

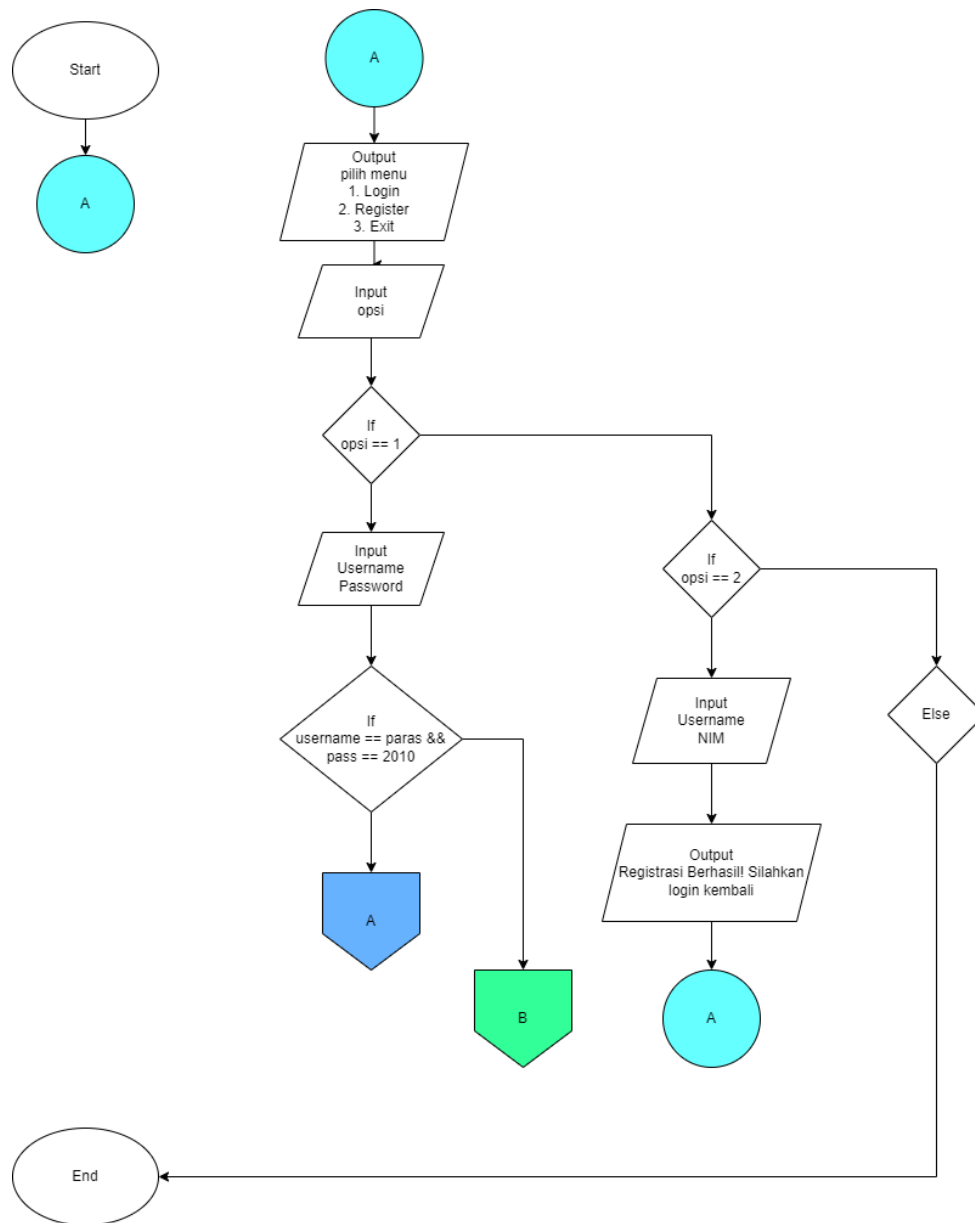
LAPORAN PRAKTIKUM
POSTTEST VI
ALGORITMA PEMROGRAMAN LANJUT



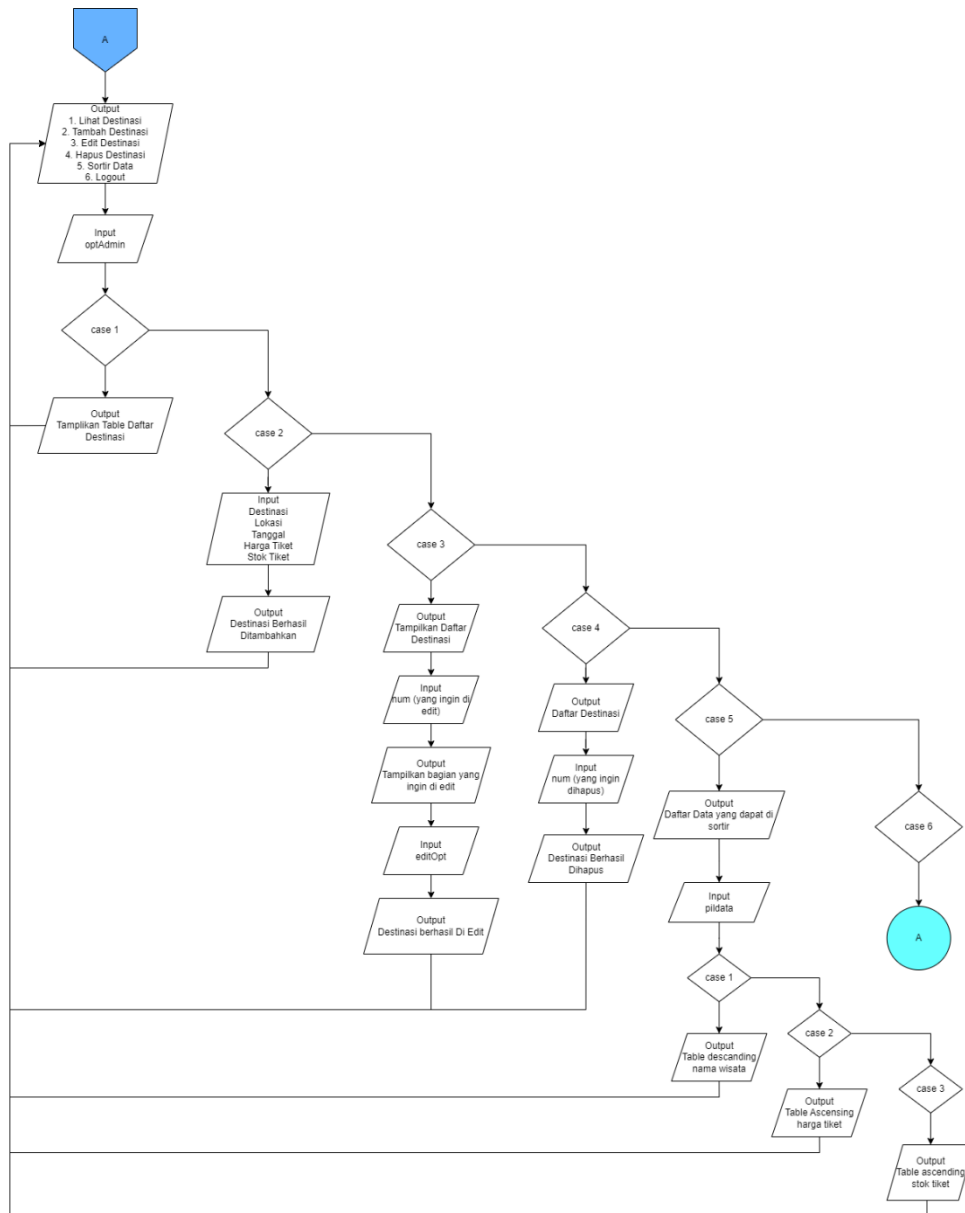
Disusun oleh:
Muhammad Farras Arhab Ince (2409106092)
Kelas (C1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

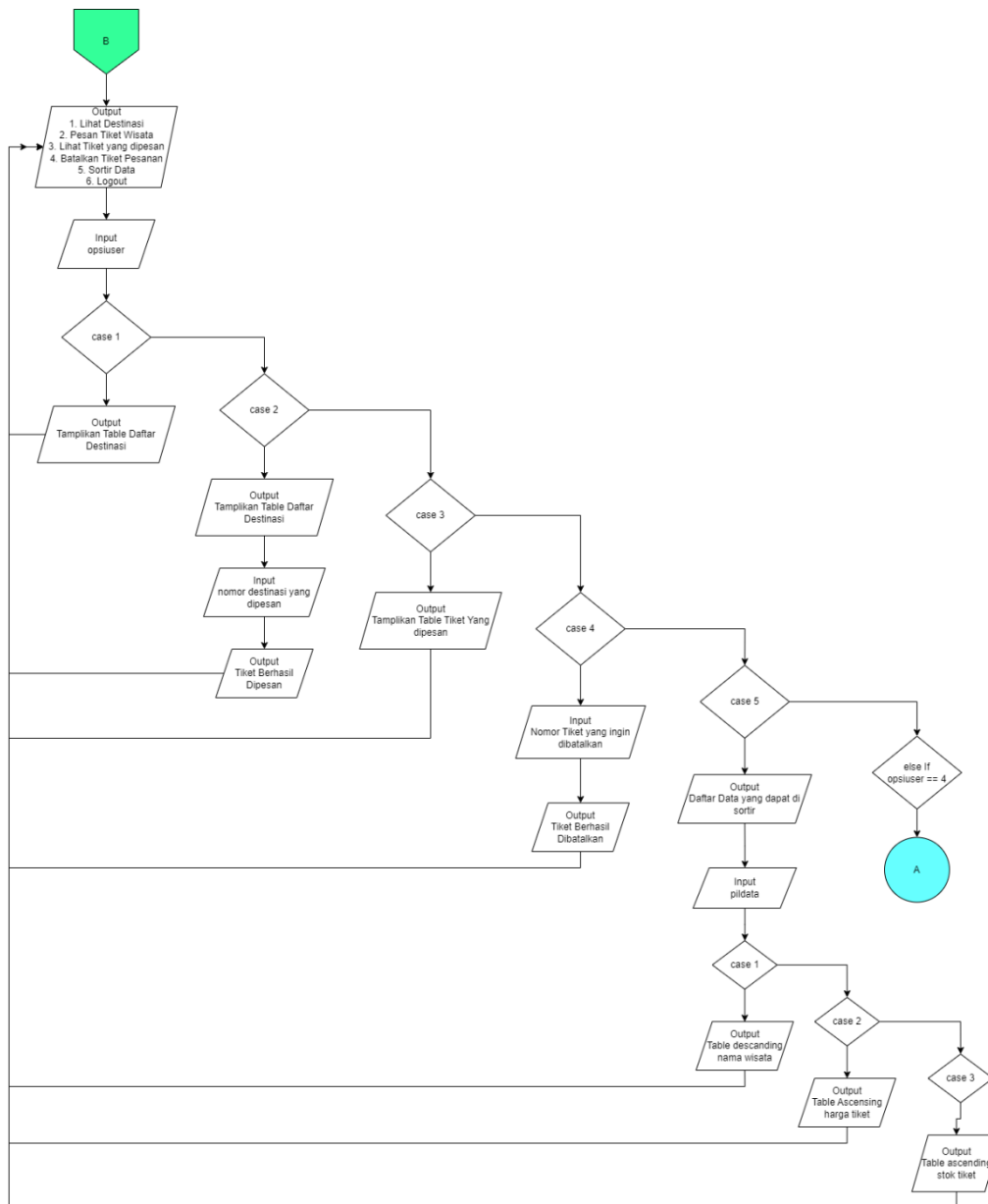
1. Flowchart



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, menghapus data destinasi, serta mensortir data agar memudahkan admin untuk melihat data berdasarkan Nama, Harga, dan Stok tiket.

b. Pengguna:

Pengguna biasa yang mendaftar melalui register hanya dapat melakukan pemesanan tiket, melihat tiket yang sudah dipesan, membatalkan tiket jika diperlukan, dan mensortir data agar mudah melihat destinasi yang akan di pesan. 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. Fungsi Sortir Nama Destinasi secara Descending dengan merge sort

Fungsi ini digunakan untuk mengurutkan destinasi berdasarkan nama dari Z ke A menggunakan algoritma Merge Sort. Hal ini memudahkan admin atau pengguna melihat daftar destinasi secara alfabetik terbalik.

```
// Merge Sort Nama, Destinasi (Descending Z - A)
void mergeNamaDesc(Destinas arr[], int l, int m, int r) {
    int n1 = m - 1 + 1, n2 = r - m;
    Destinas L[n1], R[n2];
    for (int i = 0; i < n1; i++) L[i] = arr[l + i];
    for (int j = 0; j < n2; j++) R[j] = arr[m + 1 + j];
    int i = 0, j = 0, k = l;
    while (i < n1 && j < n2) {
        if (L[i].nama > R[j].nama) arr[k++] = L[i++];
        else
            arr[k++] = R[j++];
    }
    while (i < n1) arr[k++] = L[i++];
    while (j < n2) arr[k++] = R[j++];
}

void mergeSortNamaDesc(Destinas arr[], int l, int r) {
    if (l < r) {
        int m = l + (r - l) / 2;
        mergeSortNamaDesc(arr, l, m);
        mergeSortNamaDesc(arr, m + 1, r);
        mergeNamaDesc(arr, l, m, r);
    }
}
```

B. Fungsi Sortir Harga Destinasi secara Ascending dengan Bubble sort

Fungsi ini digunakan untuk mengurutkan destinasi berdasarkan harga tiket dari termurah ke termahal menggunakan algoritma Bubble Sort. Ini memudahkan pengguna melihat destinasi wisata berdasarkan harga.

```
// Bubble Sort Angka, Harga (Ascending 0 - 9)
void bubbleSortHargaAsc(Destinas arr[], int n) {
    for (int i = 0; i < n-1; i++) {
        for (int j = 0; j < n-i-1; j++) {
            if (stoi(arr[j].harga) > stoi(arr[j+1].harga)) {
                swap(arr[j], arr[j+1]);
            }
        }
    }
}
```

C. Fungsi Sortir Stok Tiket secara Ascending dengan Selection sort

Fungsi ini digunakan untuk mengurutkan destinasi wisata berdasarkan jumlah stok tiket dari yang paling sedikit ke yang paling banyak, menggunakan Selection Sort.

```
// Selection Sort Angka, Stok Tiket (Ascending 0 - 9)
void selectionSortStokAsc(Destinas arr[], int n) {
    for (int i = 0; i < n-1; i++) {
        int minIdx = i;
        for (int j = i+1; j < n; j++) {
            if (arr[j].stok < arr[minIdx].stok) {
                minIdx = j;
            }
        }
        if (minIdx != i) {
            swap(arr[i], arr[minIdx]);
        }
    }
}
```

4. Uji Coba dan Hasil Output

Data Yang Ingin Disortir:
1. Nama Destinasi (Z - A)
2. Harga Tiket (0 - 9)
3. Tiket Tersedia (0 - 9)
Pilih Data: 1
Berhasil sortir berdasarkan Nama (Z - A)

----- Daftar Destinasi Wisata -----

NO	Destinasi	Lokasi	Tanggal	Harga	Tiket Tersedia
1	Taman Pintar	Danau Aji	5-November-2025	10000	90
2	Pulau Paus	Tenggarong	20-Oktober-2025	1000	80
3	Pulau Kumala	Timbau	4-Juli-2025	30000	100
4	Pulau Derawan	Berau	10-April-2025	1500000	50
5	Pantai Melawai	Balikpapan	5-Juli-2025	500000	40
6	Ladaya	Mangkurawang	2-Oktober-2025	60000	50
7	Kepulauan Maratua	Berau	15-Juni-2025	2000000	25
8	Danau Labuan Cermin	Biduk-biduk	1-Agustus-2025	1000000	20
9	Bukit Bangkirai	Kukar	20-Mei-2025	750000	30

Gambar 4.1 Output Sortir Nama Destinasi (Z - A)

Data Yang Ingin Disortir:
 1. Nama Destinasi (Z - A)
 2. Harga Tiket (0 - 9)
 3. Tiket Tersedia (0 - 9)
 Pilih Data: 1
 Berhasil sortir berdasarkan Nama (Z - A)

----- Daftar Destinasi Wisata -----

NO	Destinasi	Lokasi	Tanggal	Harga	Tiket Tersedia
1	Taman Pintar	Danau Aji	5-November-2025	10000	90
2	Pulau Paus	Tenggarong	20-Okttober-2025	1000	80
3	Pulau Kumala	Timbau	4-Juli-2025	30000	100
4	Pulau Derawan	Berau	10-April-2025	1500000	50
5	Pantai Melawai	Balikpapan	5-Juli-2025	500000	40
6	Ladaya	Mangkurawang	2-Okttober-2025	60000	50
7	Kepulauan Maratua	Berau	15-Juni-2025	2000000	25
8	Danau Labuan Cermin	Biduk-biduk	1-Agustus-2025	1000000	20
9	Bukit Bangkirai	Kukar	20-Mei-2025	750000	30

Gambar 4.2 Output Sortir Harga Tiket (0 - 9)

Data Yang Ingin Disortir:
 1. Nama Destinasi (Z - A)
 2. Harga Tiket (0 - 9)
 3. Tiket Tersedia (0 - 9)
 Pilih Data: 1
 Berhasil sortir berdasarkan Nama (Z - A)

----- Daftar Destinasi Wisata -----

NO	Destinasi	Lokasi	Tanggal	Harga	Tiket Tersedia
1	Taman Pintar	Danau Aji	5-November-2025	10000	90
2	Pulau Paus	Tenggarong	20-Okttober-2025	1000	80
3	Pulau Kumala	Timbau	4-Juli-2025	30000	100
4	Pulau Derawan	Berau	10-April-2025	1500000	50
5	Pantai Melawai	Balikpapan	5-Juli-2025	500000	40
6	Ladaya	Mangkurawang	2-Okttober-2025	60000	50
7	Kepulauan Maratua	Berau	15-Juni-2025	2000000	25
8	Danau Labuan Cermin	Biduk-biduk	1-Agustus-2025	1000000	20
9	Bukit Bangkirai	Kukar	20-Mei-2025	750000	30

Gambar 4.3 Output Sortir Stok Tiket (0 - 9)

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 (lagi..) 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.