**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 : 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](#_Toc201360001)

[DAFTAR ISI ii](#_Toc201360002)

[BAB I 1](#_Toc201360003)

[1. Tujuan 1](#_Toc201360004)

[2. Dasar Teori 1](#_Toc201360005)

[3. Alat dan bahan 2](#_Toc201360006)

[BAB II 3](#_Toc201360007)

[A. Lab 3 Docker 3](#_Toc201360008)

[1. Lab 3.1 : Docker Volume 3](#_Toc201360009)

[2. Lab 3.2 multi kontainer dengan docker-compose 6](#_Toc201360010)

[BAB III 8](#_Toc201360011)

[A. Kesimpulan 8](#_Toc201360012)

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**.

1. 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.

1. Alat dan bahan
2. 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.
3. 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 3 Docker
2. 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*

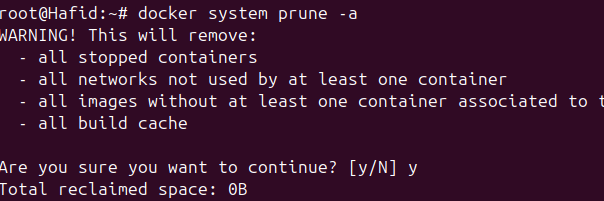
* + Menampilkan daftar container yang sedang berjalan.
  + Gunakan docker ps -a untuk menampilkan semua container, termasuk yang telah berhenti.

*$doceker ps*

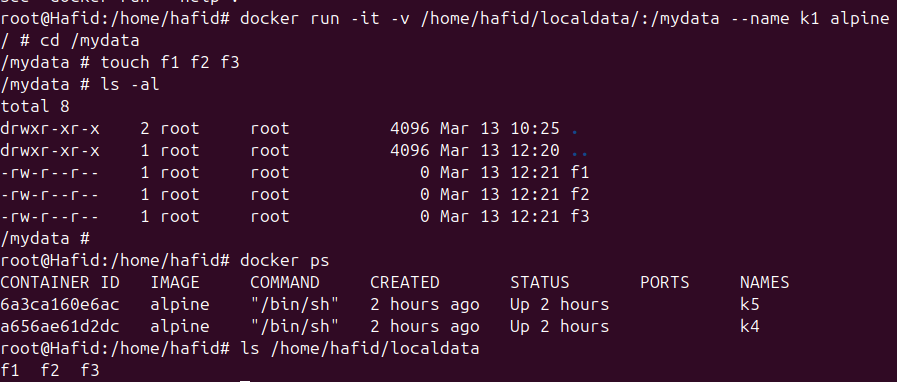
* + Menampilkan daftar container yang sedang berjalan.
  + Gunakan docker ps -a untuk menampilkan semua container, termasuk yang telah berhenti.

*$docker volume ls*

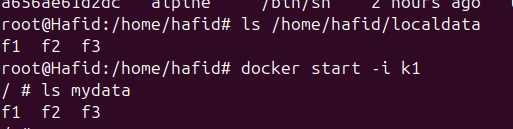
* + Menampilkan daftar volume yang tersedia di mesin Docker Anda.
  + Volume ini digunakan untuk menyimpan data yang persisten, bahkan setelah container dihentikan atau dihapus.

*$docker system prune –a*

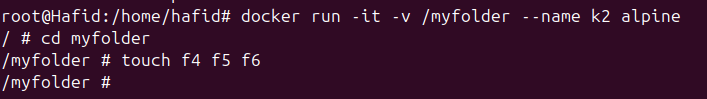
*$docker run -it -v /home/hafid/localdata/:/mydata k1 alpine*

**

Menjalankan container berbasis image k1 dengan sistem operasi alpine.

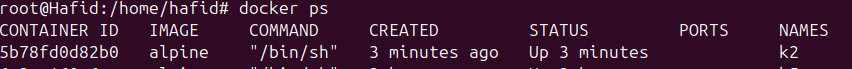


*Kemudian buat star kembali kontainer k1 nya seperti diatas*

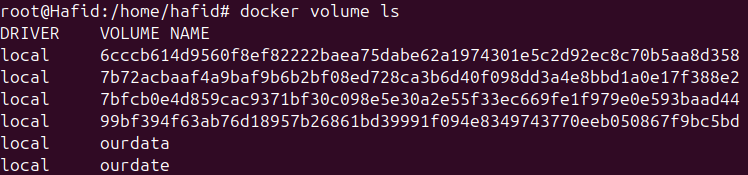


Setelah itu buka terminal 2 untuk jalan kan perintah

*$docker ps*

**

*$docker volume ls*

**

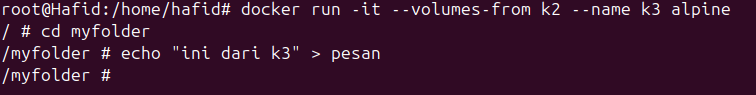
*$docker inspect k2*



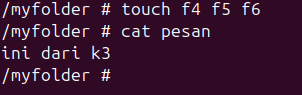
*$docker volume inspect*

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

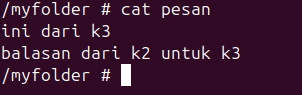


Gambar di atas Menjalankan container k3 yang dapat mengakses semua data atau volume yang digunakan oleh container k2. Cocok untuk berbagi data antar- container.



Ini tampilan pada terminal 1

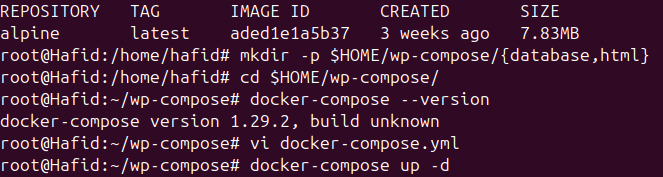
*Ketik di terminal 1 $echo “balasan dari k2 untuk k3” >> pesan*

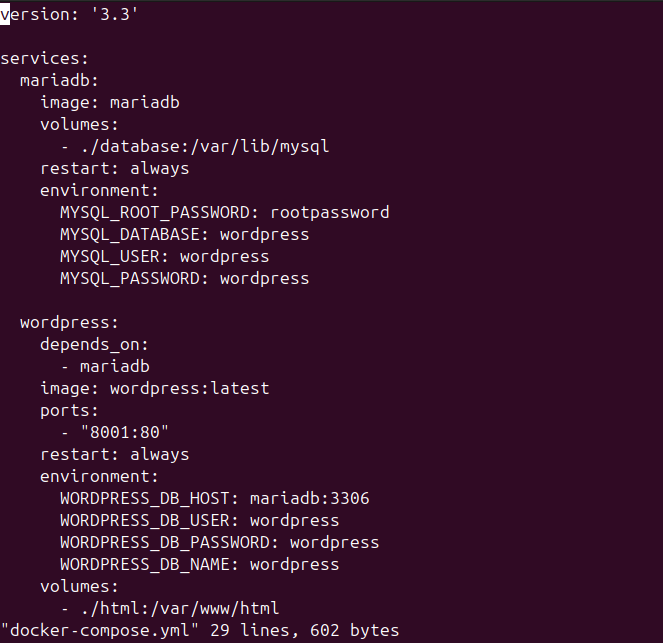


1. Lab 3.2 multi kontainer dengan docker-compose

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

*$cd $HOME/wp-compose*



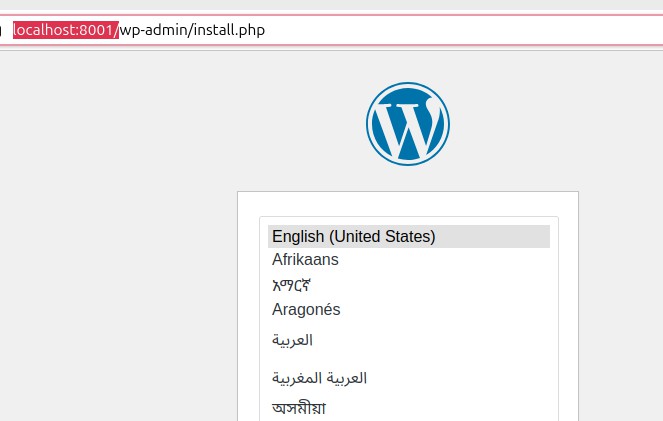
*$vi docker - compose.yml*

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](http://localhost:8001/) untuk melihat wordpress yang telah dibuat



# BAB III

KESIMPULAN

1. 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.