

**LAPORAN PRAKTIKUM**  
**ALGORITMA PEMROGRAM II**  
**APPLICATION PROGRAMMING INTERFACE**



Disusun Oleh:

Ahnaf Umar

164241025

**TEKNOLOGI SAINS DATA**  
**FAKULTAS TEKNOLOGI MAJU DAN MULTIDIPLIN**  
**UNIVERSITAS AIRLANGGA**

**2025**

Membuat kode API tanpa terkoneksi dengan database

1. Buat folder untuk menempatkan 2 file PHP. file untuk dasar API dan file untuk menangani request HTTP
2. Untuk file dasar API isi dengan kode berikut

```
<?php

class Mahasiswa #untuk mengelompokkan fungsi yang berkaitan dengan mahasiswa
{

    public function get_mhs() #fungsi untuk mengirim data mahasiswa
    {

        $data="hore aku API"; #variabel untuk memasukkan data mahasiswa ke dalam variabel response

        #untuk membuat struktur data yang akan dikirim kepada client
        $response=array(
            'status' => 1,
            'message' => 'Get List Mahasiswa Succes.',
            'data' => $data
        );
        header('Content-Type: application/json'); #untuk memberitahu output dari API adalah JSON
        echo json_encode($response); #untuk mengubah variabel response menjadi JSON dan menampilkannya
    }
}
?>
```

3. Untuk file yang mengani request bisa diisi dengan kode berikut

```
<?php
require_once "methods.php"; #untuk mengimportkan file bernama "methods.php"
$mhs = new Mahasiswa(); #membuat objek baru dengan class mahasiswa
$request_method=$_SERVER["REQUEST_METHOD"]; #untuk menyimpan metode HTTP dari request
#mengecek jenis request yang masuk
switch ($request_method) {
    case 'GET': #untuk menampilkan data
        $mhs->get_mhs();
        break;
    case 'POST': #untuk menambah atau mengganti data
        if(!empty($_GET["id"]))
        {
            $id=intval($_GET["id"]);
            $mhs->update_mhs($id);
        }
}
```

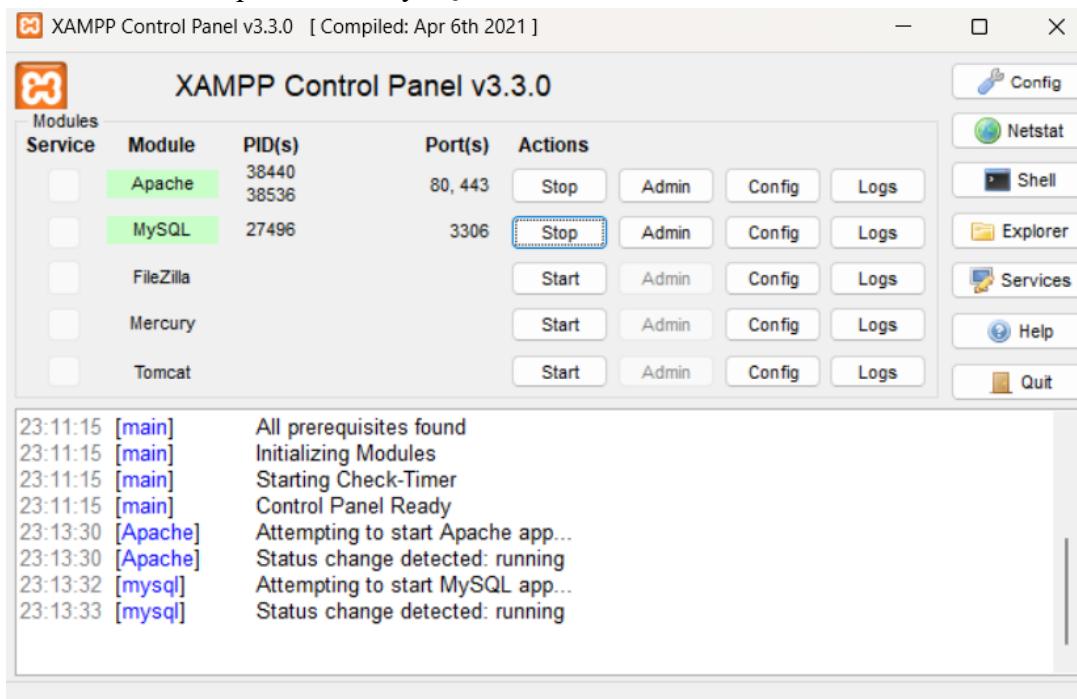
```

else
{
    $mhs->insert_mhs();
}
break;
case 'DELETE': #untuk menghapus data
    $id=intval($_GET["id"]);
    $mhs->delete_mhs($id);
    break;
default:

    header("HTTP/1.0 405 Method Not Allowed"); #untuk menampilkan pesan
error jika klien memasukkan request yang salah
    break;
break;
}
?>

```

4. Letakkan folder yang berisi kedua file tersebut kedalam folder htdocs
5. Buka XAMPP control panel
6. Klik start untuk Apache dan MySQL



7. Buka postman
8. Create new request
9. Masukkan URL dengan [http://localhost/\(nama folder api\)/\(nama file untuk menangani request\).php](http://localhost/(nama folder api)/(nama file untuk menangani request).php)
10. Klik send

The screenshot shows the Postman application interface. At the top, there's a navigation bar with Home, Workspaces, API Network, a search bar for 'Search Postman', and various icons for invite, upgrade, and settings. Below the navigation is a left sidebar with sections for Collections, Environments, Flows, and History. The main workspace is titled 'Overview' and shows a single request: a GET request to 'http://localhost/API/API.php'. The 'Params' tab is selected, showing a table with one row: 'Key' (Key) and 'Value' (Value). The 'Headers' tab shows six headers. The 'Body' tab is set to JSON and displays the following response:

```
{}  
1 {  
2     "status": 1,  
3     "message": "Get List Mahasiswa Succes.",  
4     "data": "hoile aku API"  
5 }
```

The 'Test Results' tab shows a 200 OK status with a response time of 17 ms and a size of 319 B. At the bottom, there are buttons for Runner, Start Proxy, Cookies, Vault, Trash, and a help icon.