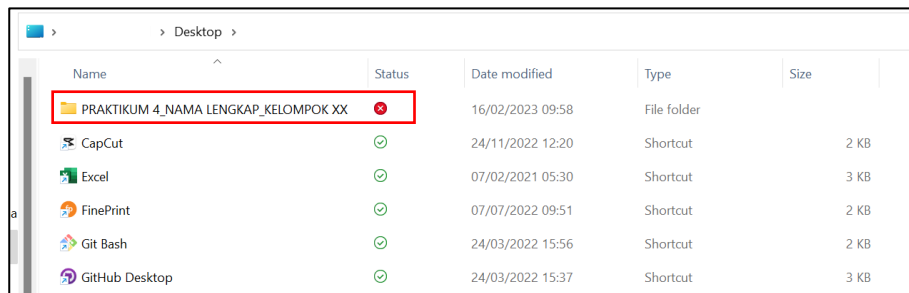


PRAKTIKUM 4 MODULE DAN GRAPHICAL USER INTERFACE (GUI)

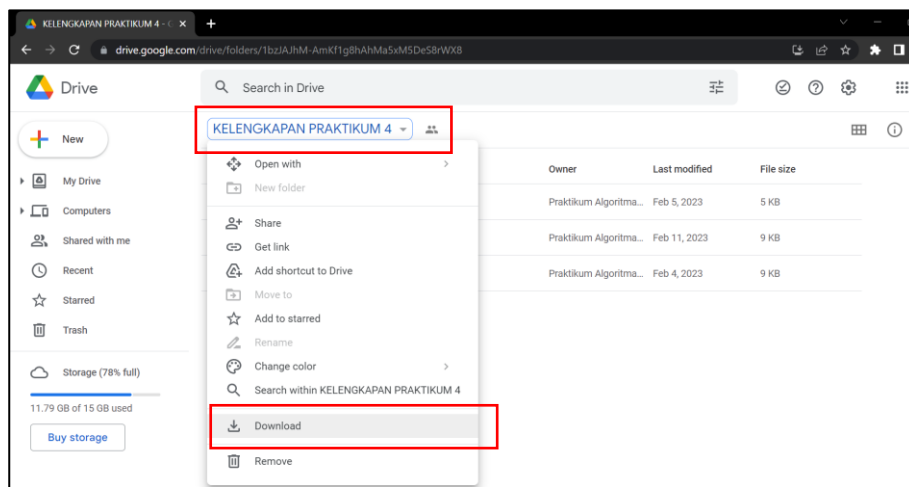
Pada praktikum ini akan dijelaskan mengenai pembuatan program berbasis *Graphical User Interface* yang dilengkapi module dengan penerapannya pada studi kasus praktikum sistem perhitungan pembelian.

1. Membuat folder pada desktop dengan nama **“PRAKTIKUM 4_NAMA LENGKAP_KELOMPOK XX”**



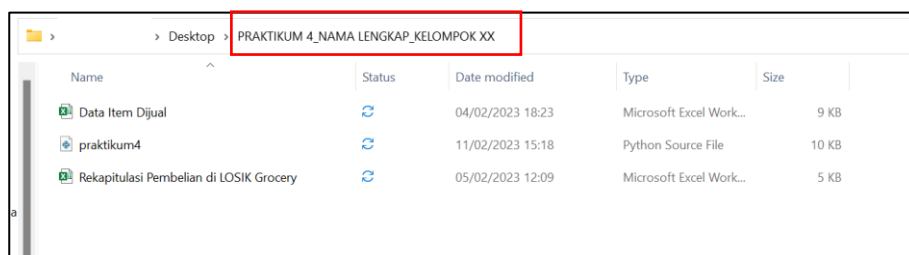
Gambar 4. 1 Membuat Folder Praktikum 4

2. Mendownload file pada <https://bit.ly/KelengkapanPraktikum4>



Gambar 4. 2 Mendownload File Kelengkapan Praktikum 4

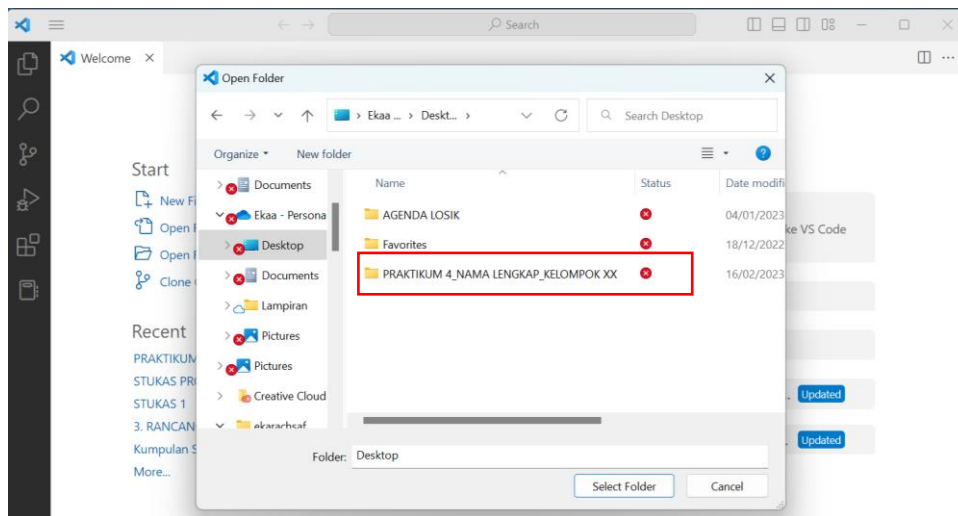
3. Men-copy paste file kelengkapan praktikum 4 pada folder yang telah dibuat tadi (**PRAKTIKUM 4_NAMA LENGKAP_KELOMPOK XX**)



Gambar 4. 3 Men-copy paste File Kelengkapan Praktikum 4

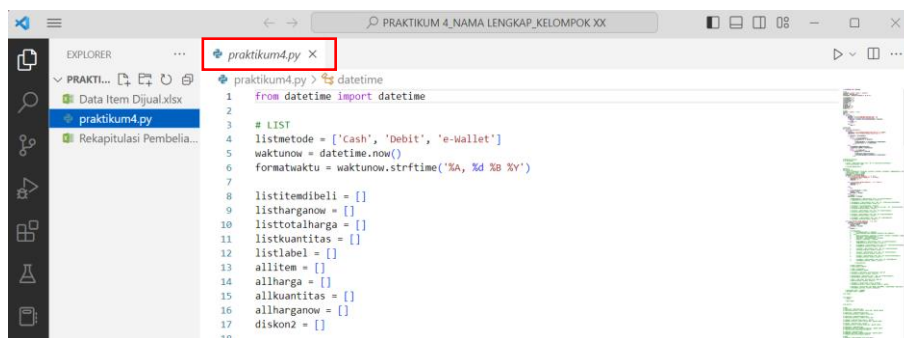
GUIDEBOOK

4. Membuka *visual studio code* dan *open folder* yang telah dibuat tadi.



Gambar 4. 4 Membuka Folder Praktikum 4 di Visual Studio Code

5. Membuka file **praktikum4.py**



Gambar 4. 5 Membuka File praktikum4.py

6. Memilih menu **Terminal -> New Terminal** dan ketik **pip install openpyxl** pada terminal baru. **Openpyxl** merupakan *library python* yang digunakan untuk melakukan baca/tulis file excel baik format xlsx atau lainnya.



Gambar 4. 6 Install Library Openpyxl

7. Menambahkan kode import tkinter seperti gambar di bawah.

```
from tkinter import *  
from tkinter import ttk  
from tkinter.messagebox import *
```

GUIDEBOOK

8. Menuliskan list kosong dan list jenis item dan harganya.

```
# LIST
```

```
listmetode = ['Cash', 'Debit', 'e-Wallet']  
itemdijual = ['sabun cair', 'sabun cuci', 'sampo keratin', 'sikat gigi',  
             'minuman Dingin', 'makanan ringan', 'mi instan', 'air mineral']  
listharga = [200000, 120000, 190000, 180000, 210000, 170000, 110000,  
             220000]  
waktunow = datetime.now()  
formatwaktu = waktunow.strftime('%A, %d %B %Y')
```

```
listitemdibeli = []  
listharganow = []  
listtotalharga = []  
listkuantitas = []  
listlabel = []
```

```
allitem = []  
allharga = []  
allkuantitas = []  
allharganow = []  
diskon2 = []
```

9. Membuat frame dan widget main window. Main window dibagi menjadi 1 frame utama yang dibagi menjadi 6 subframe. 6 subframe ini mewakili setiap def *function* yang telah dibuat sebelumnya. Kemudian, pada frame rincian dibagi lagi menjadi rincian 1 (labelframe keranjang untuk menampilkan rincian pembelian) dan rincian 2 (labelframe untuk menampilkan total pembelian dari item yang telah diinputkan).

```
# WINDOW
```

```
main_window = Tk()  
main_window.title('Kasir LOSIK GROCERY')  
main_window.withdraw()
```

```
# FRAME
```

```
frame = Frame(main_window)  
frame.pack()
```

GUIDEBOOK

```
frameitem = LabelFrame(frame, padx=10)
frameitem.grid(row=0, column=0, padx=20, pady=5)
```

```
framerincian = LabelFrame(frame)
framerincian.grid(row=1, column=0, sticky='news', padx=20, pady=5)
```

```
framerincian1 = LabelFrame(framerincian)
framerincian1.grid(row=0, column=0, sticky='news')
```

```
framerincian2 = LabelFrame(framerincian)
framerincian2.grid(row=1, column=0, sticky='news')
```

```
framebayar = LabelFrame(frame, pady= 5, padx=10)
framebayar.grid(row=2, column=0 ,sticky='news', padx=20, pady=5)
```

```
framenota = LabelFrame(frame, padx=10)
framenota.grid(row=3, column=0,sticky='news', padx=20, pady=5)
```

```
frameselesai = LabelFrame(frame)
frameselesai.grid(row=4, column=0,sticky='news', padx=20, pady=5)
frameselesai.grid_columnconfigure(1,weight=1)
```

```
framegrafik = LabelFrame(frame)
framegrafik.grid(row=5, column=0,sticky='news', padx=20, pady=5)
framegrafik.grid_columnconfigure(1,weight=1)
```

WIDGET

```
jenisitem = Label(frameitem, text='Jenis Item')
jenisitem.grid(row=0, column=0, sticky="w")
```

```
inputitem = ttk.Combobox(frameitem, values= itemdijual)
inputitem.current(0)
inputitem.grid(row=0, column=1, sticky="w", pady=10, padx=10)
```

```
jumlahitem = Label(frameitem, text='Jumlah Item')
jumlahitem.grid(row=1, column=0, sticky="w")
```

```
inputjumlah = Spinbox(frameitem, from_=1, to=100)
```

GUIDEBOOK

```
inputjumlah.grid(row=1, column=1, sticky="w", pady=10, padx=10)
```

```
buttoninput = Button(frameitem, text='Input', command= lambda:
[rekap(), tambah(), keranjang(),], bg='#7ED957', padx=20, pady=2.5, )
buttoninput.grid(row=0, column=2, rowspan=2, padx=25)
```

```
no = Label(framerincian1, text='No')
no.grid(row=0, column=0)
```

```
item = Label(framerincian1, text='Jenis Item')
item.grid(row=0, column=1)
```

```
quantity = Label(framerincian1, text='Quantity')
quantity.grid(row=0, column=2)
```

```
harga = Label(framerincian1, text='Harga')
harga.grid(row=0, column=3)
```

```
Total = Label(framerincian1, text='Total')
Total.grid(row=0, column=4)
```

```
totall = Label(framerincian2, text='Total Belanja')
totall.grid(column=0, row=0)
```

```
for i in framerincian1.winfo_children():
    i.grid_configure(padx= 17.5, pady=2)
```

```
metodebayar = Label(framebayar, text='Metode Pembayaran')
metodebayar.grid(row=0, column=0, sticky="w")
```

```
inputmetodebayar = ttk.Combobox(framebayar, values= listmetode)
inputmetodebayar.current(0)
inputmetodebayar.grid(row=0, column=1, sticky="w", pady=10, padx=10)
```

```
buttonpay = Button(framebayar, text= 'Bayar', padx=10, bg='#7ED957',
command= lambda: [pembayaran(),])
buttonpay.grid(row=1, column=0, columnspan= 2)
```


GUIDEBOOK

```
tanggal = Label(framenota, text='Tanggal Pembelian')
tanggal.grid(row=0, column=0, sticky="w")

rincian1 = Label(framenota, text='Rincian: ')
rincian1.grid(row=1, column=0, sticky="w")

pembelian = Label(framenota, text='1. Total Pembelian')
pembelian.grid(row=2, column=0, sticky="w")

diskon = Label(framenota, text='2. Total Diskon')
diskon.grid(row=3, column=0, sticky="w")

totalakhir = Label(framenota, text='3. Total Harga')
totalakhir.grid(row=4, column=0, sticky="w")

bayar = Label(framenota, text='Total Pembayaran')
bayar.grid(row=5, column=0, sticky="w")

kembali = Label(framenota, text='Total Kembalian')
kembali.grid(row=6, column=0, sticky="w")

buttonselesai = Button(frameselesai, text= 'SELESAI', padx=10,
bg='#6297FF',command=selesai )
buttonselesai.grid(row=0,column=0, sticky='news', columnspan=4)
buttonselesai['state'] = DISABLED

buttongrafik = Button(framegrafik, text= 'Tampilkan Grafik', padx=10,
bg='#6297FF',command=grafik )
buttongrafik.grid(row=0,column=0, sticky='news', columnspan=4)
buttongrafik['state'] = DISABLED
```

10. Mengubah setiap input("...") menjadi input.get(). Untuk mengambil value dalam input.get(), dibuat widget Entry (untuk memasukkan teks), DateEntry (untuk memasukkan tanggal), Combobox (untuk memasukkan *dropdown-list*), Spinbox (untuk memasukkan range angka dari sebuah list), dan Scale (untuk memasukkan value dengan menggunakan *slider*). Lalu, untuk menambah item, agar langsung memasukkan item selanjutnya tanpa menghapus value pada item sebelumnya. Maka ditambahkan input.delete(0,END). Dan

GUIDEBOOK

apabila item selanjutnya yang dimasukkan sudah ada di list, maka yang bertambah adalah jumlah itemnya saja (kuantitas).

#Input Item

```
def tambah():
    if not inputitem.get():
        showerror(title='Error', message='Silahkan Masukkan Jenis Item', )
    elif not inputjumlah.get():
        showerror(title='Error', message='Silahkan Masukkan Jumlah Item
yang akan dibeli', )
    else:
        if inputitem.get() in listitemdibeli:
            i = 0
            while i < len(listitemdibeli):
                if listitemdibeli[i] == inputitem.get():
                    x = i
                    listtotalharga[x] +=
(listharga[x]*int(inputjumlah.get()))

                    listkuantitas[x] += int(inputjumlah.get())
                    i += 1
        else:
            inputjumlah.get()
            listitemdibeli.append(inputitem.get())
            listkuantitas.append(int(inputjumlah.get()))
            i = 0
            while i < len(itemdijual):
                if itemdijual[i] == inputitem.get():
                    x = i
                    listharganow.append(listharga[x])
                    listtotalharga.append(listharga[x]*int(inputjumlah.get(
)))

                    i +=1
            inputitem.delete(0,END)
            inputjumlah.delete(0,END)
```

11. Membuat keranjang untuk menampilkan rincian pembelian berupa no, jenis item, kuantitas, harga satuan, dan harga total. Pada pembuatan label frame keranjang dibuat

GUIDEBOOK

dengan struktur perulangan *while*. Setelah, membuat keranjang, dibuat kode untuk menghitung total pembelian (sebelum mendapatkan potongan harga) dengan cara menambahkan fungsi `sum()`.

Menampilkan Keranjang

```
def keranjang():  
    global no1, item, quantity, harga, Total  
    i = 0  
    while i < len(listitemdibeli):  
  
        no1 = Label(framerincian1, text= 1 + i)  
        no1.grid(row=i+1, column=0,)  
  
        item = Label(framerincian1, text= listitemdibeli[i])  
        item.grid(row=i+1, column=1, )  
  
        quantity = Label(framerincian1, text= listkuantitas[i])  
        quantity.grid(row=i+1, column=2, )  
  
        harga = Label(framerincian1, text=  
            'Rp{:, .2f}'.format(listharganow[i]))  
        harga.grid(row=i+1, column=3, )  
  
        Total = Label(framerincian1,  
            text='Rp{:, .2f}'.format(listtotalharga[i]))  
        Total.grid(row=i+1, column=4, )  
  
        listlabel.append(no1)  
        listlabel.append(item)  
        listlabel.append(quantity)  
        listlabel.append(harga)  
        listlabel.append(Total)  
  
        i += 1  
  
    totalll = Label(framerincian2, text=': Rp.  
{:}'.format(sum(listtotalharga)))  
    totalll.grid(column=1, row=0)
```



```
listlabel.append(total111)
```

12. Pada metode pembayaran, semua input(“...”) juga diganti dengan input.get(). Kemudian untuk menampilkan rincian pembayaran, yang awalnya menggunakan *function* print() diganti menggunakan label. Sedangkan, untuk pembayaran *cash* dibuat window baru yang dimasukkan ke dalam def bayarcash() untuk memasukkan jumlah nominal yang dibayarkan pelanggan.

```
#Pembayaran
```

```
def pembayaran():
    global nilaipembelian, nilaidiskon, nilaitotal, nilaibayar,
    nilaikembali, tanggalpembelian, totaldiskon
    buttonselesai['state'] = ACTIVE
    buttongrafik['state'] = ACTIVE
    totalbayar = sum(listtotalharga)
    if not inputmetodebayar.get():
        showerror(title='Error', message='Silahkan Pilih Metode
        Bayar', )
    else:
        if inputmetodebayar.get() != 'Cash':
            if inputmetodebayar.get() == 'Debit':
                totalbayar *= 0.95
                kembalian = 0

            elif inputmetodebayar.get() == 'e-Wallet':
                totalbayar *= 0.93
                kembalian = 0
            else:
                pass

        if sum(listtotalharga) >= 500000:
            diskon2 = 20000
            totalbayar -= diskon2
        else:
            diskon2 = 0
    nominalbayar = totalbayar
```

GUIDEBOOK

```
tanggalpembelian = Label(framenota, text=':
{}'.format(formatwaktu))
tanggalpembelian.grid(row=0, column=1, sticky="w")

nilaipembelian = Label(framenota, text=': Rp{:, .2f}
'.format(sum(listttotalharga)))
nilaipembelian.grid(row=2, column=1, sticky= "w")

totaldiskon = sum(listttotalharga) - totalbayar
nilaidiskon = Label(framenota, text=': Rp{:, .2f} ({%{}
+Rp{:, .2f} '.format(totaldiskon, inputmetodebayar.get(),
diskon2))
nilaidiskon.grid(row=3, column=1, sticky='w')

nilaitotal = Label(framenota, text=':
Rp{:, .2f}'.format(totalbayar))
nilaitotal.grid(row=4, column=1, sticky='w')

nilaibayar = Label(framenota, text=':
Rp{:, .2f}'.format(nominalbayar))
nilaibayar.grid(row=5, column=1, sticky='w')

nilaikembali = Label(framenota, text=':
Rp{:, .2f}'.format(kembalian))
nilaikembali.grid(row=6, column=1, sticky='w')

elif inputmetodebayar.get() == 'Cash':
    totalbayar = (sum(listttotalharga))
    if sum(listttotalharga) >= 500000:
        diskon2 = 20000
        totalbayar -= diskon2
    else:
        diskon2 = 0

def bayarcash():
    if int(inputbayar2.get()) < totalbayar:
        showinfo(message='Jumlah Pembayaran kurang dari Total
Pembelian')
```

GUIDEBOOK

else:

```
global nilaipembelian, nilaidiskon, nilaitotal,
nilaibayar, nilaikembali, tanggalpembelian
nominalbayar = int(inputbayar2.get())
totaldiskon = sum(listtotalharga) - totalbayar
kembalian = nominalbayar - totalbayar

tanggalpembelian = Label(framenota, text=':
{}'.format(formatwaktu))
tanggalpembelian.grid(row=0, column=1, sticky="w")

nilaipembelian = Label(framenota, text=': Rp{:, .2f}
'.format(sum(listtotalharga)))
nilaipembelian.grid(row=2, column=1, sticky="w")

nilaidiskon = Label(framenota, text=':
Rp{:, .2f}'.format(totaldiskon))
nilaidiskon.grid(row=3, column=1, sticky='w')

nilaitotal = Label(framenota, text=':
Rp{:, .2f}'.format(totalbayar))
nilaitotal.grid(row=4, column=1, sticky='w')

nilaibayar = Label(framenota, text=':
Rp{:, .2f}'.format(int(nominalbayar)))
nilaibayar.grid(row=5, column=1, sticky='w')

nilaikembali = Label(framenota, text=':
Rp{:, .2f}'.format(kembalian))
nilaikembali.grid(row=6, column=1, sticky='w')

window.destroy()
```

```
window = Toplevel()
window.geometry('250x250')
```

```
frame1 = Frame(window)
frame1.pack(padx=25, pady=25)
```

GUIDEBOOK

```
totalakhir2 = Label(frame1, text='Total Harga', padx= 10)
totalakhir2.grid(row=0, column=0, sticky="w")

showtotalakhir2 = Label(frame1, text= ':
Rp{:, .2f}'.format(totalbayar))
showtotalakhir2.grid(row=0, column=1, sticky="w")

bayar2 = Label(frame1, text='Bayar Cash', padx= 10)
bayar2.grid(row=1, column=0, sticky="w")

inputbayar2 = Spinbox(frame1, from_=0, to=10**10)
inputbayar2.grid(row=1, column=1, pady=10, sticky="w", padx=10)

buttonbayar2 = Button(frame1, text='BAYAR', bg='#7ED957',
command=lambda: [bayarcash()])
buttonbayar2.grid(row=2, column=0, columnspan=2)

buttoninput['state'] = DISABLED
buttonpay['state'] = DISABLED
```

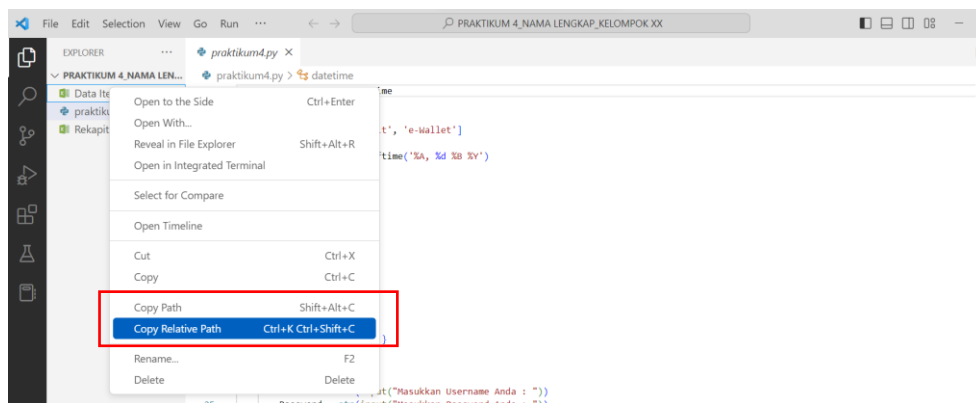
13. Membuat def rekap() yang berisi sama dengan def tambah() yang fungsinya untuk menginput setiap item yang dibeli. Hasil rekapitulasi ini nantinya akan disimpan di dalam file excel yang berisikan rekapitulasi semua item yang pernah dibeli di LOSIK Grocery.

```
def rekap():
    if not inputitem.get():
        pass
    elif not inputjumlah.get():
        pass
    else:
        if inputitem.get() in allitem:
            i = 0
            while i < len(allitem):
                if allitem[i] == inputitem.get():
                    x = i
```

GUIDEBOOK

```
allharga[x] +=  
(listharga[x]*int(inputjumlah.get()))  
  
allkuantitas[x] += int(inputjumlah.get())  
i += 1  
else:  
    inputjumlah.get()  
    allitem.append(inputitem.get())  
    allkuantitas.append(int(inputjumlah.get()))  
    i = 0  
    while i < len(itemdijual):  
        if itemdijual[i] == inputitem.get():  
            x = i  
            allharganow.append(listharga[x])  
            allharga.append(listharga[x]*int(inputjumlah.get()))  
    ))  
    i +=1
```

14. Dikarenakan data item dapat disimpan di file eksternal berbentuk *excel*. Maka dari itu, data item yang awalnya dijadikan sebuah list dijadikan dataframe dengan *library* *pandas*. Mengetikkan fungsi di bawah ini untuk membaca file pada file excel “**Data Item Dijual**”. Kemudian, *copy Path File Data Item* dan *paste* di “**pd.read_excel(“...xlsx”)**”. Kemudian pada item dijual dan list harga diberi akhiran fungsi *pandas* *series.tolist()* untuk mengkonversi *series* dataframe yang ada di *excel* menjadi sebuah list.



Gambar 4. 7 Copy Path Data Item

```
dataitem = pd.read_excel("Data Item Dijual.xlsx")  
itemdijual = dataitem['Jenis Item'].tolist()
```

```
listharga = dataitem['Harga Satuan'].tolist()
print(dataitem)
```

15. Membuat def selesai() yang mana ketika pembelian telah selesai dilakukan dan *button* selesai ditekan. Rincian pembelian akan disimpan dalam file *excel* menggunakan *library* pandas. Pandas mendukung pembacaan dan penulisan data dengan media berupa *spreadsheet*, *excel*, *CSV*, dan masih banyak lagi. Sebelum menuliskan kode dilakukan import pandas, yakni **import pandas as pd**. Lalu, **df.to_excel** digunakan untuk menyimpan data dalam file excel yang sudah dibuat di dalam folder.

```
def selesai():
    # Pandas
    #Declare Data
    data = {'Jenis Item':listitemdibeli,
            'Harga Satuan': listharganow,
            'Kuantitas': listkuantitas,
            'Harga Total':listtotalharga}

    # #Membuat Dataframe dari data diatas
    df = pd.DataFrame(data)

    #append data ke Excel
    df.to_excel("Rekapitulasi Pembelian di LOSIK Grocery.xlsx",
index=False, header=False)

    # Reset Window
    for i in listlabel:
        i.destroy()

    listitemdibeli.clear()
    listtotalharga.clear()
    listharganow.clear()
    listkuantitas.clear()

    nilaipembelian.destroy()
    nilaidiskon.destroy()
    nilaitotal.destroy()
    nilaibayar.destroy()
    nilaikembali.destroy()
    tanggalpembelian.destroy()
```



```
inputmetodebayar.delete(0,END)
```

```
buttoninput['state'] = ACTIVE
```

```
buttonpay['state'] = ACTIVE
```

```
buttonselesai['state'] = DISABLED
```

```
inputitem.current(0)
```

```
inputmetodebayar.current(0)
```

16. Membuat def grafik() yang mana ketika pembelian telah selesai dilakukan dan *button* “Tampilkan Grafik” ditekan maka akan menampilkan *piechart* yang berisikan proporsi penjualan berdasarkan jenis item yang dibeli. *Piechart* dibuat menggunakan Module Matplotlib yang mana *library* ini focus dalam visualisasi data yang disajikan dalam plot grafik (*piechart*, *linechart*, *barchart*, *etc*). Sebelum menuliskan kode, dilakukan import matplotlib yaitu, **import matplotlib.pyplot as plt**

```
def grafik():
```

```
    plt.pie(allharga,
```

```
           labels=allitem,
```

```
           autopct='%1.1f%%')
```

```
    plt.title('Grafik Proporsi Penjualan Jenis Item')
```

```
    plt.xlabel('Total Penjualan : {}'.format(int(sum(allharga) -  
totaldiskon)))
```

```
    plt.legend()
```

```
    plt.show()
```

17. Setelah semua program perhitungan pembelian item telah sesuai dan menghasilkan *output* yang diinginkan. Selanjutnya, membuat widget window login. Window login dibuat sebagai **Toplevel()** Window yang mana **Toplevel()** sendiri merupakan widget khusus tanpa adanya induk. Widget window berisikan frame login, label *username* dan *password*, *Entry username* dan *password* untuk memasukkan value, *button login* untuk menyimpan *value* yang dimasukkan dan memvalidasi apakah sudah valid atau belum.

```
#Login
```

```
window_login= Toplevel()
```

```
window_login.title('Login LOSIK')
```

```
window_login.geometry("250x250")
```

GUIDEBOOK

```
framelogin = Frame(window_login)
framelogin.pack(padx=10, fill='x')

dashboardlogin = Label(
    framelogin,
    text="SILAHKAN LOGIN",
    font=('Algerian', 15, 'bold')
)
dashboardlogin.grid(row=0, column=0, columnspan=2)

username = Label(framelogin, text='Username')
username.grid(row=1, column=0, sticky="w")

inputusername = Entry(framelogin)
inputusername.grid(row=1, column=1, padx= 20, pady=10)

password = Label(framelogin, text='Password')
password.grid(row=2,column=0, sticky="w")

inputpassword = Entry(framelogin, show='*')
inputpassword.grid(row=2,column=1)

buttonlogin= Button(framelogin, text='Login', command= lambda:
login())
buttonlogin.grid(row=3,column=1,columnspan=2,
sticky='we',padx=10,pady=10)
```

18. Dalam proses login, ketika login berhasil break diganti menjadi `window_login.destroy()` agar `window_login` hilang dan `main_window` muncul. Sedangkan, ketika `username` dan `password` salah akan menampilkan message box dan ketika sudah salah sebanyak 3x maka `main_window.destroy()` dan `window_login.destroy()` atau dengan kata lain program akan `quit()`,

```
# Login
userpass = {'admin' : '123'}
coba = 0
def login():
```

GUIDEBOOK

```
global coba
username1 = inputusername.get()
password1 = inputpassword.get()
if username1 in userpass and password1 == userpass[username1]:
    window_login.destroy()
    main_window.deiconify()
elif coba == 3:
    showwarning(message='Terminated')
    main_window.destroy()
    window_login.destroy()
else:
    showerror(message='Username / Password salah')
    inputusername.delete(0,END)
    inputpassword.delete(0,END)
coba += 1
```

19. Berikut merupakan *output* program sistem perhitungan pembelian item pada LOSIK Grocery yang ditampilkan dalam bentuk *Graphical User Interface* (GUI).

Kasir LOSIK GROCERY

Jenis Item Input

Jumlah Item

No	Jenis Item	Quantity	Harga	Total
1	Sabun Cuci	2	Rp120,000.00	Rp240,000.00
2	Sikat Gigi	3	Rp180,000.00	Rp540,000.00
3	Mi Instan	2	Rp110,000.00	Rp220,000.00
4	Air Mineral	5	Rp220,000.00	Rp1,100,000.00

Total Belanja : Rp2,100,000.00

Metode Pembayaran Bayar

Tanggal Pembelian : Thursday, 16 February 2023

Rincian:

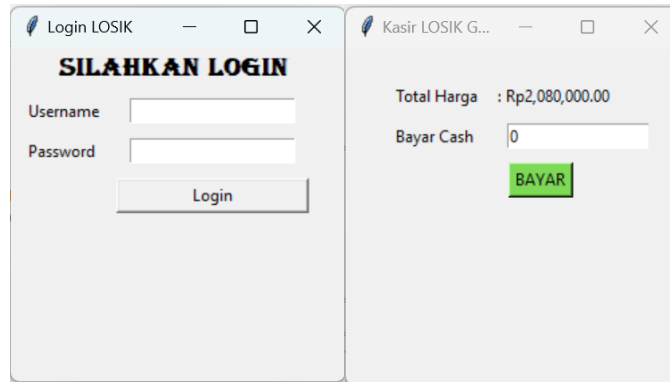
- 1. Total Pembelian : Rp2,100,000.00
- 2. Total Diskon : Rp125,000.00 (%Debit + Rp.20,000.00)
- 3. Total Harga : Rp1,975,000.00
- Total Pembayaran : Rp1,975,000.00
- Total Kembalian : Rp0.00

SELESAI

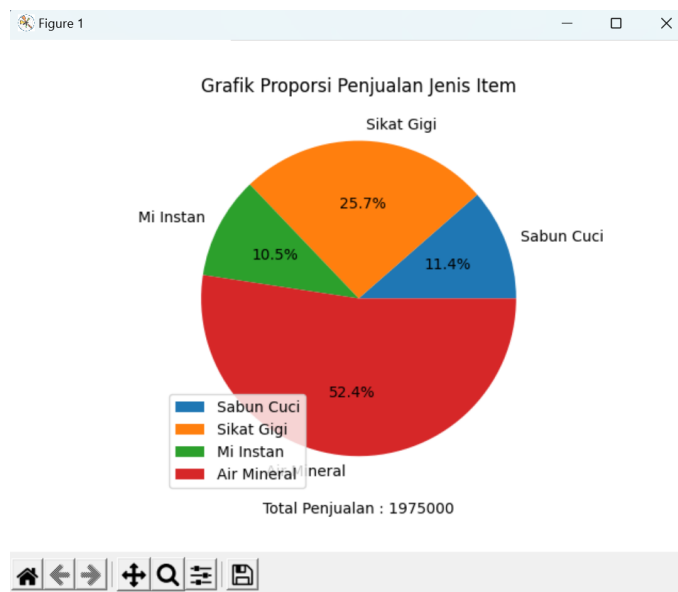
Tampilkan Grafik

Gambar 4. 8 Output Main Window Sistem perhitungan Pembelian Item di LOSIK Grocery

GUIDEBOOK



Gambar 4. 9 Output Toplevel Window Proses Login dan Pembayaran Cash



Gambar 4. 10 Output Penggunaan Matplotlib untuk Grafik Proporsi Penjualan Jenis Item

Rekapitulasi Pembelian di LOSIK Grocery - Excel

	A	B	C	D	E	F
1	Sabun Cuci	120000	2	240000		
2	Sikat Gigi	180000	3	540000		
3	Mi Instan	110000	2	220000		
4	Air Mineral	220000	5	1100000		

Data Item Dijual - Excel

	A	B	C	D
1	Jenis Item	Harga Satuan		
2	Sabun Cair	200000		
3	Sabun Cuci	120000		
4	Sampo Keratin	190000		
5	Sikat Gigi	180000		
6	Minuman Dingin	210000		
7	Makanan Ringan	170000		
8	Mi Instan	110000		
9	Air Mineral	220000		

Gambar 4. 11 Output Penggunaan Pandas untuk Menampilkan File Excel Rekapitulasi Pembelian dan Data Item Dijual