

SISTEM INTERFACE PENCUCIAN MOTOR MENGGUNAKAN GUI DENGAN TKINTER

Untuk Memenuhi Tugas Project Akhir Praktikum Pemrograman Lanjut

Disusun oleh:

- | | |
|-------------------------------|-------------------|
| 1. Eldon Parasian Simanjuntak | (215150301111032) |
| 2. Mohd Alfitra Syauqi | (215150300111025) |
| 3. Anugrah Ilhami Rizki | (215150300111027) |
| 4. Gilang Alif Pamungkas | (215150300111032) |



PROGRAM STUDI TEKNIK KOMPUTER
DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2023

A. PENJELASAN SINGKAT PROJECT

Aplikasi yang ingin dibuat adalah Sistem Interface pencucian motor menggunakan GUI pada bahasa pemrograman Python dan library tkinter. Beberapa fitur utama yang ingin dikembangkan dalam proyek ini melibatkan operasi pencucian motor secara umum seperti login, menu utama, bayar kasir dan pemesanan makanan dan minuman. Masing-masing fitur tersebut memiliki kegunaan masing-masing yaitu :

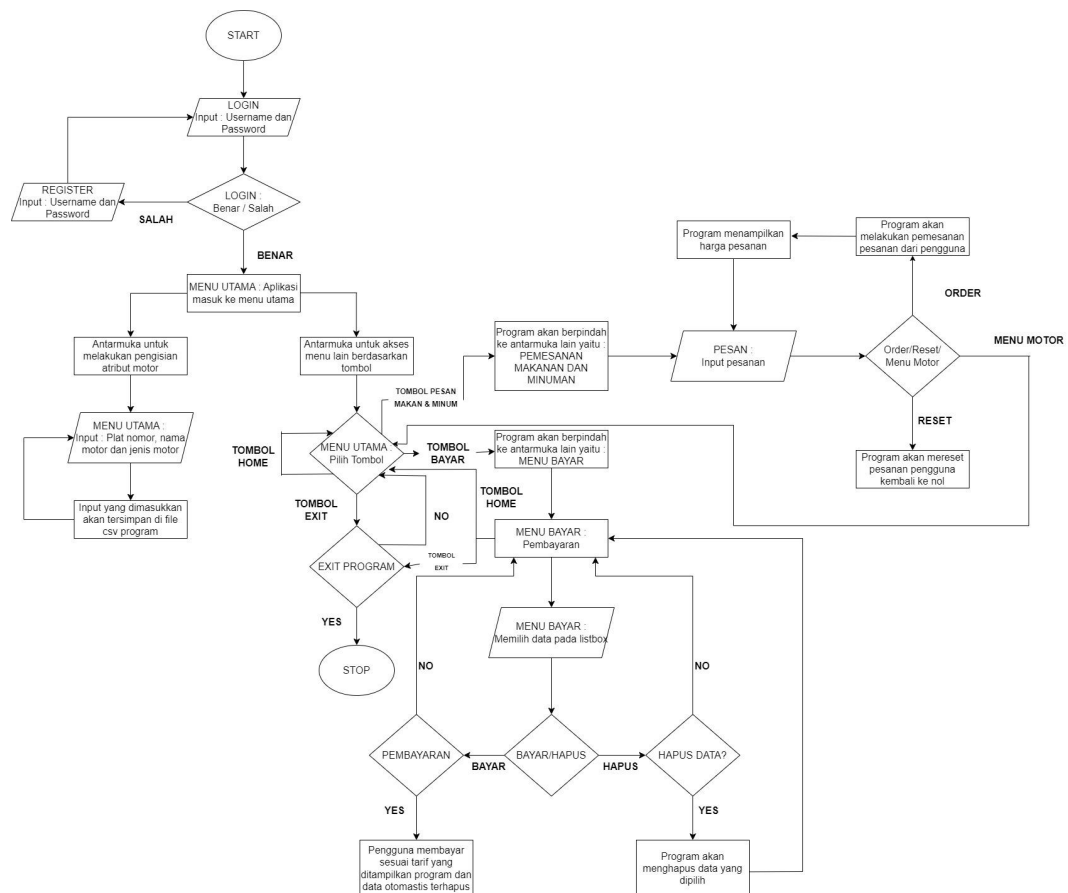
1. *Login dan Register* : Pada menu *Login* dan *Register* ini, terdapat dua pilihan awal yaitu *login* atau *register*. Jika pengguna belum memiliki akun maka bisa melakukan *register* terlebih dahulu dengan diminta untuk memasukkan *username* dan *password* yang akan digunakan untuk login nantinya. Kemudian setelah memasukkan kedua atribut tersebut akan muncul keterangan "*register successful*" dan bisa kembali ke tampilan awal menu login dengan menekan tombol "*back*". Jika sudah berhasil untuk *register* maka sudah bisa login ke aplikasi menggunakan *username* dan *password* yang telah dibuat tadi. *Login* berhasil akan mendapat keterangan "*login successful*". Kedua pilihan ini yaitu *login* dan *register* datanya akan disimpan pada file csv yang telah dibuat dan ketika sudah *login* maka aplikasi akan otomatis berubah tampilannya ke menu utama yaitu untuk pembayaran cuci motor.
2. Menu Utama : Pada menu utama ini ada beberapa tombol untuk mengakses menu lainnya yaitu tombol untuk keluar dari aplikasi, tombol kembali ke menu utama, tombol untuk pembayaran kasir dan tombol untuk pemesanan makanan atau minuman. Kemudian pada menu utama ini juga terdapat *input* yang harus dimasukkan yaitu atribut dari motor yang dicuci, yaitu plat nomor, nama motor dan jenis motor. Jenis motor pada aplikasi ini ada tiga yaitu motor kecil, motor besar dan motor trail. *Input* yang sudah dimasukkan oleh pengguna akan disimpan pada file csv yang sudah dibuat beserta dengan harga yang harus dibayar.
3. Pembayaran : Fitur Pembayaran ini bisa diakses melalui menu utama dengan menekan tombol yang ada. Setelah masuk ke fitur ini, pengguna akan mendapatkan tampilan yaitu list atribut motornya beserta harga yang harus dibayarkan. Harga ini bisa juga mendapatkan penambahan dari makanan atau minuman jika pengguna memesan makanan atau minuman. Pengguna akan memilih motor mana yang merupakan miliknya dan akan melakukan pembayaran dengan menekan tombol bayar.
4. Pemesanan Makanan atau Minuman : Fitur ini merupakan fitur dimana pengguna bisa memesan makanan atau minuman. Pengguna bisa menggunakan fitur ini dengan menekan tombol yang ada pada menu utama. Pada fitur ini disediakan 3 menu makanan dan 3 menu minuman yaitu indomie goreng, indomie goreng jumbo, indomie kuah, es teh, kopi dan es jeruk. Pada fitur ini ada beberapa tombol yang akan melakukan operasi pada aplikasi ini, dibawah masing-masing menu makanan atau minuman terdapat tombol untuk memasukkan jumlah makanan-atau minuman yang mereka pesan yang akan dikalkulasi menjadi satu dan dijumlahkan harganya. Kemudian terdapat juga tombol *order* dan *reset*, *order* akan memesan pesanan sesuai dengan yang sudah dipilih oleh

pengguna kemudian tombol *reset* akan melakukan *reset* pesanan dari pengguna sehingga jumlah pesanan pengguna akan menjadi nol semua.

Cara kerja aplikasi melibatkan interaksi pengguna dengan antarmuka grafis (GUI), dimana setiap fitur memiliki kegunaannya masing-masing seperti yang sudah dijelaskan. Fitur-fitur yang ada pada aplikasi ini bisa diakses jika pengguna sudah memiliki akun atau sudah *register* pada aplikasi ini. Pengguna akan diminta untuk *login* pada tampilan awal aplikasi untuk bisa masuk dan melakukan akses penuh ke aplikasi ini. Ketentuan dalam menjalankan aplikasi melibatkan penggunaan data registrasi yaitu *username* dan *password*. Aplikasi juga bisa memberikan pesan kesalahan jika pengguna melakukan kesalahan seperti memasukkan data yang salah atau pengguna belum memiliki akun dan kesalahan-kesalahan pengguna lainnya.

Aplikasi Sistem Interface pencucian motor menggunakan GUI pada bahasa pemrograman Python dan library tkinter ini menggunakan beberapa materi dan konsep-konsep seperti: GUI, CSV, List, function, method, OOP and exception handling, serta seleksi kondisi dan perulangan.

B. FLOW DIAGRAM



C. USE CASE SCENARIO

- Use Case 1

Use Case	Register
Tujuan	Mendaftarkan akun
Kondisi Awal	Pengguna membuka aplikasi Sistem Interface Pencucian Motor Menggunakan GUI Dengan Tkinter
Skenario Utama	<ol style="list-style-type: none">1. Pengguna memilih opsi "<i>Register</i>"2. Aplikasi meminta pengguna untuk memasukkan <i>username</i> dan <i>password</i>.3. Pengguna memasukkan <i>username</i> dan <i>password</i>4. Aplikasi meregistrasikan <i>username</i> dan <i>password</i> pengguna5. Aplikasi menampilkan keterangan "<i>register successful</i>"
Skenario Alternatif	

- Use Case 2

Use Case	Login
Tujuan	Pengguna masuk ke akun mereka untuk mengakses aplikasi secara penuh
Kondisi Awal	Pengguna membuka aplikasi Sistem Interface Pencucian Motor Menggunakan GUI Dengan Tkinter
Skenario Utama	<ol style="list-style-type: none">1. Pengguna memilih opsi "<i>Login</i>"2. Aplikasi meminta pengguna untuk memasukkan <i>username</i> dan <i>password</i>.3. Pengguna memasukkan informasi login mereka.4. Aplikasi memeriksa kebenaran <i>username</i> dan <i>password</i>.5. Jika login berhasil, aplikasi menampilkan keterangan "<i>Login successful</i>" dan masuk ke menu utama
Skenario Alternatif	Jika pengguna belum punya akun maka harus registrasi pada menu <i>register</i>

- Use Case 3

Use Case	Memasukkan Atribut Motor
Tujuan	Memungkinkan pengguna untuk memasukkan atribut motor yang akan dicuci
Kondisi Awal	Pengguna telah berhasil masuk
Skenario Utama	<ol style="list-style-type: none">1. Pengguna masuk ke menu utama2. Pengguna diminta untuk memasukkan plat nomor, nama motor dan memilih jenis motor

	<ol style="list-style-type: none"> Pengguna memasukkan plat nomor, nama motor dan memilih jenis motor Pengguna menekan tombol masukkan untuk menyimpan atribut motor
Skenario Alternatif	-

● Use Case 4

Use Case	Pembayaran Cuci Motor
Tujuan	Memungkinkan pengguna untuk mengetahui tarif dan membayar
Kondisi Awal	Pengguna telah memasukkan atribut motor
Skenario Utama	<ol style="list-style-type: none"> Pengguna memilih opsi "Bayar" dari menu utama. Aplikasi meminta pengguna untuk memilih list motor yang ada di listbox Pengguna memilih list yang ada di listbox untuk dibayar atau dihapus Aplikasi melakukan operasi sesuai keinginan pengguna yaitu untuk menghapus atau membayar tarif dari list yang dipilih pengguna.
Skenario Alternatif	-

● Use Case 5

Use Case	Membeli makanan atau minuman
Tujuan	Memungkinkan pengguna untuk membeli makanan atau minuman
Kondisi Awal	Pengguna berada di interface menu utama
Skenario Utama	<ol style="list-style-type: none"> Pengguna memilih opsi "Membeli makanan atau minuman" dengan menekan tombol bergambar keranjang Aplikasi menampilkan menu makanan dan minuman Pengguna memilih makanan atau minuman dengan memasukkan jumlah makanan atau minuman yang akan dipesan Pengguna menekan tombol "order" untuk melakukan pemesanan makanan Aplikasi akan memberikan informasi harga yang harus dibayar. Semua pesanan ditulis ke logging
Skenario Alternatif	<ol style="list-style-type: none"> Pengguna bisa melakukan reset pesanan dengan menekan tombol "reset" yang ada Pengguna bisa kembali ke case 3 untuk pengisian atribut motor dengan menekan tombol "Menu motor"

• Use Case 6

Use Case	Keluar
Tujuan	Memungkinkan pengguna untuk keluar aplikasi
Kondisi Awal	Pengguna telah berhasil masuk
Skenario Utama	<ol style="list-style-type: none"> 1. Pengguna memilih opsi "Keluar" dari menu utama dengan menekan tombol power. 2. Aplikasi meminta konfirmasi ulang pengguna apakah benar-benar ingin keluar. 3. Aplikasi menutup program dan pengguna keluar dari program
Skenario Alternatif	<ol style="list-style-type: none"> 1. Jika pengguna memilih tidak keluar setelah konfirmasi yang dikirimkan aplikasi maka pengguna diarahkan kembali ke case 3 atau menu utama.

D. KODE PROGRAM

main.py

```

1  from kasir import menu_home
2  from autentikasi import LoginGUI
3
4  def main():
5      login_gui = LoginGUI()
6      login_gui.setup()
7
8  if __name__ == "__main__":
9      main()

```

kasir.py

```

1  import tkinter as tk
2  import csv
3  import os
4  from tkinter import *
5  from tkinter import Tk, ttk, Canvas, Entry, Button, PhotoImage,
    messagebox
6  from datetime import datetime
7  from config import relative_to_assets
8  from PIL import Image, ImageTk
9  from kasirMakan import menu_makan
10
11 class menu_home:
12
13     def setup(self):
14         self.jenisMotor = []
15         self.homepage = Tk()

```

```

16         self.homepage.title("PROGRAM KASIR")
17         self.homepage.geometry("720x600")
18         self.homepage.configure(bg="#FFFFFF")
19
20         canvasH = Canvas(self.homepage, bg="#FFFFFF", height=480,
width=720, bd=0, highlightthickness=0, relief="ridge")
21         canvasH.place(x=0, y=0)
22         # entry1
23         inputP_image =
PhotoImage(file=relative_to_assets("home_entry1.png"))
24         inputP_BG = canvasH.create_image(206.5, 169.0,
image=inputP_image)
25         self.inputP = Entry(bd=0, bg="#BEDCE1", fg="#000716",
highlightthickness=0, font=("Arial 12"))
26         self.inputP.place(x=114.0, y=150.0, width=185.0,
height=38.0)
27         # entry2
28         inputP_image2 =
PhotoImage(file=relative_to_assets("home_entry1.png"))
29         inputP_BG2 = canvasH.create_image(208.5, 241.0,
image=inputP_image2)
30         self.inputP2 = Entry(bd=0, bg="#BEDCE1", fg="#000716",
highlightthickness=0, font=("Arial 12"))
31         self.inputP2.place(x=116.0, y=222.0, width=185.0,
height=38.0)
32         # combobox
33         self.framecombo = Frame(self.homepage, width=209.0,
height=40, bg="white")
34         self.framecombo.place(x=104.0, y=301.0)
35         self.combo_image =
PhotoImage(file=relative_to_assets("combobox.png"))
36         self.combol = Button(self.framecombo,
image=self.combo_image, borderwidth=0, highlightthickness=0,
command=self.comboboxD, relief="flat")
37         self.combol.place(x=0.0, y=0.0, width=209.0, height=40.0)
38         # button input data
39         dataimage =
PhotoImage(file=relative_to_assets("input_data.png"))
40         dataButton = Button(image=dataimage, borderwidth=0,
highlightthickness=0, command=self.inputdata, relief="flat")
41         dataButton.place(x=449.0, y=218.0, width=186.0,
height=45.0)
42         # Text di home
43         canvasH.create_text(109.0, 124.0, anchor="nw", text="Plat
Motor", fill="#000000", font=("OpenSansRoman Bold", 14 * -1))
44         canvasH.create_text(109.0, 198.0, anchor="nw", text="Nama
Motor", fill="#000000", font=("OpenSansRoman Bold", 14 * -1))
45         canvasH.create_text(109.0, 274.0, anchor="nw", text="Jenis
Motor", fill="#000000", font=("OpenSansRoman Bold", 14 * -1))
46         # tempat menu
47         self.frameM = Frame(self.homepage, width=720, height=70,
bg="#6E8C91")
48         self.frameM.place(x=0, y=0)

```

```

48         self.textTM = Label(self.frameM, text="MENU UTAMA")
49         self.textTM.config(font=("Open Sans", 24, 'bold'),
50 fg="white", bg="#6E8C91")
51         self.textTM.place(x=410.0, y=14.0)
52         garisIM = PhotoImage(file=relative_to_assets("garis.png"))
53         garis = Label(self.frameM, image=garisIM, bg="#6E8C91")
54         garis.place(x=315.0, y=6.0)
55         # tombol exit
56         exitBut = PhotoImage(file=relative_to_assets("exit.png"))
57         exitB = Button(image=exitBut, borderwidth=0,
58 highlightthickness=0, command=self.exitMenu, relief="flat")
59         exitB.place(x=22.0, y=8.0, width=55.0, height=55.0)
60         # tombol home
61         homeBut = PhotoImage(file=relative_to_assets("home.png"))
62         homeB = Button(image=homeBut, borderwidth=0,
63 highlightthickness=0, command=self.homeMenu, relief="flat")
64         homeB.place(x=90.0, y=8.0, width=55.0, height=55.0)
65         # tombol bayar
66         bayarBut = PhotoImage(file=relative_to_assets("bayar.png"))
67         bayarB = Button(image=bayarBut, borderwidth=0,
68 highlightthickness=0, command=self.bayarMenu, relief="flat")
69         bayarB.place(x=158.0, y=8.0, width=55.0, height=55.0)
70         #tombol makan
71         makanBut = PhotoImage(file=relative_to_assets("makan.png"))
72         makanB = Button(image=makanBut, borderwidth=0,
73 highlightthickness=0, command=self.gotoMenuMakan, relief="flat")
74         makanB.place(x=226.0, y=8.0, width=55.0, height=55.0)
75
76         #setting page home
77         self.homepage.resizable(False, False)
78         self.homepage.mainloop()
79
80     def inputdata(self):
81         platM = self.inputP.get()
82         namaM = self.inputP2.get()
83         if len(self.jenisMotor) != 0:
84             for i in range(len(self.jenisMotor) - 1,
85 len(self.jenisMotor)):
86                 jenisM = self.jenisMotor[i]
87                 if platM == "" or namaM == "" or jenisM == "":
88                     messagebox.showerror("WARNING", "Ada data yang
89 belum diinput!")
90                 else:
91                     if jenisM == 'Motor Kecil':
92                         log_time = datetime.now().strftime('%Y-%m-%d
93 %H:%M:%S')
94
95                         with open('Logging.csv', 'a', newline='') as
96 log_file:
97                             fieldnames = ['Waktu', 'PlatM', 'NamaM',
98 'JenisM', 'Harga']
99                             writer = csv.DictWriter(log_file,
100 fieldnames=fieldnames)

```


	writer.writerow({'Waktu': log_time,
90	'PlatM': platM, 'NamaM': namaM, 'JenisM': jenisM, 'Harga': 15000})
	with open('databaseM.csv', 'a', newline='') as
91	file:
	fieldnames = ['PlatM', 'NamaM', 'jenisM',
92	'Harga']
	writer = csv.DictWriter(file,
	fieldnames=fieldnames)
93	writer.writerow({'PlatM': platM, 'NamaM':
	namaM, 'jenisM': jenisM, 'Harga': 15000}))
94	
	elif jenisM == 'Motor Besar':
95	log_time = datetime.now().strftime('%Y-%m-%d
	%H:%M:%S')
96	with open('Logging.csv', 'a', newline='') as
	log_file:
97	fieldnames = ['Waktu', 'PlatM', 'NamaM',
98	'JenisM', 'Harga']
99	writer = csv.DictWriter(log_file,
	fieldnames=fieldnames)
100	writer.writerow({'Waktu': log_time,
	'PlatM': platM, 'NamaM': namaM, 'JenisM': jenisM, 'Harga': 18000}))
101	with open('databaseM.csv', 'a', newline='') as
	file:
102	fieldnames = ['PlatM', 'NamaM', 'jenisM',
	'Harga']
103	writer = csv.DictWriter(file,
	fieldnames=fieldnames)
	writer.writerow({'PlatM': platM, 'NamaM':
104	namaM, 'jenisM': jenisM, 'Harga': 18000}))
105	
	elif jenisM == 'Motor Trail':
	log_time = datetime.now().strftime('%Y-%m-%d
106	%H:%M:%S')
	with open('Logging.csv', 'a', newline='') as
107	log_file:
	fieldnames = ['Waktu', 'PlatM', 'NamaM',
108	'JenisM', 'Harga']
109	writer = csv.DictWriter(log_file,
110	fieldnames=fieldnames)
	writer.writerow({'Waktu': log_time,
111	'PlatM': platM, 'NamaM': namaM, 'JenisM': jenisM, 'Harga': 22000}))
	with open('databaseM.csv', 'a', newline='') as
112	file:
	fieldnames = ['PlatM', 'NamaM', 'jenisM',
113	'Harga']
	writer = csv.DictWriter(file,
114	fieldnames=fieldnames)
	writer.writerow({'PlatM': platM, 'NamaM':
	namaM, 'jenisM': jenisM, 'Harga': 22000}))
115	
	self.jenisMotor.pop(0)
116	self.combol.destroy()

```

117         self.combo_image =
PhotoImage(file=relative_to_assets("combobox.png"))
118         self.combo1 = Button(self.framecombo,
image=self.combo_image, borderwidth=0, highlightthickness=0,
119         command=self.comboboxD, relief="flat")
120         self.combo1.place(x=0.0, y=0.0, width=209.0,
121         height=40.0)
122         self.inputP.delete(0, END)
self.inputP2.delete(0, END)
123         messagebox.showinfo("Information", "Data berhasil
diinput!")
        else:
124             messagebox.showerror("WARNING", "Ada data yang belum
diinput!")
125
126     def comboboxD(self):
127         self.frameisiC = Frame(self.homepage, width=143,
height=111, bg="white",
128         highlightthickness=1, highlightbackground="black")
129         self.frameisiC.place(x=318.0, y=302.0)
self.comboTimage =
130         PhotoImage(file=relative_to_assets("combotrail.png"))
131         self.comboT = Button(self.frameisiC,
image=self.comboTimage, borderwidth=0, highlightthickness=0,
132         command=self.opsicombo3, relief="flat")
self.comboT.place(x=0.0, y=72.0, width=140.0, height=37.0)
133         self.comboGimage =
134         PhotoImage(file=relative_to_assets("combogede.png"))
self.comboG = Button(self.frameisiC,
135         image=self.comboGimage, borderwidth=0, highlightthickness=0,
command=self.opsicombo2, relief="flat")
self.comboG.place(x=0.0, y=40.0, width=140.0, height=31.0)
136         self.comboKimage =
137         PhotoImage(file=relative_to_assets("combokecil.png"))
self.comboK = Button(self.frameisiC,
138         image=self.comboKimage, borderwidth=0, highlightthickness=0,
command=self.opsicombo1, relief="flat")
self.comboK.place(x=0.0, y=0.0, width=140.0, height=39.0)
139
140     def opsicombo1(self):
self.frameisiC.destroy()
self.combo1.destroy()
self.combo_image =
141         PhotoImage(file=relative_to_assets("Motor_Kecil.png"))
self.combo1 = Button(self.framecombo,
image=self.combo_image, borderwidth=0, highlightthickness=0,
142         command=self.comboboxD, relief="flat")
self.combo1.place(x=0.0, y=0.0, width=209.0, height=40.0)
143         self.jenisMotor.append("Motor Kecil")
144
145     def opsicombo2(self):
self.frameisiC.destroy()
self.combo1.destroy()
146
147
148

```

```

        self.combo_image =
        PhotoImage(file=relative_to_assets("Motor_Besar.png"))
149         self.combol = Button(self.framecombo,
150 image=self.combo_image, borderwidth=0, highlightthickness=0,
151 command=self.comboboxD, relief="flat")
152         self.combol.place(x=0.0, y=0.0, width=209.0, height=40.0)
153         self.jenisMotor.append("Motor Besar")
154
155     def opsicombo3(self):
156         self.frameisiC.destroy()
157         self.combol.destroy()
158         self.combo_image =
        PhotoImage(file=relative_to_assets("Motor_Trail.png"))
159         self.combol = Button(self.framecombo,
160 image=self.combo_image, borderwidth=0, highlightthickness=0,
161 command=self.comboboxD, relief="flat")
162         self.combol.place(x=0.0, y=0.0, width=209.0, height=40.0)
163         self.jenisMotor.append("Motor Trail")
164
165     def exitMenu(self):
166         msg_box = tk.messagebox.askquestion('Keluar Aplikasi',
167 'Apakah ingin keluar dari aplikasi?', icon='warning')
168         if msg_box == 'yes':
169             self.homepage.destroy()
170         else:
171             pass
172
173     def gotoMenuMakan(self):
174         self.homepage.destroy()
175         makanGUI = menu_makan()
176         makanGUI.setup()
177
178     def cekDatabase(self):
179         if not os.path.isfile('databaseM.csv'):
180             with open('databaseM.csv', 'w', newline='') as temp2:
181                 kolom = ['plat', 'nama', 'jenis', 'harga']
182                 databaru = csv.DictWriter(temp2, kolom)
183                 databaru.writeheader()
184
185     def homeMenu(self):
186         try:
187             self.textTM2.destroy()
188             frameB.destroy()
189             frameBD.destroy()
190             textTM = Label(self.frameM, text="MENU UTAMA")
191             textTM.config(font=("Open Sans", 24, 'bold'),
192 fg="white", bg="#6E8C91")
193             textTM.place(x=410.0, y=14.0)
194         except:
195             pass
196
197     def bayarMenu(self):
198         data = []
199         try:

```

```

194         self.textTM.destroy()
195         self.textTM2 = Label(self.frameM, text="MENU BAYAR")
196         self.textTM2.config(font=("Open Sans", 24, 'bold'),
197 fg="white", bg="#6E8C91")
198         self.textTM2.place(x=413.0, y=14.0)
199     except:
200         pass
201
202     def bayar():
203         total_tmp = []
204         total = 0
205         temp1 = open('databaseM.csv', 'r')
206         harga = self.treeview.selection()
207
208         for item in harga:
209             index = self.treeview.index(item)
210             total_tmp = data[index]
211             total += int(total_tmp["harga"])
212
213         msg_box = tk.messagebox.askquestion('PEMBAYARAN',
214 f"Total pembayaran adalah {total}. Apakah Anda yakin ingin
215 membayar?")
216         temp1.close()
217
218         if msg_box == 'yes':
219             for item in harga:
220                 index = self.treeview.index(item)
221                 data.pop(index)
222
223                 with open('databaseM.csv', 'w', newline='') as
224 temp2:
225                     kolom = ['plat', 'nama', 'jenis', 'harga']
226                     databaru = csv.DictWriter(temp2, kolom)
227                     databaru.writeheader()
228                     for jumlah in data:
229                         databaru.writerow(jumlah)
230                     temp2.close()
231
232                     for item in harga[:-1]:
233                         self.treeview.delete(item)
234             else:
235                 pass
236
237     def hapusdata():
238         msg_box = tk.messagebox.askquestion('HAPUS DATA',
239 'Apakah ingin menghapus data?', icon='warning')
240         if msg_box == 'yes':
241             temp1 = open('databaseM.csv', 'r')
242             items_to_delete = self.treeview.selection()
243
244             indexes_to_delete = [self.treeview.index(item) for
245 item in items_to_delete]

```

```

240         for index in sorted(indexes_to_delete,
241 reverse=True):
242             data.pop(index)
243
244             temp1.close()
245             with open('databaseM.csv', 'w', newline='') as
246 temp2:
247                 kolom = ['plat', 'nama', 'jenis', 'harga']
248                 databaru = csv.DictWriter(temp2, kolom)
249                 databaru.writeheader()
250
251                 for jumlah in data:
252                     databaru.writerow(jumlah)
253                 temp2.close()
254
255                 for item in items_to_delete:
256                     self.treeview.delete(item)
257             else:
258                 pass
259
260     def isitabel():
261         global frameB, frameBD, butbayarIM, buthapusIM
262
263         frameB = tk.Frame(self.homepage, width=720, height=600,
264 bg="white")
265         frameB.place(x=0, y=70)
266         frameBD = tk.Frame(frameB, width=430, height=348,
267 bg="white", highlightthickness=1, highlightbackground="black")
268         frameBD.place(x=40.0, y=30)
269
270         # scrollbar
271         scrolly = tk.Scrollbar(frameBD, orient=tk.VERTICAL)
272         scrolly.pack(side=tk.RIGHT, fill=tk.Y)
273
274         # horizontal scrollbar
275         scrollx = tk.Scrollbar(frameBD, orient=tk.HORIZONTAL)
276         scrollx.pack(side=tk.BOTTOM, fill=tk.X)
277
278         # treeview with checkboxes
279         self.treeview = ttk.Treeview(frameBD, columns=("PlatM",
280 "NamaM", "jenisM", "Harga"), selectmode='extended',
281 yscrollcommand=scrolly.set, xscrollcommand=scrollx.set)
282         self.treeview.heading("#0", text="Index")
283         self.treeview.heading("PlatM", text="Plat Motor")
284         self.treeview.heading("NamaM", text="Nama Motor")
285         self.treeview.heading("jenisM", text="Jenis Motor")
286         self.treeview.heading("Harga", text="Harga")
287         self.treeview.column("#0", width=0, stretch=tk.NO)
288         self.treeview.column("PlatM", anchor=tk.W, width=150)
289         self.treeview.column("NamaM", anchor=tk.W, width=150)
290         self.treeview.column("jenisM", anchor=tk.W, width=150)
291         self.treeview.column("Harga", anchor=tk.W, width=150)

```

```

287         self.treeview.pack(expand=tk.YES, fill=tk.BOTH,
288 side=tk.LEFT)
289
290         # scrollbar config
291         scrolly.config(command=self.treeview.yview)
292         scrollx.config(command=self.treeview.xview)
293
294         try:
295             self.cekDatabase()
296             pass
297         except:
298             raise ("File already exists")
299
300         with open('databaseM.csv') as f:
301             reader = csv.DictReader(f, delimiter=',')
302             for idx, row in enumerate(reader):
303                 data.append(row)
304                 self.treeview.insert("", "end",
305 values=(row["plat"], row["nama"], row["jenis"], row["harga"]))
306                 data[-1]["index"] = idx
307
308         # tombol
309         butbayarIM =
310 tk.PhotoImage(file=relative_to_assets("tombolbayar.png"))
311         butbayar = tk.Button(frameB, image=butbayarIM,
312 borderwidth=0, highlightthickness=0, command=bayar, relief="flat")
313         butbayar.place(x=170.0, y=400.0, width=130.0,
314 height=39.0)
315
316         buthapusIM =
317 tk.PhotoImage(file=relative_to_assets("hapus.png"))
318         button_2 = tk.Button(frameB, image=buthapusIM,
319 borderwidth=0, highlightthickness=0, command=hapusdata,
320 relief="flat")
321         button_2.place(x=400.0, y=400.0, width=130.0,
322 height=39.0)
323
324         try:
325             frameB.destroy()
326             isitabel()
327         except:
328             isitabel()
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500

```

kasirMakan.py

```
1 import tkinter as tk
2 from tkinter import *
3 from PIL import Image, ImageTk
4 from tkinter import messagebox
5 from config import relative_to_assets
6
7 class Menu:
8     def __init__(self, window, frame, label_text, images,
9 spinbox_from, spinbox_to, item_type, prices):
10         self.window = window
11         self.frame = frame
12         self.label = Label(self.frame, text=label_text,
13 font=("Times New Roman", 20, "bold"))
14         self.label.grid(column=0, row=0, columnspan=3)
15
16         self.canvas = [Canvas(self.frame, width=200, height=200,
17 highlightthickness=0) for _ in range(len(images))]
18         self.images = [PhotoImage(file=relative_to_assets(img)) for
19 img in images]
20
21         self.spinbox_list = []
22         for i, img in enumerate(self.images):
23             self.canvas[i].create_image(100, 100, image=img,
24 anchor='center')
25             self.canvas[i].image = img
26             self.canvas[i].grid(column=i, row=1, pady=10)
27
28             spinbox = Spinbox(self.frame, from_=spinbox_from,
29 to=spinbox_to, width=5)
30             spinbox.grid(column=i, row=2)
31             self.spinbox_list.append(spinbox)
32
33         self.item_type = item_type
34         self.prices = prices
35
36 class FoodMenu(Menu):
37     def __init__(self, window, frame, label_text, images,
38 spinbox_from, spinbox_to):
39         prices = [15, 20, 25] # Adjust the prices for each food
40 item
41         super().__init__(window, frame, label_text, images,
42 spinbox_from, spinbox_to, "food", prices)
43
44 class DrinkMenu(Menu):
45     def __init__(self, window, frame, label_text, images,
46 spinbox_from, spinbox_to):
47         prices = [10, 12, 15] # Adjust the prices for each drink
48 item
49         super().__init__(window, frame, label_text, images,
50 spinbox_from, spinbox_to, "drink", prices)
51
52 class menu_makan:
```

```

40     def setup(self):
41         self.window = Tk()
42         self.window.title("Warmino Restaurant")
43         self.window.geometry("680x700")
44
45         bg_image = Image.open(relative_to_assets("indomiebg.png"))
46         resized_bg_image = bg_image.resize((680, 700))
47         tk_bg_image = ImageTk.PhotoImage(resized_bg_image)
48
49         background_label = Label(self.window, image=tk_bg_image)
50         background_label.image = tk_bg_image
51         background_label.place(relwidth=1, relheight=1)
52
53         self.food_frame = Frame(self.window, padx=10, pady=10)
54         self.food_frame.grid(column=0, row=3, padx=20)
55
56         self.food_menu = FoodMenu(self.window, self.food_frame,
57 "Pilih Makanan",
58                                     ["indomie.png",
59 "indomiejumbo.png", "indomiekuah.png"], 0, 10)
60
61         self.drink_frame = Frame(self.window, padx=10, pady=10)
62         self.drink_frame.grid(column=0, row=4, padx=20)
63
64         self.drink_menu = DrinkMenu(self.window, self.drink_frame,
65 "Pilih Minuman",
66                                     ["esteh.png", "kopi.png",
67 "esjeruk.png"], 0, 10)
68
69         self.finish = Button(text="Order",
70 command=self.button_clicked)
71         self.finish.place(relx=0.2, rely=0.95, anchor="center")
72
73         self.reset_button = Button(text="Reset",
74 command=self.reset_value)
75         self.reset_button.place(relx=0.5, rely=0.95,
76 anchor="center")
77
78         self.reset_button = Button(text="Menu Motor",
79 command=self.gotoJunction)
80         self.reset_button.place(relx=0.8, rely=0.95,
81 anchor="center")
82
83         self.window.resizable(False, False)
84         self.window.mainloop()
85
86     def gotoJunction(self):
87         from kasir import menu_home
88         self.window.destroy()
89         backGUI = menu_home()
90         backGUI.setup()
91
92     def reset_value(self):
93

```


84	for spinbox in self.food_menu.spinbox_list +
85	self.drink_menu.spinbox_list:
	spinbox.delete(0, tk.END)
86	spinbox.insert(0, 0)
87	
88	def button_clicked(self):
91	food_quantities = [int(spinbox.get()) for spinbox in
92	self.food_menu.spinbox_list]
	drink_quantities = [int(spinbox.get()) for spinbox in
93	self.drink_menu.spinbox_list]
	food_total = sum(quantity * price for quantity, price in
95	zip(food_quantities, self.food_menu.prices))
	drink_total = sum(quantity * price for quantity, price in
96	zip(drink_quantities, self.drink_menu.prices))
	total_bills = food_total + drink_total
97	self.rupiah = f"{total_bills}.000"
98	self.saveToCsv(food_quantities, drink_quantities)
99	
	messagebox.showinfo("Rincian pesanan", f"Total harga
101	pesanan anda adalah: Rp. {self.rupiah}")
	def saveToCsv(self, food_quantities, drink_quantities):
103	log_time = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
104	makan = ', '.join([f"{quantity} {food}" for quantity, food
105	in zip(food_quantities, ["Indomie", "Indomie Jumbo", "Indomie
	Kuah"])])
	minum = ', '.join([f"{quantity} {drink}" for quantity,
106	drink in zip(drink_quantities, ["Es Teh", "Kopi", "Es Jeruk"])])
	with open('Logging.csv', 'a', newline='') as log_file:
108	fieldnames = ['Waktu', 'Makan', 'Minum', 'Harga']
109	writer = csv.DictWriter(log_file,
110	fieldnames=fieldnames)
	writer.writerow({'Waktu': log_time, 'Makan': makan,
111	'Minum': minum, 'Harga': self.rupiah})

otentikasi.py

```

1 import tkinter as tk
2 import customtkinter
3 import csv
4 import os
5 from tkinter import PhotoImage, messagebox
6 from config import relative_to_assets
7 from kasir import menu_home
8
9 class LoginGUI:
10
11     def setup(self):
12         if 'self.window' in globals():
13             self.window.destroy()
14 
```

```

15         customtkinter.set_appearance_mode("System") # Modes:
system (default), light, dark
16         customtkinter.set_default_color_theme("green") # Themes:
blue (default), dark-blue, green
17
18         self.logintk = customtkinter.CTk()
19         self.logintk.geometry("600x440")
20         self.logintk.title('Login')
21
22         img1 = PhotoImage(file=relative_to_assets("pattern.png"))
23         l1=customtkinter.CTkLabel(master=self.logintk,image=img1)
24         l1.pack()
25
26         frame=customtkinter.CTkFrame(master=l1, width=320,
height=360, corner_radius=15)
27         frame.place(relx=0.5, rely=0.5, anchor=tk.CENTER)
28
29         l2=customtkinter.CTkLabel(master=frame, text="Admin
Login",font=('Century Gothic',20))
30         l2.place(x=50, y=45)
31
32         self.nameI=customtkinter.CTkEntry(master=frame, width=220,
placeholder_text='Username')
33         self.nameI.place(x=50, y=110)
34
35         self.passI=customtkinter.CTkEntry(master=frame, width=220,
placeholder_text='Password', show="*")
36         self.passI.place(x=50, y=165)
37
38         button1 = customtkinter.CTkButton(master=frame, width=220,
text="Login", command=self.login, corner_radius=6)
39         button1.place(x=50, y=240)
40
41         button2 = customtkinter.CTkButton(master=frame, width=220,
text="Register", command=self.register, corner_radius=6)
42         button2.place(x=50, y=280)
43         self.logintk.mainloop()
44
45         def register(self):
46             self.cekfile()
47             self.logintk.destroy()
48
49         customtkinter.set_appearance_mode("System") # Modes:
system (default), light, dark
50         customtkinter.set_default_color_theme("green") # Themes:
blue (default), dark-blue, green
51
52         self.window = customtkinter.CTk()
53         self.window.geometry("600x440")
54         self.window.title('Login')
55
56         img1 = PhotoImage(file=relative_to_assets("pattern.png"))
57         l1=customtkinter.CTkLabel(master=self.window,image=img1)
l1.pack()

```

```

58
59         frame=customtkinter.CTkFrame(master=l1, width=320,
60 height=360, corner_radius=15)
        frame.place(relx=0.5, rely=0.5, anchor=tk.CENTER)
61
62         l2=customtkinter.CTkLabel(master=frame, text="Admin
63 Register",font=('Century Gothic',20))
        l2.place(x=50, y=45)
64
65         self.nameI=customtkinter.CTkEntry(master=frame, width=220,
66 placeholder_text='Username')
        self.nameI.place(x=50, y=110)
67
68         self.passI=customtkinter.CTkEntry(master=frame, width=220,
69 placeholder_text='Password', show="*")
        self.passI.place(x=50, y=165)
70
71         button1 = customtkinter.CTkButton(master=frame, width=220,
72 text="Register", command=self.save_to_csv, corner_radius=6)
        button1.place(x=50, y=220)
73
74         button2 = customtkinter.CTkButton(master=frame, width=50,
75 text="Back", command=self.goToBack, corner_radius=6)
        button2.place(x=220, y=260)
76
77         self.window.mainloop()
78
79         def goToBack(self):
80             self.window.destroy()
81             self.setup()
82
83         def save_to_csv(self):
84             self.cekfile()
85             with open('register.csv', 'a', newline='') as file:
86                 writer = csv.writer(file)
87                 writer.writerow([self.nameI.get(), self.passI.get()])
88                 messagebox.showinfo("Success", "Register successful, please
89 go back to login menu")
            self.setup()
90
91         def login(self):
92             self.cekfile()
93             with open('register.csv', 'r') as file:
94                 reader = csv.reader(file)
95                 for row in reader:
96                     if [self.nameI.get(), self.passI.get()] == row:
97                         messagebox.showinfo("Success", "Login
98 successful")
99                         self.logintk.destroy()
100                         menu_GUI = menu_home()
101                         menu_GUI.setup()
102                         return
103                     messagebox.showerror("Error", "Login failed")

```

103	
104	def cekfile(self):
105	if not os.path.isfile('register.csv'):
106	with open('register.csv', 'w') as file:
107	writer = csv.writer(file)
108	
109	

config.py	
1	from pathlib import Path
2	
3	OUTPUT_PATH = Path(__file__).parent
4	ASSETS_PATH = OUTPUT_PATH / Path(r"assets")
5	
6	def relative_to_assets(path: str) -> Path:
7	return ASSETS_PATH / Path(path)

E. PENJELASAN KODE PROGRAM

main.py	
1	Mengimpor fungsi atau kelas menu_home dari modul kasir.
2	Mengimpor kelas LoginGUI dari modul autentikasi.
4	Mendefinisikan fungsi main() sebagai fungsi utama program.
5	Membuat objek LoginGUI dengan menggunakan kelas yang diimpor sebelumnya.
6	Memanggil metode setup() pada objek login_gui, mungkin untuk mengatur tampilan GUI atau melakukan inisialisasi lainnya.
8	Memeriksa apakah skrip dijalankan sebagai program utama (bukan diimpor sebagai modul).
9	Memanggil fungsi main() jika skrip dijalankan sebagai program utama.

kasir.py	
1-9	Program menggunakan library tkinter untuk membuat antarmuka pengguna grafis (GUI). Mengimpor berbagai modul seperti csv untuk menangani file CSV, os untuk berinteraksi dengan sistem operasi, PIL untuk menangani gambar, dan menu_makan dari kasirMakan untuk menu lainnya.
11	Program mendefinisikan kelas bernama menu_home untuk mengelompokkan fungsionalitas GUI.
13-75	Metode setup menginisialisasi jendela utama (homepage) dan menyiapkan komponen GUI, termasuk tombol, bidang masukan, label, dan frame.
77-129	Metode ini mengambil data dari bidang masukan (Plat Motor, Nama Motor) dan jenis sepeda motor yang dipilih. Kemudian data

	tersebut ditulis ke file CSV ('LOGGING.csv' dan 'databaseM.csv').
131-166	Metode ini menangani pembuatan dan interaksi dengan combobox yang memungkinkan pengguna memilih jenis sepeda motor.
168-185	exitMenu meminta konfirmasi dari pengguna sebelum menutup aplikasi. gotoMenuMakan menutup jendela saat ini dan membuka menu lain terkait makanan (menu_makan). cekDatabase memeriksa apakah file 'databaseM.csv' ada dan membuatnya jika tidak.
187-258	homeMenu mengatur kembali GUI ke menu utama. bayarMenu menampilkan menu baru untuk mengelola pembayaran, termasuk tombol untuk membayar dan menghapus item yang dipilih.
264-324	Fungsi ini mengisi dan memperbarui tabel (treeview) dengan data dari 'databaseM.csv'. Ini juga mencakup tombol untuk pembayaran dan penghapusan data.

kasirMakan.py	
1-5	Program mengimpor modul-modul yang diperlukan, termasuk tkinter untuk GUI, PIL untuk pemrosesan gambar, dan messagebox untuk menampilkan kotak pesan. Juga mengimpor fungsi relative_to_assets dari modul bernama config.
7-28	Kelas Menu adalah kelas generik untuk membuat menu dengan gambar, spinbox, dan harga terkait. Menerima parameter seperti jendela, frame, teks label, gambar, rentang spinbox, jenis item, dan harga.
30-38	FoodMenu dan DrinkMenu adalah subclass dari kelas Menu. Mereka mewarisi dari kelas Menu dan memberikan harga spesifik untuk item makanan dan minuman.
40-52	Kelas menu_makan bertanggung jawab untuk menyiapkan jendela utama, membuat frame untuk menu makanan dan minuman, dan menginisialisasi instance dari FoodMenu dan DrinkMenu.
54-76	GUI diatur dengan judul, ukuran, dan gambar latar belakang. Dua frame (food_frame dan drink_frame) dibuat untuk menampilkan menu makanan dan minuman. Instance FoodMenu dan DrinkMenu dibuat dengan parameter spesifik. Tombol Order, Reset, dan Menu Motor dibuat dengan perintah terkait.
78-93	button_clicked menghitung total harga berdasarkan kuantitas dan harga yang dipilih, lalu menampilkan kotak pesan dengan total tersebut. reset_value mengatur ulang spinbox menjadi nol. gotoJunction navigasi ke menu lain (menu_home) dengan menghancurkan jendela saat ini.

103-111	fungsi untuk menyimpan data pesanan yang diterima ke file csv bernama logging yang sama seperti file pada logging di menumotor, sehingga setiap pesanan yang ada akan disimpan dan terdapat waktu, jumlah pesanan, dan jumlah harga
---------	---

autentikasi.py

1-7	Program mengimpor berbagai modul yang dibutuhkan, termasuk tkinter untuk GUI kustom (customtkinter), csv untuk operasi CSV, os untuk operasi sistem, dan beberapa modul lainnya. Modul menu_home dari file kasir juga diimpor.
9	Kelas LoginGUI memiliki beberapa metode untuk mengatur tampilan login, registrasi, kembali ke halaman sebelumnya, menyimpan data ke file CSV, melakukan login, dan memeriksa keberadaan file.
11-43	Metode ini mengatur tampilan awal untuk login. Menggunakan modul kustom customtkinter untuk membuat antarmuka pengguna.
45-78	Metode ini digunakan untuk menangani registrasi pengguna baru. Menyertakan tombol "Back" untuk kembali ke halaman login.
80-82	Metode ini digunakan untuk kembali ke halaman login setelah registrasi.
84-90	Metode ini digunakan untuk menyimpan data username dan password ke file CSV (register.csv). Setelah penyimpanan, menampilkan pesan sukses dan mengatur ulang tampilan.
92-103	Metode ini menangani proses login. Mengecek data login dengan data yang tersimpan dalam file CSV. Jika login berhasil, menampilkan pesan sukses dan membuka halaman menu (menu_home).
106-109	Metode ini memeriksa keberadaan file register.csv. Jika file tidak ada, maka membuat file tersebut.

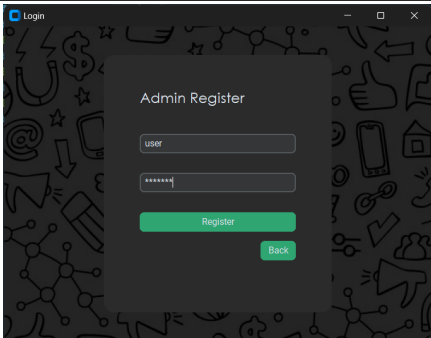
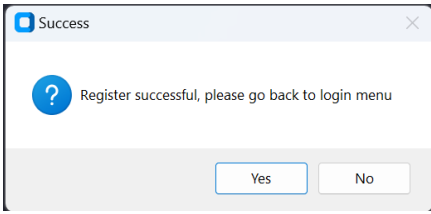
conifg.py

1	Program mengimpor kelas Path dari modul pathlib. Kelas ini digunakan untuk melakukan operasi terkait path file dan direktori.
3	OUTPUT_PATH: Mewakili path dari direktori tempat file ini (skrip ini) berada.
4	ASSETS_PATH: Mewakili path dari direktori "assets" yang ada di dalam direktori tempat file ini berada.
6	Fungsi ini digunakan untuk menghasilkan path yang bersifat relatif terhadap direktori "assets". Menerima satu parameter path yang merupakan path relatif terhadap direktori "assets" dan

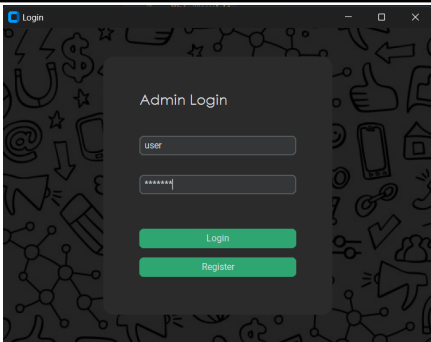
	mengembalikan objek Path yang merepresentasikan path lengkap dari file tersebut.
--	--

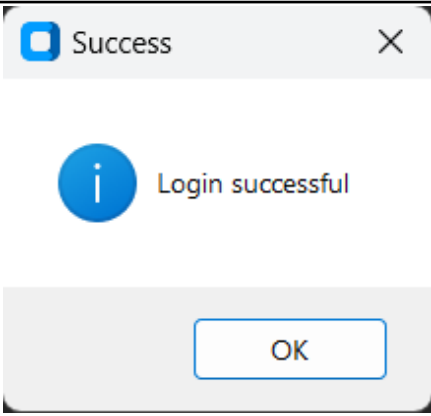
F. TEST CASE SCENARIO

• Register

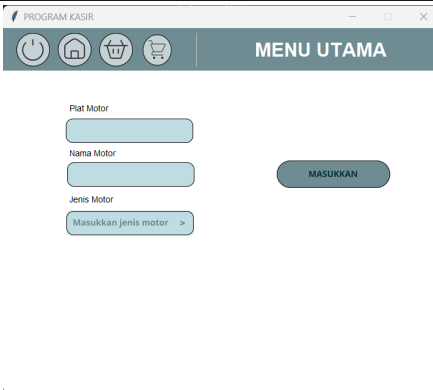
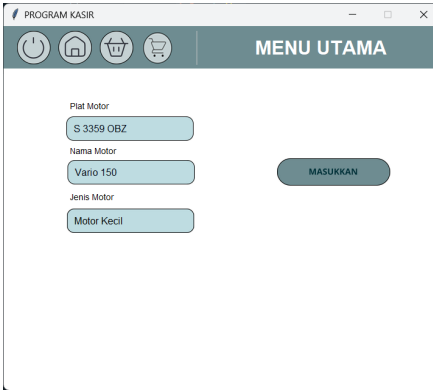
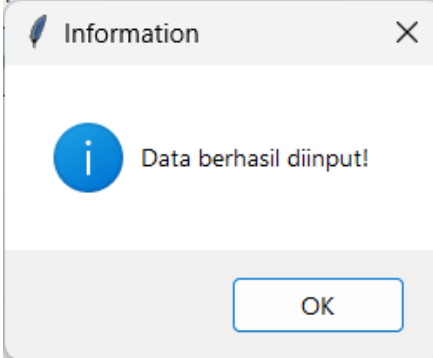
Step	Expected Result
1. Pengguna membuka aplikasi.	
2. Pengguna memasukkan username dan password yang ingin diregistrasikan.	
3. Pengguna berhasil register akun. Ketika dipencet yes maka akan otomatis kembali ke menu login, sedangkan jika dipencet no, maka akan tetap di menu register	

• Login

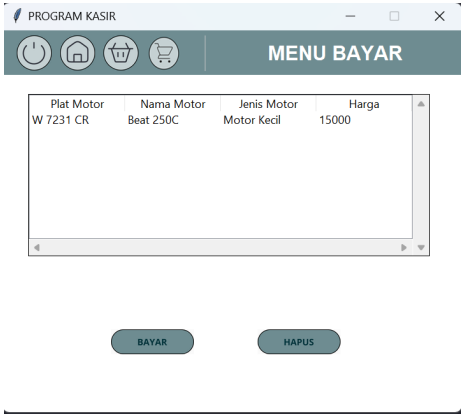
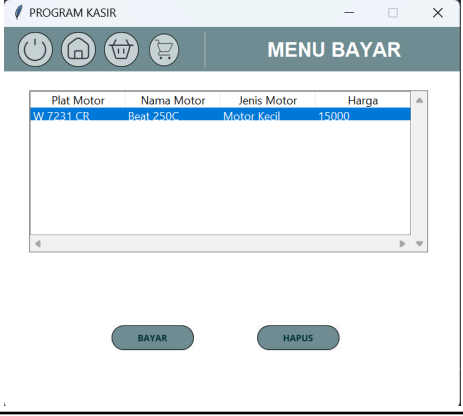
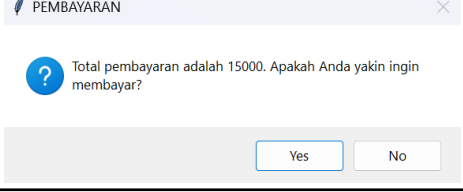
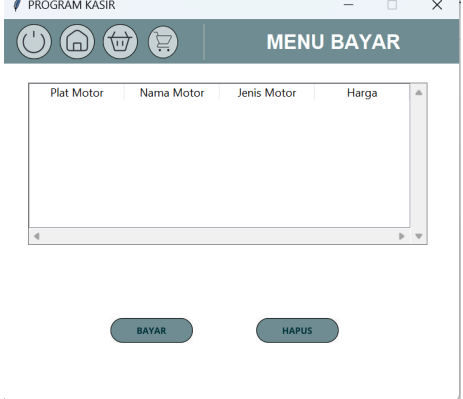
Step	Expected Result
1. Pengguna membuka aplikasi.	
2. Pengguna memasukkan username dan password yang sudah diregistrasi.	

3. Pengguna berhasil login.	
-----------------------------	--

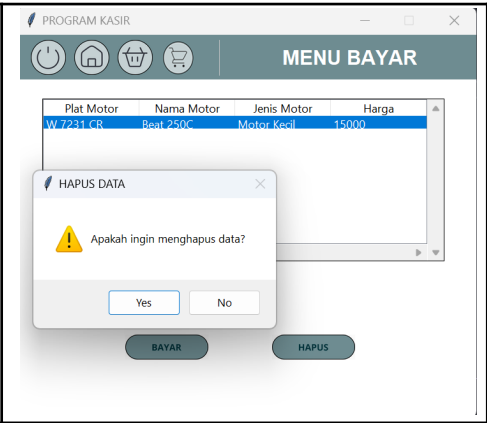
- **Menu Utama**

Step	Expected Result
1. Pengguna berhasil masuk.	
2. Setelah berhasil login, aplikasi akan langsung mengarahkan ke Menu Utama.	
3. Pengguna memasukkan Plat Motor, Nama Motor, dan Jenis Motor mereka serta klik tombol 'Masukkan'.	
4. Pengguna berhasil input data motor yang ingin dicuci.	

- **Menu Bayar**

Step	Expected Result
1. Pengguna berhasil masuk.	
2. Pengguna klik icon bayar yang terdapat di atas kiri window aplikasi.	
3. Pengguna memilih motor yang ingin dibayar dan klik tombol 'Bayar'.	
4. Aplikasi menampilkan harga pencucian motor sesuai jenis motor yang dimana pada kali ini jenis motor kecil seharga Rp15.000	
5. Setelah pengguna klik tombol 'Yes', maka daftar motor yang telah dibayar akan hilang dengan sendirinya.	

6. Jika pengguna telah memilih list box dan klik tombol hapus, maka akan menghapus data yang dipilih ditabel dan juga di file csv



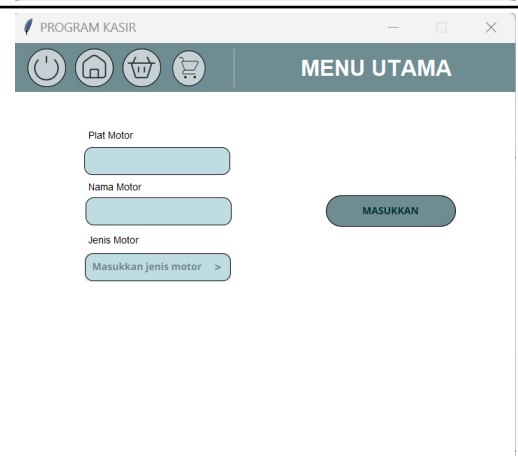
● Warmindo Restaurant

Step	Expected Result
1. Pengguna berhasil masuk.	
2. Pengguna klik icon Warmindo yang terdapat di atas kiri window aplikasi.	
3. Pengguna memilih makanan dan minuman dengan jumlah yang diinginkan. Misal Indomie Jumbo 1 dan Es Teh 1. Lalu, klik tombol ‘Order’.	<p>The screenshot shows the 'Warmindo Restaurant' application interface. It has two main sections: 'Pilih Makanan' and 'Pilih Minuman'. Under 'Pilih Makanan', there are three items: 'Indomie Jumbo', 'Mi Instan', and 'Soto Mie', each with a quantity selector set to 0. Under 'Pilih Minuman', there are three items: 'Es Teh', 'Kopi', and 'Jus', each with a quantity selector set to 0. At the bottom, there are 'Order', 'Reset', and 'Menu Motor' buttons.</p>
4. Aplikasi menampilkan rincian pesanan dan harga yang harus dibayarkan sebesar Rp25.000	<p>The screenshot shows a 'Rincian pesanan' dialog box. It contains an information icon and the text 'Total harga pesanan anda adalah : Rp. 25.000'. There is an 'OK' button at the bottom right.</p>

5. Jika user mengklik tombol reset maka widget spinbox yang semula ada nilainya akan kembali ke 0



6. Setelah itu, jika pengguna mengklik tombol menu motor maka akan kembali ke menu motor.



G. KESIMPULAN DAN SARAN

Kesimpulan:

Pembuatan aplikasi Sistem Interface Pencucian Motor Menggunakan GUI dengan Tkinter menggunakan bahasa pemrograman Python adalah tugas yang kompleks namun penting. Dalam proyek ini, kami telah berhasil membuat sejumlah fitur utama, termasuk register admin, login admin, menu utama (pemilihan tipe motor), menu bayar, dan warmino restaurant. Aplikasi ini memungkinkan pengguna untuk melakukan operasi pencucian motor dengan tambahan warmino.

Saran:

Proyek ini masih dapat ditingkatkan dengan penambahan fitur tambahan, seperti riwayat pencucian, durasi pencucian, tambahan menu yang lain pada warmino, dan lainnya. Selain itu, penting untuk selalu memastikan bahwa aplikasi ini sesuai dengan operasi pencucian motor secara umum.