

# **LAPORAN**

## **PBO**

### **“Penambahan Produk ke Dalam Dealer Motor”**

*Dosen pengampu : Bpk. Bahrul*



**Disusun oleh :**

Danendra Fausta V.

NPM : 2213020246

PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS NUSANTARA PGRI KEDIRI  
2023

# KATA PENGANTAR

Pertama-tama saya ucapkan puji syukur Alhamdulillah atas rahmat dan ridho Allah SWT beserta karunia-Nya sehingga saya dapat menyelesaikan penyusunan laporan UAS Basis Data ini dengan judul “Sistem manajemen penilaian karyawan” sesuai dengan waktu yang telah ditentukan.

Saya menyadari bahwa dalam penulisan laporan ini tidak terlepas dari bantuan banyak pihak yang dengan tulus memberikan dukungan, doa, kritik sekaligus saran sehingga laporan ini dapat terselesaikan.

Saya menyadari bahwa sepenuhnya laporan ini masih jauh dari kata sempurna, oleh karena itu saya mengharapkan saran, masukan beserta kritikan yang membangun dari berbagai pihak. Semoga laporan ini dapat memberikan manfaat, petunjuk maupun acuan khususnya bagi kita semua.

Dalam penulisan laporan ini saya menyampaikan ucapan terima kasih yang tak terhingga kepada pihak-pihak yang membantu dalam menyelesaikan laporan ini. Semoga Allah SWT memberikan imbalan yang setimpal pada kita semua yang memberikan bantuan dan dapat menjadikan semua bantuan ini sebagai ibadah, Aamiin Yaa Robbal ‘Alamiin.

Kediri, 26 Desember 2023

Danendra Fausta Viandhana

# DAFTAR ISI

	Nomor
KATA PENGANTAR .....	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	<b>Error! Bookmark not defined.</b>
BAB II PEMBAHASAN .....	1
2.1 Gambaran Pembuatan Program .....	2
2.2 Pembuatan Database .....	2
BAB III PENUTUP .....	5
3.1 Kesimpulan .....	5
3.2 Penutup .....	5
DAFTAR PUSTAKA .....	iii

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

#### **A. Permasalahan**

Saya memilih topik riwayat pembayaran dealer motor karena, pada suatu kasus data pembayaran ada banyak kemungkinan terjadi kendala dalam pengelolaannya dan mengenai data apa yang dibutuhkan didalamnya.

#### **B. Solusi**

maka dari itu saya ingin membuat riwayat pembayaran dealer\_motor dari dealer motor yang terstruktur dan lebih mudah dipahami.

# BAB II

## 2.1 Gambaran Pembuatan Program

Class Diagram yang digunakan sebagai berikut

Class Diagram:

### 1. MotorDealerApp Class:

#### a. Attributes:

- ❖ table\_widget: QWidget
- ❖ refresh\_button: QPushButton
- ❖ idproduk\_input: QLineEdit
- ❖ produk\_input: QLineEdit
- ❖ harga\_input: QLineEdit
- ❖ quantity\_input: QLineEdit
- ❖ type\_input: QLineEdit
- ❖ add\_button: QPushButton
- ❖ conn: pymysql.connect

#### b. Methods:

- ❖ \_\_init\_\_(self)
- ❖ init\_ui(self)
- ❖ refresh\_data(self)
- ❖ add\_data(self)

### 2. QTableWidgetItem Class:

#### a. Attributes:

- ❖ rowCount: int
- ❖ columnCount: int
- ❖ horizontalHeaderLabels: List[str]

#### b. Methods:

- ❖ setRowCount(self, count: int)
- ❖ insertRow(self, row: int)
- ❖ setItem(self, row: int, column: int, item: QTableWidgetItem)

3. QPushButton Class:

a. Attributes:

- ❖ text: str
- ❖ clicked: Signal

b. Methods:

- ❖ \_\_init\_\_(self, text: str)
- ❖ clicked.connect(self, slot: Callable)

4. QLineEdit Class:

a. Attributes:

- ❖ text: str

b. Methods:

- ❖ \_\_init\_\_(self)
- ❖ clear(self)
- ❖ text(self)

5. QLabel Class:

a. Attributes:

- ❖ text: str

b. Methods:

- ❖ \_\_init\_\_(self, text: str)

6. pymysql.connect Class:

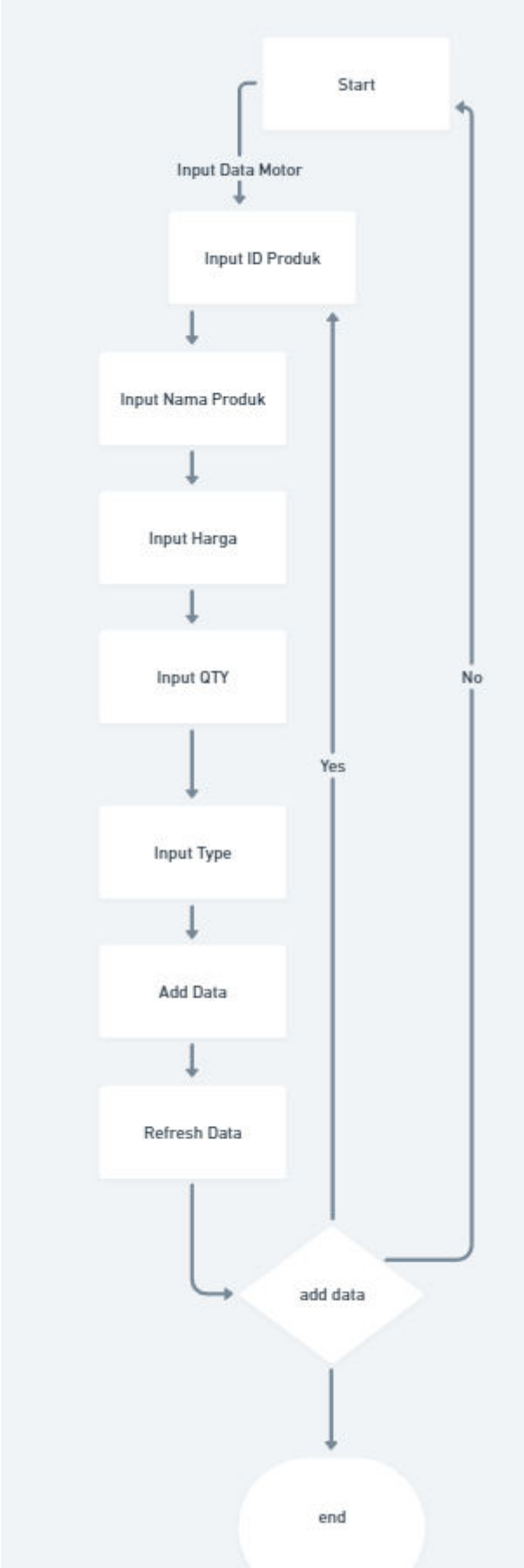
a. Attributes:

- ❖ host: str
- ❖ user: str
- ❖ password: str
- ❖ database: str

b. Methods:

- ❖ \_\_init\_\_(self, host: str, user: str, password: str, database: str)
- ❖ commit(self)
- ❖ cursor(self)

Flowchart



## Pembuatan Library

```
import sys
from PyQt5.QtWidgets import QApplication, QMainWindow, QTableWidgetItem, QVBoxLayout, QWidget, QPushButton, QLabel, QLineEdit, QFormLayout
import pymysql
```

## Pembuatan Main Window

```
class MotorDealerApp(QMainWindow):
    def __init__(self):
        super().__init__()

        self.setWindowTitle('Penambahan Produk dalam Dealer Motor')
        self.setGeometry(100, 100, 600, 400)

        self.init_ui()
```

## Pembuatan komponen-komponen yang dijadikan UI

```
def init_ui(self):
    # Komponen UI
    self.table_widget = QTableWidgetItem()
    self.table_widget.setColumnCount(5)
    self.table_widget.setHorizontalHeaderLabels(["ID", "Merek", "Harga", "Quantity", "I Type Barang"])

    self.refresh_button = QPushButton('Refresh Data')
    self.refresh_button.clicked.connect(self.refresh_data)

    self.produk_input = QLineEdit()
    self.harga_input = QLineEdit()
    self.quantity_input = QLineEdit()

    self.add_button = QPushButton('Tambah Data')
    self.add_button.clicked.connect(self.add_data)
```



## Pembuatan Layout

```
# Layout
layout = QVBoxLayout()

form_layout = QFormLayout()
form_layout.addRow('Nama Produk:', self.produk_input)
form_layout.addRow('Harga:', self.harga_input)
form_layout.addRow('QTY:', self.quantity_input)

layout.addWidget(QLabel('Input Data Motor:'))
layout.addLayout(form_layout)
layout.addWidget(self.add_button)
layout.addWidget(self.refresh_button)
layout.addWidget(self.table_widget)

central_widget = QWidget()
central_widget.setLayout(layout)
self.setCentralWidget(central_widget)
```

## Membuat Koneksi ke Database

```
# Koneksi ke database
self.conn = pymysql.connect(
    host='localhost',
    user='root',
    password='',
    database='dealer_motor'
)
```

## Membuat Pengambilan data dari Database

```
def refresh_data(self):
    # Mengambil data dari database
    with self.conn.cursor() as cursor:
        cursor.execute('SELECT * FROM produk')
        result = cursor.fetchall()

    # Menampilkan data di tabel
    self.table_widget.setRowCount(0)
    for row_number, row_data in enumerate(result):
        self.table_widget.insertRow(row_number)
        for column_number, data in enumerate(row_data):
            self.table_widget.setItem(row_number, column_number, QTableWidgetItem(str(data)))
```

## Membuat nilai dari input program

```
def add_data(self):
    # Mendapatkan nilai dari input
    merek = self.merek_input.text()
    harga = self.harga_input.text()
    quantity = self.quantity_input.text()

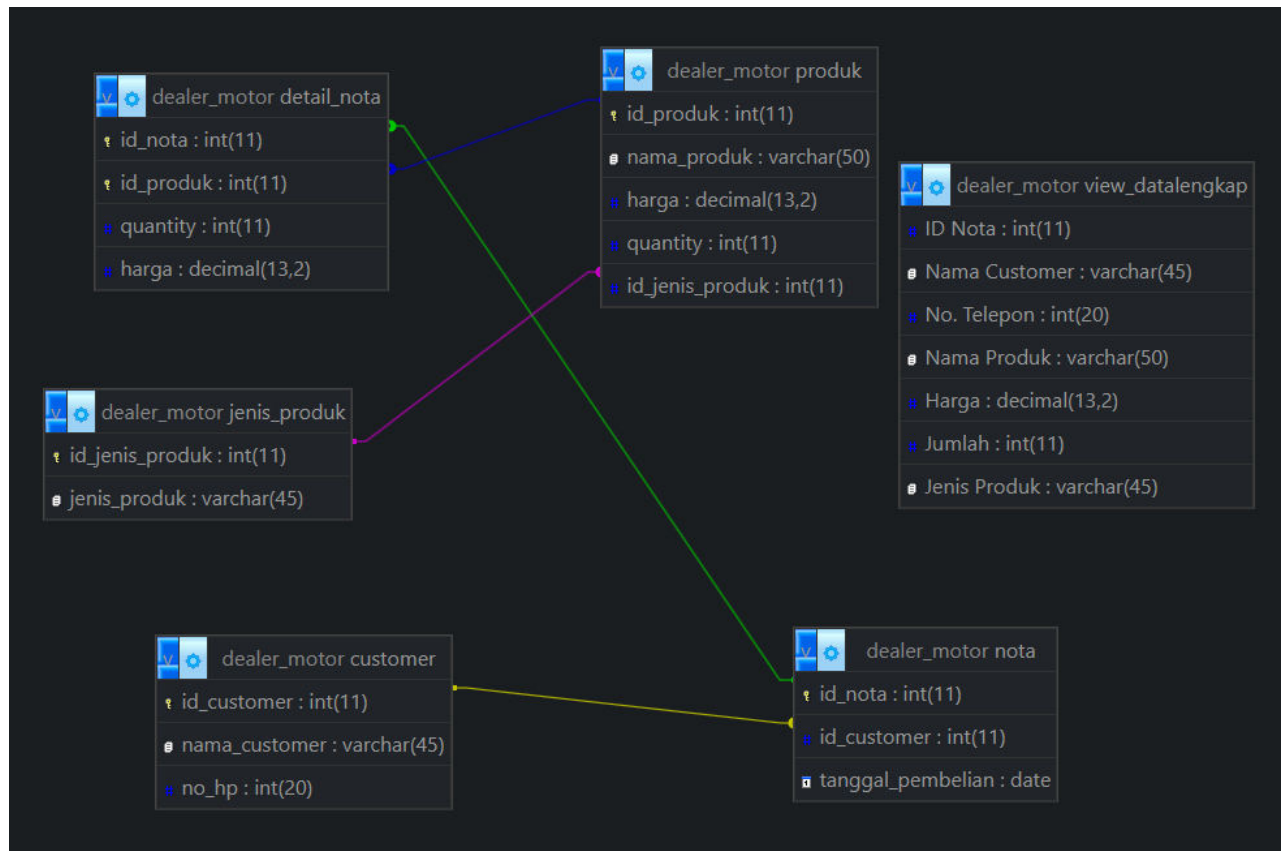
    with self.conn.cursor() as cursor:
        cursor.execute('INSERT INTO produk (id_produk, nama_produk, harga, quantity, id_jenis_produk) VALUES (%s, %s)', (merek, harga, ))
        self.conn.commit()

    # Membersihkan input setelah menambahkan data
    self.produk_input.clear()
    self.harga_input.clear()

    # Refresh data setelah menambahkan
    self.refresh_data()
```

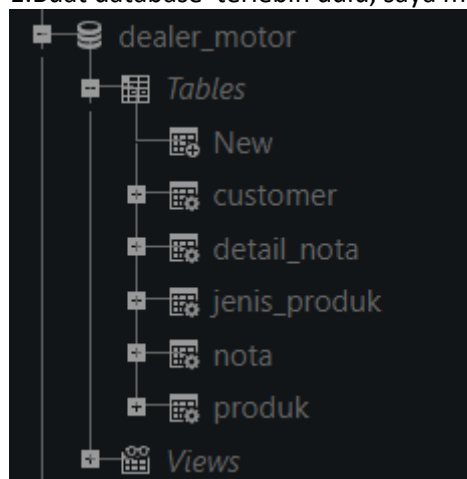
## 2. Database

### Permodelan ERD



Langkah membuat :

1. Buat database terlebih dulu, saya membuat database dengan nama dealer\_motor



Syntax yang saya gunakan adalah CREATE DATABASE dealer\_motor

## 2. Membuat table

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```

Dalam database saya membuat 5 tabel yaitu table customer, produk, jenis produk, nota, jenis nota  
Dengan menggunakan syntax diatas dan menyesuaikan dengan kolom yang diperlukan

## 3. Lakukan hal yang sama untuk mengisi table lainnya (Lakukan sesuai kebutuhan table)

## 4. Menginisiasi Trigger guna mengetahui berapa banyak stok yang dibeli oleh pelanggan dengan query berikut:

```
DELIMITER //  
CREATE TRIGGER trg_detail_nota_decrease_quantity  
AFTER INSERT ON detail_nota  
FOR EACH ROW  
BEGIN  
  
    UPDATE produk  
    SET quantity = quantity - NEW.quantity  
    WHERE id_produk = NEW.id_produk;  
END;  
//  
DELIMITER ;
```

## 5. Membuat view pada database untuk menampilkan id nota, nama customer, no telepon, harga, jumlah produk, dan jenisnya menggunakan query sebagai berikut:

```
create view view_dataengkap  
AS (  
SELECT          n.id_nota 'ID Nota',  
                c.nama_customer 'Nama Customer',  
                c.no_hp 'No. Telepon',  
                p.nama_produk 'Nama Produk',  
                p.harga 'Harga',  
                dn.quantity 'Jumlah',  
                jp.jenis_produk 'Jenis Produk'  
  
from nota n inner join customer c  
on n.id_customer = c.id_customer  
inner join detail_nota dn  
on n.id_nota = dn.id_nota  
inner join produk p  
on dn.id_produk = p.id_produk  
inner join jenis_produk jp  
on p.id_jenis_produk = jp.id_jenis_produk  
order by n.id_nota ASC);
```

## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Riwayat pembelian pada dealer motor diperlukan untuk membuktikan bahwa seseorang pernah melakukan transaksi pembelian motor pada dealer tersebut dan mengkonfirmasi pembayaran pembeli

#### **3.2 Penutup**

Dengan demikian dari penjelasan saya mengenai makalah pembuatan serta fungsi dari database dealer motor saya harap makalah ini dapat berfungsi untuk referensi kepada para pembaca. Bila ada kritik dan saran saya sangat terbuka terhadap berbagai masukan yang ada

# DAFTAR PUSTAKA

W3School."SQL Tutorial". <https://www.w3schools.com/>

ChatGPT. <https://chat.openai.com/c/0c6b8966-43d1-4d59-882d-5721cdb363c9>