LAPORAN PRAKTIKUM
PRAKTIK PEMROGRAMAN PYTHON

Disusun Oleh :

Maharrani Syah (V3922028)

Melia Madzrongatul Khoiriyah (V3922030)

Ulfiatul Khusna Sani (V3922043)

Winasis Widya Wisesa (V3922048)

Zulfa Aulia (V3922049)

Dosen :

Yusuf Fadilla Rachman S.Kom., M.Kom

PS D-III TEKNIK INFORMATIKA
SEKOLAH VOKASI
UNIVERSITAS SEBELAS MARET
2023

BAB I

PENDAHULUAN

1.1 Latar Belakang

Tkinter adalah modul bawaan Python yang digunakan untuk membuat antarmuka grafis pengguna (GUI). Tkinter didasarkan pada toolkit Tk, yang awalnya ditulis untuk bahasa pemrograman Tcl (Tool Command Language). Tkinter memungkinkan pengembang untuk membuat jendela, tombol, bidang input, dan elemen GUI lainnya yang dapat berinteraksi dengan pengguna.

1.2 Tujuan

1. Tkinter memungkinkan pengembang untuk membuat aplikasi desktop dengan antarmuka yang menarik dan berfungsi penuh. Dengan menggunakan berbagai widget dan tata letak yang disediakan oleh Tkinter, pengembang dapat merancang jendela, tombol, bidang input, dan elemen GUI lainnya.
2. Tkinter memfasilitasi interaksi antara pengguna dan aplikasi. Pengguna dapat memberikan masukan melalui tombol, kotak teks, pilihan, dan lain sebagainya.
3. Tkinter dapat digunakan untuk membuat grafik, diagram, dan tampilan visual lainnya untuk mempresentasikan data kepada pengguna.

1.3 Manfaat

1. Tkinter dirancang dengan keberlanjutan dan kesederhanaan dalam pikiran. Antarmuka Python yang intuitif dan dokumentasi yang luas membuatnya mudah dipelajari dan digunakan oleh pengembang pemula. Ini memungkinkan pengembang untuk membuat antarmuka pengguna dengan cepat dan efisien.
2. Tkinter dapat berjalan di berbagai platform seperti Windows, macOS, dan Linux. Ini memungkinkan pengembang untuk menghasilkan aplikasi yang konsisten di berbagai sistem operasi tanpa perlu mengubah kode sumber utama.
3. Tkinter terintegrasi dengan baik dengan bahasa pemrograman Python. Dengan menggunakan Python, pengembang dapat mengakses berbagai pustaka dan fitur yang tersedia, memungkinkan pengembangan aplikasi yang kompleks dengan kemampuan yang diperluas.

1.4 Alat dan Bahan

1. Laptop
2. Internet
3. Spyder LED
4. Jupyter notebook

BAB II

TINJAUAN PUSTAKA

2.1. GUI (Graphical User Interface)

GUI merupakan salah satu teknologi yang dibuat dan di teliti sejak lama, akan tetapi publik baru menikmatinya di awal tahun 1980an. Penggunaan dari teknologi ini merupakan sebuah teknologi yang digunakan untuk menggantikan sistem yang lama yang sering disebut dengan CLI. CLI atau Command Line Interface merupakan sebuah teknologi yang dibangun untuk memperbolehkan user memberikan perintah pada komputer dalam bentuk bahasa pemrograman. Dengan menggunakan CLI ini, inputan yang bisa diterima oleh komputer berasal dari keyboard saja. GUI merupakan salah satu jenis user interface yang digunakan untuk melakukan komunikasi antara manusia dengan perangkat seperti laptop, komputer, ponsel dan tablet. Hal ini menjadikan komponen GUI selalu berhubungan dengan representasi visual dari sebuah sistem operasi ataupun software.

2.2. Python

Python adalah bahasa pemrograman yang populer. Bahasa pemrograman ini dibuat oleh Guido van Rossum dan dikenalkan sejak tahun 1991. Python termasuk bahasa pemrograman yang mudah untuk dipelajari. Sampai saat ini bahasa pemrograman Python hampir dipakai di segala bidang seperti game, sistem berbasis web, dan bahkan dapat membuat mesin pencari sendiri. Jadi secara umum, bahasa pemrograman ini dipakai dalam pengembangan website, pengembangan software, matematika, dan system scripting.

2.3 Tkinter

Tkinter adalah modul python bawaan yang digunakan untuk membuat aplikasi GUI. Ini adalah salah satu modul yang paling umum digunakan untuk membuat aplikasi GUI dengan Python karena sederhana dan mudah digunakan. Anda tidak perlu khawatir tentang pemasangan modul Tkinter secara terpisah karena sudah dilengkapi dengan Python. Ini memberikan antarmuka berorientasi objek ke toolkit Tk GUI.

Berikut adalah beberapa kasus penggunaan umum untuk Tkinter:

1. Membuat jendela dan kotak dialog: Tkinter dapat digunakan untuk membuat jendela dan kotak dialog yang memungkinkan pengguna berinteraksi dengan program Anda. Ini dapat digunakan untuk menampilkan informasi, mengumpulkan input, atau menampilkan opsi kepada pengguna.
2. Membangun GUI untuk aplikasi desktop: Tkinter dapat digunakan untuk membuat antarmuka untuk aplikasi desktop, termasuk tombol, menu, dan elemen interaktif lainnya.
3. Menambahkan GUI ke program baris perintah: Tkinter dapat digunakan untuk menambahkan GUI ke program baris perintah, sehingga memudahkan pengguna untuk berinteraksi dengan program dan memasukkan argumen.
4. Membuat widget khusus: Tkinter menyertakan berbagai widget bawaan, seperti tombol, label, dan kotak teks, tetapi juga memungkinkan Anda membuat widget khusus sendiri.
5. Membuat Prototipe GUI: Tkinter dapat digunakan untuk membuat prototipe GUI dengan cepat, memungkinkan Anda untuk menguji dan mengulangi ide desain yang berbeda sebelum melakukan implementasi akhir.

BAB III
HASIL DAN PEMBAHASAN

1.Source Code:

```
from tkinter import *
import tkinter as tk
from datetime import datetime
from PIL import ImageTk, Image
from tkinter import messagebox
from tkinter import filedialog
from tkinter.scrolledtext import ScrolledText
import random


class laptop_management():



        # =============== Total Bill Code ==================

        def Total_Bill(self):
                self.Asus_Aspire_5_price = 7000000
                self.Asus_VivoBook_14_price = 8500000
                self.HP_Pavilion_x360_price = 10000000
                self.Lenovo_ThinkPadX1_Carbon_price = 22000000
                self.MSI_GS66_Stealth_price = 35000000
                self.Razer_Blade_15_price = 32000000
                self.MacBook_Air_M1_price = 17000000


                if self.Asus_Aspire_5_item.get() != "":
                        self.Asus_Aspire_cost = self.Asus_Aspire_5_price *
int(self.Asus_Aspire_5_item.get())
                else:
                        self.Asus_Aspire_cost = 0

                if self.Asus_VivoBook_14_item.get() != "":
                        self.Asus_VivoBook_cost = self.Asus_VivoBook_14_price
* int(self.Asus_VivoBook_14_item.get())
                else:
                        self.Asus_VivoBook_cost = 0

                if self.HP_Pavilion_x360_item.get() != "":
                        self.HP_Pavilion_cost = self.HP_Pavilion_x360_price *
int(self.HP_Pavilion_x360_item.get())
                else:
                        self.HP_Pavilion_cost = 0
```

```
                if self.Lenovo_ThinkPadX1_Carbon_item.get() != "":
                        self.Lenovo_TPX1_Carbon_cost =
self.Lenovo_ThinkPadX1_Carbon_price *
int(self.Lenovo_ThinkPadX1_Carbon_item.get())
                else:
                        self.Lenovo_TPX1_Carbon_cost = 0

                if self.MSI_GS66_Stealth_item.get() != "":
                        self.MSI_GS66_Stealth_cost =
self.MSI_GS66_Stealth_price * int(self.MSI_GS66_Stealth_item.get())
                else:
                        self.MSI_GS66_Stealth_cost = 0

                if self.Razer_Blade_15_item.get() != "":
                        self.Razer_Blade_15_cost = self.Razer_Blade_15_price *
int(self.Razer_Blade_15_item.get())
                else:
                        self.Razer_Blade_15_cost = 0

                if self.MacBook_Air_M1_item.get() != "":
                        self.MacBook_Air_M1_cost = self.MacBook_Air_M1_price *
int(self.MacBook_Air_M1_item.get())
                else:
                        self.MacBook_Air_M1_cost = 0


                self.Total_Bill = self.MacBook_Air_M1_cost +
self.Razer_Blade_15_cost + self.MSI_GS66_Stealth_cost +
self.Lenovo_TPX1_Carbon_cost + self.HP_Pavilion_cost + self.Asus_VivoBook_cost
+ self.Asus_Aspire_cost


                if self.items_cost != "":
                        self.items_cost.delete(0,END)
                        self.items_cost.insert(END,self.Total_Bill)
                else:
                        self.items_cost.insert(END,self.Total_Bill)

                if self.antivirus_cost != "":
                        self.antivirus_cost.delete(0,END)
                        self.antivirus_cost.insert(END,50000.0)
                else:
                        self.antivirus_cost.insert(END,50000.0)

                if self.total_bill != "":
                        self.total_bill.delete(0,END)
                        self.total_bill.insert(END,int(self.items_cost.get())+
float(self.antivirus_cost.get()))
```

```
                else:
                        self.total_bill.insert(END,int(self.items_cost.get())+
float(self.antivirus_cost.get()))


                date = datetime.now().date()
                if self.bill_details.get(1.0, "end") != "":
                        self.bill_details.delete(1.0, "end")
                        self.bill_details.insert(
                            1.0,
                            f" Billno-{random.randint(100, 1000)}\t{date}  \n
Items(q) ======= \tAmount  ======= \n {'Asus Aspire 5
('+str(self.Asus_Aspire_5_item.get()) + ')' + ' ============> ' +
str(self.Asus_Aspire_cost) + ' '  if self.Asus_Aspire_5_item.get() != 0 else
''}\n{' Asus VivoBook 14 ('+str(self.Asus_VivoBook_14_item.get()) + ')' + '
============> ' + str(self.Asus_VivoBook_cost) + '  '  if
self.Asus_VivoBook_14_item.get() != 0 else ''}\n{ ' HP Pavilion x360
('+str(self.HP_Pavilion_x360_item.get()) + ')' + ' ============> ' +
str(self.HP_Pavilion_cost) + '  '  if self.HP_Pavilion_x360_item.get() != 0
else ''}\n{' Lenovo TPX1 Carbon
('+str(self.Lenovo_ThinkPadX1_Carbon_item.get()) + ')' + ' ============> ' +
str(self.Lenovo_TPX1_Carbon_cost) + ' '  if
self.Lenovo_ThinkPadX1_Carbon_item.get() != 0 else ''}\n{' MSI GS66
Stealth('+str(self.MSI_GS66_Stealth_item.get()) + ')' + ' ============> ' +
str(self.MSI_GS66_Stealth_cost) + ' '  if self.MSI_GS66_Stealth_item.get() !=
0 else ''}\n{' Razer Blade 15('+str(self.Razer_Blade_15_item.get()) + ')' + '
============> ' + str(self.Razer_Blade_15_cost) + ' '  if
self.Razer_Blade_15_item.get() != 0 else ''}\n{' MacBook Air
M1('+str(self.MacBook_Air_M1_item.get()) + ')' + ' ============> ' +
str(self.MacBook_Air_M1_cost) + ' '  if self.MacBook_Air_M1_item.get() != 0
else ''}\n\n Item Cost ------- {self.items_cost.get()} \n Antivirus Cost -----
-- {self.antivirus_cost.get()} \n=====================\n Total -------
{self.total_bill.get()}\n ====================="
                        )



        # ====== Calculator Code ========

        def nine(self):
                if "error" in self.result.get() or '=' in self.result.get():
                        self.result.delete(0, "end")
                        self.result.insert("end", "9")
                else:
                        self.result.insert("end", "9")

        def eight(self):
                if "error" in self.result.get() or '=' in self.result.get():
```

```
                        self.result.delete(0, "end")
                        self.result.insert("end", "8")
                else:
                        self.result.insert("end", "8")

        def seven(self):
                if "error" in self.result.get() or '=' in self.result.get():
                        self.result.delete(0, "end")
                        self.result.insert("end", "7")
                else:
                        self.result.insert("end", "7")

        def six(self):
                if "error" in self.result.get() or '=' in self.result.get():
                        self.result.delete(0, "end")
                        self.result.insert("end", "6")
                else:
                        self.result.insert("end", "6")

        def five(self):
                if "error" in self.result.get() or '=' in self.result.get():
                        self.result.delete(0, "end")
                        self.result.insert("end", "5")
                else:
                        self.result.insert("end", "5")

        def four(self):
                if "error" in self.result.get() or '=' in self.result.get():
                        self.result.delete(0, "end")
                        self.result.insert("end", "4")
                else:
                    self.result.insert("end", "4")

        def three(self):
                if "error" in self.result.get() or '=' in self.result.get():
                        self.result.delete(0, "end")
                        self.result.insert("end", "3")
                else:
                        self.result.insert("end", "3")

        def two(self):
                if "error" in self.result.get() or '=' in self.result.get():
                        self.result.delete(0, "end")
                        self.result.insert("end", "2")
                else:
                        self.result.insert("end", "2")

        def one(self):
```

```
                if "error" in self.result.get() or '=' in self.result.get():
                        self.result.delete(0, "end")
                        self.result.insert("end", "1")
                else:
                        self.result.insert("end", "1")

        def zero(self):
                if "error" in self.result.get() or '=' in self.result.get():
                        self.result.delete(0, "end")
                        self.result.insert("end", "0")
                else:
                        self.result.insert("end", "0")

        def plus(self):
                if "error" in self.result.get() or '=' in self.result.get():
                        self.result.delete(0, "end")
                        self.result.insert("end", "+")
                else:
                        self.result.insert("end", "+")

        def minus(self):
                if "error" in self.result.get() or '=' in self.result.get():
                        self.result.delete(0, "end")
                        self.result.insert("end", "-")
                else:
                        self.result.insert("end", "-")

        def mul(self):
                if "error" in self.result.get() or '=' in self.result.get():
                        self.result.delete(0, "end")
                        self.result.insert("end", "*")
                else:
                        self.result.insert("end", "*")

        def divide(self):
                if "error" in self.result.get() or '=' in self.result.get():
                        self.result.delete(0, "end")
                        self.result.insert("end", "/")
                else:
                        self.result.insert("end", "/")

        def equal(self):

                if self.result.get() == "":
                        self.result.insert("end","error")
                elif self.result.get()[0] == "0" or self.result.get()[0] ==
"+" or self.result.get()[0] == "*" or self.result.get()[0] == "/":
                        self.result.delete(0,"end")
```

```
                        self.result.insert("end","error")
                elif 'error' in self.result.get() or '=' in self.result.get():
                        self.result.delete(0,"end")

                else:
                        self.res = self.result.get()
                        self.res = eval(self.res)
                        self.result.insert("end"," = ")
                        self.result.insert("end",self.res)




        # ======== Clear Fields =============


        def clear(self):
                self.result.delete(0,"end")
        def Clear(self):
                self.Asus_Aspire_5_item.delete(0,"end")
                self.Asus_VivoBook_14_item.delete(0,"end")
                self.HP_Pavilion_x360_item.delete(0,"end")
                self.Lenovo_ThinkPadX1_Carbon_item.delete(0,"end")
                self.MSI_GS66_Stealth_item.delete(0,"end")
                self.Razer_Blade_15_item.delete(0,"end")
                self.MacBook_Air_M1_item.delete(0,"end")
                self.items_cost.delete(0,"end")
                self.antivirus_cost.delete(0,"end")
                self.total_bill.delete(0,"end")


        def Save_Bill(self):

            self.root.filename =
filedialog.asksaveasfile(mode="w",defaultextension='.txt')
            if self.root.filename is None:
                    return
            file_save =  str(self.bill_details.get(1.0,END))
            self.root.filename.write(file_save)
            self.root.filename.close()


        # ======= EXIT BUTTON CODE =========
        def Quit(self):
                self.message = messagebox.askquestion('Exit',"Do you want to
exit the application")
                if self.message == "yes":
```

```
                        self.root.destroy()
                else:
                        "return"



            #========= END ==============


        def __init__(self):
                self.root = tk.Tk()
                self.root.geometry("650x400")
                self.root.title("Laptop Management")
                self.root.minsize(650, 350)
                self.root.maxsize(650, 350)
                self.root['bg'] = "white"

                self.heading = Label(self.root,text="Laptop
Management",font=('verdana',20,'bold'),fg="#248aa2",bg="white")
                self.heading.place(x=170,y=5)

                self.style1 = Label(self.root,bg="#248aa2",height=1,width=25)
                self.style1.place(x=10,y=50)
                self.style2 = Label(self.root,bg="#248aa2",height=1,width=30)
                self.style2.place(x=480,y=50)
                self.date =
Label(self.root,text=datetime.now(),font=('verdana',10,'bold'),bg="white")
                self.date.place(x=220,y=50)


                # ================ ITEMS ===============

                self.frame1 = LabelFrame(self.root,text="Laptop
Management",width=250,height=200, font=('verdana',10,'bold'),
                                         borderwidth=3,relief=RIDGE,highlightt
hickness=4,bg="white",
                                         highlightcolor="white",highlightbackg
round="white",fg="#248aa2")
                self.frame1.place(x=30,y=90)

                self.Asus_Aspire_5 =
Label(self.frame1,text="Asus_Aspire_5",font=('verdana',10,'bold'),bg="white")
                self.Asus_Aspire_5.place(x=3,y=1)
                self.Asus_Aspire_5_item =
Entry(self.frame1,width=7,borderwidth=4, relief=SUNKEN,bg="#248aa2")
                self.Asus_Aspire_5_item.place(y=1,x=180)
```

```
                self.Asus_VivoBook_14 =
Label(self.frame1,text="Asus_VivoBook_14",font=('verdana',10,'bold'),bg="white
")
                self.Asus_VivoBook_14.place(x=3,y=20)
                self.Asus_VivoBook_14_item =
Entry(self.frame1,width=7,borderwidth=4, relief=SUNKEN,bg="#248aa2")
                self.Asus_VivoBook_14_item.place(y=20,x=180)

                self.HP_Pavilion_x360 =
Label(self.frame1,text="HP_Pavilion_x360",font=('verdana',10,'bold'),bg="white
")
                self.HP_Pavilion_x360.place(x=3,y=40)
                self.HP_Pavilion_x360_item =
Entry(self.frame1,width=7,borderwidth=4,relief=SUNKEN,bg="#248aa2")
                self.HP_Pavilion_x360_item.place(y=40,x=180)

                self.Lenovo_ThinkPadX1_Carbon =
Label(self.frame1,text="Lenovo_TPX1_Carbon",font=('verdana',10,'bold'),bg="whi
te")
                self.Lenovo_ThinkPadX1_Carbon.place(x=3,y=60)
                self.Lenovo_ThinkPadX1_Carbon_item =
Entry(self.frame1,width=7,borderwidth=4,relief=SUNKEN,bg="#248aa2")
                self.Lenovo_ThinkPadX1_Carbon_item.place(y=60,x=180)

                self.MSI_GS66_Stealth =
Label(self.frame1,text="MSI_GS66_Stealth",font=('verdana',10,'bold'),bg="white
")
                self.MSI_GS66_Stealth.place(x=3,y=80)
                self.MSI_GS66_Stealth_item =
Entry(self.frame1,width=7,borderwidth=4,relief=SUNKEN,bg="#248aa2")
                self.MSI_GS66_Stealth_item.place(y=80,x=180)

                self.Razer_Blade_15 =
Label(self.frame1,text="Razer_Blade_15",font=('verdana',10,'bold'),bg="white")
                self.Razer_Blade_15.place(x=3,y=100)
                self.Razer_Blade_15_item =
Entry(self.frame1,width=7,borderwidth=4,relief=SUNKEN,bg="#248aa2")
                self.Razer_Blade_15_item.place(y=100,x=180)

                self.MacBook_Air_M1 =
Label(self.frame1,text="MacBook_Air_M1",font=('verdana',10,'bold'),bg="white")
                self.MacBook_Air_M1.place(x=3,y=120)
                self.MacBook_Air_M1_item =
Entry(self.frame1,width=7,borderwidth=4,relief=SUNKEN,bg="#248aa2")
                self.MacBook_Air_M1_item.place(y=120,x=180)


                #============ ITEM BILLS ==============
```

```
                self.frame2 = LabelFrame(self.root,text="Laptop ITems
Bills",width=180,height=160,font= ('verdana',10,'bold'),
                                         borderwidth=3,relief=RIDGE,
highlightthickness=4, bg="white",highlightcolor="white",
                                         highlightbackground="white",fg="#248a
a2")
                self.frame2.place(x=290,y=90)


                self.items_cost_lb = Label(self.frame2, text="Items
Cost",font=('verdana',10, 'bold'),bg= "white")
                self.items_cost_lb.place(x=3,y=1)
                self.items_cost = Entry(self.frame2, width=9,
borderwidth=4,relief=SUNKEN,bg="#248aa2")
                self.items_cost.place(y=1,x=100)

                self.antivirus_cost_lb
=Label(self.frame2,text="Antivirus",font=('verdana',10,'bold'),bg="white")
                self.antivirus_cost_lb.place(x=3,y=20)
                self.antivirus_cost = Entry(self.frame2,width=9,
borderwidth=4, relief=SUNKEN,bg="#248aa2")
                self.antivirus_cost.place(y=20,x=100)

                self.total_bill_lb =  Label(self.frame2, text="Total Bill",
font=('verdana', 10, 'bold'),bg="white")
                self.total_bill_lb.place(x=3,y=100)
                self.total_bill = Entry(self.frame2,width=9, borderwidth=4,
relief=SUNKEN,bg="#248aa2")
                self.total_bill.place(y=100,x=100)


                # ============= CALCULATOR ============


                self.frame3 =
LabelFrame(self.root,text="Calculator",font=('verdana',10,'bold'),
fg="#248aa2",bg="white",
                                   highlightbackground="white",width=135,height=1
50,borderwidth=3,relief=RIDGE)
                self.frame3.place(x=480,y=94)

                self.result =
Entry(self.frame3,width=19,relief=SUNKEN,borderwidth=3)
                self.result.place(x=2,y=0)

                self.nine = Button(self.frame3,text="9", padx=6,
relief=RAISED, borderwidth=2, font=('verdana', 10, 'bold'),
```

```
                                    bg='#248aa2',fg="white",
command=self.nine)
                self.nine.place(x=0,y=24)
                self.eight = Button(self.frame3, text="8", padx=6,
relief=RAISED, borderwidth=2, font=('verdana', 10, 'bold'),
                                    bg='#248aa2',fg="white",
command=self.eight)
                self.eight.place(x=32,y=24)
                self.seven = Button(self.frame3, text="7", padx=6,
relief=RAISED, borderwidth=2, font=('verdana', 10, 'bold'),
                                    bg='#248aa2', fg="white",
command=self.seven)
                self.seven.place(x=64,y=24)
                self.plus = Button(self.frame3, text="+", padx=6,
relief=RAISED, borderwidth=2, font=('verdana', 10, 'bold'),
                                bg='white', fg="black", command=self.plus)
                self.plus.place(x=96,y=24)

                self.six = Button(self.frame3, text="6",padx=6, relief=RAISED,
borderwidth=2, font=('verdana', 10, 'bold'),
                                    bg='#248aa2',fg="white", command=self.six)
                self.six.place(x=0,y=50)
                self.five = Button(self.frame3, text="5",padx=6,
relief=RAISED, borderwidth=2, font=('verdana', 10, 'bold'),
                                    bg='#248aa2',fg="white",
command=self.five)
                self.five.place(x=32,y=50)
                self.four = Button(self.frame3, text="4",padx=6,
relief=RAISED, borderwidth=2, font=('verdana', 10, 'bold'),
                                    bg='#248002',fg="white",
command=self.four)
                self.four.place(x=64,y=50)
                self.minus = Button(self.frame3, text="-", padx=8,
relief=RAISED, borderwidth=2, font=('verdana', 10, 'bold'),
                                    bg='white',fg="black", command=self.minus)
                self.minus.place(x=96,y=50)

                self.three = Button(self.frame3, text="3",padx=6,
relief=RAISED, borderwidth=2, font=('verdana', 10, 'bold'),
                            bg='#248002',fg="white", command=self.three)
                self.three.place(x=0,y=76)
                self.two = Button(self.frame3, text="2",padx=6, relief=RAISED,
borderwidth=2, font=('verdana', 10, 'bold'),
                            bg='#248002',fg="white", command=self.two)
                self.two.place(x=32,y=76)
                self.one = Button(self.frame3, text="1",padx=6, relief=RAISED,
borderwidth=2, font=('verdana', 10, 'bold'),
                                    bg='#248002',fg="white", command=self.one)
```

```
                self.one.place(x=64,y=76)
                self.multiply = Button(self.frame3, text="*",padx=7,
relief=RAISED, borderwidth=2, font=('verdana', 10, 'bold'),
                                       bg='#248002',fg="black", command=self.mul)
                self.multiply.place(x=96,y=76)

                self.zero = Button(self.frame3, text="0",padx=6,
relief=RAISED, borderwidth=2, font=('verdana', 10, 'bold'),
                                       bg='#248002',fg="white",
command=self.zero)
                self.zero.place(x=0,y=102)
                self.clear = Button(self.frame3, text="C",padx=6,
relief=RAISED, borderwidth=2, font=('verdana', 10, 'bold'),
                                       bg='#248002',fg="white",
command=self.clear)
                self.clear.place(x=32,y=102)
                self.equal = Button(self.frame3, text="=",padx=6,
relief=RAISED, borderwidth=2, font=('verdana', 10, 'bold'),
                                       bg='#248002',fg="white",
command=self.equal)
                self.equal.place(x=64,y=102)
                self.divide = Button(self.frame3, text="/",padx=7,
relief=RAISED, borderwidth=2, font=('verdana', 10, 'bold'),
                                       bg='#248002',fg="black",
command=self.divide)
                self.divide.place(x=96,y=102)


                self.Total_Bills_btn =
Button(self.root,text="Total",relief=RAISED,borderwidth=2,font=('verdana',10,'
bold'),
                                                bg='#248aa2',fg="white",command=
self.Total_Bill)
                self.Total_Bills_btn.place(x=480,y=255)

                self.bill_details =
ScrolledText(self.frame3,width=23,height=9,relief=SUNKEN,borderwidth=3,font=('
courier',9,''))
                self.bill_details.place(x=0,y=130)

                self.Save_Bills_btn =
Button(self.root,text="Save",relief=RAISED,borderwidth=2,font=('verdana',10,'b
old'),bg='#248aa2',
                                               fg="white",command=self.Save_Bill
)
                self.Save_Bills_btn.place(x=540,y=255)

                self.root.mainloop()
```

```
if __name__ == "__main__":
            laptop_management()


# In[ ]:
```

Berikut adalah penjelasan mengenai bagian-bagian utama dari script tersebut:

- Mengimpor Modul:
  1. from tkinter import * dan import tkinter as tk mengimpor modul tkinter yang digunakan untuk membuat GUI.
  2. from datetime import datetime mengimpor modul datetime yang digunakan untuk mendapatkan waktu saat ini.
  3. from PIL import ImageTk, Image mengimpor modul PIL (Python Imaging Library) yang digunakan untuk menampilkan gambar.
  4. from tkinter import messagebox mengimpor modul messagebox yang digunakan untuk menampilkan pesan dialog.
  5. from tkinter import filedialog mengimpor modul filedialog yang digunakan untuk memilih file dari sistem.
  6. from tkinter.scrolledtext import ScrolledText mengimpor modul ScrolledText yang digunakan untuk membuat area teks yang dapat di-scroll.

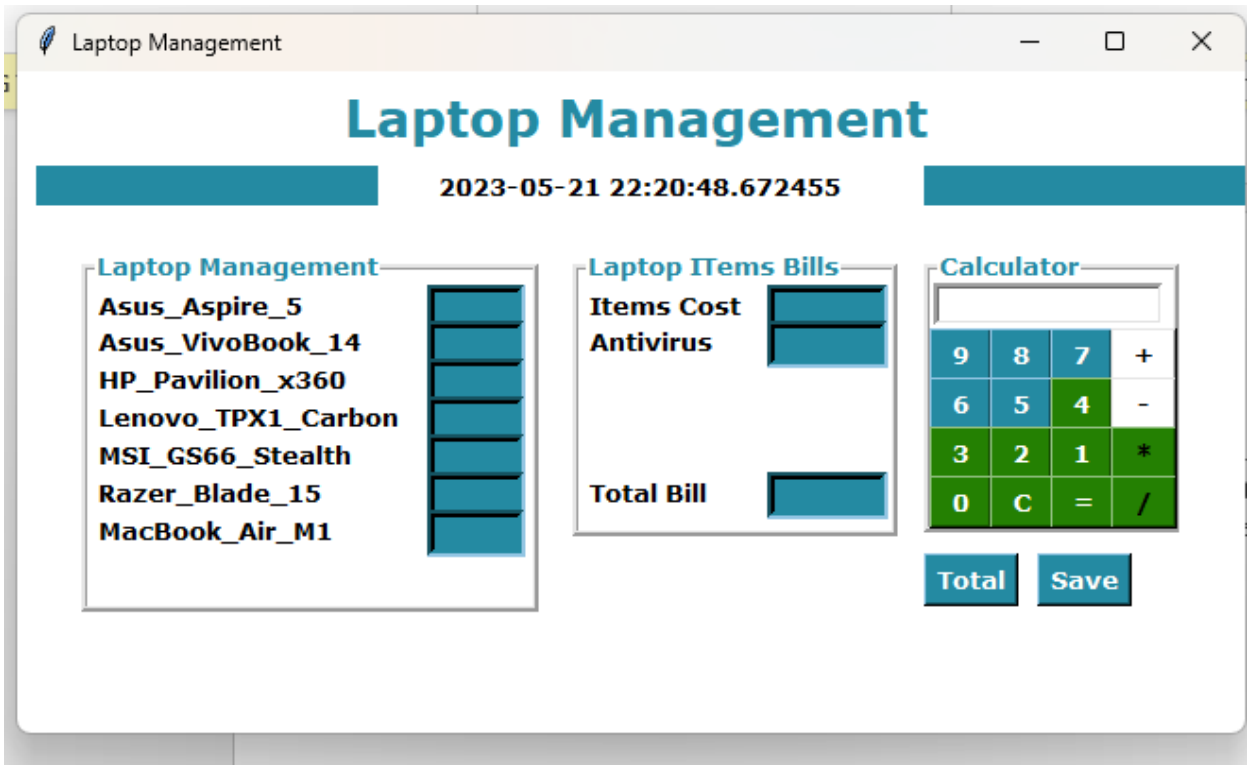
- Pembuatan Class:
  1. class ManajemenLaptop(): mendefinisikan sebuah class bernama ManajemenLaptop.

- Fungsi TotalTagihan():
  1. Fungsi ini digunakan untuk menghitung total tagihan berdasarkan harga dan jumlah item laptop yang dipilih.
  2. Setiap item laptop memiliki harga yang ditentukan.
  3. Fungsi ini akan membaca jumlah item yang diinputkan oleh pengguna dan mengalikan dengan harga untuk mendapatkan biaya tiap item.
  4. Jika pengguna tidak memasukkan jumlah item, biaya akan dianggap nol.
  5. Biaya tiap item akan dijumlahkan untuk mendapatkan total tagihan.

- Inisialisasi Harga:
  1. Terdapat beberapa variabel yang menyimpan harga dari masing-masing laptop, misalnya harga_Asus_Aspire_5, harga_Asus_VivoBook_14, dll.
  2. Harga-harga ini bisa diubah sesuai dengan kebutuhan.

- Perhitungan Biaya:
    1. Setiap variabel biaya (misalnya biaya_Asus_Aspire_5, biaya_Asus_VivoBook_14, dll.) diinisialisasi dengan nilai 0.
    2. Jika pengguna memasukkan jumlah item untuk laptop tertentu, biaya untuk laptop tersebut akan dihitung dengan mengalikan jumlah item dengan harga laptop yang bersangkutan**.**

Hasil



BAB IV
PENUTUP

Kesimpulan

Tkinter adalah modul bawaan Python yang digunakan untuk membuat antarmuka grafis pengguna (GUI). Dengan Tkinter, pengembang dapat dengan mudah membuat aplikasi desktop yang interaktif dan intuitif. Berikut adalah kesimpulan singkat tentang pengoperasian Tkinter Python:

- Tkinter memungkinkan pembuatan antarmuka pengguna yang mudah dipelajari dan digunakan.
- Antarmuka Tkinter dapat berjalan di berbagai platform, seperti Windows, macOS, dan Linux.
- Tkinter terintegrasi dengan baik dengan bahasa pemrograman Python, memungkinkan akses ke berbagai pustaka dan fitur Python.
- Tkinter menyediakan berbagai widget dan tata letak untuk membangun antarmuka GUI yang kompleks.

Saran

Cobalah untuk membuat aplikasi sederhana dengan beberapa widget dasar seperti tombol, label, atau kotak teks. Ini akan membantu untuk memahami dasar-dasar Tkinter dan membangun kepercayaan diri sebelum beralih ke proyek yang lebih kompleks.Tkinter menyediakan beberapa metode tata letak seperti grid, pack, dan place. Pelajari kelebihan dan kekurangan masing-masing metode tata letak dan pilih yang paling sesuai dengan kebutuhan. Cobalah untuk mengatur widget secara terorganisir dalam jendela aplikasi. Tkinter menyediakan pustaka tambahan bernama ttk (Themed Tkinter) yang menyediakan tampilan yang lebih modern dan kustomisasi yang lebih luas. Pengguna dapat menggunakan ttk untuk meningkatkan penampilan antarmuka pengguna dengan menggunakan tema dan gaya yang berbeda.