**Kartika Nirwana Silalahi**

**41519120006**

**Pemrograman PL/SQL**

**Paper Studi Kasus Database**

**BAB I PENDAHULUAN**

**1.1 Basis Data (Database)**

Basis data (Database) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (query) basis data disebut sistem manajemen basis data (Database management system, DBMS). Sistem basis data dipelajari dalam ilmu informasi.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan di antara objek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data: ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel.

Basis data (database) adalah Representasi kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan(redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Adapun ciri-ciri basis data diantaranya adalah sebagai berikut:

* Efisiensi meliputi kecepatan, ukuran, dan ketepatan
* Data dalam jumlah besar.
* Berbagi Pakai (dipakai bersama sama/Sharebility).
* Mengurangi bahkan menghilangkan terjadinya duplikasi dan inkonsisten data

Proses terciptanya suatu database terjadi dari beberapa tahap yaitu :

1. Menentukan kebutuhan data
2. Menjelaskan data
3. Memasukkan data

Setelah tahap penciptaan selesai dan database telah terbentuk maka database tersebut akan digunakan atau dipakai. Pengguna atau pemakai database dapat berupa orang atau program aplikasi. Orang biasanya menggunakan database dari terminal dan mengambil data dan informasi dengan menggunakan query language. Istilah query adalah permintaan informasi dari database, dan arti dari query language adalah bahasa khusus yang user friendly yang memungkinkan komputer menjawab query atau permintaan informasi atau data tersebut

Permintaan-permintaan tersebut akan dieksekusi atau dijalankan oleh database dengan proses sebagai berikut :

* Data Manipulation Language (DML) menentukan DBMS (database management system) data apa yang diperlukan.
* DBMS memeriksa skema dan subskema untuk menguji bahwa data ada dalam database.­­
* DBMS meneruskan permintaan data ke sistem operasi.
* DBMS mengambil data dan memasukkannya ke dalam area penyimpanan buffer khusus dalam penyimpanan primer.
* Data tersebut ditransfer ke dalam area input program aplikasi.
* DBMS mengembalikan pengendalian ke program aplikasi.
* Program aplikasi menggunakan data.

**1.2 Fungsi Basis Data**

Fungsi Database adalah untuk menyimpan data data mulai dari judul kolom hingga record atau Baris terakhir dalam daftar atau database.

* Database adalah koleksi “data operasional” yang tersimpan dan dipakai oleh
* sistem aplikasi dari suatu organisasi.
* Data input adalah data yang masuk dari luar sistem
* Data output adalah data yang dihasilkan sistem
* Data operasional adalah data yang tersimpan pada sistem.

**1.3 Tujuan Basis Data**

Tujuan utama dalam pengolahan data dalam sebuah basis data adalah agar kita dapat memperoleh data yang kita cari dengan mudah dan cepat . Pemanfaatan basis data dapat digunakan dengan tujuan:

* Pemanfaatan Database memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan ( manipulasi ) dan menampilkan kembali data tersebut dengan cepat dan mudah, dari pada kita menyimpan data secara manual.
* Dengan Database penggunaan ruang penyimpanan data dapat dilakukan karena kita dapat melakukan penekanan jumlah pengulangan data dengan menerapkan sejumlah pengkodean
* Pemanfaatan pengkodean atau pembentukan relasi antar data dengan penerapan aturan atau batasan tipe data dapat diterapkan dalam Database yang berguna untuk menentukan ketidakakuratan pemasukan atau penyimpanan.
* Dalam sejumlah sistem ( aplikasi ) pengolah database tidak menerapkan aspek keamanan dalam penggunaan database. Tetapi untuk sistem yang besar dan serius, aspek keamanan juga dapat diterapkan. Dengan begitu kita dapat menentukan siapa yang boleh menggunakan database dan menentukan jenis operasi-operasi apa saja yang boleh dilakukan.
* Apabila ada perubahan data pada aplikasi yang berbeda maka secara otomatis perubahan itu berlaku untuk keseluruhan (keselarasan).
* Data dapat dipakai secara bersama-sama oleh beberapa program aplikasi

**1.4 PL SQL**

SQL merupakan singkatan dari Structured Query Language. SQL atau juga sering disebut sebagai query merupakan suatu bahasa (language) yang digunakan untuk mengakses database. SQL dikenalkan pertama kali dalam IBM pada tahun 1970 dan sebuah standar ISO dan ANSII ditetapkan untuk SQL. Standar ini tidak tergantung pada mesin yang digunakan (IBM, Microsoft atau Oracle). Hampir semua software database mengenal atau mengerti SQL. Jadi, perintah SQL pada semua software database hampir sama. Terdapat 3 jenis perintah SQL, yaitu DDL, DML dan DCL. Beberapa kelebihan PL/SQL dalam database Oracle .

PL/SQL dapat digunakan di server sehingga client hanya dapat mengakses didalam server:

* Penggunaan PL/SQL mudah dimengerti oleh setiap pengguna.
* PL/SQL dapat didesain khusus untuk database Oracle dalam menggunakan
* program aplikasi

BAB II PEMBAHASAN

**(STUDI KASUS)**

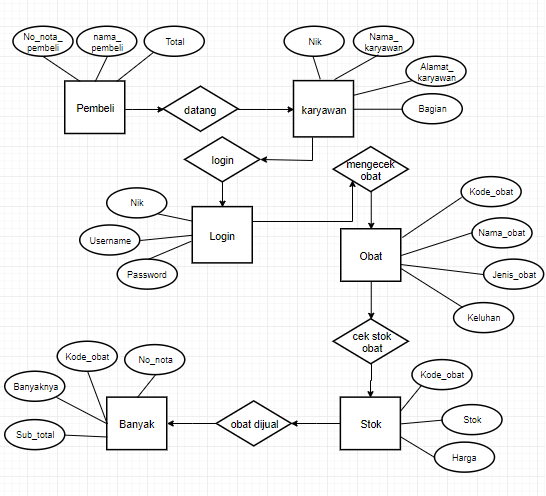
2.1 Studi Kasus

*Apotek Seger Waras ingin merubah model pembukuannya dengan komputerisasi maka diperlukan sebuah program yang dapat menampung data penjualan tersebut. Berikut ini adalah alur proses transaksi dalam apotek tersebut.*

Transaksi pembelian:

*Pembeli datang ke apotek dilayani oleh karyawan, karyawan mengecek stok obat dan apabila terdapat obat yang diinginkan maka stok obat berkurang dan karyawan mencatat no kwitansi, pembeli memperoleh obat dan kwitansi.*

2.2 ERD Apotek



ERD adalah merupakan salah satu model yang digunakan untuk mendesain database dengan tujuan menggambarkan data yang berelasi pada sebuah database. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga komponen yang digunakan, yaitu :

**a. Entitas.** Entitas/entiti merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entity ini biasanya digambarkan dengan persegi panjang.

**b. Atribut.** Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasikan isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol elips.

2.3 Rancangan Tabel

Berikut rancangan table yang akan dibuat pada database apotek.

Tabel Pembeli:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Field | Type | Size | Null | Index |
| 1 | No\_nota\_pembeli | Varchar | 8 | N | Primary Key |
| 2 | Nama\_pembeli | Varchar | 30 | Y |  |
| 3 | Total | Number | 11 |  |  |

Tabel Obat:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Field | Type | Size | Null | Index |
| 1 | Kode\_obat | Varchar | 8 | N | Primary Key |
| 2 | Nama\_obat | Varchar | 30 | N |  |
| 3 | Jenis\_obat | Varchar | 8 | N |  |
| 4 | Keluhan | Varchar | 50 | N |  |

Tabel Karyawan:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Field | Type | Size | Null | Index |
| 1 | Nik | Varchar | 8 | N | Primary Key |
| 2 | Nama\_karyawan | Varchar | 30 | N |  |
| 3 | Alamat\_karyawan | Varchar | 50 | N |  |
| 4 | Bagian | Varchar | 10 | Y |  |

Tabel Banyak:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Field | Type | Size | Null | Index |
| 1 | No\_nota | Varchar | 8 | N | Foreign Key |
| 2 | Kode\_obat | Varchar | 8 | N | Foreign Key |
| 3 | Banyaknya | Number | 9 | N |  |
| 4 | Sub\_total | Number | 9 |  |  |

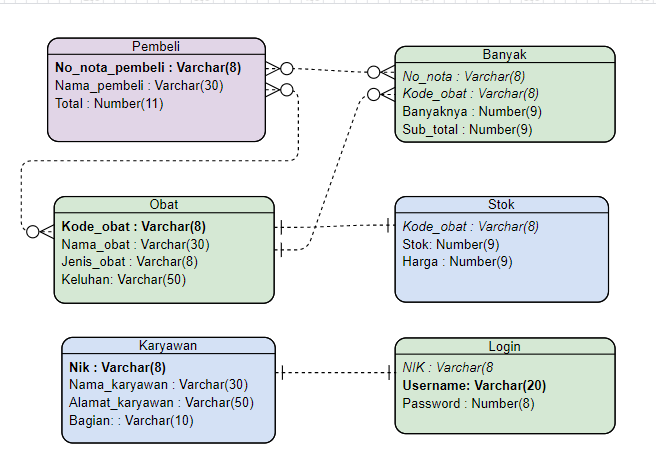
Tabel Login:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Field | Type | Size | Null | Index |
| 1 | Nik | Varchar | 8 | N | Foreign Key |
| 2 | Username | Varchar | 20 | N | Primary Key |
| 3 | Password | Number | 8 | N |  |

Tabel Stok:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Field | Type | Size | Null | Index |
| 1 | Kode\_obat | Varchar | 8 | N | Foreign Key |
| 2 | Stok | Number | 9 | N |  |
| 3 | Harga | Number | 9 | N |  |

2.4 Relasi Tabel



Relasiadalah hubungan antara tabel yang mempresentasikan hubungan antar objek di dunia nyata. Relasi merupakan hubungan yang terjadi pada suatu tabel dengan lainnya yang mempresentasikan hubungan antar objek di dunia nyata dan berfungsi untuk mengatur mengatur operasi suatu database. Berikut ini adalah 3 jenis relasi tabel database diantaranya, yaitu:

**One to one**

Relasi One to One (1 to 1) terjadi apabila sebuah data yang ada pada tabel A juga ada pada tabel B, yang mana data tersebut hanya diperbolehkan satu buah saja pada masing-masing tabel, tidak boleh terdapat beberapa data yang sama pada tabel yang sama.

**One to Many**

Relasi One to Many (1 to n) terjadi apabila sebuah data yang ada pada tabel A memiliki beberapa data yang sama pada tabel B. Pada relasi ini hanya diperbolehkan sebuah data pada tabel A dan tabel B boleh memiliki beberapa data yang sama pada tabel A.

**Many to Many**

Relasi Many to Many (n to m) terjadi apabila sebuah data yang ada pada tabel A memiliki beberapa data yang sama pada tabel B dan sebuah data pada tabel B juga memiliki beberapa data yang sama pada tabel A. Pada relasi ini tabel A dan tabel B diperbolehkan memiliki beberapa data yang sama dengan kedua tabel tersebut.

2.5 DDL (Data Definition Language)

Dalam database SQL biasanya terdiri dari tabel data dan semua data di dalamnya. DDL ini menggambarkan desain basis data secara keseluruhan dan didefinisikan dengan bahasa khusus. Dengan bahasa tersebut kita bisa membuat tabel baru, membuat indeks, mengubah tabel, mengubah struktur tabel, atau menghapus dabel. Hasil dari kompilasi DDL biasanya merupakan kumpulan dari beberapa tabel yang disimpan dalam file khusus yang disebut Kamus Data.

Yang termasuk dalam bahasa pada DDL yaitu:

1. Create
2. Drop
3. Alter

**2.5.1 CREATE**

Perintah CREATE berfungsi untuk membuat. Dalam PL/SQL perintah CREATE ini dapat membuat tabel baru, indeks, prosedur, fungsi, view, dll. Dalam pembahasan ini akan dibahas mengenai fungsi perintah CREATE untuk membuat tabel. Adapun penulisan perintah CREATE dituliskan sebagai berikut:

**CREATE TABLE nama\_tabel**

**(**

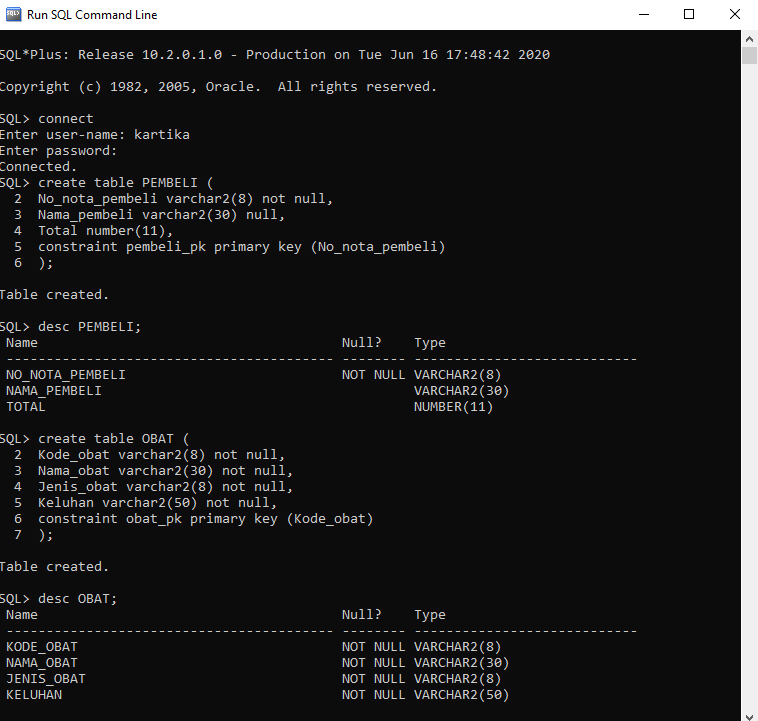
**Nama\_kolom1 tipe\_datanya(jumlah\_data) [ NULL | NOT NULL ],**

**Nama\_kolom2 tipe\_datanya(jumlah\_data) [ NULL | NOT NULL ]**

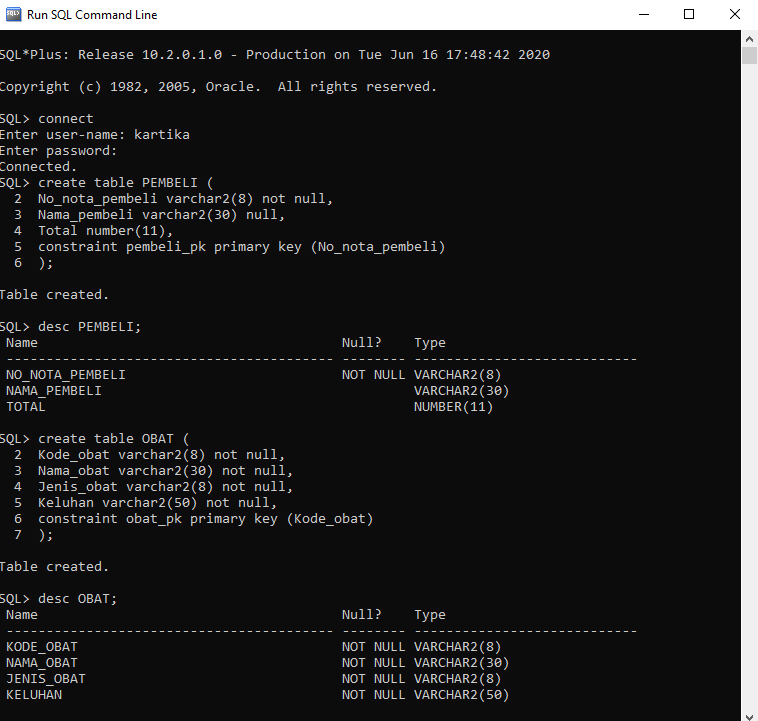
**...**

**);**

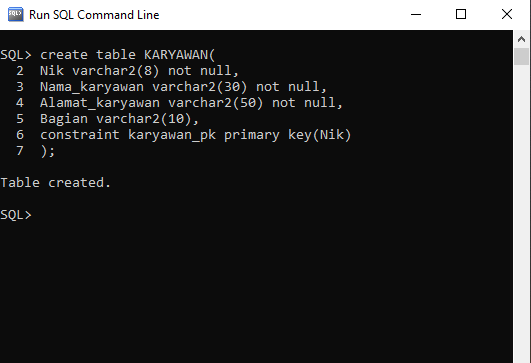
Create Table Pembeli



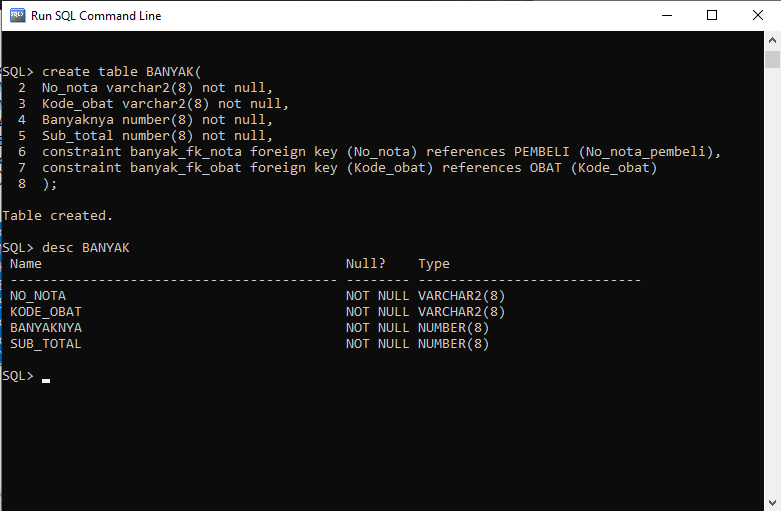
Create Table Obat:



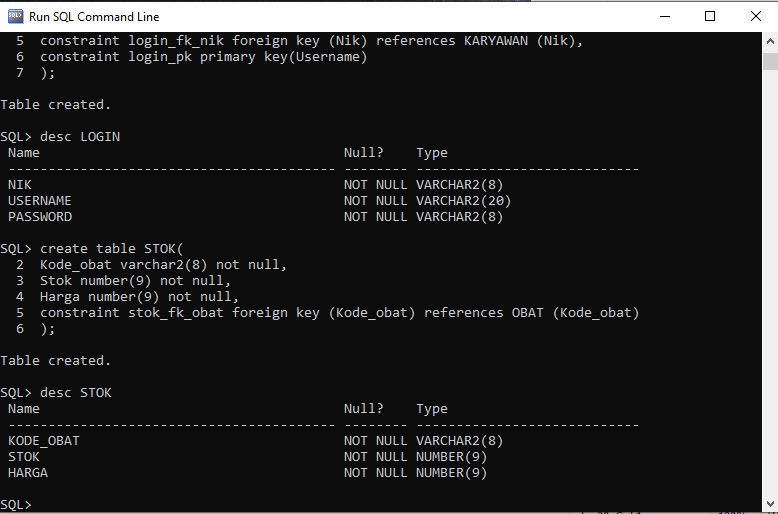
Create Table Karyawan:



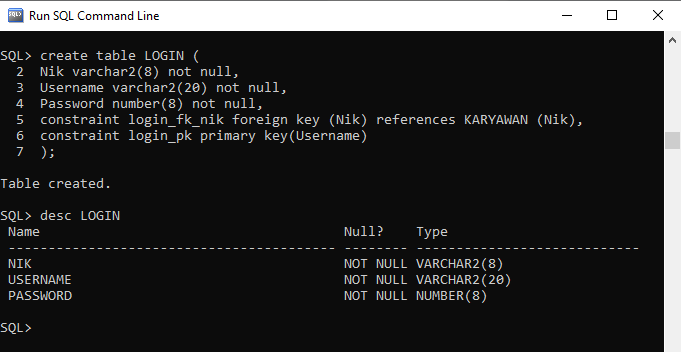
Create Table Banyak:



Create Table Stok:



Create Table Login:



**2.5.2 DROP**

Perintah DROP ini berfungsi untuk menghapus, bisa untuk menghapus tabel, kolom, view, fungsi, prosedur, dll. Dalam pembahasan ini dibahas mengenai fungsi DROP untuk menghapus tabel atau kolom. Adapun penulisan dari perintah DROP ini dituliskan sebagai berikut

**DROP TABLE nama\_tabel;**

Sebagai contoh jika ingin menghapus table PEMBELI, maka sebagai berikut:

DROP TABLE PEMBELI;

**2.5.3 ALTER**

Perintah ALTER digunakan untuk merubah tabel, dalam hal ini adalah merubah variabel yang digunakan dalam tabel seperti merubah tipe data, merubah jumlah data, dan juga bisa untuk menambah kolom baru. Berikut dituliskan perintah ALTER:

Untuk menambah kolom baru:

**ALTER TABLE nama\_tabel**

**ADD nama\_kolom definisi\_kolom;**

Sebagai contoh jika ingin menambah kolom No\_pembeli pada Tabel PEMBELI:

ALTER TABLE PEMBELI

ADD No\_pembeli VARCHAR2(12)

Perintah ALTER juga bisa digunakan untuk menambah beberapa kolom baru sekaligus dengan penulisan

**ALTER TABLE table\_name**

**ADD (nama\_kolom1 definisi\_kolom,**

**nama\_kolom2 definisi\_kolom,**

**...**

**nama\_kolom\_n definisi\_kolom);**

Perintah ALTER juga berfungsi untuk memodifikasi dari kolom seperti contoh untuk merubah definisi dari tabel. Misal pada tabel A, pada kolom nama didefinisikan dengan varchar2(7). Penulisan perintah ALTER dituliskan sebagai berikut:

**ALTER TABLE nama\_tabel**

**MODIFY nama\_kolom definisi;**

Sebagai contoh jika ingin merubah jumlah data dari kolom Nama\_obat pada Table OBAT:

ALTER TABLE OBAT

MODIFY Nama\_obat VARCHAR2(28)

Untuk memodifikasi juga bisa untuk beberapa kolom sekaligus dengan penulisan sebagai berikut:

**ALTER TABLE table\_name**

**MODIFY (nama\_kolom1 definisi,**

**nama\_kolom2 definisi,**

**...**

**nama\_kolom\_n definisi);**

Selain itu perintah ALTER dan DROP digunakan untuk menghapus kolom pada tabel dengan penulisan sebagai berikut:

**ALTER TABLE nama\_tabel**

**DROP COLUMN nama\_kolom;**

Sebagai contoh jika ingin mengahpus kolom Nama\_obat pada Table OBAT:

ALTER TABLE OBAT

DROP Nama\_obat

2.6 DML (Data Manipulation Language)

Jika pada DDL adalah bahasa untuk mendefinisikan awal mengenai basis data (tabel, view, dll) maka DML merupakan bahasa yang digunakan untuk memanipulasi data dari basis data (dalam hal ini berupa tabel). Manipulasi data dapat berupa:

* Penyisipan / penambahan data baru ke dalam tabel
* Penghapusan data dari tabel
* Pengubahan data dari tabel

Adapun perintah yang termasuk dalam DML ini yaitu :

1. INSERT
2. UPDATE
3. SELECT
4. DELETE.

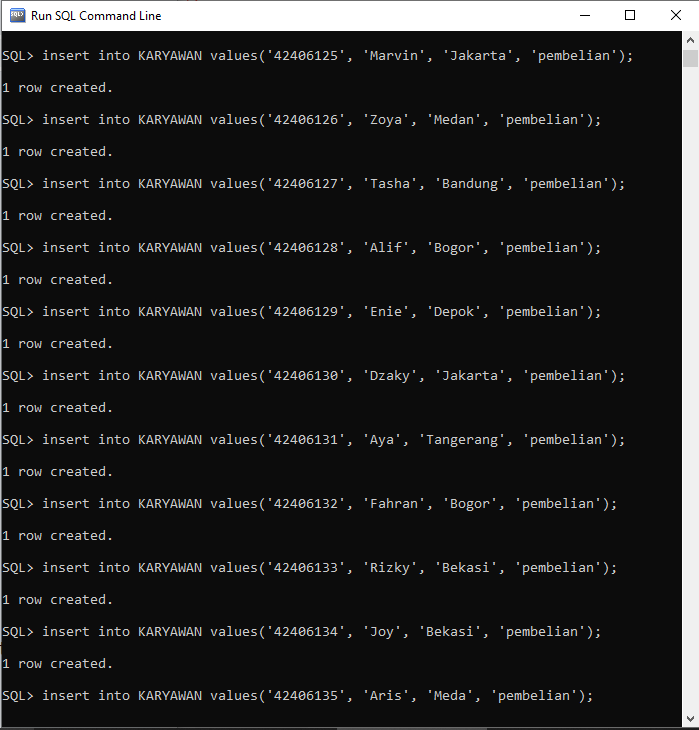
**2.6.1 INSERT**

Perintah INSERT digunakan untuk menyisipkan / menambah data baru ke dalam tabel. Adapun perintah INSERT dituliskan seperti berikut:

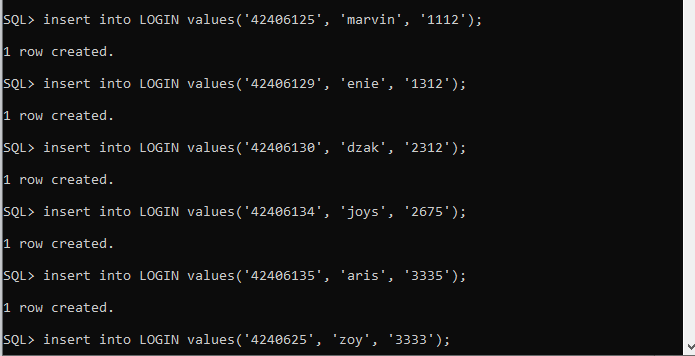
**INSERT INTO nama\_table (kolom1, kolom2,...)**

**VALUES(nilai1, nilai2,...);**

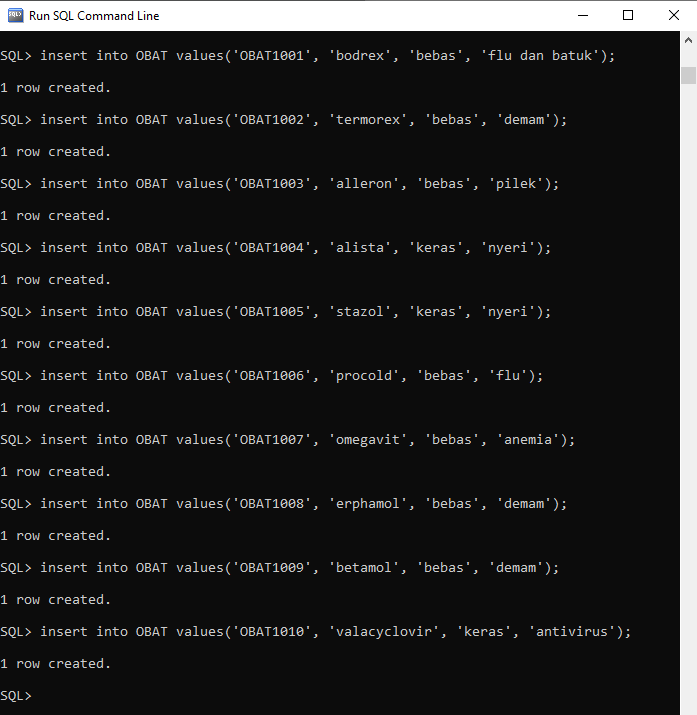
Insert Table Karyawan:



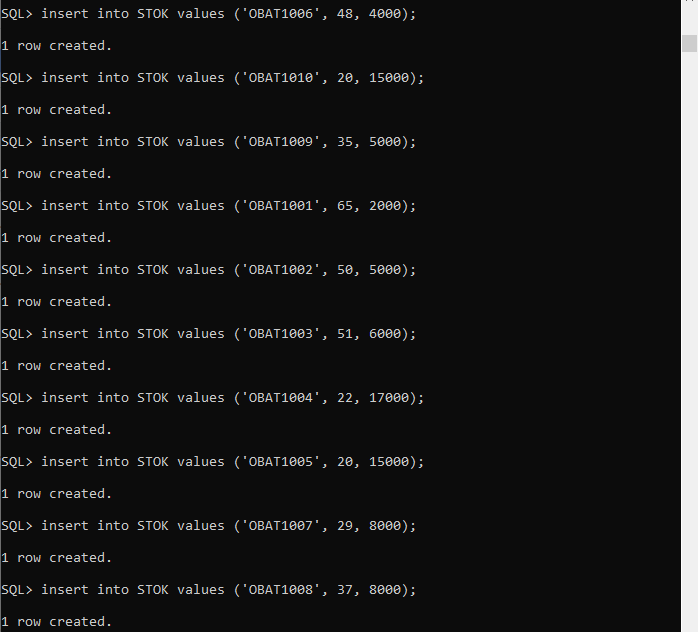
Insert Login:



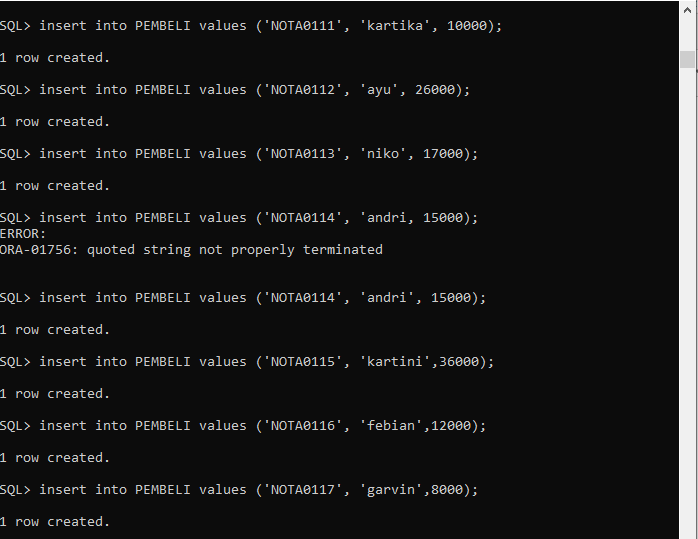
Insert Obat:



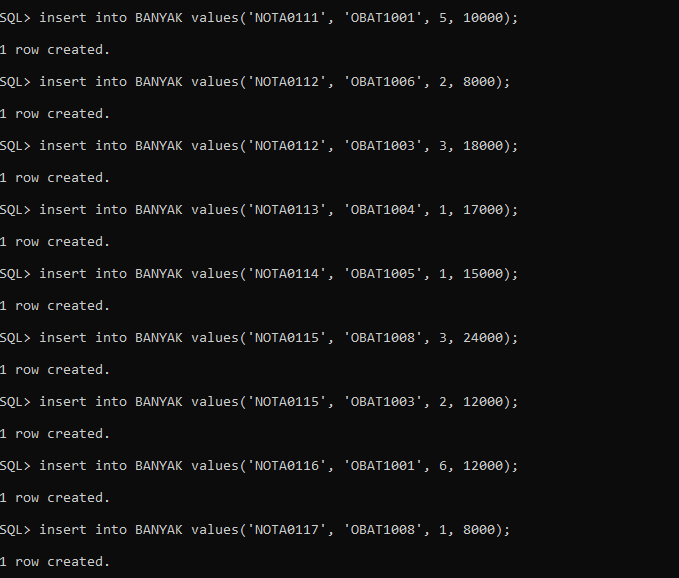
Insert Stok:



Insert Pembeli:



Insert Banyak:



**2.6.2 UPDATE**

Perintah UPDATE digunakan untuk merubah isi dari kolom (baris) yang diinginkan. Adapun penulisan perintah UPDATE dituliskan sebagai berikut:

**UPDATE nama\_tabel**

**SET kolom1 = nilai1,**

**Kolom2 = nilai2,**

**...**

**WHERE kondisi;**

Jika ingin mengupdate semua data maka, perintah WHERE tidak perlu digunakan. Perintah WHERE digunakan jika kita ingin meng-UPDATE data hanya pada kolom-kolom tertentu dengan kondisi tertentu sebagai filter nya.

**2.6.3 SELECT**

Perintah SELECT pada PL/SQL mempunyai fungsi untuk mengambil (retrieve) informasi dari database. Perintah ini juga mempunyai kemampuan seperti berikut:

1. Projection : Memilih kolom kolom dalam suatu tabel yang dihasilkan dari suatu query. Memilih beberapa atau banyak kolom yang dibutuhkan.
2. Selection : Memilih baris baris dalam suatu tabel yang dihasilkan dari suatu query. Banyak kriteria yang dapat digunakan untuk membatasi baris baris yang akan diambil.
3. Joining : Mengambil data bersama yang disimpan dalam tabel tabel yang berbeda dengan menentukan hubungan antara mereka.

Untuk mengambil data dari keseluruhan isi tabel, maka perintah SELECT ditulis menjadi

**SELECT \*FROM nama\_tabel;**

Telah dicoba sebelumnya menampilkan isi dari tabel nilai dengan menggunakan perintah SELECT dimana akan tampil semua isi dari tabel nilai. Perintah ini juga bisa menampilkan hanya salah satu atau beberapa kolom yang akan ditampilkan dengan penulisan :

**SELECT kolom1, kolom2,…,kolom n FROM nama\_tabel;**

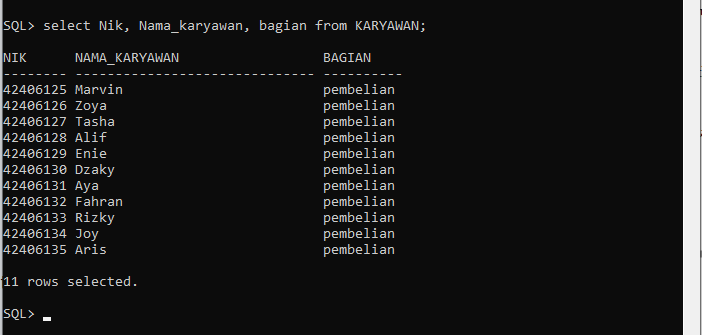
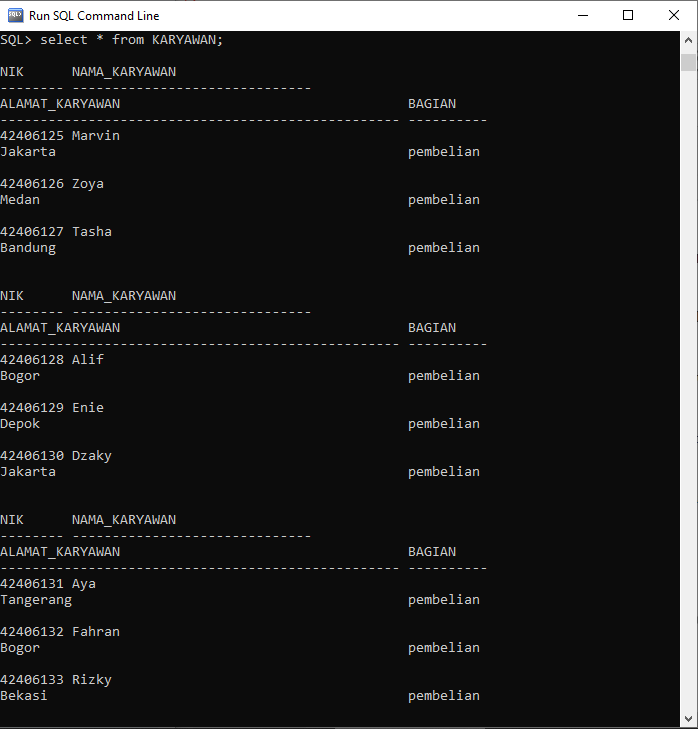
**WHERE kondisi;**

Selain itu juga perintah ini bisa memfilter data yang akan ditampilkan dengan menggunakan perintah WHERE. Perintah WHERE akan memfilter data yang disesuaikan dengan kondisi yang didefinisikan pada perintah SELECT. SELECT juga mempunyai fungsi lain yaitu dapat memasukkan nilai dari suatu variabel ke variabel lain dengan menggunakan perintah:

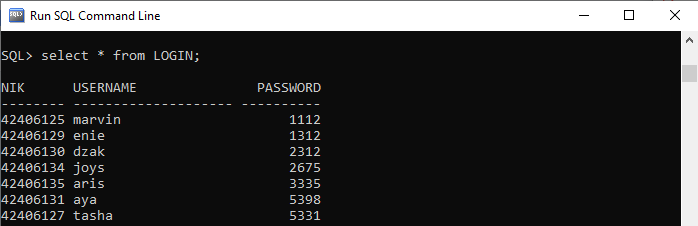
SELECT variabel\_awal INTO variabel\_tujuan from nama\_tabel

WHERE kondisi;

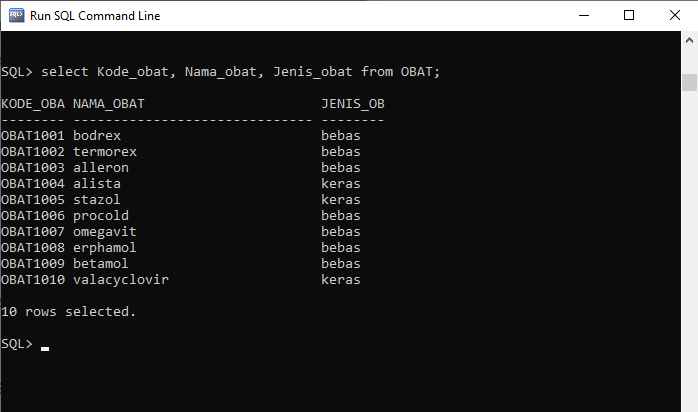
Select Table Karyawan:

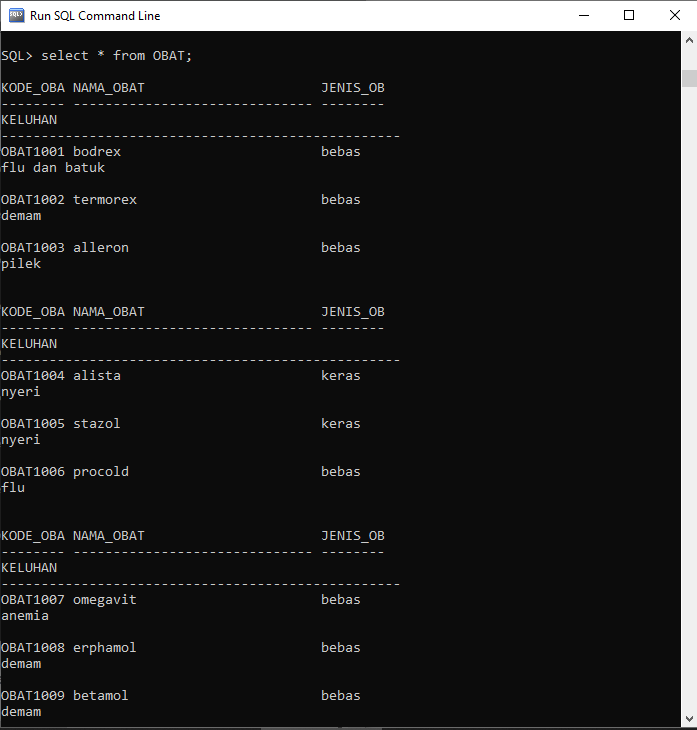


Select Tabel Login:

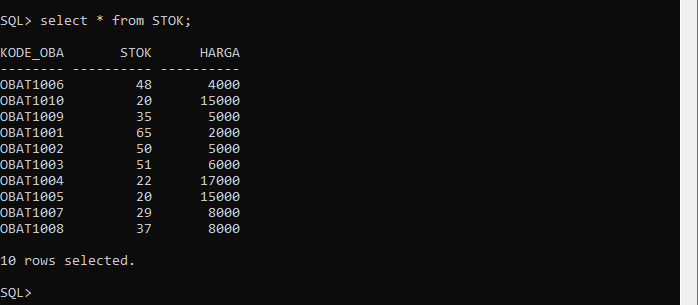


Select Table Obat:

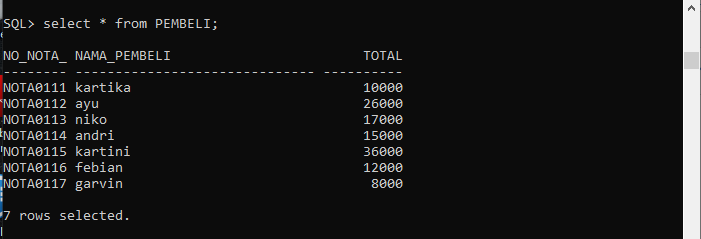




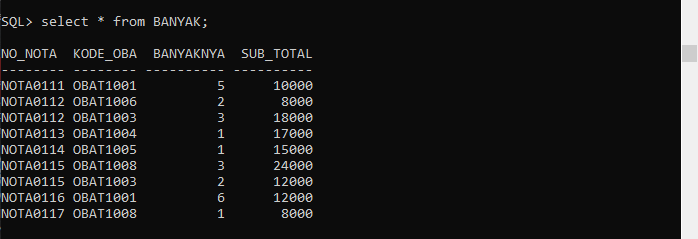
Select Table Stok:



Select Table Pembeli:



Select Table Banyak:



**2.6.4 DELETE**

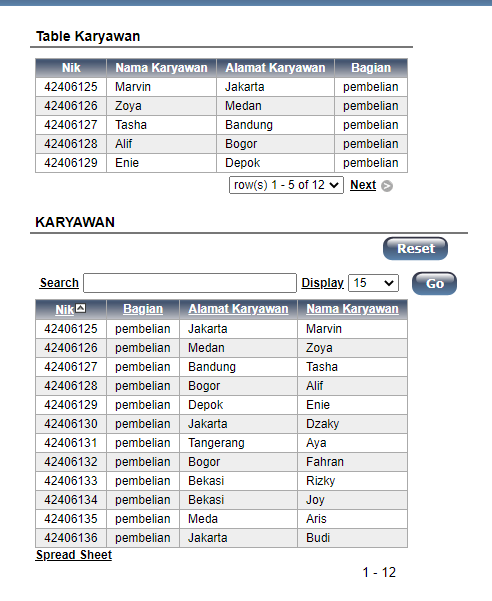
Perintah DELETE berguna untuk menghapus baris dari suatu tabel. Bisa diikuti dengan kondisi yang bersesuaian dengan data baris yang ingin dihapus. Penulisan perintah DELETE ditunjukkan sebagai berikut:

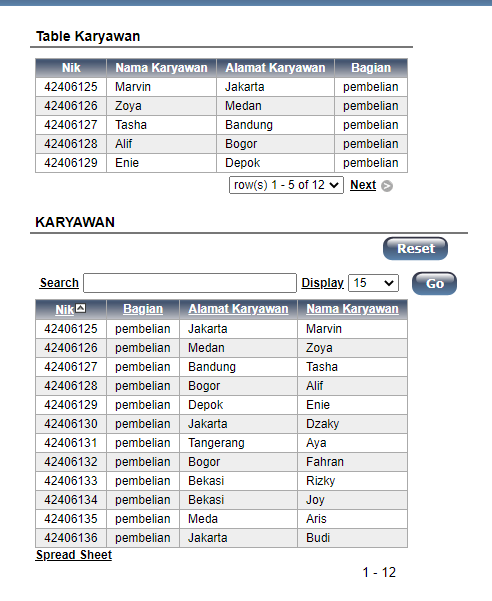
DELETE FROM nama\_tabel

WHERE conditions;

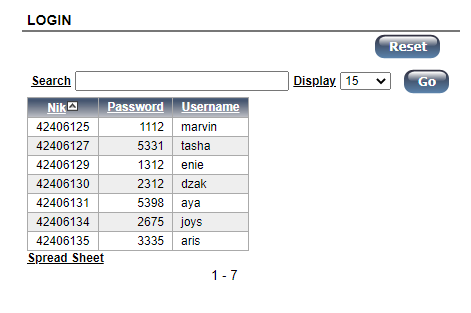
2.7 REPORT

**Report Karyawan:**

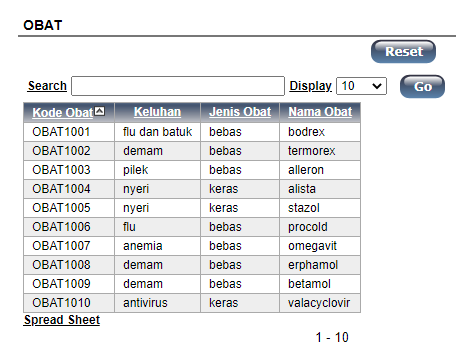
****

****

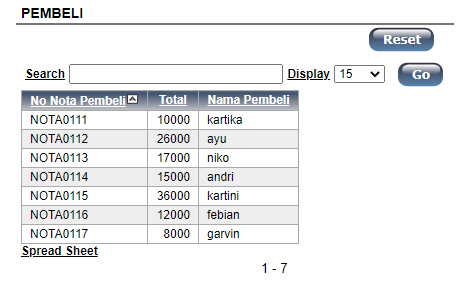
**Report Login:**

****

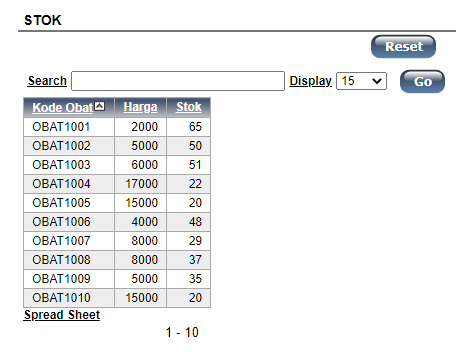
**Report Obat:**

****

**Report Pembeli:**

****

**Report Stok:**

****

**BAB III PENUTUP**

Sebuah database bisa dimanfaatkan sekaligus secara bersama oleh banyak pengguna (multi user).

1. Melakukan perlindungan dan pengamanan data. Media penyimpanan dapat berupa data di Microsoft excel, Microsoft Access dan Oracle Database
2. Setiap data hanya bisa diakses atau dimanipulasi oleh pihak yang diberi otoritas dengan memberikan login dan password terhadap masing-masing data.Agar pemakai mampu menyusun suatu pandangan (view) abstraksi dari data. Hal ini bertujuan menyederhanakan interaksi antara pengguna dengan sistemnya dan database dapat mempresentasikan pandangan yang berbeda kepada para pengguna, programmer dan administrator-nya.