





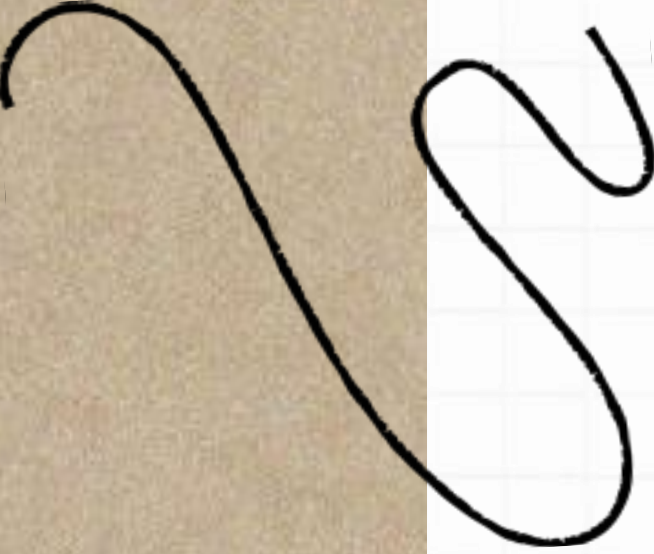

# Pemrograman Web



SANDRA HARO



# Outline materi :

- 
- 
- 
- 
1. Access Modifier (public , private, protected)
  2. Accessor Method (fungsi get dan fungsi set)
  3. Konsep DBMS
  4. Jenis DBMS
  5. Komponen DBMS
  6. Operasional DBMS






# 1. Access Modifier





***Access Modifier*** adalah sebuah konsep yang digunakan untuk mengatur akses dari property (variable) dan method (fungsi) pada sebuah object.



### ***Public***

Akses yang memungkinkan sebuah property dan method bisa diakses dari mana saja, bahkan dari luar class object yang bersangkutan.

### ***Protected***





Akses yang memungkinkan sebuah property dan method masih bisa diakses dari class lain, tetapi masih merupakan class turunan (child class) dari class yang bersangkutan,

### ***Private***


Akses yang diberikan kepada property dan method dengan tujuan agar ia tidak dapat diakses oleh class manapun, kecuali class yang bersangkutan dimana property dan method tersebut di deklarasikan.



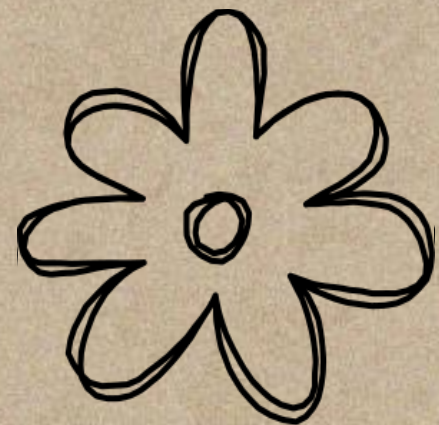
# Fungsi Access Modifier dalam PHP

- 
- 
- 
- 
1. Memberi batasan akses pada property dan method yang ada dalam sebuah class, sehingga kita bisa mengatur dan menentukan aspek apa saja yang dibutuhkan oleh client.
  2. Dapat mengidentifikasi kebutuhan yang jelas untuk sebuah object.
  3. Dapat memberikan kendali kepada kode program untuk menghindari 'bug'.

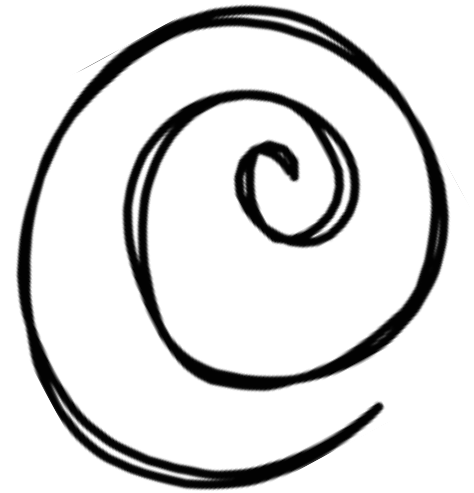




# Implementasi access modifier

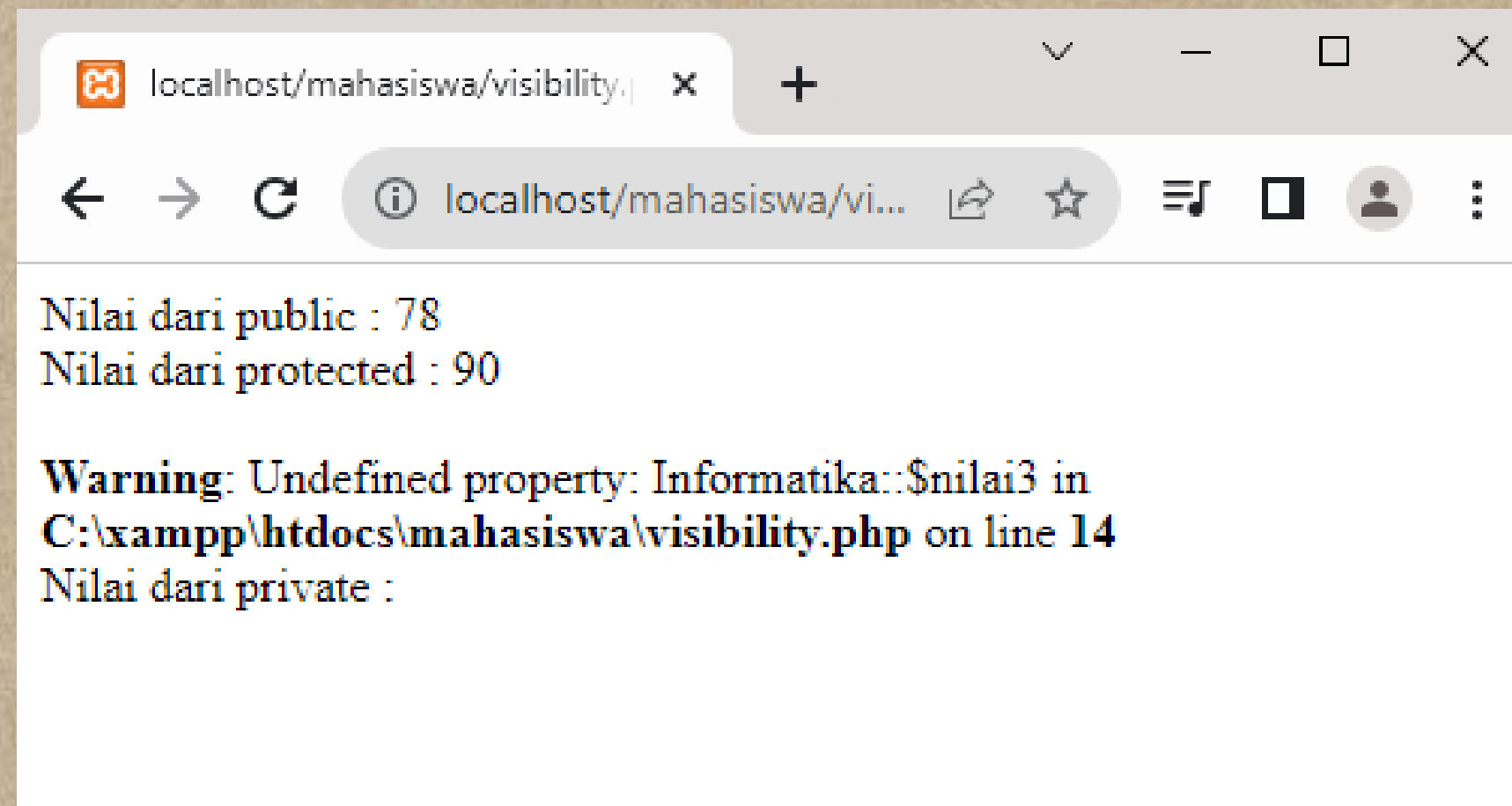


```
<?php
class Mahasiswa {
    public $nilai1=78;
    protected $nilai2=90;
    private $nilai3=65;
}
class Akuntansi extends Mahasiswa {
    public function getNilai2() {
        return $this->nilai2;
    }
}
class Informatika extends Mahasiswa {
    public function getNilai3() {
        return $this->nilai3;
    }
}
$mhs1 = new Akuntansi();
$mhs2 = new Akuntansi();
$mhs3 = new Informatika();
echo "Nilai dari public : " . $mhs1->nilai1;
echo "<br>";
echo "Nilai dari protected : " . $mhs2->getNilai2();
echo "<br>";
echo "Nilai dari private : " . $mhs3->getNilai3();
?>
```








# Output dari program









## 2. Accessor Method

### Fungsi get() dan set()



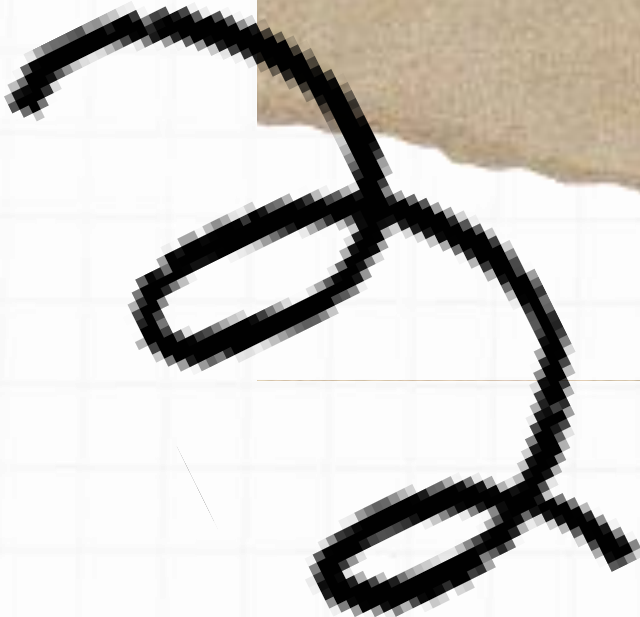
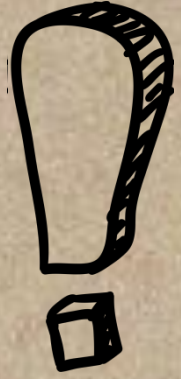


# Accessor Method




Adalah sebuah metode yang digunakan untuk mengambil (get) dan menetapkan(set) nilai dari sebuah variable dan function, untuk kemudian digunakan di class lain.

Selain itu, hampir sama dengan konsep inheritance, metode `get()` dan `set()` juga berfungsi untuk membuat sebuah hubungan atau relasi antara satu class dengan class yang lain dalam program PHP.





## Contoh implementasi fungsi get()

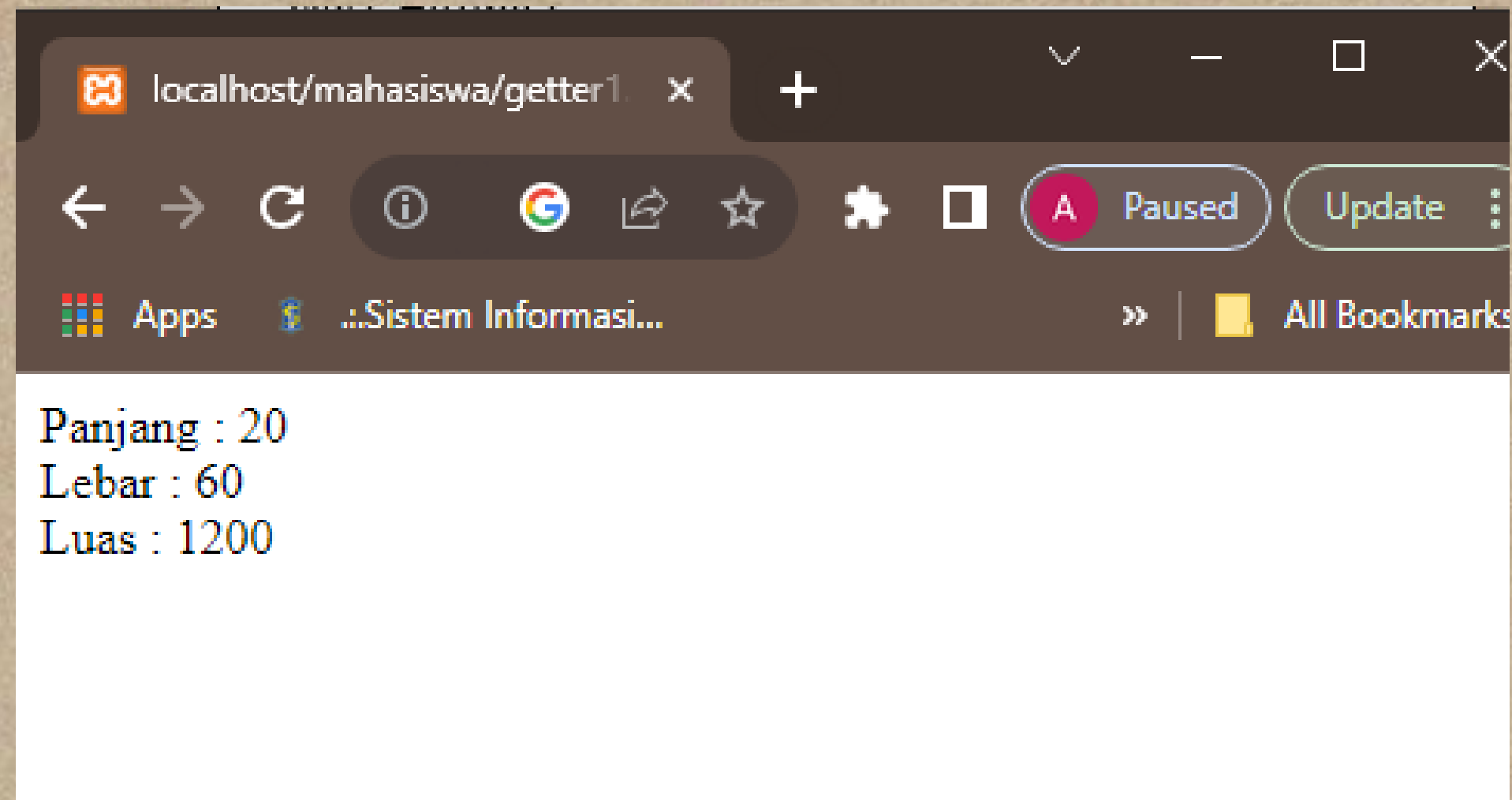


```
<?php
class Hitung {
    public $panjang,$lebar,$luas;

    public function getLuas() {
        $luas = $this->panjang * $this->lebar;
        return $luas;
    }
}
$size = new Hitung();
echo "Panjang : " . $size->panjang=20;
echo "<br>";
echo "Lebar : " . $size->lebar=60;
echo "<br>";
echo "Luas : " . $size->getLuas();
?>
```



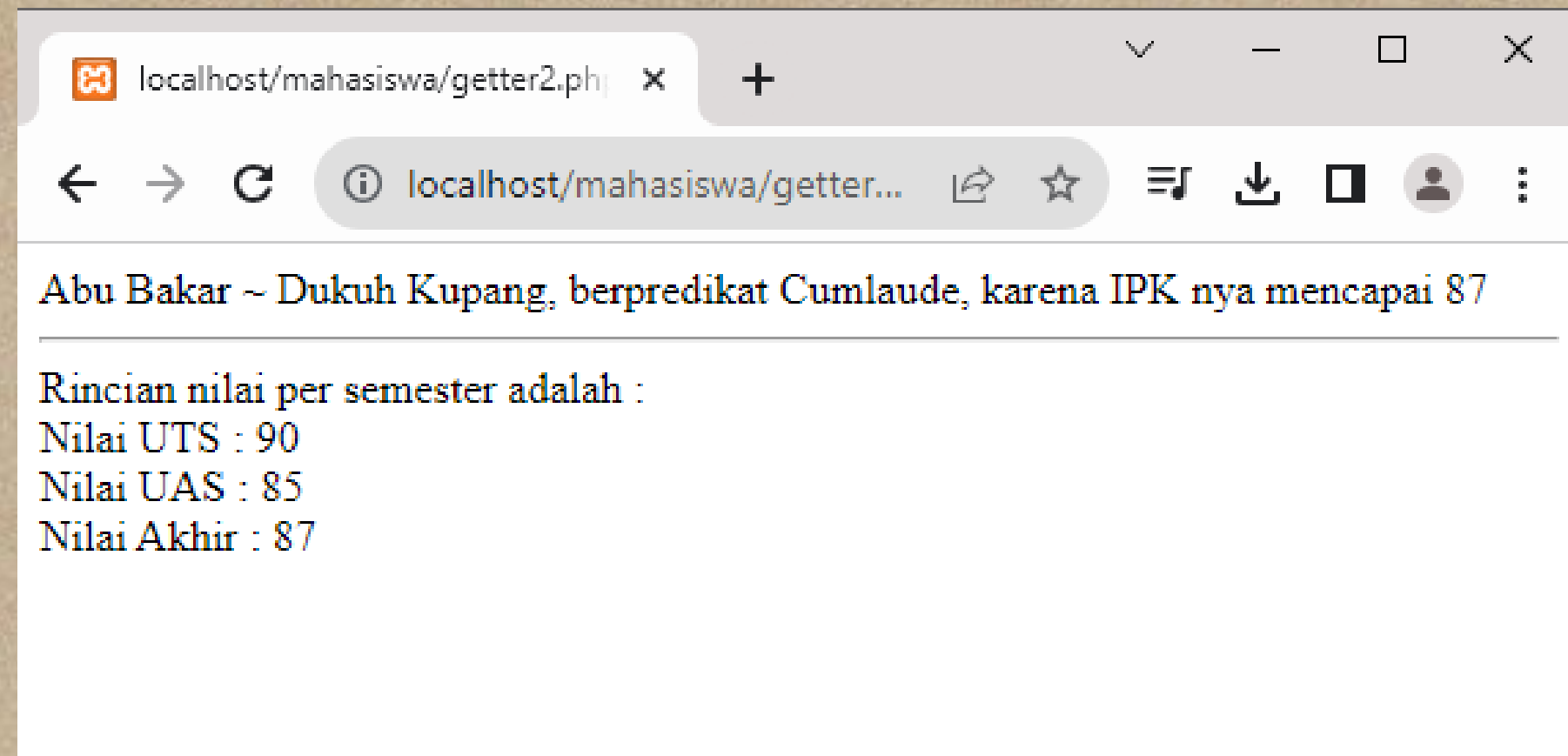
# Output dari program





# Tugas mandiri :

Buatlah fungsi get() untuk menampilkan nilai UTS, UAS,  
dan Nilai Akhir, dimana nilai akhir didapat dari :  
$$(0.4 * \text{UTS}) + (6.0 * \text{UAS})$$





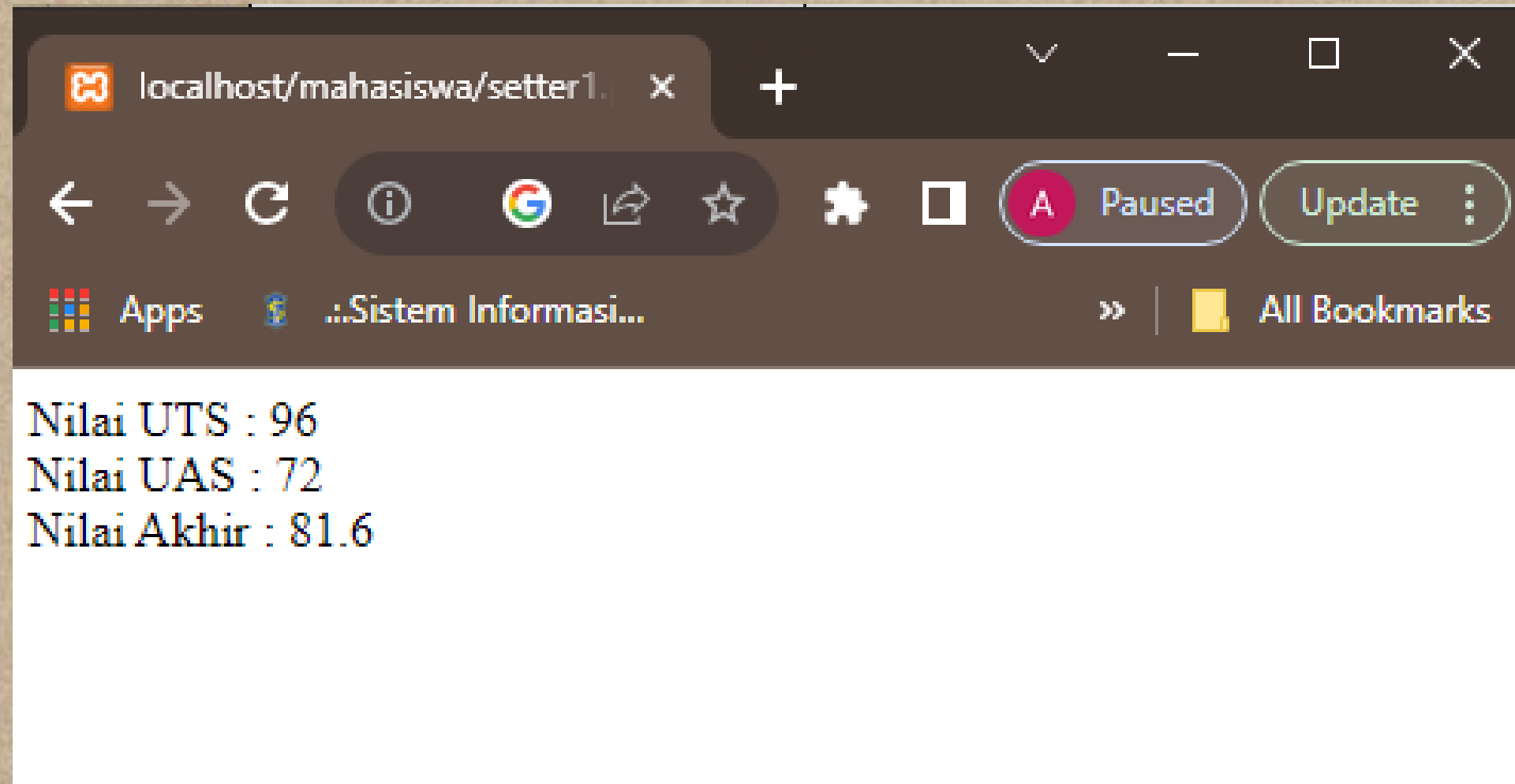
# Contoh implementasi fungsi set()

```
<?php
class KHS {
    public $uts,$uas;

    public function setUts($uts) {
        if(!is_numeric($uts)) {
            throw new exception('Nilai yang dimasukkan harus numerik !!!');
        }
        $this->uts=$uts;
    }
    public function setUas($uas) {
        $this->uas=$uas;
    }
    public function getNakhir() {
        $uts=$this->uts;
        $uas=$this->uas;
        $nakhir = (0.4 * $uts) + (0.6*$uas);
        return $nakhir;
    }
}
$nilai = new KHS();
$nilai->setUts("aaa");
$nilai->setUas(72);
echo "Nilai UTS : " . $nilai->uts;
echo "<br>";
echo "Nilai UAS : " . $nilai->uas;
echo "<br>";
echo "Nilai Akhir : " . $nilai->getNakhir();
?>
```



# Output :







# 3. Konsep DBMS



RINA MUSYAWARAH



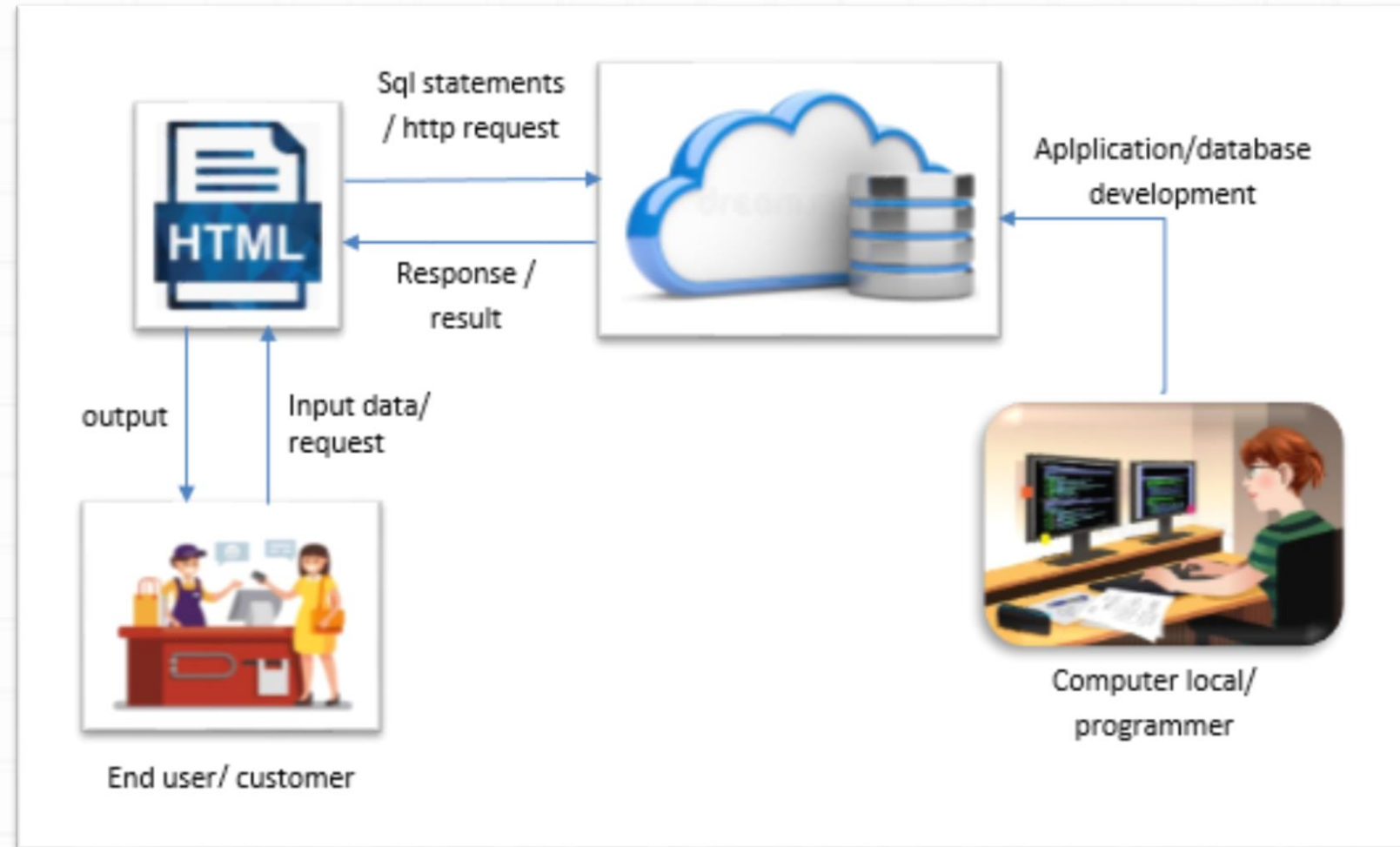
# Apa itu DBMS ?

*Database* atau Basis Data (*Pangkalan data*), adalah sekumpulan data atau informasi mengenai suatu object, yang disimpan secara rapi dan terorganisir dalam sebuah perangkat komputer, sehingga dapat diakses melalui suatu sistem komputer dengan lebih mudah.

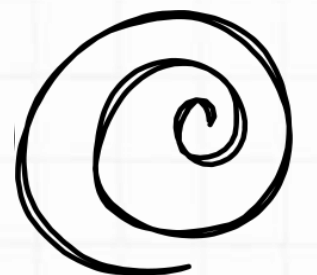
Sedangkan DBMS (Database Management System) adalah suatu sistem atau software yang digunakan untuk mengelola suatu database serta menjalankan setiap perintah atau operasi yang dilakukan oleh banyak pengguna, terhadap setiap object yang ada di dalam database tersebut.



# Tujuan digunakannya DBMS :



1. Kecepatan dan kemudahan dalam mengakses data
2. Multi-user
3. Keamanan data
4. Penghematan biaya perangkat
5. Kontrol data terpusat
6. Memudahkan dalam membuat aplikasi
7. Menghilangkan duplikasi dan inkonsistensi data
8. Menangani data dalam jumlah yang banyak atau besar







## 4. Jenis DBMS



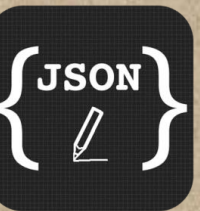
RINA MUSYAWARAH



# Operational Database

*Operational Database* atau biasa disebut dengan database OLTP (*On Line Transaction Processing*), berguna untuk mengelola data yang dinamis secara langsung atau *real-time*. Jenis ini memungkinkan para pengguna dapat melakukan, melihat, dan memodifikasi data. Modifikasi tersebut bisa berupa mengubah, menambah, menghapus data secara langsung melalui perangkat keras yang digunakan.

1. **JSON (*JavaScript Object Notation*)** adalah format *file* yang menggunakan teks untuk pengiriman data. Format ini sangat umum digunakan programmer untuk pertukaran data layaknya berkomunikasi cepat dengan melalui web browser dan web server. Sinkronisasi data bisa dilakukan secara *real-time*.
2. **XML (*Extensible Markup Language*)** adalah bahasa program *markup* yang memiliki aturan untuk memberikan dua kode dokumen berbeda yang bisa dibaca oleh manusia dan dibaca oleh komputer. Melalui XML, akan menghasilkan format data berupa teks yang dapat digunakan untuk merepresentasikan struktur basis data. Selain itu, sinkronisasi data dapat dilakukan secara *real-time* oleh pengguna.





# Database Warehouse

*Database Warehouse* adalah sistem basis data yang biasa digunakan untuk pelaporan dan analisis data. *Database Warehouse* merupakan repositori sentral data yang terpadu dari satu atau lebih sumber yang berbeda.

*Database* tersebut juga menyimpan data terkini dan historis dengan satu tempat yang digunakan untuk membuat laporan analisis.

Data yang tersimpan di *warehouse* awalnya diunggah dari sistem operasi. Data bisa melewati penyimpanan operasional dan memungkinkan untuk pembersihan data. Proses tersebut menjadi operasi tambahannya dan dapat memastikan kualitas data sebelum digunakan di warehouse sebagai pelaporannya.

## ~ Microsoft SQL Server ~

Microsoft SQL Server adalah sistem basis data yang dibangun oleh Microsoft. Sebagai *server database*, sistem ini merupakan produk perangkat lunak yang berfungsi menyimpan dan mengambil data sesuai permintaan aplikasi lainnya. Hal tersebut memungkinkan dapat berjalan baik melalui komputer yang sama atau komputer lainnya melalui jaringan internet.





DB

# Relational Database

*Relational Database* adalah basis data yang perangkat penyimpanannya bisa diletakkan pada perangkat komputer yang sama atau di beberapa perangkat lain yang tersebar melalui jaringan komputer yang saling berhubungan. Banyak sekali perangkat lunak yang menggunakan sistem ini untuk mengatur dan memelihara basis data melalui hubungan setiap data. Umumnya, semua sistem menggunakan *Structured Query Language* (SQL) sebagai bahasa pemrograman untuk pemeliharaan basis data dan *query*.

1. MySQL
2. PostgreSQL
3. MariaDB
4. *Oracle Database*
5. *Interbase*
6. *Firebird*



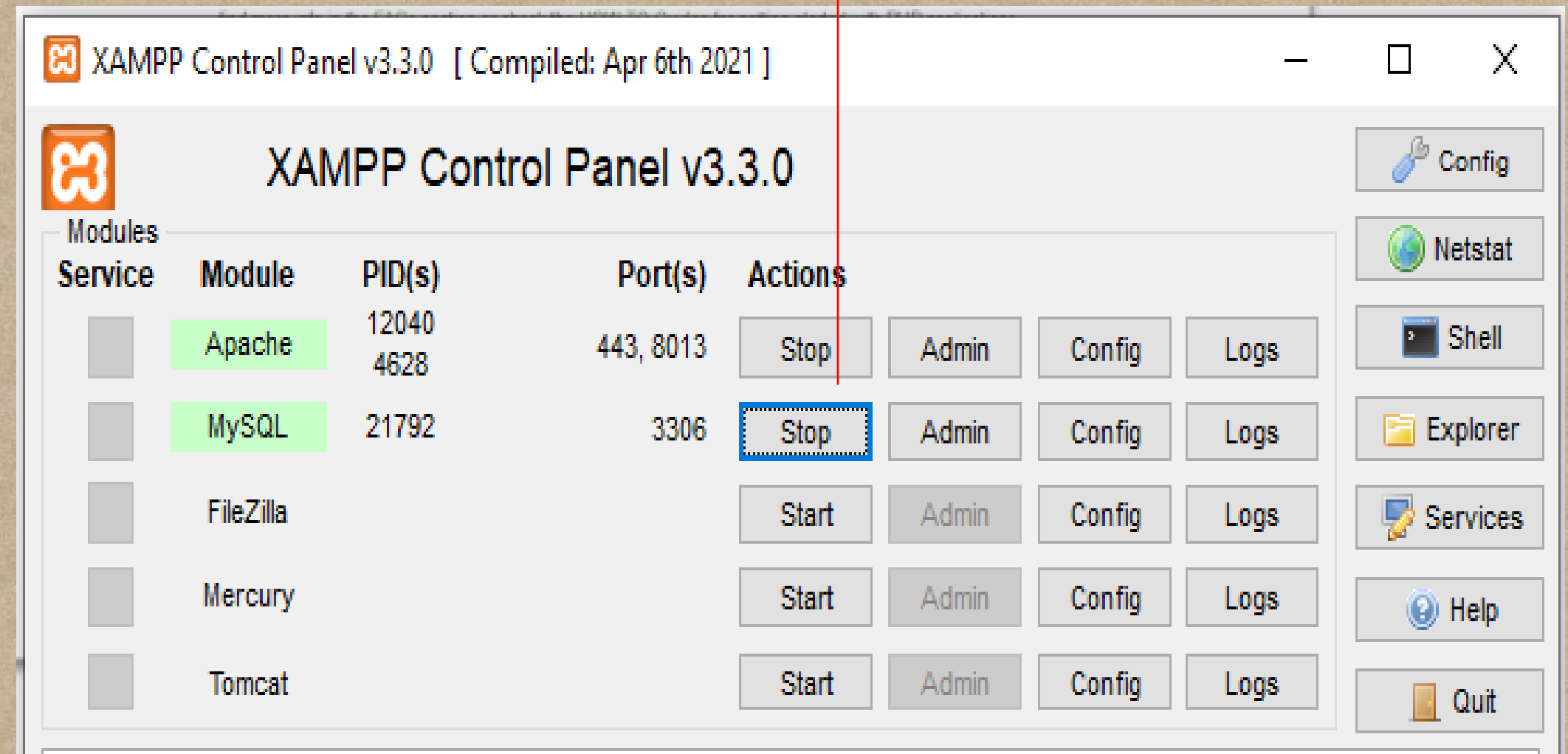
# MySQL



Database yang akan digunakan untuk pembuatan aplikasi berbasis web dengan PHP ini adalah MySQL. Untuk mendapatkan database MySQL versi open source (gratis), kita bisa menggunakan PHPMyAdmin, dimana aplikasi ini otomatis terinstall ketika kita menginstall web server XAMPP.



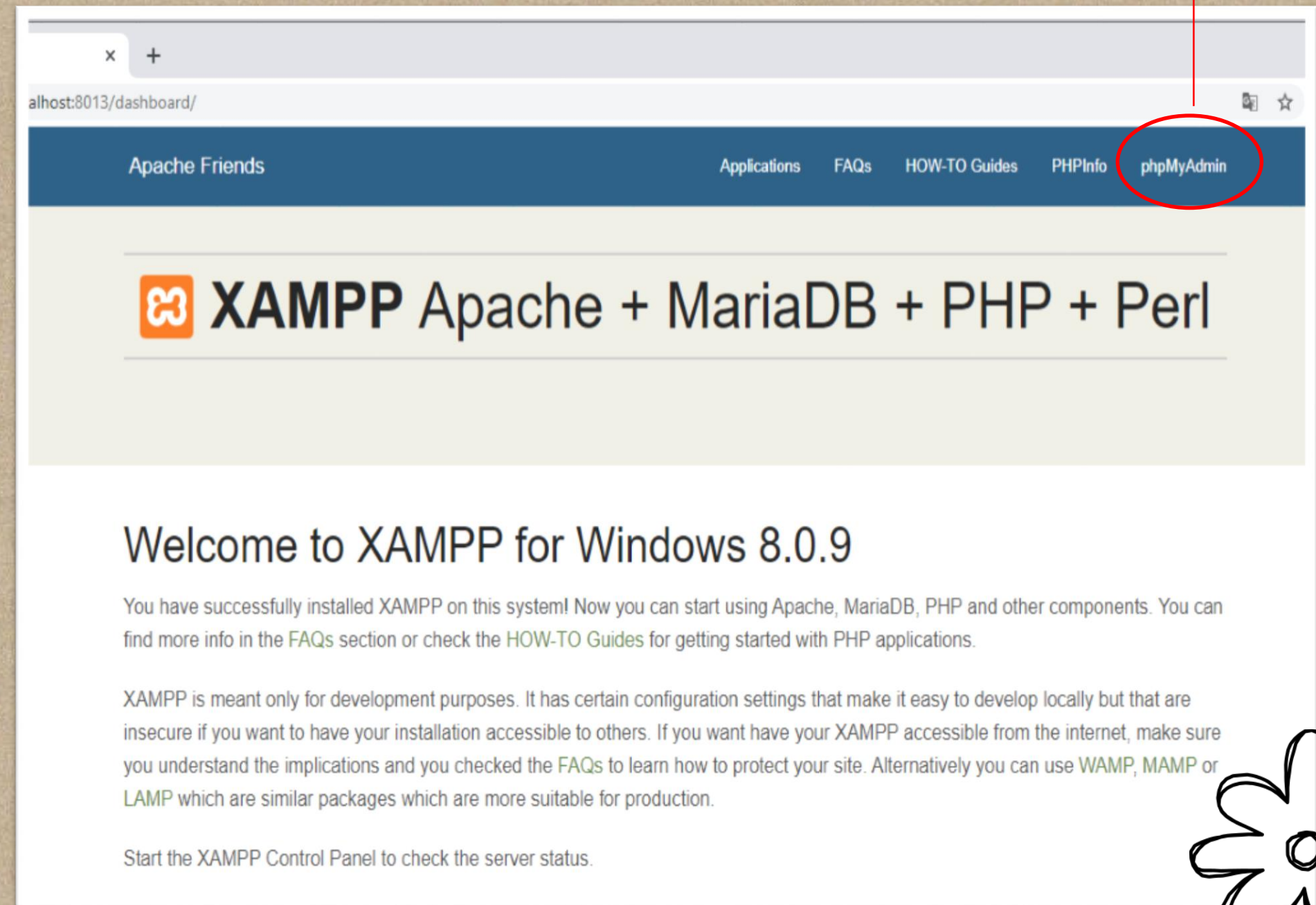
Tombol start  
service mySQL





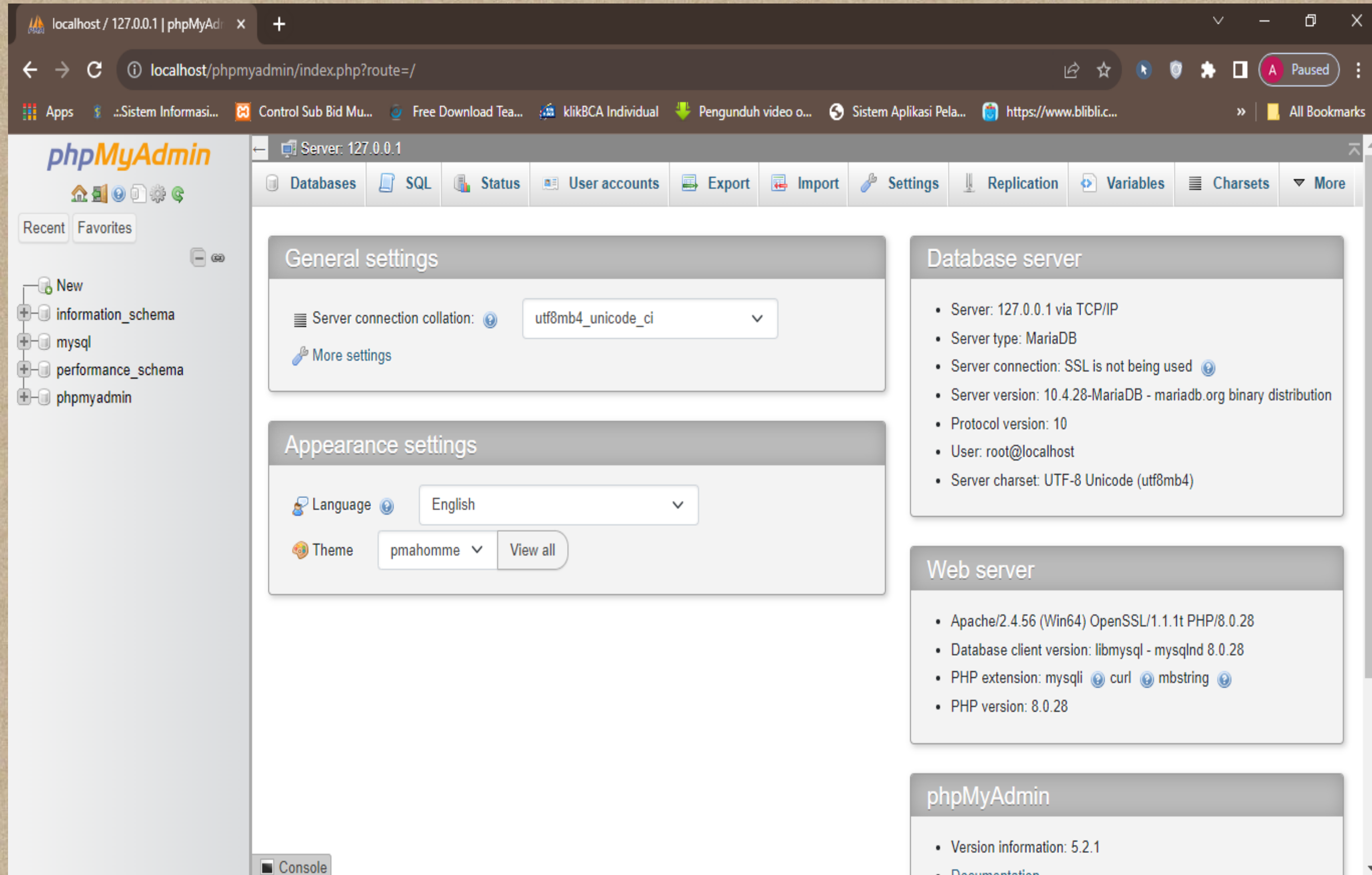
# Lingkungan MySQL

Untuk membuka aplikasi “*phpMyAdmin*”, kita tinggal mengetikkan “localhost” pada web browser, dan kemudian program akan menampilkan halaman utama (dashboard) dari XAMPP seperti berikut ini. Klik “*phpMyAdmin*” yang ada di ujung kanan atas halaman browser tersebut.





# Lingkungan MySQL



The screenshot displays the phpMyAdmin web interface in a browser window. The browser's address bar shows the URL `localhost/phpmyadmin/index.php?route=/`. The phpMyAdmin interface includes a top navigation bar with tabs for Databases, SQL, Status, User accounts, Export, Import, Settings, Replication, Variables, Charsets, and More. A left sidebar contains a tree view of databases: information\_schema, mysql, performance\_schema, and phpmyadmin. The main content area is divided into several panels:

- General settings:** Displays the server connection collation as `utf8mb4_unicode_ci` with a "More settings" link.
- Appearance settings:** Shows the language set to "English" and the theme set to "pmahomme", with a "View all" button.
- Database server:** Lists server details:
  - Server: 127.0.0.1 via TCP/IP
  - Server type: MariaDB
  - Server connection: SSL is not being used
  - Server version: 10.4.28-MariaDB - mariadb.org binary distribution
  - Protocol version: 10
  - User: root@localhost
  - Server charset: UTF-8 Unicode (utf8mb4)
- Web server:** Lists web server details:
  - Apache/2.4.56 (Win64) OpenSSL/1.1.1t PHP/8.0.28
  - Database client version: libmysql - mysqlnd 8.0.28
  - PHP extension: mysqli, curl, mbstring
  - PHP version: 8.0.28
- phpMyAdmin:** Shows version information (5.2.1) and a link to the documentation.

A "Console" tab is visible at the bottom left of the interface.



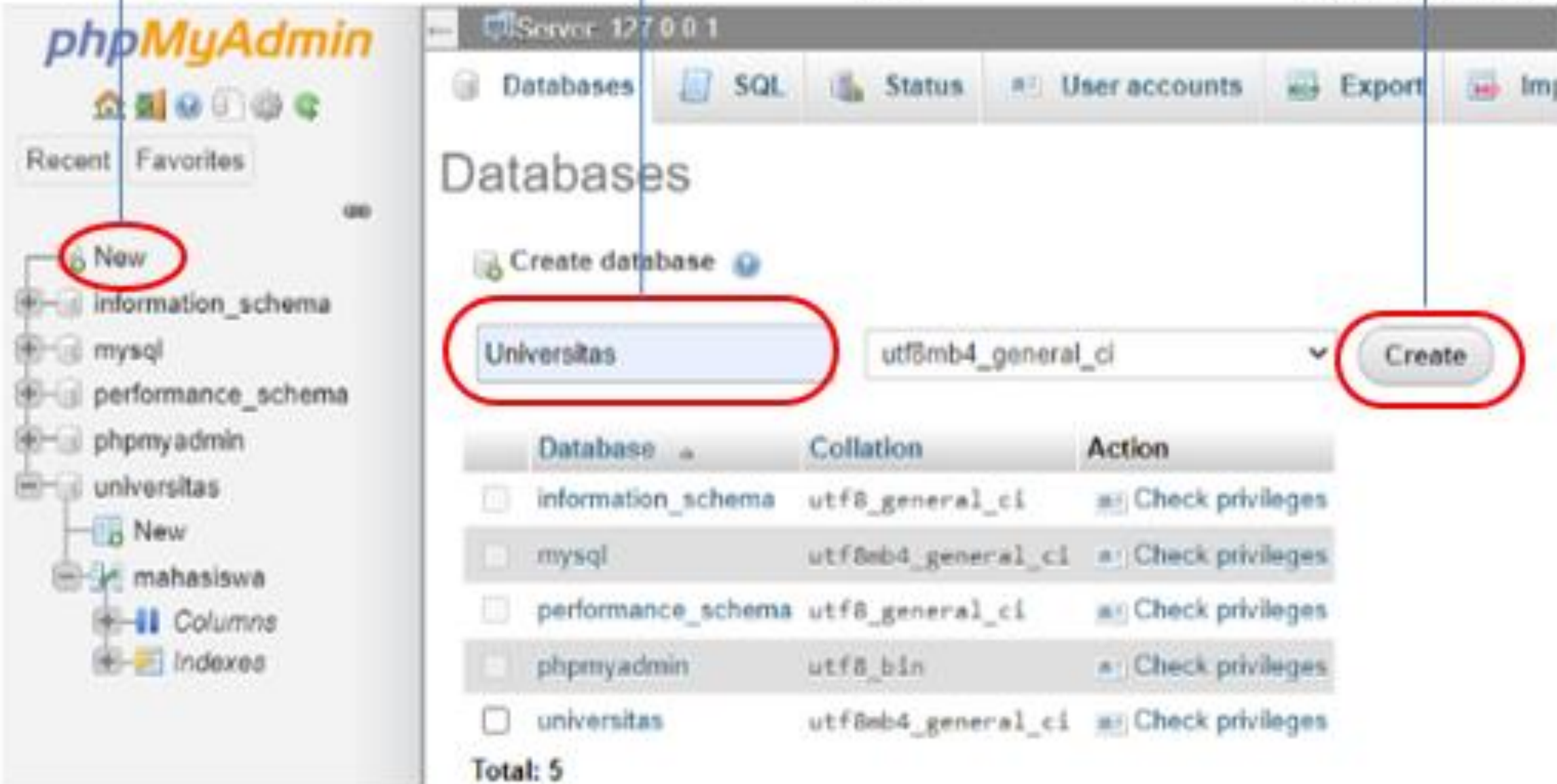


# Membuat Database baru

Tombol new untuk membuat database baru

Nama database baru

Tombol create setelah memberi nama database



phpMyAdmin

Server: 127.0.0.1

Databases SQL Status User accounts Export Import

Databases

Create database

Universitas utf8mb4\_general\_ci Create

Database	Collation	Action
<input type="checkbox"/> information_schema	utf8_general_ci	<a href="#">Check privileges</a>
<input type="checkbox"/> mysql	utf8mb4_general_ci	<a href="#">Check privileges</a>
<input type="checkbox"/> performance_schema	utf8_general_ci	<a href="#">Check privileges</a>
<input type="checkbox"/> phpmyadmin	utf8_bin	<a href="#">Check privileges</a>
<input type="checkbox"/> universitas	utf8mb4_general_ci	<a href="#">Check privileges</a>

Total: 5







# 5. Komponen DBMS



RINA MUSYAWARAH





# Komponen dalam database

## 1. Entity

Entity adalah setiap object yang dituangkan dalam sebuah database. Object atau entity tersebut bisa berupa orang, tempat, kejadian atau konsep yang informasinya direkam atau dicatat dalam database. Data mengenai entity atau object tersebut biasa dituangkan dalam bentuk **table**. Contoh entity adalah : mahasiswa, buku, pembayaran, absensi (Pada Bidang Kemahasiswaan). Pasien, dokter, obat, kamar (Pada Bidang Kedokteran).

## 2. Attribute.

Setiap entity mempunyai informasi yang berupa data atau sifat. Data atau sifat dari sebuah entity ini biasanya disebut dengan atribut. Contoh atribut dari entity mahasiswa adalah: nrp, nama, alamat, tgl lahir, dsb. Atribut inilah yang selanjutnya kita sebut sebagai **field**.

## 3. Karakter.

Karakter merupakan bagian data yang terkecil, dapat berupa karakter numerik, huruf ataupun karakter-karakter khusus yang membentuk suatu item data. Karakter ini biasanya kita sebut sebagai **type data** yang dimiliki oleh sebuah field dalam table.

## 4. Record.

Kumpulan dari field membentuk suatu baris data yang dinamakan **record**. Record menggambarkan suatu unit data individu tertentu.



# Create Table Mahasiswa

localhost / 127.0.0.1 / universitas x +

localhost/phpmyadmin/index.php?route=/database/structure&db=universitas

Apps ..Sistem Informasi... Control Sub Bid Mu... Free Download Tea... klikBCA Individual Pengunduh video o... Sistem Aplikasi Pela... https://www.blibli.c...

phpMyAdmin

Recent Favorites

New

- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- universitas

Server: 127.0.0.1 » Database: universitas

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers M

Table name: mahasiswa Add 1 column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
id	INT		None			<input type="checkbox"/>	PRIMARY
<small>Pick from Central Columns</small>							
npm	INT		None			<input type="checkbox"/>	---
<small>Pick from Central Columns</small>							
nama	VARCHAR	50	None			<input type="checkbox"/>	---
<small>Pick from Central Columns</small>							
alamat	VARCHAR	100	None			<input type="checkbox"/>	---
<small>Pick from Central Columns</small>							
kota	VARCHAR	50	None			<input type="checkbox"/>	---
<small>Pick from Central Columns</small>							
jkkelamin	VARCHAR	10	None			<input type="checkbox"/>	---
<small>Pick from Central Columns</small>							



# Struktur Table Mahasiswa

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Tracking

Table structure

Relation view

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2	npm	int(11)			No	None			Change  Drop  More
<input type="checkbox"/>	3	nama	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4	alamat	varchar(100)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	5	kota	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	6	jkelamin	varchar(10)	utf8mb4_general_ci		No	None			Change  Drop  More

☐ Check all

With selected:

Browse

Change

Drop

Primary

Unique

Index

Spatial

Fulltext

Add to central columns

Remove from central columns

Print

Propose table structure

?

Track table

Move columns

Normalize

Add

column(s)










Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit  Rename  Drop	PRIMARY	BTREE	Yes	No	id	2	A	No	





# Insert record Table Mahasiswa







 Browse	 Structure	 SQL	 Search	 Insert	 Export	 Import	 Privileges	 Operations
Column	Type	Function	Null	Value				
id	int(11)	<input type="text"/>		<input type="text"/>				
npm	int(11)	<input type="text"/>		<input type="text" value="22120022"/>				
nama	varchar(50)	<input type="text"/>		<input type="text" value="Abu Bakar"/>				
alamat	varchar(100)	<input type="text"/>		<div>Jl. <u>Nias</u> 20</div>				
kota	varchar(50)	<input type="text"/>		<input type="text" value="Surabaya"/>				
jkelamin	varchar(10)	<input type="text"/>		<input type="text" value="pria"/>				





# Data Table Mahasiswa

Extra options

<div>↩️⌵⌶➡️</div>					id	npm	nama	alamat	kota	jkelamin		
<input type="checkbox"/>		Edit		Copy		Delete	1	22120021	Abu Bakar	Jl. Nias 20	Surabaya	pria
<input type="checkbox"/>		Edit		Copy		Delete	2	22120022	Ali	Jl. Veteran 125	Surabaya	pria
<input type="checkbox"/>		Edit		Copy		Delete	3	22120023	Umar	Jl. Fatmawati	Jakarta	pria
<input type="checkbox"/>		Edit		Copy		Delete	4	22120024	Aisyah	Jl. Ampera 10	Padang	wanita
<input type="checkbox"/>		Edit		Copy		Delete	5	22120025	Khadijah	Jl. Teuku Umar	Gresik	wanita
<div>⬆️</div>	<input type="checkbox"/>	Check all	With selected:			Edit		Copy		Delete		Export





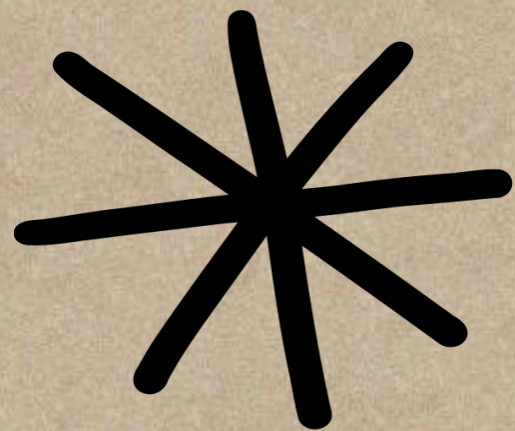
# Constraint

**Constraint** atau batasan dalam database pada dasarnya adalah pembatasan nilai-nilai yang diperbolehkan untuk diisikan dalam kolom-kolom yang terdapat dalam sebuah tabel. Dengan menerapkan batasan-batasan dalam sebuah tabel maka dapat dipastikan bahwa hanya data-data yang sesuai dengan batasan tersebut saja yang dapat diisikan dalam tabel tersebut. Fungsi utama dari penerapan constraint adalah untuk menjamin konsistensi dan integritas data-data yang diisikan dalam sebuah tabel.





## Macam-macam Constraint



- **Not Null**, berfungsi untuk membatasi setiap data yang dimasukkan di suatu kolom pada tabel database SQL harus memiliki nilai.
- **Unique**, berfungsi untuk membatasi setiap data yang dimasukkan di suatu kolom pada tabel database SQL, data yang dimasukkan harus berbeda dengan data yang ada sebelumnya sehingga tidak ada data yang sama dalam satu kolom.
- **Primary Key**, constraint ini mirip dengan constraint Unique dan biasanya berfungsi sebagai acuan dalam suatu tabel data. Constraint primary key ini hanya bisa dipasangkan pada satu kolom dalam satu tabel.
- **Foreign Key**, berfungsi sebagai rujukan ketika memasukkan data dalam suatu kolom. Jadi jika ada data yang akan dimasukkan di kolom yang memiliki constraint Foreign Key, maka data tersebut sebelumnya sudah harus ada di tabel lain yang sudah dijadikan rujukan.
- **Check**, berfungsi untuk melakukan pengecekan data sebelum disimpan di dalam kolom tabel SQL. Constraint check ini mirip quality controll, sehingga apabila data yang akan disimpan di kolom tidak sesuai persyaratan yang dibuat, maka data tersebut tidak dapat disimpan di kolom tersebut, sehingga SQL akan menampilkan pesan error bahwa data tersebut tidak lolos uji cek.
- **Default**, constraint ini menentukan nilai default untuk kolom. Default kolom adalah sebuah nilai yang akan dimasukkan dalam kolom oleh mesin database ketika pernyataan INSERT tidak menetapkan nilai tertentu pada sebuah kolom.



Buatlah sebuah table dengan nama “prodi”, kemudian isilah dengan beberapa record

Struktur table

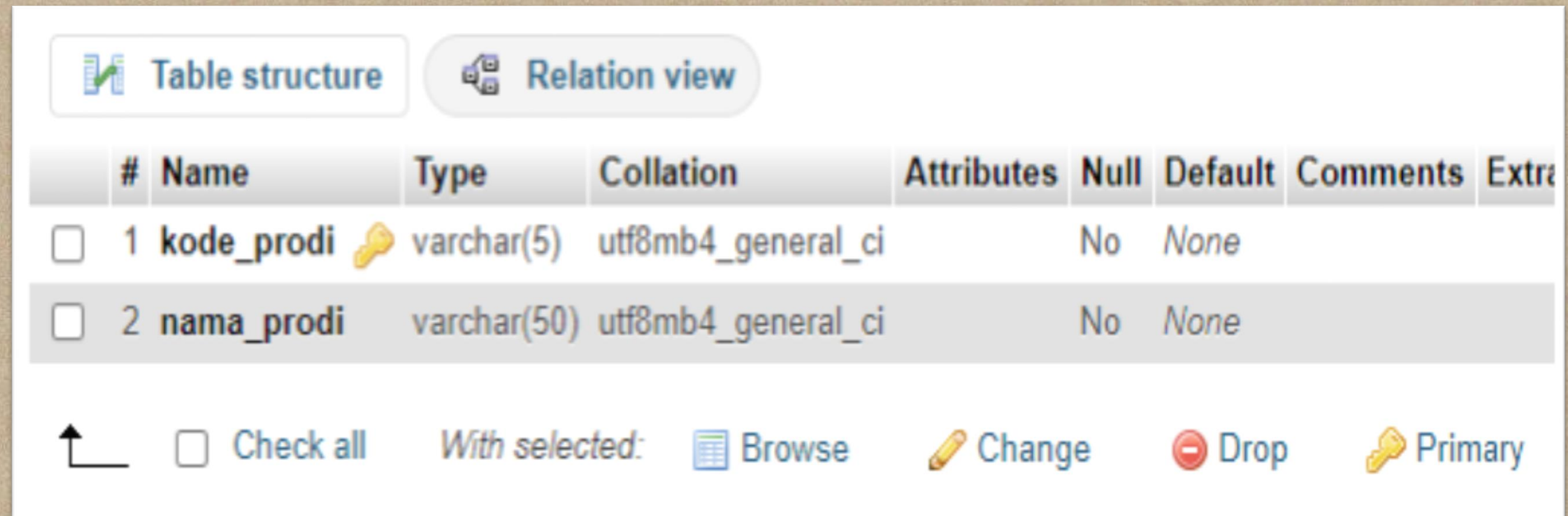


Table structure

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	kode_prodi	varchar(5)	utf8mb4_general_ci		No	None		
<input type="checkbox"/>	2	nama_prodi	varchar(50)	utf8mb4_general_ci		No	None		

↑ ☐ Check all With selected: Browse Change Drop Primary

Record table


Extra options

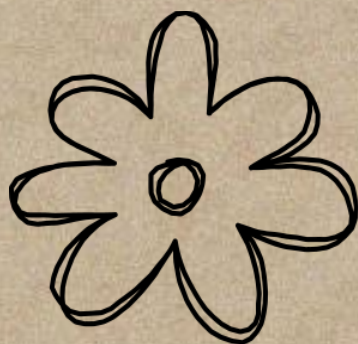
					kode_prodi	nama_prodi
<input type="checkbox"/>	Edit	Copy	Delete	P0001	Akuntansi	
<input type="checkbox"/>	Edit	Copy	Delete	P0002	Informatika	





Buatlah sebuah table dengan nama “matkul”,  
yang mempunyai struktur table sbb :

	#	Name	Type	Collation	Attributes	Null	Default	Co
<input type="checkbox"/>	1	kode_matkul 	varchar(5)	utf8mb4_general_ci		No	None	
<input type="checkbox"/>	2	nama_matkul	varchar(50)	utf8mb4_general_ci		No	None	
<input type="checkbox"/>	3	jml_sks	int(11)			No	None	
<input type="checkbox"/>	4	kode_prodi	varchar(5)	utf8mb4_general_ci		No	None	





Hubungkan ke dua table (“prodi” dan “makul”) dengan membuat sebuah constraint foreign key pada table “matkul”

Table structure

Relation view

Foreign key constraints

Actions	Constraint properties	Column ?	Foreign key constraint (INNODB)		
			Database	Table	Column
	<div>kode_prodi</div>				
ON DELETE	CASCADE	kode_prodi	universitas	prodi	kode_prodi
ON UPDATE	CASCADE				
+ Add column					
+ Add constraint					

Internal relationships

Choose column to display: kode\_prodi

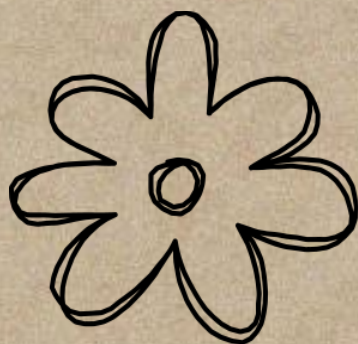
Preview SQL

Save



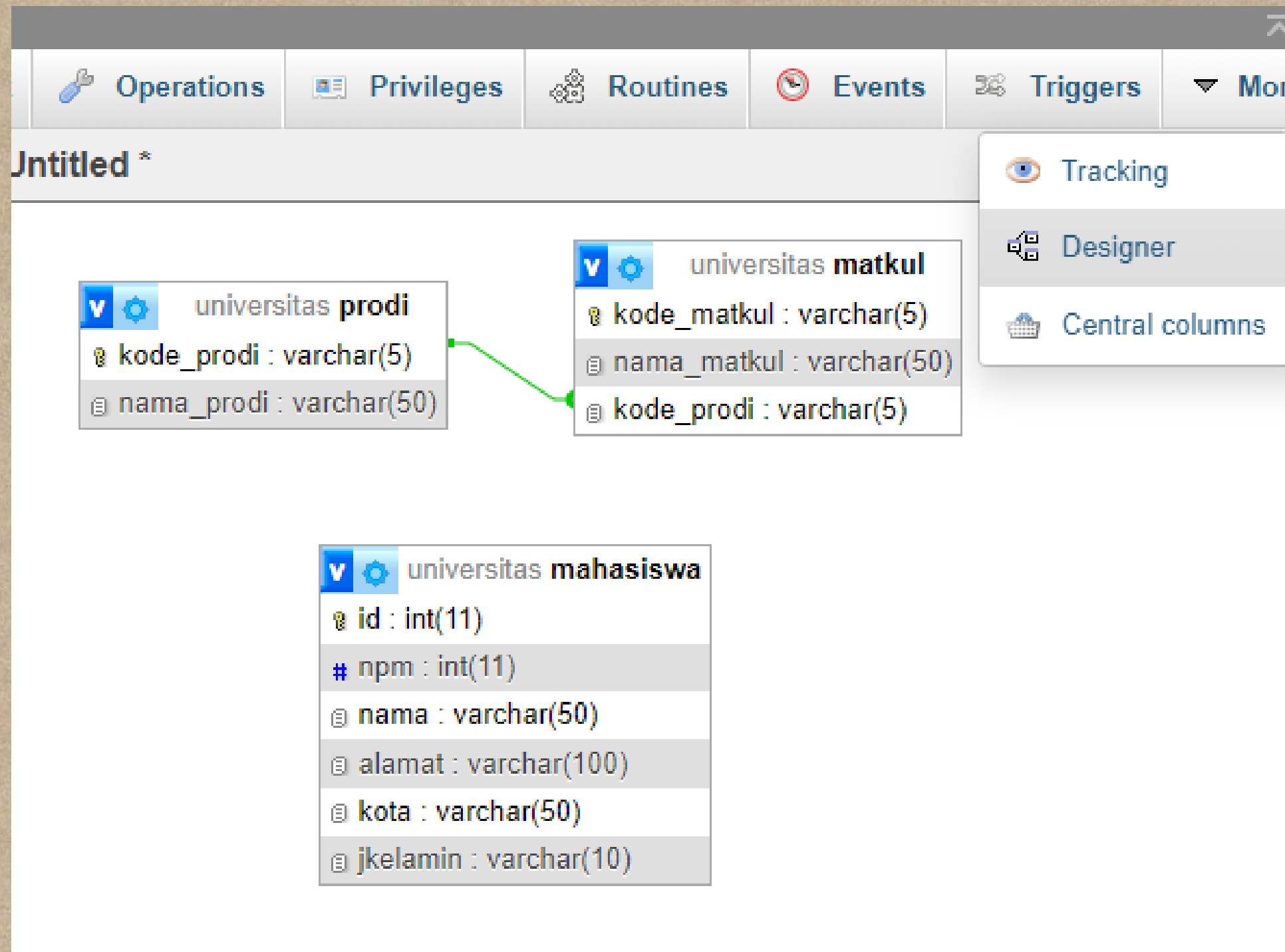
Isilah table “matkul” dengan beberapa record/data sbb :

 				kode_matkul	nama_matkul	jml_sks	kode_prodi
<input type="checkbox"/>	 Edit	 Copy	 Delete	MK001	Akuntansi Dasar	2	P0001
<input type="checkbox"/>	 Edit	 Copy	 Delete	MK002	Pengantar TIK	3	P0002
<input type="checkbox"/>	 Edit	 Copy	 Delete	MK003	Aplikasi perkntoran	3	P0001
<input type="checkbox"/>	 Edit	 Copy	 Delete	MK004	Jaringan Komputer	3	P0002





Pilih tab “designer” untuk melihat hasil relasional table yang sudah pernah dibuat dengan menggunakan foreign key







## 6. Operasional DBMS



RINA MUSYAWARAH





# Operasional DBMS

Untuk mengoperasikan sebuah database, terdapat sebuah perintah atau bahasa pemrograman yang biasa disebut dengan **SQL (Structured Query Language)**, yaitu sebuah bahasa pemrograman yang digunakan untuk mengakses serta mengolah data dari sebuah system database. Pada umumnya, programmer biasa menyebutnya dengan SQL command. Pada saat ini. Hampir semua aplikasi database mendukung perintah SQL dalam hal manajemen atau pengelolaan database. Perintahnya pun hampir sama untuk semua aplikasi database yang menerapkan konsep DBMS.



# Macam-macam operasional DBMS

## 1. **DDL (Data Definition Language)**

DDL adalah sebuah metode query SQL yang digunakan untuk memberikan definisi data pada sebuah database. Perintah ini termasuk membuat tabel baru, mengubah dataset, dan menghapus data. Ada 4 perintah dasar dari DDL, yaitu : **Create, alter, drop, dan show.**

## 2. **DML (Data Manipulation Language)**

Sesuai dengan namanya, DML atau Data Manipulation Language adalah query yang digunakan untuk memanipulasi data yang ada pada sebuah database. Perintah DML juga terbagi ke dalam beberapa jenis, beberapa diantaranya adalah:

**Insert** : perintah insert digunakan untuk menambahkan record atau data baru dalam tabel database.

**Select** : query ini bisa digunakan untuk menampilkan maupun mengambil data pada sebuah tabel, data yang diambil tidak hanya terbatas pada satu jenis.

**Update** : perintah yang digunakan jika Anda ingin melakukan pembaruan data di sebuah tabel, berguna jika ada kesalahan input pada saat membuat tabel.

**Delete** : digunakan untuk menghapus record yang ada dalam tabel database

## 3. **DCL (Data Control Language)**

Perintah ini biasanya digunakan untuk mengatur hak atau batasan yang diberikan kepada pengguna dalam mengakses data dalam sebuah database. Melalui perintah DCL, admin database bisa dengan mudah menjaga kerahasiaan atau keamanan sebuah database. Query dasar DCL dibagi menjadi beberapa perintah utama, yaitu:

**Grant** : perintah grant digunakan saat admin memberikan akses kepada user.

**Revoke** : query ini digunakan untuk membatalkan hak izin seorang user.

**Commit** : perintah commit digunakan untuk menetapkan penyimpanan database.

**Rollback** : query rollback digunakan untuk membatalkan penyimpanan database.



Contoh insert  
record mealui  
perintah SQL :

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operat

Run SQL query/queries on table universitas.matkul: ?

1

INSERT INTO `matkul`(`kode\_matkul`, `nama\_matkul`, `jml\_sks`, `kode\_prodi`) VALUES ('MK006','Akuntansi Publik',2,'P0001')

SELECT \*

SELECT

INSERT

UPDATE

DELETE

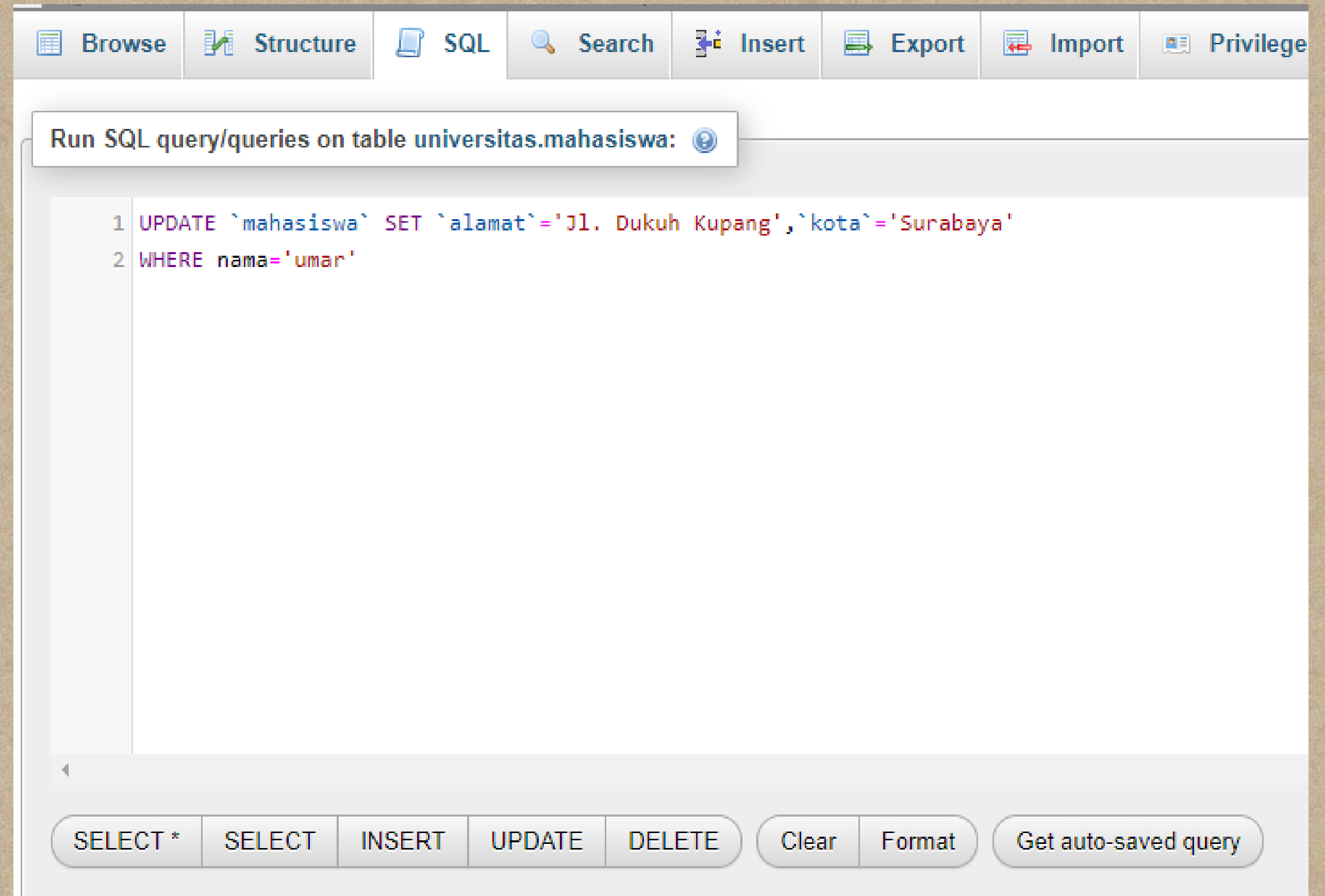
Clear

Format

Get auto-saved query

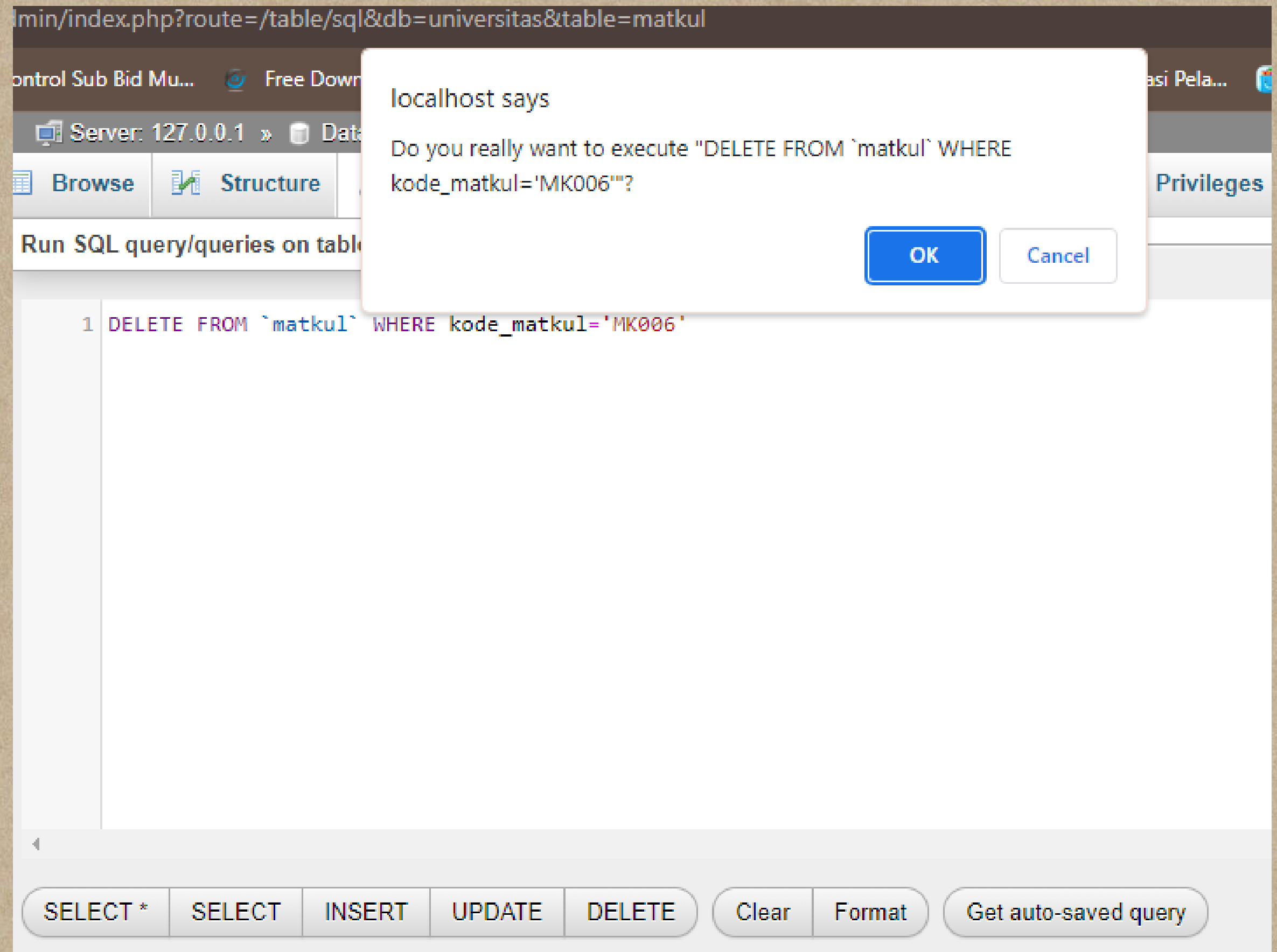


Contoh update  
record mealui  
perintah SQL :





Contoh delete  
record mealui  
perintah SQL :





Contoh select  
record mealui  
perintah SQL :

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Run SQL query/queries on table universitas.mahasiswa: ?

1

SELECT \* FROM `mahasiswa` WHERE kota='surabaya'

SELECT \*

SELECT

INSERT

UPDATE

DELETE











Clear

Format

Get auto-saved query



Query result :

 ▼				id	npm	nama	alamat	kota	jkelamin
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	22120021	Abu Bakar	Jl. Nias 20	Surabaya	pria
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	22120022	Ali	Jl. Veteran 125	Surabaya	pria
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	22120023	Umar	Jl. Dukuh Kupang	Surabaya	pria