

**LAPORAN TUGAS BESAR UJIAN AKHIR SEMESTER 2025**  
**PEMROGRAMAN BERORIENTASI OBJEK**  
**(Mata Kuliah Pemrograman Berorientasi Objek)**



**Disusun oleh :**

Luthfi Harisna Mufti

2411523019

**Dosen Pengampu :**

Jefril Rahmadoni, M.Kom

**FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS  
PADANG  
2025**

## DAFTAR ISI

<b>BAB I .....</b>	<b>3</b>
<b>ANALISIS MASALAH DAN SOLUSI .....</b>	<b>3</b>
1.1 <b>Profil Perusahaan .....</b>	<b>3</b>
1.2 <b>Identifikasi Permasalahan .....</b>	<b>3</b>
1.3 <b>Solusi dengan Pendekatan Sistem Informasi (PBO) .....</b>	<b>3</b>
1.4 <b>Penerapan JDBC sebagai Solusi Penyimpanan Permanen .....</b>	<b>4</b>
<b>BAB II .....</b>	<b>5</b>
<b>IMPLEMENTASI KODE PROGRAM .....</b>	<b>5</b>
2.1 <b>Penjelasan Program .....</b>	<b>5</b>
2.2 <b>Komponen Program .....</b>	<b>5</b>
2.3 <b>Interface: Manajemen.java .....</b>	<b>6</b>
2.4 <b>Kelas Koneksi: DatabaseConnection.java .....</b>	<b>6</b>
2.5 <b>Superclass: Optical.java .....</b>	<b>7</b>
2.6 <b>Subclass: TransaksiPelanggan.java .....</b>	<b>8</b>
2.7 <b>Kelas Data Access Object (DAO): TransaksiDAO.java .....</b>	<b>10</b>
2.8 <b>Main.java .....</b>	<b>11</b>
<b>BAB III .....</b>	<b>14</b>
<b>HASIL DAN PEMBAHASAN .....</b>	<b>14</b>
3.1 <b>Penjelasan Elemen PBO dalam Sistem .....</b>	<b>14</b>
3.2 <b>Konektivitas Database (JDBC) .....</b>	<b>14</b>
3.3 <b>Logika Program dan Manipulasi Data .....</b>	<b>15</b>
3.4 <b>Hasil Pengujian Program .....</b>	<b>15</b>
<b>PENUTUP .....</b>	<b>20</b>
4.1 <b>Kesimpulan .....</b>	<b>20</b>
4.2 <b>Saran .....</b>	<b>20</b>
<b>DAFTAR PUSTAKA.....</b>	<b>22</b>

# **BAB I**

## **ANALISIS MASALAH DAN SOLUSI**

### **1.1 Profil Perusahaan**

Harisna Optical merupakan sebuah unit bisnis yang bergerak di bidang pelayanan kesehatan mata dan penyediaan alat bantu penglihatan (optik). Perusahaan ini berlokasi strategis di Jl. Pahlawan, Kampung Jawa I, Kec. Pariaman Tengah, Kota Pariaman, Sumatera Barat. Layanan utama yang diberikan meliputi pemeriksaan tajam penglihatan (refraksi), konsultasi resep lensa, serta penjualan berbagai jenis bingkai (frame) dan lensa kacamata bagi masyarakat di wilayah Pariaman.

### **1.2 Identifikasi Permasalahan**

Berdasarkan hasil observasi pada operasional Harisna Optical, ditemukan beberapa permasalahan utama yang menghambat efisiensi pelayanan:

- a. Pengelolaan Data Konvensional: Pendataan pelanggan dan resep mata masih dilakukan secara manual pada buku besar, sehingga menyulitkan petugas dalam pencarian riwayat medis pelanggan lama.
- b. Risiko Kehilangan Data: Penyimpanan data fisik sangat rentan terhadap kerusakan atau kehilangan, serta bersifat volatile jika hanya mengandalkan ingatan petugas tanpa sistem database yang permanen.
- c. Ketidakakuratan Perhitungan: Proses penghitungan total biaya transaksi yang melibatkan harga frame, jenis lensa, dan pajak PPN sering kali mengalami kesalahan hitung (human error).
- d. Kurangnya Validasi Input: Penginputan data resep (minus/silinder) yang tidak terstandarisasi sering menyebabkan kesalahan komunikasi saat proses pembuatan kacamata di laboratorium.

### **1.3 Solusi dengan Pendekatan Sistem Informasi (PBO)**

Untuk mengatasi permasalahan tersebut, diusulkan pembangunan Sistem Informasi Manajemen Harisna Optical dengan menerapkan prinsip Pemrograman Berorientasi Objek (PBO). Pendekatan ini menawarkan solusi terstruktur melalui elemen-elemen berikut:

Abstraksi dan Encapsulation : Merepresentasikan entitas nyata seperti Pelanggan dan Kacamata ke dalam Class yang aman, di mana data sensitif terlindungi dari perubahan ilegal.

- a. Inheritance / Pewarisan: Membangun hubungan antara superclass Optical dan subclass TransaksiPelanggan untuk meningkatkan efisiensi kode dan memudahkan pengembangan sistem di masa depan.
- b. Implementasi Interface: Menggunakan interface Manajemen untuk memastikan seluruh modul sistem memiliki standar operasional yang sama dalam mengelola data.
- c. Automasi Matematika : Mengintegrasikan fungsi perhitungan otomatis untuk total biaya dan pajak guna menjamin akurasi finansial pada setiap nota transaksi.

#### 1.4 Penerapan JDBC sebagai Solusi Penyimpanan Permanen

Sesuai dengan standar JDBC (Java Database Connectivity), sistem ini menghubungkan aplikasi Java langsung ke DBMS MySQL.

- a. Penyimpanan Non-Volatile: Data tidak lagi disimpan di RAM yang bersifat sementara, melainkan disimpan secara permanen di database db\_harisna\_optical.
- b. Integritas Data dengan CRUD: Memungkinkan petugas melakukan operasi Create, Read, Update, dan Delete secara real-time dan akurat melalui PreparedStatement untuk menghindari manipulasi data yang tidak sah.

Berdasarkan analisis permasalahan di atas, penerapan Pemrograman Berorientasi Objek (PBO) bukan sekadar kebutuhan teknis, melainkan solusi strategis bagi Harisna Optical untuk menjaga keberlangsungan data resep pelanggan di Pariaman Tengah. Dengan menggunakan arsitektur JDBC yang memisahkan antara *User Interface* dan *Database*, sistem ini mampu menjamin keamanan data sekaligus memberikan kecepatan dalam pelayanan transaksi sehari-hari. Implementasi ini sekaligus menjadi bukti penerapan kompetensi mahasiswa dalam membangun perangkat lunak yang *robust*, terstruktur, dan siap pakai di dunia nyata.

## **BAB II**

### **IMPLEMENTASI KODE PROGRAM**

Pada bagian ini, dilampirkan implementasi kode program Java yang telah dibangun. Setiap kelas dirancang untuk memenuhi kriteria Pemrograman Berorientasi Objek (PBO) dan integrasi database (JDBC) sesuai dengan modul praktikum.

#### **2.1 Penjelasan Program**

Program ini merupakan sistem manajemen transaksi pada Harisna Optical berbasis Java yang terintegrasi dengan database MySQL menggunakan driver JDBC. Program diawali dengan menghubungkan sistem ke database melalui driver JDBC; jika koneksi berhasil, pesan sukses akan ditampilkan, sedangkan kegagalan akan menyebabkan program keluar.

Sistem ini berfungsi untuk mengelola data transaksi kacamata di lokasi Pariaman Tengah. Pada fitur penambahan transaksi, pengguna memasukkan informasi nama pelanggan, resep mata, harga frame, dan harga lensa yang kemudian disimpan secara permanen ke dalam database. Program ini juga memiliki validasi input menggunakan Exception Handling untuk memastikan data yang dimasukkan sesuai format, seperti input harga yang wajib berupa angka. Setelah pengguna keluar dari menu, koneksi ke database ditutup untuk memastikan semua sumber daya dilepaskan dengan benar.

#### **2.2 Komponen Program**

- A. Class
  - a. Manajemen.java: Sebagai interface yang berisi kontrak metode standar untuk pengelolaan data.
  - b. Optical.java: Sebagai superclass yang berisi atribut dan metode dasar seperti nama toko dan alamat di Pariaman.
  - c. TransaksiPelanggan.java: Sebagai subclass dari Optical yang mewarisi atribut induk dan mengimplementasikan logika transaksi.
  - d. DatabaseConnection.java & TransaksiDAO.java: Kelas yang mengatur manajemen koneksi database dan operasi CRUD.

- e. Main.java: Kelas utama yang mengatur interaksi pengguna, termasuk menu utama dan penanganan kesalahan input.

B. Object

- a. Objek Scanner: Digunakan untuk membaca input dari pengguna melalui konsol.
- b. Objek Connection: Instance dari pustaka JDBC untuk menjaga sesi komunikasi aktif dengan MySQL.
- c. Objek PreparedStatement: Dibuat untuk menjalankan perintah SQL (seperti Insert) secara aman guna menghindari SQL Injection.
- d. Objek ResultSet: Digunakan untuk menyimpan hasil query yang diambil dari database.

C. Constructor

- a. Constructor pada Kelas Optical: Berfungsi menginisialisasi atribut nama toko dan alamat saat objek dibuat, serta menginisialisasi ArrayList.
- b. Constructor pada Kelas TransaksiPelanggan: Menggunakan kata kunci super() untuk memanggil constructor superclass serta menginisialisasi detail transaksi.
- c. Constructor JDBC: Menggunakan DriverManager.getConnection() untuk membangun jembatan koneksi ke database.

### 2.3 Interface: Manajemen.java

File ini berfungsi sebagai kontrak standar (interface) untuk memastikan konsistensi metode di seluruh system.

```
// ini kontrak standar buat nentuin apa aja fungsi yang wajib ada di sistem
public interface Manajemen {
    void tambahData();
    void tampilkanData();
    void hapusData(String id);
}
```

### 2.4 Kelas Koneksi: DatabaseConnection.java

Kelas ini menangani sesi komunikasi aktif dengan DBMS MySQL menggunakan komponen utama JDBC API.

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseConnection {
    // ini alamat database mysql di laptop, nama db-nya db_harisna_optical
    private static final String URL = "jdbc:mysql://localhost:3306/db_harisna_optical";
    private static final String USER = "root";
    private static final String PASSWORD = "";

    // method ini buat panggil koneksi tiap kali mau hubungin java ke database
    public static Connection getKoneksi() throws SQLException {
        // pake drivermanager buat buka akses koneksinya
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }
}

```

## 2.5 Superclass: Optical.java

Merupakan kelas induk (superclass) yang menyimpan informasi umum organisasi dan mengimplementasikan *Collection Framework*

```

import java.util.ArrayList;

// ini kelas induk, isinya info umum toko yang bakal dipake di kelas lain
public abstract class Optical implements Manajemen {
    protected String namaToko;
    protected String alamat;
    // pake ArrayList buat catat riwayat atau log biar gak hilang selama aplikasi jalan
    protected ArrayList<String> logAktivitas = new ArrayList<>();
}

```

```
public Optical(String namaToko, String alamat) {  
    this.namaToko = namaToko;  
    this.alamat = alamat;  
}  
}
```

## 2.6 Subclass: TransaksiPelanggan.java

Kelas ini menangani logika bisnis khusus, manipulasi String, manipulasi Date, serta perhitungan matematika.

```
import java.time.LocalDate;  
import java.time.format.DateTimeFormatter;  
  
// ini kelas anak yang ambil sifat dari kelas Optical (inheritance)  
public class TransaksiPelanggan extends Optical {  
    private String namaPelanggan;  
    private String resep;  
    private double hargaFrame;  
    private double hargaLensa;  
    private LocalDate tanggal;  
  
    public TransaksiPelanggan(String nama, String resep, double frame, double lensa) {  
        // panggil constructor induk buat set lokasi toko di Pariaman  
        super("Harisna Optical", "Jl. Pahlawan, Kp. Jawa I, Pariaman Tengah");  
  
        // biar rapi, nama pelanggan otomatis dibikin kapital semua  
        this.namaPelanggan = nama.trim().toUpperCase();  
        this.resep = resep;  
        this.hargaFrame = frame;  
        this.hargaLensa = lensa;
```

```

// ambil tanggal otomatis hari ini pas transaksi dibuat
this.tanggal = LocalDate.now();
}

// rumus matematika buat hitung total belanja kacamata plus pajak 11%
public double hitungTotal() {
    return (hargaFrame + hargaLensa) * 1.11;
}

@Override
public void tambahData() {
    // masukin info input ke dalam list log history
    logAktivitas.add("Input transaksi baru untuk pelanggan: " +
namaPelanggan);
}

@Override
public void tampilkanData() {
    // ubah format tanggal biar enak dibaca (contoh: 24 December 2025)
    DateTimeFormatter formatIndo = DateTimeFormatter.ofPattern("dd
MMMM yyyy");

    System.out.println("\n--- NOTA TRANSAKSI HARISNA OPTICAL ---");
    System.out.println("Alamat Toko : " + this.alamat);
    System.out.println("Tanggal : " + tanggal.format(formatIndo));
    System.out.println("Pelanggan : " + namaPelanggan);
    System.out.println("Resep Mata : " + resep);

    // pake String.format biar angka bayarnya gak kepanjangan nol di
belakangnya
    System.out.println("Total Bayar : Rp " + String.format("%.2f",
hitungTotal()));
}

```

```

        System.out.println("-----");
    }

    @Override
    public void hapusData(String id) {}

    // method bantu buat ambil data pas mau disave ke database
    public String getNama() { return namaPelanggan; }
    public String getResep() { return resep; }
}

```

## 2.7 Kelas Data Access Object (DAO): TransaksiDAO.java

Kelas yang mengelola fungsi CRUD menggunakan PreparedStatement untuk berinteraksi dengan database

```

import java.sql.*;

public class TransaksiDAO {
    // ini fungsi buat mindahin data dari java ke tabel transaksi di database
    public void simpan(TransaksiPelanggan t) {
        String sql = "INSERT INTO transaksi (nama, resep, total) VALUES (?, ?, ?)";

        // pake try-with-resources biar koneksi otomatis ketutup sendiri kalau
        udah kelar
        try (Connection conn = DatabaseConnection.getKoneksi();
             PreparedStatement pstmt = conn.prepareStatement(sql)) {

            // isi tanda tanya di query SQL sesuai urutannya
            pstmt.setString(1, t.getNama());
            pstmt.setString(2, t.getResep());
            pstmt.setDouble(3, t.hitungTotal());
            pstmt.executeUpdate(); // eksekusi simpan ke database
        }
    }
}

```

```

        System.out.println("Data tersimpan ke database!");
    } catch (SQLException e) {
        // kalau ada masalah sama mysql-nya, pesan error muncul di sini
        System.err.println("Gagal: " + e.getMessage());
    }
}
}
}

```

## 2.8 Main.java

```

import java.util.InputMismatchException;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        // siapin scanner buat baca ketikan user di console
        Scanner sc = new Scanner(System.in);
        TransaksiDAO dao = new TransaksiDAO();

        System.out.println("==== SISTEM MANAJEMEN HARISNA OPTICAL
====");

        // perulangan biar menu terus muncul selama user gak milih keluar
        while (true) {
            System.out.println("\nMenu Utama:");
            System.out.println("1. Tambah Transaksi");
            System.out.println("2. Keluar");
            System.out.print("Pilih menu: ");

            int pil;
            try {
                pil = sc.nextInt();
            }

```

```

sc.nextLine(); // bersihin sisa enter di buffer
} catch (InputMismatchException e) {
    // kalau user malah ngetik huruf pas disuruh milih menu angka
    System.err.println("Input salah! Masukkan angka 1 atau 2.");
    sc.nextLine();
    continue;
}

if (pil == 1) {
    // handle kalau ada kesalahan pas ngetik harga
    try {
        System.out.print("Nama Pelanggan: ");
        String n = sc.nextLine();

        System.out.print("Resep Mata: ");
        String r = sc.nextLine();

        System.out.print("Harga Frame: ");
        double f = sc.nextDouble();

        System.out.print("Harga Lensa: ");
        double l = sc.nextDouble();
        sc.nextLine(); // bersihin buffer lagi

        // buat objek transaksi baru dari data yang diinput
        TransaksiPelanggan t = new TransaksiPelanggan(n, r, f, l);

        // jalankan fungsi simpan ke database mysql
        dao.simpan(t);

        // cetak nota di layar
        t.tampilkanData();
    }
}

```

```
        } catch (InputMismatchException e) {
            // pesan peringatan kalau harga diisi pake huruf
            System.err.println("Error: Harga frame dan lensa harus berupa
angka!");
            sc.nextLine();
        }
    } else if (pil == 2) {
        System.out.println("Terima kasih telah menggunakan sistem Harisna
Optical.");
        break; // stop looping dan keluar aplikasi
    } else {
        System.out.println("Pilihan tidak tersedia.");
    }
}
sc.close();
}
```

## **BAB III**

### **HASIL DAN PEMBAHASAN**

Pada bab ini, saya akan menjelaskan hasil implementasi dari sistem Harisna Optical serta bagaimana konsep-konsep Pemrograman Berorientasi Objek (PBO) dan JDBC diterapkan dalam kodingan untuk menyelesaikan masalah operasional di toko.

#### **3.1 Penjelasan Elemen PBO dalam Sistem**

Dalam membangun aplikasi ini, saya menerapkan prinsip-prinsip PBO agar kode tetap terstruktur dan mudah dikelola:

- A. Class dan Object: Saya membagi logika program ke dalam beberapa kelas, seperti TransaksiPelanggan yang berfungsi sebagai blueprint untuk mengolah data setiap pelanggan . Setiap kali data diinput melalui Main.java, sistem akan menciptakan sebuah Object baru dari kelas tersebut untuk diproses lebih lanjut .
- B. Inheritance (Pewarisan): Saya merancang kelas Optical sebagai superclass yang menyimpan atribut umum seperti nama toko dan alamat di Pariaman Tengah . Atribut ini kemudian diwariskan ke kelas TransaksiPelanggan (subclass) sehingga saya tidak perlu menulis ulang data toko di setiap kelas.
- C. Interface: Saya menggunakan interface Manajemen untuk menentukan metode standar seperti tambahData() dan tampilkanData() . Dengan cara ini, saya memastikan bahwa setiap kelas yang mengimplementasikannya memiliki struktur fungsi yang sama .
- D. Collection Framework: Untuk mencatat riwayat aktivitas di sistem, saya menggunakan ArrayList<String> pada kelas induk . Kelebihannya adalah ukuran list ini bisa bertambah secara dinamis sesuai banyaknya aktivitas yang terjadi .

#### **3.2 Konektivitas Database (JDBC)**

Sesuai dengan kriteria tugas, sistem ini sudah terhubung ke database MySQL secara permanen:

- A. Koneksi API: Saya menggunakan driver JDBC untuk menghubungkan aplikasi Java dengan database db\_harisna\_optical .

- B. Fungsi CRUD (Create): Melalui kelas TransaksiDAO, saya menerapkan fungsi Create menggunakan PreparedStatement . Saya menggunakan tanda tanya (?) pada query SQL untuk memastikan data yang masuk ke tabel transaksi aman dan tidak merusak sistem .

### 3.3 Logika Program dan Manipulasi Data

- A. Perhitungan Matematika: Program secara otomatis menghitung total bayar dengan rumus: (Harga Frame + Harga Lensa) 1.11 (untuk PPN 11%). Ini sangat membantu untuk menghindari kesalahan hitung manual yang sering terjadi di toko.
- B. Manipulasi String: Saya menggunakan fungsi .toUpperCase() agar nama pelanggan yang diinput selalu tersimpan dengan huruf kapital sehingga data di database terlihat rapi dan seragam .
- C. Manipulasi Date: Untuk nota transaksi, saya menggunakan LocalDate dan DateTimeFormatter agar tanggal yang muncul di nota mengikuti format standar Indonesia (dd MMMM yyyy) .
- D. Exception Handling: Saya membungkus bagian input dengan blok try-catch untuk menangkap InputMismatchException . Jadi, jika saya salah memasukkan huruf pada kolom harga, program tidak akan langsung error melainkan memberikan pesan peringatan.

### 3.4 Hasil Pengujian Program

Berikut adalah bukti bahwa program telah berjalan dengan sukses :

- A. Tampilan Menu: Program menampilkan menu pilihan (Tambah Transaksi atau Keluar) menggunakan perulangan while yang akan terus berjalan sampai user memilih menu keluar .

```
PS E:\kuliah\Semester 3\PBO\Tugas Besar PBO 2025 Luthfi Harisna
Mufti_2411523019_PBO C> & 'C:\Program Files\Eclipse Adoptium\jdk-
17.0.15.6-hotspot\bin\java.exe'
@C:\Users\SWIFTG~1\AppData\Local\Temp\cp_5pii67f9f36qd4d6qj7at46op.ar
gfile' 'Main'
== SISTEM MANAJEMEN HARISNA OPTICAL ==
```

Menu Utama:

1. Tambah Transaksi
2. Keluar

Pilih menu: 1

Nama Pelanggan: Budi

Resep Mata: L -2, R -5

Harga Frame: 120000

Harga Lensa: 200000

Data tersimpan ke database!

--- NOTA TRANSAKSI HARISNA OPTICAL ---

Alamat Toko : Jl. Pahlawan, Kp. Jawa I, Pariaman Tengah

Tanggal : 24 December 2025

Pelanggan : BUDI

Resep Mata : L -2, R -5

Total Bayar : Rp 355,200.00

---

Menu Utama:

1. Tambah Transaksi
2. Keluar

Pilih menu: 1

Nama Pelanggan: Rusdi

Resep Mata: L -2,5 , R -3,4

Harga Frame: 135000

Harga Lensa: 175000

Data tersimpan ke database!

--- NOTA TRANSAKSI HARISNA OPTICAL ---

Alamat Toko : Jl. Pahlawan, Kp. Jawa I, Pariaman Tengah

Tanggal : 24 December 2025

Pelanggan : RUSDI

Resep Mata : L -2,5 , R -3,4

Total Bayar : Rp 344,100.00

---

Menu Utama:

1. Tambah Transaksi
2. Keluar

Pilih menu: 1

Nama Pelanggan: Ana

Resep Mata: L -1,7 , R -2,6

Harga Frame: 170000

Harga Lensa: 210000

Data tersimpan ke database!

--- NOTA TRANSAKSI HARISNA OPTICAL ---

Alamat Toko : Jl. Pahlawan, Kp. Jawa I, Pariaman Tengah

Tanggal : 24 December 2025

Pelanggan : ANA

Resep Mata : L -1,7 , R -2,6

Total Bayar : Rp 421,800.00

---

Menu Utama:

1. Tambah Transaksi
2. Keluar

Pilih menu: 1

Nama Pelanggan: Icha

Resep Mata: L -3.5 , R -4,5

Harga Frame: 200000

Harga Lensa: 250000

Data tersimpan ke database!

Pilih menu: 1

Nama Pelanggan: Ana

Resep Mata: L -1,7 , R -2,6

Harga Frame: 170000

Harga Lensa: 210000

Data tersimpan ke database!

--- NOTA TRANSAKSI HARISNA OPTICAL ---

Alamat Toko : Jl. Pahlawan, Kp. Jawa I, Pariaman Tengah

Tanggal : 24 December 2025

Pelanggan : ANA

Resep Mata : L -1,7 , R -2,6

Total Bayar : Rp 421,800.00

---

Menu Utama:

1. Tambah Transaksi

2. Keluar

Pilih menu: 1

Nama Pelanggan: Icha

Resep Mata: L -3.5 , R -4,5

Harga Frame: 200000

Harga Lensa: 250000

Data tersimpan ke database!

--- NOTA TRANSAKSI HARISNA OPTICAL ---

Alamat Toko : Jl. Pahlawan, Kp. Jawa I, Pariaman Tengah

Tanggal : 24 December 2025

Pelanggan : ICHA

Resep Mata : L -3.5 , R -4,5

Total Bayar : Rp 499,500.00

---

```

Menu Utama:
1. Tambah Transaksi
2. Keluar

Pilih menu: 2

Terima kasih telah menggunakan sistem Harisna Optical.

PS E:\kuliah\Semester 3\PBO\Tugas Besar PBO 2025 Luthfi Harisna
Mufti_2411523019_PBO C>

```

- B. Input Data: Saya telah menguji dengan memasukkan nama pelanggan "Budi" , "Rusdi", "Ana" dan "Icha", sistem berhasil memprosesnya tanpa kendala.
- C. Output Nota: Program mencetak nota yang rapi dengan detail alamat di Jl. Pahlawan, Pariaman Tengah, dan total bayar yang sudah diformat ke angka desimal yang pas.
- D. Verifikasi Database: Saat saya mengecek di phpMyAdmin, data yang diinput tadi sudah masuk ke tabel transaksi di database db\_harisna\_optical.

The screenshot shows the phpMyAdmin interface with the 'transaksi' table selected. The table contains the following data:

	Ubah	Salin	Hapus	id	nama	resep	total	tanggal
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	5	BUDI	L-2, R-5	355200.0000000006	2025-12-24 15:59:16
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	6	RUSDI	L-2,5, R-3,4	344100.0000000006	2025-12-24 16:00:10
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	7	ANA	L-1,7, R-2,6	421800.0000000006	2025-12-24 16:01:23
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	8	ICHA	L-3,5, R-4,5	499500.0000000006	2025-12-24 16:02:02

## **BAB IV**

### **PENUTUP**

#### **4.1 Kesimpulan**

Berdasarkan proses pengembangan dan pengujian yang telah dilakukan pada Sistem Manajemen Harisna Optical, dapat diambil beberapa kesimpulan sebagai berikut:

- A. Penerapan Pemrograman Berorientasi Objek (PBO) melalui konsep *Inheritance*, *Interface*, dan *Encapsulation* terbukti dapat menciptakan struktur kode yang lebih rapi, terorganisir, dan mudah untuk dikembangkan di masa depan .
- B. Penggunaan JDBC (Java Database Connectivity) berhasil mengatasi masalah penyimpanan data yang bersifat *volatile*, sehingga data resep dan transaksi pelanggan di Harisna Optical dapat tersimpan secara permanen dan aman di dalam database MySQL .
- C. Fitur otomatisasi perhitungan matematika dan manipulasi data terbukti sangat efektif dalam meminimalisir kesalahan manusia (*human error*) saat proses pembuatan nota transaksi berlangsung.
- D. Penerapan Exception Handling seperti try-catch pada input harga dan koneksi database memberikan perlindungan bagi sistem agar tetap berjalan stabil meskipun terjadi kesalahan input atau kendala teknis pada server .

#### **4.2 Saran**

Untuk pengembangan sistem Harisna Optical ke depannya agar menjadi lebih sempurna, saya menyarankan beberapa hal berikut:

- A. Menambahkan fitur Keamanan Login dan CAPTCHA untuk memvalidasi pengguna sebelum mengakses data transaksi.
- B. Mengembangkan fitur Update dan Delete yang lebih dinamis untuk memudahkan admin dalam mengelola stok lensa atau bingkai kacamata secara *real-time* .

- C. Membangun antarmuka berbasis GUI (Graphical User Interface) agar operasional di toko Harisna Optical menjadi lebih ramah pengguna (*user-friendly*) dibandingkan tampilan konsol.
- D. Menambahkan fitur Laporan Keuangan bulanan otomatis yang dapat ditarik langsung dari database untuk memudahkan pemilik toko dalam memantau perkembangan usahanya.

## **DAFTAR PUSTAKA**

Departemen Sistem Informasi, Fakultas Teknologi Informasi, Universitas Andalas. 2024. *Modul Praktikum Pemrograman Berorientasi Objek: Pertemuan 03 - Inheritance (Pewarisan)*.

Departemen Sistem Informasi, Fakultas Teknologi Informasi, Universitas Andalas. 2024. *Modul Praktikum Pemrograman Berorientasi Objek: Pertemuan 04 - Interface dan Polimorfisme*.

Departemen Sistem Informasi, Fakultas Teknologi Informasi, Universitas Andalas. 2024. *Modul Praktikum Pemrograman Berorientasi Objek: Pertemuan 05 - Collection Framework dan Exception Handling*.

Departemen Sistem Informasi, Fakultas Teknologi Informasi, Universitas Andalas. 2024. *Modul Praktikum Pemrograman Berorientasi Objek: Pertemuan 06 - Java Database Connectivity (JDBC)*.

Departemen Sistem Informasi, Fakultas Teknologi Informasi, Universitas Andalas. 2024. *Modul Praktikum Pemrograman Berorientasi Objek: Pertemuan 07 - Operasi CRUD (Create, Read, Update, Delete) dengan JDBC*.