

LAPORAN Pengerjaan Ujian Akhir Semester (UAS)
PEMOGRAMAN DASAR
SISTEM PENDATAAN KEANGGOTAAN GYM
Semester Gasal Tahun Akademik 2025/2026



By :

Akeyla Fadillah Alfath Herliyanantha (25031554074)

Nabil Aryo Suharnoto (25031554092)

Adinda Atsilla Salsabilah (25031554262)

Lecturer:

Hasanuddin Al-Habib, S.Si., M.Si

Dr. Heri Purnawan, S.Si., M.Si.

STUDY PROGRAM DATA SCIENCE
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
STATE UNIVERSITY OF SURABAYA
2025

DAFTAR ISI

DAFTAR ISI	1
BAB I.....	4
PENDAHULUAN.....	4
1.1 LATAR BELAKANG.....	4
1.2 RUMUSAN MASALAH	4
1.3 TUJUAN.....	5
BAB II	6
ANALISIS DAN PERANCANGAN.....	6
2.1 ANALISIS KEBUTUHAN SISTEM	6
2.2 PERANCANGAN DATABASE (CSV).....	6
2.3 ALUR LOGIKA (<i>FLOWCHART</i>)	7
SBAB 3	7
IMPLEMENTASI	7
3.1 PENJELASAN KODE PROGRAM.....	8
3.1.1 Import Library dan Konfigurasi Path (Baris 1-10):.....	8
Importing Libraries:.....	8
Konfigurasi	9
3.1.2 Fungsi Keamanan Aset (Safe Open), (Baris 13-22):	9
Fungsi <code>safe_open(path, size=None)</code>	9
3.1.3 Inisialisasi Database CSV (Baris 24-30):	9
Fungsi <code>init_csv()</code> :.....	10
3.1.4 Fungsi Dasar CRUD (Create, Read, Update), (Baris 32-67):	11
Fungsi <code>read_users</code> & <code>load_user_saldo</code>	11
Fungsi <code>write_user</code> :	12
3.1.5 Fungsi Transaksi Kompleks (Baris 69-87).....	12
Fungsi <code>save_user_data</code> :.....	12
3.1.6 Algoritma Tanggal dan Utilitas UI (Baris 89-114):.....	13
Dictionary <code>DURASI_MEMBERSHIP</code> & Fungsi <code>hitung_join_date_expired</code> :	13
Fungsi <code>center_window</code> :.....	14
3.1.7 Inisialisasi Antarmuka Login (Baris 119-169).....	15
Class <code>LoginApp</code> - Method <code>__init__</code> :.....	16

3.1.8 Logika Validasi (Baris 171-188).....	16
Fungsi <code>login(self, event=None):</code>	16
3.1.9 Navigasi Jendela (Window Navigation), (Baris 190-214)	17
Fungsi <code>open_register & open_info:</code>	17
3.1.10 Class Info Kelompok (Baris 216-257)	18
Class <code>InfoKelompok:</code>	18
3.1.11 Logika Pendaftaran Akun Register, (Baris 259-338)	19
Class <code>RegisterApp</code> & Fungsi <code>register:</code>	21
3.1.12 Inisialisasi Dashboard Utama (Main Menu), (Baris 340-402)	22
Class <code>GymMenu</code> - Method <code>__init__</code> :	23
3.1.13 Fungsi Navigasi dan Pembaruan Tampilan, (Baris 404-510)	24
Fungsi <code>open_profile, update_saldo, logout, open_topup:</code>	26
3.1.14 Logika Transaksi Pembelian Membership, (Baris 512-523)	26
Fungsi <code>buy_item(self, item, price):</code>	26
3.1.15 Logika Perhitungan Status Membership (Baris 525-542)	27
Fungsi <code>load_expired & get_remaining_days:</code>	27
3.1.16 Inisialisasi Jendela Top Up dan Logika Penambahan Saldo (Top Up) (Baris 544-590)....	28
Class <code>TopUpApp</code> - Method <code>__init__</code> :	28
Fungsi <code>topup(self):</code>	29
3.1.17 Tampilan Profil dan Pengambilan Data Spesifik (Baris:591-637)	30
Class <code>ProfilePage</code> - Method <code>__init__</code> :	30
3.1.18 Logika Sisa Hari dan Logout (Baris 639-653).....	31
Fungsi <code>get_remaining_days:</code>	31
Fungsi <code>logout:</code>	31
3.1.19 Eksekusi Program Utama (Main Program), (Baris 655-670)	32
Konfigurasi Tema & Inisialisasi Root:.....	32
Pre-loading Jendela & Mainloop:.....	33
3.1.20 Fitur Tambahan: Generator QR Code dan Exit Handler	33
Fungsi Generator QR Code	33
Fungsi <code>generate_qr(username, expired) :</code>	33
Fungsi Protokol Exit Application.....	34
3.2 TAMPILAN APLIKASI (<i>SCREENSHOT</i>)	35

BAB 4	36
PENUTUP.....	36
4.1 KESIMPULAN.....	36
4.2 SARAN PENGEMBANGAN.....	36
DAFTAR PUSTAKA	37

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Perkembangan teknologi informasi saat ini telah merambah ke berbagai sektor, termasuk industri kesehatan dan kebugaran. Penggunaan perangkat lunak untuk manajemen data menjadi hal yang krusial guna meningkatkan efisiensi operasional suatu organisasi. Namun, fakta di lapangan menunjukkan masih banyak pusat kebugaran (*gym*) skala menengah yang mengelola data keanggotaan secara manual menggunakan kertas atau pembukuan sederhana.

Metode pencatatan manual memiliki risiko tinggi terhadap kesalahan manusia (*human error*), seperti hilangnya data anggota, kesalahan perhitungan saldo, hingga ketidakakuratan dalam menentukan masa berlaku keanggotaan. Sistem manual sering kali tidak efektif dalam menangani volume data yang terus bertambah seiring waktu. Oleh karena itu, diperlukan sebuah sistem terkomputerisasi yang dapat menangani pendaftaran, login, transaksi saldo, serta perhitungan masa aktif keanggotaan secara otomatis.

Proyek ini mengembangkan aplikasi "Sistem Pendataan Keanggotaan Gym" (GymEmas), sebuah sistem berbasis desktop yang dibangun menggunakan bahasa pemrograman Python. Aplikasi ini dirancang untuk mengatasi permasalahan administrasi gym dengan fitur pencatatan tanggal bergabung (*join_date*) dan tanggal kadaluarsa (*expired date*) secara *real-time*, sehingga transparansi data antara pengelola dan anggota dapat terjamin.

1.2 RUMUSAN MASALAH

Berdasarkan latar belakang tersebut, rumusan masalah dalam proyek ini adalah:

- Bagaimana merancang aplikasi berbasis GUI (*Graphical User Interface*) yang interaktif dan mudah digunakan untuk manajemen keanggotaan gym?
- Bagaimana mengimplementasikan penyimpanan data persisten menggunakan format CSV yang mampu mencatat riwayat keanggotaan?
- Bagaimana membangun logika pemrograman untuk menghitung masa aktif keanggotaan secara otomatis berdasarkan paket yang dipilih?

1.3 TUJUAN

Tujuan dari pembuatan aplikasi ini adalah:

- Menyediakan *platform* digital untuk mempermudah proses registrasi dan login anggota gym.
- Menggantikan sistem perhitungan manual dengan algoritma otomatis untuk manajemen saldo dan penentuan tanggal kadaluarsa (*expired date*).
- Menghasilkan sistem pendataan yang akurat, aman, dan mudah diakses melalui komputer desktop.

BAB II

ANALISIS DAN PERANCANGAN

2.1 ANALISIS KEBUTUHAN SISTEM

Untuk membangun aplikasi yang handal, dibutuhkan pemilihan alat pengembangan yang tepat. Bahasa pemrograman Python dipilih karena memiliki pustaka yang lengkap dan sintaksis yang mudah dipahami.

Spesifikasi Perangkat Lunak:

- Bahasa Pemrograman: Python 3.14.
- GUI Library: *CustomTkinter* digunakan untuk membangun antarmuka pengguna yang modern dan mendukung tema gelap/terang.
- Database: File CSV (*Comma Separated Values*) digunakan sebagai media penyimpanan data karena ringan dan mudah diakses tanpa memerlukan server database yang rumit.
- Image Processing: Pustaka *Pillow* (PIL) digunakan untuk memanipulasi dan menampilkan aset gambar seperti logo dan latar belakang aplikasi.
- Date Handling: Pustaka *Datetime* digunakan untuk mencatat waktu transaksi dan menghitung durasi paket membership secara otomatis.

2.2 PERANCANGAN DATABASE (CSV)

Penyimpanan data dilakukan pada file `users.csv`. Struktur tabel dirancang untuk menyimpan informasi detail mengenai status keanggotaan. Setiap baris data merepresentasikan satu akun pengguna dengan atribut sebagai berikut:

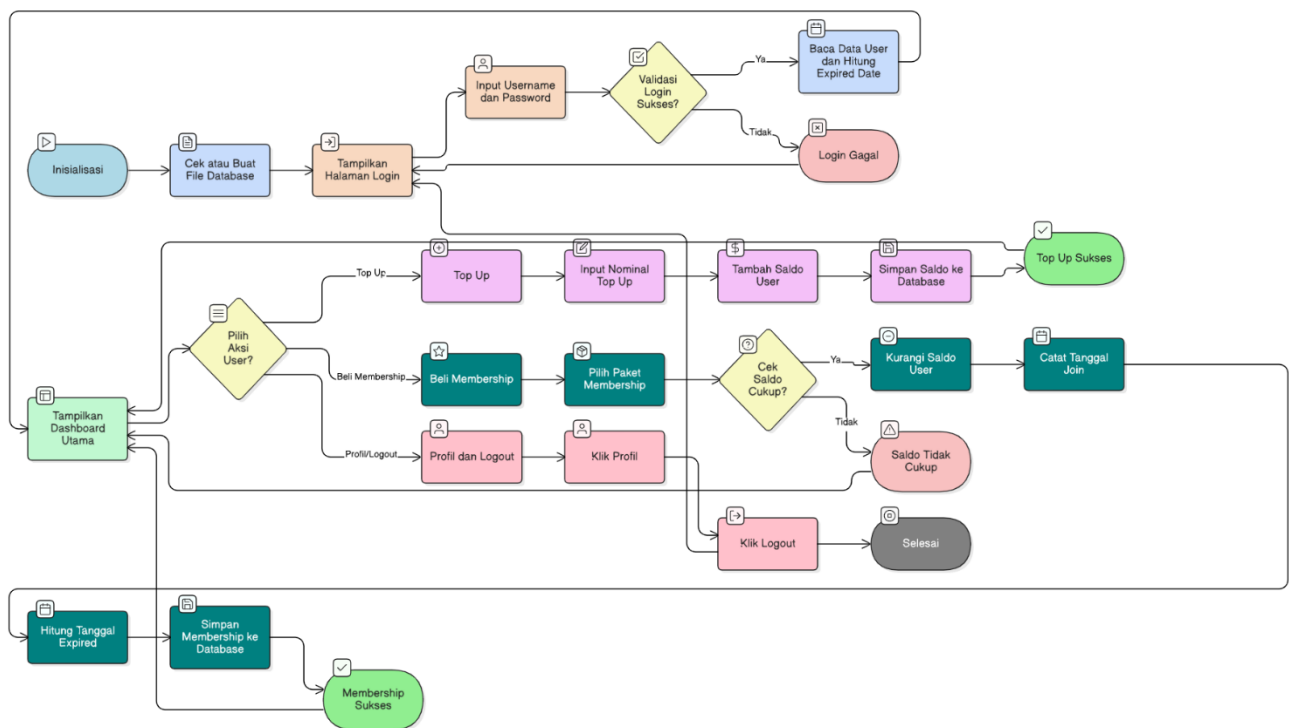
No	Nama Kolom	Fungsi
1.	<code>username</code>	Identitas unik untuk login.
2.	<code>password</code>	Kata sandi akun.
3.	<code>item</code>	Jenis paket membership (Bronze/Silver/Gold).
4.	<code>tanggal</code>	Tanggal dan waktu saat pengguna membeli paket (<i>Timestamp</i>).
5.	<code>harga</code>	Nominal harga paket yang dibeli.
6.	<code>saldo</code>	Sisa saldo <i>e-wallet</i> pengguna.
7.	<code>expired</code>	Tanggal berakhirnya masa aktif keanggotaan.

Tabel 1. Kumpulan Database

Struktur ini dipilih untuk memastikan bahwa setiap transaksi pembelian paket akan terekam kapan dilakukan dan sampai kapan berlakunya, sehingga memudahkan validasi status anggota saat masuk ke gym.

2.3 ALUR LOGIKA (*FLOWCHART*)

Sistem manajemen keanggotaan gym yang dilakukan secara manual seringkali memiliki alur data yang tidak terstruktur, terutama dalam hal validasi saldo dan penentuan masa aktif keanggotaan. Proyek ini mengatasi masalah tersebut dengan mengembangkan versi perangkat lunak yang menyediakan antarmuka interaktif, kontrol yang intuitif, serta umpan balik dinamis kepada pengguna. Diagram alir (*flowchart*) di bawah ini menguraikan urutan tindakan dan logika keputusan selama proses penggunaan aplikasi,



Gambar 2. 1 Tabel Flowchart

3.1 PENJELASAN KODE PROGRAM

3.1.1 Import Library dan Konfigurasi Path (Baris 1-10):

```
1  import tkinter as tk
2  from tkinter import messagebox as tkmb
3  import customtkinter as ctk
4  from datetime import datetime, timedelta
5  from PIL import Image
6  import csv
7  import os
8
9  # --- KONFIGURASI FILE ---
10 csv_file_path = "users.csv"
11
```

Gambar 3. 1 Import Library dan Konfigurasi Path

Importing Libraries:

- a. `import tkinter as tk` & `from tkinter import messagebox as tkmb`:
 - Fungsi Aplikasi: Fondasi utama antarmuka pengguna (GUI) dan sistem notifikasi *pop-up* (seperti pesan "Login Berhasil" atau "Saldo Kurang").
 - Fungsi Python: Mengimpor modul GUI standar Python. `as` digunakan untuk memberi nama alias (`tk` dan `tkmb`) agar penulisan kode selanjutnya lebih singkat.
- b. `import customtkinter as ctk`:
 - Fungsi Aplikasi: Memberikan tampilan widget (tombol, entry, frame) yang lebih modern dengan sudut membulat (corner radius) dan dukungan mode gelap/terang.
 - Fungsi Python: Pustaka pihak ketiga pembungkus Tkinter.
- c. `from datetime import datetime, timedelta`:
 - Fungsi Aplikasi: Otak dari sistem membership. Digunakan untuk mencatat waktu transaksi (jam/menit/detik) dan menghitung kapan membership berakhir (menambah hari).
 - Fungsi Python: `datetime` menangani objek waktu spesifik, `timedelta` menangani selisih durasi waktu.
- d. `from PIL import Image` :
 - Fungsi Aplikasi: Memproses gambar latar belakang dan ikon agar bisa ditampilkan di jendela aplikasi.
 - Fungsi Python: Bagian dari library *Pillow* untuk manipulasi citra digital.
- e. `import csv` & `import os`:

- Fungsi Aplikasi: `csv` untuk membaca/menulis data pengguna. `os` untuk mengecek apakah file database sudah ada di komputer.

Konfigurasi File:

- `csv_file_path = "users.csv"`
- Penjelasan: Mendefinisikan lokasi file database dalam variabel global agar konsisten digunakan oleh seluruh fungsi.

3.1.2 Fungsi Keamanan Aset (Safe Open), (Baris 13-22):

```

13 def safe_open(path, size=None):
14     try:
15         img = Image.open(path).convert("RGBA")
16         if size:
17             return img.resize(size, Image.LANCZOS)
18         return img
19     except Exception:
20         if size:
21             return Image.new("RGBA", size, (255, 255, 255, 0))
22         return Image.new("RGBA", (1, 1), (255, 255, 255, 0))
23 
```

Gambar 3. 2 Fungsi Keamanan Aset (Safe Open)

Fungsi `safe_open(path, size=None)`

- Tujuan Kode: Mencegah aplikasi crash (berhenti mendadak) jika file gambar (seperti background atau logo) hilang atau terhapus dari folder.
- Penjelasan Sintaks Python:
 - `try...except`: Blok penanganan error. Python mencoba menjalankan kode di dalam `try`. Jika gagal (misal file tidak ditemukan), alih-alih error, Python menjalankan blok `except`.
 - `Image.open(path).convert("RGBA")`: Membuka gambar dan memastikan format warnanya mendukung transparansi (Alpha channel).
 - `Image.new("RGBA", size, ...)`: Jika gambar asli gagal dimuat, baris ini membuat "gambar kosong" (transparan) sebagai pengganti darurat agar aplikasi tetap berjalan.

3.1.3 Inisialisasi Database CSV (Baris 24-30):

```

24 def init_csv():
25     if not os.path.exists(csv_file_path):
26         with open(csv_file_path, mode="w", newline="", encoding="utf-8") as file:
27             writer = csv.DictWriter(file, fieldnames=[
28                 "username", "password", "item", "join_date", "harga", "saldo", "expired"
29             ])
30             writer.writeheader()
31 
```

Gambar 3. 3 Inisialisasi Database CSV

Fungsi `init_csv()` :

- a. Tujuan Kode: Memastikan file penyimpanan (`users.csv`) tersedia saat aplikasi pertama kali dijalankan.
- b. Penjelasan Sintaks Python:
 - `os.path.exists(csv_file_path):` Mengecek keberadaan file. Mengembalikan `False` jika file belum ada.
 - `with open(..., mode="w"):` Membuka file dalam mode *write* (tuliskan). Jika file belum ada, Python akan membuatnya otomatis. `with` memastikan file ditutup otomatis setelah selesai ditulis untuk mencegah korupsi data.
 - `writer.writerow():` Menulis baris pertama CSV sebagai judul kolom (`username`, `password`, `saldo`, dll).

3.1.4 Fungsi Dasar CRUD (Create, Read, Update), (Baris 32-67):

```
32 def read_users(file_path):
33     users = {}
34     try:
35         with open(file_path, mode='r', newline='', encoding='utf-8') as file:
36             reader = csv.DictReader(file)
37             for row in reader:
38                 users[row["username"]] = row["password"]
39     except FileNotFoundError:
40         init_csv()
41     return users
42
43 def load_user_saldo(username, file_path="users.csv"):
44     with open(file_path, "r", encoding="utf-8") as file:
45         reader = csv.DictReader(file)
46         for row in reader:
47             if row["username"] == username:
48                 return int(row["saldo"]) if row["saldo"] else 0
49     return 0
50
51 def get_user_saldo(username):
52     return load_user_saldo(username)
53
54 def write_user(file_path, username, password):
55     with open(file_path, mode='a', newline='', encoding='utf-8') as file:
56         writer = csv.DictWriter(file, fieldnames=[
57             "username", "password", "item", "join_date", "harga", "saldo", "expired"
58         ])
59         writer.writerow({
60             "username": username,
61             "password": password,
62             "item": "",
63             "join_date": "",
64             "harga": "",
65             "saldo": 0,
66             "expired": ""
67         })
```

Gambar 3. 4 Fungsi Dasar CRUD (Create, Read, Update)

Fungsi `read_users` & `load_user_saldo`

- a. Tujuan Kode:
 - `read_users`: Mengambil data username dan password untuk pencocokan saat login.
 - `load_user_saldo`: Mengambil data saldo spesifik milik satu pengguna untuk ditampilkan di dashboard.

b. Penjelasan Sintaks Python:

- `csv.DictReader(file)`: Membaca CSV dan mengubah setiap baris menjadi Dictionary Python, sehingga data bisa dipanggil menggunakan nama kolom (contoh: `row["username"]`).
- `int(row["saldo"]) if row["saldo"] else 0`: *Ternary Operator*. Jika kolom saldo ada isinya, ubah jadi integer. Jika kosong, anggap 0. Ini mencegah error perhitungan matematika.

Fungsi `write_user`:

- Tujuan Kode: Menyimpan pengguna baru saat Registrasi.
- b. Penjelasan Sintaks Python:
- `mode='a'` (*Append*): Membuka file untuk "menambahkan" data di baris paling bawah, berbeda dengan `w` yang menghapus isi lama.

3.1.5 Fungsi Transaksi Kompleks (Baris 69-87)

```
69 def save_user_data(username, item, price, saldo, file_path="users.csv"):
70     rows = []
71     with open(file_path, "r", encoding="utf-8") as file:
72         reader = csv.DictReader(file)
73         for row in reader:
74             if row["username"] == username:
75                 duration = DURASI_MEMBERSHIP.get(item, 0)
76                 if item:
77                     row["expired"] = hitung_join_date_expired(row.get("expired", ""), duration)
78                     row["item"] = item
79                     row["harga"] = price
80                     row["join_date"] = datetime.now().strftime("%d-%m-%Y %H:%M:%S")
81                     row["saldo"] = str(saldo)
82                 rows.append(row)
83
84     with open(file_path, "w", newline="", encoding="utf-8") as file:
85         writer = csv.DictWriter(file, fieldnames=["username", "password", "item", "join_date", "harga", "saldo", "expired"])
86         writer.writeheader()
87         writer.writerows(rows)
```

Gambar 3. 5 Fungsi Transaksi Kompleks

Fungsi `save_user_data`:

- Tujuan Kode: Fungsi paling krusial untuk transaksi. Ia memperbarui saldo, item membership, dan tanggal expired pengguna tanpa menghapus pengguna lain.
- b. Penjelasan Sintaks Python:
- Logika Update: Kode membaca seluruh isi file ke dalam memori (`rows = []`). Saat perulangan (`for row in reader`) menemukan username yang sedang login, data di baris tersebut diubah (*di-overwrite*).
 - `writer.writerows(rows)`: Setelah data di memori diperbarui, seluruh daftar tersebut ditulis ulang ke file CSV, menimpa file lama dengan data terbaru.

3.1.6 Algoritma Tanggal dan Utilitas UI (Baris 89-114):

```
89 DURASI_MEMBERSHIP = {
90     "Bronze": 1,
91     "Silver": 7,
92     "Gold": 30
93 }
94
95 def hitung_join_date_expired(expired_lama, tambah_hari):
96     sekarang = datetime.now()
97     if not expired_lama:
98         return (sekarang + timedelta(days=tambah_hari)).strftime("%d-%m-%Y")
99     try:
100         expired_obj = datetime.strptime(expired_lama, "%d-%m-%Y")
101         if expired_obj > sekarang:
102             expired_baru = expired_obj + timedelta(days=tambah_hari)
103         else:
104             expired_baru = sekarang + timedelta(days=tambah_hari)
105         return expired_baru.strftime("%d-%m-%Y")
106     except:
107         return (sekarang + timedelta(days=tambah_hari)).strftime("%d-%m-%Y")
108
109 def center_window(root, width, height):
110     screen_width = root.winfo_screenwidth()
111     screen_height = root.winfo_screenheight()
112     x = (screen_width - width) // 2
113     y = (screen_height - height) // 2
114     root.geometry(f"{width}x{height}+{x}+{y}")
115
```

Gambar 3. 6 Algoritma Tanggal dan Utilitas UI

Dictionary `DURASI_MEMBERSHIP` & **Fungsi** `hitung_join_date_expired`:

- a. Fungsi Aplikasi:
 - Definisi Durasi: Variabel `DURASI_MEMBERSHIP` menentukan berapa lama setiap paket berlaku (Bronze=1 hari, Silver=7 hari, Gold=30 hari).
 - Logika Akumulasi: Fungsi `hitung_join_date_expired` memastikan pengguna tidak rugi. Jika pengguna membeli paket baru saat paket lama masih aktif (`expired_obj > sekarang`), durasi baru akan ditambahkan ke tanggal expired lama (akumulasi). Jika sudah habis, durasi dihitung dari hari ini.
- b. Penjelasan Sintaks Python:
 - `datetime.now()` : Mengambil waktu komputer saat ini sebagai titik acuan.
 - `datetime.strptime(..., "%d-%m-%Y")` : *String Parse Time*. Mengubah teks tanggal dari file CSV (misal: "20-12-2025") menjadi objek waktu Python agar bisa dibandingkan secara matematis.
 - `timedelta(days=tambah_hari)`: Fungsi matematika waktu untuk menambahkan durasi hari ke sebuah tanggal.

- `try...except`: Penanganan error. Jika format tanggal di database rusak atau tidak valid, program tidak akan crash, melainkan otomatis menghitung durasi dari hari ini (blok `except`).

Fungsi `center_window`:

- a. Fungsi Aplikasi: Meningkatkan kenyamanan pengguna (*User Experience*) dengan memastikan setiap jendela aplikasi (Login, Menu, Top Up) selalu muncul tepat di tengah layar monitor, terlepas dari resolusi layar pengguna.
- b. Penjelasan Sintaks Python:
 - `root.winfo_screenwidth()`: Perintah Tkinter untuk mendeteksi lebar layar monitor pengguna secara otomatis.
 - `// 2`: Operator *Floor Division*. Pembagian bilangan bulat untuk mencari titik tengah koordinat X dan Y tanpa menghasilkan angka desimal.
 - `root.geometry(f"...")`: Menggunakan *f-string* untuk menyusun format geometri Tkinter (lebar x tinggi + posisi_x + posisi_y).

3.1.7 Inisialisasi Antarmuka Login (Baris 119-169)

```
class LoginApp:
    def __init__(self, root, csv_path):
        self.root = root
        self.csv_path = csv_path
        self.root.title("Login - GymEmas")
        center_window(self.root, 388, 824)

        bg_img = safe_open("bg_login.png", size=(388, 824))
        self.bg_image = ctk.CTkImage(light_image=bg_img, size=(388, 824))
        self.bg_label = ctk.CTkLabel(self.root, image=self.bg_image, text="")
        self.bg_label.place(x=0, y=0, relwidth=1, relheight=1)

        self.loginname = ctk.CTkLabel(self.root, text="Login", font=("Crimson Text", 32),
                                     text_color=□ "#000000", fg_color=■ "#FFFFFF")
        self.loginname.place(x=150, y=200)

        self.entry_username = ctk.CTkEntry(self.root, placeholder_text="Username",
                                          fg_color=■ "#FFFFFF", text_color=□ "#000000",
                                          placeholder_text_color=■ "#A99090", border_color=■ "#FFFFFF",
                                          bg_color=■ "#FFFFFF", corner_radius=10, width=250, height=34)
        self.entry_username.place(x=90, y=325)

        self.entry_password = ctk.CTkEntry(self.root, placeholder_text="Password", show="*",
                                          fg_color=■ "#FFFFFF", text_color=□ "#000000",
                                          placeholder_text_color=■ "#A99090", border_color=■ "#FFFFFF",
                                          bg_color="transparent", corner_radius=10, width=250, height=21)
        self.entry_password.place(x=90, y=395)

        ctk.CTkButton(self.root, text="Login", width=170, height=42,
                      fg_color=■ "#FF9D00", hover_color=■ "#B90000", bg_color=■ "#FFF1F3",
                      command=self.login).place(x=110, y=450)

        dont = ctk.CTkLabel(self.root, text="Don't have an account?", font=("Crimson Text", 12),
                           width=111, height=16, text_color=□ "#000000", bg_color=■ "#FFFFFF")
        dont.place(x=88, y=505)

        signup = ctk.CTkButton(self.root, text="Sign Up", font=("Crimson Text", 12),
                               width=40, height=16, text_color=■ "#0092FA", fg_color=■ "#FFFFFF", bg_color=■ "#FFFFFF",
                               command=self.open_register)
        signup.place(x=210, y=503)

        info_img = safe_open("infobutton.png", size=(33, 33))
        self.info_photo = ctk.CTkImage(light_image=info_img, size=(22,22))
        ctk.CTkButton(self.root, image=self.info_photo, text="", width=22, height=22,
                      fg_color=(■ "#C96E05"), hover_color=(□ "#04103A"), bg_color=(■ "#ffffff"),
                      command=self.open_info).place(x=20, y=20)

        signup.bind("<Enter>", self.on_hover)
        signup.bind("<Leave>", self.on_leave)
        self.root.bind("<Return>", self.login)
```

Gambar 3. 7 Inisialisasi Antarmuka Login

Class `LoginApp` - **Method** `__init__`:

- a) Tujuan Kode: Membangun tampilan visual jendela Login. Kode ini mengatur latar belakang, label judul, kotak input untuk username dan password, serta tombol "Login", "Sign Up", dan tombol informasi ("i").
- b) Penjelasan Sintaks Python:
 - `self.root = root`: Menyimpan referensi jendela utama agar bisa diakses oleh metode lain di dalam class ini.
 - `ctk.CTkImage`: Objek khusus CustomTkinter untuk menampilkan gambar dengan kualitas tinggi pada widget.
 - `ctk.CTkEntry(..., show="*")`: Membuat kolom input teks. Parameter `show="*"` sangat penting untuk keamanan password, mengubah karakter yang diketik menjadi simbol bintang agar tidak terbaca orang lain.
 - `bind('<Return>', self.login)`: *Event Binding*. Mengaitkan tombol "Enter" pada keyboard dengan fungsi `login`, sehingga pengguna bisa login dengan menekan Enter tanpa harus mengklik tombol.

3.1.8 Logika Validasi (Baris 171-188)

```
171 def login(self, event=None):
172     username = self.entry_username.get()
173     password = self.entry_password.get()
174
175     users = read_users(self.csv_path)
176     if username in users and users[username] == password:
177         tkmb.showinfo("Login", "Login berhasil! Selamat datang di GymEmas.")
178         self.root.withdraw()
179         saldo_user = get_user_saldo(username)
180         try:
181             gym_menu.update_username(username)
182             gym_menu.update_saldo(saldo_user)
183             menu_app.deiconify()
184         except Exception as e:
185             tkmb.showerror("ERROR", f"Terjadi error: {e}")
186             self.root.deiconify()
187     else:
188         tkmb.showerror("Login", "Username atau password salah!")
189
```

Gambar 3. 8 Logika Validasi

Fungsi `login(self, event=None)`:

- a) Tujuan Kode: Memproses autentikasi. Kode ini mengambil teks yang diketik pengguna, membandingkannya dengan data di `users.csv`. Jika cocok, jendela login disembunyikan dan menu utama dibuka.
- b) Penjelasan Sintaks Python:

- `self.entry_username.get()`: Mengambil string (teks) yang saat itu tertulis di dalam kotak input.
- `if username in users and ...`: Logika pengecekan ganda. Python memastikan username ada di database DAN password-nya cocok dengan pasangan kuncinya di Dictionary.
- `self.root.withdraw()`: Menyembunyikan jendela Login tanpa mematikannya (*terminate*). Ini memungkinkan jendela Login dimunculkan kembali nanti saat pengguna melakukan Logout.
- `tkmb.showerror(...)`: Memunculkan *pop-up* error jika login gagal.

3.1.9 Navigasi Jendela (Window Navigation), (Baris 190-214)

```

190 | def open_register(self):
191 |     self.root.withdraw()
192 |     register_window = ctk.CTkToplevel(self.root)
193 |     register_window.lift()
194 |     register_window.attributes('-topmost', True)
195 |     register_window.focus_force()
196 |     register_window.grab_set()
197 |     RegisterApp(register_window, self, self.csv_path)
198 |
199 | def open_info(self):
200 |     self.root.withdraw()
201 |     info_window = ctk.CTkToplevel(self.root)
202 |     info_window.lift()
203 |     info_window.attributes('-topmost', True)
204 |     info_window.focus_force()
205 |     info_window.grab_set()
206 |     InfoKelompok(info_window)
207 |
208 | def on_hover(self, event):
209 |     try: event.widget.configure(font=("Crimson Text", 8, "underline"))
210 |     except: pass
211 |
212 | def on_leave(self, event):
213 |     try: event.widget.configure(font=("Crimson Text", 8))
214 |     except: pass
215 |

```

Gambar 3. 9 Navigasi Jendela (Window Navigation)

Fungsi `open_register` & `open_info`:

- Tujuan Kode: Membuka jendela baru (Register atau Info) di atas jendela Login.
- Penjelasan Sintaks Python:
 - `ctk.CTkToplevel(self.root)`: Membuat jendela sekunder (*child window*) yang terpisah dari jendela utama tetapi tetap terikat padanya.
 - `lift()` & `attributes('-topmost', True)`: Memaksa jendela baru muncul di lapisan paling depan (*foreground*) layar agar tidak tertutup jendela lain.
 - `grab_set()`: Membuat jendela menjadi *modal*. Artinya, pengguna wajib menutup jendela baru ini sebelum bisa berinteraksi kembali dengan jendela

Login di belakangnya. Ini mencegah pengguna membuka banyak jendela sekaligus.

3.1.10 Class Info Kelompok (Baris 216-257)

```
216 # =====
217 # CLASS INFO KELOMPOK
218 # =====
219 class InfoKelompok:
220     def __init__(self, root):
221         self.root = root
222         self.root.title("Info Kelompok - GymEmas")
223         center_window(self.root, 400, 300)
224
225         bgidentitas = safe_open("bg_identitas.png", size=(800, 600))
226         self.bgphoto = ctk.CTkImage(light_image=bgidentitas, size=(800, 600))
227         bg_label = ctk.CTkLabel(self.root, image=self.bgphoto, text="")
228         bg_label.place(x=0, y=0, relwidth=1, relheight=1)
229
230         developedby = ctk.CTkLabel(self.root, text="DEVELOPED BY:", font=("Crimson Text", 27,),
231                                   width=165, height=32, text_color="black", fg_color="transparent")
232         developedby.place(x=118, y=15)
233
234         Alfath = ctk.CTkLabel(self.root, text="Akeyla Fadillah Alfath Herliyanantha (25031554074)",
235                               font=("Crimson Text", 12,), text_color="black", fg_color="transparent")
236         Alfath.place(x=84, y=70)
237
238         Aryo = ctk.CTkLabel(self.root, text="Nabil Aryo Suharnoto (25031554092)",
239                             font=("Crimson Text", 12,), text_color="black", fg_color="transparent")
240         Aryo.place(x=84, y=120)
241
242         Dinda = ctk.CTkLabel(self.root, text="Adinda Atsilla Salsabilah (25031554262)",
243                               font=("Crimson Text", 12,), text_color="black", fg_color="transparent")
244         Dinda.place(x=84, y=170)
245
246         back = safe_open("tombolbacktb.png", size=(27, 27))
247         self.backphoto = ctk.CTkImage(light_image=back, size=(27, 27))
248         ctk.CTkButton(self.root, image=self.backphoto, text="", width=27, height=27,
249                       fg_color="red", hover_color="blue", bg_color="white",
250                       command=self.go_back).place(x=29, y=14)
251
252     def go_back(self):
253         self.root.destroy()
254         try:
255             login_app.deiconify()
256         except:
257             pass
```

Gambar 3. 10 Class Info Kelompok

Class InfoKelompok:

- Tujuan Kode: Menampilkan identitas pengembang aplikasi (Nama dan NIM anggota kelompok).
- Penjelasan Sintaks Python:

- `self.root.destroy()` : Berbeda dengan `withdraw` di Login, fungsi ini digunakan pada tombol "Back" untuk menghancurkan jendela Info sepenuhnya guna menghemat memori komputer, karena jendela ini tidak perlu disimpan statusnya.
- `try...except` pada `go_back`: Mencegah error jika pengguna menutup jendela utama secara paksa saat jendela Info masih terbuka.

3.1.11 Logika Pendaftaran Akun Register, (Baris 259-338)

```

259 # =====
260 # CLASS REGISTER APP
261 # =====
262 class RegisterApp:
263     def __init__(self, root, login_app, csv_path):
264         self.root = root
265         self.login_app = login_app
266         self.csv_path = csv_path
267         self.root.title("Register - GymEmas")
268         center_window(self.root, 388, 824)
269
270         bg_reg = safe_open("bg_register.png", size=(388, 824))
271         self.bg_image = ctk.CTkImage(light_image=bg_reg, size=(388, 824))
272         self.bg_label = ctk.CTkLabel(self.root, image=self.bg_image, text="")
273         self.bg_label.place(x=0, y=0, relwidth=1, relheight=1)
274
275         self.registername = ctk.CTkLabel(self.root, text="Sign Up", font=("Crimson Text", 30),
276                                         text_color="FFFFFF", bg_color="000000")
277         self.registername.place(x=150, y=200)
278
279         self.create_username = ctk.CTkEntry(self.root, placeholder_text="Create Username",
280                                             fg_color="000000", text_color="FFFFFF",
281                                             placeholder_text_color="A99090", border_color="D9D9D9",
282                                             bg_color="transparent", corner_radius=10, width=250, height=34)
283         self.create_username.place(x=90, y=325)
284
285         self.create_password = ctk.CTkEntry(self.root, placeholder_text="Create Password", show="*",
286                                             fg_color="000000", text_color="FFFFFF",
287                                             placeholder_text_color="A99090", border_color="D9D9D9",
288                                             bg_color="transparent", corner_radius=10, width=250, height=34)
289         self.create_password.place(x=90, y=395)
290
291         self.button_create = ctk.CTkButton(self.root, text="Create", width=170, height=42, corner_radius=10,
292                                           fg_color="FB880B", hover_color="FB880B", bg_color="000000",
293                                           command=self.register)
294         self.button_create.place(x=110, y=450)
295
296         self.i_have_account = ctk.CTkLabel(self.root, text="I have an account,", font=("Crimson Text", 12),
297                                           width=85, height=16, text_color="FFFFFF", fg_color="000000")
298         self.i_have_account.place(x=95, y=505)
299
300         self.loginnow = ctk.CTkButton(self.root, text="Login Here!", font=("Crimson Text", 12),
301                                       width=60, height=16, text_color="0092FA", fg_color="000000", bg_color="000000",
302                                       command=self.go_back)
303         self.loginnow.place(x=190, y=505)
304
305         self.loginnow.bind("<Enter>", self.on_hover)
306         self.loginnow.bind("<Leave>", self.on_leave)
307         self.root.bind('<Return>', self.register)

```

Gambar 3. 11 Logika Pendaftaran Akun Register (1)

```

309 ✓ def register(self, event=None):
310     username = self.create_username.get()
311     password = self.create_password.get()
312
313 ✓     if not username or not password:
314         tkmb.showerror("Error", "Semua field harus diisi!")
315         return
316
317     users = read_users(self.csv_path)
318 ✓     if username in users:
319         tkmb.showerror("Error", "Username sudah terdaftar!")
320 ✓     else:
321         write_user(self.csv_path, username, password)
322         tkmb.showinfo("Sukses", "Registrasi berhasil! Silakan login.")
323         self.go_back()
324
325 ✓ def go_back(self):
326     self.root.destroy()
327 ✓     try:
328         self.login_app.root.deiconify()
329 ✓     except Exception:
330         pass
331
332 ✓ def on_hover(self, event):
333     try: event.widget.configure(font=("Crimson Text", 8, "underline"))
334     except: pass
335
336 ✓ def on_leave(self, event):
337     try: event.widget.configure(font=("Crimson Text", 8))
338     except: pass
339

```

Gambar 3. 12 Logika Pendaftaran Akun Register (2)

```

309 ✓ def register(self, event=None):
310     username = self.create_username.get()
311     password = self.create_password.get()
312
313 ✓     if not username or not password:
314         tkmb.showerror("Error", "Semua field harus diisi!")
315         return
316
317     users = read_users(self.csv_path)
318 ✓     if username in users:
319         tkmb.showerror("Error", "Username sudah terdaftar!")
320 ✓     else:
321         write_user(self.csv_path, username, password)
322         tkmb.showinfo("Sukses", "Registrasi berhasil! Silakan login.")
323         self.go_back()
324
325 ✓ def go_back(self):
326     self.root.destroy()
327 ✓     try:
328         self.login_app.root.deiconify()
329 ✓     except Exception:
330         pass
331
332 ✓ def on_hover(self, event):
333     try: event.widget.configure(font=("Crimson Text", 8, "underline"))
334     except: pass
335
336 ✓ def on_leave(self, event):
337     try: event.widget.configure(font=("Crimson Text", 8))
338     except: pass
339

```

Gambar 3. 13 Logika Pendaftaran Akun Register (3)

Class `RegisterApp` & Fungsi `register`:

- a) Tujuan Kode: Menangani pembuatan akun baru. Sistem memvalidasi input (tidak boleh kosong) dan memastikan username belum pernah dipakai sebelumnya.
- b) Penjelasan Sintaks Python:
 - `if not username or not password`: Validasi input kosong. Di Python, string kosong `""` dianggap bernilai `False`. Jadi `not ""` menjadi `True`, memicu pesan error "Semua field harus diisi".
 - `write_user(...)` : Jika lolos validasi, fungsi ini dipanggil untuk menulis data permanen ke file CSV.
 - `focus_force()` : Memaksa kursor keyboard langsung aktif di jendela Register agar pengguna bisa langsung mengetik tanpa mengklik mouse.

3.1.12 Inisialisasi Dashboard Utama (Main Menu), (Baris 340-402)

```

40 # =====
41 # CLASS GYM MENU (DASHBOARD)
42 class GymMenu:
43     def __init__(self, root, username):
44         self.root = root
45         self.username = username
46         self.expired_date = self.load_expired(self.username)
47         self.expired_var = tk.StringVar(value=f"Expired: {self.expired_date}")
48
49         self.root.title("Menu Utama - GymEmas")
50         center_window(self.root, 398, 824)
51         bgimage = safe_open("bg gymmenu.png", size=(398, 824))
52         self.bgphoto = ctk.CTkImage(light_image=bgimage, size=(398, 824))
53         self.bglabel = ctk.CTkLabel(self.root, image=self.bgphoto, text="")
54         self.bglabel.place(x=0, y=0, relwidth=1, relheight=1)
55         self.profil = safe_open("profilegym.png", size=(48,48))
56         self.profilphoto = ctk.CTkImage(light_image=self.profil, size=(48,48))
57         self.profilbutton = ctk.CTkButton(
58             self.root, image=self.profilphoto,
59             text="", width=48, height=48,
60             bg_color="#ffbd59", fg_color="#ffbd59", hover_color="#faa422",
61             command=self.open_profile
62         )
63         self.profilbutton.place(x=5, y=10)
64
65         self.lbl_nama = ctk.CTkLabel(self.root, text=f"Hi, {self.username}",
66                                     font=("Inter", 16, "bold"), text_color="FFFFFF", bg_color="#ffbd59")
67         self.lbl_nama.place(x=65, y=10)
68
69         self.saldo = load_user_saldo(self.username)
70         self.saldo_var = tk.StringVar()
71         self.update_saldo(self.saldo)
72         self.gym_saldo = ctk.CTkLabel(self.root, textvariable=self.saldo_var,
73                                     font=("Inter", 14), text_color="#00FF00",
74                                     bg_color="#ffbd59")
75         self.gym_saldo.place(x=65, y=35)
76
77         topup = ctk.CTkButton(self.root, text="+ TOP UP", width=80, height=30,
78                             font=("Inter", 12, "bold"),
79                             hover_color="#353333", bg_color="#000000", fg_color="#000000",
80                             text_color="FFFFFF", command=self.open_topup)
81         topup.place(x=300, y=25)

```

Gambar 3. 14 Inisialisasi Dashboard Utama (Main Menu) (1)

```

382
383         ctk.CTkLabel(self.root, text="PILIH MEMBERSHIP", font=("Inter", 20, "bold"),
384                     text_color="#000000", bg_color="FFFFFF").place(x=44, y=150)
385
386         bronze_btn = ctk.CTkButton(self.root, text="BRONZE\nRp 75.000", width=300, height=80,
387                                   font=("Inter", 18, "bold"),
388                                   fg_color="#A97142", hover_color="#7A522F", text_color="white",
389                                   command=lambda: self.buy_item("Bronze", 75000))
390         bronze_btn.place(x=44, y=200)
391
392         silver_btn = ctk.CTkButton(self.root, text="SILVER\nRp 125.000", width=300, height=80,
393                                   font=("Inter", 18, "bold"),
394                                   fg_color="#C0C0C0", hover_color="#8F8F8F", text_color="black",
395                                   command=lambda: self.buy_item("Silver", 125000))
396         silver_btn.place(x=44, y=300)
397
398         gold_btn = ctk.CTkButton(self.root, text="GOLD\nRp 250.000", width=300, height=80,
399                                 font=("Inter", 18, "bold"),
400                                 fg_color="#FFD700", hover_color="#B89D00", text_color="black",
401                                 command=lambda: self.buy_item("Gold", 250000))
402         gold_btn.place(x=44, y=400)

```

Gambar 3. 14 Inisialisasi Dashboard Utama (Main Menu) (2)

Class `GymMenu` - Method `__init__` :

- a) Tujuan Kode: Menyiapkan tampilan dashboard utama. Kode ini memuat data saldo dan status *expired* pengguna dari database, lalu menampilkannya di layar. Ini juga membuat tombol untuk 3 jenis paket membership (Bronze, Silver, Gold).
- b) Penjelasan Sintaks Python:
 - `self.expired_var = tk.StringVar(...)`: Menggunakan variabel khusus Tkinter (*StringVar*). Berbeda dengan string biasa, jika nilai variabel ini diubah, teks pada Label di layar akan otomatis berubah tanpa perlu me-refresh jendela.
 - `f"Rp {saldo:,}".replace(",",".")`: Format string kompleks. Python memformat angka dengan pemisah koma (contoh: 10,000), lalu `.replace` menukarnya menjadi titik (10.000) agar sesuai standar mata uang Rupiah.
 - `command=lambda: self.buy_item(...)`: Menggunakan *Lambda Function* (fungsi anonim). Tanpa lambda, fungsi `buy_item` akan langsung dijalankan saat aplikasi dibuka. Lambda memastikan fungsi hanya jalan saat tombol diklik, sekaligus memungkinkan pengiriman argumen spesifik (misal: "Bronze", 75000) ke dalam fungsi tersebut.

3.1.13 Fungsi Navigasi dan Pembaruan Tampilan, (Baris 404-510)

```
404 def open_profile(self):
405     profil_window = ctk.CTkToplevel(self.root)
406     profil_window.lift()
407     profil_window.attributes('-topmost', True)
408     profil_window.focus_force()
409     profil_window.grab_set()
410     ProfilePage(profil_window, self.username)
411
412 def update_saldo(self, saldo):
413     self.saldo = saldo
414     self.saldo_var.set(f"Rp {saldo:,}".replace(",", "."))
415
416 def logout(self, event=None):
417     try:
418         self.root.withdraw()
419         login_app.deiconify()
420     except Exception:
421         pass
422
423 def open_topup(self):
424     topup_window = ctk.CTkToplevel(self.root)
425     topup_window.lift()
426     topup_window.attributes('-topmost', True)
427     topup_window.focus_force()
428     topup_window.grab_set()
429     TopUpApp(topup_window, self)
430
431 def update_username(self, username):
432     self.username = username
433     self.lbl_nama.configure(text=f"Hi, {self.username}")
434
435 def on_hover(self, event):
436     pass
437
438 def on_leave(self, event):
439     pass
440
441 def buy_item(self, item, price):
442     if self.saldo >= price:
443         self.saldo -= price
444         self.update_saldo(self.saldo)
445         save_user_data(self.username, item, price, self.saldo)
446         self.expired_date = (datetime.now() + timedelta(days=DURASI_MEMBERSHIP[item])).strftime('%d-%m-%Y')
447         self.expired_var.set(f"Expired: {self.expired_date}")
448         join_date_beli = datetime.now().strftime("%d-%m-%Y %H:%M:%S")
449         tkmb.showinfo("Pembelian Berhasil",
450             f"{item} berhasil dibeli!\nHarga: Rp {price:,}\njoin_date: {join_date_beli}".replace(",", "."))
451     else:
452         tkmb.showerror("Gagal", "Saldo tidak mencukupi!\nSilakan Top Up terlebih dahulu.")
453
```

Gambar 3. 15 Fungsi Navigasi dan Pembaruan Tampilan (1)

```

454 def load_expired(self, username):
455     with open("users.csv", "r", encoding="utf-8") as file:
456         reader = csv.DictReader(file)
457         for row in reader:
458             if row["username"] == username:
459                 return row.get("expired", "")
460     return ""
461
462 def get_remaining_days(self):
463     try:
464         if not self.expired_date: return "Tidak ada membership"
465         expired = datetime.strptime(self.expired_date, "%d-%m-%Y")
466         now = datetime.now()
467         if expired < now: return "Expired"
468         sisa = expired - now
469         return f"{{(exp_date - now).days}} hari"
470     except:
471         return "Data tidak valid"
472
473 def open_profile(self):
474     profil_window = ctk.CTkToplevel(self.root)
475     profil_window.lift()
476     profil_window.attributes('-topmost', True)
477     profil_window.focus_force()
478     profil_window.grab_set()
479     ProfilePage(profil_window, self.username)
480
481 def update_saldo(self, saldo):
482     self.saldo = saldo
483     self.saldo_var.set(f"Saldo: Rp {saldo:,}".replace(",", "."))
484
485 def logout(self, event=None):
486     try:
487         self.root.withdraw()
488         login_app.deiconify()
489     except Exception:
490         pass
491
492 def open_topup(self):
493     topup_window = ctk.CTkToplevel(self.root)
494     topup_window.lift()
495     topup_window.attributes('-topmost', True)
496     topup_window.focus_force()
497     topup_window.grab_set()
498     TopUpApp(topup_window, self)

```

Gambar 3. 16 Fungsi Navigasi dan Pembaruan Tampilan (2)

```

500 def update_username(self, username):
501     self.username = username
502     self.lbl_nama.configure(text=f"Hi, {self.username}")
503
504 def on_hover(self, event):
505     try: event.widget.configure(font=("Inter", 14, "underline"))
506     except: pass
507
508 def on_leave(self, event):
509     try: event.widget.configure(font=("Inter", 14))
510     except: pass

```

Gambar 3. 17 Fungsi Navigasi dan Pembaruan Tampilan (3)

Fungsi `open_profile`, `update_saldo`, `logout`, `open_topup`:

a. Tujuan Kode:

- `open_profile` & `open_topup`: Membuka jendela menu lain.
- `update_saldo`: Fungsi penting untuk memperbarui tampilan uang di pojok kiri atas secara *real-time* setelah *top-up* atau pembelian.
- `logout`: Kembali ke halaman login.

b. Penjelasan Sintaks Python:

- `self.saldo_var.set(...)`: Metode `.set()` digunakan pada objek `StringVar` untuk mengganti isi teks yang tampil di layar UI seketika itu juga.
- `self.root.withdraw()` pada fungsi `logout`: Menyembunyikan dashboard utama agar pengguna bisa melihat kembali jendela login (`login_app.deiconify()`).

3.1.14 Logika Transaksi Pembelian Membership, (Baris 512-523)

```
512 def buy_item(self, item, price):
513     if self.saldo >= price:
514         self.saldo -= price
515         self.update_saldo(self.saldo)
516         save_user_data(self.username, item, price, self.saldo)
517         self.expired_date = (datetime.now() + timedelta(days=DURASI_MEMBERSHIP[item])).strftime('%d-%m-%Y')
518         self.expired_var.set(f"Expired: {self.expired_date}")
519         join_date_beli = datetime.now().strftime("%d-%m-%Y %H:%M:%S")
520         tkmb.showinfo("Pembelian Berhasil",
521                       f"{item} berhasil dibeli!\nHarga: Rp {price:,.0f}\njoin_date: {join_date_beli}".replace(",", "."))
522     else:
523         tkmb.showerror("Gagal", "Saldo tidak mencukupi!\nSilakan Top Up terlebih dahulu.")
```

Gambar 3. 18 Logika Transaksi Pembelian Membership

Fungsi `buy_item(self, item, price)`:

- a. Tujuan Kode: Menangani proses pembelian. Logikanya: Cek apakah uang cukup -> Kurangi saldo -> Simpan ke CSV -> Perbarui tanggal expired -> Tampilkan pesan sukses.
- b. Penjelasan Sintaks Python:
- `if self.saldo >= price`: Operator perbandingan (*Comparison Operator*) untuk validasi saldo.
 - `save_user_data(...)`: Memanggil fungsi global (yang didefinisikan di awal kode) untuk menulis perubahan data ke file CSV secara permanen.
 - `DURASI_MEMBERSHIP[item]`: Mengakses nilai dari Dictionary global. Jika `item="Gold"`, kode ini otomatis mengambil angka 30.
 - `datetime.now()`: Mengambil *timestamp* (waktu saat ini) presisi detik untuk dicatat sebagai "Join Date".

3.1.15 Logika Perhitungan Status Membership (Baris 525-542)

```
525 ✓ def load_expired(self, username):
526 ✓     with open("users.csv", "r", encoding="utf-8") as file:
527         reader = csv.DictReader(file)
528         for row in reader:
529             if row["username"] == username:
530                 return row.get("expired", "")
531         return ""
532
533 ✓ def get_remaining_days(self):
534 ✓     try:
535         if not self.expired_date: return "Tidak ada membership"
536         expired = datetime.strptime(self.expired_date, "%d-%m-%Y")
537         now = datetime.now()
538         if expired < now: return "Expired"
539         sisa = expired - now
540         return f"{{(exp_date - now).days}} hari"
541 ✓     except:
542         return "Data tidak valid"
543
```

Gambar 3. 19 Logika Perhitungan Status Membership

Fungsi `load_expired` & `get_remaining_days`:

a. **Tujuan Kode:**

- `load_expired`: Membaca database untuk melihat kapan membership pengguna berakhir.
- `get_remaining_days`: Menghitung sisa hari secara matematis untuk ditampilkan di profil.

b. **Penjelasan Sintaks Python:**

- `return row.get("expired", "")`: Metode `.get()` pada Dictionary lebih aman daripada `row["expired"]` karena jika kolom "expired" tidak ada, kode tidak akan error, melainkan mengembalikan string kosong "".
- `expired < now`: Perbandingan objek waktu. Python bisa membandingkan dua tanggal secara langsung untuk menentukan mana yang lebih lampau.
- `(exp_date - now).days`: Pengurangan dua objek tanggal (`datetime`) menghasilkan objek `timedelta`. Atribut `.days` mengambil selisih hari dari hasil pengurangan tersebut.

3.1.16 Inisialisasi Jendela Top Up dan Logika Penambahan Saldo (Top Up) (Baris 544-590)

```
547 class TopUpApp:
548     def __init__(self, root, main_app,):
549         self.root = root
550         self.main_app = main_app
551         self.root.title("TopUp - GymEmas")
552         center_window(self.root, 300, 400)
553
554         self.bgtopup = safe_open("bg_topupnew.png", size=(300, 400))
555         self.bgphoto = ctk.CTkImage(light_image=self.bgtopup, size=(300, 400))
556         self.bglabel = ctk.CTkLabel(self.root, image=self.bgphoto, text="")
557         self.bglabel.place(x=0, y=0, relwidth=1, relheight=1)
558
559         ctk.CTkLabel(self.root, text="Masukkan Jumlah Top up", font=("Crimson Text", 12, "bold"),
560                     fg_color=■ "#ffbd59", text_color=(□ "#000000"), bg_color=■ "#ffbd59").place(x=80, y=133)
561
562         self.amount_entry = ctk.CTkEntry(self.root, width=111, height=21, border_color=(■ "#FFFFFF"),
563                                         placeholder_text="0", bg_color=(■ "#ffbd59"), fg_color=(□ "#767676"))
564         self.amount_entry.place(x=94, y=164)
565
566         ctk.CTkButton(self.root, text="topup", width=68, height=24, corner_radius=10,
567                     text_color=(■ "#FFFFFF"), fg_color=(□ "#000000"), hover_color=(□ "#6b6262"),
568                     bg_color=(■ "#ffbd59"), command=self.topup).place(x=115, y=233)
569
570         home = safe_open("hometopup.png", size=(262, 49))
571         self.homephoto = ctk.CTkImage(light_image=home, size=(262, 49))
572
573
574
575     def topup(self):
576         try:
577             amount = int(self.amount_entry.get())
578             if amount <= 0: raise ValueError
579             self.main_app.saldo += amount
580             saldo_baru = self.main_app.saldo
581             self.main_app.update_saldo(saldo_baru)
582             save_user_data(self.main_app.username, "", "", saldo_baru)
583             tkmb.showinfo("Top-Up Berhasil", f"Saldo berhasil ditambahkan sebesar Rp {amount:,}")
584             self.go_back()
585         except ValueError:
586             tkmb.showerror("Error", "Masukkan jumlah yang valid.")
587
588     def go_back(self):
589         self.root.destroy()
590
```

Gambar 3. 20 Inisialisasi Jendela Top Up

Class `TopUpApp` - Method `__init__`:

- Fungsi Aplikasi: Membuat jendela kecil *pop-up* yang memungkinkan pengguna memasukkan nominal uang. Jendela ini didesain agar fokus pengguna tertuju ke sini (*modal window*), sehingga mereka tidak bisa mengklik menu lain sebelum selesai top up.
- Penjelasan Sintaks Python:
 - `self.main_app = main_app`: Menyimpan referensi ke jendela Menu Utama. Ini teknik komunikasi antar-class; jendela Top Up perlu "tahu" siapa yang memanggilnya agar bisa mengirim data saldo balik ke sana.

- `bg_color="#ffbd59"`: Mengatur warna latar belakang widget secara spesifik menggunakan kode Hexadecimal (kuning emas), konsisten dengan tema "GymEmas".
- `center_window(..., 300, 400)`: Memanggil fungsi bantuan yang sudah didefinisikan di awal kode untuk memastikan jendela kecil ini muncul tepat di tengah layar monitor.

Fungsi `topup(self)`:

- a) Fungsi Aplikasi: Memproses input uang dari pengguna. Fungsi ini mengubah teks menjadi angka, memvalidasinya (harus positif), menambahkan ke saldo saat ini, menyimpan ke database, dan memperbarui tampilan saldo di menu utama seketika.
- b) Penjelasan Sintaks Python:
 - `int(self.amount_entry.get())`: Mengambil teks dari kotak input dan mengonversinya menjadi tipe data *Integer* (bilangan bulat) agar bisa dijumlahkan.
 - `if amount <= 0: raise ValueError`: *Error Raising*. Kita sengaja memicu error jika pengguna memasukkan angka negatif atau nol. Hal ini akan melempar program ke blok `except` di bawahnya.
 - `self.main_app.update_saldo(...)`: Memanggil metode milik objek `GymMenu` (jendela utama) untuk memperbarui teks "Rp ..." di layar secara *real-time* tanpa perlu menutup aplikasi.
 - `tkmb.showinfo(...)`: Menampilkan pesan sukses menggunakan *messagebox* standar Tkinter.

3.1.17 Tampilan Profil dan Pengambilan Data Spesifik (Baris:591-637)

```
class ProfilePage:
    def __init__(self, root, username):
        self.root = root
        self.username = username
        center_window(self.root, 388, 824)
        self.root.title("Profil Pengguna - GymEmas")

        self.bgprofile = safe_open("bg profile.png", size=(688, 1424))
        self.bgphoto = ctk.CTkImage(light_image=self.bgprofile, size=(388, 824))
        self.bglabel = ctk.CTkLabel(self.root, image=self.bgphoto, text="")
        self.bglabel.place(x=0, y=0, relwidth=1, relheight=1)

        self.user_data = self.get_user_data()

        ctk.CTkLabel(self.root, text="Profil", font=("Crimson Text", 32),
                     fg_color="■", text_color=(□"#000000"), bg_color="■").place(x=150, y=35)

        ctk.CTkLabel(self.root, text=f"Username : {self.user_data['username']}", font=("Crimson Text", 18),
                     fg_color="■", text_color=(□"#000000"), bg_color="■").place(x=150, y=70)

        ctk.CTkLabel(self.root, text=f"Membership : {self.user_data['item'] or '-'}", font=("Crimson Text", 18),
                     fg_color="■", text_color=(□"#000000"), bg_color="■").place(x=150, y=95)

        ctk.CTkLabel(self.root, text=f"Saldo : Rp {int(self.user_data['saldo']):,}.replace(", ", ".)",
                     font=("Crimson Text", 18), fg_color="■", text_color=(□"#000000"), bg_color="transparent").place(x=150, y=120)

        exp = self.user_data["expired"] if self.user_data["expired"] else "-"
        ctk.CTkLabel(self.root, text=f"Expired : {exp}", font=("Crimson Text", 18),
                     fg_color="■", text_color=(□"#000000"), bg_color="transparent").place(x=150, y=145)

        sisa = self.get_remaining_days()
        ctk.CTkLabel(self.root, text=f"Sisa Waktu : {sisa}", font=("Crimson Text", 18),
                     fg_color="■", text_color=(□"#000000"), bg_color="transparent").place(x=150, y=170)

        ctk.CTkButton(self.root, text="Logout", fg_color=□"#B90000",
                     command=self.logout).place(x=125, y=675)

    def get_user_data(self):
        with open("users.csv", "r", encoding="utf-8") as file:
            reader = csv.DictReader(file)
            for row in reader:
                if row["username"] == self.username:
                    return row
        return {"username": "-", "item": "-", "saldo": "0", "expired": "-"}
```

Gambar 3. 21 Tampilan Profil dan Pengambilan Data Spesifik

Class `ProfilePage` - Method `__init__`:

- Fungsi Aplikasi: Menyusun tata letak halaman profil. Halaman ini menampilkan data sensitif pengguna seperti Username, Jenis Membership yang aktif, Saldo tersisa, dan Tanggal Expired secara rapi di atas gambar latar belakang.
- Penjelasan Sintaks Python:
 - `self.user_data['item'] or '-'`: Operator Logika *Short-circuit*. Jika data membership (`item`) kosong atau `None`, Python otomatis mengambil nilai cadangan setelah `or`, yaitu `"-"`. Ini berguna untuk mencegah tampilan kosong yang membingungkan.
 - `bg_color="transparent"`: Fitur unggulan library *CustomTkinter* yang membuat latar belakang label teks menjadi tembus pandang, sehingga menyatu sempurna dengan gambar background di belakangnya.

3.1.18 Logika Sisa Hari dan Logout (Baris 639-653)

```
639     def get_remaining_days(self):
640         try:
641             expired = self.user_data["expired"]
642             if not expired: return "Tidak ada membership"
643             exp_date = datetime.strptime(expired, "%d-%m-%Y")
644             now = datetime.now()
645             if exp_date < now: return "Expired"
646             return f"{{(exp_date - now).days}} hari"
647         except:
648             return "Data tidak valid"
649
650     def logout(self):
651         self.root.destroy()
652         login_app.deiconify()
653         menu_app.withdraw()
```

Gambar 3. 22 Logika Sisa Hari dan Logout

Fungsi `get_remaining_days`:

- a) Fungsi Aplikasi: Menghitung sisa masa aktif membership secara otomatis. Jika tanggal sudah lewat, status berubah menjadi "Expired". Jika data tanggal rusak, sistem memberikan pesan "Data tidak valid" alih-alih menutup aplikasi (*crash*).
- b) Penjelasan Sintaks Python:
 - try...except: Blok penanganan error. Python mencoba menjalankan perhitungan tanggal di dalam blok try. Jika terjadi kesalahan format tanggal, blok except akan berjalan untuk mencegah error sistem.
 - datetime.strptime(..., "%d-%m-%Y"): Mengubah teks tanggal dari database menjadi objek waktu agar bisa dikurangi dengan waktu sekarang (now).
 - (exp_date - now).days: Operasi matematika tanggal untuk mendapatkan selisih hari.

Fungsi `logout`:

- a) Fungsi Aplikasi: Mengakhiri sesi pengguna dengan aman. Jendela profil ditutup, dan pengguna dikembalikan ke layar Login.
- b) Penjelasan Sintaks Python:
 - self.root.destroy(): Menghapus jendela profil dari memori komputer sepenuhnya.
 - login_app.deiconify(): Memunculkan kembali jendela Login yang sebelumnya disembunyikan.

3.1.19 Eksekusi Program Utama (Main Program), (Baris 655-670)

```
658     if __name__ == "__main__":
659         ctk.set_appearance_mode("System")
660         ctk.set_default_color_theme("blue")
661
662         login_app = ctk.CTk()
663         LoginApp(login_app, csv_file_path)
664         init_csv()
665
666         menu_app = ctk.CTkToplevel(login_app)
667         gym_menu = GymMenu(menu_app, "")
668         menu_app.withdraw()
669
670         login_app.mainloop()
```

Gambar 3. 23 Eksekusi Program Utama (Main Program)

Konfigurasi Tema & Inisialisasi Root:

a) Fungsi Aplikasi:

- Mengatur tampilan visual agar mengikuti pengaturan sistem komputer pengguna (Mode Gelap/Terang).
- Menyiapkan jendela dasar (`login_app`) yang menjadi pondasi seluruh aplikasi.
- Memastikan file database `users.csv` sudah siap sebelum aplikasi terbuka.

b) Penjelasan Sintaks Python:

- `if __name__ == "__main__":`: Baris sakral di Python. Kode di bawah blok ini hanya akan dijalankan jika file ini dijalankan secara langsung (di-klik atau di-run), bukan jika file ini diimpor sebagai modul oleh file lain.
- `ctk.set_appearance_mode("System")`:
Memerintahkan library CustomTkinter untuk mendeteksi apakah Windows/Mac pengguna sedang mode Dark atau Light, lalu menyesuaikan warna aplikasi secara otomatis.
- `ctk.set_default_color_theme("blue")`: Mengatur warna aksen utama (tombol, border) menjadi biru.

Pre-loading Jendela & Mainloop:

a) Fungsi Aplikasi:

- Aplikasi ini menggunakan teknik *Pre-loading*. Jendela Menu Utama (`menu_app`) sebenarnya sudah dibuat bersamaan dengan Login, tetapi langsung disembunyikan (`.withdraw()`). Tujuannya agar saat pengguna klik "Login", menu utama muncul instan tanpa .
- Menjaga aplikasi tetap hidup dan responsif terhadap klik mouse atau keyboard.

b) Penjelasan Sintaks Python:

- `ctk.CTk()`: Membuat objek root window utama. Jika objek ini dihancurkan, seluruh aplikasi mati.
- `ctk.CTkToplevel(login_app)`: Membuat jendela sekunder yang "menumpang" pada jendela root.
- `login_app.mainloop()`: Fungsi ini menjalankan *infinite loop* (perulangan tak terbatas). Ia menunggu event (klik, ketikan) dari pengguna. Tanpa baris ini, jendela aplikasi hanya akan muncul sekejap lalu langsung tertutup.

3.1.20 Fitur Tambahan: Generator QR Code dan Exit Handler

Fungsi Generator QR Code

```
111 v def generate_qr(username, expired):
112     data = f"GYMEMAS|USER:{username}|EXPIRED:{expired}"
113
114 v     qr = qrcode.QRCode(
115         version=1,
116         error_correction=qrcode.constants.ERROR_CORRECT_L,
117         box_size=6,
118         border=2,
119     )
120     qr.add_data(data)
121     qr.make(fit=True)
122
123     img = qr.make_image(fill_color="black", back_color="white")
124     return img
125
```

Gambar 3. 24 Fungsi Generator QR Code

Fungsi `generate_qr(username, expired)` :

a) Fungsi Aplikasi:

- Menciptakan kode QR unik untuk setiap pengguna yang berisi data terenkripsi sederhana (`GYMEMAS|USER...`).
- QR Code ini ditampilkan di halaman profil sebagai kartu identitas digital yang dapat dipindai untuk memverifikasi status keanggotaan saat masuk ke gym.

b) Penjelasan Sintaks Python:

- `import qrcode`: Menggunakan pustaka eksternal untuk mengubah teks menjadi matriks gambar kode QR.
- `qr.add_data(data)` : Memasukkan string data (gabungan username dan tanggal expired) ke dalam objek QR.
- `qr.make_image(...)` : Merender objek QR menjadi file gambar hitam-putih yang bisa ditampilkan oleh Python (menggunakan PIL).

Fungsi Protokol Exit Application

```
645 def exit_app():  
646     sys.exit(0)  
647  
648 login_app.protocol('WM_DELETE_WINDOW', exit_app)  
649 menu_app.protocol('WM_DELETE_WINDOW', exit_app)  
650
```

Gambar 3. 25 Fungsi Protokol Exit Application

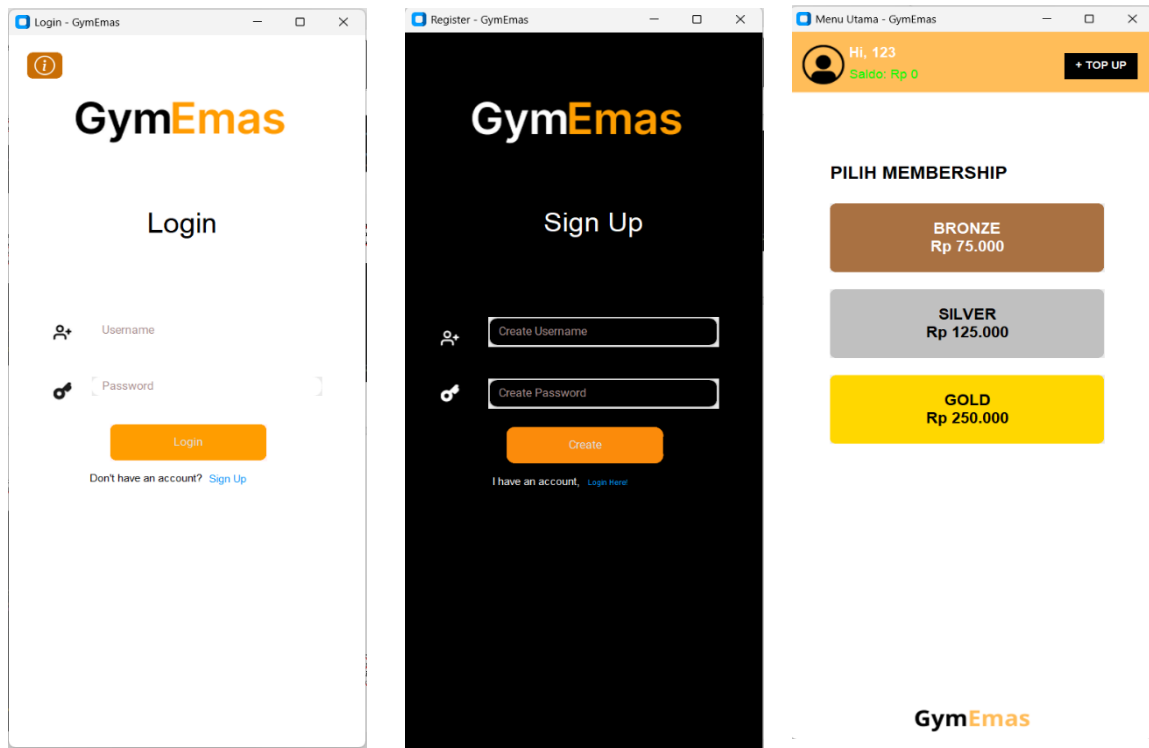
a) Fungsi Aplikasi `exit_app`:

- Memastikan aplikasi tertutup sempurna saat pengguna menekan tombol "X" di pojok kanan atas jendela.
- Mencegah aplikasi tetap berjalan di latar belakang (background process) yang dapat memakan memori komputer.

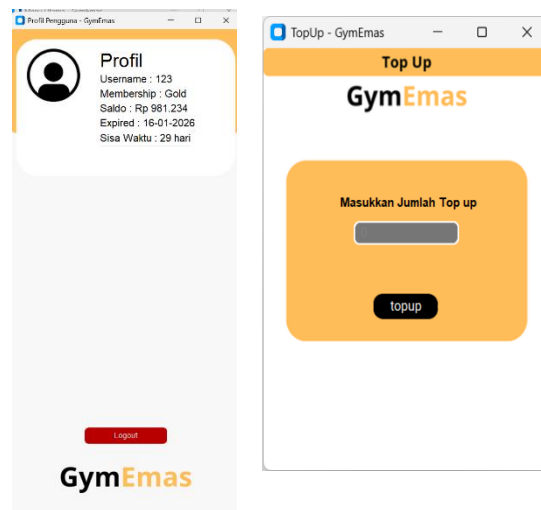
b) Penjelasan Sintaks Python:

- `sys.exit(0)`: Perintah sistem untuk menghentikan skript Python secara paksa dan bersih. Angka 0 menandakan program berhenti tanpa error.
- `protocol('WM_DELETE_WINDOW', exit_app)` : Event Listener khusus Tkinter. Kode ini "menangkap" aksi saat tombol Close (X) diklik, dan mengalihkannya untuk menjalankan fungsi `exit_app` alih-alih hanya menutup tampilan grafisnya saja.

3.2 TAMPILAN APLIKASI (*SCREENSHOT*)



Gambar 3. 26-28 Foto Tampilan Halaman Awal *Login, Register*, dan juga Halaman Utama



Gambar 3. 29-30 Foto Tampilan Halaman *Top Up* dan Profil

BAB 4

PENUTUP

4.1 KESIMPULAN

Berdasarkan implementasi dan pengujian sistem, dapat disimpulkan bahwa aplikasi Sistem Pendataan Keanggotaan *Gym* berhasil memenuhi tujuan perancangannya. Aplikasi ini mampu menggantikan pencatatan manual dengan sistem database digital berbasis CSV yang aman. Fitur otomatisasi tanggal terbukti efektif dalam memberikan informasi masa aktif keanggotaan secara akurat, mengurangi potensi kesalahan perhitungan manusia. Selain itu, antarmuka yang dibangun dengan CustomTkinter memberikan pengalaman pengguna yang modern dan responsif.

4.2 SARAN PENGEMBANGAN

Untuk pengembangan selanjutnya agar aplikasi ini semakin optimal, disarankan beberapa hal berikut:

- Integrasi Pembayaran QRIS: Mengembangkan fitur *Top Up* agar terhubung dengan *payment gateway* sungguhan (seperti QRIS atau Virtual Account), sehingga saldo dapat terisi otomatis tanpa input manual dari admin.
- Keamanan Data: Menerapkan enkripsi pada penyimpanan *password* agar data pengguna lebih terlindungi dari akses yang tidak sah.
- Fitur Notifikasi: Menambahkan sistem peringatan dini jika masa keanggotaan pengguna akan segera habis dalam waktu dekat.

DAFTAR PUSTAKA

Enterprise, J. (2019). *Otodidak Pemrograman Python*. Jakarta: Elex Media Komputindo.

Kurniawan, A. (2022). *Dasar Pemrograman Python untuk Pemula*. Jakarta: Elex Media Komputindo.

Python Software Foundation. (2024). *Python 3.14 Documentation: CSV File Reading and Writing*. <https://docs.python.org/3/library/csv.html>

Python Software Foundation. (2024). *Python 3.14 Documentation: Datetime - Basic date and time types*. <https://docs.python.org/3/library/datetime.html>

Schimansky, T. (2023). *CustomTkinter Documentation: UI-Library for Python*. GitHub.