

**LAPORAN PRAKTIKUM**  
**POSTTEST 2**  
**ALGORITMA PEMROGRAMAN DASAR**



**Disusun oleh:**  
**Nama Syifa Siti Aulia Anwar**  
**Kelas A'25**

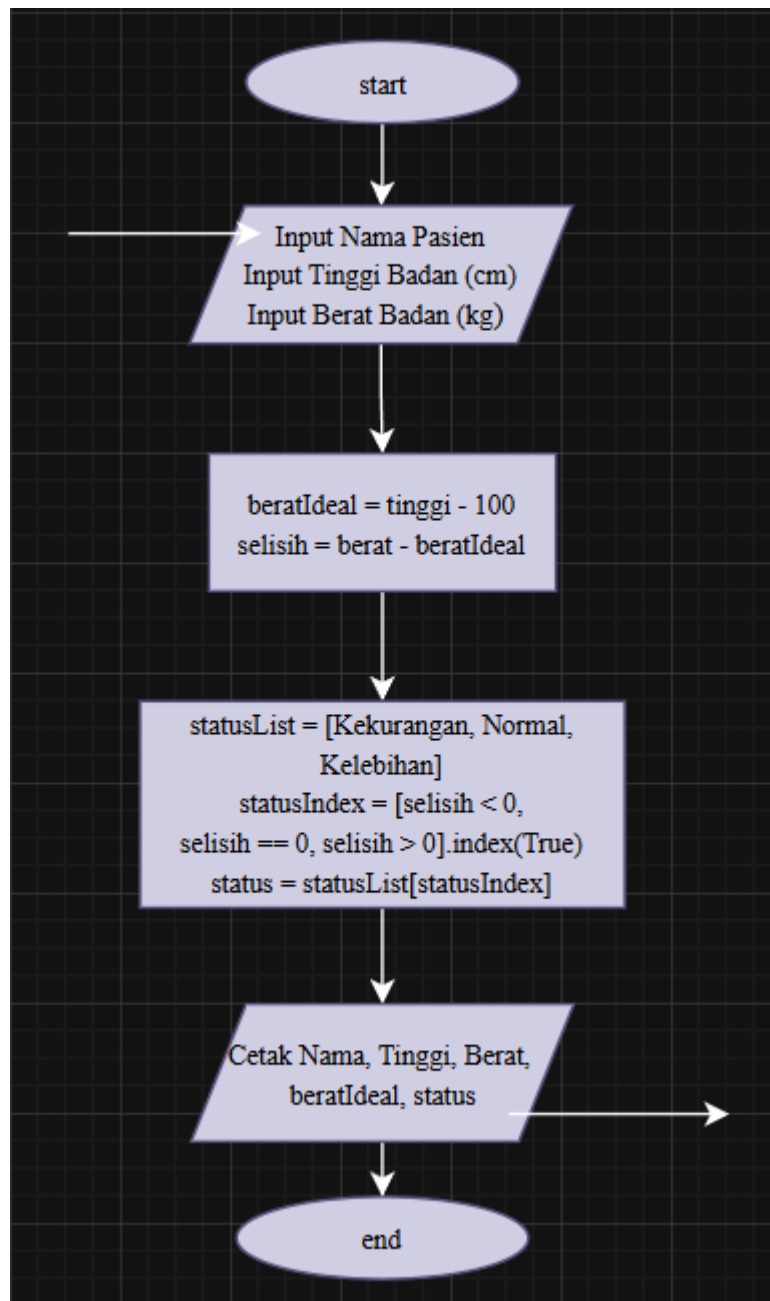
**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

## 1. FLOWCHART

Flowchart atau bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah.

Flowchart berperan penting dalam memutuskan sebuah langkah atau fungsionalitas dari sebuah proyek pembuatan program yang melibatkan banyak orang sekaligus. Selain itu dengan menggunakan bagan alur proses dari sebuah program akan lebih jelas, ringkas, dan mengurangi kemungkinan untuk salah penafsiran. Penggunaan flowchart dalam dunia pemrograman juga merupakan cara yang bagus untuk menghubungkan antara kebutuhan teknis dan non-teknis.

Berikut flowchart sederhana untuk menentukan berat badan ideal:



## 2. DESKRIPSI SINGKAT PROGRAM

Program cek berat badan ini sebenarnya simpel tapi cukup berguna. Jadi, pertama kita masukan dulu data pasien seperti nama, tinggi badan, dan berat badan. Setelah itu, program langsung hitung berat ideal pakai rumus tinggi badan dikurang 100. Dari situ, dia cari tahu selisih antara berat asli dan berat idealnya. Program ini tidak pakai percabangan if-else seperti biasanya. Sebagai gantinya, dia pakai logika boolean dan list index buat nentuin status berat badan apakah pasien kekurangan, normal, atau kelebihan berat badan. Hasil akhirnya ditampilkan dalam bentuk tabel.

Jadi, alur programnya dimulai dari "Mulai", itu tandanya proses sudah jalan. Langkah pertama, program minta kita masukan nama pasien, tinggi badan (dalam cm), dan berat badan (dalam kg). Ini bagian input data, seperti kita ngisi formulir dulu. Setelah itu, program langsung hitung berat ideal pakai rumus sederhana yaitu: tinggi badan dikurang 100. Misalnya tinggi 165 cm, berarti berat idealnya 65 kg. Dari situ dihitung juga selisih antara berat asli dan berat ideal.

Langkah berikutnya, program udah punya tiga kemungkinan status:

- a. Kekurangan berat badan
- b. Berat badan normal
- c. Kelebihan berat badan

Dia langsung bikin list status dan list kondisi boolean (True atau False) berdasarkan selisih tadi. Setelah itu dia cari posisi True di list itu, dan ambil status yang sesuai dari list status. Jadi logikanya tetap jalan, tapi tanpa percabangan. Terakhir, program cetak hasilnya dalam bentuk tabel yang isinya : nama pasien, tinggi, berat, berat ideal, dan status berat badan. Setelah itu, proses selesai.

### 3. SOURCE CODE

Ini adalah code yang saya buat untuk program menentukan berat badan ideal secara sederhana:

```
# Input data pasien
nama = input("Masukkan Nama Pasien: ")
tinggi = float(input("Masukkan Tinggi Badan (cm): "))
berat = float(input("Masukkan Berat Badan (kg): "))

# Hitung berat ideal
beratIdeal = tinggi - 100
selisih = berat - beratIdeal

# Flags boolean
isKekurangan = selisih < -5
isNormal = -5 <= selisih <= 5
isKelebihan = selisih > 5

# List status
statusList = ["Kekurangan Berat Badan", "Berat Badan Normal", "Kelebihan Berat Badan"]

# Boolean sebagai index
statusIndex = [isKekurangan, isNormal, isKelebihan].index(True)
status = statusList[statusIndex]

# Hasil cek dalam format tabel
print("=" * 65)
print(f'|{"HASIL CEK BERAT BADAN":^63}|')
print("=" * 65)
print(f'| Nama Pasien      :{nama:<45}|')
print(f'| Tinggi Badan     :{tinggi:>4.1f} cm{'':<37}|')
print(f'| Berat Badan      :{berat:>4.1f} kg{'':<38}|')
print(f'| Berat Ideal      :{beratIdeal:>4.1f} kg{'':<38}|')
print(f'| Status           :{status:<45}|')
print("=" * 65)
```

### 4. HASIL OUTPUT

```
Masukkan Nama Pasien: cipa
Masukkan Tinggi Badan (cm): 154
Masukkan Berat Badan (kg): 43

=====
|                               HASIL CEK BERAT BADAN                               |
=====
| Nama Pasien      :cipa                                                    |
| Tinggi Badan     :154.0 cm                                                |
| Berat Badan      :43.0 kg                                                 |
| Berat Ideal      :54.0 kg                                                 |
| Status           :Kekurangan Berat Badan                                |
=====
```

Jadi, pasien namanya cipa, tingginya 154 cm, dan berat badannya 43 kg. Program langsung hitung berat idealnya, yaitu  $154 - 100 = 54$  kg. Karena berat badannya cuma 43 kg, berarti masih kurang dari idealnya. Maka status yang keluar adalah "Kekurangan Berat Badan".

```

Masukkan Nama Pasien: siti
Masukkan Tinggi Badan (cm): 150
Masukkan Berat Badan (kg): 50
=====
|                                     |
|                               HASIL CEK BERAT BADAN                               |
|                                     |
| Nama Pasien      :siti                                     |
| Tinggi Badan    :150.0 cm                                 |
| Berat Badan     :50.0 kg                                  |
| Berat Ideal     :50.0 kg                                  |
| Status          :Berat Badan Normal                      |
|                                     |
=====

```

Git push adalah perintah untuk mengirim perubahan dari repositori lokal ke repositori jarak jauh (remote). Dengan git push, semua commit yang telah dibuat secara lokal akan dikirim dan disimpan di server, sehingga dapat diakses oleh tim atau perangkat lain yang terhubung dengan repositori tersebut. Perintah ini umum digunakan setelah melakukan commit, agar perubahan dapat disinkronkan ke repositori online seperti GitHub.

```

Masukkan Nama Pasien: aulia
Masukkan Tinggi Badan (cm): 152
Masukkan Berat Badan (kg): 70
=====
|                                     |
|                               HASIL CEK BERAT BADAN                               |
|                                     |
| Nama Pasien      :aulia                                    |
| Tinggi Badan    :152.0 cm                                 |
| Berat Badan     :70.0 kg                                  |
| Berat Ideal     :52.0 kg                                  |
| Status          :Kelebihan Berat Badan                    |
|                                     |
=====

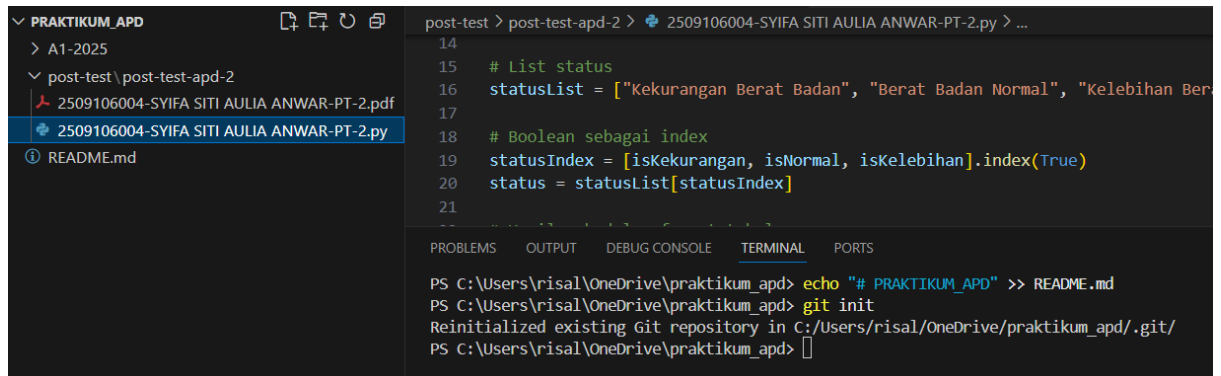
```

Jadi, pasien namanya aulia, tingginya 152 cm, dan berat badannya 70 kg. Program langsung hitung berat idealnya, yaitu  $152 - 100 = 52$  kg. Karena berat badannya 70 kg, berarti masih lebih dari idealnya. Maka status yang keluar adalah "Kelebihan Berat Badan".

## 5. LANGKAH LANGKAH GIT

Git itu alat untuk menyimpan dan mengatur versi kode. Jadi, ketika kita sedang ngoding, Git bantu kita untuk simpan perubahan, balik ke versi sebelumnya kalau ada yang salah, dan bisa kerja bersama tim tanpa membuat file berantakan. Misalnya kita nambah fitur baru, kita bisa kerja di jalur terpisah, terus gabungin ke kode utama kalau sudah selesai. Semua perubahan tercatat, jadi gampang dilacak dan aman buat kerja sama tim.

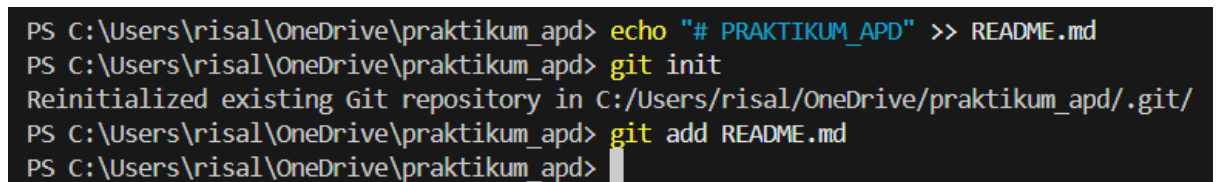
### 5.1 GIT Init



```
post-test > post-test-apd-2 > 2509106004-SYIFA SITI AULIA ANWAR-PT-2.py > ...
14
15 # List status
16 statusList = ["Kekurangan Berat Badan", "Berat Badan Normal", "Kelebihan Ber
17
18 # Boolean sebagai index
19 statusIndex = [isKekurangan, isNormal, isKelebihan].index(True)
20 status = statusList[statusIndex]
21
22
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\risal\OneDrive\praktikum_apd> echo "# PRAKTIKUM_APD" >> README.md
PS C:\Users\risal\OneDrive\praktikum_apd> git init
Reinitialized existing Git repository in C:/Users/risal/OneDrive/praktikum_apd/.git/
PS C:\Users\risal\OneDrive\praktikum_apd>
```

Git init ini fungsinya untuk mulai proyek Git dari nol. Jadi, saat kita jalankan perintah ini di folder, Git langsung bikin sistem pelacak versi di situ. Setelah itu, kita bisa mulai simpan perubahan kode, bikin commit, dan pakai fitur Git yang lainnya.

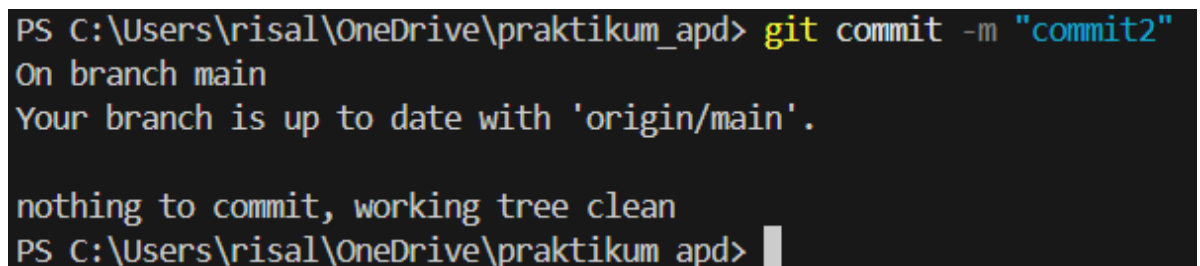
### 5.2 GIT Add



```
PS C:\Users\risal\OneDrive\praktikum_apd> echo "# PRAKTIKUM_APD" >> README.md
PS C:\Users\risal\OneDrive\praktikum_apd> git init
Reinitialized existing Git repository in C:/Users/risal/OneDrive/praktikum_apd/.git/
PS C:\Users\risal\OneDrive\praktikum_apd> git add README.md
PS C:\Users\risal\OneDrive\praktikum_apd>
```

Git add ini fungsinya untuk memberi tahu Git file mana yang mau disimpan perubahannya. Jadi misalnya kita udah ngedit file, sebelum bisa disimpan ke versi baru (commit), kita harus daftarin file itu menggunakan git add.

### 5.3 GIT Commit

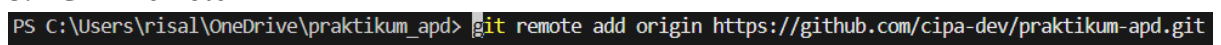


```
PS C:\Users\risal\OneDrive\praktikum_apd> git commit -m "commit2"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS C:\Users\risal\OneDrive\praktikum_apd>
```

Git commit itu untuk menyimpan perubahan dan kasih catatan kecil tentang apa yang kita ubah.

### 5.4 GIT Remote



```
PS C:\Users\risal\OneDrive\praktikum_apd> git remote add origin https://github.com/cipa-dev/praktikum-apd.git
```

Git remote adalah perintah untuk menghubungkan repositori lokal dengan repositori jarak jauh (seperti GitHub), agar bisa mengirim (push) atau mengambil (pull) perubahan kode dari server.

### 5.5 GIT Push

```
PS C:\Users\risal\OneDrive\praktikum_apd> git push -u origin main  
branch 'main' set up to track 'origin/main'.  
Everything up-to-date
```

Git push adalah perintah untuk mengirim perubahan dari repositori lokal ke repositori jarak jauh. Dengan git push, semua commit yang telah dibuat secara lokal akan dikirim dan disimpan di server, sehingga dapat diakses oleh perangkat lain yang terhubung dengan repositori tersebut.