

# **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 LHOKEUMAWE  
TAHUN AJARAN 2024-2025**

## LEMBAR PENGESAHAN

No Praktikum : 03/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
A. Lab 3 Docker .....	3
1. Lab 3.1 : Docker Volume .....	3
2. Lab 3.2 multi kontainer dengan docker-compose.....	6
BAB III.....	8
A. Kesimpulan.....	8

## 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

#### A. Lab 3 Docker

##### 1. Lab 3.1 : Docker Volume

Docker volume dibutuhkan oleh kontainer karena kontainer tidak dapat menyimpan data secara persisten. Docker volume menawarkan fisikal disk yang ada di Host sebagai media penyimpanan untuk kontainer. Dengan demikian jika kontainer selesai bekerja dan hilang dari Host, maka data tersebut tetap ada.

*\$docker info*

- Menampilkan informasi lengkap tentang konfigurasi Docker, termasuk jumlah container yang berjalan, jumlah image yang tersedia, versi Docker, dan detail lainnya

*\$docker images*

- Menampilkan daftar container yang sedang berjalan.
- Gunakan `docker ps -a` untuk menampilkan semua container, termasuk yang telah berhenti.

*\$doceker ps*

- Menampilkan daftar container yang sedang berjalan.
- Gunakan `docker ps -a` untuk menampilkan semua container, termasuk yang telah berhenti.

*\$docker volume ls*

- Menampilkan daftar volume yang tersedia di mesin Docker Anda.
- Volume ini digunakan untuk menyimpan data yang persisten, bahkan setelah container dihentikan atau dihapus.

*\$docker system prune -a*

```
root@Hafid:~# docker system prune -a
WARNING! This will remove:
- all stopped containers
- all networks not used by at least one container
- all images without at least one container associated to them
- all build cache

Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
```

*\$docker run -it -v /home/hafid/localdata:/mydata k1 alpine*

```
root@Hafid:/home/hafid# docker run -it -v /home/hafid/localdata:/mydata --name k1 alpine
/ # cd /mydata
/mydata # touch f1 f2 f3
/mydata # ls -al
total 8
drwxr-xr-x  2 root    root    4096 Mar 13 10:25 .
drwxr-xr-x  1 root    root    4096 Mar 13 12:20 ..
-rw-r--r--  1 root    root      0 Mar 13 12:21 f1
-rw-r--r--  1 root    root      0 Mar 13 12:21 f2
-rw-r--r--  1 root    root      0 Mar 13 12:21 f3
/mydata #
root@Hafid:/home/hafid# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
6a3ca160e6ac   alpine    "/bin/sh" 2 hours ago Up 2 hours      k5
a656ae61d2dc   alpine    "/bin/sh" 2 hours ago Up 2 hours      k4
root@Hafid:/home/hafid# ls /home/hafid/localdata
f1 f2 f3
```

Menjalankan container berbasis image k1 dengan sistem operasi alpine.

```
a656ae61d2dc   alpine    "/bin/sh" 2 hours ago Up 2 hours      k4
root@Hafid:/home/hafid# ls /home/hafid/localdata
f1 f2 f3
root@Hafid:/home/hafid# docker start -i k1
/ # ls mydata
f1 f2 f3
```

*Kemudian buat star kembali kontainer k1 nya seperti diatas*

```
root@Hafid:/home/hafid# docker run -it -v /myfolder --name k2 alpine
/ # cd myfolder
/myfolder # touch f4 f5 f6
/myfolder #
```

Setelah itu buka terminal 2 untuk jalan kan perintah

*\$docker ps*

```
root@Hafid:/home/hafid# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
5b78fd0d82b0   alpine    "/bin/sh" 3 minutes ago    Up 3 minutes           k2
```

*\$docker volume ls*

```
root@Hafid:/home/hafid# docker volume ls
DRIVER      VOLUME NAME
local       6cccb614d9560f8ef82222baea75dabe62a1974301e5c2d92ec8c70b5aa8d358
local       7b72acbaaf4a9baf9b6b2bf08ed728ca3b6d40f098dd3a4e8bbd1a0e17f388e2
local       7bfc0e4d859cac9371bf30c098e5e30a2e55f33ec669fe1f979e0e593baad44
local       99bf394f63ab76d18957b26861bd39991f094e8349743770eeb050867f9bc5bd
local       ourdata
local       ourdate
```

*\$docker inspect k2*

```
root@Hafid:/home/hafid# docker inspect k2
[
  {
    "Id": "5b78fd0d82b0907c58a8654be5632f423bf687f4efeb0f9953e53aa4baf44a60",
    "Created": "2025-03-13T12:31:30.471144661Z",
    "Path": "/bin/sh",
    "Args": [],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 50596,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2025-03-13T12:31:30.762107861Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    }
  }
]
```

*\$docker volume inspect*

Asynchronous volume ini dapat diakses bersama dengan kontainer lain. Buka dari terminal 2, buat kontainer baru dengan nama k3 menggunakan volume dari kontainer k2 tersebut

Buka di terminal 2 seperti gambar dibawah

```
root@Hafid:/home/hafid# docker run -it --volumes-from k2 --name k3 alpine
/ # cd myfolder
/myfolder # echo "ini dari k3" > pesan
/myfolder #
```



Gambar di atas Menjalankan container k3 yang dapat mengakses semua data atau volume yang digunakan oleh container k2. Cocok untuk berbagi data antar- container.

```
/myfolder # touch f4 f5 f6
/myfolder # cat pesan
ini dari k3
/myfolder #
```

Ini tampilan pada terminal 1

*Ketik di terminal 1 \$echo "balasan dari k2 untuk k3" >> pesan*

```
/myfolder # cat pesan
ini dari k3
balasan dari k2 untuk k3
/myfolder #
```

## 2. Lab 3.2 multi kontainer dengan docker-compose

*\$mkdir -p \$HOME/wp-compose/{database,html}*

*\$cd \$HOME/wp-compose*

```
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
alpine        latest    aded1e1a5b37   3 weeks ago    7.83MB
root@Hafid:/home/hafid# mkdir -p $HOME/wp-compose/{database,html}
root@Hafid:/home/hafid# cd $HOME/wp-compose/
root@Hafid:~/wp-compose# docker-compose --version
docker-compose version 1.29.2, build unknown
root@Hafid:~/wp-compose# vi docker-compose.yml
root@Hafid:~/wp-compose# docker-compose up -d
```

*\$vi docker - compose.yml*

```
version: '3.3'

services:
  mariadb:
    image: mariadb
    volumes:
      - ./database:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress

  wordpress:
    depends_on:
      - mariadb
    image: wordpress:latest
    ports:
      - "8001:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: mariadb:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress
    volumes:
      - ./html:/var/www/html

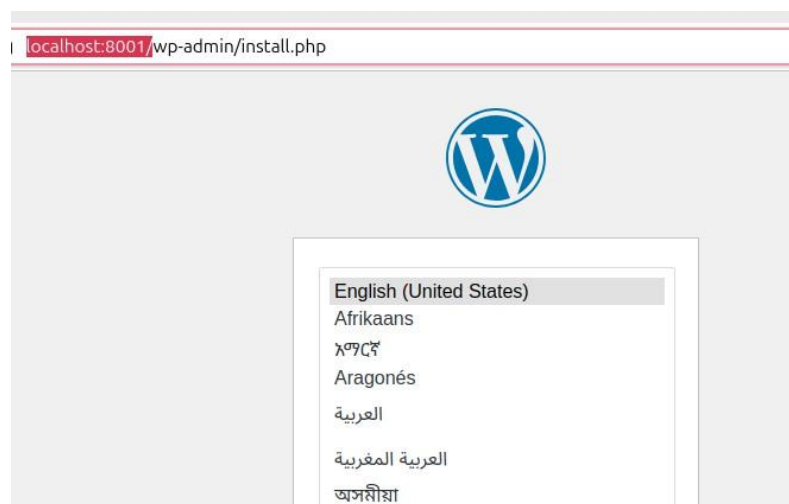
"docker-compose.yml" 29 lines, 602 bytes
```

Setelah dicopy tekan i untuk masuk mode insert kemudian paste, lalu ctrl + shift + : qw

Untuk save file

Kemudian \$docker - compose up -d Menjalankan semua layanan yang didefinisikan di docker-compose.yml secara otomatis dan berjalan di latar belakang.

Jalankan di browser <http://localhost:8001> untuk melihat wordpress yang telah dibuat



### BAB III

#### KESIMPULAN

##### A. Kesimpulan

Praktikum pada modul ini memberikan pemahaman mengenai penggunaan Docker Volume sebagai solusi untuk menyimpan data secara permanen. Volume memungkinkan data tetap tersimpan meskipun container dihentikan atau dihapus, serta dapat digunakan bersama oleh beberapa container untuk kebutuhan berbagi data.

Selain itu, praktikum juga membahas tentang Docker Compose yang mempermudah pengelolaan beberapa container sekaligus dalam satu sistem. Dengan menggunakan satu file konfigurasi `docker-compose.yml`, seluruh layanan dapat dijalankan secara otomatis dan terkoordinasi.

Melalui praktikum ini, diperoleh pemahaman lebih lanjut mengenai manajemen data dan pengelolaan layanan multi-container dalam Docker, yang merupakan salah satu keterampilan penting dalam pengembangan aplikasi modern dan deployment sistem secara efisien.