

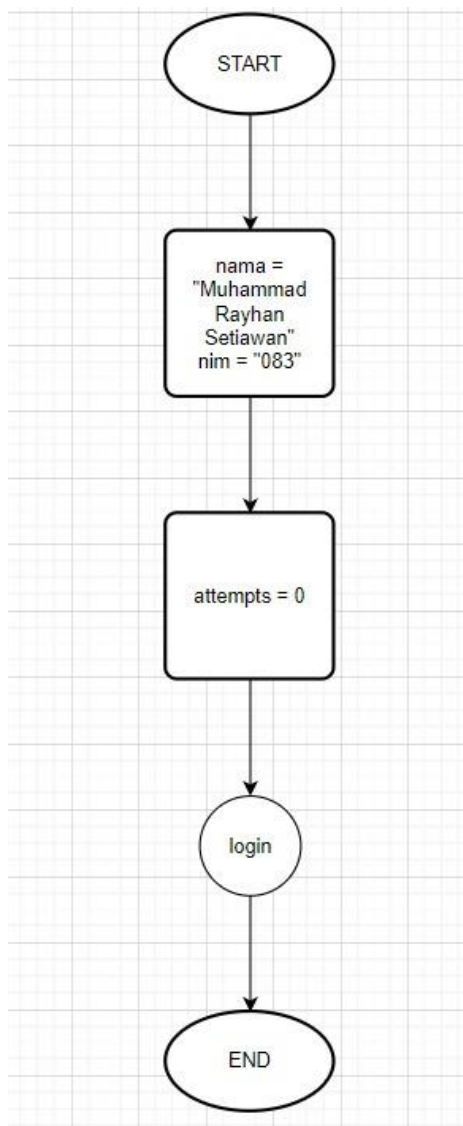
LAPORAN PRAKTIKUM
POSTTEST (6)
ALGORITMA PEMOGRAMAN LANJU



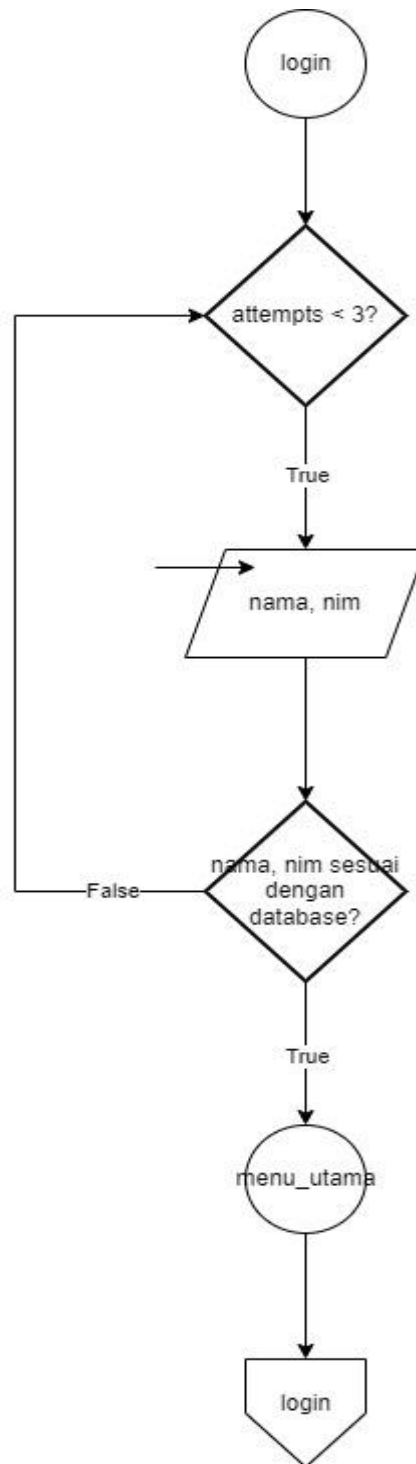
Disusun oleh:
Muhammad Rayhan setiawan (2409106083)
(B2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

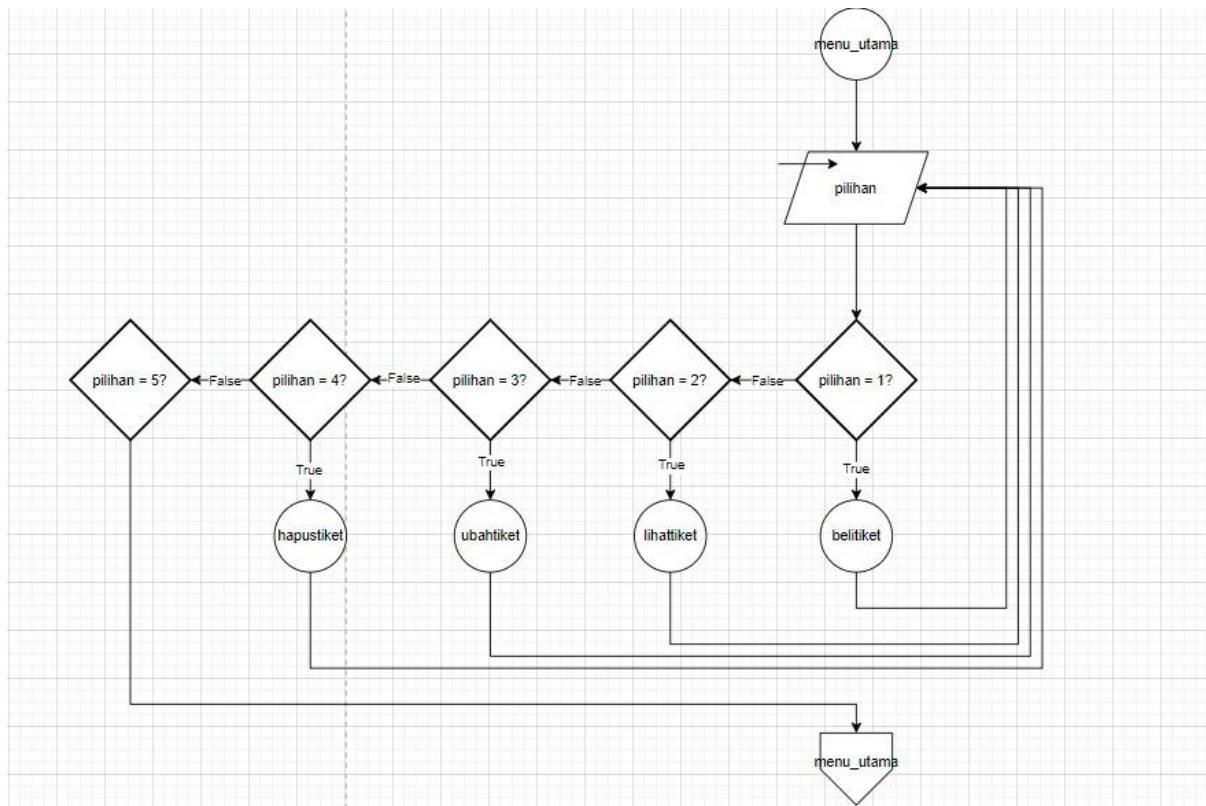
1. Flowchart



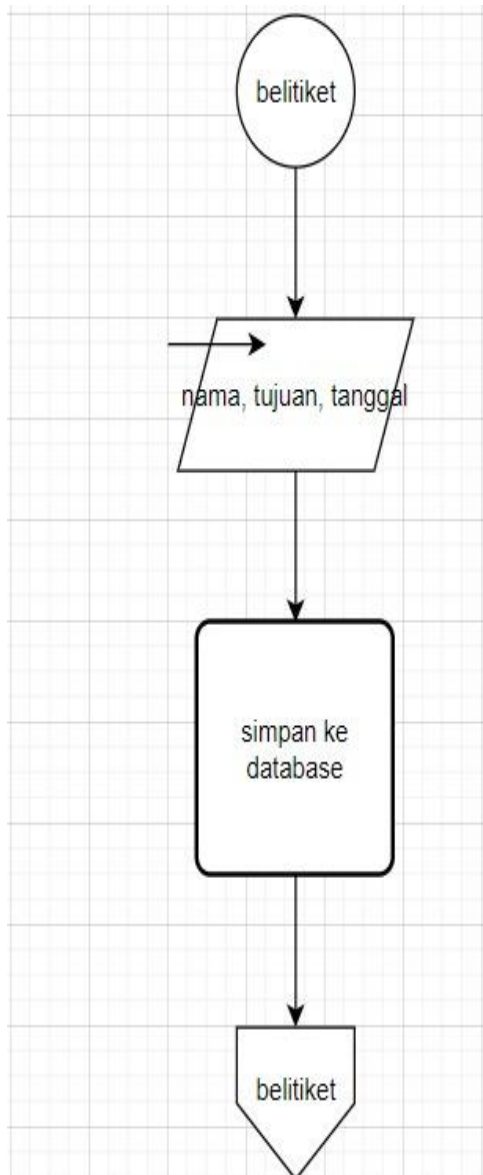
Gambar 1.1



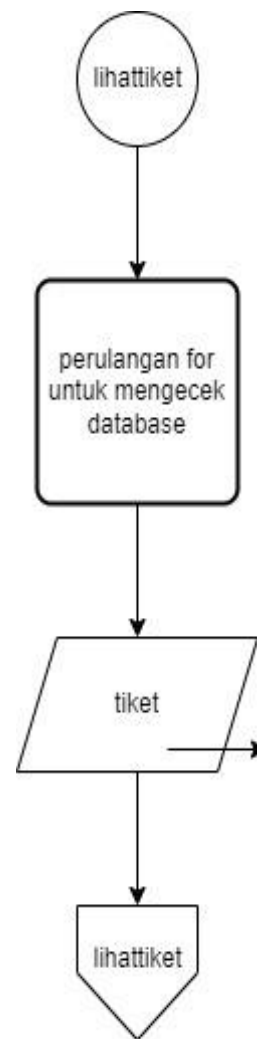
Gambar 1.2



Gambar 1.3



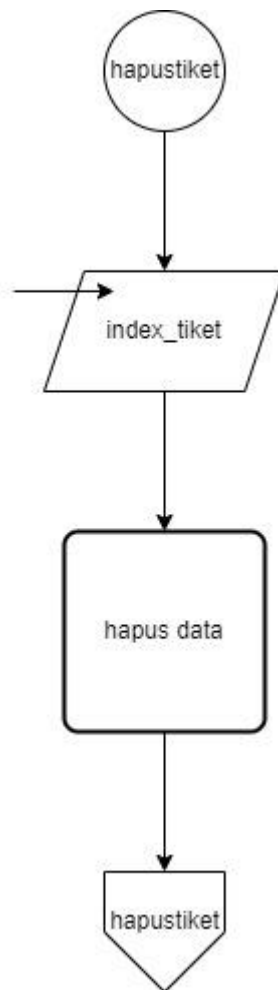
Gambar 1.4



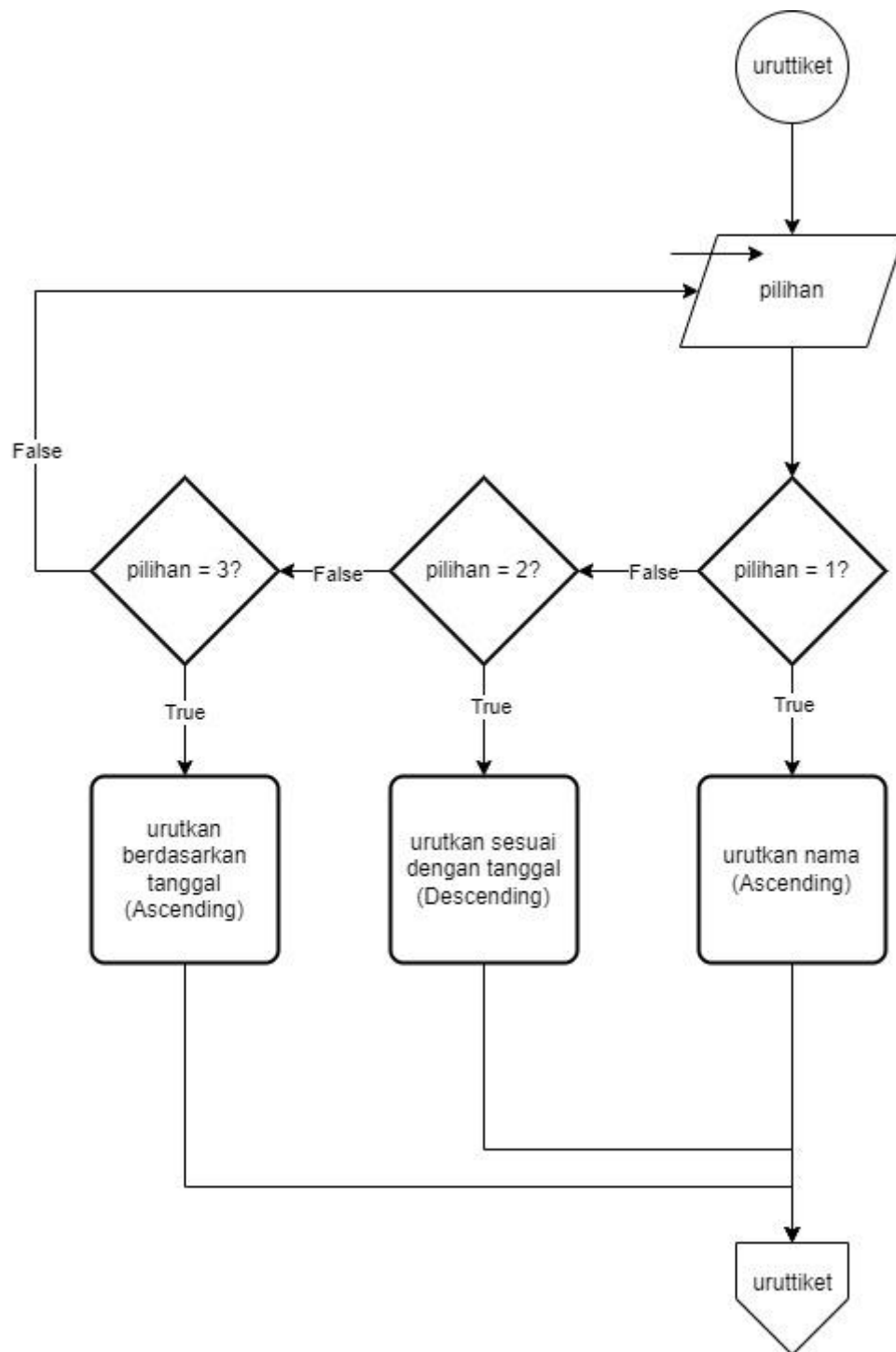
Gambar 1.5



Gambar 1.6



Gambar 1.7



Gambar 1.8

2. Analysis Program

2.1 Deskripsi Singkat Program

Program ini merupakan berbasis C++ yang digunakan untuk mengelola pemesanan tiket kapal. Program memungkinkan pengguna untuk melakukan berbagai aktivitas seperti membeli tiket, melihat daftar tiket, mengubah data tiket, menghapus tiket, serta mengurutkan tiket berdasarkan kriteria tertentu. Sebelum menggunakan system, pengguna diwajibkan untuk login menggunakan nama dan Nim yang sudah ditentukan.

2.2 Penjelasan Alur & Algoritma

- Alur Program

- 1) Program Dimulai

Program langsung menjalankan fungsi login()'

Pengguna diminta:

- Menginput Nama.
- Menginput NIM.

Jika nama dan Nim benar Login Berhasil.

Jika salah, pengguna diberi maksimal 3 kali percobaan. Jika tetap salah program akan keluar.

- 2) Masuk ke Menu Utama

Setelah login, program masuk ke menu utama yang berisi pilihan:

- Beli Tiket Kapal
- Lihat Daftar Tiket
- Ubah Tiket
- Hapus Tiket
- Sorting Tiket
- Keluar Program

Menu akan berulang (menggunakan do – while) selama user belum memilih keluar.

- 3) Aksi Berdasarkan Pilihan Menu

- Menu 1 (Beli Tiket Kapal)

Memanggil fungsi inputTiket() untuk mengisi data: nama pemesan, tujuan, dan tanggal pelayaran.

Data disimpan ke dalam array namaPemesanan, tujuanPelayaran, dan tanggalpelayaran.

- Menu 2 (Lihat Daftar Tiket)

Menampilkan seluruh tiket yang sudah tersimpan di array.

- Menu 3 (Ubah Tiket)

Memanggil fungsi ubahTiket(&nomor) berdasarkan nomor tiket yang ingin diubah. Data lama akan ditampilkan, lalu user memasukkan data baru.

- Menu 4 (Hapus Tiket)

Memanggil fungsi hapusTiket(&nomor). Data dihapus menggunakan rekursi: data setelah tiket yang dihapus akan bergeser satu posisi ke atas.

- Menu 5 (Sorting Tiket)

Menyediakan fitur sorting berdasarkan:

- Nama pemesanan (A-Z) menggunakan Bubble Sort.
- Tanggal pelayaran (terbaru ke terlama) menggunakan Selection Sort.
- Tujuan pelayaran (A_Z) menggunakan Insertion Sort.

- Penjelasan Algoritma

- 1) Login Sistem

- Input: Nama dan NIM.
- Proses: Bandingkan input user data yang sudah ditentukan.
- Output:
 - Jika cocok, login berhasil
 - Jika salah 3 kali, program keluar.

- 2) Tambah Tiket (inputTiket dan tambahTiket)

- Input: Nama pemesanan, tujuan, tanggal.
- Proses:
 - Data disimpan dalam array.
 - Jumlah tiket (jumlahTiket) bertambah 1.
- Output: Data tiket berhasil ditambahkan.

- 3) Tampilkan Tiket

- Proses:
 - Looping dari $I = 0$ hingga jumlahTiket - 1.
 - Tampilkan setiap tiket.

- 4) Ubah Tiket

- Input: Nomor tiket.
- Proses:
 - Cari tiket berdasarkan indeks (nomor - 1).
 - Update nama, tujuan, dan tanggal dengan data baru.

- 5) Hapus Tiket (dengan Rekursi)

- Input: Nomor tiket yang ingi dihapus.
- Proses:
 - Mulai dari indeks yang dihapus.
 - Geser semua data di belakangnya ke posisi sekarang.
 - Gunakan fungsi rekursif (hapusTiketrekursif(index + 1)).
 - Setelah selesai, jumlahTiket--.
- Output: Tiket berhasil dihapus.

- 6) Sorting Tiket

Program memiliki 3 metode sorting

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 login (parameter reference)
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 }
39
40 // Fungsi tambah tiket (parameter dereference pointer)
41 void tambahTiket(string* nama, string* tujuan, string* tanggal) {
42     namaPemesan[jumlahTiket] = *nama;
43     tujuanPelayaran[jumlahTiket] = *tujuan;
44     tanggalPelayaran[jumlahTiket] = *tanggal;
45     jumlahTiket++;
46 }
47
48 void inputTiket() {
49     if (jumlahTiket >= MAKS_TIKET) {
50         cout << "Tiket sudah penuh!\n";
51         return;
52     }
53
54     string nama, tujuan, tanggal;
55     cout << "Masukkan Nama Pemesan: ";
56     getline(cin, nama);
57     cout << "Masukkan Tujuan Pelayaran: ";
58     getline(cin, tujuan);
59     cout << "Masukkan Tanggal Pelayaran (DD-MM-YYYY): ";
60     getline(cin, tanggal);
```

Gambar Code 2.1

```

1  // Fungsi tambah tiket (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     tambahTiket(&nama, &tujuan, &tanggal);
24     cout << "Tiket berhasil dipesan!\n";
25 }
26
27 void tampilkanTiket

```

Gambar Code 2.2



```

1  void ubahTiket(int* nomor) {
2      if (*nomor < 1 || *nomor > jumlahTiket) {
3          cout << "Tiket tidak ditemukan.\n";
4          return;
5      }
6
7      int idx = *nomor - 1;
8      cout << "Data tiket saat ini:\n";
9      cout << "Nama: " << namaPemesan[idx] << "\n";
10     cout << "Tujuan: " << tujuanPelayaran[idx] << "\n";
11     cout << "Tanggal: " << tanggalPelayaran[idx] << "\n";
12
13     cout << "Masukkan Nama Baru: ";
14     getline(cin, namaPemesan[idx]);
15     cout << "Masukkan Tujuan Baru: ";
16     getline(cin, tujuanPelayaran[idx]);
17     cout << "Masukkan Tanggal Baru: ";
18     getline(cin, tanggalPelayaran[idx]);
19
20     cout << "Tiket berhasil diperbarui!\n";
21 }
22
23 void hapusTiketRekursif(int index) {
24     if (index >= jumlahTiket - 1) return;
25
26     namaPemesan[index] = namaPemesan[index + 1];
27     tujuanPelayaran[index] = tujuanPelayaran[index + 1];
28     tanggalPelayaran[index] = tanggalPelayaran[index + 1];
29
30     hapusTiketRekursif(index + 1);
31 }
32
33 void hapusTiket(int* nomor) {
34     if (*nomor < 1 || *nomor > jumlahTiket) {
35         cout << "Tiket tidak ditemukan.\n";
36         return;
37     }
38
39     hapusTiketRekursif(*nomor - 1);
40     jumlahTiket--;
41     cout << "Tiket

```

Gambar Code 2.3

```

1  // === FUNGSI SORTING ===
2
3  // 1. Sorting Nama Pemesan ASCENDING (Bubble Sort)
4  void sortingNamaAscending() {
5      for (int i = 0; i < jumlahTiket - 1; i++) {
6          for (int j = 0; j < jumlahTiket - i - 1; j++) {
7              if (namaPemesan[j] > namaPemesan[j + 1]) {
8                  swap(namaPemesan[j], namaPemesan[j + 1]);
9                  swap(tujuanPelayaran[j], tujuanPelayaran[j + 1]);
10                 swap(tanggalPelayaran[j], tanggalPelayaran[j + 1]);
11             }
12         }
13     }
14     cout << "Data berhasil diurutkan berdasarkan Nama (A-Z).\n";
15 }
16
17 // 2. Sorting Tanggal DESCENDING (Selection Sort)
18 void sortingTanggalDescending() {
19     for (int i = 0; i < jumlahTiket - 1; i++) {
20         int maxIdx = i;
21         for (int j = i + 1; j < jumlahTiket; j++) {
22             if (tanggalPelayaran[j] > tanggalPelayaran[maxIdx]) {
23                 maxIdx = j;
24             }
25         }
26         swap(namaPemesan[i], namaPemesan[maxIdx]);
27         swap(tujuanPelayaran[i], tujuanPelayaran[maxIdx]);
28         swap(tanggalPelayaran[i], tanggalPelayaran[maxIdx]);
29     }
30     cout << "Data berhasil diurutkan berdasarkan Tanggal (terbaru dulu).\n";
31 }
32
33 // 3. Sorting Tujuan Pelayaran ASCENDING (Insertion Sort)
34 void sortingTujuanAscending() {
35     for (int i = 1; i < jumlahTiket; i++) {
36         string keyNama = namaPemesan[i];
37         string keyTujuan = tujuanPelayaran[i];
38         string keyTanggal = tanggalPelayaran[i];
39         int j = i - 1;
40
41         while (j >= 0 && tujuanPelayaran[j] > keyTujuan) {
42             namaPemesan[j + 1] = namaPemesan[j];
43             tujuanPelayaran[j + 1] = tujuanPelayaran[j];
44             tanggalPelayaran[j + 1] = tanggalPelayaran[j];
45             j--;
46         }
47         namaPemesan[j + 1] = keyNama;
48         tujuanPelayaran[j + 1] = keyTujuan;
49         tanggalPelayaran[j + 1] = keyTanggal;
50     }

```

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. Urutkan Data Tiket\n";
12         cout << "6. Keluar\n";
13         cout << "Pilih menu: ";
14         cin >> pilihan;
15         cin.ignore();
16
17         switch (pilihan) {
18             case 1:
19                 inputTiket();
20                 break;
21             case 2:
22                 tampilkanTiket();
23                 break;
24             case 3: {
25                 int nomor;
26                 cout << "Masukkan nomor tiket yang ingin diubah: ";
27                 cin >> nomor;
28                 cin.ignore();
29                 ubahTiket(&nomor);
30                 break;
31             }
32             case 4: {
33                 int nomor;
34                 cout << "Masukkan nomor tiket yang ingin dihapus: ";
35                 cin >> nomor;
36                 cin.ignore();
37                 hapusTiket(&nomor);
38                 break;
39             }
40             case 5: {
41                 int pilihSort;
42                 cout << "\n=== Menu Sorting ===\n";
43                 cout << "1. Urutkan berdasarkan Nama (Ascending)\n";
44                 cout << "2. Urutkan berdasarkan Tanggal (Descending)\n";
45                 cout << "3. Urutkan berdasarkan Tujuan (Ascending)\n";
46                 cout << "Pilih metode sorting: ";
47                 cin >> pilihSort;
48                 cin.ignore();
49
50                 if (pilihSort == 1)
51                     sortingNamaAscending();
52                 else if (pilihSort == 2)
53                     sortingTanggalDescending();
54                 else if (pilihSort == 3)
55                     sortingTujuanAscending();
56                 else
57                     cout << "Pilihan sorting tidak valid!\n";
58
59                 break;
60             }
61             case 6:
62                 cout << "Terima kasih telah menggunakan sistem pemesanan tiket kapal.\n";
63                 break;
64             default:
65                 cout << "Pilihan tidak valid!\n";
66         }
67     } while (pilihan != 6);
68
69     return 0;
70 }
71
72

```

Gambar Code 2.5

5. Langkah Langkah Github

- Git add

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

- Git commit

```
PS D:\praktikum-apl> git commit -m "Finish Posttest 6"
[main 3549323] Finish Posttest 6
3 files changed, 247 insertions(+)
create mode 100644 post-test/post-test-apl-6/2409106083-MuhammadRayhanSeitiawan-PT-6.cpp
create mode 100644 post-test/post-test-apl-6/2409106083-MuhammadRayhanSeitiawan-PT-6.docx
create mode 100644 post-test/post-test-apl-6/2409106083-MuhammadRayhanSeitiawan-PT-6.exe
```

- Git push

```
PS D:\praktikum-apl> 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), 2.60 MiB | 1.03 MiB/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/rrrreeehannn/praktikum-apl.git
```