PEMROGRAMAN GUI DI GNU/LINUX MENGGUNAKAN GTK DENGAN GLADE

Ema Utami

Perkembangan Sistem Operasi GNU/Linux semakin pesat, tentu saja perkembangan tersebut diiringi dengan berkembangnya program-program yang berjalan pada sistem operasi GNU/Linux. GNU/Linux sudah sangat dikenal sebagai sistem operasi yang handal untuk keperluan server, program yang mendukung seperti Apache Web Server, PostgreSQL Server, Mail server dan lain sebagainya. Penggunaan GNU/Linux pada awalnya hanya berbasis teks terbatas hanya untuk administrasi seperti *backup system, management user* atau kegiatan lain yang hanya menggunakan mode teks. Namun seiring penggunaan GNU/Linux untuk keperluan dekstop maka banyak program-program yang dibutuhkan untuk berjalan di mode X Window. Sepanjang pengetahuan penulis saat ini belum ada buku yang mengulas tentang pemrograman X Window.

1. Mengenal GTK+

GTK+ (*GIMP Tool Kit*) adalah library untuk membuat antarmuka (GUI) berlisensi GPL (*General Public Lisence*) yang memudahkan untuk membuat *free software* ataupun *commercial software*. Dinamakan *GIMP Tool Kit* karena pada awalnya merupakan pengembangan *General Image Manipulation Program* (GIMP). Saat ini GTK juga digunakan pada banyak *project software* seperti *GNU Network Object Model Environment* (GNOME) project.

GTK telah menerapkan teknik *Object Oriented Programming* (OOP) dan *Application Programmers Interface* (API) menggunakan bahasa C dan menggunakan teknik pemrograman *classes system* dan *callback functio*n. Secara umum GTK+ adalah sebuah *event control* yang artinya GTK akan menunggu pada bagian gtk_main() sampai pada peristiwa (*event*) misal klik tombol mouse yang kemudian akan dikendalikan agar melewati fungsi yang telah ditentukan atau disebut *signal*.

2. Struktur GTK Dasar

GTK merupakan *Application Programmers Interface* (API) berorientasi obyek yang mudah dan sederhana untuk ditulis serta sederhana untuk dimengerti. Sebagai contoh program berikut,

```
#include <gtk/gtk.h>

int main( int   argc, char *argv[] )
{
   GtkWidget *window;

   gtk_init (&argc, &argv);

   window = gtk_window_new (GTK_WINDOW_TOPLEVEL);
   gtk_widget_show  (window);

   gtk_main ();

   return 0;
}
```

Penulisan program untuk GTK secara garis besar sama dengan pemrograman C biasa, yang membedakan hanya pemberian include gtk/gtk.h yang digunakan untuk mendeklarasikan variabel, fungsi, struktur dan lain sebagainya yang akan digunakan dalam aplikasi berbasis GTK. Variabel tipe *GtkWidget* dideklarasikan dan menunjuk pada widget yang telah ada, GtkWidget *window;

```
gtk_init (&argc, &argv);
```

Potongan kode di atas adalah memanggil fungsi gtk_init yang akan dipanggil di semua aplikasi GTK. Pemanggilan fungsi ini akan mengakibatkan beberapa hal seperti nilai warna default dan akan memanggil fungsi lainnya yakni gdk_init yang digunakan untuk menangani *signal handlers* serta melihat argumen yang dilewatkan pada *command line.* Kode selanjutnya adalah,

```
window = gtk_window_new (GTK_WINDOW_TOPLEVEL);
gtk_widget_show (window);
```

Kode tersebut akan membuat jendela (*window*) yang secara default besarnya adalah 200x200 pixel dan kemudian menampilkan jendela tersebut. Kode paling akhir adalah

```
gtk_main ();
```

Fungsi ini merupakan fungsi lain yang pasti akan ada pada setiap aplikasi GTK yang

digunakan untuk menunggu suatu *event.*

3. Kompilasi GTK

Kompilasi kode sumber ini juga sedikit berbeda dengan kompilasi program C biasa. Pengkompilasian program GTK ini perlu ditambahkan opsi `pkg-config --cflags --libs gtk+-2.0` yang digunakan untuk memanggil library GTK. Jadi misalkan kode sumber diberi nama program1.c dan hasil diberikan nama program1 maka dapat digunakan perintah sebagai berikut,

```
gcc program1.c -o program1 `pkg-config --cflags --libs gtk+-2.0`
```

Hasil dari kompilasi program di atas dapat terlihat seperti berikut,

Gambar 1: Tampilan Program 1

Contoh lain perhatikan program berikut, cobalah sambil dipahami kira-kira

maksudnya apa.

```
#include <gtk/gtk.h>

/* Prototipe signal handler */
static gint window_closed(GtkWidget* w, GdkEventAny* e, gpointer data);

int main(int argc, char* argv[])
{
  GtkWidget* window; /* Window */
  GtkWidget* label; /* Label */

/*Inisialisai GTK+ */
  gtk_init(&argc, &argv);

/*Membuat Window */
  window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
  gtk_window_set_title(GTK_WINDOW(window), "Hello");

/* Text di dalam window */
  label = gtk_label_new("Hallo Semua");
  gtk_container_add(GTK_CONTAINER(window), label);

/* Memberikan event handler */
  gtk_signal_connect(GTK_OBJECT(window), "delete_event",
                GTK_SIGNAL_FUNC(window_closed), NULL);

/* Menampilkan window dan semua isinya */
  gtk_widget_show_all(window);
  gtk_main();
 /* Keluar Program */
  return 0;
 }


/* Event handler untuk window_closed */
static gint window_closed(GtkWidget* w, GdkEventAny* e, gpointer data)
{
       gtk_main_quit();
```

```
    return FALSE;
}
```

Hasil dari kompilasi di atas dapat terlihat seperti berikut,

Gambar 2: Tampilan Program 2

Dari dua contoh di atas pada program kedua terdapat teks di window yaitu “Hallo Semua”, di samping itu terdapat perbedaan yang mendasar yaitu adanya *event handler* pada program kedua.

4. Mengenal Glade 2.0

Glade adalah *tool* yang berguna sebagai generator kode antarmuka program yang menggunakan GTK+. Glade dapat menghasilkan kode program C, C++, Perl dan sebagainya. Dengan Glade dapat dibuat suatu antarmuka (GUI) secara visual seperti dalam Delphi ataupun Visual Basic dalam *Windows*. Glade tidak menyediakan *code editor* sehingga perlu alat bantu dari *code editor* seperti *kate*, *advan editor*, *anjuta* dan lain-lain. Dalam Glade, lingkungan kerja terbagi menjadi lima bagian utama yaitu *Main Window, Pallete, Property Editor, Widget Tree, dan Clipboard*. Dalam membuat program menggunakan Glade, kita juga membutuhkan aplikasi tambahan yaitu *Linux Console* (jendela console yang berbasis text) dan *text editor / code editor* seperti vi, kate, gedit, kwrite, anjuta dan lain-lain. Penulis menganjurkan menggunakan *Anjuta* sebagai c*ode editor* karena sudah mendukung GTK+, sehingga terdapat fasilitas untuk melihat fungsi- fungsi yang terdapat dalam GTK+ dan sintaksnya yaitu dengan menekan tombol ‘Ctrl + Enter’.

5. GTK dengan Glade

Pada penjelasan di atas kita telah mengenal bagaimana cara membuat aplikasi GTK. Menuliskan program dan kemudian melakukan kompilasi secara manual yang sangat 'melelahkan'. Pada bab 3 ini kita akan menggunakan Glade yang memungkinkan seorang developer dapat me-design secara cepat dan

efisien suatu aplikasi visual tanpa harus berkutat pada masalah antar mukannya . Dengan kata lain dengan Glade developer dapat :

1.Membuat antarmuka dengan Glade.

2.Membangun *source code* Glade .

3.Mengedit *source code* dengan editor (misal vi).

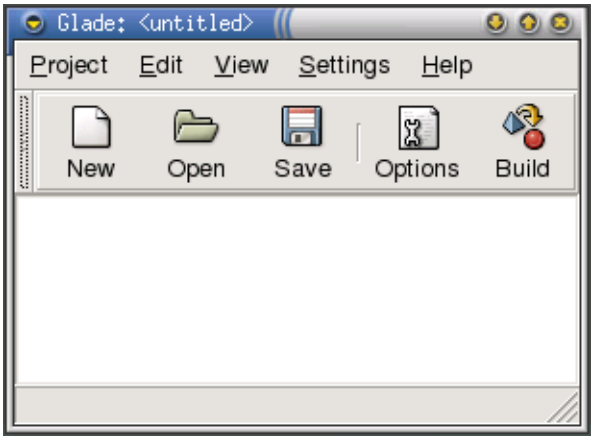
4.Mengkompile.

Untuk menjalankan Glade dapat dilakukan dengan menggunakan start menu dari gnome/KDE. Setelah dijalankan maka akan didapatkan 3 menu sebagai berikut :

1.Main Window
2.Pallete Window
3.Properties Window

Gambar main window terlihat pada gambar paling kiri. Main window digunakan untuk :

1)Menyimpan project
2)Meload project
3)Me*build* project.
4)Melihat *list* dari window aplikasi yang telah dibuat.

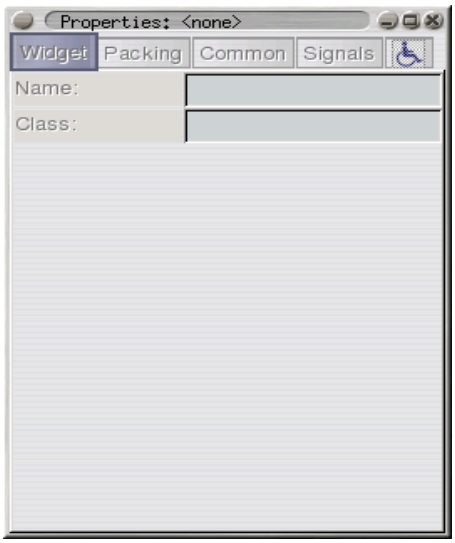


Gambar 3. Main Window

Pallete Window berisi beberapa widgets dari Glade yang dapat digunakan. Ada beberapa 'halaman' widgets yakni 'Gtk+ Basic', 'Gtk+ Additional', 'Gnome' dan 'Deprecated'. Halaman *deprecated* berisi widgets yang telah lama (dan akan dihilangkan). Pallete window juga dapat berisi widget lain tergantung dari instalasi yang dilakukan.Widget dipilih dengan cara mengkilk widget yang diinginkan. Widget window harus dipilih terlebih dahulu sebelum dapat memilih widget yang lain.

Gambar 4: Pallete Window

Window properties adalah tempat untuk melakukan konfigurasi terhadap properti dari *widget* yang dipilih. Pemberian nama, ukuran widget dan lain-lain. Ada beberapa tab yang ada pada window ini yakni: Widget, Packing, Common, Signals dan Accessibility (dengan international simbol). Di bawah widget tab dapat diset mengenai nama dan *basic look* dari suatu widget. Tab Packing digunakan untuk mengubah letak dan ukuran dari suatu widget. Tab Common digunakan untuk menset suatu parameter misalkan ukuran dari window. Tab Signals merupakan tab yang penting yang digunakan untuk menset *callback function* atau *signal handler* untuk setiap event yang diinginkan. Tab accessibility merupakan tab baru yang pada Glade2 yang digunakan untuk memudahkan aksebilitas suatu aplikasi.



Gambar 5:
Properties Windows

6. File - File yang Dihasilkan oleh Glade

Ketika suatu project disimpan maka Glade akan menyimpan 2 file dalam direktori yang telah ditentukan. Secara *default* Glade akan menyimpan dalam direktori Projects dalam direktori home masing-masing user. Project yang dibuat secara *default* akan diberi nama dengan 'Project#' dimana # adalah nomer dari project yang telah dibuat.

Dua file yang dibuat jika menu save dipilih adalah :

Project1.glade
Project1.gladep

Glade akan membangun file-file yang dibutuhkan untuk membuat aplikasi GUI jika opsi build ditekan. File-file tersebut adalah sebagai berikut,

Di bawah direktori project#

1.acconfig.h/autogen.sh/configure.in/Makefile.am/stamp-h.in

File ini digunakan untuk sistem autoconf/autogen/automake dimana akan melakukan kompilasi secara otomatis dan melihat ketergatungannya terhadap suatu *library.* Umumnya pengguna Glade cukup mengetikan ./autogen.sh diikuti dengan enter untuk mengkonfigurasi aplikasi dan membuat file Makefiles.

2.AUTHORS/ChangeLog/README/NEWS

File ini digunakan sebagai suatu "news" jika aplikasi telah diproduksi.

3.Subdirektori macros

Subdirektori ini menyimpan makro yang digunakan oleh sistem autoconf, autogen etc, systems.

4.Subdirektori po

Digunakan untuk menyimpan file translasi po. Pada umumnya digunakan jika suatu aplikasi yang ditranslasikan ke berbagai bahasa.

Dibawah subdirektori src

1. main.c

File ini berisi fungsi main(), dimana merupakan fungsi utama dalam semua program C . File ini dapat diedit sesuai dengan keinginan developer.

2.support.c

File ini bersisi fungsi pendukung dari Glade dan sebaiknya tidak diedit. Salah satu fungsi yang ada disini adalah lookup_widget(), dimana digunakan untuk mencari widget yang sesuai.

3.callbacks.c

File ini digunakan glade untuk menuliskan callbacks dan signal handlers. File ini pada umunya membutuhkan pengeditan dari developer.

4.interface.c

File ini seperti support.c, yang tidak perlu diedit. Glade menggunakan file ini untuk menuliskan fungsi untuk membuat GUI.

5.- Makefile.am/Makefile.in/Makefile

File ini dibuat oleh paket automake/autoconf yang digunakan sebagai pegangan yang akan dibutuhkan oleh gcc untuk mengkompilasi aplikasi.

7. Membuat Antarmuka dengan Glade

Pada contoh program pertama di atas akan kita buat menggunakan Glade. Ikuti langkah-langkah berikut,

Setelah Glade dijalankan maka langkah pertama yang dapat dilakukan adalah membuat Window, seperti pada program 1. Dengan menggunakan Glade maka kita cukup dengan langkah-langkah sebagai berikut

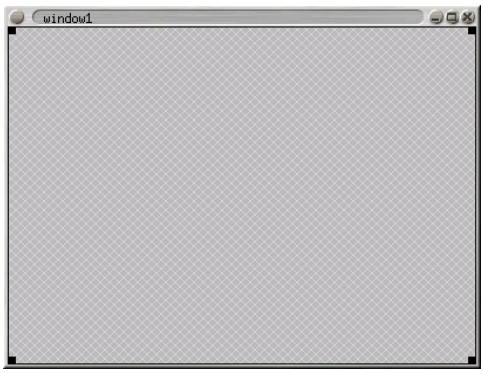
1.Pilih New pada Main Window dan akan pilih opsi New GTK+Project untuk membuat GTK Project, seperti gambar berikut

Gambar 6: Opsi New GTK+Project

2.Klik window widget pada Pallete Window.

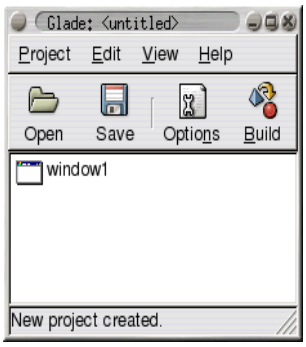
Gambar 7: Window Widget

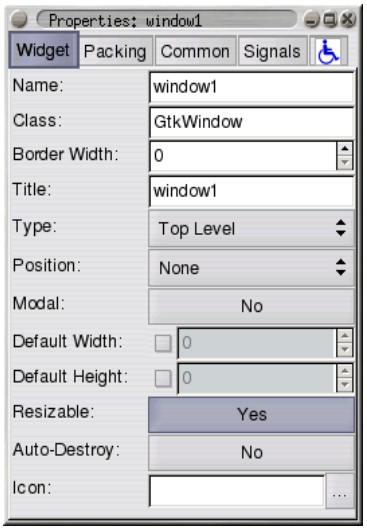
Setelah diklik maka akan muncul sebuah window baru dengan nama window1 seperti gambar di bawah.



Gambar 8: Window Baru

Main Window dan **Properties Window** akan muncul seperti berikut,

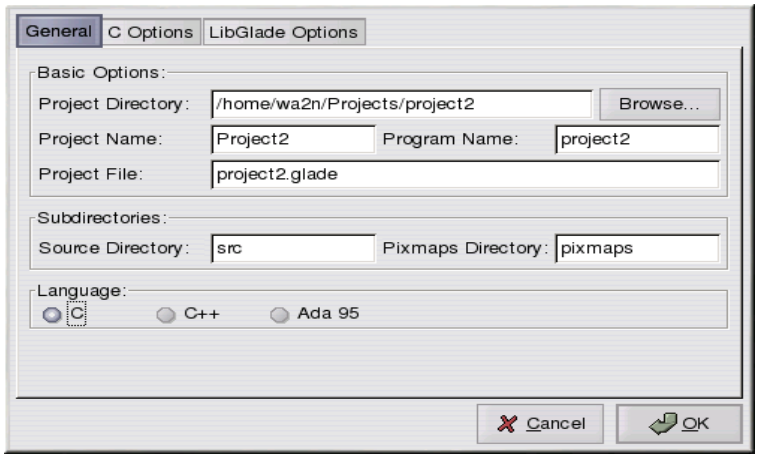




Gambar 9: Main Window dan Properties Window Aktif

3.Pada **Properties Window**, **Title** diisi **Latihan**.

4.Untuk membuat kode sumber dari window yang dibuat dapat diklik tombol **build** pada **Main Window** yang akan menghidupkan menu **save (muncul Project Option)**, seperti gambar berikut,



Gambar 11: Menu Save

Pada Project Directory : /home/group/login/LATIHAN_C/Projects/Latihan1

5.Simpan project tersebut.

6.Secara otomatis Glade akan membuat kode sumber yang ada di bawah direktori src

```
bash-2.05a$ ls src/
Makefile     Makefile.in  callbacks.h  interface.h  support.c
Makefile.am  callbacks.c  interface.c  main.c       support.h
```

File main.c merupakan file yang berisi kode sumber GTK.

```
/*
 * Initial main.c file generated by Glade. Edit as required.
 * Glade will not overwrite this file.
 */

#ifdef HAVE_CONFIG_H
#  include <config.h>
#endif

#include <gtk/gtk.h>

#include "interface.h"
#include "support.h"

int
main (int argc, char *argv[])
{
  GtkWidget *window1;

#ifdef ENABLE_NLS
  bindtextdomain (GETTEXT_PACKAGE, PACKAGE_LOCALE_DIR);
  bind_textdomain_codeset (GETTEXT_PACKAGE, "UTF-8");
  textdomain (GETTEXT_PACKAGE);
#endif
```

```
  gtk_set_locale ();
  gtk_init (&argc, &argv);

  add_pixmap_directory (PACKAGE_DATA_DIR "/" PACKAGE "/pixmaps");

  /*
   * The following code was added by Glade to create one of each component
   * (except popup menus), just so that you see something after building
   * the project. Delete any components that you don't want shown initially.
   */
  window1 = create_window1 ();
  gtk_widget_show (window1);
   gtk_main ();
  return 0;
}
```

Secara garis besar isi file ini akan sama dengan file yang dibuat secara manual pada bab 3.

7.Setelah selesai, masuk ke direktori tempat menyimpan project maka dapat diketikkan ./autogen, misalkan proyek disimpan direktori /home/s1cmp1/ti1234/LATIHAN_C/Projects/Latihan1 maka dapat dilakukan langkah berikut,

```
bash-2.05a$ pwd
   /home/s1cmp1/ti1234/LATIHAN_C/Projects/Latihan1
bash-2.05a$ ./autogen.sh
**Warning**: I am going to run `configure' with no arguments.
If you wish to pass any to it, please specify them on the
`./autogen.sh' command line.

processing .
Creating ./aclocal.m4 ...
Running glib-gettextize...  Ignore non-fatal messages.
Copying file po/Makefile.in.in
```

...
sampai dengan pesan,

```
config.status: executing default-2 commands
Now type `make' to compile.
```

8.Langkah di atas digunakan untuk membuat konfigurasi otomatis sehingga aplikasi akan siap untuk dikompile. Untuk kompilasi dapat dilakukan dengan mengetikkan perintah make pada direktori yang sama. Perintah make akan seperti berikut,

```
bash-2.05a$make
cd . && autoheader
WARNING: Using auxiliary files such as `acconfig.h', `config.h.bot'
WARNING: and `config.h.top', to define templates for `config.h.in'
WARNING: is deprecated and discouraged.
...
make[2]: Leaving directory `/home/wa2n/galdedoc/src/Projects/Latihan1'
make[1]: Leaving directory `/home/wa2n/galdedoc/src/Projects/Latihan1'
bash-2.05a$
```

Perintah make di atas akan melakukan kompilasi terhadap aplikasi yang dibuat dan akan meletakkan *executable file* di subdirektori src dengan nama yang sama dengan nama proyek, pada latihan 1 ini akan diberi nama dengan nama latihan1.

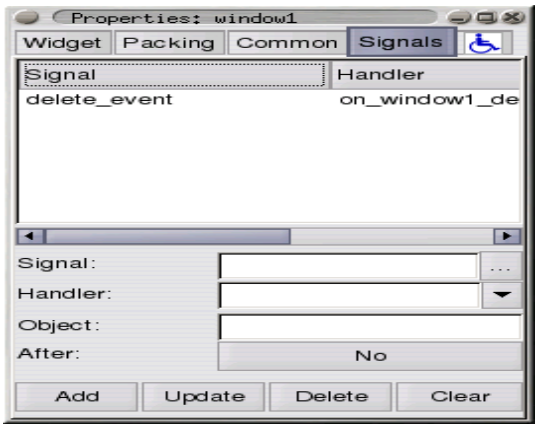
```
bash-2.05a$ ls src/
Makefile     callbacks.c  interface.c  latihan1  support.c
Makefile.am  callbacks.h  interface.h  main.c    support.h
Makefile.in  callbacks.o  interface.o  main.o    support.o
bash-2.05a$
```

9.Untuk menjalakan aplikasi yang telah dibuat dapat dilakukan dengan memanggil *executable file*nya

```
bash-2.05a$ cd src
bash-2.05a$ ./latihan1
```

Hasil yang didapat akan sama dengan program 1 pada bab 3.

Selanjutnya kita akan membuat program 2 di atas dengan menggunakan Glade. Perlu diperhatikan bahwa jika pada penjelasan di atas, *event handler* jadi satu dalam file sumber, maka dengan menggunakan Glade *event handler* akan berada dalam file callbacks.c. Pada saat membuat window maka perlu ditambahkan *signal* dan *event handler*nya melalui Properties Window dan tambahkan *event handler* seperti gambar di bawah.



Gambar 12: Event Handler

8. Penutup

Demikian penuturan mengenal sekilas Glade dengan tujuan mengenalkan dasar-dasar pemrograman berbasis X Window pada sistem operasi GNU/Linux dengan menggunakan GTK dan Glade. Tulisan ini hanya mengulas mengenai penggunaan GTK untuk membuat program berbasis X Window dilanjutkan dengan Glade yang merupakan antar muka yang mempermudah pembuatan program lingkungan X Window. Karena keterbatasan halaman maka untuk koneksi antara GTK dan database MySQL akan disampaikan pada jurnal edisi berikutnya. Semoga tulisan ini dapat bermanfaat bagi pembaca tertarik pemrograman berbasis X Window di GNU/Linux.

9. Referensi

1. Ema Utami dan Suwanto Raharjo, *Belajar C di GNU/Linux*, Penerbit Graha Ilmu Yogyakarta, 2003
2. Ema Utami dan Suwanto Raharjo, *Struktur Data Menggunakan C di GNU/Linux*, Penerbit Andi Yogyakarta, 2004
3. Ema Utami, Tamrizal dan Arief Dwi Cahyadi, *Pemrograman GUI di GNU/Linux Menggunakan Glade*, Penerbit Andi Yogyakarta, 2004
4. Herbert Schildt, *C: The Complete Reference, Fourth Edition*, Osborne/ McGraw-Hill , 2000
5. Herbert Schildt, *C++: The Complete Reference, Fourth Edition*, McGraw-Hill/Osborne, 2003
6. Paul DuBois, *MySQL*, First Printing, Indianapolis, New Riders, 2000
7. *GTK+ Reference,* gtk.org