

DDL dan DML (T-SQL)

- ✓ DDL (Data Definition Language)
 - Create, Alter, Drop Table
- ✓ DML (Data Manipulation Language)
 - CRUD (Create, Read, Update, Delete)
 - SORTING
 - SELECT Filtering

Tujuan Mata Kuliah

- ✓ Mahasiswa dapat menjalankan SQL (DDL,DML) berbasis console / SQL Text Editor
- ✓ Mahasiswa memahami perintah-perintah SQL(DDL,DML)
- ✓ Mahasiswa memahami penggunaan dan pembuatan constraint pada table

Tools yang digunakan

- ✓ SqlCmd
- ✓ SQL Server Management Studio SQL Query Editor

DATA DEFINITION LANGUAGE (DDL)

Pada Data Definition Language (DDL) di bahas mengenai hal-hal yang berkaitan dengan Struktur table. Pada bagian ini akan dibahas mengenai bagaimana cara membuat table (**Create**), Mengubah Structure table (**Alter**), dan menghapus table (**Drop**)

Membuat Table (Create)

Sebelum membahas bagaimana membuat table, ada beberapa data type yang sering digunakan pada MS. SQL Server 2008 ini:

Data Type	Category	Keterangan
Varchar [(n)]	Character Strings	Karakter data type yang memiliki ukuran panjang yang tetap. Maksimum 8000 karakter.
Char [(n)]		Sama seperti varchar hanya panjang ukuran konsisten.
Int	Exact Numerics	Tipe Number bilangan bulat Ranganya -2,147,483,648 s/d 2,147,483,647 (storage: 4 bytes)
Smallint		Tipe Number bilangan bulat Ranganya -32,768 s/d 32,767 (storage: 2 bytes)
Tinyint		Tipe Number bilangan bulat Ranganya 0 s/d 255 (storage: 1 bytes)
Decimal(p,[s])		Tipe data numerik yang memiliki batasan dan desimal.
Numeric(p,[s])		Equivalent dengan Decimal
Money		Tipe data yang merepresentasikan currency value
SmallMoney		Sama seperti Money, hanya saja range nya lebih kecil
Date	Date And Time	Define tanggal, default literal format YYYY-MM-DD
DateTime		Define kombinasi tanggal dan waktu dengan base 24 jam

Setelah mengetahui tipe data tahap selanjutnya adalah membuat table. Adapun bentuk umum dari syntax create table :

```
CREATE TABLE <Nama_Table>(
  <field1> <Data_Type> [PRIMARY KEY][IDENTITY(,)] [NULL | NOT NULL],
  <field2> <Data_Type> [default ] [REFERENCES] <table_ref>(field_ref),
  <field...n> <Data_Type>[, ] CHECK (<kondisi>)
```

Contoh :

```
CREATE TABLE MATERIAL_GROUP(
  MaterialGroup varchar(4) PRIMARY KEY NOT NULL,
  Description varchar(30) NOT NULL
)
```

```
CREATE TABLE BUKU(
  BUKU_ID int IDENTITY(1000,1) PRIMARY KEY NOT NULL,
  KETERANGAN varchar(30) NOT NULL
)
```

LATIHAN 1

BUATLAH 6 STRUTURE TABLE DIBAWAH INI.

Nama Table : MATERIAL

Nama Field	Data Type	Allow Null	Default
MaterialNumber 🏆	varchar(6)		
MaterialDescription	varchar(35)		
UoM	varchar(3)	X	
BeginningStock	Int	X	0
CurrentStock	Int	X	0
Notes	varchar(30)	X	

Nama Table : MATERIAL_GROUP

Nama Field	Data Type	Allow Null	Default
MaterialGroup 🏆	varchar(4)		
Description	varchar(35)		

Nama Table : VENDOR

Nama Field	Data Type	Allow Null	Default
VendorNumber 🏆	varchar(7)		
VendorName	varchar(35)		
Address	varchar(50)	X	
City	varchar(25)	X	
Telepon	varchar(15)	X	

Nama Table : CUSTOMER

Nama Field	Data Type	Allow Null	Default
CustomerNumber 🏆	varchar(7)		
CustomerName	varchar(35)		
Address	varchar(50)	X	
City	varchar(25)	X	
Telepon	varchar(15)	X	

Sesi2 : DDL,DML (T-SQL)

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

Nama Table : RECEIPT

Nama Field	Data Type	Allow Null	Default
ReceiptNumber 🗝️	varchar(10)		
MaterialNumber 🗝️	varchar(6)		
VendorNumber 🗝️	varchar(7)		
ReceiptDate	date	X	
QtyReceipt	Int	X	

Nama Table : ORDERS

Nama Field	Data Type	Allow Null	Default
OrderNumber 🗝️	varchar(10)		
MaterialNumber 🗝️	varchar(6)		
CustomerNumber 🗝️	varchar(7)		
OrderDate	date	X	
QtyOrder	Int	X	

Melihat Table yang sudah di create

```
SELECT * FROM SYS.TABLES WHERE type = 'U'
```

Merubah Table (Alter)

Merubah tabel berarti merubah structure table. Pada perubahan structure table ini ada beberapa yang dapat dirubah dalam artian memodifikasi salah satu atau beberapa field pada table .

Syntax untuk merubah structure table sebagai berikut:

```
ALTER TABLE <nama_table> ADD | ALTER COLUMN | DROP COLUMN <nama_field>
```

Contoh Merubah ukuran panjang dari field materialDescription pada table MATERIAL:

```
ALTER TABLE MATERIAL ALTER COLUMN MaterialDescription varchar(50)
```

Contoh Menambah field MaterialGroup pada pada table MATERIAL:

```
ALTER TABLE MATERIAL ADD MaterialGroup varchar(4)
```

Contoh Menambah field Keterangan pada pada table MATERIAL:

```
ALTER TABLE MATERIAL ADD Keterangan varchar(50) NOT NULL
```

Contoh Menghapus field Keterangan pada pada table MATERIAL:

```
ALTER TABLE MATERIAL DROP COLUMN Keterangan
```

Sesi2 : DDL,DML (T-SQL)

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

LATIHAN 2

UBAHLAH STRUCTURE TABLE YANG TELAH DIBUAT SEBELUMNYA MENJADI SEPERTI CONTOH DIBAWAH INI

Nama Table : MATERIAL

Nama Field	Data Type	Allow Null	Default
MaterialDescription	varchar(50)		
MaterialGroup	varchar(4)		
Notes	varchar(30)	X	

Color Gray=tidak termasuk pada table (hapus)

Nama Table : CUSTOMER

Nama Field	Data Type	Allow Null	Default
JoinDate	date		

CONSTRAINT

Constraint adalah ketentuan/batasan yang diterapkan pada table untuk menjaga konsistensi dan integritas data.

- PRIMARY KEY

Sebuah primary key tidak boleh memiliki data yang sama dan tidak boleh kosong (NULL)

```
ALTER TABLE <Nama_Table> ADD CONSTRAINT <nama_constraint_primarykey>
PRIMARY KEY (<Nama_Field1>,<Nama_Field2>,. . . <Nama_Field.n>)
```

Contoh

```
ALTER TABLE MATERIAL ADD CONSTRAINT PK_MATERIAL_NUMBER
PRIMARY KEY (MaterialNumber)
```

Untuk melihat nama constraint primary key dari sebuah table yang dibuat otomatis gunakan

```
SELECT name FROM SYSOBJECTS WHERE xtype = 'PK' and
parent_obj = object_id('<Nama_Table>')
```

- NOT NULL

Digunakan untuk menjaga data agar data tidak berisi kosong (NULL). Penggunaannya sama dengan penggunaan alter table hanya ditambahkan NOT NULL.

- UNIQUE

Digunakan agar data tidak duplikasi. Hampir sama dengan Primary Key, hanya saja UNIQUE boleh bernilai NULL, sekali satu data bernilai null data lain tidak boleh.

```
ALTER TABLE <Nama_Table> ADD CONSTRAINT <nama_constraint_unique>
UNIQUE (<Nama_Field>)
```


Contoh

```
ALTER TABLE MATERIAL ADD CONSTRAINT PK_MATERIAL_NUMBER
UNIQUE (MaterialNumber)
```

- CHECK

Digunakan untuk menjaga data sesuai dengan kondisi ekspresi.

```
ALTER TABLE <Nama_Table> ADD CONSTRAINT
<nama_check_constraint> CHECK (<Ekspresi>)
```

Contoh

```
ALTER TABLE MATERIAL ADD CONSTRAINT chk_stock CHECK (CurrentStok>0)
```

- FOREIGN KEY

Digunakan Sebagai reference bahwa field tersebut ada di table lain

```
ALTER TABLE <nama_table> ADD CONSTRAINT <nama_foreign_key> FOREIGN KEY
(<nama_field>) REFERENCES <nama_table_ref>(<nama_field_ref>)[ON DELETE [SET
NULL, CASCADE, SET DEFAULT, NO ACTION]]
```

Contoh:

Pada contoh ini field materialGroup di table MATERIAL akan di merrefer ke field MaterialGroup pada table MATERIAL_GROUP

```
ALTER TABLE MATERIAL ADD CONSTRAINT FK_MATERIAL_GROUP FOREIGN KEY
(MaterialGroup) REFERENCES MATERIAL_GROUP(MaterialGroup) ON DELETE CASCADE ON
UPDATE CASCADE
```

Untuk pembuatan table reference yang digunakan saat pembuatan table

```
CREATE TABLE MATERIAL(
    MaterialNumber varchar(4) PRIMARY KEY NOT NULL,
    MaterialDescription varchar(35) NOT NULL,
```

```
MaterialGroup varchar(4) NOT NULL
REFERENCES MATERIAL_GROUP(MaterialGroup)
ON DELETE CASCADE ON UPDATE CASCADE,
UoM varchar(30) NOT NULL,
CurrentStock int DEFAULT 0
)
```

Dapat juga ditulis secara explicit

```
CREATE TABLE MATERIAL(
    MaterialNumber varchar(4) PRIMARY KEY NOT NULL,
    MaterialDescription varchar(35) NOT NULL,
    MaterialGroup varchar(4) NOT NULL,
    UoM varchar(30) NOT NULL,
    CurrentStock int DEFAULT 0,
    CONSTRAINT FK_MATERIAL_GROUP FOREIGN KEY (MaterialGroup) REFERENCES
    MATERIAL_GROUP(MaterialGroup) ON DELETE CASCADE ON UPDATE CASCADE
)
```

Pada table reference terdapat 4 opsi

1. SET NULL
Opsi ini akan mengeset isi field menjadi null jika field referensi di update/delete
 2. CASCADE
Secara spesifik field akan di update ketika field referensi di update, dan juga record akan dihapus ketika field referensi dihapus
 3. SET DEFAULT
Field record akan mengeset ke Default setiap kali ada operasi update/delete
 4. NO ACTION
Ini adalah default, jika terjadi update/delete, operasi akan di tolak. Untuk dapat menghapusnya, hapus terlebih dahulu pada field referensi
- DEFAULT
Nilai default di assign otomatis oleh SQL Server, jika tidak member nilai(NULL) pada field.

```
ALTER TABLE <Nama_Table> ADD CONSTRAINT <nama_default_constraint>
DEFAULT <nilai default> FOR <Nama_field>
```

Contoh

```
ALTER TABLE MATERIAL ADD CONSTRAINT defaultMaterial DEFAULT 0 FOR  
CurrentStock
```

- IDENTITY

Digunakan untuk penomoran otomatis (auto number) ketika data tercreate.

Penggunaanya sama dengan alter table alter column.

RESET Counter

```
DBCC CHECKIDENT( '<nama_table>', RESEED, <last_number>)
```

```
DBCC CHECKIDENT( 'tbl_counter', RESEED, 300081)
```

Syntax Untuk menghapus constraint

```
ALTER TABLE <nama_table> DROP CONSTRAINT <nama_constraint>
```

Contoh :

```
ALTER TABLE MATERIAL DROP CONSTRAINT FK_MATERIAL_GROUP
```

Menghapus Table (Drop)

Sebelum mencoba menghapus, gunakan syntax berikut untuk mencopy structure dan datanya.

```
SELECT * INTO MATERIAL_BAK FROM MATERIAL
```

Keterangan

MATERIAL_BAK adalah table yang akan diciptakan (belum ada sebelumnya)
Setelah table backup (MATERIAL_BAK) telah tercopy. Hapus table tersebut. Dengan syntax

```
DROP TABLE <nama_table>
```

Menjadi

```
DROP TABLE MATERIAL_BAK
```

Notes: untuk renaming table dan field pada SQL server menggunakan stored procedure.

Rename Field pada table

```
sp_rename '<tbl>Nama_Field_Lama>', '<tbl>Nama_Field_Baru>', 'COLUMN';
```

Contoh : `sp_rename 'orders_bak.NetValue', 'orders_bak.NetValue', 'COLUMN'`

Rename table

```
sp_rename <Nama_Table_Lama>, <Nama_Table_Baru>;
```

Contoh : `sp_rename ORDERS, ORDERS_BAK`

LATIHAN 3

BUATLAH CONSTRAINT PADA TABLE DENGAN KETENTUAN BERIKUT

1. Buatlah AutoNumber pada field berikut

Nama Table	Nama Field	Last Number
MATERIAL	MaterialNumber	300000
CUSTOMER	CustomerNumber	1000000
VENDOR	VendorNumber	5000000
RECEIPT	ReceiptNumber	4900000000
ORDERS	OrderNumber	2100000000

2. Field CurrentStock di table MATERIAL tidak boleh <= kosong (no)
3. Field VendorName di table VENDOR dan Field CustomerName di table CUSTOMER tidak boleh ada duplikasi.
4. Field MaterialGroup pada table MATERIAL bereferensi ke field MaterialGroup pada table MATERIAL_GROUP. Jika data di Table MATERIAL_GROUP di hapus atau di update sesuai dengan field yang bereferensi maka pada Table MATERIAL juga terhapus dan terupdate.
5. Field JoinDate pada table CUSTOMER by default terisi tanggal sekarang
6. Pada table ORDER dan RECEIPT QtyOrder/QtyReceipt harus >0
7. Field MaterialNumber pada table ORDERS dan RECEIPT bereferensi ke field MaterialNumber pada table MATERIAL. Jika data di Table MATERIAL di hapus akan terjadi error, jika di update akan terupdate juga di table RECEIPT dan ORDER.
8. Field VendorNumber pada table RECEIPT bereferensi ke field VendorNumber pada table VENDOR. Jika data di Table VENDOR di hapus atau di update sesuai dengan field yang bereferensi maka akan terjadi error.
9. Field CustomerNumber pada table RECEIPT bereferensi ke field CustomerNumber pada table CUSTOMER. Jika data di Table CUSTOMER di hapus atau di update sesuai dengan field yang bereferensi maka akan terjadi error.

Data Manipulation Language (DML)

Pada Data Manipulation Language (DML) akan di bahas bagaimana cara memanipulasi data. Manipulasi yang dimaksud meliputi bagaimana cara memasukan data/record ke dalam table (*insert*), Memperbaharui data (*update*), menghapus data (*delete*) dan memilih atau meyeleksi data (*select*)

1. **INSERT Statement**

```
INSERT INTO <Nama_Table>( Field1, Field2, Field..n)
VALUES(<valueField1>, <valueField2>,
<Field...n>);
```

Atau

```
INSERT INTO <Nama_Table> VALUES(<valueField1>, <valueField2>,
<valueField...n>);
```

Contoh:

```
INSERT INTO MATERIAL_GROUP VALUES('ELC', 'Electronic');
```

```
INSERT INTO MATERIAL_GROUP (MaterialGroup)VALUES('ELC',);
```

Insert dengan multiple record

```
INSERT INTO MATERIAL_GROUP (MaterialGroup) VALUES ('ELC'),
('HOB'),
('COM')
```

LATIHAN 4

BUATLAH RECORD SETIAP TABLE SEPERTI DIBAWAH INI DENGAN PERINTAH INSERT

Table - dbo.MATERIAL_GROUP		Summary
materialGroup	Description	
ELC	ELECTRONIC	
KOM	KOMPUTER	
SPRT	SPORT	

Table - dbo.MATERIAL						Summary
MaterialNumber	MaterialDescription	UoM	CurrentStock	MaterialGroup		
300001	SKATEBOARD	UNT	10	SPRT		
300002	BMX	UNT	2	SPRT		
300003	IPHONE 5	PCS	2	ELC		
300004	IPHONE 10	PCS	2	ELC		
300005	IPHONE 15	PCS	2	ELC		
300006	GALAXY TAB	PCS	7	ELC		
300007	IBM series X350	UNT	1	KOM		
300008	IBM series X255	UNT	2	KOM		
300009	IBM series X220	UNT	3	KOM		
300010	IBM series X360	UNT	4	KOM		
300011	IBM series X370	UNT	1	KOM		
300013	SAMSUNG GALAXY S2	PCS	2	ELC		
300014	SAMSUNG GALAXY S3	PCS	12	ELC		
300015	LG OPTIMUS 2X	PCS	12	ELC		
300016	LG OPTIMUS ONE	PCS	0	ELC		

Sesi2 : DDL,DML (T-SQL)

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

Table - dbo.CUSTOMER*						
	CustomerNumber	CustomerName	Address	City	Telepon	joindate
▶	7000001	PT. DEFTONES	UJUNG BERUNG	BANDUNG	022123456	13/10/2012 18:22:12
	7000002	PT. SLIPKNOT	CICADAS	BANDUNG	022123451	13/10/2012 18:23:59
	7000003	PT. BESIDE	RIJUNG BANDUNG	BANDUNG	022123452	13/10/2012 18:23:59
	7000004	PT. BK	UJUNG BERUNG	BANDUNG	022123458	13/10/2012 18:23:59
	7000005	PT. DEAD SQUAD	JAKARTA	JAKARTA	021123158	13/10/2012 18:23:50

Table - dbo.VENDOR					
	VendorNumber	VendorName	Address	City	Telepon
	5000001	IBM	AMERIKA	NEW-YORK	123456
	5000002	SAMSUNG	KOREA	KOREA	123451
	5000003	HOBBIES	DAGO	BANDUNG	2123452
	5000004	LG	KOREA	KOREA	2123458
	5000005	IPHONE	AMERIKA	BROOKLYN	1123158

Table - dbo.RECEIPT					
	ReceiptNumber	MaterialNumber	VendorNumber	ReceiptDate	QtyReceipt
	4900000001	300001	5000003	01/10/2012 11:22:05	5
	4900000002	300007	5000001	02/10/2012 11:22:05	3
	4900000002	300008	5000001	02/10/2012 11:22:05	2
	4900000002	300009	5000001	02/10/2012 11:22:05	1
	4900000002	300010	5000001	02/10/2012 11:22:05	7
	4900000003	300001	5000003	05/09/2012 11:22:05	2
	4900000004	300004	5000005	10/09/2012 11:22:05	9
	4900000004	300005	5000005	10/09/2012 11:22:05	2
	4900000005	300015	5000004	04/10/2012 11:22:05	8

Table - dbo.ORDERS					
	OrderNumber	MaterialNumber	CustomerNumber	OrderDate	QtyOrder
	2100000001	300001	7000004	10/08/2012 11:30:18	1
	2100000002	300007	7000001	05/10/2012 11:30:18	2
	2100000002	300008	7000001	05/10/2012 11:30:18	2
	2100000002	300009	7000001	05/10/2012 11:30:18	1
	2100000002	300010	7000001	05/10/2012 11:30:18	3
	2100000003	300001	7000005	20/10/2012 11:30:18	4
	2100000004	300004	7000003	01/10/2012 11:30:18	2
	2100000004	300005	7000003	01/10/2012 11:30:18	4
	2100000005	300015	7000002	14/10/2012 11:30:18	6

2. SELECT Statement

Select Dengan virtual table (Dual di oracle), dapat menggunakan function built in/user defined (dibahas pada sesi selanjutnya)

Contoh Select

```
SELECT 'apa_aja';
SELECT 1+1 dua;
SELECT (3 % 2)
SELECT GETDATE() 'Waktu Sekarang'
SELECT(SELECT (SELECT (SELECT 'Jero Pizzundh ieu select, hadeuuh...'))))

SELECT
    CASE bilangan
        WHEN '1' THEN 'HIJI'
        WHEN '2' THEN 'DUA'
        WHEN '3' THEN 'TILU'
        ELSE 'TEU AYAAN'
    END hasil
FROM (
    SELECT '1' bilangan
    )dummy_table
```

Select Dengan Table

```
SELECT [DISTINCT] [TOP]<Field1>,<Field2>,<Field...n> FROM <Nama_Table>
[WHERE] [KONDISI] [GROUP BY] [HAVING] [ORDER BY]
```

Contoh:

Sesi2 : DDL,DML (T-SQL)

Praktikum Pemrograman Client Server Database

Hadi Kusumah, S.T

```
SELECT * FROM M_MATERIAL;
```

```
SELECT MaterialNumber, MaterialDescription FROM M_MATERIAL;
```

```
SELECT MaterialNumber, MaterialDescription FROM M_MATERIAL WHERE  
MaterialNumber = '300001';
```

LATIHAN 5

SETELAH TABLE TERISI TAMPILKAN SEMUA DATA PADA SETIAP TABLE DENGAN MENGGUNAKAN SELECT STATEMENT

UPDATE Statement

```
UPDATE <Nama_Table> SET <Field1> = <valueField1>, <Field1> =  
<valueField...n> [WHERE] [<Fieldx>=valueFieldx]
```

Contoh :

```
UPDATE M_GROUP SET GroupDescription = 'Elektronik' WHERE  
MaterialGroup='ELC';
```

Tanpa kondisi (all)

```
UPDATE M_GROUP SET GroupDescription = NULL
```

LATIHAN 6

RUBAHLAH DENGAN PERINTAH UPDATE SESUAI KRITER DIBAWAH INI

- RUBAH MaterialGroup 'ELC' pada table MATERIAL_GROUP menjadi 'ELK'
- RUBAH KODE BARANG 300015 MENJADI 399999 PADA TABLE MATERIAL
- RUBAH CurrentQTY ditable Material yang bernilai 0 menjadi 5

DELETE Statement

```
DELETE FROM <Nama_Table> [WHERE] [KONDISI
```

Contoh:

```
DELETE FROM M_GROUP] WHERE KODE='1234'
```

LATIHAN 7

HAPUSLAH DENGAN PERINTAH DELETE SESUAI KRITER DIBAWAH INI

- HAPUS MaterialNumber 300007 s/d 300010 Pada Table Material

SELECT Sorting

Untuk mengurutkan data berdasarkan field tertentu. Dapat satu field atau beberapa field untuk priority pengurutan, gunakan `ORDER BY` di akhir statement

- Sorting dari terkecil ke yang terbesar

```
SELECT * FROM M_MATERIAL ORDER BY CurrentStock
```

atau

```
SELECT * FROM M_MATERIAL ORDER BY CurrentStock ASC
```

Pengurutan dengan lebih dari satu field

```
SELECT * FROM M_MATERIAL ORDER BY MaterialNumber DESC, CurrentStock ASC
```

- Sorting dari terbesar ke yang terkecil

```
SELECT * FROM M_MATERIAL ORDER BY CurrentStock DESC
```

SELECT Filtering (WHERE)

```
SELECT <field-field> FROM <Nama_Table> WHERE Ekspresi;
```

Ekspresi dari WHERE statement dengan menggunakan formula

NamaField Operator Value

- DISTINCT
Memfilter suatu field pada data yang double

Contoh:

```
SELECT DISTINCT UoM FROM M_MATERIAL
```

- NULL
Digunakan untuk memfilter apakah Null atau tidak (NOT NULL) suatu record pada field

Contoh :

```
SELECT * FROM M_MATERIAL WHERE MaterialGroup IS NULL
```

Atau menggunakan negasi NOT

```
SELECT * FROM M_MATERIAL WHERE MaterialGroup IS NOT NULL
```

- Operator (=, <>, >, >=, <, <=, Between)

Contoh :

```
SELECT * FROM M_MATERIAL WHERE MaterialNumber = '300001';
```

```
SELECT * FROM M_MATERIAL WHERE CurrentStock <> 0;
```

```
SELECT * FROM M_MATERIAL WHERE CurrentStock >= 5 AND CurrentStock<= 10 ;
SELECT * FROM M_MATERIAL WHERE CurrentStock BETWEEN 5 AND 10 ;
```

- IN / NOT IN

IN digunakan untuk filter dengan array value yang termasuk

```
SELECT * FROM M_MATERIAL WHERE MaterialNumber IN ( '300002', '300009' )
```

NOT IN digunakan untuk filter dengan array value yang tidak termasuk

```
SELECT * FROM M_MATERIAL WHERE MaterialNumber NOT IN
( '300001', '300005' )
```

- LIKE

Penggunaan LIKE biasa digunakan untuk filter dengan pola.

Contoh:

Menampilkan dengan awalan 'C'

```
SELECT * FROM M_MATERIAL WHERE MaterialDescription LIKE 'C%'
```

Menampilkan dengan akhiran 'C'

```
SELECT * FROM M_MATERIAL WHERE MaterialDescription LIKE '%C'
```

Menampilkan yang mengandung kata/huruf 'C'

```
SELECT * FROM M_MATERIAL WHERE MaterialDescription LIKE '%C%'
```

Menampilkan dengan range karakter awal

```
SELECT MaterialDescription FROM M_MATERIAL WHERE MaterialDescription
LIKE '[P-S]%'
```

Menampilkan dengan range karakter awal

```
SELECT MaterialDescription FROM M_MATERIAL WHERE MaterialDescription
LIKE '%[P-S]'
```

Menampilkan dengan range karakter opposite gunakan tanda '^'

```
SELECT MaterialDescription FROM M_MATERIAL WHERE MaterialDescription
LIKE '[^P-S]%'
```

LATIHAN 8

TAMPILKAN DENGAN PERINTAH SELECT SESUAI KRITER DIBAWAH INI

- TAMPILKAN DATA YANG CUSTOMER YANG BERKOTA DI BANDUNG DAN BERALAMAT DI UJUNG BERUNG
- TAMPILKAN VENDOR YANG BERALAMAT BUKAN DI AMERIKA
- TAMPILKAN DATA MATERIAL YANG MENGANDUNG KATA 'GALAXY'
- TAMPILKAN DATA MATERIAL YANG BERAWALAN KATA 'SAMSUNG'
- URUTKAN DATA PADA TABLE RECEIPT QtyReceipt dari yang terbesar ke yang terkecil
- URUTKAN DATA PADA TABLE RECEIPT QtyOrder dari yang terkecil ke terbesar
- TAMPILKAN DATA RECEIPT DARI TANGGAL 1 OKTOBER 2012 s/d 10 OKTOBER 2012
- TAMPILKAN VARIANT MATERIAL PADA TABLE ORDERS
- TAMPILKAN DATA PADA TABLE ORDERS dengan qty dari 2 s/d 4
- TAMPILKAN MATERIAL NUMBER YANG TIDAK DI BELI
- TAMPILKAN MATERIAL YANG TIDAK TERJUAL PADA TANGGAL 5 OKTOBER 2012
- TAMPILKAN CUSTOMER YANG MEMBELI PALING BANYAK QTY DAN SEBUTKAN MATERIALNYA

~Eof~

(#root)hadi