MODUL 04

GET METHOD

A. Capaian Pembelajaran

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

1. Menangani permintaan HTTP GET pada sisi server menggunakan PHP.
2. Menerapkan penggunaan prepared statement untuk mencegah SQL Injection.
3. Mengembalikan respons HTTP yang sesuai berdasarkan hasil eksekusi operasi database.
4. Menghasilkan output dalam format JSON untuk komunikasi API.

B. Review Materi POST Method

Salah satu prinsip CRUD yang digunakan RESTful API adalah fungsi Read atau biasa dinamakan metode GET. GET digunakan ketika ingin mengambil, membaca dan menampilkan data pada pengguna. GET pada RESTful API akan mengambil data yang diinginkan dan menampilkannya oleh pengguna.

C. Deskripsi Singkat

Modul ini membimbing mahasiswa untuk membuat endpoint GET yang mengambil data pengguna dari dalam database users menggunakan PHP. Praktikum ini juga membahas pengolahan input, pengamanan melalui prepared statement, dan pengiriman respons dalam format JSON.

D. Persiapan

1. Pastikan telah memiliki web server lokal (XAMPP, Laragon, dsb).
2. Database MySQL sudah aktif.
3. Database **test_db** dan tabel **users** telah dibuat.
4. Telah membuat folder praktikum-api dengan file **db.php** yang berisi koneksi ke database

E. Membuat API GET Method

1. Buat file baru untuk membuat API dengan nama **get.php** di dalam folder praktikum-api yang telah dibuat sebelumnya.
2. Set Header response

   Tambahkan instruksi PHP untuk memastikan response dari server berupa JSON:

   ```php
   header('Content-Type: application/json');
   ```

3. Panggil file koneksi database menggunakan **`require`** dan diikuti **nama file koneksi**
4. Deklarasi variabel untuk menyimpan ID yang dikirim oleh client menggunakan **$_GET**

   ```php
   $id = $_GET['id'] ?? null;
   ```

5. Buat validasi input untuk variabel **id**

```php
if (!$id) {
    // Get by ID
} else {
    // Get all users
}
```

6. Buat prepared statement berupa query mysql untuk mengambil data dari database. Buat di dalam validasi input bagian Get by ID yang dibuat sebelumnya.

```php
$stmt = $conn->prepare("SELECT id, name, email FROM users WHERE id = ?");
```

7. Selanjutnya buatlah syntax untuk mengikat tipe data dan variabel yang telah dibuat menjadi sebuah parameter.

```php
$stmt->bind_param("i", $id);
```

8. Buat syntax untuk mengeksekusi parameter yang telah dibuat dengan menulis **`$stmt`** lalu diikuti dengan **`execute();`**
9. Buat variabel result untuk menyimpan hasil dari query yang dibuat sebelumnya

```php
$result = $stmt->get_result();
```

10. Buat variabel user untuk menyimpan **satu baris** dari result sebagai array asosiatif.

```php
$user = $result->fetch_assoc();
```

11. Buat syntax untuk mengeksekusi parameter yang telah dibuat dengan menulis **`$stmt`** lalu diikuti dengan **`close();`** untuk menutup statement
12. Masih didalam fungsi validasi input, buat pengecekan input data dengan **if** untuk mengembalikan respons sukses atau gagal.

```php
if ($user) {
    // respons suskses
} else {
    // respons gagal
}
```

13. Pada bagian untuk respons sukses tulislah **`http_response_code(200);`** untuk memberikan kode **200 OK**.
14. Lalu buat kode untuk mengembalikan pesan sukses seperti di bawah ini

```php
echo json_encode([
    // pesan sukses
]);
```

15. Isi pesan sukses dengan
    - **status = success**
    - **data = $user**
16. Setelah itu tulislah `http_response_code(404);` pada bagian respons gagal untuk mengembalikan kode **404 Not Found**. Lalu buat kode untuk mengembalikan pesan gagal seperti di bawah ini.

```
echo json_encode([
    // pesan error
]);
```

17. Isi pesan error dengan
    - **status = error**
    - **message = User tidak ditemukan**
18. Selanjutnya buat prepared statement berupa query mysql untuk mengambil data dari database. Buat di dalam validasi input bagian Get all users (Lihat langkah nomor 5). yang dibuat sebelumnya.

```
$result = $conn->query("SELECT id, name, email FROM users");
```

19. Buat variabel user untuk menyimpan **semua baris** dari result sebagai array asosiatif.

```
$users = $result->fetch_all(MYSQLI_ASSOC);
```

20. Setelah itu buat respons sukses dengan menulis `http_response_code(200);` untuk memberikan kode **201 OK**.
21. Lalu buat kode untuk mengembalikan pesan sukses seperti di bawah ini

```
echo json_encode([
    // pesan sukses
]);
```

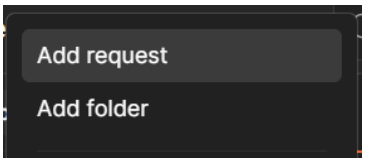
22. Isi pesan sukses dengan
    - **status = success**
    - **data = $users**
23. Terakhir tulis kode untuk menutup koneksi dengan database

```
$conn->close();
```

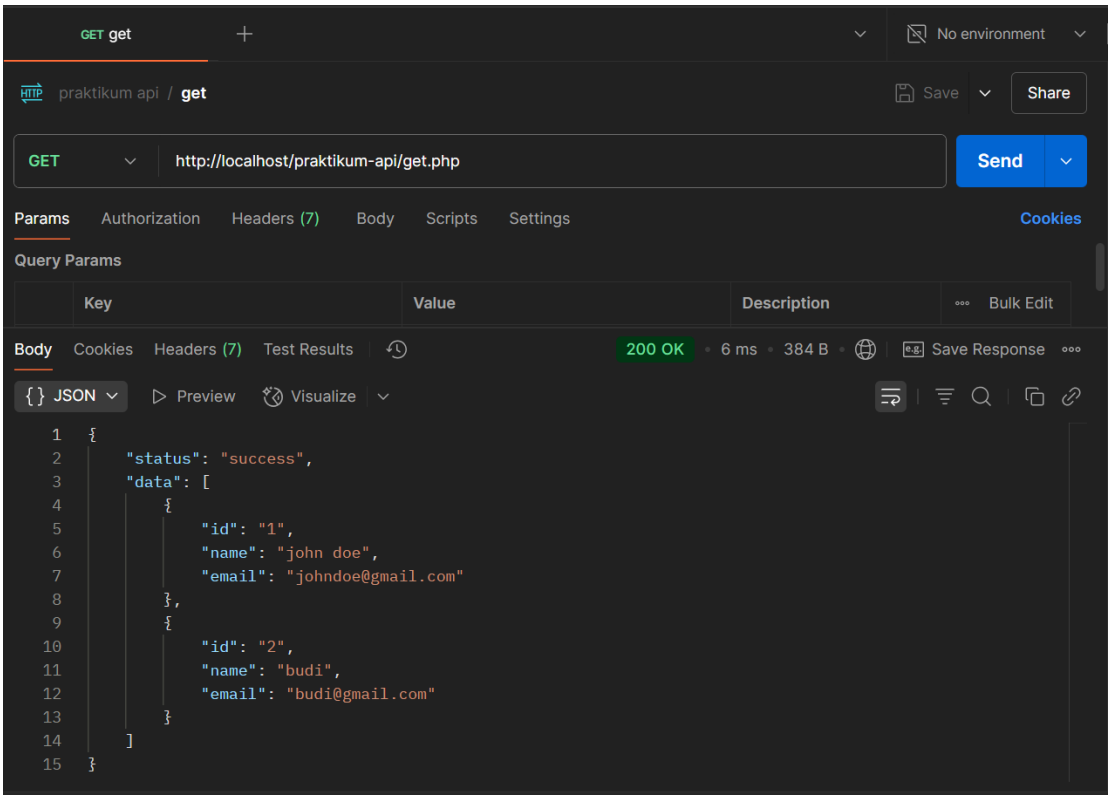
F. Pengujian Postman

1. Buka aplikasi postman
2. Buka **Collection** Praktikum API yang telah dibuat sebelumnya

3. Klik icon titik tiga pada collection dan pilih **Add Request**



4. Beri nama **get** untuk request yang baru saja dibuat.
5. Selanjutnya atur **method** menjadi **GET** dan ketikkan **url** lokal diikuti lokasi file **get.php**
6. Klik tombol **Send** di pojok kanan atas. Respons dari API akan muncul seperti gambar berikut



7. Jika ingin mengambil satu data tertentu saja maka tambahkan **?id={id}** seperti gambar berikut