

**LAPORAN PRAKTIKUM**  
**POSTTEST 5**  
**ALGORITMA PEMROGRAMAN LANJUT**

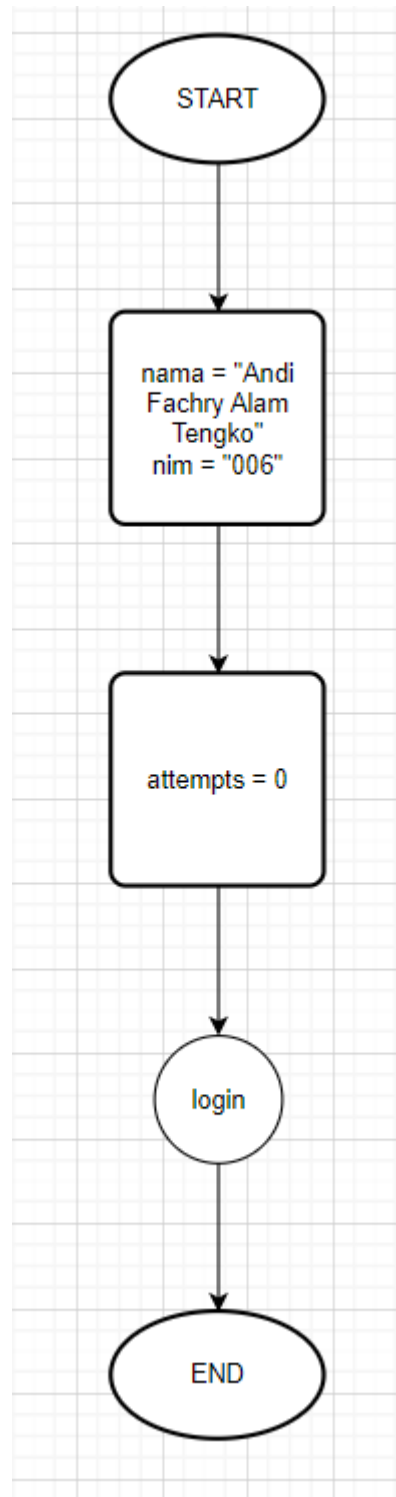


**Disusun oleh:**  
**Andi Fachry Alam Tengko**  
**2409106006**  
**Kelas A1'24**

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

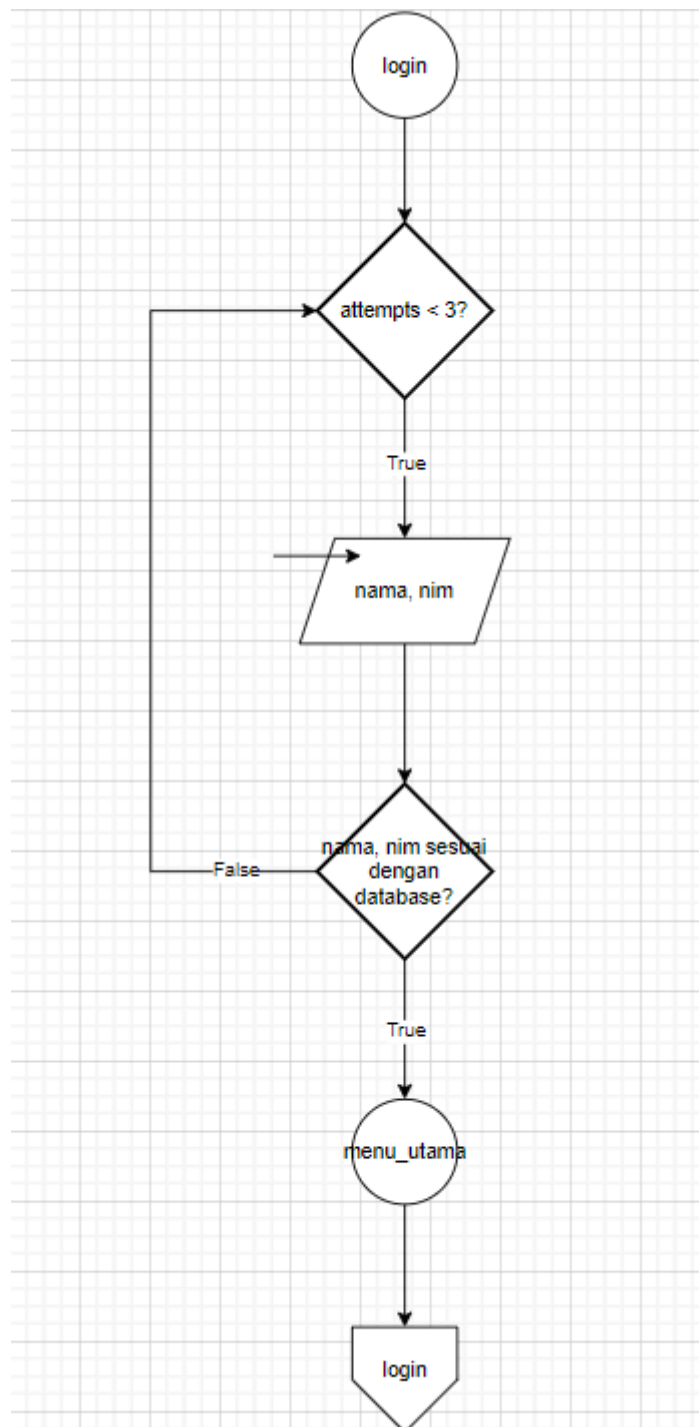
## 1. Flowchart

- Main



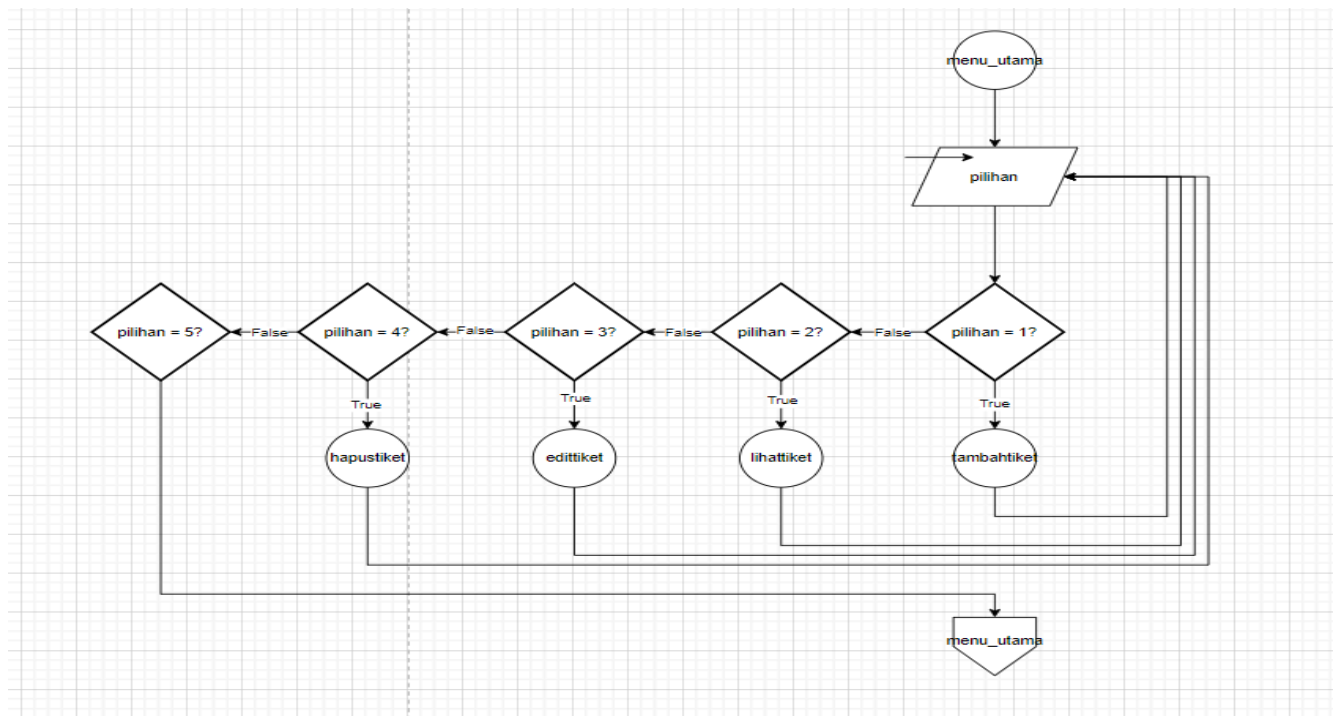
Gambar 2.1 FC-Main

- Login



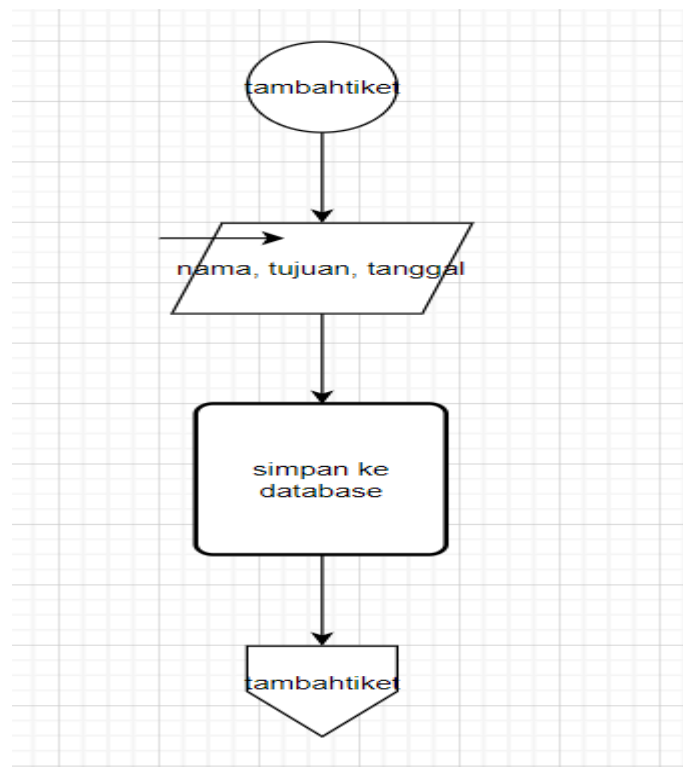
Gambar 2.2 FC-Login

- Menu Utama



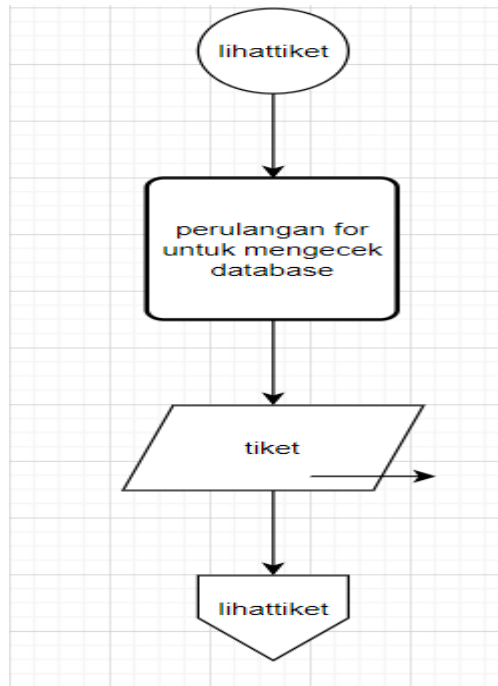
Gambar 2.3 FC-Menu-Utama

- Tambah Tiket



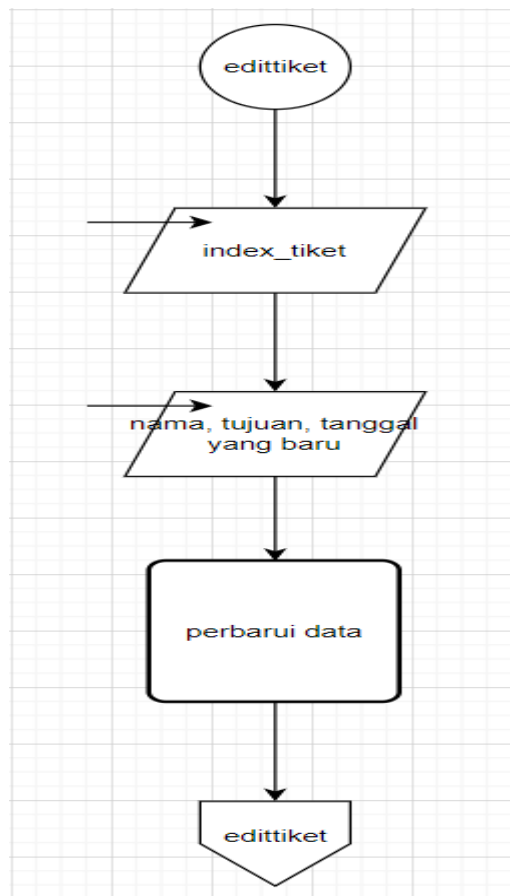
Gambar 2.4 FC-Tambah-Tiket

- Lihat Tiket



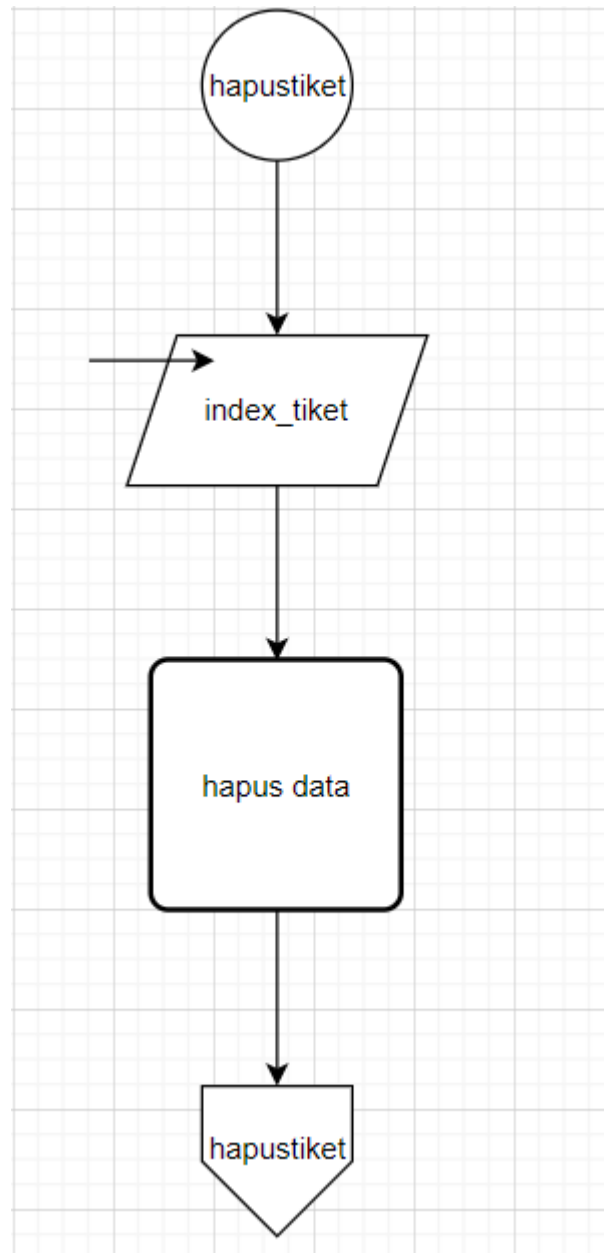
Gambar 2.5 FC-Lihat-Tiket

- Edit Tiket



Gambar 2.6 FC-Edit-Tiket

- Hapus Tiket



Gambar 2.7 FC-Hapus-Tiket

## **2. Analisis Program**

Program ini merupakan sebuah sistem pemesanan tiket pesawat. Program ini memungkinkan pengguna untuk melakukan login terlebih dahulu sebelum bisa mengakses fitur-fitur utama. Proses login ini berguna untuk membatasi akses hanya kepada pengguna yang memiliki nama dan NIM yang sesuai, serta memberikan keamanan dasar dengan membatasi percobaan login maksimal tiga kali.

Setelah berhasil login, pengguna bisa mengakses berbagai fitur seperti membeli tiket, melihat daftar tiket, mengubah data tiket yang sudah dipesan, dan menghapus tiket jika tidak lagi dibutuhkan. Ketika membeli tiket, pengguna akan diminta mengisi nama, tujuan penerbangan, dan tanggal keberangkatan. Data ini kemudian disimpan dalam array dan jumlah tiket yang sudah dipesan akan bertambah. Ada juga batasan maksimum agar jumlah tiket tidak melebihi kapasitas yang sudah ditentukan.

### 3. Source Code

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

const int MAKS_TIKET = 100;

string namaPemesan[MAKS_TIKET];
string tujuanPenerbangan[MAKS_TIKET];
string tanggalPenerbangan[MAKS_TIKET];
int jumlahTiket = 0;

void login(string namaBenar, string nimBenar) {
    string namaInput, nimInput;
    int attempts = 0;
    bool loginBerhasil = false;
    while (attempts < 3) {
        cout << "Masukkan Nama: ";
        getline(cin, namaInput);
        cout << "Masukkan NIM: ";
        cin >> nimInput;
        cin.ignore();
        if (namaInput == namaBenar && nimInput == nimBenar) {
            loginBerhasil = true;
            break;
        } else {
            cout << "Nama atau NIM salah! Coba lagi.\n";
            attempts++;
        }
    }
    if (!loginBerhasil) {
        cout << "Login gagal setelah 3 kali percobaan. Program berhenti.\n";
        exit(0);
    }
}

void tambahJumlahTiket(int* jumlah) {
    if (*jumlah < MAKS_TIKET) {
        (*jumlah)++;
    } else {
        cout << "Jumlah tiket telah mencapai batas maksimum.\n";
    }
}

void tambahTiket(string nama, string tujuan, string tanggal) {
    namaPemesan[jumlahTiket] = nama;
    tujuanPenerbangan[jumlahTiket] = tujuan;
    tanggalPenerbangan[jumlahTiket] = tanggal;
    tambahJumlahTiket(&jumlahTiket);
}
```



```

}

void tambahTiket() {
    if (jumlahTiket >= MAKS_TIKET) {
        cout << "Tiket sudah penuh!\n";
        return;
    }
    string nama, tujuan, tanggal;
    cout << "Masukkan Nama Pemesan: ";
    getline(cin, nama);
    cout << "Masukkan Tujuan: ";
    getline(cin, tujuan);
    cout << "Masukkan Tanggal Penerbangan (DD-MM-YYYY): ";
    getline(cin, tanggal);
    tambahTiket(nama, tujuan, tanggal);
    cout << "Tiket berhasil dipesan!\n";
}

void tampilkanTiket() {
    if (jumlahTiket == 0) {
        cout << "Belum ada tiket yang dipesan.\n";
        return;
    }
    cout << "Daftar Tiket yang sudah dipesan:\n";
    for (int i = 0; i < jumlahTiket; i++) {
        cout << "Tiket " << (i + 1)
            << " - Nama: " << namaPemesan[i]
            << ", Tujuan: " << tujuanPenerbangan[i]
            << ", Tanggal: " << tanggalPenerbangan[i] << endl;
    }
}

void tampilkanTiketPtr(int* index) {
    if (*index < 0 || *index >= jumlahTiket) {
        cout << "Nomor tiket tidak valid.\n";
        return;
    }
    cout << "Detail Tiket ke-" << (*index + 1) << ":\n";
    cout << "Nama: " << namaPemesan[*index] << endl;
    cout << "Tujuan: " << tujuanPenerbangan[*index] << endl;
    cout << "Tanggal: " << tanggalPenerbangan[*index] << endl;
}

void ubahTiket(int nomor) {
    if (nomor < 1 || nomor > jumlahTiket) {
        cout << "Tiket tidak ditemukan.\n";
        return;
    }
    cout << "Data tiket yang akan diubah:\n";
    cout << "Nama: " << namaPemesan[nomor - 1] << endl;
}

```

```

        cout << "Tujuan: " << tujuanPenerbangan[nomor - 1] << endl;
        cout << "Tanggal: " << tanggalPenerbangan[nomor - 1] << endl;
        cout << "Masukkan Nama Baru: ";
        getline(cin, namaPemesan[nomor - 1]);
        cout << "Masukkan Tujuan Baru: ";
        getline(cin, tujuanPenerbangan[nomor - 1]);
        cout << "Masukkan Tanggal Baru: ";
        getline(cin, tanggalPenerbangan[nomor - 1]);
        cout << "Tiket berhasil diperbarui!\n";
    }

void hapusTiketRekursif(int index) {
    if (index >= jumlahTiket - 1) return;

    namaPemesan[index] = namaPemesan[index + 1];
    tujuanPenerbangan[index] = tujuanPenerbangan[index + 1];
    tanggalPenerbangan[index] = tanggalPenerbangan[index + 1];
    hapusTiketRekursif(index + 1);
}

void hapusTiket(int nomor) {
    if (nomor < 1 || nomor > jumlahTiket) {
        cout << "Tiket tidak ditemukan.\n";
        return;
    }
    hapusTiketRekursif(nomor - 1);
    jumlahTiket--;
    cout << "Tiket berhasil dihapus!\n";
}

int main() {
    login("Andi Fachry Alam Tengko", "006");
    int pilihan;
    do {
        cout << "\nMenu Sistem Pemesanan Tiket Pesawat\n";
        cout << "1. Membeli Tiket\n";
        cout << "2. Melihat Daftar Tiket\n";
        cout << "3. Mengubah Data Pemesanan\n";
        cout << "4. Menghapus Tiket\n";
        cout << "5. Keluar\n";
        cout << "Pilih menu: ";
        cin >> pilihan;
        cin.ignore();
        switch (pilihan) {
            case 1:
                tambahTiket();
                break;
            case 2: {
                int subPilihan;
                cout << "1. Tampilkan Semua Tiket\n";

```

```

        cout << "2. Tampilkan Tiket Tertentu\n";
        cout << "Pilih opsi: ";
        cin >> subPilihan;
        cin.ignore();
        if (subPilihan == 1) {
            tampilkanTiket();
        } else if (subPilihan == 2) {
            int nomor;
            cout << "Masukkan nomor tiket yang ingin ditampilkan: ";
            cin >> nomor;
            cin.ignore();
            nomor--;
            tampilkanTiketPtr(&nomor);
        } else {
            cout << "Pilihan tidak valid.\n";
        }
        break;
    }
    case 3: {
        int nomor;
        cout << "Masukkan nomor tiket yang ingin diubah: ";
        cin >> nomor;
        cin.ignore();
        ubahTiket(nomor);
        break;
    }
    case 4: {
        int nomor;
        cout << "Masukkan nomor tiket yang ingin dihapus: ";
        cin >> nomor;
        cin.ignore();
        hapusTiket(nomor);
        break;
    }
    case 5:
        cout << "Terima kasih, program selesai.\n";
        break;
    default:
        cout << "Pilihan tidak valid, coba lagi.\n";
    }
} while (pilihan != 5);
return 0;
}

```

#### 4. Uji Coba dan Hasil Output

- Login, Membeli Tiket, Lihat Tiket.

```
Masukkan Nama: Andi Fachry Alam Tengko
Masukkan NIM: 006

Menu Sistem Pemesanan Tiket Pesawat
1. Membeli Tiket
2. Melihat Daftar Tiket
3. Mengubah Data Pemesanan
4. Menghapus Tiket
5. Keluar
Pilih menu: 1
Masukkan Nama Pemesan: Tengko
Masukkan Tujuan: Bali
Masukkan Tanggal Penerbangan (DD-MM-YYYY): 26/04/2025
Tiket berhasil dipesan!

Menu Sistem Pemesanan Tiket Pesawat
1. Membeli Tiket
2. Melihat Daftar Tiket
3. Mengubah Data Pemesanan
4. Menghapus Tiket
5. Keluar
Pilih menu: 2
1. Tampilkan Semua Tiket
2. Tampilkan Tiket Tertentu
Pilih opsi: 1
Daftar Tiket yang sudah dipesan:
Tiket 1 - Nama: Tengko, Tujuan: Bali, Tanggal: 26/04/2025
```

Gambar 4.1 Percobaan-Login-Membeli-Tiket-Lihat-Tiket

- Edit Pesanan dan Lihat Tiket.

```
Menu Sistem Pemesanan Tiket Pesawat
1. Membeli Tiket
2. Melihat Daftar Tiket
3. Mengubah Data Pemesanan
4. Menghapus Tiket
5. Keluar
Pilih menu: 3
Masukkan nomor tiket yang ingin diubah: 1
Data tiket yang akan diubah:
Nama: Tengko
Tujuan: Bali
Tanggal: 26/04/2025
Masukkan Nama Baru: Tengkoo
Masukkan Tujuan Baru: Jakarta
Masukkan Tanggal Baru: 24/05/2025
Tiket berhasil diperbarui!

Menu Sistem Pemesanan Tiket Pesawat
1. Membeli Tiket
2. Melihat Daftar Tiket
3. Mengubah Data Pemesanan
4. Menghapus Tiket
5. Keluar
Pilih menu: 2
1. Tampilkan Semua Tiket
2. Tampilkan Tiket Tertentu
Pilih opsi: 1
Daftar Tiket yang sudah dipesan:
Tiket 1 - Nama: Tengkoo, Tujuan: Jakarta, Tanggal: 24/05/2025
```

Gambar 4.2 Percobaan-Edit-Pesanan-Lihat-Tiket

- Hapus Tiket dan Lihat Tiket.

```
Menu Sistem Pemesanan Tiket Pesawat
1. Membeli Tiket
2. Melihat Daftar Tiket
3. Mengubah Data Pemesanan
4. Menghapus Tiket
5. Keluar
Pilih menu: 4
Masukkan nomor tiket yang ingin dihapus: 1
Tiket berhasil dihapus!

Menu Sistem Pemesanan Tiket Pesawat
1. Membeli Tiket
2. Melihat Daftar Tiket
3. Mengubah Data Pemesanan
4. Menghapus Tiket
5. Keluar
Pilih menu: 2
1. Tampilkan Semua Tiket
2. Tampilkan Tiket Tertentu
Pilih opsi: 1
Belum ada tiket yang dipesan.
```

Gambar 4.3 Percobaan-Hapus-Tiket-Lihat-Tiket

- Logout

```
Menu Sistem Pemesanan Tiket Pesawat  
1. Membeli Tiket  
2. Melihat Daftar Tiket  
3. Mengubah Data Pemesanan  
4. Menghapus Tiket  
5. Keluar  
Pilih menu: 5  
Terima kasih, program selesai.
```

Gambar 4.5 Percobaan-Logout

## 5. Langkah-Langkah Git pada VSCode

- Git add

```
PS D:\dump tugas\project 0\fachry\praktikum-apl> git add .
```

- Git Commit

```
PS D:\dump tugas\project 0\fachry\praktikum-apl> git commit -m "Finish Posttest 5"
[main c16713c] Finish Posttest 5
3 files changed, 191 insertions(+)
create mode 100644 post-test/post-test-apl-5/2409106006-AndiFachryAlamTengko-PT-5.cpp
create mode 100644 post-test/post-test-apl-5/2409106006-AndiFachryAlamTengko-PT-5.exe
create mode 100644 post-test/post-test-apl-5/2409106006-AndiFachryAlamTengko-PT-5.pdf
```

- Git Push

```
PS D:\dump tugas\project 0\fachry\praktikum-apl> git push
info: please complete authentication in your browser...
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 20 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 1.12 MiB | 295.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/AndiFachryAlamTengko/praktikum-apl.git
d9eaf2a..c16713c main -> main
```