

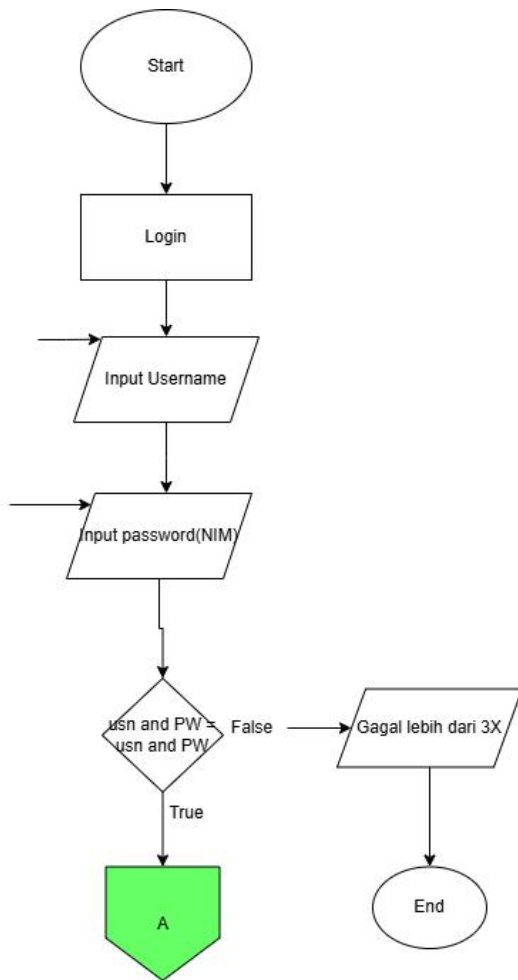
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



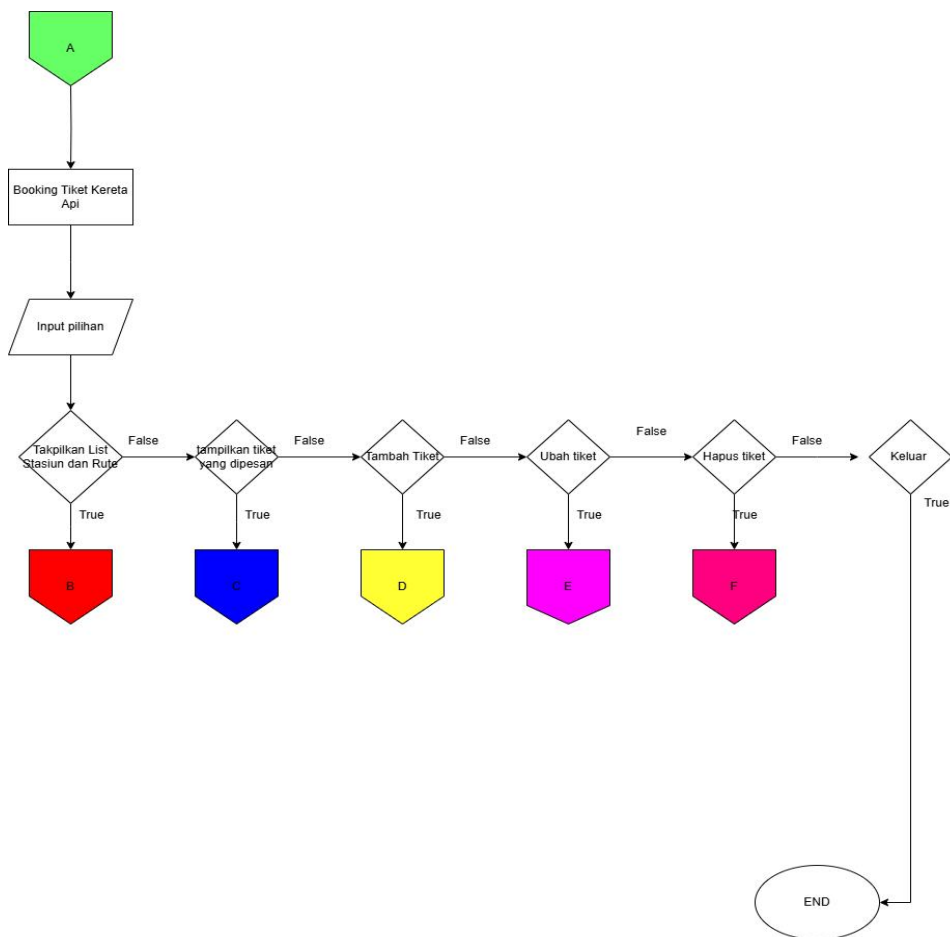
Disusun oleh:
Sayid Annashir Ikhwan (2409106097)
Kelas (C '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

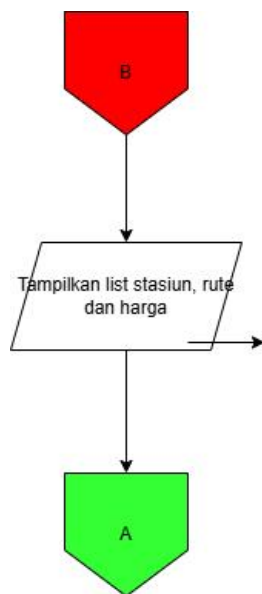
1. Flowchart



Gambar 1.1. Flowchart



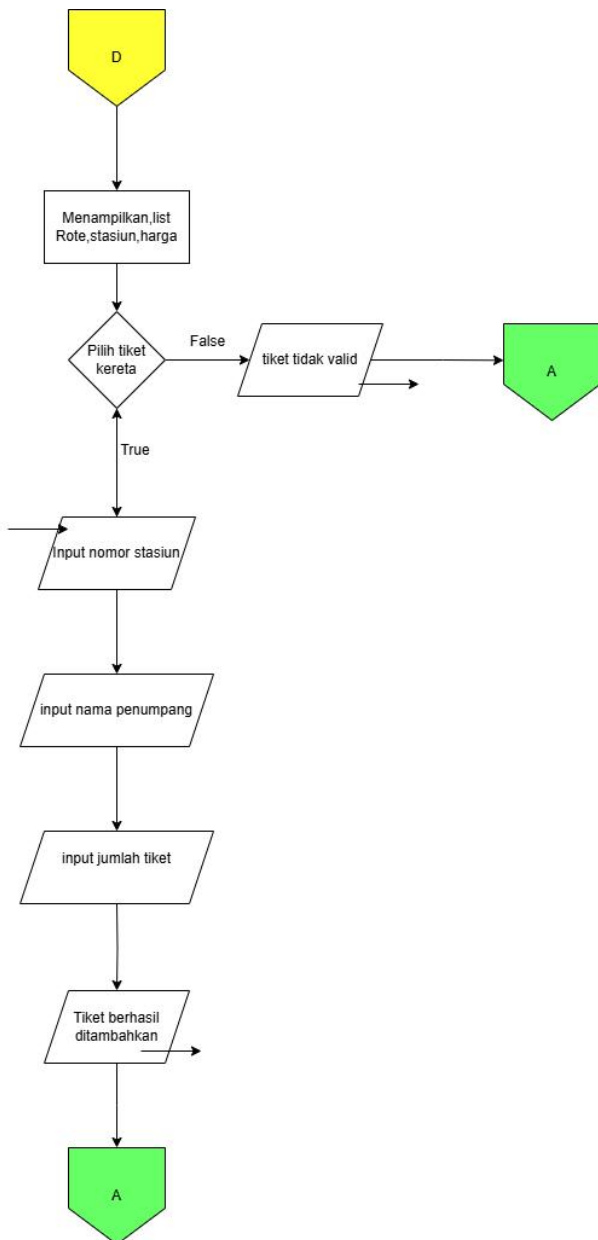
Gambar 1.2. <flowchart tampilan menu utama>



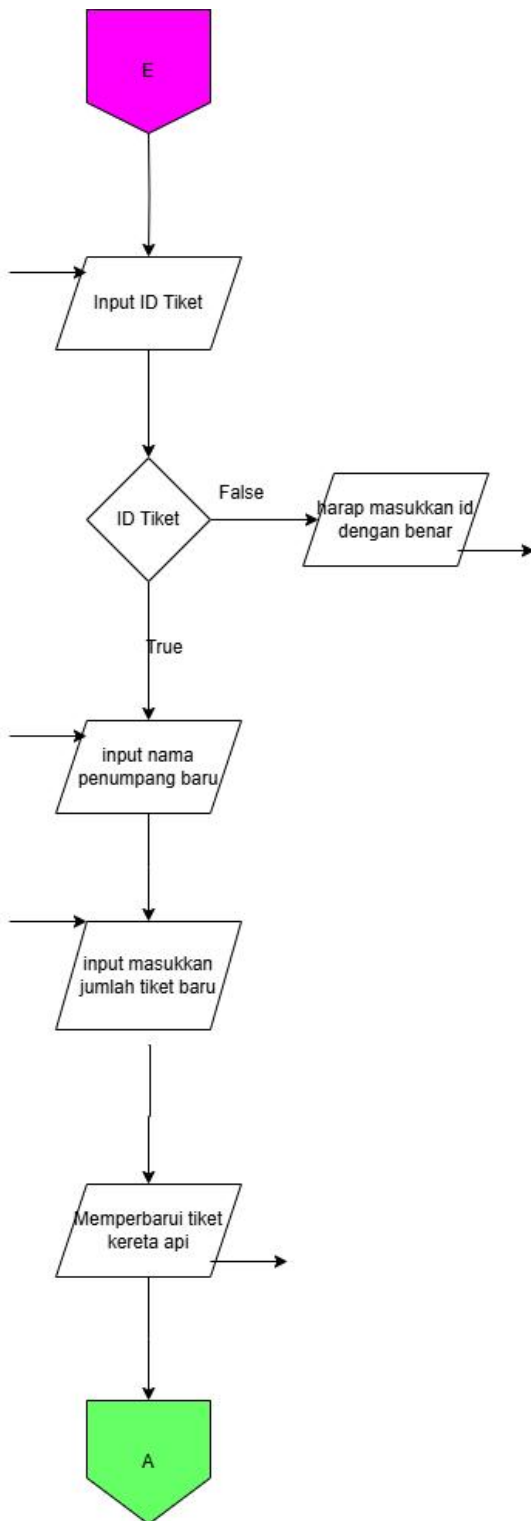
Gambar 1.3. <manampilkan list stasiun>



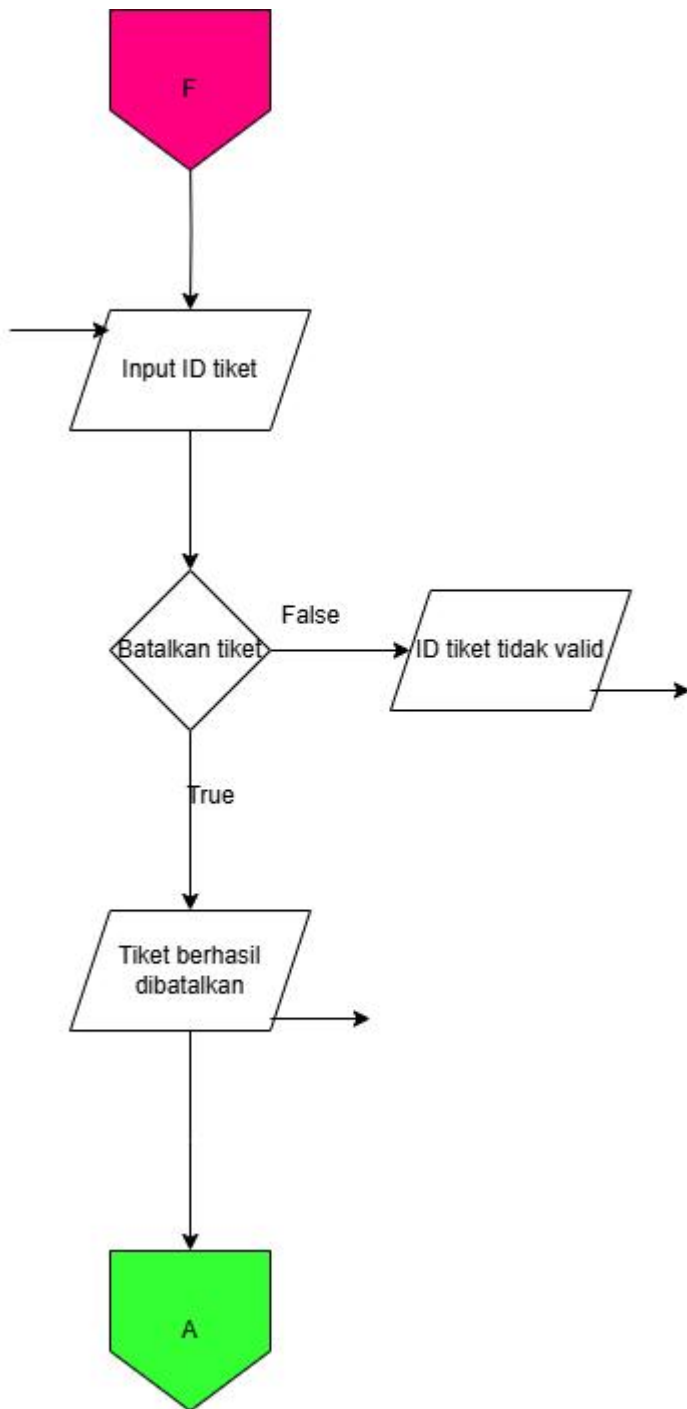
Gambar 1.4 menampilkan tiket yang sudah dipesan



Gambar 1.5 Tambah tiket



Gambar 1.6 ubah tiket



Gambar 1.7 Hapus tiket

2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini adalah sebuah sistem pemesanan tiket kereta api berbasis konsol yang memungkinkan pengguna untuk melakukan pemesanan tiket, melihat daftar stasiun dan rute, serta mengelola tiket yang telah dipesan. Program ini dirancang untuk memenuhi kebutuhan pengguna dalam memesan tiket kereta api dengan antarmuka yang sederhana dan mudah digunakan.

2.2 Penjelasan Alur & Algoritma

Login:

- Bandingkan input nama dan NIM dengan data valid.
- Jika benar, lanjut ke menu utama. Jika salah 3 kali, program berhenti.

Tampilkan List Stasiun dan Rute:

- Loop melalui array stasiun, rute, stok, dan harga, lalu tampilkan informasinya.

Tambah Tiket:

- Pilih stasiun, input nama penumpang dan jumlah tiket.
- Jika stok cukup, simpan data tiket dan kurangi stok.

Tampilkan Tiket yang Dipesan:

Jika ada tiket, tampilkan detailnya. Jika tidak, tampilkan pesan "Belum ada tiket."

Ubah Tiket:

- Pilih tiket, input nama penumpang baru atau jumlah tiket baru.
- Update data tiket dan hitung ulang total harga.

Hapus Tiket:

- Pilih tiket, hapus data tiket, dan geser data yang tersisa.

Keluar:

- Tampilkan pesan terima kasih dan hentikan program.

3. Source Code

A. Fitur login.

Program akan dimulai dengan login. memasukkan username dan password yang harus sesuai dengan yang ada di database yaitu “Sayid Annashir Ikhwan” untuk Username dan “2409106097” untuk Passwordnya. Jika kita memasukkan login salah lebih dari 3x, maka program akan berhenti

```
string namaUser, nimUser;
int loginAttempts[1] = {0}; // Menggunakan array untuk loginAttempts
int loggedIn[1] = {0}; // Menggunakan array untuk loggedIn (0 = false, 1 = true)

// Login System
while (loginAttempts[0] < 3 && loggedIn[0] == 0) {
    cout << "Masukkan Nama: ";
    getline(cin, namaUser);
    cout << "Masukkan NIM: ";
    getline(cin, nimUser);
    if (namaUser == "Sayid Annashir Ikhwan" && nimUser == "2409106097") {
        loggedIn[0] = 1; // Login berhasil
        cout << "Hore Login berhasil!" << endl;
    } else {
        loginAttempts[0]++;
        cout << "Maaf Login gagal. Percobaan " << loginAttempts[0] << " dari 3." << endl;
    }
}
if (loggedIn[0] == 0) {
    cout << "Anda telah gagal login 3 kali. Program berhenti." << endl;
    return 0;
}
```

Gambar 3,1

B. Fitur Menu Utama

bagian pertama yang muncul setelah pengguna berhasil login. Di sini, pengguna diberikan beberapa pilihan untuk melakukan Pemesanan/booking tiket kereta api atau keluar dari program. Pilihan yang tersedia adalah:

1. Tampilkan list stasiun dan Rute
2. Tampilkan tiket yaang dipesan
3. Tambah tiket
4. Ubah tiket
5. Hapus Tiket
6. Keluar

```

cout << "===== " << endl;
cout << "Selamat datang di Program Booking Tiket Kereta Api" << endl;
cout << "===== " << endl;
cout << "1. Tampilkan List Stasiun dan Rute" << endl;
cout << "2. Tampilkan Tiket yang Dipesan" << endl;
cout << "3. Tambah Tiket" << endl;
cout << "4. Ubah Tiket" << endl;
cout << "5. Hapus Tiket" << endl;
cout << "6. Keluar" << endl;
cout << "Pilihan: ";
cin >> pilihan;
cin.ignore();

```

Gambar 3,2

C. Fitur Tampilkan List Stasiun dan Rute

Fitur ini berfungsi untuk menampilkan daftar stasiun yang tersedia beserta rute perjalanan, jumlah tiket yang masih tersedia, serta harga tiket. Fitur ini dibuat agar pengguna dapat melihat informasi tiket sebelum melakukan pemesanan.

Fitur ini dapat diakses melalui menu utama dengan memilih opsi 1. Tampilkan List Stasiun dan Rute. Fitur ini diimplementasikan menggunakan array untuk menyimpan data stasiun, rute, jumlah tiket, dan harga tiket. Selanjutnya, digunakan perulangan for untuk mencetak informasi dari setiap stasiun ke layar.

```

switch (pilihan) {
    case 1: // Tampilkan List Stasiun dan Rute
        cout << "Daftar Stasiun dan Rute:" << endl;
        cout << "-----" << endl;
        for (int i = 0; i < JUMLAH_STASIUN; i++) {
            cout << "Stasiun: " << stasiun[i] << endl;
            cout << "Rute: " << rute[i] << endl;
            cout << "Jumlah Tiket Tersedia: " << jumlahTiketStasiun[i] << endl;
            cout << "Harga: Rp " << harga[i] << endl;
            cout << "-----" << endl;
        }
        break;
}

```

Gambar 3,3

D. Tampilkan Tiket yang dipesan

Fitur ini digunakan untuk menampilkan daftar tiket yang telah dipesan oleh pengguna. Jika pengguna belum memesan tiket, maka program akan menampilkan pesan bahwa belum ada tiket yang dipesan.

Fitur ini dapat diakses melalui menu utama dengan memilih opsi **2. Tampilkan Tiket yang Dipesan.**

Fitur ini bekerja dengan memeriksa apakah ada tiket yang telah dipesan menggunakan variabel panjang[0] yang menyimpan jumlah tiket yang dipesan. Jika tidak ada tiket yang tersimpan, maka akan ditampilkan pesan bahwa belum ada tiket yang dipesan. Jika ada tiket yang dipesan, maka program akan menampilkan daftar tiket tersebut dengan informasi lengkap.

```

case 2: // Tampilkan Tiket yang Dipesan
    if (panjang[0] == 0) {
        cout << "Belum ada tiket yang dipesan." << endl;
    } else {
        cout << "Daftar Tiket yang Dipesan:" << endl;
        cout << "-----" << endl;
        for (int i = 0; i < panjang[0]; i++) {
            cout << "Tiket ke-" << i + 1 << ":" << endl;
            cout << "Nama Penumpang: " << namaPenumpang[i] << endl;
            cout << "Stasiun: " << stasiun[stasiunDipilih[i]] << endl;
            cout << "Rute: " << rute[stasiunDipilih[i]] << endl;
            cout << "Jumlah Tiket: " << jumlahTiket[i] << endl;
            cout << "Total Harga: Rp " << totalHarga[i] << endl;
            cout << "-----" << endl;
        }
    }
    break;

```

Gambar 3,4

E. Tambah Tiket

Fitur Tambah Tiket memungkinkan pengguna untuk memesan tiket kereta api dengan memasukkan nama penumpang, memilih stasiun keberangkatan, dan menentukan jumlah tiket yang ingin dibeli. Jika jumlah tiket yang tersedia mencukupi, tiket akan ditambahkan ke dalam daftar pemesanan, dan stok tiket akan diperbarui.

Fitur ini dapat diakses melalui menu utama dengan memilih opsi 3. Tambah Tiket.

```

case 3: // Tambah Tiket
    if (panjang[0] < MAX_TIKET) {
        cout << "Pilih Stasiun:" << endl;
        for (int i = 0; i < JUMLAH_STASIUN; i++) {
            cout << i + 1 << ". " << stasiun[i] << " (" << rute[i] << ")" << endl;
        }
        cout << "Masukkan nomor stasiun: ";
        cin >> index;
        cin.ignore(); // Membersihkan buffer

        if (index > 0 && index <= JUMLAH_STASIUN) {
            if (jumlahTiketStasiun[index - 1] > 0) {
                cout << "Masukkan Nama Penumpang: ";
                getline(cin, namaPenumpang[panjang[0]]);
                cout << "Masukkan Jumlah Tiket: ";
                cin >> jumlahTiket[panjang[0]];
                if (jumlahTiket[panjang[0]] <= jumlahTiketStasiun[index - 1]) {
                    stasiunDipilih[panjang[0]] = index - 1;
                    totalHarga[panjang[0]] = jumlahTiket[panjang[0]] * harga[index - 1];
                    jumlahTiketStasiun[index - 1] -= jumlahTiket[panjang[0]];
                    panjang[0]++;
                    cout << "Tiket berhasil ditambahkan!" << endl;
                } else {
                    cout << "Jumlah tiket tidak mencukupi." << endl;
                }
            } else {
                cout << "Tiket untuk stasiun ini sudah habis." << endl;
            }
        } else {
            cout << "Nomor stasiun tidak benar. Harap masukkan yang betul." << endl;
        }
    }
}

```

```

    } else {
        cout << "Kapasitas tiket penuh. Tidak bisa menambah tiket lagi." << endl;
    }
    break;

```

Gambar 3,5

F. Ubah Tiket

Fitur Ubah Tiket memungkinkan pengguna untuk mengubah data tiket yang sudah dipesan, seperti nama penumpang, stasiun keberangkatan, atau jumlah tiket. Fitur ini memastikan bahwa perubahan hanya dapat dilakukan jika tiket masih tersedia dan tidak melebihi kapasitas yang tersedia di stasiun tujuan.

Fitur ini dapat diakses melalui menu utama dengan memilih opsi 4. Ubah Tiket. Fitur ini bekerja dengan meminta pengguna memasukkan nomor tiket yang ingin diubah, kemudian memberikan opsi untuk mengedit nama, stasiun, atau jumlah tiket.

```

case 4: // Ubah Tiket
    if (panjang[0] == 0) {
        cout << "Belum ada tiket untuk diubah." << endl;
    } else {
        cout << "Daftar Tiket yang Dipesan:" << endl;
        cout << "-----" << endl;
        for (int i = 0; i < panjang[0]; i++) {
            cout << "Tiket ke-" << i + 1 << ":" << endl;
            cout << "Nama Penumpang: " << namaPenumpang[i] << endl;
            cout << "Stasiun: " << stasiun[stasiunDipilih[i]] << endl;
            cout << "Rute: " << rute[stasiunDipilih[i]] << endl;
            cout << "Jumlah Tiket: " << jumlahTiket[i] << endl;
            cout << "Total Harga: Rp " << totalHarga[i] << endl;
            cout << "-----" << endl;
        }
        cout << "Masukkan nomor tiket yang akan diubah: ";
        cin >> index;
        cin.ignore(); // Membersihkan buffer

        if (index > 0 && index <= panjang[0]) {
            cout << "Masukkan Nama Penumpang baru: ";
            getline(cin, namaPenumpang[index - 1]);
            cout << "Masukkan Jumlah Tiket baru: ";
            cin >> jumlahTiket[index - 1];
            cin.ignore(); // Membersihkan buffer
            totalHarga[index - 1] = jumlahTiket[index - 1] * harga[stasiunDipilih[index - 1]];
            cout << "Tiket berhasil diubah!" << endl;
        } else {
            cout << "Nomor tiket tidak valid." << endl;
        }
    }
    break;

```

Gambar 3,6

G. Hapus Tiket

Fitur Hapus Tiket memungkinkan pengguna untuk membatalkan tiket yang sudah dipesan. Dengan fitur ini, tiket yang telah dipesan dapat dihapus dari daftar pemesanan, dan jumlah tiket yang sebelumnya terpakai akan dikembalikan ke stok tiket yang tersedia di stasiun keberangkatan.

Fitur ini dapat diakses melalui menu utama dengan memilih opsi 5. Hapus Tiket. Fitur ini bekerja dengan menampilkan daftar tiket yang telah dipesan, meminta pengguna memilih tiket yang ingin dihapus, lalu menghapus tiket tersebut dari daftar pemesanan.

```
case 5: // Hapus Tiket
    if (panjang[0] == 0) {
        cout << "Belum ada tiket untuk dihapus." << endl;
    } else {
        cout << "Daftar Tiket yang Dipesan:" << endl;
        cout << "-----" << endl;
        for (int i = 0; i < panjang[0]; i++) {
            cout << "Tiket ke-" << i + 1 << ":" << endl;
            cout << "Nama Penumpang: " << namaPenumpang[i] << endl;
            cout << "Stasiun: " << stasiun[stasiunDipilih[i]] << endl;
            cout << "Rute: " << rute[stasiunDipilih[i]] << endl;
            cout << "Jumlah Tiket: " << jumlahTiket[i] << endl;
            cout << "Total Harga: Rp " << totalHarga[i] << endl;
            cout << "-----" << endl;
        }
        cout << "Masukkan nomor tiket yang akan dihapus: ";
        cin >> index;
        cin.ignore(); // Membersihkan buffer

        if (index > 0 && index <= panjang[0]) {
            jumlahTiketStasiun[stasiunDipilih[index - 1]] += jumlahTiket[index - 1];
            for (int i = index - 1; i < panjang[0] - 1; i++) {
                namaPenumpang[i] = namaPenumpang[i + 1];
                stasiunDipilih[i] = stasiunDipilih[i + 1];
                jumlahTiket[i] = jumlahTiket[i + 1];
                totalHarga[i] = totalHarga[i + 1];
            }
            panjang[0]--;
        }
    }
}
```

Gambar 3,7

H. Fitur Logout

Fitur logout adalah opsi yang memungkinkan pengguna untuk keluar dari program. Ketika pengguna memilih opsi keluar, program akan menampilkan pesan Logout berhasil. “Terima kasih sudah menggunakan program, sampai berjumpa kembali.” dan menghentikan eksekusi program.

```
case 6: // Keluar
    cout << "Terima kasih sudah menggunakan program, sampai berjumpa kembali!" << endl;
    break;

default:
    cout << "Pilihan tidak valid. Silakan coba lagi." << endl;
    break;
```

Gambar 3,8

4. Uji Coba dan Hasil Output

4.1 Uji Coba

Skenario 1

Pada skenario ini program menjalankan bagian awal yaitu login, jika login tersebut salah melebihi 3x, maka program akan berhenti.

skenario 2

Pada skenario ini jika user login dengan kata sandi dan username dengan benar, maka program akan berhasil dan masuk ke menu utama
Gambar .

Skenario 3

Pada skenario ini, pengguna memilih opsi 1 pada menu utama untuk melihat daftar stasiun, rute, jumlah tiket yang tersedia, dan harga tiket.

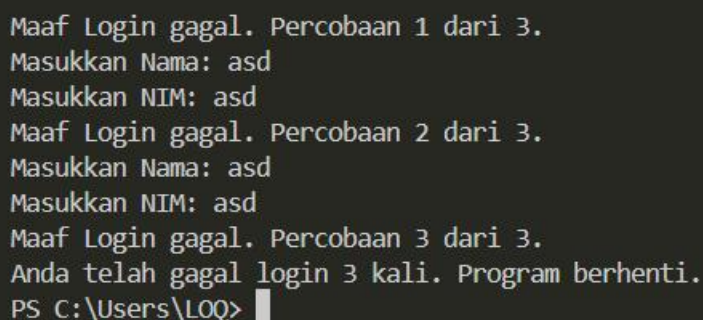
Skenario 4

Pada skenario ini, pengguna memilih opsi 2 untuk melihat tiket yang sudah dipesan. Jika belum ada tiket yang dipesan, program akan menampilkan pemberitahuan.

Skenario 5

Pada skenario ini, pengguna memilih opsi 5 untuk menghapus tiket yang sudah dipesan.

4.2 Hasil Output



```
Maaf Login gagal. Percobaan 1 dari 3.  
Masukkan Nama: asd  
Masukkan NIM: asd  
Maaf Login gagal. Percobaan 2 dari 3.  
Masukkan Nama: asd  
Masukkan NIM: asd  
Maaf Login gagal. Percobaan 3 dari 3.  
Anda telah gagal login 3 kali. Program berhenti.  
PS C:\Users\LOQ>
```

Gambar 4.1 Contoh 1

```

=====
Selamat datang di Program Booking Tiket Kereta Api
=====
1. Tampilkan List Stasiun dan Rute
2. Tampilkan Tiket yang Dipesan
3. Tambah Tiket
4. Ubah Tiket
5. Hapus Tiket
6. Keluar
Pilihan: |

```

Gambar 4.2 Contoh 2

```

Daftar Stasiun dan Rute:
-----
Stasiun: Stasiun UNMUL
Rute: Samarinda-Balikpapan
Jumlah Tiket Tersedia: 10
Harga: Rp 500000
-----
Stasiun: Stasiun AI
Rute: Samarinda-Tenggarong
Jumlah Tiket Tersedia: 10
Harga: Rp 250000
-----
Stasiun: Stasiun ISO
Rute: Balikpapan-Berau
Jumlah Tiket Tersedia: 10
Harga: Rp 350000
-----
Stasiun: Stasiun APL
Rute: Balikpapan-Tanah Grogot
Jumlah Tiket Tersedia: 10
Harga: Rp 150000
-----
Stasiun: Stasiun Masjaya
Rute: Samarinda-Bontang
Jumlah Tiket Tersedia: 10
Harga: Rp 300000
-----
Tekan Enter untuk melanjutkan...|

```

Gambar 4.3 Contoh 3

```
Daftar Tiket yang Dipesan.  
-----  
Tiket ke-1:  
Nama Penumpang: Deni  
Stasiun: Stasiun AI  
Rute: Samarinda-Tenggarong  
Jumlah Tiket: 5  
Total Harga: Rp 1250000  
-----  
Tekan Enter untuk melanjutkan...|
```

Gambar 4.4 Contoh 4

```
Tiket ke-1:  
Nama Penumpang: Deni  
Stasiun: Stasiun AI  
Rute: Samarinda-Tenggarong  
Jumlah Tiket: 5  
Total Harga: Rp 1250000  
-----  
Masukkan nomor tiket yang akan dihapus: 1  
Tiket berhasil dihapus!  
Tekan Enter untuk melanjutkan...|
```

Gambar 4.5 Contoh 5

5. GIT

```
LOQ@LAPTOP-4B4VUDGG MINGW64 ~/Desktop/GITHUB/Praktikum-Ap1 (main)
$ git add .

LOQ@LAPTOP-4B4VUDGG MINGW64 ~/Desktop/GITHUB/Praktikum-Ap1 (main)
$ git commit -m "Posttest 2"
[main 9aede97] Posttest 2
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Post-test/Post-test-2/2409106097-SayidAnnashirIkhwan-PT-2.do
CX

LOQ@LAPTOP-4B4VUDGG MINGW64 ~/Desktop/GITHUB/Praktikum-Ap1 (main)
$ git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 20 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 996.06 KiB | 943.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/wizker12/Praktikum-Ap1.git
6367618..9aede97 main -> main
branch 'main' set up to track 'origin/main'.
```

1. git add .

- Perintah ini digunakan untuk menambahkan semua perubahan (file yang baru dibuat, diubah, atau dihapus) ke dalam staging area.
- Titik (`.`) menunjukkan bahwa semua file yang ada dalam direktori kerja akan ditambahkan.

2. git commit -m "Posttest 2"

- Perintah ini membuat commit dengan pesan `"Posttest 2"`, yang mendokumentasikan perubahan yang telah ditambahkan ke staging area.
- Commit ini menyimpan snapshot dari perubahan dalam repository lokal.

3. git push -u origin main`

- Perintah ini mengunggah commit yang telah dibuat ke repository di GitHub pada branch `main`.
- `-u` (atau `--set-upstream`) digunakan untuk menghubungkan branch lokal `main` dengan branch `main` di repository remote (`origin`).
- `origin` adalah alias default untuk repository remote yang dikloning atau diatur sebelumnya.

4. Output yang dihasilkan:

- Git menghitung dan mengompresi objek sebelum mengunggahnya ke repository remote.
- Terlihat bahwa ada 7 objek yang dihitung dan 5 objek yang dikirim.
- Branch `main` kini terhubung ke `origin/main`, sehingga ke depannya pengguna dapat melakukan `git push` tanpa harus menyertakan `origin main`.

