# **BAB IX**

# **GUI PROGRAMMING**

## **9.1. Tujuan**

1. Untuk memenuhi Tugas Laporan Praktikum Modul 8 Dasar Komputer dan Pemrograman.
2. Mengenal dan mengetahui perbedaan penggunaan beberapa *IDE* atau aplikasi pemrograman dan beberapa jenis bahasa pemrograman.
3. Memahami pengertian dan konsep dari *Graphical User Interface(GUI) Programming*.
4. Mengetahui penggunaan *GUI Programming* dengan bahasa pemrograman Python, C#, Java, dan PHP.
5. Mengetahui perbedaan *GUI Programming* dalam bahasa pemrograman Python, C#, Java, dan PHP.
6. Mampu mengimplementasikan *GUI Programming* dalam bahasa pemrograman Python, C#, Java, dan PHP.
7. Mampu membuat program sederhana yang berhubungan dengan pemrograman *GUI* dengan bahasa pemrograman Java, C#, PHP dan Python.
8. Menjalin silaturahmi dengan kakak tingkat dan teman se-angkatan.

## **Dasar Teori**

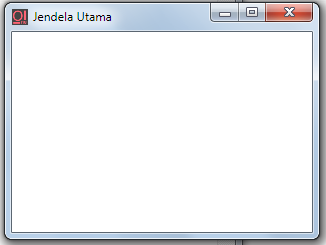
### ***GUI* pada Python**

Di dalam python terdapat beberapa *library* atau modul yang dapat digunakan untuk mengembangkan aplikasi *GUI*. Tapi kebanyakan *library* tersebut bukan *library default* dari python sendiri, melainkan itu merupakan pengembangan dari pihak ketiga. Contoh modul yang sering dipakai adalah *PyQt, Thinker, wxPython, PyGTK*, dan lain-lain. Satu-satunya *library* yang masuk dalam *library* *default* adalah *Thinker*.

Secara umum, langkah-langkah yang harus dilakukan untuk membuat *GUI* menggunakan *Thinker* kurang lebih seperti berikut ini:

1. Mengimport modul *Thinker*.
2. Membuat *Form* yang akan dijadikan *main form*.
3. Menambahkan kontrol-kontrol yang diperlukan.
4. Membuat fungsi-fungsi untuk menangani *event* pada kontrol.
5. Terakhir yaitu menampilkan *form*.

|  |
| --- |
| import tkinter  from tkinter import \*  root = Tk()  root.wm\_title("Jendela Utama")# untuk tampilkan title di window border  root.wm\_iconbitmap('one.ico')# untuk tampilkan icon di window border  # dengan syarat icon harus ada di folder yang sama dengan file python ini    root.geometry('300x200')# untuk menentukan ukuran window  root.configure(bg="white")# untuk merubah tampilan warna  root.state("normal")#digunkan untuk window fullscreen tinggal ganti normal menjadi zoomed  root.mainloop() |

Sumber: teknowledge1.blogspot.com, sutandi1.blogspot.com

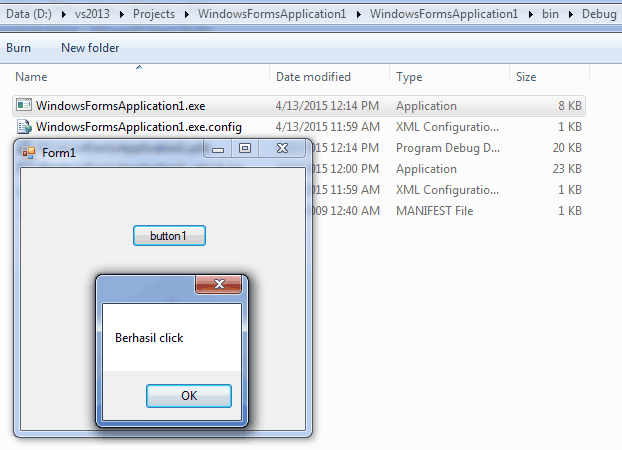
Gambar 9. 1. Output GUI dalam Python (Thinker)

### ***GUI* pada C#**

*GUI* adalah *Graphical User Interface* dalam dunia komputer. Pada komputer terdapat *GUI* atau antarmuka pengguna secara grafis. Istilah ini bukan hal yang lumrah pada saat awal kemunculan komputer. Namun setelah komputer generasi keempat mulai diciptakan, munculnya televisi berwarna (yang mendorong pada penciptaan layar monitor berwarna) serta evolusi pada perangkat penampil gambar (*graphic adapter* atau *graphic card*) membuat komputer mulai mendapatkan suatu sistem baru.

Berikut ini contoh penerapan *GUI* dalam bahasa pemrograman C# :

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace WindowsFormsApplication1  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  }  private void button1\_Click(object sender, EventArgs e)  {  MessageBox.Show(“Berhasil click”);  }  }  } |



Gambar 9. 2. Output GUI dalam C#

Sumber: proweb.co.id, belajarcsharpdi.blogspot.com

### ***GUI* pada Java**

Pada pemrograman Java, terdapat 2 cara yaitu melalui *Java Class* (dengan pembuatan tampilan dan penataan letak *full code*) dan bisa juga menggunakan *jFrame Form*. Kedua nya tentu memiliki kelebihan dan kekurangan masing-masing.

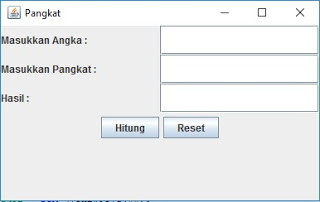
Pada *Java Class*, *script* yang dihasilkan jauh lebih sedikit sehingga ukuran *file* yang di buat juga lebih kecil, namun kelemahannya adalah pada segi waktu karena *full coding*. Berbanding terbalik apabila kita menggunakan *jFrame Form* di mana waktu pembuatan *GUI* bisa kita tekan seminimal mungkin, karena pada prakteknya kita cukup melakukan *drag and drop* komponen yang kita butuhkan, namun kode hasil *generate* tentu memiliki jumlah baris yang berkali kali lebih banyak di banding dengan metode sebelumnya.

Beberapa fungsi dan komponen GUI dalam Java, antara lain:

1. *jFrame* - Adalah sebuah fungsi yang memuat komponen *Frame* pada tampilan *GUI* di Java di mana *frame* atau jendela tersebut bisa memiliki judul, *border*, *menu*, tombol, *text field* dan beberapa komponen lain di dalamnya.
2. *jLabel* - Adalah sebuah fungsi yang memuat komponen Label di mana fungsinya adalah untuk menampilkan *string* atau gambar pada *frame*.
3. *jTextField* - Adalah sebuah fungsi yang memuat komponen *Text Field* di mana fungsinya untuk menangkap *input* pengguna, dan tempat pengguna memasukkan data.
4. *jPanel* - Adalah sebuah fungsi yang memuat komponen Panel di mana fungsinya semacam partisi atau pemisah antara *layout* 1 dengan lainnya.
5. *jButton* - Adalah sebuah fungsi yang memuat komponen *Button* dimana fungsinya adalah sebagai tombol untuk memfinalisasi sebuah proses pengambilan keputusan.

Berikut ini contoh penerapan *GUI* dalam bahasa pemrograman Java:

|  |
| --- |
| import javax.swing.\*;  import java.awt.\*;  import java.awt.event.\*;  class pangkat{  JFrame frame;  JLabel lbAngka, lbPangkat, lbHasil;  JTextField txAngka, txPangkat, txHasil;  JPanel panel1, panel2;  JButton btHitung, btReset;  public pangkat(){  frame = new JFrame();  frame.setTitle("Pangkat");  frame.setSize(400,250);  frame.getContentPane().setLayout(new GridLayout(2,1));  lbAngka = new JLabel("Masukkan Angka :");  lbPangkat = new JLabel("Masukkan Pangkat :");  lbHasil = new JLabel("Hasil :");  txAngka = new JTextField("");  txPangkat = new JTextField("");  txHasil = new JTextField("");  btHitung = new JButton("Hitung");  btReset = new JButton("Reset");  panel1 = new JPanel();  panel1.setLayout(new GridLayout(3,2));  panel1.add(lbAngka);  panel1.add(txAngka);  panel1.add(lbPangkat);  panel1.add(txPangkat);  panel1.add(lbHasil);  panel1.add(txHasil);  panel2 = new JPanel();  panel2.setLayout(new FlowLayout());  panel2.add(btHitung);  panel2.add(btReset);  frame.getContentPane().add(panel1);  frame.getContentPane().add(panel2);  frame.setVisible(true);  }  public static void main (String [] args) {  pangkat text = new pangkat();  }  } |

Sumber: jaluprasetyamulya.blogspot.com

Gambar 9. 3. Output GUI dalam Java

### ***GUI* pada PHP**

*GTK+* merupakan *toolkit* untuk menciptakan objek *user interface* yang awalnya diciptakan untuk membuat aplikasi edit gambar *GIMP*. Andrei Zmievski merupakan orang yang berjasa membuat *PHP-GTK* untuk memungkinkan orang membuat aplikasi desktop dengan memanfaatkan bahasa PHP.

Kelebihan dari *GTK* adalah *library* yang cukup lengkap. Dan aplikasi yang dibuat dengan *PHP-GTK* dapat dijalankan di *Windows*, *Linux* maupun *MacOS*. Pada saat pertama kali diusungkan *PHP-GTK* disambut antusias bagi pecinta PHP. Bahkan permintaan akan *PHP-GTK* di pasaran *freelance* juga ada. Hanya saja, perkembangan *PHP-GTK* sepertinya tersendat karena terakhir kali *update* adalah tahun 2008. Walau demikian, dari semua library untuk memungkinkan PHP membuat aplikasi *desktop*, *PHP-GTK* merupakan yang paling terkenal. *Wapache*.

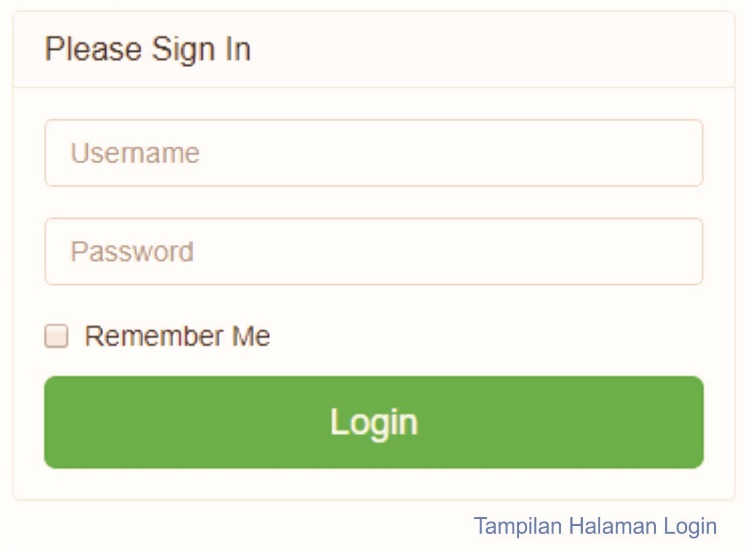
Berikut ini contoh penerapan *GUI* dalam bahasa pemrograman PHP:

*Gambar 9. 4.* Source Code GUI *dalam PHP (1)*

*Gambar 9. 5.* Source Code GUI *dalam PHP (2)*



Gambar 9. 6. Source Code GUI dalam PHP (3)

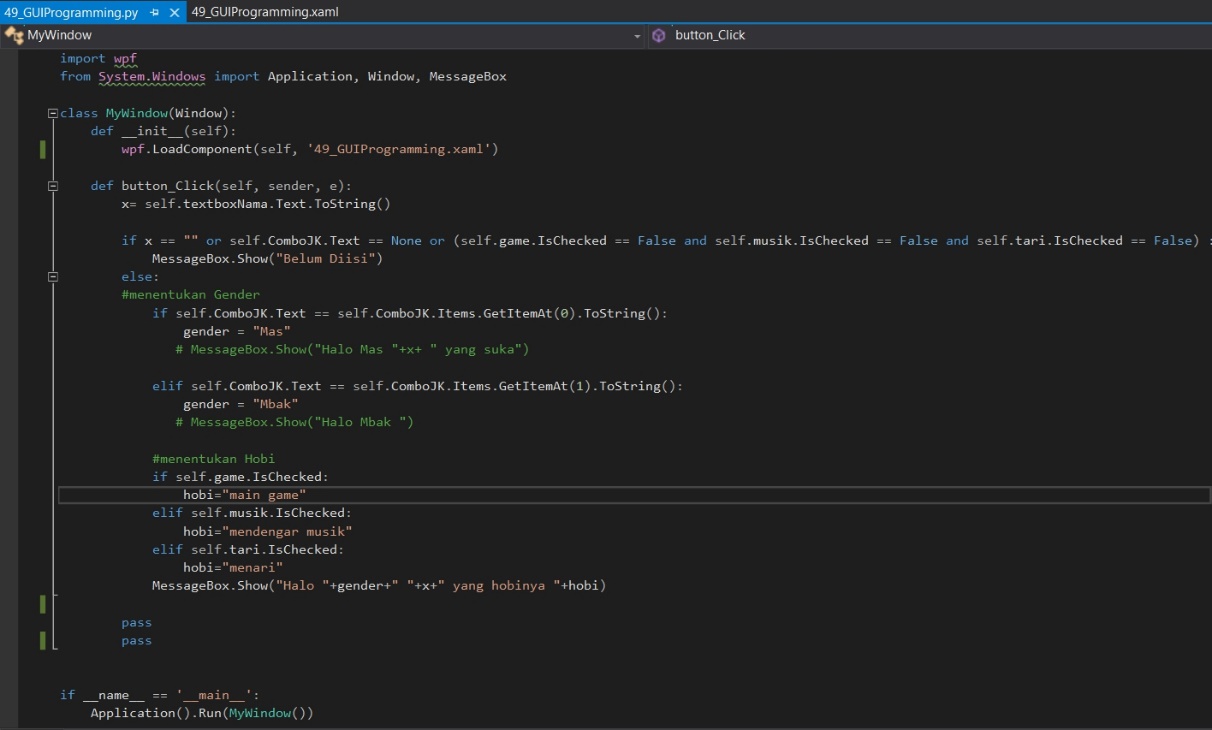
Sumber: computesta.com, idcloudhost.com 

Gambar 9. 7. Output GUI dalam PHP

## **Hasil Percobaan dan Analisis**

### ***GUI* pada Python**

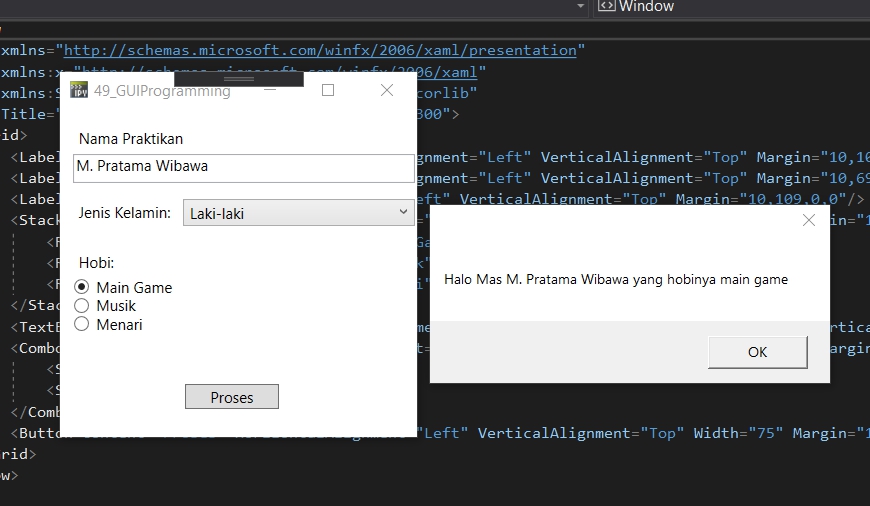
|  |
| --- |
| import wpf  from System.Windows import Application, Window, MessageBox  class MyWindow(Window):  def \_\_init\_\_(self):  wpf.LoadComponent(self, '49\_GUIProgramming.xaml')  def button\_Click(self, sender, e):  x= self.textboxNama.Text.ToString()  if x == "" or self.ComboJK.Text == None or (self.game.IsChecked == False and self.musik.IsChecked == False and self.tari.IsChecked == False) :  MessageBox.Show("Belum Diisi")  else:  #menentukan Gender  if self.ComboJK.Text == self.ComboJK.Items.GetItemAt(0).ToString():  gender = "Mas"  # MessageBox.Show("Halo Mas "+x+ " yang suka")    elif self.ComboJK.Text == self.ComboJK.Items.GetItemAt(1).ToString():  gender = "Mbak"  # MessageBox.Show("Halo Mbak ")    #menentukan Hobi  if self.game.IsChecked:  hobi="main game"  elif self.musik.IsChecked:  hobi="mendengar musik"  elif self.tari.IsChecked:  hobi="menari"  MessageBox.Show("Halo "+gender+" "+x+" yang hobinya "+hobi)  pass  pass  if \_\_name\_\_ == '\_\_main\_\_':  Application().Run(MyWindow()) |



Gambar 9. 8. Source Code GUI dalam Python



Gambar 9. 9. Tampilan GUI dalam Python

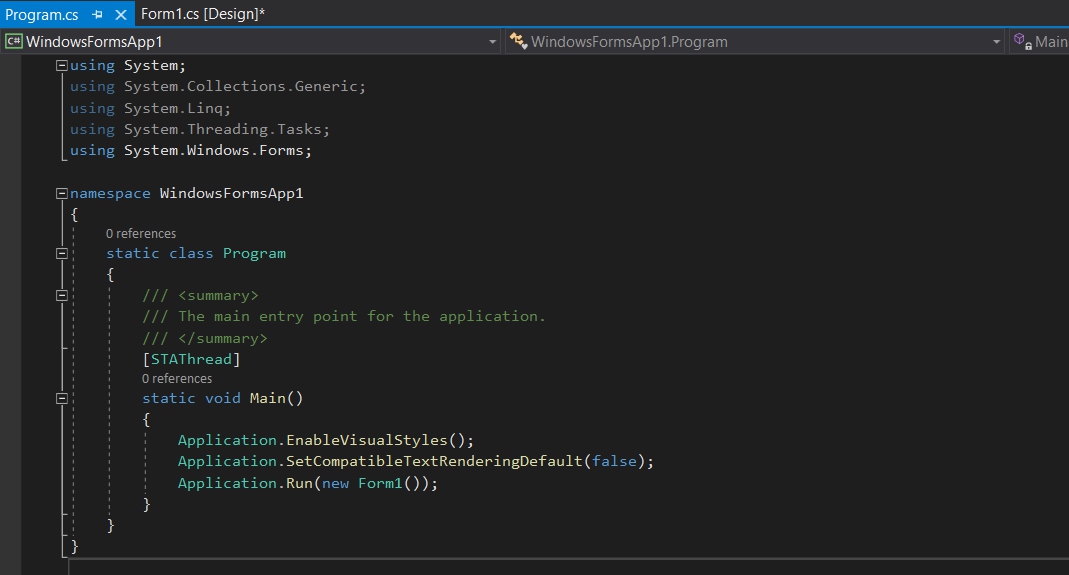


Gambar 9. 10. Output GUI dalam Python

Pada Percobaan python ini pertama kita susun *GUI* Python dan kita beri nama satu persatu yang nantinya akan menjadi perwakilan variabel didalam pemrograman kita. Setelah itu kita masuk ke pemrograman kita, pertama kita *import* *wpf* untuk pemrograman *GUI* kita, dan selanjutnya kita *import* dari *System.Windows* untuk mengimport hal-hal yang kita perlukan seperti *message box,application* dan *windows*. Setelah itu kita masuk ke kelasnya dan kita buat kelas inisialisasi dimana isinya mengimport dari file *GUI* yang telah kita susun dan berformat *xaml* yang kita buat sebelumnya. Lalu kita *double click* di *button* sebagai penanda apa yang harus program lakukan bila *button* ditekan. Selanjutnya kita isi *command list* didalam *button* tersebut, kita perintah pengkondisian dimana bila data belum diisi secara lengkap maka akan keluar *message box* bahwa data belum lengkap terisi. Selanjutnya kita buat pengambilan data, pada *text box* kita langsung *convert* ke *string* sebagai nama, untuk *combobox* dan *radio button* kita beri opsi satu persatu dengan perkondisian dimana bila pengunjung memilih salah satu pilihan maka variabelnya akan berbeda beda. Selanjutnya kita gabungkan semua variabel yang telah berisi diatas dan keluarkan hasilnya menggunakan *message box*.

### ***GUI* pada C#**

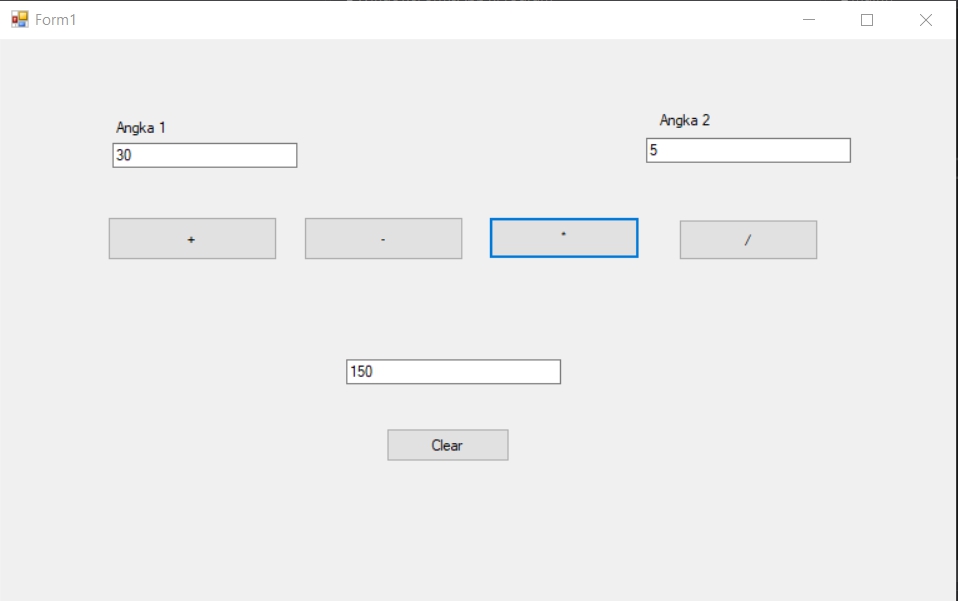
|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace WindowsFormsApp1  {  static class Program  {  /// <summary>  /// The main entry point for the application.  /// </summary>  [STAThread]  static void Main()  {  Application.EnableVisualStyles();  Application.SetCompatibleTextRenderingDefault(false);  Application.Run(new Form1());  }  }  } |



Gambar 9. 11. Source Code GUI dalam C#



Gambar 9. 12. Tampilan GUI dalam C#



Gambar 9. 13. Ouptut GUI dalam C#

Pada Percobaan C# ini pertama kita susun *GUI* C# dan kita beri nama satu persatu yang nantinya akan menjadi perwakilan variabel didalam pemrograman kita. Setelah itu kita *import* yang kita perlukan seperti *system*, *system.text*, dan masih banyak lagi. Setelah itu kita klik button satu persatu dan beri perintah di masing-masing *button* supaya program tahu apa yang akan dilakukan bila *button* tersebut ditekan. Jangan lupa kita harus menarik angka yang telah dimasukkan pengguna ke *textbox*, setelah *textbox* diisi dan *button* ditekan maka *source code* pada *button* yang telah kita isi diatas akan jalan sesuai dengan button apa yang ditekan.

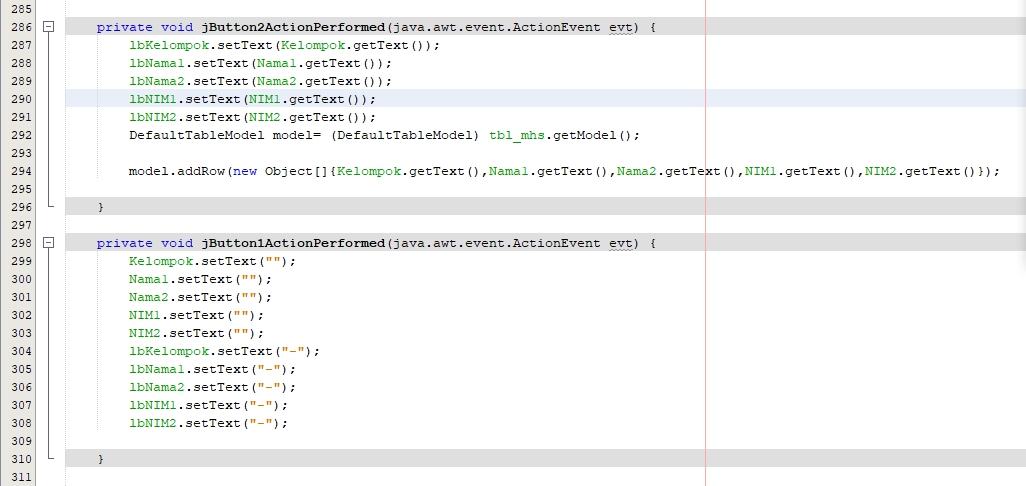
### ***GUI* pada Java**

***jButton2***

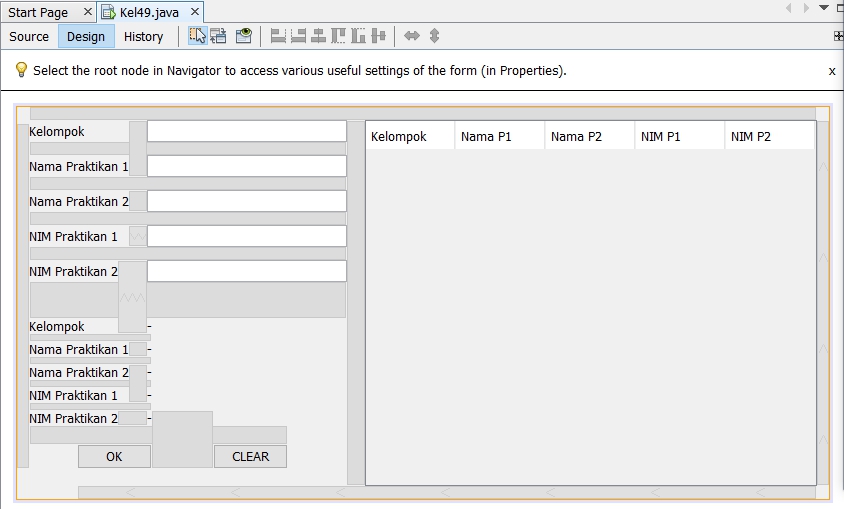
|  |
| --- |
| private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  lbKelompok.setText(Kelompok.getText());  lbNama1.setText(Nama1.getText());  lbNama2.setText(Nama2.getText());  lbNIM1.setText(NIM1.getText());  lbNIM2.setText(NIM2.getText());  DefaultTableModel model= (DefaultTableModel) tbl\_mhs.getModel();  model.addRow(new Object[]{Kelompok.getText(),Nama1.getText(),Nama2.getText(),NIM1.getText(),NIM2.getText()});  } |

***jButton1***

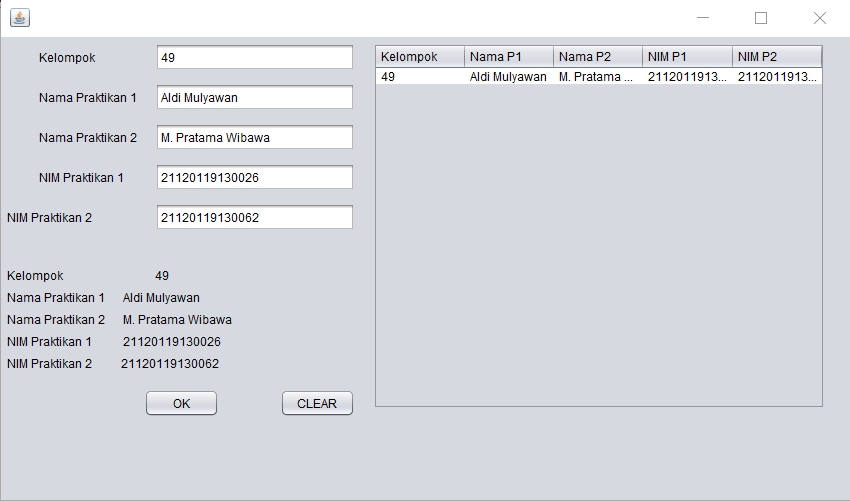
|  |
| --- |
| private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  Kelompok.setText("");  Nama1.setText("");  Nama2.setText("");  NIM1.setText("");  NIM2.setText("");  lbKelompok.setText("-");  lbNama1.setText("-");  lbNama2.setText("-");  lbNIM1.setText("-");  lbNIM2.setText("-");  } |



Gambar 9. 14. Penambahan Source Code GUI dalam Java



Gambar 9. 15. Tampilan GUI dalam Java



Gambar 9. 16. Output GUI dalam Java

Pada Percobaan Java ini pertama kita susun *GUI* Java dan kita beri nama satu persatu yang nantinya akan menjadi perwakilan variabel didalam pemrograman kita. Setelah itu kita *import* yang kita perlukan seperti *system* , *system*.*text* , dan masih banyak lagi. Setelah itu kita klik *button* satu persatu dan beri perintah di masing-masing *button* supaya program tahu apa yang akan dilakukan bila *button* tersebut ditekan. Jangan lupa kita harus menarik angka yang telah dimasukkan pengguna ke *textbox*, setelah *textbox* diisi dan *button* ditekan maka *source code* pada *button* yang telah kita isi diatas akan jalan sesuai dengan *button* apa yang ditekan. Jangan lupa di paling atas kita *import* supaya *library* model tabel dan array dapat dioperasikan.

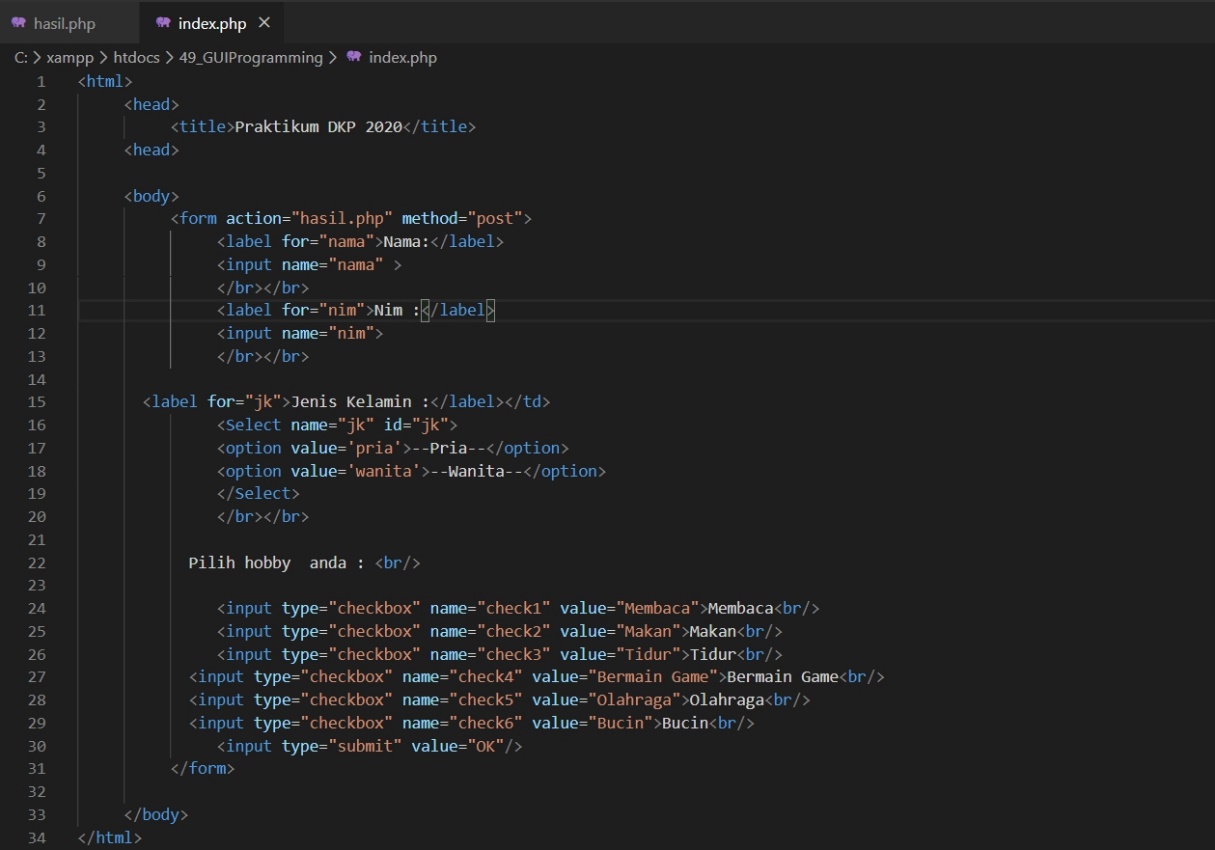
### ***GUI* pada PHP**

***index.php***

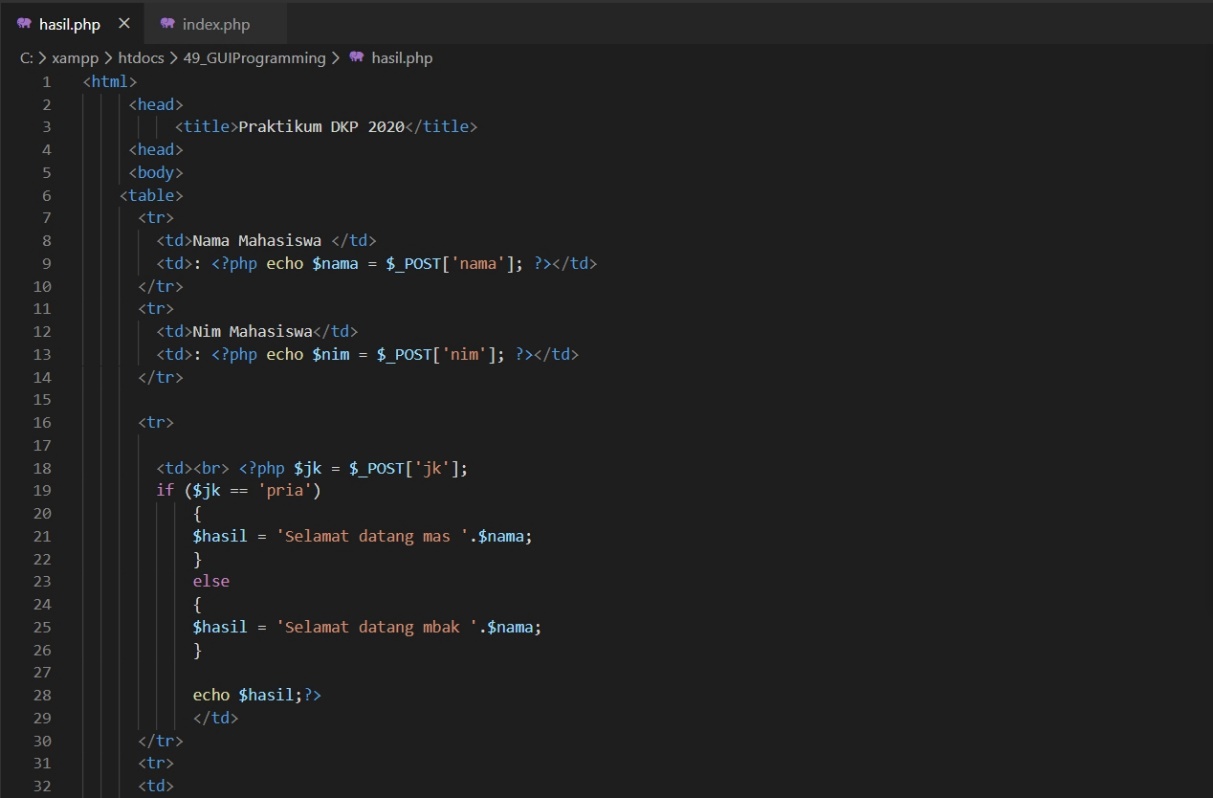
|  |
| --- |
| <html>  <head>  <title>Praktikum DKP 2020</title>  <head>    <body>  <form action="hasil.php" method="post">  <label for="nama">Nama:</label>  <input name="nama" >  </br></br>  <label for="nim">Nim :</label>  <input name="nim">  </br></br>  <label for="jk">Jenis Kelamin :</label></td>  <Select name="jk" id="jk">  <option value='pria'>--Pria--</option>  <option value='wanita'>--Wanita--</option>  </Select>  </br></br>    Pilih hobby anda : <br/>    <input type="checkbox" name="check1" value="Membaca">Membaca<br/>  <input type="checkbox" name="check2" value="Makan">Makan<br/>  <input type="checkbox" name="check3" value="Tidur">Tidur<br/>  <input type="checkbox" name="check4" value="Bermain Game">Bermain Game<br/>  <input type="checkbox" name="check5" value="Olahraga">Olahraga<br/>  <input type="checkbox" name="check6" value="Bucin">Bucin<br/>  <input type="submit" value="OK"/>  </form>    </body>  </html> |

***hasil.php***

|  |
| --- |
| <html>  <head>  <title>Praktikum DKP 2020</title>  <head>  <body>  <table>  <tr>  <td>Nama Mahasiswa </td>  <td>: <?php echo $nama = $\_POST['nama']; ?></td>  </tr>  <tr>  <td>Nim Mahasiswa</td>  <td>: <?php echo $nim = $\_POST['nim']; ?></td>  </tr>    <tr>    <td><br> <?php $jk = $\_POST['jk'];  if ($jk == 'pria')  {  $hasil = 'Selamat datang mas '.$nama;  }  else  {  $hasil = 'Selamat datang mbak '.$nama;  }    echo $hasil;?>  </td>  </tr>  <tr>  <td>  Hobby anda adalah : <br>  <?php  if (isset($\_POST['check1'])) {  echo $\_POST['check1']." <br/>";  }  if (isset($\_POST['check2'])) {  echo $\_POST['check2']." <br/>";  }  if (isset($\_POST['check3'])) {  echo $\_POST['check3']." <br/>";  }  if (isset($\_POST['check4'])) {  echo $\_POST['check4']." <br/>";  }  if (isset($\_POST['check5'])) {  echo $\_POST['check5']." <br/>";  }  if (isset($\_POST['check6'])) {  echo $\_POST['check6']." <br/>";  }  ?></td>  </tr>  </table>  </body>  </html> |



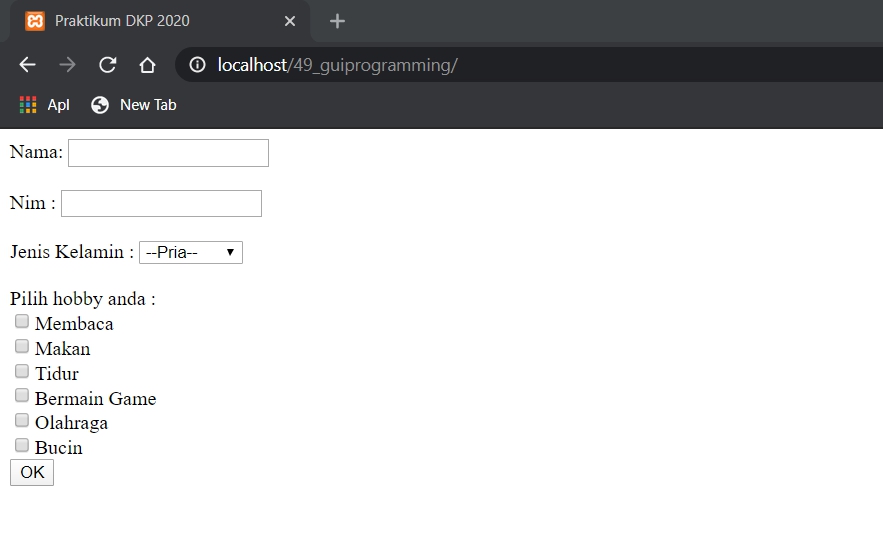
Gambar 9. 17. Source Code GUI dalam PHP (1)



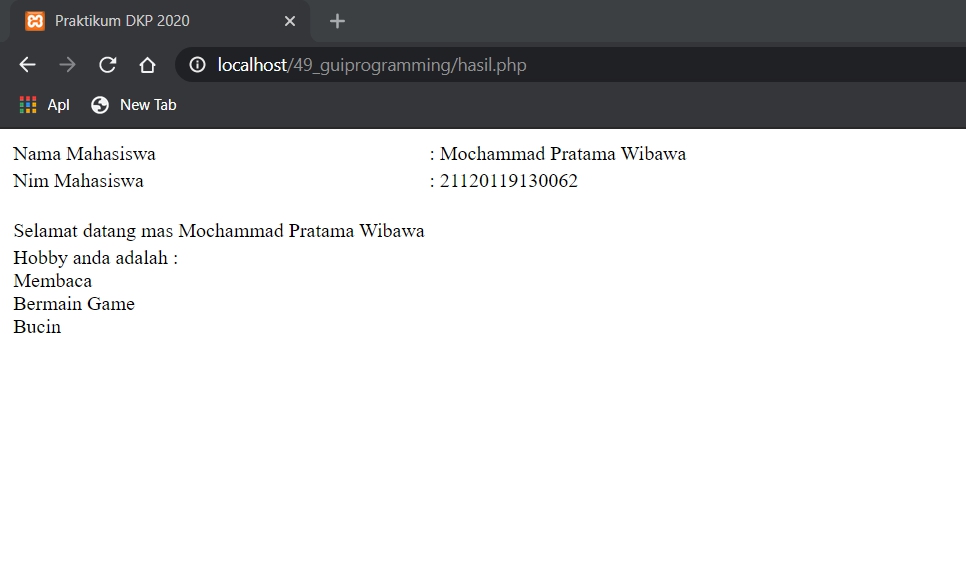
Gambar 9. 18. Source Code GUI dalam PHP (2)



Gambar 9. 19. Source Code GUI dalam PHP (3)



Gambar 9. 20. Tampilan GUI dalam PHP

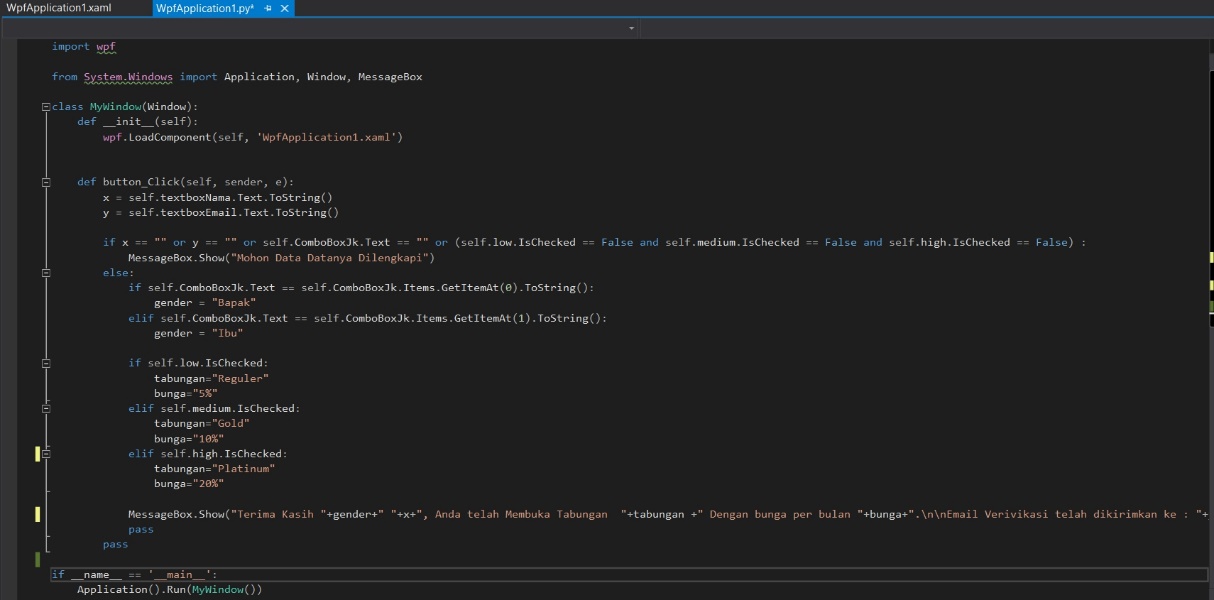


Gambar 9. 21. Output GUI dalam PHP

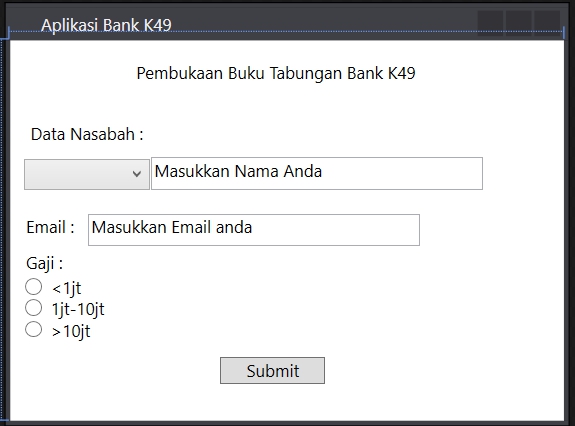
Pada percobaan PHP pertama kita buat file *index.php* untuk memasukkan *GUI* yang kita perlukan dan *hasil.php* untuk memproses hasilnya. Pertama-tama kita masuk ke *index.php*, dan kita buat format *GUI* yang kita perlukan dari nama, nim, jenis kelamin dan hobi dengan menggunakan *form*, *checkbox* dan *value*. Lalu kita masukkan *form* *action* untuk mengarah ke *hasil.php*, sekarang kita masuk ke *hasil.php* untuk memasukkan apa yang ingin dilakukan pada *input* data sebelumnya. Pada percobaan kali ini kita langsung mengeluarkan *input*-an yang telah dimasukkan dengan format yang berbeda.

## **Tugas**

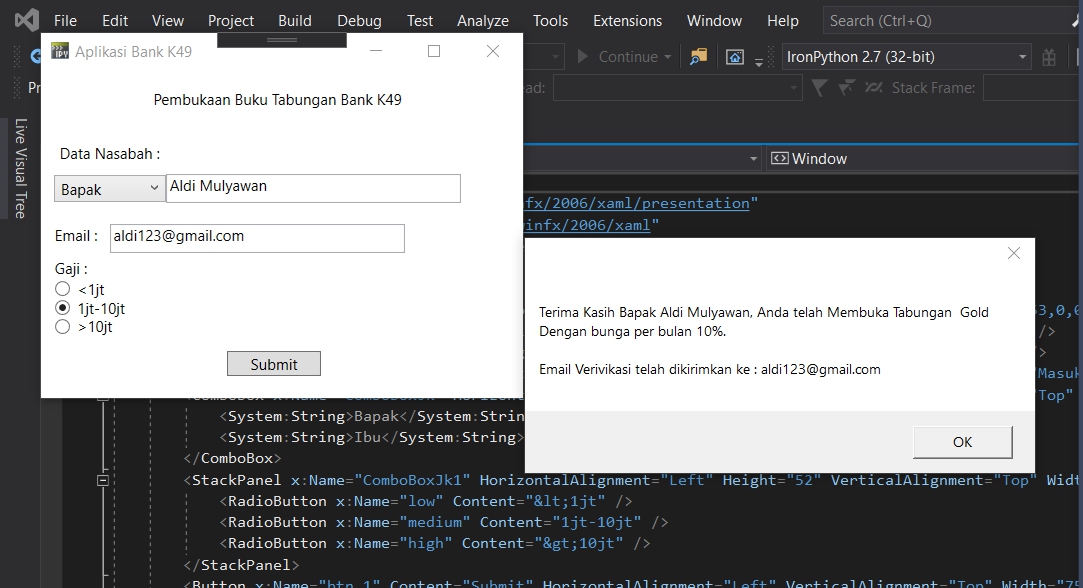
|  |
| --- |
| import wpf  from System.Windows import Application, Window, MessageBox  class MyWindow(Window):  def \_\_init\_\_(self):  wpf.LoadComponent(self, 'WpfApplication1.xaml')    def button\_Click(self, sender, e):  x = self.textboxNama.Text.ToString()  y = self.textboxEmail.Text.ToString()  if x == "" or y == "" or self.ComboBoxJk.Text == "" or (self.low.IsChecked == False and self.medium.IsChecked == False and self.high.IsChecked == False) :  MessageBox.Show("Mohon Data Datanya Dilengkapi")  else:  if self.ComboBoxJk.Text == self.ComboBoxJk.Items.GetItemAt(0).ToString():  gender = "Bapak"  elif self.ComboBoxJk.Text == self.ComboBoxJk.Items.GetItemAt(1).ToString():  gender = "Ibu"  if self.low.IsChecked:  tabungan="Reguler"  bunga="5%"  elif self.medium.IsChecked:  tabungan="Gold"  bunga="10%"  elif self.high.IsChecked:  tabungan="Platinum"  bunga="20%"  MessageBox.Show("Terima Kasih "+gender+" "+x+", Anda telah Membuka Tabungan "+tabungan +" Dengan bunga per bulan "+bunga+".\n\nEmail Verivikasi telah dikirimkan ke : "+y)  pass  pass    if \_\_name\_\_ == '\_\_main\_\_':  Application().Run(MyWindow()) |



Gambar 9. 22. Source Code Tugas GUI dalam Python



Gambar 9. 23. Tampilan Tugas GUI dalam Python



Gambar 9. 24. Output Tugas GUI dalam Python

Penjelasan *Output* :

Pada Tugas python ini pertama-tama kita susun *GUI* Python dan kita beri nama satu-satu yang nantinya akan menjadi perwakilan variabel didalam pemrograman kita. Disini kami membuat nama, *email*, jenis kelamin dan gaji dari nasabah yang ingin mendaftar. Setelah itu kita masuk ke pemrograman kita, kita import *wpf* untuk pemrograman *GUI* kita, dan selanjutnya kita *import* dari *System.Windows* untuk mengimport hal-hal yang kita perlukan seperti *message box,application* dan *windows*. Setelah itu kita masuk ke kelasnya dan kita buat kelas inisialisasi dimana isinya *mengimport* dari *file* *GUI* yang telah kita susun dan berformat *xaml* yang kita buat sebelumnya. Lalu kita *double click* di *button* sebagai penanda apa yang harus program lakukan bila *button* ditekan. Selanjutnya kita isi *command list* didalam *button* tersebut, kita perintah pengkondisian dimana bila data belum diisi secara lengkap maka akan keluar *message box* bahwa data belum lengkap terisi. Selanjutnya kita buat pengambilan data, pada *text box* kita langsung *convert* ke *string* sebagai nama dan *email*, untuk *combobox* dan *radio button* kita beri opsi satu persatu dengan perkondisian dimana bila nasabah memilih salah satu pilihan maka variabelnya akan berbeda beda. Misalkan bila nasabah memilih gaji dibawah 1 juta maka otomatis akan mendaftar ke tabungan Reguler dan bila diatas 10 juta maka akan mendaftar tabungan Platinum. Begitu juga untuk *combo box* jika memilih bapak maka panggilan di hasil akan keluar bapak, begitu juga sebaliknya. Selanjutnya kita gabungkan semua variabel yang telah berisi diatas dan keluarkan hasilnya menggunakan *message box*.

## **Kesimpulan**

* + 1. Bahasa pemrograman PHP merupakan bahasa pemrograman yang paling cocok untuk pembuatan *GUI*, karena paling stabil dalam pembuatan *GUI* nya dan didampingi dengan *HTML*.
    2. *GUI* dalam Python, C# dan Java memiliki *template* tersendiri, sedangkan PHP harus menulis manual.
    3. *Ironpython* masih jarang dipakai oleh kebanyakan orang, karena masih dalam tahap pengembangan.
    4. Diperlukan *mengimport* beberapa library baru supaya dapat menggunakan *GUI* dalam bahasa pemrograman Python, Java dan C#.
    5. Dalam pembuatan *GUI*, penamaan variabel sangat penting dan sangat sensitif. Kesalahan huruf atau spasi dapat membuat program *error*.
    6. Pada pemrograman *GUI*, untuk bahasa pemrograman Python, Java, dan C# output berupa *pop up windows* sementara itu PHP berupa halaman *web*.
    7. Dalam penggunaan *Ironpython* hanya diberitahu kode 1 bila ada kesalahan, Hal ini merupakan hal yang merepotkan untuk mencari kesalahan program.
    8. Antara bahasa pemrograman satu dengan yang lainnya kadang terdapat perbedaan-perbendaan tertentu, karena memiliki aturannya masing-masing.
    9. Diperlukan ketelitian dalam mebuat sebuah program, karena kesalahan sedikit akan membuat program tidak berjalan dengan baik.