

# **LAPORAN (UAS) SISTEM BASIS DATA**

## **“Implementasi dan Pengujian Query SQL pada Sistem Basis Data Penjualan Pet Shop”**



**UNIVERSITAS  
DUTA BANGSA  
SURAKARTA**

**Disusun oleh:**

**Nama:**

- **Hafidh Ridlo Nur Akbar, (NIM:250119008)**
- **Koujiro Inotsume (NIM:250119014)**

**PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT LUNAK  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS DUTA BANGSA SURAKARTA 2026**

## **DAFTAR ISI**

<b>BAB 1 PENDAHULUAN .....</b>	<b>3</b>
<b>BAB 2 LANDASAN TEORI.....</b>	<b>4</b>
<b>BAB 3 PERANCANGAN DAN IMPLEMENTASI .....</b>	<b>5</b>
<b>BAB 4 PENUTUP .....</b>	<b>6</b>
<b>LAMPIRAN.....</b>	<b>7</b>
<b>DAFTAR PUSTAKA.....</b>	<b>10</b>

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang dan Studi Kasus

Perkembangan teknologi informasi mendorong berbagai bidang usaha untuk beralih dari sistem manual menuju sistem terkomputerisasi, termasuk dalam bidang usaha penjualan dan pelayanan hewan peliharaan. Pet Shop & Care merupakan usaha yang menyediakan berbagai produk seperti makanan hewan, aksesoris, serta layanan perawatan hewan. Namun, dalam praktiknya, pencatatan transaksi, stok barang, dan data kasir sering kali masih dilakukan secara manual sehingga berpotensi menimbulkan kesalahan pencatatan, keterlambatan laporan, dan sulitnya pengambilan keputusan.

Berdasarkan permasalahan tersebut, diperlukan sebuah sistem basis data yang mampu mengelola data penjualan, stok barang, supplier, serta transaksi secara terstruktur dan terintegrasi. Oleh karena itu, pada project ini dibangun sebuah sistem basis data penjualan Pet Shop & Care menggunakan MySQL dengan penerapan berbagai query SQL sesuai dengan materi Pemrograman Basis Data.

### 1.2 Tujuan dan Batasan Masalah

Tujuan dari pembuatan sistem basis data ini adalah:

1. Mengimplementasikan query SQL sesuai dengan materi jurnal kuliah.
2. Mengelola data barang, supplier, kasir, dan transaksi penjualan secara terstruktur.
3. Menghasilkan laporan transaksi yang akurat melalui query JOIN dan agregasi.

Batasan masalah pada project ini meliputi:

- Pengelolaan data barang dan stok.
- Pencatatan transaksi penjualan.
- Penggunaan query DDL, DML, DQL, JOIN, dan TCL.
- Sistem belum terintegrasi dengan aplikasi berbasis web atau mobile.

### 1.3 Deskripsi Umum Sistem

Sistem basis data Pet Shop & Care dirancang menggunakan pendekatan relational database. Data dibagi ke dalam beberapa tabel utama seperti barang, supplier, kasir, transaksi, dan detail\_transaksi. Relasi antar tabel dibangun menggunakan Primary Key (PK) dan Foreign Key (FK) sehingga integritas data tetap terjaga. Sistem ini memungkinkan setiap transaksi penjualan tercatat secara detail dan stok barang dapat diperbarui secara otomatis.

## BAB 2

### LANDASAN TEORI

#### 2.1 Basis Data dan DBMS

Basis data adalah kumpulan data yang saling berhubungan dan disimpan secara sistematis di dalam komputer sehingga dapat diakses dan dikelola dengan mudah. DBMS (Database Management System) merupakan perangkat lunak yang digunakan untuk mengelola basis data, salah satunya adalah MySQL yang digunakan dalam project ini.

#### 2.2 Entity Relationship Diagram (ERD)

ERD digunakan untuk memodelkan hubungan antar entitas dalam sistem. Pada sistem Pet Shop & Care, entitas utama meliputi Barang, Supplier, Kasir, Transaksi, dan Detail Transaksi. Setiap entitas memiliki atribut yang merepresentasikan data yang dibutuhkan dalam proses bisnis.

#### 2.3 Normalisasi Data

Proses normalisasi dilakukan hingga bentuk normal ketiga (3NF) dengan tujuan:

- Menghindari duplikasi data.
- Menjaga konsistensi data.
- Mempermudah pengolahan data dalam query SQL.

#### 2.4 Konsep Primary Key dan Foreign Key

Primary Key digunakan sebagai identitas unik pada setiap tabel, sedangkan Foreign Key berfungsi untuk menghubungkan satu tabel dengan tabel lainnya. Dengan adanya PK dan FK, proses JOIN antar tabel dapat dilakukan dengan baik dan akurat.

## BAB 3

### PERANCANGAN DAN IMPLEMENTASI

#### 3.1 Perancangan Struktur Tabel (DDL)

Perancangan basis data dilakukan menggunakan perintah Data Definition Language (DDL) seperti CREATE TABLE, ALTER TABLE, dan DROP COLUMN. Tabel-tabel utama yang dibuat antara lain tabel barang, supplier, kasir, transaksi, dan detail\_transaksi. Setiap tabel dirancang dengan tipe data yang sesuai serta constraint untuk menjaga validitas data.

#### 3.2 Implementasi Data dan Transaksi (DML & TCL)

Data awal dimasukkan menggunakan perintah Data Manipulation Language (DML) seperti INSERT, UPDATE, dan DELETE. Selain itu, Transaction Control Language (TCL) diterapkan menggunakan perintah START TRANSACTION, COMMIT, dan ROLLBACK untuk memastikan proses transaksi berjalan dengan aman dan konsisten, khususnya saat pengurangan stok barang.

#### 3.3 Pengujian Query dan Analisis Data

Untuk memastikan sistem berjalan sesuai kebutuhan, dilakukan pengujian beberapa query, antara lain:

- JOIN untuk menampilkan detail transaksi penjualan secara lengkap.
- GROUP BY dan Agregasi untuk menghitung total penjualan.
- HAVING untuk menampilkan barang dengan stok rendah.
- SUBQUERY untuk analisis harga di atas rata-rata.
- VIEW untuk mempermudah pembuatan laporan transaksi.

Hasil pengujian menunjukkan bahwa query yang digunakan mampu menghasilkan informasi yang akurat dan relevan dengan kebutuhan sistem.

## **BAB 4**

### **PENUTUP**

#### **4.1 Kesimpulan**

Berdasarkan hasil perancangan dan implementasi, dapat disimpulkan bahwa sistem basis data penjualan Pet Shop & Care berhasil dibangun dengan baik menggunakan query SQL sesuai materi Pemrograman Basis Data. Sistem ini mampu mengelola data barang, transaksi, dan laporan penjualan secara terstruktur, akurat, dan efisien.

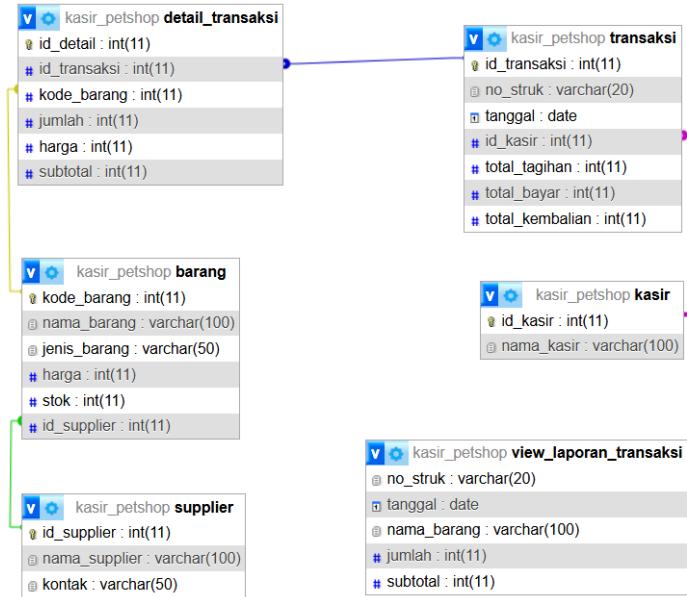
#### **4.2 Saran Pengembangan**

Untuk pengembangan selanjutnya, sistem ini dapat ditingkatkan dengan:

- Penambahan trigger untuk pengurangan stok otomatis.
- Pengembangan aplikasi berbasis web atau mobile.
- Pembuatan laporan penjualan periodik yang lebih detail.

## Lampiran

### Screenshot Skema Database



### Hasil Eksekusi Query (JOIN, GROUP BY, HAVING)

#### ○ GROUP BY

```
74 •   SELECT t.no_struk, SUM(dt.subtotal) AS total_penjualan
75     FROM transaksi t
76     JOIN detail_transaksi dt ON t.id_transaksi = dt.id_transaksi
77     GROUP BY t.no_struk;
```

Result Grid	
no_struk	total_penjualan
TRX001	62000

#### ○ HAVING

```
79 •   SELECT t.no_struk, SUM(dt.subtotal) AS total_penjualan
80     FROM transaksi t
81     JOIN detail_transaksi dt ON t.id_transaksi = dt.id_transaksi
82     GROUP BY t.no_struk
83     HAVING SUM(dt.subtotal) > 50000;
```

Result Grid	
no_struk	total_penjualan
TRX001	62000

o JOIN

```
68 •  SELECT t.no_struk, b.nama_barang, dt.jumlah, dt.subtotal  
69   FROM transaksi t  
70   JOIN detail_transaksi dt ON t.id_transaksi = dt.id_transaksi  
71   JOIN barang b ON dt.kode_barang = b.kode_barang;  
72
```

Result Grid			
no_struk	nama_barang	jumlah	subtotal
TRX001	Ori Cat Pouch	2	40000
TRX001	Cat Grooming	1	22000

## Script SQL

```
create database kasir_petshop;  
use kasir_petshop;  
  
-- DDL  
CREATE TABLE kasir (  
    id_kasir INT AUTO_INCREMENT PRIMARY KEY,  
    nama_kasir VARCHAR(100) NOT NULL  
);  
  
CREATE TABLE supplier (  
    id_supplier INT AUTO_INCREMENT PRIMARY KEY,  
    nama_supplier VARCHAR(100),  
    kontak VARCHAR(50)  
);  
  
CREATE TABLE barang (  
    kode_barang INT AUTO_INCREMENT PRIMARY KEY,  
    nama_barang VARCHAR(100),  
    jenis_barang VARCHAR(50),  
    harga INT,  
    stok INT,  
    id_supplier INT,  
    FOREIGN KEY (id_supplier) REFERENCES supplier(id_supplier)  
);  
  
CREATE TABLE transaksi (  
    id_transaksi INT AUTO_INCREMENT PRIMARY KEY,  
    no_struk VARCHAR(20),  
    tanggal DATE,  
    id_kasir INT,  
    total_tagihan INT,  
    total_bayar INT,  
    total_kembalian INT,  
    FOREIGN KEY (id_kasir) REFERENCES kasir(id_kasir)  
);  
  
CREATE TABLE detail_transaksi (  
    id_detail INT AUTO_INCREMENT PRIMARY KEY,  
    id_transaksi INT,  
    kode_barang INT,  
    jumlah INT,  
    harga INT,  
    subtotal INT,  
    FOREIGN KEY (id_transaksi) REFERENCES transaksi(id_transaksi),  
    FOREIGN KEY (kode_barang) REFERENCES barang(kode_barang)  
);
```

```

-- DML
INSERT INTO kasir (nama_kasir)
VALUES ('Andi');

INSERT INTO supplier (nama_supplier, kontak)
VALUES ('PT Pet Food', '08123456789');

INSERT INTO barang (nama_barang, jenis_barang, harga, stok, id_supplier)
VALUES
('Ori Cat Pouch', 'Makanan', 20000, 50, 1),
('Cat Grooming', 'Jasa', 22000, 20, 1);

INSERT INTO transaksi (no_struk, tanggal, id_kasir, total_tagihan, total_bayar, total_kembalian)
VALUES ('TRX001', '2026-01-10', 1, 62000, 100000, 38000);

INSERT INTO detail_transaksi (id_transaksi, kode_barang, jumlah, harga, subtotal)
VALUES
(1, 1, 2, 20000, 40000),
(1, 2, 1, 22000, 22000);

-- DQL
SELECT t.no_struk, b.nama_barang, dt.jumlah, dt.subtotal
FROM transaksi t
JOIN detail_transaksi dt ON t.id_transaksi = dt.id_transaksi
JOIN barang b ON dt.kode_barang = b.kode_barang;

SELECT t.no_struk, SUM(dt.subtotal) AS total_penjualan
FROM transaksi t
JOIN detail_transaksi dt ON t.id_transaksi = dt.id_transaksi
GROUP BY t.no_struk;

SELECT t.no_struk, SUM(dt.subtotal) AS total_penjualan
FROM transaksi t
JOIN detail_transaksi dt ON t.id_transaksi = dt.id_transaksi
GROUP BY t.no_struk
HAVING SUM(dt.subtotal) > 50000;

SELECT nama_barang, harga
FROM barang
WHERE harga > (
    SELECT AVG(harga) FROM barang
);

CREATE VIEW view_laporan_transaksi AS
SELECT t.no_struk, t.tanggal, b.nama_barang, dt.jumlah, dt.subtotal
FROM transaksi t
JOIN detail_transaksi dt ON t.id_transaksi = dt.id_transaksi
JOIN barang b ON dt.kode_barang = b.kode_barang;

START TRANSACTION;

UPDATE barang
SET stok = stok - 2
WHERE kode_barang = 1;

COMMIT;

```

## **DAFTAR PUSTAKA**

Connolly, T., & Begg, C. *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson Education, 2014.

Materi Jurnal Kuliah Pemrograman Basis Data. Universitas Duta Bangsa  
Surakarta, 2025.