LAPORAN PRAKTIKUM

POSTTEST 5

ALGORITMA PEMROGRAMAN DASAR

Heiza Rizki Pratama

2509106019

A’25

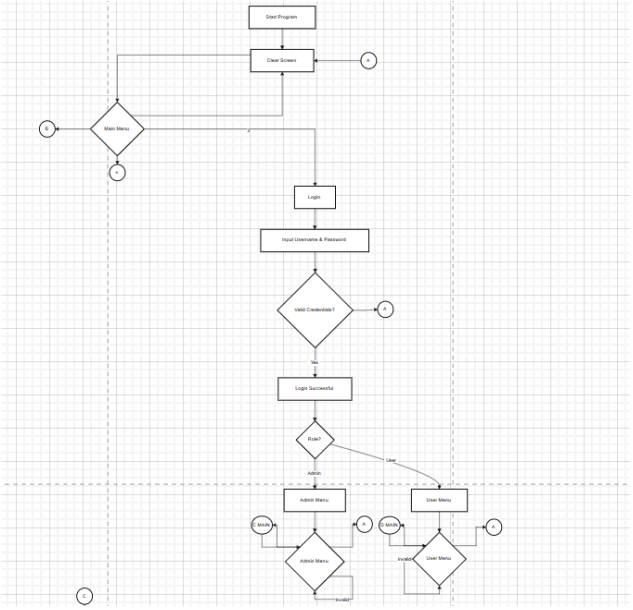
PROGRAM STUDI INFORMATIKA

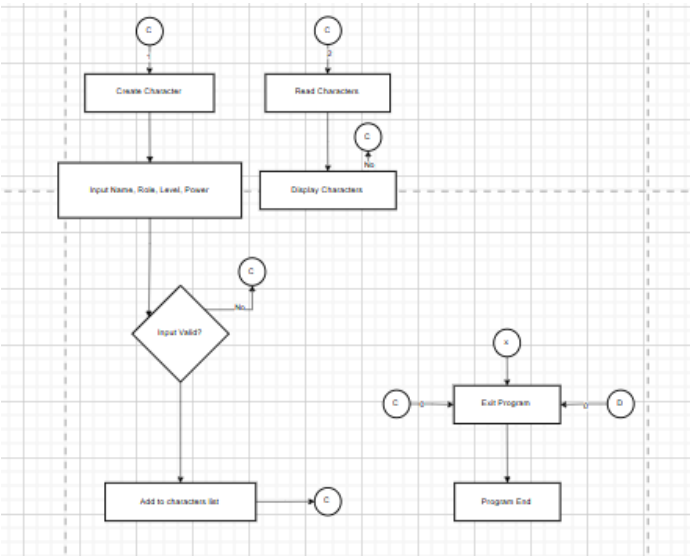
UNIVERSITAS MULAWARMAN

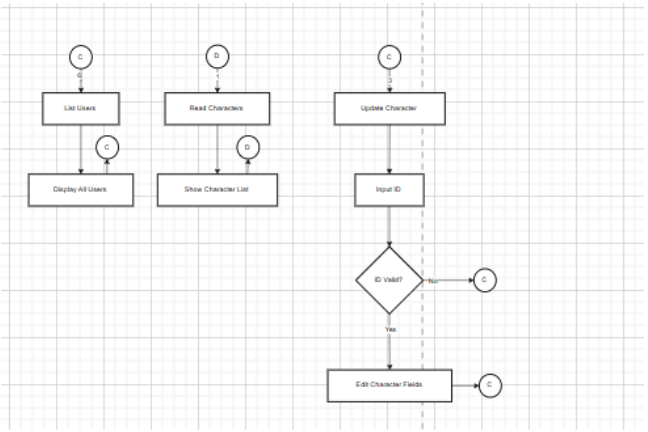
SAMARINDA

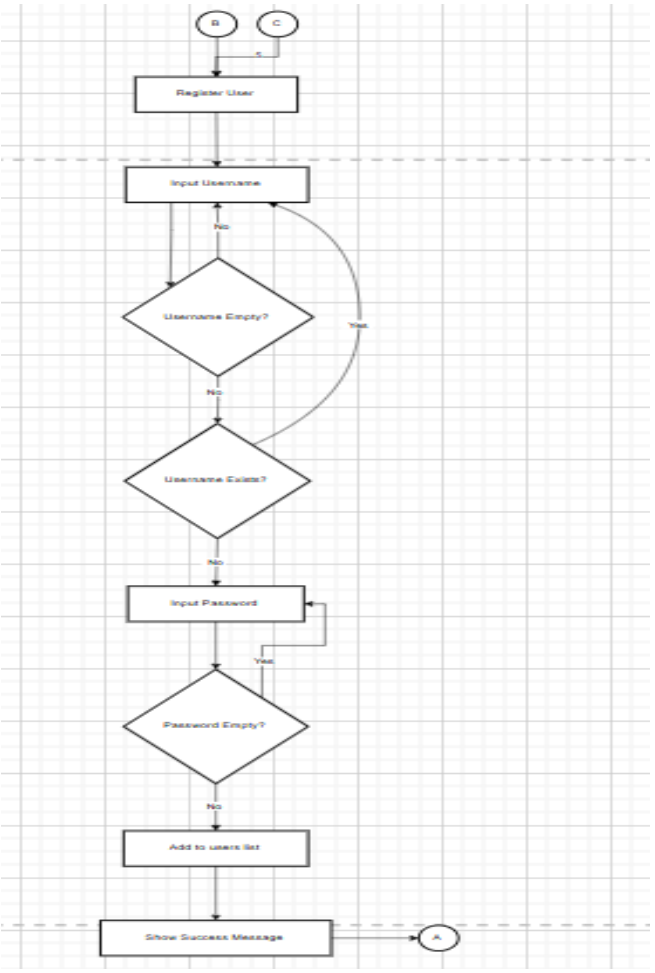
2025

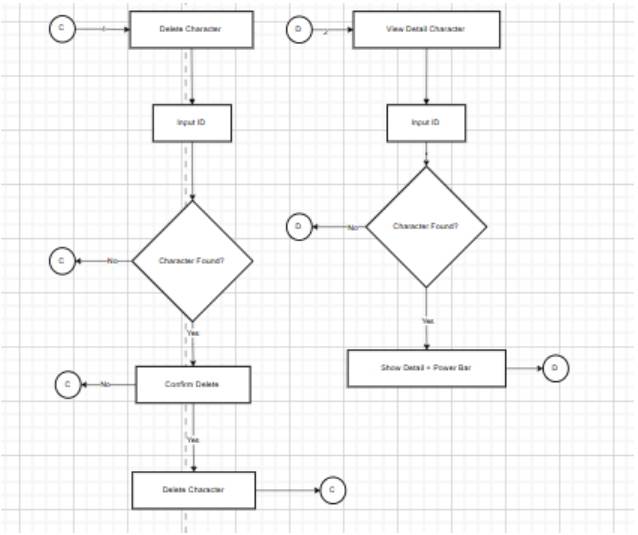
1. Flowchart


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


Create Character
Read Characters
Input Name, Role, Level, Power
Display Characters
Input Valid?
No
Add to characters list
Exit Program
Program End


List Users
Read Characters
Update Character
Display All Users
Show Character List
Input ID
ID Valid?
No
Yes
Edit Character Fields


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


Delete Character
Input ID
Character Found?
No
Yes
Confirm Delete
No
Yes
Delete Character
View Detail Character
Input ID
Character Found?
No
Yes
Show Detail + Power Bar

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

def register_user(users):
    if os.name == "nt":
        os.system("cls")
    else:
        os.system("clear")
    print("== REGISTER USER ==")
    while True:
        username = input("Enter new username: ").strip()
        if username == "":
            print("Username cannot be empty.")
            continue
        exists = any(u[0] == username for u in users)
        if exists:
            print("Username already taken.")
            continue
        password = input("Enter password: ").strip()
        if password == "":
            print("Password cannot be empty.")
            continue
```

```
        users.append([username, password, "user"])
        print(f"User '{username}' registered successfully!")
        input("\nPress Enter to continue...")
        break
    return users

users = [["admin", "admin123", "admin"]]
characters = [
    [1, "Aegis", "Tank", 5, 80],
    [2, "Lumina", "Mage", 3, 45],
]

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

    print("=== MANAGEMENT DATA CHARACTER GAME ===")
    print("1. Register")
    print("2. Login")
    print("3. Exit")
    choice = input("Choose option: ").strip()

    if choice == "1":
        users = register_user(users)
        continue

    elif choice == "2":
        if os.name == "nt":
            os.system("cls")
        else:
            os.system("clear")
```

```
        print("== LOGIN ==")
        username = input("Username: ").strip()
        password = input("Password: ").strip()
        logged_user = None
        for u in users:
            if u[0] == username and u[1] == password:
                logged_user = u
                break
        if logged_user is None:
            print("Invalid credentials.")
            input("\nPress Enter to continue...")
            continue
        print(f"Welcome, {logged_user[0]}
({logged_user[2]})")
        input("\nPress Enter to continue...")

        if logged_user[2] == "admin":
            while True:
                if os.name == "nt":
                    os.system("cls")
                else:
                    os.system("clear")
                print(f"== ADMIN MENU ({logged_user[0]})
==")
                print("1. Create Character")
                print("2. Read Characters")
                print("3. Update Character")
                print("4. Delete Character")
                print("5. Register User")
                print("6. List Users")
                print("7. Logout")
                print("0. Exit")
                ch = input("Choose: ").strip()
```

```
                if ch == "1":
                    if os.name == "nt":
                        os.system("cls")
                    else:
                        os.system("clear")
                    print("== CREATE CHARACTER ==")
                    name = input("Name: ").strip()
                    if name == "":
                        print("Name cannot be empty.")
                        input("\nPress Enter to
continue...")
                        continue
                    role = input("Role: ").strip()
                    level_str = input("Level (>=1):
").strip()
                    if not level_str.isdigit():
                        print("Level must be number.")
                        input("\nPress Enter to
continue...")
                        continue
                    power_str = input("Power (0-100):
").strip()
                    if not power_str.isdigit():
                        print("Power must be number.")
                        input("\nPress Enter to
continue...")
                        continue
                    power = int(power_str)
                    if power < 0 or power > 100:
                        print("Power out of range.")
                        input("\nPress Enter to
continue...")
```

```
                        continue
                    new_id = max([c[0] for c in characters],
default=0) + 1
                    characters.append([new_id, name, role,
int(level_str), power])
                    print(f"Character '{name}' added.")
                    input("\nPress Enter to continue...")

                elif ch == "2":
                    if os.name == "nt":
                        os.system("cls")
                    else:
                        os.system("clear")
                    print("== CHARACTERS ==")
                    if len(characters) == 0:
                        print("No data.")
                    else:
                        print(f"{'ID':<4} | {'Name':<12} |
{'Role':<8} | {'Lvl':<3} | {'Power':<8}")
                        print("-"*45)
                        for c in characters:
                            bar = "[" + "" *
int((c[4]/100)*20) + "-" * (20-int((c[4]/100)*20)) + "]"
                            print(f"{c[0]:<4} | {c[1]:<12} |
{c[2]:<8} | {c[3]:<3} | {c[4]:<8} {bar}")
                    input("\nPress Enter to continue...")

                elif ch == "3":
                    if os.name == "nt":
                        os.system("cls")
                    else:
                        os.system("clear")
                    print("== UPDATE CHARACTER ==")
```

```
                    id_str = input("Enter ID: ").strip()
                    if not id_str.isdigit():
                        print("Invalid ID.")
                        input("\nPress Enter to
continue...")
                        continue
                    id_val = int(id_str)
                    target = next((c for c in characters if
c[0] == id_val), None)
                    if target is None:
                        print("Not found.")
                        input("\nPress Enter to
continue...")
                        continue
                    new_name = input(f"Name [{target[1]}]:
").strip()
                    new_role = input(f"Role [{target[2]}]:
").strip()
                    new_level = input(f"Level [{target[3]}]:
").strip()
                    new_power = input(f"Power [{target[4]}]:
").strip()
                    if new_name != "": target[1] = new_name
                    if new_role != "": target[2] = new_role
                    if new_level.isdigit(): target[3] =
int(new_level)
                    if new_power.isdigit():
                        p = int(new_power)
                        if 0 <= p <= 100: target[4] = p
                    print("Updated.")
                    input("\nPress Enter to continue...")

                elif ch == "4":
```

```
                    if os.name == "nt":
                        os.system("cls")
                    else:
                        os.system("clear")
                    print("== DELETE CHARACTER ==")
                    id_str = input("Enter ID: ").strip()
                    if not id_str.isdigit():
                        print("Invalid ID.")
                        input("\nPress Enter to
continue...")
                        continue
                    id_val = int(id_str)
                    idx = next((i for i, c in
enumerate(characters) if c[0] == id_val), -1)
                    if idx == -1:
                        print("Not found.")
                    else:
                        confirm = input(f"Delete
{characters[idx][1]}? (y/n): ").lower()
                        if confirm == "y":
                            characters.pop(idx)
                            print("Deleted.")
                        else:
                            print("Aborted.")
                    input("\nPress Enter to continue...")

                elif ch == "5":
                    users = register_user(users)

                elif ch == "6":
                    if os.name == "nt":
                        os.system("cls")
                    else:
```

```
                    os.system("clear")
                print("== USERS ==")
                for u in users:
                    print(f"- {u[0]} ({u[2]})")
                input("\nPress Enter to continue...")

            elif ch == "7":
                break

            elif ch == "0":
                sys.exit(0)

            else:
                print("Invalid.")
                input("\nPress Enter to continue...")

    else:
        while True:
            if os.name == "nt":
                os.system("cls")
            else:
                os.system("clear")
            print(f"== USER MENU ({logged_user[0]}) ==")
            print("1. Read Characters")
            print("2. View Detail")
            print("3. Logout")
            print("0. Exit")
            ch = input("Choose: ").strip()

            if ch == "1":
                if os.name == "nt":
                    os.system("cls")
                else:
```

```
                        os.system("clear")
                    if len(characters) == 0:
                        print("No characters.")
                    else:
                        print(f"{'ID':<4} | {'Name':<12} |
{'Role':<8} | {'Lvl':<3} | {'Power':<8}")
                        print("-"*45)
                        for c in characters:
                            bar = "[" + "" *
int((c[4]/100)*20) + "-" * (20-int((c[4]/100)*20)) + "]"
                            print(f"{c[0]:<4} | {c[1]:<12} |
{c[2]:<8} | {c[3]:<3} | {c[4]:<8} {bar}")
                    input("\nPress Enter to continue...")

                elif ch == "2":
                    id_str = input("Enter ID: ").strip()
                    if not id_str.isdigit():
                        print("Invalid ID.")
                        input("\nPress Enter to
continue...")
                        continue
                    id_val = int(id_str)
                    found = next((c for c in characters if
c[0] == id_val), None)
                    if found is None:
                        print("Not found.")
                    else:
                        print(f"ID: {found[0]}")
                        print(f"Name: {found[1]}")
                        print(f"Role: {found[2]}")
                        print(f"Level: {found[3]}")
                        print(f"Power: {found[4]}")
```

```
                        bar = "[" + "" *
int((found[4]/100)*30) + "-" * (30-int((found[4]/100)*30)) +
"]"
                        print("Visual:", bar)
                    input("\nPress Enter to continue...")

                elif ch == "3":
                    break

                elif ch == "0":
                    sys.exit(0)

                else:
                    print("Invalid.")
                    input("\nPress Enter to continue...")

    elif choice == "3":
        print("Goodbye!")
        sys.exit(0)

    else:
        print("Invalid option.")
        input("\nPress Enter to continue...")
```

FILE 2 REGISTER USER

```
def register_user(users):
    """Fungsi untuk registrasi user baru."""
    print("== REGISTER USER ==")
    while True:
        username = input("Enter new username: ").strip()
        if username == "":
```

```
            print("Username cannot be empty.")
            continue
        # Cek apakah sudah ada
        exists = any(u[0] == username for u in users)
        if exists:
            print("Username already taken.")
            continue
        password = input("Enter password: ").strip()
        if password == "":
            print("Password cannot be empty.")
            continue
        users.append([username, password, "user"])
        print(f"User '{username}' registered successfully!")
        input("\nPress Enter to continue...")
        break
    return users
```

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`.