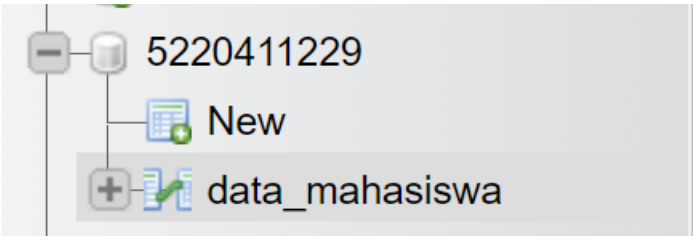
Hasil Kegiatan Praktikum	
Program Studi	Informatika
Mata Kuliah / Kelas	Pemrograman Berorientasi Objek Praktik / VII
Kelas	VI
NPM	5220411229
Nama	Alfito Juanda

Hari	jumat
Tanggal	-
Ruang	LK 2.3
Materi	Project Database

Link github:
https://github.com/aljuan14/PBOVI_5220411229_Alfito-Juanda.git

1. Screenshot database dan table yang dibuat
   a. database

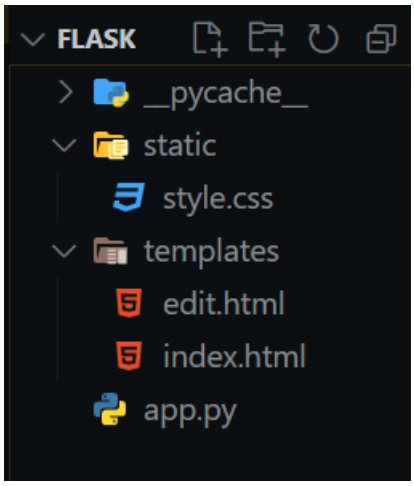


   b. table dan isinya

	npm	nama	jurusan
Edit Copy Delete	229	Alfito Juanda	Informatika
Edit Copy Delete	230	Dinda Selyani	Tata Busana
Edit Copy Delete	231	Akbar	Informatika
Edit Copy Delete	232	Burhan	Teknik Industri
Edit Copy Delete	233	Cecep	Electro
Edit Copy Delete	234	Podovski	perkucingan

2. kode program

a. struktur folder



b. isi dari app.py

```python
from flask import Flask, render_template, request, redirect
import mysql.connector

app = Flask(__name__)

class Database:
    def __init__(self, host, user, password, database):
        self.connection = mysql.connector.connect(
            host=host,
            user=user,
            passwd=password,
            database=database
        )
        self.cursor = self.connection.cursor()

    def execute_query(self, query, data=None):
        self.cursor.execute(query, data)
        self.connection.commit()

    def fetch_all(self, query):
        self.cursor.execute(query)
        return self.cursor.fetchall()

    def fetch_one(self, query, data):
        self.cursor.execute(query, data)
        return self.cursor.fetchone()

class Mahasiswa:
```

```
    def __init__(self, db):
        self.db = db

    def add_data(self, npm, nama, jurusan):
        query = "INSERT INTO data_mahasiswa (npm, nama, jurusan) VALUES
(%s, %s, %s)"
        data = (npm, nama, jurusan)
        self.db.execute_query(query, data)

    def get_all_data(self):
        query = "SELECT * FROM data_mahasiswa"
        return self.db.fetch_all(query)

    def get_data_by_npm(self, npm):
        query = "SELECT * FROM data_mahasiswa WHERE npm = %s"
        data = (npm,)
        return self.db.fetch_one(query, data)

    def update_data(self, npm, nama, jurusan):
        query = "UPDATE data_mahasiswa SET nama=%s, jurusan=%s WHERE
npm=%s"
        data = (nama, jurusan, npm)
        self.db.execute_query(query, data)

    def delete_data(self, npm):
        query = "DELETE FROM data_mahasiswa WHERE npm=%s"
        data = (npm,)
        self.db.execute_query(query, data)

# Konfigurasi Database
db_config = {
    'host': 'localhost',
    'user': 'root',
    'password': '',
    'database': '5220411229'
}

# Inisialisasi Objek Database dan Mahasiswa
database = Database(**db_config)
mahasiswa_manager = Mahasiswa(database)

@app.route('/add', methods=['POST'])
def add_data():
    npm = request.form['npm']
    nama = request.form['nama']
```

```
    jurusan = request.form['jurusan']

    mahasiswa_manager.add_data(npm, nama, jurusan)

    return redirect('/')

@app.route('/')
def index():
    data_mahasiswa = mahasiswa_manager.get_all_data()
    return render_template('index.html', data_mahasiswa=data_mahasiswa)

@app.route('/edit/<int:npm>', methods=['GET'])
def edit_data(npm):
    mahasiswa = mahasiswa_manager.get_data_by_npm(npm)
    return render_template('edit.html', mahasiswa=mahasiswa)

@app.route('/update/<int:npm>', methods=['POST'])
def update_data(npm):
    nama = request.form['nama']
    jurusan = request.form['jurusan']

    mahasiswa_manager.update_data(npm, nama, jurusan)

    return redirect('/')

@app.route('/delete/<int:npm>')
def delete_data(npm):
    mahasiswa_manager.delete_data(npm)

    return redirect('/')

if __name__ == '__main__':
    app.run(debug=True)
c.
```

d. isi dari edit.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Edit Data Mahasiswa</title>
    <link rel="stylesheet" href="../static/style.css">
</head>
<body>
```

```
    <h1>Edit Data Mahasiswa</h1>

    <form action="/update/{{ mahasiswa[0] }}" method="post">
        <label for="nama">Nama:</label>
        <input type="text" id="nama" name="nama" value="{{ mahasiswa[1]
}}"><br><br>

        <label for="jurusan">Jurusan:</label>
        <input type="text" id="jurusan" name="jurusan" value="{{
mahasiswa[2] }}"><br><br>

        <input type="submit" value="Update">
    </form>
</body>
</html>
e.
```

f. isi dari index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Data Mahasiswa</title>
    <link rel="stylesheet" href="../static/style.css">
</head>
<body>
    <h1>Data Mahasiswa</h1>

    <form action="/add" method="post">
        <label for="nama">Nama:</label>
        <input type="text" id="nama" name="nama"><br><br>

        <label for="npm">NPM:</label>
        <input type="text" id="npm" name="npm"><br><br>

        <label for="jurusan">Jurusan:</label>
        <input type="text" id="jurusan" name="jurusan"><br><br>

        <input type="submit" value="Submit">
    </form>

    <table>
        <caption></caption>
        <tr>
```

```
            <th>NPM</th>
            <th>Nama</th>
            <th>Jurusan</th>
            <th>Action</th>
        </tr>
        {% for mahasiswa in data_mahasiswa %}
            <tr>
                <td>{{ mahasiswa[0] }}</td>
                <td>{{ mahasiswa[1] }}</td>
                <td>{{ mahasiswa[2] }}</td>
                <td>
                    <a href="/edit/{{ mahasiswa[0] }}"><input
type="button1" value="Edit"></a>
                    <a href="/delete/{{ mahasiswa[0] }}"><input
type="button" value="Delete"></a>
                </td>
            </tr>
        {% endfor %}
    </table>
</body>
</html>
g.
```

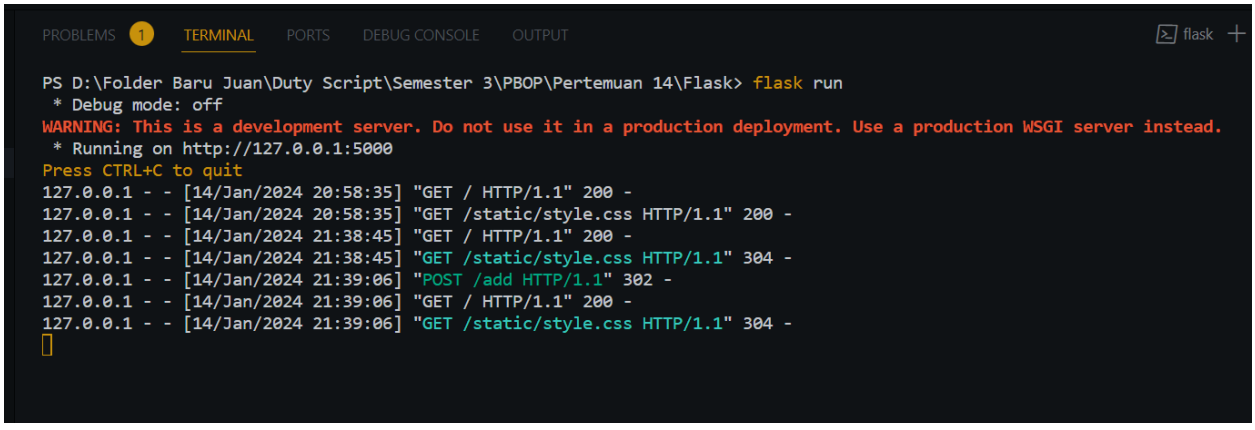
h. isi dari style.css

```
body {
    font-family: Arial, sans-serif;
}

h1 {
    text-align: center;
}

table {
    width: 50%;
    border-collapse: collapse;
    margin: 20px auto;
    border-radius: 50%;
}

th, td {
    border: 1px solid #dddddd;
    text-align: left;
    padding: 8px;
}
```

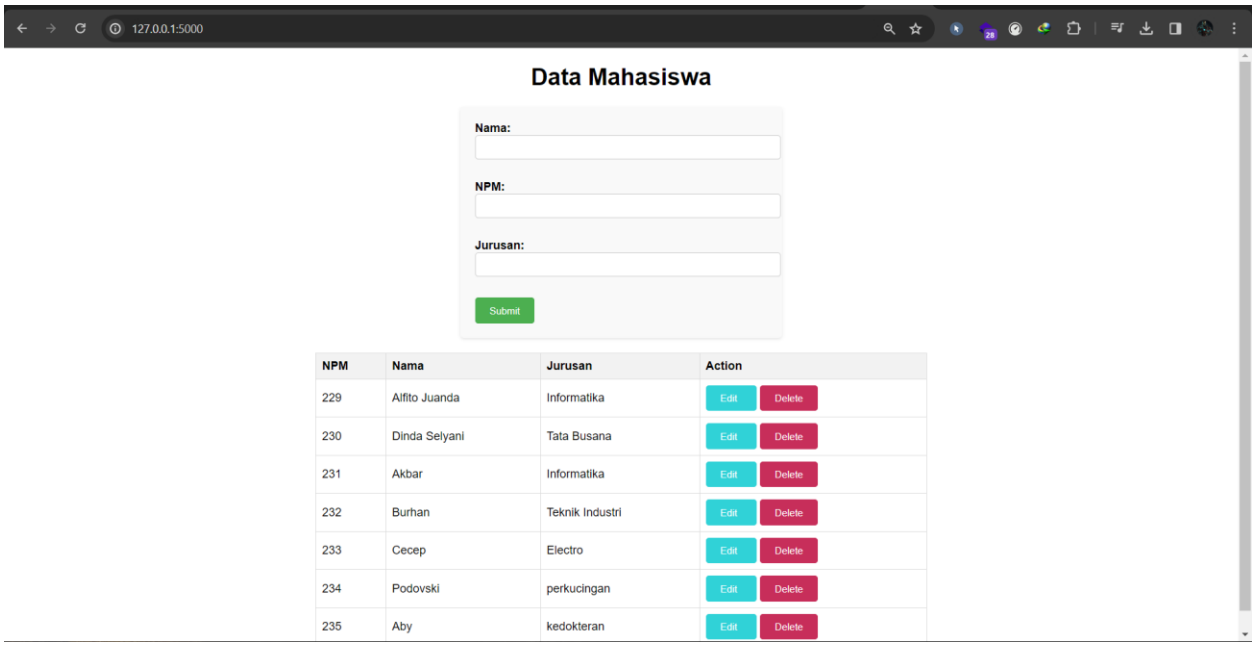
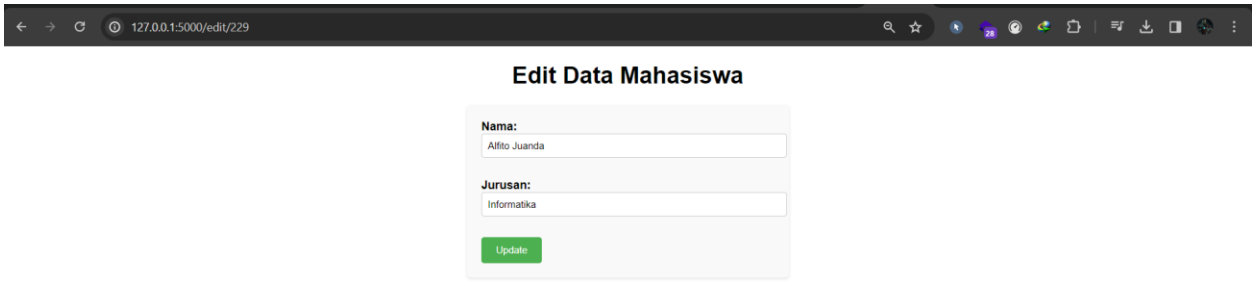
```css
th {
    background-color: #f2f2f2;
}

/* style.css */
body {
    font-family: Arial, sans-serif;
}


form {
    max-width: 400px;
    margin: 0 auto;
    background-color: #f9f9f9;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);
}

label {
    font-weight: bold;
}

input[type="text"] {
    width: 100%;
    padding: 8px;
    margin-bottom: 10px;
    border: 1px solid #ccc;
    border-radius: 4px;
}

input[type="submit"] {
    background-color: #4CAF50;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

input[type="submit"]:hover {
    background-color: #45a049;
}
input[type="button"] {
    background-color: #c72e5a;
```

```
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

input[type="button"]:hover {
    background-color: #b91c48;
}
input[type="button1"] {
    background-color: #30d2d7;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    width: 30px;
}

input[type="button1"]:hover {
    background-color: #0fb0b5;
}
i.
```

3. Hasil running program
   a. Tampilan diterminal



b. Tampilan web



Tampilan edit