

MODUL 12 RESTful API

12.1. Deskripsi Singkat

Aplikasi berbasis web telah mengalami perkembangan beberapa kali, dari yang pertama dikenal dengan Web 1.0, kemudian berkembang menjadi yang dikenal dengan Web 2.0 dalam era media sosial, lalu kemudian, sekarang, memasuki era Web 3.0 dengan kecerdasan artifisial (AI) dan blockchain. Yang menjadi salah satu kunci dalam perkembangan teknologi web ini adalah dimungkinkannya antar halaman web atau program berbasis web untuk saling berinteraksi secara langsung dan dinamis menggunakan suatu teknologi yang awalnya dikenal sebagai web services. Dalam modul ini, mahasiswa diperkenalkan dengan teknologi REST (Representational State Transfer) sebagai konsep dalam penerapan teknologi web services.

12.2. Tujuan Praktikum

Setelah menyelesaikan praktikum pada modul ini, mahasiswa diharapkan dapat mengenal bagaimana cara kerja serta dapat mengakses dan membuat web service berbasis RESTful API.

12.3. Material Praktikum

Praktikum menggunakan editor teks seperti Notepad atau Notepad++ atau aplikasi editor teks lainnya yang tersedia untuk menulis HTML dan program PHP. Pada praktikum ini juga menggunakan web server Apache dan program PHP. Selain itu, praktikum ini memerlukan sebuah tools (software alat bantu) untuk melakukan request ke web API dan melihat respon yang diterima.

12.4. Kegiatan Praktikum

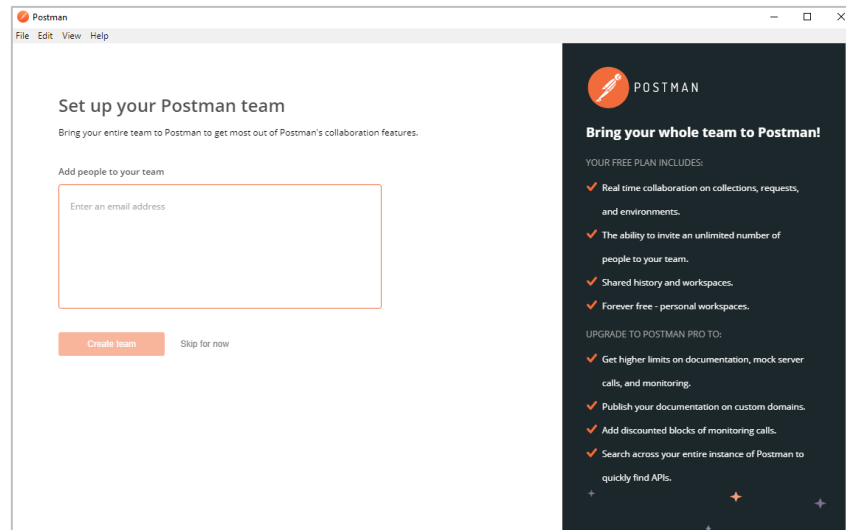
12.4.1. Instalasi Postman

Postman merupakan sebuah tools (software alat bantu) untuk melakukan request ke web API dan melihat respon yang diterima. Kelebihan dari Postman ini adalah kelengkapan fitur yang dimiliki untuk mencoba full method seperti GET, POST, PUT, DELETE dan yang lain-lain. Untuk mulai mengunduh aplikasi Postman, silahkan buka alamat berikut ini:

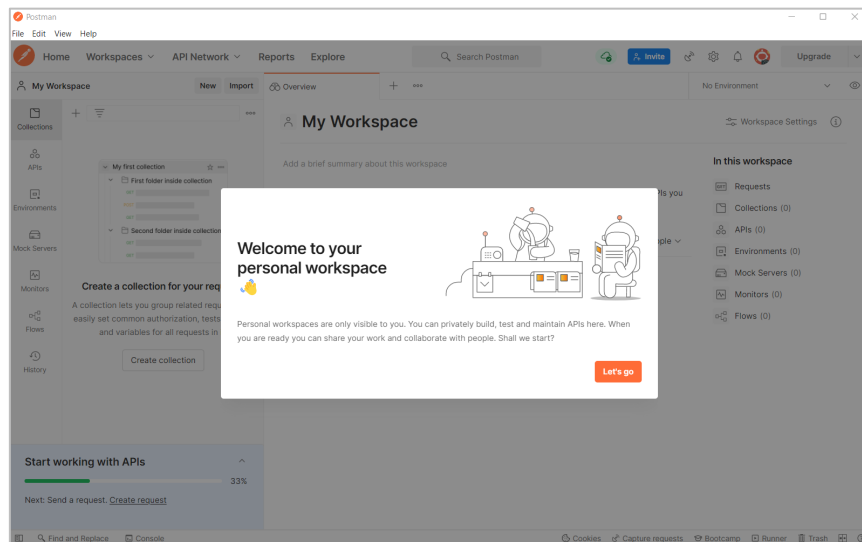
<https://www.postman.com/downloads/>

lalu pilih versi OS yang digunakan apakah 32bit atau 64bit. Setelah selesai mengunduh, jalankan setup instalasi aplikasi Postman. Setup akan menjalankan proses instalasi aplikasi Postman secara otomatis.

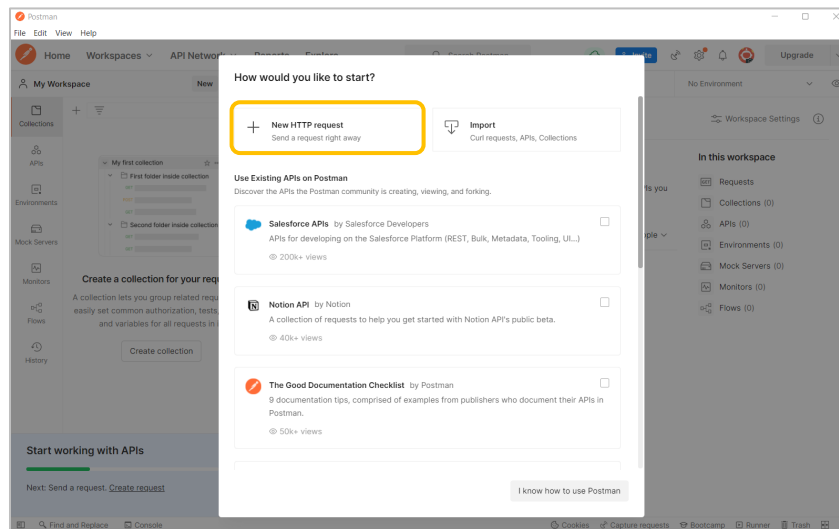
Setelah proses instalasi selesai, anda akan diarahkan untuk membuat akun Postman dan sign in menggunakan akun tersebut. Anda juga dapat melakukan sign in menggunakan akun Google. Postman memberikan kemudahan bagi anda untuk bekerja bersama secara tim, dimana anda dapat menambah rekan anda kedalam bagian workspace anda. Anda hanya perlu meng-invite rekan tim anda melalui e-mail.



Untuk praktikum ini, tahapan ini sementara dapat dilewati terlebih dahulu. Jika proses instalasi berhasil, maka Anda dapat melihat tampilan workspace pada postman.



Klik "Let's Go", maka Anda akan melihat tampilan berikut ini.



Pilih "New HTTP Request" seperti yang ditunjukkan pada gambar di atas. Lalu, ikuti Quick Lesson yang diberikan oleh Postman supaya Anda dapat mengenali tampilan workspace yang ada.

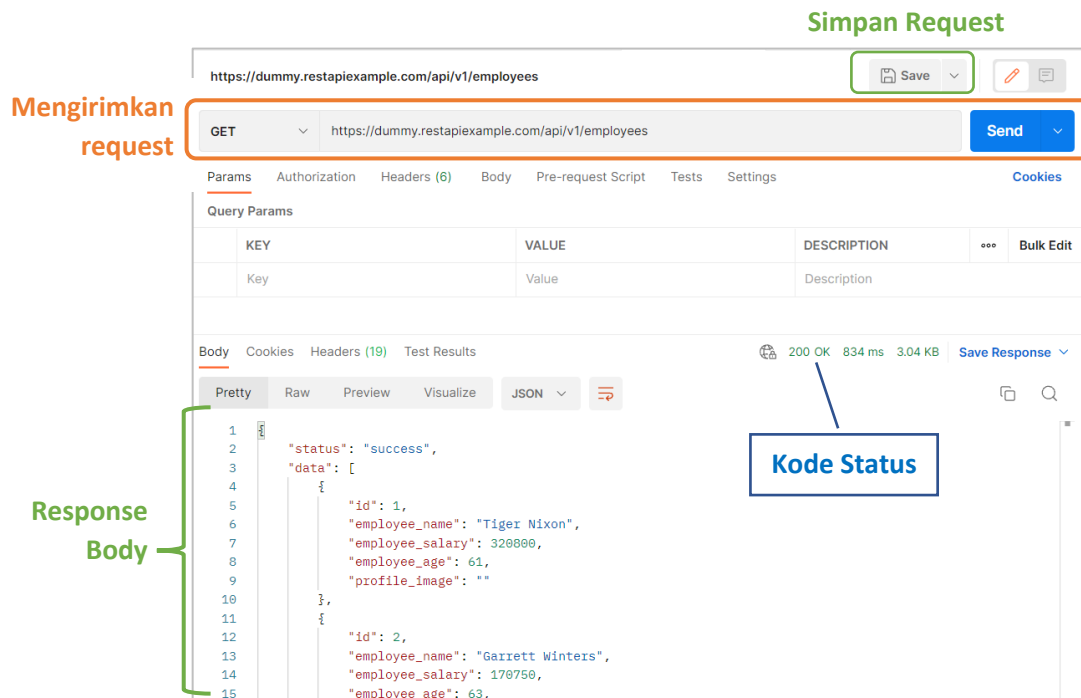
12.4.2. Mengakses API Menggunakan Postman

Agar lebih mengenali Postman, selanjutnya Anda akan mencoba menggunakan Postman untuk mengakses beberapa open API.

- Berikut ini adalah web api yang akan Anda gunakan untuk belajar: <http://dummy.restapiexample.com/>. Kunjungi tautan tersebut. disitu terdapat request URL yang bisa digunakan untuk mengambil data (GET), menambahkan data (POST), mengubah data (PUT) dan juga menghapus data (DELETE).

Welcome Dummy api example						
We're selling this sub domain including main domain(restapiexample.com), We'll provide all codes, tutorials, social pages with domain, Interested parties can Contact US .						
This page will list all of the rest services. These are fake online REST APIs for testing and prototyping sample applications that use rest calls to display listings and crud features. This rest api tutorials, faking a server, and sharing code examples can all be used.						
There are following public apis						
#	Route	Method	Type	Full route	Description	Details
1	/employee	GET	JSON	https://dummy.restapiexample.com/api/v1/employees	Get all employee data	Details
2	/employee/{id}	GET	JSON	https://dummy.restapiexample.com/api/v1/employee/1	Get a single employee data	Details
3	/create	POST	JSON	https://dummy.restapiexample.com/api/v1/create	Create new record in database	Details
4	/update/{id}	PUT	JSON	https://dummy.restapiexample.com/api/v1/update/21	Update an employee record	Details
5	/delete/{id}	DELETE	JSON	https://dummy.restapiexample.com/api/v1/delete/2	Delete an employee record	Details

- Cobalah untuk melakukan request ke service-service tersebut. Untuk membuat request baru, Anda dapat memilih menu **File > New > HTTP Request**. Masukkan URL untuk mendapatkan seluruh data employee pada postman, pastikan method yang dipilih pada postman sudah sesuai, yaitu **GET**. Klik tombol **SEND**. Jika request berhasil maka Anda akan melihat response berupa data JSON.



3. Klik tombol **Save** untuk menyimpan request di atas. Simpan dengan nama **Get All Employee** dalam collection **Praktikum 12**.

SAVE REQUEST

Request name

Get All Employee

Add description

Save to Select a collection/folder

Search for collection or folder

Praktikum12

New Collection

Cancel Save

4. Cobalah untuk mengakses service yang kedua dan kelima, yaitu mendapatkan dan menghapus satu data employee berdasarkan id employee. Caranya sama dengan tahapan nomor dan 3. Jangan lupa sesuaikan method menjadi DELETE untuk mengakses service kelima. Simpan request dengan nama **Get Employee** dan **Delete Employee**.

ID employee merupakan angka yang terdapat pada akhir request url. Coba modifikasi angka pada request url lalu kirimkan ulang requestnya. Apakah terdapat perbedaan pada response yang Anda terima?

5. Untuk menginsert data employee baru atau melakukan updating data employee, Anda dapat menggunakan method POST dan PUT. Sebagaimana insert atau update data ke database pada umumnya, terdapat data atau parameter yang harus Anda kirimkan bersamaan dengan pengiriman request. Jika Anda baca pada kolom **Details**, data atau parameter yang perlu dikirimkan saat menggunakan method POST dan PUT adalah **name (string)**, **salary (number)**, dan **age (number)**.

Menggunakan Postman, pengiriman data saat melakukan request tergantung pada method yang digunakan. Bila menggunakan method GET, data dapat diinput pada bagian **Params**. Bila menggunakan method POST, data dikirimkan melalui **body>form data**. Sedangkan saat menggunakan method PUT atau DELETE, **data dikirimkan melalui body > x-www-form-urlencoded**. Data yang akan dikirimkan diinput dalam bentuk pasangan **key-value** seperti yang dicontohkan pada gambar berikut.

The screenshot shows the Postman interface for a POST request. The URL is `https://dummy.restapixample.com/api/v1/create`. The 'Body' tab is selected, and 'form-data' is chosen as the body type. A table lists the form data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	employee1	
<input checked="" type="checkbox"/> salary	1000000	
<input checked="" type="checkbox"/> age	23	

Below the table, the 'Body' tab shows the JSON response in 'Pretty' format:

```
1 {
2   "status": "success",
3   "data": {
4     "name": "employee1",
5     "salary": "1000000",
6     "age": "23",
7     "id": 4845
8   },
9   "message": "Successfully! Record has been added."
10 }
```

Annotations in the image:

- An orange bracket on the left points to the form data table with the text: **Data yang akan Dikirimkan (Method POST)**.
- Another orange bracket on the left points to the JSON response body with the text: **Response Body**.

Buat request baru untuk mengakses service ketiga dan keempat. Simpan request dengan nama **Create Employee** dan **Update Employee**, lalu cobalah untuk menginsert dan mengupdate beberapa data employee.

12.4.3. Membuat REST API dengan Method GET

Pada bagian ini, Anda akan membuat API yang dapat direquest oleh client yang lain. Kita akan menggunakan tabel meetings yang telah dibuat pada praktikum 9. API pertama yang akan Anda buat adalah API untuk menampilkan seluruh data dari tabel meetings.

1. Hal pertama yang harus Anda siapkan adalah script PHP untuk membangun koneksi ke database dan query seluruh data yang terdapat pada tabel meetings. Buka teks editor dan masukkan markup HTML dan kode PHP berikut ini. Simpan dengan nama **php12A.php**.

```
<?php
$db_hostname = "localhost"; // Write your own db server here
$db_database = "<db_name>"; // Write your own db name here
$db_username = "<user>"; // Write your own username here
$db_password = "<password>"; // Write your own password here
                                // For the best practice, don't
                                // use your "real" password when
                                // submitting your work
$db_charset = "utf8mb4"; // Optional
$dsn =
"mysql:host=$db_hostname;dbname=$db_database;charset=$db_charset";
$opt = array(
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES => false
);
try {
    $pdo = new PDO($dsn,$db_username,$db_password,$opt);
    $sql = "select * from meetings";
    $stmt = $pdo->query($sql)->fetchAll();

    if ($stmt) { //if query result is not empty
        foreach($stmt as $row) {
            echo "Slot: ",$row["slot"]." ";
            echo "Name: ",$row["name"]." ";
            echo "Email: ",$row["email"]." ";
        }
    }

    $pdo = NULL;
}
catch (PDOException $e) {
    exit("PDO Error: ".$e->getMessage()."<br>");
}
?>
```

Jika Anda perhatikan, script di atas adalah script yang mirip dengan yang telah kita gunakan pada praktikum 9, 10, dan 11, hanya saja tanpa tag-tag HTML.

Tuliskan *username* dan *password* MySQL yang anda gunakan, menggantikan **<user>** dan **<password>** dalam kode program di atas. Perhatikan bahwa script di atas menggunakan fungsi `fetchAll()` sehingga hasil query dikembalikan dalam bentuk array. Akses `php12A.php` melalui browser dan pastikan output yang ditampilkan sudah benar.

2. Berikutnya, kita akan mencoba mengakses `php12A.php` melalui postman. Buat request baru, simpan dengan nama **php12A_API**. Copy URL untuk mengakses `php12A.php` pada field request URL yang terdapat pada postman dan pilih method GET, lalu send request. Pastikan di bagian response body terdapat output yang tadi Anda lihat di browser.
3. Sampai di sini, API untuk menampilkan seluruh data dari tabel `meetings` sudah berjalan. Namun, pada umumnya data pada response body dikirimkan dalam bentuk JSON dan memuat kode status. Untuk itu, modifikasi script pada `php12A.php`, tepatnya di bagian setelah statement untuk mengeksekusi sql dan sebelum statement `$pdo = NULL;`.

```
if ($stmt) { //if query result is not empty
    foreach($stmt as $row) {
        $item[] = array(
            'slot'=> $row["slot"],
            'name'=> $row["name"],
            'email'=>$row["email"]
        );
    }
}

$response = array(
    'status'=>'200 OK',
    'data' => $item
);

echo json_encode($response);
```

4. Kirim kembali request pada postman. Seharusnya pada bagian response body, output yang ditampilkan sudah dalam format JSON.
5. Untuk menampilkan data tertentu dari tabel `meetings` berdasarkan `slot`, Anda dapat melakukan sedikit modifikasi pada `php12A.php` dan `php12A_API`. Simpan ulang `php12A.php` dan `php12A_API` dengan nama **php12A_byslot.php** dan **php12A_API_byslot**.

Pada `php12A_byslot.php`, Anda harus menangkap `slot` data yang ingin ditampilkan menggunakan super global variabel `$_GET`. Slot tersebut kemudian digunakan pada query dengan menggunakan sintaks `WHERE`.

Saat mengirimkan request **php12A_API_byslot**, slot dikirimkan dengan cara diinput pada bagian **Params**.

6. Tambahkan kondisi untuk mengeluarkan respon berikut jika slot id yang ingin ditampilkan tidak ditemukan dalam basis data.

```
$response = array(  
    $response = array(  
        'status'=>'200 OK',  
        'data' => [],  
        'message' => 'data not found'  
    );  
);
```

12.4.4. Membuat REST API dengan Method POST

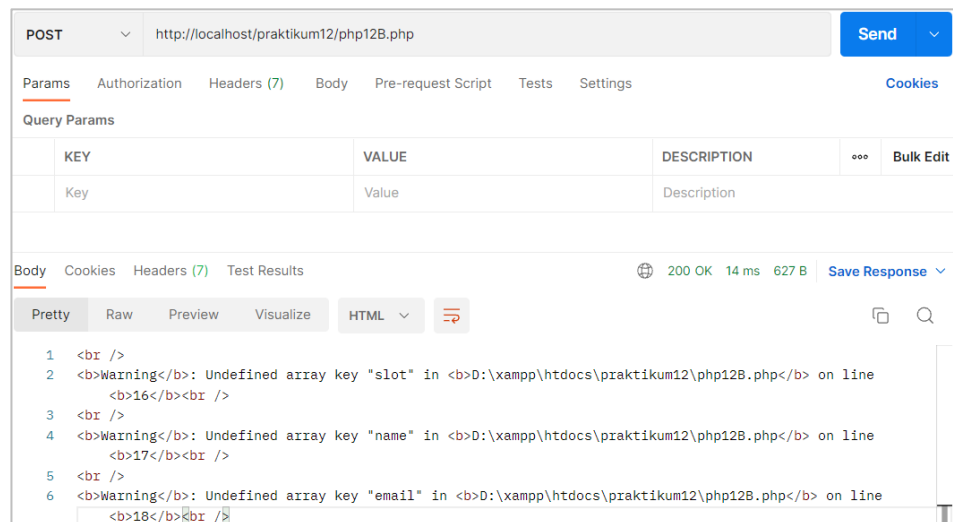
Selanjutnya, Anda akan membuat API dengan menggunakan method POST untuk menambahkan data baru ke tabel meetings.

1. Sama halnya dengan kegiatan 12.4.3., hal pertama yang harus Anda siapkan adalah script PHP untuk membangun koneksi ke database dan query untuk menginsert data baru ke tabel meetings. Simpan ulang file **php12A.php** dengan nama **php12B.php**.
2. Modifikasi sintaks query yang terdapat pada block statement `try` seperti yang telah Anda buat pada praktikum 9.

```
$pdo = new PDO($dsn,$db_username,$db_password,$opt);  
  
$slot = $_POST['slot'];  
$name = $_POST['name'];  
$email = $_POST['email'];  
  
$sql = "INSERT INTO `meetings` (`slot`, `name`, `email`) VALUES  
('$slot', ' $name', '$email')";  
  
$stmt = $pdo->query($sql);  
  
$pdo = NULL;
```

3. Buat request baru pada postman dengan nama **php12B_API**, lalu coba kirimkan request untuk mengakses `php12B.php`. Sesuaikan request URL dengan lokasi penyimpanan file `php12B.php` Anda. Jangan lupa, method yang digunakan adalah POST.

4. Pada response body, muncul pesan error Undefined array key.



Hal ini terjadi karena pada `php12B.php` kita memerlukan data `slot`, `name`, dan `email`. Pada praktikum 9, data tersebut Anda dapatkan dari formulir yang diisi oleh pengguna. Menggunakan postman, data tersebut kita kirimkan melalui **body > form data** seperti yang Anda kerjakan pada kegiatan 12.4.2.

5. Isi data-data yang akan ditambahkan ke tabel `meetings` melalui **body > form data**, lalu kirim ulang request `php12B_API`. Buka MySQL atau phpMyAdmin, pastikan data sudah berhasil ditambahkan ke tabel `meetings`.
6. Agar client mengetahui apakah data berhasil ditambahkan atau gagal, sebelum statement `$pdo = NULL;`, tambahkan pesan yang akan dikirimkan sebagai response body. Ingat, data yang dikirimkan harus dalam format JSON.

```
if ($stmt){
    $msg = "Data berhasil ditambahkan.";
} else{
    $msg = "Gagal.";
}

$response = array(
    'status'=>'201 Created',
    'message' => $msg
);

echo json_encode($response);
```

7. Kirim ulang request `php12B_API` pada postman. Seharusnya sekarang terdapat pesan berhasil atau gagal pada bagian response body. Buka MySQL atau

phpMyAdmin untuk memastikan bahwa data sudah berhasil ditambahkan ke tabel meetings.

12.4.5. Membuat REST API dengan Method PUT

Berikutnya Anda akan membuat API dengan menggunakan method PUT untuk mengupdate `nama` dan `email` pada tabel meetings berdasarkan `slot`.

1. Sama seperti sebelumnya, kita siapakan script untuk membangun koneksi ke database dan melakukan query. Simpan ulang file **php12B.php** dengan nama **php12C.php**.
2. Modifikasi sintaks query yang disimpan pada variabel `$sql` menjadi query untuk mengupdate data ke tabel meetings. Lihat kembali hasil kerja Anda pada praktikum 9. Modifikasi pula pesan yang akan dikirimkan sebagai response body.
3. Perlu Anda ketahui bahwa tidak seperti `$_GET` dan `$_POST`, PHP tidak menyediakan super global variabel untuk PUT. Sehingga untuk mendapatkan data `slot`, `nama`, dan `email`, tidak dapat dilakukan dengan cara ini.

```
$slot = $_POST['slot'];  
$name = $_POST['name'];  
$email = $_POST['email'];
```

Untuk mendapatkan data yang dikirimkan dengan menggunakan method PUT, Anda dapat menggunakan fungsi `parse_str()` seperti berikut ini.

```
parse_str(file_get_contents("php://input"),$_PUT);  
  
$slot = $_PUT['slot'];  
$name = $_PUT['name'];  
$email = $_PUT['email'];
```

`$_PUT` disini hanyalah sebuah identifier yang dapat Anda ganti dengan nama lain.

4. Buat request baru pada postman dengan nama **php12C_API**, lalu coba kirimkan request untuk mengakses `php12C.php`. Sesuaikan request URL dengan lokasi penyimpanan file `php12C.php` Anda. Jangan lupa, method yang digunakan adalah PUT. Inputkan `slot` yang ingin diupdate, beserta `name` dan `email` yang baru pada **body > x-www-form-urlencoded**. Kirim request `php12C_API`. Buka MySQL atau phpMyAdmin, pastikan data sudah berhasil diperbaharui.

12.4.6. Membuat REST API dengan Method DELETE

Terakhir, Anda akan membuat API dengan menggunakan method DELETE berdasarkan `slot`.

1. Simpan ulang file **php12C.php** dengan nama **php12D.php**. Modifikasi sintaks query yang disimpan pada variabel `$sql` menjadi query untuk menghapus data ke tabel `meetings`. Lihat kembali hasil kerja Anda pada praktikum 9. Modifikasi juga isi pesan yang akan dikirimkan sebagai response body.
2. Sama seperti PUT, PHP tidak menyediakan super global variabel untuk DELETE. Sehingga untuk mendapatkan data `slot` data yang ingin dihapus, Anda masih membutuhkan fungsi `parse_str()`.

```
parse_str(file_get_contents("php://input"),$_DELETE);  
$slot = $_DELETE['slot'];
```

3. Buat request baru pada postman dengan nama **php12D_API**, lalu coba kirimkan request untuk mengakses `php12D.php`. Sesuaikan request URL dengan lokasi penyimpanan file `php12D.php` Anda. Method yang digunakan ganti menjadi DELETE. Inputkan `slot` data yang ingin dihapus pada **body > x-www-form-urlencoded**. Kirim request `php12D_API` lalu Buka MySQL atau phpMyAdmin untuk memastikan data dengan `slot` tersebut sudah berhasil dihapus.

12.4.7. Mengakses REST API menggunakan PHP

Selain menggunakan Postman, kita juga dapat mengakses API melalui PHP menggunakan fungsi `file_get_contents()` atau library `cURL`. Sebagai contoh, anda dapat mengakses API GET yang telah anda kerjakan pada `php12A.php` dengan cara berikut ini.

```
$data = file_get_contents("http://localhost/praktikum12/php12A.php");  
var_dump($data);
```

Simpan potongan program di atas dengan nama `restClient.php`, lalu jalankan pada browser Anda. Jika berhasil, anda dapat melihat output yang ditampilkan berupa json string. Agar json string tersebut dapat diproses lebih lanjut, maka kita harus menjalankan fungsi `decode` seperti berikut ini.

```
$parse_data = json_decode($data,true);  
echo "<br/><br/>";  
var_dump($parse_data);
```

Selanjutnya anda dapat memproses dan menampilkan data yang didapatkan, sesuai dengan kebutuhan masing-masing. Paramater `true` pada fungsi `json_decode`

berfungsi untuk mengubah json string menjadi array. Jika anda menginginkan hasil decode berupa object, maka parameter `true` dapat anda hapus. Cobalah untuk menghapus parameter tersebut. Lalu jalankan kembali **restClient.php** pada browser untuk melihat perbedaan output yang dihasilkan.

Jika menggunakan library `cURL`, Anda dapat mengakses API GET yang telah anda kerjakan pada `php12A.php` dengan cara berikut ini.

```
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, "http://localhost/praktikum12/php12A.php");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
$output = curl_exec($ch);
curl_close($ch);
echo $output;

$data = json_decode($output,true);
echo "<br/><br/>";
var_dump($data);
```

Anda dapat mempelajari lebih lanjut mengenai fungsi `file_get_contents()` dan library `cURL` dokumentasi di bawah ini.

<https://www.php.net/manual/en/function.file-get-contents.php>

<https://www.php.net/manual/en/book.curl.php>

Cobalah untuk memodifikasi `restClient.php` sehingga dapat mengakses API lainnya yang telah anda buat pada `php12B.php`, `php12C.php`, `php12D.php`.

12.5. Responsi

Kerjakan sesuai dengan yang dijelaskan pada bagian Kegiatan Praktikum. Simpan tangkapan layar hasil pekerjaan Anda untuk keempat kegiatan praktikum dalam file docx. Simpan ulang file tersebut dalam format pdf, dan beri nama dengan format `<<nim>>_modul12.pdf`, contoh: `192191234_modul12.pdf`. Kumpulkan file tersebut sebagai responsi melalui Google Classroom.