MODUL 04

RESTful API - GET

Materi

1. RESTful API

REST *(Representational State Transfer)* adalah arsitektur API *(Application Programming Interface)* yang menyediakan komunikasi *client-server* untuk aplikasi web melalui protokol HTTP. RESTful API merupakan implementasi dari REST API yang benar-benar mengikuti semua prinsip REST dengan baik. RESTful API beroperasi berdasarkan prinsip CRUD *(Create, Read, Update, Delete)*.

2. GET

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.

Deskripsi Tugas

Pada praktikum ini, mahasiswa akan mempraktikkan pembuatan fungsi GET dalam RESTful API. Mahasiswa akan belajar memahami struktur kode program yang paling mendasar dari pembuatan RESTful API dengan fungsi GET.

Praktikum

No.	Langkah-langkah
1.	Pertama buat buat file dengan nama “post.php” di dalam folder “api” yang telah dibuat sebelumnya `<?php` `header("Content-Type: application/json");` `require "data.php";` **`header("Content-Type: application/json");`** ➢ Menetapkan **header response** agar output yang dikirimkan ke client berupa **JSON**.

	`require "data.php";` ➢ Mengimpor file **data.php**, yang berisi fungsi **`getData()`** dan **`saveData()`**, digunakan untuk membaca dan menyimpan data dalam format JSON.
2.	```php $data = getData(); $id = isset($_GET['id']) ? $_GET['id'] : null; ``` **`$data = getData();`** ➢ Memanggil fungsi **`getData()`** untuk membaca isi file JSON dan menyimpannya dalam variabel **`$data`** sebagai array. **`$id = isset($_GET['id']) ? $_GET['id'] : null;`** ➢ Mengecek apakah ada parameter id yang dikirim melalui **query string** (misalnya ?id=1). Jika id ada, nilainya disimpan ke dalam **`$id`**, jika tidak, maka **`$id`** akan bernilai null.
3.	```php if (is_null($id)) { http_response_code(200); echo json_encode(["status" => 200, "data" => array_values($data)]); exit; } ``` **`if(is_null($id))`** ➢ Jika **`$id`** bernilai null, maka API akan mengembalikan **seluruh data** dalam JSON. **`http_response_code(200);`** ➢ Memastikan response memiliki **status HTTP 200 OK**. **`array_values($data)`** ➢ Digunakan untuk mengonversi array asosiatif menjadi array numerik, agar format JSON lebih rapi.
4.	```php if (isset($data[$id])) { http_response_code(200); echo json_encode(["status" => 200, "data" => $data[$id]]); exit; } ``` **`if(is_null($id))`** ➢ Jika **`$id`** ada dalam **`$data`**, maka API mengembalikan data item tersebut.

	`http_response_code(200);` ➢ Memastikan status **200 OK** dikirim dalam response.
5.	```php http_response_code(404); echo json_encode(["status" => 404, "error" => "Item Not Found"], JSON_PRETTY_PRINT); exit; ``` Jika id **tidak ditemukan dalam data**, API mengembalikan **404 Not Found**. **`JSON_PRETTY_PRINT`** ➢ Membuat output JSON lebih mudah dibaca.