

DOKUMENTASI API DATABASE USER

ENDPOINT	DESKRIPSI
/user method 'GET'	Mendapatkan seluruh data user dari database user
/user method 'POST'	Input data user baru ke dalam database user

PARAMETER PERMINTAAN

PARAMETER	TIPE DATA	DESKRIPSI
nama	Varchar	Nama user
umur	Int	Umur user
alamat	Varchar	Alamat user

MEMBUAT DATABASE DENGAN FRAMEWORK FLASK DAN KONFIGURASI DATABASE MYSQL

```
from flask import Flask, jsonify, request
from flask_mysql import MySQL

app = Flask(__name__)

app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ""
app.config['MYSQL_DB'] = 'user'
mysql = MySQL(app)
```

RESPONSE JSON

Response JSON yang diberikan oleh endpoint /user adalah sebagai berikut:

Dengan Method == 'GET'

- Status Code 200 saat seluruh data berhasil ditampilkan
- Seluruh data yang sudah direquest sebelumnya dalam bentuk dictionary

```
@app.route('/user', methods=['GET', 'POST'])
def user():
    if request.method == 'GET':
        cursor = mysql.connection.cursor()
        cursor.execute("SELECT * FROM DATA_USER")

        #mendapatkan nama kolom dari cursor.description
        column_names = [i[0] for i in cursor.description]

        #fetch data dan format ke dalam list dictionary
        data = []
        for row in cursor.fetchall():
            data.append(dict(zip(column_names, row)))
        cursor.close()
        return jsonify(data)
```

Yang menghasilkan data sebagai berikut

User ID	Name	Address	Age
3	Putra	DKI Jakarta	27
4	Dedi	Kalimantan	30
5	Alfian	Jl. Cempaka Raya A4 No.11B	30
6	Alfian	Jl. Cempaka Raya A4 No.11B	30
7	Aji	Bandung	25
8	Dodi	Bandung	25
9	Dodi	Bandung	25
10	Aji	Malang	31
11	Aiu	Malang	31

Dengan Method == 'POST'

- Pesan error saat ada data yang nilainya belum diisi
- Pesan error saat nilai suatu parameter salah contohnya umur harus diisi dengan int
- Pesan data berhasil dimasukkan jika seluruh data dimasukan dengan benar.

Contoh request sebagai berikut

```
elif request.method == 'POST':  
    #get data dari request  
    nama = request.json['nama']  
    umur = request.json['umur']  
    alamat = request.json['alamat']  
  
    if not nama or not umur or not alamat:  
        return jsonify({'error': 'Field nama, umur, dan alamat wajib diisi'})  
  
    try:  
        umur = int(umur)  
    except ValueError:  
        return jsonify({'error': 'Umur harus berupa angka (integer)'})
```

POST localhost:50/user

Params Authorization Headers (8) Body Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☒ JSON

```
1 {  
2   "nama": "Alfian",  
3   "umur": "",  
4   "alamat": "Jakarta"  
5 }
```

Menampilkan pesan error ketika ada data yang belum diisi oleh user

Body Cookies Headers (5) Test Results

{ } JSON Preview Visualize

```
1 {  
2   "error": "Field nama, umur, dan alamat wajib diisi"  
3 }
```

Akan menampilkan pesan error saat user tidak memasukan umur dengan tipe data yang benar

```
1 {
2   "nama": "Alfian",
3   "umur": "Dua Tujuh",
4   "alamat": "Jakarta"
5 }
```

Body Cookies Headers (5) Test Results | ↺

{ } JSON ▾ ▶ Preview 📄 Visualize ▾

```
1 {
2   "error": "Umur harus berupa angka (integer)"
3 }
```

```
#eksekusi query insert (method post)

.....cursor = mysql.connection.cursor()

.....sql = "INSERT INTO DATA_USER (nama, umur, alamat) VALUES (%s, %s, %s)"

.....cursor.execute(sql, (nama, umur, alamat))

.....mysql.connection.commit()

.....cursor.close()

.....return jsonify({'message': 'Data user berhasil dimasukkan!'})
```

Menampilkan pesan berupa “Data user berhasil dimasukkan” ketika semua data yang diinput sudah sesuai

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▾

```
1 {
2   "nama": "Alfian",
3   "umur": "27",
4   "alamat": "Jakarta"
5 }
```

Body Cookies Headers (5) Test Results | ↺

{ } JSON ▾ ▶ Preview 📄 Visualize ▾

```
1 {
2   "message": "Data user berhasil dimasukkan!"
3 }
```