Gui Programming - III

Teguh Sutanto, M.Kom.

teguh@stikom.edu

teguh.Sutanto@gmail.com

Tujuan

- Mahasiswa dapat membuat Menu dan ToolBar
- Mahasiswa dapat membuat dialog konfirmasi

Menus

- *Menus* make selection easier and are widely used in window applications.
- Java provides five classes that implement menus: **JMenuBar**, **JMenu**, **JMenuItem, JCheckBoxMenuItem**, and **JRadioButtonMenuItem**.
- **JMenuBar** is a top-level menu component used to hold the menus.
- A menu consists of *menu items* that the user can select (or toggle on or off).
- A menu item can be an instance of **JMenuItem**, **JCheckBoxMenuItem**, or **JRadioButtonMenuItem**.
- Menu items can be associated with icons, keyboard mnemonics, and keyboard accelerators.
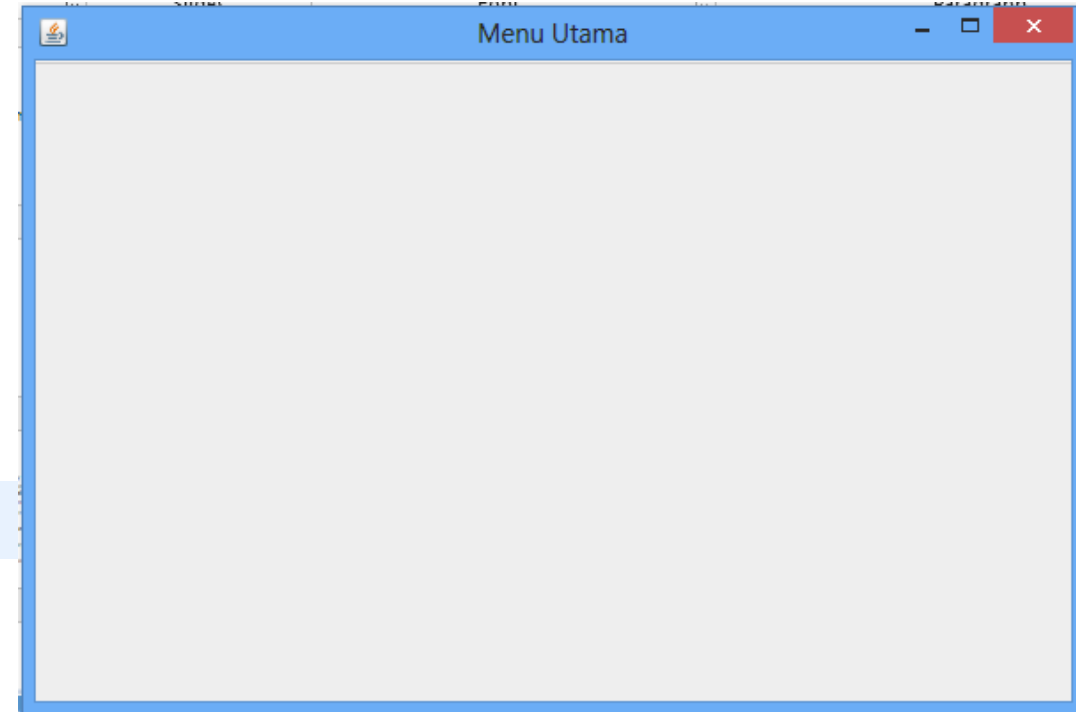- Menu items can be separated using separators.

3 Langkah Membuat Menu

Membuat Menu Bar

```java
18    JMenuBar menuBar;
19
20    public MyMenu(String judul){
21        super(judul);
22        setSize(LEBAR, TINGGI);
23        setLocation(POS_X, POS_Y);
24        setDefaultCloseOperation(EXIT_ON_CLOSE);
25
26        menuBar = new JMenuBar();
27        setJMenuBar(menuBar);
28    }
```
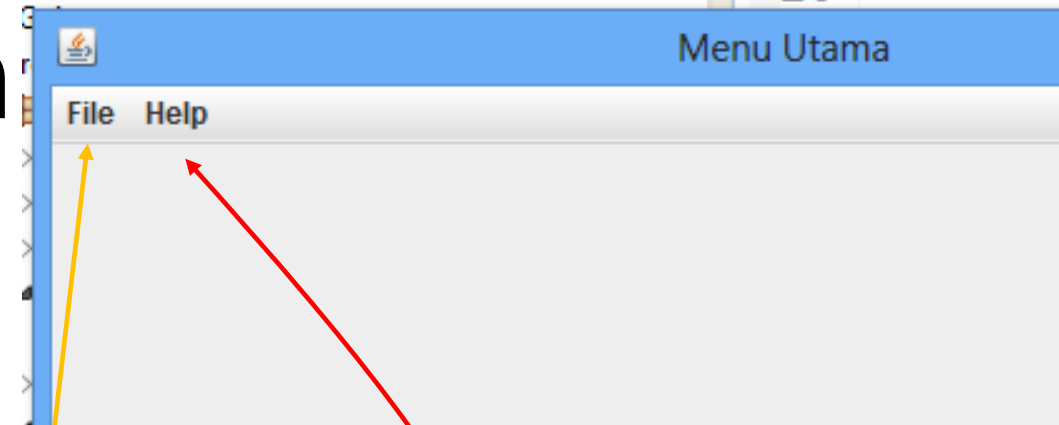
Membuat Menu (JMenu)

Menu Utama

File   Help

```java
menuBar = new JMenuBar();
setJMenuBar(menuBar);


JMenu menuFile=new JMenu("File");
JMenu menuHelp=new JMenu("Help");


menuBar.add(menuFile);
menuBar.add(menuHelp);
```
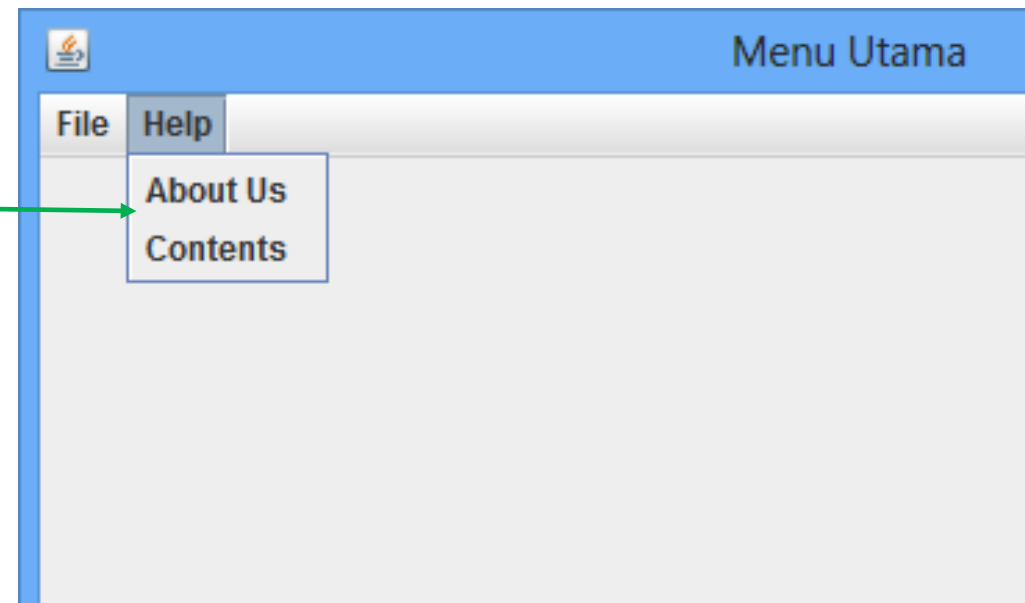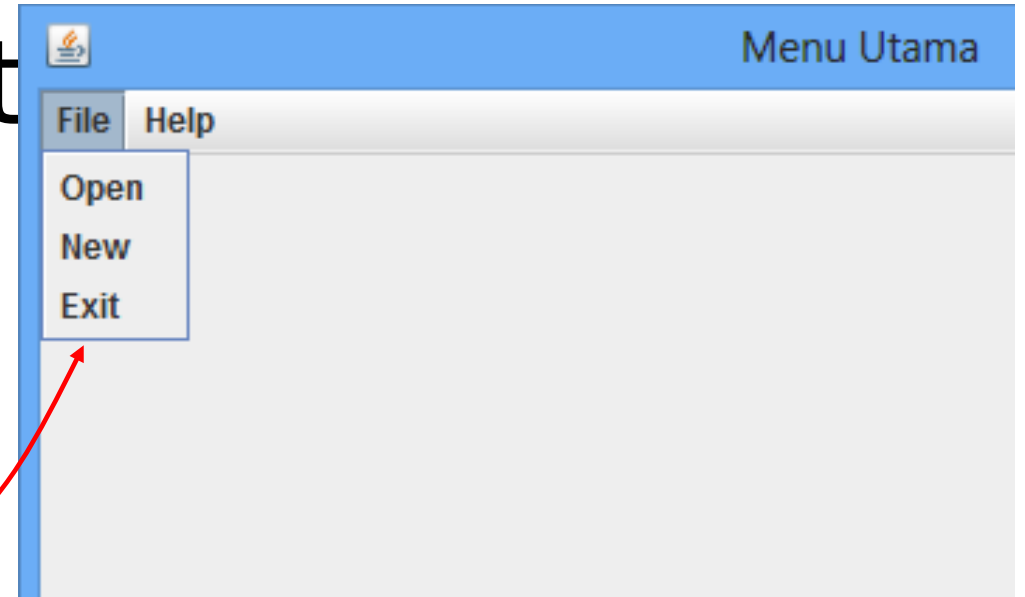
Menambahkan MenuItem

```
33
34      menuBar.add(menuFile);
35      menuBar.add(menuHelp);
36
37      JMenuItem miOpen=new JMenuItem("Open");
38      JMenuItem miNew = new JMenuItem("New");
39      JMenuItem miExit=new JMenuItem("Exit");
40      menuFile.add(miOpen);
41      menuFile.add(miNew);
42      menuFile.add(miExit);
43
44      JMenuItem miAbout=new JMenuItem("About Us");
45      JMenuItem miContent=new JMenuItem("Contents");
46      menuHelp.add(miAbout);
47      menuHelp.add(miContent);
```

Menambahkan Icon

```
JMenuItem miAbout=new JMenuItem("About Us");
JMenuItem miContent=new JMenuItem("Contents");
menuHelp.add(miAbout);
menuHelp.add(miContent);


ImageIcon iconOpen = new ImageIcon("./gambar/folder.png");
miOpen.setIcon(iconOpen);


ImageIcon iconNew = new ImageIcon("./gambar/folder_add.png");
miNew.setIcon(iconNew);


ImageIcon iconStop = new ImageIcon("./gambar/stop.png");
miExit.setIcon(iconStop);
```
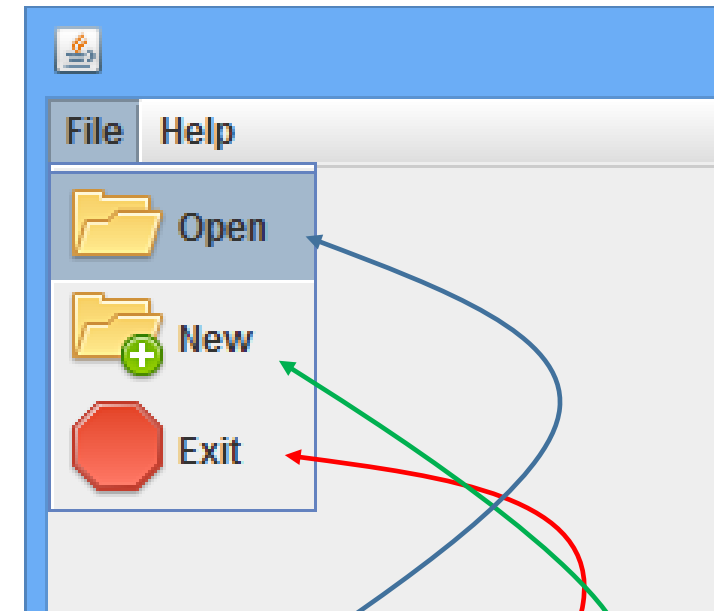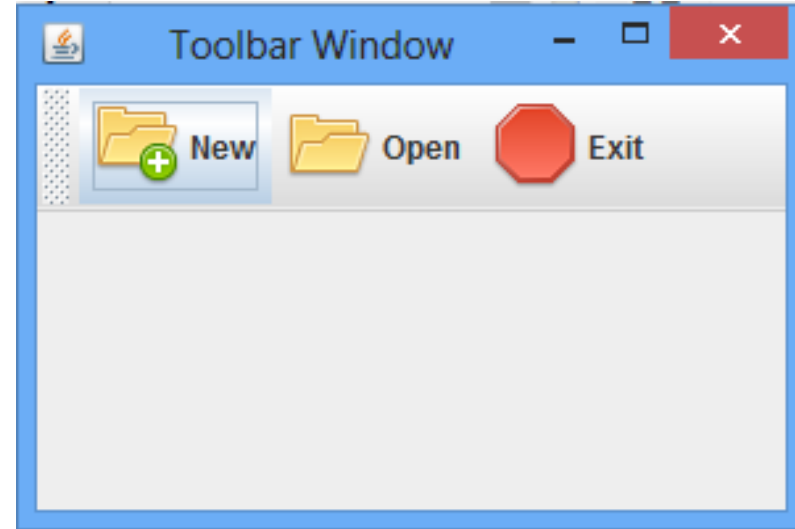
Membuat ToolBar

JButton btnOpen=**new JButton("Open");**

JButton btnNew=**new JButton("New");**

JButton btnExit=**new JButton("Exit");**

ImageIcon iconOpen = **new ImageIcon("./gambar/folder.png");**

btnOpen.setIcon(iconOpen);

ImageIcon iconNew = **new ImageIcon("./gambar/folder_add.png");**

btnNew.setIcon(iconNew);

ImageIcon iconStop = **new ImageIcon("./gambar/stop.png");**

btnExit.setIcon(iconStop);


```
JToolBar toolBar=new JToolBar();
toolBar.add(btnNew);
toolBar.add(btnOpen);
toolBar.add(btnExit);
getContentPane().setLayout(new
BorderLayout());
getContentPane().add(toolBar,BorderLayout
.NORTH);
```

Membuat Dialog (JOptionPane)

- **Message dialog** shows a message and waits for the user to click OK.
- **Confirmation dialog** shows a question and asks for confirmation, such as OK or Cancel.
- **Input dialog** shows a question and gets the user's input from a text field, combo box, or list.
- **Option dialog** shows a question and gets the user's answer from a set of options.

Membut Konfirmasi Dialog

- **public static int** showConfirmDialog(Component parentComponent,
- **public static int** showConfirmDialog(Component parentComponent, Object message, String title, **int** optionType)
- **public static int** showConfirmDialog(Component parentComponent, Object message, String title, **int** optionType, **int** messageType)
- **public static int** showConfirmDialog(Component parentComponent, Object message, String title, **int** optionType, **int** messageType, Icon icon)