

LAPORAN PROYEK Graphical User Interface (GUI)
PEMROGAMAN DASAR



Dosen Pengampu :

Dr. Atik Wintarti, M.Kom.

Siska Puspitaningsih, S.Kom., M.Kom.

Disusun oleh :

1. Yuan Fasich Tansatrisna (25031554155)
2. Wiliyan Surya (25031554155)
3. Dinda Alifia Eka Nirmala (25031554155)

HIMPUNAN MAHASISWA SAINS DATA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI SURABAYA

2025

DAFTAR ISI

BAB I	3
PENDAHULUAN	3
1.1 Latar Belakang.....	3
1.2 Rumusan Masalah.....	3
1.3 Tujuan	4
BAB II.....	6
ANALISIS DAN PERANCANGAN	6
2.1 Analisis Kebutuhan Program (Aplikasi).....	6
2.2 Diagram Alur atau Flowchart	13
BAB III.....	16
IMPLEMENTASI PENJELASAN KODE PROGRAM	16
3.1 main.py.....	16
3.2 Homepage.py	20
3.3 Payment Page.py.....	21
3.4 Profile page.py	23
DAFTAR PUSTAKA.....	26

BAB I

PENDAHULUAN

1.1 Latar Belakang

Aplikasi Travel.id merupakan sebuah platform digital yang dirancang untuk mempermudah proses pemesanan tiket perjalanan secara online. Dengan adanya aplikasi ini, pengguna dapat dengan mudah mencari, memilih, dan memesan tiket perjalanan kapan saja dan di mana saja melalui perangkat digital. Penerapan sistem pemesanan berbasis aplikasi diharapkan dapat meningkatkan efisiensi, kecepatan, serta kenyamanan pengguna dibandingkan dengan metode pemesanan konvensional.

Aplikasi ini memiliki Home Page yang berfungsi sebagai halaman utama untuk menampilkan informasi layanan dan daftar tiket yang tersedia. Selain itu, tersedia Payment Page yang mendukung metode pembayaran QRIS, sehingga pengguna dapat melakukan transaksi secara aman, cepat, dan praktis. Aplikasi ini juga dilengkapi dengan Profile Page yang menampilkan informasi data login pengguna, sehingga pengguna dapat mengelola data akun mereka dengan lebih mudah.

Dengan adanya aplikasi pemesanan tiket online ini, proses transaksi menjadi lebih terstruktur dan transparan, sementara pengguna dapat mengakses informasi pemesanan dan data akun secara terintegrasi dalam satu aplikasi. Diharapkan aplikasi Travel.id ini dapat memberikan solusi digital yang efektif dalam mendukung kebutuhan pemesanan tiket di era digital modern.

1.2 Rumusan Masalah

Proses pembelian tiket wisata yang masih dilakukan secara langsung di lokasi sering menimbulkan berbagai permasalahan bagi pengunjung. Panjangnya antrean saat pembelian tiket menyebabkan pengunjung harus menghabiskan waktu dan tenaga yang cukup besar, sehingga menjadi kurang efisien terutama bagi mereka yang memiliki keterbatasan waktu. Selain itu, pembelian tiket secara langsung juga memiliki risiko kehabisan tiket, khususnya pada waktu-waktu tertentu seperti akhir pekan dan musim liburan, yang dapat mengganggu rencana perjalanan. Pengunjung juga sering menghadapi ketidakpastian harga, di mana harga tiket yang dibeli di lokasi dapat menjadi lebih mahal dibandingkan dengan pembelian yang direncanakan sebelumnya. Sistem konvensional ini umumnya belum mendukung metode pembayaran digital seperti QRIS, sehingga proses transaksi menjadi kurang praktis. Kurangnya

informasi mengenai jam operasional dan waktu terbaik untuk berkunjung ke tempat wisata juga menjadi kendala, karena pengunjung tidak dapat menentukan waktu kunjungan secara optimal.

Untuk mengatasi permasalahan tersebut, Kami mengembangkan sebuah platform aplikasi Travel.id, sebuah Aplikasi pemesanan tiket wisata berbasis digital yang dirancang untuk memberikan kemudahan, efisiensi, dan kenyamanan bagi pengguna. Aplikasi ini memungkinkan pengguna untuk melakukan pemesanan tiket secara online tanpa harus mengantre, memperoleh informasi harga secara transparan, serta meminimalkan risiko kehabisan tiket. Selain itu, aplikasi ini mendukung metode pembayaran QRIS yang memudahkan proses transaksi secara cepat dan aman. Dengan tersedianya informasi jam operasional dan rekomendasi waktu kunjungan wisata, aplikasi ini diharapkan dapat membantu pengguna dalam merencanakan perjalanan dengan lebih baik serta meningkatkan pengalaman berwisata secara keseluruhan.

1.3 Tujuan

Aplikasi Travel.id dikembangkan sebagai solusi digital untuk mengatasi permasalahan dalam proses pemesanan tiket wisata yang masih dilakukan secara konvensional. Aplikasi ini bertujuan untuk memberikan kemudahan, efisiensi, serta kenyamanan bagi pengguna dalam melakukan pemesanan tiket secara online. Dengan memanfaatkan teknologi digital, Travel.id diharapkan mampu meningkatkan kualitas layanan pemesanan tiket serta membantu pengguna dalam merencanakan perjalanan dengan lebih baik.

Secara khusus, tujuan dari pengembangan aplikasi Travel.id adalah sebagai berikut:

1. Mengurangi antrian panjang pada pembelian tiket dengan menyediakan sistem pemesanan tiket secara online yang dapat diakses kapan saja dan di mana saja.
2. Menghemat waktu dan tenaga pengguna melalui proses pemesanan tiket yang lebih praktis dan efisien tanpa harus datang langsung ke lokasi.
3. Meminimalkan risiko kehabisan tiket dengan menampilkan informasi ketersediaan tiket secara real-time.
4. Menyediakan informasi harga tiket yang transparan, sehingga pengguna dapat mengetahui harga secara jelas dan menghindari biaya tambahan yang tidak diperlukan.
5. Mendukung pembayaran digital berbasis QRIS untuk mempermudah proses transaksi yang aman, cepat, dan modern.
6. Menyediakan informasi jam operasional tempat wisata yang akurat agar pengguna dapat mengatur jadwal kunjungan dengan lebih terencana.

7. Menyediakan fitur profil pengguna yang menampilkan data login dan informasi pemesanan sebagai sarana pengelolaan akun secara terintegrasi.

Aplikasi Travel.id diharapkan tidak hanya menjadi sarana pemesanan tiket semata, tetapi juga menjadi solusi digital yang mampu meningkatkan pengalaman pengguna dalam merencanakan dan melakukan perjalanan wisata. Dengan adanya integrasi sistem pemesanan, pembayaran digital, serta penyajian informasi yang lengkap dan akurat, aplikasi ini diharapkan dapat mendukung transformasi digital di bidang pariwisata serta memberikan manfaat jangka panjang bagi pengguna maupun pengelola layanan wisata.

BAB II

ANALISIS DAN PERANCANGAN

2.1 Analisis Kebutuhan Program (Aplikasi)

Pada bagian ini akan membahas dan menganalisa mengenai program atau sistem , menjelaskan tentang alur logika dan antar muka yang digunakan pada aplikasi Travel.id.

A. Analisa kebutuhan aplikasi

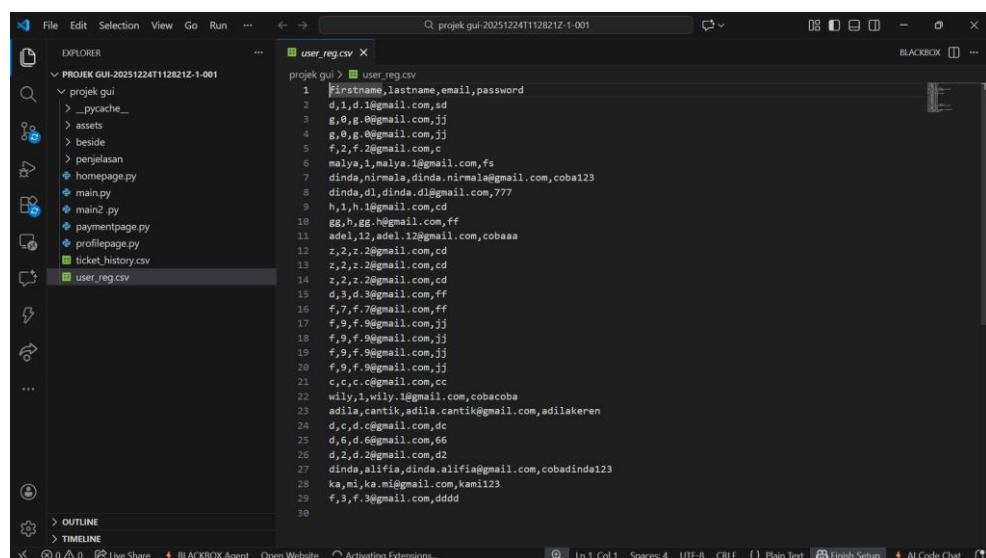
Pada bagian analisa kebutuhan kita bedakan menjadi dua kategori agar dapat memaksimalkan fungsionalitas sistem dan kenyamanan pengguna.

- Kebutuhan Fungsional

Pada bagian ini akan menjelaskan fitur-fitur utama dalam aplikasi Travel.id :

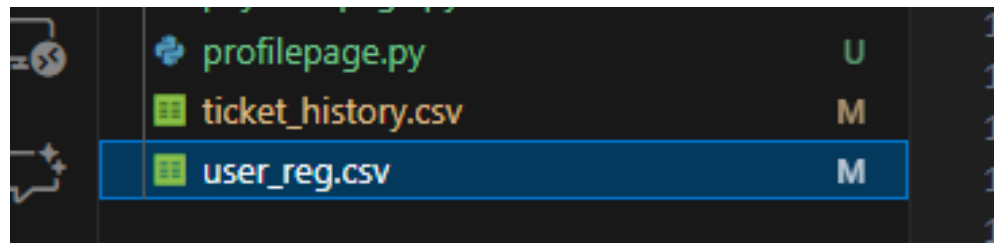
a) Manajemen akun pengguna

Pada saat pertama kali user membuka atau menjalankan program user akan diarahkan ke halaman registrasi untuk mendaftar terlebih dahulu jika belum memiliki akun , tetapi kalau user sudah memiliki akun maka user bisa langsung pindah ke halaman login untuk memasukkan email dan password yang telah di buat.Pada bagian ini aplikasi Travel.id menggunakan file yang bernama user_reg.csv untuk menampung dan menjadi rujukan data untuk memferifikasi pengguna.



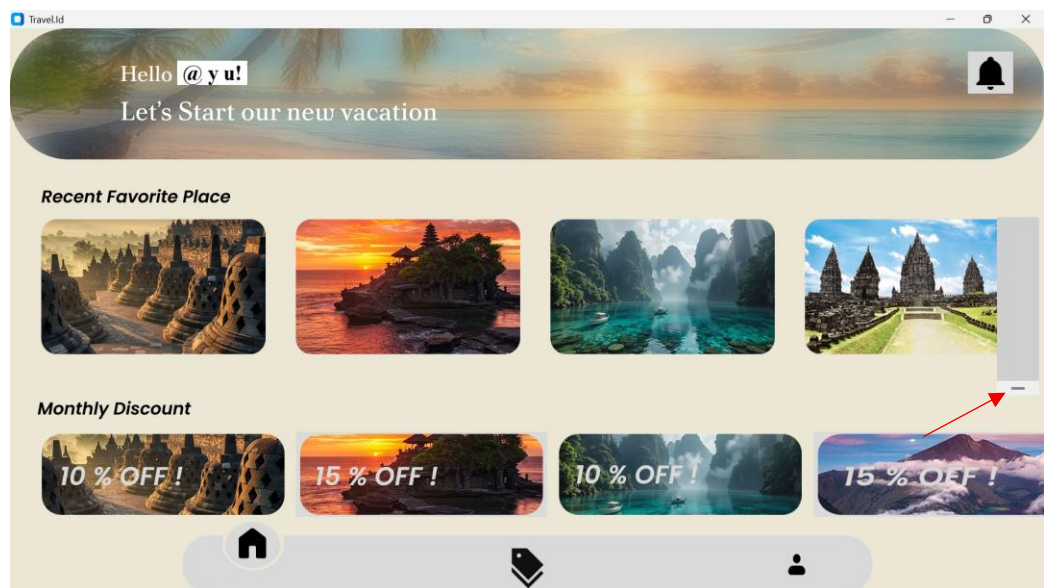
```

1  firstname,lastname,email,password
2  d,1,d.1@gmail.com,sd
3  s,0,s.0@gmail.com,jj
4  s,0,s.0@gmail.com,jj
5  f,2,f.2@gmail.com,c
6  malya,1,malya.1@gmail.com,fs
7  dinda,nirmala,dinda.nirmala@gmail.com,coba123
8  dinda,d1,dinda.d1@gmail.com,777
9  h,1,h.1@gmail.com,cd
10 gg,h,gg.h@gmail.com,ff
11 adel,12,adel.12@gmail.com,cobaaa
12 z,2,z.2@gmail.com,cd
13 z,2,z.2@gmail.com,cd
14 z,2,z.2@gmail.com,cd
15 d,3,d.3@gmail.com,ff
16 f,7,f.7@gmail.com,ff
17 f,9,f.9@gmail.com,jj
18 f,9,f.9@gmail.com,jj
19 f,9,f.9@gmail.com,jj
20 f,9,f.9@gmail.com,jj
21 c,c,c.c@gmail.com,cc
22 wily,1,wily.1@gmail.com,cobacoba
23 adila,cantik,adila.cantik@gmail.com,adilakeren
24 d,c,d.c@gmail.com,dc
25 d,6,d.6@gmail.com,66
26 d,2,d.2@gmail.com,d2
27 dinda,alifia,dinda.alifia@gmail.com,cobadinda123
28 ka,mi,ka.mi@gmail.com,kami123
29 f,3,f.3@gmail.com,dddd
30
  
```

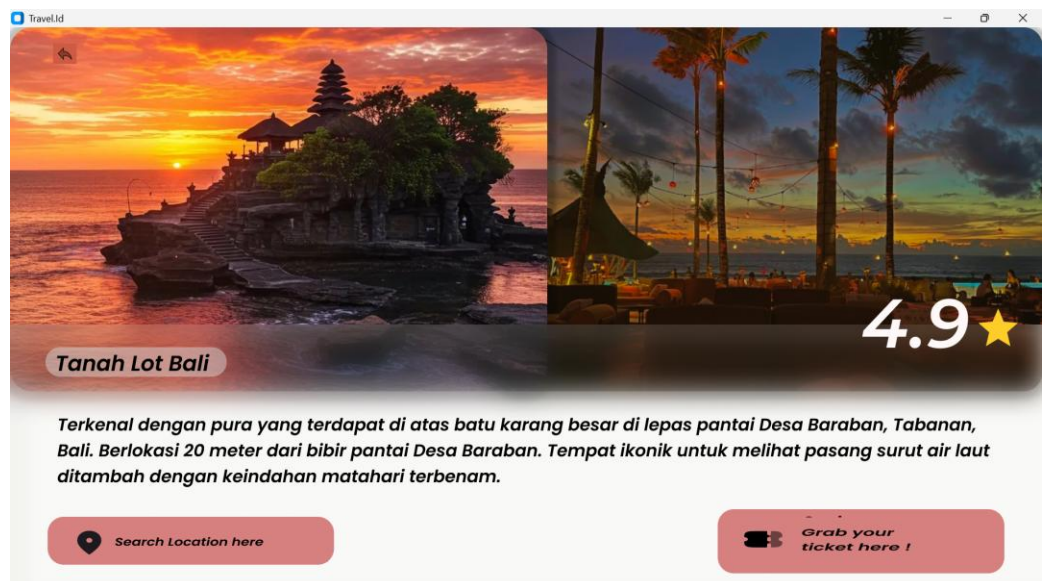


b. Eksplorasi destinasi

Pada bagian home page menyajikan destinasi-destinasi tempat wisata yang bisa di kunjungi, fitur ini didukung oleh area scrollbar untuk memudahkan user dalam pencarian destinasi wisata yakni dengan menggunakan kombinasi antara tkinter dengan CustomTkinter dan menggunakan fungsi grid untuk mengatur tata letak yang ada di dalam tkinter.canvas.



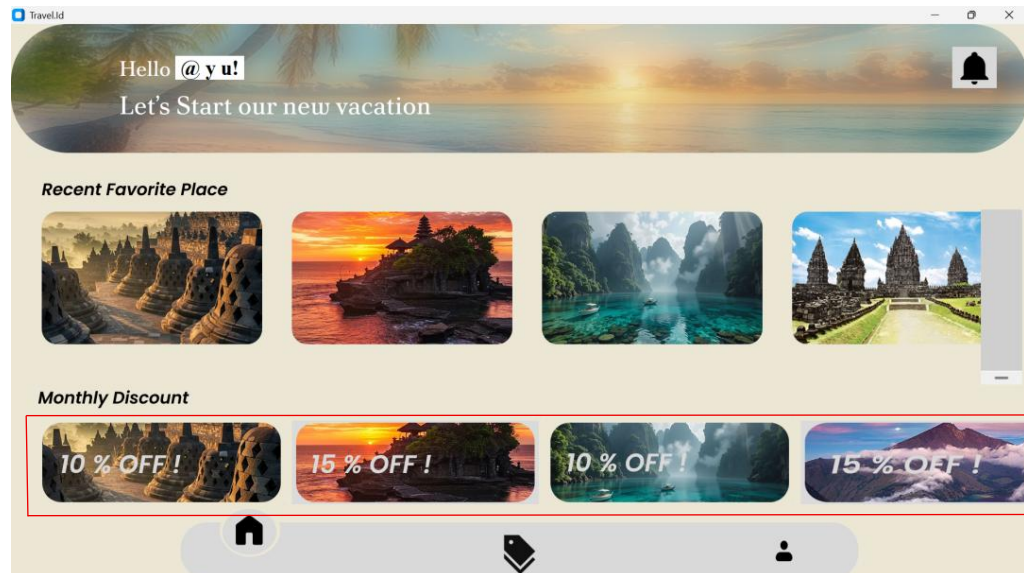
Note: (bagian scrollbar ditunjukkan oleh anak panah).



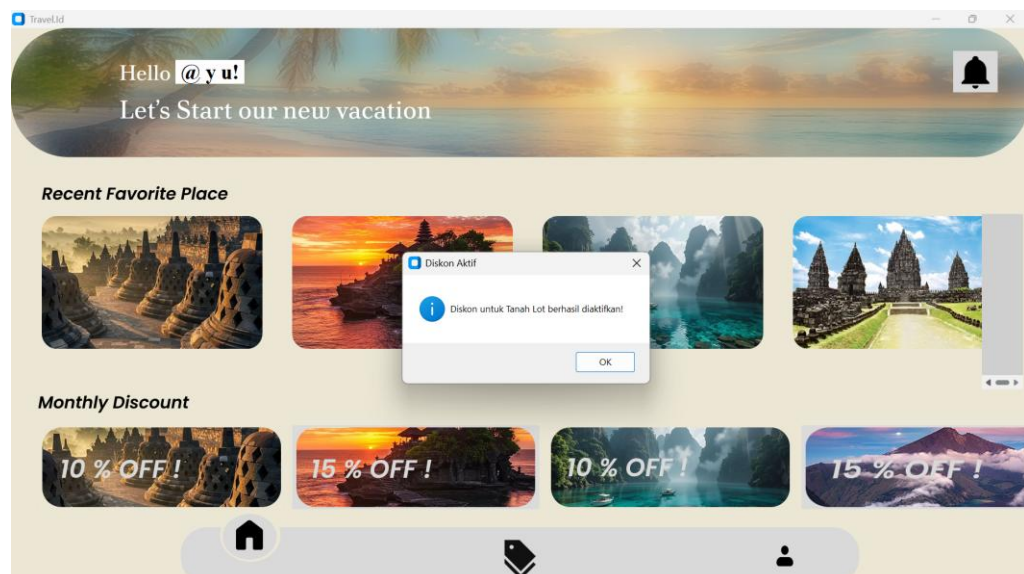
Note: (contoh tampilan ketika user mengeklik salah satu destinasi)

c. Sistem Penawaran diskon

Pada aplikasi Travel.id juga menyediakan penawaran berupa diskon di beberapa destinasi untuk menarik perhatian customer dan meringankan beban biaya bagi pengguna, bagian ini menggunakan sistem klaim untuk mengaktifkan diskon.



Note: menu diskon ada di dalam garis merah



Note: ini adalah tampilan ketika diskon berhasil di aktifkan (diklaim)

d. Sistem pemesanan tiket

User bisa membeli dan memesan tiket di laman paymen page, di bagian ini user bisa memilih tanggal menggunakan menu dateEntry dari library tkinter agar program menjadi lebih interaktif dan juga menggunakan CtkFrame dan CtkLabel untuk menampilkan popup Qris code untuk membayar tiket yang sudah di pesan

Payment

Ringkasan Pesanan

Nama	user 123
Destinasi	Tanah Lot
Jumlah ticket	4
Tanggal	2025-12-25

Rincian Pembayaran

Destinasi	: Tanah Lot
Jumlah tiket	: 4 tiket
Harga/tiket	: Rp 200.000
Diskon	: 15%
Subtotal	: Rp 800.000
Potongan	: - Rp 120.000
Total bayar	: Rp 680.000

Metode

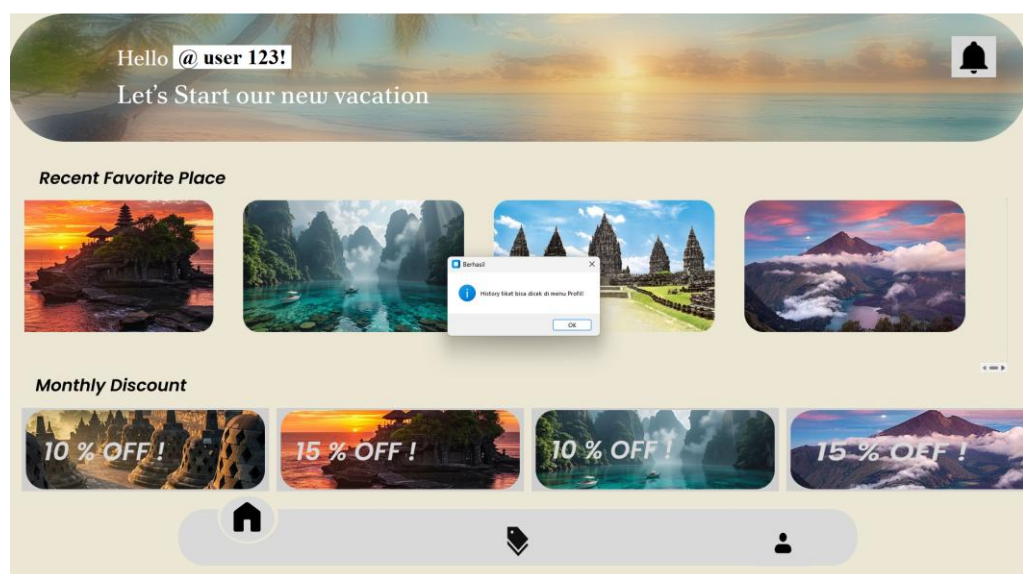
QRIS

Cek Status Pembayaran

Cek

Bottom navigation bar: Home, Payment (active), Profile

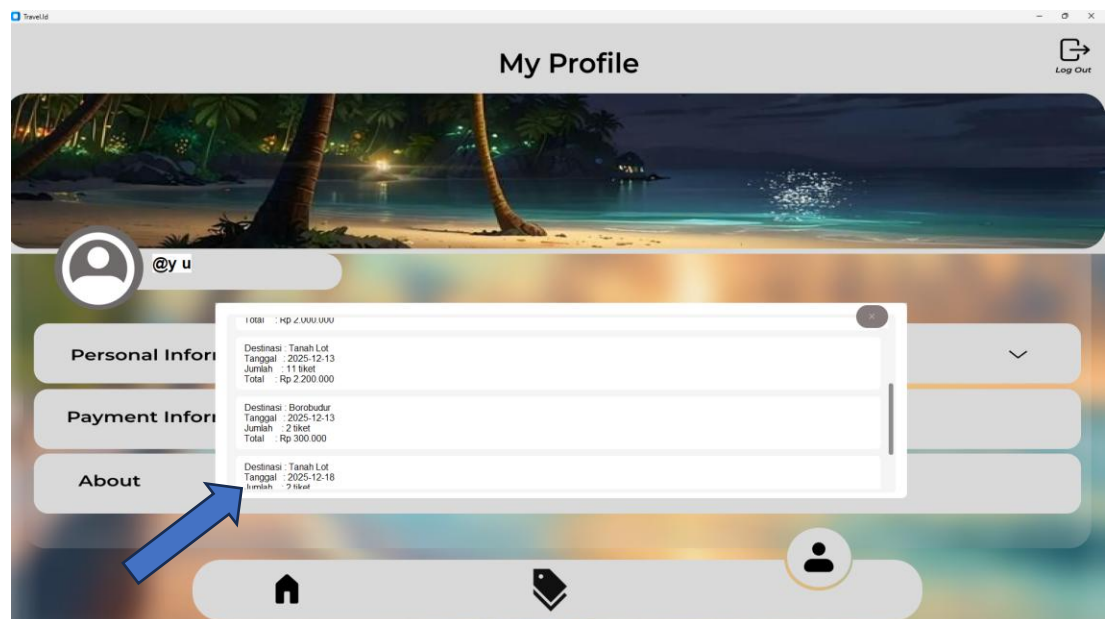
e. Penghitungan rincian Pembayaran



Pada bagian payment page akan menampilkan rincian pembayaran yang telah di inputkan oleh user. dimana nanti setelah tombol cek ditekan, akan muncul pop-up message berupa tiket telah berhasil disimpan

f. Penyimpanan history pembelian tiket

Fitur penyimpanan history dapat di lihat di laman profile page pada button payment information. bagian ini menggunakan komponen customtkinter.CtkScrollableFrame untk menampilkan tiket-tiket yang telah di beli dan mempunyai fitur scroll bar agar bisa mengeksplor riwayat tiket yang telah di beli.history pembelian tiket di simpan di file bernama ticket_history.csv menggunakan metode append untuk menambahkan setiap inputan di baris terakhir.



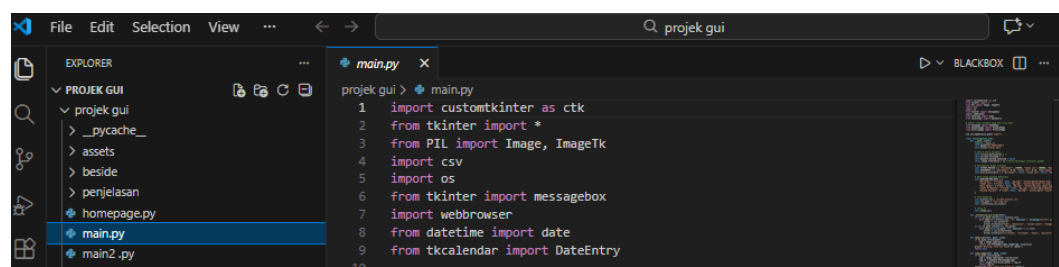
Note: payment information (riwayat pembelian) ditandai dengan anak panah

E. Kebutuhan Non-Fungsional (Pendukung)

Pada bagian ini akan menjelaskan mengenai atribut-atribut pendukung agar aplikasi berjalan sesuai dengan tujuan dan mencapai kualitas yang bagus .

a. GUI (graphical User Interface)

Aplikasi travel.id menggunakan kombinasi dari library Tkinter & CustomTkinter agar bisa menampilkan program yang lebih bagus,interaktif dan lebih modern.



b. Penyimpanan Data

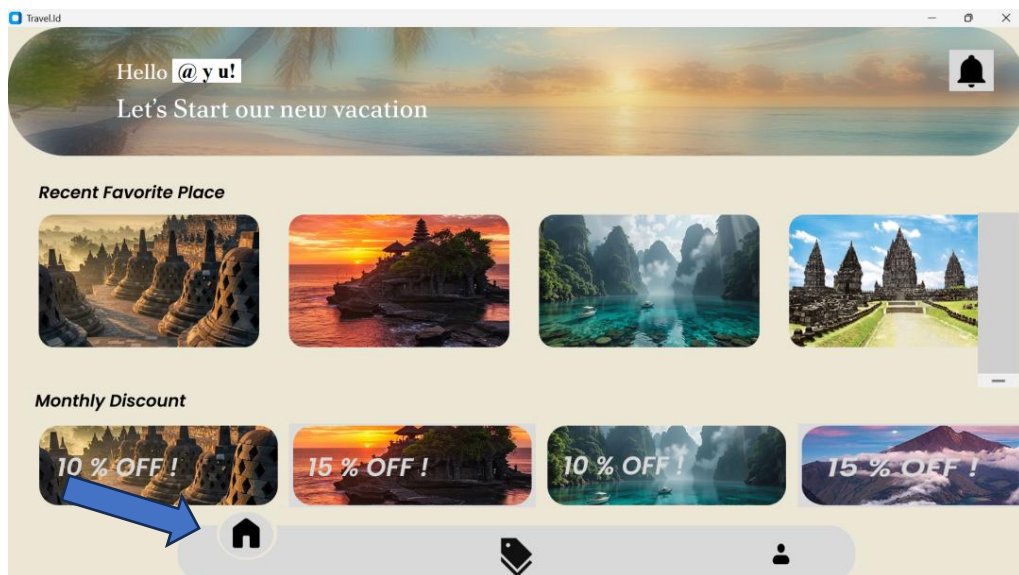
Menggunakan file eksternal .csv agar data tetap tersimpan di dalam folder meskipun aplikasi ditutup (persistence).

homepage	12/24/2025 6:34 PM	Python Source File	5 KB
main	12/24/2025 6:34 PM	Python Source File	20 KB
main2	12/24/2025 6:34 PM	Python Source File	0 KB
paymentpage	12/24/2025 6:34 PM	Python Source File	4 KB
profilepage	12/24/2025 6:34 PM	Python Source File	9 KB
ticket_history	12/24/2025 6:34 PM	Microsoft Excel Com...	2 KB
user_reg	12/24/2025 6:34 PM	Microsoft Excel Com...	1 KB
.git	12/24/2025 6:34 PM	File folder	
beside	12/24/2025 6:34 PM	File folder	
penjelasan	12/24/2025 6:34 PM	File folder	
__pycache__	12/24/2025 6:34 PM	File folder	
assets	12/24/2025 6:34 PM	File folder	

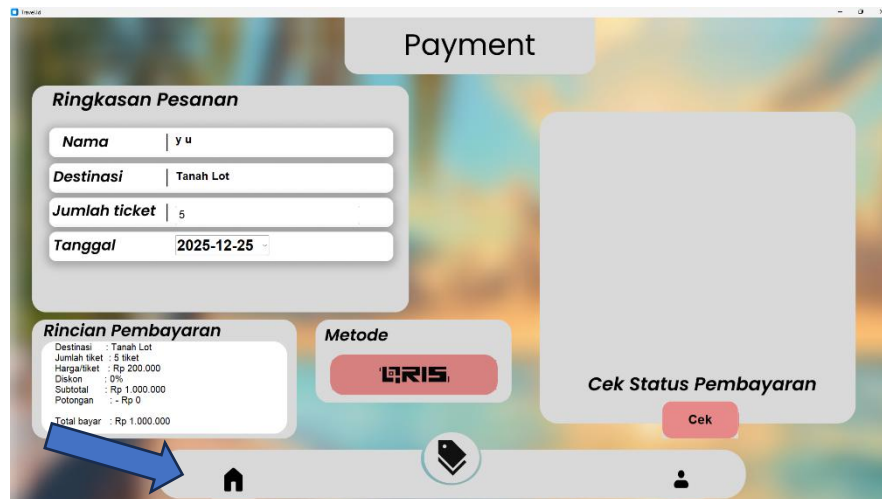
Note:file .csv yang tersimpan di folder ditandai dengan garis merah

c. Navigasi

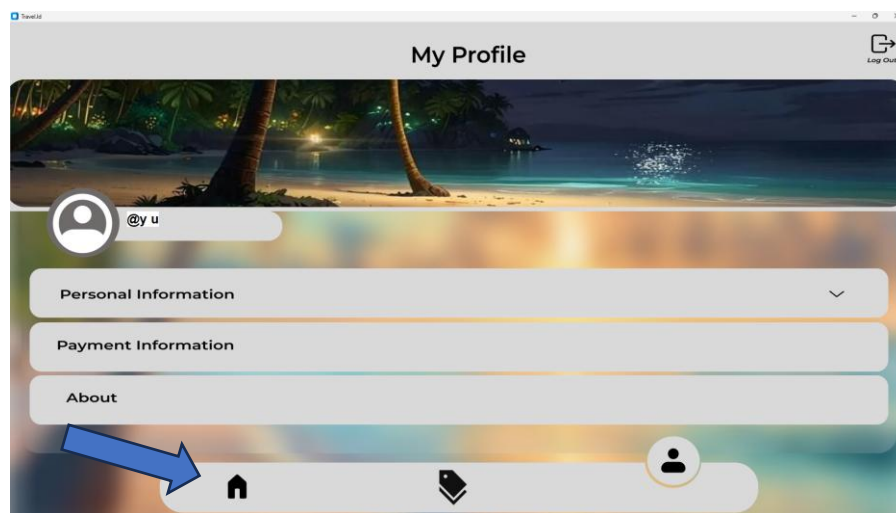
Memberikan fitur taskbar di masing-masing halaman agar memudahkan pengguna untuk berpindah halaman yang ingin di tuju agar lebih efisien.Tombol navigasi berupa Taskbar ini ada di setiap page kecuali di register dan login page.



Note: menu taskbar di Home page ditandai dengan anak panah



Note: menu taskbar di Payment page ditandai dengan anak panah



Note: menu taskbar di Profile page ditandai dengan anak panah

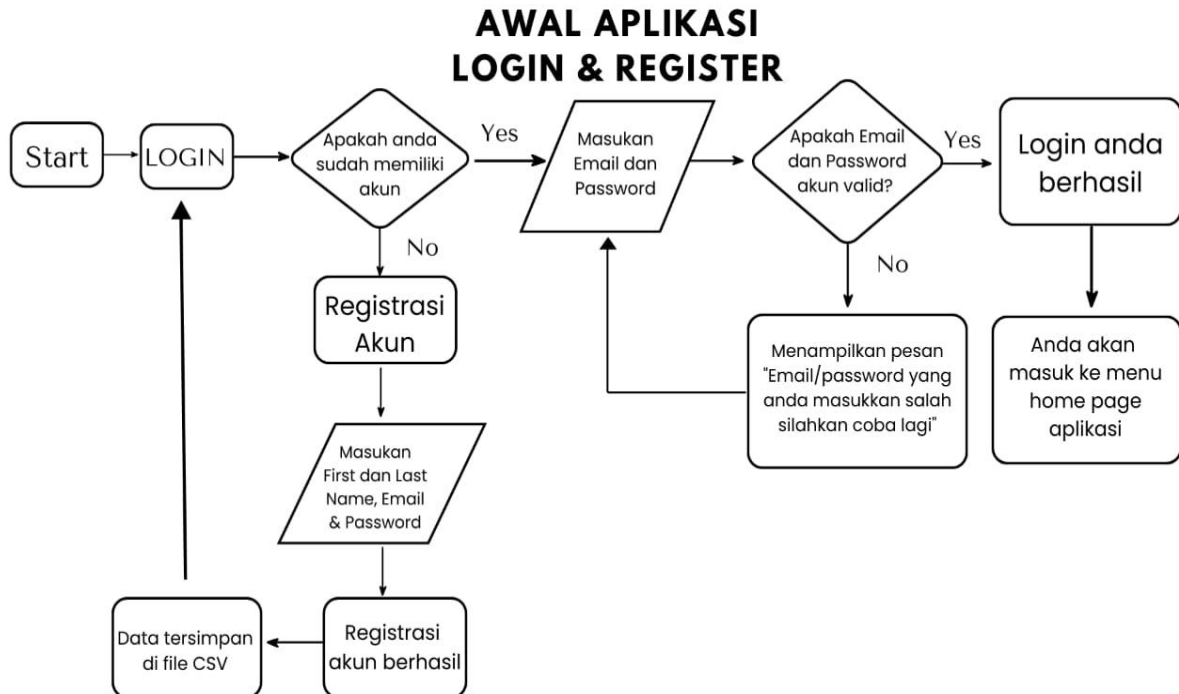
d. Kompabilitas

Aplikasi dirancang untuk berjalan pada sistem operasi Windows/Linux/macOS selama interpreter Python dan pustaka pendukung terinstal.

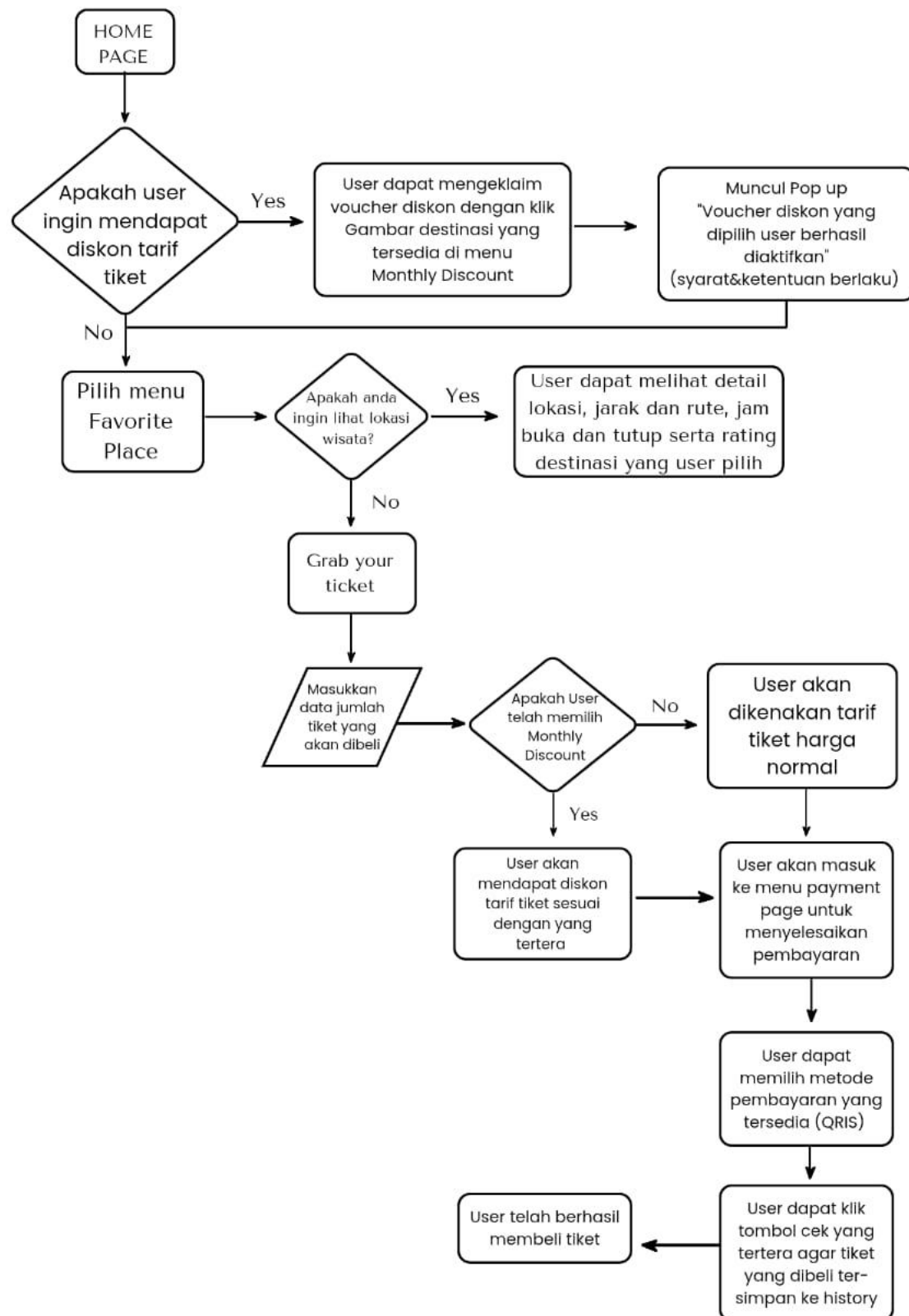


2.2 Diagram Alur atau Flowchart

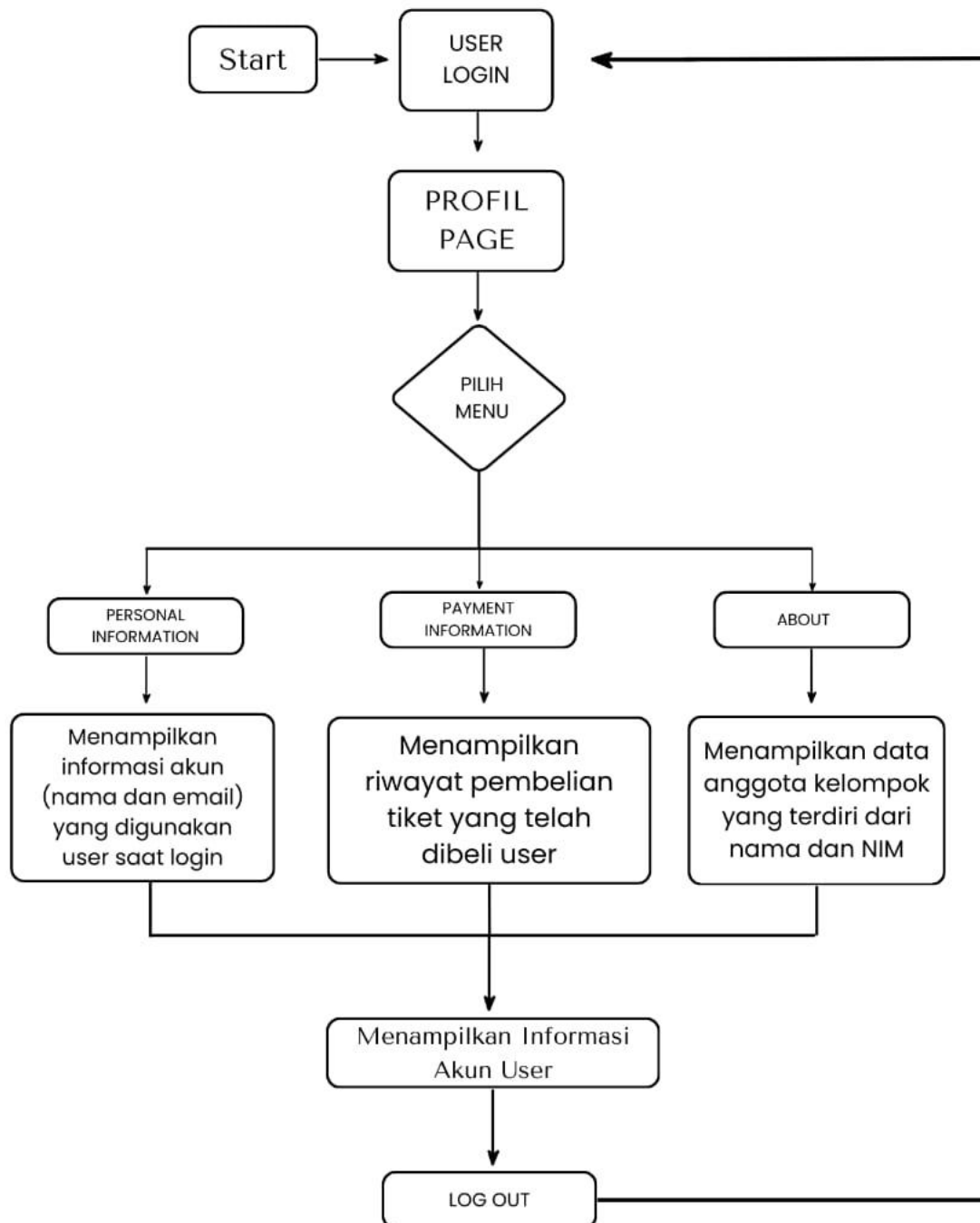
Untuk diagram alur dari program akan dimulai dari Profile page → login page → Home page → Payment page → Profile page.



HOME PAGE & PAYMENT PAGE



PROFILE PAGE



BAB III

IMPLEMENTASI PENJELASAN KODE PROGRAM

3.1 main.py

1. Library yang digunakan

```
import customtkinter as ctk
from tkinter import *
from PIL import Image, ImageTk
import csv
import os
from tkinter import messagebox
import webbrowser
from datetime import date
from tkcalendar import DateEntry

# Import halaman-halaman aplikasi
from homepage import HomePage
from paymentpage import PaymentPage
from profilepage import ProfilePage
```

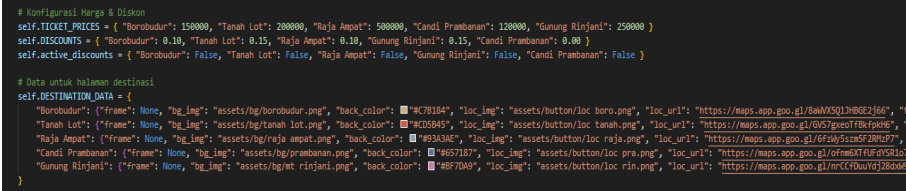
- customtkinter/CTK sebagai aplikasi utama, tempat GUI dijalankan
- tk inter/TK digunakan untuk import aplikasi tambahan sebagai pendukung GUI ketika dijalankan
- PIL digunakan untuk memproses gambar-gambar yang digunakan dalam aplikasi
- CSV, penyimpanan sederhana yang digunakan sebagai tempat penyimpanan data user login dan data pemesanan tiket
- OS, digunakan untuk memastikan aplikasi dapat berjalan diberbagai versi komputer dan memastikan bahwa file csv ada dan bisa diakses
- webbrowser, digunakan untuk akses aplikasi ke peramban laptop untuk user mengakses lokasi tiap destinasi di chrome
- datetime dan tkcalendar, digunakan user ketika mengatur tanggal pembelian tiket
- from homepage, paymentpage, profilepage. digunakan untuk mengimpor tiap page yang dibuat di berbeda agar bisa dipanggil lewat main.py

2.

```
class TravelApp(ctk.CTk):
    def __init__(self):
        super().__init__()
        self.geometry("1920x1080")
        self.title("Travel.Id")

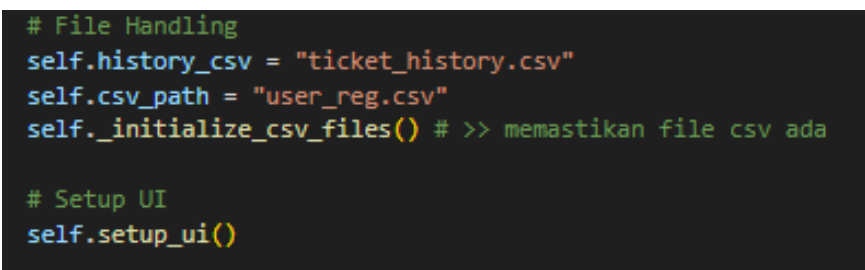
        # Data & State Aplikasi
        self.current_destination = ""
        self.current_fullname = ""
        self.payment_method_selected = False
        self._image_references = {}
```


Class atau OOP utama yang digunakan dalam aplikasi, dengan menggunakan variabel dan list kosong yang digunakan untuk menyimpan data yang akan digunakan dalam aplikasi

3. The screenshot shows a Python class configuration. It defines a dictionary for ticket prices with destinations like Borobudur, Tanah Lot, Raja Ampat, Candi Prambanan, and Gunung Rinjani. It also defines a dictionary for discounts with corresponding values. Finally, it defines a dictionary for destination data, including background images, button icons, and Google Maps URLs for each location.

Bagian ini adalah data utama aplikasi, dimana disini terdapat beberapa informasi yaitu :

- Data Destinasi, dalam sini terdapat button untuk tiap destinasi page yaitu button untuk menuju webbrowser; button pemesanan ticket; dan button untuk kembali ke menu homepage
- Data Ticket Prices, dalam sini terdapat data harga tiket tiap destinasi yang diperinci untuk tiap 1 tiket nya
- Data discount, disini terdapat data discount yang di setting pada beberapa destinasi. Untuk pengaktifan tiap discount nya ini akan dipanggil oleh fungsi `active_discount`
- Di dalam destinasi juga terdapat background image yang telah dikumpulkan dalam folder asset. dimana tiap berbeda destinasi, image yang digunakan juga berbeda

4. The screenshot shows two sections of Python code. The first section, '# File Handling', sets the path for 'ticket_history.csv' and 'user_reg.csv', and calls an initialization function. The second section, '# Setup UI', calls a function to set up the user interface.

Bagian file handling ini digunakan dengan mengimport library csv dan os yang berada di kode paling atas. Fungsinya yaitu :

- Membuat file csv saat program awal diinisiasi
- Memastikan file csv ada dan bisa diakses

5.

```
def register_page(self, parent):
    self.reg_frame = ctk.CTkFrame(parent, width=1920, height=1080)
    self.reg_frame.grid(row=0, column=0, sticky="nsew")
    self.pages['RegisterPage'] = self.reg_frame

    bg_img = self.load_image("assets/bg/register.png", (1920, 1080))
    ctk.CTkLabel(self.reg_frame, image=bg_img, text="").place(x=0, y=0)
    self.entry_firstname = ctk.CTkEntry(self.reg_frame, width=200, height=40, placeholder_text=" ", border_color="#FF8C69", fg_color="#FF8C69", font=("Arial", 20), text_color="Black", corner_radius=2)
    self.entry_firstname.place(x=100, y=400)
    self.entry_lastname = ctk.CTkEntry(self.reg_frame, width=200, height=40, placeholder_text=" ", border_color="#FF8C69", fg_color="#FF8C69", font=("Arial", 20), text_color="Black", corner_radius=2)
    self.entry_lastname.place(x=100, y=440)
    self.entry_email = ctk.CTkEntry(self.reg_frame, width=300, height=40, placeholder_text=" ", border_color="#D9534F", fg_color="#D9534F", font=("Helvetica", 20), text_color="Black", corner_radius=2)
    self.entry_email.place(x=100, y=480)
    self.entry_password = ctk.CTkEntry(self.reg_frame, width=300, height=40, placeholder_text=" ", border_color="#D9534F", fg_color="#D9534F", font=("Helvetica", 20), text_color="Black", corner_radius=2)
    self.entry_password.place(x=100, y=520)
```

Page pertama dibuat, yaitu register page dimana beberapa komponen CTK yang digunakan dalam page ini yaitu Ctk Frame, Ctk Button, dan Ctk Image. Dalam tiap page akan memiliki Frame, Button, dan Image yang berbeda

6.

```
def update_email(*args):
    first = self.entry_firstname.get().strip().lower()
    last = self.entry_lastname.get().strip().lower()
    if first and last:
        self.entry_email.delete(0, "end")
        self.entry_email.insert(0, f"{first}.{last}@gmail.com")
    self.entry_firstname.bind("<KeyRelease>", update_email)
    self.entry_lastname.bind("<KeyRelease>", update_email)
```

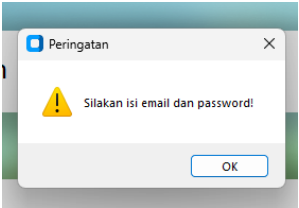
Fungsi update email ini, berfungsi dalam mengupdate otomatis email sesuai first & last email yang disusun menjadi email yang nantinya akan diisi oleh user

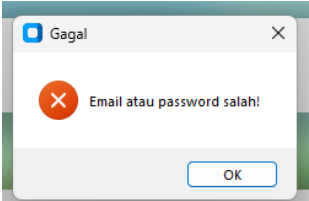
7.

```
def login_page(self, parent):
    self.login_frame = ctk.CTkFrame(parent, width=1920, height=1080)
    self.login_frame.grid(row=0, column=0, sticky="nsew")
    self.pages['LoginPage'] = self.login_frame

    bg_img = self.load_image("assets/bg/login.png", (1920, 1080))
    ctk.CTkLabel(self.login_frame, image=bg_img, text="").place(x=0, y=0)
    self.login_email = ctk.CTkEntry(self.login_frame, width=600, height=40, placeholder_text=" ", border_color="#D9534F", fg_color="#D9534F", font=("Helvetica", 20), text_color="Black", corner_radius=2)
    self.login_email.place(x=100, y=400)
    self.login_password = ctk.CTkEntry(self.login_frame, width=600, height=40, placeholder_text=" ", border_color="#D9534F", fg_color="#D9534F", font=("Helvetica", 20), text_color="Black", corner_radius=2)
    self.login_password.place(x=100, y=440)
    self.back_button = ctk.CTkButton(self.login_frame, width=100, height=40, image=self.icon_back, text="", fg_color="transparent", bg_color="#D9534F", text_color="white", corner_radius=2)
    self.back_button.place(x=100, y=480)
    self.login_button = ctk.CTkButton(self.login_frame, width=100, height=40, text="Log in", fg_color="#D9534F", border_color="#D9534F", text_color="white", corner_radius=2)
    self.login_button.place(x=100, y=520)
```

Page kedua dibuat, yaitu login page yang juga terdapat beberapa komponen CTK. beda dengan register page yang mempunyai command untuk menyimpan user pada CSV di login page ini akan mencocokkan data yang user masukkan dengan data yang ada pada CSV dimana terdapat beberapa sub-fungsi :

- 
 ketika user tidak memasukkan email/password

- 
 ketika user salah menginput

8.

```
def _create_navigation_buttons(self, parent_frame, active_page: str):
    # Hanya tampilkan tombol navigasi jika halaman tersebut TIDAK sedang aktif
    if active_page != "home":
        ctk.CTkButton(parent_frame, width=10, height=10, image=self.icon_home, text="", fg_color="#D9D9D9", bg_color="#D9D9D9")

    if active_page != "pay":
        ctk.CTkButton(parent_frame, width=75, height=75, image=self.icon_pay, text="", fg_color="#D9D9D9", bg_color="#D9D9D9")

    if active_page != "profile":
        ctk.CTkButton(parent_frame, width=30, height=30, image=self.icon_profil, text="", fg_color="#D9D9D9", bg_color="#D9D9D9")
```

Pada bagian ini terdapat task bar yang berfungsi sebagai perpindahan UI dari satu page ke page lainnya. Task bar disini menggunakan componen pop up frame yang fungsinya membantu user ketika hendak berpindah page. Semisal dari home page ke payment page, lalu ke profile page dan banyak perpindahan lainnya. Bagian ini juga membuat agar tiap page hanya dapat berpindah ke page lain tapi tidak akan mereset page itu sendiri



9.

```
def set_username(self, first, last):
    self.current_fullname = f"{first} {last}"
    if hasattr(self, 'username_label'):
        self.username_label.configure(text=f"@ {first} {last}!")
    if hasattr(self, 'profile_name_label'):
        self.profile_name_label.configure(text=f"@{first} {last}")
```

Pada bagian ini mendefinisikan bagaimana username terbentuk pada saat page register tadi. Dimana nanti, username akan diperlukan pada bagian home dan profile page



3.2 Homepage.py

10.

```
import customtkinter as ctk
from tkinter import Canvas, Scrollbar, Frame
from PIL import Image

class HomePage(ctk.CTkFrame):
    def __init__(self, master, controller):
        super().__init__(master)
        self.controller = controller
```

Page ke 3 dibuat, dimana page ini dibuat dalam kode .py yang berbeda yaitu di homepage.py. Dalam bagian ini akan mendefinisikan bagian” terkhusus yang hanya digunakan dalam homepage saja. Sehingga library yang diimpor juga hanya digunakan dalam homepage saja.

11.

```
def _create_destination_scroll(self):
    #membuat area scroll untuk destinasi
    scroll_bar = ctk.CTkFrame(self, width=1920, height=300, fg_color="#ece7d5")
    scroll_bar.place(x=30, y=350)

    canvas_scroll = Canvas(scroll_bar, width=1800, height=330, bg="#ece7d5", highlightthickness=0)
    canvas_scroll.pack(side="left", fill="both", expand=True)

    scrollbar = Scrollbar(scroll_bar, orient="horizontal", command=canvas_scroll.xview)
    scrollbar.pack(side="bottom", fill="x")
    canvas_scroll.configure(xscrollcommand=scrollbar.set)

    scroll_frame = Frame(canvas_scroll, bg="#ece7d5")
    canvas_scroll.create_window((0, 0), window=scroll_frame, anchor="nw")
```

- fungsi ini yang membuat frame scroll-able pada bagian homepage. dimana disini menggunakan library TK yaitu scroll_bar



12.

```
def hitung_total_setelah_diskon(self, dest, qty):
    harga = self.TICKET_PRICES.get(dest, 0)
    diskon = self.DISCOUNTS.get(dest, 0) if self.active_discounts.get(dest, False) else 0
    subtotal = harga * qty
    potongan = subtotal * diskon
    total = subtotal - potongan
    return harga, diskon, subtotal, potongan, total

def update_rincian_pembayaran(self):
    dest = self.current_destination or "-"
    try:
        qty = int(self.jumlah_var.get()) if self.jumlah_var.get() else 0
    except ValueError:
        qty = 0

    harga, diskon, subtotal, potongan, total = self.hitung_total_setelah_diskon(dest, qty)
    teks = (f"Destinasi      : {dest}\n"
           f"Jumlah tiket      : {qty} tiket\n"
           f"Harga/tiket       : {self.format_rupiah(harga)}\n"
           f"Diskon            : {int(diskon * 100)}%\n"
           f"Subtotal          : {self.format_rupiah(subtotal)}\n"
           f"Potongan          : - {self.format_rupiah(potongan)}\n\n"
           f"Total bayar       : {self.format_rupiah(total)}")
    if hasattr(self, 'label_rincian'):
        self.label_rincian.configure(text=teks)
```

disini terdapat fungsi untuk penghitungan tiket yang dilakukan di home page menuju ke payment page. Dikarenakan bentuk def maka fungsi ini harus di panggil dalam bentuk command pada sebuah button. Command def ini diletakkan pada bagian Grab Ticket yang ada pada tiap destinasi



3.3 Payment Page.py

13.

```
import customtkinter as ctk
from tkinter import StringVar
from tkcalendar import DateEntry
from datetime import date

class PaymentPage(ctk.CTkFrame):
    def __init__(self, master, controller):
        super().__init__(master)
        self.controller = controller
```

sama seperti homepage, pada page yang ke 3 ini kita juga mendefinisikan lagi bagian payment page yang nantinya akan di panggil di main.py. Disini, terdapat import library lain pada tkinter yaitu stringvar karena pada payment page user diminta mengisi banyak tiket yang ingin dibeli

14.

```
# Widget-widget ini dibuat di sini, tapi nilainya akan diubah oleh controller
self.label_nama_value = ctk.CTkLabel(self, text="", font=("Helvetica", 24, "bold"), text_color="black", bg_color="FFFFFF")
self.label_nama_value.place(x=360, y=250)
self.controller.label_nama_value = self.label_nama_value

self.label_dest_value = ctk.CTkLabel(self, text="", font=("Helvetica", 24, "bold"), text_color="black", bg_color="FFFFFF")
self.label_dest_value.place(x=360, y=330)
self.controller.label_dest_value = self.label_dest_value

self.jumlah_var = StringVar(value="1")
self.entry_jumlah_pay = ctk.CTkEntry(self, width=400, height=40, font=("Helvetica", 24), fg_color="white", border_color="FFFFFF", text_color="black", text="")
self.entry_jumlah_pay.place(x=360, y=410)
self.controller.jumlah_var = self.jumlah_var

self.date_entry = DateEntry(self, width=10, height=60, background="385FAA", foreground="white", borderwidth=2, date_pattern="yyyy-mm-dd", font=("ITC Avant Garde Gothic", 20))
self.date_entry.place(x=360, y=473)
self.date_entry.set_date(date.today())
self.controller.date_entry = self.date_entry

# --- Rincian Pembayaran ---
self.label_rincian = ctk.CTkLabel(self, text="", width=535, height=200, font=("ITC Avant Garde Gothic", 20), text_color="black", fg_color="FFFFFF", bg_color="FFFFFF")
self.label_rincian.place(x=60, y=785)
self.controller.label_rincian = self.label_rincian

# --- Bind & Tombol ---
# Setiap kali jumlah tiket diubah, panggil fungsi update di controller
self.jumlah_var.trace_add("write", lambda *a: self.controller.update_rincian_pembayaran())
```

Pada bagian ini, akan mengambil data yang user berikan atau user klik ketika melakukan check out dari homepage. Dari sinilah payment page akan menginisiasi dan menampilkan data yang diambil. Semisal user check out di destinasi tanah lot, maka di payment page juga akan menampilkan tanah lot

Ringkasan Pesanan	
Nama	user 123
Destinasi	Tanah Lot
Jumlah tiket	1
Tanggal	2025-12-25

Rincian Pembayaran	
Destinasi	: Tanah Lot
Jumlah tiket	: 1 tiket
Harga/tiket	: Rp 200.000
Diskon	: 0%
Subtotal	: Rp 200.000
Potongan	: - Rp 0
Total bayar	: Rp 200.000

15.

```
def simpan_tiket_ke_history(self):
    if not self.current_fullname:
        messagebox.showerror("Error", "User belum login.")
        return
    if not self.current_destination:
        messagebox.showerror("Error", "Silakan pilih destinasi terlebih dahulu.")
        return
    if not self.payment_method_selected:
        messagebox.showinfo("Info", "Selesaikan pembayaran dahulu dengan memilih metode pembayaran.")
        return
    try:
        qty = int(self.jumlah_var.get())
        if qty < 1: raise ValueError()
    except ValueError:
        messagebox.showerror("Error", "Jumlah tiket harus berupa angka dan minimal 1.")
        return
    harga, total = self.hitung_total_setelah_diskon(self.current_destination, qty)
    with open(self.history_csv, "a", newline="", encoding="utf-8") as f:
        csv.writer(f).writerow([self.current_fullname, self.current_destination, qty, self.date_entry.get(), harga, int(total)])
    if self.current_destination in self.active_discounts:
        self.active_discounts[self.current_destination] = False
```

Bagian ini adalah bagian terpenting dalam penyimpanan tiket ke history_csv. Disini payment page akan menyimpan tiket yang telah user check out dan menyimpannya melalui history_csv dengan di dalamnya akan terdapat info” yang sesuai dengan tiket yang sudah di check out

```
Willy Surya, Gunung Kinjani, 8, 2025-12-25, 250000, 127500
user 123, Tanah Lot, 1, 2025-12-25, 200000, 200000
```

16.

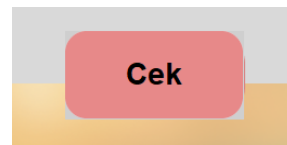
```
def show_qris_on_panel(self):
    if hasattr(self, 'qris_popup_frame'):
        self.qris_popup_frame.place(x=1173, y=224)
        self.qris_popup_frame.lift()
        self.payment_method_selected = True
```

Pada bagian ini memuat fitur show pop up frame yang berisi qris pembayaran nantinya akan di scan olehh user sebagai bentuk checkout akhir pembayaran ticket destinasi.



```
17. ctk.CTkButton(self, text="Cek", width=165, height=81, font=("Helvetica", 28, "bold"),
      fg_color=■ "#E78989", hover_color=■ "#D46F6F", text_color="black", corner_radius=20,
      command=self.controller.simpan_tiket_ke_history).place(x=1418, y=835)
```

Fungsi terakhir payment page yaitu button letak command penyimpanan tiket history, yaitu button cek yang menjadi langkah terakhir dalam proses checkout tiket



3.4 Profile page,py

```
18. import customtkinter as ctk
    import csv
    import os

    class ProfilePage(ctk.CTkFrame):
        def __init__(self, master, controller):
            super().__init__(master)
            self.controller = controller
```

di dalam profile page ini tetap sama, juga terdapat library yang nanti digunakan di dalam fitur-fitur yang nanti akan digunakan dalam profile page. Disini fitur utamanya yaitu csv dan os, dimana dalam profile page akan menampilkan history pembelian tiket oleh pengguna, dan data username serta email pengguna.

19.

```
def _create_popups(self):
    # Pop-up Personal Info
    self.profile_info_frame = ctk.CTkFrame(self, width=1200, height=400, fg_color="#FFFFFF", corner_radius=10)
    self.personal_info_list = ctk.CTkFrame(self.profile_info_frame, fg_color="transparent")
    self.personal_info_list.pack(fill="both", expand=True, padx=30, pady=30)
    ctk.CTkButton(self.profile_info_frame, text="X", width=40, height=40, fg_color="#867B7B", hover_color="#534949")

    # Pop-up Payment History
    self.payhistory_frame = ctk.CTkFrame(self, width=1200, height=360, fg_color="#FFFFFF", corner_radius=10)
    self.history_list = ctk.CTkScrollableFrame(self.payhistory_frame, width=1150, height=300, fg_color="#F5F5F5")
    self.history_list.pack(fill="both", expand=True, padx=20, pady=(20, 10))
    ctk.CTkButton(self.payhistory_frame, text="X", width=40, height=40, fg_color="#867B7B", hover_color="#534949")

    # Pop-up Info Kelompok
    self.kelompok_info_frame = ctk.CTkFrame(self, width=800, height=450, fg_color="white", corner_radius=15, border_color="black", border_width=1)
    ctk.CTkLabel(self.kelompok_info_frame, text="Informasi Anggota Kelompok", font=("Helvetica", 24, "bold"), text_color="black").pack(pady=10)
    self.kelompok_list = ctk.CTkScrollableFrame(self.kelompok_info_frame, width=1150, height=300, fg_color="#F5F5F5")
    self.kelompok_list.pack(padx=20, pady=10)
    ctk.CTkButton(self.kelompok_info_frame, text="Tutup", width=150, fg_color="#E74C3C", hover_color="#C0392B", text_color="white").pack(pady=10)
    self.setup_kelompok_info()
```

Disini terdapat 3 fitur utama profile page, yaitu :

- Profile Info : disini menampilkan info login user

Nama Lengkap : user 123

Email : user.123@gmail.com

- Payment info : disini berisi history pembelian tiket

Destinasi : Tanah Lot
Tanggal : 2025-12-25
Jumlah : 1 tiket
Total : Rp 200.000

- About info : berisi informasi developer

Informasi Anggota Kelompok

Anggota Kelompok: 10

1. Nama Lengkap : Yuan Fasich Tansatrisna
NIM : 25031554155

2. Nama Lengkap : Wiliyan Surya
NIM : 25031554168

3. Nama Lengkap : Dinda Alifia Eka Nirmala
NIM : 25031554257

Tutup

20.

```
logout_icon = self.controller.load_icon("assets/icon/logout.png", (76, 76))
ctk.CTkButton(self, image=logout_icon, text="", fg_color="#D9D9D9", hover_color="#657187")
```


Fitur terakhir yaitu logout, yang fungsinya mengembalikan user pada halaman registrasi atau halaman pertama user mendaftar akun



DAFTAR PUSTAKA

- Amos, D. (2024). *Python GUI Programming: Your Tkinter Tutorial*. Real Python. <https://customtkinter.tomschimansky.com/>. Last Access 25 Desember 2025.
- F, L. (2025). *Custom Tk-Inter, a modern and customizable python UI-Library based on Tkinter*. Python Software Foundation. <https://docs.python.org/3/library/tkinter.html>. Last Access 25 Desember 2025.
- Foundation, P. S. (2025). *tkinter-python interface to Tcl/Tk*. <https://python-pillow.org/>. Last Access 25 Desember 2025.
- Geeks, G. f. (2025). *tkcalendar 1.6.1*. <https://pypi.org/project/tkcalendar/>. Last Access 25 Desember 2025.
- team, T. P. (2025). *csv - CSV File Reading and Writing*. <https://realpython.com/python-gui-tkinter/>. Last access 25 Desember 2025.