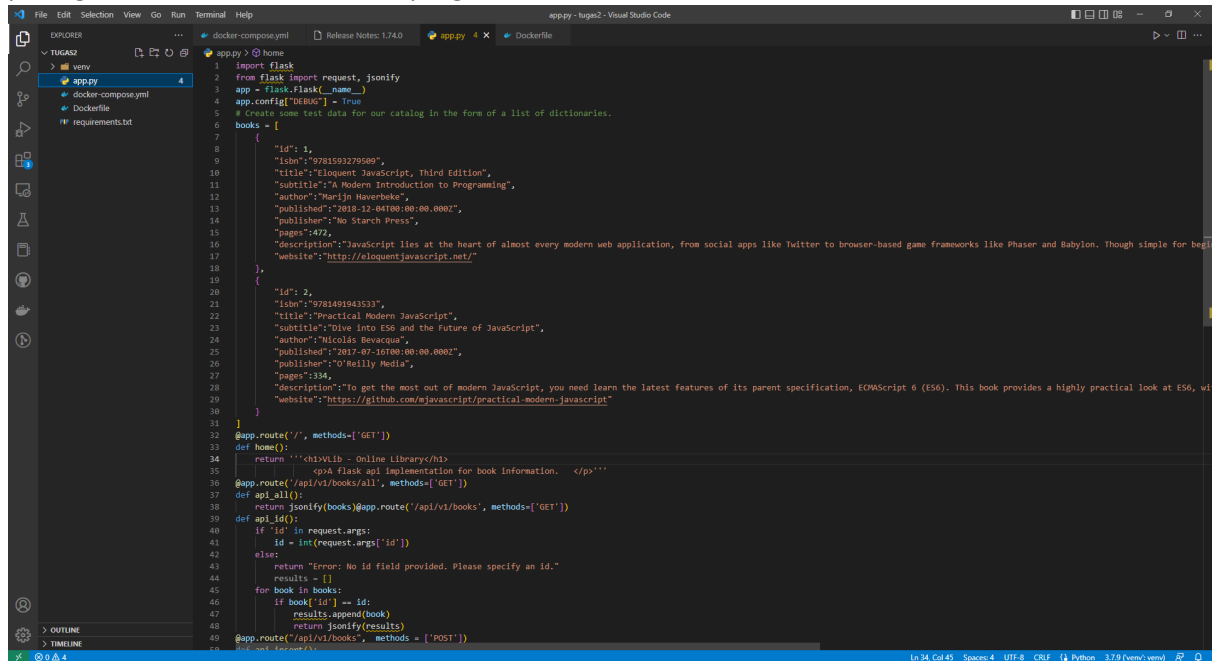


Nama : Indra Lukman Hakim

NIM : 11191030

Sistem Terdistribusi B – Tugas 2 Deploy API Docker

1. Membuat API menggunakan library *flask* dengan Bahasa pemrograman *python*, terdapat pada gambar adalah contoh data yang akan dimasukkan keadalam API.



```
1 import flask
2 from flask import request, jsonify
3 app = flask.Flask(__name__)
4 app.config['DEBUG'] = True
5 # Create some test data for our catalog in the form of a list of dictionaries.
6 books = [
7     {
8         "id": 1,
9         "isbn": "9781593279500",
10        "title": "Eloquent JavaScript, Third Edition",
11        "subtitle": "A Modern Introduction to Programming",
12        "author": "Marijn Haverbeke",
13        "published": "2010-12-04T00:00:00.000Z",
14        "publisher": "No Starch Press",
15        "pages": 472,
16        "description": "JavaScript lies at the heart of almost every modern web application, from social apps like Twitter to browser-based game frameworks like Phaser and Babylon. Though simple for begi
17        "website": "http://eloquentjavascript.net/"
18    },
19    {
20        "id": 2,
21        "isbn": "9781491943533",
22        "title": "Practical Modern JavaScript",
23        "subtitle": "Dive into ES6 and the Future of JavaScript",
24        "author": "Nicolas Beaudouin",
25        "published": "2017-07-16T00:00:00.000Z",
26        "publisher": "O'Reilly Media",
27        "pages": 284,
28        "description": "To get the most out of modern JavaScript, you need learn the latest features of its parent specification, ECMAScript 6 (ES6). This book provides a highly practical look at ES6, w
29        "website": "https://github.com/mjavascript/practical-modern-javascript"
30    }
31 ]
32 @app.route('/', methods=['GET'])
33 def home():
34     return '''ch15vLib - Online Library</h3>
35     <p>OPA flask api Implementation for book information. </p>'''
36 @app.route('/api/v1/books/all', methods=['GET'])
37 def api_all():
38     return jsonify(books)
39 @app.route('/api/v1/books', methods=['GET'])
40 def api_id():
41     if 'id' in request.args:
42         id = int(request.args['id'])
43     else:
44         return "Error: No id field provided. Please specify an id."
45     results = []
46     for book in books:
47         if book['id'] == id:
48             results.append(book)
49             return jsonify(results)
50 @app.route('/api/v1/books', methods = ['POST'])
51 def post_book():
```

2. Membuat requirements python dengan command `pip freeze > requirements.txt`. Kemudian membuat Docker file dengan perintah dibawah dengan penjelasan sebagai berikut:

- Dockerfile akan membuat berdasarkan base image alpine.
- Commad RUN akan menjalankan perintah command didalam container. Yakni Pip update dan pip upgrade.
- WORKDIR akan membuat folder baru didalam container.
- COPY akan menstransfer file dari direkotri saat ini kedalam container.

- The screenshot displays the Visual Studio Code interface with a Dockerfile open in the editor. The Dockerfile contains the following instructions:

```
Dockerfile > -
1 FROM alpine:latest
2 RUN apk update
3 RUN apk add py-pip
4 RUN apk add --no-cache python3-dev
5 RUN pip install --upgrade pip
6 WORKDIR /app
7 COPY . /app
8 RUN pip --no-cache-dir install -r requirements.txt
9 CMD ["python3", "app.py"]
```

The Explorer sidebar on the left shows the project structure with files like `venv`, `main.py`, `docker-compose.yml`, `Dockerfile`, and `requirements.txt`.

The bottom panel shows the **TERMINAL** output, which includes an `AssertionError` message and subsequent `curl` commands testing the application's response:

```
AssertionError: The setup method 'route' can no longer be called on the application. It has already handled its first request, any changes will not be applied consistently.
Make sure all imports, decorators, functions, etc. needed to set up the application are done before running it.
127.0.0.1 - - [09/Dec/2022 18:12:33] "SET /api/v1/books/all?_debugger__yes&cd=resource&f=style.css HTTP/1.1" 304 -
127.0.0.1 - - [09/Dec/2022 18:12:33] "SET /api/v1/books/all?_debugger__yes&cd=resource&f=debugger.js HTTP/1.1" 304 -
127.0.0.1 - - [09/Dec/2022 18:12:33] "SET /api/v1/books/all?_debugger__yes&cd=resource&f=console.png HTTP/1.1" 304 -

In [9]: $ curl -X GET http://localhost:5050/api/v1/books/all
$ pip freeze > requirements.txt

In [9]: $ curl -X GET http://localhost:5050/api/v1/books/all
$
```

The status bar at the bottom indicates the file is `Dockerfile` at line 9, column 4.

- ```
$ pip freeze > requirements.txt

indrag@SKTOP-Q740C: ~/HGK661 /d/p/sister/tugas2
$ docker build -t flask-rest-api .
$ docker build -t flask-rest-api
[+] Building 79.7s (14/14) FINISHED
=> [internal] load build definition from Dockerfile
=> -- transferring dockerfile: 267B
=> [internal] load .dockerignore
=> -- transferring context: 2B
=> [internal] load metadata for docker.io/library/alpine:latest
=> [auth] library/alpine:pull token for registry-1.docker.io
=> [internal] load build context
=> -- transferring context: 12.4kB
=> [3/8] FROM docker.io/library/alpine:latest@sha256:8914eb54f968791fa6a8638949e480fef81e697884fba772b3976835194c6d4
=> -- resolve docker.io/library/alpine:latest@sha256:8914eb54f968791fa6a8638949e480fef81e697884fba772b3976835194c6d4
=> -- sha256:8914eb54f968791fa6a8638949e480fef81e697884fba772b3976835194c6d4 1.64kB / 1.64kB
=> -- sha256:c0d488a800e4127c334ad20d61d7bc1b409754032717dfab52262adc02380c 528B / 528B
=> -- sha256:49176f190c7e5c8b51ac35ab6cd5e451235218198c4d9680d6f8d03fbbf3da 1.47kB / 1.47kB
=> -- sha256:c1528095176d6f233acef1f2182a2345a300f635783ac5fadc0715 3.37kB / 3.37kB
=> -- extracting sha256:c1528095176d6f233acef1f2182a2345a300f635783ac5fadc0715
=> [2/8] RUN apk update
=> [3/8] RUN apk add py-pip
=> [4/8] RUN apk add --no-cache python3-dev
=> [5/8] RUN pip install --upgrade pip
=> [6/8] Mkdir /app
=> [7/8] COPY . /app
=> [8/8] RUN pip --no-cache-dir install -r requirements.txt
=> exporting image
=> -- exporting layers
=> -- writing image sha256:74cfed4eed199122a5fcfddeda291d4f09582f5eb041922a1e169a8e9b087
=> -- naming to docker.io/library/flask-rest-api
```

- ```
indra@DESKTOP-QT0HJUC MINGW64 /d/p/sister/tugas2
$ docker images
```
- | REPOSITORY | TAG | IMAGE ID | CREATED | SIZE |
|------------------------|---------------|--------------|----------------|--------|
| flask-rest-api | latest | 74cfe6d3ee4d | 10 seconds ago | 158MB |
| php | 8-fpm | 644c138dbade | 24 hours ago | 73.7MB |
| tugas2 | latest | 644c138dbade | 24 hours ago | 73.7MB |
| mysql | latest | a3a2968869cf | 9 days ago | 538MB |
| docker/getting-started | latest | e5be50c31cb9 | 13 days ago | 29.8MB |
| nginx | stable-alpine | c31fe73098ae | 3 weeks ago | 23.5MB |

5. Run Docker image dengan command `docker run -d -p 5000:5000 flask-rest-api`. Kemudian dapat dilihat container yang sedang berjalan dengan command `docker ps -a`.

```
indra@DESKTOP-QT0HJUC MINGW64 /d/p/sister/tugas2
$ docker run -d -p 5000:5000 flask-rest-api
772943f9d550c4a054947050ce4305e851eee6d03236b3410c611d458974a31c

indra@DESKTOP-QT0HJUC MINGW64 /d/p/sister/tugas2
$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
772943f9d550   flask-rest-api "python3 app.py"        6 seconds ago Up 5 seconds  0.0.0.0:5000->5000/tcp             priceless_vaughan
a90d2e0d42fb   nginx:stable-alpine "/docker-entrypoint..." 19 hours ago   Exited (137)  19 hours ago                       nginx
07c548bfcbbe   php:8-fpm      "docker-php-entrypoi..." 20 hours ago   Exited (137)  19 hours ago                       php
dee4c98658d4   mysql:latest   "docker-entrypoint.s..." 24 hours ago   Exited (137)  19 hours ago                       mysql

indra@DESKTOP-QT0HJUC MINGW64 /d/p/sister/tugas2
$
```

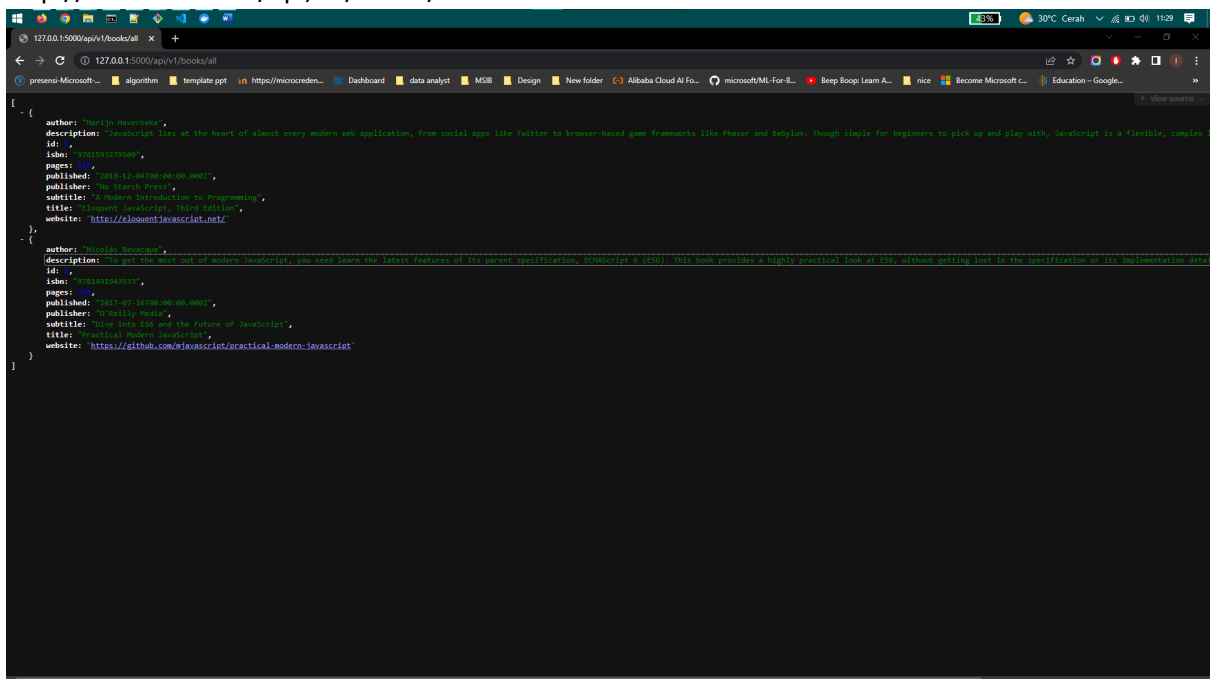
6. Kemudian kita dapat melihat log dari container yang berjalan dengan command `docker logs "ID CONTAINER"`,

```
indra@DESKTOP-QT0HJUC MINGW64 /d/p/sister/tugas2
$ docker logs 772933f9d550
Error: No such container: 772933f9d550

indra@DESKTOP-QT0HJUC MINGW64 /d/p/sister/tugas2
$ docker logs 772943f9d550
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit

indra@DESKTOP-QT0HJUC MINGW64 /d/p/sister/tugas2
$
```

7. Didapatkan api berjalan dengan IP 127.0.0.1:5000 yakni localhost dengan port 5000. Lalu dapat dilihat API method get dengan membuka tautan `http://127.0.0.1:5000/api/v1/books/all`.



```
[{"author": "Marijn Haverbeke", "description": "JavaScript lies at the heart of almost every modern web application, from social apps like Twitter to browser-based game frameworks like Phaser and Babylon. Though simple for beginners to pick up and play with, JavaScript is a flexible, complex", "id": "770103279000", "isbn": "9781091279000", "pages": 176, "published": "2012-12-04T00:00:00-0000", "publisher": "No Starch Press", "subtitle": "A Modern Introduction to Programming", "title": "Eloquent JavaScript: Third Edition", "website": "https://eloquentjavascript.net/"}, {"author": "Nicolas D'Amore", "description": "To get the most out of modern JavaScript, you need learn the latest features of the current specification (ES6+). This book provides a highly practical look at ES6, without getting lost in the specification or the implementation details", "id": "9781091343021", "isbn": "9781091343021", "pages": 176, "published": "2017-07-18T00:00:00-0000", "publisher": "O'Reilly Media", "subtitle": "How and Why the Future of JavaScript", "title": "Practical Modern JavaScript", "website": "https://github.com/mjjavascript/practical-modern-javascript"}]
```

