

NAMA : NURUL ISTIKAMA

NIM : 200250502053

## SISTEM BASIS DATA

### 1. SQL HOME

SQL home atau SQL pengantar yang dimana menggunakan bahasa standar untuk mengakses dan memanipulasi database.

Contoh: `Select * From Consumer`

### 2. SQL SINTAKS

Saat menggunakan SQL, diharuskan menggunakan sintaks yang benar. Syntax tersebut merupakan sekumpulan aturan yang elemen bahasanya digabung secara benar. Penggunaan sintaks SQL berdasarkan sintaks Bahasa Inggris yang menggunakan banyak elemen yang sama sebagai VBA sintaks (Visual Basic for Applications).

Data RDBMS (Relational Database Management System) disimpan dalam database objek yang biasa disebut dengan istilah Tabel. Tabel terdiri dari baris dan kolom yang merupakan kumpulan entri data yang saling terkait. Database biasanya terdiri dari satu atau lebih tabel yang setiap tabelnya diidentifikasi dengan nama. Baris dalam tabel berisi record dengan data.

Dalam database terdapat beberapa sistem yang memerlukan penggunaan titik koma di setiap akhir pernyataan SQL. Dalam basis data, titik koma merupakan cara standar untuk menguatkan setiap pernyataan di SQL. Serta dapat pula memungkinkan lebih dari satu sistem pernyataan SQL untuk dieksekusi ke server dalam suatu panggilan yang sama.

### 3. SQL SELECT

Pernyataan SQL Select digunakan untuk memilih data dari database. Data tersebut merupakan data dari tabel hasil, yang disebut kumpulan hasil. Untuk lebih jelasnya, SQL Select dapat difungsikan untuk mengambil baris dari beberapa tabel pada tampilan database. Pada sebagian besar aplikasi, Select merupakan perintah Bahasa manipulasi data yang paling umum digunakan.

Struktur dasar select digunakan untuk mendaftarkan atribut-atribut yang diinginkan sebagai hasil dari suatu query. Hasil dari query adalah sebuah relasi (penghubung). Perhatikan contoh pernyataan SQL berikut:

```
select nama_nasabah
```

```
from rekening
```

Hasil dari pernyataan SQL tersebut adalah relasi yang memuat atribut tunggal dengan nama\_nasabah.

Contoh kasus lain untuk menspesifikasi duplikat-duplikat yang tidak perlu dihapus, maka SQL mengizinkan penggunaan kata kunci *all*. Penulisannya sebagai berikut.

```
select all nama_nasabah
```

```
from rekening
```

Jika ingin menampilkan dua atau lebih atribut pada tampilan keluaran, dapat dilakukan dengan mendaftarkan nama atribut yang ingin ditampilkan setelah kata kunci *select*. Contohnya sebagai berikut.

```
select nama_nasabah, id_nasabah, saldo
```

```
from rekening
```

Klausula *select* dapat juga mengandung ekspresi matematika dengan operator-operator kali (\*), bagi (/), tambah (+), dan kurang (-). Berikut contoh penulisannya.

```
select nama_nasabah, id_nasabah, saldo * 0.1
```

```
from rekening
```

akan menghasilkan relasi yang atribut saldo-nya bernilai 10% kali nilai semula.

#### 4. SQL SELECT DISTINCT

Select Distinct digunakan untuk mengembalikan nilai yang berbeda (hanya nilai yang berbeda). Didalam tabel atau kolom sering kali terdapat nilai yang duplikat (nilai yang sama), untuk memasukkan nilai yang berbeda dapat menggunakan Select Distinct.

Pada bahasa permohonan formal berbasis relasional, rekaman yang tepat sama tidak boleh muncul dalam relasi. Contoh kasus untuk menghilangkan duplikasi, dapat menyisipkan kata kunci *distinct* setelah *select*.

Contoh SQL Select Distinct:

```
select distinct nama_nasabah  
from rekening
```

#### 5. SQL WHERE SQL AND,OR,NOT

SQL Where dapat digunakan untuk memfilter / menyaring catatan dan hanya mengambil catatan yang diperlukan. Klausula SQL Where digunakan untuk menentukan kondisi saat pengambilan data dari suatu tabel atau dari beberapa tabel. Klausula where dalam SQL menetapkan bahwa pernyataan Bahasa manipulasi data SQL hanya akan memengaruhi baris yang memenuhi syarat dan ketentuan tertentu.

Contoh SQL Where:

Sintaks dasar dari pernyataan Select dengan klausa Where adalah sebagai berikut :

```
select * from rekening  
where nama_nasabah = 'Nurul Istikama';
```

SQL And, Or, Not, SQL dapat menggunakan klausa penghubung logika and, or, not pada klausa Where. Operator and, or, not digunakan untuk memfilter / menyaring catatan lebih dari satu kondisi.

- a. Operator *and* menampilkan catatan jika semua kondisi yang ditampilkan adalah TRUE.

- b. Operator *or* menampilkan catatan jika salah satu kondisi yang ditampilkan adalah BENAR.
- c. Operator *not* menampilkan catatan jika kondisi yang ditampilkan adalah NOT TRUE.

Operand untuk penghubung logika dapat menggunakan operator < (lebih kecil), <= (lebih kecil atau sama dengan), > (lebih besar), >= (lebih besar atau sama dengan), dan <> (tidak sama dengan).

Contoh sintaks SQL And

```
select nama_nasabah
from rekening
where kondisi1 and kondisi2 and kondisi3
```

Contoh sintaks SQL Or

```
select nama_nasabah
from rekening
where kondisi1 or kondisi2 or kondisi3
```

Contoh sintaks SQL Not

```
select nama_nasabah
from rekening
where kondisi1 not kondisi2 not kondisi3
```

## 6. SQL ORDER BY

Klausula Order By atau kata kunci Order By dapat digunakan untuk menghasilkan permohonan yang terurut dengan urutan hasil naik atau turun. Kata kunci Order By mengurutkan hasil dalam urutan secara default.

Contoh sintaks SQL Order By:

```
select distinct nama_nasabah
from rekening,akun
where akun.no_rekening = akun.no_rekening
and cabang = 'Mamuju'
order by nama_nasabah
```

Metode pengurutan menaik dapat dituliskan dengan kata kunci *asc*, namun ketika tidak dituliskan hasil yang diberikan akan tetap sama. Untuk kaidah pengurutan menurun dapat dituliskan menggunakan kata kunci *desc*.

Contoh sintaks SQL Order By dengan menggunakan kata kunci pengurutan

```
select *
from rekening
```

order by saldo desc, no\_rekening asc

## 7. SQL INSERT INTO

Klausula insert into digunakan untuk memasukkan record baru ke dalam tabel. Pernyataan sederhana insert adalah pernyataan untuk menyisipkan satu rekaman. Biasanya nilai atribut untuk rekaman yang disisipkan harus merupakan anggota dari domain yang dimiliki atribut. pernyataan Insert Into dapat ditulis dengan menentukan nama kolom dan nilai yang akan disisipkan.

Insert into rekening (nama\_nasabah, id\_nasabah, no\_rekening)

Values ('Nurul Istikama, 'IDR-03IB-P', '7912010011')

## 8. SQL NULL VALUES

Nilai Null adalah bidang tanpa nilai. Pada tabel yang bersifat opsional disediakan tempat untuk memasukkan catatan baru atau memperbaharui catatan tanpa menambahkan nilai kebidang lain, lalu field akan disimpan dengan null values. Null values berbeda dengan nilai nol atau bidang berisi spasi. Bidang null values adalah bidang yang dibiarkan kosong selama pembuatan catatan.

Operator atau kata kunci yang digunakan untuk menguji nilai null adalah *is null* dan *is not null*. SQL juga mengizinkan menguji hasil dari suatu perbandingan dengan menggunakan klausa *is unknown* atau *is not unknown*.

Contoh SQL Sintaks IS NULL

```
select nama_nasabah  
  
from rekening  
  
where no_rekening is null;
```

Contoh SQL Sintaks IS NOT NULL

```
select nama_nasabah  
  
from rekening  
  
where no_rekening is not null;
```

Terdapat beberapa akibat dalam penggunaan nilai null pada operasi-operasi aritmetika dan perbandingan, yaitu SQL memberikan hasil *unknown* (tidak diketahui) untuk setiap perbandingan yang melibatkan nilai null. Hasil operasi matematika (+, -, \*, /) adalah null jika nilai yang masukannya null.

Pada predikat klausa where SQL mengizinkan untuk melibatkan operasi-operasi Boolean seperti *and*, *or* dan *not* pada hasil perbandingan. Definisinya sebagai berikut:

- a. **and** yaitu hasil dari *unknown* and *unknown* adalah *unknown*, *true* and *unknown* adalah *unknown*, serta *false* and *unknown* adalah *false*.

- b. **or** yaitu hasil dari *unknown or unknown* adalah *unknown*, *true or unknown* adalah *true*, serta *false or unknown* adalah *unknown*.
- c. **not** yaitu hasil dari *not unknown* adalah *known*.

Secara umum, semua fungsi *agregat* kecuali *count(\*)* pada masukannya mengabaikan nilai null. Fungsi *count(\*)* akan mengidentifikasi nilai null sebagai nol (0) sehingga fungsi yang bersangkutan dapat melakukan perhitungan seperti yang diharapkan.

## 9. SQL UPDATE

Pernyataan Update digunakan untuk mengubah nilai yang ada dalam tabel tanpa mengubah semua nilai pada rekaman. Dengan perintah update, ada baiknya untuk menghilangkan klausa *where* dalam pernyataan update untuk memudahkan semua rekaman yang ada dalam tabel.

Contoh SQL Update

```
update rekening
```

```
set saldo = saldo * 1.0
```

Untuk itu, SQL menyediakan konstruksi *case* yang mengabaikan urutan update.

```
update rekening
```

```
set saldo = case
```

```
    when saldo <= 3000000 then saldo * 1.0
```

```
    else saldo * 1.0
```

```
end
```

## 10. SQL DELET

Pernyataan delete digunakan untuk menghapus rekaman yang ada dalam tabel. Permintaan delete diekspresikan dengan cara yang mirip dengan query. Dalam klausa delete, dapat menghapus rekaman, namun tidak dapat menghapus nilai-nilai pada sebagian atribut.

Klausa *where* dalam pernyataan delete yaitu menentukan record mana yang harus dihapus. Jika menghilangkan klausa *where*, maka semua record yang ada dalam tabel akan dihapus.

Contoh SQL Delete

```
delete from rekening
```

```
where no_rekening
```

Perhatikan bahwa perintah delete hanya beroperasi pada satu relasi. Jika ingin menghapus rekaman-rekaman dari beberapa relasi, harus menggunakan perintah delete

untuk setiap relasi. Permintaan delete dapat mengandung select yang merujuk pada relasi dari mana rekaman-rekaman akan dihapus.

## 11. SQL SELECT TOP

SQL Select Top digunakan untuk menentukan jumlah rekaman yang akan dikembalikan. Select Top berguna pada tabel yang besar dengan rekaman, dapat mengembalikan sejumlah besar rekaman yang memengaruhi kinerja.

Contoh SQL Select Top

```
Select top rekening  
  
From no_rekening  
  
where kondisi;
```

Perhatikan semua penggunaan sintaks pada SQL. Tidak semua basis data sistem mendukung klausa Select Top. Pada MySQL mendukung klausa Limit untuk memilih sejumlah rekaman, sementara Oracle menggunakan Rownum.

## 12. SQL MIN DAN MAX

SQL Min berfungsi untuk mengembalikan nilai dari kolom yang dipilih, sedangkan SQL Max berfungsi untuk mengembalikan nilai terbesar dari kolom yang dipilih. Fungsi agregat adalah fungsi-fungsi yang mengambil koleksi suatu himpunan atau beberapa himpunan dan mengembalikan nilai tunggal. SQL menyediakan lima fungsi agregat baku yaitu, rata-rata(avg), terkecil (min), terbesar (max), total (sum), dan jumlah (count).

Contoh SQL Min

```
Select min (saldo)  
  
From rekening  
  
where no_rekening = '1091688433'
```

Contoh SQL Max

```
Select max (saldo)  
  
From rekening  
  
where no_rekening = '1091688433'
```

## 13. SQL COUNT, AVG, SUM

SQL Count berfungsi mengembalikan jumlah baris yang cocok dengan kriteria yang ditentukan. SQL AVG berfungsi untuk mengembalikan nilai rata-rata kolom numerik. SQL SUM berfungsi untuk mengembalikan jumlah total kolom numerik. Masukan untuk fungsi sum serta avg haruslah himpunan dari nilai-nilai numerik, tetapi

operasi-operasi yang beroperasi pada himpunan-himpunan tipe data nonnumeric seperti *string*.

#### Contoh SQL Count

```
Select count (saldo)

From rekening

where no_rekening = '1091688433'
```

#### Contoh SQL AVG

```
Select avg (saldo)

From rekening

where no_rekening = '1091688433'
```

#### Contoh SQL SUM

```
Select sum (saldo)

From rekening

where no_rekening = '1091688433'
```

### 14. SQL LIKE

SQL Like atau operator like digunakan dalam klausa Where untuk mencari pola yang ditentukan dalam kolom. SQL mengekspresikan pola-pola dengan menggunakan operator perbandingan *like*.

Ada dua karakter yang biasa digunakan dalam operator like yaitu : tanda persen mewakili nol, satu atau beberapa karakter (%) dan garis bawah mewakili satu karakter (\_). Pada Microsoft Access operator like menggunakan tanda bintang(\*).

Berikut contoh operator like yang berbeda dengan wildcard (%) dan (\_):

- LIKE 'a%' menemukan nilai apapun yang dimulai dengan 'a'
- LIKE '%a' menemukan nilai apapun yang diakhiri dengan 'a'
- LIKE '%or%' menemukan nilai apapun yang memiliki 'atau' diposisi manapun
- LIKE '\_r%' menemukan nilai apapun yang memiliki 'r' di posisi kedua
- LIKE 'a\_%' menemukan nilai apapun yang dimulai dengan 'a' dan setidaknya memiliki panjang 2 karakter
- LIKE 'a\_\_%' menemukan nilai apapun yang dimulai dengan 'a' dan panjangnya minimal 3 karakter
- LIKE 'a%o' menemukan nilai apapun yang dimulai dengan 'a' dan diakhiri dengan 'o'.

### Contoh SQL Like

```
Select nama_nasabah  
From rekening  
where akun like '%deposito%'
```

### 15. SQL WILLCARD

SQL Willcard berfungsi untuk menggantikan satu atau lebih karakter yang ada didalam string. Karakter willcard digunakan dengan operator SQL Like.

Ada beberapa karakter willcard yang ada dalam SQL Server` yaitu :

- a. % mewakili nol atau lebih karakter
- b. \_ mewakili satu karakter
- c. [] mewakili karakter tunggal apapun dalam tanda kurung
- d. ^ mewakili karakter apapun yang tidak ada dalam tanda kurung
- e. ^ mewakili berbagai karakter

### Contoh SQL Willcard

```
Select * From rekening  
where akun like 'deposito%';
```

### 16. SQL In

Operator In merupakan singkatan dari beberapa kondisi OR. SQL In difungsikan untuk menentukan nilai dalam klausa where. Kata kunci in menguji keanggotaan himpunan yang terbentuk dengan klausa select, sedangkan kata kunci not in untuk menguji sesuatu yang bukan anggota dari suatu himpunan.

### Contoh SQL In

```
Select distinct nama_nasabah  
From rekening  
where nama_nasabah in (select nama_nasabah from penabung)
```

### 17. SQL BETWEEN

Dalam SQL ada juga perbandingan dengan kata kunci between untuk menyederhanakan klausa where dengan syarat bahwa nilai lebih kecil dari atau sama dengan nilai tertentu dan lebih besar dari atau sama dengan nilai yang lain.

Operator between yaitu memilih nilai dalam rentang tertentu, nilainya berupa angka, teks atau tanggal. Klausa between bersifat inklusif artinya nilai awal dan akhir harus disertakan.



#### Contoh SQL Between

```
Select no_rekening  
From rekening  
where saldo between 1000000 and 500000
```

dengan cara yang sama dapat menggunakan klausa not between untuk syarat kondisi yang sebaliknya

```
Select no_rekening  
From rekening  
where saldo not between 1000000 and 500000
```

#### 18. SQL ALIASES

SQL Aliases digunakan untuk memberi nama pada tabel atau kolom dalam tabel. Aliases sering difungsikan dalam membuat nama kolom lebih mudah dibaca dan aliases hanya ada selama menunggu.

##### Contoh kolom SQL Aliases

```
Select no_rekening AS alias_rekening  
From nama_nasabah
```

##### Contoh tabel SQL Aliases

```
Select no_rekening (s)  
From nama_nasabah AS alias_rekening
```

#### 19. SQL JOINS

SQL tidak hanya menyediakan mekanisme-mekanisme permohonan serta operasi-operasi modifikasi basis data, tetapi juga berbagai mekanisme lain untuk menggabungkan relasi-relasi yang meliputi *condition join* dan *natural join* serta berbagai bentuk *outer join*. Operasi-operasi tambahan ini secara tipikal digunakan sebagai ekspresi dalam klausa where. Klausa joins digunakan untuk menggabungkan baris dari dua atau lebih tabel berdasarkan kolom yang terkait didalamnya.

##### Contoh SQL Joins

```
Select rekening. no_rekening, nasabah.nama_nasabah, transaksi.id_transaksi
```

From rekening

Inner join nasabah on

Rekening.nama\_nasabah = nasabah.nama\_nasabah;

## 20. SQL INNER JOIN

SQL Inner Join atau kata kunci inner Join difungsikan untuk memilih rekaman yang memiliki nilai cocok diantara kedua tabel.

Contoh SQL Inner Join

Select no\_rekening (s)

From rekening

Inner join transaksi

On rekening.no\_rekening = transaksi.no\_transaksi;

## 21. SQL LEFT JOIN

SQL Left Join berfungsi mengembalikan semua rekaman dari tabel kiri dan rekaman yang cocok dari tabel kanan sehingga menghasilkan nilai Null dari tabel kanan jika tidak ada nilai yang cocok.

Contoh SQL Left Join

Select no\_rekening (s)

From rekening

left join transaksi

On rekening.no\_rekening = transaksi.no\_transaksi;

## 22. SQL RIGHT JOIN

SQL Right Join berfungsi mengembalikan semua rekaman dari tabel kanan dan rekaman yang cocok dari tabel kiri sehingga menghasilkan nilai Null dari tabel kiri jika tidak ada nilai yang cocok.

Contoh SQL Left Join

Select no\_rekening (s)

From rekening

right join transaksi

On rekening.no\_rekening = transaksi.no\_transaksi;

### 23. SQL FULL JOIN

SQL Full Join berfungsi mengembalikan semua rekaman ketika ada kecocokan dalam catatan tabel kiri dan tabel kanan. Serta klausa Full Join dapat mengembalikan semua setelan hasil yang sangat besar

Contoh SQL Full Join

```
Select no_rekening (s)
From rekening
Full outer join transaksi
On rekening.no_rekening = transaksi.no_transaksi;
where kondisi;
```

### 24. SQL SELF JOIN

SQL Self Join merupakan gabungan regular yang ada dalam tabel tetapi berdiri sendiri dengan tabel join.

Contoh SQL Self Join

```
Select no_rekening (s)
From rekening, transaksi
Where kondisi;
```

Tabel satu dan tabel dua adalah aliases tabel yang berbeda untuk tabel yang sama.

### 25. SQL UNION

Kata kunci Union difungsikan untuk menggabungkan kumpulan hasil dari dua atau lebih pernyataan select.

Ada beberapa hal yang perlu diperhatikan dalam penggunaan operator union yaitu :

- Setiap kolom harus memiliki tipe data yang serupa
- Kolom yang ada disetiap pernyataan pilih harus dalam urutan yang sama
- Setiap pernyataan pilih dalam operator union harus memiliki jumlah kolom yang sama.

Contoh SQL Union

```
Select no_rekening (s) From rekening
union
Select no_rekening (s) From transaksi
```

## 26. SQL GROUP BY

Kata kunci Group By digunakan untuk mengelompokkan baris yang memiliki nilai yang sama kedalam baris ringkasan. Pernyataan group by ini sering digunakan dengan fungsi agregat (min, max, sum, avg, count) untuk mengelompokkan hasil yang ditetapkan oleh satu atau lebih kolom.

### Contoh SQL Group By

```
Select no_rekening (s)

From rekening

Where kondisi

Group by akun (s)

Order by nama-nasabah (s);
```

## 27. SQL HAVING

Klausula Having muncul untuk ditambahkan ke SQL karena klausula where tidak dapat digunakan pada fungsi agregat. Jika klausula where dan having muncul pada permohonan yang sama, maka SQL mengaplikasikan klausula where terlebih dahulu. Rekaman yang didapatkan dengan klausula where kemudian dikelompokkan dengan klausula group by. SQL kemudian mengaplikasikan klausula having kesetiap kelompok-kelompok yang tidak memenuhi syarat klausula having.

### Contoh SQL Having

```
Select nasabah.nama_nasabah, avg (saldo)

From rekening, transaksi, nasabah

Where nasabah.nama_nasabah = rekening.no_rekening and transaksi.no_transaksi
and akun = 'deposito'

Group by nasabah.nama_nasabah

Having count(distinct rekening.no_rekening) >= 3
```

## 28. SQL EXISTS

Klausula SQL Exists berfungsi untuk menguji rekaman apapun yang ada di subquery. Operator ini ada untuk mengembalikan nilai yang benar jika subquery mengembalikan satu atau lebih dari rekaman.

#### Contoh SQL Exists

```
Select no_rekening (s)
From nama_nasabah
Where exists
(select no_rekening from nama_nasabah where kondisi);
```

#### 29. SQL ANY, ALL

Kata kunci Any dan All digunakan dengan klausa where dan having. Untuk operator any lebih spesifiknya untuk mengembalikan benar jika salah satu nilai subquery memenuhi kondisi. Dan untuk operator all difungsikan kembali benar jika semua nilai subquery memenuhi kondisi.

#### Contoh SQL Any

```
Select no_rekening (s)
From nama_nasabah
Where no_rekening operator ANY
(select no_rekening from nama_nasabah where kondisi);
```

#### Contoh SQL All

```
Select no_rekening (s)
From nama_nasabah
Where no_rekening operator ALL
(select no_rekening from nama_nasabah where kondisi);
```

#### 30. SQL SELECT INTO

Klausa Select Into digunakan untuk menyalin data dari tabel satu ke tabel baru.

#### Contoh SQL Select Into

```
Select *
Into newtable [in dbtablebaru]
From oldtable
Where kondisi;
```

### 31. SQL INSERT INTO SELECT

SQL Insert into select digunakan untuk mengcopy data yang ada dari suatu tabel dan menyisipkannya ke tabel lain. Pernyataan insert into ini mengharuskan semua jenis data yang ada dalam tabel sumber dan target sama-sama cocok.

Contoh SQL Insert Into Select dengan menyalin semua kolom dari satu tabel ke tabel lainnya

```
insert into rekening  
  
select * from nasabah  
  
Where kondisi;
```

### 32. SQL CASE

SQL Case digunakan untuk melewati kondisi dan mengembalikan nilai saat kondisi pertama terpenuhi (seperti pernyataan IF – THEN – ELSE). Ketika suatu kondisi benar, maka itu akan berhenti membaca dan mengembalikan hasilnya. Dan jika tidak ada kondisi yang benar, maka ini akan mengembalikan nilai dalam klausa Else. Jika tidak ada bagian Else dan tidak ada kondisi yang benar maka hasilnya adalah null.

Contoh SQL Case

```
Case  
  
When kondisi1 then result1  
  
When kondisi2 then result2  
  
Else result  
  
End;
```

### 33. SQL NULL FUNCTIONS

Dalam SQL Null fungsional terdapat beberapa bagian yang akan dibahas yaitu fungsi SQL IfNull, fungsi SQL IsNull, fungsi SQL Coalesce dan fungsi SQL NVL. Jika salah satu nilai adalah Null maka hasilnya adalah null.

Fungsi SQL IfNull adalah memungkinkan untuk mengembalikan nilai alternatif jika ekspresi adalah null. Fungsi IsNull pada MS Access yaitu mengembalikan true (-1) jika ekspresi adalah nilai null, jika tidak false (0). Sedangkan fungsi Oracle NVL adalah mencapai hasil yang sama.

Contoh SQL Null Function (salah satu contoh dari fungsi)

```
Select rekening, unitprice * (CabangUnit + UnitPusat)  
  
From nasabah;
```

#### 34. SQL STORED PROCEDURES

Prosedur penyimpanan SQL untuk SQL Server. Prosedur penyimpanan merupakan kode SQL yang disiapkan untuk disimpan sehingga kode tersebut dapat digunakan secara berulang kali.

Jika memiliki fitur SQL yang dapat ditulis berulang kali maka, simpan sebagai prosedur tersimpan agar dapat dipanggil untuk menjalankannya sewaktu-waktu. Dan dapat juga menerapkan nilai parameter keprosedur tersimpan sebagai tata tertib dan dapat bertindak berdasarkan parameter yang diteruskan.

Contoh sintaks prosedur tersimpan

```
Create procedure new_name as
```

```
Sql statement
```

```
Go;
```

Untuk menjalankan prosedur tersimpan yaitu

```
Exec procedure_name;
```

#### 35. SQL COMMENTS

SQL Komentar digunakan untuk menjelaskan bagian dari pernyataan SQL atau untuk mencegah eksekusi pernyataan SQL. Pada klausa komentar baris tunggal dimulai dengan teks apapun diantara – dan akhir baris akan diabaikan atau tidak akan dieksekusi.

Contoh penggunaan salah satu SQL Comment

```
--select all;
```

```
Select * from customers;
```

#### 36. SQL OPERATORS

- Operator aritmatika SQL yaitu (+, -, \*, /, %).
- Operator SQL Bitwise : (&, |, ^).
- Perbandingan operator SQL : (=, >, <, >=, <=, <>).
- Operator senyawa SQL (+=, -=, \*=, /=, %=, &=, ^=, |=).
- Operator logika SQL : (all, and, any, between, exists, in, like, not, or, some).

#### 39. SQL DATABASE

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Database adalah representasi kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.

Database merupakan sekumpulan informasi yang saling berkaitan pada suatu subjek tertentu pada tujuan tertentu pula. Database adalah susunan record data operasional lengkap dari suatu organisasi atau perusahaan, yang diorganisir dan disimpan secara terintegrasi dengan menggunakan metode tertentu dalam komputer sehingga mampu memenuhi informasi yang optimal yang dibutuhkan oleh para pengguna.

#### Contoh SQL Database

Menampilkan database

Show databases;

Mengaktifkan database

use nama\_db;

#### 40. SQL CREATE DB

Create database statement adalah blok kode yang berisi perintah untuk membuat sebuah database baru. Database merupakan kumpulan data yang saling berkaitan dan disusun sedemikian rupa, yang dibuat dengan tujuan untuk memudahkan pengelolaan kumpulan data yang terdapat di dalamnya.

Contoh membuat database :

create database nama\_db;

#### 41. SQL DROP DB

Drop Database merupakan statement yang digunakan untuk menghapus database sekaligus semua tabel yang ada didalamnya. Oleh karenanya, berhati-hatilah ketika akan mengeksekusi perintah ini. Anda harus memiliki hak akses (*drop privilege*) untuk dapat menjalankan perintah ini.

Contoh menghapus database :

Drop database nama\_db;

#### 42. SQL BACKUP DB

Backup secara sederhana diartikan sebagai proses copy data operasional ke media lain. Pilihan metode dan strategi backup yang digunakan tergantung dari kebutuhan dan kondisi. Pernyataan Backup Database yang digunakan di SQL Server berfungsi untuk membuat cadangan penuh dari database SQL yang ada

Contoh sintaks SQL Backup DB

Backup database nama\_db

To disk = 'filepath';



Backup SQL dengan pernyataan differensial yaitu sebuah cadangan-cadangan hanya membuat cadangan dari database yang telah diubah sejak cadangan penuh terakhir.

Backup database nama\_db

To disk = 'filepath'

With differential;

#### 43. SQL CREATE TABLE

SQL Create table atau pernyataan create tabel digunakan untuk membuat suatu tabel baru di dalam database.

Contoh SQL Create table :

Create table rekening(

No\_rekening varchar (15),

Akun varchar (10))

Engine=innodb;

#### 44. SQL DROP TABLE

Drop Table dalam Bahasa Indonesia yang artinya adalah hapus tabel, digunakan untuk menghapus tabel-tabel dalam sebuah database. Perintah ini akan menghapus seluruh data/kolom. Perlu dipahami, perintah ini bukan hanya membersihkan tabel/menghapus kolomnya saja, melainkan juga nama tabel dan seluruh isi konten-konten didalamnya. Oleh sebab itu, berhati-hatilah dalam menjalankan perintah ini.

Contoh SQL Drop Table :

Drop table nama\_table;

#### 45. SQL ALTER TABLE

Alter Table adalah fungsi atau Query SQL yang digunakan untuk mengubah struktur table pada suatu database. Pernyataan Alter Table digunakan untuk menambahkan, menghapus, atau memodifikasi kolom dalam tabel yang ada. Pernyataan Alter Table juga digunakan untuk menambah dan menghilangkan berbagai kendala pada tabel yang ada.

Contoh SQL Alter table :

Alter table rekening

Add id char;

#### 46. SQL CONSTRAINTS

Constraint adalah aturan atau batasan yang diterapkan di table untuk menjaga konsistensi dan integritas data. constraint bisa di definisikan bersamaan dengan Create Table atau setelah table dibuat menggunakan perintah Alter Table ... Add Constraint. ada lima jenis constraint di Oracle, yaitu Primary Key, Unique, Check, Not null, dan Foreign Key. Constraint sebaiknya diberi nama pada saat pendefinisian, karena jika tidak diberi nama akan diberi nama oleh sistem dengan diawali dengan SYS\_Cn.

#### 47. SQL NOT NULL

Not Null merupakan constraint yang digunakan untuk menjamin pengisian record ke sebuah tabel agar nilai record tersebut harus berisi data. Sebuah kolom jika di berikan constraint Not Null, maka kolom tersebut harus berisikan nilai untuk recordnya dan tidak boleh di kosongkan. Not Null hanya bisa di definisikan dalam sebuah tabel pada level kolom dan tidak bisa pada level tabel. Perbedaan antara Not Null dan Primary Key adalah bahwa setiap kolom yang dijadikan Primary Key pasti Not Null dan tidak berlaku sebaliknya.

Contoh SQL Not Null :

```
Create table nasabah(  
  
    Id_nasabah varchar (15) not null,  
  
    Nama_nasabah varchar (20) not null)  
  
Engine= innodb;
```

#### 48. SQL UNIQUE

Constraint Unique merupakan sebuah constraint yang akan membatasi pengisian record yang sama kedalam sebuah kolom jika kolom tersebut di berikan constraint Unique dalam sebuah tabel. Constraint ini hampir sama dengan Primary Key, yaitu menjamin bahwa setiap nilai record yang ada dalam sebuah kolom Unique tidak boleh ada yang sama (unik satu sama lain).

Unique dapat dibuat pada level kolom ataupun level tabel. Sama dengan constraint Primary Key, Foreign Key, Check.

Contoh SQL Unique :

```
Create table transaksi (  
  
    Id_transaksi varchar (20) not null unique,  
  
    );
```

#### 49. SQL PRIMARY KEY

Primary Key merupakan suatu atribut atau satu set minimal atribut yang tidak hanya menjelaskan secara unik suatu kejadian spesifik tetapi juga dapat mewakili setiap kejadian-kejadian dari suatu kejadian.

Nilai field yang menjadi primary key yaitu :

- a. Unik atau tidak boleh dalam bentuk yang ganda
- b. Tidak boleh null (kosong, tidak diketahui, tidak dapat ditentukan / ditemukan)
- c. Key tersebut lebih natural untuk dijadikan acuan database tanpa data apapun yang asing.

Contoh SQL Primary key :

```
Create table transaksi (  
  
    Id_transaksi varchar (20) not null,  
  
    Periode_transaksi date,  
  
    Primary key (id_transaksi)  
  
);
```

#### 50. SQL FOREIGN KEY

Foreign Key dalam Bahasa Indonesia yang artinya adalah kunci asing. Penjelasan tersebut juga berlaku dalam pengolahan relasional database. Foreign Key merupakan sebuah atribut atau gabungan atribut-atribut yang terdapat dalam suatu tabel yang digunakan untuk menciptakan hubungan atau relasi antara dua tabel.

Dalam sebuah hubungan database, penggunaan foreign key merujuk pada suatu kunci primer yang terdapat pada tabel pertama atau dikenal juga dengan tabel master.

Nilai yang menjadi Foreign Key adalah :

- a. Foreign key merupakan suatu field dalam satu tabel yang digunakan untuk menghubungkan dua tabel. Dengan begitu, foreign key merujuk pada suatu kolom yang unik untuk mengidentifikasi baris dari satu tabel yang sama atau berbeda.
- b. Berbeda dengan primary key yang nilainya tidak boleh Null. Foreign key dapat menerima nilai Null. Oleh karena itu, foreign key tidak memiliki fungsi untuk mengidentifikasi record yang terdapat dalam tabel. Selain itu, foreign key juga dapat menerima nilai yang duplikat.
- c. Di dalam foreign key tidak terdapat *clustered index*. Namun, Anda dapat melakukannya dengan manual.
- d. Anda tidak dapat menambahkan nilai ke dalam foreign key, jika nilai tersebut tidak tersedia di kolom primary key. Sebab, syarat nilai dalam foreign key merupakan nilai rujukan dari kolom primary key.

- e. Anda dapat menghapus suatu nilai yang berasal dari kolom foreign key tanpa mengganggu record yang lain, dan juga tanpa harus mengetahui apakah nilai tersebut merupakan nilai yang dirujuk dari primary key atau tidak.

Contoh SQL Foreign Key :

```
Create table transaksi (  
  
    Id_transaksi varchar (20) not null,  
  
    Periode_transaksi date,  
  
    Primary key (id_transaksi)  
  
    Foreign key (periode_transaksi)  
  
    References periode(periode_transaksi)  
  
);
```

## 51. SQL CHECK

Check fungsinya untuk pengecekan apakah sebuah kolom memenuhi sebuah kondisi spesifik yang dievaluasi dengan nilai boolean. Jika evaluasi menghasilkan nilai false, maka oracle akan mengeluarkan eksepsi, dan klausa insert atau update akan gagal.

Contoh SQL Check :

```
Create table survey (  
  
    Nama varchar (50) not null,  
  
    Usia int,  
  
    Check (usia>=18)  
  
);
```

## 52. SQL DEFAULT

Pernyataan default digunakan untuk memberikan nilai default (nilai kosong) untuk kolom. Nilai default akan ditambahkan kesemua rekaman baru jika tidak ada nilai lain yang ditentukan.

Contoh SQL Default :

```
Create table survey (  
  
    Nama varchar (50) not null,  
  
    Usia int,
```

Kota varchar (100) default

‘sandnes’

);

### 53. SQL INDEX

Index adalah objek schema yang berisi catatan dari nilai-nilai yang muncul pada satu kolom atau kombinasi kolom di index dari sebuah tabel. Index dibuat untuk mempercepat pengaksesan data pada suatu tabel. Index ini dibuat berdasarkan pada field dari sebuah tabel. Index bisa dibuat secara otomatis untuk constraint primary key atau unique key dan secara manual melalui Create Index statement.

Contoh SQL Index :

Create index new\_index

On nama\_table (satu, dua);

### 54. SQL AUTO INCREMENT

Auto Increment merupakan fitur pada database yang memungkinkan pengguna membuat angka unik yang dihasilkan secara otomatis dan berurut untuk setiap catatan baru yang di input kedalam table database tersebut. Dan kebanyakan orang menggunakan angka unik Auto Increment ini sebagai Primary Key pada tabel database tersebut.

Fungsi dari Auto Increment itu sendiri adalah membuat angka berurut dari mulai terkecil sampai terbesar dan tidak perlu diisi secara manual ketika memasukkan data ke table contoh Tutorial.

Contoh SQL Auto Increment :

Create table transaksi (

Id\_transaksi varchar (20) not null auto increment,

Periode\_transaksi date,

Primary key (id\_transaksi)

);

### 55. SQL DATES

Kini SQL menghadirkan tipe data tanggal untuk menyimpan tanggal atau nilai tanggal / waktu dalam database. Format penulisan tipe data tanggal yaitu : YYYY-MM-DD. Format penulisan tipe data datetime yaitu : YYYY-MM-DD HH: MI: SS. Format penulisan timestamp yaitu : TTTT-BB-HH JJ: MI: SS. Format penulisan tahun yaitu : YYYY atau YY.

Contoh SQL Dates :

```
Select * from rekening where rekeningDate='2001-08-06'
```

## 56. SQL VIEWS

Views merupakan tabel virtual (yang bukan dari tabel sebenarnya) dibuat dari beberapa tabel atau view lain. Views tidak memiliki data sendiri, tetapi data-datanya berasal dari tabel-tabel atau view lain. Views dibuat atau dibangun dengan Select statement. Di dalam query Select tersebut kita bias menambahkan Order By statement untuk mengurutkan data atau tidak sama sekali. View dapat digunakan untuk memudahkan atau menyederhanakan data yang ditampilkan, mencegah menampilkan data yang penting dengan menyembunyikan nama kolomnya.

Kegunaan dari views adalah :

- a. Menyembunyikan kolom atau baris
- b. Menampilkan hasil dari penghitungan
- c. Menyembunyikan sintak SQL yang rumit
- d. Fungsi Built-in Layer
- e. Menyediakan level isolasi antara data tabel dan View data pengguna
- f. Memberikan proses permission yang berbeda untuk view yang berbeda dari tabel yang sama
- g. Memberikan trigger berbeda pada view yang berbeda dari tabel yang sama
- h. Membuat View

Contoh SQL Views :

```
Create view new_view as
```

```
Select satu,dua,
```

```
From new_table
```

```
Where kondisi;
```

## 57. SQL INJECTION

Injection atau biasa yang dikenal dengan sebutan SQL adalah suatu teknik penyerangan web dengan menggunakan kode SQL yang berbahaya untuk memanipulasi database. Tindakan ini termasuk kedalam kategori *hacking*, dimana penyerang akan mencari celah keamanan dari suatu website.

Sebagai contoh sederhana, biasanya pengguna diminta untuk memasukkan username dan password yang hanya dapat berupa karakter angka dan huruf. Sedangkan user memasukkan karakter lain, seperti petik tunggal ('), strip (-) atau tanda (=). Karakter-karakter tersebut merupakan bagian dari perintah umum yang digunakan untuk dapat berinteraksi dengan database. Keadaan itulah yang dimanfaatkan oleh penyerang untuk dapat mengakses informasi yang tersimpan didalam database.

Suatu website dibangun dengan menggunakan bahasa pemrograman, seperti HTML, PHP, ASP dan lainnya. Pada umumnya, beberapa bahasa pemrograman seperti

PHP atau Perl akan melakukan eksekusi perintah setelah menerima inputan dari pengguna tanpa adanya *filter* terlebih dahulu. Oleh karena itu, Anda perlu berhati-hati dengan melakukan pemilihan terhadap jenis database yang akan digunakan. Sebab, ada jenis database yang memberikan hak terhadap *low privileged users* untuk dapat mengakses password user lain yang terdapat didalam database.

Contoh SQL Injection :

```
Select * from users where userId = 100 Or 1=0;
```

## 58. SQL HOSTING

SQL dapat dihubungkan ke server web dengan menggunakan klausa hosting. Jika ingin situs web dapat menyimpan dan mengambil data dari database, maka web server harus memiliki akses kedalam sistem yang menggunakan Bahasa SQL. Database SQL hosting yang paling umum digunakan adalah MS SQL Server, Oracle, MySQL, dan MS Access.

Microsoft SQL Server merupakan database perangkat lunak terpopuler dengan lalu lintas tingkat tinggi, serta sistem databasenya yang sangat kuat dan berfitur lengkap.

## 59. SQL DATA TYPES

Tipe data digunakan untuk mendefinisikan suatu field atau kolom. Setiap kolom yang dibuat harus didefinisikan terlebih dahulu. Jenis – jenis tipe data ada bermacam – macam. Bisa numerik yang digunakan untuk angka dan proses perhitungan, bisa karakter / teks, tanggal atau biner.

Jenis-jenis tipe data yang ada dalam SQL :

- Tipe data String : char, varchar, binary, varbinary, tinyblob, tinytext, text, blob, mediumtext, mediumblob, longtext, longblob, enum, set.
- Tipe data numerik : bit, tinyint, bool, Boolean, smallint, mediumint, int, integer, bigint, float (d), float (p), double, double precision, decimal, dec.
- Tipe data tanggal dan waktu : date, datetime, timestamp, time, year.

## 60. SQL KEYWORD

Keyword	Description
<u>ADD</u>	Menambahkan kolom di tabel yang sudah ada
<u>ADD CONSTRAINT</u>	Menambahkan batasan setelah tabel dibuat
<u>ALTER</u>	Menambahkan, menghapus, atau mengubah kolom dalam tabel, atau mengubah tipe data kolom dalam tabel
<u>ALTER COLUMN</u>	Mengubah tipe data kolom dalam tabel
<u>ALTER TABLE</u>	Menambahkan, menghapus, atau mengubah kolom dalam tabel
<u>ALL</u>	Mengembalikan nilai benar jika semua nilai subkueri memenuhi kondisi
<u>AND</u>	Hanya menyertakan baris yang kedua ketentuannya benar
<u>ANY</u>	Mengembalikan nilai benar jika salah satu nilai subkueri memenuhi ketentuan

<u>AS</u>	Mengganti nama kolom atau tabel dengan alias
<u>ASC</u>	Mengurutkan hasil yang ditetapkan dalam urutan menaik
<u>BACKUP DATABASE</u>	Membuat cadangan dari database yang ada
<u>BETWEEN</u>	Memilih nilai dalam rentang tertentu
<u>CASE</u>	Menciptakan keluaran yang berbeda berdasarkan kondisi
<u>CHECK</u>	Batasan yang membatasi nilai yang dapat ditempatkan di kolom
<u>COLUMN</u>	Mengubah tipe data kolom atau menghapus kolom dalam tabel
<u>CONSTRAINT</u>	Menambahkan atau menghapus batasan
<u>CREATE</u>	Membuat database, indeks, tampilan, tabel, atau prosedur
<u>CREATE DATABASE</u>	Membuat database SQL baru
<u>CREATE INDEX</u>	Membuat indeks di atas tabel (memungkinkan nilai duplikat)
<u>CREATE OR REPLACE VIEW</u>	Memperbarui tampilan
<u>CREATE TABLE</u>	Membuat tabel baru di database
<u>CREATE PROCEDURE</u>	Membuat prosedur tersimpan
<u>CREATE UNIQUE INDEX</u>	Membuat indeks unik di atas tabel (tidak ada nilai duplikat)
<u>CREATE VIEW</u>	Membuat tampilan berdasarkan kumpulan hasil dari pernyataan SELECT
<u>DATABASE</u>	Membuat atau menghapus database SQL
<u>DEFAULT</u>	Batasan yang memberikan nilai default untuk kolom
<u>DELETE</u>	Menghapus baris dari tabel
<u>DESC</u>	Mengurutkan hasil dalam urutan menurun
<u>DISTINCT</u>	Hanya memilih nilai yang berbeda (berbeda)
<u>DROP</u>	Menghapus kolom, batasan, database, indeks, tabel, atau tampilan
<u>DROP COLUMN</u>	Menghapus kolom di tabel
<u>DROP CONSTRAINT</u>	Menghapus kendala UNIQUE, PRIMARY KEY, FOREIGN KEY, atau CHECK
<u>DROP DATABASE</u>	Menghapus database SQL yang sudah ada
<u>DROP DEFAULT</u>	Menghapus batasan DEFAULT
<u>DROP INDEX</u>	Menghapus indeks dalam sebuah tabel
<u>DROP TABLE</u>	Menghapus tabel yang sudah ada di database
<u>DROP VIEW</u>	Menghapus tampilan
<u>EXEC</u>	Menjalankan prosedur tersimpan
<u>EXISTS</u>	Menguji keberadaan record apa pun di subquery
<u>FOREIGN KEY</u>	Batasan yang merupakan kunci yang digunakan untuk menghubungkan dua tabel bersama-sama
<u>FROM</u>	Menentukan tabel mana yang datanya akan dipilih atau dihapus
<u>FULL OUTER JOIN</u>	Mengembalikan semua baris jika ada kecocokan di tabel kiri atau kanan
<u>GROUP BY</u>	Kelompokkan kumpulan hasil (digunakan dengan fungsi agregat: COUNT, MAX, MIN, SUM, AVG)
<u>HAVING</u>	Digunakan sebagai pengganti WHERE dengan fungsi agregat



<u>IN</u>	Memungkinkan Anda menentukan beberapa nilai dalam klausa WHERE
<u>INDEX</u>	Membuat atau menghapus indeks dalam tabel
<u>INNER JOIN</u>	Mengembalikan baris yang memiliki nilai yang cocok di kedua tabel
<u>INSERT INTO</u>	Menyisipkan baris baru dalam tabel
<u>INSERT INTO SELECT</u>	Menyalin data dari satu tabel ke tabel lain
<u>IS NULL</u>	Menguji nilai kosong
<u>IS NOT NULL</u>	Menguji nilai yang tidak kosong
<u>JOIN</u>	Menggabungkan tabel
<u>LEFT JOIN</u>	Mengembalikan semua baris dari tabel kiri, dan baris yang cocok dari tabel kanan
<u>LIKE</u>	Mencari pola tertentu di kolom
<u>LIMIT</u>	Menentukan jumlah record untuk dikembalikan dalam set hasil
<u>NOT</u>	Hanya menyertakan baris yang kondisinya tidak benar
<u>NOT NULL</u>	Batasan yang memaksa kolom untuk tidak menerima nilai NULL
<u>OR</u>	Menyertakan baris yang salah satu kondisinya benar
<u>ORDER BY</u>	Mengurutkan hasil yang ditetapkan dalam urutan menaik atau menurun
<u>OUTER JOIN</u>	Mengembalikan semua baris jika ada kecocokan di tabel kiri atau kanan
<u>PRIMARY KEY</u>	Batasan yang secara unik mengidentifikasi setiap rekaman dalam tabel database
<u>PROCEDURE</u>	Sebuah prosedur yang tersimpan
<u>RIGHT JOIN</u>	Menampilkan semua baris dari tabel kanan, dan baris yang cocok dari tabel kiri
<u>ROWNUM</u>	Menentukan jumlah record untuk dikembalikan dalam set hasil
<u>SELECT</u>	Memilih data dari database
<u>SELECT DISTINCT</u>	Hanya memilih nilai yang berbeda (berbeda)
<u>SELECT INTO</u>	Menyalin data dari satu tabel ke tabel baru
<u>SELECT TOP</u>	Menentukan jumlah record untuk dikembalikan dalam set hasil
<u>SET</u>	Menentukan kolom dan nilai mana yang harus diperbarui dalam tabel
<u>TABLE</u>	Membuat tabel, atau menambahkan, menghapus, atau mengubah kolom dalam tabel, atau menghapus tabel atau data di dalam tabel
<u>TOP</u>	Menentukan jumlah record untuk dikembalikan dalam set hasil
<u>TRUNCATE TABLE</u>	Menghapus data di dalam tabel, tetapi bukan tabel itu sendiri
<u>UNION</u>	Menggabungkan kumpulan hasil dari dua atau lebih pernyataan SELECT (hanya nilai yang berbeda)
<u>UNION ALL</u>	Menggabungkan kumpulan hasil dari dua atau lebih pernyataan SELECT (memungkinkan nilai duplikat)
<u>UNIQUE</u>	Batasan yang memastikan bahwa semua nilai dalam kolom unik
<u>UPDATE</u>	Memperbarui baris yang ada dalam sebuah tabel
<u>VALUES</u>	Menentukan nilai pernyataan INSERT INTO
<u>VIEW</u>	Membuat, memperbarui, atau menghapus tampilan
<u>WHERE</u>	Memfilter kumpulan hasil untuk menyertakan hanya rekaman yang memenuhi kondisi tertentu

## 61. MYSQL FUNCTIONS

Secara garis besar, fungsi dari MySQL adalah untuk membuat dan mengelola database pada sisi server yang memuat berbagai informasi dengan menggunakan bahasa SQL. Fungsi lain yang dimiliki adalah memudahkan pengguna dalam mengakses data berisi informasi dalam bentuk String (teks), yang dapat diakses secara personal maupun publik dalam web.

Hampir seluruh penyedia server web atau host menyediakan fasilitas untuk MySQL dalam pengembangan aplikasi berbasis website untuk dikelola oleh web developer. Kemudian, antarmuka dari MySQL adalah PHP MyAdmin. Yang berfungsi untuk menghubungkan antara bahasa pemrograman PHP dengan MySQL untuk proses pengelolaan basis data pada web.

MySQL memiliki beberapa keistimewaan atau fungsi, antara lain:

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. Multi-user. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. 'Performance tuning', MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

## 62. SQL FUNCTIONAL SERVER

Fungsi SQL server adalah objek yang berguna dalam database. Fungsi SQL server merupakan cuplikan kode yang dapat dijalankan di SQL server. Dalam T-SQL, ada beberapa aturan saat membuat fungsi di SQL Server, yaitu:

1. Fungsi hanya bekerja dengan pernyataan select
2. Fungsi dikompilasi setiap saat
3. Fungsi hanya bekerja dengan parameter masukan
4. Fungsi harus mengembalikan nilai atau hasil
5. Pernyataan coba dan tangkap tidak digunakan dalam fungsi
6. Fungsi harus memiliki nama, dan nama fungsi tidak boleh dimulai dengan karakter khusus seperti @, \$, #, dan lainnya.
7. Fungsi dapat digunakan dimana saja di SQL, seperti AVG, Count, SUM, MIN, Date, dan lainnya.

Dalam SQL server, terdapat dua jenis fungsi yang ditentukan oleh pengguna dan sistem :

1. Fungsi buatan pengguna yaitu fungsi yang dibuat oleh pengguna itu sendiri
2. Fungsi yang ditentukan dalam sistem yaitu fungsi yang dibangun didalam database itu sendiri

#### 63. MS ACCESS FUNCTIONAL

Fungsi Microsoft access digunakan para programmer untuk membuat sebuah sistem batan sendiri. Fungsi MS access kerap digunakan sebagai *front-end* program sementara produk lain yang bertindak sebagai tabel *back-end* sebuah pemrograman. Dengan kata lain fungsi MS access dapat difungsikan sebagai alat pemrograman yang sederhana.

MS access adalah sistem management basis data (DBMS) dari Microsoft yang menggabungkan Microsoft jet database engine relasional dengan antar muka pengguna grafis da alat pengembangan perangkat lunak. MS access menyimpan data dalam formatnya sendiri berdasarkan pada access jet database engine. Hal itu juga dapat mengimpor atau menautkan langsung ke data yang disimpan dalam aplikasi database lainnya.

Berikut ada beberapa fungsi Microsoft access :

- a. Untuk keperluan sekolah
- b. Untuk keperluan bisnis
- c. Untuk keperluan kantor
- d. Untuk programmer.

