

# DATABASE INTRODUCTION

Pengenalan Basis Data

# APA ITU BASIS DATA?



# PENGERTIAN BASIS DATA

Pengertian Dasar:

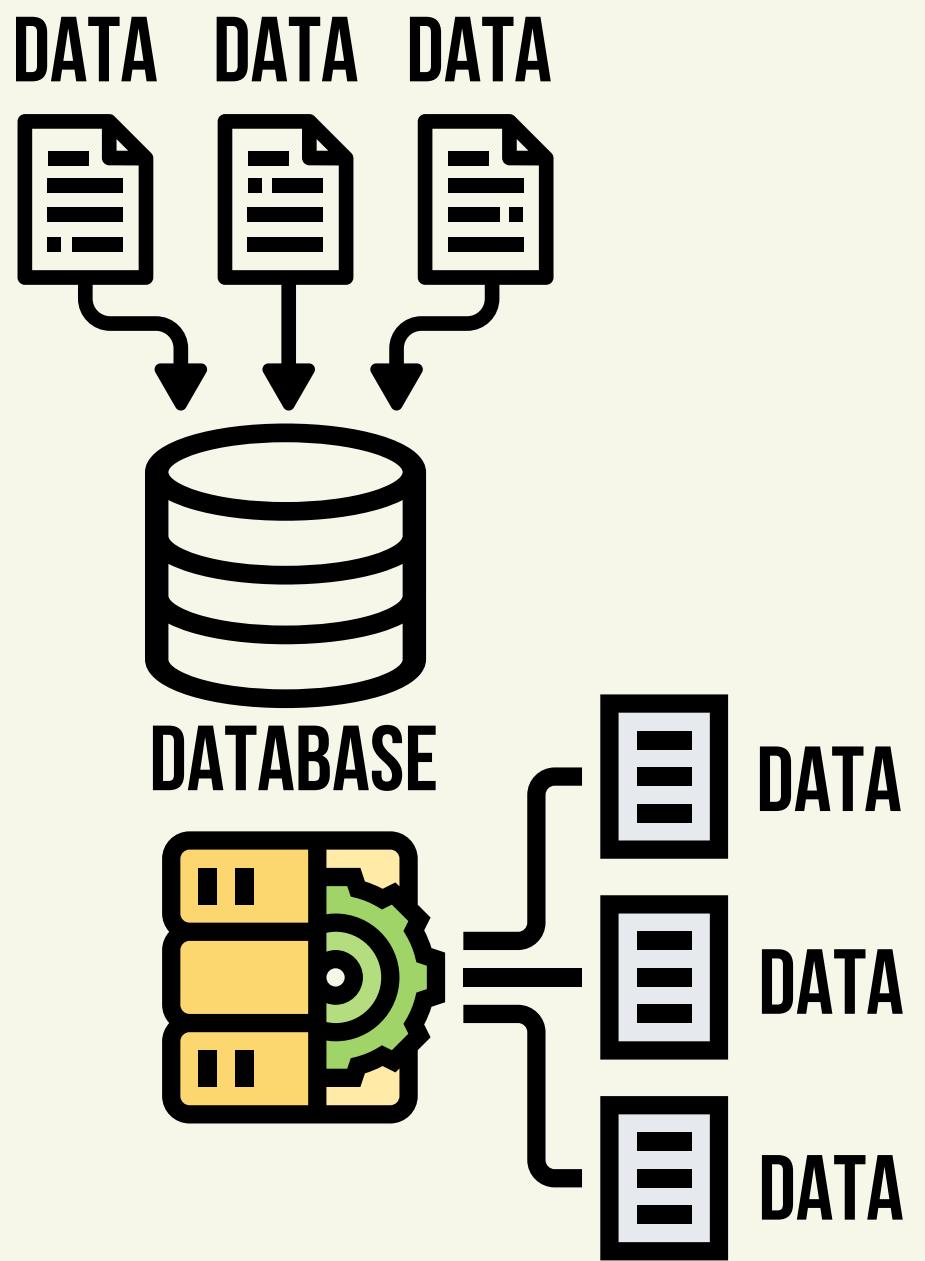
**Basis data** atau dalam bahasa inggris disebut dengan **Database** adalah **Kumpulan Data** yang disusun dengan cara tertentu sehingga mudah diakses dan dikelola.

Tujuan Database :

- Mengorganisir data agar tidak berantakan.
- Memudahkan pencarian informasi tertentu dengan cepat.
- Mengurangi kesalahan dalam pengelolaan data manual.

Mengapa Database Penting?

- Data banyak dan kompleks, perlu tempat penyimpanan yang rapi.
- Akses lebih cepat dibanding cara manual.
- Contoh penting:
  - Aplikasi belanja online untuk menyimpan data produk.
  - Sekolah untuk menyimpan data siswa dan nilai.



# PENGERTIAN BASIS DATA

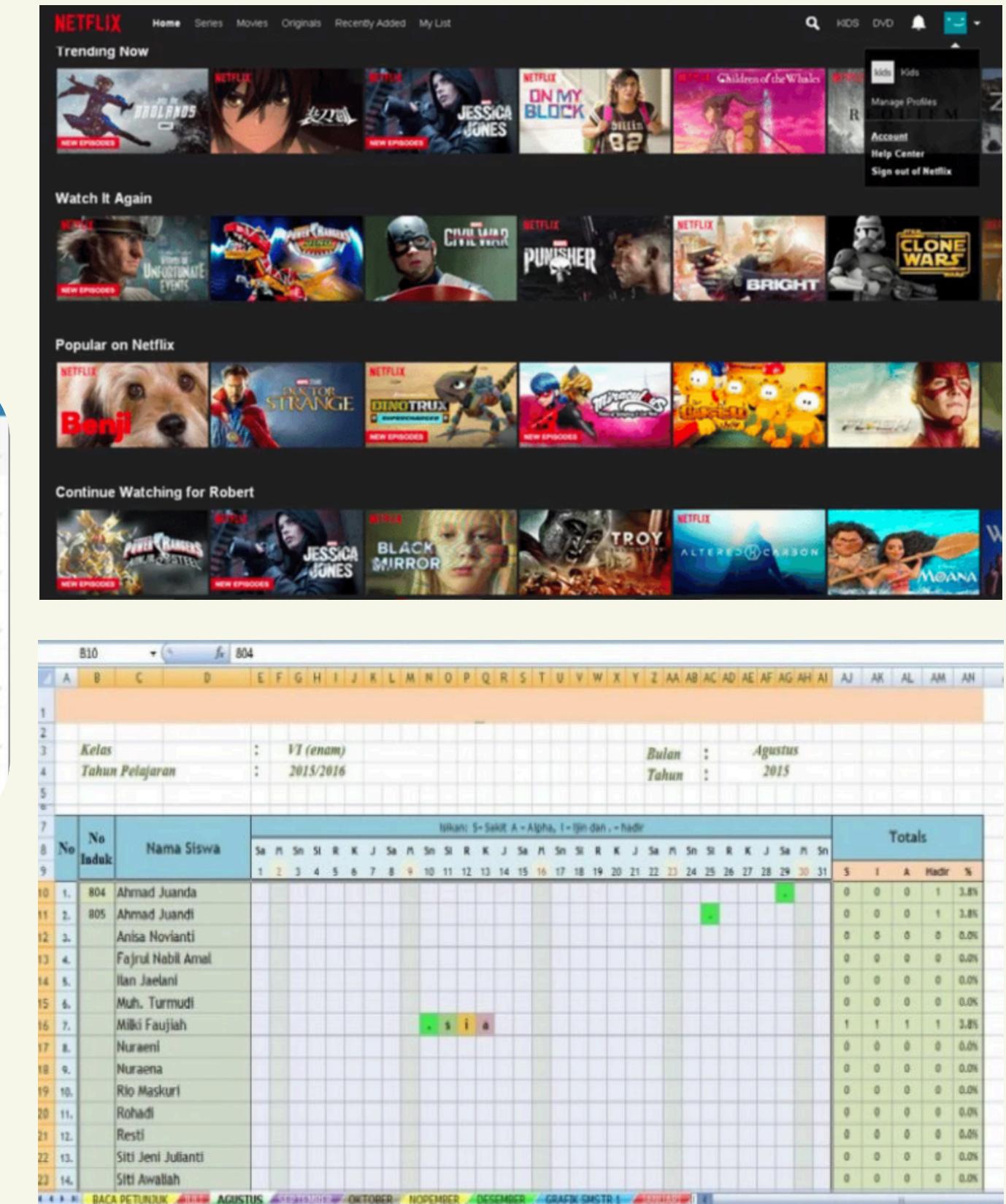
# Contoh Database dalam Kehidupan Sehari-Hari :

- Kontak di HP: Nama, nomor telepon, email.
  - Aplikasi streaming: Daftar film, aktor, dan genre.
  - Sekolah: Data siswa, guru, nilai, absensi.



## Manfaat Database :

- Menghemat waktu (akses cepat).
  - Data lebih aman karena tersimpan digital.
  - Mudah dicari dan difilter sesuai kebutuhan.





# JENIS-JENIS BASIS DATA

# JENIS BASIS DATA BERDASARKAN STRUKTUR

## 1. Basis Data Relasional

- Data tersimpan dalam bentuk tabel yang terstruktur, mirip seperti Microsoft Excel atau Google Sheets.
- Setiap tabel punya baris dan kolom. Kolom itu seperti kategori (contoh: nama, umur), dan baris itu isinya (contoh: "Ali, 16").
- Relasional artinya tabel-tabel ini bisa saling terhubung (relasi). Misalnya, tabel "Siswa" bisa terhubung ke tabel "Nilai."
- Contoh Basis Data Relasional:
  - MySQL: Paling populer untuk aplikasi web.
  - PostgreSQL: Lebih canggih untuk analisis data besar.
  - Oracle Database: Digunakan untuk aplikasi skala besar, seperti perbankan.

### Kelebihan Relasional:

- Data sangat terstruktur, cocok untuk aplikasi yang butuh data rapi.
- Mudah dicari dan dihubungkan antara tabel.

### Kekurangan Relasional:

- Kurang fleksibel kalau data terlalu beragam atau berubah-ubah bentuknya.
- Harus dirancang dulu tabelnya sebelum dipakai.

# JENIS BASIS DATA BERDASARKAN STRUKTUR

## 2. Basis Data Non-Relasional (NoSQL)

Data lebih fleksibel, tidak harus dalam bentuk tabel. Cocok kalau datanya sering berubah bentuk atau sangat beragam.

Biasanya pakai format seperti dokumen (contoh: JSON, XML), graf, atau key-value (mirip kamus).

Contoh Kasus:

- Media sosial (data user, postingan, komentar).
- E-commerce (daftar produk yang terus berubah).

Contoh Basis Data Non-Relasional:

- MongoDB: Cocok untuk aplikasi web modern.
- Firebase: Layanan Google yang sering dipakai di aplikasi mobile.
- Redis: Sering dipakai untuk cache (penyimpanan sementara).

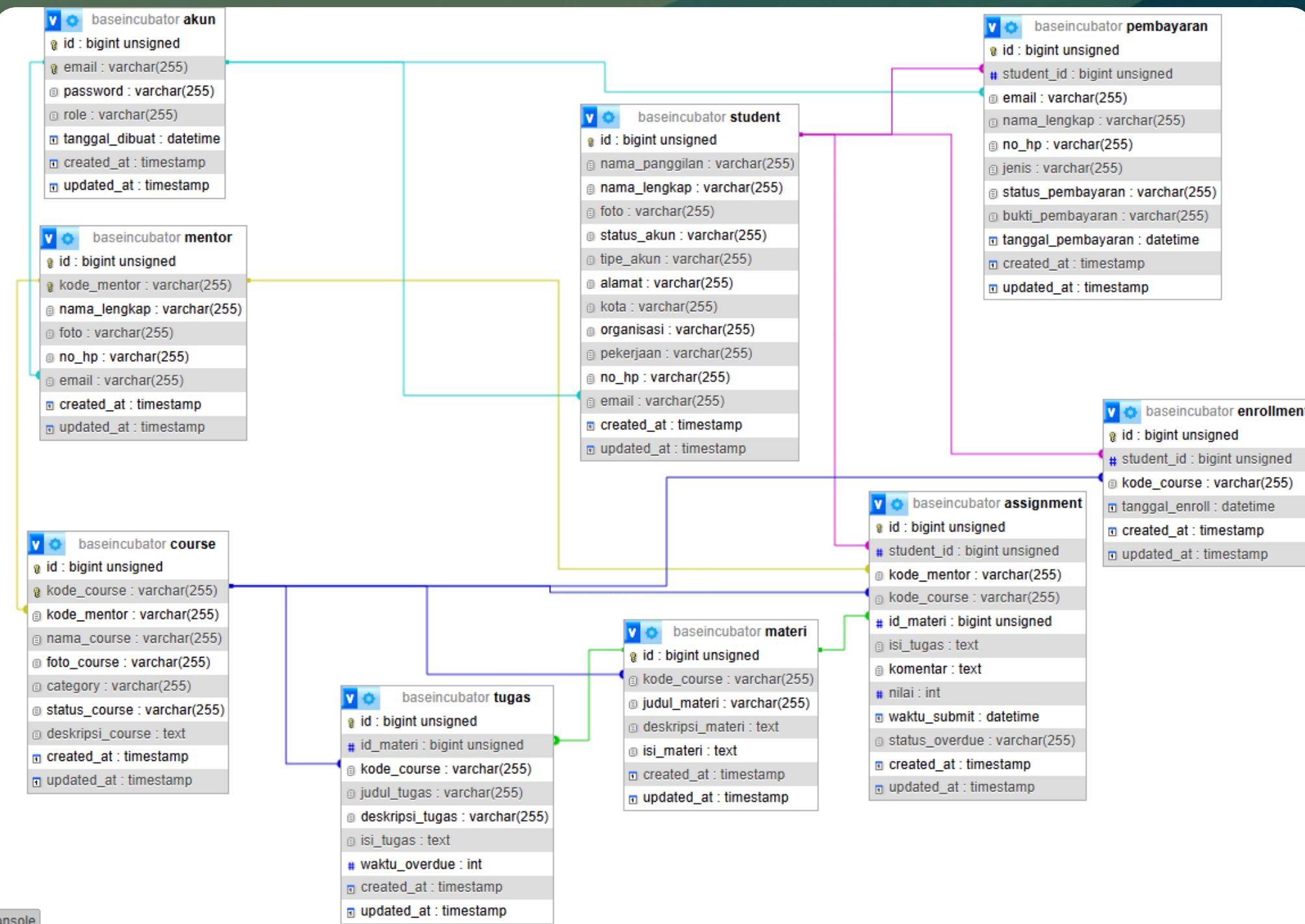
Kelebihan Non-Relasional:

- Fleksibel, tidak perlu bikin tabel.
- Bisa menangani data besar dan beragam (misalnya data gambar, video, atau teks).

Kekurangan Non-Relasional:

- Kurang cocok untuk aplikasi yang butuh data rapi seperti laporan keuangan.
- Hubungan antar data tidak sekuat basis data relasional.

# CONTOH PENERAPAN BASIS DATA RELASIONAL DENGAN MYSQL



Gambar disamping itu contoh dari penerapan database MYSQL, kotak-kotak yang menggambarkan data didalam databasenya dan garis yang terhubung itu adalah relasi antar data.

gampangnya begini, di Database sekolah ada Data Siswa dan Data Nilai. Nah cara biar tau siswa itu punya nilai berapa, antara Data Siswa dan Data Nilai harus memiliki relasi, biasanya relasinya adalah “milik”, jadi bacanya “siswa me-miliki nilai”, dan “nilai di-miliki oleh siswa”

# JENIS BASIS DATA BERDASARKAN LOKASI

## 1. Basis Data Lokal

Disimpan di perangkat tertentu, seperti komputer atau server, jadi hanya bisa diakses oleh perangkat ada Basis Datanya, kalau mau dipakai di perangkat lain Basis Datanya harus dipindah dulu.

Contoh:

- SQLite: Basis data kecil yang langsung ada di aplikasi, sering dipakai di HP atau perangkat IoT.
- Microsoft Access: Untuk aplikasi desktop ringan.

Kelebihan Lokal:

- Tidak perlu internet, cukup jaringan lokal.
- Lebih cepat kalau aplikasinya kecil.

Kekurangan Lokal:

- Data hanya bisa diakses di perangkat tertentu.
- Risiko tinggi kalau perangkat rusak dan tidak ada backup.



gampangnya, database  
lokal itu seperti  
penyimpanan hp/komputer,  
sama juga seperti harddrive,  
kartu sd, usb drive

# JENIS BASIS DATA BERDASARKAN LOKASI

## 2. Basis Data Cloud

Disimpan di server internet (cloud), jadi bisa diakses kapan saja dan dari mana saja.

Contoh:

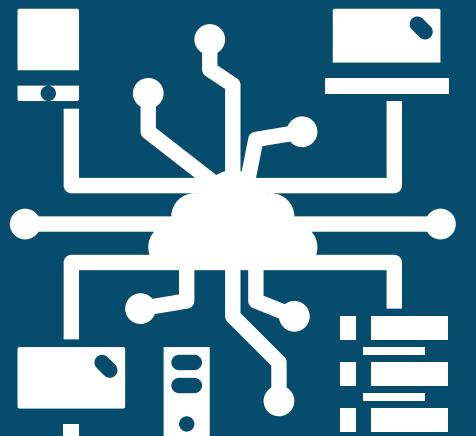
- Firebase: Basis data dari Google, populer untuk aplikasi mobile dan web.
- Amazon RDS: Layanan cloud dari Amazon untuk skala besar.

Kelebihan Cloud:

- Bisa diakses dari banyak perangkat.
- Ada fitur backup otomatis, jadi data lebih aman.

Kekurangan Cloud:

- Butuh koneksi internet yang stabil.
- Ada biaya bulanan/berlangganan, tergantung kapasitasnya.



jadi kalau database cloud itu disimpannya bukan lagi di perangkat namun diupload melalui internet ke server pusat, jadi yang mau akses databasenya ga perlu lagi harus kirim database. hanya perlu menggunakan internet untuk dapat mengakses databasenya

FYI : postingan /video reels maupun tiktok yang kalian upload akan dimasukkan ke server, saat kalian melihatnya itu kalian mengakses database cloud dari aplikasinya namun dibatasi. maka dari itu ada yang namanya kebocoran data

# PERBANDINGAN

## Relasional vs Non-Relasional:

Aspek	Relasional	Non-Relasional
Struktur	Terstruktur (tabel)	Fleksibel (dokumen)
Cocok untuk	Data rapi (keuangan)	Data beragam (media sosial)
Contoh	MySQL, PostgreSQL	MongoDB, Firebase

## Lokal vs Cloud:

Aspek	Lokal	Cloud
Aksesibilitas	Hanya di perangkat	Bisa diakses dari mana saja
Internet	Tidak perlu	Wajib internet
Keamanan Data	Risiko tinggi kalau perangkat rusak	Lebih aman dengan backup otomatis

OKE, PASTI NGANTUK LIHAT PENJELASAN KAYA GITU  
JADI LANGSUNG AJA KITA PRAKTIK YUK !

# PERTAMA KITA INSTALL DULU LARAGON



## Apa itu Laragon?

- Laragon adalah aplikasi yang memudahkan kita membuat website.
- Ibarat sebuah rumah: Laragon adalah rumah untuk website kita, di dalamnya sudah tersedia semua fasilitas yang dibutuhkan.
- Fasilitas di dalam Laragon:
  - Web server (Apache)
  - Database (MySQL)
  - Bahasa pemrograman (PHP)
  - Dan masih banyak lagi



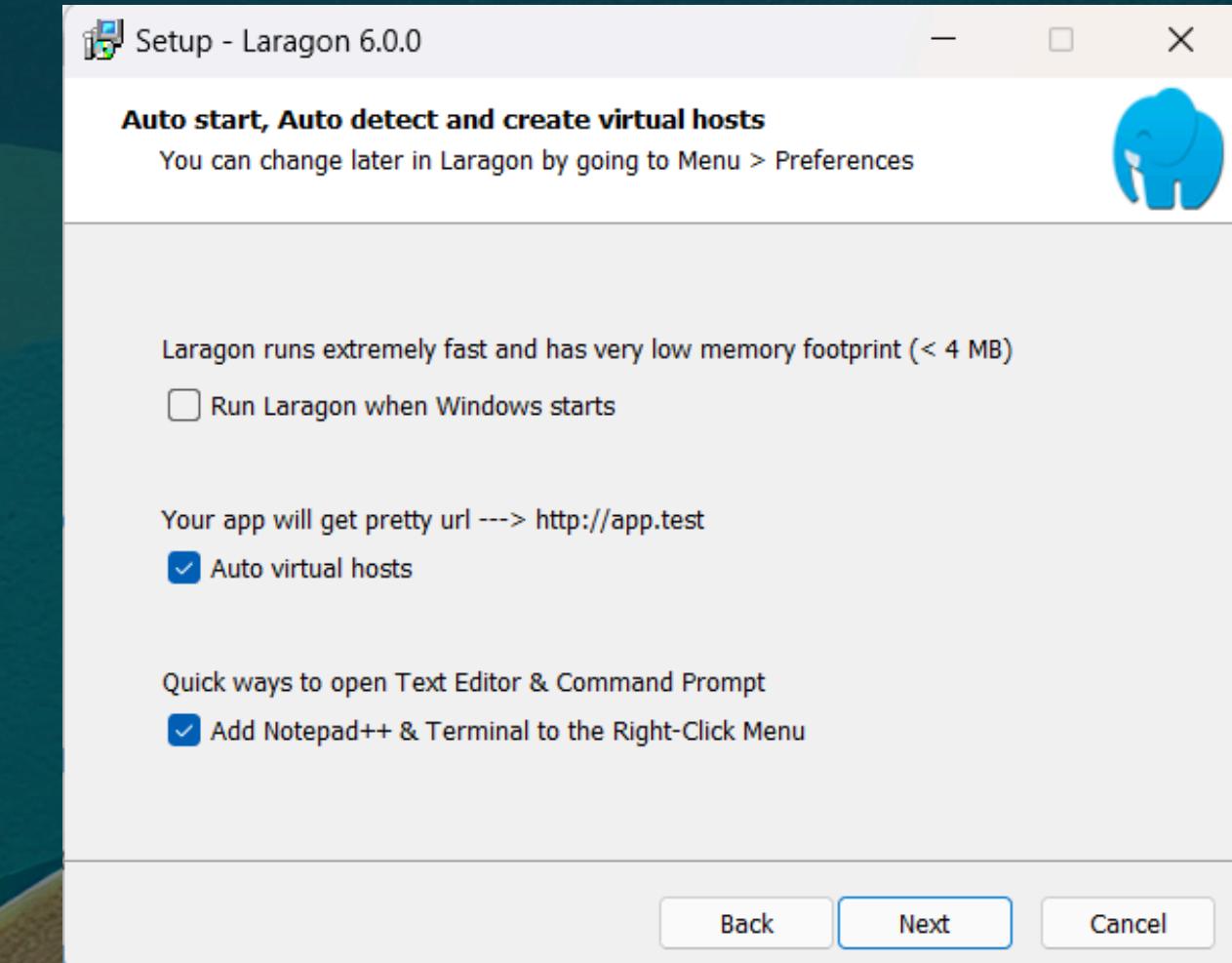
# PERTAMA KITA INSTALL DULU LARAGON

1. [HTTPS://LARAGON.ORG/DOWNLOAD](https://laragon.org/download)

2. [Download Laragon - Full \(173 MB\)](#)

3. SETELAH DOWNLOAD SELESAI, LANGSUNG INSTALL PILIH NEXT SAJA

4. KAMU BISA MATIIN OPSI PALING ATAS AGAR LARAGON GA OTOMATIS TERBUKA SAAT KOMPUTER DINYALAKAN



# MEMBUAT DATABASE

# LANGKAH - LANGKAH MEMBUAT DATABASE

- Buka Laragon, lalu klik  untuk menjalankan servernya

- Setelah mysql berjalan seperti ini :

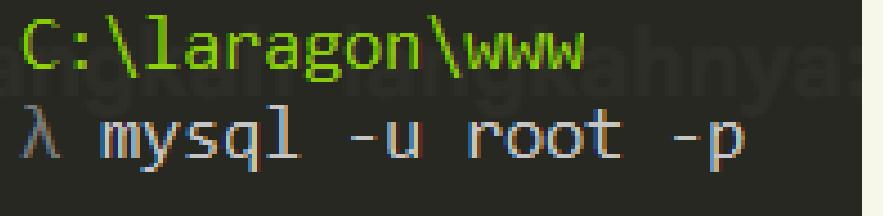


Apache httpd-2.4.54-win64-VS16 started 80 Reload  
MySQL mysql-8.0.30-winx64 started 3306

- Klik tombol  untuk membuka terminal laragon

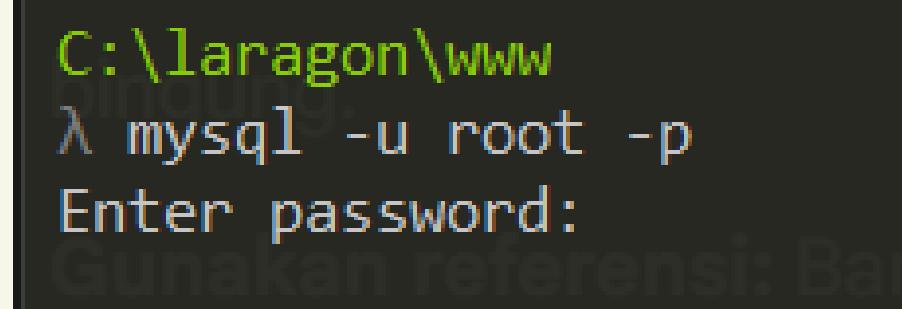
- Sekarang kita akan coba buat database dan tabel langsung dari terminal. Jadi lebih terasa kayak hacker beneran!

- Akses MySQL: Kita perlu masuk ke dalam MySQL. Caranya, ketik perintah “mysql -u root -p” di terminal :



```
C:\laragon\www
λ mysql -u root -p
```

- Saat diminta memasukkan password, langsung ENTER saja, karena setelan awal memang tidak memiliki password



```
C:\laragon\www
λ mysql -u root -p
Enter password:
Gunakan referensi: Ba
```

# LANGKAH - LANGKAH MEMBUAT DATABASE

- Setelah muncul pesan welcome dan input berubah menjadi mysql>, tandanya kamu telah berhasil masuk ke mysql, maka akan muncul tampilan

```
C:\laragon\www
mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.30 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners. Sudah mulai paham kan, serunya belajar database?

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

mysql> |
```

- Sekarang kita akan mencoba membuat database Toko Online, jalankan kode syntax “CREATE DATABASE (nama database);” :

```
CREATE DATABASE toko_online;
```

- Jika berhasil maka akan muncul notifikasi seperti ini :

```
mysql> CREATE DATABASE toko_online;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> |
```

FYI : Query OK merupakan tanda bahwa kode yang kita gunakan sudah benar  
OIYA !!! kode syntax query hanya berlaku saat tanda ; diakhir kode diberikan sebagai pengakhir  
sama seperti </> di HTML

Nah, untuk menampilkan database yang sudah kita buat bisa menggunakan kode “SHOW databases;” :

**SHOW DATABASES;**

maka akan muncul tampilan dari database yang kalian miliki seperti ini :

sekarang coba untuk hapus database yang sudah kita buat tadi dengan menggunakan kode “DROP DATABASE (nama database);”

**DROP DATABASE nama\_database;**

(ubah nama\_database menjadi toko\_online)

coba tampilkan lagi database yang kalian miliki, kalau toko\_online sudah gaada, berarti berhasil menghapus

sekarang coba buat lagi database toko\_online

kalau kalian mencoba membuat database yang sudah ada atau dengan nama yang sama, maka akan muncul error karena database tidak bisa memiliki nama yang sama

```
mysql> CREATE DATABASE toko_online;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> CREATE DATABASE toko_online;
ERROR 1007 (HY000): Can't create database 'toko_online'; database exists
mysql>
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| base      |
| baseincubator |
| bendishub  |
| incubator   |
| information_schema |
| mysql      |
| performance_schema |
| sys        |
| toko_online |
+-----+
9 rows in set (0.01 sec)
```

# MENGISI DATABASE

Buat database udah, sekarang kita isi dengan data, atau biasa disebut dengan table. Sebelum membuat table kita perlu masuk ke database yang akan kita isi dengan menggunakan kode “USE (nama database);”

```
mysql> USE toko_online;  
Database changed
```

selanjutnya, kita buat table baru, dengan kode syntax “CREATE TABLE (nama table);” lalu atribut dari tablenya, kamu bisa ikuti contoh disamping.

#### Penjelasan:

- `id_produk` : Kolom untuk menyimpan ID produk secara otomatis.
- `nama_produk` : Kolom untuk menyimpan nama produk, maksimal 100 karakter.
- `harga` : Kolom untuk menyimpan harga produk, tipe data integer.
- `deskripsi` : Kolom untuk menyimpan deskripsi produk, tipe data text (ukuran fleksibel).

```
CREATE TABLE produk (  
    id_produk INT PRIMARY KEY AUTO_INCREMENT,  
    nama_produk VARCHAR(100),  
    harga INT,  
    deskripsi TEXT  
);
```

FYI !! disetiap table harus memiliki minimal dan maximal 1 Primary KEY sebagai identitas table, atau biasa disebut ID. gampangnya seperti NISN atau No.Absen, fungsinya biar data didalam table gaboleh punya ID yang sama

Jika table berhasil dibuat maka akan muncul notifikasi Query OK. nah sekarang kita akan menambahkan atribut pada table yang sudah kita buat.

dengan menggunakan kode “ALTER TABLE (nama table) ADD (nama atribut) (tipe data);” kalian bisa mengikuti contoh berikut

```
ALTER TABLE produk ADD kategori VARCHAR(50);
```

untuk mengubah tipe data dari atribut suatu table, kalian bisa menggunakan kode “ALTER TABLE (nama table) MODIFY (nama atribut) (tipe data baru);” kalian bisa mengikuti contoh dibawah

```
ALTER TABLE produk MODIFY kategori CHAR(10);
```

nah, untuk mengubah nama dari atribut suatu table, kalian bisa menggunakan kode “ALTER TABLE (nama table) CHANGE (nama lama) (nama baru) (tipe data);” kalian bisa mengikuti contoh berikut

```
ALTER TABLE produk CHANGE id_produk id INT;
```

FYI !! nama database, nama table, dan nama atribut gabisa pakai spasi dan symbol, jadi kalau mau pakai spasi digantikan dengan garis bawah

Jika table berhasil dibuat maka akan muncul notifikasi Query OK. nah sekarang kita akan menambahkan atribut pada table yang sudah kita buat.

dengan menggunakan kode “ALTER TABLE (nama table) ADD (nama atribut) (tipe data);” kalian bisa mengikuti contoh berikut

```
ALTER TABLE produk ADD kategori VARCHAR(50);
```

sekarang mencoba mengganti nama table, menggunakan kode

“RENAME TABLE (nama lama) TO (nama baru);”

```
RENAME TABLE produk TO barang;
```

silahkan coba buat table baru dengan mengikuti ketentuan sebelumnya

lalu coba menghapus table tersebut dengan kode “DROP TABLE (nama table);”

```
DROP TABLE kelas;
```

jika kalian ingin menghapus database kalian bisa menggunakan kode “DROP DATABASE (nama database);”

```
DROP DATABASE latihan_db;
```

# YANG KALIAN PELAJARI TADI ADALAH SYNTAX QUERY DDL

## 1. Data Definition Language (DDL)

DDL digunakan untuk membuat dan mengatur struktur database.

Syntax	Penjelasan
CREATE DATABASE	Membuat database baru.
DROP DATABASE	Menghapus database beserta semua tabelnya.
USE	Memilih database yang akan digunakan.
CREATE TABLE	Membuat tabel baru dalam database.
ALTER TABLE	Mengubah struktur tabel (misalnya menambah kolom atau mengubah tipe data).
DROP TABLE	Menghapus tabel beserta semua datanya.
TRUNCATE TABLE	Menghapus semua data dalam tabel tanpa menghapus struktur tabel.

# LATIHAN PRAKTIKUM DDL

[HTTPS://DRIVE.GOOGLE.COM/FILE/D/1SIGZ02INU29SWH5DIEACD76362062EX6/VIEW?  
USP=DRIVE\\_LINK](https://drive.google.com/file/d/1SIGZ02INU29SWH5DIEACD76362062Ex6/view?usp=drive_link)

**TIPS :**

**KODE UNTUK KELUAR DARI MYSQL ADALAH “EXIT;”**

**SETELAH KELUAR DARI MYSQL KAMU DAPAT MENUTUP TERMINAL DENGAN KODE “EXIT .”**

**SELALU PERHATIKAN KODINGANMU, JANGAN SAMPAI ADA TYPO ATAUPUN SALAH SYMBOL**

# MENGELOLA DATABASE

# DML & DQL: MENGELOLA DATA DI DATABASE

Sebelumnya kita sudah belajar dasar database dan juga DDL (Data Definition Language), yaitu bagian SQL yang berhubungan dengan pembuatan struktur tabel. Nah, sekarang kita akan masuk ke DML & DQL, yaitu bagian yang berhubungan dengan manipulasi dan pengambilan data.

Bayangkan kita punya sebuah database untuk aplikasi toko online, dan sekarang kita mau memasukkan, mengedit, menghapus, serta mengambil data dari sana. Itulah fungsi DML & DQL!

## 1. Data Manipulation Language (DML) – Untuk Mengolah Data

DML digunakan untuk memasukkan, mengupdate, menghapus, dan mengambil data dalam tabel.

Berikut adalah query-query utama dalam DML:

### ◆ INSERT INTO – Menambahkan Data

Gunakan query ini untuk menambahkan data baru ke dalam tabel.

📌 Contoh: Menambahkan data ke tabel produk.

```
INSERT INTO produk (id, nama, harga, stok)
VALUES (1, 'Laptop Asus', 8000000, 10);
```

### 💡 Penjelasan:

- Kita memasukkan data produk dengan id = 1, nama = Laptop Asus, harga = 8 juta, dan stok = 10 unit.

## ◆ UPDATE – Memperbarui Data

Kalau ada data yang salah atau perlu diubah, gunakan UPDATE.

📌 Contoh: Mengupdate harga Laptop Asus di tabel produk.

```
UPDATE produk  
SET harga = 7500000  
WHERE id = 1;
```



Penjelasan:

- Kita memperbarui harga laptop dengan id = 1 menjadi 7,5 juta

## ◆ DELETE – Menghapus Data

Kalau ada data yang tidak diperlukan, kita bisa menghapusnya dengan DELETE.

📌 Contoh: Menghapus data Laptop Asus dari tabel produk.

```
DELETE FROM produk  
WHERE id = 1;
```



Penjelasan:

- Data dengan id = 1 akan dihapus dari tabel produk.

## 2. Data Query Language (DQL) – Untuk Membaca Data

DQL adalah subset dari DML yang fokusnya hanya untuk membaca data.

### ◆ SELECT – Mengambil Data

Query ini digunakan untuk menampilkan data dari tabel.

📌 Contoh: Menampilkan semua data dari tabel produk.

```
SELECT * FROM produk;
```



Penjelasan:

- \* berarti mengambil semua kolom yang ada di tabel produk.

### ◆ WHERE – Menentukan Kondisi

Gunakan WHERE kalau mau menampilkan data tertentu saja.

📌 Contoh: Menampilkan produk yang harganya di atas 5 juta.

```
SELECT * FROM produk  
WHERE harga > 5000000;
```



Penjelasan:

- Hanya produk dengan harga lebih dari 5 juta yang akan ditampilkan.

## ◆ ORDER BY – Mengurutkan Data

Kalau mau data tampil secara berurutan, gunakan ORDER BY.

📌 Contoh: Menampilkan produk dari harga termurah ke termahal.

```
SELECT * FROM produk  
ORDER BY harga ASC;
```

💡 Penjelasan:

- ASC (Ascending) = Urutan dari yang termurah ke termahal.
- DESC (Descending) = Urutan dari yang termahal ke termurah.

## ◆ GROUP BY – Mengelompokkan Data

Kalau mau mengelompokkan data berdasarkan kolom tertentu, gunakan GROUP BY.

📌 Contoh: Menghitung jumlah produk berdasarkan kategori.

```
SELECT kategori, COUNT(*) as total_produk  
FROM produk  
GROUP BY kategori;
```

💡 Penjelasan:

- Mengelompokkan data berdasarkan kategori dan menghitung jumlah produknya.

## ◆ HAVING – Filter Setelah Pengelompokan

Kalau WHERE digunakan sebelum GROUP BY, HAVING digunakan setelah GROUP BY.

📌 Contoh: Menampilkan kategori yang memiliki lebih dari 5 produk.

```
SELECT kategori, COUNT(*) as total_produk  
FROM produk  
GROUP BY kategori  
HAVING COUNT(*) > 5;
```

### 💡 Penjelasan:

- Menampilkan hanya kategori yang memiliki lebih dari 5 produk.

## ◆ LIMIT – Membatasi Jumlah Data

Gunakan LIMIT kalau mau membatasi jumlah data yang ditampilkan.

📌 Contoh: Menampilkan 3 produk pertama yang ada di tabel.

```
SELECT * FROM produk  
LIMIT 3;
```

### 💡 Penjelasan:

- Hanya akan menampilkan 3 data pertama dari tabel produk.

## 2. Data Manipulation Language (DML)

DML digunakan untuk mengolah data dalam tabel (memasukkan, memperbarui, menghapus, dan mengambil data).

Syntax	Penjelasan
INSERT INTO	Menambahkan data baru ke dalam tabel.
SELECT	Mengambil data dari tabel.
UPDATE	Memperbarui data dalam tabel.
DELETE	Menghapus data dalam tabel berdasarkan kondisi tertentu.

## 3. Data Query Language (DQL)

DQL adalah subset dari DML yang hanya digunakan untuk membaca data.

Syntax	Penjelasan
SELECT	Mengambil data dari tabel.
WHERE	Menentukan kondisi untuk memfilter data yang diambil.
ORDER BY	Mengurutkan data berdasarkan kolom tertentu (ASC/DESC).
GROUP BY	Mengelompokkan data berdasarkan nilai tertentu.
HAVING	Menyaring data setelah dilakukan pengelompokan.
LIMIT	Membatasi jumlah data yang diambil.

# LATIHAN PRAKTIKUM DML & DQL

[HTTPS://DRIVE.GOOGLE.COM/FILE/D/104WMEUPU415R9LCIU5GG\\_JPW7ICRH\\_I/VIEW?  
USP=DRIVE\\_LINK](https://drive.google.com/file/d/104WMEUPU415R9LCIU5GG_JPW7ICRH_I/view?usp=drive_link)



**COMING SOON..**