

BAB 7 GUI-SWING 2

1.1 Tujuan Pembelajaran

Mahasiswa dapat memahami penggunaan dari GUI dan dapat mengimplementasikan GUI dengan baik dan benar.

1.2 Dasar Teori

1.2.1 GUI

GUI adalah singkatan dari Graphical User Interface. GUI merupakan desain aplikasi dengan tampilan visual sehingga pengguna dapat dengan mudah menggunakan aplikasi. The Java Foundation Class (JFC), merupakan bagian penting dari Java SDK, yang termasuk dalam koleksi dari API dimana dapat mempermudah pengembangan aplikasi JAVA GUI. JFC termasuk diantara 5 bagian utama dari API yaitu AWT dan Swing. Tiga bagian yang lainnya dari API adalah Java2D, Accessibility, dan Drag dan Drop. Semua itu membantu pengembang dalam mendesain dan mengimplementasikan aplikasi visual yang lebih baik.

AWT dan Swing menyediakan komponen GUI yang dapat digunakan dalam membuat aplikasi Java dan applet. Anda akan mempelajari applet pada bab berikutnya. Tidak seperti beberapa komponen AWT yang menggunakan native code, keseluruhan Swing ditulis menggunakan bahasa pemrograman Java. Swing menyediakan implementasi platform-independent dimana aplikasi yang dikembangkan dengan platform yang berbeda dapat memiliki tampilan yang sama. Begitu juga dengan AWT menjamin tampilan look and feel pada aplikasi yang dijalankan pada dua mesin yang berbeda menjadi terlihat sama. Swing API dibangun dari beberapa API yang mengimplementasikan beberapa jenis bagian dari AWT. Kesimpulannya, komponen AWT dapat digunakan bersama-sama dengan komponen Swing.

Nama dari komponen GUI milik Swing hampir sama persis dengan komponen GUI milik AWT. Perbedaan jelas terdapat pada penamaan komponen. Pada dasarnya, nama komponen Swing sama dengan nama komponen AWT tetapi dengan tambahan huruf J pada prefixnya. Sebagai contoh, satu komponen dalam AWT adalah Button class. Sedangkan pada Swing, nama komponen tersebut menjadi JButton class.

1.2.2 Komponen GUI pada Swing

Komponen GUI pada Swing terdapat dalam paket `javax.swing`. Berikut adalah daftar dari beberapa komponen Swing:

1. JFrame

Komponen ini merupakan library dari `javax.swing.JFrame`. Dalam penerapan window frame dengan menggunakan JFrame memiliki dua buah cara. Pertama menggunakan inisialisasi JFrame sebagai bentuk Object:

```
import javax.swing.JFrame;
import javax.swing.WindowConstants;

public class App {
    App() {
        //inisialisasi objek frame
        JFrame frame = new JFrame();

        //menampilkan jendela pada layar
        frame.setVisible(true);

        //menentukan posisi jendela pada tengah layar
        frame.pack();
        frame.setLocationRelativeTo(null);

        //menentukan ukuran width dan height ada jendela
        frame.setSize(300,300);

        //berikan judul pada jendela
        frame.setTitle("Contoh Window Frame");

        //menutup jendela secara penuh
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    }

    public static void main(String[] args) {
        new App();
    }
}
```

Kedua menggunakan inisialisasi JFrame sebagai Superclass dari Subclass tertentu:

```
import javax.swing.JFrame;
import javax.swing.WindowConstants;

public class App extends
JFrame { App() {

        setVisible(
true); pack();

        setLocationRelativeTo(n
ull); setSize(300,300);
setTitle("Contoh Window Frame");

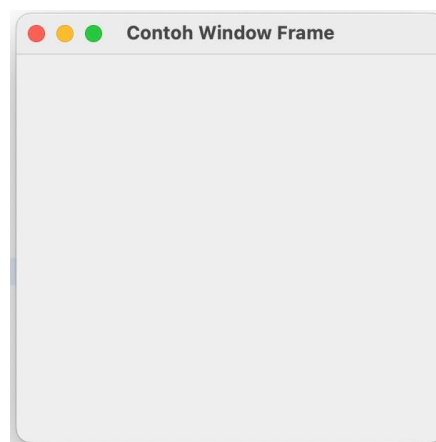
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);

    }

    public static void
main(String[] args) { new App();

    }

}
```



Method inherit pada [javax.swing.JFrame](#):

Meth od	Val ue
setResizable(boolean)	True or false
setBackground(Color)	Menggunakan library java.awt.Color Contoh: Color.BLUE
setIconImage(Image)	Menggunakan library javax.swing.ImageIcon
setSize(width, height)	Integer
setLocationRelativeTo()	
setTitle(string)	
setVisible(boolean)	True or false

2. Layout Frame

- Flow Layout

Kelas Java FlowLayout digunakan untuk mengatur komponen dalam satu baris, satu demi satu (dalam satu aliran). Ini merupakan tata letak default dari panel window frame. Library untuk menggunakan container ini ialah `java.awt.FlowLayout`:

```

import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.WindowConstants;

public class App extends JFrame {

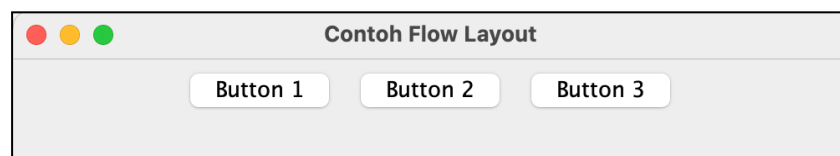
    App() {
        setVisible(true);
        pack();
        setLocationRelativeTo(null);
        setSize(500, 300);
        setTitle("Contoh Flow Layout");
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        GenerateUI(this);
    }

    public static void main(String[] args) {
        new App();
    }

    private void GenerateUI(App frame) {
        JButton button1 = new JButton("Button 1");
        JButton button2 = new JButton("Button 2");
        JButton button3 = new JButton("Button 3");

        frame.add(button1); frame.add(button2); frame.add(button3);
        frame.setLayout(new FlowLayout());
    }
}

```



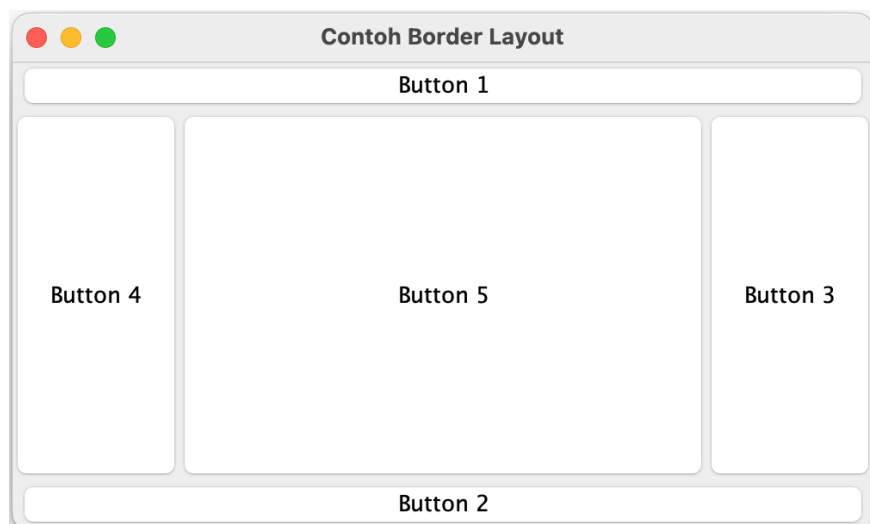
Penggunaan Align konstanta pada FlowLayout:

<code>frame.setLayout(new FlowLayout(FlowLayout.LEFT));</code>
<code>frame.setLayout(new FlowLayout(FlowLayout.RIGHT));</code>
<code>frame.setLayout(new FlowLayout(FlowLayout.CENTER));</code>
<code>frame.setLayout(new FlowLayout(FlowLayout.LEADING));</code>
<code>frame.setLayout(new FlowLayout(FlowLayout.TRAILING));</code>

- Border Layout

BorderLayout digunakan untuk mengatur komponen di lima wilayah: utara, selatan, timur, barat, dan tengah. Setiap wilayah (area) hanya dapat memuat satu komponen saja. Ini adalah tata letak default bingkai atau jendela. Penggunaan container ini dapat menggunakan library `java.awt.BorderLayout`

```
private void GenerateUI(App frame) {  
    JButton button1 = new JButton("Button 1");  
    JButton button2 = new JButton("Button 2");  
    JButton button3 = new JButton("Button 3");  
    JButton button4 = new JButton("Button 4");  
    JButton button5 = new JButton("Button 5");  
  
    frame.add(button1, BorderLayout.NORTH);  
    frame.add(button2, BorderLayout.SOUTH);  
    frame.add(button3, BorderLayout.EAST);  
    frame.add(button4, BorderLayout.WEST);  
    frame.add(button5, BorderLayout.CENTER);  
}
```

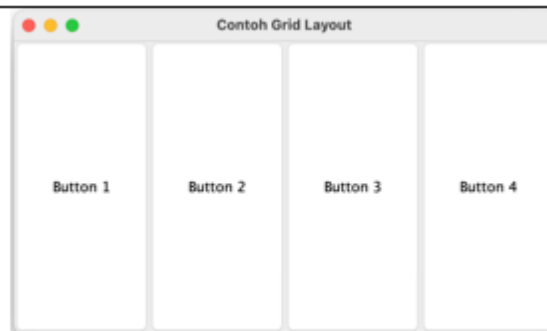


- Grid Layout

Kelas Java `GridLayout` digunakan untuk mengatur komponen dalam kotak persegi panjang. Satu komponen ditampilkan di setiap persegi panjang.

- GridLayout(): membuat tata letak kisi dengan satu kolom per komponen dalam satu baris.
- GridLayout(int rows, int column): membuat tata letak grid dengan baris dan kolom yang diberikan tetapi tidak ada celah di antara komponen.
- GridLayout(int baris, int kolom, int hgap, int vgap): membuat tata letak kisi dengan baris dan kolom yang diberikan bersama dengan celah horizontal dan vertikal yang diberikan.
- Konstruktor GridLayout() hanya membuat satu baris. Contoh berikut menunjukkan penggunaan konstruktor tanpa parameter.

```
private void GenerateUI(App frame) {
    JButton button1 = new JButton("Button 1");
    JButton button2 = new JButton("Button 2");
    JButton button3 = new JButton("Button 3");
    JButton button4 = new JButton("Button 4");
    frame.add(button1); frame.add(button2);
    frame.add(button3); frame.add(button4);
    frame.setLayout(new GridLayout());
}
```



- Non Layout

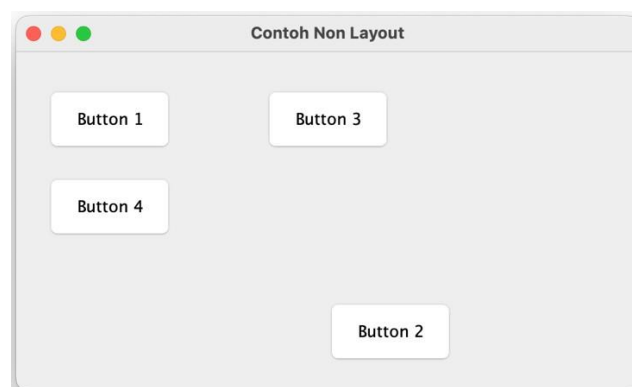
Meskipun dimungkinkan tanpa menggunakan tata letak, namun diharuskan menggunakan pengelola tata letak jika memungkinkan. Pengelola tata letak memudahkan untuk menyesuaikan tampilan komponen yang bergantung pada tampilan. Pengelola tata letak juga

dapat digunakan kembali dengan mudah oleh wadah lain, serta program lainnya.

Jika penampung menampung komponen yang ukurannya tidak terpengaruh oleh ukuran penampung atau oleh font, tampilan, atau perubahan label, maka pemosisian absolut mungkin masuk akal. Panel desktop, yang berisi bingkai internal, termasuk dalam kategori ini. Ukuran dan posisi bingkai internal tidak bergantung langsung pada ukuran panel desktop.

Namun, membuat wadah dengan wadah yang benar-benar diposisikan dapat menyebabkan masalah jika jendela yang berisi wadah diubah ukurannya. Berikut adalah contoh tata letak absolute atau tanpa menggunakan layouting:

```
private void GenerateUI(App frame) {  
    JButton button1 = new JButton("Button 1");  
    JButton button2 = new JButton("Button 2");  
    JButton button3 = new JButton("Button 3");  
    JButton button4 = new JButton("Button 4");  
    //set posisi X dan Y pada komponen button  
    button1.setBounds(25,30, 100, 50);  
    button2.setBounds(250,200, 100, 50);  
    button3.setBounds(200,30, 100, 50);  
    button4.setBounds(25,100, 100, 50);  
  
    frame.add(button1);  
    frame.add(button2);  
    frame.add(button3);  
    frame.add(button4);  
    frame.setLayout(null);  
}
```



3. JButton

Kelas JButton digunakan untuk membuat tombol berlabel yang memiliki implementasi independen platform. Hasil aplikasi dalam beberapa tindakan saat tombol ditekan. Itu mewarisi kelas AbstractButton.

Konstruktor yang biasa digunakan:

Constructor	Deskripsi
JButton()	Membuat button tanpa ada label
JButton(String s)	Membuat button dengan label text
JButton(Icon i)	Membuat button dengan objek icon

Metode kelas AbstractButton yang umum digunakan:

Method	Deskripsi
setText(String s)	Menambahkan label text pada button
getText()	Mengembalikan label text pada button
setEnabled(boolean b)	Mengatur disable pada button
setIcon(Icon a)	Menambahkan icon pada button
getIcon()	Mengembalikan icon pada button
setMnemonic(int a)	Mengatur mnemonic pada tombol
addActionListener(ActionListener a)	Menambahkan aksi klik pada tombol

Berikut adalah contoh penggunaan javax.swing.JButton:

```
private void GenerateUI(App
frame) { JButton button1 = new
JButton(); button1.setText("Klik
saya");

    button1.setIcon(new ImageIcon("D:\\btn_click.jpeg"));
    //memberikan aksi event klik
button1.addActionListener(new ActionListener() {

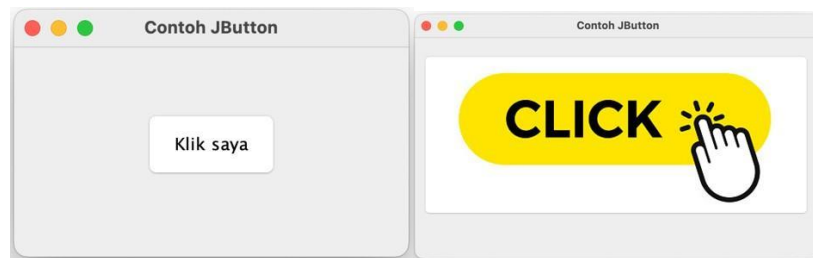
    @Override

    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Anda baru saja klik button");
    }

});

    //sampai disini

button1.setBounds(100,50, 100, 50);
```



4. JLabel

Objek kelas JLabel adalah komponen untuk menempatkan teks dalam frame. Ini digunakan untuk menampilkan satu baris teks. Teks dapat diubah oleh aplikasi tetapi pengguna tidak dapat mengeditnya secara langsung.

Konstruktor yang biasa digunakan:

Constructor	Deskripsi
JLabel()	Membuat instance JLabel dengan string kosong.
JLabel(String s)	Membuat instance JLabel dengan teks yang ditentukan.
JLabel(String s, Icon i, int horizontalAlignment)	Membuat instance JLabel dengan teks, gambar, dan perataan horizontal yang ditentukan.

Metode yang biasa digunakan:

Method	Deskripsi
setText(String s)	Menambahkan label text
getText()	Mengembalikan label text
setHorizontalAlignment(int alignment)	Mengatur posisi horizontal nilai X
getHorizontalAlignment()	Mengembalikan nilai posisi horizontal

Berikut adalah contoh penggunaan javax.swing.JLabel

```
private void GenerateUI(App frame) {  
  
    JLabel label1 = new JLabel("IBI  
Kesatuan"); label1.setBounds(10,20, 150,  
50);  
  
    JLabel label2 = new JLabel();  
label2.setText("Teknologi Informasi");  
label2.setBounds(10,70, 150, 50);  
  
    frame.add(label1);  
}
```



5. JPanel

JPanel adalah kelas kontainer yang paling sederhana. Container ini menyediakan ruang di mana aplikasi dapat melampirkan komponen lainnya.

Konstruktor yang biasa digunakan:

Constructor	Deskripsi
JPanel()	Membuat container panel
JPanel (LayoutManager x)	Membuat container panel dengan setingan LayoutManager

Berikut ini adalah contoh penggunaan javax.swing

```
private void ExpPanel(App frame) {  
  
    JPanel mainPanel = new  
JPanel(); mainPanel.setLayout(new  
BorderLayout());  
frame.setContentPane(mainPanel);  
  
    JPanel panel1 = new JPanel();  
panel1.setLayout(null);  
mainPanel.add(panel1, BorderLayout.NORTH);  
  
    JButton button1 = new JButton();  
button1.setBounds(10, 20, 475, 200);  
button1.setIcon(new ImageIcon("d://btn_click.jpeg"));  
panel1.add(button1);  
  
    JLabel lblCopyright = new JLabel("copyright IBIK @  
2022", SwingConstants.CENTER);  
  
    lblCopyright.setBackground(Color.BLUE);  
mainPanel.add(lblCopyright, BorderLayout.SOUTH);  
  
    frame.add(panel1);  
}
```



6. Jtext

- JTextField

Objek kelas JTextField adalah komponen teks yang memungkinkan pengeditan teks satu baris.

Konstruktor yang biasa digunakan:

Constructor	Deskripsi
JTextField()	Membuat textfield

JTextField (String s)	Membuat textfield dengan isi value spesifik
JTextField(String text, int columns)	Membuat textfield baru yang diinisialisasi dengan teks dan kolom yang ditentukan.

Metode yang biasa digunakan:

Method	Deskripsi
setText(String s)	Menambahkan value pada textfield
getText()	Mengembalikan value pada textfield
setFont(Font f)	Menentukan jenis font
setEditable(Boolean a)	Mengatur textfield menjadi readonly

Berikut ini adalah contoh penggunaan javax.swing

```
private void GenerateUI(App
frame) { JPanel panel1 = new
JPanel();
panel1.setLayout(null);

        JLabel lblFullname = new
JLabel("Fullname");
lblFullname.setBounds(22, 23, 80, 16);

        panel1.add(lblFullname);

        JTextField textFullname = new
JTextField(); textFullname.setBounds(105, 18,
169, 30); textFullname.setEditable(false);
textFullname.setText("Febry D F");
textFullname.setBackground(Color.LIGHT_GRAY);
panel1.add(textFullname);

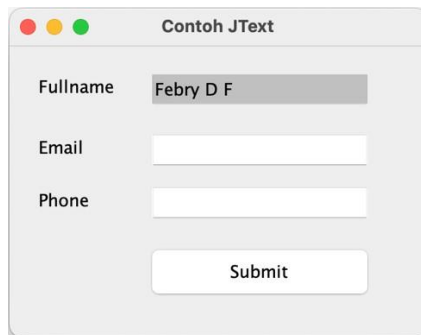
        JLabel lblEmail = new
JLabel("Email");
lblEmail.setBounds(22, 69, 61, 16);
panel1.add(lblEmail);

        JTextField textEmail = new
JTextField(); textEmail.setBounds(105,
64, 169, 30); panel1.add(textEmail);

        JLabel lblPhone = new
JLabel("Phone");
lblPhone.setBounds(22, 109, 61, 16);
panel1.add(lblPhone);

        JTextField txtPhone = new
JTextField(); txtPhone.setBounds(105,
104, 169, 30); panel1.add(txtPhone);

        JButton btnSubmit = new
JButton("Submit");
btnSubmit.setBounds(105, 152, 169, 40);
panel1.add(btnSubmit);
```



- JTextArea

Objek kelas JTextArea adalah wilayah multi baris yang menampilkan teks. Ini memungkinkan pengeditan teks beberapa baris.

Konstruktor yang biasa digunakan:

Constructor	Deskripsi
JTextArea ()	Membuat textarea
JTextArea (String s)	Membuat textarea dengan isi value spesifik
JTextField(String text, int columns)	Membuat textfield baru yang diinisialisasi dengan teks dan kolom yang ditentukan.

Metode yang biasa digunakan:

Method	Deskripsi
insert(String s, int position)	Memasukan spesifik text pada posisi tertentu
append(String s)	Menambahkan text kedalam akhir value

Berikut ini adalah contoh dari javax.swing.JTextArea:

```
private void GenerateUI(App
frame) { JPanel mainPanel = new
JPanel();

        mainPanel.setLayout(new
BorderLayout());
frame.setContentPane(mainPanel);

        JLabel lblWord = new JLabel("Masukan kalimat:");
mainPanel.add(lblWord, BorderLayout.NORTH);

        String word = "Hai";

        JTextArea textWord = new
JTextArea(); textWord.setText(word);

        textWord.insert(" kamu nanya?",
word.length()); textWord.append("\nBertanya-
tanya"); mainPanel.add(textWord,
BorderLayout.CENTER);

        JLabel lblCopyright = new JLabel("copyright IBIK @
2022", SwingConstants.CENTER);

        lblCopyright.setBackground(Color.BLUE)
```



- JPasswordField

Objek kelas JPasswordField adalah komponen teks khusus untuk entri kata sandi. Ini memungkinkan pengeditan satu baris teks.

Konstruktor yang biasa digunakan:

Constructor	Deskripsi
JPasswordField()	Membuat textarea
JPasswordField(String s)	Membuat textarea dengan isi value spesifik
JPasswordField(String text, int columns)	Membuat textfield baru yang diinisialisasi dengan teks dan kolom yang ditentukan.

Berikut ini adalah contoh penggunaan javax.swing.JPasswordField:

```
private void GenerateUI (App
frame) { JPanel mainPanel = new
JPanel();

        mainPanel.setLayout (new
BorderLayout());
frame.setContentPane (mainPanel);

        JPanel panel1 = new JPanel();
panel1.setLayout (null);
mainPanel.add (panel1, BorderLayout.NORTH);

        JLabel lblUsername = new
JLabel ("Username");
lblUsername.setBounds (22, 23, 80, 16);
panel1.add (lblUsername);

        JPasswordField textUsername = new
```

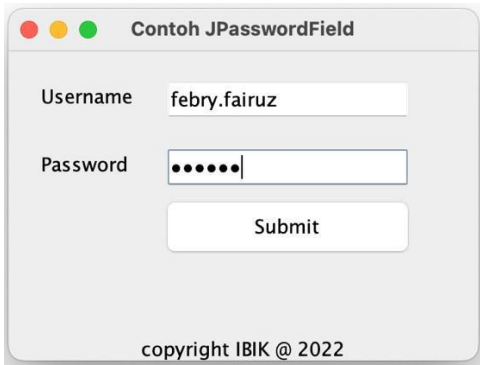
```
panel1.add(lblPassword);

JPasswordField textPassword = new JPasswordField();
textPassword.setBounds(105, 64, 169, 30);
panel1.add(textPassword);

JButton btnSubmit = new
JButton("Submit");
btnSubmit.setBounds(105, 100, 169, 40);
panel1.add(btnSubmit);

JLabel lblCopyright = new JLabel("copyright IBIK @
2022", SwingConstants.CENTER);

lblCopyright.setBackground(Color.BLUE)
; lblCopyright.setSize(300, 50);
mainPanel.add(lblCopyright, BorderLayout.SOUTH);
```



7. JCheckBox

Kelas JCheckBox digunakan untuk membuat kotak centang. Ini digunakan untuk mengaktifkan opsi (benar) atau menonaktifkan (salah). Mengklik Kotak Centang akan mengubah statusnya dari "on" menjadi "off" atau dari "off" menjadi "on".

Konstruktor yang biasa digunakan:

Constructor	Deskripsi
JCheckBox()	Membuat tombol kotak centang yang awalnya tidak dipilih tanpa teks
JCheckBox(String s)	Membuat kotak centang yang awalnya tidak dipilih dengan teks.
JCheckBox (String text, boolean selected)	Membuat kotak centang dengan teks dan menentukan apakah itu awalnya dipilih

atau tidak.

Berikut ini adalah contoh dari penggunaan javax.swing.JCheckBox:

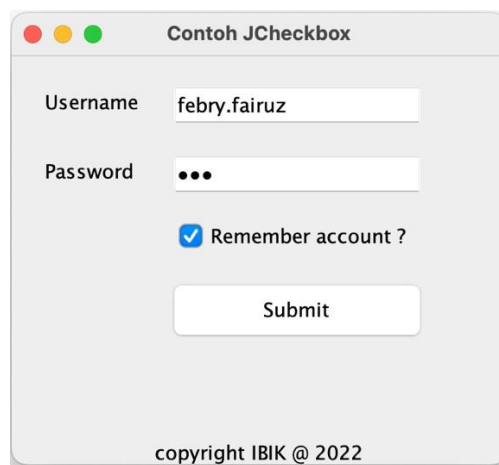
```
private void GenerateUI (App
frame) { JPanel mainPanel = new
JPanel();

        mainPanel.setLayout (new
BorderLayout());
frame.setContentPane (mainPanel);

        JPanel panel1 = new JPanel();
panel1.setLayout (null);
mainPanel.add (panel1, BorderLayout.NORTH);

        ...

        JCheckBox chkRemember = new JCheckBox ("Remember
account ?"); chkRemember.setBounds (105, 100, 169, 40);
panel1.add (chkRemember);
```



8. JRadioButton

Kelas JRadioButton digunakan untuk membuat tombol radio. Digunakan untuk memilih satu pilihan dari beberapa pilihan.

Konstruktor yang biasa digunakan:

Constructor	Deskripsi
JRadioButton()	Membuat tombol radio button yang awalnya tidak dipilih tanpa teks
JRadioButton(String s)	Membuat radio button yang awalnya

	tidak dipilih dengan teks.
JRadioButton(String text, boolean selected)	Membuat radio button dengan teks dan menentukan apakah itu awalnya dipilih atau tidak.

Berikut ini adalah contoh dari penggunaan javax.swing.JRadioButton:

```
private void GenerateUI(App
frame) { JPanel mainPanel = new
JPanel();

        mainPanel.setLayout(new
BorderLayout());
frame.setContentPane(mainPanel);

        JPanel panel1 = new JPanel();
panel1.setLayout(null);
mainPanel.add(panel1,BorderLayout.NORTH);

        ...

        JLabel lblGender = new
JLabel("Gender");
lblGender.setBounds(22, 150, 61, 16);
panel1.add(lblGender);

        JRadioButton radioGender1 = new
JRadioButton("Female"); radioGender1.setBounds(105, 145, 100,
30); panel1.add(radioGender1);

        JRadioButton radioGender2 = new
JRadioButton("Male"); radioGender2.setBounds(200,
145, 100, 30); panel1.add(radioGender2);

        ButtonGroup bg=new ButtonGroup();
bg.add(radioGender1);bg.add(radioGender2);

        ...
```

9. JComboBox

Objek kelas select box digunakan untuk menampilkan menu popup pilihan. Pilihan yang dipilih oleh pengguna ditampilkan di bagian atas menu.

Konstruktor yang biasa digunakan:

Constructor	Deskripsi
JComboBox()	Membuat inputan select box yang dengan default data model
JComboBox (Object[] item)	Membuat inputan select box yang dengan default data model object
JComboBox (Array item)	Membuat inputan select box yang dengan default data model array

Metode yang biasa digunakan:

Method	Deskripsi
addItem(Object item)	Menambahkan item pada combobox
removeItem(Object item)	Menghapus salah satu item pada combobox
removeAllItems()	Menghapus semua item pilihan pada combobox

Berikut ini adalah contoh dari penggunaan javax.swing.JComboBox:

```
private void GenerateUI(App
frame) { JPanel mainPanel = new
JPanel();

        mainPanel.setLayout(new
BorderLayout());
frame.setContentPane(mainPanel);

        ...

        JLabel lblCityzen= new
JLabel("Citizenship");
```

```

panel1.add(lblCityzen);

String
country[]={ "Indonesia", "India", "Aus", "U.S.A", "England", "Newzealand" };

JComboBox selectCitizen = new
JComboBox(country); selectCitizen.setBounds(105, 180,
169, 30); panel1.add(selectCitizen);

...

```

10. JTable

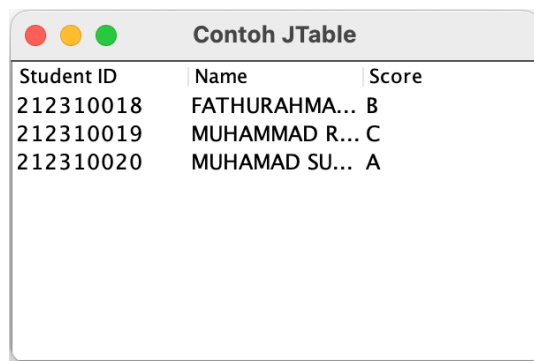
Kelas JTable digunakan untuk menampilkan data dalam bentuk tabel. Ini terdiri dari baris dan kolom.

Konstruktor yang biasa digunakan:

Constructor	Deskripsi
JTable()	Membuat tabel tanpa ada cell
JTable(Object[] rows, Object[] cols)	Membuat table dengan spesifik data

Berikut ini adalah contoh penggunaan javax.swing.JTable:

```
private void GenerateUI(App frame) {  
  
    String data[][]={ {"212310018","FATHURAHMAN AL  
FARIDZI","B"},  
  
                      {"212310019","MUHAMMAD  
RAFI ZUHAIR ARTA","C"},  
  
                      {"212310020","MUHAMAD  
SUBHAN RIZKI AFIA","A"}};  
  
    String column[]={"Student ID","Name","Score"};  
  
    JTable jt=new  
JTable(data,column);  
jt.setBounds(30,40,200,300);  
  
    JScrollPane sp=new JScrollPane(jt);  
  
    frame.add(sp);  
}
```



Student ID	Name	Score
212310018	FATHURAHMA...	B
212310019	MUHAMMAD R...	C
212310020	MUHAMAD SU...	A

1. JMenu

Kelas JMenuBar digunakan untuk menampilkan menu bar pada window frame. Dan memungkinkan memiliki beberapa menu. Untuk menambahkan menu pada JMenuBar dapat menggunakan objek kelas JMenu, JMenu adalah komponen menu pull down yang ditampilkan dari bagian menu. Sedangkan untuk membuat sub menu dapat menggunakan objek kelas JMenuItem, kelas ini menambahkan item menu berlabel sederhana. Item yang digunakan dalam menu harus milik JMenuItem atau subkelasnya.

Berikut adalah contoh penggunaan JMenuBar, JMenu dan JMenuItem:

```
import java.awt.FlowLayout;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;

public class App extends JFrame {

    App() {

        setVisible(true)
        ; setSize(500,200);
        setTitle("Contoh JMenu");

        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);

        setLayout(new FlowLayout());

        GenerateUI(this);

    }

    public static void
    main(String[] args) { new App();

    }

    private void GenerateUI(App frame) {

        JMenuBar menuBar = new JMenuBar();

        JMenu fileMenu = new
        JMenu("File"); JMenu editMenu = new
        JMenu("Edit"); JMenu helpMenu = new
        JMenu("Help");
```

```

        menuBar.add
        (fileMenu);
        menuBar.add(editMenu
        );
        menuBar.add(helpMenu
        );

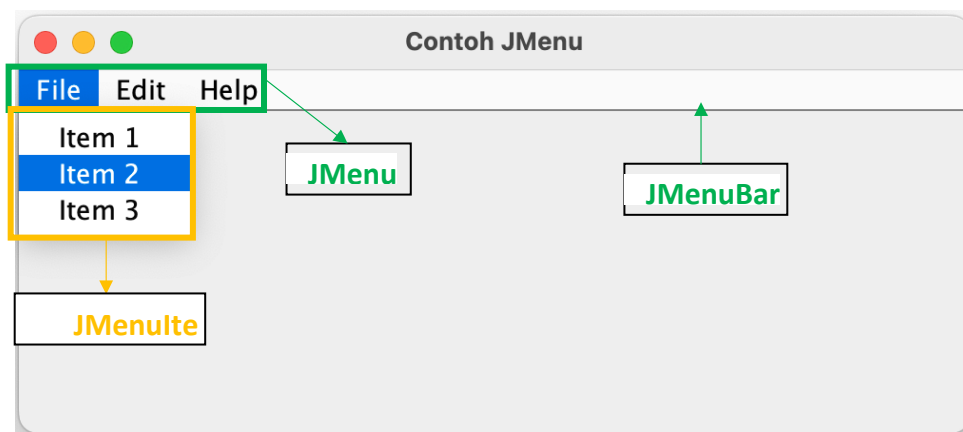
        JMenuItem item1 = new
        JMenuItem("Item 1"); JMenuItem item2 =
        new JMenuItem("Item 2"); JMenuItem item3
        = new JMenuItem("Item 3");

        fileMenu.
        add(item1);
        fileMenu.add(item2
        );
        fileMenu.add(item3
        );

        frame.setJMenuBar(menuBar);

    }
}

```



2. JFileChooser

Objek kelas JFileChooser merupakan bagian dari jendela dialog tempat pengguna dapat memilih file.

Konstruktor yang biasa digunakan:

Constructor	Deskripsi
JFileChooser()	Membuat JFileChooser yang menunjuk ke direktori default pengguna.
JFileChooser(File currentDir)	Membuat JFileChooser menggunakan File yang diberikan sebagai jalur.

Berikut ini adalah contoh penggunaan javax.swing.JFileChooser:

```
private void
GenerateUI(App frame) {
    JMenuBar menuBar = new
    JMenuBar();

    String lblMenu[] = {"File", "Edit", "Help"};
    String lblSubMenuFile[] = {"New File", "Open
File", "Exit"};

    JMenu menus[] = new
    JMenu[lblMenu.length]; for (int i =
    0; i < menus.length; i++) {

        menus[i]
        = new JMenu();
        menus[i].setText(lblM
        enu[i]);
        menuBar.add(menus[i])
        ;

    }

    //add submenu for menu File

    JMenuItem subMenu[] = new
    JMenuItem[lblSubMenuFile.length]; for (int i = 0; i <
    subMenu.length; i++) {

        subMenu[i] = new
        JMenuItem();
        subMenu[i].setText(lblSubMenuFi
        le[i]);
        menus[0].add(subMenu[i]);

    }

    //sub menu New File
    subMenu[0].addActionListener(new
    ActionListener() {

        @Override

        public void actionPerformed(ActionEvent e) {
```

```

        File file = new
File(fileChooser.getSelectedFile().getAbsolutePath());

        System.out.println(file);

    }else{

        System.out.println("Tidak ada file
yang dipilih");
    }

}

});

//submenu Open File
subMenu[1].addActionListener(new ActionListener() {

    @Override

    public void actionPerformed(ActionEvent e) {

        JFileChooser fileChooser = new

JFileChooser(); fileChooser.setCurrentDirectory(new

File("."));

        //memilih file yang akan dibuka

        int isChosen =

fileChooser.showOpenDialog(null);

        //jika file terpilih

        if(isChosen ==

JFileChooser.APPROVE_OPTION){ File file =

new

File(fileChooser.getSelectedFile().getAbsolutePath());

        System.out.println(file);

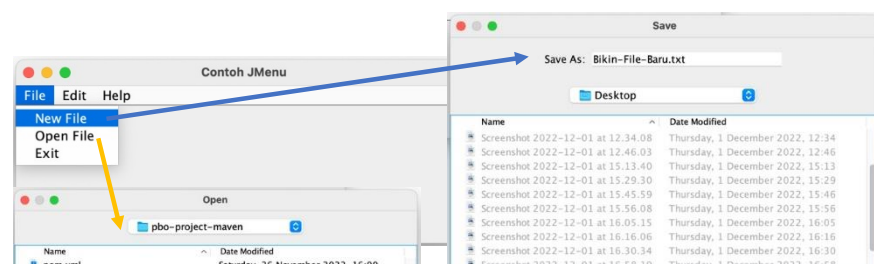
    }else{

        System.out.println("Tidak ada file
yang dipilih");
    }

}

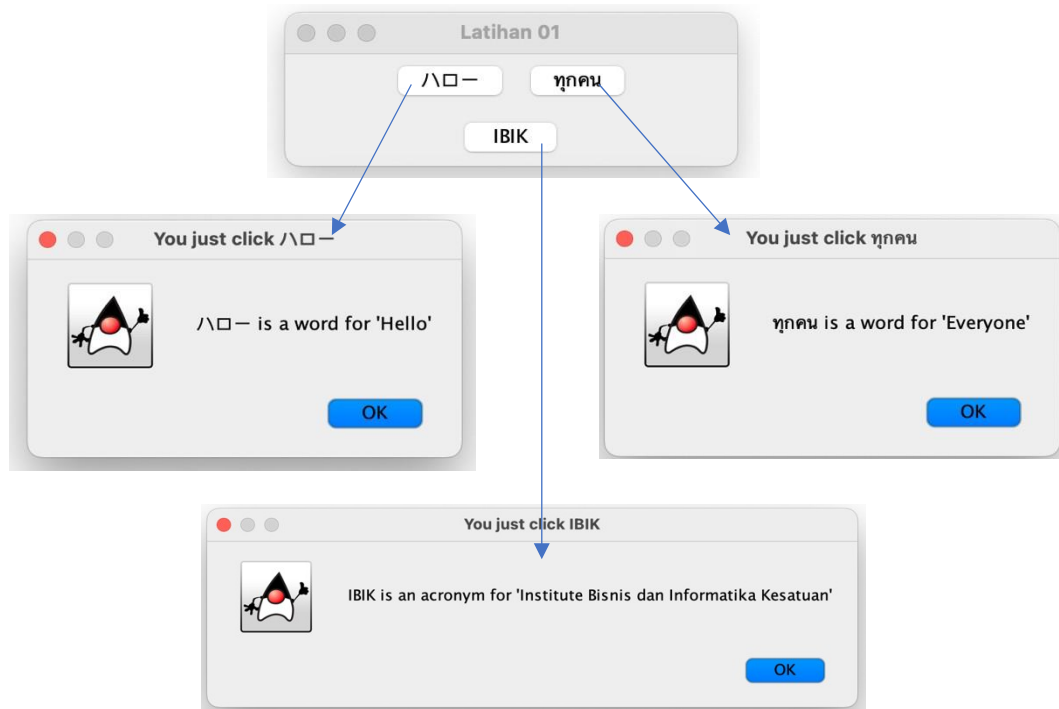
});

```



7.3 Latihan Pembelajaran

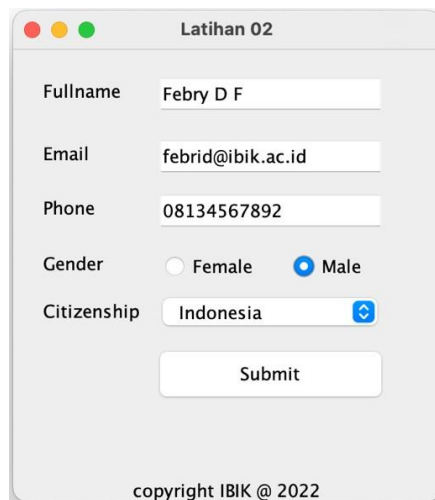
1. Buatlah project maven dengan nama PBO-NPM-Pembelajaran-7 dan package groupid bernama com.ibik.pbo.Pembelajaran.
2. Buatlah program seperti dibawah ini, dengan kondisi tertentu:



Pada gambar diatas terdapat 1 buah Frame, 3 buah Button, dan 1 buah Kotak Dialog. Jika salah satu button pada window frame di klik, maka akan menampilkan kotak dialog yang berisikan pesan informasi dari masing-masing button (sesuai panah pada gambar diatas).

Simpanlah dengan nama kelas Latihan01.

3. Buatlah program berdasarkan gambar dibawah ini:



Buatlah sebuah logic, jika mengklik tombol SUBMIT akan mengeksekusi logic dibawah ini:

- Jika salah satu isian pada komponen form tersebut tidak diisi maka menampilkan Kotak Dialog pesan kesalahan “Data tidak boleh kosong. Silakan periksa kembali isian anda.”
- Jika seluruh isian komponen form terpenuhi maka menampilkan seluruh data atau informasi tersebut dalam bentuk Kotak Dialog. Simpanlah dengan nama kelas Latihan02.

4. Buatlah sebuah form isian yang terdiri dari Email, Password dan Remember Password (dalam bentuk checkbox). Jika mengklik tombol Login, maka akan mengikuti kondisi dibawah ini:

- Jika salah satu isian tidak terpenuhi maka akan menampilkan kotak dialog pesan kesalahan “*Silahkan mengisi data dengan benar*”
- Inputan Email harus bernilai email anda, dan inputan Password harus bernilai NPM anda. Jika salah satu tidak terpenuhi maka akan menampilkan kotak dialog pesan kesalahan “*Data yang anda masukan salah*” namun Jika semua kondisi benar maka menampilkan kotak dialog “*Selamat Datang [Nama Anda]*”.

Simpanlah dengan nama kelas Latihan03.

5. Buatlah program berdasarkan gambar dibawah ini:

NPM	Nama	Nilai
212310018	FATHURAHMAN AL FARIDZI	B
212310019	MUHAMMAD RAFI ZUHAIR ARTA	C
212310020	MUHAMAD SUBHAN RIZKI ARIA	A

Pada gambar rancangan diatas terdapat beberapa komponen yang dapat digunakan diantaranya:

- JFrame
- JLabel
- JTextField
- JRadioButton
- JButton
- JTable
- JMenu

Simpanlah dengan nama kelas Latihan04.