

NAMA: DIWA MULKI

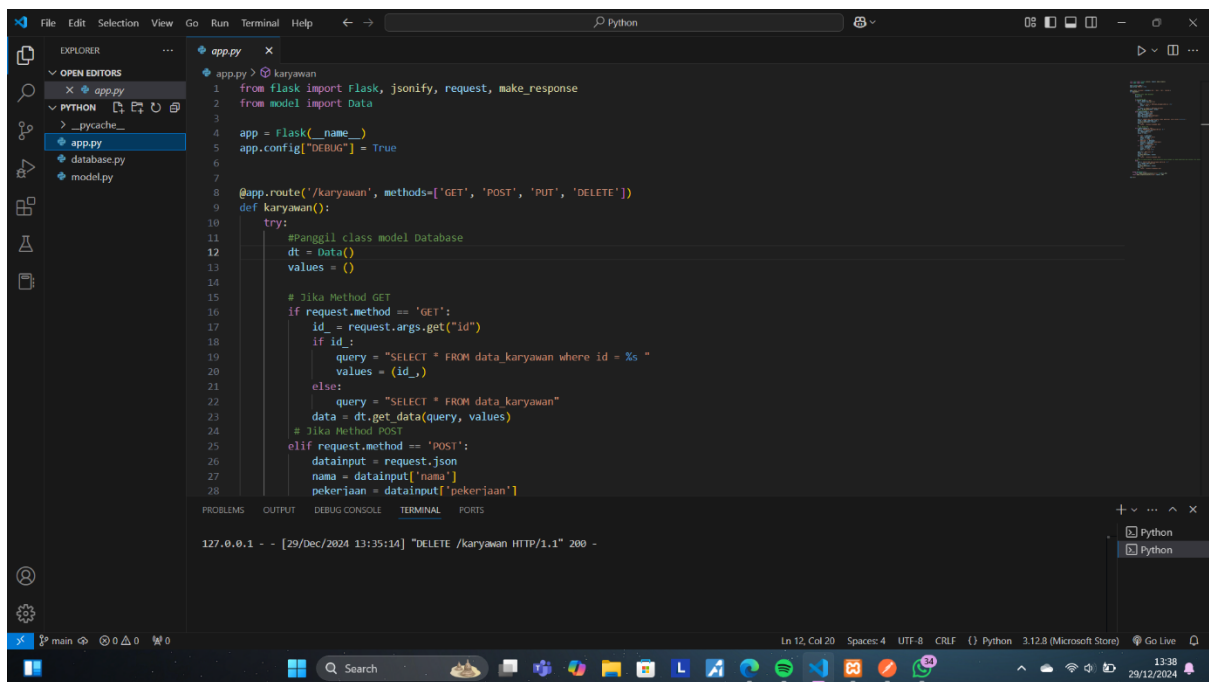
NIM: 230741112

PRODI: ILMU KOMPUTER

MATA KULIAH: KOMPUTASI PARAREL TERDISTRIBUSI

DOKUMENTASI KPT

APP.PY



```
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
16         if request.method == 'GET':
17             id_ = request.args.get("id")
18             if id_:
19                 query = "SELECT * FROM data_karyawan where id = %s "
20                 values = (id_,)
21             else:
22                 query = "SELECT * FROM data_karyawan"
23             data = dt.get_data(query, values)
24
25         # Jika Method POST
26         elif request.method == 'POST':
27             datainput = request.json
28             nama = datainput['nama']
29             pekerjaan = datainput['pekerjaan']
```

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

This screenshot shows the VS Code editor with the file `app.py` open. The code defines a `karyawan()` function that handles HTTP requests. It uses `data.insert_data(query, values)` for INSERT and `dt.insert_data(query, values)` for UPDATE. The terminal shows a `DELETE /karyawan HTTP/1.1 200 -` message.

```
def karyawan():
    usia = datainput['usia']
    query = "INSERT INTO data_karyawan (nama, pekerjaan, usia) values (%s,%s,%s) "
    values = (nama, pekerjaan, usia,)
    dt.insert_data(query, values)
    data = [{
        'pesan': 'berhasil menambah data'
    }]
    # Jika Method PUT
    elif request.method == 'PUT':
        query = "UPDATE data_karyawan SET id = %s "
        datainput = request.json
        id_ = datainput['id']
        values += (id_,)

        if 'nama' in datainput:
            nama = datainput['nama']
            values += (nama,)
            query += ", nama = %s"
        if 'pekerjaan' in datainput:
            pekerjaan = datainput['pekerjaan']
            values += (pekerjaan,)
            query += ", pekerjaan = %s"
        if 'usia' in datainput:
            usia = datainput['usia']
            values += (usia,)
            query += ", usia = %s"
```

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

This screenshot shows the VS Code editor with the file `app.py` open. The code continues the `karyawan()` function with a DELETE logic using `dt.insert_data(query, values)`. It includes an `except Exception as e:` block for error handling. The terminal shows a `DELETE /karyawan HTTP/1.1 200 -` message.

```
values += (usia,)
query += ", usia = %s"

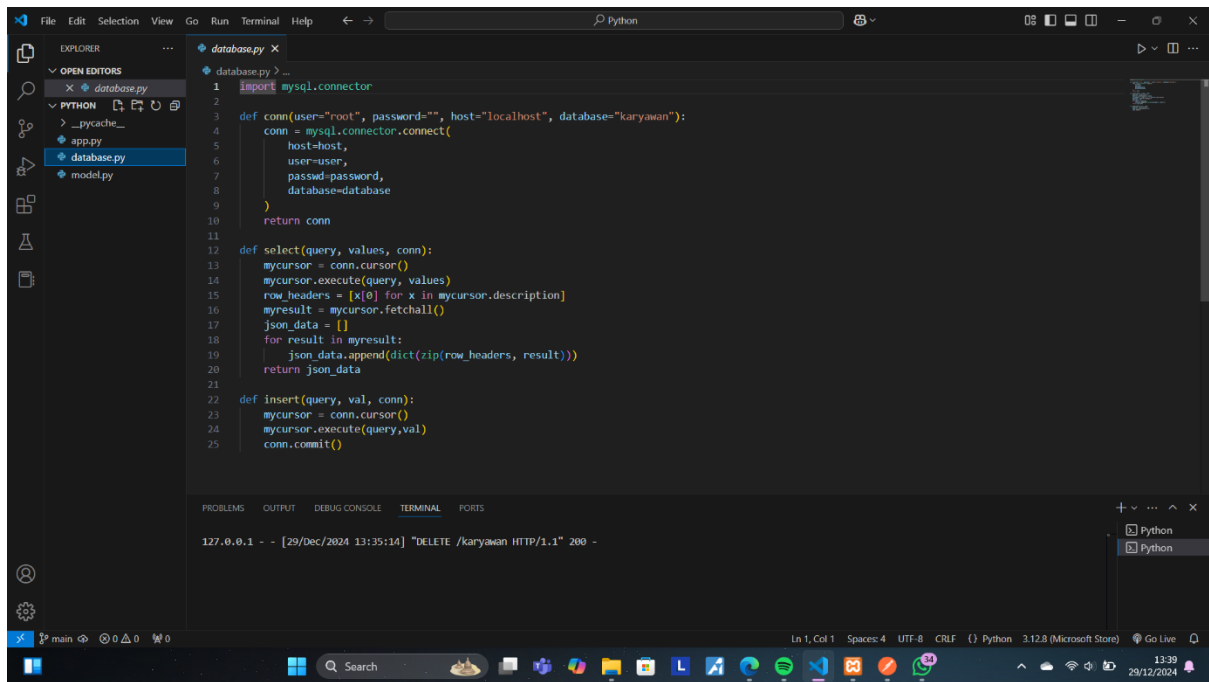
query += " where id = %s"
values += (id_,)
dt.insert_data(query, values)
data = [{
    'pesan': 'berhasil mengubah data'
}]
# Selain itu adalah DELETE , Bila ada Method selain keempat ini maka dipastikan akan langsung error karena method tidak di assign
else:
    query = "DELETE FROM data_karyawan where id = %s "
    id_ = request.args.get("id")
    values = (id_,)
    dt.insert_data(query, values)
    data = [{
        'pesan': 'berhasil menghapus data'
    }]

except Exception as e:
    return make_response(jsonify({'error':str(e)}),400)
return make_response(jsonify({'data': data}), 200)

app.run()
```

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

DATABASE.PY



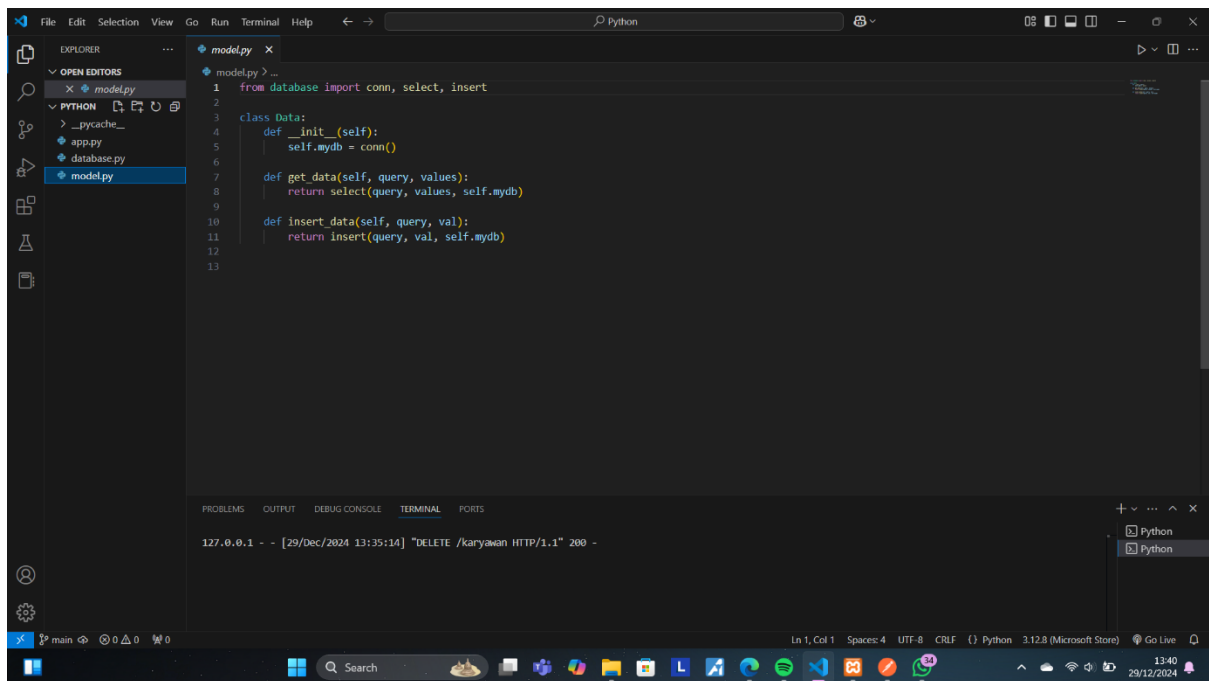
The screenshot shows the Visual Studio Code editor with a Python file named `database.py` open. The code defines a `conn` function for MySQL connection, a `select` function for querying data, and an `insert` function for inserting data. The terminal at the bottom shows a command prompt for a remote connection to 127.0.0.1.

```
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()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

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

MODEL.PY



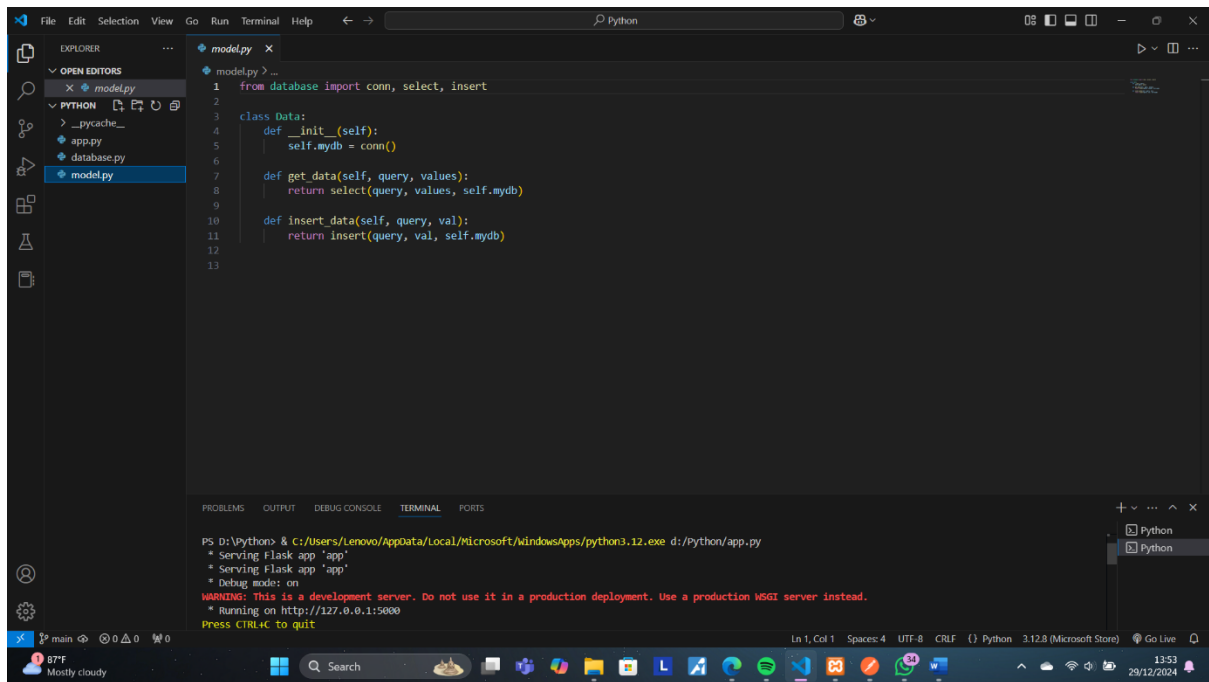
The screenshot shows the Visual Studio Code editor with a Python file named `model.py` open. The code defines a `Data` class that uses the `conn`, `select`, and `insert` functions from the `database` module. The terminal at the bottom shows the same command prompt as the previous screenshot.

```
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
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

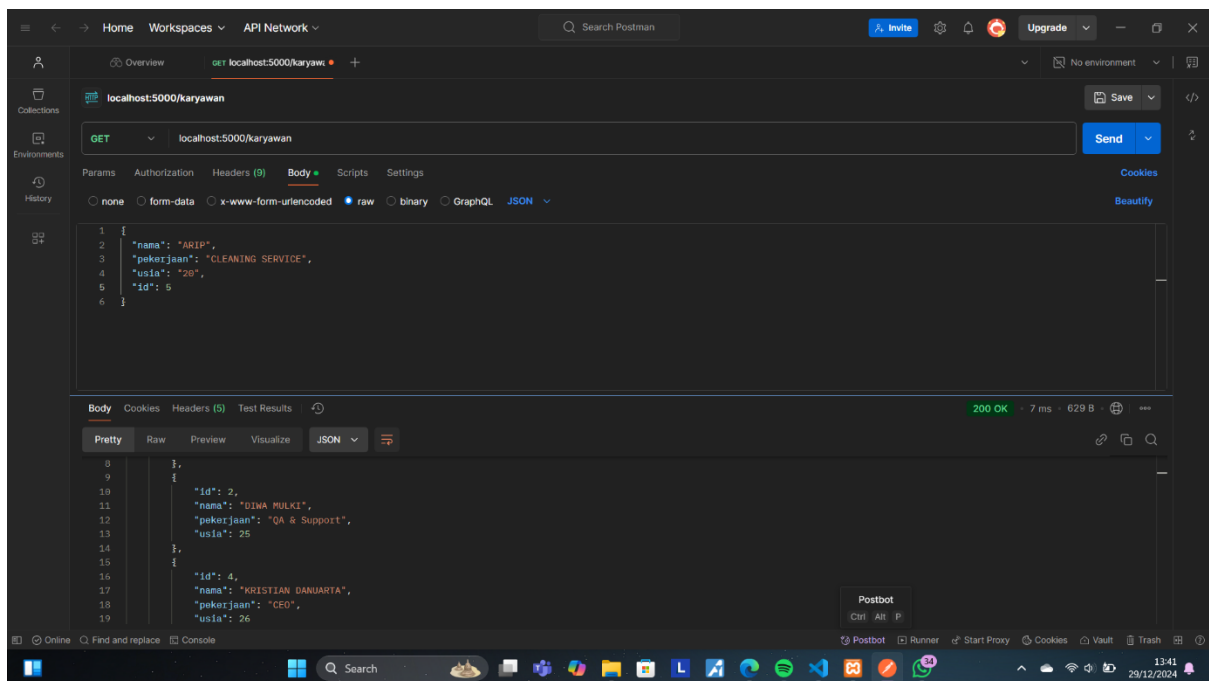
127.0.0.1 - - [29/Dec/2024 13:35:14] "DELETE /karyawan HTTP/1.1" 200 -

MENJALANKAN APP.PY

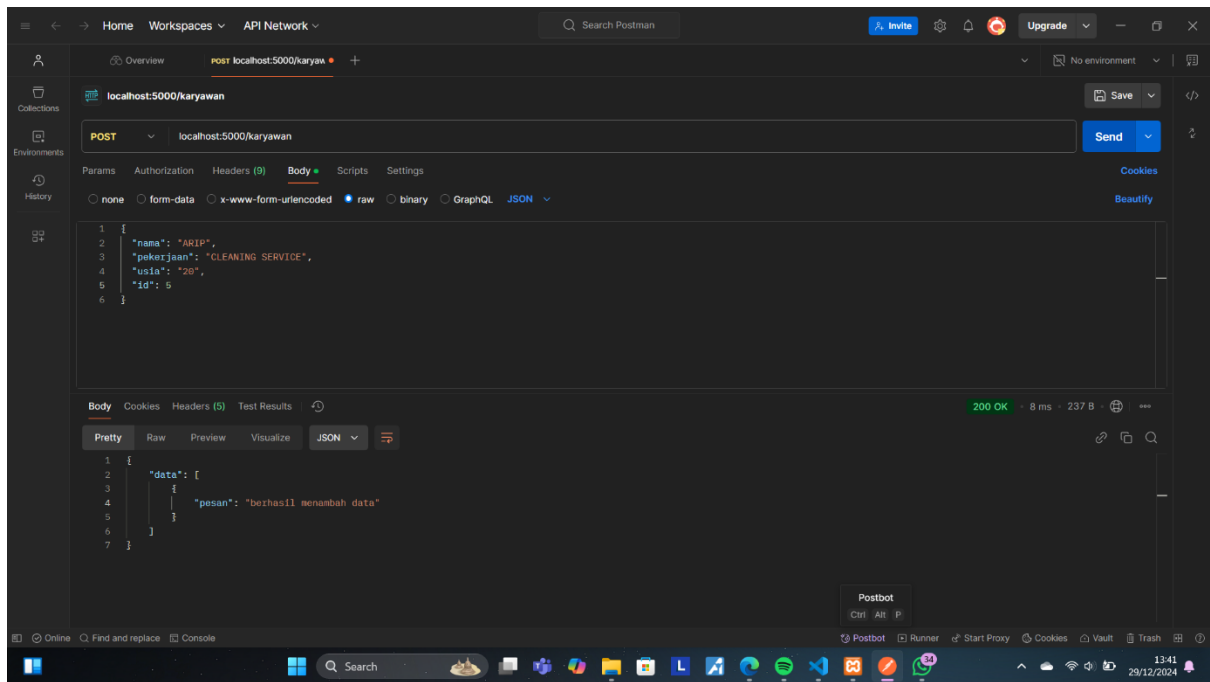


POSTMAN

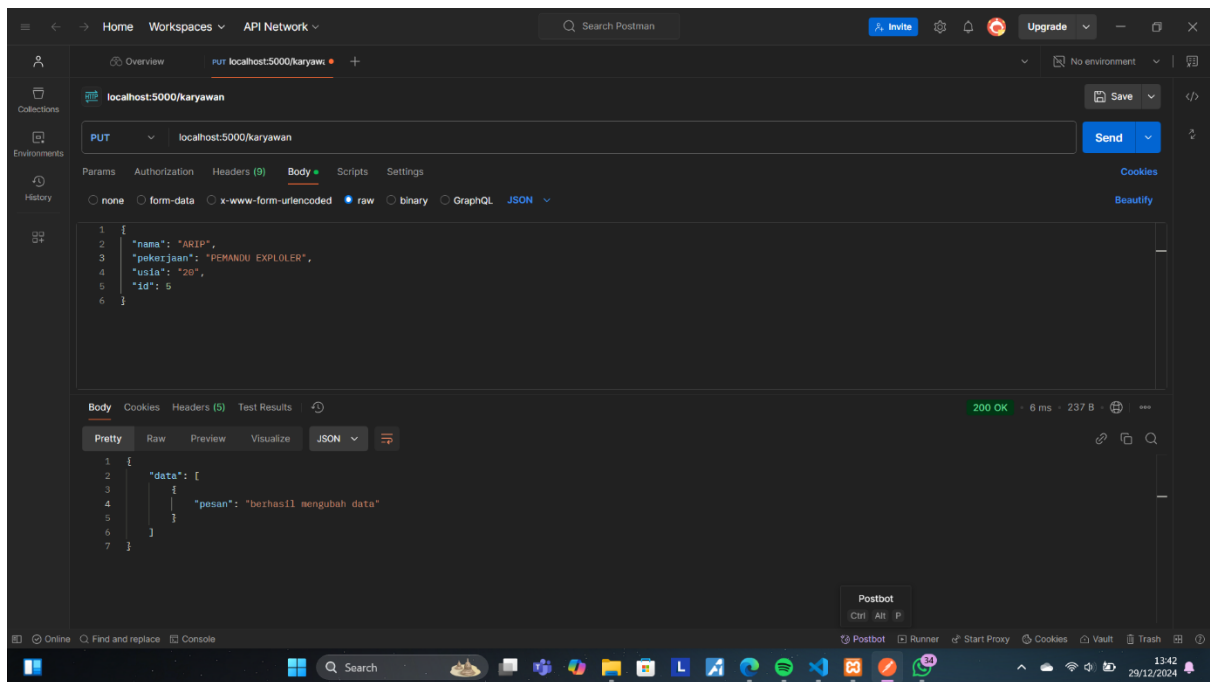
GET



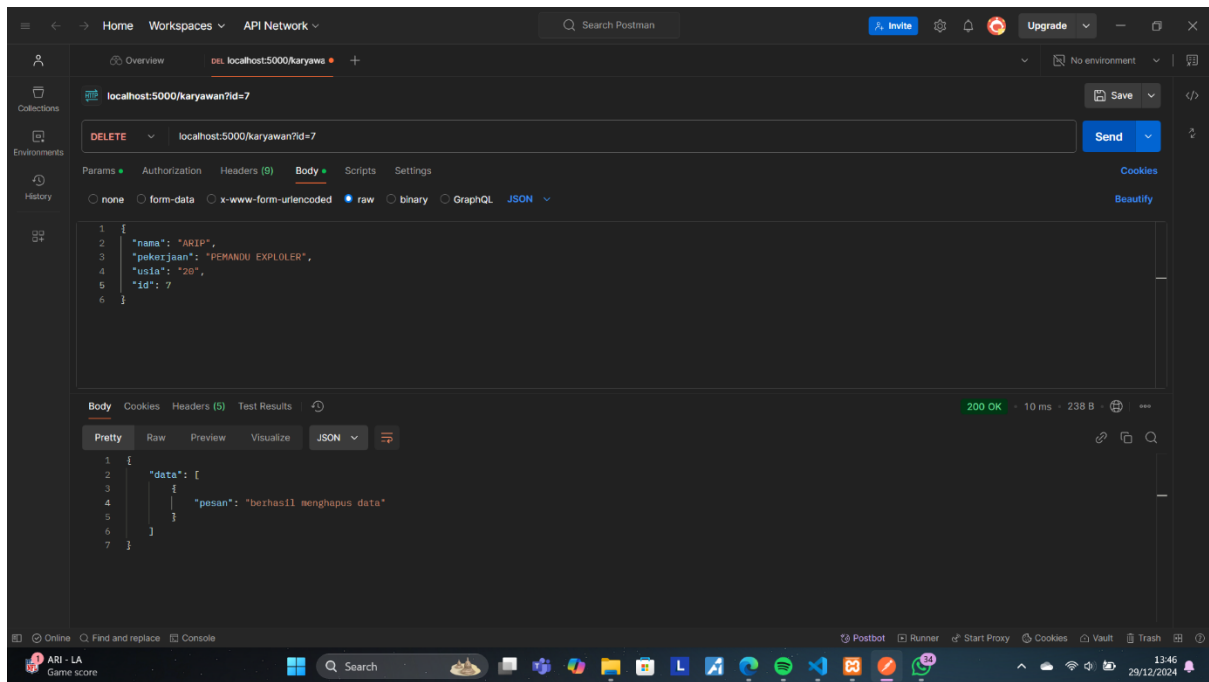
POST



PUT



DELETE



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

TERIMA KASIH