Dasar Dasar Pemrograman 1
Kelas Ekstensi
Lab 10 – Graphical User Interface

Deadline : 13 Mei 2017 23:55 WIB

Tutorial kali ini akan membahas mengenai GUI. Sebelum memasuki soal tutorial ikuti langkah-langkah dan penjelasan di bawah untuk mengetahui bagaimana cara implementasi GUI dalam *python*.

Dalam memakai GUI kita harus meng-*import library* yang mendukung implementasinya, pada kesempatan kali ini kita akan memakai modul Tkinter. Cara meng-*import*nya cukup dengan satu kalimat *simple:*

```
Import tkinter
```

Atau lebih biasa dengan kalimat:

```
from tkinter import *
```

Setelah itu kita masukkan ke dalam suatu *class,* hasilnya akan menjadi seperti berikut:

```
from tkinter import *

class simpleapp_tk(Tkinter.Tk):
```

Lalu, kita akan membuat *constructor* pada *class* tersebut dengan menambahkan *def __init__(self,parent).*

```
from tkinter import *

class simpleapp_tk(Tkinter.Tk):
      def __init__(self,parent):
            Tkinter.Tk.__init__(self,parent)
```

Perhatikan, terdapat parameter *parent* dalam init. Hal itu terjadi karena GUI merupakan hirarki dari banyak objek: sebuah *button* berada dalam sebuah panel di mana panel berada dalam sebuah tab di mana tab berada dalam sebuah window, dsb. *Button* mempunyai *parent* yakni panel, panel mempunyai *parent* yaitu tab, dan seterusnya. Karena itu kedua constructor mempunyai parameter *parent*.

Kita akan menambahkan baris baru yang menunjukkan bahwa ia merupakan *reference parent*.

```
from tkinter import *

class simpleapp_tk(Tkinter.Tk):
      def __init__(self,parent):
            Tkinter.Tk.__init__(self,parent)
            self.parent = parent
```

Selanjutnya, kita akan meng-inisialisasi GUI dengan menambahkan statement berikut:

```
from tkinter import *

class simpleapp_tk(Tkinter.Tk):
      def __init__(self,parent):
            Tkinter.Tk.__init__(self,parent)
            self.parent = parent
            self.initialize()

      def initialize(self):
            pass
```

Fungsi *initialize* digunakan untuk membuat semua elemen GUI (*button, text field,* dll.). Untuk saat ini kita hanya akan menampilkan window saja. Setelah itu, kita akan membuat main dari program dengan menambahkan kalimat berikut

```
from tkinter import *

class simpleapp_tk(Tkinter.Tk):
      def __init__(self,parent):
            Tkinter.Tk.__init__(self,parent)
            self.parent = parent
            self.initialize()

      def initialize(self):
            pass

if __name__ == “__main__”:
      app = simpleapp_tk(None)
      app.title(‘my application’)
```

Kenapa paramater simpleapp_tk adalah “*None*”? Karena kita ingin menginisiasi GUI tanpa parent terlebih dahulu. Lalu, *title* berfungsi untuk memberikan judul di *window* yang kita buat.

Sekarang, kita akan membuat program kita *loop* terus menerus dengan menambahkan kalimat berikut:

```
from tkinter import *

class simpleapp_tk(Tkinter.Tk):
      def __init__(self,parent):
            Tkinter.Tk.__init__(self,parent)
            self.parent = parent
            self.initialize()

      def initialize(self):
            pass

if __name__ == “__main__”:
      app = simpleapp_tk(None)
      app.title(‘my application’)
      app.mainloop()
```

Hal ini dilakukan sebagai tanda bahwa program menunggu suatu *event* (klik button, mengetik, dan event lainnya). Tkinter mainloop akan menerima *event* tersebut dan meng-*handle* sesuai dengan *event* yang ditemukan. Kita menamakan ini dengan *event-driven programming* (Karena program tidak akan mengerjakan apa-apa, ia hanya menunggu suatu *event* dan hanya berjalan ketika menerima *event* tersebut). Saat ini program kita sudah bisa berjalan dan hanya menampilkan windows kosong.

Lalu, kita akan mencoba memasukkan *layout manager* ke dalam GUI. Ada beberapa tipe layout silahkan dipelajari sendiri, untuk saat ini kita akan memakai layout *grid.* Layout *grid* mudah dipahami karena untuk memasukkan *widget* ke dalam *layout* ini sama seperti kerja *spreadsheet*. Contohnya: masukkan *button* pada kolom 2, baris 1. Masukkan *checkbox* pada kolom 5, baris 3, dan seterusnya. Untuk menambahkan hal ini cukup menambahkan kalimat berikut:

```
from tkinter import *

class simpleapp_tk(Tkinter.Tk):
      def __init__(self,parent):
            Tkinter.Tk.__init__(self,parent)
            self.parent = parent
            self.initialize()

      def initialize(self):
            self.grid()

if __name__ == "__main__":
      app = simpleapp_tk(None)
      app.title('my application')
      app.mainloop()
```

Setelah memberikan layout pada GUI, kita akan mencoba menambahkan *widget* di dalamnya dengan menambahkan beberapa kalimat berikut:

```
from tkinter import *

class simpleapp_tk(Tkinter.Tk):
      def __init__(self,parent):
            Tkinter.Tk.__init__(self,parent)
            self.parent = parent
            self.initialize()

      def initialize(self):
            self.grid()

            self.entry = Tkinter.Entry(self)
            self.entry.grid(column=0,row=0,sticky='EW')

if __name__ == "__main__":
      app = simpleapp_tk(None)
      app.title('my application')
      app.mainloop()
```

Perhatikan kode di atas, untuk menambahkan *widget* cukup dengan menginisialisasi *widget* dari modul Tkinter lalu masukkan ke dalam layout. Dalam hal ini, kita membuat *widget text entry* lalu kita masukkan ke *grid* (column=0, row=0), sticky digunakan untuk meng-*handle* jika *layout* yang digunakan sudah melebihi kapasitas yang ditampungnya. EW artinya East West, yang artinya *layout* akan meng-*expand* ke ujung kiri dan kanan *cell* jika layout sudah melebihi kapasitas.

Selanjutnya, kita akan mencoba menambahkan *widget button*.

```
from tkinter import *

class simpleapp_tk(Tkinter.Tk):
      def __init__(self,parent):
            Tkinter.Tk.__init__(self,parent)
            self.parent = parent
            self.initialize()

      def initialize(self):
            self.grid()

            self.entry = Tkinter.Entry(self)
            self.entry.grid(column=0,row=0,sticky='EW')

            button = Tkinter.Button(self,text=u"Click me !")
            button.grid(column=1,row=0)

if __name__ == "__main__":
      app = simpleapp_tk(None)
      app.title('my application')
      app.mainloop()
```

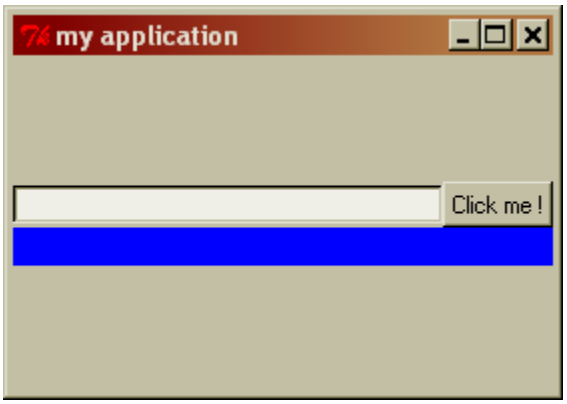
Setelah itu, kita akan menambahkan *widget* label.

```
from tkinter import *

class simpleapp_tk(Tkinter.Tk):
      def __init__(self,parent):
            Tkinter.Tk.__init__(self,parent)
            self.parent = parent
            self.initialize()

      def initialize(self):
            self.grid()

            self.entry = Tkinter.Entry(self)
            self.entry.grid(column=0,row=0,sticky='EW')

            button = Tkinter.Button(self,text=u"Click me !")
            button.grid(column=1,row=0)

            label = Tkinter.Label(self,
                                  anchor="w",fg="white",bg="blue")
            label.grid(column=0,row=1,columnspan=2,sticky='EW')

if __name__ == "__main__":
      app = simpleapp_tk(None)
      app.title('my application')
      app.mainloop()
```

Setelah memasukkan beberapa *widget,* kita akan membuat *layout* kita dapat mengubah ukurannya sendiri ketika kita mengubah ukuran *windows* dengan menambahkan kalimat berikut:

```
from tkinter import *

class simpleapp_tk(Tkinter.Tk):
      def __init__(self,parent):
            Tkinter.Tk.__init__(self,parent)
            self.parent = parent
            self.initialize()

      def initialize(self):
            self.grid()

            self.entry = Tkinter.Entry(self)
            self.entry.grid(column=0,row=0,sticky='EW')

            button = Tkinter.Button(self,text=u"Click me !")
            button.grid(column=1,row=0)

            label = Tkinter.Label(self,
                                  anchor="w",fg="white",bg="blue")
            label.grid(column=0,row=1,columnspan=2,sticky='EW')

            self.grid_columnconfigure(0,weight=1)

if __name__ == "__main__":
      app = simpleapp_tk(None)
      app.title('my application')
      app.mainloop()
```

Program kita akan terlihat seperti Gambar 1,



Gambar 1. Tampilan Aplikasi.

Jika kita meresize windows secara vertikal hasilnya akan menjadi kurang bagus, tetapi untuk saat ini kita tidak akan meng-*handle* hal itu.

Selanjutnya kita akan menambahkan *event handler*. *Event handler* yang akan ditambahkan yaitu *OnButtonClick()* dan *OnPressEnter()* dengan menambahkan beberapa kalimat berikut:

```
from tkinter import *

class simpleapp_tk(Tkinter.Tk):
      def __init__(self,parent):
            Tkinter.Tk.__init__(self,parent)
            self.parent = parent
            self.initialize()

      def initialize(self):
            self.grid()

            self.entry = Tkinter.Entry(self)
            self.entry.grid(column=0,row=0,sticky='EW')
            self.entry.bind("<Return>", self.OnPressEnter)

            button = Tkinter.Button(self,text=u"Click me !",
                                    command=self.OnButtonClick)
            button.grid(column=1,row=0)

            label = Tkinter.Label(self,
                                    anchor="w",fg="white",bg="blue")
            label.grid(column=0,row=1,columnspan=2,sticky='EW')

            self.grid_columnconfigure(0,weight=1)

      def OnButtonClick(self):
            print "You clicked the button !"

      def OnPressEnter(self,event):
            print "You pressed enter !"

if __name__ == "__main__":
      app = simpleapp_tk(None)
      app.title('my application')
      app.mainloop()
```

Sampai saat ini jika kita mengklik *button* atau menekan *Enter* akan muncul pesan pada cmd, kita akan mencoba merubah label jika mengklik *button* atau menekan *Enter* dengan mengganti beberapa kalimat berikut:

```
from tkinter import *

class simpleapp_tk(Tkinter.Tk):
      def __init__(self,parent):
            Tkinter.Tk.__init__(self,parent)
            self.parent = parent
            self.initialize()

      def initialize(self):
            self.grid()

            self.entry = Tkinter.Entry(self)
            self.entry.grid(column=0,row=0,sticky='EW')
            self.entry.bind("<Return>", self.OnPressEnter)

            button = Tkinter.Button(self,text=u"Click me !",
                                    command=self.OnButtonClick)
            button.grid(column=1,row=0)

            self.labelVariable = Tkinter.StringVar()
            label = Tkinter.Label(self,textvariable=self.labelVariable,
                                    anchor="w",fg="white",bg="blue")
            label.grid(column=0,row=1,columnspan=2,sticky='EW')

            self.grid_columnconfigure(0,weight=1)

      def OnButtonClick(self):
            self.labelVariable.set("You clicked the button !")

      def OnPressEnter(self,event):
            self.labelVariable.set("You pressed enter !")

if __name__ == "__main__":
      app = simpleapp_tk(None)
      app.title('my application')
      app.mainloop()
```

Selamat! anda hampir menyelesaikan tutorial (belum soal) dengan mengganti label yang ada. Sekarang, kita akan coba menampilkan *value text field* di label dengan menambahkan kalimat berikut:

```
from tkinter import *

class simpleapp_tk(Tkinter.Tk):
      def __init__(self,parent):
            Tkinter.Tk.__init__(self,parent)
            self.parent = parent
            self.initialize()

      def initialize(self):
            self.grid()

            self.entryVariable = Tkinter.StringVar()
            self.entry = Tkinter.Entry(self,textvariable=self.entryVariable)
            self.entry.grid(column=0,row=0,sticky='EW')
            self.entry.bind("<Return>", self.OnPressEnter)
            self.entryVariable.set(u"Enter text here.")

            button = Tkinter.Button(self,text=u"Click me !",
                                    command=self.OnButtonClick)
            button.grid(column=1,row=0)

            self.labelVariable = Tkinter.StringVar()
            label = Tkinter.Label(self,textvariable=self.labelVariable,
                                    anchor="w",fg="white",bg="blue")
            label.grid(column=0,row=1,columnspan=2,sticky='EW')
            self.labelVariable.set(u"Hello !")

            self.grid_columnconfigure(0,weight=1)

      def OnButtonClick(self):
            self.labelVariable.set(self.entryVariable.get()+ " (You clicked
the button) ")

      def OnPressEnter(self,event):
            self.labelVariable.set(self.entryVariable.get()+ " (You pressed
ENTER) ")

if __name__ == "__main__":
      app = simpleapp_tk(None)
      app.title('my application')
      app.mainloop()
```

Tambahan:
Sampai tadi kalian sudah bisa melanjutkan ke soal tetapi jika ingin menambah wawasan mengenai GUI, kalian bisa mengikuti satu tutorial tambahan berikut. Tutorial ini mengajarkan kita untuk menambahkan *auto-select* dalam *text field* dan meng-*handle resize* pada windows ketika kita mengetik "looooooooooong" pada *text field, windows* akan membesar begitu juga ketika kita mengetik kata yang pendek *window* akan mengecil. Hal-hal tersebut bisa kita *handle* dengan menambahkan beberapa kalimat berikut:

```
from tkinter import *

class simpleapp_tk(Tkinter.Tk):
    def __init__(self,parent):
        Tkinter.Tk.__init__(self,parent)
        self.parent = parent
        self.initialize()

    def initialize(self):
        self.grid()

        self.entryVariable = Tkinter.StringVar()
        self.entry = Tkinter.Entry(self,textvariable=self.entryVariable)
        self.entry.grid(column=0,row=0,sticky='EW')
        self.entry.bind("<Return>", self.OnPressEnter)
        self.entryVariable.set(u"Enter text here.")

        button = Tkinter.Button(self,text=u"Click me
!",command=self.OnButtonClick)
        button.grid(column=1,row=0)

        self.labelVariable = Tkinter.StringVar()
        label =
Tkinter.Label(self,textvariable=self.labelVariable,anchor="w",fg="white",bg="
blue")
        label.grid(column=0,row=1,columnspan=2,sticky='EW')
        self.labelVariable.set(u"Hello !")

        self.grid_columnconfigure(0,weight=1)
        self.resizable(True,False)
        self.update()
        self.geometry(self.geometry())
        self.entry.focus_set()
        self.entry.selection_range(0, Tkinter.END)

    def OnButtonClick(self):
        self.labelVariable.set(self.entryVariable.get() + " (You clicked the
button) ")
        self.entry.focus_set()
        self.entry.selection_range(0, Tkinter.END)
```

```
    def OnPressEnter(self, event):
        self.labelVariable.set(self.entryVariable.get() + " (You pressed
ENTER) ")
        self.entry.focus_set()
        self.entry.selection_range(0, Tkinter.END)

if __name__ == "__main__":
    app = simpleapp_tk(None)
    app.title('my application')
    app.mainloop()
```

----------END OF TUTORIAL----------