

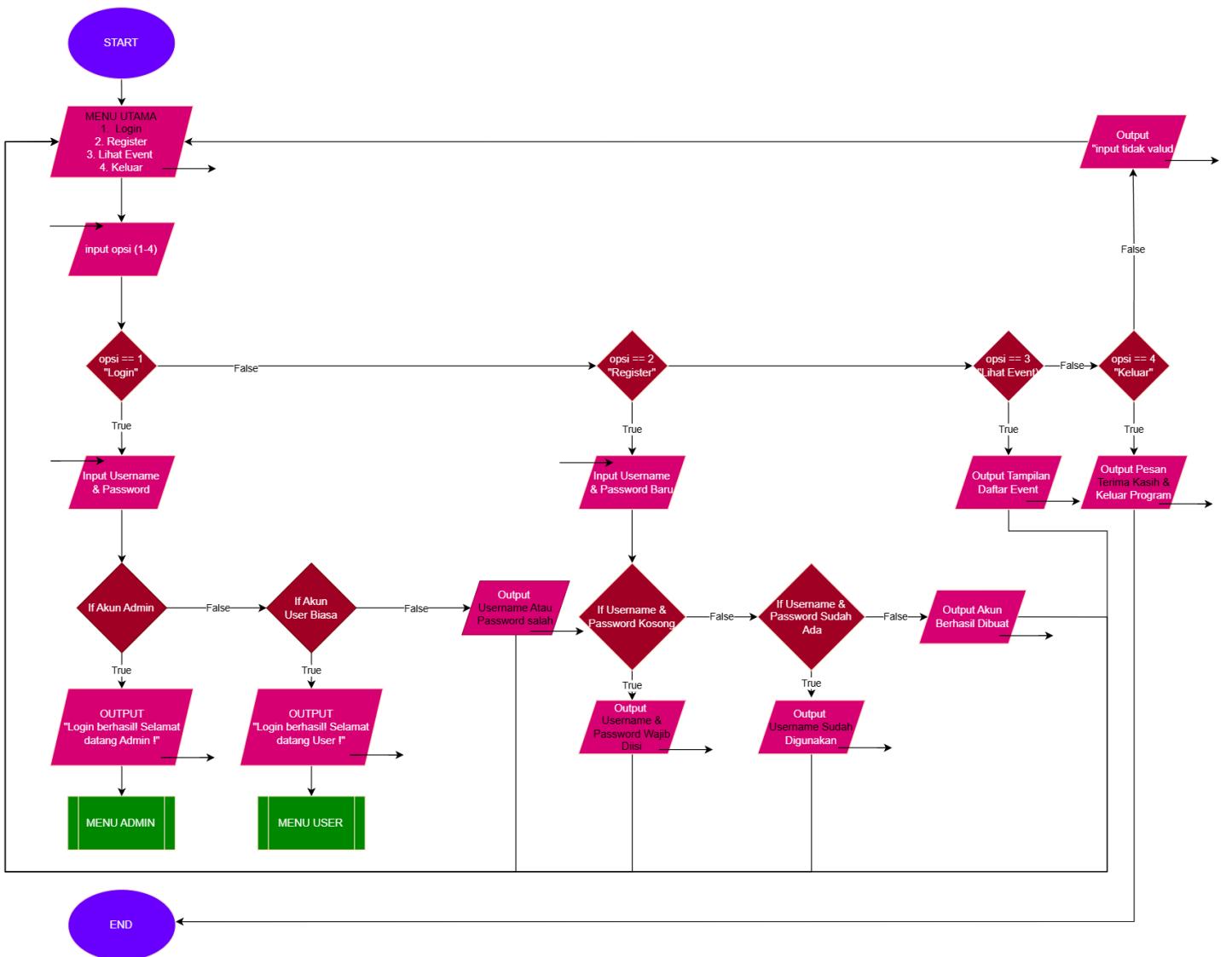
LAPORAN PRAKTIKUM
POSTTEST 8
ALGORITMA PEMROGRAMAN DASAR



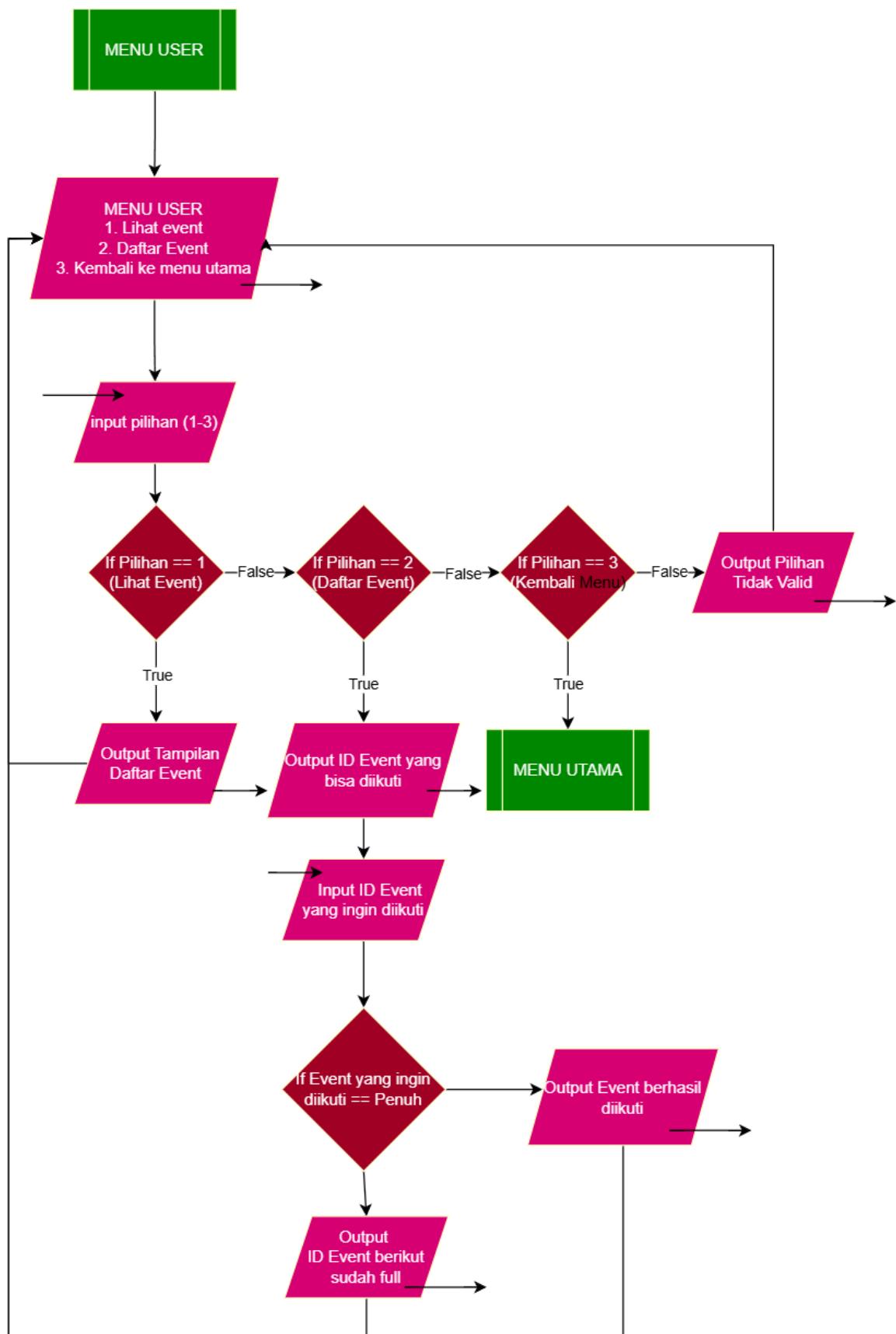
Disusun oleh:
Nuril Akmal 2509106074
Kelas B2'25

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowcart



Gambar 1.1 Flowchart Menu Utama



Gambar 1.2 Flowchart Menu User



Gambar 1.3 Flowcart Menu Admin

PENJELASAN FLOWCHART SISTEM MANAJEMEN EVENT WEB3 & HACKATHON

1. MENU UTAMA

- Start → Sistem menampilkan 4 pilihan:
 - Login
 - Register
 - Lihat Event (Guest)
 - Keluar
- Jika input tidak valid → tampil pesan error dan ulangi

2. LOGIN

- User input username & password
- Verifikasi:
 - Jika Admin → arahkan ke Menu Admin
 - Jika User Biasa → arahkan ke Menu User
 - Jika salah → tampil "Username atau password salah"

3. REGISTER

- Input nama baru & password
- Cek duplikasi username
- Jika tersedia → tambah ke database users
- Tampil "Registrasi berhasil"

4. MENU ADMIN

Opsi yang tersedia:

- Lihat Event → tampilkan semua event
- Create Event → input data event baru + validasi status
- Update Event → edit data event tertentu
- Delete Event → hapus event dari sistem
- Kelola Peserta → tambah/hapus peserta event
- Logout → kembali ke Menu Utama

5. MENU USER BIASA

Opsi yang tersedia:

- Lihat Event → tampilkan daftar event
- Daftar Event → mendaftar ke event ongoing
- Logout → kembali ke Menu Utama

6. GUEST MODE

- Lihat Event saja tanpa bisa daftar
- Kembali ke Menu Utama

7. VALIDASI & PENGULANGAN

- Setiap aksi selesai → kembali ke menu sebelumnya

- Input tidak valid → minta input ulang
- Event tidak ditemukan → tampil pesan error

8. END

- User pilih Keluar → tampil pesan penutup → Program selesai

2. Deskripsi Singkat Program

Program ini merupakan sistem manajemen event Web3 dan Hackathon berbasis terminal yang memungkinkan pengguna untuk melakukan login, registrasi, serta mengelola data event. Program ini membedakan antara akun admin dan user biasa: admin dapat menambah, melihat, memperbarui, menghapus event, serta mengelola peserta, sedangkan user biasa hanya dapat melihat daftar event dan mendaftar sebagai peserta pada event yang sedang berlangsung.

Data admin, pengguna, dan event disimpan dalam struktur data Python sederhana menggunakan dictionary dan list of dictionaries. Antarmuka program disajikan melalui menu berbasis teks yang interaktif dengan perintah os.system('cls'/'clear') untuk membersihkan layar. Program berjalan dalam loop utama hingga pengguna memilih untuk keluar dari sistem.

3. Source Code

A. Main.py (Jalankan Program)

```
from modul_util import clear_screen, input_int
from modul_user import login, register
from modul_event import tampilkan_event, create_event, update_event,
delete_event, daftar_peserta

while True:
    clear_screen()
    print("=*60)
    print("SISTEM MANAJEMEN EVENT WEB3 & HACKATHON".center(60))
    print("=*60)
    print("1. Login")
    print("2. Register")
    print("3. Lihat Event (Guest)")
    print("4. Keluar")
    pilihan = input("Pilih opsi (1-4): ")

    if pilihan == '4':
        clear_screen()
        print("Terima kasih telah menggunakan sistem ini.")
        break

    elif pilihan == '1':
        clear_screen()
        user = input("Masukkan username: ")
```

```
pwd = input("Masukkan password: ")
tipe_user = login(user, pwd)

if tipe_user == "user":
    input("Login berhasil! Tekan Enter untuk melanjutkan...")
    while True:
        clear_screen()
        print(f"\nMenu Pengguna - {user}")
        print("1. Lihat Event")
        print("2. Daftar Event")
        print("3. Logout")
        choice = input("Pilih menu (1-3): ")

        if choice == '1':
            clear_screen()
            tampilkan_event()
            input("Tekan Enter untuk kembali...")

        elif choice == '2':
            clear_screen()
            tampilkan_event()
            event_id = input_int("Masukkan ID event yang ingin
diikuti: ")
            if daftar_peserta(event_id, user):
                print("Berhasil mendaftar!")
            else:
                print("Gagal mendaftar. Event tidak ditemukan atau
sudah selesai.")
            input("Tekan Enter untuk kembali...")

        elif choice == '3':
            break

elif tipe_user == "admin":
    input("Login ADMIN berhasil! Tekan Enter untuk melanjutkan...")
    while True:
        clear_screen()
        print("\nMenu Admin")
        print("1. Lihat Event")
        print("2. Create Event")
        print("3. Update Event")
        print("4. Delete Event")
        print("5. Logout")
        choice = input("Pilih menu (1-5): ")

        if choice == '1':
            clear_screen()
            tampilkan_event()
```

```
        input("Tekan Enter untuk kembali...")

    elif choice == '2':
        clear_screen()
        nama = input("Nama event: ")
        hadiah = input("Hadiah event: ")
        while True:
            status = input("Status (ongoing/finished):
").lower()
            if status in ["ongoing", "finished"]:
                break
            print("Status harus ongoing/finished.")
        create_event(nama, hadiah, status)
        print("Event berhasil dibuat!")
        input("Tekan Enter untuk kembali...")

    elif choice == '3':
        clear_screen()
        event_id = input_int("Masukkan ID event: ")
        nama = input("Nama event baru: ")
        hadiah = input("Hadiah baru: ")
        while True:
            status = input("Status baru (ongoing/finished):
").lower()
            if status in ["ongoing", "finished"]:
                break
            print("Status harus ongoing/finished.")
        if update_event(event_id, nama, hadiah, status):
            print("Update berhasil!")
        else:
            print("Event tidak ditemukan.")
        input("Tekan Enter untuk kembali...")

    elif choice == '4':
        clear_screen()
        event_id = input_int("Masukkan ID event yang ingin
dihapus: ")
        if delete_event(event_id):
            print("Event berhasil dihapus!")
        else:
            print("Event tidak ditemukan.")
        input("Tekan Enter untuk kembali...")

    elif choice == '5':
        break

else:
    print("Username atau password salah.")
```

```

        input("Tekan Enter untuk mencoba lagi...")

    elif pilihan == '2':
        clear_screen()
        nama = input("Masukkan nama baru: ")
        pwd = input("Masukkan password baru: ")
        if register(nama, pwd):
            print("Registrasi berhasil! Silakan login.")
        else:
            print("Nama sudah terdaftar.")
        input("Tekan Enter untuk kembali...")

    elif pilihan == '3':
        clear_screen()
        tampilkan_event()
        input("Tekan Enter untuk kembali...")

    else:
        clear_screen()
        print("Opsi tidak valid.")
        input("Tekan Enter untuk melanjutkan...")

```

B. Modul.Event (Melihat Event)

```

from prettytable import PrettyTable

events = [
    [1, "ETH Hackathon Samarinda", "10.000 USDT + NFT", "ongoing", ["Team Alpha", "Team Beta"]],
    [2, "Solana DeFi Camp", "5.000 USDT", "finished", ["DeFi Ninjas"]],
]

def tampilkan_event():
    table = PrettyTable()
    table.field_names = ["ID", "Nama Event", "Hadiah", "Status", "Peserta"]
    for event in events:
        table.add_row([event[0], event[1], event[2], event[3], ", ".join(event[4])])
    print(table)

def create_event(nama_event, hadiah_event, status_event):
    event_id = len(events) + 1
    events.append([event_id, nama_event, hadiah_event, status_event, []])

def update_event(event_id, nama_event, hadiah_event, status_event):
    for event in events:
        if event[0] == event_id:
            event[1] = nama_event

```

```

        event[2] = hadiah_event
        event[3] = status_event
        return True
    return False

def delete_event(event_id):
    for event in events:
        if event[0] == event_id:
            events.remove(event)
            return True
    return False

def daftar_peserta(event_id, peserta):
    for event in events:
        if event[0] == event_id:
            if peserta not in event[4]:
                event[4].append(peserta)
                return True
    return False

```

C. Modul.User (Program buat Pengguna/Admin)

```

admin = ["Nuril Akmal", "Akmal Ganteng", "admin"]
password_admin = ["123", "321", "admin123"]
daftar_user = ["Bang Pernanda", "Mba Triya", "pengguna"]
password_user = ["123", "321", "user123"]

def login(user, pwd):
    if user in daftar_user:
        index = daftar_user.index(user)
        if pwd == password_user[index]:
            return "user"
    elif user in admin:
        index = admin.index(user)
        if pwd == password_admin[index]:
            return "admin"
    return None

def register(nama_baru, pwd_baru):
    if nama_baru in admin or nama_baru in daftar_user:
        return False
    daftar_user.append(nama_baru)
    password_user.append(pwd_baru)
    return True

```

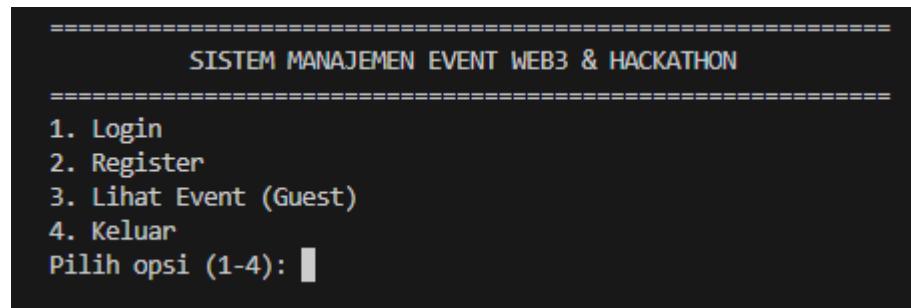
D. Modul.Util (Alat Bantu)

```
import os

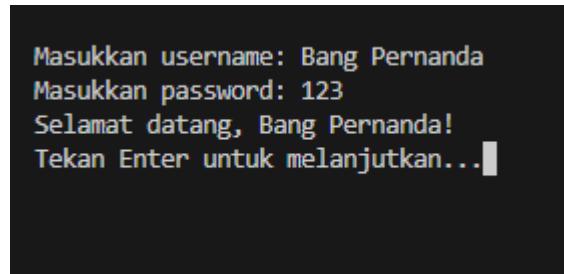
def clear_screen():
    os.system('cls' if os.name == 'nt' else 'clear')

def input_int(prompt):
    while True:
        try:
            return int(input(prompt))
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
```

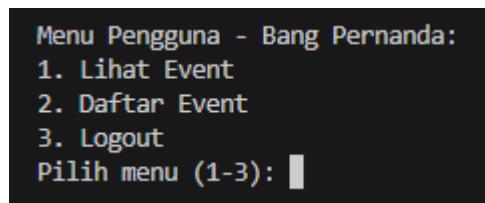
4. Hasil Output



Gambar 4.1 Menu



Gambar 4.2 Login Sebagai User(Peserta)



Gambar 4.3 Menu User

ID	Nama Event	Hadiah	Status	Peserta
1	ETH Hackathon Samarinda	10.000 USDT + NFT	ongoing	Team Alpha, Team Beta
2	Solana DeFi Camp	5.000 USDT	finished	DeFi Ninjas

Tekan Enter untuk kembali...█

Gambar 4.4 Daftar Event

Selamat datang ADMIN!

Menu:

1. Lihat Event
2. Create Event
3. Update Event
4. Delete Event
5. Kelola Peserta Event
6. Logout

Pilih menu (1-6): █

Gambar 4.5 Login Sebagai Admin

Daftar Event:
ID: 1, Nama: ETH Hackathon Samarinda, Hadiah: 10.000 USDT + NFT, Status: ongoing, Peserta: Team Alpha, Team Beta
ID: 2, Nama: Solana DeFi Camp, Hadiah: 5.000 USDT, Status: finished, Peserta: DeFi Ninjas

Tekan Enter untuk kembali ke menu...█

Gambar 4.6 Lihat Event

Masukkan nama event: Solona Run
Masukkan hadiah event: 100 USD
Masukkan status event (ongoing/finished): ongoing
Event berhasil ditambahkan!

Tekan Enter untuk kembali ke menu...█

Gambar 4.7 Creat Event

Masukkan nama event baru: Etharium Atlas Beach
Masukkan hadiah event baru: 200 USD
Masukkan status event baru (ongoing/finished): ongaing
Event berhasil diupdate!

Tekan Enter untuk kembali ke menu...█

Gambar 4.8 Update Event

```
Masukkan ID event untuk kelola peserta: 2
Peserta saat ini: DeFi Ninjas
1. Tambah Peserta
2. Hapus Peserta
Pilih opsi (1-2): 1
Masukkan nama peserta: Nibung Guysss
Peserta berhasil ditambahkan!

Tekan Enter untuk kembali ke menu...■
```

Gambar 4.9 Update Peserta

```
Masukkan ID event yang ingin dihapus: 1
Event berhasil dihapus!

Tekan Enter untuk kembali ke menu...■
```

Gambar 4.10 Delete Peserta

```
Masukkan nama baru: Dapupu teman Jarvis
Masukkan password baru: 123
Registrasi berhasil! Silakan login.
Tekan Enter untuk melanjutkan...■
```

Gambar 4.11 Register

```
Daftar Event:
ID: 2, Nama: Etharium Atlas Beach, Hadiah: 200 USD , Status: ongaing, Peserta: DeFi Ninjas, Nibung Guysss
ID: 3, Nama: Solona Run, Hadiah: 100 USD , Status: ongoing, Peserta:

Tekan Enter untuk kembali ke menu...■
```

Gambar 4.12 Lihat Event (Guest)

```
Terima kasih telah menggunakan sistem ini.
PS C:\Users\Acer\Documents\PRAKTIKUM APD\github\PRAKTIKUM_APD_B2_25\PRAKTIKUM_APD_B2-25> ■
```

Gambar 4.13 Logout

5. Langkah-langkah GIT

1. GIT Add

```
=====
PS C:\Users\Acer\Documents\PRAKTIKUM APD\github\PRAKTIKUM_APD_B2_25\PRAKTIKUM_APD_B2-25> git init
Reinitialized existing Git repository in C:/Users/Acer/Documents/PRAKTIKUM APD/github/PRAKTIKUM APD B2 25/PRAKTIKUM APD B2-25/.git/
PS C:\Users\Acer\Documents\PRAKTIKUM APD\github\PRAKTIKUM_APD_B2_25\PRAKTIKUM_APD_B2-25> git add .
PS C:\Users\Acer\Documents\PRAKTIKUM APD\github\PRAKTIKUM_APD_B2_25\PRAKTIKUM_APD_B2-25>
```

Gambar Git Add 5.1

Kegunaan Git add adalah setelah kita mengubah atau menambahkan file, kita perlu “menandai” file itu agar siap disimpan ke Git. Perintah git add berfungsi untuk memasukkan file atau perubahan ke dalam staging area, yaitu tempat sementara sebelum benar-benar disimpan permanen ke riwayat proyek.

2. GIT Commit

```
PS C:\Users\Acer\Documents\PRAKTIKUM APD\github\PRAKTIKUM_APD_B2_25\PRAKTIKUM_APD_B2-25> git add .
PS C:\Users\Acer\Documents\PRAKTIKUM APD\github\PRAKTIKUM_APD_B2_25\PRAKTIKUM_APD_B2-25>
```

Gambar Git Commit 5.2

Adapun Git Commit kegunaannya adalah melakukan perubahan sudah ditandai dengan git add, maka git commit dipakai untuk benar-benar menyimpan perubahan itu ke riwayat proyek Git. Setiap commit biasanya disertai pesan (commit message) yang menjelaskan apa yang diubah, supaya lebih mudah dilacak di kemudian hari.

3. GIT Push

```
PS C:\Users\Acer\Documents\PRAKTIKUM APD\github\PRAKTIKUM_APD_B2_25\PRAKTIKUM_APD_B2-25> git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 643 bytes | 107.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/elkuwas20/PRAKTIKUM_APD_B2-25.git
  96ca592..42c5c25 main -> main
PS C:\Users\Acer\Documents\PRAKTIKUM APD\github\PRAKTIKUM_APD_B2_25\PRAKTIKUM_APD_B2-25>
```

Gambar Git Push 5.3

Setelah kita punya repository lokal yang sudah terhubung dengan remote, kita bisa mengirim commit yang ada di komputer ke repository online dengan git push. Dengan begitu, semua perubahan akan tersimpan di server dan bisa dilihat atau digunakan oleh orang lain (atau oleh kita sendiri di komputer lain).