LAPORAN PRAKTIKUM

**POSTTEST 4**

**ALGORITMA PEMROGRAMAN LANJUT**

****

**Disusun oleh:**

**Intan Alfara Audia 2409106008**

**A1’24**

**PROGRAM STUDI INFORMATIKA**

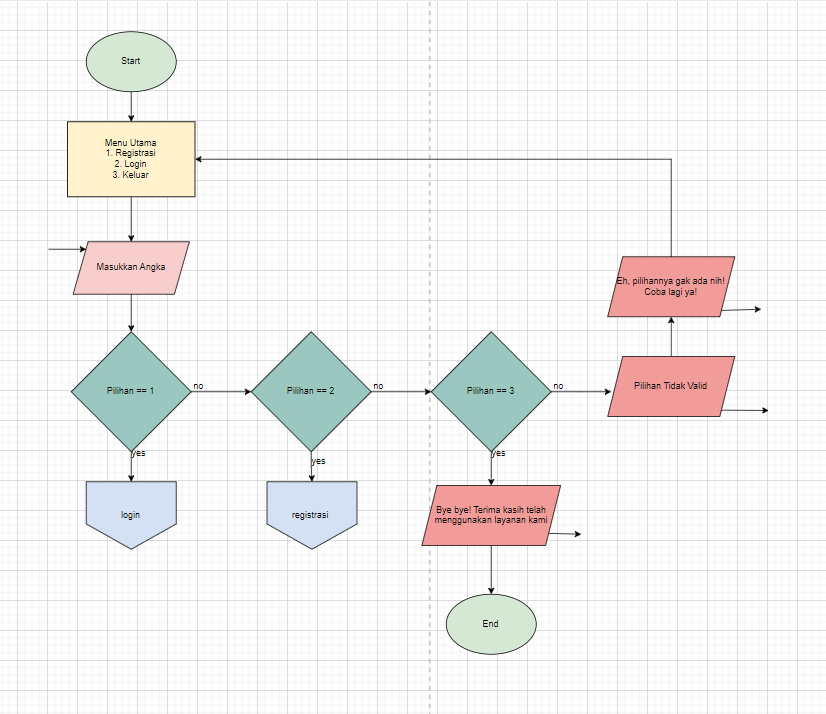
**UNIVERSITAS MULAWARMAN**

**SAMARINDA**

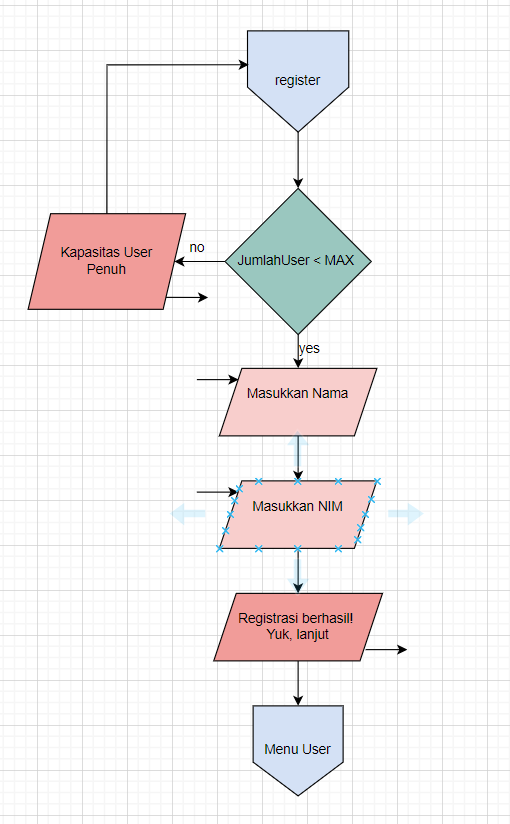
**2025**

# 

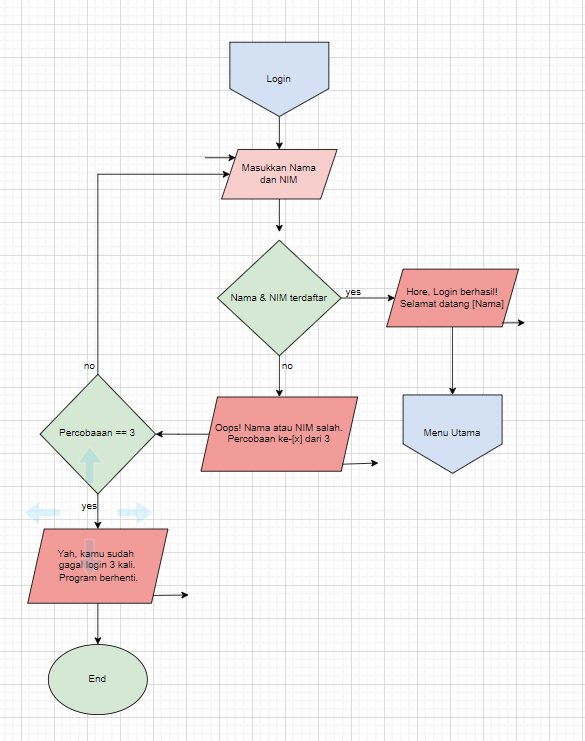
# 1. Flowchart



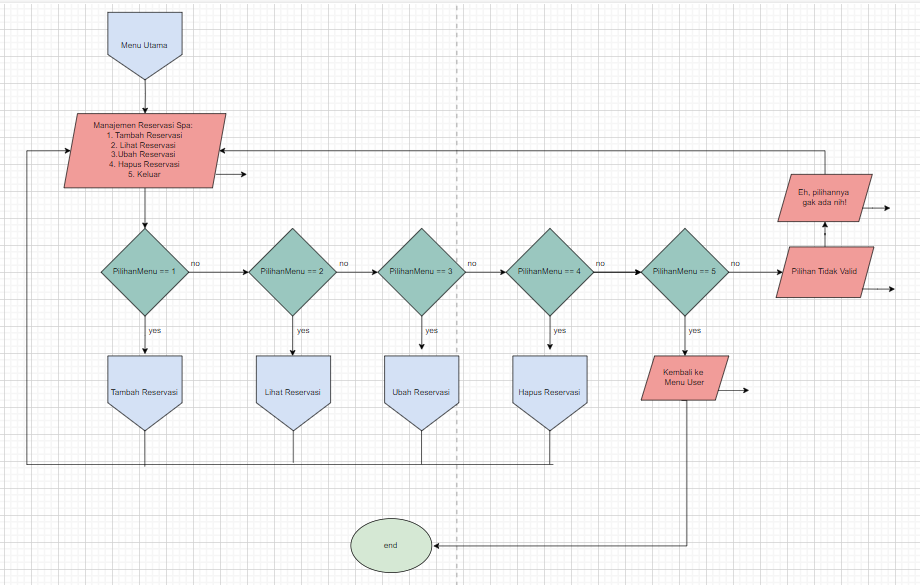
Gambar 1.1 Flowchart Menu user



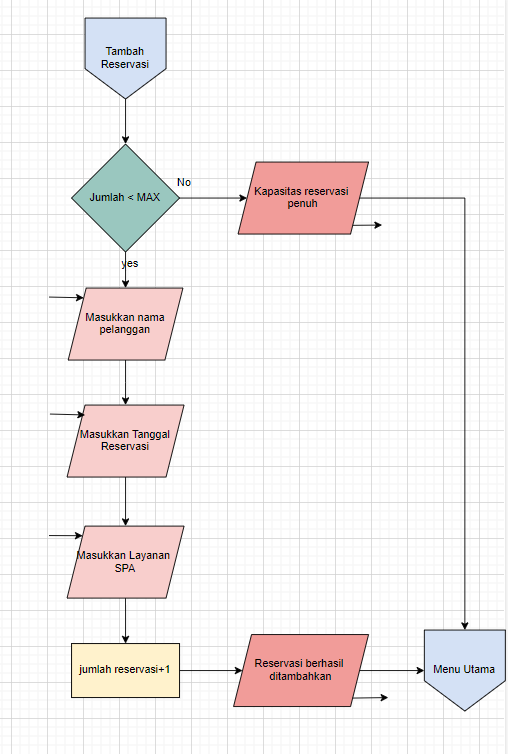
Gambar 1.2 Flowchart Register



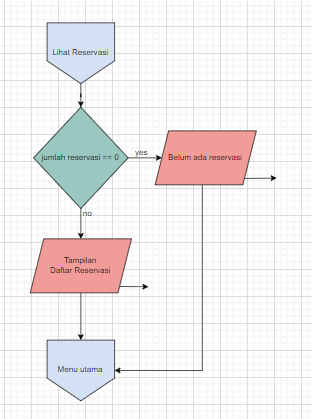
Gambar 1.3 Flowchart Login



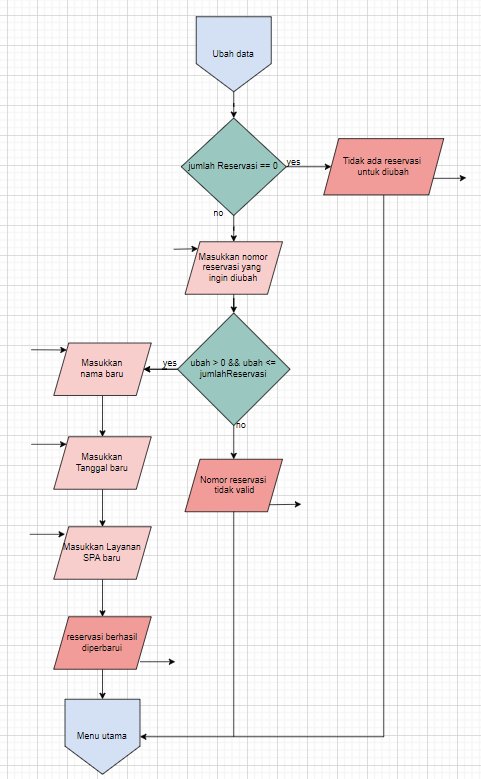
Gambar 1.4 Flowchart Menu Utama



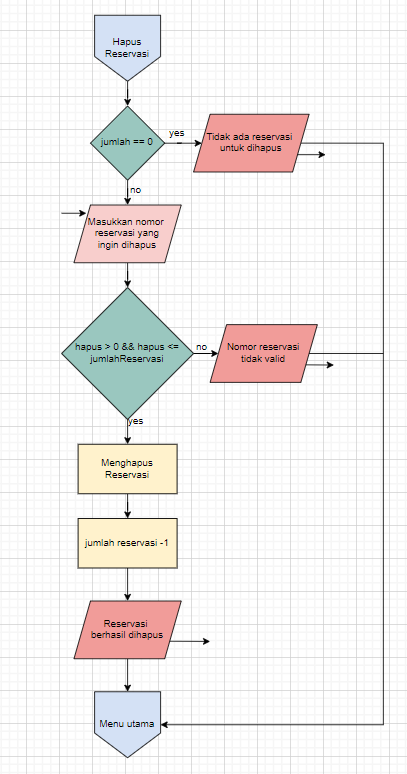
Gambar 1.5 Flochart Tambah reservasi



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 pengelolaan reservasi layanan spa melalui antarmuka berbasis teks yang sederhana dan efisien. Program berjalan secara berulang selama pengguna belum memilih opsi keluar, memberikan keleluasaan kepada pengguna dalam mengelola data reservasi. Melalui sistem login, hanya pengguna terdaftar yang dapat mengakses fitur reservasi, dan jika data login salah dimasukkan sebanyak tiga kali, program akan otomatis berhenti.

Dalam program ini, data pengguna dan reservasi dikelola menggunakan array of struct, yang menggabungkan informasi nama pengguna, NIM, serta daftar reservasinya ke dalam satu kesatuan data yang lebih terstruktur. Program ini menerapkan pemrograman modular melalui pemisahan fitur ke dalam fungsi dan prosedur dengan parameter.

Pengguna dapat melakukan berbagai operasi CRUD (Create, Read, Update, Delete) seperti:

* Menambahkan reservasi, yaitu mencatat nama pelanggan, tanggal reservasi, dan layanan spa yang diinginkan.
* Melihat daftar reservasi dalam bentuk tabel yang rapi dan mudah dibaca.
* Mengubah data reservasi, misalnya jika ada perubahan jadwal atau layanan.
* Menghapus reservasi yang tidak diperlukan lagi.

Program ini juga menerapkan:

* Fungsi rekursif untuk menghitung total pelanggan atau reservasi yang dimiliki oleh seorang user.
* Fungsi overloading untuk mencari reservasi berdasarkan nama saja, maupun berdasarkan nama dan tanggal sekaligus.

Dengan adanya sistem ini, pengelolaan reservasi spa menjadi lebih efisien, terstruktur, dan mudah digunakan oleh pengguna. Program ini juga dapat dikembangkan lebih lanjut untuk keperluan pelayanan spa yang lebih profesional dan modern.

# 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;     }      cout << "Daftar Reservasi Spa:" << endl;     for (int i = 0; i < user.jumlahReservasi; i++) {         cout << "Reservasi ke-" << i + 1 << ":" << endl;         cout << "Nama Pelanggan : " << user.daftarReservasi[i].namaPelanggan << endl;         cout << "Tanggal        : " << user.daftarReservasi[i].tanggal << endl;         cout << "Layanan Spa    : " << user.daftarReservasi[i].layanan << endl;     } } |

### **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:**

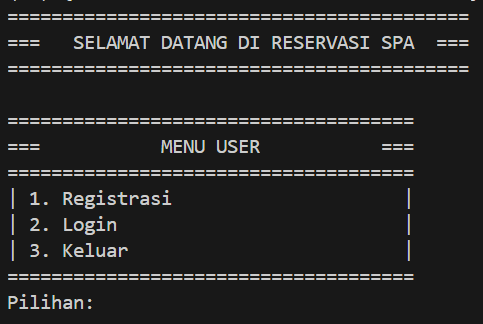
|  |
| --- |
| void hapusReservasi(User& user) {     tampilData(user);     if (user.jumlahReservasi == 0) {         cout << "Tidak ada reservasi untuk dihapus." << endl;         return;     }      int hapus;     cout << "Masukkan nomor reservasi yang ingin dihapus: ";     cin >> hapus;      if (hapus > 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;     } } |

# 4. Uji Coba dan Hasil Output

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

## 4.1 Memulai Program

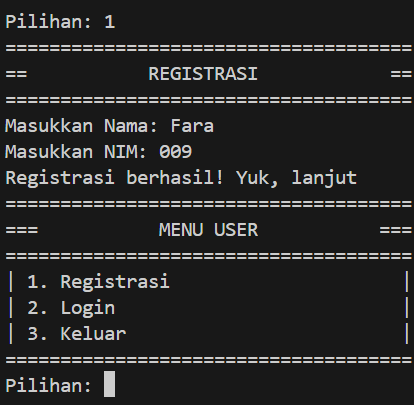
* Program menampilkan menu utama yang memungkinkan pengguna untuk registrasi, login, dan keluar.



Gambar 4.1 Menu user

## 4.2 Registrasi

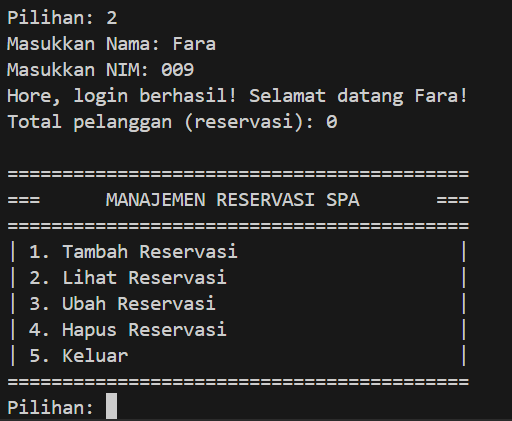
* Pengguna memilih opsi 1 (Register).
* Memasukkan Nama dan NIM.
* Jika berhasil, program akan menampilkan pesan Registrasi behasi! Yuk ,lanjut.



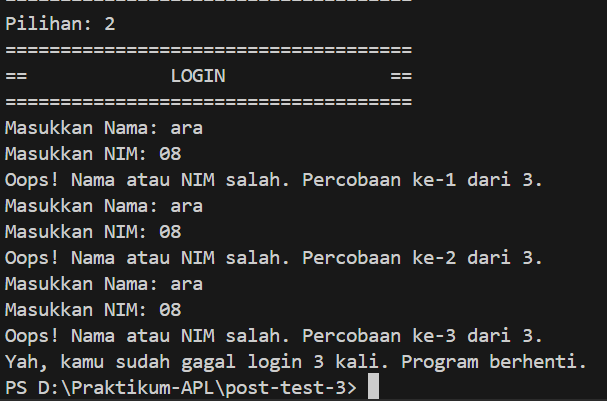
Gambar 4.2 registrasi pengguna baru

## 4.3 Login Pengguna

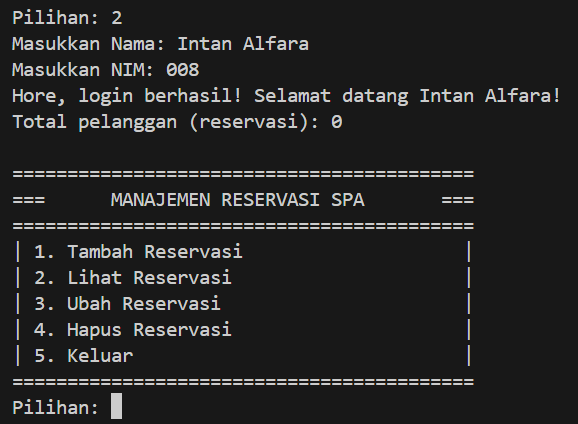
* Pengguna memilih opsi 2 (Login).
* Jika nama dan NIM benar, login berhasil dan masuk ke menu reservasi.
* Jika salah, diberikan kesempatan hingga 3 kali.
* Jika gagal 3 kali, program menampilkan pesan "Program berhenti".
* Sistem juga menampilkan jumlah pelanggan (reservasi) yang telah dibuat oleh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



Gambar 4.3.1 Login pengguna baru



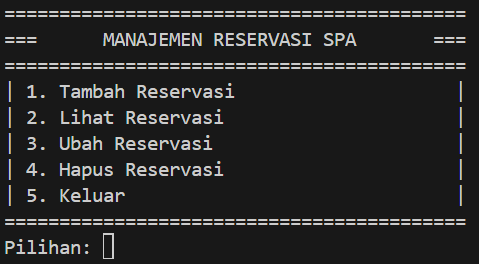
Gambar 4.3.2 Login gagal



Gambar 4.3.3 Login user

## 4.4 Menu Manajemen Reservasi SPA

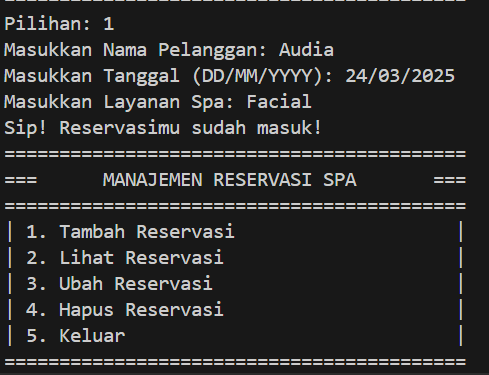
* Setelah login berhasil, program menampilkan menu reservasi.



Gambar 4.4 Menu Manajemen Reservasi SPA

## 4.5 Tambah Reservasi

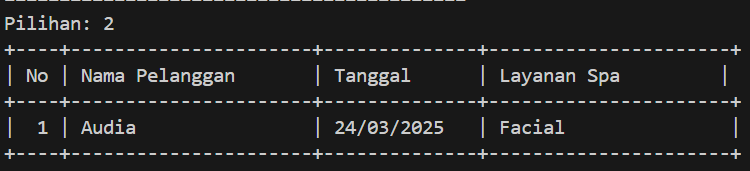
* Pengguna memilih opsi 1 (Tambah Reservasi).
* Memasukkan nama pelanggan, tanggal reservasi, dan layanan spa.
* Jika berhasil, program menampilkan pesan Sip! Reservasimu sudah masuk!.



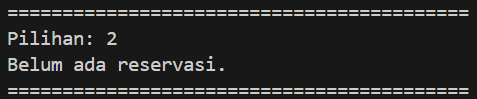
Gambar 4.5 Menambah reservasi

## 4.6 Lihat Reservasi

* Pengguna memilih opsi 2 (Lihat Reservasi).
* Jika ada data, program menampilkan daftar reservasi.
* Jika tidak ada data, program menampilkan pesan "Belum ada reservasi.".



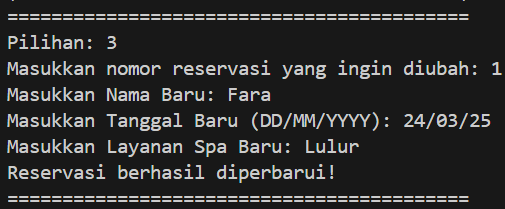
Gambar 4.6.1 Daftar Reservasi



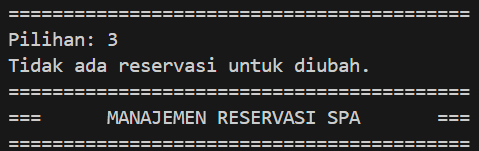
Gambar 4.6.2 Tidak Ada Daftar Reservasi

## 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.



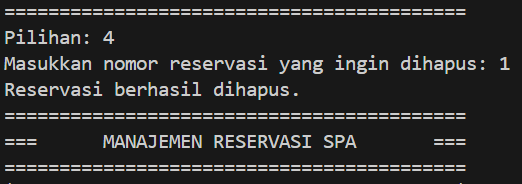
Gambar 4.7.1 Ubah Data Reservasi



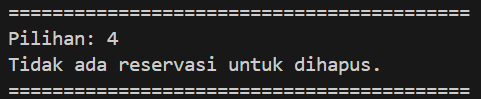
Gambar 4.7.2 Tidak Ada Data Reservasi

## 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.



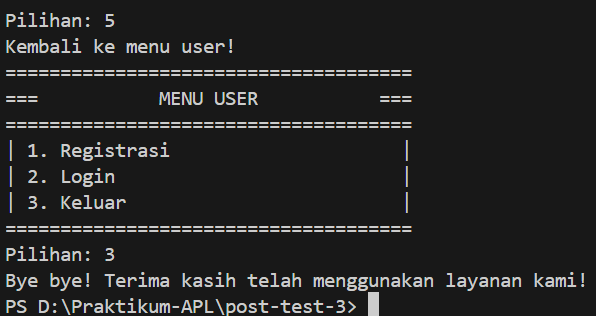
Gambar 4.8.1 Hapus Data Reservasi



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.



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.

