

TUGAS PRAKTIKUM PEMROGRAMAN WEB LANJUTAN

KELOMPOK Q



ANGGOTA KELOMPOK :

RAEHAN RAMADHAN HAMZAH (H071221005)

AAN SYAWALUDDIN ADI PUTRA (H071221031)

ANDI MUH. HAIKAL LUKMAN (H071221089)

**UNIVERSITAS HASANUDDIN
MAKASSAR
2024**

1. List Program

- Create Database

```
CREATE DATABASE db_repositori
```

- Create Table Data Prodi

```
db_repositoriCREATE TABLE data_prodi(  
    id INT(10) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    kode_prodi CHAR(5) NULL,  
    nama_prodi VARCHAR(100) NULL );
```

- Create Table Data Dosen

```
CREATE TABLE data_dosen (  
    nip VARCHAR(30) NOT NULL PRIMARY KEY,  
    nama_lengkap VARCHAR(100) NULL,  
    prodi_id INT(10) NULL,  
    FOREIGN KEY (prodi_id) REFERENCES data_prodi(id)  
);
```

- Create Table Data Dokumen

```
CREATE TABLE data_dokumen (  
    id INT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    nip VARCHAR(30) NULL,  
    type_dokumen ENUM('file','url') NOT NULL,  
    nama_dokumen VARCHAR(255) NULL,  
    nama_file VARCHAR(255) NULL,  
    FOREIGN KEY (nip) REFERENCES data_dosen(nip)  
);
```

- Service CRUD Data Prodi

```
from app import app, db  
from flask import request, jsonify  
from app.models.DataProdi import DataProdi  
  
@app.route("/data-prodi", methods=["POST", "GET"])  
def data_prodis():  
    if request.method == "POST":  
        id = request.json["id"]  
        kode_prodi = request.json["kode_prodi"]  
        nama_prodi = request.json["nama_prodi"]  
  
        new_dataprodi = DataProdi(id=id, kode_prodi=kode_prodi,  
nama_prodi=nama_prodi)  
        db.session.add(new_dataprodi)  
        db.session.commit()  
  
        return jsonify({"message": "Data Prodi berhasil ditambahkan"})
```

```

    if request.method == "GET":
        dataprodi = DataProdi.query.all()
        print(dataprodi)
        dataprodi_list = []
        for prodi in dataprodi:
            dataprodi_list.append({
                "id": prodi.id,
                "kode_prodi": prodi.kode_prodi,
                "nama_prodi": prodi.nama_prodi
            })
        return jsonify(dataprodi_list)

@app.route("/data-prodi/<id>", methods=["GET", "PUT", "DELETE"])
def data_prodi(id):
    dataprodi = DataProdi.query.get_or_404(id)

    if request.method == "GET":
        return jsonify({
            "id": dataprodi.id,
            "kode_prodi": dataprodi.kode_prodi,
            "nama_prodi": dataprodi.nama_prodi
        })

    if request.method == "PUT":
        dataprodi.id = request.json["id"]
        dataprodi.kode_prodi = request.json["kode_prodi"]
        dataprodi.nama_prodi = request.json["nama_prodi"]
        db.session.commit()
        return jsonify({"message": "Data Prodi berhasil diperbarui"})

    if request.method == "DELETE":
        db.session.delete(dataprodi)
        db.session.commit()
        return jsonify({"message": "Data Prodi berhasil dihapus"})

```

- Service CRUD Data Dosen

```

from app import app, db
from flask import request, jsonify
from app.models.DataDosen import DataDosen

@app.route("/data-dosen", methods=["POST", "GET"])
def data_dosens():
    if request.method == "POST":
        nip = request.json["nip"]
        nama_lengkap = request.json["nama_lengkap"]

```

```

        prodi_id = request.json["prodi_id"]

        new_datadosen = DataDosen(nip=nip, nama_lengkap=nama_lengkap,
prodi_id=prodi_id)
        db.session.add(new_datadosen)
        db.session.commit()

        return jsonify({"message": "Data Dosen berhasil ditambahkan"})

    if request.method == "GET":
        datadosen = DataDosen.query.all()
        print(datadosen)
        datadosen_list = []
        for dosen in datadosen:
            datadosen_list.append({
                "nip": dosen.nip,
                "nama_lengkap": dosen.nama_lengkap,
                "prodi_id": dosen.prodi_id
            })
        return jsonify(datadosen_list)

@app.route("/data-dosen/<nip>", methods=["GET", "PUT", "DELETE"])
def data_dosen(nip):
    datadosen = DataDosen.query.get_or_404(nip)

    if request.method == "GET":
        return jsonify({
            "nip": datadosen.nip,
            "nama_lengkap": datadosen.nama_lengkap,
            "prodi_id": datadosen.prodi_id
        })

    if request.method == "PUT":
        datadosen.nip = request.json["nip"]
        datadosen.nama_lengkap = request.json["nama_lengkap"]
        datadosen.prodi_id = request.json["prodi_id"]
        db.session.commit()
        return jsonify({"message": "Data Dosen berhasil diperbarui"})

    if request.method == "DELETE":
        db.session.delete(datadosen)
        db.session.commit()
        return jsonify({"message": "Data Dosen berhasil dihapus"})

```

- Service CRUD Data Dokumen

```
from app import app, db
from flask import request, jsonify
from app.models.DataDokumen import DataDokumen

@app.route("/data-dokumen", methods=["POST", "GET"])
def data_dokumens():
    if request.method == "POST":
        id = request.json["id"]
        nip = request.json["nip"]
        type_dokumen = request.json["type_dokumen"]
        nama_dokumen = request.json["nama_dokumen"]
        nama_file = request.json["nama_file"]

        if type_dokumen != "file" and type_dokumen != "url":
            return jsonify({"message": "Tipe dokumen harus file atau url"})

        new_datadokumen = DataDokumen(id=id, nip=nip,
type_dokumen=type_dokumen, nama_dokumen=nama_dokumen, nama_file=nama_file)
        db.session.add(new_datadokumen)
        db.session.commit()

        return jsonify({"message": "Data Dokumen berhasil ditambahkan"})

    if request.method == "GET":
        datadokumen = DataDokumen.query.all()
        print(datadokumen)
        datadokumen_list = []
        for dokumen in datadokumen:
            datadokumen_list.append({
                "id": dokumen.id,
                "nip": dokumen.nip,
                "type_dokumen": dokumen.type_dokumen,
                "nama_dokumen": dokumen.nama_dokumen,
                "nama_file": dokumen.nama_file,
            })
        return jsonify(datadokumen_list)

@app.route("/data-dokumen/<id>", methods=["GET", "PUT", "DELETE"])
def data_dokumen(id):
    datadokumen = DataDokumen.query.get_or_404(id)

    if request.method == "GET":
        return jsonify({
            "id": datadokumen.id,
            "nip": datadokumen.nip,
            "type_dokumen": datadokumen.type_dokumen,
            "nama_dokumen": datadokumen.nama_dokumen,
```

```

        "nama_file": datadokumen.nama_file,
    })

    if request.method == "PUT":
        datadokumen.id = request.json["id"]
        datadokumen.nip = request.json["nip"]
        datadokumen.type_dokumen = request.json["type_dokumen"]
        datadokumen.nama_dokumen = request.json["nama_dokumen"]
        datadokumen.nama_file = request.json["nama_file"]

        if datadokumen.type_dokumen != "file" and datadokumen.type_dokumen !=
"url":
            return jsonify({"message": "Tipe dokumen harus file atau url"})

        db.session.commit()
        return jsonify({"message": "Data Dokumen berhasil diperbarui"})

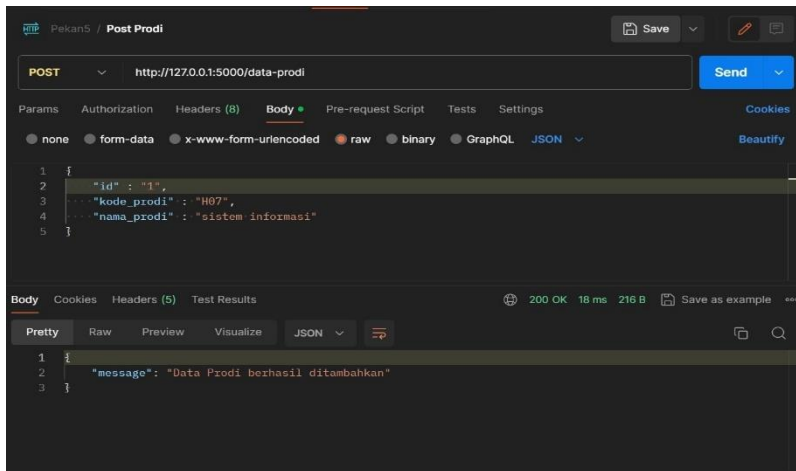
    if request.method == "DELETE":
        db.session.delete(datadokumen)
        db.session.commit()
        return jsonify({"message": "Data Dokumen berhasil dihapus"})

```

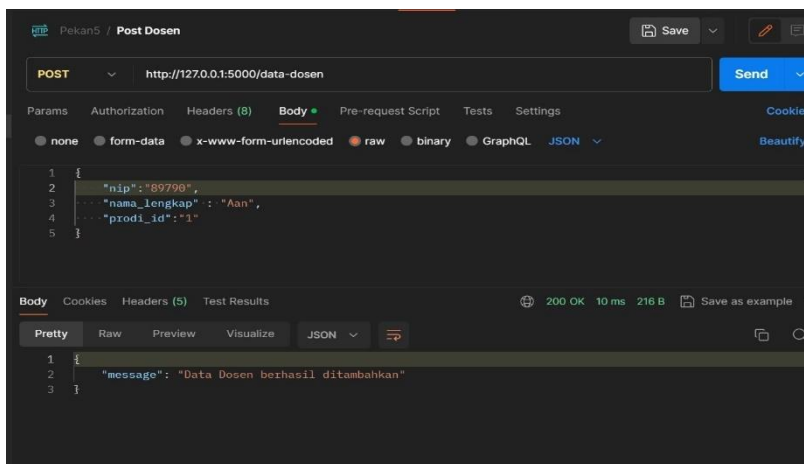
2. Preview Eksekusi Service Pada Postman

- POST

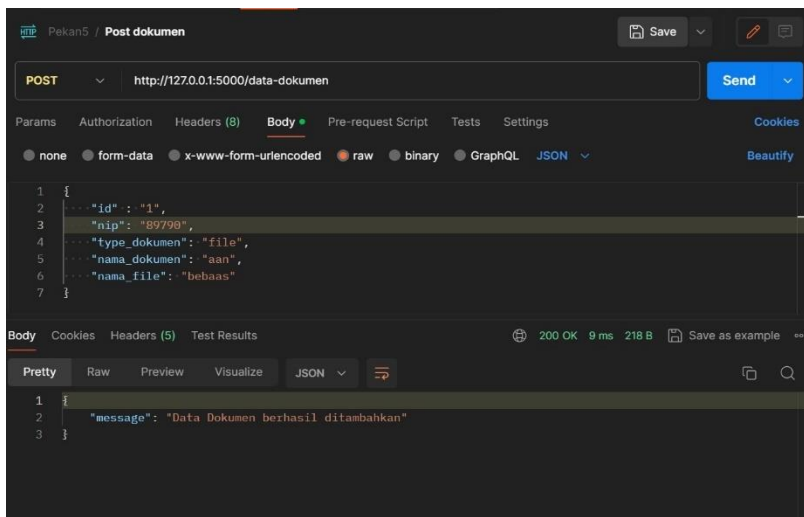
1. Data Prodi



2. Data Dosen

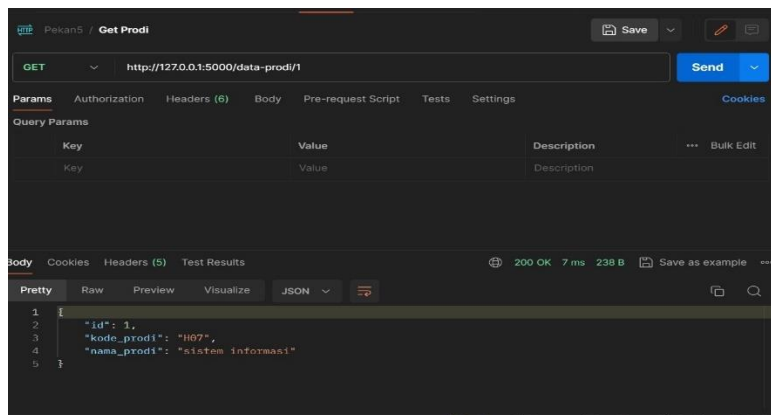


3. Data Dokumen

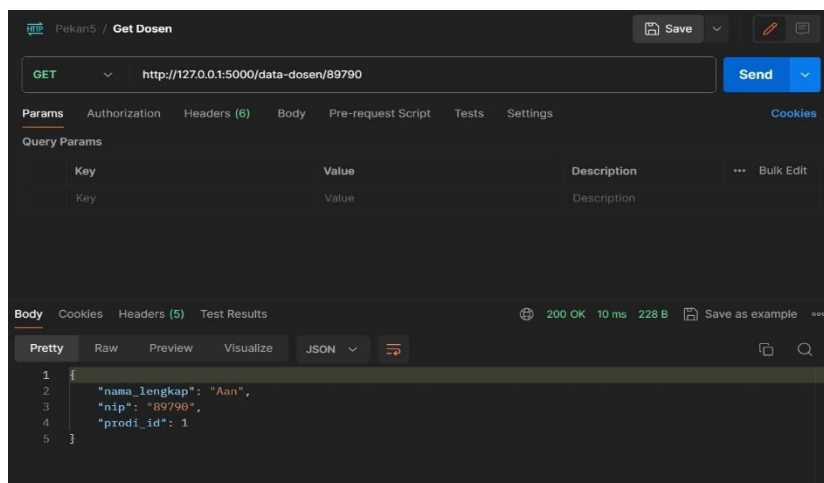


- GET

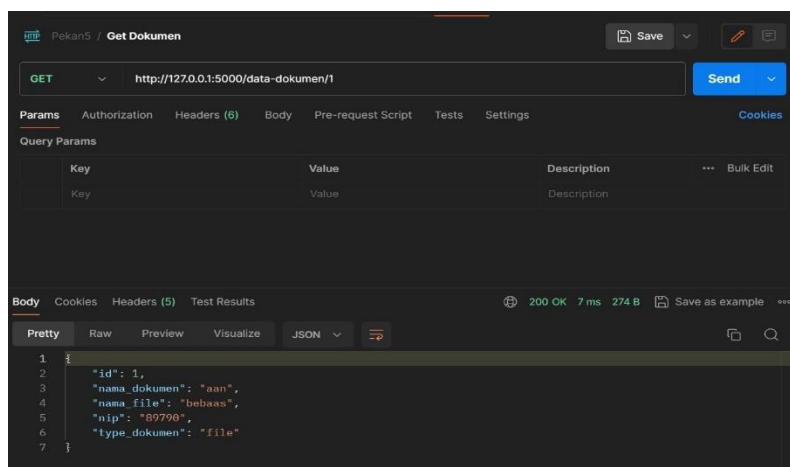
1. Data Prodi



2. Data Dosen

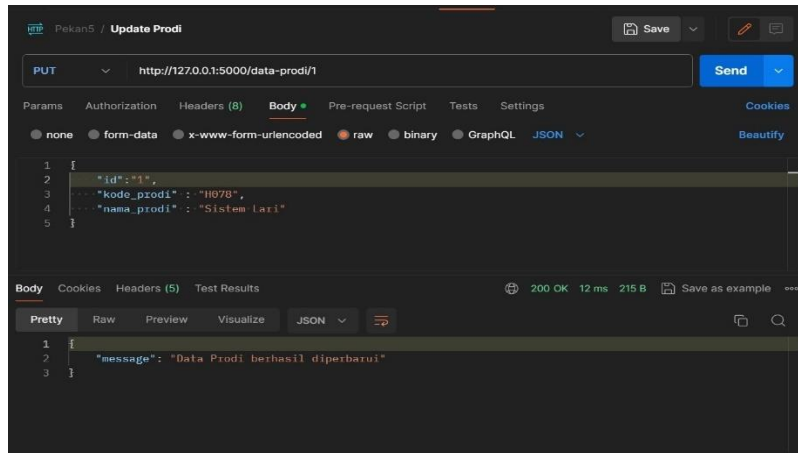


3. Data Dokumen

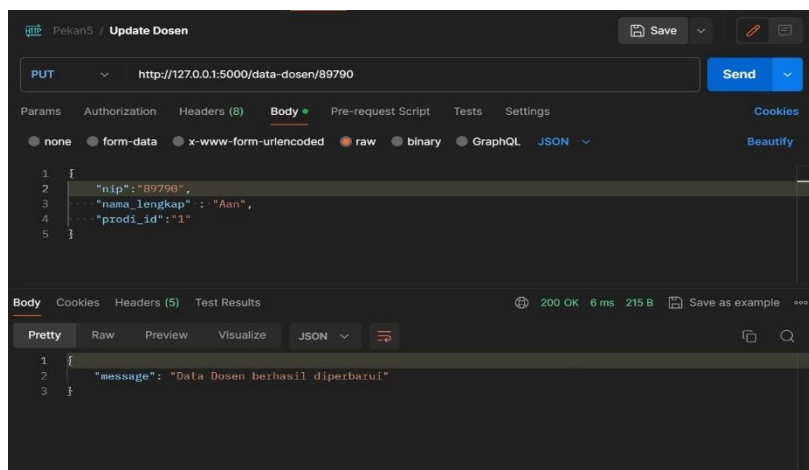


- PUT

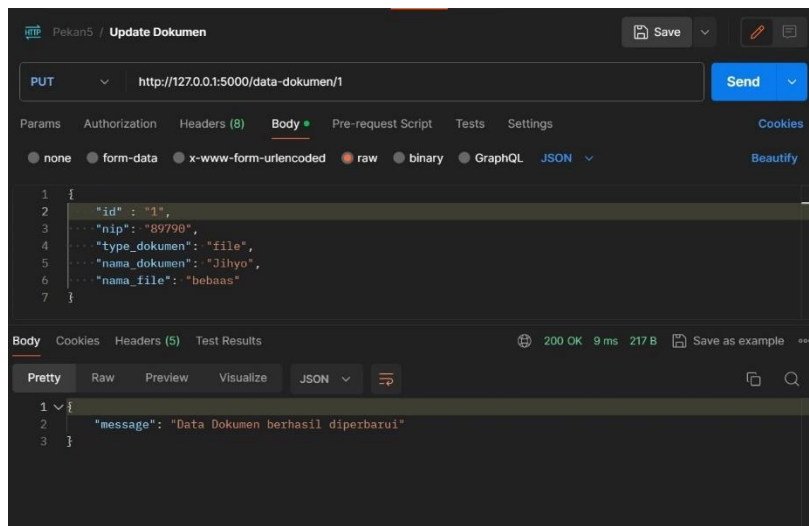
1. Data Prodi



2. Data Dosen

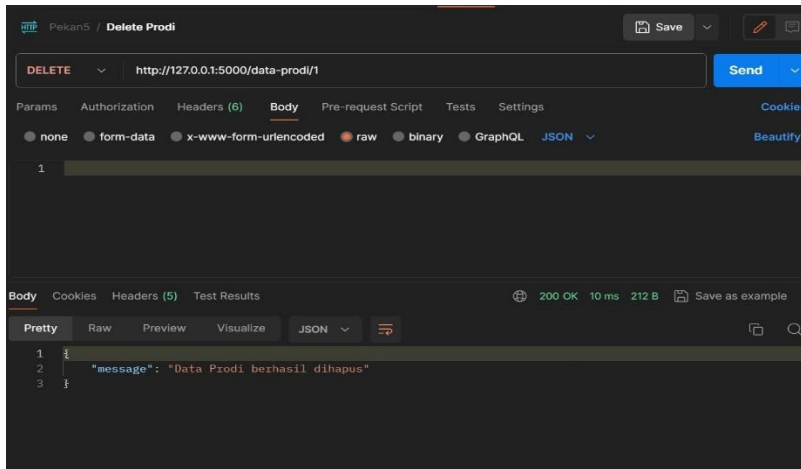


3. Data Dokumen

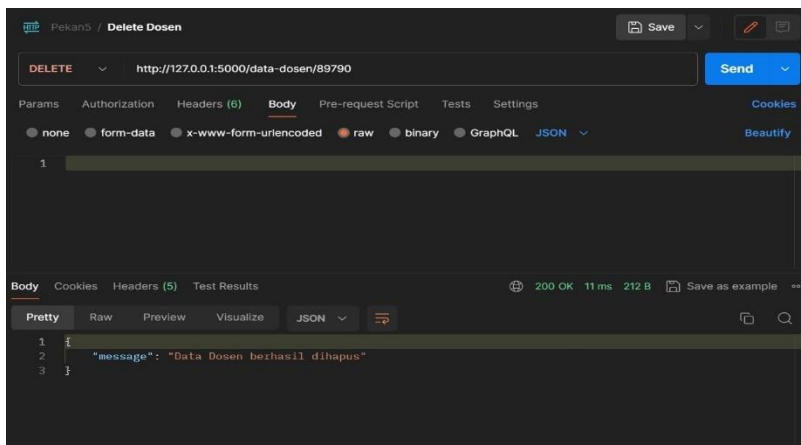


- DELETE

1. Data Prodi



2. Data Dosen



3. Data Dokumen

