Bahasa Pemrograman

(Pemrograman Visual dengan Java)

Modul Praktikum

# Topik

Penggunaan java Swing JDesktopPane (Swing Control) dan JInternalFrame (swing containers) pada aplikasi dengan menggunakan Netbeans IDE.

# Tujuan

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

1. Membuat aplikasi GUI dengan menggunakan JDesktopPane dengan Netbeans IDE.
2. Membuat aplikasi GUI dengan menggunakan JDesktopPane untuk menjalankan JDialog di dalamnya dengan Netbeans IDE.
3. Membuat aplikasi GUI dengan menggunakan JDesktopPane untuk menjalankan JInternalFrame di dalamnya dengan Netbeans IDE.
4. Membuat aplikasi GUI dengan menggunakan JDesktopPane untuk menjalankan JInternalFrame yang dibuat pada class terpisah di dalamnya dengan Netbeans IDE

# Alat dan Bahan

* Komputer dengan **Java Development Kit (JDK)** dan Netbeans IDE sudah terinstall
* Informasi download dan instalasi JDK dan Netbeans dapat diperoleh di; https://netbeans.org/downloads/index.html

# Rangkuman/Teori

## [Menjelaskan cara membuat internal Frame (frame di dalam frame) pada aplikasi menggunakan JDesktopPane dan JinternalFrame dengan Netbeans IDE]

### JDesktopPane

JDesktopPane Class digunakan untuk membuat multiple-document interface (MDI) atau virtual desktop. JDesktopPane biasa digunakan sebagai parent (induk) dari JInternalFrame, JDesktopPane berfungsi sebagai Desktop anager yang memfasilitasi pemasangan atau JInternalFrame di dalamnya.

### JInternalFrame

JInternalFrame merupakan lightweight frame (frame sederhana) yang biasanya digunakan di dalam JDesktopPane.

Terdapat dua cara untuk menggunakan JInternalFrame

* Menggunakan **Internal Frame dalam file class yang sama dengan desktopPane**. Biasanya cukup dengan menambahkan JinternalFrame (Internal Frame) (dari palette 🡪 swing containers 🡪 Internal Frame) ke dalam desktopPane.
* Menggunakan **Internal Frame dari file class terpisah dari file class tempat desktopPane**, dengan membuat suatu file class dengan jenis JintenalFrame secara terpisah kemudian class jenis JInternalFrame tersebut dipanggil untuk tampil dan digunakan di dalam desktopPane.

### Membuat Aplikasi dengan Menggunakan JinternalFrame dalam File Class yang Sama dengan File Class Tempat JDesktopPane

Untuk membuat JInternalFrame di dalam desktopPane pada aplikasi berbasis java biasanya digunakan widget-widget berikut:



Gambar 1. Widget Desktop Pane dan Widget Internal Frame.

1. Masukkan Desktop Pane ke Frame Aplikasi (click & drag Desktop Pane dari Swing Controls ke Frame Aplikasi).



Gambar 2. JDesktopPane di dalam JFrame aplikasi.

1. Ubah variabel name **JDesktopPane** (pada tab code properties) sesuai kegunaan (misal: **variable name: dpUtama**)
2. Masukkan Internal Frame ke JDesktopPane (click & drag Inetrnal Frame dari Swing Containers ke desktop), Atur ukuran InternalFrame sesuai kebutuhan.



Gambar 3. JInternalFrame di dalam JDesktopPane

1. Ubah variabel name **JInternalFrame** (pada tab code properties) sesuai kegunaan (misal: **variable name: ifrAplikasi1**).
2. Ubah title JInternalFrame (pada tabel properties) sesuai kegunaan (misal: **Title: Aplikasi 1**)
3. Tambahkan Button ke JInternalFrame untuk tombol keluar dari JInternalFrame.
4. Ubah variabel name JButton (pada tab code properties) sesuai kegunaan (misal: **variable name: btnKeluar**).
5. Ubah text pada Button (pada tabel properties) sesuai kegunaan (misal: **Text: KELUAR**)
6. Agar **btnKeluar** dapat berfungsi menutup internalFrame saat di click, tambahkan kode program untuk menutup ifrAplikasi1 (JInternalFrame) pada btnKeluar (Jbutton) dengan cara: *right-click* btnKeluar 🡪 Events 🡪 Action 🡪 actionPerformed

private void btnKeluarActionPerformed(java.awt.event.ActionEvent evt) {

 ifrAplikasi1.dispose(); //instruksi untuk menutup ifrAplikasi1 (JInternalFrame)

 }



Gambar 4. JInternalFrame dengan button keluar.

Secara default JInternalFrame akan tampil pada desktop saat aplikasi dijalankan, untuk itu kita harus menutupnya saat aplikasi tampil (componentShown) pertama kali dengan cara seperti pada langkah berikut:

1. Buat kode program untuk membuat JInternalFrame ditutup saat ComponentShown, melalui **navigator** *right-click* Jframe 🡪 Events 🡪 Component 🡪 componentShown.

Kemudian tulis instruksi yang diinginkan pada area kode tersebut, misal seperti berikut:

private void formComponentShown(java.awt.event.ComponentEvent evt) {

 ifrAplikasi1.dispose(); //instruksi untuk menutup ifrAplikasi1 (JInternalFrame)

 }

1. Tambahkan JMenuBar dan JMenuItem ke JFrame sebagai sarana untuk membuka ifrAplikasi1 (JInternalFrame).
2. Ubah text JMenuItem (pada tab properties) sesuai kegunaan (misal: **Text: Buka Aplikasi 1**)



Gambar 5. Menu pada aplikasi

1. Ubah variabel name JMenuItem (pada tab code properties) sesuai kegunaan (misal: **variable name: mnuBukaAplikasi1**).
2. Agar **mnuBukaAplikasi1** dapat berfungsi untuk membuka ifrAplikasi1 (JInternalFrame) saat di click, tambahkan kode program pada mnuBukaAplikasi1 (JMenuItem) dengan cara: *right-click* mnuBukaAplikasi1 🡪 Events 🡪 Action 🡪 actionPerformed

private void mnuBukaAplikasi1ActionPerformed(java.awt.event.ActionEvent evt) {

 if( ! ( ifrAplikasi1.isVisible() ) ) { // Memeriksa apakah ifrAplikasi1 sedang tidak tampil

this.dpUtama.add(ifrAplikasi1); // Memasukkan ifrAplikasi1 ke desktopPane

 ifrAplikasi1.setVisible(true); // Menampilkan ifrAplikasi1

 }

 else {

 JOptionPane.showMessageDialog(this, "Aplikasi 1 sudah dibuka");

 }

}

1. Selesai

### Membuat Aplikasi dengan Menggunakan JinternalFrame dari File Class yang Terpisah dari File Class Tempat JDesktopPane

### Membuat File Class Aplikasi1 jenis JinternalFrame

Untuk membuat file class Aplikasi1 dengan jenis JInternalFrame yang nanti akan digunakan di dalam aplikasi dengan JDesktopPane, gunakan langkah-langkah berikut:

1. Pada package tempat aplikasi utama (aplikasi dengan JDesktopPane), ***right-click* 🡪 New 🡪 JInternalFrame Form**



Gambar 6. Mambuat Class jenis JInternalFrame.

1. Pada Class Name isi dengan nama class yang diinginkan (sesuaikan dengan kegunaannya), pada contoh Class Name: **LatihanFrame\_8\_Aplikasi1\_JInternalFrame**



Gambar 7. Form Pembuatan Class Jenis JInternalFrame

1. Ubah title **JInternalFrame** (pada tab properties) sesuai kegunaan (misal: **Title: Aplikasi 1**)
2. Tambahkan widget yang diperlukan sesuai kebutuhan.
3. Tambahkan JButton untuk menutup Aplikasi 1, pada contoh setting JButton:
	1. Variable name: **btnKeluar**
	2. Title: KELUAR
4. Tambahkah kode program pada btnKeluar **btnKeluar** dapat berfungsi menutup Aplikasi 1 saat di click: *right-click* btnKeluar 🡪 Events 🡪 Action 🡪 actionPerformed

private void btnKeluarActionPerformed(java.awt.event.ActionEvent evt) {

 this.dispose(); //instruksi untuk menutup Aplikasi 1

 }



Gambar 8. Tampian Class **LatihanFrame\_8\_Aplikasi1\_JInternalFrame**

1. Membuat Desktop pada aplikasi utama: Buka file class aplikasi utama (*double-click* nama class).



Gambar 9. Frame Aplikasi Utama

1. Masukkan Desktop Pane ke Frame Aplikasi Utama (click & drag Desktop Pane dari Swing Controls ke Frame Aplikasi).



Gambar 10. JDesktopPane di dalam JFrame aplikasi.

1. Ubah variabel name **JDesktopPane** (pada tab code properties) sesuai kegunaan (misal: **variable name: dpUtama**)
2. Membuat object (instant) sebagai representasi File Class eksternal pada Aplikasi Utama dengan cara:
	1. Membuat Field object (instant) dari class Aplikasi 1 (yaitu: **LatihanFrame\_8\_Aplikasi2\_JInternalFrame**).
	2. Mengaktifkan object (instant) yang sudah dibuat pada Frame Utama, dengan membuat kode program pada Frame ComponentShown

Contoh:

1. *Click* **Source** kemudian pada **navigator class** *double-click* nama class utama (yang paling atas).

Pointer (cursor) akan mengarah ke deklarasi class utama, kemudian di bawah tulisan public class ….. buat Field Object dengan kode berikut:

**Private LatihanFrame\_8\_Aplikasi2\_JInternalFrame ifrAplikasi2;**

Lihat pada gambar di bawah ini:



Gambar 11. Pembuatan field object ifrAplikasi1 sebagai representasi class eksternal

1. Pada Frame *right-click* 🡪 Event 🡪 Component 🡪 componentShown , kemudian tambahkan kode program seperti berikut:

ifrAplikasi2 = new LatihanFrame\_8\_Aplikasi2\_JInternalFrame();

Lihat Gambar di bawah ini:



Gambar 12. Mengaktifkan object (instant) ifrAplikasi1

1. Tambahkan JMenuBar dan JMenuItem ke JFrame sebagai sarana untuk membuka ifrAplikasi1 (JInternalFrame).
2. Ubah text JMenuItem (pada tab properties) sesuai kegunaan (misal: **Text: Buka Aplikasi 1**)



Gambar 5. Menu pada aplikasi

1. Ubah variabel name JMenuItem (pada tab code properties) sesuai kegunaan (misal: **variable name: mnuBukaAplikasi1**).
2. Agar **mnuBukaAplikasi1** dapat berfungsi untuk membuka ifrAplikasi1 (JInternalFrame) saat di click, tambahkan kode program pada mnuBukaAplikasi1 (JMenuItem) dengan cara: *right-click* mnuBukaAplikasi1 🡪 Events 🡪 Action 🡪 actionPerformed

private void mnuBukaAplikasi1ActionPerformed(java.awt.event.ActionEvent evt) {

// Membuat object ifrAplikasi1 dari class eksternal yang sudah kita buat:

// LatihanFrame\_8\_jInternalFrame\_1

 if( ! ( ifrAplikasi1.isVisible() ) ) { // Memeriksa apakah ifrAplikasi1 sedang tidak tampil

this.dpUtama.add(ifrAplikasi1); // Memasukkan ifrAplikasi1 ke desktopPane

 ifrAplikasi1.setVisible(true); // Menampilkan ifrAplikasi1

 }

 else {

 JOptionPane.showMessageDialog(this, "Aplikasi 1 sudah dibuka");

 }

}

1. Selesai

# Analisa Hasil Praktek

Berdasarkan praktek yang telah Anda lakukan, buat kesimpulan:

# Appendix 1: Form Data Praktek

Nama : ……………………….

NIM : ……………………….

Tgl : ……………………….

|  |  |
| --- | --- |
| **Penggalan *Source code*** | **Keluaran / *Output*** |
|  |  |
|  |  |
|  |  |