

Praktikum #1

Mata Kuliah :Web Service Engineering

Dosen Pengampu :Muhayat, M.IT

Praktikum :P1 - Demo Testing API

Nama Mahasiswa :Ruqayah

NIM :230104040208

Kelas :TI23A

Tanggal Praktikum :15-09-2025

A. Tujuan Praktikum

1. Mengenal konsep dasar API (Application Programming Interface).
2. Melakukan pengujian request HTTP (GET, POST, PUT, DELETE) menggunakan Postman.
3. Menganalisis perbedaan hasil request terhadap server API JSONPlaceholder.

B. Dasar Teori

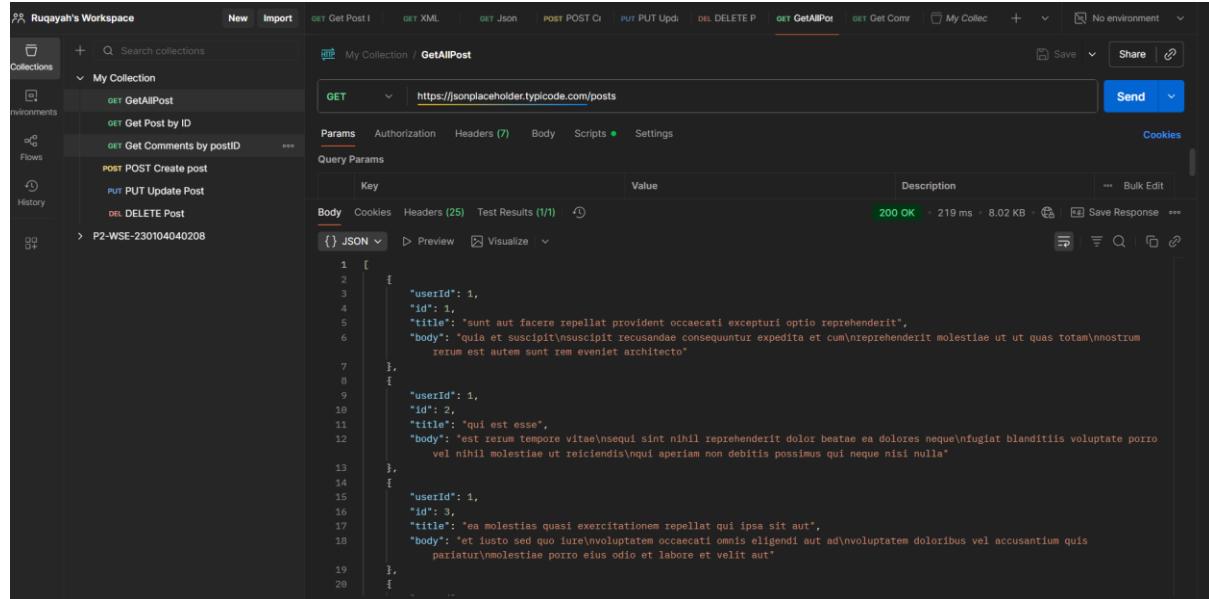
- **API** adalah antarmuka yang memungkinkan komunikasi antar sistem atau aplikasi.
- **HTTP Methods:**
 - **GET** → Mengambil data dari server.
 - **POST** → Menambahkan data baru ke server.
 - **PUT** → Memperbarui data yang sudah ada.
 - **DELETE** → Menghapus data dari server.
- **JSON (JavaScript Object Notation)** adalah format data ringan berbentuk key–value yang digunakan secara luas dalam API modern.

C. Langkah Percobaan dan Hasil

1. GET All Post

- Endpoint: <https://jsonplaceholder.typicode.com/posts>
- Method: **GET**

- Hasil: Berhasil menampilkan seluruh data postingan (100 data).



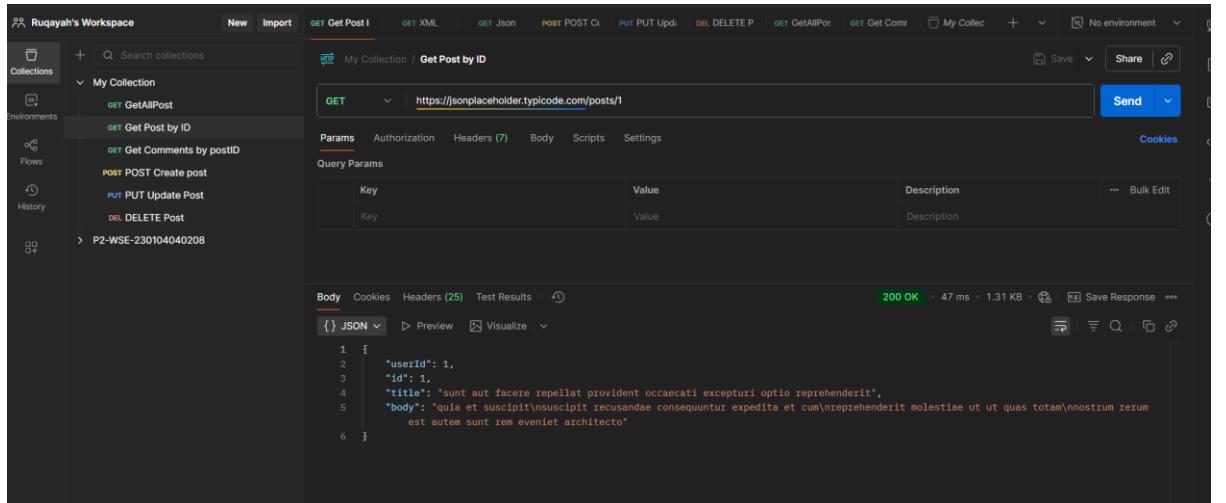
```

1 [
2   {
3     "userId": 1,
4     "id": 1,
5     "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
6     "body": "quia et suscipit\\nuscipit recusandae consequuntur expedita et cum\\nreprehenderit molestiae ut ut quas totam\\nnostrum\\nrum est autem sunt rem eveniet architecto"
7   },
8   {
9     "userId": 1,
10    "id": 2,
11    "title": "qui est esse",
12    "body": "est rerum tempore vitae\\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\\nqui aperiam non debitis possimus qui neque nisi nulla"
13 },
14 {
15   "userId": 1,
16   "id": 3,
17   "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut",
18   "body": "et iusto sed quo iure\\nvolutatem occaecati omnis eligendi aut ad\\nvolutatem doloribus vel accusantium quis pariatur\\nmolestiae porro eius odio et labore et velit aut"
19 },
20 ...
]

```

2. GET Post by ID

- Endpoint: <https://jsonplaceholder.typicode.com/posts/1>
- Method: **GET**
- Hasil: Berhasil menampilkan 1 data postingan dengan ID = 1.



```

1 {
2   "userId": 1,
3   "id": 1,
4   "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
5   "body": "quia et suscipit\\nuscipit recusandae consequuntur expedita et cum\\nreprehenderit molestiae ut ut quas totam\\nnostrum\\nrum est autem sunt rem eveniet architecto"
6 }

```

3. GET Comments by PostID

- Endpoint: <https://jsonplaceholder.typicode.com/comments?postId=1>
- Method: **GET**

- Hasil: Berhasil menampilkan daftar komentar berdasarkan postId=1.

Postman screenshot showing a successful GET request to `https://jsonplaceholder.typicode.com/comments?postId=1`. The response is a JSON array with two elements:

```

1 [
2   {
3     "postId": 1,
4     "id": 1,
5     "name": "id labore ex et quam laborum",
6     "email": "Eliseo@garner.biz",
7     "body": "laudantium enim quasi est quidem magnam voluptate ipsam eos\\ntempora quo necessitatibus\\ndolor quam autem
9       quasi\\nxiciendis et nam sapiente accusantium"
8   },
10  {
11    "postId": 1,
12    "id": 2,
13    "name": "quo vero reiciendis velit similique earum",
14    "email": "Jayne_Kuhic@snyderry.com",
15    "body": "est natus enim nihil est dolore omnis voluptatem numquam\\net omnis occaecati quod ullam at\\nvolutatem error expedita
16      pariatui\\nnihil sint nostrum voluptatem reiciendis et"
17  },
18  {
19    "postId": 1,
20    "id": 3,
21    "name": "odio adipisci rerum aut animi",
22  }
]

```

4. POST Create Post

- Endpoint: `https://jsonplaceholder.typicode.com/posts`
- Method: **POST**
- Body (JSON):
 - {
 - "title": "Belajar API",
 - "body": "Ini percobaan pertama",
 - "userId": 1
 - }
- Hasil: Berhasil membuat data baru dengan response 201 Created.

Postman screenshot showing a successful POST request to `https://jsonplaceholder.typicode.com/posts`. The response is a JSON object:

```

1 {
2   "title": "Belajar API",
3   "body": "Ini percobaan pertama",
4   "userId": 1,
5   "id": 101
6 }

```

5. PUT Update Post

- Endpoint: <https://jsonplaceholder.typicode.com/posts/1>
- Method: **PUT**
- Body (JSON):


```
{
        "id": 1,
        "title": "Belajar API Update",
        "body": "Isi sudah diperbarui",
        "userId": 1
      }
```
- Hasil: Berhasil memperbarui data pada id=1 dengan status 200 OK.

The screenshot shows the Postman interface with the following details:

- Collection:** Ruqayah's Workspace
- Request Type:** PUT
- URL:** <https://jsonplaceholder.typicode.com/posts/1>
- Body:** Raw JSON (selected)


```
{
        "id": 1,
        "title": "Belajar API Update",
        "body": "Isi sudah diperbarui",
        "userId": 1
      }
```
- Response:** 200 OK (828 ms, 1.19 KB)

6. DELETE Post

- Endpoint: <https://jsonplaceholder.typicode.com/posts/1>
- Method: **DELETE**
- Hasil: Berhasil menghapus data pada id=1 dengan response kosong {} dan status 200 OK.

The screenshot shows the Postman interface with the following details:

- Collection:** Ruqayah's Workspace
- Request Type:** DELETE
- URL:** <https://jsonplaceholder.typicode.com/posts/1>
- Body:** None
- Response:** 200 OK (270 ms, 1.11 KB)

C. Analisis

Dari hasil uji coba CRUD API:

1. **GET** menampilkan data baik seluruh koleksi maupun berdasarkan parameter tertentu.
2. **POST** berhasil membuat data baru, walaupun JSONPlaceholder hanya memberikan simulasi (tidak benar-benar menyimpan data).
3. **PUT** memperbarui data dengan struktur JSON sesuai request.
4. **DELETE** memberikan response kosong namun status 200 OK menandakan operasi berhasil.

Hal ini membuktikan bahwa API mendukung operasi CRUD penuh sesuai standar HTTP methods.

F. Kesimpulan

1. Postman dapat digunakan untuk menguji API dengan berbagai metode HTTP.
2. JSONPlaceholder menyediakan endpoint simulasi CRUD API.
3. Setiap metode HTTP (GET, POST, PUT, DELETE) memiliki fungsi berbeda yang dapat diuji secara praktis.