Modul IV
Komponen Swing Lanjutan

Tujuan

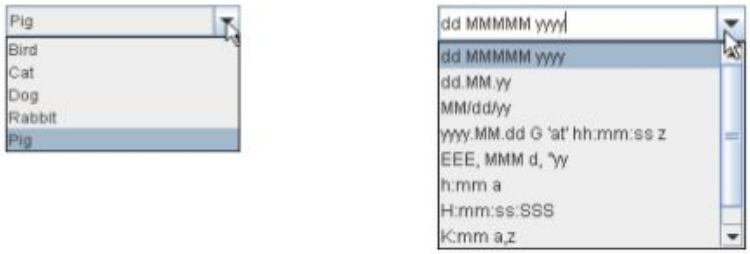
1. Mahasiswa dapat memahami dan menerapakan penggunaan komponen combobox, listbox, scrollpane, textarea.
2. Mahasiswa dapat membuat menu pada aplikasi java desktop.

Dasar Teori

Swing toolkit menyediakan banyak sekali komponen untuk membangun aplikasi GUI desktop berbasis java. Sebagian komponen dasar telah kita pelajari dalam praktikum sebelumnya.

Komponen jComboBox dan jListBox

JComboBox memerlukan tempat yang minimalis dibandingkan dengan JRadioButton, selain itu JComboBox mempunyai bentuk ComboBox yang dapat diedit, sehingga memungkinkan user untuk memilih pilihan yang tidak ada dalam item JComboBox.



Gambar 4.1 Contoh ComboBox

JList memungkinkan multiple selection dengan menekan tombol : SHIFT + Left Click atau CTRL + Left Click. Kemampuan ini membantu user jika harus melakukan multiple selection.

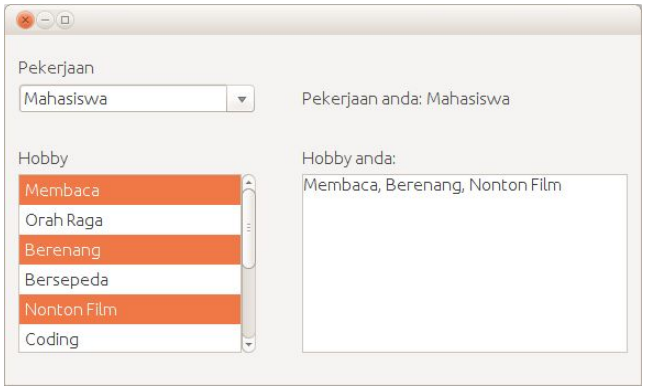
JComboBox dan JList sangat fleksibel, kita dapat menambah dan menghapus item di dalamnya dengan sangat mudah. Sehingga cocok digunakan untuk merepresentasikan pilihan yang item pilihannya bersifat dinamis.

Komponen jScrollPane dan jTextArea

- **JScrollPane** merupakan komponen yang digunakan untuk menampung komponen yang besarnya melebihi dimensi JScrollPane itu sendiri.
- **JTextArea** merupakan komponen yang digunakan untuk menerima masukan tulisan yang boleh lebih dari satu baris.

Contoh Program

Membuat program sederhana menggunakan komponen jListBox, jComboBox, jScrollPane dan jTextArea.



Gambar 4.2 Contoh Program

Bagian pertama program ini terdapat sebuah JComboBox dan JLabel, setiap kali item di dalam JComboBox dipilih, JLabel di sebelahnya akan menampilkan item yang dipilih tersebut.

Bagian kedua program ini terdapat sebuah JList dan JTextArea. Setiap kali item-item di dalam JList dipilih, JTextArea akan menampilkan item-item yang dipilih tersebut dipisahkan dengan koma (,).

Langkah membuat program:

1. Buat project baru dan tambahkan class jFrame.
2. Buatlah tampilan seperti contoh diatas, dengan menambahkan beberpa komponen:
   a) jLabel (4 buah)
   b) jCombobox (1 buah)
   c) jListBox (1 buah)
   d) jTextArea (1 buah)
   e) jScrollPane (2 buah)
3. Ubah property masing-masing komponen

Komponen	Properties
jLabel1, jLabel2, jLabel3, jLabel4	**Text:** Pekerjaan, Hoby, Pekerjaan anda:, Hobby anda:
jTextArea1	**Text**: [kosongkan] **VariableName**: txtHoby
jComboBox1	**Text**: (tidak diubah) **VariableName**: cmbPekerjaan **Model:** klik pada tombol **[...]**

	Pada bagian model di dalam Jendela Properties masukkan item: Pelajar, Mahasiswa, Programmer, Technical Writer dan Tester.
jList1	**Text**: (tidak diubah) **VariableName**: lstJurusan **Model:** klik pada tombol **[...]** Pada bagian model di dalam Jendela Properties masukkan item: Membaca, Orah Raga, Berenang, Bersepeda, Nonton Film, Coding, Makan, Tidur dan Shopping

4. Menangani pemilihan jComboBox. Klik kanan jComboBox di Jendela Design, kemudian pilih menu: **Event -> Action -> actionPerformed**, kemudian ketik kode berikut:

```
    private void cmbPekerjaanActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        lblPekerjaan.setText("Pekerjaan anda: " + cmbPekerjaan.getSelectedItem());
    }
```

method getSelectedItem dari JComboBox digunakan untuk memperoleh item yang sedang di pilih dalam JComboBox

5. Menangani event pemilihan dari JList. Event yang digunakan untuk menangani pemilihan item dari JList berbeda dengan JComboBox. JList akan mengaktifkan ListSelection event ketika user memilih item dalam JList. Untuk menangani event ini, klik kanan pada jList, kemudian pilih menu **Event -> ListSelection -> valueChanged**, kemudian ketik kode beriku:

```
private void lstHobbyValueChanged(javax.swing.event.ListSelectionEvent evt) {
    // TODO add your handling code here:
    Object[] selectedItems = lstHobby.getSelectedValues();

    if (selectedItems == null || selectedItems.length == 0) {
        txtHobby.setText(null);
    } else {
        StringBuilder strValues = new StringBuilder();
        for(Object item : selectedItems) {
            StringBuilder append = strValues.append(item.toString()).append(", ");
        }
        txtHobby.setText(strValues.substring(0, strValues.length() - 2));
    }
}
```

Method getSelectedValues dari JList mengembalikan item-item yang terpilih.

Komponen Menu dan Toolbar

Menu dan Toolbar digunakan untuk melakukan navigasi dalam aplikasi. Dengan penggunaan menu dan toolbar tersebut, navigasi dalam aplikasi menjadi lebih fleksibel dan mudah digunakan oleh user. Menu dan Toolbar pada umumnya diletakkan dibagian atas aplikasi agar mudah ditemukan oleh user.

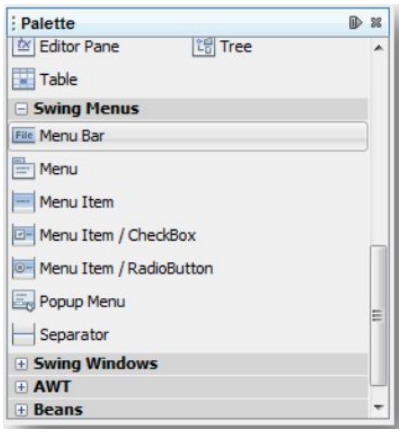
Selain menu ada juga contextual menu (popup) yang bisa muncul dimana saja sesuai dengan konteks aplikasi.

Bekerja dengan Menu dalam Java melibatkan beberapa komponen swing, antara lain:

- **jMenuBar**, class yang digunakan untuk menampung jMenu.
- **jPopupMenu**, merupakan komponen tempat menu yang dapat diterapkan pada komponen lain
- **jMenu**, merupakan komponen menu yang dapat berisikan menu lain. jMenu dapat menampung satu atau lebih JMenuItem
- **jMenuItem**, merupakan bagian terluar dari struktur menu yang tidak bisa mempunyai child
- **JCheckBoxMenuItem**, Ujung dari menu, namun bentuknya lebih mirip JCheckBox.
- **JRadioButtonMenuItem,** Ujung dari menu, namun bentuknya lebih mirip JButton.
- **JSeparator,** pemisah antar JMenuItem atau antar JMenu.

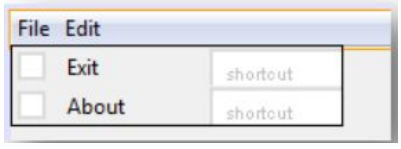
Membuat Menu

1. Buat sebuah class jFrame dan beri nama ToolbarMenu.java
2. Tambahkan jMenuBar kedalam jFrame dengan cara meng-drag MenuBar ke dalam form. MenuBar terletak di dalam *pallete* dan berada dalam kategori ***Swing Menus***



Gambar 4.3 Pallete

3. Tambahkan MenuItem kedalam kedalam Menubar sebanyak 2 buah. Ganti property text masing-masing MenuItem menjadi **Exit** dan **About**.



Gambar 4.4 Desain Menu

4. Tambahkan kode pada MenuItem Exit (menggunakan **Event -> Action -> actionPerformed**)

```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    dispose();
}
```

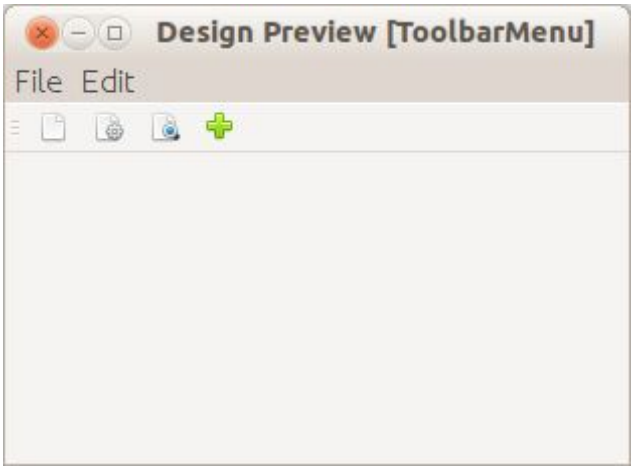
5. Tambahkan kode pada MenuItem About (menggunakan **Event -> Action -> actionPerformed**)

```
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JOptionPane.showMessageDialog(null, "Ini adalah aplikasi GUI berbasis java");
}
```

6. Build dan Run Aplikasi tersebut. Klik menu File dan pilih About. Klik menu File dan pilih Exit.

Membuat Toolbar

Toolbar memberikan dimensi lain dalam mengakses menu dbandingkan menu ataupun popup menu. Pada umumnya Toolbar merupakan cara singkat untuk mengakses menu. Menu yang diwakili toolbar adalah menu yang bersifat umum dan tidak terikat pada konteks tertentu.



Gambar 4.5 Form Design

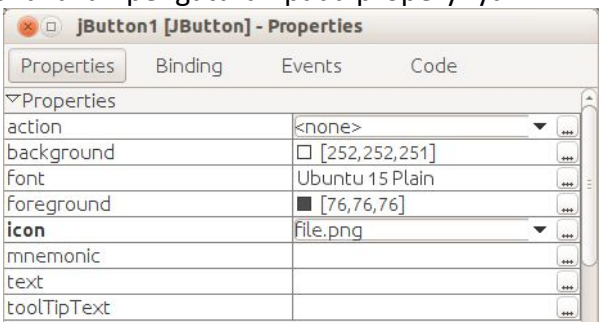
Dalam contoh program diatas kita akan membuat sebuah JToolBar dengan 4 buah JButton yang telah didekorasi dengan icon cantik. Icon yang digunakan banyak tersedia di internet, format file yang dipilih adalah .png, karena format file ini paling bagus dalam menangani transparasi komponen.

Sebelum mulai membuat JToolBar, kita perlu mempersiapkan terlebih dahulu icon yang akan digunakan sebagai dekorasi JButton. Ikuti langkah-langkah berikut ini:

1. Buatlah sebuah java package baru untuk menampung semua icon yang akan digunakan. caranya klik kanan di jendela Projects bagian nama project, pilih menu **New -> Java Package**, berinama images.
2. Masukkan icon kedalam package yang telah dibuat. Untuk memasukkan image

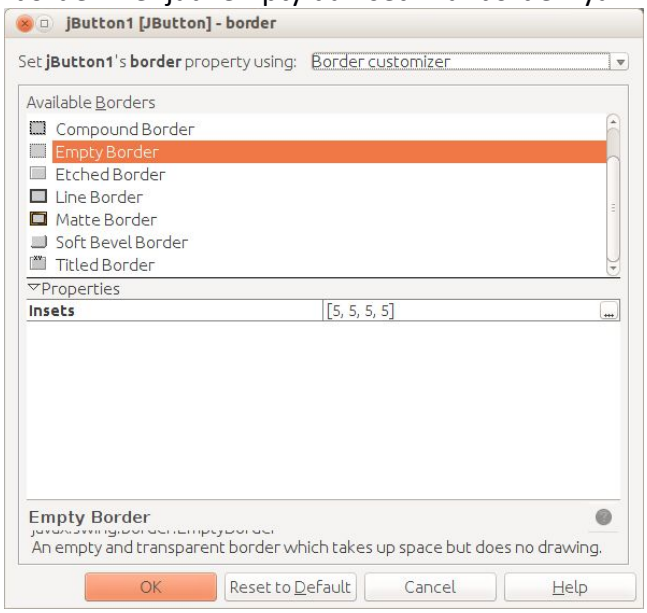
kedalam package, perlu diketahui lokasi project disimpan, misal project tersimpan pada folder **c:\latihan\javaswing**, buka folder tersebut dan letakkan file icon/gambar pada folder **c:\latihan\javaswing\src\images**.

3. Build project. Langkah ini diperlukan untuk mengcompile semua file .java menjadi file .class. Selain itu proses ini juga akan mengkopi file selain file .java (termasuk file icon) ke dalam folder build\classes. Jika proses ini tidak dilaksanakan, maka ketika program dijalankan, file icon tidak akan ditemukan dan program menjadi error
4. Tambahkan objek jToolBar kedalam jFrame, ganti namanya menjadi toolbar.
5. Tambahkan jButton kedalam jToolbar, sesuaikan properti dan variablenya.
6. Mendekorasi tampilan jButton dengan menambahkan icon agar terlihat cantik, untuk itu perlu dilakukan pengaturan pada properynya:



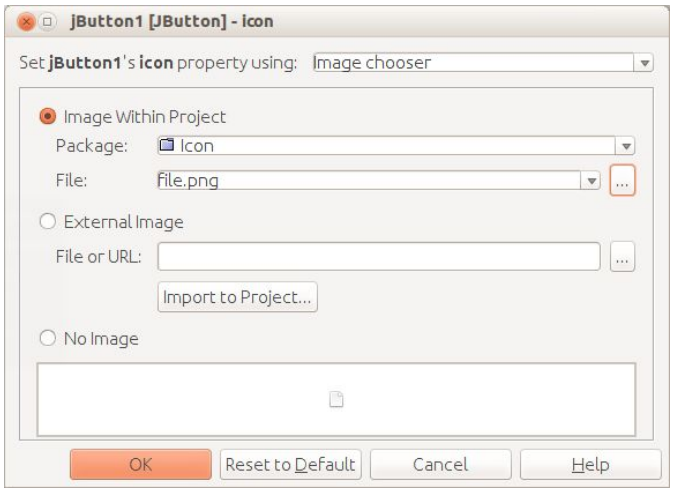
Gambar 4. Jendela Property jButton

a) Text: (kosongkan)
b) Border: pilih border menjadi empty dan set nilai bordernya menjadi [5,5,5,5].



Gambar 4. Jendela Border Editor

c) Icon: ganti icon dengan icon yang telah disiapkan sebelumnya.



Gambar 4. Jendela Icon Chooser

7. Compile dan jalankan aplikasi untuk melihat hasilnya.

Latihan

1. Tambahkan PopUp Menu pada aplikasi yang telah dibuat sebelumnya.
2. Tambahkan action/event pada toolbar yang telah dibuat sebelumnya.

Laporan Praktikum

1. Buatlah laporan praktikum berupa dokumen.
2. Jelaskan perbedaan scrollbar dengan scrollpane
3. Buatlah program sederhana yang mengaplikasikan penggunaan scrollbar dan scrollpane.