

LAPORAN PRAKTIKUM

Docker Modul



Disusun Oleh:

Nama : L Hafidl Alkhair
Kelas : TRKJ 2.C
Jurusan : Teknologi Informasi dan Komputer
Program Studi : Teknologi Rekayasa Komputer Jaringan
Dosen Pembimbing : Indrawati, SST., M.T.



**JURUSAN TEKNOLOGI INFORMASI KOMPUTER
PRODI TEKNOLOGI REKAYASA KOMPUTER JARINGAN
POLITEKNIK NEGERI LHOKSEUMAWE
TAHUN AJARAN 2024-2025**

LEMBAR PENGESAHAN

No Praktikum : 02/TIK/TRKJ-2C/Komputasi Awan
Judul : Laporan Praktikum Docker Flask Web Peminjaman
Nama : L Hafidl Alkhair
Kelas : TRKJ 2.C
Jurusan : Teknologi Informasi dan Komputer
Prodi : Teknologi Rekayasa Komputer Jaringan
Tanggal Praktikum : Jum'at 15 Mei
Tanggal Penyerahan : Jum'at, 21 Juni 2025

Buketrata, 21 Juni 2025

Dosen Pembimbing,

Indrawati, SST., M.T.

NIP. 19740815 200112 2 001

DAFTAR ISI

LEMBAR PENGESAHAN.....	i
DAFTAR ISI.....	ii
BAB I.....	1
1. Tujuan	1
2. Dasar Teori	1
3. Alat dan bahan	2
BAB II	3
1. Lab 2: Docker	3
BAB III	6
A. Kesimpulan.....	6

BAB I

PENDAHULUAN

1. Tujuan

Praktikum ini bertujuan untuk memberikan pemahaman dasar tentang Docker, termasuk cara:

- Menjalankan container menggunakan image yang tersedia di **Docker Hub**.
- Menggunakan **Shell** dalam container untuk berinteraksi dengan sistem operasi.
- Membuat akun di **Docker Hub** dan menyimpan image yang dibuat.
- Membuat image baru dan mengunggahnya ke **Docker Hub**.
- Menggunakan **Docker Volume** untuk penyimpanan data yang persisten.
- Menjalankan **multi-container** menggunakan **Docker Compose**.

2. Dasar Teori

Docker adalah sebuah platform yang digunakan untuk membuat, menjalankan, dan mengelola aplikasi dalam lingkungan yang terisolasi yang disebut container. Container lebih ringan dibandingkan virtual machine karena berbagi kernel dengan sistem operasi host, sehingga lebih cepat dan efisien.

Docker bekerja dengan menggunakan image, yaitu file template yang berisi sistem operasi dan aplikasi. Dari image ini, Docker dapat menjalankan container, yang merupakan lingkungan aplikasi yang berjalan secara independen. Image bisa diambil dari Docker Hub, sebuah repositori online yang menyediakan berbagai image siap pakai.

Dalam penggunaannya, Docker memiliki berbagai perintah dasar seperti `docker pull` untuk mengunduh image, `docker run` untuk menjalankan container, dan `docker ps` untuk melihat container yang sedang berjalan. Untuk menyimpan data agar tidak hilang saat container dihapus, digunakan Docker Volume.

Selain itu, Docker juga memiliki Docker Compose, yang memungkinkan pengguna menjalankan beberapa container secara bersamaan dengan satu perintah menggunakan file konfigurasi `docker-compose.yml`. Hal ini mempermudah pengelolaan aplikasi yang terdiri dari beberapa layanan, seperti web server dan database.

3. Alat dan bahan

1. Alat

- Komputer dengan sistem operasi yang mendukung Docker (Linux, Windows, macOS).
- Docker Engine yang telah terinstal.
- Docker Compose (untuk pengelolaan multi-container).
- Internet untuk mengakses Docker Hub.
- Terminal atau Command Prompt untuk menjalankan perintah Docker.

2. Bahan

- Docker Image yang digunakan dalam praktikum (misalnya hello-world, alpine, node, golang, dll.).
- Source Code untuk membuat container (misalnya salam.go, index.js, dan Dockerfile).
- Akun Docker Hub untuk menyimpan dan mengambil image.

BAB II PRAKTIKUM

1. Lab 2: Docker

1. Lab 2.1: Docker Image "node_project"

\$mkdir node_project

\$cd node_project

Setalah membuat direktori baru jangan lupa install

\$sudo apt install npm

\$npm -v

Untuk melihat versi dari npm yang ada pada linux

```
root@fid:/home/fid/node_project# npm -v
9.2.0
root@fid:/home/fid/node_project#
```

\$npm init

```
Press ^C at any time to quit.
package name: (node_project)
version: (1.0.0)
description: Cloud Computing
entry point: (index.js)
test command:
git repository:
keywords:
author: L Hafidl Alkhair
license: (ISC)
About to write to /home/fid/node_project/package.json:

{
  "name": "node_project",
  "version": "1.0.0",
  "description": "Cloud Computing",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "L Hafidl Alkhair",
  "license": "ISC"
}
```

\$npm install express

```
found 0 vulnerabilities
root@fid:/home/fid/node_project# cat package.json
{
  "name": "node_project",
  "version": "1.0.0",
  "description": "Cloud Computing",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "L Hafidl Alkhair",
  "license": "ISC",
  "dependencies": {
    "express": "^4.21.2"
  }
}
```

Buat file dengan index.js seperti berikut:

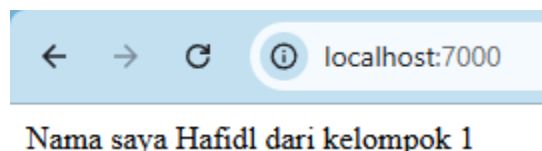
```
root@fid:/home/fid/node_project# cat index.js
const express = require("express")
const app = express()
const port = 7000

app.get("/", (req, resp) => {
  resp.send("Nama saya Hafidl dari kelompok 1");
})
app.listen(port, ()=> console.log(`listening on port ${port}`))
```

Kemudian jalan kan port dengan perintah

\$node index.js

```
root@fid:/home/fid/node_project# node index.js
listening on port 7000
```



```
Usage: docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile
root@fid:/home/fid/node_project# docker build -t hafidl/node_project:1.0 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
             Install the buildx component to build images with BuildKit:
             https://docs.docker.com/go/buildx/

Sending build context to Docker daemon   2.85MB
Step 1/7 : FROM node:latest
----> e4f23baa3e59
Step 2/7 : WORKDIR /app
----> Using cache
----> 84f3a335599e
Step 3/7 : COPY package.json .
----> Using cache
----> 26a8b3a8740f
Step 4/7 : RUN npm install
----> Using cache
----> cd40a5c7ff88
Step 5/7 : COPY . .
----> Using cache
----> cacc2b658d5f
Step 6/7 : EXPOSE 7000
----> Using cache
----> 622b23e8e38e
Step 7/7 : CMD ["node","index.js"]
----> Using cache
----> adda581a6d6a
Successfully built adda581a6d6a
Successfully tagged hafidl/node_project:1.0
root@fid:/home/fid/node_project#
```

Selanjutnya buat file dengan nama Dockerfile

```
FROM node : latest
WORKDIR / app
COPY package . json .
RUN npm install
COPY . .
EXPOSE 7000
CMD [" node ", " index .js"]
```

\$docker build -t hafidl/node_project :1.0 .

```
root@fid:/home/fid/node_project# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
b37171d6e475   hafidl/node_project:1.0             "docker-entrypoint.s..." 8 minutes ago  Up 8 minutes  7000/tcp
58f9339f6fc6   hafidl/node_project:1.0             "docker-entrypoint.s..." 12 minutes ago Up 12 minutes  0.0.0.0:80->7000/tcp, :::80->7000/
tcp_node_project
```

\$docker ps untuk melihat status image

BAB III

KESIMPULAN

A. Kesimpulan

Praktikum Docker ini memberikan pemahaman dasar yang kuat mengenai konsep dan penggunaan container dalam pengembangan aplikasi. Melalui serangkaian langkah praktis, peserta belajar bagaimana:

- Menginstal dan menggunakan Docker serta Docker Compose di lingkungan lokal.
- Menjalankan container dari image yang tersedia di Docker Hub dan mengelola container tersebut menggunakan perintah dasar seperti `docker run`, `docker ps`, dan `docker build`.
- Membangun sebuah aplikasi Node.js sederhana, membuat file Dockerfile, dan membangun image custom dari aplikasi tersebut.
- Menyimpan dan membagikan image melalui akun Docker Hub.
- Memahami konsep Docker Volume untuk menjaga data tetap tersimpan meskipun container dihentikan atau dihapus.
- Mengelola lebih dari satu container secara efisien menggunakan Docker Compose.

Dengan praktikum ini, peserta mendapatkan pengalaman langsung yang berguna untuk menerapkan Docker dalam pengembangan aplikasi nyata, khususnya dalam konteks aplikasi berbasis web yang memerlukan lingkungan konsisten dan portabel.