LAPORAN PRAKTIKUM

POSTTEST 6

ALGORITMA PEMROGRAMAN DASAR

Disusun oleh:

Muhammad Alfauzi Syahputra 2509106006

Kelas A1 ‘25

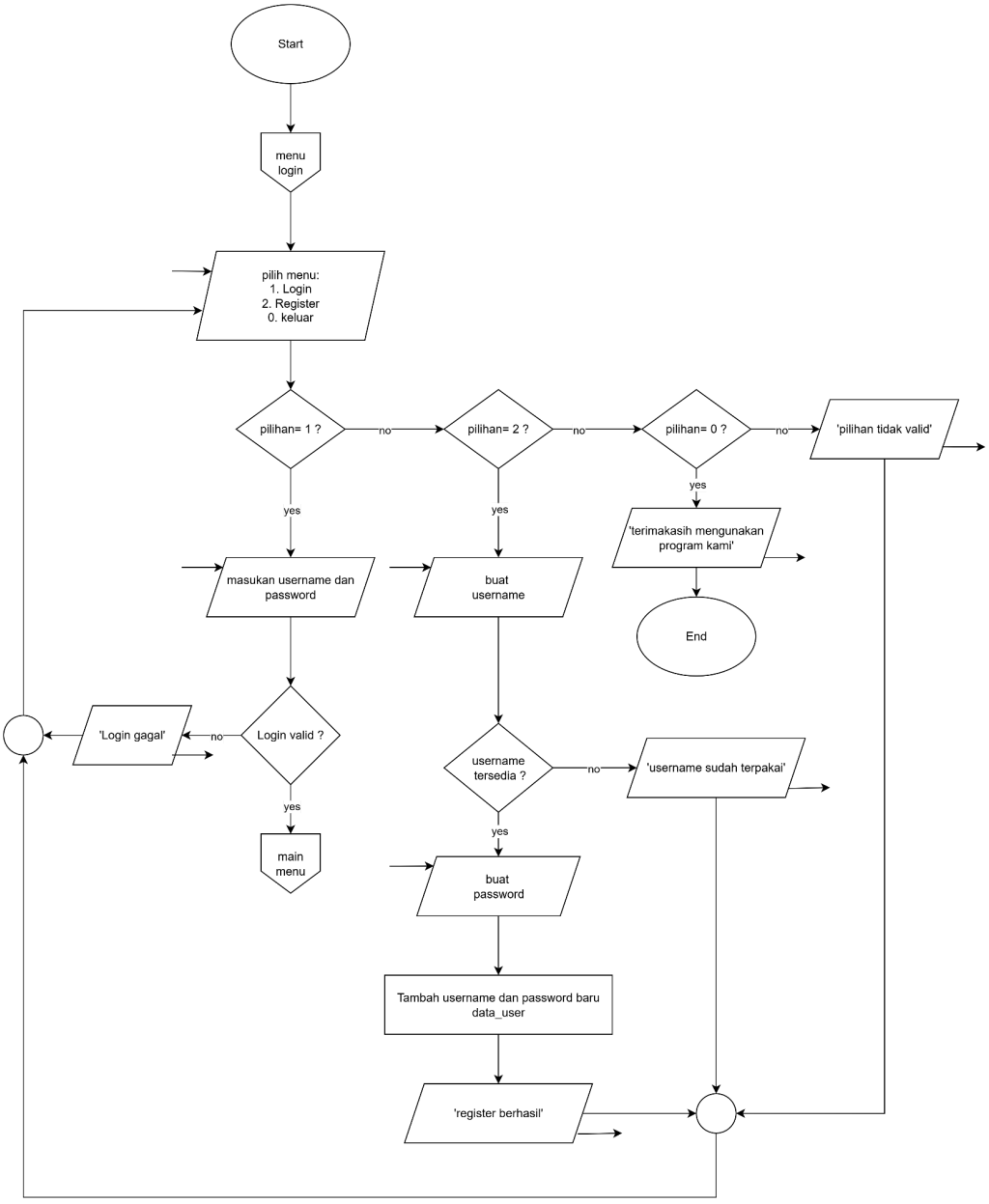
PROGRAM STUDI INFORMATIKA

UNIVERSITAS MULAWARMAN

SAMARINDA

2025

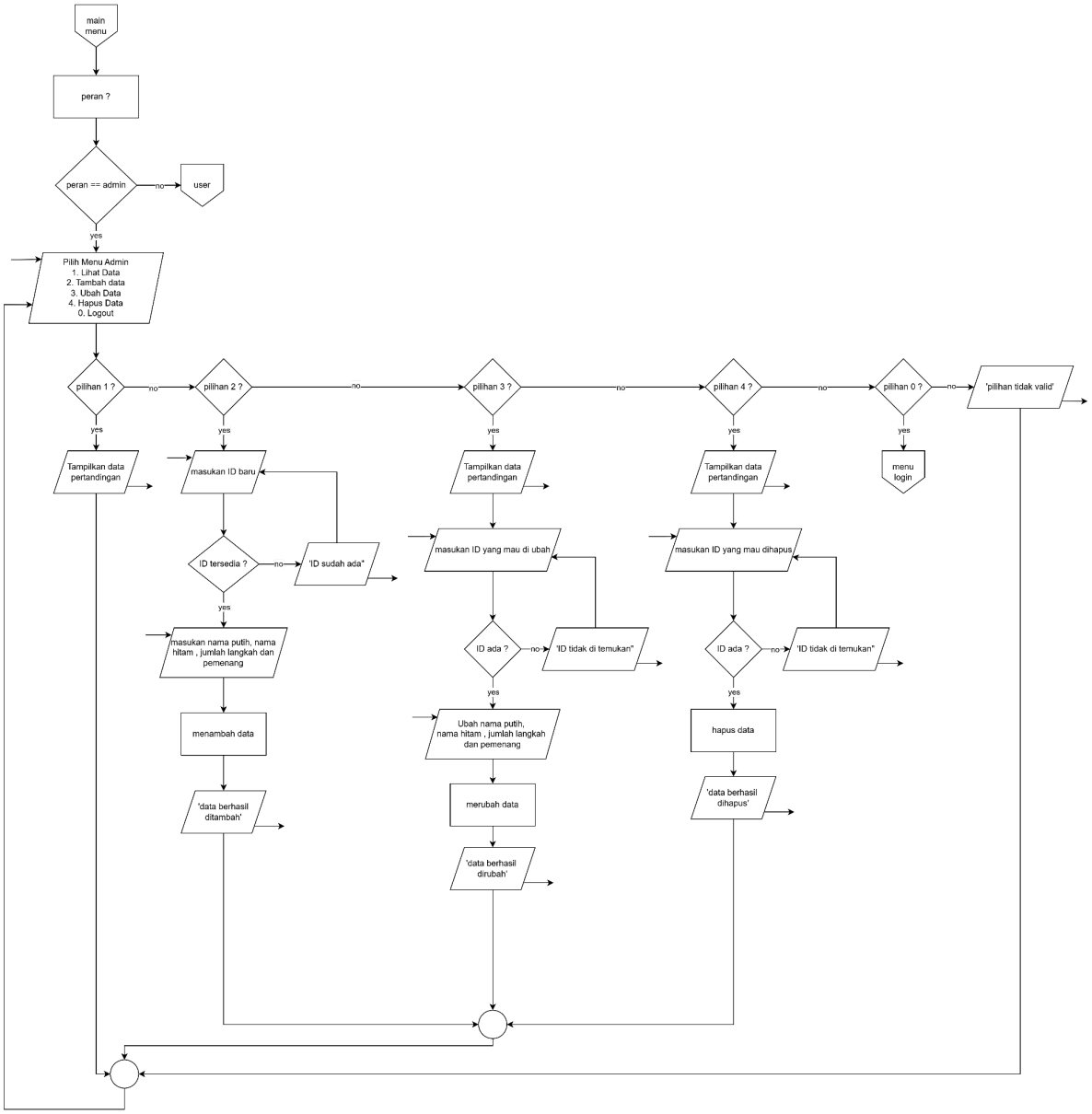
1. Flowchart



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

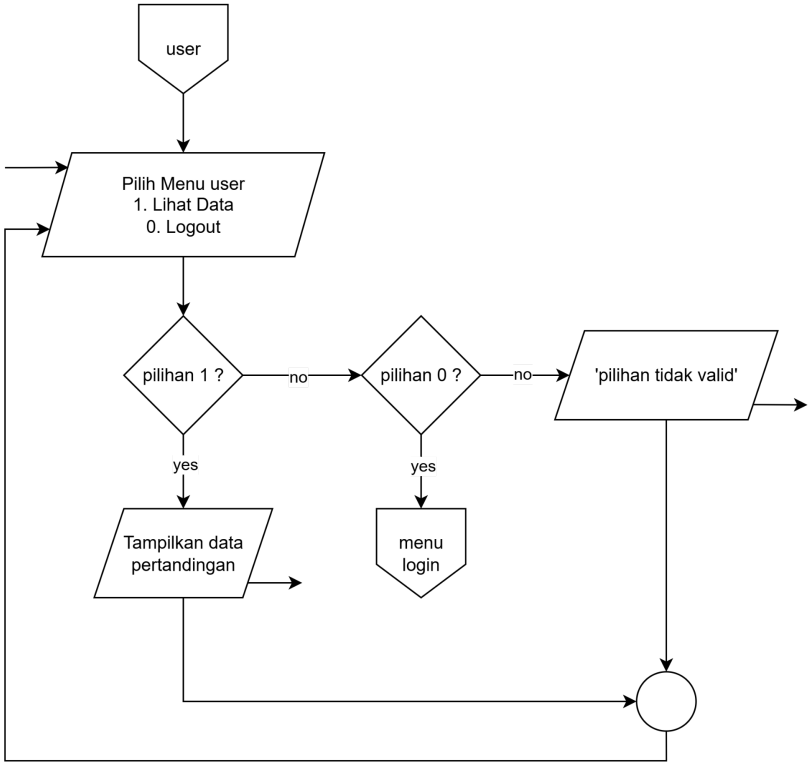
username tersebut sudah ada. Jika sudah ada, program akan menampilkan pesan error, tetapi jika tersedia, program akan meminta INPUT password baru dan menambahkan data pengguna tersebut ke sistem dengan peran default sebagai 'user'. Terakhir, jika pengguna memilih '0' (Keluar), maka program akan berakhir 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 mengimpor data dan tampilan dari file terpisah. File *data.py* berfungsi sebagai database sederhana yang menyimpan data awal pengguna (seperti *username, password*, dan *role*) serta catatan hasil pertandingan catur dalam struktur *nested dictionary*. Sementara itu, file *menu.py* menggunakan *library prettytable* untuk mendefinisikan dan merancang semua tampilan menu mulai dari menu utama, menu admin, hingga menu pengguna agar terlihat lebih rapi dan terstruktur. Dengan memisahkan data dan tampilan, kode program utama menjadi lebih bersih dan fokus pada logika fungsionalitasnya.

Source Code data,py :

```
data_user =[
    {'username': 'alfauzi', 'password': '006', 'role': 'admin'},
    {'username': 'putra', 'password': '123', 'role': 'user'},
    {'username': 'admin', 'password': 'admin123', 'role': 'admin'}
]

data_pertandingan = [
    {
        'id': 1,
        'pemain_putih': 'Magnus Carlsen',
        'pemain_hitam': 'Hikaru Nakamura',
        'jumlah_langkah': 45,
        'pemenang': 'Putih Menang',
    },
    {
        'id': 2,
        'pemain_putih': 'Anish Giri',
        'pemain_hitam': 'Fabiano Caruana',
        'jumlah_langkah': 60,
        'pemenang': 'Seri'
    },
    {
```

```
        'id': 3,
        'pemain_putih': 'Ian Nepomniachtchi',
        'pemain_hitam': 'Ding Liren',
        'jumlah_langkah': 38,
        'pemenang': 'Hitam Menang'
    }
]
```

Source Code menu,py :

```
from prettytable import PrettyTable

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

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

menu_user = PrettyTable()
menu_user.field_names = ['Pilihan', 'Deskripsi']
menu_user.align['Deskripsi'] = 'l'
menu_user.add_row(['1', 'Lihat Data Pertandingan             '])
menu_user.add_row(['0', 'Logout                              '])
```

B. Fitur Menu Login, Registrasi dan keluar

Fitur ini adalah gerbang utama program yang berjalan dalam sebuah *while True* agar menu terus ditampilkan setelah setiap operasi selesai. Program akan meminta INPUT pilihan dari pengguna dan menggunakan *if-elif-else* untuk mengontrol alur. Jika pengguna memilih '1' (Login), program akan meminta username dan password, lalu menjalankan mesin validasi. Logikanya adalah dengan melakukan perulangan pada list *data_user* untuk memeriksa setiap data satu per satu. Di dalam perulangan, sebuah if statement akan memvalidasi apakah *user['username']* (username) dan *user['password']* (password) cocok dengan input pengguna. Jika kedua kondisi ini terpenuhi, status *login_berhasil* diubah menjadi *True*, peran pengguna disimpan dari *user[2]*, dan perulangan dihentikan dengan *break*. Jika pengguna memilih '2' (Register), program juga melakukan iterasi pada *data_user* untuk memastikan username belum terdaftar sebelum menambahkannya ke list menggunakan metode *.append()*.

Terakhir, jika pengguna memilih '0' (Keluar), program akan menghentikan loop utama dengan perintah *break* dan program berakhir.

Source Code:

```
import os
from prettytable import PrettyTable
from colorama import Fore, Style, init
from data import data_pertandingan, data_user
from menu import menu_utama, menu_admin, menu_user
init(autoreset=True)

while True:
    os.system('cls || clear')
    print('PENCATATAN HASIL PERTANDINGAN CATUR')
    print(menu_utama)
    pilihan_awal = input('Masukkan pilihan Anda: ')
    login_berhasil = False

    if pilihan_awal == '1':
        os.system('cls || clear')
        print(Fore.YELLOW + ('HALAMAN LOGIN')
        input_username = ""
        while not input_username.strip():
            input_username = input('Masukkan Username: ')
            if not input_username.strip():
                print(Fore.RED + "Username tidak boleh kosong.")

        input_password = ''
        while not input_password.strip():
            input_password = input('Masukkan Password: ')
            if not input_password.strip():
                print(Fore.RED + "Password tidak boleh kosong.")

        for user in data_user:
            if user['username'] == input_username and user['password'] ==
input_password:
                peran_user = user['role']
                login_berhasil = True
                break

        if login_berhasil == True:
            print('Login berhasil')
            input('Tekan Enter untuk melanjutkan...' +

        else:
            print('Login gagal')
            input('Tekan Enter untuk kembali...')
```

```
    elif pilihan_awal == '2':
        while True:
            os.system('cls || clear')
            print('MENU REGISTER')
            username_baru = ''
            while not username_baru.strip():
                username_baru = input('Masukkan Username Baru (atau ketik "X"
untuk kembali): ')
                if not username_baru.strip():
                    print('Username tidak boleh kosong.')

            if username_baru == 'x':
                break

            username_sudah_ada = False
            for user in data_user:
                if user['username'] == username_baru:
                    username_sudah_ada = True
                    break
            if username_sudah_ada == True:
                print(Username sudah digunakan.')
                input('Tekan Enter untuk cobal lagi...')
            else:
                password_baru = ''
                while not password_baru.strip():
                    password_baru = input('Masukkan Password Baru: ')
                    if not password_baru.strip():
                        print('Password tidak boleh kosong.')
                data_user.append({'username' : username_baru, 'password'
:password_baru, 'role' : 'user'})
                print('Registrasi berhasil! Silakan login.')
                input('Tekan Enter untuk kembali...')
                break

    elif pilihan_awal == '0':
        print('Terima kasih telah menggunakan program kami!')
        break

    else:
        print('Pilihan tidak valid, silakan pilih 1, 2, atau 0.')
        input('Tekan Enter untuk mencoba lagi...')
```

C. Menu Admin READ

Fungsi "Menu Admin READ" dirancang untuk menampilkan semua data pertandingan catur dalam format tabel yang rapi. Prosesnya dimulai dengan membuat objek tabel kosong dari *PrettyTable* dan mendefinisikan header kolomnya. Bagian krusial dari logika ini ada pada blok kode yang Anda tunjukkan: pertama, *if data_pertandingan:* berfungsi sebagai pengecekan keamanan untuk memastikan bahwa program hanya akan

mencoba menampilkan data **jika list *data_pertandingan* tidak kosong**. Jika ada data, program akan menjalankan perulangan *for pertandingan in data_pertandingan:* yang akan mengiterasi setiap elemen *dictionary* di dalam list tersebut. Untuk setiap *pertandingan*, perintah *tabel.add_row([...])* dieksekusi untuk menambahkan baris baru ke dalam tabel, di mana nilai-nilai dari *dictionary* seperti *pertandingan['id']* dan *pertandingan['pemain_putih']* diambil menggunakan *key* masing-masing untuk mengisi sel-sel tabel secara berurutan. Setelah perulangan selesai memproses semua data, *print(tabel)* akan menampilkan keseluruhan tabel yang sudah terisi dan terformat dengan baik ke layar.

Source Code:

```
            while True:
                os.system('cls || clear')

                if peran_user == 'admin':
                    print(f'Selamat Datang, {input_username}')
                    print(f'MENU ADMIN')
                    print(menu_admin)
                    pilihan_menu = ''
                    while not pilihan_menu.strip():
                        pilihan_menu = input('Pilih menu: ')
                        if not pilihan_menu.strip():
                            print('Pilihan tidak boleh kosong. Silakan coba
lagi.')

                    if pilihan_menu == '1':
                        os.system('cls || clear')
                        print('DATA HASIL PERTANDINGAN')
                        tabel = PrettyTable()
                        tabel.field_names = ['ID', 'Pemain Putih', 'Pemain
Hitam', 'Jumlah Langkah', 'Pemenang']
                        if data_pertandingan:
                            for pertandingan in data_pertandingan:
                                tabel.add_row([
                                    pertandingan['id'],
                                    pertandingan['pemain_putih'],
                                    pertandingan['pemain_hitam'],
                                    pertandingan['jumlah_langkah'],
                                    pertandingan['pemenang']
                                ])
                            print(tabel)

                        else:
                            print('Belum ada data pertandingan.')
                        input('Tekan Enter untuk kembali...')
```

D. Menu Admin CREATE

Fungsi "Menu Admin CREATE" dirancang untuk menambahkan data pertandingan baru secara aman, dengan validasi berlapis untuk menjaga integritas data. Proses ini diatur dalam sebuah perulangan *while True*, yang memungkinkan admin untuk mengulang input jika terjadi kesalahan. Validasi pertama terjadi saat admin memasukkan ID, di mana program memeriksa apakah input tersebut merupakan angka menggunakan *.isdigit()*. Jika valid, program melanjutkan ke validasi kedua untuk memastikan keunikan ID dengan menginisialisasi variabel penanda *id_tersedia = True* dan melakukan perulangan pada *data_pertandingan* untuk mencari ID yang sama; jika ditemukan, penanda diubah menjadi False. Apabila setelah semua pengecekan nilai *id_tersedia* tetap *True,* program akan meminta input untuk detail pertandingan lainnya. Setelah semua data terkumpul, proses ini mencapai langkah finalnya, di mana *data_pertandingan.append({...})* dieksekusi. Perintah ini secara efisien "membungkus" semua informasi yang baru saja divalidasi ke dalam satu paket data terstruktur—sebuah *dictionary* baru—di mana setiap *key* (seperti *'id'* dan *'pemain_putih'*) dipasangkan dengan variabel yang sesuai (seperti *input_id* dan *nama_putih*). Dengan demikian, *dictionary* tersebut ditambahkan sebagai catatan baru ke dalam daftar utama pertandingan, dan perulangan dihentikan dengan *break*.

Source Code:

```
                    elif pilihan_menu == '2':
                        while True:
                            os.system('cls || clear')
                            print(('TAMBAH DATA PERTANDINGAN')
                            input_id_str = ''
                            while not input_id_str.strip():
                                input_id_str = input('Masukkan ID Pertandingan
(atau ketik "X" untuk kembali): ')
                                if not input_id_str.strip():
                                    print(Fore.RED + "ID tidak boleh kosong.
Silakan coba lagi.")

                            if input_id_str == 'x':
                                break
                            elif not input_id_str.isdigit():
                                print('ID harus berupa angka!')
                                input('Tekan Enter untuk mencoba lagi...')
                                continue

                            input_id = int(input_id_str)
                            id_tersedia = True
                            for i in data_pertandingan:
                                if i[0] == input_id:
                                    id_tersedia = False
                                    break
```

```
                        if id_tersedia == True:
                            nama_putih = ""
                            while not nama_putih.strip():
                                nama_putih = input('Nama Pemain Putih: ')
                                if not nama_putih.strip():
                                    print(Fore.RED + "Nama tidak boleh
kosong.")

                            nama_hitam = ''
                            while not nama_hitam.strip():
                                nama_hitam = input('Nama Pemain Hitam: ')
                                if not nama_hitam.strip():
                                    print(Fore.RED + "Nama tidak boleh
kosong.")

                            while True:
                                jumlah_langkah_str = input('Jumlah Langkah
Total: ')
                                if not jumlah_langkah_str.strip():
                                    print(Fore.RED + "Jumlah langkah tidak
boleh kosong.")
                                elif not jumlah_langkah_str.isdigit():
                                    print(Fore.RED + 'Jumlah langkah harus
berupa angka!')
                                else:
                                    jumlah_langkah = int(jumlah_langkah_str)
                                    break

                            while True:
                                print('Pilihan Hasil: 1. Putih Menang, 2.
Hitam Menang, 3. Seri')
                                pilihan_hasil = input('Pilihan (1/2/3): ')
                                if pilihan_hasil in ['1', '2', '3']:
                                    if pilihan_hasil == '1': hasil = 'Putih
Menang'
                                    elif pilihan_hasil == '2': hasil =
'Hitam Menang'
                                    else: hasil = 'Seri'
                                    break
                                else:
                                    print('Pilihan tidak valid.')
                            data_pertandingan.append({
                                'id': input_id,
                                'pemain_putih': nama_putih,
                                'pemain_hitam': nama_hitam,
                                'jumlah_langkah': jumlah_langkah,
                                'pemenang': hasil
                            })
```

```
                                print('Data berhasil ditambahkan!')
                                input('Tekan Enter untuk kembali...')
                                break
                            else:
                                print('ID {input_id}
 sudah digunakan!')
                                input('Tekan Enter untuk mencoba lagi...')
```

E. Menu Admin UPDATE

Fungsi ini bertugas untuk mengubah data yang sudah ada, yang diidentifikasi melalui ID unik. Prosesnya diawali dengan menampilkan seluruh data pertandingan agar admin dapat melihat ID mana yang akan diubah. Program kemudian memasuki perulangan *while True* untuk meminta input ID dan melakukan validasi, memastikan ID tersebut adalah angka dan benar-benar ada. Logika utamanya adalah menemukan indeks (posisi) data yang akan diubah dalam list *data_pertandingan*. Hal ini dilakukan dengan melakukan iterasi *for i in range(len(data_pertandingan))* untuk memeriksa setiap *dictionary* satu per satu. Jika ID yang diinput cocok dengan *data_pertandingan[i]['id']*, program akan menyimpan indeks *i* tersebut dan mengonfirmasi bahwa data telah ditemukan. Setelah ID valid ditemukan, program akan meminta admin untuk memasukkan semua data baru. Berbeda dengan membuat elemen baru, proses pembaruan data bekerja dengan mengakses *dictionary* yang ada pada indeks yang telah ditemukan dan memperbarui nilai untuk setiap *key* secara individual, seperti *data_pertandingan[i]['pemain_putih'] = nama_putih_baru.* Dengan cara ini, program secara presisi mengubah konten dari *dictionary* yang sudah ada tanpa mengganti strukturnya, lalu menghentikan perulangan setelah berhasil.

Source Code:

```
                    elif pilihan_menu == '3':
                        os.system('cls || clear')
                        print('UBAH DATA PERTANDINGAN')
                        tabel = PrettyTable()
                        tabel.field_names = ['ID', 'Pemain Putih', 'Pemain
Hitam', 'Jumlah Langkah', 'Hasil']
                        if not data_pertandingan:
                            print('Belum ada data pertandingan untuk
diubah.')
                            input('Tekan Enter untuk kembali...')
                            continue
                        if data_pertandingan:
                            for pertandingan in data_pertandingan:
                                tabel.add_row([
                                    pertandingan['id'],
                                    pertandingan['pemain_putih'],
                                    pertandingan['pemain_hitam'],
```

```
                                    pertandingan['jumlah_langkah'],
                                    pertandingan['pemenang']
                                ])
                            print(tabel)

                        while True:
                            input_id_str = ''
                            while not input_id_str.strip():
                                input_id_str = input('\nMasukkan ID yang
ingin diubah (atau ketik "x" untuk kembali): ')
                                if not input_id_str.strip():
                                    print('ID tidak boleh kosong. Silakan
coba lagi.')
                            if input_id_str == 'x':
                                break
                            elif not input_id_str.isdigit():
                                print('Input tidak valid. ID harus berupa
angka.')
                                continue

                            input_id = int(input_id_str)
                            data_ditemukan = False

                            for i in range(len(data_pertandingan)):
                                if data_pertandingan[i]['id'] == input_id:
                                    data_ditemukan = True
                                    indeks_data = i
                                    break

                            if data_ditemukan == True:
                                print(f'Mengubah data untuk ID {input_id}:
{data_pertandingan[indeks_data]}')
                                nama_putih_baru = ''
                                while not nama_putih_baru.strip():
                                    nama_putih_baru = input('Nama Pemain
Putih Baru: ')
                                    if not nama_putih_baru.strip():
                                        print('Nama Pemain Putih tidak boleh
kosong.')

                                nama_hitam_baru = ''
                                while not nama_hitam_baru.strip():
                                    nama_hitam_baru = input('Nama Pemain
Hitam Baru: ')
                                    if not nama_hitam_baru.strip():
                                        print('Nama Pemain Hitam tidak boleh
kosong.')
```

```
                                while True:
                                    jumlah_langkah_baru_str = ''
                                    while not
jumlah_langkah_baru_str.strip():
                                        jumlah_langkah_baru_str =
input('Jumlah Langkah Baru: ')
                                        if not
jumlah_langkah_baru_str.strip():
                                            print('Jumlah Langkah tidak
boleh kosong.')
                                    if jumlah_langkah_baru_str.isdigit():
                                        jumlah_langkah_baru =
int(jumlah_langkah_baru_str)
                                        break
                                    else:
                                        print('Input tidak valid, jumlah
langkah harus angka!')

                                while True:
                                    print('\nPilihan Hasil Baru: 1. Putih
Menang, 2. Hitam Menang, 3. Seri')
                                    pilihan_hasil_baru = ''
                                    while not pilihan_hasil_baru.strip():
                                        pilihan_hasil_baru = input('Pilihan
(1/2/3): ')
                                        if not pilihan_hasil_baru.strip():
                                            print('Pilihan hasil tidak boleh
kosong.')
                                    if pilihan_hasil_baru in ['1', '2',
'3']:
                                        if pilihan_hasil_baru == '1':
hasil_baru = 'Putih Menang'
                                        elif pilihan_hasil_baru == '2':
hasil_baru = 'Hitam Menang'
                                        else: hasil_baru = 'Seri'
                                        break
                                    else:
                                        print('Pilihan tidak valid.')

                                data_pertandingan[i]['pemain_putih'] =
nama_putih_baru
                                data_pertandingan[i]['pemain_hitam'] =
nama_hitam_baru
                                data_pertandingan[i]['jumlah_langkah'] =
jumlah_langkah_baru_str
                                data_pertandingan[i]['pemenang'] =
```

```
pilihan_hasil_baru
                                print('Data berhasil diubah!')
                                input('Tekan Enter untuk kembali...')
                                break
                            else:
                                print(f'ID {input_id} tidak ditemukan.
Silakan coba lagi.')
```

F. Menu Admin DELETE

Fungsi "Menu Admin DELETE" memungkinkan admin untuk menghapus data pertandingan secara permanen berdasarkan ID. Serupa dengan fitur *Update*, alurnya dimulai dengan menampilkan seluruh data yang ada agar admin bisa memilih ID mana yang akan dihapus. Program kemudian memasuki perulangan *while True* untuk meminta input ID dan memvalidasinya. Logika utamanya adalah mencari indeks (posisi) dari data yang akan dihapus dengan melakukan iterasi *for i in range(len(data_pertandingan))*. Jika ID yang dimasukkan cocok dengan *data_pertandingan[i]['id'],* program akan menyimpan nomor indeks tersebut. Inti dari proses penghapusan dieksekusi oleh baris *data_pertandingan.pop(indeks_data),* di mana metode .pop() secara spesifik menghapus elemen dari list *data_pertandingan* pada posisi *indeks_data* yang telah ditemukan. Setelah data berhasil dihapus, program memberikan notifikasi sukses dan menghentikan perulangan.

Source Code:

```
                    elif pilihan_menu == '4':
                        os.system('cls || clear')
                        print('HAPUS DATA PERTANDINGAN')
                        tabel = PrettyTable()
                        tabel.field_names = ['ID', 'Pemain Putih', 'Pemain
Hitam', 'Jumlah Langkah', 'Hasil']
                        if not data_pertandingan:
                            print(Fore.RED + '\nBelum ada data pertandingan
untuk diubah.')
                            input(Fore.CYAN + 'Tekan Enter untuk kembali...' +
Style.RESET_ALL)
                            continue
                        if data_pertandingan:
                            for pertandingan in data_pertandingan:
                                tabel.add_row([
                                    pertandingan['id'],
                                    pertandingan['pemain_putih'],
                                    pertandingan['pemain_hitam'],
                                    pertandingan['jumlah_langkah'],
                                    pertandingan['pemenang']
                                ])
                            print(tabel)
```

```
                        while True:
                            input_id_str = ''
                            while not input_id_str.strip():
                                input_id_str = input('\nMasukkan ID yang mau
dihapus (atau ketik "X" untuk kembali): ')
                                if not input_id_str.strip():
                                    print(Fore.RED + "ID tidak boleh kosong.
Silakan coba lagi.")
                            if input_id_str == 'x':
                                break
                            elif not input_id_str.isdigit():
                                print(Fore.RED + '\nID harus berupa angka!')
                                continue


                            input_id = int(input_id_str)
                            data_ditemukan = False

                            for i in range(len(data_pertandingan)):
                                if data_pertandingan[i]['id'
] == input_id:
                                    data_ditemukan = True
                                    indeks_data = i
                                    break

                            if data_ditemukan == True:
                                data_pertandingan.pop(indeks_data)
                                print(f'Data ID {input_id} berhasil dihapus!')
                                input('Tekan Enter untuk kembali...')
                                break
                            else:
                                print('ID tidak ditemukan.')
```

G. Menu Admin LOGOUT

Bagian terakhir dari menu admin adalah *elif pilihan_menu == '0'*, yang berfungsi sebagai fitur Logout. Logikanya sangat sederhana: jika pengguna memilih '0', program akan menampilkan pesan konfirmasi "Anda telah logout." Perintah *input()* digunakan untuk menjeda program sejenak, memberikan kesempatan bagi pengguna untuk membaca pesan tersebut. Setelah pengguna menekan Enter, perintah *break* akan dieksekusi. Perintah *break* ini adalah inti dari fungsionalitas logout, karena ia secara paksa menghentikan perulangan (loop) menu admin, sehingga program akan keluar dari sesi admin dan kembali ke menu utama (Login dan Register). Terakhir, blok *else* berfungsi sebagai penanganan error untuk semua input lain yang tidak valid (selain 1, 2, 3, 4, atau 0), di mana program akan menampilkan pesan "Pilihan tidak valid" dan meminta pengguna untuk mencoba lagi.

Source Code:

```
                    elif pilihan_menu == '0':
                        print('Anda telah logout.')
                        input('Tekan Enter untuk kembali...')
                        break
                    else:
                        print('Pilihan tidak valid.')
                        input('Tekan Enter untuk mencoba lagi...')
```

H. Menu User

Setelah login berhasil, program memasuki sebuah loop *while True* yang akan terus berjalan hingga pengguna memilih logout. Di dalam loop ini, program menggunakan struktur kondisional *if peran_user == 'admin':*. Karena dalam sistem ini hanya ada dua peran, blok *else:* secara efektif akan dieksekusi untuk semua pengguna yang bukan admin, yaitu user biasa. Blok ini pertama-tama akan menampilkan pesan selamat datang yang dipersonalisasi dan menu user yang lebih terbatas. Setelah menerima INPUT *pilihan_menu,* program akan menggunakan *if-elif-else* untuk menentukan tindakan: jika pengguna memilih '1', fungsionalitas untuk melihat data akan dijalankan, yang logikanya sama persis dengan menu *Read* pada admin. Jika pengguna memilih '0', fitur Logout dieksekusi; program akan menampilkan pesan konfirmasi, lalu perintah *break* akan menghentikan loop menu user, yang secara efektif mengembalikan pengguna ke menu utama. Terakhir, blok *else* di dalamnya berfungsi untuk menangani semua input lain yang tidak valid, dengan menampilkan pesan error.

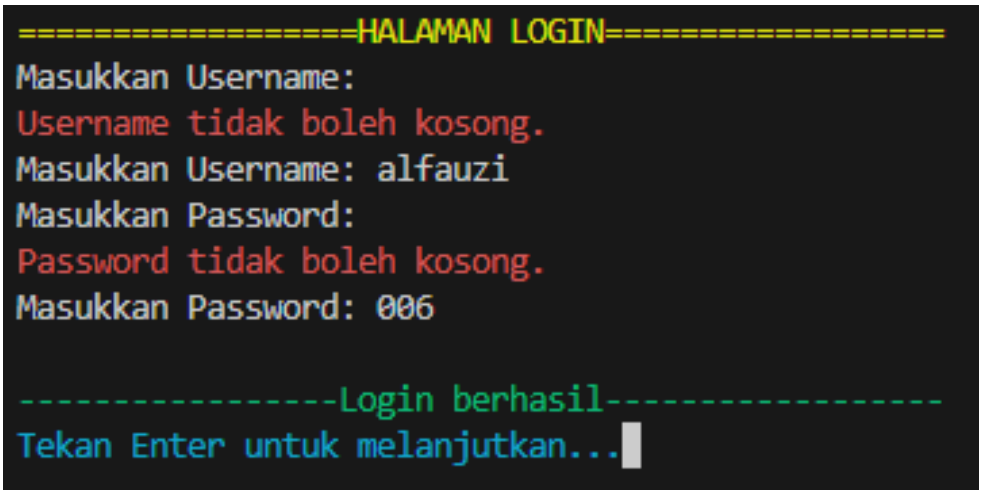
Source Code:

```
                else:
                    print(f'Selamat Datang, {input username}')
                    print(f'MENU USER')
                    print(menu_user)
                    pilihan_menu = ''
                    while not pilihan_menu.strip():
                        pilihan_menu = input('Pilih menu: ')
                        if not pilihan_menu.strip():
                            print("Pilihan tidak boleh kosong. Silakan coba
lagi.")


                    if pilihan_menu == '1':
                        os.system('cls || clear')
                        print('DATA HASIL PERTANDINGAN')
                        tabel = PrettyTable()
                        tabel.field_names = ['ID', 'Pemain Putih', 'Pemain
Hitam', 'Jumlah Langkah', 'Pemenang']
                        if data_pertandingan:
```

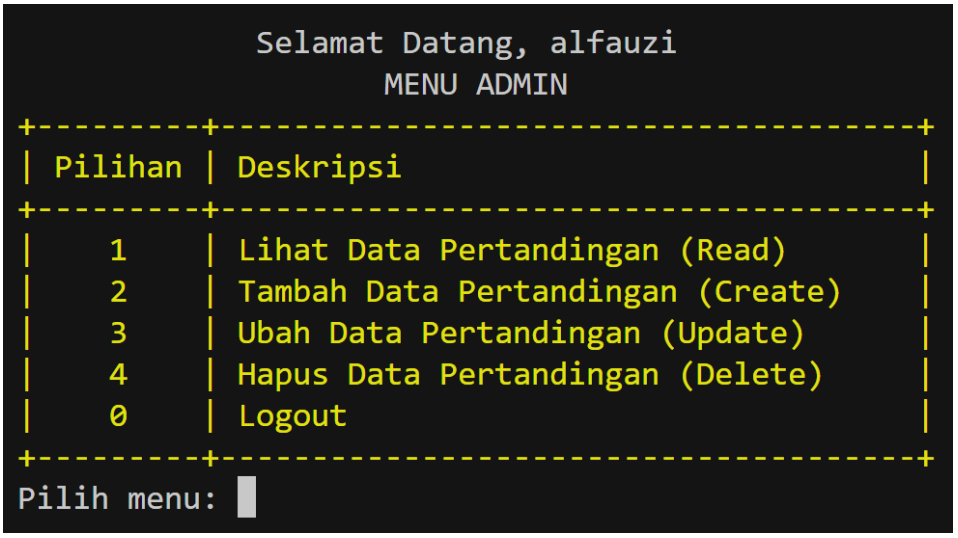
```
                        for pertandingan in data_pertandingan:
                            tabel.add_row([
                                pertandingan['id'],
                                pertandingan['pemain_putih'],
                                pertandingan['pemain_hitam'],
                                pertandingan['jumlah_langkah'],
                                pertandingan['pemenang']
                            ])
                        print(tabel)
                    else:
                        print(Fore.RED + '\nBelum ada data pertandingan.')
                    input(Fore.CYAN + '\nTekan Enter untuk kembali...' +
Style.RESET_ALL)


                elif pilihan_menu == '0':
                    print('Anda telah logout.')
                    input('Tekan Enter untuk kembali...')
                    break
                else:
                    print('Pilihan tidak valid.')
                    input('Tekan Enter untuk mencoba lagi...')
```

4. Hasil Output

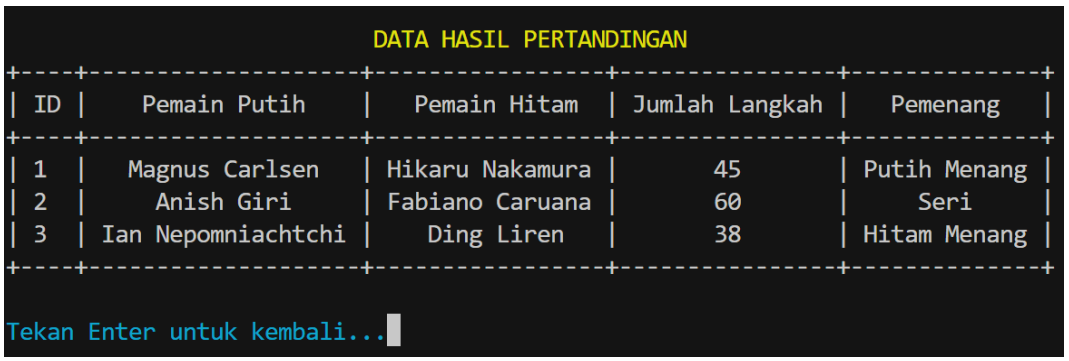
gambar 4.1 terminal menu utama



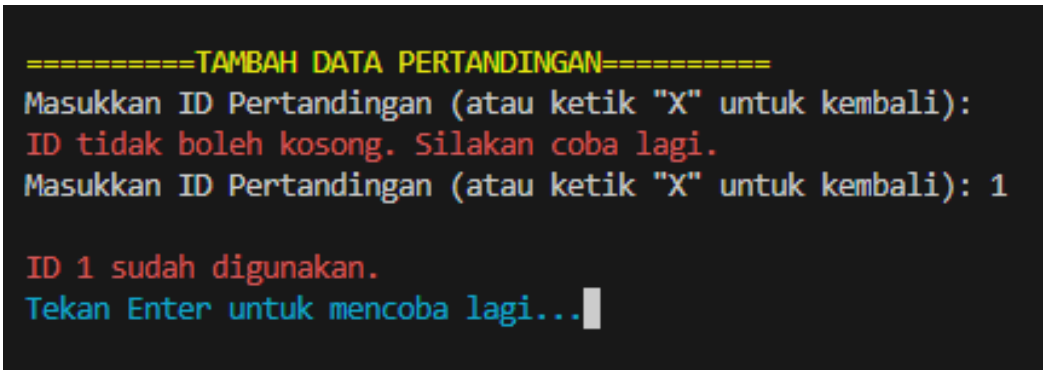
gambar 4.2 terminal menu login dan pengecekan input



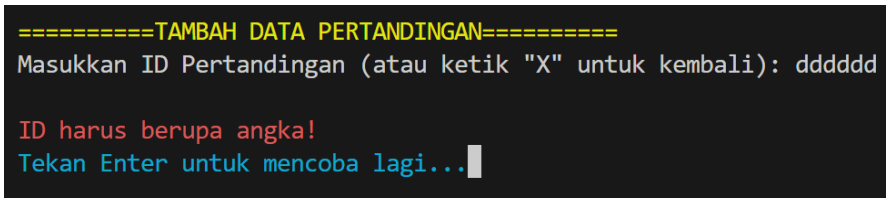
gambar 4.3 terminal menu admin



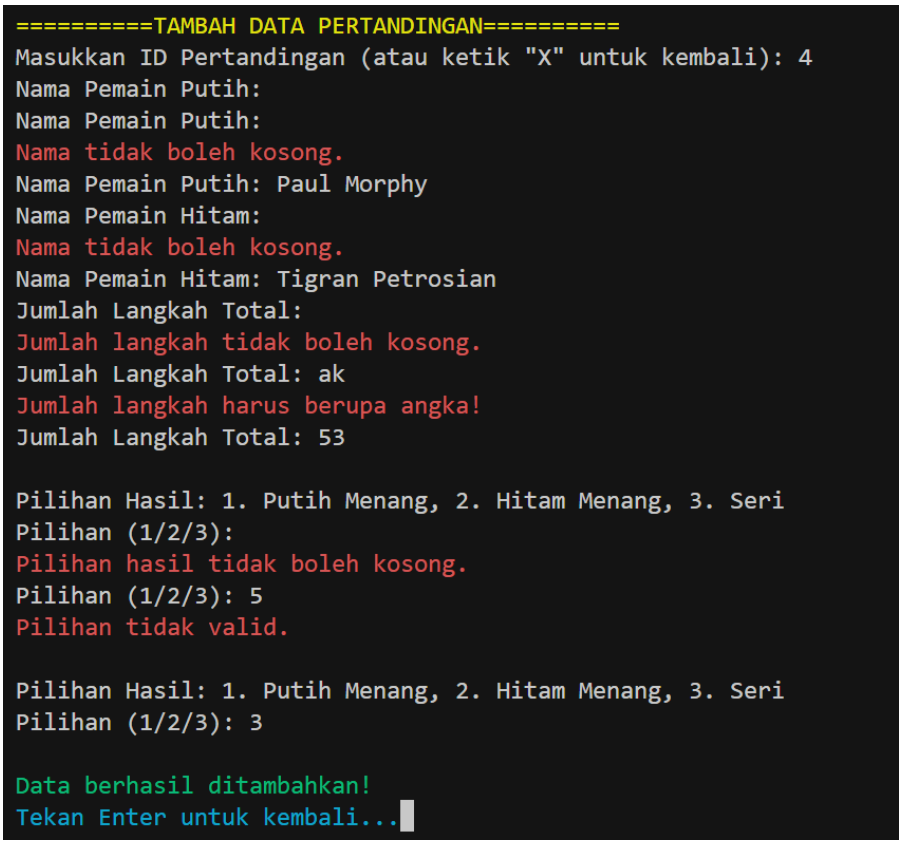
gambar 4.4 terminal menu admin READ



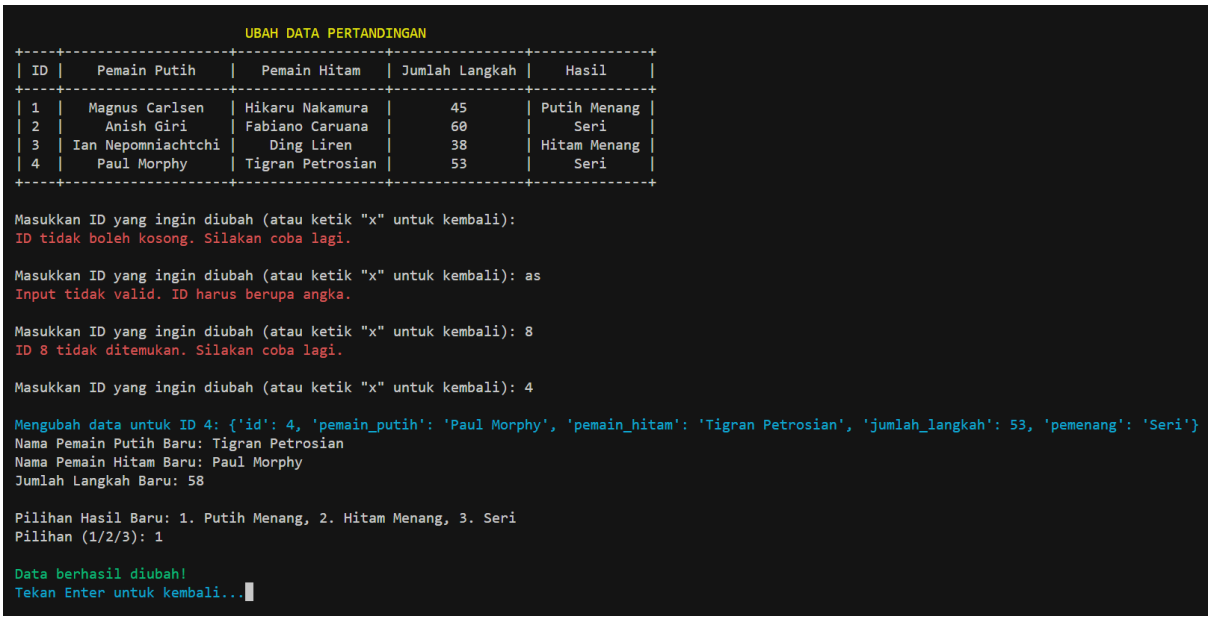
gambar 4.5 terminal menu admin CREATE ID sudah ada dan input tidak boleh kosong



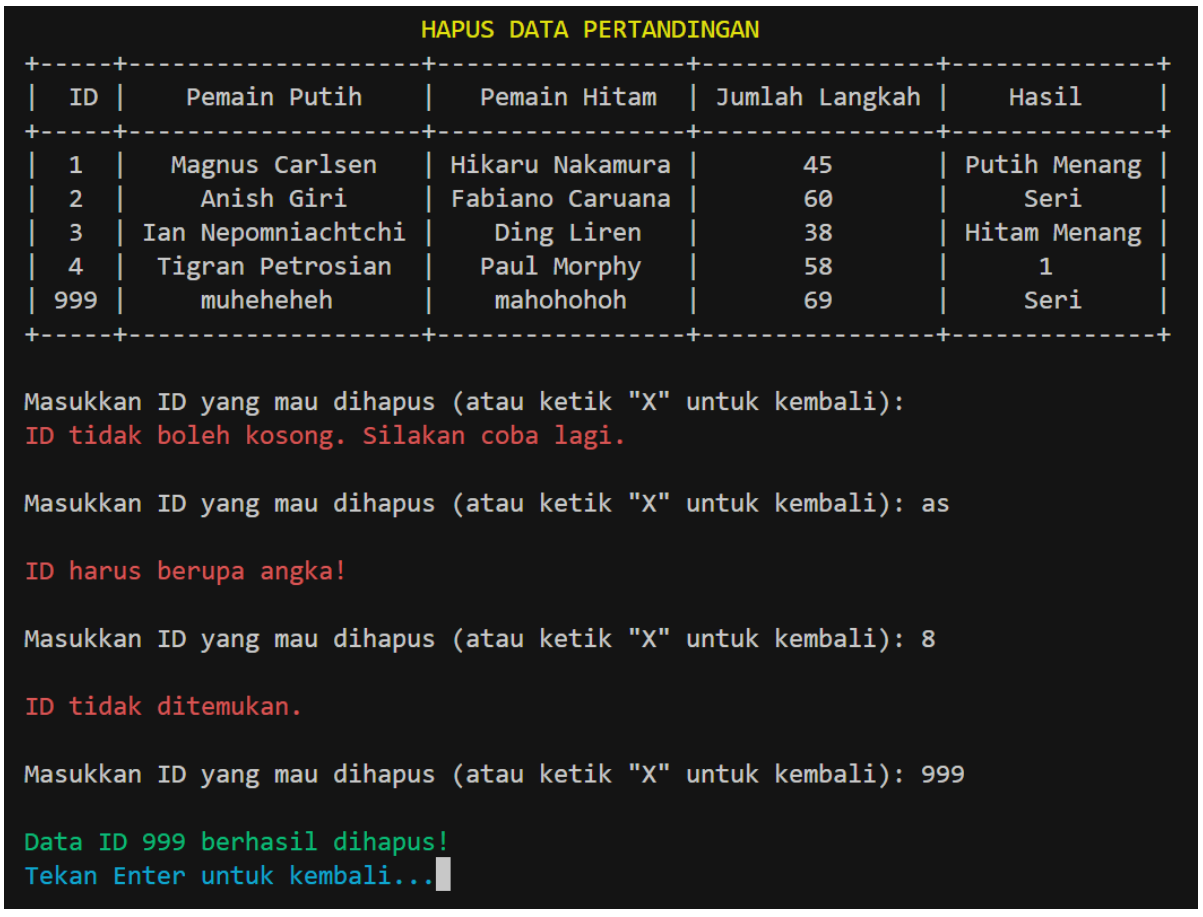
gambar 4.6 terminal menu admin CREATE input harus angka



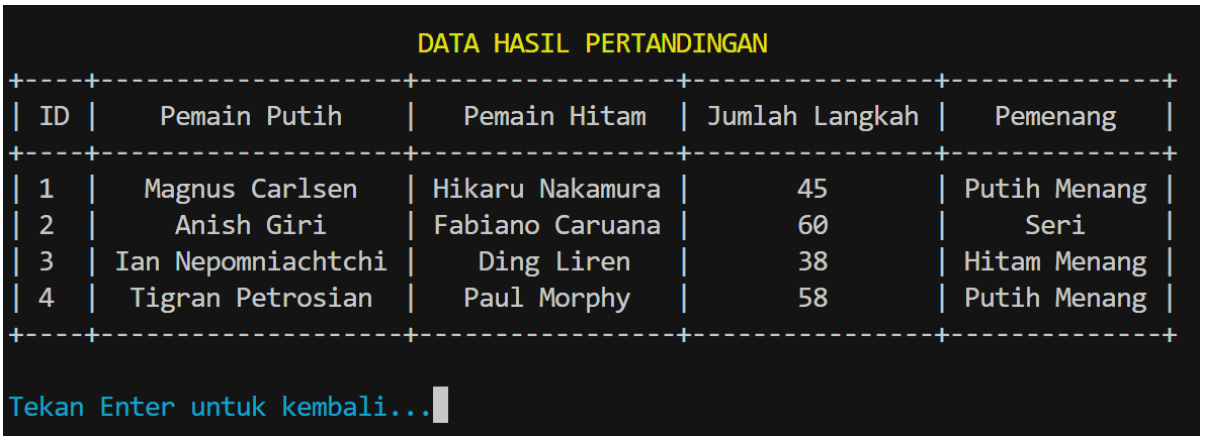
gambar 4.7 terminal menu admin CREATE pengecekan input



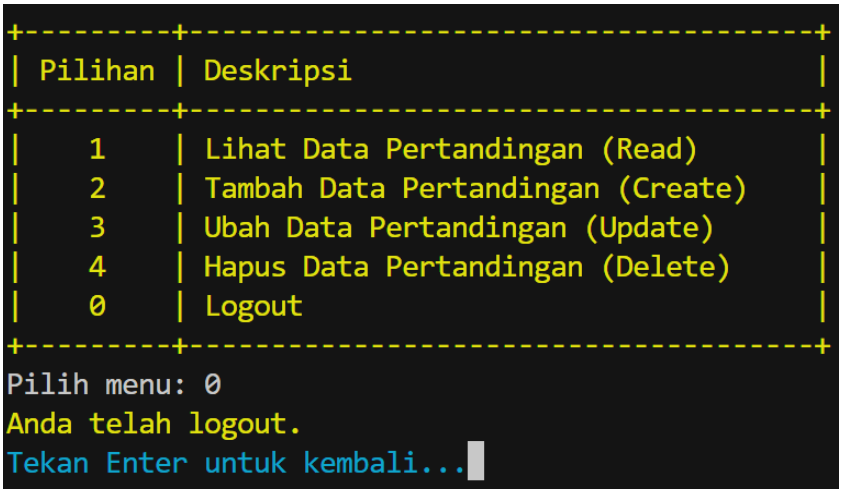
gambar 4.8 terminal menu admin CREATE pengecekan input



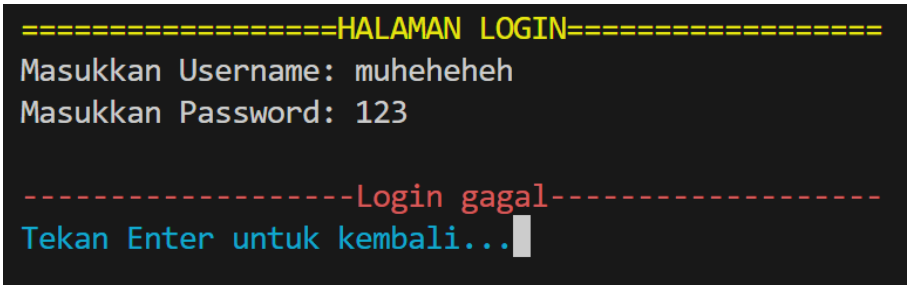
gambar 4.9 terminal menu admin DELETE pengecekan input ID999 berhasil dihapus



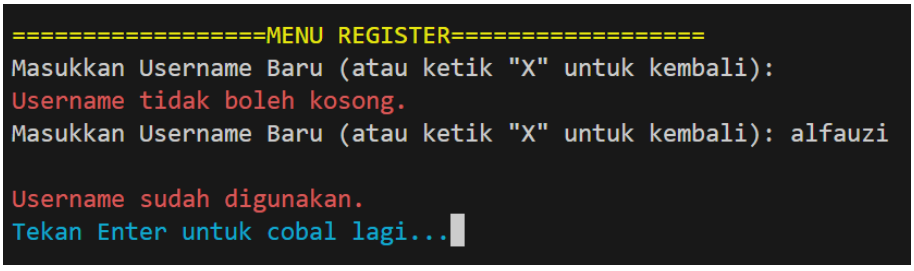
gambar 4.10 terminal menu admin READ setelah CURD



gambar 4.11 terminal menu admin LOGOUT



gambar 4.12 terminal halaman login gagal



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...
```

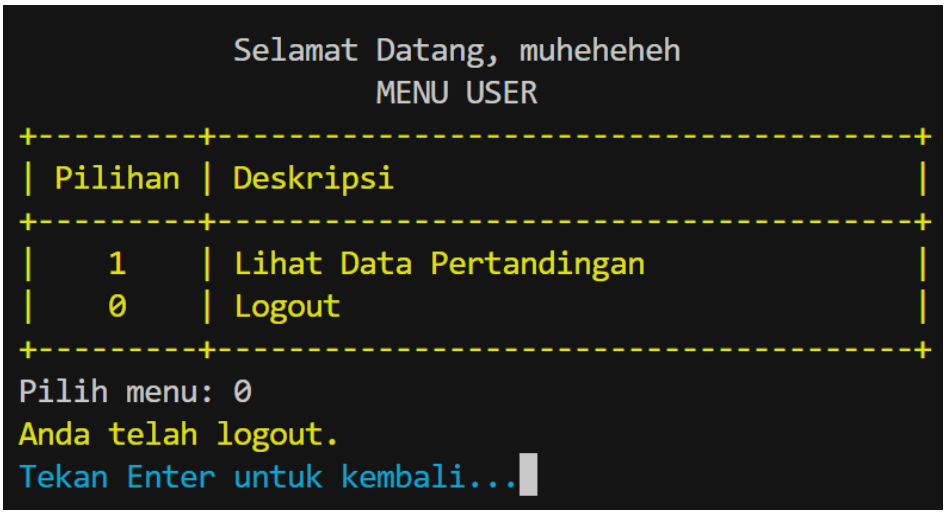
gambar 4.15 terminal halaman login dan berhasil

```
            Selamat Datang, muheheheh
                    MENU USER
+---------+-----------------------------------------+
| Pilihan | Deskripsi                               |
+---------+-----------------------------------------+
|    1    | Lihat Data Pertandingan                 |
|    0    | Logout                                  |
+---------+-----------------------------------------+
Pilih menu:
```

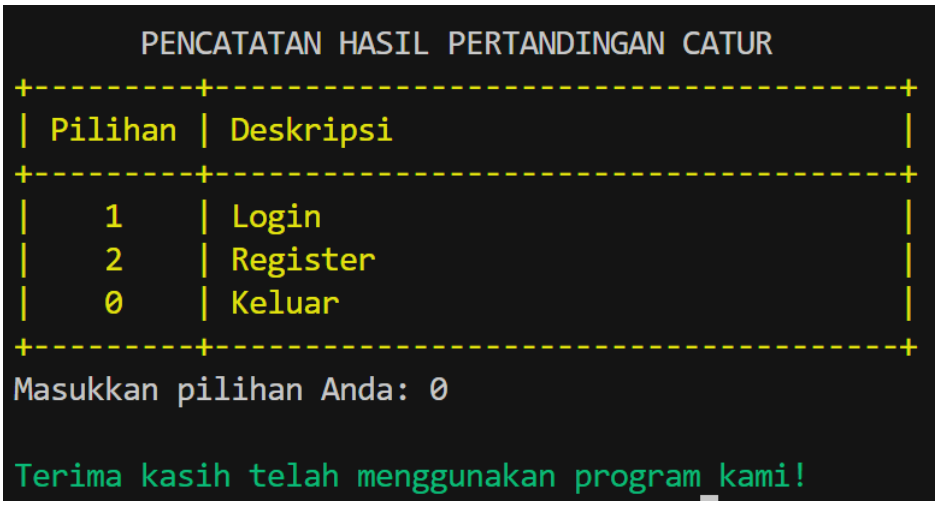
gambar 4.16 terminal menu user

```
                        DATA HASIL PERTANDINGAN
+----+-------------------+----------------+---------------+-------------+
| ID |    Pemain Putih   |  Pemain Hitam  | 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...
```

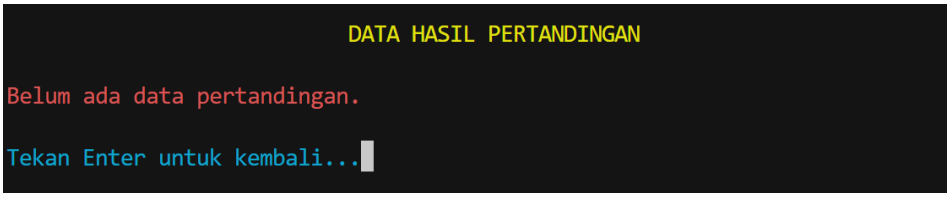
gambar 4.17 terminal menu user READ



gambar 4.18 terminal menu LOGOUT



gambar 4.19 terminal menu keluar



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:   post-test/post-test-apd-6/2509106006-Muhammad-Alfauzi-Syahputra-PT-6.py
        new file:   post-test/post-test-apd-6/__pycache__/data.cpython-313.pyc
        new file:   post-test/post-test-apd-6/__pycache__/menu.cpython-313.pyc
        new file:   post-test/post-test-apd-6/data.py
        new file:   post-test/post-test-apd-6/menu.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 'posttest 6'
[main 2b11a82] posttest 6
 5 files changed, 427 insertions(+)
 create mode 100644 post-test/post-test-apd-6/2509106006-Muhammad-Alfauzi-Syahputra-PT-6.py
 create mode 100644 post-test/post-test-apd-6/__pycache__/data.cpython-313.pyc
 create mode 100644 post-test/post-test-apd-6/__pycache__/menu.cpython-313.pyc
 create mode 100644 post-test/post-test-apd-6/data.py
 create mode 100644 post-test/post-test-apd-6/menu.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: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 16 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 4.78 KiB | 4.78 MiB/s, done.
Total 10 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Alfauzi-S/praktikum-apd.git
   96d358d..2b11a82  main -> main
```

gambar 5.3 mendorong commit dari repository lokal ke rwepository remote