

LAPORAN PRAKTIKUM
TEKNOLOGI CLOUD
PERTEMUAN KE – 11



Disusun Oleh :

NAMA	: TARISA DWI SEPTIA
NIM	: 205410126
JURUSAN	: TEKNIK INFORMATIKA
JENJANG	: S1

UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA
YOGYAKARTA
2020

DOCKER COMPOSE

A. Tujuan

- Mahasiswa mengerti tentang kegunaan docker compose
- Mahasiswa dapat menggunakan docker compose sesuai dengan kebutuhan

B. Dasar Teori

Compose adalah salah satu tool untuk mendefinisikan dan menjalankan aplikasi Docker multi-container. Dengan Compose, untuk menjalankan menggunakan file Compose untuk mengonfigurasi layanan aplikasi. Kemudian, dengan menggunakan satu perintah, dapat membuat dan memulai semua layanan dari suatu konfigurasi.

C. Praktik

1. Compose adalah salah satu tool untuk mendefinisikan dan menjalankan aplikasi Docker multi-container. Dengan Compose, untuk menjalankan menggunakan file Compose untuk mengonfigurasi layanan aplikasi. Kemudian, dengan menggunakan satu perintah, dapat membuat dan memulai semua layanan dari suatu konfigurasi.

```
C:\Users\hp>docker-compose -v
docker-compose version 1.29.2, build 5becea4c
```

2. Buatlah direktori dengan nama “docker-compose”. Dalam direktori “dockercompose” buat direktori dengan “product”. Pada direktori “product” buat dokumen dengan nama file “api.py”. Pada dokumen “api.py” ketikkan kode-kode seperti gambar dibawah ini

```
# Product Service

# Import Framework

from flask import Flask
from flask_restful import Resource, Api

# Instantiate the app
app = Flask(__name__)
api = Api(app)

class Product(Resource):
    def get(self):
        return{
            'products' : ['Ice creame', 'Chocolate', 'Fruit', 'Eggs']
        }

# Create routes
api.add_resource(Product, '/')

# Run the aplication
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=80, debug = True)
```




3. Setelah itu buatlah dokumen dengan nama file “Dockerfile” dan isikan kode-kode dibawah ini ke dalam “Dockerfile”.

```
FROM python : 3-onbuild
COPY . /usr/src/app
CMD ["python", "api.py"]
```

4. Setelah itu buatlah dokumen dengan nama file “requirements.txt” dan isikan kodekode dibawah ini kedalam “requirements.txt”.

```
Fask == 0.12
flask-restful == 0.3.5
```

5. Maka pada direktori “product” akan ada file-file seperti pada contoh dibawah ini



 api.py	12/9/2021 2:27 MALAM	Python Source File	1 KB
 Dockerfile	12/9/2021 2:33 MALAM	File	1 KB
 requirements	12/9/2021 2:36 MALAM	File	1 KB

6. Setelah itu buatlah dokumen dengan nama file “docker-compose.yml” pada direktori “docker-compose”. Isikan kode-kode seperti dibawah ini

```
version : '3'

services :
  product-services:
    build: ./product
    volumes :
      - ./product:/usr/src/app
    ports :
      - 5001:80
```

7. Maka file yang ada di dalam direktori “docker-compose” seperti contoh dibawah ini

 product	12/9/2021 2:36 MALAM	File folder	
 docker-compose.yml	12/9/2021 2:38 MALAM	Yaml Source File	0 KB

8. Buka kembali command prompt dan pastikan posisi direktori berada di dalam direktori “docker-compose” seperti contoh dibawah ini

```
C:\Users\hp>cd docker-compose
C:\Users\hp\docker-compose>
```

9. Kemudian ketikkan perintah “docker-compose up” untuk memulai docker compose seperti contoh dibawah ini

```
MINGW64/e/docker-compose
ip@DESKTOP-U90GI9G MINGW64 /e/docker-compose:
$ docker-compose up
Creating network "docker-compose_default" with the default driver
Building product-service
#2 [internal] load build definition from Dockerfile
#2 sha256:779cfb48af6f3314bd04c959f0ece2bb611dd0837d81a777f198873ed7a1da8b
#2 DONE 1.4s

#1 [internal] load .dockerignore
#1 sha256:90901532ec7bf56ea9fb996a75ebe48b730e8de34c6220247c56ce2120f0835d
#1 DONE 1.4s

#1 [internal] load .dockerignore
#1 sha256:90901532ec7bf56ea9fb996a75ebe48b730e8de34c6220247c56ce2120f0835d
#1 ...

#2 [internal] load build definition from Dockerfile
#2 sha256:779cfb48af6f3314bd04c959f0ece2bb611dd0837d81a777f198873ed7a1da8b
#2 transferring dockerfile:
#2 ...

#1 [internal] load .dockerignore
#1 sha256:90901532ec7bf56ea9fb996a75ebe48b730e8de34c6220247c56ce2120f0835d
#1 transferring context:
#1 transferring context: 2B 3.5s done
#1 ...

#2 [internal] load build definition from Dockerfile
#2 sha256:779cfb48af6f3314bd04c959f0ece2bb611dd0837d81a777f198873ed7a1da8b
#2 transferring dockerfile: 2B 3.1s done
#2 DONE 28.2s

#1 [internal] load .dockerignore
#1 sha256:90901532ec7bf56ea9fb996a75ebe48b730e8de34c6220247c56ce2120f0835d
#1 DONE 30.3s
failed to solve with frontend dockerfile.v0: failed to read dockerfile: open /var/lib/docker/tmp/buildkit-mount821619637/Dockerfile: no such file or directory
Service 'product-service' failed to build : Build failed
```

Setelah saya enter, terdapat kesalahan berupa “ failed to solve with frontend dockerfile.v0: failed to read dockerfile: open /var/lib/docker/tmp/buildkit-mount821619637/Dockerfile: no such file or directory
Service 'product-service' failed to build : Build failed “

D. Kesimpulan

Setelah melakukan praktik diatas, dapat disimpulkan bahwa mahasiswa tidak dapat mengerti tentang kegunaan docker compose dan juga mahasiwa tidak dapat menggunakan docker compose sesuai dengan kebutuhan