

LAPORAN UJIAN AKHIR SEMESTER (UAS)
SISTEM BASIS DATA
“SISTEM PENJUALAN DAN INVENTORI
BARANG PADA SUPERINDO”



**UNIVERSITAS
DUTA BANGSA
SURAKARTA**

Disusun oleh:

Nama:

- **Muhammad Ramadhan Fajri** (NIM:250119018)
- **Rafi Athallah Handokoputra** (NIM:250119020)

PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT LUNAK
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DUTA BANGSA SURAKARTA 2025

DAFTAR ISI

BAB 1 PENDAHULUAN	3
1.1 Latar Belakang dan Kasus.....	3
1.2 Tujuan dan Batasan Masalah	3
1.3 Deskripsi Umum Sistem	3
BAB 2 LANDASAN TEORI.....	4
2.1 Dasar Basis Data dan DBMS	4
2.2 ERD dan Relasi Tabel.....	4
2.3 Normalisasi Data.....	4
2.4 Entity Relationship Diagram (ERD)	4
2.5 Penjelasan Primary Key dan Foreign Key	4
BAB 3 PERANCANGAN DAN IMPLEMENTASI.....	5
3.1 Perancangan dan Struktur tabel (DDL).....	5
3.2 Implementasi DML (Pengisian Data)	5
3.3 Implementasi Transaksi Penjualan (TCL)	5
3.4 Pengujian Query SQL	5
BAB 4 PENUTUP.....	6
4.1 Hasil Pengujian dan Kendala Sistem	6
4.2 Kesimpulan dan Saran Pengembangan	6
LAMPIRAN	7
Daftar Pustaka	15

BAB 1

PENDAHULUAN

1.1 Latar Belakang dan Kasus

Superindo merupakan salah satu jenis usaha ritel atau supermarket yang menjual berbagai kebutuhan sehari-hari seperti makanan, minuman, dan barang konsumsi lainnya. Dalam operasional sehari-hari, proses pencatatan stok barang, data supplier, transaksi penjualan, serta aktivitas kasir harus dikelola dengan baik agar tidak terjadi kesalahan data.

Pada beberapa toko skala kecil hingga menengah, pencatatan data masih sering dilakukan secara manual atau terpisah-pisah, sehingga berpotensi menimbulkan masalah seperti ketidaksesuaian stok barang, kesalahan pencatatan transaksi, serta sulitnya melakukan rekap data penjualan. Oleh karena itu, dibutuhkan sebuah sistem basis data yang mampu mengelola data Superindo secara terstruktur, terintegrasi, dan mudah diakses.

1.2 Tujuan dan Batasan Masalah

Tujuan dari pembuatan sistem basis data Superindo ini adalah:

1. Mengelola data barang dan stok secara terpusat
2. Mengelola data kasir dan supplier
3. Mencatat transaksi penjualan secara detail dan terstruktur
4. Menjaga konsistensi data melalui penerapan transaksi database

Batasan masalah dalam sistem ini meliputi:

1. Pendataan kasir, supplier, dan barang
2. Pencatatan transaksi dan detail transaksi penjualan
3. Sistem tidak mencakup manajemen keuangan lanjutan atau laporan akuntansi

1.3 Deskripsi Umum Sistem

Sistem basis data Superindo ini dikembangkan menggunakan MySQL dan MySQL Workbench dengan pendekatan basis data relasional. Sistem terdiri dari beberapa tabel utama yaitu kasir, supplier, barang, transaksi, dan detail_transaksi. Seluruh tabel saling terhubung menggunakan primary key dan foreign key untuk menjaga integritas data

BAB 2

LANDASAN TEORI

2.1 Dasar Basis Data dan DBMS

Basis data adalah kumpulan data yang saling berhubungan dan tersimpan secara sistematis di dalam DBMS (Database Management System). MySQL digunakan sebagai DBMS karena bersifat relasional dan banyak digunakan dalam sistem informasi.

2.2 ERD dan Relasi Tabel

Entity Relationship Diagram (ERD) digunakan untuk memodelkan hubungan antar entitas seperti kasir, supplier, barang, dan transaksi. Relasi ini diwujudkan dalam bentuk Primary Key (PK) dan Foreign Key (FK).

2.3 Normalisasi Data

Normalisasi dilakukan hingga bentuk normal ketiga (3NF) untuk menghindari redundansi data dan menjaga konsistensi basis data.

- First Normal Form (1NF): Setiap atribut pada tabel hanya memiliki satu nilai (atomik) dan tidak ada atribut bernilai ganda.
- Second Normal Form (2NF): Setiap atribut non-primary key bergantung sepenuhnya pada primary key.
- Third Normal Form (3NF): Tidak terdapat ketergantungan transitif antar atribut non-primary key.

2.4 Entity Relationship Diagram (ERD)

ERD Sistem SUPERINDO terdiri dari lima entitas utama, yaitu Kasir, Supplier, Barang, Transaksi, dan Detail_Transaksi. Relasi antar entitas dirancang agar setiap transaksi penjualan dapat dilacak hingga ke data barang dan kasir yang bertugas.

Hubungan antar entitas:

- Supplier berelasi dengan Barang (one-to-many)
- Kasir berelasi dengan Transaksi (one-to-many)
- Transaksi berelasi dengan Detail_Transaksi (one-to-many)
- Barang berelasi dengan Detail_Transaksi (one-to-many)

2.5 Penjelasan Primary Key dan Foreign Key

- Kasir:PrimaryKey=id_kasir
- Supplier:PrimaryKey=id_supplier
- Barang:PrimaryKey=kode_barang, ForeignKey=id_supplier
- Transaksi:PrimaryKey=id_transaksi, ForeignKey=id_kasir
- Detail_Transaksi:PrimaryKey=id_detail,ForeignKey=id_transaksi & kode_barang

BAB 3

PERANCANGAN DAN IMPLEMENTASI

3.1 Perancangan dan Struktur tabel (DDL)

Implementasi basis data *SUPERINDO* diawali dengan pembuatan database dan tabel menggunakan Data Definition Language (DDL). Struktur tabel dirancang berdasarkan hasil perancangan ERD pada bab sebelumnya dengan menerapkan Primary Key (PK) dan Foreign Key (FK) untuk menjaga integritas data. Penerapan constraint seperti NOT NULL memastikan data penting tidak kosong, sedangkan foreign key digunakan untuk menjamin keterkaitan data antar tabel tetap konsisten

3.2 Implementasi DML (Pengisian Data)

Setelah struktur tabel dibuat, tahap selanjutnya adalah pengisian data awal menggunakan Data Manipulation Language (DML). Data yang dimasukkan meliputi data supplier, kasir, dan barang sebagai data awal sistem. Pengisian data ini berfungsi sebagai seed data agar sistem dapat diuji pada proses transaksi dan analisis query.

3.3 Implementasi Transaksi Penjualan (TCL)

Untuk menjaga konsistensi data pada proses penjualan, sistem menerapkan Transaction Control Language (TCL). Seluruh proses transaksi diperlakukan sebagai satu kesatuan proses. Proses transaksi dimulai dengan pencatatan data transaksi dan detail transaksi, kemudian diikuti dengan pengurangan stok barang. Apabila seluruh proses berhasil, perubahan data disimpan menggunakan COMMIT. Jika terjadi kesalahan, transaksi dapat dibatalkan menggunakan ROLLBACK sehingga konsistensi data tetap terjaga.

3.4 Pengujian Query SQL

- a) JOIN
digunakan untuk menggabungkan data dari tabel transaksi, detail transaksi, kasir, barang, dan supplier sehingga sistem dapat menampilkan informasi transaksi penjualan secara lengkap dan terstruktur
- b) Agregasi
digunakan untuk menghitung jumlah data barang serta total nilai harga barang yang tersimpan dalam basis data sebagai ringkasan data inventori.
- c) Group By & Having
digunakan untuk mengelompokkan data barang berdasarkan jenis barang dan menampilkan hanya kelompok data yang memenuhi kriteria tertentu, sehingga memudahkan pemantauan dan pengendalian stok.

BAB 4

PENUTUP

4.1 Hasil Pengujian dan Kendala Sistem

Pada tahap pengujian, sistem basis data SUPERINDO diuji menggunakan beberapa jenis query seperti JOIN, agregasi, GROUP BY, HAVING, serta transaksi database. Hasil pengujian menunjukkan bahwa relasi antar tabel telah berjalan dengan baik dan mampu menghasilkan informasi yang sesuai dengan kebutuhan sistem penjualan dan inventori. Kendala yang ditemui selama pengujian terutama berkaitan dengan konsistensi penginputan data, khususnya pada penggunaan primary key dan foreign key. Kesalahan input ID dapat menyebabkan data transaksi tidak terhubung dengan benar. Namun, kendala tersebut dapat diatasi dengan memastikan struktur tabel dan relasi telah didefinisikan secara tepat serta melakukan pengecekan data sebelum transaksi disimpan.

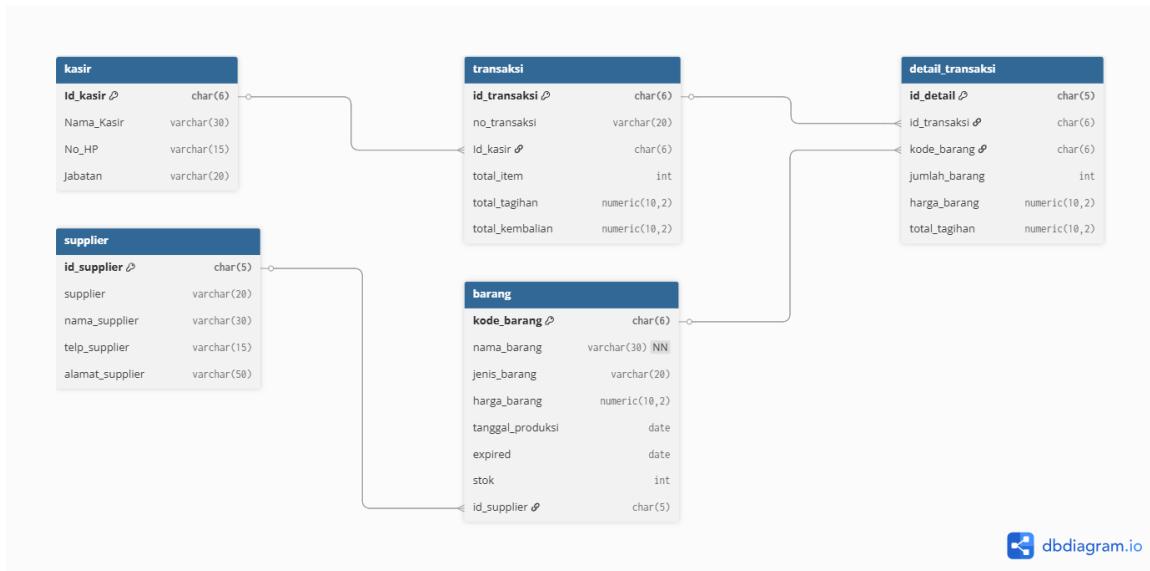
4.2 Kesimpulan dan Saran Pengembangan

Berdasarkan hasil perancangan dan pengujian, basis data SUPERINDO telah mampu mendukung proses pengelolaan data barang, supplier, kasir, serta transaksi penjualan secara terstruktur dan konsisten. Implementasi query SQL dan transaksi database memastikan data tersimpan dengan aman dan akurat.

Untuk pengembangan selanjutnya, sistem ini dapat ditingkatkan dengan menambahkan trigger otomatis untuk pengurangan dan peringatan stok serta pembuatan view laporan agar proses analisis penjualan dapat dilakukan dengan lebih efisien.

LAMPIRAN

- GitHub: <https://github.com/Rafiputra696/Project-UAS-SUPERINDO>
- Screenshot Skema Database



- Hasil Implementasi Query (Group By, having, join dll)

- GROUP BY (Untuk Mengelompokan data yang sejenis)

```
155    -- group by
156 •  select jenis_barang,count(kode_barang) as jumlah
157    from barang
158    group by jenis_barang;
159
```

Result Grid	
jenis_barang	jumlah
Makanan	4

- HAVING (mencari kategori barang yang memiliki jumlah lebih dari satu item.)

```
160 •  select jenis_barang,count(kode_barang) as jumlah
161    from barang
162    group by jenis_barang
163    having count(kode_barang) > 1;
```

Result Grid	
jenis_barang	jumlah
Makanan	4

- INNER JOIN (menggabungkan data dari empat tabel berbeda menjadi satu laporan transaksi yang lengkap.)

```

167     transaksi.no_transaksi,
168     kasir.nama_kasir,
169     barang.nama_barang,
170     detail_transaksi.jumlah_barang,
171     detail_transaksi.total_tagihan
172   from detail_transaksi
173   join transaksi
174     on detail_transaksi.id_transaksi = transaksi.id_transaksi
175   join kasir
176     on transaksi.id_kasir = kasir.id_kasir
177   join barang
178     on detail_transaksi.kode_barang=barang.kode_barang;
179

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

no_transaksi	nama_kasir	nama_barang	jumlah_barang	total_tagihan
167-19/10/2025	Santi	Nissin Sandwich Coklat	1	7900.00
168-19/10/2025	Santi	Wow Spaghetti	1	3190.00
168-19/10/2025	Santi	Emperial	1	5900.00
174-19/10/2025	Santi	Nissin Sandwich Coklat	1	7900.00

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

- RIGHT JOIN (menampilkan semua daftar supplier, terlepas dari apakah supplier tersebut memiliki barang yang terdaftar di tabel barang atau tidak.)

```

188    -- right join
189 •   select
190   barang.nama_barang,
191   supplier.nama_supplier
192   from barang
193   right join supplier
194   on barang.id_supplier = supplier.id_supplier;
195

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nama_barang	nama_supplier
Nissan Sandwich Coklat	JOKO SUSILO
Wow Spaghetti	JOKO SUSILO
Emperial	JOKO SUSILO
Aoki Coklat	SUPRATNO

Result 13 | Read Only

- LEFT JOIN (menampilkan semua daftar barang, terlepas dari apakah barang tersebut memiliki supplier atau tidak.)

```

180    -- left join
181 •   select
182   barang.nama_barang,
183   supplier.nama_supplier|
184   from barang
185   left join supplier
186   on barang.id_supplier = supplier.id_supplier;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nama_barang	nama_supplier
Nissin Sandwich Coklat	JOKO SUSILO
Wow Spaghetti	JOKO SUSILO
Emperial	JOKO SUSILO
Aoki Coklat	SUPRATNO

Result 12 | Read Only

- FULL OUTER JOIN (Query ini menampilkan semua data supplier dan barang pasokannya, serta tetap memunculkan nama supplier meskipun belum memiliki barang yang terdaftar.)

```

157 • select
158   barang.nama_barang,
159   supplier.nama_supplier
160   from barang
161   left join supplier
162   on barang.id_supplier = supplier.id_supplier
163   union
164   select
165   barang.nama_barang,
166   supplier.nama_supplier
167   from barang
168   right join supplier
169   on barang.id_supplier = supplier.id_supplier;

```

Result Grid | Filter Rows: Export: Wrap Cell Content: □

nama_barang	nama_supplier
Nisari Sandwich Coklat	JOKO SUSILO
Wow Spaghetti	JOKO SUSILO
Emperial	JOKO SUSILO
Aoki Coklat	SUPRATNO

Result Grid

- Script SQL Lengkap (Superindo)

```

CREATE database SUPERINDO;
use superindo;

-- DDL
create table kasir (
Id_kasir char (6),
Nama_Kasir varchar (30),
No_HP varchar (15),
Jabatan varchar (20),
primary key (Id_kasir)
);

create table supplier (
id_supplier char (5),
supplier varchar (20),
nama_supplier varchar (30),
telp_supplier varchar (15),
alamat_supplier varchar (50),
primary key (id_supplier)
);

create table barang (
kode_barang char (6),
nama_barang varchar (30) not null,
jenis_barang varchar (20),
harga_barang numeric (10,2),
tanggal_produksi date,
expired date,
stok int,
id_supplier char (5),
primary key (kode_barang),
foreign key (id_supplier) references supplier(id_supplier)
);

```

```

create table transaksi (
id_transaksi char (6),
no_transaksi varchar (20),
Id_kasir char (6),
total_item int,
total_tagihan numeric (10,2),
total_kembalian numeric (10,2),
primary key (id_transaksi),
foreign key (Id_kasir) references kasir(Id_kasir)
);

create table detail_transaksi (
id_detail char (5),
id_transaksi char (6),
kode_barang char (6),
jumlah_barang int,
harga_barang numeric (10,2),
total_tagihan numeric (10,2),
primary key (id_detail),
foreign key (id_transaksi) references transaksi(id_transaksi),
foreign key (kode_barang) references barang(kode_barang)
);

show tables;
describe barang;
describe supplier;
describe kasir;
describe transaksi;
describe detail_transaksi;

-- DML
insert into supplier values
('SP001','INDOFD','JOKO SUSILO','8956666666','JL.MANGGIS'),
('SP002','GARUDAFD','SUPRATNO','8977777777','JL.KACANG');

select * from supplier;

insert into kasir values
('KSR001','Santi','8999999999','Junior'),
('KSR002','Yani','8977777777','Senior');

select * from kasir;

insert into barang values
('BR001','Nissin Sandwich Coklat','Makanan',7900,'2024-10-15','2026-10-15',1200,'SP001'),
('BR002','Wow Spaghetti','Makanan',3190,'2025-05-20','2025-11-20',500,'SP001'),
('BR003','Emperial','Makanan',5900,'2025-06-11','2025-11-11',750,'SP001'),

```

```
('BR004','Aoki Coklat','Makanan',2500,'2025-11-15','2026-05-21',1000,'SP002');

select * from barang;

insert into transaksi values
('TR001','167-19/10/2025','KSR001',1,7900,12100),
('TR002','168-19/10/2025','KSR001',1,3190,16810),
('TR003','169-19/10/2025','KSR002',1,5900,14100);

select * from transaksi;

insert into detail_transaksi values
('DT001','TR001','BR001',1,7900,7900),
('DT002','TR002','BR002',1,3190,3190),
('DT003','TR002','BR003',1,5900,5900);

select * from detail_transaksi;

update barang
set stok = 1000
where kode_barang = 'BR001';

select * from barang;

select stok from barang
where kode_barang = 'BR001' ;

select nama_barang,harga_barang
from barang
where harga_barang > 5000;

select nama_barang,stok
from barang
where stok > 500;

-- where
select * from barang
where jenis_barang='makanan';

-- distinct
select distinct jenis_barang
from barang;

-- order by
select nama_barang,harga_barang
from barang
order by harga_barang asc;

-- limit
select * from barang
```

```

limit 2;

-- like
select * from barang
where nama_barang like 'Emperial';

-- between
select * from barang
where harga_barang between 3000 and 8000;

-- in
select * from barang
where kode_barang in ('BR001','BR003');

-- alias
select nama_barang as nama,harga_barang as harga from barang;

-- agregat
select count(kode_barang) as jumlah_barang
from barang;
select sum(harga_barang) as total_harga
from barang;

-- group by
select jenis_barang,count(kode_barang) as jumlah
from barang
group by jenis_barang;

select jenis_barang,count(kode_barang) as jumlah
from barang
group by jenis_barang
having count(kode_barang) > 1;

-- inner joint
select
transaksi.no_transaksi,
kasir.nama_kasir,
barang.nama_barang,
detail_transaksi.jumlah_barang,
detail_transaksi.total_tagihan
from detail_transaksi
join transaksi
    on detail_transaksi.id_transaksi = transaksi.id_transaksi
join kasir
    on transaksi.id_kasir = kasir.id_kasir
join barang
    on detail_transaksi.kode_barang=barang.kode_barang;

-- left join
select

```

```
barang.nama_barang,
supplier.nama_supplier
from barang
left join supplier
on barang.id_supplier = supplier.id_supplier;

-- right join
select
barang.nama_barang,
supplier.nama_supplier
from barang
right join supplier
on barang.id_supplier = supplier.id_supplier;

-- full outer join
select
barang.nama_barang,
supplier.nama_supplier
from barang
left join supplier
on barang.id_supplier = supplier.id_supplier
union
select
barang.nama_barang,
supplier.nama_supplier
from barang
right join supplier
on barang.id_supplier = supplier.id_supplier;

-- TCL : Transaksi Penjualan
start transaction;

insert into transaksi values
('TR008','174-19/10/2025','KSR001',1,7900,12100);

insert into detail_transaksi values
('DT008','TR008','BR001',1,7900,7900);

update barang
set stok = stok -1
where kode_barang = 'BR001';

commit;

-- sub query
select nama_barang, harga_barang
from barang
where harga_barang > (
select avg(harga_barang) from barang
);
```

```
-- view
create view view_transaksi as
select
transaksi.no_transaksi,
kasir.nama_kasir,
barang.nama_barang,
detail_transaksi.jumlah_barang,
detail_transaksi.total_tagihan
from detail_transaksi
join transaksi on detail_transaksi.id_transaksi = transaksi.id_transaksi
join kasir on transaksi.id_kasir = kasir.id_kasir
join barang on detail_transaksi.kode_barang = barang.kode_barang;
```

Daftar Pustaka

- Kadir, Abdul. (2014). Pengenalan Sistem Informasi. Yogyakarta: Andi Offset.
- Sutanta, Edhy. (2011). Basis Data dalam Tinjauan Konseptual. Yogyakarta: Andi Offset.
- Connolly, T., & Begg, C. (2015). Database Systems: A Practical Approach to Design, Implementation, and Management (6th ed.). Boston: Pearson Education.