

1. Membuat Dockerfile

```
Dockerfile
1 FROM python:3.8-slim
2 RUN apt-get update && apt-get install -y make && rm -rf /var/lib/apt/lists/*
3 COPY . /app
4 WORKDIR /app
5 RUN python -m venv venv && \
6     . venv/bin/activate && \
7     pip install --no-cache-dir -r requirements.txt
8 CMD ["make", "all"]
```

Kode di atas digunakan untuk membangun environment berbasis Python dengan menggunakan docker. Kode dockerfile ini mengambil image Python versi 3.8 slim sebagai dasar container. Instruksi RUN pada kode ini berguna untuk memperbarui daftar paket dari repository dan menginstal program make. Selain itu, instruksi ini juga akan membersihkan cache agar image lebih ringan. Instruksi COPY berguna untuk menyalin semua file dari direktori lokal yang sama dengan Dockerfile ke direktori /app di dalam container.

Instruksi WORKDIR di sini akan mengatur direktori kerja aktif di dalam container ke /app, sehingga perintah yang dijalankan setelahnya akan dilakukan di direktori /app. Dilakukan kembali instruksi RUN dengan perintah yang berbeda, yang kali ini berfungsi untuk membuat venv Python bernama venv di dalam container, mengaktifkan venv yang dibuat, dan menginstal semua library yang tercantum di file requirements.txt. Instruksi CMD di sini berguna untuk menjalankan perintah make all yang menjalankan otomatisasi sesuai definisi dalam Makefile.

2. Output

```
PS C:\Users\lyona\Documents\MLOps\docker_MLOps> docker build -t docker-mlops .
[+] Building 303.9s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 606B
=> [internal] load metadata for docker.io/library/python:3.8-slim
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/python:3.8-slim@sha256:1d52838af602b4b5a831beb13a0e4d073280665ea7be7f69ce2382f29c5a613f
=> [internal] load build context
=> => transferring context: 23.16MB
=> CACHED [2/5] RUN apt-get update && apt-get install -y make && rm -rf /var/lib/apt/lists/*
=> [3/5] COPY . /app
=> [4/5] WORKDIR /app
```

Perintah yang dilakukan pada gambar di atas berfungsi untuk membangun image dari docker file dengan nama docker-mlops.

```
=> [5/5] RUN python -m venv venv && . venv/bin/activate && pip install --no-cache-dir -r requirements.txt
=> exporting to image
=> => exporting layers
=> => writing image sha256:e639e6466e079b47bfab6a34690ad4289051e0d4becc4b2ed746010171f2ded
=> => naming to docker.io/library/docker-mlops

What's next:
View a summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Users\lyona\Documents\MLOps\docker_MLOps> docker run -it docker-mlops
make[1]: Entering directory '/app'
make[1]: Leaving directory '/app'
Best max_depth: 3, Best cross-validated accuracy: 0.9818181818181818
Model saved to models/decision_tree_model.pkl
Model loaded from models/decision_tree_model.pkl
Evaluation metrics saved to results/evaluation_metrics.json
{'accuracy': 0.9310344827586207, 'precision': 0.9416445623342174, 'recall': 0.9310344827586207, 'f1_score': 0.9295977011494252}
PS C:\Users\lyona\Documents\MLOps\docker_MLOps>
```

Perintah di atas berguna untuk menjalankan container dari image docker-mlops. -it digunakan agar hasil eksekusi dapat ditampilkan langsung di terminal.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e2ec5036d4bc	docker-mlops	"make all"	32 seconds ago	Exited (0) 7 seconds ago		elegant_dhawan

Dapat dilihat pada gambar di atas bahwa container berhasil dibuat dan berhasil di eksekusi 7 detik yang lalu.

```
PS C:\Users\lyona\Documents\MLOps\Docker_MLOps> docker run -d -p 5000:5000 docker-mlops  
5b6f137dc913807e7e576c27f0bb102ec8a7b21bae6a5f0f00b7ff279b033687
```

Terdapat cara lain untuk menjalankan docker yaitu dengan memetakan port container ke port host (5000:5000). Namun, karena proyek yang dieksekusi hanya memunculkan 1x output saja, jadi container akan langsung berhenti ketika sudah selesai mengeksekusi dan mengeluarkan output yang sesuai.

3. Repository GitHub

Berikut merupakan link repository GitHub:

https://github.com/lyonardgemilang/MLOps/tree/main/Docker_MLOps