

DOCKER COMPOSE

A. TUJUAN

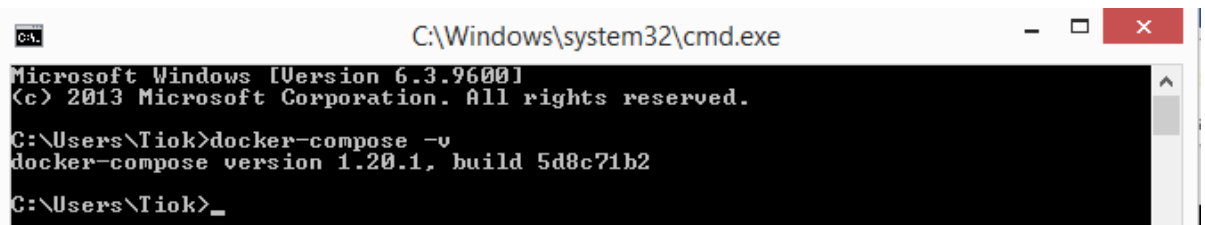
- Mahasiswa mengerti tentang kegunaan docker compose
- Mahasiswa dapat menggunakan docker compose sesuai dengan kebutuhan

B. TEORI SINGKAT

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

Sebelum memulai praktikum, kita perlu memastikan apakah docker compose sudah terinstall di komputer apa belum. Cara mengetahui yaitu dengan cara buka aplikasi command prompt dan ketikkan perintah “docker-compose -v” seperti contoh dibawah ini

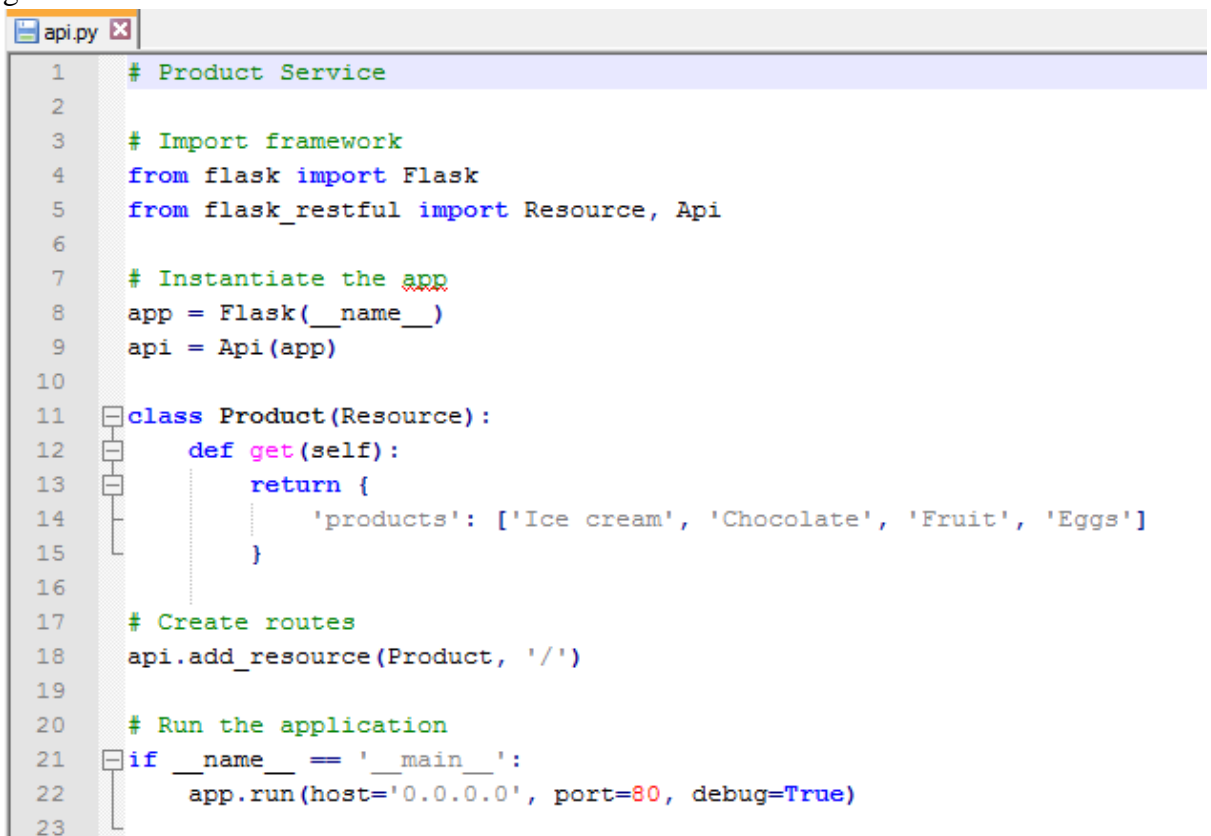


```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Tiok>docker-compose -v
docker-compose version 1.20.1, build 5d8c71b2

C:\Users\Tiok>_
```

Buatlah direktori dengan nama “docker-compose”. Dalam direktori “docker-compose” 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



```
api.py
1  # Product Service
2
3  # Import framework
4  from flask import Flask
5  from flask_restful import Resource, Api
6
7  # Instantiate the app
8  app = Flask(__name__)
9  api = Api(app)
10
11 class Product(Resource):
12     def get(self):
13         return {
14             'products': ['Ice cream', 'Chocolate', 'Fruit', 'Eggs']
15         }
16
17 # Create routes
18 api.add_resource(Product, '/')
19
20 # Run the application
21 if __name__ == '__main__':
22     app.run(host='0.0.0.0', port=80, debug=True)
23
```

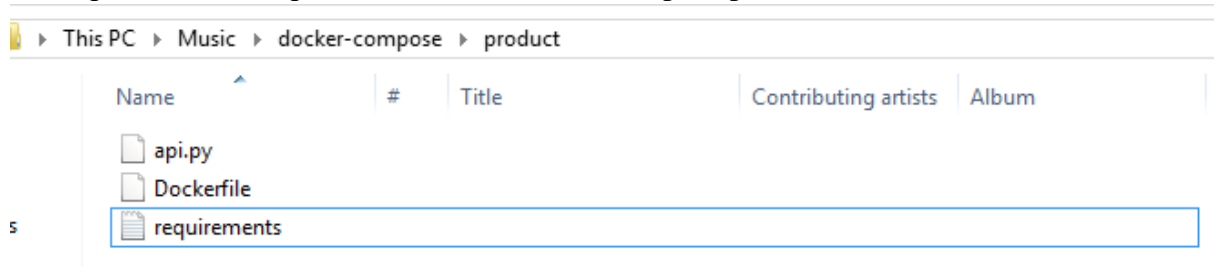
Setelah itu buatlah dokumen dengan nama file “Dockerfile” dan isikan kode-kode dibawah ini ke dalam “Dockerfile”.

```
api.py x Dockerfile x
1 FROM python:3-onbuild
2 COPY . /usr/src/app
3 CMD ["python", "api.py"]
4
```

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

```
api.py x Dockerfile x requirements.txt x
1 Flask==0.12
2 flask-restful==0.3.5
3
```

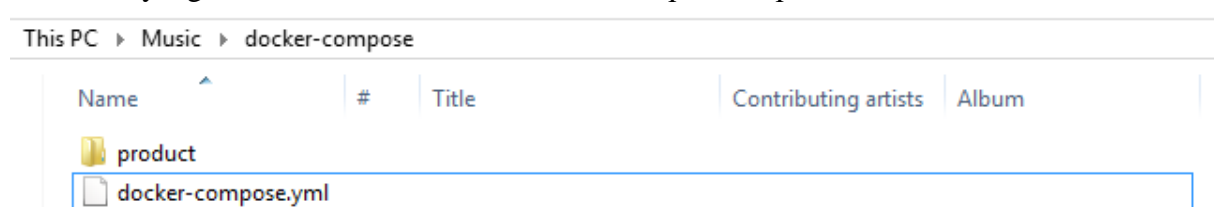
Maka pada direktori “product” akan ada file-file seperti pada contoh dibawah ini



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

```
api.py x Dockerfile x requirements.txt x docker-compose.yml x
1 version: '3'
2
3 services:
4   product-service:
5     build: ./product
6     volumes:
7       - ./product:/usr/src/app
8     ports:
9       - 5001:80
```

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



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

```
C:\Windows\system32\cmd.exe

C:\Users\Tiok\Music>cd docker-compose
C:\Users\Tiok\Music\docker-compose>_
```

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

```
C:\Windows\system32\cmd.exe

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Tiok>cd Music\docker-compose
C:\Users\Tiok\Music\docker-compose>docker-compose up_
```

Selanjutnya yaitu proses download dan tunggu hingga selesai

```
C:\Windows\system32\cmd.exe

1c7fe136a31e: Downloading [==>]
 3.218MB/54.25MBwloading [=====>]
ece825d3308b: Downloading [=====>]
ece825d3308b: Downloading [=====>]
 4.5MB/17.59MB
ece825d3308b: Downloading [=====>]
ece825d3308b: Downloading [=====>]
1c7fe136a31e: Downloading [==>]
 3.751MB/54.25MBwloading [=====>]
ece825d3308b: Downloading [=====>]
ece825d3308b: Downloading [=====>]
ece825d3308b: Downloading [=====>]
 5.414MB/17.59MB
ece825d3308b: Downloading [=====>]
1c7fe136a31e: Downloading [==>]
 4.291MB/54.25MB
ece825d3308b: Downloading [=====>]
 5.954MB/17.59MB
ece825d3308b: Downloading [=====>]
ece825d3308b: Downloading [=====>]
1c7fe136a31e: Downloading [==>]
 4.832MB/54.25MB
ece825d3308b: Downloading [=====>]
 6.679MB/17.59MB
ece825d3308b: Downloading [=====>]
```


Jika sudah selesai kita cek dengan membuka browser dan ketikkan “192.168.99.100:5001” pada url browser maka akan tampak seperti contoh dibawah ini



Pada langkah diatas yaitu menggunakan docker compose tetapi hanya membuat 1 image. Untuk selanjutnya yaitu kita coba menggunakan docker compose untuk membuat 2 image dan 2 image tersebut bisa saling terintegrasi. Sebelum memulai pastikan menghentikan docker compose yang sedang berjalan dengan cara mengetikkan perintah “docker-compose stop” seperti contoh dibawah ini

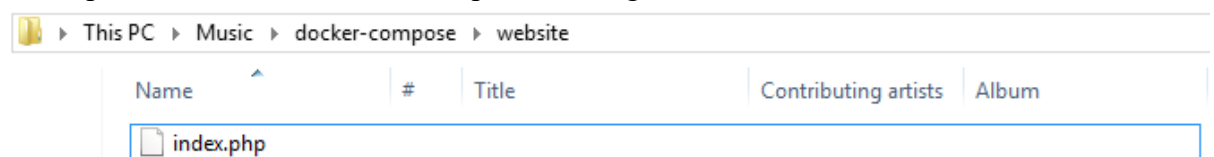
```
C:\Users\Tiok\Music\docker-compose>docker-compose stop
Stopping dockercompose_product-service_1 ... done
```

Masih menggunakan file-file yang terdapat pada direktori “docker-compose”. Tambahkan direktori “website” dan buat file dengan nama file “index.php”. Isikan kode-kode berikut pada file “index.php”.



```
1 <html>
2   <head>
3     <title>My Shop</title>
4   </head>
5
6   <body>
7     <h1>Welcome to my shop</h1>
8     <ul>
9       <?php
10
11         $json = file_get_contents('http://product-service/');
12         $obj = json_decode($json);
13
14         $products = $obj->products;
15
16         foreach ($products as $product) {
17           echo "<li>$product</li>";
18         }
19
20       ?>
21     </ul>
22   </body>
23 </html>
24
```

Maka pada direktori “website” terdapat file sebagai berikut



Setelah itu buka kembali file “docker-compose.yml” pada direktori “docker-compose”. Setelah itu tambahkan beberapa kode-kode seperti contoh dibawah ini

```
api.py x Dockerfile x requirements.txt x docker-compose.yml x
1 version: '3'
2
3 services:
4   product-service:
5     build: ./product
6     volumes:
7       - ./product:/usr/src/app
8     ports:
9       - 5001:80
10
11   website:
12     image: php:apache
13     volumes:
14       - ./website:/var/www/html
15     ports:
16       - 5000:80
17     depends_on:
18       - product-service
19
```

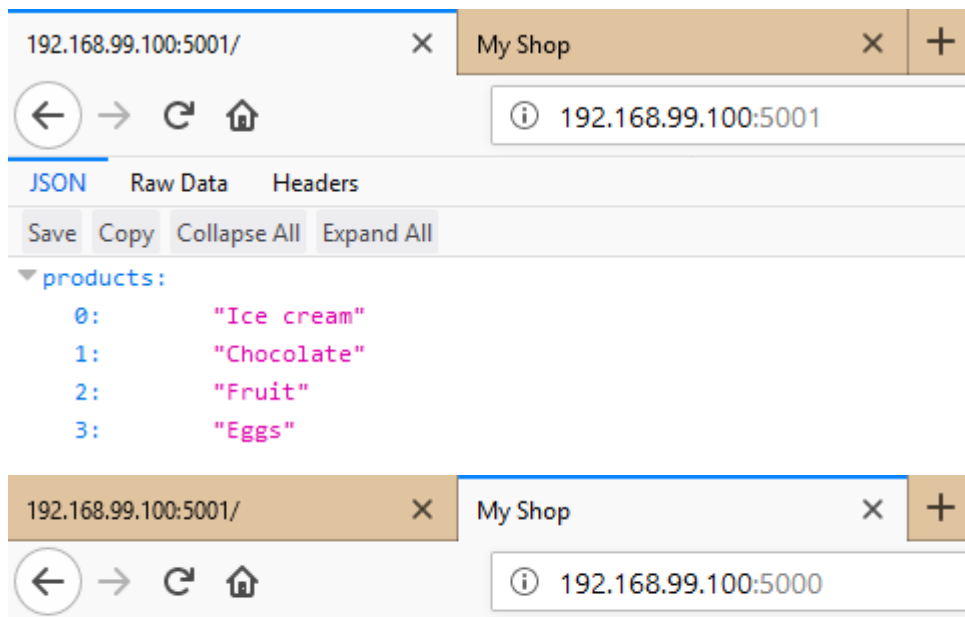
Kemudian buka kembali command prompt dan jalankan lagi “docker-compose up” seperti contoh dibawah ini

```
C:\Users\Tiok\Music\docker-compose>docker-compose up
Pulling website (php:apache)...
```

Maka selanjutnya proses download dan tunggu hingga selesai

```
C:\Windows\system32\cmd.exe - docker-compose up
Successfully tagged dockercompose_product-service:latest
WARNING: Image for service product-service was built because it did not already
exist. To rebuild this image you must use 'docker-compose build' or 'docker-comp
a5a6f2f73cd8: Downloading [=====>]
fcd9df7118ba: Downloading [=====>]
16.06MB/67.43MBwloading [=====>]
a5a6f2f73cd8: Downloading [=====>]
14.68MB/22.49MBwloading [=====>]
a5a6f2f73cd8: Downloading [=====>]
14.9MB/22.49MBwloading [=====>]
a5a6f2f73cd8: Downloading [=====>]
c425447c8835: Downloading [=====>]
13.95MB/17.13MBwloading [=====>]
c425447c8835: Downloading [=====>]
14.13MB/17.13MBwloading [=====>]
a5a6f2f73cd8: Downloading [=====>]
14.45MB/22.49MBwloading [=====>]
14.47MB/67.43MBwloading [=====>]
c425447c8835: Downloading [=====>]
13.41MB/17.13MBwloading [=====>]
12.71MB/17.13MBiting
33ed51bc30e8: Waiting
7c4215700bc4: Waiting
ef55a760eb7a: Waiting
d982e3946ac5: Waiting
```

Setelah download selesai kita coba buka browser dan ketikkan “192.168.99.100:5001” dan “192.168.99.100:5000”. Maka jika berhasil kedua alamat tersebut akan tampak seperti contoh dibawah ini



Welcome to my shop

- Ice cream
- Chocolate
- Fruit
- Eggs

Dari ilustrasi diatas menunjukkan container php mengambil data dari container python. Disini fungsi docker compose sebenarnya yaitu untuk membuat dan mengintegrasikan beberapa container agar bisa saling terhubung.

D. LATIHAN

Latihan diberikan dosen pengampu

E. TUGAS

Tugas diberikan dosen pengampu