

**PERCEPATAN STUDY MATA KULIAH
TEKNOLOGI CLOUD
(PODMAN)**

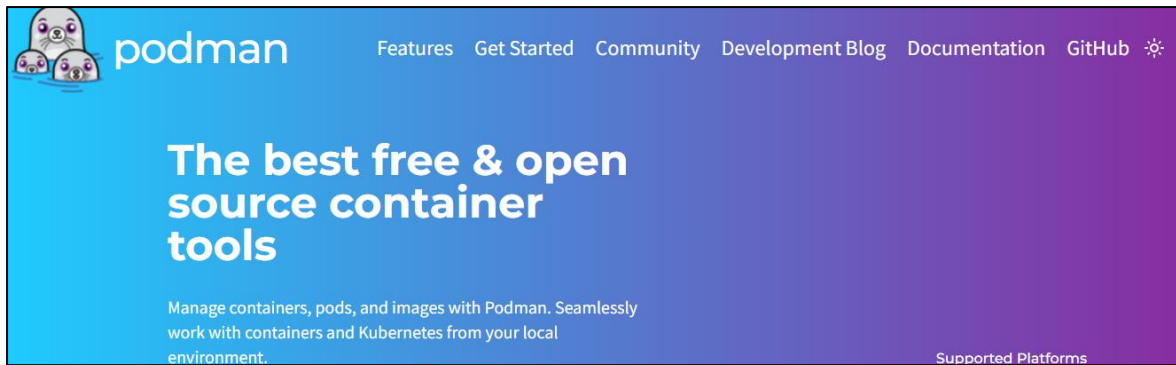


AGUNG SETIAWAN

NIM : 185610048

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA
YOGYAKARTA**

2024



Cara menjalankan program

1. Instalasi Podman

Sebelum memulai, pastikan Podman sudah terinstal. Di berbagai distribusi Linux, Anda dapat menginstalnya dengan perintah berikut:

```
sudo apt update
sudo apt install podman
```

```
sudo yum install podman
```

2. Membuat Dockerfile

Untuk memulai proyek container, Anda memerlukan sebuah Dockerfile (atau Containerfile untuk Podman). Berikut contoh Dockerfile untuk aplikasi sederhana menggunakan Node.js:

```
# Menggunakan Node.js sebagai base image
FROM node:14

# Menetapkan direktori kerja
WORKDIR /usr/src/app

# Menyalin file package.json dan package-lock.json
COPY package*.json ./

# Menginstal dependensi
RUN npm install

# Menyalin seluruh source code aplikasi
COPY . .

# Menetapkan port yang digunakan oleh aplikasi
EXPOSE 8080

# Menjalankan aplikasi
CMD [ "node", "app.js" ]
```

3. Membangun Image

Setelah memiliki docker file bangun image dengan podman, contoh membuat image dengan perintah tag `my-node app`.

```
podman build -t my-node-app .
```

4. Menjalankan container

Setelah image dibangun kita bias menjalankan container

```
podman run -d -p 8080:8080 --name my-node-container my-node-app
```

Penjelasan:

- -d menjalankan container di background.
- -p 8080:8080 memetakan port 8080 di host ke port 8080 di container.
- --name my-node-container memberikan nama pada container.

5. Memeriksa status container

Untuk melihat status container

```
podman ps
```

Untuk melihat semua status container (termasuk yang berhenti/stop)

```
podman ps -a
```

6. Masuk kedalam container

Kita perlu masuk kedalam container dengan debugging

```
podman exec -it my-node-container /bin/sh
```

7. Menghentikan dan menghapus container

```
podman stop my-node-container
```

8. Menulis Skrip Otomatisasi

Kita dapat menulis skrip shell untuk otomatisasi tugas-tugas untuk membangun seperti menjalankan container dan membersihkan.

```
#!/bin/bash

# Membangun image
podman build -t my-node-app .

# Menjalankan container
podman run -d -p 8080:8080 --name my-node-container my-node-app

# Memberikan waktu untuk aplikasi mulai
sleep 10

# Menampilkan log container
podman logs my-node-container

# Menghentikan dan menghapus container
podman stop my-node-container
podman rm my-node-container
```

Pastikan untuk hak memberi eksekusi pada skrip

```
chmod +x build-run-clean.sh
```

Kemudian menjalankan skrip

Berikut adalah Contoh Praktikum menggunakan node.js dan express, dan mengemasnya kedalam container dan menjalankan dengan podman.

1. Membuat direktori

```
mkdir my-node-app
cd my-node-app
```

2. Inisialisasi proyek node.js dan instal express

```
npm init -y
npm install express
```

3. Buat file `app.js` didalam direktori

```
const express = require('express');
const app = express();
const port = 8080;

app.get('/', (req, res) => {
  res.send('Hello World!');
});

app.listen(port, () => {
  console.log(`App listening at http://localhost:${port}`);
});
```

4. Buat dockerfile didalam direktori

```
# Menggunakan Node.js sebagai base image
FROM node:14

# Menetapkan direktori kerja
WORKDIR /usr/src/app

# Menyalin file package.json dan package-lock.json
COPY package*.json ./

# Menginstal dependensi
RUN npm install

# Menyalin seluruh source code aplikasi
COPY . .

# Menetapkan port yang digunakan oleh aplikasi
EXPOSE 8080

# Menjalankan aplikasi
CMD [ "node", "app.js" ]
```

5. Bangun image docker menggunakan podman

```
podman build -t my-node-app .
```

6. Jalankan container

```
podman run -d -p 8080:8080 --name my-node-container my-node-app
```

Buka browser dan navigasikan ke <http://localhost:8080>. Anda harus melihat pesan "Hello World!".