

LAPORAN
TUGAS GUI DENGAN PYTHON
MATA KULIAH : PEMROGRAMAN BERORIENTASI OBJEK



DISUSUN OLEH :
AJI KHARISMA ATMAJA (5230411292)

FAKULTAS SAINS DAN TEKNOLOGI
PROGRAM STUDI INFORMATIKA
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2024

LATAR BELAKANG KASUS

Kabupaten Sleman merupakan suatu wilayah atau kabupaten di Yogyakarta yang memiliki banyak sekali kegiatan social untuk membantu masyarakat, salah satunya mungkin adalah pembagian sembako kepada warganya. Pembagian sembako ini mungkin sudah terjadwal untuk pembagiannya. Lokasi untuk pembagian sembako adalah di balai pertemuan agar terdapat tempat yang sangat luas.

Namun pada saat pembagian sembako sering kali terjadi keributan akibat para warga berebut untuk mendapatkan sembako tersebut. Akibat situasi tersebut banyak sekali warga yang tidak nyaman bahkan sampai terluka karena berdesakkan. Tidak hanya itu, banyak juga paket sembako yang rusak akibat warga yang saling rebutan.

Untuk mengatasi masalah tersebut saya membuat sebuah system antrian sederhana. Studi kasus kali ini saya membuatnya berbasis desktop dengan Bahasa pemrograman python dan memanfaatkan library tkinter. Dalam system ini nanti akan membantu mengelola antrian warga yang ingin mendapatkan paket sembako. Kemudian memudahkan petugas untuk memanggil warga secara berurutan untuk mengambil paket sembako. Serta dapat memberikan informasi kepada para warga tentang jumlah paket sembako yang tersedia dan daftar antriannya.

Dengan adanya system antrian ini, diharapkan proses pembagian sembako berjalan dengan lebih tertib, teratur, dan efisien. Sistem ini juga memberikan kesempatan yang adil kepada warga yang lebih cepat mendaftar, sehingga mereka dapat menerima bantuan dengan nyaman tanpa harus berebut atau berdesak-desakan. Dan adanya system ini dapat mencegah kecurangan, karena 1 kepala keluarga hanya dapat mengambil 1 paket sembako.

PENJELASAN KODE PROGRAM

1. IMPORT UNTUK KEBUTUHAN PROGRAM

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

Mengimport tkinter yang merupakan library di python untuk membuat antar muka grafis (GUI). Pemberian alias tk agar lebih mudah digunakan di program. Import deque dari modul collections untuk struktur data antrian yang memungkinkan penambahan dan penghapusan elemen. Modul Themed Tkinter (ttk) Widgets menyediakan versi widget yang memiliki tema modern seperti tombol, label, atau tabel (Treeview). messagebox: Modul kotak dialog untuk menampilkan pesan kepada pengguna dalam bentuk pop-up (misalnya pesan kesalahan, informasi, atau peringatan).

2. MEMBUAT CLASS DAN FUNGSI KONSTRUKTOR

```
class antrian:
    def __init__(self, root):
        self.root = root
        self.root.title("Sistem Antrian Sembako")
        self.root.geometry("800x800")
        self.root.configure(bg="teal")
        self.widget_create()
```

Class yang bernama antrian itu untuk membuat GUI yang mengatur system antrian sembako. Fungsi konstruktor yang dipanggil otomatis dari kelas antrian. Dalam

fungsi ini dapat membuat judul GUI, ukuran untuk GUI dan background yang akan digunakan pada GUI. Widget create untuk membuat dan menambahkan widget (komponen GUI seperti tombol, label, input) ke dalam jendela utama.

3. FUNGSI WIDGET CREATE

A. 2 VARIABEL

```
queue_a = deque() #LOKET A
queue_b = deque() #LOKET B
```

Mendeklarasikan bahwa queue_a adalah antrian yang akan di masukkan kedalam loket A dan queue_b adalah antrian yang akan di masukkan kedalam loket B.

B. JUDUL TABEL

```
self.label = tk.Label(self.root, text="Antrian Sembako Kabupaten
Sleman", font=("Arial", 25, "bold"), background="teal", foreground="white
")
self.label.pack(pady=10)
```

Kode diatas berfungsi untuk membuat tema table dan bisa mengatur jenis font, ukuran font, warna background dan warna font nya. Kemudian mengatur jarak vertical nya yaitu 10 pixels

C. INPUT JUMLAH PAKET SEMBAKO

```
# INPUT JUMLAH PAKET SEMBAKO DI LOKET A
self.porsi_label_a = tk.Label(self.root, text="Jumlah Paket
Sembako di Loket
A:", background="teal", foreground="white", font=("Arial", 10, "bold"))
self.porsi_label_a.pack(pady=5)
self.porsi_entry_a = tk.Entry(self.root, width=10)
self.porsi_entry_a.pack(pady=5)

# INPUT JUMLAH PAKET SEMBAKO DI LOKET B
self.porsi_label_b = tk.Label(self.root, text="Jumlah Paket
Sembako di Loket
B:", background="teal", foreground="white", font=("Arial", 10, "bold"))
self.porsi_label_b.pack(pady=5)
self.porsi_entry_b = tk.Entry(self.root, width=10)
self.porsi_entry_b.pack(pady=5)
```

Dalam kode itu ada label untuk membuat teks statis bernama jumlah paket sembako di loket A dan B, membuat warna background, warna font, jenis dan ukuran font. Membuat inputan entry untuk menginputan jumlah paket sembako.

D. INPUT NAMA WARGA YANG MENGANTRI

```
# INPUT NAMA
self.nama_label = tk.Label(self.root, text="Nama Warga
:", background="teal", foreground="white", font=("Arial", 10, "bold"))
self.nama_label.pack(pady=5)
self.nama_entry = tk.Entry(self.root, width=15)
self.nama_entry.pack(pady=5)
```

Berfungsi untuk membuat nama label dimana di dalamnya juga dapat mengatur warna background, warna font, dan jenis font. Ada entry untuk membuat inputan teks yang dapat diinputkan oleh user dengan lebar 15 pixels. Semua komponen diatas ada jarak vertical nya sebesar 5 pixels.

E. TOMBOL TAMBAH ANTRIAN

```
# TOMBOL TAMBAH ANTRIAN
    frame_tambah = tk.Frame(self.root,background="teal")
    frame_tambah.pack(pady=5)

    #TOMBOL TAMBAH KE LOKET A
    self.tambah_a_button = tk.Button(frame_tambah, text="Tambah
Ke Antrian Locket
A",activebackground="red",background="white",command=self.tambah_ant
rian_a)
    self.tambah_a_button.pack(side=tk.LEFT, padx=10)
    #TOMBOL TAMBAH KE LOKET B
    self.tambah_b_button = tk.Button(frame_tambah, text="Tambah
Ke Antrian Locket
B",activebackground="red",background="white",command=self.tambah_ant
rian_b)
    self.tambah_b_button.pack(side=tk.RIGHT, padx=10)
```

Frame diatas berguna untuk mengatur agar tombol tambah itu letaknya bersampingan dan tidak atas bawah.Kemudian membuat tombol tambah untuk antrian loket A dan B dengan background aktif berwarna merah,background berwarna putih dan command itu untuk menghubungkan sebuah fungsi (atau method) yang akan dijalankan ketika tombol (widget Button) diklik oleh pengguna.Side=tk itu untuk mengatur tombol di sebelah kanan dan kiri.Jarak antar tombol yaitu 10 pixels.

F. FRAME TABEL DAN TOMBOL PANGGIL & LABEL PEMANGGILAN

```
# FRAME UNTUK TABEL ANTRIAN DAN TOMBOL PANGGIL
    frame_tabel = tk.Frame(self.root)
    frame_tabel.pack(pady=20)

    # LABEL PEMANGGILAN
    self.pemanggilan_label = tk.Label(self.root,
text="Memanggil: ", font=("Arial", 15,
"bold"),background="teal",foreground="white")
    self.pemanggilan_label.pack(pady=20)
```

Frame untuk mengatur agar table dan tombol panggil kesamping.Label pemanggilan untuk membuat teks statis memanggil dengan font arial berukuran 15, warna background dan warna font.

G. TABEL DAN TOMBOL PANGGIL LOKET A & B

```
# TABEL DAFTAR ANTRIAN LOKET A
    frame_a = tk.Frame(frame_tabel)
    frame_a.pack(side=tk.LEFT, padx=20)

    self.label_a = tk.Label(frame_a, text="Daftar Antrian di
Locket A")
    self.label_a.pack(pady=5)
    self.pesanan_label_a = ttk.Treeview(frame_a, columns=("No
Antrian", "Nama"), show="headings",height=8)
```

```

        self.pesanan_label_a.heading("No Antrian", text="No
Antrian")
        self.pesanan_label_a.heading("Nama", text="Nama")
        self.pesanan_label_a.column("No Antrian", width=100,
anchor="center")
        self.pesanan_label_a.column("Nama", width=150,
anchor="center")
        self.pesanan_label_a.pack(pady=5)

        #TOMBOL PANGGIL LOKET A
        self.panggil_a_button = tk.Button(frame_a, text="Panggil
Antrian Locket A",activebackground="red",background="yellow",
command=self.panggil_berikutnya_a)
        self.panggil_a_button.pack(pady=5)

        # TABEL DAFTAR ANTRIAN LOKET B
        frame_b = tk.Frame(frame_tabel)
        frame_b.pack(side=tk.RIGHT, padx=20)

        self.label_b = tk.Label(frame_b, text="Daftar Antrian Di
Locket B")
        self.label_b.pack(pady=5)
        self.pesanan_label_b = ttk.Treeview(frame_b, columns=("No
Antrian", "Nama"), show="headings",height=8)
        self.pesanan_label_b.heading("No Antrian", text="No
Antrian")
        self.pesanan_label_b.heading("Nama", text="Nama")
        self.pesanan_label_b.column("No Antrian", width=100,
anchor="center")
        self.pesanan_label_b.column("Nama", width=150,
anchor="center")
        self.pesanan_label_b.pack(pady=5)

        #TOMBOL PANGGIL LOKET B
        self.panggil_b_button = tk.Button(frame_b, text="Panggil
Antrian Locket B",activebackground="red",background="yellow",
command=self.panggil_berikutnya_b)
        self.panggil_b_button.pack(pady=5)

```

Pertama buat frame agar ke 2 tabel saling bersampingan. Kemudian buat teks statis nama table yaitu daftar antrian. Kemudian ada ttk.treeview untuk menampilkan data dalam bentuk tabel atau tree structure (struktur pohon). Widget ini sangat berguna untuk menampilkan data dalam bentuk baris dan kolom, seperti yang kamu lihat dalam kode di mana digunakan untuk menampilkan daftar antrian. Heading table ada No antrian dan nama. Kemudian ada tombol untuk memanggil antrian dan ada command untuk memanggil fungsi.

H. FUNGSI TAMBAH ANTRIAN

```

# TAMBAH ANTRIAN A
def tambah_antrian_a(self):

```

```

        nama = self.nama_entry.get()
        if nama in queue_a or nama in queue_b:
            messagebox.showerror("Error", f>Nama {nama} sudah ada dalam
antrian di salah satu loket.")
            return
        self.tambah_antrian(queue_a, self.porsi_entry_a,
self.pesanan_label_a)

# TAMBAH ANTRIAN B
def tambah_antrian_b(self):
    nama = self.nama_entry.get()
    if nama in queue_a or nama in queue_b:
        messagebox.showerror("Error", f>Nama {nama} sudah ada dalam
antrian di salah satu loket.")
        return
    self.tambah_antrian(queue_b, self.porsi_entry_b,
self.pesanan_label_b)

def tambah_antrian(self, queue, porsi_entry, pesanan_label):
    nama = self.nama_entry.get()
    porsi = porsi_entry.get()
    if nama != "" and porsi != "":
        try:
            porsi = int(porsi)
            if len(queue) < porsi:
                queue.append(nama)
                pesanan_label.insert("", "end",
values=(str(len(queue)), nama))
                self.nama_entry.delete(0, tk.END)
            else:
                messagebox.showinfo("Info", "Antrian penuh")
        except ValueError:
            messagebox.showerror("Error", "Porsi harus angka.")
    else:
        messagebox.showerror("Error", "Nama dan jumlah paket sembako
harus diisi.")

```

Fungsi ini berisi logika saat ingin menambahkan antrian. Nama harus terisi dan tidak boleh kosong, nama tidak boleh melebihi jumlah stok, dan nama yang diinputkan harus belum ada dalam daftar antrian, jika kondisi diatas tidak terpenuhi maka akan muncul pesan-pesan untuk memberitahu atau memperingati. Jika semua kondisi terpenuhi maka menambahkan nama ke daftar antrian berhasil dan jika table antrian akan di update sesuai nama-nama yang kita inputkan.

I. FUNGSI PANGGIL ANTRIAN

```

#PANGGIL ANTRIAN A
def panggil_berikutnya_a(self):
    if queue_a:
        # Panggil antrean pertama

```

```

        self.panggilan_label.config(text="Memanggil Untuk
Mengambil Sembako Di Loker A : " + queue_a.popleft())
        self.update_tabel_a()
    else:
        messagebox.showinfo("Info", "Antrian sudah kosong")

#PANGGIL ANTRIAN B
def panggil_berikutnya_b(self):
    if queue_b:
        # Panggil antrian pertama
        self.panggilan_label.config(text="Memanggil Untuk
Mengambil Sembako Di Loker B : " + queue_b.popleft())
        self.update_tabel_b()
    else:
        messagebox.showinfo("Info", "Antrian sudah kosong")

```

Fungsi diatas berfungsi untuk memanggil nama yang ada dalam antrian. Jika antrian tersebut masih ada maka dapat dipanggil dan akan di hapus dari table antrian jika table antrian sudah kosong maka akan ada pesan info yang muncul.

J. FUNGSI UPDATE ANTRIAN

```

def update_tabel_a(self):
    # Hapus semua data di tabel
    for item in self.pesanan_label_a.get_children():
        self.pesanan_label_a.delete(item)
    # Masukkan data antrian yang tersisa
    for idx, nama in enumerate(queue_a, start=1):
        self.pesanan_label_a.insert("", "end", values=(str(idx),
nama))

def update_tabel_b(self):
    # Hapus semua data di tabel
    for item in self.pesanan_label_b.get_children():
        self.pesanan_label_b.delete(item)
    # Masukkan data antrian yang tersisa
    for idx, nama in enumerate(queue_b, start=1):
        self.pesanan_label_b.insert("", "end", values=(str(idx),
nama))

```

Fungsi itu berfungsi untuk mengupdate antrian. Akan menghapus antrian yang sudah dipanggil dan data yang tersisa akan naik keatas, misal ada 2 antrian dan antrian pertama di panggil otomatis antrian ke 2 naik menjadi antrian pertama.

K. KODE RUNNING PROGRAM

```

if __name__ == '__main__':
    root = tk.Tk()
    app = antrian(root)
    root.mainloop()

```

Kode diatas untuk menjalankan aplikasi Tkinter dan memulai aplikasi GUI (Graphical User Interface) berbasis Python.

KODE PROGRAM YANG LENGKAP

```

import tkinter as tk
from collections import deque
from tkinter import ttk,messagebox

queue_a = deque() #LOKET A
queue_b = deque() #LOKET B

class antrian:
    def __init__(self,root):
        self.root = root
        self.root.title("Sistem Antrian Sembako")
        self.root.geometry("800x800")
        self.root.configure(bg="teal")
        self.widget_create()

    def widget_create(self):
        # TITLE TABEL
        self.label = tk.Label(self.root, text="Antrian Sembako
Kabupaten
Sleman",font=("Arial",25,"bold"),background="teal",foreground="white")
        self.label.pack(pady=10)

        # INPUT JUMLAH PAKET SEMBAKO DI LOKET A
        self.porsi_label_a = tk.Label(self.root, text="Jumlah Paket
Sembako di Locket
A:",background="teal",foreground="white",font=("Arial",10,"bold"))
        self.porsi_label_a.pack(pady=5)
        self.porsi_entry_a = tk.Entry(self.root, width=10)
        self.porsi_entry_a.pack(pady=5)

        # INPUT JUMLAH PAKET SEMBAKO DI LOKET B
        self.porsi_label_b = tk.Label(self.root, text="Jumlah Paket
Sembako di Locket
B:",background="teal",foreground="white",font=("Arial",10,"bold"))
        self.porsi_label_b.pack(pady=5)
        self.porsi_entry_b = tk.Entry(self.root, width=10)
        self.porsi_entry_b.pack(pady=5)

        # INPUT NAMA
        self.nama_label = tk.Label(self.root, text="Nama Warga
:",background="teal",foreground="white",font=("Arial",10,"bold"))
        self.nama_label.pack(pady=5)
        self.nama_entry = tk.Entry(self.root, width=15)
        self.nama_entry.pack(pady=5)

        # TOMBOL TAMBAH ANTRIAN
        frame_tambah = tk.Frame(self.root,background="teal")
        frame_tambah.pack(pady=5)

```



```

        #TOMBOL TAMBAH KE LOKET A
        self.tambah_a_button = tk.Button(frame_tambah, text="Tambah Ke
Antrian Locket
A",activebackground="red",background="white",command=self.tambah_antria
n_a)

        self.tambah_a_button.pack(side=tk.LEFT, padx=10)
        #TOMBOL TAMBAH KE LOKET B
        self.tambah_b_button = tk.Button(frame_tambah, text="Tambah Ke
Antrian Locket
B",activebackground="red",background="white",command=self.tambah_antria
n_b)

        self.tambah_b_button.pack(side=tk.RIGHT, padx=10)

        # FRAME UNTUK TABEL ANTRIAN DAN TOMBOL PANGGIL
        frame_tabel = tk.Frame(self.root)
        frame_tabel.pack(pady=20)

        # LABEL PEMANGGILAN
        self.pemanggilan_label = tk.Label(self.root, text="Memanggil:
", font=("Arial", 15, "bold"),background="teal",foreground="white")
        self.pemanggilan_label.pack(pady=20)

        # TABEL DAFTAR ANTRIAN LOKET A
        frame_a = tk.Frame(frame_tabel)
        frame_a.pack(side=tk.LEFT, padx=20)

        self.label_a = tk.Label(frame_a, text="Daftar Antrian di Locket
A")

        self.label_a.pack(pady=5)
        self.pesanan_label_a = ttk.Treeview(frame_a, columns=("No
Antrian", "Nama"), show="headings",height=8)
        self.pesanan_label_a.heading("No Antrian", text="No Antrian")
        self.pesanan_label_a.heading("Nama", text="Nama")
        self.pesanan_label_a.column("No Antrian", width=100,
anchor="center")
        self.pesanan_label_a.column("Nama", width=150, anchor="center")
        self.pesanan_label_a.pack(pady=5)

        #TOMBOL PANGGIL LOKET A
        self.panggil_a_button = tk.Button(frame_a, text="Panggil
Antrian Locket A",activebackground="red",background="yellow",
command=self.panggil_berikutnya_a)
        self.panggil_a_button.pack(pady=5)

        # TABEL DAFTAR ANTRIAN LOKET B
        frame_b = tk.Frame(frame_tabel)
        frame_b.pack(side=tk.RIGHT, padx=20)

```

```

        self.label_b = tk.Label(frame_b, text="Daftar Antrian Di Loker
B")
        self.label_b.pack(pady=5)
        self.pesanan_label_b = ttk.Treeview(frame_b, columns=("No
Antrian", "Nama"), show="headings", height=8)
        self.pesanan_label_b.heading("No Antrian", text="No Antrian")
        self.pesanan_label_b.heading("Nama", text="Nama")
        self.pesanan_label_b.column("No Antrian", width=100,
anchor="center")
        self.pesanan_label_b.column("Nama", width=150, anchor="center")
        self.pesanan_label_b.pack(pady=5)

        #TOMBOL PANGGIL LOKET B
        self.panggil_b_button = tk.Button(frame_b, text="Panggil
Antrian Loker B", activebackground="red", background="yellow",
command=self.panggil_berikutnya_b)
        self.panggil_b_button.pack(pady=5)

        # TAMBAH ANTRIAN A
        def tambah_antrian_a(self):
            nama = self.nama_entry.get()
            if nama in queue_a or nama in queue_b:
                messagebox.showerror("Error", f>Nama {nama} sudah ada dalam
antrian di salah satu loket.")
                return
            self.tambah_antrian(queue_a, self.porsi_entry_a,
self.pesanan_label_a)

        # TAMBAH ANTRIAN B
        def tambah_antrian_b(self):
            nama = self.nama_entry.get()
            if nama in queue_a or nama in queue_b:
                messagebox.showerror("Error", f>Nama {nama} sudah ada dalam
antrian di salah satu loket.")
                return
            self.tambah_antrian(queue_b, self.porsi_entry_b,
self.pesanan_label_b)

        def tambah_antrian(self, queue, porsi_entry, pesanan_label):
            nama = self.nama_entry.get()
            porsi = porsi_entry.get()
            if nama != "" and porsi != "":
                try:
                    porsi = int(porsi)
                    if len(queue) < porsi:
                        queue.append(nama)

```

```

        pesanan_label.insert("", "end",
values=(str(len(queue)), nama))
        self.nama_entry.delete(0, tk.END)
    else:
        messagebox.showinfo("Info", "Antrian penuh")
    except ValueError:
        messagebox.showerror("Error", "Porsi harus angka.")
    else:
        messagebox.showerror("Error", "Nama dan jumlah paket
sembako harus diisi.")

#PANGGIL ANTRIAN A
def panggil_berikutnya_a(self):
    if queue_a:
        # Panggil antrean pertama
        self.pemanggilan_label.config(text="Memanggil Untuk
Mengambil Sembako Di Loker A : " + queue_a.popleft())
        self.update_tabel_a()
    else:
        messagebox.showinfo("Info", "Antrian sudah kosong")

#PANGGIL ANTRIAN B
def panggil_berikutnya_b(self):
    if queue_b:
        # Panggil antrean pertama
        self.pemanggilan_label.config(text="Memanggil Untuk
Mengambil Sembako Di Loker B : " + queue_b.popleft())
        self.update_tabel_b()
    else:
        messagebox.showinfo("Info", "Antrian sudah kosong")

def update_tabel_a(self):
    # Hapus semua data di tabel
    for item in self.pesanan_label_a.get_children():
        self.pesanan_label_a.delete(item)
    # Masukkan data antrean yang tersisa
    for idx, nama in enumerate(queue_a, start=1):
        self.pesanan_label_a.insert("", "end", values=(str(idx),
nama))

def update_tabel_b(self):
    # Hapus semua data di tabel
    for item in self.pesanan_label_b.get_children():
        self.pesanan_label_b.delete(item)
    # Masukkan data antrean yang tersisa
    for idx, nama in enumerate(queue_b, start=1):

```

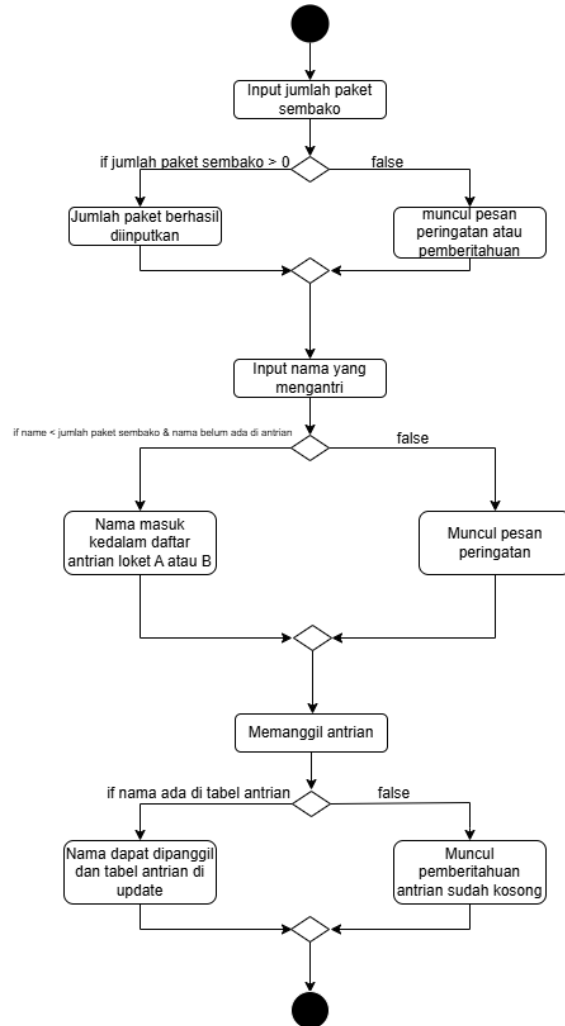
```

        self.pesanan_label_b.insert("", "end", values=(str(idx),
nama))

if __name__ == '__main__':
    root = tk.Tk()
    app = antrian(root)
    root.mainloop()

```

ACTIVITY DIAGRAM



IMPLEMENTASI

1. TAMPILAN GUI

Sistem Antrian Sembako

Antrian Sembako Kabupaten Sleman

Jumlah Paket Sembako di Loker A:

Jumlah Paket Sembako di Loker B:

Nama Warga :

Daftar Antrian di Loker A

No Antrian	Nama
------------	------

Daftar Antrian Di Loker B

No Antrian	Nama
------------	------

Memanggil:

2. PESAN ERROR APABILA INPUTAN KOSONG

Sistem Antrian Sembako

Antrian Sembako Kabupaten Sleman

Jumlah Paket Sembako di Loker A:

Jumlah Paket Sembako di Loker B:

Nama Warga :

Daftar Antrian di Loker A

No Antrian	Nama
------------	------

Daftar Antrian Di Loker B

No Antrian	Nama
------------	------

Memanggil:

Error

 Nama dan jumlah paket sembako harus diisi.

3. PESAN ERROR JIKA INPUTAN JUMLAH PAKET SEMBAKO HARUS ANGKA

Sistem Antrian Sembako

Antrian Sembako Kabupaten Sleman

Jumlah Paket Sembako di Loker A:

Jumlah Paket Sembako di Loker B:

Nama Warga :

Daftar Antrian di Loker A

No Antrian	Nama
------------	------

Daftar Antrian di Loker B

No Antrian	Nama
------------	------

Memanggil:

Error
Porsi harus angka.

4. PESAN ERROR JIKA NAMA SUDAH ADA DI TABEL ANTRIAN

Sistem Antrian Sembako

Antrian Sembako Kabupaten Sleman

Jumlah Paket Sembako di Loker A:

Jumlah Paket Sembako di Loker B:

Nama Warga :

Daftar Antrian di Loker A

No Antrian	Nama
1	jo

Daftar Antrian di Loker B

No Antrian	Nama
------------	------

Memanggil:

Error
Nama jo sudah ada dalam antrian di salah satu loket.

5. TAMPILAN ISI TABEL ANTRIAN

Sistem Antrian Sembako

Antrian Sembako Kabupaten Sleman

Jumlah Paket Sembako di Loker A:

2

Jumlah Paket Sembako di Loker B:

2

Nama Warga :

Tambah Ke Antrian Loker A Tambah Ke Antrian Loker B

Daftar Antrian di Loker A

No Antrian	Nama
1	jo
2	JE

Panggil Antrian Loker A

Daftar Antrian Di Loker B

No Antrian	Nama
1	WO
2	WE

Panggil Antrian Loker B

Memanggil:

6. PESAN INFO JIKA NAMA MELEBIHI JUMLAH PAKET SEMBAKO

Sistem Antrian Sembako

Antrian Sembako Kabupaten Sleman

Jumlah Paket Sembako di Loker A:

2

Jumlah Paket Sembako di Loker B:

2

Nama Warga :

RI

Tambah Ke Antrian Loker A Tambah Ke Antrian Loker B

Daftar Antrian di Loker A

No Antrian	Nama
1	jo
2	JE

Panggil Antrian Loker A

Daftar Antrian Di Loker B

No Antrian	Nama
1	
2	

Panggil Antrian Loker B

Memanggil:

Info

Antrian penuh

OK

7. PEMANGGILAN ANTRIAN

Sistem Antrian Sembako

Antrian Sembako Kabupaten Sleman

Jumlah Paket Sembako di Loker A:
2

Jumlah Paket Sembako di Loker B:
2

Nama Warga :
RI

Tambah Ke Antrian Loker A Tambah Ke Antrian Loker B

Daftar Antrian di Loker A

No Antrian	Nama
1	JE

Panggil Antrian Loker A

Daftar Antrian Di Loker B

No Antrian	Nama
1	WO
2	WE

Panggil Antrian Loker B

Memanggil Untuk Mengambil Sembako Di Loker A : jo

8. PESAN INFO JIKA ANTRIAN SUDAH KOSONG

Sistem Antrian Sembako

Antrian Sembako Kabupaten Sleman

Jumlah Paket Sembako di Loker A:
2

Jumlah Paket Sembako di Loker B:
2

Nama Warga :
RI

Tambah Ke Antrian Loker A Tambah Ke Antrian Loker B

Daftar Antrian di Loker A

No Antrian	Nama
------------	------

Panggil Antrian Loker A

Daftar Antrian Di Loker B

No Antrian	Nama
------------	------

Panggil Antrian Loker B

Info

Antrian sudah kosong

OK

Memanggil Untuk Mengambil Sembako Di Loker B : WE