

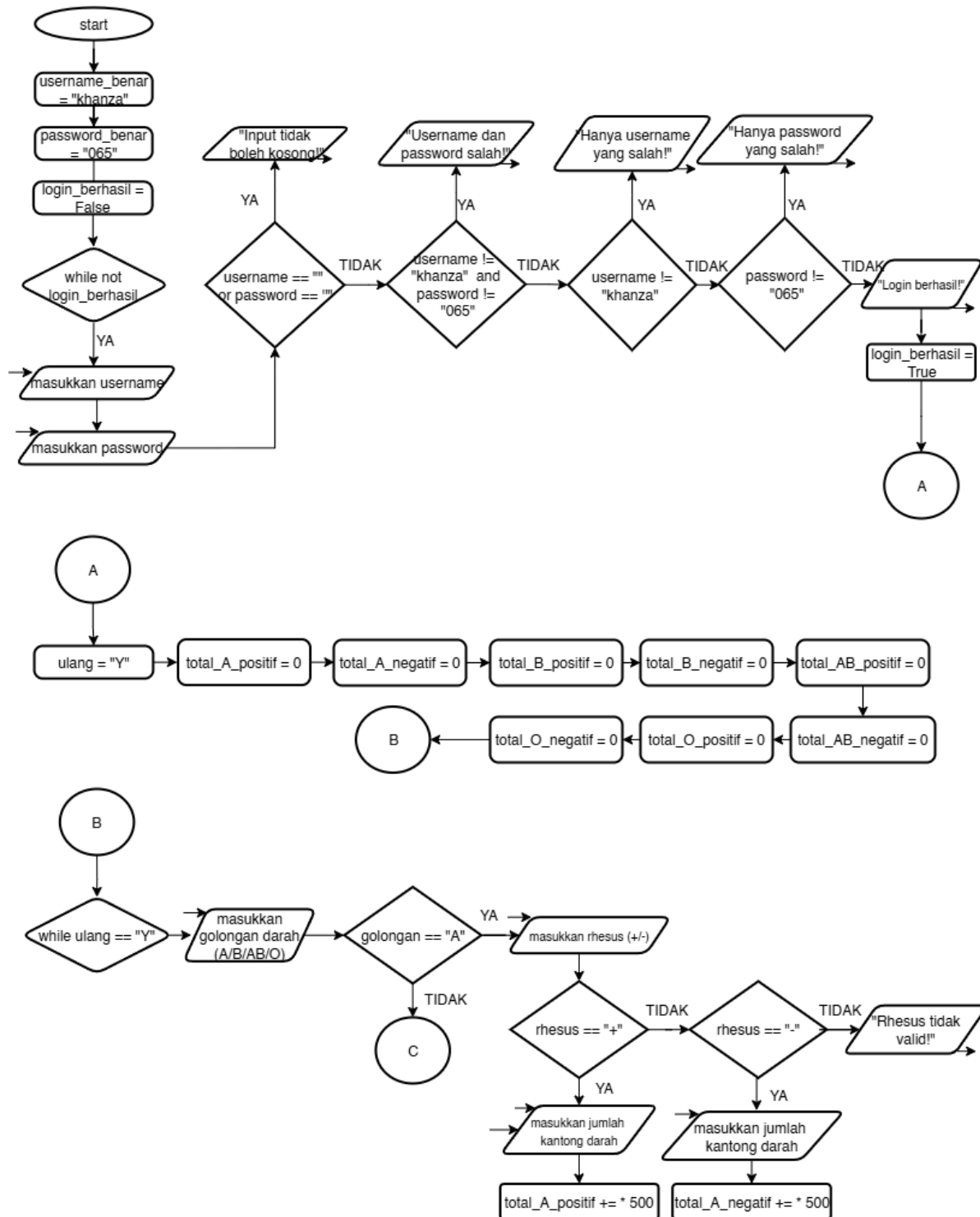
LAPORAN PRAKTIKUM
POSTTEST (4)
ALGORITMA PEMROGRAMAN DASAR

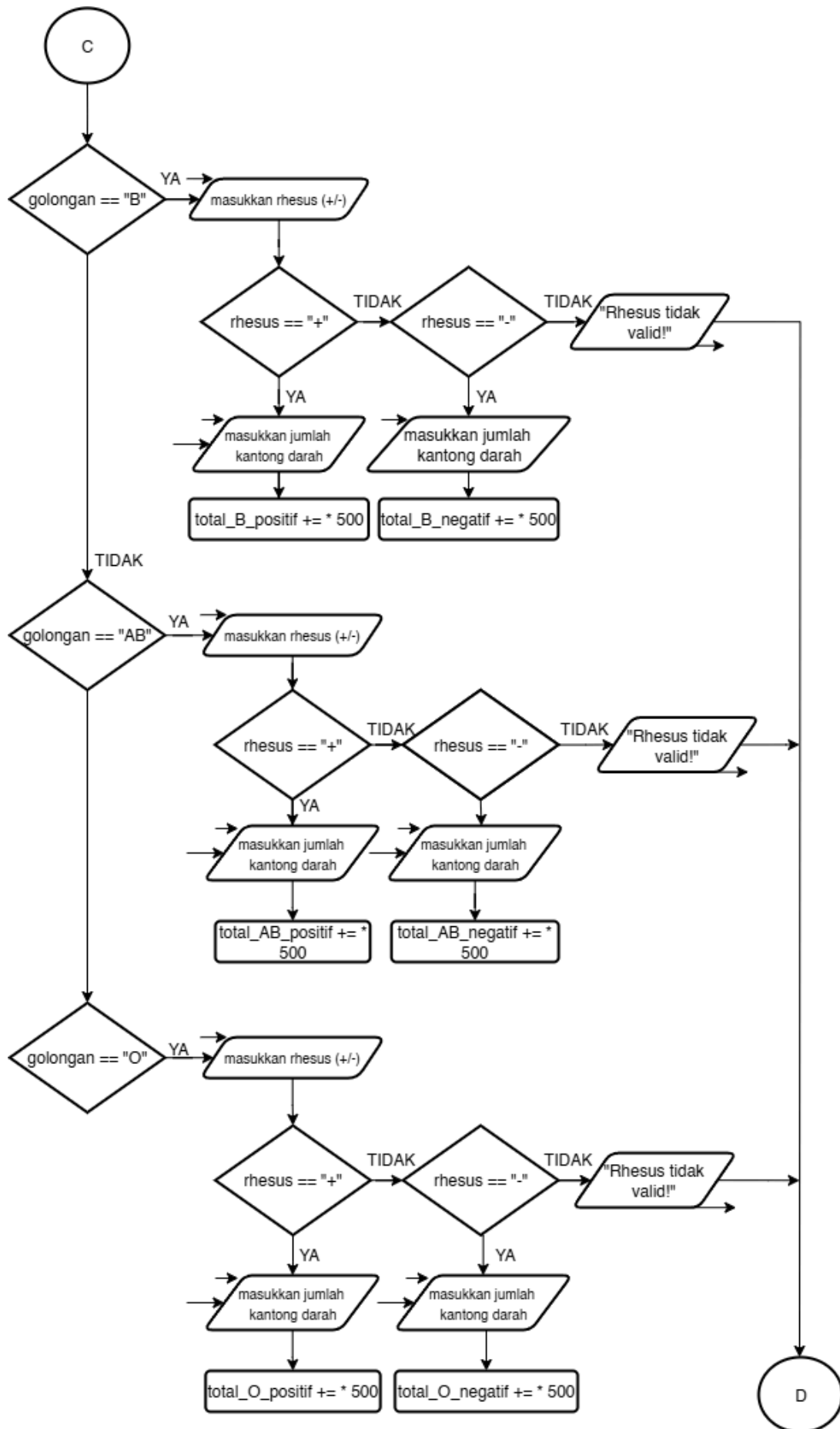


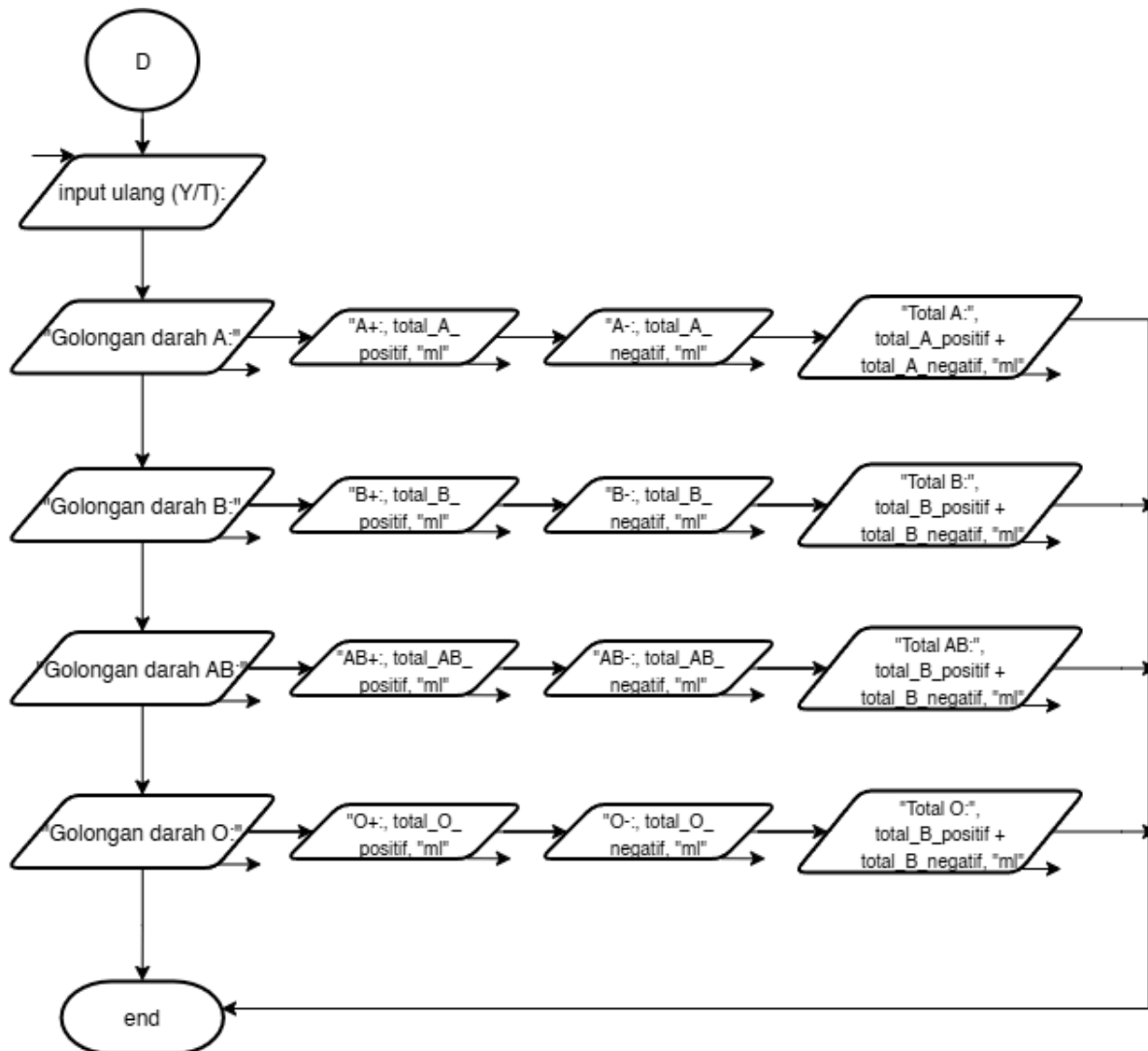
Disusun oleh:
Nama (2509106065)
Kelas (B1 '25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart







Program dimulai dengan proses login yang meminta pengguna memasukkan username dan password, dan akan terus berulang sampai data yang dimasukkan benar. Jika username atau password salah, maka akan muncul pesan kesalahan sesuai kondisi, dan jika keduanya benar program menampilkan pesan “Login berhasil” lalu lanjut ke tahap berikutnya. Setelah login, pengguna diminta menginput data golongan darah (A, B, AB, O), rhesus (+ atau -), dan jumlah kantong darah, di mana setiap kombinasi golongan dan rhesus akan menambah total volume darah sebesar 500 ml per kantong. Jika input tidak valid, program akan menampilkan pesan kesalahan dan meminta input ulang sampai pengguna memilih berhenti. Setelah semua data dimasukkan, program menampilkan ringkasan jumlah darah per golongan dan rhesus serta menghitung total keseluruhan darah yang terkumpul, kemudian program berakhir.

2. Deskripsi Singkat Program

Program ini dibuat dengan tujuan untuk melatih kemampuan dalam memahami logika dasar pemrograman, khususnya pada proses login dan pengolahan data. Fungsinya adalah untuk memverifikasi data pengguna melalui sistem login, kemudian menghitung dan menampilkan total darah berdasarkan golongan serta rhesusnya. Manfaat dari program ini yaitu membantu memahami penerapan perulangan, percabangan, dan pengolahan data agar hasilnya dapat ditampilkan secara otomatis dan terstruktur.

3. Source Code

A. Kode POST TEST 4

Source Code:

```
# Form Login
username_benar = "khanza"
password_benar = "065"
login_berhasil = False
while not login_berhasil: # Selama login masih belum
    username = input("Silakan masukkan username:")
    password = input("Silakan masukkan password:")

    # Penanganan untuk input kosong
    if username == "" or password == "":
        print("Input tidak boleh kosong!")
    elif username != "khanza" and password != "065":
        print("Username dan password salah!")
    elif username != "khanza":
        print("Hanya username yang salah!")
    elif password != "065":
        print("Hanya password yang salah!")
    else:
        print("Login berhasil.")
        login_berhasil = True # keluar dari while
```

```

# Input golongan darah dan rhesus
ulang = "Y"

# Variabel total darah
total_A_positif = 0
total_A_negatif = 0
total_B_positif = 0
total_B_negatif = 0
total_AB_positif = 0
total_AB_negatif = 0
total_O_positif = 0
total_O_negatif = 0

while ulang == "Y":
    golongan = input("Masukkan golongan darah Anda (A/B/AB/O): ")

    if golongan == "A":
        rhesus = input("Masukkan rhesus Anda (+/-): ")
        if rhesus == "+":
            total_A_positif += int(input("Masukkan jumlah kantong darah: "))
* 500
        elif rhesus == "-":
            total_A_negatif += int(input("Masukkan jumlah kantong darah: "))
* 500
        else:
            print("Rhesus tidak valid!")

    elif golongan == "B":
        rhesus = input("Masukkan rhesus Anda (+/-): ")
        if rhesus == "+":
            total_B_positif += int(input("Masukkan jumlah kantong darah: "))
* 500
        elif rhesus == "-":
            total_B_negatif += int(input("Masukkan jumlah kantong darah: "))
* 500
        else:
            print("Rhesus tidak valid!")

    elif golongan == "AB":
        rhesus = input("Masukkan rhesus Anda (+/-): ")
        if rhesus == "+":

```

```

        total_AB_positif += int(input("Masukkan jumlah kantong darah:
")) * 500
    elif rhesus == "-":
        total_AB_negatif += int(input("Masukkan jumlah kantong darah:
")) * 500
    else:
        print("Rhesus tidak valid!")

elif golongan == "0":
    rhesus = input("Masukkan rhesus (+/-): ")
    if rhesus == "+":
        total_0_positif += int(input("Masukkan jumlah kantong darah: "))
* 500
    elif rhesus == "-":
        total_0_negatif += int(input("Masukkan jumlah kantong darah: "))
* 500
    else:
        print("Rhesus tidak valid!")
else:
    print("Golongan darah tidak valid!")

ulang = input("Apakah Anda ingin input lagi? (Y/T): ")

# OUTPUT RINGKASAN (Poin Plus)
print("RINGKASAN DATA DARAH")
print("=====")
print(">> Golongan Darah A:")
print("  A+ :", total_A_positif, "ml")
print("  A- :", total_A_negatif, "ml")
print("  Total A :", total_A_positif + total_A_negatif, "ml")

print(">> Golongan Darah B:")
print("  B+ :", total_B_positif, "ml")
print("  B- :", total_B_negatif, "ml")
print("  Total B :", total_B_positif + total_B_negatif, "ml")

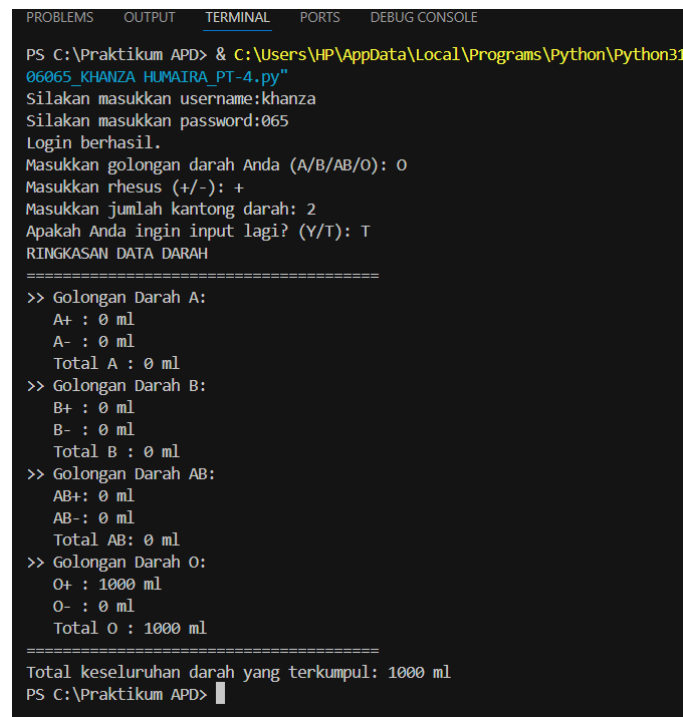
print(">> Golongan Darah AB:")
print("  AB+:", total_AB_positif, "ml")
print("  AB-:", total_AB_negatif, "ml")
print("  Total AB:", total_AB_positif + total_AB_negatif, "ml")

```

```
print(">> Golongan Darah O:")
print("  O+ :", total_O_positif, "ml")
print("  O- :", total_O_negatif, "ml")
print("  Total O :", total_O_positif + total_O_negatif, "ml")
print("=====")

# Hitung total keseluruhan semua darah
total_semua = (
    total_A_positif + total_A_negatif +
    total_B_positif + total_B_negatif +
    total_AB_positif + total_AB_negatif +
    total_O_positif + total_O_negatif
)
print(f"Total keseluruhan darah yang terkumpul: {total_semua} ml")
```


4. Hasil Output



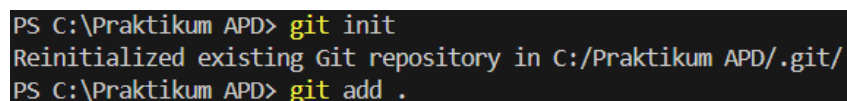
```
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
PS C:\Praktikum APD> & C:\Users\HP\AppData\Local\Programs\Python\Python31
06065_KHANZA HUMAIRA_PT-4.py"
Silakan masukkan username:khanza
Silakan masukkan password:065
Login berhasil.
Masukkan golongan darah Anda (A/B/AB/O): O
Masukkan rhesus (+/-): +
Masukkan jumlah kantong darah: 2
Apakah Anda ingin input lagi? (Y/T): T
RINGKASAN DATA DARAH
=====
>> Golongan Darah A:
A+ : 0 ml
A- : 0 ml
Total A : 0 ml
>> Golongan Darah B:
B+ : 0 ml
B- : 0 ml
Total B : 0 ml
>> Golongan Darah AB:
AB+ : 0 ml
AB- : 0 ml
Total AB: 0 ml
>> Golongan Darah O:
O+ : 1000 ml
O- : 0 ml
Total O : 1000 ml
=====
Total keseluruhan darah yang terkumpul: 1000 ml
PS C:\Praktikum APD>
```

Gambar 4.1 Screenshot hasil output program

5. Langkah-langkah GIT

5.1 GIT Add

GIT Add adalah tahap kedua setelah membuat atau mengubah file. Dengan GIT Add maka git akan otomatis tahu file mana yang harus di simpan versi terbarunya.

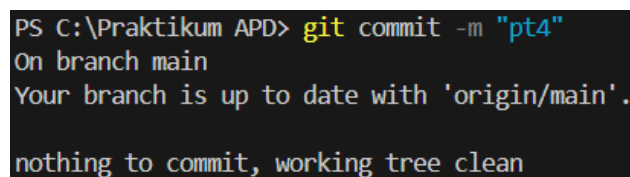


```
PS C:\Praktikum APD> git init
Reinitialized existing Git repository in C:/Praktikum APD/.git/
PS C:\Praktikum APD> git add .
```

Gambar 5.1 Screenshoot perintah git add

5.2 GIT Commit

Setelah GIT Add, maka perlu GIT Commit untuk memberi catatan pada filenya atau juga termasuk diberi pesan seperti "pt4".



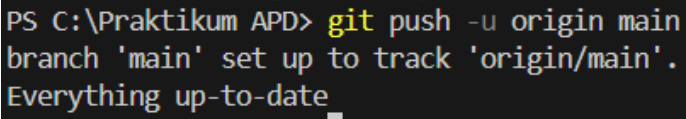
```
PS C:\Praktikum APD> git commit -m "pt4"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Gambar 5.2 Screenshoot perintah git commit

5.3 GIT Push

Tahap terakhir yaitu mendorong file repo ke github dengan git push agar dapat diakses secara online kapanpun dan dimanapun.

A screenshot of a Windows command prompt window. The text displayed is: PS C:\Praktikum APD> git push -u origin main
branch 'main' set up to track 'origin/main'.
Everything up-to-date
The text is white on a black background. The 'git' command is highlighted in yellow. A small white cursor is visible at the end of the last line.

```
PS C:\Praktikum APD> git push -u origin main
branch 'main' set up to track 'origin/main'.
Everything up-to-date
```

Gambar 5.3 Screenshoot perintah git push