

Lko Kurniawan Khannedy



Java™

SwingMakeOver

Extreme Java GUI Programming

WARNING

**BUKU INI HANYA UNTUK PEMBACA YANG TELAH MENGENAL
STANDAR PEMBUATAN APLIKASI MENGGUNAKAN SWING.**

Daftar Isi

| | |
|---|----|
| Daftar Isi..... | 3 |
| Daftar Gambar..... | 7 |
| Persiapan..... | 13 |
| Swing dan AWT..... | 13 |
| Project SwingMakeOver..... | 13 |
| Project SwingMakeOver dan NetBeans | 13 |
| Project SwingMakeOver dan Eclipse..... | 15 |
| JPanel | 19 |
| Pemanasan | 19 |
| Memanipulasi paintComponent() | 19 |
| Menggambar Bentuk | 19 |
| Mewarnai Bentuk dengan Solid Color..... | 29 |
| Mewarnai Bentuk dengan Gradient Color | 37 |
| Antialiasing..... | 44 |
| Alpha..... | 48 |
| Color | 48 |
| Composite | 50 |
| Image | 51 |
| Efek Kaca | 54 |
| Kesimpulan..... | 58 |
| Pelajaran 4 JButton | 63 |
| Glass..... | 63 |
| Tombol Kotak Lancip..... | 63 |
| Tombol Kotak Luncul..... | 66 |
| Image | 69 |
| Mouse | 72 |
| Over..... | 72 |
| Press | 75 |
| Kesimpulan..... | 80 |
| Pelajaran 5 JLabel | 81 |
| Drop Shadow Text | 81 |

| | |
|------------------------------|-----|
| Clip Text..... | 83 |
| Gradient Text..... | 84 |
| Pelajaran 6 JTextField | 86 |
| Shadow Text..... | 86 |
| Auto Complete Text..... | 89 |
| Manipulasi Background..... | 92 |
| Gradients | 92 |
| Image..... | 94 |
| Validasi Input..... | 96 |
| Kesimpulan..... | 98 |
| Pelajaran 7 JComboBox..... | 103 |
| Render..... | 103 |
| Background..... | 103 |
| Image..... | 106 |
| Kesimpulan..... | 109 |
| Pelajaran 8 JList | 115 |
| Render..... | 115 |
| Background..... | 115 |
| Image..... | 117 |
| Boolean | 118 |
| Number | 120 |
| Font | 122 |
| Model..... | 124 |
| Sorting | 124 |
| Filter | 128 |
| Kesimpulan..... | 131 |
| Pelajaran 9 JViewport | 133 |
| JTextArea | 135 |
| Background Image | 135 |
| Glass | 139 |
| Pelajaran 10 JWindow..... | 142 |
| About Screen | 142 |
| Splash Screen..... | 144 |
| Kesimpulan..... | 147 |

| | |
|--|-----|
| Pelajaran 11 JTree..... | 149 |
| Render..... | 149 |
| File and Folder | 149 |
| About Icon | 151 |
| Leaf Icon | 151 |
| Color..... | 153 |
| Background..... | 155 |
| Glass..... | 159 |
| Editor..... | 162 |
| JComboBox | 163 |
| JChekBox | 164 |
| Color..... | 166 |
| File..... | 169 |
| Yes No Editor | 172 |
| Kesimpulan..... | 178 |
| Pelajaran 12 JTable | 179 |
| Render..... | 179 |
| Color..... | 179 |
| Gradient | 181 |
| NetBeans | 183 |
| Editor..... | 200 |
| JComboBox | 200 |
| JChekBox | 202 |
| Color..... | 210 |
| Spinner | 213 |
| Custom Editor | 216 |
| Other..... | 220 |
| Sorter..... | 220 |
| Filter | 224 |
| Microsoft Excel dan OpenOffice Calc..... | 227 |
| HTML..... | 241 |
| Kesimpulan..... | 245 |
| Pelajaran 13 JTextPane | 246 |
| Highlight..... | 246 |

| | |
|--------------------------------|-----|
| Pelajaran 14 GlassPane | 249 |
| Gradient Glass | 249 |
| Block Event | 251 |
| Warning GlassPane | 255 |
| GlassPane Transition 1..... | 257 |
| GlassPane Transition 2..... | 260 |
| GlassPane dan JTable | 263 |
| Dragging | 265 |
| File..... | 266 |
| Image..... | 270 |
| Container..... | 275 |
| Show Message | 275 |
| GlassDialog | 279 |
| ProgressBar..... | 282 |
| ProgressBar2..... | 285 |
| Kesimpulan..... | 287 |
| Pelajaran 15 JLayeredPane..... | 288 |
| GlassPane | 293 |
| Kesimpulan..... | 296 |
| Sample Swing Make Over | 297 |
| Lampiran | 301 |
| Instalasi | 301 |
| 7-Zip | 301 |
| JDK 1.6 | 303 |
| Java Doc..... | 307 |
| NetBeans 6.0 | 308 |
| Eclipse 3.3..... | 312 |
| Tentang Penulis | 315 |

Daftar Gambar

| | |
|---------------------------------------|------------------------------|
| Gambar 1 Menu Libraries..... | Error! Bookmark not defined. |
| Gambar 2 Library Manager | Error! Bookmark not defined. |
| Gambar 3 New Library | Error! Bookmark not defined. |
| Gambar 4 Add Jar / Folder | Error! Bookmark not defined. |
| Gambar 5 Browse Jar / Folder | Error! Bookmark not defined. |
| Gambar 6 Library Manager | Error! Bookmark not defined. |
| Gambar 7 Project Properties..... | Error! Bookmark not defined. |
| Gambar 8 Project Properties..... | Error! Bookmark not defined. |
| Gambar 9 Add Library | Error! Bookmark not defined. |
| Gambar 10 Project Properties..... | Error! Bookmark not defined. |
| Gambar 11 Project Explorer | Error! Bookmark not defined. |
| Gambar 12 Menu Preferences | Error! Bookmark not defined. |
| Gambar 13 Preferences | Error! Bookmark not defined. |
| Gambar 14 New User Library | Error! Bookmark not defined. |
| Gambar 15 Preferences | Error! Bookmark not defined. |
| Gambar 16 Jar Selection | Error! Bookmark not defined. |
| Gambar 17 Menu Add Libraries | Error! Bookmark not defined. |
| Gambar 18 Add Library | Error! Bookmark not defined. |
| Gambar 19 Add Library..... | Error! Bookmark not defined. |
| Gambar 20 Package Explorer | Error! Bookmark not defined. |
| Gambar 21 Menu Open Project..... | 14 |
| Gambar 22 Open Project..... | 14 |
| Gambar 23 Project Explorer | 15 |
| Gambar 24 Menu Import | 15 |
| Gambar 25 Import | 16 |
| Gambar 26 Import | 17 |
| Gambar 27 Browse For Folder..... | 17 |
| Gambar 28 Import | 18 |
| Gambar 29 Package Explorer | 18 |
| Gambar 30 PanelKotak.java | 21 |
| Gambar 31 PanelKotak2.java | 22 |
| Gambar 32 PanelRound.java | 23 |
| Gambar 33 PanelRound2.java | 24 |
| Gambar 34 PanellLingkaran .java | 26 |
| Gambar 35 PanellLingkaran2.java | 27 |
| Gambar 36 PanelBentukLain.java..... | 28 |
| Gambar 37 PanelKotak.java | 30 |
| Gambar 38 PanelKotak2.java | 31 |
| Gambar 39 PanelRound.java | 32 |
| Gambar 40 PanelRound2.java | 33 |
| Gambar 41 PanelLingkaran.java | 35 |
| Gambar 42 PanellLingkaran2.java | 36 |

| | |
|--|-----|
| Gambar 43 PanelBentukLain.java | 37 |
| Gambar 44 PanelLinear.java | 39 |
| Gambar 45 PanelRadial.java..... | 41 |
| Gambar 46 PanelLinearMultiple.java | 42 |
| Gambar 47 PanelRadialMultiple.java | 44 |
| Gambar 48 PanelLingkaran sebelum antialiasing..... | 44 |
| Gambar 49 PanelLingkaranAntialiasing.java | 46 |
| Gambar 50 PanelLinear sebelum antialiasing | 46 |
| Gambar 51 PanelLinearMultipleAntialiasing.java | 48 |
| Gambar 52 PanelAlpha.java | 49 |
| Gambar 53 PanelComposite.java | 51 |
| Gambar 54 PanellImage.java | 52 |
| Gambar 55 PanellImage2.java..... | 54 |
| Gambar 56 PanellImageKaca.java | 56 |
| Gambar 57 PanellImageKaca2.java | 58 |
| Gambar 58 MyOpenOffice Login | 59 |
| Gambar 59 MyOpenOffice Menu | 59 |
| Gambar 60 MyOpenOffice Contact | 60 |
| Gambar 61 PanelEfect.java | 62 |
| Gambar 62 ButtonGlass.java..... | 64 |
| Gambar 63 ButtonGlass2.java | 66 |
| Gambar 64 ButtonGlassRound.java..... | 67 |
| Gambar 65 ButtonGlassRound2.java | 69 |
| Gambar 66 ButtonImage.java | 71 |
| Gambar 67 ButtonImage menggunakan glass diatas | 72 |
| Gambar 68 ButtonOver.java saat mouse exit | 75 |
| Gambar 69 ButtonOver.java saat mouse over | 75 |
| Gambar 70 ButtonOverPress.java saat mouse exit | 79 |
| Gambar 71 ButtonOverPress.java saat mouse over | 79 |
| Gambar 72 ButtonOverPress.java saat mouse press..... | 79 |
| Gambar 73 LabelDropShadow.java | 82 |
| Gambar 74 LabelClip.java..... | 84 |
| Gambar 75 LabelGradient.java..... | 85 |
| Gambar 76 Contoh Shadow Text..... | 86 |
| Gambar 77 TextFieldShadowText.java saat shadow text terlihat | 89 |
| Gambar 78 TextFieldShadowText.java saat user menginputkan teks | 89 |
| Gambar 79 TextFieldAutoComplete.java | 91 |
| Gambar 80 TextFieldAutoComplete.java saat user menginputkan teks | 91 |
| Gambar 81 TextFieldAutoComplete untuk website | 92 |
| Gambar 82 TextFieldBackground | 94 |
| Gambar 83 TextFieldImage.java | 96 |
| Gambar 84 TextFieldImage.java saat user menginputkan teks | 96 |
| Gambar 85 DocumentFormatedApp.java | 98 |
| Gambar 86 TextStyle.java | 100 |
| Gambar 87 TextStyle2.java | 102 |

| | |
|--|-----|
| Gambar 88 ComboColor.java | 105 |
| Gambar 89 ComboColor2.java | 106 |
| Gambar 90 Combolmage.java | 109 |
| Gambar 91 ComboGradient.java | 112 |
| Gambar 92 ComboGradient2.java | 114 |
| Gambar 93 ListColor.java | 116 |
| Gambar 94 ListColor2.java | 117 |
| Gambar 95 ListBackground.java | 118 |
| Gambar 96 ListBoolean.java..... | 120 |
| Gambar 97 ListNumber.java..... | 122 |
| Gambar 98 ListFont.java | 124 |
| Gambar 99 List menggunakanListModelUsu saat keadaan No SORT | 127 |
| Gambar 100 List menggunakanListModelUsu saat keadaan ASCENDING | 127 |
| Gambar 101 List menggunakanListModelUsu saat keadaan DESCENDING | 128 |
| Gambar 102 List menggunakanListModelUsuFilter saat tidak di filter | 130 |
| Gambar 103 List menggunakanListModelUsuFilter saat difilter | 131 |
| Gambar 104 Font Browser | 132 |
| Gambar 105 JViewPort dalam JScrollPane..... | 133 |
| Gambar 106 ViewPortTranparan.java | 135 |
| Gambar 107 ViewPortImage.java | 137 |
| Gambar 108 ViewPortImage.java | 137 |
| Gambar 109 TextArealImage.java | 139 |
| Gambar 110 TextArealImage.java | 139 |
| Gambar 111 ViewPortImageGlass.java | 141 |
| Gambar 112 Contoh JWindow | 142 |
| Gambar 113 WindowAbout.java | 144 |
| Gambar 114 WindowAbout.java saat menampilkan AboutScreen | 144 |
| Gambar 115 WindowSplashScreen.java saat splashscreen | 146 |
| Gambar 116 WindowSplashScreen.java saat form muncul | 146 |
| Gambar 117 WindowTooltip.java..... | 148 |
| Gambar 118 WindowTooltip.java saat window tooltip muncul..... | 148 |
| Gambar 119 FileRender.java | 151 |
| Gambar 120 FileRenderLeaf.java..... | 153 |
| Gambar 121 TreeColor.java | 155 |
| Gambar 122 TreeBackground.java | 157 |
| Gambar 123 TreeBackground.java | 159 |
| Gambar 124 TreeGlass.java..... | 162 |
| Gambar 125 TreeEdited.java..... | 163 |
| Gambar 126 TreeEditedComboBox.java | 164 |
| Gambar 127 TreeEditedChekBox.java | 166 |
| Gambar 128 TreeColorEditor.java | 168 |
| Gambar 129 Color Editor | 169 |
| Gambar 130 TreeFileEditor.java | 172 |
| Gambar 131 File Editor | 172 |
| Gambar 132 TreeFileEditorYesNo.java | 176 |

| | |
|--|-----|
| Gambar 133 File Editor | 176 |
| Gambar 134 Yes No Editor | 177 |
| Gambar 135 TreeFileEditorYesNo saat memilih NO..... | 177 |
| Gambar 136 TreeFileEditorYesNo saat memilih YES | 178 |
| Gambar 137 TableColor.java | 181 |
| Gambar 138 TableGradient.java..... | 183 |
| Gambar 139 PanelAlamat.java | 184 |
| Gambar 140 PanelAlamat2.java | 185 |
| Gambar 141 TableAlamat.java | 187 |
| Gambar 142 TableAlamat.java | 188 |
| Gambar 143 New File..... | 190 |
| Gambar 144 New Bean Form | 191 |
| Gambar 145 New File..... | 192 |
| Gambar 146 NetBeans GUI Builder | 192 |
| Gambar 147 Choose Bean..... | 194 |
| Gambar 148 Choose Bean..... | 194 |
| Gambar 149 Tampilan GUI Builder | 194 |
| Gambar 150 Tampilan akhir render..... | 194 |
| Gambar 151 TableAlamatCool.java | 197 |
| Gambar 152 PanelViewPort.java | 198 |
| Gambar 153 TableAlamatViewPort.java | 200 |
| Gambar 154 TableEditorComboBox.java | 202 |
| Gambar 155 TableEditorChekBox.java | 203 |
| Gambar 156 Panel render nama | 204 |
| Gambar 157 Panel render jeniskelamin..... | 204 |
| Gambar 158 Panel render menikah..... | 204 |
| Gambar 159 TableEditorStylist saat mengedit nama | 209 |
| Gambar 160 TableEditorStylist saat mengedit jenis kelamin..... | 209 |
| Gambar 161 TableEditorStylist saat mengedit menikah..... | 210 |
| Gambar 162 TableColorEditor.java..... | 212 |
| Gambar 163 Color Editor | 213 |
| Gambar 164 TableNumberEditor.java | 216 |
| Gambar 165 TableNumberEditor saat mengedit umur | 216 |
| Gambar 166 Panel editor alamat..... | 217 |
| Gambar 167 TableAlamatCoolEditor.java | 219 |
| Gambar 168 Editor Alamat..... | 220 |
| Gambar 169 TableSorter.java..... | 221 |
| Gambar 170 TableSorter saat terurut secara ascending | 222 |
| Gambar 171 TableSorter saat mengurut secara descending..... | 222 |
| Gambar 172 TableSorterGood saat mengurut ascending | 224 |
| Gambar 173 TableSorterGood saat mengurut descending | 224 |
| Gambar 174 TableSorterFilter.java..... | 226 |
| Gambar 175 TableSorterFilter saat difilter | 227 |
| Gambar 176 TableCSV.java | 230 |
| Gambar 177 Save Dialog | 230 |

| | |
|--|-----|
| Gambar 178 File CSV..... | 231 |
| Gambar 179 Microsoft Offive Excel 2007 | 231 |
| Gambar 180 Microsoft Office Excel 2007 | 232 |
| Gambar 181 Get External Data..... | 232 |
| Gambar 182 Import Text File | 233 |
| Gambar 183 Text Import Wizard | 234 |
| Gambar 184 Text Import Wizard | 235 |
| Gambar 185 Text Import Wizard | 236 |
| Gambar 186 Import Data | 236 |
| Gambar 187 Microsoft Office Excel 2007 | 237 |
| Gambar 188 OpenOffice Calc..... | 238 |
| Gambar 189 Menu Open | 238 |
| Gambar 190 Open Dialog..... | 239 |
| Gambar 191 Text Import..... | 240 |
| Gambar 192 OpenOffice Calc..... | 240 |
| Gambar 193 TableHTML.java | 244 |
| Gambar 194 Save Dialog | 244 |
| Gambar 195 File HTML..... | 245 |
| Gambar 196 Mozilla Firefox | 245 |
| Gambar 197 TextPaneHightLight.java | 248 |
| Gambar 198 TextPaneHightLight saat hightlight..... | 248 |
| Gambar 199 Posisi GlassPane..... | 249 |
| Gambar 200 GlassPaneGradientApp.java | 250 |
| Gambar 201 GlassPaneGradient2App.java | 251 |
| Gambar 202 Form yang menggunakan GlassPaneBlock..... | 253 |
| Gambar 203 Saat form diblock oleh GlassPaneBlock | 253 |
| Gambar 204 GlassPaneBlock2App.java | 255 |
| Gambar 205 GlassPaneBlock2App saat diblock | 255 |
| Gambar 206 Form yang menggunakan GlassPaneWarning..... | 257 |
| Gambar 207 Form yang menggunakan GlassPaneWarning..... | 257 |
| Gambar 208 Form yang menggunakan GlassPaneTransition | 259 |
| Gambar 209 Saat GlassPaneTransition memulai transisi..... | 260 |
| Gambar 210 Form yang menggunakan GlassPaneTransitionRotasi | 262 |
| Gambar 211 Saat GlassPaneTransitionRotasi memulai transisi..... | 263 |
| Gambar 212 TableTransition.java..... | 265 |
| Gambar 213 TableTransition saat memulai transisi | 265 |
| Gambar 214 GlassPaneFileDragApp.java | 269 |
| Gambar 215 GlassPaneFileDragApp saat glasspane menampilkan gambar saat file didrag | 269 |
| Gambar 216 GlassPaneFileDragApp saat file drag didrop | 270 |
| Gambar 217 GlassPanelImageDragApp.java | 274 |
| Gambar 218 GlassPanelImageDragApp saat user mendrag file gambar | 275 |
| Gambar 219 GlassPanelImageDragApp saat user mendrag file selain gambar | 275 |
| Gambar 220 GlassPaneMessageApp.java | 278 |
| Gambar 221 GlassPaneMessageApp saat menampilkan pesan..... | 279 |
| Gambar 222 GlassPanePanelApp.java | 281 |

| | |
|---|-----|
| Gambar 223 GlassPanePanelApp saat menampilkan panel..... | 281 |
| Gambar 224 GlassPanePanelApp.java | 282 |
| Gambar 225 GlassPaneProgressApp.java | 284 |
| Gambar 226 GlassPaneProgressApp saat proses progress berjalan | 285 |
| Gambar 227 GlassPaneProgress2App.java | 287 |
| Gambar 228 GlassPaneProgress2App saat progress berjalan..... | 287 |
| Gambar 229 Posisi JLayeredPane dalam JFrame..... | 288 |
| Gambar 230 SampleLayer.java | 290 |
| Gambar 231 LayerUpDown saat panel merah diatas | 292 |
| Gambar 232 LayerUpDown saat panel biru diatas..... | 292 |
| Gambar 233 LayerGlassPane.java | 295 |
| Gambar 234 LayerGlassPane saat menampilkan pesan | 295 |
| Gambar 235 LayerGlassPane saat menampilkan progress | 296 |
| Gambar 236 FormLogin.java | 297 |
| Gambar 237 FormIdentitas.java | 298 |
| Gambar 238 NotePad.java | 299 |
| Gambar 239 FontBrowser.java..... | 300 |
| Gambar 240 TableMahasiswa.java..... | 300 |
| Gambar 241 Choose Install Location | 301 |
| Gambar 242 Installing..... | 302 |
| Gambar 243 Complete Setup | 302 |
| Gambar 244 License Agreement | 303 |
| Gambar 245 Custom Setup | 304 |
| Gambar 246 Installing..... | 304 |
| Gambar 247 Custom Setup | 305 |
| Gambar 248 Installing Java | 306 |
| Gambar 249 Wizard Completed | 306 |
| Gambar 250 Ekstrak JavaDoc | 307 |
| Gambar 251 Extrakt File | 307 |
| Gambar 252 Ekstrak File | 308 |
| Gambar 253 Proses Ekstrak..... | 308 |
| Gambar 254 NetBeans Installer..... | 309 |
| Gambar 255 License Agreement | 310 |
| Gambar 256 NetBeans IDE Installation..... | 311 |
| Gambar 257 Summary | 312 |
| Gambar 258 Ekstrak Eclipse Java..... | 312 |
| Gambar 259 Ekstrak File | 313 |
| Gambar 260 Proses EKstrak | 313 |
| Gambar 261 File Eclipse.EXE | 314 |

Persiapan

Sebelum saya mulai menjelaskan, menceramahi ataupun mengajarkan hal-hal yang saya ketahui yang mungkin belum anda ketahui ataupun yang sudah anda ketahui saya akan memberitahu dulu tentang dasar Extreem Swing.

Swing dan AWT

Swing sebenarnya adalah sebuah pengembangan dari library GUI AWT (Abstract Window Toolkit), oleh karena itu kita bisa menemukan metode-metode yang ada di AWT pada Swing karena memang Swing keturunan AWT.

Dalam AWT mungkin kita sudah biasa dengan yang mengoveride metode paint(), namun dalam Swing kita tidak dianjurkan untuk mengoveride metode ini, kita dianjurkan untuk mengoveride metode paintComponent().

Jadi pelajaran bab ini adalah “ingat jangan mengoveride metode paint(), tapi overide metode paintComponent()”

Misal jika kita akan mengoveride metode paintComponent() sebuah JPanel :

```
import java.awt.Graphics;
import javax.swing.JPanel;

/**
 *
 * @author usu
 */
public class SamplePanel extends JPanel {

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
    }
}
```

Project SwingMakeOver

Project SwingMakeOver merupakan file source code untuk buku ini. Ada 2 versi, yaitu untuk NetBeans dan Eclipse.

Project SwingMakeOver dan NetBeans

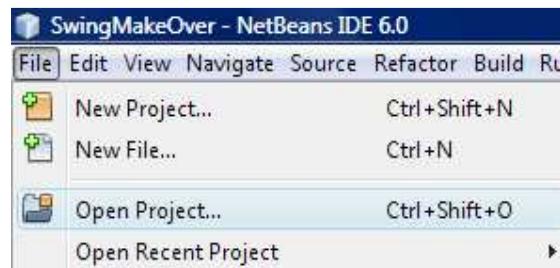
Jika anda menggunakan Eclipse, silahkan loncat ke subbab selanjutnya.

Mungkin anda bingung kenapa saya selalu mendahulukan NetBeans daripada Eclipse, bukan saya tak menyukai Eclipse, tapi karena hampir 90% pembuatan source code ini saya buat dalam NetBeans

dan selain itu NetBeans juga satu-satunya IDE untuk Swing yang bisa dibilang terbaik dibandingkan IDE yang lainnya, tapi kalo soal Code Editor, ya tak dapat dipungkiri lagi Eclipse masih memimpin.

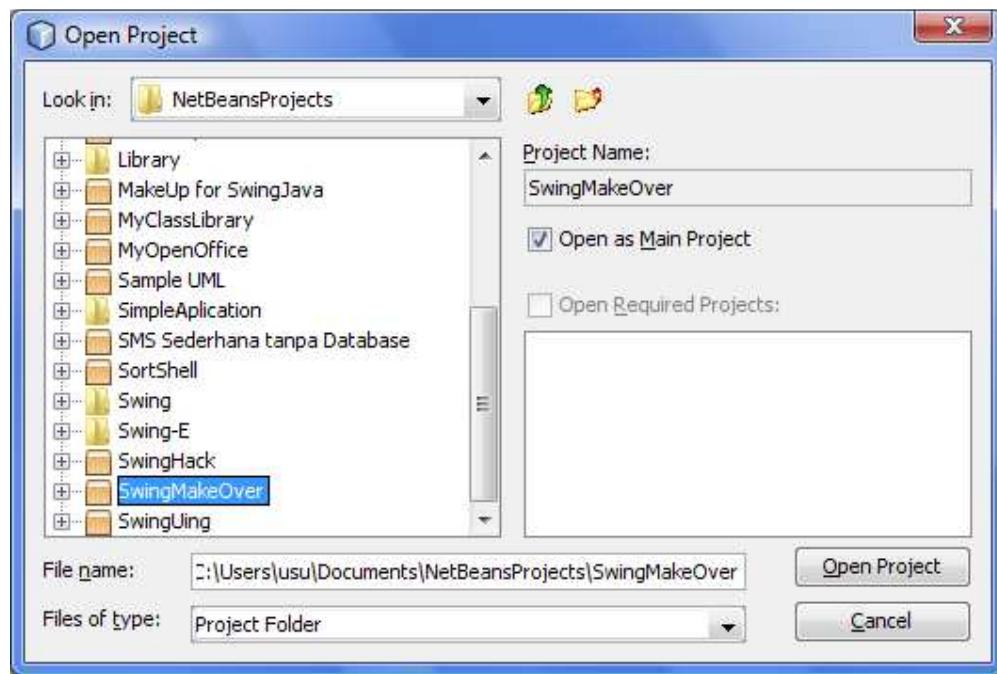
Sebenarnya tanpa saya buat tutorial ini anda pun pasti sudah tau cara membuat sebuah project dalam NetBeans, yach ini sich itung – itung membuka lembaran baru, hehehe □.

OK seperti kebanyakan Aplikasi untuk membuka sebuah file pasti lewat menu File. Dan dalam NetBeans untuk membuka Project anda bila lewat menu File > Open Project



Gambar 1 Menu Open Project

Lalu pilih project SwingMakeOver



Gambar 2 Open Project

Dan selesai, sekarang anda akan melihat project SwingMakeOver di Project Pallette

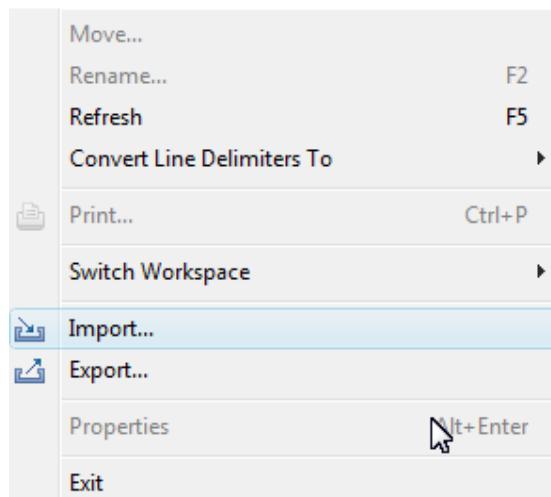


Gambar 3 Project Explorer

Project SwingMakeOver dan Eclipse

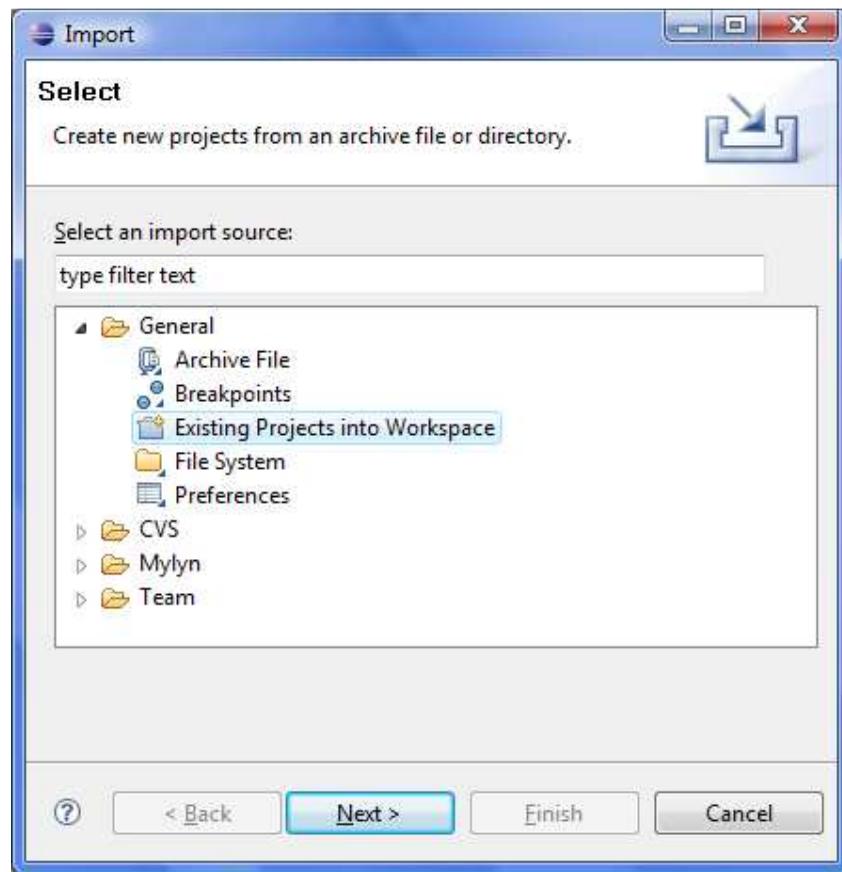
Jika anda menggunakan selain Eclipse IDE atau NetBeans IDE, anda cukup mengcopy paste source code yang ada dalam folder src.

Dalam Eclipse untuk membuka sebuah project, kita tak bisa langsung membuka project seperti yang dilakukan pada NetBeans, tapi kita bisa memanfaatkan metode Import untuk membuka sebuah project lewat menu File > Import.



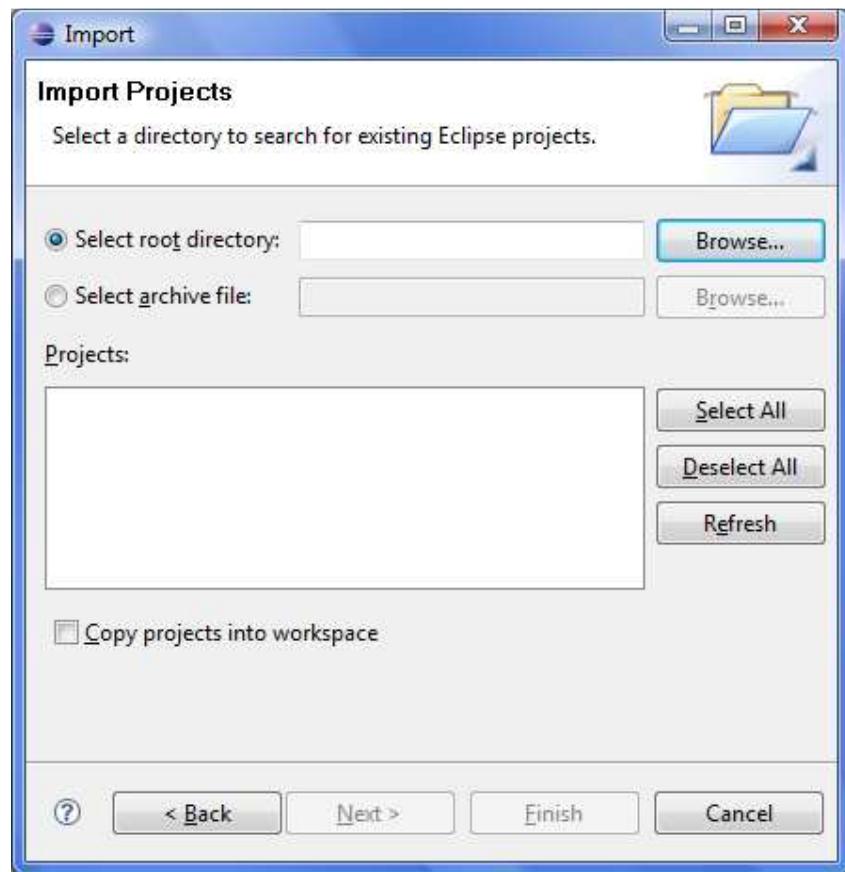
Gambar 4 Menu Import

Lalu pilih Existing Project into Workspace



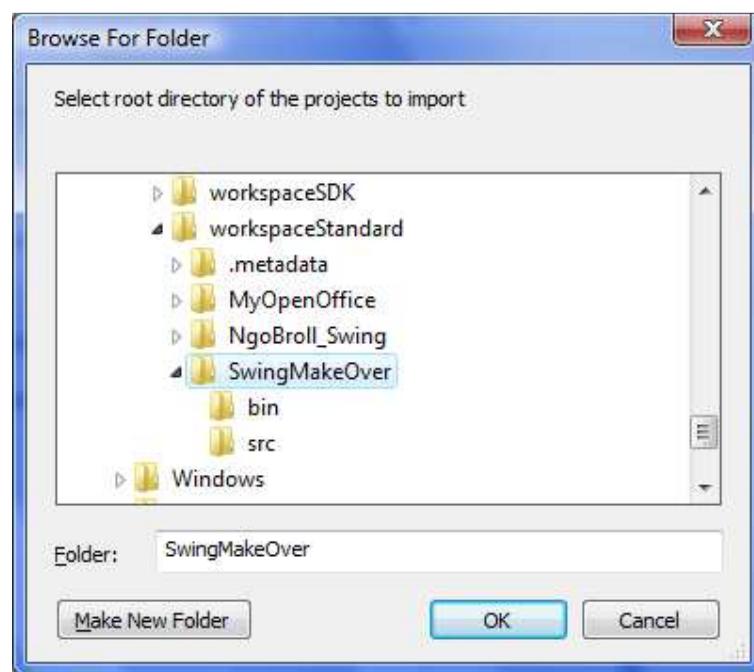
Gambar 5 Import

Pilih Slect root Directory lalu Browse



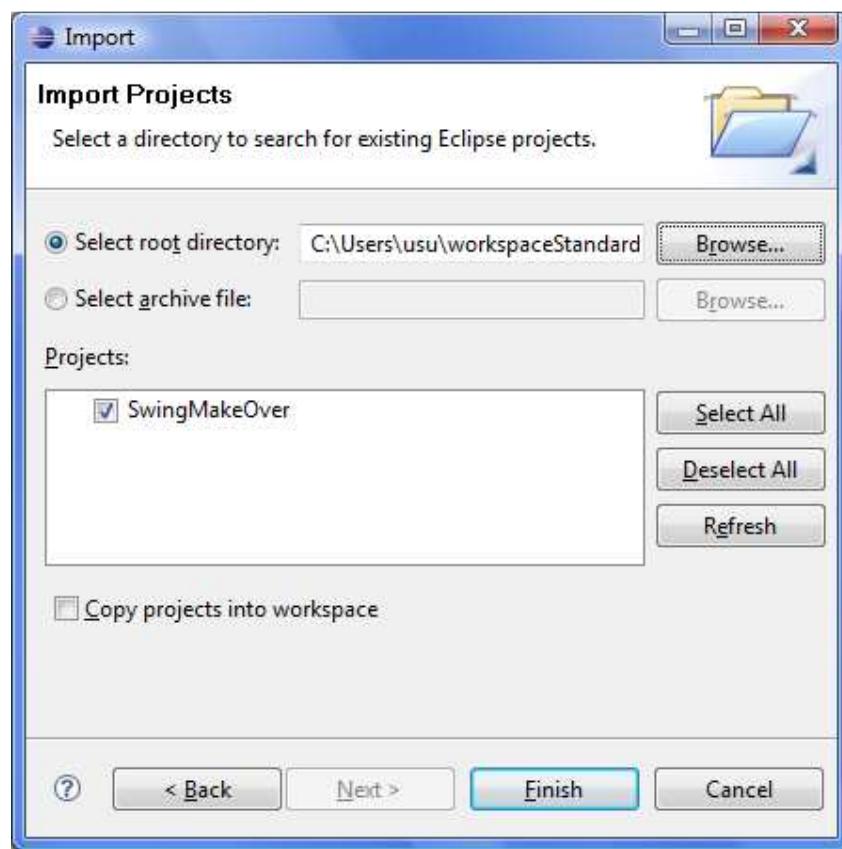
Gambar 6 Import

Pilihlah folder SwingMakeOver



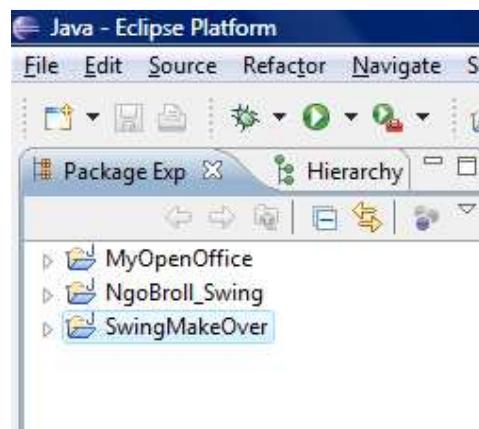
Gambar 7 Browse For Folder

Dan klik button Finish



Gambar 8 Import

Selesai. Sekarang anda bisa melihat project SwingMakeOver di project explorer



Gambar 9 Package Explorer

JPanel

Pemanasan

Mungkin ini yang kedua kalinya saya bilang JANGAN MENG OVERRIDE METODE paint(Graphics) TAPI OVERRIDE METODE paintComponent(Graphics). Mengapa demikian, karena jika kita mengoverride metode paint(Graphics), maka dengan demikian kita juga mengoverride metode paintComponent(Graphics), paintBorder(Graphics) dan paintChildren(Graphics). Kecuali emang anda ingin mengoverride seluruh metode tadi, maka gunakan paint(Graphics).

Pada bab ini kita akan banyak sekali membahas tentang menggambar dan mewarnai. Dalam paintComponent(Graphics) kita bisa melakukan semuanya, baik itu menggambar, mewarnai, bahkan memberi efek – efek yang mungkin jarang kita lihat dalam bahasa pemrograman visual yang lain. Nah bagaimana tertarik bukan? OK let's EXTREEM!

Manipulasi paintComponent()

Seperti yang telah saya beritahu sebelumnya, Extreem Swing akan banyak sekali menemui dengan metode paintComponent(), karena inilah salah satu kelebihan Swing dibandingkan bahasa pemrograman visual lainnya.

Menggambar Bentuk

Salah satu yang bisa kita lakukan dalam metode paintComponent() adalah menggambar bentuk, baik itu kotak, lingkaran atau sesuatu yang kita inginkan. Dan perlu diketahui parameter yang digunakan paintComponent() adalah Graphics, namun yang kita perlukan adalah Graphics2D, sehingga kita perlu mengconvert Graphics menjadi Graphics2D :

```
protected void paintComponent(Graphics g){  
    super.paintComponent(g);  
    Graphics2D g2 = (Graphics2D) g.create();  
}
```

Selain itu untuk menggambar bentuk dengan Graphics2D kita bisa menggunakan metode draw(Shape), sehingga kita bisa menggambar seluruh bentuk yang menjadi keturunan class Shape seperti Rectangle2D, Ellipse2D, atau RoundRectangle2D.

Kotak Lancip

Untuk membuat atau menggambar kotak dalam sebuah JPanel kita bisa membuatnya dalam metode paintComponent(). Dan untuk membuat kotak kita harus menggunakan class Rectangle2D.Double atau Rectangle2D.Float.

Untuk membuat kotak lancip gunakan kode dibawah ini :

```
protected void paintComponent( Graphics g ) {  
    super.paintComponent(g);
```

```
Rectangle2D.Double kotak = new Rectangle2D.Double(int x, int y, int lebar, int tinggi);
```

```
Graphics2D g2 = (Graphics2D) g.create();
g2.setColor(Color warna);
g2.draw(kotak);
}
```

PanelKotak.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Rectangle2D;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

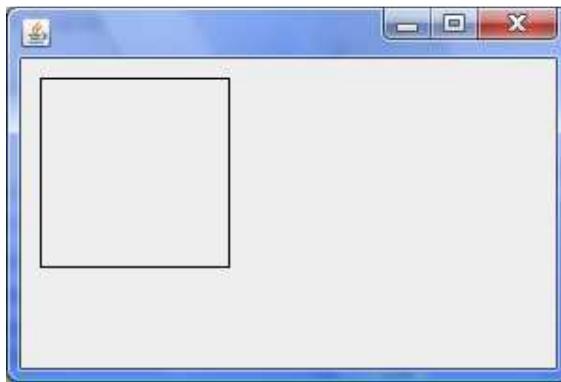
/**
 * @author usu
 */
public class PanelKotak extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
                frame.add(new PanelKotak());
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setVisible(true);
            }
        });
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final Rectangle2D.Double kotak = new Rectangle2D.Double(10, 10, 100,
100);
        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setColor(Color.BLACK);
        g2.draw(kotak);
    }
}
```



Gambar 10 PanelKotak.java

PanelKotak2.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Rectangle2D;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelKotak2 extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
                frame.add(new PanelKotak2());
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setVisible(true);
            }
        });
    }

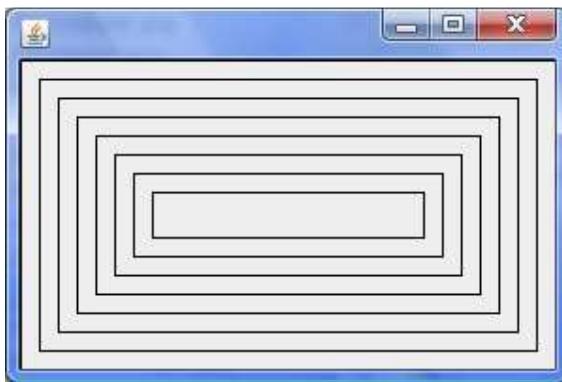
    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        Rectangle2D.Double kotak = null;

        int height = getHeight();
        int width = getWidth();
        int x = 0;
        int y = 0;

        while ((height > 10) && (width > 10)) {
            kotak = new Rectangle2D.Double(x, y, width, height);
            final Graphics2D g2 = (Graphics2D) g.create();
            g2.draw(kotak);
            g2.dispose();
            height -= 1;
            width -= 1;
            y += 1;
        }
    }
}
```

```
        g2.setColor(Color.BLACK);
        g2.draw(kotak);
        x += 10;
        y += 10;
        height -= 20;
        width -= 20;
    }
}
}
```



Gambar 11 PanelKotak2.java

Kotak Tumpul

Untuk membuat kotak kotak yang ujungnya tak lancip kita bisa menggunakan class RoundRectangle2D.

Untuk membuat kotak tumpul gunakan kode dibawah ini :

```
protected void paintComponent(Graphics g) {
    super.paintComponent(g);

    RoundRectangle2D kotak = new RoundRectangle2D.Double(int x, int y, int lebar, int tinggi, int
roundx, int roundy);

    Graphics2D g2 = (Graphics2D) g.create();
    g2.setColor(Color warna);
    g2.draw(kotak);
}
```

PanelRound.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.RoundRectangle2D;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;
```

```
/*
 * @author usu
 */
public class PanelRound extends JPanel {

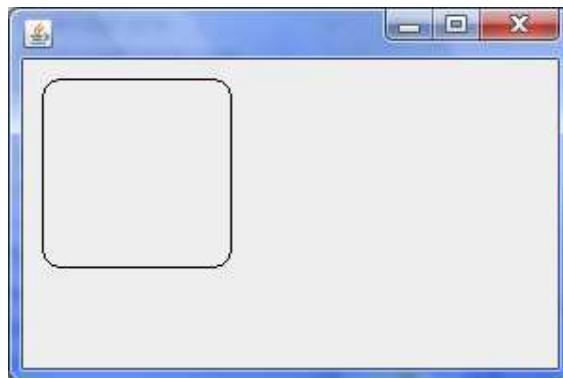
    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.add(new PanelRound());
                frame.setBounds(100, 100, 300, 200);
                frame.setVisible(true);
            }
        });
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final RoundRectangle2D kotak = new RoundRectangle2D.Double(10, 10,
100, 100, 20, 20);

        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setColor(Color.BLACK);
        g2.draw(kotak);
    }
}
```



Gambar 12 PanelRound.java

PanelRound2.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.RoundRectangle2D;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
```

```
import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelRound2 extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

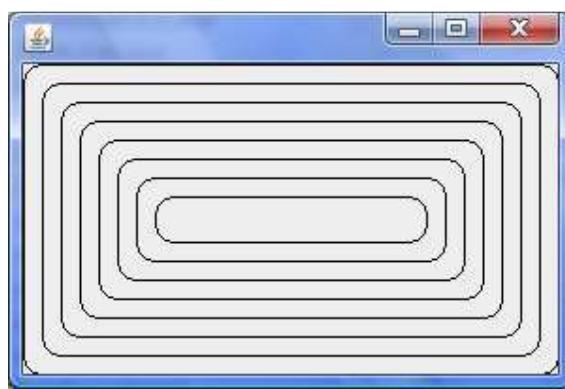
            public void run() {
                final Frame frame = new Frame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.add(new PanelRound2());
                frame.setBounds(100, 100, 300, 200);
                frame.setVisible(true);
            }
        });
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        RoundRectangle2D.Double kotak = null;

        int height = getHeight();
        int width = getWidth();
        int x = 0;
        int y = 0;

        while ((height > 10) && (width > 10)) {
            kotak = new RoundRectangle2D.Double(x, y, width, height, 20, 20);
            final Graphics2D g2 = (Graphics2D) g.create();
            g2.setColor(Color.BLACK);
            g2.draw(kotak);
            x += 10;
            y += 10;
            height -= 20;
            width -= 20;
        }
    }
}
```



Gambar 13 PanelRound2.java

Lingkaran

Sama seperti halnya membuat kotak, kita juga dapat menggambar lingkaran di JPanel anda dapat menggunakan class Ellipse2D.

Untuk membuat lingkaran, gunakan kode dibawah ini :

```
protected void paintComponent(Graphics g) {
    super.paintComponent(g);

    Ellipse2D.Double lingkaran = new Ellipse2D.Double(int x, int y, int lebar, int tinggi);
    Graphics2D g2 = (Graphics2D) g.create();
    g2.setColor(Color warna);
    g2.draw(lingkaran);
}
```

PanelLingkaran.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Ellipse2D;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

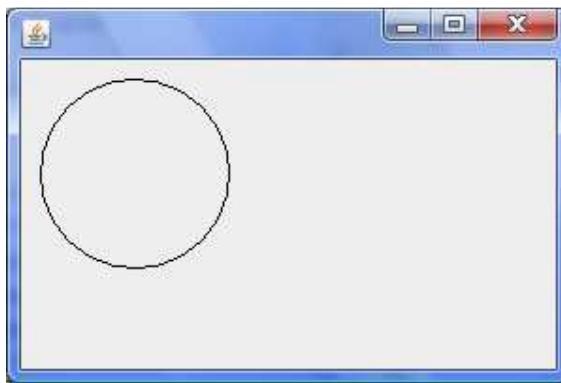
/**
 * @author usu
 */
public class PanelLingkaran extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.add(new PanelLingkaran());
                frame.setVisible(true);
            }
        });
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final Ellipse2D.Double lingkaran = new Ellipse2D.Double(10, 10, 100,
100);
        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setColor(Color.BLACK);
        g2.draw(lingkaran);
    }
}
```



Gambar 14 PanelLingkaran .java

PanelLingkaran2.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Ellipse2D;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelLingkaran2 extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new
            Runnable() {
                public void run() {
                    final Frame frame = new Frame();
                    frame.setBounds(100, 100, 300, 200);
                    frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                    frame.add(new PanelLingkaran2());
                    frame.setVisible(true);
                }
            });
    }

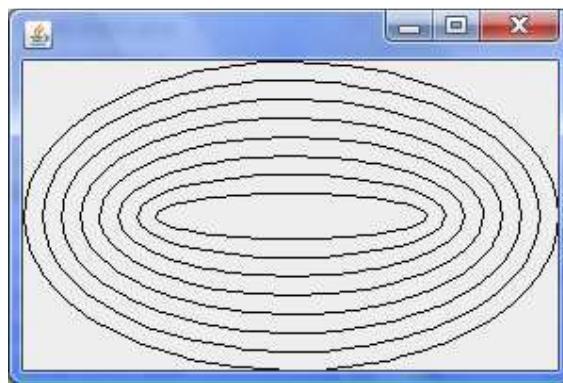
    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        Ellipse2D.Double kotak = null;

        int height = getHeight();
        int width = getWidth();
        int x = 0;
```

```
int y = 0;

while ((height > 10) && (width > 10)) {
    kotak = new Ellipse2D.Double(x, y, width, height);
    final Graphics2D g2 = (Graphics2D) g.create();
    g2.setColor(Color.BLACK);
    g2.draw(kotak);
    x += 10;
    y += 10;
    height -= 20;
    width -= 20;
}
}
```



Gambar 15 PanelLingkaran2.java

Bentuk Lain

Selain kotak atau lingkaran, anda juga bisa menggambar bentuk sesuai dengan yang anda inginkan anda dapat menggunakan class GeneralPath

```
protected void paintComponent(Graphics g) {
    super.paintComponent(g);

    GeneralPath path = new GeneralPath();
    path.moveTo(int x, int y); // titik awal path
    path...
    path...
    path.closePath(); // menutup path

    Graphics2D g2 = (Graphics2D) g.create();
    g2.setColor(Color warna);
    g2.draw(path);
}
```

PanelBentukLain.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
```

```
import java.awt.Graphics2D;
import java.awt.geom.GeneralPath;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelBentukLain extends JPanel {

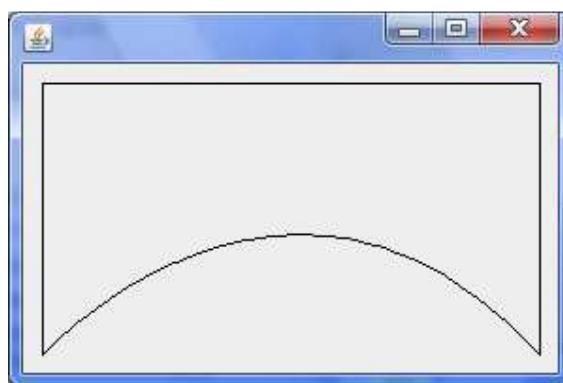
    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
                frame.add(new PanelBentukLain());
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setVisible(true);
            }
        });
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final GeneralPath path = new GeneralPath();
        path.moveTo(10, 10);
        path.lineTo(10, getHeight() - 10);
        path.curveTo(10, getHeight() - 10, getWidth() / 2, 10, getWidth() - 10, getHeight() - 10);
        path.lineTo(getWidth() - 10, 10);
        path.closePath();

        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setColor(Color.BLACK);
        g2.draw(path);
    }
}
```



Gambar 16 PanelBentukLain.java

Perlu diketahui untuk membuat GeneralPath, kita harus menentukan titik awal dulu dengan menggunakan metode `moveTo(x,y)` karena jika tidak menggunakannya maka akan terjadi Error, dan untuk menutup GeneralPath gunakan metode `closePath()`, sehingga titik terakhir path akan otomatis disambungkan dengan titik awal, karena jika tidak maka path akan terbuka.

Mewarnai Bentuk dengan Solid Color

Untuk mewarnai bentuk baik itu kotak ataupun lingkaran, kita cukup menggunakan metode `fill(Shape)` milik `Graphics2D`.

Kotak Lancip

Gunakan kode untuk membuat kotak lalu beri metode `fill()` :

PanelKotak.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Rectangle2D;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

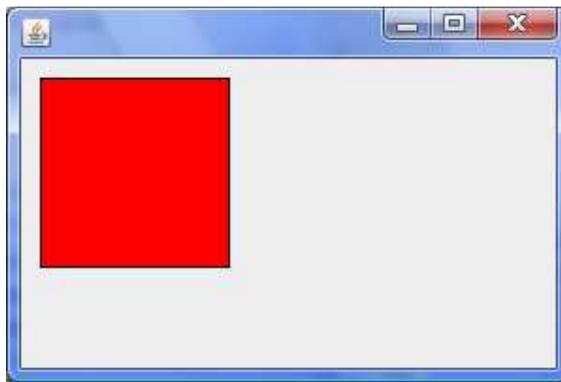
/**
 * @author usu
 */
public class PanelKotak extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
                frame.add(new PanelKotak());
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setVisible(true);
            }
        });
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final Rectangle2D.Double kotak = new Rectangle2D.Double(10, 10, 100,
100);
        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setColor(Color.RED);
        g2.fill(kotak);
        g2.setColor(Color.BLACK);
        g2.draw(kotak);
    }
}
```



Gambar 17 PanelKotak.java

PanelKotak2.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Rectangle2D;
import java.util.Random;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelKotak2 extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
                frame.add(new PanelKotak2());
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setVisible(true);
            }
        });
    }

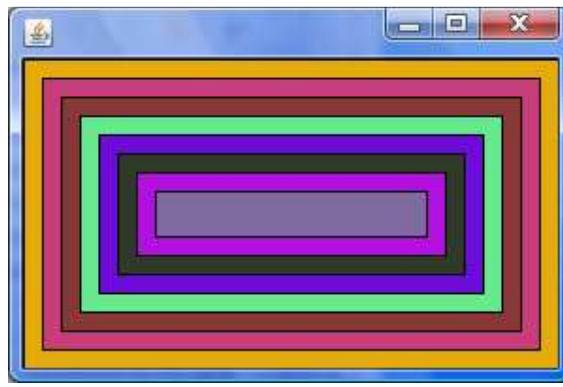
    private final Random generator = new Random();

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        Rectangle2D.Double kotak = null;

        int height = getHeight();
        int width = getWidth();
        int x = 0;
        int y = 0;
```

```
while ((height > 10) && (width > 10)) {
    kotak = new Rectangle2D.Double(x, y, width, height);
    final Graphics2D g2 = (Graphics2D) g.create();
    g2.setColor(new Color(this.generator.nextInt(255), this.generator
        .nextInt(255), this.generator.nextInt(255)));
    g2.fill(kotak);
    g2.setColor(Color.BLACK);
    g2.draw(kotak);
    x += 10;
    y += 10;
    height -= 20;
    width -= 20;
}
}
```



Gambar 18 PanelKotak2.java

Kotak Tumpul

Gunakan kode untuk membuat kotak Round lalu beri metode fill() :

PanelRound.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.RoundRectangle2D;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelRound extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

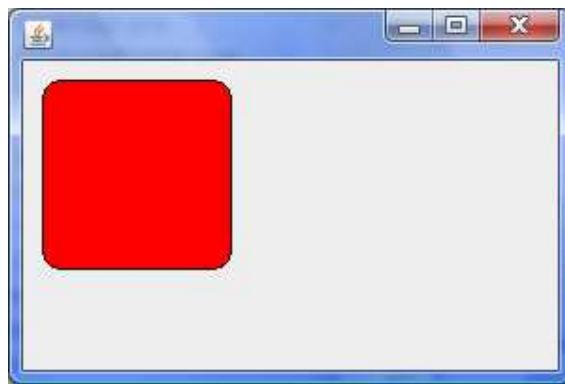
            public void run() {
                final Frame frame = new Frame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
            }
        });
    }
}
```

```
        frame.add(new PanelRound());
        frame.setBounds(100, 100, 300, 200);
        frame.setVisible(true);
    }
}

@Override
protected void paintComponent(final Graphics g) {
    super.paintComponent(g);

    final RoundRectangle2D kotak = new RoundRectangle2D.Double(10, 10,
100, 100, 20, 20);

    final Graphics2D g2 = (Graphics2D) g.create();
    g2.setColor(Color.RED);
    g2.fill(kotak);
    g2.setColor(Color.BLACK);
    g2.draw(kotak);
}
}
```



Gambar 19 PanelRound.java

PanelRound2.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.RoundRectangle2D;
import java.util.Random;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelRound2 extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {
```

```
public void run() {
    final Frame frame = new Frame();
    frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    frame.add(new PanelRound2());
    frame.setBounds(100, 100, 300, 200);
    frame.setVisible(true);
}
}

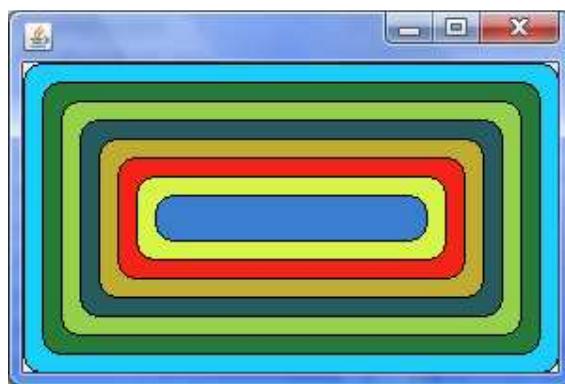
private final Random generator = new Random();

@Override
protected void paintComponent(final Graphics g) {
    super.paintComponent(g);

    RoundRectangle2D.Double kotak = null;

    int height = getHeight();
    int width = getWidth();
    int x = 0;
    int y = 0;

    while ((height > 10) && (width > 10)) {
        kotak = new RoundRectangle2D.Double(x, y, width, height, 20, 20);
        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setColor(new Color(this.generator.nextInt(255), this.generator
            .nextInt(255), this.generator.nextInt(255)));
        g2.fill(kotak);
        g2.setColor(Color.BLACK);
        g2.draw(kotak);
        x += 10;
        y += 10;
        height -= 20;
        width -= 20;
    }
}
}
```



Gambar 20 PanelRound2.java

Lingkaran

Gunakan kode untuk membuat lingkaran lalu beri metode fill() :

PanelLingkaran.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Ellipse2D;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

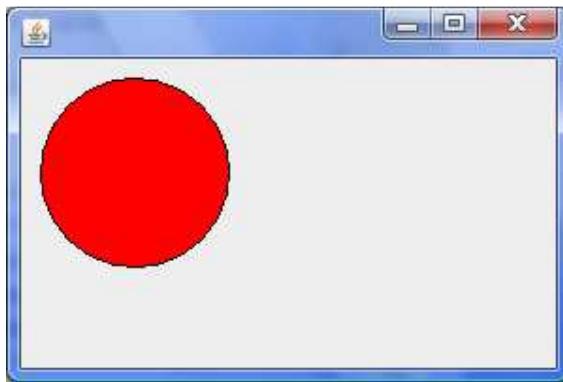
/**
 * @author usu
 */
public class PanelLingkaran extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.add(new PanelLingkaran());
                frame.setVisible(true);
            }
        });
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final Ellipse2D.Double lingkaran = new Ellipse2D.Double(10, 10, 100,
100);
        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setColor(Color.RED);
        g2.fill(lingkaran);
        g2.setColor(Color.BLACK);
        g2.draw(lingkaran);
    }
}
```



Gambar 21 PanelLingkaran.java

PanelLingkaran2.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Ellipse2D;
import java.util.Random;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelLingkaran2 extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.add(new PanelLingkaran2());
                frame.setVisible(true);
            }
        });
    }

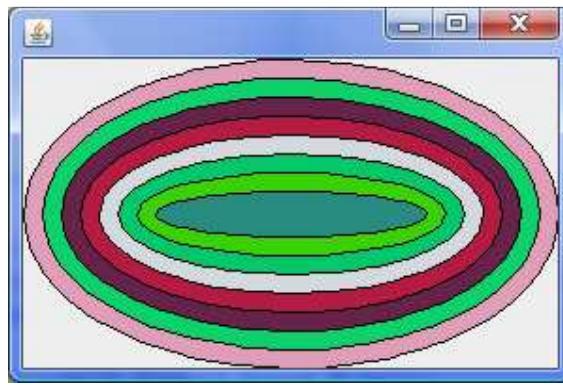
    private final Random generator = new Random();

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        Ellipse2D.Double kotak = null;

        int height = getHeight();
        int width = getWidth();
        int x = 0;
        int y = 0;
```

```
while ((height > 10) && (width > 10)) {
    kotak = new Ellipse2D.Double(x, y, width, height);
    final Graphics2D g2 = (Graphics2D) g.create();
    g2.setColor(new Color(this.generator.nextInt(255), this.generator
        .nextInt(255), this.generator.nextInt(255)));
    g2.fill(kotak);
    g2.setColor(Color.BLACK);
    g2.draw(kotak);
    x += 10;
    y += 10;
    height -= 20;
    width -= 20;
}
}
```



Gambar 22 PanelLingkaran2.java

Bentuk Lain

Gunakan kode untuk menggambar bentuk lain lalu tambahkan metode fill() :

PanelBentukLain.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.GeneralPath;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelBentukLain extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

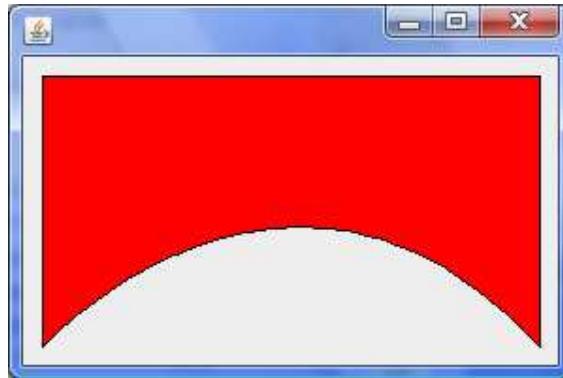
            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
            }
        });
    }
}
```

```
        frame.add(new PanelBentukLain());
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}

@Override
protected void paintComponent(final Graphics g) {
    super.paintComponent(g);

    final GeneralPath path = new GeneralPath();
    path.moveTo(10, 10);
    path.lineTo(10, getHeight() - 10);
    path.curveTo(10, getHeight() - 10, getWidth() / 2, 10, getWidth() - 10, getHeight() - 10);
    path.lineTo(getWidth() - 10, 10);
    path.closePath();

    final Graphics2D g2 = (Graphics2D) g.create();
    g2.setColor(Color.RED);
    g2.fill(path);
    g2.setColor(Color.BLACK);
    g2.draw(path);
}
}
```



Gambar 23 PanelBentukLain.java

Mewarnai Bentuk dengan Gradient Color

Mungkin saat ini jika kita memberi warna latar sebuah JPanel dengan warna solid bisa dibilang sudah kuno. Saat ini aplikasi lebih menonjolkan efek 3D dibandingkan efek 2D, jadi mau gak mau kita harus menggunakan warna gradient untuk memunculkan efek 3D

Linear Gradient

Untuk membuat linear gradient, gunakan kode dibawah ini :

```
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
```

Shape path = ... // Buat bentuk yang akan diwarnai

```
GradientPaint paint = new GradientPaint(int x1, int y1, Color warna1,int x2, int y2, Color warna);
```

```
Graphics2D g2 = (Graphics2D) g.create();
g2.setPaint(paint);
g2.fill(path);
}
```

PanelLinear.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.GeneralPath;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelLinear extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
                frame.add(new PanelLinear());
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setVisible(true);
            }
        });
    }

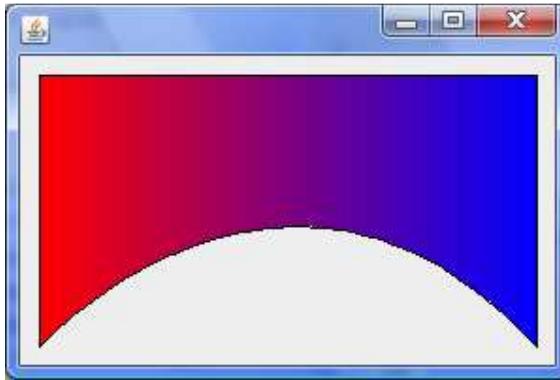
    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final GeneralPath path = new GeneralPath();
        path.moveTo(10, 10);
        path.lineTo(10, getHeight() - 10);
        path.curveTo(10, getHeight() - 10, getWidth() / 2, 10, getWidth() - 10, getHeight() - 10);
        path.lineTo(getWidth() - 10, 10);
        path.closePath();

        final GradientPaint paint = new GradientPaint(10, 10, Color.RED,
getWidth() - 10, 10,
Color.BLUE);

        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setPaint(paint);
    }
}
```

```
    g2.fill(path);
    g2.setColor(Color.BLACK);
    g2.draw(path);
}
}
```



Gambar 24 PanelLinear.java

Radial Gradient

Untuk membuat gradient lingkaran anda dapat menggunakan kode seperti dibawah :

```
protected void paintComponent(Graphics g) {
    super.paintComponent(g);

    Shape path = ... // Buat bentuk

    Point center = ...
    float radius = ...
    float[] fractions = ...
    Color[] colors = ...
    RadialGradientPaint paint = new RadialGradientPaint(center, radius, fractions, colors);

    Graphics2D g2 = (Graphics2D) g.create();
    g2.setPaint(paint);
    g2.fill(path);
}
```

PanelRadial.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Point;
import java.awt.RadialGradientPaint;
import java.awt.geom.GeneralPath;

import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
```

```
import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelRadial extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

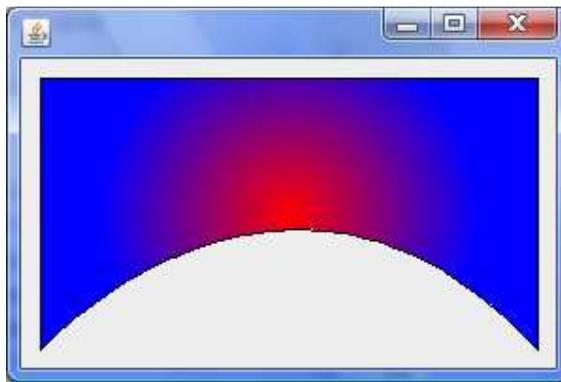
            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
                frame.add(new PanelRadial());
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setVisible(true);
            }
        });
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final GeneralPath path = new GeneralPath();
        path.moveTo(10, 10);
        path.lineTo(10, getHeight() - 10);
        path.curveTo(10, getHeight() - 10, getWidth() / 2, 10, getWidth() - 10, getHeight() - 10);
        path.lineTo(getWidth() - 10, 10);
        path.closePath();

        final Point center = new Point(getWidth() / 2, getHeight() / 2);
        final float radius = 100;
        final float[] fractions = new float[] { 0.0F, 1.0F };
        final Color[] colors = new Color[] { Color.RED, Color.BLUE };
        final RadialGradientPaint paint = new RadialGradientPaint(center, radius, fractions, colors);

        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setPaint(paint);
        g2.fill(path);
        g2.setColor(Color.BLACK);
        g2.draw(path);
    }
}
```



Gambar 25 PanelRadial.java

Multiple Linear Gradient

Kadang anda perlu membuat linear gradient dengan banyak warna, untuk membuatnya anda bisa menggunakan kode seperti dibawah ini :

```
protected void paintComponent(Graphics g) {  
    super.paintComponent(g);  
  
    Shape path = ...  
  
    Point start = ...  
    Point end = ...  
    float[] fractions = ...  
    Color[] colors = ...  
    LinearGradientPaint paint = new LinearGradientPaint(start, end, fractions, colors);  
  
    final Graphics2D g2 = (Graphics2D) g.create();  
    g2.setPaint(paint);  
    g2.fill(path);  
}
```

PanelLinearMultiple.java

```
package pelajaran3;  
  
import java.awt.Color;  
import java.awt.Graphics;  
import java.awt.Graphics2D;  
import java.awt.LinearGradientPaint;  
import java.awt.Point;  
import java.awt.geom.GeneralPath;  
  
import javax.swing.JPanel;  
import javax.swing.SwingUtilities;  
import javax.swing.WindowConstants;  
  
import pelajaran1.Frame;  
  
/**
```

```
* @author usu
*/
public class PanelLinearMultiple extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

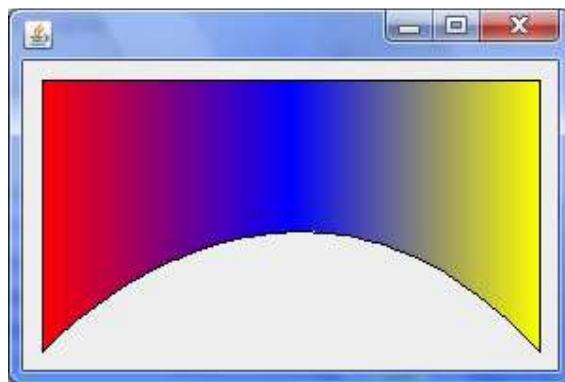
            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
                frame.add(new PanelLinearMultiple());
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setVisible(true);
            }
        });
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final GeneralPath path = new GeneralPath();
        path.moveTo(10, 10);
        path.lineTo(10, getHeight() - 10);
        path.curveTo(10, getHeight() - 10, getWidth() / 2, 10, getWidth() - 10, getHeight() - 10);
        path.lineTo(getWidth() - 10, 10);
        path.closePath();

        final Point start = new Point(10, 10);
        final Point end = new Point(getWidth() - 10, 10);
        final float[] fractions = new float[] { 0.0F, 0.5F, 1.0F };
        final Color[] colors = new Color[] { Color.RED, Color.BLUE,
        Color.YELLOW };
        final LinearGradientPaint paint = new LinearGradientPaint(start, end,
        fractions, colors);

        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setPaint(paint);
        g2.fill(path);
        g2.setColor(Color.BLACK);
        g2.draw(path);
    }
}
```



Gambar 26 PanelLinearMultiple.java

Multiple Radial Gradient

Seperti halnya membuat linear gradient dengan banyak warna, anda pun pasti kadang memerlukan warna radiel gradient yang lebih dari dua warna. Dan untuk membuatnya sama halnya seperti membuat gambar dengan radial gradient namun dengan menggunakan float[] fractions lebih dari dua dan Color[] colors lebih dari dua :

PanelRadialMultiple.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Point;
import java.awt.RadialGradientPaint;
import java.awt.geom.GeneralPath;

import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;

import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelRadialMultiple extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
                frame.add(new PanelRadialMultiple());
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setVisible(true);
            }
        });
    }

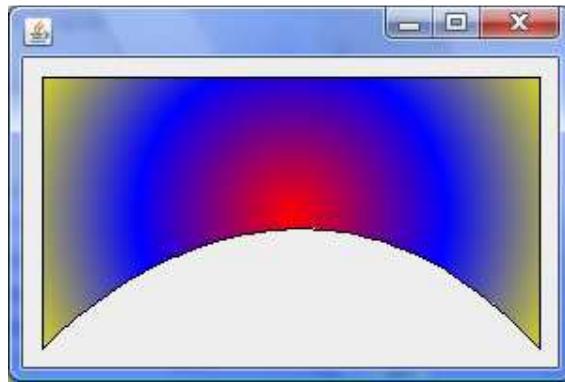
    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final GeneralPath path = new GeneralPath();
        path.moveTo(10, 10);
        path.lineTo(10, getHeight() - 10);
        path.curveTo(10, getHeight() - 10, getWidth() / 2, 10, getWidth() -
        10, getHeight() - 10);
        path.lineTo(getWidth() - 10, 10);
        path.closePath();

        final Point center = new Point(getWidth() / 2, getHeight() / 2);
        final float radius = getHeight();
        final float[] fractions = new float[] { 0.0F, 0.5F, 1.0F };
        final Color[] colors = new Color[] { Color.RED, Color.BLUE,
        Color.YELLOW };
        final RadialGradientPaint paint = new RadialGradientPaint(center,
```

```
radius, fractions, colors);

final Graphics2D g2 = (Graphics2D) g.create();
g2.setPaint(paint);
g2.fill(path);
g2.setColor(Color.BLACK);
g2.draw(path);
}
```

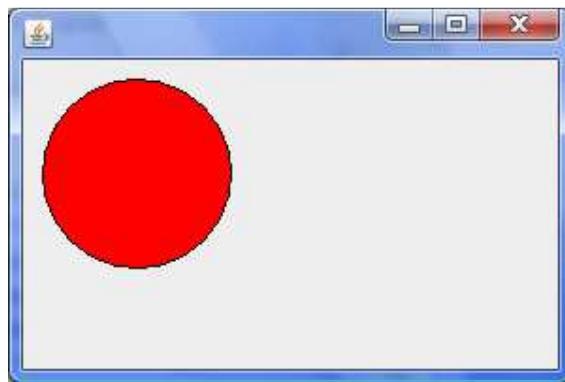


Gambar 27 PanelRadialMultiple.java

Antialiasing

Sebenarnya subbab ini bisa dibilang kelanjutan dari subbab “Memanupulasi paintComponent()”, karena memang subbab ini sangat terkait dengan subbab sebelumnya.

Mungkin anda yang sering memainkan Game PC, telah mengenal kata antialiasing atau dengan kata lain mempertajam gambar dengan cara memberi sedikit efek blur, contoh gambar yang belum kita beri antialiasing adalah gambar – gambar yang tadi kita buat dalam subbab sebelumnya :



Gambar 28 PanelLingkaran sebelum antialiasing

Terlihat dengan jelas kalo pingiran gambar lingkaran tadi tidak halus. Karena memang belum diberi antialiasing. Sehingga untuk memberi kehalusan pada gambar kita perlu memberi efek antialiasing pada gambar tersebut.

Untuk membuat antialiasing ON, gunakan :

`g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,RenderingHints.VALUE_ANTIALIAS_ON);`

PanelLingkaranAntialiassing.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
import java.awt.geom.Ellipse2D;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

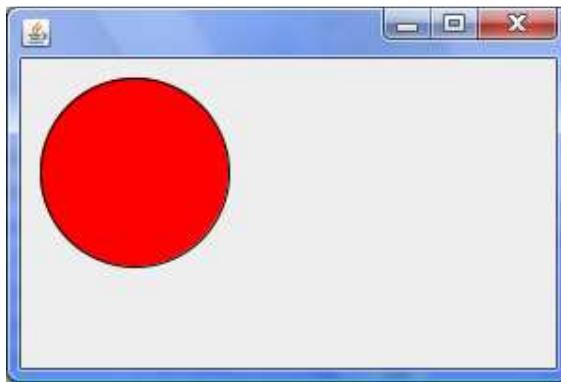
/**
 * @author usu
 */
public class PanelLingkaranAntialiassing extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.add(new PanelLingkaranAntialiassing());
                frame.setVisible(true);
            }
        });
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

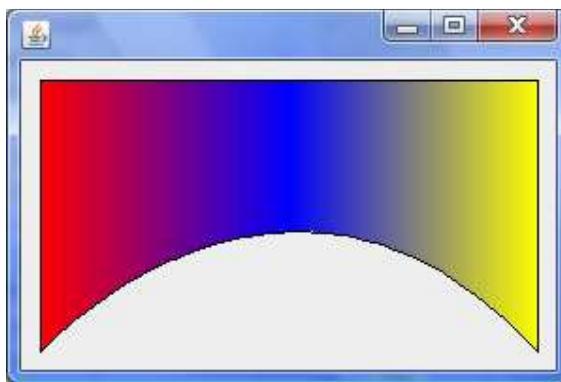
        final Ellipse2D.Double lingkaran = new Ellipse2D.Double(10, 10, 100,
100);
        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
        g2.setColor(Color.RED);
        g2.fill(lingkaran);
        g2.setColor(Color.BLACK);
        g2.draw(lingkaran);
    }
}
```



Gambar 29 PanelLingkaranAntialiasing.java

Sekarang anda bisa melihat perbedaan gambar sebelum dan setelah diberi efek antialiasing.
Contoh lain :

Sebelum diberi efek antialiasing



Gambar 30 PanelLinear sebelum antialiasing

PanelLinearMultipleAntialiassing.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.LinearGradientPaint;
import java.awt.Point;
import java.awt.RenderingHints;
import java.awt.geom.GeneralPath;

import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;

import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelLinearMultipleAntialiassing extends JPanel {
```

```
public static void main(final String[] usu) {
    SwingUtilities.invokeLater(new Runnable() {

        public void run() {
            final Frame frame = new Frame();
            frame.setBounds(100, 100, 300, 200);
            frame.add(new PanelLinearMultipleAntialiasing());
            frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
            frame.setVisible(true);
        }
    });
}

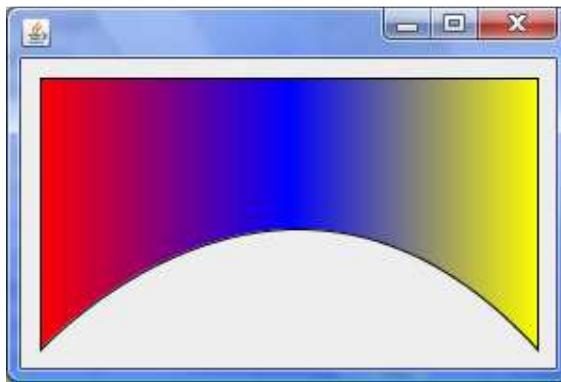
@Override
protected void paintComponent(final Graphics g) {
    super.paintComponent(g);

    final GeneralPath path = new GeneralPath();
    path.moveTo(10, 10);
    path.lineTo(10, getHeight() - 10);
    path.curveTo(10, getHeight() - 10, getWidth() / 2, 10, getWidth() - 10, getHeight() - 10);
    path.lineTo(getWidth() - 10, 10);
    path.closePath();

    final Point start = new Point(10, 10);
    final Point end = new Point(getWidth() - 10, 10);
    final float[] fractions = new float[] { 0.0F, 0.5F, 1.0F };
    final Color[] colors = new Color[] { Color.RED, Color.BLUE,
    Color.YELLOW };
    final LinearGradientPaint paint = new LinearGradientPaint(start, end,
    fractions, colors);

    final Graphics2D g2 = (Graphics2D) g.create();
    g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
    RenderingHints.VALUE_ANTIALIAS_ON);
    g2.setPaint(paint);
    g2.fill(path);
    g2.setColor(Color.BLACK);
    g2.draw(path);
}
}
```

Setelah diberi efek antialiasing



Gambar 31 PanelLinearMultipleAntialiasing.java

Alpha

Alpha yang saya maksud adalah teknik untuk membuat gambar atau warna menjadi tranparan. Sebenarnya ada 2 teknik, yang pertama adalah dengan memanipulasi warna (Color) tersebut dan yang kedua adalah dengan menggunakan Composite :

Color

Sebelumnya kita harus tahu untuk membuat object Color dengan mengatur alpha ada dua konstruktor :

Color warna = new Color(int red, int green, int blue, int alpha);

Dengan ketentuan bahwa nilai setiap parameter tidak lebih dari 255 dan tak kurang dari 0 dan semakin kecil nilai parameter alpha maka warna akan semakin tranparan.

Color warna = new Color(float red, float green, float blue, float alpha);

Dengan ketentuan bahwa nilai setiap parameter tidak lebih dari 1F dan tak kurang dari 0F sama seperti konstruktor sebelumnya semakin kecil nilai alphanya maka warna akan semakin tranparan.

PanelAlpha.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Rectangle2D;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
```

```
public class PanelAlpha extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.add(new PanelAlpha());
                frame.setVisible(true);
            }
        });
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final Rectangle2D.Double kotak = new Rectangle2D.Double(0, 0,
getWidth(), getHeight());
        final Rectangle2D.Double kotak2 = new Rectangle2D.Double(10, 10,
getWidth() - 20,
getHeight() - 20);

        final Color dark = new Color(0F, 0F, 0F, 0F);
        final Color light = new Color(1F, 0F, 0F, 0.5F);

        final GradientPaint paint = new GradientPaint(10, 10, dark, 10,
getHeight() - 20, light);

        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setColor(Color.BLUE);
        g2.fill(kotak);
        g2.setPaint(paint);
        g2.fill(kotak2);
    }
}
```



Gambar 32 PanelAlpha.java

Anda bisa melihat pada Color dark, saya buat alphanya bernilai 0F sehingga warna tersebut diperlihatkan secara transparan tergantung nilai alphanya.

Composite

Selain dengan memanipulasi warna anda juga bisa menggunakan Composite pada Graphics2D. Untuk menggunakan Composite gunakan kode dibawah ini :

```
g2.setComposite(AlphaComposite.SrcOver.derive(float value));
```

PanelComposite.java

```
package pelajaran3;

import java.awt.AlphaComposite;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Rectangle2D;

import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;

import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelComposite extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

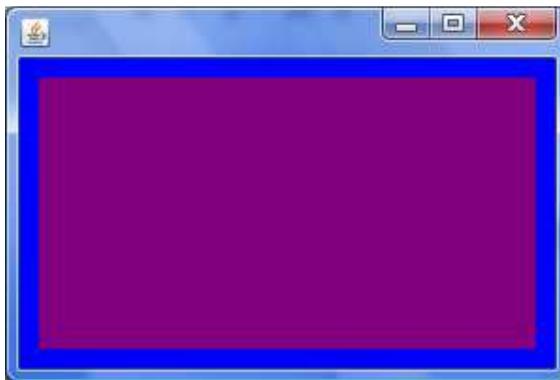
            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.add(new PanelComposite());
                frame.setVisible(true);
            }
        });
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final Rectangle2D.Double kotak = new Rectangle2D.Double(0, 0,
getWidth(), getHeight());
        final Rectangle2D.Double kotak2 = new Rectangle2D.Double(10, 10,
getWidth() - 20,
getHeight() - 20);

        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setColor(Color.BLUE);
        g2.fill(kotak);
        g2.setComposite(AlphaComposite.SrcOver.derive(0.5F));
        g2.setPaint(Color.RED);
        g2.fill(kotak2);
    }
}
```

```
}
```



Gambar 33 PanelComposite.java

Dengan memberi nilai AlphaComposite 0.5F maka warna merah akan ditampilkan transparan 50%, tergantung nilai alpha.

Perlu diketahui! Berhati – hatilah dalam penempatan AlphaComposite. Saya menempatkan setComposite() sebelum setPaint(Color.RED) sehingga dengan demikian gambar sebelumnya tak akan diberi efek transparan, sedangkan setelah pemanggilan metode setComposite(), maka gambar akan diberi efek transparan sesuai dengan tingkat transparansi yang kita berikan.

Image

Selain menggambar bentuk – bentuk yang dalam java dibilang Shape, kita juga dapat menggambar objek Image. Dan seperti penjelasan sebab sebelumnya hal inipun tak luput dari metode paintComponent() :

```
protected void paintComponent(final Graphics g) {  
    super.paintComponent(g);  
  
    Graphics2D g2 = (Graphics2D) g.create();  
    g2.drawImage(Image gambar, int x, int y, ImageObserver observer);  
}
```

PanellImage.java

```
package pelajaran3;  
  
import java.awt.Graphics;  
import java.awt.Graphics2D;  
import java.awt.Image;  
import javax.swing.ImageIcon;  
import javax.swing.JPanel;  
import javax.swing.SwingUtilities;  
import javax.swing.WindowConstants;  
import pelajaran1.Frame;
```

```
/**  
 * @author usu  
 */  
public class PanelImage extends JPanel {  
  
    public static void main(final String[] usu) {  
        SwingUtilities.invokeLater(new Runnable() {  
  
            public void run() {  
                final Frame frame = new Frame();  
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);  
                frame.setBounds(100, 100, 300, 300);  
                frame.add(new PanelImage());  
                frame.setVisible(true);  
            }  
        } );  
    }  
  
    @Override  
    protected void paintComponent(final Graphics g) {  
        super.paintComponent(g);  
  
        final Image gambar = new  
ImageIcon(getClass().getResource("/pelajaran3/My Computer.png"))  
            .getImage();  
  
        final Graphics2D g2 = (Graphics2D) g.create();  
        g2.drawImage(gambar, 10, 10, null);  
    }  
}
```



Gambar 34 PanelImage.java

PanelImage2.java

```
package pelajaran3;  
  
import java.awt.Color;
```

```
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.Rectangle2D;
import java.util.Random;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelKotak2 extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 300, 200);
                frame.add(new PanelKotak2());
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setVisible(true);
            }
        });
    }

    private final Random generator = new Random();

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        Rectangle2D.Double kotak = null;

        int height = getHeight();
        int width = getWidth();
        int x = 0;
        int y = 0;

        while ((height > 10) && (width > 10)) {
            kotak = new Rectangle2D.Double(x, y, width, height);
            final Graphics2D g2 = (Graphics2D) g.create();
            g2.setColor(new Color(this.generator.nextInt(255), this.generator.nextInt(255), this.generator.nextInt(255)));
            g2.fill(kotak);
            g2.setColor(Color.BLACK);
            g2.draw(kotak);
            x += 10;
            y += 10;
            height -= 20;
            width -= 20;
        }
    }
}
```



Gambar 35 PanellImage2.java

Efek Kaca

Kali ini saya akan memberi tahu cara yang sangat berguna, yaitu efek kaca untuk gambar. Buatlah class EfekGambar seperti yang ada dibawah ini :

EfekGambar.java

```
package pelajaran3;

import java.awt.AlphaComposite;
import java.awt.Color;
import java.awt.GradientPaint;
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.image.BufferedImage;

/**
 * @author usu
 */
public final class EfekGambar {

    public static final BufferedImage getBuffered(final Image source) {
        final int width = source.getWidth(null);
        final int height = source.getHeight(null);
        final BufferedImage dest = new BufferedImage(width, height,
        BufferedImage.TYPE_INT_ARGB);

        final Graphics2D g2 = (Graphics2D) dest.getGraphics();
        g2.drawImage(source, 0, 0, null);
        g2.dispose();

        return dest;
    }

    public static final BufferedImage getEfekKaca(final BufferedImage
source) {
```

```
final BufferedImage dest = new BufferedImage(source.getWidth(),
source.getHeight()
+ source.getHeight() / 2, BufferedImage.TYPE_INT_ARGB);

final Graphics2D g2 = (Graphics2D) dest.getGraphics();
g2.drawImage(source, 0, 0, null);
g2.scale(1, -1);
g2.drawImage(source, 0, -source.getHeight() * 2, null);
g2.scale(1, -1);
g2.translate(0, source.getHeight());
g2.setPaint(new GradientPaint(0, 0, new Color(1f, 1f, 1f, 0.3f), 0,
source.getHeight() / 2,
new Color(1f, 1f, 1f, 0f)));
g2.setComposite(AlphaComposite.DstIn);
g2.fillRect(0, 0, source.getWidth(), source.getHeight());
g2.dispose();

return dest;
}

public static final BufferedImage getEfekKaca(final Image source) {
    return EfekGambar.getEfekKaca(EfekGambar.getBuffered(source));
}
}
```

Class diatas digunakan sebagai class library untuk efek kaca, misal kita akan merubah gambar panel yang tadi sebelumnya kita gambar :

```
protected void paintComponent(final Graphics g) {
    super.paintComponent(g);

    Graphics2D g2 = (Graphics2D) g.create();
    g2.drawImage(EfekGambar.getEfekKaca(Image gambar), int x, int y, ImageObserver observer);
}
```

PanellImageKaca.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
import javax.swing.ImageIcon;
import javax.swing.JPanel;
import javax.swing.SwingConstantsUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelImageKaca extends JPanel {

    public static void main(final String[] usu) {
```

```
SwingUtilities.invokeLater(new Runnable() {

    public void run() {
        final Frame frame = new Frame();
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.setBounds(100, 100, 300, 300);
        frame.add(new PanelImageKaca());
        frame.setVisible(true);
    }
});

@Override
protected void paintComponent(final Graphics g) {
    super.paintComponent(g);

    final Image gambar = new
ImageIcon(getClass().getResource("/pelajaran3/AIM.png")).getImage();

    final Graphics2D g2 = (Graphics2D) g.create();
    g2.setPaint(new GradientPaint(0, 0, Color.BLUE, 0,
gambar.getHeight(null), Color.BLACK));
    g2.fillRect(0, 0, getWidth(), getHeight());
    g2.drawImage(EfekGambar.getEfekKaca(gambar), 10, 10, null);
}

}
```



Gambar 36 PanelImageKaca.java

PanellImageKaca2.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
```

```
import javax.swing.ImageIcon;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class PanelImageKaca2 extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setBounds(100, 100, 300, 300);
                frame.add(new PanelImageKaca2());
                frame.setVisible(true);
            }
        });
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final Image gambar = new ImageIcon(getClass().getResource(
            "/pelajaran3/My Computer.png")).getImage();

        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setPaint(new GradientPaint(0, 0, Color.BLUE, 0,
            gambar.getHeight(null), Color.BLACK));
        g2.fillRect(0, 0, getWidth(), getHeight());
        g2.drawImage(EfekGambar.getEfekKaca(gambar), 10, 10, null);
    }
}
```

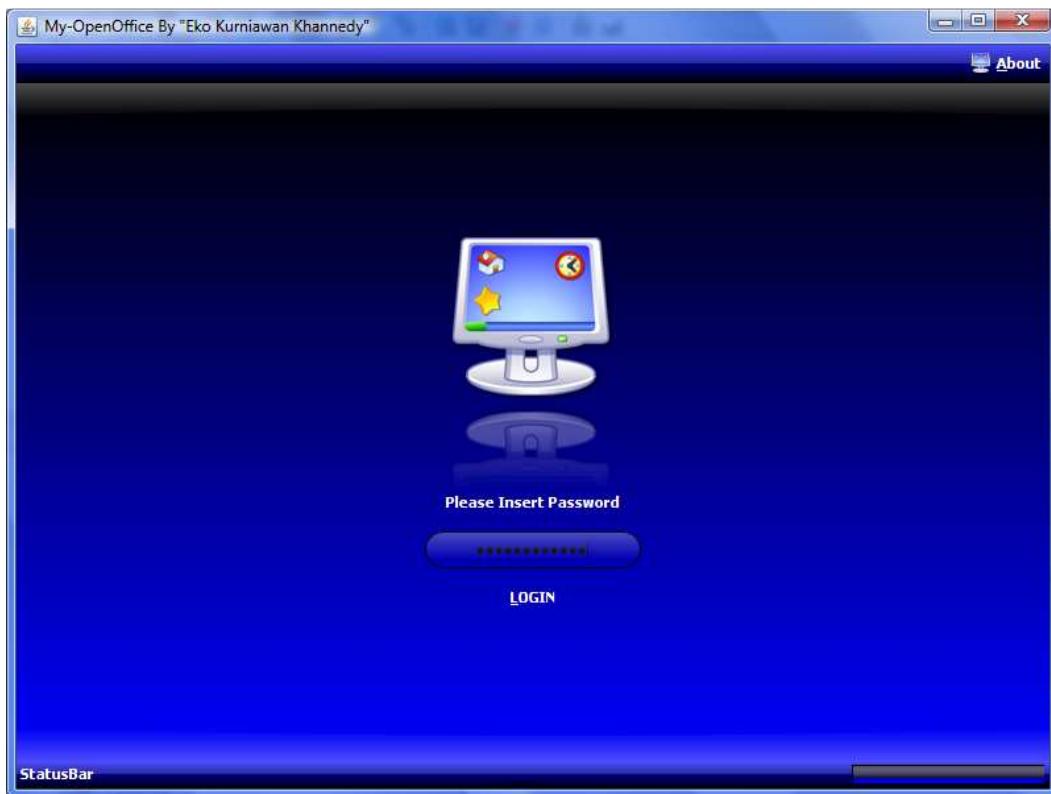


Gambar 37 PanelImageKaca2.java

Kesimpulan

Mungkin anda bertanya – tanya, apakah penting untuk mengetahui cara menggambar dan mewarnai dalam JPanel untuk membuat aplikasi berbasis Swing.

Ya tentu saja!!! Saat ini jarang sekali kita temukan aplikasi yang memiliki tampilan standar seperti jaman ketika kita menggunakan Windows 98, yang sangat kaku. Saat ini kita perlu membuat aplikasi yang selain powerfull tapi juga memiliki daya tarik atau memiliki kemewahan dalam desain, misal saja saya pernah membuat aplikasi MyOpenOffice :



Gambar 38 MyOpenOffice Login



Gambar 39 MyOpenOffice Menu



Gambar 40 MyOpenOffice Contact

Dan untuk membuat tampilan aplikasi seperti diatas, kita harus menguasai teknik menggambar dan mewarnai dalam JPanel.

Sebenarnya selain di JPanel anda juga bisa membuat gambar hampir disemua component Swing dengan mengoveride metode paintComponent() baik itu dalam JButton, JLabel ataupun dalam component – component swing lainnya.

Dari penjelasan tadi diatas, anda sudah bisa membuat JPanel yang memiliki efek – efek yang memukau contohnya :

PanelEfekt.java

```
package pelajaran3;

import java.awt.Color;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
import java.awt.geom.GeneralPath;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
```

```
public class PanelEfect extends JPanel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setBounds(100, 100, 400, 400);
                frame.add(new PanelEfect());
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setVisible(true);
            }
        });
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
        RenderingHints.VALUE_ANTIALIAS_ON);

        GradientPaint paint = new GradientPaint(0, 0, Color.BLACK, 0,
        getHeight(), Color.RED);

        g2.setPaint(paint);
        g2.fillRect(0, 0, getWidth(), getHeight());

        final int width = getWidth();
        final int height = getHeight() * 5 / 100;

        final Color light = new Color(1F, 1F, 1F, 0.5F);
        final Color dark = new Color(1F, 1F, 1F, 0.0F);

        paint = new GradientPaint(0, 0, light, 0, height, dark);
        GeneralPath path = new GeneralPath();
        path.moveTo(0, 0);
        path.lineTo(0, height);
        path.curveTo(0, height, width / 2, height / 2, width, height);
        path.lineTo(width, 0);
        path.closePath();

        g2.setPaint(paint);
        g2.fill(path);

        paint = new GradientPaint(0, getHeight(), light, 0, getHeight() -
        height, dark);
        path = new GeneralPath();
        path.moveTo(0, getHeight());
        path.lineTo(0, getHeight() - height);
        path.curveTo(0, getHeight() - height, width / 2, getHeight() - height
        / 2, width, getHeight()
        - height);
        path.lineTo(width, getHeight());
        path.closePath();

        g2.setPaint(paint);
        g2.fill(path);
    }
}
```



Gambar 41 PanelEfect.java

JButton

Glass

Tombol Kotak Lancip

Salah satu teknik yang dapat membuat JButton terlihat lebih indah adalah efek glass. Dan sekali lagi untuk membuat efek glass kita perlu mengoverride metode paintComponent();

ButtonGlass.java

```
package pelajaran4;

import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
import java.awt.geom.Rectangle2D;
import javax.swing.JButton;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class ButtonGlass extends JButton {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                final JPanel panel = new JPanel(new FlowLayout());
                panel.setBackground(Color.RED);
                frame.add(panel);
                frame.setBounds(100, 100, 300, 200);
                panel.add(new ButtonGlass("TOMBOL"));
                frame.setVisible(true);
            }
        });
    }

    public ButtonGlass(final String title) {
        super(title);
        setFocusPainted(false);
        setBorderPainted(false);
        setContentAreaFilled(false);
    }

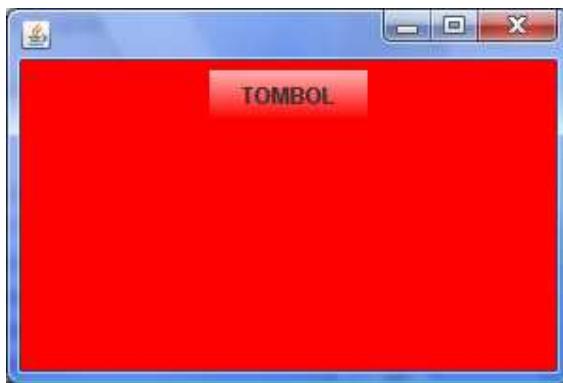
    @Override
    protected void paintComponent(final Graphics g) {
        final Color light = new Color(1F, 1F, 1F, 0.7F);
        final Color dark = new Color(1F, 1F, 1F, 0F);
    }
}
```

```
final GradientPaint paint = new GradientPaint(0, 0, light, 0,
getHeight(), dark);

final Rectangle2D.Double kotak = new Rectangle2D.Double(0, 0,
getWidth(), getHeight());

final Graphics2D g2 = (Graphics2D) g.create();
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
g2.setPaint(paint);
g2.fill(kotak);

super.paintComponent(g);
}
```



Gambar 42 ButtonGlass.java

ButtonGlass2.java

```
package pelajaran4;

import java.awt.Color;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
import java.awt.geom.Rectangle2D;
import javax.swing.JButton;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;
import pelajaran3.PanelEfect;

/**
 * @author usu
 */
public class ButtonGlass2 extends JButton {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
```

```
final Frame frame = new Frame();
frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
final JPanel panel = new PanelEffect();
panel.setLayout(null);
panel.setBackground(Color.RED);
frame.add(panel);
frame.setBounds(100, 100, 300, 200);
final ButtonGlass2 glass = new ButtonGlass2("TOMBOL");
panel.add(glass);
glass.setBounds(100, 100, 100, 25);
frame.setVisible(true);
}
} );
}

public ButtonGlass2(final String title) {
super(title);
setFocusPainted(false);
setBorderPainted(false);
setContentAreaFilled(false);
}

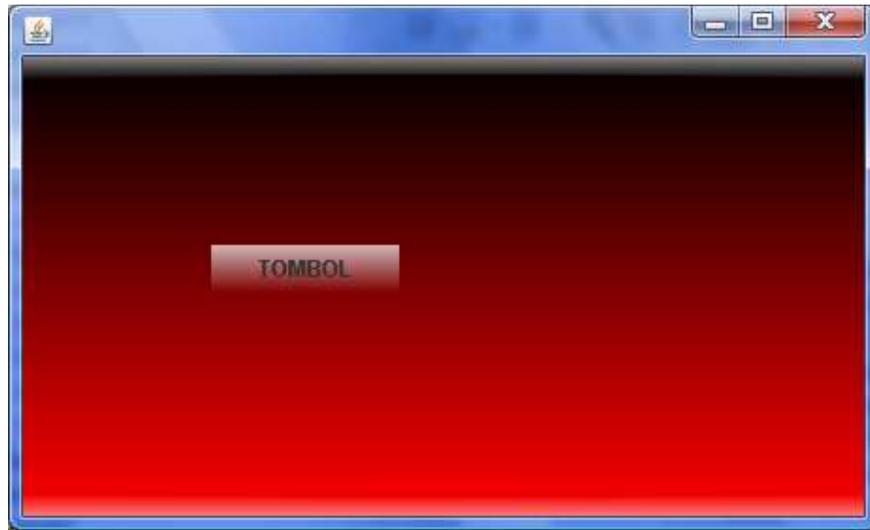
@Override
protected void paintComponent(final Graphics g) {
final Color light = new Color(1F, 1F, 1F, 0.7F);
final Color dark = new Color(1F, 1F, 1F, 0F);

final GradientPaint paint = new GradientPaint(0, 0, light, 0,
getHeight(), dark);

final Rectangle2D.Double kotak = new Rectangle2D.Double(0, 0,
getWidth(),
getHeight());

final Graphics2D g2 = (Graphics2D) g.create();
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
g2.setPaint(paint);
g2.fill(kotak);

super.paintComponent(g);
}
}
```



Gambar 43 ButtonGlass2.java

Satu hal yang harus diingat dalam pembuatan JButton glass adalah merubah atau menghapus gambar JButton sebelumnya dengan cara :

```
setFocusPainted(false);
setBorderPainted(false);
setContentAreaFilled(false);
```

Tombol Kotak Luncul

Kadang menggunakan button berbentuk kotak agak terlihat kaku sehingga untuk membuatnya tidak kaku saya sarankan untuk membuat button berbentuk round :

ButtonGlassRound.java

```
package pelajaran4;

import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
import java.awt.geom.RoundRectangle2D;
import javax.swing.JButton;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class ButtonGlassRound extends JButton {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {
```

```
public void run() {
    final Frame frame = new Frame();
    frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    final JPanel panel = new JPanel(new FlowLayout());
    panel.setBackground(Color.RED);
    frame.add(panel);
    frame.setBounds(100, 100, 300, 200);
    panel.add(new ButtonGlassRound("TOMBOL"));
    frame.setVisible(true);
}
}

public ButtonGlassRound(final String title) {
    super(title);
    setFocusPainted(false);
    setBorderPainted(false);
    setContentAreaFilled(false);
}

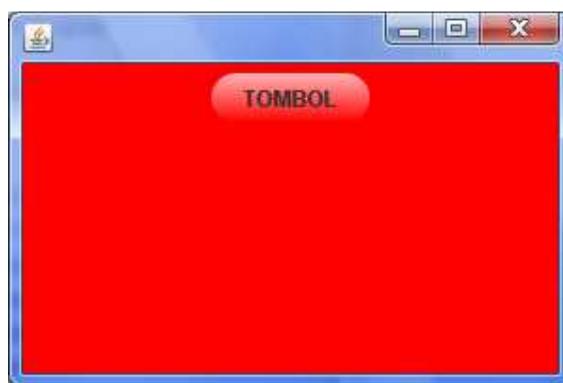
@Override
protected void paintComponent(final Graphics g) {
    final Color light = new Color(1F, 1F, 1F, 0.7F);
    final Color dark = new Color(1F, 1F, 1F, 0F);

    final GradientPaint paint = new GradientPaint(0, 0, light, 0,
getHeight(), dark);

    final RoundRectangle2D.Double kotak = new RoundRectangle2D.Double(0,
0, getWidth(),
getHeight(), getHeight(), getHeight());

    final Graphics2D g2 = (Graphics2D) g.create();
    g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
    g2.setPaint(paint);
    g2.fill(kotak);

    super.paintComponent(g);
}
}
```



Gambar 44 ButtonGlassRound.java

ButtonGlassRound2.java

```
package pelajaran4;

import java.awt.Color;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
import java.awt.geom.RoundRectangle2D;
import javax.swing.JButton;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;
import pelajaran3.PanelEfect;

/**
 * @author usu
 */
public class ButtonGlassRound2 extends JButton {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                final JPanel panel = new PanelEfect();
                panel.setLayout(null);
                panel.setBackground(Color.RED);
                frame.add(panel);
                frame.setBounds(100, 100, 300, 200);
                final ButtonGlassRound2 glass = new
ButtonGlassRound2("TOMBOL");
                panel.add(glass);
                glass.setBounds(100, 100, 100, 25);
                frame.setVisible(true);
            }
        });
    }

    public ButtonGlassRound2(final String title) {
        super(title);
        setFocusPainted(false);
        setBorderPainted(false);
        setContentAreaFilled(false);
    }

    @Override
    protected void paintComponent(final Graphics g) {
        final Color light = new Color(1F, 1F, 1F, 0.7F);
        final Color dark = new Color(1F, 1F, 1F, 0F);

        final GradientPaint paint = new GradientPaint(0, 0, light, 0,
getHeight(), dark);

        final RoundRectangle2D.Double kotak = new RoundRectangle2D.Double(0,
0,
getWidth(), getHeight(), getHeight(), getHeight());
    }
}
```

```
final Graphics2D g2 = (Graphics2D) g.create();
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
    RenderingHints.VALUE_ANTIALIAS_ON);
g2.setPaint(paint);
g2.fill(kotak);

super.paintComponent(g);
}
```



Gambar 45 ButtonGlassRound2.java

Image

Mungkin semuanya sudah tahu kalo untuk menampilkan gambar dalam JButton, kita hanya perlu menggunakan metode setIcon(), namun yang saya maksudkan adalah menampilkan gambar dengan efek kaca seperti pada pelajaran JPanel sebelumnya.

```
BufferedImage image = EfekGambar.getEfekKaca(Image gambar);
Button.setIcon(new ImageIcon(image));
```

ButtonImage.java

```
package pelajaran4;

import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
import java.awt.geom.RoundRectangle2D;
import java.awt.image.BufferedImage;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;
```

```
import pelajaran3.EfekGambar;

/**
 * @author usu
 */
public class ButtonImage extends JButton {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                final JPanel panel = new JPanel(new FlowLayout());
                panel.setBackground(Color.RED);
                frame.add(panel);
                frame.setBounds(100, 100, 300, 200);
                panel.add(new ButtonImage());
                frame.pack();
                frame.setVisible(true);
            }
        });
    }

    public ButtonImage() {
        super();
        setFocusPainted(false);
        setBorderPainted(false);
        setContentAreaFilled(false);
        final ImageIcon icon = new
ImageIcon(getClass().getResource("/pelajaran3/AIM.png"));
        final BufferedImage image = EfekGambar.getEfekKaca(icon.getImage());
        setIcon(new ImageIcon(image));
    }

    @Override
    protected void paintComponent(final Graphics g) {
        final Color light = new Color(1F, 1F, 1F, 0.7F);
        final Color dark = new Color(1F, 1F, 1F, 0F);

        final GradientPaint paint = new GradientPaint(0, 0, light, 0,
getHeight(), dark);

        final RoundRectangle2D.Double kotak = new RoundRectangle2D.Double(0,
0, getWidth(),
getHeight(), 50, 50);

        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
        g2.setPaint(paint);
        g2.fill(kotak);

        super.paintComponent(g);
    }
}
```



Gambar 46 ButtonImage.java

Pada kode diatas, saya membuat gambar di letakan di atas glass, hal ini karena saya memanggil metode super.paintComponent() di urutan paling bawah sehingga gambar dari JButton akan di gambar diatas JButton Glass, kecuali anda membalikkan letaknya :

```
@Override
protected void paintComponent(Graphics g) {
    super.paintComponent(g);

    Color light = new Color(1F, 1F, 1F, 0.7F);
    Color dark = new Color(1F, 1F, 1F, 0F);

    GradientPaint paint = new GradientPaint(0, 0, light, 0, getHeight(), dark);

    RoundRectangle2D.Double kotak = new RoundRectangle2D.Double(0, 0, getWidth(),
        getHeight(), 50, 50);

    Graphics2D g2 = (Graphics2D) g.create();
    g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
        RenderingHints.VALUE_ANTIALIAS_ON);
    g2.setPaint(paint);
    g2.fill(kotak);
}
```



Gambar 47 JButtonImage menggunakan glass diatas

Mouse

Salah satu perbedaan JButton dengan JPanel adalah interaksinya dengan Mouse, ketika mouse ditekan atau ketika mouse diatas JButton biasanya tampilan JButton berubah, sehingga jika kita membuat button seperti button – button sebelumnya, maka akan terlihat kaku.

Over

Salah satu interaksi dengan mouse adalah ketika mouse diatas button (over), dan agar mempermudah kita sewaktu koding, maka kita harus membuat metode yang mengecek apakah mouse ada diatas JButton atau tidak:

```
private boolean over;  
  
private boolean isOver(){  
    return over;  
}  
  
private void setOver(boolean value){  
    over = value;  
}
```

Lalu tambahkan mouse listener :

```
addMouseListener(new MouseAdapter() {  
  
    public void mouseEntered(final MouseEvent e) {  
        setOver(true);  
    }  
  
    public void mouseExited(final MouseEvent e) {  
        setOver(false);  
    }  
});
```

ButtonOver.java

```
package pelajaran4;

import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.geom.RoundRectangle2D;
import java.awt.image.BufferedImage;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;
import pelajaran3.EfekGambar;

/**
 * @author usu
 */
public class ButtonOver extends JButton {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                final JPanel panel = new JPanel(new FlowLayout());
                panel.setBackground(Color.RED);
                frame.add(panel);
                frame.setBounds(100, 100, 300, 200);
                panel.add(new ButtonOver());
                frame.pack();
                frame.setVisible(true);
            }
        });
    }

    private boolean over;

    public ButtonOver() {
        super();
        setFocusPainted(false);
        setBorderPainted(false);
        setContentAreaFilled(false);
        final ImageIcon icon = new
ImageIcon(getClass().getResource("/pelajaran3/AIM.png"));
        final BufferedImage image = EfekGambar.getEfekKaca(icon.getImage());
        setIcon(new ImageIcon(image));
        setOver(false);
        addMouseListener(new MouseAdapter() {

            @Override
```

```
        public void mouseEntered(final MouseEvent e) {
            setOver(true);
        }

        @Override
        public void mouseExited(final MouseEvent e) {
            setOver(false);
        }
    });

public boolean isOver() {
    return this.over;
}

@Override
protected void paintComponent(final Graphics g) {
    final Color light = new Color(1F, 1F, 1F, 0.7F);
    final Color dark = new Color(1F, 1F, 1F, 0F);

    GradientPaint paint = null;

    if (this.over) {
        paint = new GradientPaint(0, 0, dark, 0, getHeight(), light);
    } else {
        paint = new GradientPaint(0, 0, light, 0, getHeight(), dark);
    }

    final RoundRectangle2D.Double kotak = new RoundRectangle2D.Double(0,
0, getWidth(),
getHeight(), 50, 50);

    final Graphics2D g2 = (Graphics2D) g.create();
    g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
    g2.setPaint(paint);
    g2.fill(kotak);

    super.paintComponent(g);
}

public void setOver(final boolean over) {
    this.over = over;
    repaint();
}
}
```

Ketika mouse keluar dari button dan masuk ke button:



Gambar 48 ButtonOver.java saat mouse exit



Gambar 49 ButtonOver.java saat mouse over

Press

Selain interaksi ketika mose over kita juga perlu melakukan interaksi ketika mouse di press (tekan) :

```
private boolean press;
private boolean over;

private boolean isPress(){
    return press;
}

private void setPress(boolean value){
    press = value;
}

private boolean isOver(){
    return over;
}
```

```
private void setOver(boolean value){  
    over = value;  
}
```

Dan jangan lupa menambahkan mouse listener :

```
addMouseListener(new MouseAdapter() {  
  
    public void mouseEntered(final MouseEvent e) {  
        setOver(true);  
    }  
  
    public void mouseExited(final MouseEvent e) {  
        setOver(false);  
    }  
  
    public void mousePressed(final MouseEvent e) {  
        setPress(true);  
    }  
  
    public void mouseReleased(final MouseEvent e) {  
        setPress(false);  
    }  
});
```

ButtonOverPress.java

```
package pelajaran4;  
  
import java.awt.Color;  
import java.awt.FlowLayout;  
import java.awt.GradientPaint;  
import java.awt.Graphics;  
import java.awt.Graphics2D;  
import java.awt.RenderingHints;  
import java.awt.event.MouseAdapter;  
import java.awt.event.MouseEvent;  
import java.awt.geom.RoundRectangle2D;  
import java.awt.image.BufferedImage;  
import javax.swing.ImageIcon;  
import javax.swing.JButton;  
import javax.swing.JPanel;  
import javax.swing.SwingUtilities;  
import javax.swing.WindowConstants;  
import pelajaran1.Frame;  
import pelajaran3.EfekGambar;  
  
/**  
 * @author usu  
 */  
public class ButtonOverPress extends JButton {  
  
    public static void main(final String[] usu) {  
        SwingUtilities.invokeLater(new Runnable() {
```

```
public void run() {
    final Frame frame = new Frame();
    frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    final JPanel panel = new JPanel(new FlowLayout());
    panel.setBackground(Color.RED);
    frame.add(panel);
    frame.setBounds(100, 100, 300, 200);
    panel.add(new ButtonOverPress());
    frame.pack();
    frame.setVisible(true);
}
};

private boolean over;

private boolean press;

public ButtonOverPress() {
    super();
    setFocusPainted(false);
    setBorderPainted(false);
    setContentAreaFilled(false);
    final ImageIcon icon = new
ImageIcon(getClass().getResource("/pelajaran3/AIM.png"));
    final BufferedImage image = EfekGambar.getEfekKaca(icon.getImage());
    setIcon(new ImageIcon(image));
    setOver(false);
    addMouseListener(new MouseAdapter() {

        @Override
        public void mouseEntered(final MouseEvent e) {
            setOver(true);
        }

        @Override
        public void mouseExited(final MouseEvent e) {
            setOver(false);
        }

        @Override
        public void mousePressed(final MouseEvent e) {
            setPress(true);
        }

        @Override
        public void mouseReleased(final MouseEvent e) {
            setPress(false);
        }
    );
}

public boolean isOver() {
    return this.over;
}

public boolean isPress() {
    return this.press;
}
```

```
@Override
protected void paintComponent(final Graphics g) {
    final Color light = new Color(1F, 1F, 1F, 0.7F);
    final Color dark = new Color(1F, 1F, 1F, 0F);

    GradientPaint paint = null;

    if (this.over) {
        if (this.press) {
            paint = new GradientPaint(0, 0, light, 0, getHeight(), light);
        } else {
            paint = new GradientPaint(0, 0, dark, 0, getHeight(), light);
        }
    } else {
        paint = new GradientPaint(0, 0, light, 0, getHeight(), dark);
    }

    final RoundRectangle2D.Double kotak = new RoundRectangle2D.Double(0,
0, getWidth(),
getHeight(), 50, 50);

    final Graphics2D g2 = (Graphics2D) g.create();
    g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
    g2.setPaint(paint);
    g2.fill(kotak);

    super.paintComponent(g);
}

public void setOver(final boolean over) {
    this.over = over;
    repaint();
}

public void setPress(final boolean press) {
    this.press = press;
    repaint();
}
}
```

Ketika mouse diluar button, diatas button dan tak diklik, dan diatas button dan diklik :



Gambar 50 ButtonOverPress.java saat mouse exit



Gambar 51 ButtonOverPress.java saat mouse over



Gambar 52 ButtonOverPress.java saat mouse press

Kesimpulan

Untuk JButton mau gak mau anda harus membuat atau memanipulasi gambar yang dapat berinteraksi dengan Mouse, agar hasil manipulasi JButton anda benar – benar terlihat seperti JButton.

Contoh – contoh diatas hanyalah sebagian kecil dari daya kreativitas anda, jadi berkreasilah untuk membuat Extreem JButton.

JLabel

Drop Shadow Text

Kadang kita hanya membuat JLabel sebagai label biasa, namun sebenarnya kita juga dapat membuat label – label yang luar biasa, contohnya kita akan membuat label dengan drop shadow.

LabelDropShadow.java

```
package pelajaran5;

import java.awt.Color;
import java.awt.Dimension;
import java.awt.FontMetrics;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
import javax.swing.JLabel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class LabelDropShadow extends JLabel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setBounds(100, 100, 300, 200);
                final LabelDropShadow label = new LabelDropShadow();
                label.setText("Eko Kurniawan");
                label.setFont(label.getFont().deriveFont(50F));
                label.setDistance(2);
                frame.add(label);
                frame.setVisible(true);
            }
        });
    }

    private int distance;
    private Color dropShadowColor;

    public LabelDropShadow() {
        super();
        setDistance(3);
        setForeground(Color.GRAY);
        setDropShadowColor(Color.BLACK);
    }

    public int getDistance() {
        return this.distance;
    }
}
```

```
public Color getDropShadowColor() {
    return this.dropShadowColor;
}

@Override
public Dimension getPreferredSize() {
    final Dimension d = super.getPreferredSize();
    d.height += getDistance();
    d.width += getDistance();
    return d;
}

@Override
protected void paintComponent(final Graphics g) {
    final FontMetrics metric = getFontMetrics(getFont());

    final Graphics2D g2 = (Graphics2D) g.create();
    g2.setRenderingHint(RenderingHints.KEY_TEXT_ANTIALIASING,
        RenderingHints.VALUE_TEXT_ANTIALIAS_ON);
    g2.setFont(getFont());
    g2.setColor(getDropShadowColor());
    g2.drawString(getText(), getDistance(), metric.getHeight() +
getDistance());
    g2.setColor(getForeground());
    g2.drawString(getText(), 0, metric.getHeight());
}

public void setDistance(final int distance) {
    this.distance = distance;
}

public void setDropShadowColor(final Color dropShadowColor) {
    this.dropShadowColor = dropShadowColor;
}
}
```



Gambar 53 LabelDropShadow.java

Kesimpulannya untuk membuat DropShadow, kita hanya perlu menggambar teks dua kali dengan jarak bayangan sesuai distance yang telah ditentukan. Dan untuk menggambar teks kita hanya perlu menggunakan metode `drawString()` pada class `Graphics2D` seperti yang terlihat dibawah ini :

```
@Override
protected void paintComponent(Graphics g) {
    FontMetrics metric = getFontMetrics(getFont());
```

```
Graphics2D g2 = (Graphics2D) g.create();
g2.setRenderingHint(RenderingHints.KEY_TEXT_ANTIALIASING,
RenderingHints.VALUE_TEXT_ANTIALIAS_ON);
g2.setFont(getFont());
g2.setColor(getDropShadowColor());
g2.drawString(getText(), getDistance(), metric.getHeight() + getDistance());
g2.setForeground();
g2.drawString(getText(), 0, metric.getHeight());
}
```

Clip Text

Sekarang kita akan membuat JLabel yang digunakan sebagai label clip gambar.

LabelClip.java

```
package pelajaran5;

import java.awt.FontMetrics;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.Shape;
import java.awt.font.FontRenderContext;
import java.awt.font.TextLayout;
import java.awt.geom.AffineTransform;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class LabelClip extends JLabel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setSize(400, 400);
                final LabelClip clip = new LabelClip("EKO KURNIAWAN");
                clip.setFont(clip.getFont().deriveFont(50F));
                clip
                    .setImage(new
ImageIcon(getClass().getResource("/pelajaran9/image.jpg"))
                    .getImage());
                frame.add(clip);
                frame.pack();
                frame.setVisible(true);
            }
        });
    }
}
```

```
private Image image;

public LabelClip(final String text) {
    super(text);
}

public Image getImage() {
    return this.image;
}

@Override
protected void paintComponent(final Graphics g) {
    if (getImage() != null) {
        final FontMetrics metric = getFontMetrics(getFont());

        final Graphics2D g2 = (Graphics2D) g.create();

        final FontRenderContext context = g2.getFontRenderContext();
        final TextLayout layout = new TextLayout(getText(), getFont(),
context);
        final AffineTransform tarnsform =
AffineTransform.getTranslateInstance(0, 50);
        final Shape outline = layout.getOutline(tarnsform);
        g2.setClip(outline);
        for (int i = 0; i < getWidth(); i += getImage().getWidth(null)) {
            for (int j = 0; j < getHeight(); j +=
getImage().getHeight(null)) {
                g2.drawImage(getImage(), 0, 0, null);
            }
        }
    } else {
        super.paintComponent(g);
    }
}

public void setImage(final Image image) {
    this.image = image;
    repaint();
}
}
```



Gambar 54 LabelClip.java

So yang pasti untuk membuat clip tinggal menggunakan metode setClip(Shape) milik Graphics2D.

Gradient Text

Kali ini kita akan mencoba membuat JLabel yang berwarna gradient, sehingga teks yang ditampilkan oleh JLabel lebih menarik untuk dilihat dibandingkan dengan warna solid (satu warna).

LabelGradient.java

```
package pelajaran5;

import java.awt.Color;
import java.awt.FontMetrics;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import javax.swing.JLabel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class LabelGradient extends JLabel {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setBounds(100, 100, 300, 200);
                final LabelGradient label = new LabelGradient();
                label.setText("Eko Kurniawan");
                label.setFont(label.getFont().deriveFont(50F));
                frame.add(label);
                frame.pack();
                frame.setVisible(true);
            }
        });
    }

    @Override
    protected void paintComponent(final Graphics g) {
        final FontMetrics metric = getFontMetrics(getFont());

        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setPaint(new GradientPaint(0, 0, Color.RED, getWidth(), 0,
Color.BLUE));
        g2.drawString(getText(), 0, metric.getHeight() -
metric.getDescent());
    }
}
```



Gambar 55 LabelGradient.java

Sama seperti pada component – component lainnya, untuk membuat teks yang berwarna gradient, gunakan setPaint(Paint).

JTextField

Shadow Text

Maksud dari shadow text, adalah semacam tooltip yang tertulis dalam JTextField itu sendiri seperti contoh dibawah ini :

Change your password



Gambar 56 Contoh Shadow Text

Jika user belum memasukkan teks maka JTextField akan menampilkan teks berupa tooltip atau perintah.

TextFieldShadowText.java

```
package pelajaran6;

import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.event.FocusAdapter;
import java.awt.event.FocusEvent;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JButton;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TextFieldShadowText extends JTextField {

    public static void main(final String[] usu) {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
            SwingUtilities.invokeLater(new Runnable() {

                public void run() {
                    final Frame frame = new Frame();
                    frame.setLayout(new FlowLayout());
                }
            });
        }
    }
}
```

```
        final JTextFieldShadowText text = new JTextFieldShadowText();
        text.setColumns(30);
        text.setShadowText("SHADOW TEXT");
        frame.add(text);

frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.add(new JButton("TOMBOL"));
        frame.pack();
        frame.setVisible(true);
    }
}
} catch (final ClassNotFoundException ex) {

Logger.getLogger(TextFieldShadowText.class.getName()).log(Level.SEVERE,
null, ex);
} catch (final InstantiationException ex) {

Logger.getLogger(TextFieldShadowText.class.getName()).log(Level.SEVERE,
null, ex);
} catch (final IllegalAccessException ex) {

Logger.getLogger(TextFieldShadowText.class.getName()).log(Level.SEVERE,
null, ex);
} catch (final UnsupportedLookAndFeelException ex) {

Logger.getLogger(TextFieldShadowText.class.getName()).log(Level.SEVERE,
null, ex);
}

private Color defaultColorText;
private boolean shadow;
private String shadowText;

private Color shadowTextColor;

public JTextFieldShadowText() {
    super();
    this.defaultColorText = getForeground();
    setShadow(true);
    setDefaultColorText(getForeground());
    setShadowTextColor(Color.GRAY);
    addFocusListener(new FocusAdapter() {

        @Override
        public void focusGained(final FocusEvent e) {
            if (isShadow()) {
                setText("");
                setForeground(defaultColorText);
            }
        }

        @Override
        public void focusLost(final FocusEvent e) {
            setShadow(false);
            if (getText().length() < 1) {
                setForeground(shadowTextColor);
                setShadow(true);
                setText(shadowText);
            } else {
                setShadow(false);
            }
        }
    });
}
```

```
        setForeground(getDefaultColorText());
    }
}
} );
}

public Color getDefaultColorText() {
    return this.defaultColorText;
}

public String getShadowText() {
    return this.shadowText;
}

public Color getShadowTextColor() {
    return this.shadowTextColor;
}

@Override
public String getText() {
    if (isShadow()) {
        return "";
    }
    return super.getText();
}

public boolean isShadow() {
    return this.shadow;
}

public void setDefaultColorText(final Color defaultColorText) {
    this.defaultColorText = defaultColorText;
}

@Override
@Deprecated
public void setForeground(final Color fg) {
    super.setForeground(fg);
}

public void setShadow(final boolean shadow) {
    this.shadow = shadow;
}

public void setShadowText(final String shadowText) {
    this.shadowText = shadowText;
}

public void setShadowTextColor(final Color shadowTextColor) {
    this.shadowTextColor = shadowTextColor;
}
}
```

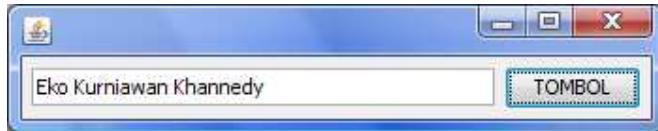
Saat JTextField berum terisi teks, maka akan ada tulisan “SHADOW TEXT”, anda bisa merubahnya dengan menggunakan :

```
text.setShadowText("SHADOW TEXT");
```



Gambar 57 TextFieldShadowText.java saat shadow text terlihat

Dan jika JTextField telah diisi teks, maka tulisan “SHADOW TEXT” tak akan muncul.



Gambar 58 TextFieldShadowText.java saat user menginputkan teks

Auto Complete Text

Kadang kita sering menemui sebuah TextField yang menggunakan auto complete text, sehingga ketika kita mengetikkan sebagian teks, maka secara otomatis TextField akan menampilkan teks yang menyerupai teks yang kita inputkan.

Buatlah model untuk autocomplete

AutoComplete.java

```
package pelajaran6;

/**
 * @author usu
 */
public interface AutoComplete {
    public String get.AutoComplete(String key);
}
```

Buat default model untuk autocomplete

DefaultAutoComplete.java

```
package pelajaran6;

import java.util.ArrayList;
import java.util.Arrays;

/**
 * @author usu
 */
public class DefaultAutoComplete implements AutoComplete {
    private final ArrayList<String> data;
    public DefaultAutoComplete(final String[] data) {
        this.data = new ArrayList<String>(Arrays.asList(data));
    }
}
```

```
public boolean add(final String e) {
    return this.data.add(e);
}

public String getAutoComplete(final String key) {
    for (final String i : this.data) {
        if (i.startsWith(key)) {
            return i;
        }
    }
    return null;
}

public boolean remove(final Object o) {
    return this.data.remove(o);
}
}
```

TextFieldAutoComplete.java

```
package pelajaran6;

import java.awt.FlowLayout;
import java.awt.GraphicsEnvironment;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.text.BadLocationException;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TextFieldAutoComplete extends JTextField {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();

                final TextFieldAutoComplete text = new TextFieldAutoComplete();
                text.setColumns(30);

                final String[] data =
GraphicsEnvironment.getLocalGraphicsEnvironment()
                    .getAvailableFontFamilyNames();
                final DefaultAutoComplete autoComplete = new
DefaultAutoComplete(data);

                text.setAutoComplete(autoComplete);

                frame.setLayout(new FlowLayout());
                frame.add(text);
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.pack();
            }
        });
    }
}
```

```
        frame.setVisible(true);
    }
}
}

private AutoComplete autoComplete;

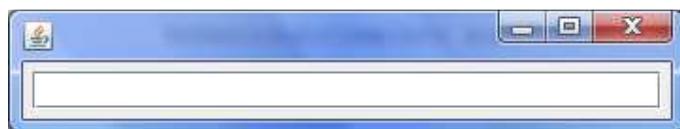
public TextFieldAutoComplete() {
    super();
    addActionListener(new ActionListener() {

        public void actionPerformed(final ActionEvent e) {
            setCaretPosition(getDocument().getLength());
        }
    });
}

public AutoComplete getAutoComplete() {
    return this.autoComplete;
}

@Override
public void replaceSelection(final String content) {
    if (getAutoComplete() == null) {
        super.replaceSelection(content);
    } else {
        try {
            super.replaceSelection(content);
            final String full = getDocument().getText(0,
getDocument().getLength());
            if (getAutoComplete().getAutoComplete(full) != null) {
                setText(getAutoComplete().getAutoComplete(full));
                setCaretPosition(getDocument().getLength());
                moveCaretPosition(full.length());
            }
        } catch (final BadLocationException ex) {
            // Error
        }
    }
}

public void setAutoComplete(final AutoComplete autoComplete) {
    this.autoComplete = autoComplete;
}
}
```



Gambar 59 TextFieldAutoComplete.java



Gambar 60 TextFieldAutoComplete.java saat user menginputkan teks

Ketika user mengetikkan “Tim”, maka secara otomatis akan keluar text auto complete “Times New Roman” yang merupakan kata terdekat dengan kata “Tim”.

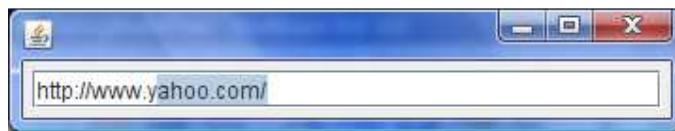
Kesimpulannya terdapat pada setAutoComplete();

Misal kita ingin membuat text auto complete untuk alamat website :

```
TextFieldAutoComplete text = new TextFieldAutoComplete();
text.setColumns(30);

String[] data = new String[]{
    "http://www.google.com/",
    "http://www.yahoo.com/",
    "http://www.netbeans.org/",
    "http://www.eclipse.org"
};
DefaultAutoComplete autoComplete = new DefaultAutoComplete(data);

text.setAutoComplete(autoComplete);
```



Gambar 61 TextFieldAutoComplete untuk website

Manipulasi Background

Mungkin kita sering merubah latar JTextField dengan menggunakan metode setBackground(Color), namun itu hanya bisa merubah warna solid. Bagaimana jika kita ingin menggunakan warna background gradien atau gambar.

Gradients

Kita telah belajar untuk membuat berbagai macam gradien tadi dalam pelajaran JPanel sehingga pada pelajaran ini kita bisa langsung praktekan.

```
@Override
protected void paintComponent(Graphics g) {
    if (isOpaque()) {
        setOpaque(false);
    }

    // lakukan menggambar background disini
    super.paintComponent(g);
}
```

Ada sedikit trik untuk menampilkan gambar yang telah kita buat yaitu dengan menggunakan :

```
if (isOpaque()) {
    setOpaque(false);
}
```

Mengapa hal ini diperlukan, karena jika tidak maka gambar yang telah kita buat akan hilang. Sebenarnya kita juga bisa tanpa menggunakan trik tersebut, namun dengan cara membalikkan kodennya :

```
@Override
protected void paintComponent(Graphics g) {
    super.paintComponent(g);

    // lakukan menggambar background disini

}
```

Namun jika anda menggunakan kode ini, anda harus menanggung resiko kalo border tak akan ditampilkan dan teks tak akan ditampilkan karena gambarnya tertimpa oleh gambar yang kita buat, jadi saya menyarankan untuk menggunakan teknik pertama.

Sekarang misal kita akan membuat JTextField dengan background gradien :

TextFieldBackground.java

```
package pelajaran6;

import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.border.LineBorder;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TextFieldBackground extends JTextField {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setLayout(new FlowLayout());
                final TextFieldBackground text = new TextFieldBackground();
                text.setColumns(30);
                text.setForeground(Color.white);
                text.setBorder(new LineBorder(Color.BLACK, 1));
                frame.add(text);
                frame.pack();
            }
        });
    }
}
```

```
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}

@Override
protected void paintComponent(final Graphics g) {
    if (isOpaque()) {
        setOpaque(false);
    }

    final GradientPaint paint = new GradientPaint(0, 0, Color.RED,
getWidth(), 0, Color.BLUE);

    final Graphics2D g2 = (Graphics2D) g.create();
    g2.setPaint(paint);
    g2.fillRect(0, 0, getWidth(), getHeight());

    super.paintComponent(g);
}
}
```



Gambar 62 TextFieldBackground

Image

Selain menggunakan background gradient, kita juga bisa menggunakan background gambar yang dapat kita tentukan sendiri. Dan untuk membuat background gambar, caranya mirip dengan menggambar di JPanel, namun dengan sedikit trik looping dalam proses penggambaran.

TextFieldImage.java

```
package pelajaran6;

import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Graphics;
import java.awt.Image;
import javax.swing.ImageIcon;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.border.LineBorder;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TextFieldImage extends JTextField {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

```

```
public void run() {
    final Frame frame = new Frame();
    frame.setLayout(new FlowLayout());

    final TextFieldImage text = new TextFieldImage();
    text.setColumns(30);
    text.setBorder(new LineBorder(Color.BLACK, 1));

    final Image gambar = new ImageIcon(TextFieldImage.class
        .getResource("/pelajaran6/Bunga.jpg")).getImage();
    text.setGambar(gambar);

    frame.add(text);
    frame.pack();
    frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    frame.setVisible(true);
}
}

private Image gambar;

public TextFieldImage() {
    super();
}

public Image getGambar() {
    return this.gambar;
}

@Override
protected void paintComponent(final Graphics g) {
    if (isOpaque()) {
        setOpaque(false);
    }

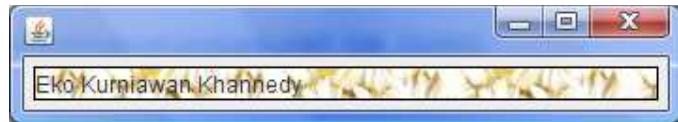
    if (getGambar() != null) {
        for (int i = 0; i <= getWidth(); i += getGambar().getWidth(null))
{
            for (int j = 0; j < getHeight(); j +=
getGambar().getHeight(null)) {
                g.drawImage(getGambar(), i, j, null);
            }
        }
    }

    super.paintComponent(g);
}

public void setGambar(final Image gambar) {
    this.gambar = gambar;
    repaint();
}
}
```



Gambar 63 TextFieldImage.java



Gambar 64 TextFieldImage.java saat user menginputkan teks

Mengapa dalam teknik penggambaran kita memerlukan trik seperti ini :

```
if (getGambar() != null) {
    for (int i = 0; i <= getWidth(); i += getGambar().getWidth(null)) {
        for (int j = 0; j < getHeight(); j += getGambar().getHeight(null)) {
            g.drawImage(getGambar(), i, j, null);
        }
    }
}
```

Alasannya adalah karena gambar yang kita jadikan background belum tentu luasnya sesuai dengan luas JTextField sehingga kita menggunakan trik looping agar seluruh area JTextField dapat terisi penuh dengan gambar.

Validasi Input

Dalam Swing kita mengenal JFormattedTextField yang berguna sebagai textbox yang memiliki kemampuan untuk memformat teks yang diinputkan oleh user. Tapi selain itu JTextField pun dapat kita gunakan untuk memvalidasi text yang diinputkan.

Sebenarnya bukan JTextFieldnya yang kita gunakan untuk memvalidasi, tapi Document milik JTextFieldnya. Jika anda tidak tau apa itu Document, Document adalah class yang digunakan untuk menampung data teks yang dimiliki oleh JtextComponent.

Buat document baru :

DocumentFormated.java

```
package pelajaran6;

import javax.swing.text.AttributeSet;
import javax.swing.text.BadLocationException;
import javax.swing.text.PlainDocument;

/**
 * @author usu
 */
public class DocumentFormated extends PlainDocument {

    public DocumentFormated() {
```

```
super();
}

@Override
public void insertString(final int offs, final String str, final
AttributeSet a)
    throws BadLocationException {

    final char[] array = str.toCharArray();
    boolean valid = false;

    for (final char c : array) {
        if (Character.isDigit(c)) {
            valid = true;
        } else {
            valid = false;
            break;
        }
    }

    if (valid) {
        super.insertString(offs, str, a);
    } else {
        // DI BLOKIR
    }
}
}
```

Misal kita akan memvalidasi teks yang berupa nomor, sehingga jika bukan nomor maka input teks atau diblokir.

```
@Override
public void insertString(int offs, String str, AttributeSet a) throws BadLocationException {

    char[] array = str.toCharArray();
    boolean valid = false;

    for (char c : array) {
        if (Character.isDigit(c)) {
            valid = true;
        } else {
            valid = false;
            break;
        }
    }

    if(valid){
        super.insertString(offs, str, a);
    }else{
        // DI BLOKIR
    }
}
}
```

```
package pelajaran6;

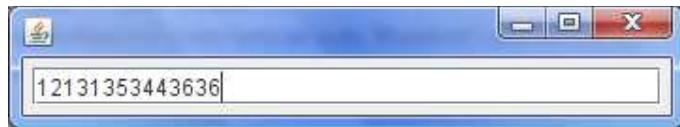
import java.awt.FlowLayout;
import javax.swing.JTextField;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class DocumentFormatedApp {

    public static void main(final String[] usu) {
        final Frame frame = new Frame();
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.setLayout(new FlowLayout());

        final JTextField text = new JTextField();
        text.setColumns(30);
        text.setDocument(new DocumentFormated());
        frame.add(text);

        frame.pack();
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }
}
```



Gambar 65 DocumentFormatedApp.java

Sekarang jika anda menginputkan teks yang selain nomber maka JTextField tak akan menerima input tersebut.

Kesimpulan

Sebenarnya saya bingung kalo setiap pelajaran harus menarik kesimpulan, karena menurut saya kesimpulan setiap orang itu berbeda – beda, jadi yang pasti anda bisa simpulkan sendiri dari pelajaran diatas.

Dan untuk memancing daya kreasi anda saya lampirkan contoh JTextField yang dapat saya buat dari pelajaran – pelajaran diatas.

TextFieldStyle.java

```
package pelajaran6;

import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
```

```
import java.awt.geom.RoundRectangle2D;
import javax.swing.JTextField;
import javax.swing.SwingConstants;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.border.EmptyBorder;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TextStyle extends JTextField {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setLayout(new FlowLayout());

                final TextStyle text = new TextStyle();
                text.setColumns(30);
                text.setHorizontalAlignment(SwingConstants.CENTER);
                frame.add(text);

                frame.pack();
                frame.setLocationRelativeTo(null);
                frame.setVisible(true);
            }
        });
    }

    public TextStyle() {
        super();
        setBorder(new EmptyBorder(3, 3, 3, 3));
    }

    @Override
    protected void paintComponent(final Graphics g) {

        if (isOpaque()) {
            setOpaque(false);
        }

        final RoundRectangle2D.Double round = new RoundRectangle2D.Double(0,
0, getWidth(),
getHeight(), getHeight(), getHeight());

        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);

        GradientPaint paint = new GradientPaint(0, 0, Color.black, 0,
getHeight(), Color.red);

        g2.setPaint(paint);
        g2.fill(round);

        final Color dark = new Color(1F, 1F, 1F, 0F);
        final Color light = new Color(1F, 1F, 1F, 0.5F);
    }
}
```

```
paint = new GradientPaint(0, 0, light, 0, getHeight(), dark);
g2.setPaint(paint);
g2.fill(round);

g2.setColor(Color.BLACK);
g2.draw(round);

super.paintComponent(g);

}
}
```



Gambar 66 TextStyle.java

TextFieldStyle2.java

```
package pelajaran6;

import java.awt.Color;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Point;
import java.awt.Rectangle;
import java.awt.RenderingHints;
import java.awt.geom.RoundRectangle2D;
import javax.swing.JTextField;
import javax.swing.SwingConstants;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.border.EmptyBorder;
import pelajaran1.Frame;
import pelajaran3.PanelEfekt;

/**
 * @author usu
 */
public class TextStyle2 extends JTextField {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setLayout(new FlowLayout());

                final TextStyle2 text = new TextStyle2();
                text.setColumns(30);
                text.setHorizontalAlignment(SwingConstants.CENTER);
                frame.setSize(400, 300);
            }
        });
    }
}
```

```
frame.setContentPane(new PanelEffect());
frame.getContentPane().setLayout(null);
frame.getContentPane().add(text);

final Dimension dim = frame.getSize();
final Dimension dim2 = new Dimension(200, 25);
final Rectangle r = new Rectangle();
r.setLocation(new Point((dim.width - dim2.width) / 2,
(dim.height) / 2 - dim2.height));
r.setSize(dim2);

text.setBounds(r);

frame.setLocationRelativeTo(null);
frame.setVisible(true);
}
} );
}

public TextFieldStyle2() {
super();
setBorder(new EmptyBorder(3, 3, 3, 3));
}

@Override
protected void paintComponent(final Graphics g) {

if (isOpaque()) {
setOpaque(false);
}

final RoundRectangle2D.Double round = new RoundRectangle2D.Double(0,
0, getWidth(), getHeight(), getHeight(), getHeight());

final Graphics2D g2 = (Graphics2D) g.create();
g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);

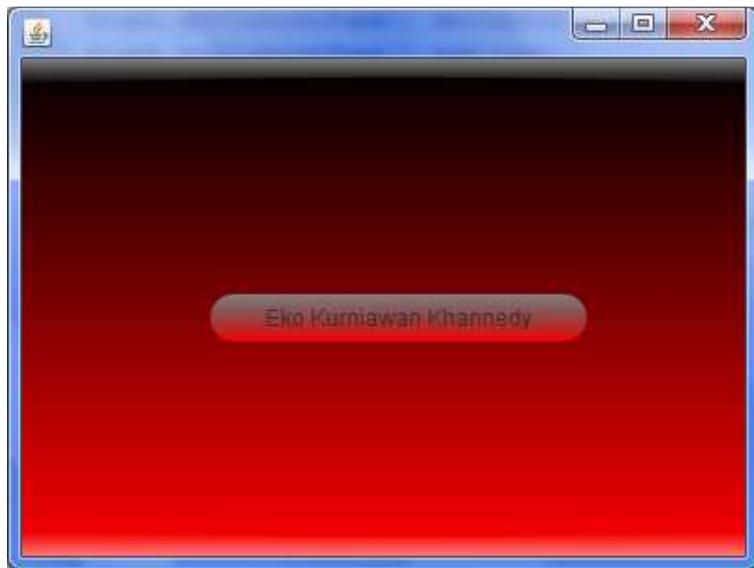
GradientPaint paint = new GradientPaint(0, 0, Color.black, 0,
getHeight(), Color.red);

g2.setPaint(paint);
g2.fill(round);

final Color dark = new Color(1F, 1F, 1F, 0F);
final Color light = new Color(1F, 1F, 1F, 0.5F);

paint = new GradientPaint(0, 0, light, 0, getHeight(), dark);
g2.setPaint(paint);
g2.fill(round);

super.paintComponent(g);
}
}
```



Gambar 67 TextStyle2.java

Pelajaran 7

JComboBox

Render

Salah satu keajaiban swing adalah fasilitas render yang dimiliki JComboBox, JTree, JList dan Jtable. Dengan adanya fasilitas Render, kita bisa memanipulasi tampilan dari component – component tersebut.

Background

Kali ini kita akan memanipulasi latar combo berdasarkan genap atau ganjil. Untuk membuat hal tersebut kita harus membuat CellRender sendiri. Untuk menset render ke JComboBox gunakan kode dibawah :

```
JComboBox combo = new JComboBox();
combo.setRenderer(ListCellRenderer render);
```

ColorRender.java

```
package pelajaran7;

import java.awt.Color;
import java.awt.Component;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.ListCellRenderer;

/**
 * @author usu
 */
public class ColorRender implements ListCellRenderer {

    public Component getListCellRendererComponent(final JList list, final
Object value,
        final int index, final boolean isSelected, final boolean
cellHasFocus) {

        final JLabel label = new JLabel(value.toString());
        label.setOpaque(true);

        if (isSelected) {
            label.setForeground(list.getSelectionForeground());
            label.setBackground(list.getSelectionBackground());
        } else {
            label.setForeground(list.getForeground());
            if (index % 2 == 0) {
                label.setBackground(Color.GREEN);
            } else {
                label.setBackground(Color.YELLOW);
            }
        }

        return label;
    }
}
```

```
    }
}
```

Sekarang tinggal menggunakan cell render tersebut ke JComboBox.

ComboColor.java

```
package pelajaran7;

import java.awt.FlowLayout;
import java.awt.GraphicsEnvironment;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JComboBox;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class ComboColor {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setLayout(new FlowLayout());

                final String[] data =
                    GraphicsEnvironment.getLocalGraphicsEnvironment()
                        .getAvailableFontFamilyNames();
                final DefaultComboBoxModel model = new
                    DefaultComboBoxModel(data);

                final JComboBox combo = new JComboBox();
                combo.setModel(model);
                combo.setRenderer(new ColorRender());

                frame.add(combo);
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.pack();
                frame.setVisible(true);
            }
        });
    }
}
```



Gambar 68 ComboColor.java

ColorRenderRandom.java

```
package pelajaran7;

import java.awt.Color;
import java.awt.Component;
import java.util.Random;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.ListCellRenderer;

/**
 * @author usu
 */
public class ColorRenderRandom implements ListCellRenderer {

    private final Random generator = new Random();

    public Component getListCellRendererComponent(final JList list,
                                                final Object value, final int index, final boolean isSelected,
                                                final boolean cellHasFocus) {
        final JLabel label = new JLabel(value.toString());
        label.setOpaque(true);

        if (isSelected) {
            label.setForeground(list.getSelectionForeground());
            label.setBackground(list.getSelectionBackground());
        } else {
            label.setForeground(list.getForeground());
            label.setBackground(new Color(this.generator.nextInt(255),
                                         this.generator.nextInt(255), this.generator.nextInt(255)));
        }

        return label;
    }
}
```

ComboColor2.java

```
package pelajaran7;
```

```
import java.awt.FlowLayout;
import java.awt.GraphicsEnvironment;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JComboBox;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class ComboColor2 {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setLayout(new FlowLayout());

                final String[] data = GraphicsEnvironment
                        .getLocalGraphicsEnvironment().getAvailableFontFamilyNames();
                final DefaultComboBoxModel model = new
                        DefaultComboBoxModel(data);

                final JComboBox combo = new JComboBox();
                combo.setModel(model);
                combo.setRenderer(new ColorRenderRandom());

                frame.add(combo);
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.pack();
                frame.setVisible(true);
            }
        });
    }
}
```



Gambar 69 ComboColor2.java

Image

Misal kita ingin membuat atau menampilkan gambar di JComboBox, bisakah? Tentu saja bisa. Sama caranya dengan sebelumnya hanya dengan sedikit trik.

ImageRender.java

```
package pelajaran7;

import java.awt.Component;
import java.io.File;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.ListCellRenderer;

/**
 * @author usu
 */
public class ImageRender implements ListCellRenderer {

    public Component getListCellRendererComponent(final JList list, final
Object value,
        final int index, final boolean isSelected, final boolean
cellHasFocus) {
        final File file = (File) value;

        final JLabel label = new JLabel(file.getName());
        label.setOpaque(true);
        label.setIcon(new ImageIcon(file.getPath()));

        if (isSelected) {
            label.setForeground(list.getSelectionForeground());
            label.setBackground(list.getSelectionBackground());
        } else {
            label.setForeground(list.getForeground());
            label.setBackground(list.getBackground());
        }

        return label;
    }
}
```

Sekarang tinggal buat JComboBoxnya

ComboImage.java

```
package pelajaran7;

import java.awt.FlowLayout;
import java.io.File;
import java.io.FilenameFilter;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JComboBox;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class ComboImage {

    public static void main(final String[] usu) {
```

```
SwingUtilities.invokeLater(new Runnable() {

    public void run() {

        final File[] files = new File("smiles").listFiles(new
FilenameFilter() {

            public boolean accept(File dir, String name) {
                if (name.toUpperCase().endsWith(".PNG")) {
                    return true;
                }
                return false;
            }
        });
        for (final File f : files) {
            System.out.println(f.getName());
        }

        final DefaultComboBoxModel model = new
DefaultComboBoxModel(files);

        final JComboBox combo = new JComboBox(model);
        combo.setRenderer(new ImageRender());
        combo.setMaximumRowCount(3);

        final Frame frame = new Frame();
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.setLayout(new FlowLayout());
        frame.add(combo);
        frame.pack();
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }
});
```



Gambar 70 ComboImage.java

Kesimpulan

Salah satu yang patut ditonjolkan dalam JComboBox adalah Render, anda bisa berkreasi dengan adanya fasilitas render tersebut. Misalnya anda ingin menampilkan gradient background seperti ini :

GradienRender.java

```
package pelajaran7;

import java.awt.Color;
import java.awt.Component;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.io.File;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.ListCellRenderer;

/**
 * @author usu
 */
public class GradienRender implements ListCellRenderer {

    public Component getListCellRendererComponent(final JList list, final
Object value,
        final int index, final boolean isSelected, final boolean
cellHasFocus) {

        final File file = (File) value;

        final JLabel label1 = new JLabel(file.getName()) {
            @Override
```

```
protected void paintComponent(Graphics g) {
    GradientPaint paint = new GradientPaint(0, 0, Color.GREEN, 0,
getHeight(), Color.YELLOW);

    Graphics2D g2 = (Graphics2D) g.create();
    g2.setPaint(paint);
    g2.fillRect(0, 0, getWidth(), getHeight());

    super.paintComponent(g);
}

final JLabel label2 = new JLabel(file.getName()) {

    @Override
    protected void paintComponent(Graphics g) {
        GradientPaint paint = new GradientPaint(0, 0, Color.YELLOW, 0,
getHeight(), Color.GREEN);

        Graphics2D g2 = (Graphics2D) g.create();
        g2.setPaint(paint);
        g2.fillRect(0, 0, getWidth(), getHeight());

        super.paintComponent(g);
    }
};

if (index % 2 == 0) {
    label1.setIcon(new ImageIcon(file.getPath()));
    return label1;
} else {
    label2.setIcon(new ImageIcon(file.getPath()));
    return label2;
}
}
```

Sekarang tinggal menggunakannya dalam JComboBox

ComboGradient.java

```
package pelajaran7;

import java.awt.FlowLayout;
import java.io.File;
import java.io.FilenameFilter;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JComboBox;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class ComboGradient {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {
```

```
public void run() {

    final File[] files = new File("smiles").listFiles(new
FilenameFilter() {

        public boolean accept(File dir, String name) {
            if (name.toUpperCase().endsWith(".PNG")) {
                return true;
            }
            return false;
        }
    });

    for (final File f : files) {
        System.out.println(f.getName());
    }

    final DefaultComboBoxModel model = new
DefaultComboBoxModel(files);

    final JComboBox combo = new JComboBox(model);
    combo.setRenderer(new GradienRender());
    combo.setMaximumRowCount(3);

    final Frame frame = new Frame();
    frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    frame.setLayout(new FlowLayout());
    frame.add(combo);
    frame.pack();
    frame.setLocationRelativeTo(null);
    frame.setVisible(true);
}
}
}
```



Gambar 71 ComboGradient.java

GradienRender2.java

```
package pelajaran7;

import java.awt.Color;
import java.awt.Component;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.io.File;
import java.util.Random;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.ListCellRenderer;

/**
 * @author usu
 */
public class GradienRender2 implements ListCellRenderer {

    private final Random r = new Random();

    public Component getListCellRendererComponent(final JList list,
        final Object value, final int index, final boolean isSelected,
        final boolean cellHasFocus) {

        final File file = (File) value;

        final Color c = new Color(this.r.nextInt(255), this.r.nextInt(255),
            this.r.nextInt(255));
        final Color c2 = new Color(this.r.nextInt(255), this.r.nextInt(255),
            this.r.nextInt(255));

        final JLabel label1 = new JLabel(file.getName()) {
```

```
    @Override
    protected void paintComponent(Graphics g) {
        GradientPaint paint = new GradientPaint(0, 0, c, 0,
getHeight(), c2);

        Graphics2D g2 = (Graphics2D) g.create();
        g2.setPaint(paint);
        g2.fillRect(0, 0, getWidth(), getHeight());

        super.paintComponent(g);
    }
};

final JLabel label2 = new JLabel(file.getName()) {

    @Override
    protected void paintComponent(Graphics g) {
        GradientPaint paint = new GradientPaint(0, 0, c2, 0,
getHeight(), c);

        Graphics2D g2 = (Graphics2D) g.create();
        g2.setPaint(paint);
        g2.fillRect(0, 0, getWidth(), getHeight());

        super.paintComponent(g);
    }
};

if (index % 2 == 0) {
    label1.setIcon(new ImageIcon(file.getPath()));
    return label1;
} else {
    label2.setIcon(new ImageIcon(file.getPath()));
    return label2;
}
}
```

ComboGradient2.java

```
package pelajaran7;

import java.awt.FlowLayout;
import java.io.File;
import java.io.FilenameFilter;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JComboBox;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class ComboGradient2 {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {
```

```
public void run() {  
  
    final File[] files = new File("smiles")  
        .listFiles(new FilenameFilter() {  
  
            public boolean accept(File dir, String name) {  
                if (name.toUpperCase().endsWith(".PNG")) {  
                    return true;  
                }  
                return false;  
            }  
        } );  
  
    for (final File f : files) {  
        System.out.println(f.getName());  
    }  
  
    final DefaultComboBoxModel model = new  
DefaultComboBoxModel(files);  
  
    final JComboBox combo = new JComboBox(model);  
    combo.setRenderer(new GradientRender2());  
    combo.setMaximumRowCount(3);  
  
    final Frame frame = new Frame();  
    frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);  
    frame.setLayout(new FlowLayout());  
    frame.add(combo);  
    frame.pack();  
    frame.setLocationRelativeTo(null);  
    frame.setVisible(true);  
}  
}  
}  
}
```



Gambar 72 ComboGradient2.java

Pelajaran 8

JList

Render

Yach!!! Lagi – lagi soal render. Gak bisa dipungkiri kalo kita bicara tentang extreem di JComboBox, JList, Jtree, dan JTable pasti melibatkan Render. Untuk menset render ke JList gunakan :

```
JList list = new JList();
list.setCellRenderer(CellRenderer render);
```

Background

Sekarang kita akan membuat JList yang dirender dengan warna background seperti tadi pada JComboBox. Dan untuk rendernya kita cukup menggunakan render color milik JComboBox tadi.

ListColor.java

```
package pelajaran8;

import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.GraphicsEnvironment;
import javax.swing.DefaultListModel;
import javax.swing.JList;
import javax.swing.JScrollPane;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;
import pelajaran7.ColorRender;

/**
 * @author usu
 */
public class ListColor {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final String[] data = GraphicsEnvironment
                        .getLocalGraphicsEnvironment().getAvailableFontFamilyNames();

                final DefaultListModel model = new DefaultListModel();
                for (final String f : data) {
                    model.addElement(f);
                }

                final JList list = new JList(model);
                list.setPreferredSize(new Dimension(200, 200));
                list.setCellRenderer(new ColorRender());

                final Frame frame = new Frame();
                frame.setLayout(new FlowLayout());
            }
        });
    }
}
```

```
        frame.add(new JScrollPane(list));
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
    }
}
}
```



Gambar 73 ListColor.java

ListColor2.java

```
package pelajaran8;

import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.GraphicsEnvironment;
import javax.swing.DefaultListModel;
import javax.swing.JList;
import javax.swing.JScrollPane;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;
import pelajaran7.ColorRenderRandom;

/**
 * @author usu
 */
public class ListColor2 {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final String[] data = GraphicsEnvironment
                        .getLocalGraphicsEnvironment().getAvailableFontFamilyNames();

                final DefaultListModel model = new DefaultListModel();
                for (final String f : data) {
                    model.addElement(f);
                }

                final JList list = new JList(model);
                list.setPreferredSize(new Dimension(200, 200));
                list.setCellRenderer(new ColorRenderRandom());
            }
        });
    }
}
```

```
        final Frame frame = new Frame();
        frame.setLayout(new FlowLayout());
        frame.add(new JScrollPane(list));
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
    }
}
}
```



Gambar 74 ListColor2.java

Image

Sekarang kita akan menggunakan render image milik JComboBox tadi untuk menampilkan gambar dalam JList

ListBackground.java

```
package pelajaran8;

import java.io.File;
import java.io.FilenameFilter;
import javax.swing.DefaultListModel;
import javax.swing.JList;
import javax.swing.JScrollPane;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;
import pelajaran7.ImageRender;

/**
 * @author usu
 */
public class ListBackground {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final File[] files = new File("smiles").listFiles(new
FilenameFilter() {

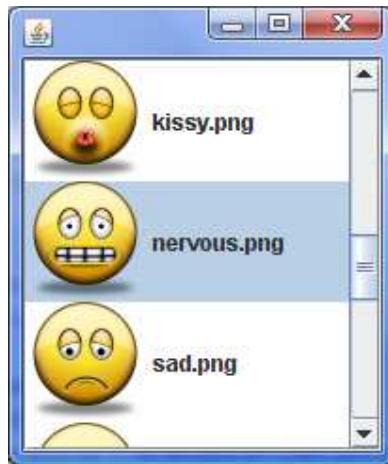
                    public boolean accept(File dir, String name) {
```

```
        if (name.toUpperCase().endsWith(".PNG")) {
            return true;
        }
        return false;
    }
} );

final DefaultListModel model = new DefaultListModel();
for (final File f : files) {
    model.addElement(f);
}

final JList list = new JList(model);
list.setCellRenderer(new ImageRender());

final Frame frame = new Frame();
frame.add(new JScrollPane(list));
frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
frame.pack();
frame.setVisible(true);
}
}
}
```



Gambar 75 ListBackground.java

Boolean

Pada pembahasan sebelumnya saya buat render menggunakan JLabel, namun sebenarnya tak hanya JLabel, kita juga bisa menampilkan render class yang merupakan turunan dari Component. Misal kali ini kita akan menampilkan render boolean menggunakan JCheckBox.

BooleanRender.java

```
package pelajaran8;

import java.awt.Component;
import javax.swing.JCheckBox;
import javax.swing.JList;
import javax.swing.ListCellRenderer;
```

```
/*
 * @author usu
 */
public class BooleanRender implements ListCellRenderer {

    public Component getListCellRendererComponent(final JList list, final
Object value,
        final int index, final boolean isSelected, final boolean
cellHasFocus) {
        final boolean nilai =
Boolean.valueOf(value.toString()).booleanValue();
        final JCheckBox chek = new JCheckBox(value.toString());
        chek.setOpaque(true);
        chek.setSelected(nilai);

        if (isSelected) {
            chek.setBackground(list.getSelectionBackground());
            chek.setForeground(list.getSelectionForeground());
        } else {
            chek.setBackground(list.getBackground());
            chek.setForeground(list.getForeground());
        }

        return chek;
    }
}
```

Sekarang tinggal kita gunakan render boolean tersebut ke JList

ListBoolean.java

```
package pelajaran8;

import java.awt.BorderLayout;
import java.util.Random;
import javax.swing.DefaultListModel;
import javax.swing.JList;
import javax.swing.JScrollPane;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class ListBoolean {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {

                final Random generator = new Random();

                final DefaultListModel model = new DefaultListModel();
                for (int i = 0; i < 100; i++) {
                    model.addElement(generator.nextBoolean());
                }
            }
        });
    }
}
```

```
    final JList list = new JList(model);
    list.setCellRenderer(new BooleanRender());

    final Frame frame = new Frame();
    frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    frame.setLayout(new BorderLayout());
    frame.add(new JScrollPane(list));
    frame.setSize(200, 200);
    frame.setVisible(true);
}
}
}
```



Gambar 76 ListBoolean.java

Number

Kembali lagi ke render menggunakan JLabel tapi kali ini saya mau memberi sedikit sentuhan kreasi nomber. Ingin tahu apa, terus baca...

NumberRender.java

```
package pelajaran8;

import java.awt.Component;
import java.awt.Font;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.ListCellRenderer;

import javax.swing.ListCellRenderer;
```



```
/**
 * @author usu
 */
public class NumberRender implements ListCellRenderer {

    public Component getListCellRendererComponent(final JList list, final
Object value,
        final int index, final boolean isSelected, final boolean
cellHasFocus) {
        final JLabel label = new JLabel(value.toString());
        label.setOpaque(true);
```

```
final int size = Integer.valueOf(value.toString()).intValue();

final Font f = new Font(label.getFont().getFamily(),
label.getFont().getStyle(), size);
label.setFont(f);

if (isSelected) {
    label.setBackground(list.getSelectionBackground());
    label.setForeground(list.getSelectionForeground());
} else {
    label.setBackground(list.getBackground());
    label.setForeground(list.getForeground());
}

return label;
}

}
```

OK selesai, sekarang tinggal menggunakannya ke JList

ListNumber.java

```
package pelajaran8;

import java.awt.BorderLayout;
import javax.swing.DefaultListModel;
import javax.swing.JList;
import javax.swing.JScrollPane;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class ListNumber {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

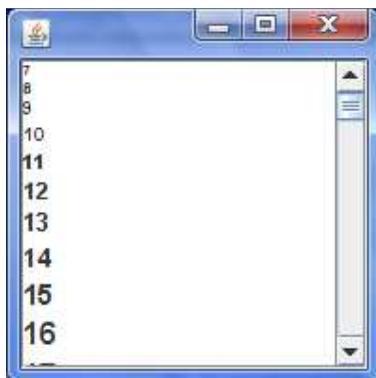
            public void run() {

                final DefaultListModel model = new DefaultListModel();
                for (int i = 0; i < 100; i++) {
                    model.addElement(i);
                }

                final JList list = new JList(model);
                list.setCellRenderer(new NumberRender());

                final Frame frame = new Frame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setLayout(new BorderLayout());
                frame.add(new JScrollPane(list));
                frame.setSize(200, 200);
                frame.setVisible(true);
            }
        });
    }
}
```

```
    }  
}
```



Gambar 77 ListNumber.java

Hahaha, trik ini sebenarnya jarang digunakan tapi lumayan buat style aja.

Font

Sekarang misalnya kita ingin membuat font chooser dan kita ingin menampilkan fontnya di JList. Itu sich gampang, bisa diatur...

FontRender.java

```
package pelajaran8;  
  
import java.awt.Component;  
import java.awt.Font;  
import javax.swing.JLabel;  
import javax.swing.JList;  
import javax.swing.ListCellRenderer;  
  
/**  
 * @author usu  
 */  
public class FontRender implements ListCellRenderer {  
  
    public Component getListCellRendererComponent(final JList list, final Object value,  
                                                final int index, final boolean isSelected, final boolean cellHasFocus) {  
        final String font = value.toString();  
  
        final JLabel label = new JLabel(font);  
        label.setOpaque(true);  
  
        final Font f = new Font(font, label.getFont().getStyle(),  
                               label.getFont().getSize());  
        label.setFont(f);  
  
        if (isSelected) {  
            label.setBackground(list.getSelectionBackground());  
            label.setForeground(list.getSelectionForeground());  
        } else {
```

```
        label.setBackground(list.getBackground());
        label.setForeground(list.getForeground());
    }

    return label;
}
}
```

Sim salabim sekarang semua font akan berubah sesuai dengan font teks

ListFont.java

```
package pelajaran8;

import java.awt.BorderLayout;
import java.awt.GraphicsEnvironment;
import javax.swing.DefaultListModel;
import javax.swing.JList;
import javax.swing.JScrollPane;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class ListFont {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {

                final String[] data =
GraphicsEnvironment.getLocalGraphicsEnvironment()
                    .getAvailableFontFamilyNames();

                final DefaultListModel model = new DefaultListModel();
                for (final String f : data) {
                    model.addElement(f);
                }

                final JList list = new JList(model);
                list.setCellRenderer(new FontRender());

                final Frame frame = new Frame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setLayout(new BorderLayout());
                frame.add(new JScrollPane(list));
                frame.setSize(200, 200);
                frame.setVisible(true);
            }
        });
    }
}
```



Gambar 78 ListFont.java

Model

Selain memanipulasi JList lewat render, kita juga dapat memanipulasi swing khususnya lebih ditekankan untuk memanipulasi data yang dimiliki JList.

Sorting

Biasanya untuk menampung data JList kita menggunakan DefaultListModel, namun sampai tulisan ini dibuat DefaultListModel belum memiliki kemampuan untuk mengurutkan data. Jadi sekarang kita akan mencoba untuk membuat ListModel yang dapat mengurutkan data.

Sebelumnya kita buat enum sorting.

Sorting.java

```
package pelajaran8;

/**
 * @author usu
 */
public enum Sorting {
    ASCENDING("ASCENDING"),
    DEFAULT("DEFAULT"),
    DESCENDING("DESCENDING");

    private String sortingMethode;

    private Sorting(final String sortingMethode) {
        this.sortingMethode = sortingMethode;
    }

    public String getSortingMethod() {
        return this.sortingMethode;
    }

    @Override
    public String toString() {
        return getSortingMethod();
    }
}
```

```
}
```

Sekarang kita buat modelnya

ListModelUsu.java

```
package pelajaran8;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import javax.swing.AbstractListModel;

/**
 * @author usu
 */
public class ListModelUsu extends AbstractListModel {

    private final ArrayList defaultData;
    private final ArrayList filterData;
    private Sorting sort;

    public ListModelUsu() {
        super();
        this.defaultData = new ArrayList();
        this.filterData = new ArrayList();
        setSort(Sorting.DEFAULT);
    }

    public void add(final Object e) {
        this.defaultData.add(e);
        filter();
    }

    @SuppressWarnings("unchecked")
    private void filter() {
        Comparator comp = null;
        if (getSort() == Sorting.ASCENDING) {
            comp = new Comparator() {

                public int compare(final Object o1, final Object o2) {
                    if (o1 instanceof Comparator) {
                        return ((Comparable) o1).compareTo(o2);
                    } else {
                        return o1.toString().compareTo(o2.toString());
                    }
                }
            };
        } else if (getSort() == Sorting.DESCENDING) {
            comp = new Comparator() {

                public int compare(final Object o1, final Object o2) {
                    if (o1 instanceof Comparator) {
                        return ((Comparable) o2).compareTo(o1);
                    } else {
                        return o2.toString().compareTo(o1.toString());
                    }
                }
            };
        }
    }
}
```

```
    } else {
        comp = null;
    }

    this.filterData.clear();
    this.filterData.addAll(this.defaultData);

    if (comp != null) {
        Collections.sort(this.filterData, comp);
    }
    fireContentsChanged(this, 0, this.filterData.size() - 1);
}

public Object getElementAt(final int index) {
    return this.filterData.get(index);
}

public int getSize() {
    return this.filterData.size();
}

public Sorting getSort() {
    return this.sort;
}

public void remove(final Object o) {
    this.defaultData.remove(o);
    filter();
}

public void setSort(final Sorting sort) {
    if (getSort() != sort) {
        this.sort = sort;
        filter();
    }
}
```

Dan sekarang anda bisa menggunakan model tersebut untuk JList

```
ListModelUsu model = new ListModelUsu();
list.setModel(model);
```

Dan untuk mengurutkan ascending

```
model.setSort(Sorting.ASCENDING);
```

Untuk mengurutkan descending

```
model.setSort(Sorting.DESCENDING);
```

Dan untuk ke urutan awal semula

model.setSort(Sorting.DEFAULT);

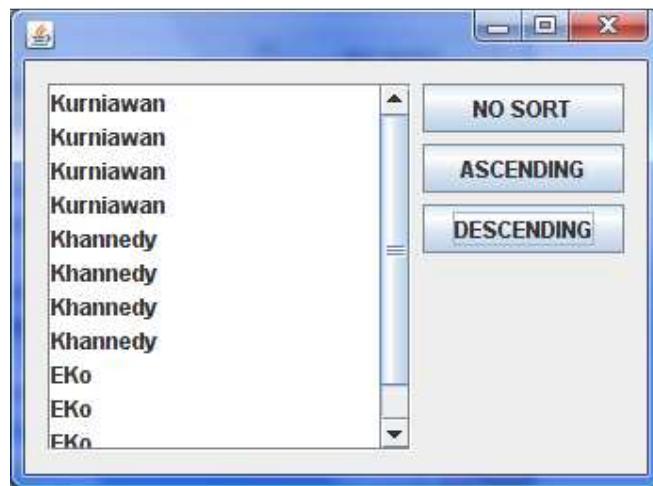
Contoh aplikasinya :



Gambar 79 List menggunakan ListModelUsu saat keadaan No SORT



Gambar 80 List menggunakan ListModelUsu saat keadaan ASCENDING



Gambar 81 List menggunakan ListModelUsu saat keadaan DESCENDING

Untuk source kode aplikasi diatas anda bisa melihatnya di project SwingMakeOver

Filter

Sekarang masih dalam topik model. Kita akan membuat model yang bisa difilter. Misal kita ingin mencari data String yang mengandung kata "ba" dalam model JList sehingga JList hanya menampilkan data yang mengandung kata "ba" tanpa menghapus data lainnya, nah itu lah filter.

Untuk membuat filter lebih baik kita gunakan model yang tadi dibuat agar selain bisa difilter model juga bisa diurutkan.

ListModelUsuFilter.java

```
package pelajaran8;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import javax.swing.AbstractListModel;

/**
 * @author usu
 */
public class ListModelUsuFilter extends AbstractListModel {

    private final ArrayList defaultData;
    private String filter;
    private final ArrayList filterData;
    private Sorting sort;

    public ListModelUsuFilter() {
        super();
        this.defaultData = new ArrayList();
        this.filterData = new ArrayList();
        setSort(Sorting.DEFAULT);
    }

    public void add(final Object e) {
        this.defaultData.add(e);
    }
}
```

```
        filter();
    }

@SuppressWarnings("unchecked")
private void filter() {
    Comparator comp = null;
    if (getSort() == Sorting.ASCENDING) {
        comp = new Comparator() {

            public int compare(final Object o1, final Object o2) {
                if (o1 instanceof Comparator) {
                    return ((Comparable) o1).compareTo(o2);
                } else {
                    return o1.toString().compareTo(o2.toString());
                }
            }
        };
    } else if (getSort() == Sorting.DESCENDING) {
        comp = new Comparator() {

            public int compare(final Object o1, final Object o2) {
                if (o1 instanceof Comparator) {
                    return ((Comparable) o2).compareTo(o1);
                } else {
                    return o2.toString().compareTo(o1.toString());
                }
            }
        };
    } else {
        comp = null;
    }

    this.filterData.clear();
    if (getFilter() == null) {
        this.filterData.addAll(this.defaultData);
    } else {
        for (final Object o : this.defaultData) {
            if (o.toString().contains(getFilter())) {
                this.filterData.add(o);
            }
        }
    }

    if (comp != null) {
        Collections.sort(this.filterData, comp);
    }
    fireContentsChanged(this, 0, this.filterData.size() - 1);
}

public Object getElementAt(final int index) {
    return this.filterData.get(index);
}

public String getFilter() {
    return this.filter;
}

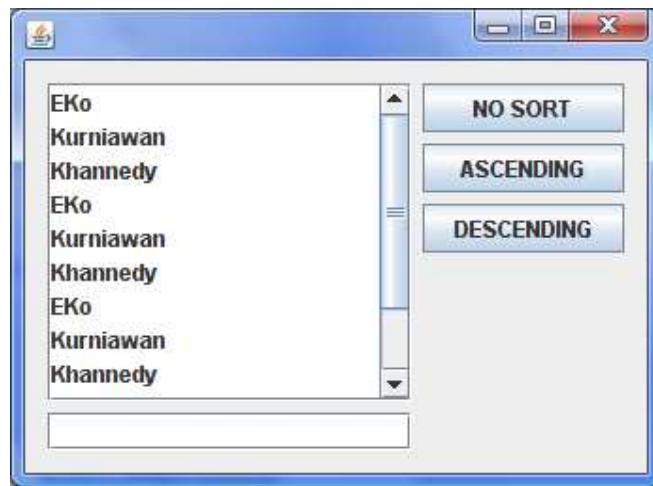
public int getSize() {
    return this.filterData.size();
}
```

```
public Sorting getSort() {
    return this.sort;
}

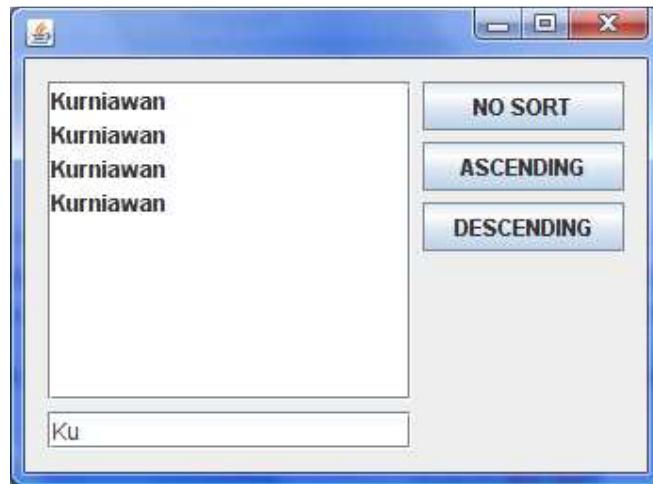
public void remove(final Object o) {
    this.defaultData.remove(o);
    filter();
}

public void setFilter(final String filter) {
    this.filter = filter;
    filter();
}

public void setSort(final Sorting sort) {
    if (getSort() != sort) {
        this.sort = sort;
        filter();
    }
}
```



Gambar 82 List menggunakan ListModelUsuFilter saat tidak di filter



Gambar 83 List menggunakan ListModelUsuFilter saat difilter

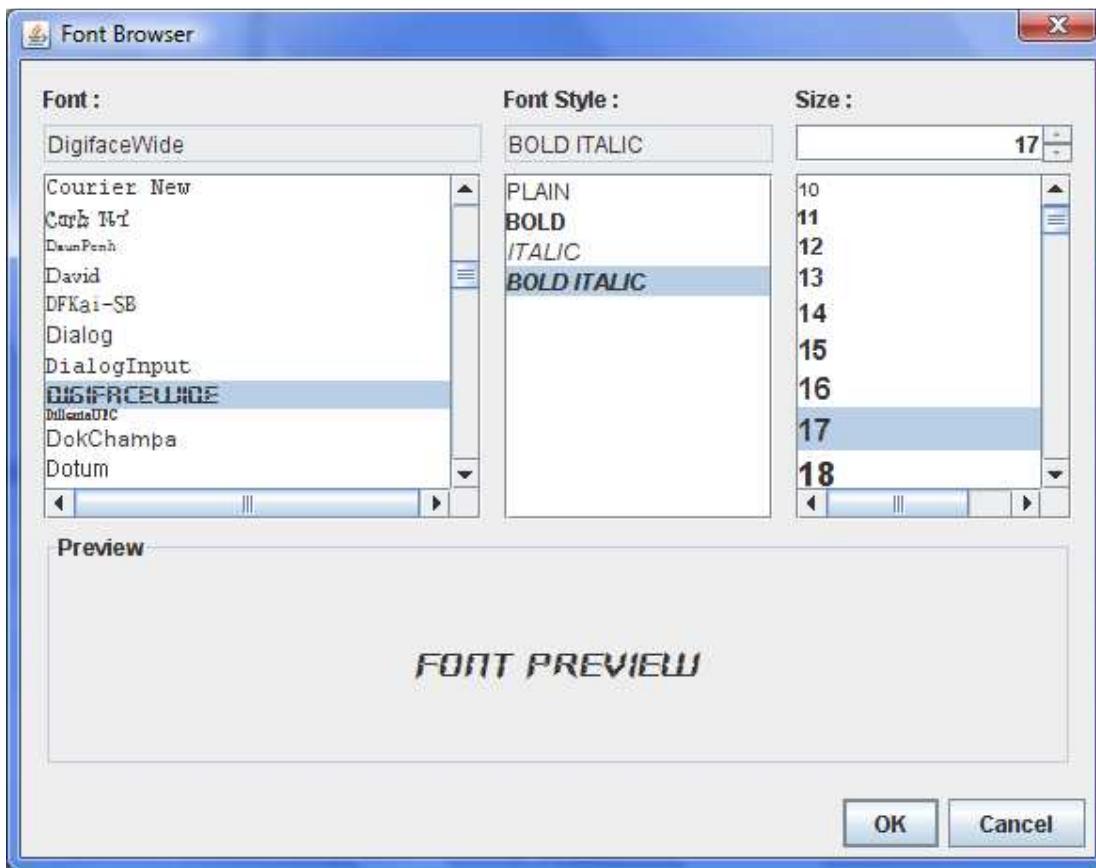
Kode untuk aplikasi diatas bisa anda dapatkan di project SwingMakeOver. Dan inti untuk memfilter diatas adalah dengan menggunakan kode :

```
filterData.clear();
if (getFilter() == null) {
    filterData.addAll(defaultData);
} else {
    for (Object o : defaultData) {
        if (o.toString().contains(getFilter())) {
            filterData.add(o);
        }
    }
}
```

Kesimpulan

Dari pembahasan diatas seputar JList anda bisa membuat berbagai macam kreasi dengan render JComboBox seperti pada render JList, karena memang JList dan JComboBox memiliki render yang sama yaitu interface ListCellRendered

Sekarang silahkan anda coba membuat font browser seperti dibawah ini :



Gambar 84 Font Browser

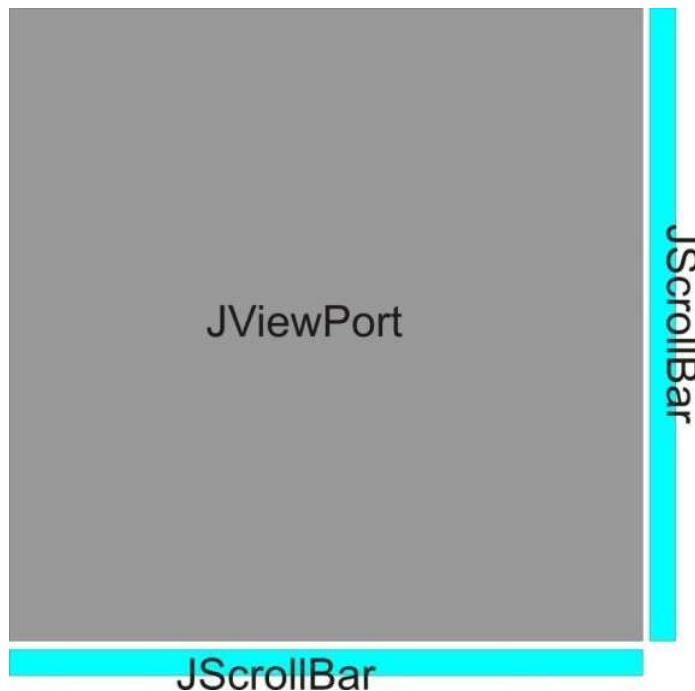
Kali ini saya sengaja tak memasukkan source kodennya karena hal ini pasti bisa anda buat dari gabungan – gabungan pembahasan sebelumnya.

Pelajaran 9

JViewport

Apa tuh JViewport? Mungkin anda pernah mendengar atau juga anda belum pernah mendengarnya. Yach tentu saja kalo anda jarang mendengar tentang ini karena memang JViewport itu jarang sekali digunakan. Lantas kenapa kita membahasnya?

Alasan yang paling utama adalah JScrollPane memerlukan JViewport untuk menampilkan component yang berada didalamnya.



Gambar 85 JViewport dalam JScrollPane

Misal kita ingin membuat JTextArea yang transparan, maka mau gak mau selain membuat komponen JTextAreanya transparan, kita juga harus membuat JViewportnya transparan.

ViewPortTransparan.java

```
package pelajaran9;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JViewport;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.border.EmptyBorder;
```

```
import pelajaran1.Frame;

/**
 * @author usu
 */
public class ViewPortTranparan {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final JTextArea area = new JTextArea();
                area.setOpaque(false);

                final JViewport port = new JViewport();
                port.setOpaque(false);
                port.setView(area);

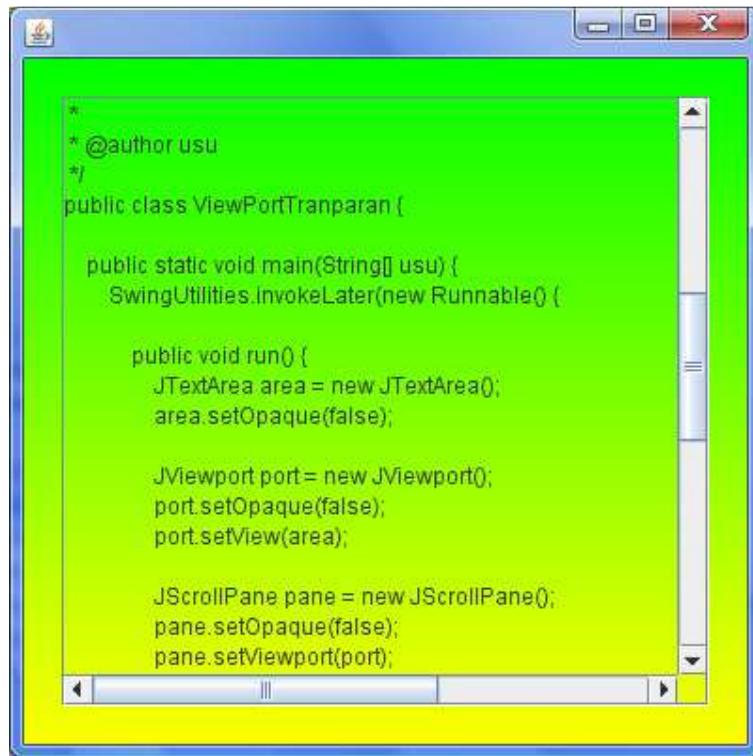
                final JScrollPane pane = new JScrollPane();
                pane.setOpaque(false);
                pane.setViewport(port);

                final JPanel panel = new JPanel(new BorderLayout()) {

                    @Override
                    protected void paintComponent(Graphics g) {
                        super.paintComponent(g);

                        Graphics2D g2 = (Graphics2D) g.create();
                        g2.setPaint(new GradientPaint(0, 0, Color.GREEN, 0,
                            getHeight(), Color.YELLOW));
                        g2.fillRect(0, 0, getWidth(), getHeight());
                    }
                };
                panel.setBorder(new EmptyBorder(20, 20, 20, 20));
                panel.add(pane);

                final Frame frame = new Frame();
                frame.setLayout(new BorderLayout());
                frame.add(panel);
                frame.setSize(400, 400);
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setVisible(true);
            }
        });
    }
}
```



Gambar 86 ViewPortTranparan.java

JTextArea

Kali ini kita akan masuk ke pembahasan tentang kolaborasi JViewport dan JTextArea

Background Image

Salah satu manfaat JViewport untuk JTextArea adalah menampilkan background gambar.

ViewPortImage.java

```
package pelajaran9;

import java.awt.BorderLayout;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
import javax.swing.ImageIcon;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JViewport;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class ViewPortImage extends JViewport {

    public static void main(final String[] usu) {
```

```
SwingUtilities.invokeLater(new Runnable() {

    public void run() {
        final JTextArea area = new JTextArea();
        area.setOpaque(false);

        final ViewPortImage port = new ViewPortImage();
        port.setView(area);
        port
            .setImage(new
ImageIcon(getClass().getResource("/pelajaran9/image.jpg"))
            .getImage());

        final JScrollPane pane = new JScrollPane();
        pane.setViewport(port);

        final Frame frame = new Frame();
        frame.setLayout(new BorderLayout());
        frame.add(pane);
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
} );
}

private Image image;

public ViewPortImage() {
    super();
}

public Image getImage() {
    return this.image;
}

@Override
protected void paintComponent(final Graphics g) {
    super.paintComponent(g);
    if (getImage() != null) {
        final Graphics2D g2 = (Graphics2D) g.create();
        for (int i = 0; i < getWidth(); i += getImage().getWidth(null)) {
            for (int j = 0; j < getHeight(); j +=
getImage().getHeight(null)) {
                g2.drawImage(getImage(), i, j, null);
            }
        }
    }
}

public void setImage(final Image image) {
    this.image = image;
    repaint();
}
}
```



Gambar 87 ViewPortImage.java



Gambar 88 ViewPortImage.java

Berbeda jika kita membuat background gambarnya di JTextArea, pada JViewport gambar background akan diam walaupun scrollbar discroll keatas, kebawah, kekanan atau kekiri. Jika kita membuat gambar dalam JTextArea maka gambar belakang akan ikut scroll (berubah) seiring dengan lokasi scrollbar dirubah.

TextArealmage.java

```
package pelajaran9;  
  
import java.awt.BorderLayout;  
import java.awt.Graphics;  
import java.awt.Graphics2D;  
import java.awt.Image;  
import javax.swing.ImageIcon;
```

```
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TextAreaImage extends JTextArea {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final TextAreaImage area = new TextAreaImage();
                area
                    .setImage(new
ImageIcon(getClass().getResource("/pelajaran9/image.jpg"))
                    .getImage());

                final Frame frame = new Frame();
                frame.setLayout(new BorderLayout());
                frame.add(new JScrollPane(area));
                frame.setSize(400, 300);
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setVisible(true);
            }
        });
    }

    private Image image;

    public TextAreaImage() {
        super();
    }

    public Image getImage() {
        return this.image;
    }

    @Override
    protected void paintComponent(final Graphics g) {
        if (isOpaque()) {
            setOpaque(false);
        }
        if (getImage() != null) {
            final Graphics2D g2 = (Graphics2D) g.create();
            for (int i = 0; i < getWidth(); i += getImage().getWidth(null)) {
                for (int j = 0; j < getHeight(); j +=
getImage().getHeight(null)) {
                    g2.drawImage(getImage(), i, j, null);
                }
            }
        }
        super.paintComponent(g);
    }

    public void setImage(final Image image) {
        this.image = image;
        repaint();
    }
}
```

```
}
```



Gambar 89 TextArealImage.java



Gambar 90 TextArealImage.java

Jadi menurut saya, lebih baik membuat background di JViewport dari pada di JTextArea.

Glass

Membuat efek glass berarti kita harus membuat efeknya diatas JTextArea, sedangkan metode paintComponent() untuk JViewport akan digambar dibelakang JTextArea, sehingga untuk membuat afek glass atau efek-efek yang berada di depan JTextArea, kita harus mengoveride metode paintChildren().

ViewPortImageGlass.java

```
package pelajaran9;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
import javax.swing.ImageIcon;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JViewport;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class ViewPortImageGlass extends JViewport {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final JTextArea area = new JTextArea();
                area.setOpaque(false);

                final ViewPortImageGlass port = new ViewPortImageGlass();
                port.setView(area);
                port
                    .setImage(new
ImageIcon(getClass().getResource("/pelajaran9/image.jpg"))
                    .getImage());

                final JScrollPane pane = new JScrollPane();
                pane.setViewport(port);

                final Frame frame = new Frame();
                frame.setLayout(new BorderLayout());
                frame.add(pane);
                frame.setSize(400, 300);
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setVisible(true);
            }
        });
    }

    private Image image;

    public ViewPortImageGlass() {
        super();
    }

    public Image getImage() {
        return this.image;
    }

    @Override
```

```
protected void paintChildren(final Graphics g) {
    super.paintChildren(g);
    final Graphics2D g2 = (Graphics2D) g.create();

    final Color dark = new Color(1F, 0F, 0F, 0F);
    final Color light = new Color(1F, 0F, 0F, 0.5F);
    final GradientPaint paint = new GradientPaint(0, 0, light, 0,
getHeight() / 2, dark);

    g2.setPaint(paint);
    g2.fillRect(0, 0, getWidth(), getHeight());
}

@Override
protected void paintComponent(final Graphics g) {
    final Graphics2D g2 = (Graphics2D) g.create();
    super.paintComponent(g);
    if (getImage() != null) {
        for (int i = 0; i < getWidth(); i += getImage().getWidth(null)) {
            for (int j = 0; j < getHeight(); j +=
getImage().getHeight(null)) {
                g2.drawImage(getImage(), i, j, null);
            }
        }
    }
}

public void setImage(final Image image) {
    this.image = image;
    repaint();
}
}
```



Gambar 91 ViewPortImageGlass.java

Jadi ingat kalo ingin membuat gambar yang diletakkan di atas component yang ditampilkan JViewport semisal JTextArea, gunakan metode paintChildren(). OK!!!

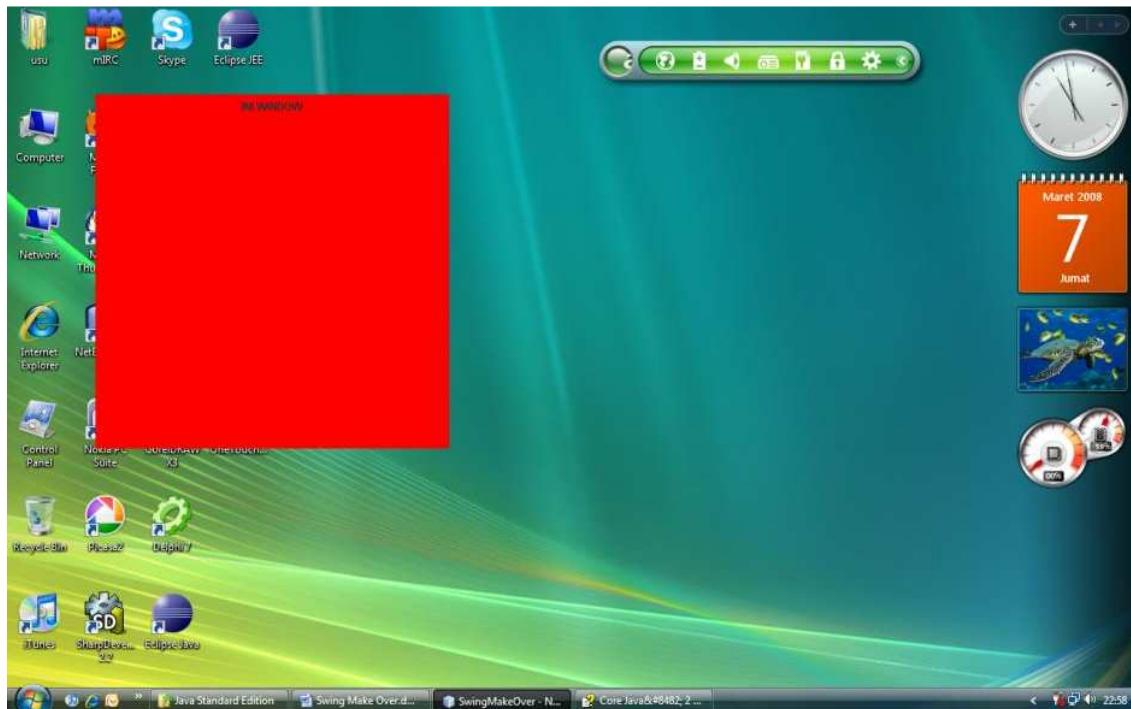
Pelajaran 10

JWindow

Seperti halnya JViewPort, JWindow pun memang jarang sekali digunakan untuk membuat aplikasi. Tapi ada kalanya JWindow berguna, misal untuk menampilkan splashscreen atau about screen.

About Screen

Layaknya JFrame, JWindow pun merupakan Form, namun tak memiliki style System Operasi, dengan kata lain JWindow terlihat seperti panel yang berdiri sendiri :



Gambar 92 Contoh JWindow

Anda dapat melihat perbedaan window dengan JFrame dari gambar diatas.

Sekarang kita akan membuat AboutScreen menggunakan JWindow.

WindowAbout.java

```
package pelajaran10;

import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.AbstractAction;
import javax.swing.ImageIcon;
import javax.swing.JButton;
```

```
import javax.swing.JLabel;
import javax.swing.JFrame;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class WindowAbout extends JFrame {

    public static void main(final String[] args) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final WindowAbout window = new WindowAbout();
                window.setAlwaysOnTop(true);
                window.setImage(new
ImageIcon(getClass().getResource("/pelajaran10/about.jpg"))
                .getImage());

                final Frame frame = new Frame();
                frame.setLayout(new FlowLayout());
                frame.add(new JButton(new AbstractAction("SHOW ABOUT") {

                    public void actionPerformed(final ActionEvent e) {
                        window.setLocationRelativeTo(null);
                        window.setVisible(true);
                    }
                }));
                frame.pack();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setVisible(true);
            }
        });
    }

    private Image image;

    private final JLabel label;

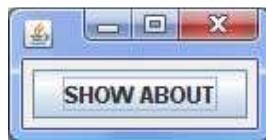
    public WindowAbout() {
        super();
        this.label = new JLabel();
        setLayout(new BorderLayout());
        add(this.label);
        addMouseListener(new MouseAdapter() {

            @Override
            public void mousePressed(final MouseEvent e) {
                setVisible(false);
            }
        });
    }

    public Image getImage() {
        return this.image;
    }

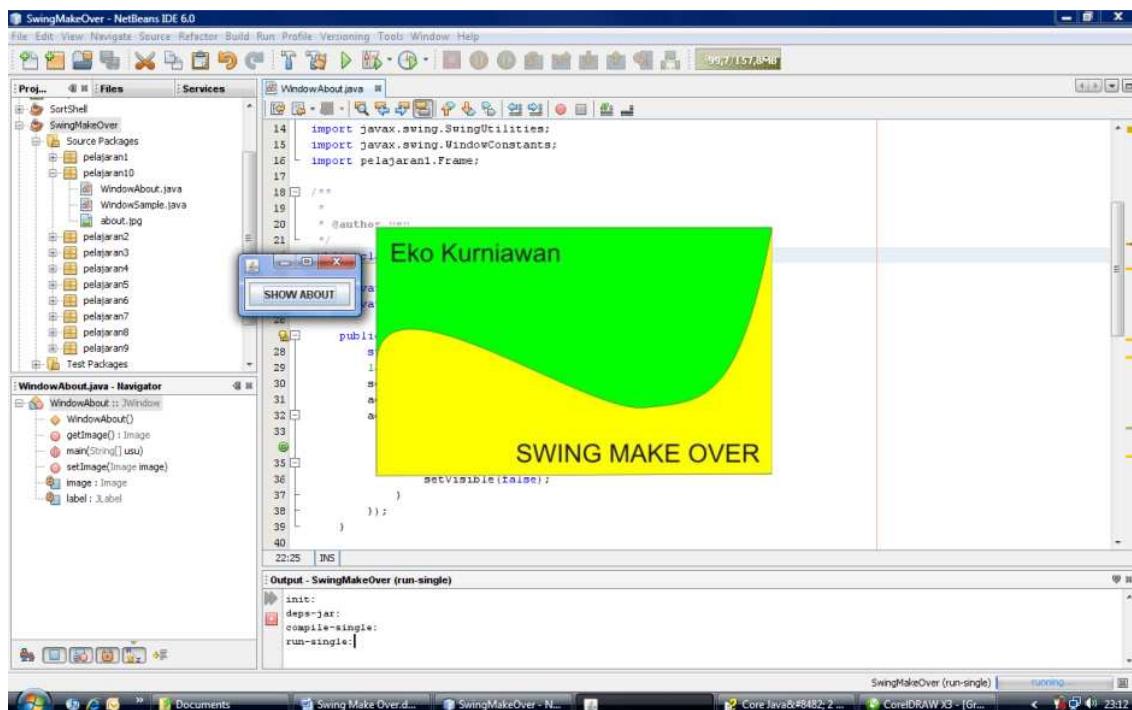
    public void setImage(final Image image) {
        this.image = image;
    }
}
```

```
        this.label.setIcon(new ImageIcon(image));
        pack();
    }
}
```



Gambar 93 WindowAbout.java

Jika kita menekan tombol SHOW ABOUT maka akan keluar JWindow About



Gambar 94 WindowAbout.java saat menampilkan AboutScreen

Splash Screen

OK sekarang kita coba membuat splash screen menggunakan JWindow.

WindowSplashScreen.java

```
package pelajaran10;

import java.awt.BorderLayout;
import java.awt.Image;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JWindow;
import javax.swing.WindowConstants;
```

```
import pelajaran1.Frame;

/**
 * @author usu
 */
public class WindowSplashScreen extends JWindow {

    public static void main(final String[] usu) {
        new Thread(new Runnable() {

            public void run() {
                try {
                    final WindowSplashScreen window = new WindowSplashScreen();
                    window.setAlwaysOnTop(true);
                    window.setImage(new
ImageIcon(getClass().getResource("/pelajaran10/about.jpg"))
                    .getImage());
                    window.setLocationRelativeTo(null);

                    window.setVisible(true);

                    Thread.sleep(3000);

                    final Frame frame = new Frame();
                    frame.setSize(400, 400);

frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                    frame.setLocationRelativeTo(null);

                    window.setVisible(false);
                    frame.setVisible(true);
                } catch (final InterruptedException ex) {
Logger.getLogger(WindowSplashScreen.class.getName()).log(Level.SEVERE,
null, ex);
                }
            }
        }).start();
    }

    private Image image;

    private final JLabel label;

    public WindowSplashScreen() {
        super();
        this.label = new JLabel();
        setLayout(new BorderLayout());
        add(this.label);
    }

    public Image getImage() {
        return this.image;
    }

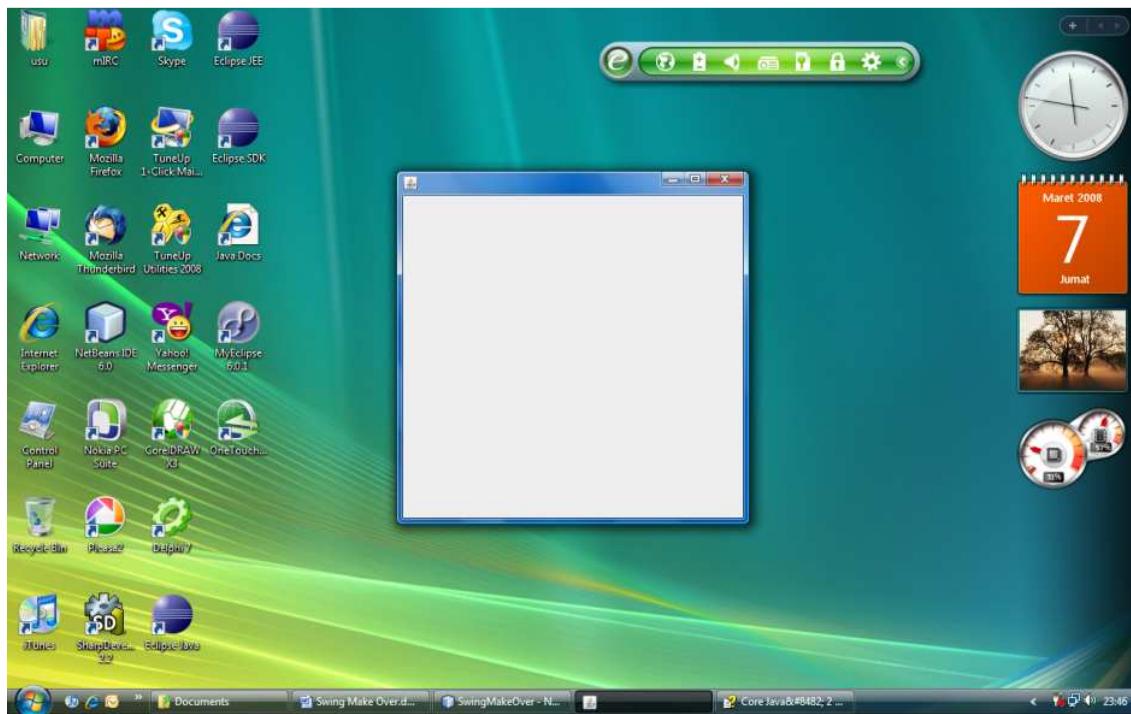
    public void setImage(final Image image) {
        this.image = image;
        this.label.setIcon(new ImageIcon(image));
        pack();
    }
}
```

Splash screen tampil selama 3 detik



Gambar 95 WindowSplashScreen.java saat splashscreen

Setelah itu splash screen menghilang dan Frame aplikasi keluar



Gambar 96 WindowSplashScreen.java saat form muncul

Kesimpulan

Walaupun JWindow jarang digunakan, namun kadang JWindow bermanfaat juga, contohnya membuat About Screen atau Splash Screen.

Tapi selain untuk kedua hal diatas, kita juga bisa memanfaatkan JWindow sebagai tooltip, misal kita ingin membuat sebuah tooltip untuk component, lalu kita beri aksi onMouseEnter, JWindow muncul dan onMouseExit, JWindow menghilang.

WindowTooltip.java

```
package pelajaran10;

import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.Point;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.JWindow;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;

/**
 * @author usu
 */
public class WindowTooltip extends JFrame {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                new WindowTooltip().setVisible(true);
            }
        });
    }

    private final JWindow window;

    public WindowTooltip() {
        super();

        this.window = new JWindow(this);
        this.window.setAlwaysOnTop(true);
        this.window.setLayout(new BorderLayout());
        this.window.add(new JScrollPane(new JTextArea("INI ADALAH TOOLTIP
WINDOW")));
        this.window.setSize(200, 200);

        final JTextField field = new JTextField();
        field.setColumns(30);
        field.addMouseListener(new MouseAdapter() {

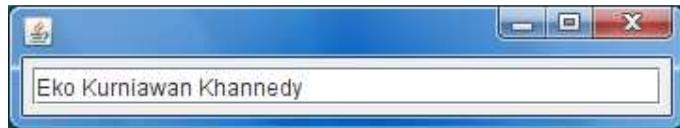
            @Override
            public void mouseEntered(MouseEvent e) {
                final Point p = field.getLocationOnScreen();
            }
        });
    }
}
```

```
p.y += field.getHeight();
WindowTooltip.this.window.setLocation(p);
WindowTooltip.this.window.setVisible(true);
}

@Override
public void mouseExited(MouseEvent e) {
    WindowTooltip.this.window.setVisible(false);
}
);

setLayout(new FlowLayout());
setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
add(field);
pack();
}
}
```

Ketika mouse berada di luar JTextField, maka JWindow tak muncul atau menghilang. Sedangkan ketika mouse masuk ke wilayah JTextField, maka secara otomatis JWindow akan muncul.



Gambar 97 WindowTooltip.java



Gambar 98 WindowTooltip.java saat window tooltip muncul

Jadi jangan anggap enteng JWindow, karena kadang perlu kreativitas untuk membuat JWindow jadi bermanfaat.

Pelajaran 11

JTree

Render

Yup, masuk lagi soal render. Tapi kali ini kita akan merender JTree. Walaupun kenyataannya saya jarang sekali menggunakan JTree, hahaha jadi saya jarang mengexpose JTree. Untuk menset render ke JTree gunakan :

```
JTree tree = new JTree();
tree.setCellRenderer(TreeCellRenderer render);
```

File and Folder

Sekarang kita akan membuat sebuah cell render untuk menampilkan JTree yang menampung data file and directory.

FileRender.java

```
package pelajaran11;

import java.awt.Component;
import java.io.File;
import javax.swing.ImageIcon;
import javax.swing.JTree;
import javax.swing.tree.DefaultTreeCellRenderer;

/**
 * @author usu
 */
public class FileRender extends DefaultTreeCellRenderer {

    public FileRender() {
        super();
    }

    @Override
    public Component getTreeCellRendererComponent(final JTree tree, final
Object value,
        final boolean sel, final boolean expanded, final boolean leaf,
final int row,
        final boolean hasFocus) {

        final File file = new File(value.toString());
        setOpenIcon(new
ImageIcon(getClass().getResource("/pelajaran11/folder.png")));
        setClosedIcon(getOpenIcon());
        setLeafIcon(new
ImageIcon(getClass().getResource("/pelajaran11/file.png")));

        final Component c = super.getTreeCellRendererComponent(tree, value,
sel, expanded, leaf, row,
            hasFocus);
        setText(file.getName());
    }
}
```

```
    return c;
}
}
```

Setelah membuat Cell Rendernya, sekarang tinggal digunakan ke JTree nya.

TreeFile.java

```
package pelajaran11;

import java.awt.BorderLayout;
import java.io.File;
import javax.swing.JScrollPane;
import javax.swing.JTree;
import javax.swing.WindowConstants;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.DefaultTreeModel;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TreeFile extends JTree {

    public static final void insertNode(final DefaultMutableTreeNode node,
final File folder) {
        for (final File f : folder.listFiles()) {
            final DefaultMutableTreeNode child = new
DefaultMutableTreeNode(f);
            if (f.isDirectory()) {
                TreeFile.insertNode(child, f);
            }
            node.add(child);
        }
    }

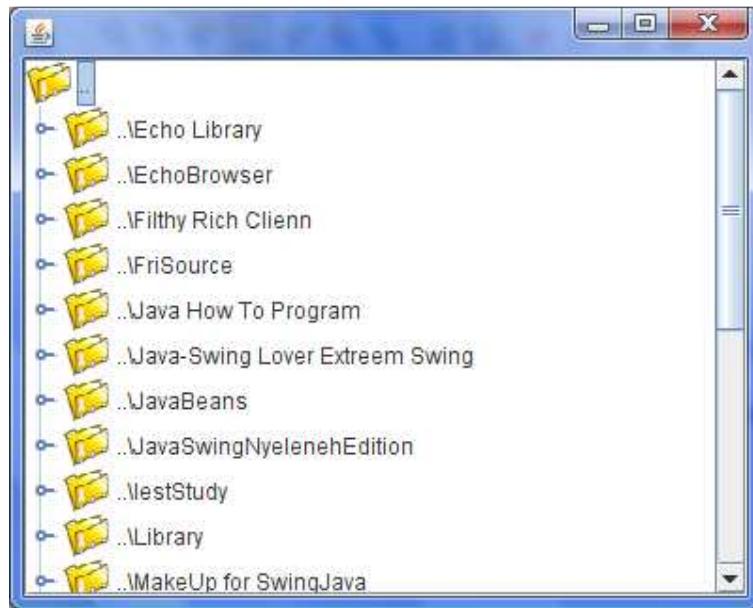
    public static void main(final String[] usu) {
        final Frame frame = new Frame();
        frame.setLayout(new BorderLayout());
        frame.add(new JScrollPane(new TreeFile()));
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }

    public TreeFile() {
        super();

        final DefaultMutableTreeNode root = new DefaultMutableTreeNode(new
File(".."));

        TreeFile.insertNode(root, new File("."));

        final DefaultTreeModel model = new DefaultTreeModel(root);
        setModel(model);
        setCellRenderer(new FileRender());
    }
}
```



Gambar 99 FileRender.java

About Icon

Oh iya, sebelum saya lanjut ke trix render selanjutnya saya mau bilang kalo di JTree terdapat 3 Icon, pertama openIcon yaitu icon yang digunakan oleh direktori (node yang memiliki anak) ketika terbuka (expand), yang kedua closeIcon yaitu icon yang digunakan oleh direktory ketika tertutup (collapse) dan yang terakhir adalah leafIcon yaitu icon yang digunakan oleh node yang tak memiliki anak.

Leaf Icon

Sebenarnya icon yang digunakan oleh leaf tak harus sama, kita bisa merubahnya sesuai dengan yang kita inginkan contohnya sekarang kita akan membuat 2 buat leaf icon. Kita ambil contoh untuk kode diatas namun menggunakan Cell Render yang berbeda.

FileRenderLeaf.java

```
package pelajaran11;

import java.awt.Component;
import java.io.File;
import javax.swing.ImageIcon;
import javax.swing.JTree;
import javax.swing.tree.DefaultTreeCellRenderer;

/**
 * @author usu
 */
public class FileRenderLeaf extends DefaultTreeCellRenderer {

    public FileRenderLeaf() {
        super();
    }

    @Override
    public Component getTreeCellRendererComponent(final JTree tree, final
```

```
Object value,
        final boolean sel, final boolean expanded, final boolean leaf,
final int row,
        final boolean hasFocus) {

    final File file = new File(value.toString());
    setText(file.getName());
    setOpenIcon(new
ImageIcon(getClass().getResource("/pelajaran11/folder.png")));
    setClosedIcon(getOpenIcon());

    if (file.toString().toUpperCase().endsWith(".JAVA")) {
        setLeafIcon(new
ImageIcon(getClass().getResource("/pelajaran11/unknow.png")));
    } else {
        setLeafIcon(new
ImageIcon(getClass().getResource("/pelajaran11/file.png")));
    }

    return super.getTreeCellRendererComponent(tree, value, sel, expanded,
leaf, row, hasFocus);
}
}
```

Sekarang kita gunakan kode JTree sebelumnya namun mengganti cell rendernya.

TreeFileLeaf.java

```
package pelajaran11;

import java.awt.BorderLayout;
import java.io.File;
import javax.swing.JScrollPane;
import javax.swing.JTree;
import javax.swing.WindowConstants;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.DefaultTreeModel;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TreeFileLeaf extends JTree {

    public static final void insertNode(final DefaultMutableTreeNode node,
final File folder) {
        for (final File f : folder.listFiles()) {
            final DefaultMutableTreeNode child = new
DefaultMutableTreeNode(f);
            if (f.isDirectory()) {
                TreeFileLeaf.insertNode(child, f);
            }
            node.add(child);
        }
    }

    public static void main(final String[] usu) {
        final Frame frame = new Frame();
        frame.setLayout(new BorderLayout());
```

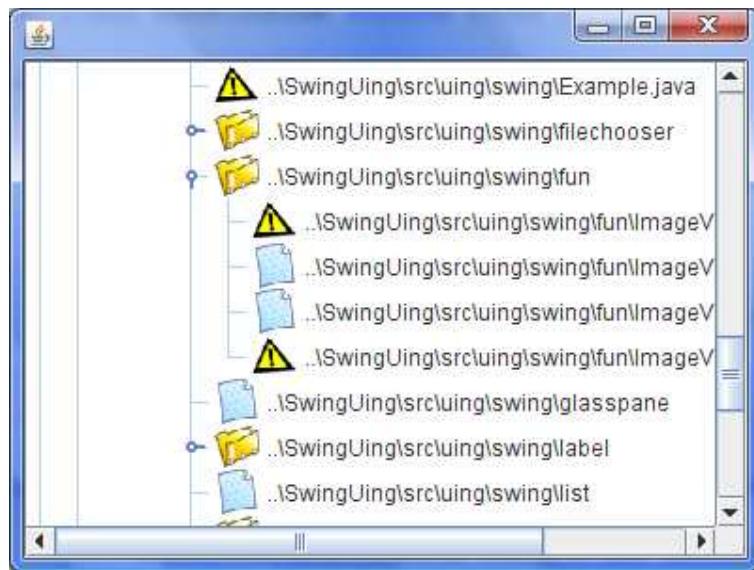
```
frame.add(new JScrollPane(new TreeFileLeaf()));
frame.setSize(400, 300);
frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
frame.setVisible(true);
}

public TreeFileLeaf() {
    super();

    final DefaultMutableTreeNode root = new DefaultMutableTreeNode(new File(".."));

    TreeFileLeaf.insertNode(root, new File(.."));

    final DefaultTreeModel model = new DefaultTreeModel(root);
    setModel(model);
    setCellRenderer(new FileRenderLeaf());
}
}
```



Gambar 100 FileRenderLeaf.java

Saya melakukan perbedaan icon berdasarkan jenis file. Jadi anda bisa melakukan perbedaan leaf icon menjadi berapa macam sesuai yang anda inginkan, selama anda sanggup berlama-lama coding, hahaha ☺

Color

Sekarang kita coba membuat render yang merepresentasikan nilai Color. Sehingga kita hapus saja penggunaan icon dalam rendernya.

ColorRender.java

```
package pelajaran11;

import java.awt.Color;
import java.awt.Component;
```

```
import javax.swing.JLabel;
import javax.swing.JTree;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.DefaultTreeCellRenderer;

/**
 * @author usu
 */
public class ColorRender extends DefaultTreeCellRenderer {

    @Override
    public Component getTreeCellRendererComponent(final JTree tree, final
Object value,
        final boolean selected, final boolean expanded, final boolean
leaf, final int row,
        final boolean hasFocus) {
        if (leaf) {
            final Color warna = (Color) ((DefaultMutableTreeNode)
value).getUserObject();

            final JLabel label = new JLabel();
            label.setOpaque(true);
            label.setBackground(warna);
            label.setText("Red:" + warna.getRed() + ", Green:" +
warna.getGreen() + ", Blue:"
                + warna.getBlue() + ", Alpha:" + warna.getAlpha());

            return label;
        } else {
            return super
                .getTreeCellRendererComponent(tree, value, leaf, expanded,
leaf, row, hasFocus);
        }
    }
}
```

Sekarang kita buat JTreenya

TreeColor.java

```
package pelajaran11;

import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.JScrollPane;
import javax.swing.JTree;
import javax.swing.WindowConstants;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.DefaultTreeModel;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TreeColor extends JTree {

    public static void main(final String[] usu) {
        final Frame frame = new Frame();
        frame.setLayout(new BorderLayout());

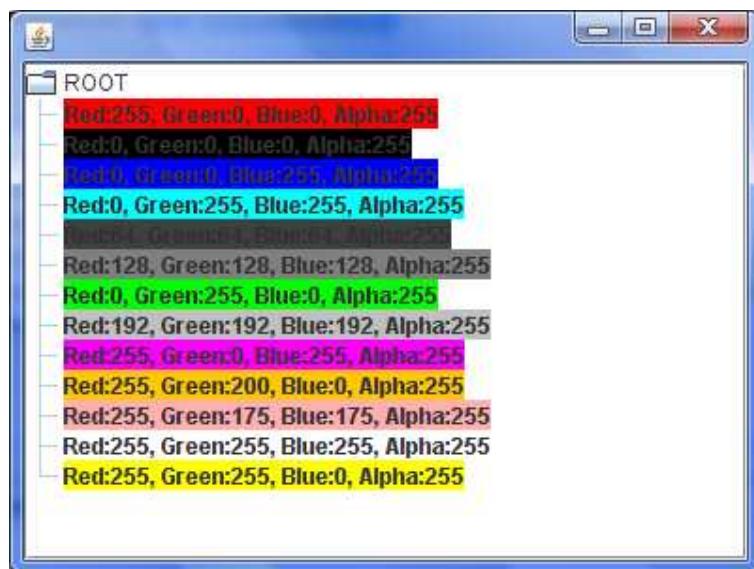
```

```
frame.add(new JScrollPane(new TreeColor()));
frame.setSize(400, 300);
frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
frame.setVisible(true);
}

public TreeColor() {
    super();

    final DefaultMutableTreeNode root = new
DefaultMutableTreeNode("ROOT");
    root.add(new DefaultMutableTreeNode(Color.RED));
    root.add(new DefaultMutableTreeNode(Color.BLACK));
    root.add(new DefaultMutableTreeNode(Color.BLUE));
    root.add(new DefaultMutableTreeNode(Color.CYAN));
    root.add(new DefaultMutableTreeNode(Color.DARK_GRAY));
    root.add(new DefaultMutableTreeNode(Color.GRAY));
    root.add(new DefaultMutableTreeNode(Color.GREEN));
    root.add(new DefaultMutableTreeNode(Color.LIGHT_GRAY));
    root.add(new DefaultMutableTreeNode(Color.MAGENTA));
    root.add(new DefaultMutableTreeNode(Color.ORANGE));
    root.add(new DefaultMutableTreeNode(Color.PINK));
    root.add(new DefaultMutableTreeNode(Color.WHITE));
    root.add(new DefaultMutableTreeNode(Color.YELLOW));

    setModel(new DefaultTreeModel(root));
    setCellRenderer(new ColorRender());
}
}
```



Gambar 101 TreeColor.java

Background

Selain merubah render kita juga jangan lupa dengan metode paintComponent atau JViewport. Karena sekarang kita akan bahas lagi soal hal itu. Kalo anda lupa wah mending balik lagi sana ke bab JViewport !!!

TreeBackground.java

```
package pelajaran11;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTree;
import javax.swing.JViewport;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;

/**
 *
 * @author usu
 */
public class TreeBackground extends JFrame {

    public TreeBackground() {
        super();

        JViewport port = new JViewport() {

            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);

                GradientPaint paint = new GradientPaint(0, 0, Color.GREEN,
0, getHeight(), Color.YELLOW);

                Graphics2D g2 = (Graphics2D) g.create();
                g2.setPaint(paint);
                g2.fillRect(0, 0, getWidth(), getHeight());
            }
        };

        JTree tree = new JTree();
        tree.setOpaque(false);
        port.setView(tree);

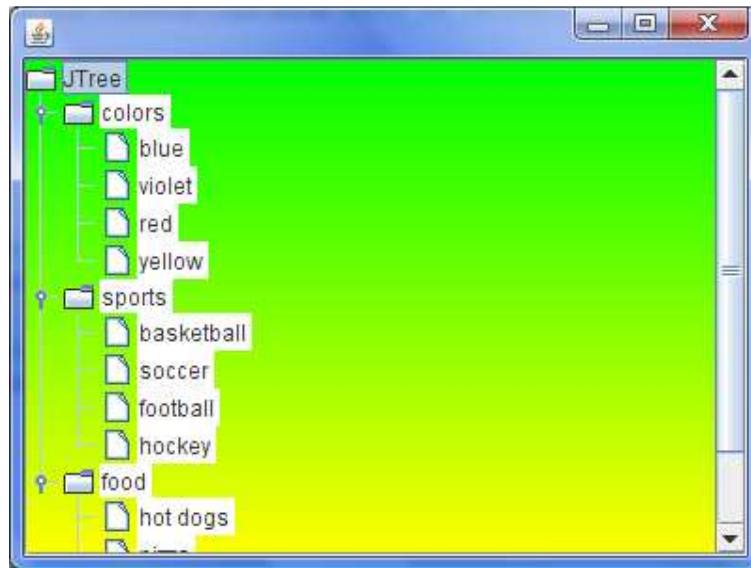
        JScrollPane pane = new JScrollPane();
        pane.setViewport(port);

        setLayout(new BorderLayout());
        add(pane);
        setSize(400, 300);
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    }

    public static void main(String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                new TreeBackground().setVisible(true);
            }
        });
    }
}
```

```
}
```



Gambar 102 TreeBackground.java

Wah sepertinya ada sedikit gangguan, bagai mana kalo kita buat CellRendernya transparan.

```
JTree tree = new JTree();
tree.setOpaque(false);
tree.setCellRenderer(new DefaultTreeCellRenderer() {

    @Override
    public Component getTreeCellRendererComponent(JTree tree, Object value, boolean sel,
boolean expanded, boolean leaf, int row, boolean hasFocus) {
        super.getTreeCellRendererComponent(tree, value, sel, expanded, leaf, row, hasFocus);
        JLabel label = new JLabel(value.toString());
        label.setOpaque(false);
        if (leaf) {
            label.setIcon(getLeafIcon());
        } else if (expanded) {
            label.setIcon(getOpenIcon());
        } else {
            label.setIcon(getClosedIcon());
        }
        return label;
    }
});
```

Atau lengkapnya seperti ini...

TreeBackground.java

```
package pelajaran11;
```

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Component;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JScrollPane;
import javax.swing.JTree;
import javax.swing.JViewport;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.tree.DefaultTreeCellRenderer;

/**
 * @author usu
 */
public class TreeBackground extends JFrame {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                new TreeBackground().setVisible(true);
            }
        });
    }

    public TreeBackground() {
        super();

        final JViewport port = new JViewport() {

            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);

                GradientPaint paint = new GradientPaint(0, 0, Color.GREEN, 0,
                getHeight(), Color.YELLOW);

                Graphics2D g2 = (Graphics2D) g.create();
                g2.setPaint(paint);
                g2.fillRect(0, 0, getWidth(), getHeight());
            }
        };

        final JTree tree = new JTree();
        tree.setOpaque(false);
        tree.setCellRenderer(new DefaultTreeCellRenderer() {

            @Override
            public Component getTreeCellRendererComponent(final JTree tree,
final Object value,
final boolean sel, final boolean expanded, final boolean
leaf, final int row,
final boolean hasFocus) {
                super.getTreeCellRendererComponent(tree, value, sel, expanded,
leaf, row, hasFocus);
                final JLabel label = new JLabel(value.toString());
                label.setOpaque(false);
            }
        });
    }
}
```

```
        if (sel) {
            label.setForeground(Color.RED);
        }
        if (leaf) {
            label.setIcon(getLeafIcon());
        } else if (expanded) {
            label.setIcon(getOpenIcon());
        } else {
            label.setIcon(getClosedIcon());
        }
        return label;
    }
});
port.setView(tree);

final JScrollPane pane = new JScrollPane();
pane.setViewport(port);

setLayout(new BorderLayout());
add(pane);
setSize(400, 300);
setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
}
}
```



Gambar 103 TreeBackground.java

Glass

Pada pembahasan JViewport kita sudah mencoba membuat Glass diatas JTextArea, sekarang kita akan mencoba membuat Glass diatas JTree namun dengan format yang berbeda seperti sebelumnya. Idenya kita mau buat gambar transparan diatas JTree, misal gambar pemandangan atau apalah terserah anda!!!

Masih ingat kan cara menggambar glass dalam JViewport? Betul, kita harus mengoverride paintChildren().

TreeGlass.java

```
package pelajaran11;

import java.awt.AlphaComposite;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Component;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JScrollPane;
import javax.swing.JTree;
import javax.swing.JViewport;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.tree.DefaultTreeCellRenderer;

/**
 * @author usu
 */
public class TreeGlass extends JFrame {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                new TreeGlass().setVisible(true);
            }
        });
    }

    public TreeGlass() {
        super();

        final JTree tree = new JTree();
        tree.setOpaque(false);
        tree.setCellRenderer(new DefaultTreeCellRenderer() {

            @Override
            public Component getTreeCellRendererComponent(final JTree tree,
final Object value,
final boolean sel, final boolean expanded, final boolean
leaf, final int row,
final boolean hasFocus) {
                super.getTreeCellRendererComponent(tree, value, sel, expanded,
leaf, row, hasFocus);
                final JLabel label = new JLabel(value.toString());
                label.setOpaque(false);
                if (sel) {
                    label.setForeground(Color.RED);
                }
                if (leaf) {
                    label.setIcon(getLeafIcon());
                } else if (expanded) {
                    label.setIcon(getOpenIcon());
                } else {

```

```
        label.setIcon(getClosedIcon());
    }
    return label;
}
);

final JViewport port = new JViewport() {

    @Override
    protected void paintChildren(Graphics g) {
        super.paintChildren(g);

        Image image = new
ImageIcon(getClass().getResource("/pelajaran9/image.jpg")).getImage();

        Graphics2D g2 = (Graphics2D) g.create();
        g2.setComposite(AlphaComposite.SrcOver.derive(0.3F));

        for (int i = 0; i < getWidth(); i += image.getWidth(null)) {
            for (int j = 0; j < getHeight(); j += image.getHeight(null))
{
                g2.drawImage(image, i, j, null);
            }
        }
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);

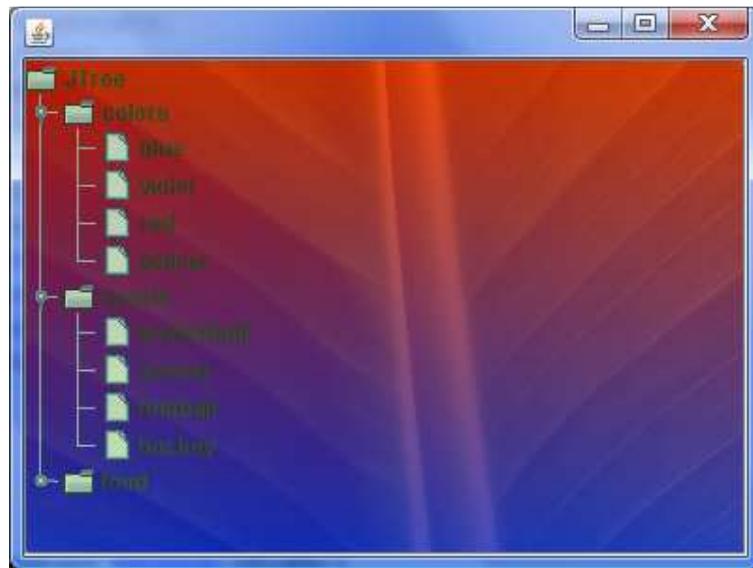
        GradientPaint paint = new GradientPaint(0, 0, Color.RED, 0,
getHeight(), Color.BLUE);

        Graphics2D g2 = (Graphics2D) g.create();
        g2.setPaint(paint);
        g2.fillRect(0, 0, getWidth(), getHeight());
    }
};

port.setView(tree);

final JScrollPane pane = new JScrollPane();
pane.setViewport(port);

setLayout(new BorderLayout());
add(pane);
setSize(400, 300);
setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
}
}
```



Gambar 104 TreeGlass.java

Editor

Selain render ada juga satu fasilitas yang dimiliki JTree yaitu Editor. Editor ini berfungsi untuk mengedit data yang ditampung oleh JTree. Tapi secara default sebuah JTree tak dapat diedit, sehingga agar bisa diedit kita harus memanggil metode setEditable(true). Untuk menset editor ke JTree gunakan :

```
JTree tree = new JTree();
tree.setCellEditor(TreeCellEditor editor);
```

TreeEdited.java

```
package pelajaran11;

import java.awt.BorderLayout;
import javax.swing.JScrollPane;
import javax.swing.JTree;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TreeEdited extends JTree {

    public static void main(final String[] usu) {
        final Frame frame = new Frame();
        frame.setLayout(new BorderLayout());
        frame.add(new JScrollPane(new TreeEdited()));
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

```
public TreeEdited() {
    super();
    setEditable(true);
}
```



Gambar 105 TreeEdited.java

Dan biasanya secara default editor yang digunakan oleh JTree adalah berupa JTextField. Namun yang menjadi kendala, bagaimana jika yang perlukan adalah nilai boolean sehingga tak mungkin jika kita menggunakan editor JTextField, karena yang lebih baik adalah menggunakan editor JCheckBox. Nah sekarang kita akan bahas tentang persoalan itu.

JComboBox

Sekarang misal kita ingin membuat editor dengan pilihan misalnya nilai yang dimiliki oleh JTree harus "Eko", "Kurniawan", atau "Khannedy". Nah hal ini tak baik jika menggunakan editor default oleh karena itu kita lebih baik menggunakan editor JComboBox.

Caranya adalah dengan menggunakan DefaultCellEditor.

```
DefaultCellEditor editor = new DefaultCellEditor(new JComboBox(new String[]{
    "Eko", "Kurniawan", "Khannedy"
}));
```

Nah sekarang tinggal menggunakan nya dalam JTree.

TreeEditedComboBox.java

```
package pelajaran11;

import java.awt.BorderLayout;
import javax.swing.DefaultCellEditor;
import javax.swing.JComboBox;
```

```
import javax.swing.JScrollPane;
import javax.swing.JTree;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TreeEditedComboBox extends JTree {

    public static void main(final String[] usu) {
        final Frame frame = new Frame();
        frame.setLayout(new BorderLayout());
        frame.add(new JScrollPane(new TreeEditedComboBox()));
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }

    public TreeEditedComboBox() {
        super();
        setEditable(true);
        final DefaultCellEditor editor = new DefaultCellEditor(new
JComboBox(new String[] { "Eko",
        "Kurniawan", "Khannedy" }));
        setCellEditor(editor);
    }
}
```



Gambar 106 TreeEditedComboBox.java

Ternyata gampang dan mengasikkan kan?

JChekBox

Nah sekarang bagaimana jika kita ingin menggunakan JChekBox sebagai editornya. Tapi sebelum menggunakan kita harus tahu dulu kalo JChekBox hanya dapat mengembalikan nilai true atau false. Jadi penggunaannya harus untuk JTree yang menampung data boolean.

Penggunaannya :

DefaultCellEditor editor = new DefaultCellEditor(new JCheckBox());

TreeEditedChekBox.java

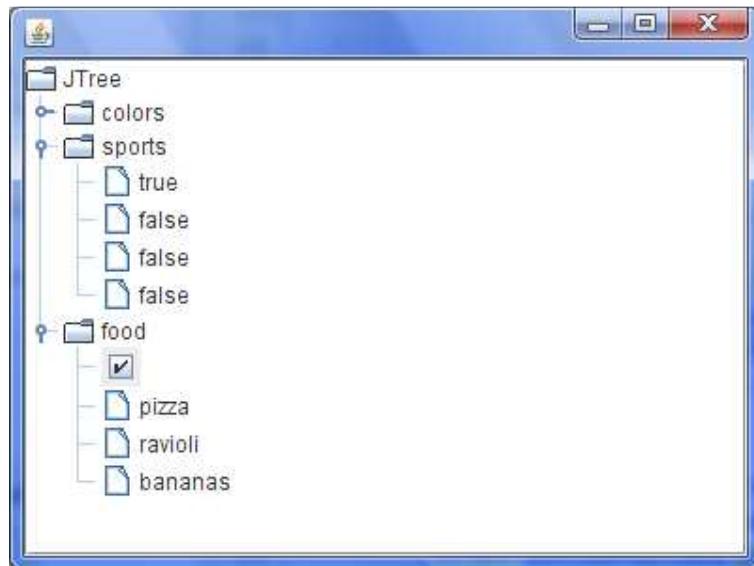
```
package pelajaran11;

import java.awt.BorderLayout;
import javax.swing.DefaultCellEditor;
import javax.swing.JCheckBox;
import javax.swing.JScrollPane;
import javax.swing.JTree;
import javax.swing.WindowConstants;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TreeEditedChekBox extends JTree {

    public static void main(final String[] usu) {
        final Frame frame = new Frame();
        frame.setLayout(new BorderLayout());
        frame.add(new JScrollPane(new TreeEditedChekBox()));
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }

    public TreeEditedChekBox() {
        super();
        setEditable(true);
        final DefaultCellEditor editor = new DefaultCellEditor(new
JCheckBox());
        editor.setClickCountToStart(2);
        setCellEditor(editor);
    }
}
```



Gambar 107 TreeEditedChekBox.java

Tapi alangkah baiknya anda juga menggunakan render JCheckBox, agar editor dan render terlihat sama.

Color

DefaultCellEditor hanya bisa menampung editor JComboBox, JCheckBox dan JTextField. Sehingga jika kita ingin membuat editor untuk Color kita tak bisa menggunakan DefaultCellEditor, oleh karena itu kita harus membuat editor secara manual. Tapi tenang gak susah kok, hampir mirip seperti membuat render.

ColorEditor.java

```
package pelajaran11;

import java.awt.Color;
import java.awt.Component;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.AbstractCellEditor;
import javax.swing.JColorChooser;
import javax.swing.JLabel;
import javax.swing.JTree;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.TreeCellEditor;

/**
 * @author usu
 */
public class ColorEditor extends AbstractCellEditor implements TreeCellEditor {

    private final JLabel label;
    private JTtree tree;
    private Color warna;

    public ColorEditor() {
```

```
super();
this.label = new JLabel();
this.label.addMouseListener(new MouseAdapter() {

    @Override
    public void mouseClicked(MouseEvent e) {
        if (e.getClickCount() >= 2) {
            final Color temp =
JColorChooser.showDialog(ColorEditor.this.tree, "Choose Color",
                        ColorEditor.this.warna);
            if (temp != null) {
                ColorEditor.this.warna = temp;
            }
            stopCellEditing();
        }
    }
} );

public Object getCellEditorValue() {
    return this.warna;
}

public Component getTreeCellEditorComponent(JTree tree, final Object value,
    final boolean isSelected, final boolean expanded, final boolean leaf, final int row) {
    this.warna = (Color) ((DefaultMutableTreeNode)
value).getUserObject();
    this.label.setOpaque(true);
    this.label.setBackground(this.warna);
    this.label.setText("Red:" + this.warna.getRed() + ", Green:" +
this.warna.getGreen()
        + ", Blue:" + this.warna.getBlue() + ", Alpha:" +
this.warna.getAlpha());
    return this.label;
}
}
```

TreeColorEditor.java

```
package pelajaran11;

import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.JScrollPane;
import javax.swing.JTree;
import javax.swing.WindowConstants;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.DefaultTreeModel;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TreeColorEditor extends JTree {

    public static void main(String[] usu) {
        final Frame frame = new Frame();
```

```
frame.setLayout(new BorderLayout());
frame.add(new JScrollPane(new TreeColorEditor()));
frame.setSize(400, 300);
frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
frame.setVisible(true);
}

public TreeColorEditor() {
    super();

    final DefaultMutableTreeNode root = new
DefaultMutableTreeNode("ROOT");
    root.add(new DefaultMutableTreeNode(Color.RED));
    root.add(new DefaultMutableTreeNode(Color.BLACK));
    root.add(new DefaultMutableTreeNode(Color.BLUE));
    root.add(new DefaultMutableTreeNode(Color.CYAN));
    root.add(new DefaultMutableTreeNode(Color.DARK_GRAY));
    root.add(new DefaultMutableTreeNode(Color.GRAY));
    root.add(new DefaultMutableTreeNode(Color.GREEN));
    root.add(new DefaultMutableTreeNode(Color.LIGHT_GRAY));
    root.add(new DefaultMutableTreeNode(Color.MAGENTA));
    root.add(new DefaultMutableTreeNode(Color.ORANGE));
    root.add(new DefaultMutableTreeNode(Color.PINK));
    root.add(new DefaultMutableTreeNode(Color.WHITE));
    root.add(new DefaultMutableTreeNode(Color.YELLOW));

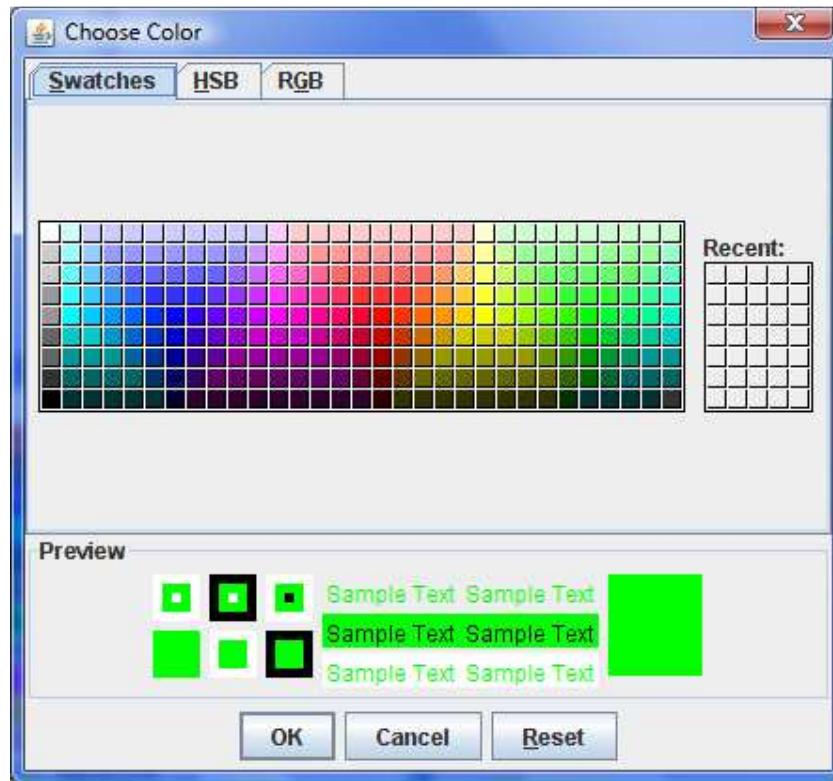
    setModel(new DefaultTreeModel(root));

    setCellRenderer(new ColorRender());
    setCellEditor(new ColorEditor());
    setEditable(true);
}
}
```



Gambar 108 TreeColorEditor.java

Jika anda mendouble klik salah satu node, maka akan keluar dialog color chooser :



Gambar 109 Color Editor

Jadi seperti itulah color editor. Sebenarnya tak harus seperti itu, karena memang tergantung gaya anda sendiri, tapi untuk yang paling simple adalah seperti itu.

File

Tadi pertama kali membuat render kita membuat render file, nah sekarang saatnya kita membuat sebuah Editor untuk File.

FileChooser.java

```
package pelajaran11;

import java.awt.Component;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.io.File;
import javax.swing.AbstractCellEditor;
import javax.swing.JFileChooser;
import javax.swing.JLabel;
import javax.swing.JTree;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.TreeCellEditor;

/**
 * @author usu
 */
public class FileEditor extends AbstractCellEditor implements TreeCellEditor {
```

```
private final JFileChooser chooser;
private File file;
private final JLabel label;
private JTree tree;

public FileEditor() {
    super();
    this.chooser = new JFileChooser();
    this.label = new JLabel();
    this.label.addMouseListener(new MouseAdapter() {

        @Override
        public void mouseClicked(MouseEvent e) {
            if (e.getClickCount() >= 2) {
                if
(FileEditor.this.chooser.showOpenDialog(FileEditor.this.tree) ==
JFileChooser.APPROVE_OPTION) {
                    FileEditor.this.file =
FileEditor.this.chooser.getSelectedFile();
                    stopCellEditing();
                } else {
                    cancelCellEditing();
                }
            }
        }
    );
}

public Object getCellEditorValue() {
    return this.file;
}

public Component getTreeCellEditorComponent(JTree tree, final
Object value,
    final boolean isSelected, final boolean expanded, final boolean
leaf, final int row) {
    this.file = (File) ((DefaultMutableTreeNode) value).getUserObject();
    this.tree = tree;
    this.label.setText(this.file.getName());
    this.label.setIcon(((JLabel) tree.getCellRenderer()).getIcon());
    return this.label;
}
}
```

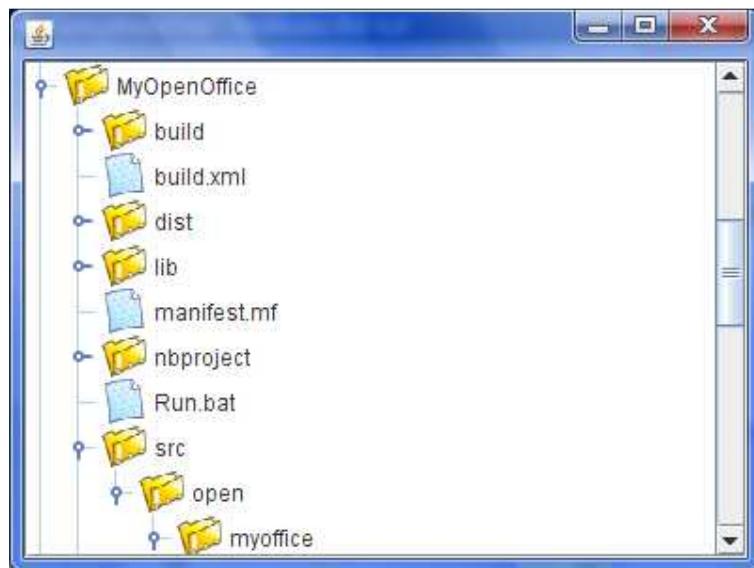
Sekarang buka kembali file FileRender.java karena akan kita gunakan lagi. Dan ayo sekarang buat JTree untuk FileEditor.

TreeFileEditor.java

```
package pelajaran11;

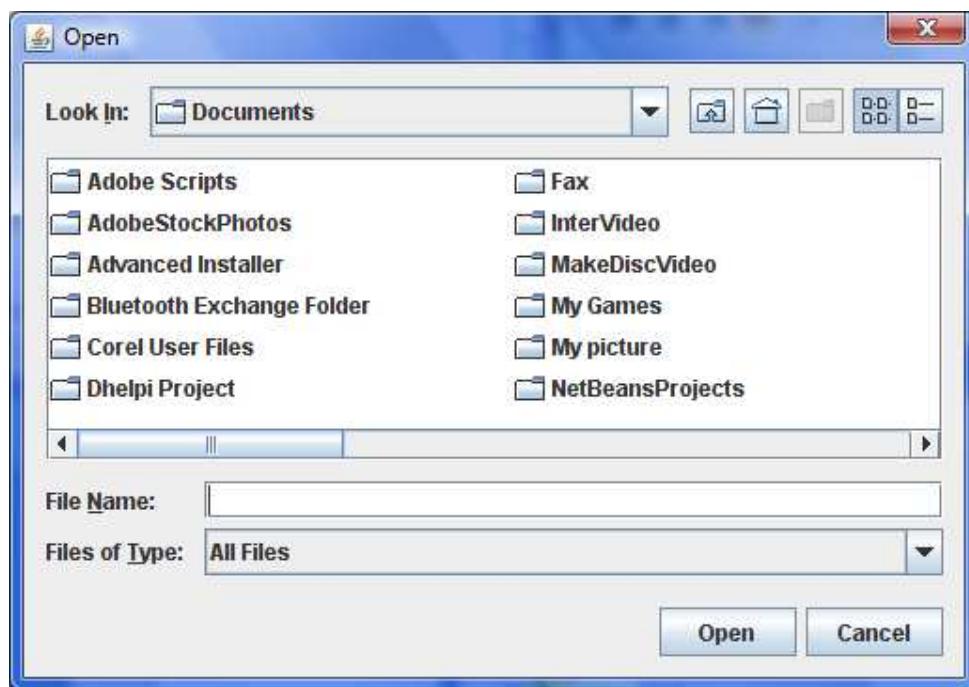
import java.awt.BorderLayout;
import java.io.File;
import javax.swing.JScrollPane;
import javax.swing.JTree;
import javax.swing.WindowConstants;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.DefaultTreeModel;
import pelajaran1.Frame;
```

```
/**  
 * @author usu  
 */  
public class TreeFileEditor extends JTree {  
  
    public static final void insertNode(final DefaultMutableTreeNode node,  
final File folder) {  
        for (final File f : folder.listFiles()) {  
            final DefaultMutableTreeNode child = new  
DefaultMutableTreeNode(f);  
            if (f.isDirectory()) {  
                TreeFileEditor.insertNode(child, f);  
            }  
            node.add(child);  
        }  
    }  
  
    public static void main(final String[] usu) {  
        final Frame frame = new Frame();  
        frame.setLayout(new BorderLayout());  
        frame.add(new JScrollPane(new TreeFileEditor()));  
        frame.setSize(400, 300);  
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);  
        frame.setVisible(true);  
    }  
  
    public TreeFileEditor() {  
        super();  
  
        final DefaultMutableTreeNode root = new DefaultMutableTreeNode(new  
File("."));  
  
        TreeFileEditor.insertNode(root, new File("."));  
  
        final DefaultTreeModel model = new DefaultTreeModel(root);  
        setModel(model);  
        setCellRenderer(new FileRender());  
        setCellEditor(new FileEditor());  
        setEditable(true);  
    }  
}
```



Gambar 110 TreeFileEditor.java

Dan jika anda double klik salah satu nodenya maka akan keluar FileEditornya seperti yang terlihat dibawah ini :



Gambar 111 File Editor

Yes No Editor

Apatuh "yes no editor"? Maksud saya adalah sebuah editor yang memiliki tombol Yes dan No atau bisa diganti Cancel. Ketika tombol yes ditekan maka perubahan akan di simpan sedangkan jika tombol no di tekan maka perubahan akan dibatalkan. Dan saat ini anda akan melihat kehebatan JWindow...

FileEditorYesNo.java

```
package pelajaran11;

import java.awt.Component;
import java.awt.GridLayout;
import java.awt.Point;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.io.File;
import javax.swing.AbstractCellEditor;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JLabel;
import javax.swing.JTree;
import javax.swing.JWindow;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.TreeCellEditor;

/**
 * @author usu
 */
public class FileEditorYesNo extends AbstractCellEditor implements
TreeCellEditor {

    private final JFileChooser chooser;
    private File file;
    private final JLabel label;
    private JTree tree;
    private final JWindow window;
    private final JButton yes, no;

    public FileEditorYesNo() {
        super();
        this.chooser = new JFileChooser();

        this.window = new JWindow();
        this.window.setLayout(new GridLayout(1, 2));
        this.window.setAlwaysOnTop(true);

        this.yes = new JButton(" OK ");
        this.yes.addActionListener(new ActionListener() {

            public void actionPerformed(final ActionEvent e) {
                stopCellEditing();
            }
        });

        this.no = new JButton(" NO ");
        this.no.addActionListener(new ActionListener() {

            public void actionPerformed(final ActionEvent e) {
                cancelCellEditing();
            }
        });

        this.window.add(this.yes);
        this.window.add(this.no);
        this.window.pack();
    }

    protected void stopCellEditing() {
        cancelCellEditing();
    }

    protected void cancelCellEditing() {
        if (file != null) {
            label.setText(file.getName());
        } else {
            label.setText("Select file");
        }
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == this.yes) {
            stopCellEditing();
        } else if (e.getSource() == this.no) {
            cancelCellEditing();
        }
    }
}
```

```
this.label = new JLabel();
this.label.addMouseListener(new MouseAdapter() {

    @Override
    public void mouseClicked(MouseEvent e) {
        if (e.getClickCount() >= 2) {
            final Point p =
FileEditorYesNo.this.label.getLocationOnScreen();
            p.y += FileEditorYesNo.this.label.getHeight();
            FileEditorYesNo.this.window.setLocation(p);
            FileEditorYesNo.this.window.setVisible(true);
            if
(FileEditorYesNo.this.chooser.showOpenDialog(FileEditorYesNo.this.tree) ==
JFileChooser.APPROVE_OPTION) {
                FileEditorYesNo.this.file =
FileEditorYesNo.this.chooser.getSelectedFile();

FileEditorYesNo.this.label.setText(FileEditorYesNo.this.file.getName());
            }
        }
    }
);

@Override
public void cancelCellEditing() {
    this.window.setVisible(false);
    super.cancelCellEditing();
}

public Object getCellEditorValue() {
    return this.file;
}

public Component getTreeCellEditorComponent(JTree tree, Object value,
final boolean isSelected, final boolean expanded, final boolean leaf, final int row) {
    this.file = ((DefaultMutableTreeNode) value).getUserObject();
    this.tree = tree;
    this.label.setText(this.file.getName());
    this.label.setIcon(((JLabel) tree.getCellRenderer()).getIcon());
    return this.label;
}

@Override
public boolean stopCellEditing() {
    this.window.setVisible(false);
    return super.stopCellEditing();
}
}
```

Sekarang kita buat JTree nya dan masih seperti tadi gunakan FileRender sebagai rendernya.

TreeFileEditorYesNo.java

```
package pelajaran11;
```

```
import java.awt.BorderLayout;
import java.io.File;
import javax.swing.JScrollPane;
import javax.swing.JTree;
import javax.swing.WindowConstants;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.DefaultTreeModel;
import pelajaran1.Frame;

/*
 * @author usu
 */
public class TreeFileEditorYesNo extends JTree {

    public static final void insertNode(final DefaultMutableTreeNode node,
final File folder) {
        for (final File f : folder.listFiles()) {
            final DefaultMutableTreeNode child = new
DefaultMutableTreeNode(f);
            if (f.isDirectory()) {
                TreeFileEditorYesNo.insertNode(child, f);
            }
            node.add(child);
        }
    }

    public static void main(final String[] usu) {
        final Frame frame = new Frame();
        frame.setLayout(new BorderLayout());
        frame.add(new JScrollPane(new TreeFileEditorYesNo()));
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }

    public TreeFileEditorYesNo() {
        super();

        final DefaultMutableTreeNode root = new DefaultMutableTreeNode(new
File("."));

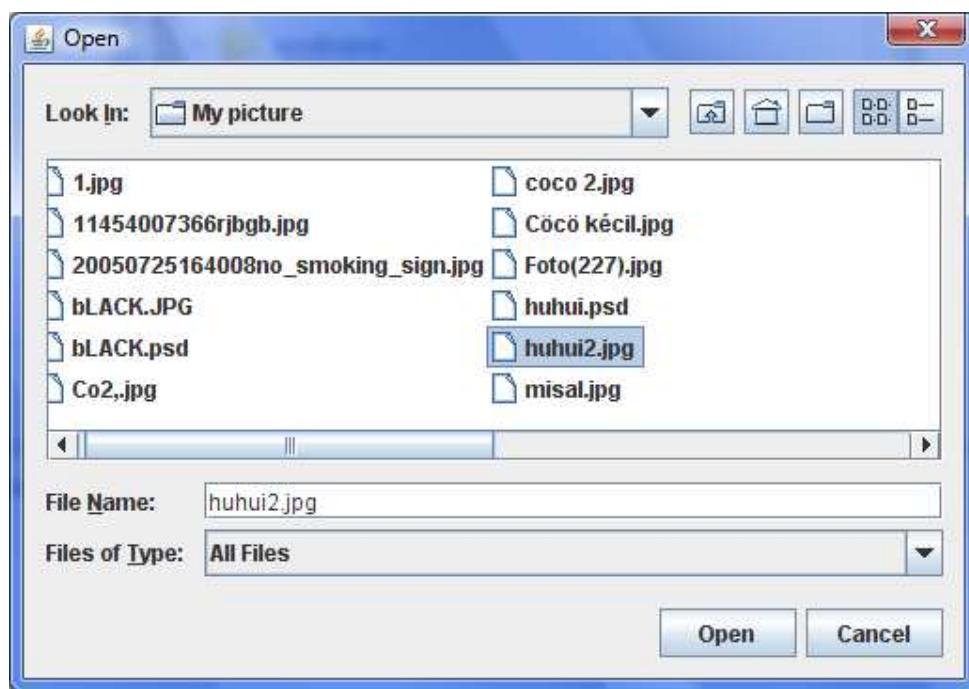
        TreeFileEditorYesNo.insertNode(root, new File("."));

        final DefaultTreeModel model = new DefaultTreeModel(root);
        setModel(model);
        setCellRenderer(new FileRender());
        setCellEditor(new FileEditorYesNo());
        setEditable(true);
    }
}
```



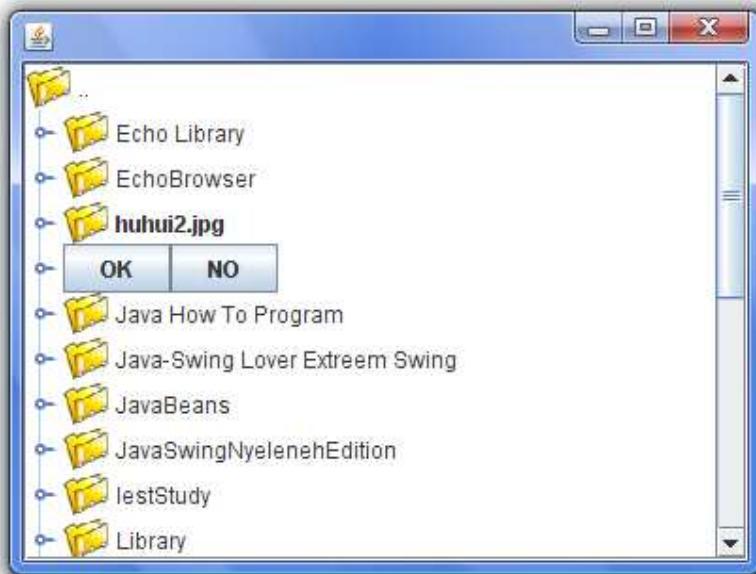
Gambar 112 TreeFileEditorYesNo.java

Jika anda mendouble klik sebuah node maka akan keluar FileEditornya.



Gambar 113 File Editor

Lalu anda pilih file baru.



Gambar 114 Yes No Editor

Maka akan terlihat tombol OK dan NO dibawah node yang telah anda edit, dan jika setuju anda klik OK dan jika tidak anda klik NO, misal saya klik NO, maka node akan kembali seperti semula.



Gambar 115 TreeFileEditorYesNo saat memilih NO

Tapi jika saya menekan tombol OK maka perubahan akan disimpan pada node JTree yang telah anda edit.



Gambar 116 TreeFileEditorYesNo saat memilih YES

Sekarang anda tahu kan kalo JWindow itu sangat berguna sekali. Selain sebelumnya kita jelaskan seperti Splash Screen, About Screen, Window Tip, ternyata JWindow juga bisa digunakan sebagai pembantu dalam JTree. Dibab selanjutnya juga JWindow akan diperlukan, khususnya dalam JTable.

Kesimpulan

Bingung nih narik kesimpulannya. Yang pasti sich “EXPLORE CREATIVITAS ANDA” salah satunya penggunaan JWindow dalam JTree Editor adalah hasil renungan saya di depan laptop, hahaha...

Pelajaran 12

JTable

Kalo ngomong soal tabel, pasti sudah tak asing lagi dikalangan para Swinger, karena memang kebanyakan user berikut di JTable. Kenapa? Selain JTable adalah komponen swing yang paling kompleks dalam artian yang paling sulit, JTable juga sangat erat kaitannya dengan database.

Tapi dalam buku ini saya tak akan mengajak anda berpusing ria, tapi mengajak anda mengexploitasi daya pikir anda untuk membuat JTable yang sangan menarik, sehingga end user nyaman dan senang berlama-lama memandang aplikasi yang kita buat.

Render

Hampir sama seperti JTree, JTable juga kebanyakan akan kita bahas dalam Render dan Editor. Jadi jika anda sudah paham dengan penjelasan di JTree tadi, saya jamin pasti anda juga akan lulus dengan mudah dalam penjelasan JTable kali ini.

Untuk menset render dalam JTable kita gunakan kode dibawah ini :

JTable.getColumnModel().getColumn(int index).setCellRenderer(TableCellRenderer render);

Color

Lagi – lagi Color ☺, yach itung – itung pemanasan dulu, hahaha.

RenderColor.java

```
package pelajaran12;

import java.awt.Color;
import java.awt.Component;
import javax.swing.JLabel;
import javax.swing.JTable;
import javax.swing.table.TableCellRenderer;

/**
 * @author usu
 */
public class RenderColor implements TableCellRenderer {

    public Component getTableCellRendererComponent(final JTable table, final
Object value,
    final boolean isSelected, final boolean hasFocus, final int row,
final int column) {
        final Color warna = (Color) value;
        final JLabel label = new JLabel();
        label.setOpaque(true);
        if (isSelected) {
            label.setBackground(table.getSelectionBackground());
            label.setForeground(warna);
        } else {
            label.setBackground(warna);
            label.setForeground(table.getForeground());
        }
    }
}
```

```
        }
        label.setText("Red:" + warna.getRed() + ", Green:" + warna.getGreen()
+ ", Blue:"
            + warna.getBlue() + ", Alpha:" + warna.getAlpha());
        return label;
    }
}
```

Setelah buat Rendernya sekarang tinngal buat JTablenya.

TableColor.java

```
package pelajaran12;

import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;
import pelajaran1.Frame;

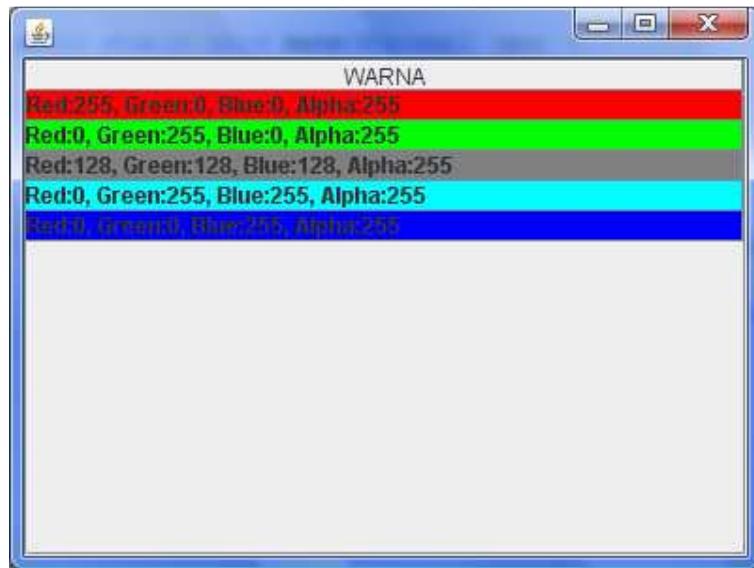
/**
 * @author usu
 */
public class TableColor extends JTable {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setLayout(new BorderLayout());
                frame.add(new JScrollPane(new TableColor()));
                frame.pack();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setSize(400, 300);
                frame.setVisible(true);
            }
        });
    }

    public TableColor() {
        super();

        final DefaultTableModel model = new DefaultTableModel();
        model.addColumn("WARNA");
        model.addRow(new Object[] { Color.RED });
        model.addRow(new Object[] { Color.GREEN });
        model.addRow(new Object[] { Color.GRAY });
        model.addRow(new Object[] { Color.CYAN });
        model.addRow(new Object[] { Color.BLUE });
        setModel(model);
        getColumnModel().getColumn(0).setCellRenderer(new RenderColor());
    }
}
```



Gambar 117 TableColor.java

Gradient

Kali ini kita akan membuat efek gradient yang berkesinambungan untuk JTable sehingga latar cell terlihat lebih indah dipandang. Puitis banget bahasanya?

GradientRender.java

```
package pelajaran12;

import java.awt.Color;
import java.awt.Component;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import javax.swing.JLabel;
import javax.swing.JTable;
import javax.swing.table.TableCellRenderer;

public class GradientRender implements TableCellRenderer {

    public Component getTableCellRendererComponent(final JTable table, final
Object value,
            final boolean isSelected, final boolean hasFocus, final int row,
final int column) {
        JLabel label = null;
        if (row % 2 == 0) {
            label = new JLabel(value.toString()) {

                @Override
                protected void paintComponent(final Graphics g) {
                    final Graphics2D g2 = (Graphics2D) g.create();
                    g2.setPaint(new GradientPaint(0, 0, Color.GREEN, 0,
getHeight(), Color.YELLOW));
                    g2.fillRect(0, 0, getWidth(), getHeight());
                }
            };
        }
        return label;
    }
}
```

```
        super.paintComponent(g);
    }
}
} else {
    label = new JLabel(value.toString()) {

        @Override
        protected void paintComponent(final Graphics g) {
            final Graphics2D g2 = (Graphics2D) g.create();
            g2.setPaint(new GradientPaint(0, 0, Color.YELLOW, 0,
getHeight(), Color.GREEN));
            g2.fillRect(0, 0, getWidth(), getHeight());
            super.paintComponent(g);
        }
    };
}
label.setOpaque(false);

if (isSelected) {
    label.setForeground(Color.RED);
} else {
    label.setForeground(table.getForeground());
}

return label;
}
}
```

TableGradient.java

```
package pelajaran12;

import java.awt.BorderLayout;
import java.awt.GraphicsEnvironment;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TableGradient extends JTable {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setLayout(new BorderLayout());
                frame.add(new JScrollPane(new TableGradient()));
                frame.pack();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setSize(400, 300);
                frame.setVisible(true);
            }
        });
    }
}
```

```
}

public TableGradient() {
    super();

    final DefaultTableModel model = new DefaultTableModel();
    model.addColumn("FONT");
    for (final String f :
GraphicsEnvironment.getLocalGraphicsEnvironment()
        .getAvailableFontFamilyNames()) {
        model.addRow(new Object[] { f });
    }
    setModel(model);
    getColumnModel().getColumn(0).setCellRenderer(new GradientRender());
    setShowHorizontalLines(false);
    setShowVerticalLines(false);
    super.setRowMargin(0);
}
}
```



Gambar 118 TableGradient.java

NetBeans

OK tadi kita membuat render yang sangat simple, yach itung-itung pemanasan. Sekarang mau-gak mau anda yang pake Eclipse harus pindah IDE ke NetBeans dulu, karena Eclipse tak memiliki GUI Builder seperti NetBeans.

Sekarang kita mulai dengan membuat tabel yang berisikan alamat-alamat. OK sekarang kita buat class alamat sebagai penampung data. Saat ini saya buat yang simple aja, tapi saat praktek itu sih terserah anda.

Alamat.java

```
package pelajaran12;

/**
```

```
* @author usu
*/
public class Alamat {

    private String alamat;
    private String email;
    private String nama;

    public Alamat(final String nama, final String alamat, final String email) {
        this.nama = nama;
        this.alamat = alamat;
        this.email = email;
    }

    public String getAlamat() {
        return this.alamat;
    }

    public String getEmail() {
        return this.email;
    }

    public String getNama() {
        return this.nama;
    }

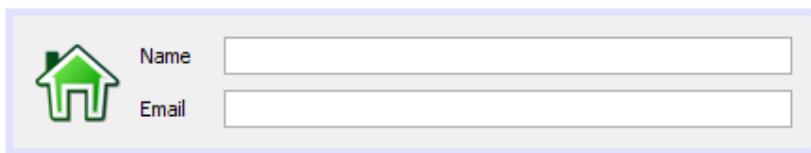
    public void setAlamat(final String alamat) {
        this.alamat = alamat;
    }

    public void setEmail(final String email) {
        this.email = email;
    }

    public void setNama(final String nama) {
        this.nama = nama;
    }
}
```

Sekarang anda buat sebuah panel seperti gambar dibawah menggunakan NetBeans, anda juga bisa menemukannya dalam project SwingMakeOver.

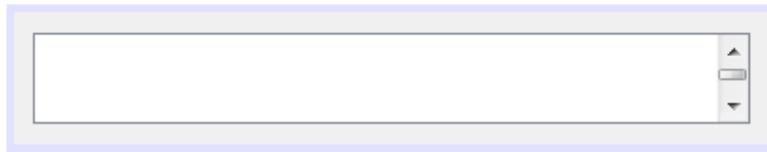
PanelAlamat.java



Gambar 119 PanelAlamat.java

Setelah itu buat panel lagi untuk menampung data alamat.

PanelAlamat2.java



Gambar 120 PanelAlamat2.java

Setelah itu buat metode pada PanelAlamat.java :

```
public void setName(String name){  
    textNama.setText(name);  
}  
  
public void setEmail(String email){  
    textEmail.setText(email);  
}
```

Dan setelah itu buat juga metode dibawah ini pada PanelAlamat2.java :

```
public void setAlamat(String alamat) {  
    textAlamat.setText(alamat);  
}
```

Sekarang tinggal buat dua buah Cell Render, karena kita akan membuat dua kolom.

RenderAlamat1.java

```
package pelajaran12;  
  
import java.awt.Component;  
import javax.swing.JTable;  
import javax.swing.table.TableCellRenderer;  
  
/**  
 * @author usu  
 */  
public class RenderAlamat1 implements TableCellRenderer {  
  
    public Component getTableCellRendererComponent(final JTable table, final  
Object value,  
        final boolean isSelected, final boolean hasFocus, final int row,  
final int column) {  
        final Alamat alamat = (Alamat) value;  
  
        final JPanel panel = new JPanel();  
        panel.setName(alamat.getNama());  
        panel.setEmail(alamat.getEmail());  
  
        if (isSelected) {  
            panel.setBackground(table.getSelectionBackground());  
        } else {  
            panel.setBackground(table.getBackground());  
        }  
    }  
}
```

```
    return panel;
}
}
```

RenderAlamat2.java

```
package pelajaran12;

import java.awt.Component;
import javax.swing.JTable;
import javax.swing.table.TableCellRenderer;

/**
 * @author usu
 */
public class RenderAlamat2 implements TableCellRenderer {

    public Component getTableCellRendererComponent(final JTable table, final
Object value,
            final boolean isSelected, final boolean hasFocus, final int row,
final int column) {
        final Alamat alamat = (Alamat) value;

        final PanelAlamat2 panel = new PanelAlamat2();
        panel.setAlamat(alamat.getAlamat());

        if (isSelected) {
            panel.setBackground(table.getSelectionBackground());
        } else {
            panel.setBackground(table.getBackground());
        }

        return panel;
    }
}
```

Sekarang tinggal membuat JTable yang akan menggunakan render tadi.

TableAlamat.java

```
package pelajaran12;

import java.awt.BorderLayout;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TableAlamat extends JTable {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

```

```
public void run() {
    final Frame frame = new Frame();
    frame.setLayout(new BorderLayout());
    frame.add(new JScrollPane(new TableAlamat()));
    frame.pack();
    frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    frame.setSize(400, 300);
    frame.setVisible(true);
}
}

public TableAlamat() {
    super();

    final DefaultTableModel model = new DefaultTableModel();
    model.addColumn("Data");
    model.addColumn("Alamat");

    for (int i = 0; i < 30; i++) {
        final Alamat data = new Alamat("Eko Kurniawan Khannedy",
"Kalijati, Subang",
"echo.khannedy@gmail.com");
        model.addRow(new Object[] { data, data });
    }

    setModel(model);
    getColumnModel().getColumn(0).setCellRenderer(new RenderAlamat1());
    getColumnModel().getColumn(1).setCellRenderer(new RenderAlamat2());
    setRowHeight(70);
    getColumnModel().getColumn(0).setMinWidth(400);
    getColumnModel().getColumn(1).setMinWidth(400);
    setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
}
}
```



Gambar 121 TableAlamat.java



Gambar 122 TableAlamat.java

Huhui, lebih OK kan kelihatannya dari pada tabel biasa. Walaupun kodenya agak panjang, tapi kan yang penting hasilnya bung!

Oh ya balik lagi ke IDE, kalo anda udah cinta mati ama Eclipse, lebih baik gunakan GUI Buildernya Eclipse yang gratis yaitu VisualEditor, tapi buat nginstallnya emang agak ribet, terlalu banyak plugin yang harus terinstall di Eclipsenya. Tapi kalo pengen yang instan, mending pake EasyEclipse.

Balik lagi ke yang tadi, bagaimana? Masih kurang keren, OK sekarang kita buat yang bener-bener Extreem.

JPanel kita buat gradient, trus JTextFieldnya kita buat Keren beken.

PanelCool.java

```
package pelajaran12;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.LinearGradientPaint;
import java.awt.Point;
import javax.swing.JPanel;

/**
 * @author usu
 */
public class PanelCool extends JPanel {

    private boolean reverse;

    public PanelCool() {
        super();
    }

    public boolean isReverse() {
        return this.reverse;
    }
}
```

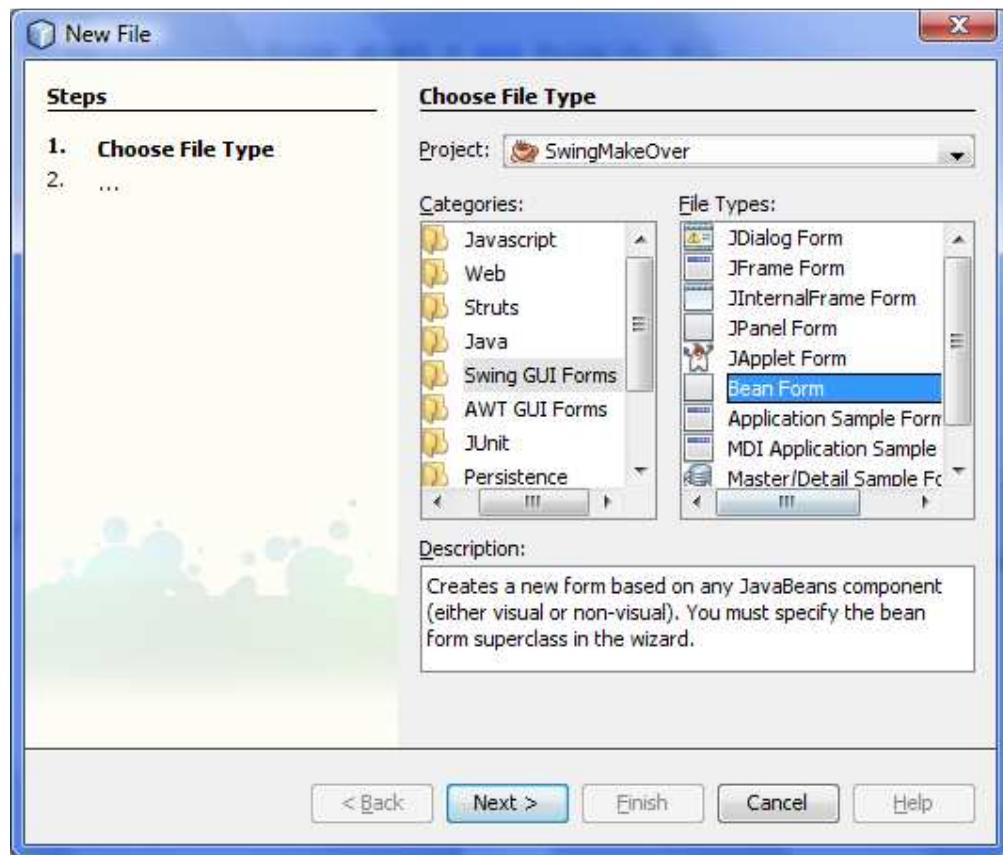
```
@Override
protected void paintComponent(final Graphics g) {
    super.paintComponent(g);

    final Point start = new Point(0, 0);
    final Point end = new Point(0, getHeight());
    final float[] fractions = new float[] { 0F, 0.5F, 1f };
    Color[] colors = null;
    if (isReverse()) {
        colors = new Color[] { Color.RED, Color.BLACK, Color.RED };
    } else {
        colors = new Color[] { Color.BLACK, Color.RED, Color.BLACK };
    }
    final LinearGradientPaint paint = new LinearGradientPaint(start, end,
fractions, colors);

    final Graphics2D g2 = (Graphics2D) g.create();
    g2.setPaint(paint);
    g2.fillRect(0, 0, getWidth(), getHeight());
}

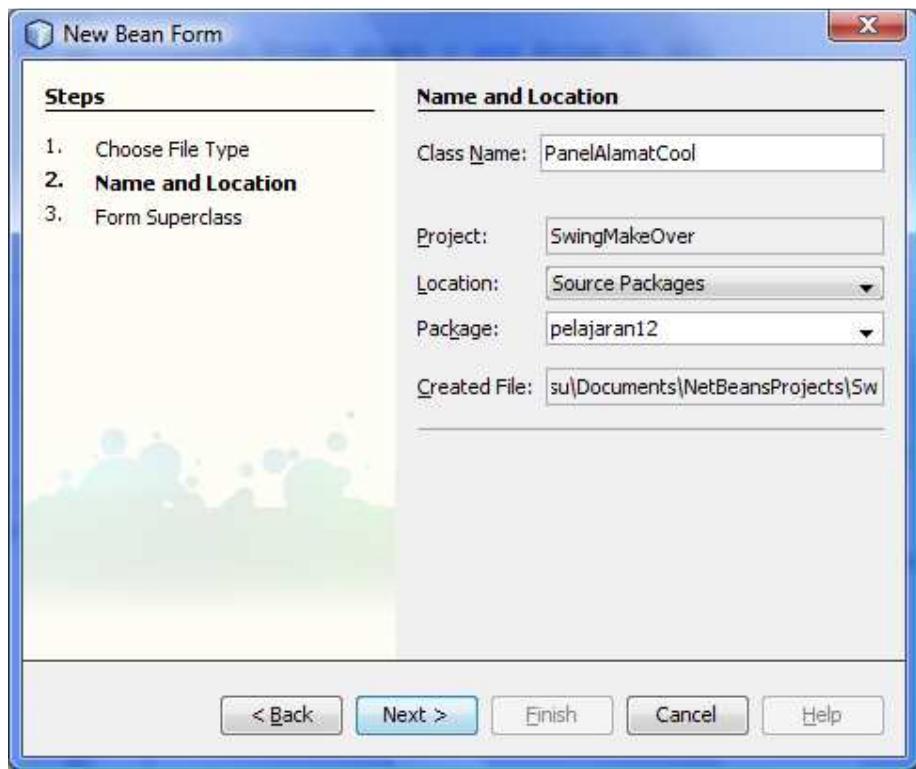
public void setReverse(final boolean reverse) {
    this.reverse = reverse;
    repaint();
}
}
```

Lalu buat file baru lewat menu File > New File...



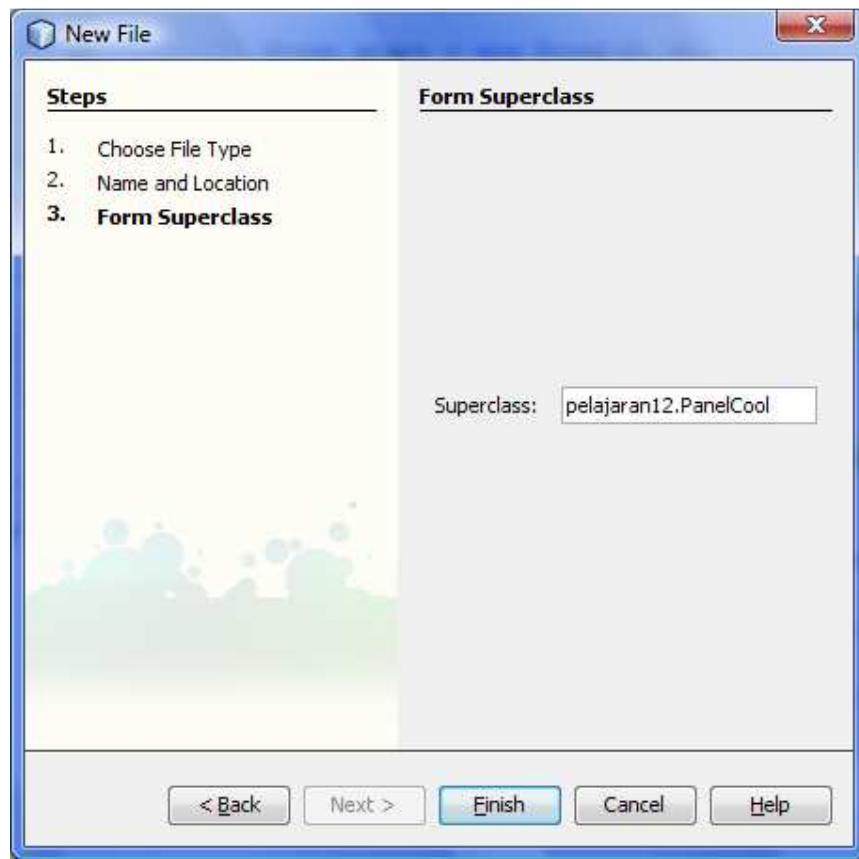
Gambar 123 New File

Pilih Swing GUI Forms lalu pilih Bean Form



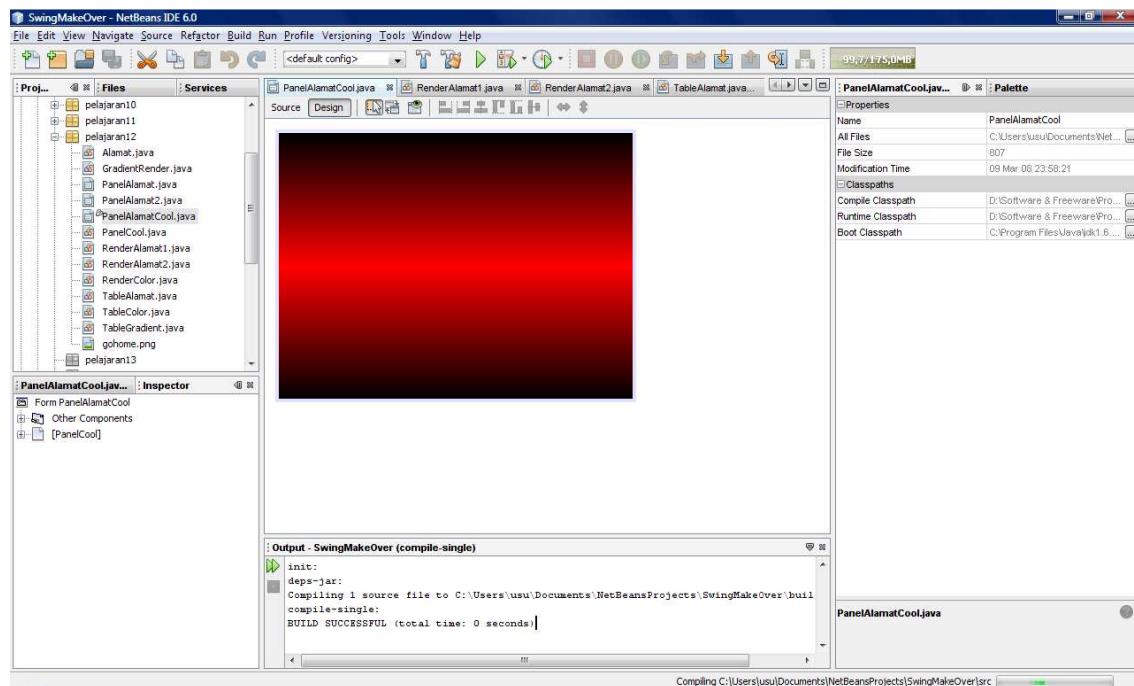
Gambar 124 New Bean Form

Beri nama dengan “PanelAlamatCool”.



Gambar 125 New File

Masukan superclassnya “pelajaran12.PanelCool”



Gambar 126 NetBeans GUI Builder

Sekarang terlihat gui editor untuk panel cool di mattise gui builder milik NetBeans. Dan sekarnag tinggal buat JTextField baru yang Cool.

TextFieldCool.java

```
package pelajaran12;

import java.awt.AlphaComposite;
import java.awt.Color;
import java.awt.Font;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Insets;
import java.awt.RenderingHints;
import java.awt.Shape;
import java.awt.geom.RoundRectangle2D;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;

/**
 * @author usu
 */
public class TextFieldCool extends JTextField {

    public TextFieldCool() {
        super();
        setBorder(new EmptyBorder(new Insets(5, 5, 5, 5)));
        setOpaque(false);
       setFont(getFont().deriveFont(Font.BOLD));
        setForeground(Color.white);
    }

    @Override
    protected void paintComponent(final Graphics g) {

        GradientPaint paint = new GradientPaint(0, 0, Color.BLACK, 0,
getHeight(), Color.BLUE);
        final Shape shape = new RoundRectangle2D.Double(0, 0, getWidth(),
getHeight(), getHeight(),
getHeight());

        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
        g2.setComposite(AlphaComposite.SrcOver.derive(0.8F));
        g2.setPaint(paint);
        g2.fill(shape);

        final Color dark = new Color(1F, 1F, 1F, 0F);
        final Color light = new Color(1F, 1F, 1F, 0.4F);
        paint = new GradientPaint(0, 0, light, 0, getHeight() / 2, dark);

        g2.setPaint(paint);
        g2.fill(shape);

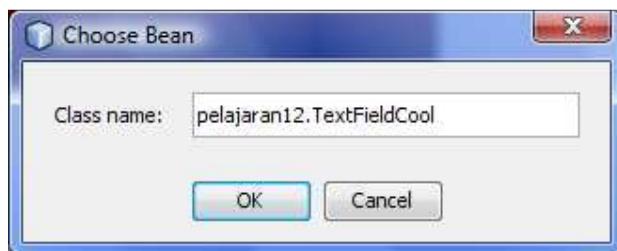
        super.paintComponent(g);
    }
}
```

Sekarang kembali ke cool panel gui editor trus buka dock pallete dan klik tombol “Chose Bean” :



Gambar 127 Choose Bean

Trus akan keluar dialog jawa bean :



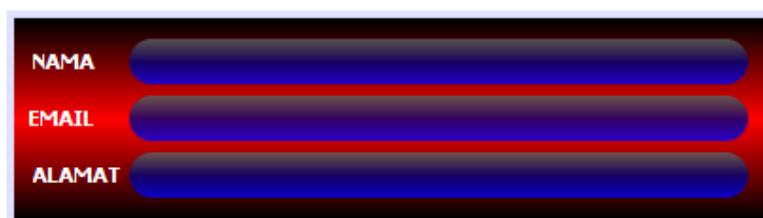
Gambar 128 Choose Bean

Masukkan “pelajaran12.TextFieldCool” dan OK



Gambar 129 Tampilan GUI Builder

Sekarang anda bebas untuk mendesain panel sesukan anda. Misal saya menggunakan desain standar seperti dibawah ini :



Gambar 130 Tampilan akhir render

Dan sekarang tambahkan metode dibawah ini pada class PanelAlamatCool.java nya :

```
public void setData(Alamat data) {
    textNama.setText(data.getNama());
    textEmail.setText(data.getEmail());
    textAlamat.setText(data.getAlamat());
}
```

Sekarang kita buat render buat tabelnya, ayo semanggat!!!

RenderAlamatCool.java

```
package pelajaran12;

import java.awt.Color;
import java.awt.Component;
import javax.swing.JLabel;
import javax.swing.JTable;
import javax.swing.table.TableCellRenderer;

/**
 * @author usu
 */
public class RenderColor implements TableCellRenderer {

    public Component getTableCellRendererComponent(final JTable table, final
Object value,
            final boolean isSelected, final boolean hasFocus, final int row,
final int column) {
        final Color warna = (Color) value;
        final JLabel label = new JLabel();
        label.setOpaque(true);
        if (isSelected) {
            label.setBackground(table.getSelectionBackground());
            label.setForeground(warna);
        } else {
            label.setBackground(warna);
            label.setForeground(table.getForeground());
        }
        label.setText("Red:" + warna.getRed() + ", Green:" + warna.getGreen()
+ ", Blue:"
                + warna.getBlue() + ", Alpha:" + warna.getAlpha());
        return label;
    }
}
```

OK tahap terakhir, kita buat JTabelnya. Let's EXTREEM!!!

TableAlamatCool.java

```
package pelajaran12;

import java.awt.BorderLayout;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;
```

```
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TableAlamatCool extends JTable {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setLayout(new BorderLayout());
                frame.add(new JScrollPane(new TableAlamatCool()));
                frame.pack();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setSize(400, 300);
                frame.setVisible(true);
            }
        });
    }

    public TableAlamatCool() {
        super();

        final DefaultTableModel model = new DefaultTableModel();
        model.addColumn("Data Tabel");

        for (int i = 0; i < 30; i++) {
            final Alamat data = new Alamat("Eko Kurniawan Khannedy",
"Kalijati, Subang",
"echo.khannedy@gmail.com");
            model.addRow(new Object[] { data });
        }

        setModel(model);
        getColumnModel().getColumn(0).setCellRenderer(new
RenderAlamatCool());
        setRowHeight(106);
        setRowMargin(0);
    }
}
```



Gambar 131 TableAlamatCool.java

Puih.. akhirnya selesai juga. Yach beginilah kalo mau EXTREEM harus ada pengorbanan.

Sekarang apa lagi yach? Hmmm gimana kalo kita buat background dengan JViewport jadi, backgroundnya gak seperti pada gambar diatas. OK let's ROCK!!!

ViewPortCool.java

```
package pelajaran12;

import java.awt.Color;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import javax.swing.JViewport;

/**
 * @author usu
 */
public class ViewPortCool extends JViewport {

    public ViewPortCool() {
        super();
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final GradientPaint paint = new GradientPaint(0, 0, Color.GREEN, 0,
getHeight(), Color.YELLOW);

        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setPaint(paint);
        g2.fillRect(0, 0, getWidth(), getHeight());
    }
}
```

```
}
```

Setelah itu buatlah panel untuk rendernya menggunakan Mattise GUI Buildernya NetBeans :

PanelViewPort.java



Gambar 132 PanelViewPort.java

Sekarang kita buat kode untuk menerima class Alamat pada PanelViewPort.java :

```
public void setData(Alamat data) {
    textAlamat.setText(data.getAlamat());
    textEmail.setText(data.getEmail());
    textNama.setText(data.getNama());
}
```

Sekarang kita buat rendernya.

RenderViewPort.java

```
package pelajaran12;

import java.awt.Component;
import javax.swing.JTable;
import javax.swing.table.TableCellRenderer;

/**
 * @author usu
 */
public class RenderViewPort implements TableCellRenderer {

    public Component getTableCellRendererComponent(final JTable table, final
Object value,
            final boolean isSelected, final boolean hasFocus, final int row,
final int column) {
        final Alamat data = (Alamat) value;

        final PanelViewPort panel = new PanelViewPort();
        panel.setOpaque(false);
        panel.setData(data);

        return panel;
    }
}
```

OK sekarang tahap terakhir, kita buat JTablenya dan ingat gunakan ViewPort yang telah kita buat sebelumnya.

TableAlamatViewPort.java

```
package pelajaran12;

import java.awt.BorderLayout;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TableAlamatViewPort extends JTable {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setLayout(new BorderLayout());

                final TableAlamatViewPort table = new TableAlamatViewPort();
                table.setOpaque(false);

                final ViewPortCool port = new ViewPortCool();
                port.setView(table);

                final JScrollPane pane = new JScrollPane();
                pane.setViewport(port);
                pane.setOpaque(false);

                frame.add(pane);
                frame.pack();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setSize(400, 300);
                frame.setVisible(true);
            }
        });
    }

    public TableAlamatViewPort() {
        super();

        final DefaultTableModel model = new DefaultTableModel();
        model.addColumn("Data Tabel");

        for (int i = 0; i < 30; i++) {
            final Alamat data = new Alamat("Eko Kurniawan Khannedy",
"Kalijati, Subang",
"echo.khannedy@gmail.com");
            model.addRow(new Object[ ] { data });
        }

        setModel(model);
    }
}
```

```
getColumnModel().getColumn(0).setCellRenderer(new RenderViewPort());  
setRowHeight(126);  
setRowMargin(0);  
}  
}
```



Gambar 133 TableAlamatViewPort.java

Jadi seperti itulah teknik untuk membuat tampilan yang beda dari sebuah JTable. Dan yang tak kalah penting lagi, saya disini hanya memancing daya kreativitas anda untuk lebih berkreasi dari contoh-contoh yang saya tulis dalam buku ini. Keep Extreem!

Editor

Yup, layaknya JTree, JTable pun memiliki Editor, tapi bedanya JTree hanya dapat menampung satu editor, tapi JTable boleh lebih dari satu editor tergantung banyak kolom, tapi syaratnya satu kolom harus memiliki satu editor.

JComboBox

Sebagai permulaan kita buat yang sederhana dulu.

TableEditorComboBox.java

```
package pelajaran1;  
  
import java.awt.BorderLayout;  
import javax.swing.DefaultCellEditor;  
import javax.swing.JComboBox;
```

```
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;

/**
 * @author usu
 */
public class TableEditorComboBox extends JTable {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final JFrame frame = new JFrame();
                frame.setLayout(new BorderLayout());
                frame.setSize(400, 300);
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.add(new JScrollPane(new TableEditorComboBox()));
                frame.setVisible(true);
            }
        });
    }

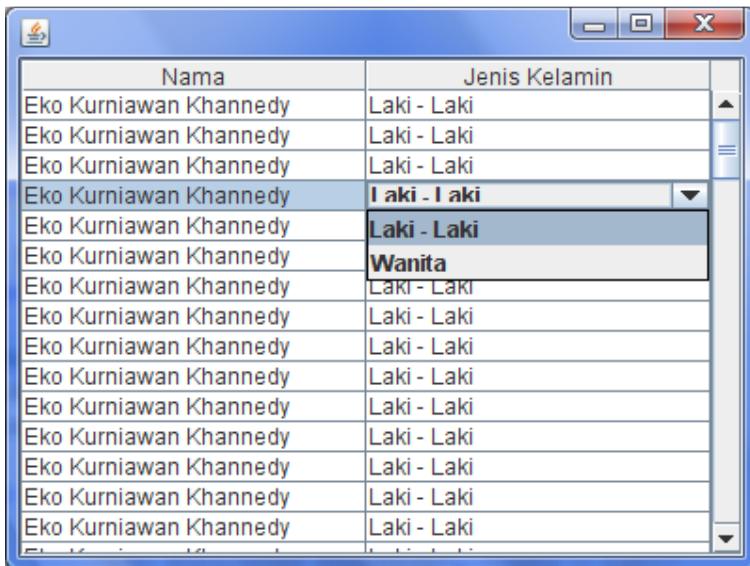
    public TableEditorComboBox() {
        super();

        final DefaultTableModel model = new DefaultTableModel();
        model.addColumn("Nama");
        model.addColumn("Jenis Kelamin");

        for (int i = 0; i < 100; i++) {
            model.addRow(new String[] { "Eko Kurniawan Khannedy", "Laki - Laki" });
        }

        setModel(model);

        getColumnModel().getColumn(0).setCellEditor(
            new DefaultCellEditor(new JTextField()));
        getColumnModel().getColumn(1).setCellEditor(
            new DefaultCellEditor(new JComboBox(new String[] { "Laki - Laki",
                "Wanita" })));
    }
}
```



Gambar 134 TableEditorComboBox.java

Hahaha, ternyata sama aja seperti membuat editor untuk JTree! Jadi gampang kan?

JChekBox

Pemanasan lagi, sekarang kita buat chekbox untuk editornya.

TableEditorChekBox.java

```
package pelajaran12;

import java.awt.BorderLayout;
import javax.swing.DefaultCellEditor;
import javax.swing.JCheckBox;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;

/**
 * @author usu
 */
public class TableEditorChekBox extends JTable {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final JFrame frame = new JFrame();
                frame.setLayout(new BorderLayout());
                frame.setSize(400, 300);
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.add(new JScrollPane(new TableEditorChekBox()));
                frame.setVisible(true);
            }
        });
    }
}
```

```
        }
    });

}

public TableEditorChekBox() {
    super();

    final DefaultTableModel model = new DefaultTableModel();
    model.addColumn("Nama");
    model.addColumn("Jenis Kelamin");
    model.addColumn("Menikah");

    for (int i = 0; i < 100; i++) {
        model.addRow(new Object[] { "Eko Kurniawan Khannedy", "Laki - Laki",
            false });
    }

    setModel(model);

    getColumnModel().getColumn(0).setCellEditor(
        new DefaultCellEditor(new JTextField()));
    getColumnModel().getColumn(1).setCellEditor(
        new DefaultCellEditor(new JComboBox(new String[] { "Laki - Laki",
            "Wanita" }))));
    getColumnModel().getColumn(2).setCellEditor(
        new DefaultCellEditor(new JCheckBox())));
}
}
```

Gambar 135 TableEditorChekBox.java

Wah tapi kayaknya kelihatan sangat formal, gimana kalo kita buat yang Extreem, kalo gak mau skip aja subbab ini, tapi saya saranin baca seluruh buku ini tanpa terlewati sedikitpun.

Mula mula kita buat dulu tampilan render untuk nama :



Gambar 136 Panel render nama

Jangan lupa menggunakan metode dibawah ini :

```
public void setNama(String nama){  
    textNama.setText(nama);  
}
```

Sekarang kita buat tampilan untuk jenis kelamin :



Gambar 137 Panel render jenis kelamin

Jangan lupa buat metode-metode dibawah ini :

```
public void setJenisKelamin(String JenisKelamin) {  
    textJenisKelamin.setText(JenisKelamin);  
    if (JenisKelamin.equals("Wanita")) {  
        labelJenisKelamin.setIcon(new  
        ImageIcon(getClass().getResource("/pelajaran12/Women.png")));  
    } else {  
        labelJenisKelamin.setIcon(new  
        ImageIcon(getClass().getResource("/pelajaran12/Men.png")));  
    }  
}
```

Sekarang kita buat tampilan untuk status menikah :



Gambar 138 Panel render menikah

Jangan lupa membuat metode ini :

```
public void setMenikah(boolean menikah) {  
    if (menikah) {  
        textMenikah.setText("Sudah Menikah");  
        labelMenikah.setIcon(new ImageIcon(getClass().getResource("/pelajaran12/OK.png")));  
    } else {  
        textMenikah.setText("Belum Menikah");  
        labelMenikah.setIcon(new ImageIcon(getClass().getResource("/pelajaran12/BelumMenikah.png")));  
    }  
}
```

```
    textMenikah.setText("Belum Menikah");
    labelMenikah.setIcon(new ImageIcon(getClass().getResource("/pelajaran12/NO.png")));
}
}
```

Ok sekarang kita tinggal membuat rendernya. Karena ada tiga kolom maka kita akan membuat tiga render.

RenderNama.java

```
package pelajaran12;

import java.awt.Component;
import javax.swing.JTable;
import javax.swing.table.TableCellRenderer;

/**
 * @author usu
 */
public class RenderNama implements TableCellRenderer {

    public Component getTableCellRendererComponent(final JTable table,
        final Object value, final boolean isSelected, final boolean
hasFocus,
        final int row, final int column) {
        final String data = (String) value;

        final TampilanNama tampilan = new TampilanNama();
        tampilan.setNama(data);

        if (isSelected) {
            tampilan.setOpaque(true);
            tampilan.setBackground(table.getSelectionBackground());
        } else {
            tampilan.setOpaque(false);
        }

        return tampilan;
    }
}
```

RenderJenisKelamin.java

```
package pelajaran12;

import java.awt.Component;
import javax.swing.JTable;
import javax.swing.table.TableCellRenderer;

/**
 * @author usu
 */
public class RenderJenisKelamin implements TableCellRenderer {

    public Component getTableCellRendererComponent(final JTable table,
        final Object value, final boolean isSelected, final boolean
```

```
hasFocus,
    final int row, final int column) {
String data = "Laki - Laki";
if (value != null) {
    data = (String) value;
}

final TampilanJenisKelamin tampilan = new TampilanJenisKelamin();
tampilan.setJeniKelamin(data);

if (isSelected) {
    tampilan.setOpaque(true);
    tampilan.setBackground(table.getSelectionBackground());
} else {
    tampilan.setOpaque(false);
}

return tampilan;
}
}
```

RenderMenikah.java

```
package pelajaran12;

import java.awt.Component;
import javax.swing.JTable;
import javax.swing.table.TableCellRenderer;

/**
 * @author usu
 */
public class RenderMenikah implements TableCellRenderer {

    public Component getTableCellRendererComponent(final JTable table,
        final Object value, final boolean isSelected, final boolean
hasFocus,
        final int row, final int column) {
    boolean data = false;
    if (value != null) {
        data = Boolean.valueOf(value.toString()).booleanValue();
    }

    final TampilanMenikah tampilan = new TampilanMenikah();
    tampilan.setMenikah(data);

    if (isSelected) {
        tampilan.setOpaque(true);
        tampilan.setBackground(table.getSelectionBackground());
    } else {
        tampilan.setOpaque(false);
    }

    return tampilan;
}
}
```

Sekarang kita buat JViewPort gradient, buat tampilan background tabel.

ViewPortCool2.java

```
package pelajaran12;

import java.awt.Color;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import javax.swing.JViewport;

/**
 * @author usu
 */
public class ViewPortCool2 extends JViewport {

    public ViewPortCool2() {
        super();
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final GradientPaint paint = new GradientPaint(0, 0, Color.GREEN,
            getWidth(), getHeight(), Color.YELLOW);

        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setPaint(paint);
        g2.fillRect(0, 0, getWidth(), getHeight());
    }
}
```

Ok sekarang tinggal buat JTabelnya..

TableEditorStylist.java

```
package pelajaran12;

import java.awt.BorderLayout;
import javax.swing.DefaultCellEditor;
import javax.swing.JCheckBox;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;

/**
 * @author usu
 */
public class TableEditorStylist extends JTable {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
```

```
final JFrame frame = new JFrame();
frame.setLayout(new BorderLayout());
frame.setSize(400, 300);
frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);

final TableEditorStylist table = new TableEditorStylist();
table.setOpaque(false);

final ViewPortCool2 port = new ViewPortCool2();
port.setView(table);

final JScrollPane pane = new JScrollPane();
pane.setOpaque(false);
pane.setViewport(port);

frame.add(pane);
frame.setVisible(true);
}

}

public TableEditorStylist() {
super();

final DefaultTableModel model = new DefaultTableModel();
model.addColumn("Nama");
model.addColumn("Jenis Kelamin");
model.addColumn("Menikah");

for (int i = 0; i < 100; i++) {
model.addRow(new Object[] { "Eko Kurniawan Khannedy", "Laki - Laki",
false });
}

setModel(model);

getColumnModel().getColumn(0).setCellRenderer(new RenderNama());
getColumnModel().getColumn(1).setCellRenderer(new
RenderJenisKelamin());
getColumnModel().getColumn(2).setCellRenderer(new RenderMenikah());

final DefaultCellEditor editorNama = new DefaultCellEditor(
new JTextField());
editorNama.setClickCountToStart(2);
getColumnModel().getColumn(0).setCellEditor(editorNama);

final DefaultCellEditor editorJenisKelamin = new DefaultCellEditor(
new JComboBox(new String[] { "Laki - Laki", "Wanita" }));
editorJenisKelamin.setClickCountToStart(2);
getColumnModel().getColumn(1).setCellEditor(editorJenisKelamin);

final DefaultCellEditor editorMenikah = new DefaultCellEditor(
new JCheckBox("Menikah?"));
editorMenikah.setClickCountToStart(2);
getColumnModel().getColumn(2).setCellEditor(editorMenikah);

setAutoResizeMode(JTable.AUTO_RESIZE_OFF);

setRowHeight(54);
for (int i = 0; i < getColumnCount(); i++) {
```

```
        getColumnModel().getColumn(i).setMinWidth(300);  
    }  
}  
}
```

| Nama | Jenis Kelamin | Menikah |
|------------------------|---------------|---------|
| Eko Kurniawan Khannedy | Laki - Laki | X |
| Nesia Oktiana | Laki - Laki | X |
| Eko Kurniawan Khannedy | Laki - Laki | X |
| Eko Kurniawan Khannedy | Laki - Laki | X |
| Eko Kurniawan Khannedy | Laki - Laki | X |

Gambar 139 TableEditorStylist saat mengedit nama

| Nama | Jenis Kelamin | Menikah |
|------------------------|-----------------------|---------|
| Eko Kurniawan Khannedy | Laki - Laki | X |
| Nesia Oktiana | Laki - Laki | X |
| Eko Kurniawan Khannedy | Laki - Laki Wanita | X |
| Eko Kurniawan Khannedy | Laki - Laki | X |
| Eko Kurniawan Khannedy | Laki - Laki | X |

Gambar 140 TableEditorStylist saat mengedit jenis kelamin



Gambar 141 TableEditorStylist saat mengedit menikah

Huh akhirnya selesai juga...

Color

Sampai saat ini DefaultCellEditor hanya mendukung editor JCheckBox, JComboBox, dan JTextField, sehingga kita ingin membuat editor untuk warna, file, atau gambar kita harus membuatnya sendiri. Sekarang kita coba membuat editor untuk Color. Jika anda sudah menguasai penggunaan editor color pada JTree, anda pasti mudah memahami editor Color untuk JTable karena hampir sama penggunaannya.

Sekarang kita buat editornya.

ColorEditor.java

```
package pelajaran12;

import java.awt.Color;
import java.awt.Component;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.AbstractCellEditor;
import javax.swing.JColorChooser;
import javax.swing.JLabel;
import javax.swing.JTable;
import javax.swing.table.TableCellEditor;

/**
 * @author usu
 */
public class ColorEditor extends AbstractCellEditor implements TableCellEditor {
```

```
    private final JLabel label;
    private JTable table;
    private Color warna;

    public ColorEditor() {
        super();
        this.label = new JLabel();
```

```
this.label.setOpaque(true);
this.label.addMouseListener(new MouseAdapter() {

    @Override
    public void mouseClicked(MouseEvent e) {
        if (e.getClickCount() >= 2) {
            final Color temp = JColorChooser.showDialog(
                ColorEditor.this.table, "Color Editor",
                ColorEditor.this.warna);
            if (temp != null) {
                ColorEditor.this.warna = temp;
            }
            stopCellEditing();
        }
    }
} );
}

public Object getCellEditorValue() {
    return this.warna;
}

public Component getTableCellEditorComponent(JTable table,
    final Object value, final boolean isSelected, final int row,
    final int column) {
    this.table = table;
    this.warna = (Color) value;
    this.label.setBackground(this.warna);
    return this.label;
}
}
```

Sekarang kita buat JTabelnya.

TableColorEditor.java

```
package pelajaran12;

import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TableColorEditor extends JTable {

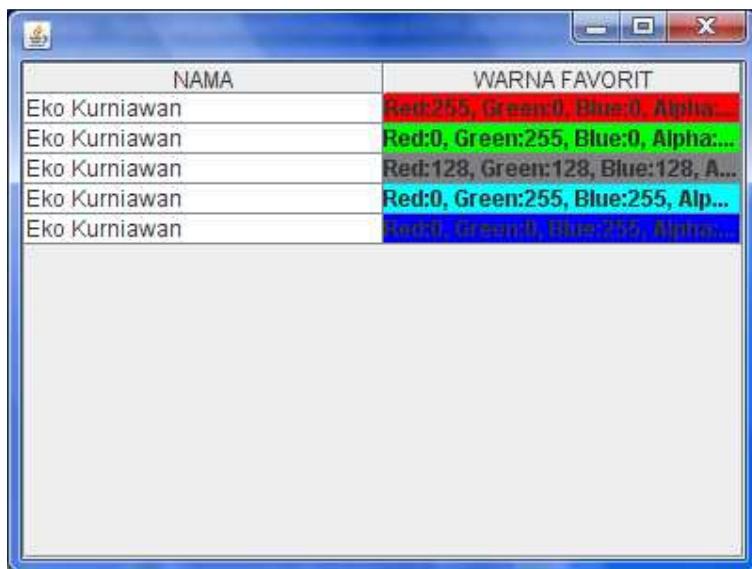
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setLayout(new BorderLayout());
                frame.add(new JScrollPane(new TableColorEditor()));
            }
        });
    }
}
```

```
        frame.pack();
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.setSize(400, 300);
        frame.setVisible(true);
    }
}

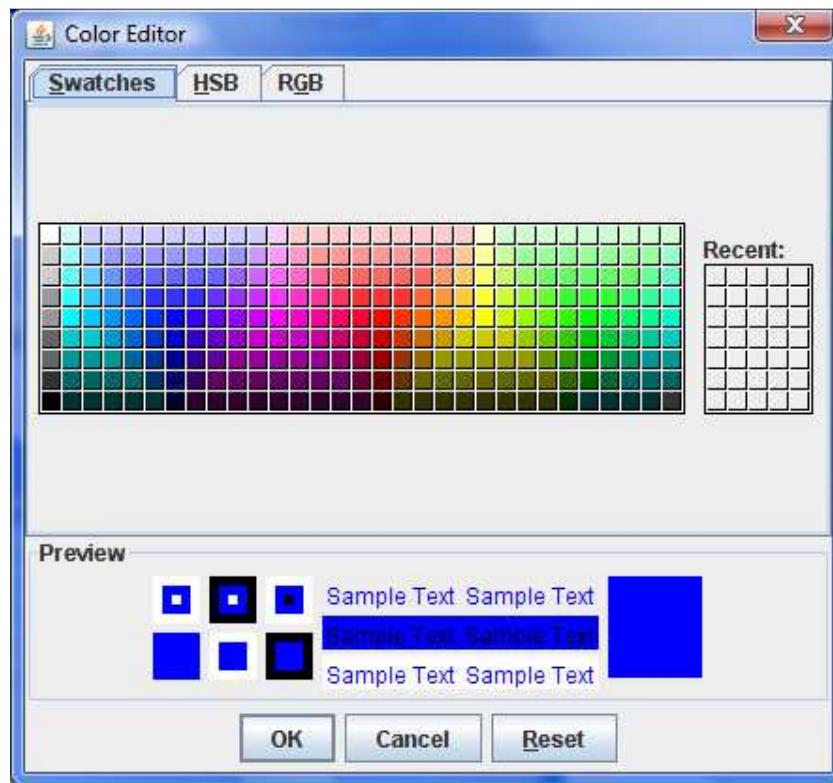
public TableColorEditor() {
    super();

    final DefaultTableModel model = new DefaultTableModel();
    model.addColumn("NAMA");
    model.addColumn("WARNA FAVORIT");
    model.addRow(new Object[] { "Eko Kurniawan", Color.RED });
    model.addRow(new Object[] { "Eko Kurniawan", Color.GREEN });
    model.addRow(new Object[] { "Eko Kurniawan", Color.GRAY });
    model.addRow(new Object[] { "Eko Kurniawan", Color.CYAN });
    model.addRow(new Object[] { "Eko Kurniawan", Color.BLUE });
    setModel(model);
    getColumnModel().getColumn(1).setCellRenderer(new RenderColor());
    getColumnModel().getColumn(1).setCellEditor(new ColorEditor());
}
}
```



Gambar 142 TableColorEditor.java

Jika kita double klik salah satu kolom warna favorit, maka akan keluar Color Editor :



Gambar 143 Color Editor

Spinner

Wah selama ini saya jarang menyinggung JSpinner, tapi sekarang saya akan buat editor untuk JTable yang menamfaatkan JSpinner.

Number

Kali ini kita akan membuat editor nomber dengan menggunakan JSpinner.

NumberEditor.java

```
package pelajaran12;

import java.awt.Component;
import java.awt.GridLayout;
import java.awt.Point;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.ParseException;
import java.util.EventObject;
import javax.swing.AbstractCellEditor;
import javax.swing.JButton;
import javax.swing.JSpinner;
import javax.swing.JTable;
import javax.swing.JWindow;
import javax.swing.SpinnerNumberModel;
import javax.swing.table.TableCellEditor;

/**
 * @author usu
```

```
/*
public class NumberEditor extends AbstractCellEditor implements
TableCellEditor {

    private Integer number;
    private final JButton ok, no;
    private final JSpinner spinner;
    private final JWindow window;

    public NumberEditor() {
        super();
        this.spinner = new JSpinner(new SpinnerNumberModel(0, null, null,
1));
        this.window = new JWindow();
        this.ok = new JButton("OK");
        this.ok.addActionListener(new ActionListener() {

            public void actionPerformed(final ActionEvent e) {
                try {
                    NumberEditor.this.spinner.commitEdit();
                } catch (final ParseException ex) {
                    // ERROR
                } finally {
                    NumberEditor.this.number = new
Integer(NumberEditor.this.spinner
                    .getValue().toString());
                    stopCellEditing();
                }
            }
        });
        this.no = new JButton("NO");
        this.no.addActionListener(new ActionListener() {

            public void actionPerformed(final ActionEvent e) {
                cancelCellEditing();
            }
        });
        this.window.setLayout(new GridLayout(1, 2));
        this.window.add(this.ok);
        this.window.add(this.no);
        this.window.setAlwaysOnTop(true);
    }

    @Override
    public void cancelCellEditing() {
        this.window.setVisible(false);
        super.cancelCellEditing();
    }

    public Object getCellEditorValue() {
        return this.number;
    }

    public Component getTableCellEditorComponent(final JTable table,
        final Object value, final boolean isSelected, final int row,
        final int column) {
        this.number = new Integer(value.toString());
        this.spinner.setValue(value);
        return this.spinner;
    }
}
```

```
@Override
public boolean shouldSelectCell(final EventObject anEvent) {
    final Point p = this.spinner.getLocationOnScreen();
    p.y += this.spinner.getHeight();
    this.window.setLocation(p);
    this.window.pack();
    this.window.setVisible(true);
    return super.shouldSelectCell(anEvent);
}

@Override
public boolean stopCellEditing() {
    this.window.setVisible(false);
    return super.stopCellEditing();
}
}
```

Sekarang kita buat JTablenya.

TableNumberEditor.java

```
package pelajaran12;

import java.awt.BorderLayout;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TableNumberEditor extends JTable {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setLayout(new BorderLayout());
                frame.add(new JScrollPane(new TableNumberEditor()));
                frame.pack();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setSize(400, 300);
                frame.setVisible(true);
            }
        });
    }

    public TableNumberEditor() {
        super();

        final DefaultTableModel model = new DefaultTableModel();
        model.addColumn("NAMA");
        model.addColumn("UMUR");
        model.addRow(new Object[] { "Eko Kurniawan", new Integer(19) });
        model.addRow(new Object[] { "Eko Kurniawan", new Integer(19) });
    }
}
```

```
model.addRow(new Object[] { "Eko Kurniawan", new Integer(19) });
model.addRow(new Object[] { "Eko Kurniawan", new Integer(19) });
model.addRow(new Object[] { "Eko Kurniawan", new Integer(19) });
setModel(model);
getColumnModel().getColumn(1).setCellEditor(new NumberEditor());
}
}
```

| NAMA | UMUR |
|---------------|------|
| Eko Kurniawan | 19 |

Gambar 144 TableNumberEditor.java

Dan jika kita mengedit kolom umur, maka akan keluar Number Editor :

| NAMA | UMUR |
|---------------|------|
| Eko Kurniawan | 28 |
| Eko Kurniawan | OK |
| Eko Kurniawan | NO |
| Eko Kurniawan | 19 |
| Eko Kurniawan | 19 |

Gambar 145 TableNumberEditor saat mengedit umur

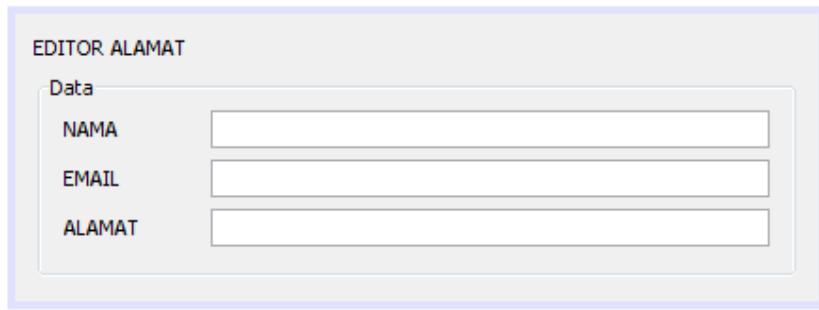
Dan lagi – lagi disini saya menggunakan JWindow untuk menampilkan tombol OK dan No.

Custom Editor

Kadang kita mendapatkan sebuah tabel yang menyimpan data berupa class yang dibuat sendiri (dalam arti tak ada dalam standar api java), sehingga mau gak mau kita harus membuat sebuah editor yang cocok dengan data class tersebut. Misalkan saja kita akan mengedit sebuah Alamat yang sebelumnya kita pernah buat.

Sebelumnya kita buat Panel editornya.

PanelEditorAlamat.java



Gambar 146 Panel editor alamat

Sekarang buat metode dibawah ini :

```
public void setAlamar(Alamat data) {  
    textAlamat.setText(data.getAlamat());  
    textEmail.setText(data.getEmail());  
    textNama.setText(data.getNama());  
}  
  
public Alamat getAlamat() {  
    return new Alamat(textNama.getText(),  
        textAlamat.getText(), textEmail.getText());  
}
```

OK sekarang kita tinggal buat editornya.

AlamatEditor.java

```
package pelajaran12;  
  
import java.awt.Component;  
import java.util.EventObject;  
  
import javax.swing.AbstractCellEditor;  
import javax.swing.JOptionPane;  
import javax.swing.JTable;  
import javax.swing.table.TableCellEditor;  
  
/**  
 * @author usu  
 */  
public class AlamatEditor extends AbstractCellEditor implements  
TableCellEditor {  
  
    private Alamat alamat;  
    private final PanelEditorAlamat panel;  
    private final PanelAlamatCool panelCool;  
  
    public AlamatEditor() {  
        this.panel = new PanelEditorAlamat();  
        this.panelCool = new PanelAlamatCool();  
    }  
}
```

```
public Object getCellEditorValue() {
    return this.alamat;
}

public Component getTableCellEditorComponent(final JTable table,
    final Object value, final boolean isSelected, final int row,
    final int column) {
    this.alamat = (Alamat) value;
    this.panel.setAlamar(this.alamat);
    this.panelCool.setData(this.alamat);
    return this.panelCool;
}

@Override
public boolean shouldSelectCell(final EventObject anEvent) {
    JOptionPane.showMessageDialog(null, this.panel);
    this.alamat = this.panel.getAlamat();
    stopCellEditing();
    return super.shouldSelectCell(anEvent);
}
}
```

Dan terakhir kita buat JTabelnya.

TableAlamatCoolEditor.java

```
package pelajaran12;

import java.awt.BorderLayout;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;
import pelajaran1.Frame;

/**
 * @author usu
 */
public class TableAlamatCoolEditor extends JTable {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final Frame frame = new Frame();
                frame.setLayout(new BorderLayout());
                frame.add(new JScrollPane(new TableAlamatCoolEditor()));
                frame.pack();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setSize(400, 300);
                frame.setVisible(true);
            }
        });
    }

    public TableAlamatCoolEditor() {
        super();
    }
}
```

```
final DefaultTableModel model = new DefaultTableModel();
model.addColumn("Data Tabel");

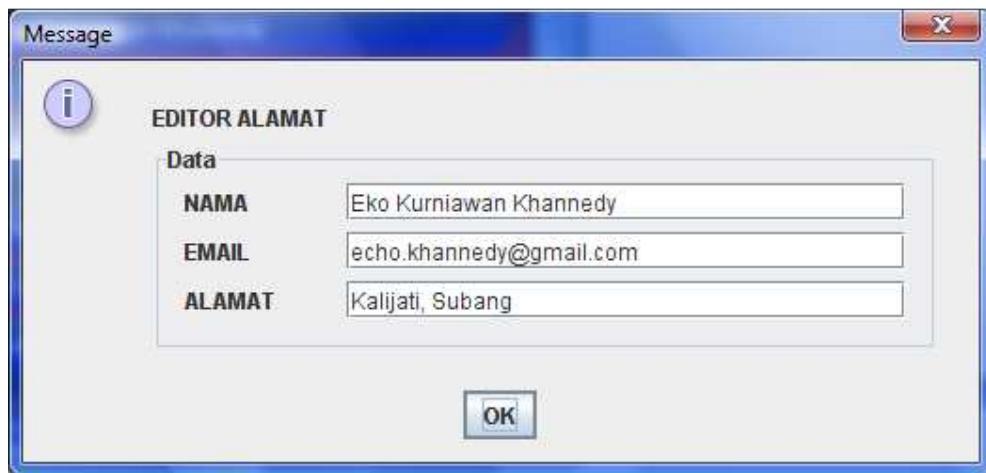
for (int i = 0; i < 30; i++) {
    final Alamat data = new Alamat("Eko Kurniawan Khannedy",
        "Kalijati, Subang", "echo.khannedy@gmail.com");
    model.addRow(new Object[] { data });
}

setModel(model);
getColumnModel().getColumn(0).setCellRenderer(new
RenderAlamatCool());
getColumnModel().getColumn(0).setCellEditor(new AlamatEditor());
setRowHeight(106);
setRowMargin(0);
}
}
```



Gambar 147 TableAlamatCoolEditor.java

Jika kita klik salah satu cellnya maka akan keluar Alamat Editor :



Gambar 148 Editor Alamat

Other

Selain Render dan Editor, masih banyak lagi extreem trix yang bisa kita gunakan dalam JTable. Dan pada subbab selanjutnya kita akan meninggalkan render dan editor, kita akan lebih fokus ke RowSorter dan TableModel.

Sorter

Pada Java versi 1.5 ke bawah , sebuah JTable tak bisa diurutkan, tapi sejak Java 1.6 JTable sudah dapat diurutkan, sekarang kita coba untuk mengurutkan JTable.

Metode yang baru pada versi 1.6 adalah setRowSorter(), metode ini digunakan untuk membuat JTable agar bisa melakukan sorting.

TableSorter.java

```
package pelajaran12;

import java.awt.BorderLayout;
import java.util.Random;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;

/**
 * @author usu
 */
public class TableSorter extends JTable {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final JFrame frame = new JFrame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
            }
        });
    }
}
```

```
        frame.setLayout(new BorderLayout());
        frame.add(new JScrollPane(new TableSorter()));
        frame.setSize(400, 300);
        frame.setVisible(true);
    }
}

private final Random generator;

private final TableRowSorter<DefaultTableModel> sorter;

public TableSorter() {
    super();

    this.generator = new Random();

    final DefaultTableModel model = new DefaultTableModel();
    model.addColumn("Nama");
    model.addColumn("Umur");
    for (int i = 0; i < 100; i++) {
        model.addRow(new Object[] { "Eko Kurniawan Khannedy",
            this.generator.nextInt(40) });
    }

    this.sorter = new TableRowSorter<DefaultTableModel>(model);

    setModel(model);
    setRowSorter(this.sorter);
}
}
```

| Nama | Umur |
|------------------------|------|
| Eko Kurniawan Khannedy | 17 |
| Eko Kurniawan Khannedy | 19 |
| Eko Kurniawan Khannedy | 24 |
| Eko Kurniawan Khannedy | 14 |
| Eko Kurniawan Khannedy | 12 |
| Eko Kurniawan Khannedy | 0 |
| Eko Kurniawan Khannedy | 19 |
| Eko Kurniawan Khannedy | 20 |
| Eko Kurniawan Khannedy | 24 |
| Eko Kurniawan Khannedy | 14 |
| Eko Kurniawan Khannedy | 30 |
| Eko Kurniawan Khannedy | 23 |
| Eko Kurniawan Khannedy | 24 |
| Eko Kurniawan Khannedy | 30 |
| Eko Kurniawan Khannedy | 20 |

Gambar 149 TableSorter.java

Dengan mengklik kolom headernya kita bisa mensorting colom tersebut berdasarkan ascending atau descending.

| Nama | Umur |
|------------------------|------|
| Eko Kurniawan Khannedy | 0 |
| Eko Kurniawan Khannedy | 0 |
| Eko Kurniawan Khannedy | 1 |
| Eko Kurniawan Khannedy | 1 |
| Eko Kurniawan Khannedy | 10 |
| Eko Kurniawan Khannedy | 10 |
| Eko Kurniawan Khannedy | 11 |
| Eko Kurniawan Khannedy | 11 |
| Eko Kurniawan Khannedy | 12 |
| Eko Kurniawan Khannedy | 13 |

Gambar 150 TableSorter saat terurut secara ascending

| Nama | Umur |
|------------------------|------|
| Eko Kurniawan Khannedy | 9 |
| Eko Kurniawan Khannedy | 9 |
| Eko Kurniawan Khannedy | 9 |
| Eko Kurniawan Khannedy | 8 |
| Eko Kurniawan Khannedy | 8 |
| Eko Kurniawan Khannedy | 8 |
| Eko Kurniawan Khannedy | 7 |
| Eko Kurniawan Khannedy | 7 |
| Eko Kurniawan Khannedy | 7 |
| Eko Kurniawan Khannedy | 6 |
| Eko Kurniawan Khannedy | 6 |
| Eko Kurniawan Khannedy | 5 |
| Eko Kurniawan Khannedy | 4 |
| Eko Kurniawan Khannedy | 39 |
| Eko Kurniawan Khannedy | 39 |
| Eko Kurniawan Khannedy | 39 |

Gambar 151 TableSorter saat mengurut secara descending

Sepertinya ada yang kesalahan dalam sortingnya. RowSorter secara default mensort dalam nilai String jadi jika ada 39 dan 9 maka menurut RowSorter lebih besar 9. Oleh karena itu kita harus membuat Comparator sendiri. Karena dalam kolom umur typenya Integer, maka kita buat Comparator Integer.

TableSorterGood.java

```
package pelajaran12;

import java.awt.BorderLayout;
import java.util.Comparator;
import java.util.Random;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
```

```
import javax.swing.JTable;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;

/**
 * @author usu
 */
public class TableRowSorterGood extends JTable {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final JFrame frame = new JFrame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setLayout(new BorderLayout());
                frame.add(new JScrollPane(new TableRowSorterGood()));
                frame.setSize(400, 300);
                frame.setVisible(true);
            }
        });
    }

    private final Random generator;

    private final TableRowSorter<DefaultTableModel> sorter;

    public TableRowSorterGood() {
        super();

        this.generator = new Random();

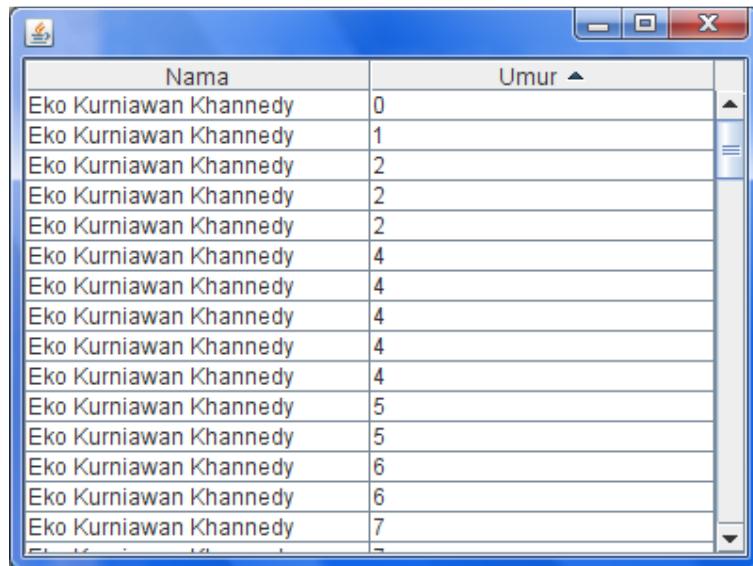
        final DefaultTableModel model = new DefaultTableModel();
        model.addColumn("Nama");
        model.addColumn("Umur");
        for (int i = 0; i < 100; i++) {
            model.addRow(new Object[] { "Eko Kurniawan Khannedy",
                new Integer(this.generator.nextInt(40)) });
        }

        this.sorter = new TableRowSorter<DefaultTableModel>(model);
        this.sorter.setComparator(1, new Comparator<Integer>() {

            public int compare(final Integer o1, final Integer o2) {
                return o1.compareTo(o2);
            }
        });

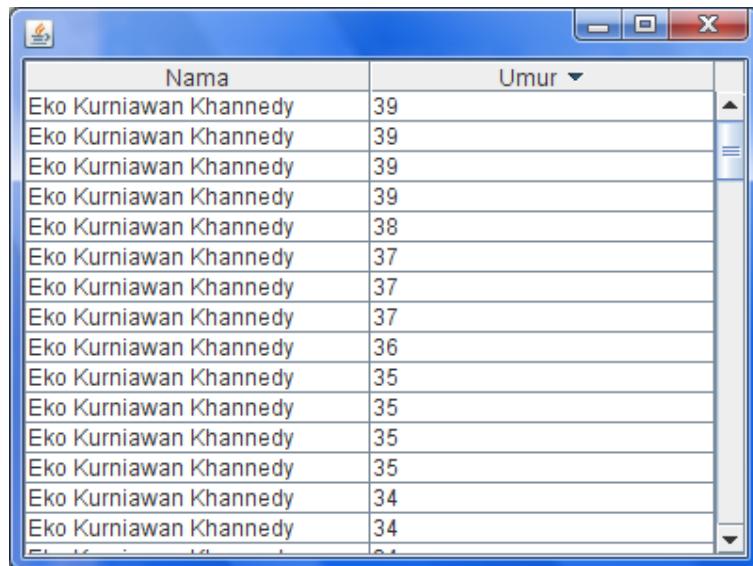
        setModel(model);
        setRowSorter(this.sorter);
    }
}
```

Sekarang RowSorter tak akan salah lagi mengurutkan Integer :



| Nama | Umur |
|------------------------|------|
| Eko Kurniawan Khannedy | 0 |
| Eko Kurniawan Khannedy | 1 |
| Eko Kurniawan Khannedy | 2 |
| Eko Kurniawan Khannedy | 2 |
| Eko Kurniawan Khannedy | 2 |
| Eko Kurniawan Khannedy | 4 |
| Eko Kurniawan Khannedy | 5 |
| Eko Kurniawan Khannedy | 5 |
| Eko Kurniawan Khannedy | 6 |
| Eko Kurniawan Khannedy | 6 |
| Eko Kurniawan Khannedy | 7 |
| Eko Kurniawan Khannedy | 7 |
| Eko Kurniawan Khannedy | 7 |

Gambar 152 TableSorterGood saat mengurut ascending



| Nama | Umur |
|------------------------|------|
| Eko Kurniawan Khannedy | 39 |
| Eko Kurniawan Khannedy | 38 |
| Eko Kurniawan Khannedy | 37 |
| Eko Kurniawan Khannedy | 37 |
| Eko Kurniawan Khannedy | 37 |
| Eko Kurniawan Khannedy | 36 |
| Eko Kurniawan Khannedy | 35 |
| Eko Kurniawan Khannedy | 34 |

Gambar 153 TableSorterGood saat mengurut descending

Filter

Selain mengurutkan kolom, RowSorter juga memiliki kemampuan untuk memfilter data. Kita pernah membuat ListModel yang dapat memfilter dalam bab JList, nah sekarang kita gak perlu susah-susah membuatTableModel yang bisa memfilter karena sudah dihandle oleh RowSorter.

Untuk memfilter RowSorter gunakan metode ini :

```
sorter.setRowFilter(RowFilter.regexFilter(String key));
```

TableSorterFilter.java

```
package pelajaran12;

import java.awt.BorderLayout;
import java.util.Comparator;
import java.util.Random;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.RowFilter;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;

/**
 * @author usu
 */
public class TableSorterFilter extends JTable {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final JFrame frame = new JFrame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setLayout(new BorderLayout());

                final TableSorterFilter table = new TableSorterFilter();
                frame.add(new JScrollPane(table));

                final JTextField text = new JTextField();
                text.getDocument().addDocumentListener(new DocumentListener() {

                    public void changedUpdate(DocumentEvent e) {
                        table.setFilter(text.getText());
                    }

                    public void insertUpdate(DocumentEvent e) {
                        table.setFilter(text.getText());
                    }

                    public void removeUpdate(DocumentEvent e) {
                        table.setFilter(text.getText());
                    }
                });
                frame.add(text, BorderLayout.NORTH);

                frame.setSize(400, 300);
                frame.setVisible(true);
            }
        });
    }

    private final Random generator;
    private final TableRowSorter<DefaultTableModel> sorter;

    public TableSorterFilter() {
        super();
    }
}
```

```
this.generator = new Random();

final DefaultTableModel model = new DefaultTableModel();
model.addColumn("Nama");
model.addColumn("Umur");
for (int i = 0; i < 100; i++) {
    model.addRow(new Object[] { "Eko Kurniawan Khannedy",
        new Integer(this.generator.nextInt(40)) });
}

this.sorter = new TableRowSorter<DefaultTableModel>(model);
this.sorter.setComparator(1, new Comparator<Integer>() {

    public int compare(final Integer o1, final Integer o2) {
        return o1.compareTo(o2);
    }
} );

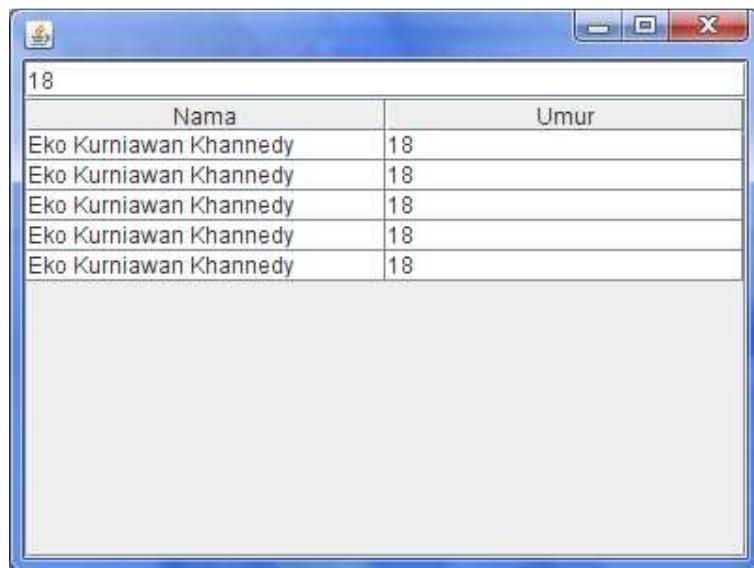
setModel(model);
setRowSorter(this.sorter);
}

public void setFilter(final String key) {
    this.sorter.setRowFilter(RowFilter.regexFilter(key));
}
}
```

| Nama | Umur |
|------------------------|------|
| Eko Kurniawan Khannedy | 33 |
| Eko Kurniawan Khannedy | 9 |
| Eko Kurniawan Khannedy | 3 |
| Eko Kurniawan Khannedy | 16 |
| Eko Kurniawan Khannedy | 29 |
| Eko Kurniawan Khannedy | 3 |
| Eko Kurniawan Khannedy | 5 |
| Eko Kurniawan Khannedy | 18 |
| Eko Kurniawan Khannedy | 17 |
| Eko Kurniawan Khannedy | 33 |
| Eko Kurniawan Khannedy | 20 |
| Eko Kurniawan Khannedy | 32 |
| Eko Kurniawan Khannedy | 6 |
| Eko Kurniawan Khannedy | 1 |

Gambar 154 TableSorterFilter.java

Sekarang jika kita inputkan sebuah key ke textfield maka otomatis RowSorter akan memfilter JTable :



Gambar 155 TableSorterFilter saat difilter

Microsoft Excel dan OpenOffice Calc

Kali ini kita akan beralih ke `TableModel`. Kita akan mencoba mengkomunikasikan `JTable` dengan MS Excel dan OO Calc. Misal kasusnya kita mau membuat laporan dari sebuah `JTable` dengan menggunakan Excel atau Calc. Dan salah satu teknik yang bisa kita gunakan adalah lewat file CSV (Comma Separate Values).

Sebelumnya kita buat library untuk mengexport ke CSV :

TableTool.java

```
package pelajaran12;

import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import javax.swing.table.TableModel;

/**
 * @author usu
 */
public class TableTool {

    public static final void exportToCSV(final TableModel model,
                                         final File target, final String separator) throws IOException {

        final PrintWriter writer = new PrintWriter(target);

        for (int j = 0; j < model.getColumnCount(); j++) {
            writer.print(model.getColumnName(j));
            writer.print(separator);
        }

        writer.print("\n");

        for (int i = 0; i < model.getRowCount(); i++) {
            writer.print("\t");
            for (int j = 0; j < model.getColumnCount(); j++) {
                writer.print(model.getValueAt(i, j));
                writer.print(separator);
            }
            writer.print("\n");
        }
    }
}
```

```
        for (int k = 0; k < model.getColumnCount(); k++) {
            writer.print(model.getValueAt(j, k));
            writer.print(separator);
        }
        writer.print("\n");
    }

    writer.close();
}

}
```

Dan sekarang kita buat JTable yang memanfaatkan TableTool.java

TableCSV.java

```
package pelajaran12;

import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.IOException;
import java.util.Comparator;
import java.util.Random;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;

/**
 * @author usu
 */
public class TableCSV extends JTable {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final JFrame frame = new JFrame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setLayout(new BorderLayout());

                final TableCSV table = new TableCSV();
                frame.add(new JScrollPane(table));

                final JFileChooser chooser = new JFileChooser();

                final JButton button = new JButton("EXPORT");
                button.addActionListener(new ActionListener() {

                    public void actionPerformed(ActionEvent e) {
                        if (chooser.showSaveDialog(frame) ==
                            JFileChooser.APPROVE_OPTION) {


```

```
        try {
            File f = chooser.getSelectedFile();
            f = new File(f.getPath() + ".CSV");
            TableTool.exportToCSV(table.getModel(), f, "\t");
        } catch (final IOException ex) {
            // ERROR
        }
    }
}

frame.add(button, BorderLayout.NORTH);

frame.setSize(400, 300);
frame.setVisible(true);
}
);
}

private final Random generator;

private final TableRowSorter<DefaultTableModel> sorter;

public TableCSV() {
    super();

    this.generator = new Random();

    final DefaultTableModel model = new DefaultTableModel();
    model.addColumn("Nama");
    model.addColumn("Umur");
    for (int i = 0; i < 100; i++) {
        model.addRow(new Object[] { "Eko Kurniawan Khannedy",
            new Integer(this.generator.nextInt(40)) });
    }

    this.sorter = new TableRowSorter<DefaultTableModel>(model);
    this.sorter.setComparator(1, new Comparator<Integer>() {

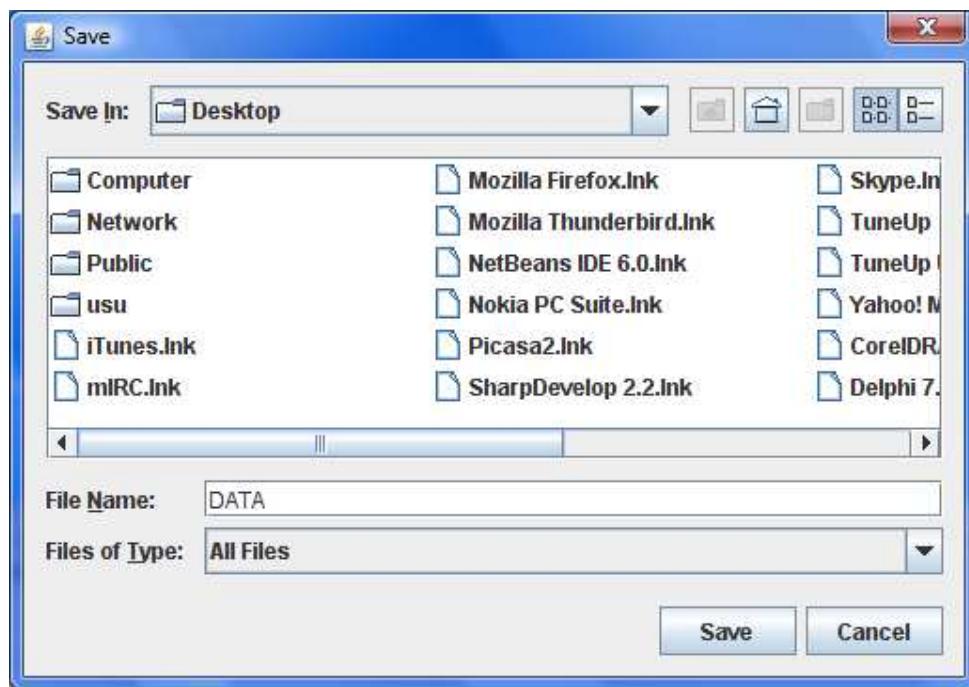
        public int compare(final Integer o1, final Integer o2) {
            return o1.compareTo(o2);
        }
    });
    setModel(model);
    setRowSorter(this.sorter);
}
}
```

Untuk kode diatas, saya gunakan separator “\t” yang artinya Tab, anda juga bisa menggunakan separator “ ” (spasi) atau yang sekiranya baik untuk anda gunakan.

| Nama | Umur |
|------------------------|------|
| Eko Kurniawan Khannedy | 22 |
| Eko Kurniawan Khannedy | 2 |
| Eko Kurniawan Khannedy | 15 |
| Eko Kurniawan Khannedy | 24 |
| Eko Kurniawan Khannedy | 8 |
| Eko Kurniawan Khannedy | 33 |
| Eko Kurniawan Khannedy | 6 |
| Eko Kurniawan Khannedy | 7 |
| Eko Kurniawan Khannedy | 33 |
| Eko Kurniawan Khannedy | 35 |
| Eko Kurniawan Khannedy | 7 |
| Eko Kurniawan Khannedy | 31 |
| Eko Kurniawan Khannedy | 31 |
| Eko Kurniawan Khannedy | 30 |

Gambar 156 TableCSV.java

Jika kita tekan tombol EXPORT, maka akan keluar FileChooser :



Gambar 157 Save Dialog

Lalu anda save data CSV-nya, misal saya simpan di Desktop dengan nama DATA.

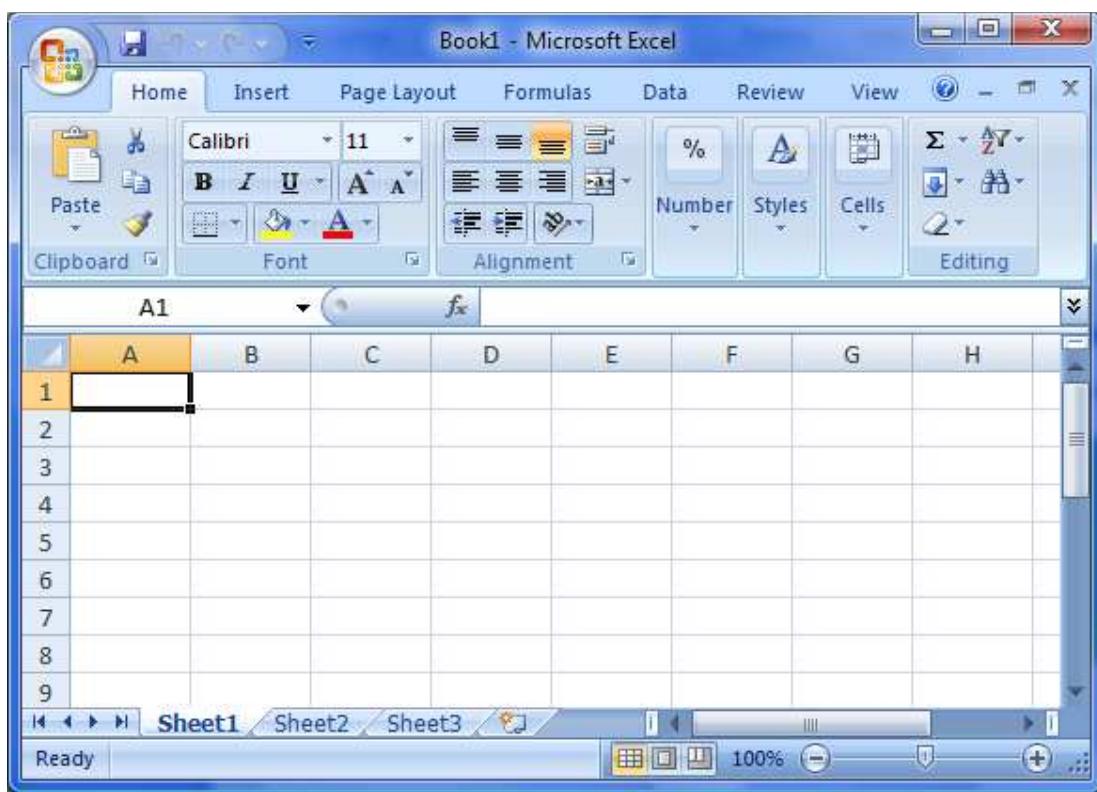


Gambar 158 File CSV

Sekarang bisa kita lihat filenya di Desktop.

Microsoft Office Excel

Sekarang bagaimana caranya menggunakan file tadi ke Excel. Pertama buka aplikasi MS Excel, saya menggunakan MS Excel 2007 :



Gambar 159 Microsoft Offive Excel 2007

Lalu masuk ke tab Data > Get External Data :



Gambar 160 Microsoft Office Excel 2007

Lalu pilih "From Text"



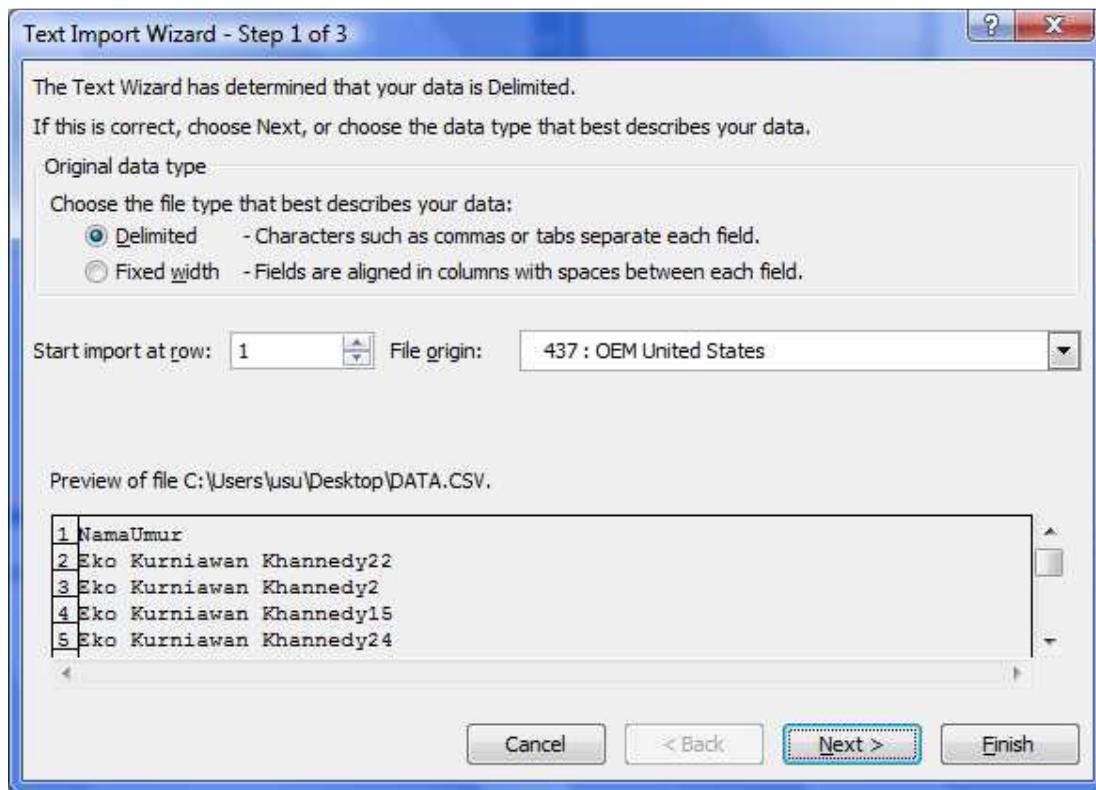
Gambar 161 Get External Data

Dan pilih file tadi :



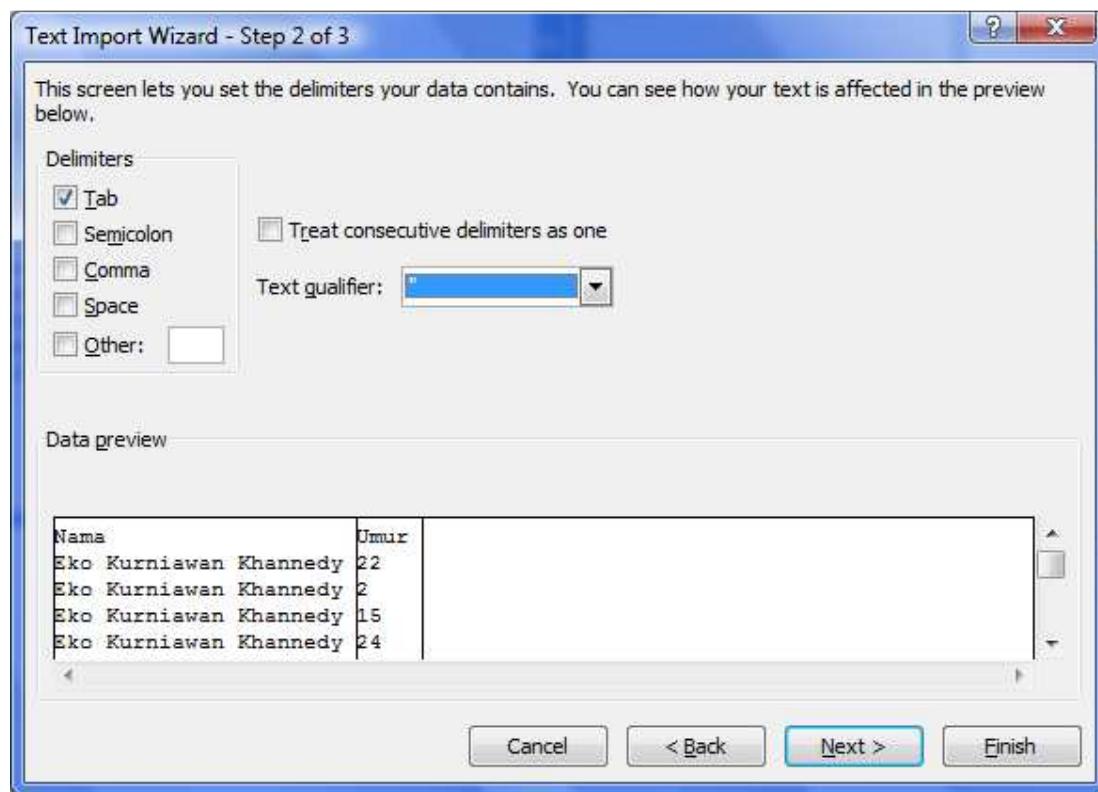
Gambar 162 Import Text File

Sekarang akan tampil dialog CSV : Pilih “Delimited” dan Next :



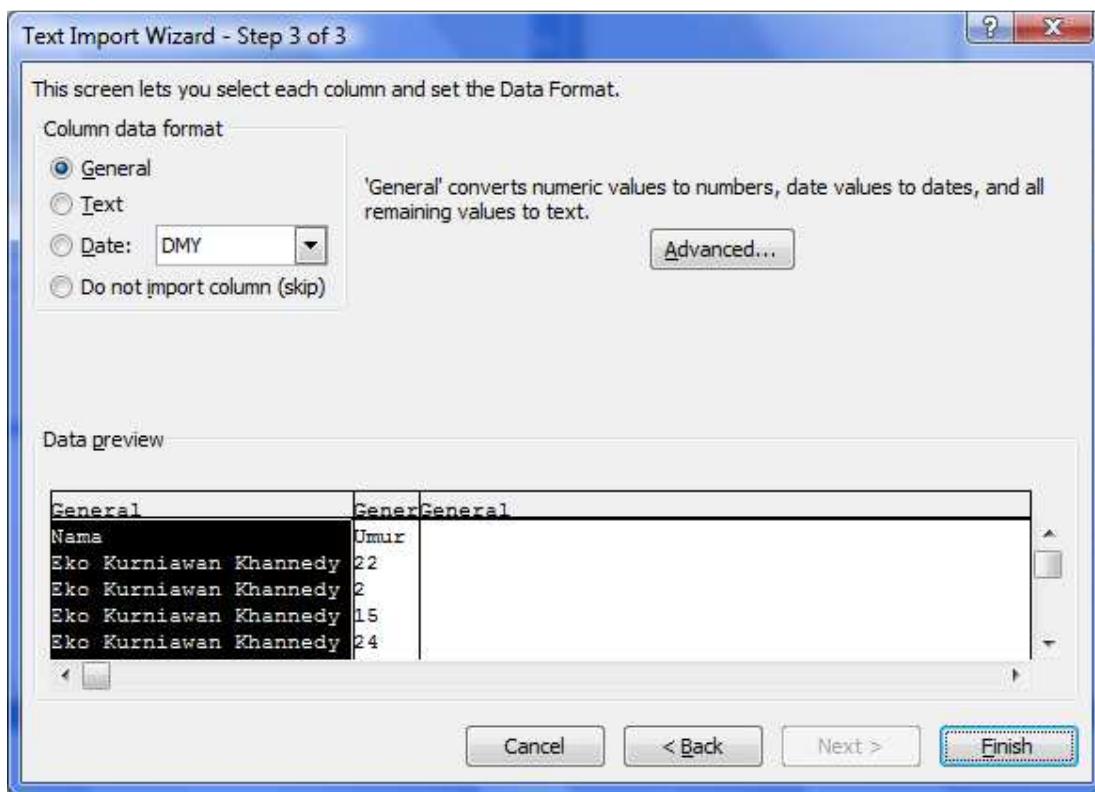
Gambar 163 Text Import Wizard

Ceklis "Tab", atau sesuai dengan separator yang tadi anda gunakan :



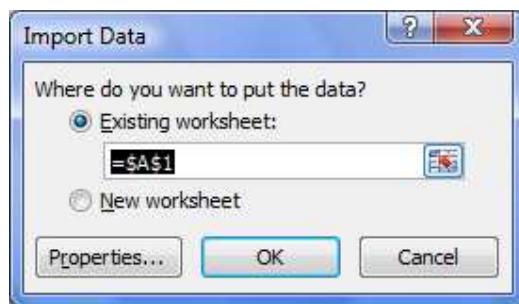
Gambar 164 Text Import Wizard

Tentukan jenis data sesuai dengan yang anda inginkan :



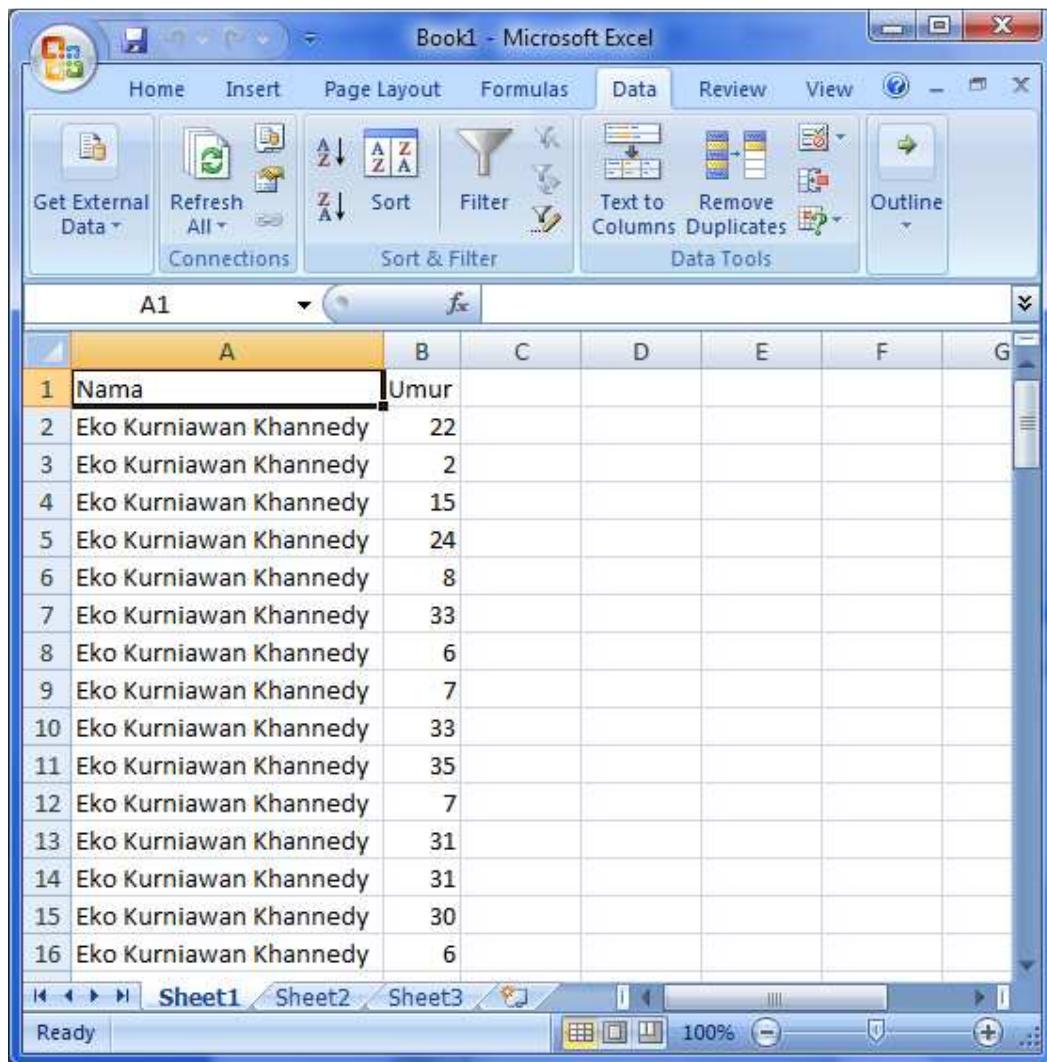
Gambar 165 Text Import Wizard

Dan klik "Finish", dan tentukan lokasi tempat tabel CSV akan dibuat :



Gambar 166 Import Data

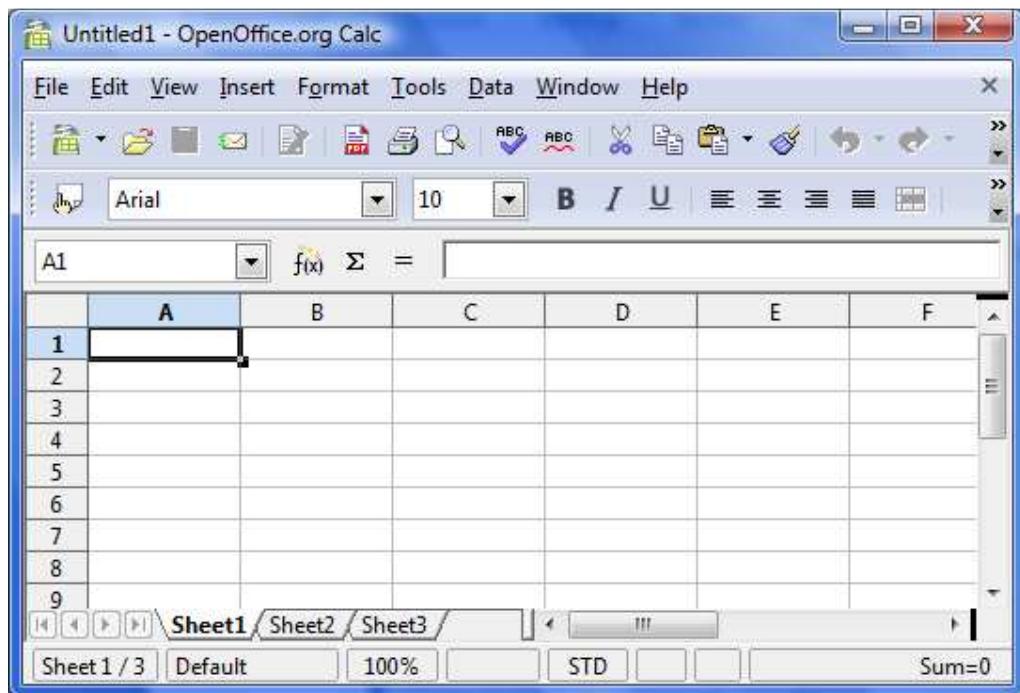
Dan OK, sekarang data dalam JTable tadi telah ada dalam MS Excel :



Gambar 167 Microsoft Office Excel 2007

OpenOffice Calc

Sekarang kita akan menggunakan OpenOffice Calc untuk membuka file CSV tadi. Pertama – tama buka aplikasi OpenOffice Calc :



Gambar 168 OpenOffice Calc

Lalu buka file CSV tadi lewat menu File > Open :



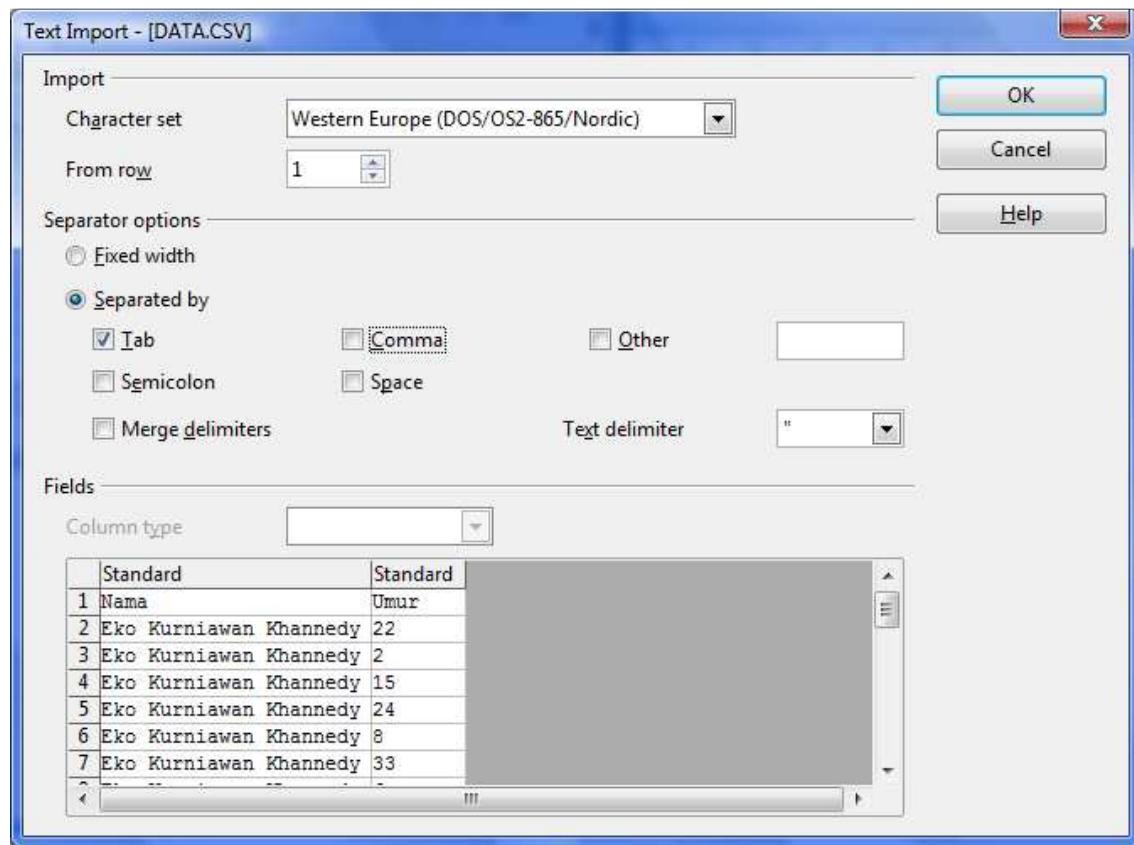
Gambar 169 Menu Open

Dan pilih file CSVnya :



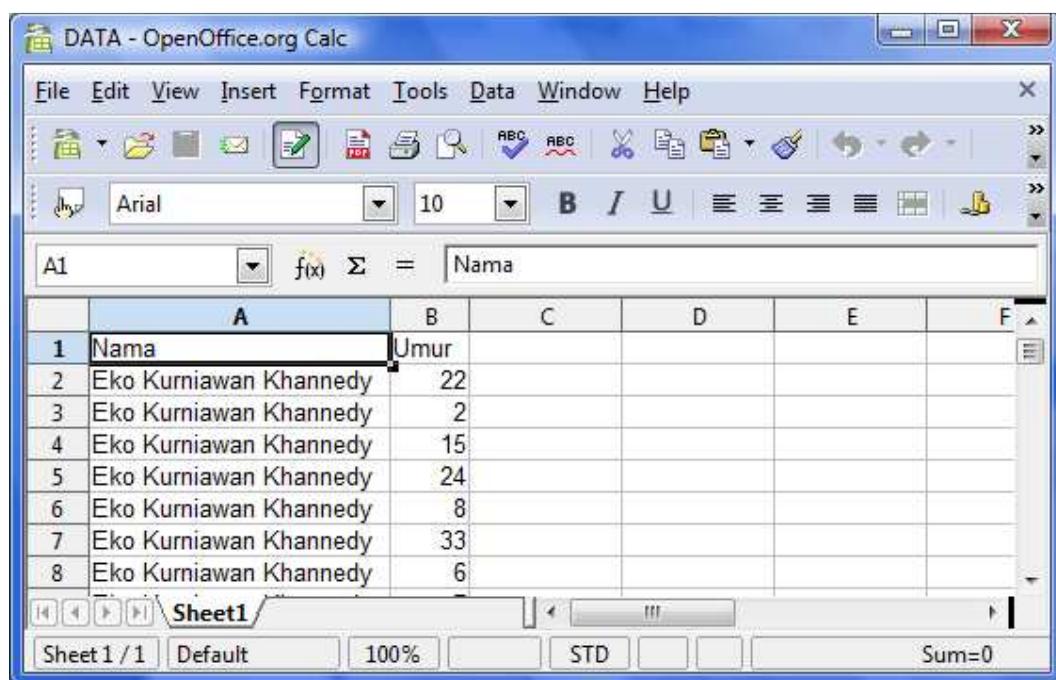
Gambar 170 Open Dialog

Sekarang akan keluar tampilan setting. Setting lah sesuai dengan yang anda inginkan tapi jangan lupa untuk menentukan “separated by” dengan separator yang tadi anda gunakan, contohnya tadi saya menggunakan Tab :



Gambar 171 Text Import

Dan OK :



Gambar 172 OpenOffice Calc

Sekarang data dalam JTable tadi ada di OpenOffice Calc.

HTML

Sekarang kita akan membahas trik untuk mengexport data dari JTable ke sebuah file HTML. Ok supaya tak banyak file yang kita gunakan, lebih baik kita edit TableTool.java agar dapat mengexport data dari JTable ke HTML.

TableTool.java

```
package pelajaran12;

import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import javax.swing.table.TableModel;

/**
 * @author usu
 */
public class TableTool {

    public static final void exportToCSV(final TableModel model,
                                         final File target, final String separator) throws IOException {
        final PrintWriter writer = new PrintWriter(target);

        for (int j = 0; j < model.getColumnCount(); j++) {
            writer.print(model.getColumnName(j));
            writer.print(separator);
        }

        writer.print("\n");

        for (int j = 0; j < model.getRowCount(); j++) {
            for (int k = 0; k < model.getColumnCount(); k++) {
                writer.print(model.getValueAt(j, k));
                writer.print(separator);
            }
            writer.print("\n");
        }

        writer.close();
    }

    public static final void exportToHTML(final TableModel model,
                                         final File target) throws IOException {
        final PrintWriter writer = new PrintWriter(target);

        writer.write("<HTML><BODY><TABLE BORDER ='1'>");
        writer.write("<TR>");
        for (int j = 0; j < model.getColumnCount(); j++) {
            writer.write("<TD>");
            writer.print(model.getColumnName(j));
            writer.write("</TD>");
        }
    }
}
```

```
writer.write("</TR>");

    for (int j = 0; j < model.getRowCount(); j++) {
        writer.write("<TR>");
        for (int k = 0; k < model.getColumnCount(); k++) {
            writer.write("<TD>");
            writer.print(model.getValueAt(j, k));
            writer.write("</TD>");
        }
        writer.write("</TR>");
    }

    writer.write("</TABLE></BODY></HTML>");
    writer.close();
}

}
```

Ok dan sekarang tinggal buat JTablenya :

TableHTML.java

```
package pelajaran12;

import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.IOException;
import java.util.Comparator;
import java.util.Random;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;

/**
 * @author usu
 */
public class TableHTML extends JTable {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                final JFrame frame = new JFrame();
                frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                frame.setLayout(new BorderLayout());

                final TableHTML table = new TableHTML();
                frame.add(new JScrollPane(table));

                final JFileChooser chooser = new JFileChooser();

```

```
final JButton button = new JButton("EXPORT");
button.addActionListener(new ActionListener() {

    public void actionPerformed(final ActionEvent e) {
        if (chooser.showSaveDialog(frame) ==
JFileChooser.APPROVE_OPTION) {
            try {
                File f = chooser.getSelectedFile();
                f = new File(f.getPath() + ".HTML");
                TableTool.exportToHTML(table.getModel(), f);
            } catch (final IOException ex) {
                // ERROR
            }
        }
    }
}) ;

frame.add(button, BorderLayout.NORTH);

frame.setSize(400, 300);
frame.setVisible(true);
}
);
}

private final Random generator;

private final TableRowSorter<DefaultTableModel> sorter;

public TableHTML() {
super();

this.generator = new Random();

final DefaultTableModel model = new DefaultTableModel();
model.addColumn("Nama");
model.addColumn("Umur");
for (int i = 0; i < 100; i++) {
    model.addRow(new Object[] { "Eko Kurniawan Khannedy",
        new Integer(this.generator.nextInt(40)) });
}

this.sorter = new TableRowSorter<DefaultTableModel>(model);
this.sorter.setComparator(1, new Comparator<Integer>() {

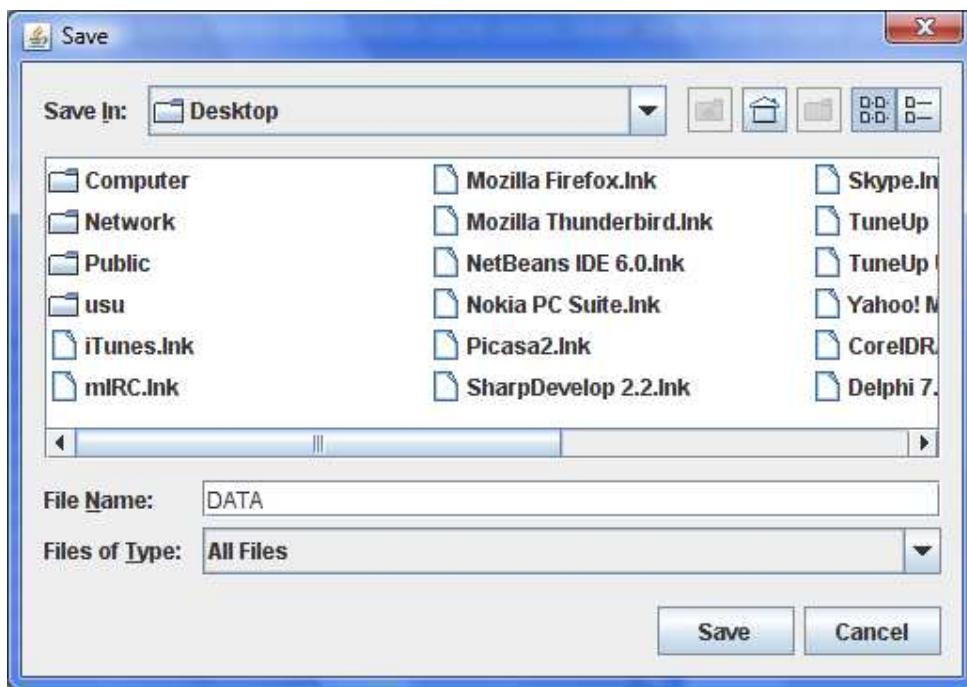
    public int compare(final Integer o1, final Integer o2) {
        return o1.compareTo(o2);
    }
});

setModel(model);
setRowSorter(this.sorter);
}
}
```

| Nama | Umur |
|------------------------|------|
| Eko Kurniawan Khannedy | 3 |
| Eko Kurniawan Khannedy | 13 |
| Eko Kurniawan Khannedy | 35 |
| Eko Kurniawan Khannedy | 15 |
| Eko Kurniawan Khannedy | 10 |
| Eko Kurniawan Khannedy | 27 |
| Eko Kurniawan Khannedy | 15 |
| Eko Kurniawan Khannedy | 27 |
| Eko Kurniawan Khannedy | 32 |
| Eko Kurniawan Khannedy | 2 |
| Eko Kurniawan Khannedy | 7 |
| Eko Kurniawan Khannedy | 13 |
| Eko Kurniawan Khannedy | 11 |
| Eko Kurniawan Khannedy | 1 |

Gambar 173 TableHTML.java

Klik tombol EXPORT, maka akan ada FileChoser untuk mensave file HTML.



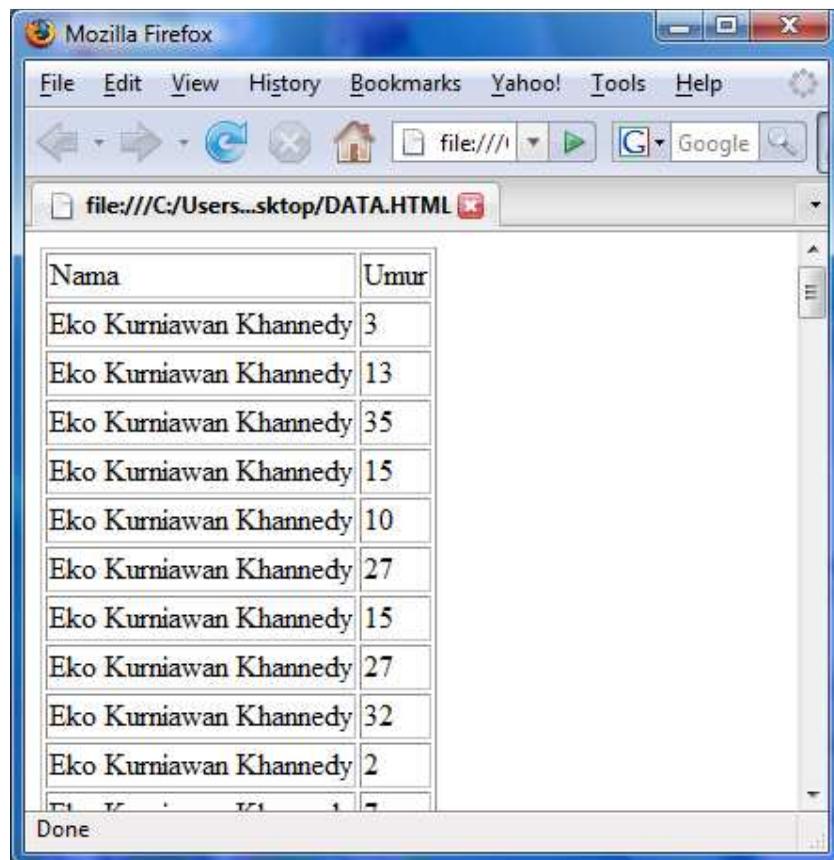
Gambar 174 Save Dialog

Misal saya save file htmlnya di Desktop, maka akan terlihat file hasil savennya di Desktop :



Gambar 175 File HTML

Sekarang tinggal anda buka file tersebut menggunakan browser yang anda miliki :



| Nama | Umur |
|------------------------|------|
| Eko Kurniawan Khannedy | 3 |
| Eko Kurniawan Khannedy | 13 |
| Eko Kurniawan Khannedy | 35 |
| Eko Kurniawan Khannedy | 15 |
| Eko Kurniawan Khannedy | 10 |
| Eko Kurniawan Khannedy | 27 |
| Eko Kurniawan Khannedy | 15 |
| Eko Kurniawan Khannedy | 27 |
| Eko Kurniawan Khannedy | 32 |
| Eko Kurniawan Khannedy | 2 |

Gambar 176 Mozilla Firefox

Kesimpulan

Huh, mungkin JTable adalah salah satu bagian yang paling sulit untuk diexpose, karena memang terlalu banyak trix yang bisa kita buat menggunakan JTable. Jadi mulai saat ini buatlah render dan editor yang interaktif untuk JTable anda, agar end user lebih dimanjakan oleh aplikasi yang anda buat.

Pelajaran 13

JTextPane

Salah satu perbedaan JTextPane dan JTextArea adalah format Document. JTextArea menggunakan PlainDocument sedangkan JTextPane menggunakan StyledDocument. Nah makanya JTextArea hanya bisa menampung satu style berbeda dengan JTextPane yang bisa menampung banyak style.

Highlight

OK salah satu kelebihan JTextPane adalah Highlight. Sekarang kita akan mewarnai teks yang ada di dalam JTextPane dengan hightlight.

Untuk menambah hightlight ke JTextPane gunakan kode ini :

```
textPane.getHighlighter().addHighlight(int p1, int p2, HighlightPainter painter);
```

Dan untuk menghapus seluruh hightlight :

```
textPane.getHighlighter().removeAllHighlights();
```

Sekarang kita buat contohnya.

TextPaneHightLight.java

```
package pelajaran13;

import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTextField;
import javax.swing.JTextPane;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;
import javax.swing.text.BadLocationException;
import javax.swing.text.DefaultHighlightPainter;

/**
 * @author usu
 */
public class TextPaneHightLight extends JFrame {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                new TextPaneHightLight().setVisible(true);
            }
        });
    }
}
```

```
public static final void setHightLight(final String key,
    final JTextPane pane, final Color color) {
    try {
        int index = 0;
        final String total = pane.getDocument().getText(0,
            pane.getDocument().getLength());

        pane.getHighlighter().removeAllHighlights();
        while ((index = total.indexOf(key, index)) > 0) {
            final DefaultHighlightPainter painter = new
DefaultHighlightPainter(
                color);
            pane.getHighlighter().addHighlight(index, index + key.length(),
                painter);
            index += key.length();
        }
    } catch (final BadLocationException ex) {
        // ERROR
    }
}

public TextPaneHightLight() {
    super();

    final JTextPane textPane = new JTextPane();

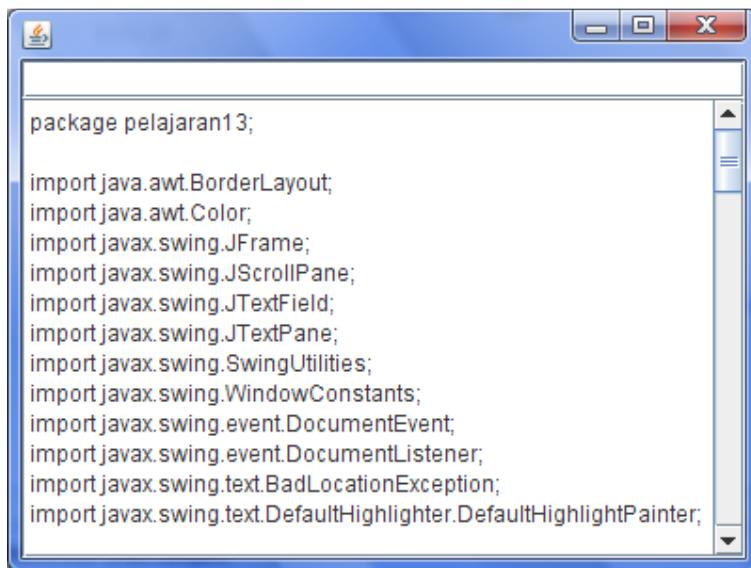
    final JTextField text = new JTextField();
    text.getDocument().addDocumentListener(new DocumentListener() {

        public void changedUpdate(final DocumentEvent e) {
            TextPaneHightLight.setHightLight(text.getText(), textPane,
                Color.GREEN);
        }

        public void insertUpdate(final DocumentEvent e) {
            TextPaneHightLight.setHightLight(text.getText(), textPane,
                Color.GREEN);
        }

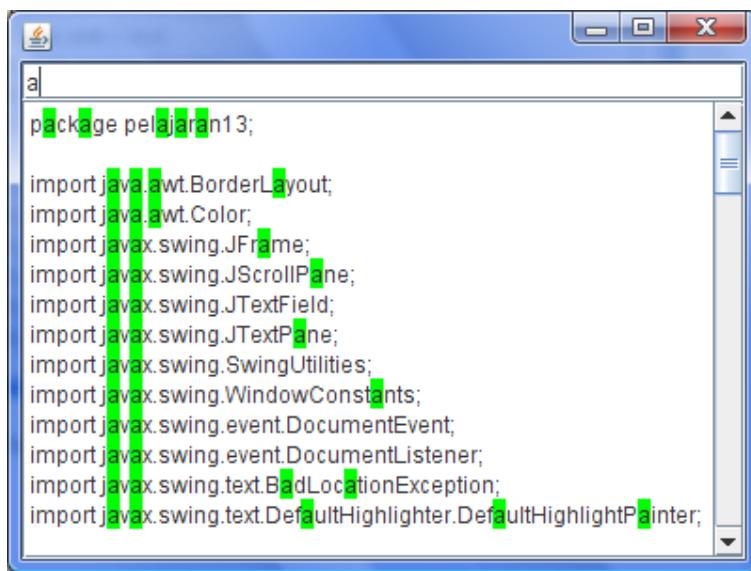
        public void removeUpdate(final DocumentEvent e) {
            TextPaneHightLight.setHightLight(text.getText(), textPane,
                Color.GREEN);
        }
    });

    setLayout(new BorderLayout());
    add(text, BorderLayout.NORTH);
    add(new JScrollPane(textPane));
    setSize(400, 300);
    setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
}
}
```



Gambar 177 TextPaneHightLight.java

Sekarang kita inputkan huruf "a", maka hightligter akan mewarnai huruf "a" :

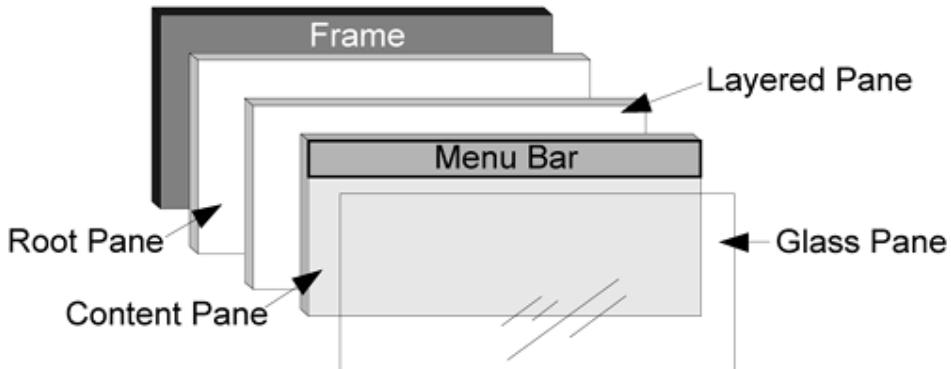


Gambar 178 TextPaneHightLight saat hightlight

Pelajaran 14

GlassPane

Haha, inilah bab yang paling saya sukai! Kenapa? Karena GlassPane adalah salah satu keajaiban Swing yang jarang sarang temui di bahasa pemrograman visual yang lain. Bagi yang baru mendengar apa itu GlassPane mari kita lihat gambar dibawah ini :



Gambar 179 Posisi GlassPane

Jadi GlassPane itu merupakan salah satu bagian JFrame ataupun JDialog yang berada pada urutan paling atas, dan biasanya GlassPane itu transparan.

Gradient Glass

Sekarang contoh kita buat glasspane yang menampilkan gambar gradient.

GlassPaneGradient.java

```
package pelajaran14;

import java.awt.AlphaComposite;
import java.awt.Color;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import javax.swing.JPanel;

/**
 * @author usu
 */
public class GlassPaneGradient extends JPanel {

    public GlassPaneGradient() {
        super();
        setOpaque(false);
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);
    }
}
```

```
final GradientPaint paint = new GradientPaint(0, 0, Color.YELLOW,
    getWidth(), getHeight(), Color.GREEN);

final Graphics2D g2 = (Graphics2D) g.create();
g2.setPaint(paint);
g2.setComposite(AlphaComposite.SrcOver.derive(0.3F));
g2.fillRect(0, 0, getWidth(), getHeight());
g2.dispose();
}

}
```

Dan sekarang tinggal menggunakannya ke JFrame. Untuk menggunakan GlassPane gunakan kode seperti dibawah ini :

```
GlassPaneGradient glasspane = new GlassPaneGradient();
frame.setGlassPane(glasspane);
frame.getGlassPane().setVisible(true);
```

GlassPaneGradientApp.java



Gambar 180 GlassPaneGradientApp.java

GlassPaneGradient2.java

```
package pelajaran14;

import java.awt.AlphaComposite;
import java.awt.Color;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.GeneralPath;
import javax.swing.JPanel;

/**
 * @author usu
 */
public class GlassPaneGradient2 extends JPanel {

    public GlassPaneGradient2() {
        super();
        setOpaque(false);
    }
}
```

```
@Override
protected void paintComponent(final Graphics g) {
    super.paintComponent(g);

    final GradientPaint paint = new GradientPaint(0, 0, Color.YELLOW,
        getWidth(), getHeight(), Color.GREEN);
    final GeneralPath path = new GeneralPath();
    path.moveTo(0, 0);
    path.lineTo(0, getHeight());
    path.curveTo(0, getHeight(), getWidth() / 5, getHeight() / 5,
    getWidth(),
        0);
    path.closePath();

    final Graphics2D g2 = (Graphics2D) g.create();
    g2.setPaint(paint);
    g2.setComposite(AlphaComposite.SrcOver.derive(0.3F));
    g2.fill(path);
    g2.dispose();
}
}
```

GlassPaneGradient2App.java



Gambar 181 GlassPaneGradient2App.java

Gimana? Keren kan? Yach itulah kegunaan GlassPane, untuk membuat efek-efek diatas komponen – komponen.

Block Event

Selain sebagai hiasan, salah satu kemampuan GlassPane adalah memblock event untuk komponent – komponen yang adan di JFrame.

GlassPaneBlock.java

```
package pelajaran14;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.event.KeyAdapter;
import java.awt.event.MouseAdapter;
import javax.swing.JPanel;
```

```
/**  
 * @author usu  
 */  
public class GlassPaneBlock extends JPanel {  
  
    private final MouseAdapter adapter1 = new MouseAdapter() {  
    };  
    private final KeyAdapter adapter2 = new KeyAdapter() {  
    };  
    private boolean block;  
  
    public GlassPaneBlock() {  
        super();  
        setBlock(false);  
        setOpaque(false);  
    }  
  
    public boolean isBlock() {  
        return this.block;  
    }  
  
    @Override  
    protected void paintComponent(final Graphics g) {  
        super.paintComponent(g);  
  
        if (isBlock()) {  
            final Graphics2D g2 = (Graphics2D) g.create();  
            g2.setColor(new Color(1F, 1F, 1F, 0.6F));  
            g2.fillRect(0, 0, getWidth(), getHeight());  
            g2.dispose();  
        }  
    }  
  
    public void setBlock(final boolean block) {  
        this.block = block;  
        if (block) {  
            addMouseListener(this.adapter1);  
            addKeyListener(this.adapter2);  
        } else {  
            removeMouseListener(this.adapter1);  
            removeKeyListener(this.adapter2);  
        }  
        repaint();  
    }  
}
```

Untuk penggunaannya seperti glasspane sebelumnya. Dan untuk mem-bloknya gunakan :

Glasspane.setBlock(true);

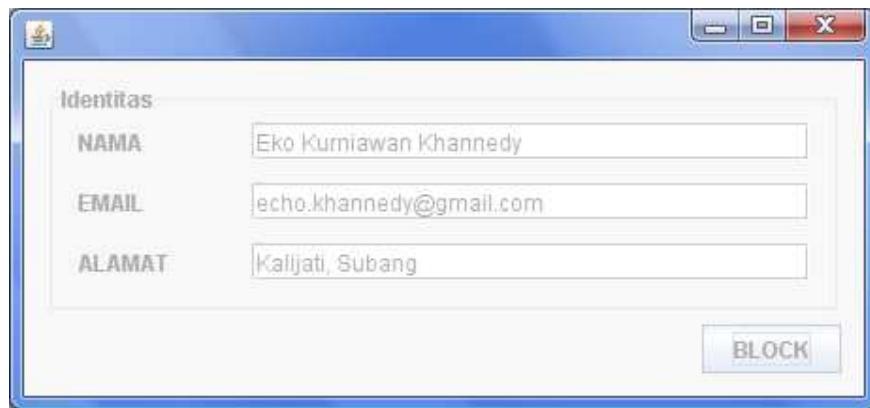
Dan untuk men-unblock gunakan :

Glasspane.setBlock(false);



Gambar 182 Form yang menggunakan GlassPaneBlock

Dan saat di block :



Gambar 183 Saat form diblock oleh GlassPaneBlock

GlassPaneBlock2.java

```
package pelajaran14;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.Point;
import java.awt.event.KeyAdapter;
import java.awt.event.MouseAdapter;
import javax.swing.ImageIcon;
import javax.swing.JPanel;

/**
 * @author usu
 */
public class GlassPaneBlock2 extends JPanel {

    private final MouseAdapter adapter1 = new MouseAdapter() {
    };
    private final KeyAdapter adapter2 = new KeyAdapter() {
    };
    private boolean block;
```

```
private final Image image;

public GlassPaneBlock2() {
    super();
    setBlock(false);
    setOpaque(false);
    this.image = new ImageIcon(getClass()
        .getResource("/pelajaran14/stop.png")).getImage();
}

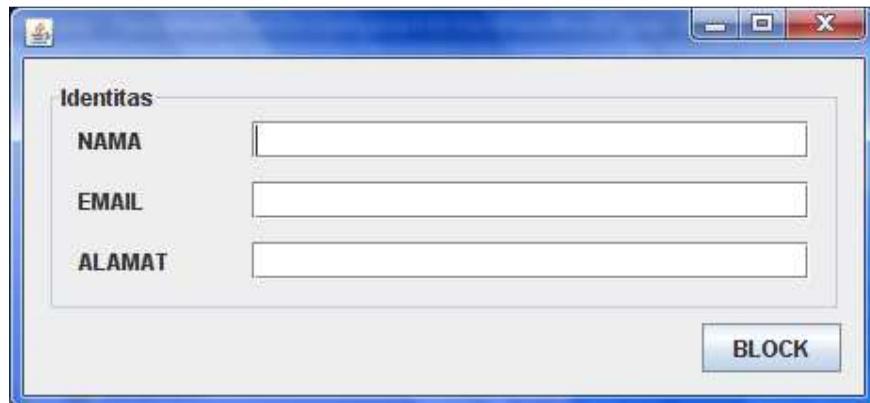
public boolean isBlock() {
    return this.block;
}

@Override
protected void paintComponent(final Graphics g) {
    super.paintComponent(g);

    if (isBlock()) {
        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setColor(new Color(1F, 1F, 1F, 0.6F));
        g2.fillRect(0, 0, getWidth(), getHeight());
        if (this.image != null) {
            final Point p = new Point(getWidth() -
this.image.getWidth(null),
                getHeight() - this.image.getHeight(null));
            g2.drawImage(this.image, p.x, p.y, null);
        }
        g2.dispose();
    }
}

public void setBlock(final boolean block) {
    this.block = block;
    if (block) {
        addMouseListener(this.adapter1);
        addKeyListener(this.adapter2);
    } else {
        removeMouseListener(this.adapter1);
        removeKeyListener(this.adapter2);
    }
    repaint();
}
}
```

GlassPaneBlock2App.java



Gambar 184 GlassPaneBlock2App.java



Gambar 185 GlassPaneBlock2App saat diblock

Pada saat diblock, seluruh event, baik itu mengklik button, mengetik textfield akan diblokir oleh GlassPane. Untuk source kode contoh aplikasi diatas bisa anda lihat di project SwingMakeOver.

Warning GlassPane

Begini artinya, glasspane dapat menampilkan gambar warning diatas sebuah komponen.

GlassPaneWarning.java

```
package pelajaran14;

import java.awt.Component;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.Point;
import java.util.ArrayList;
import javax.swing.ImageIcon;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;

/**
 * @author usu
 */
public class GlassPaneWarning extends JPanel {
```

```
private final ArrayList<Component> data = new ArrayList<Component>();
private final Image warning;

public GlassPaneWarning() {
    super();
    setOpaque(false);
    this.warning = new ImageIcon(getClass().getResource(
        "/pelajaran14/warning.png")).getImage();
}

public void addWarning(final Component e) {
    this.data.add(e);
    repaint();
}

@Override
protected void paintComponent(final Graphics g) {
    super.paintComponent(g);

    if (this.data.size() > 0) {
        for (final Component c : this.data) {
            if (this.warning != null) {
                final Point p = c.getLocationOnScreen();
                SwingUtilities.convertPointFromScreen(p, this);
                g.drawImage(this.warning, p.x - this.warning.getWidth(null),
                           p.y, null);
            }
        }
    }
}

public void removeWarning(final Object o) {
    this.data.remove(o);
    repaint();
}
}
```

Untuk menambah gambar warning di atas Component anda tinggal menggunakan kode :

Glasspane.addWarning(Component c);

Dan untuk menghilangkan warning, anda tinggal menggunakan kode :

Glasspane.removeWarning(Component c);

Contoh nya pada GlassPaneWarningApp.java yang dapat anda lihat di project SwingMakeOver.



Gambar 186 Form yang menggunakan GlassPaneWarning

Jika TextField telah terisi teks, maka gambar warning akan menghilang.



Gambar 187 Form yang menggunakan GlassPaneWarning

GlassPane Transition 1

Selain efek yang statis, glasspane juga bisa dibuat untuk efek yang dinamis. Contohnya kita akan membuat efek transisi menggunakan glasspane. Dan lagi – lagi kita sekarang memanfaatkan Timing FrameWork.

GlassPaneTransition.java

```
package pelajaran14;

import java.awt.AWTException;
import java.awt.AlphaComposite;
import java.awt.Component;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Point;
import java.awt.Rectangle;
import java.awt.Robot;
import java.awt.image.BufferedImage;

import javax.swing.JComponent;
import javax.swing.SwingUtilities;

import org.jdesktop.animation.timing.Animator;
import org.jdesktop.animation.timing.interpolation.PropertySetter;

/**
 * @author usu
 */
public class GlassPaneTransition extends JComponent {

    private float alpha;
    private final Animator animasi;
    private Component comp;
```

```

private BufferedImage image;
private Robot rbt;

public GlassPaneTransition() {
    super();
    this.alpha = 1.0f;
    this.animasi = new Animator(1000);
    this.animasi.addTarget(new PropertySetter(this, "alpha", 0.0f));
    this.animasi.setAcceleration(0.2f);
    this.animasi.setDeceleration(0.4f);
}

public float getAlpha() {
    return this.alpha;
}

public Robot getRobot() throws AWTException {
    if (this.rbt == null) {
        this.rbt = new Robot();
    }
    return this.rbt;
}

public boolean isProgress() {
    return this.animasi.isRunning();
}

@Override
protected void paintComponent(final Graphics g) {
    super.paintComponent(g);
    if (this.image != null) {
        final Graphics2D g2 = (Graphics2D) g.create();
        final Point p = SwingUtilities.convertPoint(this.comp, 0, 0,
this);
        g2.setComposite(AlphaComposite.SrcOver.derive(this.alpha));
        g2.drawImage(this.image, (int) p.getX(), (int) p.getY(), null);
    }
}

public void setAlpha(final float alpha) {
    this.alpha = alpha;
    repaint();
    if (alpha < 0.1) {
        setVisible(false);
    }
}

public void setTime(final int time) {
    this.animasi.setDuration(time);
}

public void startTransition(final Component comp) throws AWTException {
    if (comp == null) {
        return;
    }
    if (!comp.isVisible()) {
        return;
    }
    if (this.animasi.isRunning() == false) {
        setVisible(true);
        this.alpha = 1.0f;
    }
}

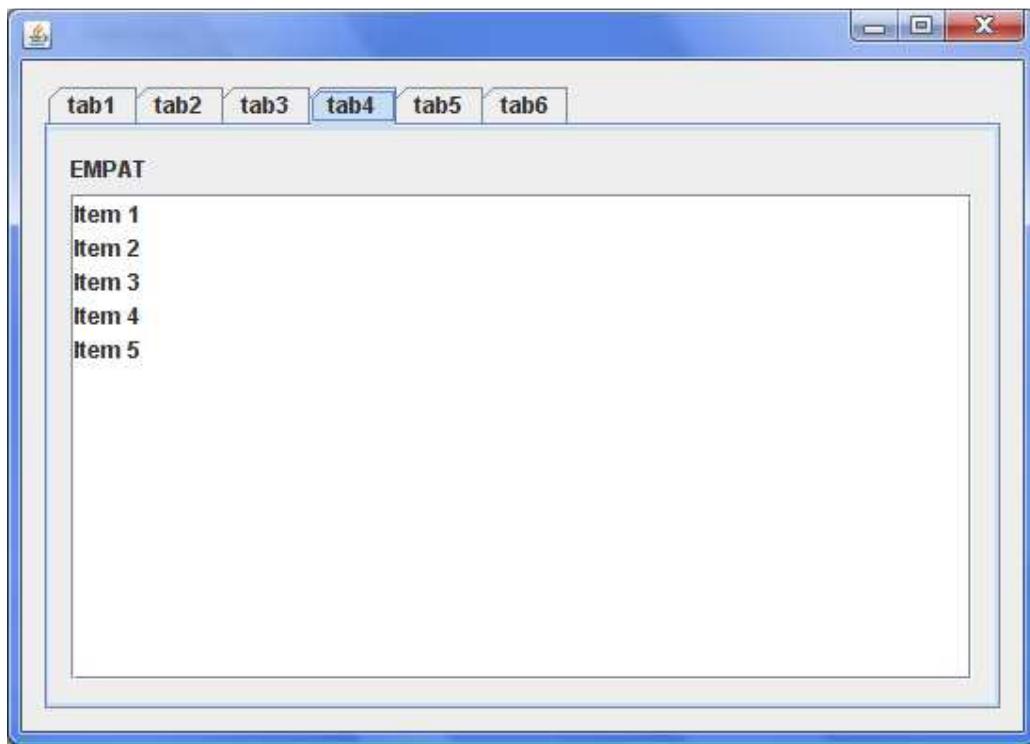
```

```
        this.comp = comp;
        this.image = getRobot().createScreenCapture(
            new Rectangle(this.comp.getLocationOnScreen(), this.comp
                .getSize()));
        this.animasi.start();
    }
}
```

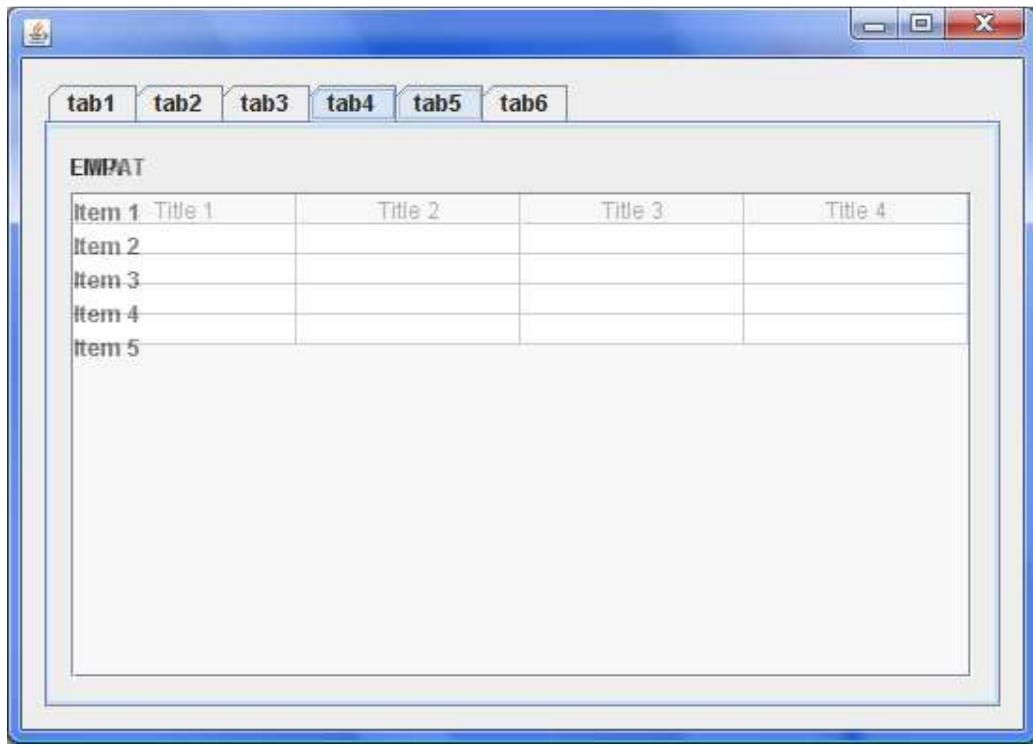
Dan untuk melakukan transisi, kita cukup menggunakan metode :

```
Glasspane.startTransition(Component c);
```

Maka secara otomatis Glasspane akan melakukan transisi diatas component tersebut.



Gambar 188 Form yang menggunakan GlassPaneTransition



Gambar 189 Saat GlassPaneTransition memulai transisi

Anda bisa melihat proses transisinya pada gambar ke 2, ketika dari tab4 akan berpindah ke tab5. Untuk source code diatas anda bisa lihat di project SwingMakeOver yang bernama GlassPaneTransitionApp1.java.

GlassPane Transition 2

Selain transisi alpha, Glasspane juga bisa melakukan transisi rotasi.

GlassPaneTransitionRotasi.java

```
package pelajaran14;

import java.awt.AWTException;
import java.awt.AlphaComposite;
import java.awt.Component;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Point;
import java.awt.Rectangle;
import java.awt.Robot;
import java.awt.image.BufferedImage;
import javax.swing.JComponent;
import javax.swing.SwingUtilities;
import org.jdesktop.animation.timing.Animator;
import org.jdesktop.animation.timing.interpolation.PropertySetter;

/**
 * @author usu
 */
public class GlassPaneTransitionRotasi extends JComponent {
```

```
private float alpha;
private final Animator animasi;
private Point centerPoint;
private Component comp;
private BufferedImage image;
private Robot rbt;

public GlassPaneTransitionRotasi() {
    super();
    this.alpha = 1.0f;
    this.animasi = new Animator(1000);
    this.animasi.addTarget(new PropertySetter(this, "alpha", 0.0f));
    this.animasi.setAcceleration(0.2f);
    this.animasi.setDeceleration(0.4f);
}

public float getAlpha() {
    return this.alpha;
}

public Robot getRobot() throws AWTException {
    if (this.rbt == null) {
        this.rbt = new Robot();
    }
    return this.rbt;
}

public boolean isProgress() {
    return this.animasi.isRunning();
}

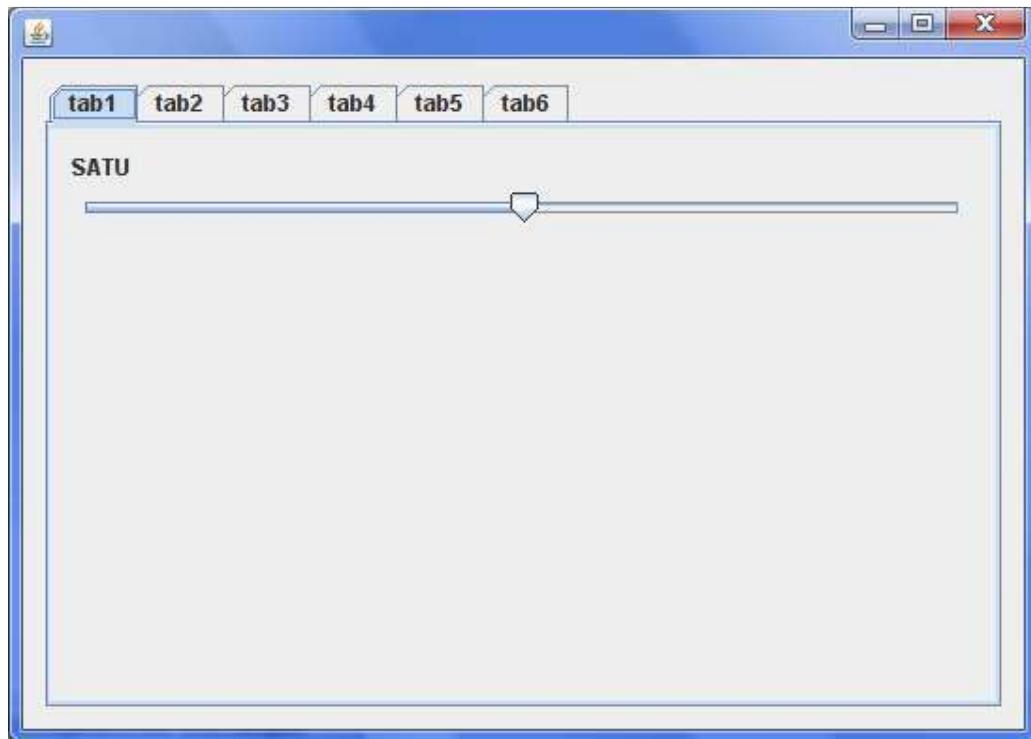
@Override
protected void paintComponent(final Graphics g) {
    super.paintComponent(g);
    if (this.image != null) {
        final Graphics2D g2 = (Graphics2D) g.create();
        final Point p = SwingUtilities.convertPoint(this.comp, 0, 0,
this);
        final Point p2 = SwingUtilities.convertPoint(this.comp,
            this.centerPoint, this);
        g2.rotate(Math.toRadians((1 - this.alpha) * 100), p2.x, p2.y);
        g2.setComposite(AlphaComposite.SrcOver.derive(this.alpha));
        g2.drawImage(this.image, (int) p.getX(), (int) p.getY(), null);
    }
}

public void setAlpha(final float alpha) {
    this.alpha = alpha;
    repaint();
    if (alpha < 0.1) {
        setVisible(false);
    }
}

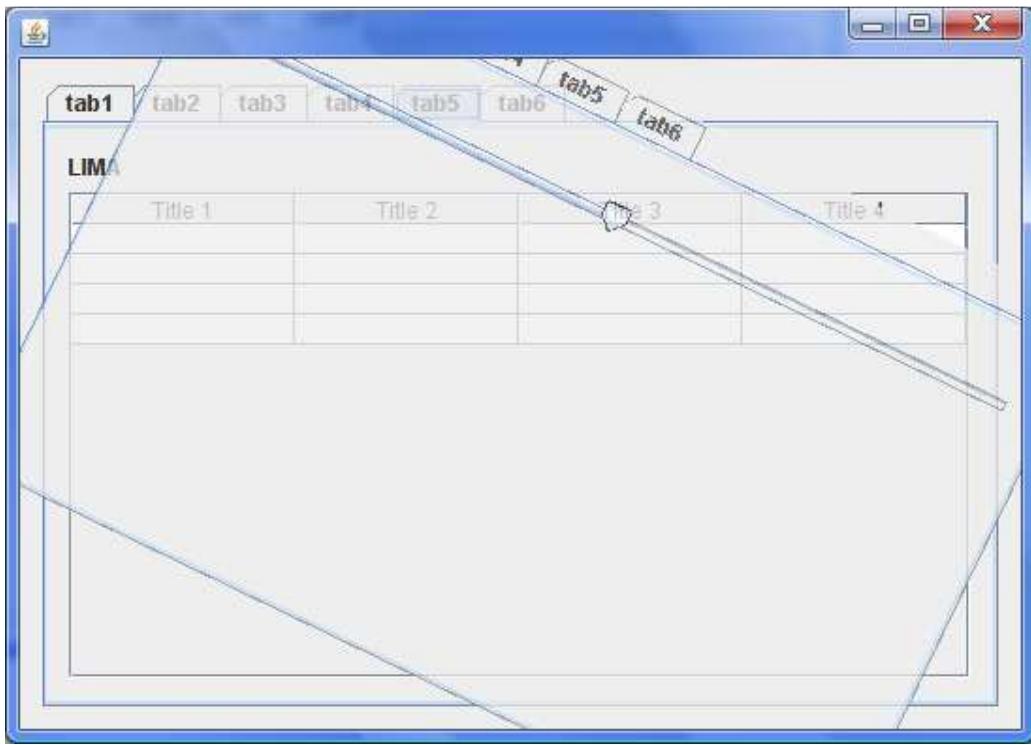
public void setTime(final int time) {
    this.animasi.setDuration(time);
}

public void startTransition(final Component comp) throws AWTException {
    if (comp == null) {
```

```
        return;
    }
    if (!comp.isVisible()) {
        return;
    }
    if (this.animasi.isRunning() == false) {
        setVisible(true);
        this.alpha = 1.0f;
        this.comp = comp;
        this.centerPoint = new Point(comp.getX() + comp.getWidth() / 2,
comp
            .getY()
            + comp.getHeight() / 2);
        this.image = getRobot().createScreenCapture(
            new Rectangle(this.comp.getLocationOnScreen(), this.comp
                .getSize()));
        this.animasi.start();
    }
}
```



Gambar 190 Form yang menggunakan GlassPaneTransitionRotasi



Gambar 191 Saat GlassPaneTransitionRotasi memulai transisi

Saat proses transisi, maka Glasspane akan memperlihatkan transisi secara rotasi. Anda bisa melihat source codenya di project SwingMakeOver pada file GlassPaneTransitionAppRotation.java.

Sebenarnya masih banyak lagi teknik transisi yang bisa digunakan menggunakan GlassPane, tapi saya gak bisa menulisnya satu – persatu karena terlalu banyak, saya sebut saja misalnya transisi menghilang ke kiri, kanan, atas, bawah atau diagonal.

GlassPane dan JTable

Ada satu trix yang bisa kita gunakan untuk JTable dan GlassPane, yaitu transisi sorter. Jadi ketika melakukan pengurutan terhadap JTable maka efek GlassPane khususnya transisi akan tampil.

TableTransition.java

```
package pelajaran14;

import java.awt.AWTException;
import javax.swing.event.RowSorterEvent;
import java.awt.BorderLayout;
import java.util.Comparator;
import java.util.Random;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import javax.swing.event.RowSorterListener;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;
```

```
/*
 * @author usu
 */
public class TableTransition extends JFrame {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                new TableTransition().setVisible(true);
            }
        });
    }

    private final Random generator;
    private final GlassPaneTransition glasspane;
    private final TableRowSorter<DefaultTableModel> sorter;

    private final JTable table;

    public TableTransition() {
        super();

        this.table = new JTable();
        this.generator = new Random();
        this.glasspane = new GlassPaneTransition();

        final DefaultTableModel model = new DefaultTableModel();
        model.addColumn("Nama");
        model.addColumn("Umur");
        for (int i = 0; i < 100; i++) {
            model.addRow(new Object[] { "Eko Kurniawan Khannedy",
                new Integer(this.generator.nextInt(40)) });
        }

        this.sorter = new TableRowSorter<DefaultTableModel>(model);
        this.sorter.setComparator(1, new Comparator<Integer>() {

            public int compare(final Integer o1, final Integer o2) {
                return o1.compareTo(o2);
            }
        });
        this.sorter.addRowSorterListener(new RowSorterListener() {

            public void sorterChanged(final RowSorterEvent e) {
                try {
                    TableTransition.this.glasspane
                        .startTransition(TableTransition.this.table);
                } catch (final AWTException ex) {
                    // ERROR
                }
            }
        });
    }

    this.table.setModel(model);
    this.table.setRowSorter(this.sorter);
    setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    setLayout(new BorderLayout());
    add(new JScrollPane(this.table));
    setSize(400, 300);
}
```

```
    setGlassPane (this.glasspane);  
}  
}
```

| Nama | Umur |
|------------------------|------|
| Eko Kurniawan Khannedy | 19 |
| Eko Kurniawan Khannedy | 19 |
| Eko Kurniawan Khannedy | 13 |
| Eko Kurniawan Khannedy | 9 |
| Eko Kurniawan Khannedy | 38 |
| Eko Kurniawan Khannedy | 0 |
| Eko Kurniawan Khannedy | 10 |
| Eko Kurniawan Khannedy | 30 |
| Eko Kurniawan Khannedy | 8 |
| Eko Kurniawan Khannedy | 17 |
| Eko Kurniawan Khannedy | 26 |
| Eko Kurniawan Khannedy | 3 |
| Eko Kurniawan Khannedy | 20 |
| Eko Kurniawan Khannedy | 26 |
| Eko Kurniawan Khannedy | 30 |

Gambar 192 TableTransition.java

| Nama | Umur |
|------------------------|------|
| Eko Kurniawan Khannedy | 09 |
| Eko Kurniawan Khannedy | 09 |
| Eko Kurniawan Khannedy | 03 |
| Eko Kurniawan Khannedy | 0 |
| Eko Kurniawan Khannedy | 08 |
| Eko Kurniawan Khannedy | 0 |
| Eko Kurniawan Khannedy | 10 |
| Eko Kurniawan Khannedy | 30 |
| Eko Kurniawan Khannedy | 8 |
| Eko Kurniawan Khannedy | 27 |
| Eko Kurniawan Khannedy | 26 |
| Eko Kurniawan Khannedy | 3 |
| Eko Kurniawan Khannedy | 20 |
| Eko Kurniawan Khannedy | 36 |
| Eko Kurniawan Khannedy | 30 |

Gambar 193 TableTransition saat memulai transisi

Anda bisa melihat efek transisi ketika JTable diurutkan.

Dragging

Selain transisi, salah satu kegunaan dari Glasspane adalah menampilkan dragging, khususnya gambar atau File. Dan kali ini kita akan menglihat kegunaan Glasspane dalam proses dragging.

“Jika anda tidak memahami Drag and Drop di Swing, anda bisa belajar dari Tutorial Java yang ada dalam CD”

File

Sekarang kita akan membuat glasspane untuk proses drag file.

GlassPaneFileDialog.java

```
package pelajaran14;

import java.awt.Graphics;
import java.awt.Image;
import java.awt.Point;
import javax.swing.ImageIcon;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;

/**
 * @author usu
 */
public class GlassPaneFileDialog extends JPanel {

    private boolean dragFile;
    private final Image image;
    private Point location;

    public GlassPaneFileDialog() {
        super();
        setOpaque(false);
        this.image = new ImageIcon(getClass()
            .getResource("/pelajaran14/file.png")).getImage();
    }

    public boolean isDragFile() {
        return this.dragFile;
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        if (isDragFile()) {
            if ((this.image != null) && (this.location != null)) {
                g.drawImage(this.image, this.location.x, this.location.y,
                           null);
            }
        }
    }

    public void setDragFile(final boolean dragFile, final Point location) {
        this.dragFile = dragFile;
        SwingUtilities.convertPointFromScreen(location, this);
        this.location = location;
        repaint();
    }
}
```

Dan sekarang tinggal kita buat JFrame yang memanfaatkan glasspane diatas.

GlassPaneFileDragApp.java

```
package pelajaran14;

import java.awt.BorderLayout;
import java.awt.Point;
import java.awt.datatransfer.DataFlavor;
import java.awt.datatransfer.Transferable;
import java.awt.datatransfer.UnsupportedFlavorException;
import java.awt.dnd.DnDConstants;
import java.awt.dnd.DropTarget;
import java.awt.dnd.DropTargetDragEvent;
import java.awt.dnd.DropTargetDropEvent;
import java.awt.dnd.DropTargetEvent;
import java.awt.dnd.DropTargetListener;
import java.io.File;
import java.io.IOException;
import java.util.List;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;

/**
 * @author usu
 */
public class GlassPaneFileDragApp extends JFrame {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                new GlassPaneFileDragApp().setVisible(true);
            }
        });
    }

    private final JTextArea area;
    private final GlassPaneFileDrag glasspane;

    public GlassPaneFileDragApp() {
        super();
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
        setSize(400, 300);

        this.glasspane = new GlassPaneFileDrag();
        setGlassPane(this.glasspane);
        getGlassPane().setVisible(true);

        this.area = new JTextArea();
        add(new JScrollPane(this.area));

        new DropTarget(this.area, new DropTargetListener() {

            public void dragEnter(final DropTargetDragEvent dtde) {
                dragOver(dtde);
            }
        });
    }
}
```

```
}

    public void dragExit(final DropTargetEvent dte) {
        final Point p = getMousePosition();
        SwingUtilities.convertPointToScreen(p,
GlassPaneFileDragApp.this);
        GlassPaneFileDragApp.this.glasspane.setDragFile(false, p);
    }

    public void dragOver(final DropTargetDragEvent dtde) {
        if (dtde.isDataFlavorSupported(DataFlavor.javaFileListFlavor))
{
            final Point p = getMousePosition();
            SwingUtilities
                .convertPointToScreen(p, GlassPaneFileDragApp.this);
            GlassPaneFileDragApp.this.glasspane.setDragFile(true, p);
        }
    }

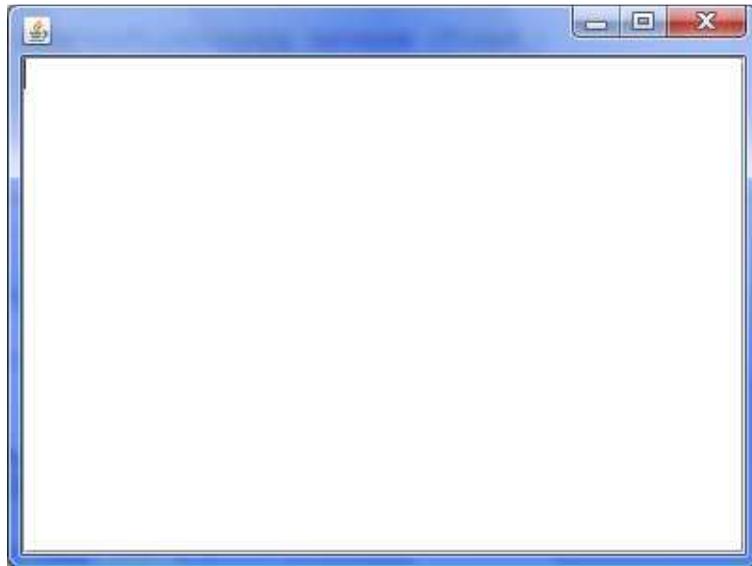
    public void drop(final DropTargetDropEvent dtde) {
        dtde.acceptDrop(DnDConstants.ACTION_COPY_OR_MOVE);
        final Transferable t = dtde.getTransferable();

        for (final DataFlavor flavor : t.getTransferDataFlavors()) {
            if (flavor.isFlavorJavaFileListType()) {
                try {
                    final List<File> list = (List<File>) t
                        .getTransferData(flavor);
                    for (final File f : list) {
                        GlassPaneFileDragApp.this.area.append(f.getPath()
                            + "\n");
                    }
                } catch (final UnsupportedFlavorException ex) {
                    // Error
                } catch (final IOException ex) {
                    // Error
                }
            }
        }

        dtde.dropComplete(true);
        final Point p = getMousePosition();
        SwingUtilities.convertPointToScreen(p,
GlassPaneFileDragApp.this);
        GlassPaneFileDragApp.this.glasspane.setDragFile(false, p);
    }

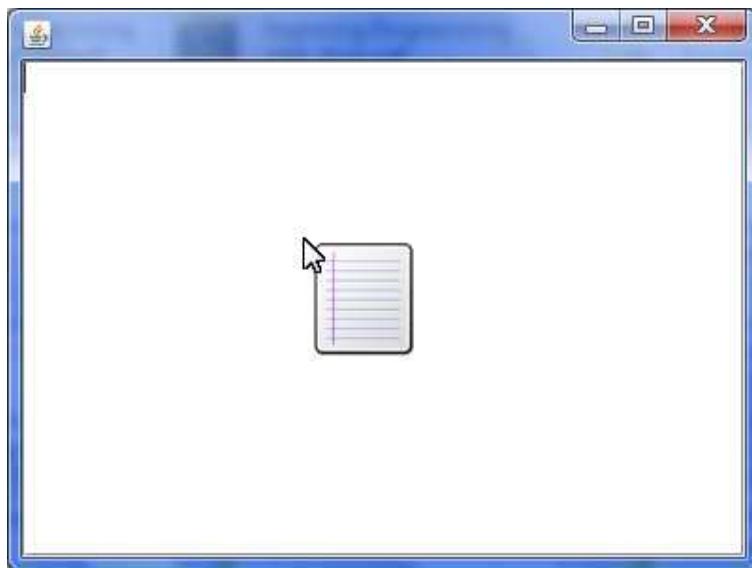
    public void dropActionChanged(final DropTargetDragEvent dtde) {
        dragOver(dtde);
    }
}
}
```

Dan sekarang anda akan melihat tampilan aplikasinya seperti dibawah ini :



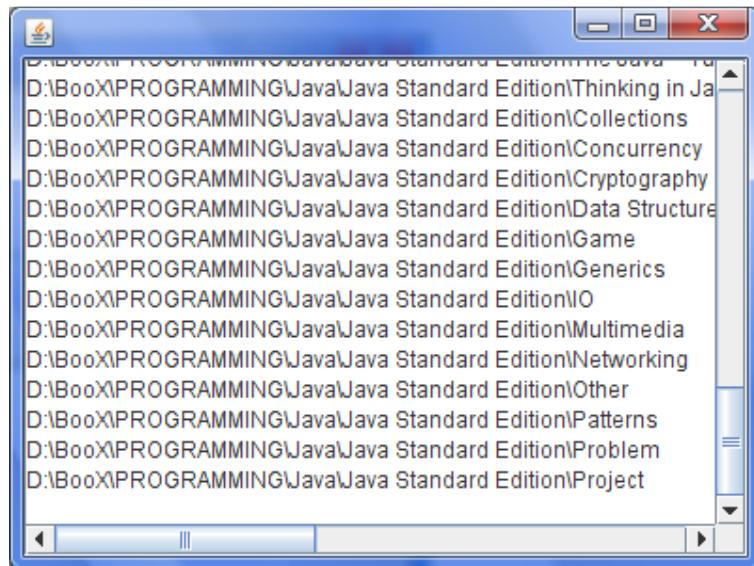
Gambar 194 GlassPaneFileDragApp.java

Jika anda mendrag file dari explorer ke dalam program ini maka akan glasspane akan memperhatikan kalo ada file yang sedang didrag :



Gambar 195 GlassPaneFileDragApp saat glasspane menampilkan gambar saat file didrag

dan jika didrop maka akan terlihat hasilnya :



Gambar 196 GlassPaneFileDragApp saat file drag didrop

Image

Tadi kita sudah membuat glasspane yang berguna menampilkan file yang didrag. Nah sekarang kita akan membuat glasspane yang dapat menampilkan gambar yang sedang didrag. Dan untuk membuatnya kita modifikasi source code GlassPaneFileDrag.java

GlassPanelImageDrag.java

```
package pelajaran14;

import java.awt.Graphics;
import java.awt.Image;
import java.awt.Point;
import java.io.File;
import javax.swing.ImageIcon;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;

/**
 * @author usu
 */
public class GlassPanelImageDrag extends JPanel {

    private boolean dragFile;
    private boolean fileImage;
    private Image gambar;
    private final Image image;
    private Point location;

    public GlassPanelImageDrag() {
        super();
        setOpaque(false);
        this.image = new ImageIcon(getClass()
                .getResource("/pelajaran14/file.png")).getImage();
    }

    public boolean isDragFile() {
```

```
        return this.dragFile;
    }

    public boolean isFileImage() {
        return this.fileImage;
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        if (isDragFile()) {
            if ((this.image != null) && (this.location != null)) {
                if (isFileImage()) {
                    g.drawImage(this.gambar, this.location.x, this.location.y,
100, 100, null);
                } else {
                    g.drawImage(this.image, this.location.x, this.location.y,
null);
                }
            }
        }
    }

    public void setDragFile(final File file, final boolean dragFile,
        final Point location) {

        if (file.getName().toUpperCase().endsWith(".JPG")) {
            this.gambar = new ImageIcon(file.getPath()).getImage();
            setFileImage(true);
        } else {
            setFileImage(false);
        }

        this.dragFile = dragFile;
        SwingUtilities.convertPointFromScreen(location, this);
        this.location = location;
        repaint();
    }

    public void setFileImage(final boolean fileImage) {
        this.fileImage = fileImage;
    }
}
```

Sekarang kita buat JFramenya :

GlassPanelImageDragApp.java

```
package pelajaran14;

import java.awt.BorderLayout;
import java.awt.Point;
import java.awt.datatransfer.DataFlavor;
import java.awt.datatransfer.Transferable;
import java.awt.datatransfer.UnsupportedFlavorException;
import java.awt.dnd.DnDConstants;
import java.awt.dnd.DropTarget;
import java.awt.dnd.DropTargetDragEvent;
```

```
import java.awt.dnd.DropTargetDropEvent;
import java.awt.dnd.DropTargetEvent;
import java.awt.dnd.DropTargetListener;
import java.io.File;
import java.io.IOException;
import java.util.List;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;

/**
 * @author usu
 */
public class GlassPaneImageDragApp extends JFrame {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                new GlassPaneImageDragApp().setVisible(true);
            }
        });
    }

    private final JTextArea area;
    private final GlassPaneImageDrag glasspane;

    public GlassPaneImageDragApp() {
        super();
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
        setSize(400, 300);

        this.glasspane = new GlassPaneImageDrag();
        setGlassPane(this.glasspane);
        getGlassPane().setVisible(true);

        this.area = new JTextArea();
        add(new JScrollPane(this.area));

        new DropTarget(this.area, new DropTargetListener() {

            private File file;
            private boolean in = false;

            public void dragEnter(final DropTargetDragEvent dtde) {
                dragOver(dtde);
            }

            public void dragExit(final DropTargetEvent dte) {
                final Point p = getMousePosition();
                SwingUtilities.convertPointToScreen(p,
                    GlassPaneImageDragApp.this);
                GlassPaneImageDragApp.this.glasspane.setDragFile(new File(""),
                    false, p);
                this.in = false;
            }
        });
    }
}
```

```

public void dragOver(final DropTargetDragEvent dtde) {
    if (dtde.isDataFlavorSupported(DataFlavor.javaFileListFlavor))
    {

        final Point p = getMousePosition();
        SwingUtilities.convertPointToScreen(p,
            GlassPaneImageDragApp.this);
        if (!this.in) {
            dtde.acceptDrag(DnDConstants.ACTION_COPY_OR_MOVE);
            final Transferable t = dtde.getTransferable();
            for (final DataFlavor flavor :
t.getTransferDataFlavors()) {
                if (flavor.isFlavorJavaFileListType()) {
                    try {
                        final List<File> list = (List<File>) t
                            .getTransferData(flavor);
                        if (this.file != null) {
                            if (!this.file.equals(list.get(0))) {
                                this.file = list.get(0);
                            }
                        } else {
                            this.file = list.get(0);
                        }
                    } catch (final UnsupportedFlavorException ex) {
                        // Error
                    } catch (final IOException ex) {
                        // Error
                    }
                }
            }
            this.in = true;
            GlassPaneImageDragApp.this.glasspane.setDragFile(this.file,
                true, p);
        } else {
            dtde.rejectDrag();
        }
    }

    public void drop(final DropTargetDropEvent dtde) {
        dtde.acceptDrop(DnDConstants.ACTION_COPY_OR_MOVE);
        final Transferable t = dtde.getTransferable();

        for (final DataFlavor flavor : t.getTransferDataFlavors()) {
            if (flavor.isFlavorJavaFileListType()) {
                try {
                    final List<File> list = (List<File>) t
                        .getTransferData(flavor);
                    for (final File f : list) {
                        GlassPaneImageDragApp.this.area.append(f.getPath()
                            + "\n");
                    }
                } catch (final UnsupportedFlavorException ex) {
                    // Error
                } catch (final IOException ex) {
                    // Error
                }
            }
        }
        final Point p = getMousePosition();
        SwingUtilities.convertPointToScreen(p,
            GlassPaneImageDragApp.this);
    }
}

```

```
GlassPaneImageDragApp.this.glasspane.setDragFile(new File(""),
    false, p);
this.in = false;
dtde.dropComplete(true);
}

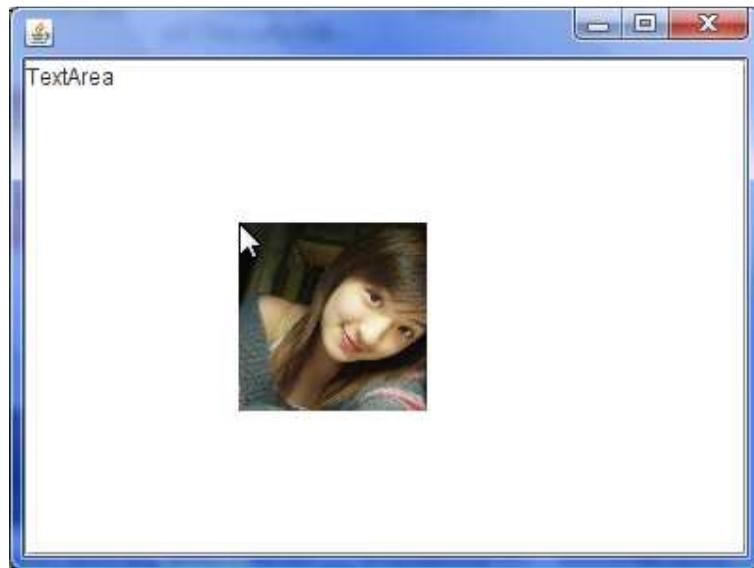
public void dropActionChanged(final DropTargetDragEvent dtde) {
    dragOver(dtde);
}
} );
}
}
```

Tampilan JFrame :



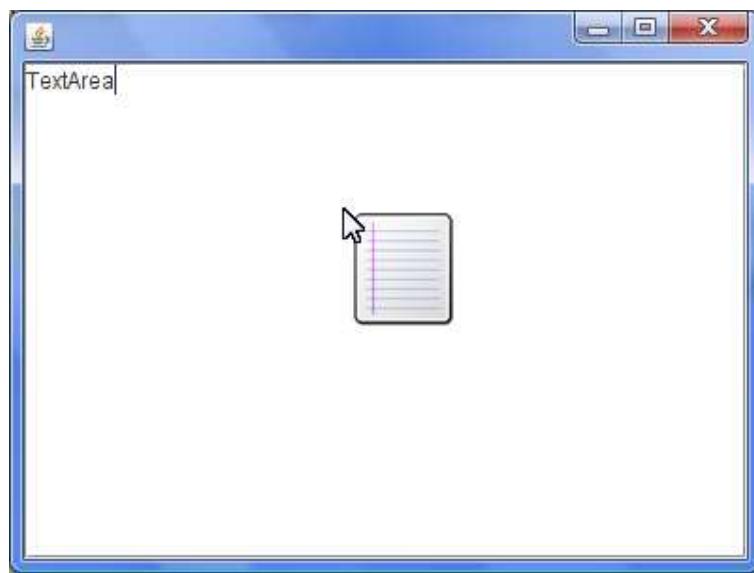
Gambar 197 GlassPanelImageDragApp.java

Saat ada fila gambar (jpg) yang di drag diatas GlassPane :



Gambar 198 GlassPanelImageDragApp saat user mendrag file gambar

Saat ada file yang bukan gambar (jpg) di drag diatas glasspane :



Gambar 199 GlassPanelImageDragApp saat user mendrag file selain gambar

Container

Salah satu kemampuan yang lain dari GlassPane adalah bisa digunakan sebagai Container. Karena setiap membuat glasspane saya selalu menggunakan class yang diturunkan dari JPanel, maka otomatis, sifat JPanel ada dalam GlassPane, misalnya dapat digunakan sebagai container untuk komponen lain.

Show Message

Biasanya anda menggunakan JOptionPane untuk menampilkan pesan, tapi sekarang kita akan membuat tampilan pesan diatas glasspane.

GlassPaneMessage.java

```
package pelajaran14;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.MouseAdapter;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.SwingConstants;
import javax.swing.border.LineBorder;

/**
 * @author usu
 */
public class GlassPaneMessage extends JPanel {

    private final JLabel label;
    private final JPanel panel;
    private boolean showMessage;
    private final JButton tombol;

    public GlassPaneMessage() {
        super();
        setOpaque(false);

        this.panel = new JPanel();
        this.panel.setLayout(new BorderLayout(20, 20));
        this.panel.setPreferredSize(new Dimension(200, 100));
        this.panel.setBorder(new LineBorder(Color.BLACK, 3));

        this.tombol = new JButton("CLOSE");
        this.tombol.addActionListener(new ActionListener() {

            public void actionPerformed(final ActionEvent e) {
                setShowMessage(false);
            }
        });

        final JPanel temp = new JPanel(new FlowLayout());
        temp.setOpaque(false);
        temp.add(this.tombol);
        this.panel.add(temp, BorderLayout.SOUTH);

        this.label = new JLabel();
        this.label.setHorizontalAlignment(SwingConstants.CENTER);
        this.label.setOpaque(false);
        this.panel.add(this.label);

        setLayout(new FlowLayout());
        add(this.panel);

        addMouseListener(new MouseAdapter() {
```

```
    } );
    addKeyListener(new KeyAdapter() {
    } );
}

public boolean isShowMessage() {
    return this.showMessage;
}

@Override
protected void paintComponent(final Graphics g) {
    super.paintComponent(g);

    if (isShowMessage()) {
        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setColor(new Color(1F, 0F, 0F, 0.5F));
        g2.fillRect(0, 0, getWidth(), getHeight());
    }
}

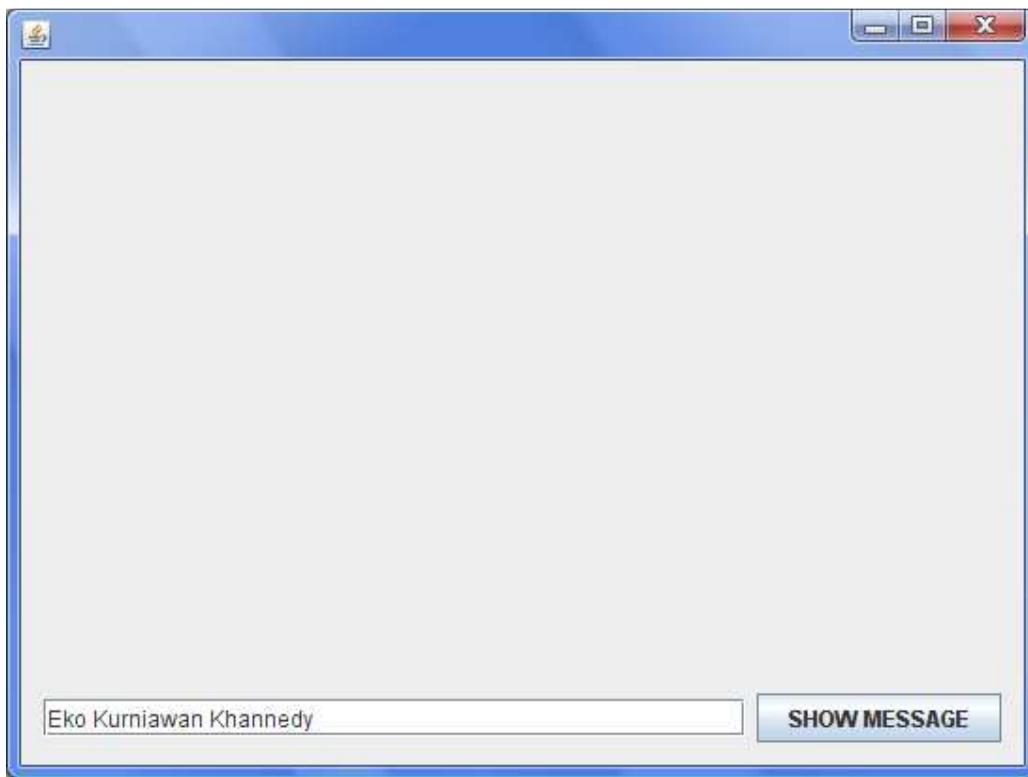
public void setMessage(final String message) {
    this.label.setText(message);
}

public void setShowMessage(final boolean showMessage) {
    this.showMessage = showMessage;
    repaint();
    setVisible(showMessage);
}

@Override
public void setVisible(final boolean aFlag) {
    if (isShowMessage())
        super.setVisible(aFlag);
    else {
        super.setVisible(false);
    }
}
```

Contoh penggunaan GlassPaneMessage :

GlassPaneMessageApp.java



Gambar 200 GlassPaneMessageApp.java

Ketika anda mengklik tombol SHOW MESSAGE, maka glasspane akan menampilkan pesan yang anda tuliskan di TextField :



Gambar 201 GlassPaneMessageApp saat menampilkan pesan

Untuk menggunakan menset tampilan teks di GlassPaneMessage anda gunakan metode :

glasspane.setMessage(String teks);

Dan untuk menampilkan pesan gunakan :

glasspane.showMessage(true);

Dan untuk menghilangkan pesan gunakan :

glasspane.showMessage(false);

GlassDialog

Tadi kita membuat glasspane yang berguna sebagai container namun hanya dapat menampilkan pesan, oleh karena itu kali ini kita akan membuat glasspane yang dapat menampilkan panel yang sesuai dengan yang user inputkan :

GlassPanePanel.java

```
package pelajaran14;  
  
import java.awt.AlphaComposite;  
import java.awt.Color;
```

```
import java.awt.Dimension;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import javax.swing.JPanel;

/**
 * @author usu
 */
public class GlassPanePanel extends JPanel {

    public GlassPanePanel() {
        super();
        setLayout(null);
        setOpaque(false);
    }

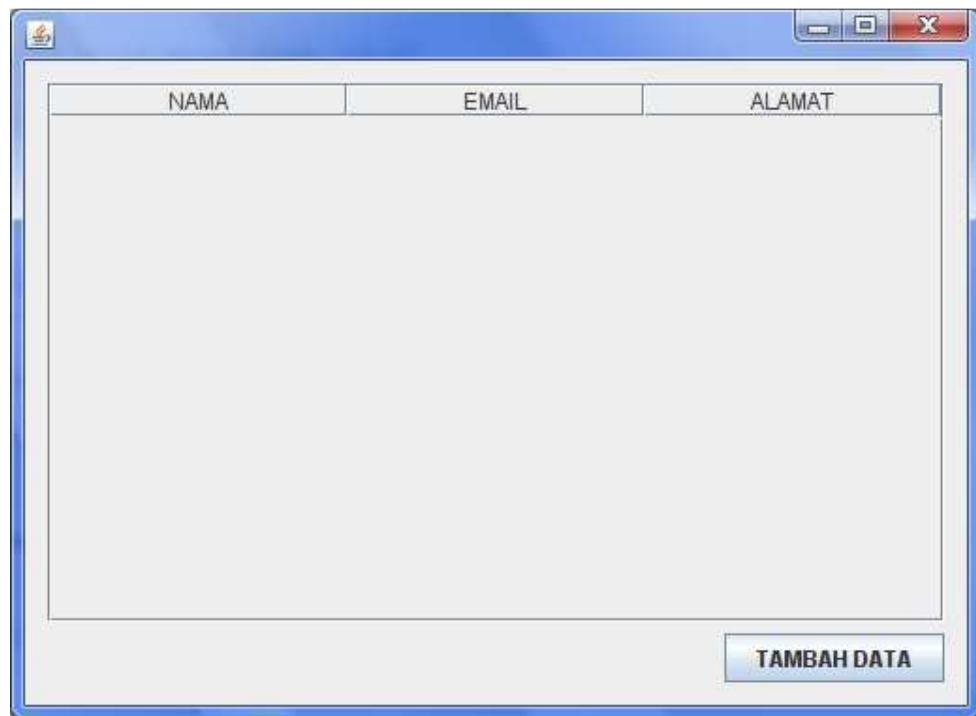
    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final GradientPaint paint = new GradientPaint(0, 0, Color.GREEN,
            getWidth(), getHeight(), Color.YELLOW);

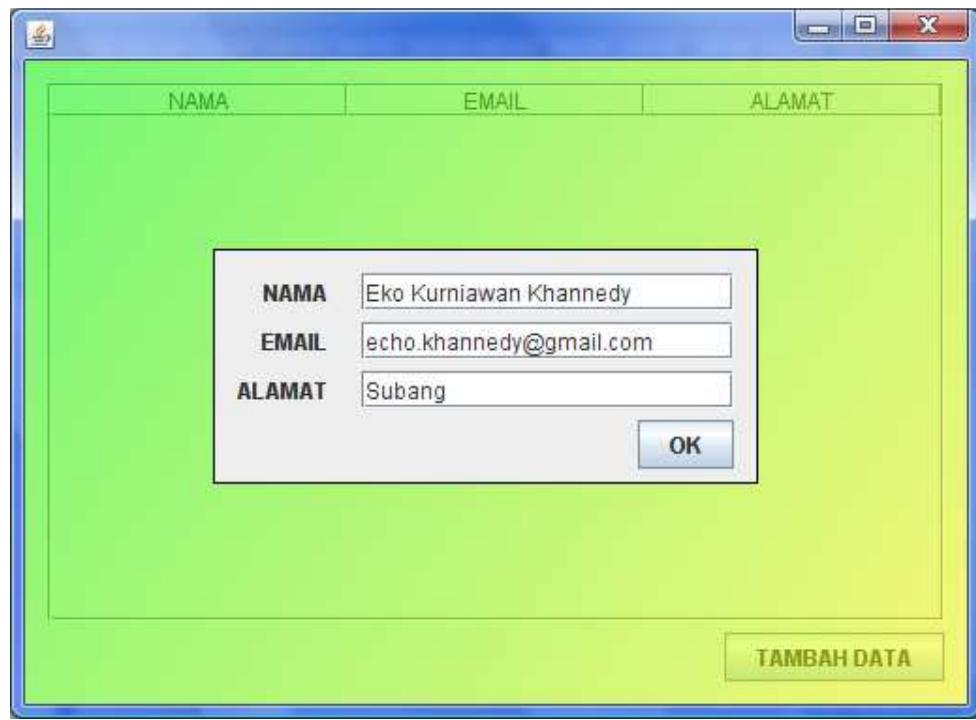
        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setComposite(AlphaComposite.SrcOver.derive(0.5F));
        g2.setPaint(paint);
        g2.fillRect(0, 0, getWidth(), getHeight());
    }

    public void showPanel(final JPanel panel, final Dimension size) {
        removeAll();
        add(panel);
        panel.setBounds(100, 100, size.width, size.height);
    }
}
```

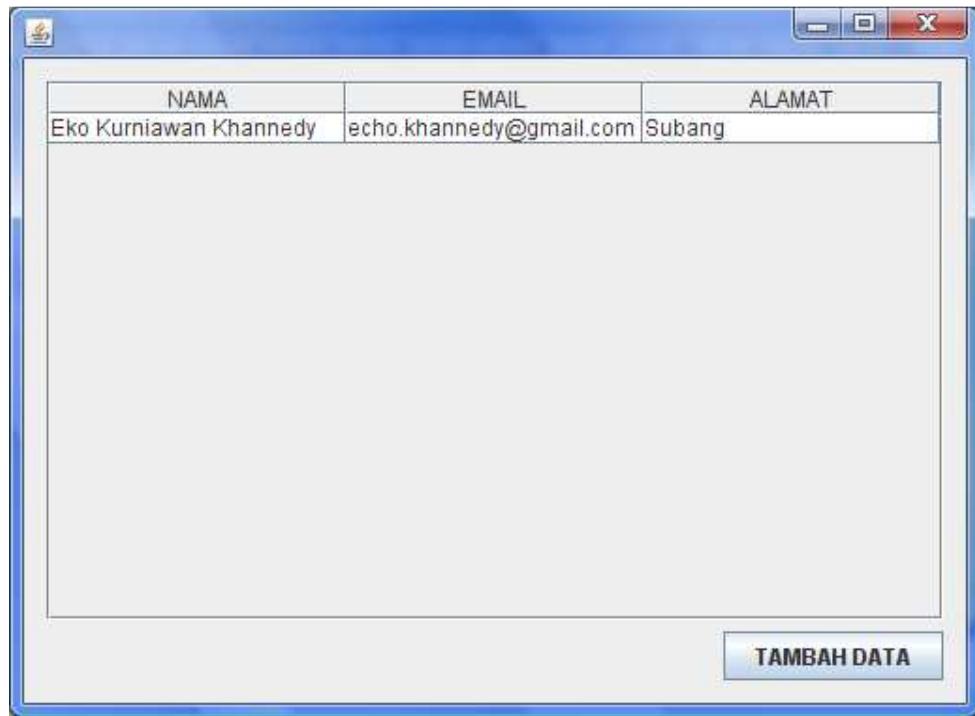
GlassPanePanelApp.java



Gambar 202 GlassPanePanelApp.java



Gambar 203 GlassPanePanelApp saat menampilkan panel



Gambar 204 GlassPanePanelApp.java

ProgressBar

Selain menampilkan pesan, salah satu kemampuan GlassPane sebagai container adalah menampilkan progressbar. Misal dari pada menampilkan progressbar di dialog mungkin menurut saya lebih baik menampilkan progressbar di GlassPane.

```
package pelajaran14;

import java.awt.Color;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.event.KeyAdapter;
import java.awt.event.MouseAdapter;
import javax.swing.JPanel;
import javax.swing.JProgressBar;

/**
 * @author usu
 */
public class GlassPaneProgress extends JPanel {

    private final JProgressBar progress;
    private boolean showProgress;

    public GlassPaneProgress() {
        super();
        this.progress = new JProgressBar();
        this.progress.setPreferredSize(new Dimension(200, this.progress
    }
}
```

```
    .getPreferredSize().height));

setLayout(new FlowLayout());
setOpaque(false);
add(this.progress);

addMouseListener(new MouseAdapter() {
});
addKeyListener(new KeyAdapter() {
});
}

public int getMaximum() {
    return this.progress.getMaximum();
}

public int getMinimum() {
    return this.progress.getMinimum();
}

public boolean isIndeterminate() {
    return this.progress.isIndeterminate();
}

public boolean isShowProgress() {
    return this.showProgress;
}

public boolean isStringPainted() {
    return this.progress.isStringPainted();
}

@Override
protected void paintComponent(final Graphics g) {
    super.paintComponent(g);

    if (isShowProgress()) {
        final Graphics2D g2 = (Graphics2D) g.create();
        g2.setColor(new Color(1F, 0F, 0F, 0.5F));
        g2.fillRect(0, 0, getWidth(), getHeight());
        g2.dispose();
    }
}

public void setIndeterminate(final boolean newValue) {
    this.progress.setIndeterminate(newValue);
}

public void setMaximum(final int n) {
    this.progress.setMaximum(n);
}

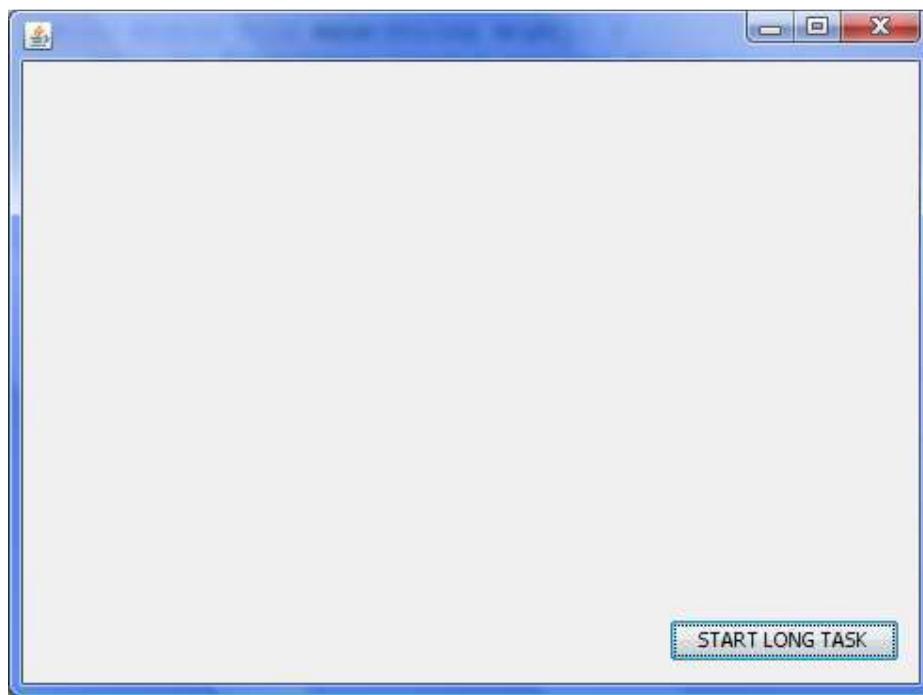
public void setMinimum(final int n) {
    this.progress.setMinimum(n);
}

public void setShowProgress(final boolean showProgress) {
    this.showProgress = showProgress;
    setVisible(showProgress);
}
```

```
public void setString(final String s) {  
    this.progress.setString(s);  
}  
  
public void setStringPainted(final boolean b) {  
    this.progress.setStringPainted(b);  
}  
  
public void setValue(final int n) {  
    this.progress.setValue(n);  
}  
}
```

Contoh penggunaan GlassPaneProgress :

GlassPaneProgressApp.java



Gambar 205 GlassPaneProgressApp.java

Jika anda mengklik tombol START LONG TASK maka glasspane akan memulai menampilkan progressbar :



Gambar 206 GlassPaneProgressApp saat proses progress berjalan

Metode dalam GlassPaneProgress kebanyakan mendelegate metode milik JprogressBar, jadi menggunakan GlassPaneProgress layaknya menggunakan JProgressBar. Untuk menampilkan glasspane gunakan :

`glasspane.setShowProgress(true);`

Dan untuk menghilangkan glasspane gunakan :

`glasspane.setShowProgress(false);`

ProgressBar2

Tadi kita membuat GlassPane ProgressBar menggunakan bantuan JProgressBar, nah sekarang kita buat yang murki kita buat sendiri tanpa bantuan komponen lain.

GlassPaneProgress2.java

```
package pelajaran14;

import java.awt.Color;
import java.awt.Dimension;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.event.KeyAdapter;
import java.awt.event.MouseAdapter;
import java.awt.geom.Rectangle2D;
import javax.swing.JPanel;
```

```
/*
 * @author usu
 */
public class GlassPaneProgress2 extends JPanel {

    private final int max = 100;
    private final int min = 0;
    private int value;

    public GlassPaneProgress2() {
        super();
        setOpaque(false);
        addMouseListener(new MouseAdapter() {
        });
        addKeyListener(new KeyAdapter() {
        });
    }

    public int getMax() {
        return this.max;
    }

    public int getMin() {
        return this.min;
    }

    public int getValue() {
        return this.value;
    }

    @Override
    protected void paintComponent(final Graphics g) {
        super.paintComponent(g);

        final Dimension dim = getSize();
        final int x = (dim.width - getMax()) / 2;
        final int y = (dim.height - 10) / 2;

        Rectangle2D.Double kotak = new Rectangle2D.Double(x, y, getValue(), 10);
        final GradientPaint paint = new GradientPaint(x, y, Color.BLACK, x, y + 10, Color.RED);

        final Graphics2D g2 = (Graphics2D) g.create();

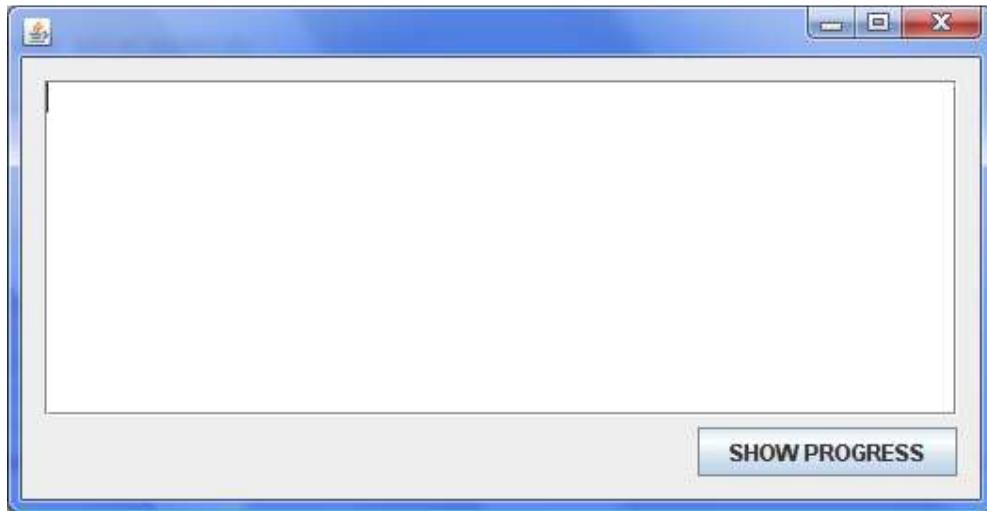
        g2.setColor(new Color(1F, 0F, 0F, 0.5F));
        g2.fillRect(0, 0, getWidth(), getHeight());
        g2.setPaint(paint);
        g2.fill(kotak);

        kotak = new Rectangle2D.Double(x, y, getMax(), 10);
        g2.setColor(Color.black);
        g2.draw(kotak);
        g2.dispose();
    }

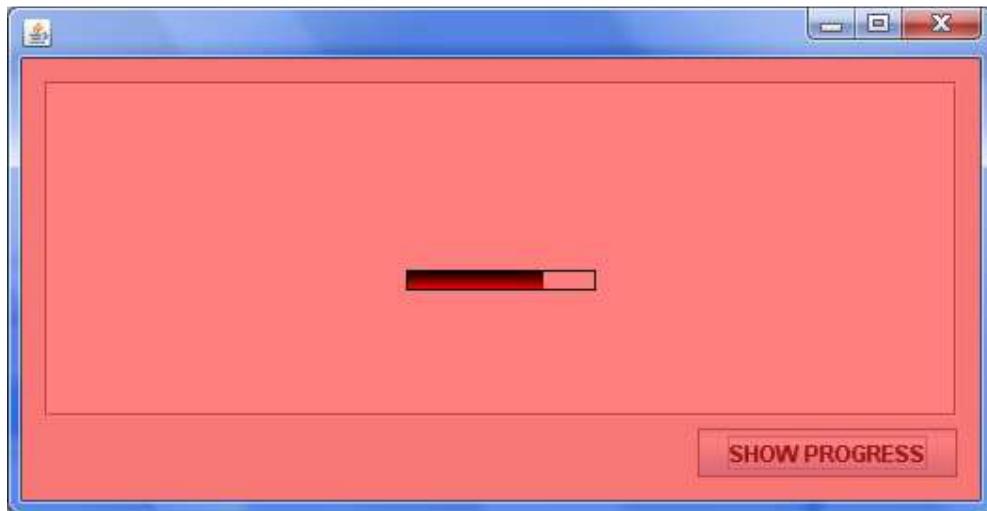
    public void setValue(final int value) throws IllegalArgumentException {
        if (value < 0) {
            throw new IllegalArgumentException();
        }
    }
}
```

```
    this.value = value;
    repaint();
}
```

GlassPaneProgress2App.java



Gambar 207 GlassPaneProgress2App.java



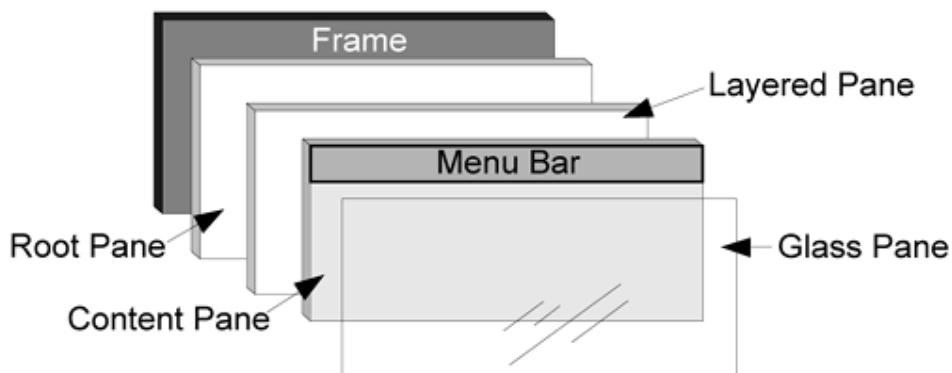
Gambar 208 GlassPaneProgress2App saat progress berjalan

Kesimpulan

Wah ini dia bagian yang paling saya benci, “MENYIMPULKAN”. Yang pasti GlassPane adalah keajaiban yang dimiliki oleh Swing yang belum tentu dimiliki oleh bahasa pemrograman visual lain.

Pelajaran 15 JLayeredPane

JLayeredPane merupakan salah satu bagian JFrame juga seperti GlassPane, namun letaknya berada di bawah GlassPane dan ContentPane. Jika anda pernah menggunakan Adobe Photoshop, mungkin anda mengerti apa itu Layer, nah seperti itulah kegunaan layer, menampilkan komponen secara bertumpuk.



Gambar 209 Posisi JLayeredPane dalam JFrame

Tapi saya sarankan untuk tidak memanipulasi JLayeredPane milik JFrame karena urutannya sudah diatur sedemikian rupa agar tampilan JFrame berjalan dengan baik. Anda perlu ketahui jika anda membuat JLayeredPane, maka anda harus menentukan lokasi dan ukuran komponen yang ditempatkan dalam JLayeredPane secara manual, karena JLayeredPane memiliki layout null.

Berbeda dengan Container yang lain untuk menambahkan komponen ke JLayeredPane gunakan :

JLayeredPane.add(Component komp, Integer index);

Semakin tinggi index maka Component akan berada di posisi lebih atas.

SampleLayer.java

```
package pelajaran15;

import java.awt.BorderLayout;
import java.awt.Color;
import javax.swing.JFrame;
import javax.swing.JLayeredPane;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;

/**
 * @author usu
 */
public class SampleLayer extends JFrame {
```

```
public static void main(final String[] usu) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new SampleLayer().setVisible(true);
        }
    });
}

private final JLayeredPane pane;
private final JPanel panel;

private final JPanel panel2;

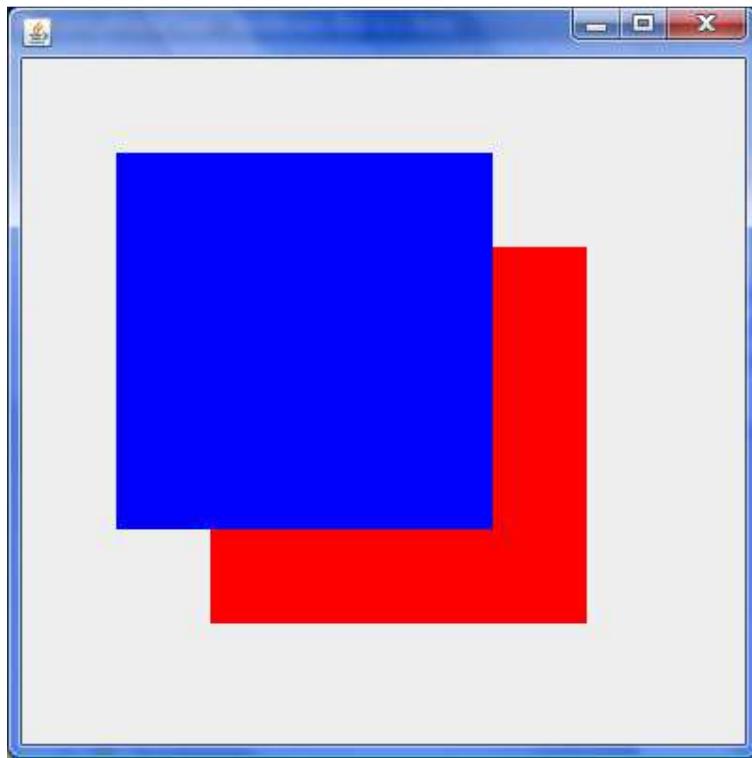
public SampleLayer() {
    super();
    setSize(400, 400);
    setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);

    this.pane = new JLayeredPane();
    this.panel = new JPanel();
    this.panel2 = new JPanel();

    this.pane.add(this.panel, new Integer(1));
    this.pane.add(this.panel2, new Integer(2));

    this.panel.setBounds(100, 100, 200, 200);
    this.panel.setBackground(Color.RED);
    this.panel2.setBounds(50, 50, 200, 200);
    this.panel2.setBackground(Color.BLUE);

    setLayout(new BorderLayout());
    add(this.pane);
}
```



Gambar 210 SampleLayer.java

Salah satu kemampuan JLayeredPane adalah dapat merubah lokasi index komponent yang berada diatasnya :

```
// merubah lokasi index
JLayeredPane.setLayer(Component comp, int index);

// menempatkan komponen ke urutan paling atas
JLayeredPane.moveToFront(Component comp);

// menempatkan komponen ke urutan paling bawah
JLayeredPane.moveToBack(Component comp);
```

LayerUpDown.java

```
package pelajaran15;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

import javax.swing.JFrame;
import javax.swing.JLayeredPane;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;

/**
```

```
* @author usu
*/
public class LayerUpDown extends JFrame {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new
            Runnable() {
                public void run() {
                    new LayerUpDown().setVisible(true);
                }
            });
    }

    private final JLayeredPane pane;
    private final JPanel panel;
    private final JPanel panel2;

    public LayerUpDown() {
        super();
        setSize(400, 400);
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);

        this.pane = new JLayeredPane();
        this.panel = new JPanel();
        this.panel2 = new JPanel();

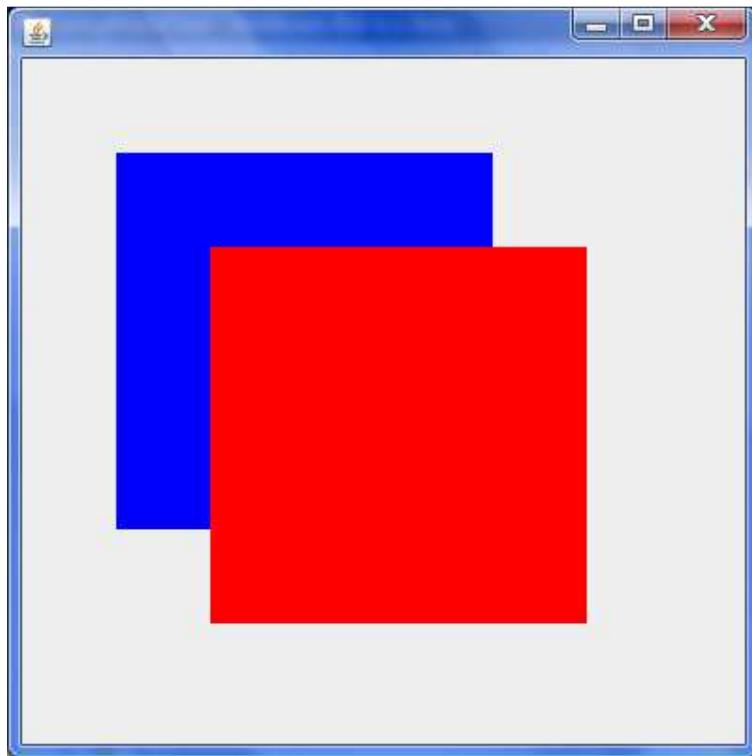
        this.pane.add(this.panel);
        this.pane.add(this.panel2);

        this.panel.setBounds(100, 100, 200, 200);
        this.panel.setBackground(Color.RED);
        this.panel2.setBounds(50, 50, 200, 200);
        this.panel2.setBackground(Color.BLUE);

        this.panel.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                LayerUpDown.this.pane.moveToFront(LayerUpDown.this.panel);
            }
        });

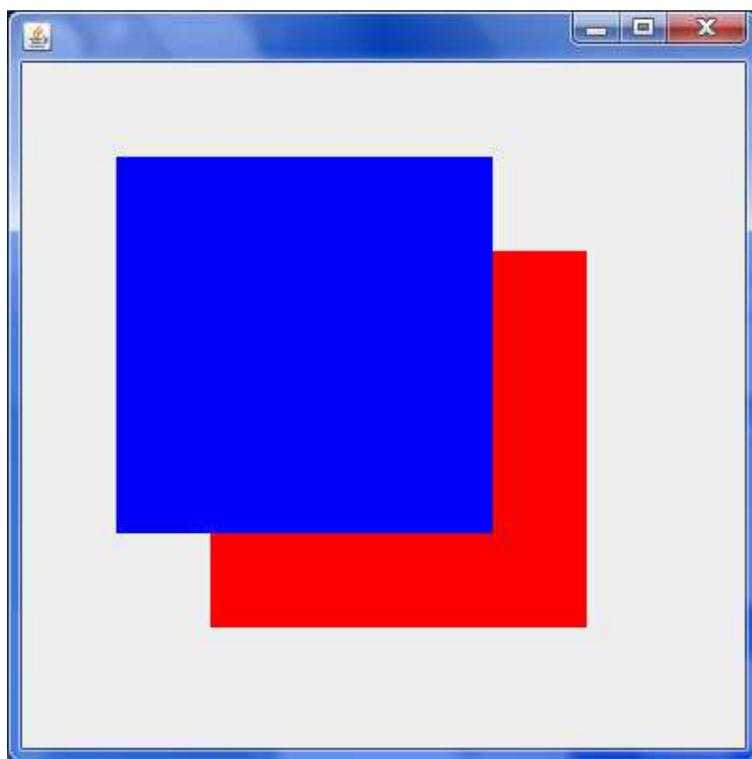
        this.panel2.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                LayerUpDown.this.pane.moveToFront(LayerUpDown.this.panel2);
            }
        });

        setLayout(new BorderLayout());
        add(this.pane);
    }
}
```



Gambar 211 LayerUpDown saat panel merah diatas

Ketika panel biru diklik, maka panel biru akan pindah layer ke urutan paling atas.



Gambar 212 LayerUpDown saat panel biru diatas

GlassPane

Jika kita menggunakan GlassPane dan kita ingin membuat beberapa efek, misal GlassPane Transisi, GlassPane Drag, dan GlassPane Progress. Itu menjadi kendala, karena susah untuk membuat beberapa GlassPane dalam JFrame, nah oleh karena itu dengan adanya JLayeredPane, hal itu bisa diatasi.

Misal kita buat GlassPane Progress dan GlassPane Message dalam satu JFrame.

LayerGlassPane.java

```
package pelajaran15;

import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ComponentAdapter;
import java.awt.event.ComponentEvent;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLayeredPane;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.WindowConstants;
import pelajaran14.GlassPaneMessage;
import pelajaran14.GlassPaneProgress;

/**
 * @author usu
 */
public class LayerGlassPane extends JFrame {

    public static void main(final String[] usu) {
        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                new LayerGlassPane().setVisible(true);
            }
        });
    }

    private final GlassPaneMessage glasspaneMessage;
    private final GlassPaneProgress glasspaneProgress;
    private final JLayeredPane layer;
    private final JPanel panelUtama;
    private final JButton tombolMessage;

    private final JButton tombolProgress;

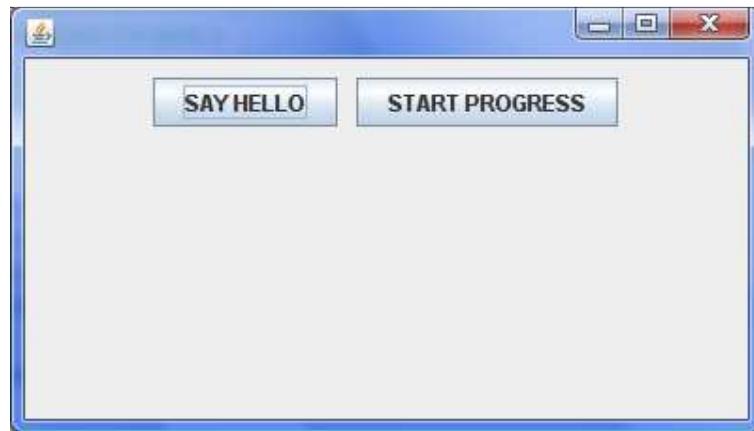
    public LayerGlassPane() {
        super();

        this.layer = new JLayeredPane();
        this.glasspaneMessage = new GlassPaneMessage();
        this.glasspaneProgress = new GlassPaneProgress();
        this.tombolMessage = new JButton("SAY HELLO");
        this.tombolProgress = new JButton("START PROGRESS");
        this.panelUtama = new JPanel(new FlowLayout(FlowLayout.CENTER, 10,
    
```

```
10));  
  
        this.panelUtama.add(this.tombolMessage);  
        this.panelUtama.add(this.tombolProgress);  
  
        this.layer.add(this.glasspaneMessage, JLayeredPane.DRAG_LAYER);  
        this.layer.add(this.glasspaneProgress,JLayeredPane.DRAG_LAYER);  
        this.layer.add(this.panelUtama);  
  
        this.glasspaneMessage.setVisible(false);  
        this.glasspaneProgress.setVisible(false);  
  
        setLayout(new BorderLayout());  
        add(this.layer);  
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);  
        setSize(400, 400);  
        setResizable(false);  
  
        addComponentListener(new ComponentAdapter() {  
  
            @Override  
            public void componentResized(ComponentEvent e) {  
                LayerGlassPane.this.glasspaneMessage.setBounds(0, 0,  
                    LayerGlassPane.this.layer.getWidth(),  
                    LayerGlassPane.this.layer.getHeight());  
                LayerGlassPane.this.glasspaneProgress.setBounds(0, 0,  
                    LayerGlassPane.this.layer.getWidth(),  
                    LayerGlassPane.this.layer.getHeight());  
                LayerGlassPane.this.panelUtama.setBounds(0, 0,  
                    LayerGlassPane.this.layer.getWidth(),  
                    LayerGlassPane.this.layer.getHeight());  
            }  
        } );  
  
        this.tombolMessage.addActionListener(new ActionListener() {  
  
            public void actionPerformed(ActionEvent e) {  
                LayerGlassPane.this.glasspaneMessage.setMessage("HELLO");  
                LayerGlassPane.this.glasspaneMessage.setShowMessage(true);  
            }  
        } );  
  
        this.tombolProgress.addActionListener(new ActionListener() {  
  
            public void actionPerformed(ActionEvent e) {  
                new Thread(new Runnable() {  
  
                    public void run() {  
                        LayerGlassPane.this.glasspaneProgress.setMinimum(0);  
                        LayerGlassPane.this.glasspaneProgress.setMaximum(5000);  
  
                        LayerGlassPane.this.glasspaneProgress.setStringPainted(true);  
  
                        LayerGlassPane.this.glasspaneProgress.setShowProgress(true);  
                        for (int i = 1; i <= 5000; i++) {  
                            try {  
                                LayerGlassPane.this.glasspaneProgress.setValue(i);  
  
                                LayerGlassPane.this.glasspaneProgress.setString("DONE "  
                                    + i + " OF 5000 TASK");  
                                LayerGlassPane.this.glasspaneProgress.repaint();  
                            }  
                        }  
                    }  
                } );  
            }  
        } );  
    }  
}
```

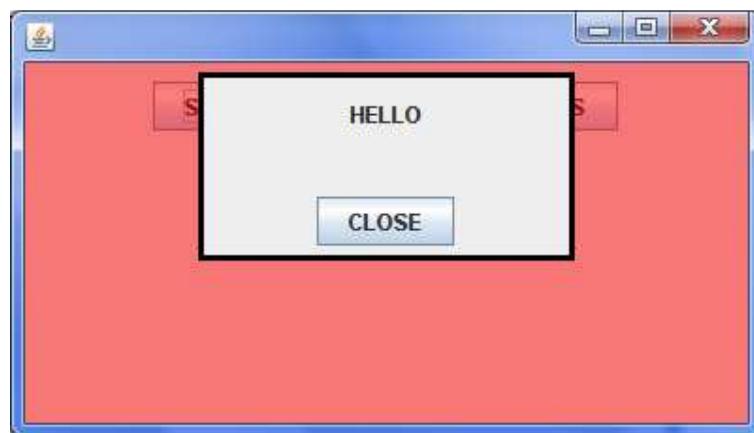
```
        Thread.sleep(1);
    } catch (final InterruptedException ex) {
        // ERROR
    }
}

LayerGlassPane.this.glasspaneProgress.setShowProgress(false);
}
).start();
}
);
}
}
```



Gambar 213 LayerGlassPane.java

Ketika anda tekan tombol SAY HELLO :



Gambar 214 LayerGlassPane saat menampilkan pesan

Ketika anda tekan tombol START PROGRESS :



Gambar 215 LayerGlassPane saat menampilkan progress

Kesimpulan

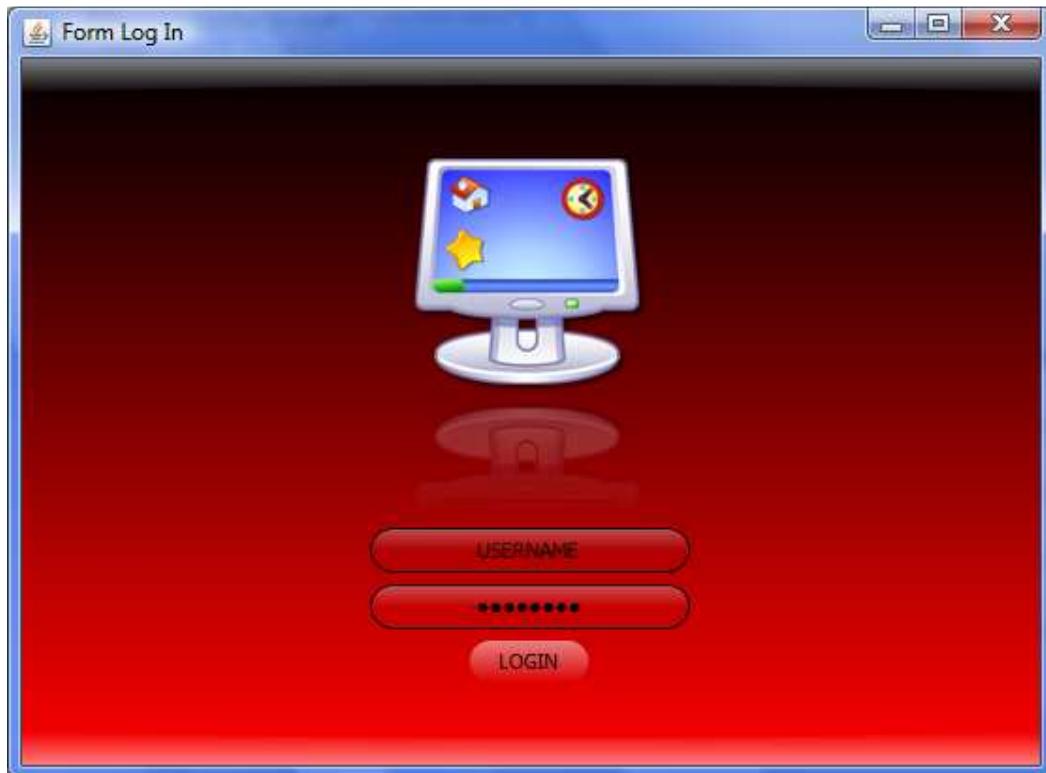
Mungkin yang saya tonjolkan dari JLayeredPane adalah GlassPane. Maksudnya jika anda akan menggunakan banyak GlassPane, lebih baik gunakan JLayeredPane. Dan jangan lupa untuk GlassPane gunakan :

JLayeredPane.add(Component glasspane, JLayeredPane.DRAG_LAYER);

Sample Swing Make Over

Semua contoh – contoh pada gambar dibawah ini dapat anda lihat source codenya di project SwingMakeOver dalam paket “sample” :

FormLogIn.java



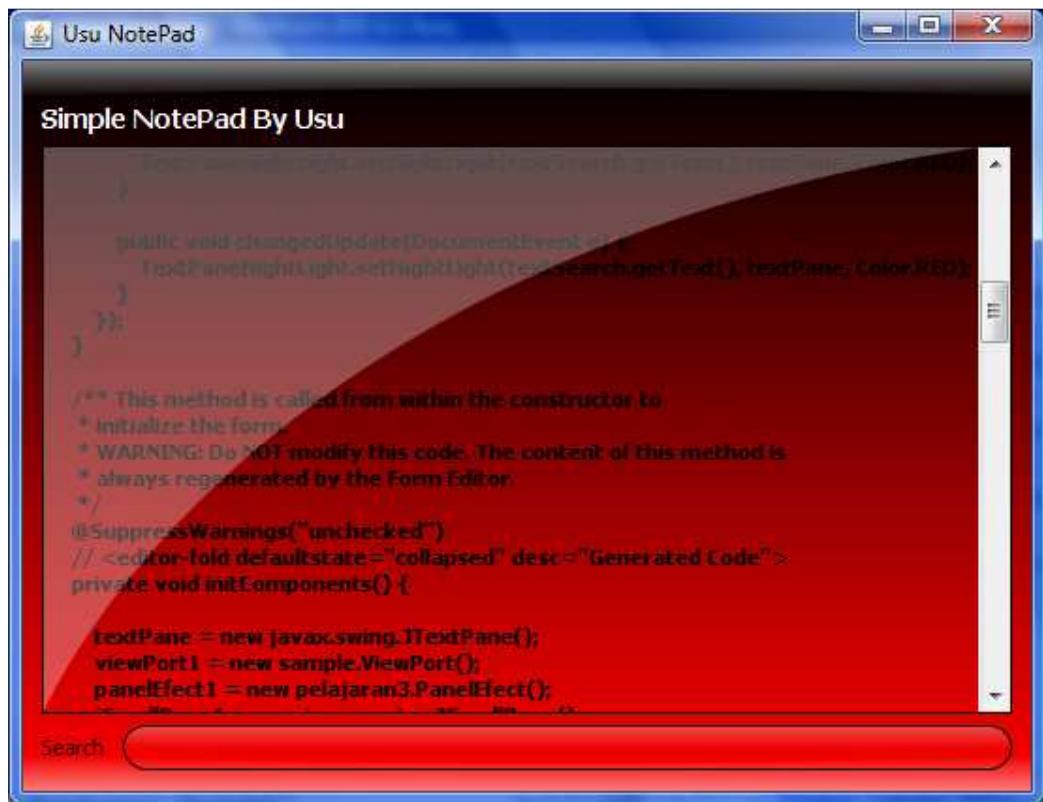
Gambar 216 FormLogIn.java

FormIdentitas.java



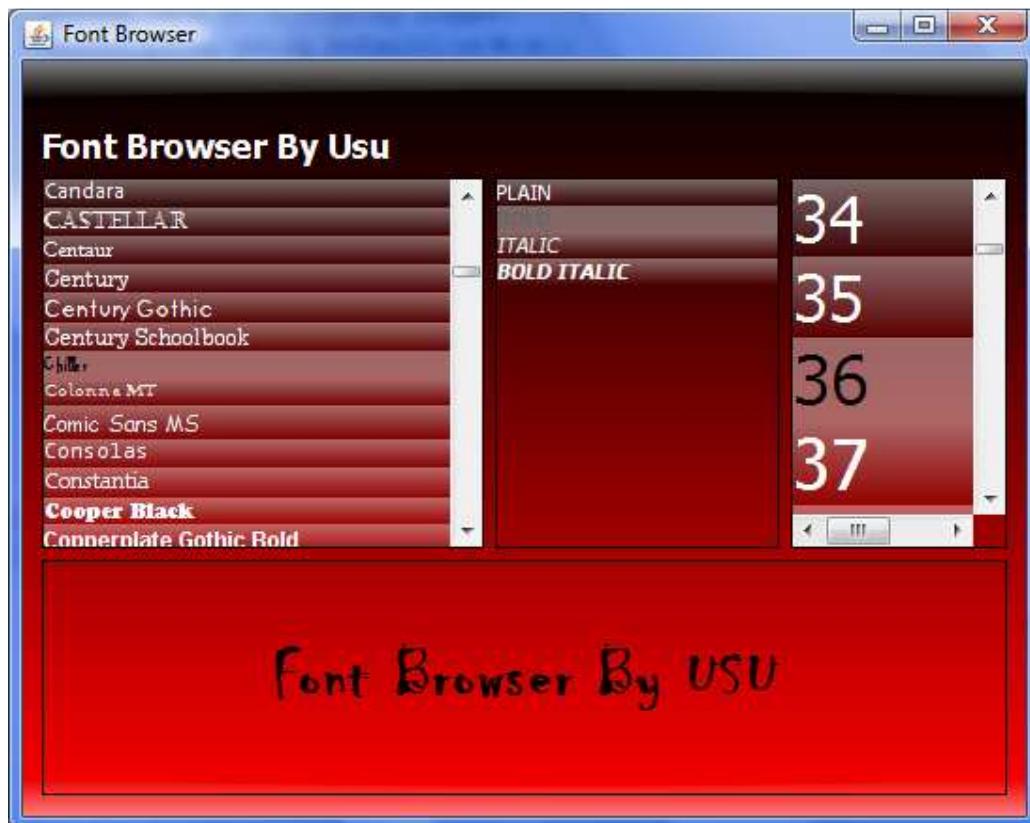
Gambar 217 FormIdentitas.java

NotePad.java



Gambar 218 NotePad.java

FontBrowser.java



Gambar 219 FontBrowser.java

TableMahasiswa.java



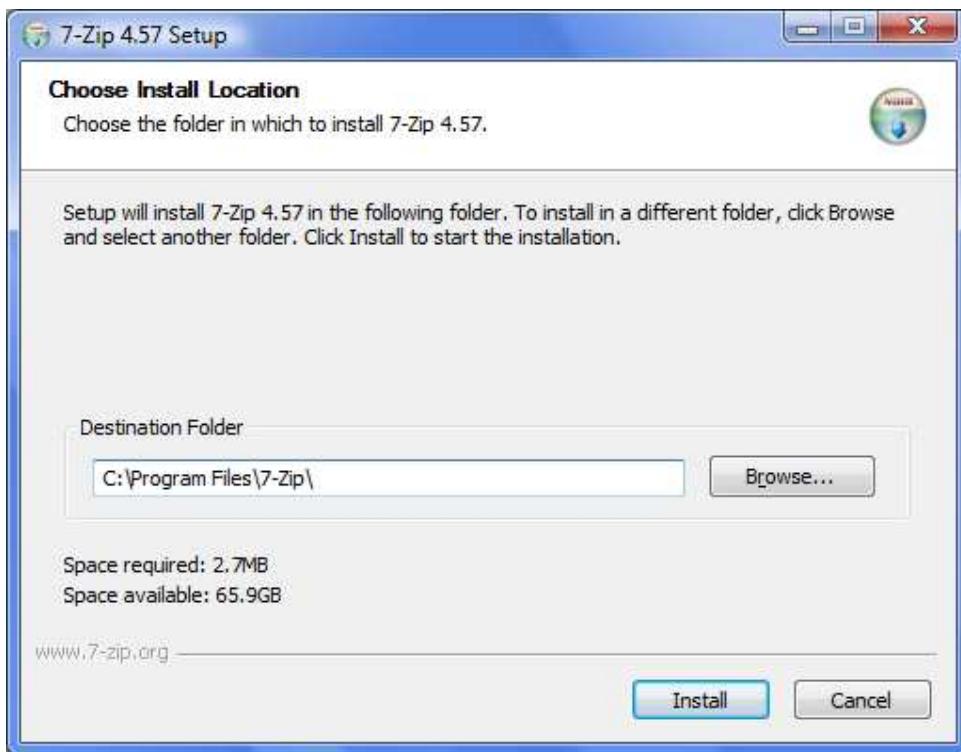
Gambar 220 TableMahasiswa.java

Lampiran

Instalasi

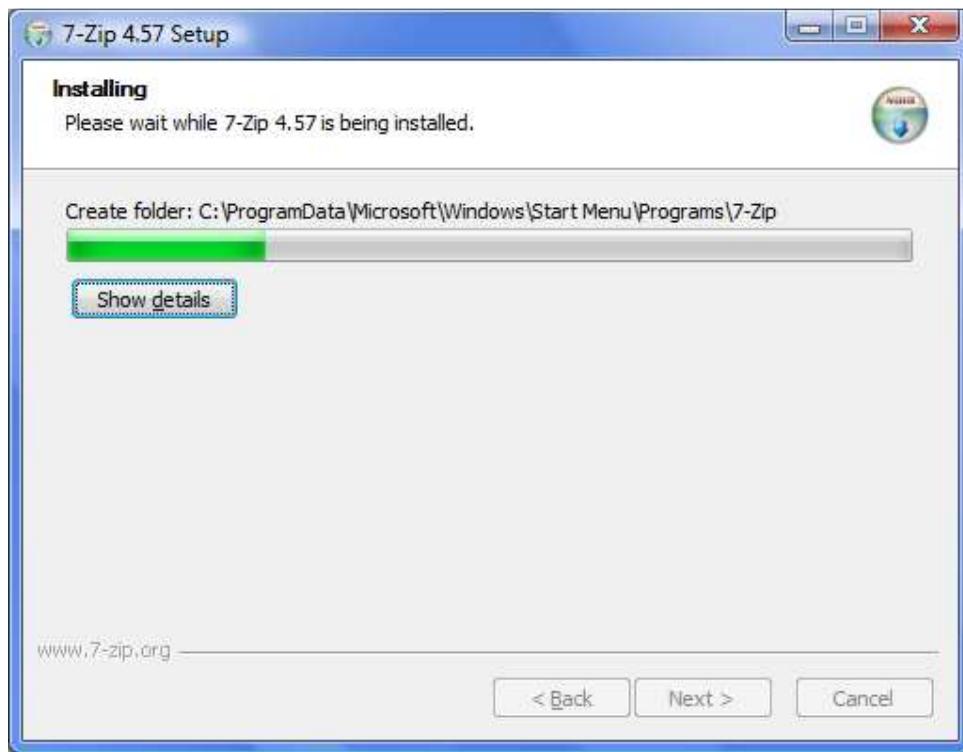
7-Zip

Aplikasi ini akan kita gunakan untuk mengekstrak seluruh file Zip yang ada dalam CD. Kenapa saya anjurkan menggunakan 7-Zip dibandingkan aplikasi lainnya? Hal pertama karena sering sekali terjadi error ketika mengekstrak file Zip jika anda menggunakan Window Zip dan WinRAR. Alasannya karena mereka tak mendukung format nama file yang melebihi dari 255 karakter. Sehingga jika anda akan mengekstrak file yang memiliki nama file yang lebih dari 255 karakter, maka akan terjadi error.



Gambar 221 Choose Install Location

Gmbar diatas adalah tampilan awal proses instalasi 7-Zip. Anda bisa menentukan direktori untuk instalasi 7-Zip dengan mengklik tombol Browse. Jika selesai menentukan lokasi, klik Install.



Gambar 222 Installing

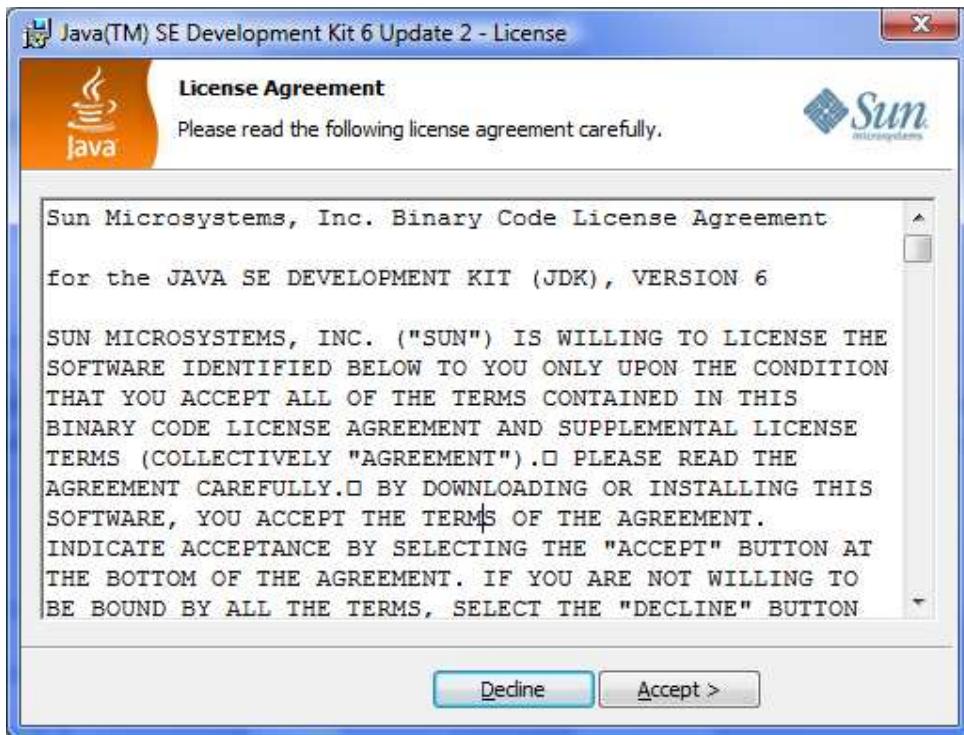
Tunggu sampai proses instalasi selesai. Dan setelah selesai anda akan melihat tampilan seperti gambar dibawah.



Gambar 223 Complete Setup

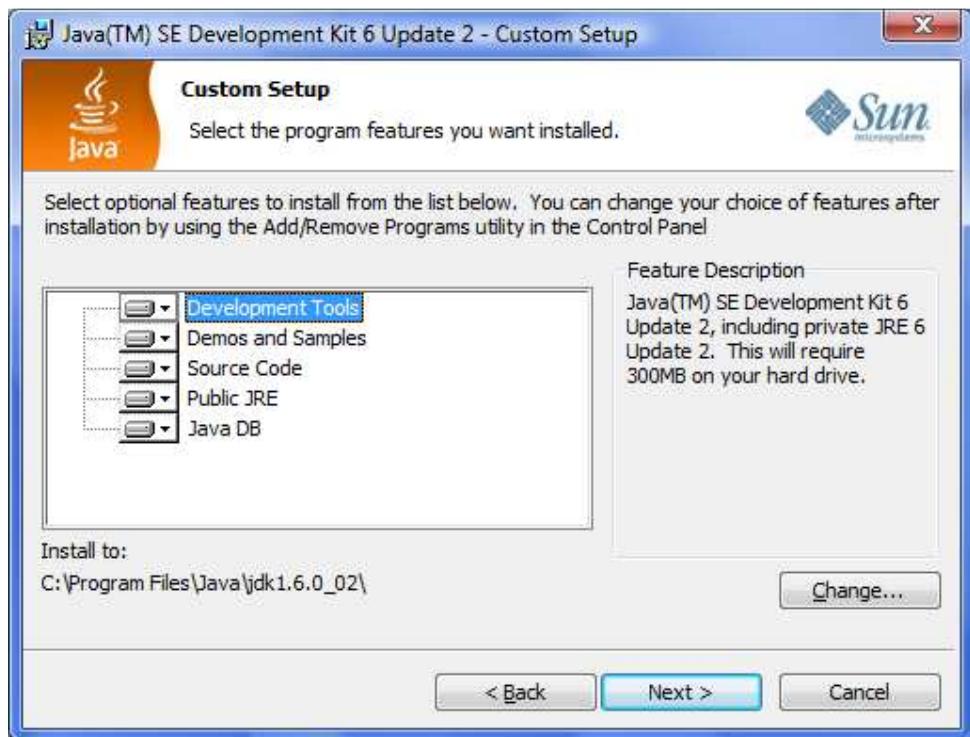
JDK 1.6

Sebelum anda membuat program berbasis java, pastilah harus terinstal JDK dalam komputer anda, dan oleh karena itu sebelum menginstal NetBeans dan Eclipse kita akan menginstal JDK terlebih dahulu.



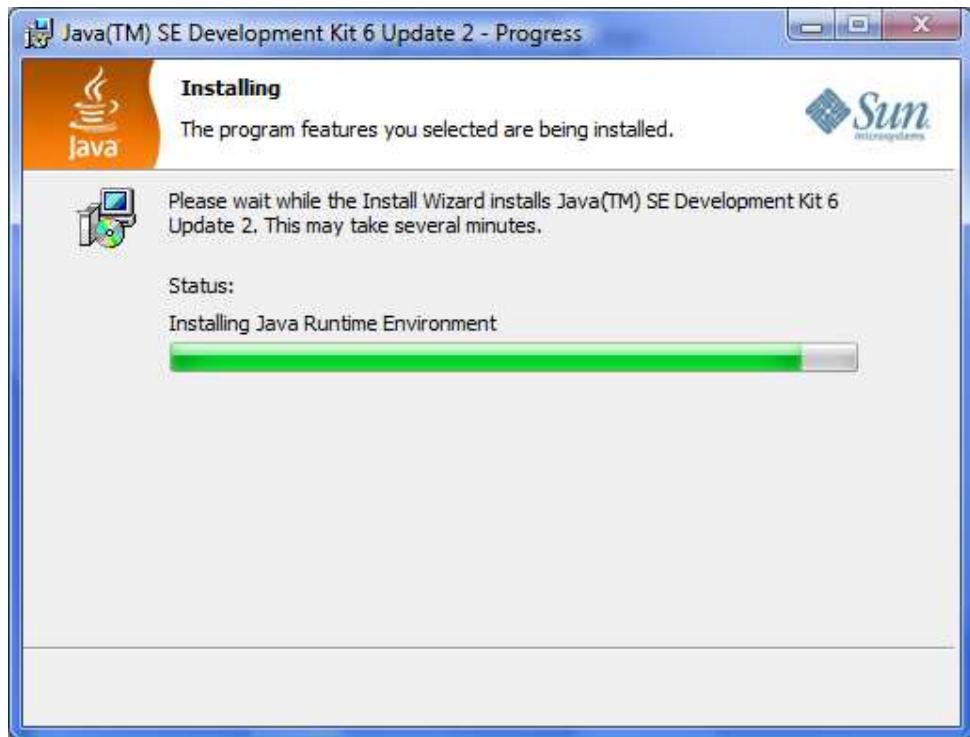
Gambar 224 License Agreement

Gambar diatas adalah tampilan awal proses instalasi JDK, klik Accept untuk melanjutkan proses instalasi.



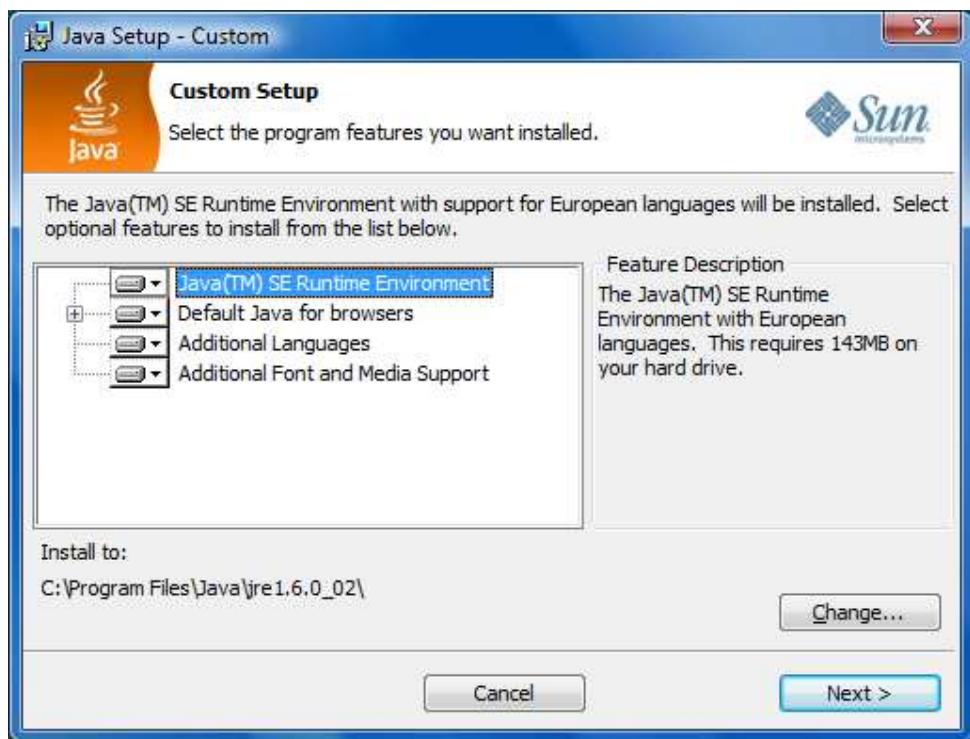
Gambar 225 Custom Setup

Tentukan lokasi instalasi JDK sesuai dengan yang anda inginkan.



Gambar 226 Installing

Gambar diatas menampilkan proses instalasi yang sedang berjalan dan setelah selesai menginstal JDK maka otomatis akan terinstal pula JRE, sehingga muncul kotak dialog seperti dibawah.



Gambar 227 Custom Setup

Tentukan lokasi JRE sesuai dengan yang anda inginkan lalu klik Next.



Gambar 228 Installing Java

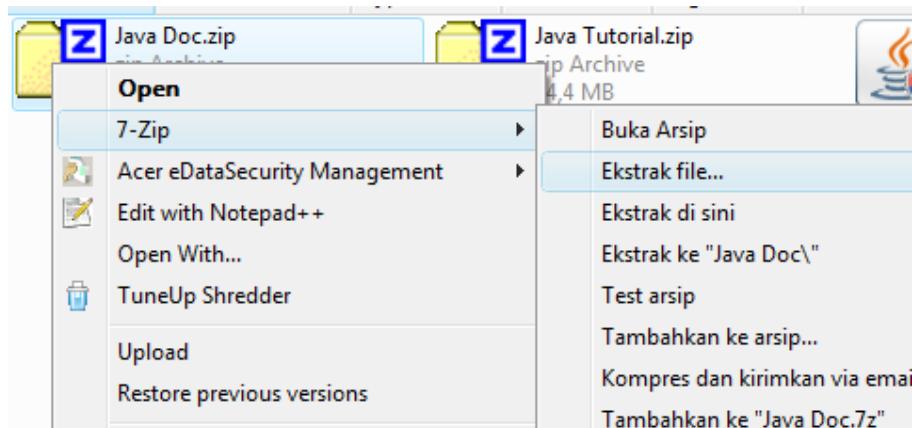
Tunggu sampai JRE selesai terinstal. Setelah selesai maka akan terlihat dialog seperti dibawah ini.



Gambar 229 Wizard Completed

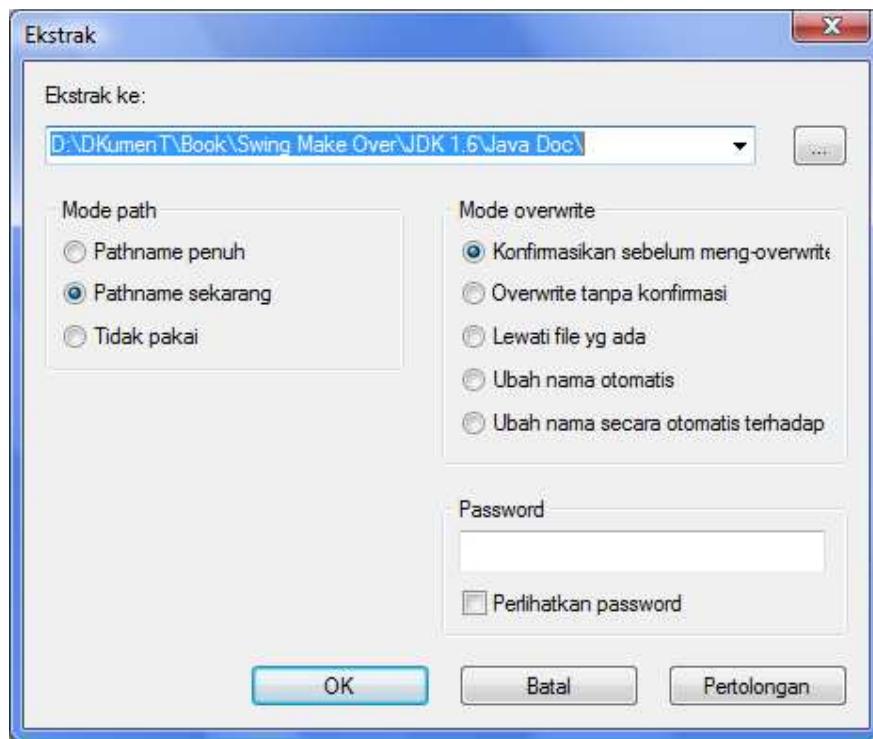
Java Doc

JavaDoc merupakan kumpulan dokumentasi java, baik itu penggunaan java sampai penggunaan API java. Dan untuk menginstalnya anda cukup mengekstraknya ke direktori tempat terinstalnya JDK.



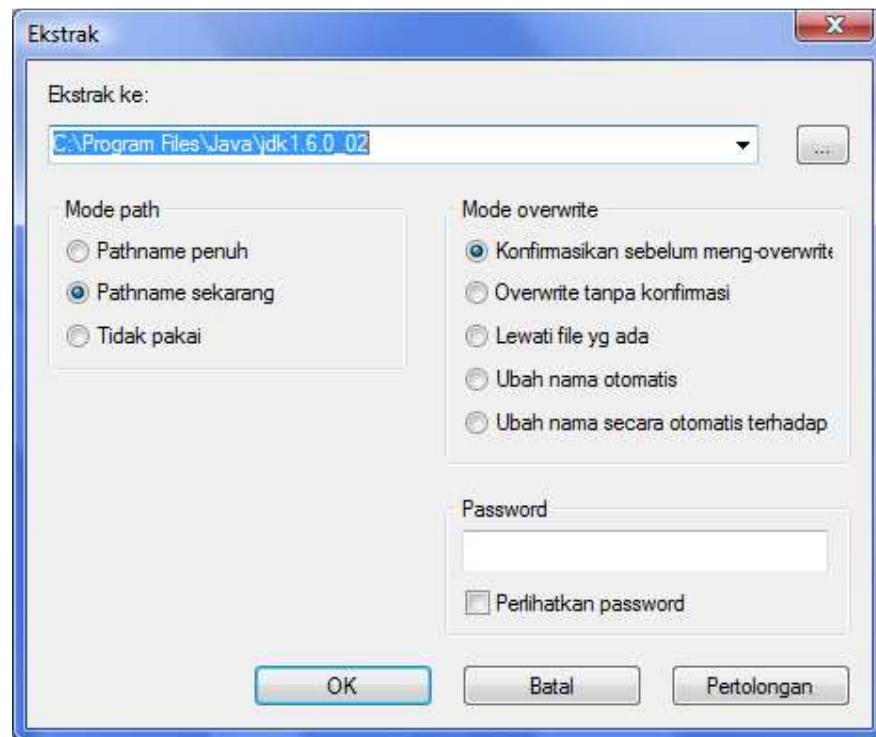
Gambar 230 Ekstrak JavaDoc

Ekstrak file Java Docs.Zip menggunakan 7-Zip dengan mengklik kanan file Java Docs.Zip lalu pulih 7-Zip > Ekstrak File...



Gambar 231 Extrak File

Ubah lokasi target direktori ekstrak ke direktori tempat terinstalnya JDK.



Gambar 232 Ekstrak File

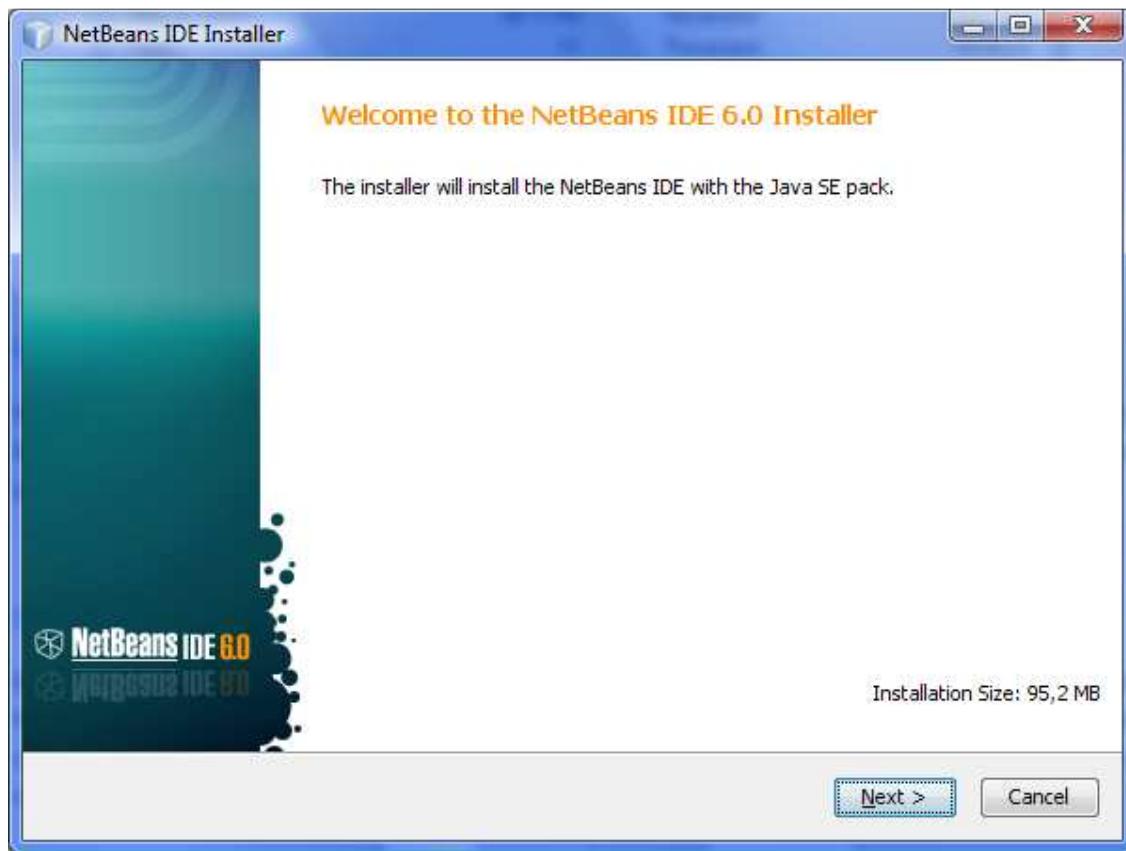
Lalu klik OK untuk mengekstrak file. Dan tunggu sampai 7-Zip mengekstrak semua java doc.



Gambar 233 Proses Ekstrak

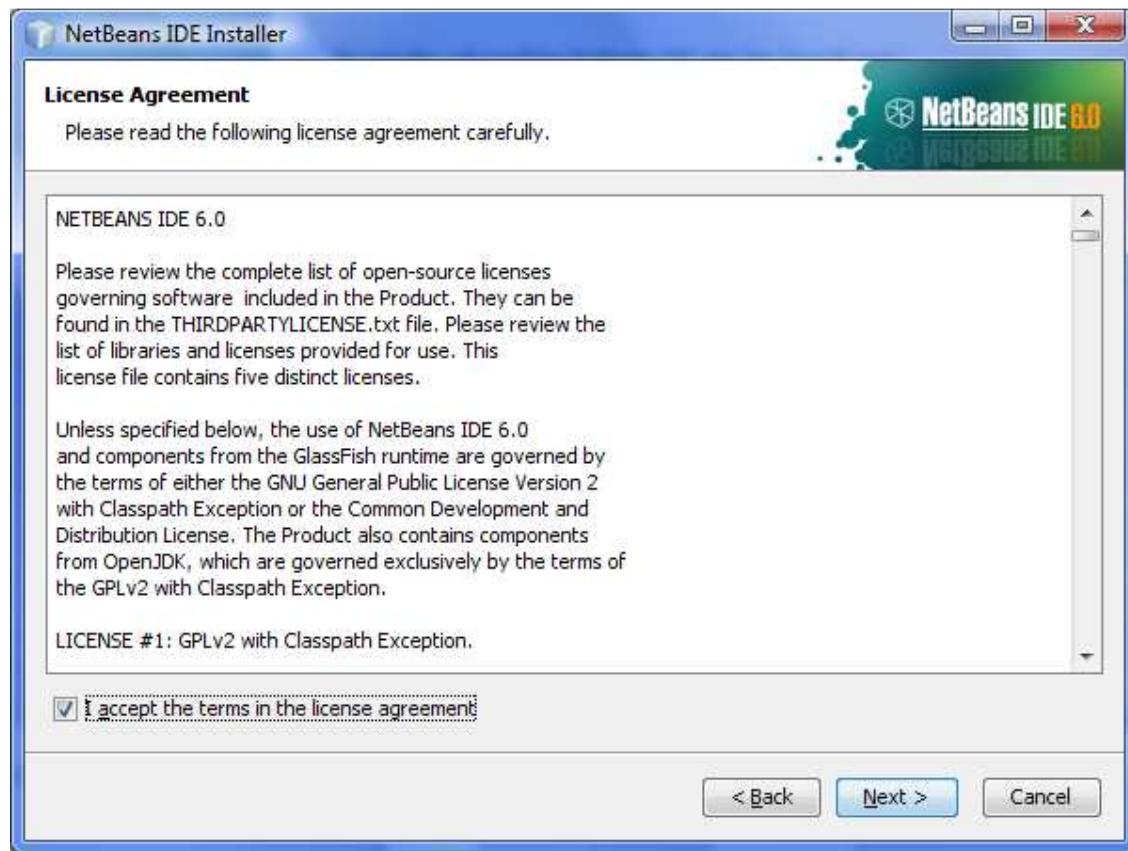
NetBeans 6.0

NetBeans merupakan IDE java yang memiliki dukungan penuh dari Sun Microsystem. Selain itu NetBeans dibuat menggunakan Swing, sehingga NetBeans bisa dibilang IDE yang paling kompeten dengan Swing.



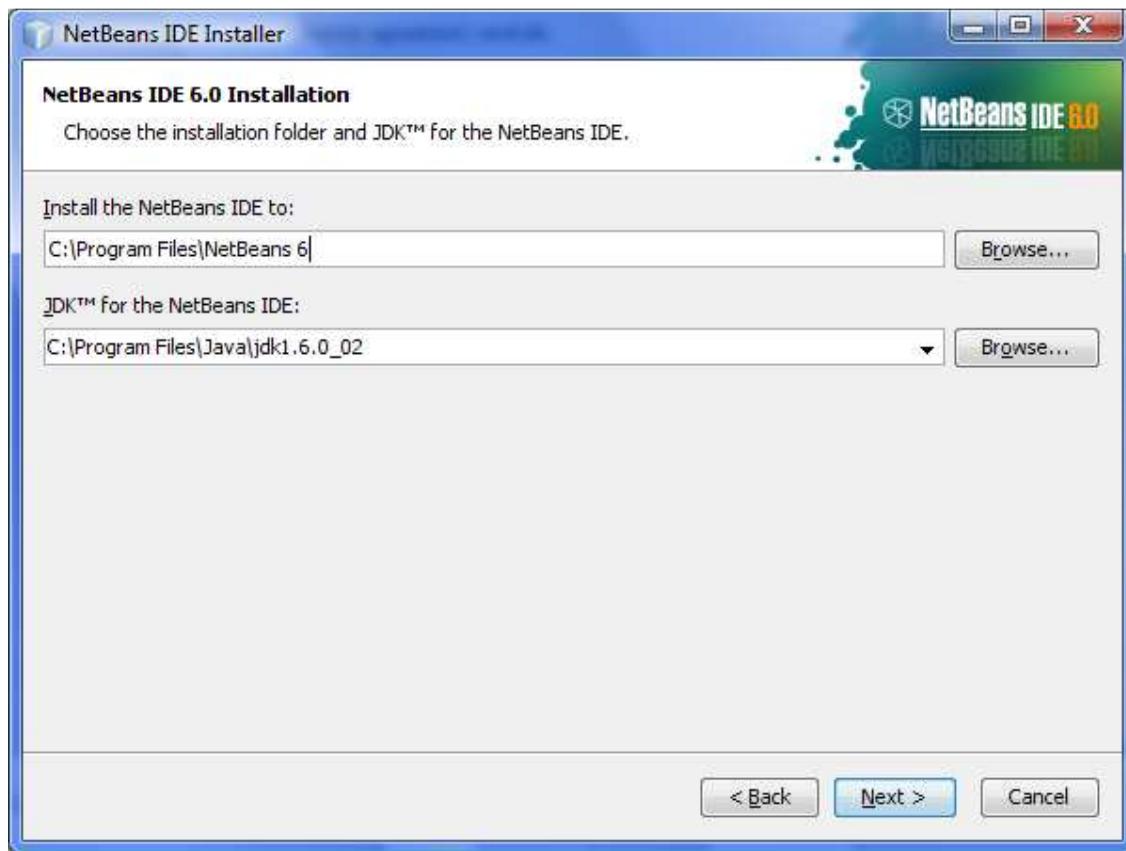
Gambar 234 NetBeans Installer

Gambar diatas merupakan tampilan awal proses instalasi NetBeans, klik Next untuk melanjutkan.



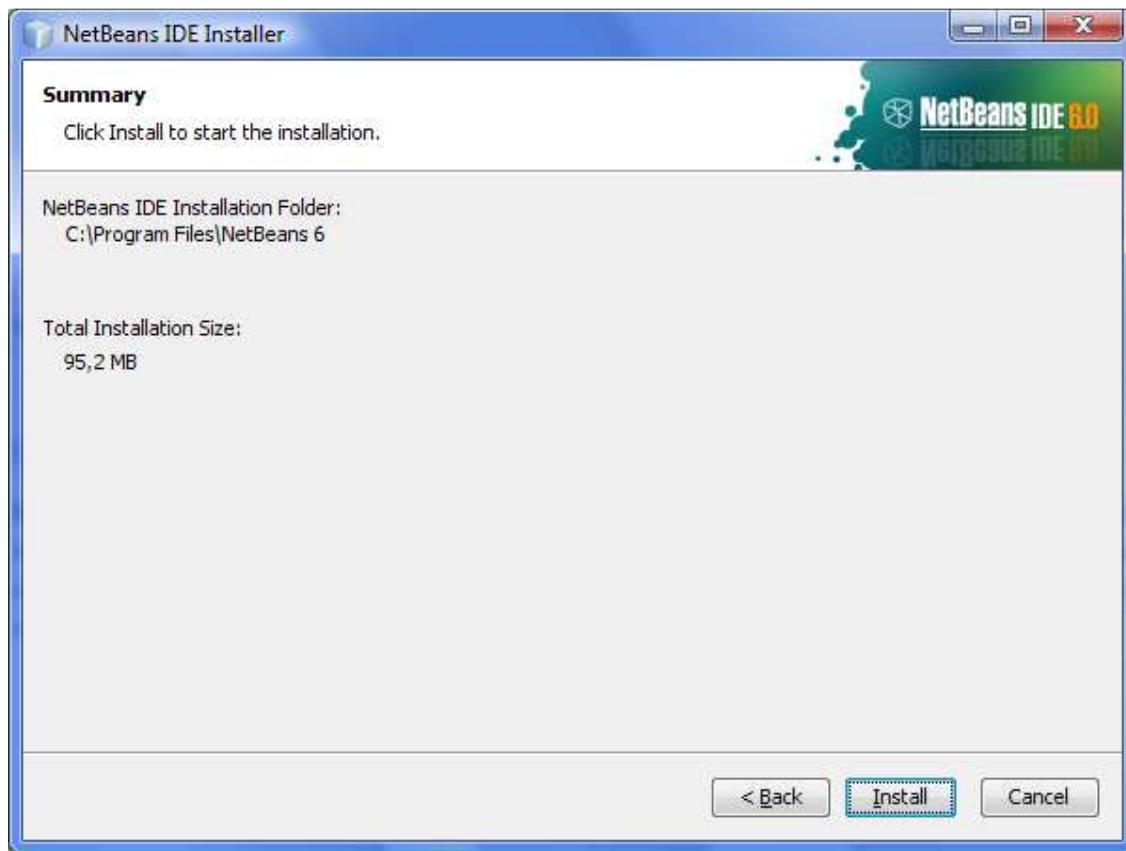
Gambar 235 License Agreement

Ceklis "I accept the terms in the license agreement" lalu klik Next.



Gambar 236 NetBeans IDE Installation

Tentukan direktori untuk instalasi NetBeans dan tentukan direktori tempat terinstalnya JDK, jika sudah klik Next.

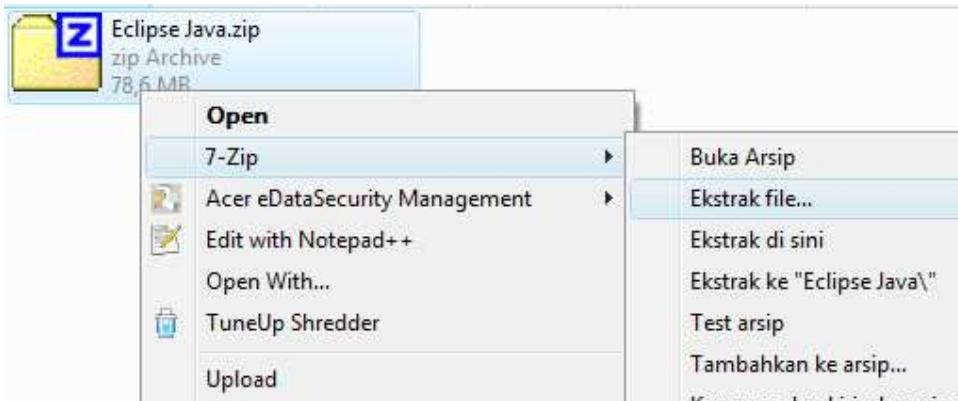


Gambar 237 Summary

Klik install untuk menginstal NetBeans. Dan setelah proses instalasi berhasil maka anda akan melihat shortcur NetBeans di Dekstop.

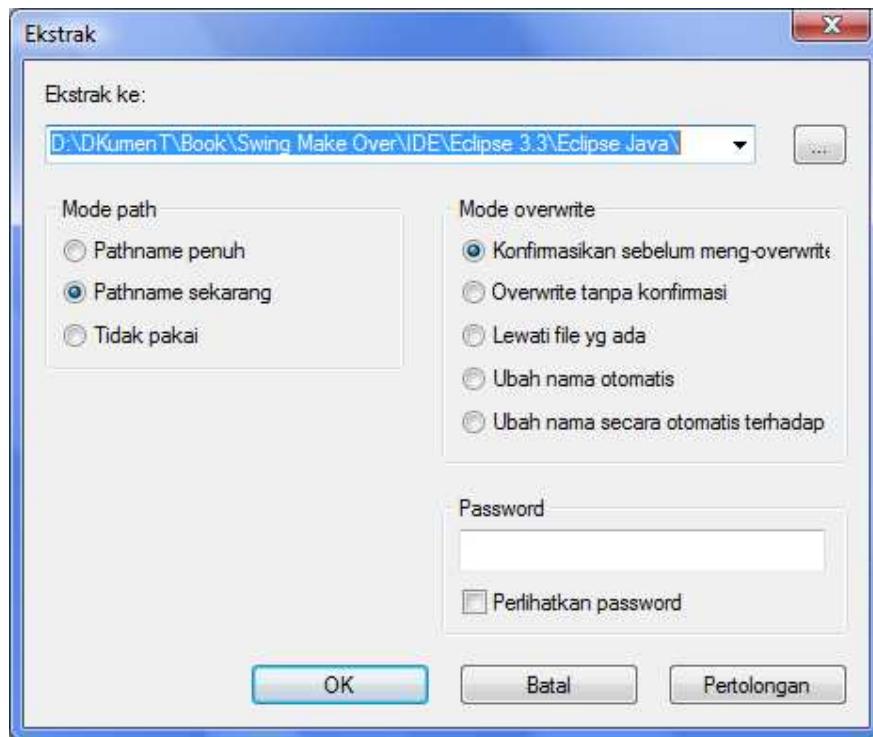
Eclipse 3.3

Berbeda dengan NetBeans, Eclipse merupakan IDE yang mendapat dukungan penuh dari IBM. Selain itu Eclipse tidak dibangun menggunakan Swing, tetapi SWT (sebuah library GUI yang dikembangkan oleh IBM). Eclipse merupakan program yang tak perlu memerlukan proses instalasi sehingga untuk menginstal Eclipse kita hanya perlu mengekstraknya.



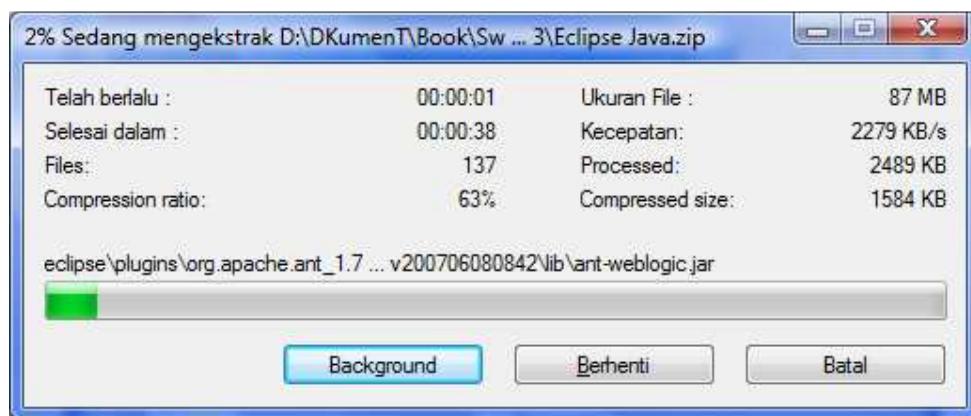
Gambar 238 Ekstrak Eclipse Java

Untuk menginstal Eclipse, ekstrak gunakan 7-Zip dengan cara mengklik kanan file Eclipse Java.Zip lalu klik menu 7-Zip > Ekstrak file...



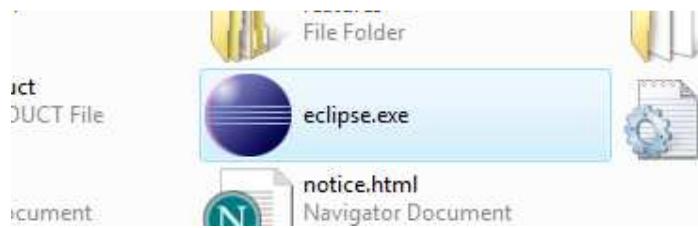
Gambar 239 Ekstrak File

Ekstrak file tersebut sesuai dengan direktori yang anda inginkan. Jika selesai klik OK untuk memulai proses ekstrak.



Gambar 240 Proses EKstrak

Tunggu hingga proses ekstrak selesai. Dan jika selesai bukalah direktori tempat anda mengekstrak lalu cari file eclipse.exe dan untuk menjalankan aplikasi Eclipse anda cukup membuka file eclipse.exe



Gambar 241 File Eclipse.EXE

Tentang Penulis



Penulis bernama **Eko Kurniawan Khannedy S.Kom.** Lahir di kota Subang tanggal 29 Desember 1988, dan besar di kota Subang. Penulis merupakan lulusan Universitas Komputer Indonesia.

Saat ini penulis menjabat sebagai **Chief Executive Officer** di **StripBandunk**, yaitu perusahaan yang bergerak di pengembangan teknologi informasi dan komunikasi.

Penulis aktif di berbagai komunitas teknologi dan saat ini penulis adalah **Leader** di komunitas **Java User Group Bandung** dan juga **Moderator** di komunitas **NetBeans Indonesia**.

Penulis dapat dihubungi di :

- echo.khannedy@gmail.com
- <http://twitter.com/khannedy>
- <http://facebook.com/khannedy>

:D