

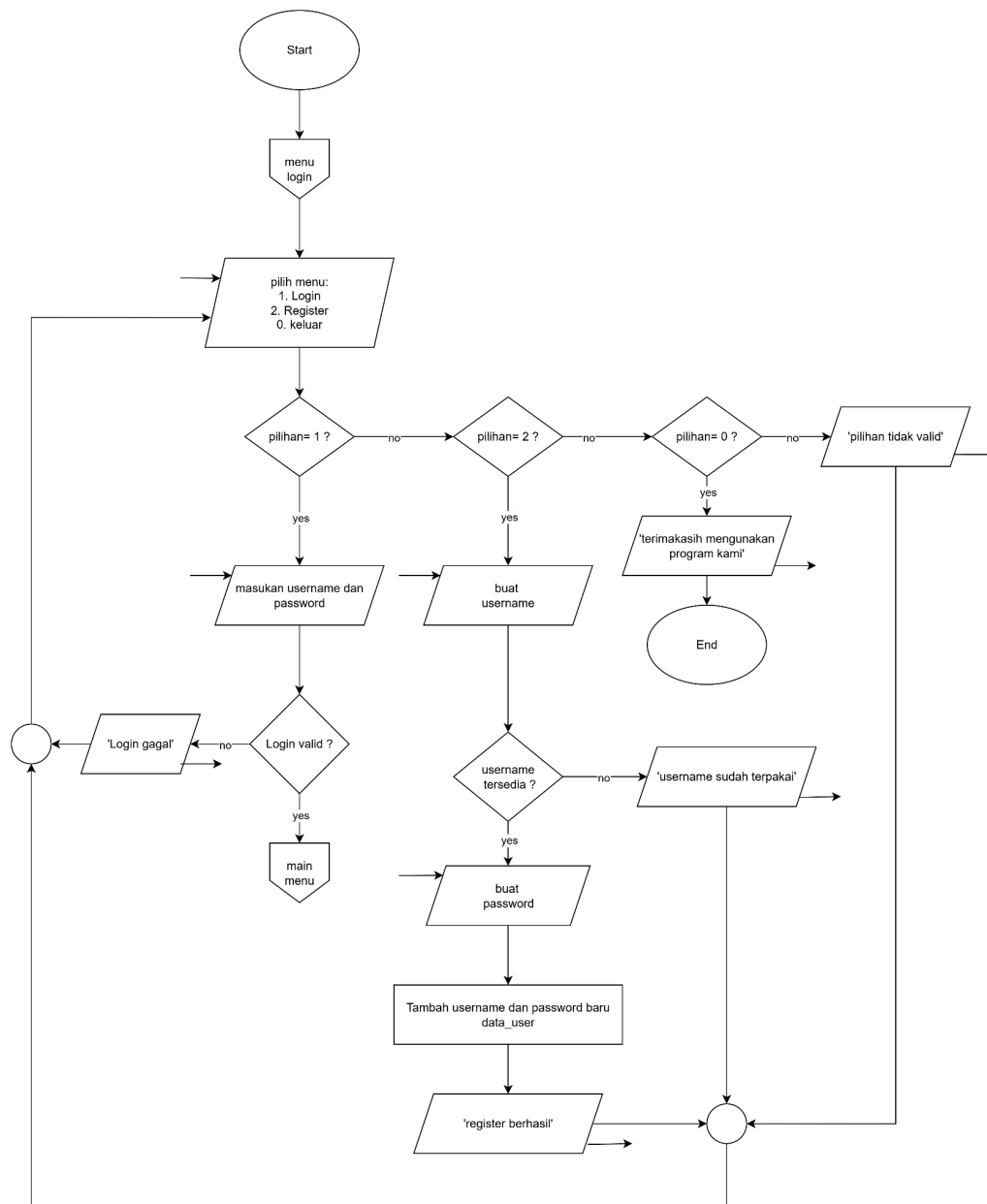
LAPORAN PRAKTIKUM
POSTTEST 7
ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:
Muhammad Alfauzi Syahputra 2509106006
Kelas A1 '25

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

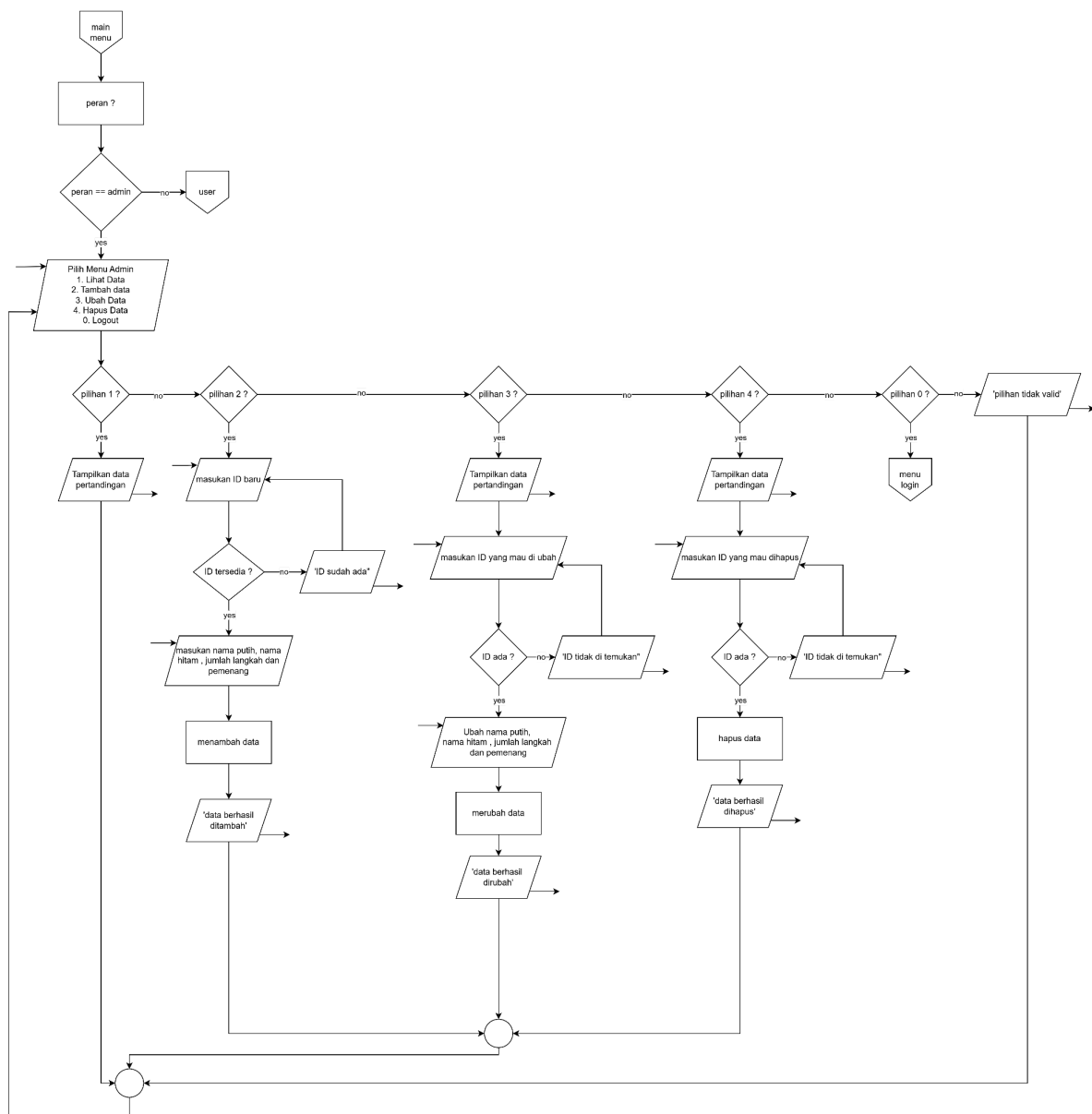
1. Flowchart



gambar 1.1 flowchart menu login dan register

Pada flowchart gambar 1.1, alur program dimulai dengan simbol Start, kemudian langsung ke proses OUTPUT di mana sistem menampilkan menu utama yang memberikan pilihan untuk Login, Register, atau Keluar. Setelah itu, sistem akan menunggu INPUT pilihan dari pengguna. Simbol Decision kemudian memeriksa pilihan tersebut. Jika pengguna memilih '1' (Login), alur berlanjut ke proses INPUT username dan password, yang kemudian divalidasi oleh Decision kedua. Jika valid, pengguna akan diarahkan ke menu sesuai perannya (admin atau user), namun jika tidak valid, program akan menampilkan pesan error. Jika pengguna memilih '2' (Register), alur akan mengarah ke proses registrasi sistem akan meminta INPUT username baru dan menggunakan Decision untuk memeriksa apakah

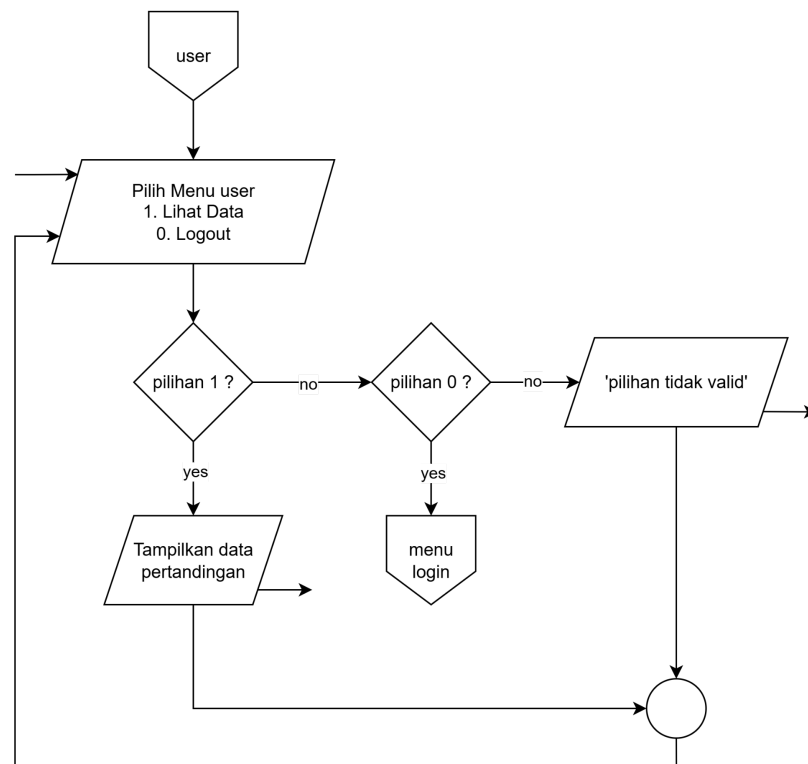
username tersebut sudah ada. Jika sudah ada, program akan menampilkan pesan error, tetapi jika tersedia, program akan meminta INPUT password baru dan menambahkan data pengguna tersebut ke sistem dengan peran default sebagai 'user'. Terakhir, jika pengguna memilih '0' (Keluar), maka program akan berakhir sesuai dengan alur yang ditandai oleh simbol End.



gambar 1.2 flowchart menu admin

Lanjutan dari flowchart sebelumnya, flowchart pada gambar 1.2 menggambarkan proses setelah admin berhasil login. Alur dimulai dengan program melakukan OUTPUT menu admin yang berisi opsi CRUD (Create, Read, Update, Delete) dan Logout. Setelah menerima INPUT pilihan menu dari admin, sebuah simbol Decision akan memeriksa pilihan tersebut dan mengarahkan alur ke proses yang sesuai: pilihan '1' akan menjalankan proses

Read untuk menampilkan semua data, '2' untuk proses Create data baru, '3' untuk proses Update data yang ada, dan '4' untuk proses Delete data. Jika admin memilih '0', alur akan mengarah ke proses Logout untuk kembali ke menu utama. Flowchart juga menunjukkan bahwa setiap proses CRUD melibatkan validasi data dan setelah selesai, alurnya akan kembali lagi untuk menampilkan menu admin, menciptakan sebuah siklus operasi hingga admin memilih untuk logout.



gambar 1.3 flowchart menu user

Pada flowchart gambar 1.3, alur untuk pengguna dengan peran 'user' ditampilkan. Setelah login berhasil, program akan melakukan OUTPUT menu user yang lebih sederhana, hanya berisi opsi untuk melihat data dan Logout. Setelah menerima INPUT dari pengguna, Decision akan memeriksa pilihan tersebut. Jika pengguna memilih '1', program akan menampilkan semua data pertandingan. Jika memilih '0', program akan mengakhiri sesi user dan kembali ke menu utama.

2. Deskripsi Singkat Program

Kode program ini dibuat untuk sistem manajemen "Pencatatan Hasil Pertandingan Catur". Program ini dirancang untuk melayani dua jenis pengguna, yaitu admin dan user biasa, dengan alur dan hak akses yang berbeda. Admin memiliki kontrol penuh untuk melakukan operasi Create, Read, Update, dan Delete (CRUD) terhadap data pertandingan. Sementara itu, user biasa hanya memiliki hak akses untuk melihat data yang ada (Read-only). Program ini juga dilengkapi dengan fitur registrasi untuk pengguna baru.

3. Source Code

A. Inisialisasi Data dan Tampilan Menu

Pada tahap inisialisasi, program mempersiapkan fondasi utamanya dengan memisahkan data inti dan definisi tampilan ke dalam file-file terpisah. File *data.py* berfungsi sebagai database sederhana yang menyimpan data awal pengguna di dalam dictionary *users* (termasuk *password* dan *role*) serta catatan hasil pertandingan catur di dalam dictionary *matches*. Sementara itu, file *menu_display.py* bertanggung jawab penuh untuk merancang semua antarmuka menu mulai dari *main_display*, *admin_display*, hingga *user_display* dengan menggunakan library *prettytable* agar tampilan menjadi rapi dan terstruktur. Dengan pemisahan ini, kode program utama menjadi lebih bersih dan dapat lebih fokus pada logika fungsionalitasnya.

Source Code *data.py* :

```
# variabel global
users = {
    'admin': {'password': '123', 'role': 'admin'},
    'alfauzi': {'password': '006', 'role': 'admin'},
    'putra': {'password': '000', 'role': 'user'}
}

# variabel global
matches = {
    1: {'putih': 'Magnus Carlsen', 'hitam': 'Hikaru Nakamura', 'langkah': 45,
        'pemenang': 'Putih Menang'},
    2: {'putih': 'Anish Giri', 'hitam': 'Fabiano Caruana', 'langkah': 60,
        'pemenang': 'Seri'},
    3: {'putih': 'Ian Nepomniachtchi', 'hitam': 'Ding Liren', 'langkah': 38,
        'pemenang': 'Hitam Menang'}
}
```

Source Code *menu_display.py* :

```
from prettytable import PrettyTable
```

```

# variabel global
main_display = PrettyTable()
main_display.field_names = ['Pilihan', 'Deskripsi']
main_display.align['Deskripsi'] = 'l'
main_display.add_row(['1', 'Login', ''])
main_display.add_row(['2', 'Register', ''])
main_display.add_row(['0', 'Keluar', ''])

# variabel global
admin_display = PrettyTable()
admin_display.field_names = ['Pilihan', 'Deskripsi']
admin_display.align['Deskripsi'] = 'l'
admin_display.add_row(['1', 'Lihat Data Pertandingan (Read)', ''])
admin_display.add_row(['2', 'Tambah Data Pertandingan (Create)', ''])
admin_display.add_row(['3', 'Ubah Data Pertandingan (Update)', ''])
admin_display.add_row(['4', 'Hapus Data Pertandingan (Delete)', ''])
admin_display.add_row(['0', 'Logout'])

# variabel global
user_display = PrettyTable()
user_display.field_names = ['Pilihan', 'Deskripsi']
user_display.align['Deskripsi'] = 'l'
user_display.add_row(['1', 'Lihat Data Pertandingan', ''])
user_display.add_row(['0', 'Logout', ''])

```

B. Fitur Main Menu

Prosedur *main* berfungsi sebagai 'otak' atau pengendali utama alur program, yang berjalan dalam sebuah *while True* agar menu terus ditampilkan setelah setiap operasi selesai. Program akan meminta input pilihan dari pengguna menggunakan fungsi *get_number* untuk memastikan input adalah angka, lalu menggunakan struktur *if-elif-else* untuk mengarahkan alur. Jika pengguna memilih 1 (Login), program akan memanggil fungsi *login()* dan menangkap tiga nilai yang dikembalikannya: sukses, *username*, dan *peran*. Logika utamanya adalah memeriksa status sukses, dan jika *True*, program akan memanggil prosedur *menu_admin* atau *menu_user* berdasarkan *peran* yang diterima. Jika pengguna memilih 2 (Register), program juga akan menangkap *nama_baru* dan *data_baru* yang dikembalikan, lalu prosedur *main* sendiri yang bertanggung jawab untuk menambahkan data tersebut ke dalam variabel global *users*. Terakhir, jika pengguna memilih 0 (Keluar), loop dihentikan dengan perintah *break*.

Source Code:

```

from colorama import Fore, Style, init
from menu_display import main_display
from tools import clear
from auth import login, register

```

```

from menu import menu_admin, menu_user
from tools import get_number
from data import users

init(autoreset=True)

# prosedur
def main():
    while True:
        clear()
        print('PENCATATAN HASIL LAGA CATUR'.center(50))
        print(Fore.YELLOW + str(main_display))
        pilihan = get_number('Masukkan pilihan Anda: ')

        if pilihan == 1:
            sukses, username, peran = login()
            if sukses:
                if peran == 'admin':
                    menu_admin(username)
                else:
                    menu_user(username)
            elif pilihan == 2:
                nama_baru, data_baru = register()
                if nama_baru is not None:
                    users[nama_baru] = data_baru
                    print(Fore.GREEN + '\nBerhasil register! Silakan login.')
                    input(Fore.CYAN + 'Tekan Enter untuk kembali...' +
Style.RESET_ALL)
                elif pilihan == 0:
                    print(Fore.GREEN + '\nTerima kasih telah menggunakan program kami!')
                    break
                else:
                    print(Fore.RED + 'Pilihan tidak valid, silakan pilih 1, 2, atau 0.')
                    input(Fore.CYAN + 'Tekan Enter untuk mencoba lagi...' +
Style.RESET_ALL)

if __name__ == "__main__":
    main()

```

C. Fitur Menu Login dan Register

Fungsi *login* dan *register* dalam file *auth.py* dirancang sebagai fungsi sejati yang bertanggung jawab penuh untuk mengumpulkan dan memvalidasi data tanpa mengubah status program secara langsung. Dalam fungsi *login*, prosesnya adalah meminta *username* dan *password* menggunakan fungsi *get_input*, kemudian memeriksa kecocokannya dengan data di dalam dictionary *users*. Bagian krusialnya adalah penggunaan *return* untuk melaporkan hasilnya: jika berhasil, fungsi akan mengembalikan *True* beserta *username* dan *peran*; jika gagal, ia akan mengembalikan *False*. Serupa dengan itu, fungsi *register* bertugas

mengumpulkan *new_username* dan *new_password*. Setelah melakukan validasi keunikan username, ia tidak langsung menyimpan data tersebut, melainkan mengembalikannya sebagai *new_username* dan *data_baru* untuk diproses lebih lanjut oleh pemanggilnya.

Source Code:

```
from colorama import Fore, Style
from data import users
from tools import get_input, clear

# fungsi tanpa paremeter
def login():
    clear()
    print(Fore.YELLOW + (f'{'='*18}{'HALAMAN LOGIN'}{'='*18}'))

    username = get_input('Masukkan Username: ')
    password = get_input('Masukkan Password: ')

    if username in users and users[username]['password'] == password:
        print(Fore.GREEN + (f'\n{'-'*18}{'Login berhasil'}{'-'*18}'))
        input(Fore.CYAN + 'Tekan Enter untuk melanjutkan...' + Style.RESET_ALL)

        peran = users[username]['role']
        return True, username, peran
    else:
        print(Fore.RED + (f'\n{'-'*19}{'Login gagal'}{'-'*19}'))
        input(Fore.CYAN + 'Tekan Enter untuk kembali...' + Style.RESET_ALL)
        return False, None, None

# fungsi tanpa parameter
def register():
    while True:
        clear()
        print(Fore.YELLOW + f'\n{'='*18}{'HALAMAN REGISTER'}{'='*18}')

        new_username = get_input('Username Baru (ketik "x" untuk kembali): ')
        if new_username.lower() == 'x':
            return None, None

        if new_username in users:
            print(Fore.RED + '\nUsername sudah digunakan.')
            input(Fore.CYAN + 'Tekan Enter untuk coba lagi...' +
Style.RESET_ALL)
        else:
            new_password = get_input('Password Baru: ')
            data_baru = {'password': new_password, 'role': 'user'}
            return new_username, data_baru
```


D. Fitur Menu Admin dan Menu User

Prosedur *menu_admin* dan *menu_user* dalam file *menu.py* berperan sebagai eksekutor atau pengontrol alur kerja setelah pengguna berhasil login. Keduanya berjalan dalam loop *while True* untuk terus menampilkan menu yang sesuai. Logika utamanya adalah memisahkan antara pengumpulan data dan eksekusi. Ketika admin memilih '2' (Tambah Data), prosedur ini akan memanggil fungsi *get_data_for_create()* dari modul *crud*. Setelah menerima *id_baru* dan *data_baru* yang dikembalikan, prosedur *menu_admin* inilah yang kemudian melakukan modifikasi pada variabel global *matches*. Pola yang sama diterapkan untuk operasi Update dan Delete, di mana *menu_admin* memanggil fungsi *get_* yang sesuai, menerima hasilnya, dan kemudian mengeksekusi perubahan (*matches[id_update] = data_update* atau *del matches[id_hapus]*).

Source Code:

```
from colorama import Fore, Style
from menu_display import admin_display, user_display
from tools import get_input, get_number, clear
from crud import get_data_for_read, get_data_for_create, get_data_for_update,
get_id_for_delete
from data import matches

# prosedur
def menu_admin(nama):
    while True:
        clear()
        print(f'Selamat Datang, {nama}'.center(50))
        print(f'MENU ADMIN'.center(50))
        print(Fore.YELLOW + str(admin_display))

        pilih = get_number('Pilih menu: ')

        if pilih == 1:
            tabel_data = get_data_for_read()
            if tabel_data:
                print(tabel_data)
                input(Fore.CYAN + '\nTekan Enter untuk kembali...' +
Style.RESET_ALL)
            else:
                print(Fore.RED + '\nBelum ada data pertandingan.')
                input(Fore.CYAN + '\nTekan Enter untuk kembali...' +
Style.RESET_ALL)

        elif pilih == 2:
            id_baru, data_baru = get_data_for_create()
            if id_baru is not None:
                matches[id_baru] = data_baru
```

```

        print(Fore.GREEN + '\nData berhasil ditambahkan!')
        input(Fore.CYAN + 'Tekan Enter untuk kembali...' +
Style.RESET_ALL)

    elif pilih == 3:
        id_update, data_update = get_data_for_update()
        if id_update is not None:
            matches[id_update] = data_update
            print(Fore.GREEN + '\nData berhasil diubah!')
            input(Fore.CYAN + 'Tekan Enter untuk kembali...' +
Style.RESET_ALL)

    elif pilih == 4:
        id_hapus = get_id_for_delete()
        if id_hapus is not None:
            del matches[id_hapus]
            print(Fore.GREEN + f'\nData ID {id_hapus} berhasil dihapus!')
            input(Fore.CYAN + 'Tekan Enter untuk kembali...' +
Style.RESET_ALL)

    elif pilih == 0:
        print(Fore.YELLOW + 'Anda telah keluar.')
        input(Fore.CYAN + 'Tekan Enter untuk kembali...' + Style.RESET_ALL)
        break
    else:
        print(Fore.RED + 'Pilihan tidak valid.')
        input(Fore.CYAN + 'Tekan Enter untuk mencoba lagi...' +
Style.RESET_ALL)

# prosedur
def menu_user(nama):
    while True:
        clear()
        print(f'Selamat Datang, {nama}'.center(50))
        print(f'MENU PENGGUNA'.center(50))
        print(Fore.YELLOW + str(user_display))

        pilih = get_number('Pilih menu: ')

        if pilih == 1:
            clear()
            tabel_data = get_data_for_read()
            if tabel_data:
                print(tabel_data)
                input(Fore.CYAN + '\nTekan Enter untuk kembali...' +
Style.RESET_ALL)
            else:
                print(Fore.RED + '\nBelum ada data pertandingan.')
                input(Fore.CYAN + '\nTekan Enter untuk kembali...' +
Style.RESET_ALL)

```

```

elif pilih == 0:
    print(Fore.YELLOW + 'Anda telah keluar.')
    input(Fore.CYAN + 'Tekan Enter untuk kembali...' + Style.RESET_ALL)
    break
else:
    print(Fore.RED + 'Pilihan tidak valid.')
    input(Fore.CYAN + 'Tekan Enter untuk mencoba lagi...' +
Style.RESET_ALL)

```

E. Fitur CRUD

Semua fungsi dalam modul *crud.py* kini dirancang sebagai fungsi sejati yang bertanggung jawab murni untuk mengumpulkan informasi dari pengguna tanpa mengubah data global. Fungsi *get_data_for_read()* bertugas menyusun data pertandingan ke dalam objek *PrettyTable* dan mengembalikannya. Fungsi *get_data_for_create()* dirancang untuk meminta semua detail pertandingan baru, melakukan validasi keunikan ID, lalu mengembalikan ID dan sebuah *dictionary* berisi data baru. Fungsi *get_data_for_update* dengan parameter *id_str* bekerja dengan terlebih dahulu menampilkan data, lalu meminta input ID dan informasi baru, kemudian mengembalikan ID beserta data yang telah diperbarui. Terakhir, fungsi *get_id_for_delete* hanya memiliki satu tugas, yaitu meminta ID yang valid dari admin dan mengembalikannya untuk dieksekusi oleh prosedur lain.

Source Code:

```

from prettytable import PrettyTable
from colorama import Fore, Style
from data import matches
from tools import get_input, get_number, clear

# READ
# fungsi tanpa parameter
def get_data_for_read():
    clear()
    print(Fore.YELLOW + 'DATA HASIL PERTANDINGAN'.center(70))
    if not matches:
        return None
    tabel = PrettyTable(['ID', 'Pemain Putih', 'Pemain Hitam', 'Langkah',
    'Pemenang'])
    for id, detail in matches.items():
        tabel.add_row([id, detail['putih'], detail['hitam'], detail['langkah'],
        detail['pemenang']])
    return tabel

# CREATE
# fungsi tanpa parameter
def get_data_for_create():

```

```

while True:
    try:
        clear()
        print(Fore.YELLOW + (f'\n{'='*10}{'TAMBAH DATA
PERTANDINGAN'}{'='*10}'))
        id_str = get_input('ID pertandingan (ketik "x" untuk kembali): ')
        if id_str.lower() == 'x': return None, None

        id = int(id_str)
        if id in matches:
            print(Fore.RED + f'\nID {id} sudah digunakan.')
            input(Fore.CYAN + 'Tekan Enter...')
        else:
            putih = get_input('Nama Pemain Putih: ')
            hitam = get_input('Nama Pemain Hitam: ')
            langkah = get_number('Jumlah Langkah: ')
            while True:
                print('\nPilihan Hasil: 1. Putih Menang, 2. Hitam Menang, 3.
Seri')

                pilih_hasil = get_number('Pilihan (1/2/3): ')
                if pilih_hasil in [1, 2, 3]:
                    hasil = 'Putih Menang' if pilih_hasil == 1 else 'Hitam
Menang' if pilih_hasil == 2 else 'Seri'
                    break
                else: print(Fore.RED + 'Pilihan tidak valid.')

            data_baru = {'putih': putih, 'hitam': hitam, 'langkah': langkah,
'pemenang': hasil}
            return id, data_baru
        except ValueError:
            print(Fore.RED + '\nID harus berupa angka!')
            input(Fore.CYAN + 'Tekan Enter...')

# UPDATE
# fungsi tanpa parameter
def get_data_for_update():
    read_table = get_data_for_read()
    if read_table:
        clear()
        print(Fore.YELLOW + 'UBAH DATA PERTANDINGAN'.center(70))
        print(read_table)
    else:
        print(Fore.RED + '\nBelum ada data pertandingan.')
        input(Fore.CYAN + '\nTekan Enter untuk kembali...' + Style.RESET_ALL)
        return None ,None

    while True:
        try:
            id_str = get_input('\nID pertandingan yang akan diubah (ketik "x"
untuk kembali): ')

```

```

        if id_str.lower() == 'x': return None, None

    id_laga = int(id_str)
    if id_laga in matches:
        print(Fore.CYAN + f'\nMengubah data ID {id_laga}')
        putih = get_input('Nama Pemain Putih Baru: ')
        hitam = get_input('Nama Pemain Hitam Baru: ')
        langkah = get_number('Jumlah Langkah Baru: ')
        while True:
            print('\nPilihan Hasil Baru: 1. Putih Menang, 2. Hitam
Menang, 3. Seri')
            pilih_hasil = get_number('Pilihan (1/2/3): ')
            if pilih_hasil in [1, 2, 3]:
                hasil = 'Putih Menang' if pilih_hasil == '1' else 'Hitam
Menang' if pilih_hasil == '2' else 'Seri'
                break
            else: print(Fore.RED + 'Pilihan tidak valid.')

            data_update = {'putih': putih, 'hitam': hitam, 'langkah':
langkah, 'pemenang': hasil}
            return id_laga, data_update
        else:
            print(Fore.RED + f'\nID {id_laga} tidak ditemukan.')
    except ValueError:
        print(Fore.RED + 'ID harus berupa angka!')

# DELETE
# fungsi tanpa parameter
def get_id_for_delete():
    read_table = get_data_for_read()
    if read_table:
        clear()
        print(Fore.YELLOW + 'HAPUS DATA PERTANDINGAN'.center(70))
        print(read_table)
    else:
        print(Fore.RED + '\nBelum ada data pertandingan.')
        input(Fore.CYAN + '\nTekan Enter untuk kembali...' + Style.RESET_ALL)
        return None

    while True:
        try:
            id_str = get_input('\nID pertandingan yang akan dihapus (ketik "x"
untuk kembali): ')
            if id_str.lower() == 'x': return None

            id = int(id_str)
            if id in matches:
                return id
            else:
                print(Fore.RED + 'ID tidak ditemukan.')

```

```
except ValueError:
    print(Fore.RED + 'ID harus berupa angka!')
```

F. Fitur Tools

Modul *tools.py* berisi kumpulan alat bantu (utilitas) yang bersifat umum dan dapat digunakan di seluruh bagian program. *clear()* adalah sebuah prosedur sederhana yang bertugas membersihkan layar terminal. Dua fungsi utamanya, *get_input* dan *get_number*, dirancang sebagai fungsi rekursif dengan parameter. Logika rekursifnya terletak pada cara fungsi ini memanggil dirinya sendiri (*return get_input(prompt)*) jika input yang diberikan tidak valid (kosong). Proses ini akan terus berulang hingga pengguna memberikan input yang valid, yang kemudian akan menjadi nilai return dan menghentikan rekursi. Fungsi *get_number* menambahkan lapisan error handling dengan *try-except* untuk memastikan bahwa input yang dikembalikan tidak hanya valid tetapi juga berupa angka.

Source Code:

```
import os
from colorama import Fore

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

# fungsi dengan parameter dan rekursif
def get_input(prompt):
    masukan = input(prompt)
    if masukan.strip():
        return masukan
    else:
        print(Fore.RED + "Input tidak boleh kosong.")
        return get_input(prompt)

# fungsi dengan parameter dan rekursif
def get_number(prompt):
    masukan = get_input(prompt)
    try:
        return int(masukan)
    except ValueError:
        print(Fore.RED + "Input harus berupa angka.")
        return get_number(prompt)
```

4. Hasil Output

```
PENCATATAN HASIL LAGA CATUR
+-----+-----+
| Pilihan | Deskripsi |
+-----+-----+
| 1       | Login    |
| 2       | Register |
| 0       | Keluar   |
+-----+-----+
Masukkan pilihan Anda:
Input tidak boleh kosong.
Masukkan pilihan Anda: halo
Input harus berupa angka.
Masukkan pilihan Anda: 1
```

gambar 4.1 terminal menu utama dan pengecekan input

```
=====HALAMAN LOGIN=====
Masukkan Username:
Input tidak boleh kosong.
Masukkan Username: alfauzi
Masukkan Password:
Input tidak boleh kosong.
Masukkan Password: 006

-----Login berhasil-----
Tekan Enter untuk melanjutkan...
```

gambar 4.2 terminal menu login dan pengecekan input

```
Selamat Datang, alfauzi
MENU ADMIN
+-----+-----+
| Pilihan | Deskripsi |
+-----+-----+
| 1       | Lihat Data Pertandingan (Read) |
| 2       | Tambah Data Pertandingan (Create) |
| 3       | Ubah Data Pertandingan (Update) |
| 4       | Hapus Data Pertandingan (Delete) |
| 0       | Logout   |
+-----+-----+
Pilih menu:
Input tidak boleh kosong.
Pilih menu: halo
Input harus berupa angka.
Pilih menu: 5
Pilihan tidak valid.
Tekan Enter untuk mencoba lagi...
```

gambar 4.3 terminal menu admin

DATA HASIL PERTANDINGAN				
ID	Pemain Putih	Pemain Hitam	Jumlah Langkah	Pemenang
1	Magnus Carlsen	Hikaru Nakamura	45	Putih Menang
2	Anish Giri	Fabiano Caruana	60	Seri
3	Ian Nepomniachtchi	Ding Liren	38	Hitam Menang

Tekan Enter untuk kembali...

gambar 4.4 terminal menu admin READ

```

=====TAMBAH DATA PERTANDINGAN=====
ID pertandingan (ketik "x" untuk kembali):
Input tidak boleh kosong.
ID pertandingan (ketik "x" untuk kembali): halo

ID harus berupa angka!
Tekan Enter...

```

gambar 4.5 terminal menu admin CREATE ID harus angka dan input tidak boleh kosong

```

=====TAMBAH DATA PERTANDINGAN=====
ID pertandingan (ketik "x" untuk kembali): 1

ID 1 sudah digunakan.
Tekan Enter...

```

gambar 4.6 terminal menu admin CREATE id sudah ada


```
=====TAMBAH DATA PERTANDINGAN=====
ID pertandingan (ketik "x" untuk kembali): 4
Nama Pemain Putih:
Input tidak boleh kosong.
Nama Pemain Putih: Paul Morphy
Nama Pemain Hitam:
Input tidak boleh kosong.
Nama Pemain Hitam: Tigran Petrosian
Jumlah Langkah:
Input tidak boleh kosong.
Jumlah Langkah: halo
Input harus berupa angka.
Jumlah Langkah: 53

Pilihan Hasil: 1. Putih Menang, 2. Hitam Menang, 3. Seri
Pilihan (1/2/3):
Input tidak boleh kosong.
Pilihan (1/2/3): halo
Input harus berupa angka.
Pilihan (1/2/3): 4
Pilihan tidak valid.

Pilihan Hasil: 1. Putih Menang, 2. Hitam Menang, 3. Seri
Pilihan (1/2/3): 3

Data berhasil ditambahkan!
Tekan Enter untuk kembali...█
```

gambar 4.7 terminal menu admin CREATE pengecekan input

```
UBAH DATA PERTANDINGAN
+-----+-----+-----+-----+
| ID | Pemain Putih | Pemain Hitam | Langkah | Pemenang |
+-----+-----+-----+-----+
| 1 | Magnus Carlsen | Hikaru Nakamura | 45 | Putih Menang |
| 2 | Anish Giri | Fabiano Caruana | 60 | Seri |
| 3 | Ian Nepomniachtchi | Ding Liren | 38 | Hitam Menang |
| 4 | Paul Morphy | Tigran Petrosian | 53 | Seri |
+-----+-----+-----+-----+

ID pertandingan yang akan diubah (ketik "x" untuk kembali):
Input tidak boleh kosong.

ID pertandingan yang akan diubah (ketik "x" untuk kembali): halo
ID harus berupa angka!

ID pertandingan yang akan diubah (ketik "x" untuk kembali): 5

ID 5 tidak ditemukan.

ID pertandingan yang akan diubah (ketik "x" untuk kembali): 4

Mengubah data ID 4
Nama Pemain Putih Baru:
Input tidak boleh kosong.
Nama Pemain Putih Baru: Tigran Petrosian
Nama Pemain Hitam Baru:
Input tidak boleh kosong.
Nama Pemain Hitam Baru: Paul Morphy
Jumlah Langkah Baru:
Input tidak boleh kosong.
Jumlah Langkah Baru: halo
Input harus berupa angka.
Jumlah Langkah Baru: 58

Pilihan Hasil Baru: 1. Putih Menang, 2. Hitam Menang, 3. Seri
Pilihan (1/2/3):
Input tidak boleh kosong.
Pilihan (1/2/3): halo
Input harus berupa angka.
Pilihan (1/2/3): 4
Pilihan tidak valid.

Pilihan Hasil Baru: 1. Putih Menang, 2. Hitam Menang, 3. Seri
Pilihan (1/2/3): 1

Data berhasil diubah!
Tekan Enter untuk kembali... 
```

gambar 4.8 terminal menu admin CREATE pengecekan input

```

HAPUS DATA PERTANDINGAN
+-----+-----+-----+-----+
| ID | Pemain Putih | Pemain Hitam | Langkah | Pemenang |
+-----+-----+-----+-----+
| 1 | Magnus Carlsen | Hikaru Nakamura | 45 | Putih Menang |
| 2 | Anish Giri | Fabiano Caruana | 60 | Seri |
| 3 | Ian Nepomniachtchi | Ding Liren | 38 | Hitam Menang |
| 4 | Tigran Petrosian | Paul Morphy | 58 | Seri |
| 999 | muheheheh | mahohohoh | 69 | Seri |
+-----+-----+-----+-----+

ID pertandingan yang akan dihapus (ketik "x" untuk kembali):
Input tidak boleh kosong.

ID pertandingan yang akan dihapus (ketik "x" untuk kembali): halo
ID harus berupa angka!

ID pertandingan yang akan dihapus (ketik "x" untuk kembali): 5
ID tidak ditemukan.

ID pertandingan yang akan dihapus (ketik "x" untuk kembali): 999

Data ID 999 berhasil dihapus!
Tekan Enter untuk kembali...

```

gambar 4.9 terminal menu admin DELETE pengecekan input ID999 berhasil dihapus

```

DATA HASIL PERTANDINGAN
+-----+-----+-----+-----+
| ID | Pemain Putih | Pemain Hitam | Langkah | Pemenang |
+-----+-----+-----+-----+
| 1 | Magnus Carlsen | Hikaru Nakamura | 45 | Putih Menang |
| 2 | Anish Giri | Fabiano Caruana | 60 | Seri |
| 3 | Ian Nepomniachtchi | Ding Liren | 38 | Hitam Menang |
| 4 | Tigran Petrosian | Paul Morphy | 58 | Seri |
+-----+-----+-----+-----+

Tekan Enter untuk kembali...

```

gambar 4.10 terminal menu admin READ setelah CURD

```

+-----+-----+
| Pilihan | Deskripsi |
+-----+-----+
| 1 | Lihat Data Pertandingan (Read) |
| 2 | Tambah Data Pertandingan (Create) |
| 3 | Ubah Data Pertandingan (Update) |
| 4 | Hapus Data Pertandingan (Delete) |
| 0 | Logout |
+-----+-----+

Pilih menu: 0
Anda telah logout.
Tekan Enter untuk kembali...

```

gambar 4.11 terminal menu admin LOGOUT

```

=====HALAMAN LOGIN=====
Masukkan Username: muheheheh
Masukkan Password: 123

-----Login gagal-----
Tekan Enter untuk kembali...

```

gambar 4.12 terminal halaman login gagal

```

=====MENU REGISTER=====
Masukkan Username Baru (atau ketik "X" untuk kembali):
Username tidak boleh kosong.
Masukkan Username Baru (atau ketik "X" untuk kembali): alfauzi

Username sudah digunakan.
Tekan Enter untuk cobal lagi...

```

gambar 4.13 terminal menu register pengecekan input

```

=====MENU REGISTER=====
Masukkan Username Baru (atau ketik "X" untuk kembali): muheheheh
Masukkan Password Baru: 123

Registrasi berhasil! Silakan login.
Tekan Enter untuk kembali...

```

gambar 4.14 terminal menu register dan berhasil register

```

=====HALAMAN LOGIN=====
Masukkan Username: muheheheh
Masukkan Password: 123

-----Login berhasil-----
Tekan Enter untuk melanjutkan...

```

gambar 4.15 terminal halaman login dan berhasil

```

                Selamat Datang, muheheheh
                MENU USER

+-----+-----+
| Pilihan | Deskripsi |
+-----+-----+
| 1       | Lihat Data Pertandingan |
| 0       | Logout    |
+-----+-----+
Pilih menu:

```

gambar 4.16 terminal menu user

DATA HASIL PERTANDINGAN				
ID	Pemain Putih	Pemain Hitam	Langkah	Pemenang
1	Magnus Carlsen	Hikaru Nakamura	45	Putih Menang
2	Anish Giri	Fabiano Caruana	60	Seri
3	Ian Nepomniachtchi	Ding Liren	38	Hitam Menang
4	Tigran Petrosian	Paul Morphy	58	Seri

Tekan Enter untuk kembali...

gambar 4.17 terminal menu user READ

Selamat Datang, muheheheh	
MENU USER	
Pilihan	Deskripsi
1	Lihat Data Pertandingan
0	Logout

Pilih menu: 0
Anda telah logout.
Tekan Enter untuk kembali...

gambar 4.18 terminal menu LOGOUT

PENCATATAN HASIL PERTANDINGAN CATUR	
Pilihan	Deskripsi
1	Login
2	Register
0	Keluar

Masukkan pilihan Anda: 0
Terima kasih telah menggunakan program kami!

gambar 4.19 terminal menu keluar

DATA HASIL PERTANDINGAN	
Belum ada data pertandingan.	

Tekan Enter untuk kembali...

gambar 4.20 terminal READ belum ada data pertandingan

5. Langkah-langkah GIT

5.1 GIT Add

git add berfungsi untuk menambahkan file ke staging area, menandai file mana saja yang akan disimpan pada commit berikutnya.

```
PS D:\Kuliah\praktikum\praktikum-apd> git add .
PS D:\Kuliah\praktikum\praktikum-apd> git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   kelas/tes.py
    new file:   post-test/post-test-apd-7/2509106006-Muhammad-Alfauzi-Syahputra-PT-7.py
    new file:   post-test/post-test-apd-7/__pycache__/auth.cpython-313.pyc
    new file:   post-test/post-test-apd-7/__pycache__/crud.cpython-313.pyc
    new file:   post-test/post-test-apd-7/__pycache__/data.cpython-313.pyc
    new file:   post-test/post-test-apd-7/__pycache__/menu.cpython-313.pyc
    new file:   post-test/post-test-apd-7/__pycache__/menu_display.cpython-313.pyc
    new file:   post-test/post-test-apd-7/__pycache__/tools.cpython-313.pyc
    new file:   post-test/post-test-apd-7/auth.py
    new file:   post-test/post-test-apd-7/crud.py
    new file:   post-test/post-test-apd-7/data.py
    new file:   post-test/post-test-apd-7/menu.py
    new file:   post-test/post-test-apd-7/menu_display.py
    new file:   post-test/post-test-apd-7/tools.py
```

gambar 5.1 menambahkan

5.2 GIT Commit

git commit digunakan untuk menyimpan perubahan yang sudah di-add ke repository lokal, biasanya disertai pesan singkat yang menjelaskan perubahan tersebut.

```
PS D:\Kuliah\praktikum\praktikum-apd> git commit -m 'posttets7'
[main 1503421] posttets7
14 files changed, 333 insertions(+)
create mode 100644 kelas/tes.py
create mode 100644 post-test/post-test-apd-7/2509106006-Muhammad-Alfauzi-Syahputra-PT-7.py
create mode 100644 post-test/post-test-apd-7/__pycache__/auth.cpython-313.pyc
create mode 100644 post-test/post-test-apd-7/__pycache__/crud.cpython-313.pyc
create mode 100644 post-test/post-test-apd-7/__pycache__/data.cpython-313.pyc
create mode 100644 post-test/post-test-apd-7/__pycache__/menu.cpython-313.pyc
create mode 100644 post-test/post-test-apd-7/__pycache__/menu_display.cpython-313.pyc
create mode 100644 post-test/post-test-apd-7/__pycache__/tools.cpython-313.pyc
create mode 100644 post-test/post-test-apd-7/auth.py
create mode 100644 post-test/post-test-apd-7/crud.py
create mode 100644 post-test/post-test-apd-7/data.py
create mode 100644 post-test/post-test-apd-7/menu.py
create mode 100644 post-test/post-test-apd-7/menu_display.py
create mode 100644 post-test/post-test-apd-7/tools.py
```

gambar 5.2 menyimpan

5.3 GIT Push

git push digunakan untuk mengirim commit dari repository lokal ke repository remote di GitHub, agar perubahan bisa terlihat online dan diakses oleh orang lain.

```
PS D:\Kuliah\praktikum\praktikum-apd> git push origin main
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 16 threads
Compressing objects: 100% (20/20), done.
Writing objects: 100% (20/20), 11.93 KiB | 643.00 KiB/s, done.
Total 20 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:Alfauzi-S/praktikum-apd.git
c1c2558..1503421 main -> main
```

gambar 5.3 mendorong commit dari repository lokal ke repository remote