

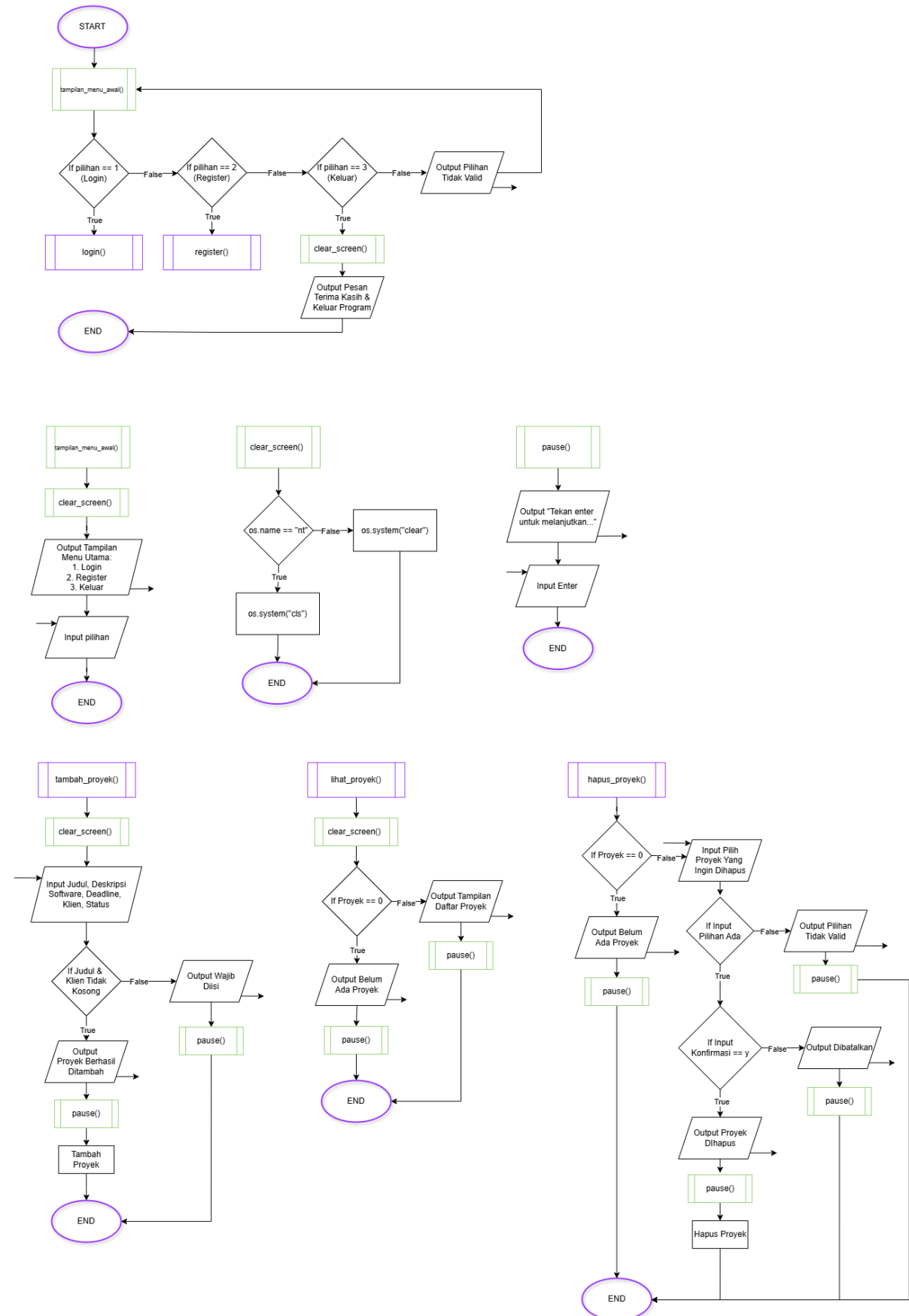
**LAPORAN PRAKTIKUM**  
**POSTTEST 7**  
**ALGORITMA PEMROGRAMAN DASAR**

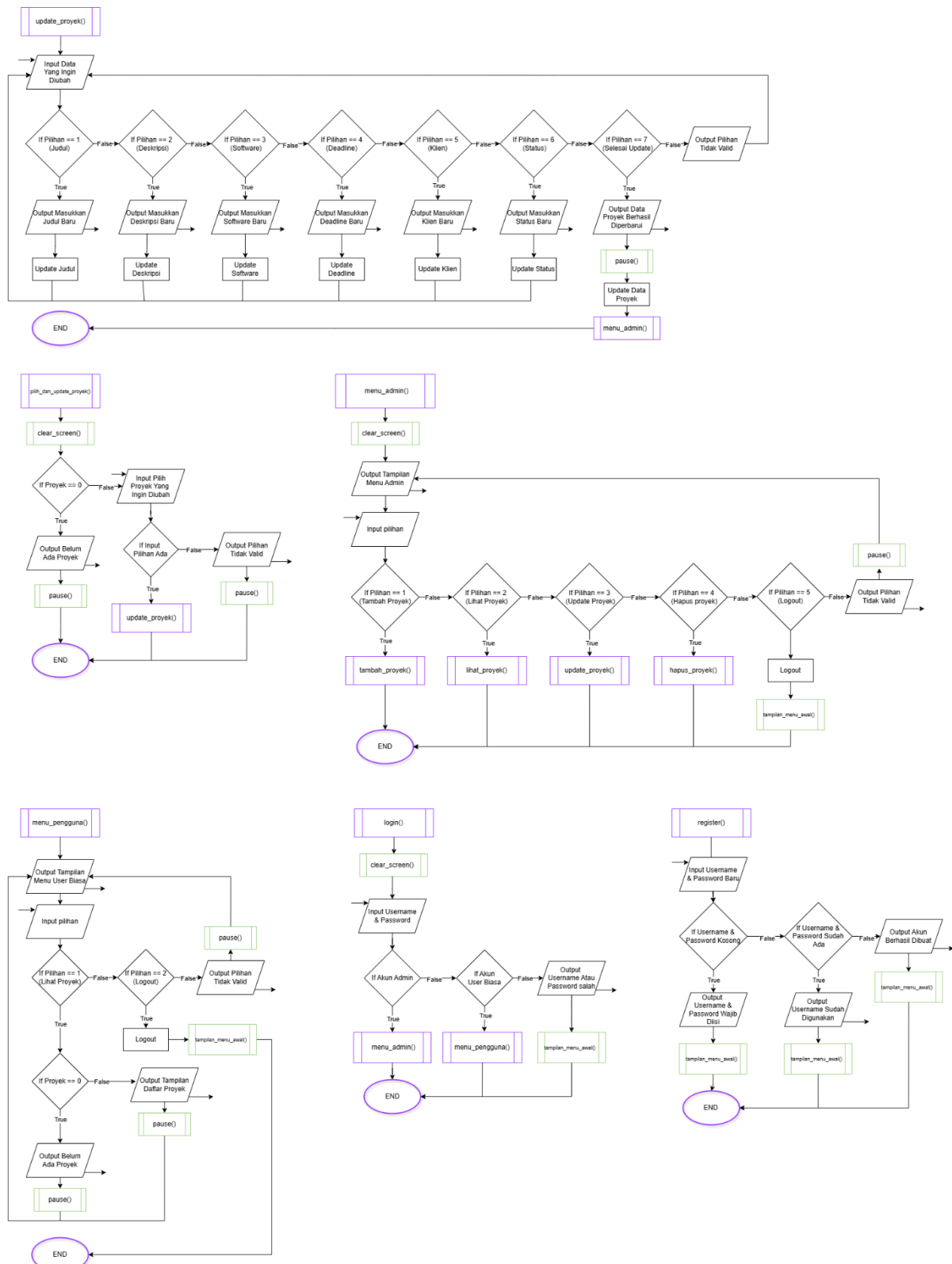


**Disusun oleh:**  
**Muhammad Ihsan Rosadi 2509106081**  
**Kelas B2'25**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

# 1. Flowchart





Gambar 1.1 Flowchart

Penjelasan singkat:

Flowchart diatas menggambarkan alur sistem manajemen proyek Desain Grafis yang memiliki dua peran utama: User Biasa dan Admin.

### 1. Menu Utama

- a. Sistem dimulai dengan menampilkan menu utama kepada pengguna.
- b. Pengguna dapat memilih untuk:
  1. Login sebagai Admin
  2. Login sebagai User Biasa
  3. Keluar dari sistem

Jika input pilihan tidak valid, sistem menampilkan pesan “Pilihan Tidak Valid”.

### 2. Login

- a. Pengguna memasukkan Username dan Password.
- b. Sistem akan memverifikasi data login:
  1. Jika username atau password salah → pesan error ditampilkan.
  2. Jika benar → pengguna diarahkan ke menu sesuai peran (Admin/User).

### 3. Menu Admin

Admin memiliki beberapa opsi utama:

- a. Tambah Proyek – Admin memasukkan judul, deskripsi, status, dan data proyek baru.
- b. Lihat Proyek – Menampilkan daftar proyek yang ada.
- c. Edit Proyek – Admin dapat mengubah data proyek tertentu (judul, deskripsi, software, deadline, client, status, dsb).
- d. Hapus Proyek – Menghapus proyek dari sistem.
- e. Logout – Keluar dari sistem.

Setiap langkah dilengkapi validasi, misalnya jika proyek tidak ditemukan atau input salah, akan muncul pesan “Pilihan Tidak Valid” atau “Proyek Belum Ada”.

### 4. Menu User Biasa

User Biasa hanya dapat:

- a. Lihat Daftar Proyek yang tersedia.
- b. Logout dari sistem.

Jika pilihan salah, sistem memberi pesan “Pilihan Tidak Valid”.

### 5. Validasi dan Pengulangan

- a. Jika terdapat input yang tidak sesuai, sistem akan meminta pengguna untuk mengulang pilihan.
- b. Setelah aksi selesai (misalnya menambah atau mengedit proyek), sistem kembali ke menu utama peran masing-masing.

### 6. End

Flowchart berakhir setelah pengguna memilih keluar dari sistem.

## 2. Deskripsi Singkat Program

Program ini merupakan sistem manajemen proyek desain grafis berbasis terminal yang memungkinkan pengguna melakukan login, registrasi, dan mengelola proyek. Program ini membedakan antara akun admin dan user biasa: admin dapat menambah, melihat, memperbarui, dan menghapus proyek, sedangkan user biasa hanya dapat melihat daftar proyek. Data pengguna dan proyek disimpan dalam daftar Python sederhana (list), dan antarmuka menggunakan menu teks yang bersih dengan perintah `os.system('cls'/'clear')` untuk membersihkan layar. Program ini berjalan dalam loop utama hingga pengguna memilih untuk keluar.

## 3. Source Code

### A. Inisialisasi & Variable Global

Menyimpan data pengguna, proyek, dan variabel global untuk pengguna yang sedang login.

```
import os

users = [
    {"username": "mihsanr81", "password": "2509106081", "role": "admin"}
]
projects = []
user_ditemukan = None
```

### B. Fungsi Utilitas (Pendukung Tampilan)

Membersihkan layar dan menunggu input sebelum lanjut, agar tampilan program lebih rapi.

```
def clear_screen():
    os.system('cls' if os.name == 'nt' else 'clear')

def pause():
    input("Tekan Enter Untuk Melanjutkan...")
```

### C. Menu Awal Program

Menampilkan menu utama (Login, Register, Keluar) dan mengembalikan pilihan pengguna.

```
def tampilkan_menu_awal():
    clear_screen()
    print("=== SISTEM MANAJEMEN PROYEK DESAIN GRAFIS ===")
    print("1. Login")
    print("2. Register")
    print("3. Keluar")
    return input("Pilih menu: ")
```

### D. Fitur Tambah Proyek

Menambahkan proyek baru ke daftar dengan validasi input.

```
def tambah_proyek(username):
```

```

global projects
clear_screen()
print("=== TAMBAH PROYEK ===")
try:
    judul = input("Judul: ")
    deskripsi = input("Deskripsi: ")
    software = input("Software: ")
    deadline = input("Deadline: ")
    klien = input("Klien: ")
    status = input("Status: ")

    if judul == "" or klien == "":
        raise ValueError("Judul dan Klien wajib diisi!")

    project_baru = {
        "judul": judul,
        "deskripsi": deskripsi,
        "software": software,
        "deadline": deadline,
        "klien": klien,
        "status": status
    }
    projects.append(project_baru)
    print("Proyek berhasil ditambah!")
except ValueError as e:
    print(e)
finally:
    pause()

```

## E. Fitur Lihat Proyek

Menampilkan semua data proyek yang tersimpan.

```

def lihat_proyek():
    clear_screen()
    print("=== DAFTAR PROYEK ===")
    if len(projects) == 0:
        print("Belum ada proyek.")
    else:
        for i, p in enumerate(projects):
            print(f"\n[{i+1}] {p['judul']}")
            print(f"Deskripsi: {p['deskripsi']}")
            print(f"Software : {p['software']}")
            print(f"Deadline : {p['deadline']}")
            print(f"Klien : {p['klien']}")
            print(f"Status : {p['status']}")
    pause()

```

## F. Fitur Update Proyek

Mengubah data proyek berdasarkan nomor indeks yang dipilih.

```

def update_proyek(index):
    global projects
    try:
        if len(projects) == 0:
            raise Exception("Belum ada proyek untuk diubah.")
        if not (0 <= index < len(projects)):
            raise IndexError("Nomor proyek tidak valid!")

        while True:
            clear_screen()
            print(f"=== UBAH PROYEK: {projects[index]['judul']} ===")
            print("1. Judul")
            print("2. Deskripsi")
            print("3. Software")
            print("4. Deadline")
            print("5. Klien")
            print("6. Status")
            print("7. Selesai Update")
            pilih = input("Pilih data yang ingin diubah: ")

            if pilih == "1":
                projects[index]["judul"] = input("Masukkan judul baru: ")
            elif pilih == "2":
                projects[index]["deskripsi"] = input("Masukkan deskripsi baru: ")
            elif pilih == "3":
                projects[index]["software"] = input("Masukkan software baru: ")
            elif pilih == "4":
                projects[index]["deadline"] = input("Masukkan deadline baru: ")
            elif pilih == "5":
                projects[index]["klien"] = input("Masukkan nama klien baru: ")
            elif pilih == "6":
                projects[index]["status"] = input("Masukkan status baru: ")
            elif pilih == "7":
                break
            else:
                print("Pilihan tidak valid!")
                pause()

            print("Data proyek berhasil diperbarui!")
    except Exception as e:
        print(e)
    finally:
        pause()

```

## G. Fitur Hapus Proyek

Menghapus proyek berdasarkan pilihan nomor dari daftar.

```

def hapus_proyek():
    global projects
    clear_screen()

```

```

print("=== HAPUS PROYEK ===")
try:
    if len(projects) == 0:
        raise Exception("Belum ada proyek.")
    for i, p in enumerate(projects):
        print(f"[{i+1}] {p['judul']}")
    nomor = input("Pilih nomor proyek: ")
    if not nomor.isdigit():
        raise ValueError("Input harus angka.")

    nomor = int(nomor) - 1
    if not (0 <= nomor < len(projects)):
        raise IndexError("Nomor tidak valid.")

    hapus = input(f"Hapus proyek '{projects[nomor]['judul']}'? (y/n): ").lower()
    if hapus == "y":
        del projects[nomor]
        print("Proyek dihapus!")
    else:
        print("Dibatalkan.")
except (ValueError, IndexError, Exception) as e:
    print(e)
finally:
    pause()

```

## H. Fitur Pilih dan Update Proyek

Menampilkan daftar proyek dan meminta pengguna memilih proyek yang ingin diubah.

```

def pilih_dan_update_proyek():
    clear_screen()
    print("=== UBAH PROYEK ===")
    if len(projects) == 0:
        print("Belum ada proyek.")
        pause()
        return
    for i, p in enumerate(projects):
        print(f"[{i+1}] {p['judul']}")
    nomor = input("\nPilih nomor proyek: ")
    if nomor.isdigit():
        update_proyek(int(nomor) - 1)
    else:
        print("Input harus angka!")
        pause()

```

## I. Menu Admin

Menu khusus admin untuk melakukan CRUD proyek dan logout.

```

def menu_admin():
    clear_screen()

```



```

print("=== MENU ADMIN ===")
print("1. Tambah Proyek")
print("2. Lihat Proyek")
print("3. Update Proyek")
print("4. Hapus Proyek")
print("5. Logout")

pilih = input("Pilih: ")

if pilih == "1":
    tambah_proyek(user_ditemukan["username"])
elif pilih == "2":
    lihat_proyek()
elif pilih == "3":
    pilih_dan_update_proyek()
elif pilih == "4":
    hapus_proyek()
elif pilih == "5":
    return
else:
    print("Pilihan tidak valid!")
    pause()

menu_admin()

```

## J. Menu Pengguna Biasa

Menu sederhana bagi user biasa untuk melihat proyek dan logout.

```

def menu_pengguna():
    while True:
        clear_screen()
        print(f"=== MENU PENGGUNA ({user_ditemukan['username']}) ===")
        print("1. Lihat Proyek")
        print("2. Logout")
        pilih_user = input("Pilih: ")

        if pilih_user == "1":
            lihat_proyek()
        elif pilih_user == "2":
            break
        else:
            print("Pilihan tidak valid!")
            pause()

```

## K. Fitur Login

Memverifikasi akun lalu arahkan ke menu admin/pengguna.

```

def login():
    global user_ditemukan
    clear_screen()

```

```

print("=== LOGIN ===")
username = input("Username: ")
password = input("Password: ")

user_ditemukan = None
for u in users:
    if u["username"] == username and u["password"] == password:
        user_ditemukan = u
        break

if user_ditemukan is None:
    print("Login gagal! Username atau password salah.")
    pause()
    return

if user_ditemukan["role"] == "admin":
    menu_admin()
else:
    menu_pengguna()

```

## L. Fitur Register

Membuat akun baru dengan pengecekan username duplikat.

```

def register():
    clear_screen()
    print("=== REGISTER ===")
    username = input("Username baru: ")
    password = input("Password: ")
    sudah_ada = any(u["username"] == username for u in users)

    if username == "" or password == "":
        print("Username dan password wajib diisi!")
    elif sudah_ada:
        print("Username sudah digunakan!")
    else:
        users.append({"username": username, "password": password, "role":
"user"})
        print("Akun berhasil dibuat!")
        pause()

```

## M. Fungsi Utama Program

Menjalankan alur utama program, menampilkan menu awal, dan mengatur navigasi antar fitur.

```

def main():
    while True:
        menu = tampilkan_menu_awal()
        if menu == "1":
            login()
        elif menu == "2":
            register()

```

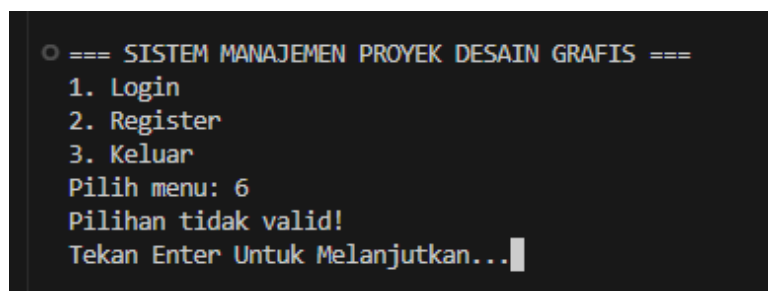
```

elif menu == "3":
    clear_screen()
    print("Terima kasih telah menggunakan program ini.")
    break
else:
    print("Pilihan tidak valid!")
    pause()

main()

```

## 4. Hasil Output

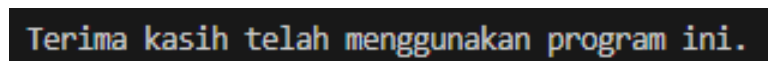


```

O === SISTEM MANAJEMEN PROYEK DESAIN GRAFIS ===
1. Login
2. Register
3. Keluar
Pilih menu: 6
Pilihan tidak valid!
Tekan Enter Untuk Melanjutkan...

```

Gambar 4.1 Fitur Menu Utama/Admin/User Biasa (Input Nomor Invalid)

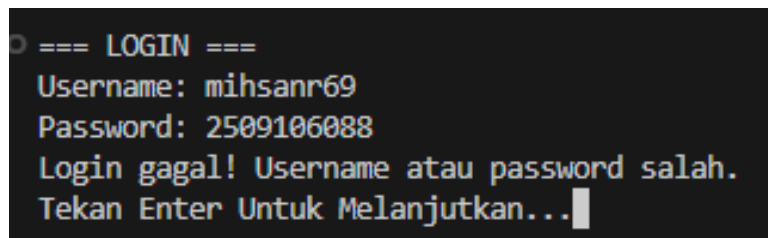


```

Terima kasih telah menggunakan program ini.

```

Gambar 4.2 Fitur Keluar

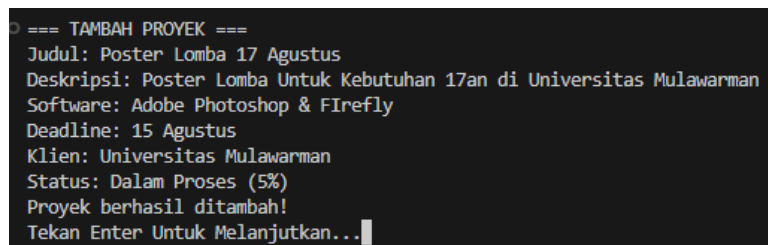


```

O === LOGIN ===
Username: mihsanr69
Password: 2509106088
Login gagal! Username atau password salah.
Tekan Enter Untuk Melanjutkan...

```

Gambar 4.3 Fitur Login (Gagal)



```

O === TAMBAH PROYEK ===
Judul: Poster Lomba 17 Agustus
Deskripsi: Poster Lomba Untuk Kebutuhan 17an di Universitas Mulawarman
Software: Adobe Photoshop & Firefly
Deadline: 15 Agustus
Klien: Universitas Mulawarman
Status: Dalam Proses (5%)
Proyek berhasil ditambah!
Tekan Enter Untuk Melanjutkan...

```

Gambar 4.4 Fitur Create/Menambah Proyek (Berhasil)

```

O === TAMBAH PROYEK ===
Judul:
Deskripsi:
Software:
Deadline:
Klien:
Status:
Wajib diisi!
Tekan Enter Untuk Melanjutkan...

```

Gambar 4.5 Fitur Create/Menambah Proyek (Gagal)

```

O === DAFTAR PROYEK ===

[1] Poster Lomba 17 Agustus
Deskripsi: Poster Lomba Untuk Kebutuhan 17an di Universitas Mulawarman
Software : Adobe Photoshop & Firefly
Deadline : 15 Agustus
Klien : Universitas Mulawarman
Status : Dalam Proses (5%)

[2] Logo Lomba 17 Agustus
Deskripsi: Logo dengan tema teknologi untuk kebutuhan lomba
Software : Adobe Illustrator, CorelDraw
Deadline : 15 Agustus
Klien : Diri sendiri
Status : Dalam proses (10%)

Tekan Enter Untuk Melanjutkan...

```

Gambar 4.6 Fitur Read/Melihat Proyek (Ada Proyek)

```

=== DAFTAR PROYEK ===
Belum ada proyek.

Tekan Enter Untuk Melanjutkan...

```

Gambar 4.7 Fitur Read/Melihat Proyek (Tidak Ada Proyek)

```

=== UBAH PROYEK: Poster Lomba 17 Agustus ===
1. Judul
2. Deskripsi
3. Software
4. Deadline
5. Klien
6. Status
7. Selesai Update
Pilih data yang ingin diubah: 4
Masukkan deadline baru: 13 Agustus

```

```

=== UBAH PROYEK: Poster Lomba 17 Agustus ===
1. Judul
2. Deskripsi
3. Software
4. Deadline
5. Klien
6. Status
7. Selesai Update
Pilih data yang ingin diubah: 7
Data proyek berhasil diperbarui!
Tekan Enter Untuk Melanjutkan...

```

Gambar 4.8 Fitur Update/Memperbarui Proyek (Ada Proyek) 2

```

=== UBAH PROYEK ===
Belum ada proyek.
Tekan Enter Untuk Melanjutkan...

```

Gambar 4.9 Fitur Update/Memperbarui Proyek (Tidak Ada Proyek)

```

=== UBAH PROYEK ===
[1] Poster Lomba 17 Agustus
[2] Logo Lomba 17 Agustus

Pilih nomor proyek: 4
Nomor proyek tidak valid.
Tekan Enter Untuk Melanjutkan...

```

Gambar 4.10 Fitur Update/Memperbarui Proyek (Input Nomor Invalid)

```

=== HAPUS PROYEK ===
[1] Poster Lomba 17 Agustus
[2] Logo Lomba 17 Agustus
Pilih nomor proyek: 2
Hapus proyek 'Logo Lomba 17 Agustus'? (y/n): y
Proyek dihapus!
Tekan Enter Untuk Melanjutkan...

```

Gambar 4.11 Fitur Delete/Menghapus Proyek (Ada Proyek (Input y))

```

=== HAPUS PROYEK ===
[1] Poster Lomba 17 Agustus
[2] Logo Lomba 17 Agustus
Pilih nomor proyek: 2
Hapus proyek 'Logo Lomba 17 Agustus'? (y/n): o
Dibatalkan.
Tekan Enter Untuk Melanjutkan...

```

Gambar 4.12 Fitur Delete/Menghapus Proyek (Ada Proyek (Input Selain y))

```

=== HAPUS PROYEK ===
Belum ada proyek.
Tekan Enter Untuk Melanjutkan...

```

Gambar 4.13 Fitur Delete/Menghapus Proyek (Tidak Ada Proyek)

```

=== HAPUS PROYEK ===
[1] Poster Lomba 17 Agustus
[2] Logo Lomba 17 Agustus
Pilih nomor proyek: 3
Nomor tidak valid.
Tekan Enter Untuk Melanjutkan...

```

Gambar 4.14 Fitur Delete/Menghapus Proyek (Input Nomor Invalid)

```

=== MENU PENGGUNA (mihsanr69) ===
1. Lihat Proyek
2. Logout
Pilih:

```

Gambar 4.15 Fitur Menu User Biasa

```

=== REGISTER ===
Username baru: mihsanr69
Password: 2555555555
Akun berhasil dibuat!
Tekan Enter Untuk Melanjutkan...

```

Gambar 4.16 Fitur Register (Berhasil)

```
=== REGISTER ===
Username baru:
Password:
Username dan password wajib diisi!
Tekan Enter Untuk Melanjutkan...
```

Gambar 4.16 Fitur Register (Gagal/Kosong)

```
=== REGISTER ===
Username baru: mihsanr69
Password: 2555555555
Username sudah digunakan!
Tekan Enter Untuk Melanjutkan...
```

Gambar 4.17 Fitur Register (Gagal/Sudah Ada)

## 5. Langkah-langkah GIT

### 5.1 GIT Add

Git Add berfungsi untuk menambahkan perubahan file dari working directory ke staging area, yaitu tempat sementara sebelum perubahan benar-benar disimpan dengan git commit, dengan kata lain, perintah ini menandai file atau perubahan tertentu agar disertakan dalam commit berikutnya.

```
● PS C:\Users\ASUS-GK\Downloads\praktikum-apd> git add .
```

Gambar 5.1.1 Git Add

### 5.2 GIT Commit

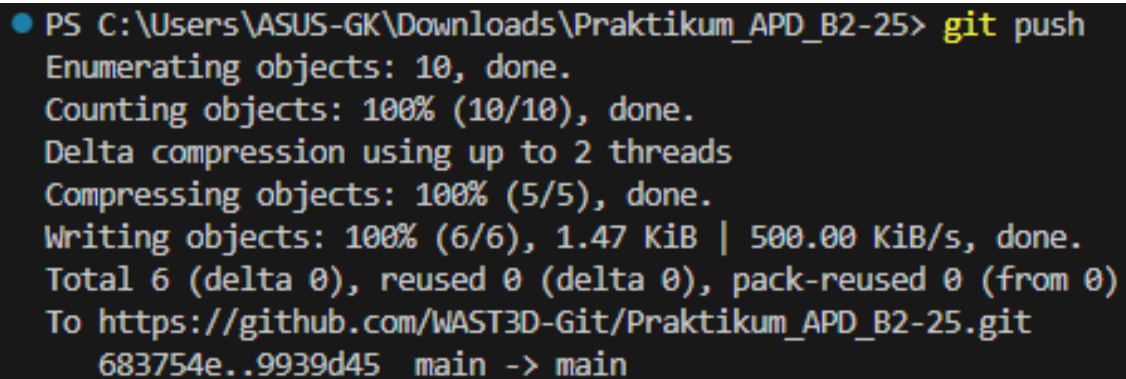
Git Commit berfungsi untuk menyimpan perubahan yang telah ditandai dengan git add ke dalam riwayat repository lokal, sehingga setiap versi proyek terdokumentasi dengan jelas melalui pesan commit yang menjelaskan perubahan tersebut.

```
● PS C:\Users\ASUS-GK\Downloads\Praktikum_APD_B2-25> git commit -m "ss buat pdf"
[main 9939d45] ss buat pdf
1 file changed, 85 insertions(+)
create mode 100644 praktikum-apd/post-test/post-test-apd-3/PT3.fprg
```

Gambar 5.2.1 Git Commit

### 5.3 GIT Push

Git Push adalah perintah Git yang digunakan untuk mengirim atau mengunggah perubahan dari repository lokal ke repository remote, sehingga commit yang sudah dibuat secara lokal dapat tersinkronisasi dan tersedia di server (misalnya GitHub atau GitLab) agar bisa diakses, digunakan, atau dikolaborasikan oleh orang lain.



```
● PS C:\Users\ASUS-GK\Downloads\Praktikum_APD_B2-25> git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 2 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 1.47 KiB | 500.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/WAST3D-Git/Praktikum_APD_B2-25.git
683754e..9939d45  main -> main
```

Gambar 5.3.1 Git Push