

# PRAKTIKUM ALGORITMA DAN PEMOGRAMAN

## MODUL 12 (TUGAS AKHIR)

NAMA : IBRAHIMOVIC ILHAM PRASTAMA

NIM : 1202307011

---

### **Aplikasi E-Commerce dengan Nama Toko (Ovic E-Store)**

#### **Deskripsi Singkat**

Kode yang saya gunakan tersebut merupakan implementasi sederhana dari aplikasi e-commerce menggunakan modul tkinter di Python. Aplikasi ini memungkinkan pengguna untuk menjelajahi produk, menambahkannya ke dalam keranjang belanja, melakukan checkout, dan memproses pembayaran dengan beberapa opsi metode pembayaran.

#### **Komponen Utama**

- **Inisialisasi Data Produk**  
Menyimpan daftar produk dan harganya dalam bentuk kamus.
- **Tampilan Produk Berdasarkan Kategori**  
Memungkinkan pengguna untuk memilih kategori produk dan menampilkan produk sesuai dengan kategori yang dipilih.
- **Keranjang Belanja**  
Menampilkan daftar produk yang ditambahkan oleh pengguna beserta jumlahnya dan total harga.
- **Pemilihan Lokasi Pengiriman**  
Mengizinkan pengguna untuk memilih lokasi pengiriman untuk perhitungan biaya pengiriman.
- **Proses Pembayaran**  
Memproses pembayaran dengan berbagai metode seperti Dana, OVO, Gopay, dsb.
- **Pengelolaan Keranjang**  
Menambah, menghapus, dan memperbarui daftar belanjaan.
- **Generasi Nota Pembelian**  
Membuat dan menampilkan ringkasan pembelian dalam bentuk nota.
- **Antarmuka Pengguna (UI)**  
Menyediakan antarmuka pengguna sederhana menggunakan tkinter untuk interaksi yang mudah bagi pengguna.

## Fitur Utama

- **Tampilan Produk Berdasarkan Kategori**  
Pengguna dapat menelusuri dan memilih produk berdasarkan kategori.
- **Tambah ke Keranjang**  
Pengguna dapat menambah produk ke dalam keranjang belanja.
- **Pilihan Lokasi Pengiriman**  
Pengguna dapat memilih lokasi pengiriman untuk estimasi biaya pengiriman.
- **Pilihan Metode Pembayaran**  
Pembayaran dapat dilakukan menggunakan berbagai opsi metode pembayaran yang telah ditentukan.
- **Checkout dan Nota Pembelian**  
Pengguna dapat melakukan checkout dan mendapatkan ringkasan pembelian dalam bentuk nota.

## Flow Chart Kode



# Penjelasan Sintaks dengan Implementasinya

## 1. Import Modul:

Skrip dimulai dengan mengimpor modul yang diperlukan. `tkinter` adalah antarmuka standar Python ke toolkit GUI Tk. `ttk` adalah submodul dari `tkinter` yang menyediakan akses ke set widget berbasis tema Tk. `messagebox` diimpor dari `tkinter` untuk memberikan kotak dialog standar.

```
1 import tkinter as tk
2 from tkinter import ttk, messagebox
```

## 2. Inisialisasi Data:

Dua kamus `produk` dan `kategori\_produk` diinisialisasi untuk menyimpan detail produk (nama dan harga) dan kategori masing-masing produk. Kamus `ongkir` menyimpan biaya pengiriman untuk lokasi yang berbeda.

- **Variabel Data Produk**

```
4 # Inisialisasi data produk
5 produk = {
6     'Lamp': 1000000,
7     'Smartphone': 5000000,
8     'Headphone': 1000000,
9     'Charger': 500000,
10    'Printer': 3000000,
11    'Mouse': 300000,
12    'Keyboard': 400000,
13    'Monitor': 1500000,
14    'Speaker': 800000,
15    'External HDD': 1200000,
16    'Tablet': 2500000,
17    'Camera': 6000000,
18    'Gaming Console': 4500000,
19    'Desk Lamp': 250000,
20    'USB Flash Drive': 150000,
21    'Wireless Router': 800000,
22    'Fitness Tracker': 900000,
23    'Portable Speaker': 700000,
24    'Bluetooth Earphones': 400000,
25    'Smartwatch': 3000000,
26    'Electric Toothbrush': 200000,
27    'Air Purifier': 1800000,
28    'Coffee Maker': 1200000,
29    'Vacuum Cleaner': 2500000,
30    'Hair Dryer': 500000,
31    'Rice Cooker': 800000,
32    'Toaster': 300000,
33    'Blender': 700000,
34    'Drones': 4000000,
35    'Projector': 3500000,
36    'Digital Camera': 2500000,
37    'Power Bank': 300000,
38    'Action Camera': 1800000,
39    'External SSD': 2000000,
40    'Gaming Mouse': 600000,
41    'Wireless Earbuds': 1500000,
42    'Robot Vacuum': 3500000,
43    'Smart Thermostat': 800000,
44    'Smart Doorbell': 1200000,
45    'E-book Reader': 1000000,
46    'Instant Pot': 1800000,
47    'VR Headset': 3000000,
48    'Wireless Printer': 2000000,
49    'Electric Kettle': 600000,
50    'Soundbar': 1200000,
51    'Computer Microphone': 500000,
52    'Smart Scale': 800000,
53    'Car Dash Cam': 1000000,
54    'Coffee Grinder': 400000,
55    'Air Fryer': 1000000,
56    'Wireless Headphones': 2500000,
57    'Digital Drawing Tablet': 3200000,
58    'Smart TV': 7000000,
59    'Electric Scooter': 3000000,
60    'Security Camera System': 4000000,
61    'Cordless Drill': 1200000,
62    'Electric Blanket': 900000,
63    'Wireless Gaming Mouse': 1800000,
64    'Smart Light Bulbs': 500000,
65    'Digital Photo Frame': 800000,
66    'Portable Air Conditioner': 2500000,
67    'Wireless Charging Dock': 1500000,
68    'Digital Alarm Clock': 300000,
69    'Mini Fridge': 1200000,
70    'Electric Skateboard': 3200000,
71    'Robot Lawn Mower': 1800000,
72    'Smart Ceiling Fan': 1000000,
73    'Home Theater System': 8000000,
74    'Robot Bartender': 600000,
75    'Smart Coffee Machine': 2000000,
76    'Wireless Meat Thermometer': 500000,
77    'Smart Window Blinds': 3000000,
78    'UV Sanitizer': 1500000,
79    'Pet Camera': 1200000,
80    'Voice Translator': 1000000,
81    'Smart Mirror': 4000000,
82    'Smart Garden': 2500000,
83    'Smart Water Bottle': 800000,
84    'Smart Air Fryer': 2000000,
85    'Foldable Electric Bike': 3500000,
86    'Solar Power Bank': 1500000,
87    'Portable Espresso Maker': 1800000,
88    'Wireless Baby Monitor': 1800000,
89    'AI-Powered Backpack': 500000,
90 }
```

- **Variabel Data Kategori Produk**

```

92 kategori_produk = {
93     'Komputer': ['Laptop', 'Desktop', 'Mouse', 'Keyboard', 'Monitor', 'Printer', 'External HDD', 'Tablet'],
94     'Gadget': ['Smartphone', 'Smartwatch', 'Bluetooth Earphones', 'Fitness Tracker', 'Power Bank', 'Wireless Earbuds'],
95     'Elektronik Rumah Tangga': ['Desk Lamp', 'Air Purifier', 'Coffee Maker', 'Vacuum Cleaner', 'Hair Dryer'],
96     'Dapur': ['Rice Cooker', 'Toaster', 'Blender', 'Coffee Grinder', 'Air Fryer'],
97     'Kamera': ['Camera', 'Digital Camera', 'Action Camera'],
98     'Gaming': ['Gaming Console', 'Gaming Mouse'],
99     'Peralatan Elektronik': ['Charger', 'External SSD', 'Wireless Router', 'Computer Microphone', 'Smart Thermostat'],
100    'Smart Home': ['Smart Doorbell', 'Smart TV', 'Smart Scale', 'Smart Light Bulbs', 'Smart Mirror'],
101    'Perkakas Elektronik': ['Electric Toothbrush', 'Electric Kettle', 'Soundbar', 'Robot Vacuum'],
102    'Outdoor': ['Drone', 'Digital Drawing Tablet', 'Electric Scooter', 'Portable Espresso Maker'],
103    'Keamanan': ['Security Camera System', 'Car Dash Cam', 'AI-Powered Backpack'],
104    'Peralatan Rumah': ['Electric Blanket', 'Mini Fridge', 'Portable Air Conditioner', 'Cordless Drill'],
105    'Hiburan': ['Speaker', 'Portable Speaker', 'Projector', 'Soundbar', 'Home Theater System'],
106    'Lainnya': ['USB Flash Drive', 'Wireless Printer', 'Instant Pot', 'E-book Reader', 'VR Headset']
107 }

```

- **Variabel Data Ongkir Lokasi**

```

111 #Ongkir
112 ongkir = {
113     'Jakarta': 40000,
114     'Bandung': 35000,
115     'Yogyakarta': 5000,
116     'Surabaya': 35000,
117     'Denpasar': 60000,
118     'Medan': 80000,
119     'Semarang': 10000,
120     'Makassar': 90000,
121     'Palembang': 30000,
122     'Balikpapan': 75000,
123     'Padang': 60000,
124     'Manado': 120000,
125     'Pekanbaru': 55000,
126     'Banjarmasin': 85000,
127     'Malang': 20000,
128     'Purwokerto': 15000,
129     'Pontianak': 55000,
130     'Samarinda': 45000,
131     'Banda Aceh': 75000,
132     'Jayapura': 90000,
133     'Ambon': 80000,
134     'Manokwari': 85000,
135     'Ternate': 95000,
136     'Mamuju': 70000,
137     'Tarakan': 50000,
138     'Kendari': 60000,
139     'Gorontalo': 55000,
140     'Palangkaraya': 45000,
141     'Tanjung Pinang': 65000,
142 }

```

Daftar Produk
Pilih Kategori:

Dapur

Tampilkan Kategori

Laptop: Rp 10.000.000  
Smartphone: Rp 5.000.000  
Headphone: Rp 1.000.000  
Charger: Rp 500.000  
Printer: Rp 2.000.000  
Mouse: Rp 300.000  
Keyboard: Rp 400.000  
Monitor: Rp 1.500.000  
Speaker: Rp 800.000  
External HDD: Rp 1.200.000

### 3. Lokasi Default:

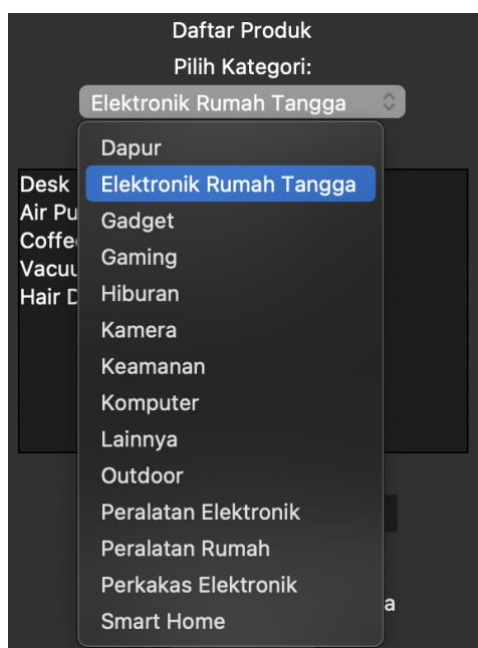
Variabel `lokasi` diatur sebagai 'Yogyakarta' sebagai lokasi pengiriman default.

```
144     lokasi = 'Yogyakarta' # Lokasi default
```

### 4. Fungsi `tampilkan\_produk\_kategori()`:

Fungsi ini memperbarui kotak daftar produk dengan produk dari kategori yang dipilih dalam menu dropdown.

```
146 def tampilkan_produk_kategori():
147     kategori = dropdown_kategori.get()
148     list_produk.delete(0, tk.END)
149     for category, items in kategori_produk.items():
150         if kategori == category:
151             for item in items:
152                 if item in produk:
153                     harga = produk[item]
154                     formatted_harga = "{:,.0f}".format(harga).replace(",", ".")
155                     list_produk.insert(tk.END, f"{item}: Rp {formatted_harga}")
```



### 5. Fungsi `tampilkan\_produk()`:

Fungsi ini mengisi kotak daftar produk dengan semua produk dan harganya saat program dimulai.

```
146 def tampilkan_produk_kategori():
147     kategori = dropdown_kategori.get()
148     list_produk.delete(0, tk.END)
149     for category, items in kategori_produk.items():
150         if kategori == category:
151             for item in items:
152                 if item in produk:
153                     harga = produk[item]
154                     formatted_harga = "{:,.0f}".format(harga).replace(",", ".")
155                     list_produk.insert(tk.END, f"{item}: Rp {formatted_harga}")
156
157
158 # Fungsi untuk menampilkan daftar produk
159 def tampilkan_produk():
160     list_produk.delete(0, tk.END)
161     for item, harga in produk.items():
162         formatted_harga = "{:,.0f}".format(harga).replace(",", ".")
163         list_produk.insert(tk.END, f"{item}: Rp {formatted_harga}")
```

Daftar Produk

Pilih Kategori:

Dapur

Tampilkan Kategori

Laptop: Rp 10.000.000  
 Smartphone: Rp 5.000.000  
 Headphone: Rp 1.000.000  
 Charger: Rp 500.000  
 Printer: Rp 2.000.000  
 Mouse: Rp 300.000  
 Keyboard: Rp 400.000  
 Monitor: Rp 1.500.000  
 Speaker: Rp 800.000  
 External HDD: Rp 1.200.000

## 6. Fungsi `tambah\_ke\_keranjang()`:

Fungsi ini menambahkan produk yang dipilih dan jumlahnya ke keranjang sambil menghitung harga total untuk item tersebut, termasuk biaya pengiriman.

```
165 def tambah_ke_keranjang():
166     pilihan = list_produk.get(tk.ACTIVE)
167     qty = int(entry_qty.get())
168
169     if qty > 0:
170         produk_terpilih = pilihan.split(':')[0].strip()
171         harga_produk = produk[produk_terpilih] * qty
172         biaya_ongkir = ongkir[lokasi]
173         total_harga_produk = harga_produk + biaya_ongkir
174
175         formatted_harga_produk = "Rp {:.0f}".format(harga_produk).replace(",", ".")
176         formatted_total_harga_produk = "Rp {:.0f}".format(total_harga_produk).replace(",", ".")
177
178         keranjang.insert('', tk.END, values=(produk_terpilih, qty, formatted_total_harga_produk))
179         update_total_keranjang()
180     else:
181         label_total_keranjang.config(text="Jumlah harus lebih besar dari 0.")
```

Ovic E-Store

Daftar Produk

Pilih Kategori:

Komputer

Tampilkan Kategori

Laptop: Rp 10.000.000  
 Mouse: Rp 300.000  
 Keyboard: Rp 400.000  
 Monitor: Rp 1.500.000  
 Printer: Rp 2.000.000  
 External HDD: Rp 1.200.000  
 Tablet: Rp 3.500.000

Jumlah:

1

Tambah ke Keranjang

Lokasi Pengiriman: Medan

Medan

Ubah Lokasi

Metode Pembayaran: Dana

Proses Pembayaran

Checkout

Keranjang		
Produk	Qty	Harga
Laptop	1	Rp 10.080.000

Hapus dari Keranjang

Total Pembelian: Rp 10.080.000

## 7. Fungsi `update\_total\_keranjang()`:

Fungsi ini menghitung ulang dan memperbarui total biaya semua item dalam keranjang belanja.

```
183 def update_total_keranjang():
184     total_harga_keranjang = sum(float(keranjang.item(item, 'values')[2].split()[1].replace(".", "").replace(",", "")) for item in keranjang.get_children())
185     formatted_total_harga_keranjang = "Rp {:.0f}".format(total_harga_keranjang).replace(".", ",")
186     label_total_keranjang.config(text=f"Total Pembelian: {formatted_total_harga_keranjang}")
```

Keranjang		
Produk	Qty	Harga
Laptop	1	Rp 10.080.000



Hapus dari Keranjang

Total Pembelian: Rp 10.080.000

## 8. Fungsi `hapus\_dari\_keranjang()`:

Fungsi ini menghapus item yang dipilih dari keranjang belanja dan memperbarui total biaya.

```
188 def hapus_dari_keranjang():
189     selected_item = keranjang.selection()[0] # Mengambil item yang dipilih
190     keranjang.delete(selected_item) # Menghapus item dari keranjang yang dipilih
191     update_total_keranjang() # Memperbarui total harga setelah penghapusan
```

Keranjang		
Produk	Qty	Harga



Hapus dari Keranjang

Total Pembelian: Rp 0

## 9. Fungsi `ganti\_lokasi()`:

Fungsi ini memperbarui lokasi pengiriman berdasarkan pilihan pengguna dari menu dropdown dan menghitung ulang biaya pengiriman untuk item dalam keranjang belanja.

```
193 def ganti_lokasi():
194     global lokasi
195     lokasi = dropdown_lokasi.get()
196     label_lokasi.config(text=f"Lokasi Pengiriman: {lokasi}")
197
198     # Memperbarui nilai biaya ongkir saat lokasi diubah
199     biaya_ongkir = ongkir[lokasi]
200
201     # Memperbarui nilai biaya ongkir untuk setiap item dalam keranjang
202     for item in keranjang.get_children():
203         qty = int(keranjang.item(item, 'values')[1])
204         produk_terpilih = keranjang.item(item, 'values')[0]
205         harga_produk = produk[produk_terpilih] * qty
206         total_harga_produk = harga_produk + biaya_ongkir
207         formatted_total_harga_produk = "Rp {:.0f}".format(total_harga_produk).replace(",", ".")
208         keranjang.item(item, values=(produk_terpilih, qty, formatted_total_harga_produk))
209
210     update_total_keranjang() # Memperbarui total harga setelah perubahan lokasi
```

Saya mengganti lokasi yaitu Jakarta, maka yang terjadi adalah menghitung ulang biaya total sesuai dengan Data ongkir

The screenshot shows the Ovic E-Store application interface. On the left, there is a list of products with their prices: Laptop (Rp 10.000.000), Mouse (Rp 300.000), Keyboard (Rp 400.000), Monitor (Rp 1.500.000), Printer (Rp 2.000.000), External HDD (Rp 1.200.000), and Tablet (Rp 3.500.000). The 'Laptop' is selected. Below the product list, there is a 'Jumlah:' field with the value '1'. To the right, there is a 'Keranjang' (Cart) table with columns 'Produk', 'Qty', and 'Harga'. The table shows one item: 'Laptop' with a quantity of '1' and a price of 'Rp 10.040.000'. Below the cart, there is a 'Total Pembelian: Rp 10.040.000'. At the bottom, there are buttons for 'Tambah ke Keranjang', 'Hapus dari Keranjang', 'Ubah Lokasi', 'Proses Pembayaran', and 'Checkout'. The 'Lokasi Pengiriman' is set to 'Jakarta'.

Produk	Qty	Harga
Laptop	1	Rp 10.040.000



## 10. Fungsi `proses\_pembayaran()`:

Fungsi ini merupakan placeholder untuk proses pembayaran berdasarkan metode pembayaran yang dipilih oleh pengguna.

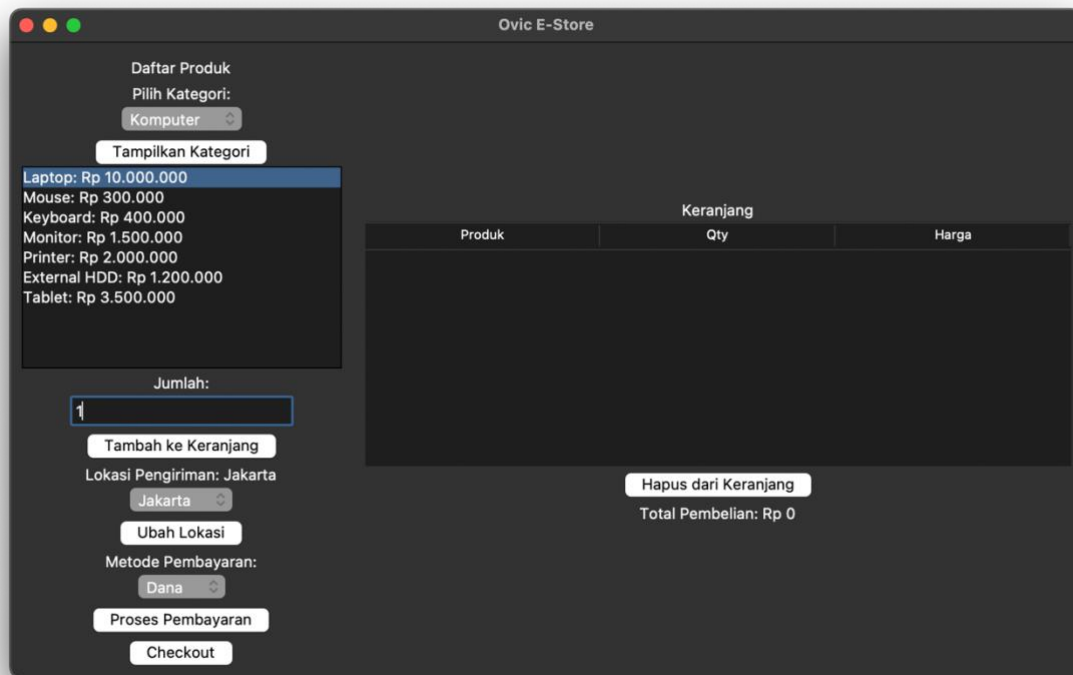
```
213 def proses_pembayaran():
214     # Memproses pembayaran berdasarkan metode yang dipilih
215     metode_pembayaran = dropdown_metode.get()
216
217     if metode_pembayaran == "Dana":
218         # Proses pembayaran dengan metode Dana
219         print("Pembayaran menggunakan Dana")
220     elif metode_pembayaran == "OVO":
221         # Proses pembayaran dengan metode OVO
222         print("Pembayaran menggunakan OVO")
223     elif metode_pembayaran == "Gopay":
224         # Proses pembayaran dengan metode Gopay
225         print("Pembayaran menggunakan Gopay")
226     elif metode_pembayaran == "BNI":
227         # Proses pembayaran dengan metode BNI
228         print("Pembayaran menggunakan BNI")
229     elif metode_pembayaran == "BRI":
230         # Proses pembayaran dengan metode BRI
231         print("Pembayaran menggunakan BRI")
232     elif metode_pembayaran == "Bank Mandiri":
233         # Proses pembayaran dengan metode Bank Mandiri
234         print("Pembayaran menggunakan Bank Mandiri")
```



## 11. Fungsi `proses\_checkout()`:

Fungsi ini mengonfirmasi checkout, menampilkan struk pembelian dengan kotak pesan, dan mengosongkan keranjang setelah pembelian.

```
237 def proses_checkout():
238     confirm = messagebox.askokcancel("Konfirmasi Pembelian", "Anda yakin ingin melakukan pembelian?")
239     if confirm:
240         # Mengumpulkan informasi dari keranjang
241         items = [(keranjang.item(item, 'values')[0], keranjang.item(item, 'values')[1], keranjang.item(item, 'values')[2]) for item in keranjang.get_children()]
242         total_harga = label_total_keranjang.cget("text")
243
244         # Menyiapkan teks nota
245         nota = f"{'='*50}\n{'Nota Pembelian':^50}\n{'='*50}\n\n"
246         nota += f"{'Nama Produk':<25}{ 'Jumlah':<10}{ 'Harga':<15}\n\n"
247         nota += f"{'-'*50}\n\n"
248         for item in items:
249             nota += f"{item[0]:<25}{item[1]:<10}{item[2]:<15}\n\n"
250         nota += f"{'-'*50}\nTotal Pembelian: {total_harga}\n{'-'*50}\n\n"
251         nota += "Pesanan sedang Diproses dan akan Dikirim ke Lokasi anda\nTerima Kasih sudah Belanja di Ovic E-Store"
252
253         # Menampilkan nota dalam pesan box
254         messagebox.showinfo("Nota Pembelian", nota)
255
256         # Mengosongkan keranjang setelah pembelian
257         keranjang.delete(*keranjang.get_children())
258         update_total_keranjang()
```



Disini anggap saja saya sudah Checkout dan melewati Fase Generate Nota. Yang terjadi adalah Keranjang secara otomatis akan Kosong.

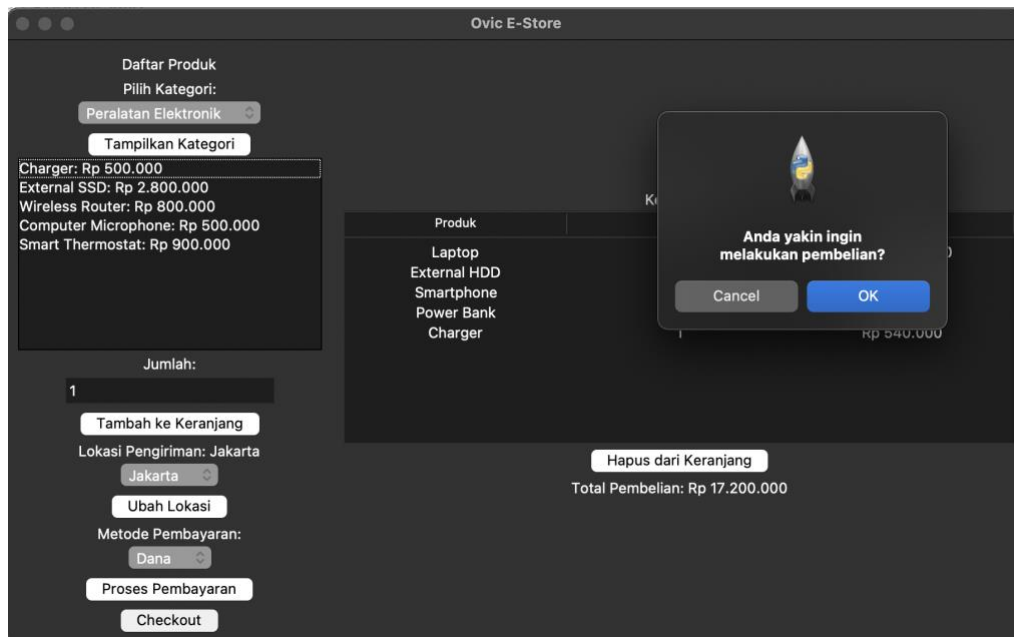
## 12. Fungsi `generate\_nota()`:

Fungsi ini tampaknya tidak digunakan untuk membuat catatan pembelian dan mengosongkan keranjang, tetapi tidak terikat pada tombol atau acara apa pun.

```

260 def generate_nota():
261     nota = f"==== Nota Pembelian =====\n"
262     total_harga = 0
263
264     for item in keranjang.get_children():
265         produk_terpilih = keranjang.item(item, 'values')[0]
266         qty = int(keranjang.item(item, 'values')[1])
267         harga_produk = produk[produk_terpilih]
268         total_harga_item = harga_produk * qty
269
270         formatted_total_harga_item = "Rp {:.0f}".format(total_harga_item).replace(",", ".")
271         nota += f"{produk_terpilih} - {qty} pcs - {formatted_total_harga_item}\n"
272         total_harga += total_harga_item
273
274     formatted_total_harga = "Rp {:.0f}".format(total_harga).replace(",", ".")
275     nota += f"Total Harga: {formatted_total_harga}\n\n"
276     nota += "Pesanan sedang Diproses dan akan Dikirim ke Lokasi anda\n"
277     nota += "Terima Kasih sudah Belanja di Ovic E-Store"
278
279     # Menampilkan nota dalam messagebox
280     tk.messagebox.showinfo("Nota Pembelian", nota)
281
282     # Bersihkan keranjang setelah pembelian
283     keranjang.delete(*keranjang.get_children())
284     update_total_keranjang()

```



Ini adalah proses setelah saya Checkout, yang terjadi adalah muncul Kolom Konfirmasi untuk melakukan Pembelian.



Setelah Checkout, maka akan muncul Tab Baru yaitu nota. Ini adalah Proses terakhir setelah Checkout yaitu Nota Pembelian dengan Rincian Nama Produk, Jumlah produk, dan Harga.

### 13. Setup Antarmuka Pengguna (GUI):

Antarmuka pengguna disusun dengan frame, label, kotak daftar, bidang entri, dan tombol. Tata letaknya dibagi menjadi dua frame utama: `frame\_produk`, yang berisi elemen terkait tampilan dan pemilihan produk, dan `frame\_keranjang`, yang berisi detail keranjang.

```
287 # Setup GUI
288 root = tk.Tk()
289 root.title("Ovic E-Store")
290
291 frame_produk = tk.Frame(root)
292 frame_produk.pack(side=tk.LEFT, padx=10, pady=10)
293
294 frame_keranjang = tk.Frame(root)
295 frame_keranjang.pack(side=tk.RIGHT, padx=10, pady=10)
296
297 label_produk = tk.Label(frame_produk, text="Daftar Produk")
298 label_produk.pack()
299
300 label_kategori = tk.Label(frame_produk, text="Pilih Kategori:")
301 label_kategori.pack()
302
303 kategori_options = sorted(list(kategori_produk.keys()))
304 dropdown_kategori = tk.StringVar(frame_produk)
305 dropdown_kategori.set(kategori_options[0])
306 dropdown_kategori_menu = tk.OptionMenu(frame_produk, dropdown_kategori, *kategori_options)
307 dropdown_kategori_menu.pack()
308
309 btn_tampilkan_kategori = tk.Button(frame_produk, text="Tampilkan Kategori", command=tampilkan_produk_kategori)
310 btn_tampilkan_kategori.pack()
311
312 list_produk = tk.Listbox(frame_produk, width=30, height=10)
313 list_produk.pack()
314
315 tampilkan_produk()
316
317 label_qty = tk.Label(frame_produk, text="Jumlah:")
318 label_qty.pack()
319 entry_qty = tk.Entry(frame_produk)
320 entry_qty.pack()
321
322 btn_tambah = tk.Button(frame_produk, text="Tambah ke Keranjang", command=tambah_ke_keranjang)
323 btn_tambah.pack()
324
325 label_lokasi = tk.Label(frame_produk, text=f"Lokasi Pengiriman: {lokasi}")
326 label_lokasi.pack()
327
328 lokasi_options = list(ongkir.keys())
329 dropdown_lokasi = tk.StringVar(frame_produk)
330 dropdown_lokasi.set(lokasi_options[0])
331 dropdown_lokasi_menu = tk.OptionMenu(frame_produk, dropdown_lokasi, *lokasi_options)
332 dropdown_lokasi_menu.pack()
333
334 btn_ganti_lokasi = tk.Button(frame_produk, text="Ubah Lokasi", command=ganti_lokasi)
335 btn_ganti_lokasi.pack()
336
337 label_keranjang = tk.Label(frame_keranjang, text="Keranjang")
338 label_keranjang.pack()
339
340 columns = ('Produk', 'Qty', 'Harga')
341 keranjang = ttk.Treeview(frame_keranjang, columns=columns, show='headings')
342
343 for col in columns:
344     keranjang.heading(col, text=col)
345     keranjang.column(col, anchor='center')
346
347 keranjang.pack()
348
349 btn_hapus = tk.Button(frame_keranjang, text="Hapus dari Keranjang", command=hapus_dari_keranjang)
350 btn_hapus.pack()
351
352 label_total_keranjang = tk.Label(frame_keranjang, text="Total Pembelian: ")
353 label_total_keranjang.pack()
354
355 label_metode_pembayaran = tk.Label(frame_produk, text="Metode Pembayaran:")
356 label_metode_pembayaran.pack()
357
358 # Daftar metode pembayaran yang ditampilkan dalam dropdown
359 metode_options = ["Dana", "OVO", "Gopay", "BNI", "BRI", "Bank Mandiri"]
360 dropdown_metode = tk.StringVar(frame_produk)
361 dropdown_metode.set(metode_options[0])
362 dropdown_metode_menu = tk.OptionMenu(frame_produk, dropdown_metode, *metode_options)
363 dropdown_metode_menu.pack()
364
365 btn_proses_pembayaran = tk.Button(frame_produk, text="Proses Pembayaran", command=proses_pembayaran)
366 btn_proses_pembayaran.pack()
367
368 # Checkout button
369 btn_checkout = tk.Button(frame_produk, text="Checkout", command=proses_checkout)
370 btn_checkout.pack()
371
372 root.mainloop()
```

#### 14. Tampilan Produk dan Keranjang:

Kotak daftar digunakan untuk menampilkan produk, dan widget treeview digunakan untuk keranjang. Tombol disediakan untuk menambahkan produk ke keranjang, mengubah lokasi pengiriman, dan menghapus item dari keranjang.

#### 15. Checkout dan Pembayaran:

Ada opsi untuk memilih metode pembayaran, memproses pembayaran, dan melanjutkan ke checkout. Tombol checkout terikat pada fungsi `proses\_checkout`, yang menyelesaikan pembelian.

### Source Code

```
import tkinter as tk
from tkinter import ttk, messagebox

# Inisialisasi data produk
produk = {
    'Laptop': 10000000,
    'Smartphone': 5000000,
    'Headphone': 1000000,
    'Charger': 500000,
    'Printer': 2000000,
    'Mouse': 300000,
    'Keyboard': 400000,
    'Monitor': 1500000,
    'Speaker': 800000,
    'External HDD': 1200000,
    'Tablet': 3500000,
    'Camera': 6000000,
    'Gaming Console': 4500000,
    'Desk Lamp': 250000,
    'USB Flash Drive': 150000,
    'Wireless Router': 800000,
    'Fitness Tracker': 900000,
    'Portable Speaker': 700000,
    'Bluetooth Earphones': 400000,
    'Smartwatch': 3000000,
    'Electric Toothbrush': 200000,
    'Air Purifier': 1800000,
    'Coffee Maker': 1200000,
    'Vacuum Cleaner': 2500000,
    'Hair Dryer': 500000,
    'Rice Cooker': 800000,
    'Toaster': 300000,
    'Blender': 700000,
    'Drone': 4000000,
    'Projector': 3500000,
    'Digital Camera': 2500000,
    'Power Bank': 300000,
    'Action Camera': 1800000,
```

```

'External SSD': 2800000,
'Gaming Mouse': 600000,
'Wireless Earbuds': 1500000,
'Robot Vacuum': 3500000,
'Smart Thermostat': 900000,
'Smart Doorbell': 1200000,
'E-book Reader': 1000000,
'Instant Pot': 1800000,
'VR Headset': 3000000,
'Wireless Printer': 2000000,
'Electric Kettle': 600000,
'Soundbar': 1200000,
'Computer Microphone': 500000,
'Smart Scale': 800000,
'Car Dash Cam': 1000000,
'Coffee Grinder': 400000,
'Air Fryer': 1800000,
'Wireless Headphones': 2500000,
'Digital Drawing Tablet': 3500000,
'Smart TV': 7000000,
'Electric Scooter': 3000000,
'Security Camera System': 4000000,
'Cordless Drill': 1200000,
'Electric Blanket': 900000,
'Wireless Gaming Mouse': 1800000,
'Smart Light Bulbs': 500000,
'Digital Photo Frame': 800000,
'Portable Air Conditioner': 2500000,
'Wireless Charging Dock': 1500000,
'Digital Alarm Clock': 300000,
'Mini Fridge': 1200000,
'Electric Skateboard': 3500000,
'Robot Lawn Mower': 5000000,
'Smart Ceiling Fan': 1800000,
'Home Theater System': 8000000,
'Robot Bartender': 6000000,
'Smart Coffee Machine': 2000000,
'Wireless Meat Thermometer': 500000,
'Smart Window Blinds': 3000000,
'UV Sanitizer': 1500000,
'Pet Camera': 1200000,
'Voice Translator': 1000000,
'Smart Mirror': 4000000,
'Smart Garden': 2500000,
'Smart Water Bottle': 800000,
'Smart Air Fryer': 2000000,
'Foldable Electric Bike': 3500000,
'Solar Power Bank': 1500000,
'Portable Espresso Maker': 1000000,
'Wireless Baby Monitor': 1800000,
'AI-Powered Backpack': 500000,
}

```

```

kategori_produk = {
    'Komputer': ['Laptop', 'Desktop', 'Mouse', 'Keyboard', 'Monitor',
'Printer', 'External HDD', 'Tablet'],
    'Gadget': ['Smartphone', 'Smartwatch', 'Bluetooth Earphones',
'Fitness Tracker', 'Power Bank', 'Wireless Earbuds'],

```

```

    'Elektronik Rumah Tangga': ['Desk Lamp', 'Air Purifier', 'Coffee
Maker', 'Vacuum Cleaner', 'Hair Dryer'],
    'Dapur': ['Rice Cooker', 'Toaster', 'Blender', 'Coffee Grinder',
'Air Fryer'],
    'Kamera': ['Camera', 'Digital Camera', 'Action Camera'],
    'Gaming': ['Gaming Console', 'Gaming Mouse'],
    'Peralatan Elektronik': ['Charger', 'External SSD', 'Wireless
Router', 'Computer Microphone', 'Smart Thermostat'],
    'Smart Home': ['Smart Doorbell', 'Smart TV', 'Smart Scale', 'Smart
Light Bulbs', 'Smart Mirror'],
    'Perkakas Elektronik': ['Electric Toothbrush', 'Electric Kettle',
'Soundbar', 'Robot Vacuum'],
    'Outdoor': ['Drone', 'Digital Drawing Tablet', 'Electric Scooter',
'Portable Espresso Maker'],
    'Keamanan': ['Security Camera System', 'Car Dash Cam', 'AI-Powered
Backpack'],
    'Peralatan Rumah': ['Electric Blanket', 'Mini Fridge', 'Portable
Air Conditioner', 'Cordless Drill'],
    'Hiburan': ['Speaker', 'Portable Speaker', 'Projector', 'Soundbar',
'Home Theater System'],
    'Lainnya': ['USB Flash Drive', 'Wireless Printer', 'Instant Pot',
'E-book Reader', 'VR Headset']
}

```

```

#Ongkir
ongkir = {
    'Jakarta': 40000,
    'Bandung': 35000,
    'Yogyakarta': 5000,
    'Surabaya': 35000,
    'Denpasar': 60000,
    'Medan': 80000,
    'Semarang': 10000,
    'Makassar': 90000,
    'Palembang': 30000,
    'Balikpapan': 75000,
    'Padang': 60000,
    'Manado': 120000,
    'Pekanbaru': 55000,
    'Banjarmasin': 85000,
    'Malang': 20000,
    'Purwokerto': 15000,
    'Pontianak': 55000,
    'Samarinda': 45000,
    'Banda Aceh': 75000,
    'Jayapura': 90000,
    'Ambon': 80000,
    'Manokwari': 85000,
    'Ternate': 95000,
    'Mamuju': 70000,
    'Tarakan': 50000,
    'Kendari': 60000,
    'Gorontalo': 55000,
    'Palangkaraya': 45000,
    'Tanjung Pinang': 65000,
}

```

```

lokasi = 'Yogyakarta' # Lokasi default

def tampilkan_produk_kategori():
    kategori = dropdown_kategori.get()
    list_produk.delete(0, tk.END)
    for category, items in kategori_produk.items():
        if kategori == category:
            for item in items:
                if item in produk:
                    harga = produk[item]
                    formatted_harga =
"{:, .0f}".format(harga).replace(",", "", ".")
                    list_produk.insert(tk.END, f"{item}: Rp
{formatted_harga}")

# Fungsi untuk menampilkan daftar produk
def tampilkan_produk():
    list_produk.delete(0, tk.END)
    for item, harga in produk.items():
        formatted_harga = "{:, .0f}".format(harga).replace(",", "", ".")
        list_produk.insert(tk.END, f"{item}: Rp {formatted_harga}")

def tambah_ke_keranjang():
    pilihan = list_produk.get(tk.ACTIVE)
    qty = int(entry_qty.get())

    if qty > 0:
        produk_terpilih = pilihan.split(':')[0].strip()
        harga_produk = produk[produk_terpilih] * qty
        biaya_ongkir = ongkir[lokasi]
        total_harga_produk = harga_produk + biaya_ongkir

        formatted_harga_produk = "Rp
{:, .0f}".format(harga_produk).replace(",", "", ".")
        formatted_total_harga_produk = "Rp
{:, .0f}".format(total_harga_produk).replace(",", "", ".")

        keranjang.insert('', tk.END, values=(produk_terpilih, qty,
formatted_total_harga_produk))
        update_total_keranjang()
    else:
        label_total_keranjang.config(text="Jumlah harus lebih besar
dari 0.")

def update_total_keranjang():
    total_harga_keranjang = sum(float(keranjang.item(item,
'values')[2].split()[1].replace(".", "").replace(",", "")) for item in
keranjang.get_children())
    formatted_total_harga_keranjang = "Rp
{:, .0f}".format(total_harga_keranjang).replace(",", "", ".")
    label_total_keranjang.config(text=f"Total Pembelian:
{formatted_total_harga_keranjang}")

def hapus_dari_keranjang():
    selected_item = keranjang.selection()[0] # Mengambil item yang
dipilih

```



```

    keranjang.delete(selected_item) # Menghapus item dari keranjang
    yang dipilih
    update_total_keranjang() # Memperbarui total harga setelah
    penghapusan

def ganti_lokasi():
    global lokasi
    lokasi = dropdown_lokasi.get()
    label_lokasi.config(text=f"Lokasi Pengiriman: {lokasi}")

    # Memperbarui nilai biaya ongkir saat lokasi diubah
    biaya_ongkir = ongkir[lokasi]

    # Memperbarui nilai biaya ongkir untuk setiap item dalam keranjang
    for item in keranjang.get_children():
        qty = int(keranjang.item(item, 'values')[1])
        produk_terpilih = keranjang.item(item, 'values')[0]
        harga_produk = produk[produk_terpilih] * qty
        total_harga_produk = harga_produk + biaya_ongkir
        formatted_total_harga_produk = "Rp
{:,.0f)".format(total_harga_produk).replace(",", ".")
        keranjang.item(item, values=(produk_terpilih, qty,
formatted_total_harga_produk))

    update_total_keranjang() # Memperbarui total harga setelah
    perubahan lokasi

def proses_pembayaran():
    # Memproses pembayaran berdasarkan metode yang dipilih
    metode_pembayaran = dropdown_metode.get()

    if metode_pembayaran == "Dana":
        # Proses pembayaran dengan metode Dana
        print("Pembayaran menggunakan Dana")
    elif metode_pembayaran == "OVO":
        # Proses pembayaran dengan metode OVO
        print("Pembayaran menggunakan OVO")
    elif metode_pembayaran == "Gopay":
        # Proses pembayaran dengan metode Gopay
        print("Pembayaran menggunakan Gopay")
    elif metode_pembayaran == "BNI":
        # Proses pembayaran dengan metode BNI
        print("Pembayaran menggunakan BNI")
    elif metode_pembayaran == "BRI":
        # Proses pembayaran dengan metode BRI
        print("Pembayaran menggunakan BRI")
    elif metode_pembayaran == "Bank Mandiri":
        # Proses pembayaran dengan metode Bank Mandiri
        print("Pembayaran menggunakan Bank Mandiri")

def proses_checkout():
    confirm = messagebox.askokcancel("Konfirmasi Pembelian", "Anda
    yakin ingin melakukan pembelian?")
    if confirm:
        # Mengumpulkan informasi dari keranjang

```

```

        items = [(keranjang.item(item, 'values')[0],
keranjang.item(item, 'values')[1], keranjang.item(item, 'values')[2])
for item in keranjang.get_children()]
        total_harga = label_total_keranjang.cget("text")

        # Menyiapkan teks nota
        nota = f"{'='*50}\n{'Nota Pembelian':^50}\n{'='*50}\n\n"
        nota += f"{'Nama Produk':<25}{'Jumlah':<10}{'Harga':<15}\n"
        nota += f"{'-'*50}\n"
        for item in items:
            nota += f"{item[0]:<25}{item[1]:<10}{item[2]}\n"
        nota += f"\n{'-'*50}\nTotal Pembelian: {total_harga}\n{'-'
'*50}\n\n"
        nota += "Pesanan sedang Diproses dan akan Dikirim ke Lokasi
anda\nTerima Kasih sudah Belanja di Ovic E-Store"

        # Menampilkan nota dalam pesan box
        messagebox.showinfo("Nota Pembelian", nota)

        # Mengosongkan keranjang setelah pembelian
        keranjang.delete(*keranjang.get_children())
        update_total_keranjang()

def generate_nota():
    nota = f"==== Nota Pembelian =====\n"
    total_harga = 0

    for item in keranjang.get_children():
        produk_terpilih = keranjang.item(item, 'values')[0]
        qty = int(keranjang.item(item, 'values')[1])
        harga_produk = produk[produk_terpilih]
        total_harga_item = harga_produk * qty

        formatted_total_harga_item = "Rp
{:, .0f}".format(total_harga_item).replace(",", ".")
        nota += f"{produk_terpilih} - {qty} pcs -
{formatted_total_harga_item}\n"
        total_harga += total_harga_item

    formatted_total_harga = "Rp
{:, .0f}".format(total_harga).replace(",", ".")
    nota += f"Total Harga: {formatted_total_harga}\n\n"
    nota += "Pesanan sedang Diproses dan akan Dikirim ke Lokasi anda\n"
    nota += "Terima Kasih sudah Belanja di Ovic E-Store"

    # Menampilkan nota dalam messagebox
    tk.messagebox.showinfo("Nota Pembelian", nota)

    # Bersihkan keranjang setelah pembelian
    keranjang.delete(*keranjang.get_children())
    update_total_keranjang()

# Setup GUI
root = tk.Tk()
root.title("Ovic E-Store")

frame_produk = tk.Frame(root)

```

```

frame_produk.pack(side=tk.LEFT, padx=10, pady=10)

frame_keranjang = tk.Frame(root)
frame_keranjang.pack(side=tk.RIGHT, padx=10, pady=10)

label_produk = tk.Label(frame_produk, text="Daftar Produk")
label_produk.pack()

label_kategori = tk.Label(frame_produk, text="Pilih Kategori:")
label_kategori.pack()

kategori_options = sorted(list(kategori_produk.keys()))
dropdown_kategori = tk.StringVar(frame_produk)
dropdown_kategori.set(kategori_options[0])
dropdown_kategori_menu = tk.OptionMenu(frame_produk, dropdown_kategori,
*kategori_options)
dropdown_kategori_menu.pack()

btn_tampilkan_kategori = tk.Button(frame_produk, text="Tampilkan
Kategori", command=tampilkan_produk_kategori)
btn_tampilkan_kategori.pack()

list_produk = tk.Listbox(frame_produk, width=30, height=10)
list_produk.pack()

tampilkan_produk()

label_qty = tk.Label(frame_produk, text="Jumlah:")
label_qty.pack()
entry_qty = tk.Entry(frame_produk)
entry_qty.pack()

btn_tambah = tk.Button(frame_produk, text="Tambah ke Keranjang",
command=tambah_ke_keranjang)
btn_tambah.pack()

label_lokasi = tk.Label(frame_produk, text=f"Lokasi Pengiriman:
{lokasi}")
label_lokasi.pack()

lokasi_options = list(ongkir.keys())
dropdown_lokasi = tk.StringVar(frame_produk)
dropdown_lokasi.set(lokasi_options[0])
dropdown_lokasi_menu = tk.OptionMenu(frame_produk, dropdown_lokasi,
*lokasi_options)
dropdown_lokasi_menu.pack()

btn_ganti_lokasi = tk.Button(frame_produk, text="Ubah Lokasi",
command=ganti_lokasi)
btn_ganti_lokasi.pack()

label_keranjang = tk.Label(frame_keranjang, text="Keranjang")
label_keranjang.pack()

columns = ('Produk', 'Qty', 'Harga')
keranjang = ttk.Treeview(frame_keranjang, columns=columns,
show='headings')

```

```

for col in columns:
    keranjang.heading(col, text=col)
    keranjang.column(col, anchor='center')

keranjang.pack()

btn_hapus = tk.Button(frame_keranjang, text="Hapus dari Keranjang",
command=hapus_dari_keranjang)
btn_hapus.pack()

label_total_keranjang = tk.Label(frame_keranjang, text="Total
Pembelian: ")
label_total_keranjang.pack()

label_metode_pembayaran = tk.Label(frame_produk, text="Metode
Pembayaran:")
label_metode_pembayaran.pack()

# Daftar metode pembayaran yang ditampilkan dalam dropdown
metode_options = ["Dana", "OVO", "Gopay", "BNI", "BRI", "Bank Mandiri"]
dropdown_metode = tk.StringVar(frame_produk)
dropdown_metode.set(metode_options[0])
dropdown_metode_menu = tk.OptionMenu(frame_produk, dropdown_metode,
*metode_options)
dropdown_metode_menu.pack()

btn_proses_pembayaran = tk.Button(frame_produk, text="Proses
Pembayaran", command=proses_pembayaran)
btn_proses_pembayaran.pack()

#Checkoutbutton
btn_checkout = tk.Button(frame_produk, text="Checkout",
command=proses_checkout)
btn_checkout.pack()

root.mainloop()

```