

Tugas Container II3160

Afif Fahreza (18219058)

Link github **repository**: <https://github.com/afiffahreza/tst-container>

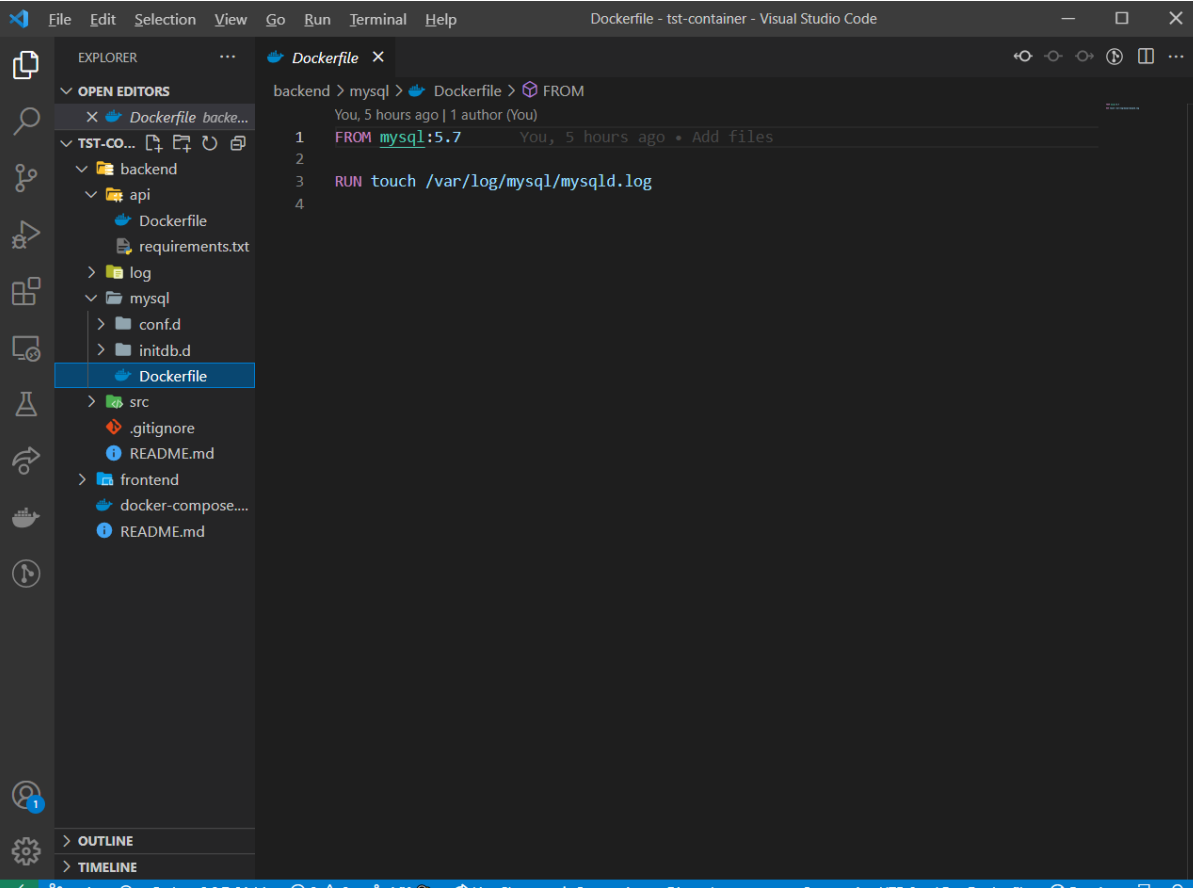
Link video **youtube**: <https://youtu.be/BrrJnVrfWsg>

Link **laporan** (jika pdf buram):

https://docs.google.com/document/d/1pp4XYDzIAR-lqImjlr0XbIIWNDILO8m_pqx4AhpB1us/edit?usp=sharing

Container yang dibuat ada 3, yaitu database (db), backend server (api), dan frontend (app). Database dibuat menggunakan mysql, backend menggunakan python FastAPI, dan frontend menggunakan React yang dijalankan dengan nginx.

Masing-masing container memiliki Dockerfile yang fungsinya berbeda-beda. Untuk database mysql, Dockerfile hanya berisi penggunaan image mysql dan pembuatan folder untuk log. Berikut adalah kode Dockerfile untuk database mysql.

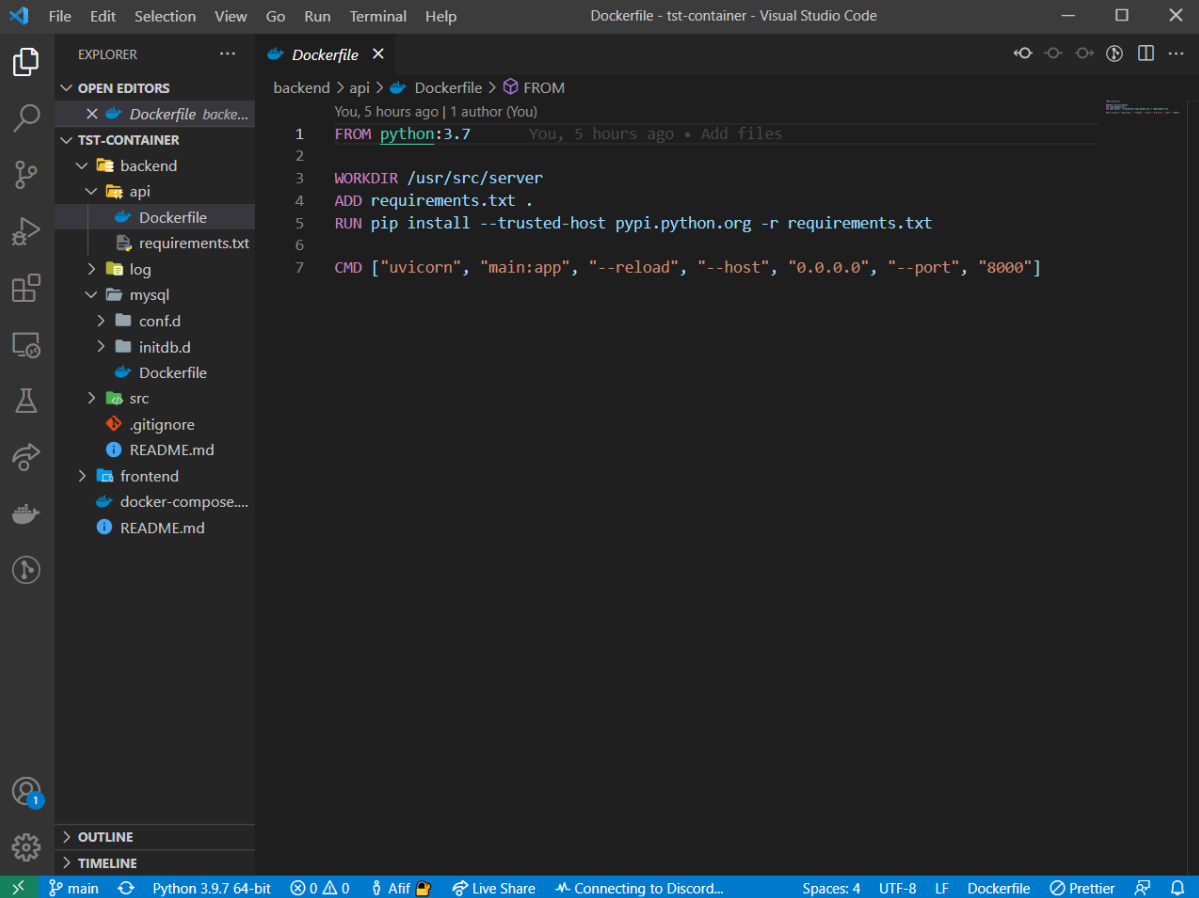


The screenshot shows the Visual Studio Code interface with a Dockerfile open in the editor. The Explorer sidebar on the left shows a project structure with folders for 'backend', 'frontend', and 'src', and files like 'Dockerfile', 'requirements.txt', '.gitignore', 'README.md', 'docker-compose...', and 'initdb.d'. The Dockerfile in the editor contains the following code:

```
backend > mysql > Dockerfile > FROM
You, 5 hours ago | 1 author (You)
1 FROM mysql:5.7 You, 5 hours ago • Add files
2
3 RUN touch /var/log/mysql/mysqlid.log
4
```

The status bar at the bottom indicates the current file is 'Dockerfile' and the editor is using 'Prettier' for formatting.

Untuk backend server, Dockerfile berisi penggunaan image python, mengganti folder work directory, melakukan instalasi library python yang digunakan, dan menjalankan server uvicorn agar API dapat diakses.



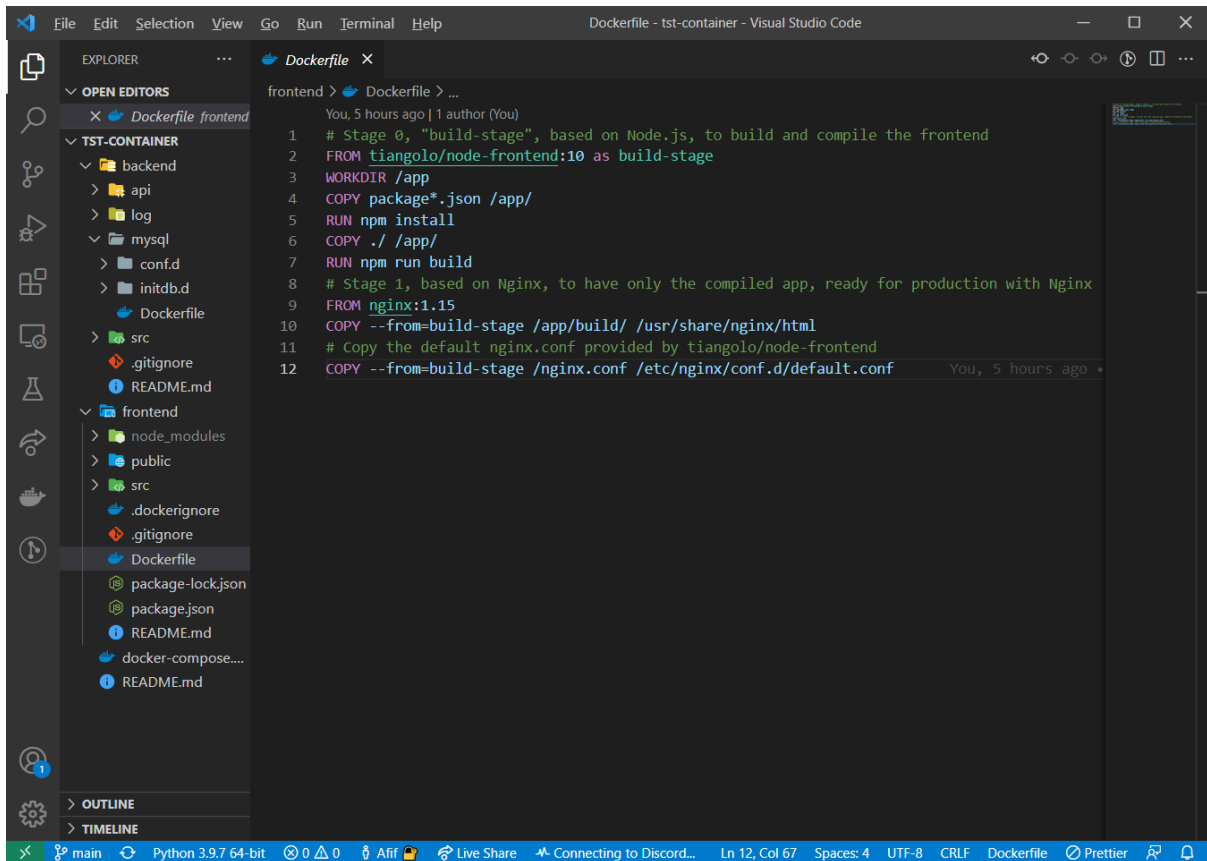
The screenshot shows the Visual Studio Code interface with a Dockerfile open in the editor. The Explorer sidebar on the left shows a project structure with folders like 'backend' and 'api', and files like 'Dockerfile', 'requirements.txt', 'log', 'mysql', 'conf.d', 'initdb.d', 'src', '.gitignore', 'README.md', 'frontend', 'docker-compose...', and 'README.md'. The Dockerfile in the editor contains the following content:

```
backend > api > Dockerfile > FROM
You, 5 hours ago | 1 author (You)
1 FROM python:3.7 You, 5 hours ago • Add files
2
3 WORKDIR /usr/src/server
4 ADD requirements.txt .
5 RUN pip install --trusted-host pypi.python.org -r requirements.txt
6
7 CMD ["uvicorn", "main:app", "--reload", "--host", "0.0.0.0", "--port", "8000"]
```

Untuk frontend, Dockerfile berisi 2 tahapan. Tahapan pertama adalah build aplikasi react menggunakan image dari tiangolo node. Tahapan kedua adalah run hasil dari build tahapan sebelumnya dengan menggunakan nginx.

Referensi:

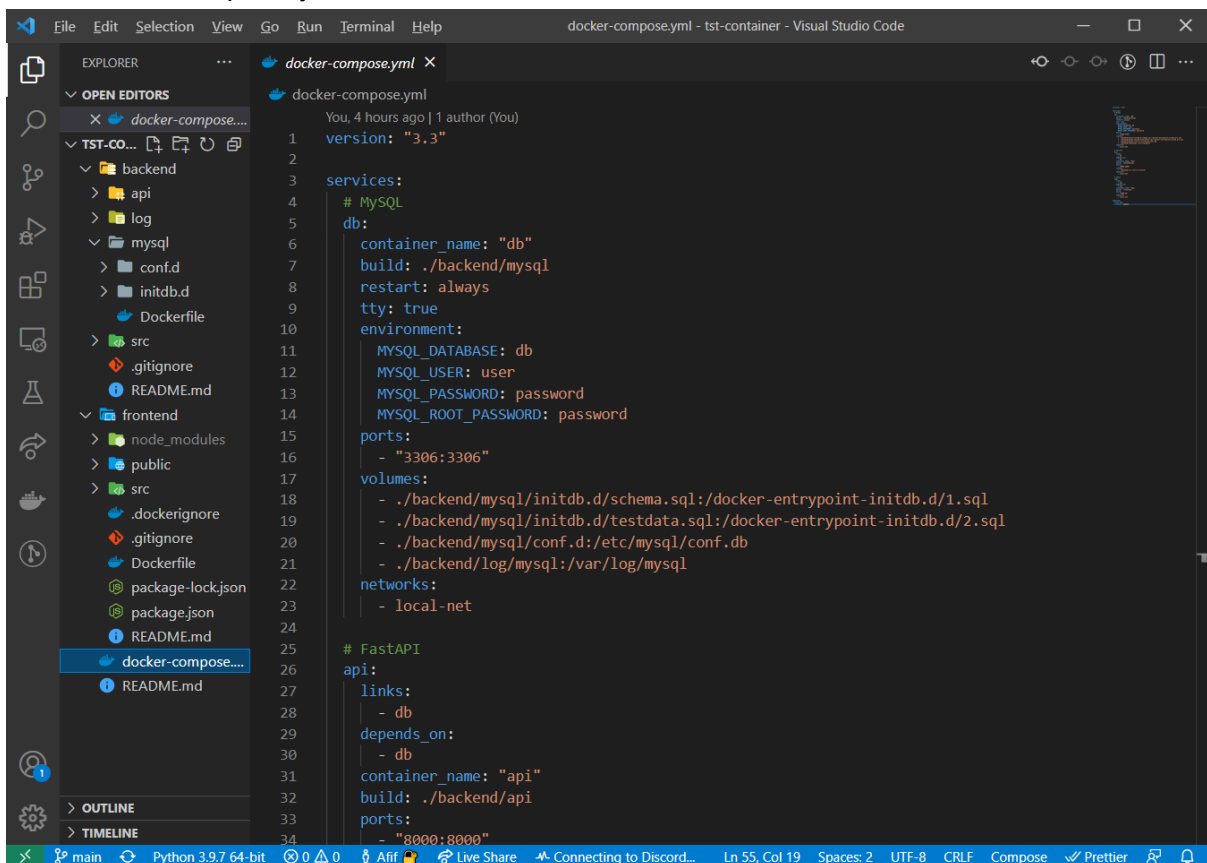
<https://tiangolo.medium.com/react-in-docker-with-nginx-built-with-multi-stage-docker-builds-including-testing-8cc49d6ec305>



The screenshot shows the Visual Studio Code interface with a Dockerfile open in the editor. The Explorer sidebar on the left shows a project structure with folders like 'backend', 'frontend', and 'src'. The Dockerfile content is as follows:

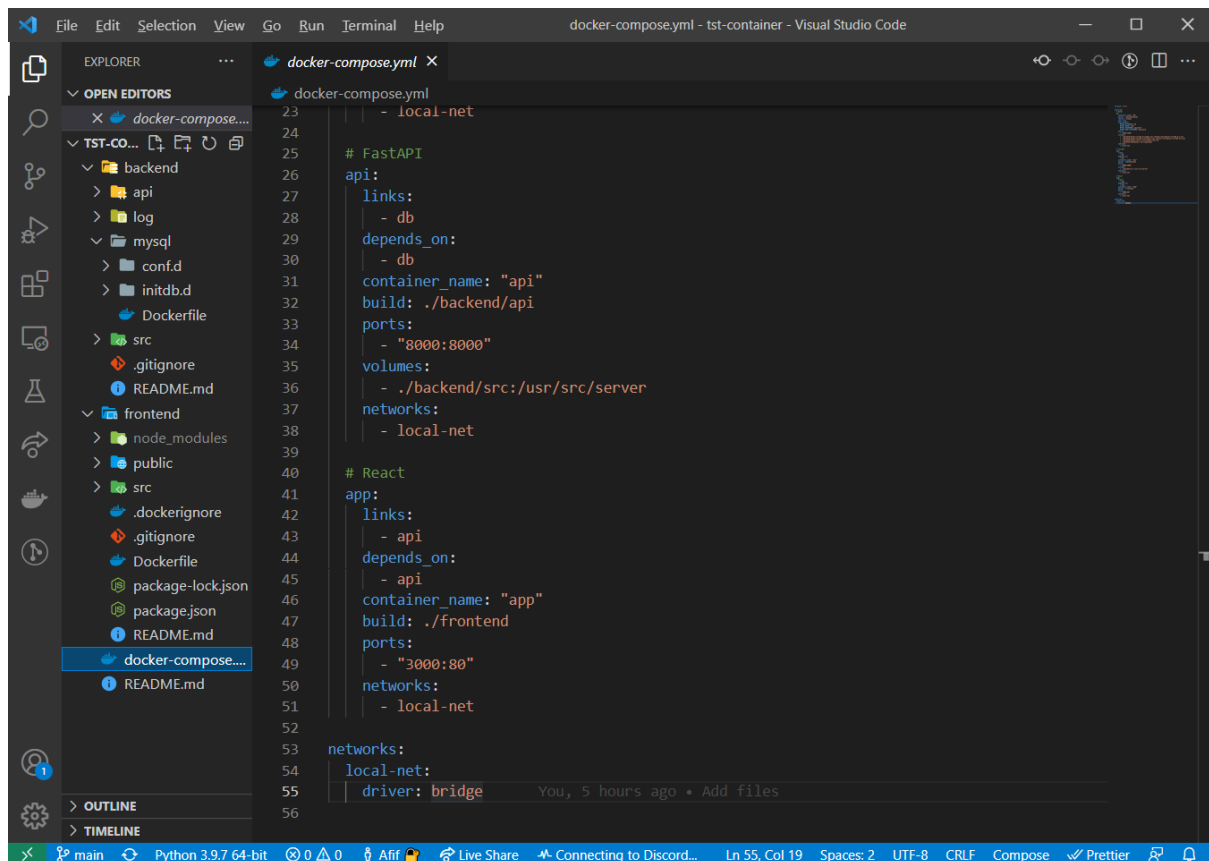
```
1 # Stage 0, "build-stage", based on Node.js, to build and compile the frontend
2 FROM tiangolo/node-frontend:10 as build-stage
3 WORKDIR /app
4 COPY package*.json /app/
5 RUN npm install
6 COPY ./ /app/
7 RUN npm run build
8 # Stage 1, based on Nginx, to have only the compiled app, ready for production with Nginx
9 FROM nginx:1.15
10 COPY --from=build-stage /app/build/ /usr/share/nginx/html
11 # Copy the default nginx.conf provided by tiangolo/node-frontend
12 COPY --from=build-stage /nginx.conf /etc/nginx/conf.d/default.conf
```

Ketiga Dockerfile tersebut dihubungkan dengan file docker-compose, berikut adalah file kode docker-compose.yml



The screenshot shows the Visual Studio Code interface with a docker-compose.yml file open in the editor. The Explorer sidebar on the left shows the same project structure as the previous screenshot. The docker-compose.yml content is as follows:

```
1 version: "3.3"
2
3 services:
4   # MySQL
5   db:
6     container_name: "db"
7     build: ./backend/mysql
8     restart: always
9     tty: true
10    environment:
11      MYSQL_DATABASE: db
12      MYSQL_USER: user
13      MYSQL_PASSWORD: password
14      MYSQL_ROOT_PASSWORD: password
15    ports:
16      - "3306:3306"
17    volumes:
18      - ./backend/mysql/initdb.d/schema.sql:/docker-entrypoint-initdb.d/1.sql
19      - ./backend/mysql/initdb.d/testdata.sql:/docker-entrypoint-initdb.d/2.sql
20      - ./backend/mysql/conf.d:/etc/mysql/conf.d
21      - ./backend/log/mysql:/var/log/mysql
22    networks:
23      - local-net
24
25   # FastAPI
26   api:
27     links:
28       - db
29     depends_on:
30       - db
31     container_name: "api"
32     build: ./backend/api
33     ports:
34       - "8000:8000"
```



```
23 | - local-net
24 |
25 | # FastAPI
26 | api:
27 |   links:
28 |     - db
29 |   depends_on:
30 |     - db
31 |   container_name: "api"
32 |   build: ./backend/api
33 |   ports:
34 |     - "8000:8000"
35 |   volumes:
36 |     - ./backend/src:/usr/src/server
37 |   networks:
38 |     - local-net
39 |
40 | # React
41 | app:
42 |   links:
43 |     - api
44 |   depends_on:
45 |     - api
46 |   container_name: "app"
47 |   build: ./frontend
48 |   ports:
49 |     - "3000:80"
50 |   networks:
51 |     - local-net
52 |
53 | networks:
54 |   local-net:
55 |     driver: bridge
56 |
```

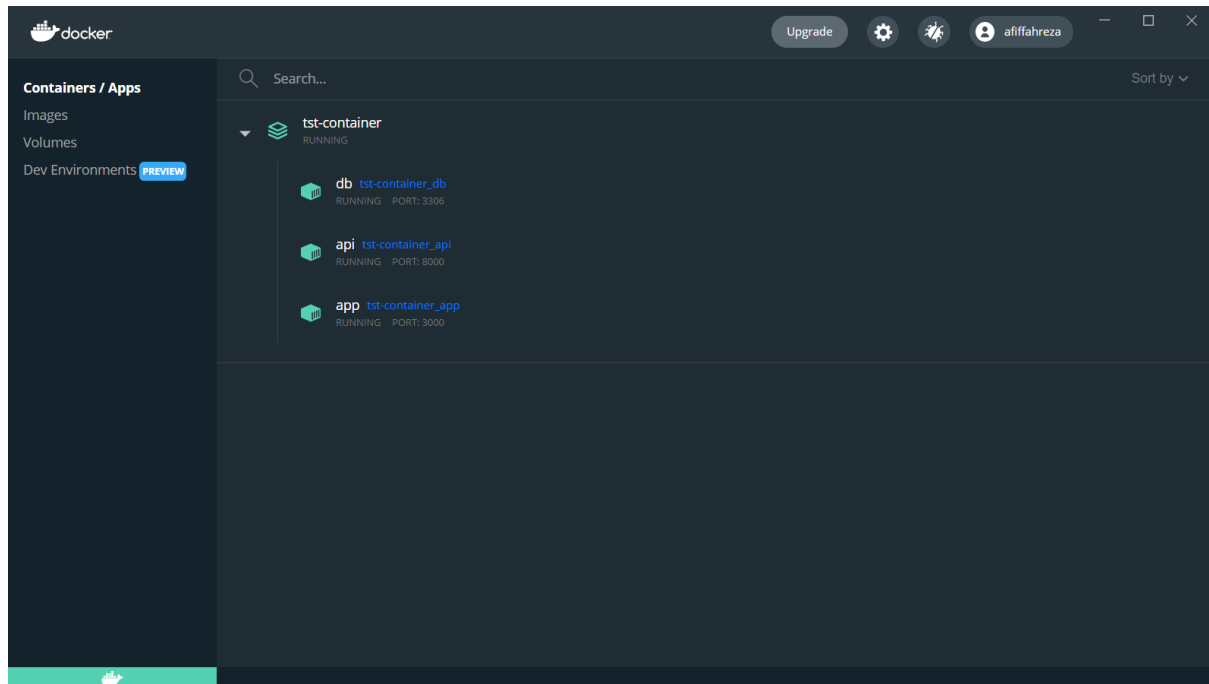
Service dibagi menjadi 3 bagian, 1 untuk masing-masing container. Pada setiap bagian ada komponen build yang diisi dengan directory dari Dockerfile container yang bersangkutan. Contoh pada container app, build directory yang digunakan adalah ./frontend karena Dockerfile terdapat pada folder frontend. Pada setiap bagian juga ada komponen networks yang diisi dengan local-net. Network local-net diberikan detail spesifikasi pada paling bawah, yaitu dengan driver bridge.

Pada container db, terdapat komponen environment yang berisi detail dari database. Komponen ports diisi dengan "3306:3306", 3306 yang kiri merupakan port local agar dapat diakses, 3306 yang kanan merupakan port default mysql pada docker. Komponen volumes diisi dengan file config dan init db untuk inisiasi database dan pengisian database.

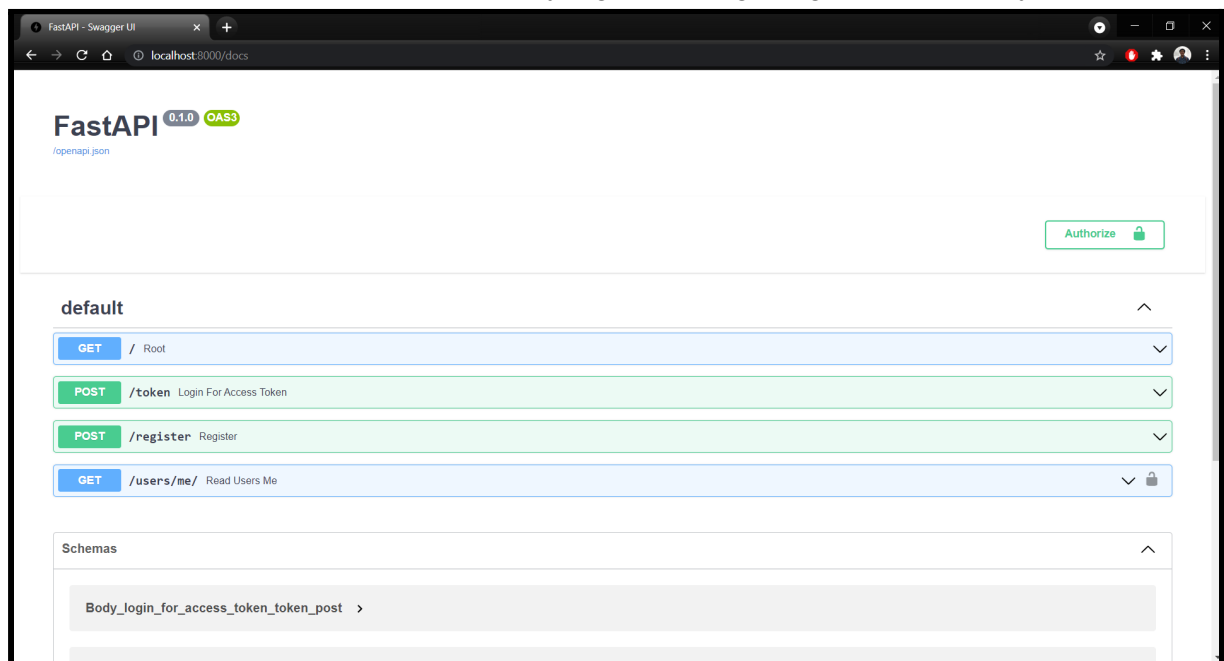
Pada container api, ports yang digunakan adalah "8000:8000", 8000 yang kiri merupakan port local, yang kanan merupakan uvicorn pada docker.

Pada container app, ports yang digunakan adalah "3000:80", 3000 untuk akses local, 80 merupakan default http server nginx.

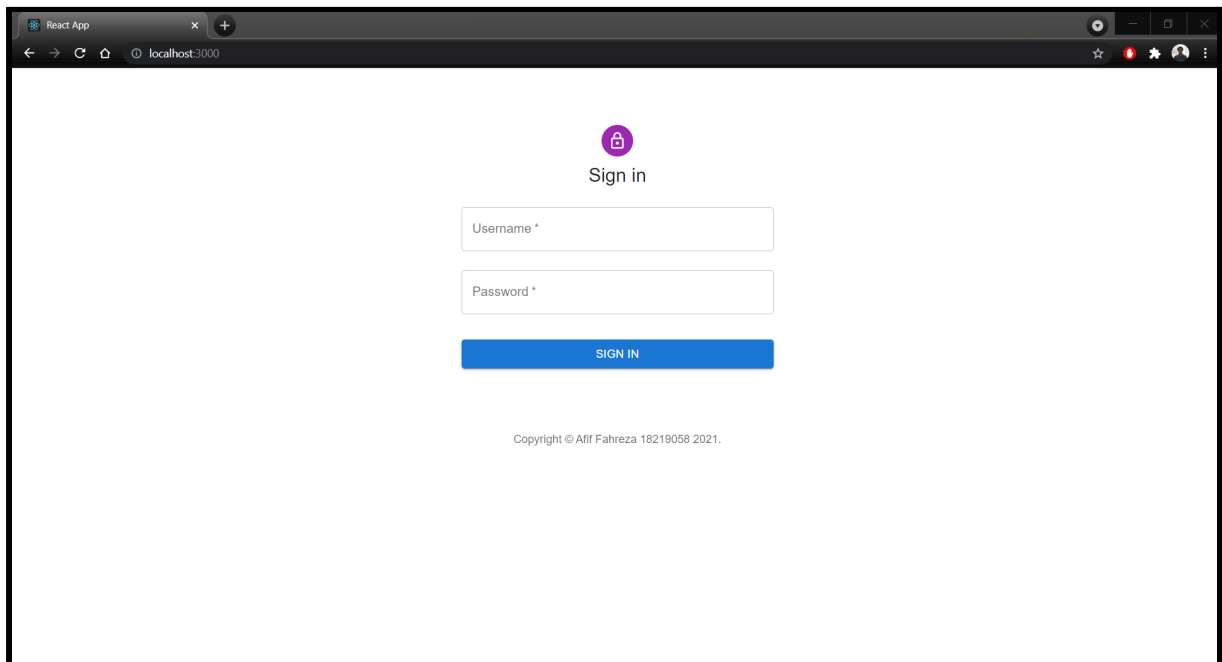
Berikut adalah tampilan docker desktop setelah docker-compose up



Berikut adalah tampilan dokumentasi API yang terhubung dengan database mysql



Berikut adalah tampilan frontend login



React App

localhost:3000

Sign in

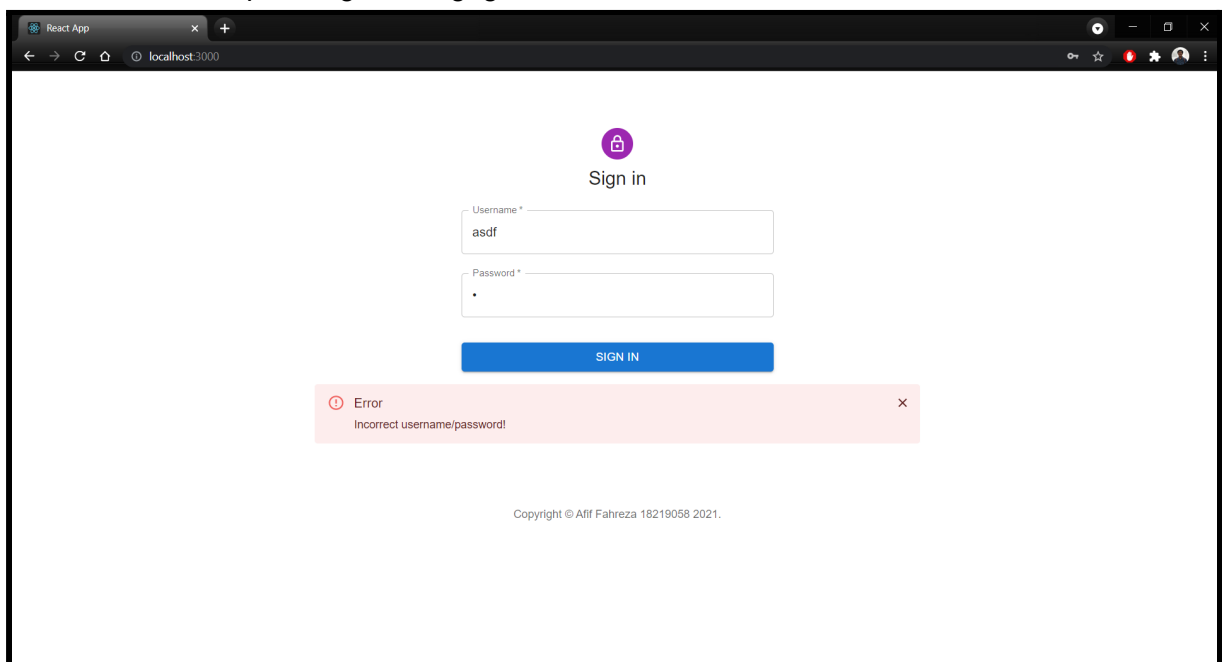
Username *

Password *

SIGN IN

Copyright © Afif Fahreza 18219058 2021.

Berikut adalah tampilan login saat gagal



React App

localhost:3000

Sign in

Username *

asdf

Password *

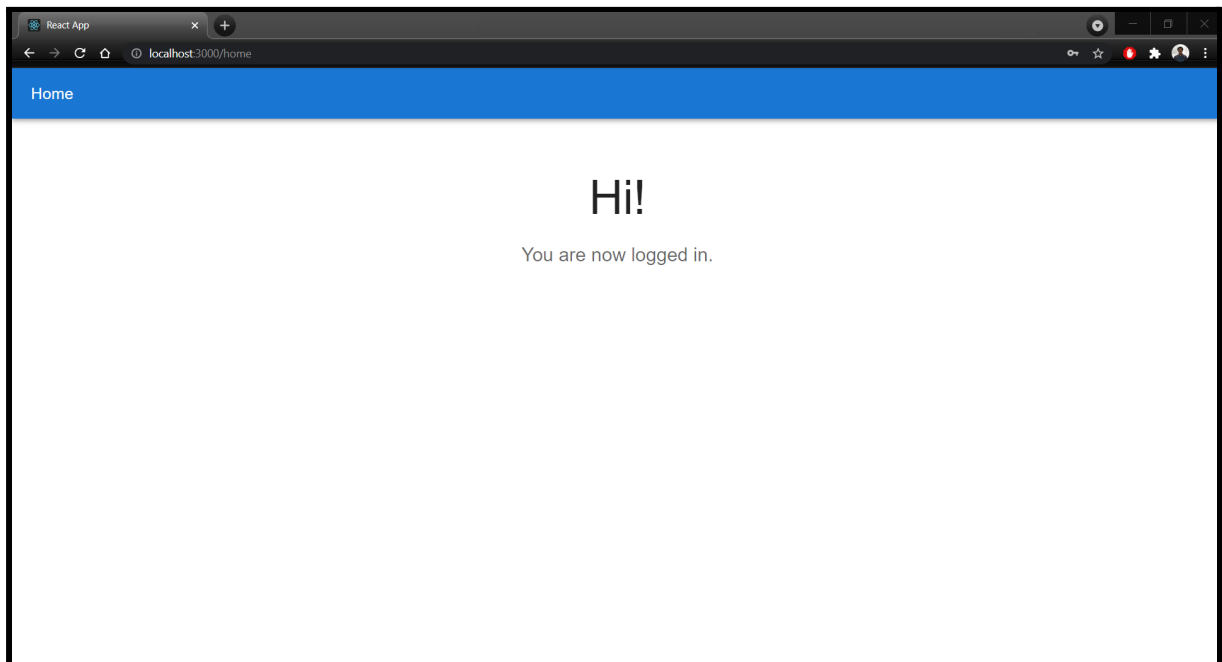
•

SIGN IN

Error
Incorrect username/password!

Copyright © Afif Fahreza 18219058 2021.

Berikut adalah tampilan frontend saat berhasil login



Berikut adalah database user yang diakses dengan CLI docker desktop

```
docker exec -it 2e84e20c0e4cb5a55fdde7e19d6e519d0baa8c46782af9a5da735980cbb433f8 /bin/sh
# mysql -p db
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.35 MySQL Community Server (GPL)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show tables
+-----+
| Tables_in_db |
+-----+
| user          |
+-----+
1 row in set (0.00 sec)

mysql> select * from user;
+-----+-----+-----+-----+
| id | username | hashed_password | disabled |
+-----+-----+-----+-----+
| 1 | asdf     | $2b$12$hGKD0wG56pxrtLj8644EH0EIrLHLc7T7JtwSmp.//cV12LuuEQuzu | 0 |
| 2 | dummy1   | dummypass1      | 0 |
| 3 | dummy2   | dummypass2      | 0 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```