

**LAPORAN HASIL TUGAS BESAR UAS PBO  
MEMBUAT APLIKASI DENGAN  
JAVA APACHE NET BEANS  
KONEKSI KE DATABASE MYSQL**

*Aplikasi Apotek*



**NAMA : HERMAN JAYADI  
NIM : TI 19220014**

**DOSEN :**

**Bapak Saikin S.Kom., M.Kom**

**TEKNIK INFORMATIKA STMIK  
LOMBOK**

**KATA PENGANTAR**

Alhamdulillahirobbil'alamiin, puji syukur penulis ucapkan kehadiran Allah SWT, yang telah memberikan limpahan karunia dan kasih sayang-Nya. Salawat dan salam selalu dilimpahkan Allah SWT kepada junjungan kita Nabi Muhammad SAW yang membawa umat manusia dari kehidupan jahiliyah menuju alam yang penuh dalam cahaya ilmu pengetahuan, aqidah yang baik dan berakhlak mulia.

Laporan ini disusun untuk memenuhi tugas Akhir Semester 3 Mata Kuliah Pemrograman Berorientasi Objek. Penulis menyadari bahwa tanpa bimbingan dan dorongan dari semua pihak, maka pembuatan makalah ini tidak akan lancar. Maka penulis ucapkan terima kasih kepada Bapak Saikin S.Kom.,M.Kom selaku dosen pengajar dan memberikan pengarahan pada mata kuliah ini. Semoga Allah SWT membalas semua jasa baik tersebut dan menjadi catatan kemuliaan disisi-Nya. Aamiin...

Penulis menyadari sepenuhnya bahwa pembuatan makalah ini masih jauh dari sempurna, sehingga masih terdapat banyak kekurangan. Karena itu penulis mengharapkan kritik dan saranyang bersifat membangun demi perbaikan dimasa yang akan datang. Akhir kata, penulis berharap dapat bermanfaat khususnya bagi penulis dan pembaca pada umumnya.

Praya, Januari 2024

Herman jayadi

## DAFTAR ISI

<b>KATA PENGANTAR</b>	<b>i</b>
<b>DAFTAR ISI</b>	<b>i</b>
<b>BAB I PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang	1
1.2 Alasan membuat aplikasi apotek	1
<b>BAB II PEMBAHASAN</b>	<b>2</b>
2.1 Struktur Aplikasi & Pembahasan Fungsi Komponen Aplikasi beserta Screenshot	2
<b>BAB III PENUTUP</b>	<b>3</b>

## BAB I PENDAHULUAN

## 1.1 . Latar Belakang

Dalam era digital yang terus berkembang, sistem kesehatan membutuhkan inovasi dalam manajemen informasi dan pelayanan kepada pasien. Salah satu aspek kritis dari sistem kesehatan adalah manajemen obat di apotek. Saat ini, masih banyak apotek yang menggunakan proses manual dalam pengelolaan stok obat, pelayanan kepada pasien, dan integrasi informasi kesehatan.

Proses manual sering kali menghasilkan beberapa masalah, termasuk kesalahan dalam pengelolaan stok obat, kekurangan koordinasi antara apotek dan tenaga medis lainnya, serta kurangnya transparansi dalam informasi obat untuk pasien. Selain itu, kebutuhan akan pemantauan kepatuhan pasien terhadap pengobatan juga menjadi tantangan yang perlu diatasi.

Untuk mengatasi kendala-kendala tersebut, pengembangan aplikasi apotik yang memanfaatkan teknologi seperti Apache NetBeans dianggap sebagai solusi yang potensial. Aplikasi ini diharapkan dapat meningkatkan efisiensi operasional apotek, memberikan akses cepat dan mudah ke informasi obat, meningkatkan koordinasi antarpetugas kesehatan, serta membantu dalam pemantauan kepatuhan pasien.

Penggunaan Apache NetBeans sebagai platform pengembangan dipilih karena fleksibilitasnya yang tinggi dalam mendukung pengembangan aplikasi berbasis Java. Dengan memanfaatkan Apache NetBeans, pengembang dapat dengan mudah membuat aplikasi yang responsif, mudah dikembangkan, dan dapat diintegrasikan dengan sistem lainnya.

Melalui pengembangan aplikasi apotik ini, diharapkan dapat memberikan kontribusi positif terhadap efisiensi dan efektivitas pelayanan di apotek, meningkatkan pengalaman pasien, serta mendukung pengelolaan informasi kesehatan secara terintegrasi. Dengan demikian, masalah-masalah yang dihadapi dalam manajemen apotek dapat diatasi, dan pelayanan kesehatan dapat terus berkembang sesuai dengan tuntutan era digital.

---

## 1.2 Alasan Membuat Aplikasi Apotek

1. **Kemudahan Akses Informasi Obat:** Aplikasi apotik memungkinkan pengguna untuk dengan mudah mengakses informasi tentang obat-obatan, termasuk dosis, efek samping, kontraindikasi, dan petunjuk penggunaan. Ini membantu pengguna mendapatkan informasi yang akurat dan tepat waktu sehubungan dengan kesehatan mereka.
2. **Pemesanan Obat secara Online:** Pengguna dapat memesan obat secara online melalui aplikasi apotik, yang memungkinkan mereka untuk menghemat waktu dan tenaga. Proses pemesanan yang mudah dan cepat juga dapat meningkatkan kepatuhan pengguna terhadap pengobatan.
3. **Pantauan Kesehatan Secara Elektronik:** Aplikasi apotik dapat membantu pengguna memantau kondisi kesehatan mereka secara elektronik, termasuk pengingat waktu minum obat, pemantauan kadar gula darah, atau pemantauan tekanan darah. Hal ini mempermudah pemantauan kondisi kesehatan secara rutin.
4. **Integrasi dengan Rekam Medis Elektronik (EHR):** Integrasi aplikasi apotik dengan sistem rekam medis elektronik memungkinkan apoteker dan profesional kesehatan lainnya untuk berbagi informasi pasien dengan lebih efisien. Ini meningkatkan koordinasi perawatan pasien dan mengurangi risiko kesalahan obat.
5. **Pengelolaan Stok dan Keberlanjutan:** Apoteker dapat menggunakan aplikasi untuk mengelola stok obat dan memantau keberlanjutan persediaan dengan lebih efektif. Hal ini membantu menghindari kekurangan atau kelebihan persediaan, meningkatkan efisiensi operasional, dan mengurangi risiko kehilangan obat.
6. **Pemberian Informasi Obat kepada Pasien:** Aplikasi apotik dapat menyediakan informasi obat kepada pasien dalam bentuk yang mudah dimengerti, membantu meningkatkan pemahaman mereka tentang pengobatan yang sedang mereka terima.
7. **Penghematan Waktu dan Biaya:** Dengan memungkinkan pemesanan online dan akses cepat ke informasi obat, aplikasi apotik dapat menghemat waktu dan biaya baik bagi pengguna maupun pihak apotik. Ini juga dapat mengurangi kerumitan administratif dan meningkatkan efisiensi.
8. **Pemantauan Kepatuhan Pasien:** Aplikasi apotik dapat memberikan pengingat kepada pasien tentang waktu minum obat dan instruksi penggunaan, sehingga meningkatkan kepatuhan pasien terhadap pengobatan yang diresepkan.

## BAB II PEMBAHASAN

### 1. Seteruktur aplikasi

- id
- nama obat
- katagori
- kode obat
- baru
- tambah
- update
- delate
- view

### 2. Pembahasan

#### • ID

Tombol ID sering digunakan untuk merujuk pada catatan atau entitas tertentu dalam data. Dengan menggunakan ID, sistem dapat dengan mudah mencari mengakses data tanpa harus menginput secara manual. Ini juga dapat memastikan bahwa tidak ada duplikat atau tumpang tindih dalam identifikasi.

#### • NAMA OBAT

Inputan nama obat di aplikasi degan tujuan agar mengetahui nama obat.

The screenshot shows the phpMyAdmin interface for a database named 'apotik'. The 'menu\_obat' table is selected, and its structure and data are displayed. The table has four columns: 'id', 'nama\_obat', 'katagori', and 'kode\_obat'. The data is as follows:

id	nama_obat	katagori	kode_obat
2	paramex	pil	1234
3	komik	cair	9902
4			
5	ale ale	minuman	biar segar
6	bodrex cin	pill	4144
7	PARACETAMOL	FILL	342

The interface also shows the SQL query used to retrieve the data: `SELECT * FROM `menu_obat``. The 'Show all' button is visible, and the 'Number of rows' is set to 25.

Inputan katagori di aplikasi degan tujuan agar mengetahui katagoroi apay g mau di ambil sama pembeli minsalnya dia memilih fil atau salep

- **KODE OBAT**

Inputan kode obat di aplikasi degan tujuan agar mengetahui kode obat yang mana mau di ambil janggan salah menulis kode ini karna kode ini berbeda di setiap nama obat.

- **TAMBAH**

proses menginput informasi atau data baru ke dalam suatu sistem atau aplikasi. biasanya TAMBAH dilakukan untuk memperbarui basis data atau penyimpanan data ke dalam aplikasi dengan informasi terbaru. yang sesuai dengan kebutuhan pembeli

- **UPDATE**

Update" digunakan untuk memperbarui atau menyunting informasi yang sudah ada. Update sering digunakan untuk mengubah atau menyunting informasi entitas tertentu untuk menerapkan perubahan atau mengganti data yang sudah diisi

- **DELETE**

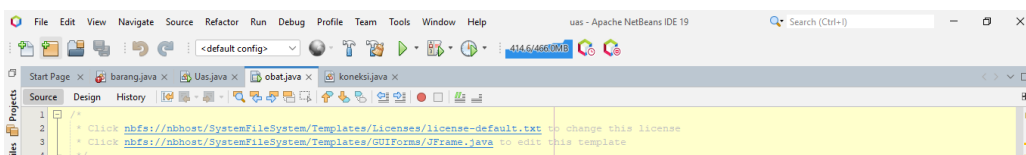
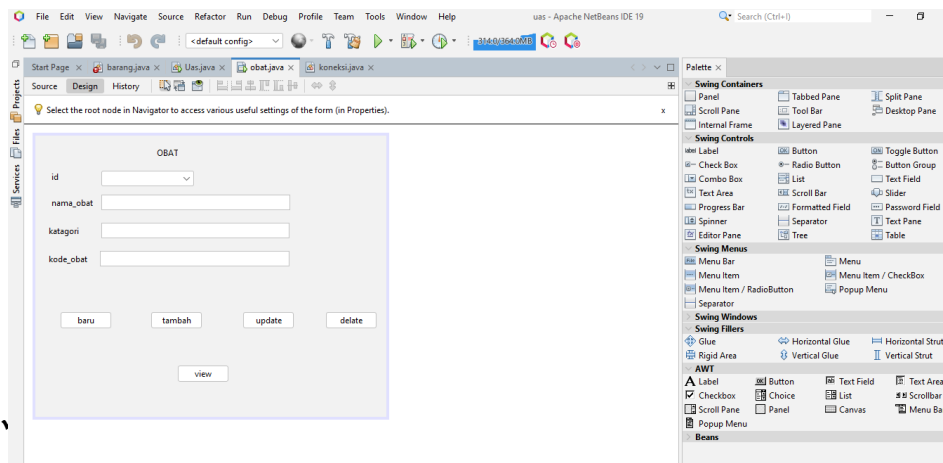
"delete" merupakan tindakan menghapus atau menghilangkan data dari suatu tempat penyimpanan, seperti basis data, sistem file, atau aplikasi.

- **NEW**

proses menginput informasi atau data baru ke dalam suatu sistem atau aplikasi. biasanya New dilakukan untuk memperbarui basis data atau penyimpanan data ke dalam aplikasi dengan informasi terbaru. yang sesuai dengan kebutuhan pembeli

### 3. POTO SEKERINSOUT

- ✓ **SETERUKTUR**







```

    try {
        String id = jComboBox1.getSelectedItem().toString();

        pst = con.prepareStatement(string: "select * from menu_obat where id=?");
        pst.setString(1, string: id);
        rs = pst.executeQuery();

        if (rs.next()==true){
            txtnama_obat.setText(s: rs.getString(1:2));
            txtkatagori.setText(s: rs.getString(1:3));
            txtkode_obat.setText(s: rs.getString(1:4));
        }

    } catch (SQLException ex) {
        Logger.getLogger(name: obat.class.getName()).log(level: Level.SEVERE, msg:null, thrown: ex);
    }

}

private void txtkode_obatActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

    try {
        String nama_obat = txtnama_obat.getText();
        String katagori = txtkatagori.getText();
        String kode_obat = txtkode_obat.getText();

```

```

        pst = con.prepareStatement(string: "insert into menu_obat (nama_obat, katagori, kode_obat)values (?,?,?)");
        pst.setString(1, string: nama_obat);
        pst.setString(2, string: katagori);
        pst.setString(3, string: kode_obat);

        int k = pst.executeUpdate();

        if (k==1){
            JOptionPane.showMessageDialog(parentComponent:this, message:"Data Berhasil Ditambahkan");
            txtnama_obat.setText(s: "");
            txtkatagori.setText(s: "");
            txtkode_obat.setText(s: "");
            txtnama_obat.requestFocus();
        }

        else{
            JOptionPane.showMessageDialog(parentComponent:this, message:"Data Gagal Ditambahkan");
        }

    } catch (SQLException ex) {
        Logger.getLogger(name: obat.class.getName()).log(level: Level.SEVERE, msg:null, thrown: ex);
    }

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    txtnama_obat.setText(s: "");
    txtkatagori.setText(s: "");
    txtkode_obat.setText(s: "");
    txtnama_obat.requestFocus();
    // TODO add your handling code here:
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

    String id = jComboBox1.getSelectedItem().toString();

    try {
        String nama_obat = txtnama_obat.getText();
        String katagori = txtkatagori.getText();
        String kode_obat = txtkode_obat.getText();

        pst = con.prepareStatement(string: "update menu_obat set nama_obat=?, katagori=?, kode_obat=? where id=?");
        pst.setString(1, string: nama_obat);
        pst.setString(2, string: katagori);
        pst.setString(3, string: kode_obat);
        pst.setString(4, string: id);

        int k = pst.executeUpdate();

        if (k==1){
            JOptionPane.showMessageDialog(parentComponent:this, message:"Data Berhasil Di Update");
            txtnama_obat.setText(s: "");
            txtkatagori.setText(s: "");
            txtkode_obat.setText(s: "");

```

```

    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            String id = jComboBox1.getSelectedItem().toString();

            pst = con.prepareStatement(string: "delete from menu_obat where id=?");

            pst.setString(i: 1, string: id);

            int k = pst.executeUpdate();

            if (k==1){
                JOptionPane.showMessageDialog(parentComponent:this, message:"Data Berhasil Di Hapus");
                txtnama_obat.setText(s: "");
                txtkatagori.setText(s: "");
                txtkode_obat.setText(s: "");
                txtnama_obat.requestFocus();
                LoadobatID();
            }

            else{
                JOptionPane.showMessageDialog(parentComponent:this, message:"Data Gagal Di Hapus");
            }
        }
    }

```

```

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new obat().setVisible(true);
        }
    });
}

```

```

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JComboBox<String> jComboBox1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JTextField txtkatagori;
private javax.swing.JTextField txtkode_obat;
private javax.swing.JTextField txtnama_obat;

```

## ✓ DATA

The screenshot shows the phpMyAdmin interface for the 'apotik' database. The 'menu\_obat' table is selected, and its data is displayed. The table has 7 rows and 4 columns: id, nama\_obat, katagori, and kode\_obat. The data is as follows:

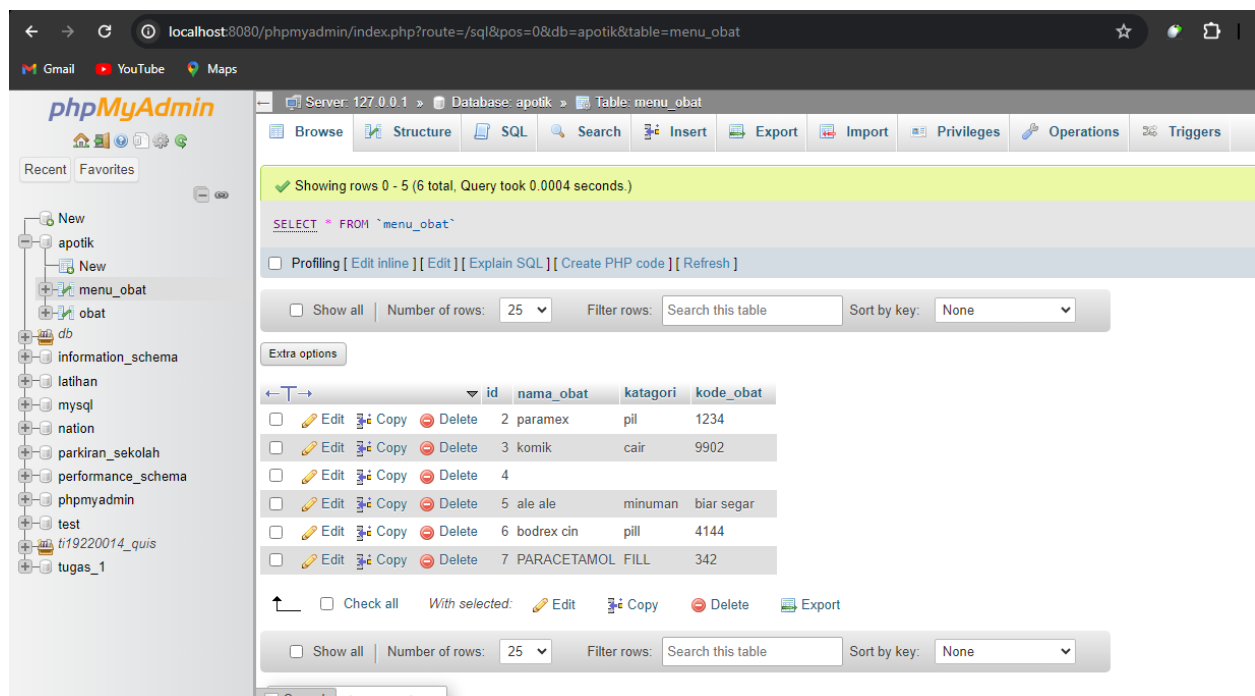
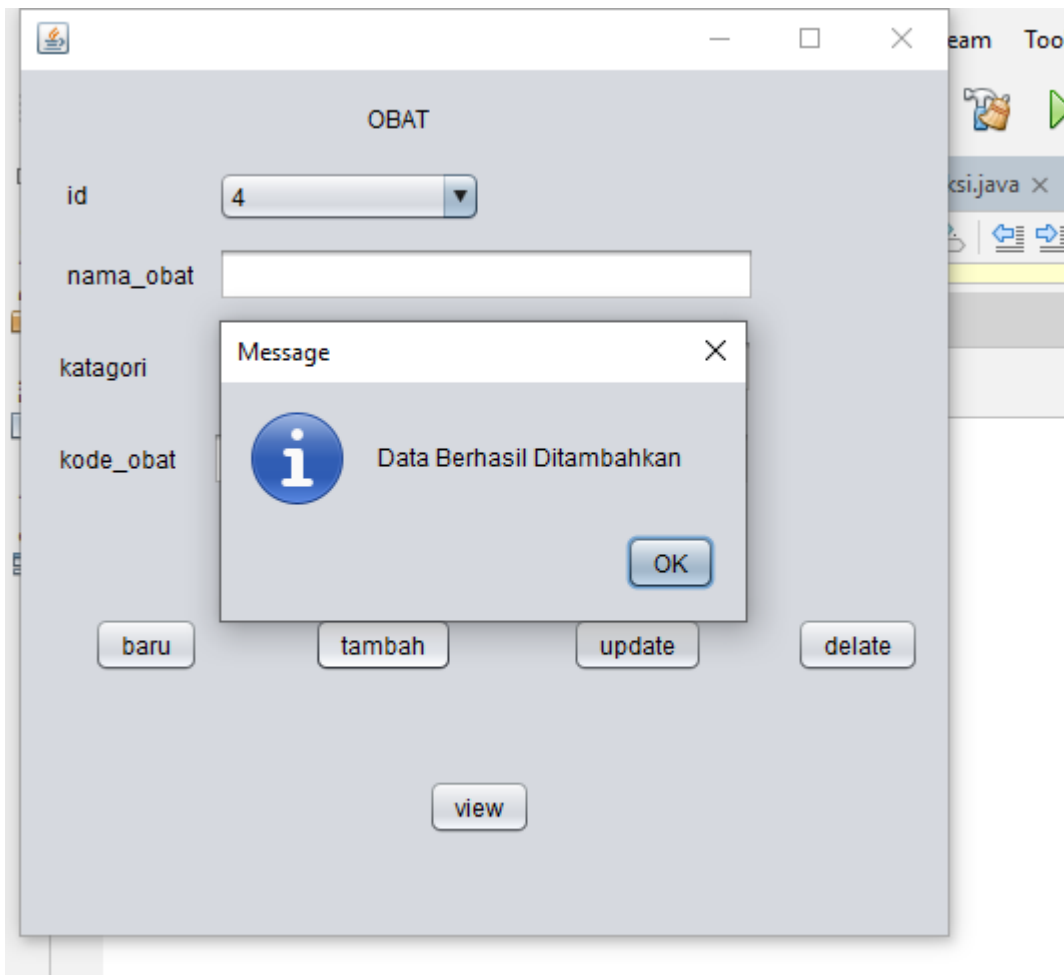
id	nama_obat	katagori	kode_obat
2	paramex	pil	1234
3	komik	cair	9902
4			
5	ale ale	minuman	biar segar
6	bodrex cin	pill	4144
7	PARACETAMOL	FILL	342

The screenshot shows a web application interface for managing medicine data. The main form is titled 'OBAT' and contains the following fields:

- id: 3
- nama\_obat: (empty)
- katagori: (empty)
- kode\_obat: (empty)

Below the form are four buttons: 'baru', 'tambah', 'update', and 'delete'. At the bottom center is a 'view' button.

A modal dialog box titled 'Message' is displayed in the center of the screen. It contains an information icon and the text 'Data Berhasil Di Hapus' (Data Successfully Deleted). There is an 'OK' button at the bottom right of the dialog.



### **BAB III**

### **PENUTUP**

Pengembangan aplikasi apotek di era digital menjadi suatu kebutuhan mendesak guna mengoptimalkan layanan kesehatan dan manajemen obat. Dengan memanfaatkan teknologi, khususnya platform pengembangan seperti Apache NetBeans, kita dapat menciptakan solusi yang lebih efisien, terintegrasi, dan responsif terhadap tuntutan zaman.

Aplikasi apotek tidak hanya memberikan kemudahan akses informasi obat kepada pengguna, tetapi juga merangsang inovasi dalam manajemen stok, pemesanan online, dan integrasi dengan sistem kesehatan. Pemantauan kesehatan pasien, keamanan data, serta peningkatan kepatuhan pasien menjadi aspek-aspek krusial yang dapat diatasi melalui implementasi teknologi aplikasi apotek.

Melalui makalah ini, kita telah menjelajahi peran penting Apache NetBeans dalam memfasilitasi pengembangan aplikasi apotek yang efektif. Keunggulan-keunggulan tersebut membawa dampak positif dalam memberikan pelayanan kesehatan yang lebih baik, meningkatkan efisiensi operasional apotek, dan memastikan keterhubungan antara berbagai elemen sistem kesehatan.

Sebagai penutup, pembahasan ini menggarisbawahi betapa pentingnya adaptasi terhadap perkembangan teknologi di dunia kesehatan. Dengan menggabungkan keahlian dan inovasi di bidang pengembangan aplikasi dengan kebutuhan nyata di apotek, kita dapat membuka pintu menuju perbaikan berkelanjutan dalam memberikan layanan kesehatan yang lebih canggih, efisien, dan terjangkau di era digital ini.