

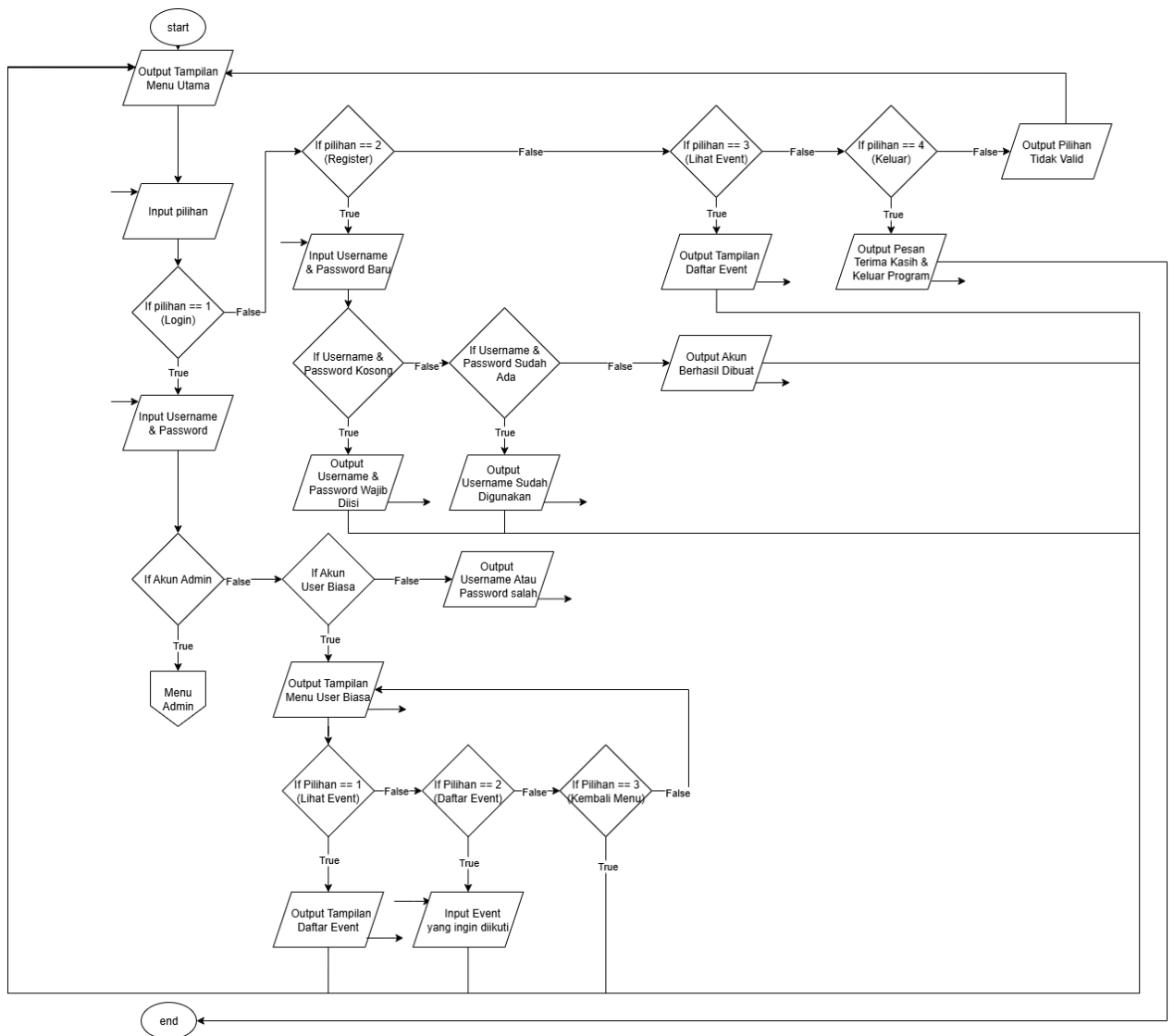
**LAPORAN PRAKTIKUM**  
**POSTTEST 6**  
**ALGORITMA PEMROGRAMAN DASAR**



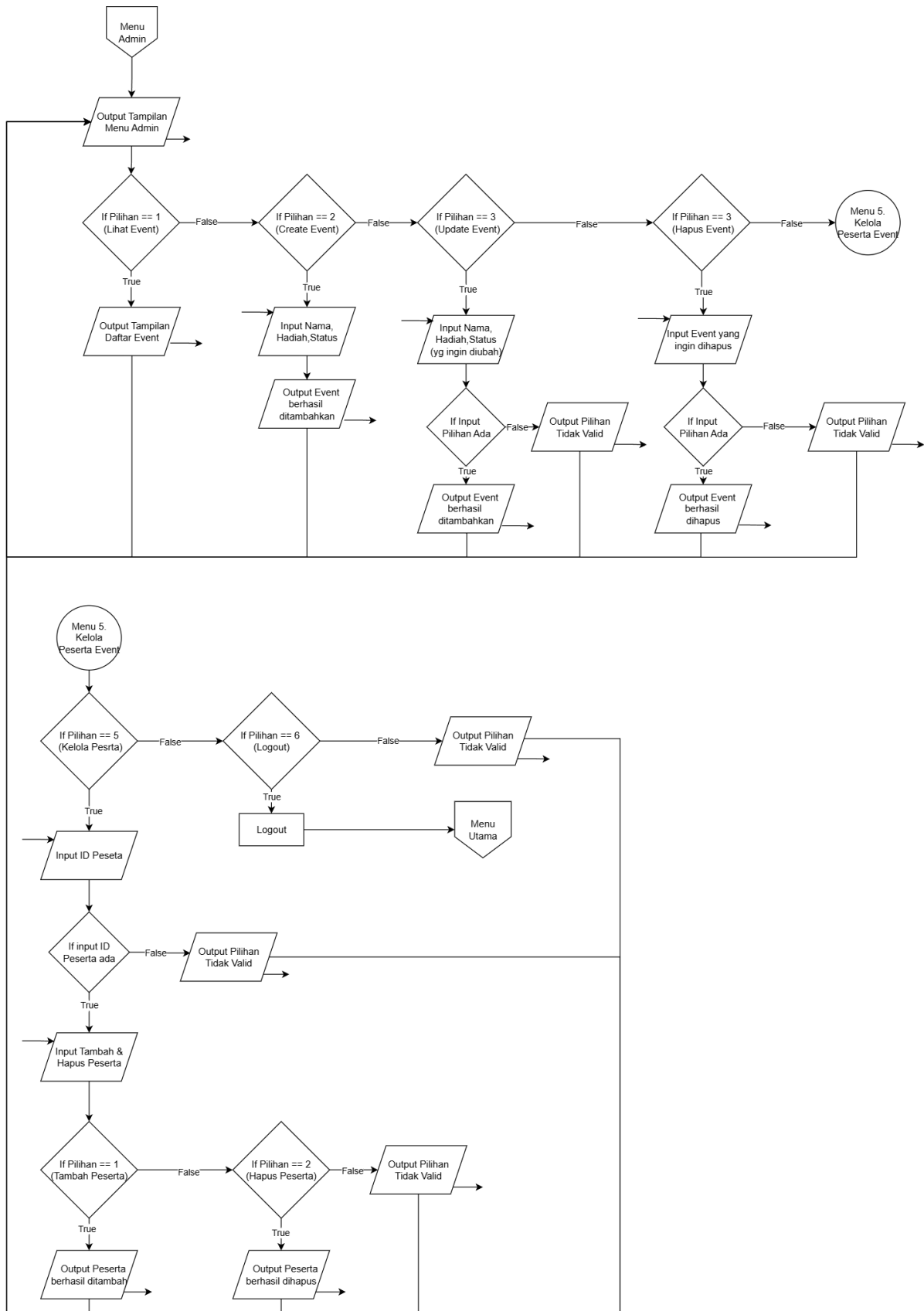
**Disusun oleh:**  
**Nuril Akmal 2509106074**  
**Kelas B2'25**

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

# 1. Flowcart



Gambar 1.1 Flowchart Menu Utama & User Biasa



Gambar 1.2 Flowchart 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. Inisialisasi Data

```
import os
import sys

admin = {
    "Nuril Akmal": "admin123",
    "Akmal Ganteng": "admin123",
    "admin": "admin123"
}
users = {
    "Bang Pernanda": "123",
    "Mba Triya": "321",
    "pengguna": "user123"
}
events = [
    {
        "id": 1,
        "nama": "ETH Hackathon Samarinda",
        "hadiah": "10.000 USDT + NFT",
    }
]
```

## B. Menu Utama

```
while True:
    os.system('cls' if os.name == 'nt' else 'clear')
    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':
        os.system('cls' if os.name == 'nt' else 'clear')
        print("Terima kasih telah menggunakan sistem ini.")
        sys.exit()
```

## C. Fitur Login

```
'.join(event['peserta']]))}")
        input("\nTekan Enter untuk kembali ke menu...")

        elif choice == '2':
            os.system('cls' if os.name == 'nt' else 'clear')
            print("\nDaftar Event yang Tersedia:")
            for event in events:
                if event['status'] == "ongoing":
                    print(f"ID: {event['id']}, Nama: {event['nama']}, Hadiah: {event['hadiah']}")

            event_id = input("Masukkan ID event yang ingin diikuti: ")

            if event_id.isdigit():
                event_id = int(event_id)
                event_ditemukan = False

                for event in events:
                    if event['id'] == event_id and event['status'] == "ongoing":
                        event_ditemukan = True
                        if user not in event['peserta']:
                            event['peserta'].append(user)
                            print(f"Berhasil mendaftar ke event {event['nama']}!")
                        else:
                            print("Anda sudah terdaftar di event ini.")
```

```

        break

        if not event_ditemukan:
            print("Event tidak ditemukan atau sudah
selesai.")
        else:
            print("ID event harus berupa angka.")

            input("\nTekan Enter untuk kembali ke menu...")

    elif choice == '3':
        print("Logout berhasil.")

```

#### D. Fitur Registrasi

```

elif pilihan == '2':
    os.system('cls' if os.name == 'nt' else 'clear')
    nama_baru = input("Masukkan nama baru: ")
    if nama_baru in admin or nama_baru in users:
        print("Nama sudah terdaftar. Silakan coba lagi.")
        input("Tekan Enter untuk melanjutkan...")
        continue
    pwd_baru = input("Masukkan password baru: ")
    users[nama_baru] = pwd_baru
    print("Registrasi berhasil! Silakan login.")
    input("Tekan Enter untuk melanjutkan...")
    continue

```

#### E. Lihat Event (Guest)

```

elif pilihan == '3':
    os.system('cls' if os.name == 'nt' else 'clear')
    print("\nDaftar Event:")
    for event in events:
        print(f"ID: {event['id']}, Nama: {event['nama']}, Hadiah:
{event['hadiah']}, Status: {event['status']}, Peserta: {'',
'.join(event['peserta'])}")
        input("\nTekan Enter untuk kembali ke menu...")
    continue

```

#### F. Keluar dari Program

```

if pilihan == '4':
    os.system('cls' if os.name == 'nt' else 'clear')
    print("Terima kasih telah menggunakan sistem ini.")
    sys.exit()

```

## G. Login sebagai Admin & Menu Admin

```
elif user in admin and admin[user] == pwd:
    print(f"Selamat datang ADMIN, {user}!")
    input("Tekan Enter untuk melanjutkan...")

    # Menu untuk admin
    while True:
        os.system('cls' if os.name == 'nt' else 'clear')
        print("\nMenu Admin:")
        print("1. Lihat Event")
        print("2. Create Event")
        print("3. Update Event")
        print("4. Delete Event")
        print("5. Kelola Peserta Event")
        print("6. Logout")
        choice = input("Pilih menu (1-6): ")
```

## H. Sub Fitur Admin

### 1. Lihat Event

```
if choice == '1':
    os.system('cls' if os.name == 'nt' else 'clear')
    print("\nDaftar Event:")
    for event in events:
        print(f"ID: {event['id']}, Nama: {event['nama']},
        Hadiah: {event['hadiah']}, Status: {event['status']}, Peserta: {'',
        '.join(event['peserta'])}")
    input("\nTekan Enter untuk kembali ke menu...")
```

### 2. Creat Event

```
Silakan coba lagi.")

    new_id = max(event['id'] for event in events) + 1 if
events else 1

    events.append({
        "id": new_id,
        "nama": nama_event,
        "hadiah": hadiah_event,
        "status": status_event,
        "peserta": []
    })
    print("Event berhasil ditambahkan!")
    input("\nTekan Enter untuk kembali ke menu...")
```



### 3. Update Event

```
elif choice == '3':
    os.system('cls' if os.name == 'nt' else 'clear')
    event_id = input("Masukkan ID event yang ingin diupdate: ")

    if event_id.isdigit():
        event_id = int(event_id)
        event_ditemukan = False

        for event in events:
            if event['id'] == event_id:
                event_ditemukan = True
                nama_event = input("Masukkan nama event
baru: ")
                hadiah_event = input("Masukkan hadiah event
baru: ")

                while True:
                    status_event = input("Masukkan status
event baru (ongoing/finished): ").lower()
                    if status_event in ["ongoing",
"finished"]:
                        break
                    else:
                        print("Status harus 'ongoing' atau
'finished'. Silakan coba lagi.")

                event['nama'] = nama_event
                event['hadiah'] = hadiah_event
                event['status'] = status_event
                print("Event berhasil diupdate!")
                break

        if not event_ditemukan:
            print("Event dengan ID tersebut tidak
ditemukan.")
        else:
            print("ID event harus berupa angka.")

    input("\nTekan Enter untuk kembali ke menu...")
```

### 4. Delete Event

```
elif choice == '4':
    os.system('cls' if os.name == 'nt' else 'clear')
```

```

        event_id = input("Masukkan ID event yang ingin dihapus: ")

    if event_id.isdigit():
        event_id = int(event_id)
        event_ditemukan = False

        for event in events:
            if event['id'] == event_id:
                event_ditemukan = True
                events.remove(event)
                print("Event berhasil dihapus!")
                break

        if not event_ditemukan:
            print("Event dengan ID tersebut tidak ditemukan.")
    else:
        print("ID event harus berupa angka.")

    input("\nTekan Enter untuk kembali ke menu...")

```

## 5. Kelola Peserta

```

elif choice == '5':
    os.system('cls' if os.name == 'nt' else 'clear')
    event_id = input("Masukkan ID event untuk kelola peserta: ")

    if event_id.isdigit():
        event_id = int(event_id)
        event_ditemukan = False

        for event in events:
            if event['id'] == event_id:
                event_ditemukan = True
                print(f"Peserta saat ini: {' '.join(event['peserta']) if event['peserta'] else 'Tidak ada peserta'}")
                print("1. Tambah Peserta")
                print("2. Hapus Peserta")
                sub_choice = input("Pilih opsi (1-2): ")

                if sub_choice == '1':

```

```

nama_peserta = input("Masukkan nama
peserta: ")
event['peserta'].append(nama_peserta)
print("Peserta berhasil ditambahkan!")
elif sub_choice == '2':
    nama_peserta = input("Masukkan nama
peserta yang ingin dihapus: ")
    if nama_peserta in event['peserta']:
event['peserta'].remove(nama_peserta)
        print("Peserta berhasil dihapus!")
    else:
        print("Peserta tidak ditemukan.")
    else:
        print("Opsi tidak valid.")
        break

    if not event_ditemukan:
        print("Event dengan ID tersebut tidak
ditemukan.")
    else:
        print("ID event harus berupa angka.")

    input("\nTekan Enter untuk kembali ke menu...")

```

## 6. Logout

```

elif choice == '6':
    print("Logout berhasil.")
    break

    else:
        print("Opsi tidak valid. Silakan coba lagi.")
        input("\nTekan Enter untuk kembali ke menu...")
    continue

```

#### 4. Hasil Output

```
=====
                        SISTEM MANAJEMEN EVENT WEB3 & HACKATHON
=====
1. Login
2. Register
3. Lihat Event (Guest)
4. Keluar
Pilih opsi (1-4): █
```

Gambar 4.1 Menu

```
Masukkan username: Bang Pernanda
Masukkan password: 123
Selamat datang, Bang Pernanda!
Tekan Enter untuk melanjutkan...█
```

Gambar 4.2 Login Sebagai User(Peserta)

```
Menu Pengguna - Bang Pernanda:
1. Lihat Event
2. Daftar Event
3. Logout
Pilih menu (1-3): █
```

Gambar 4.3 Menu User

```
Daftar Event yang Tersedia:
ID: 1, Nama: ETH Hackathon Samarinda, Hadiah: 10.000 USDT + NFT
Masukkan ID event yang ingin diikuti: 1
Berhasil mendaftar ke event ETH Hackathon Samarinda!

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

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...S
```

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...S
```

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/elkhuwas20/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).