

PENJELASAN TENTANG SQL
MATA KULIAH
SISTEM BASIS DATA
NAMA DOSEN: MUSLIADI KH, S. Kom.



NAMA : YUSNIAR
NIM : 200250502070

UNIVERSITAS TOMAKAKA MAMUJU
FAKULTAS ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA

1. SQL HOME

Bahasa standar untuk memanipulasi dan mengakses database, contohnya :
Select + from costumer

2. SQL INTRO

Digunakan untuk membuat, menghapus, dan mengubah database.

3. SQL SYNTAX

Syntax adalah aturan menulis 'kalimat' agar mampu dimengerti dengan benar oleh bahasa pemrograman. Maksud adanya sintaks atau aturan ini adalah agar kode atau script yang ditulis dapat dibenarkan dan dimengerti oleh komputer.

4. SQL SELECT

Select adalah pernyataan yang digunakan untuk menampilkan data – data pada table-tabel yang terdapat dalam database.

SELECT Syntax

```
SELECT column1, column2, ...  
FROM table_name;
```

Di sini, kolom1, kolom2, adalah nama bidang dari tabel yang ingin kita pilih datanya. Jika kita ingin memilih semua bidang yang tersedia di tabel, gunakan sintaks berikut:

```
SELECT * FROM table_name;
```

5. SQL SELECT DISTINCT

SQL Select Distinct di gunakan untuk mengembalikan hanya nilai yang berbeda dari dalam sebuah tabel, dengan kata lain semua record duplikat (record dengan nilai yang sama) yang terdapat pada tabel akan di anggap sebagai satu record/nilai.

Select Distinct Syntax

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```

6. SQL WHERE SQL AND, OR, NOT

Where di gunakan untuk mem-filter hasil SELECT dengan mengekstrak record yang memenuhi persyaratan tertentu.

WHERE Syntax

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

AND, OR dan NOT merupakan perintah dasar SQL yang biasanya di kombinasikan dengan perintah WHERE.

Operator AND dan OR digunakan untuk memfilter record berdasarkan lebih dari satu kondisi:

- Operator AND menampilkan record jika semua kondisi yang dipisahkan oleh AND adalah TRUE.
- Operator OR menampilkan record jika salah satu kondisi yang dipisahkan oleh OR adalah TRUE.
- Operator NOT menampilkan record jika kondisi TIDAK BENAR.

AND Syntax

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 AND condition2 AND condition3 ...;
```

OR Syntax

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 OR condition2 OR condition3 ...;
```

NOT Syntax

```
SELECT column1, column2, ...  
FROM table_name  
WHERE NOT condition;
```

7. SQL ORDER BY

Order By digunakan untuk mengurutkan hasil dalam urutan naik(ascending) atau turun (descending). Order By mengurutkan record dalam urutan menaik secara

default. Untuk mengurutkan record dalam urutan menurun, gunakan kata kunci DESC.

Order By Syntax

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2, ... ASC|DESC;
```

8. SQL INSERT INTO

Insert Into digunakan untuk memasukkan record baru ke dalam tabel.

Insert Into Syntax

Insert Into dapat dibuat dengan dua cara:

- Tentukan nama kolom dan nilai yang akan disisipkan:

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

- Jika Anda menambahkan nilai untuk semua kolom pada tabel, Anda tidak perlu menentukan nama kolom dalam kueri SQL. Namun, pastikan urutan nilainya dalam urutan yang sama seperti kolom di tabel. Di sini, sintaks INSERT INTO adalah sebagai berikut:

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

9. SQL NULL VALUES

Bidang dengan nilai Null adalah bidang tanpa nilai. Jika bidang dalam tabel bersifat opsional, dimungkinkan untuk menyisipkan catatan baru atau memperbarui catatan tanpa menambahkan nilai ke bidang ini. Kemudian, field akan disimpan dengan nilai Null.

Nilai Null berbeda dari nilai nol atau bidang yang berisi spasi. Bidang dengan nilai NULL adalah bidang yang dibiarkan kosong selama pembuatan rekaman!

- Tidak mungkin menguji nilai NULL dengan operator perbandingan, seperti =, <, atau <>.

- Kita harus menggunakan operator IS NULL dan IS NOT NULL sebagai gantinya.

IS NULL Syntax

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NULL;
```

IS NOT NULL Syntax

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NOT NULL;
```

10. SQL UPDATE

UPDATE di gunakan untuk memperbarui atau mengubah nilai suatu record berdasarkan kriteria tertentu. Kita harus berhati-hati saat memperbarui catatan dalam tabel! Perhatikan klausa WHERE dalam pernyataan UPDATE. Klausa WHERE menentukan rekaman mana yang harus diperbarui. Jika Anda menghilangkan klausa WHERE, semua rekaman dalam tabel akan diperbarui!

UPDATE Syntax

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

11. SQL DELET

DELETE digunakan untuk menghapus record yang ada dalam tabel. Berhati-hatilah saat menghapus record dalam tabel! Perhatikan klausa WHERE dalam pernyataan DELETE. Klausa WHERE menentukan record mana yang harus dihapus. Jika Anda menghilangkan klausa WHERE, semua record dalam tabel akan dihapus!

DELETE Syntax

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

12. SQL TOP

SELECT TOP digunakan untuk menentukan jumlah record yang akan dikembalikan.
SELECT TOP berguna pada tabel besar dengan ribuan record.

Tidak semua sistem database mendukung klausa SELECT TOP.

13. SQL Server / MS Access Syntax:

```
SELECT TOP number|percent column_name(s)  
FROM table_name  
WHERE condition;
```

14. SQL MIN AND MAX

MIN di gunakan untuk mendapatkan nilai terkecil dari suatu kolom, dan fungsi MIN hanya menampilkan satu record saja yang memenuhi kriteria yang Anda tentukan. Sedangkan MAX di gunakan untuk mendapatkan nilai terbesar dari suatu kolom.

MIN Syntax

```
SELECT MIN(column_name)  
FROM table_name  
WHERE condition;
```

MAX Syntax

```
SELECT MAX(column_name)  
FROM table_name  
WHERE condition;
```

15. SQL COUNT, AVG, SUM

COUNT berfungsi mengembalikan jumlah baris yang cocok dengan kriteria yang ditentukan.

COUNT() Syntax

```
SELECT COUNT(column_name)  
FROM table_name  
WHERE condition;
```

AVG berfungsi mengembalikan nilai rata-rata kolom numerik.

AVG() Syntax

```
SELECT AVG(column_name)  
FROM table_name  
WHERE condition;
```

SUM berfungsi mengembalikan jumlah total kolom numerik.

SUM() Syntax

```
SELECT SUM(column_name)  
FROM table_name  
WHERE condition;
```

16. SQL LIKE

LIKE digunakan dalam klausa WHERE untuk mencari pola tertentu dalam kolom. Ada dua karakter pengganti yang sering digunakan terkait dengan operator LIKE:

- Tanda persen (%) mewakili nol, satu, atau beberapa karakter
- Tanda garis bawah (_) mewakili satu karakter tunggal

LIKE Syntax

```
SELECT column1, column2, ...  
FROM table_name  
WHERE columnN LIKE pattern;
```

17. SQL WILDCARD

Karakter wildcard digunakan untuk menggantikan satu atau lebih karakter dalam sebuah string. Karakter wildcard digunakan dengan operator LIKE. Operator LIKE digunakan dalam klausa WHERE untuk mencari pola tertentu dalam kolom.

Contoh

```
SELECT * FROM Customers  
WHERE City LIKE 'ber%';
```

18. SQL IN

Operator IN memungkinkan Anda untuk menentukan beberapa nilai dalam klausa WHERE. Operator IN adalah singkatan dari beberapa kondisi OR.

IN Syntax

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name IN (value1, value2, ...);
```

atau:

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name IN (SELECT STATEMENT);
```

19. SQL BETWEEN

BETWEEN digunakan untuk memilih nilai dalam rentang tertentu. Nilainya bisa berupa angka, teks, atau tanggal. BETWEEN bersifat inklusif: nilai awal dan akhir disertakan.

BETWEEN Syntax

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name BETWEEN value1 AND value2;
```

20. SQL ALIASES

Aliases digunakan untuk memberi tabel, atau kolom dalam tabel, nama sementara. Aliases sering digunakan untuk membuat nama kolom lebih mudah dibaca. Aliases dibuat dengan kata kunci AS.

Alias Column Syntax

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

Alias Table Syntax

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


21. SQL JOINS

JOINS digunakan untuk menggabungkan baris dari dua atau lebih tabel, berdasarkan kolom terkait di antara mereka.

22. SQL INNER JOIN

Inner Join digunakan untuk mengembalikan records yang memiliki nilai yang cocok di kedua tabel.

INNER JOIN Syntax

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

23. SQL LEFT JOIN

Left Join digunakan untuk mengembalikan semua catatan dari tabel kiri (tabel1), dan catatan yang cocok dari tabel kanan (tabel2). Hasilnya adalah 0 record dari sisi kanan, jika tidak ada yang cocok.

LEFT JOIN Syntax

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

24. SQL RIGHT JOIN

Right Join digunakan untuk mengembalikan semua record dari tabel kanan (table2), dan record yang cocok dari tabel kiri (table1). Hasilnya adalah 0 record dari sisi kiri, jika tidak ada yang cocok.

RIGHT JOIN Syntax

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

25. SQL FULL

Full Outer Join berfungsi untuk mengembalikan semua rekaman ketika ada kecocokan dalam catatan tabel kiri (tabel1) atau kanan (tabel2).

FULL OUTER JOIN Syntax

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```

26. SQL SELF JOIN

SELF JOIN adalah penggabungan reguler, tetapi tabel bergabung dengan tabel itu sendiri.

Self Join Syntax

```
SELECT column_name(s)
FROM table1 T1, table1 T2
WHERE condition;
```

27. SQL UNION

UNION digunakan untuk menggabungkan kumpulan hasil dari dua atau lebih pernyataan SELECT. Setiap pernyataan SELECT dalam UNION harus memiliki jumlah kolom yang sama. Kolom juga harus memiliki tipe data yang serupa. Kolom di setiap pernyataan SELECT juga harus dalam urutan yang sama.

UNION Syntax

```
SELECT column_name(s) FROM table1
UNION
SELECT column_name(s) FROM table2;
```

28. SQL GROUP BY

GROUP BY digunakan untuk mengelompokkan baris yang memiliki nilai yang sama ke dalam baris ringkasan. Pernyataan GROUP BY sering digunakan dengan fungsi agregat (COUNT (), MAX (), MIN (), SUM (), AVG ()) untuk mengelompokkan hasil yang ditetapkan oleh satu atau lebih kolom.

GROUP BY Syntax

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

29. SQL HAVING

HAVING ditambahkan ke SQL karena kata kunci WHERE tidak dapat digunakan dengan fungsi agregat.

HAVING Syntax

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

30. SQL EXISTS

EXISTS digunakan untuk menguji keberadaan record apa pun di subquery .EXISTS mengembalikan nilai jika subquery. mengembalikan satu atau lebih record.

EXISTS Syntax

```
SELECT column_name(s)
FROM table_name
WHERE EXISTS
(SELECT column_name FROM table_name WHERE condition);
```

31. SQL ANY, ALL

ANY dan ALL memungkinkan kita melakukan perbandingan antara satu nilai kolom dan rentang nilai lainnya. ANY mengembalikan nilai boolean sebagai hasilnya, mengembalikan TRUE jika salah satu nilai subkueri memenuhi kondisi. ANY berarti bahwa kondisi tersebut akan menjadi benar jika operasi tersebut benar untuk salah satu nilai dalam rentang tersebut.

ANY Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ANY
      (SELECT column_name
       FROM table_name
       WHERE condition);
```

Operator ALL:

- mengembalikan nilai boolean sebagai hasilnya
- mengembalikan TRUE jika SEMUA nilai subkueri memenuhi kondisi
- digunakan dengan pernyataan SELECT, WHERE dan HAVING

ALL berarti bahwa kondisi akan menjadi benar hanya jika operasi itu benar untuk semua nilai dalam rentang tersebut.

ALL Syntax With SELECT

```
SELECT ALL column_name(s)
FROM table_name
WHERE condition;
```

32. SQL SELECT INTO

SELECT INTO digunakan untuk menyalin data dari satu tabel ke tabel baru.

SELECT INTO Syntax

Salin semua kolom ke dalam tabel baru:

```
SELECT *  
INTO newtable [IN externaldb]  
FROM oldtable  
WHERE condition;
```

Salin hanya beberapa kolom ke dalam tabel baru:

```
SELECT column1, column2, column3, ...  
INTO newtable [IN externaldb]  
FROM oldtable  
WHERE condition;
```

Di mana kondisi tabel baru akan dibuat dengan nama kolom dan tipe seperti yang ditentukan dalam tabel lama. Anda dapat membuat nama kolom baru menggunakan klausa AS.

33. SQL INSERT INTO SELECT

INSERT INTO SELECT digunakan untuk menyalin data dari satu tabel dan menyisipkannya ke tabel lain. Pernyataan INSERT INTO SELECT mengharuskan tipe data di tabel sumber dan target cocok.

INSERT INTO SELECT Syntax

Salin semua kolom dari satu tabel ke tabel lain:

```
INSERT INTO table2  
SELECT * FROM table1  
WHERE condition;
```

Salin hanya beberapa kolom dari satu tabel ke tabel lain:

```
INSERT INTO table2 (column1, column2, column3, ...)  
SELECT column1, column2, column3, ...  
FROM table1  
WHERE condition;
```

34. SQL CASE

CASE digunakan untuk melewati kondisi dan mengembalikan nilai saat kondisi pertama terpenuhi (seperti pernyataan if-then-else). Jadi, begitu suatu kondisi benar, itu akan berhenti membaca dan mengembalikan hasilnya. Jika tidak ada kondisi yang benar, ini mengembalikan nilai dalam klausa ELSE. Jika tidak ada bagian ELSE dan tidak ada kondisi yang benar, ia mengembalikan NULL.

CASE Syntax

CASE

```
WHEN condition1 THEN result1
WHEN condition2 THEN result2
WHEN conditionN THEN resultN
ELSE result
END;
```

35. SQL STORED PROCEDURES

Stored Procedures adalah kode SQL yang disiapkan yang dapat kita simpan, sehingga kode tersebut dapat digunakan kembali berulang kali.

Jadi jika kita memiliki kueri SQL yang kita tulis berulang kali, simpan sebagai Stored Procedures, lalu panggil saja untuk menjalankannya.

Kita juga dapat meneruskan parameter ke prosedur tersimpan, sehingga prosedur tersimpan dapat bertindak berdasarkan nilai parameter yang diteruskan.

Stored Procedure Syntax

```
CREATE PROCEDURE procedure_name
AS
sql_statement
GO;
```

36. SQL COMMENTS

Comments digunakan untuk menjelaskan bagian pernyataan SQL, atau untuk mencegah eksekusi pernyataan SQL.

37. SQL OPERATORS

Operator adalah kata atau karakter khusus yang digunakan terutama dalam klausa WHERE pernyataan SQL untuk melakukan operasi, seperti perbandingan dan operasi aritmatika. Operator ini digunakan untuk menentukan kondisi dalam pernyataan SQL dan berfungsi sebagai konjungsi untuk beberapa kondisi dalam sebuah pernyataan.

38. SQL CREATE DB

CREATE DATABASE adalah sebuah operator dalam SQL yang digunakan untuk membuat database baru.

Syntax

CREATE DATABASE *databasename*;

39. SQL DROP DB

DROP DATABASE adalah perintah yang di gunakan untuk menghapus data – data yang telah di simpan pada database.

Syntax

DROP DATABASE *databasename*;

40. SQL BACKUP DB

Backup DB adalah proses membuat data cadangan dengan cara menyalin atau membuat arsip data komputer sehingga data tersebut dapat digunakan kembali apabila terjadi kerusakan atau kehilangan data.

Syntax

```
BACKUP DATABASE databasename  
TO DISK = 'filepath';
```

41. SQL CREATE TABLE

CREATE TABLE adalah perintah dalam SQL yang digunakan untuk membuat tabel baru dalam database.

Syntax

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```

42. SQL DROP TABLE

DROP TABLE secara sederhana diartikan dengan hapus tabel, digunakan untuk menghapus tabel dalam sebuah database. Perintah ini akan menghapus seluruh data (kolom-kolom) berikut definisi yang ada dalam table tersebut.

Syntax

```
DROP TABLE table_name;
```

43. SQL ALTER TABLE

ALTER TABLE digunakan untuk menambah, menghapus, atau mengubah kolom dalam tabel yang sudah ada.

Pernyataan ALTER TABLE juga digunakan untuk menambah dan menghapus berbagai batasan pada tabel yang sudah ada.

Untuk menambahkan kolom dalam tabel, gunakan sintaks berikut:

```
ALTER TABLE table_name  
ADD column_name datatype;
```


44. SQL CONSTRAINTS

Constraints adalah sebuah opsi atau atribut yang berfungsi untuk membatasi nilai setiap data yang akan dimasukkan dalam suatu kolom di dalam tabel database

Constraints dapat ditentukan saat tabel dibuat dengan pernyataan CREATE TABLE, atau setelah tabel dibuat dengan pernyataan ALTER TABLE.

Syntax

```
CREATE TABLE table_name (  
    column1 datatype constraint,  
    column2 datatype constraint,  
    column3 datatype constraint,  
    ....  
);
```

45. SQL NOT NULL

NOT NULL merupakan batas 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.

Example

SQL berikut memastikan bahwa kolom "ID", "LastName", dan "FirstName" TIDAK akan menerima nilai NULL saat tabel "Persons" dibuat:

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255) NOT NULL,  
    Age int  
);
```

46. SQL UNIQUE

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).

Untuk memberi nama batasan UNIK, dan untuk menentukan batasan UNIK di beberapa kolom, gunakan sintaks SQL berikut:

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    CONSTRAINT UC_Person UNIQUE (ID,LastName)  
);
```

47. SQL PRIMARY KEY

Primary Key digunakan untuk menyelesaikan permasalahan dimana ingin membuat beberapa kolom data dengan isi yang berbeda dengan kolom lain, dengan kata lain hanya menginginkan tidak ada data yang sama atau kembar dari total kolom dalam baris data . Primary key berfungsi bertujuan agar data tidak kembar, dengan membuat sebuah kolom yang ditujukan sebagai kunci primer atau *primary key*. Primary key hanya boleh digunakan sekali saja. Primary key sebenarnya diciptakan untuk dijadikan kolom utama sebuah tabel, dengan demikian data pada kolom kunci utama tidak mungkin kembar.

SQL berikut membuat PRIMARY KEY di kolom "ID" ketika tabel "Persons" dibuat:

```
CREATE TABLE Persons (  
    ID int NOT NULL PRIMARY KEY,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int  
);
```

48. SQL FOREIGN KEY

Foreign key disebut juga sebagai constraint Referential Integrity, constraint ini memastikan bahwa data pada suatu tabel didefinisikan pada tabel lainnya dan mengikat kedua tabel tersebut dalam hubungan parent/child atau referenced/dependent.

SQL berikut membuat FOREIGN KEY di kolom "PersonID" ketika tabel "Pesanan" dibuat:

```
CREATE TABLE Orders (  
    OrderID int NOT NULL PRIMARY KEY,  
    OrderNumber int NOT NULL,  
    PersonID int FOREIGN KEY REFERENCES Persons(PersonID)  
);
```

49. SQL CHECK

CHECK berfungsi untuk membatasi nilai-nilai yang dapat dimasukkan ke dalam suatu kolom di dalam tabel. Sebagai contoh sederhana, kolom 'status_produk' nilainya dapat kita batasi misalnya hanya berupa "ready stock" dan "out stock" saja. Selain nilai tersebut server database akan menolak.

SQL berikut membuat batasan PERIKSA pada kolom "Usia" saat tabel "Orang" dibuat. Batasan PERIKSA memastikan bahwa usia seseorang harus berusia 18 tahun atau lebih:

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int CHECK (Age>=18)  
);
```

50. SQL DEFAULT

DEFAULT digunakan untuk menetapkan nilai default untuk kolom. Nilai default akan ditambahkan ke semua record baru, jika tidak ada nilai lain yang ditentukan.

SQL berikut menetapkan nilai DEFAULT untuk kolom "Kota" saat tabel "Orang" dibuat:

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    City varchar(255) DEFAULT 'Sandnes'  
);
```

51. SQL INDEX

INDEX berfungsi untuk mempercepat proses pengambilan data, pengurutan dan pencarian data dari suatu tabel. Data pada tabel yang sudah diindeks akan diurutkan berdasarkan kolom indeks. dengan demikian proses pencarian data dapat lebih cepat. Saat data baru sedikit mungkin pengindeksan belum terlalu berguna, namun jika data sudah mencapai ribuan atau bahkan ratusan ribu barulah terasa perbedaannya jika field tersebut belum diindeks.

Syntax

```
CREATE INDEX index_name  
ON table_name (column1, column2, ...);
```

52. SQL AUTO INCREMENT

AUTO INCREMENT digunakan apabila ingin menciptakan sebuah kolom yang dapat merekam data. dengan menggunakan fungsi ini, akan otomatis masuk sendiri tanpa harus dilakukan secara manual. Penambahan ini adalah penambahan otomatis yang sifatnya berurutan dari kecil hingga besar. *Auto increment* hanya dapat dibuat apabila data berupa angka atau integer.

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

Pernyataan SQL berikut mendefinisikan kolom "Personid" menjadi auto-increment primary key di tabel " Personid ":

```
CREATE TABLE Persons (  
    Personid int IDENTITY(1,1) PRIMARY KEY,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int  
);
```

53. SQL DATES

Dates atau tanggal merupakan Bagian tersulit saat bekerja dengan tanggal adalah memastikan bahwa format tanggal yang Anda coba sisipkan, cocok dengan format kolom tanggal dalam database. Selama data Anda hanya berisi bagian tanggal, kueri Anda akan berfungsi seperti yang diharapkan. Namun, jika melibatkan porsi waktu, itu menjadi lebih rumit.

SQL Server dilengkapi dengan tipe data berikut untuk menyimpan tanggal atau nilai tanggal / waktu dalam database:

- DATE - format YYYY-MM-DD
- DATETIME - format: YYYY-MM-DD HH:MI:SS
- SMALLDATETIME - format: YYYY-MM-DD HH:MI:SS
- TIMESTAMP - format: a unique number

54. SQL VIEWS

Dalam SQL, views adalah tabel virtual berdasarkan kumpulan hasil pernyataan SQL. views berisi baris dan kolom, seperti tabel nyata. Bidang dalam views adalah bidang dari satu atau beberapa tabel nyata dalam database. Anda dapat menambahkan pernyataan dan fungsi SQL ke views dan menyajikan data seolah-olah data tersebut berasal dari satu tabel. views dibuat dengan pernyataan `CREATE VIEW`.

CREATE VIEW Syntax

```
CREATE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

55. SQL INJECTION

SQL injection adalah teknik injeksi kode yang dapat merusak database Anda. SQL injection adalah salah satu teknik peretasan web yang paling umum. SQL injection adalah penempatan kode berbahaya dalam pernyataan SQL, melalui input halaman web.

Example

```
txtUserId = getRequestString("UserId");  
txtSQL = "SELECT * FROM Users WHERE UserId = " + txtUserId;
```

56. SQL HOSTING

Jika Anda ingin situs web Anda dapat menyimpan dan mengambil data dari database, server web Anda harus memiliki akses ke sistem database yang menggunakan bahasa SQL.

Jika server web Anda dihosting oleh Penyedia Layanan Internet (ISP), Anda harus mencari paket hosting SQL. Database SQL hosting yang paling umum adalah MS SQL Server, Oracle, MySQL, dan MS Access.

57. SQL DATA TYPES

Tipe data digunakan untuk mendefinisikan suatu field atau kolom. Setiap kolom yang dibuat harus didefinisikan terlebih dahulu. Tipe data adalah pedoman bagi SQL untuk memahami tipe data apa yang diharapkan di dalam setiap kolom, dan juga mengidentifikasi bagaimana SQL akan berinteraksi dengan data yang disimpan.

58. SQL KEYWORDS

Dalam SQL, Keywords adalah kata khusus yang digunakan untuk melakukan berbagai operasi dalam database. Ada banyak kata kunci dalam SQL dan karena SQL

tidak membedakan huruf besar / kecil, tidak masalah jika kita menggunakan misalnya SELECT atau select.

59. MYSQL FUNCTIONS

MySQL merupakan sebuah perangkat lunak system manajemen basis data SQL (bahasa inggris : data management system) atau DNMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public Licensi (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL . Tidak seperti Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan cipta untuk code sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah : david axmark, allan larsson, dan Michael “monthly widenius.

Kelebihan MySQL antara lain :

- free (bebas didownload)
- stabil dan Tangguh
- fleksibel dengan berbagai pemrograman
- Security yang baik
- dukungan dari banyak komunitas
- kemudahan management database
- mendukung transaksi
- perkembangan software yang cukup cepat.

Kekurangan :

- Tidak cocok untuk menangani data dengan jumlah yang besar, baik untuk menyimpan data maupun untuk memproses data.
- Memiliki keterbatasan kemampuan kinerja pada server ketika data yang disimpan telah melebihi batas maksimal kemampuan daya tampung server karena tidak menerapkan konsep Technology Cluster Server.

60. SQL SERVER FUNCTIONS

SQL Server merupakan sistem manajemen basis data relasional yang dirancang untuk aplikasi dengan arsitektur client server. Microsoft SQL merupakan salah satu produk unggulan dari Microsoft mengenai management dari basis data alias database, dimana software ini menggunakan Transact SQL sebagai kueri utamanya. Fitur yang terdapat pada Microsoft SQL Server yaitu memiliki kemampuan untuk membuat basis data clustering dan mirroring. Dalam Microsoft SQL Server juga sangat mendukung SQL sebagai bahasa untuk memproses query ke dalam. Microsoft SQL server sudah digunakan dalam kalangan umum.

SQL sendiri merupakan kependekan dari Structures Query Language, yang merupakan suatu jenis bahasa komputer atau kueri yang digunakan untuk mengakses data – data yang terdapat dan juga tersimpan di dalam sebuah basis data alias database. Untuk sekarang ini , hampir dari semua DBMS (database management system) sudah mempergunakan bahasa SQL untuk membantu mengimplementasikan dan juga mengkorelasikan dari pengolahan dan juga management dari sebuah database/ basis data.

Kelebihan :

- Cocok untuk perusahaan dengan skala kecil, menengah, dan besar sehingga mampu untuk mengolah data dengan jumlah yang besar.
- Memiliki kemampuan untuk management user dan tiap user bisa diatur hak akses terhadap suatu database oleh database administrator.
- Untuk diterapkan pada pembangunan suatu program aplikasi, akan mudah dalam melakukan koneksi dengan computer client yang pembangunan aplikasinya menggunakan software yang sama platform dengan MS-SQL, misalnya Microsoft Visual Basic.
- Memiliki tingkat pengamanan / security data yang baik.
- Memiliki kemampuan untuk back-up data, rollback data, dan recovery data.
- Memiliki kemampuan untuk membuat database mirroring dan clustering.

Kekurangan :

- Hanya dapat diimplementasikan pada 1 unit server, jika terdapat tambahan server maka hanya akan berfungsi sebagai pasif / standby server (tidak

memiliki kemampuan Technology Cluster Server seperti halnya pada DMBS Oracle).

- Hanya bisa berjalan pada satu platform system operasi yaitu Microsoft Windows.
- Merupakan software berlisensi dan berharga mahal untuk perusahaan skala kecil dan menengah.

61. MS ACCESS FUNCTIONS

Microsoft Access Functions adalah aplikasi yang berguna untuk membuat, mengolah, dan mengelola basis data atau lebih dikenal dengan database.

Fungsi/kegunaan utama dari Ms.Access yaitu untuk menangani proses manipulasi data dan pembuatan sebuah system. Hadirnya Microsoft Access dapat dimanfaatkan sebagai media untuk membuat aplikasi web dasar. Adapun fungsi lain dari program aplikasi ini antara lain :

- Untuk membuat program aplikasi jumlah peserta didik.
- Untuk membuat laporan keuangan.
- Untuk membuat program aplikasi gaji karyawan.
- Untuk membuat form dan sub form.
- Untuk membuat basis data.
- Untuk membuat program aplikasi penyimpan buku perpustakaan.
- Untuk membuat program aplikasi daftar hadir.
- Untuk membuat program aplikasi persediaan barang.

Kelebihan dan Kekurangan Microsoft Access

Kelebihan

1. Berbasis file sehingga lebih portabel (tinggal dicopy).
2. Lebih murah
3. Lebih mudah karena ditujukan ke desktop dan tool-tool yang memudahkan serta kemampuannya lebih dibatasi.
4. Mendukung SQL.
5. Mendukung Relational database (terbatas)

Kekurangan

1. Bukan merupakan database server sehingga tidak bisa/tdk cocok untuk keperluan enterprise dan terdistribusi (misalnya Web).
2. Bukan RDBMS murni
3. Tidak mendukung multithreaded
4. Tidak begitu cepat aksesnya (karena berbasis file)
5. Tidak bisa untuk Store procedure
6. Security kalah jauh dibanding SQL Server

Adapun manfaat microsoft access antara lain,

- Menghemat waktu dan tenaga.
- Mempermudah pelaksanaan suatu pekerjaan yang terasa berat dan melelahkan.

62. SQL QUICK REF

