

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

## **”SISTEM PENJUALAN, TRANSAKSI BARANG DAN JADWAL KARYAWAN VAPE STORE”**



**Disusun oleh:**

**Nama:**

- **Iqbal Hadi Widyadana, (NIM:250119011)**
- **Fauzan Fathoni Khoirul Huda (NIM:250119007)**

**PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT LUNAK**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS DUTA BANGSA SURAKARTA 2026**

## **DAFTAR ISI**

<b>BAB 1.....</b>	<b>3</b>
<b>PENDAHULUAN.....</b>	<b>3</b>
<b>BAB 2.....</b>	<b>4</b>
<b>LANDASAN TEORI.....</b>	<b>4</b>
<b>BAB 3.....</b>	<b>5</b>
<b>PERANCANGAN DAN IMPLEMENTASI .....</b>	<b>5</b>
<b>BAB 4.....</b>	<b>6</b>
<b>PENUTUP .....</b>	<b>6</b>
<b>LAMPIRAN.....</b>	<b>7</b>
<b>DAFTAR PUSTAKA.....</b>	<b>12</b>

# **BAB 1**

## **PENDAHULUAN**

### *1.1 Latar Belakang dan Kasus*

Studi kasus yang kami angkat adalah operasional pada "Vape Store", sebuah toko yang menjual perlengkapan seperti liquid, coil, cartridge, dan perangkat pod. Selama ini, operasional di toko Vape Store tersebut masih mengandalkan nota manual dan buku besar untuk mencatat stok barang. Kami mengidentifikasi beberapa masalah utama, yaitu sering terjadinya selisih stok karena data yang tidak ter-update secara otomatis, perhitungan total belanja yang memakan waktu lama, serta jadwal kerja karyawan yang tidak terdata dengan rapi. Kondisi ini membuat pemilik atau pelaku usaha sulit memantau performa harian secara akurat dan terpercaya.

### *1.2 Tujuan dan Batasan Masalah*

Project ini bertujuan untuk membangun sistem basis data yang bisa mengotomatisasi seluruh alur kerja toko Vape Store. Batasan sistem ini mencakup:

1. Pendataan barang (stok dan harga).
2. Manajemen akun karyawan yang bertugas sebagai kasir/operator.
3. Pencatatan transaksi penjualan secara detail per item (nota digital).
4. Pengaturan jadwal kerja shift agar kehadiran karyawan lebih terpantau.

### *1.3 Deskripsi Umum Sistem*

Sistem ini dikembangkan menggunakan MySQL Workbench dengan pendekatan relasi. Sistem membagi data ke dalam dua modul besar: Modul Penjualan dan Modul Kepegawaian. Keduanya saling terikat melalui relasi ID Karyawan dan ID Kasir. Dengan sistem ini, setiap penjualan yang tercatat akan langsung memotong stok barang dan merekam siapa operator yang bertanggung jawab pada jam transaksi tersebut.

## BAB 2

### LANDASAN TEORI

#### 2.1 Dasar Basis Data dan ERD

Dalam merancang sistem ini, kami menggunakan konsep RDBMS (Relational Database Management System). Langkah pertama dimulai dengan membuat ERD (*Entity Relationship Diagram*) untuk memetakan bagaimana entitas seperti Karyawan, Barang, dan Transaksi saling berinteraksi. Kami memastikan setiap entitas memiliki identitas unik agar data tidak tumpang tindih saat dipanggil oleh sistem aplikasi nantinya.

#### 2.2 Proses Normalisasi Data

Kami melakukan normalisasi hingga tahap 3NF (Third Normal Form) untuk memastikan data bersih dari duplikasi:

1. **1NF:** Kami memastikan setiap kolom dalam tabel hanya berisi satu nilai tunggal.
2. **2NF:** Kami memisahkan data yang tidak bergantung langsung pada Primary Key (PK). Misalnya, data barang dipisahkan dari tabel transaksi agar satu barang bisa dibeli berkali-kali tanpa harus menulis ulang nama barangnya.
3. **3NF:** Kami memisahkan data login (Operator) dari data biodata (Karyawan). Hal ini dilakukan agar sistem lebih aman dan struktur datanya tidak saling memberatkan.

#### 2.3 Desain Relasi (PK & FK)

Hasil desain kami menggunakan skema kunci yang ketat:

- **Primary Key (PK):** Seperti `id_items` dan `id_transaksi`, yang menjadi kunci unik di setiap tabel.
- **Foreign Key (FK):** Seperti `id_karyawan` di tabel `jadwal` dan `id_items` di tabel `transaksi_detail`. FK inilah yang memungkinkan kita memanggil nama barang yang dibeli hanya melalui ID-nya saja saat proses JOIN dilakukan.

## BAB 3

### PERANCANGAN DAN IMPLEMENTASI

#### 3.1 Struktur Tabel (DDL)

Kami mengimplementasikan desain ke dalam database `vape_storeV1` menggunakan perintah DDL (*Data Definition Language*). Kami menambahkan aturan *Constraint* seperti `NOT NULL` agar field penting tidak boleh kosong. Selain itu, kami menggunakan `ON DELETE CASCADE` pada tabel operator; tujuannya agar jika seorang karyawan berhenti dan datanya dihapus, maka akun operatornya otomatis ikut hilang demi keamanan system yang berjalan.

#### 3.2 Input Data dan Logika Transaksi (DML & TCL)

Data awal (seed data) kami masukkan melalui perintah DML (*Data Manipulation Language*), mencakup daftar barang populer (Liquid, Coil, Cartridge) dan biodata karyawan. Hal terpenting yang kami terapkan adalah **TCL** (*Transaction Control Language*). Kami mensimulasikan proses penjualan menggunakan `START TRANSACTION`. Logikanya: saat kasir menginput data belanja, stok barang akan langsung berkurang secara otomatis. Jika di tengah proses terjadi kegagalan sistem, perintah `ROLLBACK` akan mengembalikan stok ke jumlah semula, sehingga data tetap akurat.

#### 3.3 Analisis Data melalui Query

Kami menguji kekuatan database ini melalui beberapa query kompleks:

1. **JOIN:** Kami menggabungkan empat tabel (Transaksi, Kasir, Detail, dan Barang) untuk menghasilkan laporan nota yang lengkap bagi pembeli.
2. **Agregasi & GROUP BY:** Kami menghitung total omzet yang dihasilkan oleh masing-masing kasir untuk melihat siapa yang paling produktif.
3. **HAVING:** Query ini kami gunakan khusus untuk memantau stok. Sistem akan menampilkan hanya barang-barang yang stoknya sudah di bawah 20 unit agar segera dilakukan pengadaan kembali.

## **BAB 4**

### **PENUTUP**

#### *4.1 Evaluasi dan Kendala Lapangan*

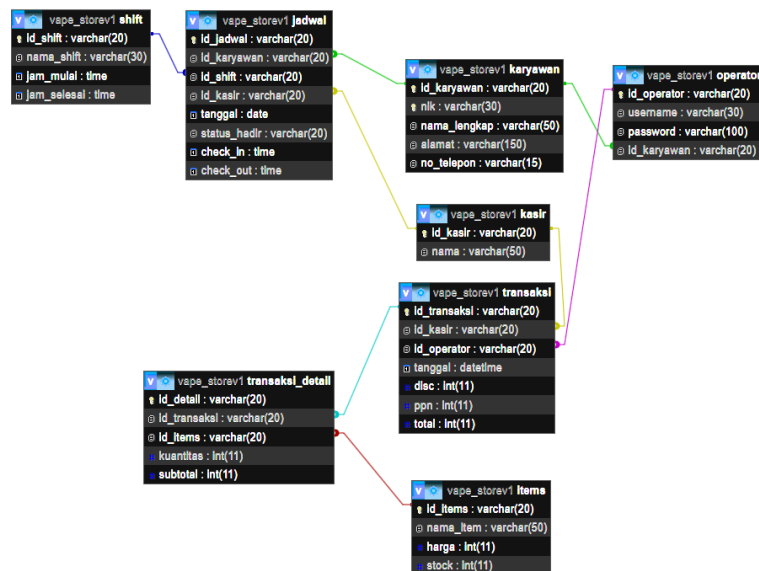
Selama pengerjaan, tantangan terbesar kami adalah menyinkronkan data jadwal karyawan dengan waktu transaksi. Kami menyadari bahwa integritas data sangat bergantung pada ketelitian penginputan ID. Namun, setelah dilakukan pengetesan, relasi antar tabel sudah berjalan stabil dan query yang dijalankan memberikan hasil yang akurat sesuai data yang ada.

#### *4.2 Kesimpulan dan Pengembangan*

Sistem basis data Vape Store ini sudah siap menggantikan sistem manual. Database ini mampu mengelola inventaris, transaksi, hingga manajemen SDM (Sumber Daya Manusia) dengan rapi. Untuk ke depannya, kami menyarankan penambahan fitur *Trigger* untuk peringatan stok otomatis dan pembuatan modul laporan laba-rugi bulanan yang lebih mendalam.

## LAMPIRAN

- GitHub: [https://github.com/iqballrawrr/DONE-PROJECT-UAS\\_SBD\\_VapeStore.git](https://github.com/iqballrawrr/DONE-PROJECT-UAS_SBD_VapeStore.git)
- Screenshot Skema Database



- Hasil Eksekusi Query (JOIN, GROUP BY, HAVING dll)
  - ORDER BY: Mengurutkan item berdasarkan harga termahal

```

141 -- 1. ORDER BY: Mengurutkan item berdasarkan harga termahal (Hal. 88-82)
142 • SELECT * FROM items
143 ORDER BY harga DESC;

```

Result Grid

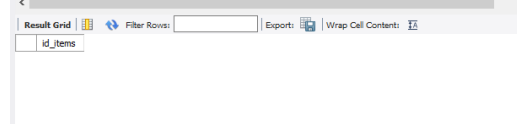
id_items	nama_item	harga	stock
BRG001	Liquid Oatberry 60ml	150000	15
BRG002	Col Allen Prebuild	45000	50
BRG003	Cartridge Ursa V2	40000	30

- **DISTINCT:** Menampilkan id\_items yang unik dari tabel transaksi\_detail

```

145 -- 2. DISTINCT: Menampilkan id_items yang unik dari tabel transaksi_detail
146 • SELECT DISTINCT id_items FROM transaksi_detail;

```



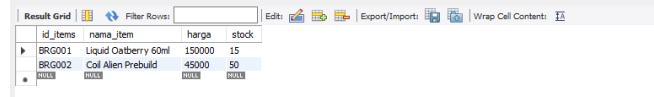
id_items
BRG001

- **OPERATOR LOGIKA (AND, OR, NOT):** Mencari item dengan harga di atas 40rb DAN stok di atas 10

```

148 -- 3. OPERATOR LOGIKA (AND, OR, NOT): Mencari item dengan harga di atas 40rb DAN stok di atas 10
149 • SELECT * FROM Items
150 WHERE harga > 40000 AND stock > 10;
151

```



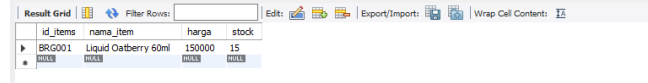
id_items	nama_item	harga	stock
BRG001	Liquid Oatberry 60ml	150000	15
BRG002	Col Allen Prebuild	45000	50

- **LIKE:** Mencari item yang mengandung kata 'Liquid'

```

151
152 -- 4. LIKE: Mencari item yang mengandung kata 'Liquid'
153 • SELECT * FROM Items
154 WHERE nama_item LIKE '%Liquid%';

```



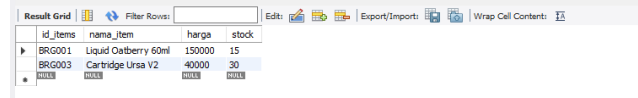
id_items	nama_item	harga	stock
BRG001	Liquid Oatberry 60ml	150000	15

- **BETWEEN:** Mencari item dengan stok antara 10 sampai 40

```

155
156 -- 5. BETWEEN: Mencari item dengan stok antara 10 sampai 40
157 • SELECT * FROM Items
158 WHERE stock BETWEEN 10 AND 40;

```



id_items	nama_item	harga	stock
BRG001	Liquid Oatberry 60ml	150000	15
BRG003	Cartridge Ursa V2	40000	30

- **CALCULATION & ALIAS:** Menghitung estimasi pajak 10% per item

```

160 -- 6. CALCULATION & ALIAS: Menghitung estimasi pajak 10% per item
161 • SELECT
162     nama_item,
163     harga AS harga_asli,
164     (0.1 * harga) AS estimasi_ppn,
165     (harga + (0.1 * harga)) AS harga_total_pajak
166 FROM items;

```

nama_item	harga_asli	estimasi_ppn	harga_total_pajak
Liquid Oatberry 60ml	150000	15000.0	165000.0
Col Allen Prebuild	45000	4500.0	49500.0
Cartridge Ursa V2	40000	4000.0	44000.0

- **FUNGSI AGREGAT LAINNYA (MIN, MAX, AVG):** Statistik harga barang

```

168 -- 7. FUNGSI AGREGAT LAINNYA (MIN, MAX, AVG): Statistik harga barang
169 • SELECT
170     MIN(harga) AS harga_termurah,
171     MAX(harga) AS harga_termahal,
172     AVG(harga) AS rata_rata_harga
173 FROM items;

```

harga_termurah	harga_termahal	rata_rata_harga
40000	150000	78333.3333

- **GROUP BY (Untuk Menghitung Total Penjualan Per Kasir)**

```

138 • SELECT
139     k.nama AS nama_kasir,
140     COUNT(t.id_transaksi) AS jumlah_transaksi,
141     SUM(t.total) AS total_omzet
142 FROM transaksi t
143 JOIN kasir k ON t.id_kasir = k.id_kasir
144 GROUP BY k.nama;
145

```

nama_kasir	jumlah_transaksi	total_omzet
Kasir Utama	1	151500

- **HAVING (Mencari produk yang stocknya di bawah 20)**

```

147 • SELECT nama_item, stock
148 FROM items
149 GROUP BY id_items
150 HAVING stock < 20;
151

```

nama_item	stock
Liquid Oatberry 60ml	14

- LEFT JOIN (Untuk mengecek karyawan yang belum mempunyai akun operator)

```

153 • SELECT
154     k.nama_lengkap,
155     o.username
156 FROM karyawan k
157 LEFT JOIN operator o ON k.id_karyawan = o.id_karyawan;
158

```

nama_lengkap	username
Akizaizen Roosevelt	admin_kasir
Joshua Ardiaz	admin_kasir2
Vyka Amanul Alam	

- JOIN (Menampilkan detail transaksi lengkap)

```

125 • SELECT
126     t.id_transaksi,
127     t.tanggal,
128     k.nama AS nama_kasir,
129     i.nama_item,
130     d.kuantitas,
131     d.subtotal
132 FROM transaksi t
133 JOIN kasir k ON t.id_kasir = k.id_kasir
134 JOIN transaksi_detail d ON t.id_transaksi = d.id_transaksi
135 JOIN items i ON d.id_items = i.id_items;
136

```

id_transaksi	tanggal	nama_kasir	nama_item	kuantitas	subtotal
TRX001	2025-01-14 22:16:48	Kasir Utama	Liquid Oatberry 60ml	1	150000

- RIGHT JOIN: Menampilkan semua data Operator dan mencocokkan dengan data Karyawan

```

174
175 -- E. RIGHT JOIN: Menampilkan semua data Operator dan mencocokkan dengan data Karyawan
176 • SELECT
177     k.nama_lengkap,
178     o.username
179 FROM karyawan k
180 RIGHT JOIN operator o ON k.id_karyawan = o.id_karyawan;

```

nama_lengkap	username
Akizaizen Roosevelt	admin_kasir
Joshua Ardiaz	admin_kasir2

- FULL JOIN (Simulasi menggunakan UNION): Menampilkan semua data Karyawan dan Operator

```
182 -- F. FULL JOIN (Simulasi menggunakan UNION): Menampilkan semua data Karyawan dan Operator
183 • SELECT k.nama_lengkap, o.username
184 FROM karyawan k
185 LEFT JOIN operator o ON k.id_karyawan = o.id_karyawan
186 UNION
187 SELECT k.nama_lengkap, o.username
188 FROM karyawan k
189 RIGHT JOIN operator o ON k.id_karyawan = o.id_karyawan
```

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content: 12

	nama_lengkap	username
▶	Alirazen Roosevelt	admin_kasir
	Joshua Ardiaz	admin_kasir2
	Vyka Amanul Alam	admin

- Script SQL Lengkap: Terlampir di link GitHub di atas.

## DAFTAR PUSTAKA

1. Connolly, T., & Begg, C. *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson Education. 2014.
2. Materi Jurnal Kuliah Pemrograman Basis Data - Universitas Duta Bangsa. 2025.