

MODUL 02

Data Preparation

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*)

Praktikum

Sebelum memulai pembuatan API dibutuhkan persiapan data yang perlu dibuat untuk mendukung pelaksanaan praktikum yang akan dilakukan.

No.	Langkah-langkah
1.	Pertama buat folder dengan nama “api”, lalu di dalam folder tersebut buat file dengan nama “data.php”
2.	<p>Buka file data.php, lalu ketikkan kode di bawah ini</p> <pre><?php \$file = "data.json";</pre> <p>\$file merupakan nama variabel yang akan menampung path file yang digunakan sebagai pengganti database</p>
3.	<p>Selanjutnya ketikkan kode berikut untuk melakukan pembuatan data</p> <pre>if (!file_exists(\$file)) { file_put_contents(\$file, json_encode(["1" => ["id" => 1, "name" => "Item A"], "2" => ["id" => 2, "name" => "Item B"], "3" => ["id" => 3, "name" => "Item C"]], JSON_PRETTY_PRINT)); }</pre> <p>if(!file_exists(\$file))</p> <ul style="list-style-type: none">➤ mengecek apakah file dengan path yang disimpan dalam variabel \$file sudah ada atau belum. Jika file tidak ada, maka perintah dalam blok { ... } akan dieksekusi.

	<p>file_put_contents(\$file, ...)</p> <ul style="list-style-type: none"> ➤ menyimpan data JSON yang dihasilkan ke dalam file yang ditentukan oleh \$file. <p>json_encode(..., JSON_PRETTY_PRINT)</p> <ul style="list-style-type: none"> ➤ Mengubah array PHP ke dalam format JSON dengan format yang lebih rapi.
4.	<p>Fungsi ini digunakan untuk membaca data dari file JSON dan mengembalikannya dalam bentuk array PHP.</p> <pre>function getData() { global \$file; return json_decode(file_get_contents(\$file), true); }</pre> <p>global \$file;</p> <ul style="list-style-type: none"> ➤ Menggunakan variabel \$file yang dideklarasikan di luar fungsi agar bisa diakses di dalam fungsi. <p>return ...;</p> <ul style="list-style-type: none"> ➤ Mengembalikan data yang telah dibaca dan didekode dari file JSON. <p>json_decode(..., true)</p> <ul style="list-style-type: none"> ➤ Mengubah data JSON dari string menjadi array asosiatif (true menandakan hasilnya berupa array, bukan objek). <p>file_get_contents(\$file)</p> <ul style="list-style-type: none"> ➤ Membaca isi file JSON yang disimpan di path \$file.
5.	<p>Fungsi ini digunakan untuk menyimpan data ke dalam file JSON setelah diubah dari array PHP menjadi format JSON.</p> <pre>function saveData(\$data) { global \$file; file_put_contents(\$file, json_encode(\$data, JSON_PRETTY_PRINT)); }</pre> <p>global \$file;</p> <ul style="list-style-type: none"> ➤ Menggunakan variabel \$file yang ada di luar fungsi agar bisa diakses dalam fungsi. <p>file_put_contents(\$file, ...)</p> <ul style="list-style-type: none"> ➤ Menyimpan JSON hasil encoding ke dalam file yang ditentukan oleh \$file. <p>json_encode(\$data, JSON_PRETTY_PRINT)</p> <ul style="list-style-type: none"> ➤ Mengubah array PHP menjadi format JSON yang rapi dan mudah dibaca (JSON_PRETTY_PRINT menambahkan indentasi).