

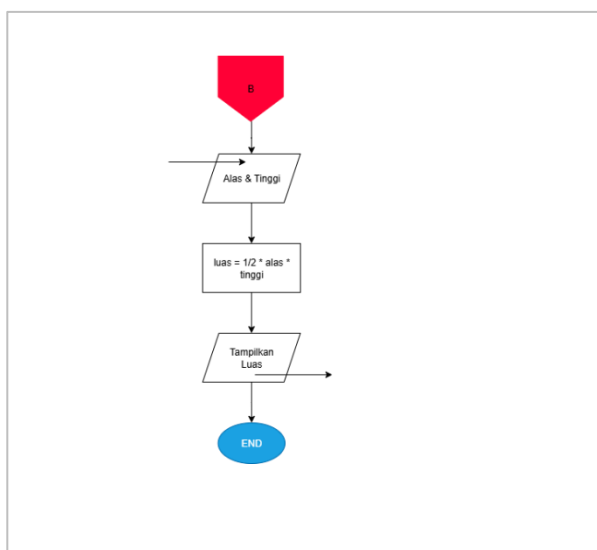
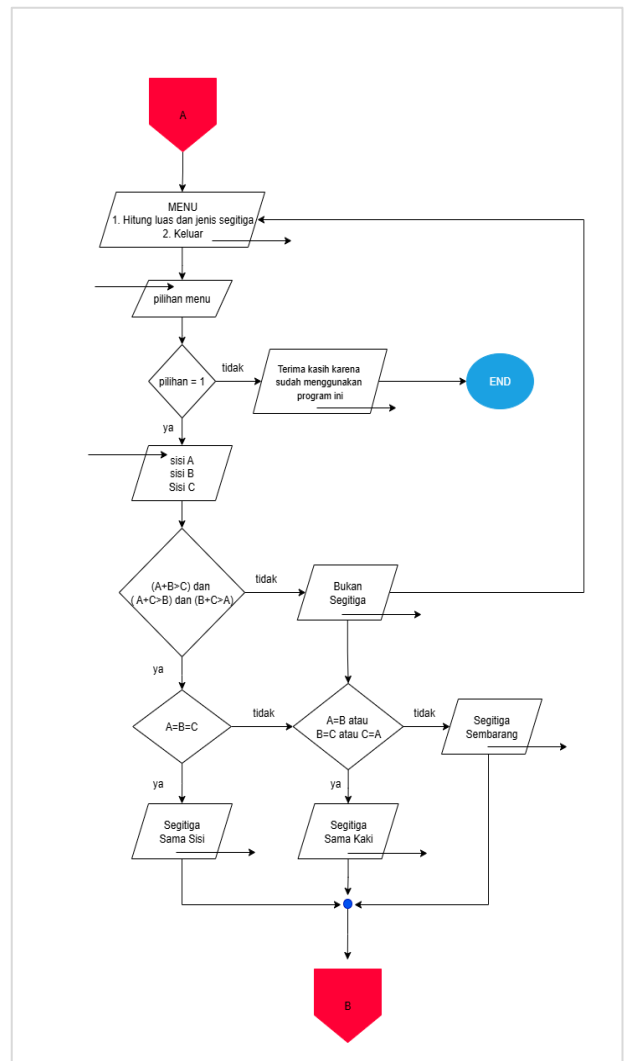
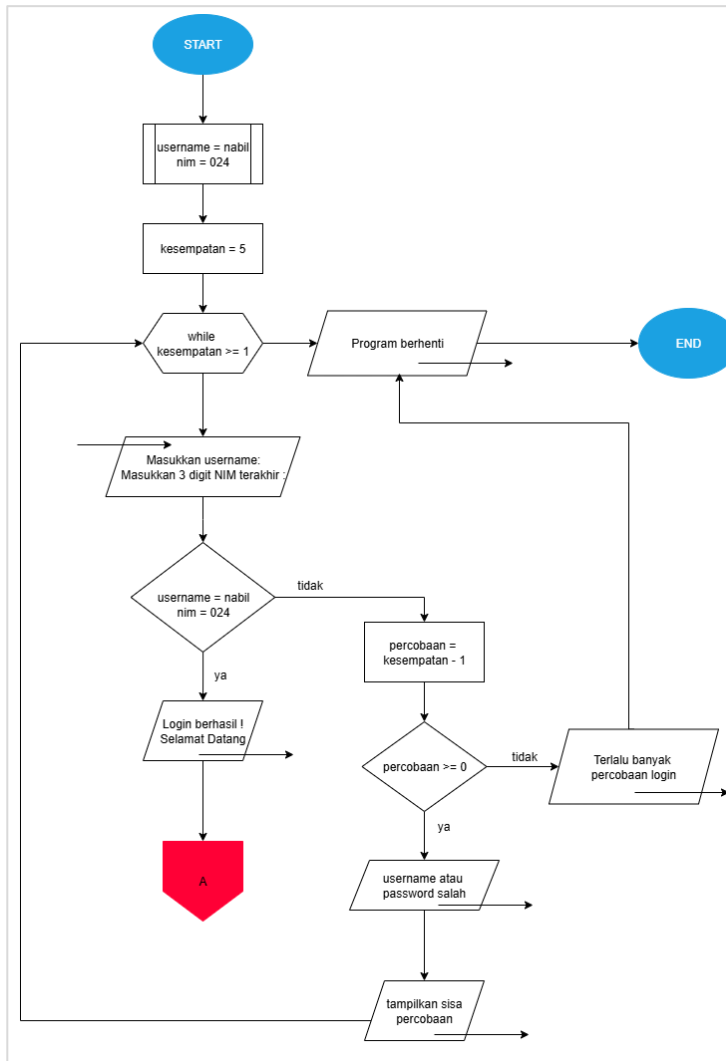
**LAPORAN PRAKTIKUM**  
**POSTTEST (4)**  
**ALGORITMA PEMROGRAMAN DASAR**



**Disusun oleh:**  
**Ferly Ahmad Nabil (2509106024)**  
**Kelas (A2 '25)**

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

# 1. Flowchart



## 2. Deskripsi Singkat Program

Program ini dibuat untuk menentukan jenis segitiga berdasarkan panjang sisi-sisinya, serta menghitung luas segitiga apabila segitiga tersebut terdefinisi dengan benar. Selain itu, program juga dilengkapi dengan fitur autentikasi login, di mana pengguna harus memasukkan username dan password yang benar sebelum bisa menjalankan program utama.

### A. Login / Autentikasi

- Username yang benar: nabil
- Password yang benar: 024
- Jika *username* atau *password* salah, pengguna diberi kesempatan hingga 5 kali percobaan.
- Jika sudah salah sebanyak 5 kali, program akan berhenti secara otomatis.

### B. Menu Utama

- Setelah login berhasil, program menampilkan dua pilihan:
  1. Menentukan jenis dan menghitung luas segitiga.
  2. Keluar dari program.

### C. Penentuan Jenis Segitiga

- Program meminta tiga nilai sisi segitiga (A, B, dan C).
- Mengecek apakah memenuhi aturan segitiga:  $(A + B > C)$  dan  $(A + C > B)$  dan  $(B + C > A)$
- Jika memenuhi aturan :
  - Ketiga sisi sama → Segitiga Sama Sisi
  - Dua sisi sama → Segitiga Sama Kaki
  - Semua sisi berbeda → Segitiga Sembarang
- Jika tidak memenuhi aturan → tampil pesan “Bukan segitiga”

### D. Menghitung Luas Segitiga

- Jika segitiga valid, program akan meminta input alas dan tinggi.
- Rumus luas yang digunakan :  $\text{luas} = \frac{1}{2} \times \text{alas} \times \text{tinggi}$
- Hasil luas akan ditampilkan di layar.

### E. Perulangan Menu

- Setelah selesai, program akan kembali ke menu utama.
- Program hanya berhenti jika pengguna memilih opsi keluar.

### F. Membersihkan Layar (*Clear Screen*)

- Program menggunakan modul os dengan perintah: `os.system('cls')`
- Agar tampilan terminal tetap rapi setiap kali berpindah tampilan

### 3. Source Code

#### A. Program Autentikasi Login

Code ini bertujuan untuk menerapkan konsep autentikasi sederhana menggunakan *input username* dan *password*

#### B. Menentukan Jenis dan Luas

Selain untuk Login code ini berfungsi juga untuk menentukan jenis dan luas segitiga dengan memasukkan panjang 3 sisi; sisi A, sisi B, sisi C

#### Source Code:

```
import os

username_benar = "nabil"
password_benar = "024"

kesempatan = 0
login = False

while kesempatan < 5:
    os.system('cls' )
    print("=== LOGIN PROGRAM JENIS DAN LUAS SEGITIGA ===")
    user = input("Masukkan Username : ")
    pw = input("Masukkan 3 digit NIM terakhir : ")

    if user.lower() == username_benar and pw == password_benar:
        print("\nLogin berhasil! Selamat datang,", user)
        login = True
        input("Tekan Enter untuk lanjut...")
        os.system('cls' )
        break
    else:
        kesempatan += 1
        print("\nUsername atau Password salah!")
        print("Sisa percobaan:", 5 - kesempatan)
        input("Tekan Enter untuk coba lagi...")

if not login:
    print("\nTerlalu banyak percobaan login. Program berhenti.")
    exit()
```

```

while True:
    print("=== MENU ===")
    print("1. Tentukan Jenis Segitiga & Hitung Luas")
    print("2. Keluar")
    pilih = input("Pilih menu (1/2): ")

    if pilih == "1":
        os.system('cls')
        print("=== MENENTUKAN JENIS SEGITIGA ===")

        a = int(input("Masukkan sisi A: "))
        b = int(input("Masukkan sisi B: "))
        c = int(input("Masukkan sisi C: "))

        if a + b > c and a + c > b and b + c > a:
            if a == b == c:
                print("\nJenis segitiga: Segitiga Sama Sisi")
            elif a == b or a == c or b == c:
                print("\nJenis segitiga: Segitiga Sama Kaki")
            else:
                print("\nJenis segitiga: Segitiga Sembarang")

        print("\n--- Hitung Luas Segitiga ---")
        alas = float(input("Masukkan alas: "))
        tinggi = float(input("Masukkan tinggi: "))
        luas = (0.5 * alas * tinggi)
        print("Luas segitiga adalah:", luas)
    else:
        print("\nBukan segitiga! (Tidak memenuhi aturan segitiga)")

        input("\nTekan Enter untuk kembali ke menu...")
        os.system('cls' )

    elif pilih == "2":
        os.system('cls' )
        print("Terima kasih sudah menggunakan program ini! #GGMU 2-0")
        break

    elif pilih == "2":
        clear()
        print("Terima kasih sudah datang bro/sis #GGMU!")
        break

```

```
else:
    print("\nPilihan tidak tersedia!")
    input("Tekan Enter untuk coba lagi...")
    clear()
```

#### 4. Hasil Output

```
=== LOGIN PROGRAM JENIS DAN LUAS SEGITIGA ===
Masukkan Username : █
```

Gambar 4.1  
Tampilan untuk *Login* masukkan Username dan NIM

```
=== LOGIN PROGRAM JENIS DAN LUAS SEGITIGA ===
Masukkan Username : nabil
Masukkan 3 digit NIM terakhir : 024

Login berhasil! Selamat datang, nabil
Tekan Enter untuk lanjut... █
```

Gambar 4.2  
Hasil *Login* berhasil setelah memasukkan *Username* dan NIM dengan benar

```
=== LOGIN PROGRAM JENIS DAN LUAS SEGITIGA ===
Masukkan Username : nabil
Masukkan 3 digit NIM terakhir : 123

Username atau Password salah!
Sisa percobaan: 3
Tekan Enter untuk coba lagi... █
```

Gambar 4.3  
Tampilan *Login* gagal beserta sisa kesempatan percobaan

```
=== MENU PROGRAM ===  
1. Tentukan Jenis Segitiga & Hitung Luas  
2. Keluar Program  
Pilih menu (1/2): █
```

Gambar 4.4  
Tampilan setelah *enter*

```
=== MENENTUKAN JENIS SEGITIGA ===  
Masukkan sisi A: 12  
Masukkan sisi B: 12  
Masukkan sisi C: █
```

Gambar 4.5  
Tampilan Menu 1 untuk menentukan Jenis Segitiga

```
=== MENENTUKAN JENIS SEGITIGA ===  
Masukkan sisi A: 12  
Masukkan sisi B: 12  
Masukkan sisi C: 12  
  
Jenis segitiga: Segitiga Sama Sisi
```

Gambar 4.6  
Tampilan haasl Segitiga Sama Sisi

```
=== MENENTUKAN JENIS SEGITIGA ===  
Masukkan sisi A: 12  
Masukkan sisi B: 12  
Masukkan sisi C: 13  
  
Jenis segitiga: Segitiga Sama Kaki
```

Gambar 4.7  
Tampilan hasil Segitiga Sama Kaki

```
=== MENENTUKAN JENIS SEGITIGA ===  
Masukkan sisi A: 12  
Masukkan sisi B: 13  
Masukkan sisi C: 14  
  
Jenis segitiga: Segitiga Sembarang
```

Gambar 4.8  
Tampilan hasil Segitiga Sembarang

```
=== MENENTUKAN JENIS SEGITIGA ===  
Masukkan sisi A: 12  
Masukkan sisi B: 14  
Masukkan sisi C: -2  
  
Bukan segitiga! (Tidak memenuhi aturan segitiga)
```

Gambar 4.9  
Tampilan hasil Bukan Segitiga

```
=== MENENTUKAN JENIS SEGITIGA ===  
Masukkan sisi A: 12  
Masukkan sisi B: 12  
Masukkan sisi C: 13  
  
Jenis segitiga: Segitiga Sama Kaki  
  
--- Hitung Luas Segitiga ---  
Masukkan alas: 13  
Masukkan tinggi: 12  
Luas segitiga adalah: 78.0
```

Gambar 4.10  
Tampilan luas segitiga setelah terdefinisi sebagai segitiga

```
Terima kasih sudah menggunakan program ini #GGMU 2-0  
PS D:\(KULIAH)\SEMESTER 1\praktikum Algoritma Pemograman Dasar\praktikum-apd> █
```

Gambar 4.11  
Tampilan menu 2 untuk mengakhiri program



```

=== MENU PROGRAM ===
1. Tentukan Jenis Segitiga & Hitung Luas
2. Keluar Program
Pilih menu (1/2): 3

Pilihan tidak tersedia!
Tekan Enter untuk coba lagi...

```

Gambar 4.12  
Tampilan ketika yang di *input* ada dalam menu

## 5. Langkah-langkah GIT

### 5.1 GIT Init

```

Asus@DESKTOP-8GBFI02 MINGW64 /d/(KULIAH)/SEMESTER 1/praktikum Algoritma Pemogram
an Dasar (main)
$ git init
Reinitialized existing Git repository in D:/(KULIAH)/SEMESTER 1/praktikum Algori
tma Pemogramman Dasar/.git/

```

Git init agar bisa *track progress* yang ada pada folder tersebut

### 5.2 GIT Add

```

Asus@DESKTOP-8GBFI02 MINGW64 /d/(KULIAH)/SEMESTER 1/praktikum Algoritma Pemogram
an Dasar (main)
$ git add .

```

Git Add menambah semua file yang ada didalam folder sementara sebelum file dikomit dalam repository

### 5.3 GIT Commit

```

Asus@DESKTOP-8GBFI02 MINGW64 /d/(KULIAH)/SEMESTER 1/praktikum Algoritma Pemogram
an Dasar (main)
$ git commit -m "update"
[main 68dc82b] update
3 files changed, 160 insertions(+)
 create mode 100644 praktikum-apd/kelas/pertemuan 4/main.py
 create mode 100644 praktikum-apd/post-test/post-test-apd-4/2509106024-FERLYAHMA
DNABIL-APD-4.pdf
 create mode 100644 praktikum-apd/post-test/post-test-apd-4/2509106024-FERLYAHMA
DNABIL-APD-4.py

```

Git commit untuk menyimpan perubahan dalam *repository* dengan commit dan pesan "*update*".

## 5.4 GIT Remote

```
Asus@DESKTOP-8GBFI02 MINGW64 /d/(KULIAH)/SEMESTER 1/praktikum Algoritma Pemogram  
an Dasar (main)  
$ git remote add origin https://github.com/ferlynabil/Praktikum-APD.git
```

Git Remote digunakan untuk menambahkan *repository remote* dengan nama *origin*.

## 5.5 GIT Push

```
ASUS@DESKTOP-8GBFI02 MINGW64 /d/(KULIAH)/SEMESTER 1/praktikum Algoritma Pemogram  
an Dasar (main)  
$ git push origin main  
info: please complete authentication in your browser...  
Enumerating objects: 14, done.  
Counting objects: 100% (14/14), done.  
Delta compression using up to 16 threads  
Compressing objects: 100% (8/8), done.  
Writing objects: 100% (10/10), 557.94 KiB | 15.94 MiB/s, done.  
Total 10 (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/ferlynabil/Praktikum-APD.git
```