

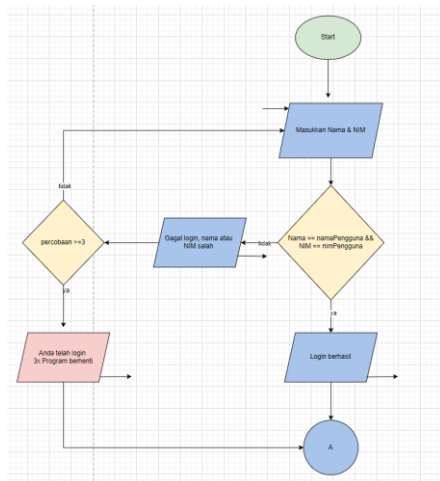
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



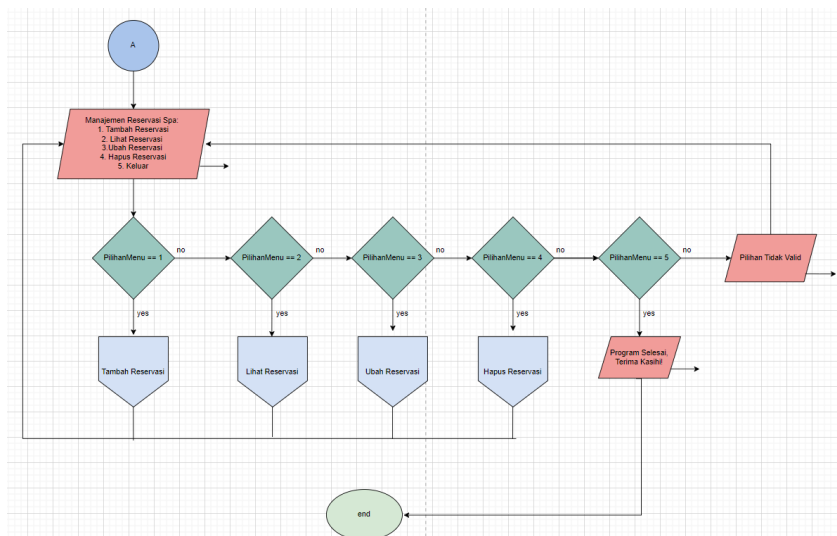
Disusun oleh:
Intan Alfara Audia 2409106008
A1'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

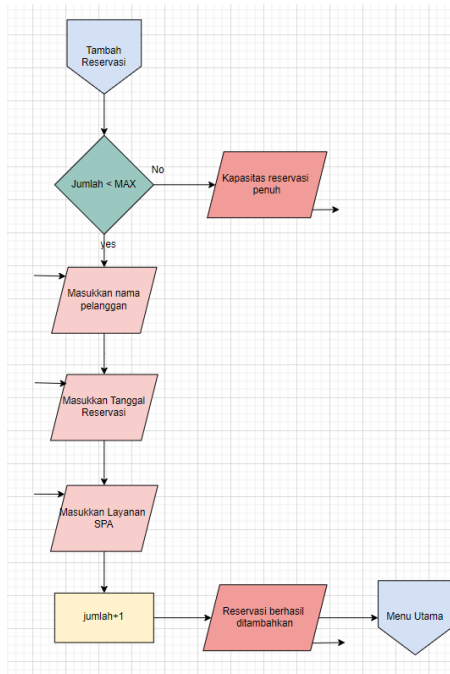
1. Flowchart



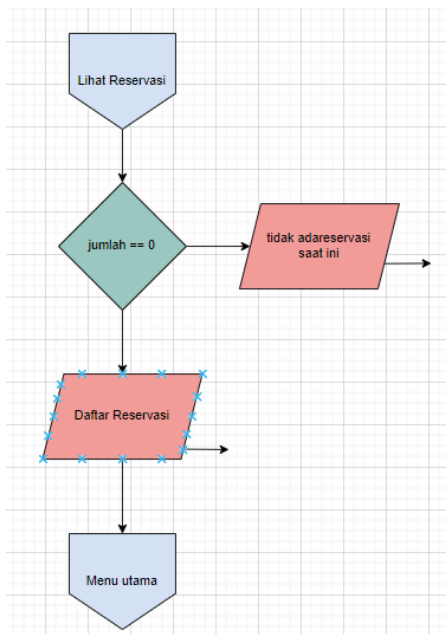
Gambar 1.1 flowchart fitur login



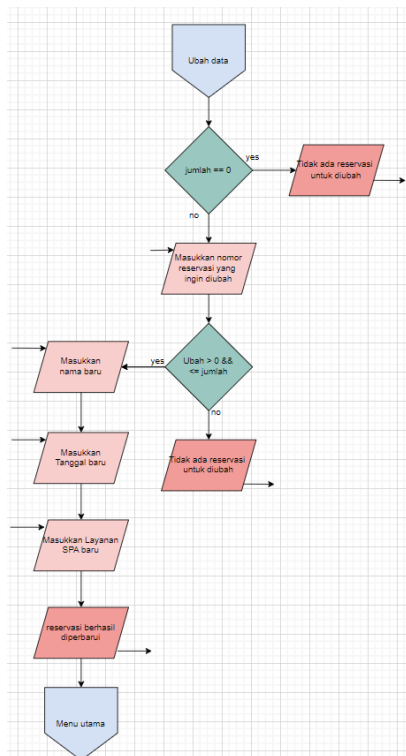
Gambar 1.2 Flowchart menu utama



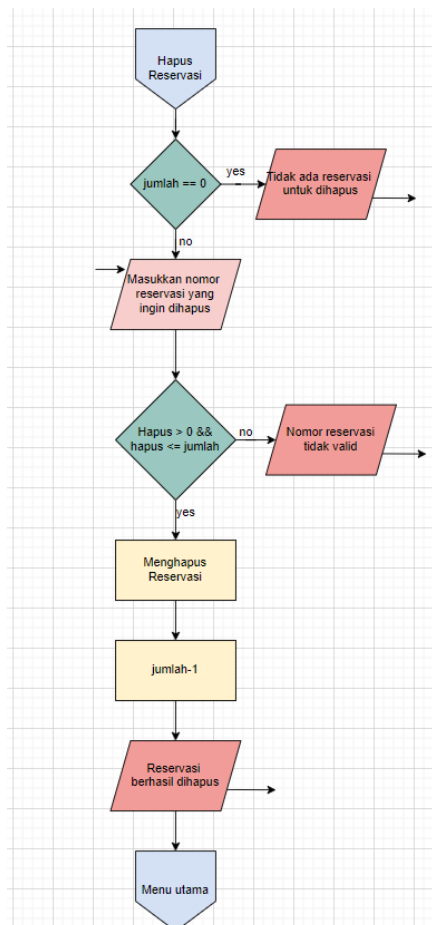
Gambar 1.3 Flowchart menambah reservasi



Gambar 1.4 Melihat reservasi



Gambar 1.5 Flowchart Mengubah reservasi



Gambar 1.6 Flowchart Menghapus reservasi

2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini dirancang untuk mempermudah pengelolaan reservasi layanan spa dengan menyediakan sistem berbasis teks yang sederhana dan efisien. Melalui fitur login, program memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses dan mengelola data reservasi. Pengguna dapat menambahkan reservasi dengan mencatat nama pelanggan, tanggal reservasi, serta jenis layanan spa yang dipilih. Selain itu, program ini juga memungkinkan pengguna untuk melihat daftar reservasi yang telah dibuat, mengubah informasi reservasi jika diperlukan, serta menghapus reservasi yang tidak lagi dibutuhkan. Dengan adanya sistem ini, pencatatan data reservasi menjadi lebih terstruktur, mengurangi risiko kesalahan dalam manajemen reservasi, serta meningkatkan efisiensi dalam pelayanan spa.

2.2 Penjelasan Alur & Algoritma

2.2.1 Penjelasan Alur

1. Memulai Program dan Login Pengguna

- Saat program dijalankan, pengguna akan diminta untuk melakukan login dengan memasukkan nama dan NIM.
- Jika pengguna memasukkan nama dan NIM yang sesuai dengan data yang telah ditentukan dalam program (Intan Alfara Audia dan 008), maka login berhasil, dan program akan menampilkan pesan selamat datang.
- Jika pengguna salah memasukkan data, mereka memiliki hingga tiga kali percobaan login sebelum program berhenti secara otomatis.

2. Menampilkan Menu Utama

- Setelah login berhasil, program menampilkan menu utama yang berisi beberapa pilihan:
 1. Tambah Reservasi
 2. Lihat Reservasi
 3. Ubah Reservasi

4. Hapus Reservasi
5. Keluar
3. Tambah Reservasi
 - Jika pengguna memilih opsi Tambah Reservasi, mereka akan diminta untuk memasukkan:
 - Nama pelanggan
 - Tanggal reservasi (format DD/MM/YYYY)
 - Jenis layanan spa
 - Data ini akan disimpan dalam array reservasi dan jumlah reservasi akan bertambah.
 - Jika kapasitas reservasi telah mencapai batas maksimum ($MAX = 25$), maka program akan menampilkan pesan bahwa kapasitas reservasi penuh.
4. Melihat Daftar Reservasi
 - Jika pengguna memilih opsi Lihat Reservasi, program akan menampilkan semua data reservasi yang telah tersimpan dalam bentuk tabel.
 - Jika tidak ada reservasi, program akan menampilkan pesan bahwa belum ada data reservasi.
5. Mengubah Reservasi
 - Jika pengguna memilih opsi Ubah Reservasi, mereka harus memasukkan nomor reservasi yang ingin diubah.
 - Jika nomor yang dimasukkan valid, pengguna dapat memperbarui nama pelanggan, tanggal reservasi, dan layanan spa yang dipilih.
 - Jika nomor tidak valid, program akan menampilkan pesan kesalahan.
6. Menghapus Reservasi
 - Jika pengguna memilih opsi Hapus Reservasi, mereka harus memasukkan nomor reservasi yang ingin dihapus.
 - Jika nomor reservasi valid, data reservasi tersebut akan dihapus, dan data setelahnya akan digeser untuk menjaga keteraturan daftar.
 - Jika nomor reservasi tidak valid, program akan menampilkan pesan kesalahan.
7. Keluar dari Program
 - Jika pengguna memilih opsi Keluar, program akan menampilkan pesan penutupan dan selesai dijalankan.

2.2.2 Penjelasan Algoritma dan Blok Kode

1. Struktur Data

- Data reservasi disimpan dalam array dua dimensi `reservasi[MAX][Info]`, di mana:
- `reservasi[i][0]` menyimpan nama pelanggan
- `reservasi[i][1]` menyimpan tanggal reservasi
- `reservasi[i][2]` menyimpan layanan spa
- Variabel jumlah digunakan untuk menghitung jumlah reservasi yang telah dibuat.

2. Proses Login

- Menggunakan perulangan `while` untuk memberikan kesempatan login maksimal 3 kali.
- `getline(cin, namaPengguna)` digunakan agar input dapat menangkap nama lengkap yang mengandung spasi.
- Jika login berhasil, pengguna akan masuk ke menu utama. Jika gagal 3 kali, program berhenti (`return 0`).

3. Menu utama

- Perulangan `do-while` digunakan untuk memastikan menu terus ditampilkan hingga pengguna memilih keluar (`pilihan == 5`).
- `cin >> pilihan` digunakan untuk menangkap input pilihan pengguna.
- `cin.ignore()` digunakan untuk menghindari kesalahan input saat menggunakan `getline()`.
- `switch-case` digunakan untuk menangani setiap opsi menu.

4. Menambahkan Reservasi

- Program mengecek apakah kapasitas reservasi penuh.
- Jika masih tersedia, pengguna memasukkan:
 - Nama pelanggan
 - Tanggal reservasi
 - Layanan spa
- Data disimpan dan jumlah reservasi bertambah.

5. Melihat Daftar Reservasi

- Jika pengguna memilih opsi Lihat Reservasi, program akan menampilkan semua data reservasi menggunakan `setw()` yang telah tersimpan dalam bentuk tabel.

- Jika tidak ada reservasi, program akan menampilkan pesan bahwa belum ada data reservasi.
6. Mengubah Reservasi
 - Program meminta nomor reservasi yang ingin diubah.
 - Jika nomor valid, pengguna memasukkan data baru.
 - Data diperbarui di dalam array.
 7. Menghapus Reservasi
 - Program meminta nomor reservasi yang ingin dihapus.
 - Jika valid, data digeser ke atas untuk menghapus reservasi.
 - Jumlah reservasi berkurang.
 8. Keluar dari program
 - Jika pengguna memilih 5 (Keluar), program berhenti.
 - Program menampilkan pesan.

3. Source Code

A. Login Pengguna

Program meminta pengguna memasukkan Nama dan NIM. Jika sesuai dengan data yang telah ditentukan, pengguna bisa masuk. Jika salah, diberikan tiga kali kesempatan sebelum program berhenti.

Source Code:

```
int percobaan = 0;
string namaPengguna, nimPengguna;

while (percobaan < 3) {
    cout << " Masukkan Nama: ";
    getline(cin, namaPengguna);

    cout << " Masukkan NIM: ";
    getline(cin, nimPengguna);

    if (namaPengguna == "Intan Alfara Audia" && nimPengguna == "008") {
        cout << "Hore, Login berhasil! Selamat datang " << namaPengguna <<
        "!" << endl;
        break;
    } else {
        percobaan++;
        cout << "Gagal login. Percobaan ke-" << percobaan << " dari 3." <<
        endl;
    }

    if (percobaan == 3) {
        cout << "Anda telah login 3 kali. Program berhenti." << endl;
        return 0;
    }
}
```

A. Menu Utama

Menu utama berfungsi sebagai navigasi utama untuk mengelola reservasi spa.

Source Code :

```

int pilihan;
do {
    cout << "MANAJEMEN RESERVASI SPA" << endl;
    cout << "1. Tambah Reservasi" << endl;
    cout << "2. Lihat Reservasi" << endl;
    cout << "3. Ubah Reservasi" << endl;
    cout << "4. Hapus Reservasi" << endl;
    cout << "5. Keluar" << endl;
    cout << "Pilihan: ";
    cin >> pilihan;
    cin.ignore();
}

```

B. Tambah Reservasi

Fitur ini memungkinkan pengguna menambahkan reservasi baru jika kapasitas belum penuh.

Source Code :

```

if (jumlah < MAX) {
    cout << "Masukkan Nama Pelanggan: ";
    getline(cin, reservasi[jumlah][0]);
    cout << "Masukkan Tanggal Reservasi (DD/MM/YYYY): ";
    getline(cin, reservasi[jumlah][1]);
    cout << "Masukkan Layanan Spa: ";
    getline(cin, reservasi[jumlah][2]);
    jumlah++;
    cout << "Reservasi berhasil ditambahkan!" << endl;
} else {
    cout << "Kapasitas reservasi penuh!" << endl;
}

```

C. Menampilkan Reservasi

Menampilkan daftar reservasi dalam format tabel jika ada data yang tersimpan.

Source Code :

```

if (jumlah == 0) {
    cout << "Tidak ada reservasi saat ini." << endl;
} else {
    cout << "\nDaftar Reservasi:" << endl;
    for (int i = 0; i < jumlah; i++) {
        cout << "Reservasi " << i + 1 << ":" << endl;
        cout << "Nama Pelanggan: " << reservasi[i][0] << endl;
        cout << "Tanggal: " << reservasi[i][1] << endl;
        cout << "Layanan Spa: " << reservasi[i][2] << endl;
    }
}
}

```

D. Ubah Reservasi

Memungkinkan pengguna memperbarui data reservasi yang telah tersimpan berdasarkan nomor reservasi yang valid.

Source Code :

```

int ubah;
if (jumlah == 0) {
    cout << "Tidak ada reservasi untuk diubah." << endl;
} else {
    cout << "Masukkan nomor reservasi yang ingin diubah: ";
    cin >> ubah;
    cin.ignore();
    if (ubah > 0 && ubah <= jumlah) {
        cout << "Masukkan Nama Baru: ";
        getline(cin, reservasi[ubah - 1][0]);
        cout << "Masukkan Tanggal Baru (DD/MM/YYYY): ";
        getline(cin, reservasi[ubah - 1][1]);
        cout << "Masukkan Layanan Spa Baru: ";
        getline(cin, reservasi[ubah - 1][2]);
        cout << "Reservasi berhasil diperbarui!" << endl;
    } else {
        cout << "Nomor reservasi tidak valid." << endl;
    }
}
}

```

E. Hapus Reservasi

Menghapus reservasi berdasarkan nomor yang dipilih dan menggeser data ke atas.

Source Data :

```

int hapus;
if (jumlah == 0) {
    cout << "Tidak ada reservasi untuk dihapus." << endl;
} else {
    cout << "Masukkan nomor reservasi yang ingin dihapus: ";
    cin >> hapus;
    if (hapus > 0 && hapus <= jumlah) {
        for (int i = hapus - 1; i < jumlah - 1; i++) {
            reservasi[i][0] = reservasi[i + 1][0];
            reservasi[i][1] = reservasi[i + 1][1];
            reservasi[i][2] = reservasi[i + 1][2];
        }
        jumlah--;
        cout << "Reservasi berhasil dihapus." << endl;
    } else {
        cout << "Nomor reservasi tidak valid." << endl;
    }
}
}

```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

Berikut adalah skenario uji coba yang dilakukan pada program Manajemen Reservasi Spa:

1. Jika login gagal
 - Pengguna memasukkan nama atau NIM yang salah.
 - Program memberikan kesempatan hingga tiga kali percobaan.
 - Jika tetap gagal, program menampilkan pesan anda telah login 3 kali. Program berhenti.
2. Jika login berhasil
 - Pengguna memasukkan nama dan NIM yang benar.
 - Program menampilkan pesan sukses dan mengarahkan ke menu utama.
3. Tambah Reservasi
 - Pengguna memilih opsi Tambah Reservasi.
 - Memasukkan nama pelanggan, tanggal reservasi, dan layanan spa.
 - Program menampilkan konfirmasi sukses dan menyimpan data reservasi.
4. Lihat Reservasi
 - Jika tidak ada data, program menampilkan pesan bahwa reservasi kosong.

- Jika ada data, program menampilkan daftar reservasi yang telah tersimpan.
- Ubah Reservasi
 - Pengguna memilih opsi Ubah Reservasi dan memasukkan nomor reservasi yang ingin diubah.
 - Jika nomor valid, pengguna memasukkan data baru untuk reservasi tersebut.
 - Program memperbarui reservasi dan menampilkan pesan sukses.
 - Hapus Reservasi
 - Pengguna memilih opsi Hapus Reservasi dan memasukkan nomor reservasi yang ingin dihapus.
 - Jika nomor valid, program menghapus data dan menampilkan pesan sukses.
 - Jika tidak ada reservasi, program menampilkan pesan bahwa data kosong.
 - Jika keluar dari program
 - Pengguna memilih opsi Keluar dari menu utama.
 - Program menampilkan pesan dan berhenti.

4.2 Hasil Output

```
08-IntanAlfaraAudia-PT-2 } ; if ($?) { .\2409106008-IntanAlfaraAudia-PT-2 }
=====
|          SELAMAT DATANG DI          |
|        MANAJEMEN RESERVASI SPA        |
|          =====          |
|
|          LOGIN DULU GUYS          |
|          =====          |
Masukkan Nama: fara
Masukkan NIM: 1
=====
Gagal login. Nama atau NIM salah. Percobaan ke-1 dari 3.
=====
|          LOGIN DULU GUYS          |
|          =====          |
Masukkan Nama: fara
Masukkan NIM: 1
=====
Gagal login. Nama atau NIM salah. Percobaan ke-2 dari 3.
=====
|          LOGIN DULU GUYS          |
|          =====          |
Masukkan Nama: fara
Masukkan NIM: 1
=====
Gagal login. Nama atau NIM salah. Percobaan ke-3 dari 3.
Anda telah login 3 kali. Program berhenti.
PS D:\Praktikum-APL\post-test\post-test-2> |
```

Gambar 4.1 Login gagal

```

=====
|          SELAMAT DATANG DI          |
|        MANAJEMEN RESERVASI SPA        |
|=====|
|          LOGIN DULU GUYS          |
|=====|
Masukkan Nama: Intan Alfara Audia
Masukkan NIM: 008
=====
Hore, Login berhasil! Selamat datang Intan Alfara Audia!

=====
===      MANAJEMEN RESERVASI SPA      ===
=====
| 1. Tambah Reservasi                |
| 2. Lihat Reservasi                 |
| 3. Ubah Reservasi                  |
| 4. Hapus Reservasi                 |
| 5. Keluar                          |
|=====|
Pilihan: █

```

Gambar 4.2 Login berhasil

```

=====
Pilihan: 1
Masukkan Nama Pelanggan: fara
Masukkan Tanggal Reservasi (DD/MM/YYYY): 15/05/25
Masukkan Layanan Spa: Facial Treatment
Reservasi berhasil ditambahkan!
=====

```

Gambar 4.3 Tambah reservasi

```

Pilihan: 2

Daftar Reservasi:
+-----+-----+-----+-----+
| No | Nama Pelanggan | Tanggal | Layanan Spa |
+-----+-----+-----+-----+
| 1 | fara           | 15/05/25 | Facial Treatment |
+-----+-----+-----+-----+

```

Gambar 4.4 Lihat Reservasi

```

Pilihan: 3
Masukkan nomor reservasi yang ingin diubah: 1
Masukkan Nama Baru: Audia
Masukkan Tanggal Baru (DD/MM/YYYY): 25/02/25
Masukkan Layanan Spa Baru: Facial Treatment
Reservasi berhasil diperbarui!

```

Gambar 4.5 Ubah reservasi

```
Pilihan: 4
Masukkan nomor reservasi yang ingin dihapus: 1
Reservasi berhasil dihapus.
```

Gambar 4.6 Hapus reservasi

```
Pilihan: 5
Program selesai.
=====
| Hore, Program selesai. Terima kasih! |
=====
```

Gambar 4.7 Program keluar

5. Penggunaan Git

1) Git Add .

Menambahkan semua perubahan file (baru, diubah, atau dihapus) ke dalam staging area sebelum dikomit. Titik (.) berarti semua file dalam folder proyek akan ditambahkan ke staging area.

```
PS D:\Praktikum-APL> git add .
```

2) Git Commit

```
PS D:\Praktikum-APL> git commit -m "Finish Post Test 2"
[main 919b5bd] Finish Post Test 2
 3 files changed, 145 insertions(+)
 create mode 100644 post-test/post-test-2/2409106008-IntanAlfaraAudia-PT-2.cpp
 create mode 100644 post-test/post-test-2/2409106008-IntanAlfaraAudia-PT-2.exe
 create mode 100644 post-test/post-test-2/2409106008-IntanAlfaraAudia-PT-2.pdf
```

Menyimpan perubahan yang ada di staging area ke dalam repository dengan pesan deskriptif. Setiap commit mencatat perubahan sebagai versi yang bisa dikembalikan nanti.

3) Git Push

Mengirim (upload) commit dari branch lokal ke repository remote yang ada di server github.

```
PS D:\Praktikum-APL> git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.22 MiB | 484.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/IntanAlfaraAudia/Praktikum-APL.git
   4ad5feb..919b5bd  main -> main
branch 'main' set up to track 'origin/main'.
PS D:\Praktikum-APL> █
```