Nama : Muhammad Irsyad

NIM : 2211102048

Kelas : S1-IF-10-Q

	Institut Teknologi Telkom Purwokerto **FAKULTAS INFORMATIKA** **PROGRAM STUDI S1 TEKNIK INFORMATIKA**				
RENCANA PENUGASAN TERSTRUKTUR MAHASISWA					
MATA KULIAH	Pemrograman Fungsional				
KODE	IF123231	SKS	3	Semester	5
DOSEN PENGAMPU	Andi Prademon Yunus, ST., M.Sc. Eng., Ph.D				
BENTUK TUGAS	Tugas Terstruktur				
WAKTU PENGERJAAN TUGAS	2 Minggu				
JUDUL TUGAS					
Tugas Terstruktur : Membuat Restfull API dengan tema Bebas					
SUB CAPAIAN PEMBELAJARAN MATA KULIAH					
Mahasiswa mampu menerapkan Fungsional Programming dalam Web Services (C4)					
DESKRIPSI TUGAS					
Mahasiswa diminta untuk membuat RESTful API dengan tema bebas tidak boleh tentang mahasiswa seperti contoh yang sudah diberikan. Skema Method RESTful API yaitu ada GET, POST, PUT, DELETE					
METODE PENGERJAAN TUGAS					
Berikut adalah panduan poin yang harus disertakan dalam video: 1. Kerjakan build RESTful API menggunakan Flask Framework method GET, POST, PUT. DELETE dengan tema bebas. 2. Database yang digunakan bebas. 3. Harus sudah terkoneksi dengan database local atau sever 4. **Berhasil deploy ke server dapat nilai bonus (tidak wajib)**					
BENTUK DAN FORMAT LUARAN					

A. Bentuk
Pembuatan RESTful API bersifat individu

B. Format Pengumpulan
1. Dokumentasi API Postman
2. Link publish documentation API Postman

INDIKATOR, KRITERIA DAN BOBOT PENILAIAN	
Implementasi RESTful API	30
Connect ke database	20
Documentation API	30
Keberagaman Tema yang diambil	20
Total	**100**
Deploy server (tidak wajib)	**+20**

DATELINE PENGUMPULAN TUGAS

Pengumpulan tugas besar Jumat 24 Januari 2024, jam 23:

SANKSI DAN PELANGGARAN

- Terlambat mengumpulkan tidak akan dinilai

DAFTAR RUJUKAN

Jawab :

Disini saya Membuat RESTful API untuk menyimpan data Pegawai Negeri sipil

• Database yang digunakan : Locahost

```python
from flask import Flask, render_template, request, url_for, flash
from werkzeug.utils import redirect
from flask_mysqldb import MySQL

app = Flask(__name__)
app.secret_key = 'many random bytes'

app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'db_pegawai'

mysql = MySQL(app)

@app.route('/')
def Index():
    cur = mysql.connection.cursor()
    cur.execute("SELECT * FROM pegawai")
    data = cur.fetchall()
    cur.close()

    return render_template('index.html', pegawai=data)

@app.route('/insert', methods = ['POST'])
def insert():
```

1. Import:

- from flask import Flask, render_template, request, url_for, flash: Mengimport fungsi-fungsi dari Flask untuk membuat aplikasi web, menampilkan template, menangani permintaan, membuat URL, dan menampilkan pesan flash.
- from werkzeug.utils import redirect: Mengimport fungsi redirect untuk mengarahkan pengguna ke halaman lain.
- from flask_mysqldb import MySQL: Mengimport ekstensi untuk menghubungkan Flask dengan database MySQL.

2. Membuat Aplikasi Flask:

- app = Flask(__name__): Membuat objek aplikasi Flask.

- app.secret_key = 'many random bytes': Mengatur kunci rahasia untuk mengamankan pesan flash dan fitur-fitur lainnya.

3. Konfigurasi Database MySQL:

- app.config['MYSQL_HOST'] = 'localhost': Mengatur host database.
- app.config['MYSQL_USER'] = 'root': Mengatur nama pengguna database.
- app.config['MYSQL_PASSWORD'] = '': Mengatur kata sandi database (kosong dalam contoh ini).
- app.config['MYSQL_DB'] = 'db_pegawai': Mengatur nama database.mysql
- MySQL(app): Membuat objek MySQL untuk terhubung ke database.

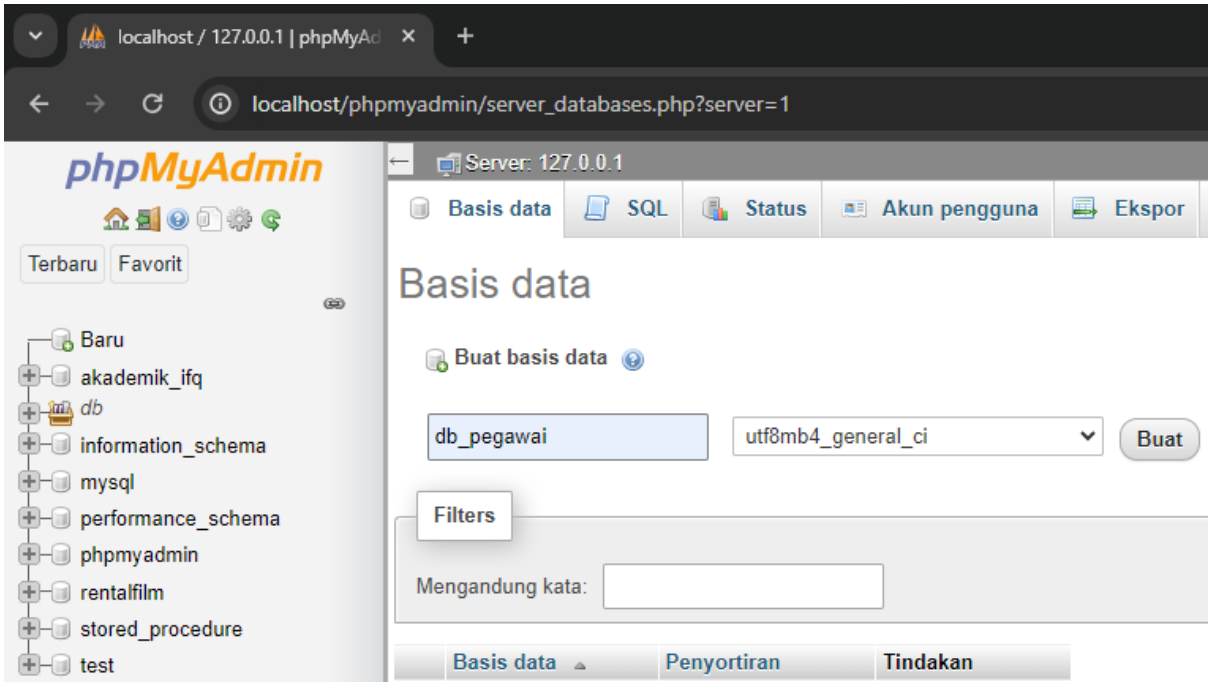
```python

@app.route('/')
def Index():
    cur = mysql.connection.cursor()
    cur.execute("SELECT * FROM pegawai")
    data = cur.fetchall()
    cur.close()

    return render_template('index.html', pegawai=data)

@app.route('/insert', methods = ['POST'])
def insert():
    if request.method == "POST":
        flash("Data Inserted Successfully")
        nama = request.form['nama']
        nip = request.form['nip']
        tgl_lahir = request.form['tgl_lahir']
        pangkat = request.form['pangkat']
        golongan = request.form['golongan']
        jabatan = request.form['jabatan']
        cur = mysql.connection.cursor()
        cur.execute("INSERT INTO pegawai (nama, nip, tgl_lahir, pangkat, golongan, jabatan) VALUES (%s, %s, %s, %s, %s, %s)",
                    (nama, nip, tgl_lahir, pangkat, golongan, jabatan))
        mysql.connection.commit()
        return redirect(url_for('Index'))

@app.route('/delete/<string:id_data>', methods = ['DELETE'])
def delete(id_data):
    flash("Record Has Been Deleted Successfully")
    cur = mysql.connection.cursor()
    cur.execute("DELETE FROM pegawai WHERE id=%s", (id_data,))
    mysql.connection.commit()
    return redirect(url_for('Index'))

@app.route('/update', methods= ['PUT', 'GET'])
def update():
    if request.method == 'PUT':
        id_data = request.form['id']
        nama = request.form['nama']
        nip = request.form['nip']
        tgl_lahir = request.form['tgl_lahir']
        pangkat = request.form['pangkat']
        golongan = request.form['golongan']
        jabatan = request.form['jabatan']

        cur = mysql.connection.cursor()
        cur.execute("""
        UPDATE pegawai SET nama=%s, nip=%s, tgl_lahir=%s, pangkat=%s, golongan=%s, jabatan=%s
        WHERE id=%s
        """, (nama, nip, tgl_lahir, pangkat, golongan, jabatan, id_data))
        mysql.connection.commit()
        flash("Data Updated Successfully")
        return redirect(url_for('Index'))

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

4. Fungsi-Fungsi Utama:

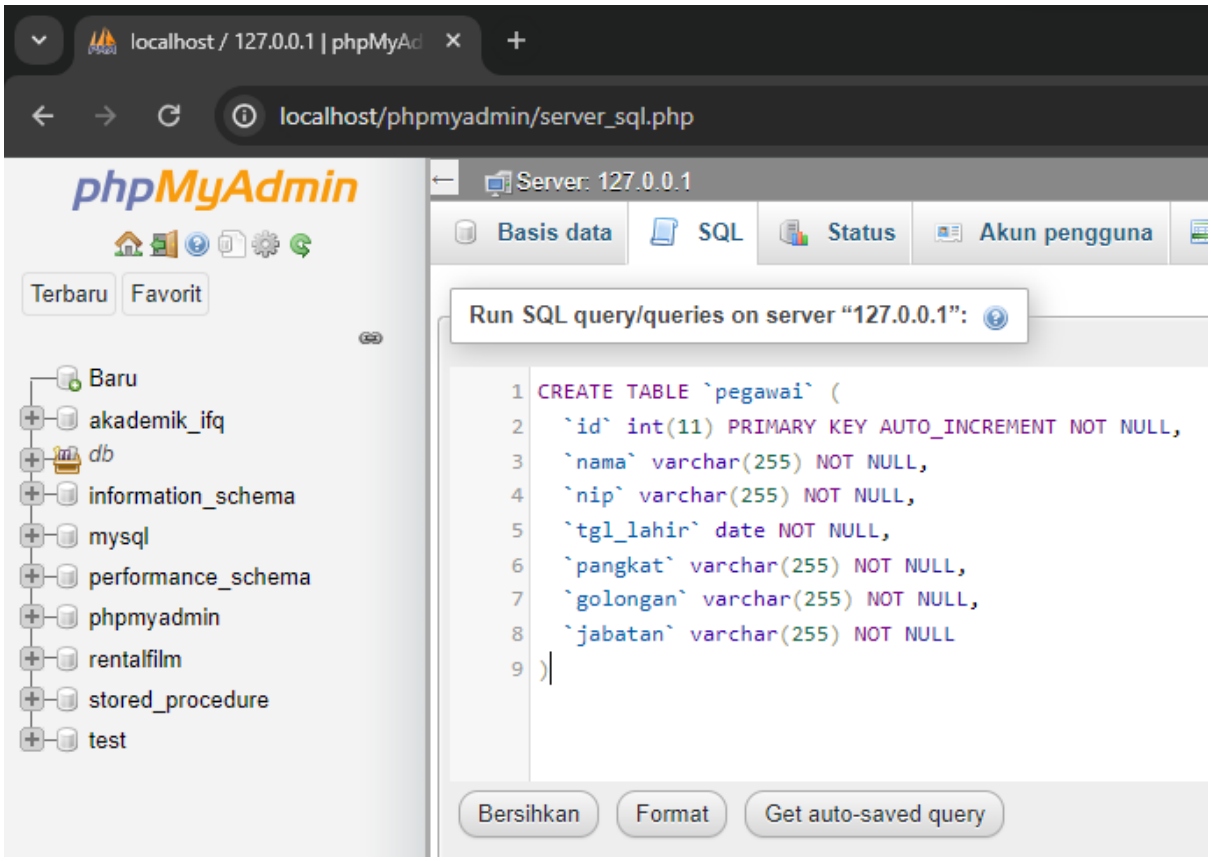
- Index(): Menampilkan halaman utama yang menampilkan daftar pegawai.
    - Mengambil data pegawai dari database menggunakan cursor MySQL.
    - Mengirimkan data pegawai ke template index.html untuk ditampilkan.
- insert(): Menambahkan data pegawai baru ke database.
    - Menerima data pegawai dari form HTML melalui request POST.
    - Memasukkan data pegawai ke dalam database menggunakan SQL INSERT.
    - Menampilkan pesan flash "Data Inserted Successfully".
    - Mengarahkan pengguna kembali ke halaman utama.
- delete(id_data): Menghapus data pegawai dengan ID yang diberikan.
    - Menghapus data pegawai dari database menggunakan SQL DELETE.
    - Menampilkan pesan flash "Record Has Been Deleted Successfully".
    - Mengarahkan pengguna kembali ke halaman utama.
- update(): Memperbarui data pegawai yang sudah ada.
    - Menerima data pegawai yang diperbarui dari form HTML melalui request POST.
    - Memperbarui data pegawai di database menggunakan SQL UPDATE.
    - Menampilkan pesan flash "Data Updated Successfully".
    - Mengarahkan pengguna kembali ke halaman utama.

5. Menjalankan Aplikasi:

- if __name__ == "__main__":: Memeriksa apakah kode ini dijalankan sebagai program utama.
- app.run(debug=True): Menjalankan aplikasi Flask dalam mode debug, yang memudahkan untuk melihat error dan perubahan.



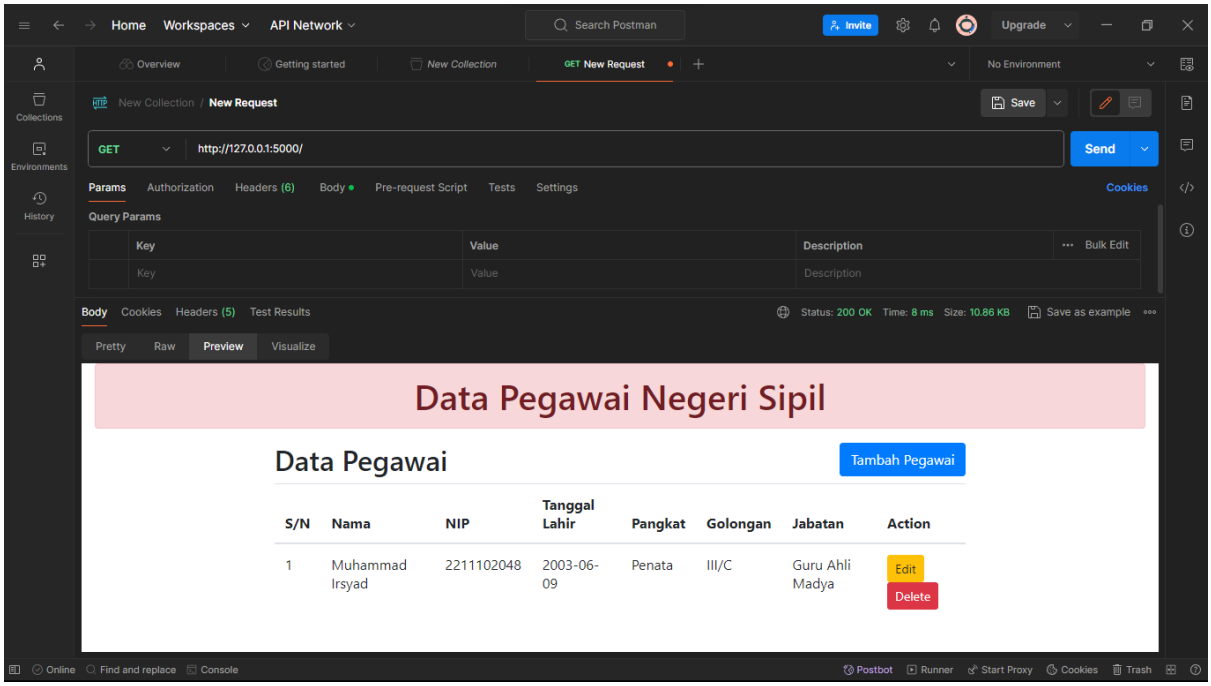
Buat database di phpMyAdmin db_pegawai sesuai denga kodingan koneksi



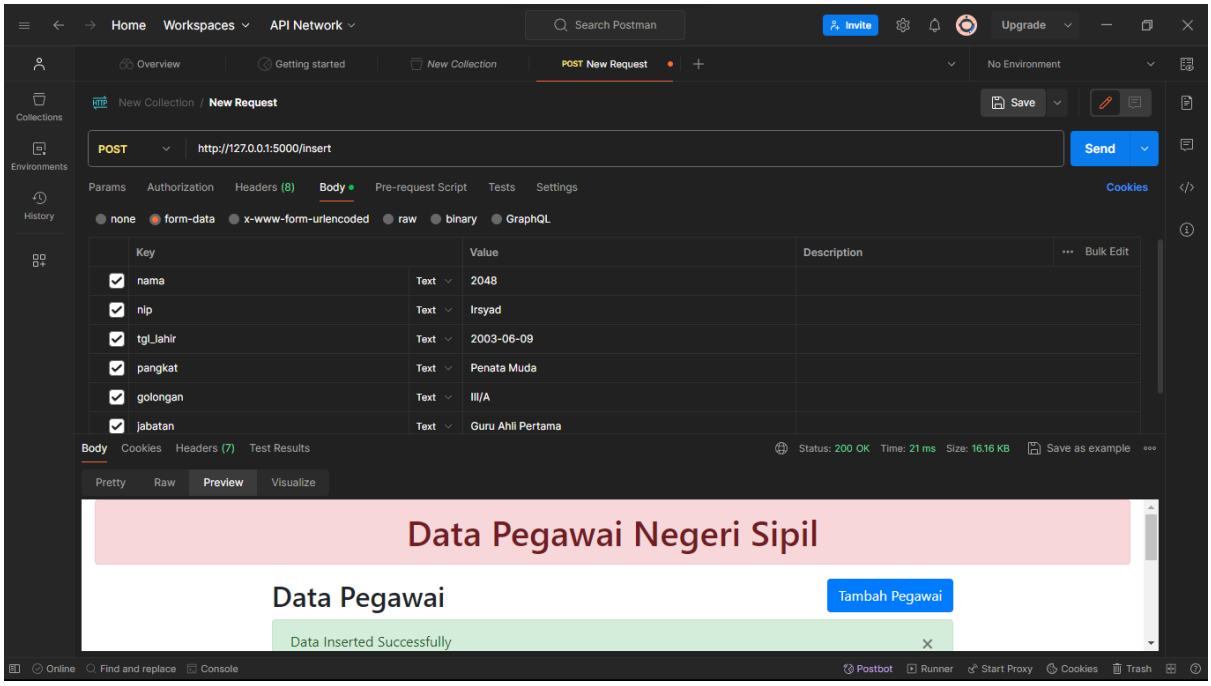
Lalu membuat tabel pegawai dengan kolom id, nama, nip, tgl_lahir, pangkat, golongan, dan jabatan

PENERAPAN POSTMAN

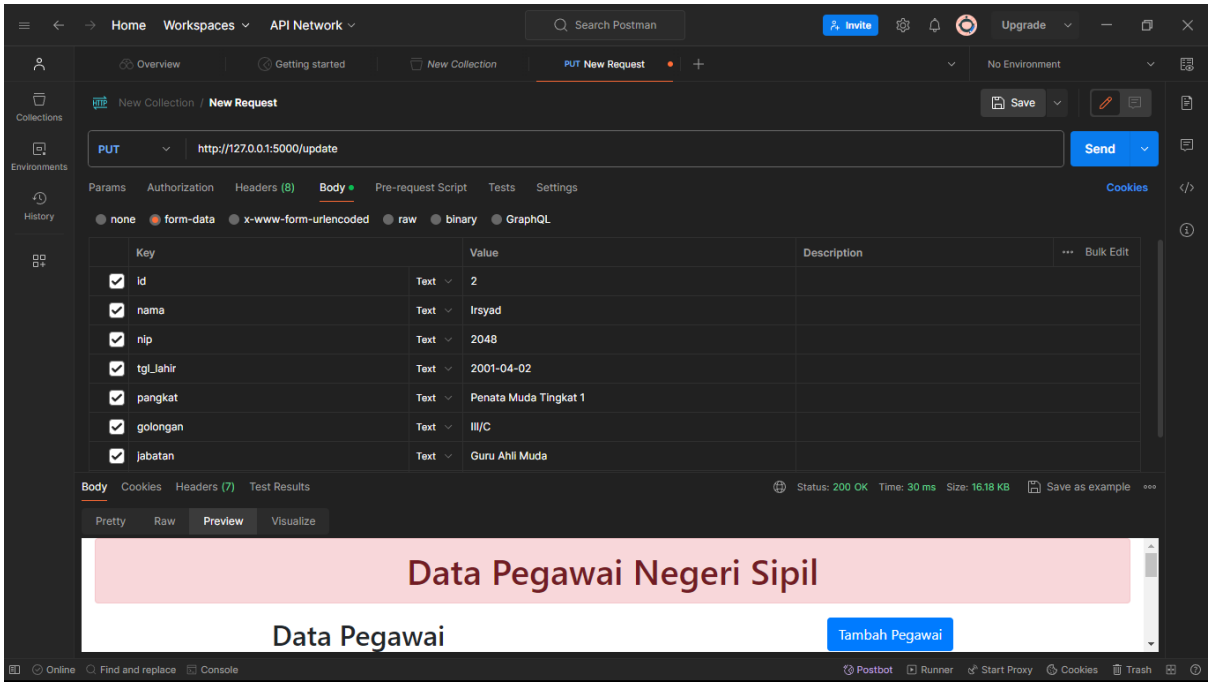
- **GET**

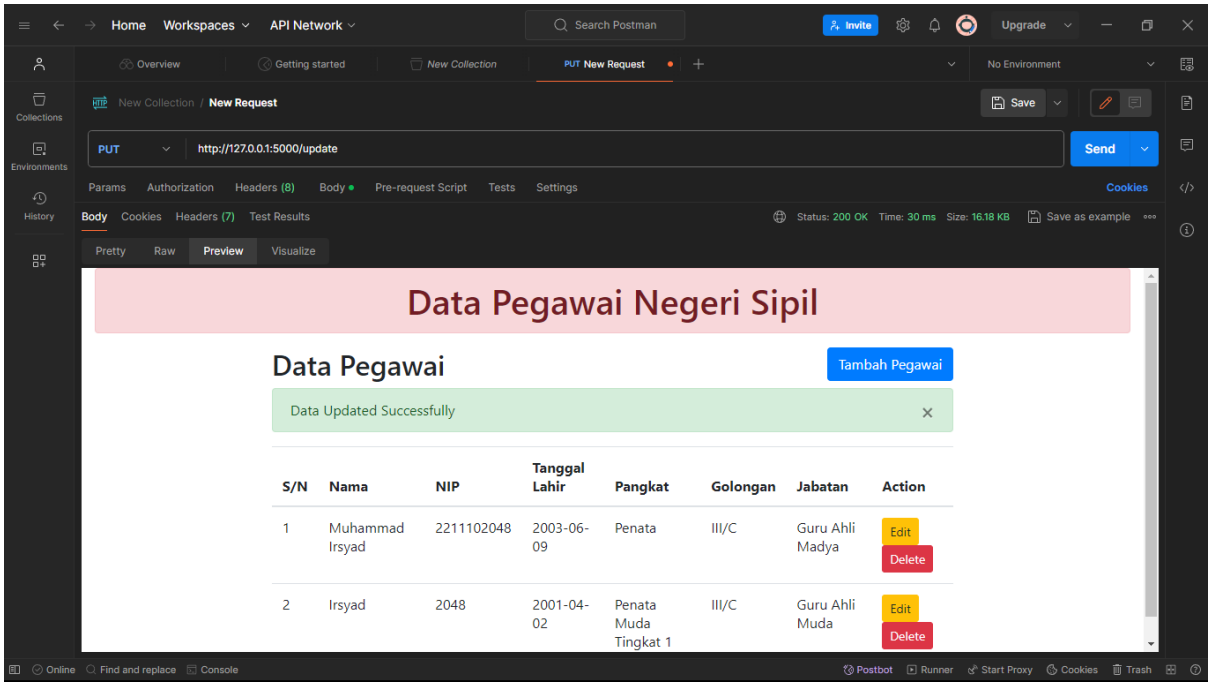


- **POST**

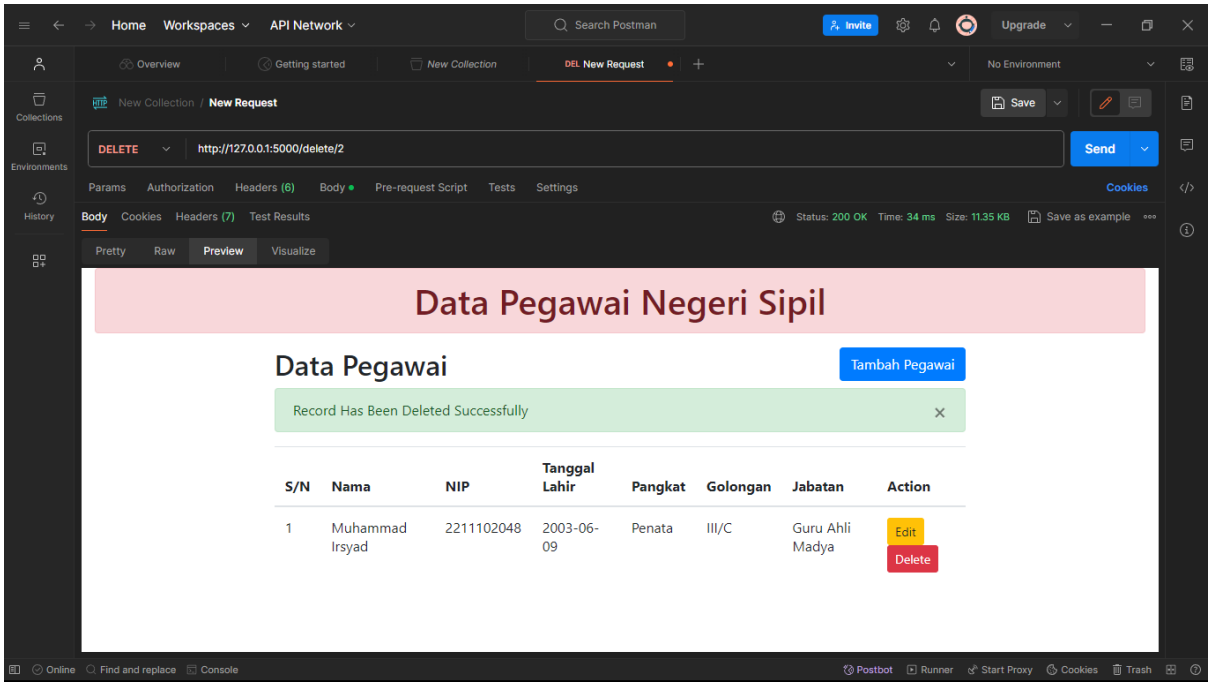


- **PUT**


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
PUT New Request
No Environment
New Collection / New Request
Save
PUT
http://127.0.0.1:5000/update
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
Key
Value
Description
Bulk Edit
id
2
nama
Irsyad
nip
2048
tgl_lahir
2001-04-02
pangkat
Penata Muda Tingkat 1
golongan
III/C
jabatan
Guru Ahli Muda
Body
Cookies
Headers (7)
Test Results
Status: 200 OK
Time: 30 ms
Size: 16.18 KB
Save as example
Pretty
Raw
Preview
Visualize
Data Pegawai Negeri Sipil
Data Pegawai
Tambah Pegawai


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
PUT New Request
No Environment
New Collection / New Request
Save
PUT
http://127.0.0.1:5000/update
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
Body
Cookies
Headers (7)
Test Results
Status: 200 OK
Time: 30 ms
Size: 16.18 KB
Save as example
Pretty
Raw
Preview
Visualize
Data Pegawai Negeri Sipil
Data Pegawai
Tambah Pegawai
Data Updated Successfully
S/N
Nama
NIP
Tanggal Lahir
Pangkat
Golongan
Jabatan
Action
1
Muhammad Irsyad
2211102048
2003-06-09
Penata
III/C
Guru Ahli Madya
Edit
Delete
2
Irsyad
2048
2001-04-02
Penata Muda Tingkat 1
III/C
Guru Ahli Muda
Edit
Delete

- **DELETE**


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
Overview
Getting started
New Collection
DEL New Request
No Environment
Collections
Environments
History
New Collection / New Request
Save
DELETE
http://127.0.0.1:5000/delete/2
Send
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
Body
Cookies
Headers (7)
Test Results
Status: 200 OK
Time: 34 ms
Size: 11.35 KB
Save as example
Pretty
Raw
Preview
Visualize
Data Pegawai Negeri Sipil
Data Pegawai
Tambah Pegawai
Record Has Been Deleted Successfully
S/N
Nama
NIP
Tanggal Lahir
Pangkat
Golongan
Jabatan
Action
1
Muhammad Irsyad
2211102048
2003-06-09
Penata
III/C
Guru Ahli Madya
Edit
Delete
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Trash