

LAPORAN PRAKTIKUM
POSTTEST (8)
ALGORITMA PEMROGRAMAN DASAR



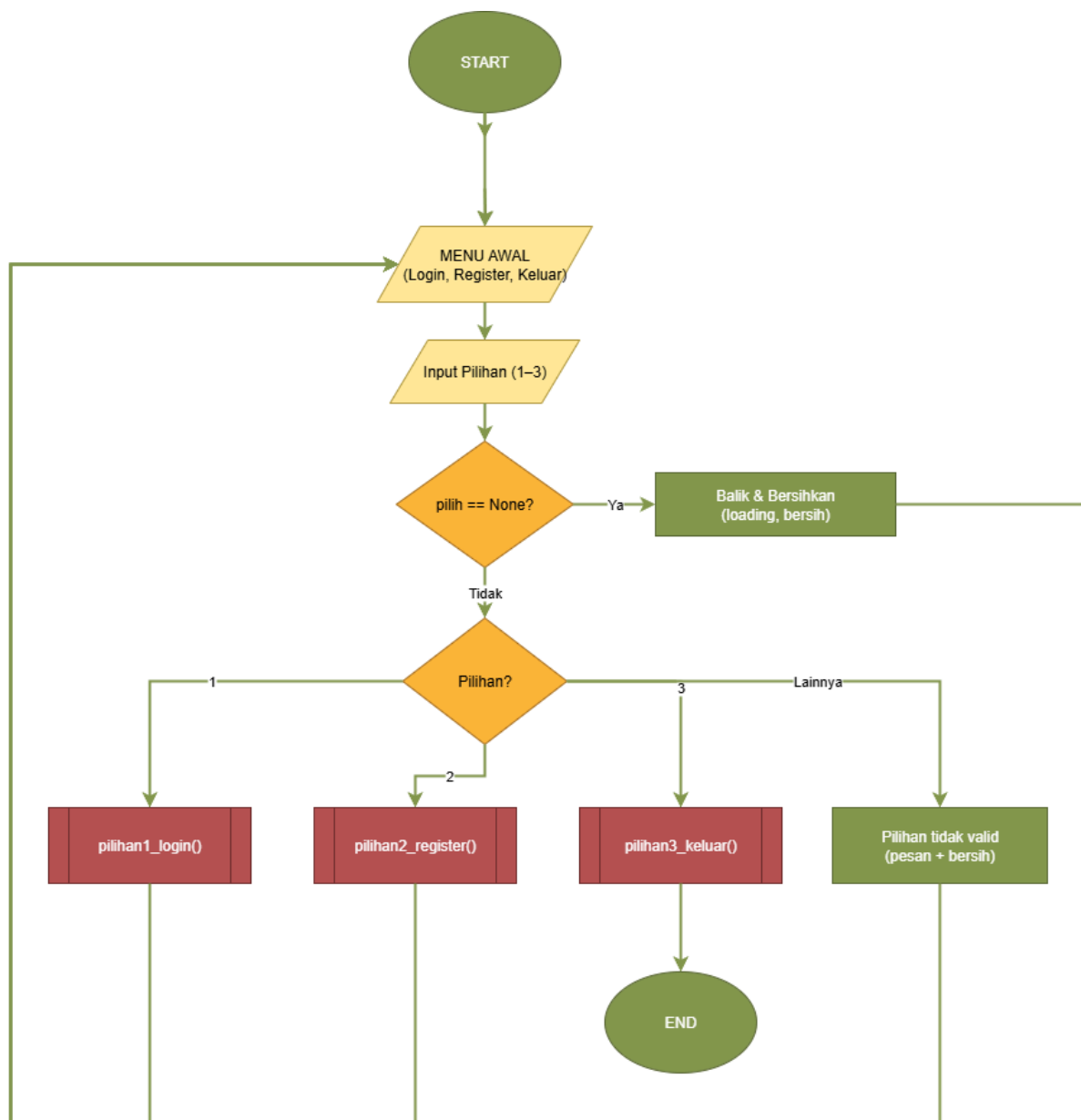
Disusun oleh:

Athasyahri Syawal Fahrezy (2509106045)

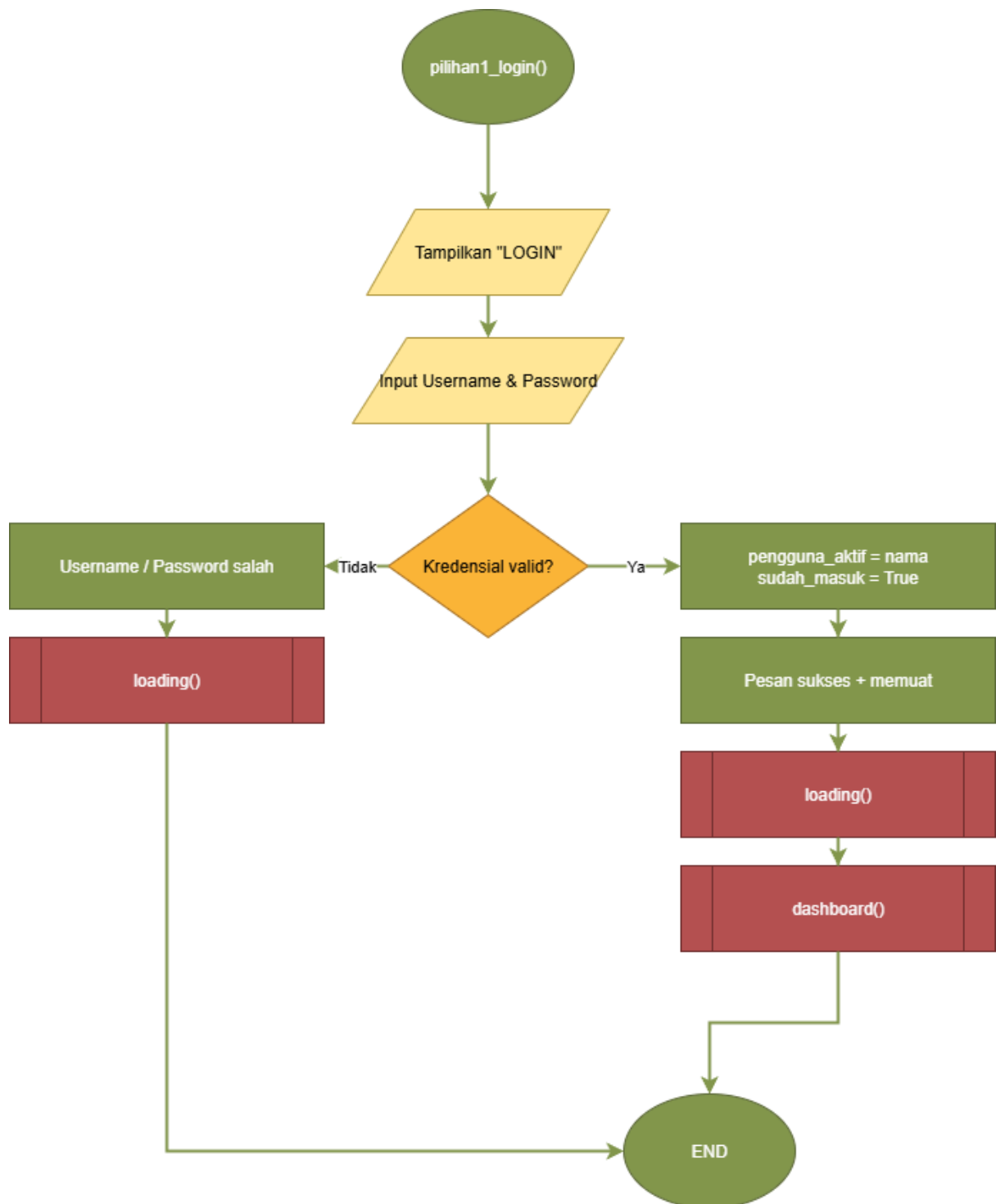
Kelas (A2'25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

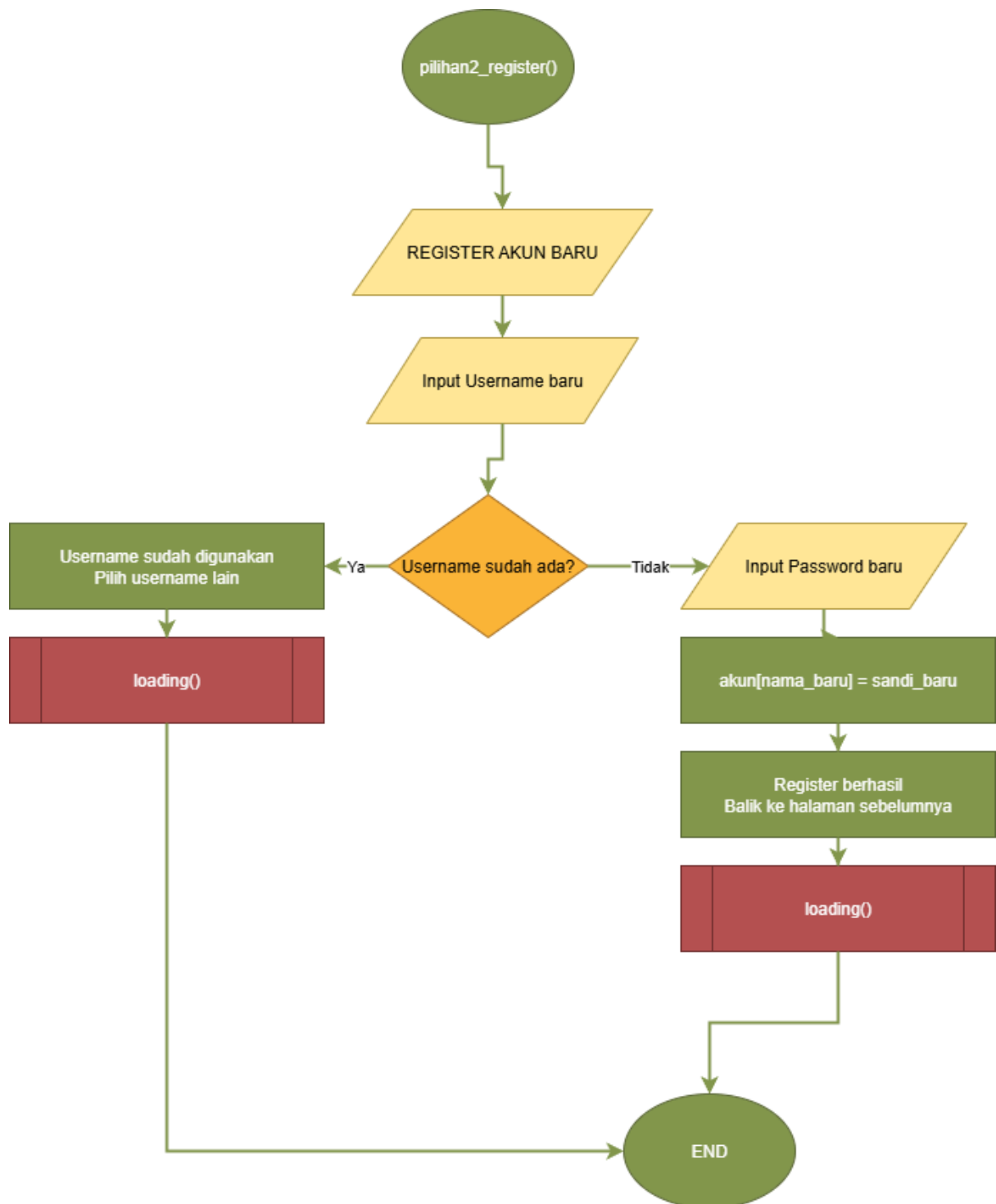
1. Flowchart



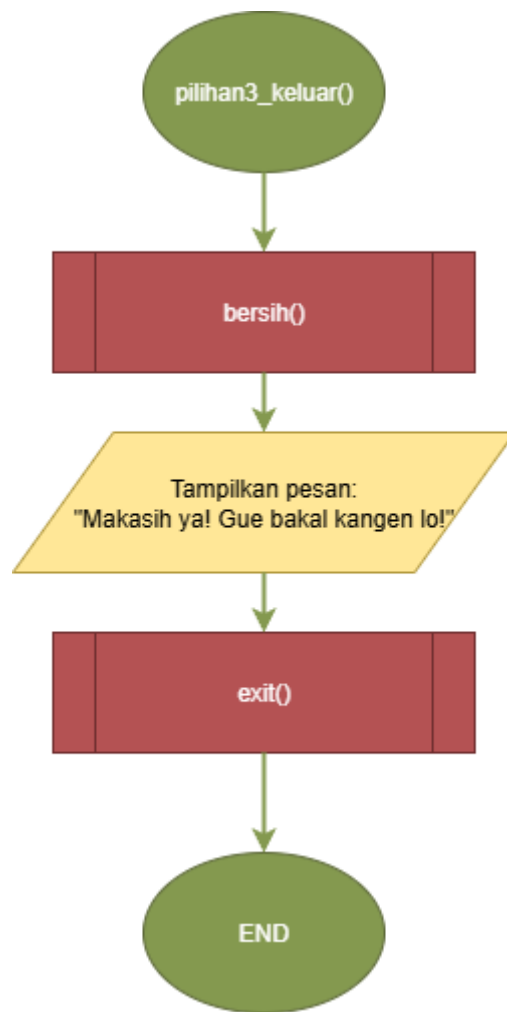
Gambar 1.1 Flowchart Dict & Menu Awal



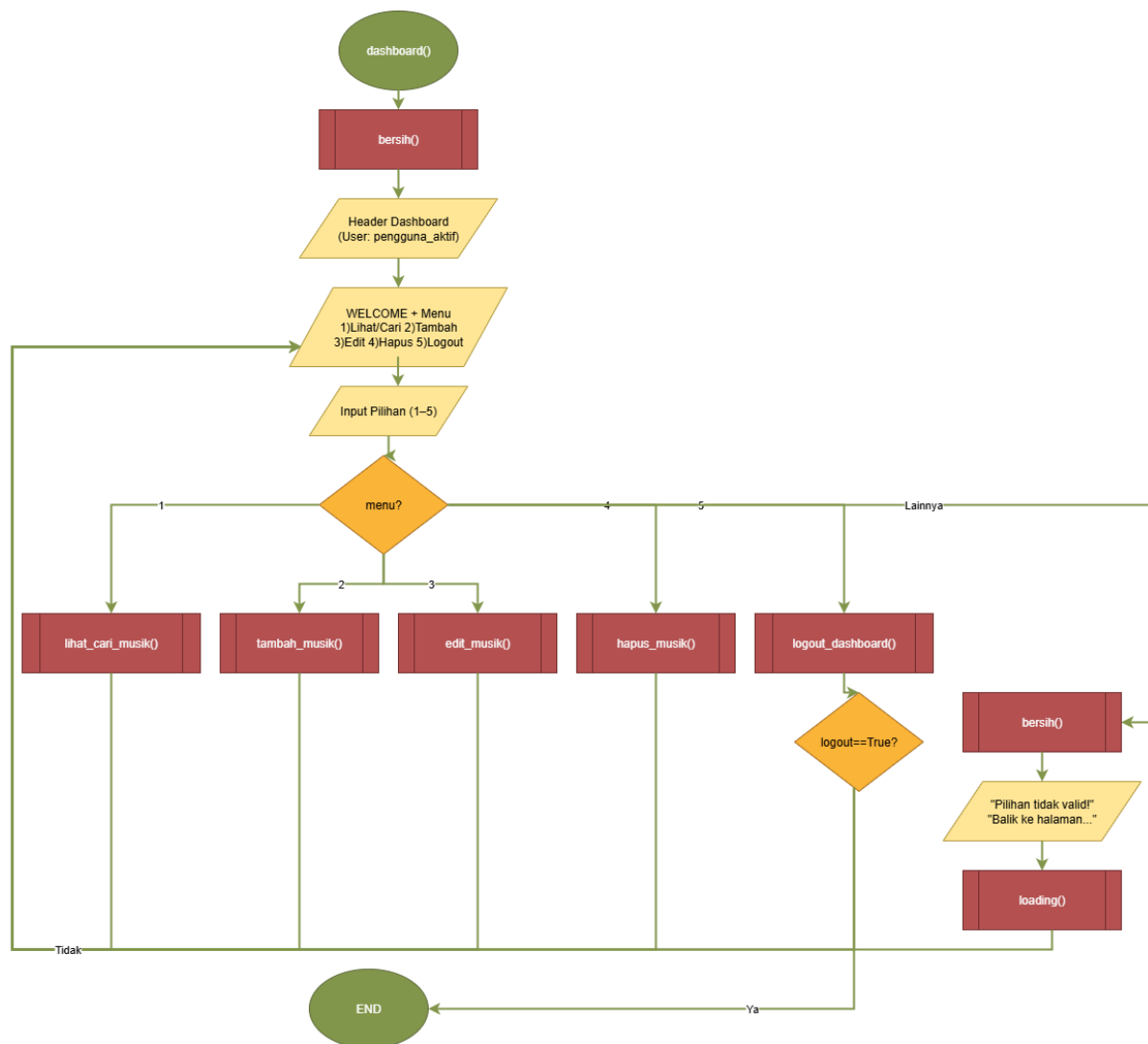
Gambar 1.2 Flowchart pilihan1_login()



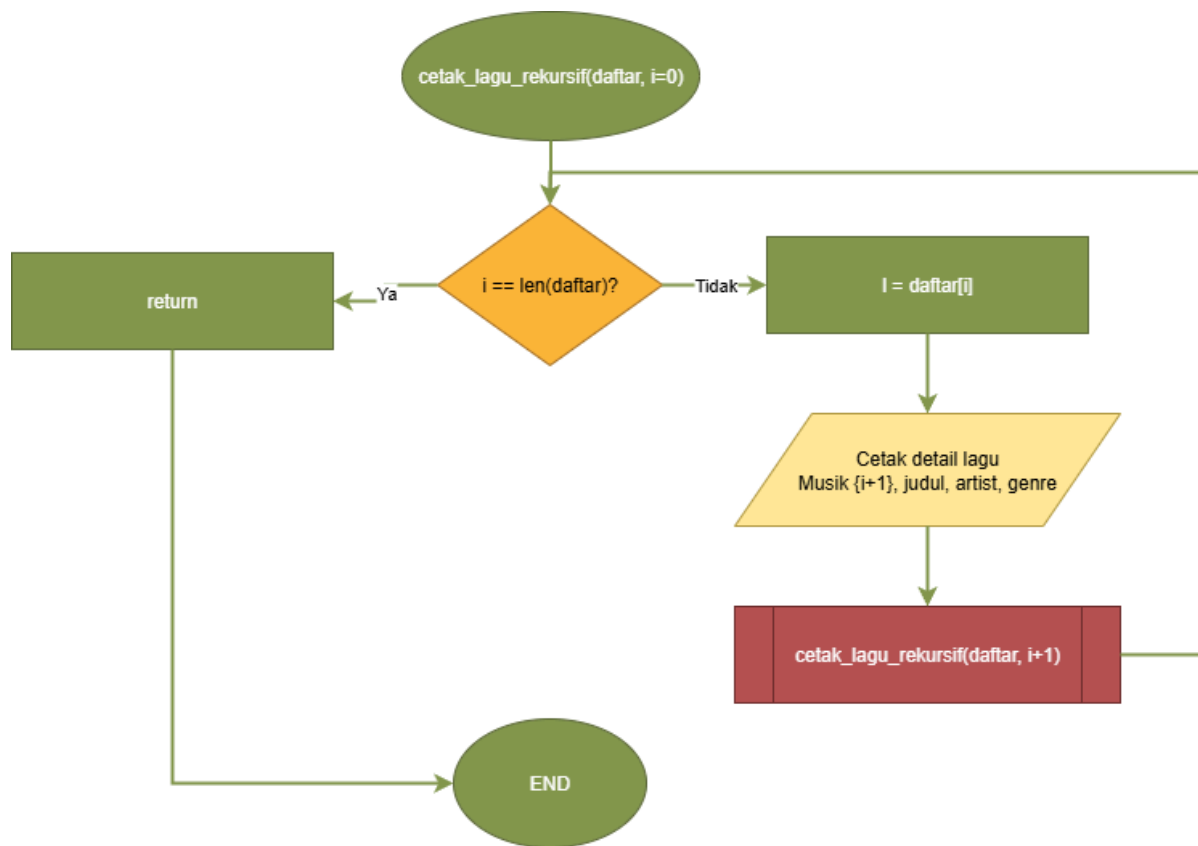
Gambar 1.3 Flowchart pilihan2_regitser()



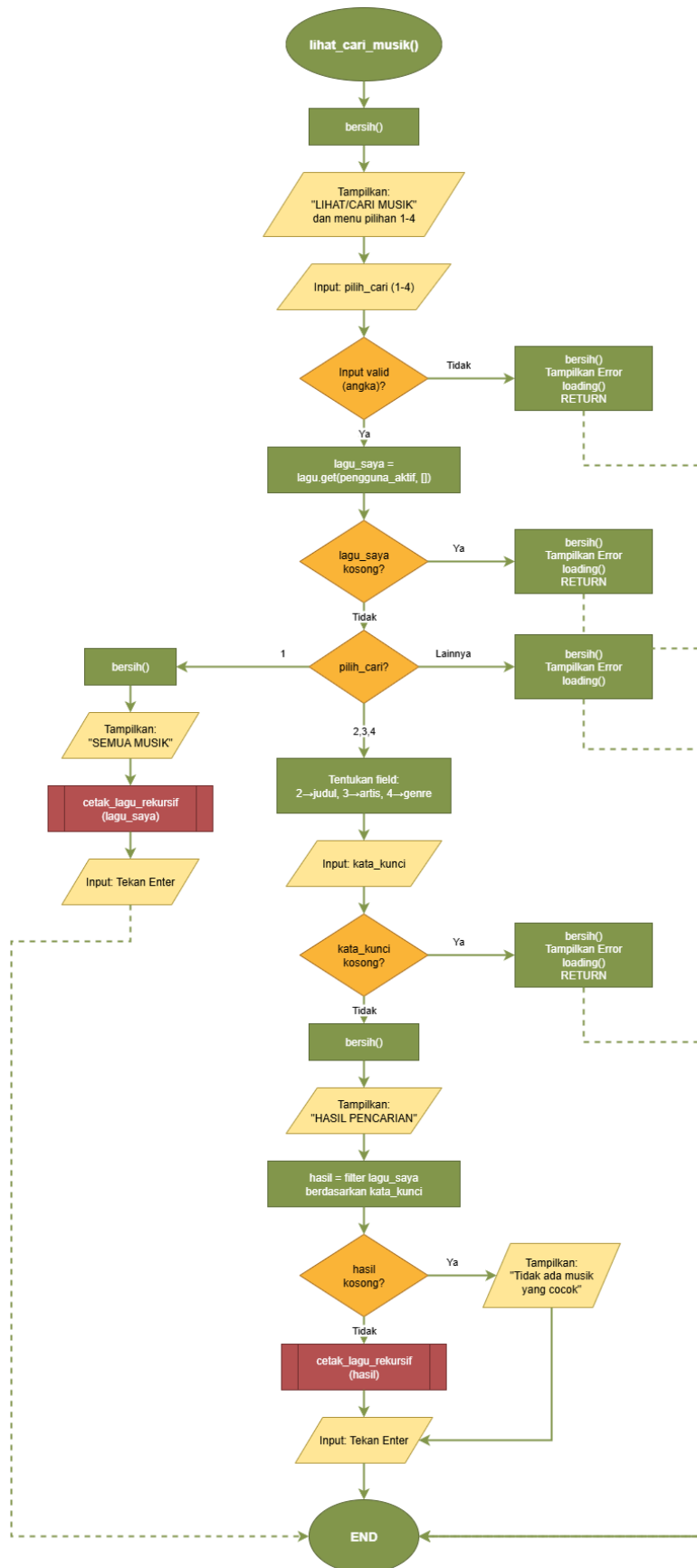
Gambar 1.3 Flowchart pilihan3_keluar()



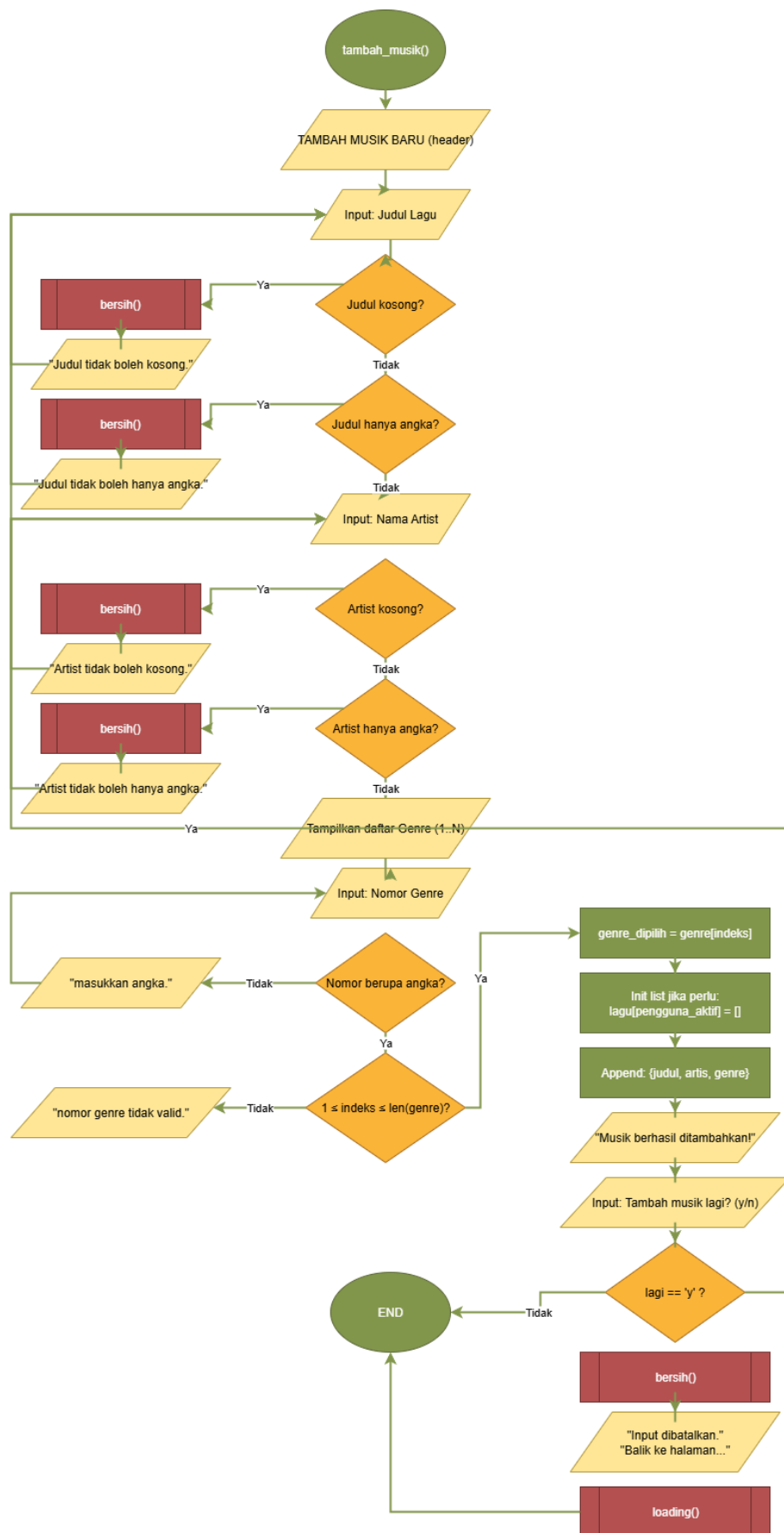
Gambar 1.4 Flowchart dashboard()



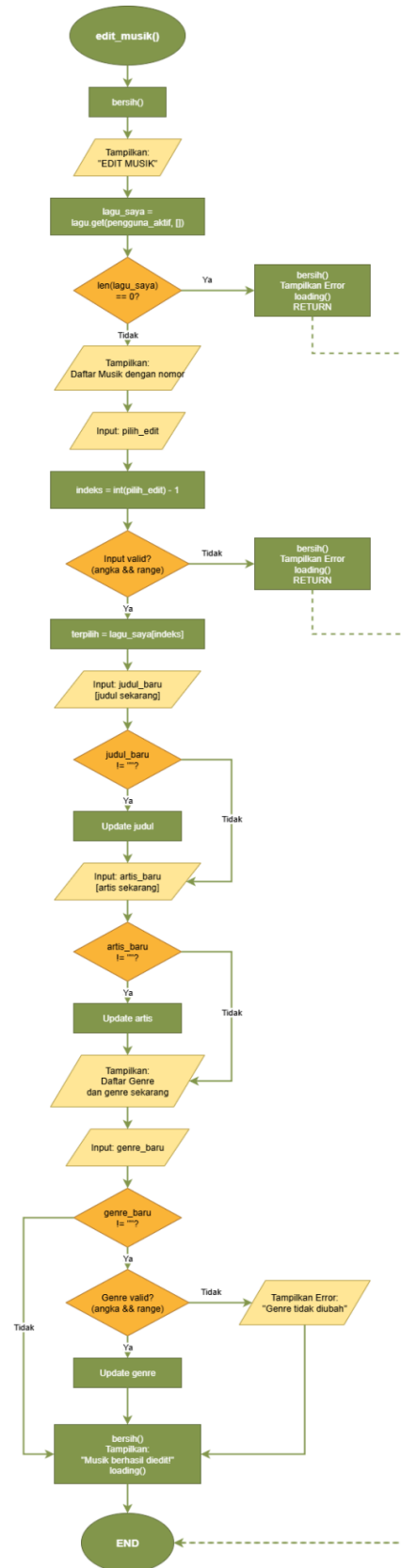
Gambar 1.5 Flowchart cetak_lagu_rekursif()



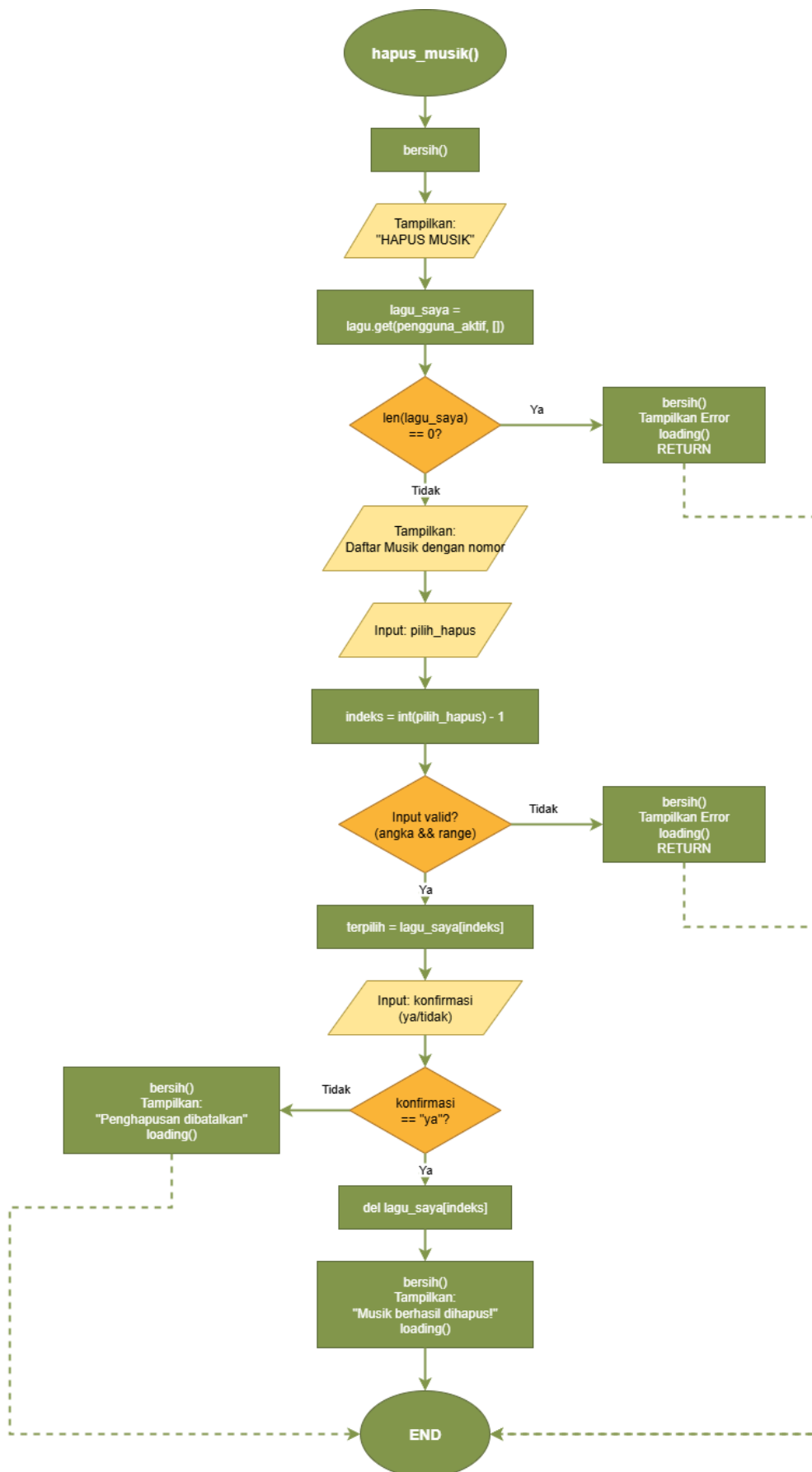
Gambar 1.6 Flowchart lihat_cari_musik()



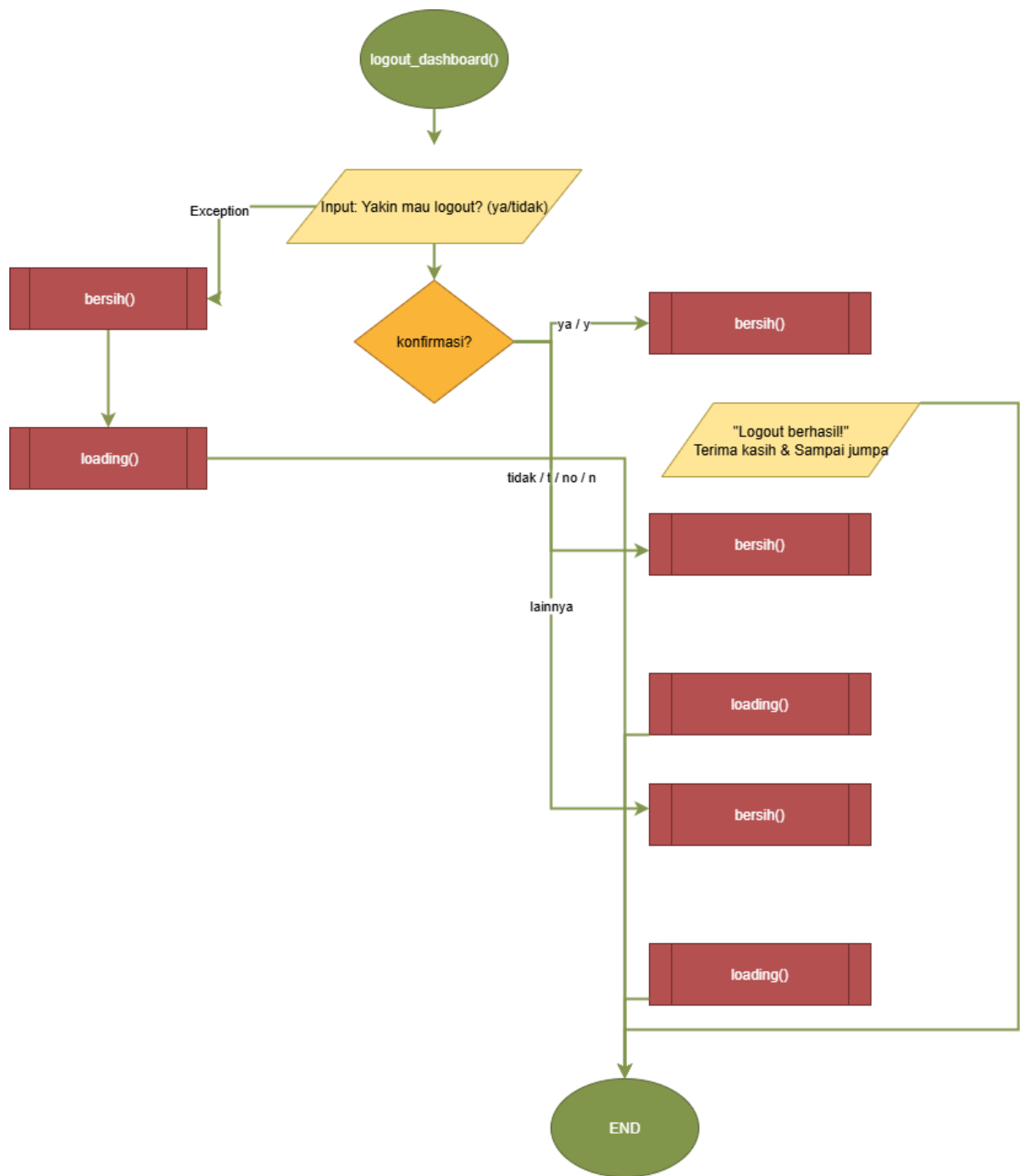
Gambar 1.7 Flowchart tambah_musik()



Gambar 1.8 Flowchart edit_musik()



Gambar 1.9 Flowchart hapus_musik()



Gambar 1.10 Flowchart logout_dashboard()

2. Deskripsi Singkat Program

Program ini adalah aplikasi manajemen musik dengan alur Login/Register → Dashboard. Saya menyimpan data di tiga struktur: akun (username & password), lagu (pemilik, judul, artist, genre), dan genre (daftar pilihan statis), serta menandai pengguna yang aktif di pengguna_aktif. Dalam Dashboard, pengguna bisa CRUD: lihat/cari musik (berdasarkan judul/artist/genre), tambah musik baru (dengan validasi input & pilihan genre), edit (ubah sebagian/semua data), dan hapus (dengan konfirmasi). Terakhir, opsi Logout/Keluar menutup program.

3. Source Code

3.1. state.py

Di bagian kode ini saya nyiapin beberapa dict yang nantinya dipakai program buat jalan. akun dipakai buat nyimpen data login dan register, lagu buat nyimpen data musik lengkap sama pemiliknya, genre isinya daftar genre yang udah disiapkan dari awal, dan pengguna_aktif saya pakai buat nyatet siapa yang lagi login saat program dijalankan.

```
akun = {"pares": "045"} # DICT MENYIMPAN DATA LOGIN DAN REGISTER

lagu = {
    "pares": [{"judul": "Bohemian Rhapsody", "artis": "Queen", "genre":
"Rock"}]
} # DICT SIMPAN DATA MUSIK

genre = {
    1: "Pop",
    2: "Rock",
    3: "Jazz",
    4: "Hip-Hop",
    5: "Electronic",
    6: "Classical",
    7: "R&B",
    8: "Country",
} # DICT MENYIMPAN GENRE

pengguna_aktif = ""
sudah_masuk = False
```

3.2. utils.py

Disini saya bikin beberapa function sebagai handler misa cek_input_kosong, clear dll

```
import os
import time

def bersih():
    os.system("cls" if os.name == "nt" else "clear")

def loading():
    hitung = 3
    while hitung > 0:
        print(hitung, "...")
        time.sleep(1)
        hitung -= 1

def input_angka(prompt):
    try:
        angka = int(input(prompt))
    except ValueError:
        print("input yang kamu masukkan bukan angka, harus angka broo!!")
        return None
    else:
        print(f"Angka yang kamu input : {angka}")
        return angka
    finally:
        print("Kelar dah")

def cek_input_kosong(pesan):
    while True:
        try:
            nilai = input(pesan).strip()
        except:
            print("\nInput dibatalkan. Silakan coba lagi.")
            continue
        if nilai == "":
            print("Input tidak boleh kosong. Coba lagi.")
        else:
            return nilai
```

3.3. menu.py

Di bagian ini saya bikin menu awal yang terus ditampilkan selama sudah_masuk masih False (artinya belum login). Program nunjukin opsi Login, Register, atau Keluar, lalu saya ambil input pengguna lewat pilih = input("Pilih menu (1-3): ") untuk nentuin tindakan selanjutnya.

```
from utils import bersih, loading, input_angka
import state
from auth import pilihan1_login, pilihan2_register, pilihan3_keluar

def menuawal():
    while state.sudah_masuk is False:
        print("\n--- MENU AWAL ---")
        print("1. Login")
        print("2. Register")
        print("3. Keluar")

        pilih = input_angka("\nPilih menu (1-3): ")

        if pilih is None:
            print("\nBalik ke halaman sebelumnya...")
            loading()
            bersih()
            continue

        if pilih == 1:
            pilihan1_login()
        elif pilih == 2:
            pilihan2_register()
        elif pilih == 3:
            pilihan3_keluar()
        else:
            bersih()
            print("\nPilihan tidak valid!")
            print("\nBalik ke halaman sebelumnya...")
            loading()
            bersih()
```


3.4. auth.py

Aplikasinya sederhana: pengguna diminta memasukkan username dan password, lalu sistem mencocokkannya dengan data di dict akun. Jika benar, kita set pengguna_aktif ke username dan sudah_masuk = True, tampilkan ucapan selamat disertai hitung mundur 3 detik. Kalau salah, tampilkan pesan error, tunggu 3 detik, lalu kembali ke menu. Begitu pengguna memilih opsi 3 (Keluar), tampilkan salam perpisahan dan program langsung berhenti dengan exit().

```
from utils import bersih, loading, cek_input_kosong
import state
from dashboard import dashboard

def pilihan1_login():
    bersih()
    print("\n" + "=" * 60)
    print("LOGIN")
    print("=" * 60)

    nama = cek_input_kosong("Username: ")
    sandi = cek_input_kosong("Password: ")

    dapatakun = False

    if nama in state.akun and state.akun[nama] == sandi:
        dapatakun = True
        state.pengguna_aktif = nama

    if dapatakun is True:
        state.sudah_masuk = True
        bersih()
        print("\nKamu berhasil masuk, WELCOME BROO", state.pengguna_aktif)
        print("\nMemuat halaman dalam...")
        loading()
        dashboard()

    else:
        bersih()
        print("\nUsername atau password salah, coba inget-inget dulu!!")
        print("\nKembali dalam...")
        loading()
        bersih()
```

```

def pilihan2_register():
    bersih()
    print("\n" + "=" * 60)
    print("REGISTER AKUN BARU")
    print("=" * 60)

    nama_baru = cek_input_kosong("Username baru: ")

    if nama_baru in state.akun:
        bersih()
        print("\nUsername sudah digunakan! Pilih username lain.")
        print("\nBalik ke halaman sebelumnya...")
        loading()
        bersih()
        return

    sandi_baru = cek_input_kosong("Password baru: ")

    state.akun[nama_baru] = sandi_baru
    bersih()
    print("\nRegister berhasil! Silakan login.")
    print("\nBalik ke halaman sebelumnya...")
    loading()
    bersih()

def pilihan3_keluar():
    bersih()
    print("\n" + "=" * 60)
    print("Makasih ya! Gue bakal kangen lo!")
    print("=" * 60)
    exit()

```

3.5. dashboard.py

Di bagian Menu Dashboard ini, saya masuk ke loop tak hingga yang menampilkan nama pengguna_aktif dan opsi utama: lihat/cari musik, tambah, edit, hapus, atau logout. Setelah itu, saya ambil input di menu = input("(1-5): ") untuk menentukan aksi yang akan dijalankan selanjutnya.

```
from utils import bersih, loading
import state

from music_read import lihat_cari_musik
from music_create import tambah_musik
from music_update import edit_musik
from music_delete import hapus_musik

def logout_dashboard():
    try:
        konfirmasi = input("\nYakin mau logout? (ya/tidak): ")
        konfirmasi.strip().lower()
    except (KeyboardInterrupt, EOFError):
        bersih()
        print("\nInput dibatalkan.")
        print("\nBalik ke halaman sebelumnya...")
        loading()
        return False

    if konfirmasi in ("ya", "y"):
        bersih()
        print("\nLogout berhasil!")
        print("\n" + "=" * 60)
        print("Terima kasih sudah menggunakan Manajemen Musik Personal!")
        print("Sampai jumpa lagi!")
        print("=" * 60)
        return True

    elif konfirmasi in ("tidak", "t", "no", "n"):
        bersih()
        print("\nLogout dibatalkan.")
        print("\nBalik ke halaman sebelumnya...")
        loading()
        return False

    else:
        bersih()
        print("\nInput tidak valid! Gunakan 'ya' atau 'tidak'.")
        print("\nBalik ke halaman sebelumnya...")
        loading()
        return False
```

```

def dashboard():
    while True:
        bersih()
        print("=" * 60)
        print("User:", state.pengguna_aktif)
        print("=" * 60)

        print("\n--- WELCOME TO DASHBOARD ---")
        print("1. Lihat/Cari musik?")
        print("2. Tambah musik baru")
        print("3. Edit musik")
        print("4. Hapus musik")
        print("5. Logout")

        menu = input("\nMasukkan pilihan (1-5): ")

        if menu == "1":
            lihat_cari_musik()
        elif menu == "2":
            tambah_musik()
        elif menu == "3":
            edit_musik()
        elif menu == "4":
            hapus_musik()
        elif menu == "5":
            if logout_dashboard():
                break
        else:
            bersih()
            print("\nPilihan tidak valid!")
            print("\nBalik ke halaman sebelumnya...")
            loading()

```

3.6. music_read.py

Di menu Read (Lihat/Cari Musik), saya ambil semua lagu milik pengguna_aktif dulu ke lagu_saya. Lalu ada submenu: 1) lihat semua, 2) cari judul, 3) cari artist, 4) cari genre—setiap pencarian pakai substring match (lowercase) dan hasilnya ditampilkan berurutan dengan nomor.

```
from utils import bersih, loading
import state

def cetak_lagu_rekursif(daftar, i=0):
    if i == len(daftar):
        return
    l = daftar[i]
    print(f"\nMusik {i+1}")
    print("-" * 40)
    print("Judul   :", l.get("judul", ""))
    print("Artist  :", l.get("artis", ""))
    print("Genre   :", l.get("genre", ""))
    cetak_lagu_rekursif(daftar, i + 1)

def lihat_cari_musik():
    bersih()
    print("=" * 60)
    print("LIHAT/CARI MUSIK")
    print("=" * 60)

    print("\n--- PILIHAN PENCARIAN ---")
    print("1. Lihat semua musik")
    print("2. Cari berdasarkan judul")
    print("3. Cari berdasarkan artist")
    print("4. Cari berdasarkan genre")

    try:
        pilih_cari = int(input("\nPilih jenis pencarian (1-4): "))
    except ValueError:
        bersih()
        print("\nInput harus angka 1-4.")
        print("\nBalik ke halaman sebelumnya...")
        loading()
        return

    lagu_saya = state.lagu.get(state.pengguna_aktif, [])
```

```

if not lagu_saya:
    bersih()
    print("\nBelum ada musik yang tersimpan.")
    print("\nBalik ke halaman sebelumnya...")
    loading()
    return

if pilih_cari == 1:
    bersih()
    print("=" * 60)
    print("SEMUA MUSIK")
    print("=" * 60)
    cetak_lagu_rekursif(lagu_saya)
    input("\nTekan Enter untuk kembali...")

elif pilih_cari in (2, 3, 4):
    field_map = {2: "judul", 3: "artis", 4: "genre"}
    field = field_map[pilih_cari]

    try:
        kata_kunci = input(f"\nMasukkan keyword {field}: ").strip()
        if not kata_kunci:
            raise ValueError("Keyword tidak boleh kosong.")
    except ValueError as e:
        bersih()
        print(f"\n{e}")
        print("\nBalik ke halaman sebelumnya...")
        loading()
        return

    bersih()
    print("=" * 60)
    print(f"HASIL PENCARIAN {field.upper()}")
    print("=" * 60)

    hasil = [l for l in lagu_saya if kata_kunci.lower() in
              l[field].lower()]

    if not hasil:
        print("\nTidak ada musik yang cocok.")
    else:
        cetak_lagu_rekursif(hasil)

    input("\nTekan Enter untuk kembali...")

```

```

else:
    bersih()
    print("\nPilihan tidak valid!")
    print("\nBalik ke halaman sebelumnya...")
    loading()

```

3.7. music_create.py

Di menu Tambah Musik, saya minta judul lalu validasi tidak kosong; lanjut minta artist dan validasi lagi. Setelah itu saya tampilkan daftar genre, pengguna pilih nomor (harus angka dan dalam rentang), lalu saya simpan ke dict lagu sebagai [pengguna_aktif, judul, artis, genre_terpilih] dan tampilkan pesan berhasil + hitung mundur singkat.

```

from utils import bersih, loading
import state

def tambah_musik():
    while True:
        try:
            print("\n=== TAMBAH MUSIK BARU ===")

            judul = input("Judul Lagu : ").strip()
            if not judul:
                bersih()
                print("Judul tidak boleh kosong.\n")
                continue
            if judul.isdigit():
                bersih()
                print("Judul tidak boleh hanya angka.\n")
                continue

            artis = input("Nama Artist: ").strip()
            if not artis:
                bersih()
                print("Artist tidak boleh kosong.\n")
                continue
            if artis.isdigit():
                bersih()
                print("Artist tidak boleh hanya angka.\n")
                continue

            print("\nPilih Genre:")
            for i in range(1, len(state.genre) + 1):
                print(f"{i}. {state.genre[i]}")
            pilih_genre = input("Nomor genre: ").strip()

```

```

try:
    indeks = int(pilih_genre)
except ValueError:
    print("masukkan angka.\n")
    continue

if indeks < 1 or indeks > len(state.genre):
    print("nomor genre tidak valid.\n")
    continue

genre_dipilih = state.genre[indeks]

if state.pengguna_aktif not in state.lagu:
    state.lagu[state.pengguna_aktif] = []
state.lagu[state.pengguna_aktif].append(
    {"judul": judul, "artis": artis, "genre": genre_dipilih}
)

print("\nMusik berhasil ditambahkan!\n")

lagi = input("Tambah musik lagi? (y/n): ").strip().lower()
if lagi != "y":
    break

except (KeyboardInterrupt, EOFError):
    bersih()
    print("\nInput dibatalkan.")
    print("\nBalik ke halaman sebelumnya...")
    loading()
    break

```


3.8. music_update.py

Di menu Edit Musik, saya ambil semua lagu milik pengguna_aktif; kalau kosong, saya beri info belum ada yang bisa diedit. Kalau ada, saya tampilkan daftar bernomor, pengguna pilih nomor (dicek harus angka & valid), lalu bisa mengubah judul, artist, dan genre (boleh tekan Enter untuk skip); perubahan disimpan dan ditampilkan pesan sukses.

```
from utils import bersih, loading
import state

def edit_musik():
    bersih()
    print("=" * 60)
    print("EDIT MUSIK")
    print("=" * 60)

    lagu_saya = state.lagu.get(state.pengguna_aktif, [])

    if len(lagu_saya) == 0:
        bersih()
        print("\nBelum ada musik yang bisa diedit.")
        print("\nBalik ke halaman sebelumnya...")
        loading()
        return

    print("\nDaftar Musik:")
    no = 1
    for l in lagu_saya:
        print(no, ".", l["judul"], "-", l["artis"])
        no += 1

    try:
        pilih_edit = input("\nPilih nomor musik yang mau diedit: ")
    except (KeyboardInterrupt, EOFError):
        bersih()
        print("\nInput dibatalkan.")
        print("\nBalik ke halaman sebelumnya...")
        loading()
        return
```

```

try:
    indeks = int(pilih_edit) - 1
except ValueError:
    bersih()
    print("\nInput harus berupa angka!")
    print("\nBalik ke halaman sebelumnya...")
    loading()
    return

if indeks < 0 or indeks >= len(lagu_saya):
    bersih()
    print("\nNomor tidak valid!")
    print("\nBalik ke halaman sebelumnya...")
    loading()
    return

terpilih = lagu_saya[indeks]

try:
    print("\nMasukkan data baru (tekan Enter jika tidak ingin
          mengubah):")
    judul_baru = input("Judul Lagu [" + terpilih["judul"] + "]: ")
    if judul_baru != "":
        terpilih["judul"] = judul_baru

    artis_baru = input("Nama Artist [" + terpilih["artis"] + "]: ")
    if artis_baru != "":
        terpilih["artis"] = artis_baru

    print("\nPilih Genre Baru:")
    i = 1
    while i <= len(state.genre):
        print(str(i) + ".", state.genre[i])
        i += 1
    print("Genre sekarang:", terpilih["genre"])

    genre_baru = input("Pilih nomor genre (atau Enter untuk tidak
                        ubah): ")

```

```

    if genre_baru != "":
        try:
            indeks_genre = int(genre_baru)
        except ValueError:
            print("\nInput harus berupa angka! Genre tidak diubah.")
        else:
            if 1 <= indeks_genre <= len(state.genre):
                terpilih["genre"] = state.genre[indeks_genre]
            else:
                print("\nNomor tidak valid! Genre tidak diubah.")

except (KeyboardInterrupt, EOFError):
    bersih()
    print("\nInput dibatalkan. Tidak ada perubahan yang disimpan.")
    print("\nBalik ke halaman sebelumnya...")
    loading()
    return

bersih()
print("\nMusik berhasil diedit!")
print("\nLoading...")
loading()

```

3.9. music_delete.py

Di menu Hapus Musik, saya ambil dulu semua lagu milik pengguna_aktif; kalau kosong, saya kasih info belum ada yang bisa dihapus. Kalau ada, saya tampilkan daftar bernomor, pengguna pilih nomor (dicek valid), lalu saya minta konfirmasi ya/tidak kalau “ya” lagu di-remove, kalau tidak, penghapusan dibatalkan.

```
from utils import bersih, loading
import state

def hapus_musik():
    bersih()
    print("=" * 60)
    print("HAPUS MUSIK")
    print("=" * 60)

    lagu_saya = state.lagu.get(state.pengguna_aktif, [])

    if len(lagu_saya) == 0:
        bersih()
        print("\nBelum ada musik yang bisa dihapus.")
        print("\nBalik ke halaman sebelumnya...")
        loading()
        return

    print("\nDaftar Musik:")
    no = 1
    for l in lagu_saya:
        print(no, ".", l["judul"], "-", l["artis"])
        no += 1

    try:
        pilih_hapus = input("\nPilih nomor musik yang mau dihapus: ")
    except (KeyboardInterrupt, EOFError):
        bersih()
        print("\nInput dibatalkan.")
        print("\nBalik ke halaman sebelumnya...")
        loading()
        return

    try:
        indeks = int(pilih_hapus) - 1
    except ValueError:
        bersih()
        print("\nInput harus berupa angka!")
```

```

        print("\nBalik ke halaman sebelumnya...")
        loading()
        return

if indeks < 0 or indeks >= len(lagu_saya):
    bersih()
    print("\nNomor tidak valid!")
    print("\nBalik ke halaman sebelumnya...")
    loading()
    return

terpilih = lagu_saya[indeks]

try:
    konfirmasi = input(
        f"Yakin mau hapus '{terpilih['judul']}'? (ya/tidak): "
    ).strip()
except (KeyboardInterrupt, EOFError):
    bersih()
    print("\nInput dibatalkan.")
    print("\nBalik ke halaman sebelumnya...")
    loading()
    return

if konfirmasi.lower() == "ya":
    try:
        del lagu_saya[indeks]
    except Exception:
        bersih()
        print("\nTerjadi kesalahan saat menghapus.")
        print("\nBalik ke halaman sebelumnya...")
        loading()
        return

    bersih()
    print("\nMusik berhasil dihapus!")
    print("\nLoading...")
    loading()
else:
    bersih()
    print("\nPenghapusan dibatalkan.")
    print("\nBalik ke halaman sebelumnya...")
    loading()

```

3.10. main.py

Ini yang di run

```
from menu import menuawal

print("=" * 60)
print("SELAMAT DATANG DI MANAJEMEN MUSIK PERSONALMU LET'S PLAY MUSIC")
print("=" * 60)

if __name__ == "__main__":
    menuawal()
```

4. Hasil Output

```
=====
SELAMAT DATANG DI MANAJEMEN MUSIK PERSONALMU LET'S PLAY MUSIC
=====

--- MENU AWAL ---
1. Login
2. Register
3. Keluar

Pilih menu (1-3):
```

Gambar 4.1 Menu Awal

```
Kamu berhasil masuk, WELCOME BROOO pares

Memuat halaman dalam...
3 ...
█
```

Gambar 4.2 Login

```
Register berhasil! Silakan login.
```

```
Balik ke halaman sebelumnya...
```

```
3 ...
```

```
2 ...
```

```
1 ...
```

```
█
```

Gambar 4.3 Register

```
=====
Terima kasih! See you next time!
=====
```

```
PS D:\praktikum-apd> █
```

Gambar 4.3 Logout dari Menu Awal

```
=====
SEMUA MUSIK
=====
```

```
Musik 1
```

```
-----
Judul   : Bohemian Rhapsody
```

```
Artist : Queen
```

```
Genre  : Rock
```

```
Tekan Enter untuk kembali... █
```

Gambar 4.4 *Read* Semua Musik

```
=====
LIHAT/CARI MUSIK
=====
```

```
--- PILIHAN PENCARIAN ---
```

1. Lihat semua musik
2. Cari berdasarkan judul
3. Cari berdasarkan artist
4. Cari berdasarkan genre

```
Pilih jenis pencarian (1-4): 2
```

```
Masukkan keyword judul: Bohe
```

Gambar 4.5 *Read Musik by Keyword Judul (1)*

```
=====
HASIL PENCARIAN JUDUL
=====
```

```
Musik 1
```

```
-----
Judul   : Bohemian Rhapsody
```

```
Artist  : Queen
```

```
Genre   : Rock
```

```
Tekan Enter untuk kembali...
```

Gambar 4.6 *Read Musik by Keyword Judul (2)*


```
=====
LIHAT/CARI MUSIK
=====
```

```
--- PILIHAN PENCARIAN ---
```

1. Lihat semua musik
2. Cari berdasarkan judul
3. Cari berdasarkan artist
4. Cari berdasarkan genre

Pilih jenis pencarian (1-4): 3

Masukkan keyword artist: Que

Gambar 4.7 *Read Musik by Keyword Artis (1)*

```
=====
HASIL PENCARIAN ARTIST
=====
```

Musik 1

```
-----
Judul   : Bohemian Rhapsody
Artist  : Queen
Genre   : Rock
```

Tekan Enter untuk kembali...

Gambar 4.8 *Read Musik by Keyword Artis (2)*

```
=====
LIHAT/CARI MUSIK
=====
```

```
--- PILIHAN PENCARIAN ---
```

1. Lihat semua musik
2. Cari berdasarkan judul
3. Cari berdasarkan artist
4. Cari berdasarkan genre

Pilih jenis pencarian (1-4): 4

Masukkan keyword genre: ro

Gambar 4.9 *Read Musik by Keyword Genre (1)*

```
=====
HASIL PENCARIAN GENRE
=====

Musik 1
-----
Judul   : Bohemian Rhapsody
Artist  : Queen
Genre   : Rock

Tekan Enter untuk kembali...|
```

Gambar 4.10 *Read Musik by Keyword Genre (2)*

```
=====
TAMBAH MUSIK BARU
=====

Judul Lagu      : Tarot
Nama Artist     : Feast

Pilih Genre:
1. Pop
2. Rock
3. Jazz
4. Hip-Hop
5. Electronic
6. Classical
7. R&B
8. Country

Pilih nomor genre: 1|
```

Gambar 4.11 *Create Musik*

```

=====
EDIT MUSIK
=====

Daftar Musik:
1 . Bohemian Rhapsody - Queen

Pilih nomor musik yang mau diedit: 1

Masukkan data baru (tekan Enter jika tidak ingin mengubah):
Judul Lagu [Bohemian Rhapsody]: Bohe
Nama Artist [Queen]: Fredy

Pilih Genre Baru:
1. Pop
2. Rock
3. Jazz
4. Hip-Hop
5. Electronic
6. Classical
7. R&B
8. Country
Genre sekarang: Rock
Pilih nomor genre (atau Enter untuk tidak ubah): 2

```

Gambar 4.12 *Update Musik*

```

=====
HAPUS MUSIK
=====

Daftar Musik:
1 . Bohe - Fredy

Pilih nomor musik yang mau dihapus: 1
Yakin mau hapus 'Bohe'? (ya/tidak): ya

```

Gambar 4.13 *Delete Musik*

```

4. Hapus musik
5. Logout

Masukkan pilihan (1-5): 5

Yakin mau logout? (ya/tidak): ya

```

Gambar 4.14 *Logout (1)*

```
Logout berhasil!

=====
Terima kasih sudah menggunakan Manajemen Musik Personal!
Sampai jumpa lagi!
=====
PS D:\praktikum-apd> 
```

Gambar 4.15 Logout (2)

5. Langkah-langkah GIT

5.1 GIT Init

```
PS D:\praktikum-apd> git init
Reinitialized existing Git repository in D:/praktikum-apd/.git/
```

Gambar 5.1 Git Init

Saya menjalankan git init, muncul seperti itu karena folder sudah pernah dijadikan repository Git sebelumnya, jadi Git hanya menginisialisasi ulang repo lama, bukan membuat yang baru.

5.2 GIT Add

```
Reinitialized existing Git repository in D:/praktikum-apd/.git/
PS D:\praktikum-apd> git add .
PS D:\praktikum-apd> 
```

Gambar 5.2 Git Add

Saya menjalankan git add, maksudnya saya sedang memasukkan perubahan file ke staging area supaya siap untuk di-commit.

5.3 GIT Commit

```
PS D:\praktikum-apd> git commit -m "pt7"
[main b4cdd86] pt7
 2 files changed, 691 insertions(+)
 create mode 100644 A2_2025/pertemuan7/main.py
 create mode 100644 post-test/post-test-apd-7/2509106045-AthasyahriSyawalFahrezy-PT-7.py
```

Gambar 5.3 Git Commit

Git commit adalah perintah untuk menyimpan perubahan yang sudah masuk staging area ke dalam riwayat repository.

5.4 GIT Remote

```
delete mode 100644 post-test/post-test.apd-2/250100049/KendayaniIsyandaniRezzy/11-2-copy.py
PS D:\praktikum-apd> git remote add origin https://github.com/asyaress/praktikum-apd-25.git
error: remote origin already exists.
```

Gambar 5.4 Git Remote

Saya menjalankan git remote add origin, muncul error karena sebenarnya remote bernama origin sudah ada sebelumnya di repo ini. Jadi Git menolak menambah ulang dengan nama yang sama, karena setiap remote hanya boleh punya satu nama unik.

5.5 GIT Push

```
PS D:\praktikum-apd> git remote set-url origin https://github.com/asyaress/praktikum-apd-25.git
PS D:\praktikum-apd> git push -u origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 4.81 KiB | 2.41 MiB/s, done.
Total 8 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/asyaress/praktikum-apd-25.git
  649511d..b4cdd86  main -> main
branch 'main' set up to track 'origin/main'.
```

Gambar 5.5 Git Push

Saya menjalankan git push -u origin main, hasilnya adalah mengirimkan (upload) commit yang ada di branch main lokal ke GitHub (remote origin).