LAPORAN PRAKTIKUM

POSTTEST 7

ALGORITMA PEMROGRAMAN DASAR

Heiza Rizki Pratama

2509106019

A’25

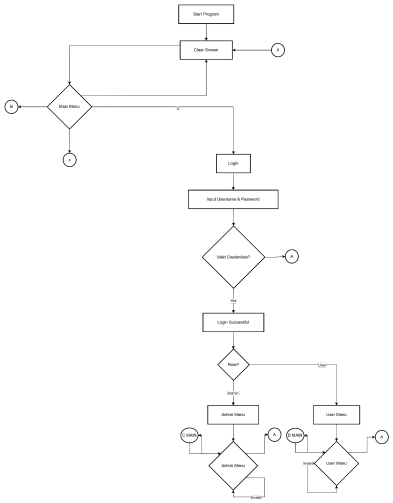
PROGRAM STUDI INFORMATIKA

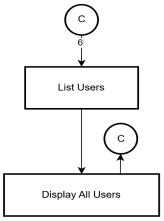
UNIVERSITAS MULAWARMAN

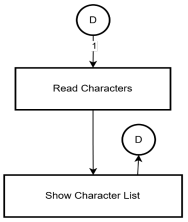
SAMARINDA

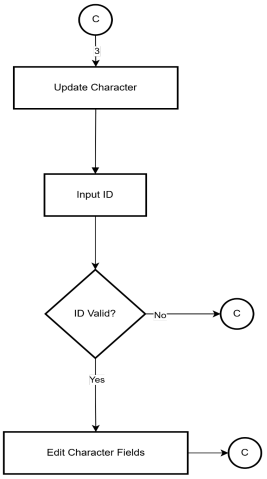
2025

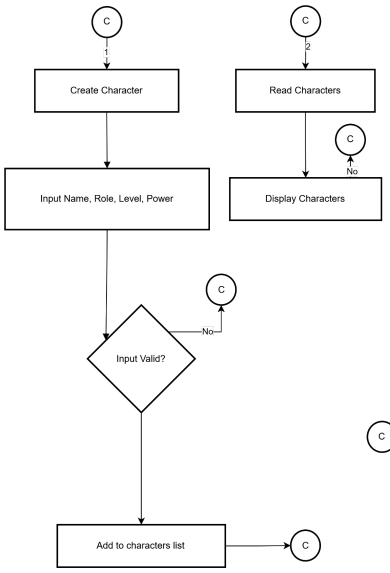
1. Flowchart


Start Program
Clear Screen
Main Menu
Login
Input Username & Password
Valid Credentials?
Yes
Login Successful
Role?
User
Admin Menu
User Menu
Admin Menu
User Menu
Invalid

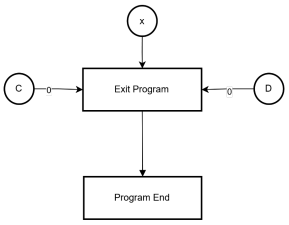

C
6
List Users
C
Display All Users

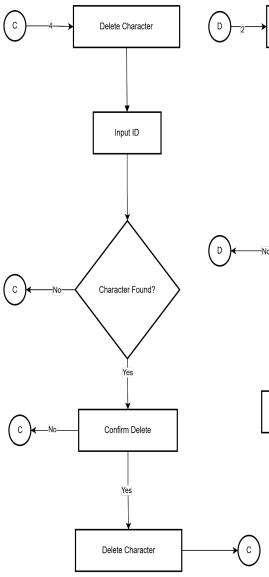

D
1
Read Characters
D
Show Character List

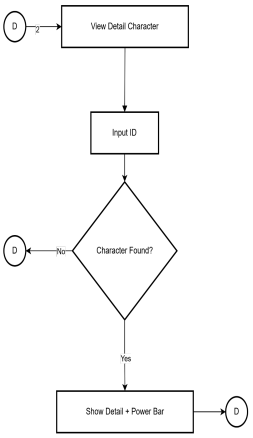

C
3
Update Character
Input ID
ID Valid?
No
C
Yes
Edit Character Fields
C

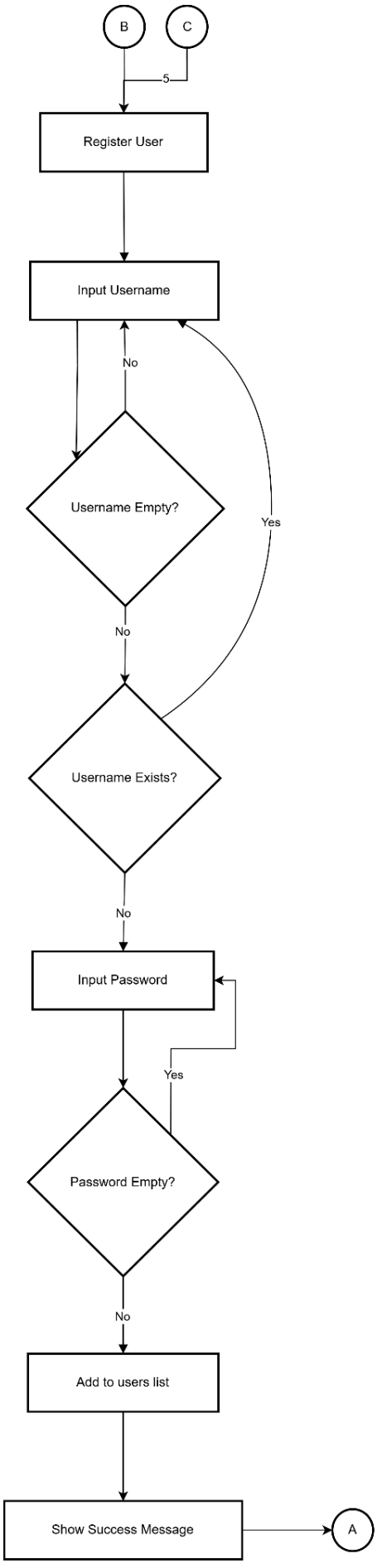

C
1
Create Character
Input Name, Role, Level, Power
Input Valid?
C
No
Add to characters list
C

C
2
Read Characters
C
No
Display Characters


x
C
0
Exit Program
0
D
Program End


C
4
Delete Character
Input ID
C
No
Character Found?
Yes
C
No
Confirm Delete
Yes
Delete Character
C


D
2
View Detail Character
Input ID
D
No
Character Found?
Yes
Show Detail + Power Bar
D


B
C
5
Register User
Input Username
No
Username Empty?
Yes
No
Username Exists?
No
Input Password
Yes
Password Empty?
No
Add to users list
Show Success Message
A

2. Deskripsi Singkat Program

Program ini adalah sistem manajemen sederhana berbasis terminal yang digunakan untuk: Mendaftarkan dan mengelola akun pengguna (admin & user), Mengelola data karakter dalam game (CRUD), Memberikan visualisasi kekuatan karakter. Menjalankan operasi dengan kontrol akses (admin memiliki hak penuh, user hanya bisa melihat data).

3. Source Code

FILE 1

```
import os
import sys

# globar variabel
users = {
    "admin": {"password": "admin123", "role": "admin"}
}
characters = {
    1: {"name": "Aegis", "role": "Tank", "level": 5,
"power": 80},
    2: {"name": "Lumina", "role": "Mage", "level": 3,
"power": 45}
}
next_char_id = max(characters.keys(), default=0) + 1

# function without parameter
def clear_screen():
    if os.name == "nt":
        os.system("cls")
    else:
        os.system("clear")
```

```
def pause():
    input("\nTekan Enter untuk melanjutkan...")

# function with parameter
def add_user(username, password, role="user"):
    global users
    if username in users:
        return False
    users[username] = {"password": password, "role": role}
    return True


def add_character(name, role_c, level, power):
    global characters, next_char_id
    cid = next_char_id
    characters[cid] = {"name": name, "role": role_c,
"level": level, "power": power}
    next_char_id += 1
    return cid

# procedure 2
def procedure_register_user():
    clear_screen()
    print("== REGISTER USER ==")
    while True:
        try:
            username = input("Masukkan username baru:
").strip()
            if username == "":
                print("Username tidak boleh kosong.")
                continue
            if username in users:
                print("Username sudah digunakan.")
```

```
                continue
            password = input("Masukkan password: ").strip()
            if password == "":
                print("Password tidak boleh kosong.")
                continue
            add_user(username, password)
            print(f"User '{username}' berhasil terdaftar!")
            pause()
            break
        except Exception as e:
            print("Terjadi kesalahan:", e)
            pause()
            break


def procedure_create_character():
    clear_screen()
    print("== CREATE CHARACTER ==")
    try:
        name = input("Nama karakter: ").strip()
        if name == "":
            print("Nama tidak boleh kosong.")
            pause()
            return
        role_c = input("Peran (role): ").strip()
        lvl = input("Level (>=1): ").strip()
        if not lvl.isdigit() or int(lvl) < 1:
            print("Level tidak valid.")
            pause()
            return
        power = input("Power (0-100): ").strip()
        if not power.isdigit() or not (0 <= int(power) <=
100):
```

```
            print("Power tidak valid.")
            pause()
            return
        cid = add_character(name, role_c, int(lvl),
int(power))
        print(f"Karakter '{name}' berhasil ditambahkan
dengan ID {cid}.")
        pause()
    except Exception as e:
        print("Kesalahan input:", e)
        pause()

# recursif function
def tampil_karakter_rekursif(id_list, idx=0):
    if idx >= len(id_list):
        return
    cid = id_list[idx]
    c = characters[cid]
    bar = "[" + "" * int((c["power"]/100)*20) + "-" *
(20-int((c["power"]/100)*20)) + "]"
    print(f"{cid:<4} | {c['name']:<12} | {c['role']:<10} |
{c['level']:<3} | {c['power']:<3} {bar}")
    tampil_karakter_rekursif(id_list, idx + 1)

# parameter function
def boost_power(cid, percent):
    if cid not in characters:
        return False
    c = characters[cid]
    c["power"] += int(c["power"] * (percent / 100))
    if c["power"] > 100:
        c["power"] = 100
    return True
```

```
# atmin
def admin_menu(username):
    while True:
        try:
            clear_screen()
            print(f"== MENU ADMIN ({username}) ==")
            print("1. Tambah Karakter")
            print("2. Lihat Semua Karakter")
            print("3. Update Karakter")
            print("4. Hapus Karakter")
            print("5. Register User Baru")
            print("6. Lihat Semua User")
            print("7. Logout")
            print("0. Keluar")
            pilih = input("Pilih menu: ").strip()

            if pilih == "1":
                procedure_create_character()
            elif pilih == "2":
                clear_screen()
                print("== DAFTAR KARAKTER ==")
                if not characters:
                    print("Tidak ada data karakter.")
                else:
                    print(f"{'ID':<4} | {'Nama':<12} |
{'Role':<10} | {'Lvl':<3} | {'Power'}")
                    print("-" * 60)

tampil_karakter_rekursif(sorted(characters.keys()))
                pause()
            elif pilih == "3":
```

```
                cid = input("Masukkan ID karakter:
").strip()
                if not cid.isdigit() or int(cid) not in
characters:
                    print("ID tidak valid.")
                    pause()
                    continue
                cid = int(cid)
                c = characters[cid]
                new_name = input(f"Nama baru [{c['name']}]:
").strip()
                new_role = input(f"Role baru [{c['role']}]:
").strip()
                new_level = input(f"Level baru
[{c['level']}]: ").strip()
                new_power = input(f"Power baru
[{c['power']}]: ").strip()
                if new_name:
                    c["name"] = new_name
                if new_role:
                    c["role"] = new_role
                if new_level.isdigit():
                    c["level"] = int(new_level)
                if new_power.isdigit() and 0 <=
int(new_power) <= 100:
                    c["power"] = int(new_power)
                print("Data karakter diperbarui!")
                pause()
            elif pilih == "4":
                cid = input("Masukkan ID karakter:
").strip()
                if not cid.isdigit() or int(cid) not in
characters:
```

```
                    print("ID tidak valid.")
                    pause()
                    continue
                cid = int(cid)
                konfirmasi = input(f"Yakin ingin hapus
{characters[cid]['name']}? (y/n): ").lower()
                if konfirmasi == "y":
                    del characters[cid]
                    print("Karakter dihapus.")
                else:
                    print("Dibatalkan.")
                pause()
            elif pilih == "5":
                procedure_register_user()
            elif pilih == "6":
                clear_screen()
                print("== DAFTAR USER ==")
                for u, info in users.items():
                    print(f"- {u} ({info['role']})")
                pause()
            elif pilih == "7":
                break
            elif pilih == "0":
                print("Sampai jumpa!")
                sys.exit(0)
            else:
                print("Pilihan tidak valid.")
                pause()
        except Exception as e:
            print("Terjadi kesalahan:", e)
            pause()

# new usr
```

```
def user_menu(username):
    while True:
        try:
            clear_screen()
            print(f"== MENU USER ({username}) ==")
            print("1. Lihat Semua Karakter")
            print("2. Detail Karakter")
            print("3. Logout")
            print("0. Keluar")
            pilih = input("Pilih menu: ").strip()

            if pilih == "1":
                clear_screen()
                print("== DAFTAR KARAKTER ==")
                if not characters:
                    print("Tidak ada karakter.")
                else:
                    print(f"{'ID':<4} | {'Nama':<12} |
{'Role':<10} | {'Lvl':<3} | {'Power'}")
                    print("-" * 60)

tampil_karakter_rekursif(sorted(characters.keys()))
                pause()
            elif pilih == "2":
                cid = input("Masukkan ID karakter:
").strip()
                if not cid.isdigit() or int(cid) not in
characters:
                    print("ID tidak valid.")
                    pause()
                    continue
                cid = int(cid)
                c = characters[cid]
```

```
                print(f"\nID: {cid}")
                print(f"Nama : {c['name']}")
                print(f"Role : {c['role']}")
                print(f"Level: {c['level']}")
                print(f"Power: {c['power']}")
                bar = "[" + "" * int((c["power"]/100)*30) +
"-" * (30-int((c["power"]/100)*30)) + "]"
                print("Visual:", bar)
                pause()
            elif pilih == "3":
                break
            elif pilih == "0":
                print("Sampai jumpa!")
                sys.exit(0)
            else:
                print("Pilihan tidak valid.")
                pause()
        except Exception as e:
            print("Terjadi kesalahan:", e)
            pause()

# main prog
def main():
    while True:
        try:
            clear_screen()
            print("=== SISTEM MANAGEMENT KARAKTER GAME ===")
            print("1. Register")
            print("2. Login")
            print("3. Exit")
            menu = input("Pilih menu: ").strip()

            if menu == "1":
```

```
                procedure_register_user()
            elif menu == "2":
                clear_screen()
                print("== LOGIN ==")
                username = input("Username: ").strip()
                password = input("Password: ").strip()
                if username in users and
users[username]["password"] == password:
                    role = users[username]["role"]
                    print(f"Selamat datang, {username}
({role})!")
                    pause()
                    if role == "admin":
                        admin_menu(username)
                    else:
                        user_menu(username)
                else:
                    print("Username atau password salah.")
                    pause()
            elif menu == "3":
                print("Terima kasih telah menggunakan
program ini!")
                sys.exit(0)
            else:
                print("Pilihan tidak valid.")
                pause()
        except KeyboardInterrupt:
            print("\nProgram dihentikan oleh user.")
            sys.exit(0)
        except Exception as e:
            print("Terjadi kesalahan:", e)
            pause()
```

```python
if __name__ == "__main__":
    main()
```

4. Hasil Output

a. MENU UTAMA

```
=== MANAGEMENT DATA CHARACTER GAME ===
1. Register
2. Login
3. Exit
Choose option:
```

b. MENU REGISTER USER

```
== REGISTER USER ==
Enter new username:
```

c. MENU USER

```
== USER MENU (Lucius) ==
1. Read Characters
2. View Detail
3. Logout
0. Exit
Choose:
```

d. MENU ADMIN

```
== ADMIN MENU (admin) ==
1. Create Character
2. Read Characters
3. Update Character
4. Delete Character
5. Register User
6. List Users
7. Logout
0. Exit
Choose:
```

5. Langkah-langkah GIT

```
PS C:\Users\Lucius\Documents\New folder> git init
Reinitialized existing Git repository in C:/Users/Lucius/Documents/New folder/.git/
PS C:\Users\Lucius\Documents\New folder> git add .
PS C:\Users\Lucius\Documents\New folder> git commit -m "ini commit"
[main 94354ce] ini commit
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\Lucius\Documents\New folder> git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 465 bytes | 155.00 KiB/s, done.
Total 6 (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/01001100-KMS/praktikum-apd.git
   23547ec..94354ce  main -> main
```

5.1 GIT Add

```
PS C:\Users\Lucius\Documents\New folder> git add .
```

Fungsinya**:** Memilih file yang sudah diubah untuk masuk ke staging **area** (daftar siap commit). Tanpa `git add`, perubahan tidak akan ikut tersimpan saat `git commit`.

Contoh:
git add index.html
→ hanya file index.html yang siap di-commit.
git add .
→ semua file yang berubah akan masuk staging area.

5.2 GIT Commit

```
PS C:\Users\Lucius\Documents\New folder> git commit -m "ini commit"
[main 94354ce] ini commit
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Fungsinya: Menyimpan perubahan yang sudah dipilih (staging area*) ke dalam riwayat repository Commit ini ibarat checkpoint atau simpan versi dari proyek. Biasanya commit disertai pesan (`-m`) agar jelas maksud perubahannya. Contoh:*

`commit - git m "Menambahkan fitur login"`

5.3 GIT Push

```
PS C:\Users\Lucius\Documents\New folder> git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 465 bytes | 155.00 KiB/s, done.
Total 6 (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/01001100-KMS/praktikum-apd.git
   23547ec..94354ce  main -> main
```

Fungsinya: Mengirim perubahan (commit) yang ada di repository lokal ke repository remote (misalnya GitHub, GitLab, Bitbucket). Supaya bisa `push`, biasanya harus sudah `git remote add origin <url>` terlebih dahulu.

Contoh:

```
git push origin main
```

→ Mengirim commit lokal ke branch `main` di repository remote bernama `origin`.