

LAPORAN AKHIR PEMROGRAMAN DASAR 2
PROYEK KASIR PADA TOKO APOTEK



Oleh:

Maulidya (2310131120003)

Tanggal Penyusunan:

17-Desember-2024

Dosen Pengampu:

Nuruddin Wiranda, S.Kom., M.Cs

Dr. Harja Santana Purba M. Kom

Ihdalhubbi Maulida, M. Kom

UNIVERSITAS LAMBUNG MANGKURAT
FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN
PROGRAM STUDI PENDIDIKAN KOMPUTER

2024

DAFTAR ISI

DAFTAR ISI	2
BAB I PENDAHULUAN	3
1.1. Latar Belakang.....	3
1.2. Tujuan Proyek	3
1.3. Ruang Lingkup.....	3
BAB II STUDI LITERATUR.....	4
2.1. System kasir	4
2.2. Java sebagai bahasa pemrograman	4
2.3. Mysql sebagai database	4
2.4. Referensi dari proyek serupa	5
BAB 3 IMPLEMENTASI PROYEK	6
3.1. Frame Login.....	6
3.2. Frame Register	8
3.3. Frame Menu.....	10
3.4. Frame Stok Barang	11
3.5. Frame Pembayaran	15
3.6. Struktur Database	18
BAB IV KESIMPULAN DAN SARAN	19
Kesimpulan	19
Saran	19

BAB I

PENDAHULUAN

1.1.Latar Belakang

Dalam era digital saat ini, teknologi informasi telah menjadi elemen penting dalam berbagai aspek kehidupan, termasuk di bidang kesehatan. Salah satu sektor yang sangat memanfaatkan teknologi ini adalah apotek. Proses transaksi di apotek sering kali melibatkan berbagai data, seperti nama obat, jumlah stok, harga, serta rincian transaksi pelanggan. Pengelolaan data yang dilakukan secara manual sering kali memunculkan berbagai kendala, seperti kesalahan pencatatan, ketidakakuratan data, dan efisiensi waktu yang rendah.

Sistem kasir berbasis aplikasi merupakan salah satu solusi untuk mengatasi permasalahan tersebut. Dengan memanfaatkan teknologi, proses pencatatan transaksi dapat dilakukan secara otomatis, cepat, dan akurat. Selain itu, sistem ini juga membantu dalam pengelolaan stok obat dan pembuatan laporan penjualan, sehingga operasional apotek menjadi lebih efisien.

1.2.Tujuan Proyek

Proyek ini bertujuan untuk mengembangkan sebuah sistem kasir apotek berbasis aplikasi desktop menggunakan Java dengan IDE NetBeans. Sistem ini dirancang untuk:

1. Mempermudah proses transaksi penjualan di apotek.
2. Meningkatkan akurasi data transaksi dan stok obat.
3. Menghasilkan laporan penjualan harian atau bulanan secara otomatis.
4. Mengurangi risiko kesalahan pencatatan yang sering terjadi pada sistem manual.

1.3.Ruang Lingkup

Proyek ini mencakup pengembangan fitur utama berikut:

1. **Manajemen Data Obat:** Meliputi penambahan, pengeditan, penghapusan, dan pencarian data obat.
2. **Transaksi Penjualan:** Fitur untuk mencatat transaksi pembelian obat oleh pelanggan.
3. **Laporan Penjualan:** Penyediaan laporan penjualan harian dan bulanan dalam format yang mudah dipahami.
4. **Pengelolaan Data Pengguna:** Sistem login untuk memastikan akses hanya dilakukan oleh pengguna yang berwenang.

Dengan ruang lingkup ini, aplikasi diharapkan dapat menjadi solusi praktis dan efisien bagi apotek dalam mengelola proses transaksi dan data.

BAB II

STUDI LITERATUR

2.1. System kasir

Sistem kasir merupakan perangkat lunak yang digunakan untuk membantu pencatatan transaksi penjualan di berbagai bisnis, termasuk apotek. Sistem ini dirancang untuk menggantikan metode pencatatan manual yang cenderung lambat, kurang efisien, dan rentan terhadap kesalahan. Dalam konteks apotek, sistem kasir tidak hanya mencatat transaksi penjualan tetapi juga mengelola stok obat, memastikan ketersediaan barang, dan menghasilkan laporan penjualan secara otomatis.

Penelitian sebelumnya menunjukkan bahwa penggunaan sistem kasir berbasis komputer dapat meningkatkan efisiensi kerja hingga 30% dan mengurangi kesalahan pencatatan hingga 50% (Sumber: [referensi artikel/jurnal, jika ada]).

2.2. Java sebagai bahasa pemrograman

Java adalah salah satu bahasa pemrograman berorientasi objek yang populer karena portabilitasnya dan kemampuannya berjalan di berbagai platform. Dalam pengembangan aplikasi kasir ini, Java digunakan karena memiliki beberapa keunggulan:

- **Ketersediaan Library:** Java menyediakan pustaka yang lengkap untuk kebutuhan pengembangan aplikasi desktop, seperti Java Swing atau JavaFX untuk antarmuka pengguna.
- **Koneksi Database yang Mudah:** Dengan dukungan JDBC (Java Database Connectivity), Java mempermudah integrasi dengan database seperti MySQL.
- **Keamanan:** Java memiliki fitur keamanan bawaan yang mendukung pengelolaan data sensitif seperti informasi transaksi.

Dalam proyek ini, NetBeans digunakan sebagai Integrated Development Environment (IDE) untuk mendukung proses pengkodean dan debugging aplikasi.

2.3. Mysql sebagai database

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang digunakan secara luas karena kestabilannya dan sifat open-source. MySQL dipilih dalam proyek ini karena memiliki fitur berikut:

- **Manajemen Data yang Efisien:** MySQL mampu mengelola data dalam jumlah besar dengan kinerja tinggi, sehingga cocok untuk aplikasi kasir apotek.
- **Kompatibilitas dengan Java:** MySQL dapat dengan mudah diintegrasikan menggunakan JDBC di Java.
- **Keamanan Data:** Mendukung autentikasi pengguna dan enkripsi untuk melindungi data sensitif seperti transaksi penjualan.

Database ini digunakan untuk menyimpan informasi seperti data obat, pengguna, dan transaksi penjualan.

2.4. Referensi dari proyek serupa

Studi proyek serupa menunjukkan pentingnya fitur berikut dalam aplikasi kasir apotek:

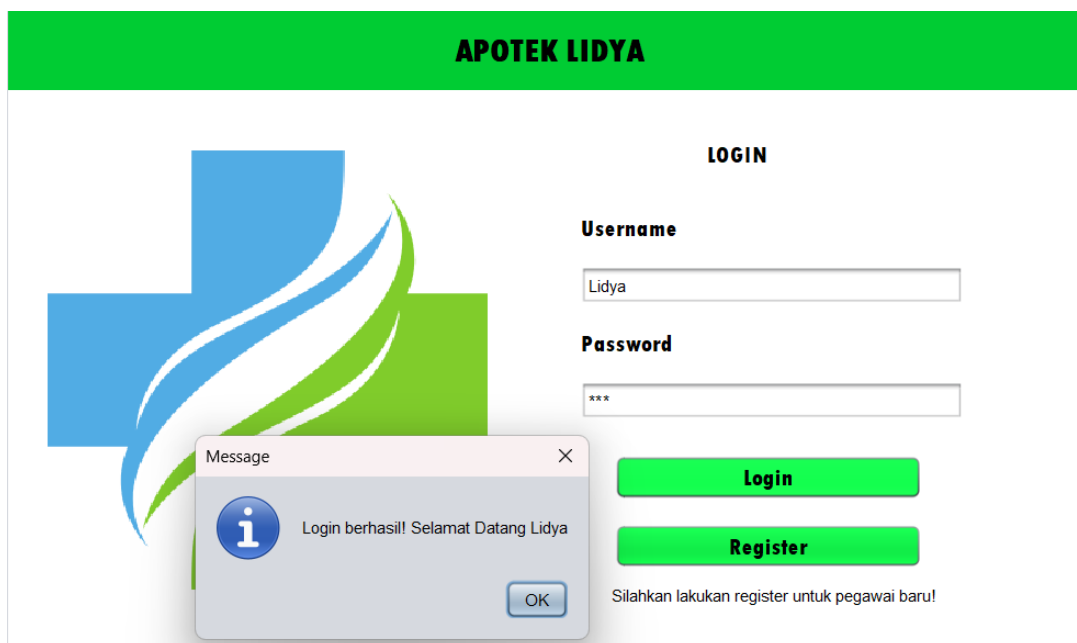
- **Manajemen Stok Obat:** Sistem harus mampu mencatat ketersediaan obat dan memberikan peringatan saat stok menipis.
- **Integrasi Laporan Penjualan:** Laporan harian dan bulanan diperlukan untuk memantau performa penjualan.
- **Kemudahan Penggunaan:** Antarmuka pengguna yang sederhana dan intuitif memudahkan pengguna non-teknis dalam mengoperasikan sistem.

BAB 3

IMPLEMENTASI PROYEK

3.1.Frame Login

Berikut adalah tampilan hasil dari Frame Login:



The screenshot displays the login interface for 'APOTEK LIDYA'. On the left is a large logo featuring a blue cross and green and white swooshes. The login form on the right includes fields for 'Username' (containing 'Lidya') and 'Password' (masked with '***'). Below these are 'Login' and 'Register' buttons. A message box in the foreground states 'Login berhasil! Selamat Datang Lidya' with an 'OK' button. A note at the bottom right says 'Silahkan lakukan register untuk pegawai baru!'.

Ini adalah tampilan dimana seseorang berhasil login pada framenya.berikut adalah tampilan awalnya:



This screenshot shows the same login interface as above, but in its initial state. The 'Username' and 'Password' fields are empty. The 'Login' and 'Register' buttons are visible, along with the note 'Silahkan lakukan register untuk pegawai baru!' at the bottom right. The logo and header remain the same.

Dalam frame login saya memasukkan kode program berikut yang terhubung dengan database apotek pada table pegawai.berikut kode program yang saya masukkan:

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package apotek;
import java.sql.Connection;
import java.sql.SQLException;
import javax.swing.JOptionPane;
/**
 *
 * @author MyBook Hype
 */
public class FormLogin extends javax.swing.JFrame {

    /**
     * Creates new form FormLogin
     */
    public FormLogin() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try {
            String username = inpUser.getText();
            String password = inpPass.getText();

            String sql = "SELECT * FROM pegawai WHERE username = ? AND password = ?";
            java.sql.Connection conn = (Connection)ConfigDB.config();
            java.sql.PreparedStatement pst = conn.prepareStatement(sql);
            pst.setString(1, username);
            pst.setString(2, password);

            java.sql.ResultSet res = pst.executeQuery();

            if (res.next()){

                JOptionPane.showMessageDialog(
                    this, "Login berhasil! Selamat Datang " + username);
                FormMenu formMenu = new FormMenu(username);
                formMenu.setVisible(true);
                this.dispose();

            }else {
                JOptionPane.showMessageDialog(this,"Username atau password salah");
            }

        } catch (SQLException e){
            JOptionPane.showMessageDialog(this, "Error: " + e.getMessage());
        }
    }

    private void btnRegisterActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        FormRegister formRegister = new FormRegister();
        formRegister.setVisible(true);
        this.dispose();
    }
}

```

Setelah user melakukan register maka data yang diinputkan akan masuk kedalam table pegawai sehingga user bisa login dengan lancer. Berikut tampilan table pada mysqlnya:

```
MariaDB [apotek]> select * from pegawai;
+-----+-----+-----+-----+
| username | password | no_telepon | alamat |
+-----+-----+-----+-----+
| ani      | 010203  | 081232987677 | Sungai |
| Lidya    | 123     | 085350084833 | amuntai |
| Rizfa    | 456     | 08312048231  | barabai |
+-----+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [apotek]> desc pegawai;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| username   | varchar(20)   | NO   | PRI | NULL    |       |
| password   | varchar(20)   | NO   |     | NULL    |       |
| no_telepon | varchar(20)   | NO   |     | NULL    |       |
| alamat     | varchar(50)   | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.029 sec)
```

3.2.Frame Register

Berikut adalah tampilan hasil dari frame Register:

APOTEK LIDYA

REGISTER



Masukkan username

Masukkan password

Masukkan nomor telpon

Masukkan alamat

Register

Silahkan lakukan register untuk pegawai baru!

Message X

Berhasil daftar!

OK

Dalam frame register saya memasukkan kode program berikut yang terhubung dengan database apotek pada table pegawai. Berikut kode program yang saya masukkan:

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java to edit this template
 */
package apotek;

/**
 *
 * @author MyBook Hype
 */
import java.sql.Connection;
import java.sql.SQLException;
import javax.swing.JOptionPane;
public class FormRegister extends javax.swing.JFrame {

    /**
     * Creates new form FormRegister
     */
    public FormRegister() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void btnRegisActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try{
            // mengambil input dari pengguna
            String username = inpUser.getText();
            String password = inpPass.getText();
            String no_telepon = inpTelepon.getText();
            String alamat = inpAlamat.getText();

            FormLogin formLogin = new FormLogin();

            String sql = "INSERT INTO pegawai VALUES ('" + username + "', '" + password + "', '"
                + no_telepon + "', '" + alamat + "')";
            java.sql.Connection conn = (Connection)ConfigDB.config();
            java.sql.PreparedStatement pstm = conn.prepareStatement(sql);

            int rowsInserted = pstm.executeUpdate();

            if (rowsInserted > 0){
                JOptionPane.showMessageDialog(null, "Berhasil daftar!");
                formLogin.setVisible(true);
                this.dispose();
            } else {
                JOptionPane.showMessageDialog(null, "Gagal daftar! Silahkan coba lagi.");
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
    }
}

```

Setelah user melakukan register maka data yang diinputkan akan masuk kedalam table pegawai sehingga user bisa login dengan lancar. Berikut tampilan table pada mysqlnya:

```
MariaDB [apotek]> select * from pegawai;
```

username	password	no_telepon	alamat
ani	010203	081232987677	Sungai
Indah	0405	083829871654	Kidaung Permai
Lidya	123	085350084833	amuntai
Rizfa	456	08312048231	barabai

```
4 rows in set (0.001 sec)
```

3.3.Frame Menu



Pada frame menu kita menampilkan pilihan untuk melihat stok barang dan melakukan pembayaran. Berikut adalah kode program yang saya masukkan:

```
private void btnStokActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    StokObat formStokObat = new StokObat();
    formStokObat.setVisible(true);
    this.dispose();
}

private void btnbayarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    FormPembayaran formPembayaran = new FormPembayaran(username);
    formPembayaran.setVisible(true);
    this.dispose();
}
```

Frame ini tidak memiliki hubungan dengan mysql karena ini digunakan untuk penghubung antara frame satu ke frame yang lainnya.

3.4.Frame Stok Barang

Berikut adalah tampilan dari Stok obat yang saya miliki:

Stok Obat

Menu Keluar

Id Obat :

Nama Obat :

Jenis Obat :

Harga :

Stok :

Diskon :

Tambah

Ubah

Hapus

Batal

No.	Nama Obat	Jenis Obat	Harga	Stok	Diskon
1	Paracetamol	Analgesik	5000	100	10
2	Amoxicillin	Antibiotik	10000	50	5
3	Ibuprofen	Anti-inflamasi	7000	75	8
4	Cetirizine	Antihistamin	4000	120	12
5	Metformin	Antidiabetik	15000	80	0
6	Salbutamol	Bronkodilator	6000	80	5
7	Aspirin	Antikoagulan	3000	150	7
8	Omeprazole	Antasida	12000	70	10
9	Ciprofloxacin	Antibiotik	11000	40	8
10	Loperamide	Antidiare	3500	130	5
11	Dexamethas...	Anti-inflamasi	8000	90	6
12	Simvastatin	Antikolesterol	9000	55	4
13	Ranitidine	Antasida	5000	85	9
14	Mefenamic A...	Analgesik	6500	95	3
15	Prednisone	Kortikosteroid	7000	60	0

Disini saya meletakkan CRUD dimana ada tombol Tambah, ubah,Hapus, dan batal.

Ini adalah tampilan jika kita menambahkan data:

Stok Obat

Menu Keluar

Id Obat :

Nama Obat :

Jenis Obat :

Harga :

Stok :

Diskon :

Tambah

Ubah

Hapus

Batal

Message

Data berhasil ditambahkan!

OK

Title 4

Title 5

Title 6

Ini adalah tampilan Ketika kita menghapus data obat:

Stok Obat

Menu Keluar

Id Obat :

Nama Obat :

Jenis Obat :

Harga :

Stok :

Diskon :

Tambah
Ubah
Hapus
Batal

Message

Data berhasil dihapus!

OK

			Harga	Stok	Diskon
			5000	100	10
			10000	50	5
			7000	75	8
			4000	120	12
			15000	60	0
			6000	80	5
			3000	150	7
8	Omeprazole	Antasida	12000	70	10
9	Ciprofloxacin	Antibiotik	11000	40	8
10	Loperamide	Antidiare	3500	130	5
11	Dexamethasone	Anti-inflamasi	8000	90	6
12	Simvastatin	Antikolesterol	9000	55	4
13	Ranitidine	Antasida	5000	85	9
14	Mefenamic Acid	Analgesik	6500	95	3
15	Prednisone	Kortikosteroid	7000	60	0

Kita juga bisa mengubah isi table atau data pada table:

Stok Obat

Menu Keluar

Id Obat :

Nama Obat :

Jenis Obat :

Harga :

Stok :

Diskon :

Tambah
Ubah
Hapus
Batal

Message

Data berhasil diubah!

OK

			Harga	Stok	Diskon
1			5000	100	10
2			10000	50	5
3			7000	75	8
4			4000	120	12
5			10000	60	2
6	Salbutamol	Bronkodilator	10000	80	5
7			3000	150	7
8			15000	70	2
9	Ciprofloxacin	Antibiotik	11000	40	8
10	Loperamide	Antidiare	3500	130	5
11	Dexamethasone	Anti-inflamasi	8000	90	6
12	Simvastatin	Antikolesterol	9000	55	4
13	Ranitidine	Antasida	5000	85	9
14	Mefenamic Acid	Analgesik	6500	95	3
15	Prednisone	Kortikosteroid	7000	60	0

Berikut adalah kode program yang saya masukkan dalam frame ini:

```
package apotek;

import java.sql.Connection;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author MyBook Hype
 */
public class StokObat extends javax.swing.JFrame {

    private void kosongkan_form() {
        txtIdObat.setText(null);
        txtNama.setText(null);
        txtJenis.setText(null);
        txtHarga.setText(null);
        txtStok.setText(null);
        txtDiskon.setText(null);
    }

    private void tampilkan_data() {
        DefaultTableModel model = new DefaultTableModel();
        model.addColumn("No.");
        model.addColumn("Nama Obat");
        model.addColumn("Jenis Obat");
        model.addColumn("Harga");
        model.addColumn("Stok");
        model.addColumn("Diskon");

        try {
            int no = 1;
            String sql = "SELECT id_obat, nama_obat, jenis_obat, harga, stok, diskon FROM stok_obat";
            java.sql.Connection conn = (Connection) ConfigDB.config();
            java.sql.Statement stm = conn.createStatement();
            java.sql.ResultSet res = stm.executeQuery(sql);
            while (res.next()) {
                model.addRow(new Object[]{no++, res.getString(1), res.getString(2),
                    res.getString(3), res.getString(4), res.getString(5), res.getString(6)});
            }
            tblBarang.setModel(model);
        } catch (SQLException e) {
            System.out.println("Error " + e.getMessage());
        }
    }

    private void txtMenuMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        FormMenu formMenu = new FormMenu();
        formMenu.setVisible(true);
        this.dispose();
    }

    private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try {
            // Ambil data dari form
            String id_obat = txtIdObat.getText().trim();
            String nama = txtNama.getText().trim();
            String jenis_obat = txtJenis.getText().trim();
            String hargaStr = txtHarga.getText().trim();
            String stokStr = txtStok.getText().trim();
            String diskonStr = txtDiskon.getText().trim();

            // Validasi input tidak boleh kosong
            if (id_obat.isEmpty() || nama.isEmpty() || jenis_obat.isEmpty() || hargaStr.isEmpty() || stokStr.isEmpty() || diskonStr.isEmpty()) {
                JOptionPane.showMessageDialog(null, "Semua field harus diisi!");
                return; // Hentikan proses jika ada field kosong
            }

            // Konversi String ke Integer
            int harga = Integer.parseInt(hargaStr);
            int stok = Integer.parseInt(stokStr);
            int diskon = Integer.parseInt(diskonStr);

            // Query UPDATE tanpa SET id_obat
            String sql = "UPDATE stok_obat SET nama_obat = ?, jenis_obat = ?, harga = ?, stok = ?, diskon = ? WHERE id_obat = ?";
            java.sql.Connection conn = ConfigDB.config(); // Pastikan ConfigDB.config() mengembalikan koneksi yang valid
            java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);

            // Set parameter untuk query (6 parameter)
            pstmt.setString(1, nama); // nama_obat
            pstmt.setString(2, jenis_obat); // jenis_obat
            pstmt.setInt(3, harga); // harga
            pstmt.setInt(4, stok); // stok
            pstmt.setInt(5, diskon); // diskon
            pstmt.setString(6, id_obat); // WHERE id_obat
        } catch (SQLException e) {
            System.out.println("Error " + e.getMessage());
        }
    }
}
```

```

// Eksekusi query
pstm.executeUpdate();

// Tampilkan pesan berhasil
JOptionPane.showMessageDialog(null, "Data berhasil diubah!");
tampilkan_data(); // Refresh data di tabel
kosongkan_form(); // Kosongkan form input
} catch (NumberFormatException e) {
    JOptionPane.showMessageDialog(this, "Input harus berupa angka pada kolom Harga, Stok, dan Diskon!");
} catch (SQLException e) {
    JOptionPane.showMessageDialog(this, "Gagal mengubah data: " + e.getMessage());
    e.printStackTrace();
}

private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        String id_obat = txtIdObat.getText();

        String sql = "DELETE from stok_obat WHERE id_obat = '" + id_obat + "'";
        java.sql.Connection conn = (Connection)ConfigDB.config();
        java.sql.PreparedStatement pstm = conn.prepareStatement(sql);
        pstm.execute();

        JOptionPane.showMessageDialog(null, "Data berhasil dihapus!");
        tampilkan_data();
        kosongkan_form();
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

private void tblBarangMouseClicked(java.awt.event.MouseEvent evt) {
    // Ambil baris yang diklik
    int baris = tblBarang.rowAtPoint(evt.getPoint());

    // Ambil data dari tabel berdasarkan kolom
    String id_obat = tblBarang.getValueAt(baris, 1).toString();
    txtIdObat.setText(id_obat);

    String nama = tblBarang.getValueAt(baris, 2).toString();
    txtNama.setText(nama);

    String jenis_obat = tblBarang.getValueAt(baris, 3).toString();
    txtJenis.setText(jenis_obat); // Mengisi jenis obat ke txtJenis

    // Konversi nilai numerik (harga, stok, diskon)
    int harga = Integer.parseInt(tblBarang.getValueAt(baris, 4).toString());
    txtHarga.setText(String.valueOf(harga));

    int stok = Integer.parseInt(tblBarang.getValueAt(baris, 5).toString());
    txtStok.setText(String.valueOf(stok));

    double diskon = Double.parseDouble(tblBarang.getValueAt(baris, 6).toString());
    txtDiskon.setText(String.valueOf(diskon));
}

```

Table yang dimasukkan akan masuk kedalam table stok_obat:

```
MariaDB [apotek]> select * from stok_obat;
```

Id_obat	Nama_obat	Jenis_obat	Harga	Stok	Diskon
1	Paracetamol	Analgesik	5000	100	10
2	Amoxicillin	Antibiotik	10000	50	5
3	Ibuprofen	Anti-inflamasi	7000	75	8
4	Cetirizine	Antihistamin	4000	120	12
5	Metformin	Antidiabetik	15000	60	0
6	Salbutamol	Bronkodilator	6000	80	5
7	Aspirin	Antikoagulan	3000	150	7
8	Omeprazole	Antasida	12000	70	10
9	Ciprofloxacin	Antibiotik	11000	40	8
10	Loperamide	Antidiare	3500	130	5
11	Dexamethasone	Anti-inflamasi	8000	90	6
12	Simvastatin	Antikolesterol	9000	55	4
13	Ranitidine	Antasida	5000	85	9
14	Mefenamic Acid	Analgesik	6500	95	3
15	Prednisone	Kortikosteroid	7000	60	0
16	Vitamin C	Suplemen	2000	200	15
17	Azithromycin	Antibiotik	13000	45	10
18	Hydrochlorothiazide	Diuretik	10000	50	5
19	Captopril	Antihipertensi	6000	110	8
20	Furosemide	Diuretik	7500	65	6
21	Betadine	cair	15000	12	2
22	Hansaplas	benda	20000	15	2
23	vitacimin	tablet	2000	20	2
24	bodrex	tablet	3000	19	3

24 rows in set (0.001 sec)

3.5.Frame Pembayaran

Berikut adalah hasil dari pembayaran:

Pembayaran

Nama pegawai: Lidya

No.	Nama Obat	Harga Obat	Stok Obat
6	Paracetamol	5000	87
7	Amoxicillin	7000	48
8	Betadine	15000	71
9	Antangin	3000	192
10	Vitamin C	12000	137
16	Betadine	15000	71
17	Vitamin C	12000	150
18	Amoxicillin	7000	48
19	Antangin	3000	196
20	Paracetamol	5000	87
21	Eucalyptus Oil	18000	40

Nama Obat : (nama) Jumlah : Total Harga : Rp.0

Stok : (Stok) Kode diskon : Bayar :

Harga : Rp.0 Potongan Harga : Rp.0 Kembalian : Rp.0

Batal

Hitung Harga

Beli

Struk

Tampilan jika data berhasil dibeli:

Pembayaran

Nama pegawai: litya

No.	Nama Obat	Harga Obat	Stok Obat
6	Paracetamol	5000	87
7	Amoxicillin	7000	48
8	Betadine	15000	71
9	Antangin	3000	192
10	Vitamin C	12000	137
16	Betadine	15000	71
17	Vitamin C	12000	150
18	Amoxicillin	7000	47
19	Antangin	3000	196
20	Paracetamol	5000	87
21	Eucalyptus Oil	18000	40

Nama Obat : (nama) Jumlah :

Stok :

Harga :

Total Harga : Rp. 0

 Bayar : 0

Rp. 0 Kembalian : Rp. 0

Hitung Harga

Beli

Struk

Tampilan jika kode diskon dimasukkan:

Pembayaran

Nama pegawai: litya

No.	Nama Obat	Harga Obat	Stok Obat
6	Paracetamol	5000	87
7	Amoxicillin	7000	48
8	Betadine	15000	71
9	Antangin	3000	192
10	Vitamin C	12000	137
16	Betadine	15000	71
17	Vitamin C	12000	150
18	Amoxicillin	7000	48
19	Antangin	3000	196
20	Paracetamol	5000	87
21	Eucalyptus Oil	18000	40

Nama Obat : Amoxicillin Jumlah :

Stok : 48 Kode diskon : sehat

Harga : Rp. 7000.0 Potongan Harga : RP. 1400.0

Total Harga : Rp. 5600.0

Bayar :

Kembalian : Rp. 4400.0

Batal

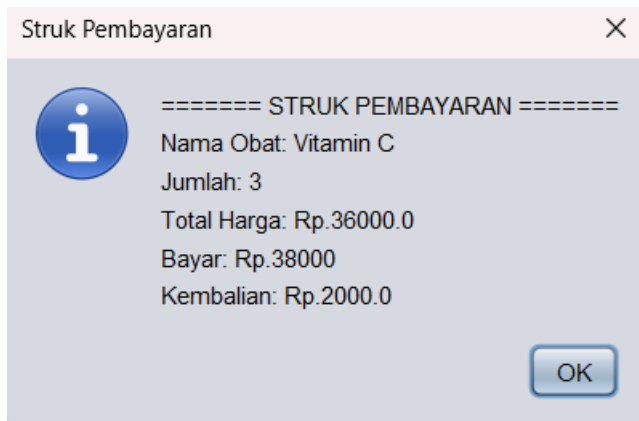
Hitung Harga

Beli

Struk

Stok akan berkurang secara otomatis.

Tampilan struk:



Barang yang sudah dibeli maka akan tampil di table penjualan:

```
MariaDB [apotek]> select * from penjualan;
```

id_obat	nama_obat	jenis_obat	harga_obat	stok_barang	diskon
6	Paracetamol	Tablet	5000	87	10%
7	Amoxicillin	Kapsul	7000	48	5%
8	Betadine	Cair	15000	71	15%
9	Antangin	Sachet	3000	196	5%
10	Vitamin C	Tablet	12000	150	10%

5 rows in set (0.001 sec)

3.6.Struktur Database

MariaDB [apotek]> desc transaksi;

Field	Type	Null	Key	Default	Extra
id_transaksi	int(11)	NO	PRI	NULL	auto_increment
id_obat	int(11)	NO	MUL	NULL	
username	varchar(20)	NO	MUL	NULL	
jumlah	int(11)	NO		NULL	
tanggal_transaksi	datetime	YES		current_timestamp()	

5 rows in set (0.037 sec)

MariaDB [apotek]> desc pegawai;

Field	Type	Null	Key	Default	Extra
username	varchar(20)	NO	PRI	NULL	
password	varchar(20)	NO		NULL	
no_telepon	varchar(20)	NO		NULL	
alamat	varchar(50)	NO		NULL	

4 rows in set (0.038 sec)

MariaDB [apotek]> desc stok_obat;

Field	Type	Null	Key	Default	Extra
Id_obat	int(20)	NO	PRI	NULL	
Nama_obat	varchar(50)	NO		NULL	
Jenis_obat	varchar(50)	NO		NULL	
Harga	int(20)	NO		NULL	
Stok	int(20)	NO		NULL	
Diskon	varchar(20)	NO		NULL	

6 rows in set (0.034 sec)

MariaDB [apotek]> desc penjualan;

Field	Type	Null	Key	Default	Extra
id_obat	int(11)	NO	PRI	NULL	auto_increment
nama_obat	varchar(100)	NO		NULL	
jenis_obat	varchar(100)	NO		NULL	
harga_obat	int(100)	NO		NULL	
stok_barang	int(100)	NO		NULL	
diskon	varchar(20)	YES		NULL	

6 rows in set (0.035 sec)

BAB IV

KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan pengembangan aplikasi pembayaran pada sistem apotek, dapat disimpulkan bahwa aplikasi ini berhasil menyediakan fitur pembayaran yang memudahkan transaksi antara pelanggan dan apotek. Fitur utama seperti pemilihan obat, perhitungan total harga, penerapan potongan harga melalui kode promo, dan perhitungan kembalian telah diimplementasikan dengan baik. Selain itu, fitur validasi stok obat juga berhasil diterapkan sehingga transaksi hanya dapat dilakukan jika stok mencukupi. Aplikasi ini terintegrasi dengan database untuk menampilkan dan memperbarui data stok obat secara real-time, memberikan transparansi dalam setiap transaksi. Dengan tampilan antarmuka yang sederhana dan intuitif, aplikasi ini memudahkan pegawai apotek dalam melayani pelanggan dengan cepat dan efisien. Selain itu, adanya fitur kode diskon turut meningkatkan fleksibilitas dalam memberikan promosi yang dapat meningkatkan kepuasan pelanggan.

Saran

Untuk pengembangan lebih lanjut, beberapa saran dapat diberikan agar aplikasi ini lebih optimal dan memenuhi kebutuhan apotek secara menyeluruh. Pertama, perlu ditingkatkan validasi input pengguna agar sistem dapat mencegah kesalahan input seperti nilai negatif, format yang tidak sesuai, atau kolom kosong. Kedua, disarankan untuk menambahkan fitur cetak struk dalam format PDF atau cetak langsung agar mempermudah dokumentasi transaksi. Ketiga, keamanan database perlu diperkuat dengan penerapan mekanisme proteksi seperti penggunaan PreparedStatement untuk mencegah serangan SQL Injection. Selanjutnya, fitur notifikasi stok rendah dapat diintegrasikan untuk memberikan peringatan ketika stok obat mendekati habis, sehingga apotek dapat segera melakukan restock. Selain itu, pengembangan fitur laporan transaksi harian, mingguan, dan bulanan sangat diperlukan untuk membantu manajemen dalam menganalisis penjualan dan pengelolaan stok obat. Terakhir, pembaruan desain antarmuka menggunakan teknologi seperti JavaFX atau library modern lainnya juga dapat dilakukan untuk meningkatkan kenyamanan pengguna dan tampilan aplikasi yang lebih profesional. Dengan penerapan saran-saran ini, aplikasi pembayaran diharapkan dapat lebih efisien, aman, dan mendukung operasional apotek secara maksimal.

