

```
cd latihan-devops
```
- Buat file baru bernama hello-world.txt menggunakan teks editor atau terminal:

```
echo "Hello, DevOps!" > hello-world.txt
```

4. Tambahkan File ke Staging Area:

- Tambahkan file ke staging area menggunakan perintah:

```
git add hello-world.txt
```

5. Commit Perubahan:

- Lakukan commit dengan pesan yang deskriptif:

```
git commit -m "Menambahkan file hello-world.txt"
```

6. Push ke GitHub:

- Kirim perubahan ke repositori GitHub:

```
git push origin main
```

Jika menggunakan branch selain main, ganti main dengan nama branch yang sesuai.

7. Verifikasi di GitHub:

- Buka repositori GitHub Anda di browser.

- Pastikan file hello-world.txt sudah muncul di repositori.

Latihan Tambahan

1. Modifikasi File:

- Edit file hello-world.txt dan tambahkan teks baru, misalnya:
Selamat datang di dunia DevOps!
- Ulangi langkah 4-6 untuk mengirim perubahan ke GitHub.

2. Buat Branch Baru:

- Buat branch baru bernama feature-1:
git checkout -b feature-1
- Tambahkan file baru bernama feature.txt dan push branch tersebut ke GitHub:
git add feature.txt
git commit -m "Menambahkan fitur baru"
git push origin feature-1

3. Merge Branch:

- Kembali ke branch main:
git checkout main
- Gabungkan branch feature-1 ke main:
git merge feature-1
- Push perubahan ke GitHub:
git push origin main

2. **Latihan 2:** Containerize aplikasi Anda menggunakan Docker dan deploy ke lingkungan lokal.
3. **Latihan 3:** Setup monitoring dasar menggunakan Prometheus dan visualisasikan data menggunakan Grafana.

6. Tugas

1. **Tugas 1:** Buat dokumentasi lengkap tentang pipeline CI/CD yang telah Anda buat, termasuk konfigurasi Jenkins, Dockerfile, dan langkah-langkah deployment.
2. **Tugas 2:** Lakukan analisis terhadap metrik yang dikumpulkan oleh Prometheus dan buat laporan singkat tentang performa aplikasi Anda.
3. **Tugas 3:** Jelaskan bagaimana prinsip-prinsip DevOps dapat meningkatkan efisiensi dan kualitas produk perangkat lunak dalam sebuah tim.

Dengan menyelesaikan modul ini, mahasiswa diharapkan memiliki pemahaman yang kuat tentang konsep DevOps dan mampu menerapkannya dalam proyek pengembangan perangkat lunak.