

LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN DASAR

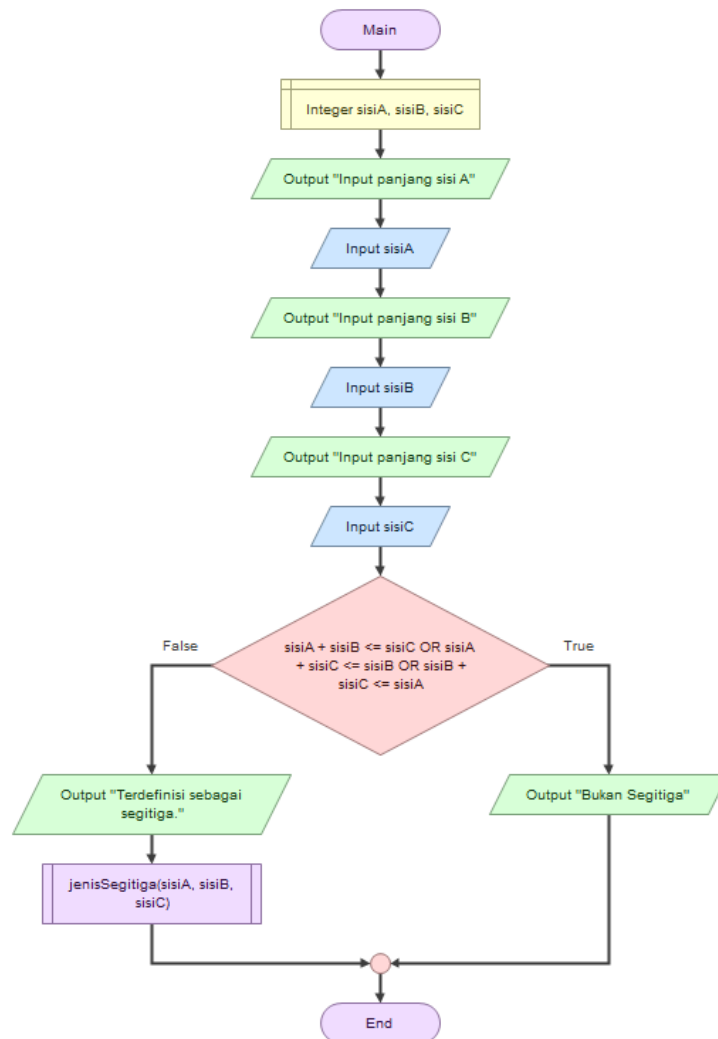


Disusun oleh:
Diftya Azzahra (2509106042)
Kelas (A2'25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart

Berikut adalah flowchart yang saya buat

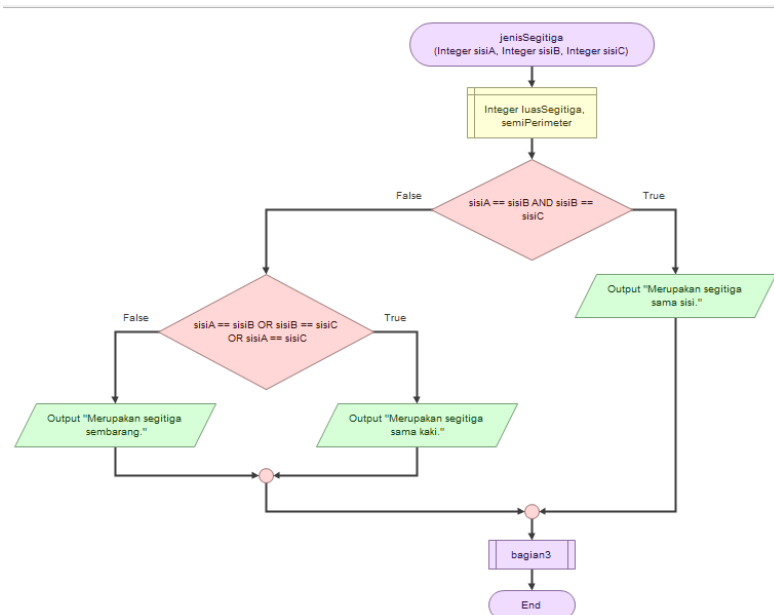


Gambar 1.1

Flowchart **Bagian Pertama**

1. 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.
2. 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**

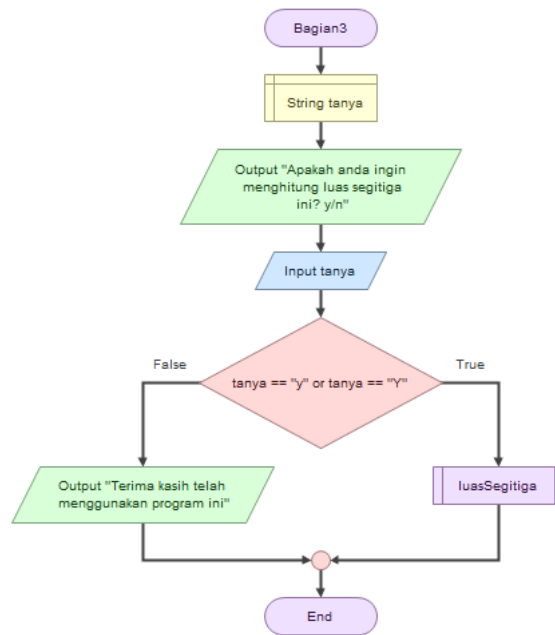
Segitiga”. Namun jika **jumlah dua sisi lebih besar dari sisi ketiga**, program akan menampilkan output berupa pesan yaitu **“Terdefinisi sebagai segitiga.”** dan melanjutkan ke fungsi kedua yaitu menentukan jenis segitiga.



Gambar 1.2

Fungsi Kedua (Menentukan Jenis Segitiga)

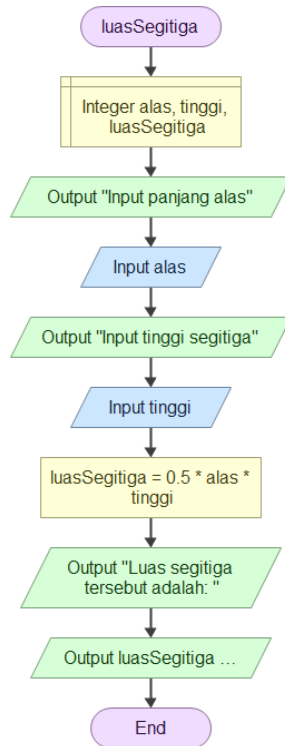
3. **Jika ketiga sisi sama panjangnya**, maka program akan mengidentifikasi segitiga tersebut sebagai **segitiga sama sisi**. Karena sesuai dengan aturan segitiga sama sisi yaitu **semua sisinya sama panjangnya**.
4. **Jika dua sisinya sama panjangnya**, maka program akan mengidentifikasi segitiga tersebut sebagai **segitiga sama kaki**. Karena sesuai dengan aturan segitiga sama kaki.
5. 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**.
6. Setelah itu, masuk ke poin nilai tambah yaitu untuk menghitung luas segitiga. Untuk selanjutnya, akan masuk ke fungsi **bagian3**.



Gambar 1.3

Fungsi Ketiga (**bagian3**)

7. Sebelum 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).
8. 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.
9. Jika jawaban **n** atau **N**, maka program akan berhenti dan menampilkan output berupa pesan “**Terima kasih telah menggunakan program ini**”.
10. Jika jawaban **y** atau **Y**, maka program akan lanjut ke fungsi terakhir yaitu fungsi menghitung luas segitiga.



Gambar 1.4
Fungsi Keempat (**Menghitung luas segitiga**)

11. 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 $1/2 \times \text{alas} \times \text{tinggi}$ **atau** $0.5 \times \text{alas} \times \text{tinggi}$.
12. Setelah deklarasi variabel, user bisa mulai menginput dari panjang alas kemudian tinggi segitiga.
13. Kemudian ditambahkan **assign** yang berisi rumus luas segitiga agar program dapat menghitung berapa hasil luas segitiga dari panjang alas dan tinggi segitiga yang diberikan.
14. 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: ..."

2. Deskripsi Singkat Program

Program ini dibuat untuk mengidentifikasi jenis segitiga berdasarkan panjang ketiga sisinya dan menghitung luas segitiga menggunakan rumus $\frac{1}{2}$ (atau 0.5) x alas x tinggi.

3. Source Code

A. Fitur Identifikasi Segitiga

Fitur ini digunakan untuk mengidentifikasi jenis segitiga.

Source Code:

```
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.")
```

B. Fitur Menghitung Luas Segitiga

```
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")
if pertanyaan_tambahan == "y" or pertanyaan_tambahan == "Y":
    alas = int(input("Input panjang alas : "))
    tinggi = int(input("Input tinggi : "))
    luas_segitiga = 0.5 * alas * tinggi
    print("Luas segitiga adalah", luas_segitiga)
```

4. Hasil Output

```
PS C:\Users\ASUS> & C:/Users/ASUS/AppData/Local/Programs/Python/Python313/python.exe c:/Users/ASUS/Documents/Kuliah/Praktikum-APD/post-test/post-test-apd-2/2509106042-Diftya_Azzahra-PT3.py
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) █
```

Gambar 4.1
Output **segitiga sama sisi** pada kodingan python

```
PS C:\Users\ASUS> & C:/Users/ASUS/AppData/Local/Programs/Python/Python313/python.exe c:/Users/ASUS/Documents/Kuliah/Praktikum-APD/post-test/post-test-apd-2/2509106042-Diftya_Azzahra-PT3.py
Input panjang sisi A: 25
Input panjang sisi B: 25
Input panjang sisi C: 10
Terdefinisi sebagai segitiga.
Merupakan segitiga sama kaki.
Apakah anda ingin menghitung luas segitiga ini? (y/n) █
```

Gambar 4.2
Output **segitiga sama kaki** pada kodingan python

```
PS C:\Users\ASUS> & C:/Users/ASUS/AppData/Local/Programs/Python/Python313/python.exe c:/Users/ASUS/Documents/Kuliah/Praktikum-APD/post-test/post-test-apd-3/2509106042-Diftya_Azzahra-PT3.py
Input panjang sisi A: 20
Input panjang sisi B: 15
Input panjang sisi C: 10
Terdefinisi sebagai segitiga.
Merupakan segitiga sembarang.
Apakah anda ingin menghitung luas segitiga ini? (y/n) █
```

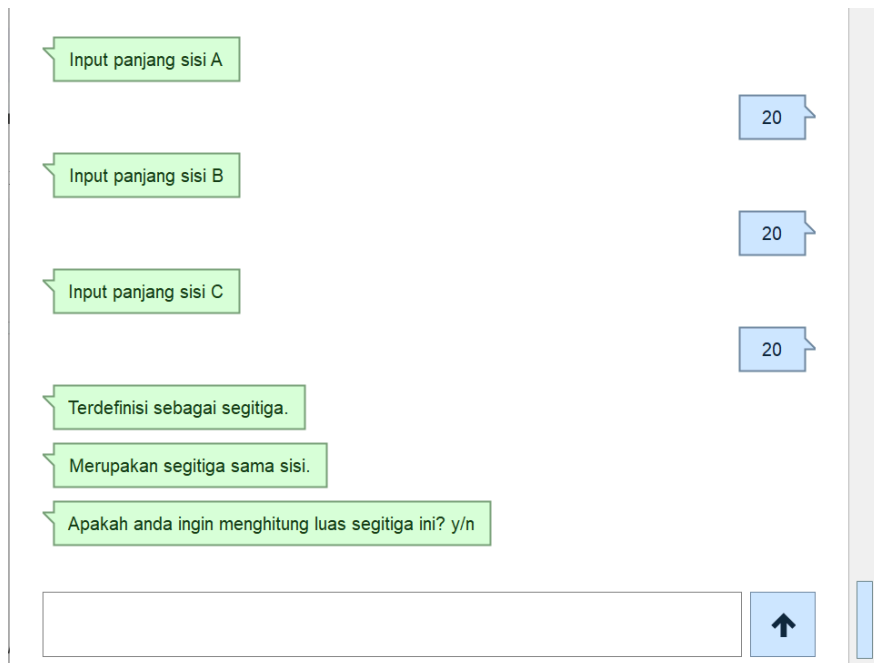
Gambar 4.3
Output **segitiga sembarang** pada kodingan python

```
PS C:\Users\ASUS> & C:/Users/ASUS/AppData/Local/Programs/Python/Python313/python.exe c:/Users/ASUS/Documents/Kuliah/Praktikum-APD/post-test/post-test-apd-3/2509106042-Diftya_Azzahra-PT3.py
Input panjang sisi A: 30
Input panjang sisi B: 50
Input panjang sisi C: 20
Bukan Segitiga
PS C:\Users\ASUS> █
```

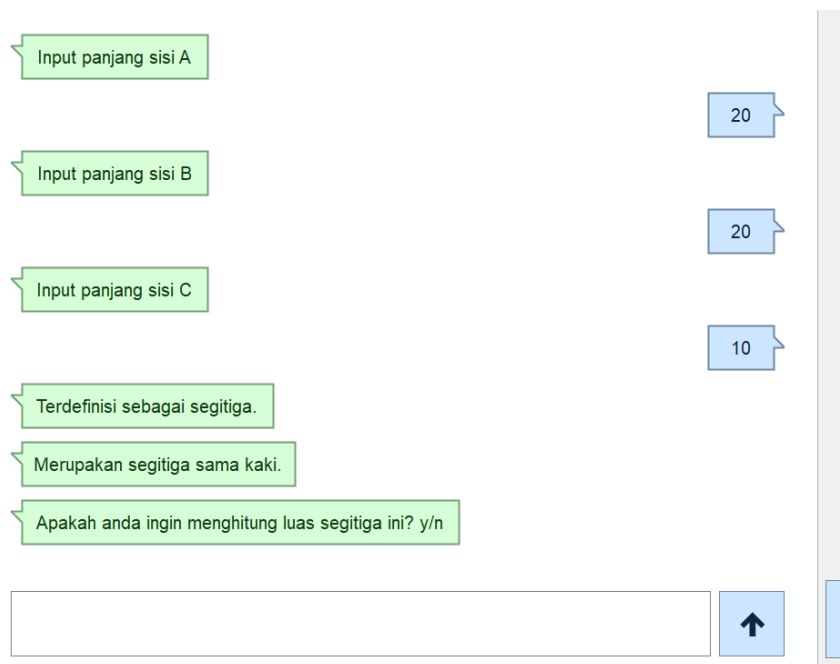
Gambar 4.4
Output jika tidak terdefinisi segitiga

```
PS C:\Users\ASUS> & C:/Users/ASUS/AppData/Local/Programs/Python/Python313/python.exe c:/Users/ASUS/Documents/Kuliah/Praktikum-APD/post-test/post-test-apd-3/2509106042-Diftya_Azzahra-PT3.py
Input panjang sisi A: 20
Input panjang sisi B: 20
Input panjang sisi C: 10
Terdefinisi sebagai segitiga.
Merupakan segitiga sama kaki.
Apakah anda ingin menghitung luas segitiga ini? (y/n) y
Terima kasih telah menggunakan program ini
Input panjang alas : 12
Input tinggi : 12
Luas segitiga adalah 72.0
PS C:\Users\ASUS> █
```

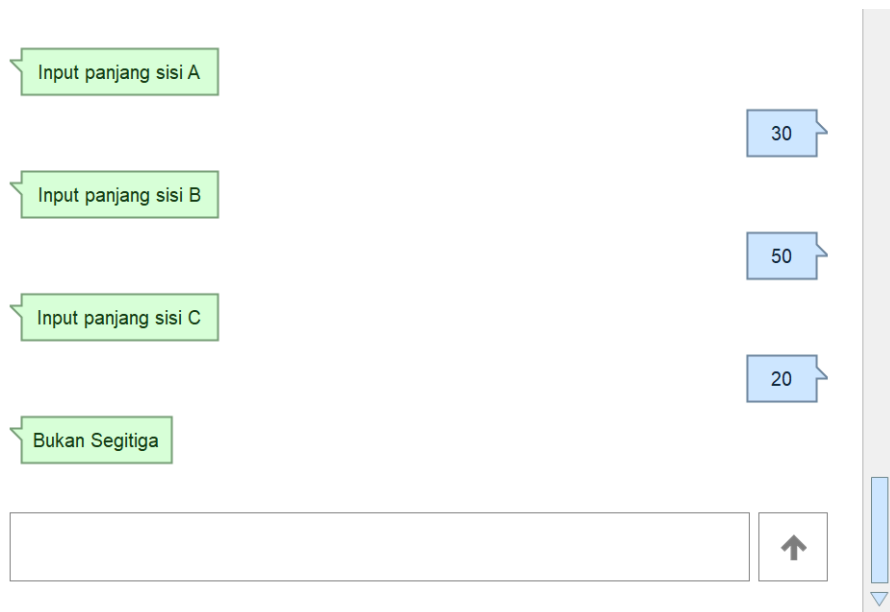
Gambar 4.5
Output program menghitung luas



Gambar 4.6
Output segitiga sama sisi pada flowchart

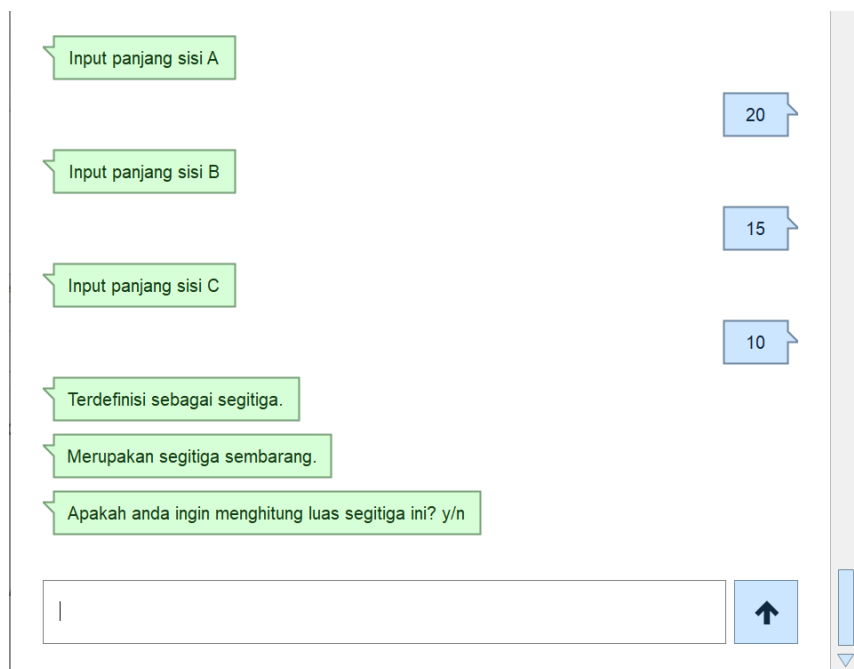


Gambar 4.7
Output segitiga sama kaki pada flowchart



Gambar 4.8

Output Bukan Segitiga pada flowchart



Gambar 4.9

Output segitiga sembarang pada flowchart

5. Langkah-langkah GIT

Berikut adalah langkah-langkah git yang saya gunakan:

5.1 GIT Init

```
C:\Users\ASUS\Documents\Kuliah\Praktikum-APD\post-test\post-test-apd-3>git init
Initialized empty Git repository in C:/Users/ASUS/Documents/Kuliah/Praktikum-APD/post-test/post-test-apd-3/.git/
```

5.2 GIT Add

```
C:\Users\ASUS\Documents\Kuliah\Praktikum-APD\post-test\post-test-apd-3>git add .
```

5.3 GIT Commit

```
PS C:\Users\ASUS\Documents\Kuliah\Praktikum-APD\post-test\post-test-apd-3> git commit -m "added 2509106042-Diftya_Azzahra-PT3
.py file"
[main 033219f] added 2509106042-Diftya_Azzahra-PT3.py file
1 file changed, 24 insertions(+)
create mode 100644 2509106042-Diftya_Azzahra-PT3.py
```

5.4 GIT Remote

```
C:\Users\ASUS\Documents\Kuliah\Praktikum-APD\post-test\post-test-apd-3>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-3>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\post-test-apd-3>git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 651 bytes | 325.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/HikaruYui/Praktikum-APD.git
   af96fa1..8897262  main -> main
branch 'main' set up to track 'origin/main'.
```

5.6 Github

The screenshot displays the GitHub web interface for a repository named "Praktikum-APD" owned by "HikaruYui". The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The left sidebar shows the file tree structure, with the "post-testt-apd-3" folder selected. The main content area shows the commit history for the selected folder, listing two commits by HikaruYui: one adding a PDF file 5 minutes ago and another adding a Python file 2 minutes ago.

Name	Last commit message	Last commit date
..		
2509106042-Diftya_Azzahra-PT-3.pdf	added: this pdf	5 minutes ago
2509106042-Diftya_Azzahra-PT3.py	added: 2509106042-Diftya_Azzahra-PT3.py file	2 minutes ago