**Modul Praktikum**

**API**

**Dirancang Oleh :**

**Jendra Sesoca S.T, M.T**

****

**JURUSAN TEKNOLOGI INFORMASI**

**FAKULTAS VOKASI**

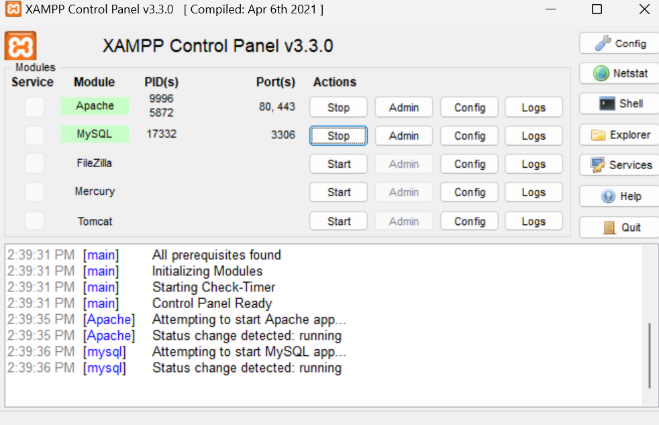
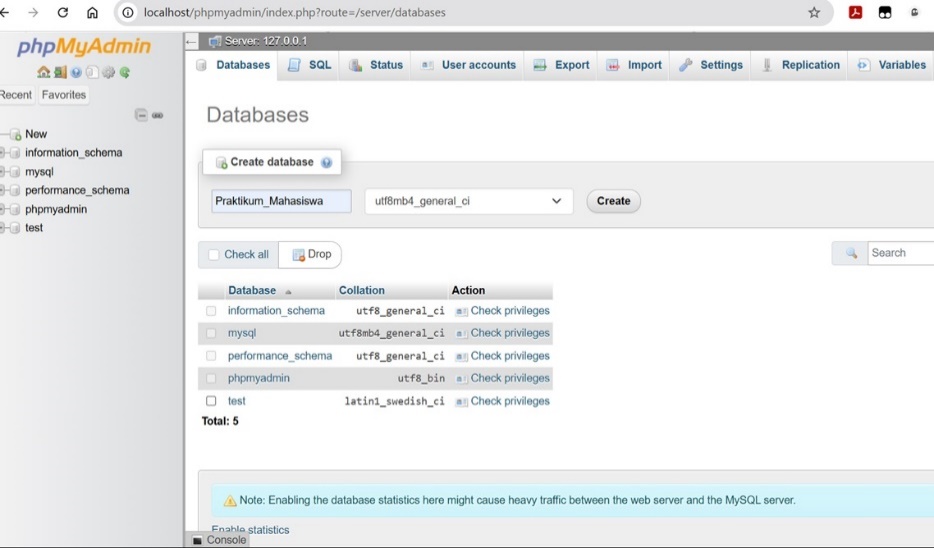
**UNIVERSITAS BRAWIJAYA**

**2025**

Praktikum API

1. Pengantar

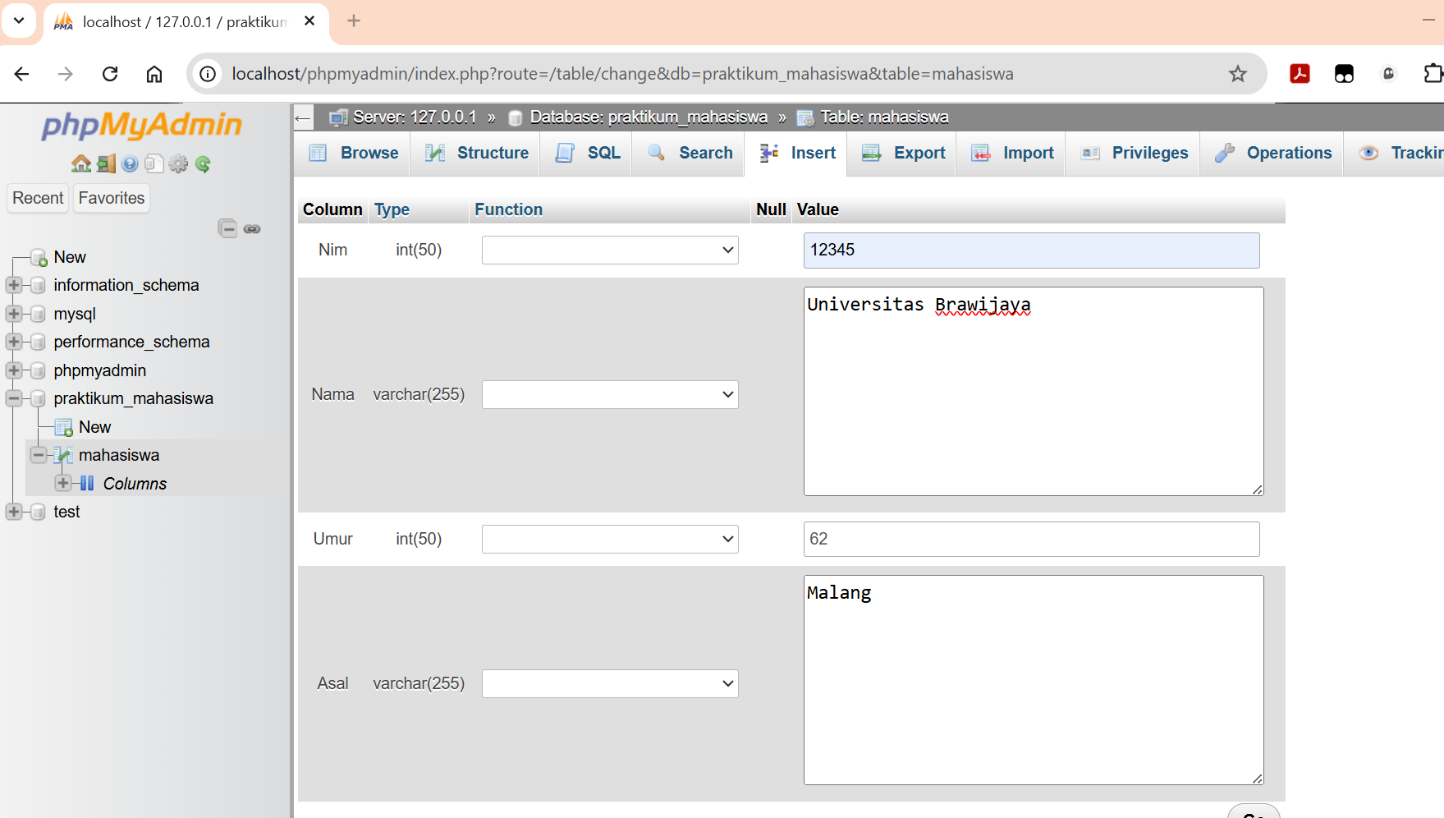
Pada praktikum ini akan mempraktekan cara membuat API CRUD sederhana, dengan perintah GET, POST, DELETE dan PUT.

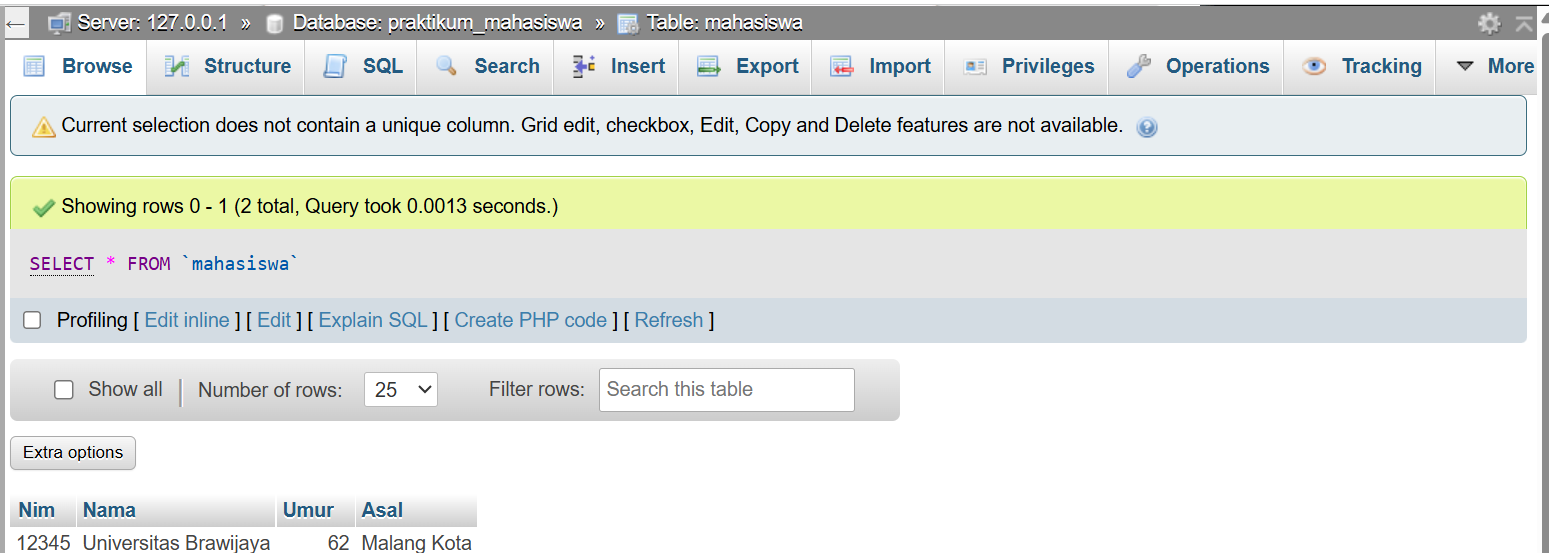
1. Praktikum
2. Pastikan pada perangkat telah terinstall aplikasi XAMPP, VISUAL CODE dan POSTMAN.
3. Buka aplikasi XAMPP dan aktifkan APACHE dan SQL.
4. Setelah itu buka web browser dan ketikan localhost/phpMyAdmin/.
5. Buatlah database bernamakan Praktikum\_Mahasiswa.
6. Buatlah Table pada Praktikum\_Mahasiswa bernamakan Mahasiswa berjumlah 4 kolom.

Isikan Nim(INT), Nama(Varchar), Umur(INT), Alamat(Varchar).

Sebuah gambar berisi teks, cuplikan layar, nomor, Paralel

AI-generated content may be incorrect.

1. Setelah table terbuat isi data table tersebut dengan klik insert lalu isikan sesuai dengan 5 teman sekelas.
2. Bila sudah diisi maka tampilan pada browse akan seperti ini:



1. Sebuah gambar berisi teks, cuplikan layar, software, tampilan

   AI-generated content may be incorrect.Selanjutnya buatlah folder pada penyimpanan aplikasi Xampp di dalam folder htdocs Bernama folder praktikum. Selanjutnya bukalah aplikasi Visualcode dan buat file praktikum mahasiswa.php, dan simpan pada folder praktikum.
2. Selanjutnya edit file praktikum mahasiswa.php dengan aplikasi visualcode. Lalu isi dengan baris perintah GET berikut :

<?php

header('Content-type: application/json; charset=utf8');

$koneksi = mysqli\_connect( "localhost","root","","Praktikum\_mahasiswa");

if($\_SERVER['REQUEST\_METHOD'] === 'GET'){

$sql = "select \* from mahasiswa";

$query = mysqli\_query($koneksi, $sql);

$array\_data = array();

while($data = mysqli\_fetch\_assoc($query)){

$array\_data[] = $data;

}

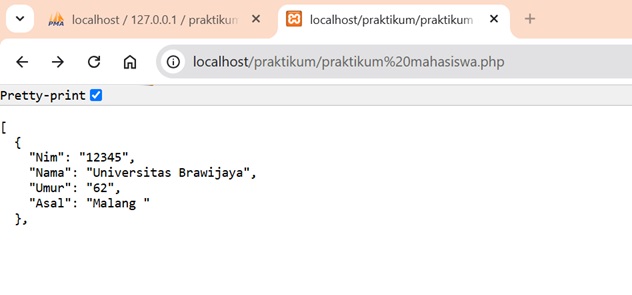
echo json\_encode($array\_data);

}

?>

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

AI-generated content may be incorrect.Setelah memasukan perintah klik save(ctrl+S)

1. Setelahnya buka browser lagi lalu ketik localhost/praktikum/, lalu klik praktikum mahasiswa.php. maka akan tampil tampilan database dalam bentuk Json.
2. Setelahnya masuk Kembali ke Visual Code hapus ?> pada baris terakhir dan isi baris perintah POST berikut:

else if ($\_SERVER['REQUEST\_METHOD'] === 'POST'){

$Nim = $\_POST['Nim'];

$Nama = $\_POST['Nama'];

$Umur = $\_POST['Umur'];

$Alamat =$\_POST[‘Alamat’];

$sql = "INSERT INTO mahasiswa (Nim, Nama, Umur, Alamat) VALUES('$Nim', '$Nama', $Umur, '$Alamat')";

$cek = mysqli\_query($koneksi, $sql);

if($cek){

$data = [

'status' => "berhasil"

];

echo json\_encode ([$data]);

}else{

$data = [

'status' => "gagal"

];

echo json\_encode ([$data]);

}

}



1. Setelahnya masukan perintah DELETE seperti berikut:

else if($\_SERVER['REQUEST\_METHOD'] === 'DELETE'){

$Nim = $\_GET['Nim'];

$sql = "DELETE FROM mahasiswa WHERE Nim='$Nim'";

$cek = mysqli\_query($koneksi, $sql);

if($cek){

$data = [

'status' => "berhasil"

];

echo json\_encode ([$data]);

}else{

$data = [

'status' => "gagal"

];

echo json\_encode ([$data]);

}

}



1. Langkah Berikutnya masukan perintah PUT seperti di bawah ini:

else if($\_SERVER['REQUEST\_METHOD'] === 'PUT'){

$Nim = $\_GET['Nim'];

$Nama = $\_GET['Nama'];

$Umur = $\_GET['Umur'];

$Alamat = $\_GET['Alamat'];

$sql = "UPDATE mahasiswa SET Nim='$Nim', Nama='$Nama', Umur='$Umur', Alamat=$Alamat WHERE Nim='$Nim'";

$cek = mysqli\_query($koneksi, $sql);

if($cek){

$data = [

'status' => "berhasil"

];

echo json\_encode ([$data]);

}else{

$data = [

'status' => "gagal"

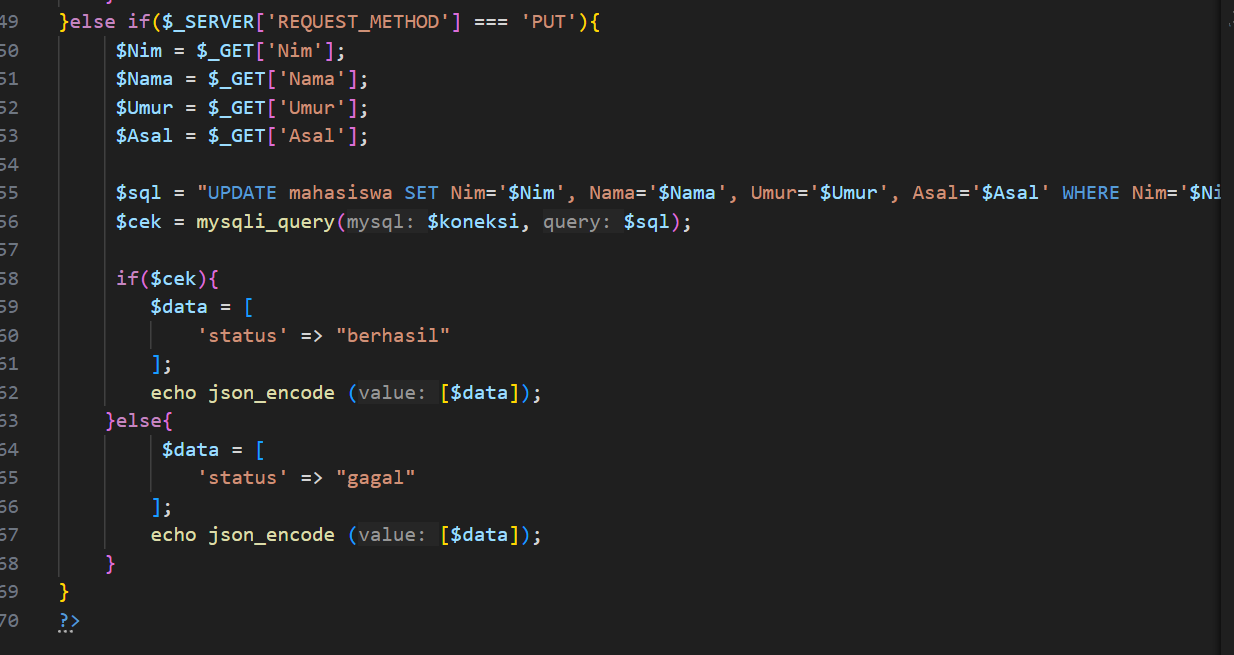
];

echo json\_encode ([$data]);

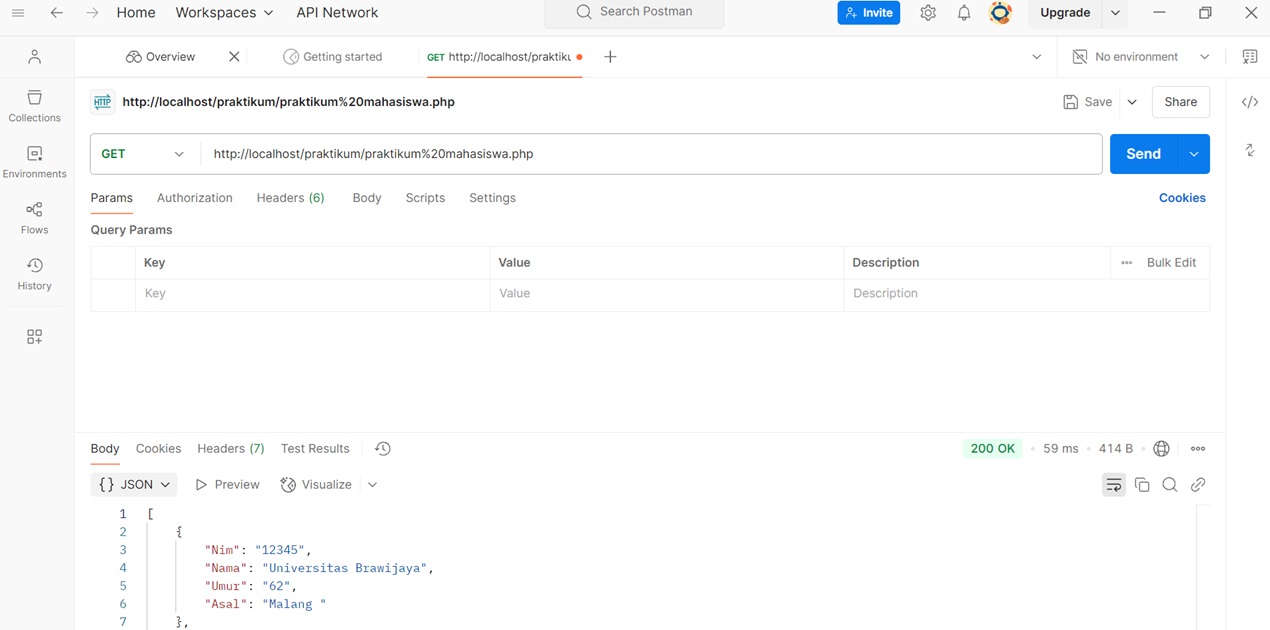
}

}

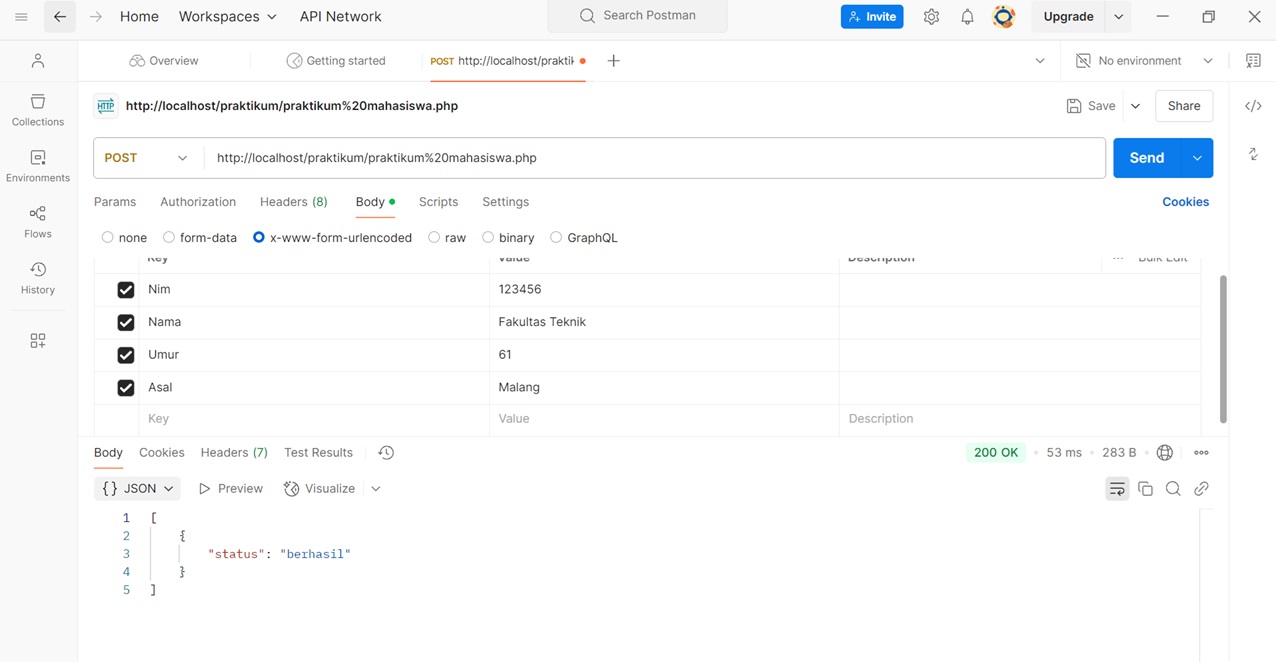
?>



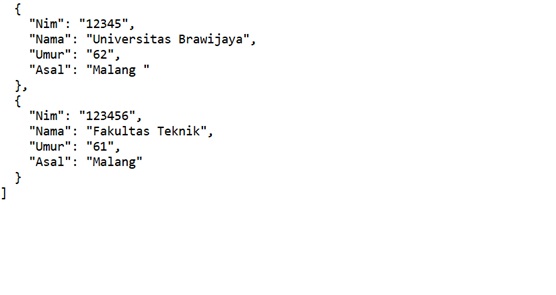
1. Berikutnya buka aplikasi Postman, pastikan terhubung dengan internet, lalu isikan localhost/Praktikum/Praktikum%20mahasiswa.php, atau copy link address localhost di browser ke Postman. Lalu pilih Get dan kirim. Maka database akan ditampilkan di aplikasi Postman.

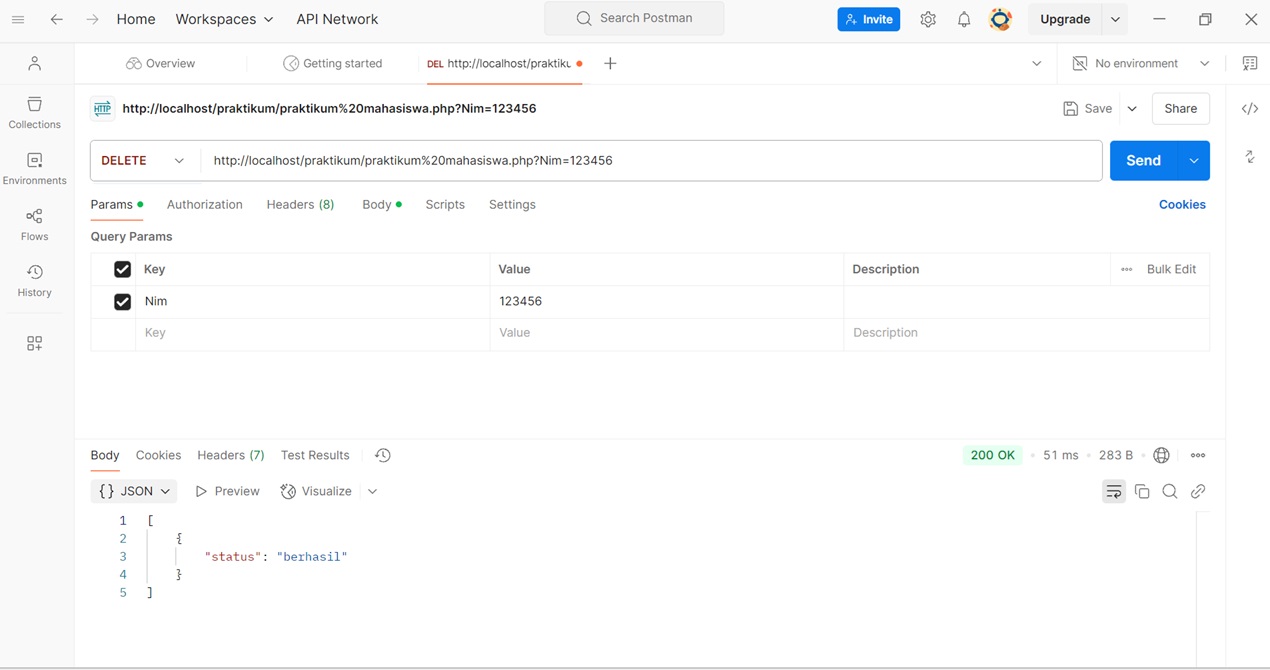
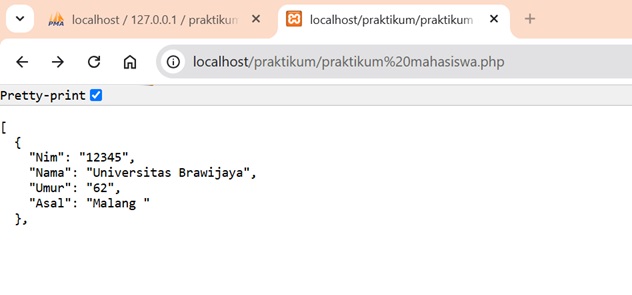


1. Selanjutnya ganti perintah ke POST=>Body=>x-www-form-urlencoded. Lalu tambahakan 1 data teman kelas.

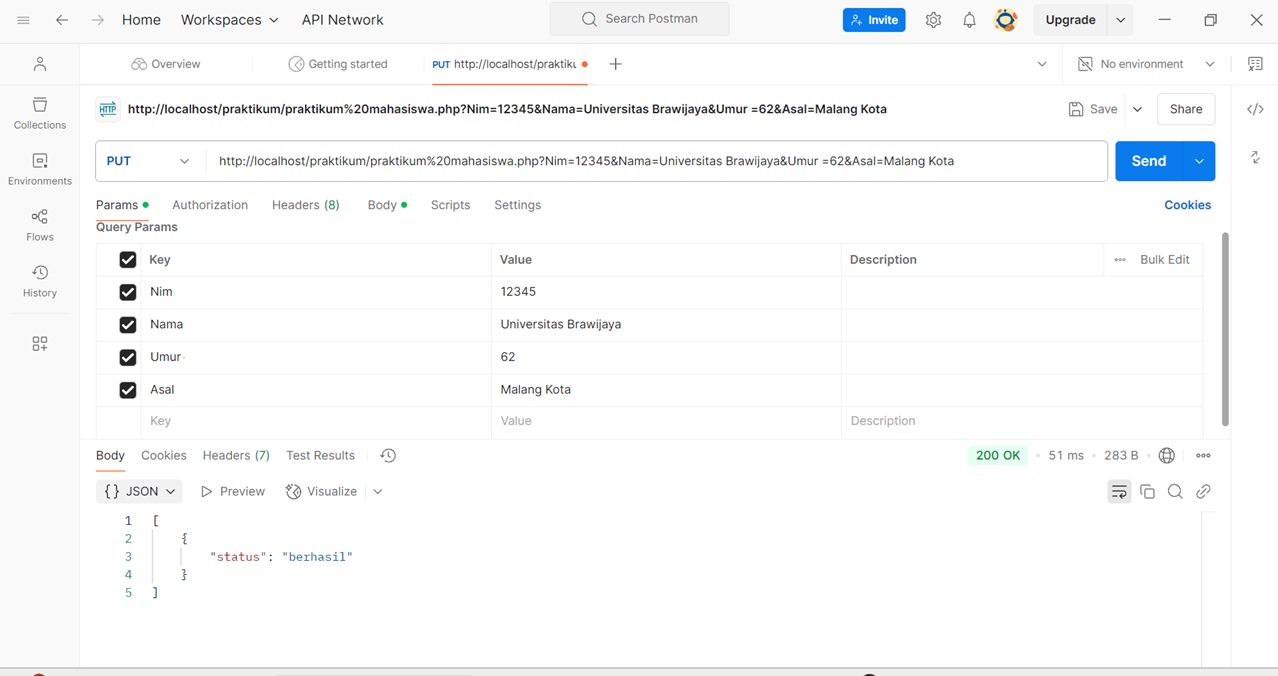


1. Lalu buka browser dan refresh localhost lihat apakah penambahan database berhasil.

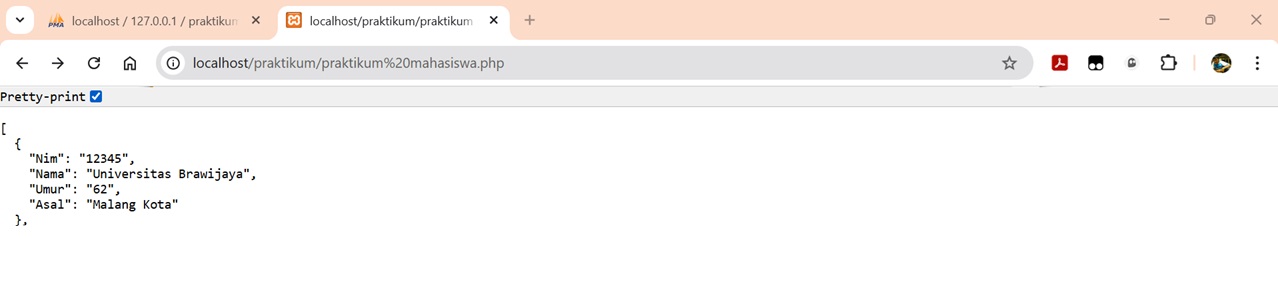


1. Selanjutnya buka Kembali aplikasi POSTMAN dan Ganti ke perintah DELETE=>Params, pada key ketikan salah Satu Nim yang akan dihapus klik kirim.
2. Lalu buka Kembali browser dan refresh lihat Kembali apakah data yang dihapus berhasil.

1. Selanjutnya lakukan perintah Put, klik di params lalu masukan data dengan parameter yang dibuat, Nim tidak bisa dirubah, cobalah rubah Asal.



1. Buka kembali browser dan refresh apakah pergantian data berhasil.



**Tugas :**

Buatlah video youtube sesuai dengan Langkah-langkah diatas.

**Lampiran :**

**1. Tipe Data (Type)**

Menentukan jenis data yang dapat disimpan dalam kolom.

Contoh:

- INT: Menyimpan angka bulat. Contoh: Umur, ID

- VARCHAR(100): Menyimpan teks pendek, hingga 100 karakter. Contoh: Nama

- TEXT: Menyimpan teks panjang. Contoh: Deskripsi

- DATE: Menyimpan tanggal. Contoh: Tanggal Lahir

- FLOAT: Menyimpan angka desimal. Contoh: Nilai, Harga

**2. Default**

Menentukan nilai awal jika tidak ada nilai yang dimasukkan.

Contoh:

- Kolom 'Umur' dengan default 18: jika user tidak mengisi umur, maka akan otomatis menjadi 18.

**3. Collation**

Mengatur cara perbandingan dan pengurutan teks.

Contoh:

- utf8\_general\_ci: Tidak membedakan huruf besar dan kecil ('a' = 'A')

- utf8\_bin: Membandingkan berdasarkan byte, membedakan huruf besar dan kecil ('a' ≠ 'A')

**4. Attributes**

Menambahkan aturan tambahan pada kolom.

Contoh:

- UNSIGNED: Hanya menerima angka positif (biasanya pada kolom bertipe INT)

- ON UPDATE CURRENT\_TIMESTAMP: Waktu akan otomatis diperbarui saat baris diubah (pada kolom tipe DATETIME)

**5. Null**

Menentukan apakah kolom boleh dikosongkan.

Contoh:

- NULL: Kolom boleh dikosongkan

- NOT NULL: Kolom wajib diisi saat data dimasukkan

**6. Index**

Meningkatkan kinerja pencarian dan memastikan keunikan data.

Contoh:

- PRIMARY KEY: Penanda unik utama, misal 'ID'

- UNIQUE: Menjamin nilai dalam kolom tidak ada yang sama, misal 'Email'

- INDEX: Mempercepat pencarian tanpa membatasi keunikan

- FULLTEXT: Untuk pencarian teks dalam kolom teks panjang

**7. Auto Increment**

Membuat nilai kolom bertambah otomatis setiap data baru ditambahkan.

Contoh:

- Kolom 'ID' dengan AUTO\_INCREMENT akan menghasilkan nilai 1, 2, 3, dst secara otomatis tanpa input manual.

**8. A\_I (Auto Increment)**

Fitur ini membuat nilai kolom bertambah otomatis setiap kali data baru ditambahkan.

Contoh:

- Kolom 'id' dengan A\_I aktif akan menghasilkan nilai 1, 2, 3, ... secara otomatis tanpa perlu diisi manual.

**9. Comments**

Komentar atau catatan untuk kolom tersebut, tidak mempengaruhi data tapi berguna untuk dokumentasi.

Contoh:

- Kolom 'email' bisa diberi komentar 'harus unik dan valid'.

Komentar ini hanya dapat dilihat dalam struktur database.

**10. Virtuality (Generated Columns)**

Menentukan apakah kolom bersifat:

- STORED: Nilainya disimpan dalam database

- VIRTUAL: Nilainya dihitung saat dibutuhkan dan tidak disimpan

Contoh:

- Kolom 'total' sebagai hasil dari 'jumlah \* harga' bisa dibuat virtual untuk menghemat penyimpanan.

**11. Move Column**

Digunakan untuk mengubah urutan kolom dalam tabel.

Contoh:

- Memindahkan kolom 'email' agar berada setelah kolom 'nama'.

**12. Media Type**

Menentukan tipe media yang disimpan di kolom tersebut.

Contoh:

- Jika kolom menyimpan gambar atau file biner, pilih media type seperti 'image/jpeg'.

**13. Browser Display Transformation**

Menentukan bagaimana data ditampilkan di browser saat melihat isi tabel di phpMyAdmin.

Contoh:

- Untuk kolom gambar (BLOB), dapat ditampilkan sebagai gambar di browser jika transformasi diatur dengan benar.

**14. Browser Display Transformation Options**

Opsi tambahan untuk menyesuaikan tampilan browser.

Contoh:

- Menentukan ukuran maksimum gambar saat ditampilkan.

**15. Input Transformation**

Menentukan bagaimana data dimasukkan atau diproses sebelum disimpan.

Contoh:

- Data input bisa diformat otomatis (misal angka diformat sebagai mata uang).

**16. Input Transformation Options**

Opsi tambahan untuk mengatur transformasi input.

Contoh:

- Menambahkan prefiks atau sufiks otomatis seperti 'Rp' pada input harga.

**17. Table Comments**

Table Comments adalah deskripsi atau catatan tambahan yang ditambahkan ke sebuah tabel di MySQL. Komentar ini membantu pengguna atau developer lain memahami tujuan dan isi tabel tersebut. Komentar ini tidak memengaruhi fungsi tabel secara langsung, tetapi sangat berguna untuk dokumentasi.

**18. Collation**

Collation adalah aturan yang digunakan untuk membandingkan dan mengurutkan string dalam database. Collation menentukan bagaimana karakter dibandingkan (case-sensitive atau tidak) dan bagaimana urutan karakter dilakukan. Contoh collation: utf8\_general\_ci, utf8mb4\_unicode\_ci. Pemilihan collation sangat penting dalam pencarian data berbasis teks.

**19. Storage Engine**

Storage Engine adalah mekanisme penyimpanan data di MySQL. Setiap storage engine memiliki kelebihan dan kekurangan. Contoh storage engine yang umum digunakan:   
- InnoDB: Mendukung transaksi, kunci asing (foreign key), dan crash recovery.  
- MyISAM: Cepat untuk operasi baca tetapi tidak mendukung transaksi.  
Pengguna dapat memilih storage engine yang paling sesuai berdasarkan kebutuhan aplikasi.

**20. Partition Definition**

Partition Definition adalah cara untuk membagi data dalam sebuah tabel menjadi beberapa bagian lebih kecil (partisi) berdasarkan aturan tertentu. Hal ini dapat meningkatkan performa query dan manajemen data. Setiap partisi diperlakukan seperti sub-tabel internal yang dikelola oleh sistem.

**21. Partition By**

Partition By adalah perintah untuk menentukan aturan atau kriteria bagaimana data dalam tabel akan dipartisi. Jenis-jenis partitioning meliputi:  
- RANGE: Berdasarkan rentang nilai.  
- LIST: Berdasarkan daftar nilai tertentu.  
- HASH: Berdasarkan hasil dari fungsi hash.  
- KEY: Berdasarkan kolom yang dipilih secara otomatis oleh sistem.

**22. Partition pada phpMyAdmin**

Di phpMyAdmin, pengguna dapat mengatur partisi tabel melalui antarmuka grafis. Saat membuat atau mengubah tabel, terdapat tab atau bagian untuk mengatur partisi. Pengguna bisa memilih jenis partitioning dan menentukan kolom serta aturan yang akan digunakan untuk partisi. Namun, fitur ini hanya tersedia jika server MySQL mendukung partitioning dan tabel menggunakan storage engine yang sesuai (seperti InnoDB).