

LAPORAN TUGAS BESAR

PROSES BISNIS PADA SISTEM E-COMMERCE



PENYUSUN

12S20010 Suprianto Dharma Sitompul

PROGRAM STUDI SARJANA SISTEM INFORMASI
FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO
INSTITUT TEKNOLOGI DEL
DESEMBER 2022

DAFTAR ISI

DAFTAR GAMBAR	3
DAFTAR TABEL.....	4
BAB I PROSES BISNIS.....	5
BAB II DIAGRAM BASIS DATA	7
2.1 EER DIAGRAM.....	6
2.2 CDM & PDM.....	7
2.2.1 CDM	7
2.2.2 PDM.....	8
2.3 TABEL FISIK.....	9
BAB III NORMALISASI	11
3.1 TABEL PENJUAL	10
3.2 TABEL PEMBELI.....	11
3.3 TABEL TRANSAKSI	12
3.4 TABEL KIRIM.....	13
3.5 TABEL DAFTAR.....	14
3.6 TABEL BARANG.....	15
BAB IV IMPLEMENTASI	17
4.1 SCHEMA.....	17
4.2 SQL SYNTAX PEMBUATAN BASIS DATA	18
4.3 TAMPILAN DATA UNTUK SETIAP TABEL	30
4.4 BASIC QUERY	32
4.5 QUERYING MULTIPLE TABLE, IMPEMENTING VIEWS.....	36
4.6 FUNCTION DAN STORED PROCEDURE	42
BAB V	48
KESIMPULAN DAN SARAN.....	48
DAFTAR PUSTAKA	49

DAFTAR GAMBAR

Gambar 1. EER Diagram	7
Gambar 2. CDM.....	8
Gambar 3. PDM	9
Gambar 4. Tabel Fisik.....	10
Gambar 5. Tampilan Tabel Pembeli	29
Gambar 6. Tampilan Tabel Penjual	29
Gambar 7. Tampilan Table Transaksi.....	30
Gambar 8. Tampilan Tabel Daftar	30
Gambar 9. Tampilan Tabel Barang.....	30
Gambar 10. Tampilan Tabel Kirim.....	30

DAFTAR TABEL

Table 1 . Tabel Penjual	11
Table 2 . Tabel Pembeli	12
Table 3. Tabel Transaksi.....	13
Table 4. Table Kirim.....	14
Table 5. Table Daftar	15
Table 6. Table Barang.....	16

BAB I

PROSES BISNIS

1.1. Registrasi

Dalam proses ini, penjual dan pembeli terlebih dahulu melakukan pendaftaran akun untuk dapat melakukan login dan melakukan proses jual beli dan transaksi antara penjual dan pembeli.

1.2. Login

Pada proses ini, setelah melakukan pendaftaran akun, penjual dan pembeli dapat melakukan proses jual beli dengan memasukkan username dan password yang sudah didaftarkan pada proses registrasi akun.

1.3. Barang

Pada proses ini, barang didaftarkan ke system agar ditampilkan pada halaman utama. Barang juga dilengkapi dengan merek atau brand nya masing masing. Serta menampilkan harga serta dan ukuran dari barang tersebut.

1.4. Pembeli (Customer)

Pembeli melakukan proses pembelian pada aplikasi dengan syarat telah melakukan pendaftaran akun dan melakukan login pada aplikasi, setelah itu pembeli dapat melakukan pencarian produk dan memilih produk yang ada pada menu-menu produk yang ada pada aplikasi dan memasukkan ke keranjang shopee di took pilihan yang menyediakan produk yang diinginkan.

1.5. Penjual (Seller)

Penjual menyiapkan/menampilkan produk pada menu produk dan melakukan pengecekan pada pemesanan produk yang dilakukan oleh pembeli untuk dapat di proses pada tahap pembayaran dan pengiriman produk.

1.6. Penjualan

Dalam proses ini, penjual hendaknya memikirkan tempat strategis dalam menawarkan produk-produk tersebut melalui toko onlinenya sekalipun melakukan iklan dan memanfaatkan e-commerce sebaik mungkin.

1.7. Pembelian

Dalam proses ini, pembeli/konsumen telah memilih percaya pada toko online shop dan merasa barang tersebut sesuai dengan keinginan dan kebutuhannya. Biasanya pembeli yakin akan produk yang dipasarkan oleh penjual karena memiliki komentar positif akan toko tersebut dan pelayanan yang baik dan memiliki nilai lebih atas kiriman sebelumnya.

1.8. Pembayaran

Dalam proses ini, pembeli terlebih dahulu memilih jenis pembayaran yang ada diantaranya E-Money dan COD dan melakukan pembayaran sesuai jenis pembayaran yang telah dipilih. Setelah proses pembayaran, pembeli akan memilih jasa pengiriman yang biasa digunakan dan biaya pengiriman yang lebih murah dan memberikan info alamat tujuan. Setelah itu Shopee akan mengecek pembayaran dan verifikasi pembayaran dan akan memberikan notifikasi ke toko online tersebut, dan toko online akan mengatur proses pengemasan barang dan mengirimkan produk sesuai jasa pengiriman yang dipilih oleh pembeli.

1.9. Pengiriman barang

Dalam proses ini, penjual akan mengirimkan barang ke jasa pengiriman di mana kurir yang mengatur proses rute pengantaran ke tangan pembeli. Penjual akan mengatur kurir untuk mengirimkan produk tersebut sesuai arahan dan mengikuti informasi pembeli dalam mengisi alamat tujuan pengiriman produk., proses penjualan bisnis online selesai

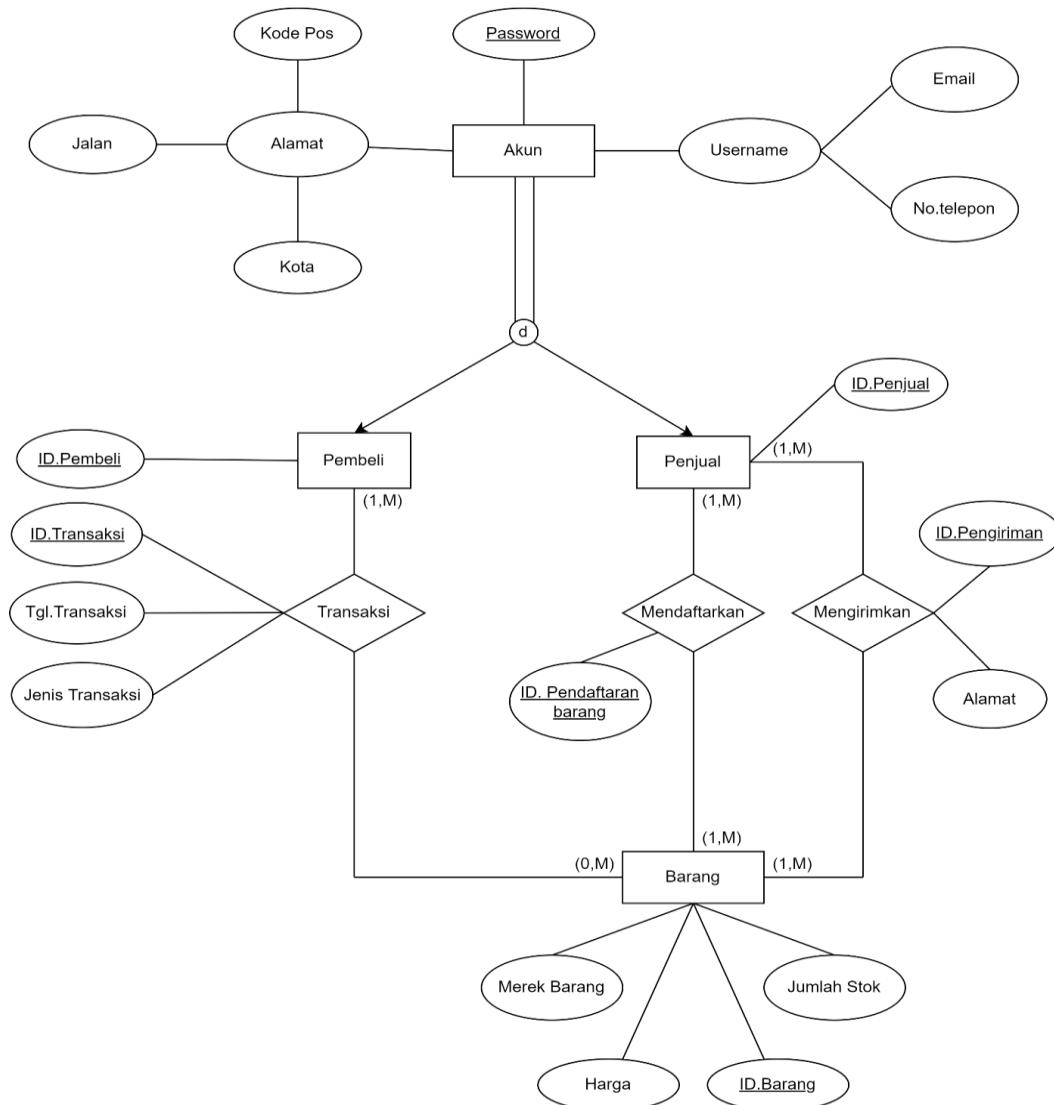
BAB II

DIAGRAM BASIS DATA

2.1. EER Diagram

ER adalah salah satu pemodelan basis data konseptual yang menggambarkan basis data ke dalam bentuk entitas-entitas dan relasi yang terjadi di antara entitas-entitas yang ada.

Berikut ER diagram dari contoh salah satu ecommerce di Indonesia.



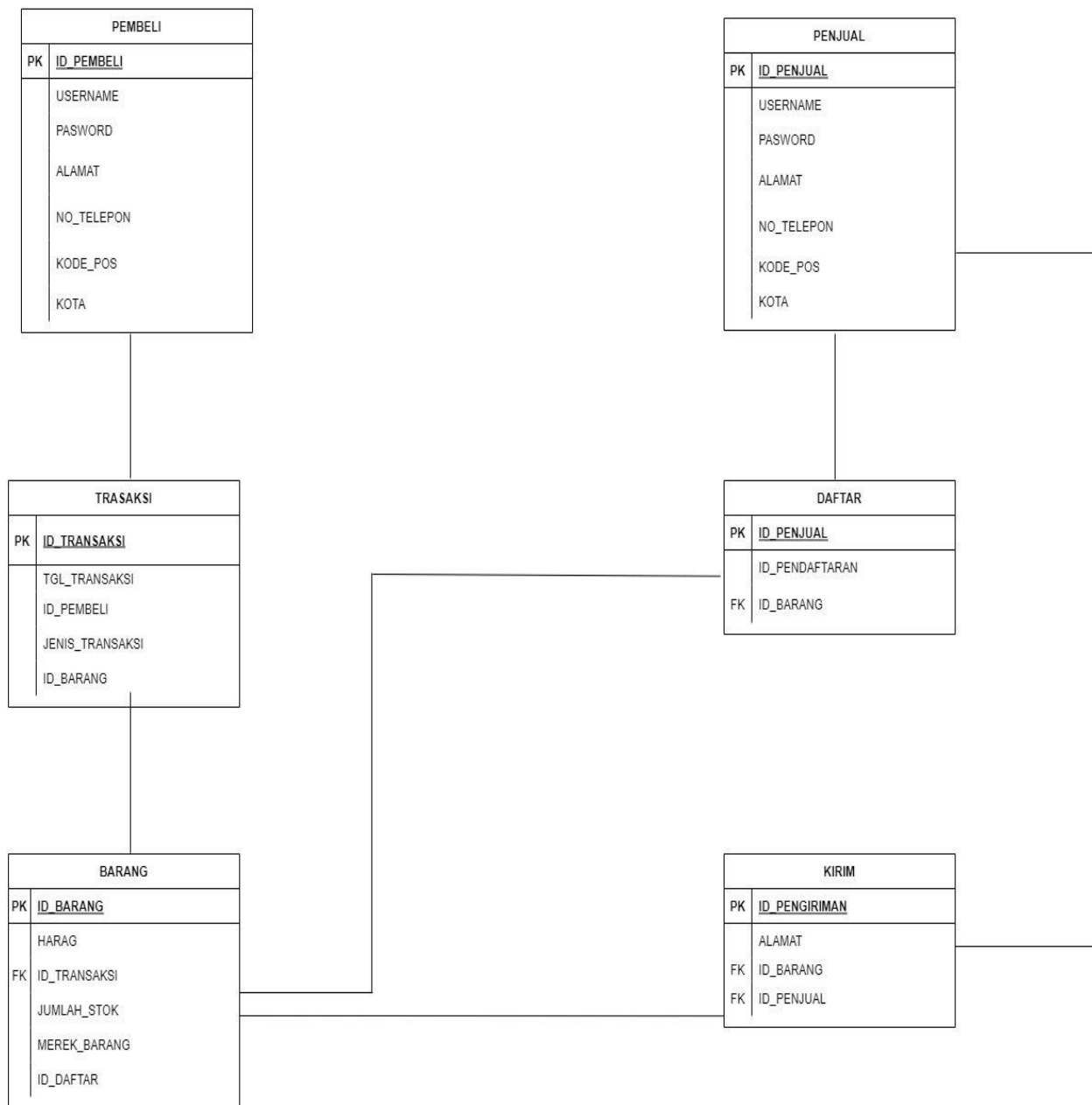
Gambar 1 – EER Diagram

2.2. CDM & PDM

2.2.1. CDM

Conceptual Data Model (CDM) atau model konsep data merupakan konsep yang berkaitan dengan pandangan pemakai terhadap data yang disimpan dalam basis data.

Berikut adalah CDM berdasarkan ER diagram dari contoh salah satu ecommerce di Indonesia

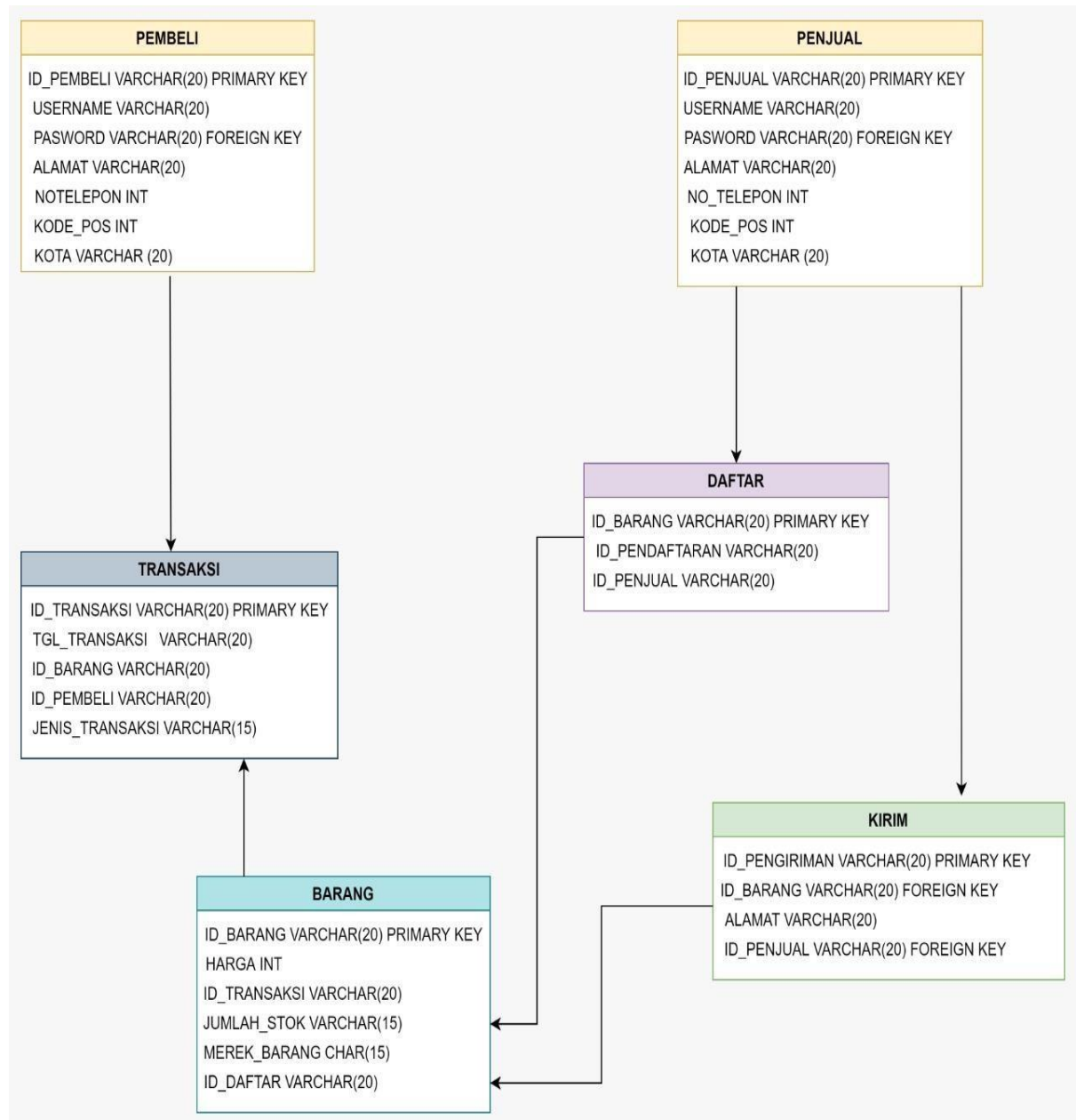


Gambar 2 - CDM

2.2.2. PDM

Physical Data Model (PDM) atau Model relasional adalah model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data.

Berikut adalah tampilan PDM berdasarkan diagram ER dan CDM yang telah Digambar sebelumnya.



Gambar 3 - PDM

2.3. TABEL FISIK

Berikut adalah transformasi erd ke basis data fisik

TABEL AKUN

PASSWORD	EMAIL	NO_TELEPON	KODE_POS	JALAN	KOTA

TABEL PEMBELI

PASSWORD	EMAIL	NO.TELEPON	KODE_POS	JALAN	KOTA	ID.PEMBELI

TABEL PENJUAL

PASSWORD	EMAIL	NO.TELEPON	KODE.POS	JALAN	KOTA	ID.PENJUAL

TABEL TRANSAKSI

ID.TRANSAKSI	ID.PEMBELI	ID.BARANG	TANGGAL TRANSAKASI	JENIS TRANSAKSI

TABEL BARANG

ID.BARANG	HARGA	JUMLAH STOK	MEREK BARANG	ID.DAFTAR

TABEL DAFTAR

ID.PENDAFTARAN	ID.PENJUAL	ID.BARANG

TABEL KIRIM

ID.PENGIRIMAN	ID.PENJUAL	ID.BARANG	ALAMAT

Gambar 4 – Tabel Fisik

BAB III

NORMALISASI

3.1. Tabel Penjual

Tabel Penjual						
ID Penjual	Username	Password	Alamat	No.telepon	Kode_pos	Kota
DN01	SUPRIANTO	12S14	JL_MERDEKA	082273671252	2889	Pasid
DN02	MASTUARI	12S24	JL_PILAR	081260813831	2889	Pasid
DN03	ANTO	12S34	JL_KENANGA	082165335651	2887	Medan
DN04	NIVERTITI	12S25	JL_DANAUTOBA	081370913100	2886	Balige
DN04	NIVER	12S26	JL_DEL_KOTA	081262247137	2885	Tarutung

1NF						
ID Penjual	Username	Password	Alamat	No.telepon	Kode_pos	Kota
DN01	SUPRIANTO	12S14	JL_MERDEKA	082273671252	2889	Pasid
DN02	MASTUARI	12S24	JL_PILAR	081260813831	2889	Pasid
DN03	ANTO	12S34	JL_KENANGA	082165335651	2887	Medan
DN04	NIVERTITI	12S25	JL_DANAUTOBA	081370913100	2886	Balige
DN04	NIVER	12S26	JL_DEL_KOTA	081262247137	2885	Tarutung

2NF			
ID Penjual	Password	Username	No.telepon
DN01	12S14	SUPRIANTO	082273671252
DN02	12S24	MASTUARI	081260813831
DN03	12S34	ANTO	082165335651
DN04	12S25	NIVERTITI	081370913100
DN04	12S26	NIVER	081262247137

ID Penjual	Kode pos	Alamat	Kota
DN01	2889	JL_MERDEKA	Pasid
DN02	2889	JL_PILAR	Pasid
DN03	2887	JL_KENANGA	Medan
DN04	2886	JL_DANAUTOBA	Balige
DN04	2885	JL_DEL_TOBA	Tarutung

3NF	
Kode pos	Kota
2889	Pasid
2887	Medan
2886	Balige
2885	Tarutung

Table 1 – Tabel Penjual

3.2. Tabel Pembeli

ID.Pembeli	Username	Password	Alamat	No_telepon	Kode_pos	Kota
12502	SUPRI	12S14	JL_PANDU	082266880407	2771	Tarutung
12506	ANTO	12S24	JL_PANDU	081346379870	2772	Siantar
12508	MASTU	12S34	JL_PODOMORO	082164677613	2773	Medan
12511	ANTON	12S25	JL_PANDUNI	082253670987	2774	Balige
12515	NIVER	12S26	JL_PODINI	082134560987	2775	Jakarta

1NF

ID.Pembeli	Username	Password	Alamat	No_telepon	Kode_pos	Kota
12502	SUPRI	12S14	JL_PANDU	082266880407	2771	Tarutung
12506	ANTO	12S24	JL_PANDU	081346379870	2772	Siantar
12508	MASTU	12S34	JL_PODOMORO	082164677613	2773	Medan
12511	ANTON	12S25	JL_PANDUNI	082253670987	2774	Balige
12515	NIVER	12S26	JL_PODINI	082134560987	2775	Jakarta

2NF

ID.Pembeli	Username	Password	No_telepon
12502	SUPRI	12S14	082266880407
12506	ANTO	12S24	081346379870
12508	MASTU	12S34	082164677613
12511	ANTON	12S25	082253670987
12515	NIVER	12S26	082134560987

ID.Pembeli	Kode_pos	Alamat	Kota
12502	2771	JL_PANDU	Tarutung
12506	2772	JL_PANDU	Siantar
12508	2773	JL_PODOMORO	Medan
12511	2774	JL_PANDUNI	Balige
12515	2775	JL_PODINI	Jakarta

Table 2 – Tabel Pembeli

3.3. Tabel Transaksi

Tabel Transaksi

id_barang	tgl_transaksi	id_transaksi	id_pembeli	Jenis Transaksi
1	40221	12502	1	BNI
2	40221	12506	2	BNI
3	50221	12508	3	BRI
4	50221	12511	4	BRI
5	60221	12515	5	BCA

1NF

id_barang	tgl_transaksi	id_transaksi	id_pembeli	Jenis Transaksi
1	40221	12502	1	BNI
2	40221	12506	2	BNI
3	50221	12508	3	BRI
4	50221	12511	4	BRI
5	60221	12515	5	BCA

2NF

id_barang	tgl_transaksi	Jenis Transaksi
1	40221	BNI
2	40221	BNI
3	50221	BRI
4	50221	BRI
5	60221	BCA

id_barang	id_transaksi	id_pembeli
1	12502	1
2	12506	2
3	12508	3
4	12511	4
5	12515	5

3NF

tgl_transaksi	Jenis Transaksi
40221	BNI
50221	BRI
60221	BCA

Table 3 – Tabel Transaksi

3.4. Tabel Kirim

Tabel Kirim			
ID.Penjual	Alamat	id_barang	id_pengiriman
DN01	JL_MERDEKA	1	JE1101
DN02	JL_PILAR	2	JE1102
DN03	JL_KENANGA	3	JE1103
DN04	JL_DANAUTOBA	4	JE1104
DN05	JL_DEL	5	JE1105

1NF			
ID.Penjual	Alamat	id_barang	id_pengiriman
DN01	JL_MERDEKA	1	JE1101
DN02	JL_PILAR	2	JE1102
DN03	JL_KENANGA	3	JE1103
DN04	JL_DANAUTOBA	4	JE1104
DN05	JL_DEL	5	JE1105

2NF	
Alamat	id_pengiriman
JL_MERDEKA	JE1101
JL_PILAR	JE1102
JL_KENANGA	JE1103
JL_DANAUTOBA	JE1104
JL_DEL	JE1105

Table 4 – Table Kirim

3.5. Tabel Daftar

Tabel Daftar		
id_penjual	id_pendaftaran	id_barang
DN01	1101	1
DN02	1102	2
DN03	1103	3
DN04	1104	4
DN05	1105	5

1NF		
id_penjual	id_pendaftaran	id_barang
DN01	1101	1
DN02	1102	2
DN03	1103	3
DN04	1104	4
DN05	1105	5

Table 5 – Table Daftar

3.6. Tabel Barang

Tabel Barang					
ID.Barang	Harga	id_transaksi	jumlah_stok	Merek Barang	id_daftar
1	150000	12502	11	GUCCI	X1
2	150000	12502	12	GUCCI	X2
3	300000	12502	16	ADIDAS	X3
4	40000	12502	19	ZARA	X4
5	50000	12502	25	NIKE	X5
6	30000	12502	35	ADIDAS	X6

1NF

ID.Barang	Harga	id_transaksi	jumlah_stok	Merek Barang	id_daftar
1	150000	12502	11	GUCCI	X1
2	150000	12502	12	GUCCI	X2
3	300000	12502	16	ADIDAS	X3
4	40000	12502	19	ZARA	X4
5	50000	12502	25	NIKE	X5
6	30000	12502	35	ADIDAS	X6

2NF

ID.Barang	Harga	Merek Barang
1	150000	GUCCI
2	150000	GUCCI
3	300000	ADIDAS
4	40000	ZARA
5	50000	NIKE
6	30000	ADIDAS

ID.Barang	id_transaksi	jumlah_stok	id_daftar
1	12502	11	X1
2	12502	12	X2
3	12502	16	X3
4	12502	19	X4
5	12502	25	X5
6	12502	35	X6

3NF

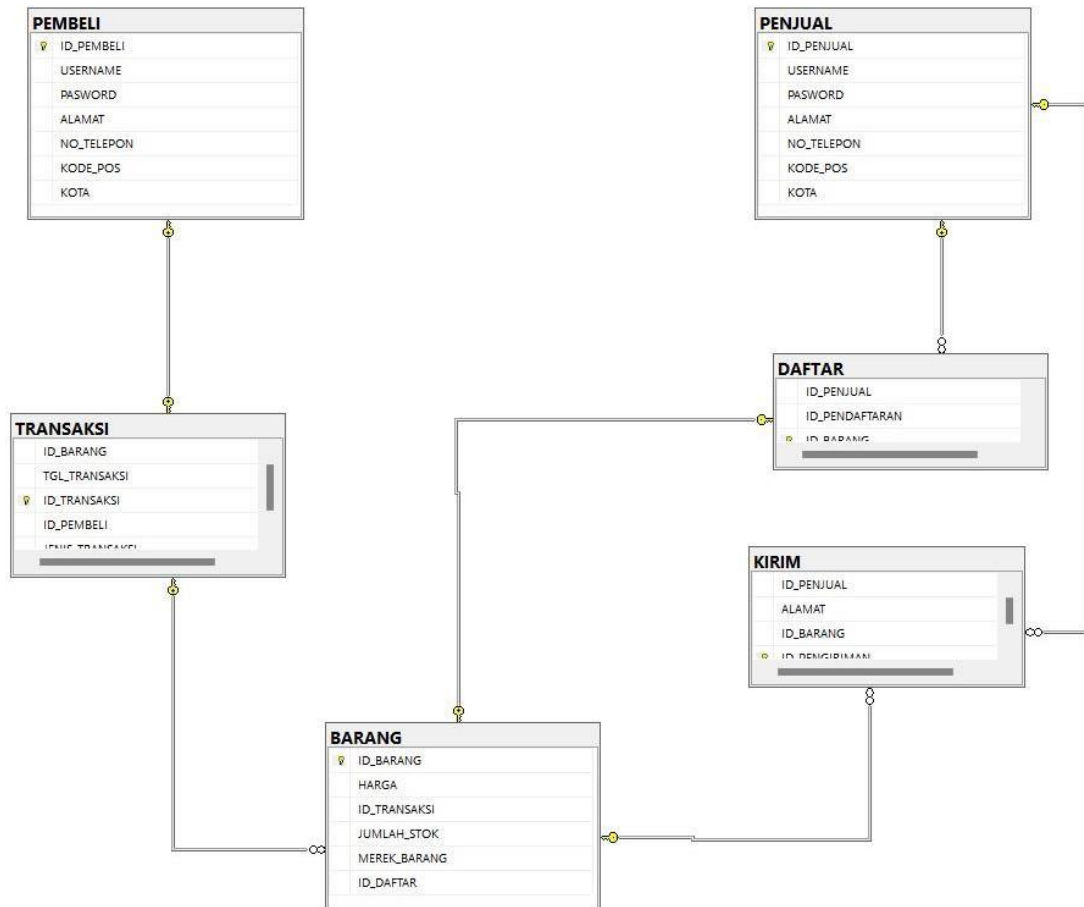
Harga	Merek Barang
150000	GUCCI
300000	ADIDAS
40000	ZARA
50000	NIKE
30000	ADIDAS

Table 6 - Table Barang

BAB IV

IMPLEMENTASI

4.1. Schema



4.2. SQL Syntax Pembuatan Basis Data

```
CREATE DATABASE BASDAT2022_12_SI
```

```
USE BASDAT2022_1_SI
```

```
CREATE TABLE PENJUAL (
```

```
    ID_PENJUAL VARCHAR(20) NOT NULL,
```

```
    USERNAME VARCHAR(20),
```

```
    PASSWORD VARCHAR(20),
```

```
    ALAMAT VARCHAR(20),
```

```
    NO_TELEPON VARCHAR(20),
```

```
    KODE_POS VARCHAR(20),
```

```
    KOTA VARCHAR(20),
```

```
    PRIMARY KEY (ID_PENJUAL));
```

```
CREATE TABLE PEMBELI (
```

```
    ID_PEMBELI VARCHAR(20) PRIMARY KEY,
```

```
    USERNAME VARCHAR(20),
```

```
    PASSWORD VARCHAR(20),
```

```
    ALAMAT VARCHAR(20),
```

```
    NO_TELEPON VARCHAR(20),
```

```
    KODE_POS VARCHAR(20),
```

```
    KOTA VARCHAR(20));
```

```
CREATE TABLE TRANSAKSI (
```

```
    ID_BARANG VARCHAR(20),
```

```
TGL_TRANSAKSI VARCHAR(20) NOT NULL,  
ID_TRANSAKSI  VARCHAR(20),  
ID_PEMBELI    VARCHAR(20),  
JENIS_TRANSAKSI VARCHAR(15) NOT NULL,  
PRIMARY KEY(ID_TRANSAKSI),  
FOREIGN KEY(ID_TRANSAKSI)  
REFERENCES PEMBELI (ID_PEMBELI));
```

```
CREATE TABLE DAFTAR (  
    ID_PENJUAL VARCHAR(20),  
    ID_PENDAFTARAN VARCHAR(20),  
    ID_BARANG   VARCHAR(20) PRIMARY KEY,  
    FOREIGN KEY (ID_PENJUAL)  
    REFERENCES PENJUAL (ID_PENJUAL));
```

```
CREATE TABLE BARANG (  
    ID_BARANG VARCHAR(20) PRIMARY KEY,  
    HARGA      INT,  
    ID_TRANSAKSI VARCHAR(20),  
    JUMLAH_STOK  VARCHAR(15),  
    MEREK_BARANG CHAR(15),  
    ID_DAFTAR   VARCHAR(20));
```

```
ALTER TABLE BARANG ADD CONSTRAINT
```

```
FK_BARANG_TRANSAKSI FOREIGN KEY
```

```
(
```

```
ID_TRANSAKSI
```

```
) REFERENCES TRANSAKSI
```

```
(
```

```
ID_TRANSAKSI
```

```
) ON UPDATE NO ACTION
```

```
ON DELETE NO ACTION
```

```
ALTER TABLE DAFTAR ADD CONSTRAINT
```

```
FK_DAFTAR_BARANG FOREIGN KEY
```

```
(
```

```
ID_BARANG
```

```
) REFERENCES BARANG
```

```
(
```

```
ID_BARANG
```

```
) ON UPDATE NO ACTION
```

```
ON DELETE NO ACTION
```

```
DROP TABLE BARANG
```

```
CREATE TABLE KIRIM(
```

```
ID_PENJUAL VARCHAR(20),
```

```

    ALAMAT          VARCHAR(20),
    ID_BARANG       VARCHAR(20),
    ID_PENGIRIMAN   VARCHAR(20) PRIMARY KEY,
    FOREIGN KEY (ID_PENJUAL) REFERENCES PENJUAL(ID_PENJUAL),
    FOREIGN KEY (ID_BARANG) REFERENCES BARANG (ID_BARANG));

```

DROP TABEL KIRIM

--- NOMOR 3A ISERTING DATA---

---PENJUAL---

```

INSERT INTO PENJUAL VALUES ('DN01', 'SUPRIANTO', '12S14',
'JL_MERDEKA',082273671252,2889,'PASID'),
('DN02', 'MASTUARI', '12S24', 'JL_PILAR',081260813831,2889,'PSID'),
('DN03', 'ANTO', '12S34', 'JL_KENANGA',082165335651,2887,'MEDAN'),
('DN04', 'NIVERTITI', '12S25', 'JL_DANAUTOBA',081370913100,2886,'BALIGE'),
('DN05', 'NIVER', '12S26', 'JL_DEL_KOTA',081262247137,2885,'TARUTUNG');
SELECT*FROM PENJUAL;

```

---PEMBELI---

```

INSERT INTO PEMBELI VALUES (12502, 'SUPRI', '12S14',
'JL_PANDU',082266880407,2771,'TARUTUNG'),
(12506, 'ANTO', '12S24', 'JL_PANDU',081346379870,2772,'SIANTAR'),
(12508, 'MASTU', '12S34', 'JL_PODOMORO',082164677613,2773,'MEDAN'),
(12511, 'ANTON', '12S25', 'JL_PANDUNI',082253670987,2774,'BALIGE'),
(12515, 'NIVER', '12S26', 'JL_PODINI',082134560987,2775,'JAKARTA');

```

```
SELECT * FROM PEMBELI;
```

```
---TRANSAKSI---
```

```
INSERT INTO TRANSAKSI VALUES (1,040221,12502,01,'BNI'),
```

```
(2,040221,12506,02,'BNI'),
```

```
(3,050221,12508,03,'BRI'),
```

```
(4,050221,12511,04,'BRI'),
```

```
(5,060221,12515,05,'BCA');
```

```
SELECT * FROM TRANSAKSI;
```

```
---BARANG---
```

```
INSERT INTO BARANG VALUES (01, 150000,12502,11,'GUCCI','X1'),
```

```
(02, 150000,12502, 12, 'GUCCI','X2'),
```

```
(03, 300000,12502, 16, 'ADIODAS','X3'),
```

```
(04, 40000,12502, 19, 'ZARA','X4'),
```

```
(05, 50000,12502, 25, 'NIKE','X5'),
```

```
(06, 30000, 12502, 35 , 'ADIDAS', 'X6');
```

```
SELECT * FROM BARANG;
```

```
---KIRIM---
```

```
INSERT INTO KIRIM VALUES ('DN01','JLMERDEKA',01,'JE1101'),
```

```
('DN02','JLPILAR',02,'JE1102'),
```

```
('DN03','JLKENANGA',03,'JE1103'),
```

```
('DN04','JLDANAUTOBA',04,'JE1104'),
```

```
('DN05','JLDEL',05,'JE1105');
```

```
SELECT * FROM KIRIM;
```

```
---DAFTAR---
```

```
INSERT INTO DAFTAR VALUES ('DN01',1101,01),
```

```
('DN02',1102, 02),
```

```
('DN03',1103, 03),
```

```
('DN04',1104, 04),
```

```
('DN05',1105, 05);
```

```
SELECT*FROM DAFTAR;
```

```
---3B SELECT*FROM---
```

```
SELECT * FROM PEMBELI
```

```
SELECT * FROM PENJUAL
```

```
SELECT * FROM TRANSAKSI
```

```
SELECT * FROM DAFTAR
```

```
SELECT * FROM BARANG
```

```
SELECT * FROM KIRIM
```

```
---3C BASIC QUERY : (SELECT, UPDATE, SET OPERATORS, AGGREGATE  
FUNCTION, NULL VALUE)---
```

```
--- 1. UPDATE ALAMAT DENGAN ID_PENJUAL = DNO1 MENJADI 'JL_MAHAKAM'
```

```
UPDATE PENJUAL
```

```
SET ALAMAT = 'JL_MAHAKAM'
```

```
WHERE ID_PENJUAL='DN01';
```

```
SELECT * FROM PENJUAL
```

---2. TAMPILKAN SELURUH DATA DARI TABEL PENJUAL DENGAN NAMA YANG BERAWALAN HURUF 'N'

```
SELECT * FROM PENJUAL  
WHERE USERNAME LIKE 'N%'
```

---3 TAMPILKAN HARGA TERTINGGI DAN TERENDAH DARI BARANG. SIMPAN DENGAN ATRIBUT DIFFERENCE

```
SELECT MAX(HARGA) - MIN(HARGA) AS DIFFERENCE  
FROM BARANG;
```

---4.MENAMPILKAN NILAI AWAL DAN AKHIR PADA TABEL BARANG---

```
SELECT ID_BARANG, HARGA "SALARY", ID_TRANSAKSI, JUMLAH_STOK  
FROM BARANG  
WHERE HARGA BETWEEN 40000 AND 150000;
```

--- 5---

```
SELECT ID_BARANG, HARGA "SALARY",  
ID_TRANSAKSI,JUMLAH_STOK,MEREK_BARANG,ID_DAFTAR  
FROM BARANG  
ORDER BY HARGA DESC;
```

---6---

```
SELECT ID_BARANG, HARGA, ID_TRANSAKSI,JUMLAH_STOK,  
MEREK_BARANG, ID_BARANG  
FROM BARANG  
WHERE HARGA >= 150000  
AND MEREK_BARANG LIKE 'G%';
```

---7 TAMPILKAN SELURUH ATRIBUT DENGAN HARGA YANG MENDAPATKAN

KENAIKAN 5 KALI SIMPAN SEBAGAI SALARY--

```
SELECT ID_BARANG, 5*HARGA AS SALARY, ID_TRANSAKSI, JUMLAH_STOK,  
MEREK_BARANG, ID_DAFTAR  
FROM BARANG
```

---8 MENAMPILKAN NILAI DALAM LIST---

```
SELECT ID_PENJUAL, ID_PENDAFTARAN, ID_BARANG  
FROM DAFTAR  
WHERE ID_PENDAFTARAN IN ('1101','1102','1105');
```

---9 URUTKAN DATA PADA TABEL BARANG BERDASARKAN HARGA
TERENDAH DAN TERBESAR---

```
SELECT ID_BARANG, HARGA "SALARY",  
ID_TRANSAKSI, JUMLAH_STOK, MEREK_BARANG, ID_DAFTAR  
FROM BARANG  
ORDER BY HARGA ASC;
```

---NOMOR 4 : QUERYING MULTIPLE TABLES, IMPLEMENTING VIEWS---

---1. TAMPILKAN HARGA TERTINGGI, ID_BARANG, DAN MEREK BARANG DARI
PRODUK YANG DIAMBIL DARI TABEL BARANG---

```
SELECT ID_BARANG, HARGA, MEREK_BARANG  
FROM BARANG  
WHERE HARGA = (SELECT MAX(HARGA) FROM BARANG);
```

---2. TAMPILKAN SELURUH DATA TRANSAKSI YANG MEMILIKI JENIS
TRANSAKSI 'BN'

```
SELECT * FROM TRANSAKSI  
JOIN BARANG  
ON BARANG.MEREK_BARANG = TRANSAKSI.ID_PEMBELI
```

```
WHERE TRANSAKSI.ID_PEMBELI = ('02');
```

---3. TAMPILKAN ID PENDAFTARAN DARI TABEL DAFTAR JOIN KE TABEL KIRIM-----

```
SELECT ID_PENDAFTARAN
FROM DAFTAR INNER JOIN KIRIM
ON DAFTAR.ID_PENJUAL = KIRIM.ID_PENJUAL;
```

---4.

```
SELECT MEREK_BARANG
FROM BARANG
WHERE ID_BARANG NOT IN('5');
```

---5

```
SELECT ID_PENJUAL
FROM PENJUAL CROSS JOIN PEMBELI
```

---4B : IMPLEMENTING VIEW---

---1---

```
CREATE VIEW AUDITBARANG
AS
SELECT ID_BARANG, HARGA, MEREK_BARANG
FROM BARANG
WHERE HARGA = (SELECT MAX(HARGA) FROM BARANG);
```

```
SELECT * FROM AUDITBARANG
```

---2---

```
CREATE VIEW PENDAFTARAN_BARANG
AS
SELECT ID_PENDAFTARAN
FROM DAFTAR INNER JOIN KIRIM
```

```
ON DAFTAR.ID_PENJUAL = KIRIM.ID_PENJUAL;
```

```
SELECT * FROM PENDAFTARAN_BARANG
```

---3---

```
CREATE VIEW LIST_MEREK
```

```
AS
```

```
SELECT MEREK_BARANG
```

```
FROM BARANG
```

```
WHERE ID_BARANG NOT IN('5');
```

```
SELECT * FROM LIST_MEREK
```

---4---

```
CREATE VIEW PENJUAL_PEMBELI
```

```
AS
```

```
SELECT ID_PENJUAL
```

```
FROM PENJUAL CROSS JOIN PEMBELI
```

```
SELECT * FROM PENJUAL_PEMBELI
```

---5---

```
CREATE VIEW PRODUKTERMURAH
```

```
AS
```

```
SELECT ID_BARANG, HARGA FROM BARANG
```

```
WHERE HARGA = (SELECT MIN(HARGA) FROM BARANG);
```

```
SELECT * FROM PRODUKTERMURAH
```

---6---

```
CREATE VIEW REVIEW_BARANG AS
```

```
SELECT * FROM BARANG
```

```
WHERE MEREK_BARANG IN ('NIKE', 'ADIDAS');
```

```
SELECT * FROM REVIEW_BARANG
```

---NOMOR 5 : FUNCTION DAN STORED PROCEDURE---

---1.FUNCTION MENAMPILKAN DATA DOSEN---

```
CREATE FUNCTION DATA_PRODUK (PRODUK VARCHAR(20))
RETURNS TABLE
(ID_BARANG VARCHAR(20),HARGA INT)
LANGUAGE 'PLPGSQL'
AS
$$
BEGIN
RETURN QUERY
(SELECT BARANG.ID_BARANG, BARANG.HARGA FROM BARANG
WHERE MEREK_BARANG = PRODUK);
END
$$
SELECT * FROM DATA_PRODUK ('ADIDAS');
```

---2. MENAMPILKAN ID_PEMBELI DAN ALAMAT DARI TABLE PEMBELI---

```
CREATE FUNCTION UDF_PEMBELI (CUSTOMER VARCHAR(20))
RETURNS TABLE (ID_PEMBELI VARCHAR(20), ALAMAT VARCHAR(20))
LANGUAGE 'PLPGSQL'
AS
$$
BEGIN
RETURN QUERY
(SELECT PEMBELI.ID_PEMBELI, PEMBELI.ALAMAT FROM PEMBELI
WHERE PEMBELI.USERNAME = CUSTOMER);
END
$$
```

```
SELECT * FROM UDF_PEMBELI('SUPRI');
```

--- STORED PROCEDURE---

---1. MEMASUKKAN DATA BARU KE DATABASE---

```
CREATE PROCEDURE I_BARANG (I_ID_BARANG VARCHAR(20), I_HARGA INT,  
I_ID_TRANSAKSI VARCHAR(20), I_JUMLAH_STOK VARCHAR(20),  
I_MERЕК_BARANG CHAR(15), I_ID_DAFTAR VARCHAR(20))
```

```
LANGUAGE 'PLPGSQL'
```

```
AS
```

```
$$
```

```
BEGIN
```

```
INSERT INTO BARANG (ID_BARANG, HARGA, ID_TRANSAKSI, JUMLAH_STOK,  
MERЕК_BARANG)
```

```
VALUES (I_ID_BARANG, I_HARGA, I_ID_TRANSAKSI, I_JUMLAH_STOK,  
I_MERЕК_BARANG);
```

```
COMMIT;
```

```
END;
```

```
$$
```

```
CALL I_BARANG ('07', '90000', '12502', '36', 'DELL', 'X7');
```

```
SELECT * FROM BARANG;
```

---2. MENGHAPUS MERЕК BARANG---

```
CREATE PROCEDURE DELETE_MERЕК_BARANG(I_ID_BARANG VARCHAR(20))
```

```
LANGUAGE 'PLPGSQL'
```

```
AS
```

```
$$
```

```

BEGIN
DELETE FROM BARANG
WHERE ID_BARANG = I_ID_BARANG;
COMMIT;
END;
$$

CALL DELETE_MERЕК_BARANG ('06');
SELECT * FROM BARANG;

```

4.3. Tampilan Data Untuk Setiap Table

BERIKUT ADALAH TAMPILAN OUTPUT SQL QUERY DIATAS.

➤ SELECT * FROM PEMBELI

Data Output

	id_pembeli (PK) character varying (20)	username character varying (20)	password character varying (20)	alamat character varying (20)	no_telepon character varying (20)	kode_pos character varying (20)	kota character varying (20)
1	12502	SUPRI	12514	JL_PANDU	82266880407	2771	Tarutung
2	12506	ANTO	12524	JL_PANDU	81346379870	2772	Siantar
3	12508	MASTU	12534	JL_PODOMORO	82164677613	2773	Medan
4	12511	ANTON	12525	JL_PANDUNI	82253670987	2774	Balige
5	12515	NIVER	12526	JL_PODINI	82134560987	2775	Jakarta

Gambar 5 – Tampilan Table Pembeli

➤ SELECT * FROM PENJUAL

Data Output

	id_penjual (PK) character varying (20)	username character varying (20)	password character varying (20)	alamat character varying (20)	no_telepon character varying (20)	kode_pos character varying (20)	kota character varying (20)
1	DN01	SUPRIANTO	12514	JL_MERDEKA	82273671252	2889	Pasid
2	DN02	MASTUARI	12524	JL_PILAR	81260813831	2889	Pasid
3	DN03	ANTO	12534	JL_KENANGA	82165335651	2887	Medan
4	DN04	NIVERTITI	12525	JL_DANAUTOBA	81370913100	2886	Balige
5	DN05	NIVER	12526	JL_DEL_KOTA	81262247137	2885	Tarutung

Gambar 6 – Tampilan Table Penjual

➤ SELECT * FROM TRANSAKSI

Data Output					
	id_barang character varying (20)	tgl_transaksi character varying (20)	id_transaksi [PK] character varying (20)	id_pembeli character varying (20)	jenis_transaksi character varying (15)
1	1	40221	12502	1	BNI
2	2	40221	12506	2	BNI
3	3	50221	12508	3	BRI
4	4	50221	12511	4	BRI
5	5	60221	12515	5	BCA

Gambar 7 – Tampilan Table Transaksi

➤ **SELECT * FROM DAFTAR**

Data Output			
	id_penjual character varying (20)	id_pendaftaran character varying (20)	id_barang [PK] character varying (20)
1	DN01	1101	1
2	DN02	1102	2
3	DN03	1103	3
4	DN04	1104	4
5	DN05	1105	5

Gambar 8 – Tampilan Table Daftar

➤ **SELECT * FROM BARANG**

Data Output						
	id_barang [PK] character varying (20)	harga integer	id_transaksi character varying (20)	jumlah_stok character varying (15)	merek_barang character (15)	id_daftar character varying (20)
1	1	150000	12502	11	GUCCI	X1
2	2	150000	12502	12	GUCCI	X2
3	3	300000	12502	16	ADIODAS	X3
4	4	40000	12502	19	ZARA	X4
5	5	50000	12502	25	NIKE	X5
6	6	30000	12502	35	ADIDAS	X6

Gambar 9 – Tampilan Table Barang

➤ **SELECT * FROM KIRIM**

Data Output				
	id_penjual character varying (20)	alamat character varying (20)	id_barang character varying (20)	id_pengiriman [PK] character varying (20)
1	DN01	JLMERDEKA	1	JE1101
2	DN02	JLPILAR	2	JE1102
3	DN03	JLKENANGA	3	JE1103
4	DN04	JLDANAUTOBA	4	JE1104
5	DN05	JLDEL	5	JE1105

Gambar 10 – Tampilan Table Kirim

4.4. Basic Query (Select, Update, Set Operators, Aggregate Function, Null Value)

Terhadap Tabel-Tabel Tersebut

Query :

```
--- 1. UPDATE ALAMAT DENGAN ID_PENJUAL = DNO1 MENJADI  
'JL_MAHAKAM'  
UPDATE PENJUAL  
SET ALAMAT = 'JL_MAHAKAM'  
WHERE ID_PENJUAL='DN01';  
SELECT * FROM PENJUAL
```

Output :

Data Output

	id_penjual [PK] character varying (20)	username character varying (20)	password character varying (20)	alamat character varying (20)	no_telepon character varying (20)	kode_pos character varying (20)	kota character varying (20)
1	DN02	MASTUARI	12S24	JL_PILAR	81260813831	2889	Psid
2	DN03	ANTO	12S34	JL_KENANGA	82165335651	2887	Medan
3	DN04	NIVERTITI	12S25	JL_DANAUTOBA	81370913100	2886	Ballige
4	DN05	NIVER	12S26	JL_DEL_KOTA	81262247137	2885	Tarutung
5	DN01	SUPRIANTO	12S14	JL_MAHAKAM	82273671252	2889	Pasid

Query :

```
---2. TAMPILKAN SELURUH DATA DARI TABEL PENJUAL DENGAN  
NAMA YANG BERAWALAN HURUF 'N'  
SELECT * FROM PENJUAL  
WHERE USERNAME LIKE 'N%'
```

Output :

Data Output

	id_penjual [PK] character varying (20)	username character varying (20)	password character varying (20)	alamat character varying (20)	no_telepon character varying (20)	kode_pos character varying (20)	kota character varying (20)
1	DN04	NIVERTITI	12S25	JL_DANAUTOBA	81370913100	2886	Ballige
2	DN05	NIVER	12S26	JL_DEL_KOTA	81262247137	2885	Tarutung

Query :

```
---3 TAMPILKAN HARGA TERTINGGI DAN TERENDAH DARI  
BARANG. SIMPAN DENGAN ATRIBUT DIFFERENCE  
SELECT MAX(HARGA) - MIN(HARGA) AS DIFFERENCE  
FROM BARANG;
```


Output :

Data Output		
	difference integer	
1	270000	

Query :

---4.MENAMPILKAN NILAI AWAL DAN AKHIR PADA TABEL BARANG---

```
SELECT ID_BARANG, HARGA "SALARY", ID_TRANSAKSI,
JUMLAH_STOK
FROM BARANG
WHERE HARGA BETWEEN 40000 AND 150000;
```

Output :

Data Output				
	id_barang [PK] character varying (20)	SALARY integer	id_transaksi character varying (20)	jumlah_stok character varying (15)
1	1	150000	12502	11
2	2	150000	12502	12
3	4	40000	12502	19
4	5	50000	12502	25

Query :

```
SELECT ID_BARANG, HARGA "SALARY",
ID_TRANSAKSI, JUMLAH_STOK, MEREK_BARANG, ID_DAFTAR
FROM BARANG
ORDER BY HARGA DESC;
```

Output :

Data Output						
	id_barang [PK] character varying (20)	SALARY integer	id_transaksi character varying (20)	jumlah_stok character varying (15)	merek_barang character (15)	id_daftar character varying (20)
1	3	300000	12502	16	ADIODAS	X3
2	1	150000	12502	11	GUCCI	X1
3	2	150000	12502	12	GUCCI	X2
4	5	50000	12502	25	NIKE	X5
5	4	40000	12502	19	ZARA	X4
6	6	30000	12502	35	ADIDAS	X6

Query :

```
SELECT ID_BARANG, HARGA, ID_TRANSAKSI, JUMLAH_STOK,
MEREK_BARANG, ID_BARANG
FROM BARANG
WHERE HARGA >= 150000
AND MEREK_BARANG LIKE 'G%';
```

Output :

Data Output						
	id_barang [PK] character varying (20)	harga integer	id_transaksi character varying (20)	jumlah_stok character varying (15)	merek_barang character (15)	id_barang character varying (20)
1	1	150000	12502	11	GUCCI	1
2	2	150000	12502	12	GUCCI	2

Query :

```
---7 TAMPILKAN SELURUH ATRIBUT DENGAN HARGA YANG
MENDAPATKAN KENAIKAN 5 KALI SIMPAN SEBAGAI SALARY--
SELECT ID_BARANG, 5*HARGA AS SALARY,
ID_TRANSAKSI, JUMLAH_STOK, MEREK_BARANG, ID_DAFTAR
FROM BARANG
```

Output :

Data Output						
	id_barang [PK] character varying (20)	salary integer	id_transaksi character varying (20)	jumlah_stok character varying (15)	merek_barang character (15)	id_daftar character varying (20)
1	1	750000	12502	11	GUCCI	X1
2	2	750000	12502	12	GUCCI	X2
3	3	1500000	12502	16	ADIODAS	X3
4	4	200000	12502	19	ZARA	X4
5	5	250000	12502	25	NIKE	X5
6	6	150000	12502	35	ADIDAS	X6

Query :

---8 MENAMPILKAN NILAI DALAM LIST---

```
SELECT ID_PENJUAL, ID_PENDAFTARAN, ID_BARANG
FROM DAFTAR
WHERE ID_PENDAFTARAN IN ('1101','1102','1105');
```

Output :

Data Output			
	id_penjual character varying (20)	id_pendaftaran character varying (20)	id_barang [PK] character varying (20)
1	DN01	1101	1
2	DN02	1102	2
3	DN05	1105	5

Query :

---9 URUTKAN DATA PADA TABEL BARANG BERDASARKAN HARGA TERENDAH HINGGA TERBESAR---

```
SELECT ID_BARANG, HARGA "SALARY",
ID_TRANSAKSI, JUMLAH_STOK, MEREK_BARANG, ID_DAFTAR
FROM BARANG
ORDER BY HARGA ASC;
```

Output :

Data Output						
	id_barang [PK] character varying (20)	SALARY integer	id_transaksi character varying (20)	jumlah_stok character varying (15)	merek_barang character (15)	id_daftar character varying (20)
1	6	30000	12502	35	ADIDAS	X6
2	4	40000	12502	19	ZARA	X4
3	5	50000	12502	25	NIKE	X5
4	1	150000	12502	11	GUCCI	X1
5	2	150000	12502	12	GUCCI	X2
6	3	300000	12502	16	ADIODAS	X3

4.5 Querying multiple tables, implementing views

1. Nested subqueries dan join table

Query :

---1. TAMPILKAN HARGA TERTINGGI, ID_BARANG, DAN MEREK BARANG DARI PRODUK YANG DIAMBIL DARI TABEL BARANG---

```
SELECT ID_BARANG, HARGA, MEREK_BARANG
```

```
FROM BARANG
```

```
WHERE HARGA = (SELECT MAX(HARGA) FROM BARANG);
```

Output :

Data Output			
	id_barang [PK] character varying (20)	harga integer	merek_barang character (15)
1	3	300000	ADIODAS

Query :

---2. TAMPILKAN SELURUH DATA TRANSAKSI YANG MEMILIKI JENIS TRANSAKSI 'BNI'

```
SELECT * FROM TRANSAKSI
```

```
JOIN BARANG
```

```
ON BARANG.MERK_BARANG = TRANSAKSI.ID_PEMBELI
```

```
WHERE TRANSAKSI.ID_PEMBELI = ('02');
```

Output :

Data Output							
id_barang character varying (20)	tgl_transaksi character varying (20)	id_transaksi character varying (20)	id_pembeli character varying (20)	jenis_transaksi character varying (15)	id_barang character varying (20)	harga integer	id_transaksi character varying (20)

Query :

---3. TAMPILKAN ID PENDAFTARAN DARI TABEL DAFTAR JOIN KE TABEL KIRIM-----

```
SELECT ID_PENDAFTARAN  
FROM DAFTAR INNER JOIN KIRIM  
ON DAFTAR.ID_PENJUAL = KIRIM.ID_PENJUAL;
```

Output :

Data Output	
	id_pendaftaran character varying (20)
1	1101
2	1102
3	1103
4	1104
5	1105

Query :

---4.---

```
SELECT MEREK_BARANG  
FROM BARANG  
WHERE ID_BARANG NOT IN('5');
```

Output :

Data Output	
	merek_barang character (15)
1	GUCCI
2	GUCCI
3	ADIDAS
4	ZARA
5	ADIDAS

Query :

---5---

```
SELECT ID_PENJUAL  
FROM PENJUAL CROSS JOIN PEMBELI
```

Output :

Data Output

	id_penjual [PK] character varying (20)
1	DN02
2	DN02
3	DN02
4	DN02
5	DN02
6	DN03
7	DN03
8	DN03
9	DN03
10	DN03
11	DN04

2. Membuat view dari setiap tabel

Query :

```
CREATE VIEW AUDITBARANG  
AS  
SELECT ID_BARANG, HARGA, MEREK_BARANG  
FROM BARANG  
WHERE HARGA = (SELECT MAX(HARGA) FROM BARANG);
```

Output :

```
SELECT * FROM AUDITBARANG
```

Data Output			
	id_barang character varying (20)	harga integer	merek_barang character (15)
1	3	300000	ADIODAS

Query :

---2---

CREATE VIEW PENDAFTARAN_BARANG

AS

SELECT ID_PENDAFTARAN

FROM DAFTAR INNER JOIN KIRIM

ON DAFTAR.ID_PENJUAL = KIRIM.ID_PENJUAL;

Output :

SELECT * FROM PENDAFTARAN_BARANG

Data Output

	id_pendaftaran character varying (20)
1	1101
2	1102
3	1103
4	1104
5	1105

Query :

CREATE VIEW LIST_MEREK

AS

SELECT MEREK_BARANG

FROM BARANG

```
WHERE ID_BARANG NOT IN('5');
```

Output :

```
SELECT * FROM LIST_MERЕК
```

Data Output

	merek_barang character (15)
1	GUCCI
2	GUCCI
3	ADIDAS
4	ZARA
5	ADIDAS

Query :

---4---

```
CREATE VIEW PENJUAL_PEMBELI
```

```
AS
```

```
SELECT ID_PENJUAL
```

```
FROM PENJUAL CROSS JOIN PEMBELI
```

Output :

```
SELECT * FROM PENJUAL_PEMBELI
```


Data Output

	id_penjual [PK] character varying (20)
1	DN02
2	DN02
3	DN02
4	DN02
5	DN02
6	DN03
7	DN03
8	DN03
9	DN03
10	DN03
11	DN04

Query :

---5---

CREATE VIEW PRODUKTERMURAH

AS

SELECT ID_BARANG, HARGA FROM BARANG

WHERE HARGA = (SELECT MIN(HARGA)FROM BARANG);

Output :

SELECT * FROM PRODUKTERMURAH

Data Output		
	id_barang character varying (20)	harga integer
1	6	30000

Query :

---6---

```
CREATE VIEW REVIEW_BARANG AS  
SELECT * FROM BARANG  
WHERE MEREK_BARANG IN ('NIKE','ADIDAS')
```

Output :

```
SELECT * FROM REVIEW_BARANG
```

Data Output

	id_barang character varying (20)	harga integer	id_transaksi character varying (20)	jumlah_stok character varying (15)	merk_barang character (15)	id_daftar character varying (20)
1	5	50000	12502	25	NIKE	X5
2	6	30000	12502	35	ADIDAS	X6

4.6 Function dan Stored Procedure

1. Funtion

Query :

---1.FUNCTION MENAMPILKAN DATA DOSEN---

```
CREATE FUNCTION DATA_PRODUK (PRODUK VARCHAR(20))  
RETURNS TABLE  
(ID_BARANG VARCHAR(20),HARGA INT)  
LANGUAGE 'plpgsql'  
AS  
$$  
BEGIN  
RETURN QUERY
```

```
(SELECT BARANG.ID_BARANG, BARANG.HARGA FROM BARANG
WHERE MEREK_BARANG = PRODUK);

END

$$
```

Output :

```
SELECT * FROM DATA_PRODUK ('ADIDAS');
```

Data Output

	id_barang character varying	harga integer
1	6	30000

✓ Successfully run. Total query runtime: 99 msec. 1 rows affected.

Query :

---2. MENAMPILKAN ID_PEMBELI DAN ALAMAT DARI TABLE PEMBELI---

```
CREATE FUNCTION UDF_PEMBELI (CUSTOMER VARCHAR(20))
```

```
RETURNS TABLE (ID_PEMBELI VARCHAR(20), ALAMAT  
VARCHAR(20))
```

```
LANGUAGE 'plpgsql'
```

```
AS
```

\$\$

BEGIN

RETURN QUERY

(SELECT PEMBELI.ID_PEMBELI, PEMBELI.ALAMAT FROM PEMBELI

WHERE PEMBELI.USERNAME = CUSTOMER);

END

\$\$

Output :

SELECT * FROM UDF_PEMBELI('SUPRI');

Data Output			
	id_pembeli character varying	alamat character varying	
1	12502	JL_PANDU	

✓ Successfully run. Total query runtime: 101 msec. 1 rows affected.

2. Stored Procedure

Query :

---1. MEMASUKKAN DATA BARU KE DATABASE---

```
CREATE PROCEDURE I_BARANG (I_ID_BARANG VARCHAR(20),  
I_HARGA INT, I_ID_TRANSAKSI VARCHAR(20), I_JUMLAH_STOK  
VARCHAR(20), I_MEREK_BARANG CHAR(15), I_ID_DAFTAR  
VARCHAR(20))
```

```
LANGUAGE 'plpgsql'
```

```
AS
```

```
$$
```

```
BEGIN
```

```
INSERT INTO BARANG (ID_BARANG, HARGA, ID_TRANSAKSI,  
JUMLAH_STOK, MEREK_BARANG)
```

```
VALUES (I_ID_BARANG, I_HARGA, I_ID_TRANSAKSI,  
I_JUMLAH_STOK, I_MEREK_BARANG);
```

```
COMMIT;
```

```
END;
```

```
$$
```

Output :

```
CALL I_BARANG ('07', '90000', '12502', '36', 'DELL', 'X7');
```

Messages

CALL

Query returned successfully in 116 msec.

SELECT * FROM BARANG;

Data Output

	id_barang [PK] character varying (20)	harga integer	id_transaksi character varying (20)	jumlah_stok character varying (15)	merek_barang character (15)	id_daftar character varying (20)
1	1	150000	12502	11	GUCCI	X1
2	2	150000	12502	12	GUCCI	X2
3	3	300000	12502	16	ADIDAS	X3
4	4	40000	12502	19	ZARA	X4
5	5	50000	12502	25	NIKE	X5
6	6	30000	12502	35	ADIDAS	X6
7	07	90000	12502	36	DELL	[null]

Query :

---2. MENGHAPUS MEREK BARANG---

```
CREATE PROCEDURE DELETE_MEREK_BARANG(I_ID_BARANG  
VARCHAR(20))
```

```
LANGUAGE 'plpgsql'
```

```
AS
```

```
$$
```

```
BEGIN
```

```
DELETE FROM BARANG
```

```
WHERE ID_BARANG = I_ID_BARANG;
```

COMMIT;

END;

\$\$

Output :

CALL DELETE_MERЕК_BARANG ('06');

Messages

CALL

Query returned successfully in 113 msec.

✓ Query returned successfully in 113 msec.

SELECT * FROM BARANG;

Data Output

	id_barang [PK] character varying (20)	harga integer	id_transaksi character varying (20)	jumlah_stok character varying (15)	merek_barang character (15)	id_daftar character varying (20)
1	1	150000	12502	11	GUCCI	X1
2	2	150000	12502	12	GUCCI	X2
3	3	300000	12502	16	ADIODAS	X3
4	4	40000	12502	19	ZARA	X4
5	5	50000	12502	25	NIKE	X5
6	6	30000	12502	35	ADIDAS	X6
7	07	90000	12502	36	DELL	[null]

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil analisis yang telah dilakukan, diketahui bahwa proses bisnis salah satu e-commerce tersebut dilakukan perancangan layanan sistem informasi dengan planning. Proses bisnis yang dibutuhkan telah teridentifikasi dan menghasilkan sebuah roadmap rencana implementasi untuk pembangunan aplikasi e-commerce tersebut. Dan kita akan terbantu dalam mengerti dan memahami pemanfaatan SQL query pada proses bisnis e-commerce. Berdasarkan laporan proyek tugas besar diatas dapat disimpulkan bahwa proyek basis data yaitu kegiatan berkaitan dengan pembelajaran di dunia pekerjaan, akan banyak digunakan pada data analysis. Penggunaan sql query tersebut kemudian diterapkan dalam kegiatan sesuai dengan penggunaan. Pembelajaran ini dilakukan, karena akan dipergunakan pada masa yang akan datang.

5.2. Saran

Penulis menyarankan beberapa hal yang berkaitan dengan proses pembelajaran, antara lain :

1. Untuk memahami pembelajaran dengan baik maka dapat mengimplementasikan eksistensi manajemen waktu belajar yang disiplin.
2. Penulis mengharapkan kritik dan saran yang membangun dalam penulisan makalah dikemudian hari
3. Penulis berharap agar di proyek selanjutnya seluruh anggota tim dapat lebih meningkatkan kerja sama .

DAFTAR PUSTAKA

<https://aantamim.id/normalisasi-database/>

<https://app.diagrams.net/>

<https://www.jojonomic.com/blog/normalisasi-database/>

https://www.w3schools.com/sql/sql_ref_add_constraint.asp

<https://profio.co.id/tutorial-membuat-function-sederhana-di-postgresql-dengan-pgadminiii/>

<https://ngodingdata.com/cara-membuat-stored-procedure-di-mysql/>