

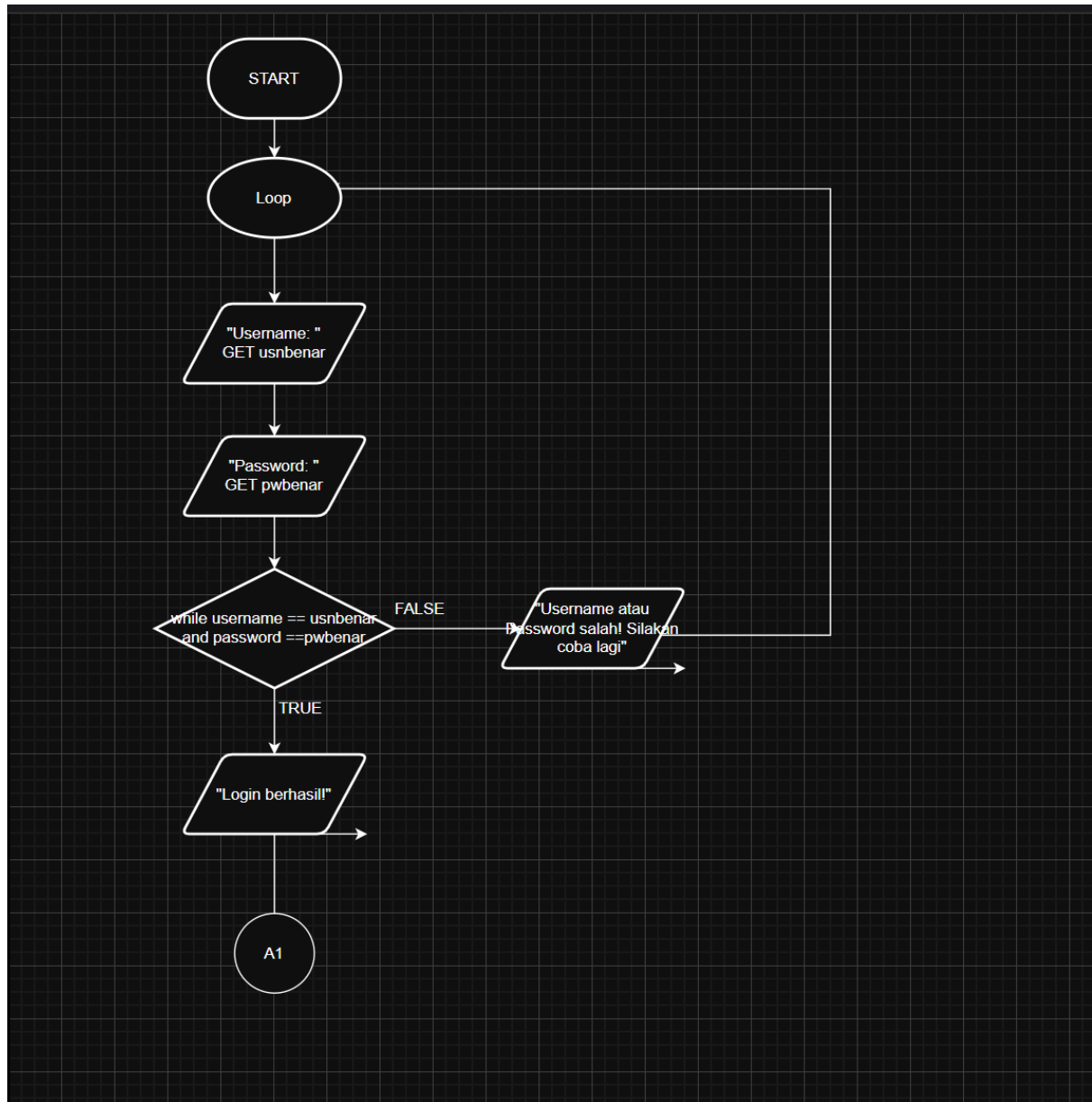
LAPORAN PRAKTIKUM
POSTTEST (4)
ALGORITMA PEMROGRAMAN DASAR

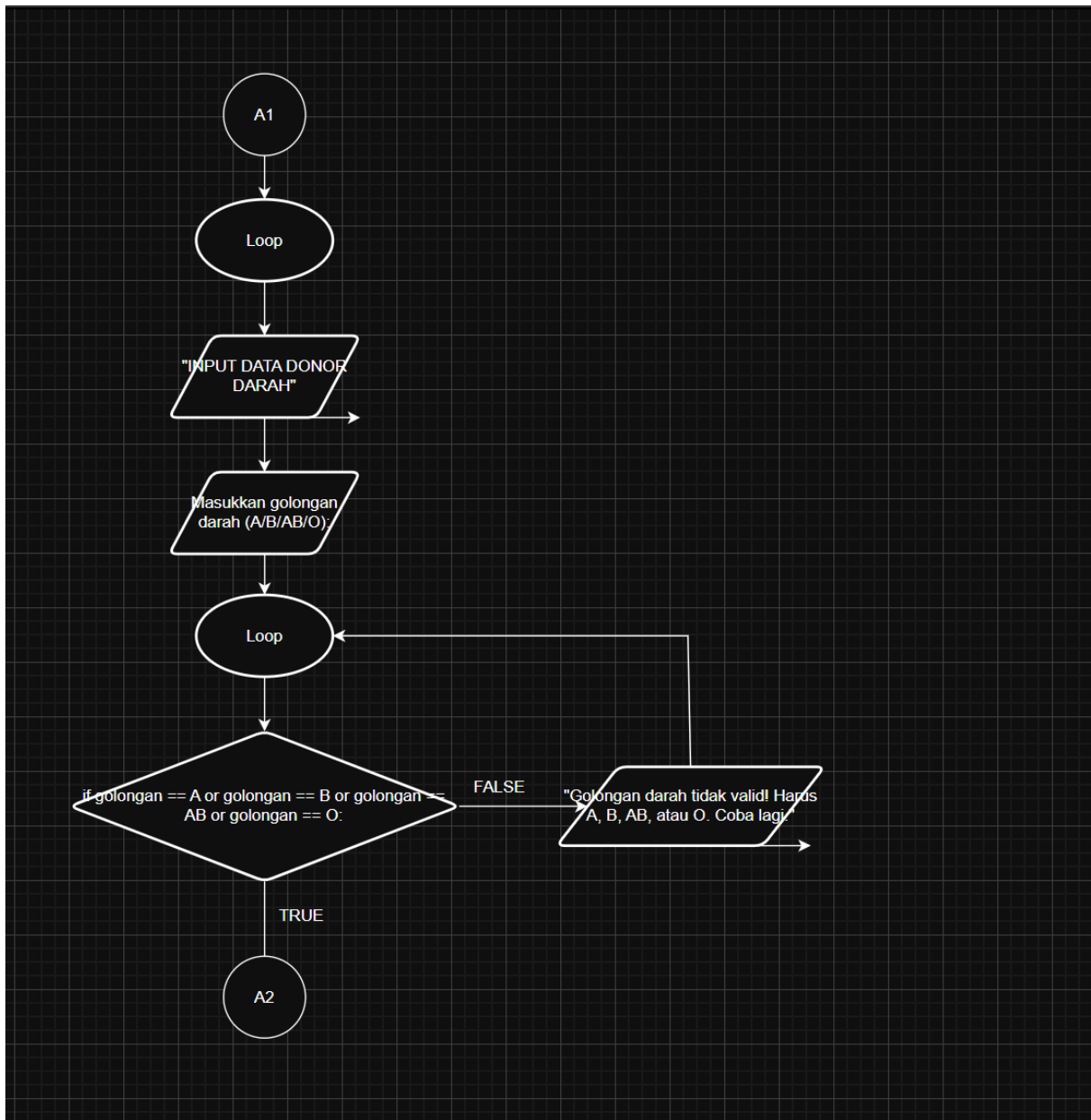


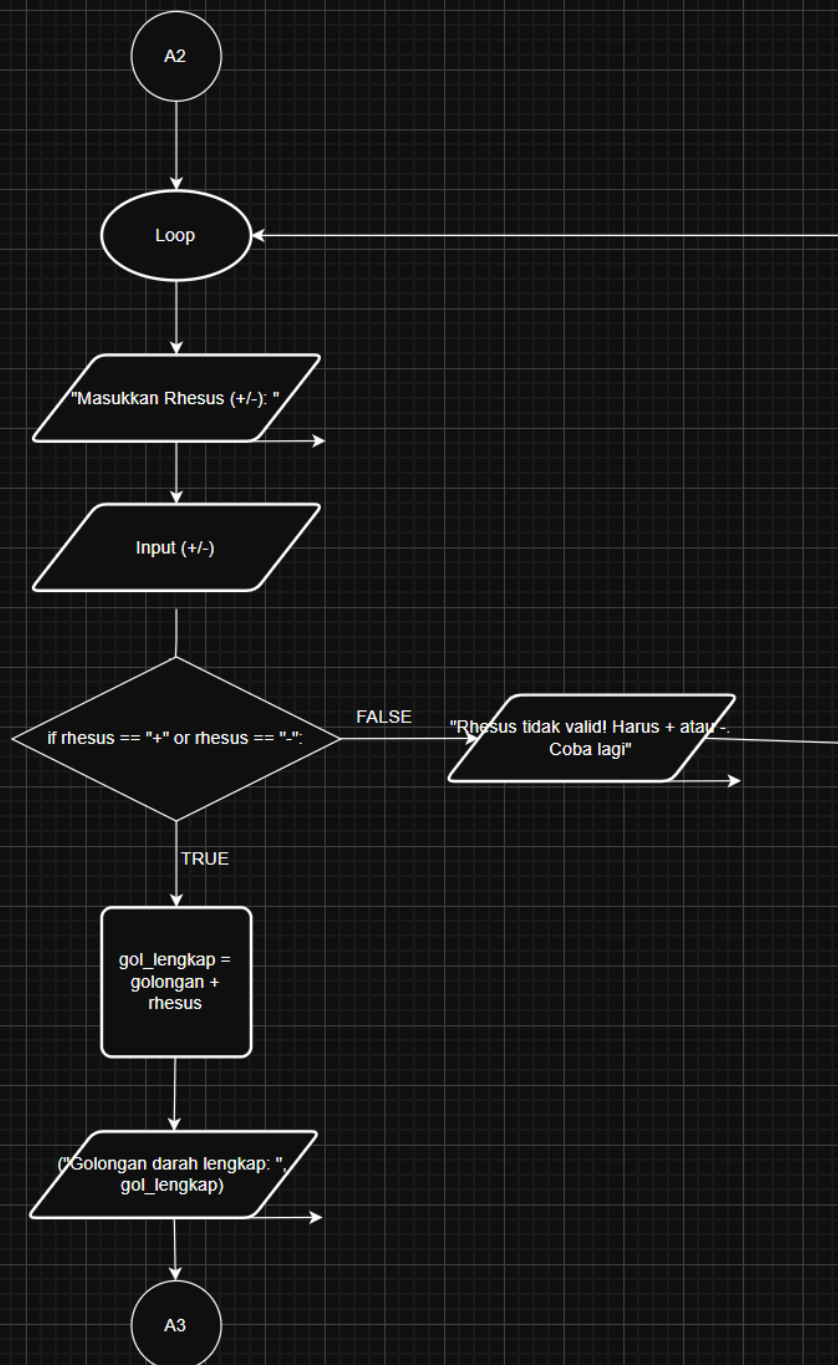
Disusun oleh:
Ulung Priskeaji Bintang Ramadhan (2509106051)
Kelas (B1'25)

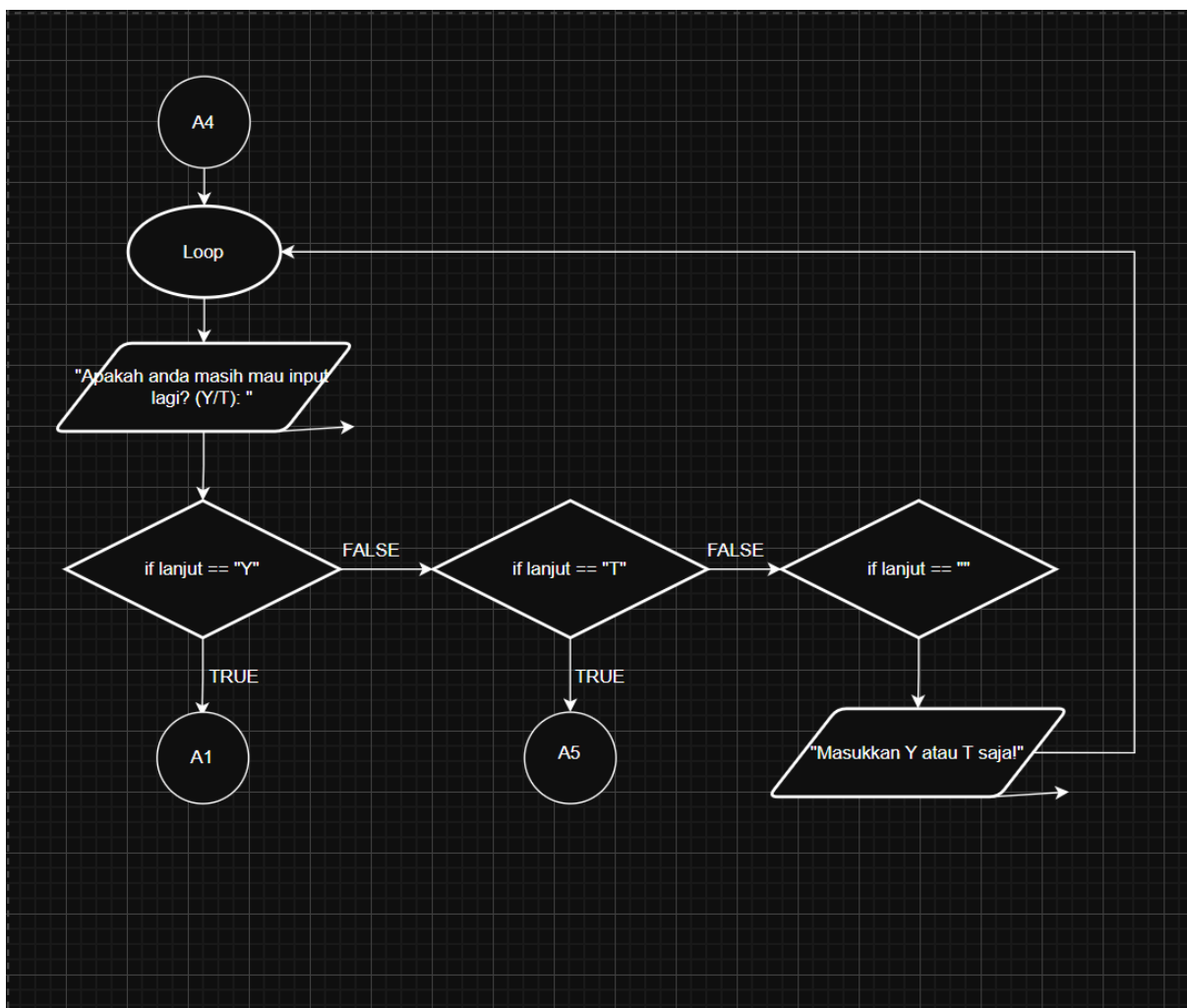
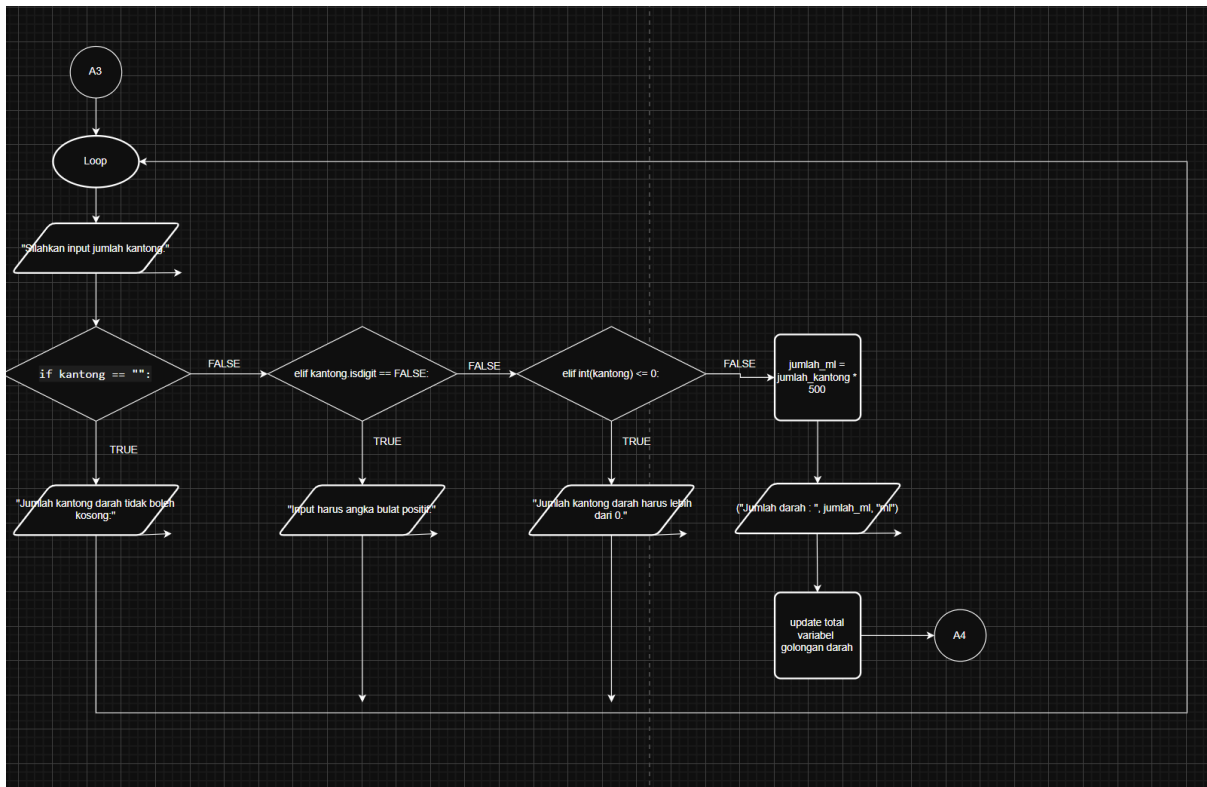
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

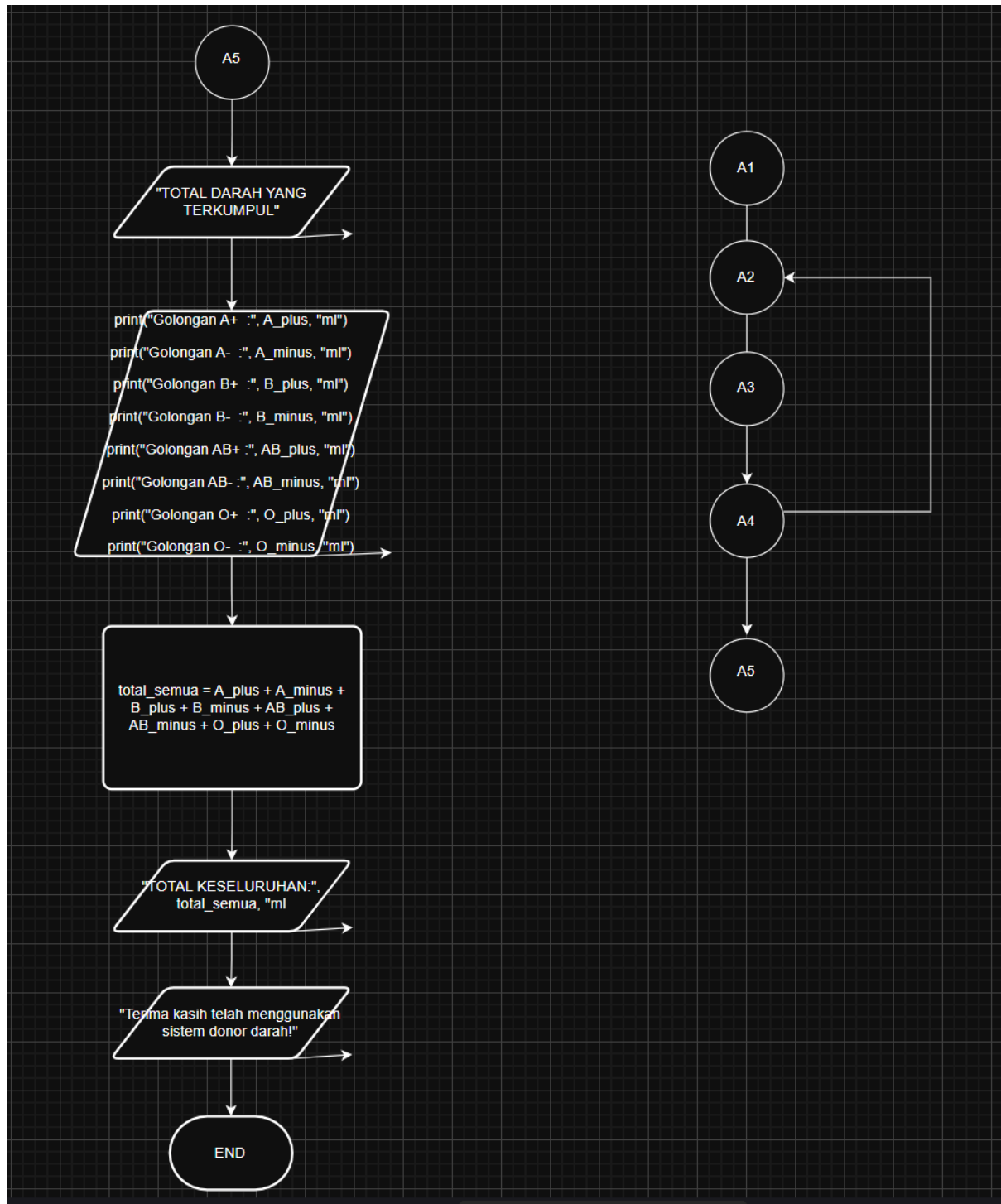
1. Flowchart











Gambar <Flowchart draw.io>

Dalam program ini, yang pertama dilakukan adalah login dengan input username dan password. Jika username dan password sesuai dengan data yang benar (ulung dan 051), maka program akan menampilkan pesan “Login berhasil!” dan melanjutkan ke tahap berikutnya.

Namun, jika salah, program akan menampilkan pesan “Username atau Password salah! Silakan coba lagi.” dan meminta pengguna untuk menginput ulang sampai benar.

Setelah berhasil login, pengguna akan masuk ke bagian input data donor darah. Pada tahap ini, pengguna diminta untuk menginput beberapa data, yaitu golongan darah (A/B/AB/O), rhesus (+/-), dan jumlah kantong darah. Setiap 1 kantong darah dihitung sebagai 500 ml. Program juga akan melakukan pengecekan agar input yang dimasukkan valid (misalnya tidak boleh selain A, B, AB, O untuk golongan darah dan rhesus harus +/-).

Setelah semua data dimasukkan, program akan menghitung jumlah total darah (dalam ml) berdasarkan golongan dan rhesus yang dipilih. Data ini kemudian disimpan ke dalam variabel sesuai golongan darahnya, misalnya A_plus, B_minus, dan seterusnya.

Setelah satu data donor selesai diinput, pengguna akan diberi pilihan apakah ingin melanjutkan input data lagi (Y) atau tidak (T). Jika memilih “Y”, program akan kembali meminta input data donor baru. Jika memilih “T”, program akan menampilkan ringkasan total darah yang telah terkumpul berdasarkan setiap golongan dan rhesus, serta menampilkan total keseluruhan darah (dalam ml). Terakhir, program menampilkan pesan penutup:

“Terima kasih telah menggunakan sistem donor darah!” dan program akan berhenti.

2. Deskripsi Singkat Program

Program ini berfungsi untuk mencatat dan menghitung total darah yang terkumpul berdasarkan golongan darah dan rhesus (+/-). Program hanya dapat dijalankan oleh pengguna yang berhasil login menggunakan username dan password yang benar. Setelah login, pengguna dapat memasukkan data golongan darah, rhesus, dan jumlah kantong darah yang akan otomatis dikonversi menjadi satuan mililiter. Pada akhir program, sistem menampilkan total darah yang terkumpul dari setiap golongan beserta total keseluruhannya. Adapun fungsi program yang digunakan adalah sebagai berikut:

Fungsi Program:

- Login (verifikasi pengguna) : Meminta input username dan password, lalu memverifikasi kecocokannya agar hanya pengguna terdaftar yang bisa mengakses sistem.
- Input data donor darah : Meminta pengguna memasukkan golongan darah (A, B, AB, O), rhesus (+/-), dan jumlah kantong darah.
- Perhitungan volume darah : Menghitung total darah (dalam ml) berdasarkan jumlah kantong darah (1 kantong = 500 ml).
- Percabangan if, elif, dan else : Digunakan untuk menentukan jenis golongan darah yang sesuai dan menambahkan hasil ke variabel yang tepat.
- Perulangan while : Digunakan untuk mengulangi proses login sampai berhasil dan memungkinkan pengguna melakukan input data donor lebih dari satu kali.
- Output hasil : Menampilkan total darah yang terkumpul per golongan serta total keseluruhan darah yang telah didonorkan.

3. Source Code

Source Code:

```
# login
usnbenar = "ulung"
pwbenar = "051"

print("Silahkan Masukkan Username dan Password Anda")

login_berhasil = False
while login_berhasil == False:
    username = input("Username: ")
    password = input("Password: ")

    if username == usnbenar and password == pwbenar:
        login_berhasil = True
        print("Login berhasil!")
    else:
        print("Username atau Password salah! Silahkan coba lagi.")

# variabel penyimpan total darah
A_plus = 0
A_minus = 0
B_plus = 0
B_minus = 0
AB_plus = 0
AB_minus = 0
O_plus = 0
O_minus = 0

# loop input data donor
lanjut = "Y"

while lanjut == "Y":
    print()
```

```

print("INPUT DATA DONOR DARAH")

# input golongan darah
golongan_valid = False
while golongan_valid == False:
    golongan = input("Masukkan golongan darah (A/B/AB/O): ").upper()
    if golongan == "A" or golongan == "B" or golongan == "AB" or
golongan == "O":
        golongan_valid = True
    else:
        print("Golongan darah tidak valid! Harus A, B, AB, atau O.")

# input rhesus
rhesus_valid = False
while rhesus_valid == False:
    rhesus = input("Masukkan Rhesus (+/-): ")
    if rhesus == "+" or rhesus == "-":
        rhesus_valid = True
    else:
        print("Rhesus tidak valid! Harus + atau -.")

gol_lengkap = golongan + rhesus
print("Golongan darah lengkap:", gol_lengkap)

# input jumlah kantong darah
kantong_valid = False
while kantong_valid == False:
    kantong = input("Masukkan jumlah kantong darah: ").strip()
    if kantong == "":
        print("Jumlah kantong darah tidak boleh kosong.")
    elif kantong.isdigit() == False:
        print("Input harus angka bulat positif.")
    elif int(kantong) <= 0:
        print("Jumlah kantong darah harus lebih dari 0.")
    else:
        kantong_valid = True

jumlah_kantong = int(kantong)
jumlah_ml = jumlah_kantong * 500
print()
print("Jumlah darah:", jumlah_kantong, "kantong =", jumlah_ml, "ml")

```

```

# update total variabel penyimpan darah
if gol_lengkap == "A+":
    A_plus += jumlah_ml
    print("Golongan Darah: A+")
elif gol_lengkap == "A-":
    A_minus += jumlah_ml
    print("Golongan Darah: A-")
elif gol_lengkap == "B+":
    B_plus += jumlah_ml
    print("Golongan Darah: B+")
elif gol_lengkap == "B-":
    B_minus += jumlah_ml
    print("Golongan Darah: B-")
elif gol_lengkap == "AB+":
    AB_plus += jumlah_ml
    print("Golongan Darah: AB+")
elif gol_lengkap == "AB-":
    AB_minus += jumlah_ml
    print("Golongan Darah: AB-")
elif gol_lengkap == "O+":
    O_plus += jumlah_ml
    print("Golongan Darah: O+")
elif gol_lengkap == "O-":
    O_minus += jumlah_ml
    print("Golongan Darah: O-")

# lanjut/tidak
lanjut_valid = False
while lanjut_valid == False:
    lanjut = input("Apakah anda masih mau input lagi? (Y/T): ").upper()
    if lanjut == "Y" or lanjut == "T":
        lanjut_valid = True
    else:
        print("Masukkan Y atau T saja!")

# ringkasan
print()
print("TOTAL DARAH YANG TERKUMPUL")

print("Golongan A+ :", A_plus, "ml")

```

```

print("Golongan A-  :", A_minus, "ml")
print("Golongan B+  :", B_plus, "ml")
print("Golongan B-  :", B_minus, "ml")
print("Golongan AB+ :", AB_plus, "ml")
print("Golongan AB- :", AB_minus, "ml")
print("Golongan O+  :", O_plus, "ml")
print("Golongan O-  :", O_minus, "ml")

total_semua = A_plus + A_minus + B_plus + B_minus + AB_plus + AB_minus +
O_plus + O_minus
print("TOTAL KESELURUHAN:", total_semua, "ml")
print()
print("Terima kasih telah menggunakan sistem donor darah!")

```

4. Hasil Output

```

Silahkan Masukkan Usermame dan Password Anda
Username: ulung
Password: 051
Login berhasil!

INPUT DATA DONOR DARAH
Masukkan golongan darah (A/B/AB/O): O
Masukkan Rhesus (+/-): -
Golongan darah lengkap: O-
Silahkan input jumlah kantong.
Input jumlah kantong darah: 7

Jumlah darah: 7 kantong = 3500 ml
Golongan Darah: O-
Apakah anda masih mau input lagi? (Y/T): Y

```

```
INPUT DATA DONOR DARAH
Masukkan golongan darah (A/B/AB/O): B
Masukkan Rhesus (+/-): -
Golongan darah lengkap: B-
Silahkan input jumlah kantong.
Input jumlah kantong darah: 2

Jumlah darah: 2 kantong = 1000 ml
Golongan Darah: B-
Apakah anda masih mau input lagi? (Y/T): T

TOTAL DARAH YANG TERKUMPUL
Golongan A+   : 0 ml
Golongan A-   : 0 ml
Golongan B+   : 0 ml
Golongan B-   : 1000 ml
Golongan AB+  : 0 ml
Golongan AB-  : 0 ml
Golongan O+   : 0 ml
Golongan O-   : 3500 ml
TOTAL KESELURUHAN: 4500 ml

Terima kasih telah menggunakan sistem donor darah!
PS C:\praktikum-apd>
```

Gambar <Hasil Output>

5. Langkah-langkah GIT

```
PS C:\praktikum-apd> git add .
PS C:\praktikum-apd> git commit -m "upload pt4"
[main 156e8b9] upload pt4
 1 file changed, 118 deletions(-)
 delete mode 100644 kelas/tes
PS C:\praktikum-apd> git push
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 28 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (11/11), 2.59 KiB | 2.59 MiB/s, done.
Total 11 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
remote: This repository moved. Please use the new location:
remote:   https://github.com/UlungPriskeaji/praktikum-apd.git
To https://github.com/UlungPriskeaji/praktikumapd.git
   22b1a6f..156e8b9  main -> main
PS C:\praktikum-apd>
```

Gambar <Proses GIT>

5.1 GIT Add

GIT add merupakan perintah yang digunakan untuk menambahkan perubahan pada folder/file ke area staging sebelum melakukan commit. Area staging adalah area perantara yang menyimpan informasi tentang perubahan yang akan dimasukkan dalam commit berikutnya.

5.2 GIT Commit

GIT Commit adalah instruksi yang digunakan untuk menyimpan perubahan yang telah dimasukkan ke dalam staging area ke dalam repository Git lokal secara tetap. Setiap commit menyimpan cuplikan proyek pada saat itu dan disertai pesan deskriptif yang menjelaskan perubahan yang telah dilakukan.

5.3 GIT Push

GIT Push digunakan untuk mengirim commit dari repository lokal ke repository remote (misalnya GitHub).