Project Proposal: Small Business Inventory and Sales System

Filbert Ethan Hermawan	(24/532773/PA/22535)
Satria Bramantio Arbian	(24/533017/PA/22559)
Keita Kawabata	(24/549127/PA/23221)

Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences,
Universitas Gadjah Mada
Yogyakarta, Indonesia

I. INTRODUCTION

This project focuses on developing a database-driven management system tailored for small businesses to efficiently track inventory and manage sales operations. The system aims to replace manual record-keeping and spreadsheet-based methods with an integrated digital solution that ensures data accuracy, consistency, and accessibility. Built using Python (Flask framework) and MySQL, this system will provide automated handling of stock levels, product categorization, sales recording, and report generation. The implementation will span five weeks, with each phase building toward a complete, functional prototype.

II. PROBLEM STATEMENT

Small businesses frequently face challenges in managing their inventory and sales due to reliance on manual processes. These traditional methods often lead to common problems such as stock inaccuracies, unrecorded transactions, delays in sales reporting, and difficulty tracking customer purchases. As a result, decision-making becomes inefficient and error-prone. To overcome these challenges, a database system is proposed to automate product management, track inventory changes, and record all sales data in a structured and reliable way.

III. OBJECTIVES

The primary objectives of this project are:

1. To design and implement a relational database for managing product inventory, categories, sales, and user access.

2. To develop a web-based interface allowing business owners or staff to manage and track business operations efficiently.

3. To provide real-time stock updates and alerts when inventory levels reach a predefined threshold.

4. To generate sales and performance reports that can support decision-making.

5. To ensure data integrity, security, and ease of maintenance across all modules.

IV. SYSTEM SCOPE AND FEATURES

A. Functional Requirements

1. Product Management: Add, modify, delete, and retrieve product data including SKU, price, quantity, and category.

2. Category Management: Organize products under categories for better classification.

3. Sales Management: Record and manage customer sales transactions, including items sold and total amounts.

4. Customer Management: Store customer details and purchase history.

5. User Management: Register and authenticate system users (operators/admins).

6. Stock Monitoring: Automatically adjust inventory levels after each sale.

7. Reporting: Generate reports summarizing sales performance, total revenue, and low-stock alerts.

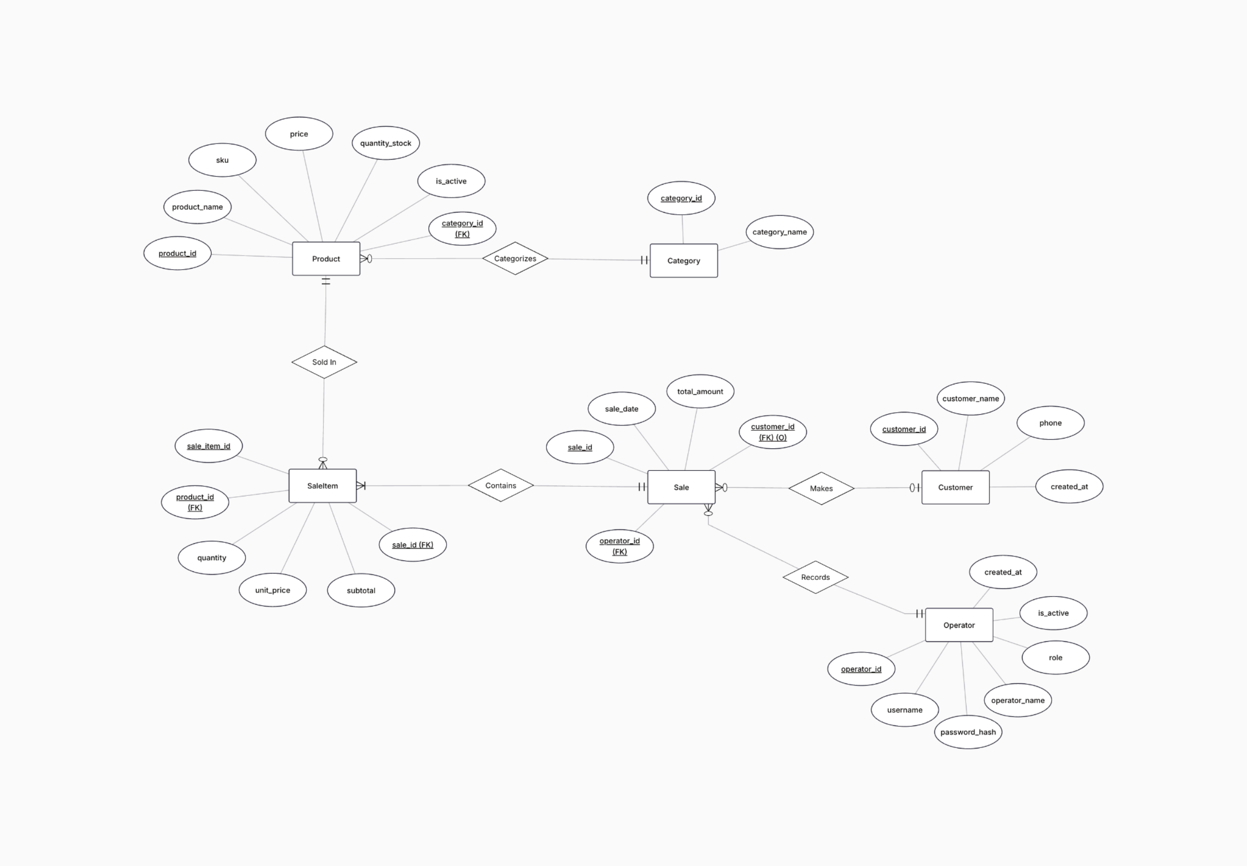
B. Non-Functional Requirements

1. Performance: System responses should be within 2 seconds for standard queries.
2. Security: All passwords must be hashed, and user roles should determine access levels.
3. Reliability: Database constraints will enforce referential integrity and prevent data loss.
4. Scalability: The database should be able to handle more users and products as the business grows.
5. Usability: Simple, intuitive web-based interface requiring minimal training.

V. DATABASE DESIGN

A. Entity-Relationship Diagram (ERD)

The ERD defines the core entities and their relationships. The system includes Product, Category, Sale, SaleItem, Customer, and Operator entities, representing business processes such as inventory control and sales recording.



Key relationships:

- Each Category contains multiple Products.
- Each Sale is recorded by one Operator and may involve one Customer.
- Each Sale consists of multiple SaleItems, linking individual Products to a Sale.
- Each Operator manages multiple Sales.

B. Database Implementation Plan

The following entities will be implemented in the relational schema:

1. Product: Stores product details (product_id, product_name, SKU, price, quantity_stock, category_id).

2. Category: Manages product categories (category_id, category_name).

3. Customer: Maintains customer information (customer_id, customer_name, phone, created_at).

4. Sale: Records each transaction (sale_id, sale_date, customer_id, operator_id, total_amount).

5. SaleItem: Links products with sales (sale_item_id, sale_id, product_id, quantity, unit_price, subtotal).

6. Operator: Manages system users (operator_id, username, password_hash, role, is_active).

VI. TECHNOLOGY USED

Component	Tool/Technology
Database Management	MySQL
Programming Language	Python
Framework	Flask
Design Tool	Draw.io / ERDPlus
IDE	Visual Studio Code
Version Control	GitHub

VII. EXPECTED OUTCOME

Upon completion, the system will deliver:

1. A fully functional MySQL database supporting inventory and sales operations.
2. A Flask-based web application allowing user interaction with the database.
3. A user-friendly interface for managing stock, recording sales, and viewing reports.
4. Documentation and SQL scripts for setup and future updates.
5. Demonstrable application as part of the final presentation.
