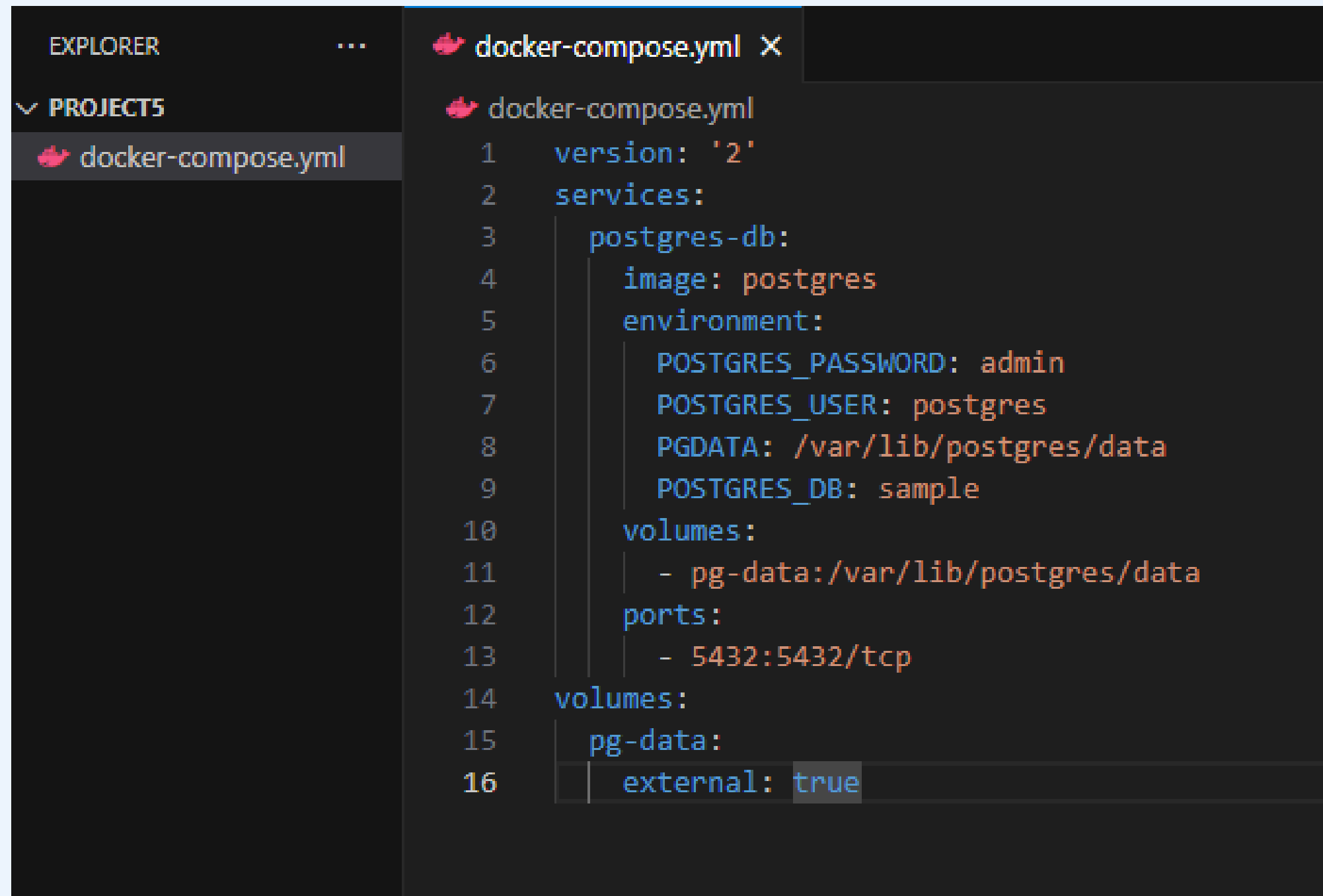


# Project 5

Kelompok 8 - Infinity



## 1. Membuat file docker-compose.yml

A screenshot of the Visual Studio Code editor interface. On the left, the 'EXPLORER' sidebar shows a project named 'PROJECT5' with a file 'docker-compose.yml' selected. The main editor area displays the content of 'docker-compose.yml'. The file is a YAML configuration for a PostgreSQL database service. It includes a version '2', a service named 'postgres-db' with image 'postgres', environment variables for password, user, data directory, and database name, a volume 'pg-data' mapped to the data directory, and a port mapping from 5432 to 5432/tcp. The volume is also defined as external.

```
1  version: '2'
2  services:
3    postgres-db:
4      image: postgres
5      environment:
6        POSTGRES_PASSWORD: admin
7        POSTGRES_USER: postgres
8        PGDATA: /var/lib/postgres/data
9        POSTGRES_DB: sample
10     volumes:
11       - pg-data:/var/lib/postgres/data
12     ports:
13       - 5432:5432/tcp
14   volumes:
15     pg-data:
16       external: true
```

## 2. Mengaktifkan docker

The screenshot displays the Docker Desktop application window. The top navigation bar includes a search bar and icons for various features. The left sidebar lists navigation options: Containers, Images, Volumes, Builds, Dev Environments (marked BETA), Docker Scout, Extensions, and Add Extensions. The main panel is titled 'Containers' and shows system usage metrics: Container CPU usage at 0.00% / 1600% (16 CPUs available) and Container memory usage at 0B / 7.29GB. Below these metrics is a search bar and a toggle for 'Only show running containers'. A table lists three containers: 'etl' (Exited, 6 days ago), 'project4' (Exited, 8 days ago), and 'project5' (Running (1/1), 9 minutes ago). The bottom section features 'Walkthroughs' for 'Multi-container applications' (8 mins) and 'Containerize your application' (3 mins). The status bar at the bottom indicates 'Engine running', system resources (RAM 2.09 GB, CPU 0.13%), and a 'Signed in' user, along with a 'New version available' notification.

docker desktop

Search for images, containers, volumes,... **Ctrl+K**

**Containers** [Give feedback](#)

Container CPU usage ⓘ 0.00% / 1600% (16 CPUs available)

Container memory usage ⓘ 0B / 7.29GB [Show charts](#)


Search

Only show running containers

	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	> etl		Exited	0%		6 days ago	<a href="#">▶</a> <a href="#">⋮</a> <a href="#">🗑</a>
<input type="checkbox"/>	> project4		Exited	0%		8 days ago	<a href="#">▶</a> <a href="#">⋮</a> <a href="#">🗑</a>
<input type="checkbox"/>	> project5		Running (1/1)	0%		9 minutes ago	<a href="#">■</a> <a href="#">⋮</a> <a href="#">🗑</a>

Showing 3 items

**Walkthroughs** [×](#)

 Multi-container applications  
8 mins

`$ docker init` Containerize your application  
3 mins

[View more in the Learning center](#)

Engine running [▶](#) [⏸](#) [🔌](#) [⋮](#) RAM 2.09 GB CPU 0.13% [👤](#) Signed in [🔔](#) New version available [🔔](#) 2

### 3. Menjalankan docker dengan docker-compose up -d

```
PS C:\Users\hallo\OneDrive\Documents\Kuliah\Stupen\project5> docker volume create pg-data
pg-data
PS C:\Users\hallo\OneDrive\Documents\Kuliah\Stupen\project5> docker-compose up -d
[+] Running 1/2
- Network project5_default          Created
✓ Container project5-postgres-db-1 Started
PS C:\Users\hallo\OneDrive\Documents\Kuliah\Stupen\project5> docker container ps
```

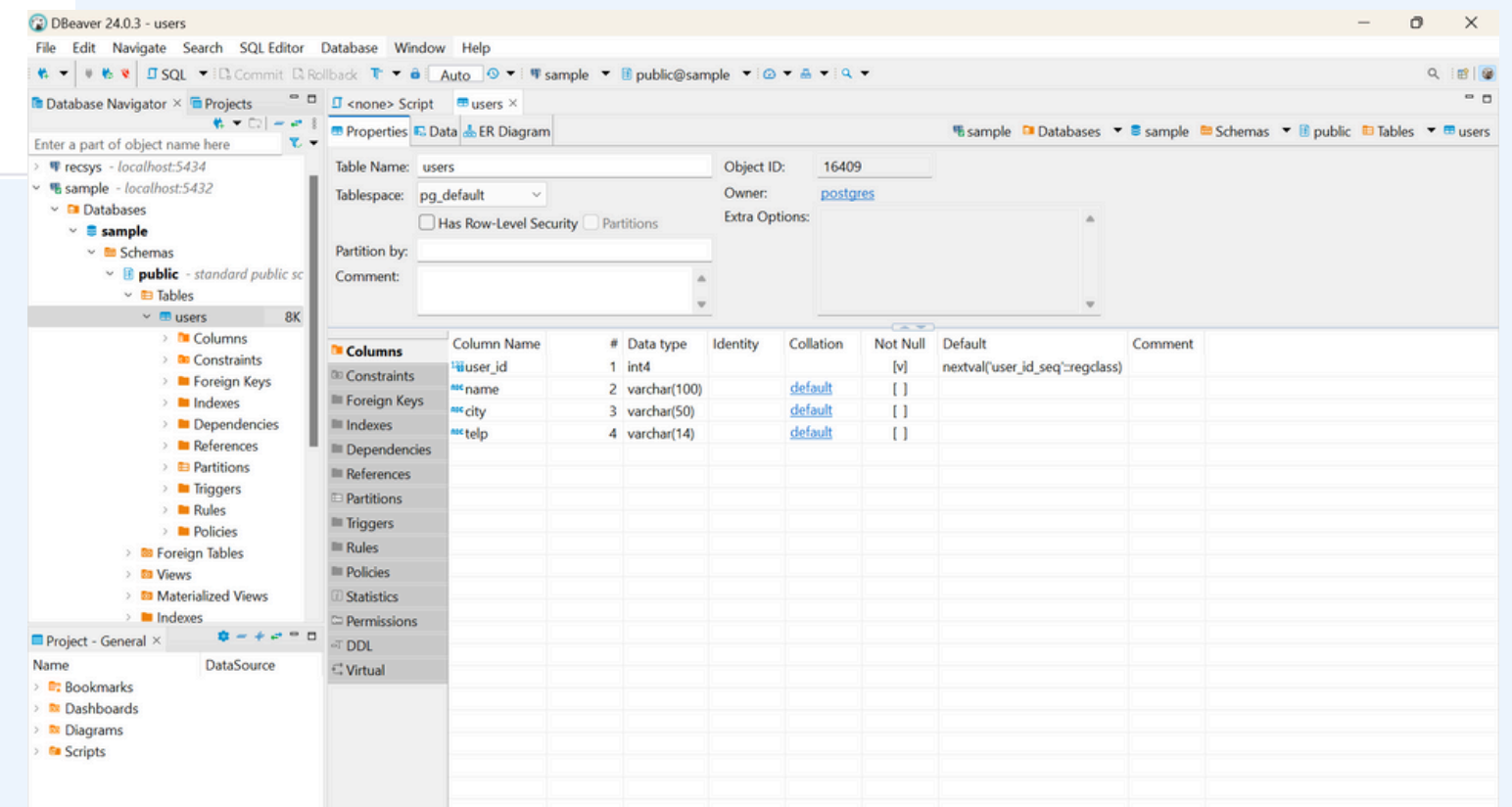
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d6ebd46c6207	postgres	"docker-entrypoint.s..."	15 seconds ago	Up 14 seconds	0.0.0.0:5432->5432/tcp	project5-postgres-db-1

## 4. Membuat connection di dbeaver postgres

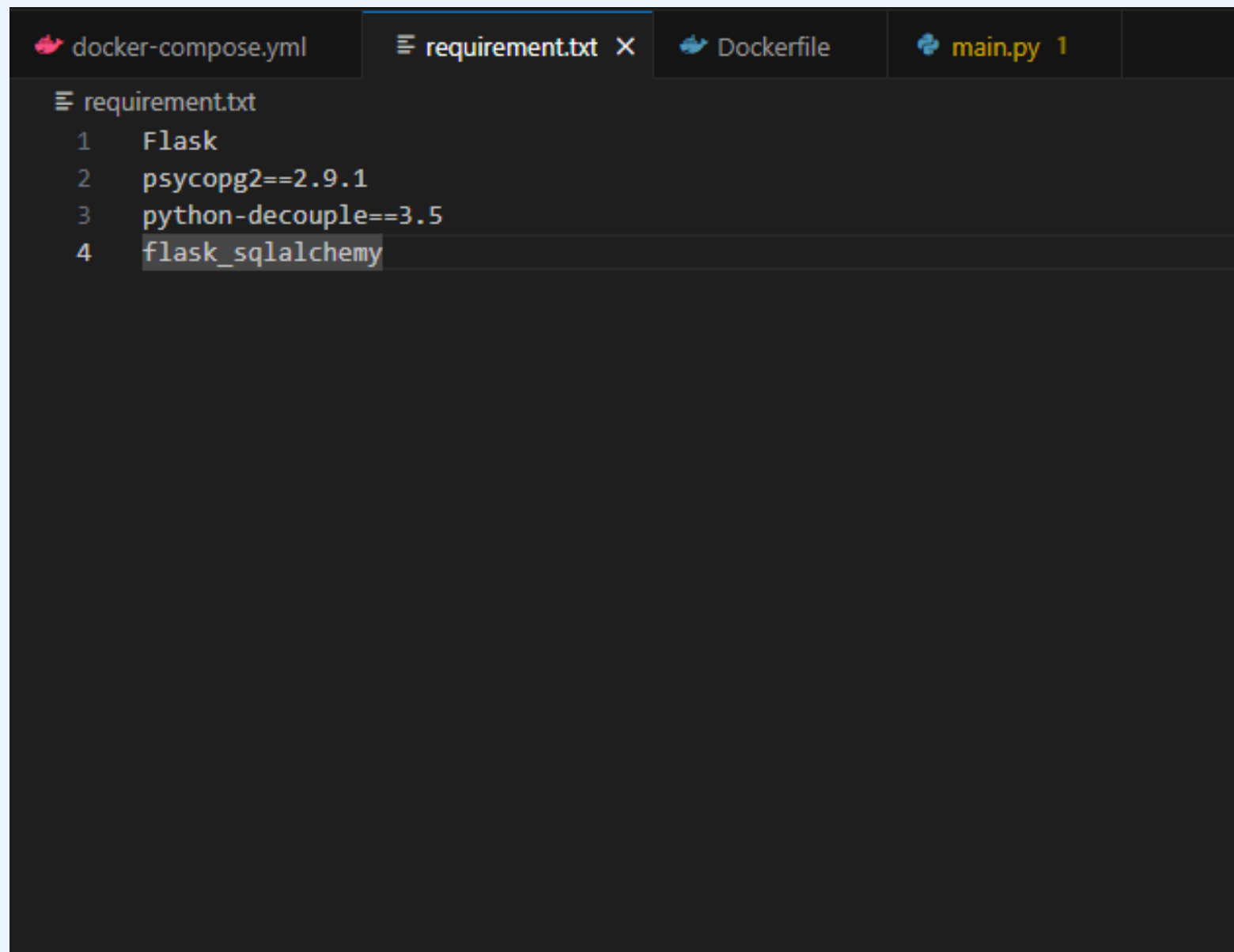
```
DROP TABLE IF EXISTS public.users;

CREATE SEQUENCE user_id_seq
  START WITH 1
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  CACHE 1;

CREATE TABLE IF NOT EXISTS public.users
(
  user_id int4 NOT NULL DEFAULT nextval('user_id_seq'::regclass),
  name varchar(100) NULL,
  city varchar(50) NULL,
  telp varchar(14) NULL,
  CONSTRAINT users_pkey PRIMARY KEY (user_id)
);
```



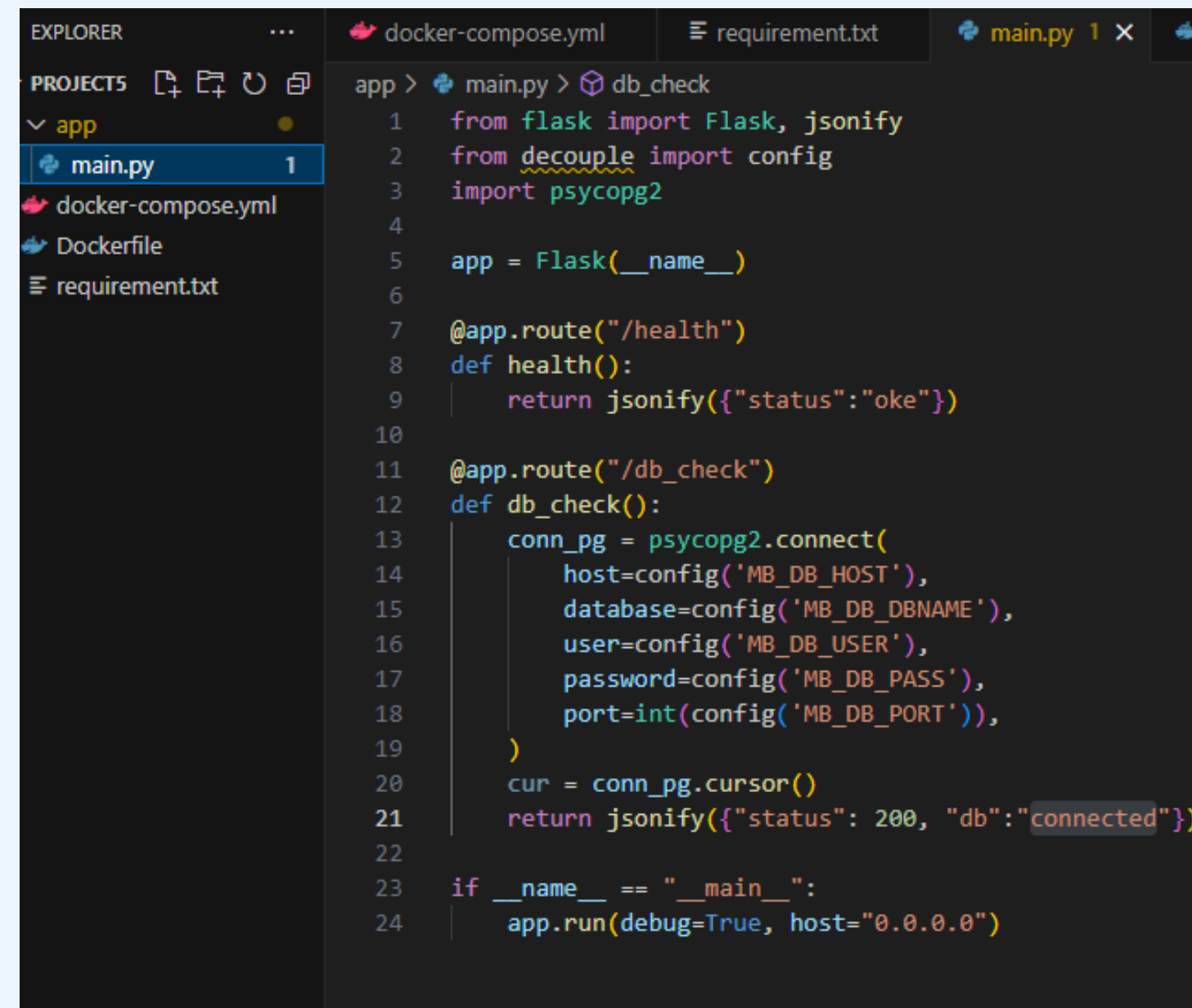
## 5. Membuat file requirement.txt

A screenshot of a code editor with a dark theme. The editor has four tabs at the top: 'docker-compose.yml', 'requirement.txt' (which is active and has a close button), 'Dockerfile', and 'main.py 1'. The 'requirement.txt' tab shows a list of Python dependencies. The text is as follows:

```
1 Flask
2 psycopg2==2.9.1
3 python-decouple==3.5
4 flask_sqlalchemy
```

The fourth line, 'flask\_sqlalchemy', is currently selected with the mouse cursor.

## 6. Membuat file main.py di dalam folder app

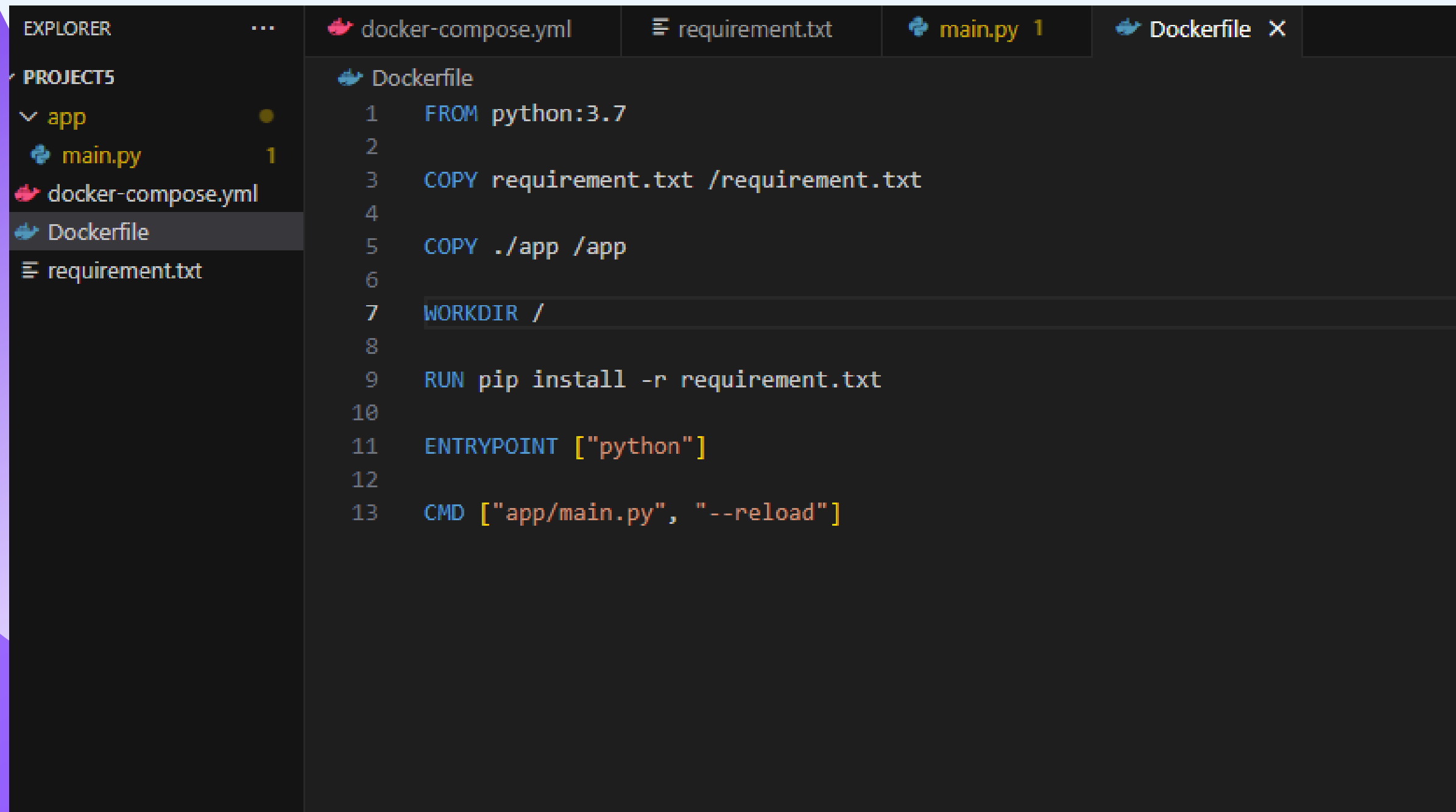


The screenshot shows a code editor with the following components:

- EXPLORER:** A sidebar on the left showing the project structure. Under the 'app' folder, 'main.py' is highlighted with a blue selection bar and a '1' icon.
- FILES:** A list of files in the project: 'docker-compose.yml', 'Dockerfile', and 'requirement.txt'.
- EDITOR:** The main workspace showing the content of 'main.py'. The file path 'app > main.py' is visible at the top of the editor. The code is as follows:

```
1 from flask import Flask, jsonify
2 from decouple import config
3 import psycopg2
4
5 app = Flask(__name__)
6
7 @app.route("/health")
8 def health():
9     return jsonify({"status": "oke"})
10
11 @app.route("/db_check")
12 def db_check():
13     conn_pg = psycopg2.connect(
14         host=config('MB_DB_HOST'),
15         database=config('MB_DB_DBNAME'),
16         user=config('MB_DB_USER'),
17         password=config('MB_DB_PASS'),
18         port=int(config('MB_DB_PORT')),
19     )
20     cur = conn_pg.cursor()
21     return jsonify({"status": 200, "db": "connected"})
22
23 if __name__ == "__main__":
24     app.run(debug=True, host="0.0.0.0")
```

## 7. Membuat Dockerfile diluar folder app



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure under 'PROJECTS': a folder named 'app' containing 'main.py', and files 'docker-compose.yml', 'Dockerfile', and 'requirement.txt'. The 'Dockerfile' file is selected. The top editor bar shows four open files: 'docker-compose.yml', 'requirement.txt', 'main.py', and 'Dockerfile'. The 'Dockerfile' editor displays the following content:

```
1 FROM python:3.7
2
3 COPY requirement.txt /requirement.txt
4
5 COPY ./app /app
6
7 WORKDIR /
8
9 RUN pip install -r requirement.txt
10
11 ENTRYPOINT ["python"]
12
13 CMD ["app/main.py", "--reload"]
```



## 8. Menjalankan Dockerfile

```
PS C:\Users\hallo\OneDrive\Documents\Kuliah\Stupen\project5> docker build -t project5 .
[+] Building 419.4s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 225B
=> [internal] load metadata for docker.io/library/python:3.7
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/python:3.7@sha256:eedf63967cdb57d8214db38ce21f105003ed4e4d0358f02bedc057341bcf92a0
=> => resolve docker.io/library/python:3.7@sha256:eedf63967cdb57d8214db38ce21f105003ed4e4d0358f02bedc057341bcf92a0
=> => sha256:16d93ae3411be3db255b6b52fd9c155a0dffa0f697c2e4e3d862caf8d978830b2 8.13kB / 8.13kB
=> => sha256:b47a222d28fa95680198398973d0a29b82a968f03e7ef361cc8ded562e4d84a3 24.03MB / 24.03MB
=> => sha256:debce5f9f3a9709885f7f2ad3cf41f036a3b57b406b27ba3a883928315787042 64.11MB / 64.11MB
=> => sha256:eedf63967cdb57d8214db38ce21f105003ed4e4d0358f02bedc057341bcf92a0 1.86kB / 1.86kB
=> => sha256:2011a37d2a08fe83dd9ff923e0f83bfd7290152e2e6afe359bde1453170d9bdc 2.01kB / 2.01kB
=> => sha256:167b8a53ca4504bc6aa3182e336fa96f4ef76875d158c1933d3e2fa19c57e0c3 49.56MB / 49.56MB
=> => sha256:1d7ca7cd2e066ae77ac6284a9d027f72a31a02a18bfc2a249ef2e7b01074338b 211.04MB / 211.04MB
=> => sha256:ff3119008f58beef8f336fa833707b0fe914db94ca6b7bb55abe3e1bf2b1ad56 6.39MB / 6.39MB
=> => extracting sha256:167b8a53ca4504bc6aa3182e336fa96f4ef76875d158c1933d3e2fa19c57e0c3
=> => extracting sha256:b47a222d28fa95680198398973d0a29b82a968f03e7ef361cc8ded562e4d84a3
=> => sha256:c2423a76a32b7ffb2ee7bb6d1e0c74bb1811237eddc3200594daf7a52d4f378 14.70MB / 14.70MB
=> => sha256:e1c98ca4926a91839805ce76d76a70225e303007331ee60f45dfabbbf55fd8c8 244B / 244B
=> => sha256:3b62c8e1d79b6554a8bffc196ff5dd822858c179f1f8dc6f0c74a288859a6fb 2.85MB / 2.85MB
=> => extracting sha256:debce5f9f3a9709885f7f2ad3cf41f036a3b57b406b27ba3a883928315787042
=> => extracting sha256:1d7ca7cd2e066ae77ac6284a9d027f72a31a02a18bfc2a249ef2e7b01074338b
=> => extracting sha256:ff3119008f58beef8f336fa833707b0fe914db94ca6b7bb55abe3e1bf2b1ad56
=> => extracting sha256:c2423a76a32b7ffb2ee7bb6d1e0c74bb1811237eddc3200594daf7a52d4f378
=> => extracting sha256:e1c98ca4926a91839805ce76d76a70225e303007331ee60f45dfabbbf55fd8c8
=> => extracting sha256:3b62c8e1d79b6554a8bffc196ff5dd822858c179f1f8dc6f0c74a288859a6fb
=> [internal] load build context
=> => transferring context: 782B
=> [2/5] COPY requirement.txt /requirement.txt
=> [3/5] COPY ./app /app
=> [4/5] RUN pip install -r requirement.txt
=> exporting to image
=> => exporting layers
=> => writing image sha256:0da99e2fa1650239d404189f7755750aa8a155deb5113d12e186f6931ec3c095
=> => naming to docker.io/library/project5

View build details: docker-desktop:///dashboard/build/default/default/nh9cwjjfnujfgve7h4bd1rnd

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
```

## 9. Mengecek docker images

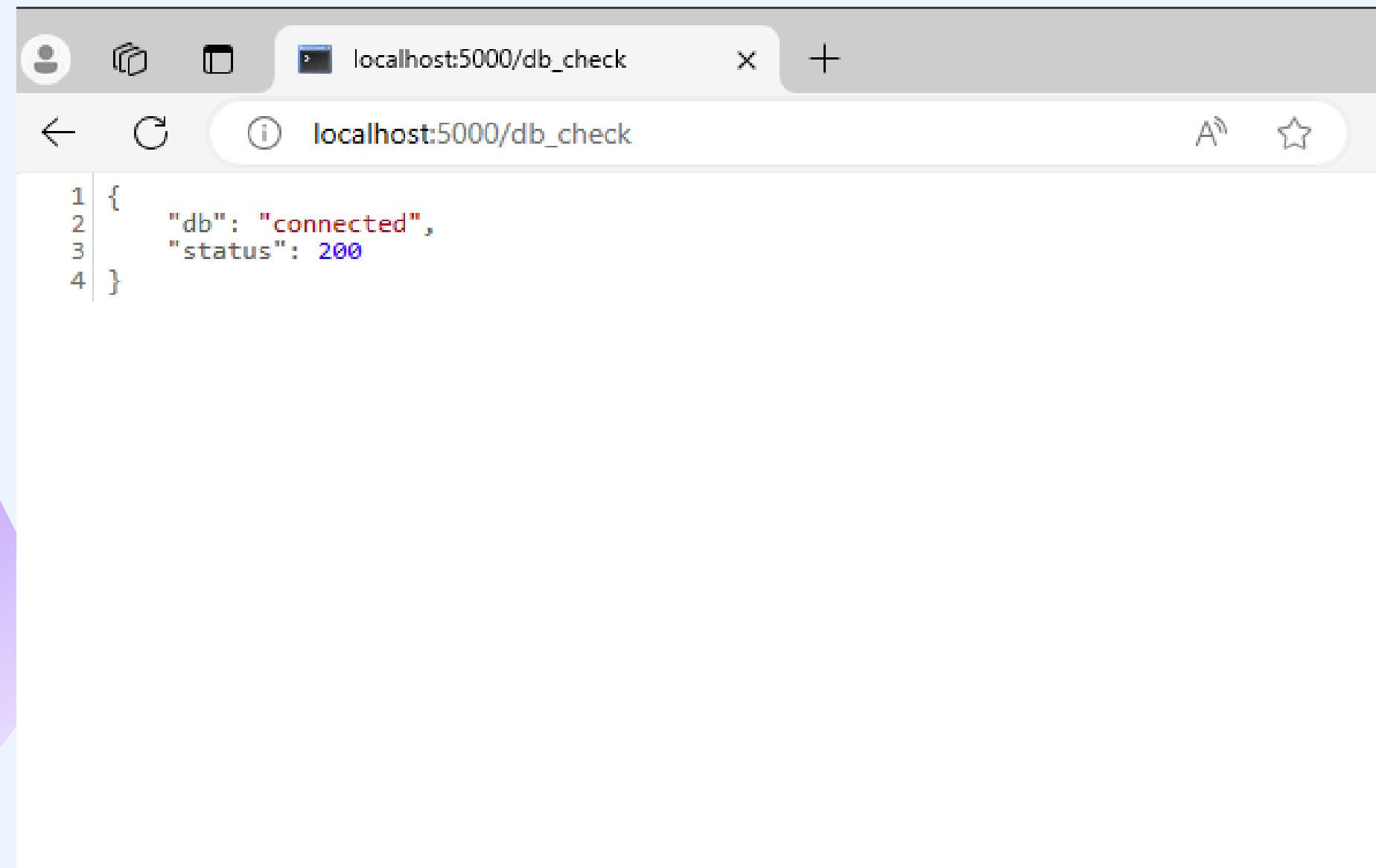
```
PS C:\Users\hallo\OneDrive\Documents\Kuliah\Stupen\project5> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
project5	latest	0da99e2fa165	13 seconds ago	1.03GB
mysql	8	6f343283ab56	5 weeks ago	632MB
postgres	latest	8e4fc9e18489	2 months ago	431MB
mysql	5.7	5107333e08a8	4 months ago	501MB
postgres	11	60a93af0bba5	22 months ago	284MB

## 10. Menjalankan ulang dockernya

```
PS C:\Users\hallo\OneDrive\Documents\Kuliah\Stupen\project5> docker-compose down
[+] Running 3/3
 ✓ Container project5-flask-app-1    Removed
 ✓ Container project5-postgres-db-1  Removed
 ✓ Network project5_default          Removed
PS C:\Users\hallo\OneDrive\Documents\Kuliah\Stupen\project5> docker-compose up -d
[+] Running 2/3
 - Network project5_default          Created
 ✓ Container project5-postgres-db-1  Started
 ✓ Container project5-flask-app-1     Started
```

## 11. Mengecek connection database dari server local



A screenshot of a web browser window. The address bar shows 'localhost:5000/db\_check'. The page content displays a JSON object indicating a successful database connection.

```
1 {  
2   "db": "connected",  
3   "status": 200  
4 }
```

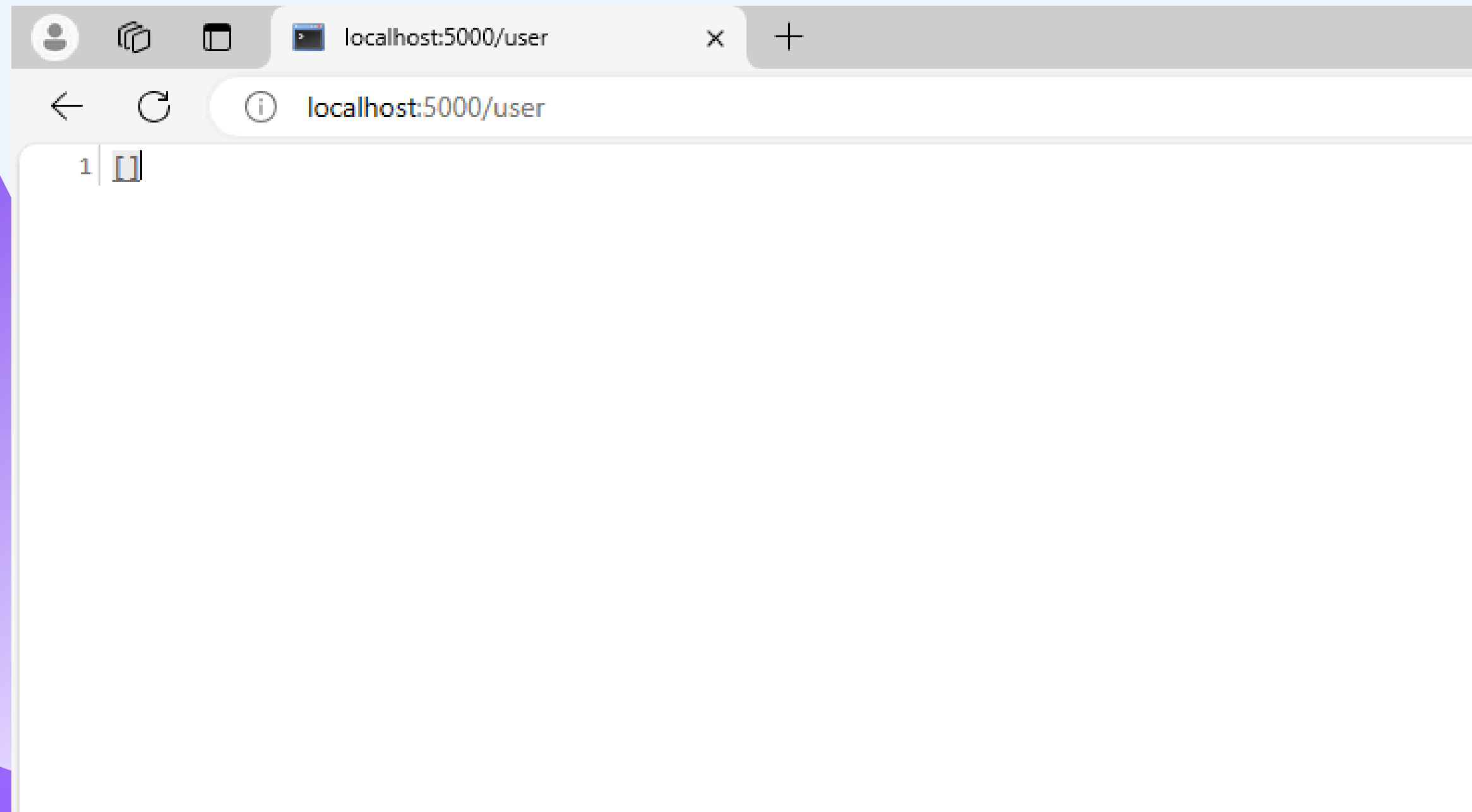
## 12. Menambahkan fungsi CRUD API pada main.py

```
app > main.py > user
1  from flask import Flask, jsonify, request
2  from decouple import config
3  import psycopg2
4  from flask_sqlalchemy import SQLAlchemy
5
6
7  DB_URI = f"postgresql+psycopg2://{config('MB_DB_USER')}:{config('MB_DB_PASS')}@{config('MB_DB_HOST')}:{str(config('MB_DB_PORT'))}/{config('MB_DB_DBNAME')}"
8  app = Flask(__name__)
9  app.config["SQLALCHEMY_DATABASE_URI"] = DB_URI
10 app.config["SQLALCHEMY_TRACK_MODIFICATIONS"] = True
11 db = SQLAlchemy(app)
12
13 class Users(db.Model):
14     id = db.Column('user_id', db.Integer, primary_key=True)
15     name = db.Column(db.String(100))
16     city = db.Column(db.String(50))
17     telp = db.Column(db.String(14))
18
19 @app.route("/health")
20 > def health(): ...
21
22
23 @app.route("/db_check")
24 > def db_check(): ...
25
26
27
28
29
30
31
32
33
34
35 @app.route("/user", methods=["GET", "POST", "PUT", "DELETE"])
36 def user():
37     if request.method == 'GET':
38         users = Users.query.all()
39         results = [{"id": u.id, "name": u.name, "city": u.city, "telp": u.telp} for u in users]
40         return jsonify(results)
41     elif request.method == 'POST':
42         user = Users(
43             name=request.form['name'],
44             city=request.form['city'],
45             telp=request.form['telp']
46         )
47         db.session.add(user)
48         db.session.commit()
49         return jsonify({"status": "ok"})
50     elif request.method == 'PUT':
51         pass
52     else:
53         return 'Method not Allowed'
54
55 if __name__ == "__main__":
56     with app.app_context():
57         db.create_all()
58     app.run(debug=True, host="0.0.0.0")
```

## 13. Menjalankan docker kembali

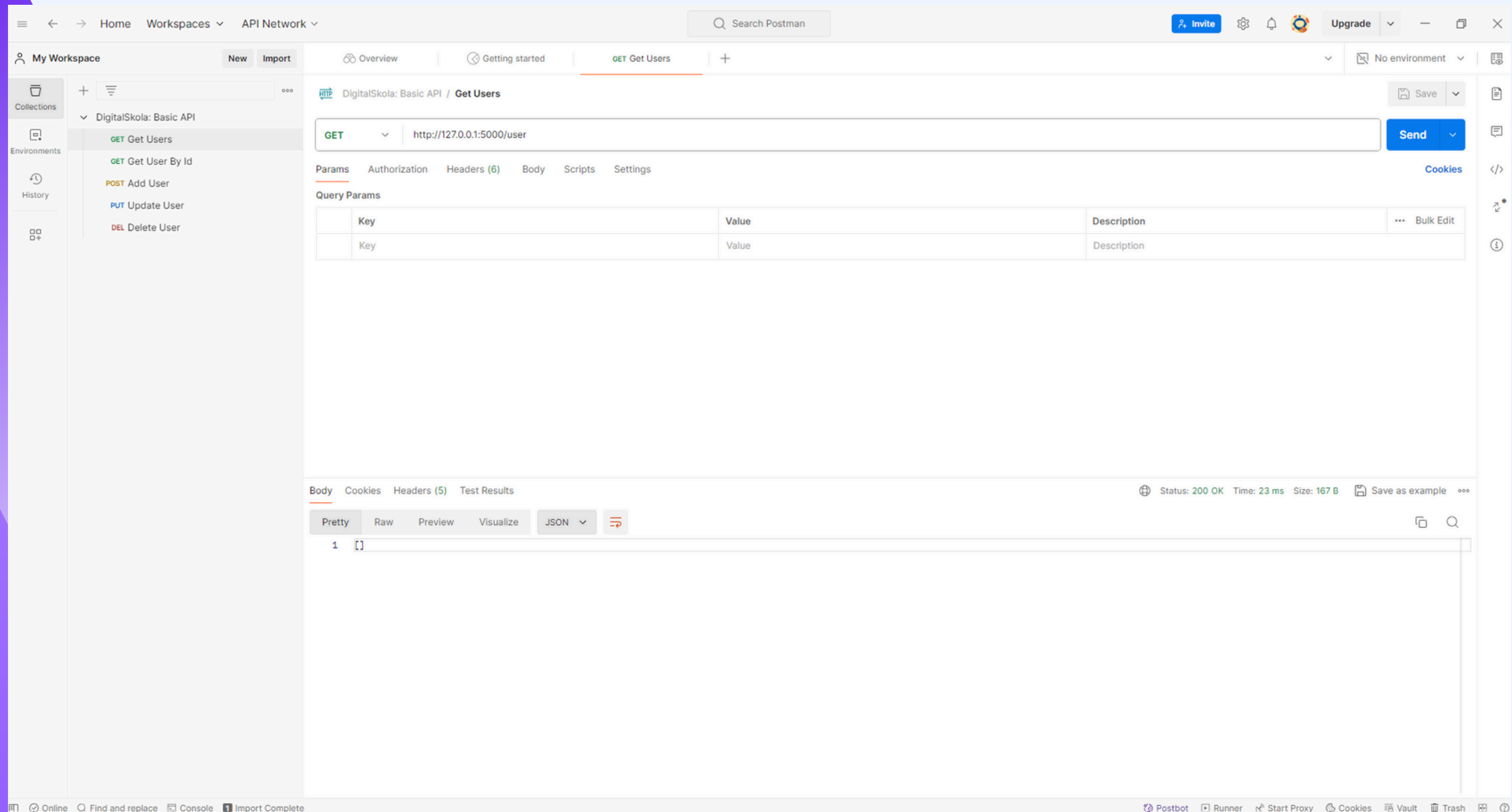
```
PS C:\Users\hallo\OneDrive\Documents\Kuliah\Stupen\project5> docker-compose down
[+] Running 3/3
✓ Container project5-flask-app-1    Removed
✓ Container project5-postgres-db-1  Removed
✓ Network project5_default          Removed
PS C:\Users\hallo\OneDrive\Documents\Kuliah\Stupen\project5> docker-compose up -d
[+] Running 2/3
- Network project5_default          Created
✓ Container project5-postgres-db-1  Started
✓ Container project5-flask-app-1     Started
PS C:\Users\hallo\OneDrive\Documents\Kuliah\Stupen\project5> docker container ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
c9f5890c5f64   project5   "python app/main.py ..." 5 seconds ago  Up 4 seconds  0.0.0.0:5000->5000/tcp             project5-flask-app-1
ab9619baacc3   postgres  "docker-entrypoint.s..." 5 seconds ago  Up 4 seconds  0.0.0.0:5432->5432/tcp             project5-postgres-db-1
PS C:\Users\hallo\OneDrive\Documents\Kuliah\Stupen\project5>
```

## 14. Mengecek koneksi user di server local





# 15. Membuka Postman





## 16. Menambahkan data user pada Add user dan menampilkan data user baru di Get user

The screenshot shows the Postman interface with a workspace named 'My Workspace'. The left sidebar lists the API collection 'DigitalSkola: Basic API' with endpoints: GET Get Users, GET Get User By Id, POST Add User (selected), PUT Update User, and DEL Delete User. The main panel displays the 'POST Add User' endpoint with the URL 'http://127.0.0.1:5000/user'. The 'Body' tab is active, showing form data with the following details:

Key	Value	Description
name	Kakang	
city	Solo	
telp	089762634343	
Key	Value	Description

The response is shown in the 'Body' tab at the bottom, indicating a 200 OK status with a response time of 32 ms and 186 B of data. The JSON response is:

```
1 {  
2   "status": "ok"  
3 }
```

The screenshot shows the Postman interface with the same workspace. The left sidebar lists the API collection 'DigitalSkola: Basic API' with endpoints: GET Get Users (selected), GET Get User By Id, POST Add User, PUT Update User, and DEL Delete User. The main panel displays the 'GET Get Users' endpoint with the URL 'http://127.0.0.1:5000/user'. The 'Body' tab is active, showing the response details:

Key	Value	Description
Key	Value	Description

The response is shown in the 'Body' tab at the bottom, indicating a 200 OK status with a response time of 12 ms and 259 B of data. The JSON response is:

```
1 [  
2   {  
3     "city": "Solo",  
4     "id": 1,  
5     "name": "Kakang",  
6     "telp": "089762634343"  
7   }  
8 ]
```

## 17. Memperbarui data user pada Update user dan menampilkan data terbaru pada Get user

The screenshot shows the Postman interface with a workspace named "My Workspace". The left sidebar lists the "DigitalSkola: Basic API" collection, with the "PUT Update User" endpoint selected. The main panel displays the "PUT" method with the URL "http://127.0.0.1:5000/user?=1". The "Query Params" section is empty. The "Body" tab is active, showing a JSON response in "Pretty" format:

```
1 {
2   "message": "User with ID 3 updated successfully",
3   "status": "ok"
4 }
```

The status bar at the bottom indicates a 200 OK response with a 16 ms latency and 238 B of data.

The screenshot shows the Postman interface with the same workspace. The left sidebar lists the "DigitalSkola: Basic API" collection, with the "GET Get Users" endpoint selected. The main panel displays the "GET" method with the URL "http://127.0.0.1:5000/user". The "Query Params" section is empty. The "Body" tab is active, showing a JSON response in "Pretty" format:

```
1 [
2   {
3     "city": "Solo",
4     "id": 1,
5     "name": "Kakang",
6     "telp": "089762634343"
7   },
8   {
9     "city": "Depok",
10    "id": 3,
11    "name": "Lalala",
12    "telp": "089762634343"
13  }
14 ]
```

The status bar at the bottom indicates a 200 OK response with an 8 ms latency and 352 B of data.

## 18. Menghapus data user di Delete User dan menampilkan hasil terbaru di Get user

The screenshot shows the Postman interface with a workspace named "My Workspace". The left sidebar lists the collections, with "DigitalSkola: Basic API" expanded. The main panel displays a DELETE request to "http://127.0.0.1:5000/user?=3". The "Params" tab is active, showing a query parameter "Key" with a value of "3". The "Body" tab is selected, showing a JSON response: 

```
{  "status": "ok"}
```

. The status bar indicates a 200 OK response with a 30 ms response time and 186 B of data.

My Workspace

New Import

Overview Getting started GET Get Users POST Add User DEL Delete User + No environment

DigitalSkola: Basic API / Delete User

DELETE http://127.0.0.1:5000/user?=3 Send

Params Authorization Headers (8) Body Scripts Settings Cookies

Query Params

Key	Value	Description
3		

Body Cookies Headers (5) Test Results

200 OK 30 ms 186 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "ok"
3 }
```

The screenshot shows the Postman interface with a workspace named "My Workspace". The left sidebar lists the collections, with "DigitalSkola: Basic API" expanded. The main panel displays a GET request to "http://127.0.0.1:5000/user". The "Params" tab is active, showing a query parameter "Key" with a value of "3". The "Body" tab is selected, showing a JSON response: 

```
[  {    "city": "Solo",    "id": 1,    "name": "Kakang",    "tel": "089762634343"  },  {    "city": "wonogiri",    "id": 3,    "name": "Joko Sindi",    "tel": "0898232634378"  }]
```

. The status bar indicates a 200 OK response with a 11 ms response time and 360 B of data.

My Workspace

New Import

Overview Getting started GET Get Users POST Add User DEL Delete User + No environment

DigitalSkola: Basic API / Get Users

GET http://127.0.0.1:5000/user Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description
3		

Body Cookies Headers (5) Test Results

200 OK 11 ms 360 B Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "city": "Solo",
4     "id": 1,
5     "name": "Kakang",
6     "tel": "089762634343"
7   },
8   {
9     "city": "wonogiri",
10    "id": 3,
11    "name": "Joko Sindi",
12    "tel": "0898232634378"
13  }
14 ]
```

## 19. Menampilkan user berdasarkan id tertentu pada Get User By Id

The screenshot shows the Postman interface with a workspace named 'My Workspace'. On the left, a collection named 'DigitalSkola: Basic API' is expanded, showing several endpoints. The 'Get User By Id' endpoint is selected. The main panel shows the details of this endpoint, including the method 'GET' and the URL 'http://127.0.0.1:5000/user/2'. The 'Send' button is visible. Below the URL bar, the 'Query Params' section is empty. The 'Body' tab is selected, showing a JSON response with an error message: 

```
{  "error": "id not found"}
```

. The status bar at the bottom indicates a 200 OK response with a 14 ms response time and 195 B of data.

My Workspace New Import

Collections + ...

DigitalSkola: Basic API

- GET Get Users
- GET Get User By Id
- POST Add User
- PUT Update User
- DEL Delete User

Environments

History

Overview Getting Started GET Get Us POST Add DEL Delete PUT Update GET Get U + - No environment -

DigitalSkola: Basic API / Get User By Id Save -

GET - http://127.0.0.1:5000/user/2 Send -

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

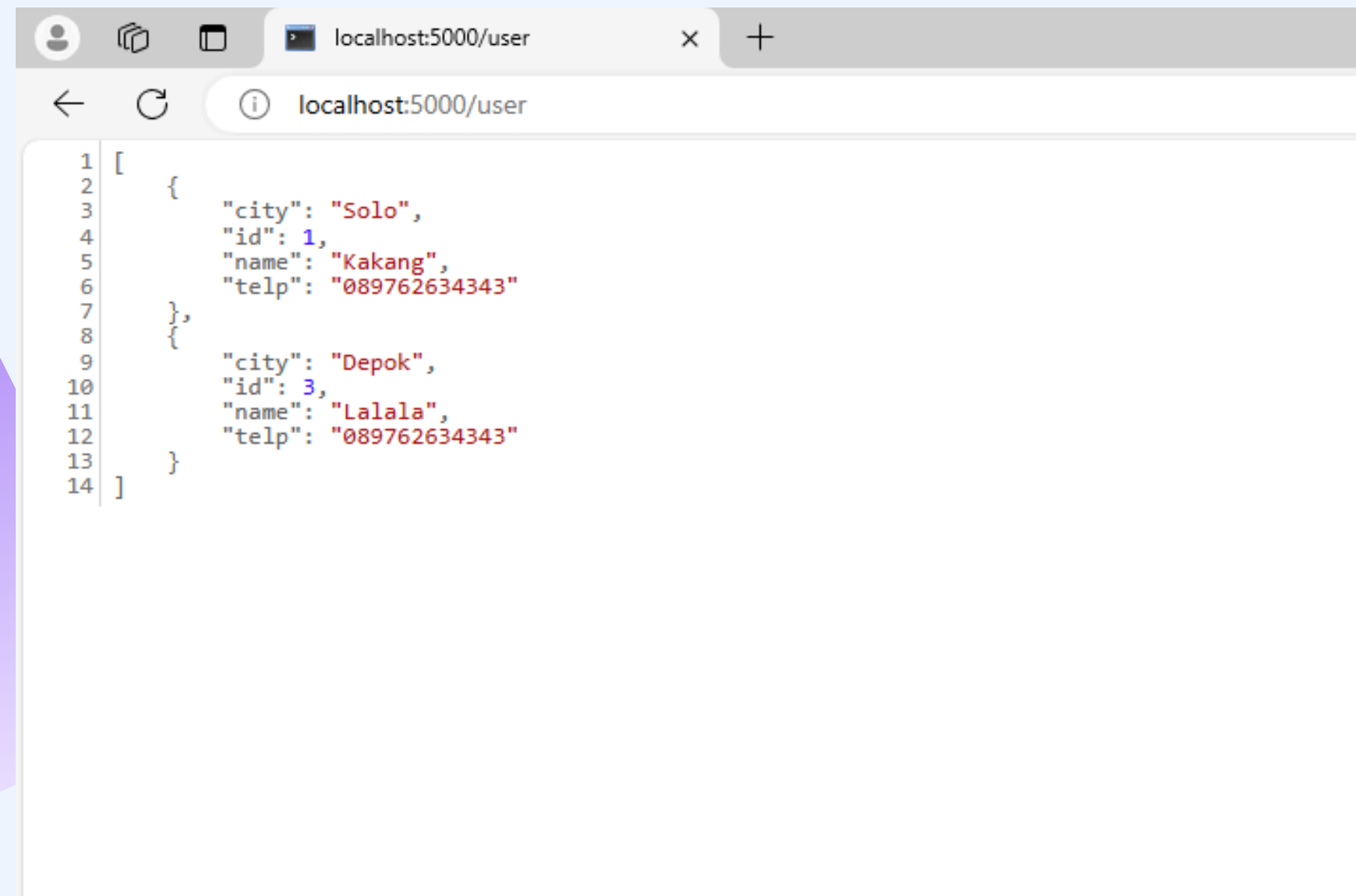
	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 14 ms 195 B Save as example ...

Pretty Raw Preview Visualize JSON - ...

```
1 {
2   "error": "id not found"
3 }
```

## 20. Menampilkan hasil data dari CRUD API di Postman pada server local

A screenshot of a web browser window. The address bar shows 'localhost:5000/user'. The page content displays a JSON array of two user objects. The first object has 'city': 'Solo', 'id': 1, 'name': 'Kakang', and 'telp': '089762634343'. The second object has 'city': 'Depok', 'id': 3, 'name': 'Lalala', and 'telp': '089762634343'. Line numbers 1 through 14 are visible on the left side of the code area.

```
1  [  
2    {  
3      "city": "Solo",  
4      "id": 1,  
5      "name": "Kakang",  
6      "telp": "089762634343"  
7    },  
8    {  
9      "city": "Depok",  
10     "id": 3,  
11     "name": "Lalala",  
12     "telp": "089762634343"  
13   }  
14 ]
```