**Tutorial Membangun Docker Image untuk Aplikasi Web Flask**

1. **Menyiapkan File .ipynb**

Sebelum memulai, pastikan sudah memiliki file .ipynb yang berisi kode untuk aplikasi web Flask yang ingin dikemas dalam Docker. File ini akan menjadi basis aplikasi yang akan kita kontainerisasi.

link file : [klik disini](https://colab.research.google.com/drive/1mqRNJLmuamJGbtcB2HBsJzH-8mgqledy?usp=drive_link)

1. **Membuat Virtual Environment**
2. Buka Visual Studio Code (VSCode).
3. Tekan Ctrl+Shift+P untuk membuka Command Palette.
4. Cari perintah "Python: Create Environment" dan pilihnya.
5. Buka terminal dan jalankan perintah berikut untuk membuat virtual environment:

python -m venv nama\_env

1. **Membuat Aplikasi Flask**

* Membuat file app.py

link file : [klik disini](https://drive.google.com/file/d/1JM_hGaA9CxvZB6-FVZCDd7uQly-wPPK0/view?usp=drive_link)

* Membuat file index.html

link file : [klik disini](https://drive.google.com/file/d/1rBy6kEV1TF1a6GbymCE2BBSPE-_xcu61/view?usp=drive_link)

1. **Membuat File requirements.txt**

Untuk mengatur dependensi aplikasi, kita akan membuat file requirements.txt. File ini akan berisi daftar semua pustaka Python yang dibutuhkan oleh aplikasi kita. Berikut adalah contoh isi dari requirements.txt

Contoh :

Flask==2.3.3

pandas==1.5.3

scikit-learn==1.2.2

gunicorn==20.1.0

Pastikan semua pustaka yang anda butuhkan terdaftar dalam file ini beserta versi yang sesuai.

1. **Membuat Dockerfile**

Sekarang, mari buat Dockerfile untuk mengemas aplikasi Flask dan dependensinya

ke dalam image Docker. Berikut adalah isi dari Dockerfile:

FROM python:3.9-slim

EXPOSE 5000

WORKDIR /app

COPY . /app

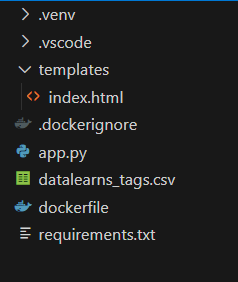
COPY requirements.txt .

RUN pip install -r requirements.txt

CMD ["gunicorn", "--bind", "0.0.0.0:5002", "app:app"]

Dalam Dockerfile ini, kita menggunakan image Python 3.9 , mengekspos port 5000, mengatur direktori kerja, menyalin kode aplikasi, menginstal dependensi, dan menjalankan aplikasi menggunakan Gunicorn.

Pastikan struktur file nya seperti dibawah ini.



1. **Membangun Docker Image**

Sekarang, kita akan membangun Docker Image dengan perintah berikut di direktori yang berisi Dockerfile:

docker build -t name\_image .

Pastikan untuk menggantikan nama\_image dengan nama yang sesuai untuk image anda.

1. **Run docker container**

Akhirnya, kita dapat menjalankan Docker container dari Image yang telah kita buat dengan perintah berikut:

docker run -dp 8083:5000 -ti --name nama\_container nama\_image

Dalam perintah ini, kita mengalihkan port dari port 5000 ke port 8083 pada container, memberikan nama container, dan menjalankannya dalam mode interaktif.

Referensi :

Docker Inc. (2023, 9 25). *Packaging your software*. Retrieved from Docker: https://docs.docker.com/build/building/packaging/

Microsoft. (2023, 9 25). *Python environments in VS Code*. Retrieved from Visual Studio Code: https://code.visualstudio.com/docs/python/environments