

KHARISMAN KHOLID HUDAYA

**PEMROGRAMAN
DESKTOP DATABASE
PYTHON-MYSQL**

dengan

BOA CONSTRUCTOR

Pemrograman Desktop Database Python-MySQL
dengan Boa Constructor

Oleh: (Kharisman Kholid Hudaya)

Copyright © 2012 by (Kharisman)

Penerbit

(.....)

Desain Sampul:

(Ahmad Qomari)

KATA PENGANTAR

Alhamdulillah, segenap puji hanya milik Allah *Subhaanahu Wa Ta’ala* semata. Hanya atas pertolongan dan perkenanNyalah buku di hadapan pembaca ini bisa tersaji.

Python adalah bahasa pemrograman yang ringkas dan sederhana. Sangat cocok untuk diperkenalkan pada orang-orang yang mulai belajar bahasa pemrograman.

Metode terbaik dalam mempelajari bahasa pemrograman adalah pelajari contoh program dalam memecahkan suatu masalah, kemudian tiru dan modifikasi. Buku ini menawarkan solusi belajar semacam itu.

Contoh-contoh kasus yang sering ditemui disajikan pemecahannya dalam baris-baris kode program secara sistematis sesuai bab masing-masing. Pembaca bisa langsung mencoba project-project yang ada dalam CD yang disertakan dalam buku ini termasuk software-software yang dibutuhkan.

Selamat bertualang dalam dunia *Python Programming*....

Probolinggo, 6 Maret 2012

Penulis

DAFTAR ISI

Kata Pengantar	3
BAB I : BERKENALAN DENGAN PYTHON	10
1.1 Mengapa Python?	10
1.2 Python Mudah dan Ringkas	11
1.3 Masalah Indentasi pada Python	12
1.4 Case Sensitive pada Python	14
1.5 Software yang Dibutuhkan	15
BAB II : PENGENALAN BOA CONSTRUCTOR	18
2.1 Pendahuluan	18
2.2 Pilihan Instalasi Boa Constructor	19
2.3 Menjalankan Boa Constructor Pertama Kali	20
2.4 Penjelasan Tiga Window	22
2.4.1 Zope Editor	22
2.4.2 Editor	26
2.4.3 Inspector	32
BAB III : PENGENALAN 3 OBJECT UTAMA	34
3.1 Pendahuluan	34
3.2 Memulai Project	34

Memulai Project Baru: Membaca dan Menampilkan Nama.....	35
BAB IV : KONVERSI TIPE DATA	55
4.1 Catatan Awal	55
4.2 Project Menghitung Luas Persegi Panjang	55
4.3 Project Menghitung Usia	58
Kejadian Setelah Tombol Enter Ditekan	59
BAB V : LOGIKA PENCABANGAN, MESSAGE DIALOG, RADIOPUSHBUTTON, DAN CHECKBOX	64
5.1 Pencabangan dengan if	64
5.1.1 Pencabangan 2 Kemungkinan	64
5.1.2 Pencabangan Lebih dari 2 Kemungkinan	68
5.2 RadioButton	71
Aplikasi Nama dan Jenis Kelamin	71
5.3 CheckBox	74
Aplikasi Biodata dengan CheckBox	74
BAB VI : COMBOBOX, LISTBOX, DAN LISTCTRL	78
6.1 wx.ComboBox	78
Aplikasi Biodata dengan ComboBox	79
Aplikasi Kelas Kamar Hotel	81

6.2 wx.ListBox	86
Aplikasi Demo ListBox	87
6.3 wx.ListCtrl	91
Aplikasi Demo wx.ListCtrl	93
Pengaturan Jumlah dan Lebar Kolom	95
BAB VII: PERULANGAN	99
Aplikasi Demo Perulangan	99
BAB VIII: DASAR AKSES KE DATABASE	103
8.1 Prinsip Dasar Akses ke Database	104
Pembuatan Database dan Struktur Tabel	107
8.2 Melihat dan Mencari Data	113
8.2.1 Mencari Data 1 Record	114
8.2.2 Menampilkan Seluruh Data pada wx.ListCtrl	121
8.3 Menyimpan Data	129
8.4 Menghapus Data	134
8.5 Edit/ Update Data	145
8.6 Aplikasi Gabungan (Simpan, Edit, Hapus, Cari)	152
BAB IX : PEMBUATAN MENU	164
9.1 Pendahuluan	164

9.2 Membuat Menu dengan Fasilitas Boa Constructor (<i>Drag and Drop</i>)	167
9.3 Membuat Menu dengan Kode	191
BAB X: AKSES DATABASE LANJUTAN	198
10.1 Pendahuluan	198
10.2 Aplikasi Biodata Database dengan RadioButton	198
10.3 ComboBox (Aplikasi Pemilihan Benua dan Negara)	214
10.4 Pencarian LIKE (Pencarian Judul Buku yang Mengandung Kata Tertentu)	222
10.5 Pengolahan Data Tanggal (<i>Date</i>)	229
10.6 Pengolahan Data Gambar (<i>Image</i>)	244
10.7 Operasi Numerik Desimal	262
10.8 Master Data Lengkap	273
BAB XI: MENCETAK LAPORAN KE EXCEL DENGAN XLWT	297
11.1 Berkenalan dengan XLWT	297
11.2 Perintah Dasar yang Sering Digunakan dalam XLWT	298
11.3 Perintah Menghapus File Lama dan Meluncurkan File Baru	299
11.4 Perintah Lain XLWT untuk Memformat Style	300

dan Menampilkan Gambar

BAB XII : APLIKASI KAMUS BILINGUAL MADURA-INGGRIS	305
12.1 Pendahuluan	305
12.2 Listing Kode wx.App	305
12.3 Struktur Database dan Tabel	306
12.4 Struktur Menu	307
12.5 Frame Isi Data (FrmlInput.py)	310
12.6 Frame Kamus (FrmKamus.py)	320
BAB XIII: APLIKASI PENJUALAN	328
13.1 Pendahuluan	328
13.2 Struktur File	328
13.3 Struktur Database dan Tabel	329
13.4 Listing Kode AppPenjualan.py	330
13.5 Struktur Menu	331
13.6 Frame Pendataan Barang (Dabar.py)	335
13.7 Frame Penjualan	348
13.8 Laporan Data Barang di Bawah Stock Minimum	375
13.9 Frame Laporan Omzet Penjualan Pada Periode Tertentu (LapPenjualan.py)	385

DAFTAR PUSTAKA	400
LAMPIRAN A : INSTALASI IDLE Python, wxPython, dan Python-MySQL Connector di Windows 7	401
LAMPIRAN B: INSTALASI XAMPP	414
LAMPIRAN C: BACKUP (EXPORT) DAN RESTORE (IMPORT) DATABASE	420
LAMPIRAN D : MEMBUKA FILE APLIKASI CONTOH DALAM BUKU	428

BAB I

BERKENALAN DENGAN PYTHON

1.1 Mengapa Python?

Python adalah salah satu bahasa pemrograman *Open Source* yang ringkas, sederhana, dan bisa digunakan di beberapa Sistem Operasi. Python gratis, bisa diunduh siapa saja dengan berbagai versi yang ada. Ia bisa diunduh di www.python.org.

Pemrograman desktop adalah pemrograman yang paling umum digunakan. Pemrograman ini bisa diterapkan pada PC (*Personal Computer*) maupun laptop/ netbook. Berbagai bentuk pemrograman desktop yang populer di antaranya : Sistem Informasi Penjualan (Kasir), Sistem Informasi Perhotelan, Sistem Informasi Perpustakaan, dan semisalnya.

Dulu, pemrograman desktop yang mudah dan praktis adalah dengan Visual Basic 6 (VB6). Saat ini dukungan terhadap VB6 sudah dihentikan oleh perusahaan pembuatnya, Microsoft. VB6 tidaklah gratis. Selain itu, VB6 juga hanya bisa diterapkan pada Sistem Operasi Windows saja.

Hal tersebut berbeda dengan Python. Python bisa dijalankan atau didesain di lingkungan Windows maupun Linux.

Pemrograman desktop di Python bisa menggunakan berbagai library, seperti PyGTK, wxPython, dan sebagainya. Library yang dibahas dalam buku ini adalah wxPython.

Python sebenarnya bisa digunakan untuk berbagai keperluan. Ia bisa digunakan untuk pemrograman Console (tampilan berbasis teks), GUI/ Desktop, mobile (pemrograman di HP), ataupun pemrograman Web.

1.2 Python Mudah dan Ringkas

Python sebenarnya sangat cocok untuk dipelajari oleh programmer pemula. Hal itu karena Python mudah dan ringkas.

Python mudah karena tidak perlu mendeklarasikan variabel seperti pada pemrograman Java, C, Pascal, atau semisalnya. Ia juga sangat ringkas. Untuk pemecahan masalah dengan Python, dibutuhkan jumlah baris kode yang lebih sedikit jika dibandingkan kalau masalah itu dipecahkan dengan bahasa pemrograman lain.

Sebagai contoh, pemrograman *console* untuk menghitung Luas Persegi Panjang. Berikut adalah contoh baris kode pemecahan dengan Python yang dibandingkan dengan kode pada Java.

Kode Menghitung Luas Persegi dengan Python

```
p = input("Panjang: ")  
l = input("Lebar : ")
```

```
Luas = p*I  
print "Luas : ", Luas
```

Kode Menghitung Luas Persegi dengan Java

```
import java.util.Scanner;  
  
public class LuasPersegi {  
  
    public static void main (String[] args) {  
  
        Scanner masukan = new Scanner(System.in);  
  
        int p, l, Luas;  
  
        System.out.print("Panjang :");  
  
        p = masukan.nextInt();  
  
        System.out.print("Lebar :");  
  
        l = masukan.nextInt();  
  
        Luas = p*l;  
  
        System.out.println("Luas Persegi Panjang : "+Luas);  
    } }
```

1.3 Masalah Indentasi Pada Python

Salah satu ciri khas pemrograman Python adalah INDENTASI. Secara sederhana, INDENTASI adalah blok program yang ditandai dengan kesamaan jarak permulaan huruf kode dari ujung kiri layar. Secara asal, kode pada Python harus dimulai

penulisan hurufnya dari ujung kiri layar. Jika terdapat blok baru, seperti pada pencabangan (if), perulangan (while dan for), atau fungsi tertentu (dimulai dengan **def**), maka blok tersebut harus diketikkan lebih menjorok ke dalam. Baris-baris kode yang berada dalam satu blok harus memiliki indentasi yang sama. Permulaan huruf kode pada suatu baris dengan baris di bawahnya yang berada dalam satu blok harus memiliki kesamaan jarak dengan ujung kiri layar.

CATATAN

Jika suatu kode yang seharusnya dimulai dari ujung kiri layar diketikkan lebih menjorok ke dalam, atau sebaliknya, maka akan ada pesan : '*indentation error*'. Ketidakkonsistenan dalam memulai awal pengetikan huruf bisa mengakibatkan error tersebut.

Perhatikan contoh kode Menghitung Luas Persegi di atas.

```
p = input("Panjang: ")  
l = input("Lebar : ")
```

Kode tersebut berarti memasukkan isian dari user ke dalam 2 variabel : **p** dan **l**. Angka yang diketikkan user setelah tulisan “Panjang”, akan disimpan dalam variable **p**, sedangkan angka yang diketikkan setelah tulisan “Lebar” akan dimasukkan dalam variable **l**. Dua baris tersebut dimulai pengetikannya dari ujung kiri layar. Jika tidak konsisten indentasinya, maka akan ada pesan error : '*indentation error*'.

Contoh Indentation Error pada Python:

```
p = input("Panjang: ")  
l = input("Lebar : ")
```

Atau...

```
p = input("Panjang: ")  
l = input("Lebar : ")
```

Kedua contoh *indentation error* di atas menunjukkan bahwa aturan penulisan kode pada Python berbeda dengan pada bahasa pemrograman lain. Kalau pada Java, C, atau yang lain, indentasi bukan masalah. Namun pada Python hal itu adalah masalah. Indentasi tersebut bisa merupakan kekuatan, sekaligus juga kelemahan Python.

Bagi programmer yang tidak biasa mengetikkan kode secara rapi dan konsisten indentasinya, hal itu terasa menyiksa dan menyulitkan. Namun, bagi programmer yang terbiasa disiplin terhadap indentansi, hal itu merupakan keuntungan untuk menghasilkan kode yang lebih ringkas dan efisien. Contoh lebih jauh akan dijelaskan pada bab 3.

1.4 Case Sensitive pada Python

Sama seperti pada Java dan C, Python memiliki sifat Case Sensitive. Artinya, penulisan variabel dengan huruf kecil dianggap beda dengan variabel yang diketikkan dengan huruf besar.

Contoh Kesalahan Kode Akibat Case Sensitive

```
p = input("Panjang: ")  
l = input("Lebar : ")  
luas = p*l  
print "Luas : ", Luas
```

Kode tersebut jika dijalankan akan menampilkan error: **name 'Luas' is not defined**. Artinya, variabel **Luas** tidak didefinisikan sebelumnya/ tidak dikenal, karena yang didefinisikan adalah variabel **luas** (dengan huruf awal kecil).

1.5 Software yang Dibutuhkan

Software yang dibutuhkan untuk membangun sebuah aplikasi desktop di Windows dengan database MySQL adalah:

1. IDLE Python Interpreter.

Ini adalah software dasar yang mendasari priranti untuk membangun sebuah aplikasi. Versi yang digunakan pada buku ini adalah 2.5. Software ini berfungsi menterjemahkan file berekstensi **.py** agar bisa dieksekusi oleh komputer.

2. WxPython

Software ini sebagai library GUI. Versi yang digunakan adalah 2.8 yang compatible dengan IDLE Python 2.5. Nama filenya: **wxPython2.8-**

win32-ansi-2.8.10.1-py25. WxPython tidak bisa berjalan tanpa IDLE Python Interpreter.

3. Boa Constructor

Boa Constructor adalah salah satu IDE (Integrated Development Environment) atau lingkungan pemrograman terintegrasi untuk membangun aplikasi berbasis GUI dengan library wxPython.

Sebenarnya, tanpa menggunakan Boa Constructor, kita bisa membangun aplikasi GUI cukup dengan IDLE Python Interpreter dan wxPython saja, namun kita akan kesulitan mengetikkan secara manual seluruh kode obyek dan desain Frame secara baik.

Dengan Boa Constructor, kita tinggal mendesain, meletakkan dan menyesuaikan obyek yang sesuai, mengatur Inspector (properties)nya, dan menekan tombol Post, selanjutnya, Boa Constructor akan menghasilkan kode-kode secara otomatis dalam bentuk file berekstensi **.py**. Praktis, mendesain Frame secara *drag and drop*.

4. MySQL Server.

Lebih praktis jika kita menginstal database MySQL Server yang sudah dikemas jadi satu dengan PHP dan Apache dalam XAMPP. Kita bisa melakukan administrasi database: menciptakan database, membuat tabel dan

strukturnya, melakukan query langsung ke database, dan semisalnya dengan phpmyadmin.

5. Connector Python ke MySQL

Instalasi software ini akan menghasilkan modul **MySQLdb**. Versi yang digunakan harus kompatibel dengan versi IDLE Python Interpreternya. File yang kita gunakan adalah **MySQL-python-1.2.2.win32-py2.5**

Dalam buku ini versi python yang digunakan adalah 2.5. Jika anda akan menggunakan versi yang lain, pastikan bahwa WxPython dan Connector Python MySQLnya adalah kompatibel dengan versi IDLE Python Interpeternya. Urut-urutan yang terpenting IDLE Python Interpreter diinstal paling awal.

BAB II

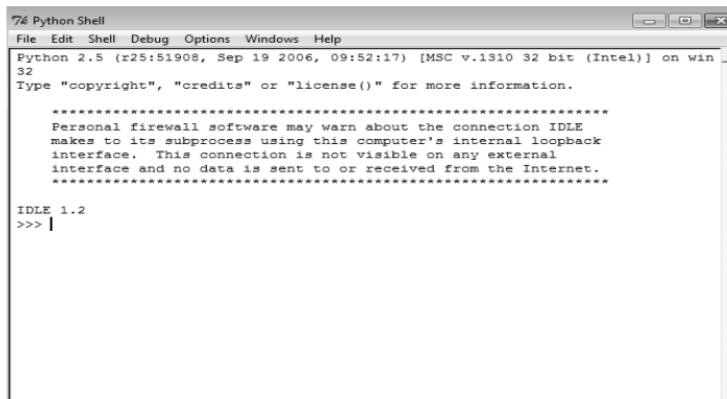
PENGENALAN BOA CONSTRUCTOR

2.1 Pendahuluan

Boa Constructor adalah salah satu IDE (Integrated Development Environment) dalam mengembangkan sebuah program aplikasi desktop dengan library wxpython. Boa Constructor tidak akan bisa berjalan jika library wx belum terinstal di komputer kita.

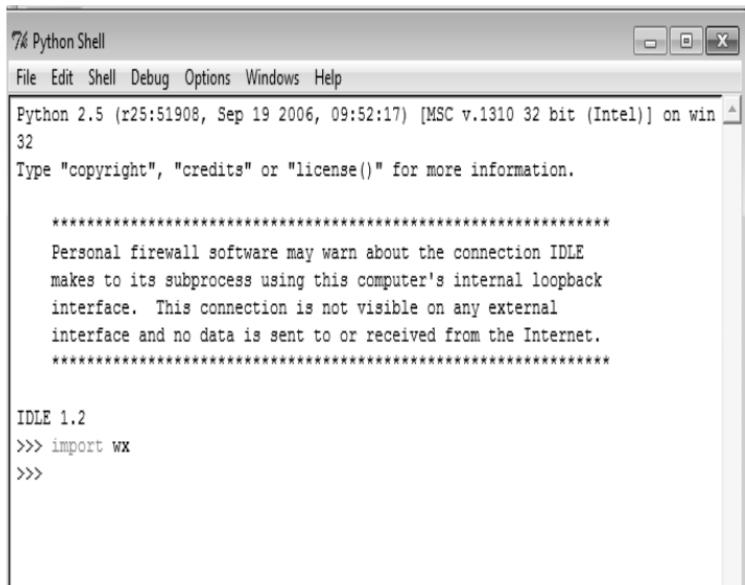
Untuk mengecek apakah library wx sudah terinstal pada komputer kita, langkah-langkahnya adalah:

1. Jalankan IDLE Python. Kali ini contoh kita adalah Python 2.5



Gambar 2.1 Tampilan Awal IDLE Python

2. Ketikkan : **import wx** kemudian tekan tombol enter. Jika tidak ada pesan error, berarti wxpython sudah terinstal dengan baik.



The screenshot shows a Python Shell window titled '74 Python Shell'. The menu bar includes File, Edit, Shell, Debug, Options, Windows, and Help. The main window displays the Python version and build information: 'Python 2.5 (r25:51908, Sep 19 2006, 09:52:17) [MSC v.1310 32 bit (Intel)] on win32'. It also shows a warning message about personal firewalls and IDLE's internal loopback interface. At the bottom, the user has typed '>>> import wx' and '>>>>'.

Gambar 2.2 Pengecekan Instalasi wx

2.2 Pilihan Instalasi Boa Constructor

Boa Constructor bisa didownload dalam 2 pilihan:

- Paket instalasi (.msi atau exe)
- Paket zip atau src

Kami sarankan anda memilih paket zip atau src. Paket ini tidak memerlukan proses instalasi. Cukup mengekstrak folder dan memindahkannya ke dalam directory lain yang kita inginkan, kemudian

menjalankan file **Boa.py**. Paket ini selain bisa dijalankan pada Windows juga bisa diterapkan pada Linux.

Boa Constructor tidak akan bisa dijalankan jika kita belum menginstal IDLE Python.

2.3 Menjalankan Boa Constructor Pertama Kali

Jika anda sudah mendapatkan file zip atau src Boa Constructor (boa-constructor-0.6.1.src), langkah-langkah untuk menjalankannya adalah sebagai berikut:

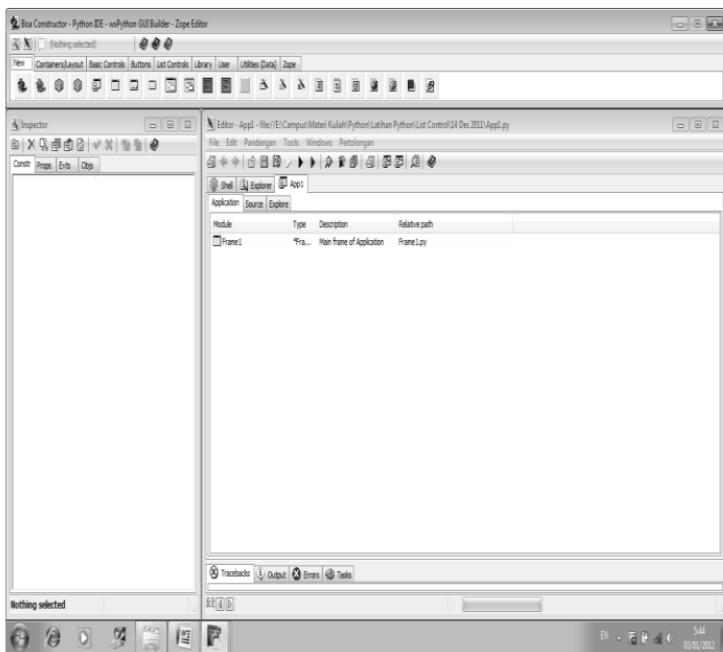
1. Ekstrak folder boa-constructor-0.6.1 ke dalam directory yang kita inginkan.
2. Di dalam folder boa-constructor-0.6.1, pilihlah file **Boa.py**. Jika IDLE Python dan wxpython sudah terinstal, klik dua kali Boa.py. (Bisa juga dengan klik kanan, kemudian **Open With...**, pilih IDLE Python 2.5).
3. *Splash screen* Boa Constructor mulai terlihat loading



Gambar 2.3 Splash Screen Boa Constructor

4. Tampil 3 window:

- a. Zope Editor
- b. Inspector
- c. Editor

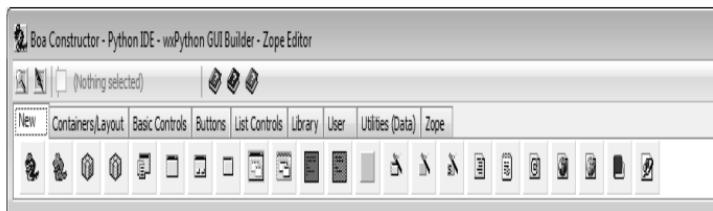


Gambar 2.4 Tampilan 3 Window Utama

Bagian paling atas adalah **Zope Editor**, bagian kanan adalah **Editor**, sedangkan bagian kiri adalah **Inspector**

2.4 Penjelasan Tiga Window

2.4.1 Zope Editor



Gambar 2.5 Window Zope Editor

Zope Editor berisi pilihan Object yang akan diletakkan pada frame (sebutan untuk form).

Zope Editor memiliki 2 icon (di kiri atas) untuk mengaktifkan Editor dan Inspector:

Icon	Fungsi
	Mengaktifkan window Inspector
	Mengaktifkan window Editor

Zope Editor juga memiliki tab-tab yang berisi Object pada frame. Tab yang dimiliki ada 9 :

1. New
2. Containers/ Layout
3. Basic Controls

4. Buttons
5. List Controls
6. Library
7. User
8. Utilities (Data)
9. Zope

Object-object yang ada tidak akan dijabarkan secara keseluruhan, namun hanya object yang akan digunakan pada buku ini saja.

Containers/Layout



Gambar 2.6 Containers/Layout

Object yang digunakan pada tab ini hanyalah **wx.Panel** yang terletak di paling kiri. Object ini adalah object yang seharusnya pertama kali diletakkan pada Frame sebelum object-object yang lain. Ia bertindak sebagai alas/ landasan bagi object yang lain.

Icon `wx.Panel` ditunjukkan gambar 2.7:



Gambar 2.7 Icon wx.Panel

Catatan Penting:

Jika suatu Frame tidak memiliki wx.Panel, maka perpindahan antar TextCtrl (textbox) tidak bisa menggunakan tombol TAB atau ENTER.

Basic Controls



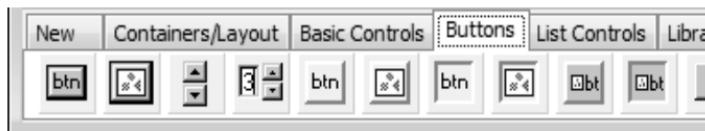
Gambar 2.8 Basic Controls

Object	Icon	Fungsi
wx.StaticText		Sebagai label
wx.TextCtrl		Semacam textbox (kolom isian dalam formulir)
wx.ComboBox		Combobox (Isian pilihan)

wx.CheckBox		Pilihan yang bisa lebih dari satu, contoh: Hoby
wx.RadioButton		Hanya bisa satu pilihan. Contoh : Jenis kelamin
wx.DatePickerCtrl		Format inputan tanggal (Date). Letaknya di nomor dua dari kanan.
wx.StaticBitmap		Menampilkan gambar statis. Letaknya no-9 dari kiri

Tabel 2.1 Object Basic Controls

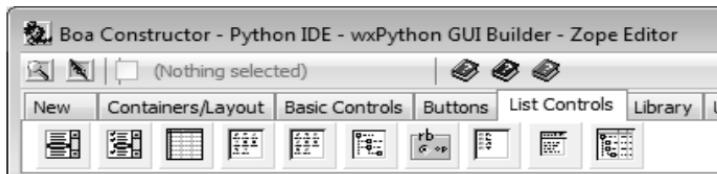
Buttons



Gambar 2.9 Tab Buttons

Pada tab **Buttons** ini hanya 1 object yang paling sering digunakan, yaitu **wx.Button**. Fungsinya sebagai tombol dalam Frame. Letaknya di ujung kiri.

List Controls



Gambar 2.10 Tab List Controls

Object yang sering digunakan pada tab **List Controls** ini adalah **wx.ListBox** dan **wx.ListCtrl**



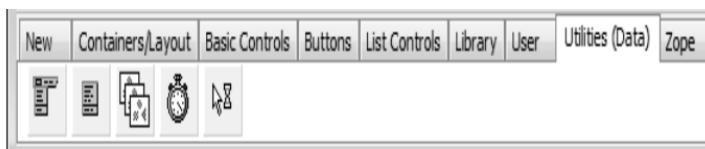
Gambar 2.11 wx.ListBox



Gambar 2.12 Icon wx.ListCtrl

wx.ListCtrl berfungsi semacam grid, yaitu menampilkan data dalam beberapa kolom dan baris.

Utilities(Data)



Gambar 2.13 Object pada Tab Utilities(Data)

Object yang sering digunakan pada Tab ini adalah **wx.MenuBar** dan **wx.Menu**. Posisi paling kiri adalah **wx.MenuBar** sedangkan sebelah kanannya adalah **wx.Menu**. Kedua object ini dibutuhkan pada saat pembuatan menu (InsyaAllah akan dibahas pada Bab IX).

2.4.2 Editor

Editor adalah window pengontrol utama Boa Constructor. Beberapa fungsinya adalah mengatur :

1. Pembuatan file baru (wx.App atau wx.Frame) dan penyimpanannya
2. Pengorganisasian file pada project Python yang kita buat.
3. Pengaktifan mode desainer pada Frame

Pada saat dijalankan pertama kali, Editor memiliki dua tab:

1. Shell.

Berupa tampilan semacam IDLE Python Interpreter yang bisa digunakan untuk menguji baris kode Python.



Gambar 2.14 Editor-Shell Boa Constructor

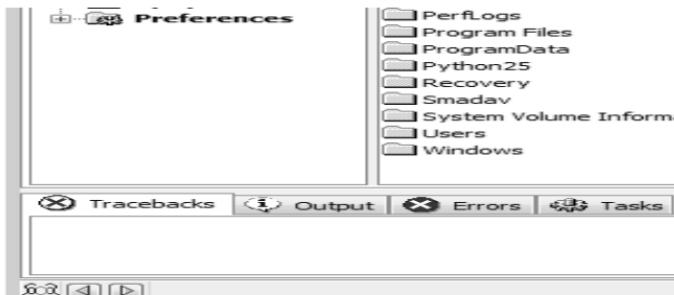
2. Explorer.

Tampilan pengorganisasian folder dan file.



Gambar 2.15 Editor - Explorer Boa Constructor

Pada bagian bawah Editor terdapat jendela untuk pengecekan pesan Error (*Traceback*) yang muncul.



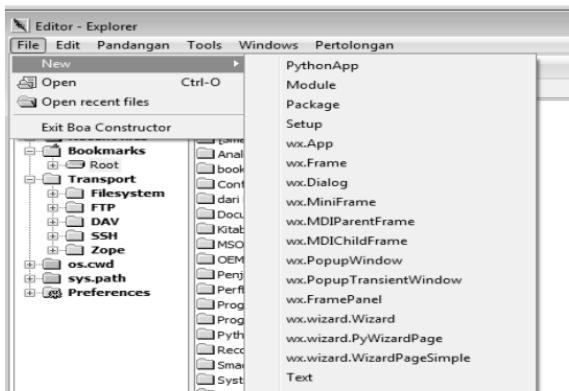
Gambar 2.16 Window Tracebacks

Menu – menu pada Editor

1. File

Pada menu File ini, pilihan yang sering digunakan adalah :

- a. New, untuk membuat file baru.



Gambar 2.17 Membuat File Baru dengan Boa

Untuk membuat aplikasi pertama kali, pilihlah **wx.App**, akan dihasilkan 2 tab baru: **wx.App** dan **wx.Frame**. Fungsi dari wx.App adalah sebagai file pengendali aplikasi, sedangkan wx.Frame sebagai form.

```
#Boa:Frame:Frame1
import wx

def create(parent):
    return Frame1(parent)

[wxID_FRAME1] = [wx.NewId() for i in range(1)]
class Frame1(wx.Frame):
    def __init__(self, parent):
        wx.Frame.__init__(self, parent)
```

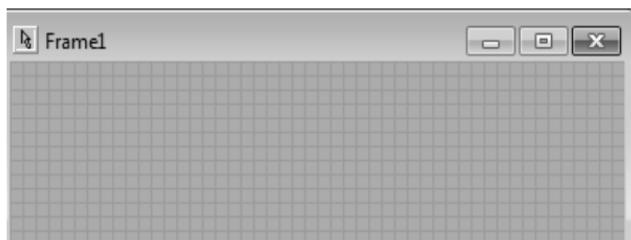
Gambar 2.18 Window Source pada Frame

Selanjutnya, klik icon Frame Designer yang terletak di bawah tulisan **Pertolongan**. Gambar icon Frame Designer ditunjukkan pada icon 2.17.



Gambar 2.19 Icon Frame Designer

Jika icon Frame Designer di-klik, maka akan tampil tampilan Frame :



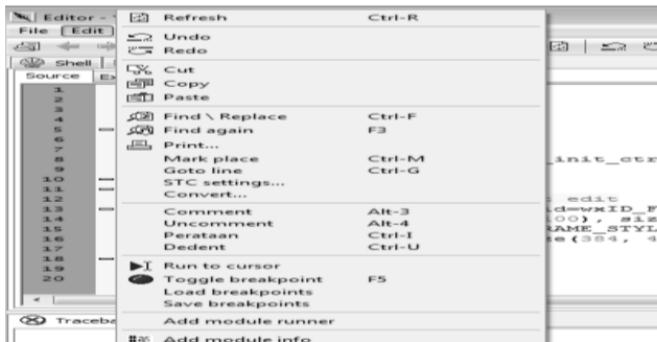
Gambar 2.20 Tampilan Frame Awal

Jika wx.Frame sudah ada, pilihan Frame Designer bisa dipilih juga pada menu **File**. Untuk mengaktifkan Frame Designer bisa juga dengan mengetikkan F12.

Selanjutnya, pengaturan Frame ini bisa dilakukan dengan meletakkan object yang berasal dari window Zope Editor dan Inspector.

- b. Open, untuk membuka file yang sudah ada.
- c. Exit, untuk keluar dari Boa Constructor

2. Edit



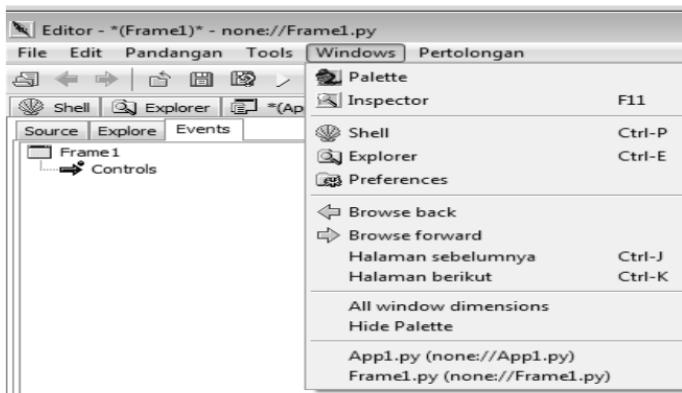
Gambar 2.21 Menu Edit Boa Constructor

Pilihan menu Edit yang paling sering digunakan adalah **Refresh**, bisa juga dengan menekan **Ctrl-R**. Ini dipilih jika kita tidak bisa mengetikkan baris kode baru atau mengedit kode Python pada Boa Constructor.

Pilihan menu Edit lain, secara umum mirip dengan menu Edit pada aplikasi pengolah kata semisal MS.Word dan lainnya.

3. Windows

Menu ini dipilih untuk menampilkan tampilan jendela/ window yang diinginkan



Gambar 2.22 Menu Windows Boa Constructor

Pilihan Palette untuk mengaktifkan toolbar/ window Zope (object-object yang akan diletakkan pada Frame). Pilihan Inspector untuk mengaktifkan jendela Inspector (semacam properties).

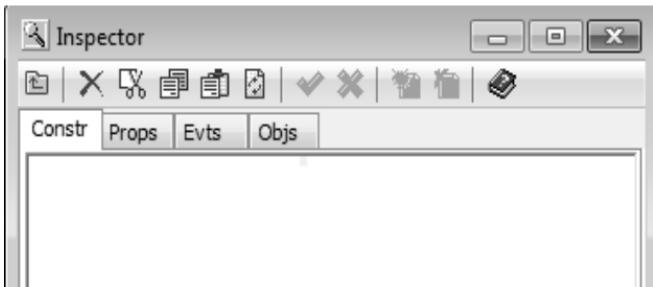
Toolbar pada Editor

Pada window Editor terdapat toolbar berisi icon-icon *shortcut* untuk melakukan perintah tertentu



Gambar 2.23 Toolbar Window Editor

2.4.3 Inspector



Gambar 2.24 Inspector

Pada Inspector terdapat 4 tab:

1. Constr
2. Props
3. Evts
4. Obj

CATATAN

Pada Inspector dan Editor terdapat icon **Post** (tanda centang berwarna biru muda). Icon ini harus di-klik untuk memastikan agar kode Python sudah *generate*. Meski file sudah di-save, jika icon Post tidak di-klik, maka tidak akan ada perubahan kode Python yang dihasilkan

BAB III

PENGENALAN 3 OBJECT UTAMA PADA FRAME

3.1 Pendahuluan

Tiga object utama pada Frame adalah : wx.StaticText, wx.TextCtrl, dan wx.Button. Ketiga object ini hampir pasti selalu ada pada Frame. Namun, sebelum object yang lain, wx.Panel harus diletakkan sebagai landasannya.

Anda akan belajar cara meletakkan object pada form, mengatur properties pada Inspector, membuat Event, serta mengetikkan kode Python. Sebelumnya akan dijelaskan juga tentang pembuatan project aplikasi dan cara menyimpannya.

3.2 Memulai Project

Project aplikasi harus memiliki wx.App. Jika kita membuat sebuah wx.App secara otomatis akan dibuatkan satu wx.Frame. Kesalahan yang sering dilakukan adalah jika pertama kali yang dibuat adalah wx.Frame tanpa menyertakan wx.App. Kedua wx tersebut nanti akan menghasilkan 2 file yang berbeda dengan ekstensi yang sama: .py.

Dari wx.App yang sudah dibuat, nanti bisa ditambahkan wx.Frame lain yang diperlukan.

Wx.App dan wx.Frame yang berkaitan harus diletakkan pada folder yang sama.

Memulai Project Baru: Membaca dan Menampilkan Nama

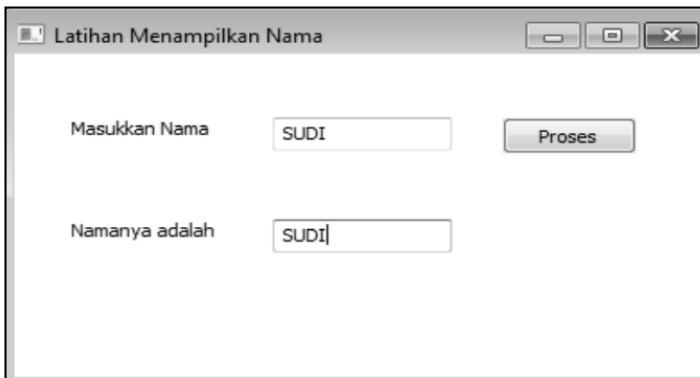
Project yang kita buat sangat sederhana. Bagaimana mengetikkan suatu nama dalam wx.TextCtrl dan menampilkannya pada wx.TextCtrl yang lain setelah ditekan tombol ‘Proses’.

Skenario yang Diinginkan

Alur aplikasi sederhana yang akan kita rancang adalah:

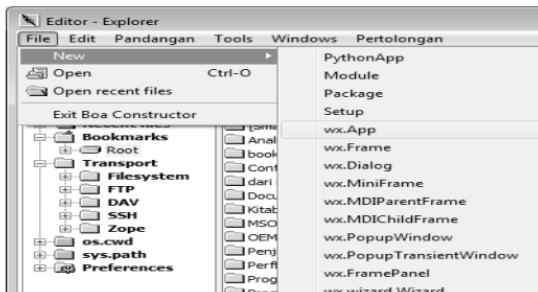
1. User memasukkan nama pada isian textctrl di samping keterangan ‘Masukkan Nama’.
2. Selanjutnya, ditekan tombol Proses.
3. Hasilnya: nama yang diinputkan di textctrl atas akan ditampilkan pula di textctrl di bawahnya (di samping keterangan ‘Namanya adalah’).

Contoh : Jika dimasukkan nama SUDI di textctrl atas, maka setelah ditekan tombol PROSES, akan ditampilkan nama SUDI pula di textctrl yang di bawahnya. TextCtrl bawah menampilkan tulisan yang sama dengan tulisan pada textctrl di atasnya



Gambar 3.1 Aplikasi Membaca dan Menampilkan Nama
Langkah-langkah untuk membuat nya adalah:

1. Aktifkan window Editor, kemudian pilih menu File > New > wx.App



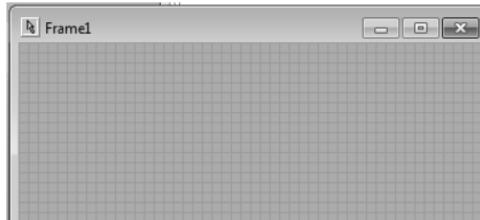
Gambar 3.2 File > New > wx.App

akan dihasilkan tab App1 dan Frame1

```
#Boa::Frame::Frame1
import wx
def create(parent):
    return Frame1(parent)
[wxID_FRAME1] = [wx.NewId() for _in
class Frame1(wx.Frame):
    def __init__(self, parent):
```

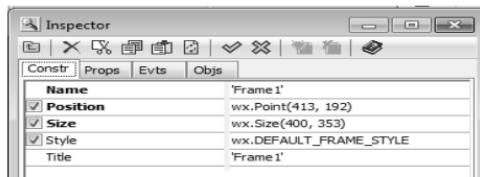
Gambar 3.3 Tercipta App1 dan Frame1

2. Pilih Frame Designer dengan cara menekan tombol F12, sehingga muncul tampilan Frame kosong yang siap didesain (sebelah kiri)



Gambar 3.4 Tampilan Awal Frame1

3. Aktifkan window Inspector dengan menekan tombol F11. Akan muncul di sebelah kiri



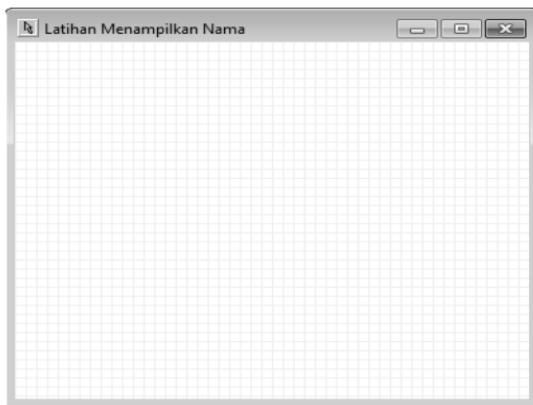
Gambar 3.5 Inspector

4. Settinglah nilai Inspector pada Frame1 sesuai tabel 3.1.

TAB	Properties	Nilai
Constr	Title	Latihan Menampilkan Nama
Props	BackgroundColour	255,255,255,255 (putih)

Tabel 3.1 Setting Inspector Frame1

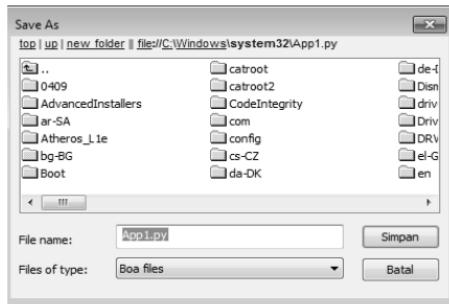
Hasilnya adalah seperti berikut:



Gambar 3.6 Hasil Setting Frame1

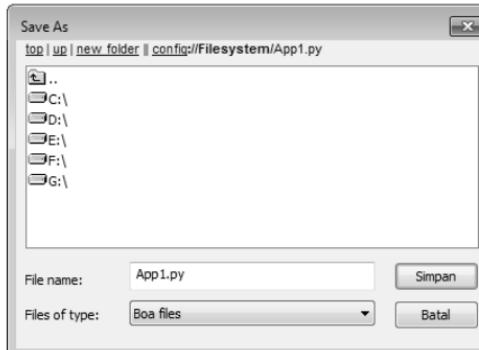
5. Simpan terlebih dahulu App1 dan Frame1.

Aktifkan tab App1 di Editor, kemudian tekan Ctrl – S



Gambar 3.7 Save As App1

Aturlah tempat tujuan direktori dengan mengklik tulisan **|| file** di bagian atas:



Gambar 3.8 Penempatan Direktori

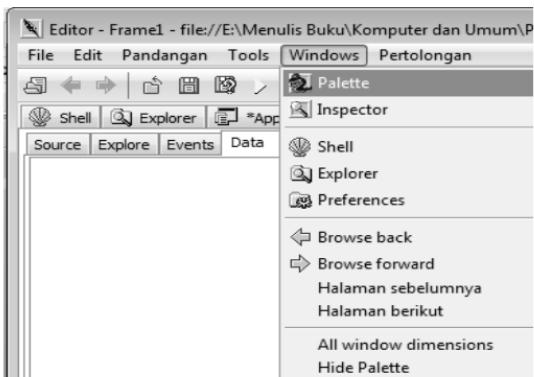
Selanjutnya, arahkan penyimpanannya ke direktori dan folder tertentu sesuai keinginan anda. Anda bisa menambahkan folder baru dengan cara klik kanan dan pilih New Folder.

Aktifkan tab Frame1 pada Editor, lakukan penyimpanan dengan cara yang sama dengan pada

saat menyimpan App1. Anda boleh memberikan nama terserah sebagai pengganti App1 dan Frame1.

6. Letakkan wx.Panel pada Frame.

Pada Editor, pilih menu Windows > Palette



Gambar 3.9 Windows > Palette

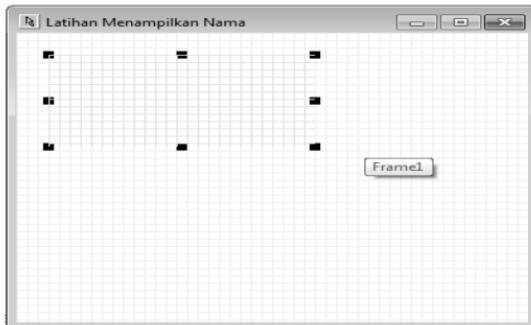
Ini akan mengaktifkan Zope Editor/ Palette (tempat pilihan object yang akan diletakkan pada Frame).

Pilih tab Containers/Layout



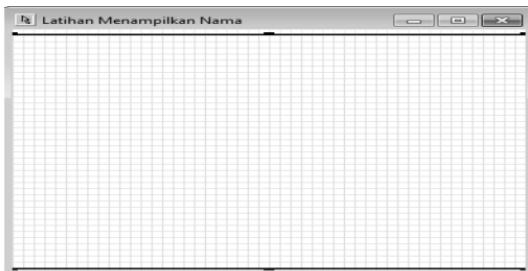
Gambar 3.10 Tab Containers/Layout

Pilih icon wx.Panel yang terletak di ujung kiri (bujursangkar berwarna abu-abu), kemudian letakkan pada Frame1.



Gambar 3.11 Meletakkan wx.Panel pada Frame1

Aturlah luas wx.Panel agar mencakup seluruh area dalam Frame1



Gambar 3.12 Mengatur Lebar wx.Panel

7. Pindah ke tab Basic Controls di Palette/ Zope Editor.

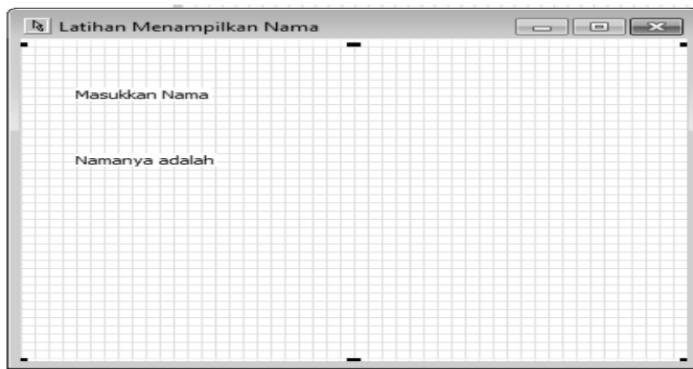


Gambar 3.13 Tab Basic Controls

Pilihlah wx.StaticText (ujung kiri) icon bertuliskan ABC, kemudian letakkan ke dalam Frame1. Letakkan 2 wx.StaticText ke Frame1. Atur Inspectornya sebagai berikut:

Object	Tab	Setting	Nilai
wx.Statictext	Constr	Label	'Masukkan Nama'
wx.Statictext	Constr	Label	'Namanya adalah'

Hasil perubahan pada Frame1 seperti ditunjukkan Gambar 3.14



Gambar 3.14 Hasil Frame1 dengan 2 Label

- Letakkan 2 wx.TextCtrl ke dalam Frame 1 bersejajar dengan masing-masing wx.StaticText.

Icon wx.TextCtrl ditunjukkan pada Gambar 3.15.

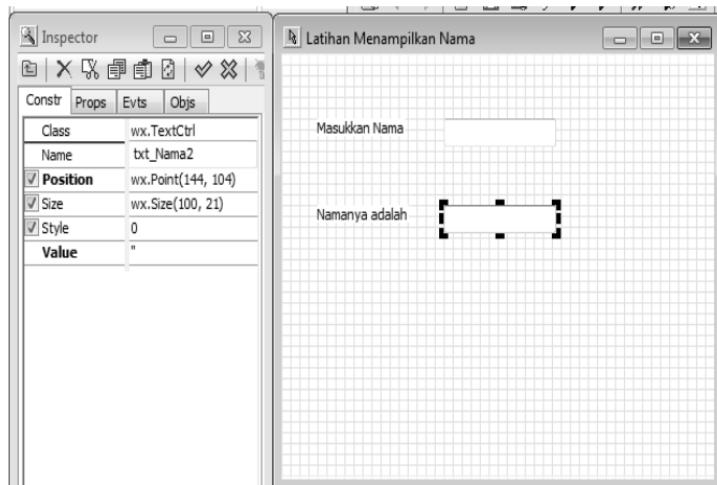


Gambar 3.15 Icon wx.TextCtrl

Letaknya nomor dua dari kiri (setelah wx.StaticText)

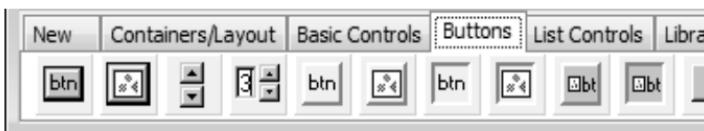
Letakkan ke dalam Frame1, dan atur Inspectornya sebagai berikut:

Object	Tab	Setting	Nilai
wx.TextCtrl	Constr	Name	txt_Nama1
		Value	(dikosongkan)
wx.TextCtrl	Constr	Name	txt_Nama2
		Value	(dikosongkan)



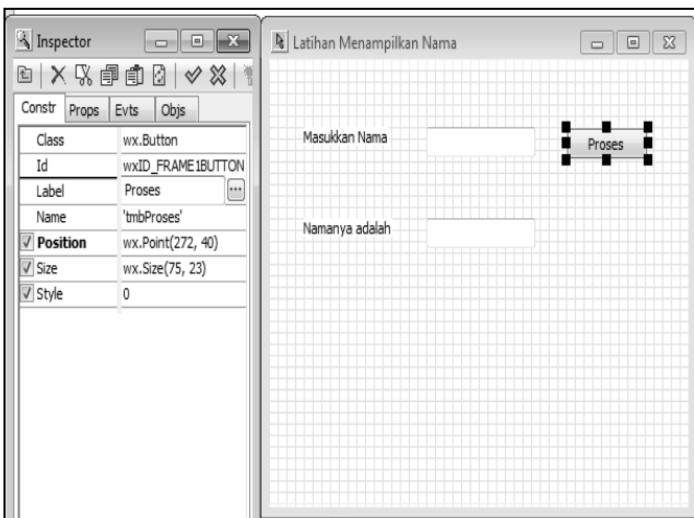
Gambar 3.16 Meletakkan 2 wx.TextCtrl

9. Pada Palette/ Zope Editor, pindah ke tab Buttons



Gambar 3.17 Tab Button

Pilih wx.Button object yang bertuliskan **btn** posisinya di ujung kiri. Letakkan pada Frame1 di samping wx.TextCtrl **txt_Nama1**.



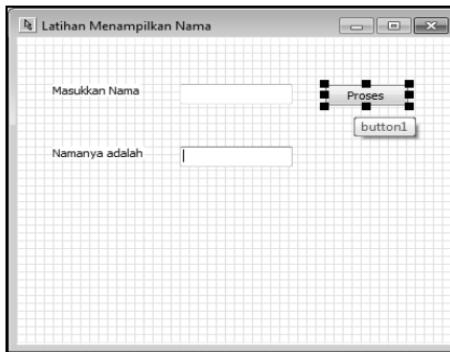
Gambar 3.18 Meletakkan wx.Button ‘Proses’

Setting Inspectornya adalah sebagai berikut:

Nama Object	Tab	Setting	Nilai
wx.Button	Constr	Label	Proses
		Name	'tmbProses'

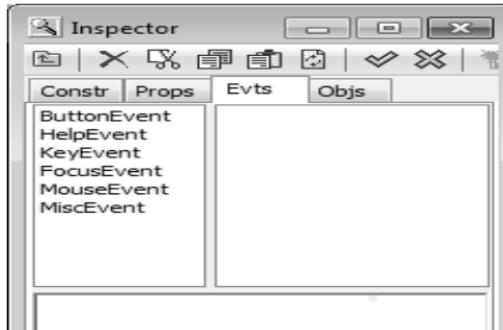
10. Buatlah Event pada wx.Button yang sudah ada.

- Pastikan object wx.Button yang terpilih (aktif).



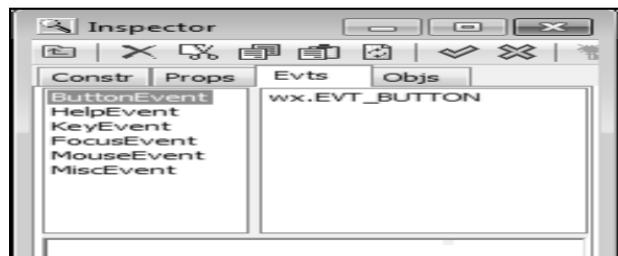
Gambar 3.19 wx.Button Terpilih (Aktif)

- Pada Inspector, pilih tab Evts



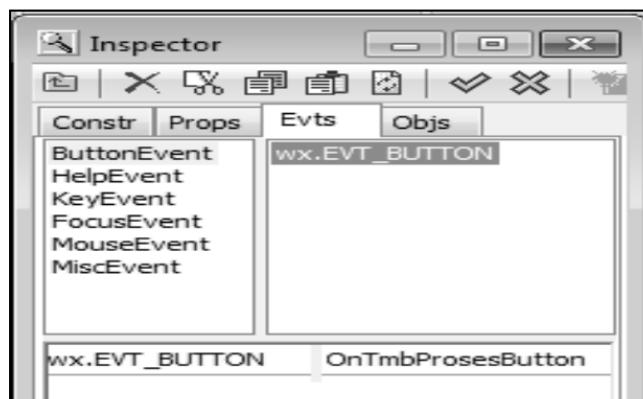
Gambar 3.20 Tab Evts pada wx.Button

Klik pada ButtonEvent sehingga muncul tulisan wx.EVT_BUTTON di sebelah kanannya



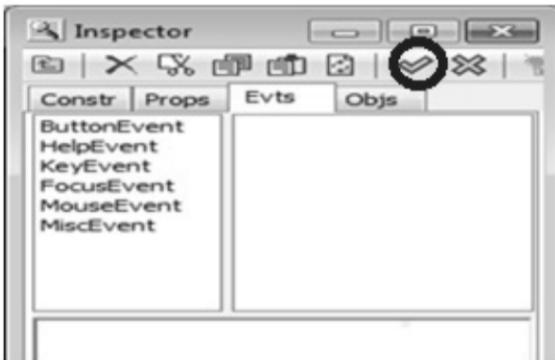
Gambar 3.21 wx.EVT_BUTTON

Klik 2 kali pada tulisan wx.EVT_BUTTON sehingga muncul tulisan **OnTmbProsesButton** di bawahnya



Gambar 3.22 Evts – OnTmbProsesButton

11. Klik tanda Post (centang warna biru) di Inspector maupun pada Editor, supaya terjadi perubahan kode Python.



Gambar 3.23 Tanda Post pada Inspector



Gambar 3.24 Tanda Post pada Toolbar Editor

12. Pada window Editor, klik tab Source untuk melihat Source Code secara keseluruhan

A screenshot of the wxPython Editor window showing the 'Source' tab selected. The title bar says 'Editor - Frame1'. The main area displays Python code for a Frame1 class:

```
#Boa:Frame:Frame1
import wx
def create(parent):
    return Frame1(parent)
[wxID_FRAME1, wxID_FRAME1PANEL1, wxID_FRAME1ST,
wxID_FRAME1STATICTEXT2, wxID_FRAME1TMBPROSES,
wxID_FRAME1TXT_NAMA2,
] = [wx.NewId() for _init_ctrls in range(7)]
class Frame1(wx.Frame):
    def __init__(self, parent):
        # generated method, don't edit
        wx.Frame.__init__(self, id=wxID_FRAME1,
pos=wx.Point(561, 157), size=wx.
```

Gambar 3.25 Tab Source Frame1 Boa Constructor

Arahkan ke Source Code (Kode Python) paling bawah. Terdapat baris kode sebagai berikut:

```
def OnTmbProsesButton(self, event):
    event.Skip()
```

Ini adalah baris kode yang dihasilkan oleh proses terakhir: pemberian Event pada tombol (wx.Button) **tmbProses**. Boa Constructor membuatkan kode otomatis : **event.Skip()**. Kode tersebut selanjutnya kita hapus dan kita ganti dengan baris-baris kode yang sesuai.

Kode yang berada di bawah **def OnTmbProsesButton(self, event):** adalah kode yang akan dijalankan pada saat tombol **Proses** ditekan oleh user (pengguna program kita).

Kode untuk tombol Proses adalah :

```
def OnTmbProsesButton(self, event):
    nama = self.txt_Nama1.GetValue()
    self.txt_Nama2.SetValue(nama)
```

Penjelasan tentang Kode :

Baris pertama :

```
nama = self.txt_Nama1.GetValue()
```

textctrl **txt_Nama1** membaca inputan dari user kemudian menyimpannya ke dalam variabel **nama**. Fungsi tersebut dilakukan oleh **GetValue**.

Baris kedua:

```
self.txt_Nama2.SetValue(nama)
```

TextCtrl **txt_Nama2** menampilkan hasil dari variabel nama.

Sebenarnya, 2 baris tersebut bisa diringkas dalam 1 baris sebagai berikut:

```
self.txt_Nama2.SetValue(self.txt_Nama1.GetValue())
```

Catatan Penting :

1. Dua method yang paling sering digunakan pada wx.TextCtrl:
 - a. GetValue() : membaca data yang diinputkan
 - b. SetValue() : menampilkan data tertentu
2. Pemanggilan sebuah object selalu dimulai dengan **self.** Contoh :
self.txt_Nama1.GetValue()
3. Tipe data yang bisa dibaca atau ditampilkan dalam wx.TextCtrl adalah String.

Cara Menjalankan Program

Tekan tombol F9 (Run Application) atau memilih menu File> Run Application pada window Editor

KODE PROGRAM SECARA KESELURUHAN

Secara umum, sebuah aplikasi minimal terdiri dari 2 file (jika nama file tidak diubah), yaitu:

1. App1.py
2. Frame1.py

KODE App1.py

```
#!/usr/bin/env python

#Boa:App:BoaApp


import wx


import Frame1


modules ={'Frame1': [1, 'Main frame of Application',
u'Frame1.py']}

class BoaApp(wx.App):

    def OnInit(self):

        self.main = Frame1.create(None)

        self.main.Show()

        self.SetTopWindow(self.main)

        return True
```

```
def main():

    application = BoaApp(0)

    application.MainLoop()

if __name__ == '__main__':

    main()
```

KODE Frame1.py

```
#Boa:Frame:Frame1

import wx

def create(parent):

    return Frame1(parent)

[wxID_FRAME1, wxID_FRAME1PANEL1,
wxID_FRAME1STATICTEXT1,

wxID_FRAME1STATICTEXT2, wxID_FRAME1TMBPROSES,
wxID_FRAME1TXT_NAMA1,

wxID_FRAME1TXT_NAMA2,
] = [wx.NewId() for _init_ctrls in range(7)]
```

```
class Frame1(wx.Frame):
    def __init__(self, prnt):
        # generated method, don't edit
        wx.Frame.__init__(self, id=wxID_FRAME1, name="",
parent=prnt,
pos=wx.Point(561, 157), size=wx.Size(400, 241),
style=wx.DEFAULT_FRAME_STYLE, title='Latihan
Menampilkan Nama')

        self.SetClientSize(wx.Size(384, 203))
        self.SetBackgroundColour(wx.Colour(255, 255, 255))

        self.panel1 = wx.Panel(id=wxID_FRAME1PANEL1,
name='panel1', parent=self,
pos=wx.Point(0, 0), size=wx.Size(384, 203),
style=wx.TAB_TRAVERSAL)

        self.staticText1 =
wx.StaticText(id=wxID_FRAME1STATICTEXT1,
label='Masukkan Nama', name='staticText1',
parent=self.panel1,
pos=wx.Point(32, 40), size=wx.Size(78, 13), style=0)

        self.staticText2 =
wx.StaticText(id=wxID_FRAME1STATICTEXT2,
```

```
        label='Namanya adalah', name='staticText2',
parent=self.panel1,
pos=wx.Point(32, 104), size=wx.Size(81, 13), style=0)

self.txt_Nama1 =
wx.TextCtrl(id=wxID_FRAME1TXT_NAMA1,
name='txt_Nama1',
parent=self.panel1, pos=wx.Point(144, 40),
size=wx.Size(100, 21),
style=0, value="")

self.txt_Nama2 =
wx.TextCtrl(id=wxID_FRAME1TXT_NAMA2,
name='txt_Nama2',
parent=self.panel1, pos=wx.Point(144, 104),
size=wx.Size(100, 21),
style=0, value="")

self.tmbProses =
wx.Button(id=wxID_FRAME1TMBPROSES, label='Proses',
name='tmbProses', parent=self.panel1,
pos=wx.Point(272, 40),
size=wx.Size(75, 23), style=0)

self.tmbProses.Bind(wx.EVT_BUTTON,
self.OnTmbProsesButton,
id=wxID_FRAME1TMBPROSES)
```

```
def __init__(self, parent):
    self._init_ctrls(parent)

def OnTmbProsesButton(self, event):
    nama = self.txt_Nama1.GetValue()
    self.txt_Nama2.SetValue(nama)
```

BAB IV

KONVERSI TIPE DATA

(STRING DAN INTEGER)

4.1 Catatan Awal

Konversi tipe data :

1. String ke Integer → int()
2. Integer ke String → str()

4.2 Project Menghitung Luas Persegi Panjang

Berikut kita akan belajar membuat project sederhana untuk menunjukkan proses konversi tipe data dari string ke integer dan sebaliknya. Langkah-langkah secara detail tidak diulang. Anda bisa melihat lagi pada bab 3 pada saat membuat project **Latihan Menampilkan Nama**.

Skenario yang Diharapkan

User diminta memasukkan ukuran panjang dan lebar persegi panjang. Jika kemudian ditekan tombol **Hitung Luas**, akan ditampilkan luas persegi panjang tersebut dengan perhitungan :

Luas = panjang x lebar.

Contoh : jika diinputkan panjang = 5 dan lebar =4, maka luasnya adalah 20.



Gambar 4.1 Tampilan Aplikasi Menghitung Luas Persegi

Setting Inspector pada Frame :

OBJECT	TAB	SETTING	NILAI
FRAME1	Constr	Title	Menghitung Luas Persegi
	Props	BackgroundColour	(255,255,255,255)
wx.Panel		diletakkan pertama kali sbg alas object lain	
wx.StaticText	Constr	Label	Panjang
wx.StaticText	Constr	Label	Lebar
wx.StaticText	Constr	Label	Luas
wx.TextCtrl	Constr	Name	txt_panjang
		Value	(dikosongkan)
		Name	txt_lebar
		Value	(dikosongkan)
		Name	txt_luas
		Value	(dikosongkan)
wx.Button	Constr	Label	Hitung Luas
		Name	tmbHitung

	Evts	Button Event	wx.EVT_BUTTON
--	------	--------------	---------------

Tabel 4.1 Setting Inspector Aplikasi Hitung Luas Persegi

Kode pada Tombol Hitung

Kode yang benar untuk tombol Hitung adalah:

```
def OnTmbHitungButton(self, event):
    p = int(self.txt_panjang.GetValue())
    l = int(self.txt_lebar.GetValue())
    luas = p * l
    self.txt_luas.SetValue(str(luas))
```

Penjelasan :

Rumus menghitung luas persegi panjang adalah : panjang x lebar. Variabel p untuk panjang dan variabel l untuk lebar. Dua wx.TextCtrl untuk menampung input adalah txt_panjang dan txt_lebar. Membaca input data dengan method: GetValue(). Karena sifat wx.TextCtrl yang hanya bisa menampung data bertipe String, maka masing-masing variable sebelum diisi, harus dikonversikan dulu ke integer.

p = int(self.txt_panjang.GetValue())

artinya : membaca input data di `txt_panjang`, kemudian dikonversikan ke integer, dan disimpan ke variable `p`.

Demikian juga cara yang sama berlaku untuk mengisi variabel `l`.

Setelah itu, hitung rumusnya dengan syntax yang dikenal oleh python, yaitu:

```
luas = p * l
```

Berikutnya, nilai variable `luas` akan ditampilkan ke dalam `wx.TextCtrl txt_luas`. Variabel `luas` bertipe integer, karena ia hasil perhitungan integer dikalikan integer. Sedangkan tipe data yang bisa ditampilkan oleh `wx.TextCtrl txt_luas` hanyalah String. Karena itu, perlu dilakukan konversi data dari integer ke String dengan perintah `str()`. Kode berikut ini:

```
self.txt_luas.SetValue(str(luas))
```

artinya adalah: menampilkan hasil di `txt_luas` dengan `SetValue()` dengan menkonversikan tipe data integer pada variabel `luas` ke String.

4.3 Project Menghitung Usia

Berikut adalah contoh aplikasi sederhana untuk menghitung usia user berdasarkan tahun lahir.

Skenario yang Diharapkan

User mengetikkan tahun lahir, selanjutnya ditampilkan usianya. Perhitungannya adalah :

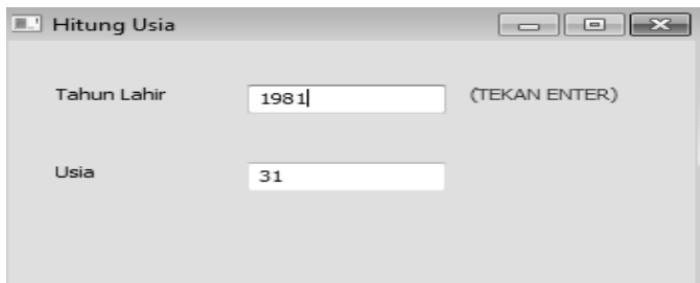
Usia = Tahun_sekarang – tahun_lahir.

(Asumsi: tahun sekarang adalah 2012)

Contoh: User memasukkan tahun lahir 1979, maka usianya saat ini adalah : $2012 - 1979 = 33$

Kejadian Setelah Tombol ENTER Ditekan

Jika pada contoh sebelumnya semua proses terjadi setelah tombol (wx.Button) ditekan, maka pada aplikasi berikut kita akan berkenalan dengan EVENT WX_EVT_TEXT_ENTER. Proses baru akan terjadi setelah ditekan tombol ENTER. Hal itu pada saat kursor masih berada di wx.TextCtrl.



Gambar 4.2 Aplikasi Hitung Usia

Setting Inspector pada Frame :

OBJECT	TAB	SETTING	NILAI
FRAME1	Constr	Title	Hitung Usia
	Props	Background Colour	Default
wx.Panel	diletakkan pertama kali sbg alas object lain		
	Props	BackGround Colour	(223,223,223,255)

wx.StaticText	Props	Label	Tahun Lahir
wx.StaticText	Props	Label	(TEKAN ENTER)
		Foreground Colour	(255,0,0,255)
wx.StaticText	Props	Label	Usia
Wx.TextCtrl	Constr	Name	th_lahir
		Value	(dikosongkan)
		Style	wx.TE_PROCESS_ENTER
	Evts	TextCtrlEvent	Wx.EVT_TEXT_ENTER
wx.TextCtrl	Constr	Name	txt_usia
		Value	(dikosongkan)

Tabel 4.2 Setting Inspector Hitung Usia

Kode Inti :

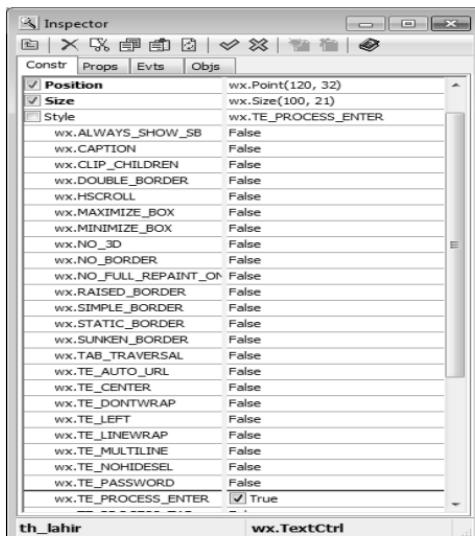
```
def OnTh_lahirTextEnter(self, event):
    Usia = 2012 - int(self.th_lahir.GetValue())
    self.txt_usia.SetValue(str(Usia))
```

Cara Membuat Event pada TextCtrl (Tekan Enter)

Agar baris – baris kode bisa dijalankan jika tombol ENTER ditekan, maka ada 2 hal utama yang harus dilakukan:

1. Buatlah Event EVT_ WX_EVT_TEXT_ENTER pada wx.TextCtrl (langkah-langkahnya akan dijelaskan nanti)

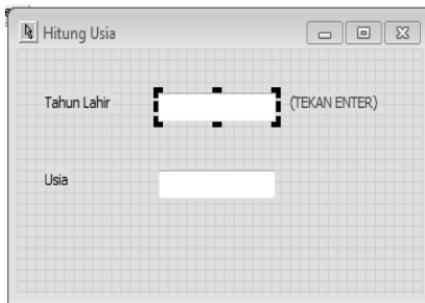
2. Settinglah pada Inspector nilai Style (tab Constr) wx.TextCtrl dibuat menjadi : wx.TE_PROCESS_ENTER



Gambar 4.3 Style TextCtrl: wx.TE_PROCESS_ENTER

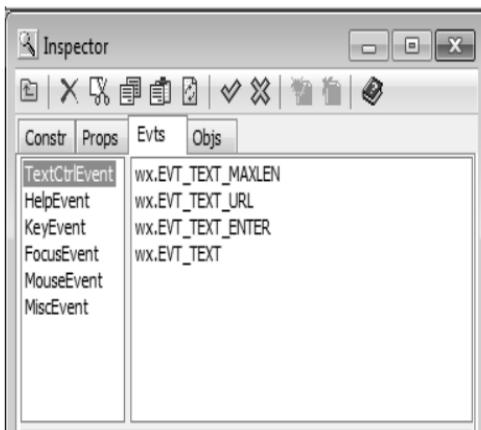
LANGKAH-LANGKAH MEMBUAT EVENT PADA TEXTCTRL

1. Pastikan wx.TextCtrl yang akan dibuatkan event sedang terpilih (aktif).



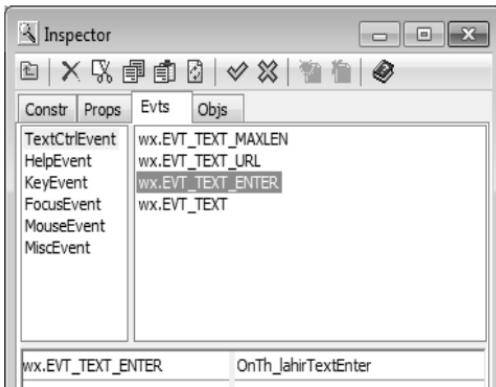
Gambar 4.4 wx.TextCtrl Terpilih (Aktif)

2. Pilih tab Evts pada Inspector, kemudian klik pada bagian **TextCtrlEvent**



Gambar 4.5 Evts-TextCtrlEvent

3. Klik dua kali pada wx.EVT_TEXT_ENTER



Gambar 4.6 Klik 2x pada wx.EVT_TEXT_ENTER

Karena wx.TextCtrl yang dibuatkan Event bernama th_lahir, maka akan dihasilkan kode :

```
def OnTh_lahirTextEnter(self, event):
    event.Skip()
```

baris event.Skip() itu dihapus dan diganti dengan kode seperti di atas.

BAB V

LOGIKA PENCABANGAN, MESSAGE DIALOG, RADIOBUTTON, DAN CHECKBOX

5.1 Pencabangan dengan if

Setiap bahasa pemrograman memiliki syntax untuk pencabangan atau seleksi kondisi. Pencabangan terbagi menjadi 2 :

1. Pencabangan dengan 2 kemungkinan
2. Pencabangan dengan lebih dari 2 kemungkinan

Kita akan mengembangkan aplikasi penghitungan usia pada bab sebelumnya untuk menunjukkan cara kerja pencabangan.

5.1.1 Pencabangan 2 Kemungkinan

Aplikasi penghitungan usia kita kembangkan agar bisa mendeteksi kategori : DEWASA atau BELUM DEWASA. Aturannya adalah:

DEWASA : > 17 TAHUN

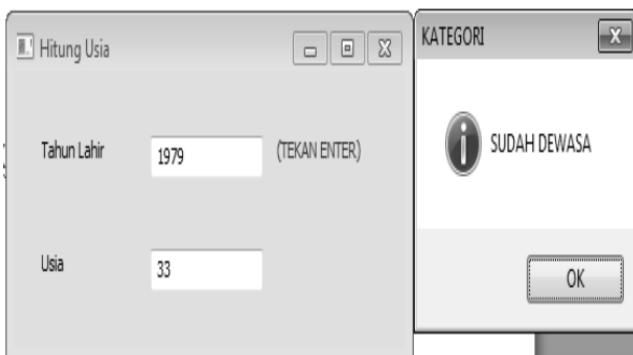
BELUM DEWASA : <= 17 TAHUN

Contoh:

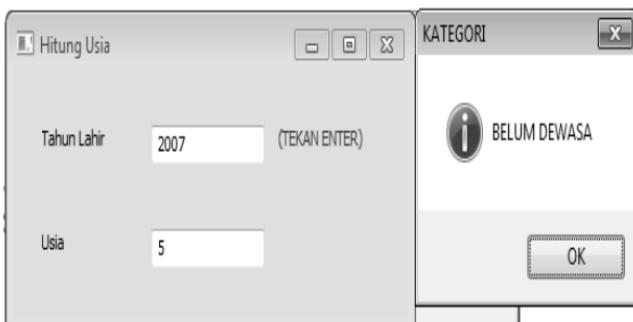
Jika User menginputkan tahun lahir : 1979, maka usianya adalah 33 (tahun) dan kategori : DEWASA.

Aplikasi berikut dikembangkan selain menggunakan logika pencabangan juga mulai diperkenalkan

penggunaan wx.MessageDialog untuk menampilkan pesan.



Gambar 5.1 Kategori DEWASA untuk Usia 33 Tahun



Gambar 5.2 Kategori BELUM DEWASA untuk Usia 5 Tahun

CATATAN :

Jika anda akan memodifikasi kode pada suatu program di dalam Boa Constructor dan ternyata tidak bisa ditambah atau dikurangi (ketika diketikkan tidak muncul karakter

apapun), maka tekanlah tombol Refresh di Toolbar Editor atau tekan Ctrl-R.

Tanda '\' berfungsi untuk memotong baris kode sehingga bisa dilanjutkan di baris berikutnya

Baris comment adalah baris yang tidak dieksekusi oleh komputer. Ia bisa berupa baris penjelasan. Diawali dengan tanda '#'

KODE INTI :

```
def OnTh_lahirTextEnter(self, event):
    Usia = 2012 - int(self.th_lahir.GetValue())
    self.txt_usia.SetValue(str(Usia))
    if Usia > 17 :
        self.pesan = wx.MessageDialog(self,\n            "SUDAH DEWASA","KATEGORI",wx.OK)
        self.pesan.ShowModal()
    else :
        self.pesan = wx.MessageDialog(self,\n            "BELUM DEWASA","KATEGORI",wx.OK)
        self.pesan.ShowModal()
```

KETERANGAN:

Syntax pencabangan 2 kemungkinan pada python adalah:

```
if (kondisi) :  
    <statement 1>  
  
else :  
    <statement 2>
```

<statement 1> yang merupakan bagian dari **if** diketikkan lebih menjorok ke dalam (indentasi), sedangkan tulisan **else** diketikkan kembali sejajar dengan **if**. **<statement2>** juga diketikkan menjorok ke dalam sejajar dengan **<statement1>**. Di akhir pernyataan **if** dan **else** terdapat tanda ‘:’ (titik dua).

Penggunaan tanda ‘\’ adalah untuk memotong penulisan kode yang sebenarnya berada dalam satu baris, namun dilanjutkan pada baris berikutnya.

Menampilkan Pesan dengan MessageDialog

Struktur kode untuk menampilkan pesan terbagi dalam 2 hal utama:

1. Setting pesan
2. Menampilkan pesan

Setting pesan pada kode di atas:

```
self.pesan = wx.MessageDialog(self,\n    "SUDAH DEWASA","KATEGORI",wx.OK)
```

sedangkan perintah untuk menampilkan pesan:

```
self.pesan.ShowModal()
```

Pesan tidak akan ditampilkan jika hanya ada setting pesan saja tanpa kode untuk menampilkan.

5.1.2 Pencabangan Lebih dari 2 Kemungkinan

Kita kembangkan lagi aplikasi tersebut sehingga mencakup logika pencabangan dengan lebih dari 2 kemungkinan. Jika sebelumnya kategori usia hanya dua: SUDAH DEWASA dan BELUM DEWASA, maka berikut ini kategorinya lebih luas:

Usia	Kategori
0 – 5 tahun	BALITA
6 – 12 tahun	ANAK-ANAK
13 – 17 tahun	REMAJA
18 - 50 tahun	DEWASA
Lebih dari 50 tahun	LANSIA

Syntax Pencabangan Lebih dari 2 Kemungkinan

```
if (kondisi 1) :  
    <statement1>  
  
elif (kondisi2) :  
    <statement2>  
  
elif (kondisi3) :
```

```
<statement3>

....  
elif <kondisi ke n-1> :  
    <statement ke n-1>  
else :  
    <statement ke n>
```

KODE INTI :

```
def OnTh_lahirTextEnter(self, event):  
  
    Usia = 2012 - int(self.th_lahir.GetValue())  
  
    self.txt_usia.SetValue(str(Usia))  
  
    if Usia > 50 :  
  
        self.pesan = wx.MessageDialog(self,\n  
        "LANSIA","KATEGORI",wx.OK)  
  
    elif Usia > 17 :  
  
        self.pesan = wx.MessageDialog(self,\n  
        "DEWASA","KATEGORI",wx.OK)  
  
    elif Usia > 12 :  
  
        self.pesan = wx.MessageDialog(self,\n  
        "REMAJA","KATEGORI",wx.OK)  
  
    elif Usia > 5 :
```

```
    self.pesan = wx.MessageDialog(self,\n        "ANAK-ANAK","KATEGORI",wx.OK)\n\n    else :\n\n        self.pesan = wx.MessageDialog(self,\n            "BALITA","KATEGORI",wx.OK)\n\n        self.pesan.ShowModal()
```

KETERANGAN

Setiap statement adalah setting pesan dan akhir kode menampilkan pesan. Kode untuk menampilkan pesan tidak menjorok ke dalam, tapi sejajar dengan **if** atau **else** karena bukan merupakan bagian dari pencabangan.

Kode di atas juga ekivalen dengan kode yang melibatkan operator **and**. Sehingga, kode tersebut bisa dibuat dengan struktur sebagai berikut :

```
if Usia >=0 and Usia<=5 :\n\n    self.pesan = wx.MessageDialog(self,\n        "BALITA","KATEGORI",wx.OK)\n\nelif Usia >=6 and Usia<=12 :\n\n    self.pesan = wx.MessageDialog(self,\n        "ANAK-ANAK","KATEGORI",wx.OK)\n\nelif Usia >=13 and Usia<=17 :\n\n    self.pesan = wx.MessageDialog(self,\n        "BALITA","KATEGORI",wx.OK)
```

```
"REMAJA","KATEGORI",wx.OK)

elif Usia >=18 and Usia<=50 :

    self.pesan = wx.MessageDialog(self,\n
    "DEWASA","KATEGORI",wx.OK)

else :

    self.pesan = wx.MessageDialog(self,\n
    "LANSIA","KATEGORI",wx.OK)

self.pesan.ShowModal()
```

5.2 RADIOPUSHBUTTON

Wx.RadioButton berfungsi untuk menampung pilihan user seperti jenis kelamin dan semisalnya. Berikut ini adalah salah satu contoh aplikasi yang mendemonstrasikan cara kerja wx.RadioButton.

Posisi RadioButton pada Toolbar ditunjukkan dalam Gambar 5.3

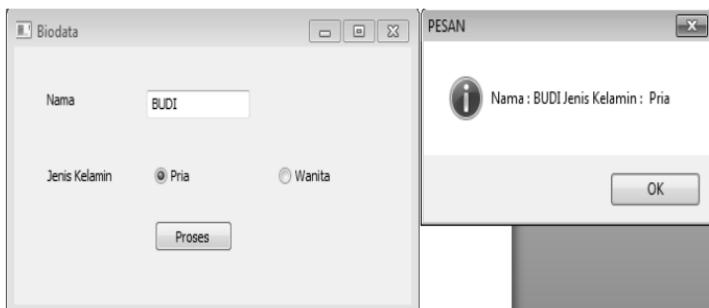


Gambar 5.3 Posisi RadioButton pada Tab Basic Controls

Aplikasi Nama dan Jenis Kelamin

Kita akan buat aplikasi sederhana menampung data inputan biodata user meliputi nama dan jenis

kelamin serta menampilkan hasil inputan tersebut dalam MessageDialog.



Gambar 5.4 Tampilan Biodata dgn RadioButton

SETTING NILAI INSPECTOR

OBJECT	TAB	SETTING	NILAI
FRAME1	Constr	Title	Biodata
	Props	BackgroundColour	default
wx.Panel	diletakkan pertama kali sbg alas object lain		
wx.StaticText	Constr	Label	Nama
wx.StaticText	Constr	Label	Jenis Kelamin
wx.TextCtrl	Constr	Name	txt_nama
		Value	(dikosongkan)
wx.RadioButton	Constr	Label	Pria
		Name	rPria
wx.RadioButton	Constr	Label	Wanita
		Name	rWanita
wx.Button	Constr	Label	Proses
	Constr	Name	tmbProses
	Evts	ButtonEvent	wx.EVT_BUTTON

KODE TOMBOL PROSES

```
def OnTmbProsesButton(self, event):
    nama = self.txt_nama.GetValue()
    if self.rPria.GetValue()==True :
        jk = "Pria"
    else :
        jk ="Wanita"
    a = "Nama : "+nama+" "+Jenis Kelamin : "+" "+jk
    self.pesan = wx.MessageDialog(self,\n        a,"PESAN",wx.OK)
    self.pesan.ShowModal()
```

PENJELASAN

Dalam contoh aplikasi di atas terdapat 2 wx.RadioButton, yaitu **rPria** dan **rWanita**. Variable **jk** digunakan untuk menampung data pilihan user terhadap RadioButton tersebut. Jika **rPria** yang terpilih, maka **jk** berisi nilai “Pria”. Sebaliknya, jika **rPria** tidak terpilih, maka **jk** terisi dengan nilai “Wanita”.

Sebelum ditampilkan ke dalam pesan, dibutuhkan variabel **a**. Variabel ini merupakan String gabungan

penjelasan nama dan jenis kelamin sesuai inputan user pada wx.TextCtrl dan wx.RadioButton.

CATATAN

Method SetValue() pada wx.RadioButton berfungsi untuk mensetting keadaan RadioButton terpilih atau tidak. Contoh: jika kita ingin membuat rWanita yang terpilih, gunakan perintah:

```
self.rWanita.SetValue(True)
```

5.3 CHECKBOX

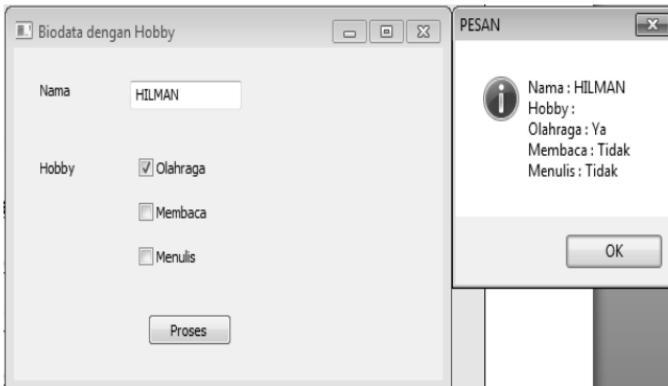
Wx.CheckBox adalah object yang dibutuhkan untuk menampung pilihan yang lebih dari satu jenis ada form. Misalkan pilihan hobby.



Gambar 5.5 Posisi CheckBox pada Tab Basic Controls

APLIKASI BIODATA DENGAN CHECKBOX

Berikut ini adalah contoh aplikasi sederhana biodata nama dan hobby yang menggunakan checkbox.



Gambar 5.6 Biodata dengan CheckBox

Object `wx.CheckBox` yang ada pada aplikasi tersebut ada 3, yaitu: **cOlahraga**, **cMembaca**, dan **cMenulis**

KODE TOMBOL PROSES

```
def OnTmbProsesButton(self, event):
    nama = self.txt_nama.GetValue()
    if self.cOlahraga.GetValue() == True :
        olahraga = "Ya"
    else :
        olahraga = "Tidak"
    if self.cMembaca.GetValue() == True :
        membaca = "Ya"
    else :
```

```
membaca = "Tidak"

if self.cMenulis.GetValue()==True :

    menulis ="Ya"

else :

    menulis ="Tidak"

pesan1 = "Nama : "+nama+"\n"\
+ "Hobby : "+"\n"\
+ "Olahraga : "+olahraga\
+"\\n"+ "Membaca : " + membaca\
+"\\n"+ "Menulis : " + menulis

self.pesan = wx.MessageDialog(self,\

pesan1,"PESAN",wx.OK)

self.pesan.ShowModal()
```

Karakter “\n” berfungsi untuk ganti baris. Method pada CheckBox sama dengan pada RadioButton, yaitu:

- `GetValue()` → membaca data yang dipilih

Contoh:

```
if self.cOlahraga.GetValue()==True :
```

artinya: jika **cOlahraga** dicentang pilihannya

- `SetValue()` → mensetting apakah suatu CheckBox tercentang atau tidak

Contoh :

`self.cOlahraga.SetValue(True)`

akibat perintah tersebut **cOlahraga** menjadi tercentang.

BAB VI

COMBOBOX, LISTBOX, DAN LISTCTRL

6.1 WX.COMBOBOX

wx.ComboBox memiliki banyak method, namun yang sering digunakan adalah:

1. Append()

Untuk menambahkan data baru ke dalam ComboBox.

Contoh:

```
self.cmbPendidikan.Append("S1")
```

menambahkan item data "S1" ke dalam wx.ComboBox bernama **cmbPendidikan**.

2. GetStringSelection()

Membaca data yang dipilih oleh user

Contoh:

```
self.cmbPendidikan.GetStringSelection()
```

3. SetStringSelection()

Menampilkan data pada Combo sesuai pilihan yang ada.

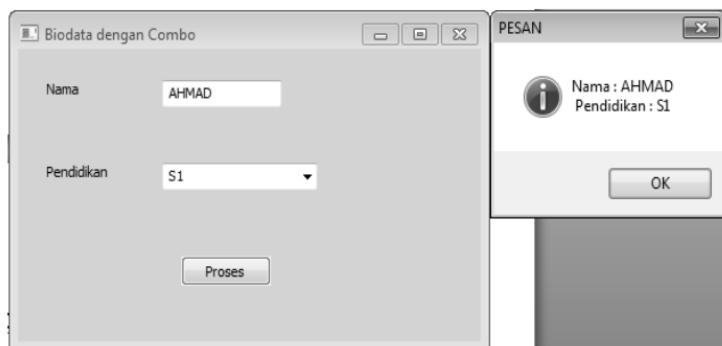
Contoh:

```
self.cmbPendidikan.SetStringSelection("S1")
```

Kode tersebut menjadikan ComboBox **cmbPendidikan** menampilkan pilihan “S1”.

APLIKASI BIODATA DENGAN COMBOBOX

Agar lebih mudah dalam memahami penggunaan ComboBox, berikut adalah contoh aplikasi sederhana Biodata Karyawan dengan ComboBox.



Gambar 6.1 Aplikasi Biodata dengan ComboBox

MENGISI ITEM PILIHAN COMBOBOX

Mengisi item pilihan pada ComboBox bisa dengan 2 cara (pilih salah satu):

1. Mengisikan langsung pada saat perancangan dengan setting Inspector.

Isikan item pada Tab **Constr** di bagian **Choices**. Pisahkan masing-masing item dengan tanda koma dan tiap item diapit oleh petik atas satu.

Contoh:

Choices : ['SD','SMP','SMA','S1']

Dengan cara ini pada saat perancangan, ComboBox sudah terlihat isinya

2. Mengisikan dengan kode program.

Kode program untuk mengisi item ComboBox diletakkan setelah kode :

```
def __init__(self, parent):  
    self._init_ctrls(parent)
```

Kode-kode yang diketikkan tepat di bawah dua baris kode tersebut akan dijalankan pada saat program pertama kali dijalankan (*running*). Mirip dengan **Load Event** atau **Activate Event** pada bahasa pemrograman lain semacam VB6, Delphi, dan semisalnya.

Pada contoh aplikasi ini, kode lengkap untuk mengisikan item pilihan ke ComboBox adalah:

```
def __init__(self, parent):  
    self._init_ctrls(parent)  
  
    self.cmbPendidikan.Append("SD")  
  
    self.cmbPendidikan.Append("SMP")  
  
    self.cmbPendidikan.Append("SMA")  
  
    self.cmbPendidikan.Append("S1")
```

KODE TOMBOL PROSES

Tombol Proses masih belum diubah *name*-nya, sehingga secara default namanya masih **Button1**.

```
def OnButton1Button(self, event):
    nama = self.txt_nama.GetValue()
    pnd = self.cmbPendidikan.GetStringSelection()
    a = "Nama : " + nama +"\n"+\
        " Pendidikan : " + pnd
    self.pesan = wx.MessageDialog(self,\n
        a,"PESAN",wx.OK)
    self.pesan.ShowModal()
```

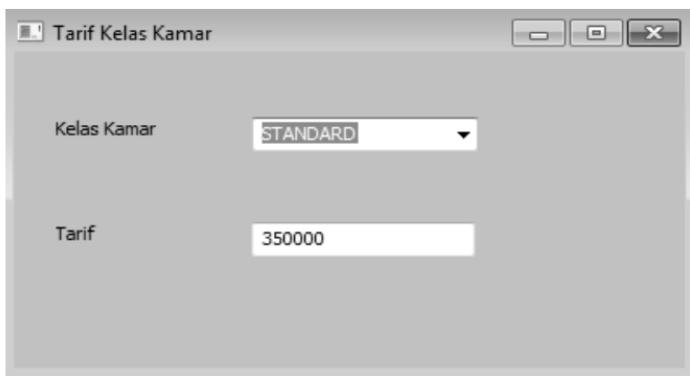
EVENT wx.EVT_COMBOBOX

Event yang terjadi saat ComboBox dipilih. Berikut adalah contoh aplikasi yang mendemonstrasikan penggunaan event wx.EVT_COMBOBOX

APLIKASI KELAS KAMAR HOTEL

User memilih kelas kamar pada ComboBox, sehingga akan ditampilkan secara otomatis tarif kamar per hari sesuai dengan kelas yang dipilih, dengan ketentuan:

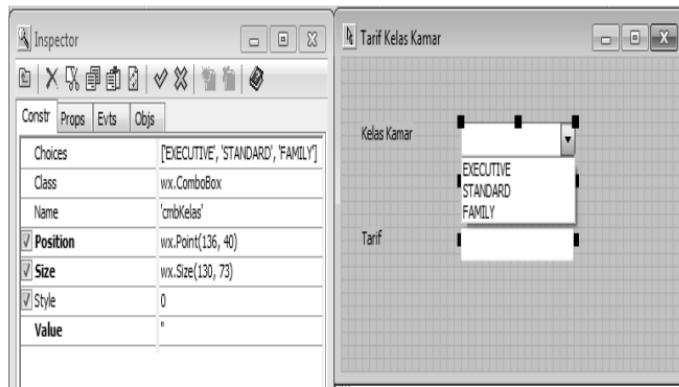
Kelas	Tarif
EXECUTIVE	650000
STANDARD	350000
FAMILY	250000



Gambar 6.2 Tampilan Aplikasi Kelas Kamar

PENGISIAN ITEM PILIHAN PADA COMBOBOX

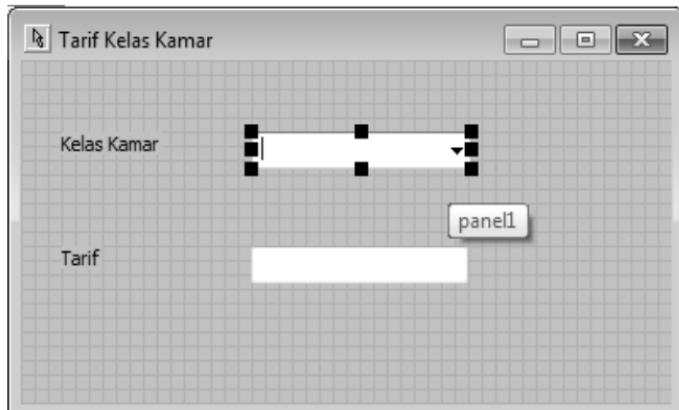
Pada aplikasi ini pengisian item pilihan pada ComboBox dilakukan melalui inspector dengan mengubah nilai **Choices** pada tab **Constr** menjadi : ['EXECUTIVE', 'STANDARD', 'FAMILY']



Gambar 6.3 Pengisian Pilihan Combo Melalui Inspector

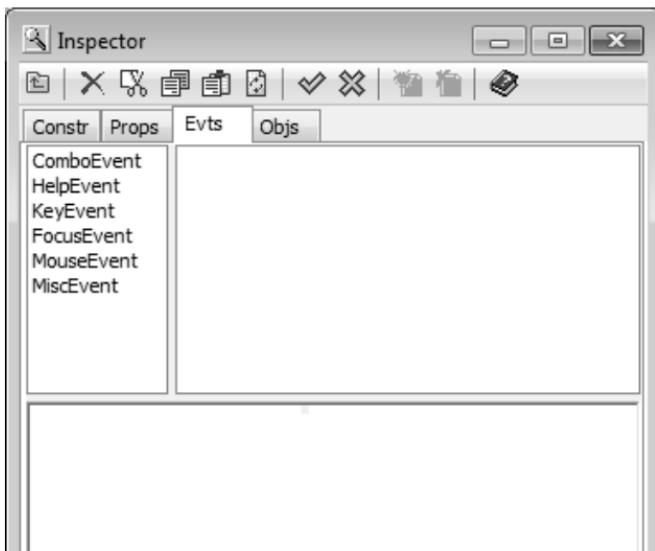
MEMBUAT EVENT wx.EVT_COMBOBOX

1. Pada tampilan desain (Editor), ComboBox aktif terpilih. Name ComboBoxnya adalah **cmbKelas**



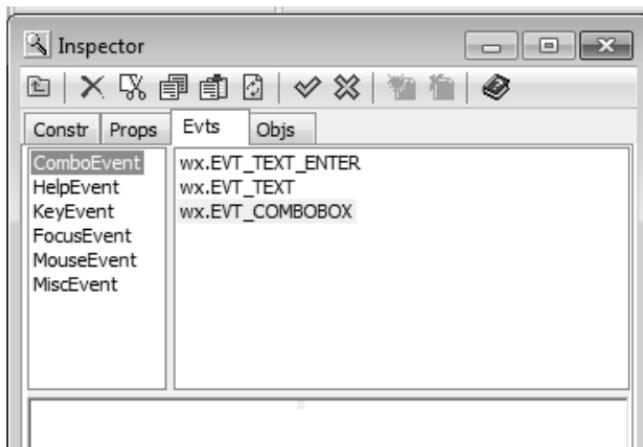
Gambar 6.4 ComboBox Terpilih

2. Pilih Tab Evts pada Inspector



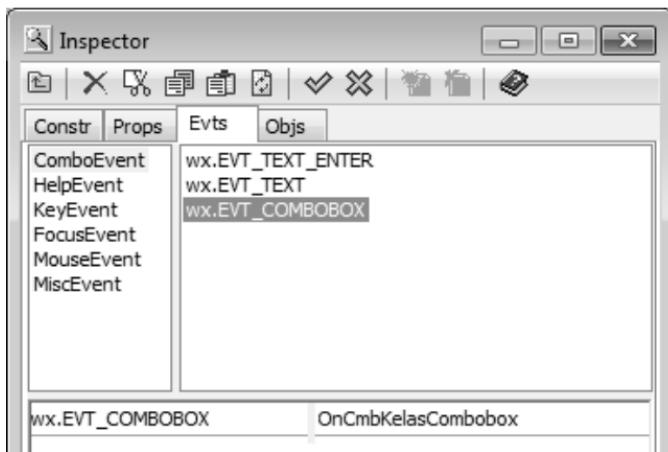
Gambar 6.5 Tab Evt Inspector untuk ComboBox

3. Klik **ComboEvent**



Gambar 6.6 ComboEvent diklik

4. Klik 2 kali pada **wx.EVT_COMBOBOX** sehingga menghasilkan **OnCmbKelasCombobox**



Gambar 6.7 Klik 2 kali pada wx.EVT_COMBOBOX menghasilkan OnCmbKelasCombobox

Hasil dari langkah-langkah tersebut adalah kode sebagai berikut:

```
def OnCmbKelasCombobox(self, event):
    event.Skip()
```

Baris **event.Skip()** dihapus dan kemudian diganti beberapa baris kode sehingga menjadi:

```
def OnCmbKelasCombobox(self, event):
    pil = self.cmbKelas.GetStringSelection()
    if pil=="EXECUTIVE" :
        self.txt_tarif.SetValue("650000")
```

```
elif pil=="STANDARD" :  
    self.txt_tarif.SetValue("350000")  
  
else :  
    self.txt_tarif.SetValue("250000")
```

6.2 WX.LISTBOX

Object wx.ListBox terletak di tab **List Controls** di bagian paling kiri



Gambar 6.8 Posisi wx.ListBox

Method yang sering digunakan pada wx.ListBox adalah:

1. Append()

Menambahkan item data baru ke dalam ListBox.

Contoh :

```
self.listBox1.Append("Ahmad")
```

2. Clear()

Menghapus seluruh item pilihan dalam ListBox.

Contoh :

```
self.listBox1.Clear()
```

3. GetStringSelection()

Membaca item data yang dipilih user.

Contoh :

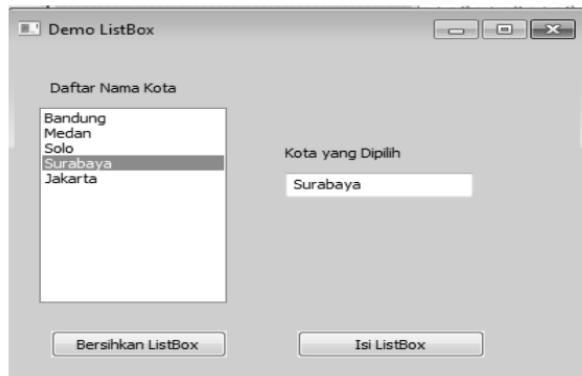
```
self.listBox1.GetStringSelection()
```

APLIKASI DEMO LISTBOX

Berikut ini adalah aplikasi sederhana yang mendemonstrasikan method-method pada ListBox tersebut.

SKENARIO ALUR PROGRAM

Pada saat form pertama kali ditampilkan ListBox akan terisi dengan daftar beberapa nama kota-kota besar di Indonesia. Jika daftar kota itu diklik pada pilihan, akan ditampilkan pada nama kota yang terpilih. Ada dua tombol, yaitu tombol **Bersihkan ListBox** dan **Isi ListBox**. Tombol **Bersihkan ListBox** akan membersihkan ListBox dari item-item data yang telah ada. Sedangkan tombol **Isi ListBox** akan mengisi ListBox dengan item-item daftar nama kota.



Gambar 6.9 Aplikasi Demo ListBox

Daftar Object pada Frame

Jenis Object	Name
ListBox	list_kota
TextCtrl	txt_kota
Button ('Bersihkan ListBox')	Button1
Button ('Isi ListBox')	Button2

KODE INTI

```
def OnList_kotaListbox(self, event):
    self.txt_kota.SetValue \
        (self.list_kota.GetStringSelection())
```

```
def OnButton1Button(self, event):
    self.list_kota.Clear()

def OnButton2Button(self, event):
    self.list_kota.Clear()

    self.list_kota.Append('Bandung')

    self.list_kota.Append('Medan')

    self.list_kota.Append('Solo')

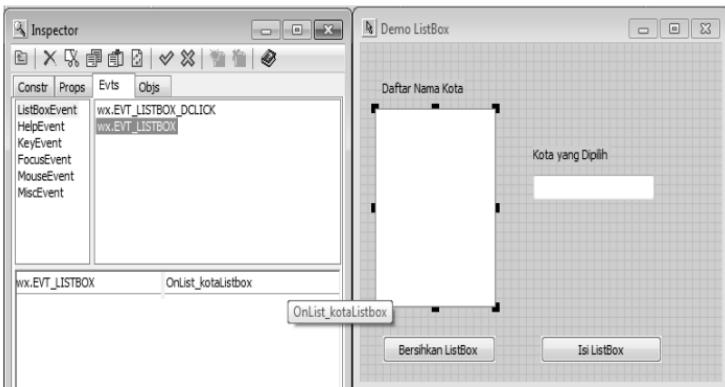
    self.list_kota.Append('Surabaya')

    self.list_kota.Append('Jakarta')
```

Catatan : Mengisikan item pada tampilan awal menggunakan setting Choices pada Inspector

MEMBUAT EVENT PADA LISTBOX

Pada contoh aplikasi di atas terdapat penggunaan event pada listbox sehingga jika listbox diklik nama kota akan ditampilkan pada textCtrl **txt_kota**. Event tersebut adalah **wx.EVT_LISTBOX**.



Gambar 6.10 Event `wx.EVT_LISTBOX`

Pembuatan event tersebut akan menghasilkan baris kode otomatis berikut:

```
def OnList_kotaListbox(self, event):
    event.Skip()
```

Selanjutnya, kode `event.Skip()` tersebut akan dihapus dan diganti, sehingga menjadi:

```
def OnList_kotaListbox(self, event):
    self.txt_kota.SetValue(
        (self.list_kota.GetStringSelection()))
```

yang berfungsi untuk menampilkan pilihan item kota pada ListBox `list_kota` ke textctrl `txt_kota`

6.3 wx.ListCtrl

Object **wx.ListCtrl** memiliki fungsi yang mirip dengan grid pada bahasa pemrograman lain, yaitu menampilkan data dalam format kolom dan baris.



Gambar 6.11 Posisi wx.ListCtrl pada Toolbar Zope Editor

Sifat wx.ListCtrl :

1. Hanya bisa menampung data berbentuk string.
2. Penghitungan nomor index kolom dan baris dimulai dari 0.

Contoh : kolom nomor 2 dari kiri dihitung sebagai kolom ke-1

3. **Style (tab Constr)** pada Inspector harus disetting menjadi **wx.LC_REPORT** saja.
4. Untuk mengatur jumlah dan caption pada tiap kolom aturlah pada bagian **Columns** (tab Props) di Inspector → akan dijelaskan selanjutnya, InsyaAllah

FUNGSI – FUNGSI YANG SERING DIGUNAKAN

Asumsikan bahwa `name` `wx.ListCtrl` yang kita gunakan adalah `lc`.

1. Menghitung Jumlah Baris

Method : `GetItemCount()`

```
jml_baris = self.lc.GetItemCount()
```

2. Menghapus Seluruh Item

Method : `DeleteAllItems()`

```
self.lc.DeleteAllItems()
```

3. Menambah Data Baru

Jika kolom yang ada adalah 2, pada kolom ke-1 akan diisi dengan tulisan ‘Assalamualaikum’ sedangkan kolom ke-2 akan berisi tulisan ‘Apa Kabar’ maka kodennya adalah sebagai berikut:

```
jml_baris = self.lc.GetItemCount()
```

```
self.lc.InsertStringItem(jml_baris,'Assalamualaikum')
```

```
self.lc.SetStringItem(jml_baris,1,'Apa Kabar')
```

4. Membaca Data pada Kolom dan Baris Tertentu

Method : `GetItem(no_baris,no_kolom).GetText()`

Contoh kode :

```
kd_brg = self.lc.GetItem(0,0).GetText()
```

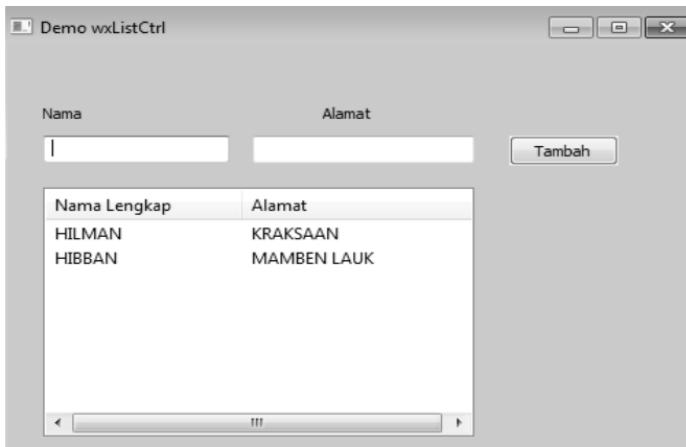
5. Membaca Nomor Baris yang Diklik

```
no_baris = event.m_itemIndex
```

Tambahkan event pada wx.ListCtrl agar bisa mengeksekusi kode pada saat diklik. Event tersebut adalah:**ListEvent>wx.EVT_LIST_ITEM_SELECTED**

APLIKASI DEMO WX.LISTCTRL

Contoh aplikasi berikut akan menunjukkan cara kerja wx.ListCtrl. Ada 2 buah textctrl untuk menerima inputan nama dan alamat. Selanjutnya, jika ditekan tombol **Tambah**, maka data nama dan alamat tersebut akan berpindah ke wx.ListCtrl. Jika data pada wx.ListCtrl di-klik, maka data tersebut akan ditampilkan pada textctrl nama dan alamat.



Gambar 6.12 Aplikasi Demo wx.ListCtrl

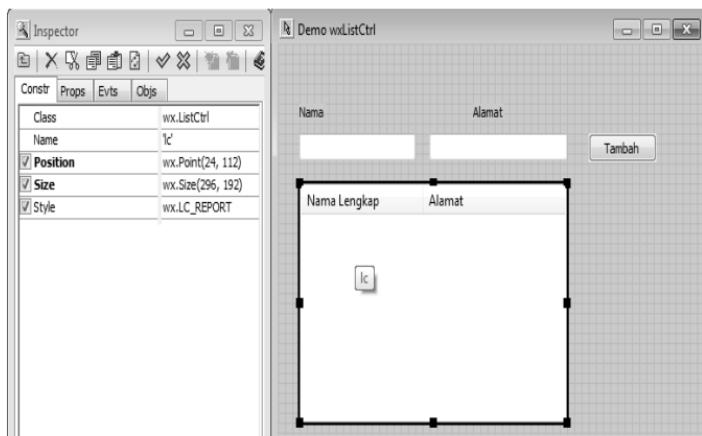
Ada 2 event utama, yaitu event pada tombol Tambah dan event ketika ListCtrl diklik.

PERANCANGAN LISTCTRL

Tahap perancangan ListCtrl meliputi setting Inspector dan pengaturan kolom (Heading/ judul dan lebarnya).

SETTING INSPECTOR LISTCTRL

Pastikan bahwa **Style** pada Inspector hanya disetting True untuk **wx.LC_REPORT**. Secara default, Stylenya adalah **wx.LC_ICON**.



Gambar 6.13 Setting Style menjadi **wx.LC_REPORT**

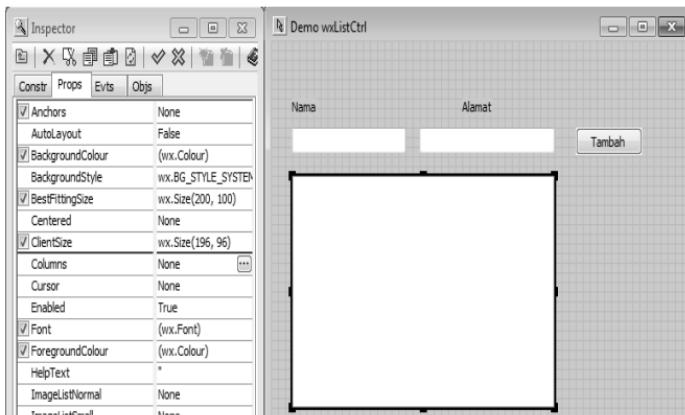
KESALAHAN YANG SERING TERJADI

Style **wx.LC_ICON** belum diset menjadi False tapi langsung menset **wx.LC_REPORT** menjadi True, akibatnya ada 2 Style

PENGATURAN JUMLAH DAN LEBAR KOLOM

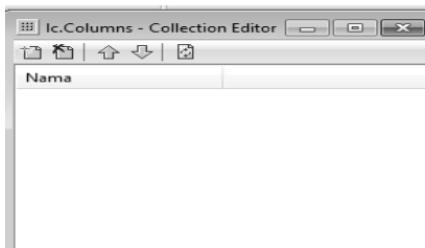
Langkah-langkah mengatur jumlah dan lebar kolom:

1. Pastikan object wx.ListCtrl terpilih (aktif) dan buka tab Props. Pada bagian Columns klik tombol kecil bertanda 3 titik (...)



Gambar 6.14 Props > Columns pada Inspector

2. Akan muncul window **Ic.Columns-Collection Editor**. Nama wx.ListCtrl kita adalah **Ic**.



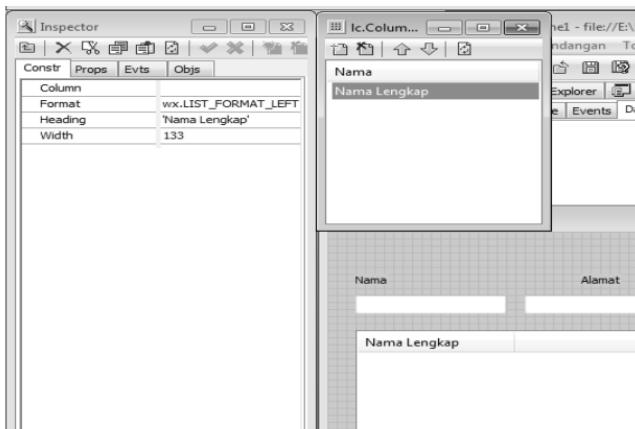
Gambar 6.15 Ic.Columns-Collection Editor

Kolom yang akan dirancang adalah 2 buah. Satu berjudul ‘Nama Lengkap’ dan satunya lagi ‘Alamat’.

3. Klik Icon tambah kolom (di pojok kiri), dan setting Headingnya menjadi ‘**Nama Lengkap**’ dan width menjadi 133.



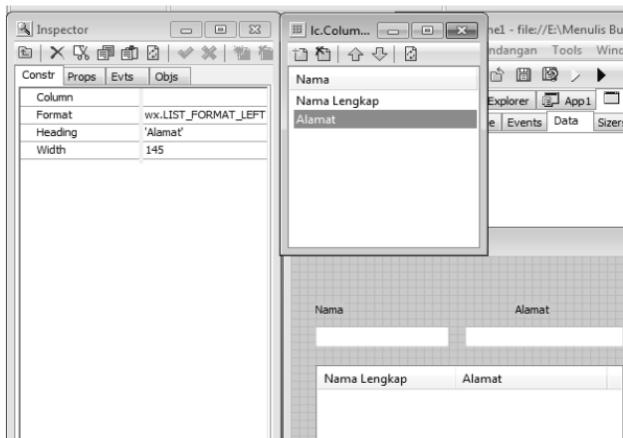
Gambar 6.16 Icon Tambah Kolom



Gambar 6.17 Penambahan Kolom ‘Nama Lengkap’

Ingin, perubahan baru terlihat jika diklik tanda **Post** pada Inspector maupun Editor.

4. Lakukan cara yang sama untuk menambah kolom berikutnya, beri nama ‘Alamat’ dan widthnya adalah 145



Gambar 6.18 Penambahan Kolom ‘Alamat’

KODE INTI APLIKASI

```
def OnTmbTambahButton(self, event):
    # Hitung Jumlah Baris
    jml_baris = self.lc.GetItemCount()
    # Isikan nama ke kolom paling kiri
    self.lc.InsertStringItem(jml_baris, \
                           self.txt_nama.GetValue())
    # Isikan alamat ke kolom ke-2
    self.lc.SetStringItem(jml_baris, 1, \
                         self.txt_alamat.GetValue())
    # counter jumlah baris naik 1
    jml_baris = jml_baris + 1
```

```
# Bersihkan Isian TextCtrl  
self.txt_nama.SetValue("")  
self.txt_alamat.SetValue("")  
# Focuskan kurSOR mouse ke txt_nama  
self.txt_nama.SetFocus()  
def OnLcListItemSelected(self, event):  
    no_baris = event.m_itemIndex  
    nama = self.lc.GetItem(no_baris,0).GetText()  
    alamat = self.lc.GetItem(no_baris,1).GetText()  
    self.txt_nama.SetValue(nama)  
    self.txt_alamat.SetValue(alamat)
```

BAB VII

PERULANGAN

Logika perulangan berfungsi untuk melakukan serangkaian instruksi yang dijalankan berulang selama suatu kondisi terpenuhi. Logika perulangan ada pada setiap bahasa pemrograman, yang membedakan hanyalah syntaxnya.

Pada bahasa pemrograman Python, syntax perulangan ada 2 :

1. while

```
while <kondisi> :  
    <pernyataan>
```

2. for

```
for <variable> in <rentang_nilai> :  
    <pernyataan>
```

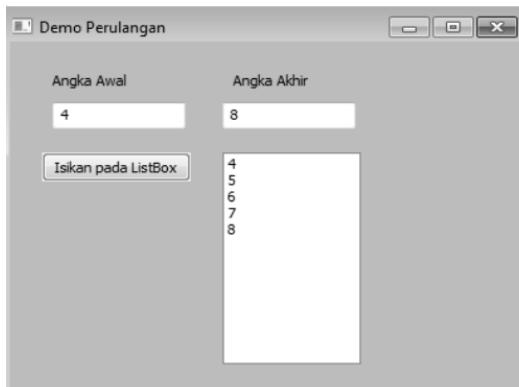
Kedua syntax tersebut akan dibahas pada bab ini

APLIKASI DEMO PERULANGAN

Berikut ini adalah aplikasi sederhana untuk menunjukkan cara kerja kedua syntax perulangan. User diminta untuk mengisikan angka awal dan angka akhir pada textbox yang tersedia. Selanjutnya, user bisa menekan tombol **Isikan pada**

ListBox untuk mengisikan daftar angka dari angka awal sampai akhir.

Contoh : jika angka awal adalah 4 dan angka akhir adalah 8, maka ketika ditekan tombol **Isikan pada ListBox**, nilai yang ditampilkan **ListBox** adalah angka 4 sampai 8.



Gambar 7.1 Aplikasi Demo Perulangan

Nama-nama Object

Jenis Object	Nama
TextCtrl	txt_awal
TextCtrl	txt_akhir
Button	tmbIsi_ListBox
ListBox	Lb

KODE INTI (Syntax while) :

```
def OnTmbIsi_ListBoxButton(self, event):
    awal = int(self.txt_awal.GetValue())
    akhir = int(self.txt_akhir.GetValue())
    while awal <= akhir :
        self.lb.Append(str(awal))
        awal = awal + 1
```

KODE INTI (Syntax for) :

```
def OnTmbIsi_ListBoxButton(self, event):
    awal = int(self.txt_awal.GetValue())
    akhir = int(self.txt_akhir.GetValue())
    i = awal
    for i in range (awal,akhir+1) :
        self.lb.Append(str(i))
```

CATATAN

1. Angka yang diinputkan pada textctrl **txt_awal** maupun **txt_akhir** tipenya adalah String, karena itu kedua-duanya dikonversikan dulu ke integer. Pengubahan ke integer ini perlu karena akan dilakukan logika pembandingan pada syntax while, dan penggunaan fungsi range pada syntax for.

Logika pembandingan pada while :

```
while awal <= akhir :
```

Penggunaan fungsi **range** pada **for**

```
for i in range (awal,akhir+1) :
```

2. Pada syntax **while**, perlu adanya counter untuk menaik atau menurunkan item perulangan.

Counter tersebut ada pada baris kode:

```
awal = awal + 1
```

artinya : nilai variabel **awal** saat ini diisi dengan nilai variable **awal** sebelumnya ditambah 1.

Sedangkan pada syntax **for**, counter secara default akan naik satu tingkat.

3. Tipe data yang bisa ditampilkan pada ListBox haruslah String, karena itu dilakukan konversi dari integer ke String dengan perintah str()

BAB VIII

DASAR AKSES KE DATABASE

Database (DBMS) yang akan digunakan dalam pembahasan kita adalah MySQL. Agar akses ke MySQL bisa dilakukan, harus ada connector Python ke MySQL. Jika IDLE Python yang dipakai adalah 2.5, anda bisa menggunakan file installer **MySQL-python-1.2.2.win32-py2.5**. Intinya, tipe connector harus kompatibel dengan versi IDLE Python yang digunakan. File sudah tersedia pada CD yang disertakan bersama buku ini.

Modul yang digunakan untuk mengakses database adalah MySQLdb. Pastikan bahwa module MySQLdb sudah terinstal dengan baik.

Cara mengecek bahwa MySQLdb sudah bisa digunakan, adalah ketikkan perintah :

```
import MySQLdb
```

pada Shell di Boa Constructor. Jika tidak ada pesan kesalahan (*error*) yang muncul berarti Connector Python-MySQL sudah terinstal dengan baik dan modul MySQLdb siap difungsikan.

The screenshot shows a window titled "Editor - Shell". The menu bar includes "File", "Edit", "Pandangan", "Tools", "Windows", and "Pertolongan". Below the menu is a toolbar with icons for file operations. A tab bar at the bottom has "Shell" selected and "Explorer". The main area displays Python code and its output:
Python 2.5 (r25:51908, Sep 19 2006, 09:52:17) [MSC v.1310 32 bit (Intel)]
wxPython 2.8.10.1, Boa Constructor 0.6.1
Type "copyright", "credits" or "license" for more information.
->>> import MySQLdb
->>> |

Gambar 8.1 Pengecekan Modul MySQLdb (berjalan sukses)

8.1 PRINSIP DASAR AKSES KE DATABASE

1. Mengimport MySQLdb

```
import MySQLdb
```

2. Buat Koneksi

Contoh Syntax :

```
conn=MySQLdb.connect(host="localhost",user="root",\  
passwd="",db="LatPython")
```

Perintah tersebut membuat suatu variable koneksi bernama **conn** yang letak servernya adalah di komputer yang sama dengan letak program yang dirancang (localhost), menggunakan user **root** dan password kosong serta nama databasenya adalah **LatPython**.

3. Buat Cursor.

Cursor adalah media perantara antara form (Frame) dengan Database. Data yang diambil dari database, ditampung oleh cursor terlebih dahulu, kemudian ditampilkan dan diproses oleh Frame. Demikian pula, data yang akan disimpan ke Database akan ditampung oleh Cursor terlebih dahulu. Cursorlah yang akan mengeksekusi perintah-perintah SQL (*Structured Query Language*) dalam mengakses database.

Contoh syntax membuat Cursor:

```
cur = conn.cursor()
```

Perintah tersebut membuat variable Cursor bernama **cur** dari koneksi bernama **conn**. Ingat, Cursor tidak bisa dibuat tanpa membuat koneksi terlebih dahulu.

4. Membuat variabel perintah SQL

Contoh Syntax :

```
sql = " insert into mhs (nim,nama) values (%s,%s)"%(txt_nim,txt_nama)
```

Perintah tersebut akan menyimpan satu record baru ke tabel **mhs** untuk field **nim** dan **nama** dari variabel **txt_nim** dan **txt_nama**. Masing-masing tipe data field tersebut adalah String sehingga formatnya menggunakan '**%s**'.

5. Mengeksekusi perintah SQL.

Contoh Syntax:

```
cur.execute(sql)
```

6. Commit pada Database

Perintah commit pada database berfungsi untuk melakukan perubahan secara permanen pada isi database setelah sebelumnya query (SQL) diproses di Cursor. Commit hanya berlaku untuk query yang mempengaruhi isi database, seperti simpan, edit, dan hapus.

Contoh Syntax :

```
conn.commit()
```

Secara ringkas, tahapan tersebut adalah sebagai berikut:

TAHAPAN	CONTOH SYNTAX
Mengimport MySQLdb	import MySQLdb
Membuat Koneksi	conn=MySQLdb.connect\ (host="localhost",\ user="root",passwd="",db="LatPython")
Membuat Cursor	cur = conn.cursor()
Membuat Variable Perintah SQL	sql = " insert into mhs \ (nim,nama) values (%s,%s)%\ (txt_nim,txt_nama)
Mengeksekusi Perintah SQL	cur.execute(sql)

TAHAPAN	CONTOH SYNTAX
Commit pada Database	conn.commit()

Tahapan tersebut harus berurutan.

Selanjutnya, kita akan melihat contoh aplikasi sederhana akses ke database untuk proses Cari/Lihat Data, Simpan, Edit, dan Hapus. Masing-masing fungsi tersebut akan ditampilkan pada Frame tersendiri untuk memudahkan pembaca memahami contoh syntax masing-masing.

Database : Akademik

Tabel : mhs

Field	Tipe Data	Lebar
nim	VarChar	8
nama	VarChar	25

PEMBUATAN DATABASE DAN STRUKTUR TABEL

DBMS yang digunakan adalah MySQL. Untuk memudahkan, sebaiknya instal MySQL yang sudah satu paket dengan PHP dan Apache. Biasanya terbungkus dalam paket XAMPP atau AppServ, dan semisalnya.

Pada contoh kita ini menggunakan XAMPP. Secara default, di Sistem Operasi Windows XAMPP akan

terinstal pada C:\xampp. Klik 2 kali pada icon XAMPP Control.



Gambar 8.2 Icon xampp-control

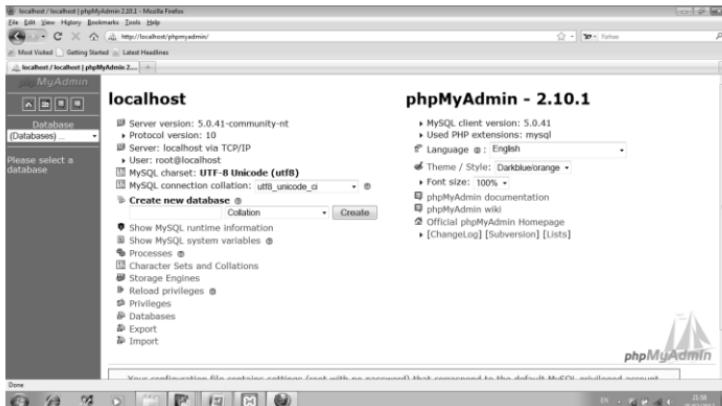
Akan muncul XAMPP Control Panel, untuk mengatur pengaktifan MySQL dan Apache Web Server.



Gambar 8.3 XAMPP Control Panel

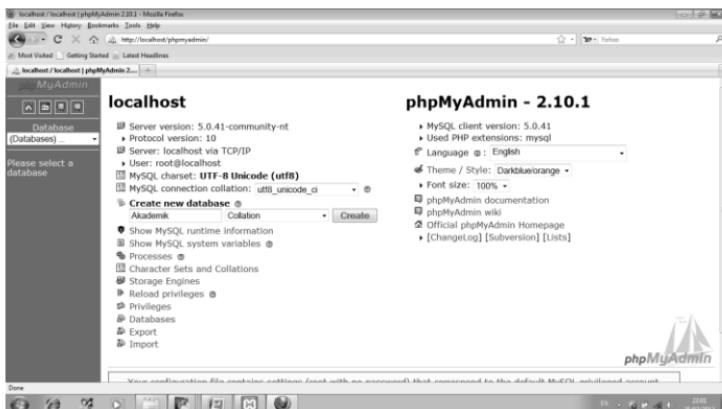
Pastikan Apache dan MySQL dalam status **Running** seperti ditunjukkan pada **Gambar 8.3**.

Selanjutnya, jalankan Web Browser semacam Mozilla Firefox atau yang lain. Pada alamat URL ketikkan : localhost/phpmyadmin. Akan muncul tampilan awal **phpmyadmin**.

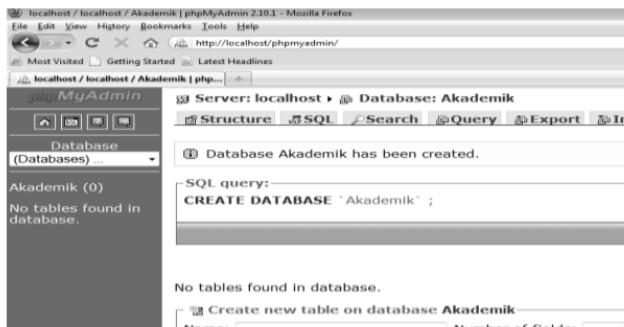


Gambar 8.4 Tampilan Awal phpmyadmin

Pada textbox di bawah tulisan **Create new database** ketikkan nama database yang akan kita buat yaitu **Akademik**, kemudian tekan tombol **Create**



Gambar 8.5 Membuat Database Baru ‘Akademik’



Gambar 8.6 Database Akademik Telah Tercipta

Selanjutnya, buatlah tabel **mhs** pada database tersebut.

Di bawah tulisan **Create new table on database Akademik**, isikanlah:

Name : **mhs**

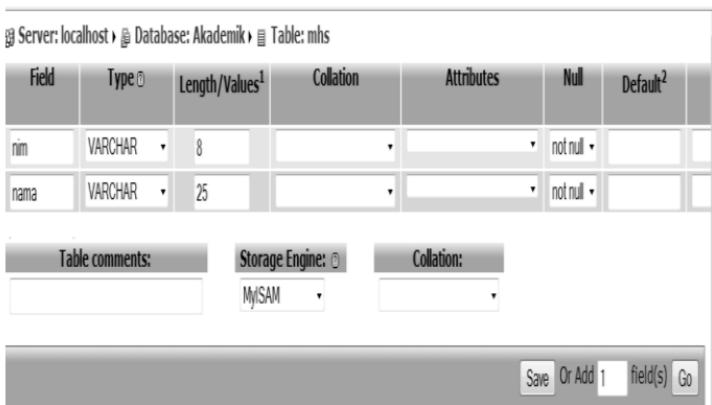
Number of fields : **2**

Kemudian tekan tombol **Go**

A screenshot of a 'Create new table on database Akademik' form. It has two input fields: 'Name:' containing 'mhs' and 'Number of fields:' containing '2'. To the right of the 'Number of fields:' field is a 'Go' button.

Gambar 8.7 Pembuatan Tabel mhs 2 field

Berikutnya, atur field nim dan nama seperti pada Gambar 8.8 kemudian tekan tombol **Save**.



Gambar 8.8 Setting Struktur Tabel mhs

The screenshot shows the MySQL Workbench interface after the table has been created. On the left, the database tree shows "Database (Databases) ..." expanded to show "Akademik (1)" which contains "mhs". On the right, a message box says "Table `Akademik`.`mhs` has been created." Below it is a "SQL query:" section with the following code:

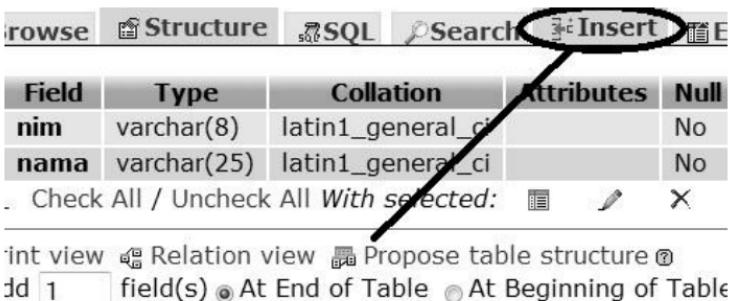
```
CREATE TABLE `mhs` (
  `nim` VARCHAR(8) NOT NULL,
  `nama` VARCHAR(25) NOT NULL
) ENGINE = MYISAM ;
```

Gambar 8.9 Table mhs sudah tercipta

Cobalah untuk mengisikan 2 record data terserah anda. Bisa juga lebih dari 2 record. Adanya data tersebut akan berguna untuk menguji aplikasi mencari data yang akan kita buat.

MENGISIKAN RECORD BARU

Kita coba mengisikan record baru ke tabel **mhs** yang sudah kita buat. Klik tab **Insert**



The screenshot shows the MySQL Workbench interface with the 'Insert' tab highlighted by a red circle. Below the tabs, there is a table definition for the 'mhs' table with two fields: 'nim' and 'nama'. Underneath the table, there is a section for selecting rows to insert and a dropdown menu for table structure. A large black diagonal slash is drawn across the entire screenshot.

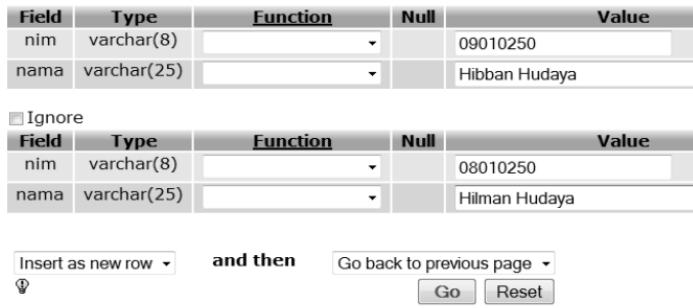
Field	Type	Collation	Attributes	Null
nim	varchar(8)	latin1_general_ci		No
nama	varchar(25)	latin1_general_ci		No

Check All / Uncheck All With selected:

int view Relation view Propose table structure dd field(s) At End of Table At Beginning of Table

Gambar 8.10 Klik Tab Insert untuk Menambah Record Baru

Isikan data pada 2 record baru sesuai dengan keinginan anda. Misalkan, seperti pada Gambar 8.11. Kalau sudah, tekan tombol **Go**.



The screenshot shows the 'Insert' form for the 'mhs' table. Two records are being inserted:

Field	Type	Function	Null	Value
nim	varchar(8)		Null	09010250
nama	varchar(25)		Null	Hibban Hudaya

Ignore

Field	Type	Function	Null	Value
nim	varchar(8)		Null	08010250
nama	varchar(25)		Null	Hilman Hudaya

Insert as new row Go back to previous page

Gambar 8.11 Mengisikan 2 Record Baru

Hasilnya, 2 record baru sudah ditambahkan

The screenshot shows the MySQL Workbench interface. On the left, there's a tree view under 'Database (Databases) ...' with 'Akademik (1)' expanded, showing a single table 'mhs'. On the right, a message box says 'Inserted rows: 2'. Below it, the SQL query is displayed:

```
SQL query:  
INSERT INTO `Akademik`.`mhs` (  
    `nim`,  
    `nama`  
)  
VALUES (  
    '09010250', 'Hibban Hudaya'  
, (  
    '08010250', 'Hilman Hudaya'  
);
```

Gambar 8.12 Dua Record Baru Ditambahkan

Klik tab **Browse** untuk melihat hasil 2 record tersebut

The screenshot shows the 'Browse' tab in MySQL Workbench. It displays a table with two rows of data. The columns are labeled 'nim' and 'nama'. The data is as follows:

	nim	nama
<input type="checkbox"/>	09010250	Hibban Hudaya
<input type="checkbox"/>	08010250	Hilman Hudaya

Below the table, there are several controls: 'Check All / Uncheck All With selected:', 'Show: 30 row(s) starting from record # 0', 'in horizontal mode and repeat headers after 100 cells', and buttons for 'horizontal', 'mode', and 'repeat headers'.

Gambar 8.13 Tampilan Browse 2 Record

Untuk selanjutnya, penambahan record baru bisa melalui aplikasi yang akan kita rancang.

8.2 Melihat dan Mencari Data

Aplikasi pertama yang kita buat adalah untuk melihat atau mencari data. Semua fungsi melihat, menyimpan, menghapus, dan mengedit kita buat pada frame tersendiri. Aplikasi melihat dan mencari data terdiri dari 2 bagian, yaitu : menampilkan data pada 1 record dan menampilkan seluruh data.

Bagian yang pertama (Mencari Data 1 Record), user menginputkan NIM yang akan dicari, selanjutnya jika ditekan tombol **Cari** akan menampilkan nama berdasarkan NIM tersebut jika ada dalam database. Jika tidak ada akan ditampilkan pesan bahwa data dengan NIM tersebut tidak ada.

Sedangkan pada bagian kedua, Frame akan menampilkan data NIM dan nama seluruh record yang ada di dalam database pada wx.ListCtrl.

8.2.1 Mencari Data 1 Record



Gambar 8.14 Cari Data Mahasiswa

LISTING KODE

```
#Boa:Frame:Frame1

import wx, MySQLdb

conn=MySQLdb.connect(host="localhost",user="root",p
asswd="",db="Akademik")

cur = conn.cursor()

def create(parent):
```

```

return Frame1(parent)

[wxID_FRAME1, wxID_FRAME1PANEL1,
wxID_FRAME1STATICTEXT1,

wxID_FRAME1STATICTEXT2, wxID_FRAME1TMBCARI,
wxID_FRAME1TXT_NAMA,

wxID_FRAME1TXT_NIM,
] = [wx.NewId() for _init_ctrls in range(7)]


class Frame1(wx.Frame):

    def __init__(self, prnt):

        # generated method, don't edit

        wx.Frame.__init__(self, id=wxID_FRAME1, name="",
parent=prnt,

pos=wx.Point(600, 242), size=wx.Size(400, 181),
style=wx.DEFAULT_FRAME_STYLE, title='Cari
Data Mahasiswa')

        self.SetClientSize(wx.Size(384, 143))

        self.panel1 = wx.Panel(id=wxID_FRAME1PANEL1,
name='panel1', parent=self,
pos=wx.Point(0, 0), size=wx.Size(384, 143),
style=wx.TAB_TRAVERSAL)

```

```
    self.panel1.SetBackgroundColour(wx.Colour(211,
211, 211))

    self.staticText1 =
wx.StaticText(id=wxID_FRAME1STATICTEXT1,
label='NIM',
           name='staticText1', parent=self.panel1,
pos=wx.Point(32, 32),
           size=wx.Size(20, 13), style=0)

    self.txt_nim =
wx.TextCtrl(id=wxID_FRAME1TXT_NIM, name='txt_nim',
parent=self.panel1, pos=wx.Point(128, 32),
size=wx.Size(100, 21),
           style=0, value="")

    self.staticText2 =
wx.StaticText(id=wxID_FRAME1STATICTEXT2,
label='Nama', name='staticText2',
parent=self.panel1,
           pos=wx.Point(32, 80), size=wx.Size(28, 13),
style=0)

    self.txt_nama =
wx.TextCtrl(id=wxID_FRAME1TXT_NAMA,
name='txt_nama',
```

```

        parent=self.panel1, pos=wx.Point(128, 80),
size=wx.Size(192, 21),

        style=0, value=")

self.tmbCari =
wx.Button(id=wxID_FRAME1TMBCARI, label='Cari',

        name='tmbCari', parent=self.panel1,
pos=wx.Point(248, 32),

        size=wx.Size(75, 23), style=0)

self.tmbCari.Bind(wx.EVT_BUTTON,
self.OnTmbCariButton,

        id=wxID_FRAME1TMBCARI)

def __init__(self, parent):
    self._init_ctrls(parent)

def OnTmbCariButton(self, event):

    sql = " select * from mhs where nim = '%s'"
%(self.txt_nim.GetValue())

    cur.execute(sql)

    hasil= cur.fetchone()

    if cur.rowcount > 0 :

        self.txt_nama.SetValue(hasil[1])

```

```
else :  
  
    self.pesan = wx.MessageDialog(self,"Data  
Tidak Ada","Konfirmasi",wx.OK)  
  
    self.pesan.ShowModal()
```

PENJELASAN KODE

Object utama pada Frame tersebut adalah 2 TextCtrl: **txt_nim** dan **txt_nama** serta 1 Button(tombol), yaitu **tmbCari**. Tombol tersebut memiliki event yang akan membangkitkan proses pencarian dalam database.

Import pada aplikasi ini menggunakan 2 hal, yaitu **wx** (sebagai library GUI) dan **MySQLDb** sebagai modul akses ke database. Kode yang digunakan :

```
import wx, MySQLDb
```

diletakkan di bagian paling atas.

Selanjutnya, membuat koneksi dengan perintah :

```
conn=MySQLDb.connect(host="localhost",user="root",pass  
wd="",db="Akademik")
```

Diikuti dengan pembuatan Cursor:

```
cur = conn.cursor()
```

Ketiga tahapan tersebut: import, membuat koneksi, dan membuat Cursor diletakkan di bagian atas sebelum kode :

```
def create(parent):
    return Frame1(parent)
.... dan seterusnya.....
```

Kode def create(parent): itu adalah kode otomatis yang diciptakan oleh Boa Constructor ketika kita merancang Frame, sehingga tidak kita bahas maknanya. Sengaja kami tampilkan listing lengkap untuk menampilkan semua kode yang dibuat secara otomatis dan yang ditambahkan/ dedit. Bagian yang kita tambahkan/ dedit ditandai dengan bentuk font yang tebal dan miring.

Kode import, membuat koneksi, dan membuat cursor akan selalu sama pada setiap aplikasi tersisa yang akan kita rancang. Untuk aplikasi selanjutnya, kami hanya akan membahas kode intinya saja. Kode inti tersebut ada pada event button atau yang terletak di bawah kode :

```
def __init__(self, parent):
    self._init_ctrls(parent)
```

Kode inti pada aplikasi ini terletak pada event tombol Cari :

```
def OnTmbCariButton(self, event):
    sql = " select * from mhs where nim = '%s'" % (self.txt_nim.GetValue())
    cur.execute(sql)
    hasil= cur.fetchone()
```

```
if cur.rowcount > 0 :  
    self.txt_nama.SetValue(hasil[1])  
  
else :  
    self.pesan = wx.MessageDialog(self,"Data Tidak  
Ada","Konfirmasi",wx.OK)  
  
    self.pesan.ShowModal()
```

Pertama, kita setting variable SQL untuk pencarian yaitu:

```
sql = " select * from mhs where nim = '%s'"  
%(self.txt_nim.GetValue())
```

maksudnya adalah mencari pada tabel **mhs** yang nilai **nim**-nya sama dengan yang diketikkan pada textctrl **txt_nim**. Karena tipe datanya adalah String (VarChar juga termasuk String), maka menggunakan format '**%s**'.

Kemudian, eksekusi perintah SQL tersebut menggunakan cursor **cur** (yang sudah kita buat sebelumnya) dengan perintah :

```
cur.execute(sql)
```

Berikutnya, ambillah satu baris yang ada di Cursor dan simpan ke variabel **hasil**, dengan perintah:

```
hasil= cur.fetchone()
```

Cek apakah datanya ada, dengan perintah:

```
if cur.rowcount > 0 :
```

Jika ada, tampilkan pada textCtrl **txt_nama**. Isikan ke textCtrl tersebut data pada indeks ke-1 yang berisi data **nama**. Perintahnya adalah:

```
self.txt_nama.SetValue(hasil[1])
```

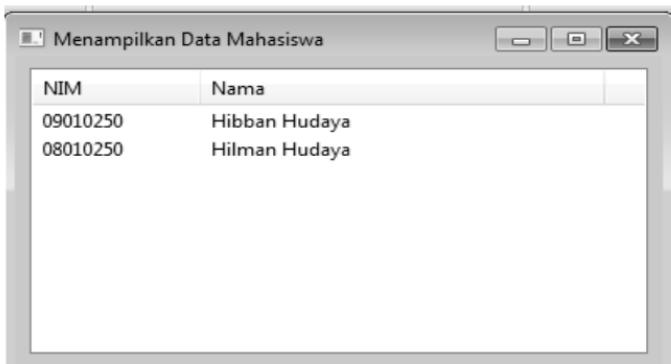
Variable hasil itu adalah sebuah tuple (struktur data mirip array) yang memiliki data lebih dari 1. Pada index ke-0 datanya adalah data **nim**, sedangkan index ke-1 adalah **nama**. Kedua-duanya didapatkan dari hasil eksekusi SQL yang dilakukan Cursor.

Terakhir, jika data yang dicari tidak ada, tampilkan pesan dengan perintah:

```
self.pesan = wx.MessageDialog(self,"Data Tidak Ada","Konfirmasi",wx.OK)  
self.pesan.ShowModal()
```

8.2.2 Menampilkan Seluruh Data pada wx.ListCtrl

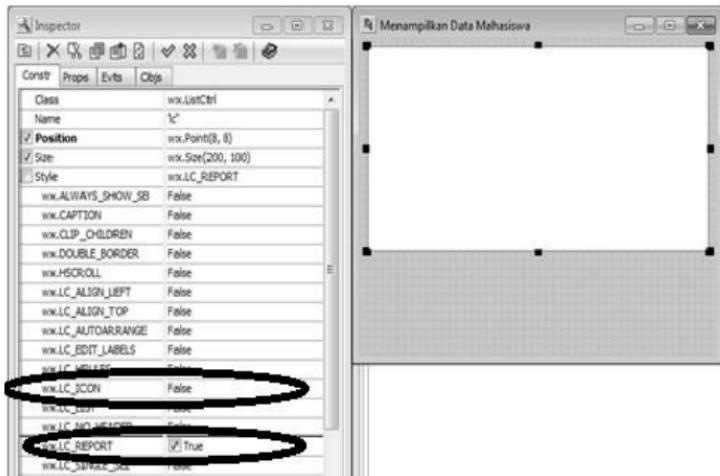
Pada bab VI yang lalu sudah dibahas tentang penggunaan **wx.ListCtrl**. Object tersebut bisa digunakan untuk menampilkan seluruh data pada tabel tertentu di database. Kita akan gunakan untuk membuat aplikasi menampilkan seluruh data mahasiswa yang ada di tabel **mhs** pada database **Akademik**.



Gambar 8.15 Menampilkan Data Seluruh Mahasiswa

PERANCANGAN WX.LISTCTRL

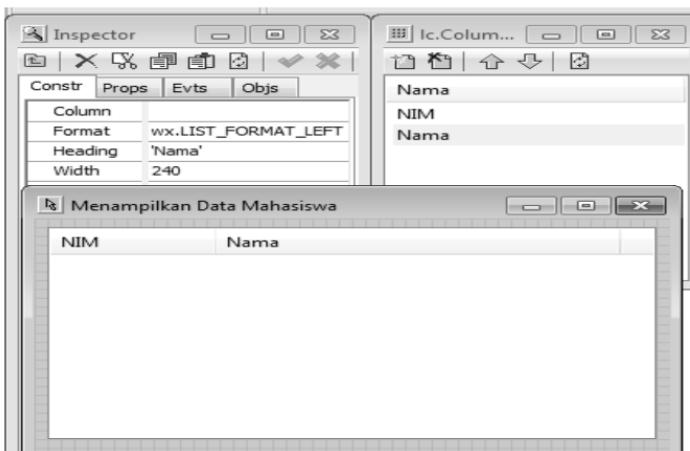
Letakkan wx.ListCtrl dalam Frame yang kita buat, kemudian beri nama dengan **lc**. Aturlah Style pada Inspector agar menjadi **wx.LC_REPORT**.



Gambar 8.16 Style wx.LC_REPORT

Selanjutnya, aturlah **Column**-nya (tab **Props**) agar memiliki 2 kolom dengan judul (Heading): **NIM** dan **Nama**.

Heading	Width
NIM	100
Nama	240



Gambar 8.17 Pengaturan Kolom NIM dan Nama

Jangan lupa untuk mengklik tanda **Post** (centang warna biru) di Inspector dan Editor, agar perubahannya terlihat. Kemudian, isikan kode import, pembuatan koneksi, dan pembuatan kursor, sebagaimana pada aplikasi sebelumnya di bagian atas **Source**

```
1 #Boa:Frame:Frame1
2
3 import wx, MySQLdb
4
5 conn=MySQLdb.connect(host="localhost",
6 user="root",passwd="",db="Akademik")
7 cur = conn.cursor()
8
9 -def create(parent):
10     return Frame1(parent)
11
12 [wxID_FRAME1, wxID_FRAME1LC, wxID_FRAME1PANEL1,
13  - from MySQLdb import Error as MySQLdbError
```

Gambar 8.17 Kode Import, Koneksi, Cursor

Berikutnya, isikan kode di bawah 2 baris kode berikut:

```
def __init__(self, parent):
    self._init_ctrls(parent)
```

sehingga menjadi seperti berikut ini:

```
def __init__(self, parent):
    self._init_ctrls(parent)

    sql = " select * from mhs "

    cur.execute(sql)

    hasil= cur.fetchall()

    k =self.lc.GetItemCount()

    for i in hasil :

        self.lc.InsertStringItem(k,"%s"%i[0])

        self.lc.SetStringItem(k,1,"%s"%i[1])
```

```
k = k + 1
```

PENJELASAN KODE PROGRAM

Pembuatan variabel SQL untuk mencari seluruh data (semua baris dan kolom dalam tabel **mhs**) menggunakan perintah :

```
sql = " select * from mhs "
```

selanjutnya, Cursor **cur** mengeksekusi perintah SQL tersebut :

```
cur.execute(sql)
```

Berikutnya, Cursor **cur** akan mengambil semua data yang sudah diambilnya dari tabel **mhs**, dan menyimpannya ke dalam variabel **hasil** menggunakan perintah :

```
hasil= cur.fetchall()
```

kalau pada contoh aplikasi yang mengambil 1 record method yang digunakan adalah **fetchone()**, maka di sini menggunakan **fetchall()**.

Sebelum data dari Cursor **cur** diisikan pada **wx.ListCtrl lc**, terlebih dahulu dihitung berapa jumlah baris pada **wx.ListCtrl lc** dan disimpan ke dalam variabel **k** menggunakan perintah:

```
k =self.lc.GetItemCount()
```

Lakukan perulangan mengisikan data dari variable **hasil** ke baris-baris **wx.ListCtrl** dengan perintah **for** :

```
for i in hasil :
```

Selanjutnya, isikan ke kolom pertama dengan perintah :

```
self.lc.InsertStringItem(k,"%s"%i[0])
```

data yang diisikan adalah data dari variabel **i** (yang mengambil data di variable **hasil**) dengan nomor index ke-0, yaitu **nim**. Karena tipe datanya adalah VarChar/ String, maka menggunakan format “**%s**”.

Mengisikan ke kolom selanjutnya menggunakan perintah:

```
self.lc.SetStringItem(k,1,"%s"%i[1])
```

Method **InsertStringItem** pada **wx.ListCtrl** digunakan untuk memasukkan data ke kolom paling kiri(disebut kolom ke-0), sedangkan **SetStringItem** untuk kolom-kolom berikutnya. Bedanya lagi, untuk **SetStringItem** harus disebutkan parameter nomor kolomnya.

Syntaxnya adalah:

```
SetStringItem(nomor_baris,nomor_kolom,isi)
```

LISTING KODE LENGKAP FRAME MELIHAT DATA (FrLihatData.py)

```
#Boa:Frame:Frame1  
  
import wx, MySQLdb  
  
conn=MySQLdb.connect(host="localhost",\br/>user="root",passwd="",db="Akademik")
```

```
cur = conn.cursor()

def create(parent):
    return Frame1(parent)

[wxID_FRAME1,
 wxID_FRAME1PANEL1,
] = [wx.NewId() for _init_ctrls in range(3)]

class Frame1(wx.Frame):
    def __init__(self, parent):
        # generated method, don't edit

        parent.InsertColumn(col=0,
format=wx.LIST_FORMAT_LEFT, heading='NIM',
width=100)

        parent.InsertColumn(col=1,
format=wx.LIST_FORMAT_LEFT, heading='Nama',
width=240)

    def __init_ctrls(self, prnt):
        # generated method, don't edit
```

```

wx.Frame.__init__(self, id=wx.ID_FRAME1, name='',
parent=prnt,
    pos=wx.Point(10, 275), size=wx.Size(400, 336),
    style=wx.DEFAULT_FRAME_STYLE,
title='Menampilkan Data Mahasiswa')

self.SetClientSize(wx.Size(384, 298))

self.panel1 = wx.Panel(id=wx.ID_FRAME1PANEL1,
name='panel1', parent=self,
    pos=wx.Point(0, 0), size=wx.Size(384, 298),
    style=wx.TAB_TRAVERSAL)

self.panel1.SetBackgroundColour(wx.Colour(201,
201, 201))

self.lc = wx.ListCtrl(id=wx.ID_FRAME1LC, name='lc',
parent=self.panel1,
    pos=wx.Point(8, 8), size=wx.Size(368, 192),
style=wx.LC_REPORT)

self._init_lc_Columns(self.lc)

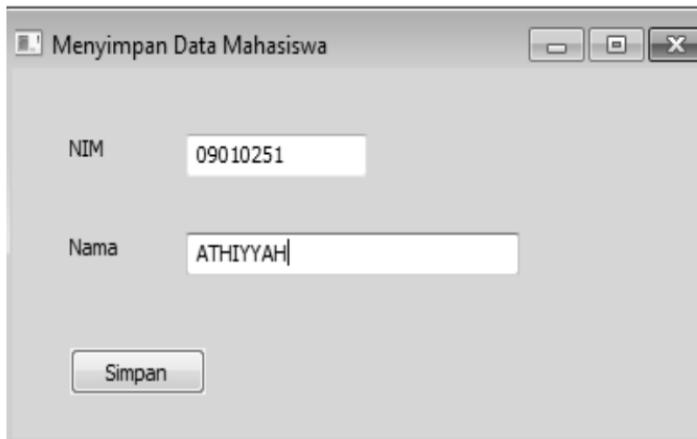
def __init__(self, parent):
    self._init_ctrls(parent)
    sql = "select * from mhs"
cur.execute(sql)

```

```
hasil= cur.fetchall()
k =self.lc.GetItemCount()
for i in hasil :
    self.lc.InsertStringItem(k,"%s"%i[0])
    self.lc.SetStringItem(k,1,"%s"%i[1])
    k = k + 1
```

8.3 Menyimpan Data

Aplikasi Menyimpan Data adalah untuk menyimpan data 1 record mahasiswa. User menginputkan **NIM** dan **Nama** mahasiswa pada TextCtrl yang disediakan, kemudian menekan tombol Simpan, data mahasiswa baru akan tersimpan dalam database.



Gambar 8.18 Simpan Data Mahasiswa

KODE INTI PADA TOMBOL SIMPAN

```
def OnTmbSimpanButton(self, event):
    sql = "insert into mhs values \
    ('%s','%s') %(self.txt_nim.GetValue(), \
    self.txt_nama.GetValue())
    cur.execute(sql)
    conn.commit()
    self.txt_nim.SetValue("")
    self.txt_nama.SetValue("")
    self.txt_nim.SetFocus()
```

Pada tombol Simpan tidak hanya melakukan proses simpan ke database namun juga membersihkan isi textctrl dan mengarahkan focus cursor mouse ke **txt_nim**.

Kode untuk membersihkan isian masing-masing textctrl adalah:

```
self.txt_nim.SetValue("")
self.txt_nama.SetValue("")
```

sedangkan kode untuk memfokuskan cursor mouse ke **txt_nim** adalah:

```
self.txt_nim.SetFocus()
```

**LISTING KODE LENGKAP APLIKASI
MENYIMPAN DATA (FrSimpan.py)**

```
#Boa:Frame:Frame1

import wx, MySQLdb

conn=MySQLdb.connect(host="localhost",user="root",password="",db="Akademik")
cur = conn.cursor()

def create(parent):
    return Frame1(parent)

[wxID_FRAME1,
 wxID_FRAME1PANEL1,
 wxID_FRAME1STATICTEXT1,
 wxID_FRAME1STATICTEXT2,
 wxID_FRAME1TMBSIMPAN, wxID_FRAME1TXT_NAMA,
 wxID_FRAME1TXT_NIM,
] = [wx.NewId() for _init_ctrls in range(7)]

class Frame1(wx.Frame):
    def __init__(self, prnt):
        # generated method, don't edit
        wx.Frame.__init__(self, id=wxID_FRAME1, name="",
parent=prnt,
```

```

pos=wx.Point(493, 238), size=wx.Size(400, 218),
style=wx.DEFAULT_FRAME_STYLE,
title='Menyimpan Data Mahasiswa')

self.SetClientSize(wx.Size(384, 180))

self.panel1 = wx.Panel(id=wxID_FRAME1PANEL1,
name='panel1', parent=self,
pos=wx.Point(0, 0), size=wx.Size(384, 180),
style=wx.TAB_TRAVERSAL)

self.panel1.SetBackgroundColour(wx.Colour(214,
214, 214))

self.staticText1 =
wx.StaticText(id=wxID_FRAME1STATICTEXT1,
label='NIM',
name='staticText1', parent=self.panel1,
pos=wx.Point(32, 32),
size=wx.Size(20, 13), style=0)

self.txt_nim =
wx.TextCtrl(id=wxID_FRAME1TXT_NIM, name='txt_nim',
parent=self.panel1, pos=wx.Point(96, 32),
size=wx.Size(100, 21),
style=0, value="")

```

```

    self.staticText2 =  

wx.StaticText(id=wxID_FRAME1STATICTEXT2,  

              label='Nama', name='staticText2',  

              parent=self.panel1,  

              pos=wx.Point(32, 80), size=wx.Size(28, 13),  

              style=0)  

    self.txt_nama =  

wx.TextCtrl(id=wxID_FRAME1TXT_NAMA,  

           name='txt_nama',  

           parent=self.panel1, pos=wx.Point(96, 80),  

           size=wx.Size(184, 21),  

           style=0, value="")  

    self.tmbSimpan =  

wx.Button(id=wxID_FRAME1TMBSIMPAN, label='Simpan',  

          name='tmbSimpan', parent=self.panel1,  

          pos=wx.Point(32, 136),  

          size=wx.Size(75, 23), style=0)  

    self.tmbSimpan.Bind(wx.EVT_BUTTON,  

self.OnTmbSimpanButton,  

                      id=wxID_FRAME1TMBSIMPAN)  

def __init__(self, parent):  

    self._init_ctrls(parent)

```

```
def OnTmbSimpanButton(self, event):  
  
    sql = "insert into mhs values \  
  
    ("'%s','%s')" %(self.txt_nim.GetValue(),  
  
    self.txt_nama.GetValue())  
  
    cur.execute(sql)  
  
    conn.commit()  
  
    self.txt_nim.SetValue("")  
  
    self.txt_nama.SetValue("")  
  
    self.txt_nim.SetFocus()
```

8.4 Menghapus Data

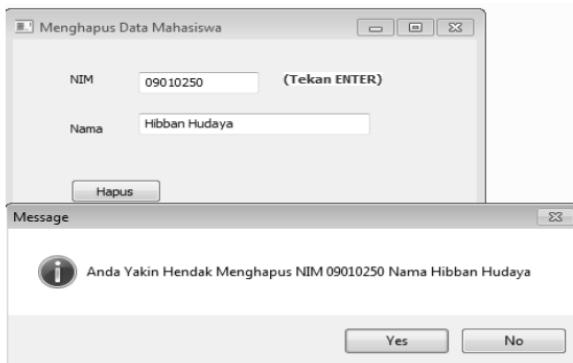
Aplikasi Menghapus Data berikut ini berfungsi untuk menghapus 1 record data. Data yang dihapus acuannya adalah NIM yang dimasukkan oleh user. Setelah user memasukkan NIM dan menekan tombol Enter, akan ditampilkan namanya jika ada dalam database. Jika data itu tidak ada, akan ditampilkan pesan.

Pada tombol hapus, akan dilakukan beberapa hal, yaitu :

1. Mengecek apakah NIM sudah terisi atau belum. Jika belum, akan menampilkan pesan.

2. Mengecek apakah NIM yang diketikkan ada dalam database atau tidak. Kalau tidak ada, akan ditampilkan pesan.
3. Mengkonfirmasi user, apakah benar-benar yakin hendak menghapus data NIM dan Nama tersebut.
4. Membersihkan textctrl NIM dan Nama setelah terjadi proses penghapusan data, dan mengarahkan focus cursor mouse ke textctrl NIM.

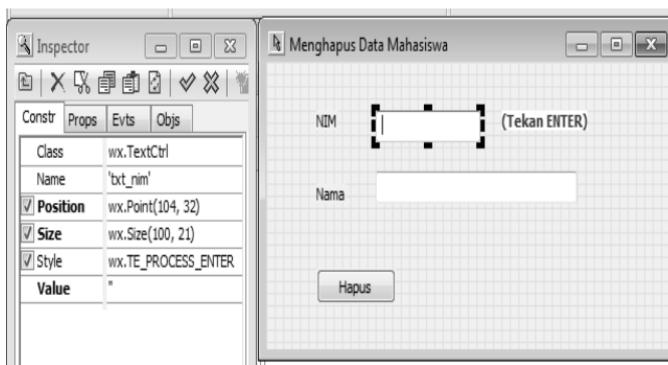
Tahapan validasi dalam proses menghapus data seharusnya memang lebih panjang dibandingkan proses lainnya.



Gambar 8.19 Aplikasi Menghapus Data Mahasiswa

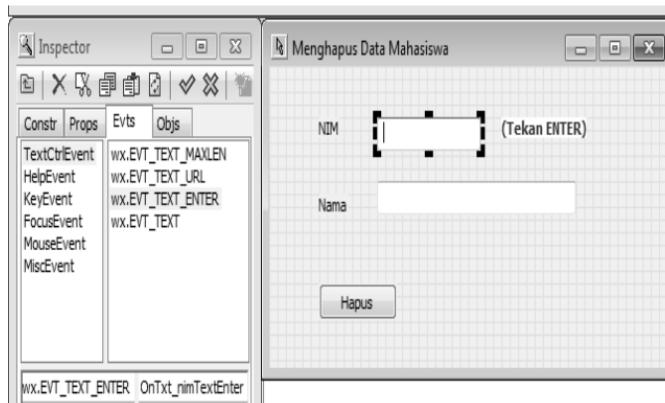
Aplikasi ini juga menggunakan event pada textctrl yang akan diaktifkan saat ditekan tombol Enter (seperti dibahas di Bab IV bagian 4.3). Hal yang harus diperhatikan pada saat perancangan :

1. Frame harus diberi dasar object Panel terlebih dahulu sebelum object yang lain diletakkan. Jika tidak, perpindahan antar object dengan menekan tombol Enter tidak akan berjalan.
2. Setting Style (tab Constr) di Inspector pada TextCtrl **txt_nim** menjadi **wx.TE_PROCESS_ENTER**



Gambar 8.20 Style wx.TE_PROCESS ENTER

3. Buat Event **wx.EVT_TEXT_ENTER** (tab Evt) pada TextCtrls **txt_nim**



Gambar 8.21 wx.EVT_TEXT_ENTER pada txt_nim

Proses pencarian nama berdasarkan NIM yang diinputkan sama dengan pada aplikasi pencarian data 1 record yang sudah dijelaskan sebelumnya. Karena itu tidak akan dibahas lebih detail di bagian ini.

KODE INTI TOMBOL HAPUS:

```
def OnTmbHapusButton(self, event):
    if self.txt_nim.GetValue() == "":
        self.pesan = wx.MessageDialog \
            (self, "NIM belum diisi", "Konfirmasi", wx.OK)
        self.pesan.ShowModal()

    sql = " select * from mhs where nim \ \
        = '%s'" % (self.txt_nim.GetValue())
    cur.execute(sql)
```

```
hasil= cur.fetchone()

if cur.rowcount>0 :

    tanya = wx.MessageDialog(self,\n
        message="Anda Yakin Hendak Menghapus NIM"\n
        +" "+self.txt_nim.GetValue()+" "+\n
        "Nama "+self.txt_nama.GetValue()\n
        ,style = wx.YES_NO)

    if tanya.ShowModal()==wx.ID_YES:

        sql = "delete from mhs where nim \\\n
            = '%s'" % (self.txt_nim.GetValue())

        cur.execute(sql)

    else :

        self.pesan = wx.MessageDialog(self,\n
            "NIM yang akan dihapus tidak terdata \\\n
            di database","Konfirmasi",wx.OK)

        self.pesan.ShowModal()

        self.txt_nim.SetValue("")\n
        self.txt_nama.SetValue("")\n
        self.txt_nim.SetFocus()
```

PENJELASAN KODE

Pengecekan apakah textctrl **txt_nim** sudah terisi atau belum menggunakan perintah :

```
if self.txt_nim.GetValue() == "" :
```

Jika masih belum terisi ditampilkan pesan. Kodenya:

```
    self.pesan = wx.MessageDialog\  
        (self, "NIM belum diisi", "Konfirmasi", wx.OK)  
        self.pesan.ShowModal()
```

Selanjutnya, dilakukan proses pencarian NIM yang diketikkan user dengan yang ada dalam database. Prosesnya sama dengan pada aplikasi pencarian data sebelumnya.

Jika ada, maka akan ditampilkan pesan konfirmasi dengan pilihan Yes atau No. Kode untuk membuat pesan tersebut:

```
tanya = wx.MessageDialog(self,\  
    message="Anda Yakin Hendak Menghapus NIM"\  
    +" "+self.txt_nim.GetValue()+" "+\  
    "Nama "+self.txt_nama.GetValue()\  
,style = wx.YES_NO)
```

Cek kondisi apakah tombol yang dipilih adalah **Yes**. Jika ya, maka dilakukan proses menghapus data.

Kodenya:

```
if tanya.ShowModal() == wx.ID_YES:
```

```
sql = "delete from mhs where nim \
      = '%s'" % (self.txt_nim.GetValue())
      cur.execute(sql)
```

LISTING KODE LENGKAP (FrHapus.py)

```
#Boa:Frame:Frame1

import wx, MySQLdb

conn=MySQLdb.connect(host="localhost",user="root",p
asswd="",db="Akademik")

cur = conn.cursor()

def create(parent):
    return Frame1(parent)

[wxID_FRAME1,
wxID_FRAME1PANEL1,
wxID_FRAME1STATICTEXT1,
wxID_FRAME1STATICTEXT2,
wxID_FRAME1STATICTEXT3, wxID_FRAME1TMBHAPUS,
wxID_FRAME1TXT_NAMA, wxID_FRAME1TXT_NIM,
] = [wx.NewId() for _init_ctrls in range(8)]
```

```
class Frame1(wx.Frame):

    def __init_ctrls(self, prnt):

        # generated method, don't edit

        wx.Frame.__init__(self, id=wx.ID_FRAME1, name="",
parent=prnt,

            pos=wx.Point(387, 245), size=wx.Size(400, 227),

            style=wx.DEFAULT_FRAME_STYLE,
title='Menghapus Data Mahasiswa')

        self.SetClientSize(wx.Size(384, 189))

        self.panel1 = wx.Panel(id=wx.ID_FRAME1PANEL1,
name='panel1', parent=self,

            pos=wx.Point(0, 0), size=wx.Size(384, 189),

            style=wx.TAB_TRAVERSAL)

        self.staticText1 = wx.StaticText(id=wx.ID_FRAME1STATICTEXT1,
label='NIM',

            name='staticText1', parent=self.panel1,
pos=wx.Point(48, 32),

            size=wx.Size(20, 13), style=0)
```

```

    self.txt_nim
wx.TextCtrl(id=wxID_FRAME1TXT_NIM, name='txt_nim',
            parent=self.panel1,      pos=wx.Point(104,      32),
size=wx.Size(100, 21),
            style=wx.TE_PROCESS_ENTER, value="")

    self.txt_nim.Bind(wx.EVT_TEXT_ENTER,
self.OnTxt_nimTextEnter,
            id=wxID_FRAME1TXT_NIM)

    self.staticText2
wx.StaticText(id=wxID_FRAME1STATICTEXT2,
              label='Nama',                  name='staticText2',
parent=self.panel1,
              pos=wx.Point(48,     80),      size=wx.Size(28,     13),
style=0)

    self.txt_nama
wx.TextCtrl(id=wxID_FRAME1TXT_NAMA,
name='txt_nama',
            parent=self.panel1,      pos=wx.Point(104,      72),
size=wx.Size(192, 21),
            style=0, value="")

    self.staticText3
wx.StaticText(id=wxID_FRAME1STATICTEXT3,

```

```

        label='(Tekan      ENTER)',      name='staticText3',
parent=self.panel1,
pos=wx.Point(224,   32),   size=wx.Size(82,   13),
style=0)

    self.staticText3.SetBackgroundColour(wx.Colour(255,
255, 255))

    self.staticText3.SetForegroundColour(wx.Colour(255,
0, 0))

    self.staticText3SetFont(wx.Font(8,           wx.SWISS,
wx.NORMAL, wx.BOLD, False,
'Tahoma'))


self.tmbHapus = wx.Button(id=wxID_FRAME1TMBHAPUS, label='Hapus',
name='tmbHapus',           parent=self.panel1,
pos=wx.Point(48, 136),
size=wx.Size(75, 23), style=0)

self.tmbHapus.Bind(wx.EVT_BUTTON,
self.OnTmbHapusButton,
id=wxID_FRAME1TMBHAPUS)

def __init__(self, parent):
    self._init_ctrls(parent)

def OnTmbHapusButton(self, event):

```

```

if self.txt_nim.GetValue() == "" :
    self.pesan = wx.MessageDialog \
    (self,"NIM belum diisi","Konfirmasi",wx.OK)
    self.pesan.ShowModal()
    event.Skip()

sql = " select * from mhs where nim \
= '%s'" % (self.txt_nim.GetValue())
cur.execute(sql)

hasil= cur.fetchone()

if cur.rowcount>0 :
    tanya = wx.MessageDialog(self,\n
    message="Anda Yakin Hendak Menghapus\n
NIM'\\"+
    +" "+self.txt_nim.GetValue()+" "+\
    "Nama "+self.txt_nama.GetValue()\
    ,style = wx.YES_NO)
    if tanya.ShowModal()==wx.ID_YES:
        sql = "delete from mhs where nim \
= '%s'" % (self.txt_nim.GetValue())
        cur.execute(sql)

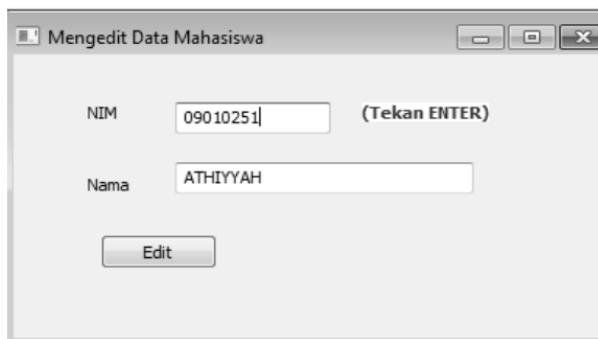
else :

```

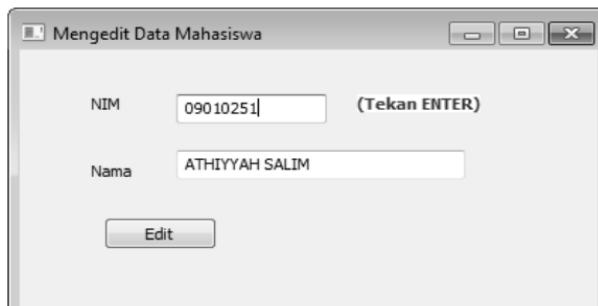
```
self.pesan = wx.MessageDialog(self,\n    "NIM yang akan dihapus tidak terdata\\\n    di database","Konfirmasi",wx.OK)\n\n    self.pesan.ShowModal()\n\n    self.txt_nim.SetValue("")\n\n    self.txt_nama.SetValue("")\n\n    self.txt_nim.SetFocus()\n\n\ndef OnTxt_nimTextEnter(self, event):\n\n    sql = " select * from mhs where nim \\n\n        = '%s'" %(self.txt_nim.GetValue())\n\n    cur.execute(sql)\n\n    hasil= cur.fetchone()\n\n    if cur.rowcount > 0 :\n\n        self.txt_nama.SetValue(hasil[1])\n\n    else :\n\n        self.pesan = wx.MessageDialog(self,\n            "Data Tidak Ada","Konfirmasi",wx.OK)\n\n        self.pesan.ShowModal()
```

8.5 Edit/ Update Data

Aplikasi terakhir yang berkaitan dengan akses ke database adalah edit/ update data. User memasukkan NIM yang akan diedit, kemudian sistem akan menampilkan nama yang sesuai dengan NIM tersebut. Selanjutnya, nama tersebut diedit dan perubahannya disimpan dalam database dengan menekan tombol Edit.



Gambar 8.22 Nama Athiyyah akan Diedit



Gambar 8.23 Nama Berubah Menjadi Athiyyah Salim

Perbedaan utama pada masing-masing akses ke database adalah pada perintah SQL-nya.

KODE TOMBOL EDIT

```
def OnTmbEditButton(self, event):  
  
    sql = "update mhs set nama ='%s' where nim= \  
        '%s'"%(self.txt_nama.GetValue(), \  
        self.txt_nim.GetValue())  
  
    cur.execute(sql)  
  
    conn.commit()  
  
    self.txt_nim.SetValue("")  
  
    self.txt_nama.SetValue("")  
  
    self.txt_nim.SetFocus()
```

Perintah SQL pada Edit data adalah :

```
sql = "update mhs set nama ='%s' where nim= \  
        '%s'"%(self.txt_nama.GetValue(), \  
        self.txt_nim.GetValue())
```

artinya, mengedit di tabel mahasiswa untuk record/baris yang memiliki nama sama dengan yang diketikkan di **txt_nama** untuk nim yang sama dengan di **txt_nim**. Karena tipe datanya adalah String/ Varchar maka formatnya adalah '%s'.

LISTING KODE LENGKAP (FrEdit.py)

```
#Boa:Frame:Frame1
```

```
import wx, MySQLdb

conn=MySQLdb.connect(host="localhost",user="root",passwd="",db="Akademik")

cur = conn.cursor()

def create(parent):
    return Frame1(parent)

[wxID_FRAME1,
 wxID_FRAME1STATICTEXT1,
 wxID_FRAME1STATICTEXT2,
 wxID_FRAME1STATICTEXT3, wxID_FRAME1TMBEDIT,
 wxID_FRAME1TXT_NAMA, wxID_FRAME1TXT_NIM,
] = [wx.NewId() for _init_ctrls in range(8)]

class Frame1(wx.Frame):
    def __init__(self, prnt):
        # generated method, don't edit
        wx.Frame.__init__(self, id=wxID_FRAME1, name='',
parent=prnt,
pos=wx.Point(387, 245), size=wx.Size(400, 227),
```

```

        style=wx.DEFAULT_FRAME_STYLE,
title='Mengedit Data Mahasiswa')

self.SetClientSize(wx.Size(384, 189))

self.panel1 = wx.Panel(id=wxID_FRAME1PANEL1,
name='panel1', parent=self,
pos=wx.Point(0, 0), size=wx.Size(384, 189),
style=wx.TAB_TRAVERSAL)

self.staticText1 =
wx.StaticText(id=wxID_FRAME1STATICTEXT1,
label='NIM',
name='staticText1', parent=self.panel1,
pos=wx.Point(48, 32),
size=wx.Size(20, 13), style=0)

self.txt_nim =
wx.TextCtrl(id=wxID_FRAME1TXT_NIM, name='txt_nim',
parent=self.panel1, pos=wx.Point(104, 32),
size=wx.Size(100, 21),
style=wx.TE_PROCESS_ENTER, value="")

self.txt_nim.Bind(wx.EVT_TEXT_ENTER,
self.OnTxt_nimTextEnter,
id=wxID_FRAME1TXT_NIM)

```

```
    self.staticText2 =  
wx.StaticText(id=wxID_FRAME1STATICTEXT2,  
  
                 label='Nama',           name='staticText2',  
                 parent=self.panel1,  
  
                 pos=wx.Point(48,    80),   size=wx.Size(28,    13),  
                 style=0)  
  
  
    self.txt_nama =  
wx.TextCtrl(id=wxID_FRAME1TXT_NAMA,  
            name='txt_nama',  
  
            parent=self.panel1,      pos=wx.Point(104,    72),  
            size=wx.Size(192, 21),  
  
            style=0, value="")  
  
  
    self.staticText3 =  
wx.StaticText(id=wxID_FRAME1STATICTEXT3,  
  
                 label='(Tekan     ENTER)',   name='staticText3',  
                 parent=self.panel1,  
  
                 pos=wx.Point(224,    32),   size=wx.Size(82,    13),  
                 style=0)  
  
    self.staticText3.SetBackgroundColour(wx.Colour(255,  
255, 255))  
  
    self.staticText3.SetForegroundColour(wx.Colour(255,  
0, 0))  
  
    self.staticText3SetFont(wx.Font(8,           wx.SWISS,  
wx.NORMAL, wx.BOLD, False,  
  
                           'Tahoma'))
```

```

        self.tmbEdit = wx.Button(id=wxID_FRAME1TMBEDIT, label='Edit',
wx.Point(56, 120),
name='tmbEdit', parent=self.panel1,
pos=,
size=wx.Size(75, 23), style=0)

        self.tmbEdit.Bind(wx.EVT_BUTTON,
self.OnTmbEditButton,
id=wxID_FRAME1TMBEDIT)

def __init__(self, parent):
    self._init_ctrls(parent)

def OnTxt_nimTextEnter(self, event):

    sql = " select * from mhs where nim \
= '%s'" %(self.txt_nim.GetValue())
    cur.execute(sql)
    hasil= cur.fetchone()
    if cur.rowcount > 0 :
        self.txt_nama.SetValue(hasil[1])
    else :
        self.pesan = wx.MessageDialog(self,\
```

```
"Data Tidak Ada","Konfirmasi",wx.OK)

self.pesan.ShowModal()

def OnTmbEditButton(self, event):

    sql = "update mhs set nama ='%s' where nim=\
        '%s'"%(self.txt_nama.GetValue(),\
        self.txt_nim.GetValue())

    cur.execute(sql)

    conn.commit()

    self.txt_nim.SetValue("")

    self.txt_nama.SetValue("")

    self.txt_nim.SetFocus()
```

8.6 Aplikasi Gabungan (SIMPAN, EDIT, HAPUS, CARI)

Pada contoh sebelumnya, tiap proses ke database adalah Frame yang terpisah. Berikut ini kita akan gabung semua proses itu dalam 1 Frame.

Proses juga dimodifikasi sehingga Tombol Simpan bisa digunakan untuk 2 proses sekaligus, yaitu Simpan/Tambah (menyisipkan record/baris baru) dan Edit. Tergantung keadaan, jika NIM yang diketikkan belum ada di database, proses yang dilakukan adalah Simpan. Tapi jika sudah ada, maka prosesnya adalah Edit. Ada juga tambahan tombol

Bersih, untuk membersihkan semua textctrl dan memfocuskan kursor mouse ke textctrl **txt_nim**.

Karena proses **Bersih** tersebut diperlukan di akhir semua proses (jika anda menyimpan data baru, pasti selanjutnya akan membersihkan seluruh textctrl, demikian juga ketika menghapus data), maka akan ditunjukkan cara membuat dan memanggil suatu method buatan sendiri, yaitu method **Bersih**. Dengan demikian, untuk menjalankan fungsi membersihkan seluruh textctrl dan memfocuskan kursor mouse ke **txt_nim** cukup dengan memanggil satu baris kode: **self.Bersih()**. Hal ini akan sangat efektif untuk fungsi yang sering digunakan.



Gambar 8.24 Input Data Mahasiswa Lengkap (Simpan, Bersih, Hapus)

MEMBUAT METHOD SENDIRI

Pada aplikasi ini kita membuat method sendiri dengan nama **Bersih**. Kode pembuatan method tersebut adalah:

```
def Bersih(self) :  
    self.txt_nim.SetValue("")  
    self.txt_nama.SetValue("")  
    self.txt_nim.SetFocus()
```

untuk memanggil method tersebut, kodenya:

```
self.Bersih()
```

KODE TOMBOL SIMPAN

Tombol Simpan pada aplikasi ini berfungsi dua hal sekaligus, yaitu Edit dan Tambah (penyisipan 1 record baru). Jika NIM yang akan disimpan sudah ada dalam database, maka yang berfungsi adalah Edit. Akan dilakukan edit nama berdasarkan NIM tersebut. Namun, jika di database belum ada NIM tersebut, maka akan disimpan 1 record baru dengan NIM dan nama itu.

Kode pada tombol Simpan adalah sebagai berikut:

```
def OnTmSimpanButton(self, event):  
    sql = " select * from mhs where nim \\\n        = '%s'" %(self.txt_nim.GetValue())  
    cur.execute(sql)  
    hasil= cur.fetchone()  
    if cur.rowcount > 0 :  
        sql =" update mhs set nama = \\\n            "
```

```

'%s' where nim ='%s'"%\\
(self.txt_nama.GetValue(), \\
self.txt_nim.GetValue())

else :

sql ="insert into mhs (nim,nama) \\
values ('%s','%s')"%{self.txt_nim.GetValue(), \\
self.txt_nama.GetValue()}

cur.execute(sql)

self.Bersih()

```

**LISTING KODE KESELURUHAN
(FrGabungan.py)**

```

#Boa:Frame:Frame1

import wx, MySQLdb

conn=MySQLdb.connect(host="localhost",user="root",p
asswd="",db="Akademik")

cur = conn.cursor()

def create(parent):
    return Frame1(parent)

```

```

[wxID_FRAME1, wxID_FRAME1PANEL1,
wxID_FRAME1STATICTEXT1, wxID_FRAME1STATICTEXT2,
wxID_FRAME1STATICTEXT3, wxID_FRAME1TMBBERSIH,
wxID_FRAME1TMBHAPUS, wxID_FRAME1TMSIMPAN,
wxID_FRAME1TXT_NAMA,
wxID_FRAME1TXT_NIM,
] = [wx.NewId() for _init_ctrls in range(10)]


class Frame1(wx.Frame):
    def __init__(self, prnt):
        # generated method, don't edit
        wx.Frame.__init__(self, id=wxID_FRAME1, name="",
parent=prnt,
pos=wx.Point(518, 261), size=wx.Size(431, 227),
style=wx.DEFAULT_FRAME_STYLE, title='Input
Data Mahasiswa')

        self.SetClientSize(wx.Size(415, 189))

        self.panel1 = wx.Panel(id=wxID_FRAME1PANEL1,
name='panel1', parent=self,
pos=wx.Point(0, 0), size=wx.Size(415, 189),
style=wx.TAB_TRAVERSAL)

```

```

        self.staticText1 =  

wx.StaticText(id=wxID_FRAME1STATICTEXT1,  

label='NIM',  

        name='staticText1', parent=self.panel1,  

pos=wx.Point(48, 32),  

        size=wx.Size(20, 13), style=0)  

        self.txt_nim =  

wx.TextCtrl(id=wxID_FRAME1TXT_NIM, name='txt_nim',  

        parent=self.panel1, pos=wx.Point(104, 32),  

size=wx.Size(100, 21),  

        style=wx.TE_PROCESS_ENTER, value='')  

        self.txt_nim.Bind(wx.EVT_TEXT_ENTER,  

self.OnTxt_nimTextEnter,  

        id=wxID_FRAME1TXT_NIM)  

        self.staticText2 =  

wx.StaticText(id=wxID_FRAME1STATICTEXT2,  

        label='Nama', name='staticText2',  

parent=self.panel1,  

        pos=wx.Point(48, 80), size=wx.Size(28, 13),  

style=0)

```

```

    self.txt_nama
wx.TextCtrl(id=wxID_FRAME1TXT_NAMA,
name='txt_nama',
parent=self.panel1,      pos=wx.Point(104,      72),
size=wx.Size(192, 21),
style=0, value=")

self.staticText3
wx.StaticText(id=wxID_FRAME1STATICTEXT3,
label='(Tekan      ENTER)',      name='staticText3',
parent=self.panel1,
pos=wx.Point(224,  32),  size=wx.Size(82,  13),
style=0)

self.staticText3.SetBackgroundColour(wx.Colour(255,
255, 255))

self.staticText3.SetForegroundColour(wx.Colour(255,
0, 0))

self.staticText3SetFont(wx.Font(8,           wx.SWISS,
wx.NORMAL, wx.BOLD, False,
'Tahoma'))


self.tmbHapus
wx.Button(id=wxID_FRAME1TMBHAPUS, label='Hapus',
name='tmbHapus',          parent=self.panel1,
pos=wx.Point(248, 128),
size=wx.Size(75, 23), style=0)

```

```

    self.tmbHapus.Bind(wx.EVT_BUTTON,
self.OnTmbHapusButton,
        id=wxID_FRAME1TMBHAPUS)

    self.tmSimpan =
wx.Button(id=wxID_FRAME1TMSIMPAN, label='Simpan',
        name='tmSimpan',           parent=self.panel1,
pos=wx.Point(48, 128),
        size=wx.Size(75, 23), style=0)

    self.tmSimpan.Bind(wx.EVT_BUTTON,
self.OnTmSimpanButton,
        id=wxID_FRAME1TMSIMPAN)

    self.tmbBersih =
wx.Button(id=wxID_FRAME1TMBBERSIH, label='Bersih',
        name='tmbBersih',           parent=self.panel1,
pos=wx.Point(152, 128),
        size=wx.Size(75, 23), style=0)

    self.tmbBersih.Bind(wx.EVT_BUTTON,
self.OnTmbBersihButton,
        id=wxID_FRAME1TMBBERSIH)

def __init__(self, parent):
    self._init_ctrls(parent)

```

```

def Bersih(self) :
    self.txt_nim.SetValue("")
    self.txt_nama.SetValue("")
    self.txt_nim.SetFocus()

def OnTmbHapusButton(self, event):
    if self.txt_nim.GetValue() == "":
        self.pesan = wx.MessageDialog\
        (self, "NIM belum diisi", "Konfirmasi", wx.OK)
        self.pesan.ShowModal()
        event.Skip()

    sql = " select * from mhs where nim \
    = '%s' % (self.txt_nim.GetValue())"
    cur.execute(sql)
    hasil= cur.fetchone()
    if cur.rowcount>0 :
        tanya = wx.MessageDialog(self,\n
        message="Anda Yakin Hendak Menghapus\n
NIM"\n
        +" "+self.txt_nim.GetValue()+" "+\n
        "Nama "+self.txt_nama.GetValue()\

```

```

,style = wx.YES_NO)

if tanya.ShowModal()==wx.ID_YES:

    sql = "delete from mhs where nim \
    = '%s'" %(self.txt_nim.GetValue())

    cur.execute(sql)

else :

    self.pesan = wx.MessageDialog(self,\n
    "NIM yang akan dihapus tidak terdata\
    di database","Konfirmasi",wx.OK)

    self.pesan.ShowModal()

    self.txt_nim.SetValue("")

    self.txt_nama.SetValue("")

    self.txt_nim.SetFocus()

def OnTxt_nimTextEnter(self, event):

    sql = " select * from mhs where nim \
    = '%s'" %(self.txt_nim.GetValue())

    cur.execute(sql)

    hasil= cur.fetchone()

    if cur.rowcount > 0 :

        self.txt_nama.SetValue(hasil[1])

```

```

else :

    self.pesan = wx.MessageDialog(self,\n
        "Data Tidak Ada","Konfirmasi",wx.OK)
    self.pesan.ShowModal()

def OnTmSimpanButton(self, event):
    sql = " select * from mhs where nim \
        = '%s'" %(self.txt_nim.GetValue())
    cur.execute(sql)
    hasil= cur.fetchone()
    if cur.rowcount > 0 :
        sql =" update mhs set nama =\
        '%s' where nim ='%s'%\n
        (%self.txt_nama.GetValue(),\
        self.txt_nim.GetValue())
    else :
        sql ="insert into mhs (nim,nama) \
        values ('%s','%s')"% (self.txt_nim.GetValue(),\
        self.txt_nama.GetValue())
    cur.execute(sql)
    self.Bersih()

```

```
def OnTmbBersihButton(self, event):  
    self.Bersih()
```

FORMAT TIPE DATA

Sebagai pengenalan, format tipe data yang digunakan hanyalah String/ Varchar yang menggunakan format '%s'. Format tipe data lain adalah Integer menggunakan format '%d'. Format tipe data date/ tanggal menggunakan format '%s' sama dengan pada String. Demikian juga tipe data float menggunakan format '%s'. Contoh-contoh penggunaannya akan dijelaskan InsyaAllah pada bab X tentang Akses Database Lanjutan.

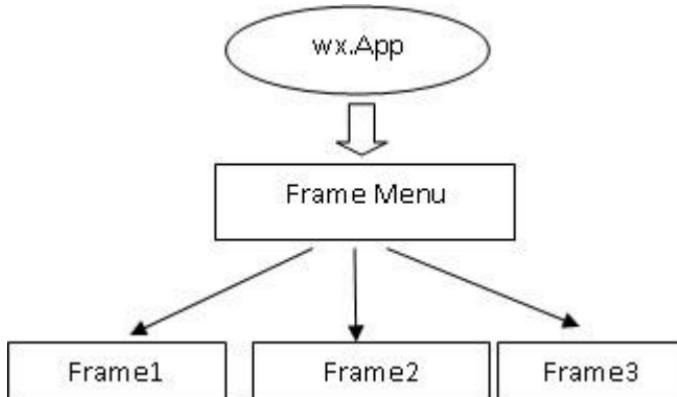
BAB IX

PEMBUATAN MENU

9.1 PENDAHULUAN

Menu pada dasarnya adalah Frame yang mengorganisasikan Frame yang lain. Seperti sudah dibahas pada akhir Bab III bahwa setiap aplikasi minimal terdiri dari 1 wx.App dan 1 wx.Frame. Pada contoh-contohnya, semua aplikasi terdiri dari 2 komponen saja: 1 wx.App dan 1 wx.Frame.

Jika kita menggunakan 1 Frame sebagai menu, maka Frame tersebut yang nantinya bertugas memanggil Frame yang lain.



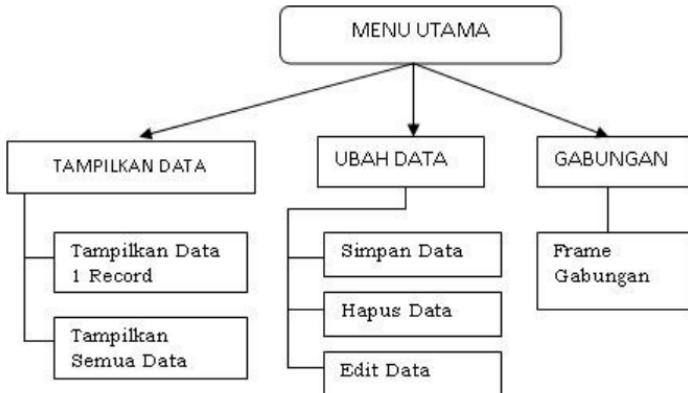
Gambar 9.1 Pengorganisasian Menu

Kita akan menggunakan contoh-contoh Frame yang sudah dibuat pada BAB VIII untuk menunjukkan cara kerja menu. Pada Bab VIII sebelumnya, Frame yang sudah dibuat adalah :

1. FrCariData.py : mencari data 1 record
2. FrLihatData.py : menampilkan seluruh data
3. FrSimpan.py : menyimpan data
4. FrHapus.py : menghapus data
5. FrEdit.py : mengedit data
6. FrGabungan.py : frame gabungan

Menu Utama yang kita buat diberi nama **FrMenu_Utama.py**.

Rancangan menu yang akan kita buat adalah : Menu Utama terbagi dalam 3 submenu, yaitu : **Tampilkan Data**, **Ubah Data**, dan **Gabungan**. Tampilkan Data terdiri dari **Tampil Data 1 Record** dan **Tampil Data Semua**. Ubah Data terdiri dari **Simpan Data**, **Edit Data**, dan **Hapus Data**. Gabungan hanya terdiri dari 1 pilihan, yaitu **Frame Gabungan**.

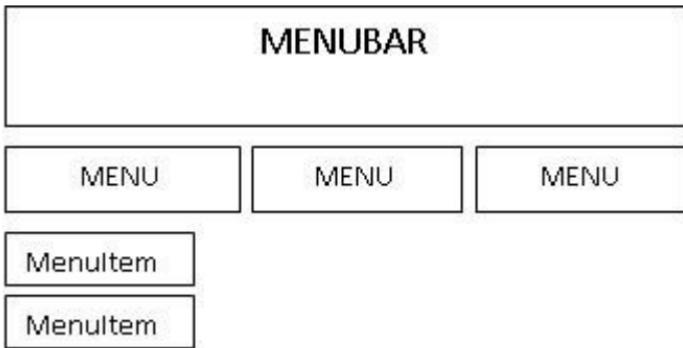


Gambar 9.2 Hierarki Menu yang Dirancang

Pada setiap pembuatan menu, pasti akan menggunakan 3 komponen utama, yaitu:

1. MenuBar
2. Menu
3. MenuItem

Contoh menu yang akan kita buat akan terdiri dari 1 MenuBar, 3 Menu (**TAMPILKAN DATA**, **UBAH DATA**, dan **GABUNGAN**). Menu **TAMPILKAN DATA** terdiri dari 2 MenuItem, yaitu: **Tampilkan Data 1 Record** dan **Tampilkan Semua Data**.



Gambar 9.3 Hierarki Menu Secara Umum

9.2 MEMBUAT MENU DENGAN FASILITAS BOA CONSTRUCTOR (*DRAG AND DROP*)

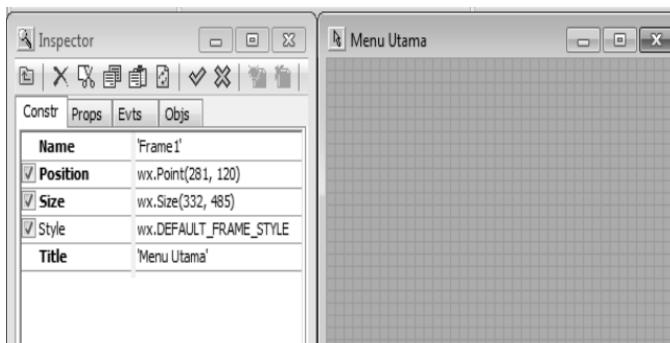
LANGKAH-LANGKAH MEMBUAT MENU:

1. Buatlah wx.App baru, sehingga menghasilkan juga 1 wx.Frame. Frame tersebut yang akan kita jadikan Menu Utama. Simpan wx.Frame tersebut menjadi **FrMenu_Utama.py**. Kemudian, copykan juga semua Frame yang sudah dibuat pada Bab VIII ke folder yang sama. File-file Frame yang sudah dibuat:

- FrCariData.py
- FrEdit.py
- FrGabungan.py
- FrHapus.py
- FrLihatData.py

- FrSimpan.py

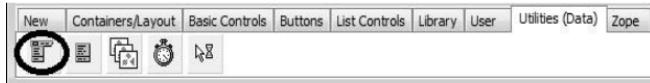
2. Ubah Title Frame menjadi ‘Menu Utama’.



Gambar 9.4 Title: Menu Utama

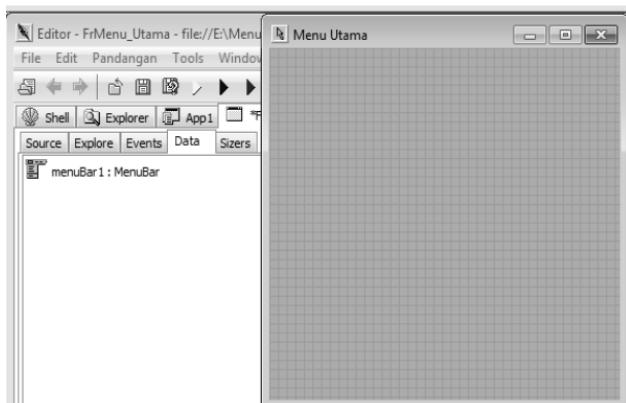
3. Letakkan Object wx.MenuBar ke dalam Frame. Object wx.MenuBar adalah Object yang tidak terlihat secara langsung setelah diletakkan pada Frame.

Letak Object wx.MenuBar adalah di bagian paling kiri tab **Utilites(Data)** pada **Zope Editor**.



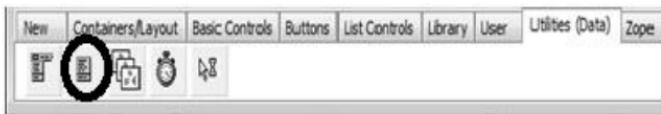
Gambar 9.5 Letak wx.MenuBar

Setelah diletakkan pada Frame, akan terlihat di tab Data (Editor) bahwa MenuBar telah masuk ke dalamnya



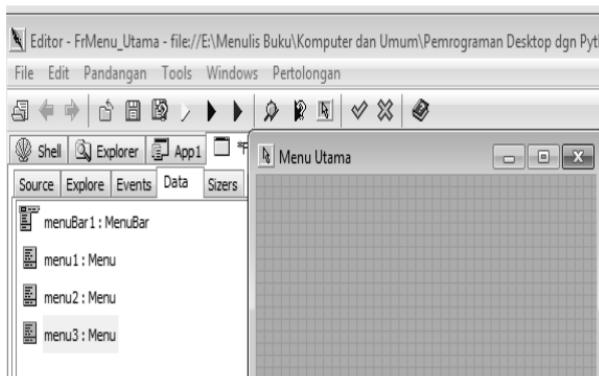
Gambar 9.6 menuBar1 telah diletakkan pada Frame

4. Letakkan 3 buah wx.Menu ke dalam Frame.



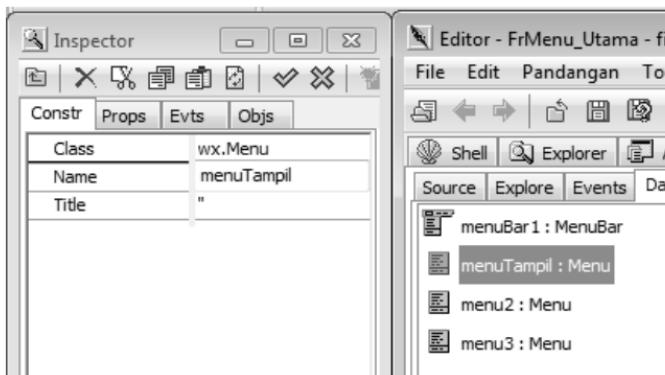
Gambar 9.7 Letak wx.Menu pada Zope Editor (Tab Utilites)

Setelah ketiga wx.Menu itu diletakkan akan terlihat di tab **Data (Editor)**



Gambar 9.8 Tiga wx.Menu Telah Diletakkan pada Frame

5. Ubahlah name pada menu1 menjadi **menuTampil**



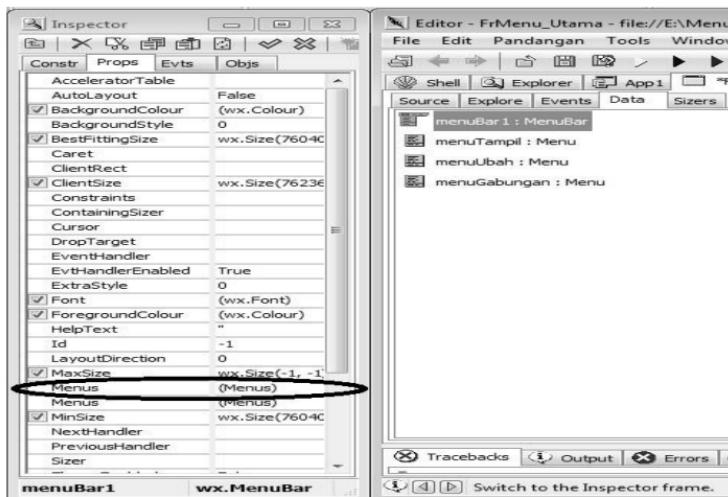
Gambar 9.9 menu1 diubah menjadi menuTampil

Ubahlah juga name menu2 menjadi **menuUbah** dan menu3 menjadi **menuGabungan**. Jangan lupa menekan icon Post pada Inspector dan Editor.



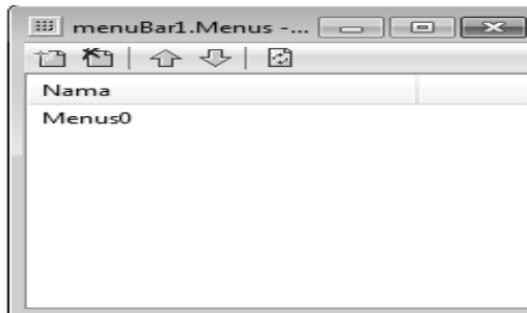
Gambar 9.10 Name ketiga Menu Telah Diubah

- Buatlah judul tulisan padaMenuBar menjadi : **TAMPILKAN DATA, UBAH DATA, dan GABUNGAN.** Caranya, aktifkan menuBar1, kemudian setting Inspector pada tab **Props**, pilih bagian Menus



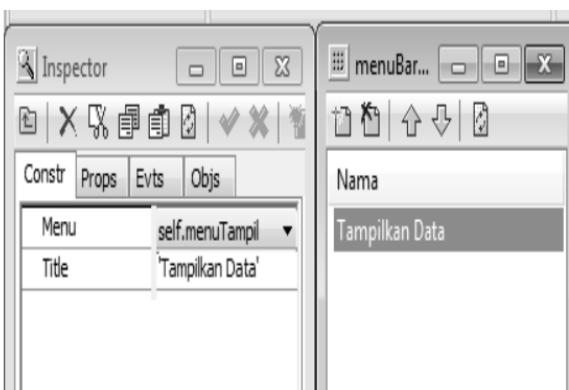
Gambar 9.11 Menus pada Inspector

Klik **Menus**, sehingga muncul tombol kecil tiga titik (...), kemudian klik tombol tersebut untuk menampilkan jendel **menuBar1.Menus**



Gambar 9.12 menuBar1.Menus

Gantilah tulisan Menus0 title-nya menjadi **Tampilkan Data**, dan Menu menjadi **self.menuTampil**

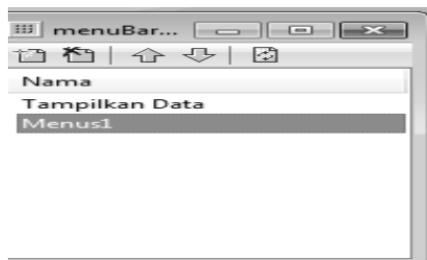


Gambar 9.13 Setting Menu dan Title ‘Tampilkan Data’

Selanjutnya, tambahkan tulisan baru dengan mengklik icon ‘Tambah’, sehingga muncul tulisan baru:**Menus1**

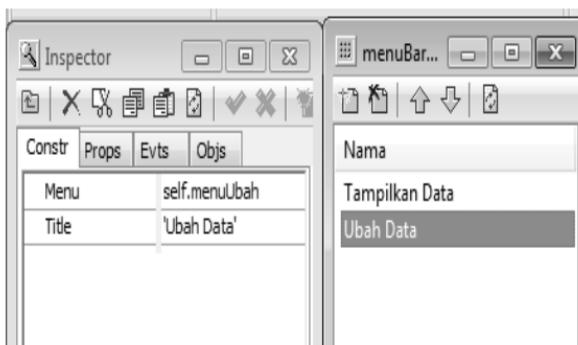


Gambar 9.14 Icon Tambah Tulisan



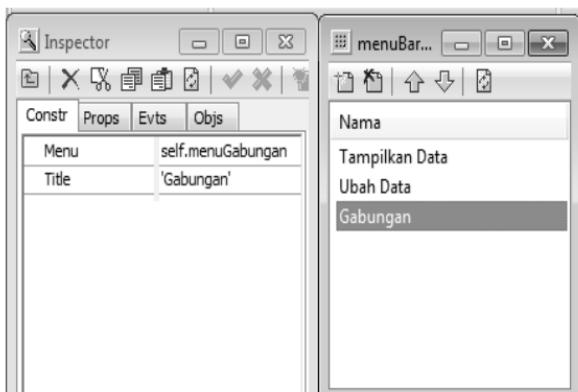
Gambar 9.15 Tulisan Baru Menus1

Gantilah Setting Inspector untuk Menus1: Title menjadi **Ubah Data** dan Menu menjadi **self.menuUbah**



Gambar 9.16 Setting Inspector untuk ‘Ubah Data’

Buatlah tulisan baru, ganti title-nya menjadi **Gabungan**, dan Menu = **self.menuGabungan**

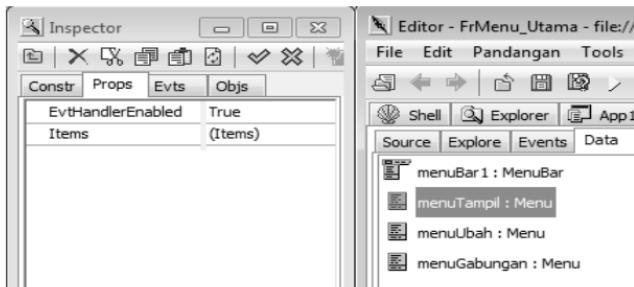


Gambar 9.17 Tulisan Menu ‘Gabungan’ Dibuat

Jika perubahan tulisan tidak terlihat, klik tombol Refresh dan tombol Post.

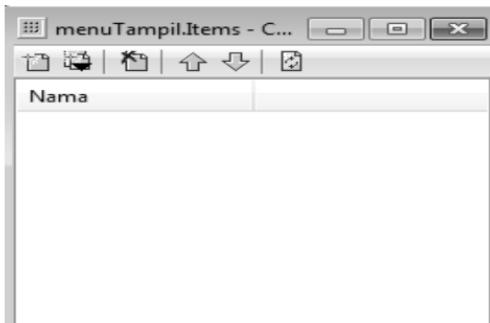
7. Membuat Menulitem pada setiap Menu

Aktifkan menu **Tampil** dan setting Inspectornya di tab **Props** pada bagian **Items**



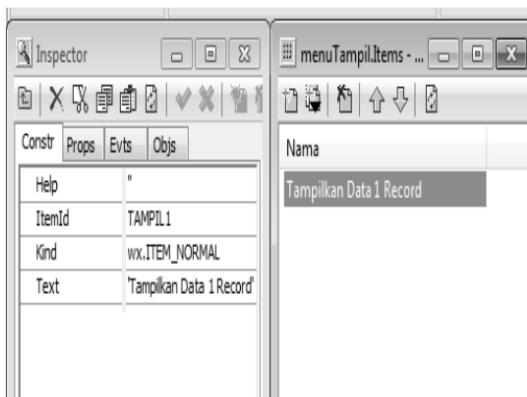
Gambar 9.18 Setting Items Inspector menuTampil

Klik pada bagian **Items** sehingga muncul tombol bertitik tiga (...), kemudian klik tombol kecil tersebut sehingga muncul jendela **menuTampil.Items**



Gambar 9.19 Jendela menuTampil.Items

Tambahkan tulisan baru, dengan mengklik tombol Tambah, ganti **Textnya** menjadi **Tampilkan Data 1 Record**, dan **ItemId** menjadi **TAMPIL1**

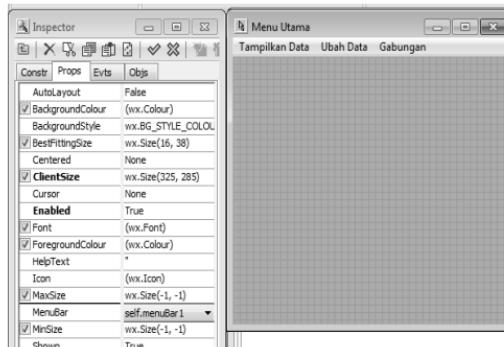


Gambar 9.20 MenuItem Tampilkan Data 1 Record

Buatlah yang semisal dengan itu untuk Menu yang lain, dengan ketentuan sebagai berikut:

Menu	Setting	Nilai
Tampilkan Data	ItemId	Tampil1
	Text	Tampilkan Data 1 Record
	ItemId	Tampil_Semua
	Text	Tampilkan Semua Data
Ubah Data	ItemId	Simpan
	Text	Simpan Data
	ItemId	Edit
	Text	Edit Data
	ItemId	Hapus
	Text	Hapus Data
Gabungan	ItemId	Gabungan
	Text	Frame Gabungan

8. Aturlah Inspector untuk Frame (tab Props) agarMenuBar-nya adalah **self.menuBar1**.

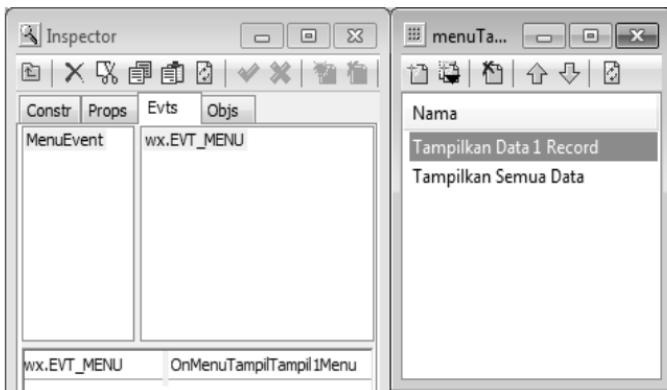


Gambar 9.21 Setting MenuBar pada Frame

9. Buat Event pada setiap MenuItem

Pada tab Evt : MenuEvent >wx.EVT_MENU > klik dua kali

Contoh : pada MenuItem ‘Tampilkan Data 1 Record’, klik **MenuEvent > wx.EVT_MENU (klik 2 kali)> OnMenuTampilTampil1Menu**



Gambar 9.22 Event pada ‘Tampilkan Data 1 Record’

Jangan lupa klik icon Post agar tampil perubahan pada kode program (Source).

Akibat pembuatan events tersebut, akan dihasilkan method baru. Contoh : Method baru akibat pembuatan event pada MenuItem ‘Tampilkan Data 1 Record’ adalah :

```
def OnMenuTampilTampil1Menu(self, event):
    event.Skip()
```

Ubahlah kode pada method baru tersebut untuk memanggil Frame yang diinginkan. Misalnya, kode untuk memanggil Frame ‘Tampil Data 1 Record’ (**FrCariData.py**) adalah :

```
def OnMenuTampilTampil1Menu(self, event):
    self.main = FrCariData.create(None)
    self.main.Show()
```

Lakukan hal yang sama pada MenuItem yang lain.

Kode untuk memanggil tiap Frame sesuai Event MenuItem yang telah dibuat :

```
def OnMenuTampilTampil1Menu(self, event):
    self.main = FrCariData.create(None)
    self.main.Show()

def OnMenuTampilTampil_semuaMenu(self, event):
    self.main = FrLihatData.create(None)
    self.main.Show()

def OnMenuUbahSimpanMenu(self, event):
    self.main = FrSimpan.create(None)
    self.main.Show()

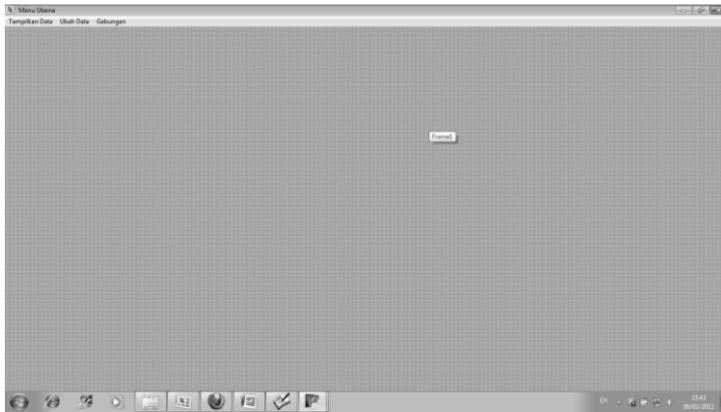
def OnMenuUbahEditMenu(self, event):
    self.main = FrEdit.create(None)
    self.main.Show()
```

```
def OnMenuUbahHapusMenu(self, event):  
    self.main = FrHapus.create(None)  
    self.main.Show()  
  
def OnMenuGabunganGabunganMenu(self, event):  
    self.main = FrGabungan.create(None)  
    self.main.Show()
```

10. Import semua Frame yang akan dipanggil menu di bagian paling atas kode (digabungkan dengan **import wx**), sehingga kodennya menjadi:

```
import wx, FrEdit, FrGabungan, FrHapus, FrSimpan,  
FrLihatData,FrCariData
```

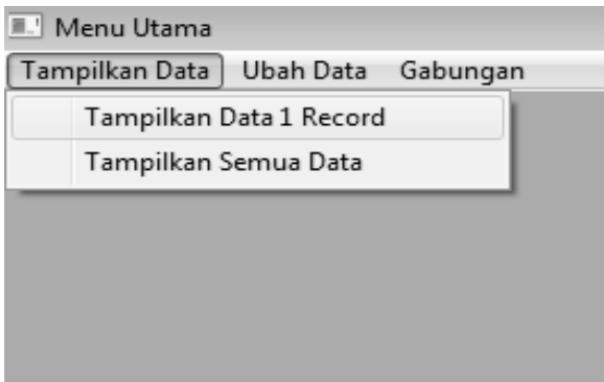
11. Aturlah Frame Menu Utama agar ukuran desainnya maksimal



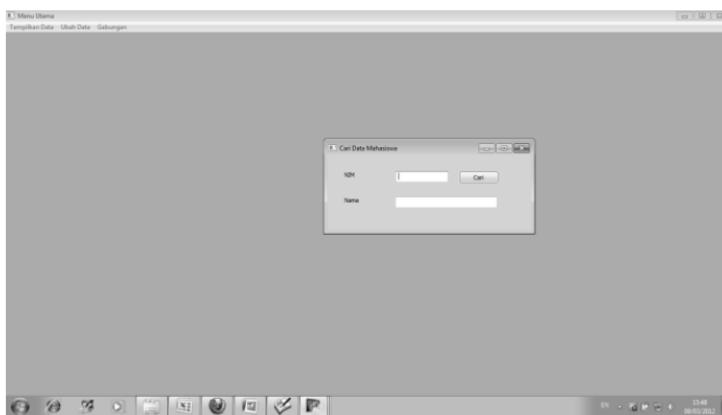
Gambar 9.23 Frame Menu Utama diset Max

Jangan lupa klik icon Post.

Jalankan Frame Menu Utama tersebut dan lihatlah hasilnya. Pembuatan menu sudah selesai kita lakukan.



Gambar 9.24 Menu Item Tampilkan Data 1 Record



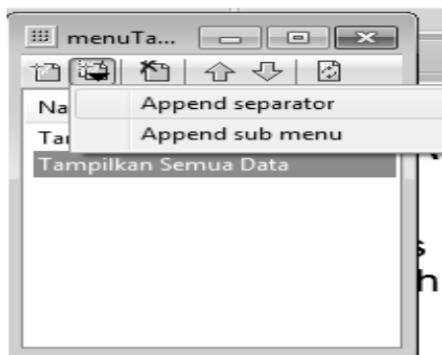
**Gambar 9.25 Frame yang Dipanggil oleh Menu Item
Tampilkan Data 1 Record (FrCariData.py)**

MEMBUAT SEPARATOR (GARIS PEMISAH ANTAR MENUITEM)

Untuk membuat separator (garis pemisah) antar menuitem, klik tombol icon ‘Tambah Separator dan SubMenu’ (icon nomor 2 dari kiri)

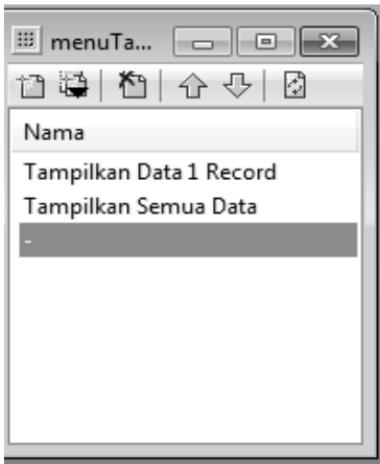


Gambar 9.26 Icon ‘Tambah Separator dan SubMenu’



Gambar 9.27 ‘Append Separator’ setelah Menu ‘Tampilkan Semua Data’

Tanda pemisah (-) akan ditunjukkan sebagai separator

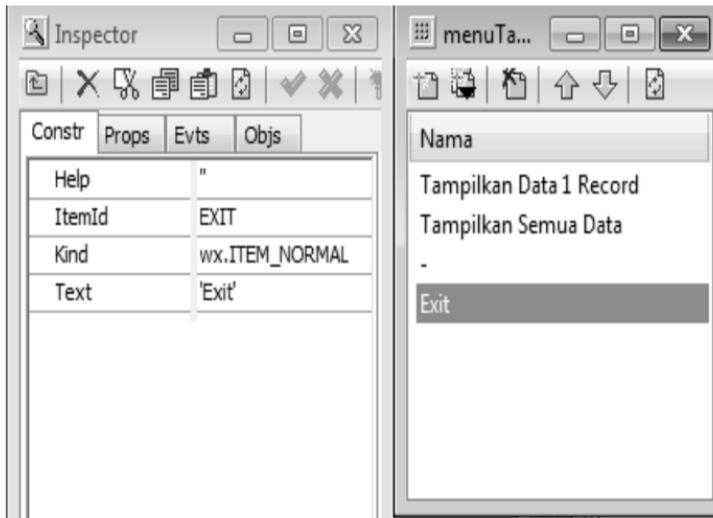


Gambar 9.28 Tanda Separator Ditambahkan

MENAMBAHKAN MENUITEM ‘EXIT’ (KELUAR)

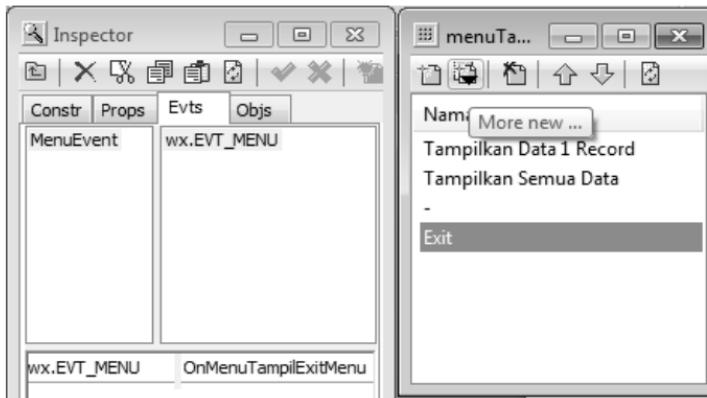
Kita akan menambahkan menuitem ‘Exit’ (keluar) setelah separator yang tadi dibuat. Jika menu tersebut dipilih, akan menyebabkan keluar dari aplikasi.

Tambahkan Menulitem Exit di bagian paling bawah pada Menu ‘Tampilkan Data’.



Gambar 9.29 Menambahkan MenuItem 'Exit'

Selanjutnya, buatlah Event pada menulitem Exit sehingga menghasilkan **OnMenuTampilExitMenu**.



**Gambar 9.30 Pembuatan Event
OnMenuTampilExitMenu**

Kode yang otomatis dihasilkan setelah pembuatan event :

```
def OnMenuTampilExitMenu(self, event):
    event.Skip()
```

kode *event.Skip()* dihapus dan diganti dengan :

```
self.Close()
```

LISTING LENGKAP KODE (FrMenu_Utama.py)

```
#Boa:Frame:Frame1

import wx, FrEdit, FrGabungan, FrHapus, FrSimpan,
FrLihatData, FrCariData

def create(parent):
    return Frame1(parent)

[wxID_FRAME1] = [wx.NewId() for _init_ctrls in range(1)]


[wxID_FRAME1MENUTAMPILEXIT,
wxID_FRAME1MENUTAMPILTAMPIL1,
wxID_FRAME1MENUTAMPILTAMPIL_SEMUA,
] = [wx.NewId() for _init_coll_menuTampil_Items in
range(3)]
```

```

[wxID_FRAME1MENUUABAHEDIT,
wxID_FRAME1MENUUABAHHAPUS,
wxID_FRAME1MENUUABAHSIMPAN,

] = [wx.NewId() for _init_coll_menuUbah_Items in range(3)]



[wxID_FRAME1MENUGABUNGANGABUNGAN]      =
[wx.NewId()   for   _init_coll_menuGabungan_Items   in
range(1)]


class Frame1(wx.Frame):

    def _init_coll_menuUbah_Items(self, parent):
        # generated method, don't edit

        parent.Append(help="",
id=wxID_FRAME1MENUUABAHSIMPAN,
        kind=wx.ITEM_NORMAL, text='Simpan Data')

        parent.Append(help="",
id=wxID_FRAME1MENUUABAHEDIT,
kind=wx.ITEM_NORMAL,
        text='Edit Data')

        parent.Append(help="",
id=wxID_FRAME1MENUUABAHHAPUS,
kind=wx.ITEM_NORMAL,
        text='Hapus Data')

        self.Bind(wx.EVT_MENU,
self.OnMenuUbahSimpanMenu,

```

```

        id=wxID_FRAME1MENUUBAHSIMPAN)

    self.Bind(wx.EVT_MENU,
self.OnMenuUbahEditMenu,

        id=wxID_FRAME1MENUUBAHEDIT)

    self.Bind(wx.EVT_MENU,
self.OnMenuUbahHapusMenu,

        id=wxID_FRAME1MENUUBAHHAPUS)

def _init_coll_menuTampil_Items(self, parent):
    # generated method, don't edit

        parent.Append(help='',
id=wxID_FRAME1MENUTAMPILTAMPIL1,
            kind=wx.ITEM_NORMAL, text='Tampilkan Data
1 Record')

    parent.Append(help='',
id=wxID_FRAME1MENUTAMPILTAMPIL_SEMUA,
            kind=wx.ITEM_NORMAL,           text='Tampilkan
Semua Data')

    parent.AppendSeparator()

    parent.Append(help='',
id=wxID_FRAME1MENUTAMPILEXIT,
            kind=wx.ITEM_NORMAL, text='Exit')

    self.Bind(wx.EVT_MENU,
self.OnMenuTampilTampil1Menu,

```

```
    id=wxID_FRAME1MENUTAMPILTAMPIL1)

    self.Bind(wx.EVT_MENU,
self.OnMenuTampilTampil_semuaMenu,

id=wxID_FRAME1MENUTAMPILTAMPIL_SEMUA)

    self.Bind(wx.EVT_MENU,
self.OnMenuTampilExitMenu,

    id=wxID_FRAME1MENUTAMPILEXIT)

def _init_coll_menuBar1_Menus(self, parent):
    # generated method, don't edit

    parent.Append(menu=self.menuTampil,
title='Tampilkan Data')

    parent.Append(menu=self.menuUbah,      title='Ubah
Data')

    parent.Append(menu=self.menuGabungan,
title='Gabungan')

def _init_coll_menuGabungan_Items(self, parent):
    # generated method, don't edit

    parent.Append(help='',
id=wxID_FRAME1MENUGABUNGANGABUNGAN,
```

```
        kind=wx.ITEM_NORMAL,           text='Frame
Gabungan')

    self.Bind(wx.EVT_MENU,
self.OnMenuGabunganGabunganMenu,
id=wxID_FRAME1MENUGABUNGANGABUNGAN)

def __init__(self):
    # generated method, don't edit
    self.menuTampil = wx.Menu(title="")

    self.menuUbah = wx.Menu(title="")

    self.menuGabungan = wx.Menu(title="")

    self.menuBar1 = wx.MenuBar()

    self.menuBar1.SetClientSize(wx.Size(77368304,
27135440))

    self.__init__(self.menuTampil)
    self.__init__(self.menuUbah)

    self.__init__(self.menuGabungan)
```

```
self._init_coll_menuBar1_Menus(self.menuBar1)

def _init_ctrls(self, prnt):
    # generated method, don't edit
    wx.Frame.__init__(self, id=wxID_FRAME1, name='',
parent=prnt,
                      pos=wx.Point(-9, 0), size=wx.Size(1382, 744),
                      style=wx.DEFAULT_FRAME_STYLE, title='Menu
Utama')
    self._init_utils()
    self.SetClientSize(wx.Size(1366, 706))
    self.Enable(True)
    self.SetMenuBar(self.menuBar1)

def __init__(self, parent):
    self._init_ctrls(parent)

def OnMenuTampilTampil1Menu(self, event):
    self.main = FrCariData.create(None)
    self.main.Show()

def OnMenuTampilTampil_semuaMenu(self, event):
```

```
self.main = FrLihatData.create(None)
self.main.Show()

def OnMenuUbahSimpanMenu(self, event):
    self.main = FrSimpan.create(None)
    self.main.Show()

def OnMenuUbahEditMenu(self, event):
    self.main = FrEdit.create(None)
    self.main.Show()

def OnMenuUbahHapusMenu(self, event):
    self.main = FrHapus.create(None)
    self.main.Show()

def OnMenuGabunganGabunganMenu(self, event):
    self.main = FrGabungan.create(None)
    self.main.Show()

def OnMenuTampilExitMenu(self, event):
    self.Close()
```

9.3 MEMBUAT MENU DENGAN KODE

Contoh sebelumnya adalah pembuatan menu secara *drag and drop* menggunakan fasilitas Boa Constructor. Berikut ini akan ditunjukkan cara membuat menu dengan kode.

Pembuatan menu dengan kode tersebut menggunakan alur utama:

1. Import semua nama frame yang akan digunakan. Diletakkan di bagian kode paling atas.

```
import wx, FrEdit, FrGabungan, FrHapus,FrSimpan,  
FrLihatData, FrCariData
```

2. Buat wx.Menu()

```
menuTampil_Data = wx.Menu()
```

3. Menambahkan MenuItem pada wx.Menu yang sudah dibuat

Syntax :

```
nama_menu.Append(nomor_akses,tulisan)
```

```
menuTampil_Data.Append(1, "Tampilkan Data 1 Record")
```

```
menuTampil_Data.Append(2, "Tampilkan Semua Data")
```

```
menuTampil_Data.Append(3, "Exit")
```

4. Membuat separator (Optional)

```
menuTampil_Data.AppendSeparator()
```

5. Membuat wx.MenuBar

```
menuBar = wx.MenuBar()
```

6. Menambahkan masing-masing wx.Menu yang sudah dibuat ke dalam wx.MenuBar

Syntax :

```
nama_menuBar.Append(nama_wx.Menu,tulisan_yg_ditampilkan)
```

```
menuBar.Append(menuTampil_Data, "Tampilkan Data")
```

```
menuBar.Append(menuUbah_Data, "Ubah Data")
```

```
menuBar.Append(menuGabungan,"Gabungan")
```

7. Mendaftarkan menuBar ke dalam Frame

```
self.SetMenuBar(menuBar)
```

8. Membuat dan mensetting StatusBar (*optional*)

```
#Membuat StatusBar
```

```
self.CreateStatusBar()
```

```
#Menuliskan tulisan pada StatusBar
```

```
self.SetStatusText("Selamat Datang di Python!")
```

9. Membuat Event pada tiap MenuItem

Syntax :

```
self.Bind(wx.EVT_MENU,self.nama_Event, id=no_akses)
```

no_akses harus sama dengan no_akses yang dibuat pada poin ke-3 (Menambahkan MenuItem pada wx.Menu yang sudah dibuat)

Contoh kode :

```
self.Bind(wx.EVT_MENU, self.OnTampil1, id=1)  
self.Bind(wx.EVT_MENU, self.OnTampil_Semua, id=2)  
self.Bind(wx.EVT_MENU, self.OnExit, id=3)
```

10. Membuat method berdasarkan event yang sudah dibuat

Contoh kode :

```
def OnTampil1(self, event):  
    self.main = FrCariData.create(None)  
    self.main.Show()  
  
def OnTampil_Semua(self, event):  
    self.main = FrLihatData.create(None)  
    self.main.Show()  
  
def OnExit(self, event):  
    self.Close()
```

CATATAN PENTING :

1. Semua kode selain import diletakkan setelah kode :

```
def __init__(self, parent):
```

```
self._init_ctrls(parent)
```

2. Kode yang dimulai dengan **def** harus sejajar dengan **def __init__(self, parent)** sedangkan kode lain lebih menjorok ke dalam, sejajar dengan **self._init_ctrls(parent)** → Ingat, Python sangat sensitif terhadap indentasi !

LISTING LENGKAP KODE (FrMenu_Utama.py)

```
#Boa:Frame:Frame1

import wx, FrEdit, FrGabungan, FrHapus,FrSimpan,
FrLihatData,FrCariData

def create(parent):
    return Frame1(parent)

[wxID_FRAME1] = [wx.NewId() for _init_ctrls in range(1)]

class Frame1(wx.Frame):
    def __init__(self, prnt):
        # generated method, don't edit
        wx.Frame.__init__(self, id=wxID_FRAME1, name='',
parent=prnt,
pos=wx.Point(-8, -8), size=wx.Size(1382, 744),
```

```
    style=wx.DEFAULT_FRAME_STYLE, title='Menu Utama')

    self.SetClientSize(wx.Size(1366, 706))

def __init__(self, parent):
    self._init_ctrls(parent)

    menuTampil_Data = wx.Menu()
    menuTampil_Data.Append(1, "Tampilkan Data 1 Record")

    menuTampil_Data.Append(2, "Tampilkan Semua Data")

    menuTampil_Data.AppendSeparator()

    menuTampil_Data.Append(3, "Exit")

    menuUbah_Data = wx.Menu()
    menuUbah_Data.Append(4, "Simpan Data")
    menuUbah_Data.Append(5, "Hapus Data")
    menuUbah_Data.Append(6, "Edit Data")

    menuGabungan = wx.Menu()
    menuGabungan.Append(7, "Frame Gabungan")

    menuBar = wx.MenuBar()

    menuBar.Append(menuTampil_Data, "Tampilkan Data")

    menuBar.Append(menuUbah_Data, "Ubah Data")
```

```
menuBar.Append(menuGabungan, "Gabungan")
self.SetMenuBar(menuBar)

#Membuat StatusBar
self.CreateStatusBar()

#Menuliskan tulisan pada StatusBar
self.SetStatusText("Selamat Datang di Python!")

self.Bind(wx.EVT_MENU, self.OnTampil1, id=1)
self.Bind(wx.EVT_MENU, self.OnTampil_Semua,
id=2)
self.Bind(wx.EVT_MENU, self.OnExit, id=3)
self.Bind(wx.EVT_MENU, self.OnSimpan_Data,
id=4)
self.Bind(wx.EVT_MENU, self.OnHapus_Data,
id=5)
self.Bind(wx.EVT_MENU, self.OnEdit_Data, id=6)
self.Bind(wx.EVT_MENU, self.OnGabungan, id=7)

def OnTampil1(self, event):
    self.main = FrCariData.create(None)
    self.main.Show()

def OnTampil_Semua(self, event):
    self.main = FrLihatData.create(None)
```

```
self.main.Show()

def OnExit(self, event):
        self.Close()

def OnSimpan_Data(self, event):
        self.main = FrSimpan.create(None)
        self.main.Show()

def OnHapus_Data(self, event):
        self.main = FrHapus.create(None)
        self.main.Show()

def OnEdit_Data(self, event):
        self.main = FrEdit.create(None)
        self.main.Show()

def OnGabungan(self, event):
        self.main = FrGabungan.create(None)
        self.main.Show()
```

BAB X

AKSES DATABASE LANJUTAN

10.1 PENDAHULUAN

Materi pembahasan bab ini meliputi :

1. RadioButton (Biodata Database dengan RadioButton)
2. ComboBox (Aplikasi Pemilihan Benua dan Negara)
3. Pencarian LIKE (Pencarian Judul Buku yang mengandung kata tertentu)
4. Tanggal (Pengolahan Data Tanggal/*Date*)
5. Gambar (Pengolahan Data Gambar/*Image*)
6. Operasi Numerik Desimal
7. Master Data Lengkap

10.2 Aplikasi Biodata Database dengan RadioButton

Pengenalan penggunaan wx.RadioButton sudah dibahas pada Bab V. Namun, kali ini akan

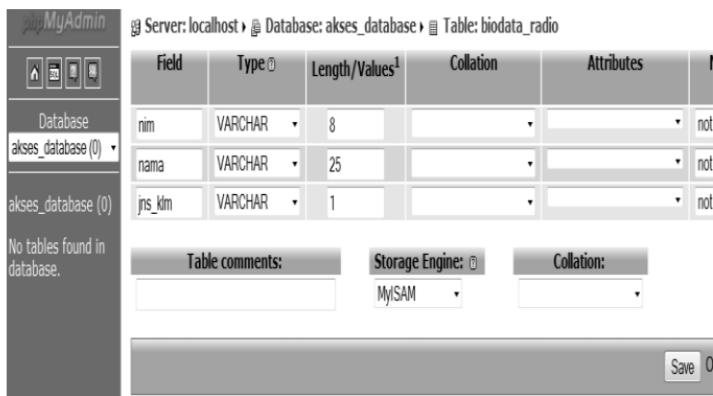
ditunjukkan penggunaan RadioButton yang terkait dengan akses ke Database.

Nama Database : akses_database

Nama Tabel : biodata_radio

Nama Field	Tipe Data	Lebar
nim	Varchar	8
nama	Varchar	25
jns_klm	Varchar	1

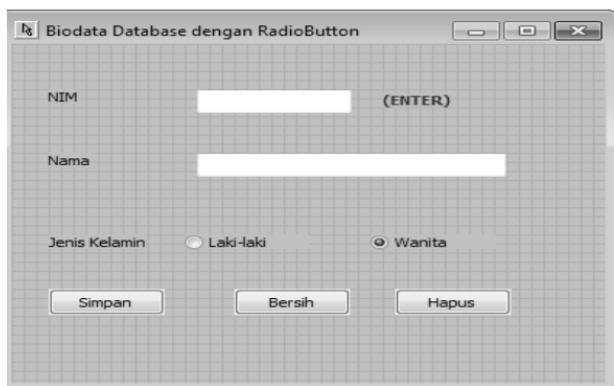
Field **jns_klm** akan menampung data jenis kelamin. Data tersebut hanya akan terisi 1 character saja, yaitu :'L' untuk Laki-laki, dan 'P' untuk Perempuan.



Gambar 10.1 Pembuatan Tabel *biodata_radio* di Database *akses_database*

(Secara detail, cara pembuatan database dan tabel bisa dilihat kembali pada bab VIII)

DESAIN FRAME



Gambar 10.2 Desain Frame Biodata Database dengan RadioButton

Frame ini sama dengan aplikasi gabungan pada bab VIII (subbab 8.6) yang memiliki fungsi pencarian, simpan, bersih, dan hapus. Namun ditambahkan wx.RadioButton untuk inputan jenis kelamin. Jenis kelamin menggunakan 2 object Radiobutton yaitu **rLaki** untuk Laki-laki dan **rWanita** untuk Wanita.

Fungsi/ fitur baru yang diperkenalkan dalam aplikasi ini ada 3, yaitu :

1. Menampilkan data dari database ke RadioButton.

Ketika user mengetikkan NIM, jika ada dalam database akan ditampilkan nama dan jenis

kelamin. Data jenis kelamin hanya ada 2 kemungkinan yaitu "L" untuk Laki-laki dan "P" untuk Perempuan. Proses ini ada dalam event **OnTxt_nimTextEnter**.

2. Menambah dan mengedit data berdasarkan kondisi terbaru di RadioButton.

Proses ini ada dalam event di tombol Simpan (**OnTmb_SimpanButton**)

3. Membersihkan RadioButton.

Proses ini ada dalam method **Bersih**.

Kode Pencarian Data (OnTxt_nimTextEnter)

```
def OnTxt_nimTextEnter(self, event):  
  
    sql = " select * from biodata_radio where nim \\  
          = '%s'" %(self.txt_nim.GetValue())  
  
    cur.execute(sql)  
  
    hasil= cur.fetchone()  
  
    if cur.rowcount > 0 :  
  
        self.txt_nama.SetValue(hasil[1])  
  
        if hasil[2]=="L" :  
  
            self.rLaki.SetValue(True)  
  
        else :  
  
            self.rWanita.SetValue(True)
```

```
else :  
    self.pesan = wx.MessageDialog(self, \  
        "Data Tidak Ada","Konfirmasi",wx.OK)  
    self.pesan.ShowModal()
```

PENJELASAN

Variable **hasil** menampung data yang diambil oleh Cursor **cur**. Jumlah field yang ada dalam tabel adalah 3, sedangkan penghitungan koefisien dimulai dari 0. Sehingga, untuk mengambil data pada field **jns_klm** menggunakan **hasil[2]**.

```
if hasil[2]=="L" :  
    self.rLaki.SetValue(True)  
else :  
    self.rWanita.SetValue(True)
```

jika data di **jns_klm** adalah “L”, maka RadioButton **rLaki** akan diset True atau terpilih. Sebaliknya, jika **jns_klm** adalah “P”, yang terpilih adalah **rWanita**.

KODE SIMPAN DAN EDIT DATA (Tombol Simpan)

```
def OnTmb_SimpanButton(self, event):  
    sql = " select * from biodata_radio where nim \  
        = '%s'" %(self.txt_nim.GetValue())  
    cur.execute(sql)
```

```
hasil= cur.fetchone()

if self.rLaki.GetValue()== True :
    jkl = "L"
else :
    jkl = "P"

if cur.rowcount > 0 :

    sql =" update biodata_radio set nama = \
        '%s', jns_klm='%s' where nim ='%s'"% \
        (self.txt_nama.GetValue(), \
        jkl,self.txt_nim.GetValue())

else :

    sql ="insert into biodata_radio (nim,nama, \
        jns_klm) \
        values ('%s','%s','%s')%(self.txt_nim.GetValue(), \
        self.txt_nama.GetValue(),jkl)

cur.execute(sql)

self.Bersih()
```

PENJELASAN

Sebelum masuk ke dalam kode menyimpan atau mengedit, diperlukan pengecekan manakah di antara 2 object wx.RadioButton yang terpilih: **rLaki**

atau **rWanita**. Hasil pengecekan tersebut disimpan dalam variabel **jkL**. Kodenya sebagai berikut:

```
if self.rLaki.GetValue() == True :  
    jkl = "L"  
  
else :  
  
    jkl = "P"
```

SQL Edit Data:

```
sql = " update biodata_radio set nama = \  
'%" s', jns_klm='%" s' where nim ='%" s'" % \  
(self.txt_nama.GetValue(), \  
jkl,self.txt_nim.GetValue())
```

SQL Tambah Data/ Simpan:

```
sql ="insert into biodata_radio (nim,nama,jns_klm) \  
values ('%" s', '%s', '%s')%" % (self.txt_nim.GetValue(), \  
self.txt_nama.GetValue(),jkl)
```

KODE BERSIHKAN ISIAN TEXTCTRL DAN RADIOBUTTON (Method Bersih)

```
def Bersih(self) :  
  
    self.txt_nim.SetValue("")  
  
    self.txt_nama.SetValue("")  
  
    self.rLaki.SetValue(False)
```

```
    self.rWanita.SetValue(False)  
    self.txt_nim.SetFocus()
```

PENJELASAN

self.rLaki.SetValue(False) → mengosongkan pilihan pada RadioButton **rLaki**.

self.rWanita.SetValue(False) → mengosongkan pilihan pada RadioButton **rWanita**.

Kesalahan yang sering terjadi adalah tulisan **False** ditulis **false** (huruf 'f' kecil), sehingga tidak akan menghasilkan hal yang diinginkan.

LISTING KODE LENGKAP (Frame Biodata Database dengan RadioButton)

```
#Boa:Frame:Frame1  
  
import wx, MySQLdb  
  
conn=MySQLdb.connect(host="localhost",user="root",p  
asswd="",db="akses_database")  
  
cur = conn.cursor()  
  
  
  
def create(parent):  
    return Frame1(parent)
```

```
[wxID_FRAME1, wxID_FRAME1PANEL1,
wxID_FRAME1RLAKI, wxID_FRAME1RWANITA,
wxID_FRAME1STATICTEXT1,
wxID_FRAME1STATICTEXT2,
wxID_FRAME1STATICTEXT3,
wxID_FRAME1STATICTEXT4,
wxID_FRAME1TMB_BERSIH, wxID_FRAME1TMB_HAPUS,
wxID_FRAME1TMB_SIMPAN, wxID_FRAME1TXT_NAMA,
wxID_FRAME1TXT_NIM,
] = [wx.NewId() for _init_ctrls in range(13)]
```



```
class Frame1(wx.Frame):
    def __init__(self, prnt):
        # generated method, don't edit
        wx.Frame.__init__(self, id=wxID_FRAME1, name='',
parent=prnt,
pos=wx.Point(866, 259), size=wx.Size(400, 337),
style=wx.DEFAULT_FRAME_STYLE,
title='Biodata Database dengan RadioButton')
        self.SetClientSize(wx.Size(384, 299))

        self.panel1 = wx.Panel(id=wxID_FRAME1PANEL1,
name='panel1', parent=self,
pos=wx.Point(0, 0), size=wx.Size(384, 299),
```

```

style=wx.TAB_TRAVERSAL)

self.panel1.SetBackgroundColour(wx.Colour(192,
192, 192))

self.staticText1 = =
wx.StaticText(id=wxID_FRAME1STATICTEXT1,
label='NIM',
name='staticText1', parent=self.panel1,
pos=wx.Point(24, 40),
size=wx.Size(20, 13), style=0)

self.txt_nim = =
wx.TextCtrl(id=wxID_FRAME1TXT_NIM, name='txt_nim',
parent=self.panel1, pos=wx.Point(120, 40),
size=wx.Size(100, 21),
style=wx.TE_PROCESS_ENTER, value="")

self.txt_nim.Bind(wx.EVT_TEXT_ENTER,
self.OnTxt_nimTextEnter,
id=wxID_FRAME1TXT_NIM)

self.staticText2 = =
wx.StaticText(id=wxID_FRAME1STATICTEXT2,
label='Nama', name='staticText2',
parent=self.panel1,
pos=wx.Point(24, 96), size=wx.Size(28, 13),
style=0)

```

```
    self.txt_nama =  
wx.TextCtrl(id=wxID_FRAME1TXT_NAMA,  
name='txt_nama',  
parent=self.panel1, pos=wx.Point(120, 96),  
size=wx.Size(200, 21),  
style=0, value='')  
  
    self.staticText3 =  
wx.StaticText(id=wxID_FRAME1STATICTEXT3,  
label='Jenis Kelamin', name='staticText3',  
parent=self.panel1,  
pos=wx.Point(24, 168), size=wx.Size(63, 13),  
style=0)  
  
    self.rLaki = wx.RadioButton(id=wxID_FRAME1RLAKI,  
label='Laki-laki',  
name='rLaki', parent=self.panel1,  
pos=wx.Point(112, 168),  
size=wx.Size(81, 13), style=0)  
    self.rLaki.SetValue(True)  
  
    self.rWanita =  
wx.RadioButton(id=wxID_FRAME1RWANITA,  
label='Wanita',
```

```
        name='rWanita',           parent=self.panel1,
pos=wx.Point(232, 168),
size=wx.Size(81, 13), style=0)

self.rWanita.SetValue(False)

self.tmb_Simpan = =
wx.Button(id=wxID_FRAME1TMB_SIMPAN,
label='Simpan',
name='tmb_Simpan',           parent=self.panel1,
pos=wx.Point(24, 216),
size=wx.Size(75, 23), style=0)

self.tmb_Simpan.Bind(wx.EVT_BUTTON,
self.OnTmb_SimpanButton,
id=wxID_FRAME1TMB_SIMPAN)

self.tmb_Bersih = =
wx.Button(id=wxID_FRAME1TMB_BERSIH, label='Bersih',
name='tmb_Bersih',           parent=self.panel1,
pos=wx.Point(144, 216),
size=wx.Size(75, 23), style=0)

self.tmb_Bersih.Bind(wx.EVT_BUTTON,
self.OnTmb_BersihButton,
id=wxID_FRAME1TMB_BERSIH)
```

```

        self.tmb_Hapus =
wx.Button(id=wxID_FRAME1TMB_HAPUS, label='Hapus',
           name='tmb_Hapus', parent=self.panel1,
           pos=wx.Point(248, 216),
           size=wx.Size(75, 23), style=0)

        self.tmb_Hapus.Bind(wx.EVT_BUTTON,
self.OnTmb_HapusButton,
           id=wxID_FRAME1TMB_HAPUS)

        self.staticText4 =
wx.StaticText(id=wxID_FRAME1STATICTEXT4,
              label='(ENTER)', name='staticText4',
              parent=self.panel1,
              pos=wx.Point(240, 43), size=wx.Size(44, 13),
              style=0)

        self.staticText4.SetForegroundColour(wx.Colour(255,
0, 0))

        self.staticText4SetFont(wx.Font(8, wx.SWISS,
wx.NORMAL, wx.BOLD, False,
           'Tahoma'))


def __init__(self, parent):
    self._init_ctrls(parent)

def Bersih(self) :
    self.txt_nim.SetValue("")

```

```

self.txt_nama.SetValue("")
self.rLaki.SetValue(False)
self.rWanita.SetValue(False)
self.txt_nim.SetFocus()

def OnTxt_nimTextEnter(self, event):
    sql = " select * from biodata_radio where nim \
    = '%s'" %(self.txt_nim.GetValue())
    cur.execute(sql)
    hasil= cur.fetchone()
    if cur.rowcount > 0 :
        self.txt_nama.SetValue(hasil[1])
        if hasil[2]=="L" :
            self.rLaki.SetValue(True)
        else :
            self.rWanita.SetValue(True)
    else :
        self.pesan = wx.MessageDialog(self,\n
        "Data Tidak Ada","Konfirmasi",wx.OK)
        self.pesan.ShowModal()

```

```

def OnTmb_SimpanButton(self, event):
    sql = " select * from biodata_radio where nim \
    = '%s'" %(self.txt_nim.GetValue())
    cur.execute(sql)
    hasil= cur.fetchone()
    if self.rLaki.GetValue()== True :
        jkl = "L"
    else :
        jkl = "P"
    if cur.rowcount > 0 :
        sql =" update biodata_radio set nama =\
        '%s', jns_klm='%s' where nim ='%s'%\ \
        (self.txt_nama.GetValue(),\
        jkl,self.txt_nim.GetValue())
    else :
        sql      ="insert      into      biodata_radio
(nim,nama,jns_klm) \
values ('%s','%s','%s')"% (self.txt_nim.GetValue(),\
self.txt_nama.GetValue(),jkl)
    cur.execute(sql)

```

```

self.Bersih()

def OnTmb_BersihButton(self, event):
    self.Bersih()

def OnTmb_HapusButton(self, event):
    if self.txt_nim.GetValue() == "":
        self.pesan = wx.MessageDialog\
            (self, "NIM belum diisi", "Konfirmasi", wx.OK)
        self.pesan.ShowModal()
        event.Skip()
    sql = " select * from biodata_radio where nim \
        = '%s'" % (self.txt_nim.GetValue())
    cur.execute(sql)
    hasil= cur.fetchone()
    if cur.rowcount>0 :
        tanya = wx.MessageDialog(self,\
            message="Anda Yakin Hendak Menghapus
NIM"\
        +" "+self.txt_nim.GetValue()+" "+\

```

```

"Nama "+self.txt_nama.GetValue()\
,style = wx.YES_NO)

if tanya.ShowModal()==wx.ID_YES:

    sql = "delete from biodata_radio \
where nim \
= '%s'" %(self.txt_nim.GetValue())

    cur.execute(sql)

else :

    self.pesan = wx.MessageDialog(self,\
"NIM yang akan dihapus tidak terdata\
di database","Konfirmasi",wx.OK)

    self.pesan.ShowModal()

```

10.3 ComboBox (Aplikasi Pemilihan Benua dan Negara)

Aplikasi berikut ini akan menunjukkan cara kerja ComboBox dalam akses ke database. Ada 2 ComboBox yaitu **cmb_benua** untuk menampilkan data benua, dan **cmb_negara** untuk menampilkan negara yang ada di benua tersebut. Isi dalam **cmb_negara** sangat tergantung pada pilihan pada **cmb_benua**. Sumber datanya berasal dari database.

Nama Database : akses_database

Nama Tabel : benua_negara

Field	Tipe	Lebar
benua	VarChar	25
negara	VarChar	25

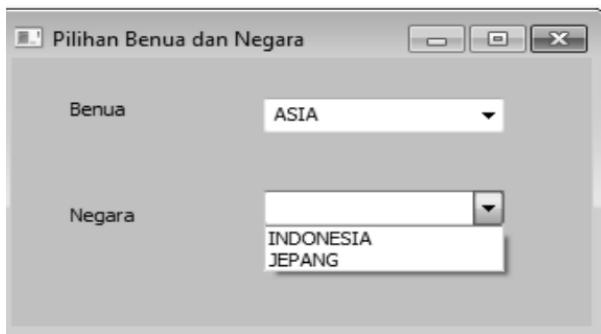
Contoh isi data dalam tabel **benua_negara**

Benua	Negara
ASIA	INDONESIA
AFRIKA	NIGERIA
AFRIKA	PANTAI GADING
EROPA	ALBANIA
AUSTRALIA	SELANDIA BARU
AMERIKA	SURINAME
ASIA	JEPANG

Jika **cmb_benua** dipilih AFRIKA, maka **cmb_negara** hanya akan menampilkan NIGERIA dan PANTAI GADING. Kalau yang dipilih di **cmb_benua** adalah ASIA maka yang ada di **cmb_negara** hanyalah INDONESIA dan JEPANG. Demikian seterusnya. Isi pada **cmb_negara** tergantung pada pilihan user terhadap **cmb_benua**.

Pada aplikasi ini juga akan diperkenalkan penggunaan perintah SQL :'SELECT DISTINCT'

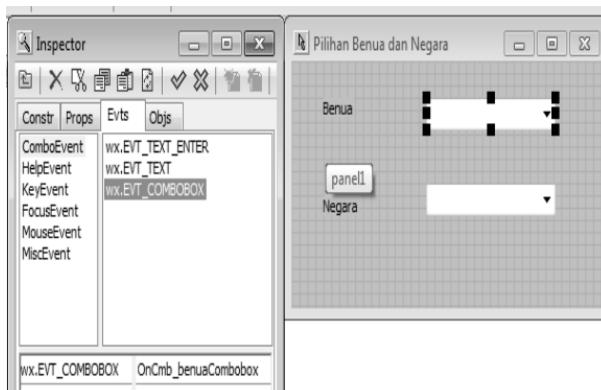
untuk mengambil data yang unik pada kolom (field) tertentu. Klik event pada ComboBox juga akan ditunjukkan, sehingga kode pengisian **cmb_negara** akan dieksekusi saat **cmb_benua** dipilih.



Gambar 10.3 Isi **cmb_negara** tergantung **cmb_benua**

KLIK EVENT PADA COMBOBOX **cmb_benua**

Pada aplikasi ini diterapkan klik event pada combobox **cmb_benua** (wx.EVT_COMBOBOX).



Gambar 10.4 **wx.EVT_COMBOBOX** pada **cmb_benua**

KODE INTI :

```
def Isi_Combo_Negara(self) :  
  
    # membersihkan semua item di cmb_negara  
  
    self.cmb_negara.Clear()  
  
    # variabel benua menyimpan data pilihan user  
  
    benua = self.cmb_benua.GetStringSelection()  
  
    sql = "select distinct(negara) from \  
benua_negara where \  
benua ='%s' "%(benua)  
  
    cur.execute(sql)  
  
    hasil= cur.fetchall()  
  
    if cur.rowcount > 0 :  
  
        for k in hasil:  
  
            # menambahkan data ke dalam cmb_negara  
  
            self.cmb_negara.Append(k[0])
```

def OnCmb_benuaCombobox(self, event):

```
# memanggil method Isi_Combo_Negara  
  
self.Isi_Combo_Negara()
```

LISTING KODE KESELURUHAN

```

#Boa:Frame:Frame1

import wx, MySQLdb

conn=MySQLdb.connect(host="localhost",user="root",p  
asswd="",db="akses_database")

cur = conn.cursor()

def create(parent):
    return Frame1(parent)

[wxID_FRAME1, wxID_FRAME1CMB_BENUA,
wxID_FRAME1CMB_NEGARA, wxID_FRAME1PANEL1,

wxID_FRAME1STATICTEXT1,
wxID_FRAME1STATICTEXT2,
] = [wx.NewId() for _init_ctrls in range(6)]


class Frame1(wx.Frame):
    def __init__(self, prnt):
        # generated method, don't edit
        wx.Frame.__init__(self, id=wxID_FRAME1, name="",
parent=prnt,
pos=wx.Point(869, 198), size=wx.Size(329, 200),

```

```
    style=wx.DEFAULT_FRAME_STYLE, title='Pilihan  
Benua dan Negara')  
  
    self.SetClientSize(wx.Size(313, 162))  
  
  
    self.panel1 = wx.Panel(id=wxID_FRAME1PANEL1,  
name='panel1', parent=self,  
  
        pos=wx.Point(0, 0), size=wx.Size(313, 162),  
  
        style=wx.TAB_TRAVERSAL)  
  
    self.panel1.SetBackgroundColour(wx.Colour(192,  
192, 192))  
  
  
  
    self.staticText1 =  
wx.StaticText(id=wxID_FRAME1STATICTEXT1,  
  
        label='Benua', name='staticText1',  
parent=self.panel1,  
  
        pos=wx.Point(32, 24), size=wx.Size(31, 13),  
style=0)  
  
  
  
    self.cmb_benua = wx.ComboBox(choices=[],  
id=wxID_FRAME1CMB_BENUA,  
  
        name='cmb_benua', parent=self.panel1,  
pos=wx.Point(136, 24),  
  
        size=wx.Size(130, 21), style=0, value="")  
  
    self.cmb_benua.SetLabel("")
```

```

        self.cmb_benua.Bind(wx.EVT_COMBOBOX,
self.OnCmb_benuaCombobox,
id=wxID_FRAME1CMB_BENUA)

self.staticText2 =
wx.StaticText(id=wxID_FRAME1STATICTEXT2,
label='Negara', name='staticText2',
parent=self.panel1,
pos=wx.Point(32, 88), size=wx.Size(36, 13),
style=0)

self.cmb_negara = wx.ComboBox(choices=[],
id=wxID_FRAME1CMB_NEGARA,
name='cmb_negara', parent=self.panel1,
pos=wx.Point(136, 80),
size=wx.Size(130, 21), style=0, value=")

self.cmb_negara.SetLabel(")

def __init__(self, parent):
    self._init_ctrls(parent)
    self.Isi_Combobox()

def Isi_Combobox (self) :
    self.cmb_benua.Clear
    sql = " select distinct(benua) from \

```

```
benua_negara "

    cur.execute(sql)

    hasil= cur.fetchall()

    if cur.rowcount > 0 :

        for k in hasil:

            self.cmb_benua.Append(k[0])

def Isi_Combo_Negara(self) :

    self.cmb_negara.Clear()

    benua = self.cmb_benua.GetStringSelection()

    sql = "select distinct(negara) from \

benua_negara where \

benua ='%s' "%(benua)

    cur.execute(sql)

    hasil= cur.fetchall()

    if cur.rowcount > 0 :

        for k in hasil:

            self.cmb_negara.Append(k[0])


def OnCmb_benuaCombobox(self, event):

    self.Isi_Combo_Negara()
```

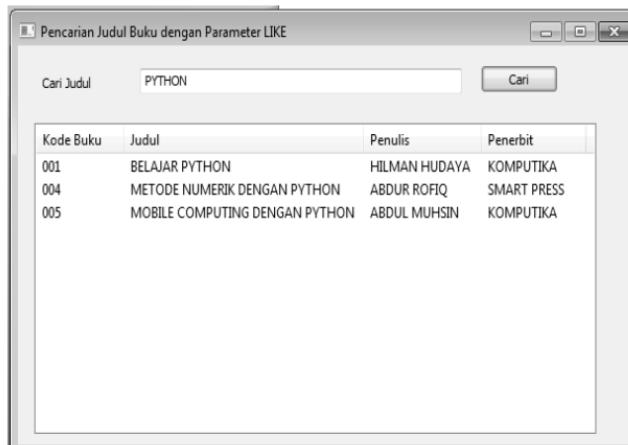
10.4 Pencarian LIKE (Pencarian Judul Buku yang mengandung kata tertentu)

Aplikasi berikut ini akan menunjukkan cara kerja pencarian suatu data yang mengandung suatu kata tertentu. Mencari data buku yang judulnya mengandung kata tertentu.

Gambar 10.5 menunjukkan tampilan seluruh data buku yang ada di tabel (database), sedangkan Gambar 10.6 menunjukkan tampilan data buku yang judulnya mengandung kata ‘PYTHON’.

Kode Buku	Judul	Penulis	Penerbit
001	BELAJAR PYTHON	HILMAN HUDAYA	KOMPUTIKA
002	BELAJAR JAVA	HIBBAN HUDAYA	SMART PRESS
003	METODE NUMERIK DENGAN JAVA	ABDURRAHMAN	PUSTAKA KRE...
004	METODE NUMERIK DENGAN PYTHON	ABDUR ROFIQ	SMART PRESS
005	MOBILE COMPUTING DENGAN PYTHON	ABDUL MUHSIN	KOMPUTIKA
006	MOBILE COMPUTING DENGAN C++	ZAKARIYA	SMART PRESS

Gambar 10.5 Seluruh Data yang Ada di Dalam Tabel



Gambar 10.6 Menampilkan Data Buku yang Judulnya Mengandung Kata 'PYHTON'

KOMPONEN PADA FRAME YANG UTAMA

Jenis Komponen	Name
wx.TextCtrl	txtcari_judul
wx.Button	tmb_Cari
wx.ListCtrl	lc

STRUKTUR DATABASE DAN TABEL

Nama Database : akses_database

Nama Tabel : buku

Field	Tipe	Lebar
kd_buku	Varchar	5

judul	Varchar	40
penulis	Varchar	25
penerbit	Varchar	25

KODE INTI (KODE PADA TOMBOL CARI)

```
def Isi_List (self) :
    self.lc.DeleteAllItems()
    sql = "select * from buku where judul LIKE
    "%s%"%(self.txtcari_judul.GetValue())
    cur.execute(sql)
    hasil= cur.fetchall()
    k =self.lc.GetItemCount()
    for i in hasil :
        self.lc.InsertStringItem(k,"%s"%i[0])
        self.lc.SetStringItem(k,1,"%s"%i[1])
        self.lc.SetStringItem(k,2,"%s"%i[2])
        self.lc.SetStringItem(k,3,"%s"%i[3])
    k = k + 1
def OnTmb_CariButton(self, event):
    # memanggil method Isi_List
    self.Isi_List()
```

LISTING KODE KESELURUHAN

```
#Boa:Frame:Frame1

import wx, MySQLdb

conn=MySQLdb.connect(host="localhost",user="root",password="",
asswd="",db="akses_database")

cur = conn.cursor()

def create(parent):
    return Frame1(parent)

[wxID_FRAME1,
 wxID_FRAME1LC,
 wxID_FRAME1PANEL1, wxID_FRAME1STATICTEXT1,
 wxID_FRAME1TMB_CARI,
 wxID_FRAME1TXTCARI_JUDUL,
] = [wx.NewId() for _init_ctrls in range(6)]

class Frame1(wx.Frame):
    def __init__(self, parent):
        # generated method, don't edit

        parent.InsertColumn(col=0,
format=wx.LIST_FORMAT_LEFT,
```

```

        heading='Kode Buku', width=85)

parent.InsertColumn(col=1,
format=wx.LIST_FORMAT_LEFT, heading='Judul',
width=232)

parent.InsertColumn(col=2,
format=wx.LIST_FORMAT_LEFT,
heading='Penulis', width=114)

parent.InsertColumn(col=3,
format=wx.LIST_FORMAT_LEFT,
heading='Penerbit', width=101)

def __init__(self, prnt):
    # generated method, don't edit

    wx.Frame.__init__(self, id=wx.ID_FRAME1, name='',
parent=prnt,
                      pos=wx.Point(387, 243), size=wx.Size(611, 367),
                      style=wx.DEFAULT_FRAME_STYLE,
                      title='Pencarian Judul Buku dengan Parameter
LIKE')

    self.SetClientSize(wx.Size(595, 329))

    self.panel1 = wx.Panel(id=wx.ID_FRAME1_PANEL1,
name='panel1', parent=self,
pos=wx.Point(0, 0), size=wx.Size(595, 329),

```

```

    style=wx.TAB_TRAVERSAL)

self.staticText1 =
wx.StaticText(id=wxID_FRAME1STATICTEXT1,
               label='Cari      Judul',      name='staticText1',
               parent=self.panel1,
               pos=wx.Point(24,    24),     size=wx.Size(48,    13),
               style=0)

self.txtcari_judul =
wx.TextCtrl(id=wxID_FRAME1TXTCARI_JUDUL,
            name='txtcari_judul',      parent=self.panel1,
            pos=wx.Point(118, 19),
            size=wx.Size(312, 21), style=0, value="")

self.tmb_Cari =
wx.Button(id=wxID_FRAME1TMB_CARI, label='Cari',
          name='tmb_Cari',      parent=self.panel1,
          pos=wx.Point(448, 16),
          size=wx.Size(75, 23), style=0)

self.tmb_Cari.Bind(wx.EVT_BUTTON,
                   self.OnTmb_CariButton,
                   id=wxID_FRAME1TMB_CARI)

```

```

    self.lc = wx.ListCtrl(id=wxID_FRAME1LC, name='lc',
parent=self.panel1,
pos=wx.Point(16, 64), size=wx.Size(544, 256),
style=wx.LC_REPORT)

    self._init_coll_lc_Columns(self.lc)

def __init__(self, parent):
    self._init_ctrls(parent)

def Isi_List (self) :

    self.lc.DeleteAllItems()

    sql = "select * from buku where judul \
LIKE '%%%%%s%%%"%(self.txtcari_judul.GetValue())

    cur.execute(sql)

    hasil= cur.fetchall()

    k =self.lc.GetItemCount()

    for i in hasil :

        self.lc.InsertStringItem(k,"%s"%i[0])

        self.lc.SetStringItem(k,1,"%s"%i[1])

        self.lc.SetStringItem(k,2,"%s"%i[2])

        self.lc.SetStringItem(k,3,"%s"%i[3])

        k = k + 1

```

```
def OnTmb_CariButton(self, event):
    self.Isi_List()
```

10.5 Pengolahan Data Tanggal (Date)

Aplikasi berikut ini untuk menunjukkan pengolahan data tanggal, yang meliputi:

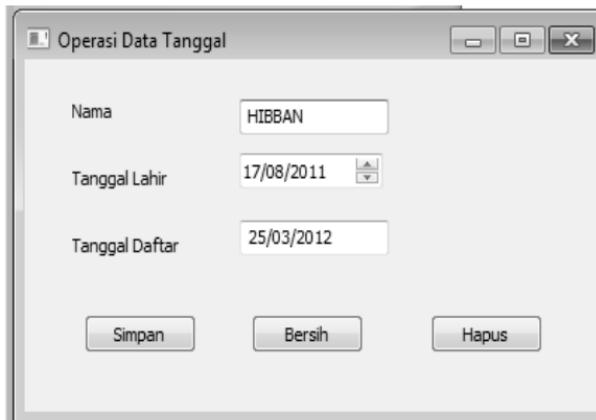
1. Penggunaan komponen wx.DatePickerCtrl
2. Menampilkan tanggal hari ini ke TextCtrl
3. Menyimpan data tanggal dari Frame ke Database MySQL tipe data *Date*.
4. Menampilkan data tanggal dari Database ke Frame

Komponen wx.DatePickerCtrl terletak nomor 2 dari kanan pada tab **Basic Controls** di Zope Editor



Gambar 10.7 Icon wx.DatePickerCtrl

Aplikasi contoh ini adalah input data identitas yang meliputi nama, tgl_lahir, dan tgl_daftar.



Gambar 10.8 Aplikasi Operasi Data Tanggal

KOMPONEN FRAME YANG UTAMA

Jenis Komponen	Name
wx.TextCtrl	txt_nama
wx.TextCtrl	tgl_daftar
wx.DatePickerCtrl	tgl_lahir
wx.Button	tmb_simpan
wx.Button	tmb_bersih
wx.Button	tmb_hapus

STRUKTUR DATABASE DAN TABEL

Nama Database : akses_database

Nama Tabel : operasi_tanggal

Nama Field	Tipe	Lebar
Nama	VarChar	25
tgl_lahir	Date	
tgl_daftar	Date	

Menampilkan Tanggal Hari ini ke TextCtrl tgl_daftar

1. Import Module untuk Tanggal

```
import datetime
```

Module ini diperlukan untuk seluruh operasi yang terkait dengan *Date* (tanggal)

2. Ambil tanggal hari ini secara utuh, kemudian pecah komponennya berdasarkan hari, bulan, dan tahun

```
# Ambil Tanggal Hari ini
skrg = datetime.date.today()

# Ambil Hari
day = skrg.day

# Ambil Bulan
month = skrg.month

# Ambil Tahun
year = skrg.year
```

Sekumpulan baris ini diletakkan di bagian atas sebelum masuk ke dalam method-method (def....). Tujuannya agar kode-kode tersebut bersifat global, dikenal di semua method.

3. Menampilkan tanggal ke textctrl **tgl_daftar** dengan memformat menjadi : dd-mm-yyyy (2 digit hari-2 digit bulan-4 digit tahun)

```
self.tgl_daftar.SetValue("%02d/%02d/%4d"%(day,  
month, year))
```

Agar pengaruhnya terlihat langsung pada saat Frame pertama kali dijalankan, maka kode tersebut diletakkan persis di bawah kode:

```
def __init__(self, parent):  
    self._init_ctrls(parent)
```

Membaca Inputan di wx.DatePickerCtrl tgl_lahir dan Menyimpan ke Dalam Database

Proses ini diperlukan pada tombol Simpan

1. Membaca data pada **tgl_lahir** dan memecahnya menjadi hari, bulan, dan tahun

```
selected = self.tgl_lahir.GetValue()  
month_lahir = selected.Month + 1  
day_lahir = selected.Day  
year_lahir = selected.Year
```

2. Mempersiapkan variabel bertipe Date bernama **tgl_lahir1** sebagai bahan baku untuk menyimpan ke database

```
tgl_lahir1 = \  
datetime.date(year_lahir,month_lahir,day_lahir)
```

Data yang dipersiapkan dalam bentuk yyyy-mm-dd (4 digit tahun-2 digit bulan-2 digit hari) sesuai dengan format default pada *Date MySQL*.

3. SQL untuk menyimpan ke tabel **operasi_tanggal**

```
sql = "insert into operasi_tanggal (tgl_lahir) values \  
(%s) %(tgl_lahir1)"
```

Kode tersebut adalah kode contoh untuk menyimpan ke satu field. Kode asli dalam aplikasi bisa dilihat di listing lengkap kode

Menampilkan Data Tanggal dari Database ke Frame

Berikut ini adalah langkah-langkah utama untuk menampilkan data tanggal bertipe *Date* dari MySQL ke komponen yang ada di Frame:

1. Membaca data di field yang bertipe Date, kemudian memecahnya menjadi hari, bulan, dan tahun

```
python_birthdate = hasil[1]  
day = python_birthdate.day
```

```
month = python_birthdate.month
```

```
year = python_birthdate.year
```

2. Mengatur format tanggal yang akan ditampilkan (DMY) dan menyimpan ke dalam variabel

```
displayed_birthday = \  
wx.DateTimeFromDMY(day,month-1,year)
```

3. Menampilkan hasilnya ke dalam wx.DatePickerCtrl

```
self.tgl_lahir.SetValue(displayed_birthday)
```

Prosesnya sama jika anda ingin menampilkan data ke TextCtrl.

LISTING KODE LENGKAP

```
#Boa:Frame:Frame1  
  
  
import datetime  
import wx, MySQLdb  
  
  
conn=MySQLdb.connect(host="localhost",\  
user="root",passwd="",db="akses_database")  
cur = conn.cursor()
```

```
# Ambil Tanggal Hari ini
skrg = datetime.date.today()

# Ambil Hari
day = skrg.day

# Ambil Bulan
month = skrg.month

# Ambil Tahun
year = skrg.year

def create(parent):
    return Frame1(parent)

[wxID_FRAME1,
 wxID_FRAME1STATICTEXT1,
 wxID_FRAME1STATICTEXT2,
 wxID_FRAME1STATICTEXT3,
 wxID_FRAME1TGL_DAFTAR,
 wxID_FRAME1TGL_LAHIR,  wxID_FRAME1TMB_BERSIH,
 wxID_FRAME1TMB_HAPUS,
 wxID_FRAME1TMB_SIMPAN, wxID_FRAME1TXT_NAMA,
] = [wx.NewId() for _init_ctrls in range(11)]
```

```

class Frame1(wx.Frame):

    def __init_ctrls(self, prnt):

        # generated method, don't edit

        wx.Frame.__init__(self, id=wxID_FRAME1, name="",
parent=prnt,

            pos=wx.Point(487, 252), size=wx.Size(400, 248),

            style=wx.DEFAULT_FRAME_STYLE,
title='Operasi Data Tanggal')

        self.SetClientSize(wx.Size(384, 210))

        self.panel1 = wx.Panel(id=wxID_FRAME1PANEL1,
name='panel1', parent=self,

            pos=wx.Point(0, 0), size=wx.Size(384, 210),

            style=wx.TAB_TRAVERSAL)

        self.staticText1 = wx.StaticText(id=wxID_FRAME1STATICTEXT1,
parent=self.panel1,

            label='Nama', name='staticText1',
parent=self.panel1,

            pos=wx.Point(32, 24), size=wx.Size(28, 13),
style=0)

        self.txt_nama = wx.TextCtrl(id=wxID_FRAME1TXT_NAMA,
name='txt_nama',
parent=self.panel1,
pos=wx.Point(32, 56), size=wx.Size(28, 24),
style=0)

```

```

        parent=self.panel1,      pos=wx.Point(144,      24),
size=wx.Size(100, 21),

        style=wx.TE_PROCESS_ENTER, value='')

self.txt_nama.Bind(wx.EVT_TEXT_ENTER,
self.OnTxt_namaTextEnter,

id=wxID_FRAME1TXT_NAMA)

        self.staticText2 = =
wx.StaticText(id=wxID_FRAME1STATICTEXT2,
label='Tanggal      Lahir',      name='staticText2',
parent=self.panel1,

pos=wx.Point(32,      64),      size=wx.Size(65,      13),
style=0)

        self.tgl_lahir = =
wx.DatePickerCtrl(id=wxID_FRAME1TGL_LAHIR,
name='tgl_lahir',                  parent=self.panel1,
pos=wx.Point(144, 56),

size=wx.Size(96,                      21),
style=wx.DP_SHOWCENTURY)

self.tgl_lahir.SetLabel(")

self.tgl_lahir.SetValue(wx.DateTimeFromDMY(24,    2,
2012, 0, 0, 0))

self.tgl_lahir.SetHelpText(")

```

```
    self.tmb_simpan =  
wx.Button(id=wxID_FRAME1TMB_SIMPAN,  
label='Simpan',  
  
          name='tmb_simpan',           parent=self.panel1,  
pos=wx.Point(40, 152),  
  
          size=wx.Size(75, 23), style=0)  
  
    self.tmb_simpan.Bind(wx.EVT_BUTTON,  
self.OnTmb_simpanButton,  
  
          id=wxID_FRAME1TMB_SIMPAN)  
  
  
  
    self.tmb_bersih =  
wx.Button(id=wxID_FRAME1TMB_BERSIH, label='Bersih',  
  
          name='tmb_bersih',           parent=self.panel1,  
pos=wx.Point(152, 152),  
  
          size=wx.Size(75, 23), style=0)  
  
    self.tmb_bersih.Bind(wx.EVT_BUTTON,  
self.OnTmb_bersihButton,  
  
          id=wxID_FRAME1TMB_BERSIH)  
  
  
  
    self.tmb_hapus =  
wx.Button(id=wxID_FRAME1TMB_HAPUS, label='Hapus',  
  
          name='tmb_hapus',           parent=self.panel1,  
pos=wx.Point(272, 152),  
  
          size=wx.Size(75, 23), style=0)  
  
    self.tmb_hapus.Bind(wx.EVT_BUTTON,  
self.OnTmb_hapusButton,
```

```

    id=wxID_FRAME1TMB_HAPUS)

self.staticText3 =  

wx.StaticText(id=wxID_FRAME1STATICTEXT3,  

label='Tanggal Daftar', name='staticText3',  

parent=self.panel1,  

pos=wx.Point(32, 104), size=wx.Size(73, 13),  

style=0)

self.tgl_daftar =  

wx.TextCtrl(id=wxID_FRAME1TGL_DAFTAR,  

name='tgl_daftar', parent=self.panel1,  

pos=wx.Point(144, 96),  

size=wx.Size(100, 21), style=0, value="")

def __init__(self, parent):  

    self._init_ctrls(parent)  

    ## Menampilkan Tanggal Hari ini  

    # Tampilkan ke dalam textctrl tgl_daftar  

    self.tgl_daftar.SetValue("%02d/%02d/%4d"%(day,  

month, year))

def Bersih(self) :  

    self.txt_nama.SetValue("")
```

```

    self.tgl_lahir.SetValue("%02d/%02d/%4d" % (day,
month, year))

def OnTmb_simpanButton(self, event):

    sql = " select * from operasi_tanggal where nama
\

    = '%s'" %(self.txt_nama.GetValue())

    cur.execute(sql)

    hasil= cur.fetchone()

    selected = self.tgl_lahir.GetValue()

    month_lahir = selected.Month + 1

    day_lahir = selected.Day

    year_lahir = selected.Year

    tgl_lahir1 =
datetime.date(year_lahir,month_lahir,day_lahir)

    tgl_daftar1 = datetime.date(year,month,day)

    if cur.rowcount > 0 :

        sql =" update operasi_tanggal set tgl_lahir = \
'%s', tgl_daftar='%s' where nama ='%s'"%\
(tgl_lahir1,\
```

```

    tgl_daftar1,self.txt_nama.GetValue()

else :

    sql      ="insert      into      operasi_tanggal
(nama,tgl_lahir,tgl_daftar) \
values
('%s','%s','%s')%"%(self.txt_nama.GetValue(),\
                     tgl_lahir1,tgl_daftar1)

cur.execute(sql)

self.Bersih()

def OnTmb_bersihButton(self, event):

    self.Bersih()

def OnTmb_hapusButton(self, event):

    if self.txt_nama.GetValue() == "" :
        self.pesan = wx.MessageDialog\
        (self,"Nama belum diisi","Konfirmasi",wx.OK)
        self.pesan.ShowModal()

    event.Skip()

    sql = " select * from operasi_tanggal where nama
\

= "%s'" %(self.txt_nama.GetValue())

```

```
cur.execute(sql)

hasil= cur.fetchone()

if cur.rowcount>0 :

    tanya = wx.MessageDialog(self,\n
        message="Anda Yakin Hendak Menghapus\n
Nama"\\
        +" "+self.txt_nama.GetValue()\
        ,style = wx.YES_NO)

    if tanya.ShowModal()==wx.ID_YES:

        sql = "delete from operasi_tanggal where\n
nama \
        = '%s'" % (self.txt_nama.GetValue())

        cur.execute(sql)

    else :

        self.pesan = wx.MessageDialog(self,\n
        "Nama yang akan dihapus tidak terdata\
        di database","Konfirmasi",wx.OK)

        self.pesan.ShowModal()

def OnTxt_namaTextEnter(self, event):
    \\\n
    sql = " select * from operasi_tanggal where nama
```

```

= "%s" %(self.txt_nama.GetValue())

cur.execute(sql)

hasil= cur.fetchone()

if cur.rowcount > 0 :

    python_birthdate = hasil[1]

    day = python_birthdate.day

    month = python_birthdate.month

    year = python_birthdate.year

    displayed_birthday = wx.DateTimeFromDMY(day,month-1,year)

    self.tgl_lahir.SetValue(displayed_birthday)

    tgl_daftar1 = hasil[2]

    day1 = tgl_daftar1.day

    month1 = tgl_daftar1.month

    year1 = tgl_daftar1.year

    displayed_birthday1 = wx.DateTimeFromDMY(day1,month1-1,year1)

    self.tgl_daftar.SetValue(displayed_birthday1)

else :

    self.pesan = wx.MessageDialog(self,\

        "Data Tidak Ada","Konfirmasi",wx.OK)

    self.pesan.ShowModal()

```

10.6 Pengolahan Data Gambar (Image)

Berikut ini adalah contoh aplikasi yang menggunakan gambar. Contoh aplikasi yang digunakan adalah pendataan data buku meliputi : Kode Buku, Judul, Penulis, Penerbit, dan Gambar. Data yang disimpan ke dalam database adalah alamat penyimpanan gambar.



Gambar 10.9 Aplikasi Operasi Data Gambar

Pada aplikasi ini akan ditunjukkan cara dalam :

1. Penggunaan `wx.StaticBitmap` untuk menampilkan gambar statis
2. Penggunaan tombol `Browse` untuk mengambil alamat gambar dan menampilkan gambar ke `wx.StaticBitmap`
3. Menyimpan alamat gambar dari `textCtrl` ke database

4. Menampilkan gambar dari database ke Frame

KOMPONEN INTI YANG DIGUNAKAN DALAM FRAME

Jenis Komponen	Name
wx.TextCtrl	txt_kd_buku
	txt_judul
	txt_penulis
	txt_penerbit
	txt_path
wx.StaticBitmap	gambar
wx.Button	tmb_Simpan
	tmb_Bersih
	tmb_Hapus
	tmb_Browse

STRUKTUR DATABASE DAN TABEL

Nama Database : akses_database

Nama Tabel : buku_gambar

Field	Tipe	Lebar
kd_buku	VarChar	3

judul	VarChar	25
penulis	VarChar	25
penerbit	VarChar	25
gambar	VarChar	60

wx.StaticBitmap

Untuk menampilkan gambar statis pada Frame dibutuhkan komponen wx.StaticBitmap. Komponen ini terletak di urutan ke-9 dari sebelah kiri pada tab **Basic Controls** pada Zope Editor.



Gambar 10.10 Icon wx.StaticBitmap

Menampilkan Gambar ke wx.StaticBitmap

1. Menentukan alamat (path) lengkap file gambar yang akan ditampilkan

```
filepath = 'C:\\\\Gambar\\\\noimage.jpg'
```

di Sistem Operasi Windows, pengenalan path pada Python sebagai pemisah antar folder adalah menggunakan tanda '\\'.

2. Setting wx.Image

```
img = wx.Image(filepath, wx.BITMAP_TYPE_ANY)
```

3. Menampilkan hasil setting `wx.Image` ke `wx.StaticBitmap` **gambar**

```
self.gambar.SetBitmap(wx.BitmapFromImage(img))
```

Tombol Browse untuk File Dialog, Mengambil Alamat Gambar dan Menampilkan Gambar di `wx.StaticText`

1. Setting wildcard (jenis file gambar yang akan ditampilkan di Open Dialog

```
wildcard = "JPEG files (*.jpg) | *.jpg"
```

2. Setting File Dialog

```
dialog = wx.FileDialog(None, "Choose a file",  
                      wildcard=wildcard,  
                      style=wx.OPEN)
```

File Dialog adalah tampilan yang memungkinkan user memilih file gambar pada komputer dengan menjelajahi Drive dan Folder.



Gambar 10.11 File Dialog setelah tombol Browse ditekan

3. Jika tombol Open (OK) ditekan, maka alamat gambar secara lengkap akan ditampilkan pada textctrl **txt_path**

```
if dialog.ShowModal() == wx.ID_OK:  
    self.txt_path.SetValue(dialog.GetPath())
```

4. File Dialog ditutup

```
dialog.Destroy()
```

5. Penyesuaian alamat (path)

```
filepath = self.txt_path.GetValue()  
filepath = filepath.replace("\\\\",'\\\\\\\\')
```

Penyesuaian dilakukan agar bisa disimpan dalam database. Jika sebelumnya alamat asli adalah : C:\Gambar\noimage.jpg, maka setelah penyesuaian menjadi :

C:\\Gambar\\noimage.jpg

6. Menampilkan gambar ke wx.StaticText

```
self.txt_path.SetValue(filepath)  
  
img = wx.Image(filepath, wx.BITMAP_TYPE_ANY)  
  
self.gambar.SetBitmap(wx.BitmapFromImage(img))
```

MENYIMPAN ALAMAT GAMBAR DARI FRAME KE MYSQL

1. Ubah *Collation* pada field yang akan menyimpan alamat gambar di MySQL dari *latin1_general_ci* ke *latin1_swedish_ci*.
2. Siapkan variabel untuk menyimpan ke database

```
gambar1= self.txt_path.GetValue()
```

3. Lakukan SQL penyimpanan data

```
sql = “insert into buku_gambar (gambar) values \\  
(%s)”%(gambar1)
```

Jika pada alamat (path) yang ditampilkan di TextCtrl adalah : **C:\\Gambar\\noimage.jpg**, maka alamat yang akan tersimpan dalam database adalah :**C:\\Gambar\\noimage.jpg**.

MENAMPILKAN GAMBAR DARI DATABASE KE FRAME

1. Menangkap data dari database, dan menyimpan ke variabel **filepath**.

```
filepath = hasil[4]
```

Kode ini menunjukkan bahwa field yang diambil adalah field ke-5. Hitungan index dimulai dari 0, sehingga ditampung dalam **hasil[4]**.

2. Menyesuaikan format alamat path.

```
filepath = filepath.replace("\\\\","\\\\\\\\")
```

3. Menampilkan gambar ke wx.StaticText **gambar** berdasarkan format alamat path yang telah disesuaikan

```
self.txt_path.SetValue(filepath)
```

```
img = wx.Image(filepath, wx.BITMAP_TYPE_ANY)
self.gambar.SetBitmap(wx.BitmapFromImage(img))
```

LISTING KODE LENGKAP

```
#Boa:Frame:Frame1

import wx, MySQLdb

conn= MySQLdb.connect(host="localhost", user="root",
passwd="",db="akses_database")

cur = conn.cursor()

def create(parent):
    return Frame1(parent)
```

```
[wxID_FRAME1, wxID_FRAME1GAMBAR,
wxID_FRAME1PANEL1, wxID_FRAME1STATICTEXT1,
wxID_FRAME1STATICTEXT2,
wxID_FRAME1STATICTEXT3,
wxID_FRAME1STATICTEXT4,
wxID_FRAME1STATICTEXT5,
wxID_FRAME1TMB_BERSIH,
wxID_FRAME1TMB_BROWSER,
wxID_FRAME1TMB_HAPUS,
wxID_FRAME1TMB_SIMPAN, wxID_FRAME1TXT_JUDUL,
wxID_FRAME1TXT_KD_BUKU, wxID_FRAME1TXT_PATH,
wxID_FRAME1TXT_PENERBIT,
wxID_FRAME1TXT_PENULIS,
] = [wx.NewId() for _init_ctrls in range(17)]
```

```
class Frame1(wx.Frame):
    def __init__(self, prnt):
        # generated method, don't edit
        wx.Frame.__init__(self, id=wxID_FRAME1, name="",
parent=prnt,
pos=wx.Point(483, 244), size=wx.Size(473, 308),
style=wx.DEFAULT_FRAME_STYLE,
title='Penggunaan Gambar')
        self.SetClientSize(wx.Size(457, 270))
```

```

    self.panel1 = wx.Panel(id=wxID_FRAME1PANEL1,
name='panel1', parent=self,
pos=wx.Point(0, 0), size=wx.Size(457, 270),
style=wx.TAB_TRAVERSAL)

self.staticText1 =
wx.StaticText(id=wxID_FRAME1STATICTEXT1,
label='Kode Buku', name='staticText1',
parent=self.panel1,
pos=wx.Point(24, 32), size=wx.Size(51, 13),
style=0)

self.txt_kd_buku =
wx.TextCtrl(id=wxID_FRAME1TXT_KD_BUKU,
name='txt_kd_buku', parent=self.panel1,
pos=wx.Point(112, 32),
size=wx.Size(100, 21),
style=wx.TE_PROCESS_ENTER, value='')

self.txt_kd_buku.Bind(wx.EVT_TEXT_ENTER,
self.OnTxt_kd_bukuTextEnter,
id=wxID_FRAME1TXT_KD_BUKU)

self.staticText2 =
wx.StaticText(id=wxID_FRAME1STATICTEXT2,

```

```

        label='Judul',           name='staticText2',
parent=self.panel1,
pos=wx.Point(24,    88),   size=wx.Size(25,    13),
style=0)

self.txt_judul = wx.TextCtrl(id=wxID_FRAME1TXT_JUDUL,
name='txt_judul',
parent=self.panel1,   pos=wx.Point(112,    88),
size=wx.Size(176, 21),
style=0, value="")

self.staticText3 = wx.StaticText(id=wxID_FRAME1STATICTEXT3,
label='Penulis',           name='staticText3',
parent=self.panel1,
pos=wx.Point(24,    144),   size=wx.Size(34,    13),
style=0)

self.txt_penulis = wx.TextCtrl(id=wxID_FRAME1TXT_PENULIS,
name='txt_penulis',         parent=self.panel1,
pos=wx.Point(112, 136),
size=wx.Size(176, 21), style=0, value="")

```

```

    self.staticText4 =  

wx.StaticText(id=wxID_FRAME1STATICTEXT4,  

             label='Penerbit',           name='staticText4',  

             parent=self.panel1,  

             pos=wx.Point(24,   192),   size=wx.Size(41,   13),  

             style=0)  

    self.txt_penerbit =  

wx.TextCtrl(id=wxID_FRAME1TXT_PENERBIT,  

            name='txt_penerbit',       parent=self.panel1,  

            pos=wx.Point(116, 184),  

            size=wx.Size(172, 21), style=0, value="")  

    self.gambar =  

wx.StaticBitmap(bitmap=wx.NullBitmap,  

                id=wxID_FRAME1GAMBAR,      name='gambar',  

                parent=self.panel1,  

                pos=wx.Point(320,  32),   size=wx.Size(104,  112),  

                style=0)  

    self.tmb_Simpan =  

wx.Button(id=wxID_FRAME1TMB_SIMPAN,  

          label='Simpan',           name='tmb_Simpan',  

          parent=self.panel1,  

          pos=wx.Point(24, 232),  

          size=wx.Size(75, 23), style=0)

```

```

    self.tmb_Simpan.Bind(wx.EVT_BUTTON,
self.OnTmb_SimpanButton,
        id=wxID_FRAME1TMB_SIMPAN)

    self.tmb_Bersih =
wx.Button(id=wxID_FRAME1TMB_BERSIH, label='Bersih',
        name='tmb_Bersih',           parent=self.panel1,
pos=wx.Point(152, 232),
        size=wx.Size(75, 23), style=0)

    self.tmb_Bersih.Bind(wx.EVT_BUTTON,
self.OnTmb_BersihButton,
        id=wxID_FRAME1TMB_BERSIH)

    self.tmb_Hapus =
wx.Button(id=wxID_FRAME1TMB_HAPUS, label='Hapus',
        name='tmb_Hapus',           parent=self.panel1,
pos=wx.Point(288, 232),
        size=wx.Size(75, 23), style=0)

    self.tmb_Hapus.Bind(wx.EVT_BUTTON,
self.OnTmb_HapusButton,
        id=wxID_FRAME1TMB_HAPUS)

    self.tmb_Browse =
wx.Button(id=wxID_FRAME1TMB_BROWSE,
label='Browse',

```

```

        name='tmb_Browse',           parent=self.panel1,
pos=wx.Point(320, 160),
size=wx.Size(104, 23), style=0)

self.tmb_Browse.Bind(wx.EVT_BUTTON,
self.OnTmb_BrowseButton,
id=wxID_FRAME1TMB_BROWSE)

self.txt_path
wx.TextCtrl(id=wxID_FRAME1TXT_PATH, name='txt_path',
parent=self.panel1,    pos=wx.Point(304, 192),
size=wx.Size(136, 21),
style=0, value="")

self.staticText5
wx.StaticText(id=wxID_FRAME1STATICTEXT5,
label='(Enter)',           name='staticText5',
parent=self.panel1,
pos=wx.Point(224, 32),   size=wx.Size(40, 13),
style=0)

self.staticText5.SetForegroundColour(wx.Colour(255,
0, 0))

self.staticText5SetFont(wx.Font(8,           wx.SWISS,
wx.NORMAL, wx.BOLD, False,
'Tahoma'))
```

```

def __init__(self, parent):
    self._init_ctrls(parent)

    filepath = 'C:\\\\Gambar\\\\noimage.jpg'

    img = wx.Image(filepath, wx.BITMAP_TYPE_ANY)
    self.gambar.SetBitmap(
        wx.BitmapFromImage(img))

def Isi_Object(self) :
    sql = "select * from buku_gambar where kd_buku
= '%s' \\
    %(self.txt_kd_buku.GetValue())
    cur.execute(sql)

    if cur.rowcount > 0 :
        hasil = cur.fetchone()
        self.txt_judul.SetValue(hasil[1])
        self.txt_penulis.SetValue(hasil[2])
        self.txt_penerbit.SetValue(hasil[3])
        filepath = hasil[4]
        filepath = filepath.replace("\\\\",'\\\\\\\\')
        self.txt_path.SetValue(filepath)
        img = wx.Image\\

```

```
(filepath, wx.BITMAP_TYPE_ANY)

self.gambar.SetBitmap(wx.BitmapFromImage(img))

def Awal(self) :

    self.txt_kd_buku.SetValue("")

    self.txt_judul.SetValue("")

    self.txt_penulis.SetValue("")

    self.txt_penerbit.SetValue("")

    self.txt_path.SetValue("")

    self.txt_kd_buku.SetFocus()

## Mengisi StaticBitmap gambar dengan tampilan
noimage

filepath = 'C:\\\\Gambar\\\\noimage.jpg'

img = wx.Image(filepath, wx.BITMAP_TYPE_ANY)

self.gambar.SetBitmap\\

(wx.BitmapFromImage(img))

def OnTxt_kd_bukuTextEnter(self, event):

    self.Isi_Object()

def OnTmb_SimpanButton(self, event):
```

```

sql = "select * from buku_gambar where
kd_buku='%s' "%\

(self.txt_kd_buku.GetValue())

cur.execute(sql)

judul1 = self.txt_judul.GetValue()

penulis1 = self.txt_penulis.GetValue()

penerbit1= self.txt_penerbit.GetValue()

kd_buku1 =self.txt_kd_buku.GetValue()

gambar1= self.txt_path.GetValue()

if cur.rowcount > 0 :

    sql = "update buku_gambar set judul
=%s,penulis=%s,penerbit=%s, gambar=%s where \
kd_buku =%s"
    "%(judul1,penulis1,penerbit1,gambar1,kd_buku1)"

else :

    sql      =      "insert      into      buku_gambar
(kd_buku,judul,penulis,penerbit,gambar) \
values ('%s','%s','%s','%s','%s') "%\
(kd_buku1,judul1,penulis1,penerbit1,gambar1)

    # Collation di MySQL diganti ke
latin1_swedish_ci

```

```

        cur.execute(sql)

        conn.commit()

        self.Awal()

def OnTmb_BersihButton(self, event):

    self.Awal()

def OnTmb_HapusButton(self, event):

    if self.txt_kd_buku.GetValue() == "":
        self.pesan = wx.MessageDialog\
            (self, "Kode Buku belum  
diisi", "Konfirmasi", wx.OK)

        self.pesan.ShowModal()

        event.Skip()

    sql = " select * from buku_gambar where kd_buku \
        = '%s'" % (self.txt_kd_buku.GetValue())

    cur.execute(sql)

    hasil= cur.fetchone()

    if cur.rowcount>0 :

        tanya = wx.MessageDialog(self,\

            message="Anda Yakin Hendak Menghapus  
Kode Buku"\


```

```

+" "+self.txt_kd_buku.GetValue()+" "+\
"Judul "+self.txt_judul.GetValue()\
,style = wx.YES_NO)

if tanya.ShowModal()==wx.ID_YES:

    sql = "delete from buku_gambar where
kd_buku \
= '%s'" %(self.txt_kd_buku.GetValue())

    cur.execute(sql)

else :

    self.pesan = wx.MessageDialog(self,\
"Kode Buku yang akan dihapus tidak terdata\
di database","Konfirmasi",wx.OK)

    self.pesan.ShowModal()

    self.Awal()

def OnTmb_BrowseButton(self, event):

    wildcard = "JPEG files (*.jpg)|*.jpg"

    dialog = wx.FileDialog(None, "Choose a file",
                           wildcard=wildcard,
                           style=wx.OPEN)

    if dialog.ShowModal() == wx.ID_OK:

```

```
    self.txt_path.SetValue(dialog.GetPath())

    dialog.Destroy()

    filepath = self.txt_path.GetValue()

    filepath = filepath.replace("\\\\",'\\\\')

    self.txt_path.SetValue(filepath)

    img = wx.Image(filepath, wx.BITMAP_TYPE_ANY)

    self.gambar.SetBitmap\
(wx.BitmapFromImage(img))
```

10.7 Operasi Numerik Desimal

Jika sebelumnya telah ditunjukkan cara penyimpanan data bertipe String (varchar), Integer, dan Date, maka berikut ini akan ditampilkan cara penyimpanan data bertipe float. Tipe data ini bisa menampung data numerik dengan beberapa angka di belakang koma (dalam setting kita tanda desimalnya adalah titik).

Aplikasi yang dijadikan contoh adalah penyimpanan data mahasiswa yang meliputi : NIM, Nama, dan IPK. Data IPK tersebut yang akan disimpan dalam database dengan tipe float.

NIM

Nama

IPK

Gambar 10.12 Aplikasi Data Mahasiswa dengan IPK

Hal-hal yang bisa dipelajari dari contoh aplikasi ini adalah:

1. Konversi data dari String ke Float
2. Menyimpan data bertipe float ke database
3. Menampilkan data bertipe float dari database ke Frame

KOMPONEN UTAMA PADA FRAME

Jenis Komponen	Name
wx.TextCtrl	txt_nim
	txt_nama
	txt_ipk

wx.Button	tmb_Simpan
	tmb_Bersih
	tmb_Hapus

STRUKTUR DATABASE DAN TABEL

Nama Database : akses_database

Nama Tabel : ipk

Field	Tipe	Lebar
nim	VarChar	3
nama	VarChar	25
ipk	Float	2

KONVERSI DATA (STRING DAN FLOAT)

Konversi data dari String ke float :

```
a = float (b)
```

a adalah variabel bertipe float, sedangkan b adalah variabel bertipe String

Konversi data dari float ke String:

```
k = str(n)
```

k adalah variabel bertipe String sedangkan n adalah variabel bertipe float.

MENYIMPAN DATA FLOAT DARI FRAME(TEXTCTRL) KE DATABASE (TABEL)

1. Konversi data String yang ada di textctrl ke variabel baru bertipe float

```
ipk1 = float(self.txt_ipk.GetValue())
```

2. SQL untuk perintah menyimpan

```
sql = " insert into ipk (ipk) values ('%s') "% (ipk1)
```

SQL tersebut sekedar contoh, kode riilnya bisa dilihat pada bagian **LISTING KODE LENGKAP**.

MENAMPILKAN DATA DARI DATABASE (TABEL) KE FRAME (TEXTCTRL)

Tampilkan data dari field yang bertipe float ke textctrl dengan terlebih dahulu mengkonversi dari float ke String:

```
self.txt_ipk.SetValue(str(hasil[2]))
```

LISTING KODE LENGKAP

```
#Boa:Frame:Frame1

import wx, MySQLdb

conn=MySQLdb.connect(host="localhost",
user="root",passwd="",db="akses_database")
cur = conn.cursor()
```

```
def create(parent):
    return Frame1(parent)

[wxID_FRAME1,
 wxID_FRAME1STATICTEXT1,
 wxID_FRAME1STATICTEXT2,
 wxID_FRAME1STATICTEXT3,
 wxID_FRAME1TMB_BERSIH,
 wxID_FRAME1TMB_HAPUS,
 wxID_FRAME1TMB_SIMPAN, wxID_FRAME1TXT_IPK,
 wxID_FRAME1TXT_NAMA, wxID_FRAME1TXT_NIM,
] = [wx.NewId() for _init_ctrls in range(11)]


class Frame1(wx.Frame):
    def __init__(self, prnt):
        # generated method, don't edit
        wx.Frame.__init__(self, id=wxID_FRAME1, name="",
parent=prnt,
pos=wx.Point(562, 248), size=wx.Size(400, 271),
style=wx.DEFAULT_FRAME_STYLE,
title='Operasi Numerik Desimal')

        self.SetClientSize(wx.Size(384, 233))
```

```

    self.panel1 = wx.Panel(id=wxID_FRAME1PANEL1,
name='panel1', parent=self,
pos=wx.Point(0, 0), size=wx.Size(384, 233),
style=wx.TAB_TRAVERSAL)

self.staticText1 =
wx.StaticText(id=wxID_FRAME1STATICTEXT1,
label='NIM',
name='staticText1', parent=self.panel1,
pos=wx.Point(32, 32),
size=wx.Size(20, 13), style=0)

self.txt_nim =
wx.TextCtrl(id=wxID_FRAME1TXT_NIM, name='txt_nim',
parent=self.panel1, pos=wx.Point(120, 32),
size=wx.Size(100, 21),
style=wx.TE_PROCESS_ENTER, value="")

self.txt_nim.Bind(wx.EVT_TEXT_ENTER,
self.OnTxt_nimTextEnter,
id=wxID_FRAME1TXT_NIM)

self.staticText2 =
wx.StaticText(id=wxID_FRAME1STATICTEXT2,
label='Nama', name='staticText2',
parent=self.panel1,

```

```
    pos=wx.Point(32,    88),    size=wx.Size(28,    13),
style=0)

self.txt_nama =  
wx.TextCtrl(id=wxID_FRAME1TXT_NAMA,  
name='txt_nama',  
parent=self.panel1,    pos=wx.Point(120,    80),  
size=wx.Size(184, 21),  
style=0, value="")

self.staticText3 =  
wx.StaticText(id=wxID_FRAME1STATICTEXT3, label='IPK',  
name='staticText3',           parent=self.panel1,  
pos=wx.Point(32, 144),  
size=wx.Size(17, 13), style=0)

self.txt_ipk = wx.TextCtrl(id=wxID_FRAME1TXT_IPK,  
name='txt_ipk',  
parent=self.panel1,    pos=wx.Point(120,    136),  
size=wx.Size(100, 21),  
style=0, value="")

self.tmb_Simpan =  
wx.Button(id=wxID_FRAME1TMB_SIMPAN,  
label='Simpan',
```

```

        name='tmb_Simpan',           parent=self.panel1,
pos=wx.Point(32, 192),
size=wx.Size(75, 23), style=0)

self.tmb_Simpan.Bind(wx.EVT_BUTTON,
self.OnTmb_SimpanButton,
id=wxID_FRAME1TMB_SIMPAN)

        self.tmb_Bersih           =
wx.Button(id=wxID_FRAME1TMB_BERSIH, label='Bersih',
name='tmb_Bersih',           parent=self.panel1,
pos=wx.Point(136, 192),
size=wx.Size(75, 23), style=0)

self.tmb_Bersih.Bind(wx.EVT_BUTTON,
self.OnTmb_BersihButton,
id=wxID_FRAME1TMB_BERSIH)

        self.tmb_Hapus           =
wx.Button(id=wxID_FRAME1TMB_HAPUS, label='Hapus',
name='tmb_Hapus',           parent=self.panel1,
pos=wx.Point(240, 192),
size=wx.Size(75, 23), style=0)

self.tmb_Hapus.Bind(wx.EVT_BUTTON,
self.OnTmb_HapusButton,
id=wxID_FRAME1TMB_HAPUS)

```

```

def __init__(self, parent):
    self._init_ctrls(parent)

def Bersih (self) :
    self.txt_nim.SetValue("")
    self.txt_nama.SetValue("")
    self.txt_ipk.SetValue("")
    self.txt_nim.SetFocus()

def OnTmb_SimpanButton(self, event):
    sql = " select * from ipk where nim \
    = '%s'" %(self.txt_nim.GetValue())
    cur.execute(sql)
    hasil= cur.fetchone()
    ipk1 = float(self.txt_ipk.GetValue())
    if cur.rowcount > 0 :
        sql =" update ipk set nama =\
        '%s', ipk=%s' where nim ='%s'"%\
        (self.txt_nama.GetValue(), \
        ipk1,self.txt_nim.GetValue())
    else :

```

```

        sql ="insert into ipk (nim,nama,ipk) \
        values
        ('%s','%s','%s')%"%(self.txt_nim.GetValue(),\
        self.txt_nama.GetValue(),ipk1)

        cur.execute(sql)

        self.Bersih()

def OnTmb_BersihButton(self, event):
    self.Bersih()

def OnTmb_HapusButton(self, event):
    if self.txt_nim.GetValue() == "" :
        self.pesan = wx.MessageDialog\
        (self,"NIM belum diisi","Konfirmasi",wx.OK)
        self.pesan.ShowModal()
        event.Skip()

    sql = " select * from ipk where nim \
    = '%s'" % (self.txt_nim.GetValue())
    cur.execute(sql)

    hasil= cur.fetchone()

    if cur.rowcount>0 :

```

```

tanya = wx.MessageDialog(self,\n
                        message="Anda Yakin Hendak Menghapus\n
NIM"\n
\n
                    +" "+self.txt_nim.GetValue()+" "+\\
\n
                    "Nama "+self.txt_nama.GetValue()\\\n
\n
                    ,style = wx.YES_NO)\n
\n
if tanya.ShowModal()==wx.ID_YES:\n
\n
    sql = "delete from ipk where nim \\n\n
\n
        = '%s'" % (self.txt_nim.GetValue())\n
\n
    cur.execute(sql)\n
\n
else :\n
\n
    self.pesan = wx.MessageDialog(self,\n
\n
        "NIM yang akan dihapus tidak terdata\\n\n
di database","Konfirmasi",wx.OK)\n
\n
    self.pesan.ShowModal()\n
\n
\n
def OnTxt_nimTextEnter(self, event):\n
\n
    sql = " select * from ipk where nim \\n\n
\n
        = '%s'" %(self.txt_nim.GetValue())\n
\n
    cur.execute(sql)\n
\n
    hasil= cur.fetchone()

```

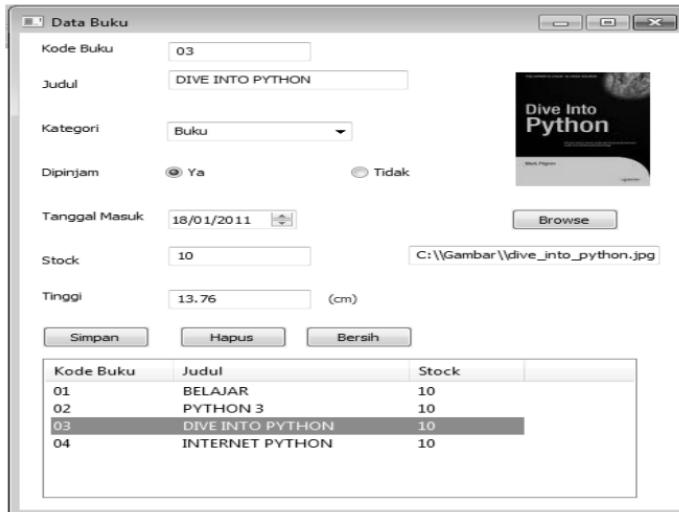
```
if cur.rowcount > 0 :  
    self.txt_nama.SetValue(hasil[1])  
    self.txt_ipk.SetValue(str(hasil[2]))  
  
else :  
  
    self.pesan = wx.MessageDialog(self,\n        "Data Tidak Ada","Konfirmasi",wx.OK)  
  
    self.pesan.ShowModal()
```

10.8 Master Data Lengkap

Aplikasi ‘Master Data Lengkap’ adalah kumpulan dari berbagai fitur/ kemampuan akses ke database yang telah dibahas sebelumnya. Contoh aplikasi yang digunakan adalah input data buku.

Semua tipe data untuk akses ke database dicontohkan. Tipe data String : Kode Buku, Judul, Kategori, Gambar (alamat gambar). Tipe data Date : Tanggal Masuk. Tipe data Integer : Stock buku, keterangan boleh tidak-nya dipinjam (tinyint: 1 untuk dipinjam, 0 untuk tidak boleh dipinjam). Tipe data float : Tinggi Buku (dalam cm).

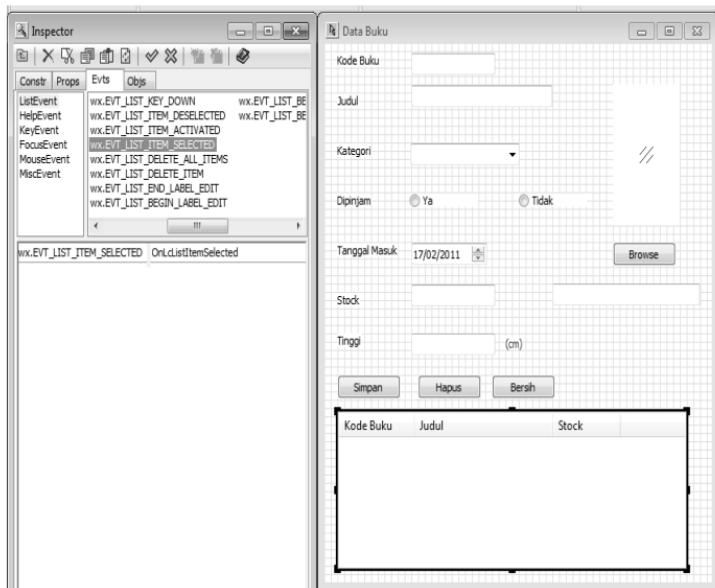
Komponen-komponen pada Frame juga beragam. Textctrl untuk Kode buku, Judul, Stock, Tinggi. RadioButton untuk pilihan status buku bisa dipinjam atau tidak. Kategori (Buku atau Majalah) menggunakan ComboBox. Tanggal Masuk menggunakan wx.DatePickerCtrl. Data ditampilkan dalam bentuk tabel dengan wx.ListCtrl, sedangkan gambar ditampilkan dengan wx.StaticBitmap.



Gambar 10.13 Aplikasi Master Gabungan Input Data Buku

Hal baru yang dipelajari pada bagian ini adalah menampilkan data pada tiap komponen ketika item pada wx.ListCtrl dipilih (diklik). Langkah-langkahnya adalah :

1. Buat Event pada wx.ListCtrl : Evts> ListEvent> wx.EVT_LIST_ITEM_SELECTED



Gambar 10.14 `wx.EVT_LIST_ITEM_SELECTED` pada `wx.ListCtrl`

2. Mengambil nomor baris index yang dipilih/ diklik, dan disimpan ke dalam variable a

```
a = event.m_itemIndex
```

3. Mengambil data pada ListCtrl untuk baris ke- a dan kolom ke-0 (kode buku) dan disimpan dalam variabel b

```
b = self.lc.GetItem(a,0).GetText()
```

4. Menampilkan data variabel b ke textctrl **txt_kd_buku**

```
self.txt_kdbuku.SetValue(b)
```

5. Memanggil method untuk mengisi komponen dengan data dari database untuk kode buku yang sesuai dengan yang tercantum di **txt_kdbuku**.

```
self.Isi_Object()
```

Method Isi_Object bisa dilihat pada **LISTING KODE LENGKAP**

KOMPONEN UTAMA PADA FRAME

Komponen	Name	Keterangan
wx.TextCtrl	txt_kdbuku	
	txt_judul	
	txt_stock	
	txt_tinggi	Tinggi buku dalam cm
	txt_path	Alamat lengkap gambar yang akan ditampilkan di wx.StaticBitmap (gambar)
wx.ComboBox	cmbkategori	Pilihan kategori: Buku atau Majalah
wx.RadioButton	rYa	Pilihan Jenis

	rTidak	Buku dipinjam atau Tidak. boleh (Ya)
wx.DatePickerCtrl	tgl_masuk	Tanggal Masuk buku ke Perpustakaan
wx.StaticBitmap	gambar	Menampilkan cover buku
wx.ListCtrl	lc	Menampilkan Data buku yang dalam 3 kolom (Kode Buku, Judul, Stock)

STRUKTUR DATABASE DAN TABEL

Nama Database : perpus

Nama Tabel : buku

Field	Tipe	Lebar	Keterangan
kd_buku	VarChar	15	
judul	Varchar	25	
kategori	VarChar	15	
dipinjam	Tinyint	1	
tgl_masuk	Date		

stock	Int	11	
tinggi	Float	2	
gambar	Varchar	60	Collation harus diubah menjadi: latin1_swedish_ci. Kolom lain tidak harus diubah collationnya.

LISTING KODE LENGKAP

```
#Boa:Frame:Frame1

import wx, MySQLdb
import datetime

conn=MySQLdb.connect(host="localhost",
user="root", passwd="",db="perpus")

cur = conn.cursor()

def create(parent):
    return Frame1(parent)
```

```
[wxID_FRAME1,           wxID_FRAME1CMBKATEGORI,
wxID_FRAME1GAMBAR, wxID_FRAME1LC,
wxID_FRAME1PANEL1,      wxID_FRAME1RTIDAK,
wxID_FRAME1RYA, wxID_FRAME1STATICTEXT1,
wxID_FRAME1STATICTEXT2,
wxID_FRAME1STATICTEXT3,
wxID_FRAME1STATICTEXT4,
wxID_FRAME1STATICTEXT5,
wxID_FRAME1STATICTEXT6,
wxID_FRAME1STATICTEXT7,
wxID_FRAME1STATICTEXT8,
wxID_FRAME1STATICTEXT9,
wxID_FRAME1TGL_MASUK,
wxID_FRAME1TMBBERSIH,
wxID_FRAME1TMBBROWSE, wxID_FRAME1TMBHAPUS,
wxID_FRAME1TMBSIMPAN, wxID_FRAME1TXT_JUDUL,
wxID_FRAME1TXT_KDBUKU,
wxID_FRAME1TXT_PATH,    wxID_FRAME1TXT_STOCK,
wxID_FRAME1TXT_TINGGI,
] = [wx.NewId() for _init_ctrls in range(26)]
```

```
class Frame1(wx.Frame):
    def __init__(self, parent):
        # generated method, don't edit
        parent.InsertColumn(col=0,
format=wx.LIST_FORMAT_LEFT,
```

```

        heading='Kode Buku', width=90)

    parent.InsertColumn(col=1,
format=wx.LIST_FORMAT_LEFT, heading='Judul',
width=165)

    parent.InsertColumn(col=2,
format=wx.LIST_FORMAT_LEFT, heading='Stock',
width=-1)

def __init__(self, prnt):
    # generated method, don't edit

    wx.Frame.__init__(self, id=wx.ID_FRAME1, name='',
parent=prnt,
pos=wx.Point(370, 120), size=wx.Size(478, 558),
style=wx.DEFAULT_FRAME_STYLE, title='Data
Buku')

    self.SetClientSize(wx.Size(462, 520))

    self.SetBackgroundColour(wx.Colour(255, 255, 255))

    self.panel1 = wx.Panel(id=wx.ID_FRAME1PANEL1,
name='panel1', parent=self,
pos=wx.Point(0, 0), size=wx.Size(462, 520),
style=wx.TAB_TRAVERSAL)

```

```

    self.staticText1 =  

wx.StaticText(id=wxID_FRAME1STATICTEXT1,  

        label='Kode Buku', name='staticText1',  

parent=self.panel1,  

        pos=wx.Point(16, 10), size=wx.Size(50, 13),  

style=0)  

    self.txt_kdbuku =  

wx.TextCtrl(id=wxID_FRAME1TXT_KDBUKU,  

        name='txt_kdbuku', parent=self.panel1,  

pos=wx.Point(104, 10),  

        size=wx.Size(100, 21),  

style=wx.TE_PROCESS_ENTER, value=")  

    self.txt_kdbuku.Bind(wx.EVT_TEXT_ENTER,  

self.OnTxt_kdbukuTextEnter,  

        id=wxID_FRAME1TXT_KDBUKU)  

    self.staticText2 =  

wx.StaticText(id=wxID_FRAME1STATICTEXT2,  

        label='Judul', name='staticText2',  

parent=self.panel1,  

        pos=wx.Point(16, 49), size=wx.Size(25, 13),  

style=0)  

    self.txt_judul =  

wx.TextCtrl(id=wxID_FRAME1TXT_JUDUL,  

name='txt_judul',  


```

```

        parent=self.panel1,    pos=wx.Point(104,    41),
size=wx.Size(168, 21),

        style=0, value=")

self.staticText3 = wx.StaticText(id=wxID_FRAME1STATICTEXT3,
label='Kategori',           name='staticText3',
parent=self.panel1,

        pos=wx.Point(16,    97),   size=wx.Size(40,    13),
style=0)

self.cmbkategori = wx.ComboBox(choices=[], id=wxID_FRAME1CMBKATEGORI,
name='cmbkategori',         parent=self.panel1,
pos=wx.Point(103, 97),

        size=wx.Size(130, 21), style=0, value=")

self.cmbkategori.SetLabel(")

self.staticText4 = wx.StaticText(id=wxID_FRAME1STATICTEXT4,
label='Dipinjam',           name='staticText4',
parent=self.panel1,

        pos=wx.Point(16,    145),   size=wx.Size(40,    13),
style=0)

```

```

self.rYa = wx.RadioButton(id=wxID_FRAME1RYA,
label='Ya', name='rYa',
parent=self.panel1, pos=wx.Point(102, 145),
size=wx.Size(81, 13),
style=0)

self.rYa.SetValue(False)

self.rTidak =
wx.RadioButton(id=wxID_FRAME1RTIDAK, label='Tidak',
name='rTidak', parent=self.panel1,
pos=wx.Point(232, 145),
size=wx.Size(81, 13), style=0)

self.rTidak.SetValue(False)

self.staticText5 =
wx.StaticText(id=wxID_FRAME1STATICTEXT5,
label='staticText5', name='staticText5',
parent=self.panel1,
pos=wx.Point(-80, 8), size=wx.Size(54, 13),
style=0)

self.staticText6 =
wx.StaticText(id=wxID_FRAME1STATICTEXT6,
label='Tanggal Masuk', name='staticText6',
parent=self.panel1,

```

```

        pos=wx.Point(16, 193), size=wx.Size(71, 13),
style=0)

self.tgl_masuk =  

wx.DatePickerCtrl(id=wxID_FRAME1TGL_MASUK,  

name='tgl_masuk', parent=self.panel1,  

pos=wx.Point(104, 193),  

size=wx.Size(90, 21),  

style=wx.DP_SHOWCENTURY)  

self.tgl_masuk.SetValue(wx.DateTimeFromDMY(17,  

1, 2011, 0, 0, 0))  

self.tgl_masuk.SetLabel('17/2/2011')

self.staticText7 =  

wx.StaticText(id=wxID_FRAME1STATICTEXT7,  

label='Stock', name='staticText7',  

parent=self.panel1,  

pos=wx.Point(16, 241), size=wx.Size(26, 13),
style=0)

self.txt_stock =  

wx.TextCtrl(id=wxID_FRAME1TXT_STOCK,  

name='txt_stock',  

parent=self.panel1, pos=wx.Point(104, 233),
size=wx.Size(100, 21),  

style=0, value="")
```

```

    self.tmbSimpan =  

wx.Button(id=wxID_FRAME1TMBSIMPAN,  

label='Simpan',  

        name='tmbSimpan',           parent=self.panel1,  

pos=wx.Point(16, 320),  

        size=wx.Size(75, 23), style=0)  

    self.tmbSimpan.Bind(wx.EVT_BUTTON,  

self.OnTmbSimpanButton,  

        id=wxID_FRAME1TMBSIMPAN)  

    self.tmbHapus =  

wx.Button(id=wxID_FRAME1TMBHAPUS, label='Hapus',  

        name='tmbHapus',           parent=self.panel1,  

pos=wx.Point(112, 320),  

        size=wx.Size(75, 23), style=0)  

    self.tmbHapus.Bind(wx.EVT_BUTTON,  

self.OnTmbHapusButton,  

        id=wxID_FRAME1TMBHAPUS)  

    self.tmbBersih =  

wx.Button(id=wxID_FRAME1TMBBERSIH, label='Bersih',  

        name='tmbBersih',           parent=self.panel1,  

pos=wx.Point(200, 320),  

        size=wx.Size(75, 23), style=0)

```

```

    self.tmbBersih.Bind(wx.EVT_BUTTON,
self.OnTmbBersihButton,
        id=wxID_FRAME1TMBBERSIH)

    self.lc = wx.ListCtrl(id=wxID_FRAME1LC, name='lc',
parent=self.panel1,
        pos=wx.Point(16, 355), size=wx.Size(416, 152),
        style=wx.LC_REPORT)
    self._init_coll_lc_Columns(self.lc)

    self.lc.Bind(wx.EVT_LIST_ITEM_SELECTED,
self.OnLcListItemSelected,
        id=wxID_FRAME1LC)

    self.gambar =
wx.StaticBitmap(bitmap=wx.NullBitmap,
        id=wxID_FRAME1GAMBAR, name='gambar',
parent=self.panel1,
        pos=wx.Point(344, 40), size=wx.Size(80, 136),
style=0)

    self.tmbBrowse =
wx.Button(id=wxID_FRAME1TMBBROWSE,
label='Browse',
        name='tmbBrowse', parent=self.panel1,
pos=wx.Point(344, 192),

```

```
size=wx.Size(75, 23), style=0)

self.tmbBrowse.Bind(wx.EVT_BUTTON,
self.OnTmbBrowseButton,
    id=wxID_FRAME1TMBBROWSE)

self.txt_path =
wx.TextCtrl(id=wxID_FRAME1TXT_PATH,
name='txt_path',
    parent=self.panel1,    pos=wx.Point(272,  232),
size=wx.Size(176, 21),
    style=0, value="")

self.staticText8 =
wx.StaticText(id=wxID_FRAME1STATICTEXT8,
    label='Tinggi',           name='staticText8',
parent=self.panel1,
    pos=wx.Point(16,  280),   size=wx.Size(29,  13),
style=0)

self.txt_tinggi =
wx.TextCtrl(id=wxID_FRAME1TXT_TINGGI,
    name='txt_tinggi',         parent=self.panel1,
pos=wx.Point(104, 280),
    size=wx.Size(100, 21), style=0, value="")
```

```

    self.staticText9 =
wx.StaticText(id=wxID_FRAME1STATICTEXT9,
                label='(cm)', name='staticText9',
                parent=self.panel1,
                pos=wx.Point(216, 283), size=wx.Size(21, 13),
                style=0)

    self.staticText9.SetForegroundColour(wx.Colour(255,
0, 0))

def __init__(self, parent):
    self._init_ctrls(parent)

    skrg = datetime.date.today()

    day = skrg.day

    month = skrg.month

    tahun = skrg.year

    displayed = wx.DateTimeFromDMY(day,month-1,tahun)

    displayed.Format("%d/%m/%Y")

    #month1, day1, year1 = tampilan.split('/')

    #myDate = wx.DateTimeFromDMY(int(day),
int(month)-1 , int(year1))

    self.tgl_masuk.SetValue(displayed)

    self.cmbkategori.Append("Buku")

    self.cmbkategori.Append("Majalah")

```

```
#self.imageCtrl.SetBitmap(wx.BitmapFromImage(img))

filepath = 'C:\\Gambar\\noimage.jpg'

img = wx.Image(filepath, wx.BITMAP_TYPE_ANY)

self.gambar.SetBitmap(wx.BitmapFromImage(img))

self.Isi_List()

def Isi_Object(self) :

    sql = "select * from buku where kd_buku = '%s' "
    %(self.txt_kdbuku.GetValue())

    cur.execute(sql)

    if cur.rowcount > 0 :

        hasil = cur.fetchone()

        self.txt_judul.SetValue(hasil[1])

        self.cmbkategori.SetStringSelection(hasil[2])

        if hasil[3] == 0 :

            self.rTidak.Value=True

        else :

            self.rYa.Value = True

        tgl = hasil[4]

        day = tgl.day

        month = tgl.month
```

```

year = tgl.year
displayed =
wx.DateTimeFromDMY(day,month-1,year)

self.tgl_masuk.SetValue(displayed)

self.txt_stock.SetValue(str(hasil[5]))

self.txt_tinggi.SetValue(str(hasil[6]))

filepath = hasil[7]

filepath = filepath.replace("\\\\",'\\\\')

self.txt_path.SetValue(filepath)

img = wx.Image(filepath,
wx.BITMAP_TYPE_ANY)

self.gambar.SetBitmap(wx.BitmapFromImage(img))

def OnTxt_kdbukuTextEnter(self, event):

    self.Isi_Object()

def OnTmbSimpanButton(self, event):

    sql = "select * from buku where kd_buku='%" + "%(self.txt_kdbuku.GetValue())"

    cur.execute(sql)

    judul1 = self.txt_judul.GetValue()

    kategori1 = self.cmbkategori.GetStringSelection()

    if self.rYa.Value == True :

```

```

dipinjam1 = 1

else :

dipinjam1 = 0

selected = self.tgl_masuk.GetValue()

month = selected.Month + 1

day = selected.Day

year = selected.Year

tgl_masuk1 = datetime.date(year,month,day)

stock1 = int(self.txt_stock.GetValue())

tinggi1 = float(self.txt_tinggi.GetValue())

kd_buku1 =self.txt_kdbuku.GetValue()

gambar1= self.txt_path.GetValue()

if cur.rowcount > 0 :

sql = "update buku set judul ='%s', kategori='%s', dipinjam='%d', tgl_masuk='%s', stock ='%d', tinggi='%s',gambar='%s' where kd_buku ='%s' '%(judul1,kategori1,dipinjam1,tgl_masuk1,stock1,tinggi1,gambar1,kd_buku1)"

else :

sql = "insert into buku(kd_buku,judul,kategori,dipinjam,tgl_masuk,stock,ga mbar) values ('%s','%s','%s','%d','%s','%d','%s','%s')"

```

```
"%(kd_buku1,judul1,kategori1,dipinjam1,tgl_masuk1,s
tock1,tinggil1,gambar1)

    cur.execute(sql)

    conn.commit()

    self.Awal()

def Awal(self) :

    self.Isi_List()

    self.txt_kdbuku.SetValue("")

    self.txt_judul.SetValue("")

    self.cmbkategori.SetValue("")

    self.rYa.Value=False

    self.rTidak.Value=False

    # mengisi ke DateTimePicker tgl saat ini

    skrg = datetime.date.today()

    day = skrg.day

    month = skrg.month

    tahun = skrg.year

    displayed = wx.DateTimeFromDMY(day,month-
1,tahun)

    displayed.Format("%d/%m/%Y")

    self.tgl_masuk.SetValue(displayed)

    self.txt_stock.SetValue("")
```

```
    self.txt_tinggi.SetValue("")  
  
    self.txt_kdbuku.SetFocus()  
  
## Kembalikan gambar ke noimage  
  
filepath = 'C:\\Gambar\\noimage.jpg'  
  
img = wx.Image(filepath, wx.BITMAP_TYPE_ANY)  
  
self.gambar.SetBitmap(wx.BitmapFromImage(img))  
  
def Isi_List(self) :  
  
    self.lc.DeleteAllItems()  
  
    sql = "select * from buku order by kd_buku"  
  
    cur.execute(sql)  
  
    hasil = cur.fetchall()  
  
    jumbar = self.lc.GetItemCount()  
  
    for i in hasil :  
  
        self.lc.InsertStringItem(jumbar,i[0])  
  
        self.lc.SetStringItem(jumbar,1,i[1])  
  
        self.lc.SetStringItem(jumbar,2,str(i[5]))  
  
    jumbar = jumbar + 1  
  
def OnTmbBrowseButton(self, event):  
  
    wildcard = "JPEG files (*.jpg)|*.jpg"  
  
    dialog = wx.FileDialog(None, "Choose a file",
```

```
wildcard=wildcard,
style=wx.OPEN)

if dialog.ShowModal() == wx.ID_OK:
    self.txt_path.SetValue(dialog.GetPath())

dialog.Destroy()

filepath = self.txt_path.GetValue()
filepath = filepath.replace("\\\\",'\\\\')
self.txt_path.SetValue(filepath)

img = wx.Image(filepath, wx.BITMAP_TYPE_ANY)

self.gambar.SetBitmap(wx.BitmapFromImage(img))

def OnLcListItemSelected(self, event):
    a = event.m_itemIndex
    # mengambil no index baris yang dipilih
    b = self.lc.GetItem(a,0).GetText()
    # no index baris dikonversi ke text/ string
    self.txt_kdbuku.SetValue(b)
    self.Isi_Object()

def OnTmbBersihButton(self, event):
```

```
self.Awal()

def OnTmbHapusButton(self, event):
    if self.txt_nim.GetValue() == "":
        self.pesan = wx.MessageDialog \
            (self, "NIM belum diisi", "Konfirmasi", wx.OK)
        self.pesan.ShowModal()
        event.Skip()

    sql = " select * from biodata_radio where nim \\\n" + \
          "= '%s'" % (self.txt_nim.GetValue())
    cur.execute(sql)
    hasil= cur.fetchone()
    if cur.rowcount>0 :
        tanya = wx.MessageDialog(self, \
            message="Anda Yakin Hendak Menghapus NIM"\ \
            +" "+self.txt_nim.GetValue()+" "+\ \
            "Nama "+self.txt_nama.GetValue()"\ \
            ,style = wx.YES_NO)
        if tanya.ShowModal()==wx.ID_YES:
            sql = "delete from biodata_radio where nim \
\\
```

```
= "%s'" % (self.txt_nim.GetValue())  
  
        cur.execute(sql)  
  
    else :  
  
        self.pesan = wx.MessageDialog(self,\n  
        "NIM yang akan dihapus tidak terdata\\  
        di database","Konfirmasi",wx.OK)  
  
        self.pesan.ShowModal()  
  
    self.Awal()
```

BAB XI

MENCETAK LAPORAN KE EXCEL

DENGAN XLWT

11.1 Berkenalan dengan XLWT

XLWT adalah modul/ kumpulan library tambahan untuk menghasilkan laporan ke Microsoft Excel. Programmer bisa memanfaatkan XLWT untuk:

- Membuat Workbook dan Sheet baru
- Memindahkan data dari Frame (TextCtrl, ListCtrl, variabel, dsb) ke cell tertentu.
- Mengatur Style/ format (lebar kolom, lebar baris, border, shading, dan semisalnya)
- Menghasilkan formula tertentu pada cell (SUM, COUNT, dan sebagainya)
- Menempatkan file gambar pada cell tertentu.
- dan semisalnya...

Modul XLWT untuk Windows sudah ada dalam CD yang disertakan dalam buku ini. Anda bisa menginstallnya. Bisa juga dengan mengcopykan folder xlwt-0.7.2 ke C:\Python25\Lib\site-packages\ (dengan asumsi IDLE Python diinstall ke alamat folder default).

11.2 PERINTAH DASAR YANG SERING DIGUNAKAN DALAM XLWT

1. Import modul xlwt

```
from xlwt import *
```

2. Menciptakan Workbook baru

```
book = Workbook()
```

3. Menambahkan sheet baru pada workbook yang sudah dibuat

```
sheet1 = book.add_sheet('Sheet1')
```

4. Menuliskan data pada cell tertentu

```
sheet1.write(0,1,"Nama Barang")
```

Perintah tersebut menuliskan ‘Nama Barang’ ke baris ke-0 dan kolom ke-1. Index kolom dan baris dimulai dari 0.

5. Mengatur lebar kolom

```
sheet1.col(0).width = 5000
```

Perintah tersebut mengatur lebar kolom ke-0 menjadi 5000.

6. Menyimpan file yang dihasilkan ke alamat tertentu

```
path1 = 'C:\cetak.xls'
```

```
book.save(path1)
```

11.3 PERINTAH MENGHAPUS FILE LAMA DAN MELUNCURKAN FILE BARU

Ketika mencetak laporan dalam aplikasi yang dirancang, user akan mendapat tampilan hasilnya dalam Excel. Ia kemudian bisa menyesuaikan data yang ada untuk dioperasikan lagi dengan formula, atau langsung mencetak ke printer.

Proses yang berlangsung adalah : hapus file lama jika ada dan ciptakan file baru di lokasi folder yang sama. Kedua proses ini menggunakan modul Sistem Operasi (os).

Kode utama untuk melakukan proses-proses ini dengan modul **os** adalah:

1. Import modul os

```
import os
```

2. Cek apakah ada file dengan nama dan alamat tertentu di folder tertentu. Jika ada, hapus file lama.

```
path1 = 'C:\\cetak.xls'  
if os.path.exists(path1) :  
    os.remove(path1)
```

3. Simpan file baru ke path yang sudah ditentukan.

Telah dibahas di atas (menggunakan modul xlwt)

4. Luncurkan file yang baru tersimpan

```
os.system("start excel.exe C:\\cetak.xls")
```

Perintah `os.system()` pada dasarnya adalah mengeksekusi file executable. Dalam contoh, file yang dieksekusi adalah **excel.exe**. Sebenarnya semua file exe bisa dieksekusi dengan `os.system()`

11.4 Perintah Lain XLWT untuk Memformat Style dan Menampilkan Gambar

1. Mengatur font yang dipakai

Pengaturan font meliputi setting font dan pengalokasiannya pada Style

- a. Pengaturan font

```
font0 = Font()  
font0.name = 'Times New Roman'  
font0.height=200  
font0.bold = True
```

Font diatur dengan jenis ‘Times New Roman’, tinggi 200 dan disetting Bold (cetak tebal).

Tinggi 200 dalam setting normal adalah sama dengan 10 (didapatkan dari 200/20). Angka 20 adalah konstanta. Misalnya, jika anda ingin ukuran font adalah 20, maka setting **height-nya** adalah $20 \times 20 = 400$.

- b. Pendefinisan style dan pendaftaran setting font pada Style

```
style0 = XFStyle()  
style0.font = font0
```

- c. Implementasi style pada cell tertentu

```
ws0.write(1, 1, 'Test', style0)
```

ws0 adalah nama Workbook, sedangkan style0 adalah nama style yang telah didefinisikan sebelumnya.

2. Mengatur Border (garis di sekitar cell)

Berikut ini adalah kode untuk membuat border di sekitar cell

```
## setting border  
borders = Borders()  
borders.left = 1  
borders.right = 1  
borders.top = 1  
borders.bottom = 1  
# Definisikan Style  
style0 = XFStyle()  
# Mendaftarkan setting border pada style  
style0.borders = borders  
#Membuat workbook dan sheet telah dijelaskan di  
#atas....  
# Menuliskan ke dalam cell sesuai style  
ws0.write(1, 1, 'Test', style0)
```

3. Mengatur Shading/ Warna BackGround cell

Kode berikut ini untuk membuat background cell menjadi abu-abu (grey)

```
# Membuat Pattern  
BkgPat = Pattern()
```

```
BkgPat.pattern = Pattern.SOLID_PATTERN  
BkgPat.pattern_fore_colour = 22  
  
# Mendefinisikan Style  
  
style0 = XFStyle()  
  
# Setting Style pada Pattern yang telah didefinisikan  
style0.pattern = BkgPat  
  
# Membuat workbook dan sheet telah dijelaskan di  
  
# atas.....  
# Menuliskan ke cell (1,1)  
ws0.write(1, 1, 'Test', style0)
```

4. Memasukkan Gambar pada cell

Ketentuan:

- File gambar (*image*) yang akan dimasukkan dalam cell diletakkan dalam folder yang sama dengan file excel yang dihasilkan
- File gambar harus berekstensi **.bmp**.

Contoh kode :

```
ws.insert_bitmap('python.bmp', 2, 2)
```

Kode tersebut berarti memasukkan gambar **python.bmp** ke cell (2,2) pada worksheet bernama **ws**.

5. Memasukkan formula ke dalam cell tertentu

Contoh kode :

```
# Mengalikan nilai di A1 dengan B1  
ws.write(0, 2, Formula("A1*B1"))  
  
# Contoh formula IF  
ws.write(12, 3, Formula('IF(A1>A2;10;0)'))  
  
# Contoh formula SUM  
ws.write(6, 3, Formula("SUM($A$1:$C$5)"))
```

ws adalah nama worksheet.

6. Memformat data angka pada cell

Kode contoh :

```
fmt ='0.00'  
  
style = XFStyle()  
  
style.num_format_str = fmt
```

Kode di atas menghasilkan format angka dengan 2 angka di belakang koma

7. Memformat data tanggal pada cell

Kode contoh :

```
from datetime import datetime  
  
fmt ='DD-MM-YYYY'  
  
style = XFStyle()  
  
style.num_format_str = fmt
```

```
ws.write(i, 4, datetime.now(), style)
```

Format pada kode di atas adalah 2 digit tanggal-2 digit bulan-4 digit tahun.

Contoh – contoh kode pembuatan laporan ke Excel bisa dilihat lebih jauh di BAB XIII Aplikasi Penjualan

BAB XII

APLIKASI KAMUS BILINGUAL

MADURA-INGGRIS

12.1 Pendahuluan

BAB XII dan BAB XIII adalah pembahasan contoh kasus aplikasi. Aplikasi yang dibahas di bab ini adalah tentang kamus bilingual Madura-Inggris dan Inggris-Madura.

Struktur aplikasi ini terdiri dari :

1. wx.App (AppKamus.py)
2. FrmMenu.py : Frame Menu
3. FrmInput.py : Frame untuk input data kosa kata Madura dan Inggris
4. FrmKamus.py : Frame untuk menggunakan kamus Madura-Inggris atau Inggris-Madura

12.2 LISTING KODE wx.App (AppKamus.py)

```
#!/usr/bin/env python  
  
#Boa:App:BoaApp  
  
  
import wx
```

```
import FrmMenu

modules = {u'FrmMenu': [1, 'Main frame of Application',
u'FrmMenu.py']}

class BoaApp(wx.App):

    def OnInit(self):
        self.main = FrmMenu.create(None)
        self.main.Show()
        self.SetTopWindow(self.main)
        return True

def main():
    application = BoaApp(0)
    application.MainLoop()

if __name__ == '__main__':
    main()
```

12.3 STRUKTUR DATABASE DAN TABEL

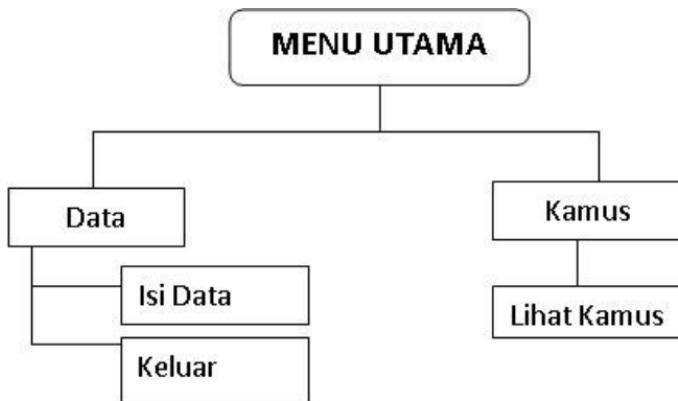
Aplikasi ini hanya terdiri dari 1 database dan 1 tabel.

Nama Database : kamus

Nama Tabel : kata

Field	Tipe	Lebar
madura	VarChar	40
inggris	VarChar	40

12.4 STRUKTUR MENU



Gambar 12.1 Struktur Menu Kamus

Isi Data : FrmInput.py

Lihat Kamus : FrmKamus.py

LISTING KODE MENU (FrmMenu.py)

```
#Boa:Frame:Frame1
```

```
import wx, FrmInput, FrmKamus

def create(parent):
    return Frame1(parent)

[wxID_FRAME1] = [wx.NewId() for _init_ctrls in range(1)]


class Frame1(wx.Frame):
    def __init__(self, prnt):
        # generated method, don't edit
        wx.Frame.__init__(self, id=wxID_FRAME1, name='',
                          parent=prnt,
                          pos=wx.Point(-4, -4), size=wx.Size(1032, 746),
                          style=wx.DEFAULT_FRAME_STYLE, title='Menu Utama')
        self.SetClientSize(wx.Size(1024, 712))

    def __init__(self, parent):
        self._init_ctrls(parent)
        menuBar = wx.MenuBar()
        menuData =wx.Menu()
        menuData.Append(1,"&Isi Data")
```

```
menuData.AppendSeparator()

menuData.Append(2,"&Keluar")

menuLihat = wx.Menu()

menuLihat.Append(3,"&Lihat Kamus")

menuBar.Append(menuData,"&Data")

menuBar.Append(menuLihat,"&Kamus")

self.SetMenuBar(menuBar)

self.Bind(wx.EVT_MENU, self.OnInput, id =1)

self.Bind(wx.EVT_MENU, self.OnKeluar, id =2)

self.Bind(wx.EVT_MENU, self.OnLihat, id =3)

def OnInput(self,event) :

    self.main = FrmInput.create(None)

    self.main.Show()

def OnKeluar (self,event) :

    self.Destroy()

def OnLihat (self,event) :

    self.main = FrmKamus.create(None)

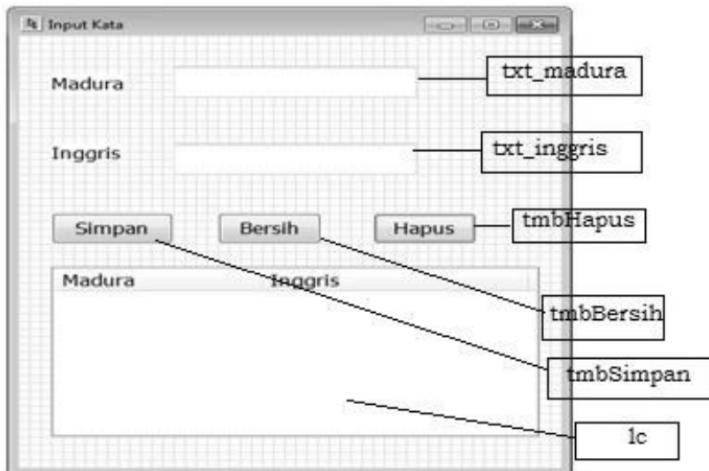
    self.main.Show()
```

12.5 Frame Isi Data (FrmlInput.py)

Frame ini berfungsi untuk menginputkan kata dalam bahasa Madura dan terjemahannya dalam bahasa Inggris atau sebaliknya.



Gambar 12.2 Frame Input Kata



Gambar 12.3 Nama Tiap Komponen Frame Input Kata

LISTING KODE FRAME INPUT KATA (FrmlInput.py)

```
#Boa:Frame:Frame1

import wx, MySQLdb

# Buat koneksi

conn
=MySQLdb.connect(host="localhost",user="root",passwd="",
,db = "kamus")

# Buat kurSOR

kurSOR = conn.cursor()
```

```
def create(parent):
    return Frame1(parent)

[wxID_FRAME1, wxID_FRAME1LC,
wxID_FRAME1PANEL1, wxID_FRAME1STATICTEXT1,
wxID_FRAME1STATICTEXT2,
wxID_FRAME1TMBBERSIH, wxID_FRAME1TMBHAPUS,
wxID_FRAME1TMBSIMPAN,
wxID_FRAME1TXT_INGGRIS,
wxID_FRAME1TXT_MADURA,
] = [wx.NewId() for _init_ctrls in range(10)]


class Frame1(wx.Frame):
    def __init__(self, parent):
        # generated method, don't edit

        parent.InsertColumn(col=0,
format=wx.LIST_FORMAT_LEFT, heading='Madura',
width=150)

        parent.InsertColumn(col=1,
format=wx.LIST_FORMAT_LEFT,
heading='Inggris', width=192)
```

```

def _init_ctrls(self, prnt):

    # generated method, don't edit

    wx.Frame.__init__(self, id=wx.ID_FRAME1, name="",
parent=prnt,

        pos=wx.Point(407, 224), size=wx.Size(400, 483),
        style=wx.DEFAULT_FRAME_STYLE, title='Input
Kata ')

    self.SetClientSize(wx.Size(392, 449))

    self.SetBackgroundColour(wx.Colour(255, 255, 255))

    selfSetFont(wx.Font(12, wx.SWISS, wx.NORMAL,
wx.NORMAL, False,
'Tahoma'))


    self.panel1 = wx.Panel(id=wx.ID_FRAME1PANEL1,
name='panel1', parent=self,
        pos=wx.Point(0, 0), size=wx.Size(392, 449),
        style=wx.TAB_TRAVERSAL)

    self.staticText1 =
wx.StaticText(id=wx.ID_FRAME1STATICTEXT1,
        label='Madura', name='staticText1',
parent=self.panel1,

```

```

        pos=wx.Point(24,    40),    size=wx.Size(52,    19),
style=0)

        self.staticText1.SetFont(wx.Font(12,           wx.SWISS,
wx.NORMAL, wx.NORMAL,
False, 'Tahoma'))


self.txt_madura =  

wx.TextCtrl(id=wxID_FRAME1TXT_MADURA,  

name='txt_madura',           parent=self.panel1,  

pos=wx.Point(112, 32),  

size=wx.Size(176,           32),  

style=wx.TE_PROCESS_ENTER, value='')

        self.txt_madura.SetFont(wx.Font(12,           wx.SWISS,
wx.NORMAL, wx.NORMAL,
False, 'Tahoma'))


        self.txt_madura.Bind(wx.EVT_TEXT_ENTER,  

self.OnTxt_maduraTextEnter,  

id=wxID_FRAME1TXT_MADURA)


self.staticText2 =  

wx.StaticText(id=wxID_FRAME1STATICTEXT2,  

label='Inggris',           name='staticText2',
parent=self.panel1,  

pos=wx.Point(24,    112),    size=wx.Size(50,    19),
style=0)

```

```

        self.staticText2.SetFont(wx.Font(12,         wx.SWISS,
wx.NORMAL, wx.NORMAL,
                           False, 'Tahoma'))



        self.txt_inggris =  

wx.TextCtrl(id=wxID_FRAME1TXT_INGGRIS,  

           name='txt_inggris',           parent=self.panel1,  

pos=wx.Point(112, 112),  

           size=wx.Size(176, 32), style=0, value="")  

        self.txt_inggris.SetFont(wx.Font(12,         wx.SWISS,
wx.NORMAL, wx.NORMAL,
                           False, 'Tahoma'))



        self.tmbSimpan =  

wx.Button(id=wxID_FRAME1TMBSIMPAN,  

label='&Simpan',  

           name='tmbSimpan',           parent=self.panel1,  

pos=wx.Point(24, 184),  

           size=wx.Size(88, 32), style=0)  

        self.tmbSimpan.Bind(wx.EVT_BUTTON,  

self.OnTmbSimpanButton,  

                           id=wxID_FRAME1TMBSIMPAN)

        self.tmbBersih =  

wx.Button(id=wxID_FRAME1TMBBERSIH, label='Bersih',

```

```

        name='tmbBersih',           parent=self.panel1,
pos=wx.Point(144, 184),
size=wx.Size(75, 32), style=0)

    self.tmbBersih.Bind(wx.EVT_BUTTON,
self.OnTmbBersihButton,
id=wxID_FRAME1TMBBERSIH)

        self.tmbHapus
wx.Button(id=wxID_FRAME1TMBHAPUS, label='Hapus',
name='tmbHapus',           parent=self.panel1,
pos=wx.Point(256, 184),
size=wx.Size(75, 32), style=0)

    self.tmbHapus.Bind(wx.EVT_BUTTON,
self.OnTmbHapusButton,
id=wxID_FRAME1TMBHAPUS)

        self.lc = wx.ListCtrl(id=wxID_FRAME1LC, name='lc',
parent=self.panel1,
pos=wx.Point(24, 240), size=wx.Size(352, 176),
style=wx.LC_REPORT)

    self._init_lc_Columns(self.lc)

    self.lc.Bind(wx.EVT_LIST_ITEM_SELECTED,
self.OnLcListItemSelected,
id=wxID_FRAME1LC)

```

```
def __init__(self, parent):
    self._init_ctrls(parent)
    self.Isi_List()

def Isi_List(self) :
    self.lc.DeleteAllItems()
    sql = " select * from kata order by madura"
    kurSOR.execute(sql)
    hasil = kurSOR.fetchall()
    baris = self.lc.GetItemCount()
    for i in hasil :
        self.lc.InsertStringItem(baris,"%s"%i[0])
        self.lc.SetStringItem(baris,1,"%s"%i[1])
        baris =baris + 1

def OnTmbSimpanButton(self, event):
    sql = " select * from kata where madura ='%s'%
(self.txt_madura.GetValue())
    kurSOR.execute(sql)
    hasil = kurSOR.fetchall()
```

```
if kurSOR.RowCount > 0 :  
    sql1 = "update kata set inggris = '%s' where  
madura = '%s'  
"%(self.txt_Inggris.GetValue(),self.txt_madura.GetValue())  
  
    kurSOR.execute(sql1)  
  
    conn.commit()  
  
else :  
  
    sql1 = "insert into kata (madura,inggris) values  
(''%s'',''%s'')  
"%(self.txt_madura.GetValue(),self.txt_Inggris.GetValue())  
  
    kurSOR.execute(sql1)  
  
    conn.commit()  
  
    self.Bersih()  
  
    self.Isi_List()  
  
def Bersih(self) :  
    self.txt_madura.SetValue("")  
    self.txt_Inggris.SetValue("")  
    self.txt_madura.SetFocus()  
  
  
  
def OnTmbBersihButton(self, event):  
    self.Bersih()  
  
  
  
def OnTmbHapusButton(self, event):
```

```
    sql1 = "delete from buku where madura ='%s'"%(self.txt_madura.GetValue())
    kurSOR.execute(sql1)
    conn.commit()
    self.Bersih()

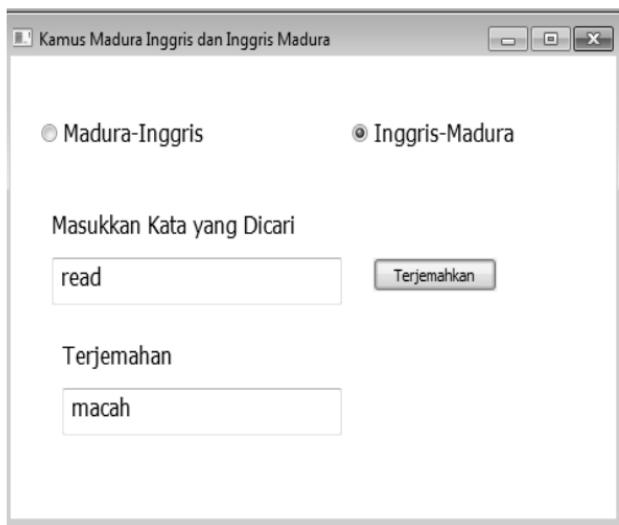
def OnLcListItemSelected(self, event):
    self.a = event.m_itemIndex
    md1 = self.lc.GetItem(self.a,0).GetText()
    self.txt_madura.SetValue(md1)
    ing1 = self.lc.GetItem(self.a,1).GetText()
    self.txt_inggris.SetValue(ing1)

def OnTxt_maduraTextEnter(self, event):
    sql = "select * from kata where madura ='%s'"%(self.txt_madura.GetValue())
    kurSOR.execute(sql)
    if kurSOR.rowCount > 0 :
        hasil = kurSOR.fetchall()
        for i in hasil :
            self.txt_inggris.SetValue(i[1])
    else :
```

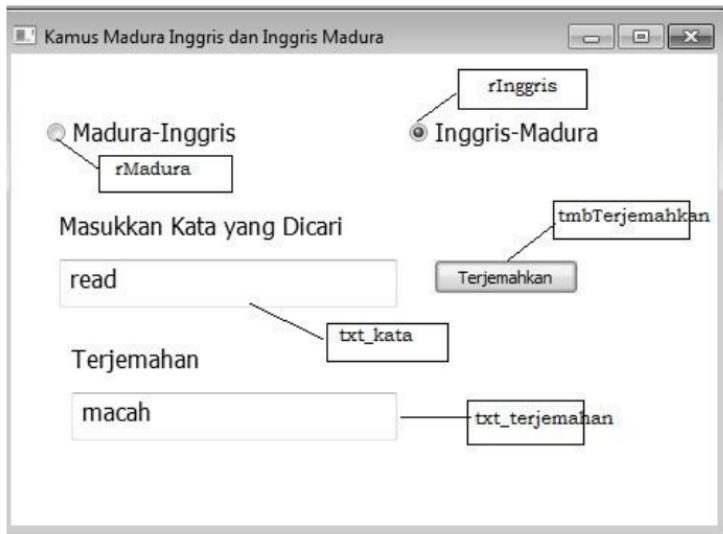
```
    self.pesan = wx.MessageDialog(self, "terjemahan  
kata "+self.txt_inggris.GetValue()+" tidak  
ada","Informasi",wx.OK)  
  
    self.pesan.ShowModal()
```

12.6 Frame Kamus (FrmKamus.py)

User bisa memilih jenis kamus apakah Madura-Inggris atau Inggris-Madura. Kemudian, ia memasukkan kata yang akan diterjemahkan, mengklik tombol Terjemahan, maka akan ditampilkan hasil terjemahannya (jika ada dalam database).



Gambar 12.4 Frame Kamus



Gambar 12.5 Komponen Frame Kamus

LISTING KODE FRAME KAMUS (FrmKamus.py)

```
#Boa:Frame:Frame1

import wx, MySQLdb

# Buat koneksi

conn
=MySQLdb.connect(host="localhost",user="root",passwd="",
,db = "kamus")

# Buat kurSOR

kurSOR = conn.cursor()
```

```
def create(parent):
    return Frame1(parent)

[wxID_FRAME1, wxID_FRAME1PANEL1,
wxID_FRAME1RINGGRIS, wxID_FRAME1RMADURA,
wxID_FRAME1STATICTEXT1,
wxID_FRAME1STATICTEXT2,
wxID_FRAME1TMBTERJEMAHKAN,
wxID_FRAME1TXT_KATA,
wxID_FRAME1TXT_TERJEMAHAN,
] = [wx.NewId() for _init_ctrls in range(9)]


class Frame1(wx.Frame):
    def __init__(self, prnt):
        # generated method, don't edit
        wx.Frame.__init__(self, id=wxID_FRAME1, name="",
parent=prnt,
pos=wx.Point(385, 232), size=wx.Size(484, 350),
style=wx.DEFAULT_FRAME_STYLE,
title='Kamus Madura Inggris dan Inggris
Madura')
        self.SetClientSize(wx.Size(468, 312))
        self.SetBackgroundColour(wx.Colour(255, 255, 255))
```

```

    self.panel1 = wx.Panel(id=wxID_FRAME1PANEL1,
name='panel1', parent=self,
pos=wx.Point(0, 0), size=wx.Size(468, 312),
style=wx.TAB_TRAVERSAL)

self.rMadura =
wx.RadioButton(id=wxID_FRAME1RMADURA,
label='Madura-Inggris', name='rMadura',
parent=self.panel1,
pos=wx.Point(24, 40), size=wx.Size(152, 24),
style=0)

self.rMadura.SetValue(True)

self.rMadura.SetFont(wx.Font(12, wx.SWISS,
wx.NORMAL, wx.NORMAL, False,
'Tahoma')))

self.rInggris =
wx.RadioButton(id=wxID_FRAME1RINGGRIS,
label='Inggris-Madura', name='rInggris',
parent=self.panel1,
pos=wx.Point(264, 40), size=wx.Size(136, 24),
style=0)

self.rInggris.SetValue(False)

```

```

        self.rInggris.SetFont(wx.Font(12,           wx.SWISS,
wx.NORMAL, wx.NORMAL, False,
'Tahoma'))



        self.staticText1 =  

wx.StaticText(id=wxID_FRAME1STATICTEXT1,  

label='Masukkan Kata yang Dicari',  

name='staticText1',  

parent=self.panel1,    pos=wx.Point(32,    104),  

size=wx.Size(188, 19),  

style=0)

        self.staticText1.SetFont(wx.Font(12,           wx.SWISS,
wx.NORMAL, wx.NORMAL,
False, 'Tahoma'))



        self.txt_kata =  

wx.TextCtrl(id=wxID_FRAME1TXT_KATA, name='txt_kata',
parent=self.panel1,    pos=wx.Point(32,    136),
size=wx.Size(224, 32),
style=0, value="")

        self.txt_kata.SetFont(wx.Font(12,           wx.SWISS,
wx.NORMAL, wx.NORMAL, False,
'Tahoma'))
```

```

    self.tmbTerjemahkan =
wx.Button(id=wxID_FRAME1TMBTERJEMAHKAN,
          label='Terjemahkan',      name='tmbTerjemahkan',
          parent=self.panel1,
          pos=wx.Point(280,  136),   size=wx.Size(96,  23),
          style=0)

    self.tmbTerjemahkan.Bind(wx.EVT_BUTTON,
self.OnTmbTerjemahkanButton,
          id=wxID_FRAME1TMBTERJEMAHKAN)

    self.staticText2 =
wx.StaticText(id=wxID_FRAME1STATICTEXT2,
              label='Terjemahan',           name='staticText2',
              parent=self.panel1,
              pos=wx.Point(40,   192),     size=wx.Size(85,   19),
              style=0)

    self.staticText2SetFont(wx.Font(12,         wx.SWISS,
wx.NORMAL, wx.NORMAL,
          False, 'Tahoma'))

    self.txt_terjemahan =
wx.TextCtrl(id=wxID_FRAME1TXT_TERJEMAHAN,
            name='txt_terjemahan',       parent=self.panel1,
            pos=wx.Point(40, 224),
            size=wx.Size(216, 32), style=0, value="")

```

```
    self.txt_terjemahan.SetFont(wx.Font(12,     wx.SWISS,
wx.NORMAL, wx.NORMAL,
False, 'Tahoma'))


def __init__(self, parent):
    self._init_ctrls(parent)

def OnTmbTerjemahkanButton(self, event):
    if self.rMadura.Value == True :
        sql = "select * from kata where madura ='%s'%
%(self.txt_kata.GetValue())
        kurSOR.execute(sql)
        if kurSOR.rowCount > 0 :
            hasil = kurSOR.fetchall()
            for i in hasil :
                self.txt_terjemahan.SetValue(i[1])
            else :
                self.pesan = wx.MessageDialog(self, "terjemahan
kata      "+self.txt_kata.GetValue()+"      tidak
ada","Informasi",wx.OK)
                self.pesan.ShowModal()
            else :
```

```
sql = "select * from kata where inggris ='%s'"%(self.txt_kata.GetValue())

kursor.execute(sql)

if kursor.rowcount > 0 :

    hasil = kursor.fetchall()

    for i in hasil :

        self.txt_terjemahan.SetValue(i[0])

    else :

        self.pesan = wx.MessageDialog(self, "terjemahan
kata      "+self.txt_kata.GetValue()+"      tidak
ada","Informasi",wx.OK)

        self.pesan.ShowModal()
```

BAB XIII

APLIKASI PENJUALAN

13.1 Pendahuluan

Aplikasi Penjualan yang dirancang ini memiliki kemampuan untuk :

1. Mendaftar barang (Kode Barang, Nama, Harga Beli, Harga Jual, Stock, dan Stock Min)
2. Melakukan Transaksi Penjualan Tunai (Kasir)
3. Mendapatkan Laporan Omzet pada Periode Tertentu
4. Mendapatkan Laporan Daftar Barang yang Di Bawah Stock Minimum

13.2 STRUKTUR FILE

File – file yang menyusun aplikasi ini adalah :

1. wx.App (AppPenjualan.py)
2. Frame Menu (frmMenu.py)
3. Frame Pendataan Barang (Dabar.py)
4. Frame Transaksi Penjualan (Jual.py)
5. Frame Laporan Omzet(LapPenjualan.py)

6. Frame Laporan Daftar Stock Minimum
(stock_min.py)

13.3 STRUKTUR DATABASE DAN TABEL

Aplikasi Penjualan ini terdiri dari 1 database dan 3 tabel : barang, jual, dan detail_jual.

Nama Database : **penjualan**

Nama Tabel : barang		
Field	Tipe	Lebar
kd_brg	Varchar	5
nama_brg	Varchar	25
hrg_beli	Int	
hrg_jual	Int	
stock	Int	
stock_min	Int	

Nama Tabel : jual		
Field	Tipe	Lebar
nota	int	
tgl	date	

total	int	
bayar	int	
kembali	Int	

Nama Tabel : detail_jual		
Field	Tipe	Lebar
Nota	int	
kd_brg	varchar	5
jml	int	
hrg_jual	int	
total	int	

13.4 LISTING KODE AppPenjualan.py

```
#!/usr/bin/env python

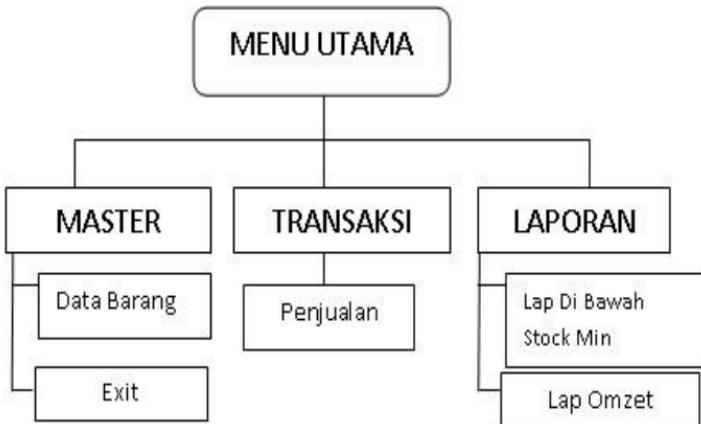
#Boa:App:BoaApp

import wx

import frmMenu
```

```
modules =[u'frmMenu': [1, 'Main frame of Application',  
u'frmMenu.py']]  
  
class BoaApp(wx.App):  
  
    def OnInit(self):  
  
        self.main = frmMenu.create(None)  
  
        self.main.Show()  
  
        self.SetTopWindow(self.main)  
  
        return True  
  
  
def main():  
  
    application = BoaApp(0)  
  
    application.MainLoop()  
  
  
if __name__ == '__main__':  
  
    main()
```

13.5 STRUKTUR MENU



Gambar 13.1 Struktur Menu Penjualan

LISTING KODE MENU (FrmMenu.py)

```

#Boa:Frame:frmMenu

import wx, Dabar, Jual, Stock_min, LapPenjualan

def create(parent):
    return frmMenu(parent)

[wxID_FRMMENU] = [wx.NewId() for _init_ctrls in range(1)]


class frmMenu(wx.Frame):
    def __init__(self, prnt):
        
```

```
# generated method, don't edit

wx.Frame.__init__(self, id=wx.ID_FRMMENU,
name='frmMenu', parent=prnt,
pos=wx.Point(-4, -4), size=wx.Size(1040, 759),
style=wx.DEFAULT_FRAME_STYLE, title='Menu
Utama Penjualan')

self.SetClientSize(wx.Size(1024, 721))
self.SetBackgroundColour(wx.Colour(255, 255, 255))

def __init__(self, parent):
    self._init_ctrls(parent)

    menuMaster = wx.Menu()
    menuMaster.Append(1, "&Data Barang...")
    menuMaster.AppendSeparator()
    menuMaster.Append(2, "E&xit")

    menuTrans = wx.Menu()
    menuTrans.Append(3, "&Penjualan...")

    menuLap = wx.Menu()
    menuLap.Append(4, "&Laporan di Bawah Stock
Min")
    menuLap.Append(5, "&Laporan Omzet")

    menuBar = wx.MenuBar()
```

```
menuBar.Append(menuMaster, "&Master")
menuBar.Append(menuTrans, "&Transaksi")
menuBar.Append(menuLap, "&Laporan")
self.SetMenuBar(menuBar)

self.CreateStatusBar()

self.SetStatusText("Selamat Datang di Python!")

self.Bind(wx.EVT_MENU, self.OnDabar, id=1)
self.Bind(wx.EVT_MENU, self.OnQuit, id=2)
self.Bind(wx.EVT_MENU, self.OnJual, id=3)
self.Bind(wx.EVT_MENU, self.OnLapStock_min, id=4)
self.Bind(wx.EVT_MENU, self.OnLapPenjualan, id=5)

def OnQuit(self, event):
    self.Close()

def OnDabar(self, event):
    self.main = Dabar.create(None)
    self.main.Show()

def OnJual(self, event):
    self.main = Jual.create(None)
    self.main.Show()
```

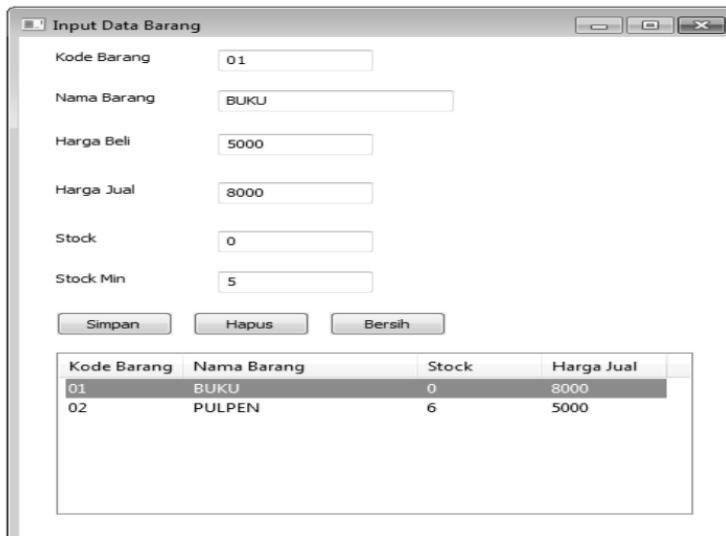
```

def OnLapStock_min(self,event) :
    self.main = Stock_min.create(None)
    self.main.Show()

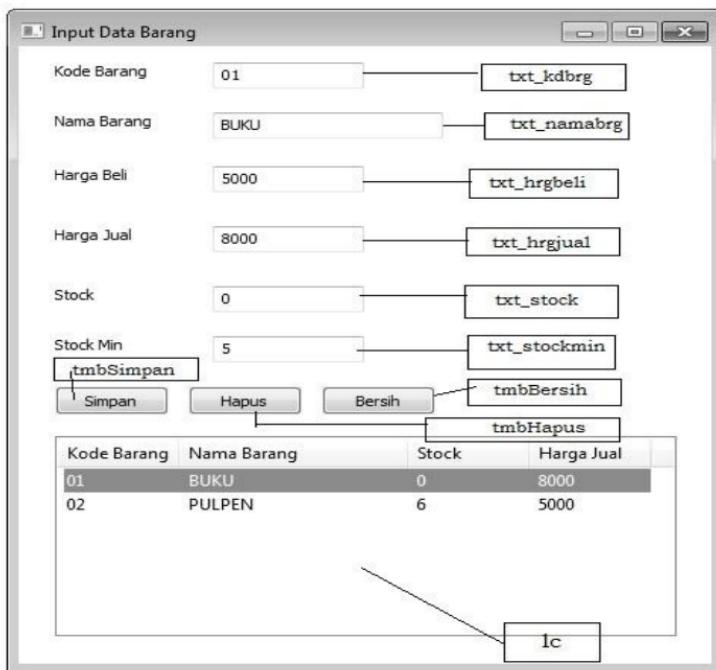
def OnLapPenjualan(self,event) :
    self.main = LapPenjualan.create(None)
    self.main.Show()

```

13.6 FRAME PENDATAAN BARANG (Dabar.py)



Gambar 13.2 Frame Input Data Barang



Gambar 13.3 Komponen Frame Input Data

LISTING KODE FRAME INPUT DATA (Dabar.py)

```
#Boa:Frame:Dabar

import wx, MySQLdb

conn= MySQLdb.connect(host="localhost", user="root",
passwd="",db="penjualan")

cur = conn.cursor()
```

```

def create(parent):
    return Dabar(parent)

[wxID_DABAR, wxID_DABARLC, wxID_DABARPANEL1,
wxID_DABARSTATICTEXT1,
wxID_DABARSTATICTEXT2, wxID_DABARSTATICTEXT3,
wxID_DABARSTATICTEXT4,
wxID_DABARSTATICTEXT5, wxID_DABARSTATICTEXT6,
wxID_DABARTMBBERSIH,
wxID_DABARTMBHAPUS,      wxID_DABARTMBSIMPAN,
wxID_DABARTXT_HRGBELI,
wxID_DABARTXT_HRGJUAL,   wxID_DABARTXT_KDBRG,
wxID_DABARTXT_NAMABRG,
wxID_DABARTXT_STOCK, wxID_DABARTXT_STOCKMIN,
] = [wx.NewId() for _init_ctrls in range(18)]


class Dabar(wx.Frame):
    def __init__(self, parent):
        # generated method, don't edit

        parent.InsertColumn(col=0,
format=wx.LIST_FORMAT_LEFT,
heading='Kode Barang', width=-1)

```

```

parent.InsertColumn(col=1,
format=wx.LIST_FORMAT_LEFT,
heading='Nama Barang', width=151)

parent.InsertColumn(col=2,
format=wx.LIST_FORMAT_LEFT, heading='Stock',
width=-1)

parent.InsertColumn(col=3,
format=wx.LIST_FORMAT_LEFT,
heading='Harga Jual', width=-1)

def __init__(self, prnt):
    # generated method, don't edit

    wx.Frame.__init__(self, id=wxID_DABAR,
name='Dabar', parent=prnt,
pos=wx.Point(397, 184), size=wx.Size(473, 532),
style=wx.DEFAULT_FRAME_STYLE, title='Input
Data Barang')

    self.SetClientSize(wx.Size(457, 494))

    self.SetBackgroundColour(wx.Colour(255, 255, 255))
    self.SetToolTipString('Input Data Barang')

    self.panel1 = wx.Panel(id=wxID_DABARPANEL1,
name='panel1', parent=self,
pos=wx.Point(0, 0), size=wx.Size(457, 494),

```

```

    style=wx.TAB_TRAVERSAL)

self.staticText1 =  

wx.StaticText(id=wxID_DABARSTATICTEXT1,  

              label='Kode Barang', name='staticText1',  

              parent=self.panel1,  

              pos=wx.Point(24, 13), size=wx.Size(61, 13),  

              style=0)

self.txt_kdbrg =  

wx.TextCtrl(id=wxID_DABARTXT_KDBRG,  

            name='txt_kdbrg',  

            parent=self.panel1, pos=wx.Point(128, 13),  

            size=wx.Size(100, 21),  

            style=wx.TE_PROCESS_ENTER, value="")  

            self.txt_kdbrg.Bind(wx.EVT_TEXT_ENTER,  

            self.OnTxt_kdbrgTextEnter,  

            id=wxID_DABARTXT_KDBRG)

self.staticText2 =  

wx.StaticText(id=wxID_DABARSTATICTEXT2,  

              label='Nama Barang', name='staticText2',  

              parent=self.panel1,  

              pos=wx.Point(24, 53), size=wx.Size(64, 13),  

              style=0)

```

```
    self.txt_namabrg =  
wx.TextCtrl(id=wxID_DABARTXT_NAMABRG,  
  
           name='txt_namabrg',          parent=self.panel1,  
           pos=wx.Point(128, 53),  
  
           size=wx.Size(152, 21), style=0, value='')  
  
  
    self.staticText3 =  
wx.StaticText(id=wxID_DABARSTATICTEXT3,  
  
           label='Harga Beli',      name='staticText3',  
           parent=self.panel1,  
  
           pos=wx.Point(24, 96),    size=wx.Size(48, 13),  
           style=0)  
  
  
    self.txt_hrgbeli =  
wx.TextCtrl(id=wxID_DABARTXT_HRGBELI,  
  
           name='txt_hrgbeli',        parent=self.panel1,  
           pos=wx.Point(128, 96),  
  
           size=wx.Size(100, 21), style=0, value='')  
  
  
    self.staticText4 =  
wx.StaticText(id=wxID_DABARSTATICTEXT4,  
  
           label='Harga Jual',       name='staticText4',  
           parent=self.panel1,  
  
           pos=wx.Point(24, 144),    size=wx.Size(51, 13),  
           style=0)
```

```
    self.txt_hrgjual =  
wx.TextCtrl(id=wxID_DABARTXT_HRGJUAL,  
  
               name='txt_hrgjual',           parent=self.panel1,  
               pos=wx.Point(128, 144),  
  
               size=wx.Size(100, 21), style=0, value="")  
  
  
  
    self.staticText5 =  
wx.StaticText(id=wxID_DABARSTATICTEXT5,  
  
               label='Stock',           name='staticText5',  
               parent=self.panel1,  
  
               pos=wx.Point(24, 192),   size=wx.Size(26, 13),  
               style=0)  
  
  
  
    self.txt_stock =  
wx.TextCtrl(id=wxID_DABARTXT_STOCK,  
name='txt_stock',  
  
               parent=self.panel1,     pos=wx.Point(128, 192),  
               size=wx.Size(100, 21),  
  
               style=0, value="")  
  
  
  
    self.staticText6 =  
wx.StaticText(id=wxID_DABARSTATICTEXT6,  
  
               label='Stock Min',      name='staticText6',  
               parent=self.panel1,  
  
               pos=wx.Point(24, 232),   size=wx.Size(45, 13),  
               style=0)
```

```

    self.txt_stockmin =  

wx.TextCtrl(id=wxID_DABARTXT_STOCKMIN,  

        name='txt_stockmin',           parent=self.panel1,  

pos=wx.Point(128, 232),  

        size=wx.Size(100, 21), style=0, value="")  

    self.tmbSimpan =  

wx.Button(id=wxID_DABARTMBSIMPAN, label='Simpan',  

        name='tmbSimpan',           parent=self.panel1,  

pos=wx.Point(24, 272),  

        size=wx.Size(75, 23), style=0)  

    self.tmbSimpan.Bind(wx.EVT_BUTTON,  

self.OnTmbSimpanButton,  

        id=wxID_DABARTMBSIMPAN)  

    self.tmbHapus =  

wx.Button(id=wxID_DABARTMBHAPUS, label='Hapus',  

        name='tmbHapus',           parent=self.panel1,  

pos=wx.Point(112, 272),  

        size=wx.Size(75, 23), style=0)  

    self.tmbHapus.Bind(wx.EVT_BUTTON,  

self.OnTmbHapusButton,  

        id=wxID_DABARTMBHAPUS)

```

```

    self.tmbBersih
wx.Button(id=wxID_DABARTMBBERSIH, label='Bersih',
          name='tmbBersih',           parent=self.panel1,
pos=wx.Point(200, 272),
          size=wx.Size(75, 23), style=0)

    self.tmbBersih.Bind(wx.EVT_BUTTON,
self.OnTmbBersihButton,
          id=wxID_DABARTMBBERSIH)

    self.lc = wx.ListCtrl(id=wxID_DABARLC, name='lc',
parent=self.panel1,
          pos=wx.Point(24, 312), size=wx.Size(410, 160),
          style=wx.LC_REPORT)

    self._init_lc_Columns(self.lc)

    self.lc.Bind(wx.EVT_LIST_ITEM_SELECTED,
self.OnLcListItemSelected,
          id=wxID_DABARLC)

def __init__(self, parent):
    self._init_ctrls(parent)
    self.Awal()

def OnTxt_kdbrgTextEnter(self, event):
    self.Isi_Object()

```

```
def OnTmbSimpanButton(self, event):

    if self.txt_kdbrg.GetValue() == "" :

        self.pesan = wx.MessageDialog(self,"Kode Barang
Belum diisi","Konfirmasi",wx.OK)

        self.pesan.ShowModal()

        self.txt_kdbrg.SetFocus()

        event.Veto()

    if self.txt_namabrg.GetValue() == "" :

        self.pesan = wx.MessageDialog(self,"Nama Barang
Belum diisi","Konfirmasi",wx.OK)

        self.pesan.ShowModal()

        self.txt_namabrg.SetFocus()

        event.Veto()

    sql = "select * from barang where kd_brg=%s"
"%(self.txt_kdbrg.GetValue())

    cur.execute(sql)

    kd_brg1=self.txt_kdbrg.GetValue()

    nama_brg1=self.txt_namabrg.GetValue()

    hrg_beli1=int(self.txt_hrgbeli.GetValue())

    hrg_jual1=int(self.txt_hrgjual.GetValue())
```

```

stock1=int(self.txt_stock.GetValue())
stock_min1=int(self.txt_stockmin.GetValue())
if cur.rowcount > 0 :
    sql = "update barang set nama_brg ='%s',
hrg_beli='%d', hrg_jual='%d', stock='%d', stock_min ='%d'
where kd_brg ='%s'
'%(nama_brg1,hrg_beli1,hrg_jual1,stock1,stock_min1,kd_
brg1)
else :
    sql      =      "insert      into      barang
(kd_brg,nama_brg,hrg_beli,hrg_jual,stock,stock_min)
values      ('%s','%s','%d','%d','%d','%d')
'%(kd_brg1,nama_brg1,hrg_beli1,hrg_jual1,stock1,stock_
min1)
cur.execute(sql)
conn.commit()
self.Awal()

def Awal(self) :
    self.Isi_List()
    self.txt_kdbrg.SetValue("")
    self.txt_namabrg.SetValue("")
    self.txt_hrgbeli.SetValue("")
    self.txt_hrgjual.SetValue("")

```

```
    self.txt_stock.SetValue("")  
    self.txt_stockmin.SetValue("")  
    self.txt_kdbrg.SetFocus()  
  
def Isi_Object(self) :  
  
    sql = "select * from barang where kd_brg = '%s' "%  
    %(self.txt_kdbrg.GetValue())  
  
    cur.execute(sql)  
  
    if cur.rowcount > 0 :  
  
        hasil = cur.fetchone()  
  
        self.txt_namabrg.SetValue(hasil[1])  
        self.txt_hrgbeli.SetValue(str(hasil[2]))  
        self.txt_hrgjual.SetValue(str(hasil[3]))  
        self.txt_stock.SetValue(str(hasil[4]))  
        self.txt_stockmin.SetValue(str(hasil[5]))  
  
    #else :  
  
    #    self.pesan = wx.MessageDialog(self,"Data Tidak  
    #Ada","Konfirmasi",wx.OK)  
  
    #    self.pesan.ShowModal()  
  
    self.txt_namabrg.SetFocus()  
  
def Isi_List(self) :  
  
    self.lc.DeleteAllItems()  
  
    sql = "select * from barang order by kd_brg"
```

```

cur.execute(sql)

hasil = cur.fetchall()

jumbar = self.lc.GetItemCount()

for i in hasil :

    self.lc.InsertStringItem(jumbar,i[0])

    self.lc.SetStringItem(jumbar,1,i[1])

    self.lc.SetStringItem(jumbar,2,str(i[4]))

    self.lc.SetStringItem(jumbar,3,str(i[3]))

    jumbar = jumbar + 1


def OnTmbHapusButton(self, event):

    if self.txt_kdbrg.GetValue() == "" :

        self.pesan = wx.MessageDialog(self,"Kode Barang
yang Akan Dihapus Belum diisi","Konfirmasi",wx.OK)

        self.pesan.ShowModal()

        self.txt_kdbrg.SetFocus()

        event.Veto()

        tanya = wx.MessageDialog(self,message="Anda yakin
handak                                         menghapus
barang"+self.txt_namabrg.GetValue()+"      ?",style =
wx.YES_NO)

        if tanya.ShowModal()==wx.ID_YES:

```

```
    sql ="delete from barang where kd_brg ='%s'  
"%(self.txt_kdbrg.GetValue())  
  
    cur.execute(sql)  
  
    conn.commit()  
  
    self.Awal()  
  
def OnTmbBersihButton(self, event):  
  
    self.Awal()  
  
  
def OnLcListItemSelected(self, event):  
  
    self.currentItem = event.m_itemIndex  
  
    # mengambil no index baris yang dipilih  
  
    b = self.lc.GetItem(self.currentItem).GetText()  
  
    # no index baris dikonversi ke text/ string  
  
    self.txt_kdbrg.SetValue(b)  
  
    self.Isi_Object()
```

13.7 Frame Penjualan

Transaksi Penjualan

No. Nota	Tanggal			
36	02/04/2012			
Kode Barang	Nama Barang	Jumlah	Harga	Total
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Tambah"/>				
Kode Barang	Nama Barang	Jumlah	Harga	Total
01	BUKU	3	8000	24000
02	PULPEN	6	5000	30000
03	TAS	3	30000	90000
04	SPIDOL	2	7000	14000

Total **158000** Bayar **160000** Kembali **2000**

Gambar 13.4 Transaksi Penjualan

Transaksi Penjualan

No. Nota	txt_nota		txt_tgl	Tanggal
36				02/04/2012
Kode Barang	txt_kdbrg	txt_namabrg	txt_jml	txt_hrg
		Nama Barang	Jumlah	Harga
			txt_jml1	txt_hrg
				txt_total
				tmbTambah
				Tambah

Kode Barang	Nama Barang	Jumlah	Harga	Total
01	BUKU	3	8000	24000
02	PULPEN	6	5000	30000
03	TAS	3	30000	90000
04	SPIDOL	2	7000	14000

le

Total	Bayar	Kembali
158000	160000	2000
<input type="button" value="Simpan"/>	txt_total_semu	txt_bayar
		txt_kembali

Gambar 13.5 Komponen Frame Penjualan

Proses yang terjadi pada Transaksi Penjualan adalah :

1. Saat Frame pertama kali dijalankan, akan muncul nomor nota dan tanggal secara otomatis.

Nomor nota mengacu pada nota terakhir+1 di tabel jual. Misalnya, di tabel jual nota terakhir

adalah 36, maka nota saat ini yang ditampilkan adalah 37.

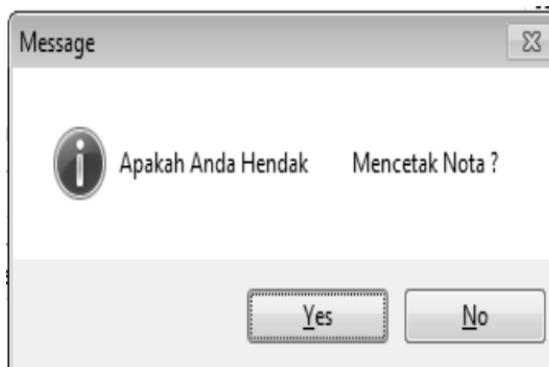
Tanggal yang ditampilkan adalah tanggal hari ini.

2. User memasukkan kode barang, kemudian menekan tombol enter, akan ditampilkan nama barang dan harga barang secara otomatis dengan mengacu pada tabel barang.
3. User memasukkan jumlah barang yang akan dibeli, dan menekan tombol enter, secara otomatis akan ditampilkan total ($\text{Total} = \text{harga} \times \text{jumlah}$)
4. User menekan tombol tambah, data item barang akan ditampilkan pada wx.ListCtrl **lc**, dan total harga yang harus dibayar ditampilkan di textctrl **txt_total_semua**. Deretan textctrl data item barang (**txt_kdbrg**, **txt_namabrg**, **txt_jml**, **txt_hrg**, dan **txt_total**) akan dibersihkan dan siap diisi data barang lain.

Proses nomor 1-4 berulang sesuai dengan banyaknya jenis barang yang ditransaksikan.

5. Jika semua jenis barang sudah diinput, maka user memasukkan nominal pembayaran di textctrl **txt_bayar** kemudian menekan tombol Enter. Nominal kembalian secara otomatis akan ditampilkan di **txt_kembali**.

6. User menekan tombol Simpan, maka akan terjadi proses perubahan pada database yang meliputi :
 - a. Menambah record baru di tabel jual
Satu transaksi satu record.
 - b. Menambah record baru di tabel detail_jual
Jumlah record yang ditambahkan sama dengan jumlah baris pada wx.ListCtrl **Ic**.
 - c. Mengurangi stock di tabel barang untuk barang-barang yang ada di wx.ListCtrl **Ic**.
7. Program akan menanyakan kepada user, apakah mau mencetak nota atau tidak. Jika user memilih tombol Yes, maka akan ditampilkan Nota dalam Microsoft Excel yang siap dicetak.



Gambar 13.6 Konfirmasi Cetak Nota

8. Proses 1 transaksi selesai, dan semua textctrl akan dibersihkan kecuali Nota dan Tanggal. Nota akan bertambah 1 angka.

	A	B	C	D	E
1		STRUK BELANJA TOKO SUKSES HUDAYA			
2	No. Nota	36			
3	Tanggal	02/04/2012			
4	Kode Barang	Nama Barang	Jumlah	Harga	Total
5	01	BUKU	3	8000	24000
6	02	PULPEN	6	5000	30000
7	03	TAS	3	30000	90000
8	04	SPIDOL	2	7000	14000
9					
10				Total Semua	158000
11				Bayar	160000
12				Kembali	2000
13					
14		TERIMA KASIH ATAS KUNJUNGAN ANDA			

Gambar 13.7 Tampilan Cetak Nota Penjualan

LISTING KODE FRAME PENJUALAN (Jual.py)

```
#Boa:Frame:Frame1

import wx, MySQLdb, datetime, os
from xlwt import *
conn= MySQLdb.connect(host="localhost", user="root",
passwd="",db="penjualan")
cur = conn.cursor()
```

```
def create(parent):
    return Frame1(parent)

[wxID_FRAME1, wxID_FRAME1LC,
wxID_FRAME1PANEL1, wxID_FRAME1STATICTEXT1,
wxID_FRAME1STATICTEXT10, wxID_FRAME1STATICTEXT11,
wxID_FRAME1STATICTEXT2,
wxID_FRAME1STATICTEXT3, wxID_FRAME1STATICTEXT4,
wxID_FRAME1STATICTEXT5,
wxID_FRAME1STATICTEXT6, wxID_FRAME1STATICTEXT7,
wxID_FRAME1STATICTEXT8,
wxID_FRAME1STATICTEXT9, wxID_FRAME1TMBSIMPAN, wxID_FRAME1TMBTAMBAH,
wxID_FRAME1TXT_BAYAR, wxID_FRAME1TXT_HRG,
wxID_FRAME1TXT_JML,
wxID_FRAME1TXT_KDBRG,
wxID_FRAME1TXT_KEMBALI,
wxID_FRAME1TXT_NAMABRG,
wxID_FRAME1TXT_NOTA, wxID_FRAME1TXT_TGL,
wxID_FRAME1TXT_TOTAL,
wxID_FRAME1TXT_TOTAL_SEMUA,
] = [wx.NewId() for _init_ctrls in range(26)]
```

```
class Frame1(wx.Frame):
```

```
def __init__coll_lc_Columns(self, parent):  
    # generated method, don't edit  
  
        parent.InsertColumn(col=0,  
format=wx.LIST_FORMAT_LEFT,  
  
            heading='Kode Barang', width=-1)  
  
        parent.InsertColumn(col=1,  
format=wx.LIST_FORMAT_LEFT,  
  
            heading='Nama Barang', width=135)  
  
        parent.InsertColumn(col=2,  
format=wx.LIST_FORMAT_LEFT, heading='Jumlah',  
  
            width=-1)  
  
        parent.InsertColumn(col=3,  
format=wx.LIST_FORMAT_LEFT, heading='Harga',  
  
            width=-1)  
  
        parent.InsertColumn(col=4,  
format=wx.LIST_FORMAT_LEFT, heading='Total',  
  
            width=-1)  
  
  
def __init__ctrls(self, prnt):  
    # generated method, don't edit  
  
    wx.Frame.__init__(self, id=wx.ID_FRAME1, name='',  
parent=prnt,  
  
        pos=wx.Point(359, 228), size=wx.Size(587, 500),
```

```

        style=wx.DEFAULT_FRAME_STYLE,
title='Transaksi Penjualan')

self.SetClientSize(wx.Size(571, 462))

self.SetBackgroundColour(wx.Colour(255, 255, 255))

self.panel1 = wx.Panel(id=wxID_FRAME1PANEL1,
name='panel1', parent=self,
pos=wx.Point(0, 0), size=wx.Size(571, 462),
style=wx.TAB_TRAVERSAL)

self.staticText1 =
wx.StaticText(id=wxID_FRAME1STATICTEXT1,
label='No. Nota', name='staticText1',
parent=self.panel1,
pos=wx.Point(8, 16), size=wx.Size(43, 13),
style=0)

self.txt_nota =
wx.TextCtrl(id=wxID_FRAME1TXT_NOTA, name='txt_nota',
parent=self.panel1, pos=wx.Point(8, 40),
size=wx.Size(100, 21),
style=0, value="")

self.txt_nota.setEditable(False)

```

```

    self.staticText2 =  

wx.StaticText(id=wxID_FRAME1STATICTEXT2,  

             label='Tanggal',           name='staticText2',  

             parent=self.panel1,  

             pos=wx.Point(480,   17),   size=wx.Size(38,   13),  

             style=0)  

    self.txt_tgl = wx.TextCtrl(id=wxID_FRAME1TXT_TGL,  

name='txt_tgl',  

parent=self.panel1,      pos=wx.Point(464,      35),  

size=wx.Size(100, 21),  

style=0, value="")  

    self.txt_tgl.SetEditable(False)  

    self.staticText3 =  

wx.StaticText(id=wxID_FRAME1STATICTEXT3,  

             label='Kode     Barang',   name='staticText3',  

             parent=self.panel1,  

             pos=wx.Point(16,   96),   size=wx.Size(61,   13),  

             style=0)  

    self.txt_kdbrg =  

wx.TextCtrl(id=wxID_FRAME1TXT_KDBRG,  

name='txt_kdbrg',  

parent=self.panel1,      pos=wx.Point(16,      120),  

size=wx.Size(100, 21),

```

```

        style=wx.TE_PROCESS_ENTER, value=")

    self.txt_kdbrg.Bind(wx.EVT_TEXT_ENTER,
self.OnTxt_kdbrgTextEnter,
id=wxID_FRAME1TXT_KDBRG)

        self.staticText4 =
wx.StaticText(id=wxID_FRAME1STATICTEXT4,
label='Nama Barang', name='staticText4',
parent=self.panel1,
pos=wx.Point(136, 96), size=wx.Size(64, 13),
style=0)

        self.txt_namabrg =
wx.TextCtrl(id=wxID_FRAME1TXT_NAMABRG,
name='txt_namabrg', parent=self.panel1,
pos=wx.Point(128, 120),
size=wx.Size(104, 21), style=0, value=")

    self.txt_namabrg.setEditable(False)

        self.staticText5 =
wx.StaticText(id=wxID_FRAME1STATICTEXT5,
label='Jumlah', name='staticText5',
parent=self.panel1,
pos=wx.Point(256, 96), size=wx.Size(33, 13),
style=0)

```

```

        self.txt_jml = wx.TextCtrl(id=wxID_FRAME1TXT_JML,
name='txt_jml',
parent=self.panel1,    pos=wx.Point(240,    120),
size=wx.Size(64, 21),
style=wx.TE_PROCESS_ENTER, value="")

        self.txt_jml.Bind(wx.EVT_TEXT_ENTER,
self.OnTxt_jmlTextEnter,
id=wxID_FRAME1TXT_JML)

        self.staticText6 =
wx.StaticText(id=wxID_FRAME1STATICTEXT6,
label='Harga',           name='staticText6',
parent=self.panel1,
pos=wx.Point(328,    96),   size=wx.Size(29,    13),
style=0)

        self.txt_hrg =
wx.TextCtrl(id=wxID_FRAME1TXT_HRG, name='txt_hrg',
parent=self.panel1,    pos=wx.Point(312,    120),
size=wx.Size(72, 21),
style=0, value="")

        self.txt_hrg.setEditable(False)

```

```

    self.staticText7 =
wx.StaticText(id=wxID_FRAME1STATICTEXT7,
              label='Total',           name='staticText7',
              parent=self.panel1,
              pos=wx.Point(400, 96),   size=wx.Size(24, 13),
              style=0)

    self.txt_total =
wx.TextCtrl(id=wxID_FRAME1TXT_TOTAL,
            name='txt_total',
            parent=self.panel1,     pos=wx.Point(392, 120),
            size=wx.Size(80, 21),
            style=0, value="")

    self.txt_total.setEditable(False)

    self.tmbTambah =
wx.Button(id=wxID_FRAME1TMBTAMBAH,
          label='Tambah',
          name='tmbTambah',       parent=self.panel1,
          pos=wx.Point(480, 120),
          size=wx.Size(75, 23), style=0)

    self.tmbTambah.Bind(wx.EVT_BUTTON,
self.OnTmbTambahButton,
          id=wxID_FRAME1TMBTAMBAH)

```

```

    self.lc = wx.ListCtrl(id=wxID_FRAME1LC, name='lc',
parent=self.panel1,
    pos=wx.Point(16, 152), size=wx.Size(536, 152),
    style=wx.LC_REPORT)

    self._init_coll_lc_Columns(self.lc)

    self.lc.Bind(wx.EVT_LIST_ITEM_SELECTED,
self.OnLcListItemSelected,
    id=wxID_FRAME1LC)

    self.staticText8 =
wx.StaticText(id=wxID_FRAME1STATICTEXT8,
    label='Total', name='staticText8',
parent=self.panel1,
    pos=wx.Point(40, 312), size=wx.Size(47, 25),
style=0)

    self.staticText8.SetFont(wx.Font(16, wx.SWISS,
wx.NORMAL, wx.NORMAL,
    False, 'Tahoma'))


    self.txt_total_semua =
wx.TextCtrl(id=wxID_FRAME1TXT_TOTAL_SEMUA,
    name='txt_total_semua', parent=self.panel1,
pos=wx.Point(16, 336),
    size=wx.Size(176, 64), style=0, value="")

```

```

        self.txt_total_semua.SetFont(wx.Font(16,    wx.SWISS,
wx.NORMAL, wx.BOLD,
False, 'Tahoma'))


        self.txt_total_semua.setEditable(False)

self.staticText9 =
wx.StaticText(id=wxID_FRAME1STATICTEXT9,
label='Total',           name='staticText9',
parent=self.panel1,
pos=wx.Point(40,   312),  size=wx.Size(47,   25),
style=0)

        self.staticText9.SetFont(wx.Font(16,      wx.SWISS,
wx.NORMAL, wx.NORMAL,
False, 'Tahoma'))


self.staticText10 =
wx.StaticText(id=wxID_FRAME1STATICTEXT10,
label='Bayar',           name='staticText10',
parent=self.panel1,
pos=wx.Point(240,   312),  size=wx.Size(54,   23),
style=0)

        self.staticText10.SetFont(wx.Font(14,      wx.SWISS,
wx.NORMAL, wx.BOLD,
False, 'Tahoma'))

```

```

    self.txt_bayar =
wx.TextCtrl(id=wxID_FRAME1TXT_BAYAR,
name='txt_bayar',
parent=self.panel1,    pos=wx.Point(200,    336),
size=wx.Size(168, 64),
style=wx.TE_PROCESS_ENTER, value="")

    self.txt_bayar.SetFont(wx.Font(16,         wx.SWISS,
wx.NORMAL, wx.NORMAL,
False, 'Tahoma'))

    self.txt_bayar.Bind(wx.EVT_TEXT_ENTER,
self.OnTxt_bayarTextEnter,
id=wxID_FRAME1TXT_BAYAR)

    self.staticText11 =
wx.StaticText(id=wxID_FRAME1STATICTEXT11,
label='Kembali',           name='staticText11',
parent=self.panel1,
pos=wx.Point(424,  312),  size=wx.Size(77,  23),
style=0)

    self.staticText11SetFont(wx.Font(14,         wx.SWISS,
wx.NORMAL, wx.BOLD,
False, 'Tahoma'))

    self.txt_kembali =
wx.TextCtrl(id=wxID_FRAME1TXT_KEMBALI,

```

```

        name='txt_kembali',           parent=self.panel1,
pos=wx.Point(392, 336),
size=wx.Size(160, 64), style=0, value="")

        self.txt_kembali.SetFont(wx.Font(16,         wx.SWISS,
wx.NORMAL, wx.NORMAL,
False, 'Tahoma'))


self.tmbSimpan = wx.Button(id=wxID_FRAME1TMBSIMPAN, label='Simpan',
name='tmbSimpan',           parent=self.panel1,
pos=wx.Point(16, 416),
size=wx.Size(88, 24), style=0)

        self.tmbSimpan.Bind(wx.EVT_BUTTON,
self.OnTmbSimpanButton,
id=wxID_FRAME1TMBSIMPAN)

def __init__(self, parent):
    self._init_ctrls(parent)
    self.Awal()

def Awal (self) :
    # Isi No Nota
    default_value = 0
    sql ="select COALESCE(max(nota), 0) from jual"

```

```
#sql = "select max(nota) as nt from jual "
cur.execute(sql)

if cur.rowcount > 0 :

    hasil = cur.fetchone()

    jml = hasil[0] + 1

    self.txt_nota.SetValue(str(jml))

else :

    self.txt_nota.SetValue("1")

# Isi Tanggal Saat ini

skrg = datetime.date.today()

day = skrg.day

month = skrg.month

year = skrg.year

self.txt_tgl.SetValue("%02d/%02d/%4d" % (day,
month, year))

# Bersihkan seluruh text di atas ListCtrl

self.txt_kdbrg.SetValue("")

self.txt_namabrg.SetValue("")

self.txt_hrg.SetValue("")
```

```
    self.txt_jml.SetValue("")  
    self.txt_total.SetValue("")  
    self.txt_kdbrg.SetFocus()  
  
    # Bersihkan ListCtrl lc  
  
    self.lc.DeleteAllItems()  
  
    # Bersihkan textctrl di bawah ListCtrl  
  
    self.txt_total_semua.SetValue("0")  
    self.txt_bayar.SetValue("0")  
    self.txt_kembali.SetValue("0")  
  
  
def OnTmbTambahButton(self, event):  
    jumbar = self.lc.GetItemCount()  
  
    self.lc.InsertStringItem(jumbar,self.txt_kdbrg.GetValue())  
  
    self.lc.SetStringItem(jumbar,1,self.txt_namabrg.GetValue())  
    self.lc.SetStringItem(jumbar,2,self.txt_jml.GetValue())  
    self.lc.SetStringItem(jumbar,3,self.txt_hrg.GetValue())  
  
    self.lc.SetStringItem(jumbar,4,self.txt_total.GetValue())  
  
    # Tambahkan Total  
  
    ta = int(self.txt_total.GetValue())
```

```
tb = int(self.txt_total_semua.GetValue())

tb = tb + ta

self.txt_total_semua.SetValue(str(tb))

# Bersihkan seluruh text di atas ListCtrl

self.txt_kdbrg.SetValue("")

self.txt_namabrg.SetValue("")

self.txt_jml.SetValue("")

self.txt_hrg.SetValue("")

self.txt_total.SetValue("")

self.txt_kdbrg.SetFocus()

def OnTmbSimpanButton(self, event):

    # Simpan ke Tabel Jual

    nota1= int(self.txt_nota.GetValue())

    skrg = datetime.date.today()

    day = skrg.day

    month = skrg.month

    year = skrg.year
```

```

tgl1 = datetime.date(year,month,day)

total1 = int(self.txt_total_semua.GetValue())

bayar1 = int(self.txt_bayar.GetValue())

kembali1 = int(self.txt_kembali.GetValue())


sql = "insert into jual (nota,tgl,total,bayar,kembali)
values \
('%d','%s','%d','%d','%d')
"%(nota1,tgl1,total1,bayar1,kembali1)

cur.execute(sql)

conn.commit()

# Simpan ke Tabel Detail_Jual

i =0

jumbar = self.lc.GetItemCount()

while i <= jumbar-1 :

    kd_brg1 = self.lc.GetItem(i,0).GetText()

    jml1 = int(self.lc.GetItem(i,2).GetText())

    hrg_jual1 = int(self.lc.GetItem(i,3).GetText())

    total1 = int(self.lc.GetItem(i,4).GetText())

    sql      =      "insert      into      detail_jual
(nota,kd_brg,jml,hrg_jual,total) \
values ('%d','%s','%d','%d','%d') "% \

```

```

(nota1,kd_brg1,jml1,hrg_jual1,total1)

cur.execute(sql)

conn.commit()

i = i+1

# Update stock barang

i =0

jumbar = self.lc.GetItemCount()

while i <= jumbar-1 :

    kd_brg1 = self.lc.GetItem(i,0).GetText()

    jml1 = int(self.lc.GetItem(i,2).GetText())

    sql = "select stock from barang where kd_brg
=%s"%(kd_brg1)

    cur.execute(sql)

    if cur.rowcount > 0 :

        hasil = cur.fetchone()

        st1 = hasil[0] - jml1

        sql = "update barang set stock =%d' where \
kd_brg = %s"%(st1,kd_brg1)

        cur.execute(sql)

        conn.commit()

    i = i +1

```

```
tanya = wx.MessageDialog(self,message="Apakah  
Anda Hendak \  
Mencetak Nota"+self.txt_namabrg.GetValue()+"  
?",style = wx.YES_NO)  
  
if tanya.ShowModal()==wx.ID_YES:  
  
    self.Cetak()  
  
    self.Awal()  
  
def Cetak(self) :  
  
    #Buat Workbook book  
  
    book = Workbook()  
  
    #Buat Worksheet sheet1  
  
    sheet1 = book.add_sheet('Sheet1')  
  
  
    ## Setting Border  
  
    borders = Borders()  
  
    borders.left = 1  
  
    borders.right = 1  
  
    borders.top = 1  
  
    borders.bottom = 1  
  
  
    ## Buat dan setting Style style0  
  
    style0 = XFStyle()
```

```
style0.borders = borders

# Hitung Jumlah Baris
jumbar =self.lc.GetItemCount()

## Cetak Header
sheet1.write(0,1,"STRUK BELANJA TOKO SUKSES
HUDAYA")
sheet1.write(1,0,"No. Nota")
sheet1.write(1,1,self.txt_nota.GetValue())
sheet1.write(2,0,"Tanggal")
sheet1.write(2,1,self.txt_tgl.GetValue())

i = 3
# Beri Judul
sheet1.write(i,0,"Kode Barang",style0)
sheet1.write(i,1,"Nama Barang",style0)
sheet1.write(i,2,"Jumlah",style0)
sheet1.write(i,3,"Harga",style0)
sheet1.write(i,4,"Total",style0)

j=0
```

```
while j<=jumbar-1 :  
    ## Isikan Item Data Barang  
    kd_brg1 = self.lc.GetItem(j,0).GetText()  
    sheet1.write(j+i+1,0,kd_brg1,style0)  
    nama_brg1 = self.lc.GetItem(j,1).GetText()  
    sheet1.write(j+i+1,1,nama_brg1,style0)  
    hrg1= self.lc.GetItem(j,2).GetText()  
    sheet1.write(j+i+1,2,int(hrg1),style0)  
    jml1 = self.lc.GetItem(j,3).GetText()  
    sheet1.write(j+i+1,3,int(jml1),style0)  
    total1 = self.lc.GetItem(j,4).GetText()  
    sheet1.write(j+i+1,4,int(total1),style0)  
    j=j+1  
    k = j+i+2  
    sheet1.write(k,3,"Total Semua ",style0)  
  
    sheet1.write(k,4,int(self.txt_total_semua.GetValue()),style0)  
    sheet1.write(k+1,3,"Bayar ",style0)  
  
    sheet1.write(k+1,4,int(self.txt_bayar.GetValue()),style0)  
    sheet1.write(k+2,3,"Kembali ",style0)
```

```
sheet1.write(k+2,4,int(self.txt_kembali.GetValue()),style0)

    sheet1.write(k+4,1,"TERIMA      KASIH      ATAS
KUNJUNGAN ANDA")

# Atur Lebar Kolom

    sheet1.col(0).width = 3500
    sheet1.col(1).width = 5000
    sheet1.col(3).width = 4000

path1 = 'C:\\cetak.xls'

# Cek apakah ada file lama

if os.path.exists(path1) :

    # Hapus file lama

    os.remove(path1)

    # Simpan file baru

    book.save(path1)

    # Luncurkan file baru

    os.system("start excel.exe C:\\cetak.xls")

def OnTxt_kdbrgTextEnter(self, event):

    sql = "select * from barang where kd_brg = '%s' \\
% (self.txt_kdbrg.GetValue())
```

```
cur.execute(sql)

if cur.rowcount > 0 :

    hasil = cur.fetchone()

    self.txt_namabrg.SetValue(hasil[1])

    self.txt_hrg.SetValue(str(hasil[3]))

    self.txt_jml.SetFocus()

def OnTxt_jmlTextEnter(self, event):

    hrg = int(self.txt_hrg.GetValue())

    jml = int(self.txt_jml.GetValue())

    tt = hrg * jml

    self.txt_total.SetValue(str(tt))

    self.tmbTambah.SetFocus()

def OnTxt_bayarTextEnter(self, event):

    tt = int(self.txt_total_semua.GetValue())

    byr = int(self.txt_bayar.GetValue())

    kmb = byr - tt

    self.txt_kembali.SetValue(str(kmb))

def OnLcListItemSelected(self, event):
```

```
self.currentItem = event.m_itemIndex  
  
# mengambil no index baris yang dipilih  
  
b = self.lc.GetItem(self.currentItem).GetText()  
  
# no index baris dikonversi ke text/ string  
  
self.txt_kdbrg.SetValue(b)  
  
self.lc.DeleteItem(self.currentItem)
```

13.8 Laporan Data Barang di Bawah Stock Minimum

Setiap suatu barang masuk dalam transaksi penjualan, stocknya akan berkurang sesuai dengan jumlah yang dijual. Pada tahap awal telah ditetapkan stock minimal sebagai acuan bahwa jika stock kurang dari stock minimal, semestinya barang tersebut ditambah (dibeli lagi dari supplier).

Untuk memudahkan melihat daftar barang apa saja yang stocknya sudah di bawah stock minimum, dirancanglah Frame untuk menghasilkan laporan semacam itu (**stock_min.py**)

Laporan Daftar Barang di Bawah Stock Min

Kode Barang	Nama Barang	Stock	Stock Min	Harga Beli
02	PULPEN	4	5	3500
03	TAS	3	5	25000

Cetak

Gambar 13.8 Daftar Barang di Bawah Stock Min

1	DAFTAR BARANG DI BAWAH STOCK MINIMUM				
2	TOKO SUKSES HUDAYA				
3					
4	Kode Barang	Nama Barang	Stock	Stock Min	Harga Beli
5	02	PULPEN	4	5	3500
6	03	TAS	3	5	25000

Gambar 13.9 Tampilan Laporan Daftar Barang di Bawah Stock Minimum

LISTING KODE FRAME LAPORAN DAFTAR BARANG DI BAWAH STOCK MIN (stock_min.py)

```
#Boa:Frame:Frame1
```

```
import wx, MySQLdb, datetime, os
from xlwt import *
conn= MySQLdb.connect(host="localhost", user="root",
passwd="",db="penjualan")
cur = conn.cursor()

def create(parent):
    return Frame1(parent)

[wxID_FRAME1, wxID_FRAME1BUTTON1,
wxID_FRAME1LC, wxID_FRAME1PANEL1,
wxID_FRAME1TMBCETAK,
] = [wx.NewId() for _init_ctrls in range(5)]


class Frame1(wx.Frame):
    def __init__(self, parent):
        # generated method, don't edit

        parent.InsertColumn(col=0,
format=wx.LIST_FORMAT_LEFT,
heading='Kode Barang', width=105)
        parent.InsertColumn(col=1,
format=wx.LIST_FORMAT_LEFT,
```

```

        heading='Nama Barang', width=160)

    parent.InsertColumn(col=2,
format=wx.LIST_FORMAT_LEFT, heading='Stock',
width=65)

    parent.InsertColumn(col=3,
format=wx.LIST_FORMAT_LEFT,
heading='Stock Min', width=77)

    parent.InsertColumn(col=4,
format=wx.LIST_FORMAT_LEFT,
heading='Harga Beli', width=100)

def __init__(self, prnt):
    # generated method, don't edit

    wx.Frame.__init__(self, id=wx.ID_FRAME1, name='',
parent=prnt,
pos=wx.Point(379, 202), size=wx.Size(658, 485),
style=wx.DEFAULT_FRAME_STYLE,
title='Laporan Daftar Barang di Bawah Stock
Min')

    self.SetClientSize(wx.Size(642, 447))

    self.panel1 = wx.Panel(id=wx.ID_FRAME1_PANEL1,
name='panel1', parent=self,
pos=wx.Point(0, 0), size=wx.Size(642, 447),

```

```

        style=wx.TAB_TRAVERSAL)

    self.panel1.SetBackgroundColour(wx.Colour(192,
192, 192))

self.lc = wx.ListCtrl(id=wxID_FRAME1LC, name='lc',
parent=self.panel1,
pos=wx.Point(16, 8), size=wx.Size(600, 376),
style=wx.LC_REPORT)

self._init_coll_lc_Columns(self.lc)

self.button1 =
wx.Button(id=wxID_FRAME1BUTTON1, label='button1',
name='button1', parent=self.panel1,
pos=wx.Point(192, 216),
size=wx.Size(75, 23), style=0)

self.tmbCetak =
wx.Button(id=wxID_FRAME1TMBCETAK, label='Cetak',
name='tmbCetak', parent=self.panel1,
pos=wx.Point(512, 400),
size=wx.Size(96, 23), style=0)

self.tmbCetak.Bind(wx.EVT_BUTTON,
self.OnTmbCetakButton,
id=wxID_FRAME1TMBCETAK)

```

```
def __init__(self, parent):  
    self._init_ctrls(parent)  
  
    sql = "select * from barang where stock < stock_min "  
    cur.execute(sql)  
  
    hasil= cur.fetchall()  
  
    k =self.lc.GetItemCount()  
  
    for i in hasil :  
  
        self.lc.InsertStringItem(k,i[0])  
        self.lc.SetStringItem(k,1,i[1])  
        self.lc.SetStringItem(k,2,str(i[4]))  
        self.lc.SetStringItem(k,3,str(i[5]))  
        self.lc.SetStringItem(k,4,str(i[2]))  
  
    k = k + 1
```

```
def OnTmbCetakButton(self, event):  
    #Buat Workbook book  
    book = Workbook()  
    #Buat Worksheet sheet1  
    sheet1 = book.add_sheet('Sheet1')
```

```
## Setting Font  
font0 = Font()  
font0.name = 'Arial'  
font0.height=200  
font0.bold = True  
  
## Setting Border  
borders = Borders()  
borders.left = 1  
borders.right = 1  
borders.top = 1  
borders.bottom = 1  
  
## Setting Pattern  
BkgPat = Pattern()  
BkgPat.pattern = Pattern.SOLID_PATTERN  
BkgPat.pattern_fore_colour = 22
```

```
## Buat dan setting Style style0  
style0 = XFStyle()  
style0.font = font0  
style0.borders = borders  
style0.pattern = BkgPat  
  
## Buat dan setting Style style1  
style1 = XFStyle()  
style1.borders = borders  
  
## Buat dan setting Style style1  
style_judul = XFStyle()  
style_judul.font = font0  
  
# Hitung Jumlah Baris  
jumbar =self.lc.GetItemCount()  
  
## Cetak Header
```

```
sheet1.write(0,1,"DAFTAR BARANG DI BAWAH  
STOCK MINIMUM",style_judul)
```

```
sheet1.write(1,1,"TOKO SUKSES  
HUDAYA ",style_judul)
```

```
i = 3
```

```
# Beri Judul
```

```
sheet1.write(i,0,"Kode Barang",style0)
```

```
sheet1.write(i,1,"Nama Barang",style0)
```

```
sheet1.write(i,2,"Stock",style0)
```

```
sheet1.write(i,3,"Stock Min",style0)
```

```
sheet1.write(i,4,"Harga Beli",style0)
```

```
j=0
```

```
while j<=jumbar-1 :
```

```
## Isikan Item Data Barang
```

```
kd_brg1 = self.lc.GetItem(j,0).GetText()
```

```
sheet1.write(j+i+1,0,kd_brg1,style1)
```

```
nama_brg1 = self.lc.GetItem(j,1).GetText()
```

```
sheet1.write(j+i+1,1,nama_brg1,style1)
```

```
stock1= self.lc.GetItem(j,2).GetText()
```

```
sheet1.write(j+i+1,2,int(stock1),style1)
```

```
stock_min1 = self.lc.GetItem(j,3).GetText()
sheet1.write(j+i+1,3,int(stock_min1),style1)

hrg_beli1 = self.lc.GetItem(j,4).GetText()
sheet1.write(j+i+1,4,int(hrg_beli1),style1)

j=j+1

# Atur Lebar Kolom
sheet1.col(0).width = 3500
sheet1.col(1).width = 5000
sheet1.col(3).width = 4000
sheet1.col(4).width = 3500

path1 = 'C:\\cetak_stock_min.xls'

# Cek apakah ada file lama
if os.path.exists(path1) :
    # Hapus file lama
    os.remove(path1)

# Simpan file baru
book.save(path1)

# Luncurkan file baru
os.system("start excel.exe C:\\cetak_stock_min.xls")
```

13.9 Frame Laporan Omzet Penjualan Pada Periode Tertentu (LapPenjualan.py)

Laporan Penjualan

Dari: 01/04/2012 Sampai: 01/04/2012

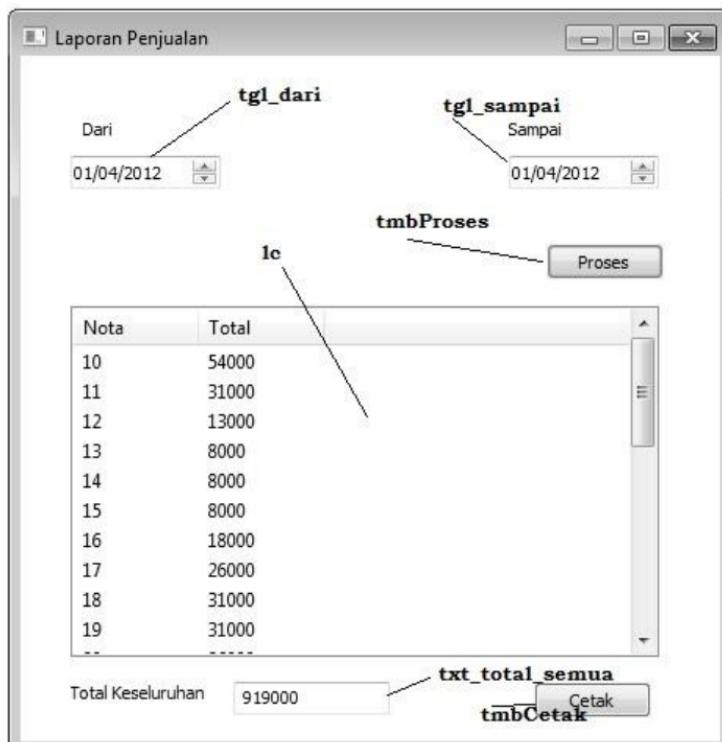
Proses

Nota	Total
10	54000
11	31000
12	13000
13	8000
14	8000
15	8000
16	18000
17	26000
18	31000
19	31000
--	-----

Total Keseluruhan: 919000

Cetak

Gambar 13.10 Frame Laporan Omzet Penjualan



Gambar 13.11 Komponen Laporan Penjualan

Proses Laporan omzet penjualan pada periode tertentu adalah :

1. User memilih tanggal 'dari' dan 'sampai' pada periode laporan
2. User menekan tombol Proses, hasilnya akan ditampilkan di wx.ListCtrl **lc** dalam bentuk 2 kolom: Nota dan Total.

Total Keseluruhan Omzet ditampilkan di textctrl **txt_total_semua**.

3. Selanjutnya, ditekan tombol Cetak, maka akan dihasilkan cetakan pada Excel

LAPORAN OMZET PENJUALAN TOKO SUKSES HUDAYA	
Dari Tgl	01-04-2012
Sampai Tgl	01-04-2012
Nota	Total Harga
10	Rp 54.000,00
11	Rp 31.000,00
12	Rp 13.000,00
13	Rp 8.000,00
14	Rp 8.000,00
15	Rp 8.000,00
16	Rp 18.000,00
17	Rp 26.000,00
18	Rp 31.000,00
19	Rp 31.000,00
20	Rp 26.000,00
21	Rp 16.000,00
22	Rp 26.000,00
23	Rp 26.000,00
24	Rp 21.000,00
25	Rp 26.000,00
26	Rp 26.000,00
27	Rp 26.000,00
28	Rp 31.000,00
29	Rp 58.000,00
30	Rp 136.000,00
31	Rp 18.000,00
32	Rp 118.000,00
33	Rp 50.000,00
34	Rp 61.000,00
35	Rp 26.000,00
Total Omzet	Rp 919.000,00

Gambar 13.12 Print Preview Laporan Omzet

LISTING KODE FRAME LAPORAN OMZET PENJUALAN (LapPenjualan.py)

```
#Boa:Frame:Frame1

import wx, MySQLdb, datetime, os
from xlwt import *
```

```
conn= MySQLdb.connect(host="localhost", user="root",
passwd="",db="penjualan")

cur = conn.cursor()

def create(parent):

    return Frame1(parent)

[wxID_FRAME1,
 wxID_FRAME1PANEL1, wxID_FRAME1STATICTEXT1,
 wxID_FRAME1STATICTEXT2,
 wxID_FRAME1STATICTEXT3, wxID_FRAME1TGL_DARI,
 wxID_FRAME1TGL_SAMPAI,  wxID_FRAME1TMBCETAK,
 wxID_FRAME1TMBPROSES,
 wxID_FRAME1TXT_TOTAL_SEMUA,
] = [wx.NewId() for _init_ctrls in range(11)]


class Frame1(wx.Frame):

    def __init__(self, parent):
        # generated method, don't edit

        parent.InsertColumn(col=0,
format=wx.LIST_FORMAT_LEFT, heading='Nota',
width=-1)
```

```

parent.InsertColumn(col=1,
format=wx.LIST_FORMAT_LEFT, heading='Total',
width=-1)

def _init_ctrls(self, prnt):
    # generated method, don't edit

    wx.Frame.__init__(self, id=wxID_FRAME1, name='',
parent=prnt,
pos=wx.Point(321, 204), size=wx.Size(463, 476),
style=wx.DEFAULT_FRAME_STYLE,
title='Laporan Penjualan')

    self.SetClientSize(wx.Size(447, 438))
    self.SetBackgroundColour(wx.Colour(255, 255, 255))

    self.panel1 = wx.Panel(id=wxID_FRAME1PANEL1,
name='panel1', parent=self,
pos=wx.Point(0, 0), size=wx.Size(447, 438),
style=wx.TAB_TRAVERSAL)

    self.tgl_dari
wx.DatePickerCtrl(id=wxID_FRAME1TGL_DARI,
name='tgl_dari', parent=self.panel1,
pos=wx.Point(32, 64),

```

```

        size=wx.Size(96,                                     21),
style=wx.DP_SHOWCENTURY)

        self.staticText1                                     =
wx.StaticText(id=wxID_FRAME1STATICTEXT1,
                label='Dari',                         name='staticText1',
parent=self.panel1,
                pos=wx.Point(40,    40),   size=wx.Size(19,    13),
style=0)

        self.staticText2                                     =
wx.StaticText(id=wxID_FRAME1STATICTEXT2,
                label='Sampai',                      name='staticText2',
parent=self.panel1,
                pos=wx.Point(312,   40),   size=wx.Size(34,    13),
style=0)

        self.tgl_sampai                                    =
wx.DatePickerCtrl(id=wxID_FRAME1TGL_SAMPAI,
                name='tgl_sampai',                  parent=self.panel1,
pos=wx.Point(312, 64),
                size=wx.Size(96,                                     21),
style=wx.DP_SHOWCENTURY)

        self.tmbProses                                     =
wx.Button(id=wxID_FRAME1TMBPROSES, label='Proses',

```

```

        name='tmbProses',           parent=self.panel1,
pos=wx.Point(336, 120),
size=wx.Size(75, 23), style=0)

self.tmbProses.Bind(wx.EVT_BUTTON,
self.OnTmbProsesButton,
id=wxID_FRAME1TMBPROSES)

self.lc = wx.ListCtrl(id=wxID_FRAME1LC, name='lc',
parent=self.panel1,
pos=wx.Point(32, 160), size=wx.Size(376, 224),
style=wx.LC_REPORT)

self._init_coll_lc_Columns(self.lc)

self.staticText3 =
wx.StaticText(id=wxID_FRAME1STATICTEXT3,
label='Total Keseluruhan', name='staticText3',
parent=self.panel1,
pos=wx.Point(32, 400), size=wx.Size(87, 13),
style=0)

self.txt_total_semua =
wx.TextCtrl(id=wxID_FRAME1TXT_TOTAL_SEMUA,
name='txt_total_semua', parent=self.panel1,
pos=wx.Point(136,
400), size=wx.Size(100, 21), style=0, value="")

```

```
    self.tmbCetak
wx.Button(id=wxID_FRAME1TMBCETAK, label='Cetak',
          name='tmbCetak', parent=self.panel1,
          pos=wx.Point(328, 400),
          size=wx.Size(75, 23), style=0)

    self.tmbCetak.Bind(wx.EVT_BUTTON,
self.OnTmbCetakButton,
          id=wxID_FRAME1TMBCETAK)

def __init__(self, parent):
    self._init_ctrls(parent)
    skrg = datetime.date.today()
    day = skrg.day
    month = skrg.month
    year = skrg.year
    displayed = wx.DateTimeFromDMY(day,month-1,year)
    displayed.Format("%d/%m/%Y")
    self.tgl_dari.SetValue(displayed)
    #self.tgl_sampai.SetValue(displayed)

def OnTmbProsesButton(self, event):
```

```
selected = self.tgl_dari.GetValue()

month = selected.Month + 1

day = selected.Day

year = selected.Year

dari1=datetime.date(year,month,day)

selected = self.tgl_sampai.GetValue()

month = selected.Month + 1

day = selected.Day

year = selected.Year

sampai1=datetime.date(year,month,day)

self.lc.DeleteAllItems()

sql = "select * from jual where tgl <=%s and tgl>=%s"
"%(sampai1,dari1)

cur.execute(sql)

if cur.rowcount > 0 :

    hasil = cur.fetchall()

    jumbar = self.lc.GetItemCount()

    total =0

    for i in hasil :

        self.lc.InsertStringItem(jumbar,str(i[0]))
```

```
        self.lc.SetStringItem(jumbar,1,str(i[2]))  
        total=total+i[2]  
  
        jumbar = jumbar + 1  
        self.txt_total_semua.SetValue(str(total))  
  
  
def OnTmbCetakButton(self, event):  
    #Buat Workbook book  
    book = Workbook()  
    #Buat Worksheet sheet1  
    sheet1 = book.add_sheet('Sheet1')  
  
  
    ## Setting Font  
    font0 = Font()  
    font0.name = 'Arial'  
    font0.height=200  
    font0.bold = True  
  
  
    ## Setting Border  
    borders = Borders()
```

```
borders.left = 1  
borders.right = 1  
borders.top = 1  
borders.bottom = 1  
  
## Setting Pattern  
BkgPat = Pattern()  
BkgPat.pattern = Pattern.SOLID_PATTERN  
BkgPat.pattern_fore_colour = 22  
  
## Buat dan setting Style style0 (Judul Kolom dan  
Rekap Total)  
style0 = XFStyle()  
style0.font = font0  
style0.borders = borders  
style0.pattern = BkgPat  
  
## Buat dan setting Style style1  
style1 = XFStyle()  
style1.borders = borders
```

```
## Buat dan setting Style style_judul
style_judul = XFStyle()
style_judul.font = font0

## Format Tanggal
fmt ='DD-MM-YYYY'
style_tgl = XFStyle()
style_tgl.num_format_str = fmt

## Format Nominal isi kolom
fmt ='Rp #,##0.00'
style_num1 = XFStyle()
style_num1.borders = borders
style_num1.num_format_str = fmt

## Format Nominal total omzet
fmt ='Rp #,##0.00'
style_num2 = XFStyle()
style_num2.borders = borders
style_num2.num_format_str = fmt
style_num2.pattern = BkgPat
```

```
style_num2.font = font0

# Hitung Jumlah Baris
jumbar =self.lc.GetItemCount()

## Ambil Data Tanggal dr wx.DatePickerCtrl
selected = self.tgl_dari.GetValue()
month = selected.Month + 1
day = selected.Day
year = selected.Year
dari1=datetime.date(year,month,day)

selected = self.tgl_sampai.GetValue()
month = selected.Month + 1
day = selected.Day
year = selected.Year
sampai1=datetime.date(year,month,day)

## Cetak Header
sheet1.write(0,0,"PENJUALAN",style_judul)           LAPORAN      OMZET
sheet1.write(1,0,",style_judul)           TOKO SUKSES HUDAYA
```

```
sheet1.write(2,0,"Dari Tgl", style_judul)
sheet1.write(2,1,dari1,style_tgl)
sheet1.write(3,0,"Sampai Tgl", style_judul)
sheet1.write(3,1,sampai1,style_tgl)

i = 4

# Beri Judul Kolom
sheet1.write(i,0,"Nota",style0)
sheet1.write(i,1,"Total Harga",style0)

j=0
while j<=jumbar-1 :
    ## Isikan Item Data Barang
    nota1 = self.lc.GetItem(j,0).GetText()
    sheet1.write(j+i+1,0,nota1,style1)
    total1 = self.lc.GetItem(j,1).GetText()
    sheet1.write(j+i+1,1,int(total1),style_num1)
    j=j+1
    k=j+i+1
    sheet1.write(k,0,"Total Omzet",style0)

sheet1.write(k,1,int(self.txt_total_semua.GetValue()),style_num2)
```

```
# Atur Lebar Kolom  
  
sheet1.col(0).width = 3500  
  
sheet1.col(1).width = 5000  
  
  
  
path1 = 'C:\\cetak_lap_omzet.xls'  
  
# Cek apakah ada file lama  
  
if os.path.exists(path1) :  
  
    # Hapus file lama  
  
    os.remove(path1)  
  
    # Simpan file baru  
  
    book.save(path1)  
  
    # Luncurkan file baru  
  
    os.system("start excel.exe C:\\cetak_lap_omzet.xls")
```

DAFTAR PUSTAKA

Hendri. 2003. *Cepat Mahir Python.*
www.ilmukomputer.com

InfoLinux. 2007. *Panduan Menguasai Pemrograman Visual Wxpython.* Jakarta : Dian Rakyat

Kadir, Abdul.2005. *Dasar Pemrograman Python.*Yogya : Andi Publisher

Peter Norton, et al. 2005. *Beginning Python.*Indianapolis: Wiley Publishing

wiki.wxpython.org/AnotherTutorial, diakses 30 Mei 2011

zetcode.com/databases/mysqlpythontutorial, diakses 30 Mei 2011

LAMPIRAN A : INSTALASI IDLE Python, wxPython, dan Python-MySQL Connector di Windows 7

I. INSTALASI IDLE PYTHON 2.5 DI WINDOWS :

1. Klik 2 kali pada Python 2.5 Installer



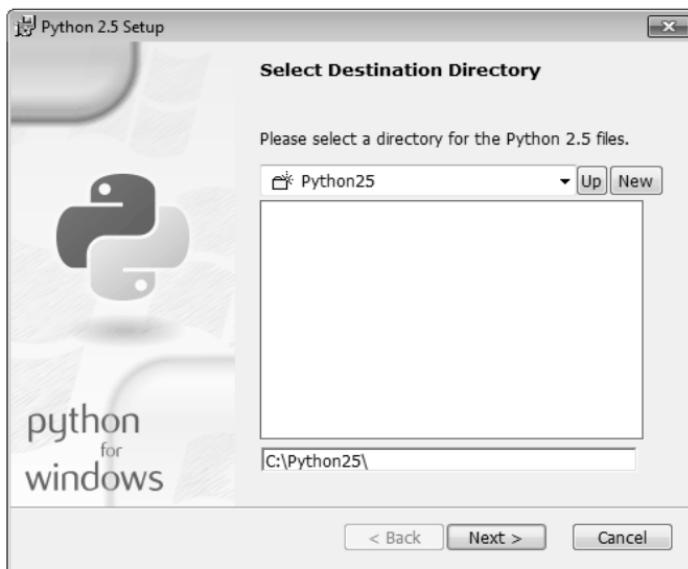
Gambar A.1 Icon Python 2.5

2. Muncul tampilan **Python 2.5 Setup**, Klik Next



Gambar A.2 Jendela Python 2.5 Setup

3. Muncul Tampilan : ‘Select Destination Directory’ (Memilih Folder tujuan).



Gambar A.3 Memilih Folder Tujuan

Jika tidak anda ubah, secara default akan tersimpan di C:\Python25\

Klik Next

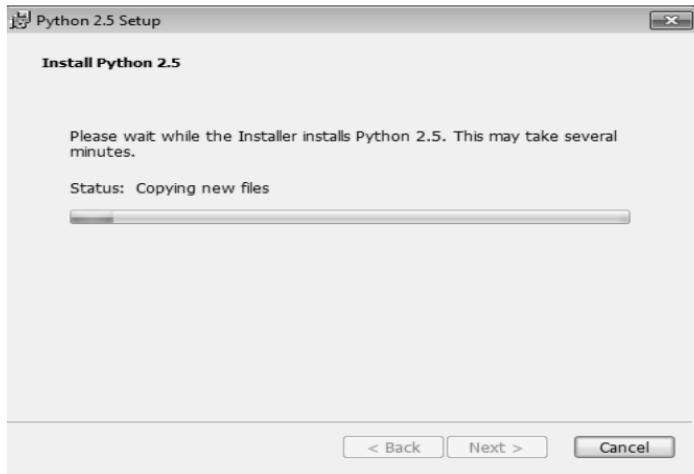
4. Muncul Tampilan : *Customize Python 2.5*



Gambar A.4 Jendela Customize Python 2.5

Pada bagian ini anda bisa menentukan apa saja yang perlu diinstal dan apa yang tidak perlu diinstal. Secara default, dibutuhkan 15 MB penyimpanan pada harddisk. Jika tidak ada yang perlu anda sesuaikan, klik Next.

5. Proses Instalasi Berlangsung



Gambar A.5 Proses Instalasi Python 2.5

6. Proses Instalasi Selesai, klik Finish



Gambar A.6 Instalasi Python 2.5 Selesai

7. Python 2.5 telah terdaftar di deretan Program pada Start Menu



Gambar A.7 Python 2.5 Telah Berada di Start Menu Programs

II. INSTALASI WXPYTHON DI WINDOWS

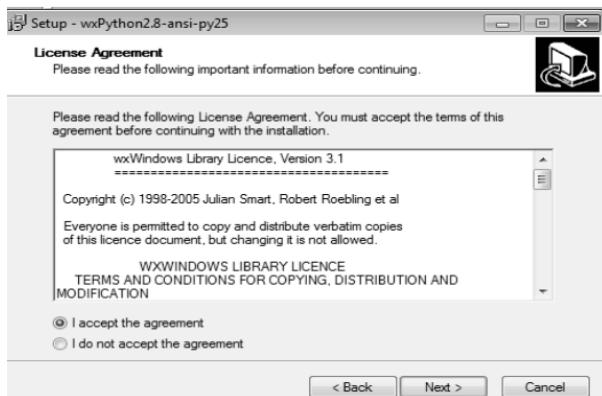
1. Klik 2 kali pada file **wxPython2.8-win32-ansi-2.8.10.1-py25**.
2. Muncul Tampilan Awal Instalasi wxPython



Gambar A.8 Tampilan Awal Setup wxPython

Klik Next untuk melanjutkan

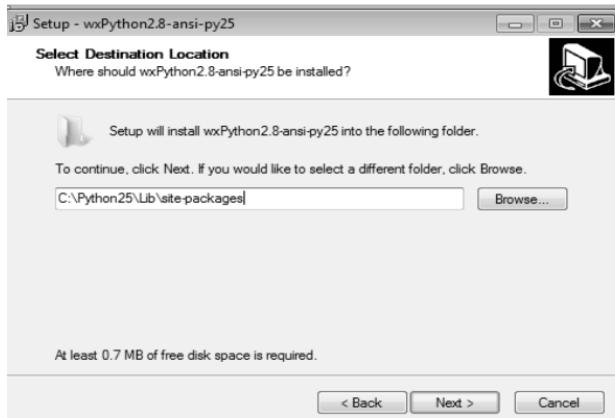
3. Tampilan License Agreement



Gambar A.9 Jendela License Agreement

Pastikan untuk memilih *I accept the agreement* sebelum menekan tombol Next

4. Memilih tujuan instalasi wxPython

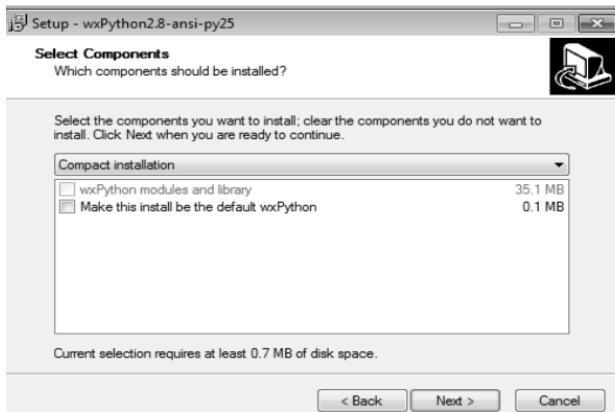


Gambar A.10 Select Destination Location wxPython

Proses instalasi wxPython ini akan mencari folder tempat IDLE Python yang sudah terinstal. Secara default, akan diarahkan pada **C:\Python25\Lib\site-packages**.

Selanjutnya, klik Next

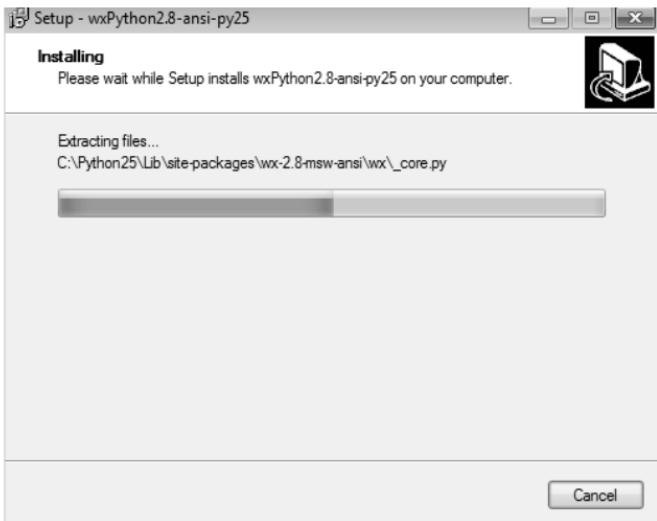
5. Memilih Komponen-komponen yang akan diinstal



Gambar A.11 Select Components wxPython

Secara default, tidak ada perubahan, klik Next

6. Proses instalasi berjalan



Gambar A.12 Proses Instalasi Berjalan

7. Proses instalasi selesai

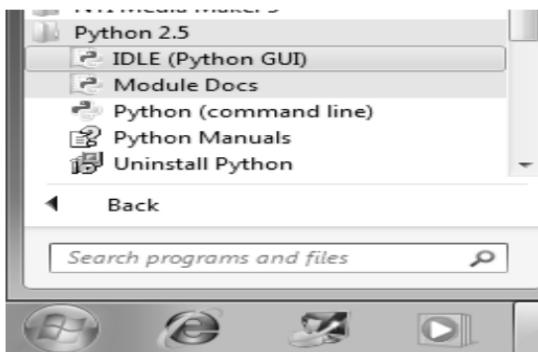


Gambar A.13 Instalasi wxPython sudah selesai

Klik tombol Finish.

Untuk mengecek apakah wxPython sudah terinstal dengan sempurna:

1. Jalankan IDLE Python 2.5 : Start Menu>All Programs> Python 2.5>IDLE (Python GUI)



Gambar A.13 Start Menu> Python 2.5

2. Ketikkan 'import wx' untuk mengecek apakah sudah terinstal dengan baik atau tidak

```
74 Python Shell
File Edit Shell Debug Options Windows Help
Python 2.5 (r25:51908, Sep 19 2006, 09:52:17) [MSC v.1310 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.

*****
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface. This connection is not visible on any external
interface and no data is sent to or received from the Internet.
*****


IDLE 1.2
>>> import wx
>>> |
```

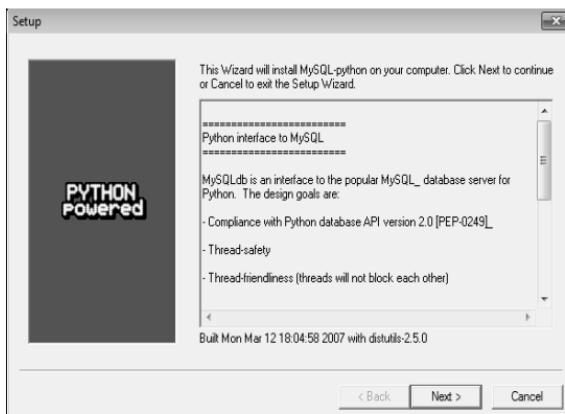
The image shows a screenshot of a Python Shell window titled '74 Python Shell'. The window has standard window controls at the top right. The menu bar includes 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Windows', and 'Help'. The main area displays the Python interpreter's prompt and its response. It starts with the Python version and build information, followed by a warning message about personal firewalls. Below this, the 'copyright', 'credits', and 'license()' functions are mentioned. The final part of the screenshot shows the user's input 'import wx' and the resulting cursor position indicator '|'. The background of the window is white, and the text is black, except for the warning message which is in a smaller font size.

Gambar A.14 Cek import wx

Jika tidak ada pesan error, berarti instalasi sudah berhasil.

III. INSTALASI CONNECTOR PYTHON-MYSQL (MySQLDb)

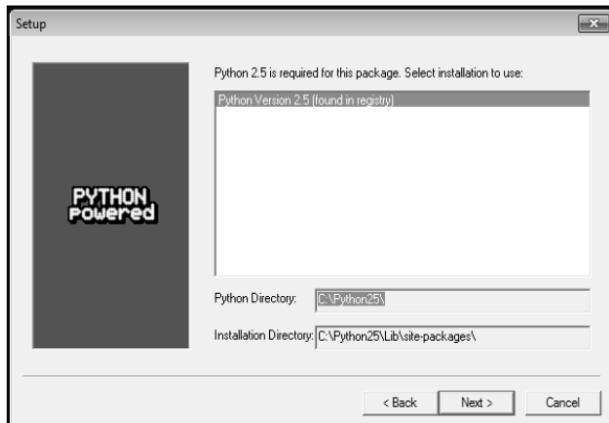
1. Klik 2 kali pada **MySQL-python-1.2.2.win32-py2.5**.
2. Muncul tampilan awal Setup MySQL-Python (MySQLDb)



Gambar A.15 Tampilan Awal MySQLdb

Klik Next

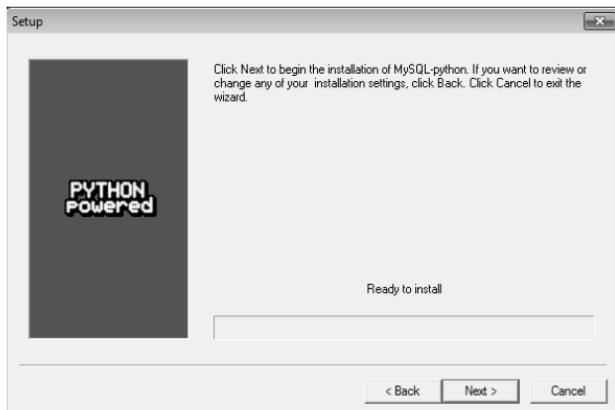
3. Penentuan letak folder yang dituju, disesuaikan dengan IDLE Python 2.5 yang telah terinstal



Gambar A.16 Folder Tujuan Instalasi

Klik Next

4. Persiapan Instalasi (*Ready to Install*)



Gambar A.17 Ready to Install

5. Proses Instalasi Berjalan

6. Instalasi Telah Selesai



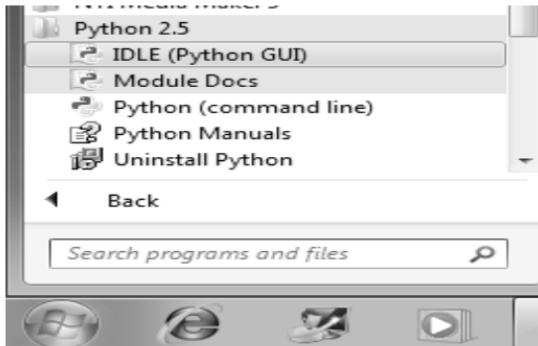
Gambar A.18 Instalasi Telah Selesai

Klik Finish. Selesailah sudah proses instalasi MySQL-Python Connector (MySQLdb)

PENGECEKAN INSTALASI MySQLdb

Untuk mengecek apakah instalasi MySQLdb sudah berjalan baik atau belum:

1. Jalankan IDLE Python 2.5 : Start Menu>All Programs> Python 2.5>IDLE (Python GUI)



Gambar A.19 Start Menu> Python 2.5

2. Ketikkan 'import MySQLdb' untuk mengecek apakah sudah terinstal dengan baik atau tidak

A screenshot of the Python Shell window. The title bar says "% Python Shell". The menu bar includes File, Edit, Shell, Debug, Options, Windows, and Help. The main window displays the following text:

```
Python 2.5 (r25:51908, Sep 19 2006, 09:52:17) [MSC v.1310 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.

*****
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface. This connection is not visible on any external
interface and no data is sent to or received from the Internet.
*****
```

IDL 1.2

```
>>> import MySQLdb
>>> |
```

Gambar A.20 Pengecekan Instalasi MySQLdb

Jika tidak ada pesan kesalahan (Error) yang ditampilkan, berarti instalasi MySQLdb sudah berjalan dengan baik.

LAMPIRAN B: INSTALASI XAMPP

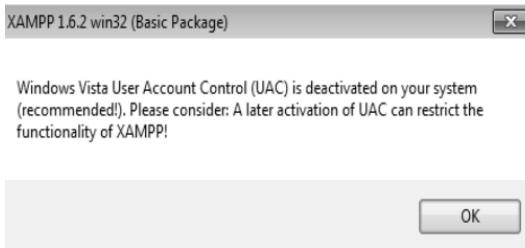
1. Klik 2 kali pada file **xampp-win32-1.6.2-installer**.
2. Pilihan Bahasa Instalasi



Gambar B.1 Pilihan Bahasa Inggris

Klik OK.

3. Konfirmasi bahwa user menonaktifkan Windows Vista UAC (User Account Control)



Gambar B.2 Konfirmasi Bahwa Windows UAC Non Aktif

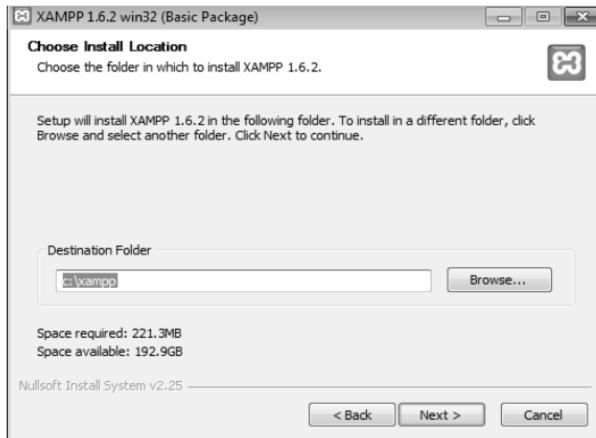
4. Tampilan Awal Instalasi XAMPP



Gambar B.3 Tampilan Awal Instalasi XAMPP

Klik Next

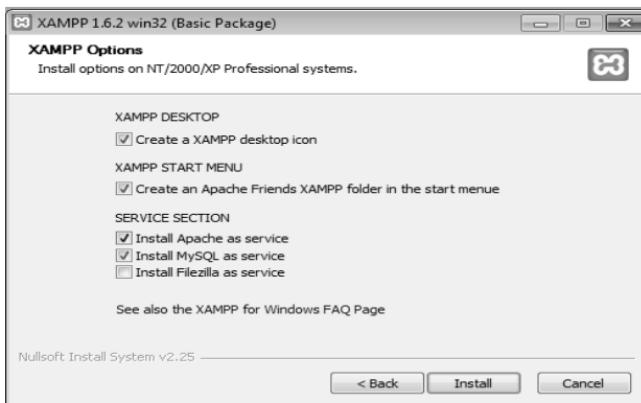
5. Pemilihan Folder Tujuan Instalasi. Secara default diletakkan di C:\xampp



Gambar B.4 Pemilihan Folder Tujuan Instalasi

Klik Next

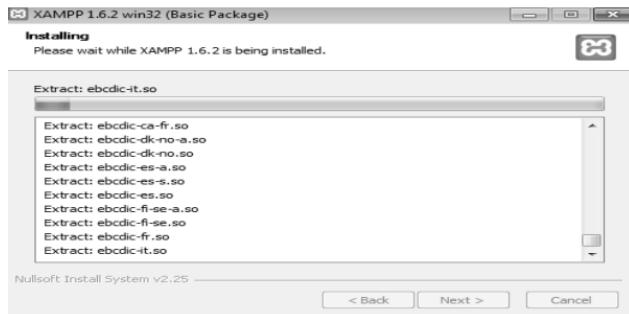
6. Pemilihan Opsi Instalasi XAMPP



Gambar B.5 Pemilihan Opsi Instalasi XAMPP

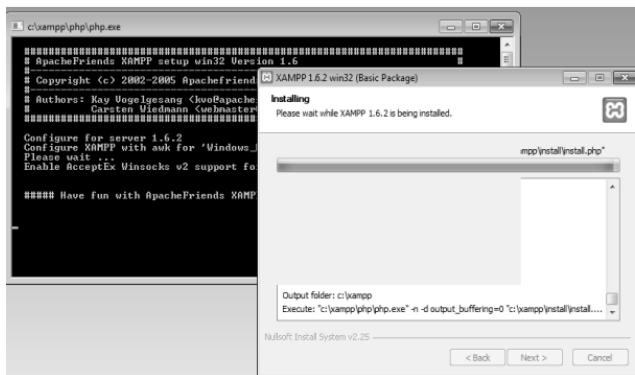
Pastikan telah memberi centang pada bagian: **Install Apache as service** dan **Install MySQL as service**. Kemudian klik tombol **Install**.

7. Proses Instalasi Sedang Berjalan



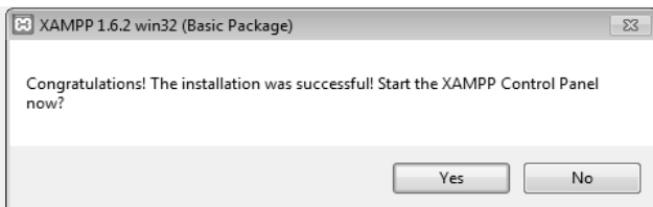
Gambar B.6 Proses Instalasi XAMPP Berjalan

8. Finalisasi Proses Instalasi XAMPP



Gambar B.7 Finalisasi Instalasi XAMPP

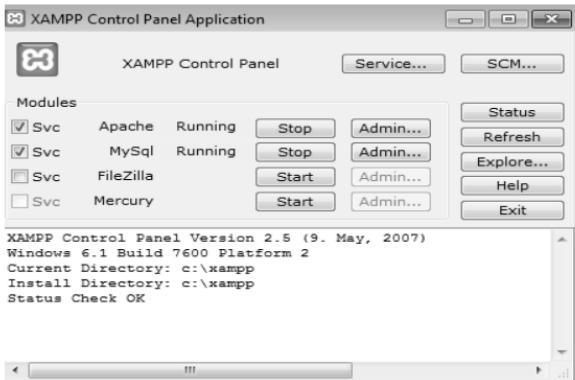
9. Instalasi XAMPP Sudah Selesai



Gambar B.8 Instalasi XAMPP Sudah Sukses

MENJALANKAN XAMPP dan phpmyadmin

1. Masuk ke C:\xampp, klik 2 kali file **xampp-control**.



Gambar B.9 XAMPP Control Panel

Pastikan **Apache** dan **MySQL** dalam kondisi *Running*.

Kadangkala, service XAMPP sudah berjalan bersamaan dengan ketika komputer kita *booting* sehingga tidak perlu kita aktifkan lagi.

2. Jalankan browser semacam Internet Explorer atau Mozilla Firefox dan semisalnya, kemudian ketikkan pada kotak alamat (URL) :
http://localhost/phpmyadmin

Phpmyadmin adalah fasilitas untuk melakukan administrasi database semacam:

- Membuat database baru
- Membuat tabel baru
- Mengubah struktur tabel, dan sebagainya



Gambar B.10 Tampilan Awal phpmyadmin

Secara default, tidak perlu user name dan password untuk masuk ke dalam phpmyadmin. Namun, jika anda ingin mengubah fasilitas security agar tidak semua pihak bisa memasukinya, silakan masuk ke <http://localhost/security>.

LAMPIRAN C: BACKUP (EXPORT) DAN RESTORE (IMPORT) DATABASE

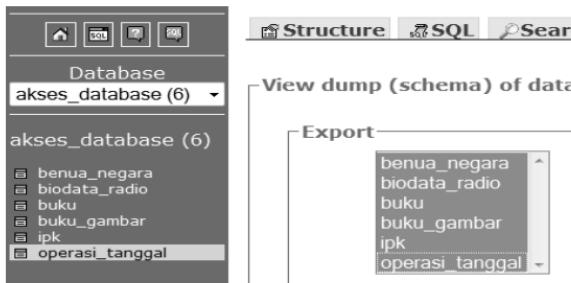
Backup (Export) Database adalah proses memindahkan struktur dan isi database ke sebuah file.

Restore (Import) Database adalah proses memasukkan file hasil Backup (Export) sehingga bisa digunakan kembali.

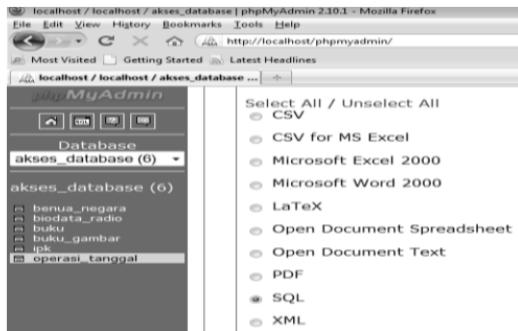
I. LANGKAH-LANGKAH BACKUP (EXPORT)

1. Jalankan phpmyadmin (ketikkan : localhost/phpmyadmin ke kotak alamat URL pada browser)
2. Pilih database yang akan diexport, misalkan : **akses_database**.
3. Klik tab **Export**
4. Pada bagian Export:
 - a. Pilih semua daftar tabel pada List (Gambar C.1)
 - b. Pilih radiobutton SQL sebagai ekstensi file tujuan yang akan dihasilkan (Gambar C.2)
 - c. Centang pada bagian **Save as file** (Gambar C.3)
 - d. Ketikkan nama file yang akan dihasilkan, di bagian **File name template** (Gambar C.3)

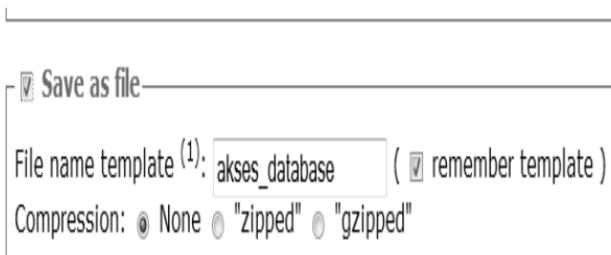
Selanjutnya, klik tombol Go



Gambar C.1 Pilih Semua Tabel di Bagian Export

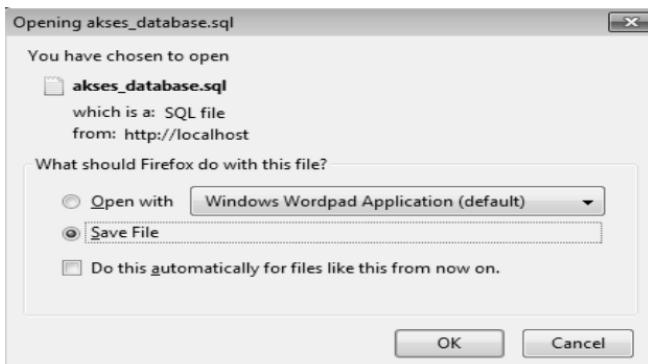


Gambar C.2 Pilih SQL sebagai Ekstensi File



Gambar C.3 Centang Save as File dan Beri Nama File

5. Tampil Pesan untuk Menyimpan File



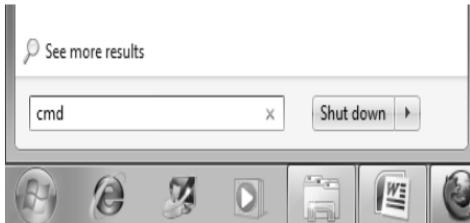
Gambar C.4 Tampilan Pesan Penyimpanan File

6. Proses mendownload file ke komputer sendiri dan tersimpan pada alamat default download pada browser.

Jika anda menggunakan Mozilla Firefox, secara default letak hasil download file adalah di C:\Users\[**Nama_User**]\Downloads. Misal, nama user adalah Hilman, maka tempatnya di C:\Users\Hilman\Downloads.

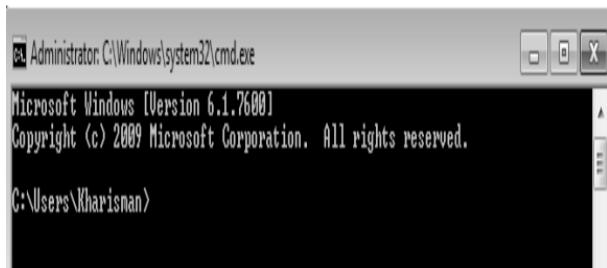
CARA LAIN BACKUP DATABASE

1. Start Menu > Ketikkan 'cmd' di text pencarian (Search)



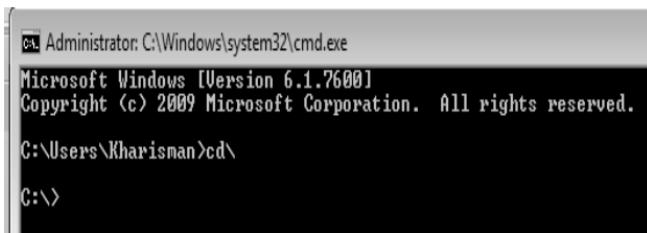
Gambar C.5 Ketik ‘cmd’ di Kotak Pencarian

2. Muncul Jendela Command Prompt



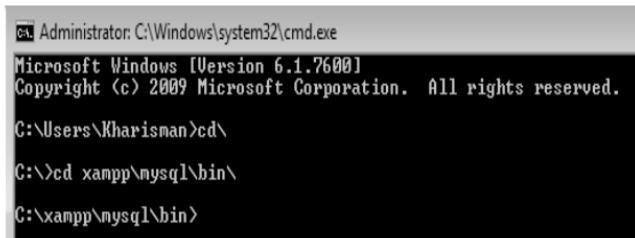
Gambar C.6 Jendela Command Prompt

3. Ketikkan ‘cd\’, kemudian tekan Enter



Gambar C.7 Ke Root Directory C

4. Ketikkan : cd xampp\mysql\bin\ (mengarahkan direktori yang aktif)

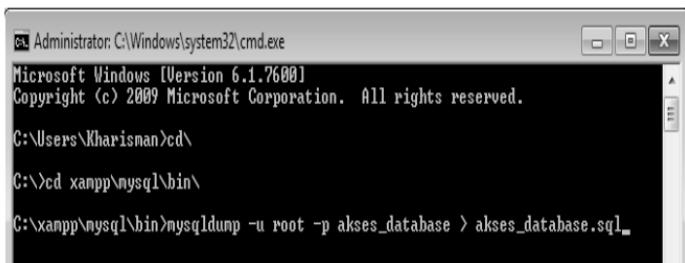


```
C:\Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\Users\Kharisman>cd\
C:\>cd xampp\mysql\bin\
C:\xampp\mysql\bin>
```

Gambar C.8 Ke Alamat MySQL\bin

5. Ketikkan : **mysqldump -u root -p akses_database > akses_database.sql**



```
C:\Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

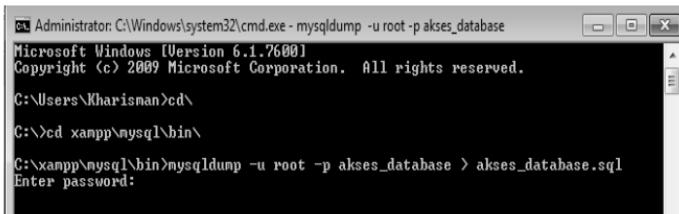
C:\Users\Kharisman>cd\
C:\>cd xampp\mysql\bin\
C:\xampp\mysql\bin>mysqldump -u root -p akses_database > akses_database.sql
```

Gambar C.9 Kode Backup Database

Kemudian tekan Enter.

(Nama database yang akan dibackup adalah **akses_database**)

6. Masukkan password MySQL (dalam kasus kita ini passwordnya kosong (default))



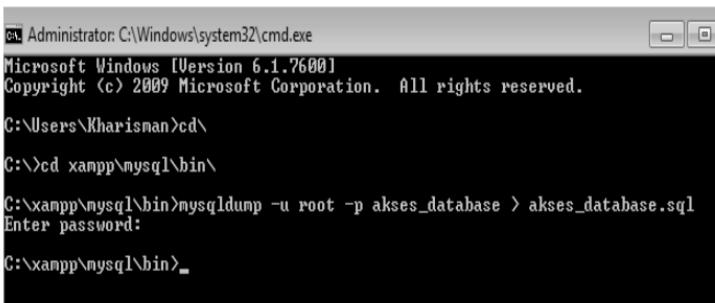
```
Administrator: C:\Windows\system32\cmd.exe - mysqldump -u root -p akses_database
Microsoft Windows [Version 6.1.7600]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\Users\Kharisman>cd\
C:\>cd xampp\mysql\bin\
C:\xampp\mysql\bin>mysqldump -u root -p akses_database > akses_database.sql
Enter password:
```

Gambar C.10 Masukkan Password MySQL

Tekan Enter.

7. Akan menunggu sejenak untuk diproses, jika tidak ada Error, maka proses Backup berhasil



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\Users\Kharisman>cd\
C:\>cd xampp\mysql\bin\
C:\xampp\mysql\bin>mysqldump -u root -p akses_database > akses_database.sql
Enter password:
C:\xampp\mysql\bin>
```

Gambar C.11 Proses Backup Sukses

8. Dihasilkan file backup : **akses_database.sql** di C:\xampp\mysql\bin

II. LANGKAH-LANGKAH RESTORE (IMPORT) DATABASE

1. Jalankan phpmyadmin (ketikkan : localhost/phpmyadmin ke kotak alamat URL pada browser)

2. Buatlah Nama Database yang akan diimport, misalnya : **kamus**. Kemudian tekan tombol **Create**



Gambar C.12 Create New Database ‘kamus’

3. Pilih tab **Import**
4. Pada bagian Import, klik tombol Browse untuk mencari lokasi file hasil backup sebelumnya.
5. Klik tombol Go

CARA LAIN RESTORE (IMPORT) DATABASE

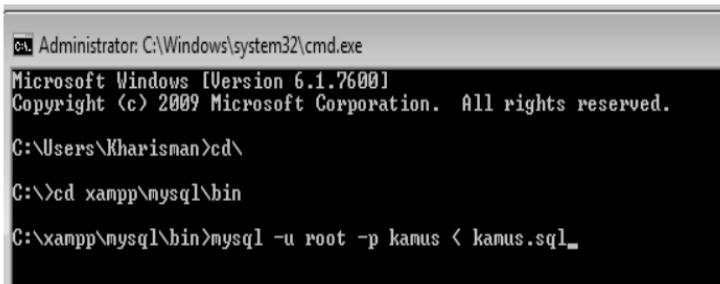
Import dengan cara di atas hanya terbatas untuk file dengan kapasitas tertentu. Jika anda ingin mengimport file yang lebih besar kapasitasnya, bisa menggunakan langkah-langkah di bawah ini :

1. Buatlah Database dengan nama yang sama dengan nama database pada file yang akan diimport.
2. Copykan file yang akan diimport (berekstensi .sql) ke C:\xampp\mysql\bin
3. Jalankan Command Prompt seperti pada langkah-langkah membuat backup database di atas.

4. Arahkan direktori aktif di command prompt ke C:\xampp\mysql\bin
5. Ketikkan perintah Restore (Import):

mysql -u -root -p kamus < kamus.sql

selanjutnya tekan Enter, dan anda akan diminta memasukkan password



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Kharisman>cd\
C:\>cd xampp\mysql\bin
C:\xampp\mysql\bin>mysql -u root -p kamus < kamus.sql
```

Gambar C.14 Perintah Restore (Import)

6. Proses Restore (Import) sudah selesai.

LAMPIRAN D : MEMBUKA FILE APLIKASI CONTOH DALAM BUKU

File-file project contoh sudah ada dalam CD yang disertakan pada buku ini. Anda tinggal pilih folder sesuai Bab yang akan dipilih. Sebagai contoh, kita akan membuka dan mencoba file-file aplikasi penjualan.

Khusus bab yang terkait dengan database : VIII,X,XII, dan XIII anda harus mengimport terlebih dahulu file .sql (hasil backup database) ke MySQL. Cara import database bisa dilihat pada Lampiran C.

Langkah-langkah untuk mencoba file aplikasi penjualan:

1. Jalankan Boa Constructor (petunjuknya ada di Bab II).
2. Buka semua file yang ada (wx.App dan wx.Frame) melalui jendela Editor.
3. Aktifkan tab wx.App
4. Jalankan aplikasi dengan mengklik tanda kuning di jendela Editor.