

Pertemuan-8

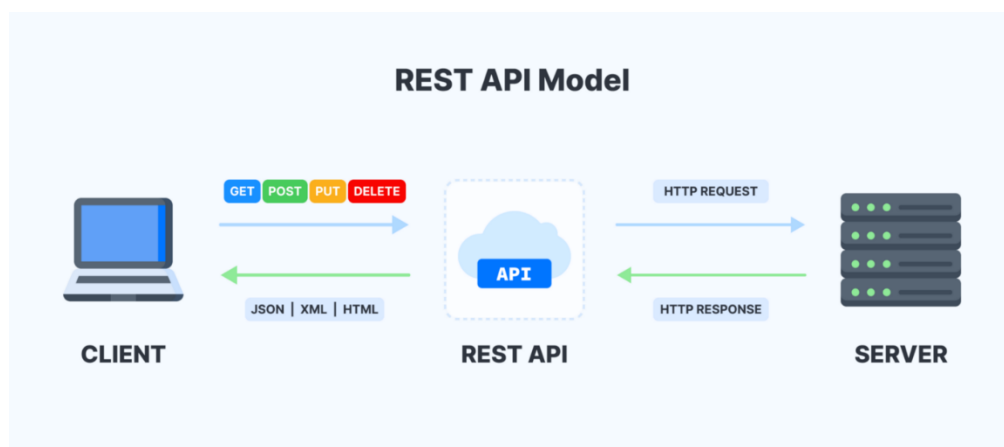
API Programming

A. API Programming

API adalah singkatan dari Application Programming Interface, yang merupakan kumpulan aturan, protokol, dan alat yang digunakan oleh pengembang perangkat lunak untuk berkomunikasi dengan aplikasi atau layanan lain. API programming adalah proses membuat atau menggunakan API untuk memungkinkan integrasi dan interaksi antara aplikasi atau layanan yang berbeda.

API programming memungkinkan pengembang perangkat lunak untuk membuat aplikasi yang dapat saling berkomunikasi, berbagi data, dan berinteraksi dengan layanan atau sistem lain. Dengan menggunakan API, pengembang dapat mengakses dan menggunakan fitur-fitur dari aplikasi atau layanan yang berbeda, tanpa perlu memahami seluruh detail bagaimana aplikasi atau layanan tersebut bekerja.

Contoh penggunaan API programming termasuk integrasi antara aplikasi pihak ketiga dengan aplikasi web, pemrosesan pembayaran, dan penggunaan layanan web seperti Google Maps atau Twitter. API programming juga menjadi penting dalam pengembangan aplikasi berbasis cloud dan Internet of Things (IoT), di mana penggunaan API dapat memudahkan integrasi dan interaksi antara berbagai perangkat dan layanan.



Gambar 1. RestAPI Model

API programming melibatkan interaksi antara client, REST API application, dan database di server. Berikut adalah penjelasan mengenai ketiga elemen tersebut:

Client

Client adalah aplikasi atau perangkat lunak yang mengakses API untuk melakukan operasi tertentu, seperti mengambil data atau melakukan aksi pada sumber daya di REST API application. Client biasanya berinteraksi dengan API melalui permintaan HTTP, seperti permintaan GET, POST, PUT, atau DELETE.

REST API Application

REST API application adalah perangkat lunak atau aplikasi yang menyediakan API yang dapat diakses oleh client. REST API biasanya menggunakan protokol HTTP untuk mengirimkan permintaan dan menerima respons dari client. REST API mengikuti konvensi tertentu dalam menentukan bagaimana sumber daya diidentifikasi dan diakses melalui API, seperti URI, metode HTTP, dan jenis konten yang dikirimkan.

Database di Server

Database di server menyimpan data dan informasi yang diakses melalui REST API. Database menyediakan antarmuka untuk mengakses dan memanipulasi data yang disimpan di dalamnya. REST API application biasanya menggunakan API database untuk mengakses data yang dibutuhkan oleh client.

Dalam hubungan antara client, REST API application, dan database di server, REST API application berfungsi sebagai jembatan antara client dan database di server. REST API application menyediakan antarmuka API yang memungkinkan client mengakses data dan informasi yang disimpan di dalam database di server. Melalui API programming, client dapat berinteraksi dengan REST API application dan database di server, memungkinkan mereka untuk memperoleh data dan melakukan tindakan yang diperlukan pada data tersebut.

Implementasi Aplikasi REST-API menggunakan PHP dengan Database MySQL

1. Mempersiapkan Database dan tabel di MySQL:

```
C:\xampp\mysql\bin>mysql -u root -p
Enter password:
```

```
MariaDB [(none)]> create database mydb;
Query OK, 1 row affected (0.126 sec)
```

```
MariaDB [(none)]> use mydb;
Database changed
```

```

MariaDB [mydb]> create table mahasiswa(
  -> id int auto_increment primary key not null,
  -> nim varchar(10) unique not null,
  -> nama varchar(20) not null,
  -> jk enum('L','P'),
  -> prodi enum('TIF','IND','PET'));

```

Query OK, 0 rows affected (0.337 sec)

```

MariaDB [mydb]> insert into
mahasiswa(nim,nama,jk,prodi)values('1001','Fikri','L','TIF');
Query OK, 1 row affected (0.267 sec)

```

MariaDB [mydb]>

2. Membuat Aplikasi REST-API:

Beberapa file yang akan dibuat adalah sebagai berikut:

- a. mahasiswa_api.php
- b. Mahasiswa.php
- c. Database.php

Dengan struktur folder sebagai berikut:

```

mhs_api
|__api
|   mahasiswa_api.php
|__class
|   Mahasiswa.php
|__db
    database.php

```

Nama file: database.php

```

<?php
class MySQLDatabase
{
    private $conn;
    private $host = 'localhost';
    private $user = 'root';
    private $password = '';
    private $database = 'mydb';

    public function __construct()
    {
        $this->conn = mysqli_connect($this->host, $this->user,
        $this->password, $this->database);
    }
}

```

```

    public function query($sql)
    {
        return mysqli_query($this->conn, $sql);
    }

    public function fetch_all($result)
    {
        return mysqli_fetch_assoc($result, MYSQLI_ASSOC);
    }

    public function insert_id()
    {
        return mysqli_insert_id($this->conn);
    }

    public function affected_rows()
    {
        return mysqli_affected_rows($this->conn);
    }

    public function escape_string($string)
    {
        return mysqli_real_escape_string($this->conn, $string);
    }

    public function close()
    {
        mysqli_close($this->conn);
    }
}

```

Nama file: mahasiswa_api.php

```

<?php
require_once '../db/database.php';
require_once '../class/Mahasiswa.php';
$db = new MySQLDatabase();
$mahasiswa = new Mahasiswa($db);
$id=0;
$nim=0;
// Check the HTTP request method
$method = $_SERVER['REQUEST_METHOD'];
// Handle the different HTTP methods
switch ($method) {
    case 'GET':
        if(isset($_GET['id'])){
            $id = $_GET['id'];
        }

```

```

    if(isset($_GET['nim'])){
        $nim = $_GET['nim'];
    }
    if($id>0){
        $result = $mahasiswa->get_by_id($id);
    }elseif($nim>0){
        $result = $mahasiswa->get_by_nim($nim);
    } else {
        $result = $mahasiswa->get_all();
    }

    $val = array();
    while ($row = $result->fetch_assoc()) {
        $val[] = $row;
    }

    header('Content-Type: application/json');
    echo json_encode($val);
    break;

case 'POST':
    // Add a new mahasiswa
    $mahasiswa->nim = $_POST['nim'];
    $mahasiswa->nama = $_POST['nama'];
    $mahasiswa->jk = $_POST['jk'];
    $mahasiswa->prodi = $_POST['prodi'];

    $mahasiswa->insert();
    $a = $db->affected_rows();
    if($a>0){
        $data['status']='success';
        $data['message']='Data Mahasiswa created successfully.';
    } else {
        $data['status']='failed';
        $data['message']='Data Mahasiswa not created.';
    }
    header('Content-Type: application/json');
    echo json_encode($data);
    break;
case 'PUT':
    // Update an existing data
    $_PUT = [];
    if(isset($_GET['id'])){
        $id = $_GET['id'];
    }
    if(isset($_GET['nim'])){
        $nim = $_GET['nim'];
    }

```

```

parse_str(file_get_contents("php://input"), $_PUT);
$mahasiswa->nim = $_PUT['nim'];
$mahasiswa->nama = $_PUT['nama'];
$mahasiswa->jk = $_PUT['jk'];
$mahasiswa->prodi = $_PUT['prodi'];
if($id>0){
    $mahasiswa->update($id);
}elseif($nim<>""){
    $mahasiswa->update_by_nim($nim);
} else {

}

$a = $db->affected_rows();
if($a>0){
    $data['status']='success';
    $data['message']='Data Mahasiswa updated successfully.';
} else {
    $data['status']='failed';
    $data['message']='Data Mahasiswa update failed.';
}
header('Content-Type: application/json');
echo json_encode($data);
break;
case 'DELETE':
    // Delete a user
    if(isset($_GET['id'])){
        $id = $_GET['id'];
    }
    if(isset($_GET['nim'])){
        $nim = $_GET['nim'];
    }
    if($id>0){
        $mahasiswa->delete($id);
    }elseif($nim>0){
        $mahasiswa->delete_by_nim($nim);
    } else {

}

$a = $db->affected_rows();
if($a>0){
    $data['status']='success';
    $data['message']='Data Mahasiswa deleted successfully.';
} else {
    $data['status']='failed';
    $data['message']='Data Mahasiswa delete failed.';
}

```

```

        header('Content-Type: application/json');
        echo json_encode($data);
        break;
    default:
        header("HTTP/1.0 405 Method Not Allowed");
        break;
    }
$db->close()
?>

```

Nama file : Mahasiswa.php

```

<?php
require_once '../db/database.php';
class Mahasiswa
{
    private $db;
    private $table = 'mahasiswa';
    public $nim = "";
    public $nama = "";
    public $jk = "";
    public $prodi = "";
    public function __construct(MySQLDatabase $db)
    {
        $this->db = $db;
    }
    public function get_all()
    {
        $query = "SELECT * FROM $this->table";
        $result_set = $this->db->query($query);
        return $result_set;
    }
    public function get_by_id(int $id)
    {
        $query = "SELECT * FROM $this->table WHERE id = $id";
        $result_set = $this->db->query($query);
        return $result_set;
    }
    public function get_by_nim(int $nim)
    {
        $query = "SELECT * FROM $this->table WHERE nim = $nim";
        $result_set = $this->db->query($query);
        return $result_set;
    }
}

```

```

public function insert(): int
{
    $query = "INSERT INTO $this->table (`nim`,`nama`,`jk`,`prodi`)
VALUES ('$this->nim','$this->nama','$this->jk','$this->prodi')";
    $this->db->query($query);
    return $this->db->insert_id();
}

public function update(int $id): int
{
    $query = "UPDATE $this->table SET nim = '$this->nim', nama =
'$this->nama', jk = '$this->jk', prodi = '$this->prodi'
WHERE id = $id";
    $this->db->query($query);
    return $this->db->affected_rows();
}

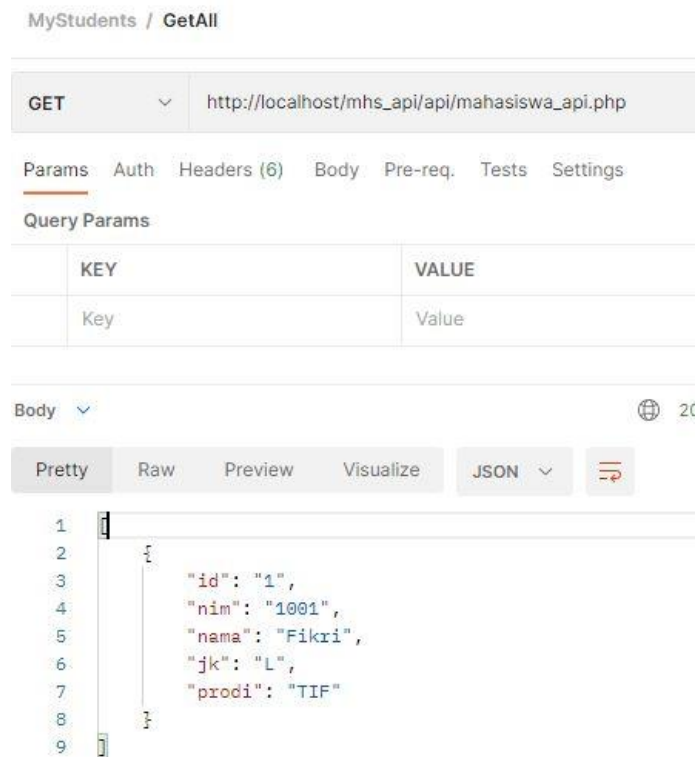
public function update_by_nim($nim): int
{
    $query = "UPDATE $this->table SET nim = '$this->nim',
nama = '$this->nama', jk = '$this->jk', prodi = '$this->prodi'
WHERE nim = $nim";
    $this->db->query($query);
    return $this->db->affected_rows();
}

public function delete(int $id): int
{
    $query = "DELETE FROM $this->table WHERE id = $id";
    $this->db->query($query);
    return $this->db->affected_rows();
}

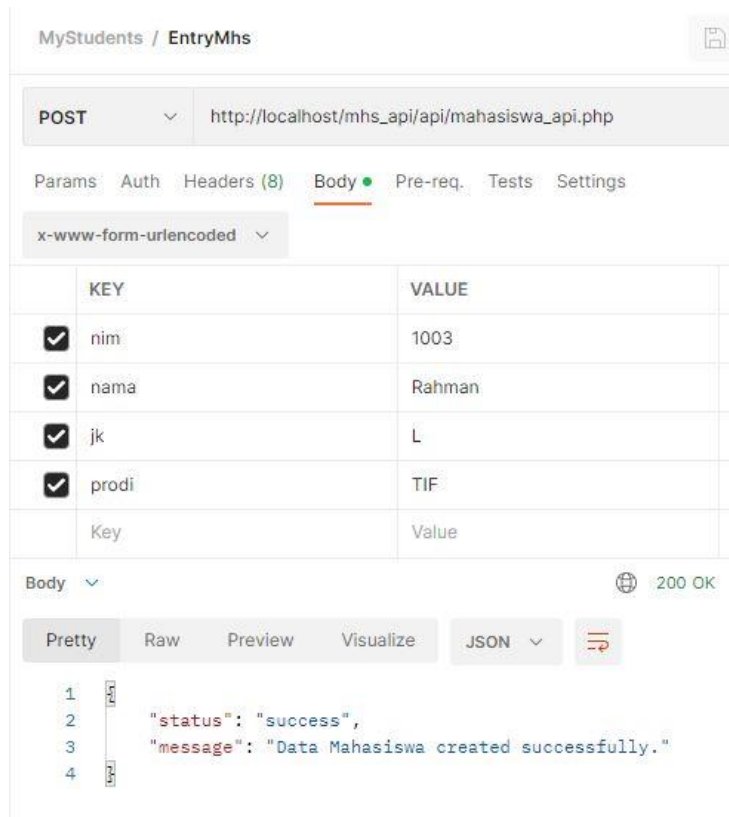
public function delete_by_nim($nim): int
{
    $query = "DELETE FROM $this->table WHERE nim = $nim";
    $this->db->query($query);
    return $this->db->affected_rows();
}
}
?>

```


Test API Access menggunakan Postman



Gambar t. Get All Data



Gambar c. Entry Data

MyStudents / UpdateMhs

PUT http://localhost/mhs_api/api/mahasiswa_api.php?nim=1003

Params Auth Headers (8) **Body** Pre-req. Tests Settings

x-www-form-urlencoded

	KEY	VALUE
<input checked="" type="checkbox"/>	nim	1003
<input checked="" type="checkbox"/>	nama	Sukarsa
<input checked="" type="checkbox"/>	jk	P
<input checked="" type="checkbox"/>	prodi	IND
	Key	Value

Body 200 OK

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "success",
3   "message": "Data Mahasiswa updated successfully."
4 }
```

Gambar x. Update Data

MyStudents / DeleteMhs

DELETE http://localhost/mhs_api/api/mahasiswa_api.php?nim=1003

Params Auth Headers (6) Body Pre-req. Tests Settings

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	nim	1003
	Key	Value

Body 200 OK

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "success",
3   "message": "Data Mahasiswa deleted successfully."
4 }
```

Gambar y. Delete Data