

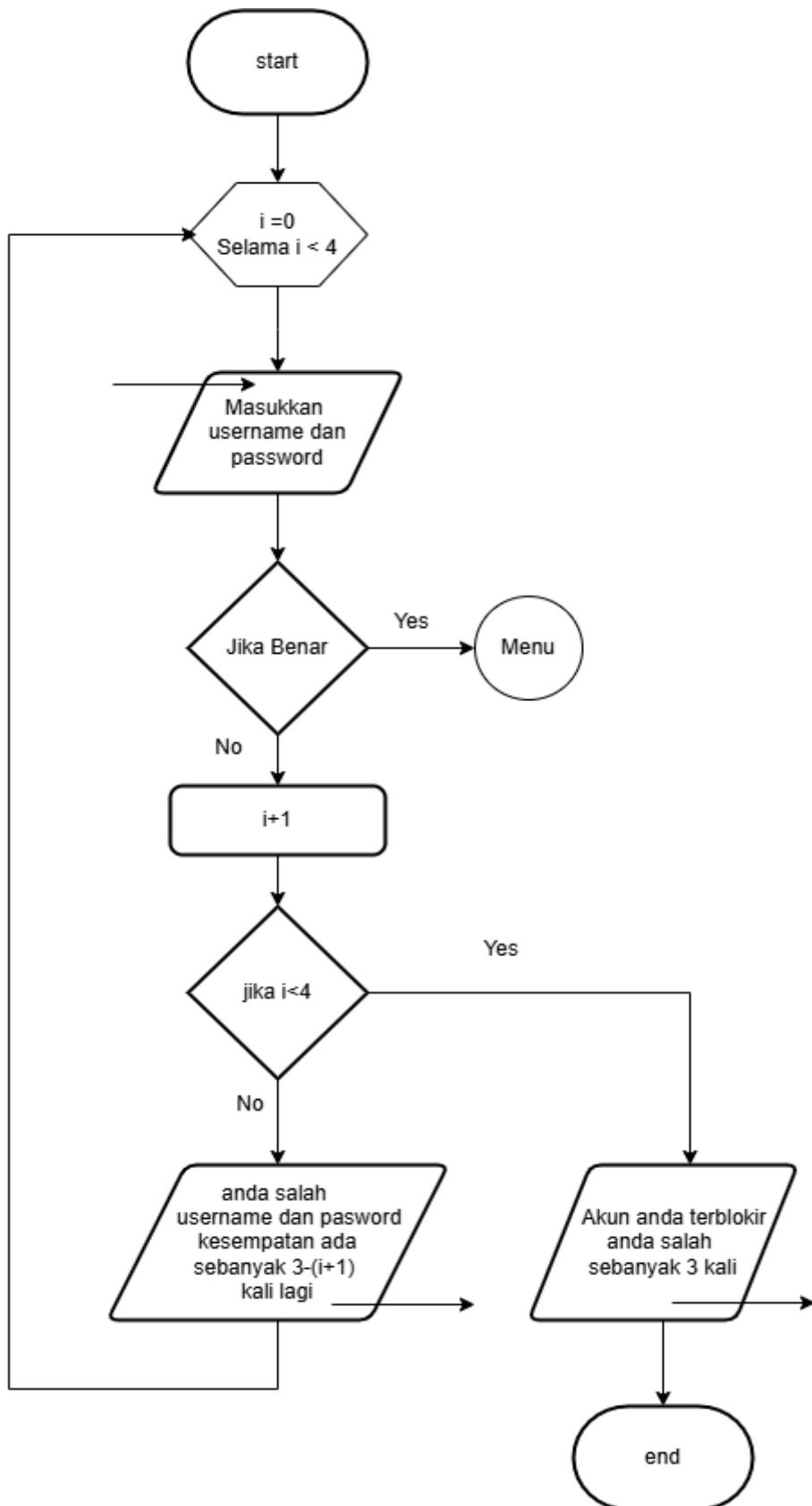
**LAPORAN PRAKTIKUM**  
**POSTTEST- 2**  
**ALGORITMA PEMROGRAMAN LANJUT**



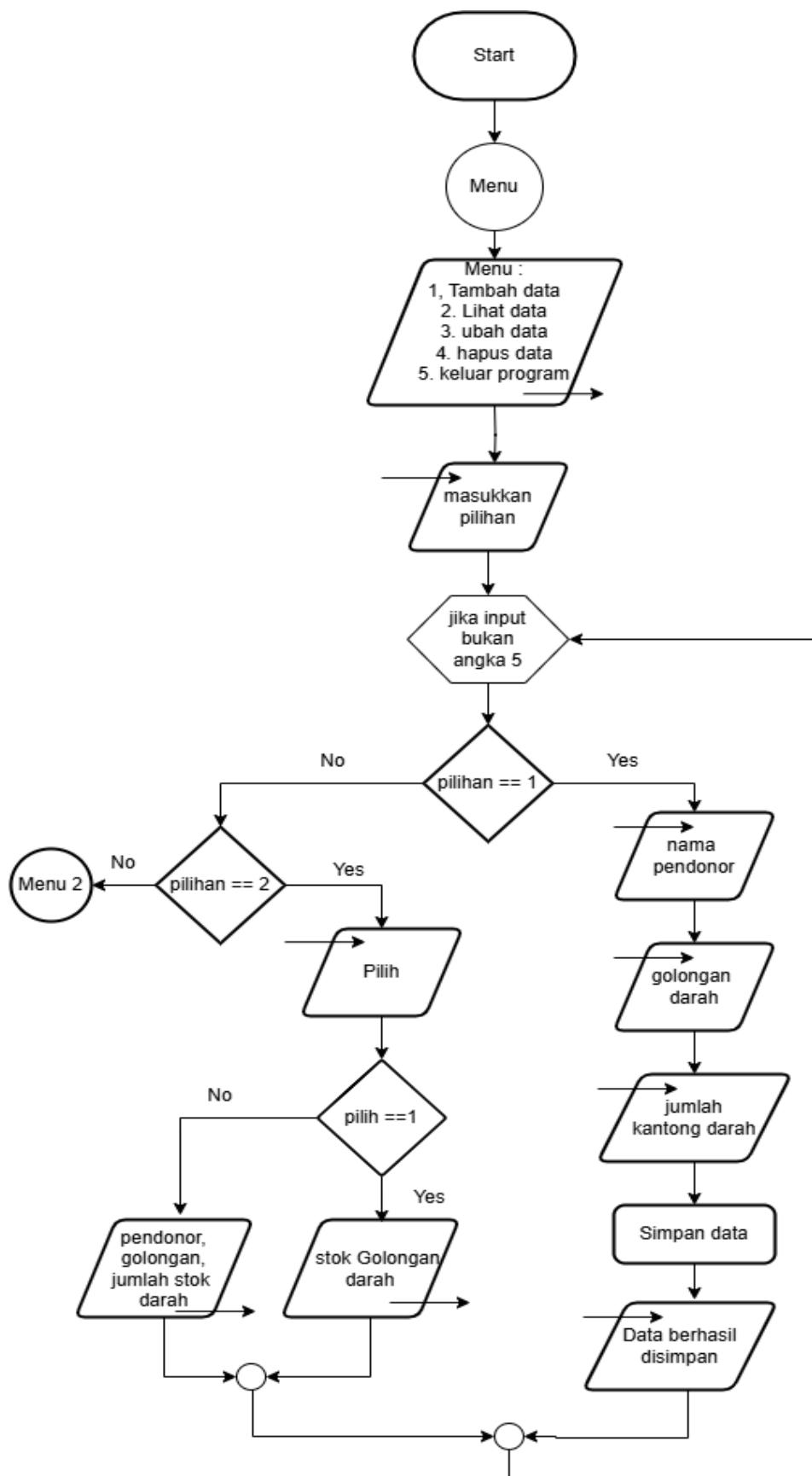
**Disusun oleh:**  
**SANTOSO PARLINDUNGAN TOGATOROP (2409106052)**  
**Kelas B1 2024**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

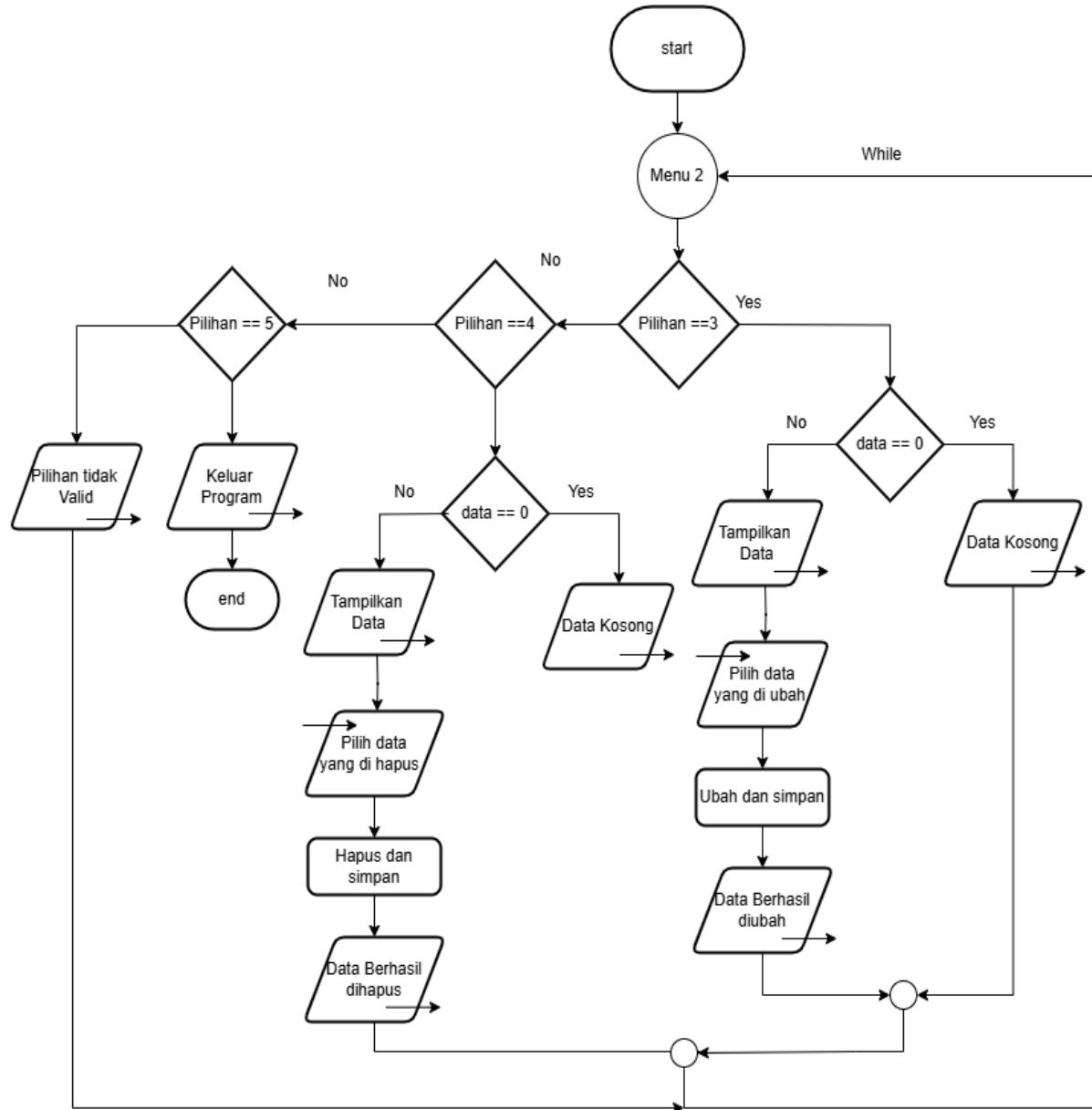
## 1. Flowchart



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.

### b. Penjelasan Alur dan Algoritma

Program ini adalah Sistem Pengelolaan Stok Kantong Darah, di awal program, user akan diminta untuk menginput *username* dan *password* (pada contoh kasus saya menggunakan Nim), apabila user salah *username* atau *password*, user akan mendapat peringatan bahwa batasan salah login adalah 3 kali apabila melewatinya akun akan terblokir.

Jika user berhasil login, user akan dihadapkan pada 4 menu pilihan, yaitu : tambah data, baca data, ubah data, hapus data, dan keluar dari program. Program tidak akan berhenti apabila user belum memilih opsi keluar dari program. Program juga akan otomatis menghitung jumlah stok kantong darah yang tersedia.

Pada saat user memilih **tambah data**, user akan diminta menginput nama pendonor, golongan darah, jumlah stok kantong darah, dan keluar program serta otomatis dikalkulasikan dan disimpan dalam sebuah variabel. Jika user memilih **baca data** program akan menampilkan 2 pilihan yaitu jumlah stok darah dan detail data (pendonor, golongan, dan jumlah stok darah), serta akan menampilkan data sesuai pilihan user. Jika user memilih **ubah data** program akan menampilkan semua isi data dan user akan memilih data yang akan diubah dan mengubahnya serta menyimpannya dalam program. Sama halnya dengan ubah data, jika user memilih **hapus data** data akan ditampilkan dan dihapus sesuai dengan yang diinput user. Jika user memilih **Keluar program** program akan berhenti dan jika memilih diluar dari ke-5 opsi program akan menampilkan “*Pilihan anda tidak valid silahkan pilih kembali*”.

Dalam program menggunakan beberapa fungsi, yaitu :

1. perulangan **for** digunakan untuk membatasi user dalam menginput Pin sebanyak 3 kali.
2. perulangan **Do While** digunakan untuk selalu menjalankan 4 menu program kecuali user menginput menu berhenti dari program.
3. **Conditional If** digunakan untuk mengarahkan user pada menu pilihannya.
4. **Conditional Case** digunakan untuk mengarahkan user ke menu sistem yang dipilih.

### 3. Source Code

#### a. Fitur Login

Fitur ini digunakan untuk memeriksa apakah *username* dan *password* sudah benar.

```
for (int i=0; i<3; i++){  
  
    // Melakukan input username dan password  
  
    cout<<"Masukkan username : ";  
    cin>>name;  
    cout<<"Masukkan password : ";  
    cin>>nim;  
  
    system("cls||clear");  
  
    if (name==nama && nim==pin){  
        switch(pilih)  
    }else if(i<2){  
        cout<<"Username atau Password salah!! Silahkan Login  
ulang"\  
        cout<<"Kesempatan anda "<<3-(i+1)<<" kali lagi"\  
    }else{  
        cout<<"Gagal login akun Anda terblokir!!"\  
        break;  
    }  
}
```

#### b. Fitur tambah data

Fitur ini adalah program untuk menambahkan data stok darah dari pendonor dan akan menyimpan jumlah stok darah yang tersedia.

```
if(jumlah>=MAX){  
    cout<<"Data sudah Penuh, Tidak dapat menambah data lagi"\  
}else{  
  
    //Melakukan input Nama, golongan darah, dan jumlah stok  
  
    cout<<"Nama Pendonor : ";  
    cin.ignore();  
    getline(cin,pendonor[jumlah]);  
    for (int i=0; i<4; i++){
```

```

        cout<<i+1<<".golongan : "<<jenis[i]<<endl;
    }
    cout<<"Golongan Darah : ";
    getline(cin,golongan[jumlah]);
    cout<<"Jumlah stok darah : ";
    cin>>stok[jumlah];
    jumlah++;
    cout<<"Data Berhasil diTambah" <<endl;
}

```

### c. Fitur Baca Data

Fitur ini menyediakan 2 pilihan bagi pengguna. Pengguna dapat hanya melihat stok kantong darah dan dapat melihat detail data pendonor, golongan, dan jumlah stok darah yang didonorkan.

```

if (pilih == 1){
    // Menampilkan stok kantong darah
    cout<<"Golongan "<<jenis[0]<<"      : "<<stokA<<endl;
    cout<<"Golongan "<<jenis[1]<<"      : "<<stokB<<endl;
    cout<<"Golongan "<<jenis[2]<<"      : "<<stokAB<<endl;
    cout<<"Golongan "<<jenis[3]<<"      : "<<stokO<<endl;
}

// Menampilkan data pendonor, golongan darah, dan stok darah

else if(pilih ==2){
    if(jumlah<=0){
        cout<<"__Data Kosong__" <<endl;
    }else {
        for (int i=0; i< jumlah; i++){
            cout<<i+1<<" |    "<<pendonor[i]<<" |  

            "<<golongan[i]<<" |    "<<stok[i]<<" | " <<endl;
        }
    }else{
        cout<<"pilihan tidak Valid" <<endl;
    }
}

```

#### d. Fitur ubah data

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

```
if(jumlah<=0){  
    cout<<"____Data kosong____";  
}else{  
  
// Melakukan Input data ulang pada data yang diubah  
  
    cout<<"ubah data ke-: ";  
    cin>>pilih;  
    cout<<"Nama Pendoron : ";  
    cin.ignore();  
    getline(cin,pendoron[pilih-1]);  
    cout<<"Golongan Darah : ";  
    getline(cin,golongan[pilih-1]);  
    cout<<"Jumlah stok darah : ";  
    cin>>stok[pilih-1];  
    cout<<"Data berhasil diubah" << endl;  
break;
```

#### e. Fitur Hapus data

Fitur ini adalah fitur hapus data

```
if(jumlah<=0){  
    cout<<"____Data Kosong____";  
}else{  
  
// Menghapus data yang dipilih  
  
    cout<<"pilihan : ";  
    cin>>pilih;  
    for (int i=pilih-1; i<jumlah-1; i++){  
        pendoron[i]=pendoron[i+1];  
        golongan[i]=golongan[i+1];  
        stok[i] = stok[i+1];  
    }  
    jumlah--;  
    cout<<"Data berhasil dihapus" << endl;  
}  
break;
```

## 4. Uji coba dan Hasil Output

### a. Uji coba

Skenario :

1. Menginput *username* dan *password* yang salah sebanyak 3 kali
2. Menginput *username* dan *password* yang benar.
3. Menginput *username* dan *password* yang benar lalu menginput menu pilihan menambah data (1).
4. Menginput *username* dan *password* yang benar lalu menginput menu pilihan membaca data (2).
5. Menginput *username* dan *password* yang benar lalu menginput menu pilihan mengubah data (3).
6. Menginput *username* dan *password* yang benar lalu menginput menu pilihan menghapus data (4).
7. Menginput *username* dan *password* yang benar lalu menginput menu pilihan berhenti dari program (5).
8. Menginput *username* dan *password* yang benar lalu menginput menu pilihan yang tidak ada pada program.
9. Menginput tipe data string pada menu login (*username* dan *password*)

### b. Hasil Output

1. Skenario ke-1

```
=====
          Menu Login
=====

Masukkan username : santoso
Masukkan password : 6052
=====
          Sistem Pengelolaan Stok Kantong Darah
=====

1. Tambah stok darah
2. Baca Data
3. Ubah Data
4. Hapus Data
5. Keluar dari Program
=====
Pilih Program : -
```

Gambar 4.1

## 2. Skenario ke-2

```
=====
      Menu Login
=====

Masukkan username : san
Masukkan password : 34
Username atau Password salah!! Silahkan Login ulang
Kesempatan anda 2 kali lagi
=====
      Menu Login
=====

Masukkan username : sav
Masukkan password : 45
Username atau Password salah!! Silahkan Login ulang
Kesempatan anda 1 kali lagi
=====
      Menu Login
=====

Masukkan username : vx
Masukkan password : 34
Gagal login akun Anda terblokir!!
```

Gambar 4.2

## 3. Skenario ke-3

```
=====
      Tambah data
=====

Nama Pendonor : Richard
1.golongan : A
2.golongan : B
3.golongan : AB
4.golongan : O
Golongan Darah : A
Jumlah stok darah : 20
Data Berhasil di Tambah
```

Gambar 4.3

## 4. Skenario ke-4

```
=====
      Data Stok Kantong Darah
=====

No|      Nama          |   Golongan    |   Jumlah darah |
1 |  Richard         |     A          |       20        |
2 |  Hendry          |     B          |       30        |
3 |  Fauzan          |   AB           |       40        |
4 |  Santoso         |     O          |       50        |
=====

Golongan A      : 20
Golongan B      : 30
Golongan AB     : 40
Golongan O      : 50
```

Gambar 4.

## 5. Skenario ke-5

Ubah Data				
No	Nama	Golongan	Jumlah darah	
1	Richard	A	20	
2	Hendry	B	30	
3	Fauzan	AB	40	
4	Santoso	O	50	

ubah data ke-: 4  
Nama Pendoron : Britama  
Golongan Darah : O  
Jumlah stok darah : 10  
Data berhasil diubah

Gambar 4.5

## 6. Skenario ke-6

Hapus Data				
No	Nama	Golongan	Jumlah darah	
1	Richard	A	20	
2	Hendry	B	30	
3	Fauzan	AB	40	
4	Britama	O	10	

pilihan : 1  
Data berhasil di hapus

No	Nama	Golongan	Jumlah darah	
1	Hendry	B	30	
2	Fauzan	AB	40	
3	Britama	O	10	

Gambar 4.6

## 7. Skenario ke-7

4. Keluar				
=====				
Pilih menu : 4				
Anda keluar dari program terimakasih				

Gambar 4.7

8. Skenario ke-8

```
=====
Sistem Pengelolaan Stok Kantong Darah
=====

1. Tambah stok darah
2. Baca Data
3. Ubah Data
4. Hapus Data
5. Keluar dari Program
=====
Pilih Program : 6
Pilihan Tidak Valid. Pilih dengan Benar
```

Gambar 4.8

9. Skenario ke-9 (perulangan tanpa henti)

```
5. Keluar dari Program
=====
Pilih Program : Pilihan Tidak Valid. Pilih dengan Benar
=====
Sistem Pengelolaan Stok Kantong Darah
=====

1. Tambah stok darah
2. Baca Data
3. Ubah Data
4. Hapus Data
5. Keluar dari Program
=====
Pilih Program : Pilihan Tidak Valid. Pilih dengan Benar
=====
Sistem Pengelolaan Stok Kantong Darah
=====

1. Tambah stok darah
2. Baca Data
=====

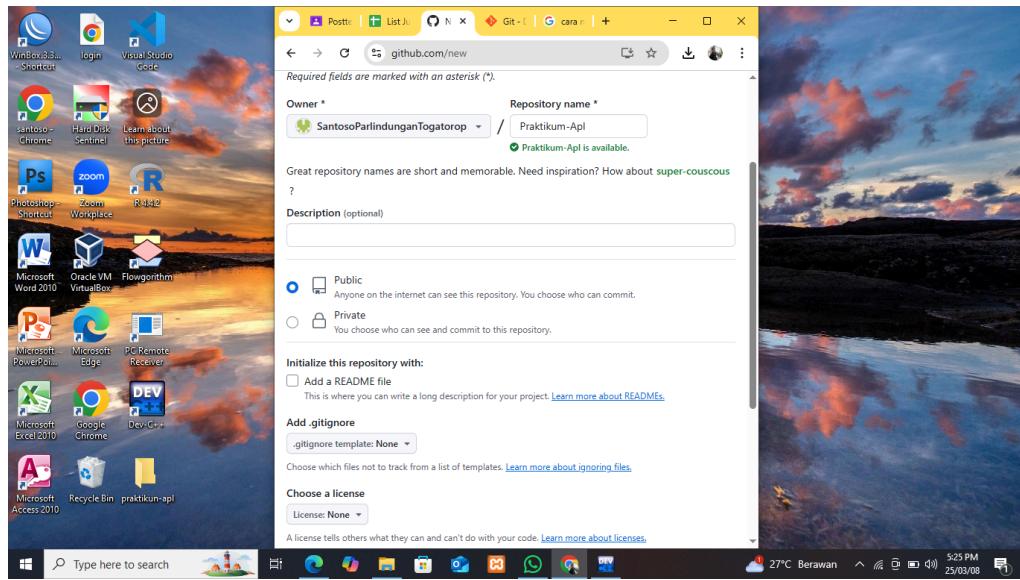
```

Gambar 4.9

## 5. Langkah-langkah membuat Git

### a. Membuat Repository pada Github

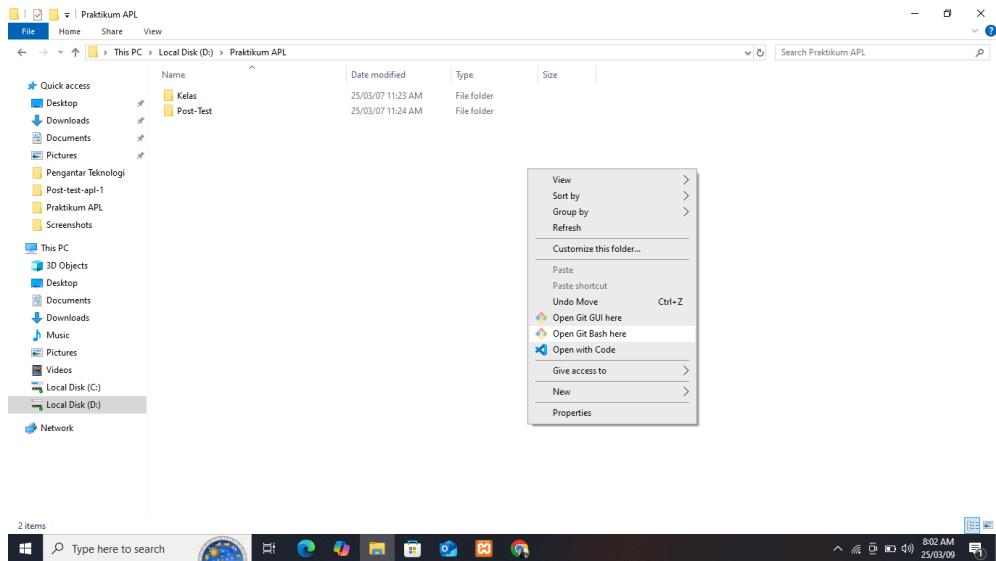
Buka repository yang sudah pernah dibuat, karena langkah ini akan menambah file bukan membuat file baru.



Gambar 5.1

### b. 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

### c. Cek status git

Cek status git ini berfungsi untuk melihat folder kita yang sudah tersimpan di folder git. file pada directory local yang ditambahkan belum terlist dalam git, hal ini karena filenya belum ditambahkan.

```
MINGW64:/d/Praktikum APL
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:   Post-Test/Post-test-apl-1/2409106052-SantosoPtogatorop-PT-1.cpp
    deleted:   Post-Test/Post-test-apl-1/2409106052-SantosoPtogatorop-PT-1.exe
    deleted:   Post-Test/Post-test-apl-1/2409106052-SantosoPtogatorop-PT-1.pdf

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Kelas/Pertemuan 2/
      Post-Test/Post-test-apl-1/2409106052-SantosoPtogatorop-PT-1.cpp
      Post-Test/Post-test-apl-1/2409106052-SantosoPtogatorop-PT-1.exe
      Post-Test/Post-test-apl-1/2409106052-SantosoPtogatorop-PT-1.pdf
      Post-Test/Post-test-apl-2/

no changes added to commit (use "git add" and/or "git commit -a")
```

Gambar 5.5

### d. Menambahkan folder

Membuat list folder apa saja yang akan dimasukkan ke dalam git. Gunakan 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.

```
SANTOSO TOGATOROP@DESKTOP-SANTOSO MINGW64 /d/Praktikum APL (main)
$ git add .
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
  add

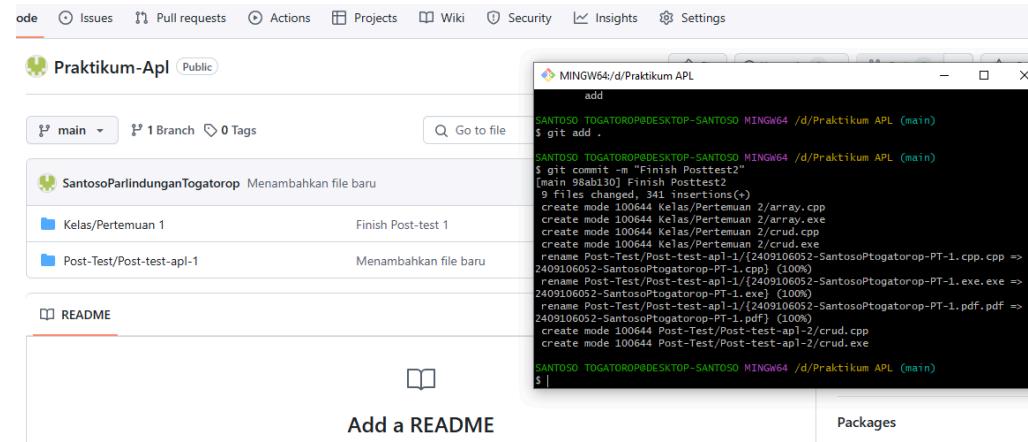
SANTOSO TOGATOROP@DESKTOP-SANTOSO MINGW64 /d/Praktikum APL (main)
$ git add .

SANTOSO TOGATOROP@DESKTOP-SANTOSO MINGW64 /d/Praktikum APL (main)
$ |
```

Gambar 5.7

## e. Commit folder

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



The screenshot shows a GitHub repository named "Praktikum-Apl". The repository has 1 branch and 0 tags. A commit from "SantosoParlindunganTogatorop" titled "Menambahkan file baru" is shown, along with another commit from "Kelas/Pertemuan 1" titled "Finish Post-test 1". Below the commits is a "README" file. On the right, there is a terminal window with the following command and output:

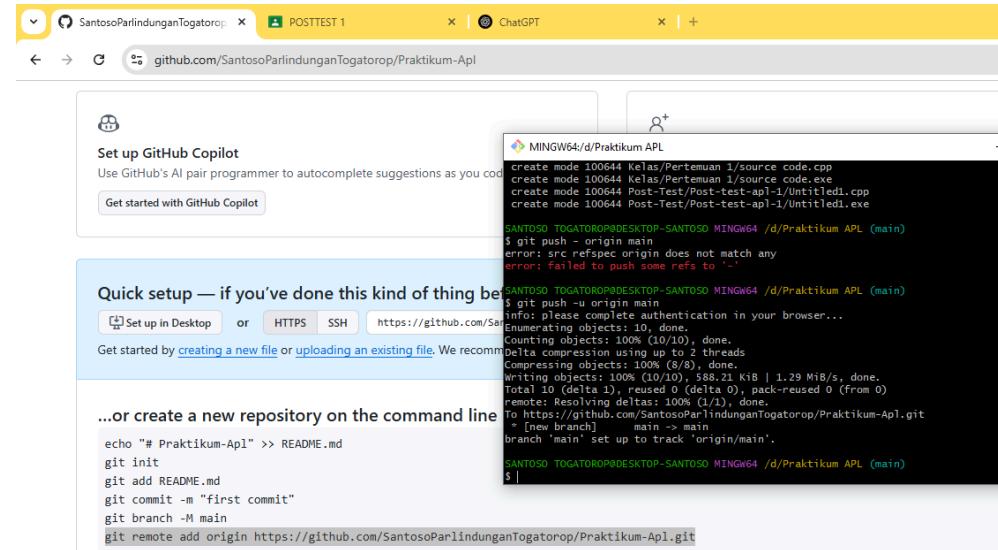
```
SANTOSO TOGATOROP@DESKTOP-SANTOSO MINGW64 /d/Praktikum APL (main)
$ git add .
SANTOSO TOGATOROP@DESKTOP-SANTOSO MINGW64 /d/Praktikum APL (main)
$ git commit -m "Finish Posttest2"
[main 98ab130] Finish Posttest2
 9 files changed, 341 insertions(+)
create mode 100644 kelas/Pertemuan 2/array.cpp
create mode 100644 kelas/Pertemuan 2/array.exe
create mode 100644 kelas/Pertemuan 2/crud.cpp
create mode 100644 kelas/Pertemuan 2/crud.exe
rename Post-Test/Post-test-apl-1/2409106052-SantosoPtogatorop-PT-1.cpp => 2409106052-SantosoPtogatorop-PT-1.exe (100%)
rename Post-Test/Post-test-apl-1/2409106052-SantosoPtogatorop-PT-1.exe => 2409106052-SantosoPtogatorop-PT-1.pdf (100%)
create mode 100644 Post-Test/Post-test-apl-2/crud.cpp
create mode 100644 Post-Test/Post-test-apl-2/crud.exe

SANTOSO TOGATOROP@DESKTOP-SANTOSO MINGW64 /d/Praktikum APL (main)
$ |
```

Gambar 5.9

## f. 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.



The screenshot shows a browser window with a GitHub Copilot sidebar. The main area displays the GitHub quick setup steps for creating a new repository. It includes instructions to "Set up GitHub Copilot", "Get started with GitHub Copilot", and "Quick setup — if you've done this kind of thing before". Below these are links to "Set up in Desktop" or "HTTPS / SSH" and a URL "https://github.com/SantosoParlindunganTogatorop/Praktikum-Apl". To the right, there is a terminal window with the following command and output:

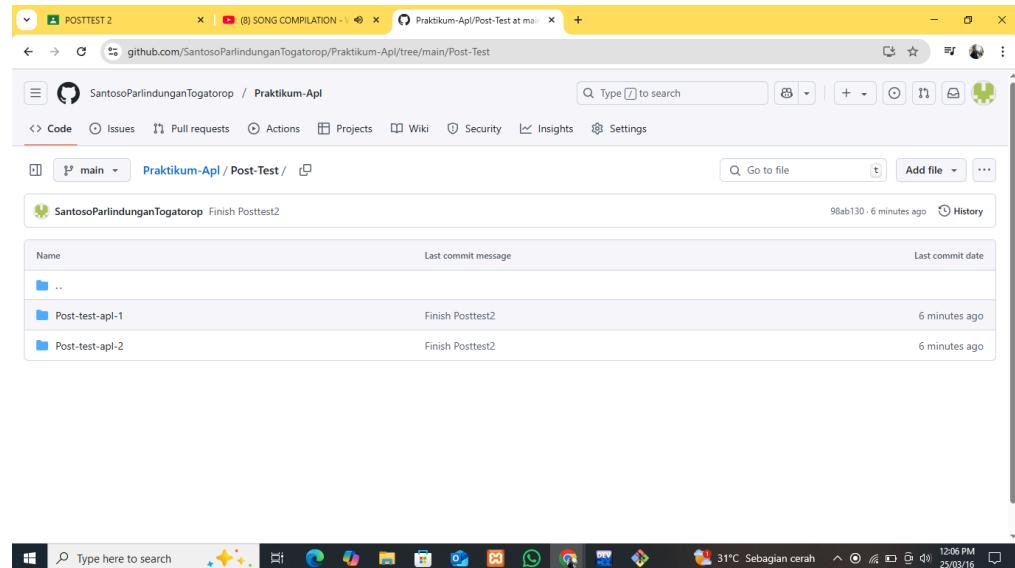
```
SANTOSO TOGATOROP@DESKTOP-SANTOSO MINGW64 /d/Praktikum APL (main)
$ git push -u origin main
error: src refspec origin does not match any
error: failed to push some refs to '.'

SANTOSO TOGATOROP@DESKTOP-SANTOSO MINGW64 /d/Praktikum APL (main)
$ git push -u origin main
Uploading objects: 100% (10/10), done.
Counting objects: 10, done.
Delta compression using up to 2 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (10/10), 588.21 KiB | 1.29 MiB/s, done.
Total 10 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/SantosoParlindunganTogatorop/Praktikum-Apl.git
 [new branch]      main    -> main
branch 'main' set up to track 'origin/main'.
```

Gambar 5.10

### g. 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