

LAPORAN
PRAKTIKUM PEMROGRAMAN VIRTUAL

MENU DAN DIALOG PADA AWT

Untuk Memenuhi Salah Satu Tugas
Mata Kuliah Praktikum Pemrograman Berorientasi Objek
Dosen Pengampu: Budianingsih, ST., MT.



Disusun Oleh:

Lindawati

3201916060

2B

PROGRAM STUDI D3 TEKNIK INFORMATIKA

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI PONTIANAK 2020

KATA PENGANTAR

Puji dan syukur saya panjatkan kepada Allah SWT atas rahmat dan karunia-nya sehingga Laporan Tugas Mandiri yang berjudul “ Menu dan Dialog pada AWT” dapat terselesaikan dengan baik. Laporan Tugas Mandiri ini merupakan salah satu tugas yang diberikan oleh dosen pengampu mata kuliah Praktikum Pemrograman visual 3 kepada mahasiswa Program Studi D3 Teknik Informatika Jurusan Teknik Elektro sebagai salah satu bagian dari komponen penilaian akademis.

Laporan Tugas Mandiri ini membahas terkait dengan AWT dan GUI sebagai salah satu bahasa Pemrograman Visual dan IDE NetBeans sebagai perangkat lunak pengembang aplikasi berorientasi objek. Demikian Laporan Tugas Mandiri ini saya buat, semoga bermanfaat.

Pontianak, 4 Oktober 2020

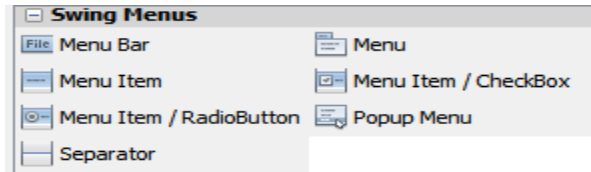
Penyusun,

(Lindawati)

Penjelasan sekilas tentang Menu dan Dialog pada AWT

[Menjelaskan cara membuat menu pada aplikasi dengan Netbeans IDE]

Untuk membuat menu pada aplikasi berbasis java biasanya digunakan widget-widget pada swing menu sebagai berikut:



Gambar 1. Widget swing menu

Pembuatan menu pada aplikasi bisa dibuat dengan melakukan kombinasi dari widget-widget swing menu seperti dapat dilihat pada gambar 1.

Terdapat 2 jenis menu yang dapat dibuat melalui Swing Menu, yaitu:

- Jenis Menu Bar
Menu Bar merupakan object menu yang melekat pada window aplikasi biasanya berada di bagian atas pada window aplikasi, pada Menu Bar biasanya ditambahkan Menu atau Menu Item sebagai isi atau sub-menu dari Menu bar.
- Jenis Popup Menu
Popup Menu merupakan object menu yang secara dinamic popup (tampil) pada posisi tertentu di dalam window aplikasi, pada Popup Menu biasanya ditambahkan Menu atau Menu Item sebagai isi atau sub-menu dari Popup Menu.

Membuat menu dengan Menu Bar

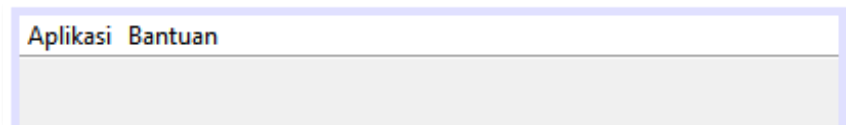
Untuk membuat menu dengan Menu Bar:

1. Masukkan Menu Bar ke Frame, (saran: letakkan di sisi atas Frame)



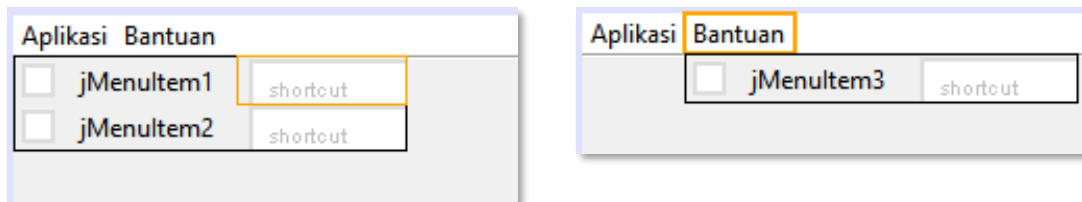
Gambar 2. Menu bar pada frame

2. Ubah variabel name (pada tab code properties) sesuai kegunaan (misal: variable name: mnuBar)
3. Ubah Text tiap-tiap JMenu (*right-click* pada tulisan di menu bar → pilih Edit Text) sesuai kegunaan. Misal perubahannya seperti berikut:



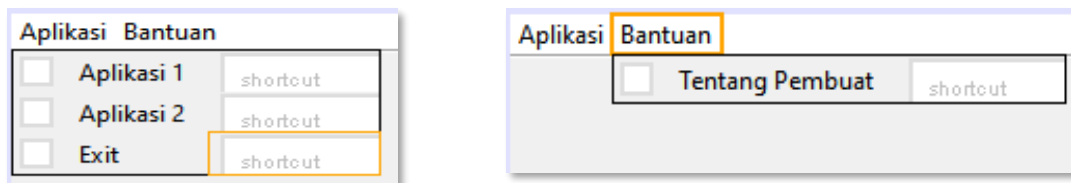
Gambar 3. Menu bar setelah di *edit text*

4. Ubah variable name untuk tiap-tiap JMenu sesuai teks yang tampil pada Menu Bar (untuk kemudahan mengingat saat membuat kode program), misal: **mnuAplikasi** untuk menu Aplikasi, dan **mnuBantuan** untuk menu Bantuan.
5. Tambahkan Menu Item ke JMenu untuk tiap-tiap JMenu sesuai kebutuhan aplikasi, misalkan seperti berikut:



Gambar 4. Menu bar beserta menu item

6. Ubah Text tiap-tiap JMenuItem (*right-click* pada tulisan di menu item → pilih Edit Text) sesuai kegunaan. Misal perubahannya seperti berikut:



Gambar 5. Menu item setelah di *edit text*

7. Ubah variable name untuk tiap-tiap JMenuItem sesuai teks yang tampil pada Menu Item (untuk kemudahan mengingat saat membuat kode program), misal: **mnuItemAplikasi1** untuk menu item Aplikasi 1, **mnuItemAplikasi2** untuk menu item Aplikasi 2 dan seterusnya.
8. Tambahkan shortcut jika diperlukan.
Untuk shortcut terdapat dua jenis:
 - a. **Accelerator**: Shortcut yang bisa langsung tanpa membuka menu (biasanya dengan kombinasi fungsi Ctrl).
Untuk membuatnya: click **menu item** yang ingin dibuatkan shortcutnya, kemudian pada **tab properties** → Accelerator: **Ctrl+CharYangDiinginkan**
Misal: Accelerator: Ctrl+x , untuk menu item Exit
 - b. **Mnemonic**: Shortcut yang berfungsi setelah menu dibuka (biasanya dengan langsung menekan tombol sesuai karakter yang di-underline)

Untuk membuatnya: click menu item yang ingin dibuatkan shortcutnya, kemudian pada tab properties → Mnemonic: *CharYangDiinginkan*
 Misal: Mnemonic: x , untuk menu item Exit

Misal seperti berikut:



Gambar 6. Design dan tampilan aplikasi Menu item dengan shortcut

9. Untuk memberikan instruksi pada menu item:

Click menu Item → *Right-click menu item* → Events → Action → actionPerformed

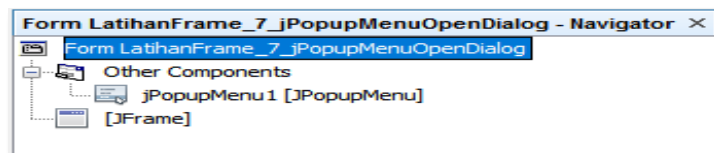
kemudian tulis instruksi yang diinginkan pada area kode tersebut, misal kodenya seperti berikut:

```
private void mnultemExitActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0); //instruksi untuk exit window aplikasi  
}
```

Membuat menu dengan Popup Menu

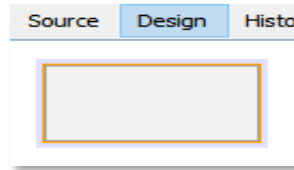
Untuk membuat menu dengan Popup Menu:

1. Masukkan Popup Menu ke Frame (jPopupMenu tidak akan terlihat)
2. Untuk mengakses dan melakukan setting terhadap JpopupMenu gunakan **navigator** class (karena JPopupMenu tidak terlihat pada JFrame) dan bahkan di navigator terlihat bahwa JPopupMenu masuk ke **other components** (tidak masuk ke JFrame).



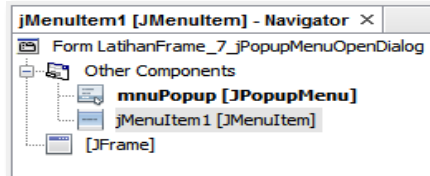
Gambar 7. JPopupMenu di dalam other components pada navigator

3. Untuk melihat PopupMenu, *double-click* pada jPopupMenu1 sehingga tulisan jPopupMenu1 menjadi bold.
4. Ubah variabel name (pada tab code properties) sesuai kegunaan (misal: **variable name: mnuPopup**)
5. Ubah ukuran Popup Menu (**preferred size** dan **Minimum Size**) *sehingga* layak dilihat (karena default-nya [0,0] sangat kecil), misal **preferred size: [100, 50]** dan **Minimum Size: [100, 50]**



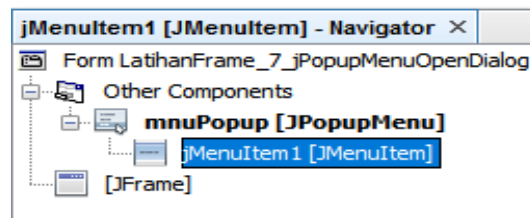
Gambar 8. Popup Menu terlihat dengan ukuran [100, 50]

6. Masukkan JMenuItem ke Popup Menu (JMenuItem tidak akan terlihat), untuk itu gunakan **Navigator class**.



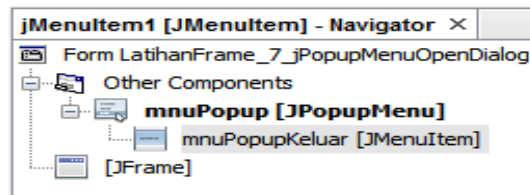
Gambar 9. jMenuItem pada Other Components

7. Pada Navigator, *click & drag* jMenuItem1 ke mnuPopup [JPopupMenu] sehingga jMenuItem sekarang berada di dalam mnuPopup [JPopupMenu] .



Gambar 10. jMenuItem berada dalam mnuPopup [JPopupMenu]

8. *Click* pada jMenuItem1 kemudian pada tab *code* ubah variable name sesuai fungsi dari menu item, misal; **Variable Name: mnuPopupKeluar**.



Gambar 11. JMenuItem dengan nama **mnuPopupKeluar** di dalam mnuPopup [JPopupMenu]

9. Pada mnuPopupKeluar Ubah Teksnya (pada tab properties→Text) sesuaikan dengan teks yang akan tampil pada popup menu, misal: **Text: Exit**
10. Ulangi langkah 6 – 9 untuk menambahkan menu item lainnya.
11. Untuk memberikan instruksi pada menu item (misal pada **mnuPopupKeluar**):

Click mnuPopupKeluar (pada navigator class) → *Right-click* mnuPopupKeluar → Events → Action → actionPerformed

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

```
private void mnuPopupKeluarActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0); //instruksi untuk exit window aplikasi  
}
```

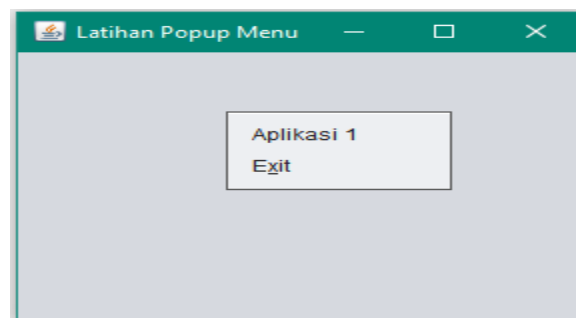
12. Untuk memberikan kode agar Popup Menu bisa tampil saat pengguna *right-click* pada frame *Right-click* pada JFrame → Events → Mouse → MouseClicked

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

```
private void formMouseClicked(java.awt.event.MouseEvent evt) {  
    if(evt.getButton()==MouseEvent.BUTTON3) {  
        mnuPopup.show(this, evt.getX(), evt.getY());  
    }  
}
```

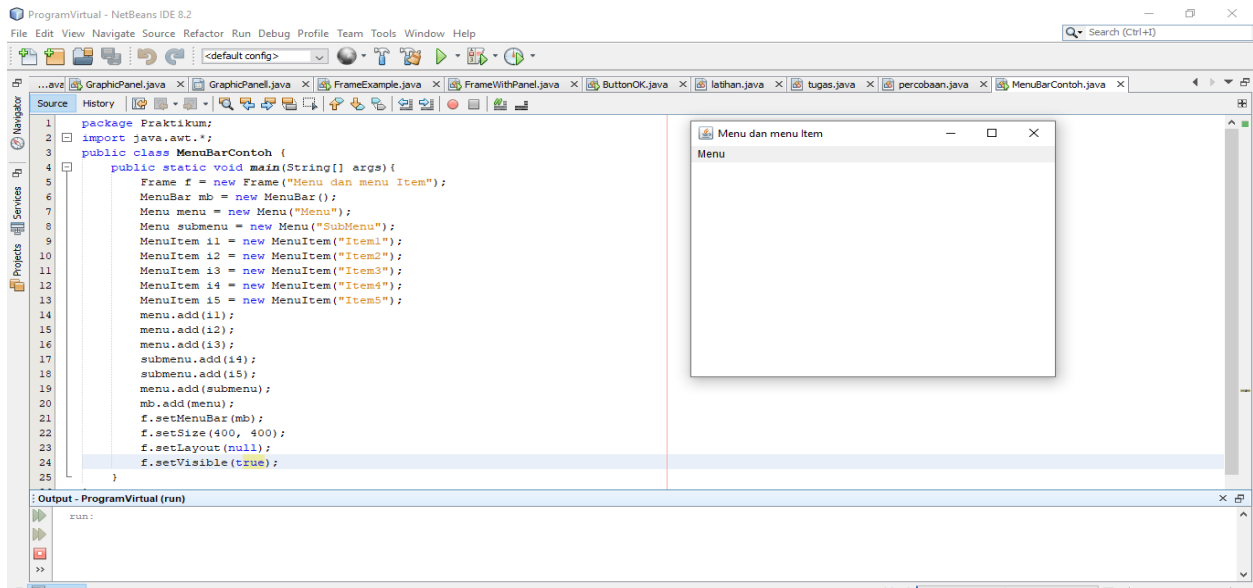
memeriksa apakah right-click

menampilkan popup menu
dengan koordinat sesuai dengan
posisi pointer



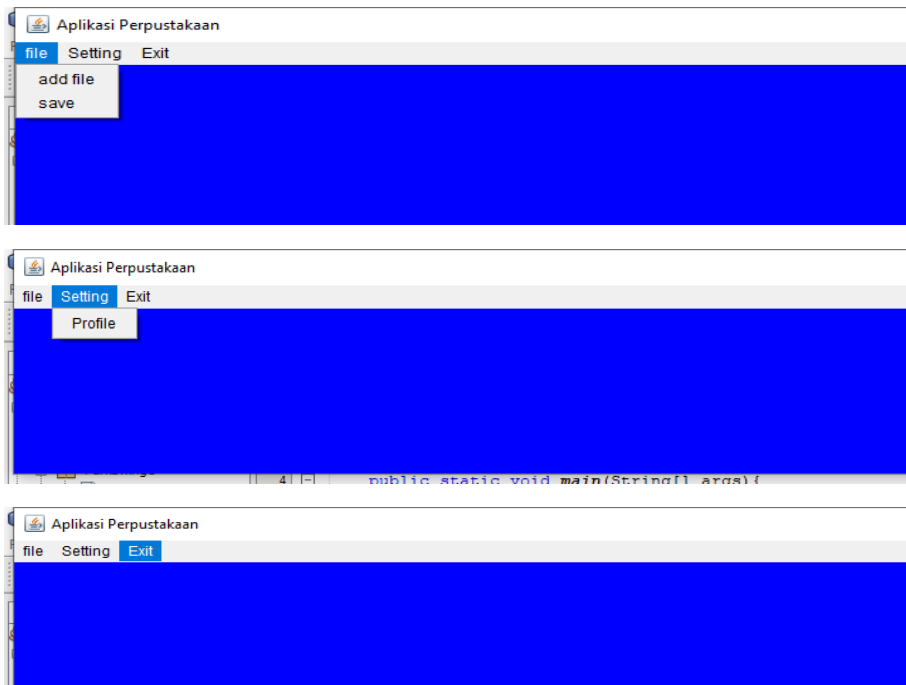
Gambar 12. Popup Menu tampil saat di *right-click* di dalam Frame

Percobaan 1



Percobaan 2





Percobaan 3

```

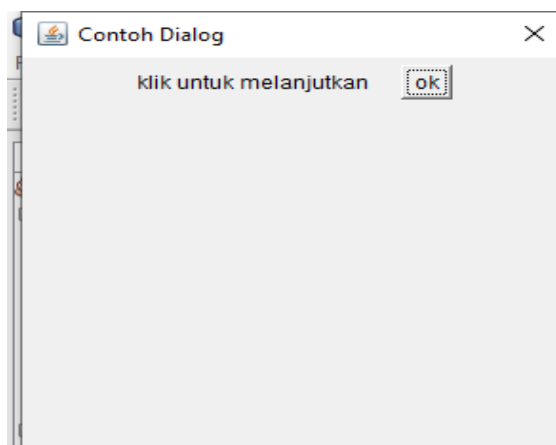
1  package Praktikum;
2  import java.awt.*;
3  import java.awt.event.ActionEvent;
4  import java.awt.event.ActionListener;
5
6  public class DialogContoh {
7      private static Dialog d;
8      DialogContoh(){
9          Frame f = new Frame();
10         d= new Dialog(f,"Contoh Dialog", true);
11         d.setLayout(new FlowLayout());
12         Button b = new Button("ok");
13         b.addActionListener(new ActionListener(){
14             public void actionaformed(ActionEvent e){
15                 d.setVisible(false);
16             }
17
18             @Override
19             public void actionPerformed(ActionEvent e) {
20                 throw new UnsupportedOperationException("Not supported yet.");

```

```

21 |         }
22 |     });
23 |
24 |     d.add(new Label("klik untuk melanjutkan"));
25 |     d.add(b);
26 |     d.setSize(300,300);
27 |     d.setVisible(true);
28 | }
29 | public static void main(String[] args) {
30 |     new DialogContoh();
31 | }

```



Latihan

1. Perbandingan antara percobaan 1 dan percobaan 2, pada percobaan 1 menggunakan Submenu merupakan bagian dari Menu yang dapat di pecah menjadi anakan sehingga di sebut Submenu yang berfungsi untuk membuat menu baru.
sedangkan pada percobaan 2 tidak menggunakan Submenu namun terdapat title, dan label dapat di deskripsikan menjadi sebuah kolom atau area yang digunakan untuk menampilkan tulisan atau gambar, sedangkan SetBackground untuk mengatur warna latar dengan memanggil getColor.
2. Fungsi skrip langkah ke 4, adapun fungsi dari import java.awt.*; yang digunakan untuk proses inputan dalam program java, sedangkan import java.awt.event.ActionEvent; dan import java.awt.event.EventListener sebuah fungsi import yang digunakan untuk proses deklarasi dari class event yang berisi komponen dari sebuah program yang dibuat, untuk mengambil semua file dari package event yang mana package event berada pada package java.awt.