

LAPORAN PRAKTIKUM
POSTTEST 4
ALGORITMA PEMROGRAMAN DASAR

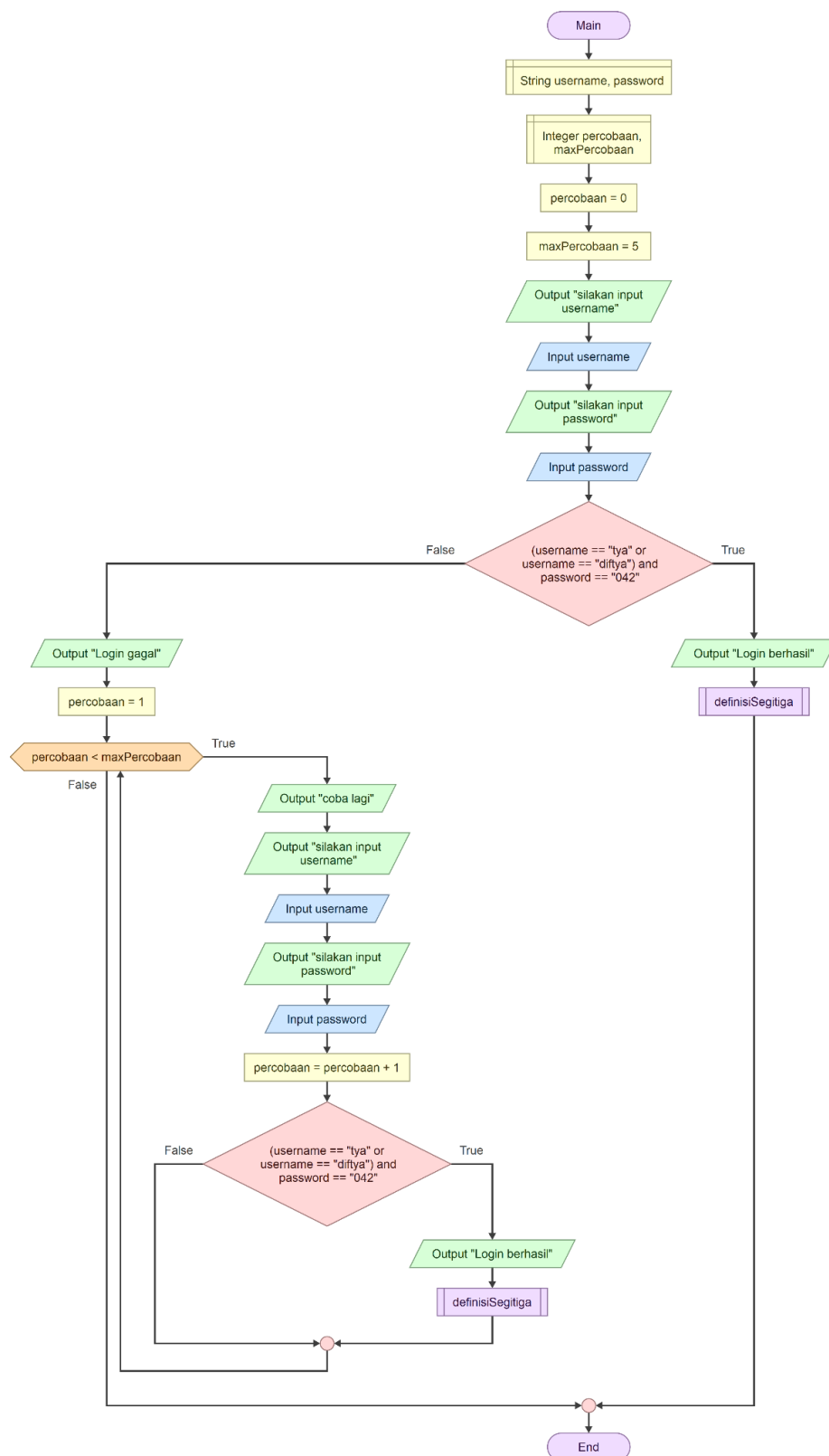


Disusun oleh:
Diftya Azzahra (2509106042)
Kelas (A2'25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

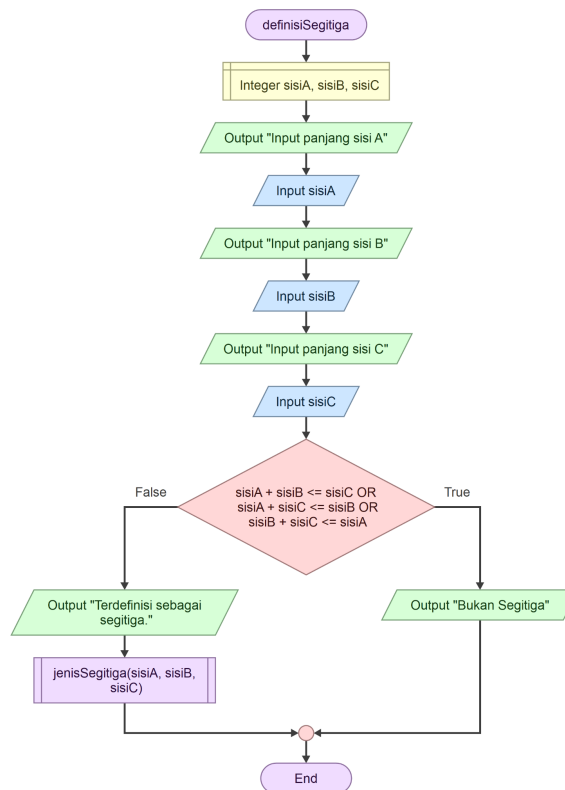
1. Flowchart

Berikut adalah flowchart yang saya buat



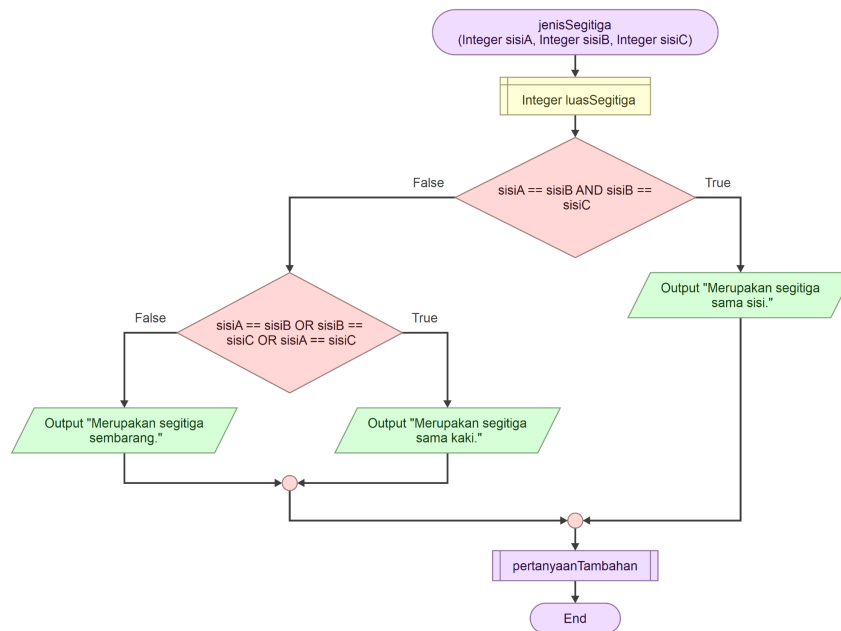
Gambar 1.1 Autentifikasi

1. **Start (Main):** Program dimulai
2. **Mendeklarasikan variabel:** username dan password adalah string, sedangkan percobaan dan maxPercobaan adalah integer.
3. **Assign:** menugaskan bahwa variabel percobaan bernilai 0 dan maksimal percobaan adalah 5. Hal ini dilakukan agar ketika perulangan terjadi, perhitungan dapat terjadi dengan baik.
4. **Output menampilkan pesan:** Silakan input username dan password. Bagian ini dibuat agar flowchart terlihat lebih menarik dan user tidak kebingungan kapan harus menginputkan username serta password.
5. **Input username :** Pada bagian ini, program akan menerima input berupa username yang diberikan oleh user.
6. **Decision:** Mengecek kondisi autentifikasi, apabila username bernilai tya atau diftya dan password bernilai 042 maka akan menghasilkan **True** dan menampilkan output berupa pesan “Login berhasil”, lalu user dapat langsung masuk ke bagian **menentukan definisi segitiga**. Jika salah, variabel **percobaan** akan bertambah 1. Setelah percobaan dinaikkan, **decision** kembali dibuat untuk mengecek kondisi apakah **percobaan < maxPercobaan**? jika benar, maka akan menampilkan output pesan berupa coba lagi dan user dapat kembali memasukkan username/password (kembali ke langkah input). Namun jika salah (**false**), maka percobaan habis dan program dihentikan.



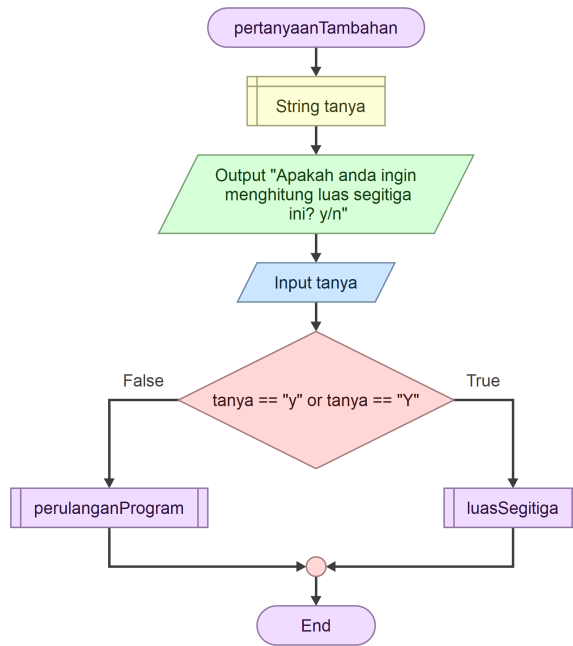
Gambar 1.2 Mendefinisikan segitiga

7. Pada bagian ini, pertama-tama dilakukan **deklarasi untuk beberapa variabel** seperti **sisiA**, **sisiB**, **sisiC**. Variabel ini digunakan untuk menampung nilai panjang sisi A, panjang sisi B, dan panjang sisi C.
8. Setelah variabel yang akan digunakan dideklarasikan, user perlu **menginput** atau memasukkan panjang masing-masing sisi agar bisa diidentifikasi lebih lanjut apakah value dari ketiga sisi memenuhi aturan segitiga, yaitu **jumlah dua sisi harus lebih besar dari sisi ketiga**. Jika setelah ketiga panjang sisi diinput namun tidak memenuhi aturan segitiga, program akan menampilkan **output** berupa pesan yaitu **“Bukan Segitiga”**. Namun jika jumlah dua sisi lebih besar dari sisi ketiga, program akan menampilkan **output** berupa pesan yaitu **“Terdefinisi sebagai segitiga.”** dan melanjutkan ke fungsi kedua yaitu **menentukan jenis segitiga**.



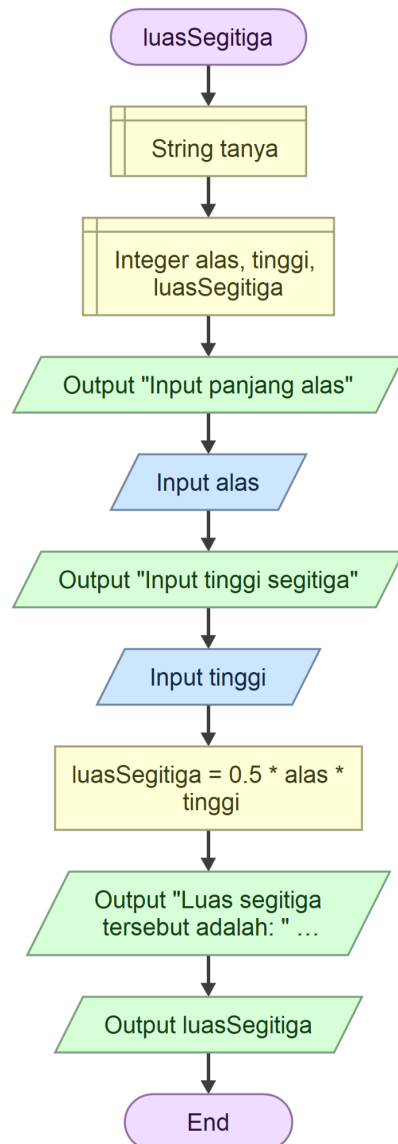
Gambar 1.3 Menentukan jenis segitiga

9. **Jika ketiga sisi sama panjangnya**, maka program akan mengidentifikasi segitiga tersebut sebagai **segitiga sama sisi**. Karena sesuai dengan aturan segitiga sama sisi yaitu semua sisinya sama panjangnya.
10. **Jika dua sisinya sama panjangnya**, maka program akan mengidentifikasi segitiga tersebut sebagai **segitiga sama kaki**. Karena sesuai dengan aturan segitiga sama kaki.
11. Karena kita membuat program ini untuk mengidentifikasi segitiga berdasarkan panjang sisi, jika ketiga sisi segitiga yang diinput user tidak memenuhi kedua kondisi tersebut, maka akan diidentifikasi sebagai **segitiga sembarang**. Dengan catatan bahwa ketiga panjang sisi tersebut **tetap mengikuti aturan segitiga**.
12. Setelah itu, masuk ke bagian untuk menghitung **luas segitiga**. Untuk selanjutnya, akan masuk ke fungsi **pertanyaan tambahan**.



Gambar 1.4 Pertanyaan tambahan

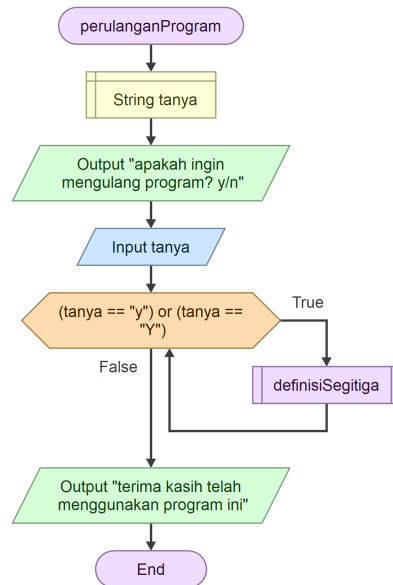
13. Masuk ke fitur menghitung luas, dilakukan pendeklarasian variabel **tanya** sebagai **string** untuk menampung variabel yang akan diinput user yaitu **y/n**. Variabel ini dibuat dengan maksud untuk menampung jawaban dari pertanyaan tambahan (jika di kodingan).
14. Setelah itu Saya membuat output berupa pesan “Apakah anda ingin menghitung luas segitiga ini? y/n” agar program terlihat lebih rapi dan menarik. Lalu program akan meminta **input** jawaban dari user. Jika jawaban **n** atau **N**, maka program akan berhenti dan menampilkan output berupa pesan “**Terima kasih telah menggunakan program ini**”.
15. Jika jawaban **y** atau **Y**, maka program akan lanjut ke fungsi terakhir yaitu fungsi menghitung **luas segitiga**.



Gambar 1.5 Menentukan luas segitiga

16. Setelah ketiga panjang sisi diinput pada fungsi utama (main) dan terdefinisi sebagai segitiga, dilakukan deklarasi variabel seperti alas, tinggi, dan luas segitiga sebagai integer. Ketiga variabel ini diperlukan karena rumus untuk mencari luas segitiga adalah $\frac{1}{2} \times \text{alas} \times \text{tinggi}$ atau $0.5 \times \text{alas} \times \text{tinggi}$.
17. Setelah deklarasi variabel, user bisa mulai menginput dari panjang alas kemudian tinggi segitiga. Kemudian ditambahkan assign yang berisi rumus luas segitiga agar program dapat menghitung berapa hasil luas segitiga dari panjang alas dan tinggi segitiga yang diberikan.

18. Program akan menghitung luas segitiga dan menampilkan output berupa hasil perhitungan luas yang ditambah dengan pesan agar lebih menarik. Output berupa "Luas segitiga tersebut adalah: ...



Gambar 1.6 Perulangan program

19. **Mendeklarasi variabel:** tanya adalah **String**.

20. Output: Berupa pesan apakah ingin mengulang program? y/n.

21. Input: User menambahkan input jawaban berupa y/n yang berarti *yes or no*.

22. **While:** **tanya == "y" atau tanya == "Y"**, jika benar panggil bagian **definisiSegitiga**. Namun jika salah atau false menampilkan **Output** berupa pesan "terima kasih telah menggunakan program ini" dan program pun berhenti

2. Deskripsi Singkat Program

Program ini bermanfaat untuk mengidentifikasi jenis segitiga berdasarkan panjang ketiga sisinya serta dapat menghitung luasnya, yang juga telah dilengkapi dengan sistem login.

3. Source Code

```
print("Program untuk menentukan jenis segitiga dan menghitung luasnya")
print("="*62)
max_percobaan = 5
```



```

percobaan = 0

while(percobaan < max_percobaan):
    username = str(input("Silakan masukkan username Anda: "))
    password = str(input("Silakan masukkan password Anda: "))
    if (username == "tya" or username == "diftya") and password == "042":
        print("Login Berhasil")
        break
    else:
        print("Login Gagal")
        print("Coba lagi")
        percobaan += 1
    if percobaan >= max_percobaan:
        print("Maaf, sisa percobaan Anda sudah habis.")
        exit()

tanya = "y"

while tanya == "y" or tanya == "Y":
    sisi_a = int(input("Input panjang sisi A: "))
    sisi_b = int(input("Input panjang sisi B: "))
    sisi_c = int(input("Input panjang sisi C: "))

    if sisi_a + sisi_b <= sisi_c or sisi_a + sisi_c <= sisi_b or sisi_b +
sisi_c <= sisi_a:
        print("Bukan Segitiga")
    else:
        print("Terdefinisi sebagai segitiga.")
        if sisi_a == sisi_b == sisi_c:
            print("Merupakan segitiga sama sisi.")
        elif sisi_a == sisi_b or sisi_b == sisi_c or sisi_a == sisi_c:
            print("Merupakan segitiga sama kaki.")
        else:
            print("Merupakan segitiga sembarang.")

    pertanyaan_tambahan = input("Apakah anda ingin menghitung luas
segitiga ini? (y/n) ")
    if pertanyaan_tambahan == "n" or pertanyaan_tambahan == "N":
        print("Terima kasih telah menggunakan program ini")

    elif pertanyaan_tambahan == "y" or pertanyaan_tambahan == "Y":

```

```

    alas = float(input("Input panjang alas : "))
    tinggi = float(input("Input tinggi : "))
    luas_segitiga = 0.5 * alas * tinggi
    print("Luas segitiga adalah", luas_segitiga)

tanya = "x"
while tanya not in ["y", "Y", "n", "N"]:
    tanya = str(input("Apakah anda ingin melanjutkan program? (y/n)
"))

if tanya == "n" or tanya == "N":
    tanya = "n"
    print("Terima kasih telah menggunakan program ini")
    exit()
elif tanya == "y" or tanya == "Y":
    tanya = "y"
    print("Program Berlanjut")
    break
else:
    print("Input tidak valid, program berhenti")
    print("Input tidak valid, silakan masukkan 'y' atau 'n'")
    continue

```

4. Hasil Output

Karena ini merupakan kelanjutan dari postest sebelumnya, pada bagian output saya hanya akan melampirkan bagian baru yaitu bagian sistem login dan perulangan.

```

Program untuk menentukan jenis segitiga dan menghitung luasnya
=====
Silakan masukkan username Anda: tya
Silakan masukkan password Anda: 042
Login Berhasil
Input panjang sisi A: 20
Input panjang sisi B: 20
Input panjang sisi C: 20
Terdefinisi sebagai segitiga.
Merupakan segitiga sama sisi.
Apakah anda ingin menghitung luas segitiga ini? (y/n) n
Terima kasih telah menggunakan program ini
Apakah anda ingin melanjutkan program? (y/n) y
Program Berlanjut
Input panjang sisi A: █

```

Gambar 4.1 Login berhasil beserta perulangan (python)

```

Program untuk menentukan jenis segitiga dan menghitung luasnya
=====
Silakan masukkan username Anda: syarafina
Silakan masukkan password Anda: 016
Login Gagal
Coba lagi
Silakan masukkan username Anda: fina
Silakan masukkan password Anda: 016
Login Gagal
Coba lagi
Silakan masukkan username Anda: fina
Silakan masukkan password Anda: 016
Login Gagal
Coba lagi
Silakan masukkan username Anda: fina
Silakan masukkan password Anda: 016
Login Gagal
Coba lagi
Silakan masukkan username Anda: fina
Silakan masukkan password Anda: 042
Login Gagal
Coba lagi
Maaf, sisa percobaan Anda sudah habis.

```

Gambar 4.2 Login gagal (python)

Main			
username	password	percobaan	maxPercobaan
syarafina	016	5	5

Gambar 4.3 Login gagal (flowchart)

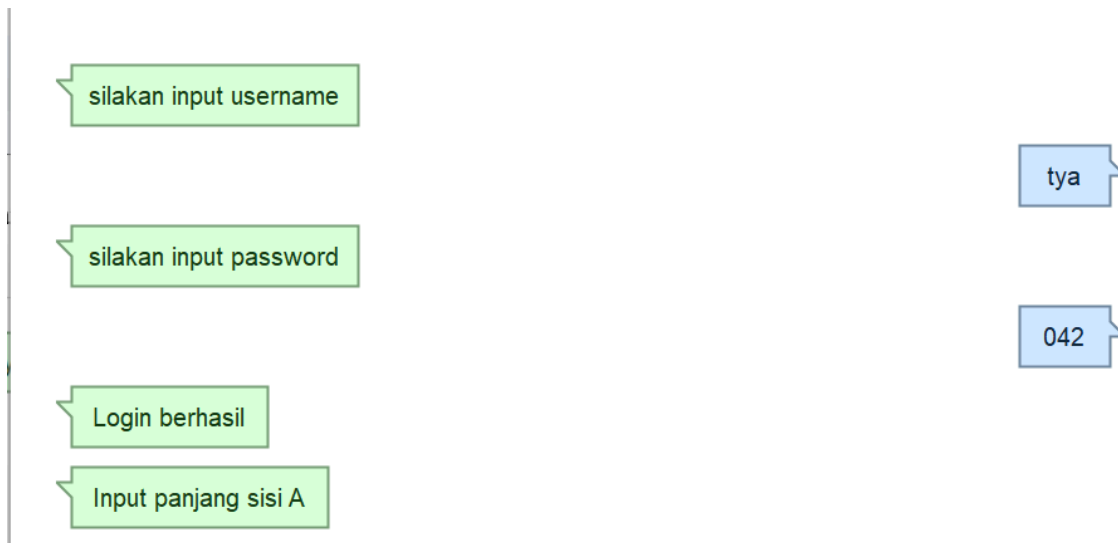
apakah ingin mengulang program? y/n

y

Input panjang sisi A

↑

Gambar 4.4 Perulangan (flowchart)



Gambar 4.5 Login berhasil (flowchart)

5. Langkah-langkah GIT

5.1 GIT Init

```

C:\Users\ASUS\Documents\Kuliah\Praktikum-APD\post-test\post-test-apd-4>git init
Initialized empty Git repository in C:/Users/ASUS/Documents/Kuliah/Praktikum-APD/post-test/post-test-apd-4/.git/
  
```

5.2 GIT Add

```

C:\Users\ASUS\Documents\Kuliah\Praktikum-APD\post-test\post-test-apd-4>git add .
C:\Users\ASUS\Documents\Kuliah\Praktikum-APD\post-test\post-test-apd-4>git commit -m "added this file"
[main 1283dec] added this file
3 files changed, 107 insertions(+)
create mode 100644 post-test-apd-4/.gitmodules
create mode 100644 post-test-apd-4/2509106042-Diftya_Azzahra-PT4.py
create mode 100644 post-test-apd-4/coba-aja.py
C:\Users\ASUS\Documents\Kuliah\Praktikum-APD\post-test\post-test-apd-4>git add remote origin https://github.com/HikaruYui/Praktikum-APD.git
fatal: pathspec 'remote' did not match any files
  
```

5.3 GIT Commit

```

C:\Users\ASUS\Documents\Kuliah\Praktikum-APD\post-test\post-test-apd-4>git add .
C:\Users\ASUS\Documents\Kuliah\Praktikum-APD\post-test\post-test-apd-4>git commit -m "added this file"
[main 1283dec] added this file
3 files changed, 107 insertions(+)
create mode 100644 post-test-apd-4/.gitmodules
create mode 100644 post-test-apd-4/2509106042-Diftya_Azzahra-PT4.py
create mode 100644 post-test-apd-4/coba-aja.py
C:\Users\ASUS\Documents\Kuliah\Praktikum-APD\post-test\post-test-apd-4>git add remote origin https://github.com/HikaruYui/Praktikum-APD.git
fatal: pathspec 'remote' did not match any files
  
```

5.4 GIT Remote

```
C:\Users\ASUS\Documents\Kuliah\Praktikum-APD\post-test\post-test-apd-4>git remote add origin https://github.com/HikaruYui/Praktikum-APD.git
error: remote origin already exists.

C:\Users\ASUS\Documents\Kuliah\Praktikum-APD\post-test\post-test-apd-4>git remote -v
origin  https://github.com/HikaruYui/Praktikum-APD.git (fetch)
origin  https://github.com/HikaruYui/Praktikum-APD.git (push)
```

5.5 GIT Push

```
C:\Users\ASUS\Documents\Kuliah\Praktikum-APD\post-test>git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 405 bytes | 202.00 KiB/s, done.
Total 3 (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/HikaruYui/Praktikum-APD.git
   232632e..ad45d2d  main -> main
```