

# Buku WxWidgets

<a href="#">Bab 1. Mengenal wxWidgets</a>	12
<a href="#">1.1 Tentang buku ini</a>	12
<a href="#">1.2 Tentang wxWidgets</a>	13
<a href="#">1.3 Fitur</a>	14
<a href="#">1.4 Language binding</a>	15
<a href="#">1.5 Tim pengembangan wxWidgets</a>	16
<a href="#">1.6 Perubahan nama</a>	16
<a href="#">1.7 Tool yang dibutuhkan</a>	16
<a href="#">1.8 Kompilasi wxWidgets</a>	17
<a href="#">1.8.1 Kompilasi pada Linux</a>	17
<a href="#">1.8.2 Kompilasi pada Windows</a>	19
<a href="#">Bab 2. Hello World!</a>	21
<a href="#">2.1 Struktur direktori aplikasi</a>	21
<a href="#">2.2 Hello world sederhana 1</a>	22
<a href="#">2.2.1 Struktur direktori source code</a>	22
<a href="#">2.2.2 Output</a>	23
<a href="#">2.2.3 Source code</a>	23
<a href="#">2.2.3.1 src/hello.cpp</a>	23
<a href="#">2.2.4 Kompilasi</a>	25
<a href="#">2.2.4.1 Kompilasi di Linux (GTK+)</a>	25
<a href="#">2.2.4.2 Kompilasi di Windows</a>	27
<a href="#">2.2.5 Melihat kebutuhan pustaka</a>	36
<a href="#">2.2.5.1 Kebutuhan pustaka di Linux</a>	36
<a href="#">2.2.5.2 Kebutuhan pustaka di Windows</a>	37
<a href="#">2.3 Hello world sederhana 2</a>	37
<a href="#">2.3.1 Struktur direktori source code</a>	38
<a href="#">2.3.2 Output</a>	38
<a href="#">2.3.3 Source code</a>	38
<a href="#">2.3.3.1 src/app.h</a>	39
<a href="#">2.3.3.2 src/app.cpp</a>	39
<a href="#">2.3.3.3 src/frame.h</a>	40
<a href="#">2.3.3.4 src/frame.cpp</a>	40
<a href="#">2.3.4 Kompilasi</a>	40
<a href="#">2.3.4.1 Kompilasi di Linux (GTK+)</a>	40
<a href="#">2.3.4.2 Kompilasi di Windows</a>	42
<a href="#">2.3.5 Melihat kebutuhan pustaka</a>	42
<a href="#">2.4 Hello world sederhana 3</a>	42
<a href="#">2.4.1 Struktur direktori source code</a>	43
<a href="#">2.4.2 Output</a>	44
<a href="#">2.4.3 Source code</a>	44
<a href="#">2.4.3.1 src/app.h</a>	44

2.4.3.2 src/app.cpp.....	44
2.4.3.3 src/frame.h.....	45
2.4.3.4 src/frame.cpp.....	45
2.4.4 Kompilasi.....	46
2.4.4.1 Kompilasi di Linux (GTK+).....	46
2.4.4.2 Kompilasi di Window.....	48
2.4.5 Melihat kebutuhan pustaka.....	48
2.5 Hello world sederhana 4.....	48
2.5.1 Struktur direktori source code:.....	49
2.5.2 Output.....	49
2.5.3 Source code.....	49
2.5.3.1 src/app.h.....	50
2.5.3.2 src/app.cpp.....	50
2.5.3.3 src/frame.h.....	51
2.5.3.4 src/frame.cpp.....	51
2.5.4 Kompilasi.....	53
2.5.4.1 Kompilasi di Linux (GTK+).....	53
2.5.4.2 Kompilasi di Windows.....	54
2.5.5 Melihat kebutuhan pustaka.....	54
2.6 Window ID yang disediakan oleh wxWidgets.....	55
2.7 Mengatur Window ID sendiri.....	58
Bab 3. Event Driven Programming.....	60
3.1 Sekilas tentang event driven programming.....	60
3.2 Event dan Event handler.....	61
3.3 Menggunakan event table.....	62
3.3.1 Contoh sederhana 1: simple event.....	63
3.3.1.1 Struktur direktori source code.....	64
3.3.1.2 Source code.....	64
3.3.1.2.1 src/app.h.....	64
3.3.1.2.2 src/app.cpp.....	65
3.3.1.2.3 src/frame.h.....	65
3.3.1.2.4 src/frame.cpp.....	66
3.3.1.3 Kompilasi.....	68
3.3.1.3.1 Kompilasi di Linux (GTK+).....	68
3.3.2 Contoh sederhana 2: hello5.....	69
3.3.2.1 Struktur direktori source code.....	69
3.3.2.2 Source code.....	70
3.3.2.2.1 src/app.h.....	70
3.3.2.2.2 src/app.cpp.....	70
3.3.2.2.3 src/frame.h.....	71
3.3.2.2.4 src/frame.cpp.....	72
3.3.2.3 Kompilasi.....	73
3.3.2.3.1 Kompilasi di Linux (GTK+).....	73
3.4 Memroses event secara dinamis.....	74

3.4.1 Contoh sederhana 1: simple_event.....	75
3.4.1.1 Struktur direktori source code.....	75
3.4.1.2 Source code .....	76
3.4.1.2.1 src/app.h.....	76
3.4.1.2.2 src/app.cpp.....	76
3.4.1.2.3 src/frame.h.....	77
3.4.1.2.4 src/frame.cpp.....	78
3.4.1.3 Kompilasi.....	79
3.4.1.3.1 Kompilasi di Linux (GTK+).....	79
3.4.2 Contoh sederhana 2: hello5.....	80
3.4.2.1 Struktur direktori source code.....	80
3.4.2.2 Source code .....	81
3.4.2.2.1 src/app.h.....	81
3.4.2.2.2 src/app.cpp.....	81
3.4.2.2.3 src/frame.h.....	82
3.4.2.2.4 src/frame.cpp.....	83
3.4.2.3 Kompilasi.....	84
3.4.2.3.1 Kompilasi di Linux (GTK+).....	84
Bab 4. Layout window menggunakan sizer.....	86
4.1 Sekilas tentang sizer.....	86
4.2 Macam-macam sizer di wxWidgets.....	87
4.3 Mengatur sizer pada wxWindow.....	87
4.4 Menambahkan window ke sizer.....	87
4.5 Menambahkan space ke sizer.....	89
4.6 Menggunakan Box Sizer (wxBoxSizer).....	89
4.6.1 Struktur direktori box_sizer.....	91
4.6.2 Source code .....	91
4.6.2.1 src/app.h.....	92
4.6.2.2 src/app.cpp.....	92
4.6.2.3 src/frame.h.....	92
4.6.2.4 src/frame.cpp.....	93
4.6.3 Kompilasi.....	95
4.6.3.1 Kompilasi di Linux (GTK+).....	95
4.7 Menggunakan Static Box Sizer (wxStaticBoxSizer).....	96
4.7.1 Struktur direktori static_box_sizer.....	98
4.7.2 Source code .....	98
4.7.2.1 src/app.h.....	98
4.7.2.2 src/app.cpp.....	98
4.7.2.3 src/frame.h.....	99
4.7.2.4 src/frame.cpp.....	99
4.7.3 Kompilasi.....	101
4.7.3.1 Kompilasi di Linux (GTK+).....	101
4.8 Menggunakan Grid Sizer (wxGridSizer).....	102
4.8.1 Struktur direktori grid_sizer.....	103

4.8.2 Source code .....	104
4.8.2.1 src/app.h.....	104
4.8.2.2 src/app.cpp.....	104
4.8.2.3 src/frame.h.....	105
4.8.2.4 src/frame.cpp.....	105
4.8.3 Kompilasi.....	107
4.8.3.1 Kompilasi di Linux (GTK+).....	107
4.9 Menggunakan Flex Grid Sizer (wxFlexGridSizer).....	108
4.9.1 Struktur direktori flex_grid_sizer.....	110
4.9.2 Source code .....	110
4.9.2.1 src/app.h.....	111
4.9.2.2 src/app.cpp.....	111
4.9.2.3 src/frame.h.....	111
4.9.2.4 src/frame.cpp.....	112
4.9.3 Kompilasi.....	114
4.9.3.1 Kompilasi di Linux (GTK+).....	114
4.10 Menggunakan Grid Bag Sizer (wxGridBagSizer).....	115
4.10.1 Struktur direktori grid_bag_sizer.....	117
4.10.2 Source code .....	117
4.10.2.1 src/app.h.....	118
4.10.2.2 src/app.cpp.....	118
4.10.2.3 src/frame.h.....	118
4.10.2.4 src/frame.cpp.....	119
4.10.3 Kompilasi.....	121
4.10.3.1 Kompilasi di Linux (GTK+).....	121
Bab 5. Berbagai class wxWidgets .....	122
5.1 Managed Window.....	122
5.1.1 wxMDIParentFrame dan wxMDIChildFrame.....	122
5.1.1.1 MDI Test.....	123
5.1.1.1.1 Struktur direktori.....	123
5.1.1.1.2 Source code.....	124
5.1.1.1.2.1 src/app.h.....	124
5.1.1.1.2.2 src/app.cpp.....	124
5.1.1.1.2.3 src/frame.h.....	125
5.1.1.1.2.4 src/frame.cpp.....	126
5.1.1.1.3 Kompilasi .....	130
5.1.1.1.3.1 Kompilasi di Linux (GTK+).....	130
5.1.2 wxMiniFrame.....	131
5.1.2.1 Test Mini Frame.....	131
5.1.2.1.1 Struktur direktori.....	132
5.1.2.1.2 Source code.....	132
5.1.2.1.2.1 src/app.h.....	132
5.1.2.1.2.2 src/app.cpp.....	132
5.1.2.1.2.3 src/frame.h.....	133

5.1.2.1.2.4 src/frame.cpp.....	134
5.1.2.1.3 Kompilasi .....	134
5.1.2.1.3.1 Kompilasi di Linux (GTK+).....	134
5.1.3 wxSplashScreen.....	135
5.1.3.1 Splash Screen.....	135
5.1.3.1.1 Struktur direktori.....	136
5.1.3.1.2 Source code.....	136
5.1.3.1.2.1 src/app.h.....	137
5.1.3.1.2.2 src/app.cpp.....	137
5.1.3.1.2.3 src/frame.h.....	138
5.1.3.1.2.4 src/frame.cpp.....	138
5.1.3.1.3 Kompilasi .....	139
5.1.3.1.3.1 Kompilasi di Linux (GTK+).....	139
5.1.4 wxTipWindow dan Tip Provider.....	140
5.1.4.1 Tip Window.....	140
5.1.4.1.1 Struktur direktori.....	140
5.1.4.1.2 Source code.....	140
5.1.4.1.2.1 src/app.h.....	141
5.1.4.1.2.2 src/app.cpp.....	141
5.1.4.1.2.3 src/frame.h.....	142
5.1.4.1.2.4 src/frame.cpp.....	142
5.1.4.1.3 Kompilasi .....	142
5.1.4.1.3.1 Kompilasi di Linux (GTK+).....	143
5.1.4.2 Tip Provider.....	143
5.1.4.2.1 Struktur direktori.....	144
5.1.4.2.2 Source code.....	144
5.1.4.2.2.1 src/app.h.....	145
5.1.4.2.2.2 src/app.cpp.....	145
5.1.4.2.2.3 src/frame.h.....	146
5.1.4.2.2.4 src/frame.cpp.....	146
5.1.4.2.3 Kompilasi .....	147
5.1.4.2.3.1 Kompilasi di Linux (GTK+).....	147
5.1.5 wxWizard.....	147
5.1.5.1 Wizard .....	148
5.1.5.1.1 Struktur direktori.....	148
5.1.5.1.2 Source code.....	149
5.1.5.1.2.1 src/app.h.....	149
5.1.5.1.2.2 src/app.cpp.....	149
5.1.5.1.2.3 src/frame.h.....	150
5.1.5.1.2.4 src/frame.cpp.....	151
5.1.5.1.3 Kompilasi .....	154
5.1.5.1.3.1 Kompilasi di Linux (GTK+).....	154
5.2 Window lain-lain.....	155
5.2.1 wxGrid .....	156

5.2.1.1 Grid 0.....	156
5.2.1.1.1 Struktur direktori.....	157
5.2.1.1.2 Source code.....	157
5.2.1.1.2.1 src/app.h.....	157
5.2.1.1.2.2 src/app.cpp.....	158
5.2.1.1.2.3 src/frame.h.....	158
5.2.1.1.2.4 src/frame.cpp.....	159
5.2.1.1.3 Kompilasi.....	160
5.2.1.1.3.1 Kompilasi di Linux (GTK+).....	160
5.2.1.2 Grid 1.....	161
5.2.1.2.1 Struktur direktori.....	161
5.2.1.2.2 Source code.....	162
5.2.1.2.2.1 src/app.h.....	162
5.2.1.2.2.2 src/app.cpp.....	162
5.2.1.2.2.3 src/frame.h.....	163
5.2.1.2.2.4 src/frame.cpp.....	163
5.2.1.2.3 Kompilasi.....	164
5.2.1.2.3.1 Kompilasi di Linux (GTK+).....	165
5.2.1.3 Grid 2.....	165
5.2.1.3.1 Struktur direktori.....	166
5.2.1.3.2 Source code.....	166
5.2.1.3.2.1 src/app.h.....	166
5.2.1.3.2.2 src/app.cpp.....	167
5.2.1.3.2.3 src/frame.h.....	167
5.2.1.3.2.4 src/frame.cpp.....	168
5.2.1.3.3 Kompilasi.....	169
5.2.1.3.3.1 Kompilasi di Linux (GTK+).....	169
5.2.1.4 Grid 3.....	170
5.2.1.4.1 Struktur direktori.....	170
5.2.1.4.2 Source code.....	170
5.2.1.4.2.1 src/app.h.....	170
5.2.1.4.2.2 src/app.cpp.....	171
5.2.1.4.2.3 src/frame.h.....	171
5.2.1.4.2.4 src/frame.cpp.....	172
5.2.1.4.3 Kompilasi.....	173
5.2.1.4.3.1 Kompilasi di Linux (GTK+).....	174
5.2.2 wxToolBar.....	174
5.2.2.1 Toolbar.....	175
5.2.2.1.1 Struktur direktori.....	175
5.2.2.1.2 Source code.....	176
5.2.2.1.2.1 src/app.h.....	176
5.2.2.1.2.2 src/app.cpp.....	176
5.2.2.1.2.3 src/frame.h.....	177
5.2.2.1.2.4 src/frame.cpp.....	178

5.2.2.1.3 Kompilasi .....	181
5.2.2.1.3.1 Kompilasi di Linux (GTK+) .....	181
5.2.3 wxNotebook .....	182
5.2.3.1 Notebook .....	182
5.2.3.1.1 Struktur direktori .....	183
5.2.3.1.2 Source code .....	183
5.2.3.1.2.1 src/app.h .....	183
5.2.3.1.2.2 src/app.cpp .....	184
5.2.3.1.2.3 src/frame.h .....	184
5.2.3.1.2.4 src/frame.cpp .....	185
5.2.3.1.3 Kompilasi .....	186
5.2.3.1.3.1 Kompilasi di Linux (GTK+) .....	186
5.2.4 wxListbook .....	187
5.2.4.1 Listbook .....	187
5.2.4.1.1 Struktur direktori .....	188
5.2.4.1.2 Source code .....	188
5.2.4.1.2.1 src/app.h .....	188
5.2.4.1.2.2 src/app.cpp .....	189
5.2.4.1.2.3 src/frame.h .....	189
5.2.4.1.2.4 src/frame.cpp .....	190
5.2.4.1.3 Kompilasi .....	191
5.2.4.1.3.1 Kompilasi di Linux (GTK+) .....	191
5.2.5 wxChoicebook .....	192
5.2.5.1 Choicebook .....	192
5.2.5.1.1 Struktur direktori .....	193
5.2.5.1.2 Source code .....	193
5.2.5.1.2.1 src/app.h .....	193
5.2.5.1.2.2 src/app.cpp .....	194
5.2.5.1.2.3 src/frame.h .....	194
5.2.5.1.2.4 src/frame.cpp .....	195
5.2.5.1.3 Kompilasi .....	196
5.2.5.1.3.1 Kompilasi di Linux (GTK+) .....	196
5.3 Common dialog .....	197
5.3.1 wxDirDialog .....	198
5.3.1.1 Dir Dialog 0 .....	198
5.3.1.1.1 Struktur direktori .....	199
5.3.1.1.2 Source code .....	199
5.3.1.1.2.1 src/app.h .....	199
5.3.1.1.2.2 src/app.cpp .....	199
5.3.1.1.2.3 src/frame.h .....	200
5.3.1.1.2.4 src/frame.cpp .....	200
5.3.1.1.3 Kompilasi .....	201
5.3.1.1.3.1 Kompilasi di Linux (GTK+) .....	201
5.3.2 wxFileDialog .....	202

5.3.2.1 File Dialog 0.....	202
5.3.2.1.1 Struktur direktori.....	203
5.3.2.1.2 Source code.....	203
5.3.2.1.2.1 src/app.h.....	203
5.3.2.1.2.2 src/app.cpp.....	204
5.3.2.1.2.3 src/frame.h.....	204
5.3.2.1.2.4 src/frame.cpp.....	205
5.3.2.1.3 Kompilasi.....	206
5.3.2.1.3.1 Kompilasi di Linux (GTK+).....	206
5.3.2.2 File Dialog 1.....	207
5.3.2.2.1 Struktur direktori.....	207
5.3.2.2.2 Source code.....	207
5.3.2.2.2.1 src/app.h.....	207
5.3.2.2.2.2 src/app.cpp.....	208
5.3.2.2.2.3 src/frame.h.....	208
5.3.2.2.2.4 src/frame.cpp.....	209
5.3.2.2.3 Kompilasi.....	210
5.3.2.2.3.1 Kompilasi di Linux (GTK+).....	210
5.3.3 wxColourDialog.....	210
5.3.3.1 Color Dialog.....	211
5.3.3.1.1 Struktur direktori.....	211
5.3.3.1.2 Source code.....	212
5.3.3.1.2.1 src/app.h.....	212
5.3.3.1.2.2 src/app.cpp.....	212
5.3.3.1.2.3 src/frame.h.....	213
5.3.3.1.2.4 src/frame.cpp.....	213
5.3.3.1.3 Kompilasi.....	214
5.3.3.1.3.1 Kompilasi di Linux (GTK+).....	214
5.3.4 wxTextEntryDialog.....	215
5.3.4.1 Text Entry Dialog.....	215
5.3.4.1.1 Struktur direktori.....	216
5.3.4.1.2 Source code.....	216
5.3.4.1.2.1 src/app.h.....	216
5.3.4.1.2.2 src/app.cpp.....	217
5.3.4.1.2.3 src/frame.h.....	217
5.3.4.1.2.4 src/frame.cpp.....	218
5.3.4.1.3 Kompilasi.....	218
5.3.4.1.3.1 Kompilasi di Linux (GTK+).....	219
5.3.5 wxPasswordEntryDialog.....	219
5.3.5.1 Password Entry Dialog.....	220
5.3.5.1.1 Struktur direktori.....	220
5.3.5.1.2 Source code.....	220
5.3.5.1.2.1 src/app.h.....	220
5.3.5.1.2.2 src/app.cpp.....	221



5.3.5.1.2.3 src/frame.h.....	221
5.3.5.1.2.4 src/frame.cpp.....	222
5.3.5.1.3 Kompilasi .....	222
5.3.5.1.3.1 Kompilasi di Linux (GTK+).....	223
5.3.6 wxFontDialog.....	223
5.3.6.1 Font Dialog.....	224
5.3.6.1.1 Struktur direktori.....	224
5.3.6.1.2 Source code.....	225
5.3.6.1.2.1 src/app.h.....	225
5.3.6.1.2.2 src/app.cpp.....	225
5.3.6.1.2.3 src/frame.h.....	226
5.3.6.1.2.4 src/frame.cpp.....	226
5.3.6.1.3 Kompilasi .....	227
5.3.6.1.3.1 Kompilasi di Linux (GTK+).....	227
5.4 Berbagai control.....	228
5.5 Menu.....	229
5.6 Window layout.....	230
5.6.1 Window layout berbasis sizer.....	230
5.6.2 Window layout berbasis constrain.....	230
5.7 Device Context.....	230
5.8 Graphic Device Interface.....	231
5.9 Event.....	232
5.10 Validator.....	233
5.11 Struktur Data.....	233
5.11.1 wxDateTime.....	234
5.11.1.1 0 date time (console).....	234
5.11.1.1.1 Struktur direktori.....	235
5.11.1.1.2 Source code.....	235
5.11.1.1.2.1 src/app.cpp.....	235
5.11.1.1.3 Kompilasi .....	236
5.11.1.1.3.1 Kompilasi di Linux (GTK+).....	236
5.11.1.2 1 date time (console).....	236
5.11.1.2.1 Struktur direktori.....	237
5.11.1.2.2 Source code.....	237
5.11.1.2.2.1 src/app.cpp.....	237
5.11.1.2.3 Kompilasi .....	238
5.11.1.2.3.1 Kompilasi di Linux (GTK+).....	238
5.11.2 wxDateSpan.....	239
5.11.2.1 0 date span (console).....	239
5.11.2.1.1 Struktur direktori.....	239
5.11.2.1.2 Source code.....	239
5.11.2.1.2.1 src/app.cpp.....	239
5.11.2.1.3 Kompilasi .....	240
5.11.2.1.3.1 Kompilasi di Linux (GTK+).....	240

<a href="#">5.11.3 wxTimeSpan</a>	241
<a href="#">5.11.3.1 0 time span (console)</a>	241
<a href="#">5.11.3.1.1 Struktur direktori</a>	241
<a href="#">5.11.3.1.2 Source code</a>	242
<a href="#">5.11.3.1.2.1 src/app.cpp</a>	242
<a href="#">5.11.3.1.3 Kompilasi</a>	242
<a href="#">5.11.3.1.3.1 Kompilasi di Linux (GTK+)</a>	243
<a href="#">5.11.4 wxArrayString</a>	243
<a href="#">5.11.4.1 0 array string (console)</a>	244
<a href="#">5.11.4.1.1 Struktur direktori</a>	244
<a href="#">5.11.4.1.2 Source code</a>	244
<a href="#">5.11.4.1.2.1 src/app.cpp</a>	244
<a href="#">5.11.4.1.3 Kompilasi</a>	245
<a href="#">5.11.4.1.3.1 Kompilasi di Linux (GTK+)</a>	245
<a href="#">5.11.5 wxString</a>	246
<a href="#">5.11.5.1 0 string (console)</a>	246
<a href="#">5.11.5.1.1 Struktur direktori</a>	246
<a href="#">5.11.5.1.2 Source code</a>	247
<a href="#">5.11.5.1.2.1 src/app.cpp</a>	247
<a href="#">5.11.5.1.3 Kompilasi</a>	247
<a href="#">5.11.5.1.3.1 Kompilasi di Linux (GTK+)</a>	248
<a href="#">5.11.5.2 1 string (console)</a>	248
<a href="#">5.11.5.2.1 Struktur direktori</a>	248
<a href="#">5.11.5.2.2 Source code</a>	249
<a href="#">5.11.5.2.2.1 src/app.cpp</a>	249
<a href="#">5.11.5.2.3 Kompilasi</a>	250
<a href="#">5.11.5.2.3.1 Kompilasi di Linux (GTK+)</a>	250
<a href="#">5.11.5.3 2 string (console)</a>	251
<a href="#">5.11.5.3.1 Struktur direktori</a>	251
<a href="#">5.11.5.3.2 Source code</a>	251
<a href="#">5.11.5.3.2.1 src/app.cpp</a>	251
<a href="#">5.11.5.3.3 Kompilasi</a>	252
<a href="#">5.11.5.3.3.1 Kompilasi di Linux (GTK+)</a>	252
<a href="#">5.11.5.4 3 string (console)</a>	253
<a href="#">5.11.5.4.1 Struktur direktori</a>	253
<a href="#">5.11.5.4.2 Source code</a>	253
<a href="#">5.11.5.4.2.1 src/app.cpp</a>	253
<a href="#">5.11.5.4.3 Kompilasi</a>	254
<a href="#">5.11.5.4.3.1 Kompilasi di Linux (GTK+)</a>	254
<a href="#">5.12 RTTI (Run Time Type Information)</a>	255
<a href="#">5.13 Logging</a>	255
<a href="#">5.14 Debugging</a>	255
<a href="#">5.15 Networking</a>	256
<a href="#">5.16 IPC (Interprocess Communication)</a>	256

5.17 Framework Document/View.....	257
5.18 Framework Pencetakan.....	257
5.19 Drag and Drop dan akses clipboard.....	258
5.20 Bekerja dengan File.....	258
5.21 Stream.....	259
5.22 Threading.....	260
5.23 HTML.....	260
5.24 Virtual File System.....	261
5.25 XRC.....	261
5.26 Online Help.....	261
5.27 Database.....	261
5.28 Berbagai class lainnya.....	262
5.28.1 wxApp.....	263
5.28.1.1 App.....	263
5.28.1.1.1 Struktur direktori.....	264
5.28.1.1.2 Source code.....	264
5.28.1.1.2.1 src/app.h.....	264
5.28.1.1.2.2 src/app.cpp.....	264
5.28.1.1.2.3 src/frame.h.....	265
5.28.1.1.2.4 src/frame.cpp.....	266
5.28.1.1.3 Kompilasi.....	266
5.28.1.1.3.1 Kompilasi di Linux (GTK+).....	266
Bab 6. Pembuatan custom dialog.....	268
6.1 Contoh custom dialog 1.....	268
6.1.1 Struktur direktori custom_dialog1.....	269
6.1.2 Source code.....	269
6.1.2.1 src/app.h.....	270
6.1.2.2 src/app.cpp.....	270
6.1.2.3 src/frame.h.....	271
6.1.2.4 src/frame.cpp.....	272
6.1.2.5 src/de_dialog.h.....	275
6.1.2.6 src/de_dialog.cpp.....	278
6.1.3 Kompilasi.....	282
6.1.3.1 Kompilasi di Linux (GTK+).....	282
6.2 Contoh custom dialog 2.....	283
6.2.1 Struktur direktori custom_dialog2.....	284
6.2.2 Source code.....	284
6.2.2.1 src/app.h.....	285
6.2.2.2 src/app.cpp.....	285
6.2.2.3 src/frame.h.....	286
6.2.2.4 src/frame.cpp.....	287
6.2.2.5 src/de_dialog.h.....	289
6.2.2.6 src/de_dialog.cpp.....	292
6.2.3 Kompilasi.....	298

6.2.3.1 Kompilasi di Linux (GTK+)	298
Bab 7. Proyek Aplikasi	300
7.1 Simple Text Editor	300
7.1.1 Struktur direktori source code	301
7.1.2 Source code	301
7.1.2.1 src/app.h	302
7.1.2.2 src/app.cpp	302
7.1.2.3 src/textctrl.h	303
7.1.2.4 src/textctrl.cpp	303
7.1.2.5 src/frame.h	305
7.1.2.6 src/frame.cpp	306
7.1.3 Kompilasi	313
7.1.3.1 Kompilasi di Linux (GTK+)	313
7.2 Simple Image Viewer	314
7.2.1 Struktur direktori source code	315
7.2.2 Source code	316
7.2.2.1 src/app.h	316
7.2.2.2 src/app.cpp	316
7.2.2.3 src/frame.h	317
7.2.2.4 src/frame.cpp	318
7.2.3 Kompilasi	322
7.2.3.1 Kompilasi di Linux (GTK+)	322
rm -f src/*.o build/\$(WXPLAT)/\$(PROGRAM)	323
Bab 8. Membaca Dokumentasi wxWidgets	324
8.1 Format dokumentasi	324
8.2 Menggunakan format HTB	325
8.3 Hal-hal yang perlu diperhatikan	326
8.3.1 File header	326
8.3.2 Base class	326
8.3.3 Perhatikan constructornya	327
8.3.4 Perhatikan function overloading secara umum	327
8.3.5 Perhatikan bagian see also	328
8.3.6 Perhatikan bagian event handling	328
8.3.7 Perhatikan isu multiplatform	328

## Bab 1. Mengenal wxWidgets

Bab ini membahas tentang buku ini secara umum, pengenalan singkat wxWidgets serta kompilasi wxWidgets pada platform Windows dan Linux (GTK+). Setelah membaca buku ini, developer diharapkan mengetahui wxwidgets secara umum, serta siap untuk memulai pengembangan aplikasi memanfaatkan wxWidgets.

## **1.1 Tentang buku ini**

Buku ini dibagi dalam beberapa bab berikut:

- Bab 1: Mengenal wxWidgets. Membahas tentang buku ini secara umum, pengenalan singkat wxWidgets dan kompilasi wxWidgets pada Windows dan Linux (GTK+).
- Bab 2: Hello World! Contoh aplikasi sederhana wxWidgets serta membahas mengenai kompilasi aplikasi pada platform Windows dan Linux (GTK+).
- Bab 3: Event Driven Programming. Pembahasan mengenai event driven programming dengan wxWidgets.
- Bab 4: MenggunakanSizer. Pembahasan mengenai peletakan widget/layout menggunakan sizer.
- Bab 5: Berbagai class wxWidgets. Pembahasan tentang berbagai class wxWidgets.
- Bab 6: Custom Dialog. Pembahasan mengenai pembuatan custom dialog.
- Bab 7: Proyek aplikasi. Membangun aplikasi siap pakai dengan wxWidgets.
- Bab 8: Membaca dokumentasi. Disajikan sebagai referensi untuk memanfaatkan dokumentasi yang datang bersama wxWidgets.

Buku ini memfokuskan pengembangan aplikasi GUI multiplatform dengan bahasa pemrograman C++. Fokus akan diberikan pada pemrograman GUI dengan wxWidgets, dimana dasar pemrograman C++ (termasuk sintaks dasar C++) berada diluar cakupan buku ini.

Pembaca disarankan untuk membaca bab 1 sampai bab 4 secara berurutan. Bab 5 dapat dibaca sesuai kebutuhan. Bab 6 dan bab 7 disarankan untuk diikuti, namun tidaklah wajib. Bab 8 dapat dibaca sesuai kebutuhan.

Buku ini akan membahas pembangunan aplikasi di dalam platform Linux, namun, di bab 2, proses kompilasi aplikasi di platform Windows dan Linux akan dibahas. Setelah itu, dalam pembahasan berbagai contoh (di bab-bab seterusnya), proses kompilasi yang dibahas hanyalah pada platform Linux. Makefile yang disajikan juga hanya makefile pada platform Linux (GTK+).

Semua source code di dalam tulisan ini hanya mengasumsikan untuk dicompile dengan GCC ( ataupun Microsoft Visual C++ 6 pada bab 2).

## 1.2 Tentang wxWidgets

wxWidgets adalah framework pengembangan aplikasi multiplatform untuk bahasa C++ (dan bahasa-bahasa pemrograman lainnya) yang menyediakan fasilitas GUI, networking, multi threading, HTML, Printing, database dan lain sebagainya. Rilis stabil terbaru pada saat buku ini ditulis adalah 2.6.3. wxWidgets bisa didapatkan di websitenya, <http://www.wxwidgets.org>.



Framework ini dapat berjalan pada berbagai platform-platform berikut:

- Microsoft Windows
  - Windows 95/98/ME/NT/2000/XP/2003 (wxMSW)
  - Windows CE Pocket PC/Smartphone (wxWinCe)
- Unix (Linux, Solaris, HP-UX, IRIX, FreeBSD, OpenBSD, AIX dan lainnya)
  - dengan GTK+ 1.x dan GTK+ 2.x (wxGTK)
  - dengan X11 (wxX11)
  - dengan Motif/Lesstif (wxMotif)
- Mac OS
  - Mac OS 9.x dan OS X (wxMac)
  - Mac OS X Cocoa API (wxCocoa)
- OS/2 (wxOS2)
- Palm OS (wxPalmOS)

Dalam menyebut port/platform, istilah-istilah seperti wxMSW (Microsoft Windows), wxGTK (GTK+), wxMac (Mac OS) umum dipergunakan.

Sebagai catatan, kematangan wxWidgets pada platform-platform yang didukung tidaklah sama. Pada saat ini, wxWidgets untuk Microsoft Windows (95/98/ME/NT/2000/XP), Unix (GTK+) dan Mac OS (Carbon) dipertimbangkan sebagai port yang matang.

Dengan luasnya platform yang didukung, seorang developer dapat membangun aplikasi yang berjalan pada berbagai platform, dengan modifikasi kode minimal. Port yang akan dibahas dalam buku ini adalah wxMSW dan wxGTK.

WxWidgets juga mendukung berbagai compiler seperti:

- gcc dan mingw
- Borland C++
- Microsoft Visual C++
- Metrowerks CodeWarrior
- OpenWatcom
- dan lain sebagainya

Dengan banyaknya compiler yang didukung, developer juga memiliki kesempatan untuk memilih compiler favorit yang akan digunakan. Compiler yang akan digunakan di dalam buku ini adalah gcc 3.3.5 dan Microsoft Visual C++ 6.

### **1.3 Fitur**

Berikut ini adalah garis besar fitur yang datang bersama wxWidgets.

- cross platform. Seperti telah disebutkan sebelumnya, wxWidgets mendukung sangat banyak platform. Aplikasi yang dibangun pun akan dapat berjalan pada berbagai platform tersebut. WxWidgets akan berusaha untuk menampilkan native look and feel, namun, apabila suatu fungsi/widgets tidak tersedia pada suatu platform, wxWidgets akan berusaha untuk mengemulasikannya pada platform tersebut.
- Open source. WxWidgets merupakan free software dan open source
- Lisensi yang fleksibel. Lisensi wxWidgets memungkinkan developer untuk membangun aplikasi open source ataupun close source.
- Dokumentasi dan contoh yang lengkap. WxWidgets tergolong proyek yang didokumentasikan dengan baik. Puluhan contoh juga telah disiapkan untuk membantu proses pembelajaran.
- Dukungan widget yang kaya. WxWidgets mendukung hampir semua widget yang dibutuhkan untuk membangun aplikasi GUI dengan user interface yang modern. Berikut ini adalah contoh-contoh widget yang didukung:
  - Widget standar seperti button, text entry, scrolling list, combobox, checkbox, dan lain sebagainya

- Widget tambahan seperti Tree Control, List Control, Notebook, MDI (diemulasikan dengan tab pada platform non Windows), Scrolled Window, Splitter Window, Sash Window, Gauge, Toolbar, Property List, Grid dan lain sebagainya.
- Sistem event yang powerful. Sistem event di wxWidgets memungkinkan developer untuk mengasosiasikan event dengan fungsi pada compile time (static) dan runtime (dynamic).
- Dukungan printing. Menggunakan Windows Printing pada Windows dan menghasilkan PostScript pada Unix.
- Menyediakan berbagai fasilitas debugging
- Dukungan database. WxWidgets memiliki dukungan akan ODBC ataupun pustaka database 3<sup>rd</sup> party lainnya.
- Mendukung model Document/View.
- Mendukung drag and drop.
- Mendukung sistem update user interface yang mudah dan fleksibel
- Menyediakan dukungan yang luas akan file konfigurasi. Contoh file konfigurasi yang didukung: file INI, Registry Windows, dan lain sebagainya.
- Mendukung multithreading. WxWidgets menyediakan berbagai class untuk bekerja dengan thread dan mutex.
- Dukungan pemrograman networking. WxWidgets datang diantaranya dengan berbagai class untuk bekerja dengan protokol TCP/IP seperti FTP dan HTTP.
- Dukungan HTML. Wxwidgets datang dengan dukungan untuk menampilkan dan mencetak HTML. Sangat membantu untuk menampilkan informasi.
- Integrasi OpenGL.
- Mendukung penggambaran low level
- Dukungan terbatas untuk OLE Automation
- Dapat digunakan pada bahasa pemrograman selain C++

Sebagai catatan, wxWidgets dikembangkan dengan aktif. Developer disarankan untuk mengupdate ke versi terbaru untuk mendapatkan perbaikan/fitur baru.

## **1.4 Language binding**

Sejatinya, wxWidgets dikembangkan sebagai framework untuk bahasa pemrograman C++. Namun, saat ini, tersedia pula binding untuk bahasa-bahasa pemrograman lain seperti:

- wxBasic untuk bahasa pemrograman BASIC
- wxPerl untuk bahasa pemrograman Perl.
- WxPython untuk bahasa pemrograman Python. Sangat aktif dikembangkan. Sangat up to date.
- WxRuby untuk bahasa pemrograman Ruby
- wxSqueak untuk bahasa pemrograman SmallTalk
- wxErlang untuk bahasa pemrograman Erlang



- wxLua untuk bahasa pemrograman Lua
- wx4j untuk Java
- wxJS untuk bahasa pemrograman JavaScript
- wxHaskell untuk Haskell
- wxEiffel untuk bahasa pemrograman Eiffel
- wx.NET untuk C#/.NET
- wxCL (wxLisp) untuk Common Lisp
- wxAda untuk bahasa pemrograman Ada
- wxPHP untuk bahasa pemrograman PHP
- wxEuphoria untuk bahasa pemrograman Euphoria

## **1.5 Tim pengembangan wxWidgets**

wxWidgets dikembangkan oleh tim yang terdiri dari tim inti dan kontributor mayor (dengan bantuan dari kontributor lainnya di seluruh dunia). Berikut ini adalah anggota tim inti wxWidgets:

- Dr. Robert Roebling. Bertanggung jawab untuk port GTK+ dan membantu port X11. Saat ini berkonsentrasi pada penggunaan wxWidgets untuk embedded Linux.
- Dr. Vadim Zeitlin. Bekerja pada sebagian besar port dan class-class penting baru.
- Dr. Stefan Csomor. Bertanggung jawab pada wxMac.
- Dr. Julian Smart. Penemu wxWidgets dan sampai saat ini masih sangat aktif dalam pengembangan wxWidgets.
- Vaclav Slavik. bertanggung jawab untuk dukungan HTML, XRC wxMGL, build system yang baru dan lain sebagainya.
- Robin Dunn. Bertanggung jawab untuk pengembangan wxPython.

## **1.6 Perubahan nama**

wxWidgets sebelumnya bernama wxWindows. Pada september 2003, Microsoft mendekati Julian Smart untuk mendiskusikan perihal nama wxWindows, di mana, Microsoft memiliki trademark nama Windows di UK.

Diskusi tersebut menghasilkan:

- Setelah permintaan yang sopan dari Microsoft, dan setelah berpikir dengan cermat, tim memutuskan untuk mengganti nama project menjadi wxWidgets.
- Meminta maaf kepada semua pengguna wxWindows dan developer project yang menggunakan wxWindows, untuk ketidaknyamanan (waktu singkat) yang disebabkan oleh perubahan nama.

Terdapat tiga point utama sesuai kesepakatan:

- Julian Smart dan Robin Dunn akan berhenti menggunakan nama wxWindows

pada Agustus 2004

- Microsoft akan meredireksi akses ke domain lama ke domain baru, wxwidgets.org, selama paling sedikit 1 tahun setelahnya.
- Akan ada kompensasi dari sisi finansial, dimana setelah dipotong biaya, akan digunakan untuk membiayai pengembangan pustaka dan tool. Kontributor dapat mengajukan klaim kepada Julian Smart untuk kompensasi waktu yang terbuang akibat perubahan nama.

## **1.7 Tool yang dibutuhkan**

Berikut ini adalah tool yang dibutuhkan untuk bekerja dengan wxWidgets.

### Windows

- Microsoft Visual C++. Buku ini menggunakan versi 6.
- Windows yang digunakan penulis adalah Windows 2000.

### Linux

- Gcc. Buku ini menggunakan versi 3.3.5
- Make. Buku ini menggunakan GNU make versi 3.80.
- Sebuah editor yang memadai. Buku ini menggunakan Vim. Fasilitas yang diaktifkan:
  - Nowrap (set nowrap)
  - Syntax highlighting (syntax on)
  - Konversi tab menjadi 4 spasi: (set tabstop=4, set shiftwidth=4, set expandtab)

## **1.8 Kompilasi wxWidgets**

Walaupun binary wxWidgets tersedia pada berbagai distribusi Linux, buku ini akan menggunakan pendekatan kompilasi sendiri dari source code. Pembahasan kompilasi akan dibahas pada platform Windows dan Linux.

Source code yang dibutuhkan adalah wxWidgets-2.6.3.tar.bz2, yang bisa didapatkan pada CDROM pelengkap buku ini, atau dapat didownload dari website wxWidgets.

### **1.8.1 Kompilasi pada Linux**

Sebagai catatan, kompilasi wxWidgets pada sistem Linux (GTK+) mewajibkan kita untuk memiliki header-header/file development GTK+ (dan dependency-nya; cukup banyak). File-file tersebut umumnya dipaketkan dengan akhiran nama file paket -devel (Red Hat, SUSE dan teman-teman) atau -dev (Debian dan teman-teman). Umumnya, semua paket yang diperlukan sudah dipaketkan ke dalam media

distribusi yang digunakan. Karena instalasi paket per paket secara manual cukup merepotkan (disebabkan oleh dependency), disarankan untuk menggunakan tool seperti APT untuk mempermudah instalasi. Instalasi file-file development tersebut berada di luar cakupan buku ini.

File development lain seperti pustaka gambar (PNG, JPEG, TIFF, dan lain-lain) juga disarankan untuk tersedia.

Berikut ini adalah langkah-langkah kompilasi wxWidgets pada sistem Linux setelah semua file development yang dibutuhkan telah terinstall.

- Ekstraklah arsip wxWidgets-2.6.3.tar.bz2

```
$ tar jxvf wxWidgets-2.6.3.tar.bz2
```

- Masuklah ke direktori hasil ekstraksi:

```
$ cd wxWidgets-2.6.3
```

- Lakukan konfigurasi. Berikanlah prefix yang diinginkan. Di dalam buku ini, prefix yang digunakan adalah /home/nop/wxgtk. Untuk menggunakan GTK+, berikanlah opsi `--with-gtk`. Buku ini juga menggunakan build dengan unicode diaktifkan (`--enable-unicode`). Berikut ini adalah contoh perintah konfigurasi:

```
$ ./configure --prefix=/home/nop/wxgtk --enable-unicode --with-gtk
```

Berikut ini adalah ringkasan hasil konfigurasi (ditampilkan setelah `./configure` selesai dieksekusi):

Configured wxWidgets 2.6.3 for `i686-pc-linux-gnu'

Which GUI toolkit should wxWidgets use?	GTK+ 2
Should wxWidgets be compiled into single library?	no
Should wxWidgets be compiled in debug mode?	no
Should wxWidgets be linked as a shared library?	yes
Should wxWidgets be compiled in Unicode mode?	yes
What level of wxWidgets compatibility should be enabled?	
wxWidgets 2.2	no
wxWidgets 2.4	yes
Which libraries should wxWidgets use?	
jpeg	sys
png	sys
regex	builtin
tiff	sys
zlib	sys
odbc	no
expat	sys

<b>libmspack</b>	<b>no</b>
<b>sdl</b>	<b>no</b>
<b>gnomeprint</b>	<b>no</b>
<b>hildon</b>	<b>no</b>

Keluaran pada sistem lain mungkin berbeda, namun, karena kita akan menggunakan GTK+, maka pastikan pada Which GUI toolkit should wxWidgets use? Tertulis GTK+ (disarankan GTK+ 2). Pastikan juga pada Should wxWidgets be compiled in Unicode mode? Tertulis yes.

- Lakukan kompilasi dengan menjalankan make. Pada sistem penulis (Celeron 2.5 Ghz, RAM 768 MB, Debian GNU/Linux 3.1), instalasi dilakukan dalam waktu kurang lebih 50 menit.

\$ make

- Lakukan instalasi dengan menjalankan make install:

\$ make install

Pastikan proses instalasi diakhiri oleh tulisan berikut ini.

-----

The installation of wxWidgets is finished. On certain platforms (e.g. Linux) you'll now have to run ldconfig if you installed a shared library and also modify the LD\_LIBRARY\_PATH (or equivalent) environment variable.

wxWidgets comes with no guarantees and doesn't claim to be suitable for any purpose.

Read the wxWidgets Licence on licencing conditions.

-----

- Pastikan program wx-config berada di dalam path. Karena prefix penulis adalah di /home/nop/wxgtk, maka penulis perlu mendaftarkan /home/nop/wxgtk/bin di dalam \$PATH. Penulis menambahkan entry berikut ini ke dalam file ~/.bashrc:

```
export PATH=$PATH:/home/nop/wxgtk/bin
```

Sebelum memulai kompilasi, pastikan baris perintah tersebut telah dieksekusi (bisa dengan logout dan login kembali), secara eksplisit menjalankan (meng-source)

~/bashrc (source ~/.bashrc), atau secara eksplisit menjalankan perintah tersebut di shell.

- Setelah itu, pastikanlah semua pustaka wxWidgets dapat diakses oleh aplikasi. Dalam buku ini, dimana instalasi wxWidgets dilakukan pada /home/nop/wxgtk, maka /home/nop/wxgtk/lib/ perlu didaftarkan sebagai lokasi pustaka sistem. Penulis menambahkan entry berikut ini ke dalam file ~/.bashrc:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/nop/wxgtk/lib
```

Sebelum memulai menjalankan aplikasi yang dibangun dengan wxWidgets, pastikan baris perintah tersebut telah dieksekusi (bisa dengan logout dan login kembali), secara eksplisit menjalankan (meng-source) ~/.bashrc (source ~/.bashrc), atau secara eksplisit menjalankan perintah tersebut di shell.

## 1.8.2 Kompilasi pada Windows

Sebelum melakukan kompilasi, pastikanlah Microsoft Visual C++ 6 telah terinstall di sistem.

Pada buku ini, IDE Microsoft Visual C++ 6 tidak akan digunakan. Sebagai gantinya, makefile lah yang akan digunakan.

Untuk melakukan kompilasi wxWidgets, lakukanlah langkah-langkah berikut:

- Ekstraklah arsip wxWidgets-2.6.3.tar.bz2 ke [c:\wx](#)
- Bukalah command line (dos prompt)
- Masuklah ke direktori c:\wx\build\msw
- Berikanlah perintah:

```
nmake -f makefile.vc BUILD=release UNICODE=1
```

- Tunggulah selama proses kompilasi berlangsung.

## Bab 2. Hello World!

Bab ini akan membahas secara mendetail contoh aplikasi sederhana memanfaatkan wxWidgets. Bab ini juga akan membahas proses kompilasi pada platform Windows dan Linux. Diharapkan, setelah membaca bab ini, developer memahami dasar pembangunan aplikasi dengan wxWidgets.

### 2.1 Struktur direktori aplikasi

Buku ini akan menggunakan kesepakatan struktur direktori berikut untuk satu aplikasi:

```
<nama_singkat_aplikasi>/
  src/
    <source code>
    <source code>
    <source code>
    ...
  build/
    <platform>/
      <nama_executable>
    <platform>/
    <platform>/
    ...
  makefile.<platform>
  makefile.<platform>
  makefile.<platform>
  ...
```

Sebagai contoh:

```
minimal/
  src/
    minimal.cpp
  build/
    gtk/
      minimal
    msw/
```

minimal.exe  
makefile.gtk  
makefile.msw

## 2.2 Hello world sederhana 1

Aplikasi hello world yang dibangun akan memiliki fitur-fitur berikut:

- Memiliki sebuah window utama yang akan memiliki title Hello World!
- User dapat melakukan klik pada tombol close pada title bar untuk menterminasi aplikasi



Versi pertama dari hello world sederhana ini adalah versi dimana developer hanya menggunakan satu file source code saja. Di dalam source code ini, developer mendeklarasikan class dan menuliskan implementasi class tersebut. Cara penulisan source code seperti ini sangat membantu proses pembelajaran ataupun pembuatan aplikasi kecil, namun disarankan untuk tidak digunakan dalam pembangunan aplikasi (yang berpotensi) besar.

Cara penulisan source code ini dibahas untuk tujuan pembahasan contoh sederhana aplikasi yang dibangun dengan wxWidgets. Untuk selanjutnya, cara ini tidak digunakan lagi.

### 2.2.1 Struktur direktori source code:

```
hello1/  
  build/  
    gtk/  
      hello  
    msw/  
      hello.exe  
  src/  
    hello.cpp  
  makefile.gtk  
  makefile.vc
```

### 2.2.2 Output

Ketika build/gtk/hello dijalankan:

- Sebuah frame kosong akan ditampilkan
- Frame tersebut akan memiliki title bar yang bertuliskan Hello World!
- Apabila tombol close pada title bar diklik, maka aplikasi akan diterminasi

### 2.2.3 Source code

#### 2.2.3.1 *src/hello.cpp*

Berikut ini adalah source code src/hello.cpp:

```
#include "wx/wx.h"
```

```
class App : public wxApp  
{  
public:  
  
    virtual bool OnInit ();  
};
```

```
class Frame: public wxFrame  
{
```



public:

```
Frame ( const wxString& title );
```

```
};
```

```
DECLARE_APP ( App )
```

```
IMPLEMENT_APP ( App )
```

```
bool App :: OnInit ()
```

```
{  
    Frame * frame = new Frame ( _ ("Hello World!") );  
  
    frame -> Show ( true );  
  
    return true;  
}
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
```

```
{  
  
}
```

Penjelasan source code:

- Setiap aplikasi wxWidgets mendefinisikan sebuah class aplikasi, yang diturunkan dari class wxApp. Dalam contoh ini nama class yang dipergunakan adalah class App.
- Hanya terdapat satu instance dari class tersebut, dan instance tersebut mewakili aplikasi yang sedang berjalan
- Class aplikasi seharusnya mendefinisikan sebuah method OnInit () yang akan dijalankan pada saat wxWidgets telah siap mengeksekusi kode user. Method OnInit() ini ekuivalen dengan fungsi main() pada C, atau WinMain() pada aplikasi win32.
- Method OnInit() umumnya berisikan kode untuk pembuatan window utama, memeriksa argumen command line yang diberikan oleh user, mengatur struktur data aplikasi, dan tugas-tugas inisialisasi lainnya.
- Method OnInit() mengembalikan tipe data bool, yang bernilai true atau false.

Apabila mengembalikan nilai true, maka event loop akan dijalankan (aplikasi akan berjalan). Apabila mengembalikan nilai false, maka wxWidgets akan membersihkan memori yang terpakai, dan aplikasi akan diterminasi.

- Dalam contoh ini, di dalam OnInit(), terdapat dua tugas:
  - pembuatan top level Window menggunakan wxFrame dan memberikan title Hello World!
  - Meminta agar frame yang baru dibuat sebelumnya untuk ditampilkan. Window top level harus secara eksplisit ditampilkan. Berbeda dengan halnya child window.
- Terdapat dua macro yaitu IMPLEMENT\_APP yang membantu developer untuk membuat objek aplikasi dan DECLARE\_APP yang akan membantu developer untuk akses kepada reference objek aplikasi.
- Di dalam aplikasi sederhana ini, terdapat dua class yang dipergunakan:
  - App (diturunkan public dari wxApp)
    - Berisikan method OnInit()
  - Frame (diturunkan public dari wxFrame)
    - Berisikan constructor Frame ( const wxString& title ). Di dalam App :: OnInit(), terdapat deklarasi pembuatan objek Frame dengan memberikan satu argumen berupa title frame. Di dalam Frame :: Frame (), title yang diberikan kepadanya akan dilewatkan kepada wxFrame :: wxFrame ().
    - Frame :: Frame () sendiri saat ini belum melakukan tindakan khusus. Umumnya, di dalam constructor ini, developer akan membuat menu, toolbar, statusbar dan lain sebagainya.
- Terdapat satu macro \_() yang dipergunakan. Gunakanlah macro ini ketika bekerja dengan string. Macro ini dapat diartikan sebagai:
  - mempersiapkan string dan karakter untuk mengijinkan aplikasi dikompilasi dalam mode unicode
  - mempersiapkan string untuk dapat diterjemahkan ke bahasa lain.

## 2.2.4 Kompilasi

### 2.2.4.1 Kompilasi di Linux (GTK+)

Kompilasi di Linux memanfaatkan penggunaan makefile. Dalam hal ini, nama file makefilenya adalah makefile.gtk. Berikut ini adalah isi makefile.gtk:

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk

PROGRAM = hello
OBJECTS = src/hello.o

# system
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES: .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all: $(PROGRAM)

$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

#### Penjelasan makefile:

- Penulis telah membuat sebuah contoh makefile sederhana untuk memudahkan kompilasi. Makefile ini dapat dimodifikasi sesuai keperluan.
- Makefile memiliki dua tugas:
  - all, yang akan melakukan kompilasi
  - clean, yang akan menghapus semua file object dan executable
- Yang mungkin diubah oleh user adalah:
  - WXCONFIG, nama file wx-config. Defaultnya adalah wx-config.
  - WXPLAT, nama platform/port. Usahakan untuk tidak memberikan spasi di dalam nama file. Usahakan pula untuk memberi nama sesuai port wxWidgets. Misal, gtk, x11, msw dan lain sebagainya.
  - PROGRAM, nama executable
  - OBJECTS, file-file object. Apabila lebih dari satu file, pisahkanlah dengan spasi.

Untuk melakukan kompilasi, berikanlah perintah berikut ini:

```
$ make -f makefile.gtk
g++ -c `wx-config --cxxflags` -o src/hello.o src/hello.cpp
g++ -o hello src/hello.o `wx-config --libs`
mkdir -p build/gtk
mv hello build/gtk
rm -f src/*.o
```

Setelah kompilasi dilakukan, akan terbentuk sebuah executable dengan nama hello di dalam direktori build/gtk/.

Apabila penghapusan file-file hasil kompilasi ingin dilakukan (cleaning), berikanlah argumen clean. Sebagai contoh:

```
$ make -f makefile.gtk clean
rm -f src/*.o build/gtk/hello
```

Setelah perintah tersebut diberikan, semua jejak hasil kompilasi akan dihapus.

### **2.2.4.2 Kompilasi di Windows**

Kompilasi dalam buku ini memanfaatkan penggunaan makefile. Dalam hal ini, nama file makefilenya adalah makefile.vc. Sebagai catatan, file ini juga disertakan dalam CDROM buku ini.

Berikut ini adalah isi makefile.vc:

```
# simple makefile for simple wx project
# (c) Nop, 5 Jul 2006
# adapted from wx sample's makefile.vc
# please edit:
# - variables (below)
# - objs (near EOF)

# variables
PROGRAM_NAME = hello

PROGRAM = $(WXPLAT)\$(PROGRAM_NAME).exe
OBJECTS = src\hello.obj
```

```
EXTRA_INCLUDES = /I"C:\Program Files\Microsoft Visual Studio\VC98\Include" \
                /lc:\wx\include\msvc \
                /lc:\wx\include\wx \
                /lc:\wx\include
```

```
UNICODE = 1
BUILD = release
WXPLAT = msw
```

```
# set if needed
CC = cl
CXX = cl
CFLAGS =
CXXFLAGS =
CPPFLAGS =
LDFLAGS =
CPP = $(CC) /EP /nologo
SHARED = 0
WXUNIV = 0
MSLU = 0
TARGET_CPU = $(CPU)
DEBUG_INFO = default
DEBUG_FLAG = default
DEBUG_RUNTIME_LIBS = default
MONOLITHIC = 0
USE_GUI = 1
USE_HTML = 1
USE_XRC = 1
USE_OPENGL = 0
USE_ODBC = 0
USE_QA = 0
USE_EXCEPTIONS = 1
USE_RTTI = 1
USE_THREADS = 1
OFFICIAL_BUILD = 0
VENDOR = custom
WX_FLAVOUR =
WX_LIB_FLAVOUR =
CFG =
CPPUNIT_CFLAGS =
CPPUNIT_LIBS =
RUNTIME_LIBS = dynamic
```

```
WX_RELEASE_NODOT = 26
```

```
LIBDIRNAME = c:\wx\lib\vc$(DIR_SUFFIX_CPU)_$(LIBTYPE_SUFFIX)$(CFG)
```

```
SETUPHDIR = \
```

```
$(LIBDIRNAME)\$(PORTNAME)$(WXUNIVNAME)$(WXUNICODEFLAG)$  
(WXDEBUGFLAG)  
APP_CXXFLAGS = /M$(__RUNTIME_LIBS_8)$(__DEBUGRUNTIME_3) /DWIN32 \  
$(__DEBUGINFO_0) /Fd$(WXPLAT)\$(PROGRAM_NAME).pdb $  
(__DEBUGRUNTIME_2_p) \  
$(__OPTIMIZEFLAG_4) $(__NO_VC_CRTDBG_p) /D_WXMSW_ $  
(__WXUNIV_DEFINE_p) \  
$(__DEBUG_DEFINE_p) $(__EXCEPTIONS_DEFINE_p) $(__RTTI_DEFINE_p) \  
$(__THREAD_DEFINE_p) $(__UNICODE_DEFINE_p) $(__MSLU_DEFINE_p) \  
/I$(SETUPHDIR) $(EXTRA_INCLUDES) /W4 /I. $(__DLLFLAG_p) /D_WINDOWS \  
/DNOPCH $(__RTTI_FLAG_9) $(__EXCEPTIONS_FLAG_10) $(CPPFLAGS) \  
$(CXXFLAGS)
```

```
# system, please do not edit
```

```
### Conditionally set variables: ###
```

```
WXPLAT = build\$(WXPLAT)
```

```
!if "$(USE_GUI)" == "0"
```

```
PORTNAME = base
```

```
!endif
```

```
!if "$(USE_GUI)" == "1"
```

```
PORTNAME = msw
```

```
!endif
```

```
!if "$(BUILD)" == "debug" && "$(DEBUG_FLAG)" == "default"
```

```
WXDEBUGFLAG = d
```

```
!endif
```

```
!if "$(DEBUG_FLAG)" == "1"
```

```
WXDEBUGFLAG = d
```

```
!endif
```

```
!if "$(UNICODE)" == "1"
```

```
WXUNICODEFLAG = u
```

```
!endif
```

```
!if "$(WXUNIV)" == "1"
```

```
WXUNIVNAME = univ
```

```
!endif
```

```
!if "$(TARGET_CPU)" == "amd64"
```

```
DIR_SUFFIX_CPU = _amd64
```

```
!endif
```

```
!if "$(TARGET_CPU)" == "amd64"
DIR_SUFFIX_CPU = _amd64
!endif
!if "$(TARGET_CPU)" == "ia64"
DIR_SUFFIX_CPU = _ia64
!endif
!if "$(TARGET_CPU)" == "ia64"
DIR_SUFFIX_CPU = _ia64
!endif
!if "$(SHARED)" == "1"
WXDLLFLAG = dll
!endif
!if "$(SHARED)" == "0"
LIBTYPE_SUFFIX = lib
!endif
!if "$(SHARED)" == "1"
LIBTYPE_SUFFIX = dll
!endif
!if "$(TARGET_CPU)" == "amd64"
LINK_TARGET_CPU = /MACHINE:AMD64
!endif
!if "$(TARGET_CPU)" == "amd64"
LINK_TARGET_CPU = /MACHINE:AMD64
!endif
!if "$(TARGET_CPU)" == "ia64"
LINK_TARGET_CPU = /MACHINE:IA64
!endif
!if "$(TARGET_CPU)" == "ia64"
LINK_TARGET_CPU = /MACHINE:IA64
!endif
!if "$(MONOLITHIC)" == "0"
EXTRALIBS_FOR_BASE =
!endif
!if "$(MONOLITHIC)" == "1"
EXTRALIBS_FOR_BASE =
!endif
!if "$(BUILD)" == "debug" && "$(DEBUG_INFO)" == "default"
__DEBUGINFO_0 = /Zi
!endif
!if "$(BUILD)" == "release" && "$(DEBUG_INFO)" == "default"
__DEBUGINFO_0 =
!endif
!if "$(DEBUG_INFO)" == "0"
__DEBUGINFO_0 =
```

```
endif
!if "$(DEBUG_INFO)" == "1"
__DEBUGINFO_0 = /Zi
endif
!if "$(BUILD)" == "debug" && "$(DEBUG_INFO)" == "default"
__DEBUGINFO_1 = /DEBUG
endif
!if "$(BUILD)" == "release" && "$(DEBUG_INFO)" == "default"
__DEBUGINFO_1 =
endif
!if "$(DEBUG_INFO)" == "0"
__DEBUGINFO_1 =
endif
!if "$(DEBUG_INFO)" == "1"
__DEBUGINFO_1 = /DEBUG
endif
!if "$(BUILD)" == "debug" && "$(DEBUG_RUNTIME_LIBS)" == "default"
__DEBUGRUNTIME_2_p = /D_DEBUG
endif
!if "$(BUILD)" == "release" && "$(DEBUG_RUNTIME_LIBS)" == "default"
__DEBUGRUNTIME_2_p =
endif
!if "$(DEBUG_RUNTIME_LIBS)" == "0"
__DEBUGRUNTIME_2_p =
endif
!if "$(DEBUG_RUNTIME_LIBS)" == "1"
__DEBUGRUNTIME_2_p = /D_DEBUG
endif
!if "$(BUILD)" == "debug" && "$(DEBUG_RUNTIME_LIBS)" == "default"
__DEBUGRUNTIME_2_p_1 = /d_DEBUG
endif
!if "$(BUILD)" == "release" && "$(DEBUG_RUNTIME_LIBS)" == "default"
__DEBUGRUNTIME_2_p_1 =
endif
!if "$(DEBUG_RUNTIME_LIBS)" == "0"
__DEBUGRUNTIME_2_p_1 =
endif
!if "$(DEBUG_RUNTIME_LIBS)" == "1"
__DEBUGRUNTIME_2_p_1 = /d_DEBUG
endif
!if "$(BUILD)" == "debug" && "$(DEBUG_RUNTIME_LIBS)" == "default"
__DEBUGRUNTIME_3 = d
endif
!if "$(BUILD)" == "release" && "$(DEBUG_RUNTIME_LIBS)" == "default"
```



```
__DEBUGRUNTIME_3 =
!endif
!if "$(DEBUG_RUNTIME_LIBS)" == "0"
__DEBUGRUNTIME_3 =
!endif
!if "$(DEBUG_RUNTIME_LIBS)" == "1"
__DEBUGRUNTIME_3 = d
!endif
!if "$(BUILD)" == "debug"
__OPTIMIZEFLAG_4 = /Od
!endif
!if "$(BUILD)" == "release"
__OPTIMIZEFLAG_4 = /O2
!endif
!if "$(USE_THREADS)" == "0"
__THREADSFLAG_7 = L
!endif
!if "$(USE_THREADS)" == "1"
__THREADSFLAG_7 = T
!endif
!if "$(RUNTIME_LIBS)" == "dynamic"
__RUNTIME_LIBS_8 = D
!endif
!if "$(RUNTIME_LIBS)" == "static"
__RUNTIME_LIBS_8 = $(__THREADSFLAG_7)
!endif
!if "$(USE_RTTI)" == "0"
__RTTIFLAG_9 =
!endif
!if "$(USE_RTTI)" == "1"
__RTTIFLAG_9 = /GR
!endif
!if "$(USE_EXCEPTIONS)" == "0"
__EXCEPTIONSFLAG_10 =
!endif
!if "$(USE_EXCEPTIONS)" == "1"
__EXCEPTIONSFLAG_10 = /EHsc
!endif
!if "$(BUILD)" == "debug" && "$(DEBUG_RUNTIME_LIBS)" == "0"
__NO_VC_CRTDBG_p = /D__NO_VC_CRTDBG__
!endif
!if "$(BUILD)" == "release" && "$(DEBUG_FLAG)" == "1"
__NO_VC_CRTDBG_p = /D__NO_VC_CRTDBG__
!endif
```

```
!if "$(BUILD)" == "debug" && "$(DEBUG_RUNTIME_LIBS)" == "0"
__NO_VC_CRTDBG_p_1 = /d __NO_VC_CRTDBG__
!endif
!if "$(BUILD)" == "release" && "$(DEBUG_FLAG)" == "1"
__NO_VC_CRTDBG_p_1 = /d __NO_VC_CRTDBG__
!endif
!if "$(WXUNIV)" == "1"
__WXUNIV_DEFINE_p = /D__WXUNIVERSAL__
!endif
!if "$(WXUNIV)" == "1"
__WXUNIV_DEFINE_p_1 = /d __WXUNIVERSAL__
!endif
!if "$(BUILD)" == "debug" && "$(DEBUG_FLAG)" == "default"
__DEBUG_DEFINE_p = /D__WXDEBUG__
!endif
!if "$(DEBUG_FLAG)" == "1"
__DEBUG_DEFINE_p = /D__WXDEBUG__
!endif
!if "$(BUILD)" == "debug" && "$(DEBUG_FLAG)" == "default"
__DEBUG_DEFINE_p_1 = /d __WXDEBUG__
!endif
!if "$(DEBUG_FLAG)" == "1"
__DEBUG_DEFINE_p_1 = /d __WXDEBUG__
!endif
!if "$(USE_EXCEPTIONS)" == "0"
__EXCEPTIONS_DEFINE_p = /DwxNO_EXCEPTIONS
!endif
!if "$(USE_EXCEPTIONS)" == "0"
__EXCEPTIONS_DEFINE_p_1 = /d wxNO_EXCEPTIONS
!endif
!if "$(USE_RTTI)" == "0"
__RTTI_DEFINE_p = /DwxNO_RTTI
!endif
!if "$(USE_RTTI)" == "0"
__RTTI_DEFINE_p_1 = /d wxNO_RTTI
!endif
!if "$(USE_THREADS)" == "0"
__THREAD_DEFINE_p = /DwxNO_THREADS
!endif
!if "$(USE_THREADS)" == "0"
__THREAD_DEFINE_p_1 = /d wxNO_THREADS
!endif
!if "$(UNICODE)" == "1"
__UNICODE_DEFINE_p = /D_UNICODE
```

```
endif
!if "$(UNICODE)" == "1"
__UNICODE_DEFINE_p_1 = /d _UNICODE
endif
!if "$(MSLU)" == "1"
__MSLU_DEFINE_p = /DwxUSE_UNICODE_MSLU=1
endif
!if "$(MSLU)" == "1"
__MSLU_DEFINE_p_1 = /d wxUSE_UNICODE_MSLU=1
endif
!if "$(SHARED)" == "1"
__DLLFLAG_p = /DWXUSINGDLL
endif
!if "$(SHARED)" == "1"
__DLLFLAG_p_1 = /d WXUSINGDLL
endif
!if "$(MONOLITHIC)" == "0"
__WXLIB_CORE_p = \
    wx$(PORTNAME)$ (WXUNIVNAME)$ (WX_RELEASE_NODOT)$ (WXUNICODEFLAG)$
(WXDEBUGFLAG)$ (WX_LIB_FLAVOUR)_core.lib
endif
!if "$(MONOLITHIC)" == "0"
__WXLIB_BASE_p = \
    wxbase$(WX_RELEASE_NODOT)$ (WXUNICODEFLAG)$ (WXDEBUGFLAG)$
(WX_LIB_FLAVOUR).lib
endif
!if "$(MONOLITHIC)" == "1"
__WXLIB_MONO_p = \
    wx$(PORTNAME)$ (WXUNIVNAME)$ (WX_RELEASE_NODOT)$ (WXUNICODEFLAG)$
(WXDEBUGFLAG)$ (WX_LIB_FLAVOUR).lib
endif
!if "$(USE_GUI)" == "1"
__LIB_TIFF_p = wxtiff$(WXDEBUGFLAG).lib
endif
!if "$(USE_GUI)" == "1"
__LIB_JPEG_p = wxjpeg$(WXDEBUGFLAG).lib
endif
!if "$(USE_GUI)" == "1"
__LIB_PNG_p = wxpng$(WXDEBUGFLAG).lib
endif
!if "$(MSLU)" == "1"
__UNICOWS_LIB_p = unicows.lib
endif
```

```

all: $(WXPLAT)
$(WXPLAT):
    -if not exist $(WXPLAT) mkdir $(WXPLAT)

### Targets: ###

all: $(PROGRAM)

clean:
    -if exist $(WXPLAT)\*.obj del $(WXPLAT)\*.obj
    -if exist $(WXPLAT)\*.res del $(WXPLAT)\*.res
    -if exist $(WXPLAT)\*.pch del $(WXPLAT)\*.pch
        -if exist $(WXPLAT)\$(PROGRAM_NAME).exe del $(WXPLAT)\$(
(PROGRAM_NAME).exe
    -if exist $(WXPLAT)\$(PROGRAM_NAME).ilk del $(WXPLAT)\$(PROGRAM_NAME).ilk
        -if exist $(WXPLAT)\$(PROGRAM_NAME).pdb del $(WXPLAT)\$(
(PROGRAM_NAME).pdb

$(PROGRAM): $(OBJECTS)
    link /NOLOGO /OUT:$@ $(LD_FLAGS) $(__DEBUGINFO_1) $(LINK_TARGET_CPU)
/LIBPATH:$(LIBDIRNAME) /SUBSYSTEM:WINDOWS @<<
    $(OBJECTS) $(__WXLIB_CORE_p) $(__WXLIB_BASE_p) $(__WXLIB_MONO_p) $
(__LIB_TIFF_p) $(__LIB_JPEG_p) $(__LIB_PNG_p) wxzlib$(WXDEBUGFLAG).lib wxregex$
(W
XUNICODEFLAG)$$(WXDEBUGFLAG).lib wxexpat$(WXDEBUGFLAG).lib $
(EXTRALIBS_FOR_BASE) $(__UNICOWS_LIB_p) kernel32.lib user32.lib gdi32.lib
comdlg32.lib winspool.lib
winmm.lib shell32.lib comctl32.lib ole32.lib oleaut32.lib uuid.lib rpcrt4.lib advapi32.lib
wsock32.lib odbc32.lib
<<
    -if exist src\*.obj del src\*.obj
    -if exist src\*.res del src\*.res

# objs
src\hello.obj: src\hello.cpp
    $(CXX) /c /nologo /TP /Fo$@ $(APP_CXXFLAGS) $**

```

Penjelasan makefile:

- Makefile ini dapat dimodifikasi sesuai keperluan.
- Makefile memiliki dua tugas:
  - all, yang akan melakukan kompilasi

- clean, yang akan menghapus semua file object dan executable
- Ubahlah variabel pada bagian variables apabila diperlukan.

Untuk melakukan kompilasi, berikanlah perintah berikut ini:

```
> nmake -f makefile.vc
```

Microsoft (R) Program Maintenance Utility Version 6.00.8168.0

Copyright (C) Microsoft Corp 1988-1998. All rights reserved.

```
cl /c /nologo /TP /Fosrc\hello.obj /MD /DWIN32 /Fdbuild\msw\hello.pdb
/O2 /D__WXMSW__ /D_UNICODE /Ic:\wx\lib\vc_lib\mswu /I"C:\Program Fil
es\Microsoft Visual Studio\VC98\Include" /Ic:\wx\include\msvc /Ic:\wx\include
\wx /Ic:\wx\include /W4 /I. /D_WINDOWS /DNOPCH /GR /EHsc src\hello.cpp
hello.cpp
link /NOLOGO /OUT:build\msw\hello.exe /LIBPATH:c:\wx\lib\vc_lib /SUB
SYSTEM:WINDOWS @C:\DOCUME~1\nop\LOCALS~1\Temp\nma00480.
LINK : warning LNK4089: all references to "ADVAPI32.dll" discarded by /OPT:REF
LINK : warning LNK4089: all references to "OLEAUT32.dll" discarded by /OPT:REF
if exist src\*.obj del src\*.obj
if exist src\*.res del src\*.res
```

Setelah kompilasi dilakukan, akan terbentuk sebuah executable dengan nama hello.exe di dalam direktori build/msw/.

Apabila penghapusan file-file hasil kompilasi ingin dilakukan (cleaning), berikanlah argumen clean. Sebagai contoh:

```
>nmake -f makefile.vc clean
```

Microsoft (R) Program Maintenance Utility Version 6.00.8168.0

Copyright (C) Microsoft Corp 1988-1998. All rights reserved.

```
if exist build\msw\*.obj del build\msw\*.obj
if exist build\msw\*.res del build\msw\*.res
if exist build\msw\*.pch del build\msw\*.pch
if exist build\msw\hello.exe del build\msw\hello.exe
if exist build\msw\hello.ilc del build\msw\hello.ilc
if exist build\msw\hello.pdb del build\msw\hello.pdb
```

Setelah perintah tersebut diberikan, semua jejak hasil kompilasi akan dihapus.

## 2.2.5 Melihat kebutuhan pustaka

### 2.2.5.1 Kebutuhan pustaka di Linux

Untuk melihat pustaka apa saja yang dibutuhkan oleh aplikasi hello, masuklah ke direktori build/gtk/ dan berikanlah perintah berikut ini:

```
$ ldd hello
        libwx_gtk2u_xrc-2.6.so.0 => /home/nop/wxgtk/lib/libwx_gtk2u_xrc-2.6.so.0
(0x40018000)
        libwx_gtk2u_qa-2.6.so.0 => /home/nop/wxgtk/lib/libwx_gtk2u_qa-2.6.so.0
(0x4009d000)
        libwx_gtk2u_html-2.6.so.0 => /home/nop/wxgtk/lib/libwx_gtk2u_html-2.6.so.0
(0x400bc000)
        libwx_gtk2u_adv-2.6.so.0 => /home/nop/wxgtk/lib/libwx_gtk2u_adv-2.6.so.0
(0x4014e000)
        libwx_gtk2u_core-2.6.so.0 => /home/nop/wxgtk/lib/libwx_gtk2u_core-2.6.so.0
(0x401f3000)
        libwx_baseu_xml-2.6.so.0 => /home/nop/wxgtk/lib/libwx_baseu_xml-2.6.so.0
(0x40560000)
        libwx_baseu_net-2.6.so.0 => /home/nop/wxgtk/lib/libwx_baseu_net-2.6.so.0
(0x40569000)
        libwx_baseu-2.6.so.0 => /home/nop/wxgtk/lib/libwx_baseu-2.6.so.0 (0x40597000)
        libstdc++.so.5 => /usr/lib/libstdc++.so.5 (0x40734000)
        libm.so.6 => /lib/tls/libm.so.6 (0x407ee000)
        libgcc_s.so.1 => /lib/libgcc_s.so.1 (0x40810000)
        libpthread.so.0 => /lib/tls/libpthread.so.0 (0x40819000)
        libc.so.6 => /lib/tls/libc.so.6 (0x40828000)
        libz.so.1 => /usr/lib/libz.so.1 (0x4095e000)
        libdl.so.2 => /lib/tls/libdl.so.2 (0x40970000)
        libgtk-x11-2.0.so.0 => /usr/lib/libgtk-x11-2.0.so.0 (0x40973000)
        libgdk-x11-2.0.so.0 => /usr/lib/libgdk-x11-2.0.so.0 (0x40c51000)
        libatk-1.0.so.0 => /usr/lib/libatk-1.0.so.0 (0x40cd0000)
        libgdk_pixbuf-2.0.so.0 => /usr/lib/libgdk_pixbuf-2.0.so.0 (0x40cee000)
        libpangoft-1.0.so.0 => /usr/lib/libpangoft-1.0.so.0 (0x40d05000)
        libpangox-1.0.so.0 => /usr/lib/libpangox-1.0.so.0 (0x40d0c000)
        libpango-1.0.so.0 => /usr/lib/libpango-1.0.so.0 (0x40d17000)
        libgobject-2.0.so.0 => /usr/lib/libgobject-2.0.so.0 (0x40d50000)
```

```

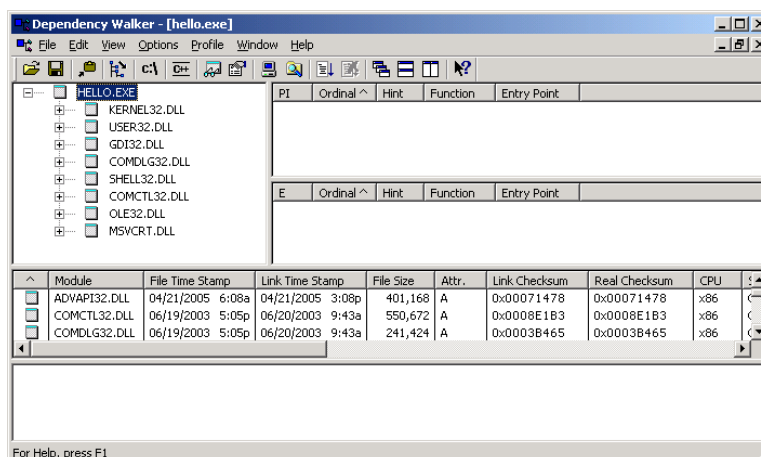
libgmodule-2.0.so.0 => /usr/lib/libgmodule-2.0.so.0 (0x40d84000)
libgthread-2.0.so.0 => /usr/lib/libgthread-2.0.so.0 (0x40d88000)
libglib-2.0.so.0 => /usr/lib/libglib-2.0.so.0 (0x40d8e000)
libpng12.so.0 => /usr/lib/libpng12.so.0 (0x40e0e000)
libjpeg.so.62 => /usr/lib/libjpeg.so.62 (0x40e33000)
libtiff.so.4 => /usr/lib/libtiff.so.4 (0x40e51000)
libexpat.so.1 => /usr/lib/libexpat.so.1 (0x40ea2000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
libX11.so.6 => /usr/X11R6/lib/libX11.so.6 (0x40ec2000)
libXrandr.so.2 => /usr/X11R6/lib/libXrandr.so.2 (0x40f89000)
libXi.so.6 => /usr/X11R6/lib/libXi.so.6 (0x40f8d000)
libXext.so.6 => /usr/X11R6/lib/libXext.so.6 (0x40f95000)
libXft.so.2 => /usr/lib/libXft.so.2 (0x40fa4000)
libfreetype.so.6 => /usr/lib/libfreetype.so.6 (0x40fb7000)
libfontconfig.so.1 => /usr/lib/libfontconfig.so.1 (0x41024000)
libXcursor.so.1 => /usr/lib/libXcursor.so.1 (0x41053000)
libXrender.so.1 => /usr/lib/libXrender.so.1 (0x4105c000)
libpangoff2-1.0.so.0 => /usr/lib/libpangoff2-1.0.so.0 (0x41065000)

```

Keluaran dari perintah sebelumnya mungkin bisa berbeda di setiap komputer. Ketika mendistribusikan aplikasi dalam bentuk binary, pastikanlah agar sistem tujuan memiliki semua pustaka yang dibutuhkan.

### 2.2.5.2 Kebutuhan pustaka di Windows

Untuk melihat pustaka apa saja yang dibutuhkan oleh aplikasi hello.exe, penulis menggunakan program Dependency Walker, yang dapat didownload di <http://www.dependencywalker.com/>.



## 2.3 Hello world sederhana 2

Aplikasi hello world yang dibangun akan memiliki fitur-fitur berikut:

- Memiliki sebuah window utama yang akan memiliki title Hello World!
- User dapat melakukan klik pada tombol close pada title bar untuk menterminasi aplikasi

Versi kedua dari hello world sederhana ini adalah versi dimana developer memisahkan header dan implementation. Cara penulisan source code seperti ini sangat disarankan untuk digunakan dalam pembangunan aplikasi (yang berpotensi) besar. Selanjutnya, di dalam buku ini, cara seperti ini yang akan digunakan.

### 2.3.1 Struktur direktori source code:

```
hello2/  
  build/  
    gtk/  
      hello  
    msw/  
      hello.exe  
  src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
  makefile.gtk  
  makefile.vc
```

### 2.3.2 Output

Ketika build/gtk/hello dijalankan:

- Sebuah frame kosong akan ditampilkan
- Frame tersebut akan memiliki title bar yang bertuliskan Hello World!
- Apabila tombol close pada title bar diklik, maka aplikasi akan diterminasi



### 2.3.3 Source code

Dalam contoh ini, dibandingkan dengan hello world sederhana 1, sebuah file hello.cpp digantikan dengan empat buah file:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

Untuk penjelasan source code, lihatlah juga pada bagian penjelasan hello world sederhana 1.

#### 2.3.3.1 src/app.h

Berikut ini adalah source code src/app.h:

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

#### 2.3.3.2 src/app.cpp

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
```

```
{  
    Frame * frame = new Frame ( _ ("Hello World!") );  
  
    frame -> Show ( true );  
  
    return true;  
}
```

Penjelasan source code:

- Deklarasi class App disimpan pada file app.h. Di dalam file app.cpp ini, developer perlu menginclude file app.h.
- Karena terdapat pembuatan frame (class Frame), maka developer perlu menginclude file frame.h

### **2.3.3.3 src/frame.h**

```
#ifndef _FRAME_H_  
#define _FRAME_H_  
  
class Frame: public wxFrame  
{  
public:  
  
    Frame ( const wxString& title );  
  
};  
  
#endif // _FRAME_H_
```

### **2.3.3.4 src/frame.cpp**

```
#include "wx/wx.h"  
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )  
{  
  
}
```

Penjelasan source code:

- Deklarasi class Frame berada di dalam file frame.h. Developer perlu menginclude file frame.h ini.

## 2.3.4 Kompilasi

### 2.3.4.1 Kompilasi di Linux (GTK+)

Kompilasi di Linux memanfaatkan penggunaan makefile. Dalam hal ini, nama file makefilenya adalah makefile.gtk. Berikut ini adalah isi makefile.gtk:

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006

# variables
WXCONFIG = wx-config
WXPLAT = gtk

PROGRAM = hello
OBJECTS = src/app.o src/frame.o

# system
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES: .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all:  $(PROGRAM)

$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o
```

clean:

```
rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

Penjelasan Makefile:

- Lihatlah juga pembahasan makefile pada hello world sederhana 1
- Perbedaan dengan makefile yang dibahas pada hello world sederhana 1 adalah penambahan file object. Karena source code yang terlibat menjadi lebih banyak, maka secara eksplisit perlu ditambahkan. Dalam contoh ini, file object menjadi dua, yaitu app.o (dari app.h dan app.cpp) dan frame.o (dari frame.h dan frame.cpp).

Untuk melakukan kompilasi, berikanlah perintah berikut ini:

```
$ make -f makefile.gtk
g++ -c `wx-config --cxxflags` -o src/app.o src/app.cpp
g++ -c `wx-config --cxxflags` -o src/frame.o src/frame.cpp
g++ -o hello src/app.o src/frame.o `wx-config --libs`
mkdir -p build/gtk
mv hello build/gtk
rm -f src/*.o
```

Setelah kompilasi dilakukan, akan terbentuk sebuah executable dengan nama hello di dalam direktori build/gtk/. Perlu diperhatikan, perbedaan dengan keluaran proses kompilasi hello world sederhana 1 adalah informasi bertambahnya file yang dikompilasi.

Apabila penghapusan file-file hasil kompilasi ingin dilakukan (cleaning), berikanlah argumen clean. Sebagai contoh:

```
$ make -f makefile.gtk clean
rm -f src/*.o build/gtk/hello
```

Setelah perintah tersebut diberikan, semua jejak hasil kompilasi akan dihapus.

### **2.3.4.2 Kompilasi di Windows**

Kompilasi dalam buku ini memanfaatkan penggunaan makefile. Dalam hal ini, nama file makefilenya adalah makefile.vc. Sebagai catatan, karena terlalu panjang, maka isi file makefile.vc tidak ditampilkan di dalam sub bab ini. Lihatlah contoh sebelumnya, atau lihatlah pada CDROM penyerta buku ini.

### 2.3.5 Melihat kebutuhan pustaka

Tidak ada perbedaan kebutuhan pustaka antara hello world sederhana 1 dan hello world sederhana 2. Silahkan merujuk kepada penjelasan kebutuhan pustaka untuk hello world sederhana 1.

## 2.4 Hello world sederhana 3

Aplikasi hello world yang dibangun akan memiliki fitur-fitur berikut:

- Memiliki sebuah window utama yang akan memiliki title Hello World!
- User dapat melakukan klik pada tombol close pada title bar untuk menterminasi aplikasi
- Sebuah tombol Hello akan ditambahkan pada frame. Button tersebut merupakan button satu-satunya, dan apabila frame diresize, ukuran button juga akan mengikuti. Saat ini, tidak ada aksi apapun yang akan dilakukan apabila user melakukan klik pada button tersebut. Tujuan hello world sederhana 3 ini adalah mendemonstrasikan bagaimana menambahkan widget pada frame.



Seperti yang telah disebutkan pada bagian sebelumnya, dalam penulisan source code, kita akan memisahkan file header dan implementation.

### 2.4.1 Struktur direktori source code:

```
hello3/  
  build/  
    gtk/  
      hello  
    msw/  
      hello.exe  
  src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
  makefile.gtk  
  makefile.vc
```

### 2.4.2 Output

Ketika build/gtk/hello dijalankan:

- Sebuah frame kosong akan ditampilkan
- Frame tersebut akan memiliki title bar yang bertuliskan Hello World!
- Apabila tombol close pada title bar diklik, maka aplikasi akan diterminasi
- Terdapat satu button Hello pada frame. Button tersebut tidak akan melakukan aksi apapun ketika diklik.

## 2.4.3 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

Untuk penjelasan source code, lihatlah juga pada bagian penjelasan hello world sederhana 1.

### 2.4.3.1 src/app.h

Berikut ini adalah source code src/app.h:

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

### 2.4.3.2 src/app.cpp

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
```

```
Frame * frame = new Frame ( _ ("Hello World!") );  
  
frame -> Show ( true );  
  
return true;  
}
```

Penjelasan source code:

- Deklarasi class App disimpan pada file app.h. Di dalam file app.cpp ini, developer perlu menginclude file app.h.
- Karena terdapat pembuatan frame (class Frame), maka developer perlu menginclude file frame.h

### **2.4.3.3 src/frame.h**

```
#ifndef _FRAME_H_  
#define _FRAME_H_  
  
class Frame: public wxFrame  
{  
public:  
  
    Frame ( const wxString& title );  
  
};  
  
#endif // _FRAME_H_
```

### **2.4.3.4 src/frame.cpp**

```
#include "wx/wx.h"  
#include "wx/button.h"  
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )  
{  
  
    wxButton * button1 = new wxButton ( this, wxID_ANY, _ ( "hello" ) );
```



```
}
```

Penjelasan source code:

- Deklarasi class Frame berada di dalam file frame.h. Developer perlu menginclude file frame.h ini.
- Sebuah wxButton dibuat. Sebuah header tambahan digunakan, yaitu wx/button.h.
- Pada saat pembuatan wxButton, developer setidaknya memberikan tiga parameter, yaitu:
  - orang tua. Dalam hal ini, orang tua wxButton adalah Frame. Dengan cara seperti ini, secara otomatis, wxButton yang dibuat akan ditampilkan di dalam Frame.
  - ID button. ID button sangat penting sebagai pengenal (terutama dalam merespon event). Saat ini, karena button belum digunakan (misal: tidak merespon klik tombol mouse), maka ID yang diberikan menjadi tidak penting. Sebuah wxID\_ANY diassign dan wxWidgets akan secara otomatis generate ID berupa bilangan negatif. WxWidgets menyediakan beberapa ID yang bisa digunakan. Selengkapnya mengenai ID dapat dibaca pada bagian akhir bab ini.
  - Title button.

## 2.4.4 Kompilasi

### 2.4.4.1 Kompilasi di Linux (GTK+)

Kompilasi di Linux memanfaatkan penggunaan makefile. Dalam hal ini, nama file makefilenya adalah makefile.gtk. Berikut ini adalah isi makefile.gtk:

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk
```

```
PROGRAM = hello
OBJECTS = src/app.o src/frame.o
```

```
# system
```

```
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES:    .o .cpp
```

```
.cpp.o :
```

```
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<
```

```
all:  $(PROGRAM)
```

```
$(PROGRAM):  $(OBJECTS)
```

```
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
```

```
    mkdir -p build/$(WXPLAT)
```

```
    mv $(PROGRAM) build/$(WXPLAT)
```

```
    rm -f src/*.o
```

```
clean:
```

```
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

Penjelasan Makefile:

- Lihatlah juga pembahasan makefile pada hello world sederhana 2

Untuk melakukan kompilasi, berikanlah perintah berikut ini:

```
$ make -f makefile.gtk
```

```
g++ -c `wx-config --cxxflags` -o src/app.o src/app.cpp
```

```
g++ -c `wx-config --cxxflags` -o src/frame.o src/frame.cpp
```

```
g++ -o hello src/app.o src/frame.o `wx-config --libs`
```

```
mkdir -p build/gtk
```

```
mv hello build/gtk
```

```
rm -f src/*.o
```

Apabila penghapusan file-file hasil kompilasi ingin dilakukan (cleaning), berikanlah argumen clean. Sebagai contoh:

```
$ make -f makefile.gtk clean
```

```
rm -f src/*.o build/gtk/hello
```

Setelah perintah tersebut diberikan, semua jejak hasil kompilasi akan dihapus.

#### **2.4.4.2 Kompilasi di Window**

Kompilasi dalam buku ini memanfaatkan penggunaan makefile. Dalam hal ini, nama file makefilenya adalah makefile.vc. Sebagai catatan, karena terlalu panjang, maka isi file makefile.vc tidak ditampilkan di dalam sub bab ini. Lihatlah contoh sebelumnya, atau lihatlah pada CDROM penyerta buku ini.

#### **2.4.5 Melihat kebutuhan pustaka**

Tidak ada perbedaan kebutuhan pustaka antara hello world sederhana 3 dengan hello world sederhana 1 dan hello world sederhana 2. Silahkan merujuk kepada penjelasan kebutuhan pustaka untuk hello world sederhana 1.

### **2.5 Hello world sederhana 4**

Aplikasi hello world yang dibangun akan memiliki fitur-fitur berikut:

- Memiliki sebuah window utama yang akan memiliki title Hello World!
- User dapat melakukan klik pada tombol close pada title bar untuk menterminasi aplikasi
- Sebuah menu File akan ditambahkan. Menu file tersebut berisikan sebuah anak menu, yaitu Quit (dengan shortcut CTRL-Q; dengan hint Quit application). Namun, apabila menu Quit diklik, belum ada aksi apapun yang akan dilakukan.
- Sebuah statusbar akan dibuat. Pertama kali aplikasi dijalankan, statusbar akan menampilkan pesan "Welcome to Hello World!". Pada saat mouse melewati menu File | Quit, statusbar akan menampilkan hint dari menu File | Quit tersebut, yaitu "Quit application".



Seperti yang telah disebutkan pada bagian sebelumnya, dalam penulisan source code, kita akan memisahkan file header dan implementation.

### 2.5.1 Struktur direktori source code:

```
hello4/  
  build/  
    gtk/  
      hello  
    msw/  
      hello.exe  
  src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
  makefile.gtk  
  makefile.vc
```

### 2.5.2 Output

Ketika build/gtk/hello dijalankan:

- Sebuah frame kosong akan ditampilkan
- Frame tersebut akan memiliki title bar yang bertuliskan Hello World!
- Apabila tombol close pada title bar diklik, maka aplikasi akan diterminasi

- Terdapat statusbar pada frame
- Terdapat menu File pada frame. Menu File tersebut memiliki anak menu Quit.

### 2.5.3 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

Untuk penjelasan source code, lihatlah juga pada bagian penjelasan hello world sederhana 1.

#### 2.5.3.1 src/app.h

Berikut ini adalah source code src/app.h:

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

#### 2.5.3.2 src/app.cpp

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )
```

```
bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Hello World!") );

    frame -> Show ( true );

    return true;
}
```

Penjelasan source code:

- Deklarasi class App disimpan pada file app.h. Di dalam file app.cpp ini, developer perlu menginclude file app.h.
- Karena terdapat pembuatan frame (class Frame), maka developer perlu menginclude file frame.h

### **2.5.3.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

### **2.5.3.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "wx/menu.h"
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{

    CreateStatusBar ();

    SetStatusText ( _( "Welcome to Hello World!" ), 0 );

    wxMenu * fileMenu = new wxMenu;

    fileMenu -> Append ( wxID_ANY, _( "&Quit \tCTRL-Q" ), _( "Quit application" ) );

    wxMenuBar * menuBar = new wxMenuBar;

    menuBar -> Append ( fileMenu, _( "&File" ) );

    SetMenuBar ( menuBar );

}
```

Penjelasan source code:

- Deklarasi class Frame berada di dalam file frame.h. Developer perlu menginclude file frame.h ini.
- WxMenu dan wxMenuBar digunakan. Sebuah header tambahan wx/menu.h diinclude.
- Sebuah frame dapat mengandung menubar, toolbar dan statusbar. Sebuah frame juga dapat mengandung window yang bukan frame ataupun dialog. Contoh hello world sederhana 3 mendemonstrasikan frame yang mengandung window lain, yaitu wxButton.
- Pembuatan statusbar dapat dilakukan dengan memanggil CreateStatusBar (). Pengaturan teks statusbar dapat dilakukan dengan memanggil SetStatusText ():

```
virtual void SetStatusText(const wxString& text, int number = 0)
```

```
SetStatusText ( _( "Welcome to Hello World!" ), 0 );
```

Parameter pertama adalah teks yang ingin diberikan. Parameter kedua adalah nomor bagian pada statusbar. Statusbar bisa dibagi menjadi bagian-bagian. Pada statusbar di hello world sederhana 4, hanya ada satu bagian yang dibuat.

- Pembuatan menu dapat dilakukan dengan:
  - Membuat menu-menu (wxMenu) tunggal terlebih dahulu. Dalam hal ini, menu yang dimaksudkan sebagai menu File dibuat terlebih dahulu:

```
wxMenu * fileMenu = new wxMenu;
```

- Setelah menu dibuat, developer pun akan menambahkan anak menu ke dalamnya. Dalam contoh ini, anak menu Quit ditambahkan. Penambahan dengan Append () pada contoh ini membutuhkan tiga parameter:
  - ID. Saat ini, karena belum digunakan, wxID\_ANY digunakan.
  - Title menu. Gunakan karakter & untuk menampilkan underline untuk huruf disamping kanan &. Shortcut juga langsung didefinisikan. Dalam hal ini dengan memberikan spasi dan escape \t dan diikuti oleh shortcut yang diinginkan. Terakhir, diberikan juga sebuah Help String, yang akan ditampilkan secara otomatis pada StatusBar (apabila tersedia).

```
wxMenuItem* Append(int id, const wxString& item, const wxString& helpString = "",  
wxItemKind kind = wxITEM_NORMAL)
```

```
fileMenu -> Append ( wxID_ANY, _( "&Quit \tCTRL-Q" ), _( "Quit application" ) );
```

- Setelah menu-menu tunggal selesai dibuat, developer akan membuat sebuah wxMenuBar, yang akan 'menampung' berbagai menu-menu tunggal tersebut.

```
wxMenuBar * menuBar = new wxMenuBar;
```

- Setelah wxMenuBar dibuat, maka developer pun menambahkan menu File yang telah dibuat sebelumnya (ulangi untuk semua menu yang ingin ditambahkan, apabila ada)

```
bool Append(wxMenu *menu, const wxString& title)
```

```
menuBar -> Append ( fileMenu, _( "&File" ) );
```

- Kemudian, developer meminta kepada Frame untuk mengatur menu bar:

```
SetMenuBar ( menuBar );
```

## 2.5.4 Kompilasi

### 2.5.4.1 Kompilasi di Linux (GTK+)



Kompilasi di Linux memanfaatkan penggunaan makefile. Dalam hal ini, nama file makefilenya adalah makefile.gtk. Berikut ini adalah isi makefile.gtk:

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006

# variables
WXCONFIG = wx-config
WXPLAT = gtk

PROGRAM = hello
OBJECTS = src/app.o src/frame.o

# system
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES:    .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all:  $(PROGRAM)

$(PROGRAM):  $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

Penjelasan Makefile:

- Lihatlah juga pembahasan makefile pada hello world sederhana 2

Untuk melakukan kompilasi, berikanlah perintah berikut ini:

```
$ make -f makefile.gtk
g++ -c `wx-config --cxxflags` -o src/app.o src/app.cpp
g++ -c `wx-config --cxxflags` -o src/frame.o src/frame.cpp
```

```
g++ -o hello src/app.o src/frame.o `wx-config --libs`  
mkdir -p build/gtk  
mv hello build/gtk  
rm -f src/*.o
```

Apabila penghapusan file-file hasil kompilasi ingin dilakukan (cleaning), berikanlah argumen clean. Sebagai contoh:

```
$ make -f makefile.gtk clean  
rm -f src/*.o build/gtk/hello
```

Setelah perintah tersebut diberikan, semua jejak hasil kompilasi akan dihapus.

#### **2.5.4.2 Kompilasi di Windows**

Kompilasi dalam buku ini memanfaatkan penggunaan makefile. Dalam hal ini, nama file makefilenya adalah makefile.vc. Sebagai catatan, karena terlalu panjang, maka isi file makefile.vc tidak ditampilkan di dalam sub bab ini. Lihatlah contoh sebelumnya, atau lihatlah pada CDROM penyerta buku ini.

#### **2.5.5 Melihat kebutuhan pustaka**

Tidak ada perbedaan kebutuhan pustaka antara hello world sederhana 4 dengan hello world sederhana 1, hello world sederhana 2 dan hello world sederhana 3. Silahkan merujuk kepada penjelasan kebutuhan pustaka untuk hello world sederhana 1.

### **2.6 Window ID yang disediakan oleh wxWidgets**

Window ID adalah integer dan digunakan untuk menandai secara unik setiap window di dalam konteks tertentu. Hal ini diperlukan terutama ketika bekerja dengan event.

Di dalam beberapa contoh sebelumnya, `wxID_ANY` digunakan, terutama ketika developer tidak mementingkan ID unik, misal karena tidak ingin memproses event. Ketika `wxID_ANY` diberikan, wxWidgets akan menggenerate bilangan negatif secara otomatis.

Selain `wxID_ANY`, wxWidgets juga menyediakan beberapa ID standar yang dapat digunakan. Semua ID tersebut bernilai positif. User dapat mendefinisikan ID sendiri, dimana nilainya diusahakan untuk tidak konflik dengan ID standar yang telah disediakan, serta harus selalu merupakan integer positif.

Developer disarankan untuk menggunakan ID yang telah disediakan sebisa mungkin. Pada beberapa sistem, menggunakan ID yang bersesuaian dapat mengakibatkan ditampilkannya grafik yang bersesuaian pula (misal `wxID_OK` akan menampilkan grafik yang bersesuaian dengan OK). Komponen seperti `wxTextCtrl` juga dapat menangani menu item atau button yang diberikan ID seperti `wxID_COPY`, `wxID_PASTE` dan `wxID_UNDO`.

Berikut ini adalah ID standar yang disediakan:

<i><b>ID</b></i>	<i><b>Nilai</b></i>	<i><b>Deskripsi</b></i>
<code>wxID_ANY</code>	-1	
<code>wxID_LOWEST</code>	4999	Nilai terendah
<code>wxID_OPEN</code>	5000	File open
<code>wxID_CLOSE</code>	5001	Window close
<code>wxID_NEW</code>	5002	New window, file atau dokumen
<code>wxID_SAVE</code>	5003	File save
<code>wxID_SAVEAS</code>	5004	File save as
<code>wxID_REVERT</code>	5005	revert
<code>wxID_EXIT</code>	5006	Keluar dari aplikasi
<code>wxID_UNDO</code>	5007	undo
<code>wxID_REDO</code>	5008	redo
<code>wxID_HELP</code>	5009	Help secara umum
<code>wxID_PRINT</code>	5010	print
<code>wxID_PRINT_SETUP</code>	5011	Print setup
<code>wxID_PREVIEW</code>	5012	Print preview
<code>wxID_ABOUT</code>	5013	Dialog about
<code>wxID_HELP_CONTENTS</code>	5014	Isi help

<i><b>ID</b></i>	<i><b>Nilai</b></i>	<i><b>Deskripsi</b></i>
wxID_HELP_COMMANDS	5015	Perintah aplikasi
wxID_HELP_PROCEDURES	5016	Prosedur aplikasi
wxID_HELP_CONTEXT	5017	
wxID_CLOSE_ALL	5018	Tutup semua
wxID_PREFERENCES	5019	preferensi
wxID_CUT	5030	cut
wxID_COPY	5031	copy
wxID_PASTE	5032	paste
wxID_CLEAR	5033	clear
wxID_FIND	5034	find
wxID_DUPLICATE	5035	duplikasi
wxID_SELECTALL	5036	Memilih semua
wxID_DELETE	5037	delete
wxID_REPLACE	5038	replace
wxID_REPLACE_ALL	5039	Replace all
wxID_PROPERTIES	5040	Menampilkan properti
wxID_VIEW_DETAILS	5041	View secara detil
wxID_VIEW_LARGEICONS	5042	View icon besar
wxID_VIEW_SMALLICONS	5043	View icon kecil
wxID_VIEW_LIST	5044	View dalam bentuk list
wxID_VIEW_SORTDATE	5045	Urutkan berdasarkan waktu
wxID_VIEW_SORTNAME	5046	Urutkan berdasarkan nama
wxID_VIEW_SORTSIZE	5047	Urutkan berdasarkan ukuran
wxID_VIEW_SORTTYPE	5048	Urutkan berdasarkan tipe
wxID_FILE1	5050	Recent file
wxID_FILE2	5051	Recent file
wxID_FILE3	5052	Recent file
wxID_FILE4	5053	Recent file
wxID_FILE5	5054	Recent file
wxID_FILE6	5055	Recent file
wxID_FILE7	5056	Recent file

<b>ID</b>	<b>Nilai</b>	<b>Deskripsi</b>
wxID_FILE8	5057	Recent file
wxID_FILE9	5058	Recent file
wxID_OK	5100	OK
wxID_CANCEL	5101	cancel
wxID_APPLY	5102	apply
wxID_YES	5103	yes
wxID_NO	5104	no
wxID_STATIC	5105	Identifier untuk static text atau bitmap
wxID_FORWARD	5106	Forward (navigasi)
wxID_BACKWARD	5107	backward (navigasi)
wxID_DEFAULT	5108	Default setting
wxID_MORE	5109	more
wxID_SETUP	5110	Setup dialog
wxID_RESET	5111	Reset setting
wxID_CONTEXT_HELP	5112	Context-sensitive help
wxID_YESTOALL	5113	Yes to all
wxID_NOTOALL	5114	No to all
wxID_ABORT	5115	abort
wxID_RETRY	5116	retry
wxID_IGNORE	5117	ignore
wxID_ADD	5118	add
wxID_REMOVE	5119	remove
wxID_UP	5120	Up (navigasi)
wxID_DOWN	5121	Down (navigasi)
wxID_HOME	5122	Home (navigasi)
wxID_REFRESH	5123	refresh
wxID_STOP	5124	Stop operasi
wxID_INDEX	5125	Menampilkan index
wxID_BOLD	5126	Bold
wxID_ITALIC	5127	italic
wxID_JUSTIFY_CENTER	5128	center

<b>ID</b>	<b>Nilai</b>	<b>Deskripsi</b>
wxID_JUSTIFY_FILL	5129	format
wxID_JUSTIFY_RIGHT	5130	Rata kanan
wxID_JUSTIFY_LEFT	5131	Rata kiri
wxID_UNDERLINE	5132	underline
wxID_INDENT	5133	indent
wxID_UNINDENT	5134	unindent
wxID_ZOOM_100	5135	Zoom 100%
wxID_ZOOM_FIT	5136	Fit to page
wxID_ZOOM_IN	5137	Zoom in
wxID_ZOOM_OUT	5138	Zoom out
wxID_UNDELETE	5139	undelelete
wxID_REVERT_TO_SAVED	5140	Revert to saved state
wxID_SYSTEM_MENU	5200	System menu (digunakan oleh wxUniv)
wxID_CLOSE_FRAME	5201	System menu (digunakan oleh wxUniv)
wxID_MOVE_FRAME	5202	System menu (digunakan oleh wxUniv)
wxID_RESIZE_FRAME	5203	System menu (digunakan oleh wxUniv)
wxID_MAXIMIZE_FRAME	5204	System menu (digunakan oleh wxUniv)
wxID_ICONIZE_FRAME	5205	System menu (digunakan oleh wxUniv)
wxID_RESTORE_FRAME	5206	System menu (digunakan oleh wxUniv)
wxID_FILEDLGG	5900	Generic file dialog
wxID_HIGHEST	5999	ID terbesar

## **2.7 Mengatur Window ID sendiri**

Apabila window ID standar yang disediakan oleh wxWidgets tidak sesuai dengan yang dikehendaki, developer selalu bisa membuat ID sendiri. Aturan berikut ini disarankan:

- Gunakan integer positif
- Gunakan enum, bukan dengan #define atau static const int
- Definisikan di dalam scope global, bukan di dalam class
- Gunakan huruf besar (semua) untuk ID
- Gunakan underscore untuk memisahkan kata

- Gunakan nilai lebih besar sama dengan `wxID_HIGHEST + n`, dimana `wxID_HIGHEST + n` adalah nilai terendah ID yang ingin digunakan.

Sebagai contoh:

```
enum
{
    ID_WORDWRAP = wxID_HIGHEST + 100,

    ID_USE_RICH_FORMAT
};
```

## Bab 3. Event Driven Programming

Bab ini membahas tentang event driven programming di wxWidgets. Setelah membaca bab ini, developer diharapkan mampu bekerja dengan sistem event di wxWidgets, yang dibutuhkan untuk membangun aplikasi yang berfungsi penuh.

### 3.1 Sekilas tentang event driven programming

Semua aplikasi GUI adalah event-driven (dikendalikan oleh event). Hal ini berarti, aplikasi berjalan dan menunggu event yang dibangkitkan oleh user ataupun sumber lain seperti sistem operasi. Contoh event adalah tombol mouse di klik pada tombol tertentu, window diresize, menu diklik dan lain sebagainya. Untuk menangani event yang terjadi, developer kemudian membangun event handler.

Sistem event ini mirip dengan kehidupan sehari-hari. Contoh event adalah pintu diketuk. Apabila handler untuk event pintu diketuk tersebut telah didefinisikan, maka contoh aksi yang mungkin terjadi adalah pintu kemudian dibuka, dan penghuni di dalam ruangan mungkin akan memperbolehkan si pengetuk pintu untuk masuk.

Setiap class wxWidgets yang diturunkan dari class wxEvtHandler dapat mengandung event table untuk memberitahu wxWidgets bagaimana event dipertemukan dengan event handler. Class-class apa saja yang diturunkan dari class wxEvtHandler? Pada prinsipnya, semua class window (diturunkan dari wxWindow) dan class aplikasi, merupakan class yang diturunkan dari wxEvtHandler. Berikut ini adalah beberapa contohnya:

- frame
- tombol
- menu
- dokumen
- dan lain sebagainya. Untuk lebih lengkapnya, silahkan merujuk kepada dokumentasi wxWidgets.

Di dunia wxWidgets, terdapat dua cara untuk menangani event yang terjadi:

- Cara statis menggunakan event table. Cara ini relatif lebih mudah dan umum untuk digunakan. Dibuat pada saat compile time.
- Cara dinamis. Cara ini lebih fleksibel. Dibuat pada saat runtime.

Kedua cara tersebut akan dibahas dalam buku ini.



### 3.2 Event dan Event handler

Untuk diperhatikan, terdapat bermacam-macam event yang dikenal di wxWidgets. Event untuk tombol diklik dan event ketika window diresize tentunya berbeda-beda. Berikut ini adalah event-event yang dikenal di wxWidgets.

<i><b>Class event</b></i>	<i><b>Deskripsi</b></i>
wxActivateEvent	Aktifasi window atau aplikasi
wxCalendarEvent	Digunakan dengan wxCalendarCtrl
wxCalculateLayoutEvent	Kalkulasi layout window
wxCloseEvent	Close window atau end session
wxCommandEvent	Event dari berbagai kontrol standar
wxContextMenuEvent	Event untuk context menu
wxDateEvent	Digunakan dengan wxDatePickerCtrl
wxDialUpEvent	Event yang dikirim oleh wxDialUpManager
wxDropFilesEvent	Drop file
wxEraseEvent	Background erase
wxEvent	Base class untuk event
wxFindDialogEvent	Event yang dikirim oleh wxFindReplaceDialog
wxFocusEvent	Window focus
wxKeyEvent	Penekanan tombol (keypress)
WxIconizeEvent	Event iconize/restore
wxIdleEvent	Idle event
wxInitDialogEvent	Inisialisasi dialog
WxJoystickEvent	Event joystick
wxListEvent	Event list control
wxMaximizeEvent	Event maximize
wxMenuEvent	Event untuk menu
wxMouseCaptureChangedEvent	Perubahan mouse capture
wxMouseEvent	Event mouse
wxMoveEvent	Event move
WxNotebookEvent	Event untuk notebook control

<b><i>Class event</i></b>	<b><i>Deskripsi</i></b>
wxNotifyEvent	Event notifikasi
wxPaintEvent	Event paint
wxProcessEvent	Process ending
wxQueryLayoutInfoEvent	Query informasi layout
wxScrollEvent	Scroll dari scrollbar, slider dan spin button
WxScrollWinEvent	Event dari scrolled window
wxSizeEvent	Event size
wxSocketEvent	Event socket
wxSpinEvent	Event dari wxSpinButton
wxSplitterEvent	Event dari wxSplitterWindow
wxSysColourChangedEvent	Perubahan warna sistem
wxTimerEvent	Timer expiration
wxTreeEvent	Event untuk tree control
wxUpdateUIEvent	Perubahan pada user interface
WxWindowCreateEvent	Pembuatan window
wxWindowDestroyEvent	Penghapusan window
wxWizardEvent	Event wizard

Berkebalikan dengan banyaknya macam event yang dikenal, cara mendeklarasikan event handler adalah seragam. Berikut ini adalah aturan event handler:

- Tipe kembalian adalah void
- Bukan virtual
- Membutuhkan satu parameter berupa objek tipe event yang diproses (dari class event yang dibahas dalam tabel sebelumnya). Fungsi kemudian bisa memanfaatkan objek event tersebut, misal untuk mendapatkan tombol keyboard yang ditekan untuk key event. Pada kasus yang sederhana seperti tombol yang ditekan, umumnya objek event tersebut bisa diabaikan.

Dengan seragamnya bentuk event handler, developer menjadi lebih mudah. Apa yang perlu diingat oleh developer adalah tipe event yang ingin dihandle.

### **3.3 Menggunakan event table**

Seperti yang telah disebutkan sebelumnya, cara yang satu ini relatif lebih mudah dan umum untuk digunakan. Ketika menggunakan event table, developer mengatur

tabel event-event yang akan diproses, lengkap dengan event handlernya.

Berikut ini adalah langkah-langkah memproses event menggunakan event table:

- Mendeklarasikan class yang diturunkan baik secara langsung atau tidak langsung dari `wxEvtHandler`
- Menambahkan method/member function untuk class tersebut, yang dimaksudkan sebagai event handler untuk event yang ingin diproses.
- Mendeklarasikan event table di dalam class dengan `DECLARE_EVENT_TABLE`
- Mengimplementasikan event table di source code dengan `BEGIN_EVENT_TABLE...END_EVENT_TABLE`
- Menambahkan entry event table ke dalam event table, untuk memetakan setiap event yang ingin diproses dengan event handler (method/function member) yang telah dibuat.

Event juga dapat diproses oleh window selain window yang dikenakan event. Sebagai contoh, sebuah frame yang mengandung sebuah tombol (`wxButton`). Pada saat tombol di klik, `wxWidgets` akan memeriksa class `wxButton` akan adanya handler yang bersesuaian. Pada saat tidak ditemukan, `wxWidgets` akan memeriksa parentnya dalam hal ini, frame. Ketika ditemukan, maka handler yang bersesuaian akan dijalankan.

Namun, harap diperhatikan bahwa hanya command event (event yang diturunkan secara langsung atau tidak langsung dari `wxCommandEvent`) diaplikasikan secara rekursif kepada event handler window parent. Berikut ini adalah daftar event yang tidak dikirim ke event handler parent:

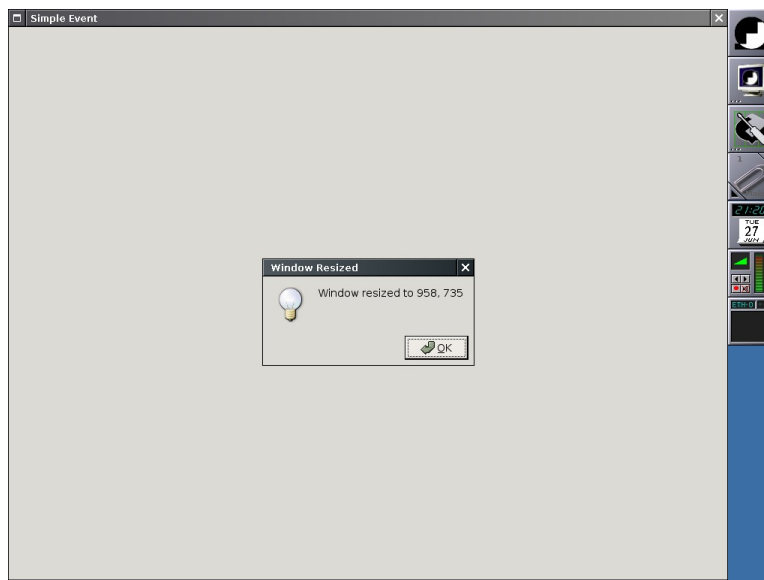
- `wxActivateEvent`
- `wxCloseEvent`
- `wxEraseEvent`
- `wxFocusEvent`
- `wxKeyEvent`
- `wxIdleEvent`
- `wxInitDialogEvent`
- `wxJoystickEvent`
- `wxMenuEvent`
- `wxMouseEvent`
- `wxMoveEvent`
- `wxPaintEvent`
- `wxQueryLayoutInfoEvent`
- `wxSizeEvent`
- `wxScrollWinEvent`
- `wxSysColourChangedEvent`.

Event-event tersebut dimaksudkan untuk window tertentu.

### 3.3.1 Contoh sederhana 1: simple\_event

Aplikasi simple\_event yang dibangun akan memiliki fitur-fitur berikut:

- Memiliki sebuah window utama yang akan memiliki title Simple Event
- User dapat melakukan klik pada tombol close pada title bar untuk menterminasi aplikasi
- Setiap kali window utama diresize (termasuk ketika pertama kali aplikasi dijalankan), sebuah message box dengan title Window Resized akan menampilkan pesan: Window resized to <lebar>,<tinggi>.



#### 3.3.1.1 Struktur direktori source code

```
simple_event/  
  build/  
    gtk/  
      simple_event  
  src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
  makefile.gtk
```

### **3.3.1.2 Source code**

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

#### **3.3.1.2.1 src/app.h**

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();

};

#endif // _APP_H_
```

#### **3.3.1.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Simple Event") );

    frame -> Show ( true );

    return true;
```

```
}
```

### 3.3.1.2.3 src/frame.h

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

    void OnSize ( wxSizeEvent& event );

private:

    DECLARE_EVENT_TABLE ()

};

#endif // _FRAME_H_
```

Penjelasan source code:

- Deklarasi class Frame dilakukan. Class Frame, yang diturunkan dari wxFrame, merupakan turunan dari wxEvtHandler
- Sebuah method/function member berikut telah dideklarasikan, dan dimaksudkan untuk memproses event pada saat frame diresize.

```
void OnSize ( wxSizeEvent& event );
```

Bisa dilihat, seperti telah disebutkan sebelumnya, event handler ini:

- memiliki tipe kembalian adalah void
  - Bukan virtual
  - Membutuhkan satu parameter berupa objek tipe event yang diproses, dalam hal ini adalah wxSizeEvent.
- Deklarasi event table telah dilakukan dengan DECLARE\_EVENT\_TABLE

### 3.3.1.2.4 src/frame.cpp

```
#include "wx/wx.h"  
#include "frame.h"
```

```
BEGIN_EVENT_TABLE ( Frame, wxFrame )
```

```
    EVT_SIZE ( Frame :: OnSize )
```

```
END_EVENT_TABLE ()
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )  
{  
  
}
```

```
void Frame :: OnSize ( wxSizeEvent& event )  
{  
    wxString msg;  
  
    wxSize size;  
  
    size = GetSize ();  
  
    msg.Printf ( _("Window resized to %d, %d"), size.GetWidth (), size.GetHeight () );  
  
    wxMessageBox ( msg, _( " Window Resized " ), wxOK | wxICON_INFORMATION,  
this );  
}
```

Penjelasan source code:

- Event table telah dideklarasikan dengan BEGIN\_EVENT\_TABLE...END\_EVENT\_TABLE. Harap perhatikan, event table yang dideklarasikan adalah milik Frame, yang menjadi anak class dari wxFrame.
- Sebuah entry yang memetakan event wxSizeEvent ke event handler Frame :: OnSize () telah dibuat. Harap diperhatikan macro EVT\_SIZE yang dipergunakan.

EVT\_SIZE(func)

EVT\_SIZE ( Frame :: OnSize )

Macro ini akan menerima method/member function/event handler yang menerima wxSizeEvent sebagai argumennya.

- Penjelasan OnSize ():
  - Sebuah wxString dengan nama msg dideklarasikan. Variabel ini akan dimaksudkan untuk menampung pesan yang akan ditampilkan pada message box. Mengenai wxString, akan dibahas pada bab yang membahas berbagai class wxWidgets.
  - Sebuah wxSize dengan nama size dideklarasikan. Variabel ini kemudian diassign nilai berupa nilai kembalian dari wxWindow :: GetSize ():

wxSize GetSize() const

variabel size akan berisikan ukuran window aktif

- msg kemudian akan diformat untuk berisikan: "Window resized to <lebar>,<tinggi>
- Selanjutnya, message box informasi dengan title "Window Resized" pun ditampilkan, dengan pesan variabel msg.

### **3.3.1.3 Kompilasi**

#### **3.3.1.3.1 Kompilasi di Linux (GTK+)**

Makefile yang digunakan: makefile.gtk:

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk
```



```
PROGRAM = simple_event
OBJECTS = src/app.o src/frame.o

# system
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES:    .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all:  $(PROGRAM)

$(PROGRAM):  $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

### 3.3.2 Contoh sederhana 2: hello5

Aplikasi hello5 yang dibangun akan memiliki fitur-fitur berikut:

- Memiliki sebuah window utama yang akan memiliki title Hello World!
- User dapat melakukan klik pada tombol close pada title bar untuk menterminasi aplikasi
- Sebuah menu File akan ditambahkan. Menu file tersebut berisikan sebuah anak menu, yaitu Quit (dengan shortcut CTRL-Q; dengan hint Quit application). Apabila menu Quit ini diklik, maka aplikasi akan diterminasi.
- Sebuah statusbar akan dibuat. Pertama kali aplikasi dijalankan, statusbar akan menampilkan pesan "Welcome to Hello World!". Pada saat mouse melewati menu File | Quit, statusbar akan menampilkan hint dari menu File | Quit tersebut, yaitu "Quit application".

Aplikasi hello5 merupakan modifikasi dari hello4 yang dibahas pada bab

sebelumnya.



### **3.3.2.1 Struktur direktori source code**

```
hello5/  
  build/  
    gtk/  
      hello5  
  src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
  makefile.gtk
```

### **3.3.2.2 Source code**

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

### 3.3.2.2.1 src/app.h

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();

};

#endif // _APP_H_
```

### 3.3.2.2.2 src/app.cpp

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Hello World!") );

    frame -> Show ( true );

    return true;
}
```

### 3.3.2.2.3 src/frame.h

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

    void OnQuit ( wxCommandEvent& event );

private:

    DECLARE_EVENT_TABLE ()

};

#endif // _FRAME_H_
```

Penjelasan source code:

- Deklarasi class Frame dilakukan. Class Frame, yang diturunkan dari wxFrame, merupakan turunan dari wxEvtHandler
- Sebuah method/function member berikut telah dideklarasikan, dan dimaksudkan untuk memproses event pada saat menu File | Quit diklik.

```
void OnQuit ( wxCommandEvent& event );
```

Bisa dilihat, seperti telah disebutkan sebelumnya, event handler ini:

- memiliki tipe kembalian adalah void
  - Bukan virtual
  - Membutuhkan satu parameter berupa objek tipe event yang diproses, dalam hal ini adalah wxCommandEvent
- Deklarasi event table telah dilakukan dengan DECLARE\_EVENT\_TABLE

#### 3.3.2.2.4 src/frame.cpp

```
#include "wx/wx.h"
```

```
#include "wx/menu.h"
#include "frame.h"

BEGIN_EVENT_TABLE ( Frame, wxFrame )

    EVT_MENU ( wxID_EXIT, Frame :: OnQuit )

END_EVENT_TABLE ()

Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{

    CreateStatusBar ();

    SetStatusText ( _( "Welcome to Hello World!" ), 0 );

    wxMenu * fileMenu = new wxMenu;

    fileMenu -> Append ( wxID_EXIT, _( "&Quit \tCTRL-Q" ), _( "Quit application" ) );

    wxMenuBar * menuBar = new wxMenuBar;

    menuBar -> Append ( fileMenu, _( "&File" ) );

    SetMenuBar ( menuBar );

}

void Frame :: OnQuit ( wxCommandEvent& event )
{

    Close ();

}
```

Penjelasan source code:

- Event table telah dideklarasikan dengan BEGIN\_EVENT\_TABLE...END\_EVENT\_TABLE. Harap perhatikan, event table yang dideklarasikan adalah milik Frame, yang menjadi anak class dari wxFrame.

- Sebuah entry yang memetakan event wxCommandEvent ke event handler Frame :: OnQuit () telah dibuat. Harap diperhatikan macro EVT\_MENU yang dipergunakan.

```
EVT_MENU(id, func)
```

```
EVT_MENU ( wxID_EXIT, Frame :: OnQuit )
```

Macro ini akan menerima method/member function/event handler yang menerima wxCommandEvent sebagai argumennya. Membutuhkan Window ID.

- Harap diperhatikan pada saat penambahan menu item File | Quit, window ID yang digunakan adalah wxID\_EXIT
- Penjelasan OnQuit ():
  - Di dalam fungsi ini, wxWindow :: Close () akan menutup frame utama, yang akan menutup aplikasi

```
bool Close(bool force = false)
```

### **3.3.2.3 Kompilasi**

#### **3.3.2.3.1 Kompilasi di Linux (GTK+)**

Makefile yang digunakan: makefile.gtk:

```
# simple makefile for simple wx project  
# (c) Nop, 5 Feb 2006
```

```
# variables  
WXCONFIG = wx-config  
WXPLAT = gtk
```

```
PROGRAM = hello5  
OBJECTS = src/app.o src/frame.o
```

```
# system
```

```
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES:    .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all:  $(PROGRAM)

$(PROGRAM):  $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

### 3.4 Memroses event secara dinamis

Pemrosesan event juga dapat dilakukan secara dinamis pada saat runtime. Umumnya, developer memroses event secara dinamis karena:

- developer ingin menggunakan event handler yang berbeda pada saat berbeda
- menggunakan bahasa pemrograman lain yang tidak dapat memroses event secara statis, seperti Python.

Untuk memroses event secara dinamis, developer akan menggunakan:

- `wxEvtHandler::Connect ()`
- `wxEvtHandler::Disconnect ()`. Umumnya tidak perlu dipanggil secara eksplisit karena proses disconnection ini akan terjadi secara otomatis pada saat object window didestroy.

Berbeda dengan penanganan event memanfaatkan event table, developer tidak perlu menggunakan `DECLARE_EVENT_TABLE` ataupun mendeklarasikan event table dengan `BEGIN_EVENT_TABLE ... END_EVENT_TABLE`. Namun, tentu saja, method/function member yang berfungsi sebagai event handler tetap harus dideklarasikan. Tidak ada perbedaan antara event handler pada penggunaan

event table ataupun pada pemrosesan event secara dinamis.

Apa yang berbeda hanyalah pemanggilan `wxEvtHandler::Connect ()` atau `wxEvtHandler::Disconnect ()`.

```
void Connect(int id, int lastId, wxEventType eventType, wxObjectEventFunction  
function, wxObject* userData = NULL, wxEvtHandler* eventSink = NULL)
```

```
void Connect(int id, wxEventType eventType, wxObjectEventFunction function,  
wxObject* userData = NULL, wxEvtHandler* eventSink = NULL)
```

```
void Connect(wxEventType eventType, wxObjectEventFunction function, wxObject*  
userData = NULL, wxEvtHandler* eventSink = NULL)
```

```
bool Disconnect(wxEventType eventType = wxEVT_NULL, wxObjectEventFunction  
function = NULL, wxObject* userData = NULL, wxEvtHandler* eventSink = NULL)
```

```
bool Disconnect(int id = wxID_ANY, wxEventType eventType = wxEVT_NULL,  
wxObjectEventFunction function = NULL, wxObject* userData = NULL, wxEvtHandler*  
eventSink = NULL)
```

```
bool Disconnect(int id, int lastId = wxID_ANY, wxEventType eventType = wxEVT_NULL,  
wxObjectEventFunction function = NULL, wxObject* userData = NULL, wxEvtHandler*  
eventSink = NULL)
```

Untuk memperjelas pembahasan, sekaligus melihat perbedaan antara penggunaan event table dan pemrosesan dinamis, buku ini akan menulis ulang contoh-contoh penggunaan event table sebelumnya.

### **3.4.1 Contoh sederhana 1: simple\_event**

Aplikasi `simple_event` yang dibangun akan memiliki fitur-fitur berikut:

- Memiliki sebuah window utama yang akan memiliki title Simple Event
- User dapat melakukan klik pada tombol close pada title bar untuk menterminasi aplikasi
- Setiap kali window utama diresize (termasuk ketika pertama kali aplikasi dijalankan), sebuah message box dengan title Window Resized akan menampilkan pesan: Window resized to <lebar>,<tinggi>.

#### **3.4.1.1 Struktur direktori source code**



```
simple_event2/  
  build/  
    gtk/  
      simple_event2  
  src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
  makefile.gtk
```

### **3.4.1.2 Source code**

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

Untuk penjelasan source code, lihatlah juga pada contoh yang sama pada bagian penanganan event dengan event table, apabila diperlukan.

#### **3.4.1.2.1 src/app.h**

```
#ifndef _APP_H_  
#define _APP_H_  
  
class App : public wxApp  
{  
public:  
  
    virtual bool OnInit ();  
  
};  
  
#endif // _APP_H_
```

#### **3.4.1.2.2 src/app.cpp**

```
#include "wx/wx.h"
```

```
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Simple Event") );

    frame -> Show ( true );

    return true;
}
```

### 3.4.1.2.3 src/frame.h

```
#ifndef _FRAME_H_
#define _FRAME_H_

enum
{
    ID_FRAME = wxID_HIGHEST + 1
};

class Frame: public wxFrame
{
public:
    Frame ( const wxString& title );

    void OnSize ( wxSizeEvent& event );
};
```

```
#endif // _FRAME_H_
```

Penjelasan source code:

- Deklarasi class Frame dilakukan. Class Frame, yang diturunkan dari wxFrame, merupakan turunan dari wxEvtHandler
- Sebuah method/function member berikut telah dideklarasikan, dan dimaksudkan untuk memproses event pada saat frame diresize.

```
void OnSize ( wxSizeEvent& event );
```

Bisa dilihat, seperti telah disebutkan sebelumnya, event handler ini:

- memiliki tipe kembalian adalah void
  - Bukan virtual
  - Membutuhkan satu parameter berupa objek tipe event yang diproses, dalam hal ini adalah wxSizeEvent.
- Sebuah window ID ID\_FRAME disiapkan sebagai ID Frame.

#### **3.4.1.2.4 src/frame.cpp**

```
#include "wx/wx.h"  
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, ID_FRAME, title )  
{  
  
    Connect ( ID_FRAME, wxEVT_SIZE, wxSizeEventHandler ( Frame :: OnSize ) );  
  
}
```

```
void Frame :: OnSize ( wxSizeEvent& event )  
{  
    wxString msg;  
  
    wxSize size;  
  
    size = GetSize ();
```

```
msg.Printf ( _("Window resized to %d, %d"), size.GetWidth (), size.GetHeight () );
```

```
    wxMessageBox ( msg, _( " Window Resized " ), wxOK | wxICON_INFORMATION,  
this );  
}
```

Penjelasan source code:

- Lihatlah juga pada contoh yang sama pada bagian penanganan event dengan event table
- Perhatikan pembuatan Frame berikut, yang telah menggunakan ID baru:

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, ID_FRAME, title )
```

- Tidak ada event table yang dideklarasikan. Sebagai gantinya, `wxEvtHandler :: Connect ()` yang digunakan:

```
Connect ( ID_FRAME, wxEVT_SIZE, wxSizeEventHandler ( Frame :: OnSize ) );
```

- Penjelasan parameter `wxEvtHandler :: Connect ()`
  - `ID_FRAME` adalah ID milik Frame
  - `wxEVT_SIZE` adalah nama event yang akan diproses. Harap dibedakan dengan macro `EVT_SIZE` yang digunakan pada event table.
  - Parameter terakhir adalah pointer ke event handler. Sebuah macro `wxSizeEventHandler` dipergunakan untuk menginformasikan compiler (cast secara otomatis dilakukan pada event table static). Untuk event `wxSizeEvent` yang diproses, macro yang digunakan adalah `wxSizeEventHandler`. Secara umum, Untuk event `wxXYZEvent` yang diproses, macro yang digunakan adalah `wxXYZEventHandler`.

### **3.4.1.3 Kompilasi**

#### **3.4.1.3.1 Kompilasi di Linux (GTK+)**

Makefile yang digunakan: `makefile.gtk`:

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006

# variables
WXCONFIG = wx-config
WXPLAT = gtk

PROGRAM = simple_event2
OBJECTS = src/app.o src/frame.o

# system
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES:    .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all:  $(PROGRAM)

$(PROGRAM):  $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

### 3.4.2 Contoh sederhana 2: hello5

Aplikasi hello5 yang dibangun akan memiliki fitur-fitur berikut:

- Memiliki sebuah window utama yang akan memiliki title Hello World!
- User dapat melakukan klik pada tombol close pada title bar untuk menterminasi aplikasi
- Sebuah menu File akan ditambahkan. Menu file tersebut berisikan sebuah anak menu, yaitu Quit (dengan shortcut CTRL-Q; dengan hint Quit application). Apabila

menu Quit ini diklik, maka aplikasi akan diterminasi.

- Sebuah statusbar akan dibuat. Pertama kali aplikasi dijalankan, statusbar akan menampilkan pesan "Welcome to Hello World!". Pada saat mouse melewati menu File | Quit, statusbar akan menampilkan hint dari menu File | Quit tersebut, yaitu "Quit application".

Aplikasi hello5 merupakan modifikasi dari hello4 yang dibahas pada bab sebelumnya.

### **3.4.2.1 Struktur direktori source code**

```
hello5_2/  
  build/  
    gtk/  
      hello5_2  
  src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
  makefile.gtk
```

### **3.4.2.2 Source code**

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

Untuk penjelasan source code, lihatlah juga pada contoh yang sama pada bagian penanganan event dengan event table, apabila diperlukan.

#### **3.4.2.2.1 src/app.h**

```
#ifndef _APP_H_  
#define _APP_H_
```

```
class App : public wxApp
{
public:

    virtual bool OnInit ();

};

#endif // _APP_H_
```

#### **3.4.2.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Hello World!") );

    frame -> Show ( true );

    return true;
}
```

#### **3.4.2.2.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
```

public:

```
Frame ( const wxString& title );
```

```
void OnQuit ( wxCommandEvent& event );
```

```
};
```

```
#endif // _FRAME_H_
```

Penjelasan source code:

- Deklarasi class Frame dilakukan. Class Frame, yang diturunkan dari wxFrame, merupakan turunan dari wxEvtHandler
- Sebuah method/function member berikut telah dideklarasikan, dan dimaksudkan untuk memproses event pada saat menu File | Quit diklik.

```
void OnQuit ( wxCommandEvent& event );
```

Bisa dilihat, seperti telah disebutkan sebelumnya, event handler ini:

- memiliki tipe kembalian adalah void
- Bukan virtual
- Membutuhkan satu parameter berupa objek tipe event yang diproses, dalam hal ini adalah wxCommandEvent

#### **3.4.2.2.4 src/frame.cpp**

```
#include "wx/wx.h"
```

```
#include "wx/menu.h"
```

```
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )  
{
```

```
CreateStatusBar ();
```



```
SetStatusText ( _( "Welcome to Hello World!" ), 0 );

wxMenu * fileMenu = new wxMenu;

fileMenu -> Append ( wxID_EXIT, _( "&Quit \tCTRL-Q" ), _( "Quit application" ) );

wxMenuBar * menuBar = new wxMenuBar;

menuBar -> Append ( fileMenu, _( "&File" ) );

SetMenuBar ( menuBar );

        Connect ( wxID_EXIT, wxEVT_COMMAND_MENU_SELECTED,
wxCommandEventHandler ( Frame :: OnQuit ) );

}

void Frame :: OnQuit ( wxCommandEvent& event )
{

    Close ();

}
```

Penjelasan source code:

- Lihatlah juga pada contoh yang sama pada bagian penanganan event dengan event table
- Tidak ada event table yang dipergunakan. Sebagai gantinya, wxEvtHandler :: Connect () yang digunakan:

```
        Connect ( wxID_EXIT, wxEVT_COMMAND_MENU_SELECTED,
wxCommandEventHandler ( Frame :: OnQuit ) );
```

- Penjelasan parameter wxEvtHandler :: Connect ()
  - wxID\_EXIT adalah ID milik menu File | Quit
  - wxEVT\_COMMAND\_MENU\_SELECTED adalah nama event yang akan diproses. Harap dibedakan dengan macro EVT\_MENU yang digunakan pada event table.
  - Parameter terakhir adalah pointer ke event handler. Sebuah macro

wxCommandEventHandler dipergunakan untuk menginformasikan compiler (cast secara otomatis dilakukan pada event table static). Untuk event wxCommandEvent yang diproses, macro yang digunakan adalah wxCommandEventHandler. Secara umum, Untuk event wxXYZEvent yang diproses, macro yang digunakan adalah wxXYZEventHandler.

### **3.4.2.3 Kompilasi**

#### **3.4.2.3.1 Kompilasi di Linux (GTK+)**

Makefile yang digunakan: makefile.gtk:

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk
```

```
PROGRAM = hello5_2
OBJECTS = src/app.o src/frame.o
```

```
# system
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES: .o .cpp
```

```
.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<
```

```
all: $(PROGRAM)
```

```
$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
```

```
mkdir -p build/$(WXPLAT)
mv $(PROGRAM) build/$(WXPLAT)
rm -f src/*.o
```

clean:

```
rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

## Bab 4. Layout window menggunakan sizer

Bab ini membahas bagaimana developer bekerja dengan sistem layout di wxWidgets. Bagaimana berbagai widget/window ditempatkan di dalam satu frame dan dapat tetap tampil proporsional ketika frame di resize. Setelah membaca bab ini, diharapkan developer dapat mulai merancang frame utama aplikasi yang mengandung lebih satu widget/window.

### 4.1 Sekilas tentang sizer

Di dalam sistem layout berbagai bahasa atau pustaka, developer dapat menyusun widget/window di dalam satu frame, menggunakan peletakan berbasis posisi absolut. Dengan demikian, sebuah tombol misalnya, bisa dengan mudah ditempatkan pada posisi kiri 10 pixel, posisi atas 100 pixel, relatif terhadap frame yang mengandung tombol tersebut. Widget/window lain bisa ditempatkan sesuai kebutuhan. Menyusun berbagai window tersebut rasanya mudah sekali, terutama kalau bahasa pemrograman tersebut dilengkapi RAD tool yang memungkinkan perancangan dilakukan secara visual.

Walaupun unggul dari sisi kemudahan, beberapa bahasa atau pustaka tidak melakukan penyesuaian layout ketika suatu frame diresize. Akibatnya, frame tersebut yang pada awalnya dilayout dengan rapi tiba-tiba menjadi tidak proporsional. Mungkin saja tiba-tiba terdapat ruang kosong yang begitu besar di sisi kanan atau bawah. Mungkin saja beberapa window menjadi terpotong (atau, scrollbar akan ditampilkan). Belum lagi ketika terjadi perubahan font atau ukuran font yang digunakan, atau bahkan perubahan bahasa yang digunakan.

Beberapa bahasa atau pustaka (misal Java, GTK+, Qt) kemudian membangun algoritma layout yang lebih fleksibel. Begitupun wxWidgets dengan sizer-nya. Ide yang diusung adalah, setiap window melaporkan ukuran minimal yang diperlukan, dan kemungkinan window tersebut diregangkan (stretch) apabila ukuran window parent berubah. Sebagai catatan, sebuah sizer dapat berisikan sizer lainnya.

Dengan model seperti ini, umumnya, developer tidak mengatur ukuran inisial sebuah dialog, melainkan, dialog tersebut akan diassign sebuah sizer, yang kemudian akan diminta ukuran yang direkomendasikan. Sizer tersebut kemudian meminta informasi kepada anak-anaknya (bisa berupa window-window, ruang kosong, sizer lain).

Sizer memiliki fitur umum sebagai berikut:

- Ukuran minimal
- Border (ruang kosong untuk memisahkan elemen). Bisa border atas, bawah, kiri, kanan atau kombinasinya.
- Alignment. Bisa berupa atas/tengah/bawah atau kiri/tengah/kanan.
- Proporsi. Bisa berupa sebuah window tidak akan diresize lebih dari ukuran minimalnya, atau bagaimana sebuah window memiliki proporsi lebih besar/kecil dari window lainnya dalam memakan space yang tersedia.

## **4.2 Macam-macam sizer di wxWidgets**

Terdapat bermacam-macam sizer di wxWidgets, tergantung dari mana yang lebih disukai oleh developer/lebih cocok digunakan pada kondisi atau kebutuhan tertentu. Masing-masing sizer tersebut berbeda dari sisi fitur yang dimiliki.

Berikut ini adalah macam-macam sizer yang didukung:

- Box Sizer (wxBoxSizer)
- Static Box Sizer (wxStaticBoxSizer)
- Grid Sizer (wxGridSizer)
- Flex Grid Sizer (wxFlexGridSizer)
- Grid Bag Sizer (wxGridBagSizer)

Semua class tersebut diturunkan dari class wxSizer.

Semua dari sizer tersebut akan dibahas di dalam bab ini. Namun, untuk bab-bab seterusnya, hanya Box Sizer (wxBoxSizer) yang akan digunakan.

## **4.3 Mengatur sizer pada wxWindow**

Untuk mengatur sizer pada wxWindow (atau pada turunannya, misal pada wxFrame dan turunannya), gunakanlah wxWindow :: SetSizer () atau wxWindow :: SetSizerAndFit (). Yang terakhir pada prinsipnya sama dengan yang pertama, hanya, SetSizerAndFit juga akan mengatur size hint untuk window berdasarkan ukuran minimal sizer.

```
void SetSizer(wxSizer* sizer, bool deleteOld=true)
```

```
void SetSizerAndFit(wxSizer* sizer, bool deleteOld=true)
```

## **4.4 Menambahkan window ke sizer**

Untuk menambahkan window anak ke dalam sizer (wxBoxSizer, wxStaticBoxSizer, wxGridSizer, wxFlexGridSizer), gunakanlah wxSizer :: Add

```
wxSizerItem* Add(wxWindow* window, const wxSizerFlags& flags)
```

```
wxSizerItem* Add(wxWindow* window, int proportion = 0, int flag = 0, int border = 0, wxObject* userData = NULL)
```

```
wxSizerItem* Add(wxSizer* sizer, const wxSizerFlags& flags)
```

```
wxSizerItem* Add(wxSizer* sizer, int proportion = 0, int flag = 0, int border = 0, wxObject* userData = NULL)
```

```
wxSizerItem* Add(int width, int height, int proportion = 0, int flag = 0, int border = 0, wxObject* userData = NULL)
```

Khusus untuk wxGridBagSizer, gunakanlah wxGridBagSizer :: Add:

```
wxSizerItem* Add(wxWindow* window, const wxGBPosition& pos, const wxGBSpan& span = wxDefaultSpan, int flag = 0, int border = 0, wxObject* userData = NULL)
```

```
wxSizerItem* Add(wxSizer* sizer, const wxGBPosition& pos, const wxGBSpan& span = wxDefaultSpan, int flag = 0, int border = 0, wxObject* userData = NULL)
```

```
wxSizerItem* Add(int width, int height, const wxGBPosition& pos, const wxGBSpan& span = wxDefaultSpan, int flag = 0, int border = 0, wxObject* userData = NULL)
```

```
wxSizerItem* Add(wxGBSizerItem* item)
```

Penjelasan parameter:

- window. Window yang ingin ditambahkan ke dalam sizer. Ukuran inisial (ditentukan manual atau hasil kalkulasi internal dengan wxDefaultSize) dianggap sebagai ukuran minimal.
- Sizer. Sizer anak yang akan ditambahkan. Akan membantuk hirarki sizer.
- Width dan height. Dimensi spacer yang ditambahkan ke dalam sizer.
- Proportion. Proporsi ukuran window yang ditambahkan dalam sizer. Proporsi 0 dapat diartikan sebagai 'ukuran tidak berubah'.
- Flag. Berbagai flag yang dikenal.
  - WxTOP, border pada sisi atas
  - wxBOTTOM, border pada sisi bawah
  - wxLEFT, border pada sisi kiri

- wxRIGHT, border pada sisi kanan
- wxALL (sama dengan wxLEFT | wxRIGHT | wxTOP | wxBOTTOM)
- wxEXPAND atau wxGROW, item akan diexpand untuk memenuhi ruang kosong yang diassign
- wxSHAPED, item akan diexpand sambil tetap memperhatikan aspect ratio
- wxFIXED\_MINSIZE, mempertahankan ukuran semula yang telah diset
- wxALIGN\_CENTER, rata tengah horizontal dan vertical. Sama dengan wxALIGN\_CENTER\_VERTICAL | wxALIGN\_CENTER\_HORIZONTAL
- wxALIGN\_LEFT, rata kiri
- wxALIGN\_RIGHT, rata kanan
- wxALIGN\_TOP, rata atas
- wxALIGN\_BOTTOM, rata bawah
- wxALIGN\_CENTER\_VERTICAL, rata tengah vertical
- wxALIGN\_CENTER\_HORIZONTAL, rata tengah horizontal
- border. Ukuran border apabila diset di flag
- userData. Mengijinkan objek ekstra di-attach ke sizer item, untuk digunakan pada kelas turunan pada saat informasi sizing lebih kompleks daripada yang diijinkan oleh proportion dan flag.
- Flags, objek wxSizerFlags
- pos, posisi grid bag (lihatlah pada pembahasan wxGridBagSizer untuk lebih jelasnya)
- span, span pada grid bag (lihatlah pada pembahasan wxGridBagSizer untuk lebih jelasnya)

## 4.5 Menambahkan space ke sizer

Developer bisa pula menambahkan spacer (ruang kosong) ke dalam sizer. Terdapat dua macam space:

- non-stretchable space. Spacer berukuran tetap.

Ditambahkan dengan:

```
wxSizer::AddSpacer
```

```
wxSizerItem* AddSpacer(int size)
```

Membutuhkan parameter berupa ukuran spacer.

- stretchable space. Spacer yang dapat diregangkan.

Ditambahkan dengan:

```
wxSizer::AddStretchSpacer
```

```
wxSizerItem* AddStretchSpacer(int prop = 1)
```

Membutuhkan parameter berupa proporsi spacer.

Sebagai catatan, spacer akan bergantung pada jenis/sifat sizer yang digunakan.

## 4.6 Menggunakan Box Sizer (*wxBoxSizer*)

Sizer *wxBoxSizer* melayout anak-anak windownya secara vertikal atau horizontal, tergantung style yang diberikan pada constructor. Terdapat dua style yang didukung, yaitu vertical dan horizontal:

- Apabila style yang digunakan adalah vertical, maka anak-anak window dapat diposisikan rata kiri, rata kanan atau tengah. Posisi anak-anak window akan disusun berderet secara vertical (atas bawah).
- Apabila style yang digunakan adalah horizontal, maka anak-anak window dapat diposisikan rata atas, rata bawah atau tengah. Posisi anak-anak window akan disusun berderet secara horizontal (kiri kanan).

Constructor:

```
wxBoxSizer(int orient)
```

- orient bisa berupa *wxHORIZONTAL* ataupun *wxVERTICAL*

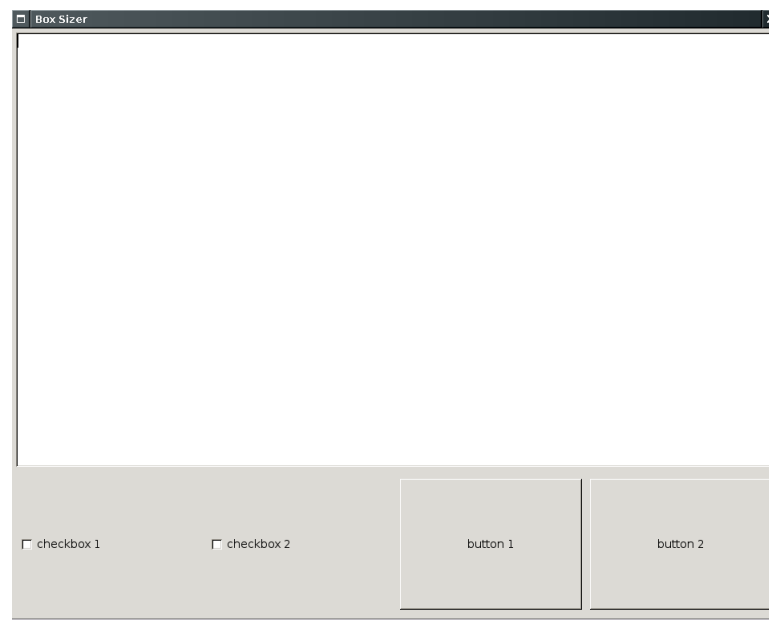
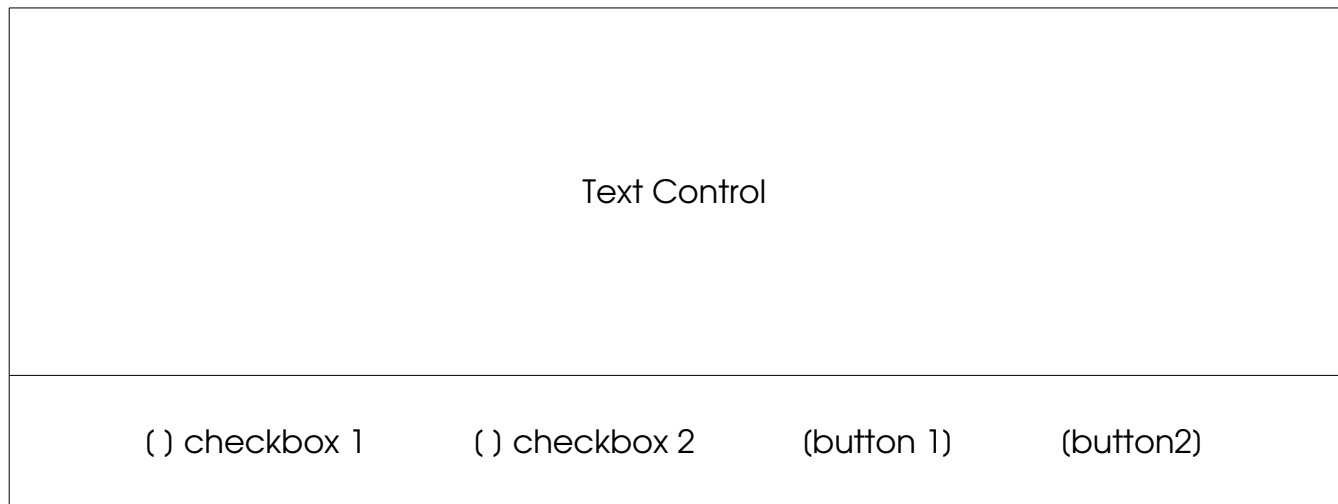
Untuk lebih jelasnya, sebuah contoh aplikasi akan dibuat dengan layout menggunakan *wxBoxSizer*. Aplikasi tersebut terdiri dari satu frame, di mana di dalam frame, terdapat window-window berikut:

- Sebuah Text Control multiline pada bagian atas. Proporsi Text Control ini adalah 3 kali yang lainnya.
- Dua buah checkbox dan dua buah tombol pada bagian bawah

Untuk saat ini, tidak ada event handler yang digunakan. Aplikasi tersebut juga tidak memiliki fungsi apapun. Hal ini dimaksudkan agar pembahasan terfokus pada sizer.

Berikut ini adalah gambaran visual frame utama aplikasi:





#### 4.6.1 Struktur direktori box\_sizer

```
box_sizer/  
  build/  
    gtk/
```

```
        box_sizer
src/
    app.h
    app.cpp
    frame.h
    frame.cpp
makefile.gtk
```

## 4.6.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

### 4.6.2.1 *src/app.h*

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

### 4.6.2.2 *src/app.cpp*

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"
```

```
DECLARE_APP ( App )
```

```
IMPLEMENT_APP ( App )
```

```
bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Box Sizer") );

    frame -> Show ( true );

    return true;
}
```

#### **4.6.2.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### **4.6.2.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "wx/button.h"
#include "wx/checkbox.h"
#include "wx/textctrl.h"
#include "wx/sizer.h"
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{

    wxButton * button1 = new wxButton ( this, wxID_ANY, _ ( "button 1" ) );

    wxButton * button2 = new wxButton ( this, wxID_ANY, _ ( "button 2" ) );

    wxCheckBox * check1 = new wxCheckBox ( this, wxID_ANY, _ ( "checkbox 1" ) );

    wxCheckBox * check2 = new wxCheckBox ( this, wxID_ANY, _ ( "checkbox 2" ) );

        wxTextCtrl * textCtrl1 = new wxTextCtrl ( this, wxID_ANY, wxEmptyString,
wxDefaultPosition,
            wxDefaultSize, wxTE_MULTILINE );


    wxBoxSizer * sizerMain = new wxBoxSizer ( wxVERTICAL );

    wxBoxSizer * sizerBottom = new wxBoxSizer ( wxHORIZONTAL );


    sizerBottom -> Add ( check1, 1, wxEXPAND | wxALL, 5 );

    sizerBottom -> Add ( check2, 1, wxEXPAND | wxALL, 5 );

    sizerBottom -> Add ( button1, 1, wxEXPAND | wxALL, 5 );

    sizerBottom -> Add ( button2, 1, wxEXPAND | wxALL, 5 );


    sizerMain -> Add ( textCtrl1, 3, wxEXPAND | wxALL, 5 );

    sizerMain -> Add ( sizerBottom, 1, wxEXPAND | wxALL, 5 );


    SetSizerAndFit ( sizerMain );

}
```

Penjelasan source code:

- Pembuatan tombol (button) dapat dilakukan dengan sintaks:

```
wxButton * button1 = new wxButton ( this, wxID_ANY, _ ( "button 1" ) );
```

- Parameter pertama adalah parent, parameter kedua adalah window ID dan parameter ketiga adalah title button

- Pembuatan checkbox dapat dilakukan dengan sintaks:

```
wxCheckBox * check1 = new wxCheckBox ( this, wxID_ANY, _ ( "checkbox 1" ) );
```

- Parameter pertama adalah parent, parameter kedua adalah window ID dan parameter ketiga adalah title checkbox

- Pembuatan text control dapat dilakukan dengan sintaks:

```
wxTextCtrl * textCtrl1 = new wxTextCtrl ( this, wxID_ANY, wxEmptyString,  
wxDefaultPosition, wxDefaultSize, wxTE_MULTILINE );
```

- Parameter pertama adalah parent, parameter kedua adalah window ID, parameter ketiga adalah value inisial, parameter keempat adalah posisi default, parameter kelima adalah ukuran default, dan parameter keenam adalah style.
- Parameter keenam (style) diberikan berupa wxTE\_MULTILINE untuk meminta agar text control dapat menerima lebih dari satu baris (multi line).
- Untuk sizer, pertama-tama, disepakati bahwa akan terdapat dua sizer:
  - Sizer pertama adalah untuk mengandung text control dan deretan checkbox dan button. Karena dilayout atas bawah, maka orientasinya adalah vertical.
  - Sizer kedua adalah untuk mengandung deretan checkbox dan button. Karena dilayout kiri kanan, maka orientasinya adalah horizontal.
  - Sizer pertama akan mengandung text control dan sizer kedua.
- Pembuatan box sizer berorientasi vertical:

```
wxBoxSizer * sizerMain = new wxBoxSizer ( wxVERTICAL );
```

- Contoh penambahan button:

```
sizerBottom -> Add ( button1, 1, wxEXPAND | wxALL, 5 );
```

- Parameter pertama adalah button yang ingin ditambahkan. Parameter kedua adalah proporsi. Parameter ketiga adalah flag yang menandakan window akan diresize sesuai ruang kosong yang disediakan (wxEXPAND) dan memiliki border di semua sisi (wxALL), sebesar 5 pixel (parameter keempat).
- Contoh penambahan sizer:  
  
sizerMain -> Add ( sizerBottom, 1, wxEXPAND | wxALL, 5 );
- Terakhir, wxWindow :: SetSizerAndFit () dipanggil untuk mengatur sizer sekaligus mengatur size hint

## 4.6.3 Kompilasi

### 4.6.3.1 Kompilasi di Linux (GTK+)

Makefile yang digunakan: makefile.gtk:

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk
```

```
PROGRAM = box_sizer
OBJECTS = src/app.o src/frame.o
```

```
# system
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES: .o .cpp
```

```
.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<
```

```
all: $(PROGRAM)
```

```
$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

## 4.7 Menggunakan Static Box Sizer (*wxStaticBoxSizer*)

Static Box Sizer (*wxStaticBoxSizer*) diturunkan dari *wxBoxSizer*, namun menambahkan static box di sekeliling sizer. Static box tersebut bisa dibuat terpisah ataupun dibuatkan secara otomatis oleh sizer. Lihatlah juga penjelasan subbab sebelumnya tentang Box Sizer.

Constructor:

```
wxStaticBoxSizer(wxStaticBox* box, int orient)
```

```
wxStaticBoxSizer(int orient, wxWindow *parent, const wxString& label = wxEmptyString)
```

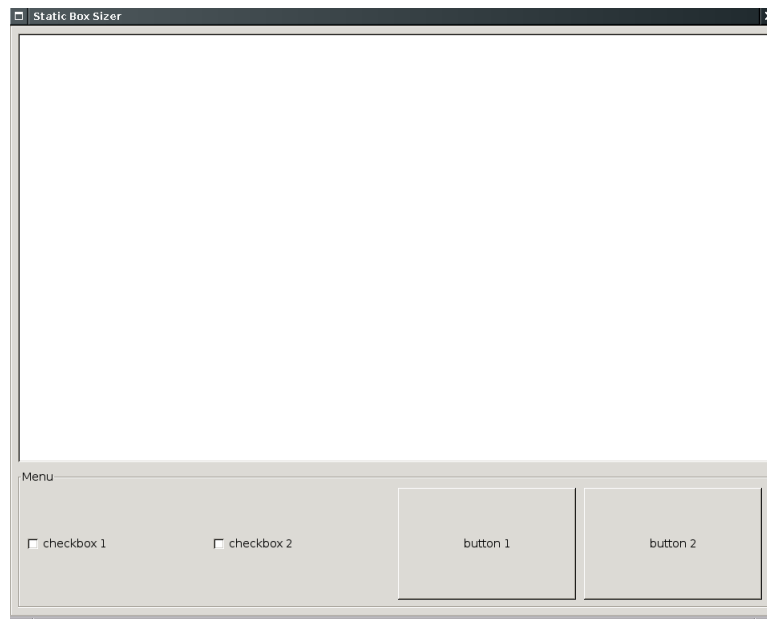
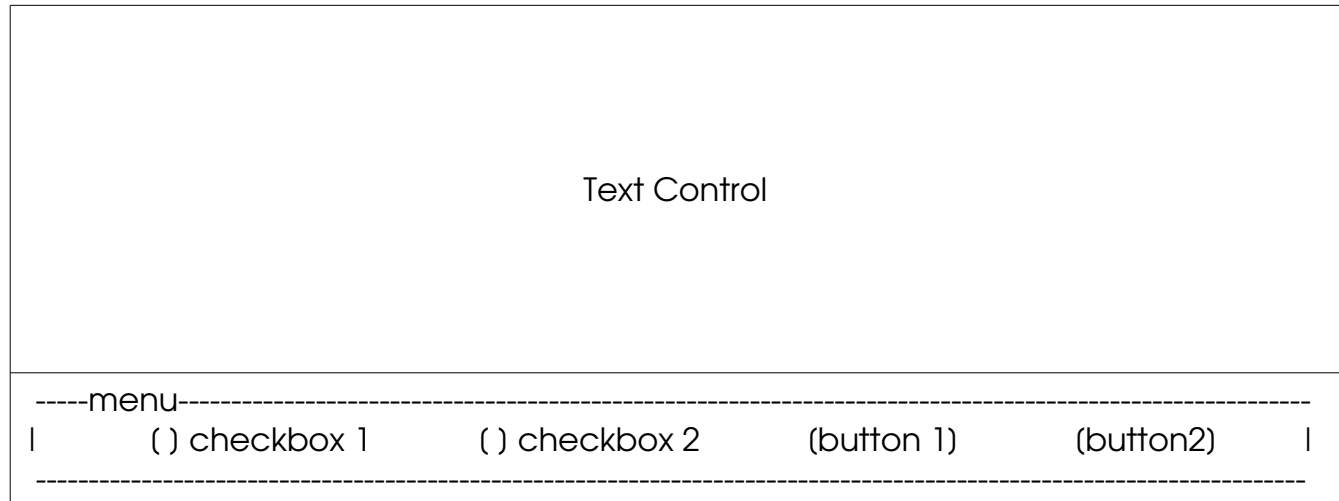
– orient bisa berupa *wxHORIZONTAL* ataupun *wxVERTICAL*

Untuk lebih jelasnya, sebuah aplikasi hasil modifikasi contoh penggunaan box sizer akan dibuat dengan layout menggunakan *wxStaticBoxSizer*. Aplikasi tersebut terdiri dari satu frame, di mana di dalam frame, terdapat window-window berikut:

- Sebuah Text Control multiline pada bagian atas. Proporsi Text Control ini adalah 3 kali yang lainnya.
- Dua buah checkbox dan dua buah tombol pada bagian bawah. Sebuah static box dengan title 'menu' akan mengelilingi mereka.

Untuk saat ini, tidak ada event handler yang digunakan. Aplikasi tersebut juga tidak memiliki fungsi apapun. Hal ini dimaksudkan agar pembahasan terfokus pada sizer.

Berikut ini adalah gambaran visual frame utama aplikasi:



#### 4.7.1 Struktur direktori static\_box\_sizer

static\_box\_sizer/  
build/



```
        gtk/  
            static_box_sizer  
src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
makefile.gtk
```

## 4.7.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

Untuk penjelasan, lihatlah juga pembahasan subbab sebelumnya tentang wxBoxSizer.

### 4.7.2.1 *src/app.h*

```
#ifndef _APP_H_  
#define _APP_H_  
  
class App : public wxApp  
{  
public:  
  
    virtual bool OnInit ();  
};  
  
#endif // _APP_H_
```

### 4.7.2.2 *src/app.cpp*

```
#include "wx/wx.h"  
#include "frame.h"
```

```
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Static Box Sizer" ) );

    frame -> Show ( true );

    return true;
}
```

#### **4.7.2.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### **4.7.2.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "wx/button.h"
#include "wx/checkbox.h"
#include "wx/textctrl.h"
#include "wx/sizer.h"
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{
```

```
    wxButton * button1 = new wxButton ( this, wxID_ANY, _ ( "button 1" ) );
```

```
    wxButton * button2 = new wxButton ( this, wxID_ANY, _ ( "button 2" ) );
```

```
    wxCheckBox * check1 = new wxCheckBox ( this, wxID_ANY, _ ( "checkbox 1" ) );
```

```
    wxCheckBox * check2 = new wxCheckBox ( this, wxID_ANY, _ ( "checkbox 2" ) );
```

```
        wxTextCtrl * textCtrl1 = new wxTextCtrl ( this, wxID_ANY, wxEmptyString,
wxDefaultPosition,
            wxDefaultSize, wxTE_MULTILINE );
```

```
    wxStaticBoxSizer * sizerMain = new wxStaticBoxSizer ( wxVERTICAL, this );
```

```
        wxStaticBoxSizer * sizerBottom = new wxStaticBoxSizer ( wxHORIZONTAL, this,
_( "Menu" ) );
```

```
    sizerBottom -> Add ( check1, 1, wxEXPAND | wxALL, 5 );
```

```
    sizerBottom -> Add ( check2, 1, wxEXPAND | wxALL, 5 );
```

```
    sizerBottom -> Add ( button1, 1, wxEXPAND | wxALL, 5 );
```

```
    sizerBottom -> Add ( button2, 1, wxEXPAND | wxALL, 5 );
```

```
    sizerMain -> Add ( textCtrl1, 3, wxEXPAND | wxALL, 5 );
```

```
sizerMain -> Add ( sizerBottom, 1, wxEXPAND | wxALL, 5 );
```

```
SetSizerAndFit ( sizerMain );
```

```
}
```

Penjelasan source code:

- Lihatlah juga pada pembahasan subbab sebelumnya mengenai wxBoxSizer
- Pembuatan wxStaticBoxSizer berorientasi vertical, tanpa title:

```
wxStaticBoxSizer * sizerMain = new wxStaticBoxSizer ( wxVERTICAL, this );
```

- Pembuatan wxStaticBoxSizer berorientasi horizontal, tanpa title 'menu':

```
wxStaticBoxSizer * sizerBottom = new wxStaticBoxSizer ( wxHORIZONTAL, this,  
_( "Menu" ) );
```

## 4.7.3 Kompilasi

### 4.7.3.1 Kompilasi di Linux (GTK+)

Makefile yang digunakan: makefile.gtk:

```
# simple makefile for simple wx project  
# (c) Nop, 5 Feb 2006
```

```
# variables  
WXCONFIG = wx-config  
WXPLAT = gtk
```

```
PROGRAM = static_box_sizer  
OBJECTS = src/app.o src/frame.o
```

```
# system  
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES:    .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all:  $(PROGRAM)

$(PROGRAM):  $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

## 4.8 Menggunakan Grid Sizer (wxGridSizer)

Grid Sizer (wxGridSizer) adalah sizer yang melayout anak-anak windownya dalam tabel dua dimensi dimana semua sel pada tabel memiliki ukuran yang sama. Lebar dari setiap sel adalah lebar dari anak window yang terlebar. Tinggi setiap sel adalah tinggi dari anak window yang tertinggi.

Constructor:

```
wxGridSizer(int rows, int cols, int vgap, int hgap)
```

```
wxGridSizer(int cols, int vgap = 0, int hgap = 0)
```

- rows adalah jumlah baris dalam tabel
- cols adalah jumlah kolom dalam tabel
- vgap dan hgap adalah space ekstra diantara anak-anak window

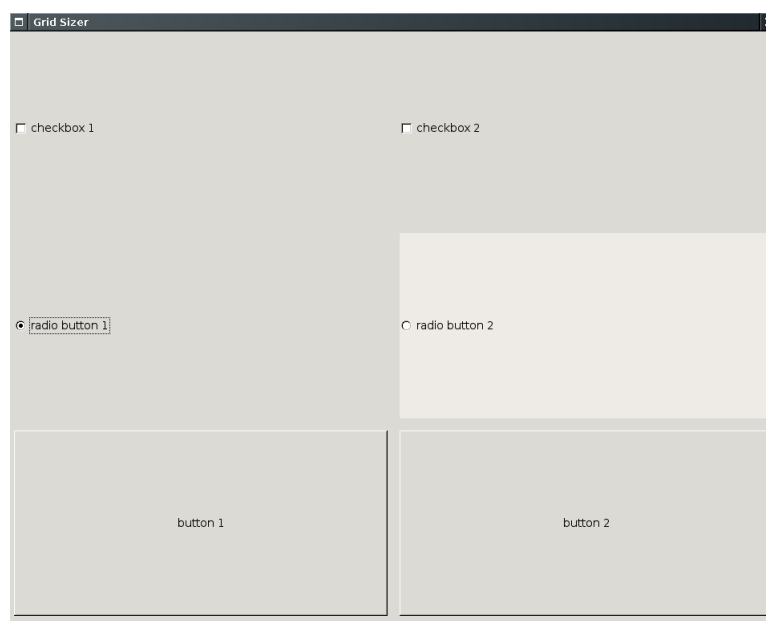
Untuk lebih jelasnya, sebuah contoh aplikasi akan dibuat dengan layout menggunakan wxGridSizer. Aplikasi tersebut terdiri dari satu frame, di mana di dalam frame, terdapat window-window berikut:

- Dua buah checkbox pada baris pertama
- Dua buah radio button pada baris kedua
- dan dua buah tombol pada baris ketiga

Untuk saat ini, tidak ada event handler yang digunakan. Aplikasi tersebut juga tidak memiliki fungsi apapun. Hal ini dimaksudkan agar pembahasan terfokus pada sizer.

Berikut ini adalah gambaran visual frame utama aplikasi:

<input type="checkbox"/> checkbox1	<input type="checkbox"/> checkbox 2
<input type="radio"/> radio button 1	<input type="radio"/> radio button 2
(button 1)	(button 2)



### 4.8.1 Struktur direktori grid\_sizer

```
grid_sizer/  
    build/  
        gtk/  
            grid_sizer  
    src/  
        app.h  
        app.cpp  
        frame.h  
        frame.cpp  
    makefile.gtk
```

### 4.8.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

Lihatlah pembahasan pada subbab sebelumnya apabila diperlukan.

#### 4.8.2.1 src/app.h

```
#ifndef _APP_H_  
#define _APP_H_  
  
class App : public wxApp  
{  
public:  
  
    virtual bool OnInit ();  
};  
  
#endif // _APP_H_
```

#### **4.8.2.2 *src/app.cpp***

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Grid Sizer") );

    frame -> Show ( true );

    return true;
}
```

#### **4.8.2.3 *src/frame.h***

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### **4.8.2.4 *src/frame.cpp***



```
#include "wx/wx.h"
#include "wx/button.h"
#include "wx/checkbox.h"
#include "wx/radiobut.h"
#include "wx/sizer.h"
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{
```

```
    wxButton * button1 = new wxButton ( this, wxID_ANY, _ ( "button 1" ),
wxDefaultPosition, wxSize ( 300, 100 ) );
```

```
    wxButton * button2 = new wxButton ( this, wxID_ANY, _ ( "button 2" ) );
```

```
    wxCheckBox * check1 = new wxCheckBox ( this, wxID_ANY, _ ( "checkbox 1" ) );
```

```
    wxCheckBox * check2 = new wxCheckBox ( this, wxID_ANY, _ ( "checkbox 2" ) );
```

```
    wxRadioButton * radio1 = new wxRadioButton ( this, wxID_ANY, _ ( "radio button
1" ) );
```

```
    wxRadioButton * radio2 = new wxRadioButton ( this, wxID_ANY, _ ( "radio button
2" ) );
```

```
    wxGridSizer * sizerMain = new wxGridSizer ( 3, 2, 5, 5 );
```

```
    sizerMain -> Add ( check1, 1, wxEXPAND | wxALL, 5 );
```

```
    sizerMain -> Add ( check2, 1, wxEXPAND | wxALL, 5 );
```

```
    sizerMain -> Add ( radio1, 1, wxEXPAND | wxALL, 5 );
```

```
    sizerMain -> Add ( radio2, 1, wxEXPAND | wxALL, 5 );
```

```
    sizerMain -> Add ( button1, 1, wxEXPAND | wxALL, 5 );
```

```
sizerMain -> Add ( button2, 1, wxEXPAND | wxALL, 5 );
```

```
SetSizerAndFit ( sizerMain );
```

```
}
```

Penjelasan source code:

- Lihatlah pembahasan subbab sebelumnya apabila diperlukan
- Pembuatan radio button dapat dilakukan dengan sintaks:

```
wxRadioButton * radio1 = new wxRadioButton ( this, wxID_ANY, _ ( "radio button 1"
```

- Parameter pertama adalah parent, parameter kedua adalah window ID dan parameter ketiga adalah title radio button

- Sebuah button yang dibuat dengan:

```
wxButton * button1 = new wxButton ( this, wxID_ANY, _ ( "button 1" ),  
wxDefaultPosition, wxSize ( 300, 100 ) );
```

- akan memiliki lebar inisial 300 pixel dan tinggi inisial 100 pixel.
- Dengan demikian, maka lebar inisial setiap sel adalah 300 pixel dan tinggi inisial setiap sel adalah 100 pixel.
- Hanya ada satu sizer yang diperlukan dalam aplikasi ini.
- Pembuatan Grid sizer dilakukan dengan perintah:

```
wxGridSizer * sizerMain = new wxGridSizer ( 3, 2, 5, 5 );
```

- yang mana akan mengandung 3 baris dan 2 kolom
- dan vgap dan hgap masing-masing adalah 5 pixel
- Penambahan window ke dalam grid sizer sama dengan cara penambahan window ke dalam Box sizer.

### 4.8.3 Kompilasi

#### **4.8.3.1 Kompilasi di Linux (GTK+)**

Makefile yang digunakan: makefile.gtk:

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006

# variables
WXCONFIG = wx-config
WXPLAT = gtk

PROGRAM = grid_sizer
OBJECTS = src/app.o src/frame.o

# system
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES:    .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all:  $(PROGRAM)

$(PROGRAM):  $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

### **4.9 Menggunakan Flex Grid Sizer (wxFlexGridSizer)**

Flex Grid Sizer (wxFlexGridSizer) adalah sizer yang melayout anak-anak windownya

dalam tabel dua dimensi dimana semua sel pada satu baris memiliki tinggi yang sama dan semua sel pada satu kolom memiliki lebar yang sama. Semua baris dan kolom tidak harus memiliki ukuran yang sama seperti pada `wxGridSizer`.

Constructor:

```
wxFlexGridSizer(int rows, int cols, int vgap, int hgap)
```

```
wxFlexGridSizer(int cols, int vgap = 0, int hgap = 0)
```

- `rows` adalah jumlah baris dalam tabel
- `cols` adalah jumlah kolom dalam tabel
- `vgap` dan `hgap` adalah space ekstra diantara anak-anak window

Secara default, baris dan kolom tidak akan `resize` untuk menyesuaikan ukuran apabila sizer `resize` (misal, ketika lebih dari ukuran minimumnya). Developer perlu secara eksplisit memanggil `wxFlexGridSizer::AddGrowableCol` ataupun `wxFlexGridSizer::AddGrowableRow`:

```
void AddGrowableCol(size_t idx, int proportion = 0)
```

```
void AddGrowableRow(size_t idx, int proportion = 0)
```

- `idx` adalah index baris atau kolom. Dimulai dari 0.
- `proportion` adalah parameter proporsi, sama dengan penjelasan pada `wxSizer::Add`, hanya, apabila proporsi di sini diberikan nilai 0, maka semua baris atau kolom akan `resize` sama besar (berbeda dengan `wxSizer::Add`, dimana apabila diberikan nilai 0, tidak `resize` sama sekali).

Untuk menyatakan bahwa kolom atau baris tertentu tidak lagi growable, gunakanlah `wxFlexGridSizer::RemoveGrowableCol` atau `wxFlexGridSizer::RemoveGrowableRow`:

```
void RemoveGrowableCol(size_t idx)
```

```
void RemoveGrowableRow(size_t idx)
```

- `idx` adalah index baris atau kolom. Dimulai dari 0.

Untuk lebih jelasnya, sebuah contoh aplikasi akan dibuat dengan layout menggunakan `wxFlexGridSizer`. Aplikasi tersebut terdiri dari satu frame, di mana di dalam frame, terdapat window-window berikut:

- Dua buah checkbox pada baris pertama

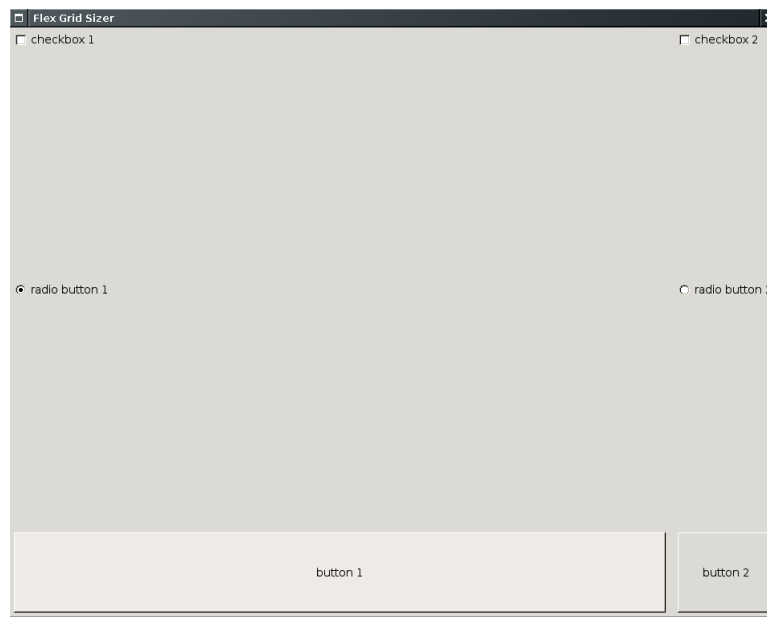
- Dua buah radio button pada baris kedua
- dan dua buah tombol pada baris ketiga

Di dalam aplikasi tersebut, hanya kolom pertama (index 0) dan baris kedua (index 1) yang growable

Untuk saat ini, tidak ada event handler yang digunakan. Aplikasi tersebut juga tidak memiliki fungsi apapun. Hal ini dimaksudkan agar pembahasan terfokus pada sizer.

Berikut ini adalah gambaran visual frame utama aplikasi:

<input type="checkbox"/> checkbox1	<input type="checkbox"/> checkbox 2
<input type="radio"/> radio button 1	<input type="radio"/> radio button 2
(button 1)	(button 2)



```
flex_grid_sizer/  
  build/  
    gtk/  
      flex_grid_sizer  
  src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
  makefile.gtk
```

### 4.9.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

Lihatlah pembahasan pada subbab sebelumnya apabila diperlukan.

#### **4.9.2.1 *src/app.h***

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

#### **4.9.2.2 *src/app.cpp***

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Flex Grid Sizer") );

    frame -> Show ( true );

    return true;
}
```

#### **4.9.2.3 *src/frame.h***

```
#ifndef _FRAME_H_
#define _FRAME_H_
```

```
class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### **4.9.2.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "wx/button.h"
#include "wx/checkbox.h"
#include "wx/radiobut.h"
#include "wx/sizer.h"
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{

    wxButton * button1 = new wxButton ( this, wxID_ANY, _ ( "button 1" ),
wxDefaultPosition, wxSize ( 300, 100 ) );

    wxButton * button2 = new wxButton ( this, wxID_ANY, _ ( "button 2" ) );

    wxCheckBox * check1 = new wxCheckBox ( this, wxID_ANY, _ ( "checkbox 1" ) );

    wxCheckBox * check2 = new wxCheckBox ( this, wxID_ANY, _ ( "checkbox 2" ) );

    wxRadioButton * radio1 = new wxRadioButton ( this, wxID_ANY, _ ( "radio button
1" ) );

    wxRadioButton * radio2 = new wxRadioButton ( this, wxID_ANY, _ ( "radio button
2" ) );
```



```
wxFlexGridSizer * sizerMain = new wxFlexGridSizer ( 3, 2, 5, 5 );

sizerMain -> AddGrowableCol ( 0 );

sizerMain -> AddGrowableRow ( 1 );


sizerMain -> Add ( check1, 1, wxEXPAND | wxALL, 5 );

sizerMain -> Add ( check2, 1, wxEXPAND | wxALL, 5 );


sizerMain -> Add ( radio1, 1, wxEXPAND | wxALL, 5 );

sizerMain -> Add ( radio2, 1, wxEXPAND | wxALL, 5 );


sizerMain -> Add ( button1, 1, wxEXPAND | wxALL, 5 );

sizerMain -> Add ( button2, 1, wxEXPAND | wxALL, 5 );


SizerAndFit ( sizerMain );

}
```

Penjelasan source code:

- Lihatlah pembahasan subbab sebelumnya apabila diperlukan
- Sebuah button yang dibuat dengan:

```
wxButton * button1 = new wxButton ( this, wxID_ANY, _ ( "button 1" ),
wxDefaultPosition, wxSize ( 300, 100 ) );
```

- akan memiliki lebar inisial 300 pixel dan tinggi inisial 100 pixel.
- Dengan demikian, maka lebar inisial setiap sel di dalam kolom dimana button tersebut diletakkan adalah 300 pixel dan tinggi inisial setiap sel dalam baris di mana button tersebut diletakkan adalah 100 pixel.

- Pembuatan Flex Grid sizer dilakukan dengan perintah:

```
wxFlexGridSizer * sizerMain = new wxFlexGridSizer ( 3, 2, 5, 5 );
```

- yang mana akan mengandung 3 baris dan 2 kolom
  - dan vgap dan hgap masing-masing adalah 5 pixel
- Penambahan window ke dalam grid sizer sama dengan cara penambahan window ke dalam Box sizer.
- Agar kolom pertama (index 0) dan baris kedua (index 1) growable, perintah-perintah tersebut diberikan:

```
sizerMain -> AddGrowableCol ( 0 );
```

```
sizerMain -> AddGrowableRow ( 1 );
```

## 4.9.3 Kompilasi

### 4.9.3.1 Kompilasi di Linux (GTK+)

Makefile yang digunakan: makefile.gtk:

```
# simple makefile for simple wx project  
# (c) Nop, 5 Feb 2006
```

```
# variables  
WXCONFIG = wx-config  
WXPLAT = gtk
```

```
PROGRAM = flex_grid_sizer  
OBJECTS = src/app.o src/frame.o
```

```
# system  
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

.SUFFIXES: .o .cpp

.cpp.o :

\$(CXX) -c `\$(WXCONFIG) --cxxflags` -o \$@ \$<

all: \$(PROGRAM)

\$(PROGRAM): \$(OBJECTS)

\$(CXX) -o \$(PROGRAM) \$(OBJECTS) `\$(WXCONFIG) --libs`

mkdir -p build/\$(WXPLAT)

mv \$(PROGRAM) build/\$(WXPLAT)

rm -f src/\*.o

clean:

rm -f src/\*.o build/\$(WXPLAT)/\$(PROGRAM)

## 4.10 Menggunakan Grid Bag Sizer (wxGridBagSizer)

Grid Bag Sizer (wxGridBagSizer) adalah sizer yang melayout anak-anak windownya dalam tabel dua dimensi seperti wxFlexGridSizer, namun item bisa ditempatkan secara eksplisit pada sel tertentu dengan wxGBPosition, dan item juga dapat merentang (span) lebih dari satu kolom dan atau baris dengan wxGBSpan.

Constructor:

wxGridBagSizer(int vgap = 0, int hgap = 0)

- vgap dan hgap adalah space ekstra diantara anak-anak window

Constructor wxGBPosition:

wxGBPosition()

wxGBPosition(int row, int col)

- row adalah baris, dimulai dari 0
- col adalah kolom, dimulai dari 0

Constructor wxGBSpan:

`wxGBSpan()`

`wxGBSpan(int rowspan, int colspan)`

- rowspan adalah rentang baris. Defaultnya adalah 1.
- colspan adalah rentang kolom. Defaultnya adalah 1.
- `wxGBSpan(1,1)` adalah nilai default, yang dimaksudkan item/anak window akan memakan tempat seluas 1 sel.

Sebagai catatan, `wxGridBagSizer` diturunkan dari `wxFlexGridSizer`. Oleh karena itu, secara default, baris dan kolom tidak akan diresize untuk menyesuaikan ukuran apabila sizer diresize (misal, ketika lebih dari ukuran minimumnya). Developer perlu secara eksplisit memanggil `wxFlexGridSizer::AddGrowableCol` ataupun `wxFlexGridSizer::AddGrowableRow`. Lihatlah pembahasan tentang `wxFlexGridSizer` untuk lebih jelasnya.

Untuk lebih jelasnya, sebuah contoh aplikasi akan dibuat dengan layout menggunakan `wxGridBagSizer`. Aplikasi tersebut terdiri dari satu frame, di mana di dalam frame, terdapat window-window berikut:

- tombol pada sel 0,0
- tombol pada sel 1,1
- tombol pada sel 3,3 merentang sampai 3,4
- tombol pada sel 4,3
- tombol pada sel 4,4

Di dalam aplikasi tersebut, hanya kolom pertama (index 0) dan baris pertama (index 0) yang growable

Untuk saat ini, tidak ada event handler yang digunakan. Aplikasi tersebut juga tidak memiliki fungsi apapun. Hal ini dimaksudkan agar pembahasan terfokus pada sizer.

Berikut ini adalah gambaran visual frame utama aplikasi:

(button 0,0 to 0,0)				
	(button 1,1 to 1,1)			

			(button 3,3 to 3,4)	
			(button 4,3 to 4,3)	(button 4,4 to 4,4)



### 4.10.1 Struktur direktori grid\_bag\_sizer

```
grid_bag_sizer/  
  build/  
    gtk/
```

```
src/
    grid_bag_sizer
    app.h
    app.cpp
    frame.h
    frame.cpp
    makefile.gtk
```

## 4.10.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

Lihatlah pembahasan pada subbab sebelumnya apabila diperlukan.

### 4.10.2.1 src/app.h

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

### 4.10.2.2 src/app.cpp

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"
```

```
DECLARE_APP ( App )
```

```
IMPLEMENT_APP ( App )
```

```
bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Grid Bag Sizer") );

    frame -> Show ( true );

    return true;
}
```

#### **4.10.2.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### **4.10.2.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "wx/button.h"
```

```
#include "wx/checkbox.h"
#include "wx/radiobut.h"
#include "wx/sizer.h"
#include "wx/gbsizer.h"
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{

    wxButton * button00 = new wxButton ( this, wxID_ANY, _ ( "button 0,0 to 0,0" ) );

    wxButton * button11 = new wxButton ( this, wxID_ANY, _ ( "button 1,1 to 1,1" ) );

    wxButton * button33_34 = new wxButton ( this, wxID_ANY, _ ( "button 3,3 to 3,4" ) );

    wxButton * button43 = new wxButton ( this, wxID_ANY, _ ( "button 4,3 to 4,3" ) );

    wxButton * button44 = new wxButton ( this, wxID_ANY, _ ( "button 4,4 to 4,4" ) );

    wxGridBagSizer * sizerMain = new wxGridBagSizer ( 10, 10 );

    sizerMain -> AddGrowablesCol ( 0 );

    sizerMain -> AddGrowablesRow ( 0 );

    sizerMain -> Add ( button00, wxGBPosition ( 0, 0 ), wxGBSpan ( 1, 1 ), wxEXPAND );

    sizerMain -> Add ( button11, wxGBPosition ( 1, 1 ) );

    sizerMain -> Add ( button33_34, wxGBPosition ( 3, 3 ), wxGBSpan ( 1, 2 ),
wxEXPAND );

    sizerMain -> Add ( button43, wxGBPosition ( 4, 3 ) );

    sizerMain -> Add ( button44, wxGBPosition ( 4, 4 ) );

    SetSizerAndFit ( sizerMain );

}
```



Penjelasan source code:

- Lihatlah pembahasan subbab sebelumnya apabila diperlukan
- Pembuatan Grid Bag sizer dilakukan dengan perintah:

```
wxGridBagSizer * sizerMain = new wxGridBagSizer ( 10, 10 );
```

- dan vgap dan hgap masing-masing adalah 10 pixel

- Untuk menambahkan item pada baris kedua (index 1) dan kolom kedua (index 1), gunakanlah wxGBPosition (1, 1):

```
sizerMain -> Add ( button11, wxGBPosition ( 1, 1 ) );
```

- Untuk menambahkan item pada baris pertama (index 0) dan kolom pertama (index 0), gunakanlah wxGBPosition (0, 0). Agar anak window dapat diexpand, berikanlah flag wxEXPAND. Agar anak window hanya merentang 1 baris dan 1 kolom (default), berikanlah wxGBSpan (1, 1):

```
sizerMain -> Add ( button00, wxGBPosition ( 0, 0 ), wxGBSpan ( 1, 1 ), wxEXPAND );
```

- Untuk menambahkan item pada baris keempat (index 3) dan kolom keempat (index 3), gunakanlah wxGBPosition (3, 3). Agar anak window dapat diexpand, berikanlah flag wxEXPAND. Agar anak window merentang 1 baris dan 2 kolom, berikanlah wxGBSpan (1, 2):

```
sizerMain -> Add ( button33_34, wxGBPosition ( 3, 3 ), wxGBSpan ( 1, 2 ), wxEXPAND );
```

## 4.10.3 Kompilasi

### 4.10.3.1 Kompilasi di Linux (GTK+)

Makefile yang digunakan: makefile.gtk:

```
# simple makefile for simple wx project  
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk

PROGRAM = grid_bag_sizer
OBJECTS = src/app.o src/frame.o

# system
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES: .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all: $(PROGRAM)

$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

## Bab 5. Berbagai class wxWidgets

Sebagai framework yang terhitung cukup komplit, wxWidgets datang dengan sangat banyak class untuk memudahkan developer membangun aplikasi. Terdapat berbagai class yang berhubungan dengan GUI, networking, multi threading, database, dan lain sebagainya.

Bab ini akan membahas berbagai class yang tersedia di wxWidgets. Setelah membaca bab ini, diharapkan developer dapat menggunakan berbagai class yang tersedia untuk membantu menghadirkan aplikasi yang diinginkan.

Pembahasan class akan dilakukan berdasarkan kategori. Sebagai catatan, contoh untuk berbagai class juga dapat dibaca pada bab-bab lainnya.

### 5.1 Managed Window

Berbagai class window yang dimanage secara langsung oleh window manager.

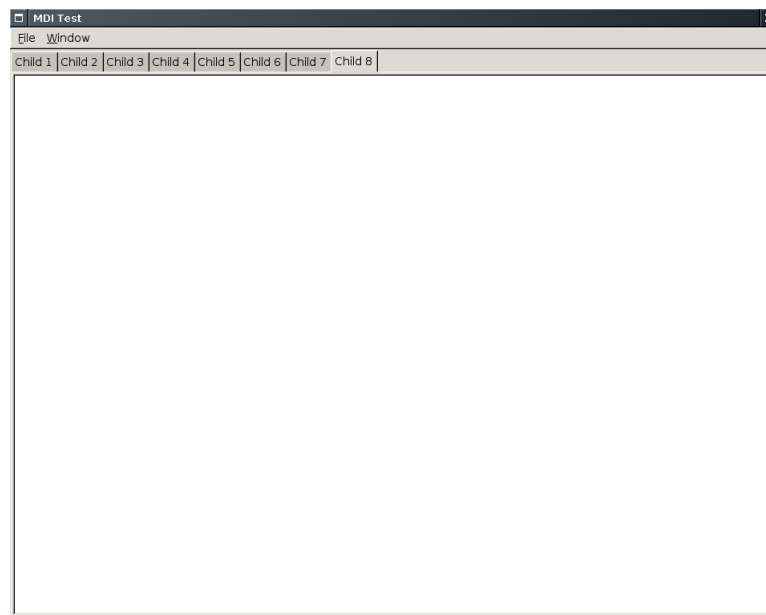
<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxTopLevelWindow	Top level window, dialog atau frame
wxDialog	Dialog box
wxFrame	Frame normal
wxMDIChildFrame	Frame MDI child
wxMDIParentFrame	Frame MDI parent
wxMiniFrame	Frame dengan title bar berukuran kecil
wxSplashScreen	Splash screen
wxPropertySheetDialog	Dialog Property sheet
wxTipWindow	Menampilkan tip di dalam window kecil
wxWizard	wizard

#### 5.1.1 wxMDIParentFrame dan wxMDIChildFrame

- wxMDIParentFrame
  - include files:
    - wx/mdi.h
  - derived from:
    - wxFrame
    - wxWindow
    - wxEvtHandler
    - wxObject
- wxMDIChildFrame
  - include files:
    - wx/mdi.h
  - derived from:
    - wxFrame
    - wxWindow
    - wxEvtHandler
    - wxObject

#### 5.1.1.1 MDI Test

Aplikasi demo untuk bekerja dengan wxMDIParentFrame dan wxMDIChildFrame. User bisa membuka window baru, menutup window yang ada, berpindah ke window tertentu dengan bantuan menu dan lain sebagainya.



##### 5.1.1.1.1 Struktur direktori

```
MDI/0_mdi/  
    build/  
        gtk/  
            mdi  
src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
makefile.gtk
```

#### 5.1.1.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

##### 5.1.1.1.2.1 src/app.h

```
#ifndef _APP_H_  
#define _APP_H_  
  
class App : public wxApp  
{  
public:  
  
    virtual bool OnInit ();  
};  
  
#endif // _APP_H_
```

##### 5.1.1.1.2.2 src/app.cpp

```
#include "wx/wx.h"
```

```
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("MDI Test") );

    frame -> Show ( true );

    return true;
}
```

#### 5.1.1.1.2.3 src/frame.h

```
#ifndef _FRAME_H_
#define _FRAME_H_

enum
{
    ID_MDI_PREV = wxID_HIGHEST + 1,

    ID_MDI_NEXT,

    ID_MDI_CLOSE
};
```

```
class Frame: public wxMDIParentFrame
```

```
{
public:

    Frame ( const wxString& title );

    wxMenu * m_windowMenu;

    int m_startWindowId;

    int m_startMenuId;

    long m_windowCount;

    long m_currentWindowCount;

    void OnNew ( wxCommandEvent& event );
    void OnQuit ( wxCommandEvent& event );
    void OnCloseChild ( wxCommandEvent& event );
    void OnSelectChild ( wxCommandEvent& event );
    void OnSelectPrev ( wxCommandEvent& event );
    void OnSelectNext ( wxCommandEvent& event );

private:

    DECLARE_EVENT_TABLE ()

};

#endif // _FRAME_H_
```

#### **5.1.1.1.2.4 src/frame.cpp**

```
#include "wx/wx.h"
```

```
#include "frame.h"
```

```
BEGIN_EVENT_TABLE ( Frame, wxMDIParentFrame )
```

```
    EVT_MENU ( wxID_NEW, Frame :: OnNew )
```

```
    EVT_MENU ( wxID_EXIT, Frame :: OnQuit )
```

```
    EVT_MENU ( ID_MDI_CLOSE, Frame :: OnCloseChild )
```

```
    EVT_MENU ( ID_MDI_PREV, Frame :: OnSelectPrev )
```

```
    EVT_MENU ( ID_MDI_NEXT, Frame :: OnSelectNext )
```

```
END_EVENT_TABLE ()
```

```
Frame :: Frame ( const wxString& title ) : wxMDIParentFrame ( NULL, wxID_ANY, title )  
{
```

```
    wxMenu * fileMenu = new wxMenu;
```

```
    fileMenu -> Append ( wxID_NEW, _( "&New \tCTRL-N" ), _( "New child window" ) );
```

```
    fileMenu -> Append ( wxID_EXIT, _( "&Quit \tCTRL-Q" ), _( "Quit application" ) );
```

```
    m_windowMenu = new wxMenu;
```

```
    m_windowMenu -> Append ( ID_MDI_CLOSE, _( "&Close \tCTRL-W" ), _( "Close  
active child" ) );
```

```
    m_windowMenu -> AppendSeparator ();
```

```
    m_windowMenu -> Append ( ID_MDI_PREV, _( "Select &Prev \tF3" ), _( "Select  
previous child" ) );
```

```
    m_windowMenu -> Append ( ID_MDI_NEXT, _( "Select &Next \tF4" ), _( "Select next  
child" ) );
```

```
    m_windowMenu -> AppendSeparator ();
```

```
    wxMenuBar * menuBar = new wxMenuBar;
```



```
menuBar -> Append ( fileMenu, _ ( "&File" ) );

menuBar -> Append ( m_windowMenu, _ ( "&Window" ) );

SetMenuBar ( menuBar );


m_windowCount = 0;

m_currentWindowCount = m_windowCount;


m_startWindowId = 10000;

m_startMenuId = 20000;

}


void Frame :: OnNew ( wxCommandEvent& event )
{

    m_windowCount++;

    m_currentWindowCount++;

    wxString tempStr;

    tempStr.Printf ( _ ( "Child %ld" ), m_windowCount );

    wxMDIChildFrame * child = new wxMDIChildFrame ( this, m_startWindowId +
m_windowCount, tempStr );

    child -> Show ( true );

    child -> Activate ();

    wxTextCtrl * text = new wxTextCtrl ( child, wxID_ANY );

    m_windowMenu -> Append ( m_startMenuId + m_windowCount, tempStr,
tempStr );
```

```
    Connect ( m_startMenuId + m_windowCount, wxEVT_COMMAND_MENU_SELECTED,  
wxCommandEventHandler ( Frame :: OnSelectChild ) );
```

```
}
```

```
void Frame :: OnQuit ( wxCommandEvent& event )
```

```
{
```

```
    Close ();
```

```
}
```

```
void Frame :: OnCloseChild ( wxCommandEvent& event )
```

```
{
```

```
    if ( m_currentWindowCount > 0 )
```

```
    {
```

```
        wxMDIChildFrame * curr = (wxMDIChildFrame * ) FindWindow ( this ->  
GetActiveChild () -> GetId () );
```

```
        m_currentWindowCount--;
```

```
        curr -> Destroy ();
```

```
        m_windowMenu -> Delete ( (m_startMenuId - m_startWindowId) + curr -> GetId  
( ) );
```

```
        GetClientWindow() -> Refresh ();
```

```
        GetClientWindow() -> Update ();
```

```
    }
```

```
}
```

```
void Frame :: OnSelectPrev ( wxCommandEvent& event )
```

```
{  
  
    if (m_currentWindowCount > 0)  
    {  
  
        ActivatePrevious ();  
  
    }  
}  
  
void Frame :: OnSelectNext ( wxCommandEvent& event )  
{  
    if (m_currentWindowCount > 0)  
    {  
  
        ActivateNext ();  
  
    }  
}  
  
void Frame :: OnSelectChild ( wxCommandEvent& event )  
{  
  
    wxMDIChildFrame * temp = (wxMDIChildFrame *) FindWindow ( event.GetId () -  
m_startWindowId );  
  
    temp -> Activate ();  
  
}
```

### **5.1.1.1.3 Kompilasi**

#### **5.1.1.1.3.1 Kompilasi di Linux (GTK+)**

```
makefile.gtk:  
# simple makefile for simple wx project  
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk

PROGRAM = mdi
OBJECTS = src/app.o src/frame.o

# system
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES: .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all: $(PROGRAM)

$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

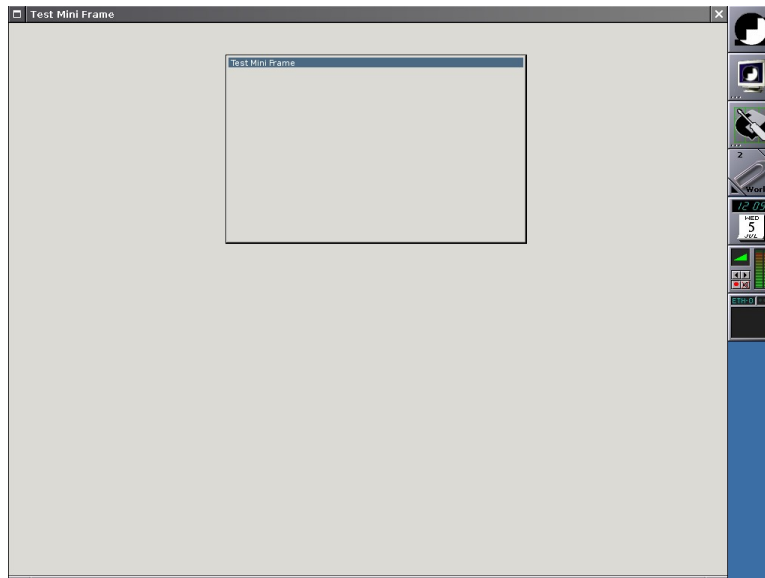
## 5.1.2 wxMiniFrame

- wxMiniFrame
  - include files:
    - wx/minifram.h
  - derived from:
    - wxFrame
    - wxWindow

- wxEvtHandler
- wxObject

### 5.1.2.1 Test Mini Frame

Aplikasi demo untuk bekerja dengan wxMiniFrame. Pada saat frame utama aplikasi ditampilkan, sebuah miniframe juga akan ikut tampil.



#### 5.1.2.1.1 Struktur direktori

```
miniframe/0_miniframe/  
    build/  
        gtk/  
            miniframe  
    src/  
        app.h  
        app.cpp  
        frame.h  
        frame.cpp  
    makefile.gtk
```

#### 5.1.2.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App

- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

#### **5.1.2.1.2.1 src/app.h**

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

#### **5.1.2.1.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Test Mini Frame") );

    frame -> Show ( true );

    return true;
}
```

**5.1.2.1.2.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

**5.1.2.1.2.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "wx/minifram.h"

Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{

    wxMiniFrame * miniFrame = new wxMiniFrame ( this, wxID_ANY, title );

    miniFrame -> Show ( true );
```

```
}
```

### 5.1.2.1.3 Kompilasi

#### 5.1.2.1.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:
```

```
# simple makefile for simple wx project
```

```
# (c) Nop, 5 Feb 2006
```

```
# variables
```

```
WXCONFIG = wx-config
```

```
WXPLAT = gtk
```

```
PROGRAM = miniframe
```

```
OBJECTS = src/app.o src/frame.o
```

```
# system
```

```
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES: .o .cpp
```

```
.cpp.o :
```

```
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<
```

```
all: $(PROGRAM)
```

```
$(PROGRAM): $(OBJECTS)
```

```
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
```

```
    mkdir -p build/$(WXPLAT)
```

```
    mv $(PROGRAM) build/$(WXPLAT)
```

```
    rm -f src/*.o
```

```
clean:
```

```
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

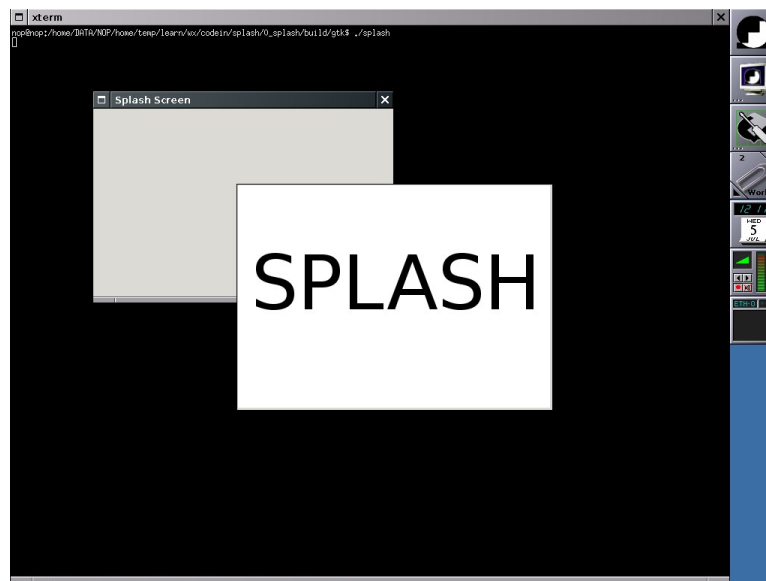


### 5.1.3 wxSplashScreen

- wxSplashScreen
  - include files:
    - wx/splash.h
  - derived from:
    - wxFrame
    - wxWindow
    - wxEvtHandler
    - wxObject

#### 5.1.3.1 *Splash Screen*

Aplikasi demo untuk bekerja dengan wxSplashScreen. Pada saat aplikasi dijalankan, sebuah splash screen akan ditampilkan selama dua detik.



#### 5.1.3.1.1 Struktur direktori

```
splash/0_splash/  
  build/  
    gtk/  
      splash  
      splash.png  
  src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
  makefile.gtk
```

#### 5.1.3.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

Sebuah file splash.png akan digunakan sebagai gambar di splash screen.

**5.1.3.1.2.1 src/app.h**

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

**5.1.3.1.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "wx/splash.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    wxBitmap bitmap;

    if ( bitmap.LoadFile ( _("splash.png") ) )
    {
        wxSplashScreen * splash = new wxSplashScreen ( bitmap,
wxSPLASH_CENTRE_ON_SCREEN | wxSPLASH_TIMEOUT, 2000, NULL, wxID_ANY,
```

```
        wxDefaultPosition, wxDefaultSize, wxSIMPLE_BORDER | wxSTAY_ON_TOP );
    }

    wxYield ();

    Frame * frame = new Frame ( _ ("Splash Screen") );

    frame -> Show ( true );

    return true;
}
```

#### **5.1.3.1.2.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### **5.1.3.1.2.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{
}
}
```

### 5.1.3.1.3 Kompilasi

#### 5.1.3.1.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006

# variables
WXCONFIG = wx-config
WXPLAT = gtk

PROGRAM = splash
OBJECTS = src/app.o src/frame.o

# system
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES: .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all: $(PROGRAM)

$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
```

```
mv $(PROGRAM) build/$(WXPLAT)
rm -f src/*.o
```

clean:

```
rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

## 5.1.4 wxTipWindow dan Tip Provider

- wxTipWindow
  - include files:
    - wx/tipwin.h
  - derived from:
    - wxPopupTransientWindow
    - wxPopupWindow
    - wxWindow
    - wxEvtHandler
    - wxObject

### 5.1.4.1 Tip Window

Aplikasi demo untuk bekerja dengan wxTipWindow. Pada saat aplikasi dijalankan, sebuah tip kecil akan ditampilkan.

#### 5.1.4.1.1 Struktur direktori

```
tip/1_tipwin/
  build/
    gtk/
      tip
  src/
    app.h
    app.cpp
    frame.h
```

```
frame.cpp  
makefile.gtk
```

#### 5.1.4.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

##### 5.1.4.1.2.1 src/app.h

```
#ifndef _APP_H_  
#define _APP_H_  
  
class App : public wxApp  
{  
public:  
  
    virtual bool OnInit ();  
};  
  
#endif // _APP_H_
```

##### 5.1.4.1.2.2 src/app.cpp

```
#include "wx/wx.h"  
#include "frame.h"  
#include "app.h"  
  
DECLARE_APP ( App )  
  
IMPLEMENT_APP ( App )
```

```
bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Tip Window") );

    frame -> Show ( true );

    return true;
}
```

#### **5.1.4.1.2.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### **5.1.4.1.2.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "wx/tipwin.h"
```



```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{

    wxTipWindow * tip = new wxTipWindow ( this, _( "Tip 1" ), 200 );

}
```

### 5.1.4.1.3 Kompilasi

#### 5.1.4.1.3.1 Kompilasi di Linux (GTK+)

makefile.gtk:

# simple makefile for simple wx project  
# (c) Nop, 5 Feb 2006

# variables

WXCONFIG = wx-config

WXPLAT = gtk

PROGRAM = tip

OBJECTS = src/app.o src/frame.o

# system

CXX = \$(shell \$(WXCONFIG) --cxx)

# implementation

.SUFFIXES: .o .cpp

.cpp.o :

\$(CXX) -c \$(WXCONFIG) --cxxflags` -o \$@ \$<

all: \$(PROGRAM)

\$(PROGRAM): \$(OBJECTS)

\$(CXX) -o \$(PROGRAM) \$(OBJECTS) \$(WXCONFIG) --libs`

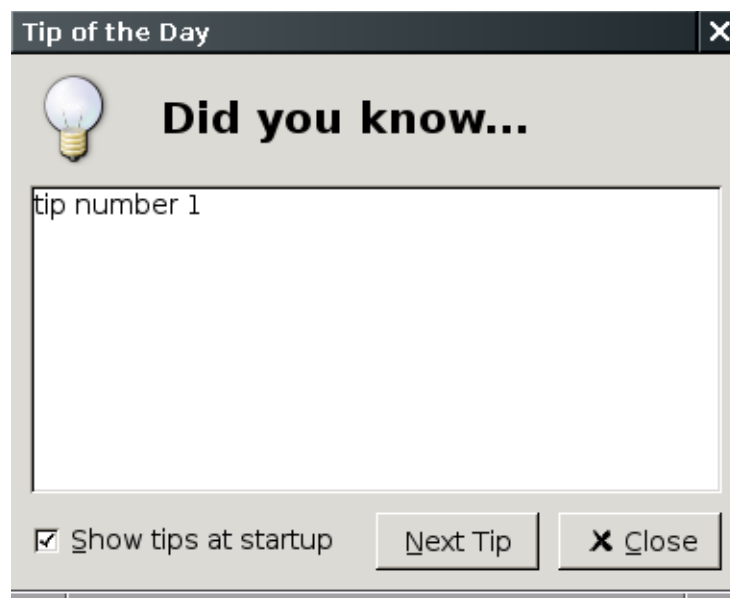
```
mkdir -p build/$(WXPLAT)
mv $(PROGRAM) build/$(WXPLAT)
rm -f src/*.o
```

clean:

```
rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

#### 5.1.4.2 Tip Provider

Aplikasi demo untuk bekerja dengan tip provider. Pada saat aplikasi dijalankan, sebuah tip of the day akan ditampilkan.



##### 5.1.4.2.1 Struktur direktori

```
tip/0_tipdlg/
    build/
        gtk/
            tip
            tips.txt
```

```
src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
makefile.gtk
```

#### 5.1.4.2.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

Sebuah file tips.txt akan digunakan sebagai database tips of the day. Isi file tips.txt:

```
_("tip number 1 ")  
_("tip number 2 ")  
_("tip number 3 ")  
_("tip number 4 ")  
_("tip number 5 ")
```

##### 5.1.4.2.2.1 src/app.h

```
#ifndef _APP_H_  
#define _APP_H_  
  
class App : public wxApp  
{  
public:  
  
    virtual bool OnInit ();  
};  
  
#endif // _APP_H_
```

**5.1.4.2.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Tip Provider") );

    frame -> Show ( true );

    return true;
}
```

**5.1.4.2.2.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### 5.1.4.2.2.4 src/frame.cpp

```
#include "wx/wx.h"
#include "frame.h"
#include "wx/tipdlg.h"

Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{

    wxTipProvider * tipProvider = wxCreateFileTipProvider ( _( "tips.txt" ), 0 );

    wxShowTip ( this, tipProvider );

    delete tipProvider;

}
```

#### 5.1.4.2.3 Kompilasi

##### 5.1.4.2.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk
```

```
PROGRAM = tip
OBJECTS = src/app.o src/frame.o
```

```
# system
```

```
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES:    .o .cpp
```

```
.cpp.o :
```

```
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<
```

```
all:  $(PROGRAM)
```

```
$(PROGRAM):  $(OBJECTS)
```

```
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
```

```
    mkdir -p build/$(WXPLAT)
```

```
    mv $(PROGRAM) build/$(WXPLAT)
```

```
    rm -f src/*.o
```

```
clean:
```

```
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

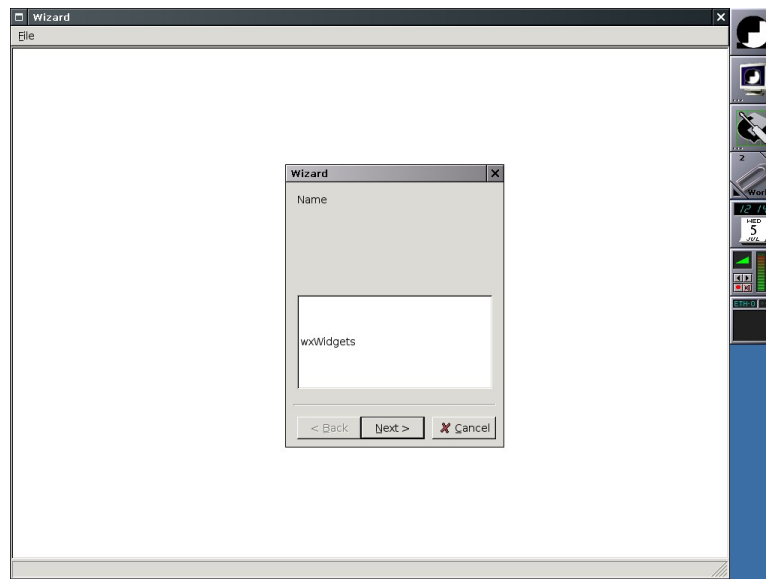
## 5.1.5 wxWizard

- wxWizard
  - include files:
    - wx/wizard.h
  - derived from:
    - wxDialog
    - wxPanel
    - wxWindow
    - wxEvtHandler
    - wxObject
- wxWizardPageSimple
  - include files:
    - wx/wizard.h
  - derived from:
    - wxWizardPage
    - wxPanel
    - wxWindow
    - wxEvtHandler

- wxObject

### 5.1.5.1 Wizard

Aplikasi demo untuk bekerja dengan wxWizard. Sebuah aplikasi dengan frame utama berisikan menu dan text control multiline ditampilkan. User dapat mengakses menu untuk menampilkan wizard. Ketika wizard selesai dieksekusi, text control tersebut akan menampilkan informasi yang diisikan oleh user.



#### 5.1.5.1.1 Struktur direktori

```
wizard/  
  build/  
    gtk/  
      wizard  
  src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
  makefile.gtk
```

#### 5.1.5.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

#### **5.1.5.1.2.1 src/app.h**

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

#### **5.1.5.1.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "textctrl.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Wizard" ) );
```



```
    frame -> Show ( true );  
  
    return true;  
}
```

#### 5.1.5.1.2.3 src/frame.h

```
#ifndef _FRAME_H_  
#define _FRAME_H_  
  
enum  
{  
  
    ID_TEXTCTRL = wxID_HIGHEST + 1,  
  
    ID_WIZARD,  
  
    ID_NAME,  
  
    ID_ADDR,  
  
    ID_NOTE  
  
};  
  
class Frame: public wxFrame  
{  
public:  
  
    Frame ( const wxString& title );  
  
    void OnQuit ( wxCommandEvent& event );  
  
    void OnSettings ( wxCommandEvent& event );  
};
```

private:

DECLARE\_EVENT\_TABLE ()

};

#endif // \_FRAME\_H\_

#### **5.1.5.1.2.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "wx/wizard.h"
#include "frame.h"
#include "textctrl.h"
```

BEGIN\_EVENT\_TABLE ( Frame, wxFrame )

EVT\_MENU ( wxID\_EXIT, Frame :: OnQuit )

EVT\_MENU ( wxID\_NEW, Frame :: OnSettings )

END\_EVENT\_TABLE ()

Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID\_ANY, title,  
wxDefaultPosition, wxSize ( 640, 480 ) )

{

Centre ();

wxMenu \* fileMenu = new wxMenu;

fileMenu -> Append ( wxID\_NEW, \_( "&Settings \tF2" ), \_( "Settings" ) );

fileMenu -> Append ( wxID\_EXIT, \_( "&Quit \tCTRL-Q" ), \_( "Quit application" ) );

wxMenuBar \* menuBar = new wxMenuBar;

```
menuBar -> Append ( fileMenu, _( "&File" ) );
```

```
SetMenuBar ( menuBar );
```

```
    wxTextCtrl * textCtrl = new wxTextCtrl ( this, ID_TEXTCTRL, wxEmptyString,  
wxDefaultPosition,  
    wxDefaultSize, wxTE_MULTILINE | wxTE_READONLY | wxTE_DONTWRAP );
```

```
CreateStatusBar ();
```

```
}
```

```
void Frame :: OnQuit ( wxCommandEvent& event )
```

```
{
```

```
    Close ();
```

```
}
```

```
void Frame :: OnSettings ( wxCommandEvent& event )
```

```
{
```

```
    wxWizard * wizard = new wxWizard ( this, ID_WIZARD, _( "Wizard" ) );
```

```
    wxWizardPageSimple * page1 = new wxWizardPageSimple ( wizard );
```

```
    wxBoxSizer * sizer1 = new wxBoxSizer ( wxVERTICAL );
```

```
    wxStaticText * lblName = new wxStaticText ( page1, wxID_ANY, _( "Name" ) );
```

```
    wxTextCtrl * textName = new wxTextCtrl ( page1, ID_NAME );
```

```
    sizer1 -> Add ( lblName, 1, wxEXPAND | wxALL, 5 );
```

```
    sizer1 -> Add ( textName, 1, wxEXPAND | wxALL, 5 );
```

```
    page1 -> SetSizer ( sizer1 );
```

```
wxWizardPageSimple * page2 = new wxWizardPageSimple ( wizard );  
  
wxBoxSizer * sizer2 = new wxBoxSizer ( wxVERTICAL );  
  
wxStaticText * lblAddr = new wxStaticText ( page2, wxID_ANY, _( "Address" ) );  
  
wxTextCtrl * textAddr = new wxTextCtrl ( page2, ID_ADDR );  
  
sizer2 -> Add ( lblAddr, 1, wxEXPAND | wxALL, 5 );  
  
sizer2 -> Add ( textAddr, 1, wxEXPAND | wxALL, 5 );  
  
page2 -> SetSizer ( sizer2 );
```

```
wxWizardPageSimple * page3 = new wxWizardPageSimple ( wizard );  
  
wxBoxSizer * sizer3 = new wxBoxSizer ( wxVERTICAL );  
  
wxStaticText * lblNote = new wxStaticText ( page3, wxID_ANY, _( "Note" ) );  
  
wxTextCtrl * textNote = new wxTextCtrl ( page3, ID_NOTE );  
  
sizer3 -> Add ( lblNote, 1, wxEXPAND | wxALL, 5 );  
  
sizer3 -> Add ( textNote, 1, wxEXPAND | wxALL, 5 );  
  
page3 -> SetSizer ( sizer3 );
```

```
page1 -> SetNext ( page2 );  
  
page2 -> SetPrev ( page1 );  
  
page2 -> SetNext ( page3 );  
  
page3 -> SetPrev ( page2 );
```

```
wxString entry_name;
```

```
wxString entry_addr;
```

```
wxString entry_note;

FindWindow ( ID_NAME ) -> SetValidator ( wxTextValidator ( wxFILTER_ALPHA,
&entry_name ) );

FindWindow ( ID_ADDR ) -> SetValidator ( wxTextValidator ( wxFILTER_ALPHA,
&entry_addr ) );

FindWindow ( ID_NOTE ) -> SetValidator ( wxTextValidator ( wxFILTER_ALPHA,
&entry_note ) );

wizard -> RunWizard ( page1 );

wizard -> Destroy ();

wxString msg;

msg.Printf ( _( "Information from wizard:\nName: %s\nAddress: %s\nNote: %s\n" ),
            entry_name.c_str (), entry_addr.c_str (), entry_note.c_str () );

wxTextCtrl * textCtrl = (wxTextCtrl * ) FindWindow ( ID_TEXTCTRL );

textCtrl -> SetValue ( msg );

}
```

### 5.1.5.1.3 Kompilasi

#### 5.1.5.1.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk

PROGRAM = wizard
OBJECTS = src/app.o src/frame.o

# system
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES: .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all: $(PROGRAM)

$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

## 5.2 Window lain-lain

Berbagai class yang diturunkan dari wxWindow

<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxPanel	Panel
wxScrolledWindow	Window scroll
wxGrid	Grid

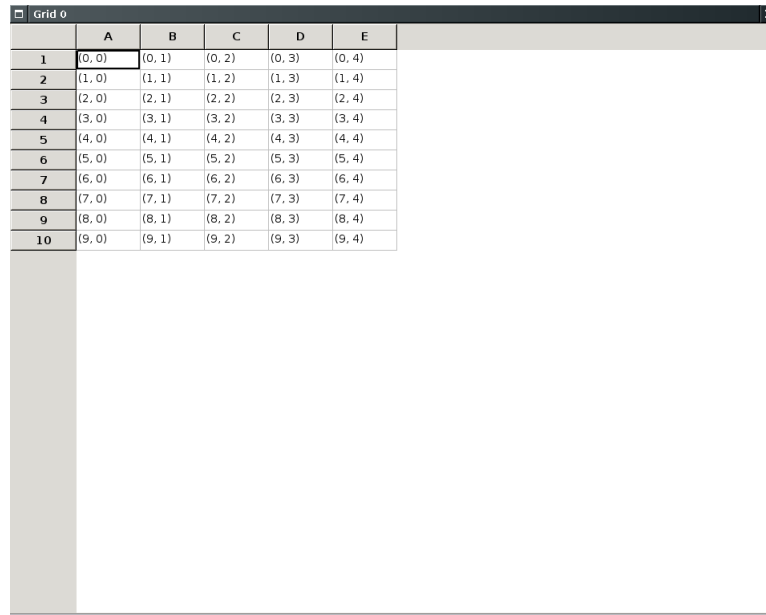
<b><i>Class</i></b>	<b><i>Deskripsi</i></b>
wxSplitterWindow	Window yang dapat displit secara vertical atau horizontal
wxStatusBar	Statusbar
wxToolBar	toolbar
wxNotebook	notebook
wxListbook	Mirip dengan notebook, namun menggunakan list control
wxChoicebook	Mirip dengan notebook, namun menggunakan choice control
wxSashWindow	Window dengan empat sash (ikat pinggang) yang bisa didrag
wxSashLayoutWindow	Sash Layout
wxVScrolledWindow	Seperti wxScrolledWindow, namun mendukung baris dengan tinggi variabel
wxWizardPage	Base class untuk halaman pada wizard
wxWizardPageSimple	Halaman pada wizard

### 5.2.1 wxGrid

- wxGrid
  - include files:
    - wx/grid.h
  - derived from:
    - wxScrolledWindow
    - wxWindow
    - wxEvtHandler
    - wxObject

#### 5.2.1.1 Grid 0

Aplikasi demo untuk bekerja dengan wxGrid. Contoh penggunaan wxGrid :: SetCellValue().



	A	B	C	D	E
1	(0, 0)	(0, 1)	(0, 2)	(0, 3)	(0, 4)
2	(1, 0)	(1, 1)	(1, 2)	(1, 3)	(1, 4)
3	(2, 0)	(2, 1)	(2, 2)	(2, 3)	(2, 4)
4	(3, 0)	(3, 1)	(3, 2)	(3, 3)	(3, 4)
5	(4, 0)	(4, 1)	(4, 2)	(4, 3)	(4, 4)
6	(5, 0)	(5, 1)	(5, 2)	(5, 3)	(5, 4)
7	(6, 0)	(6, 1)	(6, 2)	(6, 3)	(6, 4)
8	(7, 0)	(7, 1)	(7, 2)	(7, 3)	(7, 4)
9	(8, 0)	(8, 1)	(8, 2)	(8, 3)	(8, 4)
10	(9, 0)	(9, 1)	(9, 2)	(9, 3)	(9, 4)

#### 5.2.1.1.1 Struktur direktori

```
grid/0_grid/  
    build/  
        gtk/  
            grid  
src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
makefile.gtk
```

#### 5.2.1.1.2 Source code

Source code akan melibatkan empat file berikut:



- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

#### **5.2.1.1.2.1 src/app.h**

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

#### **5.2.1.1.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Grid 0") );

    frame -> Show ( true );
```

```
    return true;
}
```

#### **5.2.1.1.2.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### **5.2.1.1.2.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "wx/grid.h"
#include "frame.h"

Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{
```

```
wxGrid * grid = new wxGrid ( this, wxID_ANY );

int initRows = 10;

int initCols = 5;

grid -> CreateGrid ( initRows, initCols );

int row;

int col;

for ( row = 0; row < initRows; row ++ )
{
    for ( col = 0; col < initCols ; col ++ )
    {
        wxString temp;

        temp.Printf ( _( "%d, %d" ), row, col );

        grid -> SetCellValue ( row, col, temp );
    }
}
}
```

### 5.2.1.1.3 Kompilasi

#### 5.2.1.1.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
```

```
WXCONFIG = wx-config
```

```
WXPLAT = gtk
```

```
PROGRAM = grid
```

```
OBJECTS = src/app.o src/frame.o
```

```
# system
```

```
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES: .o .cpp
```

```
.cpp.o :
```

```
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<
```

```
all: $(PROGRAM)
```

```
$(PROGRAM): $(OBJECTS)
```

```
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
```

```
    mkdir -p build/$(WXPLAT)
```

```
    mv $(PROGRAM) build/$(WXPLAT)
```

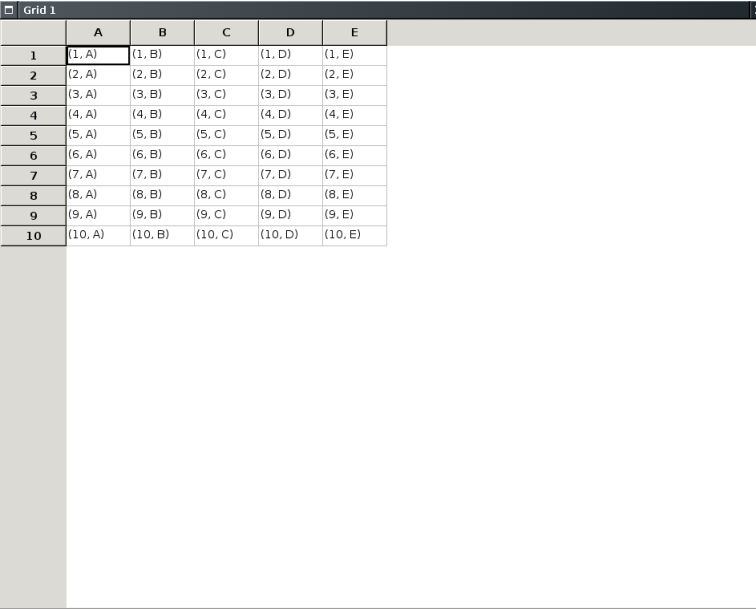
```
    rm -f src/*.o
```

```
clean:
```

```
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

### **5.2.1.2 Grid 1**

Aplikasi demo untuk bekerja dengan wxGrid. Contoh penggunaan wxGrid :: GetColLabelValue().



	A	B	C	D	E
1	(1, A)	(1, B)	(1, C)	(1, D)	(1, E)
2	(2, A)	(2, B)	(2, C)	(2, D)	(2, E)
3	(3, A)	(3, B)	(3, C)	(3, D)	(3, E)
4	(4, A)	(4, B)	(4, C)	(4, D)	(4, E)
5	(5, A)	(5, B)	(5, C)	(5, D)	(5, E)
6	(6, A)	(6, B)	(6, C)	(6, D)	(6, E)
7	(7, A)	(7, B)	(7, C)	(7, D)	(7, E)
8	(8, A)	(8, B)	(8, C)	(8, D)	(8, E)
9	(9, A)	(9, B)	(9, C)	(9, D)	(9, E)
10	(10, A)	(10, B)	(10, C)	(10, D)	(10, E)

#### 5.2.1.2.1 Struktur direktori

```
grid/1_grid/  
    build/  
        gtk/  
            grid  
src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
makefile.gtk
```

#### 5.2.1.2.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

##### 5.2.1.2.2.1 src/app.h

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

#### **5.2.1.2.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Grid 1") );

    frame -> Show ( true );

    return true;
}
```

#### 5.2.1.2.2.3 src/frame.h

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### 5.2.1.2.2.4 src/frame.cpp

```
#include "wx/wx.h"
#include "wx/grid.h"
#include "frame.h"

Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{

    wxGrid * grid = new wxGrid ( this, wxID_ANY );

    int initRows = 10;

    int initCols = 5;
```

```
grid -> CreateGrid ( initRows, initCols );

int row;

int col;

for ( row = 0; row < initRows; row ++ )
{
    for ( col = 0; col < initCols ; col ++ )
    {
        wxString temp;

        temp.Printf ( _( "%d, %s" ), row + 1, grid -> GetColLabelValue ( col ).c_str () );

        grid -> SetCellValue ( row, col, temp );
    }
}
}
```

### 5.2.1.2.3 Kompilasi

#### 5.2.1.2.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk
```

```
PROGRAM = grid
```



```
OBJECTS = src/app.o src/frame.o
```

```
# system
```

```
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES:    .o .cpp
```

```
.cpp.o :
```

```
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<
```

```
all:  $(PROGRAM)
```

```
$(PROGRAM):  $(OBJECTS)
```

```
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
```

```
    mkdir -p build/$(WXPLAT)
```

```
    mv $(PROGRAM) build/$(WXPLAT)
```

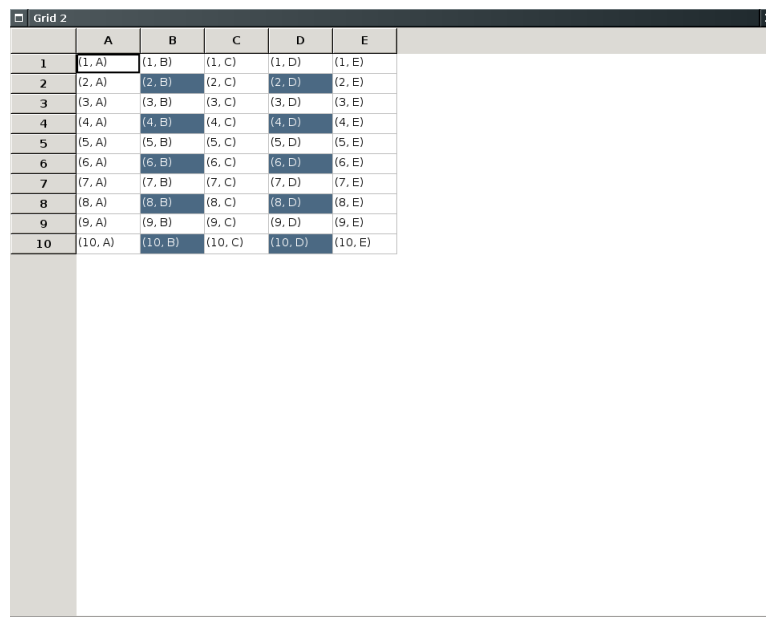
```
    rm -f src/*.o
```

```
clean:
```

```
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

### **5.2.1.3 Grid 2**

Aplikasi demo untuk bekerja dengan wxGrid. Contoh penggunaan wxGrid :: SelectBlock().



	A	B	C	D	E
1	(1, A)	(1, B)	(1, C)	(1, D)	(1, E)
2	(2, A)	(2, B)	(2, C)	(2, D)	(2, E)
3	(3, A)	(3, B)	(3, C)	(3, D)	(3, E)
4	(4, A)	(4, B)	(4, C)	(4, D)	(4, E)
5	(5, A)	(5, B)	(5, C)	(5, D)	(5, E)
6	(6, A)	(6, B)	(6, C)	(6, D)	(6, E)
7	(7, A)	(7, B)	(7, C)	(7, D)	(7, E)
8	(8, A)	(8, B)	(8, C)	(8, D)	(8, E)
9	(9, A)	(9, B)	(9, C)	(9, D)	(9, E)
10	(10, A)	(10, B)	(10, C)	(10, D)	(10, E)

### 5.2.1.3.1 Struktur direktori

```
grid/2_grid/  
    build/  
        gtk/  
            grid  
src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
    makefile.gtk
```

### 5.2.1.3.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

#### 5.2.1.3.2.1 src/app.h

```
#ifndef _APP_H_
```

```
#define _APP_H

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

#### 5.2.1.3.2.2 src/app.cpp

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Grid 2") );

    frame -> Show ( true );

    return true;
}
```

#### 5.2.1.3.2.3 src/frame.h

```
#ifndef _FRAME_H_
#define _FRAME_H_
```

```
class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### **5.2.1.3.2.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "wx/grid.h"
#include "frame.h"

Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{

    wxGrid * grid = new wxGrid ( this, wxID_ANY );

    int initRows = 10;

    int initCols = 5;

    grid -> CreateGrid ( initRows, initCols );

    int row;

    int col;

    for ( row = 0; row < initRows; row ++ )
    {

        for ( col = 0; col < initCols ; col ++ )
        {

            wxString temp;
```

```
temp.Printf ( _ ( "%d, %s" ), row + 1, grid -> GetColLabelValue ( col ).c_str () );

grid -> SetCellValue ( row, col, temp );

if ( col % 2 == 1 && row % 2 == 1 )
{
    grid -> SelectBlock ( row, col, row, col, true );
}

}

}

}
```

### 5.2.1.3.3 Kompilasi

#### 5.2.1.3.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk
```

```
PROGRAM = grid
OBJECTS = src/app.o src/frame.o
```

```
# system
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES:    .o .cpp
```

```
.cpp.o :
```

```
$(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<
```

```
all: $(PROGRAM)
```

```
$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o
```

```
clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

### 5.2.1.4 Grid 3

Aplikasi demo untuk bekerja dengan wxGrid. Contoh penggunaan event. Klik gandalah pada judul kolom.

#### 5.2.1.4.1 Struktur direktori

```
grid/3_grid/
  build/
    gtk/
      grid
  src/
    app.h
    app.cpp
    frame.h
    frame.cpp
  makefile.gtk
```

#### 5.2.1.4.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

**5.2.1.4.2.1 src/app.h**

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

**5.2.1.4.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Grid 3") );

    frame -> Show ( true );

    return true;
}
```

#### 5.2.1.4.2.3 src/frame.h

```
#include "wx/grid.h"

#ifndef _FRAME_H_
#define _FRAME_H_

enum
{
    ID_GRID = wxID_HIGHEST + 1
};

class Frame: public wxFrame
{
public:
    Frame ( const wxString& title );

    void OnColAutoSize ( wxGridEvent& event );

private:
    DECLARE_EVENT_TABLE ()
};

#endif // _FRAME_H_
```

#### 5.2.1.4.2.4 src/frame.cpp

```
#include "wx/wx.h"
#include "wx/grid.h"
```



```
#include "frame.h"
```

```
BEGIN_EVENT_TABLE ( Frame, wxFrame )
```

```
    EVT_GRID_CMD_LABEL_LEFT_DCLICK ( ID_GRID, Frame :: OnColAutoSize )
```

```
END_EVENT_TABLE ()
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )  
{
```

```
    wxGrid * grid = new wxGrid ( this, ID_GRID );
```

```
    int initRows = 10;
```

```
    int initCols = 5;
```

```
    grid -> CreateGrid ( initRows, initCols );
```

```
    int row;
```

```
    int col;
```

```
    for ( row = 0; row < initRows; row ++ )  
    {
```

```
        for ( col = 0; col < initCols ; col ++ )  
        {
```

```
            wxString temp;
```

```
            temp.Printf ( _( "%d, %s" ), row + 1, grid -> GetColLabelValue ( col ).c_str () );
```

```
            grid -> SetCellValue ( row, col, temp );
```

```
        }
```

```
    }
```

```
}
```

```
void Frame :: OnColAutoSize ( wxGridEvent& event )
```

```
{  
  
    int col = event.GetCol ();  
  
    wxGrid * grid = (wxGrid * ) FindWindow ( ID_GRID );  
  
    grid -> AutoSizeColumn ( col );  
  
    grid -> Refresh ();  
}
```

### 5.2.1.4.3 Kompilasi

#### 5.2.1.4.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:  
# simple makefile for simple wx project  
# (c) Nop, 5 Feb 2006  
  
# variables  
WXCONFIG = wx-config  
WXPLAT = gtk  
  
PROGRAM = grid  
OBJECTS = src/app.o src/frame.o  
  
# system  
CXX = $(shell $(WXCONFIG) --cxx)  
  
# implementation  
  
.SUFFIXES:    .o .cpp  
  
.cpp.o :  
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<  
  
all:  $(PROGRAM)
```

```
$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

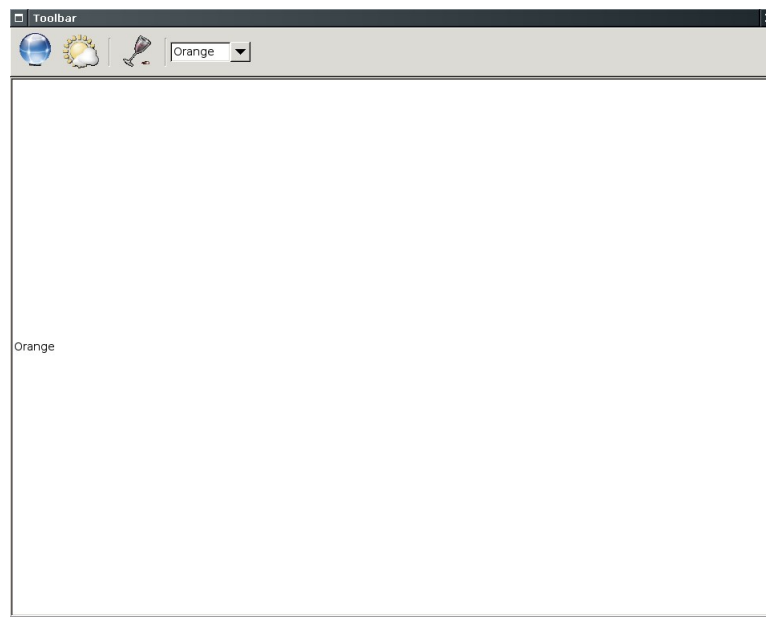
clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

## 5.2.2 wxToolBar

- wxToolBar
  - include files:
    - wx/toolbar.h
  - derived from:
    - wxToolBarBase
    - wxControl
    - wxWindow
    - wxEvtHandler
    - wxObject

### 5.2.2.1 Toolbar

Aplikasi demo untuk bekerja dengan wxToolBar. Menampilkan sebuah toolbar yang terdiri dari tiga tombol dan sebuah combobox. Mendemonstrasikan pula penanganan event.



#### 5.2.2.1.1 Struktur direktori

```
toolbar/0_toolbar/  
    build/  
        gtk/  
            toolbar  
            1.png  
            2.png  
            3.png  
    src/  
        app.h  
        app.cpp  
        frame.h  
        frame.cpp  
    makefile.gtk
```

#### 5.2.2.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

Terdapat pula penggunaan 1.png, 2.png dan 3.png untuk gambar pada tombol di

toolbar.

#### **5.2.2.1.2.1 src/app.h**

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

#### **5.2.2.1.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Toolbar") );

    frame -> Show ( true );

    return true;
```

```
}
```

#### 5.2.2.1.2.3 src/frame.h

```
#ifndef _FRAME_H_
#define _FRAME_H_

enum
{
    ID_TEXTCTRL = wxID_HIGHEST + 1,

    ID_TOOL1,

    ID_TOOL2,

    ID_COMBO1
};

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

    void OnTool1 ( wxCommandEvent& event );

    void OnTool2 ( wxCommandEvent& event );

    void OnQuit ( wxCommandEvent& event );

    void OnCombo ( wxCommandEvent& event );

    wxComboBox * m_combo;
```

```
public:
```

```
    DECLARE_EVENT_TABLE ();
```

```
};
```

```
#endif // _FRAME_H_
```

#### **5.2.2.1.2.4 src/frame.cpp**

```
#include "wx/wx.h"
```

```
#include "wx/toolbar.h"
```

```
#include "frame.h"
```

```
BEGIN_EVENT_TABLE ( Frame, wxFrame )
```

```
    EVT_TOOL ( ID_TOOL1, Frame :: OnTool1 )
```

```
    EVT_TOOL ( ID_TOOL2, Frame :: OnTool2 )
```

```
    EVT_TOOL ( wxID_EXIT, Frame :: OnQuit )
```

```
    EVT_COMBOBOX ( ID_COMBO1, Frame :: OnCombo )
```

```
END_EVENT_TABLE ()
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )  
{
```

```
    wxTextCtrl * textCtrl = new wxTextCtrl ( this, ID_TEXTCTRL );
```

```
wxToolBar * toolBar = new wxToolBar ( this, wxID_ANY );
```

```
wxBitmap bmp1;
```

```
bmp1.LoadFile ( _( "1.png" ) );
```

```
toolBar -> AddTool ( ID_TOOL1, _( "Tool 1" ), bmp1 );
```

```
wxBitmap bmp2;
```

```
bmp2.LoadFile ( _( "2.png" ) );
```

```
toolBar -> AddTool ( ID_TOOL2, _( "Tool 2" ), bmp2 );
```

```
toolBar -> AddSeparator ();
```

```
wxBitmap bmp3;
```

```
bmp3.LoadFile ( _( "3.png" ) );
```

```
toolBar -> AddTool ( wxID_EXIT, _( "Quit" ), bmp3 );
```

```
wxArrayString arrStr;
```

```
arrStr.Add ( _( "Apple" ) );
```

```
arrStr.Add ( _( "Orange" ) );
```

```
arrStr.Add ( _( "Pineapple" ) );
```

```
m_combo = new wxComboBox ( toolBar, ID_COMBO1, wxEmptyString,  
wxDefaultPosition, wxSize ( 100, 50 ), arrStr );
```

```
toolBar -> AddSeparator ();
```

```
toolBar -> AddControl ( m_combo );
```

```
toolBar -> Realize ();
```

```
SetToolBar ( toolBar );
```



```
}

void Frame :: OnTool1 ( wxCommandEvent& event )
{

    wxString msg;

    wxTextCtrl * text = (wxTextCtrl *) FindWindow ( ID_TEXTCTRL );

    msg.Printf ( _( "Tool1" ) );

    text -> SetValue ( msg );

}

void Frame :: OnTool2 ( wxCommandEvent& event )
{

    wxString msg;

    wxTextCtrl * text = (wxTextCtrl *) FindWindow ( ID_TEXTCTRL );

    msg.Printf ( _( "Tool2" ) );

    text -> SetValue ( msg );

}

void Frame :: OnCombo ( wxCommandEvent& event )
{

    wxTextCtrl * text = (wxTextCtrl *) FindWindow ( ID_TEXTCTRL );

    text -> SetValue ( m_combo -> GetValue () );
```

```
}  
  
void Frame :: OnQuit ( wxCommandEvent& event )  
{  
  
    Close ();  
}
```

### 5.2.2.1.3 Kompilasi

#### 5.2.2.1.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:  
# simple makefile for simple wx project  
# (c) Nop, 5 Feb 2006  
  
# variables  
WXCONFIG = wx-config  
WXPLAT = gtk  
  
PROGRAM = toolbar  
OBJECTS = src/app.o src/frame.o  
  
# system  
CXX = $(shell $(WXCONFIG) --cxx)  
  
# implementation  
  
.SUFFIXES:    .o .cpp  
  
.cpp.o :  
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<  
  
all:  $(PROGRAM)  
  
$(PROGRAM):  $(OBJECTS)  
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`  
    mkdir -p build/$(WXPLAT)
```

```
mv $(PROGRAM) build/$(WXPLAT)
rm -f src/*.o
```

clean:

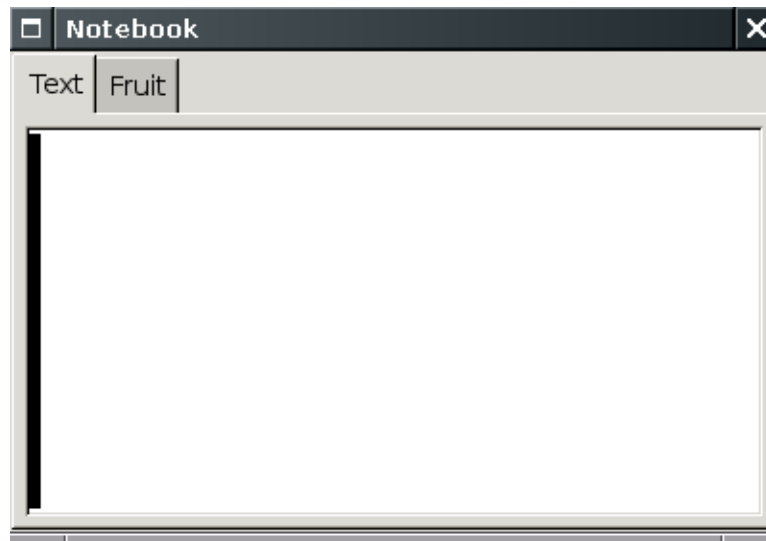
```
rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

### 5.2.3 wxNotebook

- wxNotebook
  - include files:
    - wx/notebook.h
  - derived from:
    - wxControl
    - wxWindow
    - wxEvtHandler
    - wxObject

#### 5.2.3.1 Notebook

Aplikasi demo untuk bekerja dengan wxNotebook. Terdiri dari dua book. Book pertama berisikan text control, book kedua berisikan Radio Box.



#### 5.2.3.1.1 Struktur direktori

```
notebook/0_notebook/  
    build/  
        gtk/  
            notebook  
    src/  
        app.h  
        app.cpp  
        frame.h  
        frame.cpp  
    makefile.gtk
```

#### 5.2.3.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

**5.2.3.1.2.1 src/app.h**

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

**5.2.3.1.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Notebook") );

    frame -> Show ( true );

    return true;
}
```

**5.2.3.1.2.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

**5.2.3.1.2.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "frame.h"

Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{
    wxNotebook * noteBook = new wxNotebook ( this, wxID_ANY );

    wxPanel * panel1 = new wxPanel ( noteBook, wxID_ANY );

    wxTextCtrl * textCtrl = new wxTextCtrl ( panel1, wxID_ANY );

    wxBoxSizer * sizer1 = new wxBoxSizer ( wxVERTICAL );

    sizer1 -> Add ( textCtrl, 1, wxEXPAND | wxALL, 5 );
```

```
panel1 -> SetSizer ( sizer1 );

noteBook -> AddPage ( panel1, _( "Text" ) );

wxPanel * panel2 = new wxPanel ( noteBook, wxID_ANY );

wxArrayString fruit;

fruit.Add ( _( "Apple" ) );
fruit.Add ( _( "Banana" ) );
fruit.Add ( _( "Cherry" ) );
fruit.Add ( _( "Tomato" ) );
fruit.Add ( _( "Pineapple" ) );
fruit.Add ( _( "Grape" ) );

wxRadioButton * radioButton = new wxRadioButton ( panel2, wxID_ANY, _( "Choose your
favourite fruit" ),
                                wxDefaultPosition, wxDefaultSize, fruit );

wxBoxSizer * sizer2 = new wxBoxSizer ( wxVERTICAL );

sizer2 -> Add ( radioButton, 1, wxEXPAND | wxALL, 5 );

panel2 -> SetSizer ( sizer2 );

noteBook -> AddPage ( panel2, _( "Fruit" ) );

}
```

### 5.2.3.1.3 Kompilasi

#### 5.2.3.1.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk

PROGRAM = notebook
OBJECTS = src/app.o src/frame.o

# system
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES:    .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all:  $(PROGRAM)

$(PROGRAM):  $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

## 5.2.4 wxListbook

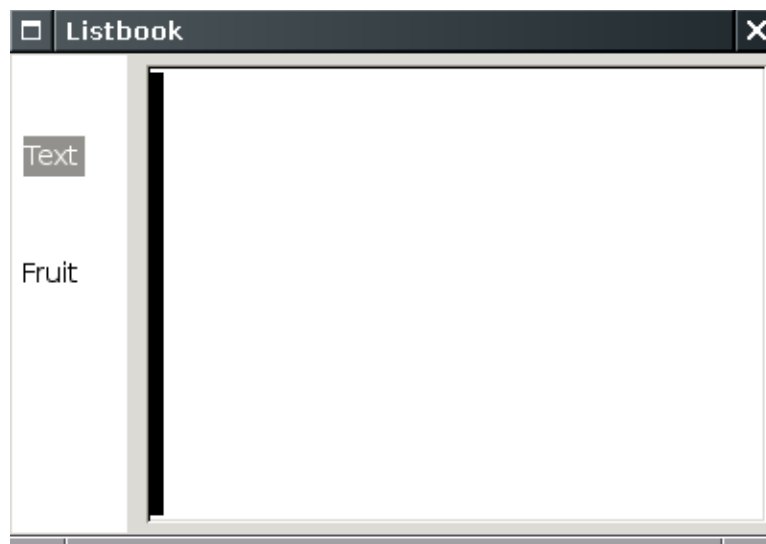
– wxListbook



- include files:
  - wx/listbook.h
- derived from:
  - wxControl
  - wxWindow
  - wxEvtHandler
  - wxObject

#### 5.2.4.1 Listbook

Aplikasi demo untuk bekerja dengan wxListbook. Terdiri dari dua book. Book pertama berisikan text control, book kedua berisikan Radio Box.



##### 5.2.4.1.1 Struktur direktori

```
listbook/0_listbook/  
  build/  
    gtk/  
      listbook  
  src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp
```

makefile.gtk

#### 5.2.4.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

##### 5.2.4.1.2.1 src/app.h

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

##### 5.2.4.1.2.2 src/app.cpp

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )
```

```
bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Listbook") );

    frame -> Show ( true );

    return true;
}
```

#### **5.2.4.1.2.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### **5.2.4.1.2.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "wx/listbook.h"
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{
    wxListbook * listBook = new wxListbook ( this, wxID_ANY );

    wxPanel * panel1 = new wxPanel ( listBook, wxID_ANY );

    wxTextCtrl * textCtrl = new wxTextCtrl ( panel1, wxID_ANY );

    wxBoxSizer * sizer1 = new wxBoxSizer ( wxVERTICAL );

    sizer1 -> Add ( textCtrl, 1, wxEXPAND | wxALL, 5 );

    panel1 -> SetSizer ( sizer1 );

    listBook -> AddPage ( panel1, _( "Text" ) );

    wxPanel * panel2 = new wxPanel ( listBook, wxID_ANY );

    wxArrayString fruit;

    fruit.Add ( _( "Apple" ) );
    fruit.Add ( _( "Banana" ) );
    fruit.Add ( _( "Cherry" ) );
    fruit.Add ( _( "Tomato" ) );
    fruit.Add ( _( "Pineapple" ) );
    fruit.Add ( _( "Grape" ) );

    wxRadioBox * radioButton = new wxRadioBox ( panel2, wxID_ANY, _( "Choose your
favourite fruit" ),
                                                wxDefaultPosition, wxDefaultSize, fruit );

    wxBoxSizer * sizer2 = new wxBoxSizer ( wxVERTICAL );

    sizer2 -> Add ( radioButton, 1, wxEXPAND | wxALL, 5 );

    panel2 -> SetSizer ( sizer2 );

    listBook -> AddPage ( panel2, _( "Fruit" ) );
}
```

### 5.2.4.1.3 Kompilasi

#### 5.2.4.1.3.1 Kompilasi di Linux (GTK+)

makefile.gtk:

# simple makefile for simple wx project

# (c) Nop, 5 Feb 2006

# variables

WXCONFIG = wx-config

WXPLAT = gtk

PROGRAM = listbook

OBJECTS = src/app.o src/frame.o

# system

CXX = \$(shell \$(WXCONFIG) --cxx)

# implementation

.SUFFIXES: .o .cpp

.cpp.o :

\$(CXX) -c \$(WXCONFIG) --cxxflags` -o \$@ \$<

all: \$(PROGRAM)

\$(PROGRAM): \$(OBJECTS)

\$(CXX) -o \$(PROGRAM) \$(OBJECTS) \$(WXCONFIG) --libs`

mkdir -p build/\$(WXPLAT)

mv \$(PROGRAM) build/\$(WXPLAT)

rm -f src/\*.o

clean:

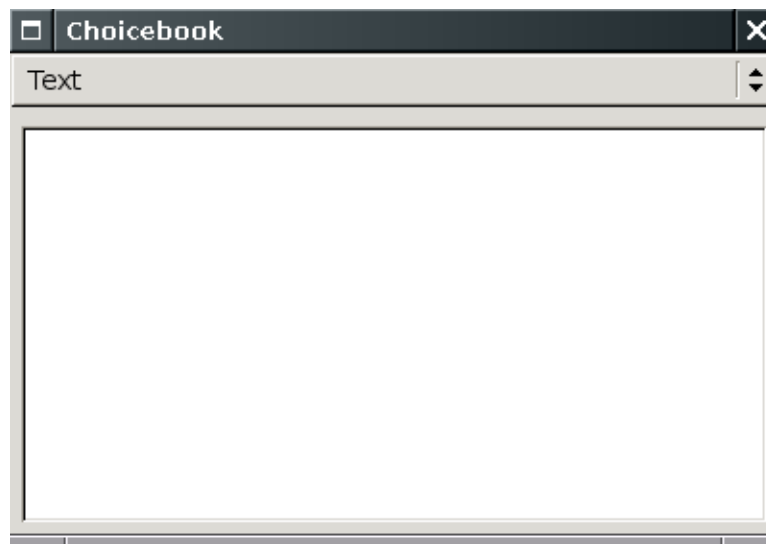
rm -f src/\*.o build/\$(WXPLAT)/\$(PROGRAM)

## 5.2.5 wxChoicebook

- wxChoicebook
  - include files:
    - wx/choicebk.h
  - derived from:
    - wxControl
    - wxWindow
    - wxEvtHandler
    - wxObject

### 5.2.5.1 Choicebook

Aplikasi demo untuk bekerja dengan wxChoicebook. Terdiri dari dua book. Book pertama berisikan text control, book kedua berisikan Radio Box.



### 5.2.5.1.1 Struktur direktori

```
choicebook/0_choicebook/  
    build/  
        gtk/  
            choicebook  
src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
makefile.gtk
```

### 5.2.5.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

#### 5.2.5.1.2.1 src/app.h

```
#ifndef _APP_H_  
#define _APP_H_  
  
class App : public wxApp  
{  
public:  
  
    virtual bool OnInit ();  
};  
  
#endif // _APP_H_
```

**5.2.5.1.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Choicebook") );

    frame -> Show ( true );

    return true;
}
```

**5.2.5.1.2.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};
```



```
#endif // _FRAME_H_
```

#### 5.2.5.1.2.4 src/frame.cpp

```
#include "wx/wx.h"  
#include "wx/choicebk.h"  
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )  
{  
    wxChoicebook * choiceBook = new wxChoicebook ( this, wxID_ANY );
```

```
    wxPanel * panel1 = new wxPanel ( choiceBook, wxID_ANY );
```

```
    wxTextCtrl * textCtrl = new wxTextCtrl ( panel1, wxID_ANY );
```

```
    wxBoxSizer * sizer1 = new wxBoxSizer ( wxVERTICAL );
```

```
    sizer1 -> Add ( textCtrl, 1, wxEXPAND | wxALL, 5 );
```

```
    panel1 -> SetSizer ( sizer1 );
```

```
    choiceBook -> AddPage ( panel1, _( "Text" ) );
```

```
    wxPanel * panel2 = new wxPanel ( choiceBook, wxID_ANY );
```

```
    wxArrayString fruit;
```

```
    fruit.Add ( _( "Apple" ) );
```

```
    fruit.Add ( _( "Banana" ) );
```

```
    fruit.Add ( _( "Cherry" ) );
```

```
    fruit.Add ( _( "Tomato" ) );
```

```
    fruit.Add ( _( "Pineapple" ) );
```

```
    fruit.Add ( _( "Grape" ) );
```

```
    wxRadioBox * radioBox = new wxRadioBox ( panel2, wxID_ANY, _( "Choose your
favourite fruit" ),
                                wxDefaultPosition, wxDefaultSize, fruit );

    wxBoxSizer * sizer2 = new wxBoxSizer ( wxVERTICAL );

    sizer2 -> Add ( radioBox, 1, wxEXPAND | wxALL, 5 );

    panel2 -> SetSizer ( sizer2 );

    choiceBook -> AddPage ( panel2, _( "Fruit" ) );

}
```

### 5.2.5.1.3 Kompilasi

#### 5.2.5.1.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006

# variables
WXCONFIG = wx-config
WXPLAT = gtk

PROGRAM = choicebook
OBJECTS = src/app.o src/frame.o

# system
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES:    .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all:  $(PROGRAM)
```

```
$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

### 5.3 Common dialog

Berbagai dialog yang umum ditemukan di dalam berbagai aplikasi

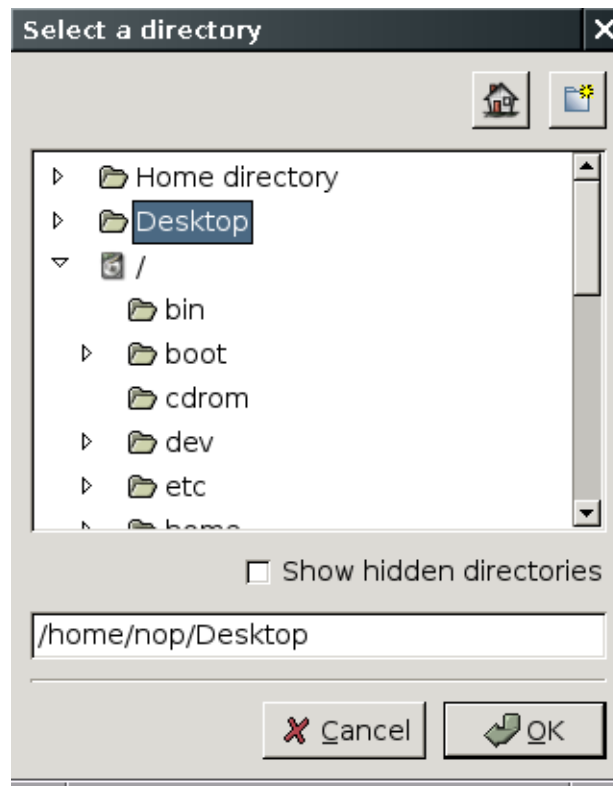
<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxDialog	Base class untuk berbagai dialog umum
wxColourDialog	Color chooser
wxDirDialog	Directory selector
wxFileDialog	File selector
wxFindReplaceDialog	Text search/replace
wxMultiChoiceDialog	Multiple choice
wxSingleChoiceDialog	Single choice
wxTextEntryDialog	Text entry
wxPasswordEntryDialog	Password entry
wxFontDialog	Font selector
wxPageSetupDialog	Page setup standard
wxPrintDialog	Print dialog
wxProgressDialog	Indikasi progress
wxMessageDialog	Message box sederhana
wxWizard	wizard

### 5.3.1 wxDirDialog

- wxDirDialog
  - include files:
    - wx/dirdlg.h
  - derived from:
    - wxDialog
    - wxWindow
    - wxEvtHandler
    - wxObject

#### 5.3.1.1 Dir Dialog 0

Aplikasi demo untuk bekerja dengan wxDirDialog. User akan diminta untuk memilih direktori tertentu. Setelah user memilih, message box yang berisi nama direktori yang dipilih akan ditampilkan.



##### 5.3.1.1.1 Struktur direktori

```
dir_dialog/0_dir_dialog/  
    build/  
        gtk/  
            dir_dialog  
src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
makefile.gtk
```

### 5.3.1.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

#### 5.3.1.1.2.1 src/app.h

```
#ifndef _APP_H_  
#define _APP_H_  
  
class App : public wxApp  
{  
public:  
  
    virtual bool OnInit ();  
};  
  
#endif // _APP_H_
```

#### 5.3.1.1.2.2 src/app.cpp

```
#include "wx/wx.h"  
#include "frame.h"
```

```
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Dir Dialog 0") );

    frame -> Show ( true );

    return true;
}
```

#### **5.3.1.1.2.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### **5.3.1.1.2.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "wx/dirdlg.h"
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{
```

```
    wxDirDialog * dirDialog = new wxDirDialog (this, _( "Select a directory" ) );
```

```
    if ( dirDialog -> ShowModal () == wxID_OK )
    {
```

```
        wxString dirPath = dirDialog -> GetPath ();
```

```
        wxMessageBox ( dirPath , _( "Dir Dialog" ), wxOK, this );
```

```
    }
```

```
}
```

### 5.3.1.1.3 Kompilasi

#### 5.3.1.1.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk
```

```
PROGRAM = dir_dialog
OBJECTS = src/app.o src/frame.o
```

```
# system
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES:    .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all:  $(PROGRAM)

$(PROGRAM):  $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

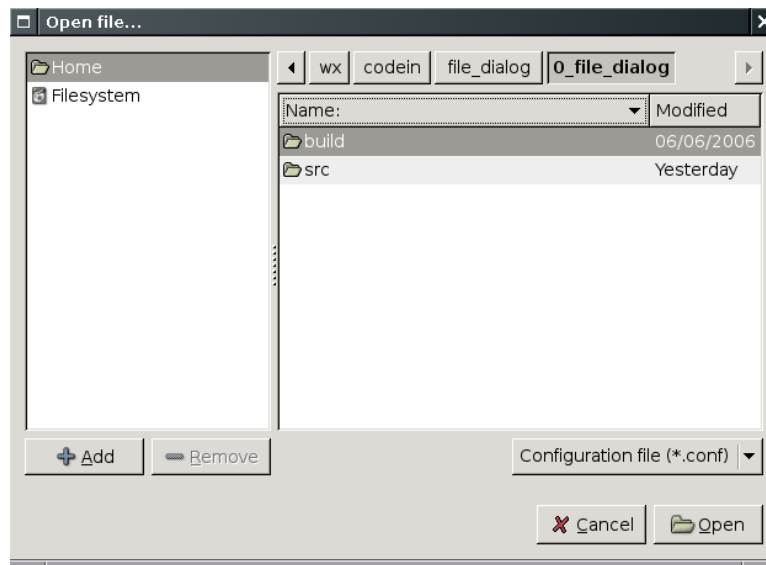
### 5.3.2 wxFileDialog

- wxFileDialog
  - include files:
    - wx/filedlg.h
  - derived from:
    - wxDialog
    - wxWindow
    - wxEvtHandler
    - wxObject



### 5.3.2.1 File Dialog 0

Aplikasi demo untuk bekerja dengan `wxFileDialog`. User akan diminta untuk membuka file tertentu. Setelah user memilih, message box yang berisi nama file yang dipilih akan ditampilkan.



#### 5.3.2.1.1 Struktur direktori

```
file_dialog/0_file_dialog/  
    build/  
        gtk/  
            file_dialog  
src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
makefile.gtk
```

#### 5.3.2.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

#### **5.3.2.1.2.1 src/app.h**

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

#### **5.3.2.1.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("File Dialog 0") );

    frame -> Show ( true );
```

```
    return true;
}
```

#### 5.3.2.1.2.3 src/frame.h

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### 5.3.2.1.2.4 src/frame.cpp

```
#include "wx/wx.h"
#include "wx/filedlg.h"
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{
```

```
    wxFileDialog * openFileDialog = new wxFileDialog (this, _( "Open file..." ), wxEmptyString
```

```
, wxEmptyString,  
    _( "Configuration file (*.conf) | *.conf | Text file (*.txt) | *.txt | All files (*.*) | *.*" ),  
wxOPEN );  
  
if ( openFileDialog -> ShowModal () == wxID_OK )  
{  
  
    wxString filePath = openFileDialog -> GetPath ();  
  
    wxMessageBox ( filePath , _( "File Dialog" ), wxOK, this );  
  
}  
  
}
```

### 5.3.2.1.3 Kompilasi

#### 5.3.2.1.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:  
# simple makefile for simple wx project  
# (c) Nop, 5 Feb 2006
```

```
# variables  
WXCONFIG = wx-config  
WXPLAT = gtk
```

```
PROGRAM = file_dialog  
OBJECTS = src/app.o src/frame.o
```

```
# system  
CXX = $(shell $(WXCONFIG) --cxx)
```

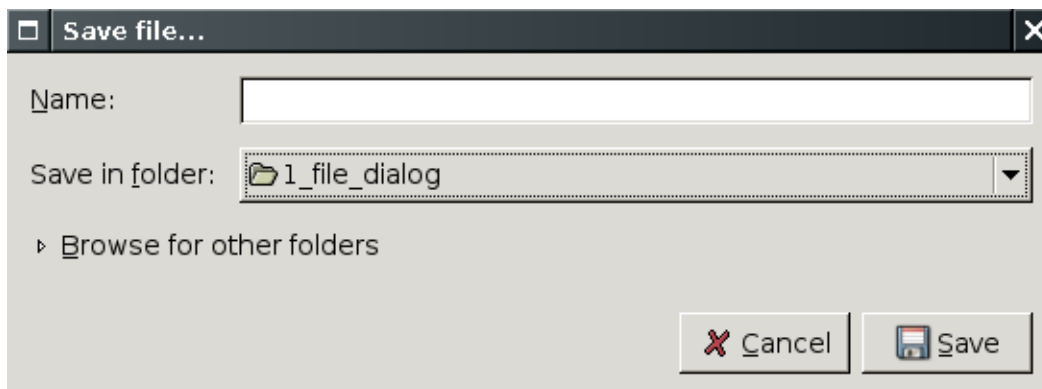
```
# implementation
```

```
.SUFFIXES:    .o .cpp
```

```
.cpp.o :  
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<  
  
all:  $(PROGRAM)  
  
$(PROGRAM):  $(OBJECTS)  
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`  
    mkdir -p build/$(WXPLAT)  
    mv $(PROGRAM) build/$(WXPLAT)  
    rm -f src/*.o  
  
clean:  
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

### 5.3.2.2 File Dialog 1

Aplikasi demo untuk bekerja dengan wxFileDialog. User akan diminta untuk menyimpan file. Setelah user memasukkan nama file, message box yang berisi nama file yang dimasukkan oleh user akan ditampilkan.



#### 5.3.2.2.1 Struktur direktori

```
file_dialog/1_file_dialog/  
    build/  
        gtk/  
            file_dialog
```

```
src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
makefile.gtk
```

### 5.3.2.2.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

#### 5.3.2.2.2.1 src/app.h

```
#ifndef _APP_H_  
#define _APP_H_  
  
class App : public wxApp  
{  
public:  
  
    virtual bool OnInit ();  
};  
  
#endif // _APP_H_
```

#### 5.3.2.2.2.2 src/app.cpp

```
#include "wx/wx.h"  
#include "frame.h"  
#include "app.h"
```

```
DECLARE_APP ( App )
```

```
IMPLEMENT_APP ( App )
```

```
bool App :: OnInit ()  
{  
    Frame * frame = new Frame ( _ ("File Dialog 1") );  
  
    frame -> Show ( true );  
  
    return true;  
}
```

#### **5.3.2.2.2.3 src/frame.h**

```
#ifndef _FRAME_H_  
#define _FRAME_H_  
  
class Frame: public wxFrame  
{  
public:  
  
    Frame ( const wxString& title );  
  
};  
  
#endif // _FRAME_H_
```

#### **5.3.2.2.2.4 src/frame.cpp**

```
#include "wx/wx.h"
```

```
#include "wx/filedlg.h"
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{
```

```
    wxFileDialog * saveDialog = new wxFileDialog (this, _( "Save file..." ), wxEmptyString ,
wxEmptyString, wxEmptyString, wxSAVE );
```

```
    if ( saveDialog -> ShowModal () == wxID_OK )
    {
```

```
        wxString filePath = saveDialog -> GetPath ();
```

```
        wxMessageBox ( filePath , _( "File Dialog" ), wxOK, this );
```

```
    }
```

```
}
```

### 5.3.2.2.3 Kompilasi

#### 5.3.2.2.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk
```

```
PROGRAM = file_dialog
OBJECTS = src/app.o src/frame.o
```



```
# system
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES:    .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all:  $(PROGRAM)

$(PROGRAM):  $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

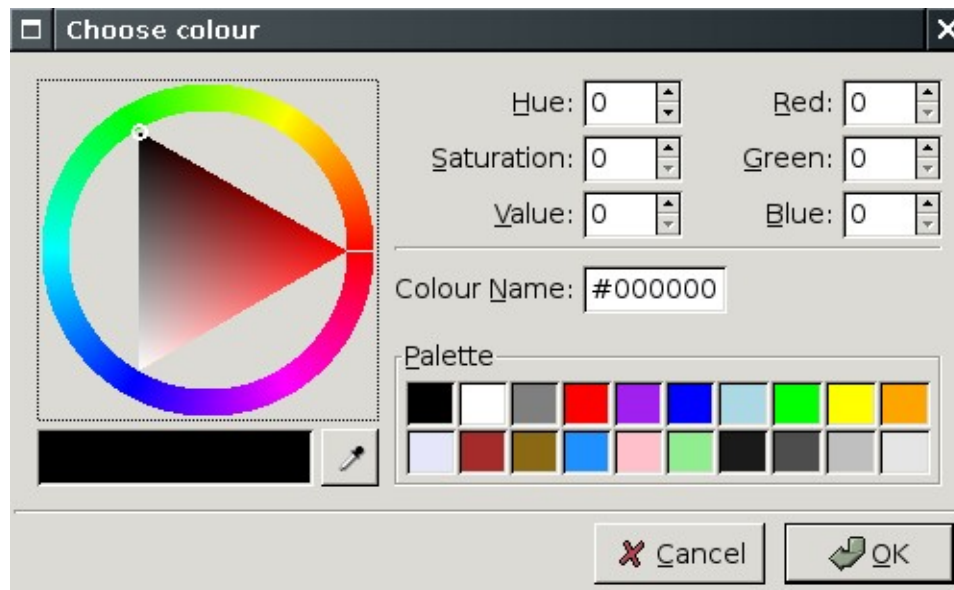
### 5.3.3 wxColourDialog

- wxColourDialog
  - include files:
    - wx/colordlg.h
  - derived from:
    - wxDialog
    - wxWindow
    - wxEvtHandler
    - wxObject

#### 5.3.3.1 Color Dialog

Aplikasi demo untuk bekerja dengan wxColourDialog. User akan diminta untuk memilih warna tertentu. Setelah user memilih, message box berisikan informasi warna

pilihan user akan ditampilkan dan warna background frame utama akan diubah sesuai pilihan user tersebut.



#### 5.3.3.1.1 Struktur direktori

```
color_dialog/0_color_dialog/  
    build/  
        gtk/  
            color_dialog  
src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
makefile.gtk
```

#### 5.3.3.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

**5.3.3.1.2.1 src/app.h**

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

**5.3.3.1.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Color Dialog") );

    frame -> Show ( true );

    return true;
}
```

#### 5.3.3.1.2.3 src/frame.h

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### 5.3.3.1.2.4 src/frame.cpp

```
#include "wx/wx.h"
#include "wx/colordlg.h"
#include "wx/cmndata.h"
#include "wx/colour.h"
#include "frame.h"

Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{

    wxColourDialog * colorDialog = new wxColourDialog ( this );

    if ( colorDialog -> ShowModal () == wxID_OK )
    {
```

```
wxColourData colorData = colorDialog -> GetColourData ();

wxColour color = colorData. GetColour ();

wxString temp;

temp = temp.Format ( _( "Color data\nRed: %d\nGreen: %d\nBlue: %d\n" ),
color.Red (), color.Green (), color.Blue () );

wxMessageBox ( temp , _( "Color Dialog" ), wxOK, this );

SetBackgroundColour ( color );

}

}
```

### 5.3.3.1.3 Kompilasi

#### 5.3.3.1.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk
```

```
PROGRAM = color_dialog
OBJECTS = src/app.o src/frame.o
```

```
# system
CXX = $(shell $(WXCONFIG) --cxx)
```

# implementation

.SUFFIXES: .o .cpp

.cpp.o :  
\$(CXX) -c `\$(WXCONFIG) --cxxflags` -o \$@ \$<

all: \$(PROGRAM)

\$(PROGRAM): \$(OBJECTS)  
\$(CXX) -o \$(PROGRAM) \$(OBJECTS) `\$(WXCONFIG) --libs`  
mkdir -p build/\$(WXPLAT)  
mv \$(PROGRAM) build/\$(WXPLAT)  
rm -f src/\*.o

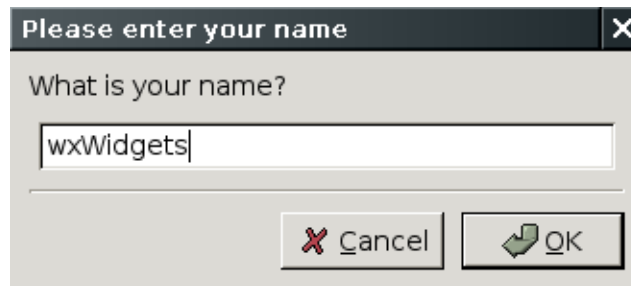
clean:  
rm -f src/\*.o build/\$(WXPLAT)/\$(PROGRAM)

## 5.3.4 wxTextEntryDialog

- wxTextEntryDialog
  - include files:
    - wx/textdlg.h
  - derived from:
    - wxDialog
    - wxWindow
    - wxEvtHandler
    - wxObject

### 5.3.4.1 Text Entry Dialog

Aplikasi demo untuk bekerja dengan wxTextEntryDialog. User akan diminta untuk memasukkan teks. Setelah itu, message box yang berisikan teks yang dimasukkan tersebut akan ditampilkan.



#### 5.3.4.1.1 Struktur direktori

```
text_entry_dialog/0_text_entry_dialog/  
    build/  
        gtk/  
            text_entry_dialog  
    src/  
        app.h  
        app.cpp  
        frame.h  
        frame.cpp  
    makefile.gtk
```

#### 5.3.4.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

##### 5.3.4.1.2.1 src/app.h

```
#ifndef _APP_H_  
#define _APP_H_  
  
class App : public wxApp  
{
```

```
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

#### **5.3.4.1.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Text Entry Dialog" ) );

    frame -> Show ( true );

    return true;
}
```

#### **5.3.4.1.2.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:
```



```
Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### 5.3.4.1.2.4 src/frame.cpp

```
#include "wx/wx.h"
#include "wx/textdlg.h"
#include "frame.h"

Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{

    wxTextEntryDialog * input = new wxTextEntryDialog ( this, _( "What is your name?" ),
    _( "Please enter your name" ) );

    if ( input -> ShowModal () == wxID_OK )
    {

        wxString temp;

        wxString value = input -> GetValue ();

        temp = temp.Format ( _( "Hello, %s\nHow are you today?" ), value.c_str () );

        wxMessageBox ( temp , _( "Text Entry Dialog" ), wxOK, this );

    }

}
```

### 5.3.4.1.3 Kompilasi

#### 5.3.4.1.3.1 Kompilasi di Linux (GTK+)

makefile.gtk:

# simple makefile for simple wx project

# (c) Nop, 5 Feb 2006

# variables

WXCONFIG = wx-config

WXPLAT = gtk

PROGRAM = text\_entry\_dialog

OBJECTS = src/app.o src/frame.o

# system

CXX = \$(shell \$(WXCONFIG) --cxx)

# implementation

.SUFFIXES: .o .cpp

.cpp.o :

\$(CXX) -c `\$(WXCONFIG) --cxxflags` -o \$@ \$<

all: \$(PROGRAM)

\$(PROGRAM): \$(OBJECTS)

\$(CXX) -o \$(PROGRAM) \$(OBJECTS) `\$(WXCONFIG) --libs`

mkdir -p build/\$(WXPLAT)

mv \$(PROGRAM) build/\$(WXPLAT)

rm -f src/\*.o

clean:

rm -f src/\*.o build/\$(WXPLAT)/\$(PROGRAM)

### 5.3.5 wxPasswordEntryDialog

- wxPasswordEntryDialog
  - include files:
    - wx/textdlg.h
  - derived from:
    - wxTextEntryDialog
    - wxDialog
    - wxWindow
    - wxEvtHandler
    - wxObject

#### 5.3.5.1 Password Entry Dialog

Aplikasi demo untuk bekerja dengan wxPasswordEntryDialog. User akan diminta untuk memasukkan teks password. Setelah itu, message box yang berisikan teks password yang dimasukkan tersebut akan ditampilkan.

##### 5.3.5.1.1 Struktur direktori

```
passwd_entry_dialog/0_passwd_entry_dialog/  
    build/  
        gtk/  
            passwd_entry_dialog  
src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
makefile.gtk
```

##### 5.3.5.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App

- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

#### **5.3.5.1.2.1 src/app.h**

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

#### **5.3.5.1.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Password Entry Dialog") );

    frame -> Show ( true );

    return true;
}
```

**5.3.5.1.2.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

**5.3.5.1.2.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "wx/textdlg.h"
#include "frame.h"

Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{

    wxPasswordEntryDialog * input = new wxPasswordEntryDialog ( this, _( "Enter your
password" ), _( "Password needed" ) );

    if ( input -> ShowModal () == wxID_OK )
    {

        wxString temp;

        wxString value = input -> GetValue ();
```

```
temp = temp.Format ( _( "You have entered:\n%s" ), value.c_str () );  
  
wxMessageBox ( temp , _( "Password Entry Dialog" ), wxOK, this );  
  
}  
  
}
```

### 5.3.5.1.3 Kompilasi

#### 5.3.5.1.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:  
# simple makefile for simple wx project  
# (c) Nop, 5 Feb 2006  
  
# variables  
WXCONFIG = wx-config  
WXPLAT = gtk  
  
PROGRAM = passwd_entry_dialog  
OBJECTS = src/app.o src/frame.o  
  
# system  
CXX = $(shell $(WXCONFIG) --cxx)  
  
# implementation  
  
.SUFFIXES:    .o .cpp  
  
.cpp.o :  
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<  
  
all:  $(PROGRAM)  
  
$(PROGRAM):  $(OBJECTS)
```

```
$(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`  
mkdir -p build/$(WXPLAT)  
mv $(PROGRAM) build/$(WXPLAT)  
rm -f src/*.o
```

clean:

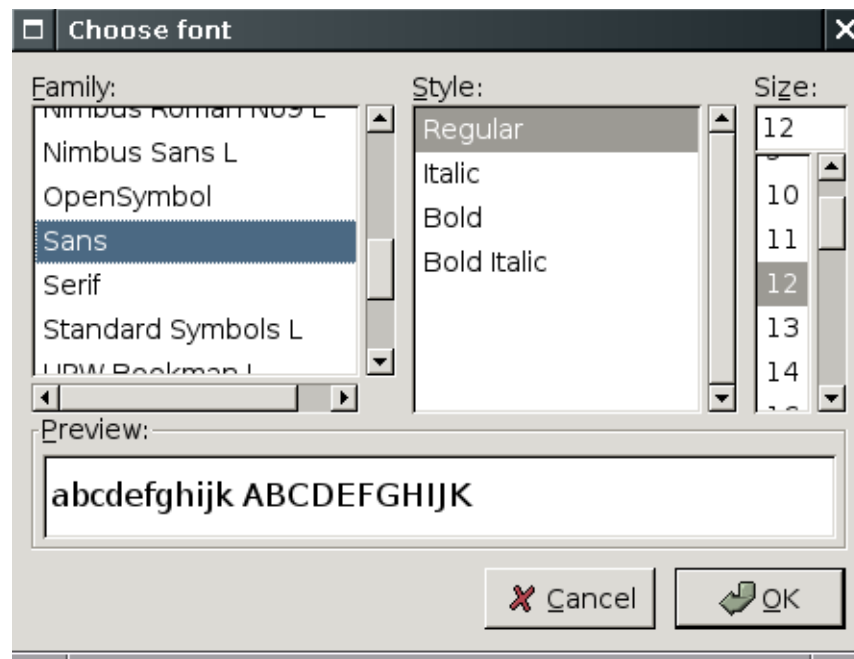
```
rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

### 5.3.6 wxFontDialog

- wxFontEntryDialog
  - include files:
    - wx/fontdlg.h
  - derived from:
    - wxDialog
    - wxWindow
    - wxEvtHandler
    - wxObject

#### 5.3.6.1 Font Dialog

Aplikasi demo untuk bekerja dengan wxFontDialog. User akan diminta untuk memilih font. Setelah itu, font pada text control frame utama akan diset sesuai font yang dipilih oleh user.



#### 5.3.6.1.1 Struktur direktori

```
font_dialog/0_font_dialog/  
    build/  
        gtk/  
            font_dialog  
src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
    makefile.gtk
```

#### 5.3.6.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame



**5.3.6.1.2.1 src/app.h**

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

**5.3.6.1.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _ ("Font Dialog") );

    frame -> Show ( true );

    return true;
}
```

#### 5.3.6.1.2.3 src/frame.h

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### 5.3.6.1.2.4 src/frame.cpp

```
#include "wx/wx.h"
#include "wx/fontdlg.h"
#include "wx/textctrl.h"
#include "frame.h"

Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{

    wxTextCtrl * textCtrl = new wxTextCtrl ( this, wxID_ANY, _( "This is multiline Text
Control.\nThis is line 2." ),
        wxDefaultPosition, wxDefaultSize, wxTE_MULTILINE );

    wxFontDialog * fontDialog = new wxFontDialog ( this );
```

```
if ( fontDialog -> ShowModal () == wxID_OK )
{

    wxFontData fontData = fontDialog -> GetFontData ();

    wxFont font = fontData.GetChosenFont ();

    wxTextAttr textAttr ( fontData.GetColour (), wxNullColour, font );

    wxString val = textCtrl -> GetValue ();

    textCtrl -> SetStyle ( 0, val.Len (), textAttr );

}

}
```

### 5.3.6.1.3 Kompilasi

#### 5.3.6.1.3.1 Kompilasi di Linux (GTK+)

```
makefile.gtk:
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk
```

```
PROGRAM = font_dialog
OBJECTS = src/app.o src/frame.o
```

```
# system
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES:    .o .cpp
```

```
.cpp.o :  
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<
```

```
all:    $(PROGRAM)
```

```
$(PROGRAM):    $(OBJECTS)  
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`  
    mkdir -p build/$(WXPLAT)  
    mv $(PROGRAM) build/$(WXPLAT)  
    rm -f src/*.o
```

```
clean:  
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

## 5.4 Berbagai control

Window kecil yang menyediakan interaksi dengan user.

<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxControl	Base class untuk control
wxBUTTON	Button/tombol, menampilkan teks
wxBitmapButton	Button/tombol, menampilkan gambar
wxToggleButton	Button/tombol toggle
wxCalendarCtrl	Menampilkan kalender
wxCheckBox	Checkbox
wxCheckListBox	List checkbox
wxChoice	Choice (combo box tanpa editable area)
wxComboBox	Combobox
wxDatePickerCtrl	Date picker
wxGauge	Gauge
wxGenericDirCtrl	Menampilkan tree directory

<b><i>Class</i></b>	<b><i>Deskripsi</i></b>
wxHtmlListBox	Listbox (menampilkan HTML)
wxStaticBox	Groupbox visual
wxListBox	List string
wxListCtrl	List untuk menampilkan string/icon, dengan kemampuan multicolumn
wxListView	List sederhana
wxTextCtrl	Text editing (single line atau multi line)
wxTreeCtrl	Tree control
wxScrollBar	Scrollbar
wxSpinButton	Tombol spin up/down
wxSpinCtrl	Spin control
wxStaticText	Non editable text, single line atau multi line
wxStaticBitmap	Menampilkan gambar
wxRadioBox	Group radio button
wxRadioButton	Radio button
wxSlider	Slider yang dapat didrag oleh user
wxVListBox	Listbox yang mendukung row dengan height variabel

## 5.5 Menu

Control menu pada wxWidgets

<b><i>Class</i></b>	<b><i>Deskripsi</i></b>
wxMenu	Menu (deretan menu item)
wxMenuBar	Menubar (deretan menu)
wxMenuItem	Merepresentasikan satu menu item

## 5.6 Window layout

Sistem layout yang didukung wxWidgets.

### 5.6.1 Window layout berbasis sizer

<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxSizer	Base class abstrak
wxGridSizer	Grid Sizer (lihat pembahasan tentang sizer)
wxFlexGridSizer	Flex Grid Sizer (lihat pembahasan tentang sizer)
wxGridBagSizer	Grid Bag Sizer (lihat pembahasan tentang sizer)
wxBoxSizer	Box Sizer (lihat pembahasan tentang sizer)
wxStaticBoxSizer	Static Box Sizer (lihat pembahasan tentang sizer)

### 5.6.2 Window layout berbasis constrain

Sistem layout yang satu ini telah deprecated, namun masih tersedia.

<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxIndividualLayoutConstrain†	Single constraint dimension
wxLayoutConstraints	Constrain untuk class window

## 5.7 Device Context

Permukaan (surface) yang dapat digambar.

<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxBufferedDC	Helper DC untuk double buffering
wxBufferedPaintDC	Helper DC untuk double buffering di dalam event OnPaint
wxClientDC	DC untuk mengakses client area diluar event OnPaint
wxPaintDC	DC untuk mengakses client area didalam event OnPaint

<b><i>Class</i></b>	<b><i>Deskripsi</i></b>
wxWindowDC	DC untuk mengakses non client area
wxScreenDC	DC untuk mengakses keseluruhan layar
wxDC	Base class DC
wxMemoryDC	DC untuk menggambar ke dalam bitmap
wxMetafileDC	DC untuk menggambar ke dalam metafile
wxMirrorDC	Proxy DC untuk mirroring
wxPostScriptDC	DC untuk menggambar ke dalam file postscript
wxPrinterDC	DC untuk menggambar ke printer

## **5.8 Graphic Device Interface**

Class yang berhubungan dengan penggambaran pada DC dan window

<b><i>Class</i></b>	<b><i>Deskripsi</i></b>
wxColour	Merepresentasikan elemen warna Red, Green dan Blue
wxDCClipper	Pengaturan daerah clipping
wxBitmap	Merepresentasikan sebuah bitmap
wxBrush	Mengisi area pada DC
wxBrushList	Daftar dari brush yang sebelumnya dibuat
wxCursor	Merepresentasikan kursor
wxFont	Merepresentasikan font
wxFontList	Merepresentasikan font yang sebelumnya dibuat
wxIcon	Icon
wxImage	Platform independent image
wxImageList	Image list
wxMask	Merepresentasikan Mask untuk bekerja pada bitmap, untuk kebutuhan penggambaran transparan
wxPen	Menggambar line pada DC
wxPenList	Daftar pen yang sebelumnya dibuat

<b>Class</b>	<b>Deskripsi</b>
wxPalette	Palet
wxRegion	Region
wxRendererNative	High level drawing

## 5.9 Event

Berbagai class Event.

<b>Class</b>	<b>Deskripsi</b>
wxActivateEvent	Aktifasi window atau aplikasi
wxCalendarEvent	Digunakan dengan wxCalendarCtrl
wxCalculateLayoutEvent	Kalkulasi layout window
wxCloseEvent	Close window atau end session
wxCommandEvent	Event dari berbagai kontrol standar
wxContextMenuEvent	Event untuk context menu
wxDateEvent	Digunakan dengan wxDatePickerCtrl
wxDialUpEvent	Event yang dikirim oleh wxDialUpManager
wxDropFilesEvent	Drop file
wxEraseEvent	Background erase
wxEvent	Base class untuk event
wxFindDialogEvent	Event yang dikirim oleh wxFindReplaceDialog
wxFocusEvent	Window focus
wxKeyEvent	Penekanan tombol (keypress)
WxIconizeEvent	Event iconize/restore
wxIdleEvent	Idle event
wxInitDialogEvent	Inisialisasi dialog
WxJoystickEvent	Event joystick
wxListEvent	Event list control
wxMaximizeEvent	Event maximize
wxMenuEvent	Event untuk menu
wxMouseCaptureChangedEvent	Perubahan mouse capture



<b>Class</b>	<b>Deskripsi</b>
wxMouseEvent	Event mouse
wxMoveEvent	Event move
WxNotebookEvent	Event untuk notebook control
wxNotifyEvent	Event notifikasi
wxPaintEvent	Event paint
wxProcessEvent	Process ending
wxQueryLayoutInfoEvent	Query informasi layout
wxScrollEvent	Scroll dari scrollbar, slider dan spin button
WxScrollWinEvent	Event dari scrolled window
wxSizeEvent	Event size
wxSocketEvent	Event socket
wxSpinEvent	Event dari wxSpinButton
wxSplitterEvent	Event dari wxSplitterWindow
wxSysColourChangedEvent	Perubahan warna sistem
wxTimerEvent	Timer expiration
wxTreeEvent	Event untuk tree control
wxUpdateUIEvent	Perubahan pada user interface
WxWindowCreateEvent	Pembuatan window
wxWindowDestroyEvent	Penghapusan window
wxWizardEvent	Event wizard

## 5.10 Validator

Berbagai window validator.

<b>Class</b>	<b>Deskripsi</b>
wxValidator	Base class untuk validator
wxTextValidator	Validator untuk text control
wxGenericValidator	Validator untuk generic control

## 5.11 Struktur Data

Berbagai class struktur data yang disediakan oleh wxWidgets.

<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxCmdLineParser	Command line parser
wxDateSpan	Interval waktu logikal
wxDateTime	Class untuk memanipulasi date/time
wxArray	Implementasi dynamic array
wxArrayString	Kontainer untuk wxString
wxHashMap	Implementasi hashmap
wxHashSet	Implementasi hashset
wxHashTable	Implementasi hash table (deprecated, gunakanlah hashmap)
wxList	linked list
wxLongLong	Tipe integer 64 bit portabel
wxNode	Node pada linked list (wxList)
wxObject	Root untuk sebagian besar class wxWidgets
wxPathList	Class untuk membantu pencarian multiple path
wxPoint	Merepresentasikan sebuah point
wxRect	Merepresentasikan sebuah rectangle
wxRegEx	Regular expression
wxRegion	Merepresentasikan region
wxString	String
wxStringTokenizer	Interpretasi string sebagai daftar token
wxRealPoint	Representasi point menggunakan floating point
wxSize	Merepresentasikan size
wxTimeSpan	Interval waktu
wxURI	Merepresentasikan Uniform Resource Identifier
wxVariant	Class varian (menyimpan tipe yang dapat berubah pada runtime)

## 5.11.1 wxDateTime

- wxDateTime
  - include files:
    - wx/datetime.h

### 5.11.1.1 0\_date\_time (console)

Aplikasi demo untuk bekerja dengan wxDateTime. Menampilkan tanggal dan waktu aktif dalam timezone UTC dan UTC+7.

#### 5.11.1.1.1 Struktur direktori

```
date_time/0_date_time/  
    build/  
        date_time  
    src/  
        app.cpp  
    makefile.gtk
```

#### 5.11.1.1.2 Source code

Source code akan melibatkan file berikut:

- app.cpp: program utama

##### 5.11.1.1.2.1 src/app.cpp

```
#include "wx/wx.h"  
#include "wx/datetime.h"
```

```
int main ()  
{
```

```
    wxDateTime now = wxDateTime :: Now ();
```

```
wxString temp;

temp = now.Format ( _( "now (UTC): %c" ), wxDateTime :: UTC ).c_str () ;

std :: cout << temp.mb_str () << std :: endl;

temp = now.Format ( _( "now (UTC+7): %c" ), wxDateTime :: GMT7 ).c_str () ;

std :: cout << temp.mb_str () << std :: endl;

return 0;
}
```

### 5.11.1.1.3 Kompilasi

#### 5.11.1.1.3.1 Kompilasi di Linux (GTK+)

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT =
```

```
PROGRAM = date_time
OBJECTS = src/app.o
```

```
# system
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES: .o .cpp
```

```
.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<
```

```
all: $(PROGRAM)
```

```
$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o
```

```
clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

#### **5.11.1.2 1\_date\_time (console)**

Aplikasi demo untuk bekerja dengan wxDateTime. Menampilkan tanggal dan waktu aktif dalam timezone UTC dan UTC+7, dalam format yang lebih menarik dibanding contoh sebelumnya.

##### **5.11.1.2.1 Struktur direktori**

```
date_time/1_date_time/
    build/
        date_time
    src/
        app.cpp
    makefile.gtk
```

##### **5.11.1.2.2 Source code**

Source code akan melibatkan file berikut:

- app.cpp: program utama

###### **5.11.1.2.2.1 src/app.cpp**

```
#include "wx/wx.h"
#include "wx/datetime.h"
```

```
int main ()
{

    wxDateTime now = wxDateTime :: Now ();

    wxString temp;

    temp = now.Format ( _( "now (UTC):%nDay:%A%nDate:%d%nMonth:%B%nYear:%Y
%nTZ:%Z%nTime:%H:%M:%S%n" ), wxDateTime :: UTC ).c_str () ;

    std :: cout << temp.mb_str () << std :: endl;

    temp = now.Format ( _( "now (UTC+7):%nDay:%A%nDate:%d%nMonth:%B%nYear:
%Y%nTZ:%Z%nTime:%H:%M:%S%n" ), wxDateTime :: GMT7 ).c_str () ;

    std :: cout << temp.mb_str () << std :: endl;

    return 0;
}
```

### 5.11.1.2.3 Kompilasi

#### 5.11.1.2.3.1 Kompilasi di Linux (GTK+)

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT =
```

```
PROGRAM = date_time
OBJECTS = src/app.o
```

```
# system
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES:    .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all:  $(PROGRAM)

$(PROGRAM):  $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

### 5.11.2 wxDateSpan

- wxDateTime
  - include files:
    - wx/datetime.h

#### 5.11.2.1 0\_date\_span (console)

Aplikasi demo untuk bekerja dengan wxDateSpan. Menampilkan tanggal aktif dan tanggal 2 bulan kemudian.

### 5.11.2.1.1 Struktur direktori

```
date_span/0_date_span/  
    build/  
        date_span  
    src/  
        app.cpp  
    makefile.gtk
```

### 5.11.2.1.2 Source code

Source code akan melibatkan file berikut:

- app.cpp: program utama

#### 5.11.2.1.2.1 src/app.cpp

```
#include "wx/wx.h"  
#include "wx/datetime.h"
```

```
int main ()  
{
```

```
    wxDateTime date1 = wxDateTime :: Now ();
```

```
    wxDateTime date2 = wxDateTime :: Now ();
```

```
    wxDateSpan ds;
```

```
    date2 = date2.Add ( ds.Months( 2 ) );
```

```
    wxString temp;
```

```
    temp = date1.Format ( _( "now (UTC): %A, %d/%m/%Y" ), wxDateTime :: UTC ).c_str  
();
```

```
    std :: cout << temp.mb_str () << std :: endl;
```



```
temp = date2.Format ( _( "2 month later (UTC): %A, %d/%m/%Y" ), wxDateTime ::
UTC ).c_str () ;

std :: cout << temp.mb_str () << std :: endl;

return 0;
}
```

### 5.11.2.1.3 Kompilasi

#### 5.11.2.1.3.1 Kompilasi di Linux (GTK+)

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT =
```

```
PROGRAM = date_span
OBJECTS = src/app.o
```

```
# system
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES: .o .cpp
```

```
.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<
```

```
all: $(PROGRAM)
```

```
$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
```

```
mkdir -p build/$(WXPLAT)
mv $(PROGRAM) build/$(WXPLAT)
rm -f src/*.o
```

clean:

```
rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

## 5.11.3 wxTimeSpan

- wxTimeSpan
  - include files:
    - wx/datetime.h

### 5.11.3.1 0\_time\_span (console)

Aplikasi demo untuk bekerja dengan wxTimeSpan. Menampilkan tanggal dan waktu aktif (UTC), dan tanggal dan waktu sepuluh jam kemudian (UTC).

#### 5.11.3.1.1 Struktur direktori

```
time_span/0_time_span/
    build/
        time_span
    src/
        app.cpp
    makefile.gtk
```

#### 5.11.3.1.2 Source code

Source code akan melibatkan file berikut:

- app.cpp: program utama

### 5.11.3.1.2.1 src/app.cpp

```
#include "wx/wx.h"
#include "wx/datetime.h"

int main ()
{

    wxDateTime date1 = wxDateTime :: Now ();

    wxDateTime date2 = wxDateTime :: Now ();

    date2 = date2.Add ( wxTimeSpan :: Hours ( 10 ) );

    wxString temp;

    temp = date1.Format ( _( "now (UTC): %A, %d/%m/%Y %H:%M:%S" ), wxDateTime ::
UTC ).c_str () ;

    std :: cout << temp.mb_str () << std :: endl;

    temp = date2.Format ( _( "10 hours later (UTC): %A, %d/%m/%Y %H:%M:%S" ),
wxDateTime :: UTC ).c_str () ;

    std :: cout << temp.mb_str () << std :: endl;

    return 0;
}
```

### 5.11.3.1.3 Kompilasi

#### 5.11.3.1.3.1 Kompilasi di Linux (GTK+)

```
# simple makefile for simple wx project
```

# (c) Nop, 5 Feb 2006

# variables

WXCONFIG = wx-config

WXPLAT =

PROGRAM = time\_span

OBJECTS = src/app.o

# system

CXX = \$(shell \$(WXCONFIG) --cxx)

# implementation

.SUFFIXES: .o .cpp

.cpp.o :

\$(CXX) -c `\$(WXCONFIG) --cxxflags` -o \$@ \$<

all: \$(PROGRAM)

\$(PROGRAM): \$(OBJECTS)

\$(CXX) -o \$(PROGRAM) \$(OBJECTS) `\$(WXCONFIG) --libs`

mkdir -p build/\$(WXPLAT)

mv \$(PROGRAM) build/\$(WXPLAT)

rm -f src/\*.o

clean:

rm -f src/\*.o build/\$(WXPLAT)/\$(PROGRAM)

## 5.11.4 wxArrayString

- wxArrayString
  - include files:
    - wx/arrstr.h

#### **5.11.4.1 0\_array\_string (console)**

Aplikasi demo untuk bekerja dengan wxArrayString.

##### **5.11.4.1.1 Struktur direktori**

```
array_string/0_array_string/  
    build/  
        array_string  
    src/  
        app.cpp  
    makefile.gtk
```

##### **5.11.4.1.2 Source code**

Source code akan melibatkan file berikut:

- app.cpp: program utama

###### **5.11.4.1.2.1 src/app.cpp**

```
#include "wx/wx.h"  
#include "wx/arrstr.h"
```

```
int main ()  
{  
  
    wxArrayString arr;  
  
    const int max = 20;  
  
    int i;
```

```
for ( i = 0; i < max; i++ )
{
    wxString temp;

    temp.Printf ( _( "Test %d" ), i );

    arr.Add ( temp );
}

for ( i = 0; i < arr.GetCount () ; i++ )
{
    std :: cout << arr.Item ( i ).mb_str () << std :: endl;
}

return 0;
}
```

### 5.11.4.1.3 Kompilasi

#### 5.11.4.1.3.1 Kompilasi di Linux (GTK+)

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT =
```

```
PROGRAM = array_string
OBJECTS = src/app.o
```

```
# system
CXX = $(shell $(WXCONFIG) --cxx)
```

# implementation

.SUFFIXES: .o .cpp

.cpp.o :  
\$(CXX) -c `\$(WXCONFIG) --cxxflags` -o \$@ \$<

all: \$(PROGRAM)

\$(PROGRAM): \$(OBJECTS)  
\$(CXX) -o \$(PROGRAM) \$(OBJECTS) `\$(WXCONFIG) --libs`  
mkdir -p build/\$(WXPLAT)  
mv \$(PROGRAM) build/\$(WXPLAT)  
rm -f src/\*.o

clean:  
rm -f src/\*.o build/\$(WXPLAT)/\$(PROGRAM)

### 5.11.5 wxString

- wxString
  - include files:
    - wx/string.h

#### 5.11.5.1 0\_string (console)

Aplikasi demo untuk bekerja dengan wxString.

##### 5.11.5.1.1 Struktur direktori

string/0\_string/

```
build/  
    string  
src/  
    app.cpp  
makefile.gtk
```

### 5.11.5.1.2 Source code

Source code akan melibatkan file berikut:

- app.cpp: program utama

#### 5.11.5.1.2.1 src/app.cpp

```
#include "wx/wx.h"  
#include "wx/string.h"  
  
int main ()  
{  
  
    wxString temp = wxEmptyString;  
  
    temp.Append ( _( "Test" ) );  
  
    std :: cout << temp.mb_str () << std :: endl;  
  
    temp.MakeLower ();  
  
    std :: cout << temp.mb_str () << std :: endl;  
  
    temp.MakeUpper ();  
  
    std :: cout << temp.mb_str () << std :: endl;  
  
    wxString temp2 = temp.Format ( _( "String length: %d" ), temp.Len() );  
  
    std :: cout << temp2.mb_str () << std :: endl;
```



```
    return 0;
}
```

### 5.11.5.1.3 Kompilasi

#### 5.11.5.1.3.1 Kompilasi di Linux (GTK+)

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT =
```

```
PROGRAM = string
OBJECTS = src/app.o
```

```
# system
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES:    .o .cpp
```

```
.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<
```

```
all:  $(PROGRAM)
```

```
$(PROGRAM):  $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o
```

```
clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

### **5.11.5.2 1\_string (console)**

Aplikasi demo untuk bekerja dengan wxString.

#### **5.11.5.2.1 Struktur direktori**

```
string/1_string/  
    build/  
        string  
    src/  
        app.cpp  
    makefile.gtk
```

#### **5.11.5.2.2 Source code**

Source code akan melibatkan file berikut:

- app.cpp: program utama

##### **5.11.5.2.2.1 src/app.cpp**

```
#include "wx/wx.h"  
#include "wx/string.h"  
  
int main ()  
{  
  
    wxString temp = wxEmptyString;  
  
    temp.Append ( _( "This is test string" ) );  
  
    std :: cout << temp.mb_str () << std :: endl;  
  
    wxString temp2 = wxEmptyString;
```

```
temp2.Append ( _( "test" ) );

std :: cout << temp2.mb_str () << std :: endl;

if ( temp.Contains ( temp2 ) )
{
    int pos;
    pos = temp.Find ( temp2.wc_str () );
    std :: cout << temp.mb_str () << " contains " << temp2.mb_str () << std :: endl;
    std :: cout << "found in position " << pos << std :: endl;
}
else
{
    std :: cout << temp.mb_str () << " does not contain " << temp2.mb_str () << std ::
endl;
}

return 0;
}
```

### 5.11.5.2.3 Kompilasi

#### 5.11.5.2.3.1 Kompilasi di Linux (GTK+)

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT =
```

```
PROGRAM = string
OBJECTS = src/app.o
```

```
# system
```

```
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES:    .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all:  $(PROGRAM)

$(PROGRAM):  $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

### **5.11.5.3 2\_string (console)**

Aplikasi demo untuk bekerja dengan wxString.

#### **5.11.5.3.1 Struktur direktori**

```
string/2_string/
    build/
        string
    src/
        app.cpp
    makefile.gtk
```

#### **5.11.5.3.2 Source code**

Source code akan melibatkan file berikut:

- app.cpp: program utama

#### 5.11.5.3.2.1 src/app.cpp

```
#include "wx/wx.h"
#include "wx/string.h"

int main ()
{
    wxString temp = wxEmptyString;

    temp.Append ( _( "ABCDEFGHJIJ" ) );

    std :: cout << "Full String: " << temp.mb_str () << std :: endl;

    wxString temp2 = temp.Left ( 3 );

    std :: cout << "First 3 character: " << temp2.mb_str () << std :: endl;

    wxString temp3 = temp.Right ( 3 );

    std :: cout << "Last 3 character: " << temp3.mb_str () << std :: endl;

    wxString temp4 = temp.Mid ( 3, 3 );

    std :: cout << "Mid (3,3): " << temp4.mb_str () << std :: endl;

    return 0;
}
```

#### 5.11.5.3.3 Kompilasi

##### 5.11.5.3.3.1 Kompilasi di Linux (GTK+)

```
# simple makefile for simple wx project
```

# (c) Nop, 5 Feb 2006

```
# variables
WXCONFIG = wx-config
WXPLAT =

PROGRAM = string
OBJECTS = src/app.o

# system
CXX = $(shell $(WXCONFIG) --cxx)

# implementation

.SUFFIXES: .o .cpp

.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<

all: $(PROGRAM)

$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

#### **5.11.5.4 3\_string (console)**

Aplikasi demo untuk bekerja dengan wxString.

##### **5.11.5.4.1 Struktur direktori**

```
string/3_string/
    build/
        string
```

```
src/  
    app.cpp  
    makefile.gtk
```

#### 5.11.5.4.2 Source code

Source code akan melibatkan file berikut:

- app.cpp: program utama

##### 5.11.5.4.2.1 src/app.cpp

```
#include "wx/wx.h"  
#include "wx/string.h"  
  
int main ()  
{  
  
    wxString temp = wxString();  
  
    temp.Append ( _( "123456789" ) );  
  
    long l1;  
  
    temp.ToLong ( &l1 );  
  
    std :: cout << "Conversion from " << temp.mb_str () << " to long (base 10) : " << l1 <<  
    std :: endl;  
  
    return 0;  
}
```

#### 5.11.5.4.3 Kompilasi

**5.11.5.4.3.1 Kompilasi di Linux (GTK+)**

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT =
```

```
PROGRAM = string
OBJECTS = src/app.o
```

```
# system
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES: .o .cpp
```

```
.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<
```

```
all: $(PROGRAM)
```

```
$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o
```

```
clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```



## 5.12 RTTI (Run Time Type Information)

<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxClassInfo	Menampung Run Time class info
wxObject	Root untuk sebagian besar class wxWidgets

## 5.13 Logging

Dukungan untuk message logging.

<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxLog	Base untuk class log
wxLogStderr	Log message ke STDOUT C stream
wxLogStream	Log message ke iostream C++
wxLogTextCtrl	Log message ke wxTextCtrl
wxLogWindow	Log message ke log frame
wxLogGui	Log target default untuk program GUI
wxLogNull	Sementara meniadakan message logging
wxLogChain	Merantai dua log target
wxLogPassThrough	Filter log message
wxStreamToTextRedirector	Redireksi output ke cout ke wxTextCtrl

## 5.14 Debugging

Debugging aplikasi melalui class, fungsi dan macro.

<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxDebugContext	Menyediakan fasilitas pengecekan memori
wxDebugReport	Base class untuk membuat debug report pada saat program crash
wxDebugReportCompress	Class untuk membuat debug report terkompres

<b>Class</b>	<b>Deskripsi</b>
wxDebugReportUpload	Class untuk mengupload debug report lewat protokol HTTP
wxDebugReportPreview	Base class abstrak untuk mempreview isi debug report
wxDebugReportPreviewStd	Implementasi standar untuk wxDebugReportPreview

## 5.15 Networking

Dukungan networking oleh wxWidgets

<b>Class</b>	<b>Deskripsi</b>
wxDialUpManager	Menyediakan fungsi untuk membuat dan memeriksa status koneksi jaringan
wxIPv4address	Merepresentasikan alamat IP
wxIPaddress	Merepresentasikan alamat IP
wxSocketBase	Socket Base
wxSocketClient	Socket client
wxSocketServer	Socket server
wxSocketEvent	Socket event
wxFTP	Protokol FTP
wxHTTP	Protokol HTTP
wxURL	Merepresentasikan URL

## 5.16 IPC (Interprocess Communication)

IPC berbasis Windows DDE, namun tersedia pada sebagian besar platform dengan TCP.

<b>Class</b>	<b>Deskripsi</b>
wxClient	Merepresentasikan client

<b><i>Class</i></b>	<b><i>Deskripsi</i></b>
wxDDEClient	Merepresentasikan client
wxConnection	Merepresentasikan koneksi
wxDDEConnection	Merepresentasikan koneksi
wxServer	Merepresentasikan server
wxDDEServer	Merepresentasikan server

## **5.17 Framework Document/View**

Dukungan untuk framework Document/View

<b><i>Class</i></b>	<b><i>Deskripsi</i></b>
wxDocument	Merepresentasikan document
wxView	Merepresentasikan view
wxDocTemplate	Mengatur hubungan antara class document dan class view
wxDocManager	Mengatur document dan view di dalam aplikasi
wxDocChildFrame	Frame anak untuk menampilkan view document
wxDocParentFrame	Frame parent mengandung view

## **5.18 Framework Pencetakan**

Dukungan pencetakan wxWidgets

<b><i>Class</i></b>	<b><i>Deskripsi</i></b>
wxPreviewFrame	Frame untuk menampilkan print preview
wxPreviewCanvas	Canvas untuk menampilkan print preview
wxPreviewControlBar	Control bar standard untuk print preview
wxPrintDialog	Print dialog standar
wxPageSetupDialog	Page setup dialog standar
wxPrinter	Class untuk merepresentasikan printer

<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxPrinterDC	DC Printer
wxPrintout	Merepresentasikan printout
wxPrintPreview	Merepresentasikan print preview
wxPrintData	Merepresentasikan informasi dokumen yang sedang dicetak
wxPrintDialogData	Merepresentasikan informasi tentang print dialog
wxPageSetupDialogData	Merepresentasikan informasi tentang page setup dialog

### **5.19 Drag and Drop dan akses clipboard**

<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxDataObject	Data object
WxDataFormat	Merepresentasikan format data
wxTextDataObject	Data text
wxFileDataObject	Data file
wxBitmapDataObject	Data bitmap
wxCustomDataObject	Data custom
wxClipboard	clipboard
wxDropTarget	Drop target
wxFileDropTarget	Drop target file
wxTextDropTarget	Drop target text
wxDropSource	Drop source

### **5.20 Bekerja dengan File**

Beberapa class yang dapat digunakan untuk bekerja dengan file.

<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxFileName	Operasi pada nama file
wxDir	Enumerasi file/direktori

<b><i>Class</i></b>	<b><i>Deskripsi</i></b>
wxDirTraverser	Bekerja dengan wxDir untuk enumerasi file/direktori secara rekursif
wxFile	Low level file I/O
wxFile	Low level file I/O
wxTempFile	Mereplace file yang ada dengan aman
wxTextFile	Bekerja dengan file text (sebagai array baris/string)
wxStandardPaths	Path untuk direktori standar

## **5.21 Stream**

wxWidgets memiliki sejumlah class stream, sebagai alternatif untuk stream standar.

<b><i>Class</i></b>	<b><i>Deskripsi</i></b>
wxStreamBase	Base class untuk stream
wxStreamBuffer	Stream buffer
wxInputStream	Input stream
wxOutputStream	Output stream
wxCountingOutputStream	Menghitung ukuran stream
wxFilterInputStream	Filtered input stream
wxFilterOutputStream	Filtered output stream
wxBufferedInputStream	Buffered input stream
wxBufferedOutputStream	Buffered output stream
wxMemoryInputStream	Memory input stream
wxMemoryOutputStream	Memory output stream
wxDataInputStream	Binary data input stream
wxDataOutputStream	Binary data output stream
wxTextInputStream	Text data input stream
wxTextOutputStream	Text data output stream
wxFileInputStream	File input stream
wxFileOutputStream	File output stream
wxFileInputStream	File input stream

<b><i>Class</i></b>	<b><i>Deskripsi</i></b>
wxFileOutputStream	File output stream
wxTempFileOutputStream	Mereplace file yang ada dengan aman
wxStringInputStream	String input stream
wxStringOutputStream	String output stream
wxZlibInputStream	Zlib input stream
wxZlibOutputStream	Zlib output stream
wxZipInputStream	Zip input stream
wxZipOutputStream	Zip output stream
wxSocketInputStream	Socket input stream
wxSocketOutputStream	Socket output stream

## 5.22 Threading

wxWidgets memiliki sejumlah class untuk bekerja dengan thread.

<b><i>Class</i></b>	<b><i>Deskripsi</i></b>
wxThread	Thread
wxThreadHelper	Mengatur background thread dengan mudah
wxMutex	Mutex
wxMutexLocker	Mutex locker utility
wxCriticalSection	Critical section
wxCriticalSectionLocker	Critical section locker utility
wxCondition	Condition
wxSemaphore	Semaphore

## 5.23 HTML

Berbagai class untuk menampilkan format HTML.

<b><i>Class</i></b>	<b><i>Deskripsi</i></b>
wxHtmlHelpController	HTML Help Controller
wxHtmlWindow	HTML window

<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxHtmlEasyPrinting	Mencetak HTML
wxHtmlPrintout	HTML wxPrintOut generic
wxHtmlParser	HTML parser generic
wxHtmlTagHandler	Tag handler, pluginable ke wxHtmlParser
wxHtmlWinParser	HTML parser untuk wxHtmlWindow
wxHtmlWinTagHandler	Tag handler, pluginable ke wxHtmlWinParser

## 5.24 Virtual File System

Implementasi virtual file system.

<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxFSFile	Merepresentasikan sebuah file dalam VFS
wxFileSystem	Interface utama untuk VFS
wxFileSystemHandler	Meng-announce tipe file system

## 5.25 XRC

Berbagai class untuk bekerja dengan XRC

<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxXmlResource	Class utama untuk bekerja dengan resource
wxXmlResourceHandler	Base class untuk XML resource handler

## 5.26 Online Help

Berbagai class untuk bekerja dengan online help

<i><b>Class</b></i>	<i><b>Deskripsi</b></i>
wxHelpController	Mengontrol help window
wxHtmlHelpController	HTML Help Controller
wxContextHelp	Memungkinkan aplikasi dalam mode context-sensitive help
wxContextHelpButton	Button untuk bekerja dengan wxContextHelp

<b>Class</b>	<b>Deskripsi</b>
wxHelpProvider	Context-Sensitive Help Provider (abstract)
wxSimpleHelpProvider	Context-Sensitive Help Provider sederhana
wxHelpControllerHelpProvider	Context-Sensitive Help Provider dengan help controller
wxToolTip	tooltip

## 5.27 Database

Berbagai class untuk bekerja dengan ODBC

<b>Class</b>	<b>Deskripsi</b>
wxDb	ODBC database connection
wxDbTable	Akses database table
wxDbInf	Informasi tentang database connection
wxDbTableInf	Informasi tabel
wxDbColDef	Informasi kolom
wxDbColInf	Informasi kolom
wxDbColDataPtr	Pointer ke dynamic column definition
wxDbColFor	Menangani international formatting (terutama date dan float)
wxDbConnectInf	Menyimpan data yang diperlukan untuk koneksi kepada ODBC data source
wxDbIdxDef	Pembuatan non-primary index

## 5.28 Berbagai class lainnya

<b>Class</b>	<b>Deskripsi</b>
wxApp	application
wxCaret	Caret (cursor)
wxCmdLineParser	Command line parser
wxConfig	Baca/tulis konfigurasi. File INI atau registry



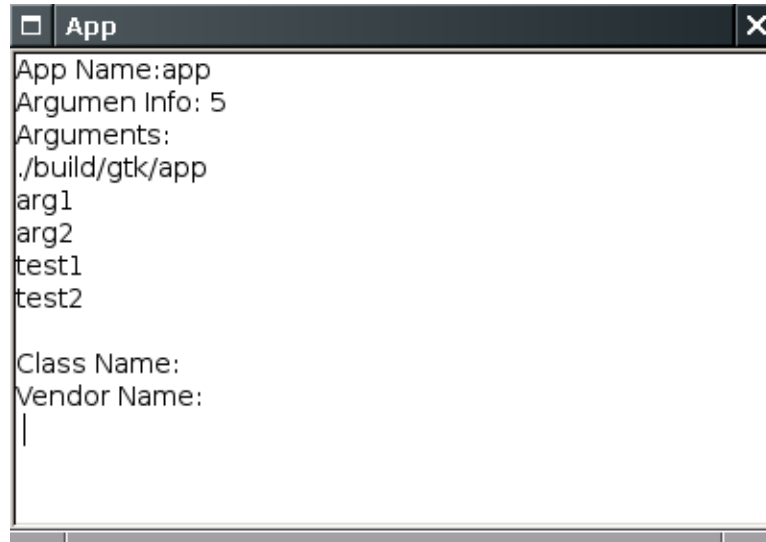
<b><i>Class</i></b>	<b><i>Deskripsi</i></b>
wxDllLoader	Bekerja dengan shared library
wxGLCanvas	Canvas OpenGL
wxGLContext	Memudahkan sharing data resource OpenGL
wxLayoutAlgorithm	Window layout
wxProcess	Proses
wxTimer	Timer
wxStopWatch	Stop watch
wxMimeTypesManager	MIME-type manager
wxSystemSettings	System setting
wxSystemOptions	System option
wxAcceleratorTable	Mengijinkan aplikasi untuk menspesifikasi tabel shortcut keyboard
wxAutomationObject	OLE Automation
wxFontMapper	Font mapper
wxEncodingConverter	Encoding Converter
wxCalendarDateAttr	Digunakan bersama wxCalendarCtrl
wxQuantize	Reduksi warna pada wxImage
wxSingleInstanceChecker	Application single instance checker

### 5.28.1 wxApp

- wxApp
  - include files:
    - wx/app.h
  - derived from:
    - wxEvtHandler
    - wxObject

#### 5.28.1.1 App

Aplikasi demo untuk bekerja dengan wxApp. Mendemonstrasikan bekerja dengan command line argumen sederhana pada wxApp :: OnInit ().



#### 5.28.1.1.1 Struktur direktori

```
app/0_app/  
  build/  
    gtk/  
      app  
  src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
  makefile.gtk
```

#### 5.28.1.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

##### 5.28.1.1.2.1 src/app.h

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

#### **5.28.1.1.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{

    wxString msg;

    Frame * frame = new Frame ( _ ("App") );

    frame -> Show ( true );

    wxString args;

    int counter = 0;

    for ( counter; counter < argc; counter ++ )
    {
        args.Append ( argv ( counter ) );
    }
}
```

```
    argvs.Append ( _ ( "\n" ) );
}

msg.Printf ( _ ( "App Name:%s\nArgumen Info: %d\nArguments:\n%s\nClass Name:
%s\nVendor Name:%s\n " ),
            GetAppName ().c_str (), argc, argvs.c_str (), GetClassName ().c_str (),
            GetVendorName ().c_str () );

wxTextCtrl * textCtrl = new wxTextCtrl ( frame, wxID_ANY, msg, wxDefaultPosition,
            wxDefaultSize, wxTE_MULTILINE | wxTE_READONLY, wxDefaultValidator,
            wxTextCtrlNameStr );

return true;
}
```

#### **5.28.1.1.2.3 src/frame.h**

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame: public wxFrame
{
public:

    Frame ( const wxString& title );

};

#endif // _FRAME_H_
```

#### **5.28.1.1.2.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "frame.h"
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )
{
}
}
```

### 5.28.1.1.3 Kompilasi

#### 5.28.1.1.3.1 Kompilasi di Linux (GTK+)

makefile.gtk:

# simple makefile for simple wx project  
# (c) Nop, 5 Feb 2006

# variables  
WXCONFIG = wx-config  
WXPLAT = gtk

PROGRAM = app  
OBJECTS = src/app.o src/frame.o

# system  
CXX = \$(shell \$(WXCONFIG) --cxx)

# implementation

.SUFFIXES: .o .cpp

.cpp.o :  
 \$(CXX) -c `\$(WXCONFIG) --cxxflags` -o \$@ \$<

all: \$(PROGRAM)

\$(PROGRAM): \$(OBJECTS)  
 \$(CXX) -o \$(PROGRAM) \$(OBJECTS) `\$(WXCONFIG) --libs`  
 mkdir -p build/\$(WXPLAT)  
 mv \$(PROGRAM) build/\$(WXPLAT)  
 rm -f src/\*.o

clean:

```
rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

## Bab 6. Pembuatan custom dialog

Bab ini akan membahas contoh pembuatan custom dialog. Setelah membaca bab ini, developer diharapkan mampu membangun custom dialog sesuai kebutuhan.

Tak jarang, sebuah aplikasi membutuhkan dialog yang dirancang khusus. Sebagai contoh, dialog yang dirancang untuk entry data customer. Dialog tersebut merupakan kebutuhan yang sangat spesifik, dan tentu saja tidak disediakan oleh wxWidgets. Namun, wxWidgets menyediakan semua yang dibutuhkan untuk pembuatan dialog.

Di dalam bab ini, dua buah contoh dialog sederhana akan disajikan. Kedua contoh tersebut akan dibahas dalam subbab terpisah. Untuk mengikuti contoh-contoh yang dibahas, diharapkan developer telah mampu menggunakan berbagai class wxWidgets yang tersedia (seperti wxTextCtrl, wxButton) dan mampu menggunakan sizer (semua contoh yang dibahas akan menggunakan wxBoxSizer).

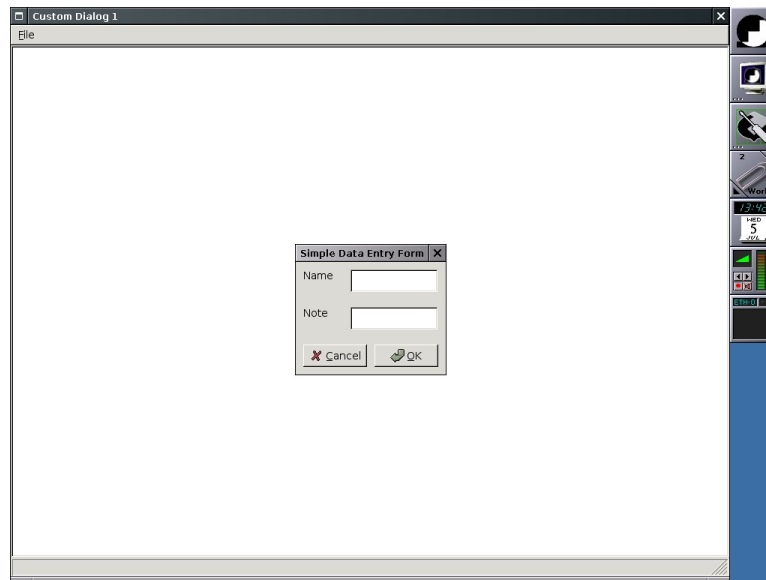
Satu hal yang perlu diperhatikan, wxWidgets menyediakan cukup banyak dialog yang dapat digunakan sesuai kebutuhan. Lihatlah bab tentang pembahasan berbagai class wxWidgets untuk mengetahui apakah dialog yang ingin dibangun telah tersedia. Apabila dialog yang diinginkan telah tersedia, tentu saja developer tidak perlu repot-repot membangun dialog sendiri.

### 6.1 Contoh custom dialog 1

Aplikasi yang dibangun dalam contoh ini akan memiliki fitur-fitur sebagai berikut:

- Memiliki sebuah frame utama, yang memiliki:
  - sebuah wxTextCtrl multiline
  - sebuah menubar yang mengandung menu File, yang terdiri dari:
    - File | Simple data entry. Dapat diaktifkan pula dengan shortcut F2. Akan menampilkan sebuah custom dialog.
    - File | Quit. Dapat diaktifkan pula dengan shortcut CTRL-Q. Akan menterminasi aplikasi.
  - Sebuah statusbar yang akan menampilkan help string dari menu
- Memiliki sebuah custom dialog yang:
  - memiliki dua input, masing-masing diisi dengan wxTextCtrl single line
    - Name (nama)
    - Note (catatan)
  - memiliki dua buah tombol

- OK
- Cancel
- akan ditutup apabila user menekan tombol ESC.
- User bisa mengisi nama dan catatan di custom dialog. Setelah dialog ditutup dengan menekan tombol OK, maka di wxTextCtrl multiline frame utama, akan tertulis semua informasi yang dimasukkan oleh user.



### 6.1.1 Struktur direktori custom\_dialog1

```
custom_dialog1/  
  build/  
    gtk/  
      custom_dialog1  
  src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
    de_dialog.h  
    de_dialog.cpp  
  makefile.gtk
```



## 6.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame
- de\_dialog.h: header class DataEntryDialog (custom dialog)
- de\_dialog.cpp: implementasi class DataEntryDialog (custom dialog)

### 6.1.2.1 src/app.h

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{

public:

    virtual bool OnInit ();

};

#endif // _APP_H_
```

### 6.1.2.2 src/app.cpp

```
#include "wx/wx.h"
#include "app.h"
#include "frame.h"
#include "de_dialog.h"
```

```
DECLARE_APP ( App )
```

```
IMPLEMENT_APP ( App )
```

```
bool App :: OnInit ()  
{  
    Frame * frame = new Frame ( _( "Custom Dialog 1" ) );  
    frame -> Show ( true );  
    return true;  
}
```

Penjelasan source code:

- Perhatikanlah bahwa kita menggunakan header de\_dialog.h

### **6.1.2.3 src/frame.h**

```
#include "de_dialog.h"  
  
#ifndef _FRAME_H_  
#define _FRAME_H_  
  
enum  
{  
    ID_DE_FORM = wxID_HIGHEST + 1,  
  
    ID_TEXTCTRL  
};  
  
class Frame : public wxFrame  
{  
  
public:  
    DataEntryDialog * m_deDialog;  
  
    Frame ( const wxString& title );
```

```
void OnQuit ( wxCommandEvent& event );
```

```
void OnDataEntryDialog ( wxCommandEvent& event );
```

```
private:
```

```
DECLARE_EVENT_TABLE ()
```

```
};
```

```
#endif // _FRAME_H_
```

Penjelasan source code:

- Perhatikanlah bahwa kita menggunakan header de\_dialog.h
- Perhatikanlah juga window ID yang didefinisikan

```
enum
```

```
{  
    ID_DE_FORM = wxID_HIGHEST + 1,
```

```
    ID_TEXTCTRL
```

```
};
```

- ID\_DE\_FORM dimaksudkan untuk digunakan sebagai ID DataEntryDialog
- ID\_TEXTCTRL dimaksudkan untuk digunakan sebagai ID wxTextCtrl multiline pada frame utama
- Class Frame juga memiliki sebuah member berupa DataEntryDialog

```
DataEntryDialog * m_deDialog;
```

#### **6.1.2.4 src/frame.cpp**

```
#include "wx/wx.h"
```

```
#include "frame.h"
```

```
#include "de_dialog.h"
```

```
BEGIN_EVENT_TABLE ( Frame, wxFrame )
```

```
    EVT_MENU ( wxID_EXIT, Frame :: OnQuit )
```

```
    EVT_MENU ( ID_DE_FORM, Frame :: OnDataEntryDialog )
```

```
END_EVENT_TABLE ()
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )  
{
```

```
    wxMenu * fileMenu = new wxMenu ();
```

```
    fileMenu -> Append ( ID_DE_FORM, _( "&Simple Data entry \tF2" ), _( "Simple Data  
entry form" ) );
```

```
    fileMenu -> Append ( wxID_EXIT, _( "&Quit \tCTRL-Q" ), _( "Quit application" ) );
```

```
    wxMenuBar * menuBar = new wxMenuBar ();
```

```
    menuBar -> Append ( fileMenu, _( "&File" ) );
```

```
    SetMenuBar ( menuBar );
```

```
        wxTextCtrl * textCtrl = new wxTextCtrl ( this, ID_TEXTCTRL, wxEmptyString,  
wxDefaultPosition,  
        wxDefaultSize, wxTE_MULTILINE | wxTE_DONTWRAP );
```

```
    CreateStatusBar ();
```

```
}
```

```
void Frame :: OnQuit ( wxCommandEvent& event )  
{
```

```
    Close ();
```

```
}
```

```
void Frame :: OnDataEntryDialog ( wxCommandEvent& event )
{
    m_deDialog = new DataEntryDialog ( this, wxID_ANY, _( "Simple Data Entry
Form" ), wxDEFAULT_DIALOG_STYLE );

    m_deDialog -> SetName ( wxEmptyString );

    m_deDialog -> SetNote ( wxEmptyString );

    if ( m_deDialog -> ShowModal () == wxID_OK )
    {

        wxTextCtrl * t_textCtrl = ( wxTextCtrl * ) FindWindow ( ID_TEXTCTRL );

        wxString temp;

        temp.Printf ( _( "From:%s\nInfo:\nName: %s\nNote: %s\n" ),
m_deDialog -> GetTitle ().c_str (), m_deDialog -> GetName ().c_str (),
m_deDialog -> GetNote ().c_str () );

        t_textCtrl -> SetValue ( temp );

    }

    m_deDialog -> Destroy();
}
```

Penjelasan source code:

- Perhatikanlah bahwa kita menggunakan header de\_dialog.h
- Perhatikanlah window ID yang digunakan pada Frame :: Frame ()
- Perhatikanlah event table:

```
BEGIN_EVENT_TABLE ( Frame, wxFrame )
```

```
    EVT_MENU ( wxID_EXIT, Frame :: OnQuit )
```

```
    EVT_MENU ( ID_DE_FORM, Frame :: OnDataEntryDialog )
```

```
END_EVENT_TABLE ()
```

- OnQuit, handler untuk klik pada menu File | Quit, hanya berisikan satu perintah: Close ();

- Sementara, `OnDataEntryDialog`, yang merupakan inti pembahasan, melakukan hal-hal sebagai berikut:
  - Pertama-tama, sebuah objek `DataEntryDialog` dibuat:

```
m_deDialog = new DataEntryDialog ( this, wxID_ANY, _( "Simple Data Entry Form" ), wxDEFAULT_DIALOG_STYLE );
```

- Kemudian, memanggil `DataEntryDialog::SetName` dan `DataEntryDialog::SetNote`, untuk memberikan `wxEmptyString` sebagai parameter keduanya.

```
m_deDialog -> SetName ( wxEmptyString );
```

```
m_deDialog -> SetNote ( wxEmptyString );
```

- Kemudian, menampilkan dialog secara modal dan memeriksa apakah `ShowModal()` mengembalikan `wxID_OK`. Apabila demikian, maka sebuah pesan yang terdiri dari beberapa baris, yang berisikan informasi yang diinput oleh user, akan ditampilkan sebagai isi dari `wxTextCtrl` pada frame utama.

```
if ( m_deDialog -> ShowModal () == wxID_OK )  
{
```

```
    wxTextCtrl * t_textCtrl = ( wxTextCtrl * ) FindWindow ( ID_TEXTCTRL );
```

```
    wxString temp;
```

```
    temp.Printf ( _( "From:%s\nInfo:\nName: %s\nNote: %s\n" ),  
m_deDialog -> GetTitle ().c_str (), m_deDialog -> GetName ().c_str (),  
m_deDialog -> GetNote ().c_str () );
```

```
    t_textCtrl -> SetValue ( temp );
```

```
}
```

- Perintah berikut dimaksudkan untuk mencari window `wxTextCtrl` dengan memberikan window ID-nya. Hal ini diperlukan agar `wxTextCtrl` utama dapat diakses.

```
wxTextCtrl * t_textCtrl = ( wxTextCtrl * ) FindWindow ( ID_TEXTCTRL );
```

- Terakhir, objek didestroy:

```
m_deDialog -> Destroy();
```

#### **6.1.2.5 src/de\_dialog.h**

```
#include "wx/wx.h"
```

```
#ifndef _DE_DIALOG_H_  
#define _DE_DIALOG_H_
```

```
enum  
{
```

```
    ID_NAME = wxID_HIGHEST + 1,
```

```
    ID_NOTE
```

```
};
```

```
class DataEntryDialog : public wxDialog  
{
```

```
public:
```

```
    DataEntryDialog ( wxWindow* parent, wxWindowID id, const wxString& title, long  
style );
```

```
    void Init ();
```

```
void SetName ( const wxString& name );

wxString GetName () const;

void SetNote ( const wxString& note );

wxString GetNote () const;

void SetDialogValidators ();

void CreateControls ();

wxString m_name;

wxString m_note;

};

#endif // _DE_DIALOG_H_
```

Penjelasan source code:

- Perhatikanlah window ID yang digunakan:

```
enum
{
    ID_NAME = wxID_HIGHEST + 1,
    ID_NOTE

};
```

- ID\_NAME akan digunakan sebagai window ID untuk entry name
- ID\_NOTE akan digunakan sebagai window ID untuk entry note



- Constructor:

`DataEntryDialog ( wxWindow* parent, wxWindowID id, const wxString& title, long style );`

- Akan membutuhkan parameter berupa window parent, window ID, title dan style
- Perhatikan member berikut:

`wxString m_name;`

`wxString m_note;`

- `m_name` dimaksudkan sebagai variabel untuk menampung isi entry name
- `m_note` dimaksudkan sebagai variabel untuk menampung isi entry note

- Perhatikan beberapa method berikut:

`void SetName ( const wxString& name );`

`wxString GetName () const;`

`void SetNote ( const wxString& note );`

`wxString GetNote () const;`

- `SetName ()` digunakan untuk set variabel `m_name`. `GetName()` mengembalikan `m_name`.
- `SetNote ()` digunakan untuk set variabel `m_note`. `GetNote()` mengembalikan `m_note`.
- Perhatikanlah juga beberapa method berikut:

`void Init ();`

`void CreateControls ();`

- `Init()` akan menginisialisasi struktur data yang digunakan.

- CreateControls() akan membangun user interface
- Method berikut ini cukup spesial:  
  
void SetDialogValidators ();
- Apa yang dilakukan method ini adalah:
  - Membuat validator wxTextValidator dengan filter wxFILTER\_ALPHA untuk window dengan ID ID\_NAME, agar hanya bisa menerima alfabet.
  - Membuat validator wxGenericValidator untuk window dengan ID ID\_NOTE
- Apakah validator itu?
  - Dengan adanya validator, dialog akan lebih mudah untuk ditulis
  - validator adalah objek yang dapat dilekatkan ke sebuah control (misal: wxTextCtrl) dan menghubungkan struktur data C++ dan control tersebut.
  - Sekaligus melakukan transfer data dan melakukan validasi.
  - Developer bisa mempergunakan validator yang tersedia atau membuatnya sendiri.

#### **6.1.2.6 src/de\_dialog.cpp**

```
#include "de_dialog.h"  
#include "wx/valgen.h"
```

```
DataEntryDialog :: DataEntryDialog ( wxWindow* parent, wxWindowID id, const  
wxString& title, long style )  
: wxDialog ( parent, id, title, wxDefaultPosition, wxDefaultSize, style, wxEmptyString )
```

```
{
```

```
    Init ();
```

```
    CreateControls ();
```

```
    SetDialogValidators ();
```

```
}
```

```
void DataEntryDialog :: Init ()
```

```
{  
  
    m_name = wxEmptyString;  
  
    m_note = wxEmptyString;  
  
}  
  
void DataEntryDialog :: CreateControls ()  
{  
  
    wxStaticText * stextName = new wxStaticText ( this, wxID_ANY, _( "Name" ) );  
  
    wxStaticText * stextNote = new wxStaticText ( this, wxID_ANY, _( "Note" ) );  
  
    wxTextCtrl * textName = new wxTextCtrl ( this, ID_NAME );  
  
    wxTextCtrl * textNote = new wxTextCtrl ( this, ID_NOTE );  
  
  
    wxButton * btnOK = new wxButton ( this, wxID_OK, _ ( "&OK" ) );  
  
    wxButton * btnCancel = new wxButton ( this, wxID_CANCEL, _ ( "&Cancel" ) );  
  
  
    wxBoxSizer * sizerMain = new wxBoxSizer ( wxVERTICAL );  
  
    wxBoxSizer * sizerName = new wxBoxSizer ( wxHORIZONTAL );  
  
    wxBoxSizer * sizerNote = new wxBoxSizer ( wxHORIZONTAL );  
  
    wxBoxSizer * sizerButton = new wxBoxSizer ( wxHORIZONTAL );  
  
  
    sizerName -> Add ( stextName, 1, wxEXPAND | wxALL, 5 );  
  
    sizerName -> Add ( textName, 2, wxEXPAND | wxALL , 5 );  
  
  
    sizerNote -> Add ( stextNote, 1, wxEXPAND | wxALL, 5 );
```

```
sizerNote -> Add ( textNote, 2, wxEXPAND | wxALL, 5 );
```

```
sizerButton -> Add ( btnCancel, 1, wxEXPAND | wxALL, 5 );
```

```
sizerButton -> Add ( btnOK, 1, wxEXPAND | wxALL, 5 );
```

```
sizerMain -> Add ( sizerName, 1, wxEXPAND | wxALL, 5 );
```

```
sizerMain -> Add ( sizerNote, 1, wxEXPAND | wxALL, 5 );
```

```
sizerMain -> Add ( sizerButton, 1, wxEXPAND | wxALL, 5 );
```

```
SetSizerAndFit ( sizerMain );
```

```
}
```

```
void DataEntryDialog :: SetName ( const wxString& name )
```

```
{
```

```
    m_name = name;
```

```
}
```

```
wxString DataEntryDialog :: GetName () const
```

```
{
```

```
    return m_name;
```

```
}
```

```
void DataEntryDialog :: SetNote ( const wxString& note )
```

```
{
```

```
    m_note = note;
```

```
}
```

```
wxString DataEntryDialog :: GetNote () const
{
    return m_note;
}

void DataEntryDialog :: SetDialogValidators ()
{
    FindWindow ( ID_NAME ) -> SetValidator ( wxTextValidator ( wxFILTER_ALPHA,
&m_name ) );

    FindWindow ( ID_NOTE ) -> SetValidator ( wxGenericValidator ( &m_note ) );
}
```

Penjelasan source code:

- Constructor akan menjalankan:

```
Init ();
```

```
CreateControls ();
```

```
SetDialogValidators ();
```

- Sementara, di dalam Init(), nilai inisial m\_name dan m\_note adalah wxString:

```
void DataEntryDialog :: Init ()
{
    m_name = wxString;

    m_note = wxString;
}
```

- Pembuatan control dan penggunaan sizer dilakukan dalam CreateControls(). Silahkan merujuk kepada bab yang membahas penggunaan sizer untuk informasi lebih lanjut. Di sini, perhatikanlah window ID yang digunakan.
- Perhatikanlah juga cara penggunaan validator:

```
void DataEntryDialog :: SetDialogValidators ()
{
    FindWindow ( ID_NAME ) -> SetValidator ( wxTextValidator ( wxFILTER_ALPHA,
&m_name ) );

    FindWindow ( ID_NOTE ) -> SetValidator ( wxGenericValidator ( &m_note ) );
}
```

- Di sini, terlihat bahwa selain memberikan filter, validator (wxTextValidator) menghubungkan m\_name dan window dengan ID ID\_NAME. Selain itu, validator juga menghubungkan m\_note dengan window dengan ID ID\_NOTE, namun tidak melakukan validasi atau filtering tertentu. WxGenericValidator hanya akan melakukan data transfer tanpa melakukan validasi atau filtering.

## 6.1.3 Kompilasi

### 6.1.3.1 Kompilasi di Linux (GTK+)

Makefile yang digunakan: makefile.gtk:

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006

# variables
WXCONFIG = wx-config
WXPLAT = gtk

PROGRAM = custom_dialog1
OBJECTS = src/app.o src/frame.o src/de_dialog.o

# system
CXX = $(shell $(WXCONFIG) --cxx)
```

# implementation

.SUFFIXES: .o .cpp

.cpp.o :

\$(CXX) -c `\$(WXCONFIG) --cxxflags` -o \$@ \$<

all: \$(PROGRAM)

\$(PROGRAM): \$(OBJECTS)

\$(CXX) -o \$(PROGRAM) \$(OBJECTS) `\$(WXCONFIG) --libs`

mkdir -p build/\$(WXPLAT)

mv \$(PROGRAM) build/\$(WXPLAT)

rm -f src/\*.o

clean:

rm -f src/\*.o build/\$(WXPLAT)/\$(PROGRAM)

Penjelasan makefile:

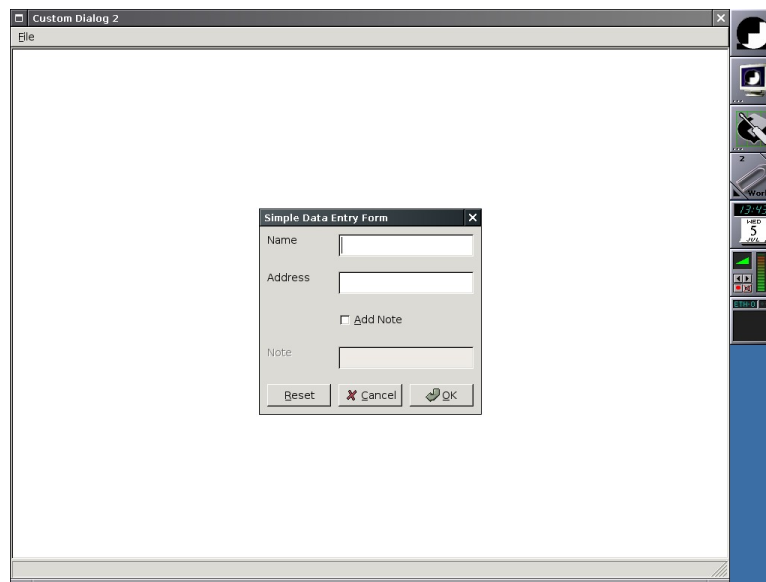
- Harap diperhatikan bahwa kita telah menambahkan src/de\_dialog.o pada OBJECTS

## 6.2 Contoh custom dialog 2

Aplikasi yang dibangun dalam contoh ini akan memiliki fitur-fitur sebagai berikut:

- Memiliki sebuah frame utama, yang memiliki:
  - sebuah wxTextCtrl multiline
  - sebuah menubar yang mengandung menu File, yang terdiri dari:
    - File | Simple data entry. Dapat diaktifkan pula dengan shortcut F2. Akan menampilkan sebuah custom dialog.
    - File | Quit. Dapat diaktifkan pula dengan shortcut CTRL-Q. Akan menterminasi aplikasi.
  - Sebuah statusbar yang akan menampilkan help string dari menu
- Memiliki sebuah custom dialog yang:
  - memiliki tiga input, masing-masing diisi dengan wxTextCtrl single line
    - Name (nama)

- Address (alamat)
- Note (catatan)
- memiliki tiga buah tombol
  - Reset
  - OK
  - Cancel
- memiliki sebuah checkbox Add Note, yang apabila dicheck, akan mengijinkan user untuk memberikan catatan. Apabila tidak diaktifkan, maka input untuk catatan juga akan didisable.
- akan ditutup apabila user menekan tombol ESC.
- User bisa mengisikan nama dan alamat di custom dialog. User juga dapat mengisikan catatan setelah checkbox Add Note dicheck.
- Setelah dialog ditutup dengan menekan tombol OK, maka di wxTextCtrl multiline frame utama, akan tertulis semua informasi yang dimasukkan oleh user.
- Contoh ini mengandung penanganan event di dalam dialog.



### 6.2.1 Struktur direktori custom\_dialog2

```
custom_dialog2/  
  build/  
    gtk/  
      custom_dialog2  
  src/  
    app.h  
    app.cpp  
    frame.h
```



```
frame.cpp
de_dialog.h
de_dialog.cpp
makefile.gtk
```

## 6.2.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame
- de\_dialog.h: header class DataEntryDialog (custom dialog)
- de\_dialog.cpp: implementasi class DataEntryDialog (custom dialog)

Lihatlah juga pembahasan contoh sebelumnya apabila diperlukan.

### 6.2.2.1 *src/app.h*

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:
    virtual bool OnInit ();
};

#endif // _APP_H_
```

### 6.2.2.2 *src/app.cpp*

```
#include "wx/wx.h"
```

```
#include "app.h"
#include "frame.h"
#include "de_dialog.h"
```

```
DECLARE_APP ( App )
```

```
IMPLEMENT_APP ( App )
```

```
bool App :: OnInit ()
{
    Frame * frame = new Frame ( _( "Custom Dialog 2" ) );

    frame -> Show ( true );

    return true;
}
```

### **6.2.2.3 src/frame.h**

```
#include "de_dialog.h"

#ifndef _FRAME_H_
#define _FRAME_H_

enum
{
    ID_DE_FORM = wxID_HIGHEST + 1,

    ID_TEXTCTRL
};

class Frame : public wxFrame
{
public:

    DataEntryDialog * m_deDialog;
```

```
Frame ( const wxString& title );  
  
void OnQuit ( wxCommandEvent& event );  
  
void OnDataEntryDialog ( wxCommandEvent& event );
```

```
private:
```

```
DECLARE_EVENT_TABLE ()
```

```
};
```

```
#endif // _FRAME_H_
```

#### **6.2.2.4 src/frame.cpp**

```
#include "wx/wx.h"  
#include "frame.h"  
#include "de_dialog.h"
```

```
BEGIN_EVENT_TABLE ( Frame, wxFrame )
```

```
    EVT_MENU ( wxID_EXIT, Frame :: OnQuit )
```

```
    EVT_MENU ( ID_DE_FORM, Frame :: OnDataEntryDialog )
```

```
END_EVENT_TABLE ()
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title )  
{
```

```
    wxMenu * fileMenu = new wxMenu ();
```

```
    fileMenu -> Append ( ID_DE_FORM, _( "&Simple Data entry \tF2" ), _( "Simple Data  
entry form" ) );
```

```
    fileMenu -> Append ( wxID_EXIT, _( "&Quit \tCTRL-Q" ), _( "Quit application" ) );
```

```
wxMenuBar * menuBar = new wxMenuBar ();

menuBar -> Append ( fileMenu, _( "&File" ) );

SetMenuBar ( menuBar );

        wxTextCtrl * textCtrl = new wxTextCtrl ( this, ID_TEXTCTRL, wxEmptyString,
wxDefaultPosition,
        wxDefaultSize, wxTE_MULTILINE | wxTE_DONTWRAP );

CreateStatusBar ();

}

void Frame :: OnQuit ( wxCommandEvent& event )
{

    Close ();

}

void Frame :: OnDataEntryDialog ( wxCommandEvent& event )
{

    m_deDialog = new DataEntryDialog ( this, wxID_ANY, _( "Simple Data Entry
Form" ), wxDEFAULT_DIALOG_STYLE );

    m_deDialog -> SetName ( wxEmptyString );

    m_deDialog -> SetNote ( wxEmptyString );

    if ( m_deDialog -> ShowModal () == wxID_OK )
    {

        wxTextCtrl * t_textCtrl = ( wxTextCtrl * ) FindWindow ( ID_TEXTCTRL );

        wxString temp;

        temp.Printf ( _( "From:%s\nInfo:\nName: %s\nAddress: %s\nNote: %s\n" ),
m_deDialog -> GetTitle ().c_str (), m_deDialog -> GetName ().c_str (),
m_deDialog -> GetAddress ().c_str (), m_deDialog -> GetNote ().c_str () );
```

```
t_textCtrl -> SetValue ( temp );  
  
}  
  
m_deDialog -> Destroy();  
}
```

Penjelasan source code:

- Lihatlah pembahasan contoh sebelumnya apabila diperlukan
- Perbedaan Frame :: OnDataEntryDialog () dengan contoh sebelumnya hanyalah penambahan informasi di dalam wxTextCtrl di frame utama.

```
temp.Printf ( _( "From:%s\nInfo:\nName: %s\nAddress: %s\nNote: %s\n" ),  
m_deDialog -> GetTitle ().c_str (), m_deDialog -> GetName ().c_str (),  
m_deDialog -> GetAddress ().c_str (), m_deDialog -> GetNote ().c_str () );
```

### **6.2.2.5 src/de\_dialog.h**

```
#include "wx/wx.h"  
  
#ifndef _DE_DIALOG_H_  
#define _DE_DIALOG_H_  
  
enum  
{  
  
    ID_NAME = wxID_HIGHEST + 1,  
  
    ID_ADDR,  
  
    ID_NOTE,  
  
    ID_RESET,  
  
    ID_ADD_NOTE_TEXT,  
  
    ID_ADD_NOTE,
```

```
};
```

```
class DataEntryDialog : public wxDialog  
{
```

```
public:
```

```
    DataEntryDialog ( wxWindow* parent, wxWindowID id, const wxString& title, long  
style );
```

```
    void Init ();
```

```
    void SetName ( const wxString& name );
```

```
    wxString GetName () const;
```

```
    void SetAddress ( const wxString& address );
```

```
    wxString GetAddress () const;
```

```
    void SetNote ( const wxString& note );
```

```
    wxString GetNote () const;
```

```
    void SetDialogValidators ();
```

```
    void CreateControls ();
```

```
    void OnReset ( wxCommandEvent& event );
```

```
    void OnAddNote ( wxCommandEvent& event );
```

```
    wxString m_name;  
  
    wxString m_addr;  
  
    wxString m_note;
```

```
private:
```

```
    DECLARE_EVENT_TABLE ();
```

```
};
```

```
#endif // _DE_DIALOG_H_
```

Penjelasan source code:

- Lihatlah pembahasan contoh sebelumnya apabila diperlukan
- Beberapa window ID tambahan dibuat:

```
enum  
{  
  
    ID_NAME = wxID_HIGHEST + 1,  
  
    ID_ADDR,  
  
    ID_NOTE,  
  
    ID_RESET,  
  
    ID_ADD_NOTE_TEXT,  
  
    ID_ADD_NOTE,  
  
};
```

- DECLARE\_EVENT\_TABLE diberikan:

private:

```
DECLARE_EVENT_TABLE ();
```

- Member tambahan:

```
void SetAddress ( const wxString& address );
```

```
wxString GetAddress () const;
```

```
wxString m_addr;
```

- Member tambahan (event handler):

```
void OnReset ( wxCommandEvent& event );
```

```
void OnAddNote ( wxCommandEvent& event );
```

- OnReset () digunakan sebagai event handler untuk penekanan tombol Reset
- OnAddNote () digunakan sebagai event handler untuk checkbox Add Note.

#### **6.2.2.6 src/de\_dialog.cpp**

```
#include "de_dialog.h"  
#include "wx/wx.h"  
#include "wx/valgen.h"  
#include "wx/textctrl.h"  
#include "wx/button.h"  
#include "wx/checkbox.h"
```

```
BEGIN_EVENT_TABLE ( DataEntryDialog, wxDialog )
```

```
EVT_BUTTON ( ID_RESET, DataEntryDialog :: OnReset )
```

```
EVT_CHECKBOX ( ID_ADD_NOTE, DataEntryDialog :: OnAddNote )
```



END\_EVENT\_TABLE ()

```
DataEntryDialog :: DataEntryDialog ( wxWindow* parent, wxWindowID id, const  
wxString& title, long style )  
: wxDialog ( parent, id, title, wxDefaultPosition, wxDefaultSize, style, wxEmptyString )
```

```
{
```

```
    Init ();
```

```
    CreateControls ();
```

```
    SetDialogValidators ();
```

```
}
```

```
void DataEntryDialog :: Init ()
```

```
{
```

```
    m_name = wxEmptyString;
```

```
    m_note = wxEmptyString;
```

```
    m_addr = wxEmptyString;
```

```
}
```

```
void DataEntryDialog :: CreateControls ()
```

```
{
```

```
    wxStaticText * stextName = new wxStaticText ( this, wxID_ANY, _( "Name" ) );
```

```
    wxStaticText * stextAddr = new wxStaticText ( this, wxID_ANY, _( "Address" ) );
```

```
    wxStaticText * stextNote = new wxStaticText ( this, ID_ADD_NOTE_TEXT, _( "Note" ) );
```

```
    wxTextCtrl * textName = new wxTextCtrl ( this, ID_NAME );
```

```
wxTextCtrl * textAddr = new wxTextCtrl ( this, ID_ADDR );
```

```
wxTextCtrl * textNote = new wxTextCtrl ( this, ID_NOTE );
```

```
wxCheckBox * checkAddNote = new wxCheckBox ( this, ID_ADD_NOTE, _ ( "&Add  
Note" ) );
```

```
checkAddNote -> SetValue ( false );
```

```
stextNote -> Enable ( checkAddNote -> GetValue () );
```

```
textNote -> Enable ( checkAddNote -> GetValue () );
```

```
wxButton * btnOK = new wxButton ( this, wxID_OK, _ ( "&OK" ) );
```

```
wxButton * btnCancel = new wxButton ( this, wxID_CANCEL, _ ( "&Cancel" ) );
```

```
wxButton * btnReset = new wxButton ( this, ID_RESET, _ ( "&Reset" ) );
```

```
wxBoxSizer * sizerMain = new wxBoxSizer ( wxVERTICAL );
```

```
wxBoxSizer * sizerName = new wxBoxSizer ( wxHORIZONTAL );
```

```
wxBoxSizer * sizerAddr = new wxBoxSizer ( wxHORIZONTAL );
```

```
wxBoxSizer * sizerCheck = new wxBoxSizer ( wxHORIZONTAL );
```

```
wxBoxSizer * sizerAddNote = new wxBoxSizer ( wxHORIZONTAL );
```

```
wxBoxSizer * sizerButton = new wxBoxSizer ( wxHORIZONTAL );
```

```
sizerName -> Add ( stextName, 1, wxEXPAND | wxALL, 5 );
```

```
sizerName -> Add ( textName, 2, wxEXPAND | wxALL, 5 );
```

```
sizerAddr -> Add ( stextAddr, 1, wxEXPAND | wxALL, 5 );
```

```
sizerAddr -> Add ( textAddr, 2, wxEXPAND | wxALL, 5 );
```

```
sizerCheck -> AddStretchSpacer ( 1 );
```

```
sizerCheck -> Add ( checkAddNote, 2, wxEXPAND | wxALL, 5 );
```

```
sizerAddNote -> Add ( stextNote, 1, wxEXPAND | wxALL | wxALIGN_CENTER, 5 );
```

```
sizerAddNote -> Add ( textNote, 2, wxEXPAND | wxALL, 5 );
```

```
sizerButton -> Add ( btnReset, 1, wxEXPAND | wxALL, 5 );
```

```
    sizerButton -> Add ( btnCancel, 1, wxEXPAND | wxALL, 5 );
```

```
    sizerButton -> Add ( btnOK, 1, wxEXPAND | wxALL, 5 );
```

```
sizerMain -> Add ( sizerName, 1, wxEXPAND | wxALL, 5 );
```

```
    sizerMain -> Add ( sizerAddr, 1, wxEXPAND | wxALL, 5 );
```

```
sizerMain -> Add ( sizerCheck, 1, wxEXPAND | wxALL, 5 );
```

```
sizerMain -> Add ( sizerAddNote, 1, wxEXPAND | wxALL, 5 );
```

```
    sizerMain -> Add ( sizerButton, 1, wxEXPAND | wxALL, 5 );
```

```
SetSizerAndFit ( sizerMain );
```

```
}
```

```
void DataEntryDialog :: SetName ( const wxString& name )
```

```
{  
    m_name = name;  
}
```

```
wxString DataEntryDialog :: GetName () const  
{  
  
    return m_name;  
}
```

```
void DataEntryDialog :: SetNote ( const wxString& note )  
{  
  
    m_note = note;  
}
```

```
wxString DataEntryDialog :: GetNote () const  
{  
  
    return m_note;  
}
```

```
void DataEntryDialog :: SetAddress ( const wxString& address )  
{  
  
    m_addr = address;  
}
```

```
wxString DataEntryDialog :: GetAddress () const  
{  
  
    return m_addr;  
}
```

```
void DataEntryDialog :: SetDialogValidators ()
```

```
{

    FindWindow ( ID_NAME ) -> SetValidator ( wxTextValidator ( wxFILTER_ALPHA,
&m_name ) );

    FindWindow ( ID_ADDR ) -> SetValidator ( wxGenericValidator ( &m_addr ) );

    FindWindow ( ID_NOTE ) -> SetValidator ( wxGenericValidator ( &m_note ) );

}

void DataEntryDialog :: OnReset ( wxCommandEvent& event )
{

    Init ();

    wxTextCtrl * textName = ( wxTextCtrl * ) FindWindow ( ID_NAME );
    wxTextCtrl * textAddr = ( wxTextCtrl * ) FindWindow ( ID_ADDR );
    wxTextCtrl * textNote = ( wxTextCtrl * ) FindWindow ( ID_NOTE );
    wxStaticText * stextNote = ( wxStaticText * ) FindWindow ( ID_ADD_NOTE_TEXT );
    wxCheckBox * checkAddNote = ( wxCheckBox * ) FindWindow ( ID_ADD_NOTE );

    textName -> SetValue ( wxEmptyString );
    textAddr -> SetValue ( wxEmptyString );
    textNote -> SetValue ( wxEmptyString );

    checkAddNote -> SetValue ( false );

}
```

```
void DataEntryDialog :: OnAddNote ( wxCommandEvent& event )
{
    wxTextCtrl * textNote = ( wxTextCtrl * ) FindWindow ( ID_NOTE );

    wxStaticText * stextNote = ( wxStaticText * ) FindWindow ( ID_ADD_NOTE_TEXT );

    stextNote -> Enable ( event.IsChecked () );

    textNote -> Enable ( event.IsChecked () );

}
```

Penjelasan source code:

- Lihatlah pembahasan contoh sebelumnya apabila diperlukan
- BEGIN\_EVENT\_TABLE ... END\_EVENT\_TABLE telah disiapkan

```
BEGIN_EVENT_TABLE ( DataEntryDialog, wxDialog )
```

```
    EVT_BUTTON ( ID_RESET, DataEntryDialog :: OnReset )
```

```
    EVT_CHECKBOX ( ID_ADD_NOTE, DataEntryDialog :: OnAddNote )
```

```
END_EVENT_TABLE ()
```

- Untuk mengatur agar sebuah checkbox dicheck, gunakan wxCheckBox::SetValue ( true ). Sebaliknya, wxCheckBox :: SetValue ( false ):

```
checkAddNote -> SetValue ( false );
```

- Untuk mengatur agar sebuah widget di enable/disable, gunakan wxWindow::Enable ()

```
stextNote -> Enable ( checkAddNote -> GetValue () );
```

```
textNote -> Enable ( checkAddNote -> GetValue () );
```

- Pada saat tombol Reset di klik, window akan dicari berdasarkan ID, dan value yang bersesuaian akan di-reset:

```
Init ();
```

```
wxTextCtrl * textName = ( wxTextCtrl * ) FindWindow ( ID_NAME );
```

```
...  
...  
...
```

```
textName -> SetValue ( wxEmptyString );
```

- Pada OnAddNote (), pencarian window dengan ID untuk stextNote dan textNote dilakukan. Setelah itu, status enable/disable diset dengan memanggil : wxCommandEvent::IsChecked () .

## 6.2.3 Kompilasi

### 6.2.3.1 Kompilasi di Linux (GTK+)

Makefile yang digunakan: makefile.gtk:

```
# simple makefile for simple wx project  
# (c) Nop, 5 Feb 2006
```

```
# variables  
WXCONFIG = wx-config  
WXPLAT = gtk
```

```
PROGRAM = custom_dialog2  
OBJECTS = src/app.o src/frame.o src/de_dialog.o
```

```
# system  
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES: .o .cpp
```

```
.cpp.o :  
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<  
  
all: $(PROGRAM)  
  
$(PROGRAM): $(OBJECTS)  
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`  
    mkdir -p build/$(WXPLAT)  
    mv $(PROGRAM) build/$(WXPLAT)  
    rm -f src/*.o  
  
clean:  
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

Penjelasan makefile:

- Harap diperhatikan bahwa kita telah menambahkan src/de\_dialog.o pada OBJECTS



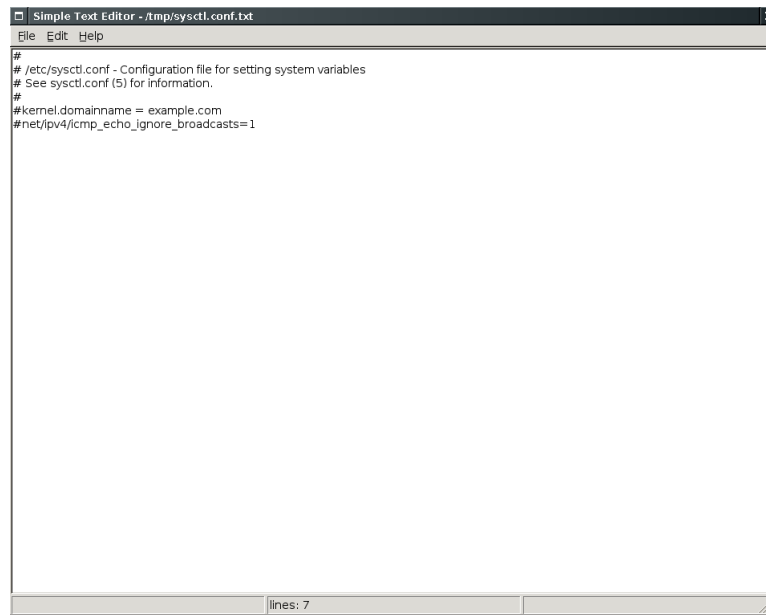
## Bab 7. Proyek Aplikasi

Bab ini membahas dua contoh aplikasi yang dibangun dengan wxWidgets. Setelah membaca bab ini, developer diharapkan semakin siap untuk membangun aplikasi yang dibutuhkan.

### 7.1 Simple Text Editor

Aplikasi pertama yang dibangun adalah simple text editor, yang mana frame utamanya dilengkapi:

- menu:
  - File
    - New: membuat file baru. Apabila ada file yang sedang terbuka dan belum disimpan, maka akses pada menu ini akan memicu konfirmasi untuk menyimpan file aktif.
    - Open: Membuka file. Apabila ada file yang sedang terbuka dan belum disimpan, maka akses pada menu ini akan memicu konfirmasi untuk menyimpan file aktif.
    - Save: Menyimpan file. Apabila belum ada nama file yang diberikan, maka file dialog akan ditampilkan.
    - Quit: Keluar dari aplikasi. Apabila ada file yang sedang terbuka dan belum disimpan, maka akses pada menu ini akan memicu konfirmasi untuk menyimpan file aktif.
  - Edit
    - Copy: mengopi teks
    - Cut: meng-cut teks
    - Paste: mem-paste teks
    - Select all: memilih keseluruhan isi file
    - Word wrap: menentukan apakah word wrap diaktifkan atau tidak
  - Help
    - About: menampilkan informasi about
- statusbar dengan bagian-bagian:
  - bagian 1: tempat menampilkan help string dari menu
  - bagian 2: menampilkan jumlah baris
  - bagian 3: menampilkan tulisan 'modified' apabila file telah termodifikasi. Selebihnya, berisikan string kosong.
- Title bar yang akan menampilkan nama file yang sedang diedit



### 7.1.1 Struktur direktori source code

```
proj_1_simple_editor/  
  build/  
    gtk/  
      textedit  
  src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
    textctrl.h  
    textctrl.cpp  
    makefile.gtk
```

### 7.1.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame
- textctrl.h: header class TextCtrl (diturunkan dari wxTextCtrl)
- textctrl.cpp: implementasi class TextCtrl

### **7.1.2.1 src/app.h**

```
#ifndef _APP_H_
#define _APP_H_

class App : public wxApp
{
public:

    virtual bool OnInit ();
};

#endif // _APP_H_
```

### **7.1.2.2 src/app.cpp**

```
#include "wx/wx.h"
#include "frame.h"
#include "textctrl.h"
#include "app.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
```

```
Frame * frame = new Frame ( _ ("Simple Text Editor") );  
  
frame -> Show ( true );  
  
return true;  
}
```

### **7.1.2.3 src/textctrl.h**

```
#ifndef _TEXTCTRL_H  
#define _TEXTCTRL_H  
  
class TextCtrl : public wxTextCtrl  
{  
public:  
  
    TextCtrl ( wxWindow * parent, wxWindowID id, const wxString& value, long style);  
  
    wxStatusBar * m_statusBar;  
  
    void SetStatusBar ( wxStatusBar* statusBar );  
  
    void OnKeyDown ( wxKeyEvent& event );  
  
    void OnChar ( wxKeyEvent& event );  
  
    long m_linesCount;  
  
private:  
  
    DECLARE_EVENT_TABLE ()  
  
};  
  
#endif // _TEXTCTRL_H
```

Penjelasan source code:

- Aplikasi ini tidak mengharuskan developer untuk menurunkan class sendiri dari

wxTextCtrl. Namun, hal ini ditujukan untuk mendemonstrasikan bahwa developer – apabila diperlukan – bisa menurunkan class yang ada.

#### **7.1.2.4 src/textctrl.cpp**

```
#include "wx/wx.h"  
#include "textctrl.h"
```

```
BEGIN_EVENT_TABLE ( TextCtrl, wxTextCtrl )
```

```
    EVT_KEY_DOWN ( TextCtrl :: OnKeyDown )
```

```
    EVT_CHAR ( TextCtrl :: OnKeyDown )
```

```
END_EVENT_TABLE ()
```

```
TextCtrl :: TextCtrl ( wxWindow * parent, wxWindowID id, const wxString& value, long  
style ) : wxTextCtrl ( parent, id, value, wxDefaultPosition, wxDefaultSize  
, style, wxDefaultValidator, wxTextCtrlNameStr )  
{  
  
}
```

```
void TextCtrl :: OnKeyDown ( wxKeyEvent& event )  
{
```

```
    m_linesCount = GetNumberOfLines ();
```

```
    wxString msgLines;
```

```
    msgLines.Printf ( _( "lines: %ld" ), m_linesCount );
```

```
    m_statusBar -> SetStatusText ( msgLines, 1);
```

```
    if ( IsModified () )  
    {
```

```
        m_statusBar -> SetStatusText ( _( "modified" ), 2);
```

```
    }

    event.Skip ();

}

void TextCtrl :: SetStatusBar ( wxStatusBar* statusBar )
{
    m_statusBar = statusBar;
}
```

Penjelasan source code:

- Pada saat user menekan tombol, jumlah baris akan didapatkan dengan `wxTextCtrl :: GetNumberOfLines()`, kemudian ditampilkan pada statusbar.
- Pada saat user menekan tombol, status modified akan diperiksa dan ditampilkan pada statusbar

#### **7.1.2.5 src/frame.h**

```
#include "textctrl.h"

#ifndef _FRAME_H_
#define _FRAME_H_

enum
{
    ID_WORDWRAP = wxID_HIGHEST + 1
};

class Frame: public wxFrame
{
public:

    TextCtrl * m_textCtrl;

    wxMenu * m_fileMenu;

    wxMenu * m_editMenu;

    wxMenu * m_helpMenu;
```

```
wxMenuBar * m_menuBar;

wxString m_filePath;

wxString m_fileBuffer;

Frame ( const wxString& title );

void OnQuit ( wxCommandEvent& event );

void OnOpen ( wxCommandEvent& event );

void OnSave ( wxCommandEvent& event );

void OnNew ( wxCommandEvent& event );

void OnQuitQuery ( wxCloseEvent& event );

void OnText ( wxCommandEvent& event );

void OnCopy ( wxCommandEvent& event );

void OnCut ( wxCommandEvent& event );

void OnPaste ( wxCommandEvent& event );

void OnSelectAll ( wxCommandEvent& event );

void OnWordWrap ( wxCommandEvent& event );

void OnAbout ( wxCommandEvent& event );


void DoSaveFilePathCheck ();

void DoModifiedCheck ();

private:

    DECLARE_EVENT_TABLE ()

};

#endif // _FRAME_H_
```

Penjelasan source code:

- Di dalam aplikasi ini, kontrol dimasukkan sebagai anggota class. Namun, developer juga tentunya dapat mengeluarkannya dari class dan menggunakan `wxWindow :: FindWindow ()` sebagai gantinya, seperti ditemukan pada banyak contoh di dalam buku ini.
- Method-method berikut ini:
  - `void DoSaveFilePathCheck ()`, digunakan untuk memeriksa adanya nama file yang diasosiasikan dengan file aktif.
  - `void DoModifiedCheck ()`, digunakan untuk memeriksa apakah file dimodifikasi atau tidak

### **7.1.2.6 src/frame.cpp**

```
#include "wx/wx.h"  
#include "frame.h"  
#include "textctrl.h"
```

```
BEGIN_EVENT_TABLE ( Frame, wxFrame )
```

```
    EVT_MENU ( wxID_EXIT, Frame :: OnQuit )
```

```
    EVT_MENU ( wxID_OPEN, Frame :: OnOpen )
```

```
    EVT_MENU ( wxID_SAVE, Frame :: OnSave )
```

```
    EVT_MENU ( wxID_NEW, Frame :: OnNew )
```

```
    EVT_MENU ( wxID_COPY, Frame :: OnCopy )
```

```
    EVT_MENU ( wxID_CUT, Frame :: OnCut )
```

```
    EVT_MENU ( wxID_PASTE, Frame :: OnPaste )
```

```
    EVT_MENU ( wxID_SELECTALL, Frame :: OnSelectAll )
```

```
    EVT_MENU ( ID_WORDWRAP, Frame :: OnWordWrap )
```

```
    EVT_MENU ( wxID_ABOUT, Frame :: OnAbout )
```



```
EVT_CLOSE ( Frame :: OnQuitQuery )
```

```
EVT_TEXT ( wxID_ANY, Frame :: OnText )
```

```
END_EVENT_TABLE ()
```

```
Frame :: Frame ( const wxString& title ) : wxFrame ( NULL, wxID_ANY, title,  
wxDefaultPosition, wxSize ( 640, 480 ) )  
{
```

```
Centre ();
```

```
m_fileMenu = new wxMenu;
```

```
m_fileMenu -> Append ( wxID_NEW, _( "&New \tCtrl-N" ), _( "Create new file" ) );
```

```
m_fileMenu -> Append ( wxID_OPEN, _( "&Open \tCtrl-O" ), _( "Open file..." ) );
```

```
m_fileMenu -> Append ( wxID_SAVE, _( "&Save \tCtrl-S" ), _( "Save file..." ) );
```

```
m_fileMenu -> AppendSeparator ();
```

```
m_fileMenu -> Append ( wxID_EXIT, _( "&Quit \tCtrl-Q" ), _( "Quit application" ) );
```

```
m_editMenu = new wxMenu;
```

```
m_editMenu -> Append ( wxID_COPY, _( "&Copy \tCtrl-C" ), _( "Copy selected text"  
) );
```

```
m_editMenu -> Append ( wxID_CUT, _( "C&ut \tCtrl-X" ), _( "Cut selected text" ) );
```

```
m_editMenu -> Append ( wxID_PASTE, _( "&Paste \tCtrl-V" ), _( "Paste from  
clipboard" ) );
```

```
m_editMenu -> Append ( wxID_SELECTALL, _( "Select &All \tCtrl-A" ), _( "Select  
All" ) );
```

```
m_editMenu -> AppendSeparator ();

m_editMenu -> AppendCheckItem ( ID_WORDWRAP, _( "&Word Wrap" ), _( "Toggle
word wrap" ) );

m_helpMenu = new wxMenu ;

m_helpMenu -> Append ( wxID_ABOUT, _( "&About \tF1" ), _( "About this program..."
) );

m_menuBar = new wxMenuBar;

m_menuBar -> Append ( m_fileMenu, _( "&File" ) );

m_menuBar -> Append ( m_editMenu, _( "&Edit" ) );

m_menuBar -> Append ( m_helpMenu, _( "&Help" ) );

SetMenuBar (m_menuBar);

m_fileBuffer = wxEmptyString;

m_textCtrl = new TextCtrl ( this, wxID_ANY, m_fileBuffer, wxTE_MULTILINE |
wxTE_DONTWRAP );

m_textCtrl -> SetFocus ();

CreateStatusBar ( 3 );

SetStatusText ( _( "lines: 1" ), 1);

m_textCtrl -> SetStatusBar ( GetStatusBar () );

}

void Frame :: OnQuit ( wxCommandEvent& event )
{
```

```
    Close ();

}

void Frame :: OnQuitQuery ( wxCloseEvent& event )
{

    DoModifiedCheck ();

    event.Skip ();
}

void Frame :: OnText ( wxCommandEvent& event )
{

    if ( m_textCtrl -> IsModified () )
    {
        SetStatusText ( _( "modified" ), 2);
    }
}

void Frame :: OnOpen ( wxCommandEvent& event )
{

    DoModifiedCheck ();

    wxFileDialog * openDialog = new wxFileDialog (this, _( "Open file..." ), wxEmptyString
, wxEmptyString, _( "*.txt" ), wxOPEN );

    wxString temp;

    if ( openDialog -> ShowModal () == wxID_OK )
    {

        m_filePath = openDialog -> GetPath ();

        m_textCtrl -> LoadFile ( m_filePath );

        temp = _( "Simple Text Editor - " );

        temp.Append ( m_filePath );
```

```
        SetTitle ( temp );

        SetStatusText ( wxEmptyString, 2);

    }

}

void Frame :: OnSave ( wxCommandEvent& event )
{

    wxString temp;

    DoSaveFilePathCheck ();

    m_textCtrl -> SaveFile ( m_filePath );

    temp = _( "Simple Text Editor - " );

    temp.Append ( m_filePath );

    SetTitle ( temp );

    SetStatusText ( wxEmptyString, 2);

}

void Frame :: OnNew ( wxCommandEvent& event )
{

    DoModifiedCheck ();

    m_filePath = wxEmptyString;

    SetTitle ( _( "Simple Text Editor" ) );
```

```
m_textCtrl -> Clear ();

SetStatusText ( wxEmptyString, 2);
}
```

```
void Frame :: DoSaveFilePathCheck ()
{
```

```
    wxFileDialog * saveDialog = new wxFileDialog (this, _( "Save file..." ), wxEmptyString,
wxEmptyString, _( "*.txt" ), wxSAVE );
```

```
    if ( !m_filePath )
    {
        if ( saveDialog -> ShowModal () == wxID_OK )
        {

            m_filePath = saveDialog -> GetPath ();

        }
    }
}
```

```
void Frame :: DoModifiedCheck ()
{
    if ( m_textCtrl -> IsModified () )
    {
```

```
        int newOK;
```

```
        newOK = wxMessageBox ( _( "Active file is still open and modified. Do you want
to save?" ),
```

```
                                _( "Confirmation" ), wxOK | wxCANCEL | wxICON_QUESTION, this );
```

```
        if ( newOK == wxOK )
        {
```

```
            DoSaveFilePathCheck ();
```

```
            m_textCtrl -> SaveFile ( m_filePath );
```

```
            SetStatusText ( wxEmptyString, 2);
```

```
    }
```

```
  }
```

```
}
```

```
void Frame :: OnCopy ( wxCommandEvent& event )
```

```
{
```

```
    m_textCtrl -> Copy ();
```

```
}
```

```
void Frame :: OnCut ( wxCommandEvent& event )
```

```
{
```

```
    m_textCtrl -> Cut ();
```

```
}
```

```
void Frame :: OnPaste ( wxCommandEvent& event )
```

```
{
```

```
    m_textCtrl -> Paste ();
```

```
}
```

```
void Frame :: OnSelectAll ( wxCommandEvent& event )
```

```
{
```

```
    m_textCtrl -> SetSelection ( -1, -1 );
```

```
}
```

```
void Frame :: OnAbout ( wxCommandEvent& event )
```

```
{
```

```
    wxMessageBox ( _( "Simple Text Editor\n(c) Noprianto, June 2006." ),  
                  _( "About" ), wxOK | wxICON_INFORMATION, this );
```

```
}
```

```
void Frame :: OnWordWrap ( wxCommandEvent& event)
{
    m_fileBuffer = m_textCtrl -> GetValue ();

    m_textCtrl -> Destroy ();

    if ( m_editMenu -> IsChecked ( ID_WORDWRAP ) )
    {

        m_textCtrl = new TextCtrl ( this, wxID_ANY, m_fileBuffer, wxTE_MULTILINE |
wxTE_WORDWRAP );

    }
    else
    {
        m_textCtrl = new TextCtrl ( this, wxID_ANY, m_fileBuffer, wxTE_MULTILINE |
wxTE_DONTWRAP );
    }

    m_textCtrl -> MarkDirty ();

    SetStatusText ( _( "modified" ), 2);
}
```

Penjelasan source code:

- Teknik yang digunakan untuk mengaktifkan/menonaktifkan word wrap adalah dengan menampung terlebih dahulu isi text control, mendestroy text control yang ada dan membuat yang baru dengan window style dilengkapi/tidak dilengkapi word wrap.
- Setiap kali word wrap diaktifkan/dinonaktifkan, status text control selalu dirty (modified).

### 7.1.3 Kompilasi

### **7.1.3.1 Kompilasi di Linux (GTK+)**

makefile.gtk:

```
# simple makefile for simple wx project
# (c) Nop, 5 Feb 2006
```

```
# variables
WXCONFIG = wx-config
WXPLAT = gtk
```

```
PROGRAM = textedit
OBJECTS = src/app.o src/frame.o src/textctrl.o
```

```
# system
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES: .o .cpp
```

```
.cpp.o :
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<
```

```
all: $(PROGRAM)
```

```
$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o
```

```
clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

## **7.2 Simple Image Viewer**

Aplikasi kedua yang dibangun adalah simple image view, yang mana frame utamanya dilengkapi:

- menu:

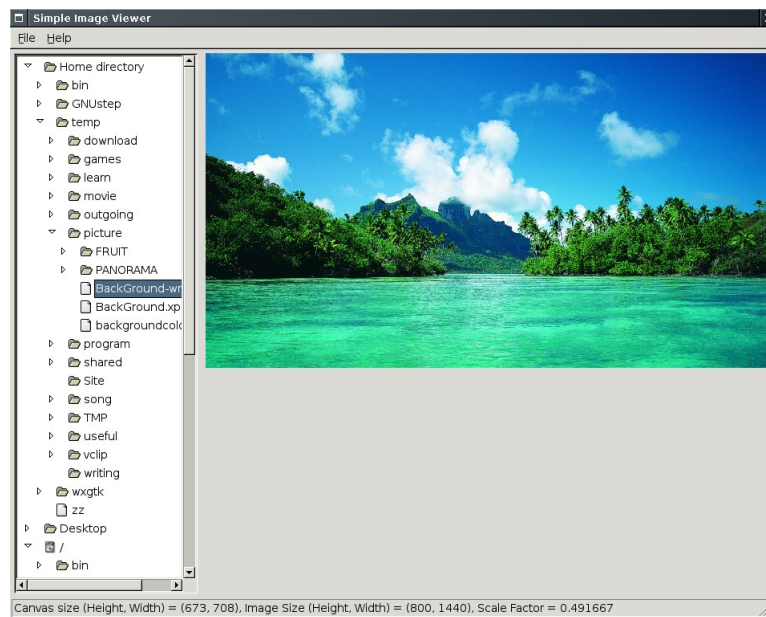


- File
  - Quit: Keluar dari aplikasi.
- Help
  - About: menampilkan informasi about
- statusbar, yang akan menampilkan informasi:
  - Ukuran canvas
  - Ukuran gambar
  - Faktor scale
- Tree view di panel kiri (generic directory control). User dapat memilih file yang ingin ditampilkan dari tree view ini.

Sebagai catatan, dalam menampilkan gambar, aplikasi ini akan:

- Menampilkan gambar dalam ukuran apa adanya apabila ukuran gambar lebih kecil sama dengan ukuran canvas (scale factor : 1)
- Menampilkan gambar dalam ukuran sesuai ukuran canvas apabila ukuran gambar lebih besar dari ukuran canvas (scale factor < 1). Namun, rasio panjang dan lebar gambar akan tetap dipertahankan.

Apabila ada format gambar yang tidak diketahui, maka aplikasi akan diam, tanpa pesan apapun. Aplikasi telah menginisialisasikan semua format yang dikenal oleh wxWidgets.



## 7.2.1 Struktur direktori source code

```
proj_2_simple_imgviewer/  
    build/
```

```
        gtk/  
            imgviewer  
src/  
    app.h  
    app.cpp  
    frame.h  
    frame.cpp  
makefile.gtk
```

## 7.2.2 Source code

Source code akan melibatkan empat file berikut:

- app.h: header class App
- app.cpp: implementasi class App
- frame.h: header class Frame
- frame.cpp: implementasi class Frame

### 7.2.2.1 *src/app.h*

```
#ifndef _APP_H_  
#define _APP_H_  
  
class App : public wxApp  
{  
  
public:  
    virtual bool OnInit ();  
  
};  
  
#endif // _APP_H_
```

### **7.2.2.2 *src/app.cpp***

```
#include "wx/wx.h"
#include "app.h"
#include "frame.h"

DECLARE_APP ( App )

IMPLEMENT_APP ( App )

bool App :: OnInit ()
{
    Frame * frame = new Frame ( _( "Simple Image Viewer" ) );

    frame -> Show ( true );

    return true;
}
```

### **7.2.2.3 *src/frame.h***

```
#ifndef _FRAME_H_
#define _FRAME_H_

class Frame : public wxFrame
{
public:
    wxGenericDirCtrl * m_dirCtrl;

    wxPanel * m_panel;

    wxSize m_canvasSize;
```

```
    wxStaticBitmap * m_bitmap;

    wxBoxSizer * m_sizerMain;

    Frame ( const wxString& title );

    void OnQuit ( wxCommandEvent& event );

    void OnAbout ( wxCommandEvent& event );

    void OnFileSelect ( wxTreeEvent& event );

    void OnSize ( wxSizeEvent& event );

private:

    DECLARE_EVENT_TABLE ();

};

#endif // _FRAME_H_
```

#### **7.2.2.4 src/frame.cpp**

```
#include "wx/wx.h"
#include "wx/dirctrl.h"
#include "wx/gdicmn.h"
#include "wx/filename.h"
#include "wx/image.h"
#include "wx/statbmp.h"
#include "frame.h"

BEGIN_EVENT_TABLE ( Frame, wxFrame )

    EVT_MENU ( wxID_EXIT, Frame :: OnQuit )

    EVT_MENU ( wxID_ABOUT, Frame :: OnAbout )

    EVT_SIZE ( Frame :: OnSize )
```

END\_EVENT\_TABLE ()

```
Frame :: Frame ( const wxString &title ) : wxFrame ( NULL, wxID_ANY, title )
{

    wxSize frameSize = wxSize ( 800, 600 );

    SetSize ( frameSize );

    Centre ();

    wxMenu * fileMenu = new wxMenu;

    fileMenu -> Append ( wxID_EXIT, _( "&Quit \tCTRL-Q" ), _( "Quit application" ) );

    wxMenu * helpMenu = new wxMenu;

    helpMenu -> Append ( wxID_ABOUT, _( "&About \tF1" ), _( "About this
program" ) );

    wxMenuBar * menuBar = new wxMenuBar ;

    menuBar -> Append ( fileMenu, _( "&File" ) );

    menuBar -> Append ( helpMenu, _( "&Help" ) );

    SetMenuBar ( menuBar );

    m_panel = new wxPanel ( this );

    m_bitmap = new wxStaticBitmap ( m_panel, wxID_ANY, wxBitmap () );

    m_dirCtrl = new wxGenericDirCtrl ( this, wxID_ANY, wxDirDialogDefaultFolderStr,
wxDefaultPosition,
        wxDefaultSize, wxDIRCTRL_3D_INTERNAL | wxSUNKEN_BORDER );
```

```
m_sizerMain = new wxBoxSizer ( wxHORIZONTAL );

m_sizerMain -> Add ( m_dirCtrl, 1, wxEXPAND | wxALL, 5 );

m_sizerMain -> Add ( m_panel, 3, wxEXPAND | wxALL, 5 );

SetSizer ( m_sizerMain );

CreateStatusBar ( 1 );

Connect ( m_dirCtrl -> GetTreeCtrl () ->GetId (),
wxEVT_COMMAND_TREE_SEL_CHANGED,
wxTreeEventHandler ( Frame :: OnFileSelect ) );
}

void Frame :: OnQuit ( wxCommandEvent& event )
{
    Close ();
}

void Frame :: OnAbout ( wxCommandEvent& event )
{
    wxMessageBox ( _( "Simple Image Viewer\n(c) Noprianto, June 2006." ),
        _( "About this program" ), wxOK | wxICON_INFORMATION, this );
}

void Frame :: OnFileSelect ( wxTreeEvent& event )
{
    wxString filePath = m_dirCtrl -> GetPath ();

    wxStructStat buf;

    wxStat ( filePath, & buf );
```

```
m_bitmap -> SetBitmap ( wxBitmap ( 1, 1 ) );

SetStatusText ( wxEmptyString, 0 );

if ( buf.st_mode & S_IFREG )
{
    m_canvasSize = m_panel -> GetSize ();

    wxInitAllImageHandlers ();

    wxLogNull logno;

    wxImage * img = new wxImage ( filePath, wxBITMAP_TYPE_ANY );

    if ( img -> Ok () )
    {
        int imgHeight = img -> GetHeight ();

        int imgWidth = img -> GetWidth ();

        float scaleFactor = 1;

        if ( imgWidth > m_canvasSize.GetWidth () || imgHeight >
m_canvasSize.GetHeight () )
        {
            if ( imgWidth >= imgHeight )
            {
                scaleFactor = m_canvasSize.GetWidth () / ( float ) imgWidth;
            }
            else
            {
                scaleFactor = m_canvasSize.GetHeight () / ( float ) imgHeight;
            }
        }
    }
}
```

```
    }

    int newImgWidth = ( int ) round ( scaleFactor * imgWidth ) ;

    int newImgHeight = ( int ) round ( scaleFactor * imgHeight ) ;

    img -> Rescale ( newImgWidth, newImgHeight );
}

wxBitmap bmp ( img );

m_bitmap -> SetBitmap ( bmp );

wxString info;

    info.Printf ( _( "Canvas size (Height, Width) = (%d, %d), Image Size
    (Height, Width) = (%d, %d), Scale Factor = %f" ),
    m_canvasSize.GetHeight (), m_canvasSize.GetWidth (), imgHeight,
    imgWidth, scaleFactor );

    SetStatusText ( info, 0 );

}

img -> Destroy ();

}

}

void Frame :: OnSize ( wxSizeEvent& event )
{

    Layout ();
```



```
m_dirCtrl -> Update ();  
  
m_dirCtrl -> Refresh ();  
  
wxTreeEvent dummy;  
  
OnFileSelect ( dummy );  
  
}
```

## 7.2.3 Kompilasi

### 7.2.3.1 Kompilasi di Linux (GTK+)

makefile.gtk:

```
# simple makefile for simple wx project  
# (c) Nop, 5 Feb 2006
```

```
# variables  
WXCONFIG = wx-config  
WXPLAT = gtk
```

```
PROGRAM = imgviewer  
OBJECTS = src/app.o src/frame.o
```

```
# system  
CXX = $(shell $(WXCONFIG) --cxx)
```

```
# implementation
```

```
.SUFFIXES: .o .cpp
```

```
.cpp.o :  
    $(CXX) -c `$(WXCONFIG) --cxxflags` -o $@ $<
```

```
all: $(PROGRAM)
```

```
$(PROGRAM): $(OBJECTS)
    $(CXX) -o $(PROGRAM) $(OBJECTS) `$(WXCONFIG) --libs`
    mkdir -p build/$(WXPLAT)
    mv $(PROGRAM) build/$(WXPLAT)
    rm -f src/*.o

clean:
    rm -f src/*.o build/$(WXPLAT)/$(PROGRAM)
```

## Bab 8. Membaca Dokumentasi wxWidgets

wxWidgets datang dengan dokumentasi yang terhitung komplit dan up to date. Dokumentasi tersebut boleh dikatakan sebagai alat bantu utama bagi developer dalam membangun aplikasi dengan wxWidgets.

Bab ini akan membahas beberapa hal yang perlu diperhatikan dalam membaca dokumentasi wxWidgets. Setelah membaca bab ini, diharapkan, developer mampu memanfaatkan dokumentasi yang tersedia selama mengembangkan aplikasi.

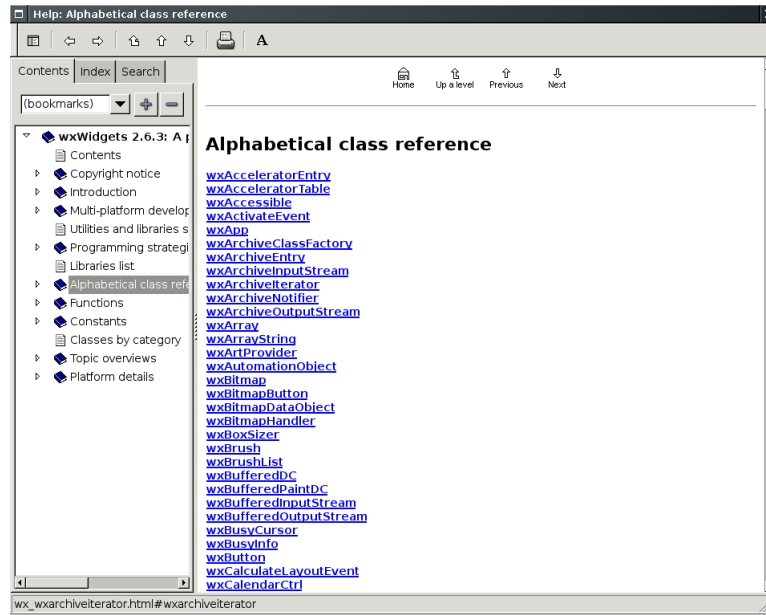
Di dalam distribusi wxWidgets, selain dokumentasi utama, datang pula dokumentasi tambahan untuk program tex2rtf, object graphic library, pustaka gizmoz, dan frame layout.

### 8.1 Format dokumentasi

Dokumentasi wxWidgets datang dalam berbagai format, diantaranya:

- Format HTML.
  - Format universal
  - datang dalam banyak file.
  
- Format PDF.
  - Format siap cetak. Namun, berpikirlah berkali-kali sebelum mencetak, karena dokumentasi wxWidgets dalam format PDF ini berukuran lebih dari 2000 halaman.
  
- Format HTML Help (file CHM).
  - Sangat berguna, terutama bagi pengguna Windows. Pengguna Unix (termasuk Linux) bisa memanfaatkan xchm (<http://xchm.sf.net/>) untuk membuka format ini.
  - Sedekar informasi, xchm juga dibangun dengan wxWidgets.
  - Keunggulan dokumentasi dalam format ini adalah fasilitas searching.

- Format HTB
  - Merupakan file HTML yang dikompres dalam format ZIP
  - Format dokumentasi yang didukung wxWidgets
  - Mendukung fasilitas searching
  - Tersedia program contoh untuk membuka help dalam format ini.



## 8.2 Menggunakan format HTB

Bagi developer yang ingin menggunakan format HTB, sebuah contoh aplikasi telah disediakan. Developer bisa mengkompilasi source code contoh tersebut dan menggunakan aplikasi yang dihasilkan untuk membuka dokumentasi dalam format HTB.

Akseslah contoh tersebut di direktori `samples/html/helpview/`.

Untuk melakukan kompilasi, berikanlah perintah `make`. Contoh:

```
$ make
.././././bk-deps g++ -c -o helpview_helpview.o -D_WXGTK__ -I. -DWXUSINGDLL
-I././././samples -DGTK_NO_CHECK_CASTS -I./././lib/wx/include/gtk2-unicode-
release-2.6 -I./././include -DXTHEADS -pthread -I/usr/include/gtk-2.0 -I/usr/lib/gtk-
2.0/include -I/usr/X11R6/include -I/usr/include/atk-1.0 -I/usr/include/pango-1.0
```

```
-I/usr/include/freetype2 -I/usr/include/glib-2.0 -I/usr/lib/glib-2.0/include
-D_FILE_OFFSET_BITS=64 -D_LARGE_FILES -D_LARGEFILE_SOURCE=1 -O2 -fno-strict-
aliasing -pthread -Wall -Wundef -Wno-ctor-dtor-privacy ./helpview.cpp
g++ -o helpview helpview_helpview.o -pthread -L/usr/X11R6/lib -L../..../lib -Wl,-
rpath,..../lib -lwx_gtk2u_html-2.6 -lwx_gtk2u_core-2.6 -lwx_baseu-2.6
-lwxregexu-2.6 -pthread -L/usr/X11R6/lib -Wl,--version-script,..../version-script -lz -ldl
-lm -Wl,--export-dynamic -pthread -lgtk-x11-2.0 -lgdk-x11-2.0 -latk-1.0 -lgdk_pixbuf-2.0
-lm -lpangoft-1.0 -lpango-1.0 -lpango-1.0 -lgobject-2.0 -lgmodule-2.0 -ldl -lgthread-
2.0 -lglib-2.0 -lXinerama -lXxf86vm -lpng -lz -ljpeg -ltiff
```

Setelah kompilasi, sebuah program dengan nama helpview akan dihasilkan.

Program tersebut dapat dijalankan dengan memberikan argumen dokumentasi-dokumentasi dalam format HTB yang ingin dibuka. Sebagai catatan, program helpview tersebut bisa membuka lebih dari satu file dokumentasi. Contoh:

```
$ ./helpview wx.htb
```

atau

```
$ ./helpview wx.htb tex2rtf.htb ogl.htb fl.htb
```

Ekstensi nama file dokumentasi bisa berupa .htb ataupun .zip.

## 8.3 Hal-hal yang perlu diperhatikan

Berikut ini adalah *contoh beberapa hal* yang bisa diperhatikan, yang tertulis di dalam dokumentasi wxWidgets.

### 8.3.1 File header

Di pembahasan class wxWidgets, sebuah section *Includes files* akan ditampilkan. Perhatikanlah file header yang dibutuhkan untuk class yang ingin digunakan.

Sebagai contoh untuk class wxFrame:

Include files

```
<wx/frame.h>
```

### 8.3.2 Base class

Di pembahasan class wxWidgets, sebuah section *Derived from* akan ditampilkan. Section yang satu ini wajib diperhatikan benar-benar oleh developer.

Terkadang, method tertentu tidak ditemukan di dalam dokumentasi class tertentu, namun ketika ditelusuri ke class orang tuanya, method tersebut ditemukan. Sebagai contoh, di dalam dokumentasi wxButton, tidak ditemukan method Enable (). Namun, karena wxButton adalah turunan dari:

```
wxControl  
wxWindow  
wxEvtHandler  
wxObject
```

maka developer dapat menemukan Enable () di dalam dokumentasi class wxWindow (wxWindow :: Enable () ).

### 8.3.3 Perhatikan constructornya

Sebuah class dapat memiliki lebih dari satu constructor. Perhatikanlah baik-baik constructor yang tersedia, dan pilihlah yang diperlukan. Sebagai contoh, berikut ini adalah constructor milik wxStaticBoxSizer:

```
wxStaticBoxSizer(wxStaticBox* box, int orient)
```

```
wxStaticBoxSizer(int orient, wxWindow *parent, const wxString& label = wxEmptyString)
```

### 8.3.4 Perhatikan function overloading secara umum

Sama seperti pembahasan sebelumnya, berbagai class di wxWidgets menggunakan function overloading, oleh karena itu, pilihlah yang diperlukan. Sebagai contoh, berikut ini adalah wxSizer :: Add ():

```
wxSizerItem* Add(wxWindow* window, const wxSizerFlags& flags)
```

```
wxSizerItem* Add(wxWindow* window, int proportion = 0,int flag = 0, int border = 0,
```

```
wxObject* userData = NULL)
```

```
wxSizerItem* Add(wxSizer* sizer, const wxSizerFlags& flags)
```

```
wxSizerItem* Add(wxSizer* sizer, int proportion = 0, int flag = 0, int border = 0,  
wxObject* userData = NULL)
```

```
wxSizerItem* Add(int width, int height, int proportion = 0, int flag = 0, int border = 0,  
wxObject* userData = NULL)
```

### 8.3.5 Perhatikan bagian see also

Di dokumentasi class, umumnya terdapat section See also. Apabila diperlukan, akseslah class atau topic yang dirujuk sebagai see also. Sebagai contoh, di dalam dokumentasi wxButton, terdapat bagian see also yang merujuk kepada wxBitmapButton.

See also

wxBitmapButton

### 8.3.6 Perhatikan bagian event handling

Apabila bekerja dengan event, maka dokumentasi adalah perlengkapan wajib yang dimiliki. Di dalam dokumentasi berbagai class, terdapat section Event handling yang sangat berguna. Di dalam section tersebut, umumnya disebutkan macro yang bisa digunakan untuk event table atau nama event yang dapat digunakan dalam penanganan event secara dinamis.

Sebagai contoh, class wxButton:

Event handling

EVT\_BUTTON(id, func) Process a wxEVT\_COMMAND\_BUTTON\_CLICKED event, when the button is clicked.

### 8.3.7 Perhatikan isu multiplatform

wxWidgets adalah framework multiplatform. Tidaklah mudah untuk menghadirkan user interface dan look and feel yang sama persis pada setiap platform.

- Beberapa fitur tersedia penuh pada suatu platform, beberapa tersedia secara parsial, beberapa diemulasikan, beberapa mungkin tidak tersedia sama sekali.
- Terkadang, bahkan didokumentasikan pula penggunaan fitur tertentu yang berbahaya.
- Beberapa isu pada platform tertentu juga umumnya dibahas.

Apabila terdapat fitur semacam itu, maka umumnya, di dalam dokumentasi, akan tertulis dengan baik.

Contoh pertama adalah window style pada wxButton:

Window styles

wxBU_LEFT	Left-justifies the label. Windows and GTK+ only.
wxBU_TOP	Aligns the label to the top of the button. Windows and GTK+ only.
wxBU_RIGHT	Right-justifies the bitmap label. Windows and GTK+ only.
wxBU_BOTTOM	Aligns the label to the bottom of the button. Windows and GTK+ only.
wxBU_EXACTFIT	Creates the button as small as possible instead of making it of the standard size (which is the default behaviour ).
wxNO_BORDER	Creates a flat button. Windows and GTK+ only.

Dari dokumentasi tersebut, jelas terlihat bahwa wxNO\_BORDER, wxBU\_BOTTOM, wxBU\_RIGHT, wxBU\_TOP dan wxBU\_LEFT hanya tersedia pada Windows dan GTK+.

Contoh kedua adalah MDI Parent Frame:

The implementation is native in Windows, and simulated under Motif. Under Motif, the child window frames will often have a different appearance from other frames because the window decorations are simulated.



Dari dokumentasi tersebut, terlihat bahwa MDI diimplementasikan secara natif di Windows, dan disimulasikan di Motif.

Contoh ketiga adalah isu wxStaticBitmap pada Windows 9x:

A static bitmap control displays a bitmap. It is meant for display of the small icons in the dialog boxes and is not meant to be a general purpose image display control. In particular, under Windows 9x the size of bitmap is limited to 64\*64 pixels and thus you should use your own control if you want to display larger images portably

Dari dokumentasi tersebut, terlihat bahwa penggunaan class tersebut memiliki isu tertentu pada platform tertentu.

Contoh keempat adalah isu wxStaticBox pada platform tertentu:

Please note that a static box should not be used as the parent for the controls it contains, instead they should be siblings of each other. Although using a static box as a parent might work in some versions of wxWidgets, it results in a crash under, for example, wxGTK.

Also, please note that because of this, the order in which you create new controls is important. Create your wxStaticBox control before any siblings that are to appear inside the wxStaticBox in order to preserve the correct Z-Order of controls.

Perhatikanlah baik-baik isu yang ada, agar portabilitas aplikasi semakin baik. Tidaklah mudah memang, untuk membangun aplikasi multiplatform yang sempurna.