

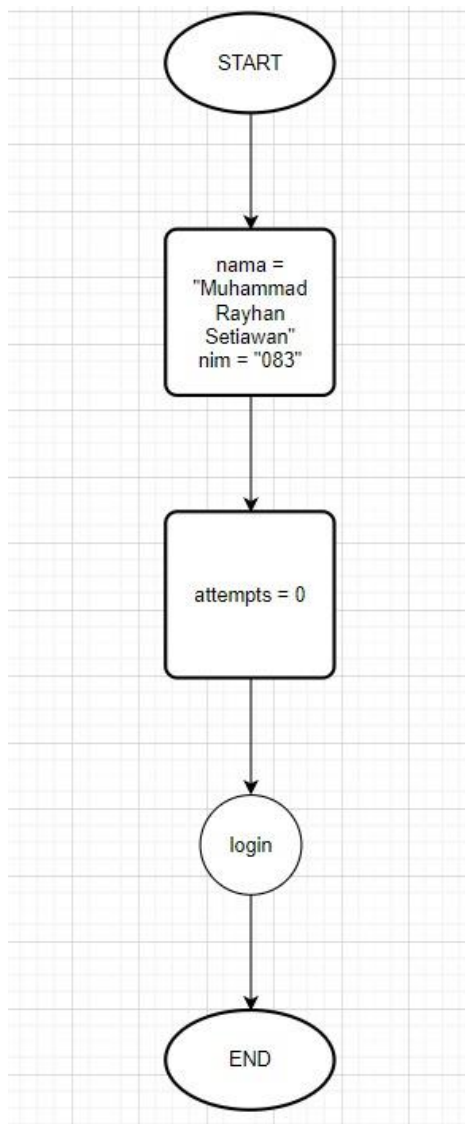
LAPORAN PRAKTIKUM
POSTTEST (5)
ALGORITMA PEMOGRAMAN LANJU



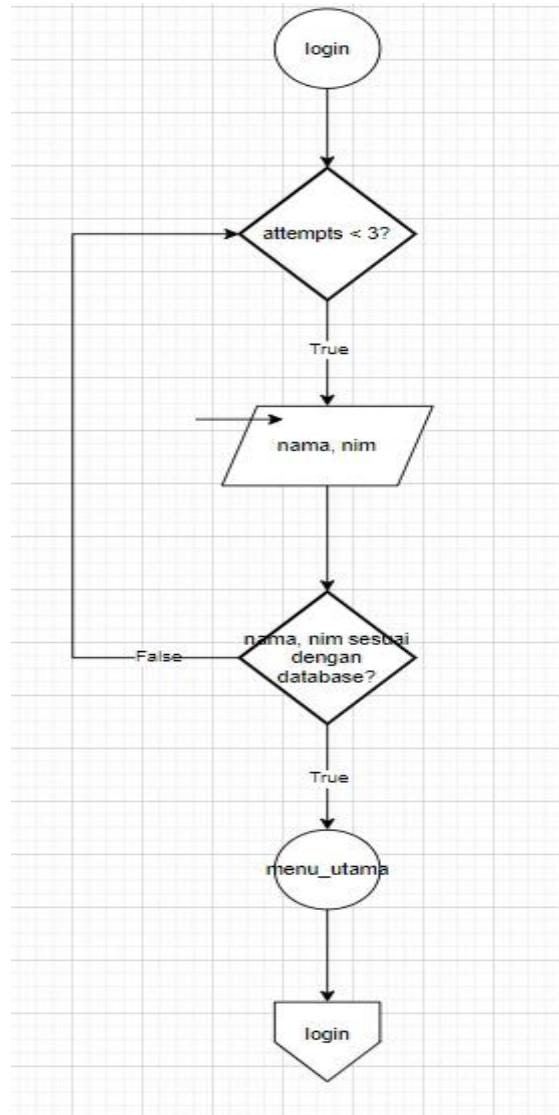
Disusun oleh:
Muhammad Rayhan setiawan (2409106083)
(B2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

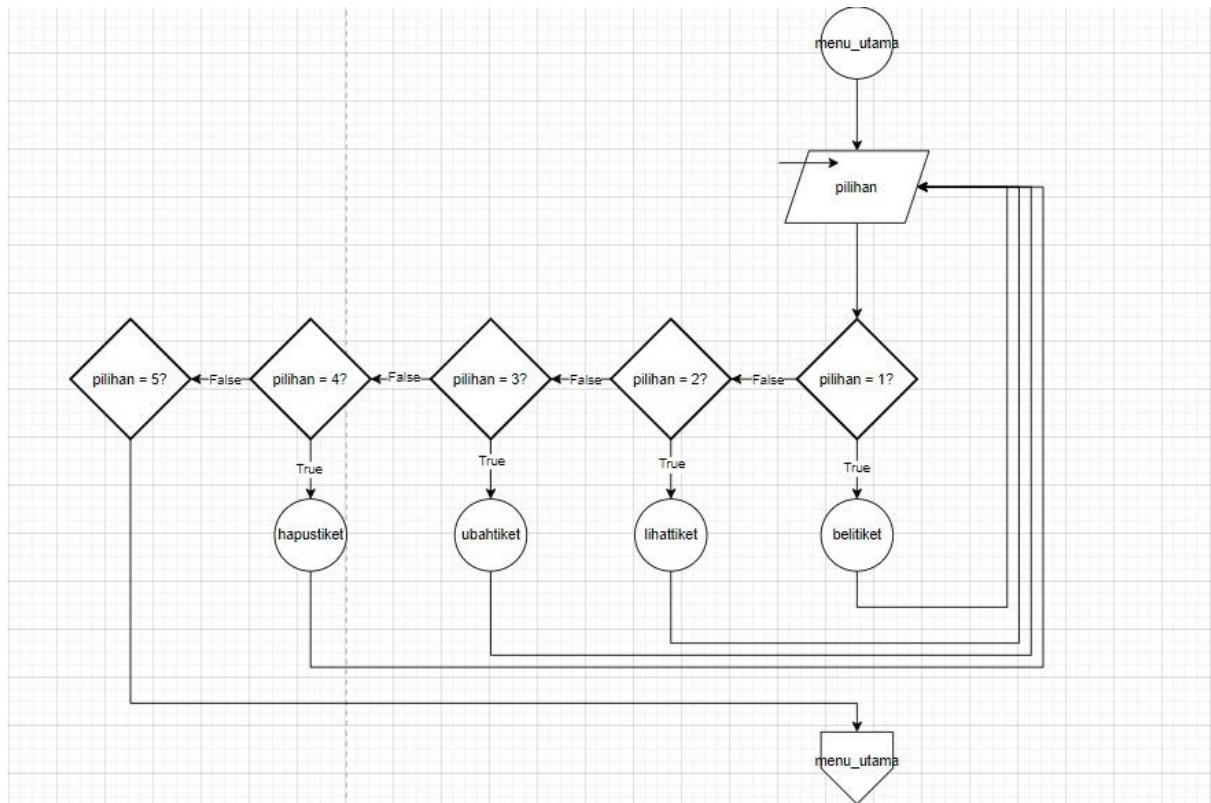
1. Flowchart



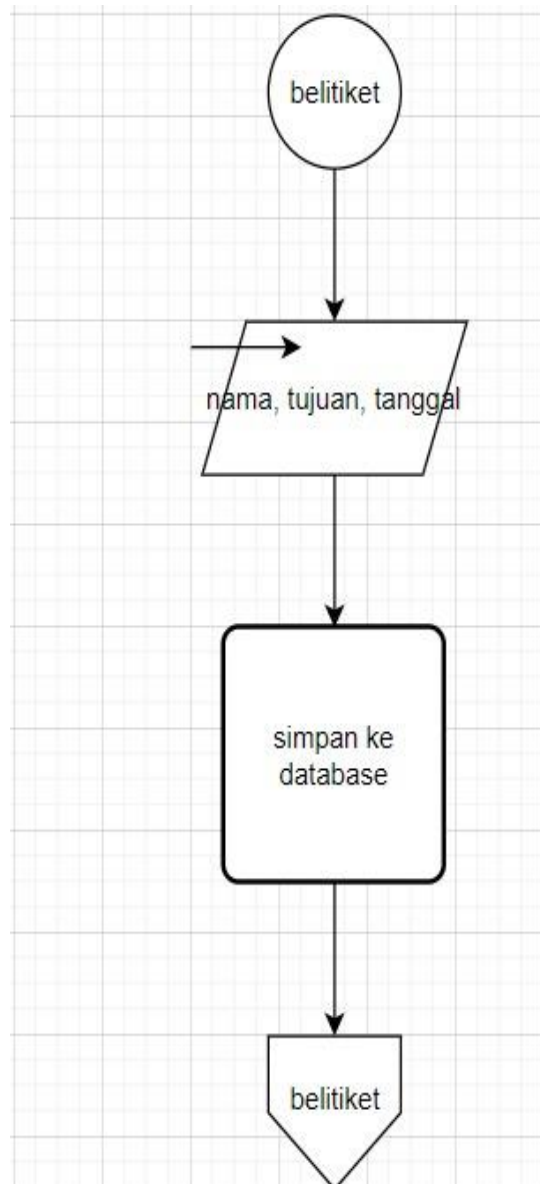
Gambar 1.1



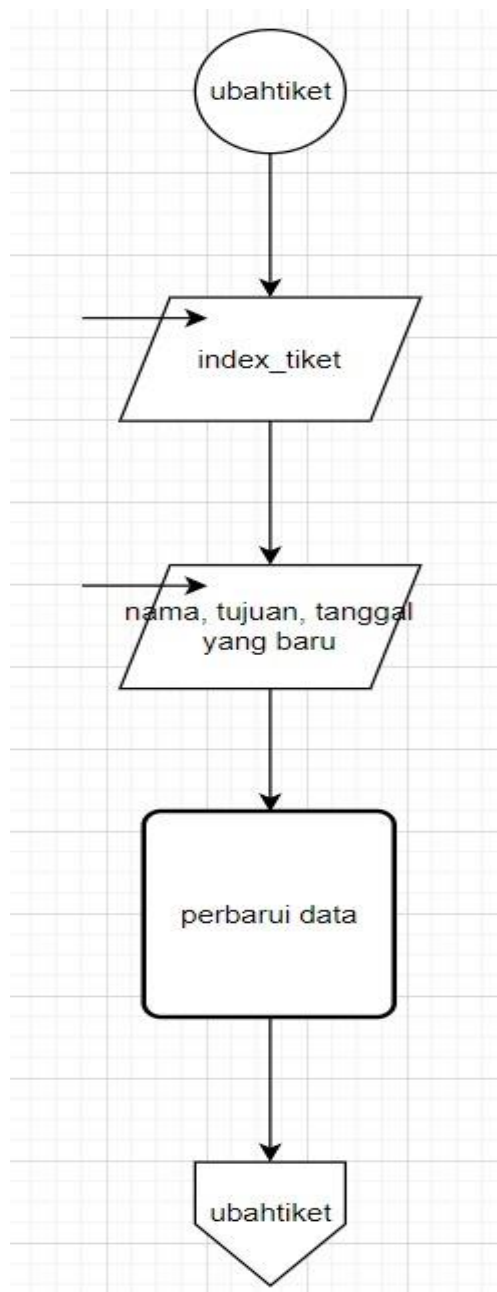
Gambar 1.2



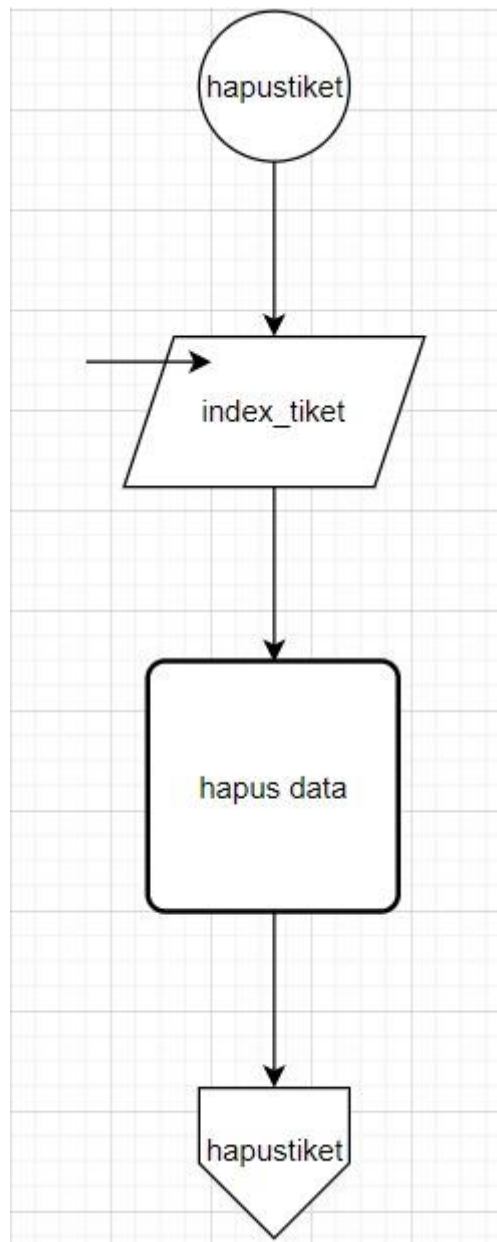
Gambar 1.3



Gambar 1.5



Gambar 1.6



Gambar 1.7

2. Analysis Program

2.1 Deskripsi Singkat Program

Program ini dibuat untuk membantu proses pemesanan tiket kapal secara digital dan sederhana. Program dirancang dengan berbagai fitur penting seperti login, input data tiket, pengubahan tiket, penghapusan tiket, dan penamilikan daftar tiket

2.2 Penjelasan Alur & Algoritma

1) Tujuan Program

Program ini dibuat untuk mempermudah proses pemesanan tiket kapal secara digital. Melalui program ini, pengguna dapat melakukan login, membeli tiket kapal, melihat daftar tiket yang telah dibeli, mengubah informasi tiket, dan menghapus tiket yang sudah tidak diperlukan.

2) Struktur Data

Program menggunakan tiga array paralel untuk menyimpan informasi tiket, yaitu:

- `namaPemesanan[]`: menyimpan nama orang yang memesan tiket.
- `tujuanPelayaran[]`: Menyimpan tujuan perjalanan kapal.
- `tanggalPelayaran[]`: menyimpan tanggal keberangkatan.

Jumlah tiket yang tersimpan dilacak menggunakan variabel `jumlahTiket`. Kapasitas maksimal data adalah 100 tiket, sesuai dengan konstanta `MAKS_TIKET`.

3) Fungsi Login

Fungsi `login()` meminta pengguna memasukkan nama dan Nim. Nama dan Nim yang benar sudah ditentukan dalam parameter fungsi. Jika pengguna salah memasukkan informasi sebanyak 3 kali, maka program akan otomatis berhenti menggunakan `exit(0)`. Fungsi ini menggunakan parameter referensi (`const string&`) agar lebih efisien dan tidak membuat Salinan data.

4) Fungsi Tambah Tiket

Fungsi `tambahTiket()` digunakan untuk menambahkan tiket baru ke dalam array. Fungsi ini menerima parameter berupa pointer, lalu nilai dari pointer tersebut diakses dengan cara dereference untuk disimpan ke array.

5) Fungsi Input Tiket

Fungsi ini digunakan untuk meminta input dari pengguna, yaitu nama pemesanan, tujuan pelayaran, dan tanggal pelayaran. Data yang dimasukkan kemudian dikirim ke fungsi `tambahTiket()` dalam bentuk Alamat memori (pointer). Hal ini menunjukkan penerapan konsep pointer dalam program.

6) Fungsi Tampilkan Tiket

Fungsi `tampilkanTiket()` mencetak seluruh tiket yang telah dipesan ke layar. Jika belum ada tiket, maka program akan memberi tahu pengguna bahwa daftar masih kosong.

7) Fungsi Ubah Tiket

Fungsi ini meminta nomor tiket yang ingin di ubah, kemudian mengganti isi tiket tersebut sesuai data baru yang dimasukkan pengguna. Nomor tiket diteruskan ke fungsi dalam bentuk pointer, lalu diakses untuk melakukan validasi dan update isi array.

8) Fungsi Hapus Tiket

Proses penghapusan tiket dibagi menjadi dua fungsi:

- `hapusTiketrekursif()`: Fungsi rekursif yang memindahkan isi array dari indeks setelahnya ke indeks saat ini, sehingga data akan “terhapus.”
- `hapusTiket()`: Fungsi utama yang menerima nomor tiket dan memanggil fungsi rekursif untuk menghapus data tersebut.

Pengguna rekursi di sini menunjukkan penerapan konsep algoritma rekursif, meskipun untuk penghapusan data sebenarnya lebih umum menggunakan perulangan.

9) Menu Utama

Menu utama ditampilkan menggunakan perulangan do-while agar program terus berjalan sampai pengguna memilih menu keluar.

Pengguna dapat memilih antara:

1. Membeli tiket kapal
2. Melihat daftar tiket
3. Mengubah tiket
4. Menghapus tiket
5. Keluar dari program

Input diproses dengan switch-case, dan `cin.ignore()` digunakan untuk mengatasi masalah input saat beralih dari input angka ke string.

2.2 Source Code

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  const int MAKS_TIKET = 100;
6
7  string namaPemesan[MAKS_TIKET];
8  string tujuanPelayaran[MAKS_TIKET];
9  string tanggalPelayaran[MAKS_TIKET];
10 int jumlahTiket = 0;
11
12 // Fungsi dengan parameter reference (address-of)
13 void login(const string& namaBenar, const string& nimBenar) {
14     string namaInput, nimInput;
15     int attempts = 0;
16     bool loginBerhasil = false;
17
18     while (attempts < 3) {
19         cout << "Masukkan Nama: ";
20         getline(cin, namaInput);
21         cout << "Masukkan NIM: ";
22         cin >> nimInput;
23         cin.ignore();
24
25         if (namaInput == namaBenar && nimInput == nimBenar) {
26             loginBerhasil = true;
27             break;
28         } else {
29             cout << "Nama atau NIM salah! Coba lagi.\n";
30             attempts++;
31         }
32     }
33
34     if (!loginBerhasil) {
35         cout << "Login gagal setelah 3 kali percobaan. Program berhenti.\n";
36         exit(0);
37     }
38 }
```

Gambar Code 2.1

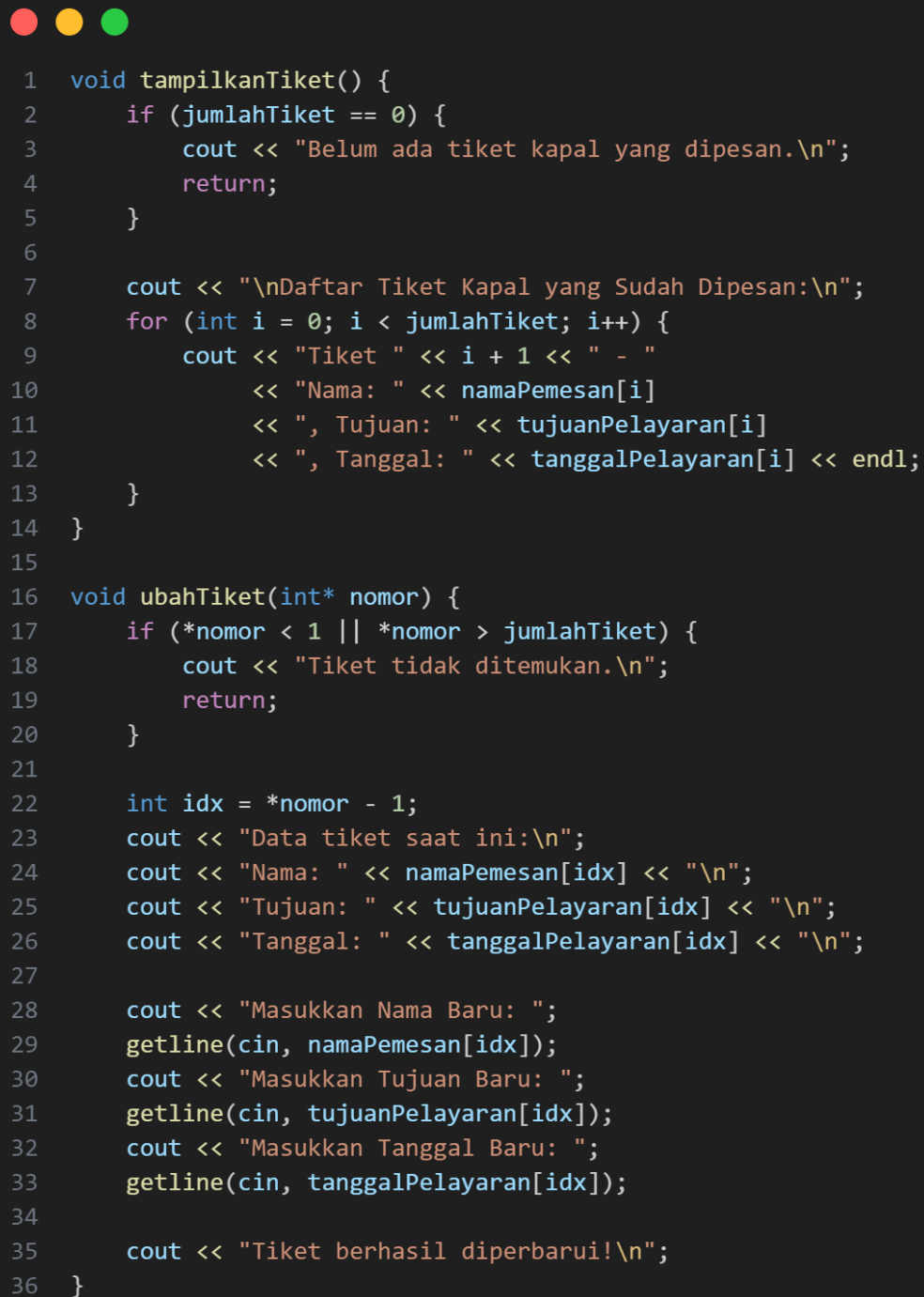


```

1  // Fungsi dengan parameter dereference pointer
2  void tambahTiket(string* nama, string* tujuan, string* tanggal) {
3      namaPemesan[jumlahTiket] = *nama;
4      tujuanPelayaran[jumlahTiket] = *tujuan;
5      tanggalPelayaran[jumlahTiket] = *tanggal;
6      jumlahTiket++;
7  }
8
9  void inputTiket() {
10     if (jumlahTiket >= MAKS_TIKET) {
11         cout << "Tiket sudah penuh!\n";
12         return;
13     }
14
15     string nama, tujuan, tanggal;
16     cout << "Masukkan Nama Pemesan: ";
17     getline(cin, nama);
18     cout << "Masukkan Tujuan Pelayaran: ";
19     getline(cin, tujuan);
20     cout << "Masukkan Tanggal Pelayaran (DD-MM-YYYY): ";
21     getline(cin, tanggal);
22
23     // Menggunakan pointer
24     tambahTiket(&nama, &tujuan, &tanggal);
25     cout << "Tiket berhasil dipesan!\n";
26 }

```

Gambar Code 2.2



```

1 void tampilkanTiket() {
2     if (jumlahTiket == 0) {
3         cout << "Belum ada tiket kapal yang dipesan.\n";
4         return;
5     }
6
7     cout << "\nDaftar Tiket Kapal yang Sudah Dipesan:\n";
8     for (int i = 0; i < jumlahTiket; i++) {
9         cout << "Tiket " << i + 1 << " - "
10            << "Nama: " << namaPemesan[i]
11            << ", Tujuan: " << tujuanPelayaran[i]
12            << ", Tanggal: " << tanggalPelayaran[i] << endl;
13     }
14 }
15
16 void ubahTiket(int* nomor) {
17     if (*nomor < 1 || *nomor > jumlahTiket) {
18         cout << "Tiket tidak ditemukan.\n";
19         return;
20     }
21
22     int idx = *nomor - 1;
23     cout << "Data tiket saat ini:\n";
24     cout << "Nama: " << namaPemesan[idx] << "\n";
25     cout << "Tujuan: " << tujuanPelayaran[idx] << "\n";
26     cout << "Tanggal: " << tanggalPelayaran[idx] << "\n";
27
28     cout << "Masukkan Nama Baru: ";
29     getline(cin, namaPemesan[idx]);
30     cout << "Masukkan Tujuan Baru: ";
31     getline(cin, tujuanPelayaran[idx]);
32     cout << "Masukkan Tanggal Baru: ";
33     getline(cin, tanggalPelayaran[idx]);
34
35     cout << "Tiket berhasil diperbarui!\n";
36 }

```

Gambar Code 2.3



```
1 void hapusTiketRekursif(int index) {
2     if (index >= jumlahTiket - 1) return;
3
4     namaPemesan[index] = namaPemesan[index + 1];
5     tujuanPelayaran[index] = tujuanPelayaran[index + 1];
6     tanggalPelayaran[index] = tanggalPelayaran[index + 1];
7
8     hapusTiketRekursif(index + 1);
9 }
10
11 void hapusTiket(int* nomor) {
12     if (*nomor < 1 || *nomor > jumlahTiket) {
13         cout << "Tiket tidak ditemukan.\n";
14         return;
15     }
16
17     hapusTiketRekursif(*nomor - 1);
18     jumlahTiket--;
19     cout << "Tiket berhasil dihapus!\n";
20 }
```

Gambar Code 2.4

```

1  int main() {
2      login("Muhammad Rayhan Setiawan", "083");
3
4      int pilihan;
5      do {
6          cout << "\n=== Menu Sistem Pemesanan Tiket Kapal ===\n";
7          cout << "1. Beli Tiket Kapal\n";
8          cout << "2. Lihat Daftar Tiket\n";
9          cout << "3. Ubah Tiket\n";
10         cout << "4. Hapus Tiket\n";
11         cout << "5. Keluar\n";
12         cout << "Pilih menu: ";
13         cin >> pilihan;
14         cin.ignore();
15
16         switch (pilihan) {
17             case 1:
18                 inputTiket();
19                 break;
20             case 2:
21                 tampilkanTiket();
22                 break;
23             case 3: {
24                 int nomor;
25                 cout << "Masukkan nomor tiket yang ingin diubah: ";
26                 cin >> nomor;
27                 cin.ignore();
28                 ubahTiket(&nomor);
29                 break;
30             }
31             case 4: {
32                 int nomor;
33                 cout << "Masukkan nomor tiket yang ingin dihapus: ";
34                 cin >> nomor;
35                 cin.ignore();
36                 hapusTiket(&nomor);
37                 break;
38             }
39             case 5:
40                 cout << "Terima kasih telah menggunakan sistem pemesanan tiket kapal.\n";
41                 break;
42             default:
43                 cout << "Pilihan tidak valid!\n";
44         }
45     } while (pilihan != 5);
46
47     return 0;
48 }

```

Gambar Code 2.5

5. Langkah Langkah Github

1. Git add

```
PS D:\praktikum-apl\post-test\post-test-apl-5> git add .
```

2. Git commit

```
PS D:\praktikum-apl\post-test\post-test-apl-5> git commit -m "Finish Posttest 5"
[main 4797aba] Finish Posttest 5
3 files changed, 173 insertions(+)
create mode 100644 post-test/post-test-apl-5/2409106083-MuhammadRayhanSeitiawan-PT-5.cpp
create mode 100644 post-test/post-test-apl-5/2409106083-MuhammadRayhanSeitiawan-PT-5.exe
create mode 100644 post-test/post-test-apl-5/2409106083-MuhammadRayhanSeitiawan-PT-5.pdf
```

3. Git Push

```
PS D:\praktikum-apl\post-test\post-test-apl-5> git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 16 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 2.01 MiB | 533.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/rrrreeehannn/praktikum-apl.git
   bb986aa..4797aba  main -> main
branch 'main' set up to track 'origin/main'.
```