

## MODUL 03

### RESTful API - POST

#### 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. POST

Salah satu prinsip CRUD yang digunakan RESTful API adalah fungsi *Create* atau biasa dinamakan metode POST. POST digunakan ketika ingin membuat atau menambah data baru ke dalam sistem. POST pada RESTful API akan membuat data baru yang dimasukkan oleh pengguna dan mengirimkannya ke server untuk disimpan.

#### Deskripsi Tugas

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

#### Praktikum

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

	<pre><b>require "data.php";</b></pre> <ul style="list-style-type: none"> <li>➤ Mengimpor file <b>data.php</b>, yang berisi fungsi <b>getData()</b> dan <b>saveData()</b>, digunakan untuk membaca dan menyimpan data dalam format JSON.</li> </ul>
2.	<p>Memanggil fungsi <b>getData()</b> untuk membaca data dari file JSON dan menyimpannya dalam variabel <b>\$data</b>.</p> <pre><b>\$data = getData();</b></pre>
3.	<p>Ketikkan kode di bawah untuk membaca input dari request body</p> <pre><b>\$input = json_decode(file_get_contents("php://input"), true);</b></pre> <p><b>file_get_contents("php://input")</b></p> <ul style="list-style-type: none"> <li>➤ Membaca raw JSON dari request body (biasanya dikirim dalam permintaan HTTP POST atau PUT).</li> </ul> <p><b>json_decode(..., true)</b></p> <ul style="list-style-type: none"> <li>➤ Mengubah JSON menjadi array asosiatif PHP.</li> </ul>
4.	<p>Buat validasi input datanya</p> <pre><b>if (!isset(\$input['id'])    !isset(\$input['name'])) {     http_response_code(400);     echo json_encode([         "status" =&gt; 400,         "error" =&gt; "Invalid Input"     ]);     exit; }</b></pre> <p><b>if(!isset(\$input['id'])    !isset(\$input['name']))</b></p> <ul style="list-style-type: none"> <li>➤ Mengecek apakah input <b>tidak memiliki id atau name</b>.</li> <li>➤ Jika salah satu tidak ada, kirim <b>status 400, error response "invalid input"</b> dan hentikan eksekusi dengan <b>exit</b>.</li> </ul> <p><b>http_response_code(400);</b></p> <ul style="list-style-type: none"> <li>➤ Mengatur status HTTP response 'Bad Request' yang dikirim ke klien (browser atau aplikasi yang memanggil API).</li> </ul>
5.	<p>Buat validasi pengecekan untuk memeriksa apakah id telah ada atau belum</p>

	<pre> if (isset(\$data[\$input['id']])) {     http_response_code(400);     echo json_encode([         "status" =&gt; 400,         "error" =&gt; "Item Already Exists"     ]);     exit; } </pre> <p>Jika <b>ID</b> sudah ada dalam <b>\$data</b>, maka kirim <b>status 400</b>, <b>error response</b> dan hentikan eksekusi.</p> <p><b>http_response_code(400) ;</b></p> <ul style="list-style-type: none"> <li>➤ Mengatur status HTTP response 'Bad Request' yang dikirim ke klien (browser atau aplikasi yang memanggil API).</li> </ul>
6.	<p>Simpan data yang diinputkan dengan kode berikut</p> <pre> \$data[\$input['id']] = \$input; saveData(\$data); </pre> <ul style="list-style-type: none"> <li>➤ Menyimpan data baru ke array <b>\$data</b> dengan <b>ID</b> sebagai <b>key</b>.</li> <li>➤ Memanggil <b>saveData(\$data)</b> untuk menyimpan perubahan ke file JSON.</li> </ul>
7.	<p>Buat respons untuk data yang sukses dibuat</p> <pre> http_response_code(201); echo json_encode([     "status" =&gt; 201,     "message" =&gt; "Item Created",     "produk" =&gt; \$data[\$input['id']] ]); </pre> <p><b>http_response_code(201) ;</b></p> <ul style="list-style-type: none"> <li>➤ Mengatur status HTTP response 'Created' yang dikirim ke klien (browser atau aplikasi yang memanggil API).</li> </ul> <p>Kode di atas akan mengirim respons dalam format <b>JSON</b> setelah item berhasil ditambahkan.</p>