

MODUL 02

RESTful API - GET

Tujuan Pembelajaran

1. Mahasiswa memahami konsep RESTful API
2. Mahasiswa mampu membuat fungsi GET dengan RESTful API

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. Struktur kode program tersebut memiliki 2 susunan kode, yaitu kode dengan respons sukses dan respons gagal. Struktur kode yang pertama terdiri dari pembuatan fungsi *public* untuk prinsip *Read*, diikuti dengan fungsi *if* yang diperlukan untuk validasi data yang akan diambil serta diakhiri dengan respons gagal. Sedangkan struktur kode kedua diisi dengan respons sukses.

Langkah Praktikum:

1. Deklarasi **class** untuk bagian awal dari penulisan kode, tulis seperti contoh di bawah ini

```
1  <?php
2  class CaseRead {
3
4  }
5  ?>
```

2. Buat sebuah variabel **private** sebagai properti dari **class** dengan nama **\$data** dan letakkan di dalam **class**. Variabel data akan menyimpan daftar item yang akan dimasukkan dalam bentuk **array**.

```
2  class CaseRead {
3      private $data = [];
4  }
```

3. Setelah itu buatlah fungsi **public** untuk melakukan **read** data. Fungsi ini berada setelah variabel **private** yang telah dibuat sebelumnya. Fungsi ini menerima satu parameter **\$id** yang merupakan **id** milik data item yang ingin diambil. Ketik seperti contoh berikut

```
2  class CaseRead {
3      private $data = [];
4
5      public function get($id) {
6
7      }
8  }
```

4. Buat sebuah validasi menggunakan **if** di dalam fungsi **public**. Validasi data digunakan untuk mengecek apakah **\$id** yang dimasukkan dimiliki oleh data yang tersimpan dalam sistem.

```
5      public function get($id) {
6          if (!isset($this->data[$id])) {
7
8          }
9      }
```

5. Setelah membuat pemilihan, buatlah kembalian dengan **return** untuk memberikan respons jika validasi data gagal. Fungsi ini akan mengembalikan status **HTTP 404 (Not Found)** dengan pesan kesalahan.

```

5      public function get($id) {
6          if (!isset($this->data[$id])) {
7              return [
8                  'status' => 404,
9                  'response' => ['message' => 'Item Not Found']
10             ];
11         }
12     }

```

6. Jika validasi sukses maka diperlukan kode program untuk menampilkan data yang telah diambil sebelumnya, maka buat kembalian dengan **return** untuk validasi data sukses. Fungsi ini akan mengembalikan status **HTTP 200 (OK)** dengan pesan bahwa item berhasil ditampilkan kepada pengguna.

```

13         return [
14             'status' => 200,
15             'response' => ['data' => $this->data[$id]]
16         ];
17     }

```