

# Laporan Praktikum WSE #1

Mata Kuliah : Web Service Engineering  
Dosen Pengampu : Muhayat, M.IT  
Praktikum : P1 - **Demo Testing API**  
Nama Mahasiswa : Husna Norgina  
NIM : 230104040056  
Kelas : TI23B  
Link : <https://github.com/husna-norgina/P1-WSE-230104040056>  
Tanggal Praktikum : 15-09-2025

## A. Tujuan Praktikum

1. Memahami konsep dasar Web Service dan REST API.
2. Melakukan pengujian API menggunakan metode HTTP GET, POST, PUT, dan DELETE.
3. Mengirim request API dan membaca response dalam format JSON.
4. Memahami fungsi masing-masing metode HTTP dalam pertukaran data client-server.
5. Menggunakan Postman sebagai tools pengujian Web Service.

## B. Alat & Bahan

1. Laptop / PC
2. Postman
3. Internet

## C. Langkah Kerja

### 1. Install Postman Opsi

- a. Download Postman Desktop (Windows/Mac/Linux)  
Pastikan buat Collection Baru dan beri nama: P1-WSE-230104040056
- b. Kita gunakan JSONPlaceholder (API dummy gratis untuk belajar)  
Base URL: <https://jsonplaceholder.typicode.com>

### 2. Eksperimen di Browser DevTools

- a. GET – Ambil semua data postingan Klik + New Request → beri nama GET All Posts  
Method: GET URL: <https://jsonplaceholder.typicode.com/posts>  
Klik Send Hasil: muncul JSON array (daftar postingan)  
GET = ambil data (ibarat melihat menu restoran)
- b. GET – Ambil satu data tertentu → ubah nama GET Post by ID  
URL: <https://jsonplaceholder.typicode.com/posts/1>  
Klik Send Hasil: muncul 1 objek JSON dengan id=1
- c. GET – Dengan Query Parameter New Request → nama GET Comments by postId  
Method: GET URL: <https://jsonplaceholder.typicode.com/comments?postId=1>

Klik Send Hasil: daftar komentar dengan postId=1 Query Parameter digunakan untuk memfilter data tertentu.

- d. POST – Menambahkan data baru New Request → nama POST Create Post

Method: POST URL: <https://jsonplaceholder.typicode.com/posts>

Body → pilih raw JSON:

```
{ "title": "Belajar API", "body": "Ini percobaan pertama", "userId": 1 }
```

Klik Send Hasil: hasil: objek baru dengan id tambahan (simulasi)

POST = menambah data (ibarat pesan makanan ke dapur )

- e. PUT – Mengubah data New Request → nama PUT Update Post

Method: PUT URL: <https://jsonplaceholder.typicode.com/posts/1>

Body → raw JSON:

```
{ "id": 1, "title": "Belajar API Update", "body": "Isi sudah diperbarui", "userId": 1 }
```

Klik Send Hasil: data berubah (simulasi)

PUT = memperbarui data (ibarat mengubah pesanan di restoran)

- f. DELETE – Menghapus data New Request → nama DELETE Post

Method: DELETE URL: <https://jsonplaceholder.typicode.com/posts/1>

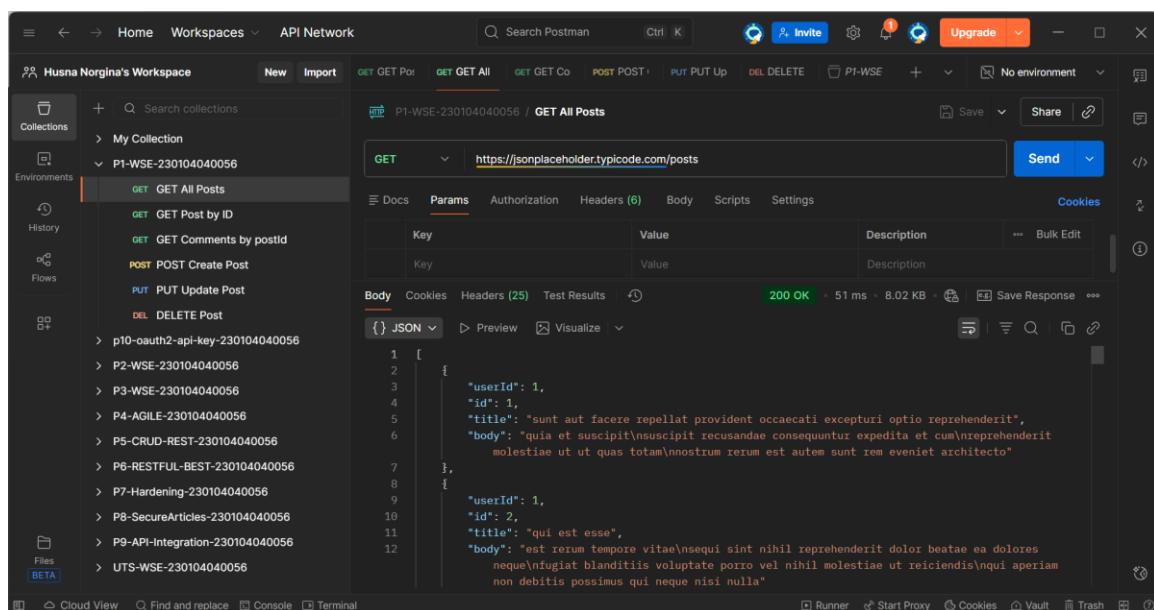
Klik Send Hasil: status 200 OK )

DELETE = menghapus data (ibarat membatalkan pesanan di restoran)

## D. Hasil & Pembahasan

### Hasil Uji

1. GET [jsonplaceholder.typicode.com/posts](https://jsonplaceholder.typicode.com/posts) → Menampilkan daftar postingan (200 OK)



The screenshot shows the Postman application interface. On the left, there's a sidebar with collections, environments, history, and flows. The main area shows a 'My Collection' section with several requests: 'GET All Posts', 'GET Post by ID', 'GET Comments by postId', 'POST POST Create Post', 'PUT PUT Update Post', and 'DEL DELETE Post'. Below this is another collection named 'p10-oauth2-api-key-230104040056' containing requests like 'P1-WSE-230104040056', 'P2-WSE-230104040056', etc. In the center, a specific request is selected: 'P1-WSE-230104040056 / GET All Posts'. The method is 'GET' and the URL is 'https://jsonplaceholder.typicode.com/posts'. The response status is '200 OK' with a response time of '51 ms'. The response body is shown as JSON, displaying two posts:

```
[{"id": 1, "userId": 1, "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit", "body": "quia et suscipit\\nscipit recusandae consequuntur expedita et cum\\nreprehenderit molestiae ut ut quas totam\\nnostrum rerum est autem sunt rem eveniet architecto"}, {"id": 2, "userId": 1, "title": "qui est esse", "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"}]
```

2. GET [jsonplaceholder.typicode.com/posts/1](https://jsonplaceholder.typicode.com/posts/1) → Menampilkan postingan id=1 (200 OK)

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections', 'Environments', 'History', 'Flows', and 'Files (BETA)'. The main area displays a collection named 'P1-WSE-230104040056' under 'My Collection'. A specific POST request titled 'GET Post by ID' is selected. The 'Params' tab is active, showing a key-value pair 'postId' with the value '1'. The 'Body' tab shows a JSON response with one object containing fields: userId (1), id (1), title ("sunt aut facere repellat provident occaecati excepturi optio reprehenderit"), and body ("quia et suscipit\\nscipit recusandae consequuntur expedita et cum\\nreprehenderit molestiae ut ut quas totam\\nnostrum rerum est autem sunt rem eveniet architecto"). The status bar at the bottom indicates a 200 OK response.

3. GET [jsonplaceholder.typicode.com/comments?postId=1](https://jsonplaceholder.typicode.com/comments?postId=1) → Daftar komentar postId=1 (200 OK)

The screenshot shows the Postman application interface. The sidebar and collection structure are identical to the previous screenshot. A new POST request titled 'GET Comments by postId' is selected. The 'Params' tab is active, showing a key-value pair 'postId' with the value '1'. The 'Body' tab shows a JSON response with two objects. Each object has fields: postId (1), id (1), name ("id labore ex et quam laborum"), email ("Elisoe@gardner.biz"), and body ("laudantium enim quasi est quidem magnam voluptate ipsam eos\\ntempora quo necessitatibus\\ndolor quam autem quasi\\nreiciendis et nam sapiente accusantium"). The status bar at the bottom indicates a 200 OK response.

#### 4. POST <https://jsonplaceholder.typicode.com/posts> → Menambah objek baru (201 Created)

The screenshot shows the Postman interface with the following details:

- Collection:** Husna Norgina's Workspace
- Request:** POST /POST Create Post
- URL:** <https://jsonplaceholder.typicode.com/posts>
- Body:** Raw JSON (selected)

```
1 {
2   "title": "Belajar API",
3   "body": "Ini percobaan pertama",
4   "userId": 1
5 }
```
- Response Status:** 201 Created
- Response Body:** Raw JSON

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

#### 5. PUT <https://jsonplaceholder.typicode.com/posts/1> → Mengubah data (200 OK)

The screenshot shows the Postman interface with the following details:

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

```
1 {
2   "id": 1,
3   "title": "Belajar API Update",
4   "body": "Isi sudah diperbarui",
5   "userId": 1
6 }
```
- Response Status:** 200 OK
- Response Body:** Raw JSON

```
1 {
2   "id": 1,
3   "title": "Belajar API Update",
4   "body": "Isi sudah diperbarui",
5   "userId": 1
6 }
```

## 6. DELETE <https://jsonplaceholder.typicode.com/posts/1> → Menghapus data (200 OK)

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Husna Norgina's Workspace' containing collections, environments, history, flows, and files. The main area shows a collection named 'P1-WSE-230104040056' with several requests: 'GET All Posts', 'GET Post by ID', 'GET Comments by postID', 'POST Create Post', 'PUT Update Post', and 'DEL DELETE Post'. The 'DEL DELETE Post' request is selected. The URL in the header is 'https://jsonplaceholder.typicode.com/posts/1'. The 'Params' tab is active, showing a table with one row: 'Key' (Key) and 'Value' (Value). Below the table, the response status is '200 OK' with a response body of '{}'. Other tabs like 'Body', 'Cookies', 'Headers', and 'Test Results' are also visible.

## Pembahasan

Berdasarkan hasil praktikum yang telah dilakukan menggunakan Postman, dapat diketahui bahwa metode HTTP GET, POST, PUT, dan DELETE berfungsi sesuai dengan konsep REST API. Metode GET berhasil menampilkan data, baik seluruh data maupun data berdasarkan ID dan query parameter. Metode POST berhasil mensimulasikan penambahan data baru, metode PUT digunakan untuk memperbarui data yang sudah ada, dan metode DELETE digunakan untuk menghapus data berdasarkan ID tertentu. Seluruh response yang diterima menggunakan format JSON, sehingga mudah dibaca dan dipahami. JSONPlaceholder sebagai API dummy membantu dalam proses pertukaran data.

## E. Kesimpulan

Praktikum 1 berhasil memperkenalkan proses pengujian Web Service menggunakan Postman dengan metode HTTP GET, POST, PUT, dan DELETE pada API JSONPlaceholder. Setiap metode diuji sesuai fungsinya, di mana GET digunakan untuk mengambil data, POST untuk menambahkan data baru, PUT untuk memperbarui data, dan DELETE untuk menghapus data berdasarkan ID tertentu. Seluruh response yang diterima menggunakan format JSON sehingga mudah dibaca dan dipahami. Hasil pengujian menunjukkan bahwa pertukaran data antara client dan server berjalan dengan baik, meskipun perubahan data yang dilakukan bersifat simulasi.