

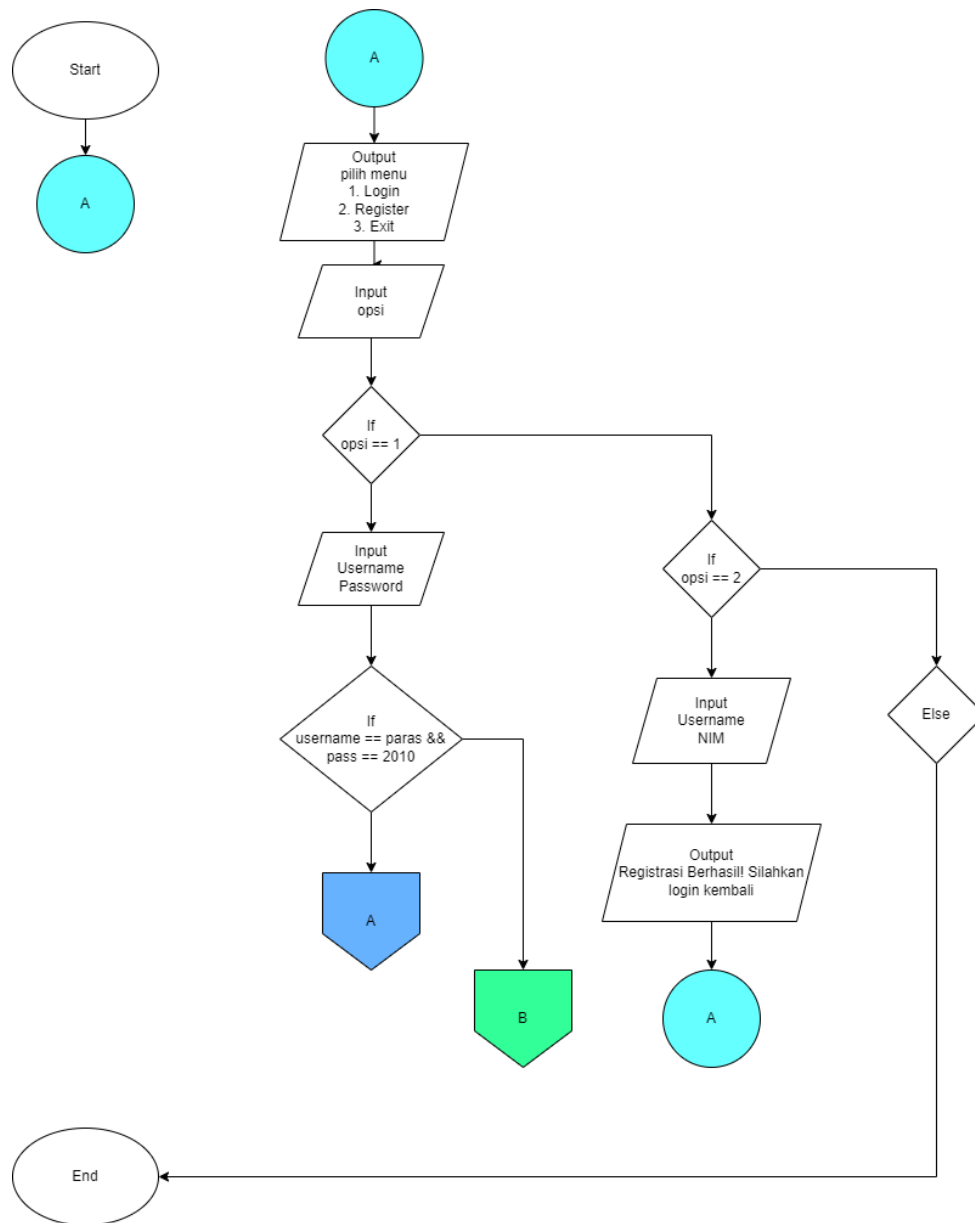
LAPORAN PRAKTIKUM
POSTTEST V
ALGORITMA PEMROGRAMAN LANJUT



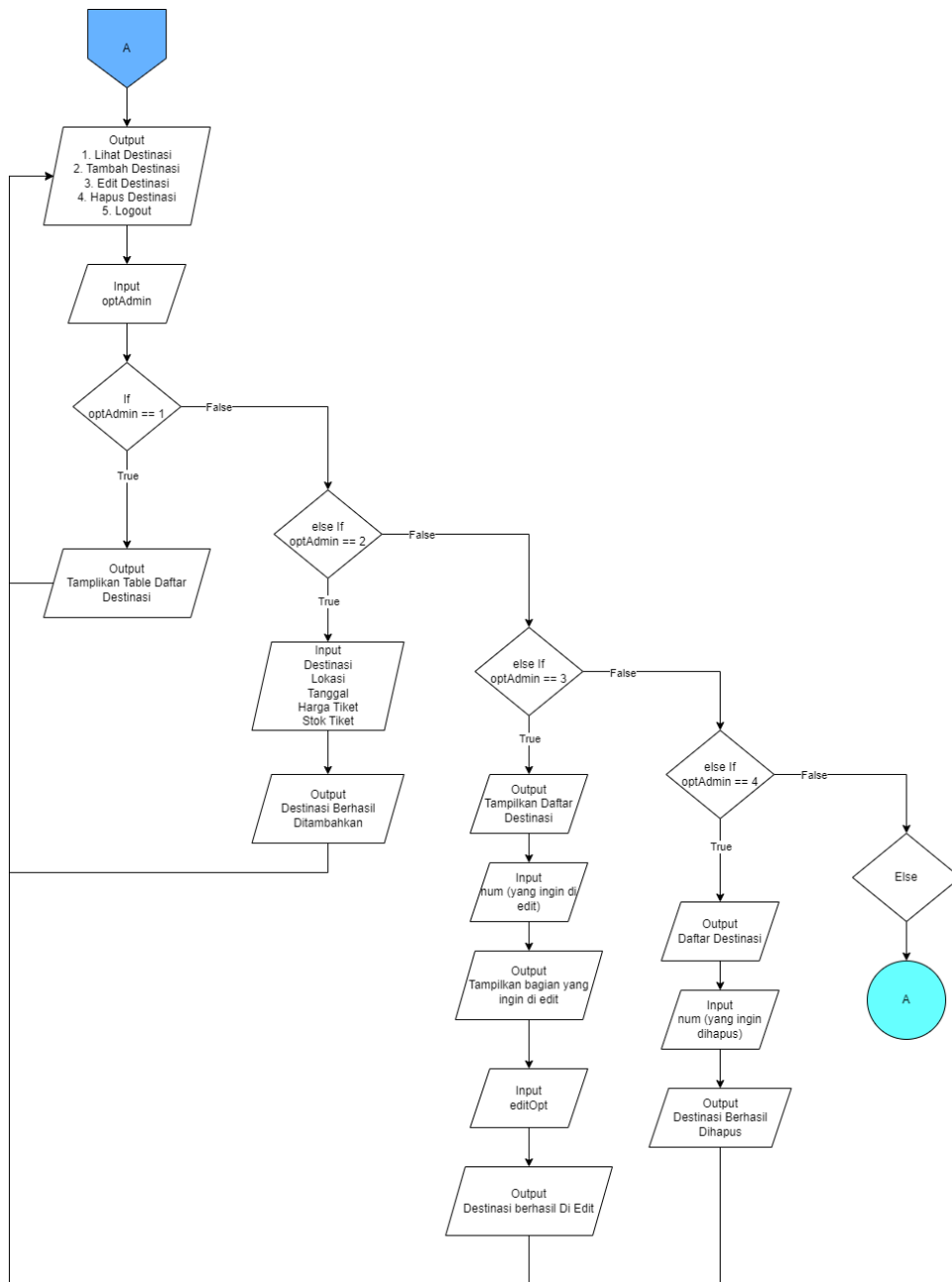
Disusun oleh:
Muhammad Farras Arhab Ince (2409106092)
Kelas (C1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

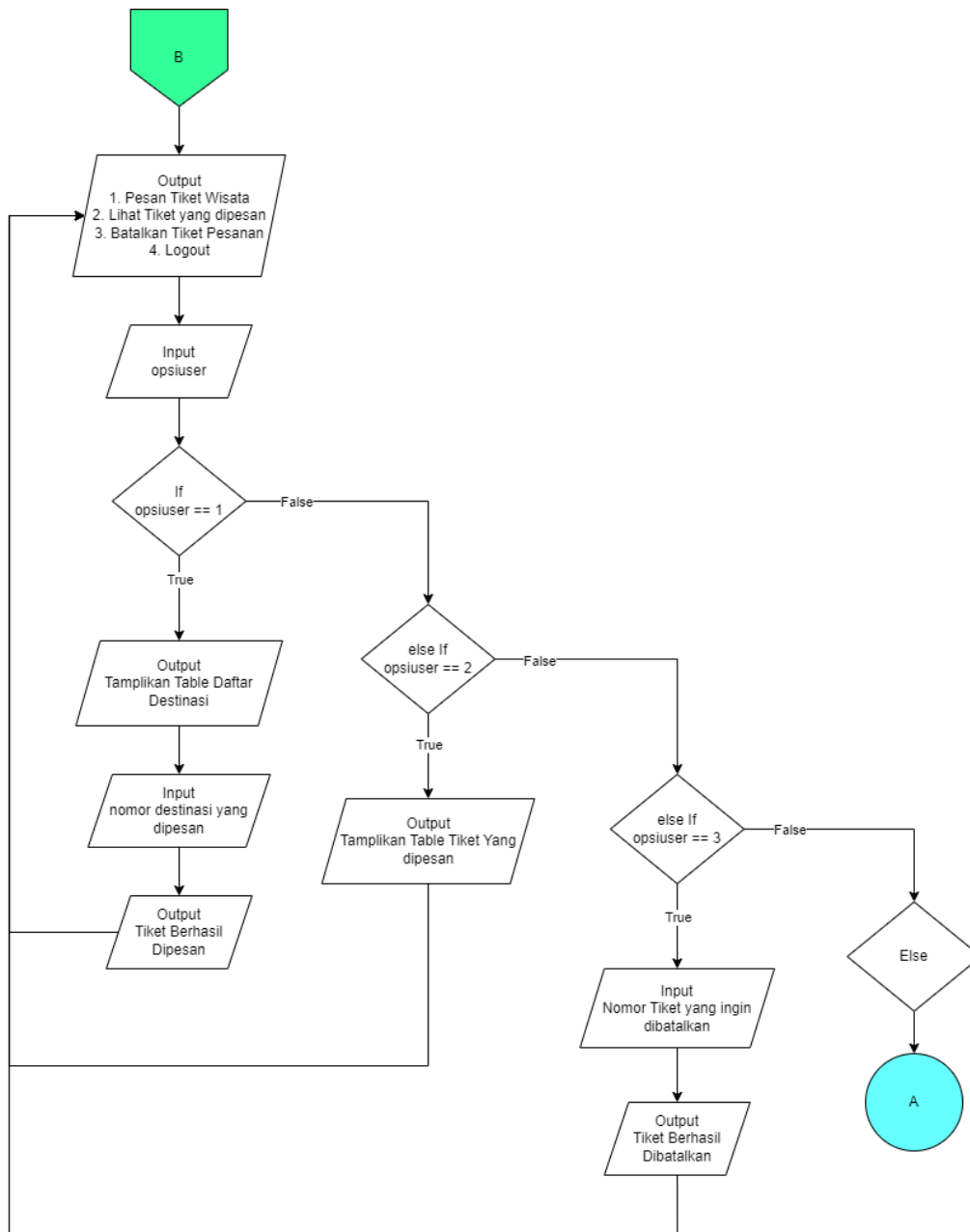
1. Flowchart



Gambar 1.1 Flowchart awal



Gambar 1.2 Flowchart Menu Admin



Gambar 1.3 Flowchart Menu Pengguna

2. Analisis Program

Program ini dibuat untuk memudahkan proses pemesanan tiket wisata dengan menggunakan sistem multiuser. Artinya, ada dua jenis pengguna yaitu admin dan pengguna biasa. Program ini memiliki beberapa tujuan dan fungsi utama sebagai berikut:

1. Pembuatan Akun:

Program menyediakan fitur register agar siapa saja bisa membuat akun baru. Saat pengguna mendaftar, data seperti nama dan NIM disimpan ke dalam array of struct bernama Akun. Akun yang dibuat melalui register secara otomatis dikategorikan sebagai pengguna biasa. Dengan demikian, setiap orang yang mendaftar tidak perlu melalui proses pembuatan akun secara terpisah karena data akun sudah tersimpan di program.

2. Sistem Login:

Setelah mendaftar, pengguna harus melakukan login dengan memasukkan nama dan NIM. Proses login dilakukan dengan memeriksa data yang dimasukkan apakah cocok dengan data yang ada di array akun. Jika data yang dimasukkan sesuai, maka pengguna bisa mengakses sistem. Terdapat batas maksimum tiga kali percobaan login untuk menghindari kesalahan input yang terus-menerus.

3. Pembagian Hak Akses (Admin vs Pengguna):

Program ini membedakan dua jenis pengguna.

a. Admin:

Admin adalah akun default yang sudah ada sejak awal (dalam kode, admin diwakili oleh akun pertama). Admin memiliki hak akses lebih untuk mengelola data destinasi wisata, misalnya bisa menambah, mengedit, dan menghapus data destinasi.

b. Pengguna:

Pengguna biasa yang mendaftar melalui register hanya dapat melakukan pemesanan tiket, melihat tiket yang sudah dipesan, dan membatalkan tiket jika diperlukan. Dengan demikian, sistem membatasi akses berdasarkan peran masing-masing.

4. Pengelolaan Data Wisata:

Data destinasi wisata disimpan dalam sebuah array dua dimensi yang berisi informasi seperti:

- a. Nama destinasi
- b. Lokasi
- c. Tanggal pelaksanaan wisata
- d. Harga tiket
- e. Stok tiket yang tersedia

Admin memiliki kemampuan untuk mengubah data ini melalui menu khusus, sehingga jika ada destinasi baru atau ada perubahan informasi, admin dapat meng-update data dengan mudah.

5. Proses Pemesanan dan Pembatalan Tiket:

Pengguna biasa dapat melihat daftar destinasi wisata yang tersedia dan memilih destinasi yang ingin dipesan tiketnya. Saat memesan, jumlah tiket yang diinginkan akan diperiksa apakah masih tersedia berdasarkan stok yang ada. Jika cukup, maka tiket berhasil dipesan dan stok tiket akan berkurang. Begitu juga jika pengguna membatalkan tiket, maka stok tiket akan dikembalikan sesuai dengan jumlah tiket yang dibatalkan.

Secara keseluruhan, program ini membantu mengatur dan memudahkan proses pemesanan tiket wisata dengan menggunakan sistem login dan register. Program ini juga memastikan bahwa hanya pengguna yang terverifikasi yang dapat mengakses menu sesuai dengan hak aksesnya, sehingga admin dapat mengelola data destinasi dengan lebih baik dan pengguna biasa dapat melakukan pemesanan tiket dengan mudah.

3. Source Code

A. Menu Pengguna

Fungsi MenuPengguna ini berfungsi untuk memberikan menu interaksi bagi pengguna dalam sistem pemesanan tiket wisata. Pengguna dapat melakukan beberapa tindakan, seperti memesan tiket, melihat tiket yang sudah dipesan, membatalkan pesanan tiket, dan logout.

```
void MenuPengguna(Destinasia wisata[], int &wisataCount, Booking bookings[], int &bookingCount) {
    int opsiuser;
    while (true) {
        cout << "\n===== Menu Pengguna =====\n";
        cout << "1. Pesan Tiket Wisata\n";
        cout << "2. Lihat Tiket yang Dipesan\n";
        cout << "3. Batalkan Tiket Pesanan\n";
        cout << "4. Logout\n";
        cout << "=====\n";
        cout << "Pilih opsi: ";
        cin >> opsiuser;
        cin.ignore();

        switch(opsiuser) {
            case 1: {
                tampilkan(wisata, wisataCount);
                cout << "Nomor destinasi yang ingin dipesan: ";
                int num;
                cin >> num;
                cin.ignore();
                if (num >= 1 && num <= wisataCount) {
                    Destinasi *dipilih = &wisata[num - 1];
                    cout << "Jumlah tiket yang ingin dipesan: ";
                    int jumlah;
                    cin >> jumlah;
                    cin.ignore();
                    if (jumlah <= dipilih->stok) {
                        bookings[bookingCount] = {dipilih->nama, dipilih->lokasi, dipilih->tanggal, dipilih->harga, jumlah};
                        bookingCount++;
                        dipilih->stok -= jumlah;
                        cout << "Tiket berhasil dipesan!\n";
                    } else {
                        cout << "Stok tiket tidak mencukupi.\n";
                    }
                } else {
                    cout << "Nomor tidak valid.\n";
                }
                break;
            }
        }
    }
}
```

```

        case 2:
            tampilkan(bookings, bookingCount);
            break;
        case 3: {
            cout << "\nDaftar Tiket yang Dipesan:\n";
            for (int i = 0; i < bookingCount; i++) {
                cout << i + 1 << ". " << bookings[i].destinasi << " (" <<
bookings[i].jumlah << " tiket)\n";
            }
            cout << "Nomor pesanan yang ingin dibatalkan: ";
            int num;
            cin >> num;
            cin.ignore();
            if (num >= 1 && num <= bookingCount) {
                int indeks = num - 1;
                for (int j = 0; j < wisataCount; j++) {
                    if (wisata[j].nama == bookings[indeks].destinasi) {
                        tambahStok(&wisata[j], bookings[indeks].jumlah);
                        break;
                    }
                }
                cetakBooking(&bookings[indeks]);
                for (int i = indeks; i < bookingCount - 1; i++) {
                    bookings[i] = bookings[i + 1];
                }
                bookingCount--;
                cout << "Pesanan berhasil dibatalkan.\n";
            } else {
                cout << "Nomor tidak valid.\n";
            }
            break;
        }
        case 4:
            return;
        default:
            cout << "Opsi tidak valid.\n";
    }
}
}

```

B. Fungsi tambah stok dengan parameter address-of.

Fungsi ini akan menambahkan jumlah tiket yang telah dibeli ke dalam stok yang tersedia. Jadi, jika seseorang membatalkan pemesanan tiket, atau ada tambahan stok, fungsi ini akan menambah stok tiket yang ada.


```
// Fungsi pointer dengan parameter address-of
void tambahStok(Destinas *d, int jumlah) {
    d->stok += jumlah;
}
```

C. Fungsi mencetak booking dengan parameter dereference.

Fungsi ini bertujuan untuk menampilkan detail pemesanan tiket yang telah dilakukan oleh pengguna. Fungsi ini menggunakan pointer ke tipe data Booking yang berisi informasi mengenai tiket yang telah dibeli bahwa sudah di batalkan di fungsi menupengguna switch case3.

```
// Fungsi pointer dengan parameter dereference
void cetakBooking(Booking *b) {
    cout << "\n==== Detail Tiket Booking =====\n";
    cout << "Destinasi: " << b->destinasi << endl;
    cout << "Lokasi: " << b->lokasi << endl;
    cout << "Tanggal: " << b->tanggal << endl;
    cout << "Harga: Rp" << b->harga << endl;
    cout << "Jumlah Tiket: " << b->jumlah << endl;
}
```

4. Uji Coba dan Hasil Output

```
===== Menu Pengguna =====
1. Pesan Tiket wisata
2. Lihat Tiket yang Dipesan
3. Batalkan Tiket Pesanan
4. Logout
=====
Pilih opsi: 1

----- Daftar Destinasi Wisata -----
+-----+-----+-----+-----+-----+
| NO | Destinasi | Lokasi | Tanggal | Harga | Tiket Tersedia |
+-----+-----+-----+-----+-----+
| 1 | Pulau Derawan | Berau | 10-April-2025 | 1500000 | 50 |
| 2 | Bukit Bangkirai | Kukar | 20-Mei-2025 | 750000 | 30 |
| 3 | Kepulauan Maratua | Berau | 15-Juni-2025 | 2000000 | 25 |
| 4 | Pantai Melawai | Balikpapan | 5-Juli-2025 | 500000 | 40 |
| 5 | Danau Labuan Cermin | Biduk-biduk | 1-Agustus-2025 | 1000000 | 20 |
+-----+-----+-----+-----+-----+

Nomor destinasi yang ingin dipesan: 1
Jumlah tiket yang ingin dipesan: 2
Tiket berhasil dipesan!
```

Gambar 4.1 Output Pesan Tiket

----- Daftar Destinasi Wisata -----					
NO	Destinasi	Lokasi	Tanggal	Harga	Tiket Tersedia
1	Pulau Derawan	Berau	10-April-2025	1500000	48
2	Bukit Bangkirai	Kukar	20-Mei-2025	750000	30
3	Kepulauan Maratua	Berau	15-Juni-2025	2000000	25
4	Pantai Melawai	Balikpapan	5-Juli-2025	500000	40
5	Danau Labuan Cermin	Biduk-biduk	1-Agustus-2025	1000000	20

Gambar 4.2 Output Destinasi Wisata setelah di booking

```

===== Menu Pengguna =====
1. Pesan Tiket Wisata
2. Lihat Tiket yang Dipesan
3. Batalkan Tiket Pesanan
4. Logout
=====
Pilih opsi: 3

Daftar Tiket yang Dipesan:
1. Pulau Derawan (2 tiket)
Nomor pesanan yang ingin dibatalkan: 1

===== Detail Tiket Booking =====
Destinasi: Pulau Derawan
Lokasi: Berau
Tanggal: 10-April-2025
Harga: Rp1500000
Jumlah Tiket: 2
Pesanan berhasil dibatalkan.

```

Gambar 4.3 Output Batalkan Pemesanan

5. Langkah-Langkah Git pada VSCode

```
Farra@LAPTOP-SQ4287S4 MINGW64 /d/Kuliah/SEMESTER 2/ALGORITMA PEMROGRAMAN LANJUT
(APL)/PRAKTIKUM APL/praktikum-apl (main)
$ git add .

Farra@LAPTOP-SQ4287S4 MINGW64 /d/Kuliah/SEMESTER 2/ALGORITMA PEMROGRAMAN LANJUT
(APL)/PRAKTIKUM APL/praktikum-apl (main)
$ git commit -m "kumpul kode"
[main 1949366] kumpul kode
1 file changed, 1 insertion(+), 1 deletion(-)

Farra@LAPTOP-SQ4287S4 MINGW64 /d/Kuliah/SEMESTER 2/ALGORITMA PEMROGRAMAN LANJUT
(APL)/PRAKTIKUM APL/praktikum-apl (main)
$ git push origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 444 bytes | 444.00 KiB/s, done.
Total 5 (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/parashrb/praktikum-apl.git
   36c01ef..1949366  main -> main

Farra@LAPTOP-SQ4287S4 MINGW64 /d/Kuliah/SEMESTER 2/ALGORITMA PEMROGRAMAN LANJUT
(APL)/PRAKTIKUM APL/praktikum-apl (main)
$
```

Gambar 5 Terminal Git Bash

Gambar hanya pemanis sebab saya lupa screenshot saat benar benar mengumpul, jadi saya bikin ulang. Berikut merupakan penjelasannya:

1. `git add .` digunakan untuk menambahkan semua perubahan yang telah dibuat ke dalam staging area agar siap dikomit.
2. Perintah `git commit -m` digunakan untuk menyimpan perubahan yang telah ditambahkan ke staging area ke dalam repository Git, sekaligus memberikan pesan singkat yang menjelaskan tujuan atau isi dari commit tersebut.
3. Perintah `git push origin main` digunakan untuk mengirim (push) commit yang ada di branch main pada repository lokal ke repository remote (seperti GitHub) agar perubahan dapat tersimpan dan diakses secara online.