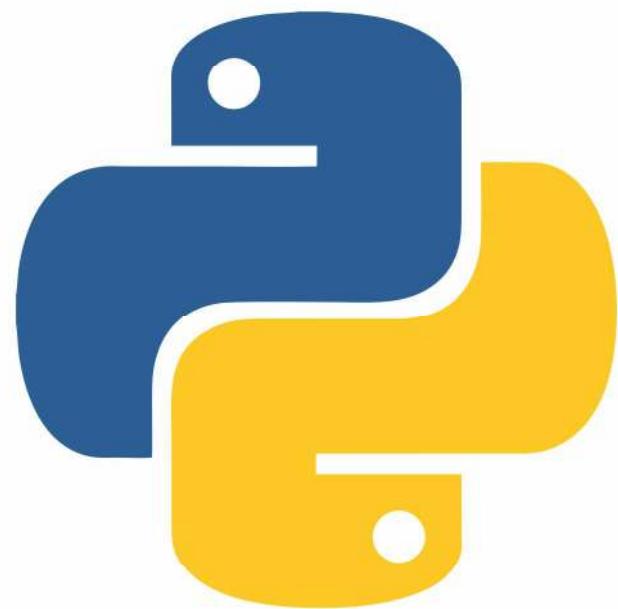


PYTHON

Pemrograman IV



PYTHON

Tkinter

Tkinter adalah graphic user interface (GUI) standar python digunakan untuk membuat tampilan aplikasi dengan komponen-komponen yang ada di modul tkinter seperti Button, Textbox, Label, Frame, Window yang mana sangat mendukung dalam penciptaan aplikasi GUI. Untuk dapat memahami cara kerja **Tkinter** ini Anda harus paham menggunakan paradigma **Object Oriented Programming** dalam modul sebelumnya.

13.1 Komponen-Komponen Tkinter

KOMPONEN	DESKRIPSI
Button	Komponen Button berfungsi untuk menampilkan sebuah tombol.
Canvas	Komponen Canvas digunakan untuk menggambar bentuk seperti garis, lingkaran, poligon, dan kotak.
Checkbutton	Komponen Checkbutton berfungsi menampilkan sejumlah pilihan yang ditandai dengan tanda 'centang'. Pengguna aplikasi dapat memilih lebih dari satu pilihan.
Entry	Komponen Entry berfungsi menampilkan kotak teks satu-baris untuk menerima masukan dari pengguna.
Frame	Komponen Frame berfungsi sebagai kontainer bagi komponen lain.
Label	Komponen Label berfungsi memberikan keterangan untuk komponen lain. Komponen ini juga dapat diisi gambar.
Listbox	Komponen Listbox berfungsi menyediakan daftar pilihan untuk pengguna.
Menobutton	Komponen Menobutton berfungsi menampilkan menu.
Menu	Komponen Menu berfungsi memberikan berbagai perintah kepada pengguna. Perintah-perintah ini juga tercantum dalam Menobutton.
Message	Komponen Message berfungsi menampilkan teks yang terdiri dari sejumlah baris untuk menerima beberapa nilai dari pengguna.
Radiobutton	Komponen Radiobutton berfungsi menampilkan sejumlah pilihan dalam bentuk tombol radio.
Scale	Komponen Scale berfungsi menampilkan skala geser.
Scrollbar	Komponen Scrollbar berguna untuk menambahkan fungsi geser (scroll) pada beberapa komponen, seperti komponen Listbox.
Text	Komponen Text berfungsi menampilkan teks dalam multi baris.
Toplevel	Komponen Toplevel digunakan untuk membuat sebuah kontainer window yang terpisah.

PYTHON

13.2 Memanggil Modul Tkinter

Modul Tkinter berisi berbagai kelas dan komponen yang diperlukan untuk membangun sebuah aplikasi GUI. Untuk menggunakan komponen Tkinter, kita harus memanggilnya terlebih dahulu. Contoh di bawah ini akan menunjukkan bagaimana kita memanggil modul Tkinter.

```
import Tkinter
```

Kita juga bisa memanggil semua metode, kelas, dan atribut dari modul Tkinter dengan menggunakan cara seperti di bawah ini.

```
from Tkinter import *
```

13.3 Membuat Jendela Aplikasi GUI

Setiap aplikasi GUI seharusnya memiliki sebuah jendela utama (root window), yang dapat berisi berbagai komponen yang diperlukan dalam aplikasi yang akan dibuat. Untuk membuat jendela utama dari suatu aplikasi, kita menggunakan pernyataan seperti berikut:

```
root = Tkinter.Tk()
```

Jika kita menggunakan pernyataan from-import untuk memanggil modul Tkinter, maka kita cukup menuliskan :

```
root = Tk()
```

13.4 Menambahkan Komponen Pada Aplikasi

Dengan menggunakan Tkinter, kita dapat menambahkan beberapa komponen pada aplikasi yang kita buat. Komponen ini bisa berdiri sendiri atau dapat diisi komponen lain (disebut Container). Komponen yang berdiri sendiri merupakan komponen yang tidak dapat diisi oleh komponen lain, misalkan tombol, cekbox, dan label. Komponen container merupakan komponen yang bisa berisi/ditempeli oleh komponen lain seperti frame dan window.

13.5 Menambahkan Kode mainLoop

Setelah kita mendesain sebuah aplikasi, tentu saja dengan menambahkan beberapa komponen di dalamnya, kita perlu mengeksekusi program tersebut. Ketika sebuah program kita eksekusi, maka program tersebut akan mengalami perputaran tak berhingga (infinite loop). Kode untuk memasukkan program kita dalam kondisi perputaran ini yaitu :

```
Tkinter.mainloop()
```

PYTHON

Jika kita menggunakan from-import untuk mengambil semua fungsi pada Tkinter, kita cukup menuliskan

```
root.mainloop()
```

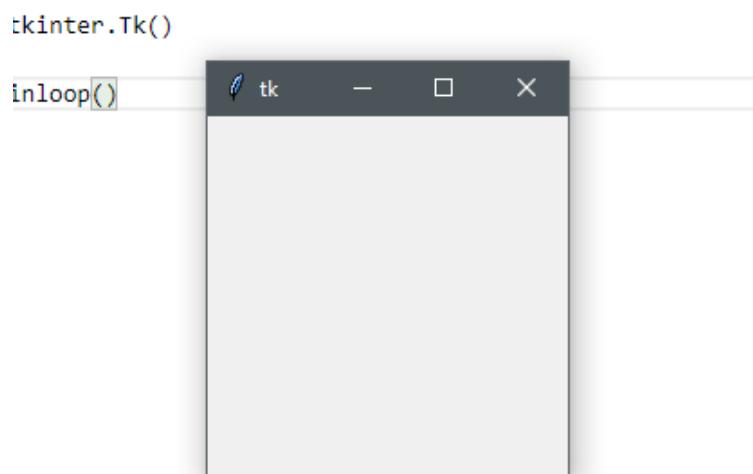
Dengan root pada kode di atas mengacu pada window utama (toplevel). Berikut ini merupakan sebuah listing program untuk menampilkan sebuah window.

13.6 Tampilan GUI Sederhana

```
tkinter_blank_gui1.py ▶ ...
1  import tkinter
2
3  root = tkinter.Tk()
4
5  root.mainloop()
```

Gambar 97 Kode Tampilan GUI Sederhana

Setelah di-run akan tampil seperti gambar dibawah.



Gambar 98 Tampilan GUI Sederhana

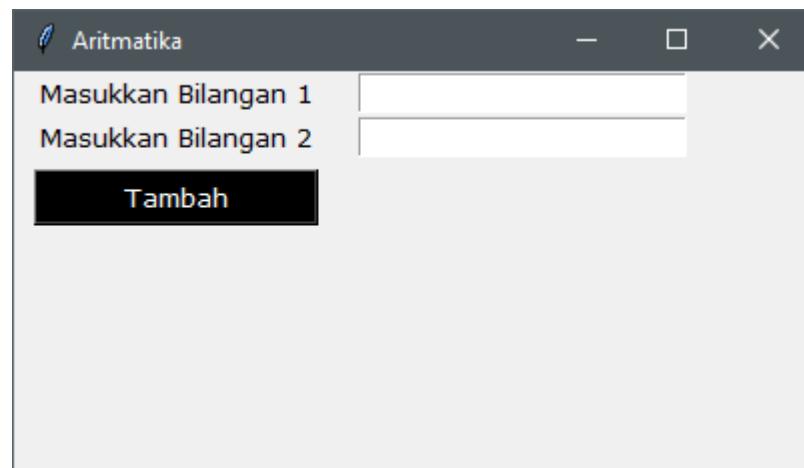
PYTHON

13.7 Contoh Aplikasi Aritmatika Pertambahan

```
tkinter_aritmatika.py ▶ ...
1  import tkinter as tk
2  from functools import partial
3
4  def pertambahan(labelHasil, bil1, bil2):
5      b1 = int(bil1.get())
6      b2 = int(bil2.get())
7      hasil = b1 + b2
8      # config digunakan untuk mengakses object atribut setelah inisialisasi
9      labelHasil.config(text=hasil)
10     return
11
12 root = tk.Tk()
13
14 # 400x200 adalah lebar dan tinggi window
15 # 500 adalah posisi secara horizontal
16 # 200 adalah posisi secara vertikal
17 root.geometry('400x200+500+200')
18
19 # mengubah font
20 root.option_add('*font', ('Verdana', 10, 'normal'))
21 |
22 # untuk tampilkan title di window border
23 root.title('Aritmatika')
24
25 # configure untuk mengubah tampilan warna
26 #root.configure(bg="#FFFFFF")
27
28 labelBilangan1 = tk.Label(root, text="Masukkan Bilangan 1")
29 labelBilangan1.grid(row=0, column=0, padx=(10,20))
30 labelBilangan2 = tk.Label(root, text="Masukkan Bilangan 2")
31 labelBilangan2.grid(row=1, column=0, padx=(10,20))
32 # StringVar() digunakan untuk menampung inputan tipe String
33 bilangan1 = tk.StringVar()
34 bilangan2 = tk.StringVar()
35
36
37 inputBilangan1 = tk.Entry(root, textvariable=bilangan1)
38 inputBilangan1.grid(row=0, column=1)
39 inputBilangan2 = tk.Entry(root, textvariable=bilangan2)
40 inputBilangan2.grid(row=1, column=1)
41
42 labelHasil = tk.Label(root)
43 labelHasil.grid(row=2, column=1)
44
45 # functools.partial untuk membuat fungsi baru atau fungsi versi baru dengan argumen
46 pertambahan = partial(pertambahan, labelHasil, bilangan1, bilangan2)
47 tombolTambah = tk.Button(root, text="Tambah", command=pertambahan)
48 # sticky digunakan untuk penyesuaian widget di dalam cell
49 # sticky="W" yang artinya West adalah posisi widget di kiri (di dalam cell)
50 # sticky="E" yang artinya East adalah posisi widget di kanan (di dalam cell)
51 # sticky="WE" artinya memenuhi cell atau alignment rata penuh
52 # padx adalah padding horizontal
53 # pady adalah padding vertikal
54 tombolTambah.grid(row=2, column=0, sticky="WE", padx=(10,20), pady=(5,0))
55 tombolTambah.configure(bg="#000", fg="#FFF")
56
57 root.mainloop()
```

Gambar 99 Kode Aplikasi Aritmatika Pertambahan

PYTHON



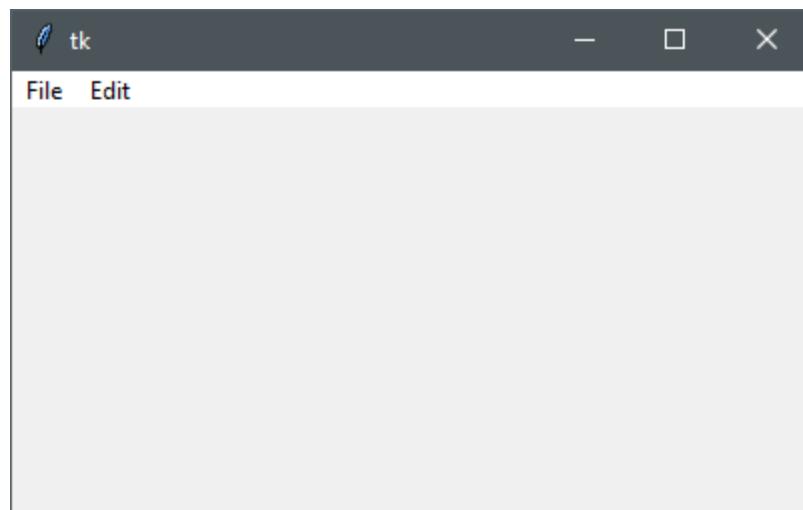
Gambar 100 Aplikasi Aritmatika Pertambahan

13.8 Membuat Menu

```
tkinter_menu.py > ...
1  import tkinter as tk
2
3  root = tk.Tk()
4  root.geometry('400x200+500+200')
5
6  menubar = tk.Menu(root)
7
8  file = tk.Menu(menubar, tearoff=0)
9
10 # add_cascade digunakan untuk menampilkan submenu
11 menubar.add_cascade(label="File", menu=file)
12
13 # add_command digunakan untuk menambah menu item
14 file.add_command(label="New")
15 file.add_command(label="Open")
16 file.add_command(label="Save")
17 file.add_command(label="Close")
18
19 # add_separator untuk menampilkan pemisah
20 file.add_separator()
21 file.add_command(label="Exit", command=root.quit)
22
23 edit = tk.Menu(menubar, tearoff=0)
24 menubar.add_cascade(label="Edit", menu=edit)
25 edit.add_command(label="Cut")
26 edit.add_command(label="Copy")
27 edit.add_command(label="Paste")
28
29 # menampilkan menu
30 root.config(menu=menubar)
31 root.mainloop()
```

Gambar 101 Kode Tampilan Menu

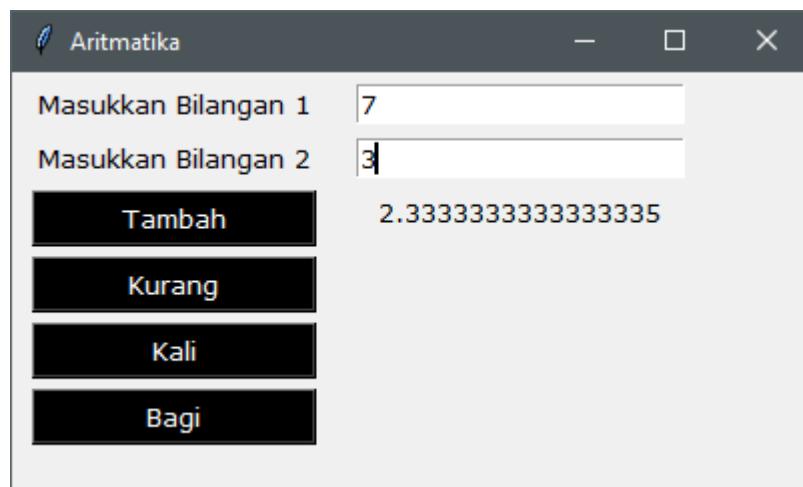
PYTHON



Gambar 102 Tampilan Menu

13.9 Latihan

Buatlah aplikasi aritmatika pertambahan, pengurangan, perkalian, dan pembagian seperti gambar dibawah.



Gambar 103 Aplikasi Aritmatika Lanjut