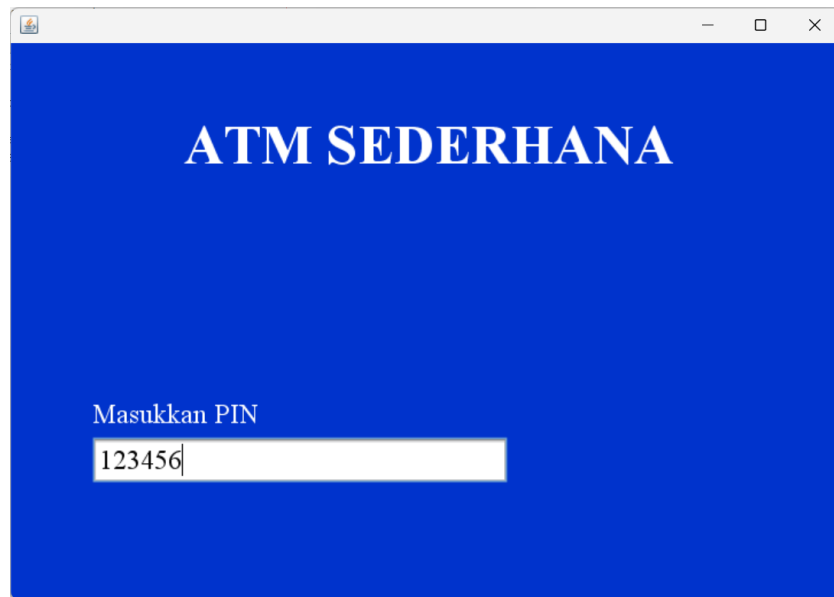


Penjelasan Halaman dan Source Code

- **Halaman Utama “Landing”**

Halaman ini meminta pengguna untuk memasukkan PIN untuk mengakses layanan ATM.



Gambar 1 Tampilan Menu Utama

- Berikut adalah kode “**Landing**” serta penjelasan setiap code

```

Start Page -> Order Java -> Landing.java -> JumahLain.java ->
Source Design History
package TempLain;

import javax.swing.*;

public class Landing extends javax.swing.JFrame {

    public Landing() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // Generated Code

    private void CodeLainActionPerformed(java.awt.event.ActionEvent evt) {
        boolean isInteger = evt.getActionCommand().matches("\\d+");

        // Kalau isinya bukan angka kasih tahu dia
        // supaya dia nggak harus angka.
        if (!isInteger) {
            JOptionPane.showMessageDialog(this, "FIT harus berupa angka.");
        } else {
            // Kalau benar angka
            int maxLength = evt.getActionCommand().length();

            if (maxLength > 6) {
                JOptionPane.showMessageDialog(this, "Jumlah maksimal FIT adalah 6 karakter.");
            } else if (evt.getActionCommand().matches("123456")) {
                JOptionPane.showMessageDialog(this, "FIT Boleh");
                new Order().setVisible(true);
                this.setVisible(false);
            } else {
                JOptionPane.showMessageDialog(this, "FIT Salah");
            }
        }
    }

    public static void main(String args[]) {
        // Look and feel setting code (optional)

        javax.swing.SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new Landing().setVisible(true);
            }
        });
    }
}

```

Kode ini adalah implementasi GUI sederhana menggunakan Java Swing untuk memvalidasi input PIN. Kelas Landing berfungsi sebagai jendela utama aplikasi, yang merupakan turunan dari JFrame. Program ini memanfaatkan pustaka javax.swing untuk membangun antarmuka grafis, dengan komponen diinisialisasi melalui metode initComponents().

Saat pengguna melakukan aksi (seperti menekan tombol), metode kodeOtpActionPerformed dipanggil untuk menangani input. Input diambil menggunakan evt.getActionCommand() dan divalidasi. Validasi pertama memeriksa apakah input hanya berupa angka menggunakan regex `\\d+`. Jika input tidak valid, pesan "PIN harus berupa angka" akan ditampilkan, dan proses dihentikan. Selanjutnya, panjang input diperiksa untuk memastikan tidak lebih dari 6 karakter. Jika melebihi batas, muncul pesan "PIN maksimal 6 karakter."

Setelah lolos validasi, input diperiksa apakah cocok dengan PIN yang diharapkan, yaitu "123456". Jika sesuai, pesan "PIN Benar" ditampilkan, dan jendela GUI ditutup dengan `this.setVisible(false)`. Jika tidak sesuai, pesan "PIN Salah" muncul.

Metode main adalah titik masuk aplikasi. Program memastikan GUI dijalankan di thread yang benar menggunakan `EventQueue.invokeLater`, lalu menampilkan jendela utama. Kode ini secara keseluruhan memastikan input PIN valid sebelum menampilkan hasil, dengan mekanisme kesalahan ditampilkan melalui dialog `JOptionPane`.

Fitur utama dari program validasi PIN sederhana menggunakan Java Swing adalah:

- Memastikan PIN berupa angka.
- Membatasi panjang PIN maksimal 6 karakter.
- Memeriksa kecocokan PIN dengan nilai yang telah ditentukan.

Jika validasi gagal, pesan kesalahan ditampilkan melalui dialog `JOptionPane`. Jika validasi berhasil, jendela GUI ditutup, menandakan keberhasilan proses.

- **Menu “Menu Utama”**

Halaman ini menampilkan menu utama yang menyediakan opsi transaksi



Gambar 2 Menu Tarik Tunai

- Berikut adalah kode “**order**” serta penjelasan setiap code

```

1 package Tempilan;
2
3 import javax.swing.JOptionPane;
4
5 public class Order extends javax.swing.JFrame {
6
7     public Order() {
8         initComponents();
9     }
10
11     @SuppressWarnings("unchecked")
12     // Generated Code
13
14     private void KembaliActionPerformed(java.awt.event.ActionEvent evt) {
15         this.setVisible(false);
16         new Landing().setVisible(true);
17     }
18
19     private void dana1ActionPerformed(java.awt.event.ActionEvent evt) {
20         String nominal = evt.getActionCommand().toString();
21         JOptionPane.showMessageDialog(this, "Dana dana berhasil!\n\nJumlah : " + nominal);
22     }
23
24     private void dana2ActionPerformed(java.awt.event.ActionEvent evt) {
25         String nominal = evt.getActionCommand().toString();
26         JOptionPane.showMessageDialog(this, "Dana dana berhasil!\n\nJumlah : " + nominal);
27     }
28
29     private void dana3ActionPerformed(java.awt.event.ActionEvent evt) {
30         String nominal = evt.getActionCommand().toString();
31         JOptionPane.showMessageDialog(this, "Dana dana berhasil!\n\nJumlah : " + nominal);
32     }
33
34     private void dana4ActionPerformed(java.awt.event.ActionEvent evt) {
35         String nominal = evt.getActionCommand().toString();
36         JOptionPane.showMessageDialog(this, "Dana dana berhasil!\n\nJumlah : " + nominal);
37     }
38
39     private void JumlahActionPerformed(java.awt.event.ActionEvent evt) {
40         this.setVisible(false);
41         new JumlahLain().setVisible(true);
42     }
43
44 }

```

```

150: public static void main(String args[]) {
151:
152:     // Look and feel setting code (optional)
153:
154:
155:     javax.swing.EventQueue.invokeLater(new Runnable() {
156:         public void run() {
157:             new Order().setVisible(true);
158:
159:         }
160:     });
161:
162:     // Variables declaration - do not modify
163:     private javax.swing.JButton jButton1;
164:     private javax.swing.JButton jButton2;
165:     private javax.swing.JButton jButton3;
166:     private javax.swing.JButton jButton4;
167:     private javax.swing.JButton jButton5;
168:     private javax.swing.JButton jButton6;
169:     private javax.swing.JLabel jLabel1;
170:
171:     // End of variables declaration
172: }

```

Order

5. Metode main Metode ini adalah titik masuk utama untuk menjalankan aplikasi. Ini memastikan tampilan GUI dimulai di thread yang benar dengan menggunakan `java.awt.EventQueue.invokeLater`. Kemudian, objek `Order` dibuat dan ditampilkan dengan `setVisible(true)`.

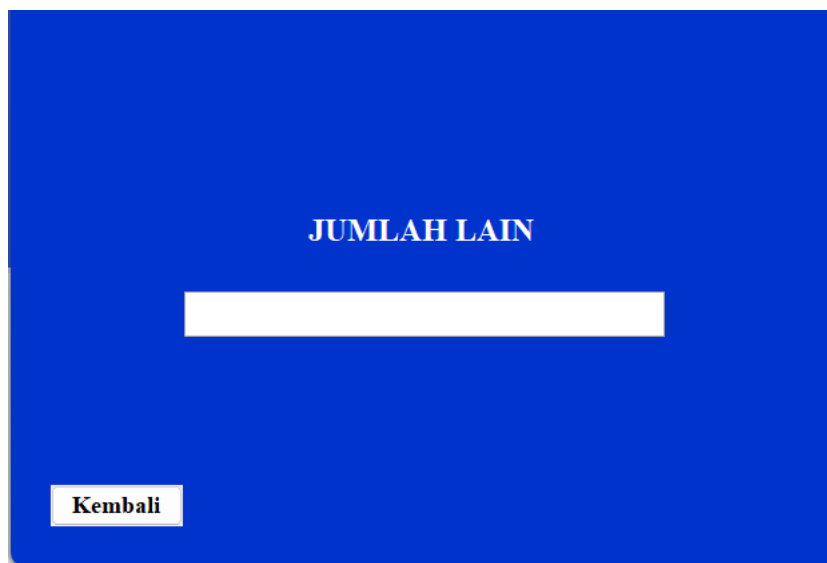
Kode ini merupakan implementasi GUI untuk melakukan pengelolaan dana, dengan fitur seperti:

- Navigasi antar jendela (misalnya, kembali ke menu utama atau menuju pengisian jumlah lain).
- Penanganan input pengguna melalui tombol-tombol untuk mengambil nominal tertentu.
- Konfirmasi aksi menggunakan dialog dengan JOptionPane.

Setiap tombol memiliki aksi spesifik, tetapi prinsip operasinya serupa, yaitu menangkap input nominal dan menampilkan hasilnya

– **JumlahLain**

halaman ini digunakan untuk memasukkan jumlah atau nominal tertentu.



Gambar 3 Menu Jumlah Lain

- Berikut adalah kode “**jumlaHLain**” serta penjelasan setiap code

```

1 package Tampilan;
2
3 import javax.swing.*;
4
5 public class JumlahLain extends JFrame {
6
7     public JumlahLain() {
8         initComponents();
9     }
10
11     @SuppressWarnings("unchecked")
12     // Generated Code
13
14     private void DanaActionPerformed(java.awt.event.ActionEvent evt) {
15         String nominal = evt.getActionCommand().toString();
16         int jawab = JOptionPane.showConfirmDialog(this, "Apakah anda yakin?", "Konfirmasi", JOptionPane.YES_NO_OPTION);
17
18         try {
19             if (jawab == 0) {
20                 JOptionPane.showMessageDialog(this, "Dana anda berhasil! " + "Jumlah : " + nominal);
21             }
22         } catch (Exception e) {
23             e.printStackTrace();
24         }
25     }
26
27     public static void main(String args[]) {
28         // Look and Feel setting code (optional)
29
30         java.awt.EventQueue.invokeLater(new Runnable() {
31             public void run() {
32                 new JumlahLain().setVisible(true);
33             }
34         });
35     }
36 }

```

Kode ini adalah bagian dari sebuah aplikasi Java berbasis Swing yang menggunakan antarmuka grafis untuk melakukan aksi tertentu, seperti menarik dana dengan konfirmasi pengguna.

1. Deklarasi Paket dan Import

Kode dimulai dengan deklarasi paket Tampilan, yang digunakan untuk mengorganisasi file Java ke dalam grup tertentu. Selanjutnya, semua kelas dalam paket javax.swing diimport untuk membangun antarmuka grafis, seperti JFrame, JOptionPane, dan lain-lain.

2. Deklarasi dan Struktur Kelas

Kelas JumlahLain dideklarasikan sebagai subclass dari JFrame, sehingga mewarisi kemampuan untuk membuat jendela grafis. Kelas ini memiliki sebuah konstruktor bernama JumlahLain yang memanggil metode initComponents() untuk menginisialisasi komponen grafis (tombol, label, area teks, dll.). Metode initComponents() biasanya dihasilkan secara otomatis oleh IDE seperti NetBeans.

3. Metode DanaActionPerformed

Metode ini dipanggil saat suatu aksi, seperti klik tombol, dilakukan oleh pengguna. Fungsi utamanya adalah:

- Mengambil data dari aksi tombol (misalnya jumlah nominal).
- Menampilkan dialog konfirmasi kepada pengguna menggunakan JOptionPane.showConfirmDialog, dengan pilihan "Ya" atau "Tidak".

- Jika pengguna memilih "Ya" (nilai jawab adalah 0), maka pesan sukses ditampilkan melalui `JOptionPane.showMessageDialog` yang mencantumkan jumlah nominal.
- Jika terjadi kesalahan, metode `catch` menangkap dan mencetak informasi kesalahan menggunakan `e.printStackTrace()`.

4. Metode main

Metode ini adalah titik awal program Java. Dengan menggunakan `EventQueue.invokeLater`, program memastikan bahwa semua operasi GUI dijalankan di thread yang aman untuk event dispatching. Metode ini membuat instance dari kelas `JumlahLain` dan menampilkan antarmuka grafisnya dengan `setVisible(true)`.

5. Deklarasi Variabel Komponen

Variabel-variabel seperti `Dana`, `jEditorPane1`, `jLabel1`, dan `jScrollPane1` digunakan untuk merepresentasikan elemen antarmuka grafis, seperti:

- **Dana:** Input teks untuk memasukkan nominal dana.
- **jLabel1:** Label untuk menampilkan teks statis.
- **jScrollPane1:** Komponen scroll untuk area teks atau elemen grafis lainnya.

Kode ini merupakan implementasi dasar dari sebuah antarmuka grafis untuk simulasi penarikan dana dengan konfirmasi pengguna. Dengan memanfaatkan elemen-elemen seperti dialog konfirmasi dan pesan sukses, aplikasi memastikan interaksi yang mudah dipahami pengguna. Struktur ini cocok untuk diterapkan dalam aplikasi yang membutuhkan interaksi sederhana dan tanggapan cepat terhadap input pengguna.