

NAMA: MUHAMMAD QODRAT HANIF FADHILA

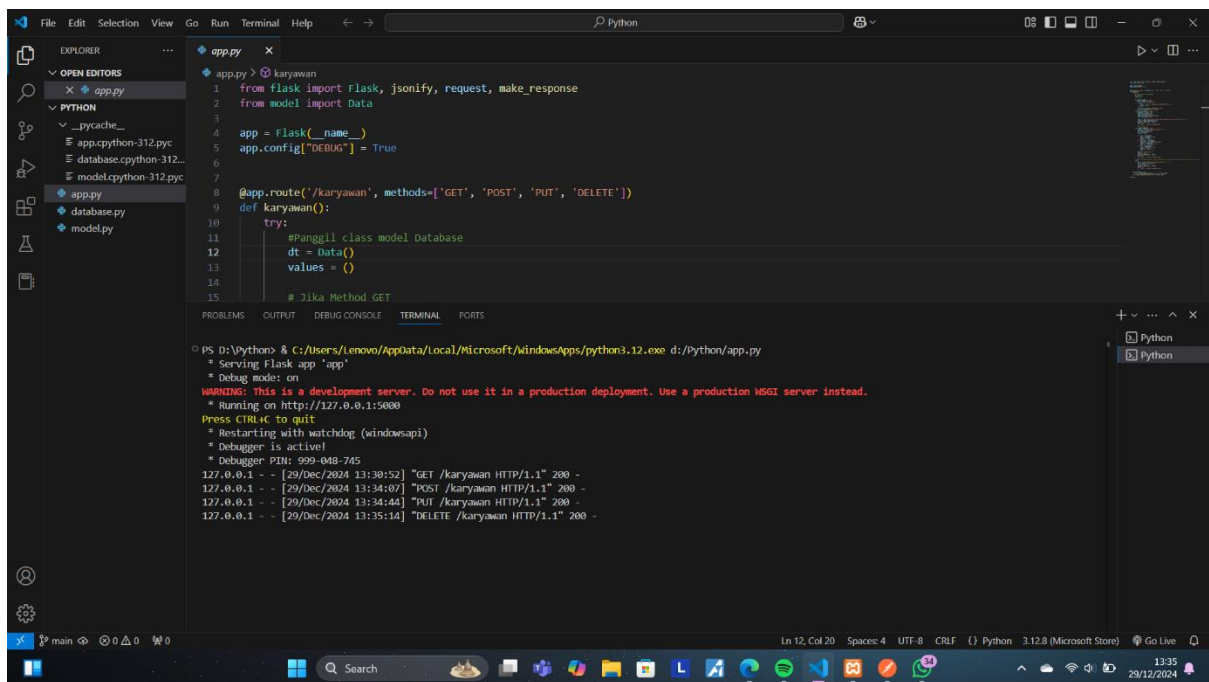
NIM: 230741110

PRODI: ILMU KOMPUTER

MATA KULIAH: KOMPUTASI PARAREL TERDISTRIBUSI

DOKUMENTASI KPT

APP.PY



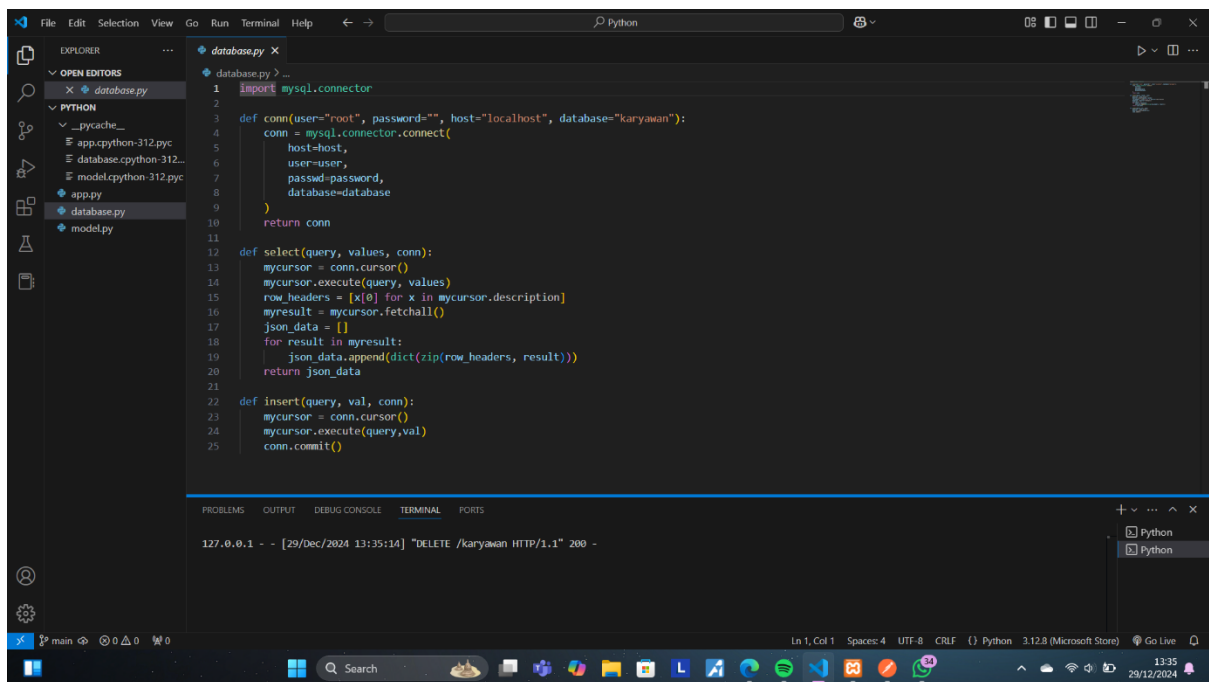
The screenshot displays the Visual Studio Code (VS Code) interface. The Explorer sidebar on the left shows a project structure with files: `app.py`, `database.py`, and `model.py`. The main editor window is open to `app.py`, which contains the following code:

```
1 from flask import Flask, jsonify, request, make_response
2 from model import Data
3
4 app = Flask(__name__)
5 app.config["DEBUG"] = True
6
7
8 @app.route('/karyawan', methods=['GET', 'POST', 'PUT', 'DELETE'])
9 def karyawan():
10     try:
11         # Panggil class model Database
12         dt = Data()
13         values = {}
14
15         # Jika Method GET
```

The TERMINAL panel at the bottom shows the command prompt output for running the application:

```
PS D:\Python> C:\Users\Lenovo\AppData\Local\Microsoft\WindowsApps\python3.12.exe d:/Python/app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 999-048-745
127.0.0.1 - - [29/Dec/2024 13:30:52] "GET /karyawan HTTP/1.1" 200 -
127.0.0.1 - - [29/Dec/2024 13:34:07] "POST /karyawan HTTP/1.1" 200 -
127.0.0.1 - - [29/Dec/2024 13:34:44] "PUT /karyawan HTTP/1.1" 200 -
127.0.0.1 - - [29/Dec/2024 13:35:14] "DELETE /karyawan HTTP/1.1" 200 -
```

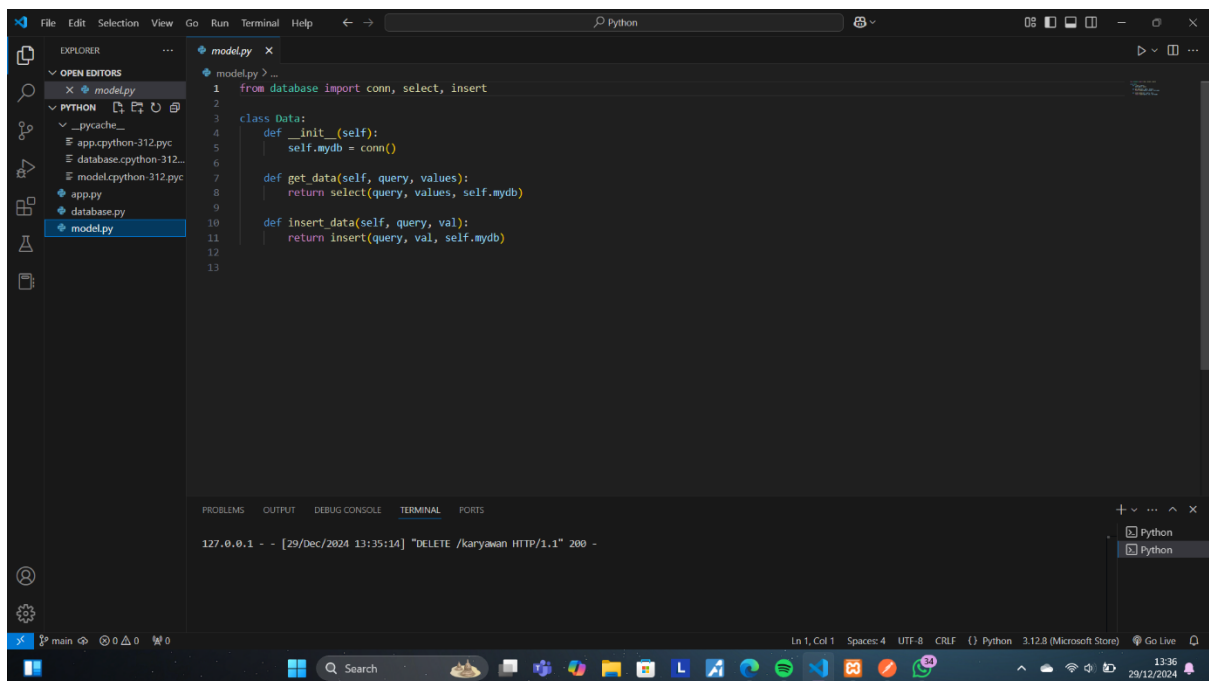
DATABASE.PY



```
1 import mysql.connector
2
3 def conn(user="root", password="", host="localhost", database="karyawan"):
4     conn = mysql.connector.connect(
5         host=host,
6         user=user,
7         passwd=password,
8         database=database
9     )
10    return conn
11
12 def select(query, values, conn):
13     mycursor = conn.cursor()
14     mycursor.execute(query, values)
15     row_headers = [x[0] for x in mycursor.description]
16     myresult = mycursor.fetchall()
17     json_data = []
18     for result in myresult:
19         json_data.append(dict(zip(row_headers, result)))
20    return json_data
21
22 def insert(query, val, conn):
23     mycursor = conn.cursor()
24     mycursor.execute(query, val)
25     conn.commit()
```

127.0.0.1 - - [29/Dec/2024 13:35:14] "DELETE /karyawan HTTP/1.1" 200 -

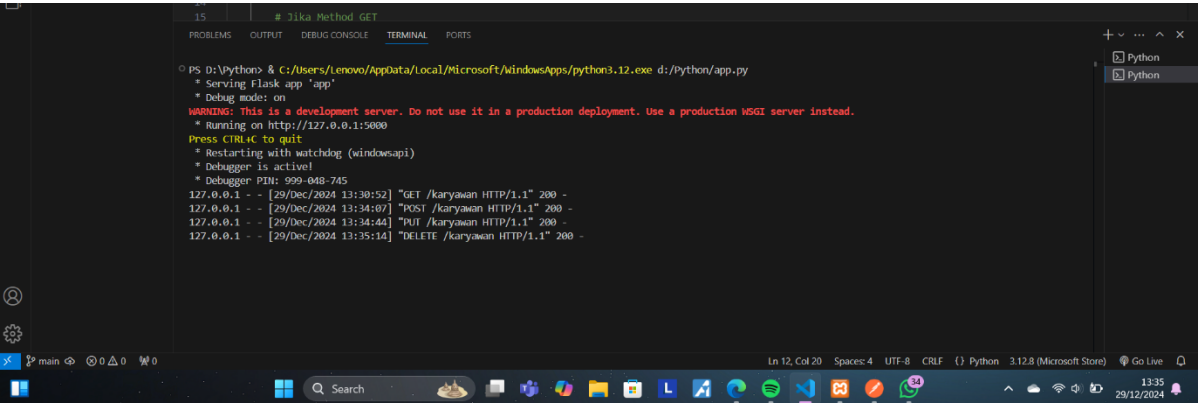
MODEL.PY



```
1 from database import conn, select, insert
2
3 class Data:
4     def __init__(self):
5         self.mydb = conn()
6
7     def get_data(self, query, values):
8         return select(query, values, self.mydb)
9
10    def insert_data(self, query, val):
11        return insert(query, val, self.mydb)
12
13
```

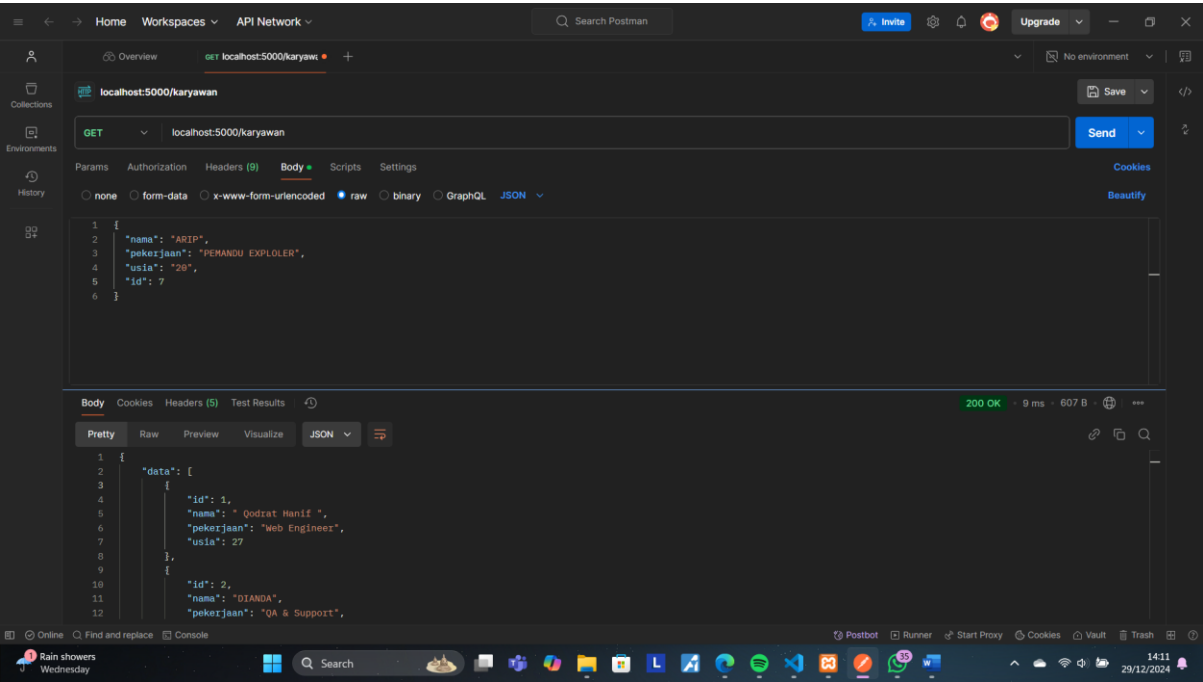
127.0.0.1 - - [29/Dec/2024 13:35:14] "DELETE /karyawan HTTP/1.1" 200 -

MENJALANKAN APP.PY

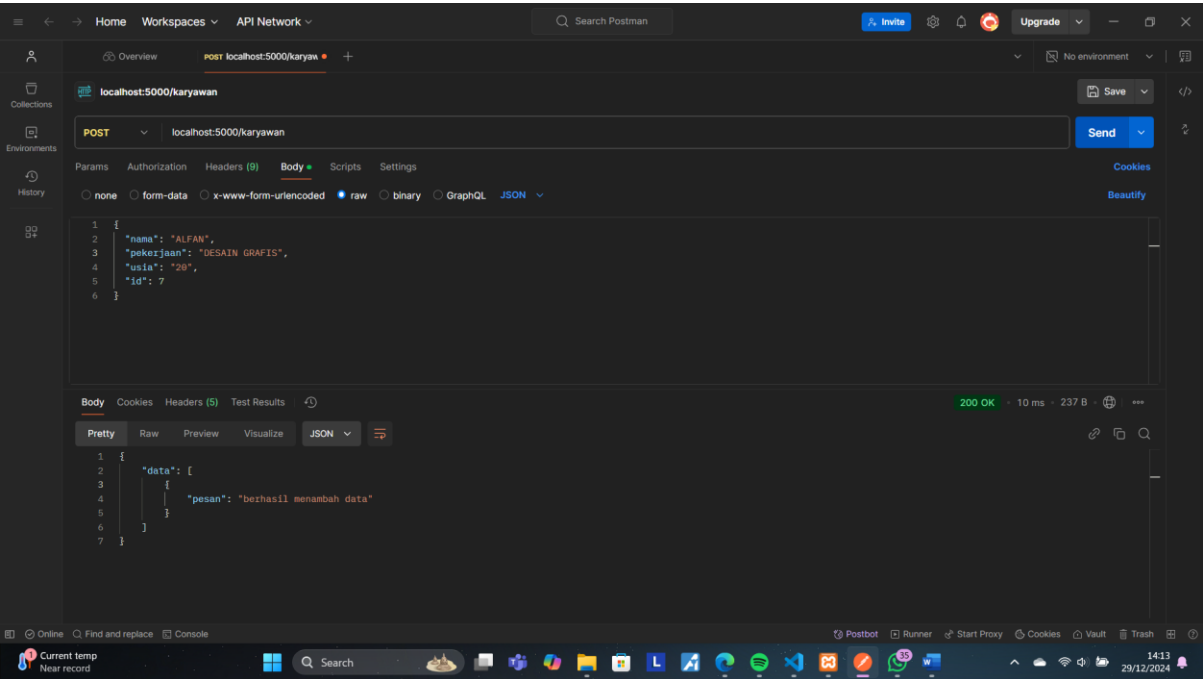


POSTMAN

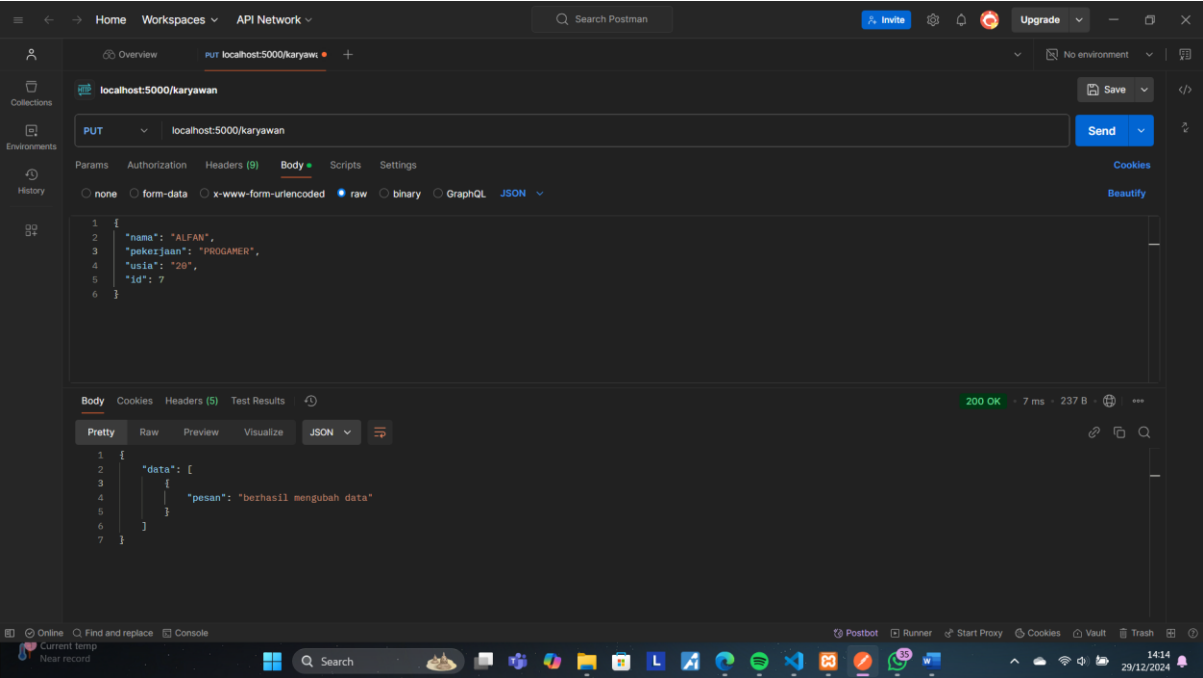
GET



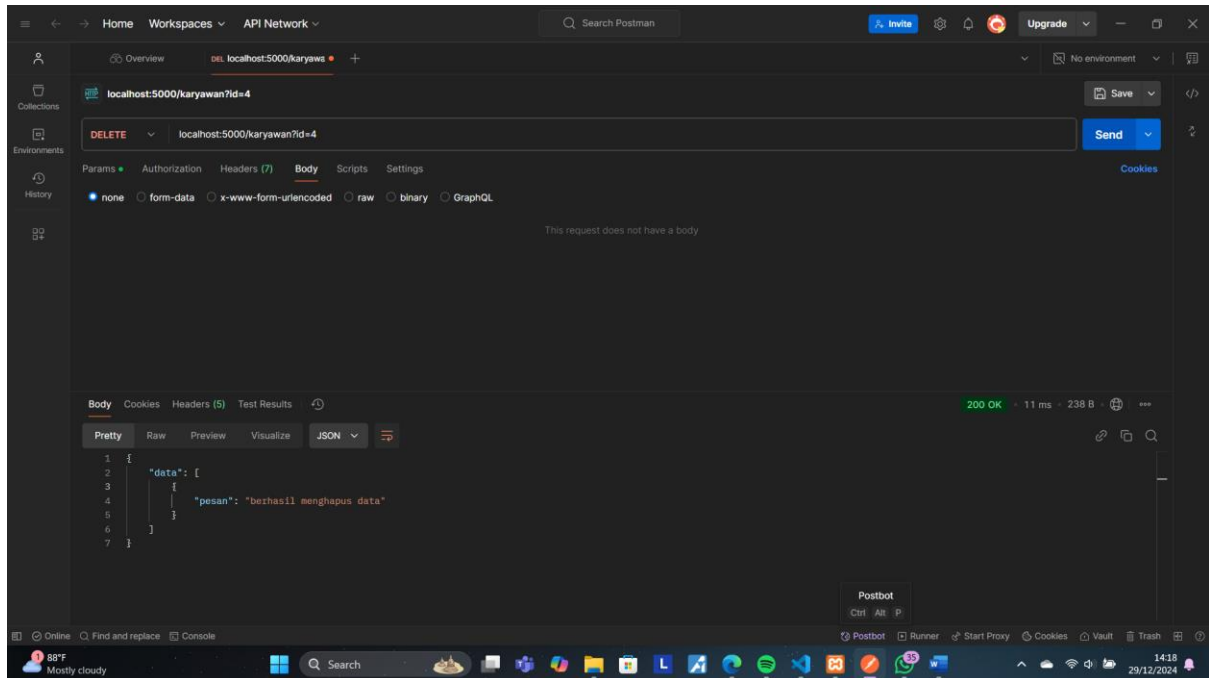
POST



PUT



DELETE



DEMIKIAN DOKUMENTASI MENJALANKAN POSTMAN DIWA DENGAN METODE GET, POS, PUT, DELETE. SEKIAN ATAS PERHATIANNYA .

TERIMA KASIH