

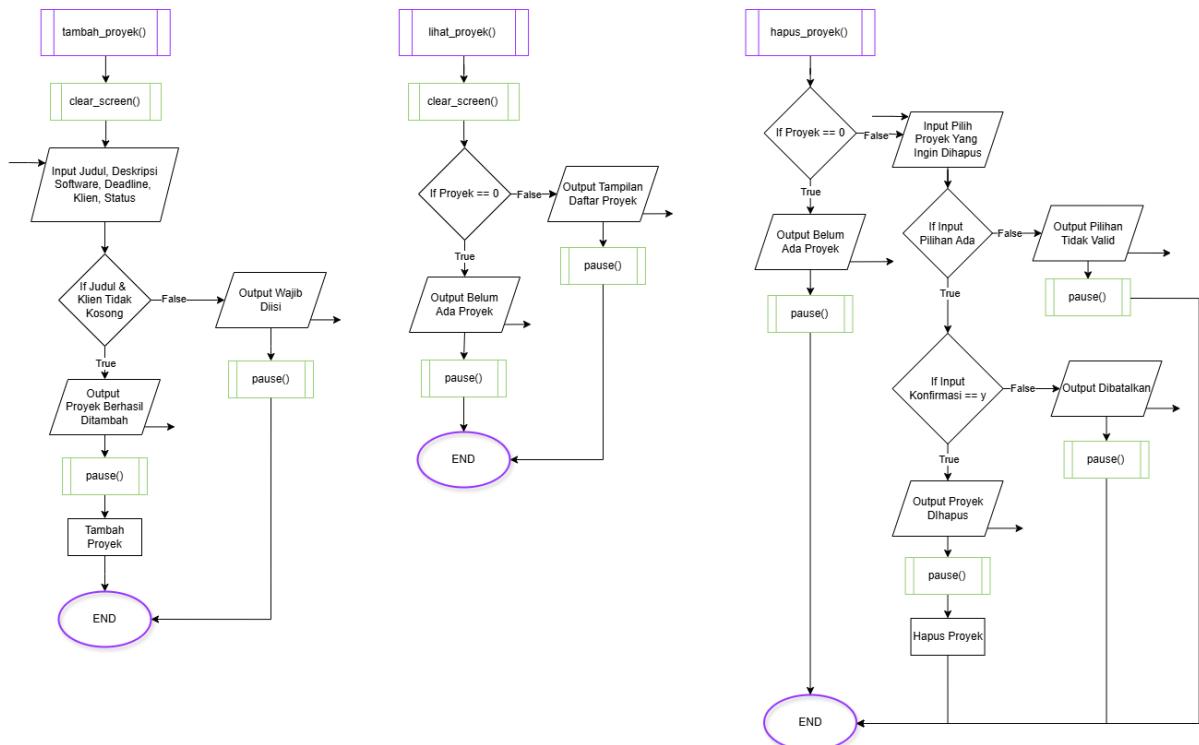
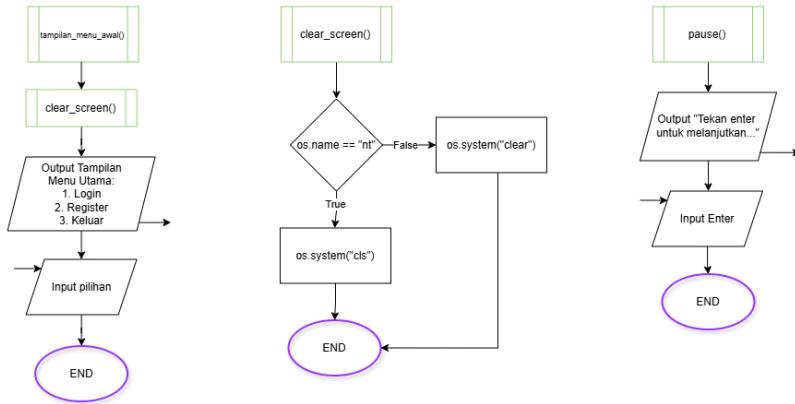
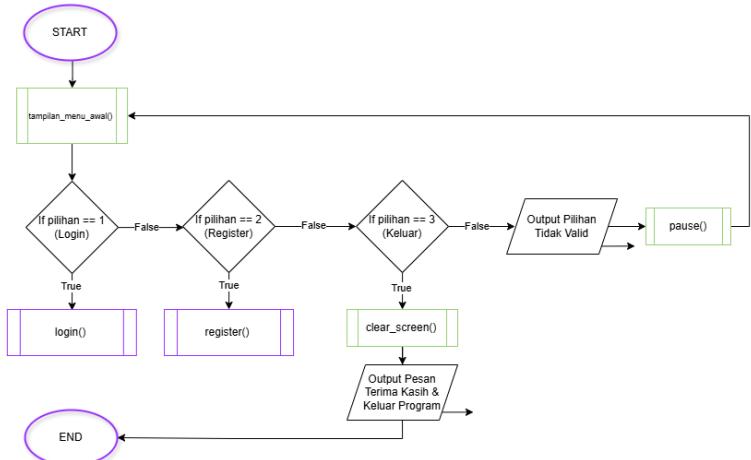
**LAPORAN PRAKTIKUM**  
**POSTTEST 8**  
**ALGORITMA PEMROGRAMAN DASAR**

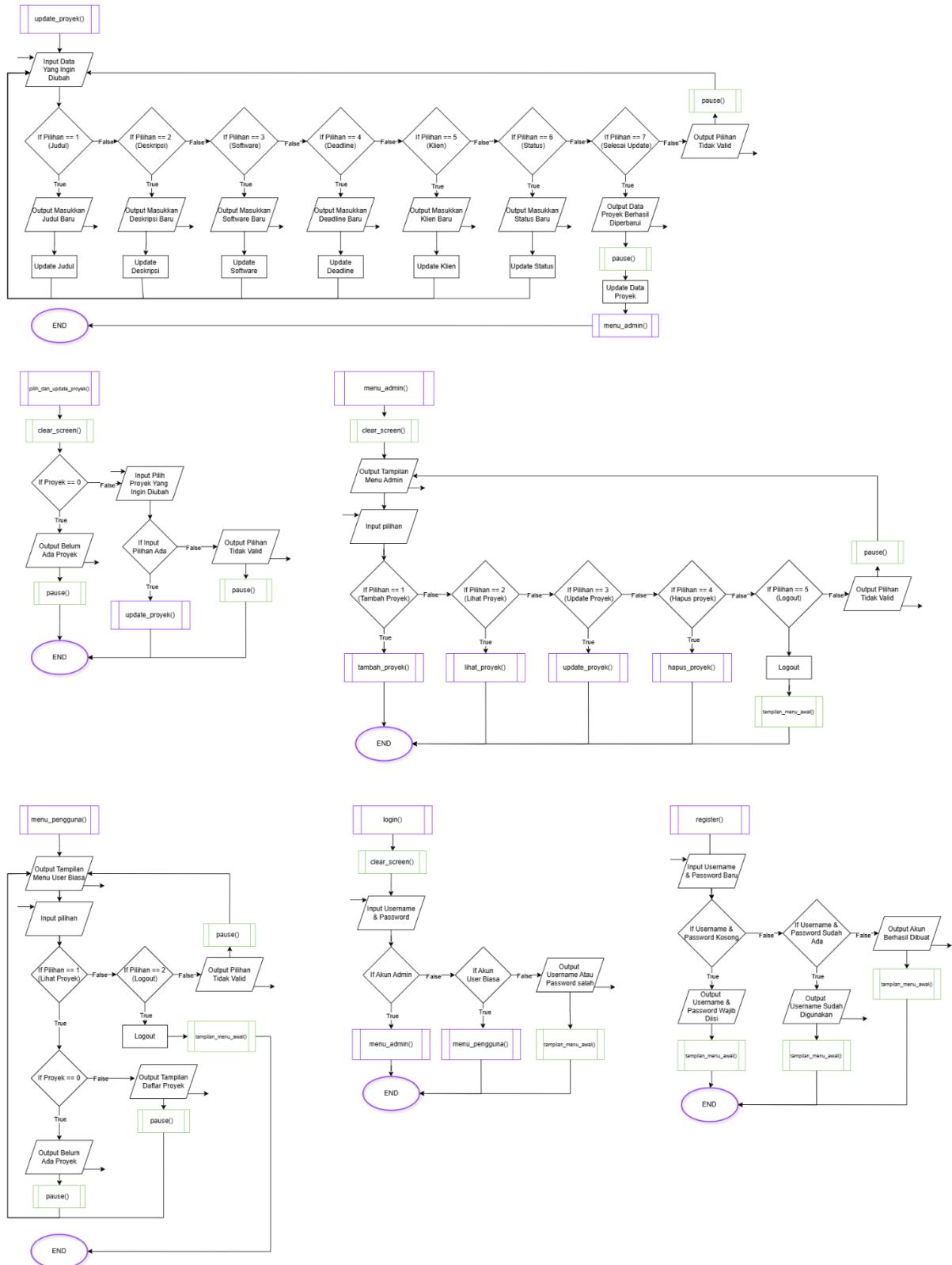


**Disusun oleh:**  
**Muhammad Ihsan Rosadi 2509106081**  
**Kelas B2'25**

**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. Login.py

Kode dalam file ini memverifikasi login dengan TinyDB dan menampilkan menu sesuai peran pengguna.

```
from tinydb import Query
from database.db_config import users_table
from utils.helpers import clear_screen, pause
from projects.project_menu import menu_admin, menu_pengguna

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

    user = users_table.get((Query().username == username) & (Query().password == password))

    if not user:
        print("Login gagal! Username atau password salah.")
        pause()
        return

    if user["role"] == "admin":
        menu_admin()
    else:
        menu_pengguna(user["username"])
```

### B. register.py

Kode dalam file ini membuat akun baru dengan menyimpan username dan password ke TinyDB jika belum terdaftar.

```
from tinydb import Query
from database.db_config import users_table
from utils.helpers import clear_screen, pause
```

```

def register():
    clear_screen()
    print("==== REGISTER ===")
    username = input("Username baru: ")
    password = input("Password: ")

    if not username or not password:
        print("Username dan password wajib diisi.")
    elif users_table.get(Query().username == username):
        print("Username sudah digunakan.")
    else:
        users_table.insert({"username": username, "password": password, "role": "user"})
        print("Akun berhasil dibuat.")
    pause()

```

### C. db\_config.py

Kode dalam file ini membuat koneksi ke database TinyDB dan menyiapkan tabel users serta projects untuk menyimpan data.

```

from tinydb import TinyDB

db = TinyDB('database/database.json')
users_table = db.table('users')
projects_table = db.table('projects')

```

### D. project\_crud.py

Kode dalam file ini mengelola data proyek (tambah, lihat, ubah, hapus) di TinyDB dengan tampilan tabel dan menu interaktif.

```

from prettytable import PrettyTable
from textwrap import wrap
import inquirer
from database.db_config import projects_table
from utils.helpers import clear_screen, pause

def lihat_proyek():
    clear_screen()
    print("==== DAFTAR PROYEK ===")
    projects = projects_table.all()

    if len(projects) == 0:
        print("Belum ada proyek.")
    else:
        table = PrettyTable()
        table.field_names = ["No", "Judul", "Deskripsi", "Software", "Deadline",
"Klien", "Status"]
        table.align = "l"
        max_widths = {"Judul": 20, "Deskripsi": 30, "Software": 15, "Deadline": 15}

```

```

15, "Klien": 15, "Status": 12}

    for i, p in enumerate(projects, start=1):
        row = [
            str(i),
            "\n".join(wrap(p.get("judul", ""), max_widths["Judul"])),
            "\n".join(wrap(p.get("deskripsi", ""), max_widths["Deskripsi"])),
            "\n".join(wrap(p.get("software", ""), max_widths["Software"])),
            "\n".join(wrap(p.get("deadline", ""), max_widths["Deadline"])),
            "\n".join(wrap(p.get("klien", ""), max_widths["Klien"])),
            "\n".join(wrap(p.get("status", ""), max_widths["Status"]))
        ]
        table.add_row(row)

    print(table)
    pause()

def tambah_proyek():
    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!")

        projects_table.insert({
            "judul": judul,
            "deskripsi": deskripsi,
            "software": software,
            "deadline": deadline,
            "klien": klien,
            "status": status
        })
        print("\nProyek berhasil ditambahkan.")
    except ValueError as e:
        print(e)
    finally:
        pause()

def pilih_dan_update_proyek():
    clear_screen()
    print("==== UBAH PROYEK ===")
    from database.db_config import projects_table
    projects = projects_table.all()

```

```

if projects:
    choices = [f"{i+1}. {p['judul']}" for i, p in enumerate(projects)]
    pilih = inquirer.prompt([inquirer.List("pilih", message="Pilih proyek",
choices=choices)])
    index = int(pilih["pilih"].split(".")[0]) - 1
    update_proyek(index)
else:
    print("Belum ada proyek.")
    pause()

def update_proyek(index):
    clear_screen()
    projects = projects_table.all()
    if not projects:
        print("Belum ada proyek.")
        pause()
        return

    clear_screen()
    while True:
        print(f"== UBAH PROYEK: {projects[index]['judul']} ==")
        pertanyaan = [
            inquirer.List("pilih", message="Pilih data yang ingin diubah",
choices=[
                "Judul", "Deskripsi", "Software", "Deadline", "Klien", "Status",
                "Selesai Update"
            ])
        ]
        jawaban = inquirer.prompt(pertanyaan)
        pilih = jawaban["pilih"]
        if pilih == "Selesai Update":
            break
        new_value = input(f"Masukkan {pilih.lower()} baru: ")
        projects_table.update({pilih.lower(): new_value},
doc_ids=[projects[index].doc_id])
        clear_screen()
        print("\nData proyek berhasil diperbarui.")
        pause()
def hapus_proyek():
    clear_screen()
    print("== HAPUS PROYEK ==")
    projects = projects_table.all()
    if not projects:
        print("Belum ada proyek.")
        pause()
        return

    choices = [f"{i+1}. {p['judul']}" for i, p in enumerate(projects)]
    questions = [inquirer.List("pilih", message="Pilih proyek yang ingin
dihapus", choices=choices)]
    jawaban = inquirer.prompt(questions)

```

```

nomor = int(jawaban["pilih"].split(".")[0]) - 1

konfirmasi = input(f"Hapus proyek '{projects[nomor]['judul']}'? (y/n):").lower()
if konfirmasi == "y":
    projects_table.remove(doc_ids=[projects[nomor].doc_id])
    print("Proyek berhasil dihapus.")
else:
    print("Dibatalkan.")
pause()

```

## E. project\_menu.py

Kode dalam file ini menampilkan menu interaktif untuk admin dan pengguna agar bisa mengelola atau melihat proyek sesuai perannya.

```

import inquirer
from utils.helpers import clear_screen
from .project_crud import tambah_proyek, lihat_proyek, hapus_proyek,
pilih_dan_update_proyek

def menu_admin():
    while True:
        clear_screen()
        print("== MENU ADMIN ==")
        pertanyaan = [
            inquirer.List("menu", message="Pilih tindakan:", choices=[
                "Tambah Proyek", "Lihat Proyek", "Update Proyek", "Hapus
Proyek", "Logout"
            ])
        ]
        jawaban = inquirer.prompt(pertanyaan)
        pilihan = jawaban["menu"]

        if pilihan == "Tambah Proyek":
            tambah_proyek()
        elif pilihan == "Lihat Proyek":
            lihat_proyek()
        elif pilihan == "Update Proyek":
            pilih_dan_update_proyek()
        elif pilihan == "Hapus Proyek":
            hapus_proyek()
        elif pilihan == "Logout":
            break

def menu_pengguna(username):
    while True:
        clear_screen()
        print(f"== MENU PENGGUNA ==")
        pertanyaan = [

```

```

        inquirer.List("menu", message="Pilih tindakan", choices=["Lihat
Proyek", "Logout"])
    ]
    jawaban = inquirer.prompt(pertanyaan)
    pilihan = jawaban["menu"]

    if pilihan == "Lihat Proyek":
        lihat_proyek()
    elif pilihan == "Logout":
        break

```

## F. helpers.py

Kode dalam file ini berfungsi untuk membersihkan layar terminal dan menjeda program hingga pengguna menekan Enter.

```

import os

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

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

```

## G. main.py

Kode dalam file ini menjalankan menu utama program manajemen proyek desain grafis, memungkinkan pengguna untuk login, register, atau keluar dari program.

```

import inquirer
from utils.helpers import clear_screen
from auth.login import login
from auth.register import register

def tampilan_menu_awal():
    clear_screen()
    print("== SISTEM MANAJEMEN PROYEK DESAIN GRAFIS ==")
    pertanyaan = [
        inquirer.List("menu", message="Pilih menu:", choices=["Login",
"Register", "Keluar"])
    ]
    jawaban = inquirer.prompt(pertanyaan)
    return jawaban["menu"]

def main():
    while True:
        pilihan = tampilan_menu_awal()
        if pilihan == "Login":
            login()
        elif pilihan == "Register":

```

```
register()
elif pilihan == "Keluar":
    clear_screen()
    print("Terima kasih telah menggunakan program ini.")
    break

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

#### 4. Hasil Output

```
Terima kasih telah menggunakan program ini.
```

Gambar 4.1 Fitur Keluar

```
==== LOGIN ====
Username: mihsanr69
Password: 2509106088
Login gagal! Username atau password salah.

Tekan Enter untuk melanjutkan...
```

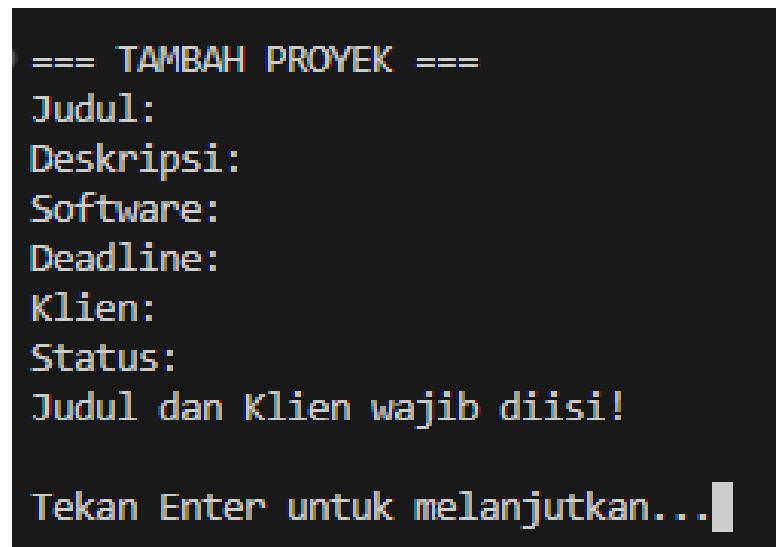
Gambar 4.2 Fitur Login (Gagal)

```
==== TAMBAH PROYEK ====
Judul: Poster Promo Makanan
Deskripsi: Promosi Makanan Lokal
Software: Photoshop
Deadline: 8 November
Klien: Warung Makan Lokal
Status: Dalam Proses

Proyek berhasil ditambahkan.

Tekan Enter untuk melanjutkan...
```

Gambar 4.3 Fitur Create/Menambah Proyek (Berhasil)

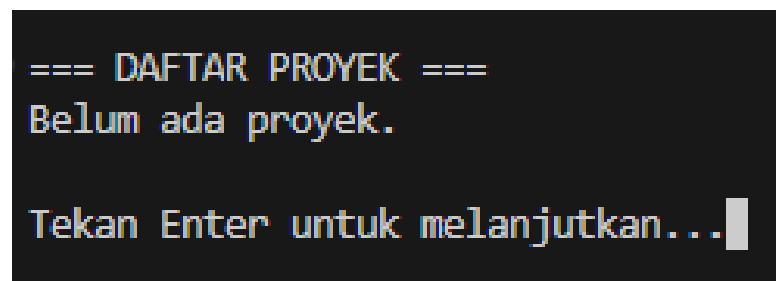


Gambar 4.4 Fitur Create/Menambah Proyek (Gagal)

== DAFTAR PROYEK ==						
No	Judul	Deskripsi	Software	Deadline	Klien	Status
1	Poster Promo Makanan	Promosi Makanan Lokal	Photoshop	8 November	Warung Makan Lokal	Dalam Proses
2	Logo Sekolah	Pembaruan logo sekolah	Illustrator, CorelDraw	10 November	SMK 15 Samarinda	Selesai
3	Infografis Covid-19	Infografis tentang covid-19	Canva	11 November	Pemerintah Daerah	Ditunda
4	Desain Brosur Event	Brosur untuk acara kampus tahunan	InDesign	13 November	Universitas Negeri	Dalam Proses

Tekan Enter untuk melanjutkan...■

Gambar 4.5 Fitur Read/Melihat Proyek (Ada Proyek)



Gambar 4.6 Fitur Read/Melihat Proyek (Tidak Ada Proyek)

```
==== UBAH PROYEK ====
[?] Pilih proyek:
> 1. Poster Promo Makanan
  2. Logo Sekolah
  3. Infografis Covid-19
  4. Desain Brosur Event

==== UBAH PROYEK: Poster Promo Makanan ====
[?] Pilih data yang ingin diubah:
  Judul
  Deskripsi
  Software
> Deadline
  Klien
  Status
  Selesai Update

Masukkan deadline baru: 12 November

==== UBAH PROYEK: Poster Promo Makanan ====
[?] Pilih data yang ingin diubah:
  Judul
  Deskripsi
  Software
  Deadline
  Klien
  Status
> Selesai Update

Data proyek berhasil diperbarui.

Tekan Enter untuk melanjutkan...
```

Gambar 4.7 Fitur Update/Memperbarui Proyek (Ada Proyek)

```
==== UBAH PROYEK ====
Belum ada proyek.

Tekan Enter untuk melanjutkan...|
```

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

```
==== HAPUS PROYEK ====
[?] Pilih proyek yang ingin dihapus:
> 1. Poster Promo Makanan
  2. Logo Sekolah
  3. Infografis Covid-19
  4. Desain Brosur Event

Hapus proyek 'Poster Promo Makanan'? (y/n): y
Proyek berhasil dihapus.

Tekan Enter untuk melanjutkan...|
```

Gambar 4.9 Fitur Delete/Menghapus Proyek (Ada Proyek (Input y))

```
==== HAPUS PROYEK ====
[?] Pilih proyek yang ingin dihapus:
  1. Poster Promo Makanan
  2. Logo Sekolah
> 3. Infografis Covid-19
  4. Desain Brosur Event

Hapus proyek 'Infografis Covid-19'? (y/n): o
Dibatalkan.

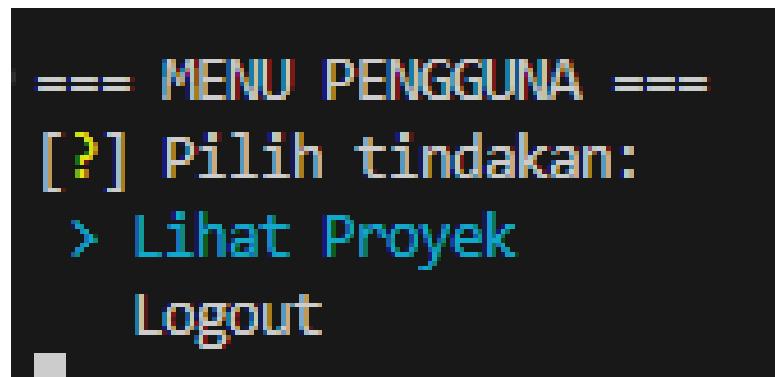
Tekan Enter untuk melanjutkan...|
```

Gambar 4.10 Fitur Delete/Menghapus Proyek (Ada Proyek (Input Selain y))

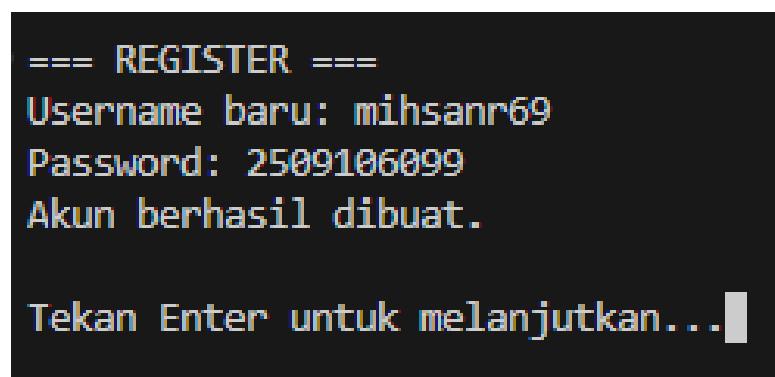
```
==== HAPUS PROYEK ====
Belum ada proyek.

Tekan Enter untuk melanjutkan...|
```

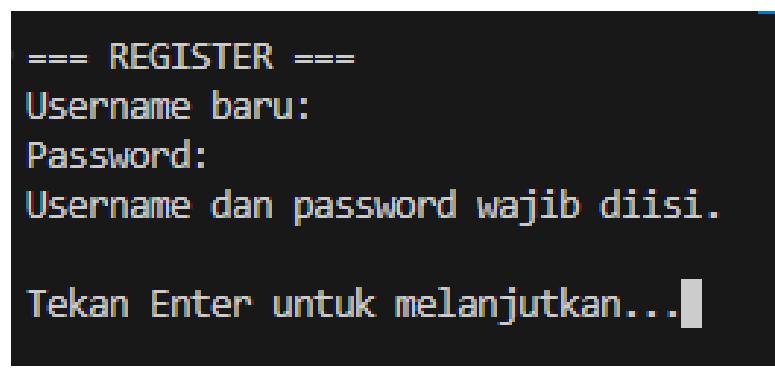
Gambar 4.11 Fitur Delete/Menghapus Proyek (Tidak Ada Proyek)



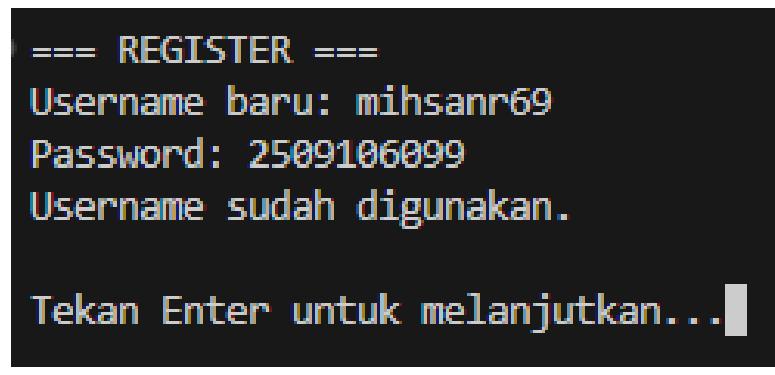
Gambar 4.12 Fitur Menu User Biasa



Gambar 4.13 Fitur Register (Berhasil)



Gambar 4.14 Fitur Register (Gagal/Kosong)



Gambar 4.15 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