

**PERANCANGAN DAN IMPLEMENTASI APLIKASI
PENJUALAN MOTOR BERBASIS DESKTOP
MENGUNAKAN JAVA**



Disusun oleh :

Nama : Agesta Ari Nugraha
NIM : 24131310143
Kelas : TI C1/3

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS TANGERANG RAYA**

2026

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan Ujian Akhir Semester (UAS) mata kuliah Pemrograman Berorientasi Objek (PBO) yang berjudul “Perancangan dan Implementasi Aplikasi Penjualan Motor Berbasis Desktop Menggunakan Java” dengan baik dan tepat waktu.

Laporan ini disusun sebagai bentuk pertanggungjawaban akademik dalam memenuhi tugas UAS mata kuliah PBO. Dalam laporan ini dijelaskan proses analisis, perancangan sistem menggunakan UML, serta implementasi aplikasi berbasis Java dengan menerapkan konsep Object Oriented Programming (OOP).

Penulis menyadari bahwa laporan ini masih memiliki kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun untuk perbaikan di masa yang akan datang.

Semoga laporan ini dapat memberikan manfaat dan menambah wawasan mengenai penerapan konsep PBO dalam pengembangan sistem informasi berbasis desktop.

Tangerang, 10 Februari 2026

Penulis

Agesta Ari Nugraha

DAFTAR ISI

KATA PENGANTAR.....	I
DAFTAR ISI	II
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Manfaat.....	2
BAB II LANDASAN TEORI	3
2.1 Sistem Informasi.....	3
2.2 Konsep Object Oriented Programming (OOP).....	3
2.2.1 Class dan Object.....	4
2.2.2 Encapsulation	4
2.2.3 Method	5
BAB III ANALISIS DAN PERANCANGAN SISTEM	6
3.1 Analisis Sistem	6
3.2 Use Case Diagram	6
3.3 Class Diagram	7
3.4 Perancangan Logika Sistem.....	8
BAB IV IMPLEMENTASI SISTEM	9
4.1 Bahasa dan Tools yang Digunakan.....	9
4.2 Tampilan Antarmuka (GUI	9
4.3 Implementasi Program.....	10
4.4 Pengujian Sistem	10
BAB V PENUTUP 18	11
5.1 Kesimpulan.....	11
5.2 Saran.....	11
DAFTAR PUSTAKA	
LAMPIRAN	

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi mendorong berbagai sektor usaha untuk menggunakan sistem komputerisasi dalam kegiatan operasionalnya. Salah satu bidang usaha yang masih banyak menggunakan sistem manual adalah dealer motor skala kecil.

Proses transaksi penjualan motor biasanya dilakukan dengan cara menghitung menggunakan kalkulator dan mencatat data pembeli secara manual. Hal ini menimbulkan beberapa permasalahan seperti:

Kesalahan perhitungan diskon dan total pembayaran

- Proses transaksi yang lambat
- Tidak adanya pencatatan data secara sistematis
- Kurangnya efisiensi kerja

Untuk mengatasi permasalahan tersebut, maka dirancang dan dibangun **Aplikasi Penjualan Motor Berbasis Desktop menggunakan Java** sebagai solusi praktis dan efisien.

1.2 Rumusan Masalah

Bagaimana merancang sistem penjualan motor berbasis desktop?

Bagaimana mengimplementasikan konsep OOP dalam aplikasi tersebut?

Bagaimana membuat sistem yang mampu menghitung diskon dan total bayar secara otomatis?

1.3 Tujuan

Membuat aplikasi penjualan motor berbasis Java

Menerapkan konsep Object Oriented Programming (OOP)

Mengotomatisasi proses perhitungan transaksi

1.4 Manfaat

Mempermudah proses transaksi

Mengurangi kesalahan perhitungan

Meningkatkan efisiensi kerja

BAB II

LANDASAN TEORI

2.1 Sistem Informasi

Sistem informasi merupakan suatu sistem yang terdiri dari komponen-komponen yang saling berhubungan untuk mengumpulkan, mengolah, menyimpan, dan mendistribusikan informasi guna mendukung proses pengambilan keputusan dalam suatu organisasi. Sistem informasi biasanya melibatkan kombinasi antara manusia, perangkat keras (hardware), perangkat lunak (software), data, serta prosedur kerja.

Dalam konteks aplikasi penjualan motor yang dibuat, sistem informasi berfungsi untuk membantu proses transaksi penjualan agar lebih efektif dan efisien. Sistem ini mengolah data seperti nama pembeli, merk motor, jenis motor, harga, metode pembayaran, hingga perhitungan diskon dan total bayar secara otomatis. Dengan adanya sistem informasi berbasis komputer, proses perhitungan menjadi lebih cepat, akurat, dan meminimalkan kesalahan manusia (human error).

2.2 Konsep Object Oriented Programming (OOP)

Object Oriented Programming (OOP) adalah metode pemrograman yang berorientasi pada objek. Dalam OOP,

program dibangun berdasarkan konsep class dan object yang merepresentasikan suatu entitas dalam dunia nyata.

Pada aplikasi penjualan motor ini, konsep OOP diterapkan untuk membuat struktur program yang lebih terorganisir, modular, dan mudah dikembangkan.

Beberapa konsep dasar OOP yang digunakan dalam aplikasi ini adalah:

1. Class

Class merupakan blueprint atau cetakan untuk membuat objek. Class berisi atribut (data) dan method (fungsi). Contohnya, class *Motor* dan class *Transaksi* yang digunakan dalam aplikasi.

2. Object

Object adalah hasil instansiasi dari class. Object digunakan untuk mengakses atribut dan method yang terdapat dalam class. Dalam program ini, objek dibuat untuk memproses data motor dan transaksi.

3. Encapsulation

Encapsulation adalah proses menyembunyikan data dalam suatu class dan hanya dapat diakses melalui method tertentu.

Hal ini bertujuan untuk menjaga keamanan data dan mencegah perubahan yang tidak diinginkan.

4. Method

Method adalah fungsi atau prosedur yang terdapat dalam class dan digunakan untuk melakukan suatu proses, seperti menghitung diskon dan total pembayaran.

Dengan menerapkan konsep OOP, aplikasi penjualan motor yang dibuat menjadi lebih terstruktur, mudah dipahami, serta lebih fleksibel untuk dikembangkan di masa mendatang.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Sistem memungkinkan pengguna untuk:

Menginput nama pembeli

Memilih merk motor

Memilih jenis motor

Menginput harga

Memilih metode pembayaran

Menghitung diskon

Menampilkan total pembayaran

3.2 Use Case Diagram (Penjelasan)

Aktor: Admin/Kasir

Use Case:

Input data pembeli

Pilih merk dan jenis motor

Hitung transaksi

Reset data

Keluar aplikasi

3.3 Class Diagram

Class Motor

Atribut:

merk : String

jenis : String

harga : double

Method:

setMerk()

setJenis()

setHarga()

getHarga()

Class Transaksi

Atribut:

namaPembeli : String

diskon : double

totalBayar : double

Method:

hitungDiskon(double harga, String metodeBayar)

hitungTotal(double harga, double diskon)

◆ Class MainForm (GUI)

Berfungsi untuk:

Mengambil input dari user

Memanggil class Motor dan Transaksi

Menampilkan hasil perhitungan

3.4 Perancangan Logika Sistem

Ketentuan diskon:

Pembayaran Tunai → Diskon 20%

Pembayaran Kredit → Diskon 10%

Rumus:

Diskon = harga × persen diskon

Total Bayar = harga – diskon

Contoh:

Harga = 13.500.000

Diskon 20% = 2.700.000

Total Bayar = 10.800.000

Hasil tersebut sesuai dengan output pada aplikasi.

BAB IV

IMPLEMENTASI SISTEM

4.1 Bahasa Pemrograman

Java

NetBeans IDE

Java Swing (GUI)

4.2 Tampilan Form

Form terdiri dari:

TextField Nama Pembeli

ComboBox Merk Motor

RadioButton Jenis Motor (Bebek, Sport, Matik)

TextField Harga

ComboBox Metode Pembayaran

Tombol Hitung

Tombol Hitung Lagi

Tombol Selesai

TextField Diskon

TextField Total Bayar

4.3 Implementasi Tombol Hitung

Proses saat tombol HITUNG ditekan:

Ambil nilai harga

Cek metode pembayaran

Hitung diskon sesuai aturan

Hitung total bayar

Tampilkan hasil ke TextField

Logika program berjalan sesuai konsep OOP dengan **memanggil method dari class Transaksi.**

4.4 Pengujian Sistem

Pengujian dilakukan dengan:

Input pembayaran tunai → diskon 20%

Input pembayaran kredit → diskon 10%

Hasil pengujian menunjukkan perhitungan berjalan dengan benar.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Aplikasi Penjualan Motor berbasis Java Desktop berhasil dibuat dan mampu:

Mengotomatisasi perhitungan diskon

Menghitung total pembayaran secara akurat

Menerapkan konsep OOP

Meningkatkan efisiensi transaksi

Sistem ini cocok digunakan untuk dealer motor skala kecil hingga menengah.

5.2 Saran Pengembangan

Untuk pengembangan selanjutnya, sistem dapat ditambahkan:

Database MariaDB untuk menyimpan data transaksi

Fitur cetak struk

Laporan penjualan bulanan

Sistem login admin

PENUTUP

Dengan dirancangnya aplikasi ini, diharapkan proses penjualan motor dapat dilakukan secara lebih cepat, akurat, dan efisien. Implementasi konsep Object Oriented Programming dalam pengembangan sistem ini menunjukkan bahwa pendekatan berorientasi objek mampu menghasilkan aplikasi yang terstruktur, modular, dan mudah dikembangkan di masa depan.

DAFTAR PUSTAKA

Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education.

Rosa, A. S., & Shalahuddin, M. (2018). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.

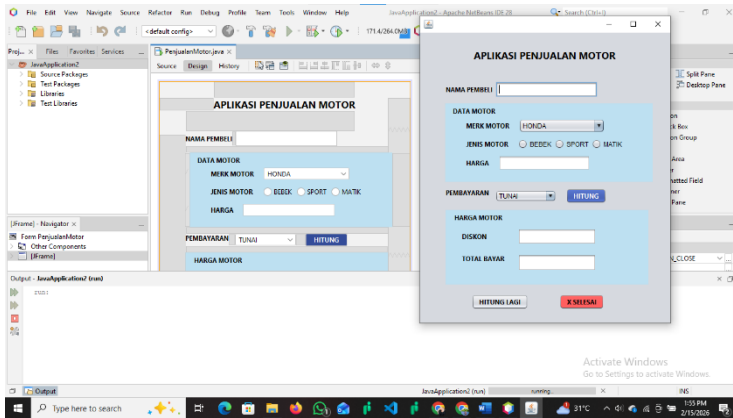
Pressman, R. S., & Maxim, B. R. (2015). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill Education.

Deitel, H. M., & Deitel, P. J. (2017). *Java: How to Program* (11th ed.). Pearson.

Oracle. (2023). *Java Documentation*. Retrieved from <https://docs.oracle.com/javase/>

LAMPIRAN

A. APLIKASI



B. SOURCE CODE PROGRAM

1. CLASS MOTOR

```
public class Motor {  
  
    private String merk;  
  
    private String jenis;  
  
    private double harga;  
  
    // Setter  
  
    public void setMerk(String merk) {  
        this.merk = merk;  
    }  
  
    public void setJenis(String jenis) {
```

```

this.jenis = jenis;
    }
    public void setHarga(double harga) {
        this.harga = harga;
    }
    // Getter
    public String getMerk() {
        return merk;
    }
    public String getJenis() {
        return jenis;
    }
    public double getHarga() {
        return harga;
    }
}

```

2. Class Transaksi

```

public class Transaksi {
    private String namaPembeli;
    private double diskon;
    private double totalBayar;
    public void setNamaPembeli(String namaPembeli) {
        this.namaPembeli = namaPembeli;
    }
}

```

```

    }

    public double hitungDiskon(double harga, String metodeBayar) {
        if (metodeBayar.equals("Tunai")) {
            diskon = harga * 0.20;
        } else if (metodeBayar.equals("Kredit")) {
            diskon = harga * 0.10;
        } else {
            diskon = 0;
        }
        return diskon;
    }

    public double hitungTotal(double harga) {
        totalBayar = harga - diskon;
        return totalBayar;
    }

    public double getDiskon() {
        return diskon;
    }

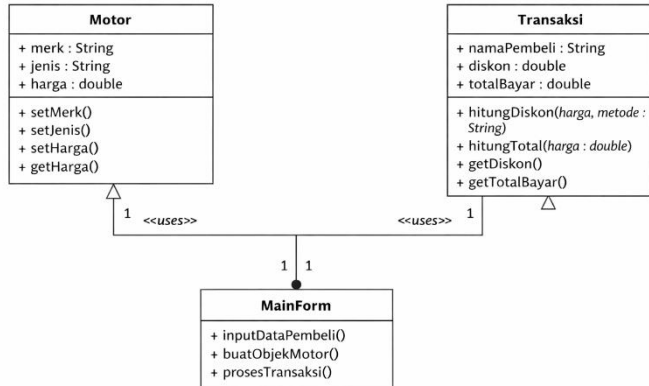
    public double getTotalBayar() {
        return totalBayar;
    }
}

```

3. Kode Tombol HITUNG (di MainForm)

```
private void btnHitungActionPerformed(java.awt.event.ActionEvent evt) {  
  
    Motor motor = new Motor();  
    Transaksi transaksi = new Transaksi();  
  
    String nama = txtNama.getText();  
    String merk = cmbMerk.getSelectedItem().toString();  
    String metode = cmbMetode.getSelectedItem().toString();  
    double harga = Double.parseDouble(txtHarga.getText());  
  
    motor.setMerk(merk);  
    motor.setHarga(harga);  
    transaksi.setNamaPembeli(nama);  
  
    double diskon = transaksi.hitungDiskon(harga, metode);  
    double total = transaksi.hitungTotal(harga);  
  
    txtDiskon.setText(String.valueOf(diskon));  
    txtTotal.setText(String.valueOf(total));  
}
```

4. Diagram UML



5. Dokumentasi Pengujian

No	Input	Metode Bayar	Hasil Diskon	Total Bayar
1	15.000.000	Tunai	3.000.000	12.000.000