

# SQL (Structure Query Language)

Gentisya Tri Mardiani, S.Kom





# SQL

- Structure Query Language (SQL) adalah bahasa query standar yang digunakan untuk mengakses basis data relasional.
- Kemampuan SQL:
  - Query (memperoleh data)
  - Pendefinisian struktur data
  - Pengubahan data
  - Pengaturan sekuritas, dll



# Antarmuka SQL terhadap DBMS

Penggunaan SQL sebagai:

- bahasa administrasi basis data
- bahasa query interaktif
- bahasa pemrograman basis data
- bahasa client/server



# Elemen SQL

- Pernyataan SQL

Pernyataan adalah perintah SQL yang meminta suatu tindakan kepada DBMS. Pernyataan dasar SQL meliputi : INSERT, SELECT, DELETE, CREATE, UPDATE, dll.

- Nama

Nama digunakan sebagai identitas bagi objek-objek pada DBMS. Contoh objek pada DBMS adalah table, kolom dan pengguna

- Type Data

Setiap data mempunyai type data. Type data standar, contoh : Char, Integer, Numeric, dll.



# Elemen SQL

- Konstanta

Konstanta menyatakan nilai tetap. Contoh : Konstanta Numeric (123,-245, 5.45) dan konstanta string ('ABC')
- Ekspresi
  - Segala sesuatu yang menghasilkan nilai
  - Ekspresi digunakan untuk menghitung nilai
  - Ekspresi aritmatika yang digunakan : \*, /, +, -
  - Contoh : (LABA/MODAL)\*1000
- Fungsi Bawaan
  - Fungsi adalah sebuah sub program yang menghasilkan suatu nilai jika dipanggil.
  - SQL mempunyai sejumlah fungsi bawaan, yaitu : Max, Min, AVG, dll.



# Kelompok Pernyataan SQL

- DDL (Data Definition Language)

Kelompok perintah yang berfungsi untuk mendefinisikan atribut atribut basis data, table, batasan terhadap suatu atribut serta hubungan antar tabel

contoh: CREATE, DROP, ALTER

- DML (Data Manipulation Language)

Kelompok perintah untuk memanipulasi data dalam basis data

contoh: SELECT, INSERT, DELETE, UPDATE

- DCL (Data Control Language)

untuk mengendalikan pengaksesan data. Pengendalian dapat dilakukan berdasar per pengguna, per table, per kolom maupun per operasi yang boleh dilakukan

contoh: GRANT, REVOKE, LOCK TABLE



# Kelompok Pernyataan SQL

- Pengendali Transaksi  
Perintah yang berfungsi untuk mengendalikan pengeksekusian transaksi  
contoh: COMMIT, ROLLBACK
- Pengendali Programatik  
Mencakup pernyataan-pernyataan yang berhubungan dengan pemanfaatan SQL dalam bahasa lain contohnya pada COBOL  
contoh : DECLARE, OPEN, FETCH, CLOSE



# Bekerja dengan SQL

- **CREATE TABLE**

Create Table Pengarang (  
KdPengarang Integer Unique Not Null,  
Nama Char (15) Not Null,  
Alamat Char (30),  
Kota Char (15))

- Unique : Nilai kolom tidak boleh ada yang sama harus unik
- Not Unique : Nilai kolom boleh kembar / sama
- Null : Nilai kolom boleh tidak diisi
- Not Null : Nilai kolom harus diisi



# Bekerja dengan SQL

- **CREATE INDEX**

CREATE INDEX Idx\_(Nama Atribut) On Nama table (Nama Atribut)

- Contoh :

Create Index Idx\_KdPengarang On Pengarang (KdPengarang)

Create Unique Index Idx\_KdPengarang On Pengarang(KdPengarang)



# Bekerja dengan SQL

- **Tambah Data (INSERT)**
- Insert Into Nama table Values ( , )

Contoh :

Insert Into Pengarang Values (1,'Ahmad','Jl.Dago','Bandung')

Insert Into Pengarang Values (2,'Deni','Jl.Pahlawan','Bandung')

Insert Into Pengarang Values (3,'Cahyadi','Jl.Dipatiukur','Jakarta')

Hasil :

KdPengarang	Nama	Alamat	Kota
1	Ahmad	Jl.Dago	Bandung
2	Deni	Jl.Pahlawan	Bandung
3	Cahyadi	Jl.Dipatiukur	Jakarta



# Bekerja dengan SQL

- Ubah Data (UPDATE)

Update Nama Table

Set Atribut = IsiAtributBaru (Value baru)

Where Kondisi

- Contoh :

Update Pengarang

Set Alamat='Jl.Ir.Juanda'

Where KdPengarang=1

KdPengarang	Nama	Alamat	Kota
1	Ahmad	Jl.Ir.Juanda	Bandung
2	Deni	Jl.Pahlawan	Bandung
3	Cahyadi	Jl.Dipatiukur	Jakarta



# Bekerja dengan SQL

- Menghapus Data (DELETE)  
Delete from Nama Table  
Where Kondisi
- Contoh :  
Delete from Pengarang  
Where KdPengarang=1

KdPengarang	Nama	Alamat	Kota
2	Deni	Jl.Pahlawan	Bandung
3	Cahyadi	Jl.Dipatiukur	Jakarta

# Ekuivalensi aljabar relasional dan SQL



	Aljabar Relasional	SQL
Selection	$\sigma_P(E)$  Contoh: $\sigma_{\text{kota}='Bandung'}(\text{Pengarang})$	Select * from E where P  Contoh: Select * from pengarang where kota='Bandung'
Projection	$\pi_{\text{column}}(E)$  Contoh: $\pi_{\text{KdPengarang, Nama}}(\text{Pengarang})$	Select column from E  Contoh: Select KdPengarang, Nama from Pengarang

	Aljabar Relasional	SQL
Union	$E1 \cup E2$  Contoh: $Pegawai \cup Pekerjaan$	$\text{select * from E1 union select * from E2}$  Contoh: $\text{Select * from pegawai union select * from pekerjaan}$
Set Difference	$E1 - E2$  Contoh: $\pi_{NIP} (Pegawai) - \pi_{NIP} (Pekerjaan)$	$\text{Select * from E1 except select * from E2}$  Contoh: $\text{Select NIP from pegawai except select NIP from pekerjaan}$
Cartesian Product	$E1 \times E2$  Contoh: $\pi_{NIP, Nama, Gaji} (\sigma_{Pegawai.NIP=Pekerjaan.NIP} (Pegawai \times Pekerjaan))$	$\text{Select * from E1, E2}$  Contoh: $\text{Select Pegawai.NIP, Pegawai>Nama, Pekerjaan.Gaji from Pegawai, Pekerjaan where Pegawai.NIP = Pekerjaan.NIP;}$



	Aljabar Relasional	SQL
Set Intersection	$E1 \cap E2$  Contoh: $Pegawai \cap Pekerjaan$	<code>select * from E1 intersect select * from E2</code>  Contoh: <code>Select * from pegawai intersect select * from pekerjaan</code>
Join	$E1 \bowtie E2$  Contoh: $Pegawai \bowtie_{Pegawai.NIP=Pekerjaan.NIP} Pekerjaan$	<code>Select * from E1 join E2</code>  Contoh: <code>Select * from pegawai join Pekerjaan on Pegawai.NIP= Pekerjaan.NIP</code>

