

BAB XII

PERINTAH SQL (Structured Query Language) JAVA DATABASE

A. CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat dapat memahami Perintah SQL Java Database
2. Mahasiswa dapat menggunakan Perintah SQL Select, Select Where Java Database
3. Mahasiswa dapat menggunakan Perintah SQL Insert into, Update, delete dan kapan di gunakannya pada Java Database

B. MATERI

1. Perintah SQL

Pada materi pertemuan sebelumnya, kita telah membuat record di tabel database Java DB kita. Dalam pertemuan ini, kita akan mempelajari beberapa perintah SQL sehingga Anda dapat memanipulasi record dalam sebuah tabel.

SQL singkatan dari *Structured Query Language*, dan merupakan cara untuk query database. Kita dapat

memilih *records* dengan *select*, menyisipkan dengan *insert into*, menghapus dengan *Delete*, dan memperbarui catatan dengan *Update*, membuat tabel, menjatuhkan tabel, dan banyak lagi. SQL adalah bahasa yang cukup handal.

2. Kategori Perintah SQL

Secara umum, SQL terdiri dari 3 kategori yaitu :

a. DDL (Data definition language)

merupakan perintah-perintah yang digunakan oleh seorang Database Administrator untuk mendefinisikan struktur dari database, baik membuat tabel baru, menentukan struktur penyimpanan tabel, model relasi antar tabel, validasi data, dll Contoh perintah : Create, Use, Alter, Drop

b. DML (Data Manipulation language)

Secara umum digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah yang umum dilakukan adalah:

- 1) **SELECT**
- 2) **INSERT**
- 3) **UPDATE**
- 4) **DELETE**

c. DCL (Data Control language)

Bagian ini berkenaan dengan cara mengendalikan data, seperti siapa saja yang bisa melihat isi data, bagaimana data bisa digunakan oleh banyak user, dll. Contoh perintah : Grant, Invoke

a. Perintah Select

Berfungsi untuk menampilkan data, atau sejumlah data, atau menampilkan data berdasarkan kriteria perintah dasarnya seperti berikut :

SELECT [field | *] FROM nama_tabel [WHERE Kriteria];

*Kondisi/Kriteria adalah syarat yang harus dipenuhi suatu data agar ditampilkan.
kondisi_agregat adalah syarat khusus untuk fungsi agregat.
Kondisi dapat dihubungkan dengan operator logika, misalnya AND, OR, dan sebagainya.*

SQL menggunakan kata kunci sederhana untuk melakukan pekerjaan. Jika kita ingin memilih semua catatan dari tabel, kata SELECT dan FROM digunakan, bersama dengan simbol "semua records", yaitu dengan simbol * (Asterisk):

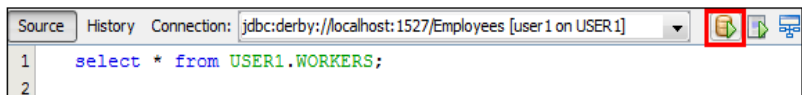
SELECT * FROM table_name

Jika kita lihat bagian atas jendela NetBeans, kita akan melihat bahwa pernyataan SELECT telah disiapkan: (CATATAN: SQL tidak membedakan huruf besar-kecil) contoh

select * from ADMIN.WORKERS

Perintah ini mengatakan "Pilih/Tampilkan semua record dari tabel *Workers*". (Bagian ADMIN, sebelum titik *Workers* , adalah sesuatu yang disebut Skema. Ini menjelaskan struktur database, tetapi juga mengidentifikasi hal-hal seperti pengguna dan hak istimewa yang mereka miliki. Jangan khawatir tentang skema, karena ini hanya opsi yang boleh tidak di ketik)

Pada NetBeans, kita menjalankan pernyataan SQL dengan mengklik tombol Run pada toolbar:



Gambar 12. 1 Contoh Perintah Sql Select

Hasil dari pernyataan SQL kemudian ditampilkan di bagian bawah jendela:

#	ID	FIRST_NAME	LAST_NAME	JOB_TITLE
1		1 Helen	James	IT Manager
2		2 Eric	Khan	Programmer
3		3 Tommy	Lee	Ststems Analyst
4		4 Priyanka	Collins	Programmer

Gambar 12. 2 Hasil Pernyataan SQL

b. Perintah dengan Clausa WHERE

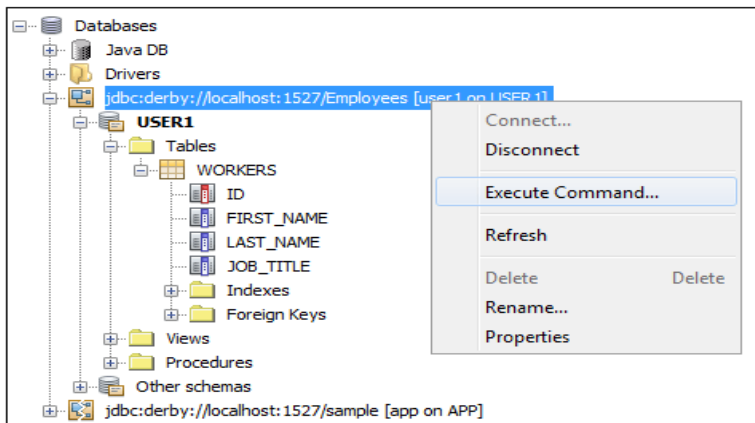
Untuk mempersempit hasil pencarian Anda, Anda dapat menggunakan klausa ***WHERE*** dengan pernyataan ***SELECT***:

```
SELECT * FROM table_name WHERE column_name=value
```

Setelah kata kunci *WHERE* kita memerlukan nama kolom dari tabel kita. kemudian mengetik tanda sama dengan, diikuti dengan nilai. Sebagai contoh, berikut adalah pernyataan SQL yang mengembalikan semua programmer di tabel kita:

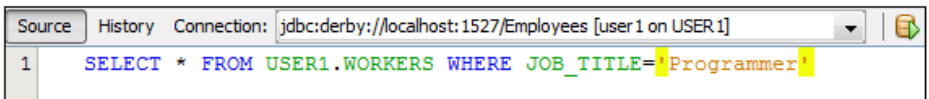
```
SELECT * FROM ADMIN.WORKERS WHERE  
JOB_TITLE='Programmer'
```

Untuk mencoba perintah pernyataan SQL klik kanan pada nama tabel pada **Services** area. Dari menu yang muncul, pilih **Execute Command**:



Gambar 12. 3 Mencoba Execute

Ketika Anda mengklik Execute Command, sebuah jendela baru akan muncul. Ketik Pernyataan SQL di atas, lalu klik ikon Run :



Gambar 12. 4 Run Execute Command

Hasilnya akan ditampilkan di bagian bawah jendela:

#	ID	FIRST_NAME	LAST_NAME	JOB_TITLE
1		2 Eric	Khan	Programmer
2		4 Priyanka	Collins	Programmer

Gambar 12. 5 Hasil Run Exeute Command

Seperti yang kita lihat, dua baris dikembalikan dari kueri.

Kita juga dapat menggunakan kata kunci LIKE dengan klausa WHERE. Ini kemudian menggantikan tanda sama dengan. LIKE biasanya digunakan dengan karakter wildcard. Karakter wildcard % berarti "karakter apa saja", misalnya, sedangkan garis bawah digunakan hanya untuk satu karakter.

Disamping tanda sama dengan atau kata kunci LIKE, Anda juga dapat menggunakan operator kondisional (Lebih Besar Dari, Kurang Dari, dll.). Jika

kita memiliki kolom Salary, kita bisa mencari semua pekerja yang dibayar lebih dari 1000 seminggu:

```
SELECT * FROM ADMIN.WORKERS WHERE  
SALARY > 1000
```

Contoh contoh penggunaan perintah Select

```
SELECT * FROM user WHERE total_transaksi = 0
```

Menampilkan pengguna yang tidak pernah bertransaksi.

```
SELECT IDUser, NamaUser FROM user WHERE  
jml_transaksi < 10000 AND total_transaksi > 50000
```

Menampilkan ID User Nama User dari tabel user yang bertransaksi kurang dari 10000 dan Total transaksi lebih dari 5.0000.

```
SELECT * FROM user ORDER BY jml_transaksi DESC
```

Menampilkan seluruh data dari tabel user diurutkan berdasarkan jumlah transaksi terbesar ke terkecil.

3. Fungsi Select Agregat

Fungsi agregat, yaitu fungsi-fungsi khusus yang melibatkan sekelompok data (agregat). Secara umum fungsi agregat adalah:

SUM untuk menghitung total nominal data

COUNT untuk menghitung jumlah kemunculan data

AVG untuk menghitung rata-rata sekelompok data

MAX dan MIN untuk mendapatkan nilai maksimum/minimum dari sekelompok data.

Fungsi agregat digunakan pada bagian SELECT.

Syarat untuk fungsi agregat diletakkan pada bagian HAVING, bukan WHERE

Contoh :

```
SELECT Sum(Quantity) AS Totalnya, Avg(quantity) AS  
RataRatanya, Count(quantity) AS JumlahTransaksi  
FROM [Order Details];
```

```
SELECT sum(quantity) AS Total2 FROM [order details]  
WHERE ProductId=2;
```

```
SELECT ProductId, Sum(Quantity) AS TotalQuantity  
FROM [Order Details]  
GROUP BY [Order Details].ProductId;
```


4. Perintah INSERT INTO

Pernyataan INSERT INTO digunakan untuk menyisipkan record baru ke dalam tabel.

untuk menulis pernyataan INSERT INTO dapat dilakukan dengan dua cara:

a. Tentukan nama kolom dan nilai yang akan disisipkan:

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

b. Jika kita menambahkan nilai untuk semua kolom tabel, kita tidak perlu menentukan nama kolom dalam kueri SQL. Namun, pastikan urutan nilai dalam urutan yang sama dengan kolom dalam tabel. Di sini, sintaks INSERT INTO adalah sebagai berikut:

```
INSERT INTO table_name  
VALUES (value1, value2, value3, ...);
```

Contoh penggunaan pernyataan Insert into :

Contoh 1 :

```
INSERT INTO Customers (CustomerName,  
ContactName, Address, City, PostalCode, Country)  
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen  
21', 'Stavanger', '4006', 'Norway');
```

Contoh 2 :

```
INSERT INTO Customers  
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen  
21', 'Stavanger', '4006', 'Norway');
```

5. Perintah UPDATE

Pernyataan UPDATE digunakan untuk mengubah record yang ada dalam tabel.

UPDATE Syntax

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

Catatan: Hati-hati saat memperbarui catatan dalam tabel! Perhatikan klausa WHERE dalam pernyataan UPDATE. Klausa WHERE menentukan record mana yang harus diperbarui. Jika Anda menghilangkan klausa WHERE, semua catatan dalam tabel akan diperbarui!

Contoh demo penggunaan update mengambil dari database Nwind sebagai berikut :

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Akan merubah nilai ContactName = **Alfred Schmidt** dan City = **Frankfurt** dimana mempunyai Customer ID= 1

Perintah Update :

UPDATE Customers

SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'

WHERE CustomerID = 1;

Hasil Run perintah tersebut adalah :

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Alfred Schmidt	Obere Str. 57	Frankfurt	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Gambar 12. 6 Hasil Run perintah Update

6. Perintah DELETE

Pernyataan DELETE digunakan untuk menghapus record yang ada dalam tabel.

DELETE Syntax :

```
DELETE FROM table_name WHERE condition;
```

Catatan: Hati-hati saat menghapus record dalam tabel! Perhatikan klausa WHERE dalam pernyataan DELETE. Klausa WHERE menentukan record mana yang harus dihapus. Jika Anda menghilangkan klausa WHERE, semua record dalam tabel akan dihapus!

Contoh Perintah DELETE :

Di bawah ini adalah pilihan dari tabel "Customers" di database sampel Northwind:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Gambar 12. 7 Record Tabel Customers

Pernyataan SQL berikut menghapus customers

"Alfreds Futterkiste" dari tabel " customers ":

Perintahnya :

```
DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';
```

Tabel "Customers" sekarang akan terlihat seperti dibawah ini:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Gambar 12. 8 Hasil Run Perintah Delete

Delete Semua Records

Dimungkinkan untuk menghapus semua baris dalam tabel tanpa menghapus tabel. Ini berarti bahwa struktur tabel, atribut, dan indeks akan tetap utuh:

Perintah :

```
DELETE FROM table_name
```

Pernyataan SQL berikut menghapus semua baris dalam tabel "Customers", tanpa menghapus tabel:

```
DELETE FROM Customers;
```

C. LATIHAN

Sumber Database menggunakan Nwind.accdb

1. Tampilkan data Order Id, Ship Via, Ship Address, Ship City dari tabel Orders yang mempunyai nilai Ship City = New York
2. Tampilkan semua data Order date dengan batasan tanggal order =15 Jan 2006 s/d tanggal order =07 Apr 2006
3. Tampilkan semua data dengan kriteria payment type =credit card, antara batasan paid date = 22 Jan 2006 s/d paid date = 22 Apr 2006.
4. Tampilkan rata-rata shipping fee dari tabel Orders
5. Tampilkan Total semua shipping fee dari tabel Orders
6. Tampilkan Total shipping fee antara paid date = 15 Jan 2006 s/d paid date = 22 Apr 2006 dari tabel Orders
7. Tampilkan Total shipping fee yang mempunyai ship City =Las Vegas
8. Rubah semua data Ship City =Bandung dari Ship =New York dari tabel order

D. REFERENSI