

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Algoritma dan Pemrograman 2B

Kelas : 1IA23

Praktikum Ke- : 5

Tanggal : 12 May 2023

Materi : Request API

NPM : 51422041

Nama : Muhammad Faris Rasyid Raharjo

Ketua Asisten : Fernando Michael Hebert

Nama Asisten :

Paraf Asisten :

Jumlah Lembar : 9 Lembar



LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2023

LISTING

Program 1

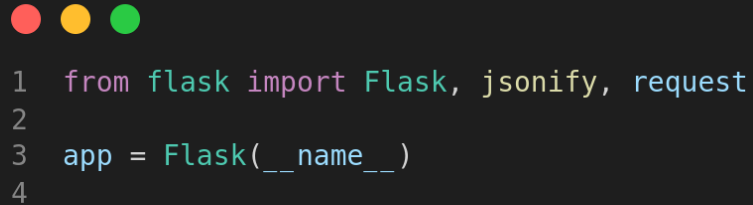
```
1 from flask import Flask, jsonify, request
2
3 app = Flask(__name__)
4
5 # sample data
6 students = [
7     {'id': 1, 'name': 'Rahmad', 'age': 19},
8     {'id': 2, 'name': 'Wicaksono', 'age': 22},
9     {'id': 3, 'name': 'Adi', 'age': 20}
10 ]
11
12 # GET all students data
13 @app.route('/students', methods=['GET'])
14 def get_students():
15     return jsonify(students)
16
17 # GET student data by ID
18 @app.route('/students/<int:id>', methods=['GET'])
19 def get_student(id):
20     student = [student for student in students if student['id'] == id]
21     return jsonify(student)
22
23 # CREATE new student data
24 @app.route('/students', methods=['POST'])
25 def create_student():
26     student = {
27         'id': request.json['id'],
28         'name': request.json['name'],
29         'age': request.json['age']
30     }
31     students.append(student)
32     return jsonify(student)
33
34 # UPDATE student data by ID
35 @app.route('/students/<int:id>', methods=['PUT'])
36 def update_student(id):
37     student = [student for student in students if student['id'] == id]
38     student[0]['name'] = request.json['name']
39     student[0]['age'] = request.json['age']
40     return jsonify(student[0])
41
42 # DELETE student data by ID
43 @app.route('/students/<int:id>', methods=['DELETE'])
44 def delete_student(id):
45     student = [student for student in students if student['id'] == id]
46     students.remove(student[0])
47     return jsonify({'result': True})
48
49 if __name__ == '__main__':
50     app.run(debug=True)
```

Program 2

```
1 import requests
2
3 # GET all students data
4 response = requests.get('http://localhost:5000/students')
5 print('Mengambil Seluruh Data')
6 print(response.json())
7 print("=====")
8
9 # GET student data by ID
10 response = requests.get('http://localhost:5000/students/1')
11 print('Mengambil Data dengan ID = 1')
12 print(response.json())
13 print("=====")
14
15 # CREATE new student data
16 new_student = {'id': 4, 'name': 'Prastowo', 'age': 21}
17 response = requests.post('http://localhost:5000/students', json=new_student)
18 print('Menambahkan Data Baru ke Json')
19 print(response.json())
20
21 # GET all students data after creating new student
22 response = requests.get('http://localhost:5000/students')
23 print(response.json())
24 print("=====")
25 # UPDATE student data by ID
26 update_student = {'name': 'Jordan', 'age': 19}
27 response = requests.put('http://localhost:5000/students/2', json=update_student)
28 print('Memperbarui Data pada ID = 2')
29 print(response.json())
30
31 # GET all students data after updating student
32 response = requests.get('http://localhost:5000/students')
33 print(response.json())
34 print("=====")
35
36 # DELETE student data by ID
37 response = requests.delete('http://localhost:5000/students/3')
38 print('Menghapus Data dengan ID = 3')
39 print(response.json())
40
41 # GET all students data after deleting student
42 response = requests.get('http://localhost:5000/students')
43 print(response.json())
44
```

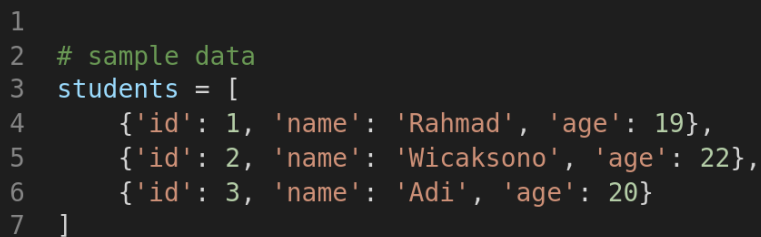
LOGIKA

Program 1



```
1 from flask import Flask, jsonify, request
2
3 app = Flask(__name__)
4
```

Mengimpor modul-modul yang diperlukan dari Flask: `Flask` (untuk membuat aplikasi Flask), `jsonify` (untuk mengonversi data menjadi format JSON), dan `request` (untuk mengakses data permintaan HTTP). Setelah itu Membuat instance Flask dengan menggunakan `Flask(__name__)`. Instance ini akan menjadi wadah untuk aplikasi Flask.



```
1
2 # sample data
3 students = [
4     {'id': 1, 'name': 'Rahmad', 'age': 19},
5     {'id': 2, 'name': 'Wicaksono', 'age': 22},
6     {'id': 3, 'name': 'Adi', 'age': 20}
7 ]
```

Mendefinisikan data mahasiswa sebagai contoh. Data ini berupa daftar dictionary yang berisi informasi mahasiswa. Yang penting dari nama kelas dan umur. Data tersebut berbentuk array yang terdiri dari beberapa object

```

1  # GET all students data
2  @app.route('/students', methods=['GET'])
3  def get_students():
4      return jsonify(students)
5
6  # GET student data by ID
7  @app.route('/students/<int:id>', methods=['GET'])
8  def get_student(id):
9      student = [student for student in students if student['id'] == id]
10     return jsonify(student)

```


Berikut adalah endpoint untuk mengambil data. Membuat *route* atau *endpoint* kedua dengan menggunakan decorator `@app.route()`. *Route* ini akan menangani permintaan GET untuk mendapatkan data mahasiswa berdasarkan ID. Dan Membuat *route* atau *endpoint* ketiga dengan menggunakan decorator `@app.route()`. *Route* ini akan menangani permintaan POST untuk membuat data mahasiswa baru.

```

1  # CREATE new student data
2  @app.route('/students', methods=['POST'])
3  def create_student():
4      student = {
5          'id': request.json['id'],
6          'name': request.json['name'],
7          'age': request.json['age']
8      }
9      students.append(student)
10     return jsonify(student)

```

Membuat *route* atau *endpoint* ketiga dengan menggunakan decorator `@app.route()`. Dengan membuat constructor yang akan di jalankan pertama kali ketika endpoint dipanggil, *Route* ini akan menangani permintaan POST untuk membuat data mahasiswa baru.




```

1 # UPDATE student data by ID
2 @app.route('/students/<int:id>', methods=['PUT'])
3 def update_student(id):
4     student = [student for student in students if student['id'] == id]
5     student[0]['name'] = request.json['name']
6     student[0]['age'] = request.json['age']
7     return jsonify(student[0])
8

```

Membuat *route* atau *endpoint* keempat dengan menggunakan decorator `@app.route()`. *Route* ini akan menangani permintaan PUT untuk memperbarui data mahasiswa berdasarkan ID dari endpoint setelah `/students/<nomor-id>`.



```

1 # DELETE student data by ID
2 @app.route('/students/<int:id>', methods=['DELETE'])
3 def delete_student(id):
4     student = [student for student in students if student['id'] == id]
5     students.remove(student[0])
6     return jsonify({'result': True})
7
8 if __name__ == '__main__':
9     app.run(debug=True)

```

Membuat *route* atau *endpoint* kelima dengan menggunakan decorator `@app.route()`. *Route* ini akan menangani permintaan DELETE untuk menghapus data mahasiswa berdasarkan ID. Terakhir, `name == 'main':`, program akan menjalankan aplikasi Flask dengan menggunakan `app.run(debug=True)`. Debug mode diaktifkan agar memudahkan pemecahan masalah selama pengembangan.

Program 2

```
1 import requests
2
3 # GET all students data
4 response = requests.get('http://localhost:5000/students')
5 print('Mengambil Seluruh Data')
6 print(response.json())
7 print("=====")
8
```

Pertama, mengimpor modul `requests` yang digunakan untuk mengirim permintaan HTTP ke server dan Pada bagian ini, program mengirim permintaan GET ke URL `http://localhost:5000/students` untuk mendapatkan semua data mahasiswa. Respon dari server diterima dan dicetak di konsol menggunakan `response.json()`.

```
1 # GET student data by ID
2 response = requests.get('http://localhost:5000/students/1')
3 print('Mengambil Data dengan ID = 1')
4 print(response.json())
5 print("=====")
6
```

Program mengirim permintaan GET ke URL `http://localhost:5000/students/1` untuk mendapatkan data mahasiswa dengan ID 1. Respon dari server diterima dan dicetak di konsol.

```
1 # CREATE new student data
2 new_student = {'id': 4, 'name': 'Prastowo', 'age': 21}
3 response = requests.post('http://localhost:5000/students', json=new_student)
4 print('Menambahkan Data Baru ke Json')
5 print(response.json())
```

Program mengirim permintaan POST ke URL `http://localhost:5000/students` dengan data mahasiswa baru yang ditentukan dalam variabel `new_student`. Data mahasiswa dikirim dalam format JSON menggunakan argumen `json`. Respon dari server diterima dan dicetak di konsol.

```
1 # UPDATE student data by ID
2 update_student = {'name': 'Jordan', 'age': 19}
3 response = requests.put('http://localhost:5000/students/2', json=update_student)
4 print('Memperbarui Data pada ID = 2')
5 print(response.json())
```

Program mengirim permintaan PUT ke URL `http://localhost:5000/students/2` dengan data pembaruan yang ditentukan dalam variabel `update_student`. Data pembaruan dikirim dalam format JSON menggunakan argumen `json`. Respon dari server diterima dan dicetak di konsol.

```
1 # DELETE student data by ID
2 response = requests.delete('http://localhost:5000/students/3')
3 print('Menghapus Data dengan ID = 3')
4 print(response.json())
```

Program mengirim permintaan DELETE ke URL `http://localhost:5000/students/3` untuk menghapus data mahasiswa dengan ID 3 dari parameter pada endpoint tersebut. Respon dari server diterima dan dicetak di konsol.

OUTPUT

Program 1

```
1 (farisrasyid@jendela)-[~/Documents/FTI/AP2B/pertemuan 5]
2 $ /bin/python3.10 "/home/farisrasyid/Documents/FTI/AP2B/pertemuan 5/P5_Code1_AP2B.py" * Serving Flask app 'P5_Code1_AP2B' (lazy loading) * Environment: production
3 WARNING: This is a development server. Do not use it in a production deployment.
4 Use a production WSGI server instead.
5 * Debug mode: on
6 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
7 * Restarting with stat
8 * Debugger is active!
9 * Debugger PIN: 758-032-658
```

Program 2

```
1 Mengambil Seluruh Data
2 [{u'age': 19, u'id': 1, u'name': u'Rahmad'}, {u'age': 22, u'id': 2, u'name': u'Wicaksono'}, {u'age': 20, u'id': 3, u'name': u'Adi'}]
3 =====
4 Mengambil Data dengan ID = 1
5 [{u'age': 19, u'id': 1, u'name': u'Rahmad'}]
6 =====
7 Menambahkan Data Baru ke Json
8 {u'age': 21, u'id': 4, u'name': u'Prastowo'}
9 [{u'age': 19, u'id': 1, u'name': u'Rahmad'}, {u'age': 22, u'id': 2, u'name': u'Wicaksono'}, {u'age': 20, u'id': 3, u'name': u'Adi'}, {u'age': 21, u'id': 4, u'name': u'Prastowo'}]
10 =====
11 Memperbarui Data pada ID = 2
12 {u'age': 19, u'id': 2, u'name': u'Jordan'}
13 [{u'age': 19, u'id': 1, u'name': u'Rahmad'}, {u'age': 19, u'id': 2, u'name': u'Jordan'}, {u'age': 20, u'id': 3, u'name': u'Adi'}, {u'age': 21, u'id': 4, u'name': u'Prastowo'}]
14 =====
15 Menghapus Data dengan ID = 3
16 {u'result': True}
17 [{u'age': 19, u'id': 1, u'name': u'Rahmad'}, {u'age': 19, u'id': 2, u'name': u'Jordan'}, {u'age': 21, u'id': 4, u'name': u'Prastowo'}]
18
```