

# **LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK**

## **PRAKTIKUM 6**

### **EXCEPTION HANDLING**



**Nama: Siti Fadhilah Rahmi**

**NIM: 2311532003**

**Dosen Pengampu:**

**DEPARTEMEN INFORMATIKA**

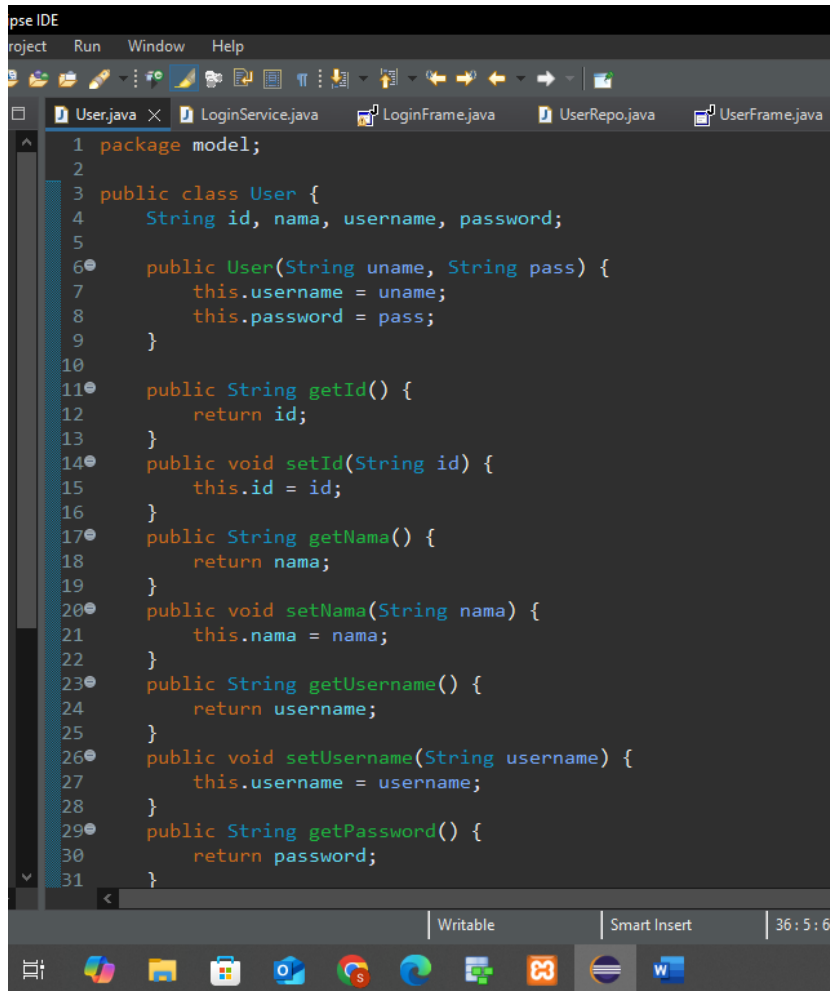
**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**2024**

## A. PRAKTIKUM

### 1. Membuat Class User



```
1 package model;
2
3 public class User {
4     String id, nama, username, password;
5
6     public User(String uname, String pass) {
7         this.username = uname;
8         this.password = pass;
9     }
10
11     public String getId() {
12         return id;
13     }
14     public void setId(String id) {
15         this.id = id;
16     }
17     public String getNama() {
18         return nama;
19     }
20     public void setNama(String nama) {
21         this.nama = nama;
22     }
23     public String getUsername() {
24         return username;
25     }
26     public void setUsername(String username) {
27         this.username = username;
28     }
29     public String getPassword() {
30         return password;
31     }
32 }
```

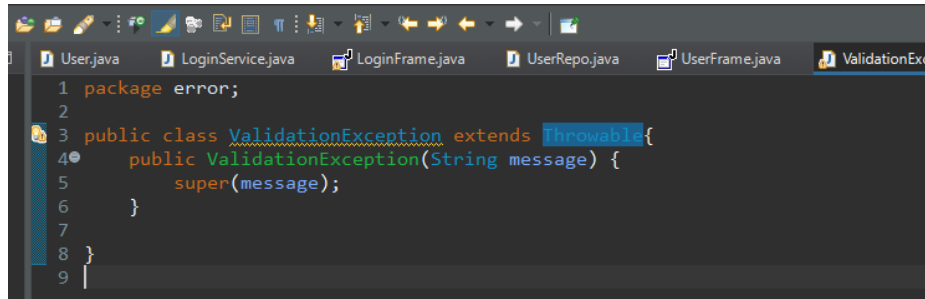
- Pada praktikum sebelumnya, kelas user telah dibuat, namun pada praktikum ini, kelas user dimodifikasi dengan menambahkan constructor User.
- Konstruktor kelas User menerima dua parameter (uname dan pass) untuk menginisialisasi atribut username dan password.

### 2. Membuat Class Exception

Di Java, error direpresentasikan dengan istilah exception dan semuanya direpresentasikan dalam bentuk class exception. Kita bisa membuat class exception sendiri atau menggunakan yang sudah disediakan oleh Java. Jika kita ingin membuat exception, maka kita harus membuat class yang extends class Throwable. Exception biasanya terjadi di method, sehingga ketika kita membuat exception di sebuah method maka method tersebut harus ditandai dengan kata kunci throws diikuti dengan class

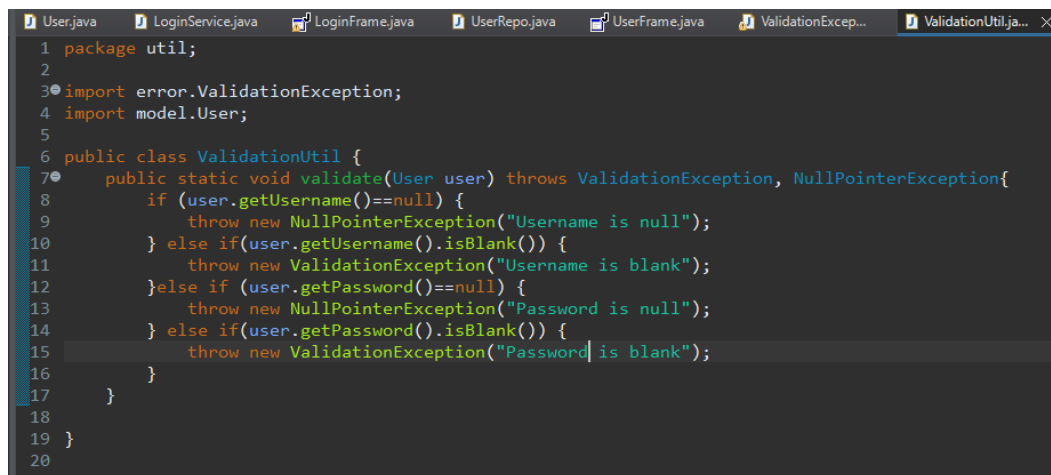
exceptionnya. Jika method tersebut bisa menimbulkan lebih dari satu jenis exception, kita bisa menambahkan lebih dari satu class exception.

- Buat package baru dengan nama error, kemudian di dalam package error buat class baru dengan nama ValidationException.



```
1 package error;
2
3 public class ValidationException extends Throwable{
4     public ValidationException(String message) {
5         super(message);
6     }
7 }
8
9 |
```

- Buat package baru dengan nama util dan di dalamnya buat sebuah class baru dengan nama ValidationUtil. Class ValidationUtil memiliki static method dengan nama validate yang akan memvalidasi proses login yang jika terjadi error akan memberitahu pemanggilnya



```
1 package util;
2
3 import error.ValidationException;
4 import model.User;
5
6 public class ValidationUtil {
7     public static void validate(User user) throws ValidationException, NullPointerException{
8         if (user.getUsername()==null) {
9             throw new NullPointerException("Username is null");
10        } else if (user.getUsername().isBlank()) {
11            throw new ValidationException("Username is blank");
12        } else if (user.getPassword()==null) {
13            throw new NullPointerException("Password is null");
14        } else if (user.getPassword().isBlank()) {
15            throw new ValidationException("Password is blank");
16        }
17    }
18 }
19
20
```

### 3. Membuat Class LoginService

Class LoginService ini berfungsi untuk melakukan otentikasi username dan password yang dimasukkan pengguna. Username dan password yang dimasukkan pengguna akan dicocokkan dengan data yang ada di database, jika cocok maka login pengguna berhasil namun jika tidak cocok maka login pengguna gagal.

- Buat package baru dengan nama service, kemudian di dalam package service ini buat class baru dengan nama LoginService.

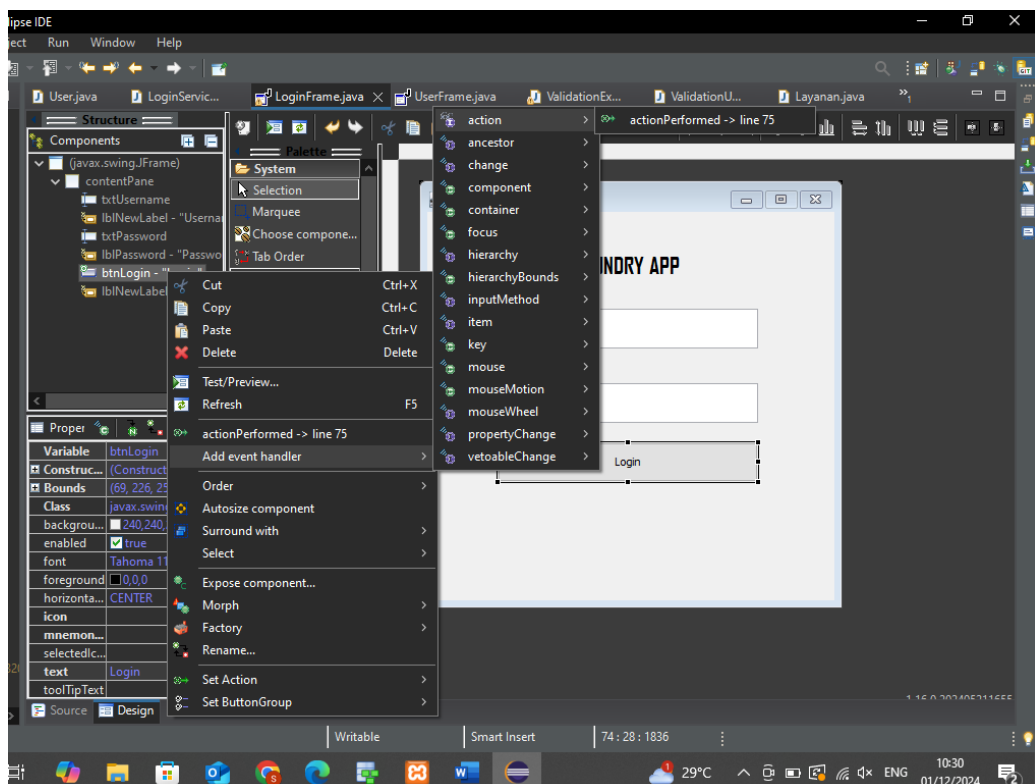
```

1 package service;
2 import java.sql.Connection;
3 import java.sql.PreparedStatement;
4 import java.sql.ResultSet;
5 import java.sql.SQLException;
6 import config.Database;
7 import model.User;
8
9 public class LoginService {
10     public boolean authenticate(User user) {
11         String query = "SELECT * FROM user WHERE username = ? AND password = ?";
12
13         try (Connection conn = Database.koneksi();
14              PreparedStatement statement = conn.prepareStatement(query)){
15             statement.setString(1, user.getUsername());
16             statement.setString(2, user.getPassword());
17
18             ResultSet resultSet = statement.executeQuery();
19             return resultSet.next();
20         } catch (SQLException e) {
21             e.printStackTrace();
22         }
23         return false;
24     }
25 }
26 }
27

```

#### 4. Membuat Tampilan Login Menggunakan JFrame

- Masuk ke class LoginFrame pada package ui.
- Klik kanan pada button login, pilih add event handler → action → actionPerformed



- Selanjutnya, lakukan validasi terhadap inputan pengguna setelah tombol login diklik. Validasi dilakukan dengan memanggil method validate yang telah dibuat pada tahapan sebelumnya.

```

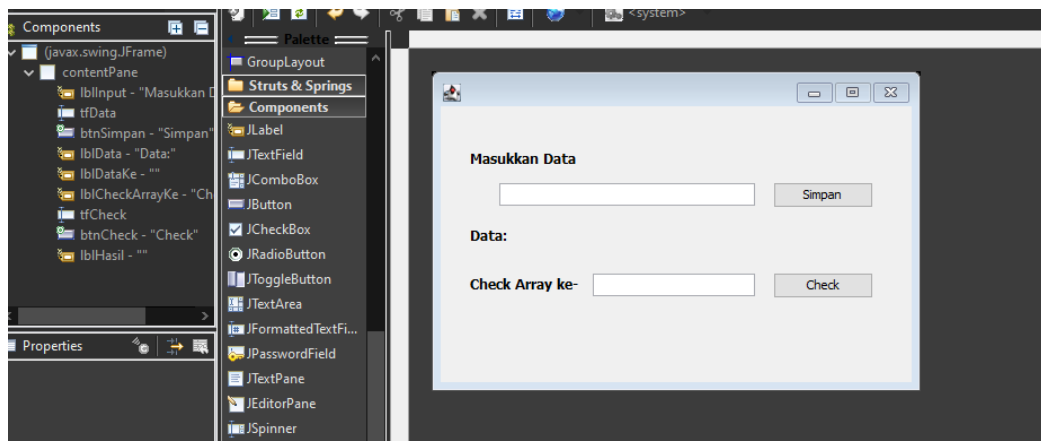
77 JButton btnLogin = new JButton("Login");
78 btnLogin.addActionListener(new ActionListener() {
79     public void actionPerformed(ActionEvent e) {
80         String userValue = txtUsername.getText();
81         String passValue = txtPassword.getText();
82
83         User user = new User(userValue, passValue);
84
85         try {
86             ValidationUtil.validate(user);
87             LoginService loginService = new LoginService();
88             if(LoginService.authenticate(user)) {
89                 System.out.println("Login Successful");
90                 new MainFrame().setVisible(true);
91                 dispose();
92             }
93             else {
94                 System.out.println("Invalid username or password");
95                 JOptionPane.showMessageDialog(null, "Login Gagal, Invalid username or password");
96             }
97         } catch (ValidationException | NullPointerException exception) {
98             System.out.println("Data tidak valid : "+exception.getMessage());
99             JOptionPane.showMessageDialog(null, "Login Gagal: "+exception.getMessage());
100
101         } finally {
102             System.out.println("Selalu dieksekusi");
103         }
104     }
105 });

```

- Saat memanggil method yang sebuah method yang kemungkinan bisa menyebabkan exception maka kita wajib menggunakan try-catch expression di Java.
- Ini bertujuan agar bisa menangkap exception yang terjadi, karena jika tidak ditangkap ketika terjadi exception maka program secara otomatis akan terhenti.
- Cara menggunakan try-catch expression di Java adalah di block try tinggal panggil method yang bisa menyebabkan exception dan di block catch bisa melakukan sesuatu jika terjadi exception.

## B. TUGAS

1. Setelah berhasil login dan masuk ke MainFrame, buat sebuah JFrame baru.

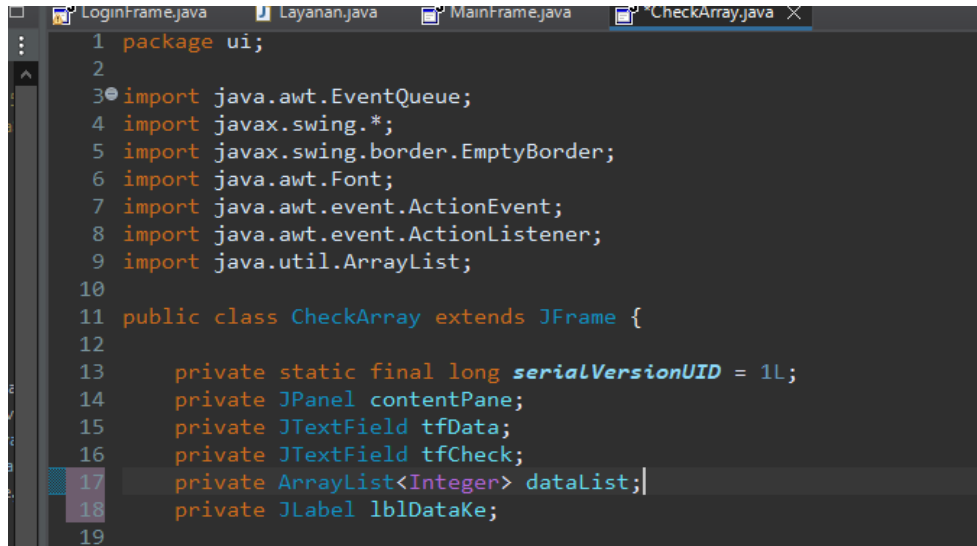


2. Pengguna bisa memasukan data angka-angka yang ketika tombol Simpan diklik data tersebut akan disimpan pada variabel dalam bentuk Array dan ditampilkan di GUI.

- Pertama saya mendeklarasikan:

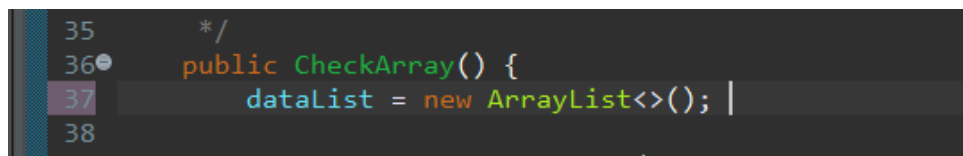
`private ArrayList<Integer> dataList; // Untuk menyimpan data angka dalam bentuk ArrayList`

`private JLabel lblDataKe; // Untuk menampilkan data yang disimpan`



```
1 package ui;
2
3 import java.awt.EventQueue;
4 import javax.swing.*;
5 import javax.swing.border.EmptyBorder;
6 import java.awt.Font;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9 import java.util.ArrayList;
10
11 public class CheckArray extends JFrame {
12
13     private static final long serialVersionUID = 1L;
14     private JPanel contentPane;
15     private JTextField tfData;
16     private JTextField tfCheck;
17     private ArrayList<Integer> dataList;
18     private JLabel lblDataKe;
19 }
```

- Inisialisasi Array untuk menyimpan data



```
35     */
36     public CheckArray() {
37         dataList = new ArrayList<>();
38     }
```

- Buat method updateLabel untuk menampilkan data yang diinputkan pengguna



```
130
131     private void updateLabel(JLabel lblDataKe) {
132         StringBuilder dataText = new StringBuilder();
133         for (int i = 0; i < dataList.size(); i++) {
134             dataText.append(dataList.get(i));
135             if (i < dataList.size() - 1) {
136                 dataText.append(", ");
137             }
138         }
139         lblDataKe.setText(dataText.toString());
140     }
141 }
142
```

- Tambahkan event listener pada btnSimpan untuk menyimpan data yang diinputkan pengguna

```

90
91     btnSimpan.addActionListener(new ActionListener() {
92     @Override
93     public void actionPerformed(ActionEvent e) {
94         String input = tfData.getText();
95         try {
96             String[] numberStrings = input.split(",");
97             for (String numStr : numberStrings) {
98                 dataList.add(Integer.parseInt(numStr.trim()));
99             }
100             tfData.setText("");
101             updateLabel(lblDataKe);
102         } catch (NumberFormatException ex) {
103             JOptionPane.showMessageDialog(
104                 CheckArray.this,
105                 "Harap masukkan angka yang valid, dipisahkan dengan koma.",
106                 "Input Error",
107                 JOptionPane.ERROR_MESSAGE
108             );
109         }
110     }
111 });
112

```

- Tambahkan action Listener pada tombol Check untuk memeriksa Array

```

112
113     btnCheck.addActionListener(new ActionListener() {
114     @Override
115     public void actionPerformed(ActionEvent e) {
116         String indexInput = tfCheck.getText();
117         try {
118             int index = Integer.parseInt(indexInput.trim());
119             if (index >= 0 && index < dataList.size()) {
120                 lblHasil.setText("Hasil: Index ke-" + index + " adalah " + dataList.get(index));
121             } else {
122                 lblHasil.setText("Error: Index di luar jangkauan!");
123             }
124         } catch (NumberFormatException ex) {
125             lblHasil.setText("Error: Masukkan index yang valid.");
126         }
127     }
128 });
129

```

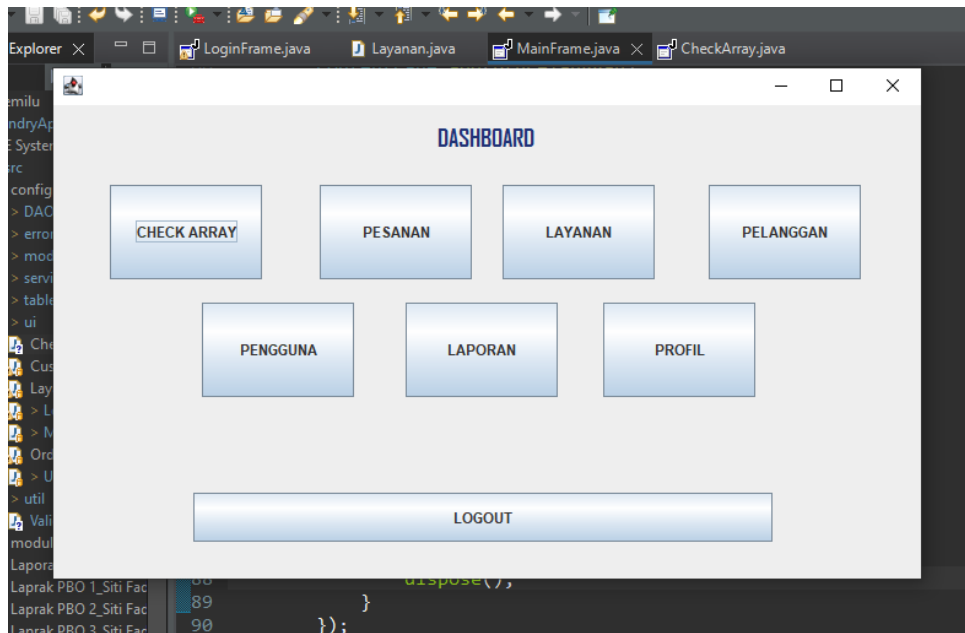
- Sambungkan dengan MainFrame. Ketika tombol pada MainFrame diklik, maka akan diarahkan ke Frame CheckArray ini.

```

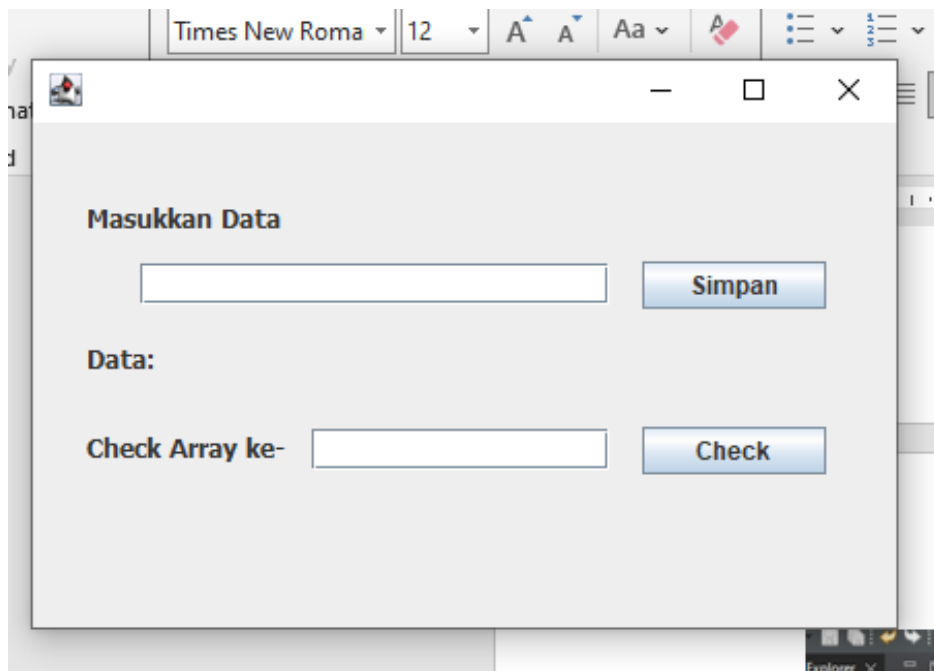
83
84     JButton btnCheckArray = new JButton("CHECK ARRAY");
85     btnCheckArray.addActionListener(new ActionListener() {
86     public void actionPerformed(ActionEvent e) {
87         new CheckArray().setVisible(true);
88         dispose();
89     }
90 });
91

```

**HASIL:**



Ketika tombol Check Array ditekan maka akan tampil:



Masukkan data lalu klik tombol Simpan, maka akan tampil datanya



**Masukkan Data**

**Simpan**

**Data:** 1, 2, 3, 4, 5, 6, 7, 8

**Check Array ke-**

**Check**

Lakukan check dengan memasukkan index array yang ingin diketahui lalu klik tombol Check

**Masukkan Data**

**Simpan**

**Data:** 1, 2, 3, 4, 5, 6, 7, 8

**Check Array ke-** 5 **Check**

Hasil: Index ke-5 adalah 6