

Laporan Pemrograman API



Oleh :

Verani Fajrin	(22091397018)
Ihda Anisa Ulfa	(22091397019)
Roy Nurfaza	(22091397035)

Dosen Pengampu :

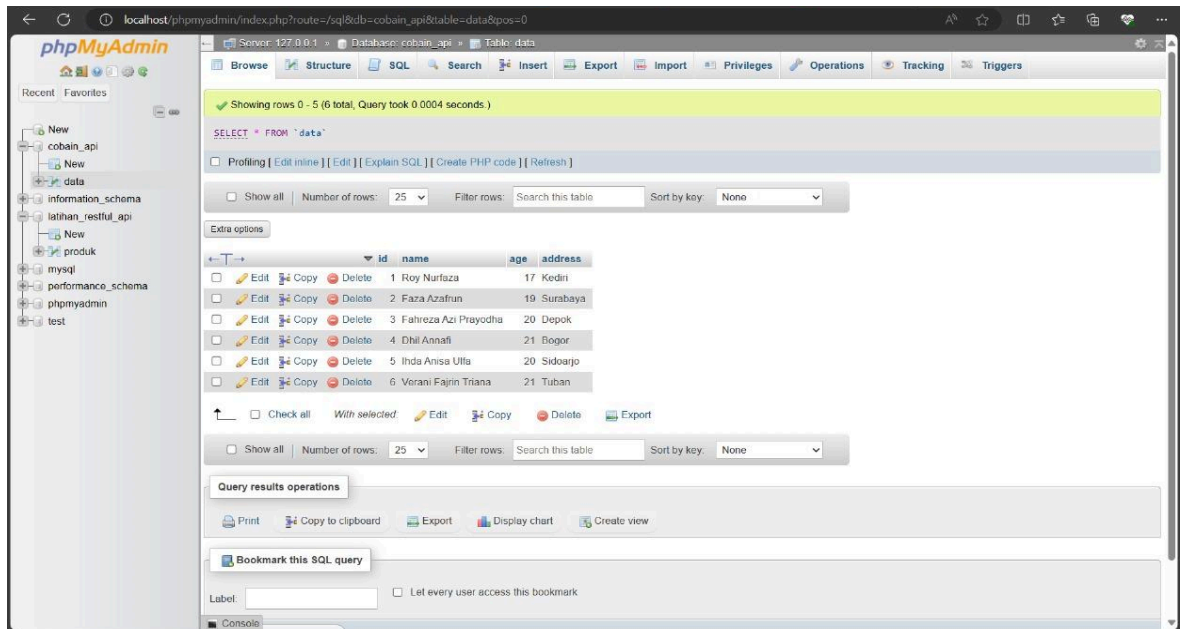
Andi Iwan
NIP.197810272008121002

Github : <https://github.com/22091397018VeraniFajrinTriana/MI2022A>
https://github.com/ihdaanisa/Kelompok08_IhdaAnisaUlfa
https://github.com/roynurff/pemrogramanAPI_1

Program Studi Manajemen Informatika
Fakultas Vokasi
Universitas Negeri Surabaya
2024

Tahap tahap :

1. Membuka XAMPP dan menyalakan module Apache dan MySQL untuk menjalankan server MySQL sampai berwarna hijau.
2. Setelah itu membuka web “phpmyadmin” untuk membuat database dan mengisi tabel data pribadi. Isi dari data tabel dibawah adalah “id”, “name”, “age” dan “address”.



3. Setelah membuat database, maka dilanjutkan dengan membuat code php seperti dibawah ini.



• Penjelasan Code :

- define('HOST', 'localhost');: Baris ini mendefinisikan konstanta HOST dengan nilai 'localhost'. Konstanta ini digunakan untuk mengidentifikasi alamat server database.
- define('USER', 'root');: Konstanta USER didefinisikan dengan nilai 'root'. Ini adalah nama pengguna yang akan digunakan untuk mengakses database.
- define('PASS', "");: Konstanta PASS didefinisikan dengan nilai string kosong ("). Ini menunjukkan bahwa tidak ada kata sandi yang diperlukan untuk mengakses database.

- `define('DB', 'cobain_api');`: Konstanta DB didefinisikan dengan nilai 'cobain_api'. Ini adalah nama database yang akan digunakan dalam koneksi.
 - `$connection = mysqli_connect(HOST, USER, PASS, DB) or die ('Unable connect');`: Baris ini mencoba membuat koneksi ke database menggunakan fungsi `mysqli_connect()`. Jika koneksi gagal, pesan 'Unable connect' akan ditampilkan dan skrip akan berhenti.
4. Langkah selanjutnya yaitu membuat file php baru dengan format “data.php” berikut adalah codenya.

```

1  <?php
2
3      require_once('connection.php');
4
5      if (empty($_GET)) {
6          header("Content-Type: JSON");
7          $query = mysqli_query($connection, "SELECT * FROM data");
8
9          $result = array();
10         while($row = mysqli_fetch_array($query)) {
11             array_push( $result, array (
12                 'id'      => $row['id'],
13                 'name'    => $row['name'],
14                 'age'     => $row['age'],
15                 'address' => $row['address']
16             ));
17         }
18
19         echo json_encode (
20             array ('result' => $result)
21         );
22     } else {
23         $query = mysqli_query( $connection, "SELECT * FROM data WHERE id=" . $_GET["id"] );
24
25         $result = array();
26         while($row = $query->fetch_assoc()) {
27             $result = array (
28                 'id'      => $row['id'],
29                 'name'    => $row['name'],
30                 'age'     => $row['age'],
31                 'address' => $row['address']
32             );
33         }
34
35         echo json_encode ($result);
36     }
37
38     ?>

```

● Penjelasan Code :

- `require_once('connection.php');`: Baris ini memuat file `connection.php`. Biasanya, file ini berisi konfigurasi koneksi ke database, seperti host, nama pengguna, kata sandi, dan nama database.
- `if (empty($_GET)) { ... } else { ... }`: Kode ini memeriksa apakah ada parameter yang diteruskan melalui URL (query string). Jika tidak ada parameter, maka akan mengambil semua data dari tabel “data”. Jika ada parameter “id” dalam query string, maka hanya data dengan ID yang sesuai yang akan diambil.
- `header("Content-Type: JSON");`: Baris ini mengatur tipe konten HTTP header sebagai JSON. Ini menandakan bahwa hasil yang dikembalikan akan berupa data JSON.

- `$query = mysqli_query($connection, "SELECT * FROM data");`: Kode ini menjalankan query SQL untuk mengambil semua data dari tabel “data” menggunakan koneksi yang telah dibuat sebelumnya.
- `while($row = mysqli_fetch_array($query)) { ... }`: Loop ini mengambil setiap baris hasil query dan memasukkannya ke dalam array `$result`. Setiap baris memiliki kolom seperti “id”, “name”, “age”, dan “address”.
- `echo json_encode(array('result' => $result));`: Hasil akhir dari query diubah menjadi format JSON dan dikirimkan sebagai respons ke klien (browser).
- `else { ... }`: Jika ada parameter “id” dalam query string, maka kode ini akan menjalankan query untuk mengambil data dengan ID yang sesuai. Data ini juga diubah menjadi format JSON dan dikirimkan sebagai respons.

5. Selanjutnya yaitu membuat file php baru bernama “post.php”. Berikut adalah codenya :

```

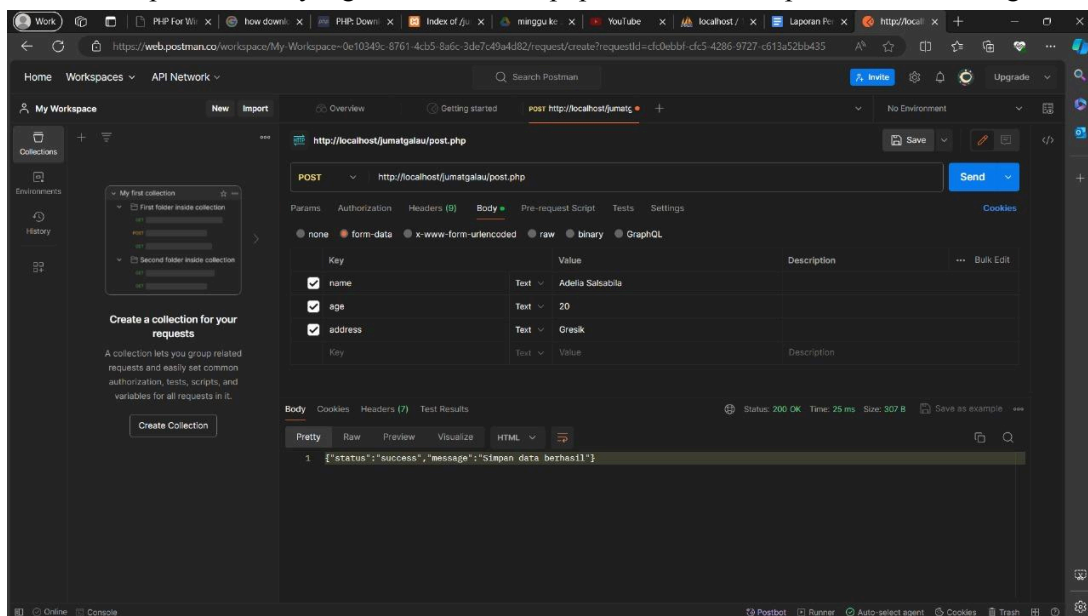
1  <?php
2
3      include "connection.php";
4
5      $nama_mhs = isset($_POST["name"]) ? $_POST['name'] : ""; //Mengambil nilai dari
6      //echo $nama_mhs;
7
8      $umur = isset($_POST["age"]) ? $_POST['age'] : "";
9      //echo $umur;
10
11     $alamat = isset($_POST["address"]) ? $_POST['address'] : "";
12     //echo $alamat;
13
14
15     $sql = "INSERT INTO `data` (`name`, `age`, `address`)
16     VALUES ('".$nama_mhs."', '".$umur."', '".$alamat."')"; //menyimpan data ke
17
18     //echo $sql;
19
20     $query = mysqli_query($connection, $sql);
21     if($query){
22         $msg = "Simpan data berhasil";
23     }else{
24         $msg = "Gagal mencetak hasil";
25     }
26
27     //echo $msg;
28
29     $response = array(
30         "status" => "success",
31         "message" => $msg
32     );
33
34     echo json_encode($response);
35 ?>

```

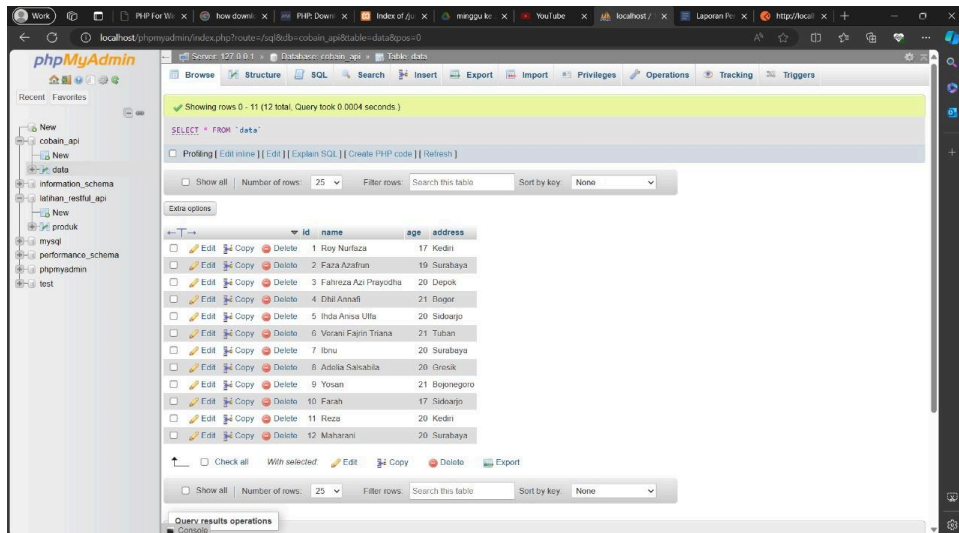
● Penjelasan Code :

- `include "connection.php";`: Baris ini memuat file `connection.php`. File ini kemungkinan berisi konfigurasi koneksi ke database, seperti host, nama pengguna, kata sandi, dan nama database.
- `$nama_mhs = isset($_POST["name"]) ? $_POST['name'] : "";`: Kode ini mengambil nilai dari parameter POST dengan nama “name”. Jika parameter “name” ada, maka nilai akan disimpan dalam variabel `$nama_mhs`. Jika tidak ada, variabel akan diisi dengan string kosong.

- `$umur = isset($_POST["age"]) ? $_POST['age'] : "";` Sama seperti sebelumnya, kode ini mengambil nilai dari parameter POST dengan nama “age” dan menyimpannya dalam variabel \$umur.
 - `$alamat = isset($_POST["address"]) ? $_POST['address'] : "";` Kode ini mengambil nilai dari parameter POST dengan nama “address” dan menyimpannya dalam variabel \$alamat.
 - `$sql = "INSERT INTO data (name, age, address) VALUES ('".$nama_mhs."', '".$umur."', '".$alamat."')";` Baris ini membangun query SQL untuk menyimpan data ke dalam tabel “data”. Data yang akan disimpan adalah nama, umur, dan alamat yang diambil dari variabel-variabel sebelumnya.
 - `$query = mysqli_query($connection, $sql);` Kode ini menjalankan query SQL menggunakan koneksi ke database yang telah dibuat sebelumnya. Query ini akan menyimpan data ke dalam tabel “data”.
 - `if($query){ ... }else{ ... };` Kode ini memeriksa apakah query berhasil dieksekusi atau tidak. Jika berhasil, variabel \$msg akan berisi pesan “Simpan data berhasil”. Jika gagal, variabel akan berisi pesan “Gagal mencetak hasil”.
 - `$response = array("status"=> "success", "message" => $msg);` Hasil akhir dari proses disusun dalam bentuk array dengan status “success” dan pesan yang sesuai.
 - `echo json_encode($response);` Hasil akhir diubah menjadi format JSON dan dikirimkan sebagai respons ke klien (browser).
6. Setelah penulisan code diatas sudah benar dan tidak ada error, maka selanjutnya adalah membuka web “postman” untuk menjalankan “Restful API” . Untuk cara kerjanya, yaitu bisa melakukan login di website “web.postman.co” ataupun bisa mengunduhnya terlebih dahulu. Setelah itu upload folder yang sudah ada code php dan setelah terupload maka bisa digunakan.



Pada gambar diatas digunakan untuk menambahkan data pada database, hasil penambahan data bisa dilihat pada gambar dibawah ini.



7. Setelah semua code berjalan normal tanpa ada error, maka dilanjutkan dengan membuat fungsi “Put” yang digunakan untuk mengupdate database. Berikut adalah codenya

```

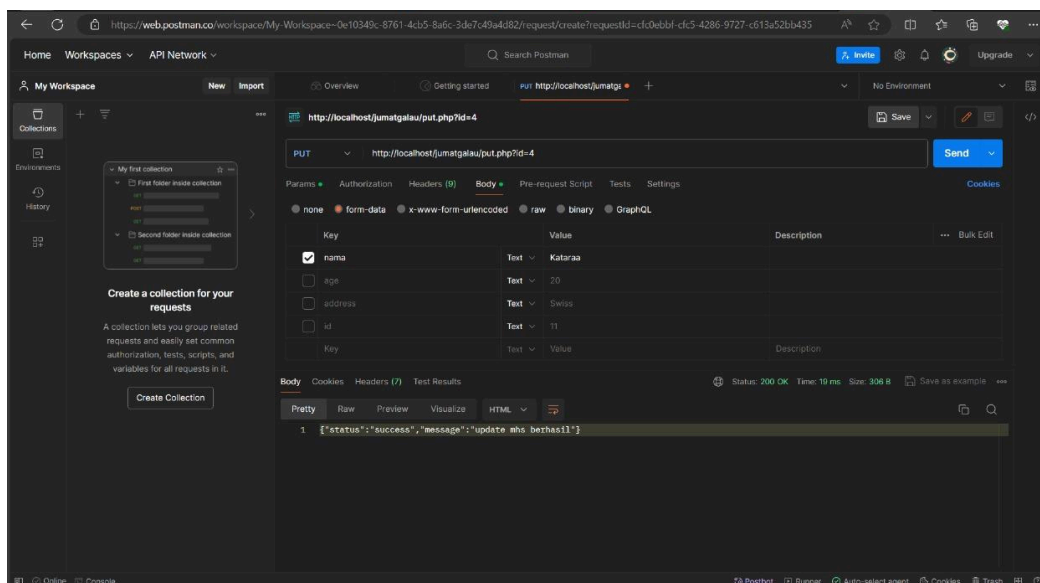
1  <?php
2
3      include "connection.php";
4
5      // echo "update mhs";
6
7      $id = $_GET['id'];
8      // echo $id;
9
10     $nama_mhs = isset($_POST["nama"]) ? $_POST['nama'] : ""; //Mengambil nilai dari
11     //echo $nama_mhs;
12
13     $umur = isset($_POST["age"]) ? $_POST['age'] : "";
14     //echo $umur;
15
16     $alamat = isset($_POST["address"]) ? $_POST['address'] : "";
17
18     $sql = "UPDATE `data` SET `name` = '".$nama_mhs."', `address` = '".$alamat.'"
19     WHERE `data`.`id` = ".$id.";";
20     //echo $sql;
21
22     $query = mysqli_query($connection, $sql);
23     if($query){
24         $msg = "update mhs berhasil";
25     }else{
26         $msg = "update mhs gagal";
27     }
28
29     $response = array(
30         "status"=> "success",
31         "message" => $msg
32     );
33
34     echo json_encode($response);
35
36     ?>

```

- Penjelasan Code :

- include "connection.php";: Baris ini memuat file connection.php. File ini kemungkinan berisi konfigurasi koneksi ke database, seperti host, nama pengguna, kata sandi, dan nama database. \$nama_mhs = isset(\$_POST["name"]) ?
- \$_POST["name"] : "";: Kode ini mengambil nilai dari parameter POST dengan nama "name". Jika parameter "name" ada, maka nilai akan disimpan dalam variabel \$nama_mhs. Jika tidak ada, variabel akan diisi dengan string kosong.
- \$umur = isset(\$_POST["age"]) ? \$_POST["age"] : "";: Sama seperti sebelumnya, kode ini mengambil nilai dari parameter POST dengan nama "age" dan menyimpannya dalam variabel \$umur.
- \$alamat = isset(\$_POST["address"]) ? \$_POST["address"] : "";: Kode ini mengambil nilai dari parameter POST dengan nama "address" dan menyimpannya dalam variabel \$alamat.
- \$sql = "INSERT INTO data (name, age, address) VALUES ('".\$nama_mhs."','".\$umur."','".\$alamat."')";: Baris ini membangun query SQL untuk menyimpan data ke dalam tabel "data". Data yang akan disimpan adalah nama, umur, dan alamat yang diambil dari variabel-variabel sebelumnya.
- \$query = mysqli_query(\$connection, \$sql);: Kode ini menjalankan query SQL menggunakan koneksi ke database yang telah dibuat sebelumnya. Query ini akan menyimpan data ke dalam tabel "data".
- if(\$query){ ... }else{ ... }: Kode ini memeriksa apakah query berhasil dieksekusi atau tidak. Jika berhasil, variabel \$msg akan berisi pesan "Simpan data berhasil". Jika gagal, variabel akan berisi pesan "Gagal mencetak hasil".
- \$response = array("status"=> "success", "message"=> \$msg);: Hasil akhir dari proses disusun dalam bentuk array dengan status "success" dan pesan yang sesuai.
- echo json_encode(\$response);: Hasil akhir diubah menjadi format JSON dan dikirimkan sebagai respons ke klien (browser).

8. Setelah code diatas berhasil, maka kembali lagi ke website "Postman" untuk mencoba fungsi "PUT". Gambar di Bawah Ini Menunjukkan Hasil Percobaan Fungsi PUT. Pada gambar tersebut, terlihat bahwa fungsi "PUT" berjalan dengan baik dan berhasil mengupdate data di dalam database mahasiswa.



9. Untuk fungsi yang terakhir adalah fungsi “DELETE”. Fungsi ini bekerja untuk menghapus data dalam database. Berikut adalah codenya

```
1  <?php
2
3  $method = $_SERVER['REQUEST_METHOD'];
4
5  $result = array();
6
7  if($method=="DELETE"){
8
9      parse_str(file_get_contents('php://input'), $_DELETE);
10
11      if(isset($_DELETE['id'])){
12
13          $id = $_DELETE['id'];
14
15          $result['status'] = [
16              "code" => 200,
17              "msg"=> "successful request",
18          ];
19
20          $servername = "localhost";
21          $username = "root";
22          $password = "";
23          $dbname = "cobain_api";
24
25
26          $conn = new mysqli($servername, $username, $password, $dbname);
27
28          $sql = "DELETE FROM data WHERE id='$id' ";
29
30          $conn->query($sql);
31
32          $result['results'] = [
33              "id" => $id
34          ];
35
36      }else {
37          $result["status"] = [
38              'code'=>400,
39              'msg'=>'Missing parameter'
40          ];
41      }
42
43  } else {
44      $result['status'] = [
45          'code'=>400,
46          'msg'=> 'Method not implemented for this address'
47      ];
48  }
49
50
51  echo json_encode($result);
52
53
54  ?>
```

● Penjelasan Code :

- include “connection.php”;; Baris ini memuat file connection.php. File connection.php kemungkinan berisi konfigurasi koneksi ke database, seperti host, nama pengguna, kata sandi, dan nama database.
- \$method = \$_SERVER['REQUEST_METHOD']; Kode ini mengambil nilai dari variabel lingkungan \$_SERVER['REQUEST_METHOD']. Nilai ini menunjukkan metode permintaan HTTP yang digunakan (misalnya, GET, POST, atau DELETE).
- parse_str(file_get_contents('php://input'), \$_DELETE); Kode ini menguraikan parameter dari tubuh permintaan menggunakan fungsi

`file_get_contents('php://input')`). Hasilnya disimpan dalam variabel `$_DELETE`.

- `if(isset($_DELETE['id'])) {...} else {...}`: Kode ini memeriksa apakah parameter id ada dalam permintaan DELETE. Jika ada, kode akan melanjutkan untuk menghapus data dengan ID yang sesuai dari tabel. Jika tidak ada, kode akan memberikan pesan “Parameter hilang.”
- Koneksi ke Database: Kode selanjutnya membuat koneksi ke database MySQL menggunakan informasi server, nama pengguna, kata sandi, dan nama database yang telah ditentukan.
- `$sql = "DELETE FROM data WHERE id='$id' ";` Kode ini membangun perintah SQL untuk menghapus data dari tabel data berdasarkan nilai id yang diterima.
- `$conn->query($sql);` Kode ini menjalankan perintah SQL menggunakan koneksi database yang telah dibuat sebelumnya.
- `$result['results'] = ["id" => $id];` Hasil operasi penghapusan akan dikemas dalam bentuk respon JSON. ID yang dihapus akan disertakan dalam respon.
- Respon JSON: Hasil operasi (berhasil atau gagal) akan dikemas dalam bentuk respon JSON. Status kode dan pesan yang sesuai akan disertakan dalam respon.

10. Setelah code diatas berhasil, maka kembali lagi ke website “Postman” untuk mencoba fungsi “DELETE”. Gambar di Bawah Ini Menunjukkan Hasil Percobaan Fungsi “DELETE”. Pada gambar tersebut, terlihat bahwa fungsi “PUT” berjalan dengan baik dan berhasil menghapus data di dalam database.

