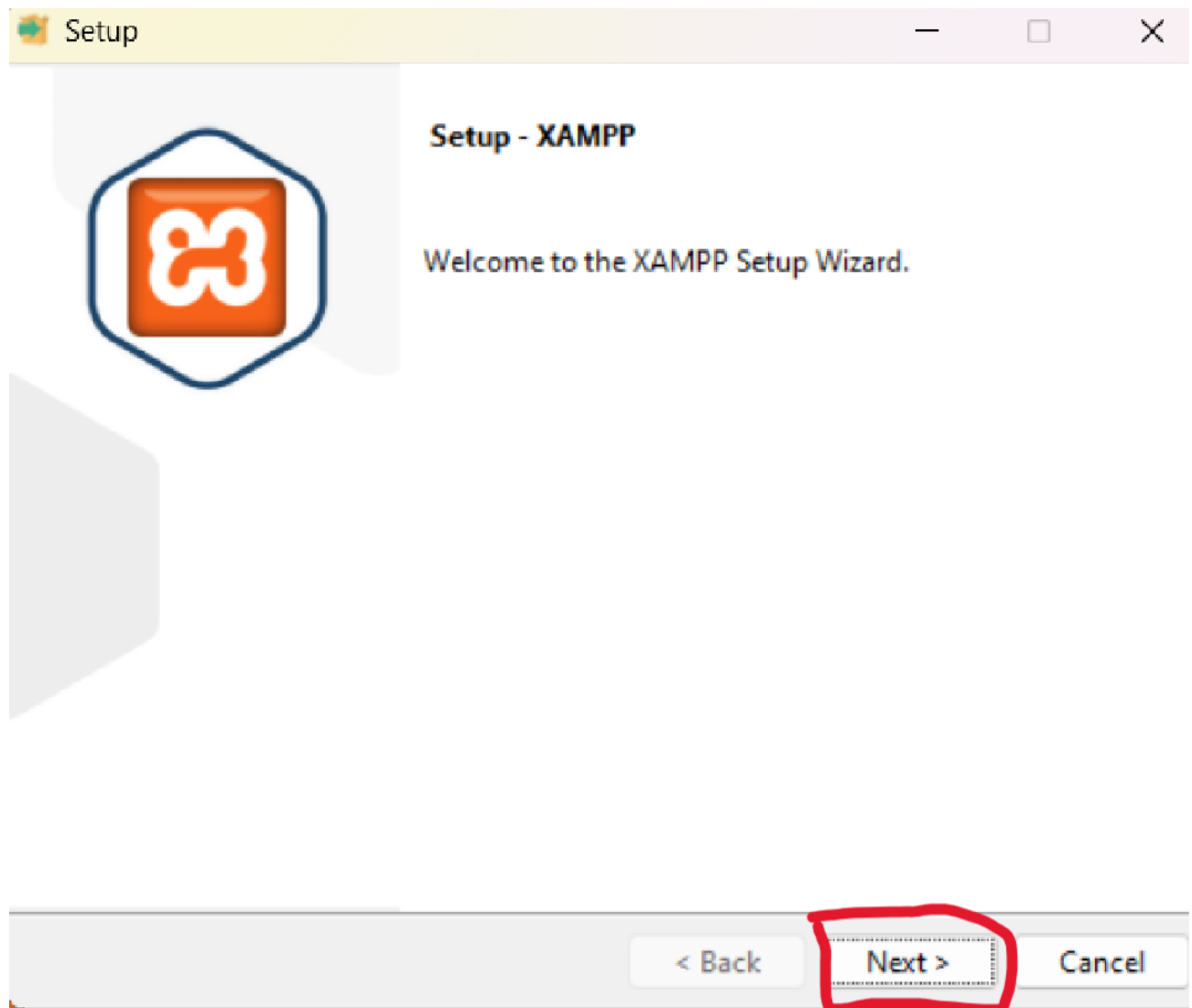


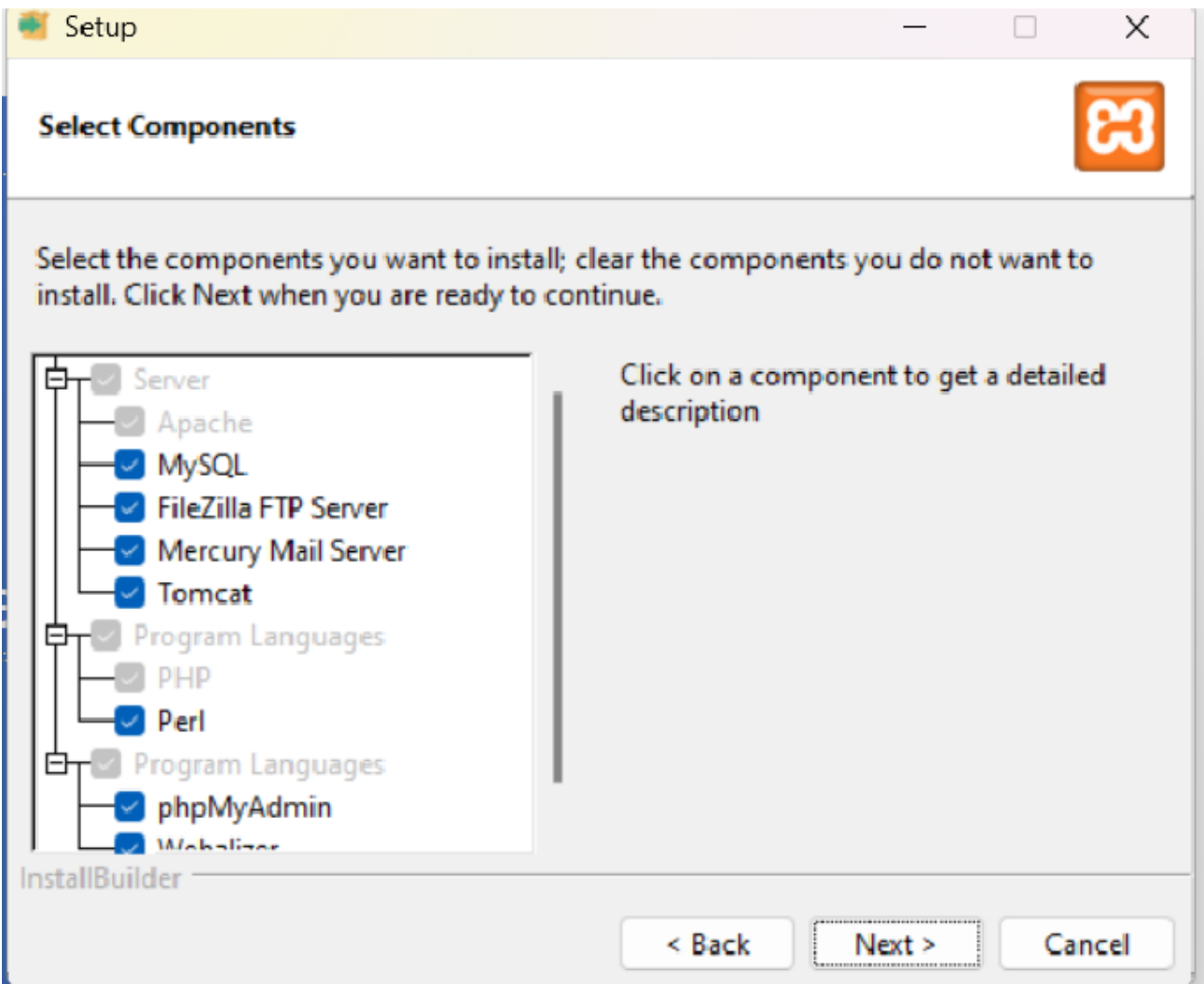
Instalasi & Query Awal Database

Instalasi Mysql

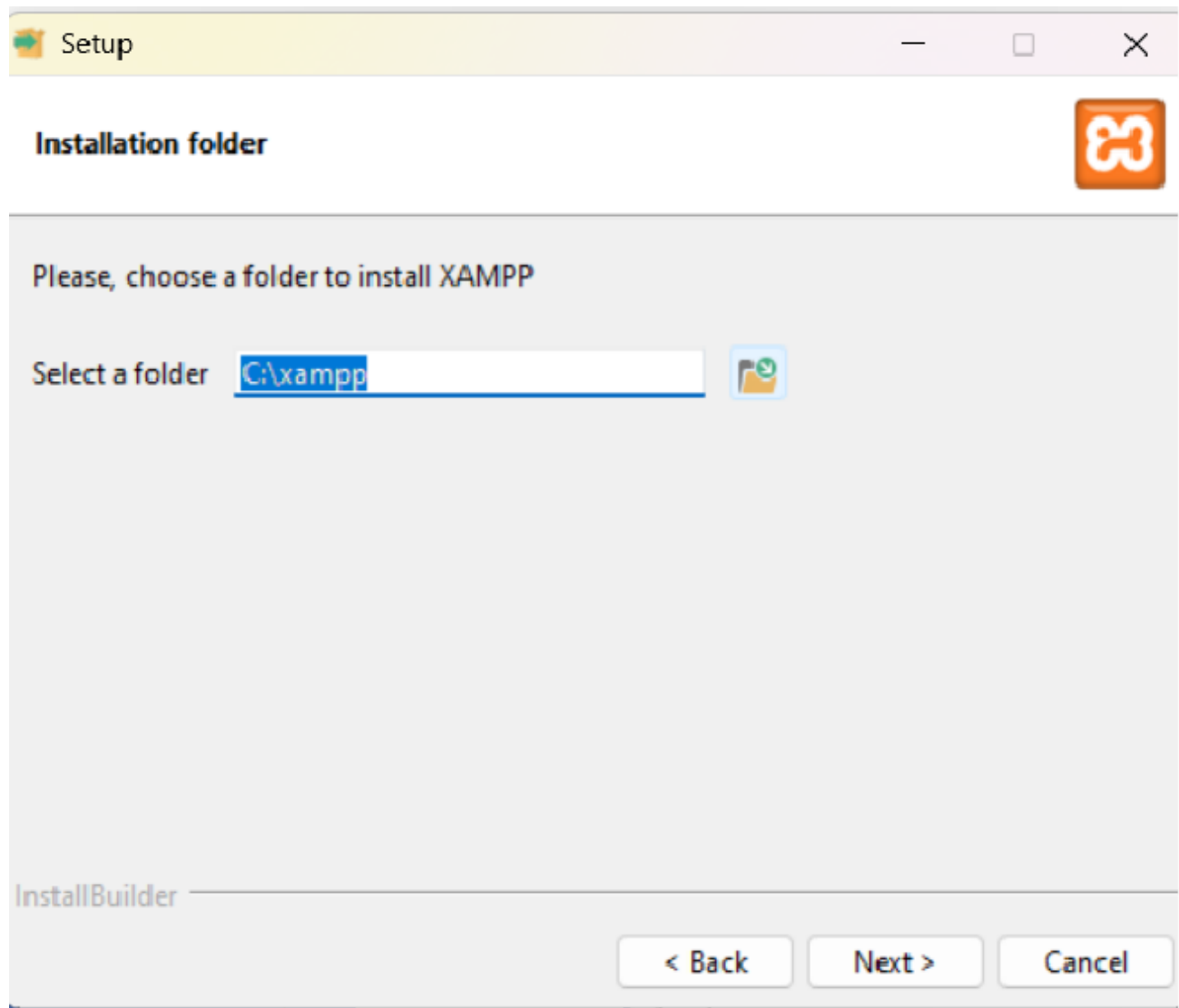
1. Kunjungi tautan berikut untuk mengakses halaman download Xampp:
<https://www.apachefriends.org/download.html>
2. Klik Download Pada versi terbaru
3. Setelah download selesai, anda dapat menemukannya di folder "Download"
4. klik installer Xampp untuk memulai pemasangan di komputer anda
5. Setelah di klik, kemungkinan akan muncul pop up konfirmasi untuk menginstall, silahkan pilih "Yes"
6. Selanjutnya muncul Pop up peringatan untuk tidak memasang Xampp di drive C dikarenakan adanya batasan-batasan akses pada drive tersebut sehingga fitur-fitur Xampp mungkin akan berjalan kurang maksimal, tekan "OK" untuk melanjutkan pemasangan.
7. Klik "next" untuk lanjutkan.
Hasil



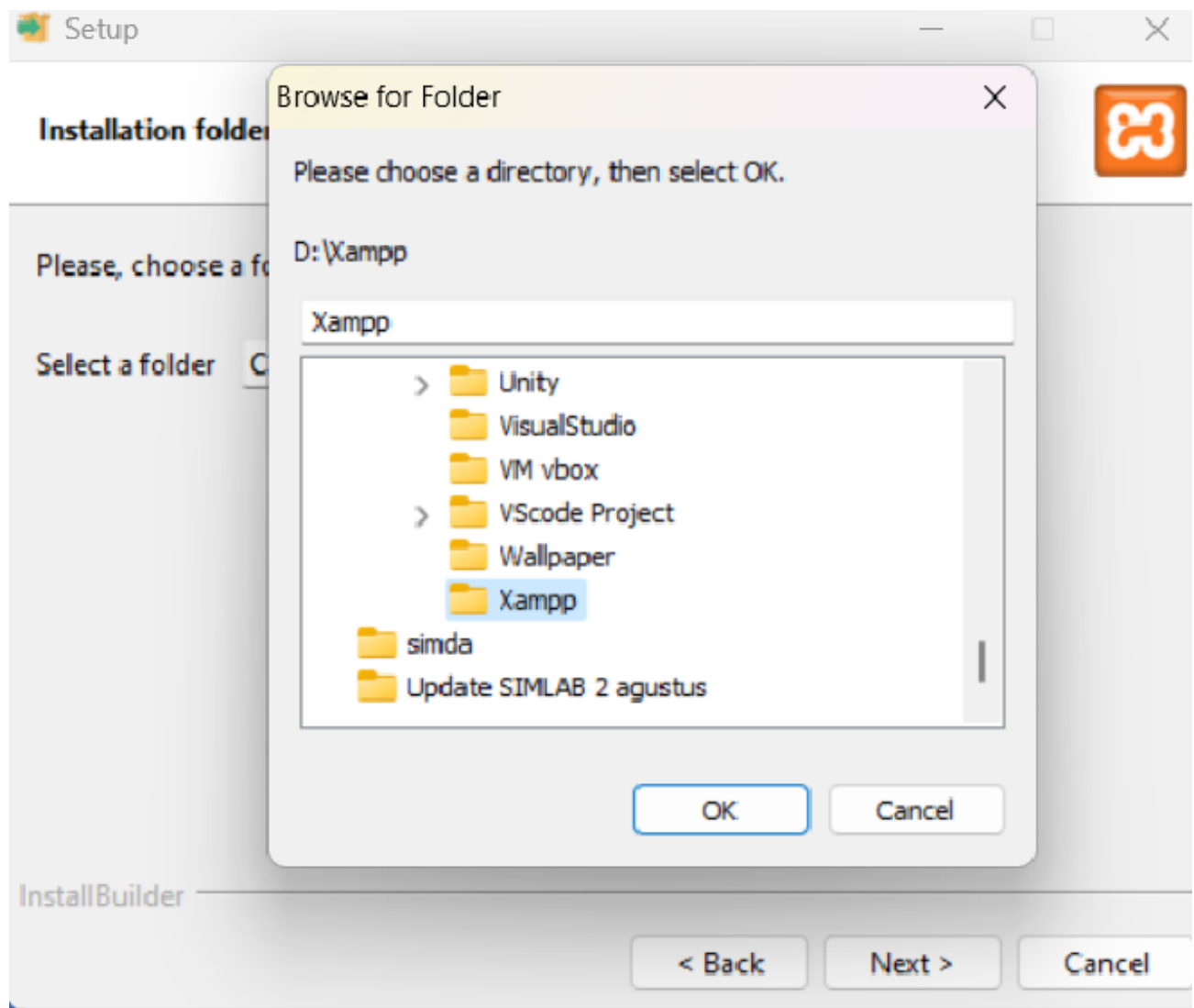
8. Sesuaikan dengan kebutuhan : MySQL merupakan yang utama, sangat disarankan untuk tetap mencentang phpMyAdmin agar nantinya database dapat diakses dengan GUI, selain itu tidak ada salahnya mencentang yang lain.



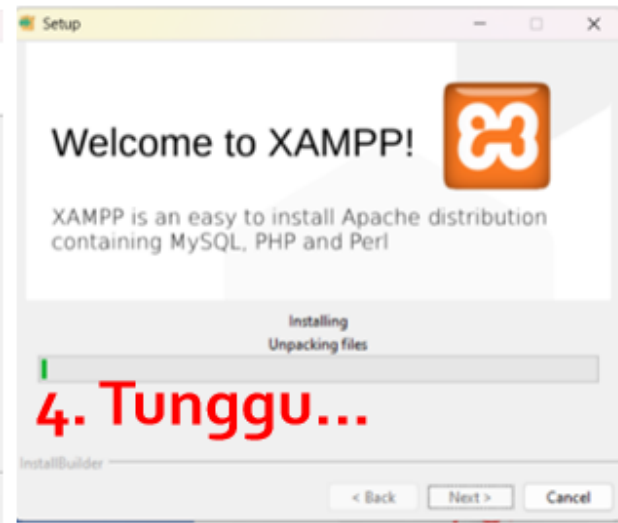
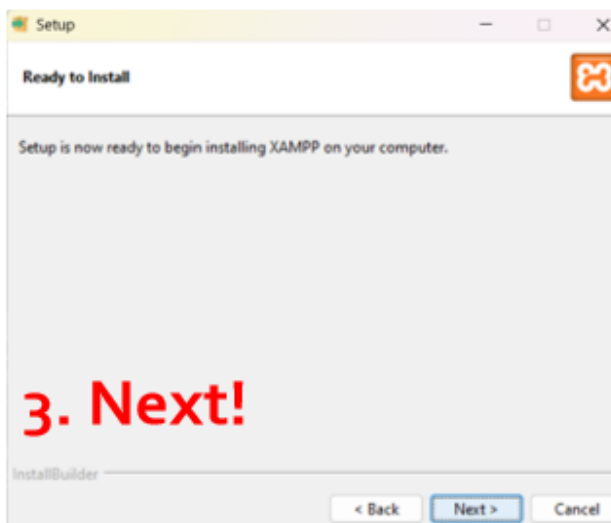
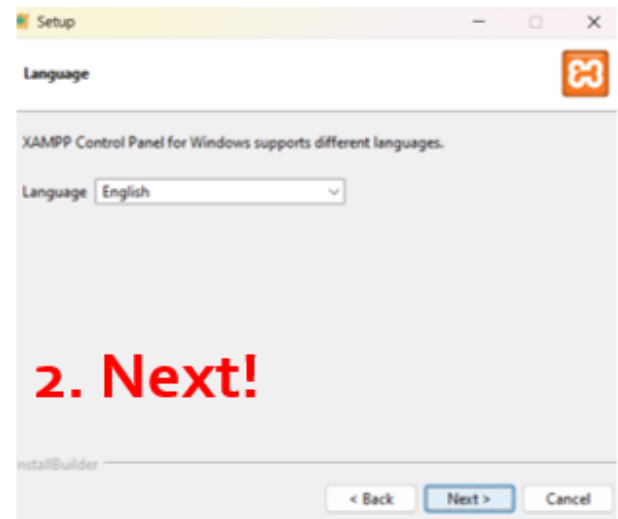
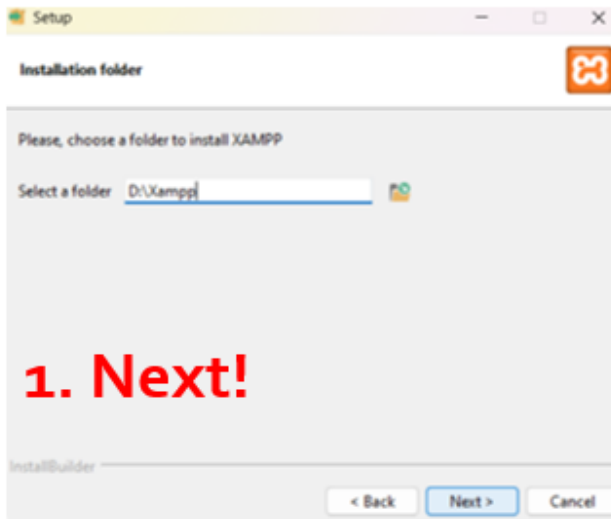
9. Pengaturan Folder Pemasangan XAMPP : Mengingat peringatan sebelumnya, ada baiknya anda menghindari drive C, ikuti arahan gambar disamping untuk mengubah folder pemasangan



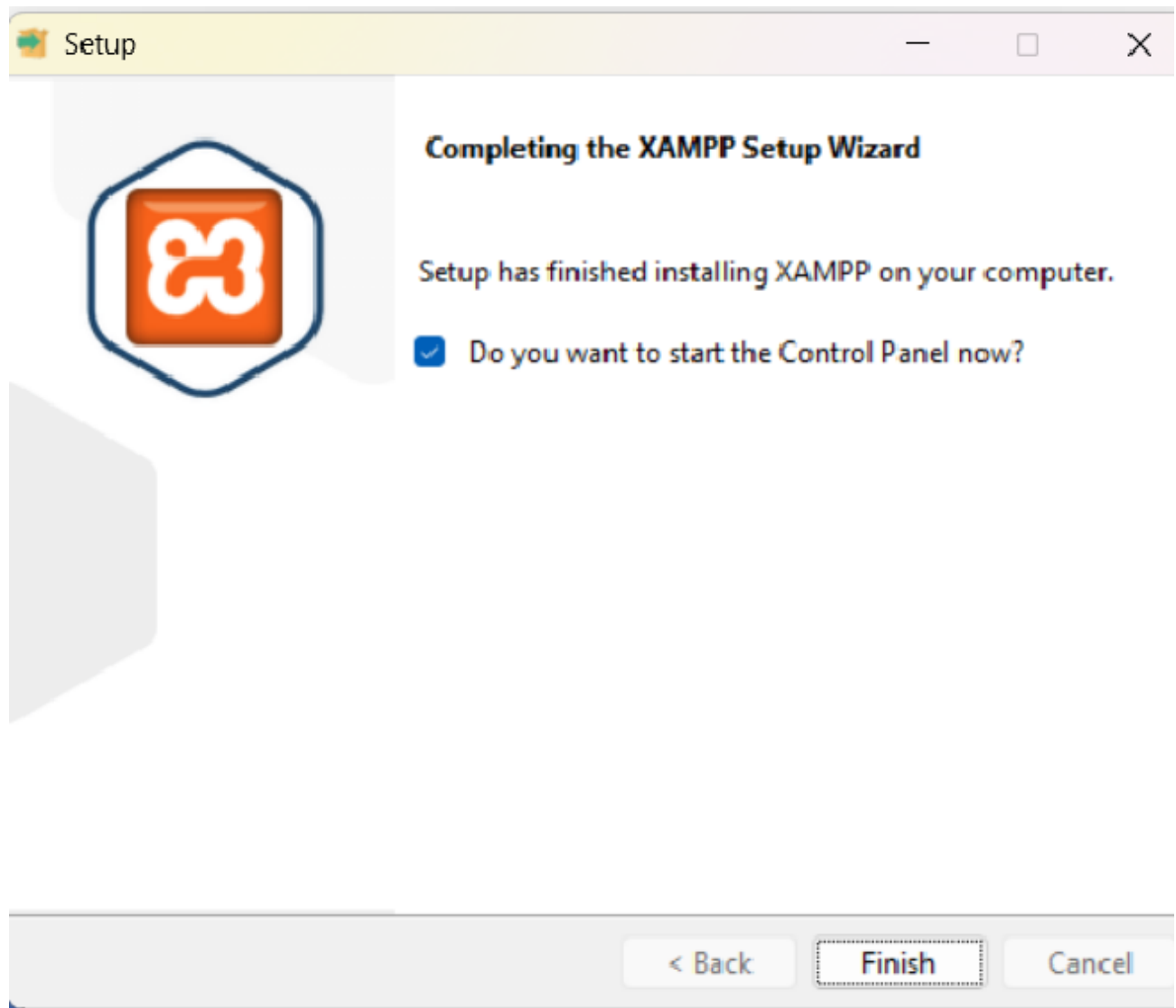
10. Cari Folder : Pastikan folder yang anda pilih benar-benar kosong!
Buat folder baru jika tidak ada folder kosong



11. Di next terus...

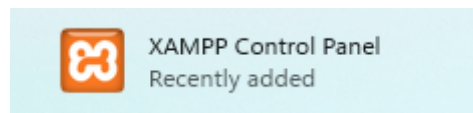


12. Selesai : Biarkan centangnya agar Control Panel Xampp langsung terbuka setelah anda mengklik "Finish"

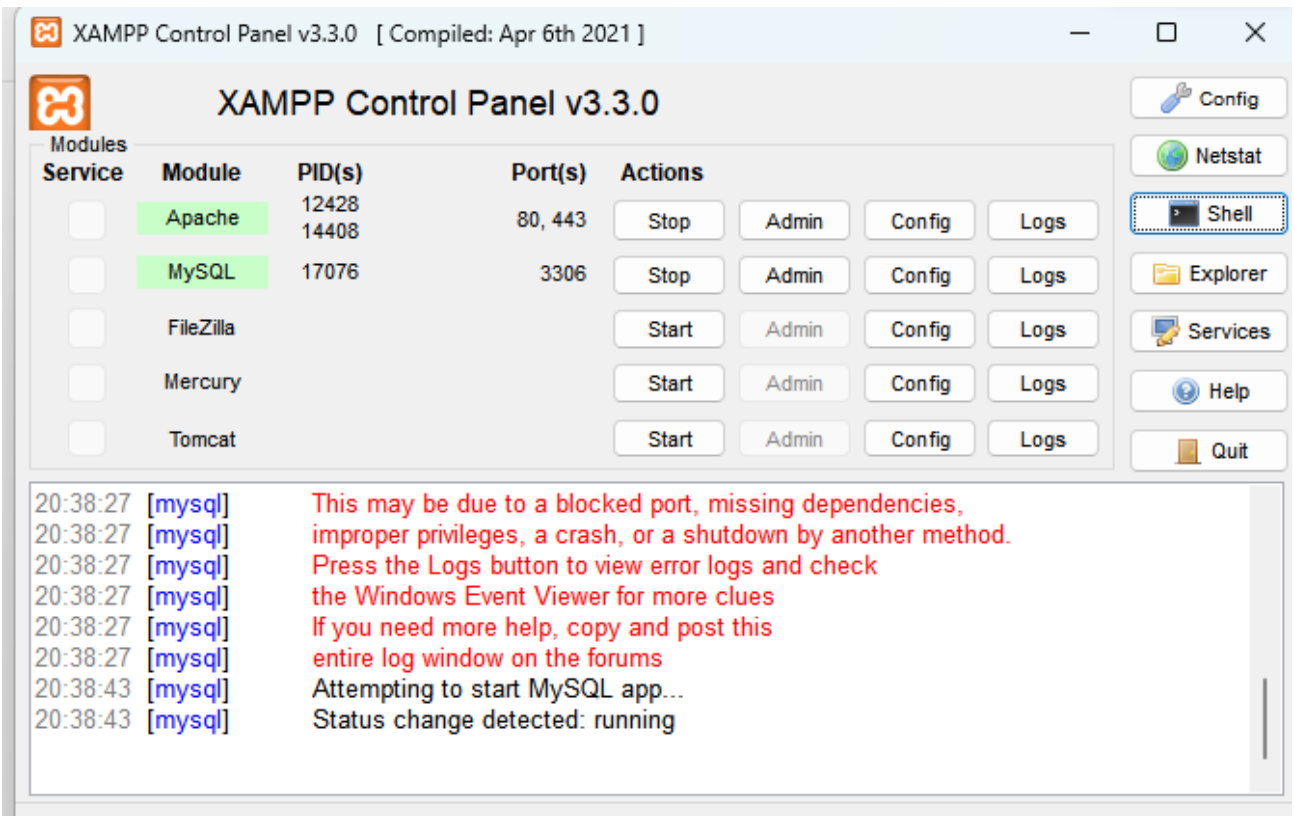


Menggunakan XAMPP

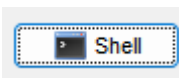
1. Buka aplikasi XAMPP



2. Klik `start` di MySQL



3. Klik Shell



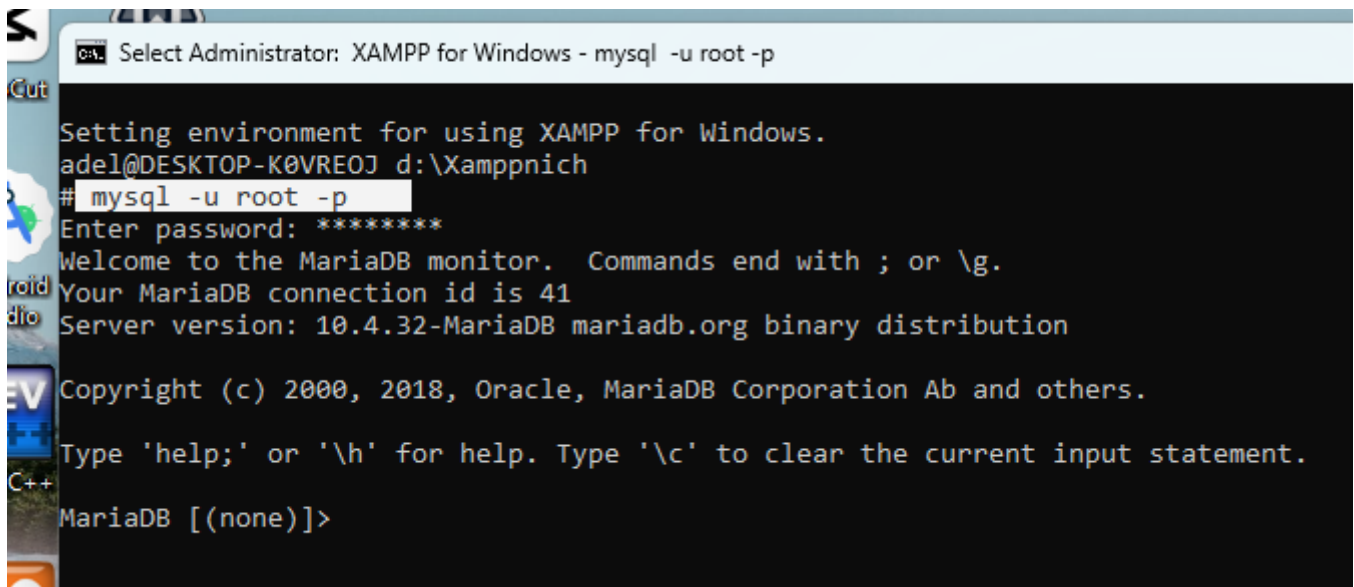
Referensi Video Youtube

Penggunaan Awal MySQL

Query

```
mysql -u root -p
```

Hasil



```

C:\> Select Administrator: XAMPP for Windows - mysql -u root -p

Setting environment for using XAMPP for Windows.
adel@DESKTOP-K0VRE0J d:\Xamppnich
# mysql -u root -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 41
Server version: 10.4.32-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Analisis Kesimpulan

- `MySQL` Ini adalah perintah yang digunakan untuk memulai klien MySQL di baris perintah. Klien MySQL memungkinkan kita untuk berinteraksi dengan server MySQL.
- `-U root` Dalam contoh ini, pengguna yang digunakan adalah «root.» Pengguna «root» biasanya memiliki hak akses penuh ke server MySQL dan dapat melakukan tindakan administratif.
- `-p` Opsi ini digunakan untuk meminta kata sandi setelah perintah dijalankan. Ini adalah langkah keamanan yang umum digunakan untuk memastikan hanya pengguna yang sah yang dapat mengakses server MySQL.

Database

Database adalah **kumpulan data yang disimpan dengan sistem tertentu, dan saling berhubungan, sehingga dapat dikelola dengan mudah**. Database penting untuk mengatur data yang jumlahnya banyak, dan selalu bertambah. Sebagai contoh, program website, aplikasi, dan lainnya.

Buat Database

Query

```
create database [nama database]
```

Contoh Query

```
create database XI_RPL_1;
```

Hasil

```
MariaDB [(none)]> create database XI_RPL_1;  
Query OK, 1 row affected (0.002 sec)
```

Analisis

- `CREATE DATABASE` adalah perintah untuk membuat database baru.
- `[XI_RPL_1]` adalah nama yang Anda pilih untuk database baru Anda.

Kesimpulan

Tampilkan Database

Query

```
show database
```

```
show databases;
```

Hasil

```
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| phpmyadmin |  
| test |  
| xi_rpl_1 |  
+-----+  
6 rows in set (0.001 sec)
```

Analisis Kesimpulan

- `SHOW DATABASES` digunakan untuk menampilkan daftar database yang ada dalam sistem manajemen basis data (DBMS). Perintah ini dapat digunakan di beberapa DBMS seperti MySQL, PostgreSQL, dan beberapa DBMS lainnya. Namun, perintahnya dapat sedikit berbeda tergantung pada DBMS yang digunakan.

Hapus Database

Query

```
drop database [nama_database]
```

Contoh Query

```
drop database xi_rpl_1;
```

Hasil

```
MariaDB [(none)]> drop database xi_rpl_1;  
Query OK, 0 rows affected (0.112 sec)
```

Analisis Kesimpulan

"DROP DATABASE" digunakan untuk menghapus sebuah database beserta semua objek yang terkait seperti tabel, indeks, tampilan, dan lainnya dari sistem manajemen basis data (DBMS).

- `[XI_RPL_1]` adalah nama database yang ingin Anda hapus.

Gunakan Database

Query

```
use [nama_database]
```

Hasil

```
MariaDB [(none)]> use XI_RPL_1;  
Database changed  
MariaDB [XI_RPL_1]>
```

Analisis Kesimpulan

Dengan menggunakan perintah `use`, Anda dapat mengatur database aktif yang akan digunakan untuk menjalankan perintah-perintah selanjutnya.

- `USE` adalah perintah yang digunakan untuk beralih ke database yang ditentukan.

- `[nama_database]` adalah nama database yang ingin Anda gunakan. Anda perlu mengganti "`[nama_database]`" dengan nama sebenarnya dari database yang ingin Anda akses.

Tipe Data

Angka

1. Integer (Bilangan Bulat): Tipe data ini digunakan untuk merepresentasikan bilangan bulat. Contoh tipe data integer adalah 1, 2, 100, -10, dst.
2. Floating Point (Bilangan Pecahan): Tipe data ini digunakan untuk merepresentasikan bilangan pecahan atau desimal. Contoh tipe data floating point adalah 3.14, 2.5, -0.75, dst. Tipe data ini biasanya direpresentasikan dalam format desimal atau notasi ilmiah (misalnya 3.14e-2).
3. Long (Bilangan Bulat Panjang): Tipe data ini digunakan untuk merepresentasikan bilangan bulat yang sangat besar atau sangat kecil yang tidak dapat direpresentasikan oleh tipe data integer biasa. Contoh tipe data long adalah 10000000000L, -9000000000000L, dst. Tanda "L" pada akhir angka menunjukkan bahwa nilai tersebut adalah long.
4. Double (Bilangan Pecahan Ganda): Tipe data ini merupakan tipe data floating point yang lebih presisi dibandingkan dengan tipe data float. Contoh tipe data double adalah 3.14159, 2.71828, -1.4142, dst. Tipe data ini biasanya digunakan jika diperlukan presisi yang lebih tinggi dalam perhitungan.

Teks

1. String: Tipe data string digunakan untuk merepresentasikan kumpulan karakter. Contoh tipe data string adalah "Halo", "Nama Saya", "123", dst. Tipe data string biasanya digunakan untuk menyimpan teks, kata-kata, kalimat, atau data teks lainnya.
2. Character: Tipe data character digunakan untuk merepresentasikan satu karakter. Contoh tipe data character adalah 'A', 'b', '3', dst. Karakter diapit oleh tanda kutip tunggal (' ').
3. Char: Tipe data char adalah singkatan dari character dan digunakan dalam beberapa bahasa pemrograman untuk merepresentasikan satu karakter.
4. StringBuilder: Tipe data StringBuilder digunakan untuk memanipulasi dan mengubah string secara efisien. Tipe data ini cocok digunakan jika terdapat perubahan yang sering dilakukan pada string, karena objek StringBuilder dapat dimodifikasi tanpa harus membuat objek baru setiap kali.
5. StringBuffer: Tipe data StringBuffer juga digunakan untuk memanipulasi dan mengubah string, mirip dengan StringBuilder. Namun, StringBuffer bersifat thread-safe, artinya dapat digunakan dalam lingkungan yang melibatkan penggunaan bersama oleh beberapa thread.

Tanggal

1. Date: Tipe data Date digunakan untuk merepresentasikan tanggal dalam bentuk bulan, tanggal, dan tahun. Tipe data ini biasanya digunakan untuk operasi-operasi dasar terkait tanggal seperti penghitungan selisih antara dua tanggal atau format tanggal tertentu.

2. Calendar: Tipe data Calendar digunakan untuk merepresentasikan tanggal dengan lebih banyak informasi seperti hari, bulan, tanggal, tahun, jam, menit, detik, dan zona waktu. Tipe data Calendar menyediakan lebih banyak fitur dan operasi terkait penanganan tanggal dan waktu.
3. LocalDateTime: Tipe data LocalDateTime digunakan untuk merepresentasikan tanggal dan waktu lokal tanpa zona waktu. Tipe data ini biasanya digunakan untuk operasi-operasi terkait tanggal dan waktu dalam lingkungan yang tidak memerlukan zona waktu.
4. ZonedDateTime: Tipe data ZonedDateTime digunakan untuk merepresentasikan tanggal dan waktu dengan zona waktu. Tipe data ini menyimpan informasi tentang tanggal, waktu, dan zona waktu.
5. Instant: Tipe data Instant digunakan untuk merepresentasikan titik waktu tertentu dalam bentuk jumlah detik yang telah berlalu sejak epoch Unix (1 Januari 1970 00:00:00 UTC). Tipe data ini sering digunakan untuk perhitungan operasi terkait waktu atau untuk mengukur interval waktu.

Boolean

Tipe data boolean digunakan untuk merepresentasikan nilai kebenaran (true atau false) dalam pemrograman. Tipe data boolean sering digunakan dalam struktur pengkondisian dan pengontrol aliran program. Terdapat dua nilai boolean yang mungkin:

1. true: Mewakili kondisi yang benar atau terpenuhi.
2. false: Mewakili kondisi yang salah atau tidak terpenuhi.

Pilihan

Table

Buat tabel

Struktur Query

```
create table [nama table]
```

Contoh Query

```
create table mobil;
```

Hasil

```
MariaDB [rental_adel]> create table mobil
```

Analisis

Perintah "CREATE TABLE" digunakan dalam SQL untuk membuat sebuah tabel baru dalam basis data. Berikut adalah penjelasan mengenai sintaks dan bagaimana menggunakan perintah "CREATE TABLE":

1. Kata kunci "CREATE TABLE" diikuti oleh nama tabel yang ingin Anda buat. Misalnya, jika Anda ingin membuat tabel dengan nama **=mobil=**.

Kesimpulan

Struktur buat tabel

Query

```
create table [nama table] (  
  namakolom tipedata(lebar) cons,  
  -----,  
)
```

Hasil

```
MariaDB [rental_adel]> create table mobil (  
  -> nama_mobil varchar(15) primary key not null,  
  -> plat_mobil char(10) not null unique,  
  -> warna_mobil varchar(10) not null unique);  
Query OK, 0 rows affected (0.529 sec)
```

Analisis Kesimpulan

1. Kolom "nama_mobil" didefinisikan sebagai VARCHAR(15), yang berarti itu akan menyimpan data string dengan panjang maksimal 15 karakter. Kemudian, PRIMARY KEY menandakan bahwa kolom ini akan menjadi kunci utama untuk tabel, yang berarti nilainya harus unik dan tidak boleh kosong (NOT NULL).
2. Kolom "plat_mobil" didefinisikan sebagai CHAR(10), yang berarti itu akan menyimpan data karakter dengan panjang tetap sebanyak 10 karakter. Selanjutnya, NOT NULL menunjukkan bahwa kolom ini harus memiliki nilai (tidak boleh kosong), dan UNIQUE menandakan bahwa nilai-nilai dalam kolom ini harus unik (tidak ada duplikat).
3. Kolom "warna_mobil" didefinisikan sebagai VARCHAR(10), yang berarti itu akan menyimpan data string dengan panjang maksimal 10 karakter. Seperti kolom sebelumnya, NOT NULL menunjukkan bahwa kolom ini harus memiliki nilai (tidak boleh kosong), dan UNIQUE menandakan bahwa nilai-nilai dalam kolom ini harus unik (tidak ada duplikat). b

Struktur Tabel

Query

```
describe [nama table];
```

Hasil

```
MariaDB [rental_adel]> describe mobil;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nama_mobil | varchar(15) | NO   | PRI | NULL    |       |
| plat_mobil | char(10)    | NO   | UNI | NULL    |       |
| warna_mobil | varchar(10) | NO   | UNI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.084 sec)
```

Analisis Kesimpulan

Perintah "DESCRIBE" digunakan dalam SQL untuk mendapatkan informasi tentang struktur kolom dalam sebuah tabel. Ini berguna untuk mengetahui nama kolom, tipe data, dan batasan yang diterapkan pada kolom tersebut. Berikut adalah penjelasan mengenai penggunaan perintah "DESCRIBE":

1. Kata "DESCRIBE" diikuti oleh nama tabel yang ingin di periksa. Misalnya, jika Anda ingin mengetahui informasi mengenai tabel dengan nama `=mobil=`.
2. Setelah menjalankan perintah "DESCRIBE", Anda akan mendapatkan hasil yang menampilkan informasi mengenai kolom-kolom dalam tabel tersebut. Informasi yang biasanya ditampilkan antara lain:
 - Nama kolom
 - Tipe data kolom
 - Panjang maksimal kolom (maximum length)
 - Apakah kolom dapat memiliki nilai NULL atau tidak (nullable)
 - Apakah kolom memiliki batasan unik (unique)

Show Table

Query

```
show tables;
```

Hasil

```
MariaDB [rental_adel]> show tables;
+-----+
| Tables_in_rental_adel |
+-----+
| mobil                  |
| pelanggan              |
+-----+
2 rows in set (0.163 sec)
```

Analisis Kesimpulan

Perintah "SHOW TABLES" digunakan dalam SQL untuk menampilkan daftar tabel yang ada dalam basis data. Ini memungkinkan Anda melihat tabel-tabel yang telah dibuat sebelumnya.

- Setelah menjalankan perintah "SHOW TABLES", Anda akan mendapatkan hasil yang menampilkan daftar tabel yang ada dalam basis data tersebut. Hasilnya dapat berupa daftar tabel dalam bentuk kolom tunggal atau kolom ganda, tergantung pada implementasi sistem manajemen basis data (DBMS) yang digunakan.

QNA

❓ PERBEDAAN PRIMARY KEY & UNIQUE >

- **PRIMARY KEY** digunakan untuk membuat kolom atau kombinasi kolom yang unik dan menjadi primary key dari tabel. Primary key digunakan sebagai identitas unik untuk setiap baris data pada tabel.
- **UNIQUE** digunakan untuk membuat kolom atau kombinasi kolom yang unik pada tabel. Namun, primary key dan unique tidak harus sama. Primary key selalu unik, sedangkan unique hanya memastikan bahwa setiap nilai pada kolom tersebut hanya muncul sekali.

❓ Mengapa hanya kolom id_pelanggan yang menggunakan primary key? >

memastikan integritas data dengan mencegah duplikasi dan memastikan bahwa setiap baris memiliki identitas yang jelas.

❓ Mengapa pada kolom no_telp yang menggunakan tipe data char bukan varchar? >

karena char merupakan tipe data yang bisa digabung antara karakter dan angka

❓ Mengapa hanya kolom `no_telp` yang menggunakan constraint `UNIQUE`? >

Constraint `UNIQUE` pada kolom `no_telp` digunakan untuk mengharuskan bahwa setiap nomor telepon yang ada pada tabel pelanggan harus unik. Ini berarti bahwa tidak ada dua pelanggan yang memiliki nomor telepon yang sama.

❓ Mengapa kolom `no_telp` tidak memakai constraint `NOT NULL`, sementara kolom lainnya > menggunakan constraint tersebut

Kolom `no_telp` pada tabel pelanggan tidak memakai constraint `NOT NULL` karena mungkin ada beberapa kasus dimana pelanggan tidak memiliki nomor telepon. Misalnya, pelanggan yang baru saja mendaftar dan belum memiliki nomor telepon, atau pelanggan yang tidak ingin menggunakan nomor telepon untuk kontak.

Insert

Insert 1 data

Struktur

```
INSERT INTO nama_tabel  
values (nilai1, nilai2, nilai3)
```

Contoh Query

```
insert into pelanggan  
values (1, 'adel', 'hrv', '081243935227');
```

Hasil

```
MariaDB [rental_adel]> insert into pelanggan  
-> values ('1', 'adel', 'hrv', '081243935');  
Query OK, 1 row affected (0.096 sec)
```

Analisis

Perintah `INSERT INTO` dalam MySQL digunakan untuk menyisipkan data baru ke dalam sebuah tabel dalam basis data MySQL. Dengan perintah `(1, 'adel', 'hrv', '081243935227');`, kita dapat menambahkan baris-baris data baru ke dalam tabel yang telah kita tentukan.

Kesimpulan

`insert into` digunakan untuk menambahkan data

Insert >1 data

Struktur Query

```
INSERT INTO nama_tabel  
values (nilai1, nilai2, nilai3),  
        (nilai1, nilai2, nilai3),  
        (nilai1, nilai2, nilai3);
```

Contoh Query

```
insert into pelanggan  
values (2, 'alwi', 'rizkyansyah', '0823456780'),  
        (3, 'adel', 'anil', '08234479423'),  
        (4, 'aril', 'haq', '0823456280');
```

Hasil

```
MariaDB [rental_adel]> insert into pelanggan  
-> values (2, 'alwi', 'rizkyansyah', '0823456780'),  
->      (3, 'adel', 'anil', '08234479423'),  
->      (4, 'aril', 'haq', '0823456280');  
Query OK, 3 rows affected (0.142 sec)  
Records: 3  Duplicates: 0  Warnings: 0
```

Analisis

1. `insert into` merupakan perintah yang digunakan untuk menginput isi tabel
2. `pelanggan` nama tabel yang akan di isi
3. `values` digunakan untuk memasukkan nilai ke dalam field
4. `(2, 'alwi', 'rizkyansyah', '0823456780'), (3, 'adel', 'anil', '08234479423'), (4, 'aril', 'haq', '0823456280');` merupakan nilai yang akan di masukkan

Kesimpulan

menambahkan beberapa data baru ke tabel pelanggan dengan nilai `id_pelanggan = 2`, `nama_depan = 'alwi'`, `nama_belakang = 'rizkyansyah'`, `no_telp = '0823456780'`, `id_pelanggan = 3`, `nama_depan = 'adel'`, `nama_belakang = 'anil'`, `no_telp = '08234479423'`, `id_pelanggan = 4`, `nama_depan = 'aril'`, `nama_belakang = 'haq'`, `no_telp = '0823456280'`.

Menyebut kolom

Struktur Query

```
Insert into [nama tabel]
(kolom1, kolom2, kolom3,) values (nilai1, nilai2, nilai3);
```

Contoh Query

```
insert into pelanggan
-> (id_pelanggan, nama_depan, nama_belakang, no_telp) values (5, 'azikin',
'sofyan', '93629205');
```

Hasil

```
MariaDB [rental_adel]> insert into pelanggan
-> (id_pelanggan, nama_depan, nama_belakang, no_telp) values (5, 'azikin', 'sofyan', '93629205');
Query OK, 1 row affected (0.062 sec)
```

Analisis

kolom yang akan di isi :

- id_pelanggan: Nilai 5
- nama_depan: 'azikin'
- nama_belakang: 'sofyan'
- no_telp: '93629205'

Pada sintaks setelah perintah `INSERT INTO` akan terdapat sintaks `(id_pelanggan, nama_depan, nama_belakang, no_telp)`. Sintaks inilah yang mengidentifikasi nama kolom yang mau ditambahkan.

Kesimpulan

kita dapat menambahkan nilai dengan menyebut nama kolom tersebut.

SELECT

Seluruh data

Struktur Query

```
select * from [nama tabel];
```

Contoh Query

```
select * from pelanggan;
```

Hasil

```
MariaDB [rental_adel]> select * from pelanggan;
+-----+-----+-----+-----+
| id_pelanggan | nama_depan | nama_belakang | no_telp |
+-----+-----+-----+-----+
| 1 | adel | hrv | 081243935 |
| 2 | alwi | rizkyansyah | 0823456780 |
| 3 | adel | anil | 08234479423 |
| 4 | aril | haq | 0823456280 |
| 5 | azikin | sofyan | 93629205 |
+-----+-----+-----+-----+
5 rows in set (0.001 sec)
```

Analisis

- `SELECT * FROM` adalah perintah untuk melihat item tabel
- `akun` adalah nama tabel yang datanya mau dilihat.

Kesimpulan

Query `SELECT * FROM pelanggan;` digunakan untuk menampilkan semua data dari tabel pelanggan.

Data kolom tertentu

Struktur

```
select [nama_kolom1], [nama_kolom2], [nama_kolom3]....
from [nama_tabel];
```

Contoh

```
select nama_depan from pelanggan;
```

Hasil

```
MariaDB [rental_adel]> select nama_depan from pelanggan;
+-----+
| nama_depan |
+-----+
| adel       |
| alwi       |
| adel       |
| aril       |
| azikin     |
+-----+
5 rows in set (0.001 sec)
```

Analisis

untuk menampilkan data nama saja caranya yaitu menggunakan perintah `SELECT` kemudian diikuti oleh judul kolom yang ingin ditampilkan, yaitu `nama_depan` Kemudian di lanjutkan dengan mengetik perintah `FROM` dan diikuti dengan nama tabel. Dengan cara seperti ini kita telah menampilkan item data pada kolom `nama_depan` pada tabel `pelanggan`.

Kesimpulan

Query `SELECT nama_depan FROM pelanggan;` digunakan untuk menampilkan data `nama_depan` dari tabel `pelanggan`

Klausula WHERE

Struktur

```
select [nama_kolom/*] from [nama_tabel]
      WHERE [kondisi];
```

Contoh

```
select nama_depan from pelanggan
where id=2;
```

Hasil

```
MariaDB [rental_adel]> select nama_depan from pelanggan
-> where id_pelanggan=2;
+-----+
| nama_depan |
+-----+
| alwi       |
+-----+
1 row in set (0.109 sec)
```

Analisis

1. **SELECT** : Kata kunci ini digunakan untuk menentukan kolom yang harus dikembalikan dalam kumpulan hasil. Dalam hal ini, kami memilih kolom nama_depan.
2. **FROM** : Kata kunci ini digunakan untuk menentukan tabel dari mana kita ingin mengambil data. Dalam hal ini, kami mengambil data dari tabel pelanggan.
3. **WHERE** : Kata kunci ini digunakan untuk memfilter baris yang harus disertakan dalam kumpulan hasil. Dalam hal ini, kami memfilter baris berdasarkan ketentuan bahwa kolom id harus sama dengan 2.
4. **id=2** : Ini adalah kondisi yang kita gunakan untuk memfilter baris. Artinya kita hanya tertarik pada baris yang kolom id-nya sama dengan 2.

Kesimpulan

Query SQL `SELECT nama_depan FROM pelanggan WHERE id = 2;` digunakan untuk memilih nilai kolom nama_depan dari tabel pelanggan di mana idnya adalah 2.

Update

Struktur Query

```
update nama_tabel set nama_kolom where kondisi;
```

Contoh Query

```
update pelanggan set no_telp="09834834892" where id_pelanggan="1";
```

Hasil

```

MariaDB [rental_adel]> update pelanggan set no_telp="09834834892" where id_pelanggan="1";
Query OK, 1 row affected (0.152 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [rental_adel]> select * from pelanggan;
+-----+-----+-----+-----+
| id_pelanggan | nama_depan | nama_belakang | no_telp |
+-----+-----+-----+-----+
| 1 | adel | hrv | 09834834892 |
| 2 | alwi | rizkyansyah | 0823456780 |
| 3 | adel | anil | 08234479423 |
| 4 | aril | haq | 0823456280 |
| 5 | azikin | sofyan | 93629205 |
+-----+-----+-----+-----+
5 rows in set (0.001 sec)

```

Analisis

Dalam query ini, menggunakan perintah `UPDATE` untuk mengubah data yang sudah ada di tabel `pelanggan`. Dan menggunakan klausa `SET` untuk menentukan kolom yang ingin diubah dan nilai baru yang ingin diberikan. klausa `WHERE` digunakan untuk menentukan baris yang ingin diubah berdasarkan kondisi yang diberikan.

Kesimpulan

Query `UPDATE pelanggan SET no_telp = '09834834892' WHERE id_pelanggan = 1` digunakan untuk mengubah data `no_telp` pada tabel `pelanggan` dengan `id_pelanggan` yang sama dengan `1` menjadi `09834834892`

Delete

Struktur Query

```
delete from nama_tabel where kondisi;
```

```
delete from pelanggan where id_pelanggan="5";
```

Contoh Query

Hasil

```
MariaDB [rental_adel]> delete from pelanggan where id_pelanggan="5";
Query OK, 1 row affected (0.125 sec)

MariaDB [rental_adel]> select * from pelanggan;
+-----+-----+-----+-----+
| id_pelanggan | nama_depan | nama_belakang | no_telp |
+-----+-----+-----+-----+
| 1 | adel | hrv | 09834834892 |
| 2 | alwi | rizkyansyah | 0823456780 |
| 3 | adel | anil | 08234479423 |
| 4 | aril | haq | 0823456280 |
+-----+-----+-----+-----+
4 rows in set (0.002 sec)
```

Analisis

Ketika query ini dijalankan, hanya satu baris data yang akan dihapus, yaitu baris dengan `id_pelanggan` yang sama dengan 5. Jika tidak ada data dengan `id_pelanggan` yang sama dengan 5, maka tidak akan terjadi perubahan apa-apa pada tabel `pelanggan`.

Kesimpulan

Query `DELETE FROM pelang WHERE id_pelanggan = "5"` digunakan untuk menghapus data pelanggan dengan `id_pelanggan` yang sama dengan 5.

Hapus Tabel

Struktur Query

```
drop table [nama_tabel]
```

Contoh Query

```
drop table mobil;
```

Hasil

```
MariaDB [rental_adel]> drop table mobil;
Query OK, 0 rows affected (0.249 sec)
```

Analisis

Ketika query ini dijalankan, semua data dan struktur tabel mobil akan dihapus secara permanen dari database. Ini berarti bahwa data yang ada pada tabel mobil tidak dapat dikembalikan setelah tabel mobil dihapus.

Kesimpulan

Query `DROP TABLE mobil;` digunakan untuk menghapus tabel mobil dari database.