

Muhammad Alfin Rahmadani

### SP 3.1

Library Requests adalah alat yang sangat penting untuk melakukan komunikasi HTTP dalam Python. Mari saya jelaskan secara detail:

#### 1. Fungsi Utama:

- Library Requests digunakan untuk mengirim permintaan HTTP ke website atau API
- Memungkinkan kita untuk mengambil data dari internet secara programatis
- Mempermudah interaksi dengan web services
- Menangani berbagai jenis response (HTML, JSON, dll)
- 

#### 2. Syntax Dasar:

```
import requests
```

```
response = requests.method('URL', params={key: value})
```

Dimana:

- method adalah jenis request yang ingin dilakukan
- URL adalah alamat website target
- params adalah parameter opsional yang bisa ditambahkan

#### 3. Metode-metode HTTP Request yang tersedia:

- `requests.get()`: Mengambil data dari server (paling umum digunakan)
- `requests.post()`: Mengirim data ke server
- `requests.put()`: Memperbarui data yang sudah ada di server
- `requests.delete()`: Menghapus data dari server
- `requests.patch()`: Memperbarui sebagian data
- `requests.head()`: Mengambil header saja tanpa konten

#### 4. Fitur-fitur Penting:

- Penanganan Response:

`response.text` # Mendapatkan konten dalam bentuk text

`response.json()` # Mendapatkan konten dalam format JSON

`response.status_code` # Mengecek status response (200, 404, dll)

`response.headers` # Melihat header responseKeunggulan:

- Syntax yang sederhana dan mudah dipahami
- Otomatis menangani encoding
- Mendukung berbagai metode autentikasi
- Dapat menangani cookies dan session
- Mendukung SSL/TLS untuk keamanan

#### 6. Contoh Penggunaan Praktis:

# Contoh basic request

```
response = requests.get('https://www.example.com')
```

# Request dengan parameter

```
params = {'key1': 'value1', 'key2': 'value2'}
```

```
response = requests.get('https://api.example.com', params=params)
```

# Request dengan header kustom

```
headers = {'User-Agent': 'Mozilla/5.0'}
```

```
response = requests.get('https://www.example.com', headers=headers)
```

Library Requests sangat berguna dalam web scraping karena memungkinkan kita untuk:

- Mengakses halaman web secara programatis
- Mengambil data dari API
- Mengirim data ke server
- Menangani berbagai jenis response dengan mudah

- Melakukan otentikasi jika diperlukan

Dengan library ini, proses pengambilan data dari internet menjadi jauh lebih sederhana dibandingkan dengan menggunakan library bawaan Python seperti urllib. Syntax yang intuitif dan dokumentasi yang lengkap membuatnya menjadi pilihan utama untuk melakukan HTTP requests dalam Python.