

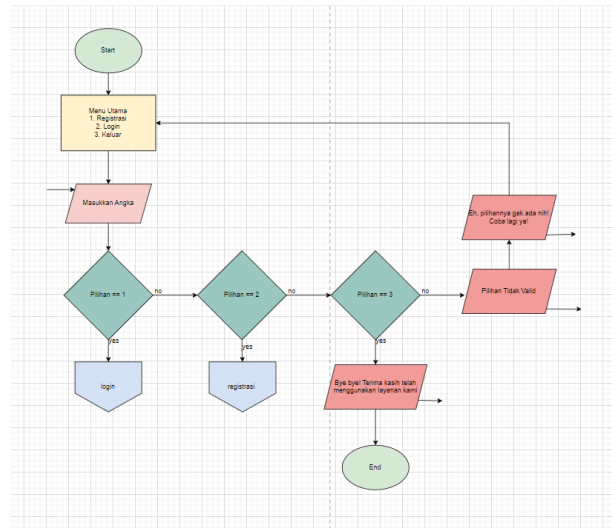
**LAPORAN PRAKTIKUM**  
**POSTTEST 6**  
**ALGORITMA PEMROGRAMAN LANJUT**



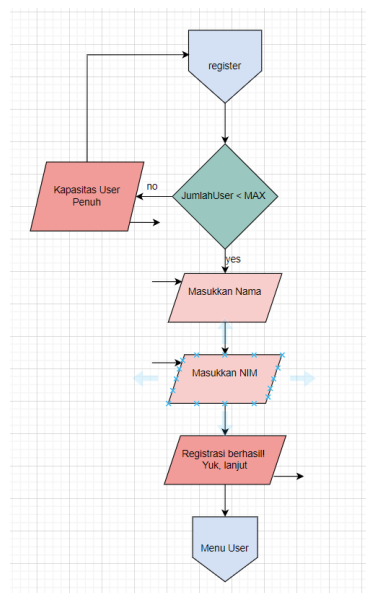
**Disusun oleh:**  
**Intan Alfara Audia 2409106008**  
**A1'24**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

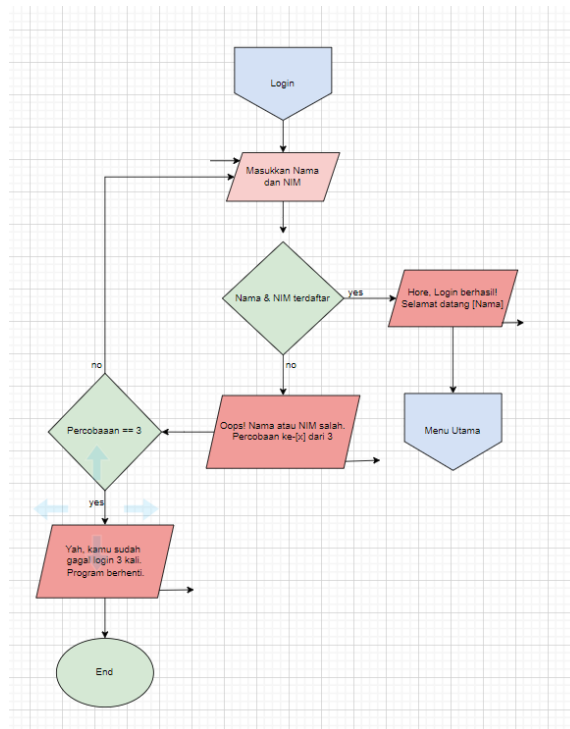
## 1. Flowchart



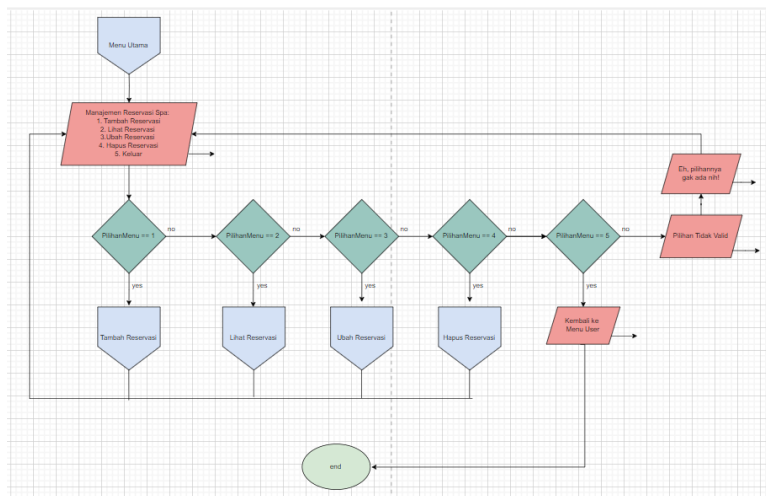
Gambar 1.1 Flowchart Menu user



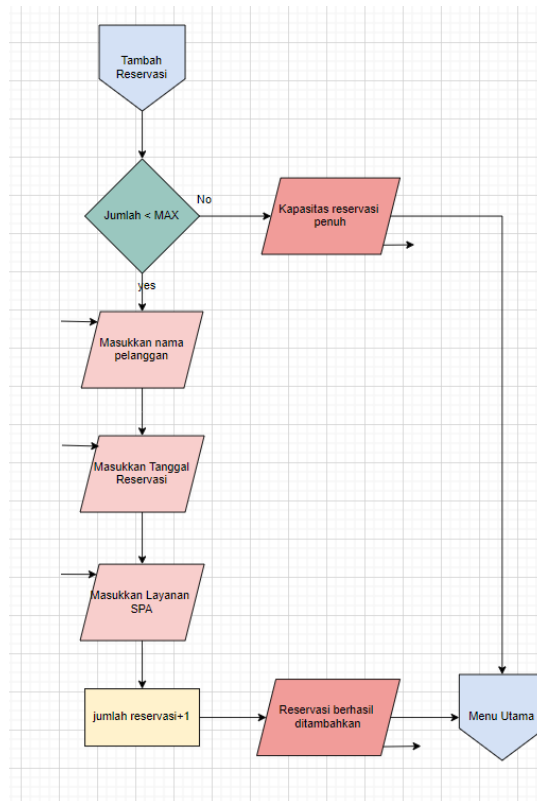
Gambar 1.2 Flowchart Register



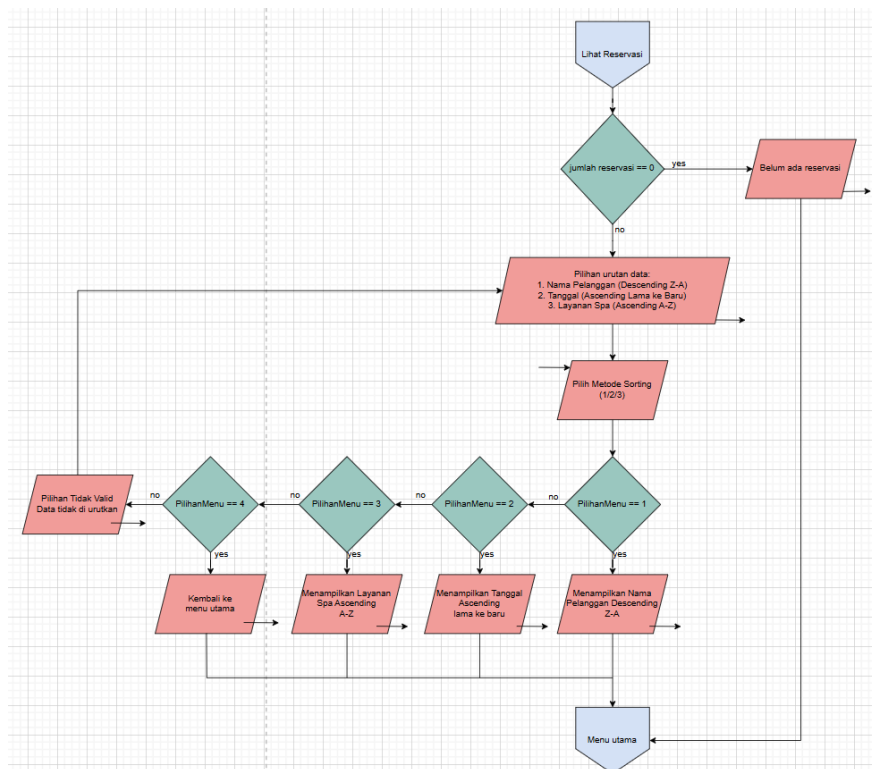
Gambar 1.3 Flowchart Login



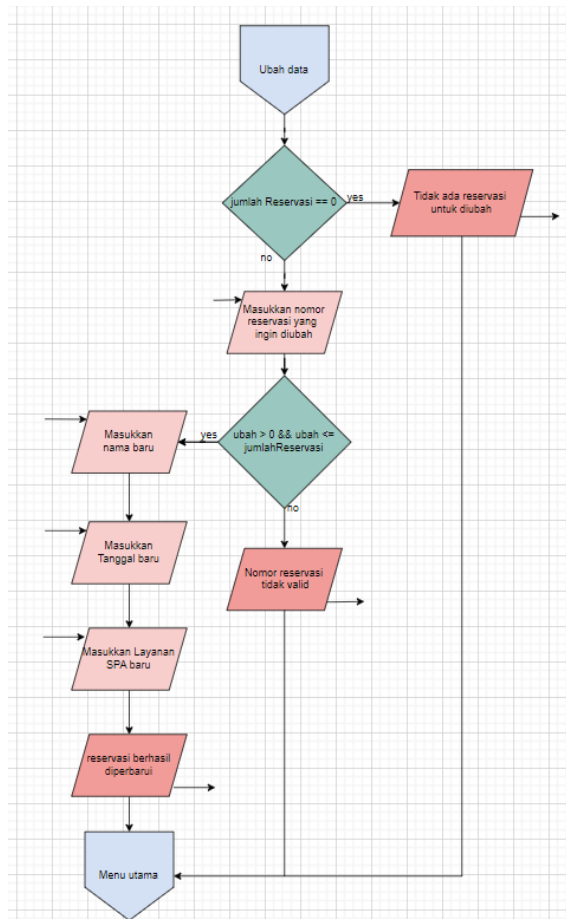
Gambar 1.4 Flowchart Menu Utama



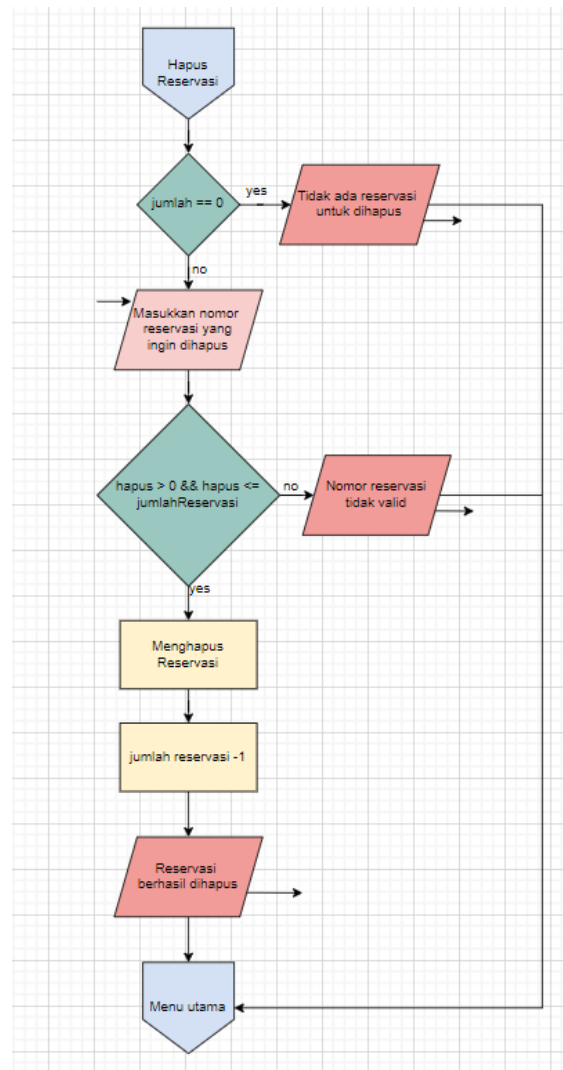
Gambar 1.5 Flochart Tambah reservasi



Gambar 1.6 Flowchart Lihat Reservasi



Gambar 1.7 Flowchart Ubah Reservasi



Gambar 1.8 Flowchart Hapus Reservasi

## 2. Analisis Program

Program ini dirancang untuk mempermudah proses pemesanan dan pengelolaan data reservasi spa oleh pengguna dan admin melalui antarmuka berbasis teks yang ramah pengguna. Program memiliki sistem login dan registrasi, di mana hanya pengguna terdaftar yang dapat mengakses fitur reservasi. Selain itu, terdapat akses khusus bagi admin untuk melihat seluruh data reservasi dari semua pengguna.

Struktur data program menggunakan array of struct untuk menyimpan informasi pengguna dan daftar reservasinya. Setiap pengguna memiliki data nama, NIM, status admin, dan array layanan yang mereka pesan. Program ini dirancang secara modular, dengan fungsi-fungsi terpisah untuk setiap operasi utama

Fitur CRUD (Create, Read, Update, Delete) dalam program memungkinkan pengguna untuk:

- Menambahkan reservasi layanan spa (nama pelanggan, tanggal, jenis layanan).

- Menampilkan data reservasi dalam bentuk tabel yang rapi.
- Mengubah data reservasi apabila terdapat perubahan layanan atau jadwal.
- Menghapus reservasi yang sudah tidak dibutuhkan.

Program ini juga mengimplementasikan:

- Fungsi rekursif untuk menghitung total pelanggan atau jumlah reservasi yang telah dibuat oleh pengguna.
- Fungsi overloading untuk mencari data reservasi, baik berdasarkan nama pelanggan saja, maupun kombinasi nama dan tanggal reservasi.
- Pointer digunakan untuk mempermudah manipulasi data antar fungsi, khususnya dalam pengelolaan data user dan layanan.

Dengan fitur-fitur tersebut, program ini memberikan kemudahan bagi pengguna untuk melakukan reservasi secara mandiri, serta bagi admin untuk memantau seluruh aktivitas reservasi.

Dalam program ini, dilakukan pengurutan (sorting) data berdasarkan ketentuan:

- Sorting Huruf (Nama Pelanggan):  
Menggunakan Bubble Sort secara Descending (Z - A).
- Sorting Angka (Tanggal Reservasi):  
Menggunakan Merge Sort secara Ascending (Tanggal lebih awal - lebih baru).
- Sorting Tambahan (Jenis Layanan Spa):  
Menggunakan Selection Sort secara Ascending (A - Z).

Dengan penerapan minimal 3 metode sorting yang berbeda (Bubble, Merge, Selection) sesuai instruksi.

### 3. Source Code

#### 3.1 Fitur Register

Fitur ini digunakan untuk menambahkan pengguna baru ke dalam sistem dengan memasukkan nama dan NIM. Jika kapasitas pengguna penuh, maka registrasi tidak dapat dilakukan.

**Source Code:**

```
void registrasi() {
    if (jumlahUser >= MAX) {
        cout << "Kapasitas user penuh!" << endl;
        return;
    }
    cin.ignore();
    cout << "Masukkan Nama: ";
    getline(cin, users[jumlahUser].namaPengguna);
    cout << "Masukkan NIM: ";
    getline(cin, users[jumlahUser].nimPengguna);
    users[jumlahUser].isAdmin = 0;
    users[jumlahUser].jumlahReservasi = 0;
    jumlahUser++;
    cout << "Registrasi berhasil! Yuk lanjut login" << endl;
}
```

#### 3.2 Fitur Login

Fitur ini memungkinkan pengguna untuk masuk ke dalam sistem dengan nama dan NIM yang sesuai. Jika pengguna salah memasukkan data sebanyak 3 kali, maka program akan berhenti.

**Source Code:**

```
int login(string nama, string nim) {
    for (int i = 0; i < jumlahUser; i++) {
        if (users[i].namaPengguna == nama && users[i].nimPengguna == nim) {
            return i;
        }
    }
    return -1;
}
```

#### 3.3 Fitur Menambahkan Reservasi

Fitur ini digunakan untuk menambahkan data reservasi pelanggan yang berisi nama pelanggan, tanggal, dan layanan spa yang dipilih.



### Source Code:

```
void tambahReservasi(User* user) {
    if (user->jumlahReservasi >= 25) {
        cout << "Kapasitas reservasi penuh!" << endl;
        return;
    }
    cin.ignore();
    int idx = user->jumlahReservasi;
    cout << "Masukkan Nama Pelanggan: ";
    getline(cin, user->daftarReservasi[idx].namaPelanggan);
    cout << "Masukkan Tanggal (DD/MM/YYYY): ";
    getline(cin, user->daftarReservasi[idx].tanggal);
    cout << "Masukkan Layanan Spa: ";
    getline(cin, user->daftarReservasi[idx].layanan);
    user->jumlahReservasi++;
    cout << "Sip! Reservasimu sudah masuk!" << endl;
}
```

### 3.4 Fitur Melihat Reservasi

Fitur ini digunakan untuk menampilkan daftar reservasi yang telah dibuat. Jika belum ada reservasi, maka akan muncul pesan bahwa belum ada reservasi.

### Source Code:

```
void tampilData(User* user) {
    if (user->jumlahReservasi == 0) {
        cout << "Belum ada reservasi." << endl;
        return;
    }

    int pilihanSort;
    cout << "\nPilihan urutan data:\n";
    cout << "1. Urutkan Nama Pelanggan Descending (Bubble Sort)\n";
    cout << "2. Urutkan Tanggal Ascending (Merge Sort)\n";
    cout << "3. Urutkan Layanan Spa Ascending (Selection Sort)\n";
    cout << "Pilih metode sorting (1/2/3): ";
    cin >> pilihanSort;

    switch (pilihanSort) {
        case 1: bubbleSortNamaDescending(user); break;
        case 2: mergeSortTanggal(user->daftarReservasi, 0, user->jumlahReservasi - 1); break;
        case 3: selectionSortLayananAscending(user); break;
        default: cout << "Pilihan tidak valid! Data tidak diurutkan.\n"; break;
    }
}
```

### 3.5 Fitur Mengubah Reservasi

Fitur ini memungkinkan pengguna untuk mengedit informasi reservasi yang sudah ada, seperti mengubah nama pelanggan, tanggal, atau jenis layanan.

**Source Code :**

```
void ubahReservasi(User* user) {
    tampilData(user);
    if (user->jumlahReservasi == 0) {
        cout << "Tidak ada reservasi untuk diubah." << endl;
        return;
    }

    int ubah;
    cout << "Masukkan nomor reservasi yang ingin diubah: ";
    cin >> ubah;
    cin.ignore();

    if (ubah > 0 && ubah <= user->jumlahReservasi) {
        cout << "Masukkan Nama Baru: ";
        getline(cin, user->daftarReservasi[ubah - 1].namaPelanggan);
        cout << "Masukkan Tanggal Baru (DD/MM/YYYY): ";
        getline(cin, user->daftarReservasi[ubah - 1].tanggal);
        cout << "Masukkan Layanan Spa Baru: ";
        getline(cin, user->daftarReservasi[ubah - 1].layanan);
        cout << "Reservasi berhasil diperbarui!" << endl;
    } else {
        cout << "Nomor reservasi tidak valid." << endl;
    }
}
```

### 3.6 Fitur Menghapus Reservasi

Fitur ini memungkinkan pengguna untuk menghapus reservasi tertentu berdasarkan nomor urut reservasi.

**Source Code:**

```

void hapusReservasi(User* user) {
    tampilData(user);
    if (user->jumlahReservasi == 0) {
        cout << "Tidak ada reservasi untuk dihapus." << endl;
        return;
    }

    int hapus;
    cout << "Masukkan nomor reservasi yang ingin dihapus: ";
    cin >> hapus;

    if (hapus > 0 && hapus <= user->jumlahReservasi) {
        for (int i = hapus - 1; i < user->jumlahReservasi - 1; i++) {
            user->daftarReservasi[i] = user->daftarReservasi[i + 1];
        }
        user->jumlahReservasi--;
        cout << "Reservasi berhasil dihapus." << endl;
    } else {
        cout << "Nomor reservasi tidak valid." << endl;
    }
}

```

### 3.7 Fitur Hitung Total Reservasi (menggunakan parameter address-of dan rekursi)

Fitur ini Menghitung jumlah reservasi yang dimiliki user secara rekursif menggunakan pointer.

**Source Code:**

```

int hitungTotalPelanggan(User* user, int indeks) {
    if (indeks == user->jumlahReservasi) return 0;
    return 1 + hitungTotalPelanggan(user, indeks + 1);
}

```

### 3.8 Fitur Sorting (Bubble Sort)

Fitur ini Mengurutkan data dengan membandingkan dan menukar elemen yang berdekatan hingga data terurut disini saya menggunakan descending.

**Source Code:**

```

void bubbleSortNamaDescending(User* user) {
    for (int i = 0; i < user->jumlahReservasi - 1; i++) {
        for (int j = 0; j < user->jumlahReservasi - i - 1; j++) {
            if (user->daftarReservasi[j].namaPelanggan < user->daftarReservasi[j + 1].namaPelanggan) {
                swap(user->daftarReservasi[j], user->daftarReservasi[j + 1]);
            }
        }
    }
}

```

### 3.9 Fitur Sorting (Merge Sort)

Fitur ini Mengurutkan data dengan membagi, mengurutkan, dan menggabungkan kembali data disini saya menggunakan ascending.

**Source Code:**

```
void merge(ReservasiSpa arr[], int l, int m, int r) {
    int n1 = m - l + 1;
    int n2 = r - m;
    ReservasiSpa 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].tanggal <= R[j].tanggal) {
            arr[k++] = L[i++];
        } else {
            arr[k++] = R[j++];
        }
    }
    while (i < n1) arr[k++] = L[i++];
    while (j < n2) arr[k++] = R[j++];
}

void mergeSortTanggal(ReservasiSpa arr[], int l, int r) {
    if (l < r) {
        int m = l + (r - l) / 2;
        mergeSortTanggal(arr, l, m);
        mergeSortTanggal(arr, m + 1, r);
        merge(arr, l, m, r);
    }
}
```

### 3.10 Fitur Sorting (Selection Sort)

Mengurutkan data dengan memilih elemen terkecil/terbesar dan menukarnya dengan elemen yang belum terurut.

## Source Code:

```
void selectionSortNamaAscending(User* user) {
    for (int i = 0; i < user->jumlahReservasi - 1; i++) {
        int min_idx = i;
        for (int j = i + 1; j < user->jumlahReservasi; j++) {
            if (user->daftarReservasi[j].namaPelanggan < user->daftarReservasi[min_idx].namaPelanggan) {
                min_idx = j;
            }
        }
        swap(user->daftarReservasi[min_idx], user->daftarReservasi[i]);
    }
}

void selectionSortLayananAscending(User* user) {
    for (int i = 0; i < user->jumlahReservasi - 1; i++) {
        int min_idx = i;
        for (int j = i + 1; j < user->jumlahReservasi; j++) {
            if (user->daftarReservasi[j].layanan < user->daftarReservasi[min_idx].layanan) {
                min_idx = j;
            }
        }
        swap(user->daftarReservasi[min_idx], user->daftarReservasi[i]);
    }
}
```

## 4. Uji Coba dan Hasil Output

Berikut adalah skenario uji coba lengkap dari awal program berdasarkan kode yang telah dibuat.

### 4.1 Memulai Program

- Program menampilkan menu utama yang memungkinkan pengguna untuk registrasi, login, dan keluar.

```
=====
===  SELAMAT DATANG DI RESERVASI SPA  ===
=====

=====
===          MENU USER          ===
=====
| 1. Registrasi |
| 2. Login      |
| 3. Keluar     |
=====
Pilihan:
```

Gambar 4.1 Menu user

## 4.2 Registrasi

- Pengguna memilih opsi 1
- Memasukkan Nama dan NIM.
- Jika berhasil, program akan menampilkan pesan Registrasi berhasil! Yuk ,lanjut.

```
Pilihan: 1
=====
==          REGISTRASI          ==
=====

Masukkan Nama: Fara
Masukkan NIM: 009
Registrasi berhasil! Yuk, lanjut
=====
===          MENU USER          ===
=====
| 1. Registrasi |
| 2. Login      |
| 3. Keluar     |
=====
Pilihan: █
```

*Gambar 4.2 registrasi pengguna baru*

## 4.3 Login Pengguna

- Pengguna memilih opsi 2 (Login).
- Jika nama dan NIM benar, login berhasil dan masuk ke menu reservasi.
- Jika salah, diberikan kesempatan hingga 3 kali.
- Jika gagal 3 kali, program menampilkan pesan "Program berhenti".
- Sistem juga menampilkan jumlah pelanggan (reservasi) yang telah dibuat oleh pengguna tersebut.

```

Pilihan: 2
Masukkan Nama: Fara
Masukkan NIM: 009
Hore, login berhasil! Selamat datang Fara!
Total pelanggan (reservasi): 0

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

```

Gambar 4.3.1 Login pengguna baru

```

Pilihan: 2
=====
==          LOGIN          ==
=====
Masukkan Nama: ara
Masukkan NIM: 08
Oops! Nama atau NIM salah. Percobaan ke-1 dari 3.
Masukkan Nama: ara
Masukkan NIM: 08
Oops! Nama atau NIM salah. Percobaan ke-2 dari 3.
Masukkan Nama: ara
Masukkan NIM: 08
Oops! Nama atau NIM salah. Percobaan ke-3 dari 3.
Yah, kamu sudah gagal login 3 kali. Program berhenti.
PS D:\Praktikum-APL\post-test-3> █

```

Gambar 4.3.2 Login gagal

```
Pilihan: 2
Masukkan Nama: Intan Alfara
Masukkan NIM: 008
Hore, login berhasil! Selamat datang Intan Alfara!
Total pelanggan (reservasi): 0

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

Gambar 4.3.3 Login user

#### 4.4 Menu Manajemen Reservasi SPA

- Setelah login berhasil, program menampilkan menu reservasi.

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

Gambar 4.4 Menu Manajemen Reservasi SPA

#### 4.5 Tambah Reservasi

- Pengguna memilih opsi 1 (Tambah Reservasi).
- Memasukkan nama pelanggan, tanggal reservasi, dan layanan spa.
- Jika berhasil, program menampilkan pesan Sip! Reservasimu sudah masuk!.



```

Pilihan: 1
Masukkan Nama Pelanggan: Audia
Masukkan Tanggal (DD/MM/YYYY): 24/03/2025
Masukkan Layanan Spa: Facial
Sip! Reservasimu sudah masuk!

=====
===      MANAJEMEN RESERVASI SPA      ===
=====

| 1. Tambah Reservasi |
| 2. Lihat Reservasi  |
| 3. Ubah Reservasi   |
| 4. Hapus Reservasi  |
| 5. Keluar           |
=====

```

Gambar 4.5 Menambah reservasi

#### 4.6 Lihat Reservasi

- Pengguna memilih opsi 2 (Lihat Reservasi).
- Jika ada data, program menampilkan pilihan urutan data daftar reservasi.
  1. Jika pengguna memilih opsi 1 yaitu Mengurutkan nama pelanggan secara descending Z-A menggunakan metode sorting bubble sort.
  2. Jika pengguna memilih opsi 2 yaitu Mengurutkan tanggal reservasi secara ascending menggunakan metode sorting merge sort sesuai yang terlebih dahulu yang melakukan reservasi.
  3. Jika pengguna memilih opsi 3 yaitu Mengurutkan layanan spa secara ascending A-Z menggunakan metode sorting selection sort.
- Jika tidak ada data, program menampilkan pesan "Belum ada reservasi.".

```

Pilihan urutan data:
1. Urutkan Nama Pelanggan Descending (Bubble Sort)
2. Urutkan Tanggal Ascending (Merge Sort)
3. Urutkan Layanan Spa Ascending (Selection Sort)
Pilih metode sorting (1/2/3): 1
+---+-----+-----+-----+
| No | Nama Pelanggan | Tanggal | Layanan Spa |
+---+-----+-----+-----+
| 1  | Fara           | 26/04/2025 | Pijat       |
| 2  | Audia          | 24/03/2025 | Facial      |
| 3  | Alfara         | 25/04/2025 | Facial      |
+---+-----+-----+-----+

```

Gambar 4.6.1 Metode Sorting (descending)

```

Pilihan urutan data:
1. Urutkan Nama Pelanggan Descending (Bubble Sort)
2. Urutkan Tanggal Ascending (Merge Sort)
3. Urutkan Layanan Spa Ascending (Selection Sort)
Pilih metode sorting (1/2/3): 2
+---+-----+-----+-----+
| No | Nama Pelanggan | Tanggal | Layanan Spa |
+---+-----+-----+-----+
| 1 | Audia          | 24/03/2025 | Facial      |
| 2 | Alfara         | 25/04/2025 | Facial      |
| 3 | Fara           | 26/04/2025 | Pijat       |
+---+-----+-----+-----+

```

Gambar 4.6.2 Metode Sorting (Ascending)

```

Pilihan urutan data:
1. Urutkan Nama Pelanggan Descending (Bubble Sort)
2. Urutkan Tanggal Ascending (Merge Sort)
3. Urutkan Layanan Spa Ascending (Selection Sort)
Pilih metode sorting (1/2/3): 3
+---+-----+-----+-----+
| No | Nama Pelanggan | Tanggal | Layanan Spa |
+---+-----+-----+-----+
| 1 | Audia          | 24/03/2025 | Facial      |
| 2 | Alfara         | 25/04/2025 | Facial      |
| 3 | Fara           | 26/04/2025 | Pijat       |
+---+-----+-----+-----+

```

Gambar 4.6.3 Metode Sorting (Ascending)

```

=====
Pilihan: 2
Belum ada reservasi.
=====

```

Gambar 4.6.4 Tidak Ada Daftar Reservasi

## 4.7 Ubah Reservasi

- Pengguna memilih opsi 3 (Ubah Reservasi).
- Jika ada reservasi, pengguna bisa memasukkan data baru.
- Jika tidak ada reservasi, program akan menampilkan pesan error.

```
=====
Pilihan: 3
Masukkan nomor reservasi yang ingin diubah: 1
Masukkan Nama Baru: Fara
Masukkan Tanggal Baru (DD/MM/YYYY): 24/03/25
Masukkan Layanan Spa Baru: Lulur
Reservasi berhasil diperbarui!
=====
```

Gambar 4.7.1 Ubah Data Reservasi

```
=====
Pilihan: 3
Tidak ada reservasi untuk diubah.
=====
===          MANAJEMEN RESERVASI SPA          ===
=====
```

Gambar 4.7.2 Tidak Ada Data Reservasi

## 4.8 Hapus Reservasi

- Pengguna memilih opsi 4 (Hapus Reservasi).
- Jika ada data, pengguna memasukkan nomor reservasi yang ingin dihapus.
- Jika tidak ada data, program menampilkan pesan error.

```

=====
Pilihan: 4
Masukkan nomor reservasi yang ingin dihapus: 1
Reservasi berhasil dihapus.
=====
===          MANAJEMEN RESERVASI SPA          ===
=====

```

*Gambar 4.8.1 Hapus Data Reservasi*

```

=====
Pilihan: 4
Tidak ada reservasi untuk dihapus.
=====

```

*Gambar 4.8.2 Tidak Ada Data Reservasi*

#### 4.9 Keluar dari Program

- Pengguna memilih opsi 5 (Keluar).
- Program kembali ke menu user atau keluar sepenuhnya.

```

Pilihan: 5

=====
===          MENU USER          ===
=====
| 1. Registrasi |
| 2. Login      |
| 3. Keluar     |
=====
Pilihan: 3
PS D:\Praktikum-APL\post-test\post-test-6>

```

*Gambar 4.9 Keluar*

## 5. Langkah-Langkah Git pada VSCode

### 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.

### 2) Git Commit

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 add .
PS D:\Praktikum-APL> git commit -m "Finish Post Test 6"
[main c59e19e] Finish Post Test 6
 2 files changed, 315 insertions(+)
   create mode 100644 post-test/post-test-6/2409106008_IntanAlfaraAudia_PT-6.cpp
   create mode 100644 post-test/post-test-6/2409106008_IntanAlfaraAudia_PT-6.exe
PS D:\Praktikum-APL> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 681.75 KiB | 5.68 MiB/s, done.
Total 6 (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/IntanAlfaraAudia/Praktikum-APL.git
   96d4974..c59e19e  main -> main
branch 'main' set up to track 'origin/main'.
PS D:\Praktikum-APL> █
```