

LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN DASAR



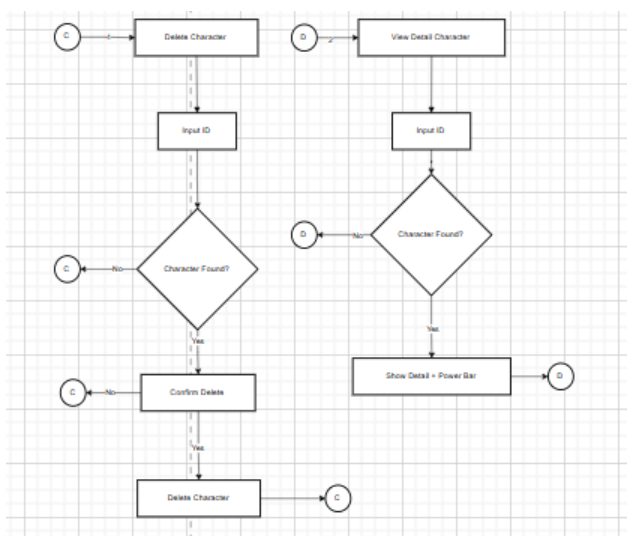
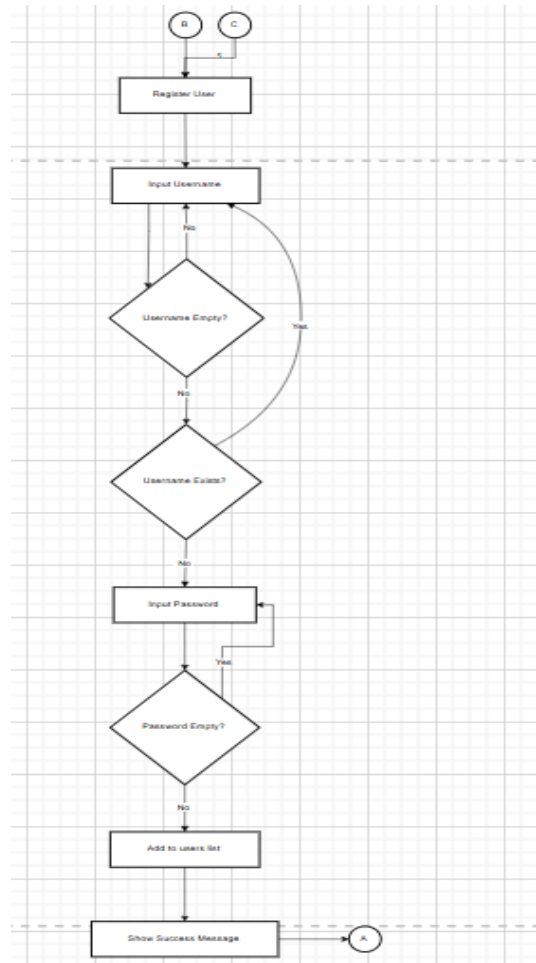
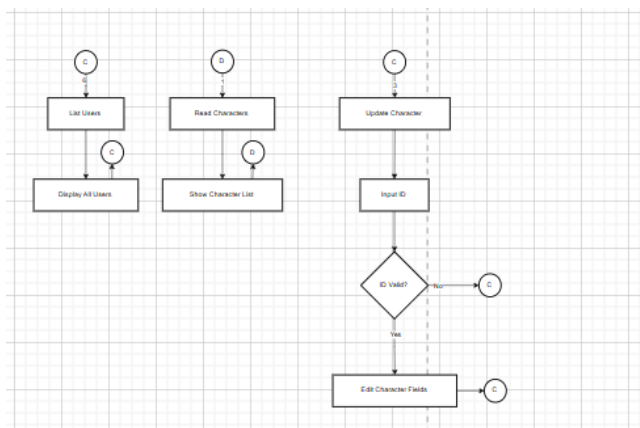
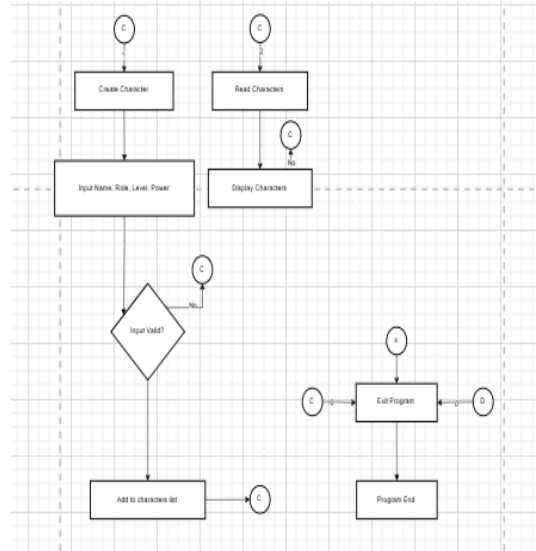
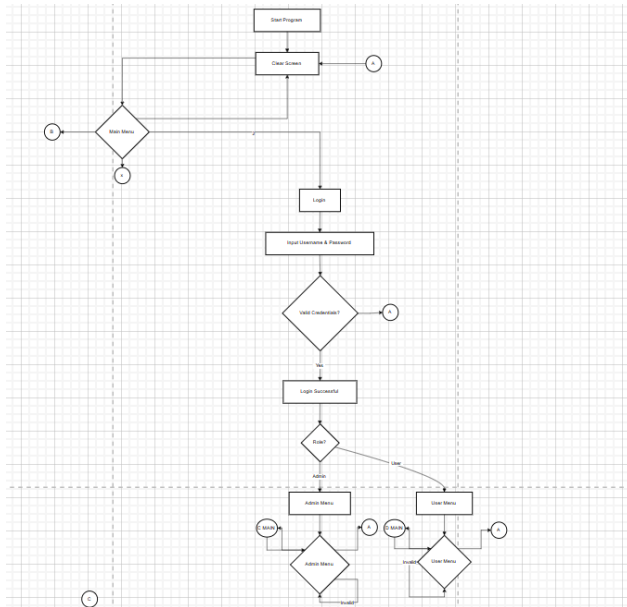
Heiza Rizki Pratama

2509106019

A'25

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



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

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}
}

# main prog

while True:
    # Clear layar
    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")
menu = input("Choose option: ").strip()

# regist pros
if menu == "1":
    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
        if username in users:
            print("Username already taken.")
            continue
        password = input("Enter password: ").strip()
        if password == "":
            print("Password cannot be empty.")
            continue
        users[username] = {"password": password, "role":
"user"}
        print(f"User '{username}' registered
successfully!")
        input("\nPress Enter to continue...")
        break

```

```

# login
elif menu == "2":
    if os.name == "nt":
        os.system("cls")
    else:
        os.system("clear")
    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"Welcome, {username} ({role})")
        input("\nPress Enter to continue...")

# if admin
if role == "admin":
    while True:
        if os.name == "nt":
            os.system("cls")
        else:
            os.system("clear")
        print(f"== ADMIN MENU ({username}) ==")
        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")
        admin_choice = input("Choose: ").strip()

```

```

# create
if admin_choice == "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_c = input("Role: ").strip()
    lvl = input("Level (>=1): ").strip()
    if not lvl.isdigit() or int(lvl) <
1:

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

        continue
    power = input("Power (0-100):
").strip()

    if not power.isdigit() or not (0 <=
int(power) <= 100):

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

        continue
    new_id = max(characters.keys(),
default=0) + 1

```

```

        characters[new_id] = {"name": name,
"role": role_c, "level": int(lvl), "power": int(power)}
        print(f"Character '{name}' added.")
        input("\nPress Enter to
continue...")

# read
elif admin_choice == "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 cid, c in
characters.items():
            bar = "[" + "█" *
int((c["power"]/100)*20) + "-" *
(20-int((c["power"]/100)*20)) + "]"
            print(f"{cid:<4} |
{c['name']:<12} | {c['role']:<8} | {c['level']:<3} |
{c['power']:<8} {bar}")
            input("\nPress Enter to
continue...")

# upt
elif admin_choice == "3":
    if os.name == "nt":

```

```

        os.system("cls")
    else:
        os.system("clear")
    print("== UPDATE CHARACTER ==")
    cid = input("Enter ID: ").strip()
    if not cid.isdigit() or int(cid) not
in characters:

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

        continue
    cid = int(cid)
    c = characters[cid]
    new_name = input(f"Name
[{c['name']}]: ").strip()
    new_role = input(f"Role
[{c['role']}]: ").strip()
    new_level = input(f"Level
[{c['level']}]: ").strip()
    new_power = input(f"Power
[{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("Updated successfully.")
    input("\nPress Enter to
continue...")

```



```

        # del
        elif admin_choice == "4":
            if os.name == "nt":
                os.system("cls")
            else:
                os.system("clear")
            print("== DELETE CHARACTER ==")
            cid = input("Enter ID: ").strip()
            if not cid.isdigit() or int(cid) not
in characters:

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

                continue
            cid = int(cid)
            confirm = input(f"Delete
{characters[cid]['name']}? (y/n): ").lower()
            if confirm == "y":
                del characters[cid]
                print("Deleted.")
            else:
                print("Aborted.")
                input("\nPress Enter to
continue...")

        # reg usr (admin)
        elif admin_choice == "5":
            if os.name == "nt":
                os.system("cls")
            else:
                os.system("clear")
            print("== REGISTER USER ==")
            while True:

```

```

new_user = input("Enter new
username: ").strip()

if new_user == "":
    print("Username cannot be
empty.")

    continue
if new_user in users:
    print("Username already
taken.")

    continue
new_pass = input("Enter
password: ").strip()

if new_pass == "":
    print("Password cannot be
empty.")

    continue
users[new_user] = {"password":
new_pass, "role": "user"}

print(f"User '{new_user}'
registered successfully!")

input("\nPress Enter to
continue...")

break

# users list
elif admin_choice == "6":
    if os.name == "nt":
        os.system("cls")
    else:
        os.system("clear")
    print("== USERS ==")
    for u, info in users.items():
        print(f"- {u} ({info['role']})")

```

```

        input("\nPress Enter to
continue...")

    elif admin_choice == "7":
        break

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

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

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

        if choice == "1":
            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 cid, c in characters.items():
                bar = "[" + "█" *
int((c["power"]/100)*20) + "-" *
(20-int((c["power"]/100)*20)) + "]"
                print(f"{cid:<4} | {c['name']:<12} | {c['roLe']:<8} | {c['level']:<3} | {c['power']:<8} {bar}")
            input("\nPress Enter to continue...")

        elif choice == "2":
            cid = input("Enter ID: ").strip()
            if not cid.isdigit() or int(cid) not
in characters:
                print("Invalid ID.")
                input("\nPress Enter to continue...")

            continue
            cid = int(cid)
            c = characters[cid]
            print(f"ID: {cid}")
            print(f"Name: {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)
        input("\nPress Enter to
continue...")

    elif choice == "3":
        break

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

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

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

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

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

```

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.