



MODUL 13

Pemrograman GUI (*Graphical User Interface*)

A. Tujuan

1. Mampu memahami konsep dari GUI (*Graphical User Interface*)
2. Mampu menerapkan penggunaan dari GUI (*Graphical User Interface*)

B. Materi Pengantar

Sama seperti pemrograman lain pada umumnya, Python juga dapat digunakan untuk mengembangkan aplikasi berbasis GUI (*Graphical User Interface*). Terdapat beberapa GUI toolkit yang dapat digunakan untuk membuat aplikasi visual dalam python. Contoh GUI toolkit yang populer adalah, PyQt, Tkinter, PyGObject, wxPython dan Pyside.

C. Pemrograman GUI dengan Tkinter

Tkinter termasuk kedalam pustaka standar python. Oleh karena itu, untuk menggunakan pustaka tersebut kita tidak perlu melakukan instalasi apapun. Satu hal yang perlu dilakukan yaitu mengimport modul tkinter.

Contoh 13.1

```
import tkinter

mainform = tkinter.Tk()

mainform.wm_title('Hello')
mainform.mainloop()
```

Wm_title adalah metode didalam kelas Tk yang digunakan untuk mengubah judul form.

D. Membuat dan Menempatkan Kontrol di dalam Form

Tkinter menyediakan kontrol – kontrol standar untuk pengembangan aplikasi GUI. Untuk menempatkan kontrol ke dalam form, kita perlu membuat objek atau instance dari kelas kontrol bersangkutan terlebih dahulu.



Contoh 13.2

```
import tkinter

mainform = tkinter.Tk()
#membuat objek label
lbl = tkinter.Label(mainform)
lbl['text'] = "Masukkan nama anda"
lbl.pack()
#membuat objek entry
txt = tkinter.Entry(mainform)
txt['width'] = 40
txt.pack()
#membuat objek button
btn = tkinter.Button(mainform)
btn['text'] = "Lanjut"
btn.pack()

mainform.wm_title('Hello!')
mainform.mainloop()
```

E. Menangani Event

Event adalah notifikasi atau pemberitahuan yang dikirim oleh sistem operasi terhadap kejadian yang sedang menimpa kontrol tertentu. Misalnya, ketika kontrol Button di klik, tidak ada aktivitas yang dikerjakan oleh kontrol tersebut, Untuk membuat form tersebut interaktif, kita perlu mendefinisikan suatu fungsi untuk merespon kejadian. Ini berarti bahwa fungsi tersebut akan dipanggil pada saat suatu event terjadi pada kontrol tertentu. Fungsi ini sering disebut callback, atau event – handler.

Contoh 13.3

```
import tkinter
import tkinter.messagebox as messagebox

mainform = tkinter.Tk()
#membuat event
def buttonclick () :
    messagebox.showinfo("Hallo", "Hallo %s, apa kabar?" % (txt.get()))
#membuat objek label
lbl = tkinter.Label(mainform)
lbl['text'] = "Masukkan nama anda"
lbl.pack()
#membuat objek entry
txt = tkinter.Entry(mainform)
txt['width'] = 40
txt.pack()
#membuat objek button
btn = tkinter.Button(mainform, command=buttonclick)
btn['text'] = "Lanjut"
btn.pack()

mainform.wm_title('Hello')
mainform.mainloop()
```



F. Mengatur Tampilan Form dengan Grid Manager

Dalam Tkinter, pengaturan tampilan (layout) dari kontrol kontrol yang terdapat di dalam form dapat dilakukan menggunakan tiga metode yaitu pack manager, grid manager, dan place manager.

Pack manager, yang digunakan untuk menempatkan suatu kontrol ke dalam form adalah pack(). Pada Grid manager kita perlu menggantinya dengan metode grid(). Grid manager akan membagi layar form menjadi tampilan tabel dua dimensi, yang tersusun atas baris dan kolom. Sel dalam Grid manager dapat dibentangkan menjadi beberapa kolom maupun baris, sama seperti pada saat anda membuat tabel menggunakan Microsoft Word.

Contoh 13.4

```
import tkinter

mainform = tkinter.Tk()
#membuat objek label
lbl1 = tkinter.Label(mainform)
lbl1['text'] = "Masukkan nama anda"
lbl1.grid(row=0, column=0, sticky=tkinter.E)

#membuat objek entry
txt1 = tkinter.Entry(mainform)
txt1['width'] = 40
txt1.grid(row=0, column=1, columnspan=2)

txt2 = tkinter.Entry(mainform)
txt2['width'] = 40
txt2.grid(row=1, column=1, columnspan=2)

chk = tkinter.Checkbutton(mainform)
chk['text'] = "Remember me"
chk.grid(row=2, column=0)

#membuat objek button
btn1 = tkinter.Button(mainform)
btn1['text'] = "Lanjut"
btn1.grid(row=2, column=1, sticky=tkinter.N+tkinter.E+tkinter.S+tkinter.W)

btn2 = tkinter.Button(mainform)
btn2['text'] = "Exit"
btn2.grid(row=2, column=2, sticky=tkinter.N+tkinter.E+tkinter.S+tkinter.W)
mainform.wm_title('Hello')
mainform.mainloop()
```

G. Menggunakan Kelas Frame

Sebenarnya kita dapat membuat form menggunakan kelas frame. Objek dari kelas frame berperan sebagai wadah atau kontainer dari kontrol kontrol lain. Contoh penggunaan kelas Frame adalah sebagai berikut :



Contoh 13.5

```
import tkinter

mainform = tkinter.Frame()
mainform.grid()

btnout = tkinter.Button(mainform, text="keluar", command=mainform.destroy)
btnout.grid(sticky=tkinter.E+tkinter.S, padx=90, pady=90)
mainform.master.title("Demo Frame")
mainform.mainloop()
```

H. Kontrol Button

Kontrol ini digunakan untuk membuat tombol di dalam form. Dalam tkinter, informasi yang tampil di dalam kontrol Button dapat berupa teks maupun gambar. Terdapat beberapa opsi opsi yang dapat digunakan pada kontrol tkinter yaitu

Opsi	Keterangan
Activebackground	Warna latar ketika tombol ditekan namun tombol mouse belum dilepas
Activeforeground	Warna teks ketika tombol ditekan namun tombol mouse belum dilepas
Bd	Menentukan tebal bingkai
Bg	Menentukan warna background
Command	Menentukan fungsi yang akan dipanggil
Fg	Warna teks yang tampil di dalam tombol
Font	Jenis huruf yang akan digunakan
Height	Menentukan tinggi tombol
Image	Menampilkan gambar di dalam tombol
Justify	Menentukan posisi teks di dalam tombol
Padx	Menentukan jarak tambah di sebelah kiri atau bawah
Pady	Menentukan jarak tambahan di atas atau di bawah teks
Relief	Menentukan tampilan tombol
State	Menentukan status tombol
Text	Menentukan label atau teks yang akan tampil dalam tombol
Underline	Membuat garis bawah pada teks
Width	Menentukan lebar tombol
Wraplength	Menentukan lebar teks

Berikut contoh penggunaan beberapa opsi diatas.



Contoh 13.6

```
import tkinter
import tkinter.messagebox

def buttonclick():
    tkinter.messagebox.showinfo("Informasi", "Hello!")

def main():
    mainform = tkinter.Tk()
    mainform.title("Demo button")
    mainform.geometry("280x200")

    button = tkinter.Button(mainform, text="Click here!", background="#334477", width=10, command=buttonclick)
    button.grid(sticky=tkinter.E+tkinter.S, padx=90, pady=120)
    mainform.mainloop()

if __name__ == '__main__':
    main()
```

I. Kontrol CheckButton

Kontrol ini digunakan untuk membuat opsi yang dapat menampung banyak pilihan, misalnya hobi. Contoh penggunaan kontrol CheckButton adalah seperti berikut ini

Contoh 13.7

```
import tkinter
import tkinter.messagebox

def main():
    mainform = tkinter.Tk()
    mainform.title("Demo button")
    mainform.geometry("280x200")
    #Variabel yang akan dihubungkan
    var1 = tkinter.IntVar()
    var2 = tkinter.IntVar()
    var3 = tkinter.IntVar()

    #Fungsi lokal
    def getchoice():
        pilihan = []
        if var1.get() == 1: pilihan.append("Musik")
        if var2.get() == 1: pilihan.append("Olahraga")
        if var3.get() == 1: pilihan.append("Membaca")
        tkinter.messagebox.showinfo("Informasi", str(pilihan))

    #Kontrol label
    I = tkinter.Label(mainform, text="Hobi anda (pilih salah satu): ")
    I.grid(row=0, sticky=tkinter.W)
```



```
#Membuat checkbutton
c1 = tkinter.Checkbutton(mainform, variable=var1, text="Musik")
c1.grid(row=1, sticky=tkinter.W)

c1 = tkinter.Checkbutton(mainform, variable=var2, text="Olahraga")
c1.grid(row=2, sticky=tkinter.W)

c1 = tkinter.Checkbutton(mainform, variable=var3, text="Membaca")
c1.grid(row=3, sticky=tkinter.W)

#Membuat kontrol button
b1 = tkinter.Button(mainform, text="OK", command=getchoice, width=10)
b1.grid(row=4, sticky=tkinter.W, padx=4, pady=4)
b2 = tkinter.Button(mainform, text="Keluar", command=quit, width=10)
b2.grid(row=4, sticky=tkinter.W, padx=90, pady=4)

mainform.mainloop()
if __name__ == '__main__':
    main()
```

J. Kontrol Entry

Kontrol ini digunakan oleh user untuk memasukkan informasi informasi (berupa teks) yang diperlukan oleh program. Bentuk umum pembuatan kontrol entry adalah sebagai berikut:

Contoh 13.8

```
import tkinter
import tkinter.messagebox

def main():
    mainform = tkinter.Tk()
    mainform.title("Demo button")
    mainform.geometry("155x100")
    #Variabel yang akan dihubungkan
    var1 = tkinter.StringVar()
    #Menentukan nilai default
    var1.set("Python...")
    #Fungsi callback
    def kosongkan():
        var1.set("")
    def ambilteks():
        tkinter.messagebox.showinfo("Informasi", var1.get())

    #Membuat kontrol label
    l = tkinter.Label(mainform, text="Masukkan teks")
    l.grid(row=0, column=0, columnspan=2, sticky=tkinter.W+tkinter.E, padx=4, pady=4)

    #Membuat kontrol entry
    e = tkinter.Entry(mainform, textvariable=var1)
    e.grid(row=1, column=0, columnspan=2, sticky=tkinter.W+tkinter.E, padx=4, pady=4)

    #Membuat kontrol button
    b1 = tkinter.Button(mainform, text="Kosongkan", command=kosongkan)
    b1.grid(row=2, column=0, padx=4, pady=4)
    b2 = tkinter.Button(mainform, text="Ambil teks", command=ambilteks)
    b2.grid(row=2, column=1, padx=4, pady=4)

    mainform.mainloop()
if __name__ == '__main__':
    main()
```