Konsep Dasar

- Relatsional dan RDBMS
- User dan Schema Database
- Membuat Tabel dan Mendefinisikan Constraint
- Input Data ke dalam Tabel

1. Relasional Database dan RDBMS

Sebelum membahas berbagai jenis perintah SQL akan lebih baik kalau kita bicara tentang Relational Database dan Relational Database Management System atau biasa di kenal dengan RDBMS.

Relational Database sebenarnya adalah salah satu konsep penyimpanan data, sebelum konsep database relasional muncul sebenarnya sudah ada dua model database yaitu Network Database dan Hierarchie Database. Dalam database relasional, data disimpan dalam bentuk relasi atau tabel dua dimensi, dan antar tabel satu dengan tabel lainnya terdapat hubungan atau relationship, sehingga sering kita baca diberbagai literatur, database didefinisikan sebagai "kumpulan dari sejumlah tabel yang saling hubungan atau keterkaitan". Nah, kumpulan dari data yang diorganisasikan sebagai tabel tadi disimpan dalam bentuk data elektronik di dalam hardisk komputer. Untuk membuat struktur tabel, mengisi data ke tabel, mengubah data jika diperlukan dan menghapus data dari tabel diperlukan software. Software yang digunakan membuat tabel, isi data, ubah data dan hapus data disebut Relational Database Management
System atau dikenal dengan singkatan RDBMS sedangkan perintah yang digunakan untuk membuat tabel, isi, ubah dan hapus data disebut perintah SQL yang merupakan singkatan dari Structure Query Language. Jadi, setiap software RDBMS pasti bisa digunakan untuk menjalankan perintah SQL.

Sebenarnya fungsi RDBMS bukan cuma buat tabel, isi data, ubah dan hapus data, untuk manajemen data dalam skala besar dan agar bisa mendukung proses bisnis yang kontinyu dan real time suatu RDBMS dituntut untuk mempunyai kemampuan manajemen user dan keamanan data, backup dan recovery data serta kemampuan lainnya yang berkaitan dengan kecepatan pemrosesan data (performance).

Salah satu software RDBMS yang ada dipasaran saat ini dan cukup banyak digunakan adalah Oracle Database.

Berinteraksi dengan Database Oracle

Untuk mengakses data yang ada di database digunakan perintah SQL, perintah-perintah SQL ini ditulis atau diinput dengan tools yang sudah disediakan oleh Oracle yaitu SQL*PLus, iSQL*Plus dan SQL Developer. Perintah SQL dikelompokkan berdasarkan fungsinya sebagai berikut:

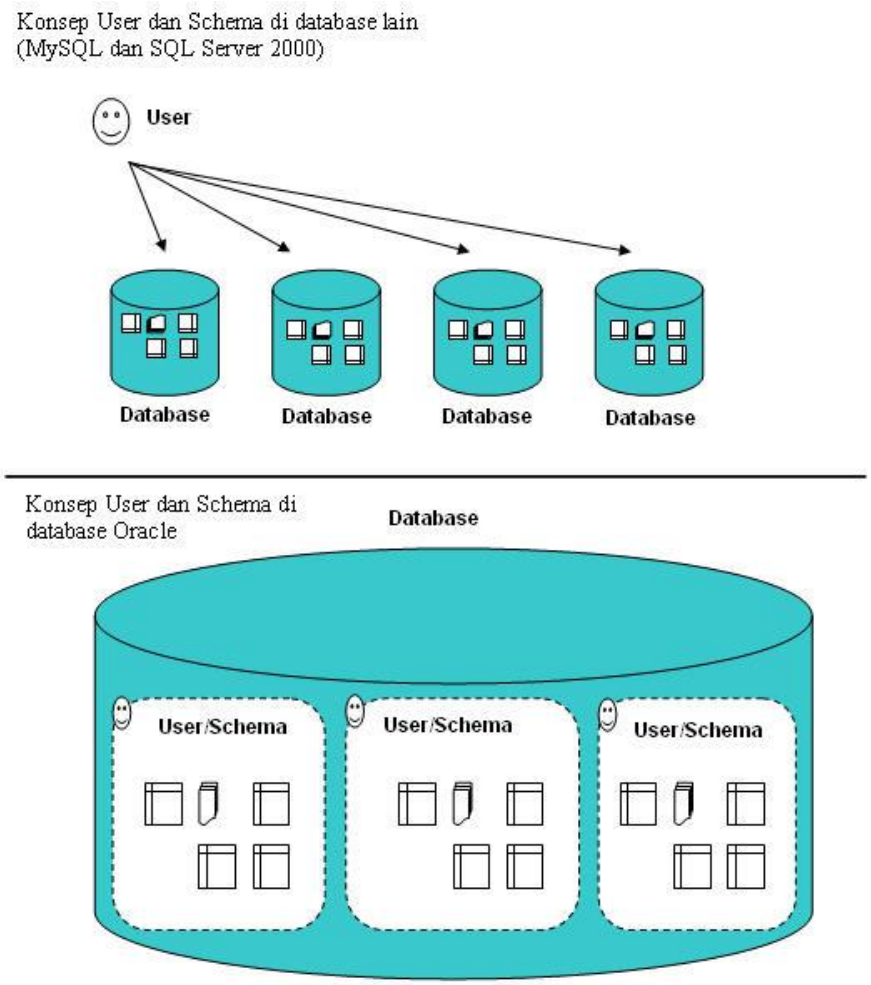
- Perintah untuk pendefinisian/pembuatan objek (Data Definition Language / DDL)
  - CREATE
  - ALTER
  - RENAME
  - DROP
  - TRUNCATE

- Perintah untuk menampilkan data (Data Retrieval)
  - SELECT
- Perintah untuk memanipulasi data (Data Manipulation Language /DML)
  - INSERT
  - UPDATE
  - DELETE
  - MERGE
- Perintah untuk mengontrol transaksi (Transaction Control Language /TCL)
  - COMMIT
  - ROLLBACK
  - SAVEPOINT
- Perintah untuk mengatur wewenang atau privilege (Data Control Language /DCL)
  - GRANT
  - REVOKE

2. User dan Schema Database

Konsep user

Setiap orang yang akan mengakses ke suatu database Oracle harus memiliki database user account atau biasa dikenal dengan user name. Pada database Oracle, user ada didalam database artinya user merupakan bagian dari suatu database. Berbeda dengan konsep user yang ada di database lain seperti MySQL atau SQL Server 2000, pada kedua database tersebut user ada diluar database. Ilustrasi perbedaan konsep user antara database Oracle dengan database MySQL atau SQL Server 2000 bisa dilihat pada gambar berikut:



Jadi agar user bisa mengakses ke database database dibuat dulu user account-nya. User yang berhak membuat user account adalah user SYS atau user SYSTEM. User SYS dan SYSTEM adalah user yang sudah ada di dalam database dan mempunyai wewenang untuk melakukan administrasi database. User SYS dan SYSTEM dibuat bersamaan dengan proses membuat database. Jadi begitu anda selesai membuat database atomatis sudah ada user SYS dan SYSTEM.

Perhatikan lagi gambar diatas, dalam satu database akan terdapat banyak user dan setiap user akan mempunyai banyak objek seperti tabel, indek, trigger, procedure dan function. Pengelompokan objek-objek secara logik di dalam database berdasarkan user pemiliknya disebut **schema**. Jadi setiap user pasti punya schema, schema atomatis terbentuk ketika user dibuat. Satu user hanya dihubungkan ke satu schema dan nama user sama dengan nama schema. Berarti user dan schema adalah hal yang sama.

Membuat User

Ingat, tadi sudah disebutkan bahwa yang berhak untuk membuat user adalah user SYS atau SYSTEM atau user yang sudah diberi privilege untuk CREATE USER (tentang privilige akan dibahas tersendiri). Lakukan koneksi ke database sebagai user SYSTEM, ingat password user SYSTEM pada saat create database.

1. Gunakan tools SQL*Plus, dari command prompt ketik **sqlplus**

```
C:\>sqlplus

SQL*Plus: Release 10.2.0.3.0 - Production on Fri Apr 6 11:05:09 2018
Copyright (c) 1982, 2006, Oracle.  All Rights Reserved.

Enter user-name: system/systempass

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL>
```

2. Buat user BUDI dengan password 'budipass'

```
SQL> CREATE USER budi IDENTIFIED BY budipass
  2  DEFAULT TABLESPACE users
  3  QUOTA unlimited ON users;
```

3. Beri ijin user budi untuk login ke database, membuat tabel dan membuat index

```
SQL> GRANT create session,
  2        create table,
  3        create indextype
  4  TO budi;
```

4. Login ke database sebagai user BUDI

```
SQL> CONNECT budi/budipass
Connected.
```

Untuk memastikan saat ini Anda login ke database sebagai user siapa atau di schema mana, ketik SHOW USER

```
SQL> SHOW USER
USER is "BUDI"
```

3. Membuat Tabel dan Mendefinisikan Constraint

Prasyarat :

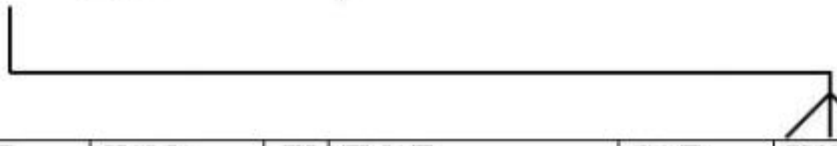
- Instalasi Software Oracle database sudah berhasil
- Anda sudah membuat database Oracle
- Sebaiknya anda baca Materi tentang User dan Schema

Membuat Tabel

Untuk belajar membuat tabel perhatikan struktur tabel di bawah ini :

Tabel : **BAGIAN**

KODE	NAMA_BAGIAN
10	Administrasi
20	Marketing
30	Support



NIP	NAMA	JK	EMAIL	GAJI	KODE_BAG
100	AHMAD	L	ahmad@dbicon.com	4.000.000	20
101	ERTIN	P	ertin@dbicon.com	4.250.000	10
102	EDWIN	L	edwin@dbicon.com	5.000.000	20
103	BUDI	L	budi@dbicon.com	5.500.000	30

Tabel : **PEGAWAI**

Login sebagai user BUDI. (Catatatan: user BUDI sudah dibuat pada bahasan Materi tentang User dan Schema).

```
SQL> CONNECT budi/budipass
Connected.
```

Buat tabel PEGAWAI dan BAGIAN

```
SQL> CREATE TABLE PEGAWAI
2 (NIP NUMBER(4),
3 NAMA VARCHAR2(15),
4 JK CHAR(1),
5 EMAIL VARCHAR2(20),
```

```
6 GAJI NUMBER(10),
7 KODE_BAG NUMBER(2));

SQL> CREATE TABLE BAGIAN
2 (KODE NUMBER(2),
3 NAMA_BAGIAN VARCHAR2(20));
```

Mendefinisikan Constraint

Constraint adalah batasan atau ketentuan yang diterapkan di tabel untuk menjaga konsistensi dan integritas data. Ada 5 jenis constraint di Oracle, yaitu :

- Primary Key
- Unique
- Not Null
- Check
- Foreign Key

Penggunaan dan cara mendefinisikan constraint adalah sebagai berikut:

- **Primary key**
  Defenisikan kolom NIP pada tabel PEGAWAI sebagai primary key.

```
SQL> ALTER TABLE PEGAWAI
2 ADD CONSTRAINT PK_PEGAWAI PRIMARY KEY (nip);
```

  Defenisikan kolom KODE pada tabel BAGIAN sebagai primary key.

```
SQL> ALTER TABLE BAGIAN
2 ADD CONSTRAINT PK_BAGIAN PRIMARY KEY (kode);
```

- **Not Null**
  Definisikan kolom NAMA pada tabel PEGAWAI harus selalu diisi (Not Null)

```
SQL> ALTER TABLE PEGAWAI
2 MODIFY nama NOT NULL;
```

- **Check**
  Definisikan kolom JK (jenis kelamin) pada tabel PEGAWAI hanya boleh diisi oleh ‘L’ dan ‘P’

```
SQL> ALTER TABLE PEGAWAI
2 ADD CONSTRAINT ck_jk CHECK (JK IN (’L’,’P’));
```

  Pastikan kolom GAJI pada tabel PEGAWAI minimal 1000000

```
SQL> ALTER TABLE PEGAWAI
2 ADD CONSTRAINT ck_gaji_1jt CHECK (GAJI >= 1000000);
```

- **Unique**
  Pastikan data untuk kolom EMAIL pada tabel PEGAWAI tidak boleh ada yang sama (Unique)

```
SQL> ALTER TABLE PEGAWAI
2 ADD CONSTRAINT UQ_EMAIL UNIQUE(email);
```

- **Foreign Key**
  Definisikan agar kolom KODE_BAG pada tabel PEGAWAI selalu merujuk ke kolom KODE pada tabel BAGIAN. (pendefinisian Foreign Key)

```
SQL> ALTER TABLE PEGAWAI
2 ADD CONSTRAINT fk_kode_bag FOREIGN KEY (kode_bag)
3* REFERENCES bagian(kode);
```

4. Input Data ke dalam tabel

Prasyarat :

- Anda sudah membaca dan menjalankan perintah-perintah yang ada di Materi tentang Membuat Tabel dan Mendefinisikan Constraint

■ Perintah INSERT

Untuk mengisi data ke tabel digunakan perintah INSERT. Syntax perintah INSERT adalah sebagai berikut :

INSERT INTO table [(column [, column...])]
VALUES (value [, value...]);

Sebelum melakukan pengisian data dengan perintah INSERT, kita harus lihat dulu strukutr tabel yang akan kita isi, karena urutan kolom dalam perintah INSERT sangat penting.

Lakukan koneksi ke database dengan user BUDI

```
SQL> connect budi/budipass
Connected.
```

Atau kalau dari command prompt

```
C:\>sqlplus budi/budipass

SQL*Plus: Release 10.2.0.3.0 - Production on Fri Apr 6 12:43:23 2018

Copyright (c) 1982, 2006, Oracle.  All Rights Reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options
SQL>
```

Lihat, tabel apa saja yang dimiliki oleh user BUDI

```
SQL> SELECT table_name
2 FROM user_tables;

TABLE_NAME
-----------
PEGAWAI
BAGIAN
```

Lihat struktur tabel BAGIAN dan PEGAWAI

SQL> DESC pegawai;

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
NIP                                       NOT NULL NUMBER(4)
NAMA                                      NOT NULL VARCHAR2(15)
JK                                                 CHAR(1)
EMAIL                                              VARCHAR2(20)
GAJI                                               NUMBER(10)
KODE_BAG                                           NUMBER(2)
```

SQL> DESC bagian;

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
KODE                                      NOT NULL NUMBER(2)
NAMA_BAGIAN                                        VARCHAR2(20)
```

- **Input data dengan urutan kolom sesuai dengan struktur tabel**

Isi tabel BAGIAN dengan data kode : 10, nama_bagian : ADMINISTRATION

```
SQL> INSERT INTO bagian(kode,nama_bagian)
VALUES(10,'ADMINISTRASI');
1 row created.
```

Sebenarnya jika kita akan memasukkan data dan kita sudah mengetahui struktur tabelnya dan data tersebut akan diinput urut sesuai nama kolom, maka nama kolom tidak perlu disebutkan. Dengan demikian maka penulisan perintah INSERT menjadi lebih sederhana, seperti dibawah ini :

```
SQL> INSERT INTO bagian VALUES(20,'MARKETING');
1 row created.
```

- **Input data untuk kolom tertentu saja**

Nama kolom perlu disebutkan jika kita akan mengisi nilai hanya ke beberapa kolom saja. Misalkan isi data PEGAWAI untuk kolom NIP: 100 NAMA: 'EDWIN', maka perintah INSERT ditulis sebagai berikut:

SQL> INSERT INTO pegawai (nip,nama) VALUES(100,'ANTON');
1 row created.

Untuk kolom lain pada baris tersebut akan diisi dengan NULL(kosong).

Jika kita lihat, dengan menggunakan perintah INSERT untuk menginput satu baris data diperlukan satu perintah INSERT. Sebenarnya masih ada beberapa utility lain di Oracle

yang bisa digunakan untuk memasukkan data ke tabel seperti External Table, SQL*Loader dan Data Pump. Utility itu nanti akan dibahas pada bagian tersendiri.

Untuk latihan dan nanti akan digunakan untuk materi selanjutnya, isi tabel BAGIAN dan PEGAWAI dengan data berikut :

```
- isi tabel BAGIAN
insert into bagian values(10,'Administrasi');
insert into bagian values(11,'Penjualan');
insert into bagian values(12,'Gudang');

- isi data pegawai
insert into pegawai values(1000,'WIRA', 'L',null,7000000,10);
insert into pegawai values(1100,'BUDI', 'L',null,5000000,12);
insert into pegawai values(1200,'ERTIN', 'P',null,4250000,10);
insert into pegawai values(1300,'NOVI', 'P',null,5500000,11);
insert into pegawai values(1201,'AHMAD', 'L',null,2575000,10);
insert into pegawai values(1202,'ESTI', 'P',null,3000000,10);
insert into pegawai values(1305,'EDWIN', 'L',null,3250000,11);
insert into pegawai values(1306,'AMRA', 'L',null,3100000,11);
insert into pegawai values(1101,'TONO', 'L',null,2250000,12);
insert into pegawai values(1102,'SUTEJO','L',null,2750000,12);
insert into pegawai values(1301,'DIANA', 'P',null,3000000,11);
insert into pegawai values(1302,'YULI', 'P',null,2750000,11);
insert into pegawai values(1303,'RINA', 'P',null,3100000,11);
insert into pegawai values(1304,'DHILA', 'P',null,2500000,11);
insert into pegawai values(1103,'ZAENAL','L',null,1800000,12);
commit;
```