

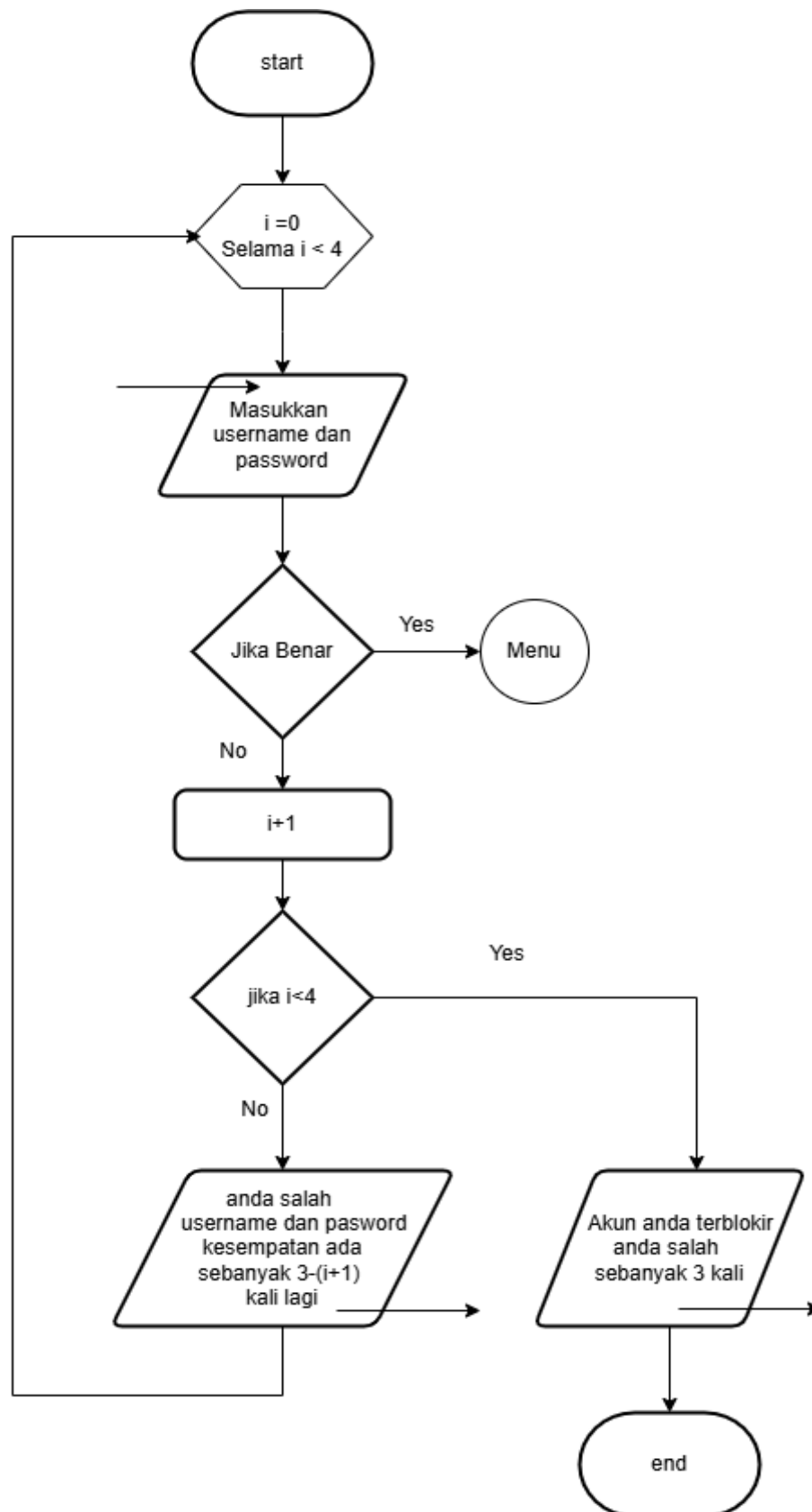
LAPORAN PRAKTIKUM
POSTTEST- 4
ALGORITMA PEMROGRAMAN LANJUT



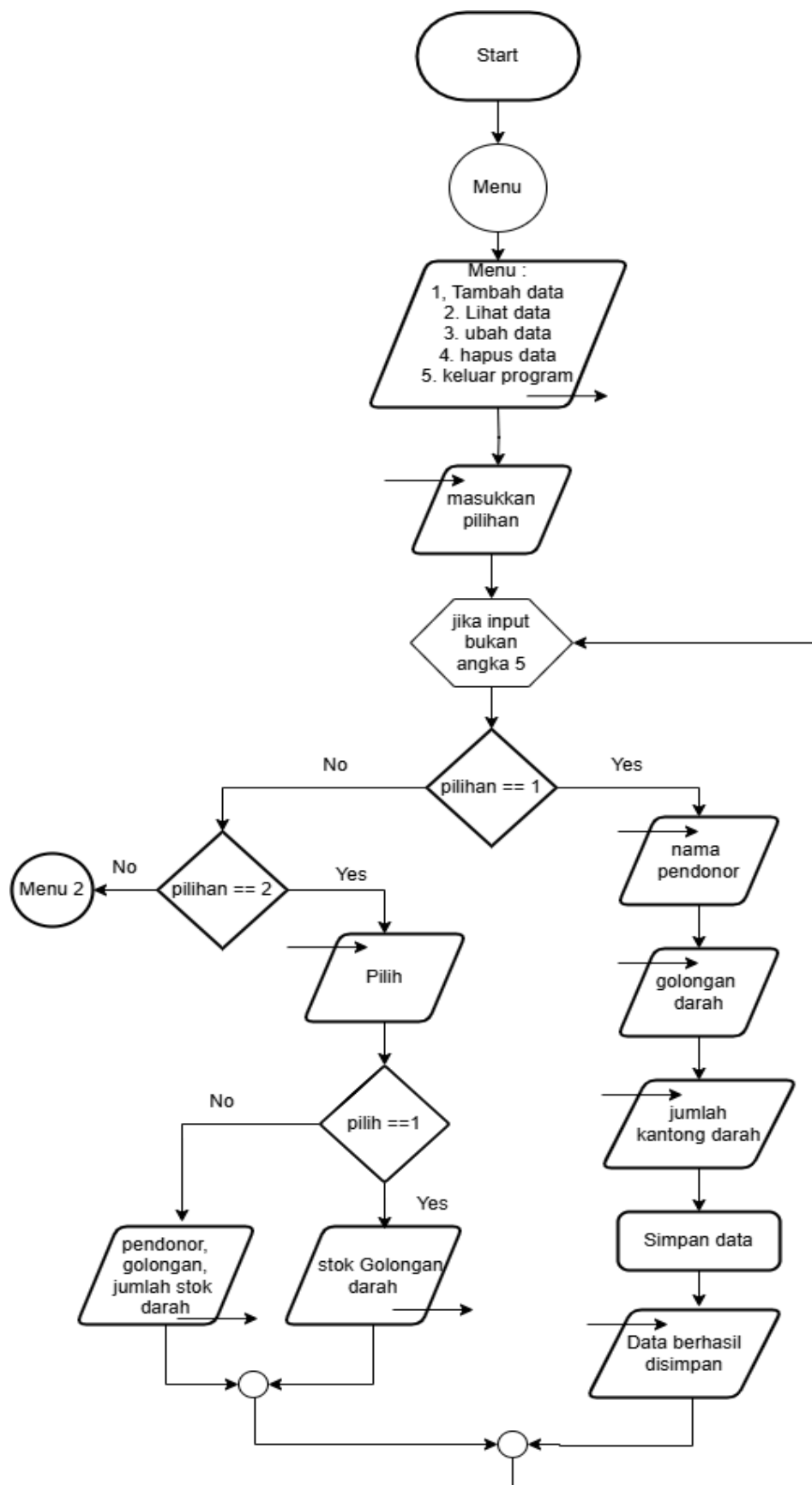
Disusun oleh:
SANTOSO PARLINDUNGAN TOGATOROP (2409106052)
Kelas B1 2024

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

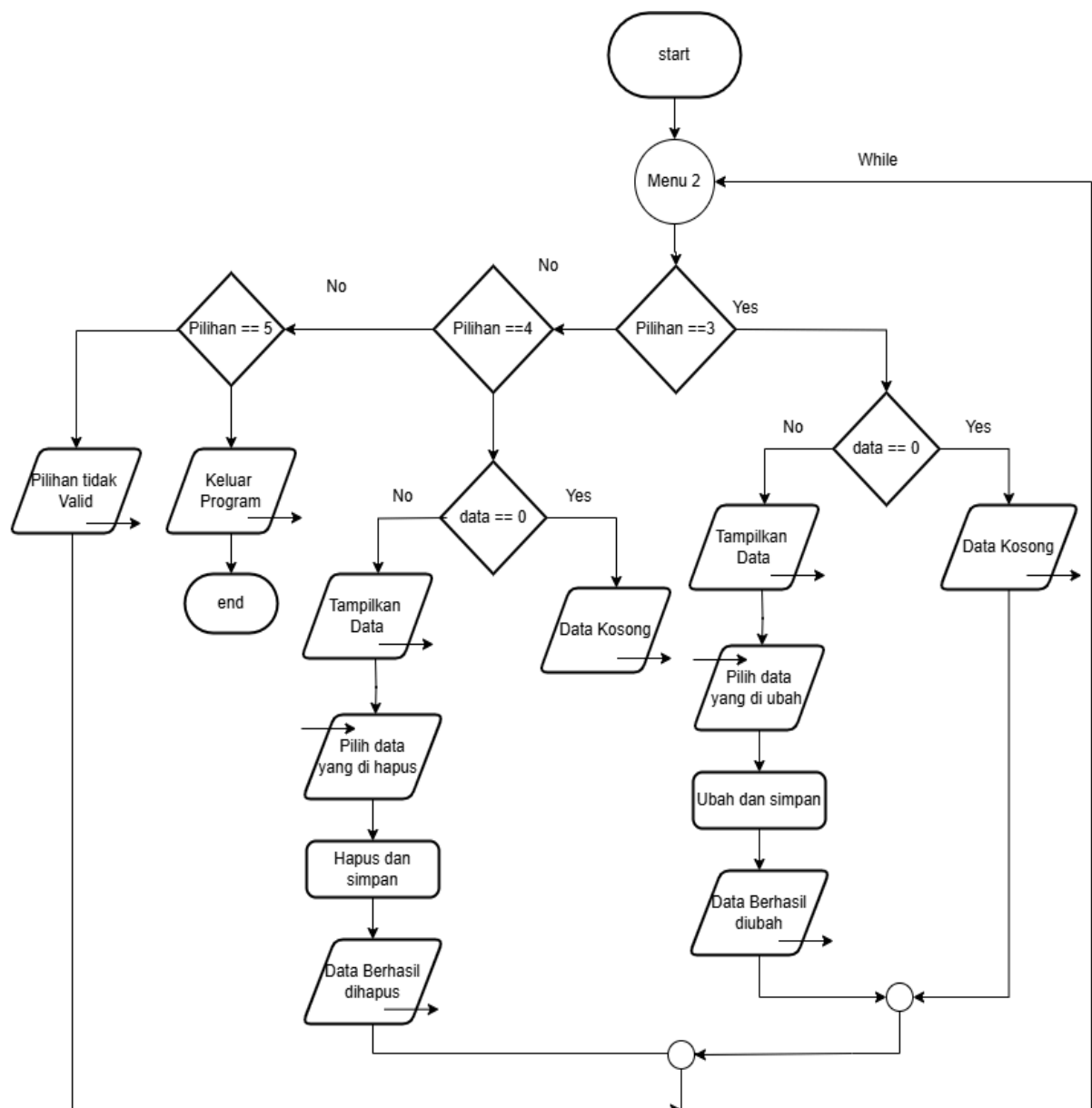
1. Flowchart



Gambar 1.1



Gambar 1.2



Gambar 1.3

2. Analisis Program

a. Deskripsi singkat program

Tujuan dari pembuatan program ini adalah memahami cara menjalankan pemrograman berbasis *CRUD* (*Create*, *Read*, *Update*, dan *Delete*) dalam sebuah data. Fungsinya untuk membuat program yang dapat dikelola oleh admin dan login menggunakan *username* dan *password*, dan dapat melakukan manajemen di dalam sistem. Sistem ini berisi manajemen stok kantong darah dan merekam data dari pendonor darah. Dalam program ini, user tidak diperbolehkan salah mengisi *username* dan *password* sebanyak 3 kali demi keamanan data.

Pada source code program ini akan menggunakan procedure dan parameter, sebagai syarat posttest-4 ini. Penggunaan procedure ini sendiri untuk mengurangi pembuatan blok kode yang berulang-ulang, contohnya pada fungsi hapus dan update data, program harus menampilkan data yang ada. Dengan menggunakan procedure ini, memungkinkan source code untuk menampilkan data hanya 1 blok kode saja.

3. Source Code

a. Fitur dan Fungsi Login

Fitur ini digunakan untuk memeriksa apakah *username* dan *password* sudah benar. Jika username dan password yang diinput user benar, user dapat masuk ke menu program dan mengelolanya.

```
for (int i = 0; i < 3; i++) {
    cout << "Masukkan Username: ";
    cin >> nama;
    cout << "Masukkan Password: ";
    cin >> sandi;
    system("cls||clear");

    if (nama==username && sandi==password) {
        //MENU//
    } else {
        cout << "Login gagal! Kesempatan tersisa " << 2 - i << " kali"
<<endl;
    }
}
```

b. Fitur Struct

Struct adalah fitur untuk menyimpan data pendonor, sehingga lebih memudahkan untuk mengelola data-data yang ada di dalamnya.

```
struct Alamat{
    string jalan;
    string kota;
};

struct Pendonor{
    string nama;
    string golongan;
    int stok;
    Alamat alamat;
};
```

b. Procedure Menampilkan Data

Procedure ini adalah kode yang akan menampilkan data pendonor stok kantong darah. Procedure ini akan dipanggil di beberapa fitur lainnya agar tidak perlu menulis ulang source code untuk menampilkan data lagi.

```
void tampilData(Pendonor donor[], int jumlah) {
    if (jumlah == 0){
        cout<<"____Data Kosong____"<<endl;
        return;
    }else{
        cout<<"No\t"<<"|\tNama\t\t|\tGolongan\t|Jumlah
darah\t|\tAlamat\t\t|\tKota\t\t|"<<endl;
        cout << "-----" <<
endl;
        for (int i = 0; i < jumlah; i++) {

            cout<<i+1<<"\t|\t"<<donor[i].nama<<"\t|\t"<<donor[i].golongan<<"\t\t|\t"<<donor[
i].stok<<"\t|\t"

            <<donor[i].alamat.jalan<<"\t|\t"<<donor[i].alamat.kota<<"\t|"<<endl;
                }
        }
    }
}
```

c. Procedure Tambah Data

Fitur ini adalah program untuk menambahkan data stok darah dari pendonor dan akan menyimpan jumlah stok darah yang tersedia. User akan mudah memanajemen data stok kantong darah yang ada jika program menyimpan data yang detail.

```
void tambahData() {
    if (jumlah >= MAX) {
        cout << "____Data Penuh____" << endl;
        return ;
    }

    cout << "Nama Pendonor: ";
    cin.ignore();
    getline(cin, donor[jumlah].nama);

    for (int i = 0; i < 4; i++) {
        cout << i + 1 << ". Golongan: " << jenis[i] << endl;
    }
    cout << "Golongan Darah: ";
    getline(cin, donor[jumlah].golongan);

    cout << "Jumlah stok darah: ";
    cin >> donor[jumlah].stok;
```

```

        if (donor[jumlah].golongan == "A") stokA += donor[jumlah].stok;
        else if (donor[jumlah].golongan == "B") stokB += donor[jumlah].stok;
        else if (donor[jumlah].golongan == "AB") stokAB += donor[jumlah].stok;
        else if (donor[jumlah].golongan == "O") stokO += donor[jumlah].stok;

        cin.ignore();
        cout << "Alamat: ";
        getline(cin, donor[jumlah].alamat.jalan);

        cout << "Kota: ";
        getline(cin, donor[jumlah].alamat.kota);

        jumlah++;
        system("cls||clear");
        cout << "Data berhasil ditambahkan!" << endl;
    }
}

```

d. Procedure Baca Data

Fitur ini menyediakan 2 pilihan bagi pengguna. User dapat hanya melihat stok kantong darah dan dapat melihat detail data pendonor, golongan, dan jumlah stok darah yang didonorkan. Jika user menginput pilihan program akan menampilkan sesuai pilihan.

```

void golongan(int stokA, int stokB, int stokAB, int stokO) {
    cout << "Golongan " << jenis[0] << " : " << stokA << endl;
    cout << "Golongan " << jenis[1] << " : " << stokB << endl;
    cout << "Golongan " << jenis[2] << " : " << stokAB << endl;
    cout << "Golongan " << jenis[3] << " : " << stokO << endl;
}

void tampilData(Pendonor donor[], int jumlah) {
    if (jumlah == 0){
        cout<<"___Data Kosong___"<<endl;
        return;
    }else{
        cout<<"No\t"<<"|\tNama\t\t|\tGolongan\t|Jumlah  
darah\t|\tAlamat\t\t|\tKota\t\t|"<<endl;
        cout << "-----" << endl;
        for (int i = 0; i < jumlah; i++) {
            cout<<i+1<<"\t|\t"<<donor[i].nama<<"\t|\t"<<donor[i].golongan<<"\t\t|\t"<<do  
nor[i].stok<<"\t|\t"  
<<donor[i].alamat.jalan<<"\t|\t"<<donor[i].alamat.kota<<"\t|"<<endl;
        }
    }
}
}

```


e. Procedure Ubah Data

Fitur ini adalah program untuk mengubah data yang sudah ada dan menginput kembali data yang diperlukan dalam perubahannya.

```
void ubahData() {
    tampilData(donor, jumlah);
    if (jumlah!=0){
        int pilih;
        cout << "Pilih data ke-berapa yang ingin diubah: ";
        cin >> pilih;
        pilih--;

        cin.ignore();
        cout << "Nama Baru: ";
        getline(cin, donor[pilih].nama);

        for (int i = 0; i < 4; i++) {
            cout << i + 1 << ". Golongan: " << jenis[i] << endl;
        }
        cout << "Golongan Baru: ";
        getline(cin, donor[pilih].golongan);

        cout << "Stok Baru: ";
        cin >> donor[pilih].stok;
        cin.ignore();

        cout << "Alamat Baru: ";
        getline(cin, donor[pilih].alamat.jalan);

        cout << "Kota Baru: ";
        getline(cin, donor[pilih].alamat.kota);

        system("cls||clear");
        cout << "Data berhasil diubah!" << endl;
    }
}
```

f. Fitur Hapus data

Fitur ini adalah fitur hapus data, Jika user memilih data pendonor yang akan dihapus, program akan menghapus data dan mengurangi jumlah stok darah sesuai jumlah stok yang didonorkan.

```
void hapusData() {
    tampilData(donor, jumlah);
    if(jumlah!=0){
        int pilih;
        cout << "Pilih data ke-berapa yang ingin dihapus: ";
        cin >> pilih;
        pilih--;

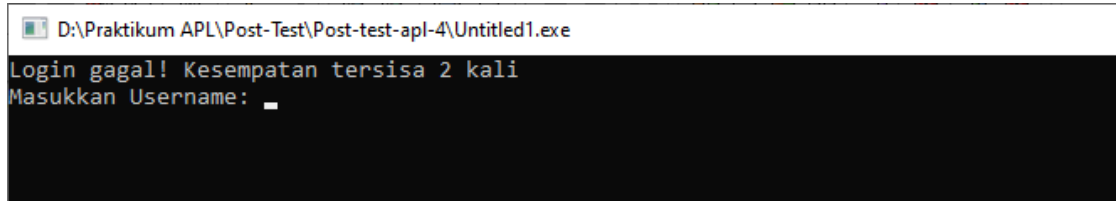
        if (donor[pilih].golongan == "A") stokA -= donor[pilih].stok;
        else if (donor[pilih].golongan == "B") stokB -= donor[pilih].stok;
        else if (donor[pilih].golongan == "AB") stokAB -= donor[pilih].stok;
        else if (donor[pilih].golongan == "O") stokO -= donor[pilih].stok;

        for (int i = pilih; i < jumlah - 1; i++) {
            donor[i] = donor[i + 1];
        }
        jumlah--;

        system("cls||clear");
        cout << "Data berhasil dihapus!" << endl;
    }
}
```

4. Hasil Output

a. Tampilan Salah Login



Gambar 4.1

b. Fitur Menu

```
D:\Praktikum APL\Post-Test\Post-test-apl-4\Untitled1.exe

=====
SISTEM PENGELOLAAN STOK KANTUNG DARAH
=====
1. Tambah Data
2. Lihat Data
3. Ubah Data
4. Hapus Data
5. Keluar
=====
Pilih:
```

Gambar 4.2

c. Fitur Tambah Data

```
D:\Praktikum APL\Post-Test\Post-test-apl-4\Untitled1.exe

Nama Pendonor: Hendry
1. Golongan: A
2. Golongan: B
3. Golongan: AB
4. Golongan: O
Golongan Darah: A
Jumlah stok darah: 20
Alamat: Perjuangan
Kota: Samarinda
```

Gambar 4.3

d. Fitur Lihat data

```
D:\Praktikum APL\Post-Test\Post-test-apl-4\Untitled1.exe

1. Lihat Jumlah Stok Darah
2. Detail Pendonor
Pilih: 1
Golongan A : 20
Golongan B : 30
Golongan AB : 0
Golongan O : 0
```

Gambar 4.4

```
D:\Praktikum APL\Post-Test\Post-test-apl-4\Untitled1.exe

1. Lihat Jumlah Stok Darah
2. Detail Pendonor
Pilih: 2
No | Nama | Golongan | Jumlah darah | Alamat | Kota
-----|-----|-----|-----|-----|-----
1 | Hendry | B | 30 | Bengalon | Samarinda
2 | Richard | AB | 40 | Gelatiks | Samarinda
3 | Santoso | B | 60 | Sambaliung | Samarinda
Press any key to continue . . .
```

Gambar 4.5

e. Fitur Ubah data

```
D:\Praktikum APL\Post-Test\Post-test-apl-4\Untitled1.exe
No | Nama | Golongan | Jumlah darah | Alamat | Kota
-----|-----|-----|-----|-----|-----
1 | Hendry | B | 30 | Bengalon | Samarinda
2 | Richard | AB | 40 | Gelatiks | Samarinda
3 | santoso | B | 60 | sambaliung | samarinda
Press any key to continue . . .
Pilih data ke-berapa yang ingin diubah: 3
Nama Baru: Septial
1. Golongan: A
2. Golongan: B
3. Golongan: AB
4. Golongan: O
Golongan Baru: B
Stok Baru: 60
Alamat Baru: Sambaliung
Kota Baru: Samarinda
```

Gambar 4.6

```
D:\Praktikum APL\Post-Test\Post-test-apl-4\Untitled1.exe
1. Lihat Jumlah Stok Darah
2. Detail Pendonor
Pilih: 2
No | Nama | Golongan | Jumlah darah | Alamat | Kota
-----|-----|-----|-----|-----|-----
1 | Hendry | B | 30 | Bengalon | Samarinda
2 | Richard | AB | 40 | Gelatiks | Samarinda
3 | Septial | B | 60 | Sambaliung | Samarinda
Press any key to continue . . .
```

Gambar 4.7

f. Fitur Hapus Data

```
D:\Praktikum APL\Post-Test\Post-test-apl-4\Untitled1.exe
No | Nama | Golongan | Jumlah darah | Alamat | Kota
-----|-----|-----|-----|-----|-----
1 | Richard | AB | 40 | Gelatiks | Samarinda
2 | Septial | B | 60 | Sambaliung | Samarinda
Press any key to continue . . .
Pilih data ke-berapa yang ingin dihapus: 2
```

Gambar 4.8

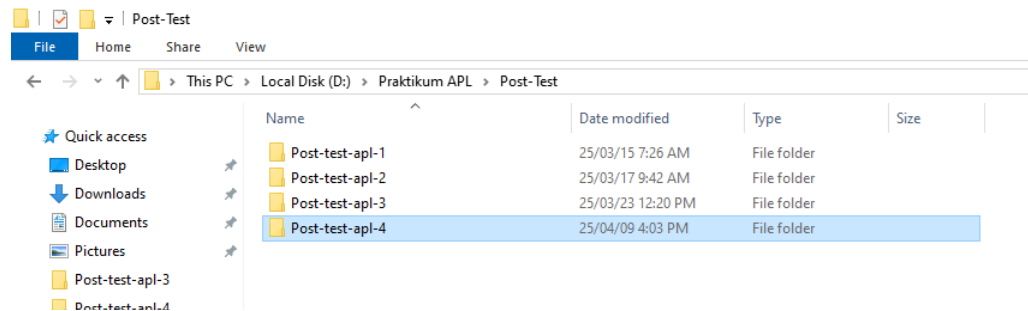
```
D:\Praktikum APL\Post-Test\Post-test-apl-4\Untitled1.exe
1. Lihat Jumlah Stok Darah
2. Detail Pendonor
Pilih: 2
No | Nama | Golongan | Jumlah darah | Alamat | Kota
-----|-----|-----|-----|-----|-----
1 | Richard | AB | 40 | Gelatiks | Samarinda
Press any key to continue . . .
```

Gambar 4.9

5. Langkah-langkah membuat Git

a. Buka Folder lokal yang akan dipindahkan

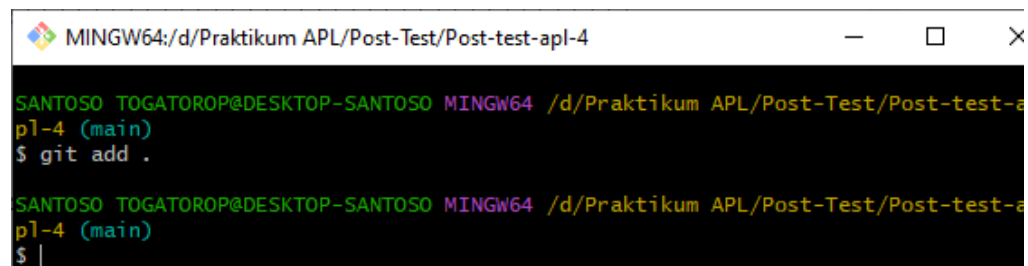
Buat folder lokal yang akan dipindahkan ke repository, tambahkan file baru yang akan disimpan ke git. lalu klik kanan pada folder yang akan dipindahkan dan pilih “Open git bash Here”, untuk membuka terminal pada git.



gambar 5.4

b. Menambahkan folder

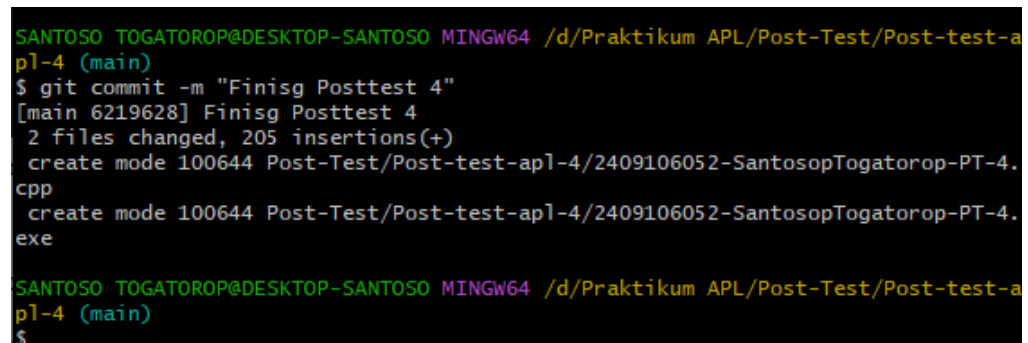
Buka file yang akan disimpan, ketik perintah “**git add .**”. Perintah ini berfungsi untuk menambahkan folder secara keseluruhan, bukan per folder lagi. Sehingga file yang baru disimpan di file local akan tersimpan di git.



Gambar 5.7

c. Commit folder

Memasukkan file yang ada pada folder lokal komputer ke dalam folder git dengan perintah “**git commit -m “nama commit (bebas)”**”.



Gambar 5.9

d. Push folder

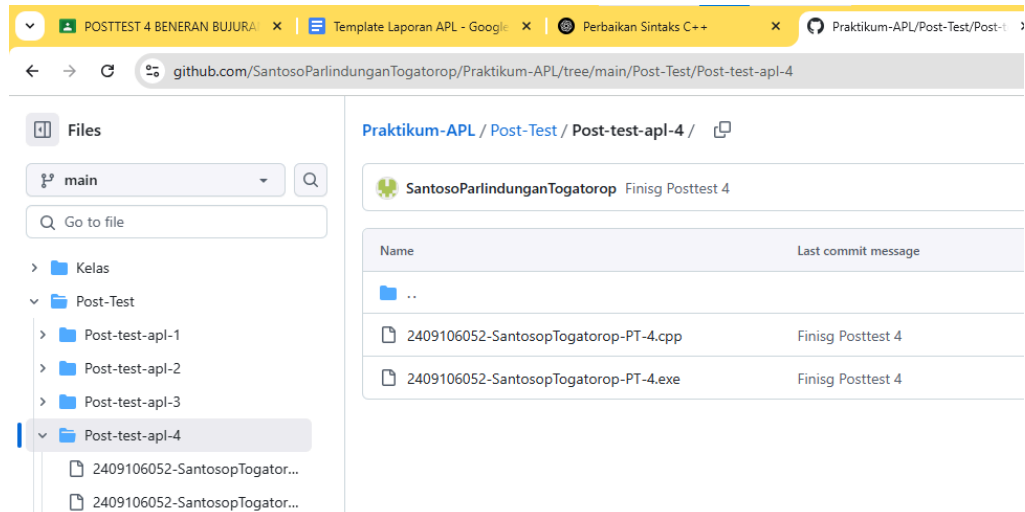
Jika commit untuk memasukkan folder, maka pada push ini kita memindahkan file folder lokal komputer ke dalam git dan ini merupakan langkah terakhir dalam pemindahan file lokal ke dalam git. Ketik perintah “git push -u origin main” pada terminal git bash.

```
SANTOSO TOGATOROP@DESKTOP-SANTOSO MINGW64 /d/Praktikum APL/Post-Test/Post-test-  
p1-4 (main)  
$ git push -u origin main  
Enumerating objects: 8, done.  
Counting objects: 100% (8/8), done.  
Delta compression using up to 2 threads  
Compressing objects: 100% (6/6), done.  
Writing objects: 100% (6/6), 450.48 KiB | 2.07 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/SantosoParlindunganTogatorop/Praktikum-APL.git  
f6d6dab..6219628 main -> main  
branch 'main' set up to track 'origin/main'.
```

Gambar 5.10

e. Folder github

Setelah semua tahapan dijalankan dengan benar maka file dari lokal komputer akan dipindahkan ke dalam folder github. namun disini nam commit akan terganti secara keseluruhan karena memindahkan file secara keseluruhan bukan per file.



Gambar 5.11