

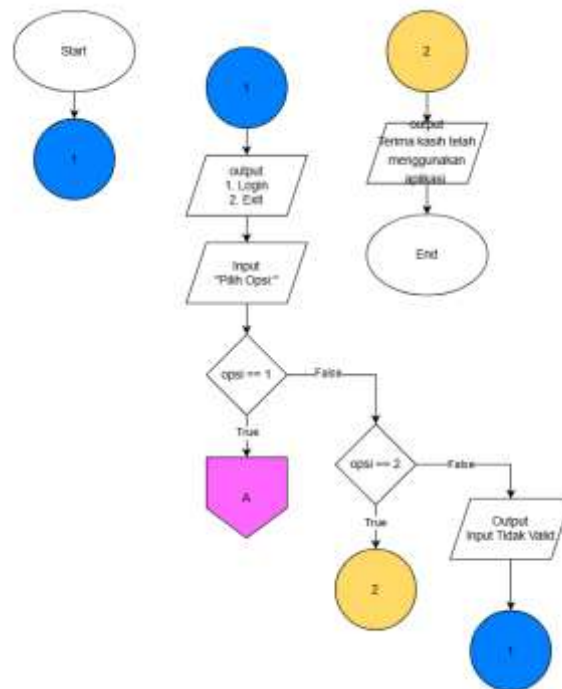
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



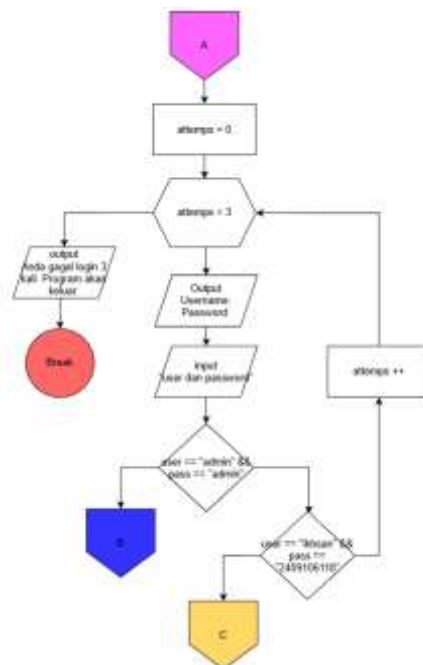
Disusun oleh:
Ikhsan (2409106118)
Kelas (C2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

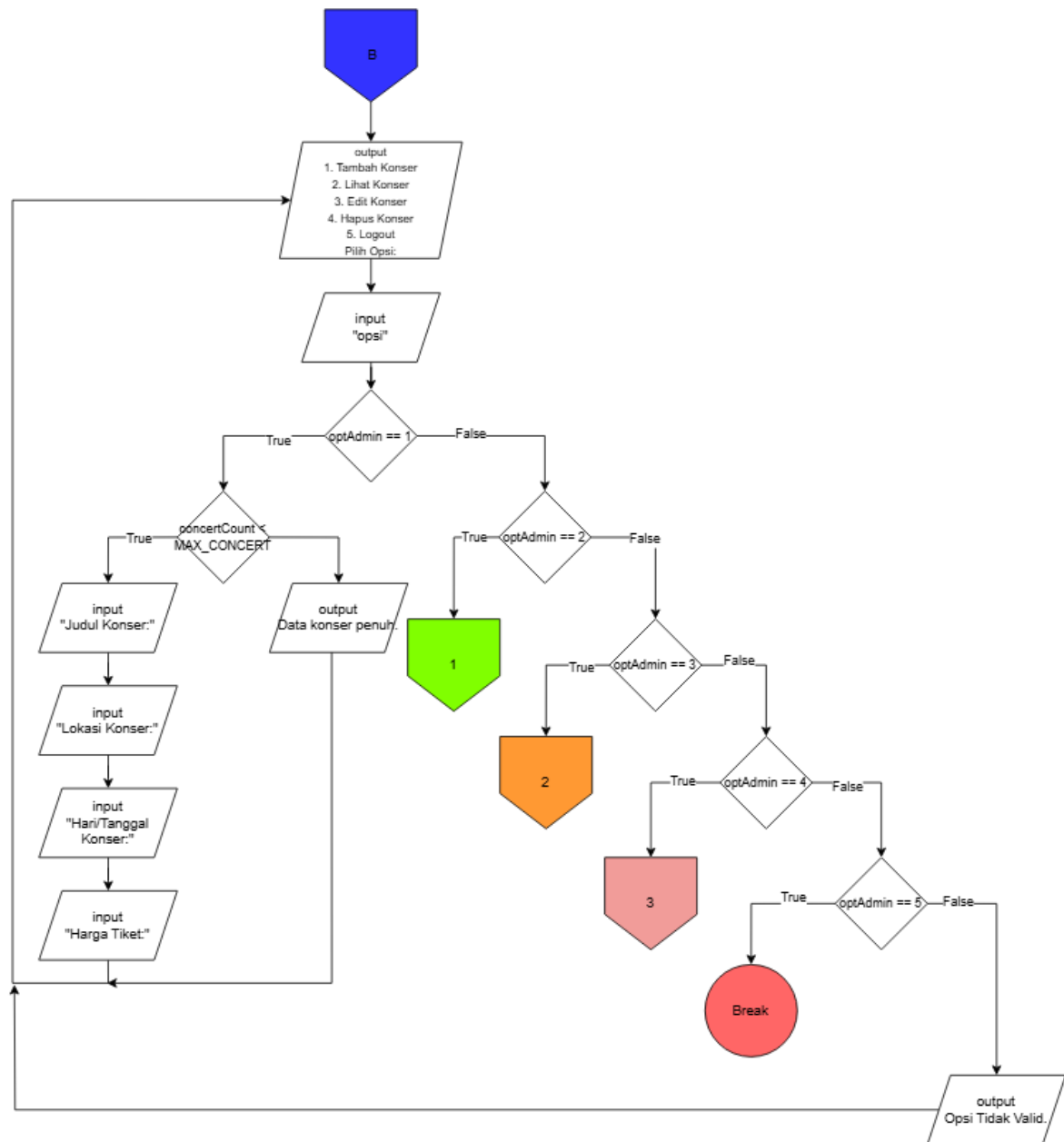
1. Flowchart



Gambar 1 Flowchart



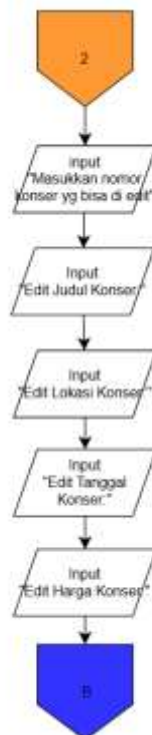
Gambar 2 Flowchart



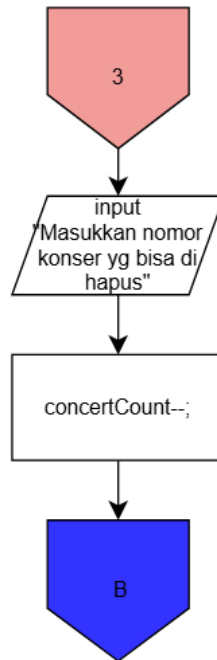
Gambar 3 Flowchart



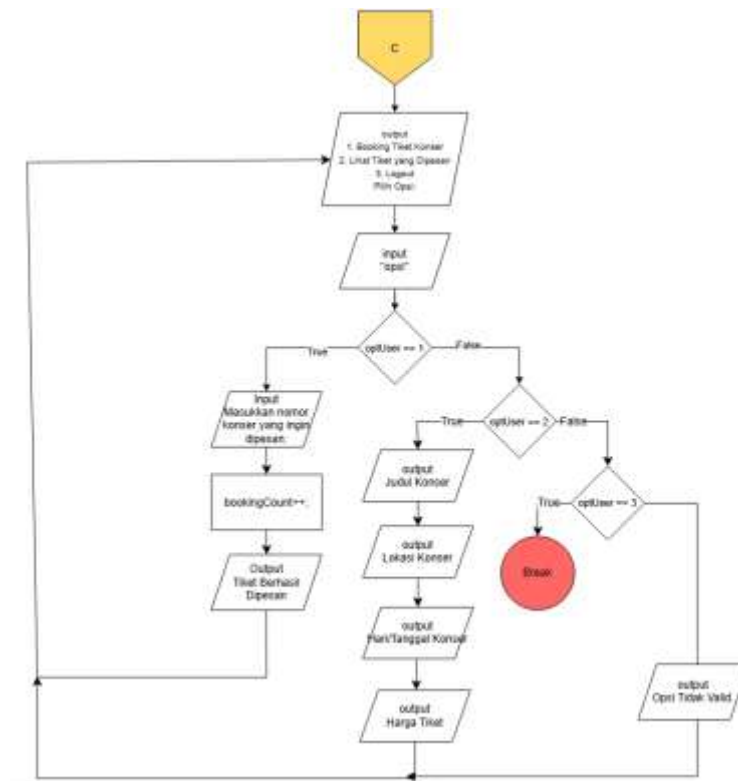
Gambar 4 Flowchart



Gambar 5 Flowchart



Gambar 6 Flowchart



Gambar 7 Flowchart

2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini merupakan aplikasi berbasis console yang digunakan untuk pemesanan tiket konser. Data konser disimpan dalam array dua dimensi yang memuat informasi seperti judul, lokasi, tanggal, dan harga tiket. Program ini mendukung dua akun:

Admin: (username: "admin", password: "admin")

Admin dapat menambah konser baru, melihat daftar konser, mengedit data konser, dan menghapus konser dari daftar.

Pengguna: (username: "Ikhsan", password: "2409106118")

Pengguna dapat melakukan pemesanan tiket konser dan melihat tiket-tiket yang telah dipesan, dengan tampilan data dalam format tabel yang rapi.

Fitur login membatasi percobaan maksimal tiga kali. Jika login gagal tiga kali, program akan keluar. Setelah login berhasil, menu yang sesuai dengan jenis akun akan ditampilkan dan pengguna dapat memilih opsi yang diinginkan hingga memilih untuk logout atau keluar dari aplikasi.

2.2 Penjelasan Alur & Algoritma

1. Pendahuluan

Program ini adalah aplikasi sederhana pemesanan tiket konser yang dijalankan di konsol. Data konser disimpan dalam sebuah array dua dimensi (dengan batas maksimal 100 entri), yang setiap barisnya menyimpan informasi konser berupa judul, lokasi, tanggal, dan harga tiket. Selain itu, terdapat array lain untuk menyimpan tiket yang telah dipesan oleh pengguna. Program mendukung dua jenis akun:

Admin (username: "admin", password: "admin")

Pengguna (username: "Ikhsan", password: "2409106118")

Program juga membatasi percobaan login maksimal tiga kali. Setelah login berhasil, menu yang muncul berbeda sesuai dengan jenis akun yang digunakan.

2. Alur Program

Menu Utama:

Program menampilkan menu utama dengan opsi Login atau Exit.

- a) Jika pengguna memilih Exit, program langsung berhenti.
- b) Jika memilih Login, program meminta input username dan password.

Proses Login dengan Batas 3 Kali:

Program memberikan kesempatan hingga tiga kali percobaan.

- a) Jika input tidak sesuai dengan salah satu akun yang valid, maka akan muncul pesan kesalahan dan diberi tahu berapa kali tersisa.
- b) Jika gagal tiga kali, program akan menampilkan pesan dan keluar.

Setelah Login Berhasil:**Jika akun adalah Admin:**

Program menampilkan menu Admin yang memiliki opsi:

a) Tambah Konser

Admin memasukkan data konser baru (judul, lokasi, tanggal, dan harga). Data baru akan ditambahkan ke array konser.

b) Lihat Konser

Program menampilkan daftar konser dalam format tabel sederhana.

c) Edit Konser

Admin memilih nomor konser yang ingin diubah, lalu memasukkan data baru untuk mengedit konser tersebut.

d) Hapus Konser

Admin memilih nomor konser yang ingin dihapus. Data konser yang dihapus akan digeser agar tidak ada celah pada array.

e) Logout

Kembali ke menu utama.

Jika akun adalah Pengguna (Ikhsan):

Menu pengguna memiliki opsi:

a) Booking Tiket Konser

Program menampilkan daftar konser. Pengguna memilih nomor konser yang ingin dipesan, kemudian data konser tersebut disalin ke array booking dan jumlah tiket dipesan bertambah.

b) Lihat Tiket yang Dipesan

Program menampilkan data tiket yang telah dipesan dalam format tabel yang rapi.

c) Logout

Kembali ke menu utama.

Loop Utama:

Program akan kembali menampilkan menu utama setelah logout dari salah satu menu, sehingga pengguna bisa mencoba login lagi atau keluar dari program.

3. Source Code

```
#include <iostream>
using namespace std;

#define MAX_CONCERT 100 // Ukuran maksimum array konser
#define INFO 4 // Kolom: {judul, lokasi, tanggal, harga}
#define MAX_BOOKING 100 // Ukuran maksimum array booking

int main() {
    // Data konser: setiap baris berisi {judul, lokasi, tanggal, harga}
    string concerts[MAX_CONCERT][INFO] = {
        {"Midnighth Serenade", "Samarinda", "11-Oktober-2024", "200000"},
        {"Arctic Monkey", "Balikpapan", "12-Oktober-2024", "500000"}
    };
    int concertCount = 2;

    // Array untuk menyimpan tiket yang dipesan oleh pengguna (khusus untuk
    akun Ikhsan)
    string bookings[MAX_BOOKING][INFO];
    int bookingCount = 0;

    while (true) {
        cout << "\n\n=====\\n";
        cout << "            Pemesanan Tiket Konser\\n";
        cout << "    Pilih opsi: 1. Login    2. Exit\\n";
        cout << "=====\\n";
        cout << "Pilih opsi: ";
        int opsi;
        cin >> opsi;
        cin.ignore(); // bersihkan newline
        cout << "\\n";

        if (opsi == 1) {
            // Batas login maksimal 3 kali
            int attempts = 0;
            bool loggedIn = false;
            string user, pass;
            while (attempts < 3 && !loggedIn) {
                cout << "Username: ";
                cin >> user;
                cout << "Password: ";
                cin >> pass;
                cin.ignore();
                if ((user == "admin" && pass == "admin") ||
                    (user == "Ikhsan" && pass == "2409106118"))
                    loggedIn = true;
                else {
```



```

                                cout << "Edit Harga Tiket (" <<
concerts[idx][3] << "): ";
                                getline(cin, concerts[idx][3]);
                                cout << "\nKonser berhasil diubah.\n";
                                }
                                }
                                }
                                else if (optAdmin == 4) { // Hapus Konser
                                    if (concertCount == 0)
                                        cout << "\nTidak ada konser yang bisa dihapus.\n";
                                    else {
                                        cout << "\nDaftar Konser:\n";
                                        for (int i = 0; i < concertCount; i++) {
                                            cout << i+1 << ". " << concerts[i][0] << "\n";
                                        }
                                        cout << "Masukkan nomor konser yang ingin dihapus:
";

                                        int num;
                                        cin >> num;
                                        cin.ignore();
                                        if (num < 1 || num > concertCount)
                                            cout << "Nomor tidak valid.\n";
                                        else {
                                            int idx = num - 1;
                                            for (int i = idx; i < concertCount - 1; i++) {
                                                concerts[i][0] = concerts[i+1][0];
                                                concerts[i][1] = concerts[i+1][1];
                                                concerts[i][2] = concerts[i+1][2];
                                                concerts[i][3] = concerts[i+1][3];
                                            }
                                            concertCount--;
                                            cout << "\nKonser berhasil dihapus.\n";
                                        }
                                    }
                                }
                                else if (optAdmin == 5) { // Logout
                                    break;
                                }
                                else {
                                    cout << "\nOpsi tidak valid.\n";
                                }
                                cout << "\nTekan Enter untuk melanjutkan...";
                                cin.ignore();
                                cin.get();
                            }
                            else if (user == "Ikhsan" && pass == "2409106118") { // Menu
Pengguna

```



```

        cout << "|" << bookings[i][0];
        int sp = 20 - bookings[i][0].length();
        for (int j = 0; j < sp; j++) cout << " ";
        cout << "|" << bookings[i][1];
        sp = 14 - bookings[i][1].length();
        for (int j = 0; j < sp; j++) cout << " ";
        cout << "|" << bookings[i][2];
        sp = 20 - bookings[i][2].length();
        for (int j = 0; j < sp; j++) cout << " ";
        cout << "|" << bookings[i][3];
        sp = 10 - bookings[i][3].length();
        for (int j = 0; j < sp; j++) cout << " ";
        cout << "|\\n";
    }
    cout << "+-----+-----+-----+\\n";
+-----+-----+-----+\\n";
    }
}
else if (optUser == 3) {
    break;
}
else {
    cout << "\\nOpsi tidak valid.\\n";
}
cout << "\\nTekan Enter untuk melanjutkan...";
cin.ignore();
cin.get();
}
}
else {
    cout << "\\nLogin gagal. Username atau password salah.\\n";
    cout << "\\nTekan Enter untuk melanjutkan...";
    cin.ignore();
    cin.get();
}
}
else if (opsi == 2) {
    cout << "\\nTerima kasih telah menggunakan aplikasi.\\n";
    break;
}
else {
    cout << "\\nInput tidak valid.\\n";
    cout << "\\nTekan Enter untuk melanjutkan...";
    cin.ignore();
    cin.get();
}
}
return 0;

```

```
}
```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

Skenario 1.1: Login Berhasil (Pengguna)

- **Input:**
 - Username: "Ikhsan"
 - Password: "2409106118"
- **Output yang diharapkan:**
 - Pesan bahwa login berhasil dan kemudian muncul menu pengguna dengan opsi:
 1. Booking Tiket Konser
 2. Lihat Tiket yang Dipesan
 3. Logout

Skenario 1.2: Login Berhasil (Admin)

- **Input:**
 - Username: "admin"
 - Password: "admin"
- **Output yang diharapkan:**
 - Pesan bahwa login berhasil dan kemudian muncul menu admin dengan opsi:
 1. Tambah Konser
 2. Lihat Konser
 3. Edit Konser
 4. Hapus Konser
 5. Logout

Skenario 1.3: Login Gagal

- **Input:**
 - Misalnya: Username: "Admin", Password: "1234"
- **Output yang diharapkan:**
 - Setelah percobaan pertama:
"Login gagal. Coba lagi (2 kali tersisa)."
 - Setelah percobaan kedua:
"Login gagal. Coba lagi (1 kali tersisa)."
 - Setelah percobaan ketiga:
"Anda gagal login 3 kali. Program akan keluar."
 - Program kemudian berhenti.

Skenario 2.1: Tambah Konser (Admin)

- **Prasyarat:**
Login berhasil sebagai admin.
- **Input:**
 - Pilih opsi 1 pada menu admin (Tambah Konser).
 - Masukkan data konser:
 - Judul: "Rock Fest"
 - Lokasi: "Jakarta"
 - Hari/Tanggal: "20-November-2024"
 - Harga: "350000"
- **Output yang diharapkan:**
 - Pesan: "Konser berhasil ditambahkan."
 - Jika kemudian memilih opsi *Lihat Konser*, daftar konser akan memuat data baru tersebut.

Skenario 2.2: Edit Konser (Admin)

- **Prasyarat:**
Login sebagai admin dan setidaknya terdapat satu konser di daftar.
- **Input:**
 - Pilih opsi 3 pada menu admin (Edit Konser).
 - Tampilkan daftar konser, misalnya nomor 1 dipilih.
 - Masukkan data baru untuk konser tersebut, misalnya:
 - Judul: "Midnight Serenade"
 - Lokasi: "Samarinda"
 - Hari/Tanggal: "11-Oktober-2024"
 - Harga: "210000"
- **Output yang diharapkan:**
 - Pesan: "Konser berhasil diubah."
 - Jika memilih opsi *Lihat Konser*, data konser nomor 1 akan diperbarui.

Skenario 2.3: Hapus Konser (Admin)

- **Prasyarat:**
Login sebagai admin dan daftar konser tidak kosong.
- **Input:**
 - Pilih opsi 4 pada menu admin (Hapus Konser).
 - Tampilkan daftar konser dan pilih nomor konser yang akan dihapus (misalnya nomor 2).
- **Output yang diharapkan:**
 - Pesan: "Konser berhasil dihapus."
 - Data konser tersebut tidak muncul lagi di daftar konser.

Skenario 3.1: Booking Tiket (Pengguna)

- **Prasyarat:**
Login berhasil sebagai pengguna (username: "Ikhsan").
- **Input:**
 - Pilih opsi 1 pada menu pengguna (Booking Tiket Konser).
 - Tampilkan daftar konser dan pilih nomor konser (misalnya nomor 1).
- **Output yang diharapkan:**
 - Pesan: "Tiket berhasil dipesan."
 - Data tiket dari konser nomor 1 disalin ke daftar tiket yang dipesan.

Skenario 3.2: Lihat Tiket yang Dipesan (Pengguna)

- **Prasyarat:**
Telah ada tiket yang dipesan (misalnya dari Skenario 3.1).
- **Input:**
 - Pilih opsi 2 pada menu pengguna (Lihat Tiket yang Dipesan).
- **Output yang diharapkan:**
 - Program menampilkan tabel yang rapi berisi data tiket yang telah dipesan, lengkap dengan header kolom:
 - Judul Konser
 - Lokasi Konser
 - Hari/Tanggal Konser
 - Harga Tiket

4.2 Hasil Output

```
=====
                Pemesanan Tiket Konser
        Pilih opsi: 1. Login   2. Exit
=====
Pilih opsi: 1

Username: Ikhsan
Password: 2409106118

===== Menu Pengguna =====
1. Booking Tiket Konser
2. Lihat Tiket yang Dipesan
3. Logout
=====
Pilih opsi: █
```

Gambar 4.2.1 Pengujian Login Pengguna


```

          Pemesanan Tiket Konser
        Pilih opsi: 1. Login   2. Exit
        =====
        Pilih opsi: 1

        Username: admin
        Password: admin

        ===== Menu Admin =====
        1. Tambah Konser
        2. Lihat Konser
        3. Edit Konser
        4. Hapus Konser
        5. Logout
        =====
        Pilih opsi: █

```

Gambar 4.2.2 Pengujian Login Admin

```

          Pemesanan Tiket Konser
        Pilih opsi: 1. Login   2. Exit
        =====
        Pilih opsi: 1

        Username: Rifky
        Password: Jamal

        Login gagal. Coba lagi (2 kali tersisa).
        Username: IkhsanGantengBanget
        Password: 12345

        Login gagal. Coba lagi (1 kali tersisa).
        Username: Ikhsan Mau Jadi Aslab
        Password:
        Anda gagal login 3 kali. Program akan keluar.
        PS D:\GITHUB\Praktikum-An1\Post-test\Post-test-2> █

```

Gambar 4.2.3 Pengujian Login Gagal

```

        Pilih opsi: 1

        === Tambah Konser ===
        Judul Konser: Rock Fest
        Lokasi Konser: Jakarta
        Hari/Tanggal Konser: 20-November-2025
        Harga Tiket: 350000

        Konser berhasil ditambahkan.

        Tekan Enter untuk melanjutkan... █

```

Gambar 4.2.4 Pengujian Tambah Konser Admin

```
Pilih opsi: 2

----- Daftar Konser -----
+-----+-----+-----+-----+
| Judul Konser | Lokasi Konser | Hari/Tanggal Konser | Harga Tiket |
+-----+-----+-----+-----+
| Midnigth Serenade | Samarinda | 11-Okttober-2024 | 200000 |
| Arctic Monkey | Balikpapan | 12-Okttober-2024 | 500000 |
| Rock Fest | Jakarta | 20-November-2025 | 350000 |
+-----+-----+-----+-----+

Tekan Enter untuk melanjutkan...
```

Gambar 4.2.5 Pengujian Lihat Konser Admin

```
Pilih opsi: 3

Daftar Konser:
1. Midnigth Serenade
2. Arctic Monkey
3. Rock Fest
Masukkan nomor konser yang ingin diedit: 3
Edit Judul Konser (Rock Fest): Midnight Serenade2
Edit Lokasi Konser (Jakarta): Samarinda
Edit Hari/Tanggal (20-November-2025): 5-April-2025
Edit Harga Tiket (350000): 200000

Konser berhasil diubah.

Tekan Enter untuk melanjutkan...
```

Gambar 4.2.6 Pengujian Edit Konser Admin

```
Pilih opsi: 4

Daftar Konser:
1. Midnigth Serenade
2. Arctic Monkey
3. Midnight Serenade2
Masukkan nomor konser yang ingin dihapus: 3

Konser berhasil dihapus.

Tekan Enter untuk melanjutkan...
```

Gambar 4.2.7 Pengujian Hapus Konser Admin

```

Pilih opsi: 1

----- Daftar Konser -----
1. Midnight Serenade | Samarinda | 11-Oktober-2024 | 200000
2. Arctic Monkey | Balikpapan | 12-Oktober-2024 | 500000

Masukkan nomor konser yang ingin dipesan: 1

Tiket berhasil dipesan.

Tekan Enter untuk melanjutkan...

```

Gambar 4.2.8 Pengujian Booking Konser Pengguna

```

Pilih opsi: 2

===== Tiket yang Dipesan =====
+-----+-----+-----+-----+
| Judul Konser      | Lokasi Konser | Hari/Tanggal Konser | Harga Tiket |
+-----+-----+-----+-----+
| Midnight Serenade | Samarinda     | 11-Oktober-2024     | 200000      |
+-----+-----+-----+-----+

Tekan Enter untuk melanjutkan...

```

Gambar 4.2.9 Pengujian Lihat Konser Pengguna

5. Git

```
ASUS@DESKTOP-8AN5J09 MINGW64 /d/GITHUB/Praktikum-Apl (main)
$ git add .

ASUS@DESKTOP-8AN5J09 MINGW64 /d/GITHUB/Praktikum-Apl (main)
$ git commit -m "Exe Posttest2 Finish"
[main 4bcc389] Exe Posttest2 Finish
1 file changed, 0 insertions(+), 0 deletions(-)

ASUS@DESKTOP-8AN5J09 MINGW64 /d/GITHUB/Praktikum-Apl (main)
$ git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 545 bytes | 272.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Ikhsan-118/Praktikum-Apl.git
   d4940c5..4bcc389  main -> main
branch 'main' set up to track 'origin/main'.

ASUS@DESKTOP-8AN5J09 MINGW64 /d/GITHUB/Praktikum-Apl (main)
$
```

Gambar 5 Terminal Git Bash

1. **git add .**

Perintah ini menambahkan semua perubahan (file baru atau perubahan pada file) ke staging area.

2. **git commit -m "Exe Posttest2 Finish"**

Perintah ini membuat commit baru dengan pesan “Exe Posttest2 Finish”. Commit berisi snapshot perubahan yang sebelumnya telah di-*add*.

3. **git push -u origin main**

Perintah ini mengirim (push) commit yang baru saja dibuat ke repositori jarak jauh (remote), yakni origin, pada cabang (branch) main.

Opsi -u menandakan bahwa branch lokal main diatur untuk melacak branch main pada remote origin.

Dari tangkapan layar, terlihat bahwa proses push berjalan sukses, objek-objek commit telah dikirim ke GitHub, dan branch lokal main sekarang terhubung dengan branch main di remote (origin).