

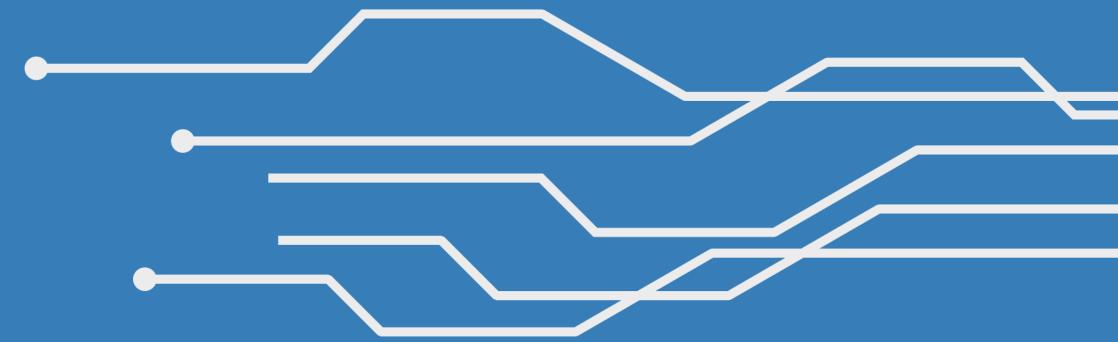
Basic Python Programming

Database, Koneksi SQLite

Advanced - Pertemuan 10, 11



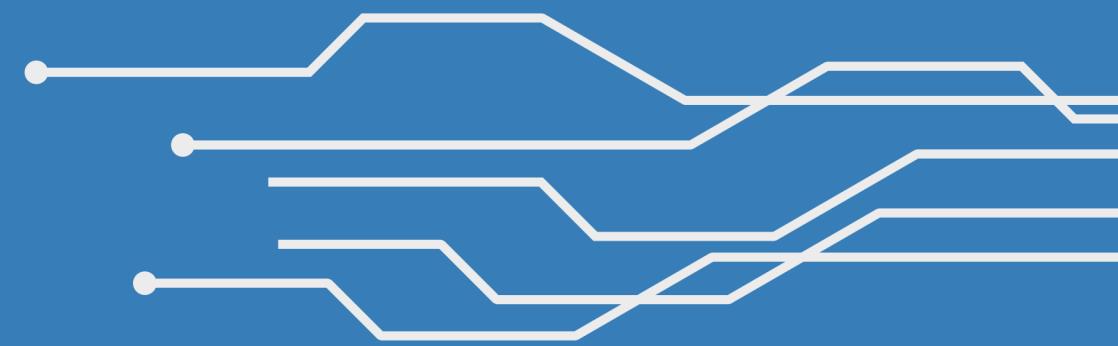
Roadmap



- Database
- SQL
- Perintah Dasar SQL
- Latihan Koneksi Database SQLite
- Latihan

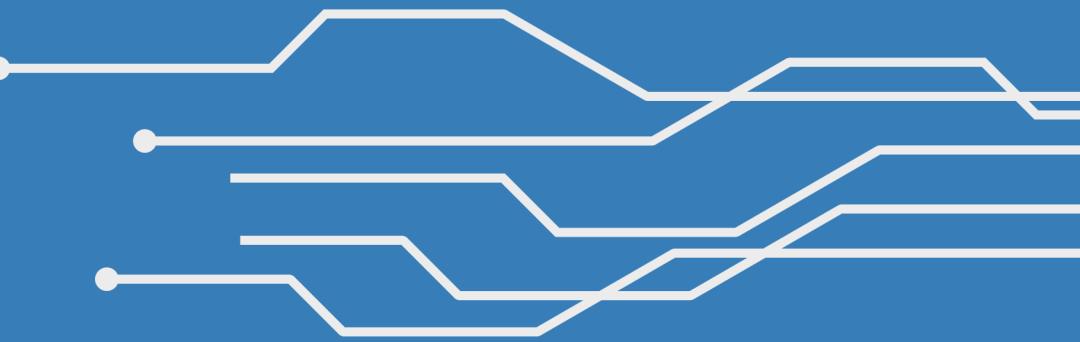
Database

Materi



- Apa itu Database?
- Komponen Database
- Fungsi Database?
- Jenis-jenis Database

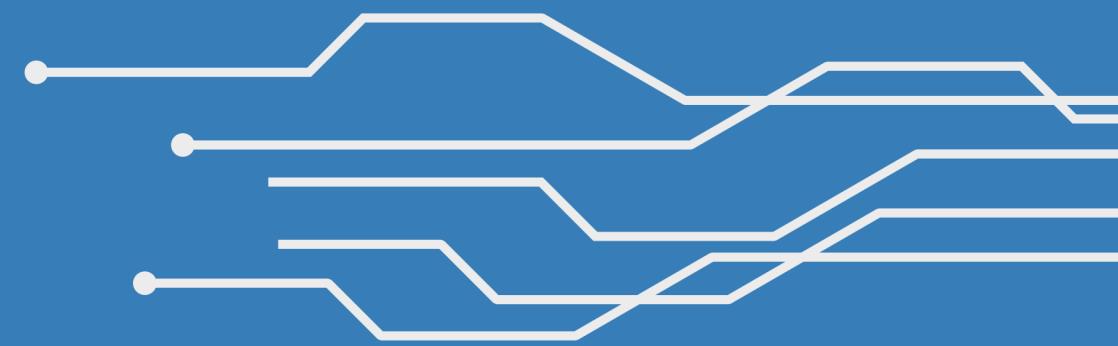
Apa itu Database?



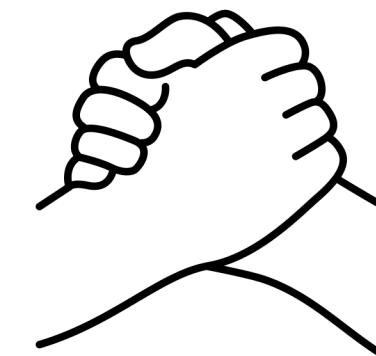
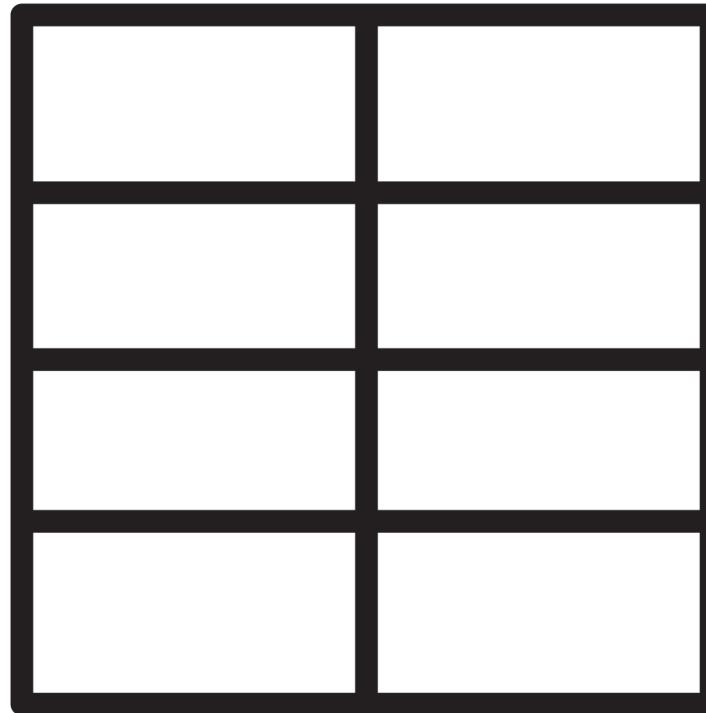
- **Database** adalah **kumpulan data yang disimpan dengan sistem tertentu**, dan saling berhubungan, sehingga dapat dikelola dengan mudah.
- Tujuan: Untuk **menyimpan dan mengelola data** yang dapat diakses dan dimanfaatkan oleh pengguna atau aplikasi.



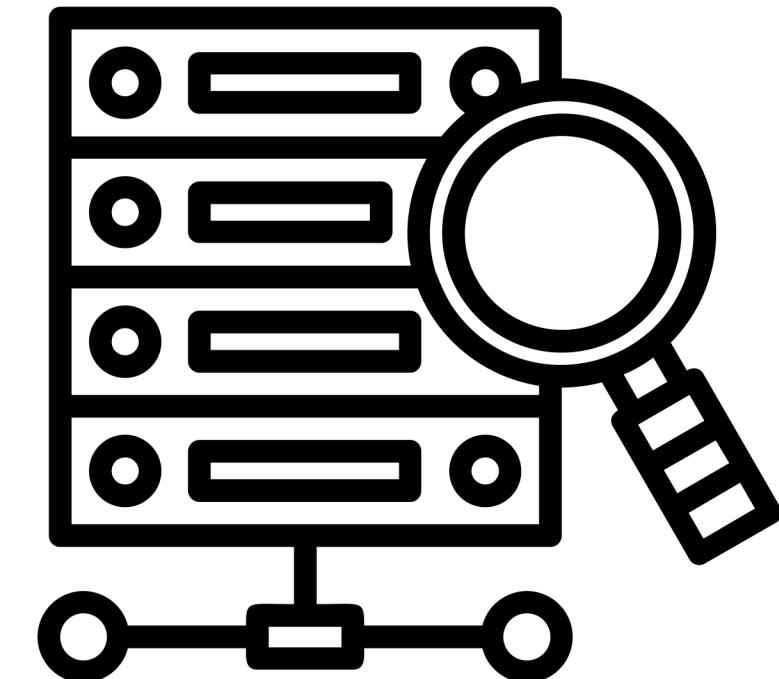
Komponen Database



Table



Query



Komponen Database

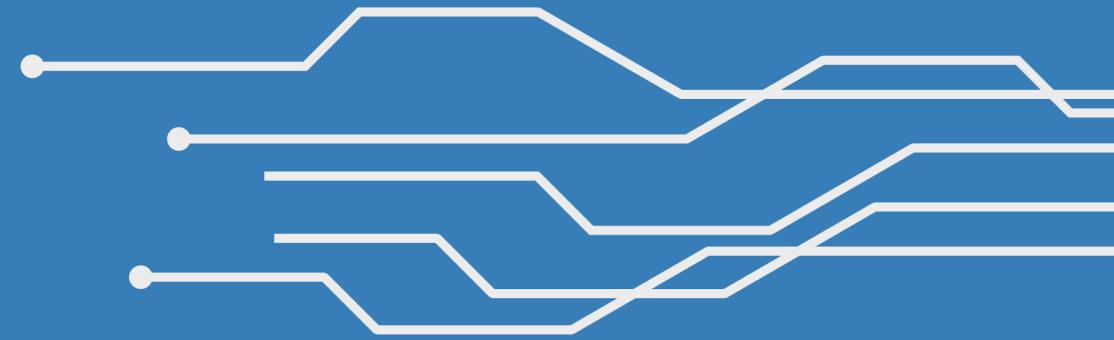


Table - merupakan kumpulan data atau informasi yang diorganisasikan dalam bentuk baris(record) dan kolom (field)

Komponen Database

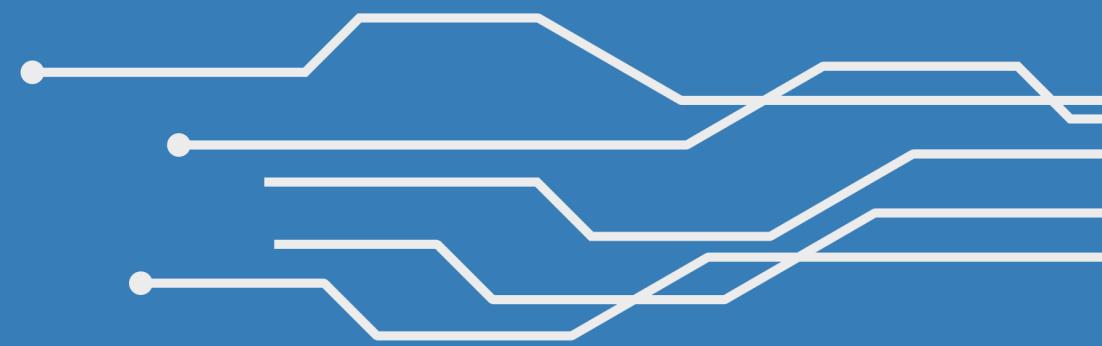


Diagram illustrating the components of a database using a table structure:

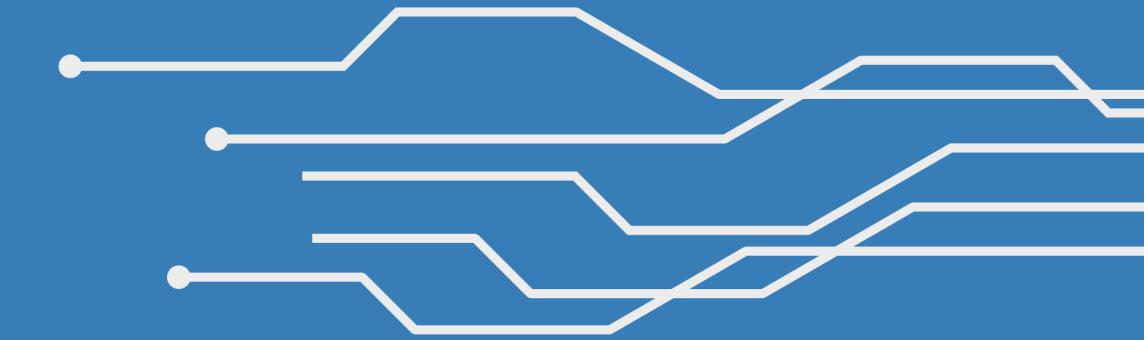
The table has four columns labeled **ID**, **NAMA**, **TELP**, and **ALAMAT**. There are three rows of data, each representing a **Record**. The first column (**ID**) contains values 1, 2, and 3. The second column (**NAMA**) contains names: Fillah Nur Fauzi, Renanda Alif, and M. Fauzan. The third column (**TELP**) contains the same phone number: 0812312341234. The fourth column (**ALAMAT**) contains the same address: Jl. Yogyakarta 123.

Annotations on the right side of the table:

- A curved arrow points from the word **Field** to the header of the **ALAMAT** column.
- A curved arrow points from the word **Record** to the first data row (ID 1).

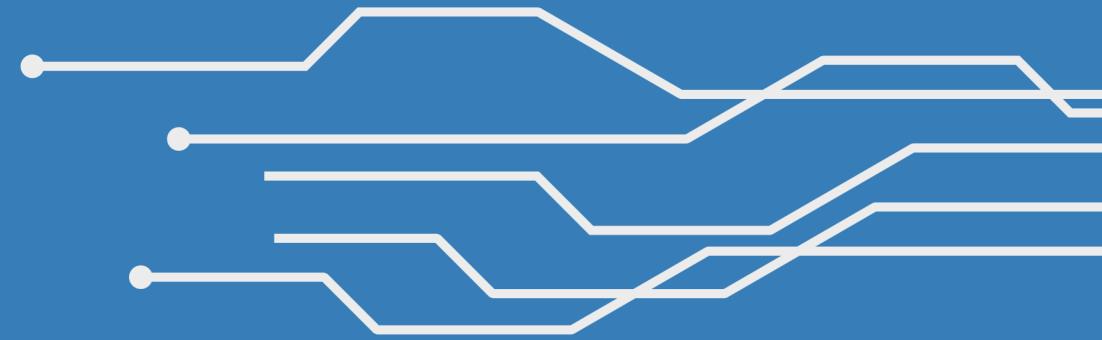
ID	NAMA	TELP	ALAMAT
1	Fillah Nur Fauzi	0812312341234	Jl. Yogyakarta 123
2	Renanda Alif	0812312341234	Jl. Yogyakarta 123
3	M. Fauzan	0812312341234	Jl. Yogyakarta 123

Istilah Penting(Dasar)

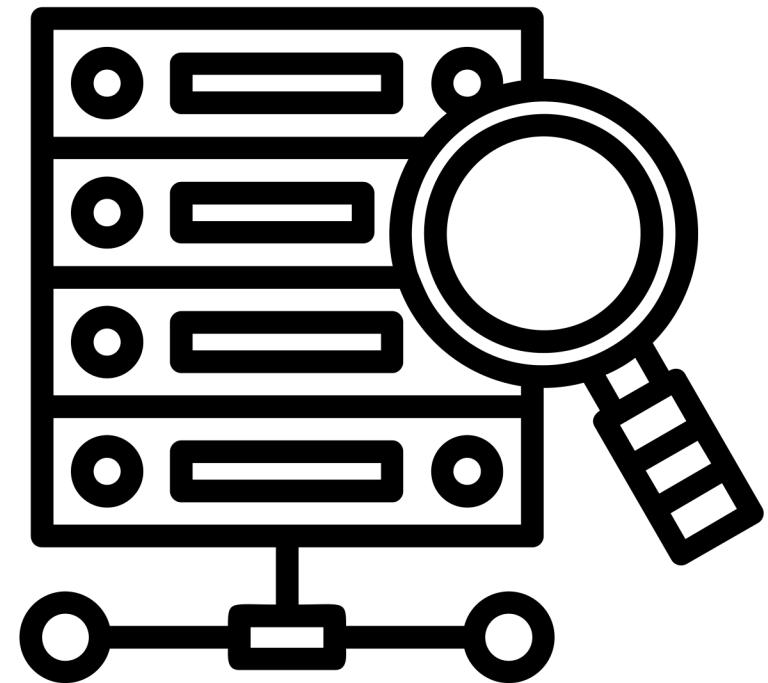


- **Data Type:** INT, TEXT, VARCHAR, CHAR, DOUBLE, DATE TIME.
- **Primary Key:** untuk field yang memiliki kunci unik didalam table.
- **Auto Increment:** fungsi data type numerik untuk membuat nilai yang berurutan (increment).

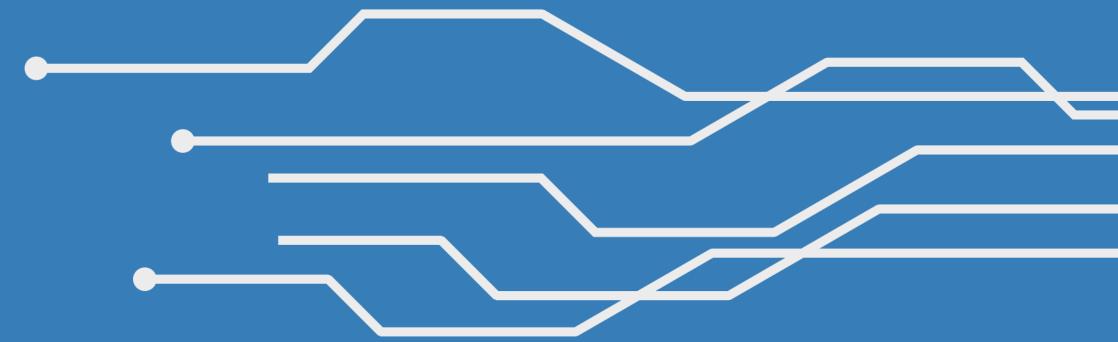
Query



Query merupakan syntax perintah digunakan untuk mengakses dan menampilkan data pada sistem database.



Jenis-jenis Query

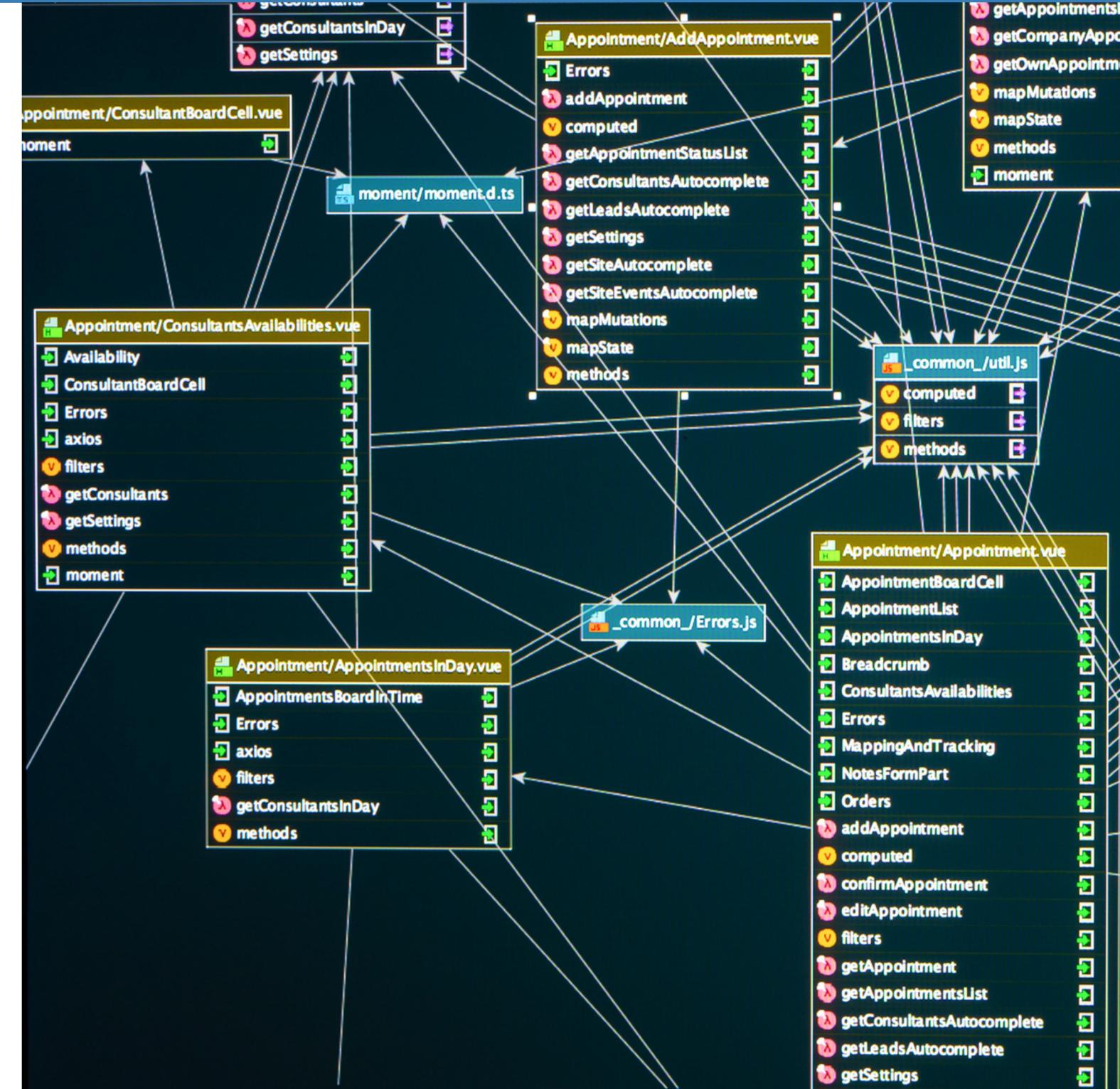


- Data Definition Languange(DDL)
- Data Manipulation Languange (DML)
- Data Control Languange (DCL)

Data Definition Language

DDL digunakan untuk mendefinisikan data dalam database.

- Membuat table baru.
- Membuat database baru.
- Menghapus table dan database.
- Mengubah struktur database.



Data Manipulation Language

DML digunakan untuk memanipulasi isi dalam database.

- Menambah data record table
- Mengubah data record table
- Menghapus data record table



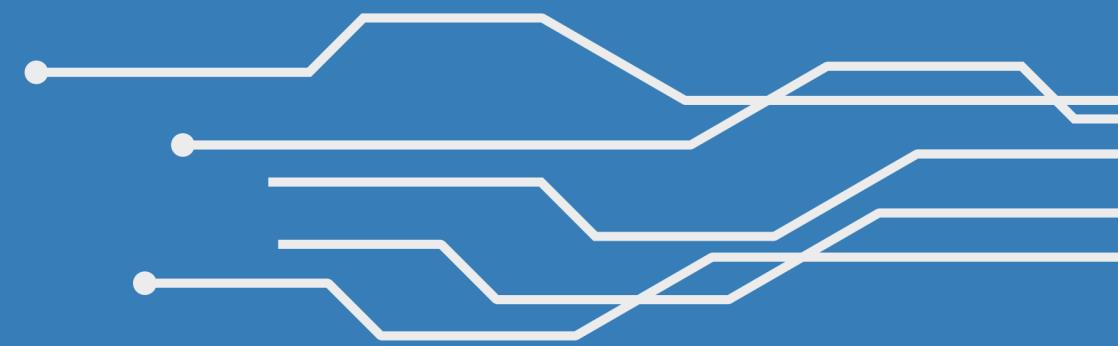
Data Manipulation Language

DCL digunakan untuk mengatur hak akses user terhadap database.

- Mengizinkan user akses database
- Membatalkan izin akses.
- Menetapkan penyimpanan database.

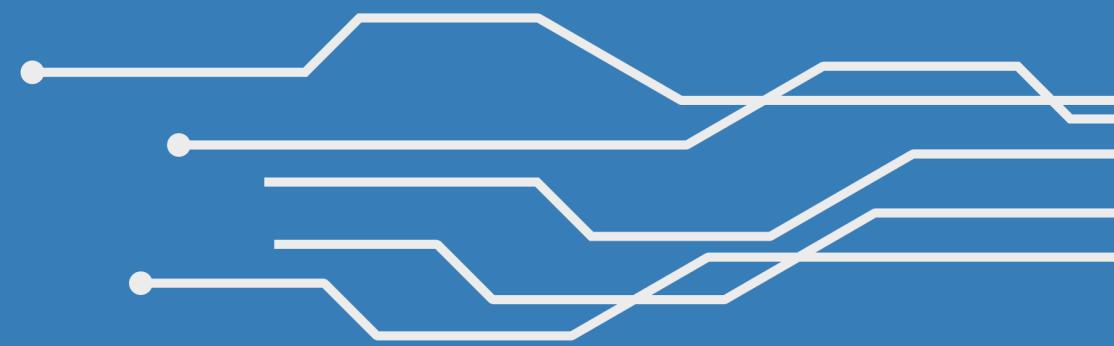


Fungsi Database



- Mempercepat dan Mempermudah Identifikasi Data
- Mengontrol Data Secara Terpusat
- Menghindari Duplikasi Data
- Menyimpan Data dengan Lebih Aman
- Menghemat Biaya
- Dapat Diakses Multi-User

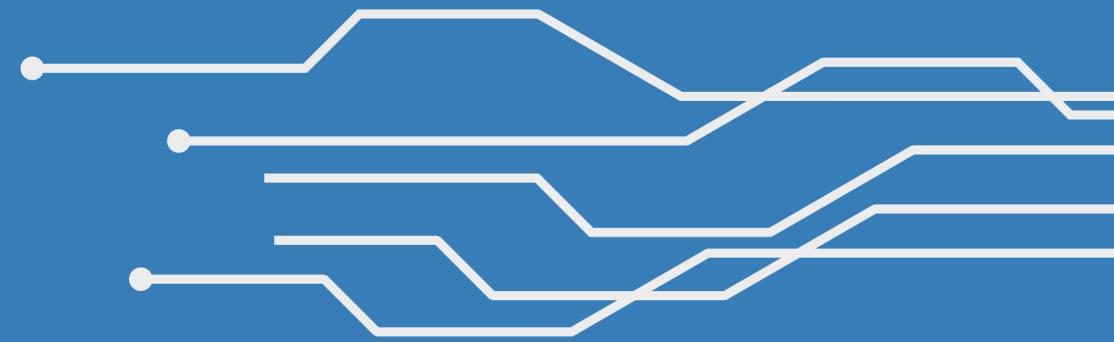
Jenis-jenis Database



- Relational Database
- Not Relational – Not Only SQL (NoSQL)

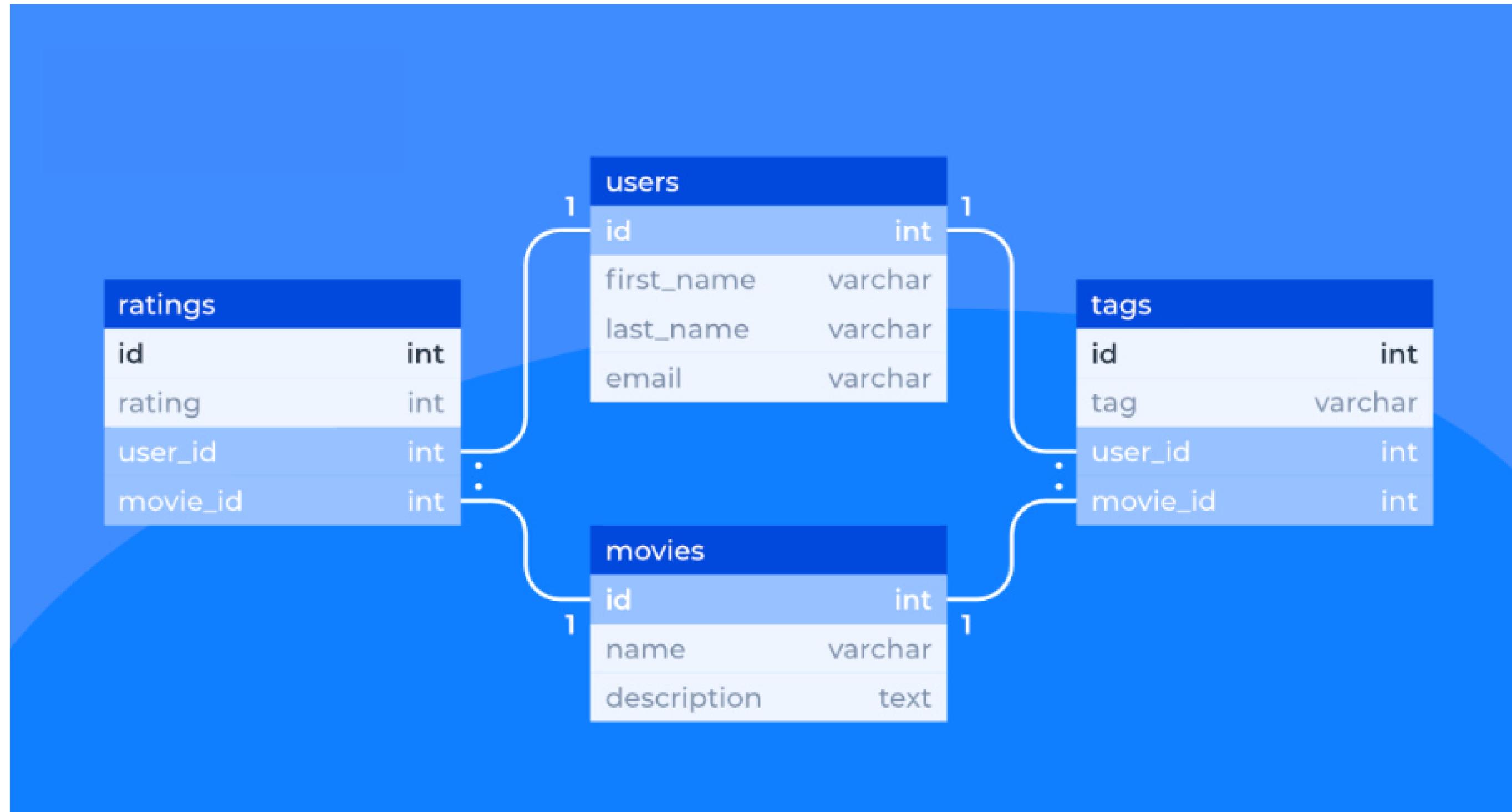
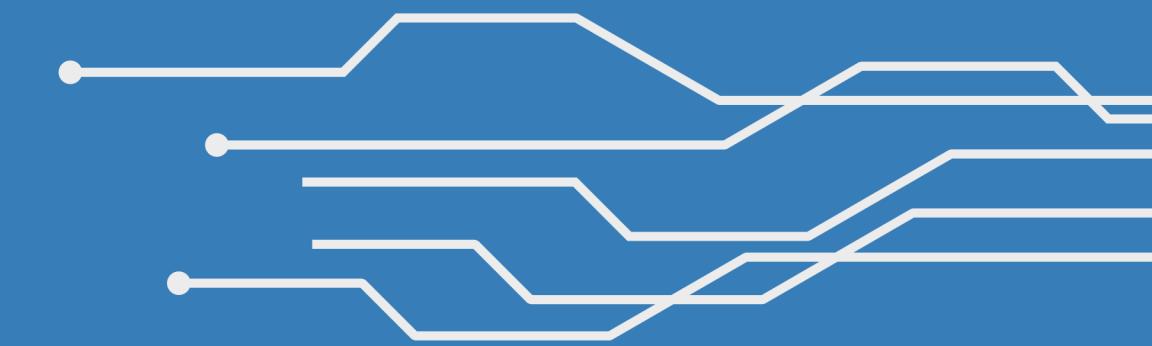


Relational Database

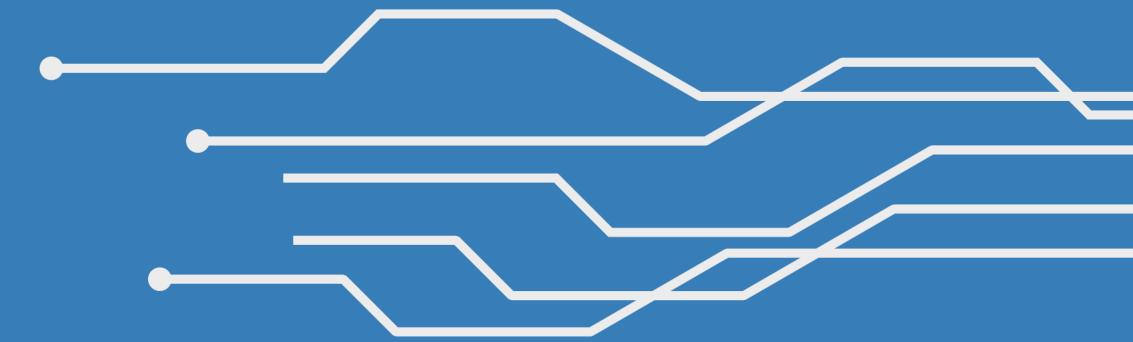


- Jenis database ini paling sering digunakan. Dinamakan relational, karena data disimpan dalam beberapa **tabel yang saling terkait atau berhubungan (membentuk relasi)**.

Relational Database



NoSQL

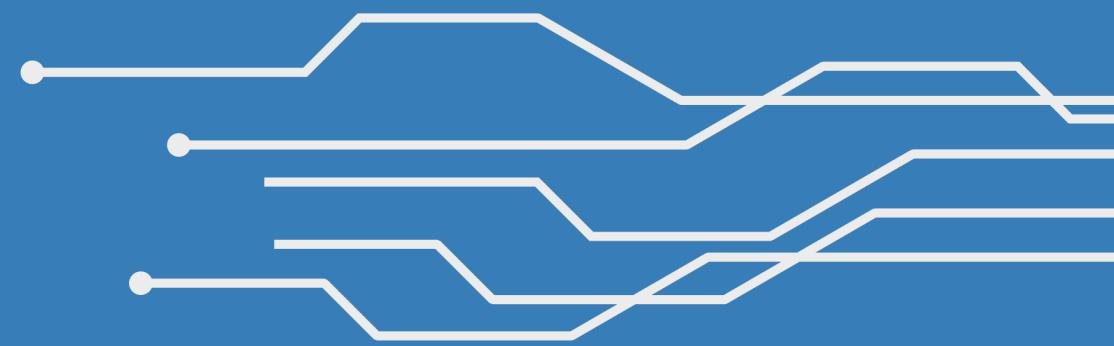


- **NoSQL** atau Not Only SQL adalah database non-relational. Artinya, data yang dapat **tersimpan bukan hanya berbentuk tabel**, melainkan juga data tipe lainnya.

SQL

(Standard Query Languange)

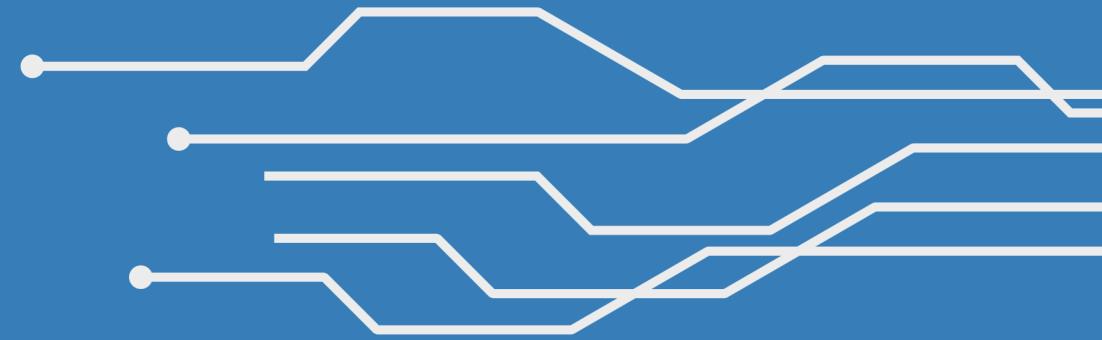
Materi



- Apa itu SQL?
- Fungsi SQL
- Perintah-perintah Dasar SQL



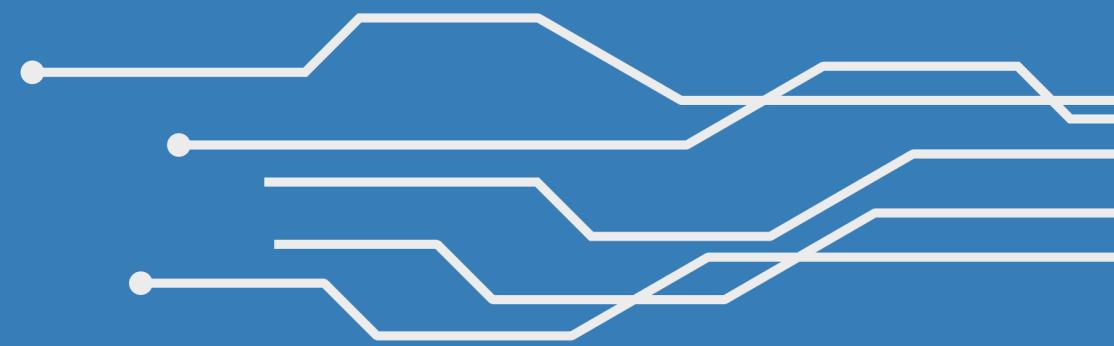
Apa itu SQL



- SQL adalah **bahasa pemrograman** yang memberikan perintah untuk mengakses dan mengolah data berbasis relasional pada sebuah sistem database.
- Jadi secara singkat, dapat diartikan bahwa SQL adalah **Relational Database Management System (RDBMS)**.



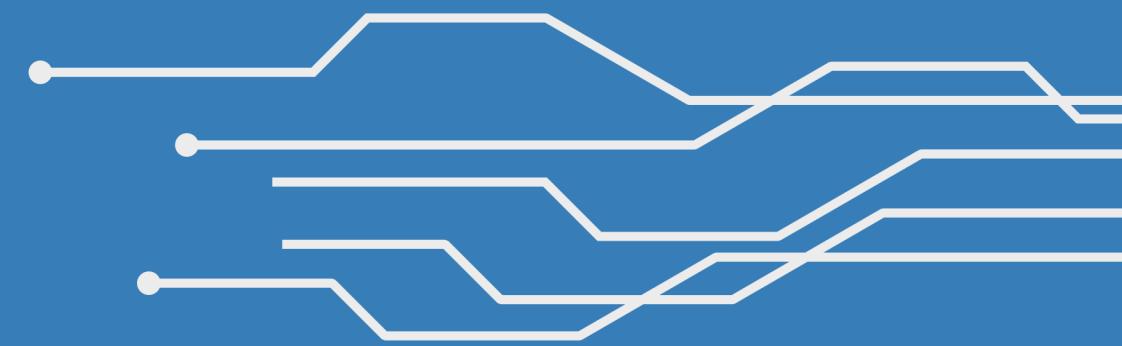
Fungsi SQL



- Mengeksekusi Query Database
- Mengakses dan Memanipulasi data
- Mengatur Hak Akses User



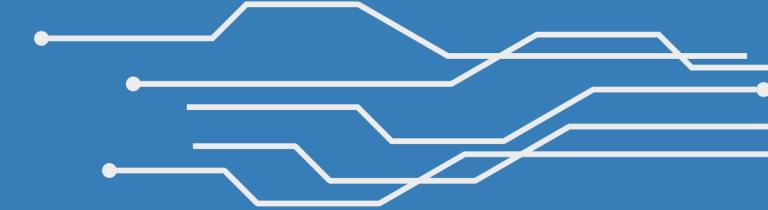
Perintah Dasar SQL



- **Data Definition Language (DDL)**
- **Data Manipulation Languange (DML)**
- Data Control Languange (DCL)
- Constraints
- Clauses
- Function



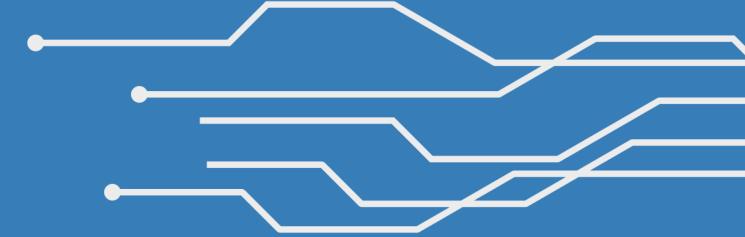
Data Definition Language(DDL)



- DDL merupakan perintah dasar yang bertujuan untuk membuat serta mendefinisikan struktur database.
- **CREATE DATABASE, TABLE** - membuat database dan table
- **RENAME TABLE** - mengubah nama pada table
- **DROP TABLE** - menghapus data pada database
- **ALTER TABLE** - menghapus/menambahkan field



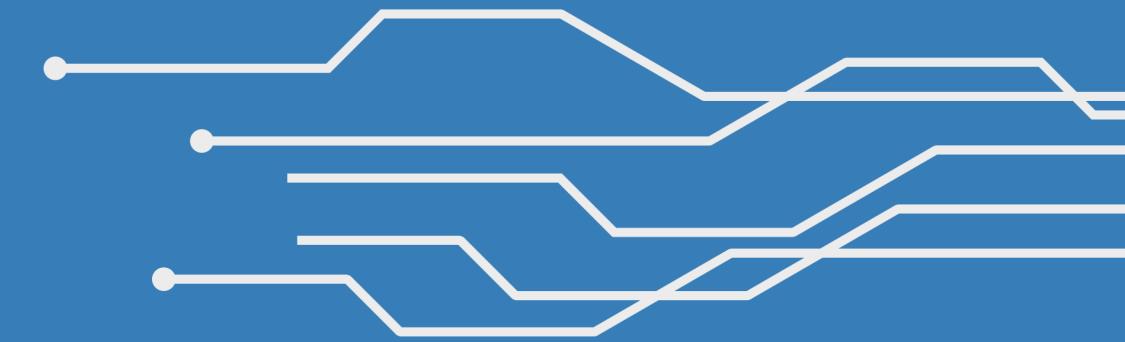
Data Definition Language(DDL)



- Perintah dasar satu ini digunakan untuk mengolah data dan memanipulasi data dalam suatu tabel database.
- **SELECT** - Mengambil/menampilkan data dari tabel.
- **INSERT INTO** - Menambahkan data baru ke tabel
- **UPDATE** - Memperbarui data yang sudah ada.
- **DELETE FROM** - Menghapus data dari tabel.

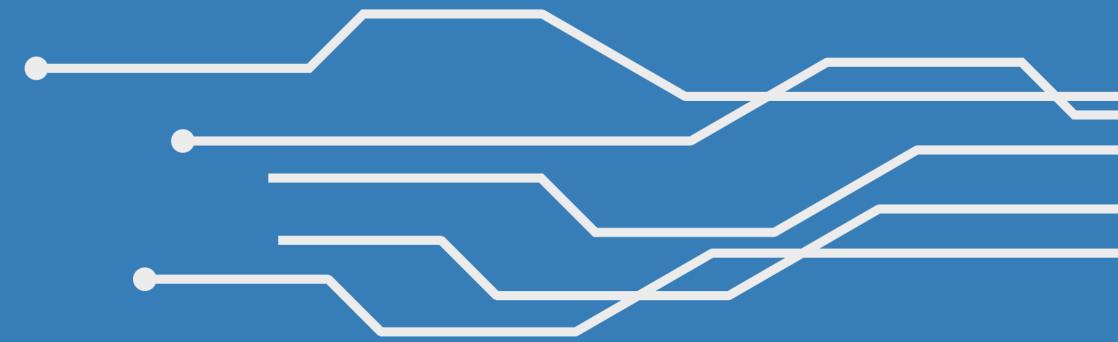


Constraints



- **PRIMARY KEY:** Kunci unik yang mengidentifikasi setiap baris dalam tabel.
- **FOREIGN KEY:** Kunci yang menghubungkan baris dalam tabel yang berbeda.
- **NOT NULL:** Mencegah kolom memiliki nilai NULL.
- **CHECK:** Memvalidasi nilai sebelum dimasukkan ke dalam kolom.

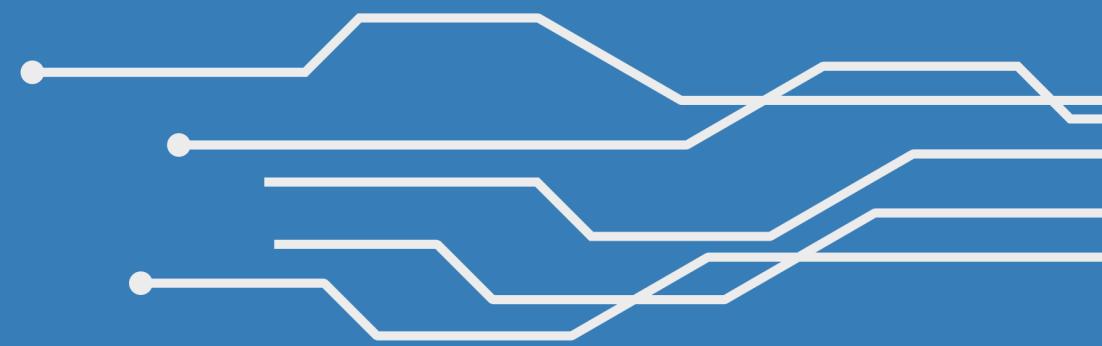
Clauses



- **WHERE:** Menetapkan kriteria dalam operasi SELECT, UPDATE, DELETE.
- **GROUP BY:** Mengelompokkan hasil berdasarkan kolom tertentu.
- **HAVING:** Mirip dengan WHERE, tetapi digunakan setelah GROUP BY.
- **ORDER BY:** Mengurutkan hasil berdasarkan kolom tertentu.



Function



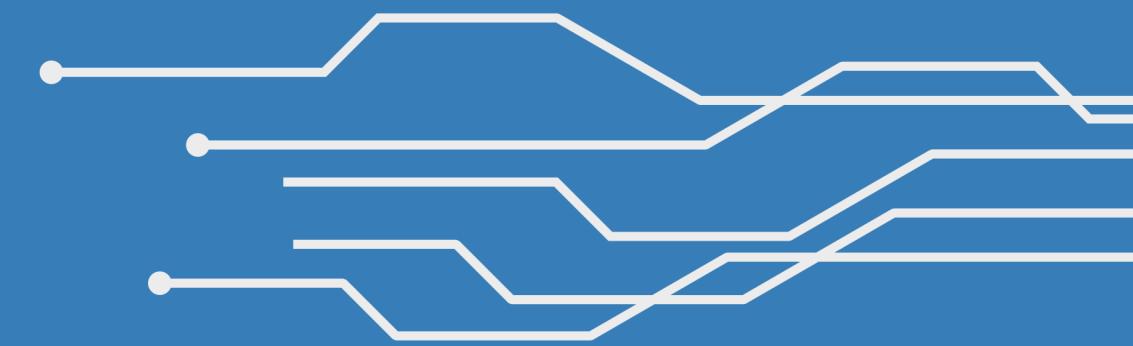
- **SUM(), COUNT(), MAX(), MIN()**: Fungsi agregasi untuk melakukan operasi pada data.
- **SUBSTR(), UPPER(), LOWER()**: Fungsi untuk manipulasi string.



Praktik Connect

Database SQLite

Persiapan

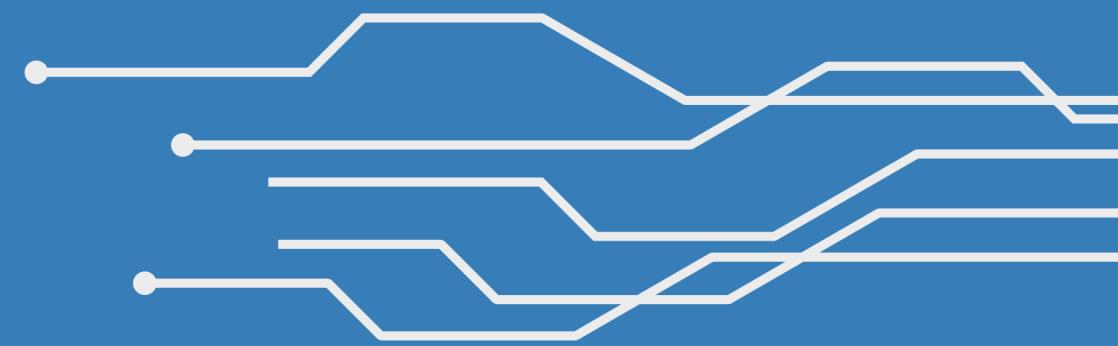


- DB Browser for SQLite

<https://sqlitebrowser.org/dl/>

- VS Code
- Terminal Git Bash

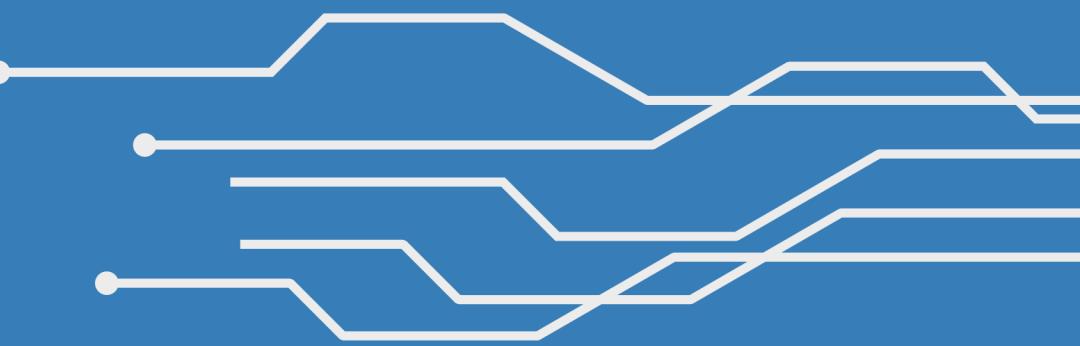
Connect Database 1



- Buat project baru di VS CODE Anda.
- Buat file baru diberi nama **sql-connect.py**

```
● ● ●  
# Koneksi dengan Database SQLITE  
import sqlite3  
koneksi = sqlite3.connect('database_pegawai.db')  
  
# Buat Database dan Table Pegawai  
koneksi.execute('''  
    CREATE TABLE PEGAWAI(  
        id_pegawai INTEGER PRIMARY KEY AUTOINCREMENT,  
        nama VARCHAR(50),  
        jabatan VARCHAR(50),  
        kota VARCHAR(50),  
        gaji INTEGER(11)  
    )  
    ''')  
koneksi.close()
```

Connect Database 2

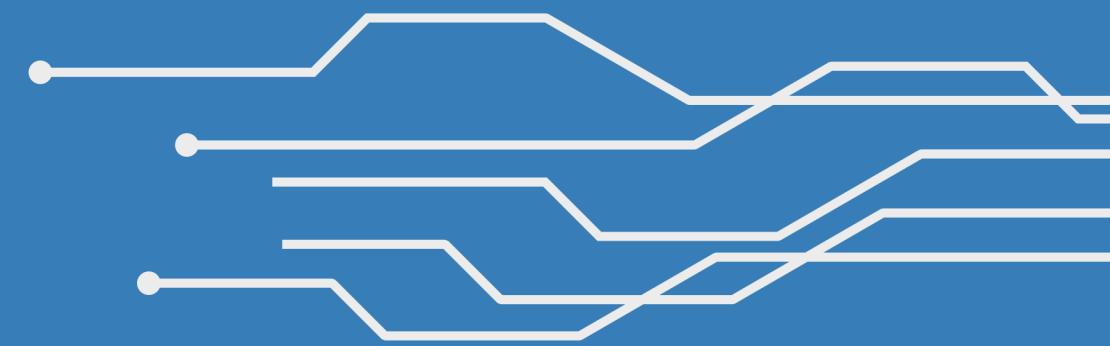


- Setelah dibuat, jika berhasil maka akan muncul file baru berekstensi **.db(database)**.
- nama file sesuai yang dibuat sebelumnya yaitu **(database_pegawai.db)**

A screenshot of a code editor interface, likely Visual Studio Code, showing a Python script for connecting to a SQLite database. The interface includes a top menu bar with File, Edit, Selection, View, Go, Run, and others. On the left is an Explorer sidebar with icons for files, folders, and databases. The main area shows a file named 'sql-pegawai-connect.py' with the following code:

```
1 # KONEKSI DB
2 import sqlite3
3 koneksi = sqlite3.connect('database_pegawai.db')
4
5 # Buat Database
6 koneksi.execute('CREATE TABLE')
7
8
9
10
11
12
13
14
15 koneksi.close()
```

Connect Database 3



- Buka DBrowser SQLite.
- Pastikan Anda ingat direktori project yang tersimpan file Database-nya.
- **Open Database**
- Lalu pilih file database (database_pegawai.db)
- Maka hasilnya akan seperti disamping.

The screenshot shows the DB Browser for SQLite interface. The title bar indicates the connection is to 'DB Browser for SQLite - D:\ASDOS-2023\ALPRO-PRAKTIK-V-JUMAT\SQLITE-PERT10\database_pegawai.db'. The menu bar includes File, Edit, View, Tools, and Help. Below the menu is a toolbar with New Database, Open Database, Write Changes, Revert Changes, Open Project, and Save Project. The main area has tabs for Database Structure, Browse Data, Edit Pragmas, and Execute SQL, with sub-tabs for Create Table, Create Index, Modify Table, Delete Table, and Print. The Database Structure tab is selected, displaying a table named 'PEGAWAI' with columns: id_pegawai (INTEGER), nama (VARCHAR(50)), jabatan (VARCHAR(50)), kota (VARCHAR(50)), and gaji (INTEGER(11)). It also lists other database components like sqlite_sequence, Indices (0), Views (0), and Triggers (0). To the right of the table, the CREATE TABLE statement for 'PEGAWAI' is shown: `CREATE TABLE PEGAWAI(id_pegawai INTEGER PRIMARY KEY, nama VARCHAR(50), jabatan VARCHAR(50), kota VARCHAR(50), gaji INTEGER(11))`. The schema for 'sqlite_sequence' is also listed: `CREATE TABLE sqlite_sequence(name,seq)`.

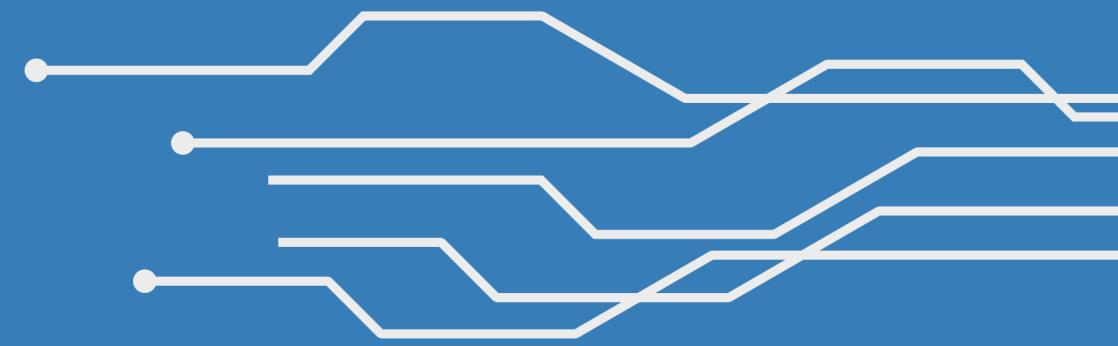
Insert Query 1

- Buat file baru diberi nama sql-insert.py.
- Setelah itu Run file python.
- Jika berhasil maka tidak akan ada error.



```
import sqlite3
koneksi = sqlite3.connect('pegawai_kelasV.db')
# INSERT DATA KE TABEL
koneksi.execute(f'''
    INSERT INTO PEGAWAI (nama, jabatan, kota, gaji)
    VALUES('Rinto', 'UI/UX Design', 'Kalimantan', '1900000')
    ''')
koneksi.execute(f'''
    INSERT INTO PEGAWAI (nama, jabatan, kota, gaji)
    VALUES('Devano', 'Backend', 'Riau', '1300000')
    ''')
koneksi.execute(f'''
    INSERT INTO PEGAWAI (nama, jabatan, kota, gaji)
    VALUES('Riska', 'Frontend Dev', 'Banten', '2500000')
    ''')
koneksi.commit()
koneksi.close()
```

Select Query (Menampilkan Query) 1



- Buat file baru diberi nama sql-select.py
- Masukkan kode berikut untuk kodingan awal.

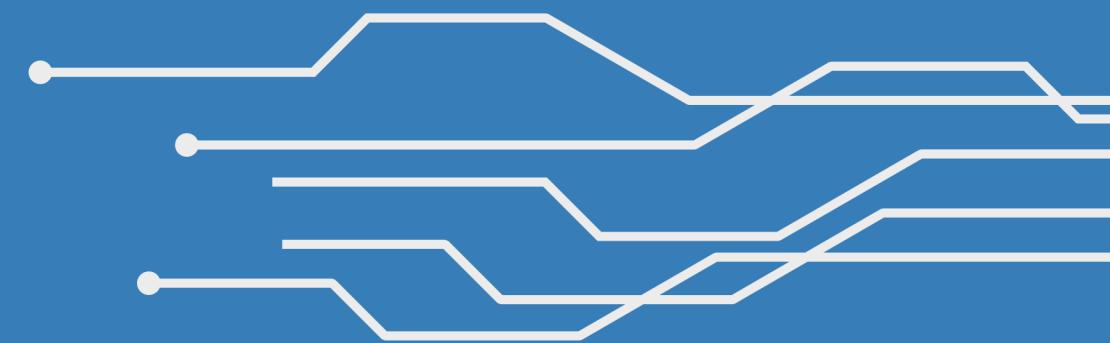
```
# SELECT ALL DATA PEGAWAI
import sqlite3
koneksi = sqlite3.connect('database_pegawai.db')

kursor = koneksi.cursor()
# Mengambil semua data dalam tabel dan tampilkan
kursor.execute("SELECT * FROM PEGAWAI")
# Tampilkan dalam bentuk baris
baris_tabel = kursor.fetchall()

# Membuat format table dengan method format()
print("Data Pegawai Konoha 2023")
print("*"*80)
print("{:<5} {:<20} {:<20} {:<20}".format("ID", "Nama", "Jabatan", "Kota", "Gaji"))
print("-"*80)
# Tampilkan data sesuai format tabel dg perulangan
for baris in baris_tabel:
    print("{:<5} {:<20} {:<20} {:<20} ".format(baris[0], baris[1], baris[2], baris[3], baris[4]))

koneksi.close()
```

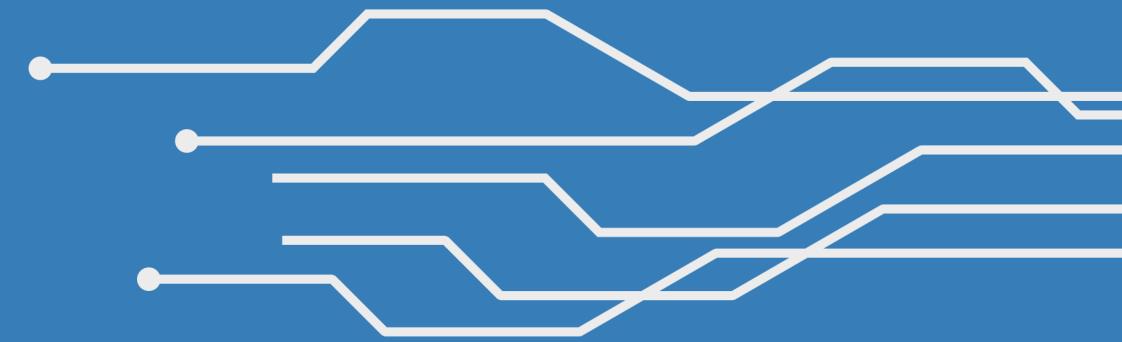
Select Query (Menampilkan Query) 2



- Setelah dijalankan maka hasilnya seperti berikut.

Data Pegawai Konoha 2023				
ID	Nama	Jabatan	Kota	Gaji
1	Rinto	UI/UX Design	Kalimantan	1900000
2	Devano	Backend	Riau	1300000
3	Riska	Frontend Dev	Banten	2500000

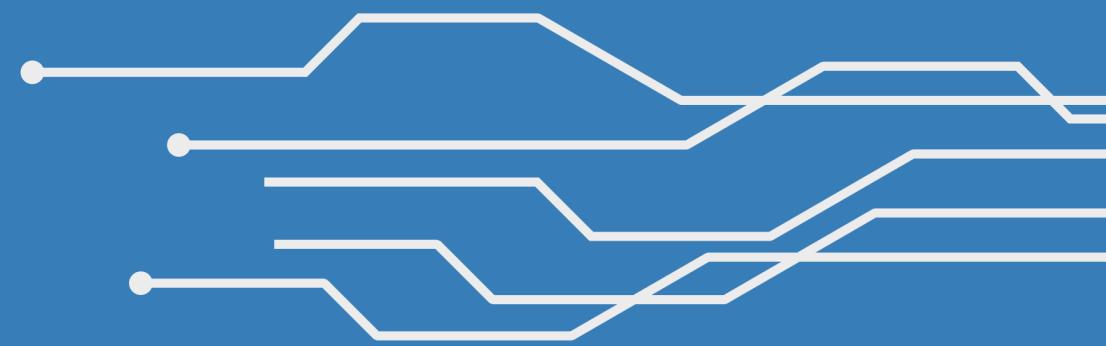
Select Where/And/Or



- Select Where
digunakan untuk
mengambil data
berdasarkan filter
tertentu misalnya
hanya id = 1 saja.
- Sekarang hanya
diganti bagian
kursor.execute()
saja.

```
● ● ●  
# Where id = 1  
kursor.execute("SELECT * FROM PEGAWAI WHERE id_pegawai = 1")  
# Where AND  
kursor.execute("SELECT * FROM PEGAWAI WHERE jabatan = 'Developer' AND kota = 'Bandung'")  
# Where OR  
kursor.execute("SELECT * FROM PEGAWAI WHERE jabatan = 'Developer' OR kota = 'Bandung'")
```

SUM & AVG



- SUM = Menjumlahkan nilai data.
- AVG = Menghitung rata-rata.

```
● ● ●

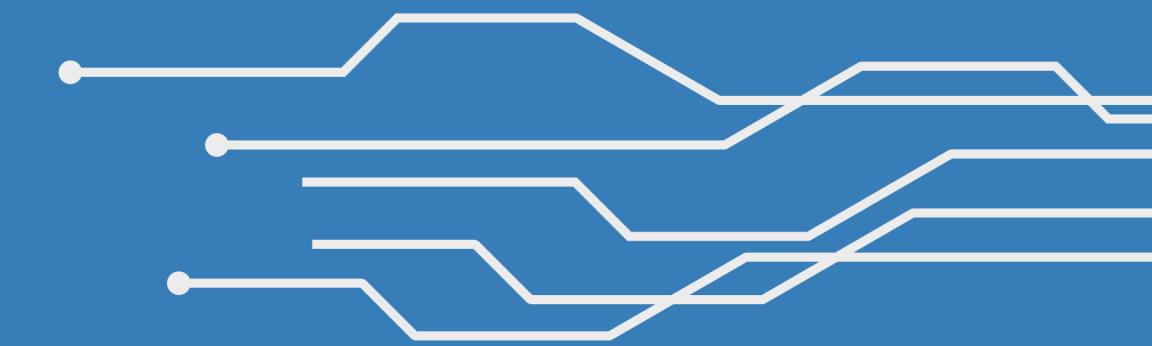
# Ambil data berdasarkan rata-rata gaji AVG( ) dan SUM( )
kursor.execute("SELECT SUM(gaji) FROM PEGAWAI")
total = kursor.fetchone()[0] # ambil data gaji jadikan baris baru dimulai dari indeks 0

# Ambil data berdasarkan rata-rata gaji AVG( ) dan SUM( )
kursor.execute("SELECT AVG(gaji) FROM PEGAWAI")
rata_rata = kursor.fetchone()[0] # ambil data gaji jadikan baris baru dimulai dari indeks 0
```

Update Query

- Update data harus ditentukan dengan primary key dari database.
- Contohnya primary key db pegawai adalah id_pegawai.

```
● ● ●  
import sqlite3  
koneksi = sqlite3.connect('database_pegawai.db')  
kursor = koneksi.cursor()  
# Ubah berdasarkan id_pegawai  
id_pegawai = 2  
gaji_baru= 5200000  
# Gunakan Query UPDATE SET  
kursor.execute(f"UPDATE PEGAWAI SET gaji = {gaji_baru} WHERE id_pegawai = {id_pegawai}")  
koneksi.commit()  
# Cek apakah data berhasil diubah atau belum  
if kursor.rowcount > 0: #cek berdasarkan adanya baris atau tdk  
    print(f"Data dengan ID {id_pegawai} berhasil Diubah!")  
else:  
    print(f"Tidak ada data pegawai dengan ID{id_pegawai}!")  
# Putuskan koneksi  
koneksi.close
```



Delete Query

- Delete data pun sama kita harus menggunakan primary key.
- Dihapus berdasarkan id_pegawai.

```
import sqlite3
koneksi = sqlite3.connect('database_pegawai.db')
kursor = koneksi.cursor()
# Ubah berdasarkan id_pegawai
id_pegawai = 2
# Gunakan Query UPDATE SET
kursor.execute(f"DELETE FROM PEGAWAI WHERE id_pegawai = ?", (id_pegawai,))
koneksi.commit()
# Cek apakah data berhasil diubah atau belum
if kursor.rowcount > 0: #cek berdasarkan adanya baris atau tdk
    print(f"Data dengan ID {id_pegawai} berhasil dihapus!")
else:
    print(f"Tidak ada data pegawai dengan ID{id_pegawai}!")
# Putuskan koneksi
koneksi.close
```

Terima Kasih

