

PBO

Implementasi tampilan GUI studikusus struk pembayaran toko menggunakan python

**Materi ini disusun untuk memenuhi tugas mata kuliah
Pemmrogra berbasis objek**



Disusun oleh:

Rafli Haikal Putra (5230411301)

Informatika F

**PROGRAM STUDI INFORMATIKA FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA**

2024

Untuk studi kasus Sistem Manajemen Laundry Baju dengan GUI menggunakan Python dan Tkinter, kita bisa membuat aplikasi yang memungkinkan pengguna untuk melakukan berbagai hal seperti:

1. Menambah data transaksi laundry: Menginputkan informasi mengenai jenis pakaian, berat, harga, dan status pencucian (selesai/belum).
2. Melihat daftar transaksi: Menampilkan daftar semua transaksi yang telah dimasukkan.
3. Mencari transaksi tertentu: Mencari transaksi berdasarkan ID atau nama pelanggan.
4. Menghitung total pembayaran: Berdasarkan jenis pakaian dan berat, menghitung total harga.

Fitur yang akan ada:

- Input Transaksi: Form untuk menambah data transaksi.
- Daftar Transaksi: Menampilkan transaksi yang sudah dimasukkan.
- Pencarian Transaksi: Mencari transaksi berdasarkan ID atau nama pelanggan.
- Perhitungan Harga: Menghitung harga laundry berdasarkan berat dan harga per jenis pakaian.

Desain UI:

1. Form Input: Menyediakan form untuk memasukkan nama pelanggan, jenis pakaian, berat, dan status.
2. Tabel Daftar Transaksi: Menampilkan transaksi yang sudah dimasukkan.
3. Tombol Operasi: Tombol untuk menambah transaksi, mencari transaksi, dan menghitung total.

Berikut adalah Codingan implementasi aplikasi Sistem Manajemen Laundry Baju

```
import tkinter as tk
```

```
from tkinter import messagebox
```

```
from tkinter import ttk
```

```
# List untuk menyimpan data transaksi laundry
```

```
transaksi_list = []
```

```
# Fungsi untuk menambah transaksi
```

```
def tambah_transaksi():
```

```
    try:
```

```

# Ambil data dari inputan
nama_pelanggan = entry_nama.get()
jenis_pakaian = combo_jenis.get()
berat = float(entry_berat.get())
harga_per_kg = harga_jenis[jenis_pakaian]
total_harga = harga_per_kg * berat

# Menyimpan transaksi ke dalam list
transaksi = {
    'ID': len(transaksi_list) + 1,
    'Nama Pelanggan': nama_pelanggan,
    'Jenis Pakaian': jenis_pakaian,
    'Berat (kg)': berat,
    'Harga (IDR)': total_harga,
    'Status': 'Belum Selesai'
}
transaksi_list.append(transaksi)

# Tambahkan transaksi ke tabel
tampilkan_transaksi()

# Reset input form
entry_nama.delete(0, tk.END)
entry_berat.delete(0, tk.END)

messagebox.showinfo("Sukses", "Transaksi berhasil ditambahkan!")
except ValueError:
    messagebox.showerror("Input Error", "Silakan masukkan angka yang valid untuk berat.")

```

```
# Fungsi untuk menampilkan transaksi di tabel
```

```
def tampilkan_transaksi():
```

```
    for row in treeview.get_children():
```

```
        treeview.delete(row)
```

```
    for transaksi in transaksi_list:
```

```
        treeview.insert("", tk.END, values=(transaksi['ID'], transaksi['Nama Pelanggan'], transaksi['Jenis Pakaian'],
```

```
            transaksi['Berat (kg)'], transaksi['Harga (IDR)'], transaksi['Status']))
```

```
# Fungsi untuk mencari transaksi berdasarkan ID
```

```
def cari_transaksi():
```

```
    search_id = entry_search_id.get()
```

```
    if not search_id.isdigit():
```

```
        messagebox.showerror("Input Error", "ID Transaksi harus berupa angka.")
```

```
    return
```

```
    search_id = int(search_id)
```

```
    found = False
```

```
    for transaksi in transaksi_list:
```

```
        if transaksi['ID'] == search_id:
```

```
            found = True
```

```
            treeview.delete(*treeview.get_children())
```

```
            treeview.insert("", tk.END, values=(transaksi['ID'], transaksi['Nama Pelanggan'], transaksi['Jenis Pakaian'],
```

```
                transaksi['Berat (kg)'], transaksi['Harga (IDR)'], transaksi['Status']))
```

```
            break
```

```
    if not found:
```

```
messagebox.showwarning("Tidak Ditemukan", "Transaksi dengan ID tersebut tidak ditemukan.")
```

```
# Fungsi untuk mengupdate status transaksi menjadi selesai
```

```
def update_status():
```

```
    try:
```

```
        selected_item = treeview.selection()[0]
```

```
        transaksi_id = treeview.item(selected_item, 'values')[0]
```

```
        for transaksi in transaksi_list:
```

```
            if transaksi['ID'] == transaksi_id:
```

```
                transaksi['Status'] = 'Selesai'
```

```
            break
```

```
        tampilkan_transaksi()
```

```
        messagebox.showinfo("Sukses", "Status transaksi berhasil diperbarui menjadi Selesai.")
```

```
    except IndexError:
```

```
        messagebox.showwarning("Peringatan", "Silakan pilih transaksi yang ingin diupdate.")
```

```
# Definisi harga per kg untuk jenis pakaian
```

```
harga_jenis = {
```

```
    "Kaos": 15000,
```

```
    "Celana": 20000,
```

```
    "Jaket": 25000,
```

```
    "Kemeja": 18000
```

```
}
```

```
# Membuat root window
```

```
root = tk.Tk()
```

```
root.title("Sistem Manajemen Laundry Baju")
```

```
# Form Input Transaksi
```

```
frame_input = tk.Frame(root)
frame_input.grid(row=0, column=0, padx=10, pady=10, sticky="w")

label_nama = tk.Label(frame_input, text="Nama Pelanggan:")
label_nama.grid(row=0, column=0, padx=5, pady=5)
entry_nama = tk.Entry(frame_input)
entry_nama.grid(row=0, column=1, padx=5, pady=5)

label_jenis = tk.Label(frame_input, text="Jenis Pakaian:")
label_jenis.grid(row=1, column=0, padx=5, pady=5)
combo_jenis = ttk.Combobox(frame_input, values=["Kaos", "Celana", "Jaket", "Kemeja"])
combo_jenis.grid(row=1, column=1, padx=5, pady=5)
combo_jenis.set("Kaos")

label_berat = tk.Label(frame_input, text="Berat (kg):")
label_berat.grid(row=2, column=0, padx=5, pady=5)
entry_berat = tk.Entry(frame_input)
entry_berat.grid(row=2, column=1, padx=5, pady=5)

button_tambah = tk.Button(frame_input, text="Tambah Transaksi", command=tambah_transaksi)
button_tambah.grid(row=3, column=0, columnspan=2, pady=10)

# Pencarian Transaksi
frame_search = tk.Frame(root)
frame_search.grid(row=1, column=0, padx=10, pady=10, sticky="w")

label_search_id = tk.Label(frame_search, text="Cari ID Transaksi:")
label_search_id.grid(row=0, column=0, padx=5, pady=5)
entry_search_id = tk.Entry(frame_search)
```

```
entry_search_id.grid(row=0, column=1, padx=5, pady=5)
```

```
button_cari = tk.Button(frame_search, text="Cari", command=cari_transaksi)
```

```
button_cari.grid(row=0, column=2, padx=5, pady=5)
```

```
# Tabel Daftar Transaksi
```

```
frame_tabel = tk.Frame(root)
```

```
frame_tabel.grid(row=2, column=0, padx=10, pady=10)
```

```
columns = ("ID", "Nama Pelanggan", "Jenis Pakaian", "Berat (kg)", "Harga (IDR)", "Status")
```

```
treeview = ttk.Treeview(frame_tabel, columns=columns, show="headings")
```

```
treeview.grid(row=0, column=0)
```

```
for col in columns:
```

```
    treeview.heading(col, text=col)
```

```
# Tombol Update Status
```

```
button_update_status = tk.Button(root, text="Update Status Selesai", command=update_status)
```

```
button_update_status.grid(row=3, column=0, pady=10)
```

```
# Menjalankan aplikasi
```

```
root.mainloop()
```

Penjelasan:

1. Form Input Transaksi:

- Pengguna bisa memasukkan nama pelanggan, memilih jenis pakaian (kaos, celana, jaket, kemeja), dan memasukkan berat pakaian.
- Aplikasi akan menghitung total harga berdasarkan jenis pakaian dan berat yang dimasukkan.

2. Daftar Transaksi:

- Semua transaksi yang dimasukkan akan ditampilkan dalam tabel dengan ID, nama pelanggan, jenis pakaian, berat, harga, dan status.

3. Fitur Pencarian:

- Pengguna dapat mencari transaksi berdasarkan ID yang dimasukkan. Jika transaksi ditemukan, akan ditampilkan di tabel.

4. Update Status:

- Pengguna dapat memperbarui status transaksi yang dipilih menjadi "Selesai".

Cara Menggunakan:

1. **Tambah Transaksi:** Isi form dan klik "Tambah Transaksi".
2. **Cari Transaksi:** Masukkan ID transaksi dan klik "Cari".
3. **Update Status:** Pilih transaksi yang ingin diupdate dan klik "Update Status Selesai".

Contoh Penggunaan:

Figure 1

Sistem Manajemen Laundry Baju

Nama Pelanggan:

Jenis Pakaian:

Kaos

Berat (kg):

Tambah Transaksi

Cari ID Transaksi:

Cari

ID	Nama Pelanggan	Jenis Pakaian	Berat (kg)	Harga (IDR)	Status
----	----------------	---------------	------------	-------------	--------

Update Status Selesai

Figure 2

Sistem Manajemen Laundry Baju

Nama Pelanggan:

Jenis Pakaian:

Kemeja

Berat (kg):

Tambah Transaksi

Cari ID Transaksi:

Cari

ID	Nama Pelanggan	Jenis Pakaian	Berat (kg)	Harga (IDR)	Status
1	RH	Kaos	7.0	105000.0	Belum Selesai
2	HY	Kaos	10.0	150000.0	Belum Selesai
3	KEZMN	Celana	2.0	40000.0	Belum Selesai
4	KK	Jaket	3.0	75000.0	Belum Selesai
5	EPL	Kemeja	5.0	90000.0	Belum Selesai

Update Status Selesai

Figure 3

Sistem Manajemen Laundry Baju

Nama Pelanggan:

Jenis Pakaian:

Kemeja

Berat (kg):

Tambah Transaksi

Cari ID Transaksi:

2

Cari

ID	Nama Pelanggan	Jenis Pakaian	Berat (kg)	Harga (IDR)	Status
2	HY	Kaos	10.0	150000.0	Belum Selesai

Update Status Selesai

Sistem Manajemen Laundry Baju

Nama Pelanggan:

Jenis Pakaian:

Berat (kg):

Cari ID Transaksi:

ID	Nama Pelanggan	Jenis Pakaian	Berat (kg)	Harga (IDR)	Status
3	KEZMN	Celana	2.0	40000.0	Belum Selesai

Figure 5

Sistem Manajemen Laundry Baju

Nama Pelanggan:

Jenis Pakaian:

Berat (kg):

Cari ID Transaksi:

ID	Nama Pelanggan	Jenis Pakaian	Berat (kg)	Harga (IDR)	Status
5	EPL	Kemeja	5.0	90000.0	Belum Selesai