



NAMA : FAHTA HANDINSA MAULANA PUTRA

NIM : 22.01.53.0052

PRODI : TEKNIK INFORMATIKA

WEB SERVICE

ULANGAN TENGAH SEMESTER

1. Membuat Database `mma_gym` yang diberi 10 tabel

The screenshot shows the phpMyAdmin interface for a database named 'mma_gym'. The left sidebar shows the database structure, and the main area displays a table of 10 tables. The table has columns: Tabel, Tindakan, Baris, Jenis, Penyortiran, Ukuran, and Beban. The tables listed are: attendance, classes, equipment, events, feedback, instructors, members, memberships, payments, and schedules. Each table has a 'Tindakan' column with icons for 'Jelajahi', 'Struktur', 'Cari', 'Tambahkan', 'Kosongkan', and 'Hapus'. The 'Baris' column shows the number of rows for each table. The 'Jenis' column shows the engine (InnoDB). The 'Penyortiran' column shows the collation (latin1_swedish_ci). The 'Ukuran' column shows the size in KB. The 'Beban' column shows the character set (utf8).

Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban
attendance	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	latin1_swedish_ci	48 KB	-
classes	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	latin1_swedish_ci	32 KB	-
equipment	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	latin1_swedish_ci	16 KB	-
events	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	latin1_swedish_ci	16 KB	-
feedback	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	latin1_swedish_ci	32 KB	-
instructors	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	latin1_swedish_ci	32 KB	-
members	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	2	InnoDB	latin1_swedish_ci	48 KB	-
memberships	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	3	InnoDB	latin1_swedish_ci	16 KB	-
payments	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	latin1_swedish_ci	32 KB	-
schedules	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	latin1_swedish_ci	32 KB	-
10 tabel	Jumlah	5	InnoDB	latin1_swedish_ci	304 KB	0 B

2. Membuat `api.php`

Masing masing memiliki perintah untuk GET,POST,PUT,DELETE

```
C: > samp > htdocs > mma > api.php
1 <?php
2 // Pengatur header untuk format JSON
3 header("Content-Type: application/json");
4
5 // Menghubungkan ke database
6 $host = 'localhost'; // Ganti dengan host database Anda
7 $db = 'mma_gym'; // Nama database
8 $user = 'root'; // Ganti dengan username database Anda
9 $pass = ''; // Ganti dengan password database Anda
10
11 try {
12     $pdo = new PDO("mysql:host=$host;dbname=$db", $user, $pass);
13     $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
14 } catch (PDOException $e) {
15     echo json_encode(["error" => "Connection failed: " . $e->getMessage()]);
16     exit();
17 }
18
19 // Menangani permintaan API
20 $request_method = $_SERVER["REQUEST_METHOD"];
21
22 switch ($request_method) {
23     case 'GET':
24         if (isset($_GET['id'])) {
25             getMember($pdo, $_GET['id']);
26         } else {
27             getMembers($pdo);
28         }
29         break;
30
31     case 'POST':
32         addMember($pdo);
33         break;
34
35     case 'PUT':
36         updateMember($pdo);
37         break;
```

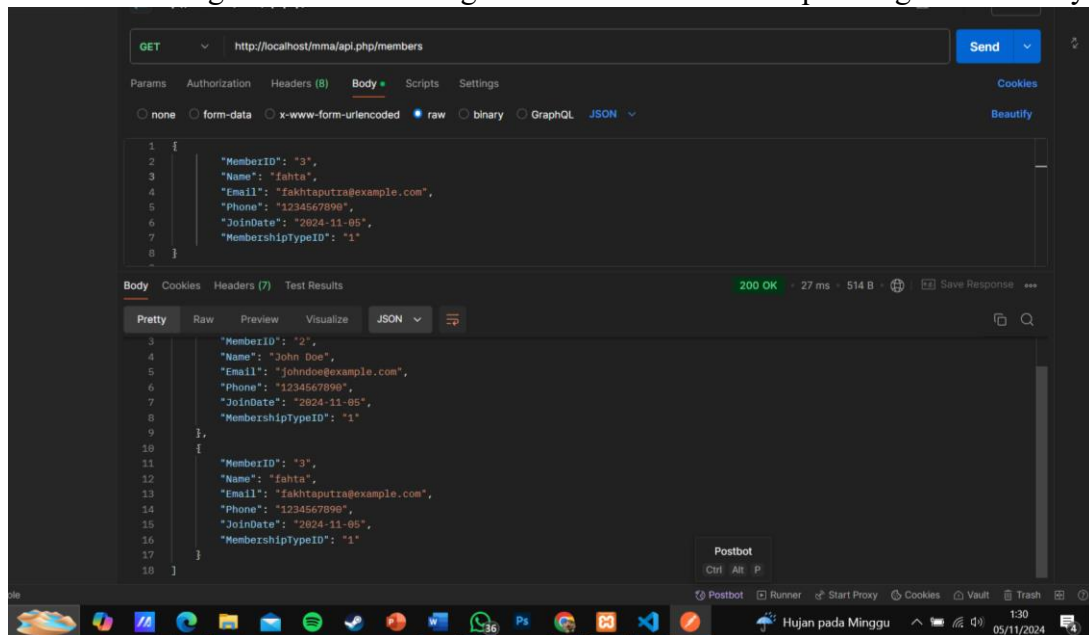
```

C:\xampp\htdocs> mma > api.php
22 switch ($request_method) {
39     case 'DELETE':
40         deleteMember($pdo);
41         break;
42     default:
43         echo json_encode(['error' => 'Invalid request method']);
44         break;
45     }
46 }
47
48 // Fungsi untuk mendapatkan semua anggota
49 function getMembers($pdo) {
50     $stmt = $pdo->query("SELECT * FROM Members");
51     $members = $stmt->fetchAll(PDO::FETCH_ASSOC);
52     echo json_encode($members);
53 }
54
55 // Fungsi untuk mendapatkan anggota berdasarkan ID
56 function getMember($pdo, $id) {
57     $stmt = $pdo->prepare("SELECT * FROM Members WHERE MemberID = ?");
58     $stmt->execute([$id]);
59     $member = $stmt->fetch(PDO::FETCH_ASSOC);
60
61     if ($member) {
62         echo json_encode($member);
63     } else {
64         echo json_encode(['error' => 'Member not found']);
65     }
66 }
67
68 // Fungsi untuk menambahkan anggota
69 function addMember($pdo) {
70     $data = json_decode(file_get_contents("php://input"));
71
72     $stmt = $pdo->prepare("INSERT INTO Members (Name, Email, Phone, JoinDate, MembershipTypeID) VALUES (?, ?, ?, ?, ?)");
73     if ($stmt->execute([$data->Name, $data->Email, $data->Phone, $data->JoinDate, $data->MembershipTypeID])) {
74         echo json_encode(['success' => 'Member added successfully']);
75     }
76 }
77
78
79
80 // Fungsi untuk memperbarui anggota
81 function updateMember($pdo) {
82     $data = json_decode(file_get_contents("php://input"));
83
84     $stmt = $pdo->prepare("UPDATE Members SET Name = ?, Email = ?, Phone = ?, JoinDate = ?, MembershipTypeID = ? WHERE MemberID = ?");
85     if ($stmt->execute([$data->Name, $data->Email, $data->Phone, $data->JoinDate, $data->MembershipTypeID, $data->MemberID])) {
86         echo json_encode(['success' => 'Member updated successfully']);
87     } else {
88         echo json_encode(['error' => 'Failed to update member']);
89     }
90 }
91
92 // Fungsi untuk menghapus anggota
93 function deleteMember($pdo) {
94     $data = json_decode(file_get_contents("php://input"));
95
96     $stmt = $pdo->prepare("DELETE FROM Members WHERE MemberID = ?");
97     if ($stmt->execute([$data->MemberID])) {
98         echo json_encode(['success' => 'Member deleted successfully']);
99     } else {
100         echo json_encode(['error' => 'Failed to delete member']);
101     }
102 }
103
104

```

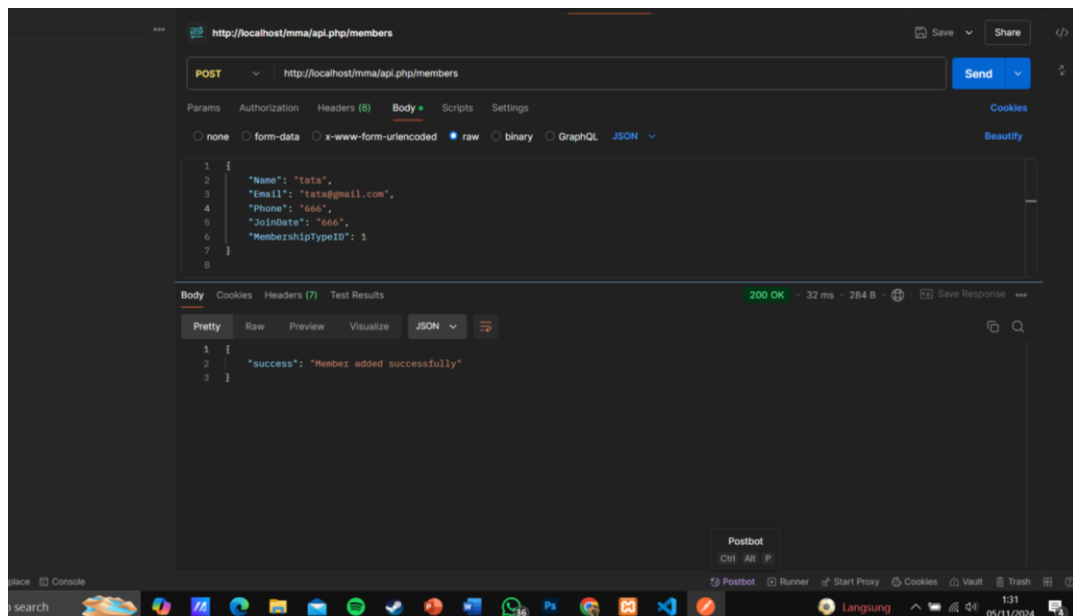
3. Menjalankan Perintah Api.php GET

Metode GET digunakan untuk mengambil data dari server tanpa mengubah statusnya.

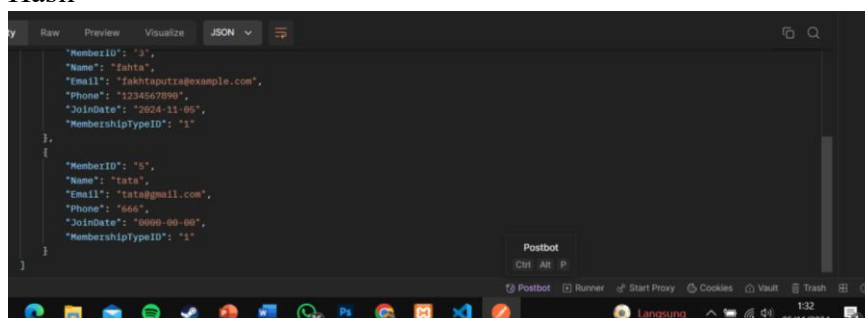


4. Menjalankan Perintah API :

digunakan untuk mengirim data baru ke server, yang dalam hal ini adalah menambahkan data baru ke members

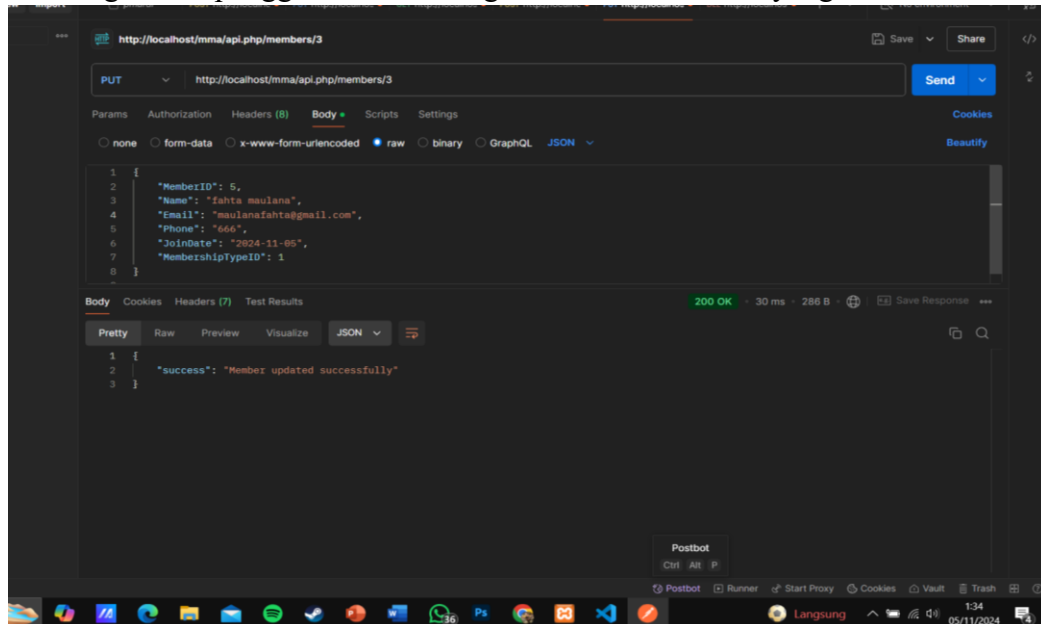


Hasil

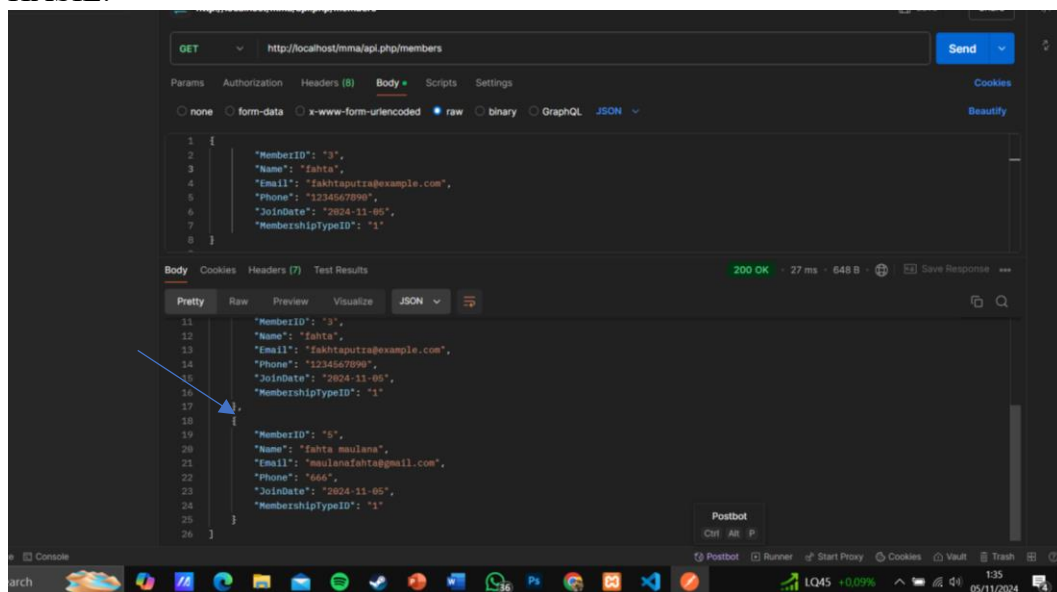


5. Menjalankan Perintah API : PUT

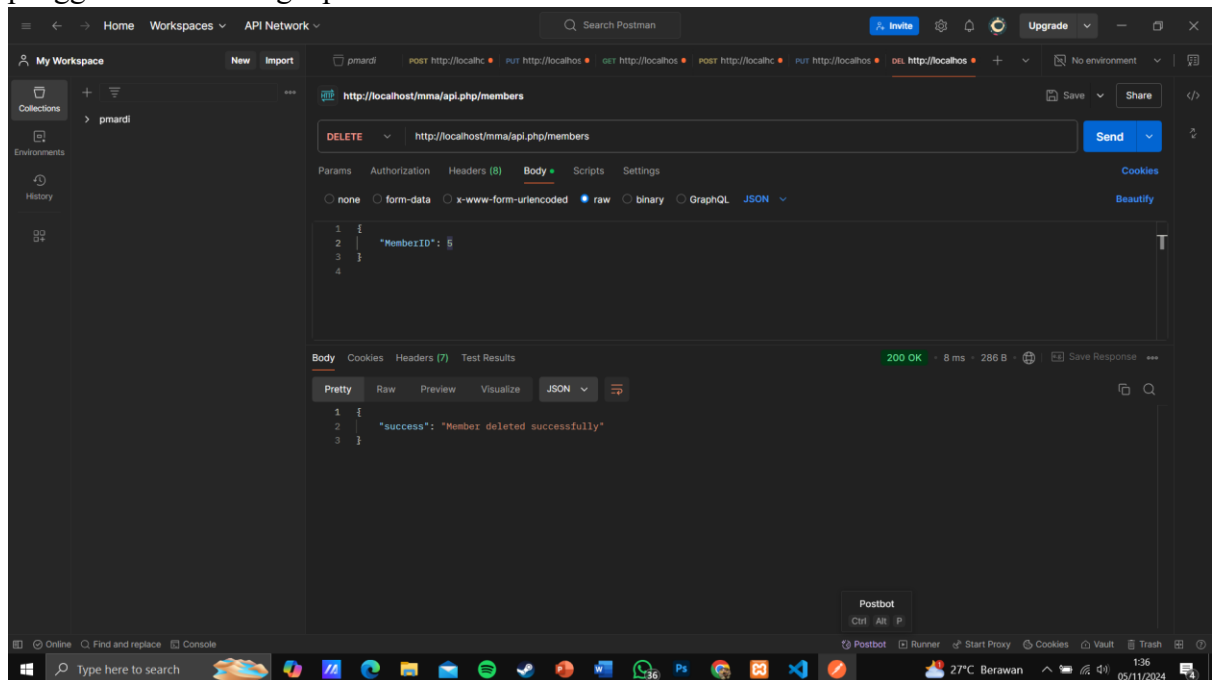
digunakan untuk memperbarui data yang sudah ada di server berdasarkan ID. Ini memungkinkan pengguna untuk mengubah informasi buku yang sudah ada.



(mengubah data sebelumnya menjadi fahta maulana dari yang sebelumnya tata)
HASIL:

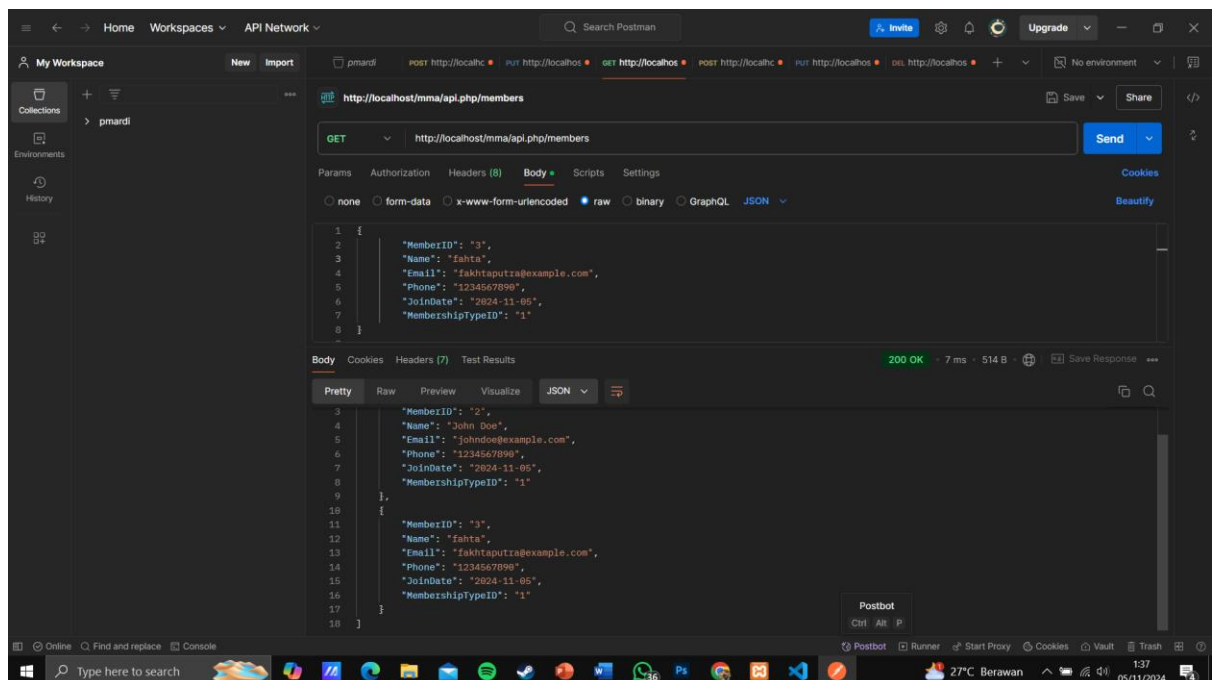


6. Menjalankan Perintah API : Delete digunakan untuk menghapus data dari server berdasarkan ID. Ini memungkinkan pengguna untuk menghapus buku tertentu dari database.



(Menghapus data ke 5)

HASIL :



Kosong