

LAPORAN PRAKTIKUM
POSTTEST 8
ALGORITMA PEMROGRAMAN DASAR

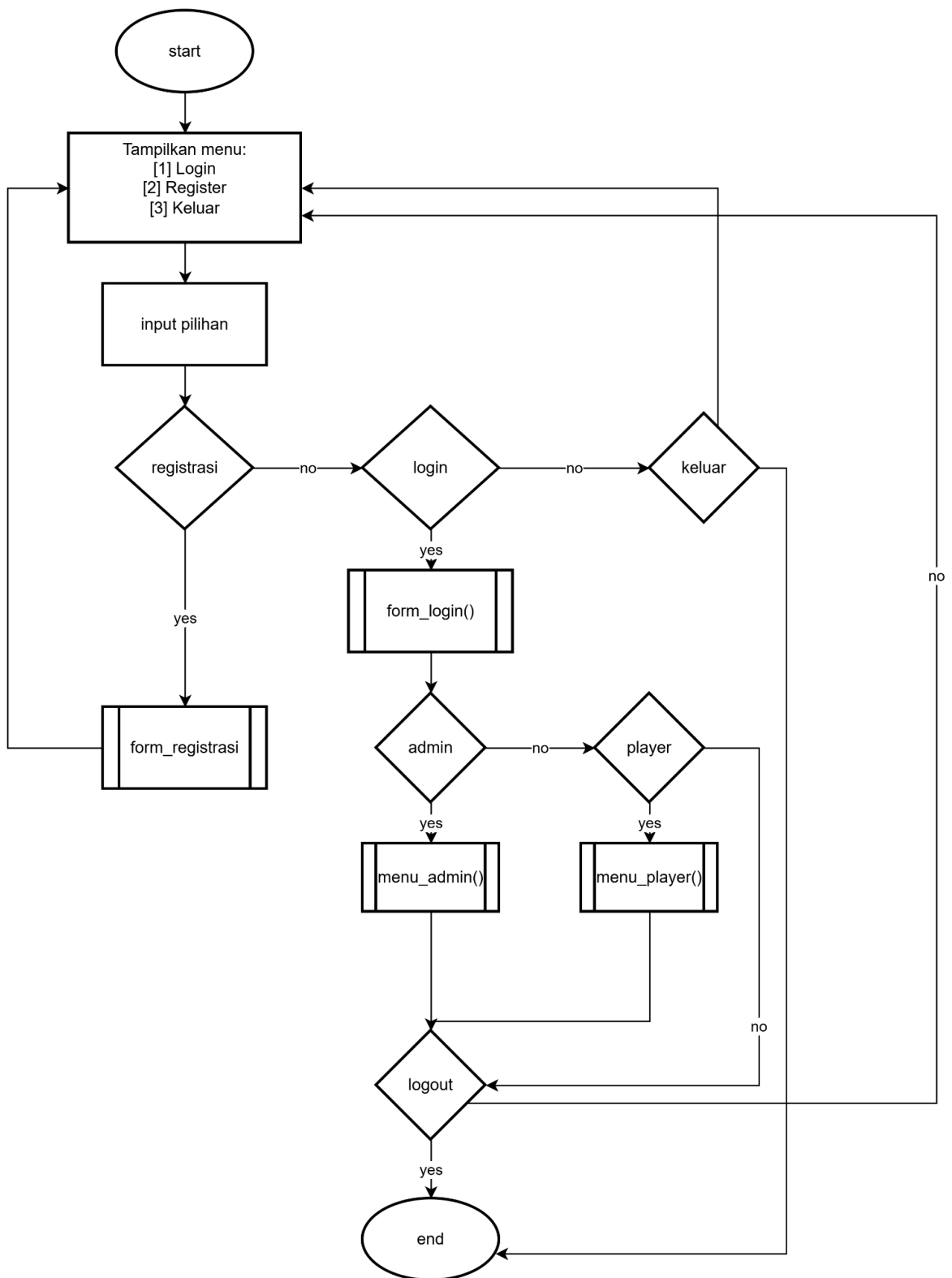


Disusun oleh:
Luvita Khairana Salwa
2509106041
A'25

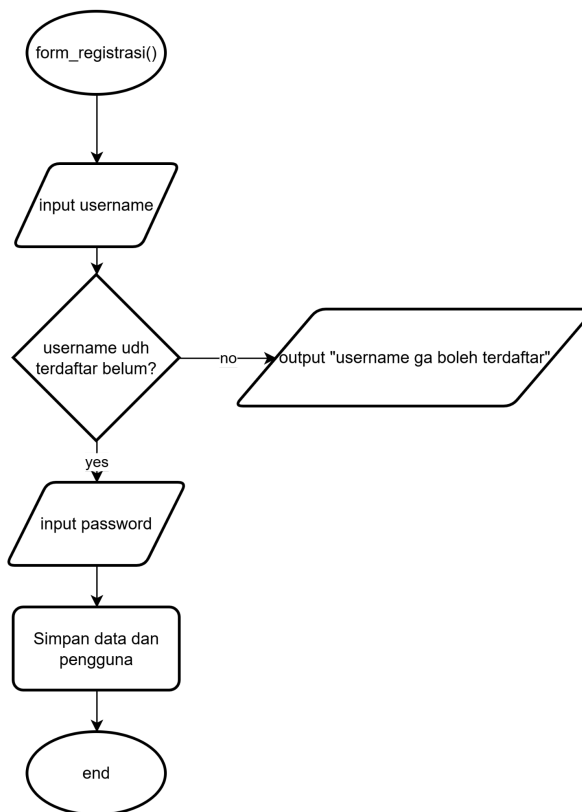
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart

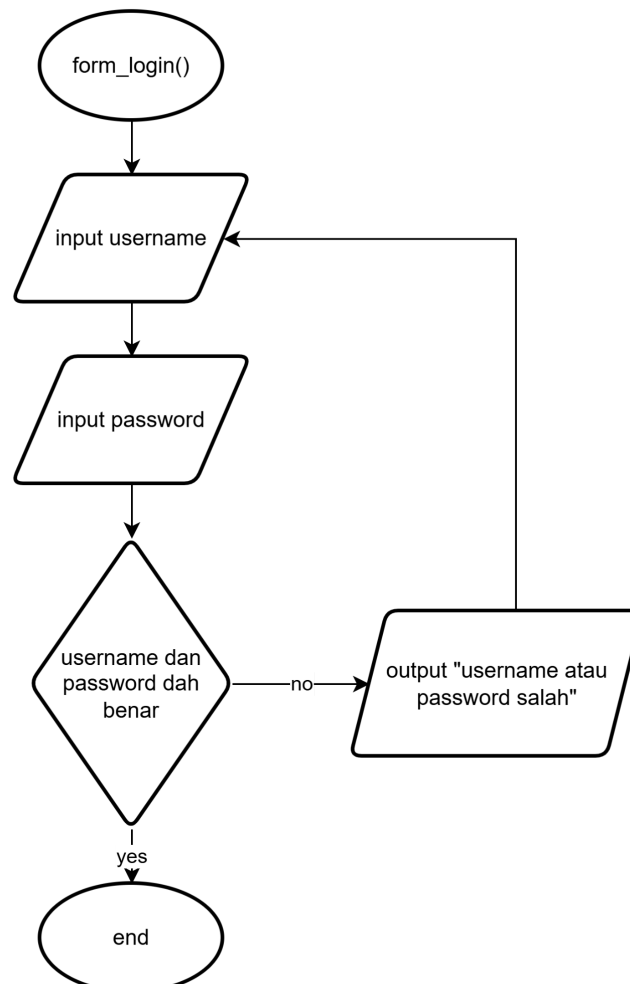
a. Menu utama



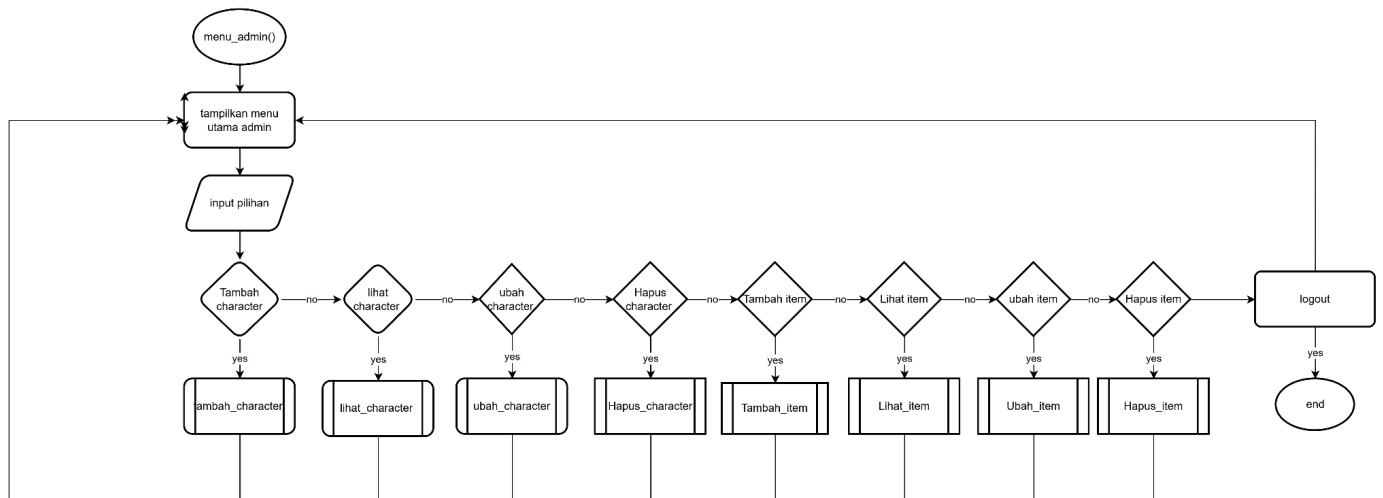
b. Menu registrasi



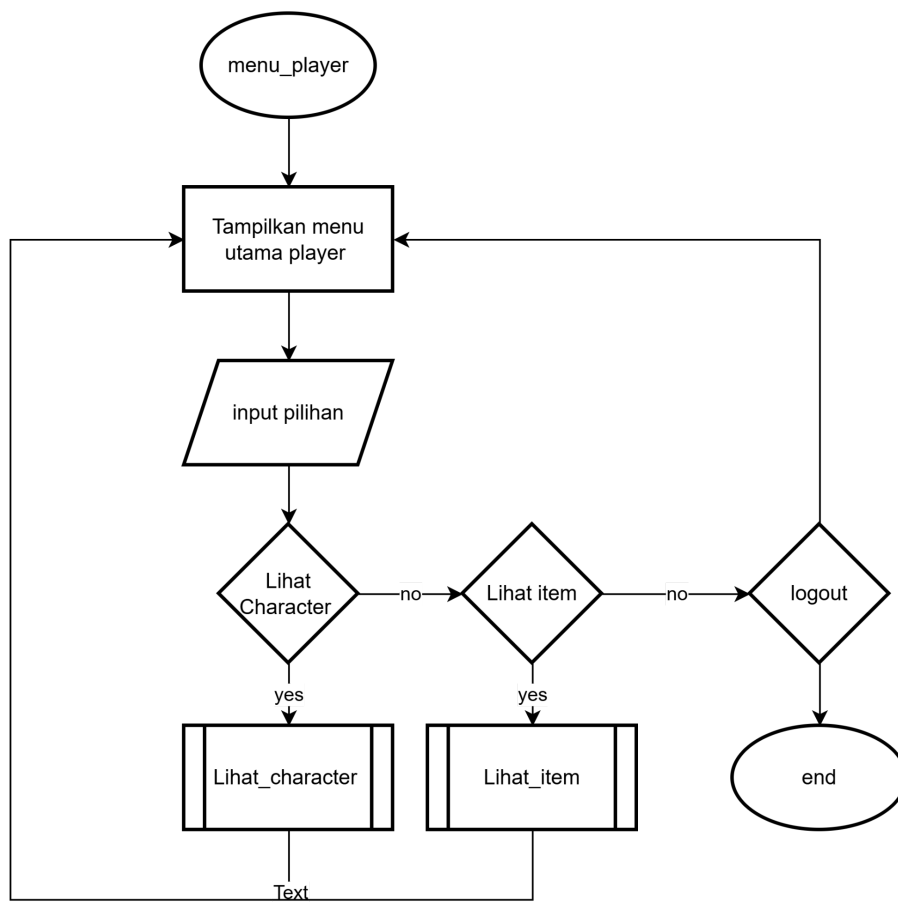
c. menu login



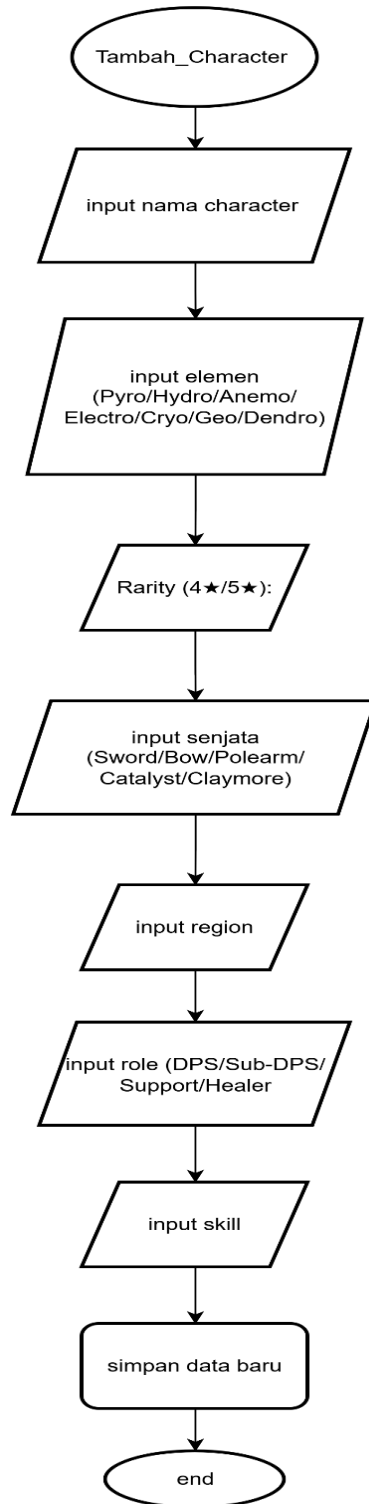
d. Menu admin



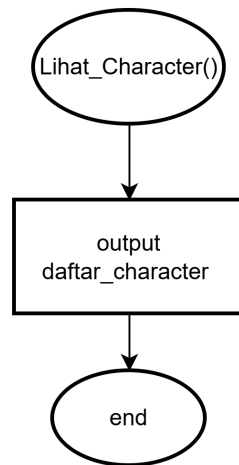
e. Menu Player



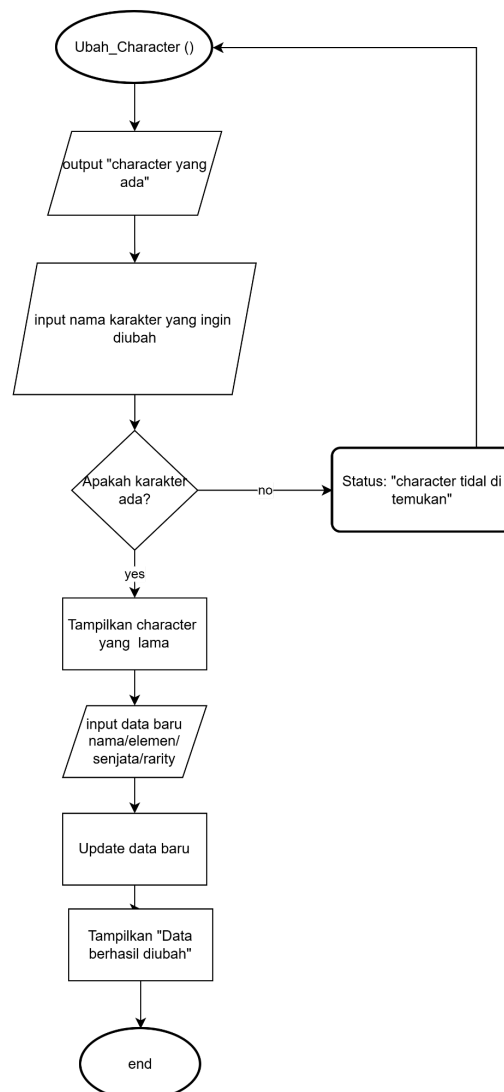
F. Tambah Character



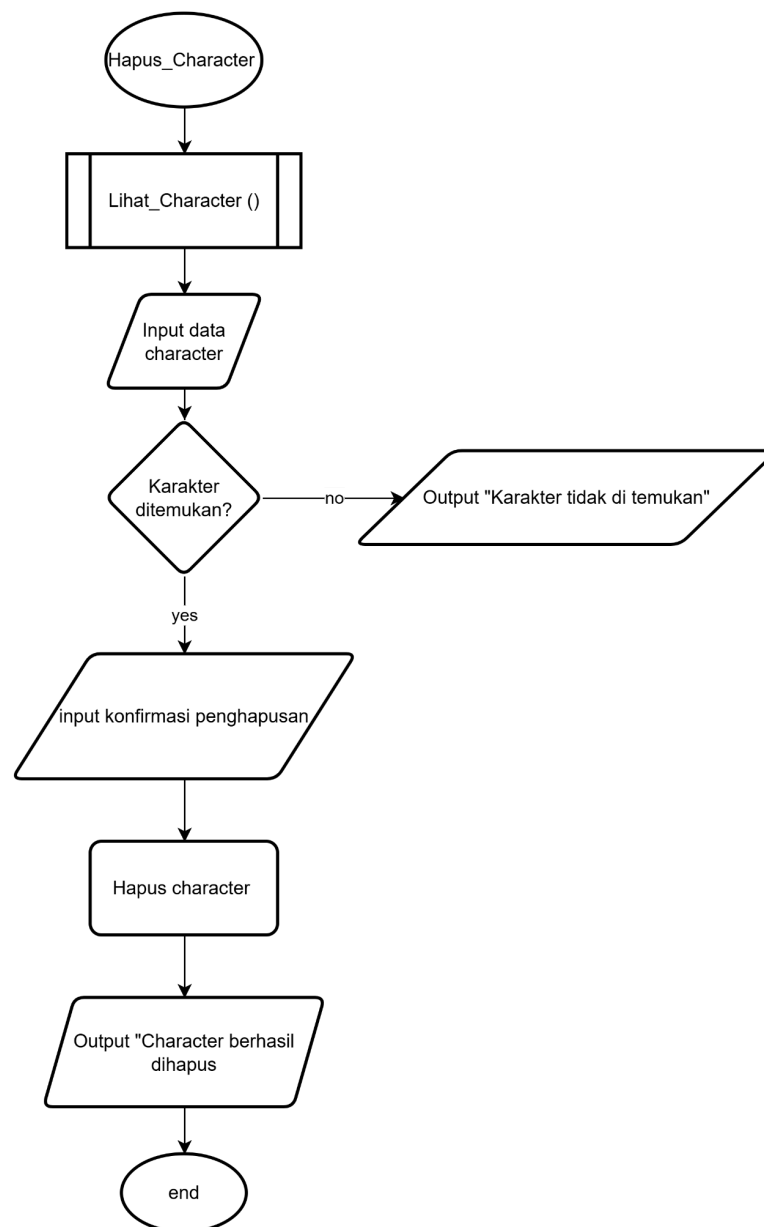
G. Lihat character



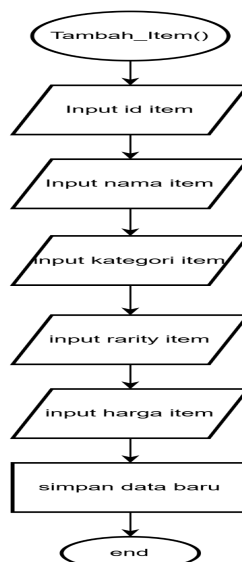
H. Ubah Character



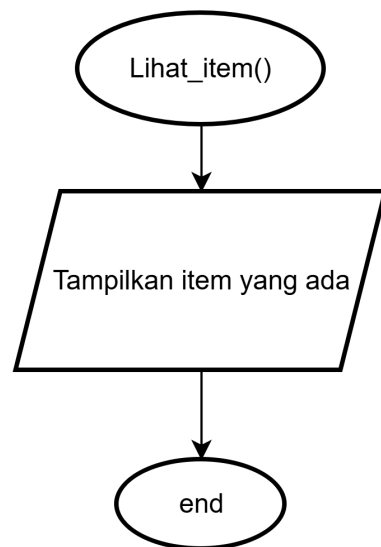
. Hapus character



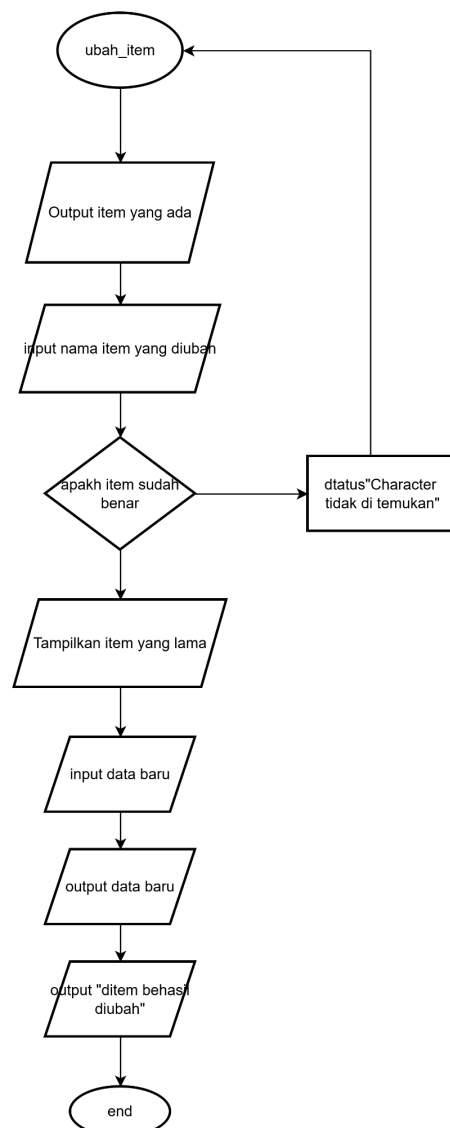
J. Tambah item



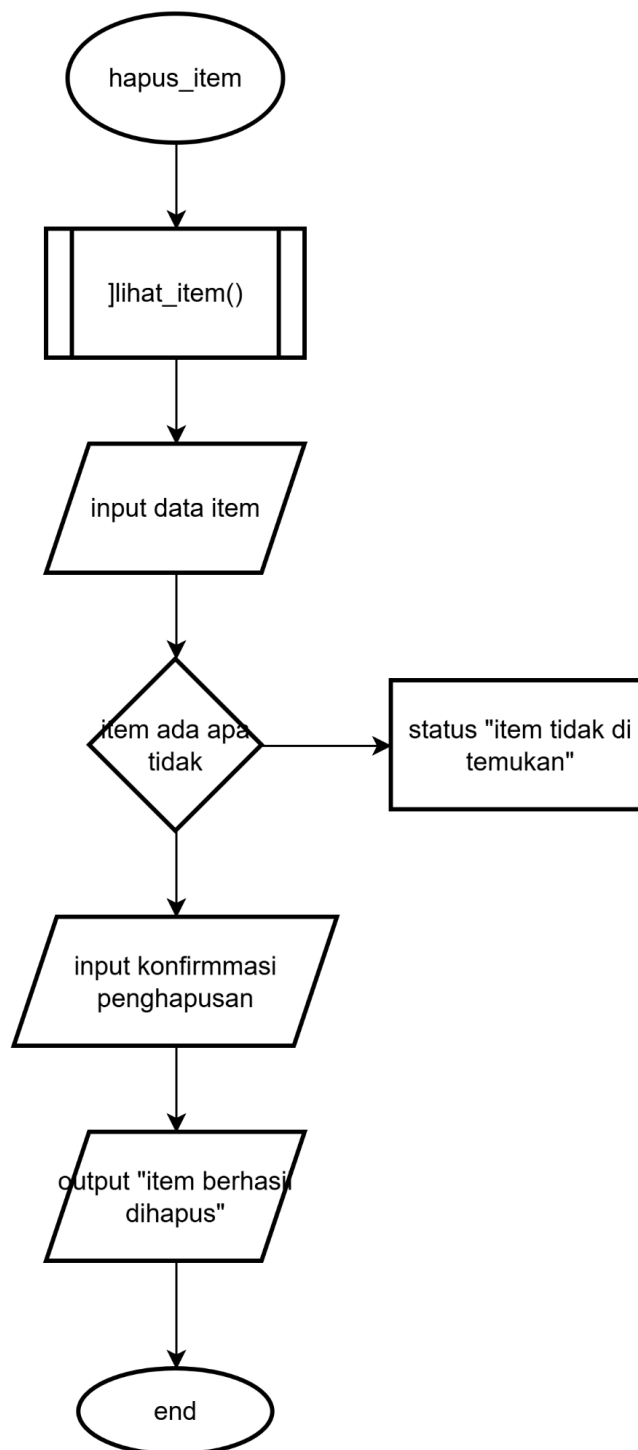
K. Lihat item



L. Ubah item



M. Hapus item



2. Deskripsi Singkat Program

berfungsi untuk:

Program ini merupakan sebuah aplikasi berbasis terminal yang dirancang untuk mengelola data karakter dan item dalam sebuah permainan RPG sederhana.

Program ini mengadopsi sistem seperti pada game Genshin Impact, di mana terdapat dua jenis pengguna dengan peran berbeda, yaitu penguasa (admin) dan player (pemain biasa). Program ini juga dilengkapi dengan sistem login dan register akun, sehingga setiap pengguna harus memiliki akun terlebih dahulu untuk dapat mengakses fitur-fitur yang tersedia.

Tujuan utama dari program ini adalah untuk memberikan gambaran bagaimana sistem manajemen data dan kontrol akses bekerja dalam sebuah aplikasi sederhana. Dengan struktur menu yang jelas, pengguna dapat memilih berbagai fitur sesuai dengan perannya. Program ini dibuat menggunakan bahasa pemrograman Python dengan memanfaatkan struktur perulangan, percabangan, function sebagai dasar pengelolaan data.

3. Source Code

- Data.py

```
users = [  
    {"username": "akito yamada", "password": "akane", "role":  
    "penguasa"}  
]  
  
characters = []  
items = []  
  
login_login = 6
```

- Main.py

```
import os  
from inquirer import list_input  
from data import users, login_login  
from menuAdmin import menu_admin  
from menuPlayer import menu_player  
  
def show_title(teks):  
    print("=" * 60)  
    print(f"|| {teks.center(56)} ||")  
    print("=" * 60)  
  
def login(attempt=1):  
    if attempt > login_login:
```

```

        print("Terlalu banyak gagal login. Program keluar.")
        exit()
    uname = input("Username: ")
    pw = input("Password: ")
    for u in users:
        if u["username"] == uname and u["password"] == pw:
            print(f"Login sukses! Selamat datang {uname.title()}")
            return u
    print("Login salah! Coba lagi.")
    return login(attempt + 1)

while True:
    os.system("cls" if os.name == "nt" else "clear")
    show_title("SISTEM GENSHIN IMPACT")

    pilih = list_input("Pilih menu", choices=["Login", "Register",
"Keluar"])
    if pilih == "Login":
        user_login = login()
        if user_login["role"] == "penguasa":
            menu_admin()
        else:
            menu_player()

    elif pilih == "Register":
        uname = input("Username baru: ")
        if any(u["username"] == uname for u in users):
            print("Username sudah digunakan!")
        else:
            pw = input("Password: ")
            users.append({"username": uname, "password": pw, "role":
"player"})
            print("Akun berhasil dibuat!")
            input("Tekan Enter...")

    elif pilih == "Keluar":
        print("Program selesai! Terima kasih bermain.")
        break

```

- MenuAdmin.py

```

import os
from prettytable import PrettyTable
from data import characters, items
from inquirer import list_input

```

```

def show_title(teks):
    print("=" * 60)
    print(f"|| {teks.center(56)} ||")
    print("=" * 60)

def tampilkan_characters():
    if not characters:
        print("Belum ada character.")
        return
    table = PrettyTable()
    table.field_names = ["Nama", "Elemen", "Rarity", "Senjata", "Region",
"Role", "Skill"]
    for c in characters:
        table.add_row([c['nama'], c['elemen'], c['rarity'], c['senjata'],
c['region'], c['role'], c['skill']])
    print(table)

def tampilkan_items():
    if not items:
        print("Belum ada item.")
        return
    table = PrettyTable()
    table.field_names = ["ID", "Nama", "Kategori", "Rarity", "Efek",
"Harga"]
    for i in items:
        table.add_row([i['id'], i['nama'], i['kategori'], i['rarity'],
i['efek'], i['harga']])
    print(table)

def cari_character(nama):
    for c in characters:
        if c["nama"].lower() == nama.lower():
            return c
    return None

def cari_item(id_item):
    for i in items:
        if i["id"].lower() == id_item.lower():
            return i
    return None

def menu_admin():
    while True:
        os.system("cls" if os.name == "nt" else "clear")
        show_title("MENU PENGUASA GAME")

        pilih = list_input("Pilih menu", choices=[
            "Tambah Character",

```

```

        "Lihat Character",
        "Ubah Character",
        "Hapus Character",
        "Tambah Item",
        "Lihat Item",
        "Ubah Item",
        "Hapus Item",
        "Logout"
    ])

    if pilih == "Tambah Character":
        show_title("TAMBAH CHARACTER")
        nama = input("Nama: ")
        elemen = input("Elemen: ")
        rarity = input("Rarity: ")
        senjata = input("Senjata: ")
        region = input("Region: ")
        role = input("Role: ")
        skill = input("Ultimate Skill: ")
        characters.append({
            "nama": nama, "elemen": elemen, "rarity": rarity,
            "senjata": senjata, "region": region, "role": role, "skill":
skill
        })
        print("Character berhasil ditambah.")

    elif pilih == "Lihat Character":
        show_title("DAFTAR CHARACTER")
        tampilkan_characters()

    elif pilih == "Ubah Character":
        nama = input("Nama character: ")
        c = cari_character(nama)
        if c:
            c["senjata"] = input("Senjata baru: ")
            print("Berhasil diubah.")
        else:
            print("Character tidak ditemukan.")

    elif pilih == "Hapus Character":
        nama = input("Nama character: ")
        c = cari_character(nama)
        if c:
            characters.remove(c)
            print("Character dihapus.")
        else:
            print("Tidak ditemukan.")

```

```

elif pilih == "Tambah Item":
    id_item = input("ID Item: ")
    nama_item = input("Nama Item: ")
    kategori = input("Kategori: ")
    rarity = input("Rarity: ")
    efek = input("Efek: ")
    harga = input("Harga: ")
    items.append({
        "id": id_item, "nama": nama_item, "kategori": kategori,
        "rarity": rarity, "efek": efek, "harga": harga
    })
    print("Item berhasil ditambah.")

elif pilih == "Lihat Item":
    show_title("DAFTAR ITEM")
    tampilkan_items()

elif pilih == "Ubah Item":
    id_item = input("ID Item: ")
    i = cari_item(id_item)
    if i:
        i["efek"] = input("Efek baru: ")
        print("Berhasil diubah.")
    else:
        print("Item tidak ditemukan.")

elif pilih == "Hapus Item":
    id_item = input("ID Item: ")
    i = cari_item(id_item)
    if i:
        items.remove(i)
        print("Item dihapus.")
    else:
        print("Tidak ditemukan.")

elif pilih == "Logout":
    break

input("Tekan Enter...")

```

- [menuPlayer.py](#)

```

import os
from prettytable import PrettyTable
from data import characters, items
from inquirer import list_input

```

```

def show_title(teks):
    print("=" * 60)
    print(f"|| {teks.center(56)} ||")
    print("=" * 60)

def tampilkan_characters():
    if not characters:
        print("Belum ada character.")
        return
    table = PrettyTable()
    table.field_names = ["Nama", "Elemen", "Rarity", "Senjata"]
    for c in characters:
        table.add_row([c['nama'], c['elemen'], c['rarity'], c['senjata']])
    print(table)

def tampilkan_items():
    if not items:
        print("Belum ada item.")
        return
    table = PrettyTable()
    table.field_names = ["ID", "Nama", "Kategori", "Harga"]
    for i in items:
        table.add_row([i['id'], i['nama'], i['kategori'], i['harga']])
    print(table)

def menu_player():
    while True:
        os.system("cls" if os.name == "nt" else "clear")
        show_title("MENU PLAYER")

        pilih = list_input("Pilih menu", choices=[
            "Lihat Character",
            "Lihat Item",
            "Logout"
        ])

        if pilih == "Lihat Character":
            show_title("CHARACTER TERSEDIA")
            tampilkan_characters()
        elif pilih == "Lihat Item":
            show_title("ITEM TERSEDIA")
            tampilkan_items()
        elif pilih == "Logout":
            break

        input("Tekan Enter...")

```

4. HASIL OUTPUT

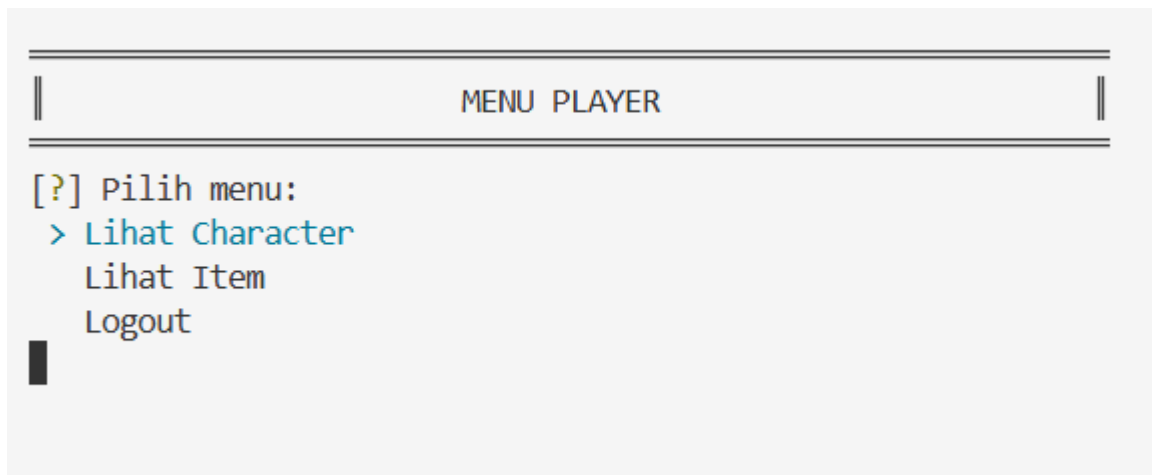
- Menu utama

```
||
||          SISTEM GENSIN IMPACT          ||
||
[?] Pilih menu:
> Login
  Register
  Keluar
█
```

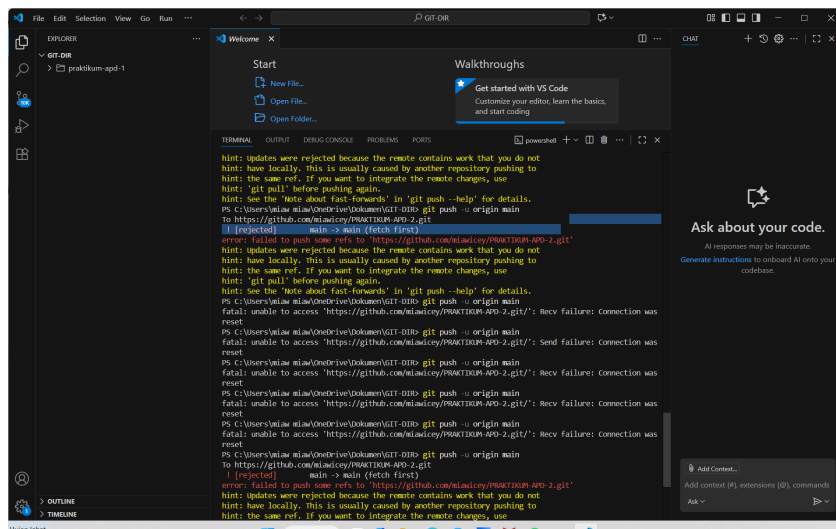
- menu admin

```
||
||          MENU PENGUASA GAME          ||
||
[?] Pilih menu:
> Tambah Character
  Lihat Character
  Ubah Character
  Hapus Character
  Tambah Item
  Lihat Item
  Ubah Item
  Hapus Item
  Logout
█
```


- Menu player



5. Langkah-langkah GIT



5.1 GIT Init

git init → Perintah untuk membuat repository Git baru di folder lokal.

- Fungsinya: Menginisialisasi folder agar bisa digunakan Git (ada folder **.git** di dalamnya). Contoh: “git init”

5.2 GIT Add

git add → Perintah untuk menambahkan file atau perubahan file ke staging area.

- Fungsinya: Menandai file mana yang siap untuk disimpan (commit).
Contoh: “git add index.html”(menambahkan satu file) “git add .” (menambahkan semua file)

5.3 GIT Commit

`git commit` → Perintah untuk menyimpan perubahan dari staging area ke repository lokal.

- Fungsinya: Mencatat snapshot dari kode dengan pesan deskriptif.
Contoh: “`git commit -m`” (Menambahkan halaman index)

5.4 GIT Remote

`git remote` → Perintah untuk menghubungkan repository lokal dengan repository di server (misalnya GitHub/GitLab).

- Fungsinya: Agar repository lokal bisa disinkronkan dengan repository online.
Contoh: “`git remote add origin`” <https://github.com/user/nama-repo.git>

5.5 GIT Push

`git push` → Perintah untuk mengirim commit dari repository lokal ke repository remote (GitHub, GitLab, dll).

- Fungsinya: Mengupload kode yang sudah di-commit ke server.
Contoh: “`git push -u origin main`”