

LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT

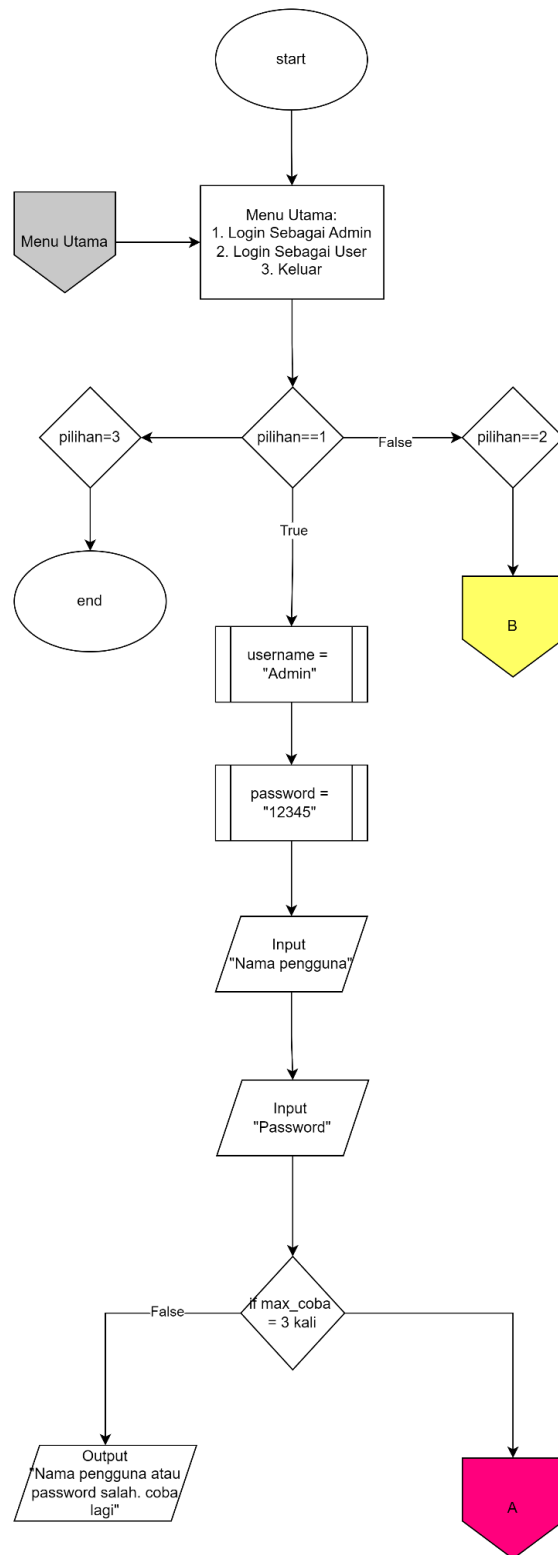


Disusun oleh:
Nazla Salsabila
Informatika C'24

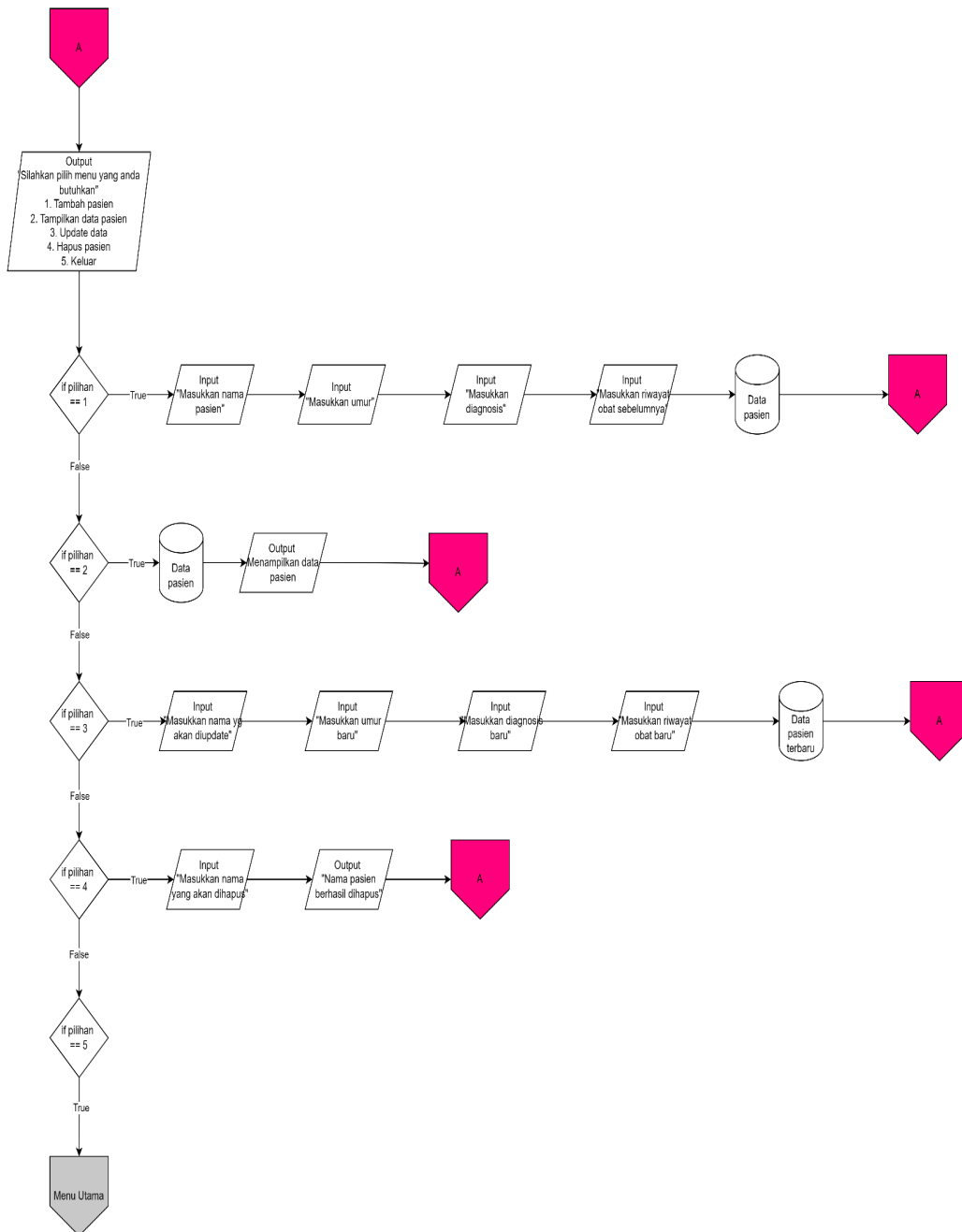
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA

2025

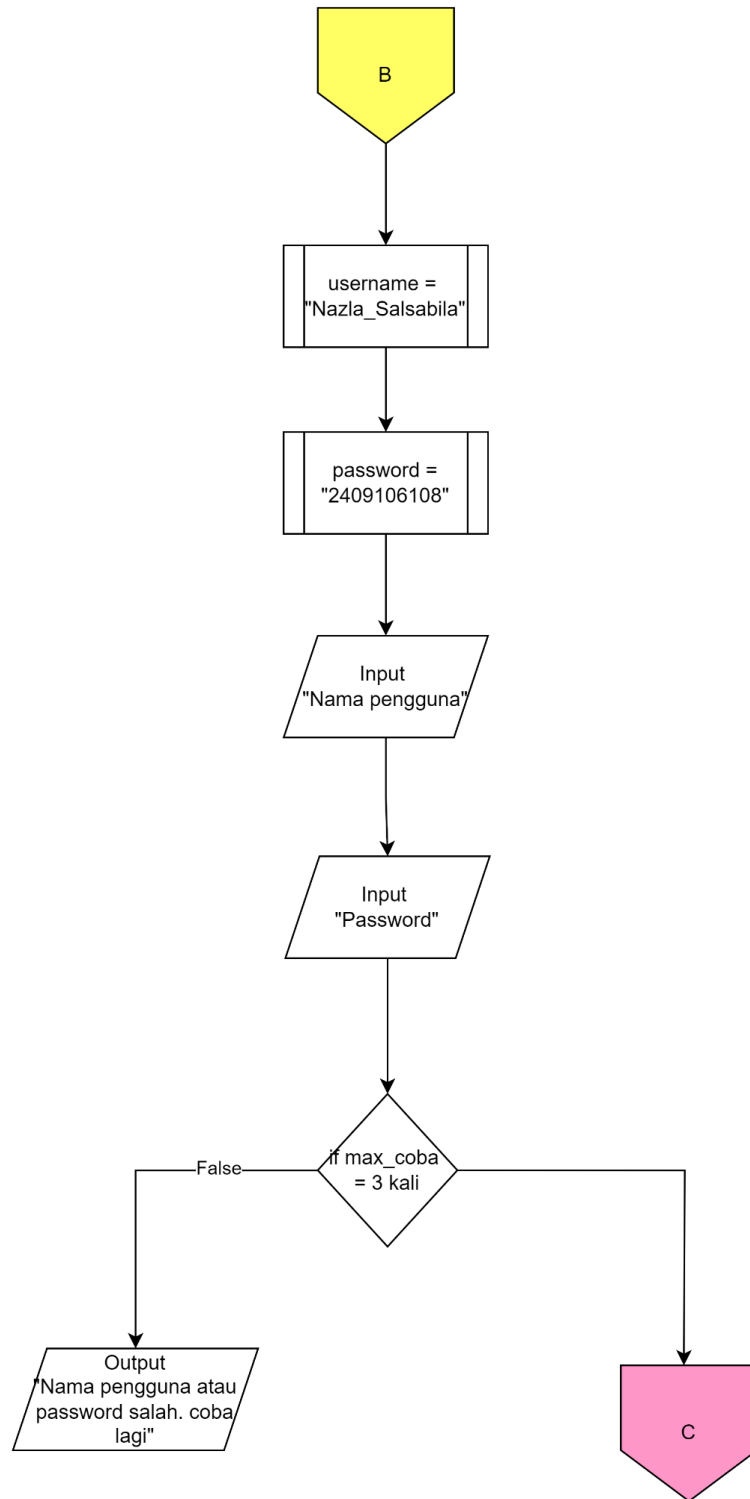
1. FLOWCHART



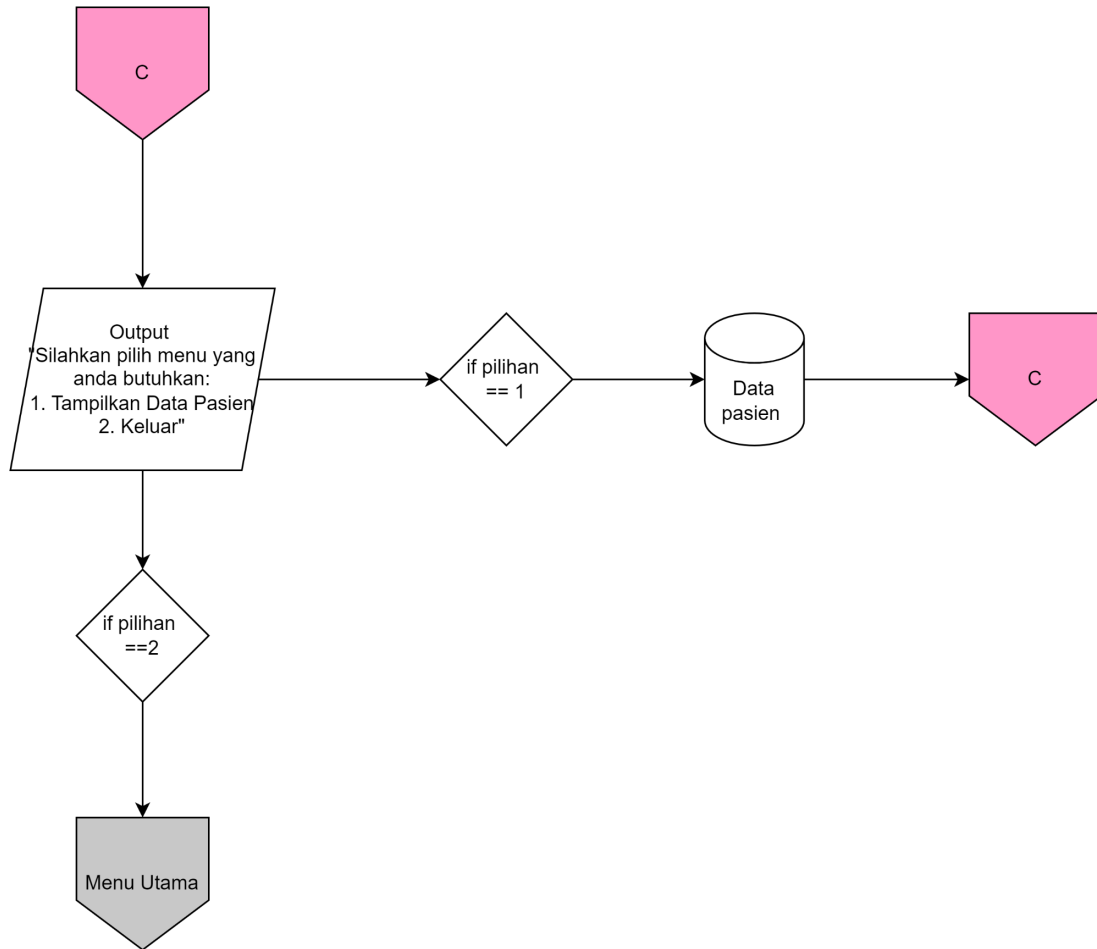
Gambar 1.1 Flowchart page 1



Gambar 1.2 Flowchart page 2



Gambar 1.3 Flowchart page 3



Gambar 1.4 Flowchart page 4

Deskripsi dari flowchart yang dibuat:

Untuk menggambarkan flowchart ini, kita menggunakan simbol-simbol berikut:

- Oval** : Untuk titik mulai dan akhir dari program atau alur
- Persegi panjang** : Untuk menggambarkan langkah atau operasi yang dilakukan dalam suatu proses
- Jajar gemcang** : Untuk melakukan proses input dan output
- Belah ketupat** : Untuk keputusan yaitu sebagai validasi dan pilihan
- Panah** : Untuk menunjukkan alur dari satu langkah ke langkah-langkah selanjutnya
- Segiempat** : Menunjukan bahwa alur proses akan dilanjutkan ke halaman lain
- Tabung** : Menunjukkan bahwa data disimpan ke dalam database

1. *Start* merupakan titik awal akan berjalannya program.
2. Menampilkan proses untuk mengakses tiga menu utama pada program
3. Jika pengguna memilih menu == 1 maka dia akan login sebagai pengguna admin, jika memilih menu == 2 maka dia akan login sebagai pengguna user, jika memilih menu == 3 maka program akan berhenti atau keluar.
4. Login sebagai pengguna admin, output nya akan meminta pengguna untuk memasukkan username dan password yang sudah dideklarasikan.
5. Maksimal percobaan hanya sebanyak tiga kali saja, lebih dari tiga kali mencoba atau gagal maka program akan berhenti.
6. Jika berhasil login sebagai pengguna admin, maka output yang ditampilkan adalah menu CRUD yang bisa bisa diakses
 - Jika pilihan == 1, maka admin bisa menambah data pasien dengan cara memasukkan nama pasien, umur pasien, diagnosa penyakit pasien, dan riwayat obat sebelumnya dari pasien. Kemudian data tersebut disimpan ke dalam database pasien
 - Jika pilihan == 2, menampilkan database pasien yang sudah ditambahkan
 - Jika pilihan == 3, mengubah data pasien dengan data yang baru akibat kesalahan atau kekeliruan. Input nama pasien yang datanya akan diperbarui, masukkan usia pasien, masukkan diagnosis baru pasien,

masukkan riwayat obat baru pasien, dan data yang ter update tersebut akan disimpan lagi dalam database pasien.

- Jika pilihan == 4, yaitu masukkan nama pasien yang ingin dihapus, kemudian beralih ke alur “Nama pasien sudah dihapus”
- Jika pilihan == 5, keluar dari menu CRUD admin, maka akan kembali lagi berulang ke menu utama

7. Langkah selanjutnya, login sebagai pengguna user outputnya akan meminta pengguna untuk memasukkan username dan password yang sudah dideklarasikan.
8. Maksimal percobaan hanya sebanyak tiga kali saja, lebih dari tiga kali mencoba atau gagal maka program akan berhenti.
9. Jika berhasil login sebagai pengguna user, maka output yang ditampilkan hanya dua menu saja yaitu menampilkan data pasien dan keluar.
 - Jika pilihan == 1, menampilkan database pasien yang sudah ditambahkan, di update, bahkan data pasien yang sudah dihapus oleh admin
 - Jika pilihan == 2, keluar dari menu user dengan output “Berhasil logout” dan berulang kembali ke menu utama.
10. Jika ingin keluar dari program maka pengguna menginput menu ke tiga pada bagian menu utama yaitu keluar dan program akan berhenti.

2. ANALISIS PROGRAM

2.1 Deskripsi Singkat Program

Program yang dibuat ini 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, user dapat memilih menu yang ingin dilakukan. Program ini memiliki dua akses akun yaitu ada akun khusus untuk admin yang 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, untuk akun user hanya bisa melakukan menu seperti menambahkan data pasien terbaru (seperti tambah nama, umur pasien, diagnosa penyakit, dan riwayat obat pasien), melihat tampilan data, dan keluar dari program.

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 ini untuk melatih pemahaman tentang penggunaan array multidimensi dalam menyimpan sebuah data dan dengan adanya sistem login, program ini membantu untuk melindungi data pasien yang tidak sah misalnya.

2.2 Penjelasan Alur & Algoritma

Langkah-langkah berjalannya program rekam medis pasien ini adalah sebagai berikut

1. Langkah pertama kita harus mengenal kode kode yang digunakan di awal program ini sebelum melakukan proses input nama pengguna. Kata kunci yang digunakan untuk mendeklarasikan bahwa nilai dari variabel tersebut tidak dapat diubah, maksimal mencoba login yaitu sebanyak tiga kali, jika lebih maka program akan berhenti dan jumlah maksimum yang dapat dicatat dalam program berjumlah 100 pasien saja.

struct akun sebagai definisi sebuah struktur yang beranggotakan 'string username', 'string password', dan 'string role' (peran pengguna yang bisa berupa admin atau user).

Kode program:

```
const int max_coba = 3;

const int max_pasien = 100;

struct Akun {

    string username;

    string password;

    string role;

};
```

2. 'pengguna' ini adalah array dari tipe 'Akun' dimana dia akan menyimpan beberapa objek dari stuktur 'Akun' terdapat tiga elemen dalam array 'pengguna'. Kemudian pada data pasien menyimpan informasi tentang pasien [max_pasien][4] deklarasi array multidimensi yang artinya array ini hanya menyimpan data hingga 100 pasien dan angka 4 dalam array itu menunjukkan bahwa setiap pasien memiliki empat atribut yang disimpan dalam array yang terdiri dari: Nama pasien, Umur pasien, Diagnosis pasien, dan Riwayat obat sebelumnya.

Kode program:

```
int main() {  
  
    Akun pengguna[] = {  
  
        {"Admin", "12345", "admin"},  
  
        {"Nazla_Salsabila", "2409106108", "user"},  
  
        {"Najla", "1701", "admin"}  
  
    };  
  
    string dataPasien[max_pasien][4];  
  
    int hitungPasien = 0;
```

3. Kita akan diminta untuk memasukkan akun login kita sebagai admin atau sebagai user dengan tiga opsi pada menu utama

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

Loop while true digunakan untuk menampilkan menu utama ini berulang kali, jika kita memilih opsi 3 maka pengguna akan berhenti menjalankan programnya.

Jika pengguna menginput menu 1, maka dia akan menjalankan program ini sebagai admin, jika pengguna menginput menu 2, maka dia akan menjalankan program ini

sebagai user, dan jika pengguna menginput menu 3, program akan otomatis keluar dan berhenti.

4. Disini kita akan melakukan proses input untuk masuk ke dalam program baik sebagai admin atau sebagai user pada menu utama tadi, dengan nama dan password yang sudah ditentukan. Bagian kode ini bertujuan untuk meminta pengguna menginput nama pengguna dan password hingga tiga kali. Jika pengguna gagal login setelah tiga kali percobaan, program akan berhenti.

```
string username, password, role;

    int masuk = 0;

    for (int i = 0; i < max_coba; ++i) {

        cout << "Nama Pengguna: ";

        cin >> username;

        cout << "Password: ";

        cin >> password;
```

Setelah menginput nama pengguna, program akan memproses kecocokan username dan password login yang digunakan

```
for (const auto & user : pengguna) {

    if (user.username == username && user.password == password)
    {

        masuk = 1;

        role = user.role;

        break;

    }

}
```

Jika login program gagal sebanyak tiga kali percobaan maka program akan mengeluarkan output seperti pada gambar kode program berikut:

```
if (masuk) {  
  
    break;  
  
} else {  
  
    cout << "Nama pengguna atau password salah. Coba lagi." <<  
endl;  
  
    }  
  
    }  
  
    if (!masuk) {  
  
        cout << "Terlalu banyak percobaan login yang gagal. Keluar  
program." << endl;  
  
        return 1;  
  
    }
```

5. Jika sudah berhasil login, output yang ditampilkan untuk pengguna yaitu bisa memilih menu yang ditampilkan

kode program:

```
while (true) {  
  
    int pilihan;  
  
    cout << "\nSilahkan pilih menu yang anda butuhkan:\n";  
  
    if (role == "admin") {  
  
        cout << "1. Tambah Pasien\n";  
  
        cout << "2. Tampilkan Pasien\n";  
  
        cout << "3. Update Pasien\n";  
  
    }
```

```

        cout << "4. Hapus Pasien\n";

        cout << "5. Keluar\n";

    } else if (role == "user") {

        cout << "1. Tambah Pasien\n";

        cout << "2. Tampilkan Pasien\n";

        cout << "5. Keluar\n";

    }

    cout << "Masukkan pilihan Anda: ";

    cin >> pilihan;

```

Jika dia terdaftar sebagai admin maka fitur yang didapatkan dan ditampilkan adalah

- Jika memilih menu 1 (Tambah Pasien)

Fitur admin yang pertama ini adalah admin dapat menambahkan data pasien dimana, jika pasien yang ditambahkan lebih dari jumlah maksimalnya maka program akan mengeluarkan output “Daftar penuh”. Jumlah pasien diproses menggunakan kode program seperti di bawah ini

```

if (role == "admin") {

    if (pilihan == 1) {

        if (hitungPasien >= max_pasien) {

            cout << "Daftar penuh." << endl;

            continue;

        }

    }

}

```

Sudah selesai memproses jumlah, maka program akan menampilkan data apa saja yang dibutuhkan untuk menambah pasien

```

cout << "Masukkan nama: ";

    cin.ignore();

    getline(cin, dataPasien[hitungPasien][0]);

    cout << "Masukkan umur: ";

    getline(cin, dataPasien[hitungPasien][1]);

    cout << "Masukkan diagnosis: ";

    getline(cin, dataPasien[hitungPasien][2]);

    cout << "Masukkan riwayat obat sebelumnya: ";

    getline(cin, dataPasien[hitungPasien][3]);

    hitungPasien++;

```

Disini pengguna admin diminta untuk menginput data data mulai dari memasukkan umur hingga memasukkan riwayat obat.

- Jika memilih menu 2 (Tampilkan data pasien)

Setelah berhasil menambahkan data data pasien yang diinginkan program, admin memiliki fitur untuk bisa melihat data yang sudah dimasukkan, data tersebut akan ditampilkan pada program menggunakan kode seperti pada gambar berikut

```

else if (pilihan == 2) {

    cout << left << setw(20) << "Nama" << setw(5) <<
"Umur" << setw(20) << "Diagnosis" << setw(15) << "Riwayat obat"
<< endl;

    cout <<
"-----"
---" << endl;

    for (int i = 0; i < hitungPasien; ++i) {

```

```

        cout << left << setw(20) << dataPasien[i][0]
<< setw(5) << dataPasien[i][1] << setw(20) << dataPasien[i][2] <<
setw(15) << dataPasien[i][3] << endl;

    }

```

Dimana data yang ditampilkan pada output ini nanti hanya memiliki garis pembatas antar header (judul data) dan data pasien yang dimasukkan. Kita menggunakan jarak pembatas yang sudah di set agar data bisa dibaca dengan rapi dan tidak berdempetan penempatannya.

- Jika memilih menu 3 (Memperbarui data pasien)

Fitur ini hanya admin yang bisa mengaksesnya untuk memperbarui data pasien terbaru dengan menjalankan kode program seperti ini

```

else if (pilihan == 3) {

    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][0] == nama) {

            cout << "Masukkan umur baru: ";

            getline(cin, dataPasien[i][1]);

            cout << "Masukkan diagnosis baru: ";

            getline(cin, dataPasien[i][2]);

            cout << "Masukkan riwayat obat baru: ";

```

```

        getline(cin, dataPasien[i][3]);

        temukan = 1;

        break;
    }
}

```

Dimana pengguna admin bisa memperbarui atau bahkan mengubah data terbaru pasiennya dengan data pasien yang lama atau sebelumnya, dengan cara menginput kembali nama pasien yang ingin diubah datanya, jika berhasil maka admin menginput kembali data terbaru pasien, jika tidak program akan mengeluarkan output bahwa nama pasien itu tidak ditemukan dan kembali melakukan penginputan untuk memilih menu. Kode program:

```

if (!temukan) {

    cout << "Pasien tidak ditemukan." << endl;
}

```

- Jika memilih menu 4 (Menghapus data pasien)

Sama seperti fitur ketiga khusus admin, fitur ini hanya bisa dijalankan oleh pengguna admin untuk menghapus data pasien yang sudah lama atau yang ingin dihapus kode porgramnya

```

else if (pilihan == 4) {

    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][0] == nama) {
                for (int j = i; j < hitungPasien - 1;
++j) {
                    for (int k = 0; k < 4; ++k) {
                        dataPasien[j][k] = dataPasien[j +
1][k];
                    }
                }
            }
        }

```

Dimana admin harus memasukkan nama pasien yang ingin dihapus, maka program akan otomatis menghapus seluruh data atas nama pasien tersebut. Jika sudah benar memasukkan nama pasien yang terdaftar maka kode program yang dijalankan adalah

```

hitungPasien--;

        cout << "Nama Pasien berhasil dihapus."
<< endl;

        temukan = 1;

        break;

```

Jika tidak ditemukan maka program akan menjalankan kode berikut

```

if (!temukan) {

        cout << "Nama Pasien tidak ditemukan." <<
endl;

```

Lalu pengguna admin diminta kembali untuk melakukan perulangan memilih menu yang dibutuhkan.

- Jika memilih menu 5 (Keluar program)

Program akan kembali ke menu utama untuk memilih kembali login sebagai pengguna yang mana, dan program akan benar benar berhenti jika pengguna memilih opsi keluar di bagian menu utama.

```
else if (pilihan == 5) {
    cout << "Keluar program" << endl;
    break;
} else {
    cout << "Pilihan tidak valid. Mohon coba lagi."
<< endl;
}
```

Jika ingin memilih menu selain 5 untuk keluar program, maka program akan menampilkan output “Pilihan tidak valid. Mohon coba lagi.” Lalu berulang kembali untuk memilih menu yang dibutuhkan sampai pengguna memilih menu untuk berhenti.

6. Cara mengakses akun user, yaitu pengguna harus memilih login sebagai user pada bagian menu utama, kemudian dilanjut menginput username dan password yang sudah ditentukan di dalam variabel array di bagian awal program. Untuk pengguna user hanya bisa mengakses dua fitur saja yaitu, fitur melihat tampilan data pasien, dan keluar dari program.

Kode program

```
else if (role == "user") {
    if (pilihan == 1) {
        cout << left << setw(20) << "Nama" << setw(5)
<< "Umur" << setw(20) << "Diagnosis" << setw(15) << "Riwayat obat" <<
endl;
```

```

cout <<
"-----"
<< endl;

    for (int i = 0; i < hitungPasien; ++i) {

        cout << left << setw(20) <<
dataPasien[i][0] << setw(5) << dataPasien[i][1] << setw(20) <<
dataPasien[i][2] << setw(15) << dataPasien[i][3] << endl;

    }

```

Disini login sebagai user hanya bisa melihat tampilan data yang sudah ditambahkan, diubah, dan dihapus oleh admin.

Langkah untuk mengakses dua fitur user ini untuk melihat data pasien dan keluar dari program hampir sama saja caranya dengan mengakses pada fitur admin, yang membedakannya hanya untuk akun user yang tidak memiliki akses untuk menambahkan data pasien, mengubah data pasien serta menghapus data pasien.

3. SOURCE CODE

```

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

using namespace std;

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

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

int main() {
    Akun pengguna[] = {
        {"Admin", "12345", "admin"},

```

```

        {"Nazla_Salsabila", "2409106108", "user"},
        {"Najla", "1701", "admin"}
    };

    string dataPasien[max_pasien][4];
    int hitungPasien = 0;

    string username, password, role;
    int masuk = 0;

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

        if (menuUtama == 3) {
            cout << "Program berhenti.\n";
            break;
        }

        if (menuUtama == 1 || menuUtama == 2) {
            for (int i = 0; i < max_coba; ++i) {
                cout << "Nama Pengguna: ";
                cin >> username;
                cout << "Password: ";
                cin >> password;

                for (const auto &user : pengguna) {
                    if (user.username == username && user.password ==
password) {
                        masuk = 1;
                        role = user.role;
                        break;
                    }
                }

                if (masuk) {
                    break;
                }
            }
        }
    }

```

```

        } else {
            cout << "Nama pengguna atau password salah. Coba lagi." <<
endl;

        }
    }

    if (!masuk) {
        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) {
                if (hitungPasien >= max_pasien) {
                    cout << "Daftar penuh." << endl;
                    continue;
                }

                cout << "Masukkan nama: ";
                cin.ignore();
                getline(cin, dataPasien[hitungPasien][0]);
                cout << "Masukkan umur: ";
                getline(cin, dataPasien[hitungPasien][1]);
                cout << "Masukkan diagnosis: ";
                getline(cin, dataPasien[hitungPasien][2]);
            }
        }
    }
}

```

```

        cout << "Masukkan riwayat obat sebelumnya: ";
        getline(cin, dataPasien[hitungPasien][3]);
        hitungPasien++;
    } else if (pilihan == 2) {
        cout << left << setw(20) << "Nama" << setw(5) <<
"Umur" << setw(20) << "Diagnosis" << setw(15) << "Riwayat obat" << endl;
        cout <<
"-----" << endl;

        for (int i = 0; i < hitungPasien; ++i) {
            cout << left << setw(20) << dataPasien[i][0] <<
setw(5) << dataPasien[i][1] << setw(20) << dataPasien[i][2] << setw(15) <<
dataPasien[i][3] << endl;
        }
    } else if (pilihan == 3) {
        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][0] == nama) {
                cout << "Masukkan umur baru: ";
                getline(cin, dataPasien[i][1]);
                cout << "Masukkan diagnosis baru: ";
                getline(cin, dataPasien[i][2]);
                cout << "Masukkan riwayat obat baru: ";
                getline(cin, dataPasien[i][3]);
                temukan = 1;
                break;
            }
        }

        if (!temukan) {
            cout << "Pasien tidak ditemukan." << endl;
        }
    } else if (pilihan == 4) {
        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][0] == nama) {
                for (int j = i; j < hitungPasien - 1; ++j) {
                    for (int k = 0; k < 4; ++k) {
                        dataPasien[j][k] = dataPasien[j +
1][k];
                    }
                }
                hitungPasien--;
                cout << "Nama Pasien berhasil dihapus." <<
endl;

                temukan = 1;
                break;
            }
        }

        if (!temukan) {
            cout << "Nama Pasien tidak ditemukan." << endl;
        }
    } else if (pilihan == 5) {
        cout << "Program berhenti." << endl;
        break;
    } else {
        cout << "Pilihan tidak valid. Mohon coba lagi." <<
endl;
    }
} else if (role == "user") {
    if (pilihan == 1) {
        cout << left << setw(20) << "Nama" << setw(5) <<
"Umur" << setw(20) << "Diagnosis" << setw(15) << "Riwayat obat" << endl;
        cout <<
"-----" << endl;

        for (int i = 0; i < hitungPasien; ++i) {
            cout << left << setw(20) << dataPasien[i][0] <<
setw(5) << dataPasien[i][1] << setw(20) << dataPasien[i][2] << setw(15) <<
dataPasien[i][3] << endl;
        }
    } 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;
    }
}

```

4.UJI COBA DAN HASIL OUTPUT

4.1 Uji Coba

A. Pengujian Menu utama

Input: Menampilkan tiga pilihan menu utama kepada pengguna untuk login sebagai admin atau user dan memilih keluar.

Output yang diinginkan: Nama pengguna, Password login. Kemudian program akan menjalankan fitur loginnya masing masing. Program akan berhenti jika memilih menu keluar

B. Pengujian Login Akun Admin

Akun admin berhasil masuk

Pada program yang dibuat, saya membuat dua akun khusus admin.

Input: Username = "Admin", Password = "12345"

Input: Username = "Najla", Password = "1701"

Output yang diinginkan : Menampilkan menu atau fitur fitur CRUD khusus yang hanya bisa diakses admin.

Pada bagian ini, jika menginput username dan password yang gagal sebanyak tiga kali, maka program akan berhenti. Username dan Password admin sudah ditentukan di dalam sebuah variabel array.

Input program akan berhenti jika

Input: Jika melakukan kesalahan dalam input username dan password pada akun pengguna admin sebanyak tiga kali

Output yang diinginkan: "Terlalu banyak percobaan login yang gagal. Keluar program."

Pengujian Login Akun User

Akun user berhasil masuk

Input: Username = "Nazla_Salsabila", Password = "2409106108"

Output yang diinginkan: Menampilkan tiga fitur pada program CRUD yaitu hanya bisa menambah data pasien, melihat tampilan data pasien yang ditambahkan, dan keluar dari program.

Input program akan berhenti jika

Input: Jika melakukan kesalahan dalam input username dan password pada akun pengguna user sebanyak tiga kali

Output yang diinginkan: "Terlalu banyak percobaan login yang gagal. Keluar program."

C. Pengujian Menu CRUD Pada Akun Admin

Memeriksa Create

Input: Pilihan 1, Tambah pasien

Output yang diinginkan: "Masukkan nama, Masukkan umur, Masukkan diagnosis, Masukkan riwayat obat sebelumnya"

Memeriksa Read

Input: Pilihan 2, Tampilkan data pasien

Output: Menampilkan data data pasien yang sudah ditambahkan

Memeriksa Update

Input: Pilihan 3, Mengupdate atau mengubah data pasien

Output yang diinginkan: "Masukkan nama yang akan di update"

Memeriksa Delete

Input: Pilihan menu 4, Menghapus pasien

Output yang diinginkan: "Masukkan nama yang akan dihapus".

D. Pengujian menu untuk keluar program

Input: Pilihan menu 5, keluar program

Output yang diinginkan: Kembali ke menu utama atau perulangan kembali, program tidak akan berhenti jika pengguna tidak memilih opsi keluar pada menu utamanya.

Pengujian input tidak valid

Input: Selain dari kelima menu yang ditampilkan dan memilih opsi lain, maka program akan terus berulang meminta input pilihan menu kembali

Output yang diinginkan: "Pilihan tidak valid. Mohon coba lagi"

E. Pengujian CRUD pada akun user

Memeriksa Read

Input: Pilihan 2, Tampilkan data pasien

Output: Menampilkan data data pasien

Keluar Program

Input: Pilihan menu 5, keluar program

Output yang diinginkan: Kembali ke menu utama

4.2 Hasil Output

A. Pengujian menu utama

```
011a-PT-2 } ; if ($?) { .\4
Menu Utama:
1. Login Sebagai Admin
2. Login Sebagai User
3. Keluar
Masukkan pilihan Anda: █
```

Gambar 4.2.1 Tampilan fitur utama

```
011a-PT-2 } ; if ($?) { .\4
Menu Utama:
1. Login Sebagai Admin
2. Login Sebagai User
3. Keluar
Masukkan pilihan Anda: 3
Program berhenti.
PS C:\praktikum-apl\post-test\post-test-2> █
```

Gambar 4.2.2 Pengujian program berhenti

B. Pengujian Login Akun Admin

```
Menu Utama:  
1. Login Sebagai Admin  
2. Login Sebagai User  
3. Keluar  
Masukkan pilihan Anda: 1  
Nama Pengguna: Admin  
Password: 12345  
  
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 4.2.3 Pengujian login akun admin berhasil

Pengujian Menu CRUD Pada Akun Admin

- 1) Memeriksa Menu Create. Disini admin mencoba menambahkan lima nama pasien

```
Masukkan pilihan Anda: 1  
Masukkan nama: Dina  
Masukkan umur: 20  
Masukkan diagnosis: Asam lambung  
Masukkan riwayat obat sebelumnya: Antasida
```

Gambar 4.2.4 Pengujian menu Create admin

```
Masukkan pilihan Anda: 1
Masukkan nama: Lala
Masukkan umur: 19
Masukkan diagnosis: Radang usus
Masukkan riwayat obat sebelumnya: Metronidazole
```

Gambar 4.2.5 Pengujian menu Create admin

```
Masukkan pilihan Anda: 1
Masukkan nama: Andini
Masukkan umur: 25
Masukkan diagnosis: Maag akut
Masukkan riwayat obat sebelumnya: Sucralfate
```

Gambar 4.2.6 Pengujian menu Create admin

```
Masukkan pilihan Anda: 1
Masukkan nama: Jaki Anwar
Masukkan umur: 22
Masukkan diagnosis: Asma
Masukkan riwayat obat sebelumnya: Ventolin inhaler
```

Gambar 4.2.7 Pengujian menu Create admin

```
Masukkan pilihan Anda: 1
Masukkan nama: Sarah sechan
Masukkan umur: 30
Masukkan diagnosis: Diare
Masukkan riwayat obat sebelumnya: Oralit imodium
```

Gambar 4.2.8 Pengujian menu Create admin

2) Memeriksa menu Read.

Masukkan pilihan Anda: 2

Nama	Umur	Diagnosis	Riwayat obat
Dina	20	Asam lambung	Antasida
Lala	19	Radang usus	Metronidazole
Andini	25	Maag akut	Sucralfate
Jaki Anwar	22	Asma	Ventolin inhaler
Sarah sechan	30	Diare	Oralit imodium

Gambar 4.2.9 Pengujian menu Read admin

3) Memeriksa menu Update

Masukkan pilihan Anda: 3
Masukkan nama yang akan diupdate: Lala
Masukkan umur baru: 19
Masukkan diagnosis baru: Diabetes melitus
Masukkan riwayat obat baru: Metformin

Gambar 4.2.10 Pengujian menu Update admin

Masukkan pilihan Anda: 2

Nama	Umur	Diagnosis	Riwayat obat
Dina	20	Asam lambung	Antasida
Lala	19	Diabetes melitus	Metformin
Andini	25	Maag akut	Sucralfate
Jaki Anwar	22	Asma	Ventolin inhaler
Sarah sechan	30	Diare	Oralit imodium

Gambar 4.2.11 Tampilan data setelah di update

4) Memeriksa menu Delete

```
Masukkan pilihan Anda: 4
Masukkan nama yang akan dihapus: Jaki Anwar
Nama Pasien berhasil dihapus.
```

Gambar 4.2.12 Pengujian menu Delete admin

5) Pengujian menu keluar program

```
Masukkan pilihan Anda: 5
Program berhenti.
Menu Utama:
1. Login Sebagai Admin
2. Login Sebagai User
3. Keluar
Masukkan pilihan Anda: 2
```

Gambar 4.2.13 Pengujian keluar program

C. Pengujian Input Tidak Valid

```
Silahkan pilih menu yang anda butuhkan:
1. Tambah Pasien
2. Tampilkan Data Pasien
3. Update Pasien
4. Hapus Pasien
5. Keluar
Masukkan pilihan Anda: 6
Pilihan tidak valid. Mohon coba lagi.
```

Gambar 4.2.12 Pengujian input tidak valid

D. Pengujian Menu CRUD Pada Akun User

1) Pengujian login akun user

```
1. Login Sebagai Admin
2. Login Sebagai User
3. Keluar
Masukkan pilihan Anda: 2
Nama Pengguna: Nazla_Salsabila
Password: 2409106108

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

Gambar 4.2.13 Pengujian login akun user

2) Pengujian menu user

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

Nama	Umur	Diagnosis	Riwayat obat
Dina	20	Asam lambung	Antasida
Lala	19	Diabetes melitus	Metformin
Andini	25	Maag akut	Sucralfate
Sarah sechan	30	Diare	Oralit imodium

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

Gambar 4.2.14 Pengujian menu user

3) Pengujian menu keluar pada user

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

Gambar 4.2.15 Logout akun user

5. PENJELASAN GIT

```
MINGW64/c/praktikum-apl
Nazla Salsabila@LAPTOP-A2AJ1HBK MINGW64 /c/praktikum-apl (master)
$ git pull origin master --rebase
error: cannot pull with rebase: Your index contains uncommitted changes.
error: Please commit or stash them.

Nazla Salsabila@LAPTOP-A2AJ1HBK MINGW64 /c/praktikum-apl (master)
$ git add .

Nazla Salsabila@LAPTOP-A2AJ1HBK MINGW64 /c/praktikum-apl (master)
$ git commit -m "posttest 2 finish"
[master e9e6789] posttest 2 finish
 9 files changed, 1163 insertions(+)
 create mode 100644 kelas/pertemuan-2.cpp
 create mode 100644 kelas/pertemuan-2/pertemuan-2.cpp
 create mode 100644 kelas/pertemuan-2/pertemuan-2.exe
 create mode 100644 post-test/post-test-1/. $flowchart-pt-2.drawio.bkp
 create mode 100644 post-test/post-test-1/flowchart-pt-2.drawio
 create mode 100644 post-test/post-test-2/2409106108-NazlaSalsabila-PT-2.cpp
 create mode 100644 post-test/post-test-2/2409106108-NazlaSalsabila-PT-2.exe
 create mode 100644 post-test/post-test-2/flowchart-pt-2.drawio
 create mode 100644 post-test/post-test-2/tempCodeRunnerFile.cpp

Nazla Salsabila@LAPTOP-A2AJ1HBK MINGW64 /c/praktikum-apl (master)
$ git push -u origin master
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 20 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (15/15), 946.81 KiB | 2.44 MiB/s, done.
Total 15 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/NazlaSalsabila108/praktikum-apl.git
   c34a186..e9e6789  master -> master
branch 'master' set up to track 'origin/master'.

Nazla Salsabila@LAPTOP-A2AJ1HBK MINGW64 /c/praktikum-apl (master)
$
```

Gambar 5 Tampilan terminal git bash

Caranya :

\$ git pull origin master --rebase

Perintah ini digunakan untuk memperbarui branch lokal dengan perubahan dari branch master di remote repository origin, dengan melakukan rebase. Rebase berarti mengaplikasikan commit-commit di branch lokal di atas commit-commit dari branch master yang baru saja diambil. Rebase ini menjaga riwayat commit agar lebih bersih.

\$ git add .

Perintah ini menambahkan semua file yang ada di dalam folder ke staging area, yaitu tempat sementara, sebelum file di komit ke dalam repository.

\$ git commit -m "posttest 2 finish"

Perintah ini menyimpan perubahan dalam repository dengan commit dan pesan tersebut.

\$ git push -u origin master

Perintah ini mengunggah (push) perubahan ke repository remote pada branch master, artinya perintah git push -u origin master akan mengirimkan semua commit yang ada di cabang master lokal ke cabang master di repository jarak jauh yang bernama origin, dan juga mengatur cabang lokal untuk melacak cabang jarak jauh tersebut.