# LAPORAN PROGRES 3

**Pembuatan Dockerfile + image**

Disusun untuk memenuhi Tugas Akhir Mata Kuliah Sistem Operasi

Dosen Pengampu:

Ferdi Chahyadi, S.Kom., M.Cs


Disusun oleh:


| | |
|---|---|
| Akbar Risky Lingga | 2401020003 |
| Dzaky Ribal Faiz | 2401020035 |
| Muhammad Al-Fikry Akbar | 2401020031 |
| Al Adhlu sodri niwrad | 2401020015 |

**PRODI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK DAN TEKNOLOGI KEMARITIMAN**

**UNIVERSITAS MARITIM RAJA ALI HAJI**


**2025/2026**

# BAB I

# PENDAHULUAN

## 1.1 Latar Belakang Progres

Laporan ini merupakan dokumentasi progres kedua dari proyek penerapan teknologi container menggunakan Docker pada mata kuliah Sistem Operasi. Setelah proposal proyek diselesaikan pada minggu sebelumnya, kegiatan pada progres ini difokuskan pada pembuatan Dockerfile dan Docker image sesuai dengan rencana yang telah disusun.

Pada tahap ini, aplikasi dikemas ke dalam image Docker agar dapat dijalankan secara terisolasi dan konsisten. Setiap proses pembuatan image didokumentasikan melalui screenshot dan disertai penjelasan singkat untuk memudahkan pemahaman, evaluasi, serta menjadi dasar pada tahap implementasi pembatasan resource pada progres selanjutnya.

## 1.2 Tujuan Progres 3

- Membuat Dockerfile untuk aplikasi yang digunakan.
- Membangun Docker image berdasarkan Dockerfile yang telah dibuat.
- Memastikan aplikasi dapat dijalankan di dalam container Docker.
- Memahami proses pengemasan aplikasi menggunakan teknologi container.
- Menyiapkan lingkungan aplikasi untuk tahap pembatasan resource pada progres selanjutnya.

## 1.3 Ruang Lingkup Progres 3

Ruang lingkup pada Progres 2 dibatasi pada tahapan pembuatan Dockerfile dan proses build Docker image dari aplikasi yang telah disiapkan. Progres ini berfokus pada pengemasan aplikasi ke dalam container serta memastikan image berhasil dibuat dan dapat dijalankan dengan baik. Seluruh proses didokumentasikan sebagai bagian dari laporan.

Adapun ruang lingkup Progres 2 meliputi:

- Penyusunan Dockerfile sesuai kebutuhan aplikasi.
- Pengelolaan dependensi aplikasi menggunakan file pendukung.
- Proses build Docker image menggunakan perintah Docker.

- Pengujian hasil build image untuk memastikan aplikasi dapat berjalan.
- Dokumentasi proses melalui screenshot dan penjelasan singkat.

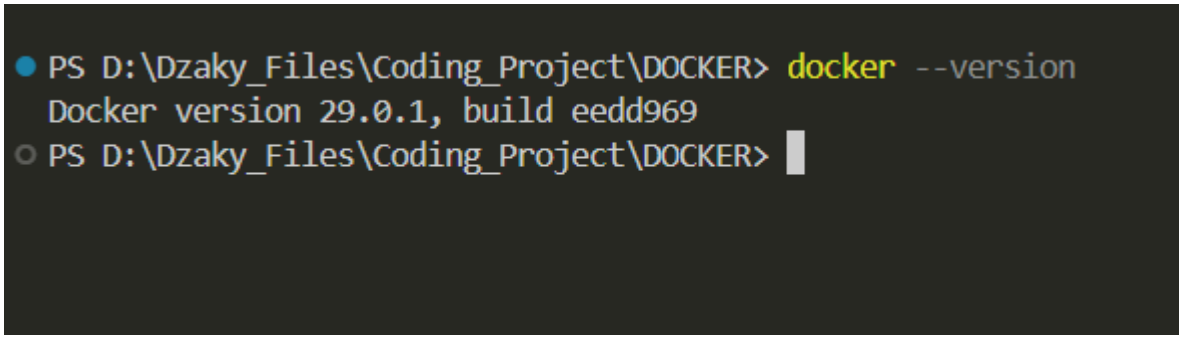
Pembatasan resource CPU dan memori menggunakan cgroups tidak termasuk dalam ruang lingkup Progres 2 dan akan dibahas pada progres selanjutnya.

# BAB II

# Pembahasan

## 2.1 Lingkungan kerja

Pada tahap ini dilakukan persiapan lingkungan kerja dengan memastikan Docker telah terpasang dan dapat dijalankan pada sistem operasi yang digunakan. Pemeriksaan dilakukan menggunakan perintah docker --version untuk memastikan Docker berjalan dengan baik. Tahap ini penting agar proses pembuatan Docker image dapat dilakukan tanpa kendala.




Berdasarkan hasil pengecekan, diperoleh informasi bahwa Docker telah terinstal dengan versi 29.0.1 (build eedd969).

Hasil tersebut menunjukkan bahwa Docker sudah siap digunakan sehingga proses pembuatan Dockerfile dan Docker image pada Progres 2 dapat dilaksanakan tanpa kendala terkait instalasi lingkungan kerja.

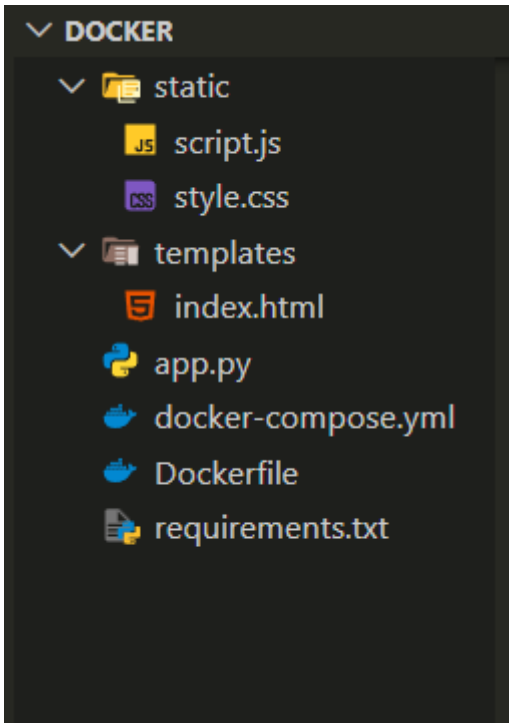
## 2.2 Struktur Projek

Struktur project pada Progres 2 disusun dalam satu folder utama bernama DOCKER. Di dalam folder ini terdapat beberapa file dan folder pendukung yang digunakan untuk menjalankan aplikasi berbasis Flask di dalam container Docker. Penyusunan struktur ini bertujuan agar file project lebih rapi dan mudah dikelola.

Folder DOCKER berisi file app.py sebagai source code utama aplikasi, Dockerfile sebagai konfigurasi pembuatan Docker image, docker-compose.yml untuk menjalankan container aplikasi dan database, serta requirements.txt yang berisi daftar dependensi aplikasi. Selain itu, terdapat folder static yang berisi file style.css untuk pengaturan tampilan aplikasi, dan folder templates yang berisi file index.html sebagai tampilan antarmuka utama aplikasi.

Dengan struktur tersebut, proses build Docker image dan pengelolaan aplikasi dapat dilakukan dengan lebih terorganisir dan efisien.

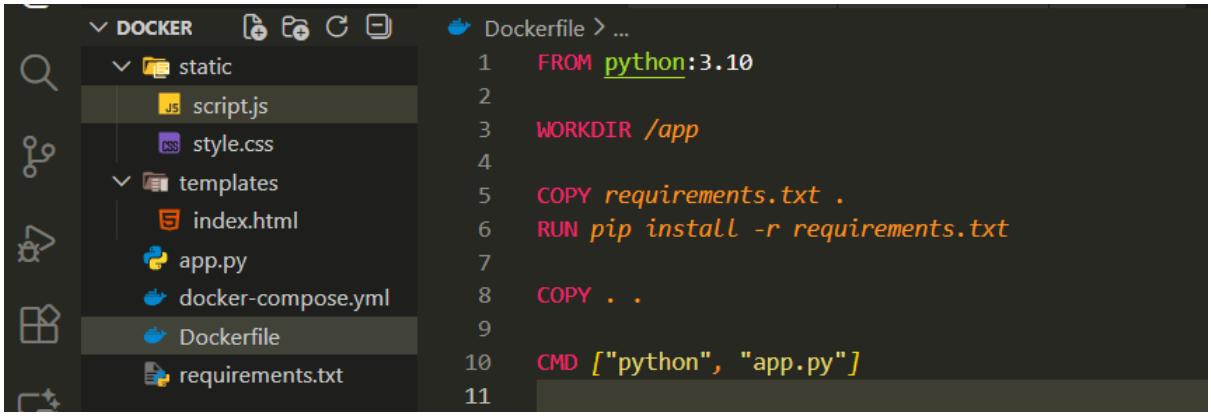
Berikut ini merupakan Struktur projek yang telah dibentuk :




## 2.3 Pembuatan dockerfile

Dockerfile dibuat sebagai panduan untuk membangun Docker image aplikasi. Pada Dockerfile ditentukan base image Python, direktori kerja di dalam container, proses instalasi dependensi aplikasi, serta perintah untuk menjalankan aplikasi Flask. Dengan adanya Dockerfile, aplikasi dapat dikemas dan dijalankan secara konsisten di dalam container.

Berikut merupakan isi dari dockerfile yang telah dibangun :




Penjelasan :

*"FROM python:3.10"*

Baris ini digunakan untuk menentukan base image yang digunakan. Image python:3.10 dipilih karena aplikasi dikembangkan menggunakan bahasa Python dan membutuhkan lingkungan Python untuk dapat dijalankan.

*"WORKDIR /app"*

Perintah ini digunakan untuk menentukan direktori kerja di dalam container. Semua perintah berikutnya akan dijalankan di dalam folder /app.

*"COPY requirements.txt ."*

Perintah ini berfungsi untuk menyalin file requirements.txt dari sistem host ke dalam container. File ini berisi daftar library yang dibutuhkan oleh aplikasi.

*"RUN pip install -r requirements.txt"*

Perintah ini digunakan untuk menginstal seluruh dependensi aplikasi yang tercantum pada file requirements.txt menggunakan pip.

*"COPY . ."*

Perintah ini menyalin seluruh file dan folder project, termasuk app.py, folder static, dan folder templates, ke dalam direktori kerja container.

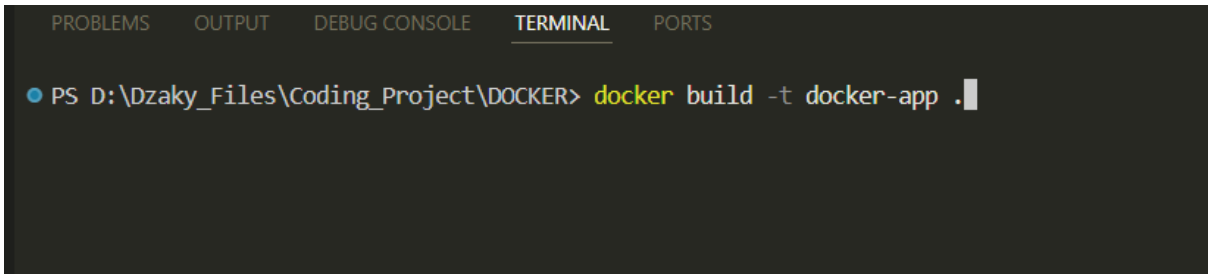
*"CMD ["python", "app.py"]"*

Perintah ini digunakan untuk menjalankan aplikasi ketika container dijalankan. Docker akan mengeksekusi file app.py menggunakan Python secara otomatis saat container aktif.

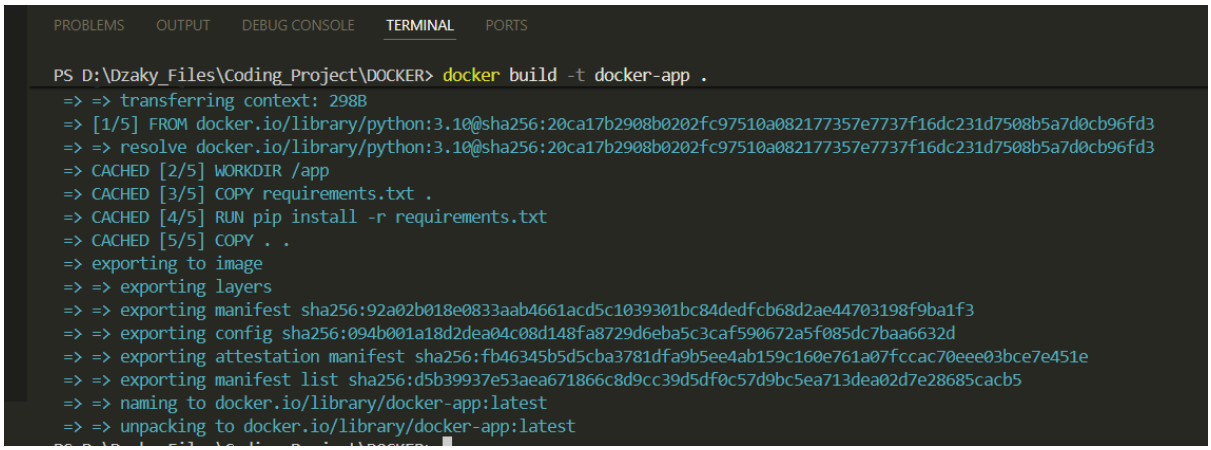
## 2.4 Proses build Docker Image

Setelah Dockerfile dibuat, proses build image dilakukan menggunakan perintah docker build. Pada tahap ini Docker membaca setiap instruksi dalam Dockerfile dan membangun image secara bertahap. Jika proses build berhasil tanpa error, maka Docker image siap digunakan untuk menjalankan container.

Perintah yang digunakan untuk melakukan build Docker image adalah docker build. Pada perintah ini, Docker akan membaca setiap baris instruksi dalam Dockerfile secara berurutan. Instruksi dimulai dari penggunaan base image Python, penentuan direktori kerja, penyalinan file requirements.txt, instalasi dependensi aplikasi, hingga penyalinan seluruh source code aplikasi ke dalam image. Setiap instruksi akan dijalankan sebagai layer terpisah, sehingga jika terjadi perubahan pada salah satu bagian, Docker hanya akan membangun ulang layer yang terdampak.




Apabila seluruh proses berjalan tanpa error, Docker akan menampilkan pesan bahwa build image telah berhasil. Image yang dihasilkan kemudian disimpan di sistem dan dapat digunakan untuk menjalankan container aplikasi. Keberhasilan proses build ini menjadi indikator bahwa Dockerfile telah dikonfigurasi dengan benar dan aplikasi siap dijalankan di dalam container.



## 2.5 Hasil build Docker image

Keberhasilan proses build diverifikasi menggunakan perintah docker images. Dari hasil perintah tersebut terlihat bahwa Docker image aplikasi telah berhasil dibuat dan tersimpan di sistem. Image ini akan digunakan pada tahap menjalankan container.

```
PS D:\Dzaky_Files\Coding_Project\DOCKER> docker images

IMAGE                ID             DISK USAGE   CONTENT SIZE   EXTRA
docker-app:latest    d5b39937e53a       1.8GB          479MB
mysql:8              5cdee9be17b6      1.07GB          233MB
python:3.10          20ca17b2908b      1.58GB          406MB
PS D:\Dzaky_Files\Coding_Project\DOCKER>
```

Berdasarkan hasil perintah docker images, terdapat beberapa Docker image yang tersimpan di sistem. Image docker-app merupakan image utama yang dibangun dari Dockerfile pada Progres 2 dan digunakan untuk menjalankan aplikasi Flask.

Image python:3.10 berfungsi sebagai base image yang digunakan dalam proses pembuatan image docker-app, sedangkan image mysql:8 digunakan sebagai layanan database pendukung aplikasi dan berasal dari image resmi Docker.

# BAB III

# PENUTUP

## 3.1 Kesimpulan

Berdasarkan hasil pembahasan pada bab ini, dapat disimpulkan bahwa proses pembuatan Dockerfile dan build Docker image telah berhasil dilakukan sesuai dengan tujuan Progres 2. Dockerfile berhasil disusun untuk mengemas aplikasi berbasis Flask beserta seluruh dependensinya ke dalam sebuah Docker image.

Docker image yang dihasilkan dapat dibangun tanpa kendala dan tersimpan dengan baik di sistem. Dengan demikian, aplikasi telah siap digunakan untuk menjalankan container pada tahap selanjutnya, yaitu implementasi pembatasan resource dan analisis performa

## 3.2 Bukti Kerja


msi
CYBORG 15 A12U