



Materi 2 : Klausa dan Keyword dalam SQL

TUJUAN UMUM : Mahasiswa memahami dan terampil membuat Query dengan klausa between, Keyword IN, Not In, Alias, sub query, Klausa Order By, Having

1. Buatlah Table seperti dibawah ini kemudian isilah data-nya sesuai table

PEGAWAI						ANAK			
*NIP	NAMA	KOTA	TGLLAHIR	JK	KJ	NIP	NAMA	JK	UMUR
N01	YAHYA	JAKARTA	20/01/1980	L	J02	N02	HARIS SETIAWAN	L	10
N02	BAYU	JAKARTA	08/02/1980	L	J01	N02	HANNY	P	5
N03	M ZAINI DAHLAN	CIBINONG	03/12/1981	L	J03	N05	S HALIZAH	P	9
N04	AHMAD RIZAL	JAKARTA	17/03/1983	L	J03	N05	M SULTAN RAISYAH	L	1
N05	MULIA	JAKARTA	05/06/1986	L	J04				
N06	UMI	JAKARTA	06/01/1986	P	J06				
N07	LUKMANUL HAKIM	DEPOK	24/06/1982	L	J03				

JABATAN			Catatan :	
KJ	JABATAN	GAJI	Primary Key : (dan huruf digaris bawah)	
J01	MARKETING LEADER	7000000	JK :	Jenis Kelamin
J02	SYSTEM ANALYST	6000000	KJ :	Kode Jabatan
J03	PROGRAMMER	5500000	Tipe Data :	
J04	NETWORK ENGINEER	4500000	NIP, NAMA, JK, KJ, KOTA :	Varchar
J05	DATABASE ADMINISTRATOR	5000000	GAJI	Money
J06	SEKRETARIS	3000000	UMUR	Integer
			TGLLAHIR :	Date

a. Klausa DISTINCT

Digunakan untuk menghilangkan record yang berulang.

Contoh :

SELECT NIP AS [NIP YANG SUDAH PUNYA ANAK] FROM ANAK

Jika Menggunakan klausa **DISTINCT**, NIP pada contoh diatas tidak perlu berulang

Contoh :

SELECT DISTINCT NIP AS [NIP YANG SUDAH PUNYA ANAK] FROM ANAK

b. Klausa WHERE

Digunakan untuk menampilkan data dengan menempatkan kondisi pada perintah SELECT unuk tujuan hasil yang lebih spesifik

Contoh :

SELECT NIP, NAMA, KOTA FROM PEGAWAI WHERE NIP = 'N02' OR NIP = 'N06'



c. Penggunaan BETWEEN

Adalah operator yang digunakan dalam klausa **WHERE**, untuk menampilkan range data yang sudah di tentukan.

Contoh :

```
SELECT * FROM JABATAN WHERE GAJI BETWEEN 3000000 AND 5000000
```

Jika ditambahkan NOT :

```
SELECT * FROM JABATAN WHERE GAJI NOT BETWEEN 3000000 AND 5000000
```

d. Penggunaan LIKE

Adalah operator yang digunakan dalam klausa **WHERE**, untuk menampilkan data yang mengandung string tertentu.

Contoh :

```
SELECT * FROM PEGAWAI WHERE NAMA LIKE '%ZA%'
```

e. Penggunaan Alias Nama Table

Digunakan untuk alias dari nama table dan dimanfaatkan menyingkat nama table, jika nama table terlalu panjang.

Contoh :

```
SELECT NAMA, JABATAN FROM PEGAWAI AS P, JABATAN AS J  
WHERE P.KJ = J.KJ
```

Atau tanpa keyword AS

```
SELECT NAMA, JABATAN FROM PEGAWAI P, JABATAN J  
WHERE P.KJ = J.KJ
```

f. Keyword TOP

Jika pada database sudah tersimpan banyak record dan misalkan anda hanya ingin melihat beberapa jumlah (n) record teratas.

Contoh melihat tiga record teratas :

```
SELECT TOP 3 * FROM JABATAN
```

g. Keyword IN dan NOT IN

Keyword IN dan NOT IN digunakan dalam klausa WHERE untuk menentukan daftar nilai dimana suatu field nilainya harus sama atau tidak sama.

Contoh :

```
SELECT * FROM JABATAN WHERE KJ IN ('J01','J02','J03')
```



h. SubQuery

Pada pembahasan klausa WHERE, terdapat keyword IN dan NOT IN, dan EXISTS dan NOT EXISTS. Keyword-keyword ini penggunaannya sering digunakan untuk query yang mengandung subquery. SubQuery adalah query biasa yang hasilnya (result set) digunakan untuk Query lain-nya.

Misalkan kita ingin mencari data pegawai yang sudah mempunyai anak. Tanpa sub query, kita dapat menuliskan sintaks seperti berikut ini :

```
SELECT DISTINCT P.* FROM PEGAWAI P, ANAK A WHERE P.NIP = A.NIP
```

Kita juga dapat menggunakan SubQuery untuk menghasilkan output yang sama. Contoh dibawah ini menggunakannya dengan keyword IN :

```
SELECT * FROM PEGAWAI WHERE NIP IN (SELECT NIP FROM ANAK)
```

Sedangkan jika kita menggunakan keyword EXISTS , sintaks-nya adalah sebagai berikut

```
SELECT * FROM PEGAWAI WHERE EXISTS ( SELECT * FROM ANAK WHERE ANAK.NIP = PEGAWAI.NIP)
```

Ketiga Query diatas akan menampilkan output/hasil yng sama.

i. Klausa ORDER BY

Digunakan untuk mendefinisikan pengurutan hasil query berdasarkan suatu kolom tertentu. Pada klausa ORDER BY dapat di gunakan kombinasi dari column pada table

Contoh jika kita ingin mengurutkan berdasarkan field jabatan, dapat digunakan query sebagai berikut :

```
SELECT * FROM JABATAN ORDER BY JABATAN
```

j. Fungsi Aggregate

Untuk menghasilkan suatu nilai kembalian yang merupakan rangkuman tertentu, kita dapat menggunakan fungsi-fungsi Aggregate yang disediakan oleh SQL SERVER.

Fungsi Aggregate	Keterangan
AVG	Mencari Nilai Rata-rata
SUM	Mencari Jumlah Nilai
MIN	Mencari Nilai Minimum
MAX	Mencari Nilai Maximum
COUNT	Menghitung jumlah data yang terdeteksi

k. Klausa GROUP BY

Penggunaan klausa GROUP BY biasanya bersama dengan **aggregate function**, **Aggregate** adalah fungsi yang bekerja pada sekumpulan data.

Jika **Aggregate** digunakan tanpa GROUP BY, Aggregate tersebut akan mengkalkulasi seluruh record pada table, dan jika menggunakan GROUP BY, kalkulasi aggregate dikelompokkan sesuai nilai pada suatu field.



Contoh query untuk menampilkan NIP, Nama Pegawai dan Jumlah anaknya :

```
SELECT P.NIP, P. NAMA, COUNT(A.NIP) AS [JUMLAH ANAK] FROM PEGAWAI  
P, ANAK A WHERE P.NIP = A.NIP  
GROUP BY P.NIP, P.NAMA
```

1. Klausula HAVING

Klausula ini hanya digunakan jika query juga menggunakan klausula GROUP BY. Klausula HAVING diterapkan pada nilai aggregate yang di kelompokkan, jadi penggunaannya erat kaitannya dengan penggunaan function dan GROUP BY.

Misalkan kita ingin menampilkan data jabatan, jumlah pegawai yang mempunyai jabatan itu, dan total gaji yang harus dikeluarkan perusahaan untuk tiap Jabatan tersebut.

Contoh Query tanpa menggunakan klausula Having :

```
SELECT J.JABATAN, COUNT(P.NIP) AS [JUMLAH PEGAWAI], SUM(GAJI) AS  
TOTAL GAJI YANG DIKELUARKAN] FROM PEGAWAI P, JABATAN J  
WHERE P.KJ = J.KJ  
GROUP BY J.GAJI, J.JABATAN
```

Sedangkan jika kita ingin membatasinya hanya untuk menampilkan yang total gajinya lebih besar dari lima juta, kita dapat menggunakan klausula HAVING.

```
SELECT J.JABATAN, COUNT(P.NIP) AS [JUMLAH PEGAWAI], SUM(GAJI) AS  
TOTAL GAJI YANG DIKELUARKAN] FROM PEGAWAI P, JABATAN J  
WHERE P.KJ = J.KJ  
GROUP BY J.GAJI, J.JABATAN  
HAVING SUM (J.GAJI) > 5000000
```



PERCOBAAN PRAKTIKUM (Latihan-1)

1. Tampilkan 3 jabatan dengan gaji paling besar
2. Tampilkan karyawan yang tidak mempunyai anak
3. Tampilkan Total jumlah pegawai
4. Tampilkan Total gaji karyawan
5. Tampilkan Total pegawai yang dengan jabatan programmer
6. Tampilkan rata-rata gaji pegawai
7. Tampilkan anak dari pegawai yang usianya paling muda