

PERTEMUAN KE – 4

DATA DEFINITION LANGUAGE (DDL), DCL (DATA CONTROL LANGUAGE)

A. TUJUAN

Mahasiswa mampu menggunakan perintah SQL (DDL) untuk membuat database, tabel serta relationship-nya

B. TEORI SINGKAT

MySQL adalah salah satu produk RDBMS yang sangat populer juga bersifat open source. Seperti kebanyakan database server lainnya, MySQL dapat diakses oleh aplikasi klien dengan bahasa standar, yaitu SQL (Structure Query Language).

DDL (Data Definition Language) adalah perintah-perintah yang digunakan untuk menjelaskan objek dari database. Dengan kata lain DDL digunakan untuk mendefinisikan kerangka database.

Data Definition Language, yang termasuk golongan ini adalah:

- Membuat database
- Menghapus database
- Membuat tabel
- Menghapus tabel
- Mengubah tabel

1. Membuat database

Perintah DDL yang digunakan untuk membuat database.

Syntax : `CREATE DATABASE nama_db;`

Contoh: `CREATE DATABASE Akademik;`

2. Menghapus database

Perintah DDL yang digunakan untuk menghapus database

Syntax : `Drop database nama_db;`

Contoh: `Drop database Akademik;`

3. Membuat tabel

Perintah DDL yang digunakan untuk membuat table.

Syntax:

```
CREATE TABLE nama_tabel (nama_kolom1, type_kolom1, nama_kolom2,
type_kolom2,...);
```

Aturan nama tabel dan kolom:

- Harus diawali dengan huruf
- Panjang nama / karakter antara 1–30 characters
- Mengandung A–Z, a–z, 0–9, _, \$, dan #
- Tidak boleh terdapat nama yang sama (duplikat) dengan object database lain.
- Tidak boleh menggunakan keyword SQL database

Contoh: `CREATE TABLE Mahasiswa (NoMhs char(8) not null, Nama char(25) not null, Alamat char(30) not null);`

4. Menghapus tabel

Perintah DDL yang digunakan untuk menghapus tabel

Syntax : `Drop table nama_table;`

Contoh : `Drop table Mahasiswa;`

5. Mengubah tabel

a. Menambah Kolom Baru Pada Tabel

Syntax :

`ALTER TABLE namatabel ADD fieldbaru tipe;`

namatabel adalah nama tabel yang akan ditambah fieldnya. Fieldbaru adalah nama kolom yang akan ditambahkan, tipe adalah tipe data dari kolom yang akan ditambahkan.

b. Mengubah Tipe Data atau Lebar Kolom Pada Tabel

Syntax :

```
ALTER TABLE namatabel MODIFY COLUMN field tipe;
```

namatabel adalah nama tabel yang akan diubah tipe data atau lebar kolomnya. Field adalah kolom yang akan diubah tipe data atau lebarnya. Tipe adalah tipe data baru atau tipe data lama dengan lebar kolom yang berbeda.

c. Mengubah Nama Kolom

Syntax :

```
ALTER TABLE namatabel CHANGE COLUMN namalamakolom namabarukolom  
tipedatanya;
```

namatabel adalah nama tabel yang akan diubah nama kolomnya, namalamakolom adalah kolom yang akan diganti namanya, namabarukolom adalah nama baru kolom, typedatanya adalah tipe data dari kolom tersebut.

d. Menghapus Kolom Pada Tabel

Syntax :

```
ALTER TABLE namatabel DROP COLUMN namakolom;
```

6. Mendefinisikan Foreign Key Pada Tabel

Untuk mendefinisikan foreign key, maka harus dipastikan bahwa tabel dan atribut yang dirujuk (tabel induk dari foreign key) sudah didefinisikan terlebih dahulu.

Syntax :

```
CREATE TABLE namatabel  
(  
Field1 TipeData1,  
Field2 TipeData2,  
FOREIGN KEY (Field2) REFERENCES namatabelinduk  
(namakolominduk) ON UPDATE CASCADE  
ON DELETE NO ACTION  
)
```

atau

```
ALTER TABLE namatabel ADD CONSTRAINT namaconstraint FOREIGN KEY  
(namakolom) REFERENCES namatabelinduk (namakolominduk) ON UPDATE CASCADE  
ON DELETE NO ACTION;
```

7. CONSTRAINT

Constraint adalah batasan atau aturan yang ada pada table. Constraint mencegah penghapusan data dari suatu table yang mempunyai keterkaitan dengan table yang lain.

Beberapa tipe constraint :

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK

a. Constraint NOT NULL

Suatu kolom yang didefinisikan dengan constraint NOT NULL tidak boleh berisi nilai NULL. Kolom yang berfungsi sebagai kunci primer (primary key) otomatis tidak boleh NULL.

Contoh : Lihat pada point 2 (Contoh Create table)

b. Constraint UNIQUE

Constraint UNIQUE mendefinisikan suatu kolom menjadi bersifat unik.

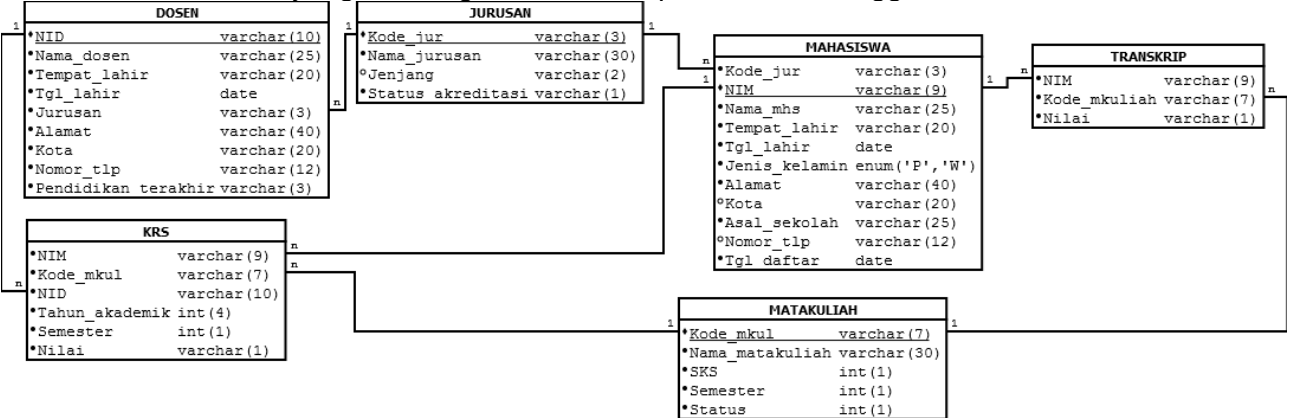
Contoh : penggunaan nama e-mail pada tabel pegawai.

- c. **Constraint PRIMARY KEY**
 Constraint PRIMARY KEY membentuk key yang unik untuk suatu table. Kolom yang didefinisikan sebagai PRIMARY KEY akan mengidentifikasi suatu baris data menjadi unik.
 Contoh : `CREATE TABLE Mahasiswa (NoMhs char(8) not null, Nama char(25) not null, Alamat char(30) not null), Primary Key (Nomhs);`
- d. **Constraint FOREIGN KEY**
 FOREIGN KEY constraint didefinisikan pada suatu kolom yang ada pada suatu table, dimana kolom tersebut juga dimiliki oleh table yang lain sebagai suatu PRIMARY KEY.
 Keyword dari Constraint FOREIGN KEY :
 FOREIGN KEY : Mendefinisikan kolom yang ada pada tabel anak
 REFERENCES : Mengidentifikasi tabel dan kolom dalam tabel induk
 ON DELETE CASCADE : Penghapusan baris dependent dalam tabel anak pada saat sebuah baris pada tabel induk dihapus
 ON DELETE SET NULL : Mengkonversi dependent foreign key ke nilai NULL.

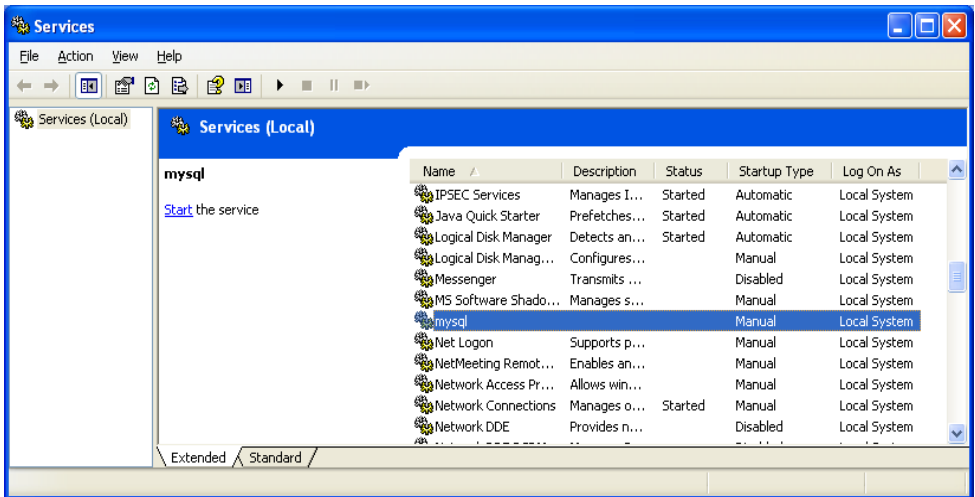
8. **Membacap database**
 Untuk membuat backup harian dari database MySql bisa memanfaatkan fasilitas Mysqldump bawaan server MySQL, yang berfungsi untuk membackup database mysql.
 Syntax dari mysqldump adalah :
`mysqldump --user [user name] --password=[password] [database name] > [dump file]`
 Keterangan perintah di atas :
- **--user [user name]** : user name adalah user account dalam MySql server
 - **--password=[password]** : password dari user name
 - **[database name]** : nama database yang akan di backup , apabila kita ingin membackup semua database yang ada gunakan options **--all-databases**
 - **[dump file]** : adalah nama file hasil backup database, bisa dengan menggunakan path , misal `/var/backup/mysql.sql`
- Contoh penggunaan :
`mysqldump --user root --password=g123 akademik > akademik_bckup.sql`

C. PRAKTIK

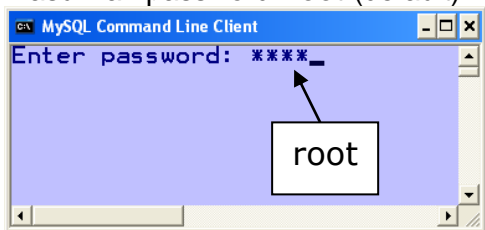
Berikut adalah skema yang akan digunakan untuk pertemuan 4 hingga 14:



1. Perhatikan skema tabel di atas.
 MySQL server dapat diaktifkan (dilihat) lewat
 Start > Setting > ControlPanel > Administrative Tools > Services



Klik Start the service, sehingga kolom Status menjadi started
 Menjalankan program MySQL Command line Client
 Klik **Start > All Programs > AppServ > MySql Command Line Client**
 Anda akan mengakses MySQL dari client, dengan user root.
 masukkan password: **root** (default)



setelah dimasukkan password, akan muncul tampilan:

```
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.0.51b-community-nt-log MySQL Community Edition
(GPL)
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

Keluar dari program MySQL Command line Client salah satu dari ini:

```
mysql> EXIT
mysql> QUIT
mysql> \q
```

Memasukan Query

Setelah user connect ke server MySQL, selanjutnya user bisa menuliskan query pada prompt mysql.
 Contoh :

```
SELECT VERSION(), CURRENT_DATE;
```

Hasilnya:

```
+-----+-----+
| VERSION() | CURRENT_DATE |
+-----+-----+
| 5.0.45-community-nt | 2015-02-23 |
+-----+-----+
1 row in set (0.02 sec)
```

```
SELECT VERSION(); SELECT NOW();
```

Hasilnya:

```
+-----+
| VERSION() |
+-----+
| 5.0.45-community-nt |
+-----+
1 row in set (0.00 sec)
```

```
+-----+
| NOW() |
+-----+
| 2015-02-23 08:57:22 |
+-----+
1 row in set (0.00 sec)
```

2. Buat database akademik dengan perintah sebagai berikut

```
create database akademik;
```

3. Menampilkan database yang ada

```
show databases;
```

4. Menghapus database

```
drop database akademik;
```

5. Setelah dihapus, tampilkan lagi database yang ada seperti perintah no.3

6. Buat lagi database akademik seperti perintah no.2

7. Masuk ke lingkungan database akademik dengan menggunakan perintah :

```
use akademik;
```

8. Buatlah tabel Dosen pada database akademik

```
create table Dosen (
    NID varchar(10) not null primary key,
    Nama_dosen varchar(25),
    Tempat_lahir varchar(20),
    Tgl_lahir date,
    Jurusan varchar(3),
    Alamat varchar(40),
    Kota varchar(20),
    Nomor_tlp varchar(12),
    Pendidikan_terakhir varchar(3)
);
```

9. Buatlah tabel Mahasiswa pada database akademik

```
create table Mahasiswa (
    NIM varchar(9) not null primary key,
    Nama_mahasiswa varchar(25),
    Tempat_lahir varchar(20),
    Tgl_lahir date,
    Jenis_kelamin enum('P','W'),
    Jurusan varchar(3),
    Alamat varchar(40),
    Kota varchar(20),
    Asal_sekolah varchar(25),
    No_telp varchar(12),
    Tgl_daftar date
);
```

10. Buatlah tabel Jurusan pada database akademik dengan 3 buah kolom sebagai berikut : Kode_jur varchar(3), Nama_jurusan varchar(30), Jenjang varchar(2) dengan primary keynya adalah Kode_jur.

11. Melihat daftar tabel

```
show tables;
```

12. Melihat struktur tabel

```
desc Jurusan;
```

13. Lakukan hal yang sama untuk tabel yang lain.

14. Mengubah nama kolom pada suatu tabel

```
ALTER TABLE Mahasiswa CHANGE COLUMN Nama_mahasiswa Nama_mhs  
varchar(25);
```

15. Cek kembali struktur tabel Mahasiswa dengan cara yang sama seperti di no.12, apakah nama field sudah berubah?

16. Tambahkan field Status_akreditasi dengan tipe data varchar(1) pada tabel Jurusan

D. LATIHAN

- Buatlah database dan tabel untuk sistem informasi transaksi buku perpustakaan sesuai schema tabel yang sudah dibuat pada latihan di modul sebelumnya

E. TUGAS

- Buatlah database dan tabel untuk sistem informasi penjualan online sesuai schema tabel sudah dibuat pada tugas di modul sebelumnya

PERTEMUAN KE – 5

DATA MANIPULATION LANGUAGE (DML)

A. TUJUAN

Mahasiswa mampu menggunakan perintah SQL (DML) untuk memanipulasi data pada 1 tabel

B. TEORI SINGKAT

Statement DML adalah perintah-perintah yang digunakan untuk mengoperasikan atau memanipulasi isi database.

SQL menyediakan 4 perintah DML yaitu :

1. Insert
2. Update
3. Delete
4. Select

1. INSERT

Statemen INSERT INTO digunakan untuk menyisipkan record baru ke dalam sebuah tabel.

Syntax :

```
INSERT INTO nama_table [(nama_kolom,...)] VALUES (data1,...);
```

Contoh :

```
INSERT INTO Mahasiswa VALUES ('09541010','Wine Rooney','Jl.janti');
```

Atau

```
INSERT INTO Mahasiswa (NoMhs>Nama>Alamat)  
VALUES ('09541010','Wine Rooney','Jl.janti');
```

2. UPDATE

Statemen UPDATE digunakan untuk mengupdate record pada tabel yang sudah ada. Sintaks WHERE merupakan bagian opsional dari statemen UPDATE. WHERE dapat pula digunakan pada perintah delete atau select. Dengan klausa WHERE Pengguna dapat menentukan satu atau lebih kondisi yang diinginkan. Kondisi ini digunakan untuk membandingkan nilai yang diberikan dengan nilai field yang tersedia dalam tabel.

Syntax :

```
UPDATE nama_tabel SET nama_kolom=ekspresi WHERE kondisi;
```

Contoh:

Ubah alamat menjadi Jl.Solo untuk mahasiswa yang memiliki NoMhs '09541010'

```
UPDATE MHS SET Alamat='Depok' WHERE NoMhs='12100748';
```

3. DELETE

Statemen DELETE digunakan untuk menghapus records pada sebuah tabel.

Syntax :

```
DELETE FROM nama_table [WHERE kondisi];
```

Contoh:

```
DELETE FROM Mahasiswa WHERE NoMhs='095410010';
```

4. SELECT

Berguna menampilkan data dari tabel.

Sintaks : `SELECT field1,field2,... FROM nama_tabel;`

WHERE : berfungsi untuk memberikan kondisi pemilihan sehingga hasil perintah select sesuai dengan yang dibutuhkan.

Sintak:

SELECT field1,field2,... FROM nama_tabel WHERE kondisi;
Terdapat operator relasional dengan tanda =, <>, <, >, <=, >=, serta operasi logika AND, OR, XOR yang digunakan sebagai kondisi pada where.

C. PRAKTIK

- 1. Buka command line pada AppServ dan masuk ke mysql
- 2. Buka kembali database akademik pada pertemuan ke-4
- 3. Isikan data pada tabel Dosen seperti di bawah :

NID	Nama_dosen	Tempat_lahir	Tgl_lahir	Jur	Alamat	Kota	No_telp	Pend. Terakhir
1020300001	Sinta	Bantul	1960-08-01	TI	Jl.Mawar No.2	Yogya	081788888888	S2
1020300002	Suharyono	Klaten	1980-05-30	SI	Jl.Bareng No.52	Klaten	085644444444	S2
1020300003	Pujiyanto	Sleman	1975-09-02	TI	Jl.Melati No.II	Yogya	085799999999	S2
1020300004	Suharti	Semarang	1985-10-10	TK	Jl.Anggur No.I	Solo	085755555555	S2
1020300005	Fransiska	Bantul	1978-01-23	MI	Jl.Panglima No.2	Yogya	089222222222	S3
1020300006	Paryono	Cilacap	1979-05-21	MI	Jl. Wolter Monginsidi No. 4	Yogya	089222299222	S2
1020300007	Indri	Jakarta	1958-04-12	KA	Jl. Pengok No. 2	Yogya	089222222233	S3
1020300008	Raymond	Sydney	1978-12-23	TI	Jl. Tentara Mataram No. 5	Yogya	089225522245	S3
1020300009	Rika	Banyuwangi	1982-11-30	MI	Jl. Panglima No. 44	Yogya	089222266721	S2

```
INSERT INTO Dosen (NID, Nama_dosen, Tempat_lahir, Tgl_lahir, Jurusan, Alamat, Kota, No_telp, Pendidikan_terakhir) VALUES ('1020300001','Sinta','Bantul','1960-08-01','TI','Jl.Mawar No.2','Yogya','081788888888','S2');
```

Jika semua field akan diisi, dapat juga menggunakan perintah seperti dibawah

```
INSERT INTO Dosen VALUES ('1020300002','Suharyono','Klaten','1980-05-30','SI','Jl.Bareng No.52','Klaten','085644444444','S3');
```

- 4. Isikan data pada tabel Mahasiswa dan Jurusan dengan data sebagai berikut :
Data Mahasiswa

NIM	Nama_mhs	Tmpt_lahir	Tgl_lahir	JK	Jur	Alamat	Kota	Asal Sekolah	No_tel p	Tgl_daftar
145410001	Felix Raharja	Bantul	1998-08-01	P	TI	Jl.Afandi No.56	Yogya	SMA BOPKRI 1	0876 6666 6666	2014-02-20
145410002	Rahmat Umam	Klaten	1998-04-30	P	TI	Jl.Jeruk No.5	Klaten	SMA N 1 Klaten	0859 9999 9999	2014-07-05
145610001	Cicilia Rihastuti	Sleman	1998-10-02	W	SI	Jl.Arjuna No.21	Yogya	SMA BOPKRI 2	08755 5555 555	2014-04-22
143110001	Danang Septiawan	Klaten	1997-11-10	P	TK	Jl.Blibis No.2	Solo	SMA N 2 Klaten	08574 44444 44	2014-04-20
143310001	Beta Sonata	Bantul	1997-01-12	W	MI	Jl.Seraton No.9	Yogya	SMA N 5 Yogya	08188 8888 888	2014-02-28

143210001	Harjono	Sleman	1998-03-05	P	KA	Jl.Pandawa No.3	Yogya	SMA N 3 Yogya	08177 77777 77	2014- 03-10
143210002	Priska	Gunung Kidul	1997-03-09	W	KA	Jl. Pancasila No.3	Yogya	SMK N 2 Playen	08572 22222 2	2014- 02-10

Data Jurusan

Kode_jur	Nama_jurusan	Jenjang	Status_akreditasi
TI	Teknik Informatika	SI	B
SI	Sistem Informasi	SI	B
KA	Komputerisasi Akuntansi	D3	B
TK	Teknik Komputer	D3	B
MI	Manajemen Informatika	D3	B

5. Melihat isi tabel

```
select * from Dosen;
```

6. Lakukan hal yang sama untuk tabel yang lain

7. Ubah nama pada tabel Dosen dengan NID 1020300002 menjadi **Suharjono**

```
UPDATE Dosen SET Nama_dosen='Suharjono' WHERE NID ='1020300002';
```

8. Ubah data pada tabel Mahasiswa yang kotanya ‘Yogya’ menjadi ‘Yogyakarta’

```
UPDATE Mahasiswa SET Kota='Yogyakarta' WHERE Kota='Yogya';
```

9. Tampilkan data NID, nama dosen dan Jurusan dari tabel Dosen

```
SELECT NID,Nama_dosen,Jurusan FROM Dosen;
```

10. Tampilkan NIM dan nama mahasiswa yang jurusannya ‘TI’ dan ‘SI’

```
SELECT NIM,Nama_mhs FROM Mahasiswa WHERE Jurusan='TI' OR Jurusan='SI';
```

11. Tampilkan NIM dan nama mahasiswa yang tanggal daftarnya sebelum tanggal ‘2014-03-30’

```
SELECT NIM,Nama_mhs,Tgl_daftar FROM Mahasiswa WHERE Tgl_daftar < '2014-03-30';
```

12. Hapus data pada tabel Mahasiswa yang nim nya 143210002

```
DELETE FROM Mahasiswa WHERE NIM='143210002';
```

- 13. Tampilkan Data Mahasiswa yang Tempat lahirnya di Sleman
- 14. Tampilkan Dosen yang pendidikan terakhirnya S3
- 15. Tampilkan Data Dosen jurusan TK dan MI
- 16. Tampilkan data mahasiswa yang jenis kelaminnya Pria

D. LATIHAN

- Diberikan oleh dosen pengampu pada saat praktikum.

E. TUGAS

- Masukkan masing-masing 5 record data pada tabel yang sudah dibuat ditugas sebelumnya (pada sistem informasi penjualan online)
- Gunakan perintah update dan delete untuk memodifikasi tabel – tabel tersebut
- Tugas tambahan yang diberikan oleh dosen pengampu pada saat praktikum

PERTEMUAN KE – 6
MANIPULASI DAN PENGAMBILAN DATA

A. TUJUAN

Mahasiswa mampu memahami perintah-perintah SQL untuk mengambil atau menampilkan data dalam suatu basis data dengan kondisi tertentu.

B. TEORI SINGKAT

Operator comparison ANY dan ALL

- a. Operator ANY digunakan berkaitan dengan *subquery*. Operator ini menghasilkan TRUE (benar) jika paling tidak salah satu perbandingan dengan hasil *subquery* menghasilkan nilai TRUE.
- b. Operator ALL digunakan untuk melakukan perbandingan dengan *subquery*. Kondisi dengan ALL menghasilkan nilai TRUE (benar) jika *subquery* tidak menghasilkan apapun atau jika perbandingan menghasilkan TRUE untuk setiap nilai *query* terhadap hasil *subquery*.

DISTINCT

Keyword DISTINCT digunakan untuk memunculkan hanya nilai distinct (berbeda). Statemen SELECT menghasilkan informasi dari kolom tabel. Pernyataan Distinct digunakan jika hanya menginginkan elemen yang nilainya berbeda, dengan mengabaikan jika ada nilai yang sama.

Syntax :

```
SELECT [(DISTINCT)] nama_kolom
FROM nama_table
[WHERE kondisi]
[GROUP BY nama_kolom]
[HAVING kondisi]
[ORDER BY nama_kolom [ASC/DESC]];
```

Contoh :

- Tampilkan alamat mahasiswa, dimana alamat yang sama hanya ditampilkan satu kali saja.
SELECT DISTINCT Alamat FROM Mahasiswa;

KLAUSA LIKE

Untuk membandingkan data dengan pola / struktur tertentu. untuk satu karakter dipakai (_) dan untuk string digunakan tanda (%)

Contoh :

- Tampilkan nama mahasiswa yang diawali dengan huruf "P"
SELECT Nama FROM Mahasiswa WHERE Nama LIKE "P%";
- Tampilkan nama mahasiswa untuk nama mahasiswa yang tidak mengandung unsur huruf "a"
SELECT Nama FROM MHS WHERE Nama NOT LIKE "%a%";
- Tampilkan nama mahasiswa untuk nama mahasiswa yang karakter keduanya huruf "u"
SELECT Nama FROM MHS WHERE Nama LIKE "_U%";

LOGICAL OPERATOR

operator logika yang dapat digunakan dalam perintah SQL.

Operator	Keterangan
And	Dan
Or	Atau
Not	Tidak

Contoh :

- Tampilkan semua data nilai untuk yang memperoleh nilai mid sekurang-kurangnya 60 atau nilai final diatas 75

```
SELECT * FROM NILAI WHERE MID>=60 OR FINAL>75;
```

- Tampilkan npm, kode mata kuliah dan nilai mid untuk nilai mid antara 70 sampai dengan 100

```
SELECT NPM,KDMK,MID FROM NILAI WHERE BETWEEN 70 AND 100;
```

C. PRAKTIK

1. Tampilkan data dosen yang jurusannya selain TI

```
SELECT * FROM dosen WHERE Jurusan <> 'TI';
atau
SELECT * FROM dosen WHERE not Jurusan = 'TI';
```

2. Tampilkan NIM, Nama_mhs dan Kota mahasiswa yang kotanya Klaten dan Solo

```
SELECT NIM,Nama_mhs,Kota FROM Mahasiswa WHERE Kota='Klaten' OR
Kota = 'Solo';
atau
SELECT NIM,Nama_mhs,Kota FROM Mahasiswa WHERE Kota
IN('Klaten','Solo');
```

3. Tampilkan NIM, Nama_mhs dan Kota mahasiswa yang kotanya selain Klaten dan Solo

```
SELECT NIM,Nama_mhs,Kota FROM Mahasiswa WHERE Kota NOT
IN('Klaten','Solo');
```

4. Tampilkan dosen yang tanggal lahirnya antara tanggal '1970-01-01' dan '1980-12-30'

```
SELECT * FROM Dosen WHERE Tgl_lahir>='1970-01-01' AND Tgl_lahir
<= '1980-12-30';
atau
SELECT * FROM Dosen WHERE Tgl_lahir BETWEEN '1970-01-01' AND
'1980-12-30';
```

5. Tampilkan nama dosen, Jurusan dan pendidikan terakhir yang mempunyai nama dengan huruf depannya 'S'

```
SELECT Nama_dosen,Jurusan,Pendidikan_terakhir FROM dosen WHERE
Nama_dosen LIKE 'S%';
```

6. Tampilkan nama mahasiswa untuk nama mahasiswa yang tidak mengandung unsur huruf "a"

```
SELECT Nama_mhs FROM mahasiswa WHERE Nama_mhs NOT LIKE '%e%';
```

7. Tampilkan nama mahasiswa yang belakangnya huruf 'a'
8. Tampilkan data jurusan yang nama_jurusannya mengandung kata 'informa'
9. Tampilkan tempat lahir dosen dimana tempat lahir yang sama hanya ditampilkan satu kali saja.

```
SELECT DISTINCT Tempat_lahir FROM dosen;
```

10. Tampilkan tempat lahir dosen jurusan TI dimana tempat lahir yang sama hanya ditampilkan satu kali saja.

```
SELECT DISTINCT Tempat_lahir FROM dosen WHERE Jurusan='TI';
```

11. Ketikkan query di bawah, amati hasilnya, jelaskan

```
SELECT Nama_matakuliah FROM matakuliah WHERE semester > ANY
(SELECT semester FROM matakuliah WHERE Semester > 2);
```

Modifikasi menjadi

```
SELECT Nama_matakuliah FROM matakuliah WHERE semester < ANY
```

```
(SELECT semester FROM matakuliah WHERE Semester > 2);
```

D. LATIHAN

Gunakan perintah – perintah yang ada di modul 6 pada tabel yang berada pada sistem informasi transaksi buku perpustakaan

E. TUGAS

Gunakan perintah – perintah yang ada di modul 6 pada tabel yang berada pada sistem informasi penjualan online

PERTEMUAN KE – 7
PENGUNAAN FUNGSI ARITMETIKA SQL (1)

A. TUJUAN

- Mahasiswa mampu memahami perintah SQL untuk melakukan OPERASI aritmatika dari data Yang tersimpan didalam table
- Memahami fungsi-fungsi Aritmathic yang ada dalam MySQL dan bagaimana menggunakannya

B. TEORI SINGKAT

Mengenal operator

Operator-operator aritmatika yang dapat digunakan untuk melakukan operasi-operasi ekspresi yang bertipe Numerik.

Operator	Fungsi
+	Menjumlahkan
-	Mengurangkan
*	Mengalikan
/	Membagikan

operator kondisi yang dapat digunakan di perintah Select:

Operator	Keterangan
=	Sama Dengan
<	Lebih Kecil Dari
>	Lebih Besar Dari
<=	Lebih Kecil Dari Atau Sama Dengan
>=	Lebih Besar Dari Atau Sama Dengan
<>	Tidak Sama Dengan

Fungsi Agregat

Fungsi adalah sebuah subprogram yang menghasilkan suatu nilai jika dipanggil. Fungsi agregat adalah fungsi standar di dalam SQL, suatu fungsi yang digunakan untuk melakukan summary, fungsi statistik standar yang dikenakan pada suatu tabel atau query.

- a. AVG(ekspresi)
Fungsi ini digunakan untuk mencari rata-rata nilai dalam suatu kolom dari suatu tabel atau ekspresi. Ekspresi dalam fungsi AVG umumnya adalah nama kolom. Kolom yang dicari nilai rata-ratanya adalah kolom dengan tipe data numerik.
- b. COUNT(x)
Fungsi ini digunakan untuk menghitung jumlah record (baris) dari suatu kolom dari suatu tabel. X adalah nama kolom yang ingin dicari jumlah barisnya.
- c. MAX(ekspresi)
Fungsi ini digunakan untuk mencari nilai terbesar dari suatu kolom dari suatu tabel. Kolom yang dicari nilai terbesarnya memiliki tipe data numerik.
- d. MIN(ekspresi)
Fungsi ini digunakan untuk mencari nilai terkecil dari suatu kolom dari suatu tabel. Kolom yang dicari nilai terkecilnya memiliki tipe data numerik.
- e. SUM(ekspresi)
Fungsi ini digunakan untuk mendapatkan nilai total dari suatu kolom pada suatu tabel.

Alias

Kita dapat memberikan nama lain atau alias pada sebuah tabel atau sebuah kolom. Pemberian nama alias ini baik digunakan jika kita mempunyai nama tabel atau kolom yang

panjang serta query yang kompleks. Nama alias bisa apa saja, tetapi biasanya Alias diberikan dengan nama yang singkat.

Sintak SQL Alias untuk tabel

```
SELECT column_name(s)
FROM table_name
AS alias_name
```

Sintak SQL Alias untuk kolom atau field

```
SELECT column_name AS alias_name
FROM table_name
```

C. PRAKTIK

Tambahkan tabel Matakuliah pada database akademik yang telah anda buat sebelumnya dengan struktur sebagai berikut :

Tabel Matakuliah

Field	Type	Null	Key	Default	Extra
Kode_mkul	varchar(7)	NO	PRI		
Nama_matakuliah	varchar(30)	YES		NULL	
SKS	int(1)	YES		NULL	
Semester	int(1)	YES		NULL	
Status	varchar(2)	YES		NULL	

Isikan datanya seperti berikut :

Kode_mkul	Nama_matakuliah	SKS	Semester	Status
TI01	SBD	3	2	W
TI02	Algoritma dan Pemrograman	3	1	W
SI01	SQA	2	3	P
SI02	Pengantar Bisnis	3	4	W
SI03	Teori Organisasi	2	4	P
SI04	Matematika Dasar	3	1	W
TK01	Dasar Teknik Listrik	3	1	W
TK02	Fisika	3	3	W
TK03	Administrasi Jaringan	2	5	P
KA01	Akuntansi Pengantar	3	1	W
KA02	Hukum Pajak	2	2	W
KA03	Perpajakan	2	3	W
MI01	Matematika Bisnis	2	2	P
MI02	Pemrograman Berorientasi	2	2	W
	Obyek 1			
MI03	Statistika Dasar	2	3	P

1. Hitung jumlah record pada tabel jurusan yang jenjangnya D3

```
Select COUNT(Jenjang) FROM Jurusan WHERE Jenjang='D3';
```

2. Hitung jumlah record pada tabel matakuliah yang statusnya 'W'

```
SELECT COUNT(Status) FROM matakuliah WHERE Status='W';
```

3. Tampilkan NIM tertinggi mahasiswa yang jurusanya TI

```
SELECT MAX(NIM) FROM Mahasiswa WHERE Jurusan='TI';
```

4. Hitung total sks dari semua matakuliah yang ada

```
SELECT SUM(SKS) FROM matakuliah;
```

5. Tampilkan SKS minimal dari suatu matakuliah yang kode matakuliahnya diawali huruf 'SI'

```
SELECT MIN(SKS) FROM matakuliah WHERE Kode_mkul LIKE 'SI%';
```

6. Menampilkan jumlah spp variabel dari masing-masing matakuliah

```
SELECT Nama_matakuliah,SKS,(SKS * 75000) FROM matakuliah;
```

//Menggunakan Alias

7. Menampilkan data matakuliah

```
SELECT matakuliah.Kode_mkul,matakuliah.Nama_matakuliah  
FROM matakuliah;
```

8. Menampilkan data matakuliah dengan alias tabel

```
SELECT MK.Kode_mkul,MK.Nama_matakuliah  
FROM matakuliah AS MK;
```

9. Menampilkan jumlah SPP variabel per matakuliah dengan alias pada tabel

```
SELECT MK.Nama_matakuliah,MK.SKS,(MK.SKS * 75000)  
FROM matakuliah AS MK;
```

10. Menampilkan jumlah SPP variabel per matakuliah dengan alias pada kolom dan tabel

```
SELECT MK.Nama_matakuliah,MK.SKS,(MK.SKS * 75000) as Jumlah  
FROM matakuliah AS MK;
```

D. LATIHAN

- Diberikan oleh dosen pengampu pada saat praktikum.

E. TUGAS

- Tugas diberikan oleh dosen pengampu pada akhir praktikum.