MODUL PRAKTIKUM

PROGRAMA KOMPUTER

PROGRAM STUDI TEKNIK INDUSTRI
UNIVERSITAS SEBELAS MARET

TIM ASISTEN LABORATORIUM PERANCANGAN DAN OPTIMASI SISTEM INDUSTRI 2020

MODUL X

GUI (*Graphical User Interface*)

A. Tujuan

Berikut merupakan tujuan dari Praktikum Programa Komputer Modul X.

1. Memahami teknik membuat dan penyusunan *syntax* dalam membuat GUI.
2. Mengetahui dan memahami modul yang digunakan dalam membuat GUI pada Python.
3. Mampu menerapkan penggunaan modul dalam membuat GUI pada Python.

B. Pendahuluan

Graphical User Interface atau kerap disebut GUI adalah sebuah sistem komponen visual yang interaktif yang dapat digunakan pada komputer, ponsel pintar, laptop, maupun barang elektronik lainnya. Disebut interaktif karena komponen visual yang dibuat dengan program tersebut akan berubah warna, bentuk, ataupun menampilkan interaksi jika terdapat stimulus dari penggunanya seperti berupa suara dan sentuhan layar atau tombol.

Karena didesain menggunakan program, maka GUI dapat dikembangkan juga menggunakan Python sebagai salah satu bahasa pemrogramannya. Dalam penggunaannya, sudah ada beberapa modul yang populer digunakan dalam membuat GUI, seperti Tkinter, PyQt, PyGObject, wxPython, dan Pyside.

Pada modul ini akan dicontohkan penggunaan salah satu modul GUI pada Python, yaitu Tkinter. Berikut merupakan penggunaan Tkinter pada GUI.

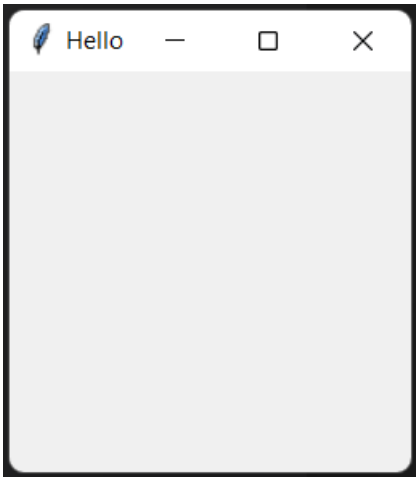
C. Membuat dan Menampilkan Sebuah Form

Form digunakan untuk menampilkan komponen visual yang sudah dibuat menggunakan Tkinter. Selain itu, dapat juga ditambahkan nama untuk form yang akan dibuat. Berikut merupakan contoh *syntax* yang dapat digunakan.

```
import tkinter as tk

mainform = tk.Tk()

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

Gambar 1 *Syntax* Form Kosong dengan Judul

Maka, bentuk *form* yang akan dihasilkan akan menjadi seperti berikut.



Gambar 2 Tampilan Form Kosong dengan Judul

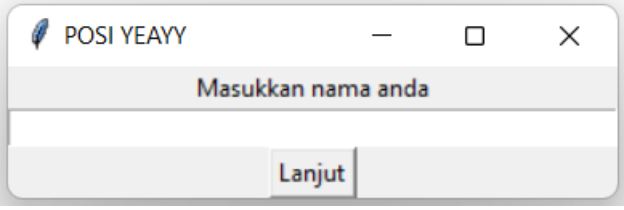
Tampilan tersebut akan berisi kosongan dengan judul di bagian atasnya.

D. Membuat dan Menempatkan Kontrol di Dalam *Form*

Tkinter dapat juga membuat sebuah komponen visual untuk mengontrol dalam pengembangan aplikasi GUI. Sebelum menempatkan komponen kontrol di dalam *form*, dilakukan pembuatan objek dari jenis atau kelas kontrol yang dikehendaki terlebih dahulu. Berikut merupakan contoh *syntax* dalam membuat dan menempatkan komponen kontrol di dalam *form*.

```
import tkinter as tk

mainform = tk.Tk()

#membuat objek berupa label
lbl = tk.Label(mainform)
lbl['text'] = 'Masukkan nama anda'
lbl.pack()

#membuat objek berupa entry
ent = tk.Entry(mainform)
ent['width'] = 50
ent.pack()

#membuat objek button/tombol
tombol = tk.Button(mainform)
tombol['text'] = 'Lanjut'
tombol.pack()

mainform.wm_title('POSI YEAYY')
mainform.mainloop()
```

Gambar 3 *Syntax* Label, *Entry*, dan Tombol



Gambar 4 Tampilah Label, *Entry*, dan Tombol

Gambar di atas merupakan hasil dari eksekusi *syntax* di atas, pada form tersebut, ditampilkan sebuah label berupa perintah "Masukkan nama anda" dan ada sebuah *entry* untuk memasukkan *input* ke dalam *form*. Selain itu, terdapat juga objek tombol yang dapat berinteraksi dengan pengguna dengan cara di klik.

E. Menangani Sebuah Event

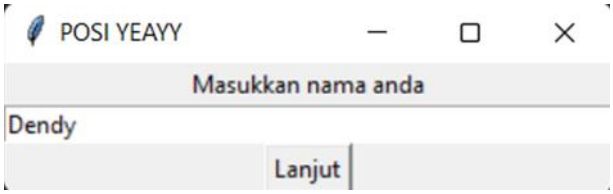
Event adalah sebuah pemberitahuan yang akan dikirim oleh sistem untuk menanggapi *input* dari user terhadap sebuah komponen kontrol tertentu.

Contoh, misalkan jika komponen kontrol tombol "Lanjut" di klik, maka akan menampilkan sebuah pemberitahuan atau mengeksekusi sebuah kondisi fungsi yang sudah didefinisikan dan dipanggil lagi di dalam program. Fungsi tersebut sering disebut dengan *callback* atau *event-handler*.

```
import tkinter as tk
import tkinter.messagebox
mainform = tk.Tk()

#mendefinisikan fungsi
def sapadong() :
    tk.messagebox.showinfo('Hallo', 'Haloo %s, apa kabs?'%(ent.get()))

#membuat objek berupa label
lbl = tk.Label(mainform)
lbl['text'] = 'Masukkan nama anda'
lbl.pack()

#membuat objek berupa entry
ent = tk.Entry(mainform)
ent['width'] = 50
ent.pack()

#membuat objek button/tombol
tombol = tk.Button(mainform,
                   command=sapadong)
tombol['text'] = 'Lanjut'
tombol.pack()

mainform.wm_title('POSI YEAYY')
mainform.mainloop()
```

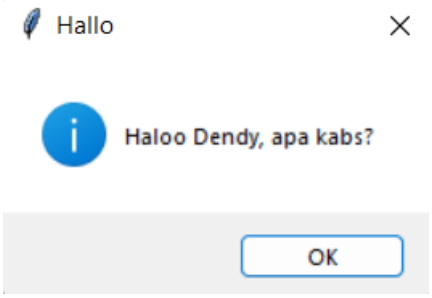
Gambar 5 *Syntax* Penanganan *Event*

Setelah itu, kita coba masukkan nama "Dendy" seperti *form* berikut.



Gambar 6 Tampilan *Form* Penanganan *Event*

Setelah itu, klik tombol "Lanjut" untuk memunculkan event **sapadong()** seperti sebagai berikut.



Gambar 7 Tampilan Hasil dari Penanganan *Event*

F. Mengatur Tampilan Form

Ada 3 metode yang dapat digunakan untuk mengatur tampilan (*layout*) dari kontrol yang terdapat dalam *form*. Ketiga metode tersebut, diantaranya adalah sebagai berikut.

1. *Pack Manager* menggunakan fungsi **pack()**.
2. *Grid Manager* menggunakan fungsi **grid()** yang akan membagi form menjadi tabel dua dimensi yang terdiri dari kolom dan baris. Sel dalam *grid manager* dapat dibentangkan menjadi beberapa kolom dan baris.
3. *Place Manager* menggunakan fungsi **place()**.

Berikut merupakan salah satu contoh penggunaan metode dari ketiga metode di atas, yaitu **grid()**.

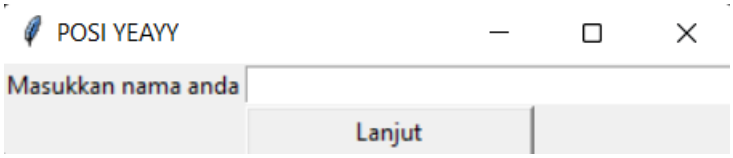
```
import tkinter as tk
import tkinter.messagebox
from tkinter import *
mainform = tk.Tk()

#mendefinisikan fungsi
def sapadong() :
    tk.messagebox.showinfo('Hallo', 'Haloo %s, apa kabs?'%(ent.get()))

#membuat objek berupa label
lbl = tk.Label(mainform)
lbl['text'] = 'Masukkan nama anda'
lbl.grid(row=0, column=0, sticky=tkinter.E)
```

Gambar 8 *Syntax Form* Penanganan *Event*

```
#membuat objek berupa entry
ent = tk.Entry(mainform)
ent['width'] = 40
ent.grid(row=0, column=1, columnspan=2)

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

mainform.wm_title('POSI YEAYY')
mainform.mainloop()
```

Gambar 8 *Syntax Form* Penanganan *Event* (Lanjutan)

Maka akan menampilkan tampilan *form* seperti berikut.



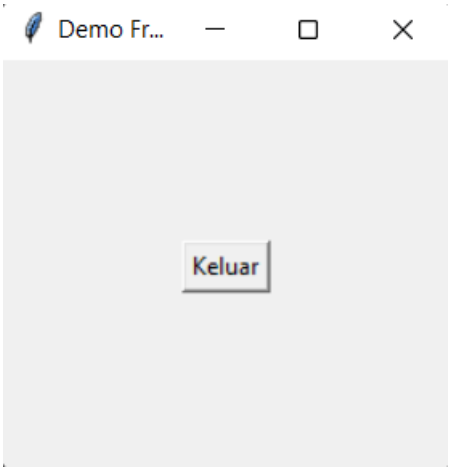
Gambar 9 Tampilan *Form* Penanganan *Event*

G. Menggunakan Kelas *Frame*

Dalam membuat form, dapat digunakan juga kelas *frame*. Objek dari kelas *frame* berperan sebagai wadah ataupun kontainer dari kontrol lain. Contoh penggunaan kelas *frame* adalah sebagai berikut.

```
import tkinter

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

tombol = tkinter.Button(mainform, text = 'Keluar', command = quit)
tombol.grid(sticky=tkinter.E+tkinter.S, padx=90, pady=90)
mainform.master.title('Demo Frame')
mainform.mainloop()
```

Gambar 10 *Syntax Form* Kelas *Frame*



Gambar 11 Tampilan *Form* Kelas *Frame*

H. Komponen Kontrol *Button*/Tombol

Kontrol digunakan untuk membuat tombol dalam *form*. Dalam Tkinter, informasi yang akan ditampilkan untuk mengontrol *Button*. Berikut merupakan beberapa opsi yang dapat digunakan dalam kontrol Tkinter seperti berikut.

Tabel 1 Opsi Kontrol Tkinter

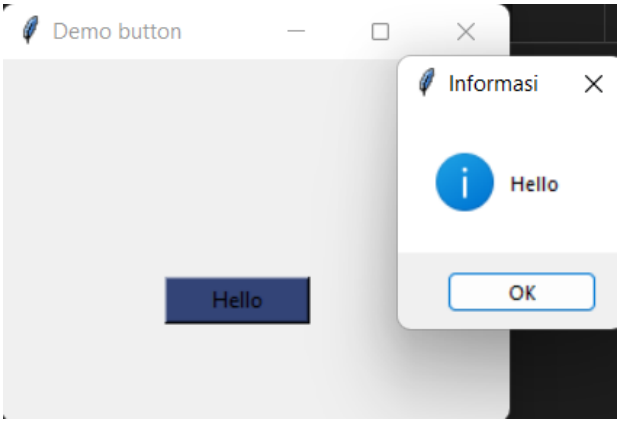
Opsi	Keterangan
Activebackground()	Warna latar ketika tombol ditrkan namun tombol mouse belum dilepas.
Activeforeground()	Warna latar ketika tombol ditrkan namun tombol mouse belum dilepas.
Bd()	Menentukan tebal bingkai
Bg()	Menentukan warna background
Command()	Menentukan fungsi yang akan dipanggil
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 diatas atau dibawah teks
Relief()	Menentukan tampilan tombol
State()	Menentukan status tombol
Text()	Menentukan label atau teks yang akan tampil di dalam tombol
Underline()	Membuat garis bawah pada teks

Width()	Menentukan lebar tombol
Wraplength()	Menentukan lebar teks

Berikut contoh penggunaan beberapa opsi di atas.

```
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='Hello',
                            background='#334477',
                            width = 10,
                            command = buttonclick)
    button.grid(sticky=tkinter.E+tkinter.S, padx = 90, pady=120)
    mainform.mainloop()
if __name__ == '__main__':
    main()
```

Gambar 12 *Syntax* Penggunaan Beberapa Opsi Tkinter



Gambar 13 Tampilan Hasil Penggunaan Beberapa Opsi Tkinter

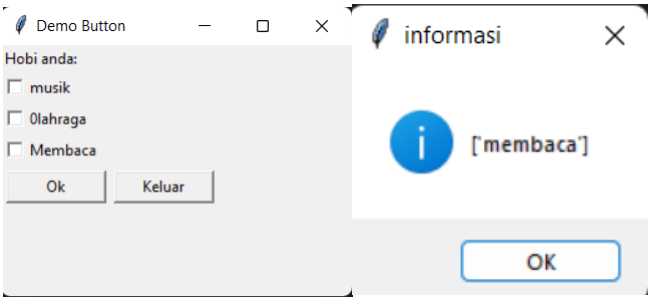
I. Kontrol *Check Button*

Kontrol ini digunakan untuk membuat *checkbox* untuk menampilkan banyak pilihan. Contohnya adalah hobi. Contoh penggunaan kontrol *checkbutton* adalah sebagai berikut.

```
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:')
    I.grid(row=0, sticky=tkinter.W)
    #membuat kontrol checkbutton
    c1 = tkinter.Checkbutton(mainform, variable=var1, text='musik')
    c1.grid (row=1, sticky=tkinter.W)

    c2 = tkinter.Checkbutton(mainform, variable=var2, text='0lahraga')
    c2.grid (row=2, sticky=tkinter.W)

    c3 = tkinter.Checkbutton(mainform, variable=var3, text='Membaca')
    c3.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()
```

Gambar 14 *Syntax* Kontrol *Check Button*



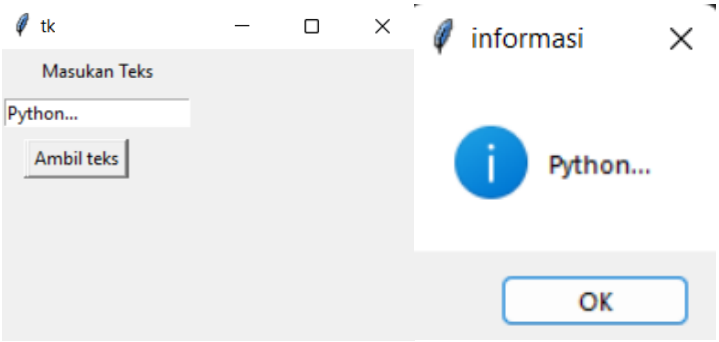
Gambar 15 Tampilan Hasil Kontrol *Check Button*

J. Kontrol *Entry*

Kontrol ini akan digunakan untuk memasukkan informasi yang berupa teks yang diperlukan oleh program. Bentuk umum dari pembuatan kontrol *entry* adalah sebagai berikut.

```
import tkinter
import tkinter.messagebox
def main():
    mainform=tkinter.Tk()
    mainform.geometry("280x200")
    #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='Masukan 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=0, padx=4, pady=4)
    mainform.mainloop()
if __name__ == '__main__':
    main()
```

Gambar 16 *Syntax* Kontrol *Entry*



Gambar 17 *Syntax* Kontrol *Entry*

Referensi

https://www.pythontutorial.net/tkinter/tkinter-grid/

https://www.geeksforgeeks.org/python-place-method-in-tkinter/