

**LAPORAN PRAKTIKUM**  
**POSTTEST 4**  
**ALGORITMA PEMROGRAMAN LANJUT**

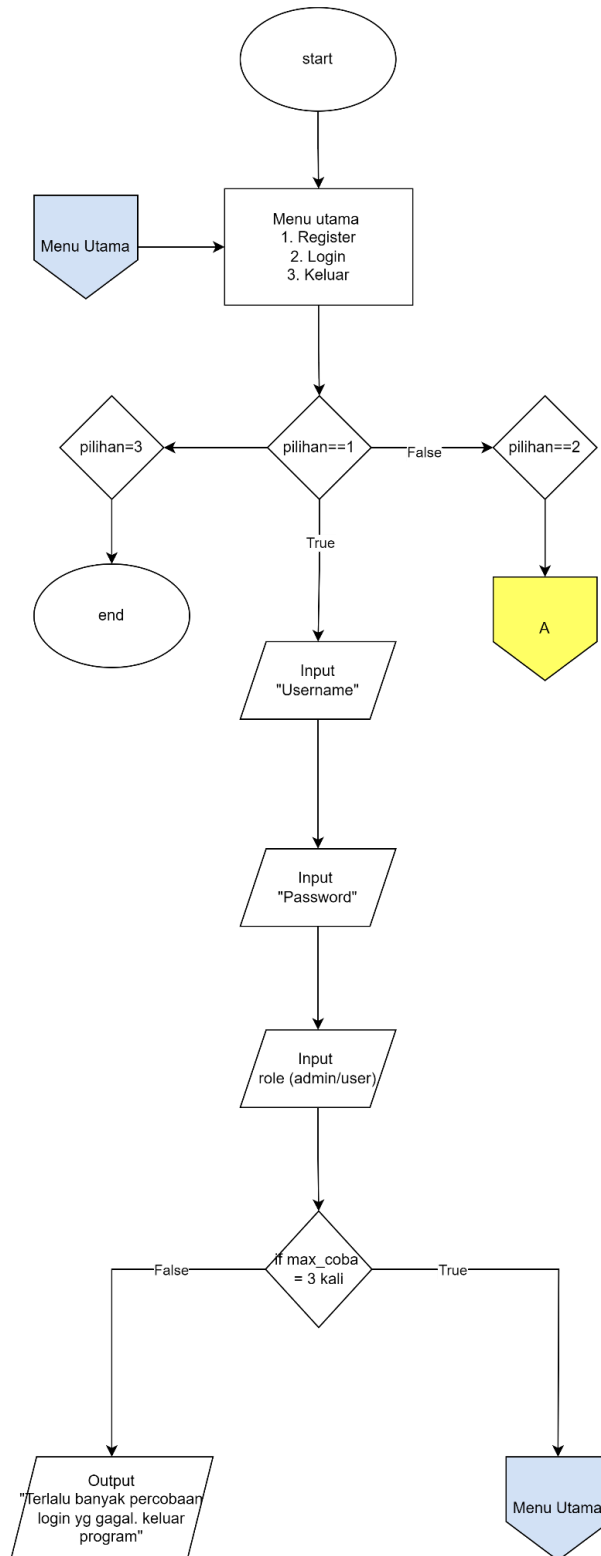


**Disusun oleh:**  
**Nazla Salsabila**  
**Informatika C'24**

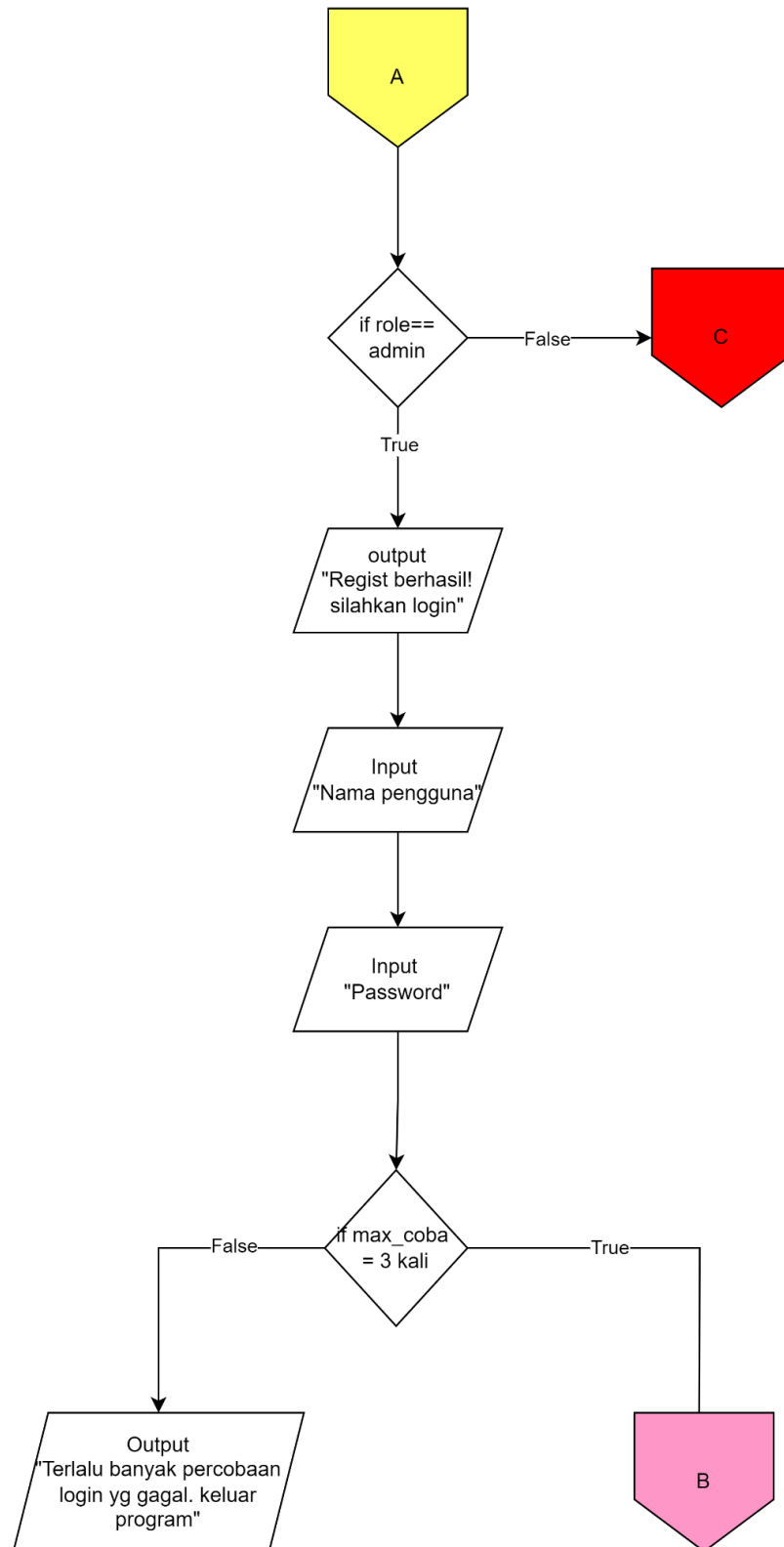
**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**

**2025**

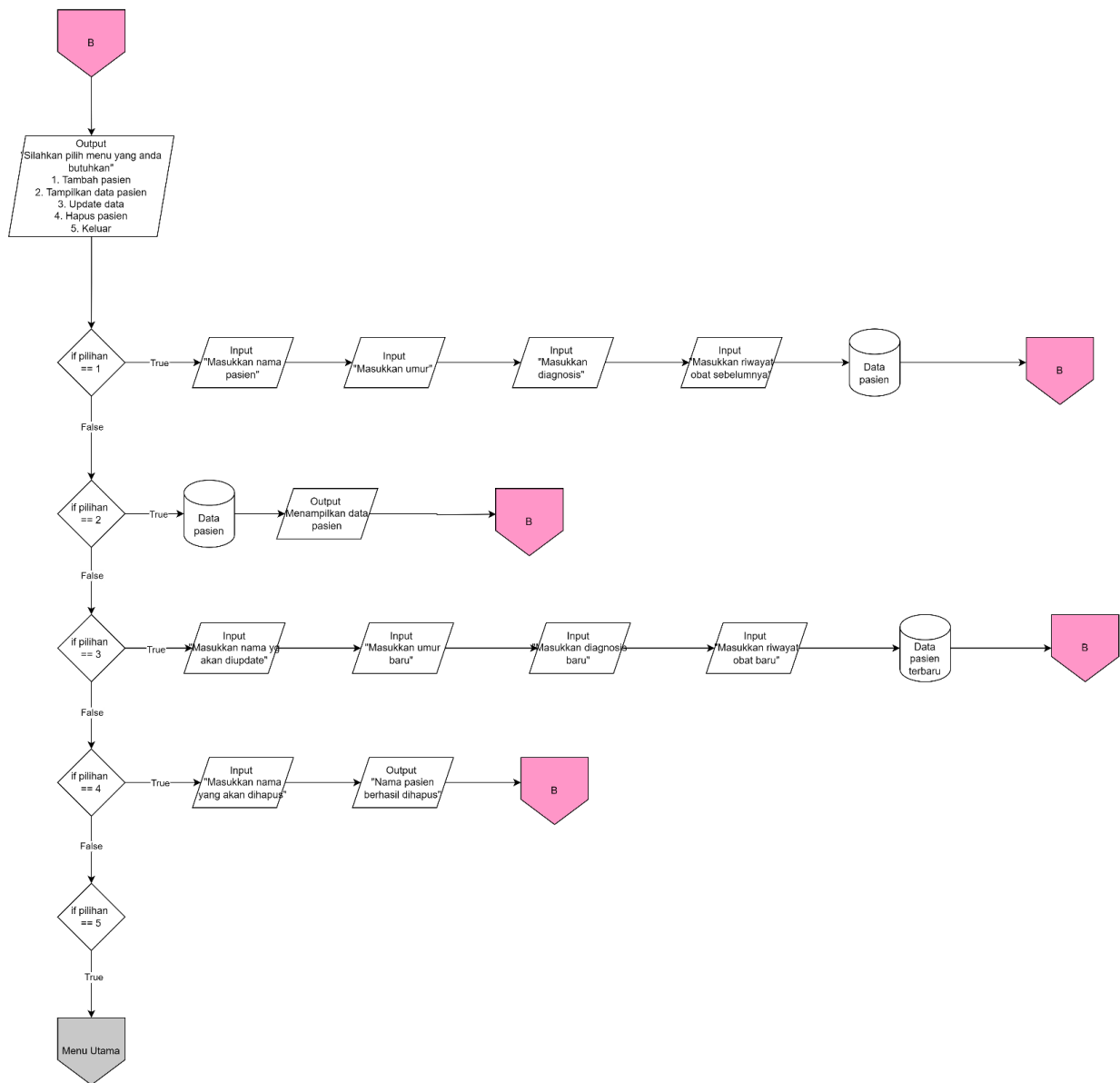
## 1. FLOWCHART



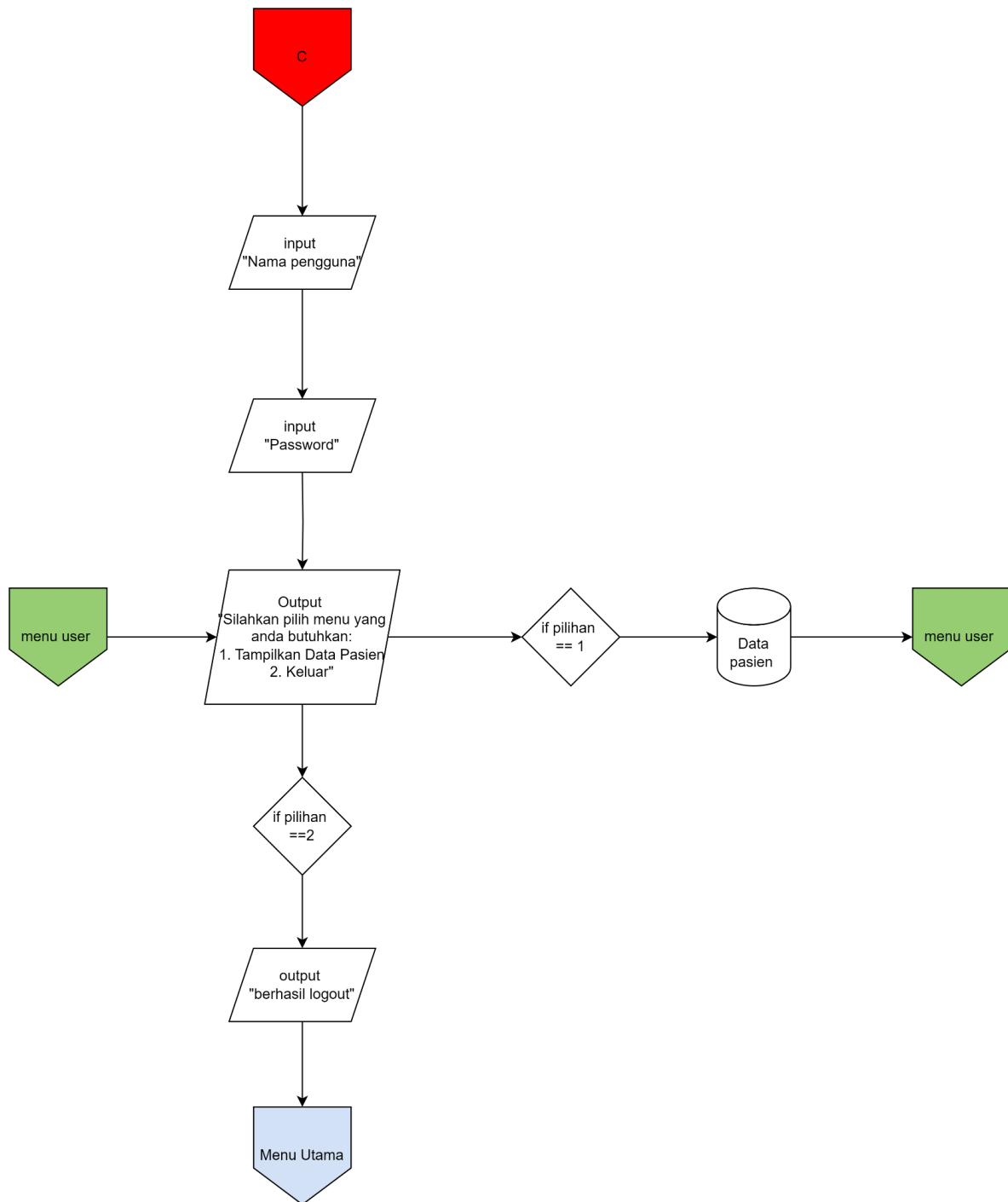
Gambar 1.1 Flowchart page 1



*Gambar 1.2 Flowchart page 2*



*Gambar 1.3 Flowchart page 3*



*Gambar 1.4 Flowchart page 4*

## 2. ANALISIS PROGRAM

Program yang dibuat ini masih tentang penggunaan CRUD yaitu Create, Read, Update, dan Delete pada program rekam medis pasien. Disini pengguna dapat memilih menu yang ingin dilakukan dengan syarat melakukan input nama dan nim atau password pengguna sebagai login, jika salah memasukkan inputan sebanyak tiga kali maka program akan berhenti. Kemudian, program ini menambahkan akses untuk register dan login, sebelum melanjutkan program pengguna harus melakukan registrasi atau mendaftar akun terlebih dahulu, jika sudah maka dilanjutkan ke login menggunakan akun yang sudah berhasil diregistrasi dan pengguna diminta untuk memilih role nya sebagai admin atau user. Role admin disini bisa menambahkan data pasien yang memuat (data seperti nama pasien, umur pasien, diagnosa penyakit dari pasien, dan riwayat obat sebelumnya yang diberikan kepada pasien), melihat tampilan data pasien, dapat mengubah atau mengupdate data pasien terbaru, serta bisa menghapus data pasien tersebut. Sedangkan, role sebagai user hanya bisa melakukan menu melihat tampilan data pasien, dan keluar dari program. Program ini akan terus berulang sampai user ingin berhenti menjalankan program.

Dalam program yang dibuat kali ini diubah agar program memiliki fungsi yang wajib menggunakan parameter dan wajib menggunakan prosedur serta menerapkan penggunaan fungsi rekursif.

Tujuan dibuatnya sebuah program rekam medis pasien ini adalah membantu dalam pengelolaan data pasien, termasuk penambahan, pembaruan, penghapusan dan penampilan informasi pasien secara lebih efisien, tujuan berikutnya program ini memiliki sistem login yang membedakan antara peran admin dan user, sehingga memberikan kontrol akses yang sesuai dan mendapatkan fitur khususnya masing - masing.

Manfaat dari dibuatnya program menggunakan fungsi parameter dan prosedur ini karena parameter memungkinkan fungsi atau prosedur untuk menangani berbagai kasus tanpa perlu menulis logika yang sama berulang kali. Dengan memberikan nama parameter yang deskriptif, kode menjadi lebih mudah dipahami. Parameter menjelaskan data apa saja yang sedang diproses oleh fungsi atau prosedur. Kemudian, manfaat diterapkannya fungsi rekursif pada program sederhana ini adalah dapat menyelesaikan masalah seperti pencarian dalam struktur data pasien, rekursif sering kali menghasilkan kode yang lebih ringkas dan mudah dimengerti, terutama untuk masalah yang bersifat berulang.

Perlu diingat disini bahwa fungsi void dalam program ini artinya adalah fungsi yang tidak mengembalikan nilai sama sekali, biasanya digunakan untuk menjalankan suatu prosedur. Sedangkan, fungsi bool adalah fungsi yang mengembalikan nilai boolean, yaitu true atau false biasanya digunakan untuk memeriksa suatu kondisi atau melakukan evaluasi logika, dan memberikan hasil berupa nilai true atau false.

### 3. SOURCE CODE

```
#include <iostream>
#include <string>
#include <iomanip>

using namespace std;

const int max_coba = 3;
const int max_pasien = 100;
const int max_pengguna = 50;

struct RiwayatObat {
    string diagnosis;
    string riwayatObat;
};

struct Pasien {
    string nama;
    int umur;
    RiwayatObat riwayat;
};

struct Akun {
    string username;
    string password;
    string role;
};

void registrasi(Akun pengguna[], int &jumlahPengguna);
bool login(Akun pengguna[], int jumlahPengguna, string &role);
void tambahPasien(Pasien dataPasien[], int &hitungPasien);
void tampilkanDataPasien(Pasien dataPasien[], int hitungPasien);
```

```

void updatePasien(Pasien dataPasien[], int hitungPasien);
void hapusPasien(Pasien dataPasien[], int &hitungPasien);
void tampilkanDataPasienRekursif(Pasien dataPasien[], int index, int
hitungPasien);
void tampilkanDataPasienRekursif(Pasien dataPasien[], int hitungPasien);

int main() {
    Akun pengguna[max_pengguna];
    int jumlahPengguna = 0;

    Pasien dataPasien[max_pasien];
    int hitungPasien = 0;

    while (true) {
        int menuUtama;
        cout << "Menu Utama:\n";
        cout << "1. Registrasi\n";
        cout << "2. Login\n";
        cout << "3. Keluar\n";
        cout << "Masukkan pilihan Anda: ";
        cin >> menuUtama;

        if (menuUtama == 3) {
            cout << "Program berhenti. Terimakasih sudah mengakses layanan
kami\n";
            break;
        }

        if (menuUtama == 1) {
            registrasi(pengguna, jumlahPengguna);
        } else if (menuUtama == 2) {
            string role;
            bool loginBerhasil = login(pengguna, jumlahPengguna, role);

            if (!loginBerhasil) {
                cout << "Terlalu banyak percobaan login yang gagal. Keluar
program.\n";
                break;
            }

            while (true) {
                int pilihan;

```



```

        cout << "\nSilahkan pilih menu yang anda butuhkan:\n";
        if (role == "admin") {
            cout << "1. Tambah Pasien\n";
            cout << "2. Tampilkan Data Pasien\n";
            cout << "3. Update Pasien\n";
            cout << "4. Hapus Pasien\n";
            cout << "5. Keluar\n";
        } else if (role == "user") {
            cout << "1. Tampilkan Data Pasien\n";
            cout << "2. Keluar\n";
        }
        cout << "Masukkan pilihan Anda: ";
        cin >> pilihan;

        if (role == "admin") {
            if (pilihan == 1) {
                tambahPasien(dataPasien, hitungPasien);
            } else if (pilihan == 2) {
                tampilkanDataPasien(dataPasien, hitungPasien);
            } else if (pilihan == 3) {
                updatePasien(dataPasien, hitungPasien);
            } else if (pilihan == 4) {
                hapusPasien(dataPasien, hitungPasien);
            } else if (pilihan == 5) {
                cout << "Berhasil Logout." << endl;
                break;
            } else {
                cout << "Pilihan tidak valid. Mohon coba lagi." <<
endl;
            }
        } else if (role == "user") {
            if (pilihan == 1) {
                tampilkanDataPasien(dataPasien, hitungPasien);
            } else if (pilihan == 2) {
                cout << "Berhasil Logout." << endl;
                break;
            } else {
                cout << "Pilihan tidak valid. Mohon coba lagi." <<
endl;
            }
        }
    }
}

```

```

        } else {
            cout << "Pilihan tidak valid. Mohon coba lagi." << endl;
        }
    }
    return 0;
}

void registrasi(Akun pengguna[], int &jumlahPengguna) {
    if (jumlahPengguna >= max_pengguna) {
        cout << "Mohon maaf jumlah pengguna sudah melewati maksimum." << endl;
        return;
    }

    Akun akunBaru;
    cout << "Masukkan username: ";
    cin >> akunBaru.username;
    cout << "Masukkan password: ";
    cin >> akunBaru.password;
    cout << "Masukkan role (admin/user): ";
    cin >> akunBaru.role;

    pengguna[jumlahPengguna] = akunBaru;
    jumlahPengguna++;
    cout << "Registrasi berhasil! Silahkan login" << endl;
}

bool login(Akun pengguna[], int jumlahPengguna, string &role) {
    string username, password;
    for (int i = 0; i < max_coba; ++i) {
        cout << "Nama Pengguna: ";
        cin >> username;
        cout << "Password: ";
        cin >> password;

        for (int j = 0; j < jumlahPengguna; ++j) {
            if (pengguna[j].username == username && pengguna[j].password ==
password) {
                role = pengguna[j].role;
                return true;
            }
        }
    }
}

```

```

        cout << "Nama pengguna dan password tidak ditemukan. Silahkan
Registrasi terlebih dahulu" << endl;
    }
    return false;
}

void tambahPasien(Pasien dataPasien[], int &hitungPasien) {
    if (hitungPasien >= max_pasien) {
        cout << "Daftar pasien penuh." << endl;
        return;
    }

    cout << "Masukkan nama: ";
    cin.ignore();
    getline(cin, dataPasien[hitungPasien].nama);
    cout << "Masukkan umur: ";
    cin >> dataPasien[hitungPasien].umur;
    cin.ignore();
    cout << "Masukkan diagnosis: ";
    getline(cin, dataPasien[hitungPasien].riwayat.diagnosis);
    cout << "Masukkan riwayat obat sebelumnya: ";
    getline(cin, dataPasien[hitungPasien].riwayat.riwayatObat);
    hitungPasien++;
}

void tampilkanDataPasien(Pasien dataPasien[], int hitungPasien) {
    cout << left << setw(20) << "Nama" << setw(5) << "Umur" << setw(20) <<
"Diagnosis" << setw(15) << "Riwayat obat" << endl;
    cout <<
"-----" << endl;
    tampilkanDataPasienRekursif(dataPasien, 0, hitungPasien);
}

void tampilkanDataPasienRekursif(Pasien dataPasien[], int index, int
hitungPasien) {
    if (index >= hitungPasien) return;
    cout << left << setw(20) << [index].nama
        << setw(5) << dataPasien[index].umur
        << setw(20) << dataPasien[index].riwayat.diagnosis
        << setw(15) << dataPasien[index].riwayatdataPasient.riwayatObat <<
endl;
    tampilkanDataPasienRekursif(dataPasien, index + 1, hitungPasien);
}

```

```

}

void tampilkanDataPasienRekursif(Pasien dataPasien[], int hitungPasien) {
    tampilkanDataPasienRekursif(dataPasien, 0, hitungPasien);
}

void updatePasien(Pasien dataPasien[], int hitungPasien) {
    string nama;
    cout << "Masukkan nama yang akan diupdate: ";
    cin.ignore();
    getline(cin, nama);

    int temukan = 0;
    for (int i = 0; i < hitungPasien; ++i) {
        if (dataPasien[i].nama == nama) {
            cout << "Masukkan umur baru: ";
            cin >> dataPasien[i].umur;
            cin.ignore();
            cout << "Masukkan diagnosis baru: ";
            getline(cin, dataPasien[i].riwayat.diagnosis);
            cout << "Masukkan riwayat obat baru: ";
            getline(cin, dataPasien[i].riwayat.riwayatObat);
            temukan = 1;
            break;
        }
    }

    if (!temukan) {
        cout << "Pasien tidak ditemukan." << endl;
    }
}

void hapusPasien(Pasien dataPasien[], int &hitungPasien) {
    string nama;
    cout << "Masukkan nama yang akan dihapus: ";
    cin.ignore();
    getline(cin, nama);

    int temukan = 0;
    for (int i = 0; i < hitungPasien; ++i) {
        if (dataPasien[i].nama == nama) {
            for (int j = i; j < hitungPasien - 1; ++j) {

```

```
        dataPasien[j] = dataPasien[j + 1];
    }
    hitungPasien--;
    cout << "Nama Pasien berhasil dihapus." << endl;
    temukan = 1;
    break;
}
}

if (!temukan) {
    cout << "Nama Pasien tidak ditemukan." << endl;
}
}
```

#### 4. SCREENSHOT HASIL PROGRAM

```
asalsabila-PT-5 } ; if ($?) { .\2409106108-N
Menu Utama:
1. Registrasi
2. Login
3. Keluar
Masukkan pilihan Anda: 1
Masukkan username: Nazla_Salsabila
Masukkan password: 2409106108
Masukkan role (admin/user): admin
Registrasi berhasil! Silahkan login
Menu Utama:
1. Registrasi
2. Login
3. Keluar
Masukkan pilihan Anda: █
```

*Gambar 4.1 Menu registrasi sebagai admin*

```
Registrasi berhasil! Silahkan login
Menu Utama:
1. Registrasi
2. Login
3. Keluar
Masukkan pilihan Anda: 2
Nama Pengguna: Nazla_Salsabila
Password: 2409106108

Silahkan pilih menu yang anda butuhkan:
1. Tambah Pasien
2. Tampilkan Data Pasien
3. Update Pasien
4. Hapus Pasien
5. Keluar
Masukkan pilihan Anda: █
```

*Gambar 4.2 menu login sebagai admin*

```
Silahkan pilih menu yang anda butuhkan:  
1. Tambah Pasien  
2. Tampilkan Data Pasien  
3. Update Pasien  
4. Hapus Pasien  
5. Keluar  
Masukkan pilihan Anda: 1  
Masukkan nama: Zaki  
Masukkan umur: 21  
Masukkan diagnosis: Radang tenggorokan  
Masukkan riwayat obat sebelumnya: Coolings 5
```

*Gambar 4.3 Tambah pasien*

```
Silahkan pilih menu yang anda butuhkan:  
1. Tambah Pasien  
2. Tampilkan Data Pasien  
3. Update Pasien  
4. Hapus Pasien  
5. Keluar  
Masukkan pilihan Anda: 1  
Masukkan nama: Amelia  
Masukkan umur: 25  
Masukkan diagnosis: Demam berdarah  
Masukkan riwayat obat sebelumnya: Oralit
```

*Gambar 4.4 Tambah pasien*

```

Silahkan pilih menu yang anda butuhkan:
1. Tambah Pasien
2. Tampilkan Data Pasien
3. Update Pasien
4. Hapus Pasien
5. Keluar
Masukkan pilihan Anda: 1
Masukkan nama: Ujang
Masukkan umur: 32
Masukkan diagnosis: Influenza
Masukkan riwayat obat sebelumnya: Paracetamol

```

*Gambar 4.5 Tambah pasien*

```

Silahkan pilih menu yang anda butuhkan:
1. Tambah Pasien
2. Tampilkan Data Pasien
3. Update Pasien
4. Hapus Pasien
5. Keluar
Masukkan pilihan Anda: 2

```

Nama	Umur	Diagnosis	Riwayat obat
Zaki	21	Radang tenggorokan	Coolings 5
Amelia	25	Demam berdarah	Oralit
Ujang	32	Influenza	Paracetamol

```

Silahkan pilih menu yang anda butuhkan:
1. Tambah Pasien
2. Tampilkan Data Pasien
3. Update Pasien
4. Hapus Pasien
5. Keluar
Masukkan pilihan Anda: █

```

*Gambar 4.6 Tampilan data pasien*



```

Silahkan pilih menu yang anda butuhkan:
1. Tambah Pasien
2. Tampilkan Data Pasien
3. Update Pasien
4. Hapus Pasien
5. Keluar
Masukkan pilihan Anda: 3
Masukkan nama yang akan diupdate: Zaki
Masukkan umur baru: 22
Masukkan diagnosis baru: Malaria
Masukkan riwayat obat baru: Primaquine

```

*Gambar 4.7 Update data pasien*

```

Silahkan pilih menu yang anda butuhkan:
1. Tambah Pasien
2. Tampilkan Data Pasien
3. Update Pasien
4. Hapus Pasien
5. Keluar
Masukkan pilihan Anda: 2
Nama          Umur  Diagnosis          Riwayat obat
-----
Zaki          22    Malaria            Primaquine
Amelia        25    Demam berdarah     Oralit
Ujang         32    Influenza          Paracetamol

Silahkan pilih menu yang anda butuhkan:
1. Tambah Pasien
2. Tampilkan Data Pasien
3. Update Pasien
4. Hapus Pasien
5. Keluar
Masukkan pilihan Anda: █

```

*Gambar 4.8 Tampilan data setelah di update*

Silahkan pilih menu yang anda butuhkan:

1. Tambah Pasien
2. Tampilkan Data Pasien
3. Update Pasien
4. Hapus Pasien
5. Keluar

Masukkan pilihan Anda: 4

Masukkan nama yang akan dihapus: Ujang

Nama Pasien berhasil dihapus.

*Gambar 4.9 Hapus pasien*

Silahkan pilih menu yang anda butuhkan:

1. Tambah Pasien
2. Tampilkan Data Pasien
3. Update Pasien
4. Hapus Pasien
5. Keluar

Masukkan pilihan Anda: 2

Nama	Umur	Diagnosis	Riwayat obat
------	------	-----------	--------------

-----

Zaki	22	Malaria	Primaquine
------	----	---------	------------

Amelia	25	Demam berdarah	Oralit
--------	----	----------------	--------

Silahkan pilih menu yang anda butuhkan:

1. Tambah Pasien
2. Tampilkan Data Pasien
3. Update Pasien
4. Hapus Pasien
5. Keluar

Masukkan pilihan Anda: █

*Gambar 4.10 Tampilan setelah data pasien dihapus*

```
Masukkan pilihan Anda: 5
Berhasil Logout.
Menu Utama:
1. Registrasi
2. Login
3. Keluar
Masukkan pilihan Anda: █
```

*Gambar 4.11 Keluar dari menu*

```
Menu Utama:
1. Registrasi
2. Login
3. Keluar
Masukkan pilihan Anda: 1
Masukkan username: Najlaa
Masukkan password: 1701
Masukkan role (admin/user): user
Registrasi berhasil! Silahkan login
Menu Utama:
1. Registrasi
2. Login
3. Keluar
Masukkan pilihan Anda: █
```

*Gambar 4.12 Registrasi sebagai user*

```

Registrasi berhasil! Silahkan login
Menu Utama:
1. Registrasi
2. Login
3. Keluar
Masukkan pilihan Anda: 2
Nama Pengguna: Najlaa
Password: 1701

Silahkan pilih menu yang anda butuhkan:
1. Tampilkan Data Pasien
2. Keluar
Masukkan pilihan Anda: █

```

*Gambar 4.13 login sebagai user*

```

Silahkan pilih menu yang anda butuhkan:
1. Tampilkan Data Pasien
2. Keluar
Masukkan pilihan Anda: 1

```

Nama	Umur	Diagnosis	Riwayat obat
Zaki	22	Malaria	Primaquine
Amelia	25	Demam berdarah	Oralit

```

Silahkan pilih menu yang anda butuhkan:
1. Tampilkan Data Pasien
2. Keluar
Masukkan pilihan Anda: █

```

*Gambar 4.14 Tampilan pilihan 1 untuk user*

```
Silahkan pilih menu yang anda butuhkan:  
1. Tampilkan Data Pasien  
2. Keluar  
Masukkan pilihan Anda: 2  
Berhasil Logout.  
Menu Utama:  
1. Registrasi  
2. Login  
3. Keluar  
Masukkan pilihan Anda: █
```

*Gambar 4.15 Tampilan pilihan 2 untuk user*

```
Menu Utama:  
1. Registrasi  
2. Login  
3. Keluar  
Masukkan pilihan Anda: 3  
Program berhenti. Terimakasih sudah mengakses layanan kami  
PS C:\praktikum-apl\post-test\post-test-3> █
```

*Gambar 4.16 Keluar dari program*

```
Masukkan pilihan Anda: 1
Masukkan username: Nazla_Salsabila
Masukkan password: 2409106108
Masukkan role (admin/user): admin
Registrasi berhasil! Silahkan login
Menu Utama:
1. Registrasi
2. Login
3. Keluar
Masukkan pilihan Anda: 2
Nama Pengguna: najla
Password: 1111
Nama pengguna atau password salah. Coba lagi.
Nama Pengguna: nazel
Password: 2345
Nama pengguna atau password salah. Coba lagi.
Nama Pengguna: salsabila
Password: 1701
Nama pengguna atau password salah. Coba lagi.
Terlalu banyak percobaan login yang gagal. Keluar program.
PS C:\praktikum-apl\post-test\post-test-3>
```

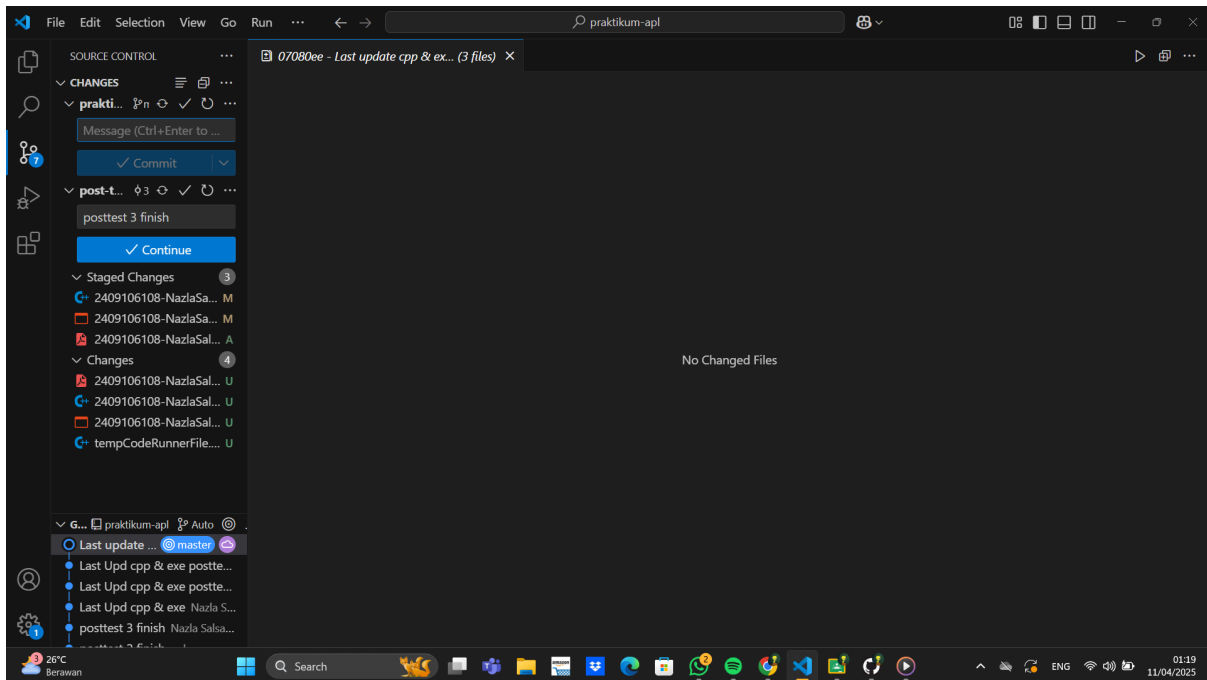
*Gambar 4.17 Percobaan login sebanyak 3 kali*

```
Menu Utama:
1. Registrasi
2. Login
3. Keluar
Masukkan pilihan Anda: 1
Masukkan username: Nazlasalsabila
Masukkan password: 2409106108
Masukkan role (admin/user): jamet
Registrasi berhasil! Silahkan login
Menu Utama:
1. Registrasi
2. Login
3. Keluar
Masukkan pilihan Anda: 2
Nama Pengguna: Nazlasalsabila
Password: 2409106108

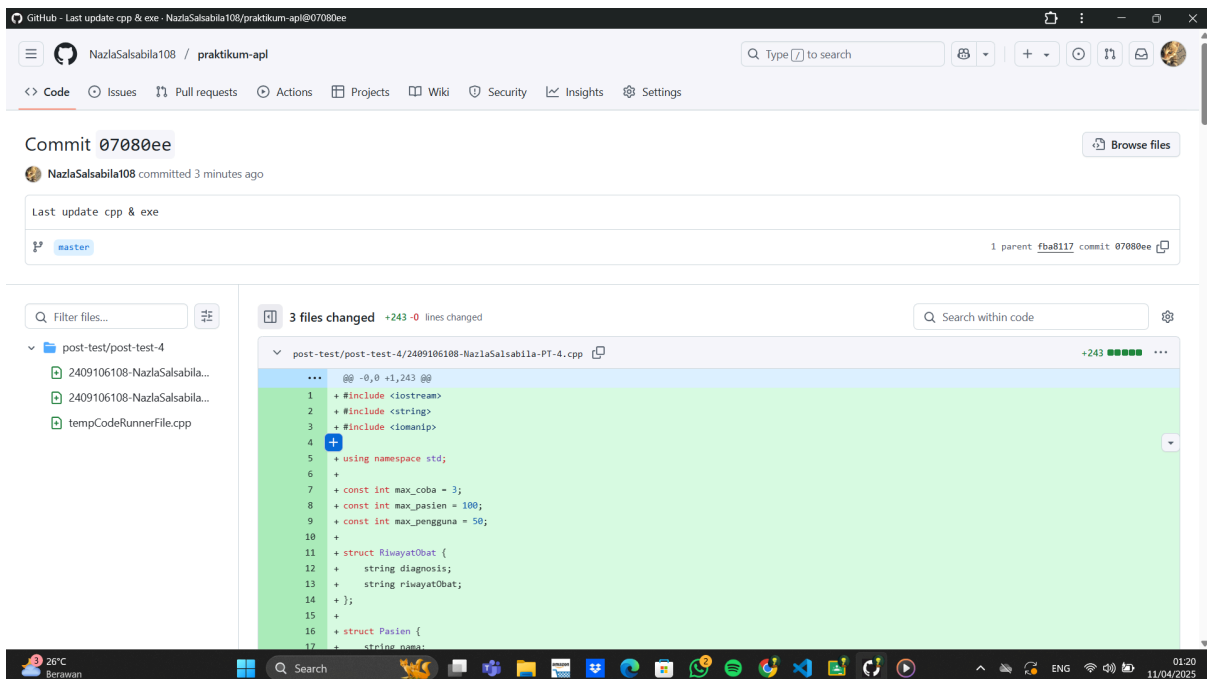
Silahkan pilih menu yang anda butuhkan:
Masukkan pilihan Anda: █
```

*Gambar 4.18 Percobaan memilih selain dalam role*

## 5. PENJELASAN GIT



Gambar 5.1 Tampilan terminal git bash di vscode



Gambar 5.2 Tampilan dalam GitHub



Caranya :

### **Buka file praktikum-apl**

Select folder tersebut untuk menampilkan semua isi file file dalam folder praktikum-apl, kita fokuskan ke change untuk melihat tambahan folder baru yang ditambahkan

### **Lakukan git push dengan message “Last Upd cpp & exe”**

Jika github dan vscode sudah terhubung maka langkah selanjutnya adalah langsung melakukan git & push pada message di change dengan pesan “Last Upd cpp & exe” tunggu beberapa detik. Cek github, maka dengan otomatis file yang sudah kita push akan muncul di repository github kita, lalu pada bagian bawah buka format GRAPH dan pencet pada bagian awal Last Upd cpp & exe untuk melihat perubahan yang dilakukan tampilan file nya seperti pada gambar di atas.