**My TA GUI project**

buat sebuah program dengan bahasa python dan berbasis GUI. Program ini harus mengandung pengkondisian, perulangan, function method, OOP, stack and queue serta yang paling penting GUI. Program ini nantinya berjudul random food recipe picker by Nisa. Ini adalah program pemilih resep makanan. Buat halaman pertama adalah halaman selamat datang dengan gambar, judul dan tombol mulai. Lalu setelah tombol mulai di klik, munculkan halaman login. halaman login berisi username, password, checkbox syarat ketentuan. Kondisikan bahwa username dan password harus diisi dan checkbox harus dicentang. Setelah login, beralih ke halaman yang menampilkan judul 'mau masak apa hari ini?' Selanjutnya di bawah judul itu ada 3 pilihan jenis makanan dengan gambar dan keterangan Lauk, sayuran dan camilan. Setelah kita memilih salah satu, akan muncul resep makanan random yang ditampilkan beserta gambarnya

|  |
| --- |
| import tkinter as tk  from tkinter import messagebox  from random import choice  class RecipeApp:  def \_\_init\_\_(self, root):  self.root = root  self.root.title("Random Food Recipe Picker by Nisa")  self.root.geometry("400x500")  # Stack and Queue for recipe lists  self.recipe\_stack = {  "Lauk": ["Ayam Goreng", "Ikan Bakar", "Rendang"],  "Sayuran": ["Tumis Kangkung", "Capcay", "Sayur Asem"],  "Camilan": ["Bakwan", "Cireng", "Martabak Mini"]  }  self.recipe\_queue = {  "Lauk": [],  "Sayuran": [],  "Camilan": []  }  self.welcome\_page()  def welcome\_page(self):  # Welcome page UI  for widget in self.root.winfo\_children():  widget.destroy()    welcome\_frame = tk.Frame(self.root)  welcome\_frame.pack(pady=50)  welcome\_label = tk.Label(welcome\_frame, text="Selamat Datang di Random Food Recipe Picker", font=("Arial", 16))  welcome\_label.pack(pady=20)  start\_button = tk.Button(welcome\_frame, text="Mulai", font=("Arial", 14), command=self.login\_page)  start\_button.pack(pady=10)  def login\_page(self):  # Login page UI  for widget in self.root.winfo\_children():  widget.destroy()    login\_frame = tk.Frame(self.root)  login\_frame.pack(pady=50)  tk.Label(login\_frame, text="Login", font=("Arial", 16)).pack(pady=20)  tk.Label(login\_frame, text="Username:").pack()  self.username\_entry = tk.Entry(login\_frame)  self.username\_entry.pack()  tk.Label(login\_frame, text="Password:").pack()  self.password\_entry = tk.Entry(login\_frame, show="\*")  self.password\_entry.pack()  self.terms\_var = tk.IntVar()  terms\_checkbox = tk.Checkbutton(login\_frame, text="Saya menyetujui syarat & ketentuan", variable=self.terms\_var)  terms\_checkbox.pack(pady=10)  login\_button = tk.Button(login\_frame, text="Login", command=self.validate\_login)  login\_button.pack(pady=10)  def validate\_login(self):  # Login validation  username = self.username\_entry.get()  password = self.password\_entry.get()  if not username or not password:  messagebox.showwarning("Error", "Username dan Password harus diisi")  elif not self.terms\_var.get():  messagebox.showwarning("Error", "Anda harus menyetujui syarat & ketentuan")  else:  self.recipe\_category\_page()  def recipe\_category\_page(self):  # Recipe category selection page  for widget in self.root.winfo\_children():  widget.destroy()    category\_frame = tk.Frame(self.root)  category\_frame.pack(pady=50)  tk.Label(category\_frame, text="Mau masak apa hari ini?", font=("Arial", 16)).pack(pady=20)  lauk\_button = tk.Button(category\_frame, text="Lauk", font=("Arial", 14), command=lambda: self.show\_recipe("Lauk"))  lauk\_button.pack(pady=10)  sayuran\_button = tk.Button(category\_frame, text="Sayuran", font=("Arial", 14), command=lambda: self.show\_recipe("Sayuran"))  sayuran\_button.pack(pady=10)  camilan\_button = tk.Button(category\_frame, text="Camilan", font=("Arial", 14), command=lambda: self.show\_recipe("Camilan"))  camilan\_button.pack(pady=10)  def show\_recipe(self, category):  # Display random recipe based on category  recipe\_frame = tk.Frame(self.root)  recipe\_frame.pack(pady=50)    # Use stack to get a random recipe  if not self.recipe\_queue[category]:  self.recipe\_queue[category] = list(self.recipe\_stack[category])  recipe = choice(self.recipe\_queue[category])  self.recipe\_queue[category].remove(recipe)  for widget in self.root.winfo\_children():  widget.destroy()  tk.Label(recipe\_frame, text=f"Resep untuk {category}", font=("Arial", 16)).pack(pady=20)  tk.Label(recipe\_frame, text=recipe, font=("Arial", 14)).pack(pady=10)  back\_button = tk.Button(recipe\_frame, text="Kembali", command=self.recipe\_category\_page)  back\_button.pack(pady=20)  # Run the application  root = tk.Tk()  app = RecipeApp(root)  root.mainloop() |

**Penjelasan:**

1. **Halaman Selamat Datang** - Menampilkan pesan selamat datang dengan tombol "Mulai" untuk memulai aplikasi.
2. **Halaman Login** - Meminta username, password, dan konfirmasi checkbox untuk melanjutkan. Bila input belum lengkap, pesan peringatan akan muncul.
3. **Halaman Kategori Resep** - Pengguna memilih jenis makanan (Lauk, Sayuran, atau Camilan).
4. **Menampilkan Resep Acak** - Berdasarkan kategori yang dipilih, resep acak akan ditampilkan dari recipe\_stack dengan bantuan stack dan queue, serta menghindari pengulangan sampai habis.

apakah program ini mengandung pengkondisian, perulangan, function method, OOP, stack adn queue dan GUI?

**ChatGPT bilang:**

ChatGPT

Ya, program ini telah mencakup semua aspek yang Anda minta. Berikut adalah rincian dari masing-masing komponen tersebut dalam program:

1. **Pengkondisian**:
   * Terdapat pada fungsi validate\_login, di mana ada pengkondisian untuk memeriksa apakah input username, password sudah diisi dan checkbox syarat & ketentuan sudah dicentang. Jika salah satu dari syarat ini tidak terpenuhi, pesan peringatan ditampilkan.
   * Juga ada pengkondisian pada bagian show\_recipe, yaitu memeriksa apakah recipe\_queue kosong, dan jika kosong, data dari recipe\_stack ditambahkan kembali ke dalam recipe\_queue.
2. **Perulangan**:
   * Perulangan terjadi saat program menggunakan stack dan queue untuk menambahkan atau mengurangi resep ke/dari recipe\_queue.
   * Saat show\_recipe menampilkan pilihan resep, data di-stack ke dalam list yang kemudian diambil secara acak dan dihapus (untuk mencegah pengulangan sampai pilihan habis).
3. **Function/Method**:
   * Setiap fungsi, seperti welcome\_page, login\_page, validate\_login, recipe\_category\_page, dan show\_recipe, merupakan method dalam kelas RecipeApp.
   * Fungsi ini menangani bagian yang berbeda dari antarmuka dan logika aplikasi secara modular.
4. **Object-Oriented Programming (OOP)**:
   * Program ini menggunakan pendekatan OOP dengan mendefinisikan kelas RecipeApp yang berfungsi sebagai kerangka aplikasi.
   * self.root digunakan untuk mengatur elemen GUI tkinter dan self.recipe\_stack serta self.recipe\_queue digunakan untuk mengelola data resep dengan pendekatan OOP.
5. **Stack dan Queue**:
   * self.recipe\_stack (stack) menyimpan daftar resep dalam format dictionary.
   * self.recipe\_queue (queue) diisi dari self.recipe\_stack secara berulang ketika elemen recipe\_queue kosong, memastikan tidak ada pengulangan resep sampai semua resep dalam kategori tersebut telah ditampilkan.
6. **GUI**:
   * tkinter digunakan untuk membangun antarmuka grafis dengan beberapa halaman: halaman selamat datang, halaman login, halaman pilihan kategori resep, dan halaman resep.
   * Setiap halaman dibuat dengan menggunakan elemen GUI seperti Label, Button, Checkbutton, dan Entry.

Jadi, program ini sudah memenuhi semua kriteria yang diminta.

|  |
| --- |
| def recipe\_category\_page(self):          # Recipe category selection page          for widget in self.root.winfo\_children():              widget.destroy()            category\_frame = tk.Frame(self.root, bg='wheat')          category\_frame.pack(pady=50)          tk.Label(category\_frame, text="Mau masak apa hari ini?", font=("Century", 30)).pack(pady=20)          lauk\_button = tk.Button(category\_frame, text="Lauk", font=("Century", 14), command=lambda: self.show\_recipe("Lauk"))          lauk\_button.pack(pady=10)          sayuran\_button = tk.Button(category\_frame, text="Sayuran", font=("Arial", 14), command=lambda: self.show\_recipe("Sayuran"))          sayuran\_button.pack(pady=10)          camilan\_button = tk.Button(category\_frame, text="Camilan", font=("Arial", 14), command=lambda: self.show\_recipe("Camilan"))          camilan\_button.pack(pady=10)      def show\_recipe(self, category):          # Display random recipe based on category          recipe\_frame = tk.Frame(self.root)          recipe\_frame.pack(pady=50)            # Use stack to get a random recipe          if not self.recipe\_queue[category]:              self.recipe\_queue[category] = list(self.recipe\_stack[category])          recipe = choice(self.recipe\_queue[category])          self.recipe\_queue[category].remove(recipe)          for widget in self.root.winfo\_children():              widget.destroy()          tk.Label(recipe\_frame, text=f"Resep untuk {category}", font=("Arial", 16)).pack(pady=20)          tk.Label(recipe\_frame, text=recipe, font=("Arial", 14)).pack(pady=10)          back\_button = tk.Button(recipe\_frame, text="Kembali", command=self.recipe\_category\_page)          back\_button.pack(pady=20)  # Run the application  root = tk.Tk()  app = RecipeApp(root)  root.mainloop() |

|  |
| --- |
| import tkinter as tk  from tkinter import messagebox  from random import choice  from PIL import Image, ImageTk  class RecipeApp:      def \_\_init\_\_(self, root):          self.root = root          self.root.title("Random Food Recipe by Nisa")          self.root.geometry("900x600")          self.root.configure(bg='salmon')            # Stack and Queue for recipe lists          self.recipe\_stack = {              "Lauk pauk": ["Ayam Goreng", "Ikan Bakar", "Rendang"],              "Sayuran": ["Tumis Kangkung", "Capcay", "Sayur Asem"],              "Camilan": ["Bakwan", "Cireng", "Martabak Mini"],              "Nabati": ["Tempe Goreng", "Tahu Isi"],              "Hewani": ["Ayam Bakar", "Ikan Goreng"],              "Berkuah": ["Sayur Sop", "Sayur bayam"],              "Bersantan": ["Bobor kangkung", "Sayur Lodeh"],              "Tumis": ["Tumis Kangkung", "Tumis Buncis"],              "Manis": ["Kue Cubit", "Donat kentang"],              "Asin": ["Cireng Isi", "Risol Mayo"]          }          self.recipe\_queue = {              "Lauk": [],              "Sayuran": [],              "Camilan": [],              "Nabati": [],              "Hewani": [],              "Berkuah": [],              "Bersantan": [],              "Tumis": [],              "Manis": [],              "Asin": []          }          self.welcome\_page()      def welcome\_page(self):          # Welcome page UI          for widget in self.root.winfo\_children():              widget.destroy()            welcome\_frame = tk.Frame(self.root, bg='wheat')          welcome\_frame.pack(pady=50)          welcome\_label = tk.Label(welcome\_frame, text="Selamat Datang", font=("Century", 25), bg='ivory')          welcome\_label.pack(pady=20)            welcome\_label = tk.Label(welcome\_frame, text="Random Food Recipe Picker by Nisa", font=("Century", 20), bg='ivory')          welcome\_label.pack(pady=20)          start\_button = tk.Button(welcome\_frame, text="Mulai", font=("Century", 30), bg='ivory', command=self.login\_page)          start\_button.pack(pady=10)          try:              image = Image.open("A.jpg")              image = image.resize((400, 250))              self.welcome\_photo = ImageTk.PhotoImage(image)              label\_image = tk.Label(welcome\_frame, image=self.welcome\_photo)              label\_image.pack(pady=10)          except FileNotFoundError:              messagebox.showerror("Error", "Image 'A.jpg' not found.")      def login\_page(self):          # Login page UI          for widget in self.root.winfo\_children():              widget.destroy()          welcome\_frame = tk.Frame(self.root, bg='wheat')          welcome\_frame.pack(pady=50)          welcome\_label = tk.Label(welcome\_frame, text="Sebelum lanjut, Login dulu yuk", font=("Century", 25), bg='ivory')          welcome\_label.pack(pady=20)            login\_frame = tk.Frame(self.root, bg='wheat')          login\_frame.pack(pady=10)          tk.Label(login\_frame, text="Login", font=("Century", 40), bg='ivory').grid(row=0, column=0, columnspan=2, pady=20)          # Username Label dan Box disamping          tk.Label(login\_frame, text="Username:", font=("Century", 20), bg='ivory').grid(row=1, column=0, sticky="e", padx=10, pady=10)          self.username\_entry = tk.Entry(login\_frame, font=("Century", 20))          self.username\_entry.grid(row=1, column=1, padx=10, pady=10)          tk.Label(login\_frame, text="Password:", font=("Century", 20), bg='ivory').grid(row=2, column=0, sticky="e", padx=10, pady=10)          self.password\_entry = tk.Entry(login\_frame, show="\*", font=("Century", 20))          self.password\_entry.grid(row=2, column=1, padx=10, pady=10)          self.terms\_var = tk.IntVar()          terms\_checkbox = tk.Checkbutton(login\_frame, text="Saya menyetujui syarat & ketentuan", bg='ivory', variable=self.terms\_var)          terms\_checkbox.grid(row=3, column=0, columnspan=2, pady=10)          login\_button = tk.Button(login\_frame, text="Login", font=("Century", 20), bg='ivory', command=self.validate\_login)          login\_button.grid(row=4, column=0, columnspan=2, pady=10)      def validate\_login(self):          # Login validation          username = self.username\_entry.get()          password = self.password\_entry.get()          if not username or not password:              messagebox.showwarning("Error", "Username dan Password harus diisi")          elif not self.terms\_var.get():              messagebox.showwarning("Error", "Anda harus menyetujui syarat & ketentuan")          else:              self.recipe\_category\_page()      def recipe\_category\_page(self):          # Recipe category selection page          for widget in self.root.winfo\_children():              widget.destroy()          category\_frame = tk.Frame(self.root, bg='salmon')          category\_frame.pack(pady=50)          tk.Label(category\_frame, text="Pilih Kategori Resep", font=("Century", 30), bg='ivory').pack(pady=60)          images = {}          frames = tk.Frame(category\_frame, bg='Salmon')          frames.pack()          for category, img\_file in zip(["Lauk pauk", "Sayuran", "Camilan"], ["Lauk.png", "Sayurr.jpg", "camilan.jpg"]):              try:                  img = Image.open(img\_file).resize((200, 150))                  images[category] = ImageTk.PhotoImage(img)              except FileNotFoundError:                  messagebox.showerror("Error", f"Image '{img\_file}' not found.")                  return          for idx, category in enumerate(["Lauk pauk", "Sayuran", "Camilan"]):              frame = tk.Frame(frames, bg='salmon')              frame.pack(side=tk.LEFT, padx=20)              img\_label = tk.Label(frame, image=images[category],bg='ivory')              img\_label.image = images[category]              img\_label.pack()              button = tk.Button(frame, text=category, bg='ivory',font=("Century", 20), command=lambda c=category: self.show\_subcategories(c))              button.pack(pady=10)      def show\_subcategories(self, category):          # Show subcategories for the selected category          for widget in self.root.winfo\_children():              widget.destroy()          subcategory\_frame = tk.Frame(self.root, bg='salmon')          subcategory\_frame.pack(pady=60)          tk.Label(subcategory\_frame, text=f"Pilih Subkategori {category}", font=("Century", 30), bg='ivory').pack(pady=60)          images = {}          subcategories = []          if category == "Lauk pauk":              subcategories = [("Nabati", "nabati.jpg"), ("Hewani", "hewani.jpg.webp")]          elif category == "Sayuran":              subcategories = [("Berkuah", "berkuahh.jpg"), ("Bersantan", "bersantan.jpg"), ("Tumis", "tumis.jpg")]          elif category == "Camilan":              subcategories = [("Manis", "manis.jpg"), ("Asin", "asin.jpg")]          frames = tk.Frame(subcategory\_frame, bg='salmon')          frames.pack()          for subcategory, img\_file in subcategories:              try:                  img = Image.open(img\_file).resize((180, 150))                  images[subcategory] = ImageTk.PhotoImage(img)              except FileNotFoundError:                  messagebox.showerror("Error", f"Image '{img\_file}' not found.")                  return          for subcategory, img in images.items():              frame = tk.Frame(frames, bg='salmon')              frame.pack(side=tk.LEFT, padx=20)              img\_label = tk.Label(frame, image=img, bg='ivory')              img\_label.image = img              img\_label.pack()              button = tk.Button(frame, text=subcategory, bg='ivory',  font=("Century", 20), command=lambda s=subcategory: self.show\_recipe(s))              button.pack(pady=10)          # Add back button to go back to category page          back\_button = tk.Button(subcategory\_frame, text="Kembali", font=("Century", 20),  bg='ivory', command=self.recipe\_category\_page)          back\_button.pack(pady=10)      def show\_recipe(self, subcategory):          # Display a random recipe from the selected subcategory          recipe = choice(self.recipe\_stack[subcategory])          messagebox.showinfo("Resep Pilihan", f"Resep untuk subkategori {subcategory}: {recipe}")  # Inisialisasi aplikasi  root = tk.Tk()  app = RecipeApp(root)  root.mainloop() |

|  |
| --- |
| import tkinter as tk  from tkinter import messagebox  from random import choice  from PIL import Image, ImageTk  class RecipeApp:      def \_\_init\_\_(self, root):          self.root = root          self.root.title("Random Food Recipe by Nisa")          self.root.geometry("900x600")          self.root.configure(bg='salmon')          # Stack for recipes          self.recipe\_stack = {              "Lauk pauk": {                  "Nabati": [                      {"name": "Tempe Goreng", "details": "200g tempe, 2 siung bawang putih, 1 sdt ketumbar, sejumput garam, minyak goreng."},                      {"name": "Tahu Goreng", "details": "3 buah tahu putih, 1 siung bawang putih, 1 sdt kunyit bubuk, garam secukupnya, minyak goreng."}                  ],                  "Hewani": [                      {"name": "Ayam Goreng", "details": "1 ekor ayam, 3 siung bawang putih, 2 sdm ketumbar, garam secukupnya, minyak goreng."},                      {"name": "Ikan Bakar", "details": "1 ekor ikan kembung, 2 sdm kecap manis, 1 buah jeruk nipis, 2 siung bawang merah, 3 buah cabai."}                  ]              },              "Sayuran": {                  "Berkuah": [                      {"name": "Sayur Asem", "details": "Melinjo, kacang panjang, jagung, asam jawa, bawang merah, cabai, garam."},                      {"name": "Sop Ayam", "details": "250g ayam, wortel, kentang, daun bawang, bawang putih, bawang merah, sejumput lada."}                  ],                  "Bersantan": [                      {"name": "Lodeh Terong", "details": "1 buah terong, santan cair, 2 siung bawang putih, 3 siung bawang merah, cabai rawit."},                      {"name": "Sayur Nangka", "details": "300g nangka muda, santan kental, daun salam, lengkuas, bawang merah, cabai merah."}                  ],                  "Tumis": [                      {"name": "Tumis Kangkung", "details": "1 ikat kangkung, 2 siung bawang putih, 2 siung bawang merah, sejumput garam."},                      {"name": "Tumis Buncis", "details": "200g buncis, 2 siung bawang putih, 1 sdm saus tiram, cabai merah."}                  ]              },              "Camilan": {                  "Manis": [                      {"name": "Bolu Kukus", "details": "150g tepung terigu, 100g gula pasir, 2 butir telur, 100ml susu cair, 1 sdt baking powder."},                      {"name": "Kue Putri Salju", "details": "100g tepung terigu, 50g gula halus, 100g mentega, 50g keju parut."}                  ],                  "Asin": [                      {"name": "Keripik Tempe", "details": "200g tempe iris tipis, 100g tepung beras, 1 siung bawang putih, garam, air."},                      {"name": "Pastel Isi", "details": "Kulit pastel, bihun, wortel parut, ayam cincang, bawang merah, bawang putih."}                  ]              }          }          self.welcome\_page()      def welcome\_page(self):          # Welcome page UI          for widget in self.root.winfo\_children():              widget.destroy()            welcome\_frame = tk.Frame(self.root, bg='wheat')          welcome\_frame.pack(pady=50)          welcome\_label = tk.Label(welcome\_frame, text="Selamat Datang", font=("Century", 25), bg='ivory')          welcome\_label.pack(pady=20)            welcome\_label = tk.Label(welcome\_frame, text="Random Food Recipe Picker by Nisa", font=("Century", 20), bg='ivory')          welcome\_label.pack(pady=20)          start\_button = tk.Button(welcome\_frame, text="Mulai", font=("Century", 30), bg='ivory', command=self.login\_page)          start\_button.pack(pady=10)          try:              image = Image.open("A.jpg")              image = image.resize((400, 250))              self.welcome\_photo = ImageTk.PhotoImage(image)              label\_image = tk.Label(welcome\_frame, image=self.welcome\_photo)              label\_image.pack(pady=10)          except FileNotFoundError:              messagebox.showerror("Error", "Image 'A.jpg' not found.")      def login\_page(self):          # Login page UI          for widget in self.root.winfo\_children():              widget.destroy()          welcome\_frame = tk.Frame(self.root, bg='wheat')          welcome\_frame.pack(pady=50)          welcome\_label = tk.Label(welcome\_frame, text="Sebelum lanjut, Login dulu yuk", font=("Century", 25), bg='ivory')          welcome\_label.pack(pady=20)            login\_frame = tk.Frame(self.root, bg='wheat')          login\_frame.pack(pady=10)          tk.Label(login\_frame, text="Login", font=("Century", 40), bg='ivory').grid(row=0, column=0, columnspan=2, pady=20)          # Username Label dan Box disamping          tk.Label(login\_frame, text="Username:", font=("Century", 20), bg='ivory').grid(row=1, column=0, sticky="e", padx=10, pady=10)          self.username\_entry = tk.Entry(login\_frame, font=("Century", 20))          self.username\_entry.grid(row=1, column=1, padx=10, pady=10)          tk.Label(login\_frame, text="Password:", font=("Century", 20), bg='ivory').grid(row=2, column=0, sticky="e", padx=10, pady=10)          self.password\_entry = tk.Entry(login\_frame, show="\*", font=("Century", 20))          self.password\_entry.grid(row=2, column=1, padx=10, pady=10)          self.terms\_var = tk.IntVar()          terms\_checkbox = tk.Checkbutton(login\_frame, text="Saya menyetujui syarat & ketentuan", bg='ivory', variable=self.terms\_var)          terms\_checkbox.grid(row=3, column=0, columnspan=2, pady=10)          login\_button = tk.Button(login\_frame, text="Login", font=("Century", 20), bg='ivory', command=self.validate\_login)          login\_button.grid(row=4, column=0, columnspan=2, pady=10)      def validate\_login(self):          # Login validation          username = self.username\_entry.get()          password = self.password\_entry.get()          if not username or not password:              messagebox.showwarning("Error", "Username dan Password harus diisi")          elif not self.terms\_var.get():              messagebox.showwarning("Error", "Anda harus menyetujui syarat & ketentuan")          else:              self.recipe\_category\_page()      def recipe\_category\_page(self):          # Recipe category selection page          for widget in self.root.winfo\_children():              widget.destroy()          category\_frame = tk.Frame(self.root, bg='salmon')          category\_frame.pack(pady=50)          tk.Label(category\_frame, text="Pilih Kategori Resep", font=("Century", 30), bg='ivory').pack(pady=60)          images = {}          frames = tk.Frame(category\_frame, bg='Salmon')          frames.pack()          for category, img\_file in zip(["Lauk pauk", "Sayuran", "Camilan"], ["Lauk.png", "Sayurr.jpg", "camilan.jpg"]):              try:                  img = Image.open(img\_file).resize((200, 150))                  images[category] = ImageTk.PhotoImage(img)              except FileNotFoundError:                  messagebox.showerror("Error", f"Image '{img\_file}' not found.")                  return          for idx, category in enumerate(["Lauk pauk", "Sayuran", "Camilan"]):              frame = tk.Frame(frames, bg='salmon')              frame.pack(side=tk.LEFT, padx=20)              img\_label = tk.Label(frame, image=images[category], bg='ivory')              img\_label.image = images[category]              img\_label.pack()              button = tk.Button(frame, text=category, bg='ivory', font=("Century", 20), command=lambda c=category: self.show\_subcategories(c))              button.pack(pady=10)      def show\_subcategories(self, category):          # Show subcategories for the selected category          for widget in self.root.winfo\_children():              widget.destroy()          subcategory\_frame = tk.Frame(self.root, bg='salmon')          subcategory\_frame.pack(pady=60)          tk.Label(subcategory\_frame, text=f"Pilih Subkategori {category}", font=("Century", 30), bg='ivory').pack(pady=60)          images = {}          subcategories = []          if category == "Lauk pauk":              subcategories = [("Nabati", "nabati.jpg"), ("Hewani", "hewani.jpg.webp")]          elif category == "Sayuran":              subcategories = [("Berkuah", "berkuahh.jpg"), ("Bersantan", "bersantan.jpg"), ("Tumis", "tumis.jpg")]          elif category == "Camilan":              subcategories = [("Manis", "manis.jpg"), ("Asin", "asin.jpg")]          frames = tk.Frame(subcategory\_frame, bg='salmon')          frames.pack()          for subcategory, img\_file in subcategories:              try:                  img = Image.open(img\_file).resize((180, 150))                  images[subcategory] = ImageTk.PhotoImage(img)              except FileNotFoundError:                  messagebox.showerror("Error", f"Image '{img\_file}' not found.")                  return          for subcategory, img in images.items():              frame = tk.Frame(frames, bg='salmon')              frame.pack(side=tk.LEFT, padx=20)              img\_label = tk.Label(frame, image=img, bg='ivory')              img\_label.image = img              img\_label.pack()              button = tk.Button(frame, text=subcategory, bg='ivory', font=("Century", 20), command=lambda s=subcategory: self.show\_recipe(category, s))              button.pack(pady=10)          # Add back button to go back to category page          back\_button = tk.Button(subcategory\_frame, text="Kembali", font=("Century", 20), bg='ivory', command=self.recipe\_category\_page)          back\_button.pack(pady=10)      def show\_recipe(self, category, subcategory):          # Display a random recipe from the selected subcategory          recipes = self.recipe\_stack[category][subcategory]          recipe = choice(recipes)          recipe\_text = f"Resep {recipe['name']}\n\n{recipe['details']}"          for widget in self.root.winfo\_children():              widget.destroy()          recipe\_frame = tk.Frame(self.root, bg='salmon')          recipe\_frame.pack(pady=60)          tk.Label(recipe\_frame, text=f"Resep {subcategory}", font=("Century", 30), bg='ivory').pack(pady=30)            recipe\_label = tk.Label(recipe\_frame, text=recipe\_text, font=("Century", 20), bg='wheat', justify="left")          recipe\_label.pack(pady=20)          back\_button = tk.Button(recipe\_frame, text="Kembali", font=("Century", 20), bg='ivory', command=self.recipe\_category\_page)          back\_button.pack(pady=10)  # Inisialisasi aplikasi  root = tk.Tk()  app = RecipeApp(root)  root.mainloop() |

|  |
| --- |
| import tkinter as tk  from tkinter import messagebox  from random import choice  from PIL import Image, ImageTk  class RecipeApp:      def \_\_init\_\_(self, root):          self.root = root          self.root.title("Random Food Recipe by Nisa")          self.root.geometry("900x600")          self.root.configure(bg='salmon')          # Stack for recipes          self.recipe\_stack = {              "Lauk pauk": {                  "Nabati": [                      {"name": "Tempe Goreng",                       "details": [                           "1 bungkus tempe",                           "1 siung bawang putih",                           "1/2 sdt ketumbar",                           "Tepung bumbu",                           "Garam secukupnya",                           "Minyak goreng."                       ]},                      {"name": "Tahu Goreng",                       "details": [                           "4 buah tahu putih",                           "1 siung bawang putih",                           "1 sdt kunyit bubuk",                           "Garam secukupnya",                           "Minyak goreng."                       ]},                       {"name": "Tempe Mendoan",                       "details": [                           "1 bungkus tempe",                           "2 siung bawang putih",                           "1/2 sdt ketumbar",                           "2 sdm tepung bumbu",                           "5 sdm Tepung terigu",                           "3 batang daun bawang",                           "Sejumput garam",                           "Minyak goreng."                       ]},                       {"name": "Tahu Kecap",                       "details": [                           "5 buah tahu",                           "2 siung bawang putih",                           "1 siung bawang merah",                           "3/4 sdt garam",                           "1 sdt gula",                           "3 buah cabe merah",                           "Kecap manis sesuai selera",                           "Minyak secukupnya."                       ]},                       {"name": "Tahu Tempe Bacem",                       "details": [                           "7 buah tahu",                           "1 bungkus tempe",                           "4 lembar daun salam",                           "1 sdt garam",                           "2 sdm kecap manis",                           "2 sdm air asam jawa",                           "2 siung bawang putih",                           "3 siung bawang merah",                           "50 gram gula jawa",                           "1 sdt ketumbar",                           "3 cm lengkuas"                       ]}                  ],                  "Hewani": [                      {"name": "Ayam Goreng",                       "details": [                           "1 ekor ayam",                           "4 siung bawang putih",                           "6 siung bawang merah"                           "2 sdm ketumbar",                           "1 sdm garam",                           "1 sdt gula pasir"                           "3 lembar daun jeruk dan salam",                           "2 batang serai",                           "2 cm kunyit",                           "Minyak goreng"                       ]},                      {"name": "Ikan Nila Bakar",                       "details": [                           "2 ekor ikan nila",                           "2 sdm kecap manis",                           "1 buah jeruk nipis",                           "3 siung bawang merah",                           "3 siung bawang putih",                           "3 buah cabai",                           "2 sdt garam",                           "1 sdm kaldu ayam bubuk",                           "1/4 sdt merica",                           "1/2 cm kunyit dan jahe",                           "minyak secukupnya."                       ]},                       {"name": "Rendang Sapi",                       "details": [                           "1 kg daging sapi",                           "Santan peras dari 3 butir kelapa",                           "4 lembar daun jeruk",                           "3 batang serai",                           "4 butir asam kandis",                           "100 ml minyak goreng",                           "bumbu rendang daging sapi 1 kg",                           "1 sdm garam",                           "6 siung bawang merah dan bawang putih",                           "10 cabe merah besar",                           "Lengkuas, cengkeh dan kemiri"                       ]},                       {"name": "Semur Telur",                       "details": [                           "6 telur ayam yang sudah direbus dan dikupas",                           "2 lembar daun salam",                           "1 serai",                           "3 sdm kecap manis",                           "1 sdm gula merah",                           "1 sdt garam",                           "150 ml santan kental",                           "300 ml santan encer",                           "6 butir bawang merah",                           "3 siung bawang putih",                           "kemiri, pala dan merica bubuk secukupnya"                       ]},                        {"name": "Udang Balado",                       "details": [                           "1 kg udang kupas",                           "1/4 kg bawang merah",                           "5 siung bawang putih",                           "1/4 kg cabe merah",                           "1 buah tomat",                           "4 buah jeruk limau",                           "Garam dan gula secukupnya",                           "1 buah jeruk nipis (untuk merendam udang)",                           "Minyak secukupnya",                           "Air secukupnya",                       ]}                  ]              },              "Sayuran": {                  "Berkuah": [                      {"name": "Sayur Asem",                       "details": [                           "Sayuran (wortel, kacang panjang, jagung manis, dll)",                           "Asam jawa secukupnya",                           "2 cm lengkuas",                           "3 lembar daun salam",                           "5 siung bawang merah",                           "2 siung bawang putih",                           "2 buah cabai merah",                           "1 sdt garam",                           "1/2 sdt gula",                           "1 liter air",                           "Penyedap rasa secukupnya."                       ]},                      {"name": "Sup Jamur Kancing",                       "details": [                           "125 gram jamur kancing",                           "100 gram dada ayam fillet (potong dadu)",                           "Sayuran(kentang, wortel, brokoli, dll)",                           "750 ml air",                           "2 siung bawang putih",                           "1 siung bawang merah",                           "1/2 bawang bombay",                           "Sejumput lada",                           "Penyedap rasa secukupnya."                       ]},                       {"name": "Sayur Bayam",                       "details": [                           "1 ikat bayam",                           "1 jagung manis",                           "2 siung bawang merah",                           "3 cm kunci",                           "700 ml air",                           "1 siung bawang merah",                           "1 sdt garam",                           "1/2 sdt gula pasir",                           "Penyedap rasa secukupnya."                       ]},                       {"name": "Sayur Sawi Putih dan Bakso",                       "details": [                           "1 bonggol sawi putih",                           "8 buah bakso, iris",                           "4 siung bawang putih",                           "Garam secukupnya",                           "Kaldu bubuk secukupnya",                           "1 sdt kecap ikan",                           "Air secukupnya",                       ]},                       {"name": "Sayur Gambas",                       "details": [                           "1 buah gambas/ oyong",                           "6 jagung muda",                           "2 bungkus kecil soun",                           "3 siung bawang merah",                           "2 siung bawang putih",                           "750 ml air",                           "1/2 sdt garam",                           "1/2 sdt gula pasir",                           "Kaldu bubuk secukupnya."                       ]},                       {"name": "Tomyam",                       "details": [                           "4 buah tahu sutra",                           "Crabstick, udang, bakso seafood secukupnya",                           "Jamur kancing dan pakcoy secukupnya",                           "2 sdm minyak ikan",                           "1 wortel",                           "4 cabai rawit",                           "1/2 sdt garam",                           "3 batang serai",                           "5 lembar daun jeruk",                           "1 sdm ebi",                           "3 siung bawang putih",                           "1,5 liter kaldu udang."                       ]}                  ],                  "Bersantan": [                      {"name": "Lodeh Kacang Terong",                       "details": [                           "1 buah terong",                           "150 gram kacang panjang",                           "1 jagung manis",                           "2 siung bawang putih",                           "3 siung bawang merah",                           "Cabai rawit",                           "1 lembar daun salam",                           "3 cm lengkuas",                           "800 ml santan",                           "1 sdt garam",                           "1/2 sdt gula pasir",                           "Penyedap rasa secukupnya."                       ]},                      {"name": "Sayur Bobor Kangkung",                       "details": [                           "1 ikat kangkung",                           "1 labu siam besar",                           "1 wortel",                           "Kemangi secukupnya",                           "1 siung bawang merah",                           "1 siung bawang putih",                           "1/2 sdt garam",                           "1 kemiri",                           "1 bungkus santan instan (65 ml)",                           "Daun salam."                       ]},                       {"name": "Gulai Nangka",                       "details": [                           "1/4 kg nangka muda",                           "1/4 kg daging sapi tetelan",                           "1/2 bungkus santan kelapa kental",                           "Serai, daun salam, daun jeruk purut",                           "sejempol lengkuas",                           "Air secukupnya",                           "1/2 sdt garam",                           "Air dan gula secukupnya",                           "3 siung bawang putih",                           "4 siung bawang merah",                           "Cabai merah secukupnya",                           "Kunyit, jahe dan kencur."                       ]},                       {"name": "Selada Air Kuah Santan",                       "details": [                           "2 ikat selada air",                           "5 butir bawang merah",                           "3 siung bawang putih",                           "2 buah cabai merah",                           "segenggam udang kering, goreng",                           "500 ml santan sedang",                           "Garam dan gula secukupnya."                       ]},                  ],                  "Tumis": [                      {"name": "Tumis Kangkung Saus Tiram",                       "details": [                           "1 ikat kangkung",                           "2 siung bawang putih",                           "2 siung bawang merah",                           "Garam secukupnya",                           "1 sdm saus tiram",                           "75 ml air",                           "2 buah otak-otak(opsional)",                           "1 sdm minyak goreng"                       ]},                      {"name": "Tumis Buncis",                       "details": [                           "250g buncis",                           "2 siung bawang putih",                           "1 sdm saus tiram",                           "2 buah cabai merah",                           "2 butir telur ayam",                           "3 butir bawang merah",                           "100 ml air",                           "Garam secukupnya"                           "Royco ayam secukupnya",                           "1 sdm kecap manis",                           "Sejumput merica",                           "1 sdm minyak goreng"                       ]},                       {"name": "Tumis Tauge",                       "details": [                           "200g tauge",                           "4 siung bawang putih",                           "1 sdm saus tiram",                           "4 buah cabai merah",                           "1/2 tomat merah",                           "6 butir bawang merah",                           "30 ml air",                           "Garam secukupnya",                           "1/2 sdt gula",                           "Sejumput merica",                           "1 sdm minyak goreng"                       ]},                  ]              },              "Camilan": {                  "Manis": [                      {"name": "Bolu Kukus Pandan",                       "details": [                           "125g tepung terigu",                           "100g gula pasir",                           "2 butir telur",                           "120ml susu cair",                           "1 sdt baking powder",                           "1 sdt emulsifier (SP)",                           "1 sdt pasta pandan"                       ]},                      {"name": "Kue Putri Salju",                       "details": [                           "100gr butter",                           "70gr gula halus",                           "50gr margarin",                           "50g keju parut",                           "Sedikit garam",                           "1 butir kuning telur",                           "2 kuning telur rebus (haluskan)",                           "175gr tepung terigu",                           "15gr tepung maizena",                           "15gr susu bubuk."                       ]},                       {"name": "Bika Ambon",                       "details": [                           "12 butir kuning telur",                           "150 gram gula pasir",                           "100 gram tepung sagu",                           "200 ml santan kental",                           "1/4 sdt garam",                           "10 lembar daun jeruk",                           "1 serai",                           "1 sdt ragi instan",                           "1 sdm gula pasir",                           "1 sdm terigu",                           "50 ml air hangat"                       ]},                       {"name": "Mochi Strawberry",                       "details": [                           "130 gr tepung ketan",                           "130 gr air",                           "65 ml santan instan",                           "2 1/2 sdm gula",                           "1/2 sdt garam",                           "2 sdm minyak goreng",                           "Pewarna makanan merah secukupnya",                           "5 sdm maizena",                           "Strawberry segar",                           "Selai strawberry",                           "Krim strawberry."                       ]},                       {"name": "Donat Kentang",                       "details": [                           "Tepung protein tinggi 250 gr",                           "Terigu 250 gr",                           "Topping sesuai selera (meises,keju,glaze,gula halus)",                           "Ragi instan 8gr",                           "Roti improver 3gr",                           "Gula 50gr",                           "Susu bubuk skim 25gr",                           "Garam 8gr",                           "Kuning telur 60gr",                           "Margarin untuk kue 75gr",                           "Air Dingin 225ml"                       ]},                  ],                  "Asin": [                      {"name": "Risol Mayo",                       "details": [                           "10 buah kulit lumpia siap pakai",                           "5 buah sosis",                           "5 lembar keju",                           "4 butir telur rebus",                           "Mayones secukupnya",                           "Mentega",                           "Minyak untuk menggoreng",                           "====Bahan Pelapis====",                           "¼ kg tepung panir",                           "Tepung terigu secukupnya",                           "½ sdt garam",                           "Air secukupnya."                       ]},                      {"name": "Cireng Ayam Suwir Pedas",                       "details": [                           "15 sdm tepung terigu",                           "15 sendok makan tepung tapioka",                           "2 sachet kaldu bubuk",                           "air panas secukupnya",                           "====Bahan Isian====",                           "2 potong dada ayam yang telah direbus",                           "10 siung bawang merah",                           "10 siung bawang putih",                           "10 buah cabai merah",                           "2 sendok teh cabai bubuk",                           "Gula dan garam secukupnya",                           "Daun bawang secukupnya"                       ]},                        {"name": "Cilor",                       "details": [                           "10 sdm tepung sagu",                           "7 sdm tepung terigu",                           "2 siung bawang putih parut",                           "1/2 sdt garam",                           "1/2 sdt kaldu bubuk",                           "Secukupnya air matang",                           "1 butir telur kocok lepas."                       ]},                       {"name": "Dimsum Ayam",                       "details": [                           "3 potong ayam paha, fillet dagingnya",                           "1 sdt garam",                           "1 sdm gula pasir",                           "1/2 sdt lada halus",                           "3/4 sdm kecap asin",                           "3/4 sdm saus tiram",                           "25 gram sagu tani",                           "1 lembar daun bawang,iris",                           "kulit pangsit rebus 15 buah",                           "====Bahan Saus====",                           "5 sachet saus sambal",                           "1 sdt gula pasir",                           "3 sdm air panas."                       ]},                  ]              }          }          self.welcome\_page()      def welcome\_page(self):          # Welcome page UI          for widget in self.root.winfo\_children():              widget.destroy()            welcome\_frame = tk.Frame(self.root, bg='wheat')          welcome\_frame.pack(pady=50)          welcome\_label = tk.Label(welcome\_frame, text="Selamat Datang", font=("Century", 25), bg='ivory')          welcome\_label.pack(pady=20)            welcome\_label = tk.Label(welcome\_frame, text="Random Food Recipe Picker by Nisa", font=("Century", 20), bg='ivory')          welcome\_label.pack(pady=20)          welcome\_label = tk.Label(welcome\_frame, text="Full screen untuk pengalaman lebih nyaman", font=("Century", 15), bg='ivory')          welcome\_label.pack(pady=20)          start\_button = tk.Button(welcome\_frame, text="Mulai", font=("Century", 30), bg='ivory', command=self.login\_page)          start\_button.pack(pady=10)          try:              image = Image.open("A.jpg")              image = image.resize((400, 250))              self.welcome\_photo = ImageTk.PhotoImage(image)              label\_image = tk.Label(welcome\_frame, image=self.welcome\_photo)              label\_image.pack(pady=10)          except FileNotFoundError:              messagebox.showerror("Error", "Image 'A.jpg' not found.")      def login\_page(self):          # Login page UI          for widget in self.root.winfo\_children():              widget.destroy()          welcome\_frame = tk.Frame(self.root, bg='wheat')          welcome\_frame.pack(pady=50)          welcome\_label = tk.Label(welcome\_frame, text="Sebelum lanjut, Login dulu yuk", font=("Century", 25), bg='ivory')          welcome\_label.pack(pady=20)            login\_frame = tk.Frame(self.root, bg='wheat')          login\_frame.pack(pady=10)          tk.Label(login\_frame, text="Login", font=("Century", 40), bg='ivory').grid(row=0, column=0, columnspan=2, pady=20)          # Username Label dan Box disamping          tk.Label(login\_frame, text="Username:", font=("Century", 20), bg='ivory').grid(row=1, column=0, sticky="e", padx=10, pady=10)          self.username\_entry = tk.Entry(login\_frame, font=("Century", 20))          self.username\_entry.grid(row=1, column=1, padx=10, pady=10)          tk.Label(login\_frame, text="Password:", font=("Century", 20), bg='ivory').grid(row=2, column=0, sticky="e", padx=10, pady=10)          self.password\_entry = tk.Entry(login\_frame, show="\*", font=("Century", 20))          self.password\_entry.grid(row=2, column=1, padx=10, pady=10)          self.terms\_var = tk.IntVar()          terms\_checkbox = tk.Checkbutton(login\_frame, text="Saya menyetujui syarat & ketentuan", bg='ivory', variable=self.terms\_var)          terms\_checkbox.grid(row=3, column=0, columnspan=2, pady=10)          login\_button = tk.Button(login\_frame, text="Login", font=("Century", 20), bg='ivory', command=self.validate\_login)          login\_button.grid(row=4, column=0, columnspan=2, pady=10)      def validate\_login(self):          # Login validation          username = self.username\_entry.get()          password = self.password\_entry.get()          if not username or not password:              messagebox.showwarning("Error", "Username dan Password harus diisi")          elif not self.terms\_var.get():              messagebox.showwarning("Error", "Anda harus menyetujui syarat & ketentuan")          else:              self.recipe\_category\_page()      def recipe\_category\_page(self):          # Recipe category selection page          for widget in self.root.winfo\_children():              widget.destroy()          category\_frame = tk.Frame(self.root, bg='salmon')          category\_frame.pack(pady=50)          tk.Label(category\_frame, text="Pilih Kategori Resep", font=("Century", 30), bg='ivory').pack(pady=70)          images = {}          frames = tk.Frame(category\_frame, bg='Salmon')          frames.pack()          for category, img\_file in zip(["Lauk pauk", "Sayuran", "Camilan"], ["Lauk.png", "Sayurr.jpg", "camilan.jpg"]):              try:                  img = Image.open(img\_file).resize((200, 150))                  images[category] = ImageTk.PhotoImage(img)              except FileNotFoundError:                  messagebox.showerror("Error", f"Image '{img\_file}' not found.")                  return          for idx, category in enumerate(["Lauk pauk", "Sayuran", "Camilan"]):              frame = tk.Frame(frames, bg='salmon')              frame.pack(side=tk.LEFT, padx=20, pady=30)              img\_label = tk.Label(frame, image=images[category], bg='ivory')              img\_label.image = images[category]              img\_label.pack()              button = tk.Button(frame, text=category, bg='ivory', font=("Century", 20), command=lambda c=category: self.show\_subcategories(c))              button.pack(pady=60)      def show\_subcategories(self, category):          # Show subcategories for the selected category          for widget in self.root.winfo\_children():              widget.destroy()          subcategory\_frame = tk.Frame(self.root, bg='salmon')          subcategory\_frame.pack(pady=60)          tk.Label(subcategory\_frame, text=f"Pilih Subkategori {category}", font=("Century", 30), bg='ivory').pack(pady=50)          images = {}          subcategories = []          if category == "Lauk pauk":              subcategories = [("Nabati", "nabati.jpg"), ("Hewani", "hewani.jpg.webp")]          elif category == "Sayuran":              subcategories = [("Berkuah", "berkuahh.jpg"), ("Bersantan", "bersantan.jpg"), ("Tumis", "tumis.jpg")]          elif category == "Camilan":              subcategories = [("Manis", "manis.jpg"), ("Asin", "asin.jpg")]          frames = tk.Frame(subcategory\_frame, bg='salmon')          frames.pack()          for subcategory, img\_file in subcategories:              try:                  img = Image.open(img\_file).resize((180, 150))                  images[subcategory] = ImageTk.PhotoImage(img)              except FileNotFoundError:                  messagebox.showerror("Error", f"Image '{img\_file}' not found.")                  return          for subcategory, img in images.items():              frame = tk.Frame(frames, bg='salmon')              frame.pack(side=tk.LEFT, padx=20, pady=40)              img\_label = tk.Label(frame, image=img, bg='ivory')              img\_label.image = img              img\_label.pack()              button = tk.Button(frame, text=subcategory, bg='ivory', font=("Century", 20), command=lambda s=subcategory: self.show\_recipe(category, s))              button.pack(pady=35)          # Add back button to go back to category page          back\_button = tk.Button(subcategory\_frame, text="Kembali", font=("Century", 20), bg='ivory', command=self.recipe\_category\_page)          back\_button.pack(pady=10)      def format\_recipe\_details(self, details):          """          Mengubah daftar bahan menjadi string multi-baris dengan tanda titik (-) di awal setiap baris.          """          return "\n".join(f"- {item}" for item in details)      def show\_recipe(self, category, subcategory):          # Display a random recipe from the selected subcategory          recipes = self.recipe\_stack[category][subcategory]          recipe = choice(recipes)          formatted\_details = self.format\_recipe\_details(recipe['details'])          recipe\_text = f"{recipe['name']}\n\nBahan yang dibutuhkan\n{formatted\_details}"          for widget in self.root.winfo\_children():              widget.destroy()          recipe\_frame = tk.Frame(self.root, bg='salmon')          recipe\_frame.pack(pady=20, padx=30, fill=tk.BOTH, expand=True)          tk.Label(recipe\_frame, text=f"Resep {subcategory}", font=("Century", 30), bg='ivory').pack(pady=10)            # Menggunakan Label dengan opsi justify dan anchor untuk perataan kiri          recipe\_label = tk.Label(recipe\_frame, text=recipe\_text, font=("Century", 20), bg='wheat', justify="left", anchor="w", padx=10)          recipe\_label.pack(pady=10, fill=tk.BOTH, expand=True)          back\_button = tk.Button(recipe\_frame, text="Kembali", font=("Century", 20), bg='ivory', command=lambda: self.show\_subcategories(category))          back\_button.pack(pady=10)  # Inisialisasi aplikasi  if \_\_name\_\_ == "\_\_main\_\_":      root = tk.Tk()      app = RecipeApp(root)      root.mainloop() |

Ya, kode ini mengandung semua elemen yang disebutkan: **pengkondisian**, **perulangan**, **function method**, **OOP**, dan **GUI**. Berikut penjelasannya:

**1. Pengkondisian**

Pengkondisian ada dalam bentuk *conditional statements* seperti if dan elif. Contohnya:

* Pada metode validate\_login:

python

Salin kode

if not username or not password:

messagebox.showwarning("Error", "Username dan Password harus diisi")

elif not self.terms\_var.get():

messagebox.showwarning("Error", "Anda harus menyetujui syarat & ketentuan")

Ini adalah pengkondisian untuk mengecek apakah semua input telah diisi dan checkbox disetujui.

* Pada metode show\_subcategories:

python

Salin kode

if category == "Lauk pauk":

subcategories = [("Nabati", "nabati.jpg"), ("Hewani", "hewani.jpg.webp")]

elif category == "Sayuran":

subcategories = [("Berkuah", "berkuahh.jpg"), ("Bersantan", "bersantan.jpg"), ("Tumis", "tumis.jpg")]

elif category == "Camilan":

subcategories = [("Manis", "manis.jpg"), ("Asin", "asin.jpg")]

**2. Perulangan**

Perulangan digunakan untuk menampilkan elemen-elemen dalam daftar secara dinamis:

* Dalam metode recipe\_category\_page dan show\_subcategories:

python

Salin kode

for category, img\_file in zip(["Lauk pauk", "Sayuran", "Camilan"], ["Lauk.png", "Sayurr.jpg", "camilan.jpg"]):

# Iterasi untuk menampilkan setiap kategori

python

Salin kode

for subcategory, img in images.items():

# Iterasi untuk menampilkan subkategori

* Fungsi format\_recipe\_details juga menggunakan *loop* untuk memformat daftar bahan:

python

Salin kode

return "\n".join(f"- {item}" for item in details)

**3. Function Method**

Kode menggunakan berbagai *method* atau fungsi yang didefinisikan dalam kelas. Contoh fungsi:

* welcome\_page
* login\_page
* validate\_login
* recipe\_category\_page
* show\_subcategories
* show\_recipe
* format\_recipe\_details

Setiap fungsi ini bertugas untuk menangani bagian tertentu dari alur aplikasi.

**4. OOP (Object-Oriented Programming)**

Kode ini menggunakan paradigma OOP. Berikut ciri-cirinya:

* **Kelas:** Aplikasi dibangun menggunakan kelas RecipeApp.
* **Encapsulation:** Semua logika aplikasi dibungkus dalam metode di dalam kelas.
* **Constructor:** Fungsi \_\_init\_\_ digunakan untuk inisialisasi objek:

python

Salin kode

def \_\_init\_\_(self, root):

self.root = root

self.recipe\_stack = {...}

**5. GUI (Graphical User Interface)**

Kode ini menggunakan modul **Tkinter** untuk membangun GUI. Contohnya:

* **Komponen GUI:**
  + Label untuk teks.
  + Button untuk tombol interaktif.
  + Entry untuk input pengguna.
  + Checkbutton untuk persetujuan syarat.
* **Layout:**
  + Menggunakan manajemen tata letak seperti pack, grid.
* **Interaktivitas:**
  + Event handling dilakukan melalui command pada tombol.

**Kesimpulan**

Kode ini mencakup semua elemen yang diminta, menggabungkan paradigma **OOP** dengan antarmuka **GUI**, serta menggunakan pengkondisian, perulangan, dan fungsi untuk memastikan alur kerja aplikasi.