



# PPT Praktikum PBO Week 14

JAVA GUI

---

# Java Swing



# Java Swing

- Dalam dunia pengembangan perangkat lunak, antarmuka pengguna (GUI — Graphical User Interface) memainkan peran penting dalam memberikan pengalaman pengguna yang baik.
- Java Swing adalah sebuah toolkit yang kuat yang digunakan untuk membangun antarmuka pengguna dalam aplikasi Java.
- Java Swing adalah bagian dari API (Application Programming Interface) Java yang digunakan untuk membangun antarmuka pengguna dalam aplikasi desktop.
- Swing menawarkan berbagai komponen GUI, termasuk tombol, kotak teks, daftar, tabel, dan banyak lagi.



## Mengapa Java Swing

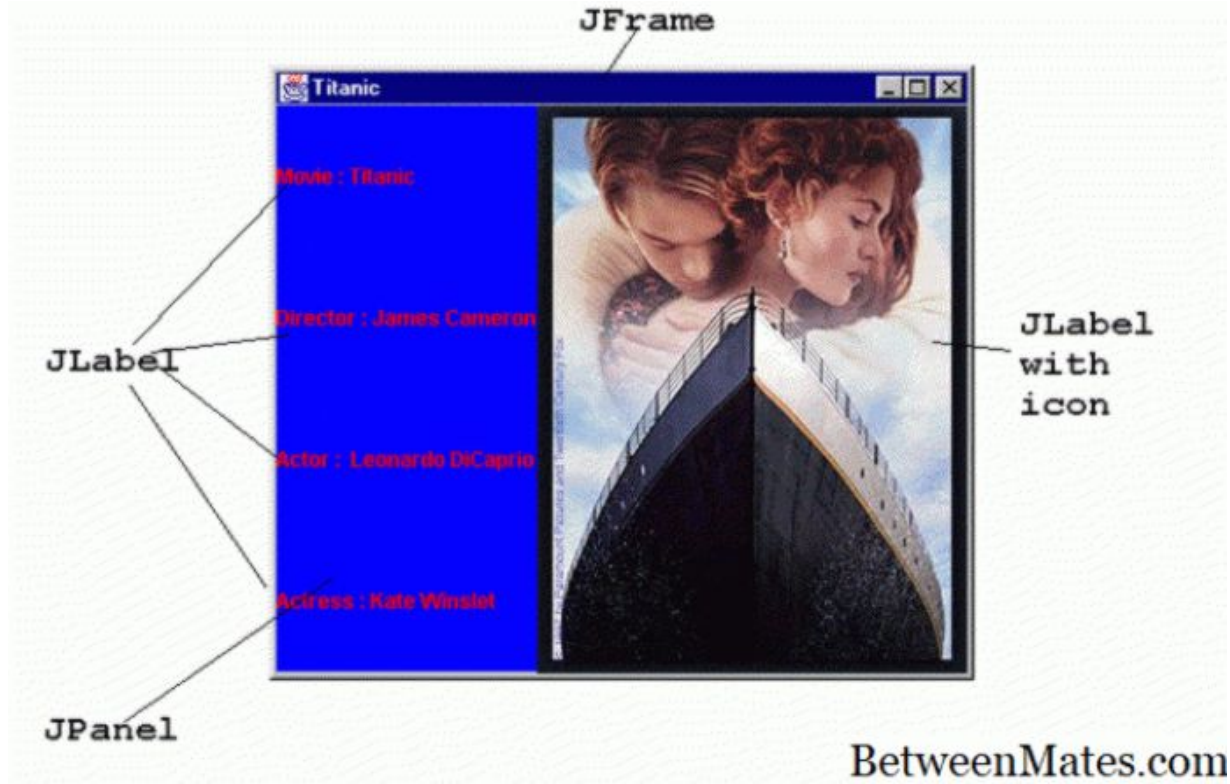
- Aplikasi Swing dapat dijalankan di berbagai sistem operasi seperti Windows, macOS, dan Linux tanpa perubahan signifikan dalam kode sumber.
- Swing menyediakan banyak komponen GUI yang dapat digunakan untuk membangun antarmuka pengguna yang kompleks. Komponen ini dapat dikustomisasi sesuai kebutuhan aplikasi.
- Dengan Swing, tampilan dan perasaan antarmuka pengguna akan tetap konsisten di berbagai sistem operasi, memberikan pengalaman yang lebih baik bagi pengguna.
- Swing memungkinkan pengembang untuk menggabungkan komponen GUI dengan mudah, serta memanipulasi tampilan dan perilaku mereka sesuai dengan kebutuhan aplikasi.



# Toolkit GUI pada Java

- Dalam Java Swing, JFrame, JPanel, dan berbagai komponen lain yang diawali dengan "J" adalah bagian dari toolkit GUI yang digunakan untuk membuat aplikasi desktop.
- Kita akan mempelajari satu-persatu pengertiannya

# Visual Dasar Java Swing



---

# JFrame



## JFrame

- JFrame adalah salah satu komponen utama dalam Java Swing yang digunakan untuk membuat jendela utama aplikasi desktop. Ini adalah bagian dari Java Foundation Classes (JFC) dan menyediakan antarmuka untuk membuat, mengelola, dan menampilkan jendela GUI.
- JFrame berada dalam paket `javax.swing`
- JFrame adalah subkelas dari `java.awt.Frame` dan `javax.swing.RootPaneContainer`. Ini berarti JFrame adalah jenis Frame yang merupakan window container dengan fitur tambahan dari Swing.





## JFrame

- JFrame secara otomatis memiliki title bar dan border default yang berisi tombol minimize, maximize, dan close.
- JFrame menyediakan metode untuk menangani event seperti menutup jendela (`setDefaultCloseOperation`).
- Anda dapat mengatur layout jendela menggunakan layout manager seperti `BorderLayout`, `FlowLayout`, `GridLayout`.

# Contoh Penggunaan JFrame



```
// Membuat JFrame
JFrame frame = new JFrame("Contoh JFrame");

// Mengatur ukuran jendela
frame.setSize(400, 300);

// Menetapkan close operation
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Menambahkan label ke jendela
JLabel label = new JLabel("Ini adalah JLabel di dalam JFrame");
frame.getContentPane().add(label);

// Menampilkan jendela
frame.setVisible(true);
```

---

**JPanel**



## JPanel

- JPanel adalah salah satu komponen dasar dalam Java Swing yang digunakan sebagai container untuk menampung dan mengelola tata letak komponen GUI lainnya. JPanel sangat berguna dalam mengorganisir antarmuka pengguna dengan lebih terstruktur dan teratur.
- JPanel berada dalam paket `javax.swing`.
- JPanel adalah subkelas dari `javax.swing.JComponent`, yang berarti JPanel adalah komponen ringan (lightweight) yang tidak tergantung pada elemen GUI bawaan dari sistem operasi.



## JPanel

- JPanel digunakan untuk mengelompokkan komponen GUI lainnya seperti tombol (JButton), label (JLabel), kotak teks (JTextField), dan banyak lagi.
- JPanel mendukung berbagai layout manager yang menentukan bagaimana komponen di dalamnya diatur, seperti FlowLayout, BorderLayout, GridLayout, BoxLayout, dan lain-lain.
- JPanel mendukung perubahan warna latar belakang, warna batas, dan berbagai properti lainnya yang memungkinkan kustomisasi tampilan sesuai kebutuhan.



## JPanel

- JPanel digunakan untuk mengelompokkan komponen GUI lainnya seperti tombol (JButton), label (JLabel), kotak teks (JTextField), dan banyak lagi.
- JPanel mendukung berbagai layout manager yang menentukan bagaimana komponen di dalamnya diatur, seperti FlowLayout, BorderLayout, GridLayout, BoxLayout, dan lain-lain.
- JPanel mendukung perubahan warna latar belakang, warna batas, dan berbagai properti lainnya yang memungkinkan kustomisasi tampilan sesuai kebutuhan.

# Contoh JPanel



```
// Membuat JPanel dengan layout FlowLayout
JPanel panel = new JPanel();
panel.setLayout(new FlowLayout());

// Menambahkan komponen ke JPanel
panel.add(new JButton("Button 1"));
panel.add(new JButton("Button 2"));
panel.add(new JLabel("Ini adalah JLabel"));

// Mengubah warna latar belakang JPanel
panel.setBackground(Color.LIGHT_GRAY);

// Menambahkan JPanel ke JFrame
frame.add(panel);
```

---

# Action Event





## Action Event

- ActionEvent dalam Java adalah salah satu jenis event yang digunakan dalam pemrograman GUI untuk menangani aksi yang dilakukan oleh pengguna, seperti menekan tombol, memilih item dari menu, atau melakukan tindakan lain yang memicu event.
- ActionEvent adalah bagian dari paket `java.awt.event`.
- ActionEvent digunakan untuk mewakili berbagai jenis aksi yang dipicu oleh pengguna, seperti: Menekan tombol (`JButton`), Memilih item dari menu (`JMenuItem`), Mengaktifkan kotak centang (`JCheckBox`), Menekan tombol Enter di dalam kotak teks (`TextField`)

# Contoh Action Event



```
JFrame frame = new JFrame("ActionEvent Example");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(300, 200);

JButton button = new JButton("Click Me");

// Menambahkan ActionListener ke tombol
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Menangani aksi tombol
        System.out.println("Button clicked! ActionCommand: " + e.getActionCommand());
    }
});

frame.add(button);
frame.setVisible(true);
```

---

# Runnable



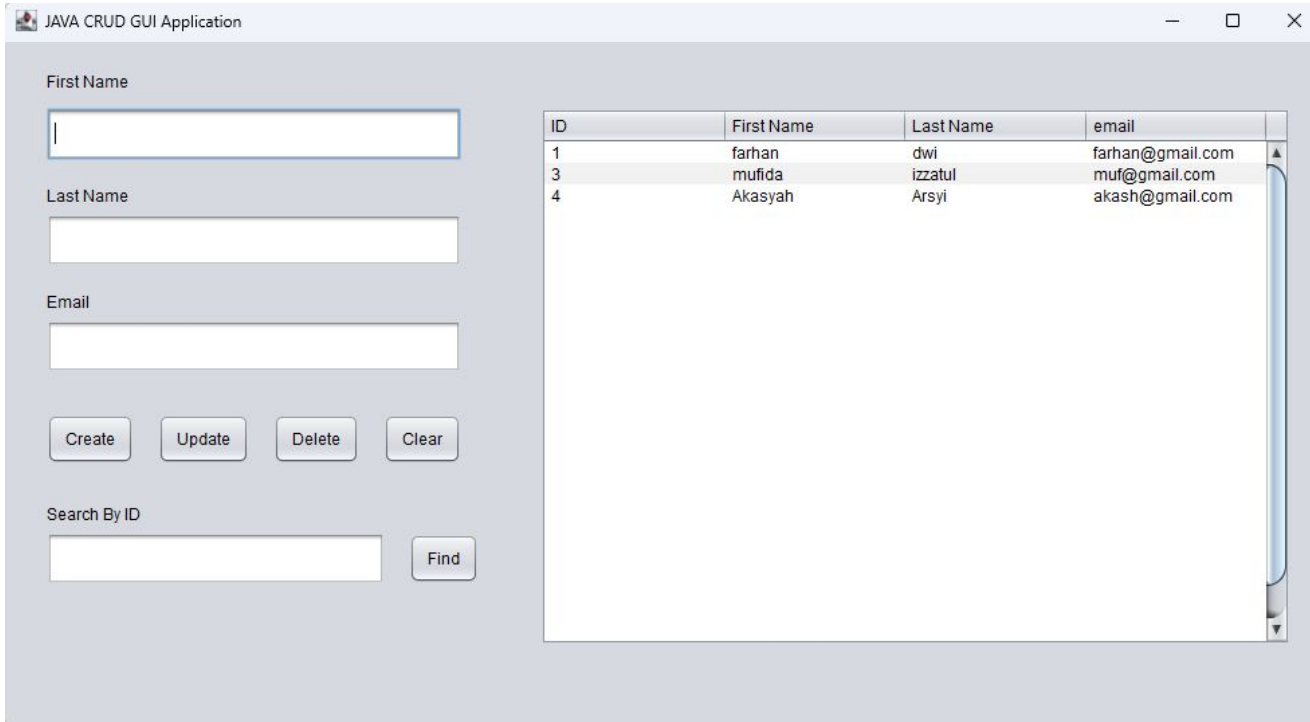
# Runnable

- Runnable adalah antarmuka (interface) di Java yang digunakan untuk mendefinisikan sebuah tugas yang akan dijalankan oleh sebuah thread. Konsep Runnable sangat penting dalam pemrograman concurrent (concurrent programming) di Java.
- Runnable berada dalam paket `java.lang`.
- Untuk menggunakan Runnable, Anda perlu membuat kelas yang mengimplementasikan antarmuka ini dan menyediakan implementasi untuk metode `run()`.
- Selain menggunakan Runnable, Anda juga bisa membuat kelas dengan meng-extend Thread secara langsung, namun pendekatan ini kurang disarankan karena Java tidak mendukung multiple inheritance. Menggunakan Runnable lebih fleksibel karena memungkinkan kelas untuk meng-extend kelas lain.

---

# Membuat Aplikasi CRUD dengan Java GUI

# Tampilan Aplikasi



First Name

Last Name

Email

Create Update Delete Clear

Search By ID

Find

ID	First Name	Last Name	email
1	farhan	dwi	farhan@gmail.com
3	mufida	izzatul	muf@gmail.com
4	Akasyah	Arsyi	akash@gmail.com

# Contoh Action Event submitActionPerformed

```
private void submitActionPerformed(java.awt.event.ActionEvent evt) {  
    String fN,lN,em,query;  
  
    //connect to database  
    try{  
        Class.forName("com.mysql.cj.jdbc.Driver");  
  
        String url = "jdbc:mysql://localhost:3306/java_users_db";  
        String user = "root";  
        String pass = "";  
  
        Connection con = DriverManager.getConnection(url,user,pass);  
        Statement st = con.createStatement();  
  
        if("").equals(fName.getText())){  
            JOptionPane.showMessageDialog(new JFrame(), "First Name is Required", "Dialog", JOptionPane.ERROR_MESSAGE);  
        }else if("").equals(lName.getText())){  
            JOptionPane.showMessageDialog(new JFrame(), "Last Name is Required", "Dialog", JOptionPane.ERROR_MESSAGE);  
        }else if("").equals(email.getText())){  
            JOptionPane.showMessageDialog(new JFrame(), "Email is Required", "Dialog", JOptionPane.ERROR_MESSAGE);  
        }else{  
            fN = fName.getText();  
            lN = lName.getText();  
            em = email.getText();  
  
            query = "INSERT INTO user (first_name, last_name, email) VALUES ('"+fN+"','"+lN+"','"+em+"')";  
        }  
    }  
}
```

# Contoh Action Event submitActionPerformed



```
        st.executeUpdate(query);

        fName.setText("");
        lName.setText("");
        email.setText("");
        con.close();
    }
} catch (Exception e) {
    System.out.println("Error " + e.getMessage());
}
}
```



# Contoh Action Event loadData

```
public void loadData(){
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");

        String url = "jdbc:mysql://localhost:3306/java_users_db";
        String user = "root";
        String pass = "";

        Connection con = DriverManager.getConnection(url, user, pass);
        Statement st = con.createStatement();

        DefaultTableModel model = new DefaultTableModel(new String[]{"ID", "First Name", "Last Name", "email"}, 0);

        Table.setModel(model);
        String sql = "SELECT * FROM user";

        ResultSet rs = st.executeQuery(sql);
        String i, f, l, e;
        while(rs.next()){
            i = rs.getString("id");
            f = rs.getString("first_name");
            l = rs.getString("last_name");
            e = rs.getString("email");
            model.addRow(new Object[]{i, f, l, e});
        }
    } catch (Exception e) {
        System.out.println("Error " + e.getMessage());
    }
}
```

# Contoh Action updateActionPerformed

```
private void updateActionPerformed(java.awt.event.ActionEvent evt) {  
    String ID;  
    int notFound = 0;  
    String fN, lN, em;  
    try {  
        Class.forName("com.mysql.cj.jdbc.Driver");  
  
        String url = "jdbc:MySQL://localhost:3306/java_users_db";  
        String user = "root";  
        String pass = "";  
  
        Connection con = DriverManager.getConnection(url, user, pass);  
        Statement st = con.createStatement();  
  
        ID = serachData.getText();  
        if("").equals(ID)){  
            JOptionPane.showMessageDialog(new JFrame(), "ID is require", "Dialog",  
                JOptionPane.ERROR_MESSAGE);  
        }else {  
            String sql = "SELECT * FROM user WHERE id="+ID;  
            ResultSet rs = st.executeQuery(sql);  
            while(rs.next()){  
                notFound = 1;  
                fN = fName.getText();  
                lN = lName.getText();  
                em = email.getText();  
                String sql2 = "UPDATE user SET first_name='"+fN+"', last_name='"+lN+"', email='"+em+"' WHERE id="+ID;  
                st.executeUpdate(sql2);  
            }  
        }  
    }  
}
```



## Contoh Action updateActionPerformed

```
        loadData();  
        con.close();  
    }  
    if(notFound == 0){  
        JOptionPane.showMessageDialog(new JFrame(), "invalid ID", "Dialog",  
                                     JOptionPane.ERROR_MESSAGE);  
    }  
}  
} catch (Exception e){  
    System.out.println("Error " + e.getMessage());  
}  
}
```

# Contoh Action searchBtnActionPerformed

```
String ID;
int notFound = 0;
try {
    Class.forName("com.mysql.cj.jdbc.Driver");

    String url = "jdbc:mysql://localhost:3306/java_users_db";
    String user = "root";
    String pass = "";

    Connection con = DriverManager.getConnection(url, user, pass);
    Statement st = con.createStatement();

    ID = serachData.getText();
    if ("".equals(ID)) {
        JOptionPane.showMessageDialog(new JFrame(), "ID is require", "Dialog",
            JOptionPane.ERROR_MESSAGE);
    } else {
        String sql = "SELECT * FROM user WHERE id="+ID;
        ResultSet rs = st.executeQuery(sql);
        while(rs.next()){
            fName.setText(rs.getString("first_name"));
            lName.setText(rs.getString("last_name"));
            email.setText(rs.getString("email"));
            notFound = 1;
        }

        con.close();
    }
}
```

# Contoh Action deleteActionPerformed

```
String ID;
int notFound = 0;
try {
    Class.forName("com.mysql.cj.jdbc.Driver");

    String url = "jdbc:mysql://localhost:3306/java_users_db";
    String user = "root";
    String pass = "";

    Connection con = DriverManager.getConnection(url, user, pass);
    Statement st = con.createStatement();

    ID = serachData.getText();
    if("").equals(ID){
        JOptionPane.showMessageDialog(new JFrame(), "ID is require", "Dialog",
                                     JOptionPane.ERROR_MESSAGE);
    }else {
        String sql = "SELECT * FROM user WHERE id="+ID;
        ResultSet rs = st.executeQuery(sql);
        while(rs.next()){
            notFound = 1;
            String sql2 = "DELETE FROM user WHERE id="+ID;
            st.executeUpdate(sql2);
            loadData();
            con.close();
        }
        if(notFound == 0){
            JOptionPane.showMessageDialog(new JFrame(), "invalid ID", "Dialog",
```



## Contoh Action deleteActionPerformed

```
        JOptionPane.ERROR_MESSAGE);  
    }  
}  
} catch (Exception e) {  
    System.out.println("Error " + e.getMessage());  
}
```



## Contoh Action ClearActionPerformed

```
private void ClearActionPerformed(java.awt.event.ActionEvent evt) {  
    fName.setText("");  
    lName.setText("");  
    email.setText("");  
}
```



**Thank You**