

LAPORAN PRAKTIKUM

POSTTEST- 6

ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:

SANTOSO PARLINDUNGAN TOGATOROP (2409106052)

Kelas B1 2024

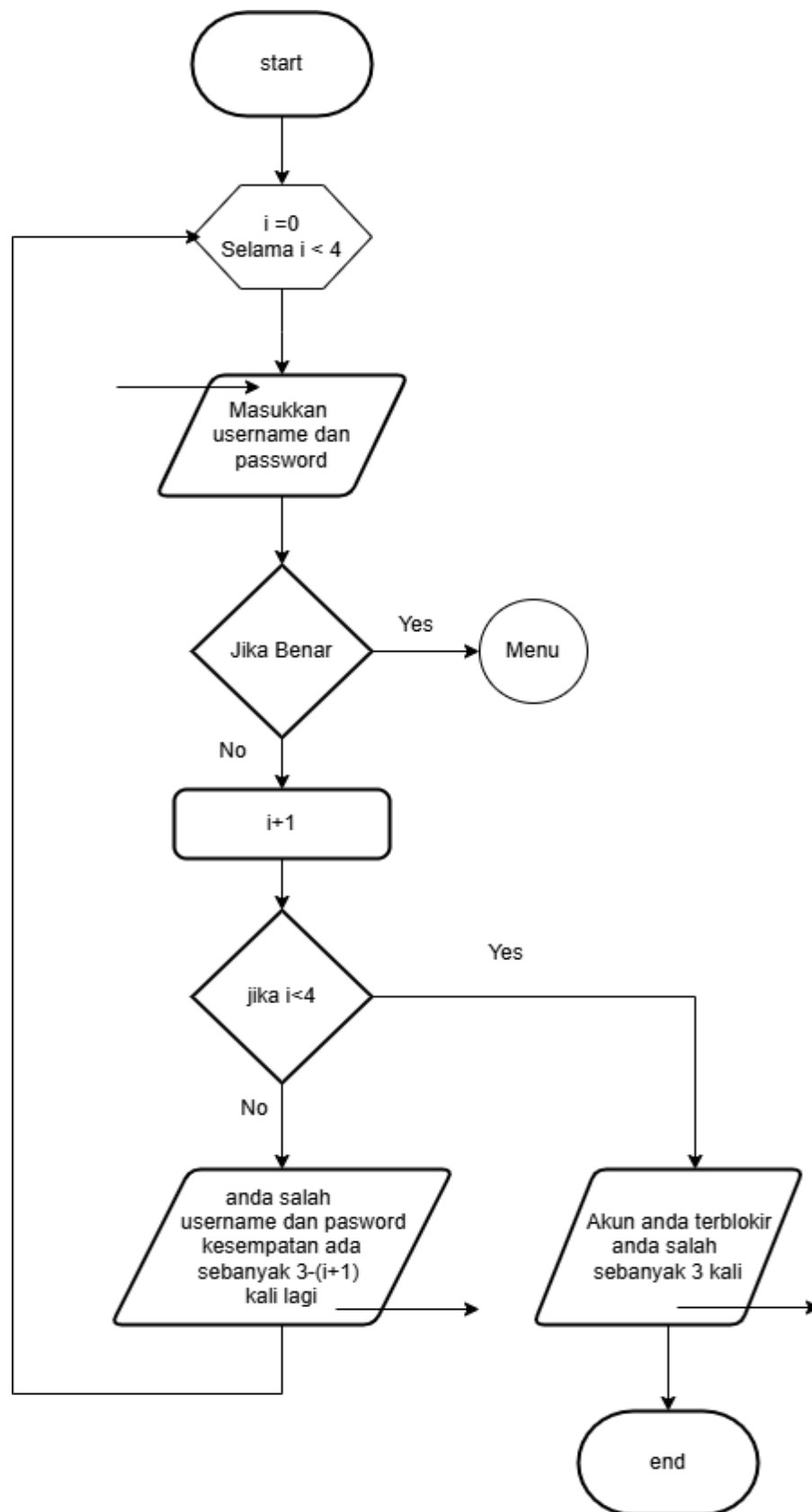
PROGRAM STUDI INFORMATIKA

UNIVERSITAS MULAWARMAN

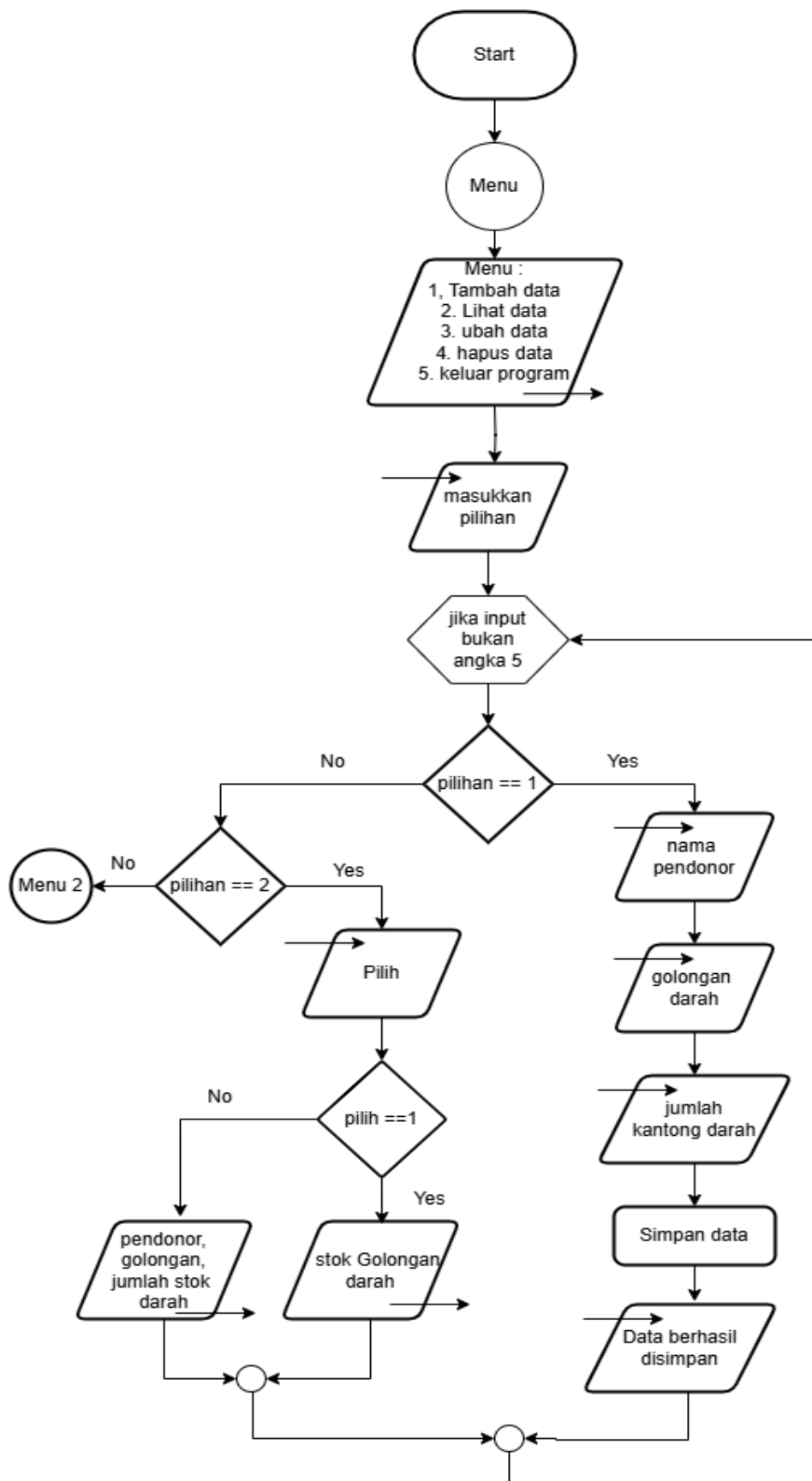
SAMARINDA

2025

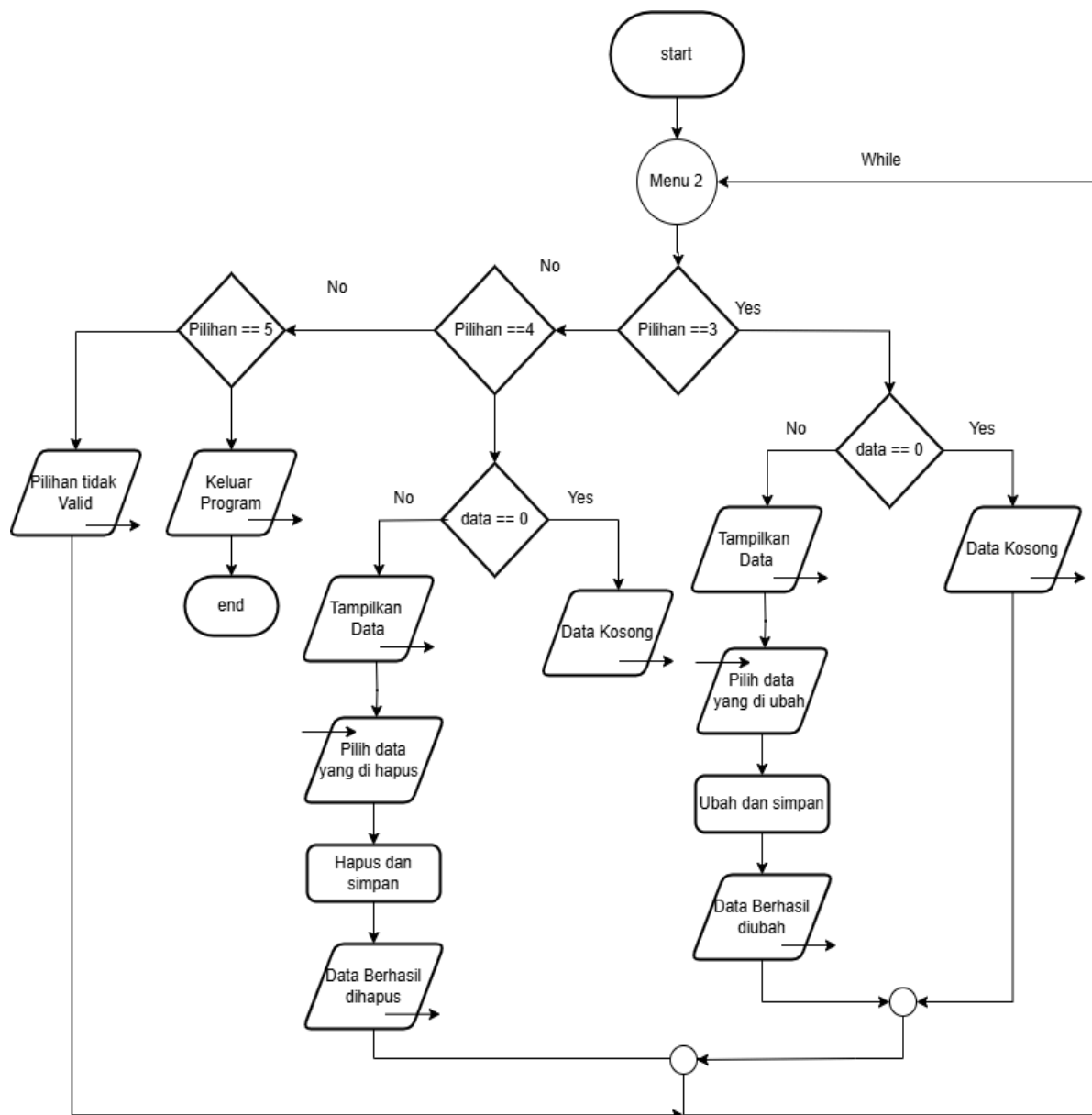
1. Flowchart



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 sebelumnya. 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.

Pada Program ini menggunakan pointer, dengan menggunakan Pointer ini akan mengurangi penggunaan memori saat mengubah data yang sama tanpa duplicate. Dalam program ini juga menggunakan fungsi dengan parameter address-of dan parameter deference.

Program ini Menggunakan Metode Sorting dalam tampilan datanya. Tampilan data lengkap pendonor akan diurutkan (Sort) dari yang terbesar ke yang terkecil Z-A (Descending), sedangkan tampilan jumlah stok darah akan diurutkan dari yang terkecil ke yang terbesar A-Z (Ascending), dan yang terakhir sorting opsional, pada program ini menambah satu fitur untuk mengurutkan data lengkap pendonor namun dari yang terkecil ke yang terbesar (Ascending).

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

b. Fitur Struct dan Pointer

Struct adalah fitur untuk menyimpan data pendonor, sehingga lebih memudahkan untuk mengelola data-data yang ada di dalamnya. sedangkan “Pendonor data[MAX]” adalah array of struct dan “Pendonor*donor[MAX}” adalah Pointer yang digunakan untuk mengakses dan menyimpan elemen ke array sebelumnya.

```
struct Alamat {  
    string jalan;  
    string kota;  
};  
  
struct Pendonor {  
    string nama;  
    string golongan;  
    int stok;  
    Alamat alamat;  
};  
  
#define MAX 100  
  
string jenis[4] = {"A", "B", "AB", "O"};  
Pendonor data[MAX];  
Pendonor* donor[MAX]; // Array of pointer  
int stok[4];
```

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. Dalam procedure ini diterapkan metode sorting yaitu **Bubble Sort**, supaya data yang ditampilkan oleh program akan diurutkan dari Z-A (Descending).

```
void tampilData(Pendonor* donor[], int jumlah) {
    if (jumlah == 0) {
        cout << "____Data Kosong____" << endl;
        return;
    }

    //Mengurutkan Nama Pendonor secara descending (Z-A)
    // Bubble Sort

    for (int i = 0; i < jumlah; i++) {
        for (int j = 0; j < (jumlah - i - 1); j++) {
            if (donor[j]->nama < donor[j+1]->nama) {
                Pendonor* temp = donor[j];
                donor[j] = donor[j+1];
                donor[j+1] = temp;
            }
        }
    }

    cout << "No\t|\tNama\t|\tGolongan\t|Jumlah\t|\tAlamat\t|\tKota\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|"
        << donor[i]->stok << "\t|\t" << donor[i]->alamat.jalan << "\t|\t"
        << donor[i]->alamat.kota << "\t|" << endl;
    }
    system("pause");
}
```

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(Pendonor* donor[], int&jumlah) {
    if (jumlah >= MAX) {
        cout << "____Data Penuh____" << endl;
        return;
    }

    donor[jumlah] = &data[jumlah]; // Hubungkan pointer dengan struct asli

    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") stok[0] += donor[jumlah]->stok;
else if (donor[jumlah]->golongan == "B") stok[1] += donor[jumlah]->stok;
else if (donor[jumlah]->golongan == "AB") stok[2] += donor[jumlah]->stok;
else if (donor[jumlah]->golongan == "O") stok[3] += 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. Fungsi Tampilan Jumlah Stok

Fitur ini untuk Menampilkan jumlah stok kantong darah yang masih tersedia, dalam fitur ini menggunakan metode sorting yaitu **Selection Sort** untuk mengurutkan data dari A-Z (Ascending).

```

int golongan() {

    //Mengurutkan Jumlah stok secara Ascending (A-Z)
    //Selection Sort

    for (int i = 0; i < 4; i++) {
        int min = i;
        for (int j = i + 1; j < 4; j++) {
            if (stok[j] < stok[min]) {
                min = j;
            }
        }
        int tempStok = stok[i];
        stok[i] = stok[min];
        stok[min] = tempStok;

        string tempJenis = jenis[i];
        jenis[i] = jenis[min];
        jenis[min] = tempJenis;
    }

    // Menampilkan hasil sorting
    cout << "Stok Darah Berdasarkan Golongan (Ascending):" << endl;
}

```



```
for (int i = 0; i < 4; i++) {  
    cout << "Golongan " << jenis[i] << " : " << stok[i] << endl;  
}  
  
return stok[0] + stok[1] + stok[2] + stok[3];  
}
```

e. Procedure Ubah Data

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

```
void ubahData(Pendonor* donor[], int *) {
    tampilData(donor, *jumlah);
    if (jumlah == 0) return;

    int pilih;
    cout << "Pilih data ke-berapa yang ingin diubah: ";
    cin >> pilih;
    pilih--;

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

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

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

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

    cout << "Kota : ";
    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(Pendonor* donor[], int& jumlah) {
    tampilData(donor, jumlah);
    if (jumlah == 0) return;

    int pilih;
    cout << "Pilih data ke-berapa yang ingin dihapus: ";
    cin >> pilih;
    pilih--;

    if (donor[pilih]->golongan == "A") stok[0] -= donor[pilih]->stok;
    else if (donor[pilih]->golongan == "B") stok[1] -= donor[pilih]->stok;
    else if (donor[pilih]->golongan == "AB") stok[2] -= donor[pilih]->stok;
    else if (donor[pilih]->golongan == "O") stok[3] -= 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;
}
```

g. Fitur Insertion Sorting

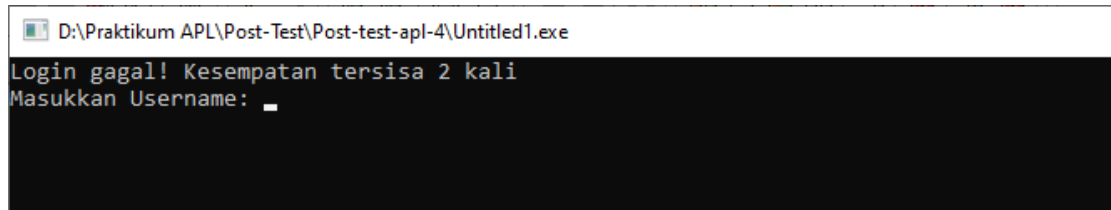
Fitur ini adalah fitur tambahan untuk mengurutkan data berdasarkan nama pendonor dari A-Z (Ascending).

```
void sortdata(Pendonor* donor[], int panjang) {
    for (int i = 1; i < panjang; i++) {
        Pendonor* key = donor[i];
        int j = i - 1;
        while (j >= 0 && donor[j]->nama > key->nama) {
            donor[j + 1] = donor[j];
            j = j - 1;
        }
        donor[j + 1] = key;
    }

    cout << "No\t|\tNama\t|\tGolongan\t|\tJumlah\t|\tAlamat\t|\t\t\t|\tKota\t|\t\t|" << endl;
    cout << "-----" << endl;
    for (int i = 0; i < panjang; i++) {
        cout << i + 1 << "\t|\t" << donor[i]->nama << "\t|\t" << donor[i]->golongan << "\t\t|"
            << donor[i]->stok << "\t|\t" << donor[i]->alamat.jalan << "\t|\t"
            << donor[i]->alamat.kota << "\t|" << endl;
    }
    system("pause");
}
```

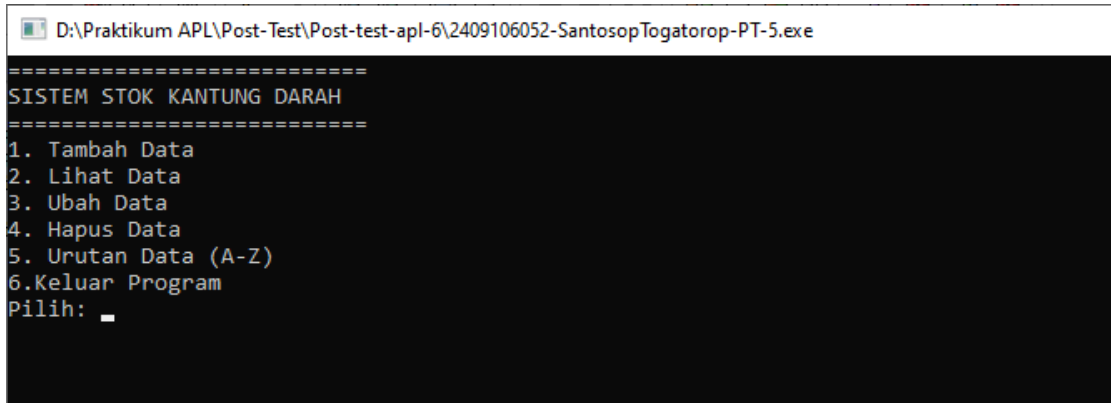
4. Hasil Output

a. Tampilan Salah Login



Gambar 4.1

b. Fitur Menu

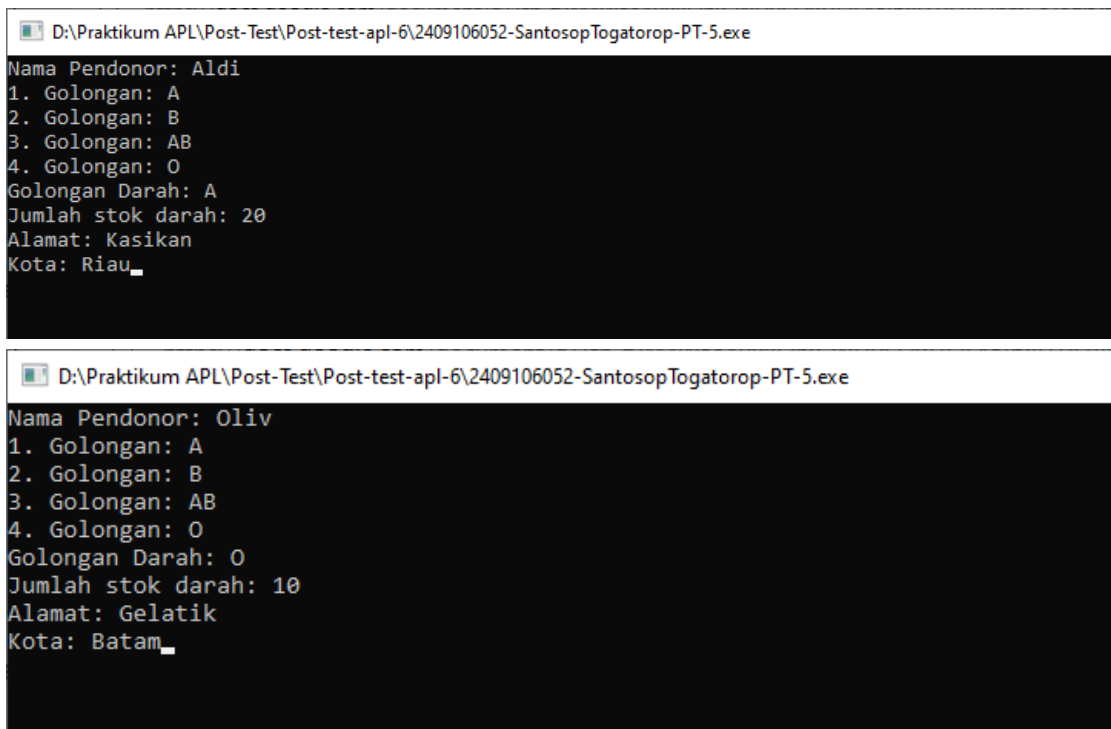


```
D:\Praktikum APL\Post-Test\Post-test-apl-6\2409106052-SantosopTogatorop-PT-5.exe

=====
SISTEM STOK KANTUNG DARAH
=====
1. Tambah Data
2. Lihat Data
3. Ubah Data
4. Hapus Data
5. Urutan Data (A-Z)
6. Keluar Program
Pilih: _
```

Gambar 4.2

c. Fitur Tambah Data



```
D:\Praktikum APL\Post-Test\Post-test-apl-6\2409106052-SantosopTogatorop-PT-5.exe

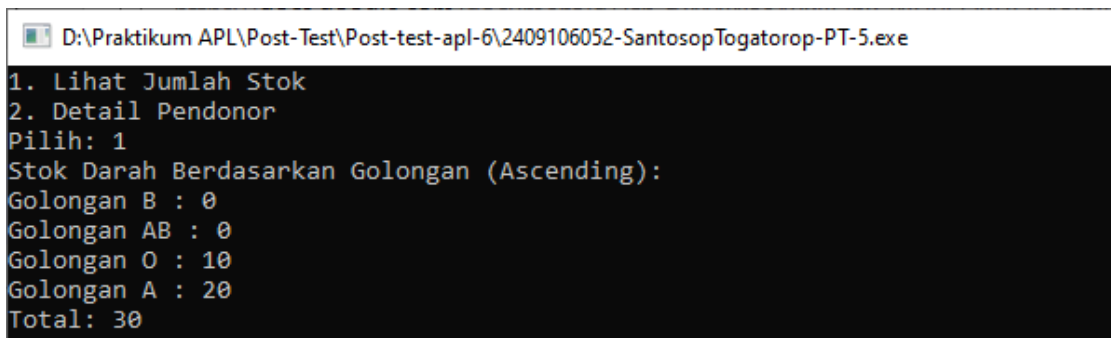
Nama Pendoror: Aldi
1. Golongan: A
2. Golongan: B
3. Golongan: AB
4. Golongan: O
Golongan Darah: A
Jumlah stok darah: 20
Alamat: Kasikan
Kota: Riau_

D:\Praktikum APL\Post-Test\Post-test-apl-6\2409106052-SantosopTogatorop-PT-5.exe

Nama Pendoror: Oliv
1. Golongan: A
2. Golongan: B
3. Golongan: AB
4. Golongan: O
Golongan Darah: O
Jumlah stok darah: 10
Alamat: Gelatik
Kota: Batam_
```

Gambar 4.3

d. Fitur Lihat data



```
D:\Praktikum APL\Post-Test\Post-test-apl-6\2409106052-SantosopTogatorop-PT-5.exe

1. Lihat Jumlah Stok
2. Detail Pendoror
Pilih: 1
Stok Darah Berdasarkan Golongan (Ascending):
Golongan B : 0
Golongan AB : 0
Golongan O : 10
Golongan A : 20
Total: 30
```

Gambar 4.4

```
D:\Praktikum APL\Post-Test\Post-test-apl-6\2409106052-SantosopTogatorop-PT-5.exe
1. Lihat Jumlah Stok
2. Detail Pendoror
Pilih: 2
No      |      Nama      |      Golongan      |Jumlah |      Alamat      |      Kota      |
-----|-----|-----|-----|-----|-----|
1      |      Oliv      |      O      |10     |      Gelatik      |      Batam      |
2      |      Aldi      |      A      |20     |      Kasikan      |      Riau      |
Press any key to continue . . .
```

Gambar 4.5

e. Fitur Ubah data

```
D:\Praktikum APL\Post-Test\Post-test-apl-6\2409106052-SantosopTogatorop-PT-5.exe
No      |      Nama      |      Golongan      | Jumlah |      Alamat      |      Kota      |
-----|-----|-----|-----|-----|-----|
1      |      Oliv      |      O              | 10     |      Gelatik     |      Batam     |
2      |      Aldi      |      A              | 20     |      Kasikan     |      Riau      |
Press any key to continue . . .
Pilih data ke-berapa yang ingin diubah:
1
Nama Baru: Ucok
1. Golongan: B
2. Golongan: AB
3. Golongan: O
4. Golongan: A
Golongan Baru: AB
Stok Baru: 40
Alamat Baru: Gelatik
Kota Baru: Batam
```

Gambar 4.6

```
D:\Praktikum APL\Post-Test\Post-test-apl-6\2409106052-SantosopTogatorop-PT-5.exe
No      |      Nama      |      Golongan      | Jumlah |      Alamat      |      Kota      |
-----|-----|-----|-----|-----|-----|
1      |      Ucok      |      AB             | 40     |      Gelatik     |      Batam     |
2      |      Aldi      |      A              | 20     |      Kasikan     |      Riau      |
Press any key to continue . . .
```

Gambar 4.7

f. Fitur Hapus Data

```
D:\Praktikum APL\Post-Test\Post-test-apl-6\2409106052-SantosopTogatorop-PT-5.exe
No      |      Nama      |      Golongan      | Jumlah |      Alamat      |      Kota      |
-----|-----|-----|-----|-----|-----|
1      |      Ucok      |      AB             | 40     |      Gelatik     |      Batam     |
2      |      Aldi      |      A              | 20     |      Kasikan     |      Riau      |
Press any key to continue . . .
Pilih data ke-berapa yang ingin dihapus: 2
```

Gambar 4.8

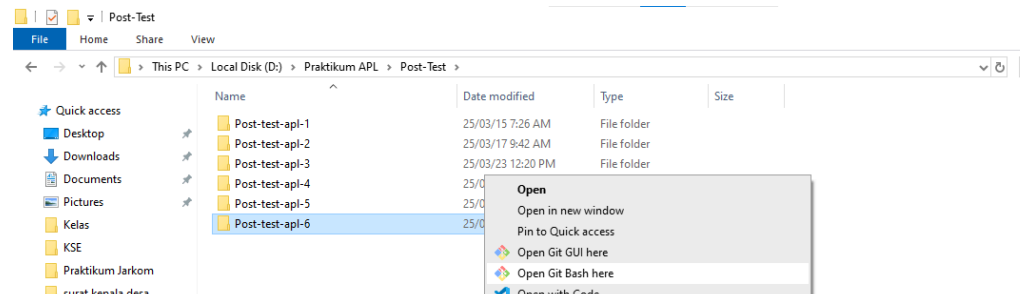
```
D:\Praktikum APL\Post-Test\Post-test-apl-6\2409106052-SantosopTogatorop-PT-5.exe
1. Lihat Jumlah Stok
2. Detail Pendoron
Pilih: 2
No      |      Nama      |      Golongan      | Jumlah |      Alamat      |      Kota      |
-----|-----|-----|-----|-----|-----|
1      |      Ucok      |      AB             | 40     |      Gelatik     |      Batam     |
Press any key to continue . . .
```

Gambar 4.9

5. Langkah-langkah membuat Git

a. Buka Folder lokal yang akan dipindahkan

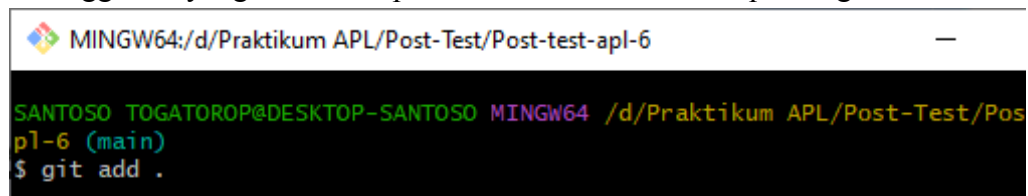
Buat folder lokal yang akan dipindahkan ke repository, tambahkan file baru yang akan disimpan ke git. lalu klik kanan pada folder yang akan dipindahkan dan pilih “Open git bash Here”, untuk membuka terminal pada git.



gambar 5.4

b. Menambahkan folder

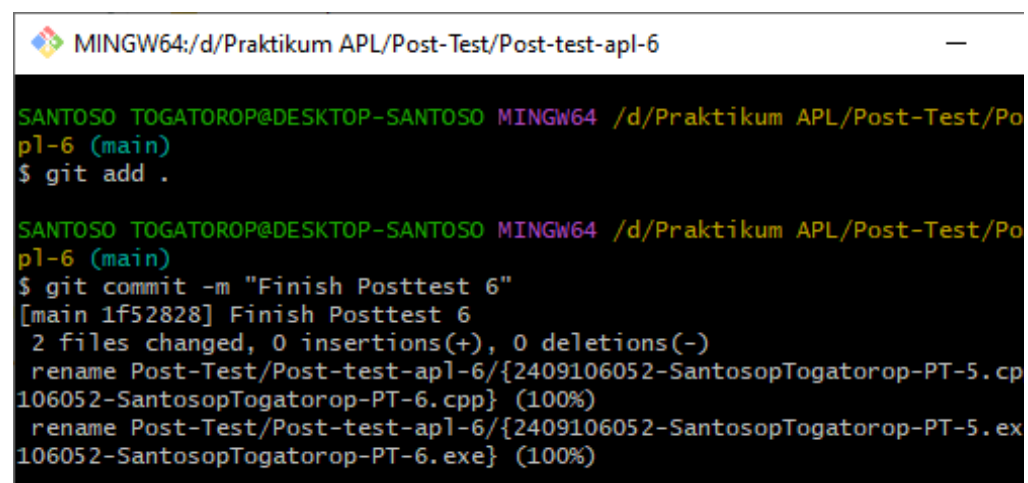
Buka file yang akan disimpan, ketik perintah “**git add .**”. Perintah ini berfungsi untuk menambahkan folder secara keseluruhan, bukan per folder lagi. Sehingga file yang baru disimpan di file local akan tersimpan di git.



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

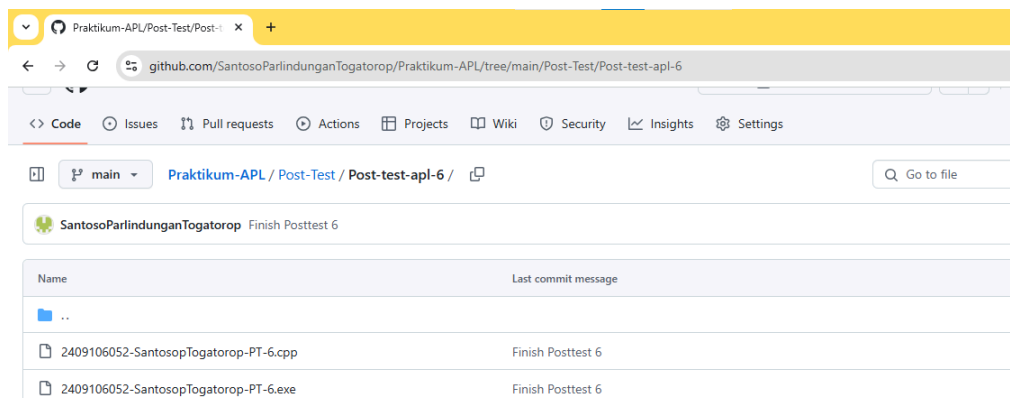
Jika commit untuk memasukkan folder, maka pada push ini kita memindahkan file folder lokal komputer ke dalam git dan ini merupakan langkah terakhir dalam pemindahan file lokal ke dalam git. Ketik perintah “git push -u origin main” pada terminal git bash.

```
SANTOSO TOGATOROP@DESKTOP-SANTOSO MINGW64 /d/Praktikum APL/Post-Test/Post-test-apl-6 (main)
$ git push -u origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 2 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 451.37 KiB | 2.59 MiB/s, done.
Total 10 (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/SantosoParlindunganTogatorop/Praktikum-APL.git
 27b95b8..1f52828 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