

Laporan Jurnal Manajemen Basis Data
“Basis Data Penjualan menggunakan Relasi model EER dan di SQL”



Oleh :
Kelompok 4

Dio Abiyyu Zidane Ginting (23451039)
Fadhilah Azzahra (23451084)
Deo Vani (23451015)
Akmal Eka Evandi (23451006)
Cris Fahri (23451094)

Dosen Pengampu:
Ibu Melda Pita Uli Sitompul, M.Kom

JENJANG PENDIDIKAN STRATA 1
PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
KAPUTAMA
BINJAI 2023

ABSTRAK

Tujuan penelitian ini adalah untuk menganalisis sistem penjualan melalui pemodelan basis data yang terdiri dari beberapa tabel yang saling berelasi: customer, produk, order, orderdetails, pembayaran, dan supplier. Tabel customer menyimpan informasi pelanggan, tabel produk berisi data produk, tabel order mencatat pesanan pelanggan, tabel detail order mencatat detail setiap item pesanan, dan tabel pembayaran mencatat transaksi pembayaran.

Penelitian ini menggunakan metode PostgreSQL sebagai perangkat lunak manajemen basis data. Identifikasi kebutuhan sistem, perancangan basis data melalui diagram Enhanced EntityRelationship (EER), implementasi, pengumpulan data, pengujian, validasi, dan analisis data adalah semua bagian dari proses penelitian.

Hasil penelitian menunjukkan bahwa mengelola data penjualan dari pemesanan hingga pembayaran lebih efisien dan akurat ketika tabel-tabel ini terhubung satu sama lain. Penggunaan PostgreSQL untuk pemodelan basis data penjualan meningkatkan operasional bisnis, membantu integrasi data penjualan, dan membuat sistem yang efektif dan akurat. Rekomendasi untuk pengembangan sistem basis data yang lebih komprehensif diberikan.

Kata kunci: penjualan, basis data, pelanggan, produk, permintaan, detail permintaan, pembayaran, supplier, relasi tabel, PostgreSQL, dan hubungan entitas yang diperkaya (EER).

DAFTAR ISI

DAFTAR ISI.....	i
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	1
1.3 Manfaat	1
BAB II TINJAUAN PUSTAKA	2
2.1 Pengertian Sistem Informasi.....	2
2.2 Basis Data Relasional	2
2.3 Entity-Entity Relationship (EER) Diagram.....	2
2.4 Penelitian Terdahulu.....	2
2.5 PostgreSql	2
BAB III METODE PENELITIAN	3
3.1 Alat dan Teknologi	3
3.2 Prosedur Implementasi Enhanced Entity-Relationship Diagram (EER).....	5
3.3 Prosedur Implementasi PostgreSQL	6
BAB IV ANALISIS DAN HASIL.....	7
4.1 Metode Analisis Kebutuhan Sistem	7
4.2 Perancangan Model Enhanced Entity-Relationship	9
4.3 Implementasi Database menggunakan SQL Shell (psql)	13
BAB V KESIMPULAN	24
5.1 Kesimpulan	24
5.2 Saran	24
Daftar Pustaka.....	26

BAB I

PENDAHULUAN

1.1 Latar Belakang

Untuk tetap kompetitif di pasar saat ini, perusahaan harus mengelola data penjualan dengan baik. Banyak bisnis mengalami kesulitan menggabungkan berbagai bagian data seperti data pelanggan, pesanan, produk, pembayaran, dan pemasok ke dalam satu sistem yang terpadu. Akibatnya, diperlukan sebuah sistem informasi yang mampu mengelola data ini dengan baik. Sistem informasi penjualan yang menggunakan PostgreSQL sebagai basis data dirancang untuk mempermudah akses dan pengelolaan data penjualan, meningkatkan efisiensi operasional perusahaan.

1.2 Tujuan

Penelitian ini bertujuan untuk mencapai: Mengembangkan sistem informasi penjualan yang terintegrasi dengan PostgreSQL.mengevaluasi bagaimana tabel-tabel dalam basis data, termasuk tabel pelanggan, pesanan, detail pesanan, produk, pembayaran, dan pemasok, berhubungan satu sama lain.Hubungan Entity-Entity (EER) Diagram dibuat untuk mensimulasikan relasi antar tabel.mengevaluasi kinerja sistem informasi yang dirancang untuk meningkatkan efisiensi manajemen data penjualan.

1.3 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut: memberikan solusi yang efisien untuk mengelola data penjualan yang rumit.Sistem informasi yang terintegrasi meningkatkan efisiensi operasional bisnis.Membuat data penjualan lebih mudah diakses dan dianalisis, yang dapat membantu pengambilan keputusan bisnis.memberikan model basis data yang dapat digunakan sebagai dasar untuk pengembangan sistem informasi yang sebanding di masa mendatang.

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Sistem Informasi

Sistem informasi adalah kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi tersebut untuk mendukung operasi dan manajemen. Sistem ini biasanya digunakan untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi. Sistem informasi yang baik dapat meningkatkan efisiensi dan efektivitas operasional suatu organisasi.

2.2 Basis Data Relasional

Basis data relasional adalah tipe basis data yang menggunakan model tabel untuk mewakili data dan hubungan antar data tersebut. PostgreSQL adalah salah satu sistem manajemen basis data relasional yang populer dan andal, dikenal karena kemampuannya dalam menangani transaksi yang kompleks dan dukungan terhadap berbagai tipe data.

2.3 Entity-Entity Relationship (EER) Diagram

EER Diagram adalah alat pemodelan data yang digunakan untuk menggambarkan struktur logis basis data. EER Diagram menunjukkan entitas, atribut, dan hubungan antar entitas dalam basis data dengan lebih detail dibandingkan dengan ERD. Penggunaan EER Diagram dalam perancangan basis data dapat membantu dalam visualisasi dan perencanaan struktur basis data yang efisien.

2.4 Penelitian Terdahulu

Berbagai penelitian telah dilakukan untuk mengembangkan sistem informasi penjualan yang efektif. Studi-studi tersebut menunjukkan bahwa penggunaan basis data relasional dan pemodelan EER Diagram dapat meningkatkan efisiensi pengelolaan data dan memberikan kemudahan akses informasi. Penelitian ini akan melanjutkan upaya tersebut dengan fokus pada penerapan PostgreSQL sebagai basis data untuk sistem informasi penjualan.

2.5 PostgreSql

PostgreSQL adalah system manajemen basis data relasional (RDBMS) open-source yang dikenal dengan skalabilitas, keandalan, dan kinerja tinggi dalam mengelola transaksi data kompleks. PostgreSQL dikenal dengan skalabilitasnya yang tinggi, keandalannya, dan kinerjanya yang unggul dalam mengelola transaksi besar dan kompleks.

BAB III

METODE PENELITIAN

3.1 Alat dan Teknologi

1. **PostgreSQL**: Sebagai sistem manajemen basis data.
2. **SQL**: Bahasa untuk membuat database, tabel, dan melakukan operasi data.
3. **Visio**: Sebagai tools yang digunakan untuk merancang EER

Penelitian ini menggunakan metodologi penelitian pengembangan sistem yang terdiri dari beberapa tahapan:

1. Pengumpulan Data

Tahap ini bertujuan untuk mengumpulkan informasi yang diperlukan untuk memahami kebutuhan dan proses penjualan dalam bisnis. Teknik pengumpulan data yang digunakan meliputi:

Studi Literatur: Mengkaji literatur yang relevan tentang sistem penjualan, database Postgres, dan model EER untuk mendapatkan pemahaman teoritis dan referensi.

2. Analisis Kebutuhan Sistem

Tahap ini bertujuan untuk menganalisis data yang telah dikumpulkan untuk mengidentifikasi kebutuhan fungsional dan non-fungsional sistem penjualan. Kebutuhan fungsional terkait dengan fungsi-fungsi yang harus dijalankan oleh sistem, sedangkan kebutuhan non-fungsional terkait dengan aspek-aspek seperti kinerja, keamanan, dan skalabilitas sistem.

3. Perancangan Model EER

Tahap ini bertujuan untuk merancang model EER yang menggambarkan entitas, atribut, dan hubungan antar entitas yang terlibat dalam proses penjualan. Model EER ini akan membantu dalam memahami struktur data dan interaksi antar entitas dalam sistem penjualan.

4. Implementasi Sistem Penjualan

Tahap ini bertujuan untuk mengimplementasikan sistem penjualan yang telah dirancang menggunakan database Postgres dan SQL Shell. Langkah-langkah dalam tahap ini meliputi:

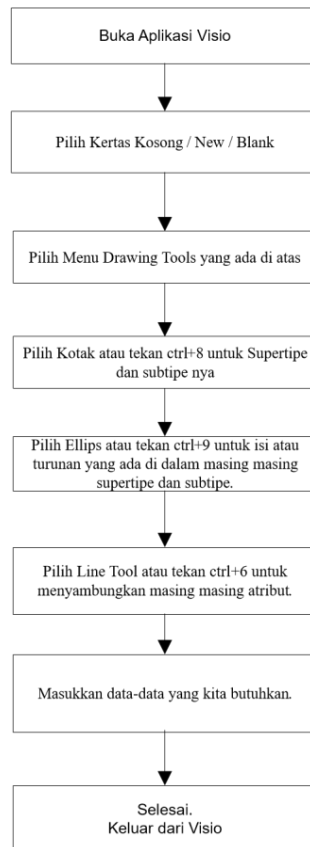
1. **Pembuatan Tabel**: Membuat tabel-tabel yang sesuai dengan entitas dalam model EER. Setiap tabel memiliki kolom yang sesuai dengan atribut entitas.
2. **Pembuatan Database**: Membuat database baru di sistem manajemen basis data Postgres.

3. Pemberian Relasi: Mendefinisikan relasi antar tabel untuk menunjukkan hubungan antar entitas.
4. Penerapan Constraint: Menerapkan constraint pada tabel untuk memastikan integritas data, seperti primary key, foreign key, dan not null.
5. Perancangan Model PostgreSQL untuk Sistem Penjualan

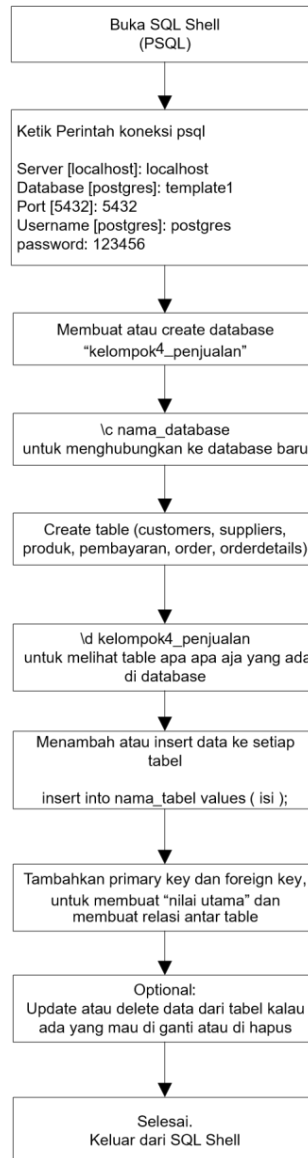
Dalam PostgreSQL, implementasi model EER untuk sistem penjualan dapat dilakukan dengan langkah-langkah berikut:

- 1) Membuat Database :
 - Gunakan perintah `CREATE DATABASE` untuk membuat database baru untuk sistem penjualan.
- 2) Membuat Tabel Entitas:
 - Buat tabel untuk setiap entitas utama, seperti customers, produk, order, orderdetails, pembayaran dan suppliers
 - Definisikan kolom untuk setiap atribut entitas, termasuk tipe data, constraint, dan nilai default.
- 3) Menentukan Hubungan Antar Entitas:
 - Gunakan constraint `FOREIGN KEY` untuk mereferensikan kolom kunci utama yang sesuai.

3.2 Prosedur Implementasi Enhanced Entity-Relationship Diagram (EER)



3.3 Prosedur Implementasi PostgreSQL



BAB IV

ANALISIS DAN HASIL

4.1 Metode Analisis Kebutuhan Sistem

Proses analisis kebutuhan sistem merupakan langkah utama dalam pengembangan sistem yang efektif. Dalam proses ini, berbagai data dan informasi dikumpulkan dari berbagai sumber, seperti jurnal, dokumentasi, blog, dan sumber terpercaya lainnya. Data-data ini kemudian diolah dan dirangkum menjadi satu kesatuan yang koheren untuk digunakan dalam pemodelan EER (Entity-Relationship Diagram) dan penerapan tabel dalam database SQL. Berikut adalah data-data yang sudah dirangkum sebagai berikut:

a) Tabel Penjualan

- ❖ id_penjualan
 - ❖ id_customer
 - ❖ id_produk
 - ❖ jumlah_barang_terjual
 - ❖ total_harga
 - ❖ cara_pembayaran
- Primary Key = id_penjualan
 - Foreign Key = id_customer (Customers), id_produk(Produk)

b) Tabel Customers

- ❖ id_customer
 - ❖ nama_customer
 - ❖ alamat
 - ❖ nomor_hp
 - ❖ email
- Primary Key = id_penjualan

c) Tabel Supplier

- ❖ id_supplier
- ❖ nama_supplier
- ❖ contact

- ❖ Alamat

- Primary Key = id_supplier

d) Tabel Produk

- ❖ id_produk

- ❖ id_supplier

- ❖ nama_produk

- ❖ deskripsi

- ❖ harga

- ❖ stok

- Primary Key = id_supplier

- Foreign Key = id_supplier (Supplier)

e) Tabel Order

- ❖ id_order

- ❖ id_customer

- ❖ orderdate

- ❖ harga_satuan

- Primary Key = id_order

- Foreign Key = id_customer (Customers)

f) Tabel OrderDetails

- ❖ id_orderdetails

- ❖ id_order

- ❖ id_produk

- ❖ jumlah

- ❖ status

- Primary Key = id_orderdetails

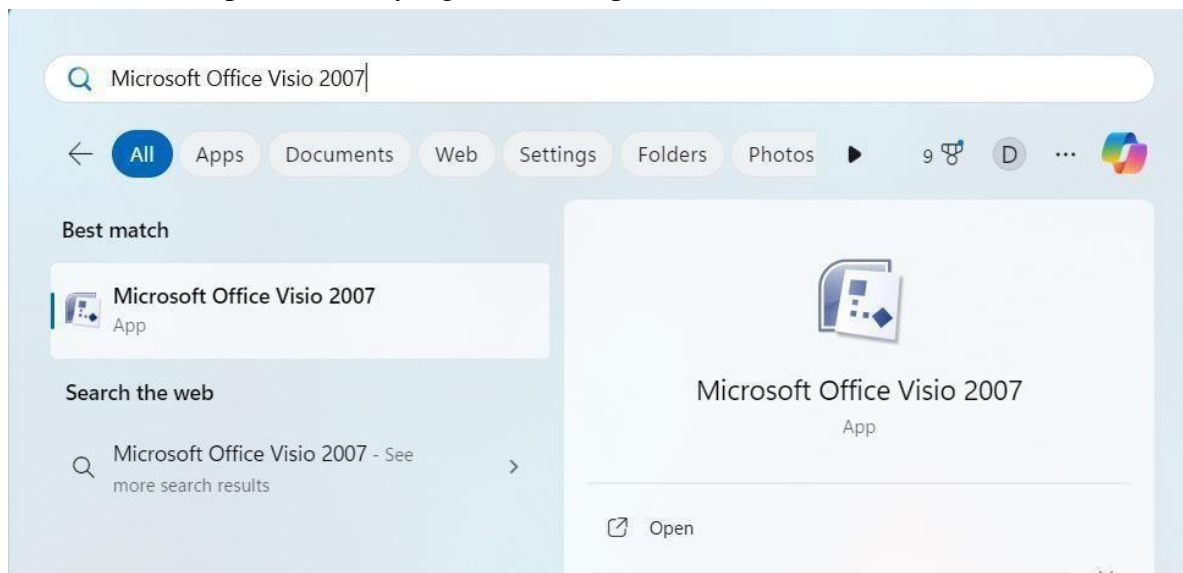
- Foreign Key = id_orders (Orders), id_produk(Produk)

g) Tabel Pembayaran

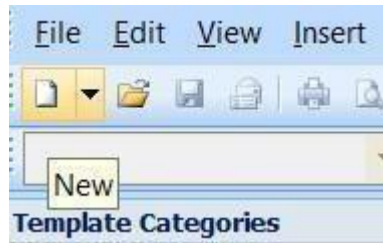
- ❖ id_pembayaran
 - ❖ id_order
 - ❖ tanggal_pembayaran
 - ❖ total_harga
 - ❖ cara_pembayaran
- Primary Key = id_pembayaran
 - Foreign Key = id_order (Orders)

4.2 Perancangan Model Enhanced Entity-Relationship

1) Buka Aplikasi Visio yang ada di Komputer anda



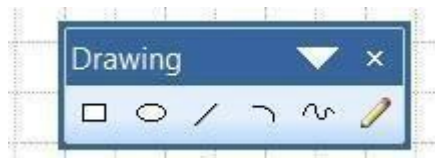
- 2) Pilih Kertas Kosong / New / Blank



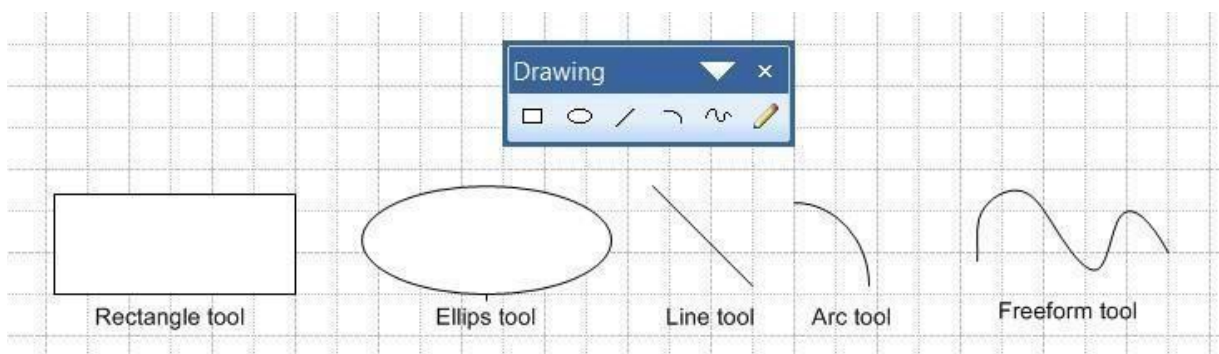
- 3) Jika di layar kamu sudah muncul kertas kosong, selanjutnya pilih menu Drawing Tools



- 4) Jika sudah di klik, maka akan tampil gambar di bawah ini berada disamping atau di dalam kertas kosong tadi



- 5) Gunakan Menu Menu ini sesuai kebutuhan anda. Disini saya hanya butuh 4 menu (Rectangle tool, Ellips tool, Line tool, Arc tool) untuk membuat EER



- 6) Sebelum membuat rangkaian EER nya. Perlu untuk mengerti kegunaan masing masing menu diatas

- **Rectangle tool (Alat Persegi Panjang):** Alat ini dapat digunakan untuk mewakili entitas supertipe dalam model EER. Entitas supertipe adalah entitas umum yang memiliki satu atau lebih subtype.

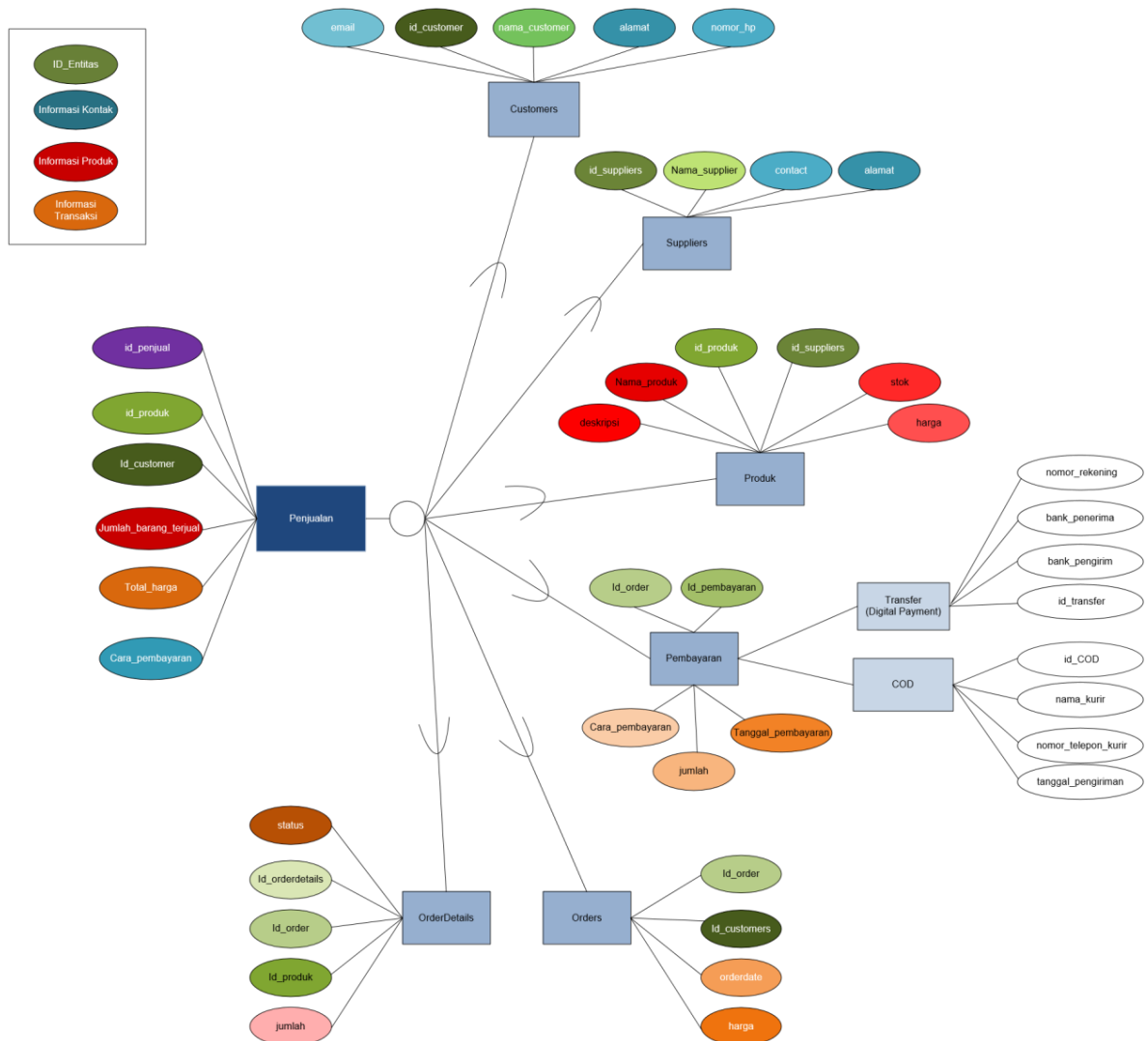
- **Ellips tool (Alat Lingkaran atau Elips):** Alat ini bisa digunakan untuk mewakili subtype dari entitas supertipe dalam model EER. Setiap lingkaran atau elips yang terhubung dengan persegi panjang (entitas) mewakili atribut dari entitas tersebut.

- **Line tool (Alat Garis):** Alat ini digunakan untuk menghubungkan antara entitas atau tabel dalam model EER. Garis ini mewakili relasi antara entitas.

- **Arc tool (Alat Busur):** Digunakan untuk menunjukkan konstruksi hubungan yang lebih kompleks antara entitas dalam model EER, seperti hubungan manyto-many atau hubungan dengan atribut. Busur ini dapat digunakan untuk menunjukkan hubungan yang lebih rinci antara entitas.

- **Freeform Tool (Alat Bentuk Bebas):** Digunakan untuk membuat bentukbentuk yang tidak terbatas, seperti garis-garis atau poligon. Jika Anda memiliki penjelasan tambahan yang tidak terhubung langsung dengan entitas, Anda bisa menggunakan alat ini untuk menambahkan teks atau simbol tambahan dalam diagram EER.

- 7) Buatlah sebuah rangkaian EER menggunakan data yang ada. Berikut rangkaian EER “Penjualan” yang sudah di buat

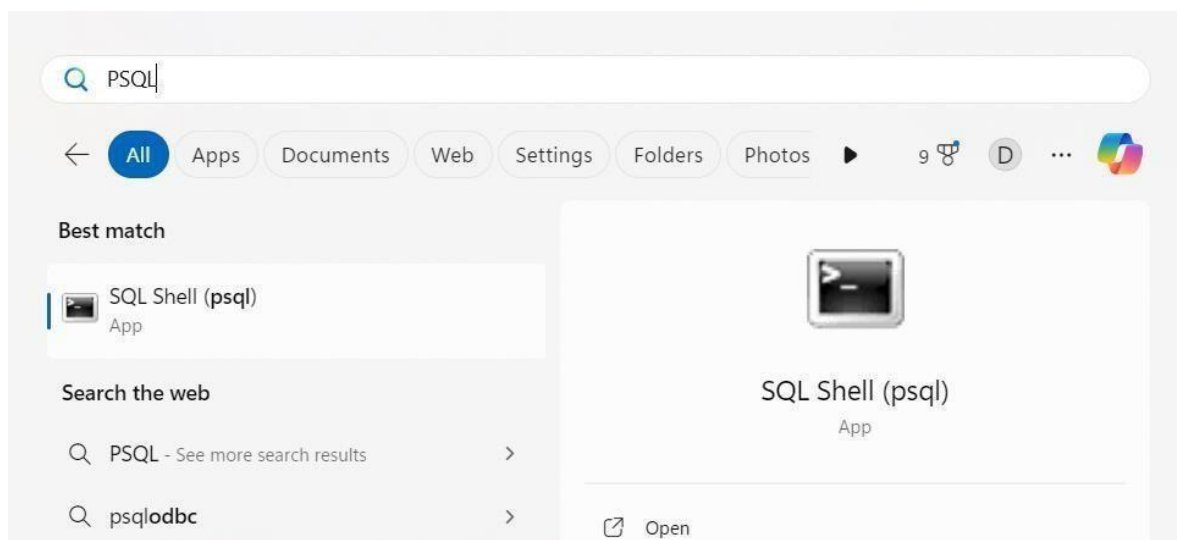


4.3 Implementasi Database menggunakan SQL Shell (psql)

Setelah Membuat rangkai EER, tahap selanjutnya adalah membuat database menggunakan SQL Shell (psql). Berikut adalah tahap-tahapan membuat Database. Sebelum masuk ke sql shell dan memulai nya. Perlu untuk mengetahui beberapa query berikut ini:

- Create database nama_database; (untuk membuat database)
- \c nama_database; (untuk menghubungkan ke database)
- Create table nama_table (data , record, constraint); (untuk membuat table)
- \d nama_table; (untuk menampilkan kolom-kolom yang ada dalam table tersebut)
- \d nama_database; (untuk menampilkan daftar tabel, indeks, dan berbagai objek lain yang ada dalam database)
- Drop table nama_table; (untuk menghapus table)
- Drop database nama_database; (untuk menghapus database)
- UPDATE nama_tabel SET kolom1 = nilai1, kolom2 = nilai2 WHERE kondisi; (Untuk melakukan pembaruan (update) data yang ada dalam sebuah tabel).
- DELETE FROM nama_tabel WHERE kondisi; (Untuk menghapus data yang ada dalam sebuah tabel)
- Insert into nama_table values; (isi data sesuai table masing masing)
- Select * from name_table; (untuk menampilkan data data yang sudah di input)

- 1) Buka sql shell yang sudah terinstal



- 2) Setelah masuk ke sql shell, isi data seperti berikut:

Server [localhost]:

Database [postgres]: template1 Port [5432]:

Username [postgres]: postgres

Password for user postgres: 123456

```
Server [localhost]:
Database [postgres]: template1
Port [5432]:
Username [postgres]: postgres
Password for user postgres:

psql (15.5)
WARNING: Console code page (437) differs from Windows code page (1252)
        8-bit characters might not work correctly. See psql reference
        page "Notes for Windows users" for details.
Type "help" for help.

template1=# |
```

- 3) Jangan khawatir jika password nya tidak muncul. Mari buat database yang didalamnya terdapat tabel-tabel yang berelasi satu sama lain.

1. Ketik Query ini untuk membuat database : Create database kelompok4_penjualan; 2.

Query ini untuk menghubungkan ke database kamu tadi:

```
\c kelompok4_penjualan;
```

```
template1=# create database kelompok4_penjualan;
CREATE DATABASE
template1=# \c kelompok4_penjualan;
You are now connected to database "kelompok4_penjualan" as user "postgres".
kelompok4_penjualan=# |
```

- 4) Ketik Query-Query ini untuk membuat table Penjualan beserta isi datanya:

a) Buat table Penjualan

```
Create table Penjualan(id_penjual char(10) PRIMARY KEY, id_customer char(10),
id_produk char(10), jumlah_barang_terjual int, total_harga decimal(10,
2), cara_pembayaran varchar(30), FOREIGN KEY (id_customer) REFERENCES
Customers(id_customer), FOREIGN KEY (id_produk) REFERENCES
Produk(id_produk));
```

b) Tampilkan struktur dan detail dari tabel tersebut

\d Penjualan;

c) Masukkan data untuk masing masing table

- Insert into Penjualan values('A1021', 'H202401', 'K12U1', 10, '200000.00', 'Debit Card');
- Insert into Penjualan values('A1022', 'H202429', 'K12U2', 10, '30000.00', 'Shopeepay');
- Insert into Penjualan values('A1023', 'H202494', 'K12U3', 20, '120000.00', 'Dana');
- Insert into Penjualan values('A1024', 'H202436', 'K12U4', 20, '160000.00', 'Shopeepay');
- Insert into Penjualan values('A1025', 'H202436', 'K12U5', 20, '20000.00', 'Gopay');
- Insert into Penjualan values('A1026', 'H202458', 'K12U5', 20, '20000.00', 'Shopeepay');

d) Tampilkan semua data yang ada dalam table penjualan:

Select * from Penjualan;

Tabel Penjualan

```
kelompok4_penjualan=# create table Penjualan(id_penjual char(10) PRIMARY KEY, id_customer char(10), id_produk char(10), jumlah_barang_terjual int, total_harga decimal(10, 2), cara_pembayaran varchar(30), FOREIGN KEY (id_customer) REFERENCES Customers(id_customer), FOREIGN KEY (id_produk) REFERENCES Produk(id_produk));
CREATE TABLE
kelompok4_penjualan=# \d Penjualan;
          Table "public.penjualan"
   Column   |      Type      | Collation | Nullable | Default
-----|-----|-----|-----|-----
 id_penjual | character(10)   |           | not null |
 id_customer | character(10)   |           |          |
 id_produk  | character(10)   |           |          |
 jumlah_barang_terjual | integer        |           |          |
 total_harga | numeric(10,2)   |           |          |
 cara_pembayaran | character varying(30) |           |          |
Indexes:
    "penjualan_pkey" PRIMARY KEY, btree (id_penjual)
Foreign-key constraints:
    "penjualan_id_customer_fkey" FOREIGN KEY (id_customer) REFERENCES customers(id_customer)
    "penjualan_id_produk_fkey" FOREIGN KEY (id_produk) REFERENCES produk(id_produk)

kelompok4_penjualan=# insert into Penjualan values('A1021','H202401','K12U1',10,'200000.00', 'Debit Card');
INSERT 0 1
kelompok4_penjualan=# insert into Penjualan values('A1022','H202429','K12U2',10,'30000.00', 'Shopeepay');
INSERT 0 1
kelompok4_penjualan=# insert into Penjualan values('A1023','H202494','K12U3',20,'120000.00', 'Dana');
INSERT 0 1
kelompok4_penjualan=# insert into Penjualan values('A1024','H202436','K12U4',20,'160000.00', 'Shopeepay');
INSERT 0 1
kelompok4_penjualan=# insert into Penjualan values('A1025','H202436','K12U5',20,'20000.00', 'Gopay');
INSERT 0 1
kelompok4_penjualan=# insert into Penjualan values('A1026','H202458','K12U5',20,'20000.00', 'Shopeepay');
INSERT 0 1
kelompok4_penjualan=# select * from Penjualan;
 id_penjual | id_customer | id_produk | jumlah_barang_terjual | total_harga | cara_pembayaran
-----|-----|-----|-----|-----|-----
 A1021      | H202401    | K12U1     | 10                  | 200000.00   | Debit Card
 A1022      | H202429    | K12U2     | 10                  | 30000.00    | Shopeepay
 A1023      | H202494    | K12U3     | 20                  | 120000.00   | Dana
 A1024      | H202436    | K12U4     | 20                  | 160000.00   | Shopeepay
 A1025      | H202436    | K12U5     | 20                  | 20000.00    | Gopay
 A1026      | H202458    | K12U5     | 20                  | 20000.00    | Shopeepay
(6 rows)
```

5) Ketik Query-Query ini untuk membuat table Customers beserta isi datanya:

- Buat tabel Customers `create table Customers (id_customer char(10) PRIMARY KEY, nama_customer varchar(30), Alamat char(30), nomor_hp char(20), email varchar(30));`
- Tampilkan struktur dan detail dari tabel tersebut
`\d Customers;`
- Masukkan data untuk masing masing tabel:
 - `Insert into Customers values ('H202401', 'Zico Lee', 'Jalan Senayan No.2', '+62895675413', 'zicoo@gmail.com');`
`Insert into Customers values ('H202429', 'Rena Liova', 'Jalan Tanah Merah', '+62813756328', 'reliva@gmail.com');` `Insert into Customers values ('H202494', 'Rami Jeera', 'Jalan Samanhudi No. 210', '+6281358632', 'rajeera@gmail.com');`
 - `Insert into Customers values ('H202436', 'Willie', 'Jalan Asia No. 24', '+6281385859', 'willie78@gmail.com');`
 - `Insert into Customers values ('H202458', 'Lee Jion', 'Jalan Asia No. 87', '+6281386468', 'jionl@gmail.com');`
- Tampilkan seluruh data dalam tabel Customers;
`Select * from Customers;`

Tabel Customers

```
kelompok4_penjualan=# create table Customers (id_customer char(10) PRIMARY KEY, nama_customer varchar(30), alamat char(30), nomor_hp char(20), email varchar
(30));
CREATE TABLE
kelompok4_penjualan=# insert into Customers values ('H202401','Zico Lee', 'Jalan Senayan No.2', '+62895675413', 'zicoo@gmail.com');
INSERT 0 1
kelompok4_penjualan=# insert into Customers values ('H202429','Rena liova', 'Jalan Tanah Merah', '+62813756328', 'reliva@gmail.com');
INSERT 0 1
kelompok4_penjualan=# insert into Customers values ('H202494','Rami Jeera', 'Jalan Samanhudi NO.210', '+6281358632', 'rajeera@gmail.com');
INSERT 0 1
kelompok4_penjualan=# insert into Customers values ('H202436','Willie', 'Jalan Asia No.24', '+6281385859', 'willie78@gmail.com');
INSERT 0 1
kelompok4_penjualan=# insert into Customers values ('H202458','Lee Jion', 'Jalan Asia No.87', '+6281386468', 'jionl@gmail.com');
INSERT 0 1
kelompok4_penjualan=#
kelompok4_penjualan=# select * from Customers;
 id_customer | nama_customer | alamat | nomor_hp | email
-----
H202401 | Zico Lee | Jalan Senayan No.2 | +62895675413 | zicoo@gmail.com
H202429 | Rena liova | Jalan Tanah Merah | +62813756328 | reliva@gmail.com
H202494 | Rami Jeera | Jalan Samanhudi NO.210 | +6281358632 | rajeera@gmail.com
H202436 | Willie | Jalan Asia No.24 | +6281385859 | willie78@gmail.com
H202458 | Lee Jion | Jalan Asia No.87 | +6281386468 | jionl@gmail.com
(5 rows)

kelompok4_penjualan=#

kelompok4_penjualan=# \d customers;
Table "public.customers"
Column | Type | Collation | Nullable | Default
-----
id_customer | character(10) | | not null |
nama_customer | character varying(30) | | |
alamat | character(30) | | |
nomor_hp | character(20) | | |
email | character varying(30) | | |
Indexes:
"customers_pkey" PRIMARY KEY, btree (id_customer)
Referenced by:
TABLE "orders" CONSTRAINT "orders_id_customer_fkey" FOREIGN KEY (id_customer) REFERENCES customers(id_customer)

kelompok4_penjualan=#
```

6) Ketik Query-Query ini untuk membuat table Suppliers beserta isi datanya:

a) Buat tabel Suppliers

create table Suppliers (id_supplier char(10) PRIMARY KEY, nama_supplier varchar(30), contact char(30), alamat char(30));

b) Tampilkan struktur dan detail dari tabel tersebut.

\d Suppliers;

c) Masukkan data untuk masing-masing tabel:

- **Insert into Suppliers values('S1151', 'Leon', '+6295836282', 'Jalan Rumbia');**
- **Insert into Suppliers values('S1152', 'Olivia', '+6295756382', 'Jalan Veteran');**
- **Insert into Suppliers values ('S1153', 'Piona', '+6213513547', 'Jalan Melati No.2');**

Insert into Suppliers values ('S1154', 'Randy Satria', '+6213646219', 'Jalan Asia Afrika');

Insert into Suppliers values ('S1155', 'Reka Putri', '+62813934739', 'Jalan Setiabudi');

d) Tampilkan seluruh data dalam tabel Suppliers

Select * from Suppliers;

Table suppliers

```
kelompok4_penjualan=# create table Suppliers (id_supplier char(10) PRIMARY KEY, nama_supplier varchar(30), contact char(30), alamat char(30));
CREATE TABLE
kelompok4_penjualan=# insert into Suppliers values ('S1151','Leon','+6295836282','Jalan Rumbia');
INSERT 0 1
kelompok4_penjualan=# insert into Suppliers values ('S1152','Olivia','+6295756382','Jalan Veteran');
INSERT 0 1
kelompok4_penjualan=# insert into Suppliers values ('S1153','Piona','+6213513547','Jalan Melati No.2');
INSERT 0 1
kelompok4_penjualan=# insert into Suppliers values ('S1154','Randy Satria','+6213646219','Jalan Asia Afrika');
INSERT 0 1
kelompok4_penjualan=# insert into Suppliers values ('S1155','Reka Putri','+62813934739','Jalan Setiabudi');
INSERT 0 1
kelompok4_penjualan=# select * from Suppliers;
 id_supplier | nama_supplier | contact | alamat
-----
S1151       | Leon          | +6295836282 | Jalan Rumbia
S1152       | Olivia        | +6295756382 | Jalan Veteran
S1153       | Piona         | +6213513547 | Jalan Melati No.2
S1154       | Randy Satria  | +6213646219 | Jalan Asia Afrika
S1155       | Reka Putri    | +62813934739 | Jalan Setiabudi
(5 rows)
```

kelompok4_penjualan=#

```
kelompok4_penjualan=# \d Suppliers;
Table "public.suppliers"
  Column      |          Type          | Collation | Nullable | Default
-----
id_supplier   | character(10)           |           | not null |
nama_supplier | character varying(30)   |           |          |
contact       | character(30)           |           |          |
alamat        | character(30)           |           |          |
Indexes:
  "suppliers_pkey" PRIMARY KEY, btree (id_supplier)
Referenced by:
  TABLE "produk" CONSTRAINT "produk_id_supplier_fkey" FOREIGN KEY (id_supplier) REFERENCES suppliers(id_supplier)
```

kelompok4_penjualan=#

7) Ketik Query-Query ini untuk membuat tabel Produk beserta isi datanya:

a) Buat tabel Produk

```
create table Produk(id_produk char(10) PRIMARY KEY, id_supplier
char(10), nama_produk varchar(20), deskripsi TEXT, harga decimal(10, 2),
stok INT,
FOREIGN KEY (id_supplier) REFERENCES Suppliers(id_supplier));
```

b) Tampilkan struktur dan detail dari tabel tersebut.

\d Produk;

c) Masukkan data untuk masing-masing tabel:

- Insert into Produk values('K12U1', 'S1151', 'Milo', 'Susu Coklat yang lezat', '20000.00', 30);
- Insert into Produk values('K12U2', 'S1152', 'Indomie', 'Mie Instan yang lezat dan beragam rasa', '3000.00', 50);
- Insert into Produk values('K12U3', 'S1153', 'AICE', 'Ice Cream beragam rasa dan lezat', '6000.00', 20);
- Insert into Produk values('K12U4', 'S1154', 'Pop Mie', 'Mie Instan Populer dengan rasa yang lezat dan praktis', '8000.00', 60);
- Insert into Produk values('K12U5', 'S1155', 'Milkuat', 'Susu Coklat yang lezat dan menyegarkan', '4000.00', 10);

d) Tampilkan seluruh data dalam tabel Produk:

Select * from Produk;

Table Produk

```
kelompok4_penjualan=# create table Produk(id_produk char(10) PRIMARY KEY, id_supplier char(10), nama_produk varchar(20), deskripsi TEXT, harga decimal(10, 2), stok INT, FOREIGN KEY (id_supplier) REFERENCES Suppliers(id_supplier));
CREATE TABLE
kelompok4_penjualan=# insert into Produk values('K12U1','S1151','Milo','Susu Coklat Yang Lezat','20000.00',30);
INSERT 0 1
kelompok4_penjualan=# insert into Produk values('K12U2','S1152','Indomie','Mie Instan yang lezat dan beragam rasa','3000.00',50);
INSERT 0 1
kelompok4_penjualan=# insert into Produk values('K12U3','S1153','AICE','Ice Cream beragam rasa dan lezat','6000.00',20);
INSERT 0 1
kelompok4_penjualan=# insert into Produk values('K12U4','S1154','Pop Mie','Mie Instan Populer dengan rasa yang lezat dan praktis','8000.00',60);
INSERT 0 1
kelompok4_penjualan=# insert into Produk values('K12U5','S1155','Milkuat','Susu Coklat yang lezat dan menyegarkan','4000.00',10);
INSERT 0 1
kelompok4_penjualan=# |
```

```
kelompok4_penjualan=# \d produk;
Table "public.produk"
Column      |          Type          | Collation | Nullable | Default
-----|-----|-----|-----|-----
id_produk   | character(10)           |           | not null |
id_supplier | character(10)           |           |         |
nama_produk | character varying(20)   |           |         |
deskripsi   | text                   |           |         |
harga       | numeric(10,2)           |           |         |
stok        | integer                |           |         |
Indexes:
    "produk_pkey" PRIMARY KEY, btree (id_produk)
Foreign-key constraints:
    "produk_id_supplier_fkey" FOREIGN KEY (id_supplier) REFERENCES suppliers(id_supplier)
Referenced by:
    TABLE "orderdetails" CONSTRAINT "orderdetails_id_produk_fkey" FOREIGN KEY (id_produk) REFERENCES produk(id_produk)
kelompok4_penjualan=# |
```

```
kelompok4_penjualan=# select * from Produk;
 id_produk | id_supplier | nama_produk |          deskripsi          |  harga  |  stok
-----|-----|-----|-----|-----|-----
 K12U1    | S1151      | Milo       | Susu Coklat Yang Lezat     | 20000.00 |   30
 K12U2    | S1152      | Indomie    | Mie Instan yang lezat dan beragam rasa |  3000.00 |   50
 K12U3    | S1153      | AICE       | Ice Cream beragam rasa dan lezat |  6000.00 |   20
 K12U4    | S1154      | Pop Mie    | Mie Instan Populer dengan rasa yang lezat dan praktis |  8000.00 |   60
 K12U5    | S1155      | Milkuat     | Susu Coklat yang lezat dan menyegarkan |  4000.00 |   10
(5 rows)
```

8) Ketik Query-Query ini untuk membuat tabel Orders beserta isi datanya:

- Buat tabel Orders `create table Orders(id_order int PRIMARY KEY, id_customer char(10), orderdate DATE, harga_satuan decimal(10,2), FOREIGN KEY(id_customer) REFERENCES Customers(id_customer));`
- Tampilkan struktur dan detail dari tabel tersebut.
`\d Orders;`
- Masukan data untuk masing-masing tabel:
 - `Insert into Orders values (131,'H202401','19-06-2024',20000.00);`
 - `Insert into Orders values (186,'H202429','23-05-2024',3000.00);`
 - `Insert into Orders values (191,'H202429','30-07-2024',6000.00);`
 - `Insert into Orders values (155,'H202494','30-07-2024',9000.00);`
 - `Insert into Orders values (143,'H202436','25-06-2024',4000.00);`
 - `Insert into Orders values (194,'H202458','23-06-2024',4000.00);`
- Tampilkan seluruh data dalam tabel Produk:
`Select * from Orders;`

Table orders

```
kelompok4_penjualan=# create table Orders (id_order int PRIMARY KEY, id_customer char(10), orderdate DATE, harga_satuan decimal(10, 2), FOREIGN KEY(id_customer) REFERENCES Customers(id_customer));
CREATE TABLE

kelompok4_penjualan=# \d Orders;
Table "public.orders"
  Column      |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----
 id_order     | integer         |           | not null |
 id_customer  | character(10)   |           |          |
 orderdate    | date            |           |          |
 harga_satuan | numeric(10,2)   |           |          |
Indexes:
    "orders_pkey" PRIMARY KEY, btree (id_order)
Foreign-key constraints:
    "orders_id_customer_fkey" FOREIGN KEY (id_customer) REFERENCES customers(id_customer)

kelompok4_penjualan=# insert into Orders values (131,'H202401','19-06-2024',20000.00);
INSERT 0 1
kelompok4_penjualan=# insert into Orders values (186,'H202429','23-05-2024',3000.00);
INSERT 0 1
kelompok4_penjualan=# insert into Orders values (191,'H202429','30-07-2024',6000.00);
INSERT 0 1
kelompok4_penjualan=# insert into Orders values (155,'H202494','30-07-2024',9000.00);
INSERT 0 1
kelompok4_penjualan=# insert into Orders values (143,'H202436','25-06-2024',4000.00);
INSERT 0 1
kelompok4_penjualan=# insert into Orders values (194,'H202458','23-06-2024',4000.00);
INSERT 0 1
kelompok4_penjualan=# select * from Orders;
 id_order | id_customer | orderdate | harga_satuan
-----+-----+-----+-----
    131  | H202401    | 2024-06-19 | 20000.00
    186  | H202429    | 2024-05-23 |  3000.00
    191  | H202429    | 2024-07-30 |  6000.00
    155  | H202494    | 2024-07-30 |  9000.00
    143  | H202436    | 2024-06-25 |  4000.00
    194  | H202458    | 2024-06-23 |  4000.00
(6 rows)
```


9) Ketik Query-Query ini untuk membuat tabel OrderDetails beserta isi datanya: a) Buat tabel Orders **create table OrderDetails(id_orderdetails int PRIMARY KEY, id_order int, id_produk char(10), jumlah int, status varchar(30), FOREIGN KEY (id_order) REFERENCES Orders(id_produk));**

b) Tampilkan struktur dan detail dari tabel tersebut.
\d OrderDetails;

c) Masukkan data untuk masing-masing tabel:

- **Insert into OrderDetails values (1001, 131, 'K12U1', 10, 'DIKIRIM');**
- **Insert into OrderDetails values (1272, 186, 'K12U2', 10, 'Telah Sampai');**
- **Insert into OrderDetails values (1254, 191, 'K12U3', 20, 'Dalam Perjalanan');**
- **Insert into OrderDetails values (1295, 155, 'K12U4', 20, 'Masih Dikemas');**
- **Insert into OrderDetails values (1232, 143, 'K12U5', 5, 'DIKIRIM');**
- **Insert into OrderDetails values (1266, 194, 'K12U5', 5, 'Dalam Perjalanan');**

d) Tampilkan seluruh data dalam tabel Produk: **Select * from OrderDetails;**

Table OrderDetails

```
SQL Shell (psql)
kelompok4_penjualan=# create table OrderDetails(id_orderdetails int PRIMARY KEY, id_order int, id_produk char(10), jumlah int, status varchar(30), FOREIGN KEY (id_order) REFERENCES Orders(id_order), FOREIGN KEY (id_produk) REFERENCES Produk(id_produk));
CREATE TABLE
kelompok4_penjualan=# \d OrderDetails;
Table "public.orderdetails"
Column | Type | Collation | Nullable | Default
-----|-----|-----|-----|-----
id_orderdetails | integer | | not null |
id_order | integer | | |
id_produk | character(10) | | |
jumlah | integer | | |
status | character varying(30) | | |
Indexes:
    "orderdetails_pkey" PRIMARY KEY, btree (id_orderdetails)
Foreign-key constraints:
    "orderdetails_id_order_fkey" FOREIGN KEY (id_order) REFERENCES orders(id_order)
    "orderdetails_id_produk_fkey" FOREIGN KEY (id_produk) REFERENCES produk(id_produk)

kelompok4_penjualan=# insert into OrderDetails values(1001,131,'K12U1',10,'DIKIRIM');
INSERT 0 1
kelompok4_penjualan=# insert into OrderDetails values(1272,186,'K12U2',10,'Telah Sampai');
INSERT 0 1
kelompok4_penjualan=# insert into OrderDetails values(1254,191,'K12U3',20,'Dalam Perjalanan');
INSERT 0 1
kelompok4_penjualan=# insert into OrderDetails values(1295,155,'K12U4',20,'Masih Dikemas');
INSERT 0 1
kelompok4_penjualan=# insert into OrderDetails values(1232,143,'K12U5',5,'DIKIRIM');
INSERT 0 1
kelompok4_penjualan=# insert into OrderDetails values(1266,194,'K12U5',5,'Dalam Perjalanan');
INSERT 0 1
kelompok4_penjualan=# select * from OrderDetails;
 id_orderdetails | id_order | id_produk | jumlah | status
-----|-----|-----|-----|-----
          1001 |      131 | K12U1 |      10 | DIKIRIM
          1272 |      186 | K12U2 |      10 | Telah Sampai
          1254 |      191 | K12U3 |      20 | Dalam Perjalanan
          1295 |      155 | K12U4 |      20 | Masih Dikemas
          1232 |      143 | K12U5 |       5 | DIKIRIM
          1266 |      194 | K12U5 |       5 | Dalam Perjalanan
(6 rows)
```

10) Ketik Query-Query ini untuk membuat tabel Pembayaran beserta isi datanya:

a) Buat tabel Pembayaran

```
create table Pembayaran(id_pembayaran char(10) PRIMARY KEY, id_order int,
tanggal_pembayaran date, total_harga decimal(10, 2), cara_pembayaran
varchar(30), FOREIGN KEY (id_order) REFERENCES Orders(id_order));
```

b) Tampilkan struktur dan detail dari tabel tersebut.

```
\d Pembayaran;
```

c) Masukkan data untuk masing-masing tabel:

- Insert into Pembayaran values('10094371','131','19-06-2024',200000.00,'Debit Card');
- Insert into Pembayaran values('10094371','186','23-05-2024',30000.00,'Shopeepay');
- Insert into Pembayaran values('10094353','191','23-06-2024',120000.00,'Dana');
- Insert into Pembayaran values('10094444','155','30-07-2024',160000.00,'Shopeepay');
- Insert into Pembayaran values('10098465','143','28-06-2024',20000.00,'Gopay');
- Insert into Pembayaran values('10098456','194','25-06-2024',20000.00,'Shopeepay');

d) Tampilkan seluruh data dalam tabel Pembayaran:

```
Select * from Pembayaran;
```

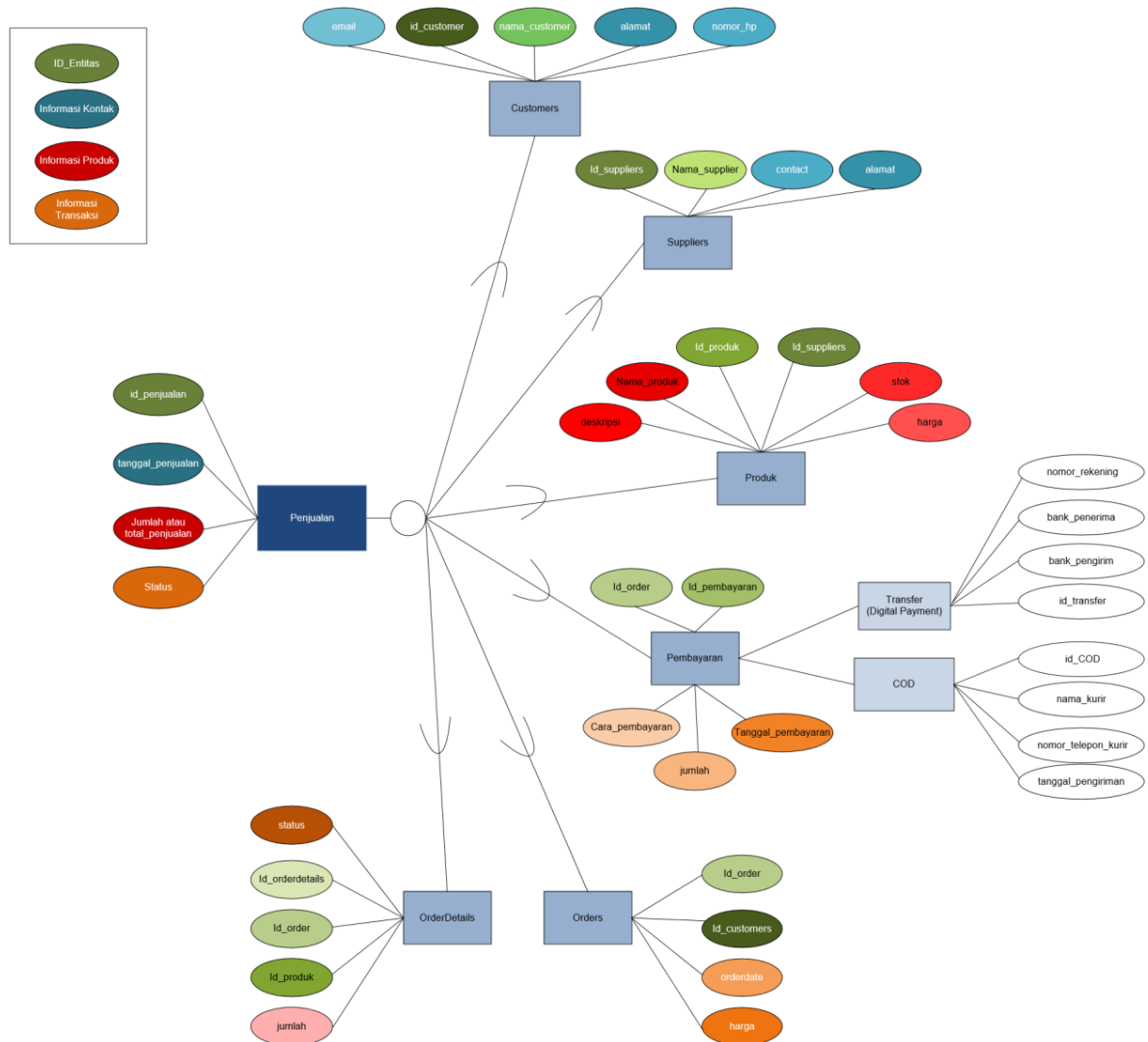
Tabel Pembayaran

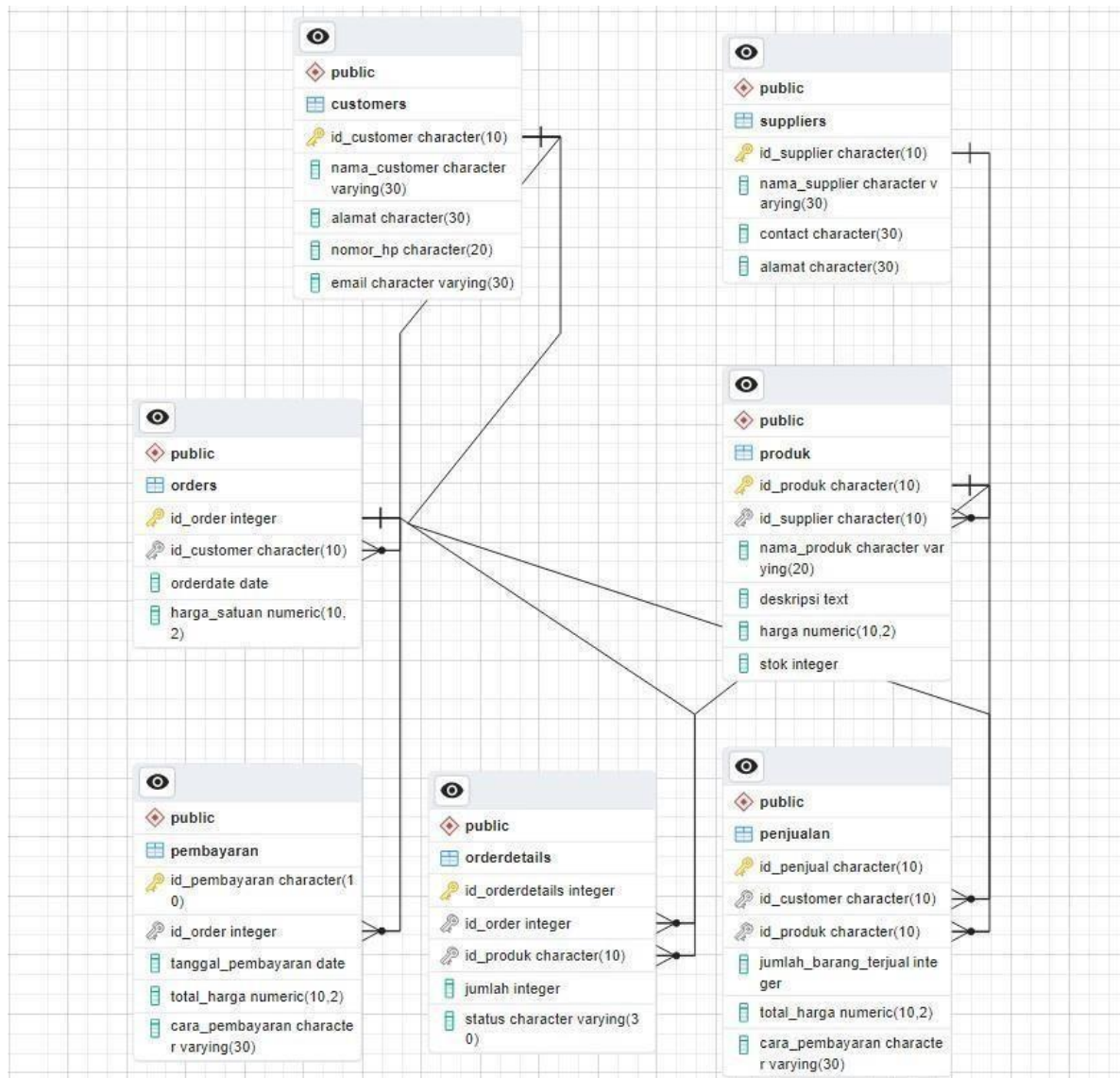
```
kelompok4_penjualan=# create table Pembayaran(id_pembayaran char(10) PRIMARY KEY, id_order int, tanggal_pembayaran date, total_harga decimal(10, 2), cara_pembayaran varchar(30), FOREIGN KEY (id_order) REFERENCES Orders(id_order));
CREATE TABLE
kelompok4_penjualan=# \d Pembayaran;
Table "public.pembayaran"
Column          | Type          | Collation | Nullable | Default
-----
id_pembayaran   | character(10) |           | not null |
id_order        | integer       |           |          |
tanggal_pembayaran | date         |           |          |
total_harga     | numeric(10,2) |           |          |
cara_pembayaran | character varying(30) |           |          |
Indexes:
    "pembayaran_pkey" PRIMARY KEY, btree (id_pembayaran)
Foreign-key constraints:
    "pembayaran_id_order_fkey" FOREIGN KEY (id_order) REFERENCES orders(id_order)
```

```
kelompok4_penjualan=# insert into Pembayaran values('10094371','131','19-06-2024',200000.00,'Debit Card');
INSERT 0 1
kelompok4_penjualan=# insert into Pembayaran values('10094522','186','23-05-2024',30000.00,'Shopeepay');
INSERT 0 1
kelompok4_penjualan=# insert into Pembayaran values('10094353','191','23-06-2024',120000.00,'Dana');
INSERT 0 1
kelompok4_penjualan=# insert into Pembayaran values('10094444','155','30-07-2024',160000.00,'Shopeepay');
INSERT 0 1
kelompok4_penjualan=# insert into Pembayaran values('10098465','143','28-06-2024',20000.00,'Gopay');
INSERT 0 1
kelompok4_penjualan=# insert into Pembayaran values('10098456','194','25-06-2024',20000.00,'Shopeepay');
INSERT 0 1
kelompok4_penjualan=# select * from Pembayaran;
 id_pembayaran | id_order | tanggal_pembayaran | total_harga | cara_pembayaran
-----
10094371      | 131      | 2024-06-19          | 200000.00   | Debit Card
10094522      | 186      | 2024-05-23          | 30000.00    | Shopeepay
10094353      | 191      | 2024-06-23          | 120000.00   | Dana
10094444      | 155      | 2024-07-30          | 160000.00   | Shopeepay
10098465      | 143      | 2024-06-28          | 20000.00    | Gopay
10098456      | 194      | 2024-06-25          | 20000.00    | Shopeepay
(6 rows)
```

```
kelompok4_penjualan=#
```


Enhanced Entity Relationship





BAB V

KESIMPULAN

Dalam penelitian ini, telah dilakukan pengembangan sistem informasi penjualan menggunakan PostgreSQL sebagai sistem manajemen basis data. Berikut adalah kesimpulan dan saran yang dapat diambil dari penelitian ini:

5.1 Kesimpulan

1. **Pengembangan Sistem Informasi:** Pengembangan sistem informasi penjualan bertujuan untuk meningkatkan efisiensi dan efektivitas dalam pengelolaan data penjualan di perusahaan.
2. **Penerapan PostgreSQL:** PostgreSQL dipilih sebagai sistem manajemen basis data karena fleksibilitasnya dan dukungan terhadap basis data relasional, yang sesuai dengan kebutuhan pengelolaan data penjualan yang kompleks.
3. **Implementasi Basis Data:** Database penjualan telah berhasil dibuat dengan menggunakan SQL untuk membuat tabel-tabel yang diperlukan dan mengisi data dummy untuk pengujian sistem.
4. **Pembangunan Aplikasi:** Aplikasi penjualan telah dikembangkan yang terhubung dengan basis data PostgreSQL untuk mengelola data pelanggan, pesanan, produk, dan transaksi penjualan.
5. **Pengujian Sistem:** Sistem telah diuji untuk memastikan fungsionalitasnya sesuai dengan yang diharapkan dan data tersimpan dengan benar.
6. **Kesesuaian dengan Kebutuhan:** Sistem informasi penjualan yang dikembangkan telah memenuhi kebutuhan perusahaan dalam mengelola data penjualan dengan lebih efektif dan efisien.

5.2 Saran

1. **Peningkatan Performa:** Lakukan evaluasi lebih lanjut terhadap performa sistem, termasuk waktu respons dan kebutuhan memori, untuk memastikan sistem dapat berjalan dengan efisien saat digunakan secara aktif.
2. **Pengembangan Fitur Tambahan:** Pertimbangkan untuk mengembangkan fitur tambahan yang dapat meningkatkan kinerja sistem, seperti analisis data penjualan yang lebih mendalam atau integrasi dengan sistem keuangan perusahaan.
3. **Pemeliharaan Berkala:** Lakukan pemeliharaan rutin terhadap sistem, termasuk backup data secara berkala dan pemantauan terhadap integritas data, untuk memastikan kehandalan dan keamanan sistem.
4. **Pelatihan Pengguna:** Berikan pelatihan lanjutan kepada pengguna agar mereka dapat memanfaatkan sistem dengan maksimal dan memahami potensi fitur-fitur yang ada.
5. **Pengembangan Selanjutnya:** Pertimbangkan untuk mengembangkan sistem lebih lanjut dengan menambahkan modul atau fitur baru sesuai dengan perkembangan kebutuhan perusahaan.

Dengan demikian, pengembangan sistem informasi penjualan menggunakan PostgreSQL telah memberikan kontribusi yang signifikan dalam meningkatkan efisiensi operasional perusahaan dalam mengelola data penjualan.

Daftar Pustaka

1. Connolly, T., & Begg, C. (2014). Database Systems: A Practical Approach to Design, Implementation, and Management. Pearson Education Limited.
2. Date, C. J. (2004). An Introduction to Database Systems. Addison-Wesley.
3. Ramakrishnan, R., & Gehrke, J. (2003). Database Management Systems. McGraw-Hill.
4. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010). Database System Concepts. McGraw-Hill.
5. Elmasri, R., & Navathe, S. B. (2015). Fundamentals of Database Systems. Pearson Education.
6. PostgreSQL Documentation. (Diakses pada 10 Juni 2024, dari <https://www.postgresql.org/docs/>).
7. Oracle, "Introduction to Entity-Relationship Modeling". (Diakses pada 10 Juni 2024, dari https://docs.oracle.com/database/121/ADXDB/app_dev_er_concepts.htm).
8. Chong, F. C., & Ng, W. K. (2017). Developing a database system: from ER diagram to relational model and normalization. In Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends (pp. 82-104). IGI Global.

Catatan: Daftar pustaka ini merupakan referensi yang digunakan untuk pemahaman konsep dasar sistem informasi, basis data relasional, EER Diagram, dan PostgreSQL.