

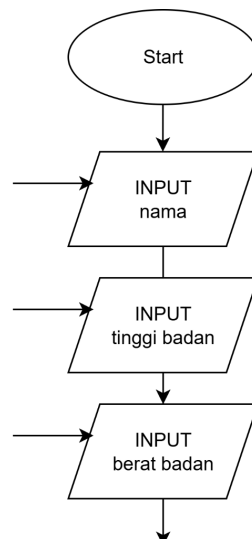
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:
Muhammad Alfauzi Syahputra 2509106006
Kelas A1 '25

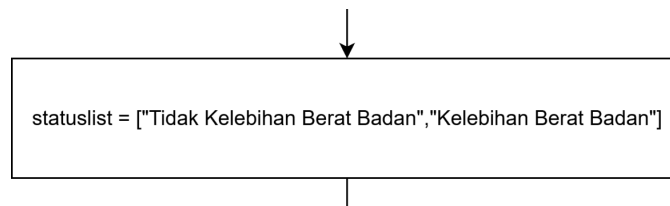
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



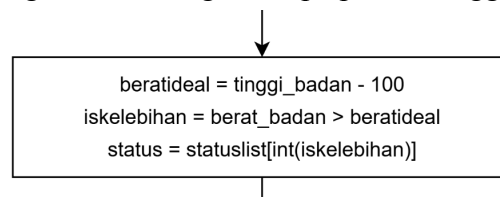
gambar 1.1 flowchart Start Dan Input

Pada flowchart gambar 1.1, flowchart dimulai dengan simbol Start yang menandai awal jalannya program. Setelah itu, pengguna diminta untuk memasukkan tiga data penting, yaitu nama pasien, tinggi badan dalam satuan cm, dan berat badan dalam satuan kg. Data ini akan menjadi dasar perhitungan berat ideal dan pengecekan status berat badan.



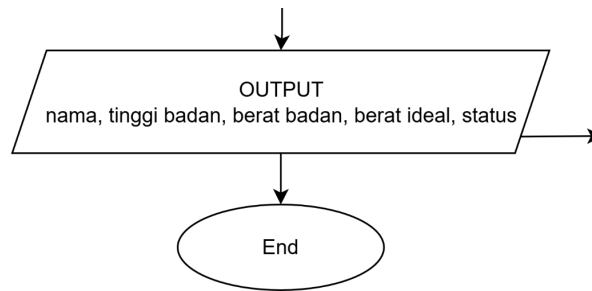
gambar 1.2 flowchart list

Lanjut ke flowchart gambar 1.2, program menyiapkan sebuah list bernama *statuslist* yang berisi dua kemungkinan hasil, yaitu “Tidak Kelebihan Berat Badan” dan “Kelebihan Berat Badan”. List ini berfungsi sebagai penentu status pasien sehingga program dapat menentukan hasil hanya dengan indeks angka, tanpa perlu menggunakan percabangan *if-else*.



gambar 1.3 flowchart Proses

Pada flowchart gambar 1.3, program mulai melakukan perhitungan. Pertama, *beratideal* dihitung dengan rumus $tinggi_badan - 100$. Selanjutnya, program melakukan perbandingan antara *berat_badan* pasien dengan *beratideal*. Hasil perbandingan ini berupa nilai logika (*True* jika berat badan lebih besar, *False* jika tidak), kemudian nilai tersebut diubah menjadi angka 1 atau 0. Angka inilah yang digunakan sebagai indeks untuk mengambil status pasien dari list yang sudah disiapkan sebelumnya.



gambar 1.4 flowchart Output Dan End

Terakhir, di flowchart gambar 1.4, semua hasil perhitungan ditampilkan dalam bentuk tabel yang rapi dan informatif. Tabel mencantumkan nama pasien, tinggi badan, berat badan, berat ideal, dan status akhir apakah berat badan pasien ideal atau kelebihan. Setelah seluruh informasi ditampilkan, flowchart diakhiri dengan simbol End yang menandakan program selesai dijalankan.

2. Deskripsi Singkat Program

Tujuan dari program ini adalah untuk membantu pengguna mengecek apakah berat badan seseorang termasuk kelebihan berat badan atau tidak kelebihan berat badan berdasarkan tinggi badan. Fungsi utamanya adalah untuk menghitung berat badan ideal, membandingkannya dengan berat badan aktual, dan kemudian menampilkan hasil tersebut secara informatif dalam bentuk tabel di terminal. Hal ini bertujuan untuk memberikan informasi yang cepat dan jelas kepada pengguna, sehingga memudahkan mereka dalam memantau kesehatan.

3. Source Code

A. Input Data

menampilkan judul dan garis bingkai, kemudian meminta pengguna memasukkan nama pasien, tinggi badan (cm), dan berat badan (kg) dengan label yang dirapikan agar rapi.

Source Code:

```

print("="*80)
print(f'{"PROGRAM CEK BERAT BADAN IDEAL"}.center(80))
print("="*80)
nama_pasien = str(input(f'{"Masukkan nama pasien":<{30}}: "))
tinggi_badan = float(input(f'{"Masukkan tinggi badan (cm)":<{30}}: "))
berat_badan = float(input(f'{"Masukkan berat badan (kg)":<{30}}: "))
print("="*80)
  
```

B. Proses Perhitungan

Menghitung berat badan ideal dengan rumus tinggi badan dikurangi 100. Untuk menentukan status pasien, program membandingkan berat badan aktual dengan berat ideal,

menghasilkan nilai Boolean (*True* jika berat badan lebih dari ideal, *False* jika sama atau kurang). Nilai Boolean ini kemudian dikonversi menjadi integer menggunakan fungsi *int()*, di mana *False* menjadi 0 dan *True* menjadi 1. Nilai 0 atau 1 ini digunakan sebagai indeks untuk mengambil status dari daftar *statuslist*, sehingga jika indeks 0, statusnya “Tidak Kelebihan Berat Badan”, dan jika indeks 1, statusnya “Kelebihan Berat Badan”. Dengan cara ini, status pasien dapat ditentukan secara otomatis tanpa menggunakan if-else

Source Code:

```
statuslist = ["Tidak Kelebihan Berat Badan", "Kelebihan Berat Badan"]
beratideal = tinggi_badan - 100
iskelebihan = berat_badan > beratideal
status = statuslist[int(iskelebihan)]
```

C. Output Hasil

menampilkan hasil dalam bentuk tabel kotak rapi dengan lebar 78 karakter, mencakup nama pasien, tinggi badan, berat badan, berat ideal, dan status.

Source Code:

```
lebar = 78
print(f"┌{'-'*lebar}┐")
print(f"│{'HASIL CEK BERAT BADAN':^{lebar}}│")
print(f"└{'-'*lebar}┘")
print("│{:<20} : {:<54}│".format("Nama Pasien", nama_pasien))
print("│{:<20} : {:<54}│".format("Tinggi Badan", f"{tinggi_badan:.0f} cm"))
print("│{:<20} : {:<54}│".format("Berat Badan", f"{berat_badan:.0f} kg"))
print("│{:<20} : {:<54}│".format("Berat Ideal", f"{beratideal:.0f} kg"))
print("│{:<20} : {:<54}│".format("Status", status))
print(f"└{'-'*lebar}┘")
```

4. Hasil Output

```
=====
                        PROGRAM CEK BERAT BADAN IDEAL
=====
Masukkan nama pasien      : Alfauzi
Masukkan tinggi badan (cm) : 165
Masukkan berat badan (kg)  : 60
=====

HASIL CEK BERAT BADAN

Nama Pasien      : Alfauzi
Tinggi Badan     : 165 cm
Berat Badan      : 60 kg
Berat Ideal      : 65 kg
Status           : Tidak Kelebihan Berat Badan
```

gambar 4.1 Terminal dengan output tidak kelebihan berat badan

```
=====
                        PROGRAM CEK BERAT BADAN IDEAL
=====
Masukkan nama pasien      : Syahputra
Masukkan tinggi badan (cm) : 155
Masukkan berat badan (kg)  : 80
=====

HASIL CEK BERAT BADAN

Nama Pasien      : Syahputra
Tinggi Badan     : 155 cm
Berat Badan      : 80 kg
Berat Ideal      : 55 kg
Status           : Kelebihan Berat Badan
```

gambar 4.2 Terminal dengan output kelebihan berat badan

5. Langkah-langkah GIT

5.1 GIT Init

git init digunakan untuk membuat repository Git baru di folder lokal, sehingga folder tersebut siap melacak semua perubahan yang terjadi pada file di dalamnya.

```
PS D:\Kuliah\Tugas\APDPraktikum\posttest2> git init
Initialized empty Git repository in D:/Kuliah/Tugas/APDPraktikum/posttest2/.git/
```

gambar 5.1 inisiasi

5.2 GIT Add

git add berfungsi untuk menambahkan file ke staging area, menandai file mana saja yang akan disimpan pada commit berikutnya.

```
PS D:\Kuliah\Tugas\APDPraktikum\posttest2> git add .
PS D:\Kuliah\Tugas\APDPraktikum\posttest2> git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   2509106006-Muhammad-Alfauzi-Syahputra-PT-2.py
```

gambar 5.2 menambahkan

5.3 GIT Commit

git commit digunakan untuk menyimpan perubahan yang sudah di-add ke repository lokal, biasanya disertai pesan singkat yang menjelaskan perubahan tersebut.

```
PS D:\Kuliah\Tugas\APDPraktikum\posttest2> git commit -m 'posttest2'
[main 1a4eaeb] posttest2
1 file changed, 26 insertions(+)
```

gambar 5.3 menyimpan

5.4 GIT Remote

git remote berfungsi untuk menghubungkan repository lokal dengan repository di GitHub atau server lain, sehingga commit lokal bisa dikirim ke remote repository.

```
PS D:\Kuliah\Tugas\APDPraktikum\posttest2> git remote add origin https://github.com/Alfauzi-S/praktikum-apd.git
```

gambar 5.4 menghubungkan

5.5 GIT Push

git push digunakan untuk mengirim commit dari repository lokal ke repository remote di GitHub, agar perubahan bisa terlihat online dan diakses oleh orang lain.

```
PS D:\Kuliah\Tugas\APDPraktikum\posttest2> git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 698 bytes | 698.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Alfauzi-S/praktikum-apd.git
   e0a5607..1a4eaeb  main -> main
branch 'main' set up to track 'origin/main'.
```

gambar 5.4 mendorong commit dari repository lokal ke repository remote