

LAPORAN PRAKTIKUM PENGENALAN APLIKASI BERBASIS INTERNET

MENCIPTAKAN DAN MENGGUNAKAN RESTFUL API



Agus Pranata Marpaung

13323033

DIII TEKNOLOGI KOMPUTER

**INSTITUT TEKNOLOGI DEL
FAKULTAS VOKASI**

Judul Praktikum

Minggu/Sesi	:	IX/2
Kode Mata Kuliah	:	1331204
Nama Mata Kuliah	:	Pengenalan Aplikasi Berbasis Internet
Setoran	:	<i>Softcopy</i>
Batas Waktu Setoran	:	<i>21 Maret 2024 jam 15:00</i>
Tujuan	:	<i>1. Students Know the hands on of Rest API</i>

Menciptakan dan menggunakan Rest API

Praktikum ini membantu untuk membuat operasi CRUD REST api tanpa menggunakan Framework REST. Anda dapat mendefinisikan struktur apa saja dan disesuaikan sesuai kebutuhan Anda, Anda dapat menetapkan aturan untuk framework REST API dan mudah mengaksesnya. Tidak wajib untuk memiliki framework kerja untuk membangun API REST Anda menggunakan php. Anda dapat membuat api RESTnya menggunakan kode PHP inti.

Ada beberapa framework REST menggunakan untuk PHP adalah sebagai berikut.

Laravel
Phalcon
Lumen
Slim
Silex

Mari kita mulai membuat contoh REST API, saya akan membuat folder dengan nama '**restful-api**'. Folder ini akan menyimpan semua file yang diperlukan untuk praktikum ini. mahasiswa akan membuat Rest API untuk modul karyawan yang akan memiliki HTTP Get untuk permintaan untuk mengambil catatan dari mysql, Post untuk tambahkan catatan ke MySQL, Put untuk perbarui catatan ke mysql dan Delete untuk hapus record data yang ditentukan dari mysql. Detail dari REST API adalah sebagai berikut:

Route	Method	Type	Posted JSON	Description
/employees	GET	JSON	-	Dapatkan semua data karyawan
/employees/{id}	GET	JSON	-	Get a single employee data
/employees	POST	JSON	<pre>{ "employee_name": "Moses", "employee_age": "19", "employee_salary" : "9000000" }</pre>	Masukkan catatan karyawan baru ke dalam basis data
/employees/13319020	PUT	JSON	<pre>{ "employee_name": "Moses", "employee_age": "27", "employee_salary" : "12000000" }</pre>	Perbarui catatan karyawan ke dalam basis data
/employees	DELETE	JSON	<pre>{" id" : 13319020 }</pre>	Hapus catatan karyawan tertentu dari database

Mahasiswa akan menggunakan file-file berikut untuk praktikum REST API ini:

index.php: File ini adalah file entri, File ini mencegah navigasi file folder. **connection.php:**

File ini akan digunakan untuk memiliki string koneksi mysql.

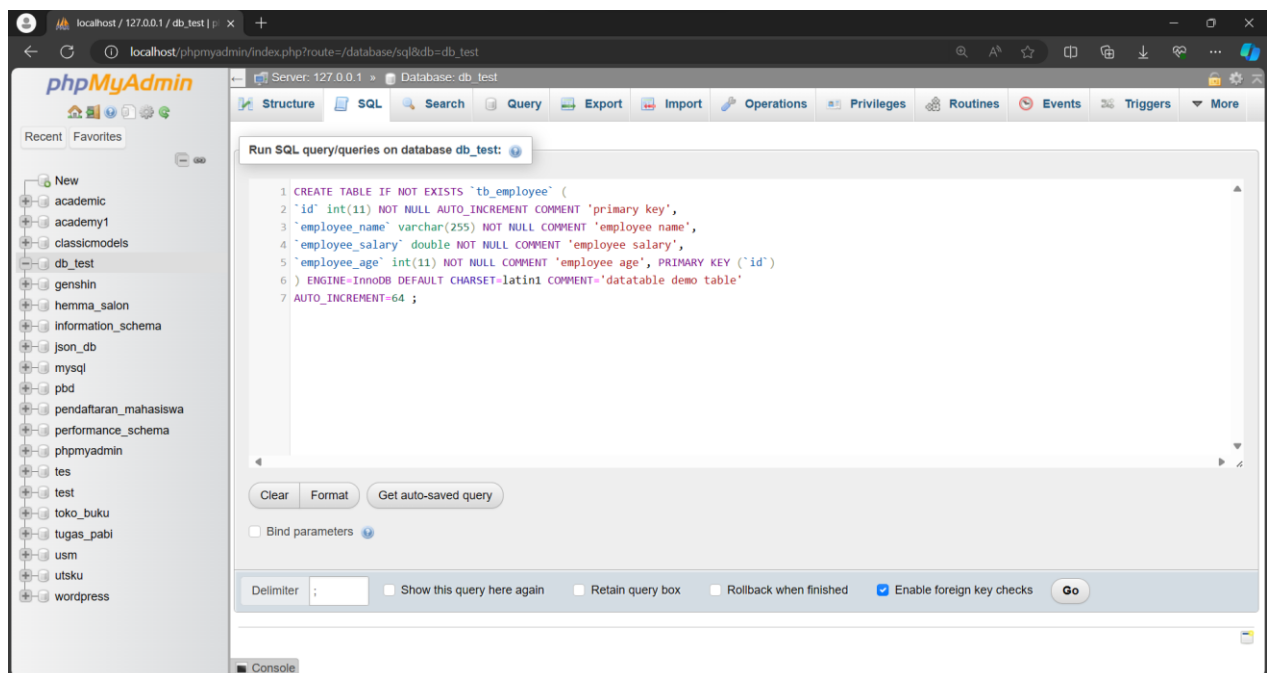
v1 / employee.php: File ini akan berisi semua titik akhir istirahat dan metode tindakan. **v1 /**

.htaccess: File ini digunakan untuk tujuan perutean/routing ke endpoint.

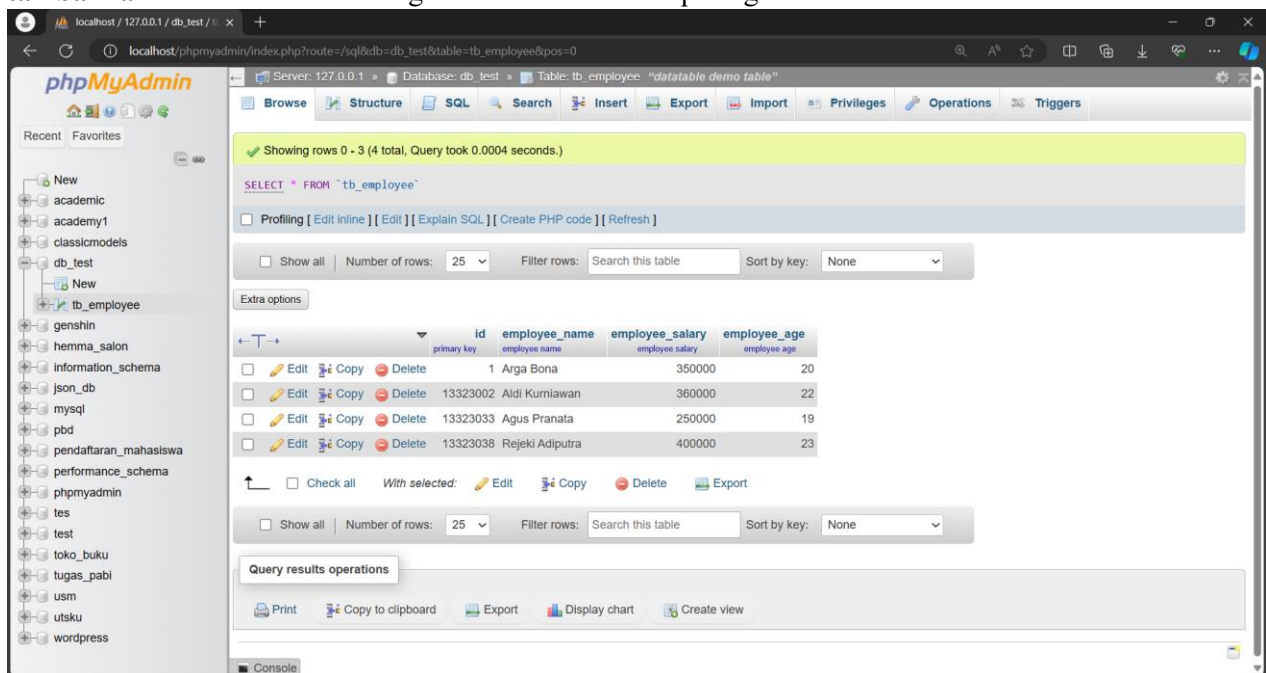
Skema Tabel Pegawai

Buat '**db_test**' database MySQL, jika Anda sudah memiliki database maka jalankan sql di bawah ke dalam window-query SQL database.

```
CREATE TABLE IF NOT EXISTS `tb_employee` (  
  `id` int(11) NOT NULL AUTO_INCREMENT COMMENT 'primary key',  
  `employee_name` varchar(255) NOT NULL COMMENT 'employee name',  
  `employee_salary` double NOT NULL COMMENT 'employee salary',  
  `employee_age` int(11) NOT NULL COMMENT 'employee age', PRIMARY  
  KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='datatable demo table'  
AUTO_INCREMENT=64 ;
```



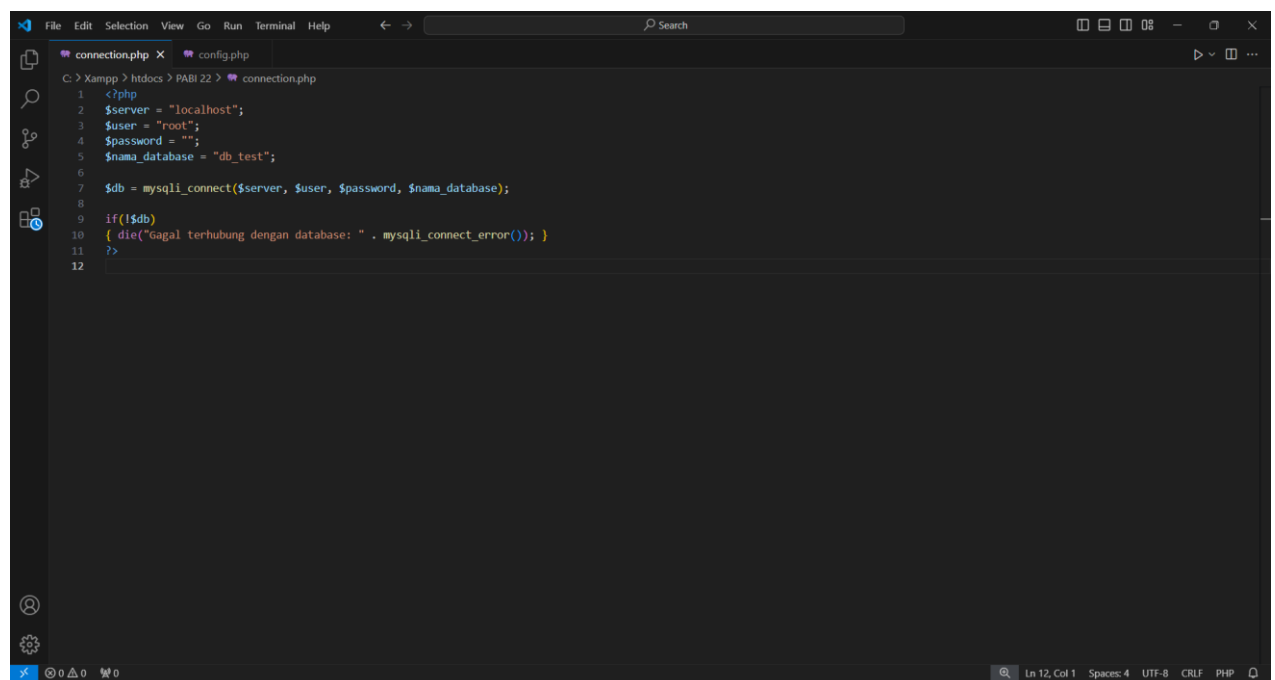
Selanjutnya isi setiap kolom pada tabel anda dengan dummy data/data percobaan dengan klik menu **tambahkan** dan isilah sesuai keinginan anda. Ini terlihat pada gambar berikut:

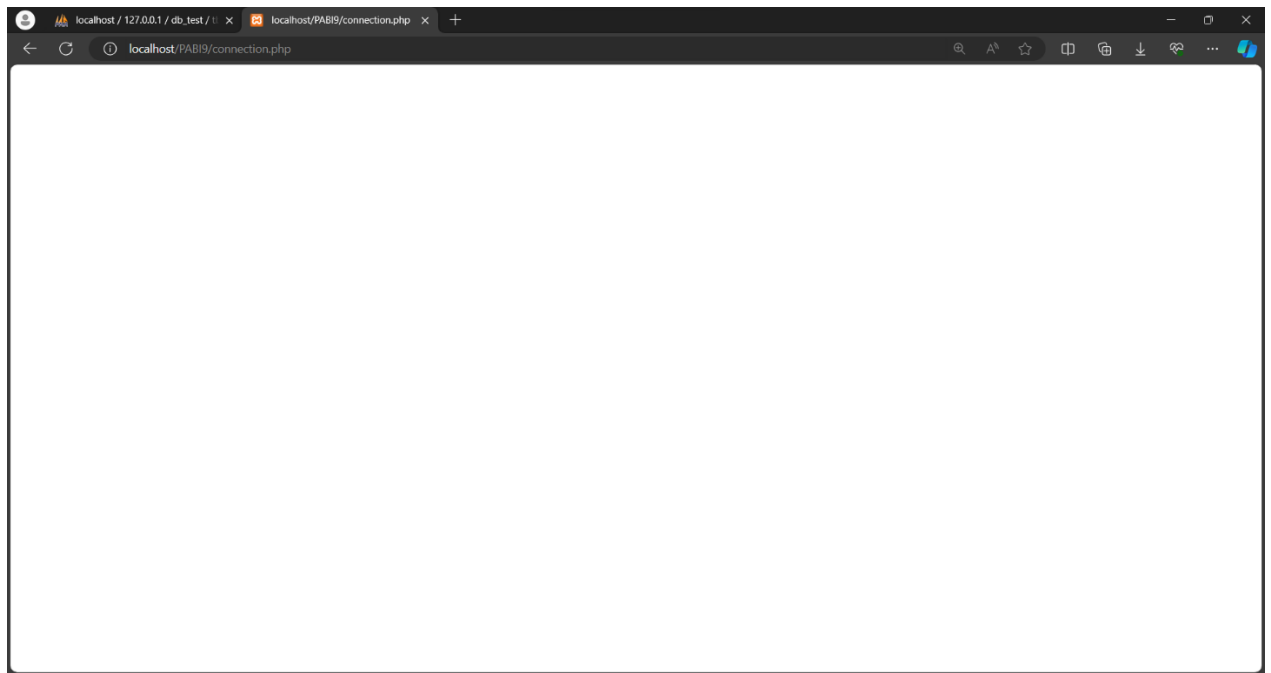


Jadi, saya sudah punya 4 data dummy yang terisi pada masing-masing kolom.

Koneksi Database MySQL Dengan PHP Menggunakan Mysqli

Buatlah file **connection.php** untuk menghubungkan MySQL dengan php menggunakan metode **mysqli_connect()**. Tambahkan kode di bawah ini ke file tersebut.





Catatan:

Anda perlu mengganti **nama basis data**, **nama pengguna db**, **kata sandi db** dan **nama host** sesuai konfigurasi database anda.

.Htaccess Rewriterule For Clean URLs

kita akan membuat file **.htaccess** v1 / folder dan menulis beberapa aturan untuk REST API dengan URL yang rapi. kita akan menambahkan aturan berikut.

```
RewriteEngine On # Turn on the rewriting engine
RewriteRule ^employees/?$ employees.php [NC,L]
RewriteRule ^employees/([0-9]+)/?$ employees.php?id=$1 [NC,L]
```

REST API Untuk Mengambil Semua Record Dari MySQL

kita akan membuat Permintaan HTTP Request dengan jenis GET untuk mengakses semua record/data karyawan dari MySQL. kita akan menggunakan query MySQL untuk mengambil data dari tabel karyawan dan mengirim array data JSON ke klien sebagai response object.

Sekarang kita akan membuat file **employee.php** didalam folder **v1** dan menambahkan file koneksi MySQL untuk mengakses data tabel database.

```
1 <?php
2 // Connect to database
3 include("../connection.php");
4 $connection = $db;
5 $request_method = $_SERVER["REQUEST_METHOD"];
6
```

Catatan:

Kita akan menulis kode program pada file **employee.php** secara bertahap untuk tiap fungsi. Kode program yang lengkap akan saya cantumkan pada akhir modul praktikum ini.

Saya juga menggunakan metode **\$_SERVER** untuk mengakses Informasi action, seperti rest request untuk menambah, mengedit, atau menghapus.

kita akan membuat metode **SWITCH-CASE** pada php untuk mengakses *method* yang tepat terhadap *rest request*.

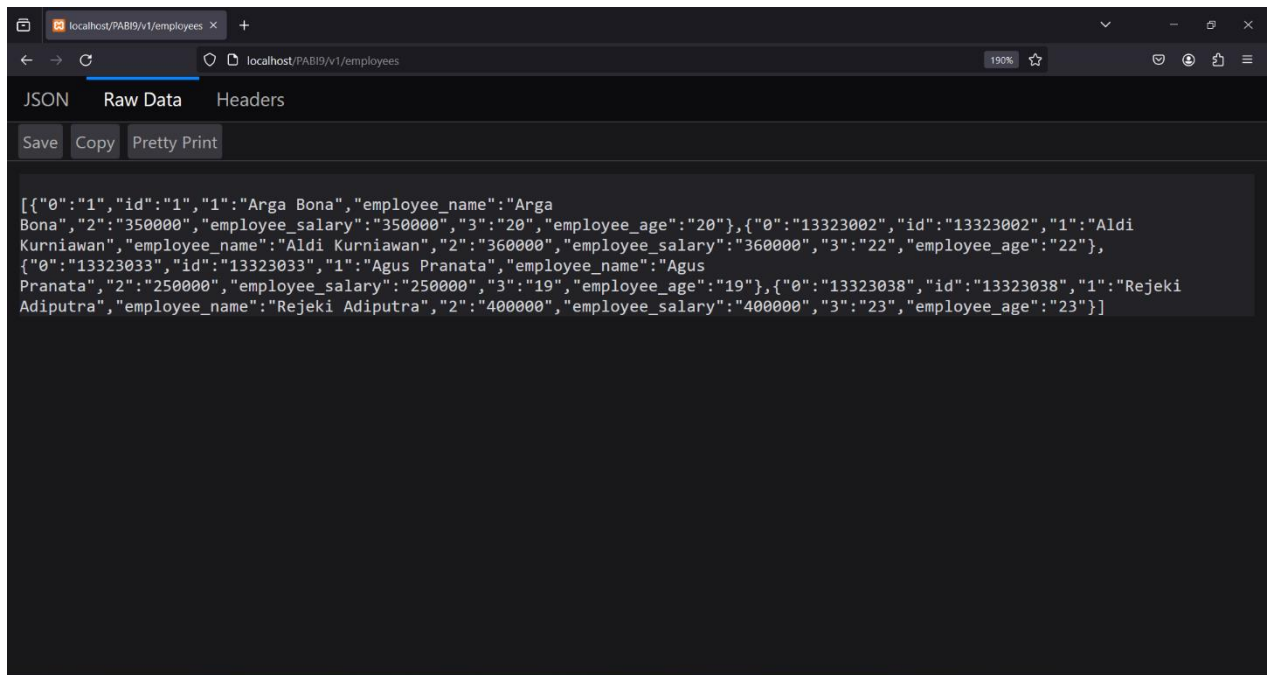
```
switch($request_method) {  
    case 'GET':  
        // Retrive Products  
        if (!empty($_GET["id"])) {  
            $id = intval($_GET["id"]);  
            get_single_employee($id);  
        } else {  
            get_all_employees();  
        }  
        break;  
}
```

kita telah membuat permintaan GET untuk mengambil semua data karyawan dari database mysql, untuk data karyawan tunggal kita melakukan *passing* id karyawan pada argument fungsi. kita akan membuat metode/function **get_employees()** seperti di bawah ini.

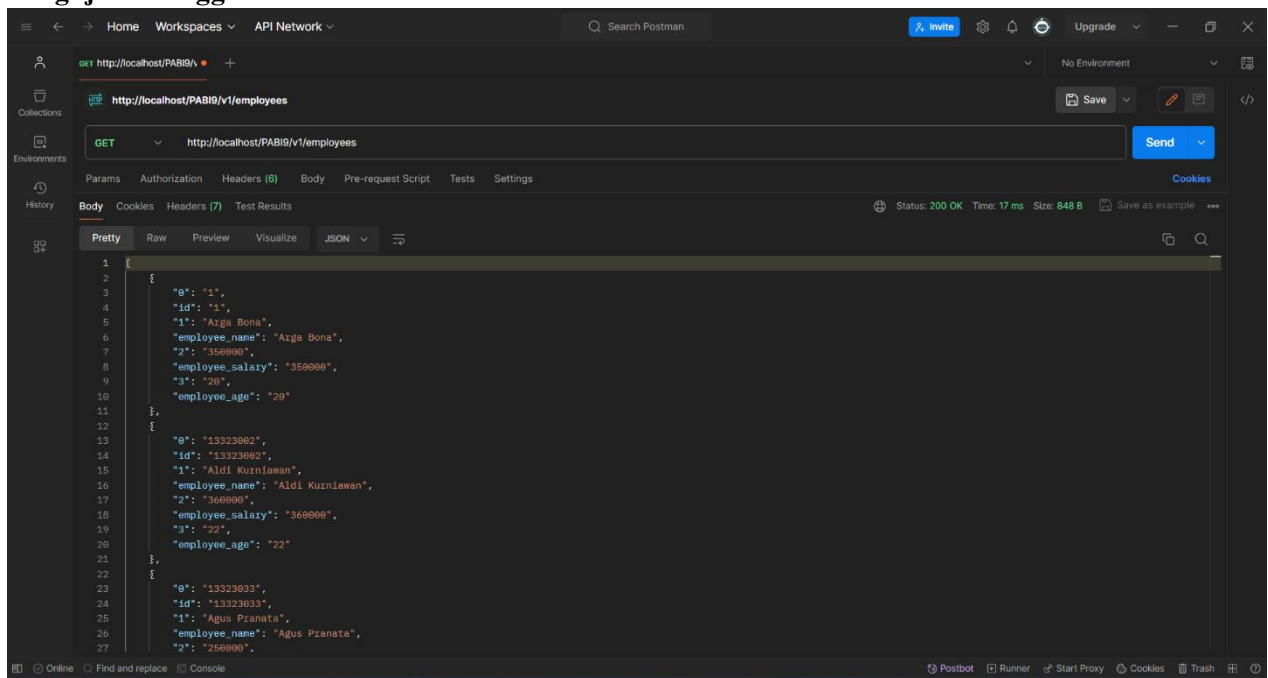
```
40  
41 function get_all_employees() {  
42     global $connection;  
43     $query = "SELECT * FROM tb_employee";  
44     $response = array();  
45     $result = mysqli_query($connection, $query);  
46     while($row = mysqli_fetch_array($result)) {  
47         $response[] = $row;  
48     }  
49     header('Content-Type: application/json');  
50     echo json_encode($response);  
51 }  
52 ?>  
53 |
```

metode **mysqli_query ()** mengambil data dari tabel karyawan MySQL dan disimpan sebagai variabel ke dalam variabel '**result**'. Metode **json_encode ()** mengubah array data menjadi string json.

Sekarang akses <http://localhost/PABI/Restful-API/v1/employees> . Ini adalah REST API url dan anda akan mendapatkan semua catatan karyawan dari tabel karyawan mysql.



Pengujian menggunakan tools Postman



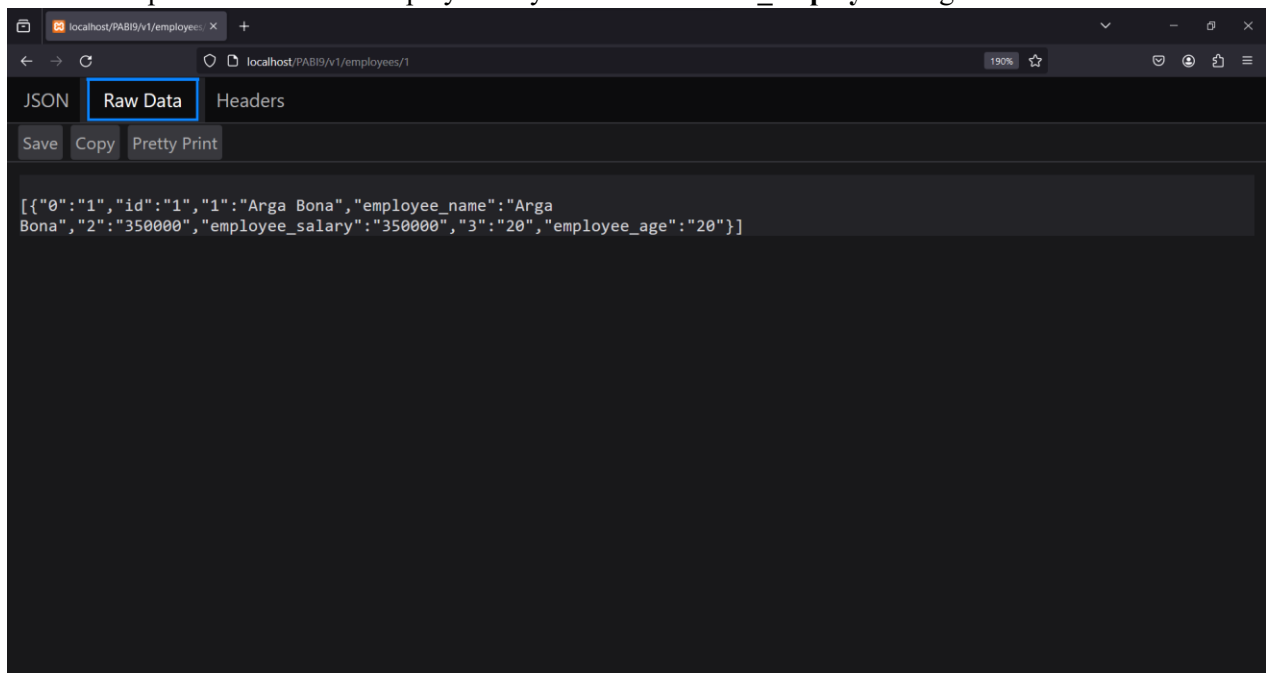
Jika kita menggunakan HTTP request GET untuk mengambil semua data dari database lewat **URI/ endpoint** rest server dan kita klik **SEND**, maka akan menghasilkan output berupa data yang sama. Artinya, REST-API dengan fungsi GET semua record sudah berhasil kita ciptakan dan kita uji menggunakan postman sebagai REST client.

Restful API Untuk Mendapatkan Single record/data tunggal Dari MySQL Menggunakan PHP

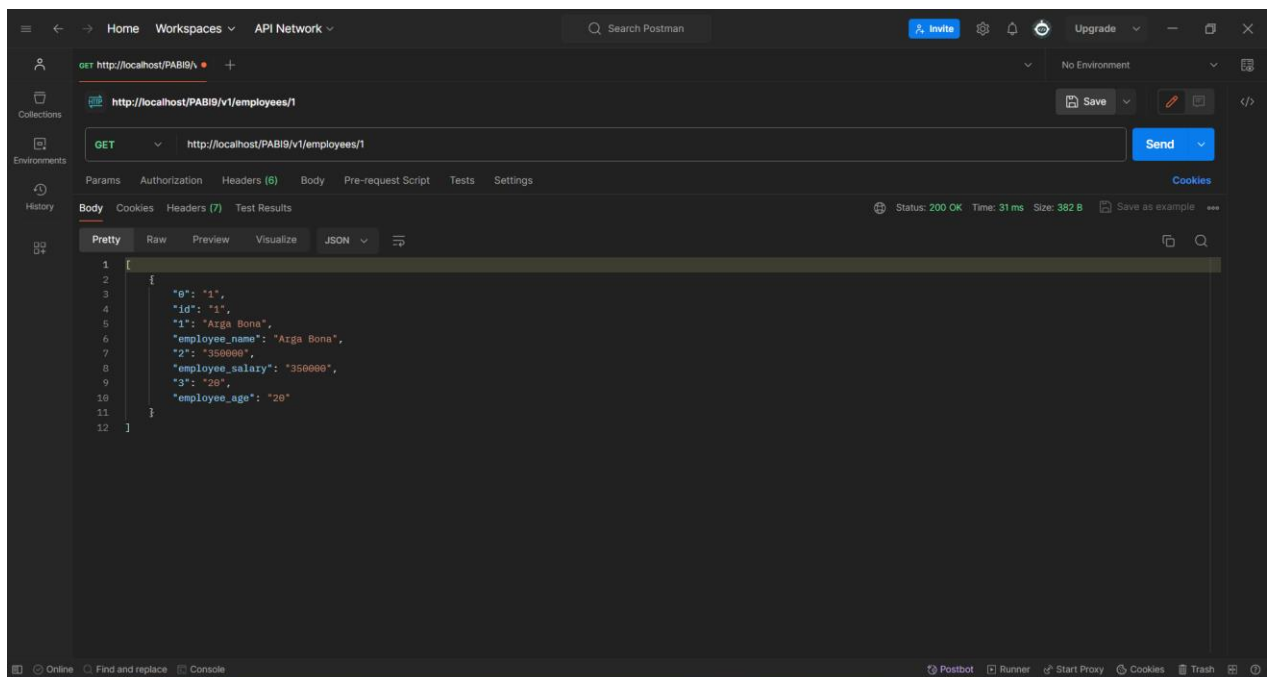
kita akan membuat rest request dengan jenis HTTP GET untuk mengakses single/satu record karyawan dari database MySQL menggunakan php. sama seperti pemanggilan REST API sebelumnya dengan melakukan passing/melewatkan employee id sebagai argumen/parameter fungsi.

```
function get_all_employees() {  
    global $connection;  
    $query = "SELECT * FROM tb_employee";  
    $response = array();  
    $result = mysqli_query($connection, $query);  
    while($row = mysqli_fetch_array($result)) {  
        $response[] = $row;  
    }  
    header('Content-Type: application/json');  
    echo json_encode($response);  
}
```

Sekarang akses <http://localhost/PAB/Restful-API/v1/employees/1> sebagai REST API endpoint dan anda akan mendapatkan satu record employee/karyawan dari tabel **tb_employee** dengan **id =1**.



Pengujian menggunakan tools Postman



Rest Api Untuk Membuat Record Baru Ke MySQL

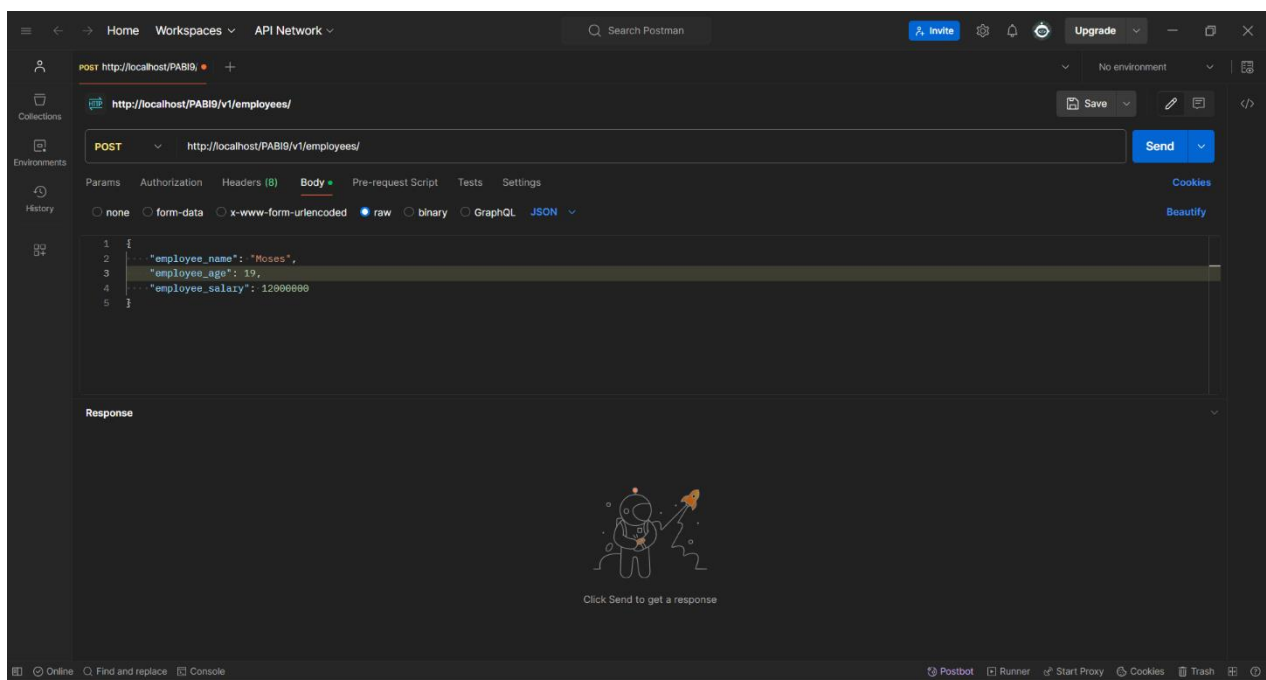
Kita akan membuat Rest Api baru untuk memasukkan Record karyawan baru ke dalam MySQL menggunakan php. Saya akan membuat REST Request dengan jenis POST karena kita akan memposting beberapa data JSON ke server php. kita akan menambahkan **CASE** baru ke dalam metode **SWITCH** seperti di bawah ini.

```
case 'POST':  
    // Insert Product  
    insert_employee();  
    break;
```

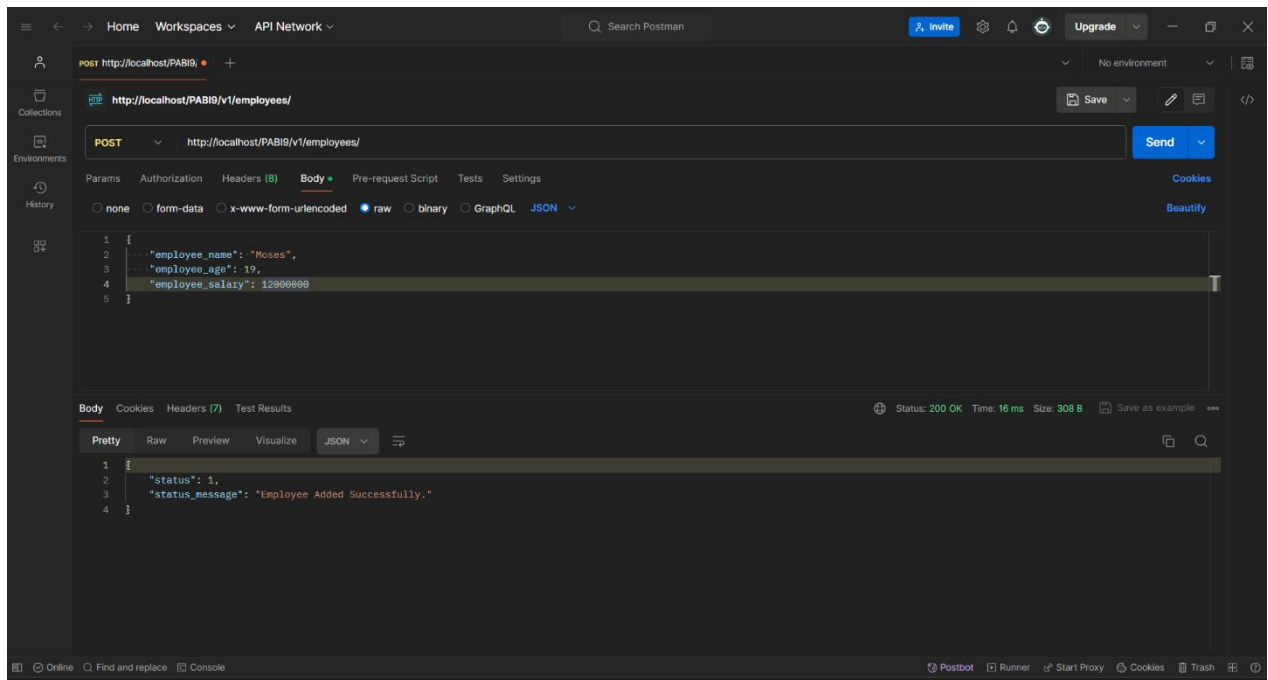
Now we will create `insert_employee()` method into `employees.php` file.

```
function insert_employee()
{
    global $connection;
    $data = json_decode(file_get_contents('php://input'), true);
    $employee_name = $data["employee_name"];
    $employee_salary = $data["employee_salary"];
    $employee_age = $data["employee_age"];
    $query = "INSERT INTO tb_employee SET employee_name='" . $employee_name . "', employee_salary='" . $employee_salary . "', employee_age='" . $employee_age . "'";
    if (mysqli_query($connection, $query)) {
        $response = array(
            'status' => 1,
            'status_message' => 'Employee Added Successfully.'
        );
    } else {
        $response = array(
            'status' => 0,
            'status_message' => 'Employee Addition Failed.'
        );
    }
    header('Content-Type: application/json');
    echo json_encode($response);
}
```

Pengujian menggunakan tools Postman

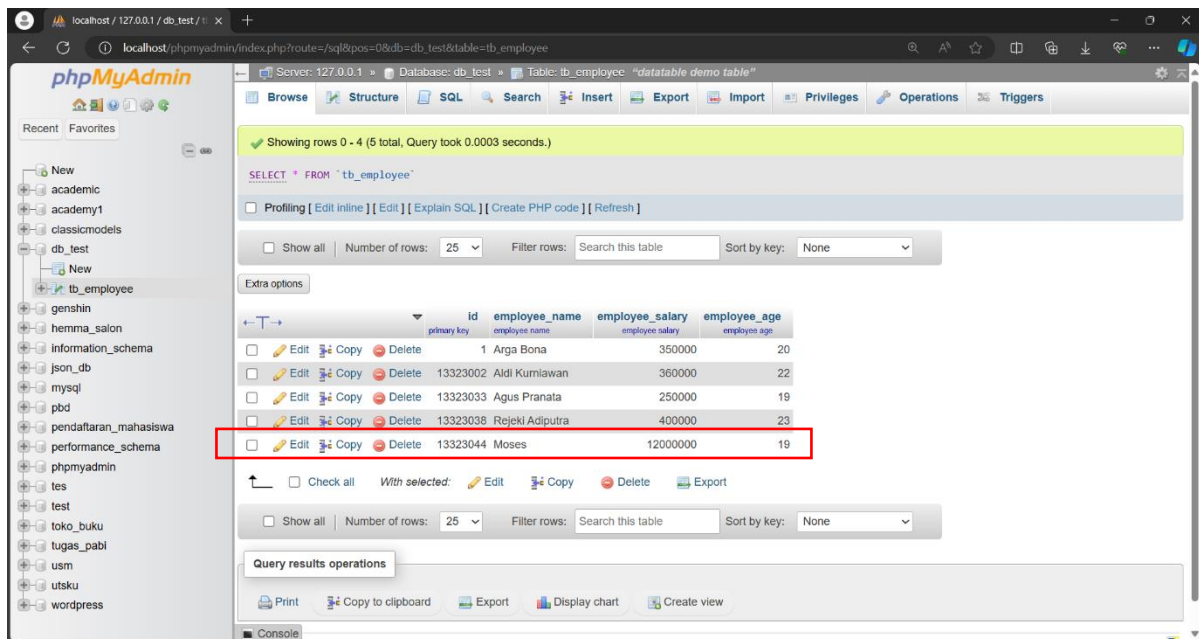


Jika anda kirimkan request POST ke endpoint rest-server yaitu request yang menambahkan sebuah data bar ke database dengan klik tombol **Send**, maka Response body akan menghasilkan output sebagai berikut:



Perubahan data pada database

Silahkan anda cek database, maka anda akan melihat data baru telah berhasil ditambahkan di database anda:



Rest Api Untuk Update Record Ke MySQL Menggunakan PHP

kita akan membuat jenis HTTP PUT Request Restful API baru untuk memperbarui data ke dalam basis data MySQL. Kita harus *passing* id karyawan sebagai parameter yang akan digunakan untuk memperbarui catatan ke tabel karyawan.

Kita akan menambahkan **CASE** baru ke dalam metode **SWITCH** seperti di bawah ini.

```
case 'PUT':  
    // Update Product  
    $id = intval($_GET["id"]);  
    update_employee($id);  
    break;
```

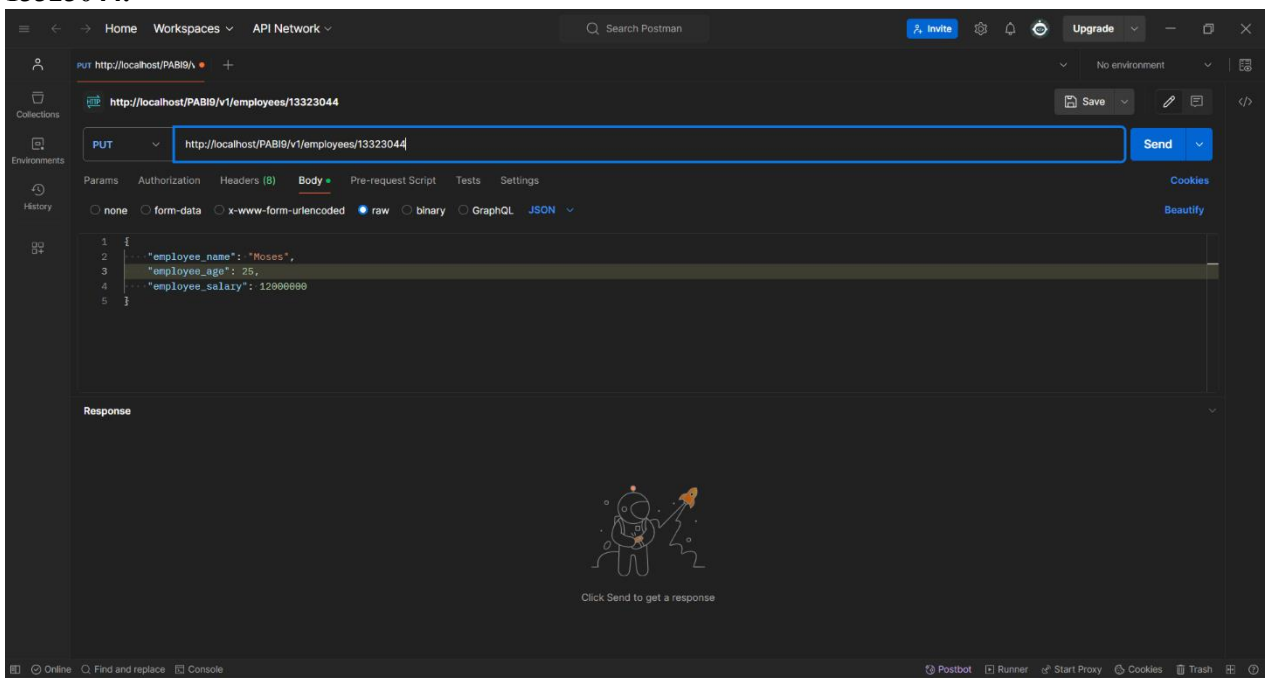
Sekarang kita akan membuat metode **update_employee ()** ke dalam file **employee.php**. Kita akan menggunakan **employee_id** yang ingin kita perbarui dan yang kedua adalah data yang diperbarui dalam format json.

```
96 function update_employee($id)  
97 {  
98     global $connection;  
99     $post_vars = json_decode(file_get_contents("php://input"), true);  
100     $employee_name = $post_vars["employee_name"];  
101     $employee_salary = $post_vars["employee_salary"];  
102     $employee_age = $post_vars["employee_age"];  
103     $query = "UPDATE tb_employee SET employee_name='" . $employee_name . "', employee_salary='" . $employee_salary . "', employee_age='" . $employee_age . "' WHERE id = " . $id;  
104     if (mysqli_query($connection, $query)) {  
105         $response = array(  
106             'status' => 1,  
107             'status_message' => 'Employee Updated Successfully.'  
108         );  
109     } else {  
110         $response = array(  
111             'status' => 0,  
112             'status_message' => 'Employee Updation Failed.'  
113         );  
114     }  
115     header('Content-Type: application/json');  
116     echo json_encode($response);  
117 }
```

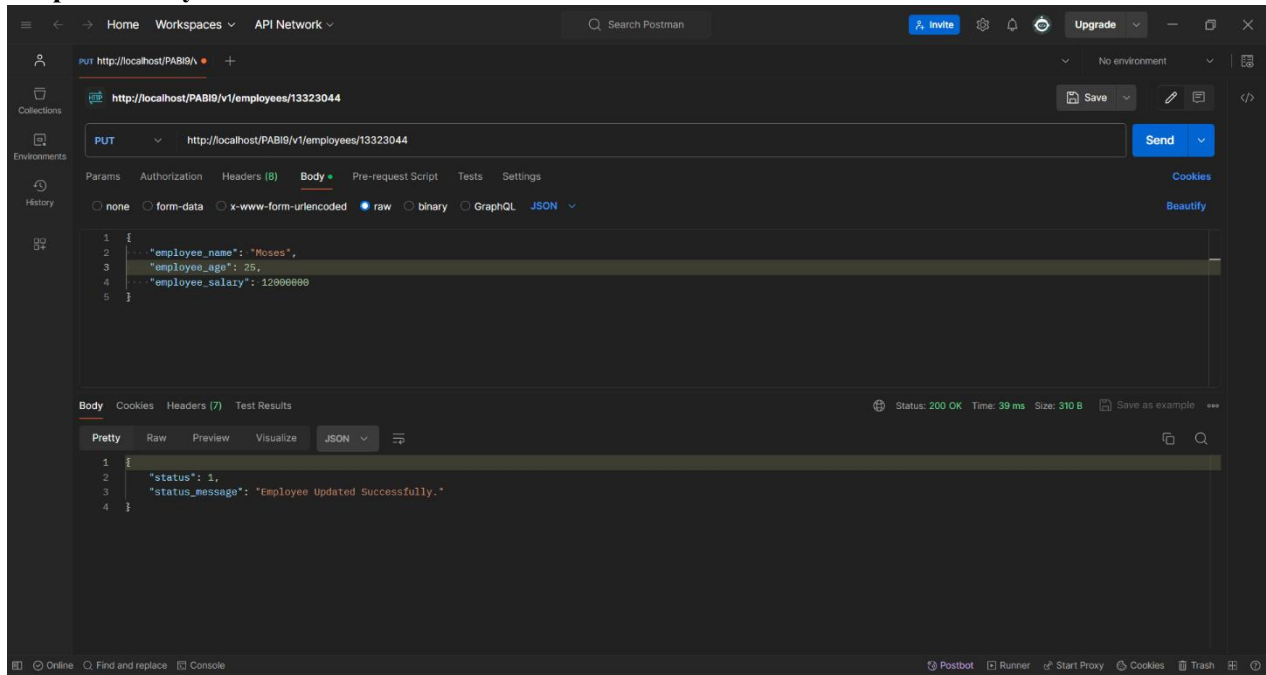
Kami telah mengambil data **employee_id** sebagai parameter dan json untuk memperbarui record ke dalam tabel mysql, jika semuanya benar kemudian mengirim **'updated message'** jika tidak, mengirim **error message**.

Pengujian menggunakan tools Postman

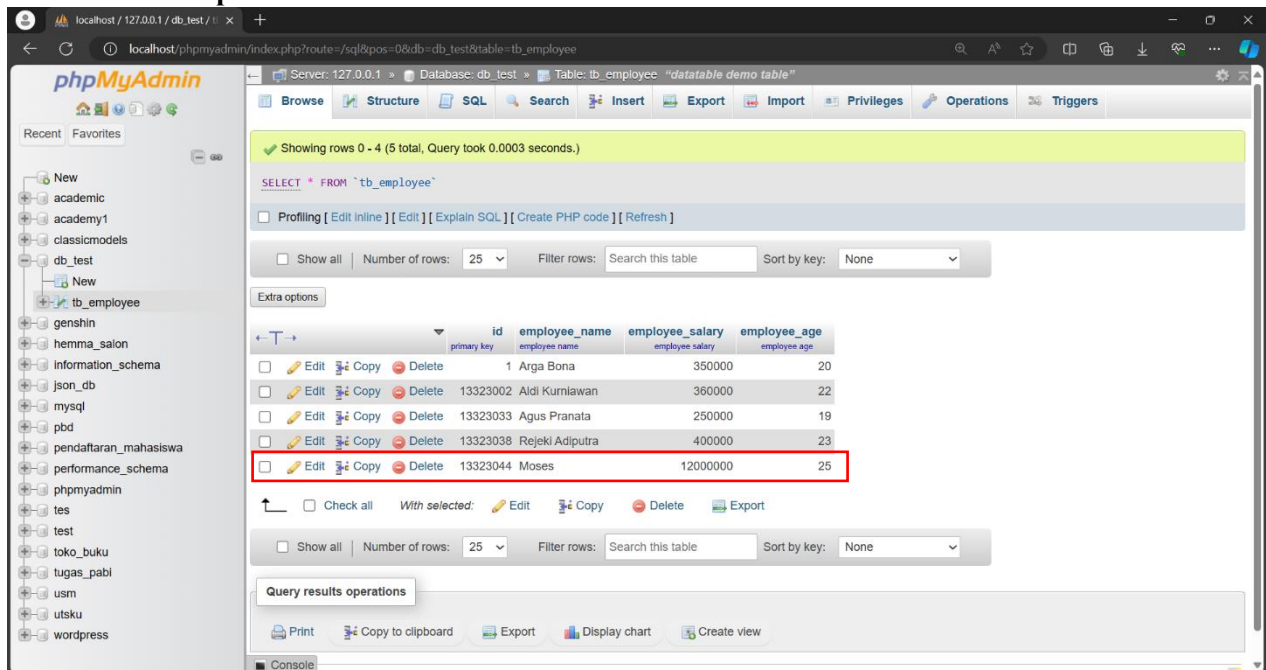
Misalnya moses sudah bekerja bertahun-tahun dan usia moses sekarang sudah 25 tahun dan karena moses adalah karyawan yang cerdas maka gaji moses juga sekarang sudah 12.000.000. Sehingga kita wajib melakukan update data moses (**employee_age** dan **employee_salary**) pada database. Kita akan menguji fungsi update tersebut pada data moses. Pilih method PUT dan ketikkan endpoint sesuai dengan **id moses = 13323044**.



Response Body



Perubahan data pada database



Rest API Untuk Menghapus Record Dari MySQL

kita akan membuat DELETE Type baru permintaan api restful menggunakan php untuk menghapus record karyawan dari database MySQL. Kami akan passing/melewatkan **employee_id** sebagai parameter yang ingin Anda hapus dari tabel karyawan.

Kami akan menambahkan **CASE** baru ke dalam metode **SWITCH** seperti di bawah ini

```

case 'DELETE':
    // Delete Product
    $id = intval($_GET["id"]);
    delete_employee($id);
    break;

```

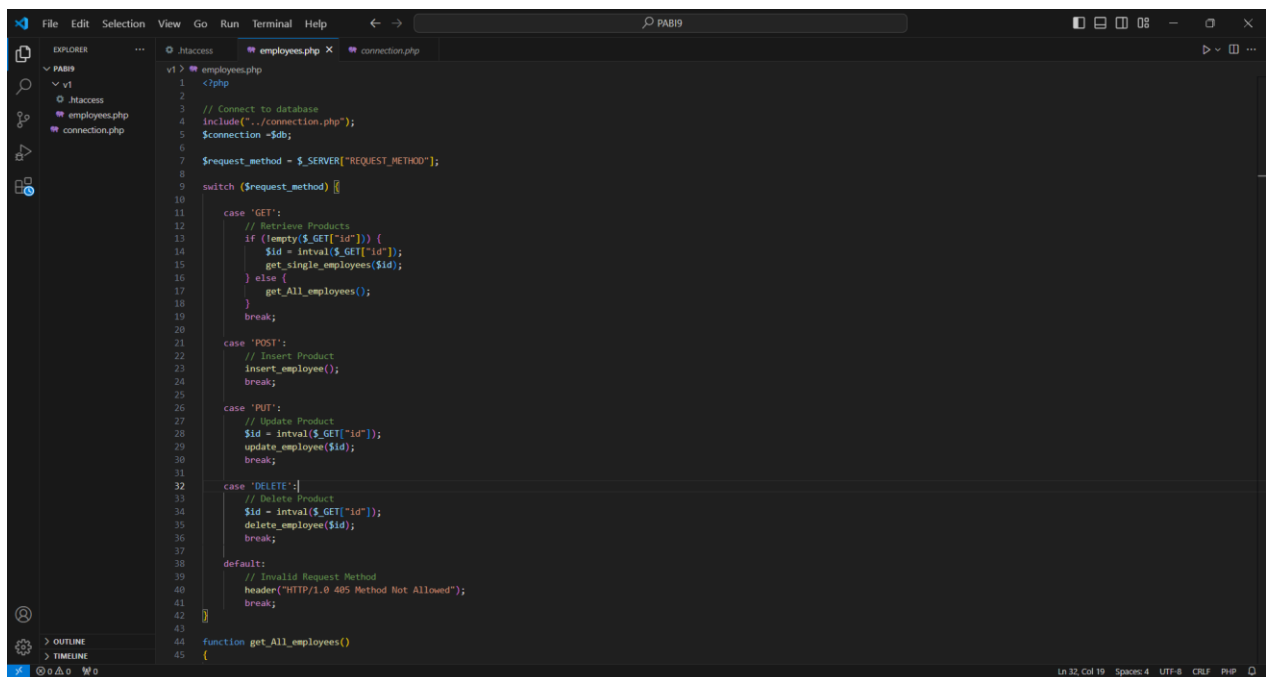
Sekarang kita akan membuat metode `delete_employee ()` ke dalam file `employee.php`. Kita akan menggunakan `employee_id` yang ingin kita hapus record dari tabel `tb_employee`.

```

function delete_employee($id)
{
    global $connection;
    $query = "DELETE FROM tb_employee WHERE id=" . $id;
    if (mysqli_query($connection, $query)) {
        $response = array(
            'status' => 1,
            'status_message' => 'Employee Deleted Successfully.'
        );
    } else {
        $response = array(
            'status' => 0,
            'status_message' => 'Employee Deletion Failed.'
        );
    }
    header('Content-Type: application/json');
    echo json_encode($response);
}

```

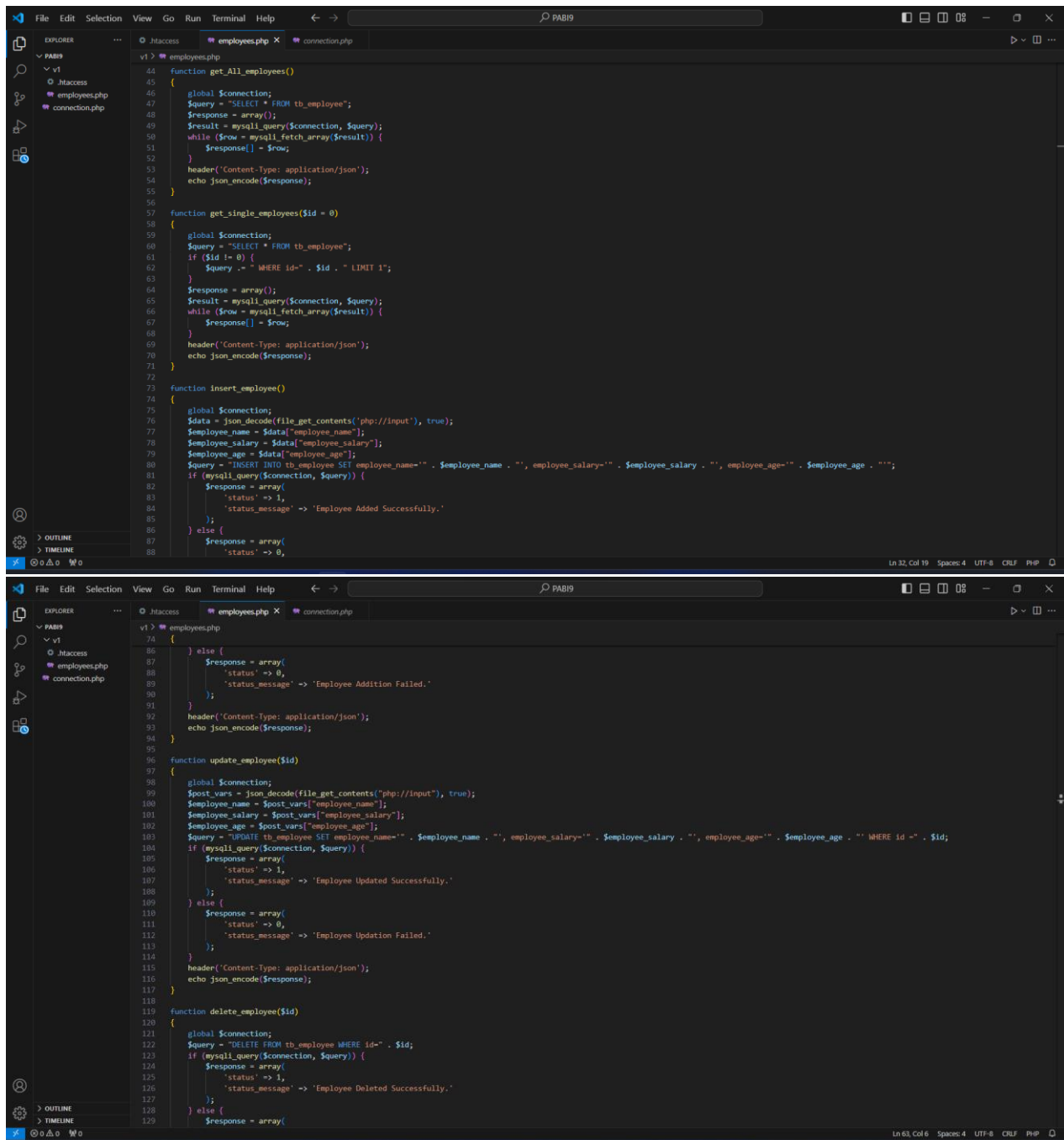
Kode program yang lengkap di `employee.php` adalah berikut.



```

1 <?php
2
3 // Connect to database
4 include("../connection.php");
5 $connection = $db;
6
7 $request_method = $_SERVER["REQUEST_METHOD"];
8
9 switch ($request_method) {
10
11     case 'GET':
12         // Retrieve Products
13         if (!empty($_GET["id"])) {
14             $id = intval($_GET["id"]);
15             get_single_employees($id);
16         } else {
17             get_all_employees();
18         }
19         break;
20
21     case 'POST':
22         // Insert Product
23         insert_employee();
24         break;
25
26     case 'PUT':
27         // Update Product
28         $id = intval($_GET["id"]);
29         update_employee($id);
30         break;
31
32     case 'DELETE':
33         // Delete Product
34         $id = intval($_GET["id"]);
35         delete_employee($id);
36         break;
37
38     default:
39         // Invalid Request Method
40         header("HTTP/1.0 405 Method Not Allowed");
41         break;
42 }
43
44 function get_all_employees()
45 {

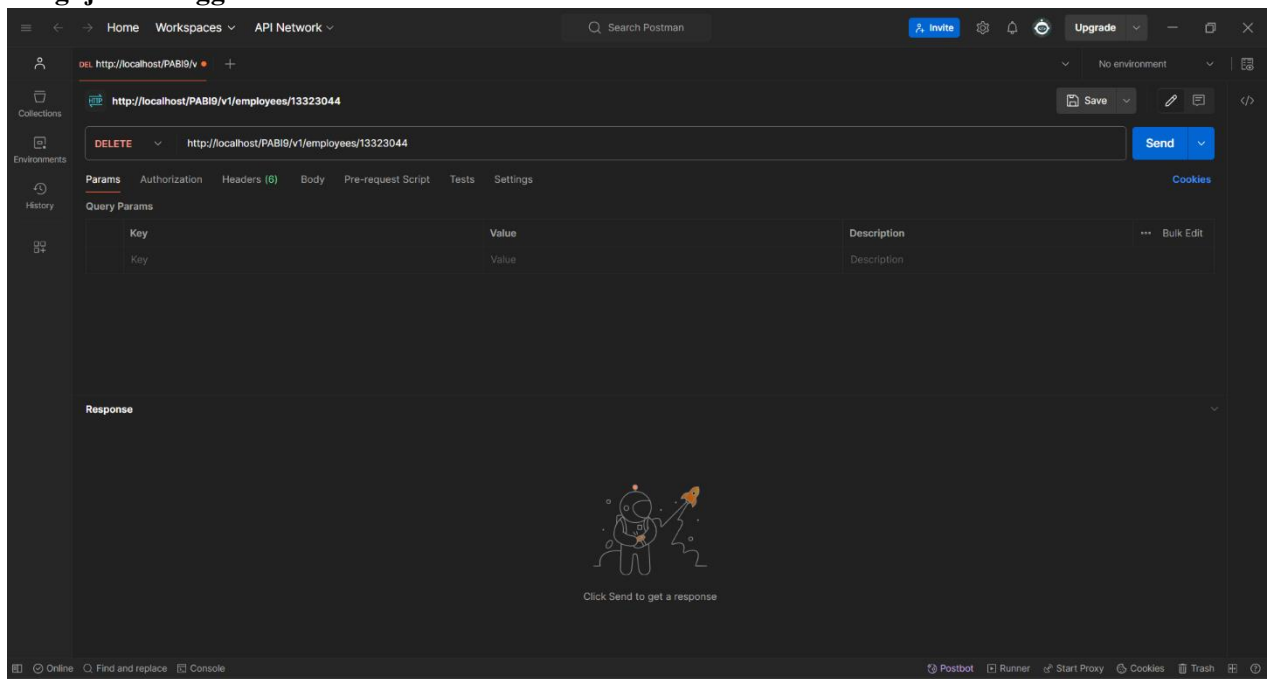
```



The screenshot shows a code editor with a dark theme. The Explorer panel on the left shows a project structure with folders 'PABIG' and 'v1', and files 'employees.php' and 'connection.php'. The main editor area displays the code for 'employees.php'. The code includes a function 'delete_employee(\$id)' that uses a global '\$connection' to execute a DELETE SQL query. It returns a JSON response with 'status' and 'status_message'.

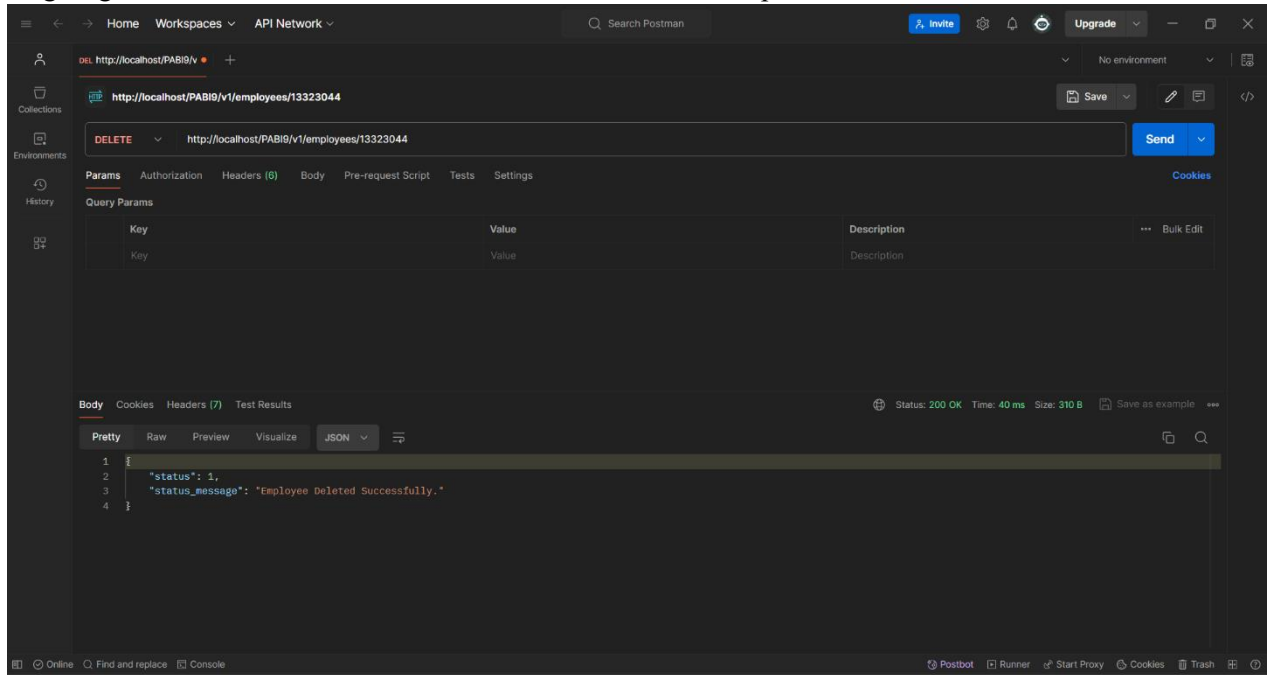
```
97 {
115     header('Content-Type: application/json');
116     echo json_encode($response);
117 }
118
119 function delete_employee($id)
120 {
121     global $connection;
122     $query = "DELETE FROM tb_employee WHERE id=" . $id;
123     if (mysql_query($connection, $query)) {
124         $response = array(
125             'status' => 1,
126             'status_message' => 'Employee Deleted Successfully.'
127         );
128     } else {
129         $response = array(
130             'status' => 0,
131             'status_message' => 'Employee Deletion Failed.'
132         );
133     }
134     header('Content-Type: application/json');
135     echo json_encode($response);
136 }
137
138 ?>
139
```

Pengujian menggunakan tools Postman



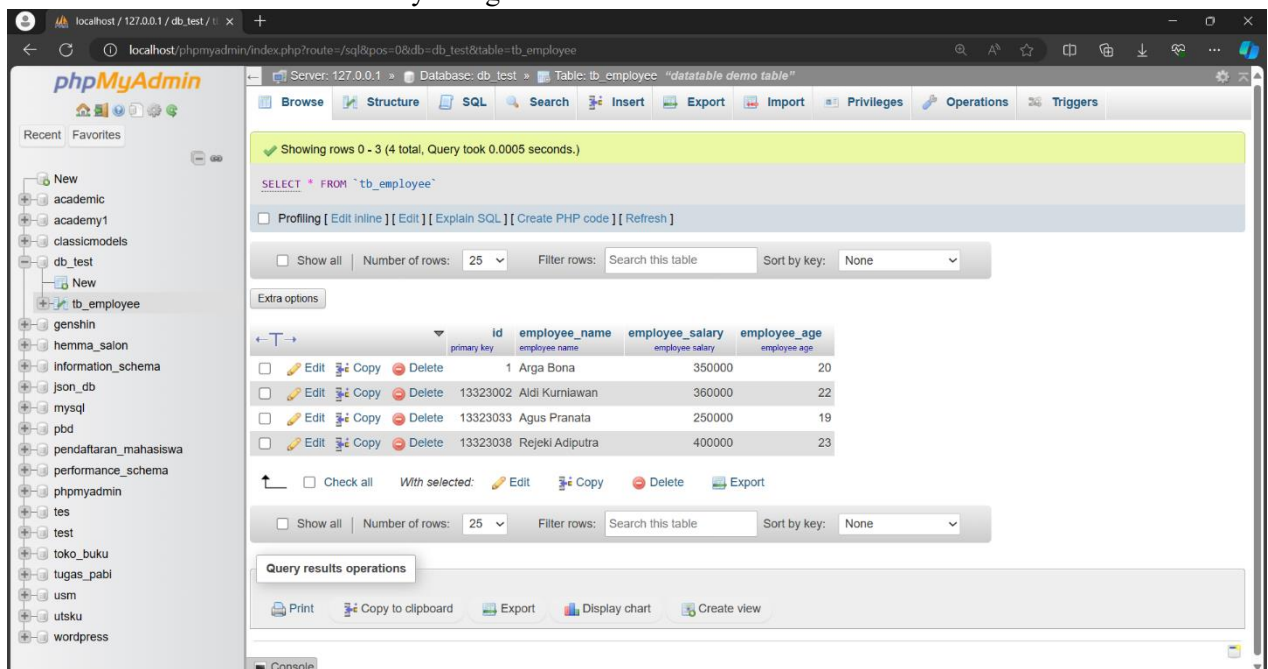
Output

Response body menghasilkan status code 200. Artinya proses delete berhasil. Silahkan anda cek dan refresh langsung ke database untuk memastikan bahwa data sudah terhapus.



Perubahan data pada database

Data Moses sudah tidak ada. Artinya fungsi delete telah berhasil dibuktikan.



Kesimpulan

Kami telah membuat RESTful API untuk mengambil semua karyawan, mendapatkan record karyawan tunggal, membuat entri karyawan baru, memperbarui data karyawan dan menghapus data karyawan dari database MySQL. Anda dapat menambahkan opsi terkait keamanan seperti API token untuk membatasi akses rest api dari pengguna yang tidak diotorisasi.