**PENGGUNAAN BASIS DATA PADA SISTEM PENJUALAN DAN PENYEDIAAN OBAT DI APOTEK**

**D-IV TEKNIK INFORMATIKA**

Disusun untuk memenuhi tugas mata kuliah Sistem Basis Data

Oleh:

Alvira Putrina Darajat 181524002

Zara Veda Mahendra 181524031



**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA**

**POLITEKNIK NEGERI BANDUNG**

**2019**

1. **Topik**

Transaksi Penjualan dan Penyediaan Obat di Apotek.

1. **Tujuan**

Merancang suatu sistem basis data untuk mengolah data penjualan dan penyediaan obat di apotek serta membuat faktur yang sesuai.

1. **Manfaat**

Mampu menyediakan proses otomatisasi dalam penjualan dan penyediaan obat di Apotek.

.

1. **Deskripsi**

Apotek merupakan salah satu jenis usaha di bidang perobatan. Apotek menyediakan obat-obatan baik obat generik berwarna hijau, biru, merah, obat keras dengan dosis tinggi, psikotropika, maupun obat dengan resep dokter. Selain itu, apotek berbeda dengan toko obat, karena yang memimpin sebuah apotek ialah seorang apoteker.

Fungsi dari apotek adalah untuk pemerataan distribusi obat dan salah satu tempat pelayanan informasi, apotek merupakan sarana pelayanan kesehatan yang berkewajiban untuk menyediakan dan menyalurkan obat dan perbekalan farmasi lainnya yang dibutuhkan oleh masyarakat.

Pada sistem basis data ini, perancangan dibuat untuk proses penyediaan obat dan penjualan obat. Prosedur penyediaan dan penjualan obat adalah sebagai berikut :

* 1. Prosedur Penyediaan Obat
     1. Pemesanan obat
        1. Terdapat daftar obat yang telah mencapai stok minimum, telah mencapai *expired date*, atau daftar obat baru.
        2. Daftar obat yang diberikan berisi nama obat, merk obat, supplier obat, jumlah stok obat yang tersisa, jumlah stok obat yang seharusnya disimpan, dan harga beli obat tersebut.
        3. Faktur pemesanan obat dibuat. Data pada faktur pemesanan obat terdiri dari nomor faktur, daftar obat yang dipesan beserta produsennya, harga setiap obat, dan harga total.
        4. Setelah obat diterima, data obat seperti expired date, production date, nama obat, id obat, id batch obat, unit (kapsul, sirup, kaplet, tablet), jenis (luar/dalam), kandungan obat, aturan pakai, merk, dan stok obat ditambahkan ke dalam sistem.
  2. Prosedur Penjualan Obat
     1. Non-resep
        1. Konsumen menyerahkan data obat non-resep kepada kasir.
        2. Kasir menerima data obat non-resep, Jika konsumen setuju, faktur jual dibuat.
        3. Data pada faktur jual terdiri dari nomor faktur, daftar nama obat, harga setiap obat, dan harga total.

1. **Batasan Sistem**

Sistem ini dirancang untuk kebutuhan pada proses penyediaan dan penjualan obat. Terdapat beberapa batasan pada rancangan sistem basis data ini, yaitu :

* Sistem mampu membantu proses penyediaan obat yang diproduksi dan dikemas oleh suatu produsen tertentu.
* Sistem mampu membantu proses penjualan obat yang diproduksi dan dikemas oleh suatu produsen tertentu.

Selain itu, dari batasan ini didapatkan data-data yang diperlukan. Beberapa data yang diperlukan pada rancangan sistem basis data ini adalah sebagai berikut :

* 1. Obat

Obat menjadi sebuah *entity* dengan atribut sebagai berikut :

|  |  |
| --- | --- |
| Atribut | Tipe Data |
| id\_obat | Char[8] |
| nama\_obat | Char[32] |
| aturan\_pakai | Text |
| apotek\_stok | Integer |
| batas\_min\_stok | Integer |
| harga\_jual | Integer |

* 1. *Supplier*

*Supplier* menjadi sebuah *entity* karena memiliki hubungan dengan apotek, yaitu sebagai penyedia obat. Berikut adalah atribut yang ada pada *entity* produsen obat :

|  |  |
| --- | --- |
| Atribut | Tipe Data |
| id\_supplier | Char[8] |
| nama\_supplier | Char[32] |
| alamat | Text |
| no\_telp\_supplier | String[12] |

* 1. Apotek’s Order

Apotek’s Order merupakan *entity* pengajuan pemesanan kepada *supplier* (produsen) obat. Atribut dari Apotek’s Order adalah sebagai berikut :

|  |  |
| --- | --- |
| Atribut | Tipe Data |
| no\_order\_apotek | Char[8] |
| tanggal\_pemesanan | Date |

* 1. Konsumen’s Order

Konsumen’s Order merupakan *entity* pembelian obat yang diajukan oleh konsumen kepada Apotek. Atribut Konsumen’s Order adalah sebagai berikut :

|  |  |
| --- | --- |
| Atribut | Tipe Data |
| no\_order\_konsumen | Char[8] |
| tanggal\_pembelian | Date |

* 1. Batch

*Entity* Batch merupakan *entity* yang menandakan batch dari suatu obat. Atributnya adalah sebagai berikut:

|  |  |
| --- | --- |
| Atribut | Tipe Data |
| id\_batch | Char[6] |
| exp\_date | Date |
| prod\_date | Date |

* 1. Satuan Obat

|  |  |
| --- | --- |
| Atribut | Tipe Data |
| satuan\_obat | Char[30] |

* 1. Kategori Obat

|  |  |
| --- | --- |
| Atribut | Tipe Data |
| nama\_kategori | Character[25] |

* 1. Komposisi

|  |  |
| --- | --- |
| Atribut | Tipe Data |
| nm\_zat | Character[40] |

* 1. Rak

|  |  |
| --- | --- |
| Atribut | Tipe Data |
| id\_rak | Character[2] |

* 1. Lokasi Obat

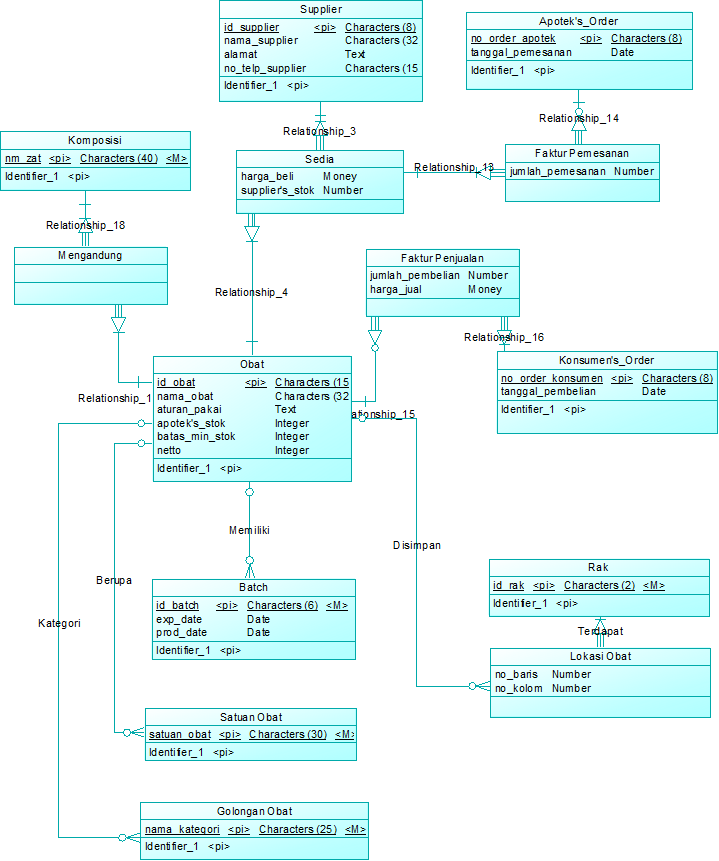
|  |  |
| --- | --- |
| Atribut | Tipe Data |
| no\_baris | Integer |
| no\_kolom | Integer |

1. **Kebutuhan Informasi**

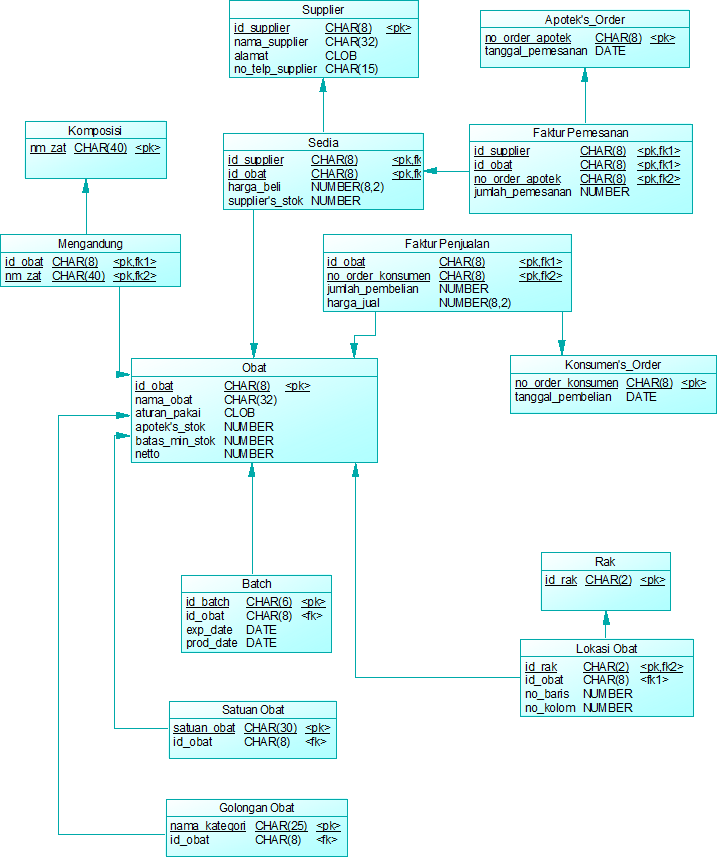
Adapun informasi-informasi yang ingin didapat dari sistem adalah sebagai berikut :

* 1. Laporan penjualan obat, terdiri dari :
     1. Tanggal
     2. Nama obat
     3. Stok
     4. Satuan
     5. Jumlah terjual
     6. Harga jual
  2. Laporan pembelian obat, terdiri dari :
     1. Tanggal pembelian
     2. Nama obat
     3. Supplier
     4. Jumlah pemesanan
     5. Harga
  3. Menentukan obat pengganti jika obat yang dicari habis. Obat dikatakan sebagai obat pengganti apabila kandungan obat tersebut mirip dengan obat yang dicari.
  4. Aplikasi membuat order secara otomatis apabila terdapat stok obat yang sudah mencapai batas minimum stok atau sudah mendekati expired date.

1. **Rancangan Diagram *Entity Relationship***

****

*Diagram Conceptual Data Model*



*Diagram Physical Data Model*

1. **Implementasi**
   1. **SQL Script**

|  |
| --- |
| alter table BATCH  drop constraint FK\_BATCH\_MEMILIKI\_\_OBAT;  alter table FAKTUR\_PEMESANAN  drop constraint FK\_FAKTUR\_P\_RELATIONS\_SEDIA;  alter table FAKTUR\_PEMESANAN  drop constraint FK\_FAKTUR\_P\_RELATIONS\_APOTEK\_S;  alter table FAKTUR\_PENJUALAN  drop constraint FK\_FAKTUR\_P\_RELATIONS\_OBAT;  alter table FAKTUR\_PENJUALAN  drop constraint FK\_FAKTUR\_P\_RELATIONS\_KONSUMEN;  alter table GOLONGAN\_OBAT  drop constraint FK\_GOLONGAN\_KATEGORI\_OBAT;  alter table LOKASI\_OBAT  drop constraint FK\_LOKASI\_O\_DISIMPAN\_OBAT;  alter table LOKASI\_OBAT  drop constraint FK\_LOKASI\_O\_TERDAPAT\_RAK;  alter table MENGANDUNG  drop constraint FK\_MENGANDU\_RELATIONS\_OBAT;  alter table MENGANDUNG  drop constraint FK\_MENGANDU\_RELATIONS\_KOMPOSIS;  alter table RESEP\_OBAT  drop constraint FK\_RESEP\_OB\_MEMILIKI\_\_KONSUMEN;  alter table SATUAN\_OBAT  drop constraint FK\_SATUAN\_O\_BERUPA\_OBAT;  alter table SEDIA  drop constraint FK\_SEDIA\_RELATIONS\_SUPPLIER;  alter table SEDIA  drop constraint FK\_SEDIA\_RELATIONS\_OBAT;  drop table APOTEK\_S\_ORDER cascade constraints;  drop index MEMILIKI\_2\_FK;  drop table BATCH cascade constraints;  drop index RELATIONSHIP\_14\_FK;  drop index RELATIONSHIP\_13\_FK;  drop table FAKTUR\_PEMESANAN cascade constraints;  drop index RELATIONSHIP\_16\_FK;  drop index RELATIONSHIP\_15\_FK;  drop table FAKTUR\_PENJUALAN cascade constraints;  drop index KATEGORI\_FK;  drop table GOLONGAN\_OBAT cascade constraints;  drop table KOMPOSISI cascade constraints;  drop table KONSUMEN\_S\_ORDER cascade constraints;  drop index DISIMPAN\_FK;  drop table LOKASI\_OBAT cascade constraints;  drop index RELATIONSHIP\_18\_FK;  drop index RELATIONSHIP\_17\_FK;  drop table MENGANDUNG cascade constraints;  drop table OBAT cascade constraints;  drop table RAK cascade constraints;  drop table RESEP\_OBAT cascade constraints;  drop index BERUPA\_FK;  drop table SATUAN\_OBAT cascade constraints;  drop index RELATIONSHIP\_4\_FK;  drop index RELATIONSHIP\_3\_FK;  drop table SEDIA cascade constraints;  drop table SUPPLIER cascade constraints;  /\*==============================================================\*/  /\* Table: APOTEK\_S\_ORDER \*/  /\*==============================================================\*/  create table APOTEK\_S\_ORDER  (  NO\_ORDER\_APOTEK CHAR(8) not null,  TANGGAL\_PEMESANAN DATE,  constraint PK\_APOTEK\_S\_ORDER primary key (NO\_ORDER\_APOTEK)  );  /\*==============================================================\*/  /\* Table: BATCH \*/  /\*==============================================================\*/  create table BATCH  (  ID\_BATCH CHAR(6) not null,  ID\_OBAT CHAR(8),  EXP\_DATE DATE,  PROD\_DATE DATE,  constraint PK\_BATCH primary key (ID\_BATCH)  );  /\*==============================================================\*/  /\* Index: MEMILIKI\_2\_FK \*/  /\*==============================================================\*/  create index MEMILIKI\_2\_FK on BATCH (  ID\_OBAT ASC  );  /\*==============================================================\*/  /\* Table: FAKTUR\_PEMESANAN \*/  /\*==============================================================\*/  create table FAKTUR\_PEMESANAN  (  ID\_SUPPLIER CHAR(8) not null,  ID\_OBAT CHAR(8) not null,  NO\_ORDER\_APOTEK CHAR(8) not null,  JUMLAH\_PEMESANAN INTEGER,  constraint PK\_FAKTUR\_PEMESANAN primary key (ID\_SUPPLIER, ID\_OBAT, NO\_ORDER\_APOTEK)  );  /\*==============================================================\*/  /\* Index: RELATIONSHIP\_13\_FK \*/  /\*==============================================================\*/  create index RELATIONSHIP\_13\_FK on FAKTUR\_PEMESANAN (  ID\_SUPPLIER ASC,  ID\_OBAT ASC  );  /\*==============================================================\*/  /\* Index: RELATIONSHIP\_14\_FK \*/  /\*==============================================================\*/  create index RELATIONSHIP\_14\_FK on FAKTUR\_PEMESANAN (  NO\_ORDER\_APOTEK ASC  );  /\*==============================================================\*/  /\* Table: FAKTUR\_PENJUALAN \*/  /\*==============================================================\*/  create table FAKTUR\_PENJUALAN  (  ID\_OBAT CHAR(8) not null,  NO\_ORDER\_KONSUMEN CHAR(8) not null,  JUMLAH\_PEMBELIAN INTEGER,  HARGA\_JUAL NUMBER(8,2),  constraint PK\_FAKTUR\_PENJUALAN primary key (ID\_OBAT, NO\_ORDER\_KONSUMEN)  );  /\*==============================================================\*/  /\* Index: RELATIONSHIP\_15\_FK \*/  /\*==============================================================\*/  create index RELATIONSHIP\_15\_FK on FAKTUR\_PENJUALAN (  ID\_OBAT ASC  );  /\*==============================================================\*/  /\* Index: RELATIONSHIP\_16\_FK \*/  /\*==============================================================\*/  create index RELATIONSHIP\_16\_FK on FAKTUR\_PENJUALAN (  NO\_ORDER\_KONSUMEN ASC  );  /\*==============================================================\*/  /\* Table: GOLONGAN\_OBAT \*/  /\*==============================================================\*/  create table GOLONGAN\_OBAT  (  NAMA\_KATEGORI CHAR(25) not null,  ID\_OBAT CHAR(8),  constraint PK\_GOLONGAN\_OBAT primary key (NAMA\_KATEGORI)  );  /\*==============================================================\*/  /\* Index: KATEGORI\_FK \*/  /\*==============================================================\*/  create index KATEGORI\_FK on GOLONGAN\_OBAT (  ID\_OBAT ASC  );  /\*==============================================================\*/  /\* Table: KOMPOSISI \*/  /\*==============================================================\*/  create table KOMPOSISI  (  NM\_ZAT CHAR(40) not null,  constraint PK\_KOMPOSISI primary key (NM\_ZAT)  );  /\*==============================================================\*/  /\* Table: KONSUMEN\_S\_ORDER \*/  /\*==============================================================\*/  create table KONSUMEN\_S\_ORDER  (  NO\_ORDER\_KONSUMEN CHAR(8) not null,  TANGGAL\_PEMBELIAN DATE,  constraint PK\_KONSUMEN\_S\_ORDER primary key (NO\_ORDER\_KONSUMEN)  );  /\*==============================================================\*/  /\* Table: LOKASI\_OBAT \*/  /\*==============================================================\*/  create table LOKASI\_OBAT  (  ID\_RAK CHAR(3) not null,  ID\_OBAT CHAR(8),  NO\_BARIS INTEGER,  NO\_KOLOM INTEGER,  constraint PK\_LOKASI\_OBAT primary key (ID\_RAK)  );  /\*==============================================================\*/  /\* Index: DISIMPAN\_FK \*/  /\*==============================================================\*/  create index DISIMPAN\_FK on LOKASI\_OBAT (  ID\_OBAT ASC  );  /\*==============================================================\*/  /\* Table: MENGANDUNG \*/  /\*==============================================================\*/  create table MENGANDUNG  (  ID\_OBAT CHAR(8) not null,  NM\_ZAT CHAR(40) not null,  constraint PK\_MENGANDUNG primary key (ID\_OBAT, NM\_ZAT)  );  /\*==============================================================\*/  /\* Index: RELATIONSHIP\_17\_FK \*/  /\*==============================================================\*/  create index RELATIONSHIP\_17\_FK on MENGANDUNG (  ID\_OBAT ASC  );  /\*==============================================================\*/  /\* Index: RELATIONSHIP\_18\_FK \*/  /\*==============================================================\*/  create index RELATIONSHIP\_18\_FK on MENGANDUNG (  NM\_ZAT ASC  );  /\*==============================================================\*/  /\* Table: OBAT \*/  /\*==============================================================\*/  create table OBAT  (  ID\_OBAT CHAR(8) not null,  NAMA\_OBAT CHAR(32),  ATURAN\_PAKAI CLOB,  APOTEK\_S\_STOK INTEGER,  BATAS\_MIN\_STOK INTEGER,  NETTO INTEGER,  constraint PK\_OBAT primary key (ID\_OBAT)  );  /\*==============================================================\*/  /\* Table: RAK \*/  /\*==============================================================\*/  create table RAK  (  ID\_RAK CHAR(2) not null,  constraint PK\_RAK primary key (ID\_RAK)  );  /\*==============================================================\*/  /\* Table: RESEP\_OBAT \*/  /\*==============================================================\*/  create table RESEP\_OBAT  (  NO\_ORDER\_KONSUMEN CHAR(8) not null,  NAMA\_DOKTER CHAR(32),  constraint PK\_RESEP\_OBAT primary key (NO\_ORDER\_KONSUMEN)  );  /\*==============================================================\*/  /\* Table: SATUAN\_OBAT \*/  /\*==============================================================\*/  create table SATUAN\_OBAT  (  SATUAN\_OBAT CHAR(30) not null,  ID\_OBAT CHAR(8),  constraint PK\_SATUAN\_OBAT primary key (SATUAN\_OBAT)  );  /\*==============================================================\*/  /\* Index: BERUPA\_FK \*/  /\*==============================================================\*/  create index BERUPA\_FK on SATUAN\_OBAT (  ID\_OBAT ASC  );  /\*==============================================================\*/  /\* Table: SEDIA \*/  /\*==============================================================\*/  create table SEDIA  (  ID\_SUPPLIER CHAR(10) not null,  ID\_OBAT CHAR(15) not null,  HARGA\_BELI NUMBER(8,2),  SUPPLIER\_S\_STOK INTEGER,  constraint PK\_SEDIA primary key (ID\_SUPPLIER, ID\_OBAT)  );  /\*==============================================================\*/  /\* Index: RELATIONSHIP\_3\_FK \*/  /\*==============================================================\*/  create index RELATIONSHIP\_3\_FK on SEDIA (  ID\_SUPPLIER ASC  );  /\*==============================================================\*/  /\* Index: RELATIONSHIP\_4\_FK \*/  /\*==============================================================\*/  create index RELATIONSHIP\_4\_FK on SEDIA (  ID\_OBAT ASC  );  /\*==============================================================\*/  /\* Table: SUPPLIER \*/  /\*==============================================================\*/  create table SUPPLIER  (  ID\_SUPPLIER CHAR(8) not null,  NAMA\_SUPPLIER CHAR(32),  ALAMAT CLOB,  NO\_TELP\_SUPPLIER CHAR(15),  constraint PK\_SUPPLIER primary key (ID\_SUPPLIER)  );  alter table BATCH  add constraint FK\_BATCH\_MEMILIKI\_\_OBAT foreign key (ID\_OBAT)  references OBAT (ID\_OBAT);  alter table FAKTUR\_PEMESANAN  add constraint FK\_FAKTUR\_P\_RELATIONS\_SEDIA foreign key (ID\_SUPPLIER, ID\_OBAT)  references SEDIA (ID\_SUPPLIER, ID\_OBAT);  alter table FAKTUR\_PEMESANAN  add constraint FK\_FAKTUR\_P\_RELATIONS\_APOTEK\_S foreign key (NO\_ORDER\_APOTEK)  references APOTEK\_S\_ORDER (NO\_ORDER\_APOTEK);  alter table FAKTUR\_PENJUALAN  add constraint FK\_FAKTUR\_P\_RELATIONS\_OBAT foreign key (ID\_OBAT)  references OBAT (ID\_OBAT);  alter table FAKTUR\_PENJUALAN  add constraint FK\_FAKTUR\_P\_RELATIONS\_KONSUMEN foreign key (NO\_ORDER\_KONSUMEN)  references KONSUMEN\_S\_ORDER (NO\_ORDER\_KONSUMEN);  alter table GOLONGAN\_OBAT  add constraint FK\_GOLONGAN\_KATEGORI\_OBAT foreign key (ID\_OBAT)  references OBAT (ID\_OBAT);  alter table LOKASI\_OBAT  add constraint FK\_LOKASI\_O\_DISIMPAN\_OBAT foreign key (ID\_OBAT)  references OBAT (ID\_OBAT);  alter table LOKASI\_OBAT  add constraint FK\_LOKASI\_O\_TERDAPAT\_RAK foreign key (ID\_RAK)  references RAK (ID\_RAK);  alter table MENGANDUNG  add constraint FK\_MENGANDU\_RELATIONS\_OBAT foreign key (ID\_OBAT)  references OBAT (ID\_OBAT);  alter table MENGANDUNG  add constraint FK\_MENGANDU\_RELATIONS\_KOMPOSIS foreign key (NM\_ZAT)  references KOMPOSISI (NM\_ZAT);  alter table RESEP\_OBAT  add constraint FK\_RESEP\_OB\_MEMILIKI\_\_KONSUMEN foreign key (NO\_ORDER\_KONSUMEN)  references KONSUMEN\_S\_ORDER (NO\_ORDER\_KONSUMEN);  alter table SATUAN\_OBAT  add constraint FK\_SATUAN\_O\_BERUPA\_OBAT foreign key (ID\_OBAT)  references OBAT (ID\_OBAT);  alter table SEDIA  add constraint FK\_SEDIA\_RELATIONS\_SUPPLIER foreign key (ID\_SUPPLIER)  references SUPPLIER (ID\_SUPPLIER);  alter table SEDIA  add constraint FK\_SEDIA\_RELATIONS\_OBAT foreign key (ID\_OBAT)  references OBAT (ID\_OBAT); |

* 1. **Procedure**

|  |
| --- |
| create or replace PROCEDURE addDataObat (IN\_ID\_OBAT IN OBAT.ID\_OBAT%TYPE, IN\_NAMA\_OBAT IN OBAT.NAMA\_OBAT%TYPE, IN\_EXP IN BATCH.EXP\_DATE%TYPE, IN\_PROD IN BATCH.PROD\_DATE%TYPE, IN\_GOLONGAN IN GOLONGAN\_OBAT.NAMA\_KATEGORI%TYPE, IN\_SATUAN IN SATUAN\_OBAT.SATUAN\_OBAT%TYPE, IN\_KOMPOSISI IN KOMPOSISI.NM\_ZAT%TYPE, IN\_ATURAN\_PAKAI IN OBAT.ATURAN\_PAKAI%TYPE, IN\_APOTEK\_S\_STOK IN OBAT.APOTEK\_S\_STOK%TYPE, IN\_BATAS\_MIN\_STOK IN OBAT.BATAS\_MIN\_STOK%TYPE, IN\_NETTO IN OBAT.NETTO%TYPE) AS  N\_ID\_OBAT VARCHAR2 (8);  BEGIN  N\_ID\_OBAT := CONCAT('OBT', TO\_CHAR(ID\_OBAT\_SEQ.NEXTVAL, '00000'));  ADDOBAT(N\_ID\_OBAT, IN\_NAMA\_OBAT, IN\_ATURAN\_PAKAI, IN\_APOTEK\_S\_STOK, IN\_BATAS\_MIN\_STOK, IN\_NETTO);  ADDBATCH(N\_ID\_OBAT, IN\_EXP, IN\_PROD);  ADDGOLONGAN\_OBAT(N\_ID\_OBAT, IN\_GOLONGAN);  ADDSATUAN\_OBAT(N\_ID\_OBAT, IN\_SATUAN);  ADDKOMPOSISI(IN\_KOMPOSISI);  END; |
| create or replace PROCEDURE addOBAT (IN\_ID\_OBAT IN OBAT.ID\_OBAT%TYPE, IN\_NAMA\_OBAT IN OBAT.NAMA\_OBAT%TYPE, IN\_ATURAN\_PAKAI IN OBAT.ATURAN\_PAKAI%TYPE, IN\_APOTEK\_S\_STOK IN OBAT.APOTEK\_S\_STOK%TYPE, IN\_BATAS\_MIN\_STOK IN OBAT.BATAS\_MIN\_STOK%TYPE, IN\_NETTO IN OBAT.NETTO%TYPE) AS  BEGIN  INSERT INTO OBAT(ID\_OBAT, NAMA\_OBAT, ATURAN\_PAKAI, APOTEK\_S\_STOK, BATAS\_MIN\_STOK, NETTO)  VALUES (IN\_ID\_OBAT, IN\_NAMA\_OBAT, IN\_ATURAN\_PAKAI, IN\_APOTEK\_S\_STOK, IN\_BATAS\_MIN\_STOK, IN\_NETTO);  END; |
| create or replace PROCEDURE addBatch (ENTRY\_OBAT IN BATCH.ID\_OBAT%TYPE, ENTRY\_EXP IN BATCH.EXP\_DATE%TYPE, ENTRY\_PROD IN BATCH.PROD\_DATE%TYPE) AS    BEGIN  INSERT INTO BATCH(ID\_OBAT, EXP\_DATE, PROD\_DATE)  VALUES (ENTRY\_OBAT, ENTRY\_EXP, ENTRY\_PROD);  END; |
| create or replace PROCEDURE addApotekOrder (ENTRY\_NO IN APOTEK\_S\_ORDER.NO\_ORDER\_APOTEK%TYPE) AS    BEGIN  INSERT INTO APOTEK\_S\_ORDER(NO\_ORDER\_APOTEK, TANGGAL\_PEMESANAN)  VALUES (ENTRY\_NO, SYSDATE);  END; |
| create or replace PROCEDURE addFakturPemesanan (ENTRY\_ID\_SUP IN FAKTUR\_PEMESANAN.ID\_SUPPLIER%TYPE, ENTRY\_ID\_OBAT IN FAKTUR\_PEMESANAN.ID\_OBAT%TYPE, ENTRY\_NO\_ORDER IN FAKTUR\_PEMESANAN.NO\_ORDER\_APOTEK%TYPE, ENTRY\_JUMLAH IN FAKTUR\_PEMESANAN.JUMLAH\_PEMESANAN%TYPE) AS    BEGIN  INSERT INTO FAKTUR\_PEMESANAN(ID\_SUPPLIER, ID\_OBAT, NO\_ORDER\_APOTEK, JUMLAH\_PEMESANAN)  VALUES (ENTRY\_ID\_SUP, ENTRY\_ID\_OBAT, ENTRY\_NO\_ORDER, ENTRY\_JUMLAH);  END; |
| create or replace PROCEDURE addFAKTUR\_PENJUALAN (inID\_OBAT FAKTUR\_PENJUALAN.ID\_OBAT%TYPE, inNO\_ORDER\_KONSUMEN IN FAKTUR\_PENJUALAN.NO\_ORDER\_KONSUMEN%TYPE,  inJUMLAH\_PEMBELIAN IN FAKTUR\_PENJUALAN.JUMLAH\_PEMBELIAN%TYPE, inHARGA\_JUAL IN FAKTUR\_PENJUALAN.HARGA\_JUAL%TYPE) AS  BEGIN  INSERT INTO FAKTUR\_PENJUALAN(ID\_OBAT,NO\_ORDER\_KONSUMEN, JUMLAH\_PEMBELIAN, HARGA\_JUAL)  VALUES (inID\_OBAT, inNO\_ORDER\_KONSUMEN, inJUMLAH\_PEMBELIAN, inHARGA\_JUAL);  END; |
| create or replace PROCEDURE addGOLONGAN\_OBAT (inID\_OBAT GOLONGAN\_OBAT.ID\_OBAT%TYPE, inNAMA\_KATEGORI IN GOLONGAN\_OBAT.NAMA\_KATEGORI%TYPE) AS  BEGIN  INSERT INTO GOLONGAN\_OBAT(NAMA\_KATEGORI, ID\_OBAT)  VALUES (inNAMA\_KATEGORI, inID\_OBAT);  END; |
| create or replace PROCEDURE addKomposisi (ENTRY\_NAME IN KOMPOSISI.NM\_ZAT%TYPE) AS    BEGIN  INSERT INTO KOMPOSISI(NM\_ZAT)  VALUES (ENTRY\_NAME);  END; |
| create or replace PROCEDURE addKonsumenOrder (ENTRY\_NO IN KONSUMEN\_S\_ORDER.NO\_ORDER\_KONSUMEN%TYPE) AS    BEGIN  INSERT INTO KONSUMEN\_S\_ORDER(NO\_ORDER\_KONSUMEN, TANGGAL\_PEMBELIAN)  VALUES (ENTRY\_NO, SYSDATE);  END; |
| create or replace PROCEDURE addLOKASI\_OBAT (inID\_RAK IN LOKASI\_OBAT.ID\_RAK%TYPE, inNO\_BARIS IN LOKASI\_OBAT.NO\_BARIS%TYPE, inNO\_KOLOM IN LOKASI\_OBAT.NO\_KOLOM%TYPE) AS    BEGIN  INSERT INTO LOKASI\_OBAT(ID\_RAK, NO\_BARIS, NO\_KOLOM)  VALUES (inID\_RAK, inNO\_BARIS, inNO\_KOLOM);  END; |
| create or replace PROCEDURE addSATUAN\_OBAT (inID\_OBAT IN SATUAN\_OBAT.ID\_OBAT%TYPE, inSATUAN\_OBAT IN SATUAN\_OBAT.SATUAN\_OBAT%TYPE) AS  BEGIN  INSERT INTO SATUAN\_OBAT(SATUAN\_OBAT, ID\_OBAT)  VALUES (inSATUAN\_OBAT, inID\_OBAT);  END; |
| create or replace PROCEDURE addSedia(ENTRY\_ID\_SUP IN SEDIA.ID\_SUPPLIER%TYPE, ENTRY\_ID\_OBAT IN SEDIA.ID\_OBAT%TYPE, ENTRY\_HARGA IN SEDIA.HARGA\_BELI%TYPE, ENTRY\_STOK IN SEDIA.SUPPLIER\_S\_STOK%TYPE) AS    BEGIN  INSERT INTO SEDIA(ID\_SUPPLIER, ID\_OBAT, HARGA\_BELI, SUPPLIER\_S\_STOK)  VALUES (ENTRY\_ID\_SUP, ENTRY\_ID\_OBAT, ENTRY\_HARGA, ENTRY\_STOK);  END; |
| create or replace PROCEDURE addSupplier (ENTRY\_ID IN SUPPLIER.ID\_SUPPLIER%TYPE, ENTRY\_NAME IN SUPPLIER.NAMA\_SUPPLIER%TYPE, ENTRY\_ALAMAT IN SUPPLIER.ALAMAT%TYPE, ENTRY\_NO\_TELP IN SUPPLIER.NO\_TELP\_SUPPLIER%TYPE) AS    BEGIN  INSERT INTO SUPPLIER(ID\_SUPPLIER, NAMA\_SUPPLIER, ALAMAT, NO\_TELP\_SUPPLIER)  VALUES (ENTRY\_ID, ENTRY\_NAME, ENTRY\_ALAMAT, ENTRY\_NO\_TELP);  END; |

* 1. **Trigger**

|  |
| --- |
| create or replace TRIGGER LOKASI\_OBAT\_T1  BEFORE INSERT  ON LOKASI\_OBAT  REFERENCING NEW AS NEW OLD AS OLD  FOR EACH ROW  WHEN (NEW.ID\_OBAT <> 0)  BEGIN  if (KOLOM\_SEQ.currval = 5) THEN  :NEW.NO\_KOLOM := :NEW.NO\_KOLOM || KOLOM\_SEQ.nextval;  :NEW.NO\_BARIS := :NEW.NO\_BARIS || BARIS\_SEQ.nextval;  ELSE  :NEW.NO\_KOLOM := :NEW.NO\_KOLOM || KOLOM\_SEQ.nextval;  :NEW.NO\_BARIS := :NEW.NO\_BARIS || BARIS\_SEQ.currval;  END IF;  END OBAT\_BI; |
| create or replace TRIGGER BATCH\_BEFORE\_INSERT  BEFORE  INSERT ON BATCH  FOR EACH ROW  WHEN (NEW.ID\_BATCH = NULL)  BEGIN  :NEW.ID\_BATCH := :NEW.ID\_BATCH || CONCAT('B', TO\_CHAR(BATCH\_SEQ.NEXTVAL, '00000'));  END; |