

LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT

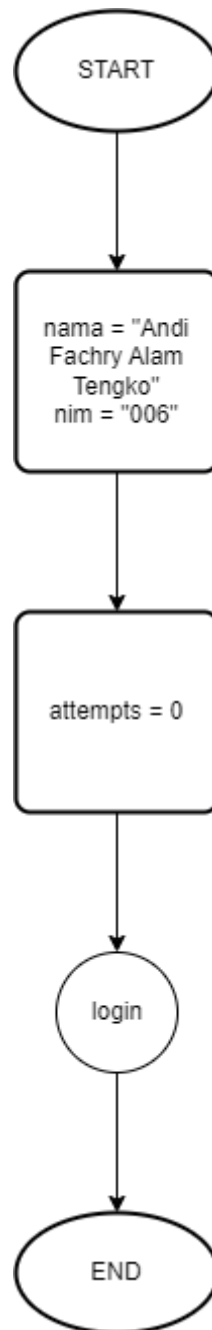


Disusun oleh:
Andi Fachry Alam Tengko 2409106006
Kelas A1'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

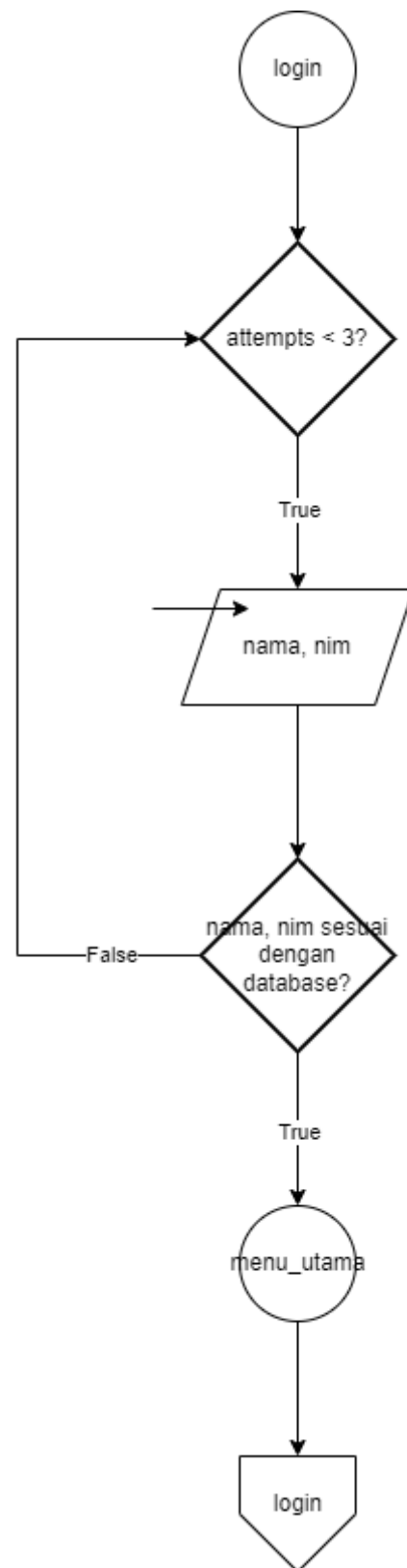
1. Flowchart

- Main



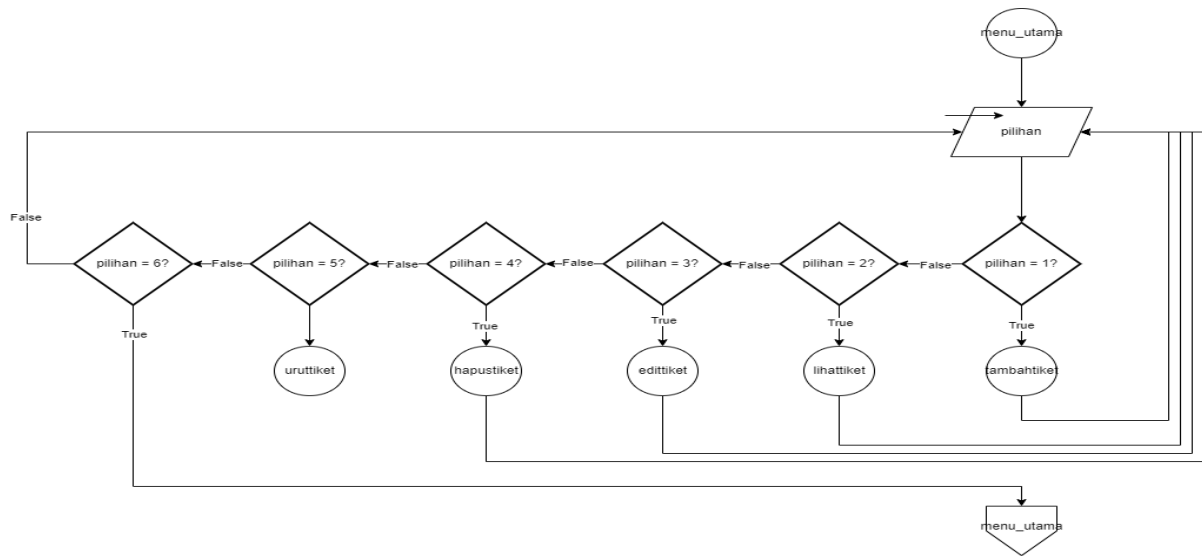
Gambar 2.1 FC-Main

- Login



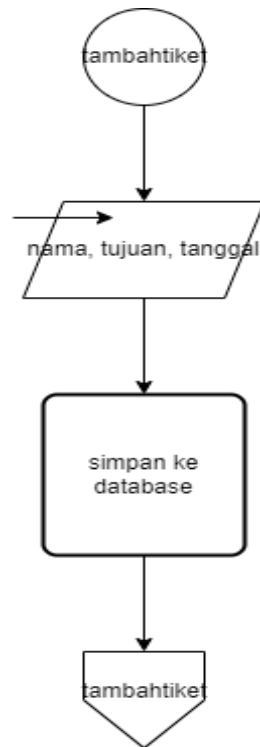
Gambar 2.2 FC-Login

- Menu Utama



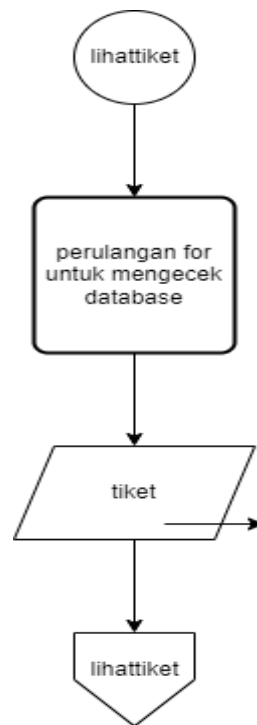
Gambar 2.3 FC-Menu-Utama

- Tambah Tiket



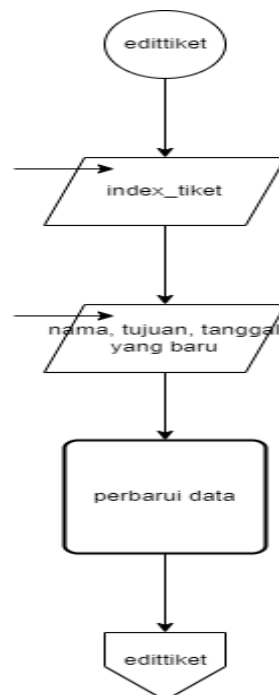
Gambar 2.4 FC-Tambah-Tiket

- Lihat Tiket



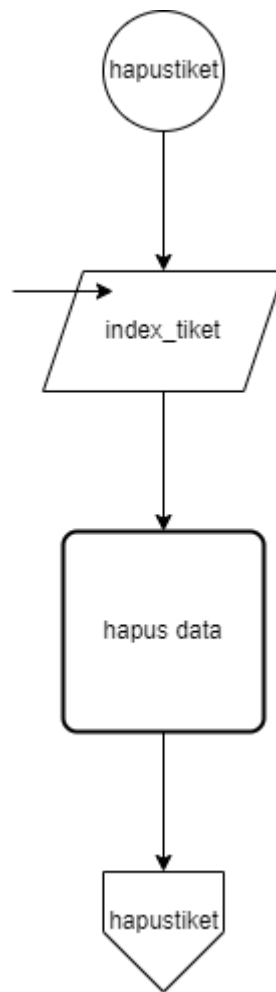
Gambar 2.5 FC-Lihat-Tiket

- Edit Tiket



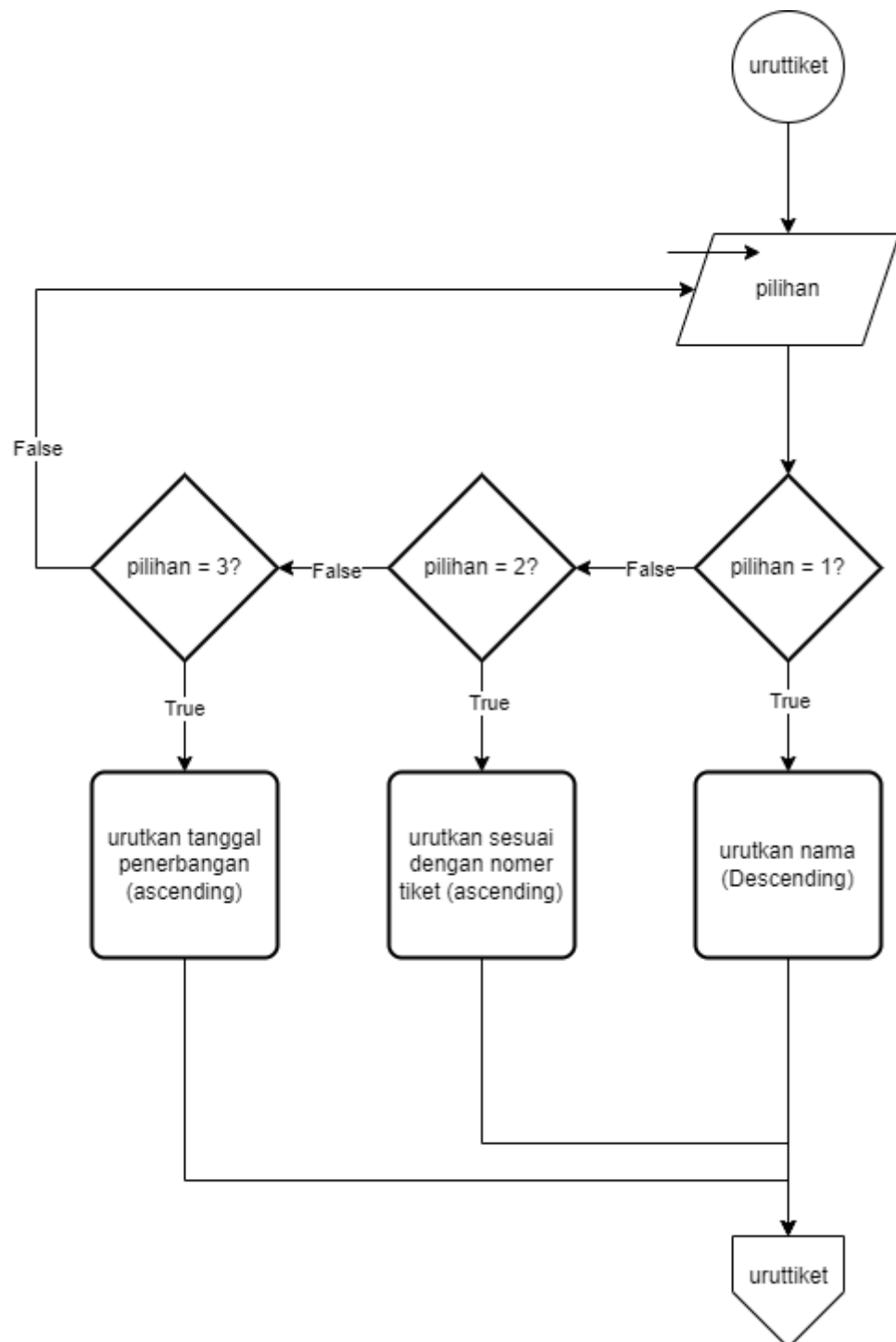
Gambar 2.6 FC-Edit-Tiket

- Hapus Tiket



Gambar 2.7 FC-Hapus-Tiket

- Urut Tiket



Gambar 2.8 FC-Urut-Tiket

2. Analisis Program

Program ini berfungsi untuk mengelola data pemesanan tiket pesawat dengan sistem login user. Setelah login berhasil, pengguna dapat melakukan beberapa fitur utama, yaitu memesan tiket baru, melihat daftar tiket yang telah dipesan, mengubah data tiket yang sudah ada, serta menghapus tiket yang tidak diperlukan. Program ini juga menyediakan fitur untuk menampilkan tiket tertentu berdasarkan indeks menggunakan pointer.

Selain itu, program menerapkan fungsi rekursif untuk menghapus data tiket, di mana data setelah tiket yang dihapus akan digeser ke kiri secara berurutan. Terdapat juga penggunaan overloading fungsi pada proses penambahan tiket, di mana fungsi dapat dipanggil dengan parameter maupun tanpa parameter untuk fleksibilitas input.

Dalam hal pengurutan data, program menyediakan tiga metode sorting. Metode pertama adalah pengurutan nama pemesan secara huruf descending (Z–A). Metode kedua adalah pengurutan nomor tiket secara angka ascending berdasarkan urutan input. Metode ketiga adalah pengurutan tanggal penerbangan secara ascending, yang mengurutkan tiket berdasarkan tanggal dalam format DD-MM-YYYY.

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";
}

void sortNamaDescending() {
    for (int i = 0; i < jumlahTiket - 1; i++) {
        for (int j = 0; j < jumlahTiket - i - 1; j++) {
            if (namaPemesan[j] < namaPemesan[j + 1]) {
                swap(namaPemesan[j], namaPemesan[j + 1]);
                swap(tujuanPenerbangan[j], tujuanPenerbangan[j + 1]);
                swap(tanggalPenerbangan[j], tanggalPenerbangan[j + 1]);
            }
        }
    }
    cout << "Data berhasil diurutkan berdasarkan Nama (Z ke A).\n";
}

void sortNomorTiketAscending() {

```

```

        cout << "Menampilkan data berdasarkan urutan input (nomor tiket)
ascending.\n";
        tampilkanTiket();
    }

void sortTanggalAscending() {
    for (int i = 0; i < jumlahTiket - 1; i++) {
        for (int j = 0; j < jumlahTiket - i - 1; j++) {
            if (tanggalPenerbangan[j] > tanggalPenerbangan[j + 1]) {
                swap(namaPemesan[j], namaPemesan[j + 1]);
                swap(tujuanPenerbangan[j], tujuanPenerbangan[j + 1]);
                swap(tanggalPenerbangan[j], tanggalPenerbangan[j + 1]);
            }
        }
    }
    cout << "Data berhasil diurutkan berdasarkan Tanggal (ascending).\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 << "6. Sorting Data\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;
case 6: {
    int sortPilihan;
    cout << "\nMenu Sorting\n";
    cout << "1. Urutkan Nama (Z - A)\n";
    cout << "2. Urutkan berdasarkan Nomor Tiket (ascending)\n";
    cout << "3. Urutkan Tanggal Penerbangan (ascending)\n";
    cout << "Pilih opsi sorting: ";
    cin >> sortPilihan;
    cin.ignore();
    switch (sortPilihan) {
        case 1:
            sortNamaDescending();
            break;
        case 2:
            sortNomorTiketAscending();
            break;
        case 3:
            sortTanggalAscending();
            break;
        default:

```

```

        cout << "Pilihan tidak valid.\n";
    }
    break;
}
default:
    cout << "Pilihan tidak valid, coba lagi.\n";
}
} while (pilihan != 5);
return 0;
}

```

4. Uji Coba dan Hasil Output

- Login dan Tambah 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
6. Sorting Data
Pilih menu: 1
Masukkan Nama Pemesan: Tengko
Masukkan Tujuan: Jakarta
Masukkan Tanggal Penerbangan (DD-MM-YYYY): 23-02-2005
Tiket berhasil dipesan!

```

Gambar 4.1 Percobaan-Login-Tambah-Tiket

- Lihat Tiket dan Update Tiket

```
Menu Sistem Pemesanan Tiket Pesawat
1. Membeli Tiket
2. Melihat Daftar Tiket
3. Mengubah Data Pemesanan
4. Menghapus Tiket
5. Keluar
6. Sorting Data
Pilih menu: 2
1. Tampilkan Semua Tiket
2. Tampilkan Tiket Tertentu
Pilih opsi: 1
Daftar Tiket yang sudah dipesan:
Tiket 1 - Nama: Tengko, Tujuan: Jakarta, Tanggal: 23-02-2005
Tiket 2 - Nama: Tengko1, Tujuan: Bali, Tanggal: 23-05-2025

Menu Sistem Pemesanan Tiket Pesawat
1. Membeli Tiket
2. Melihat Daftar Tiket
3. Mengubah Data Pemesanan
4. Menghapus Tiket
5. Keluar
6. Sorting Data
Pilih menu: 3
Masukkan nomor tiket yang ingin diubah: 1
Data tiket yang akan diubah:
Nama: Tengko
Tujuan: Jakarta
Tanggal: 23-02-2005
Masukkan Nama Baru: Tengkoo
Masukkan Tujuan Baru: Bandung
Masukkan Tanggal Baru: 23-05-2026
Tiket berhasil diperbarui!
```

Gambar 4.2 Lihat-Minyak-Edit-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
6. Sorting Data
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
6. Sorting Data
Pilih menu: 2
1. Tampilkan Semua Tiket
2. Tampilkan Tiket Tertentu
Pilih opsi: 1
Daftar Tiket yang sudah dipesan:
Tiket 1 - Nama: Tengko1, Tujuan: Bali, Tanggal: 23-05-2025

```

Gambar 4.3 Hapus-Tiket-Lihat-Tiket

- Urutkan Tiket

```

Menu Sorting
1. Urutkan Nama (Z - A)
2. Urutkan berdasarkan Nomor Tiket (ascending)
3. Urutkan Tanggal Penerbangan (ascending)
Pilih opsi sorting: 3
Data berhasil diurutkan berdasarkan Tanggal (ascending).

```

Gambar 4.5 Urutkan-Tiket

- Logout

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

Gambar 4.6 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 6"
[main f75f964] Finish Posttest 6
4 files changed, 247 insertions(+)
create mode 100644 post-test/post-test-apl-6/2409106006-AndiFachryAlamTengko-PT-6.cpp
create mode 100644 post-test/post-test-apl-6/2409106006-AndiFachryAlamTengko-PT-6.exe
create mode 100644 post-test/post-test-apl-6/2409106006-AndiFachryAlamTengko-PT-6.pdf
```

- git push

```
PS D:\dump tugas\project 0\fachry\praktikum-apl> git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 20 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 1.09 MiB | 437.00 KiB/s, done.
Total 9 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/AndiFachryAlamTengko/praktikum-apl.git
a00f875..f75f964 main -> main
branch 'main' set up to track 'origin/main'.
```