

LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:
Syarifah Anargya Rizky (2509106007)
Kelas A1'25

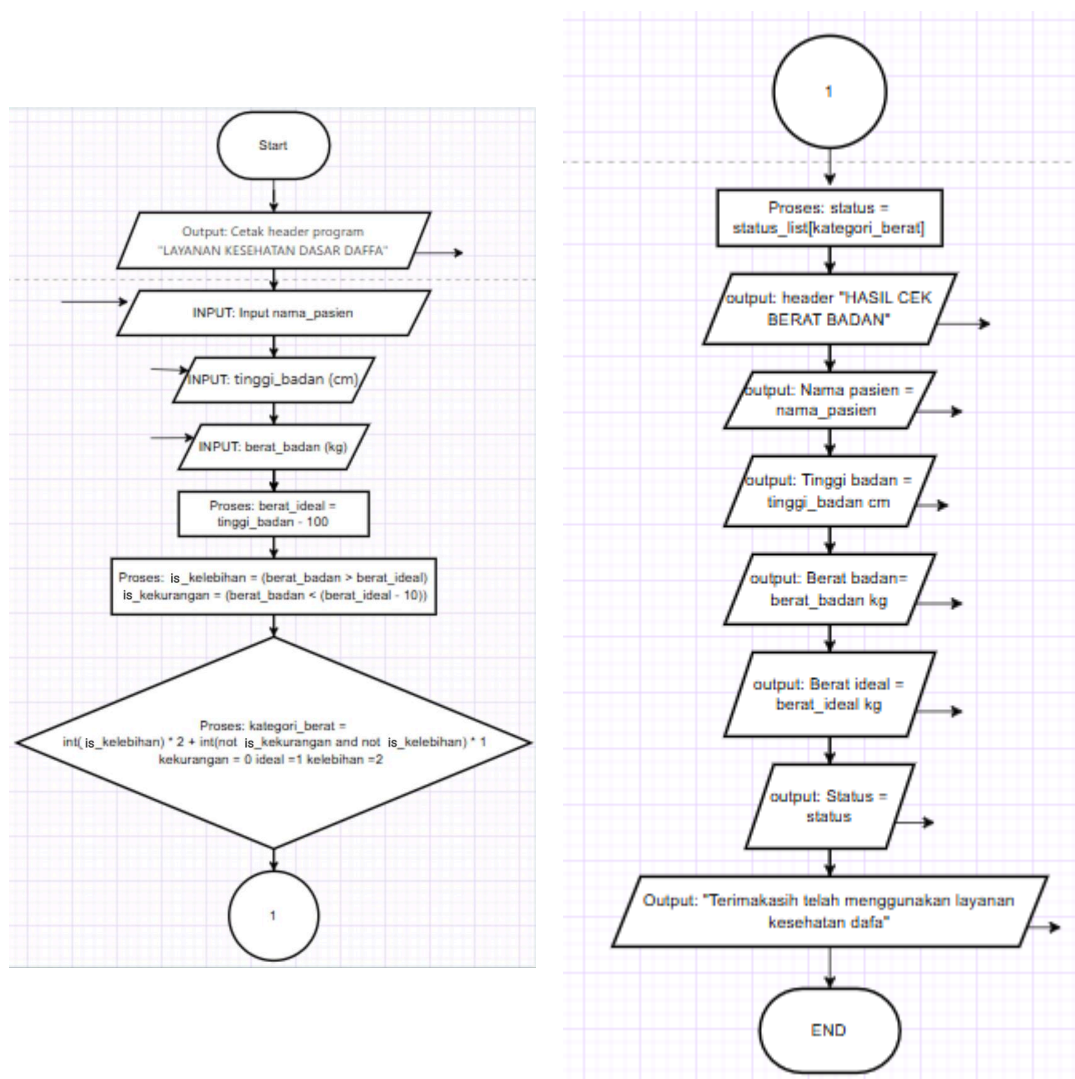
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

Daftar isi

Daftar isi.....	1
1. Flowchart.....	2
Gambar 1.1 Flowchart program menentukan berat badan ideal secara sederhana...	2
2. Deskripsi Singkat Program.....	3
3. Source Code.....	4
4. Hasil Output.....	5
Gambar 4.1 Output berat badan pasien kurang.....	5
Gambar 4.2 Output berat badan pasien ideal.....	5
Gambar 4.3 Output berat badan pasien berlebih.....	6
5. Langkah-langkah GIT.....	7
5.1 GIT Init.....	7
Gambar 5.1.1 Mengubah foldernya menjadi repository git.....	7
5.2 GIT Add.....	7
Gambar 5.2.1 Mengubah foldernya menjadi repository git.....	7
5.3 GIT Commit.....	8
Gambar 5.3.1 Menyimpan perubahan.....	8
5.4 GIT Remote.....	8
Gambar 5.4.1 Menghubungkan folder proyek kita ke server git.....	8
5.5 GIT Push.....	8
Gambar 5.5.1 Menghubungkan respiratory lokal ke github.....	8

1. Flowchart

Flowchart merupakan diagram yang biasa digunakan dalam dunia informatika dengan tujuan agar dapat menggambarkan dengan lebih jelas dari suatu langkah-langkah, keputusan, dan alur kerja dari suatu proses dalam sistem. Flowchart terdiri dari berbagai simbol-simbol dengan makna tersendiri yang digunakan untuk menyederhanakan rangkaian prosedur agar memudahkan pemahaman terhadap informasi yang ingin disampaikan.



Gambar 1.1 Flowchart program menentukan berat badan ideal secara sederhana

Alur flowchart ini dimulai dengan memasukkan INPUT sekaligus OUTPUT (jadi data yang sudah disimpan dalam input itu nanti bakal di print lagi buat outputnya) judul program, lalu masukan INPUT sekaligus OUTPUT data data pasien yang dibutuhkan seperti nama pasien, tinggi badan (cm), berat badan (kg) disini saya pakai FLOAT biar angka tinggi badan dan juga berat badan pasien bisa berupa angka desimal. Lakukan PROSES untuk menentukan berat badan yang ideal dengan rumus $\text{tinggi pasien} - 100$, selanjutnya kita PROSES untuk tolak ukur penentuan status kekurangan ataupun kelebihan berat badannya dengan cara proses variabel $\text{is_kelebihan} = (\text{berat_badan} > \text{berat_ideal})$ untuk kelebihan berat badan dan $\text{is_kekurangan} = (\text{berat_badan} < (\text{berat_ideal} - 10))$. Lakukan PROSES perhitungan untuk menentukan kategori berat berdasarkan dua kondisi yakni variabel is_kelebihan dan is_kekurangan yang merupakan variabel boolean, kategori berat badan ini ditentukan berdasarkan kondisi kekurangan bernilai 0, ideal bernilai 1, dan kelebihan bernilai 2, jadi jika $\text{is_kelebihan} = \text{true}$ maka $\text{INTEGER}(\text{is_kelebihan})$ akan menjadi 1 dan hasilnya akan dikalikan dengan 2, sama pula jika ternyata $\text{is_kelebihan} = \text{false}$ maka hasilnya akan menjadi 0 dikali 2, selanjutnya kita cek apakah berat badannya pasien tidak kekurangan dan tidak kelebihan, jika iya maka berarti berat badannya pasien itu ideal, jika kondisi ideal terpenuhi maka $\text{INTEGER}(\text{true})$ menjadi 1 lalu hasilnya dikalikan 1 pula, dan jika ternyata berat badannya tidak ideal maka hasilnya akan menjadi 0. Lalu kita masuk ke PROSES terakhir yaitu memproses status yang sudah kita ambil berdasarkan indeks tadi (kekurangan bernilai 0, ideal bernilai 1, dan kelebihan bernilai 2), dan masuk ke INPUT sekaligus OUTPUT terakhir kita yang akan mencetak nama, tinggi badan, berat badan pasien, serta berat ideal seharusnya juga status berat badan pasiennya apakah ideal, kekurangan ataupun kelebihan, setelah itu selesai.

2. Deskripsi Singkat Program

Program sederhana untuk mengukur berat badan ideal dari pelayanan kesehatan dasar daffa ini dibuat dengan tujuan agar dapat membantu pasien untuk mengetahui indeks atau status berat badan mereka, apakah berat badan pasien sudah ideal, kekurangan (ini fitur yang saya tambahkan sendiri) maupun kelebihan. Program ini akan sangat membantu bagi para pasien atau user yang menggunakan layanan kesehatan dasar dafa ini agar dapat mengetahui berat badan yang ideal untuk para pasien dengan tolak ukur tinggi badan, sehingga para user atau pasien dapat menjaga ataupun memperbaiki pola hidupnya agar mendapatkan berat badan yang ideal serta menghindari banyak penyakit.

3. Source Code

Berikut ini merupakan source code yang saya buat untuk program menentukan berat badan ideal secara sederhana.

```
# Program untuk cek berat badan di layanan kesehatan Daffa yang berguna untuk menentukan berat badan ideal
print("=" * 85)
print("|          LAYANAN KESEHATAN DASAR DAFFA          |")
print("|          PROGRAM CEK BERAT BADAN          |")
print("=" * 85)

# Meminta data pasien dari pengguna
nama_pasien = input("Masukkan Nama Pasien: ")
tinggi_badan = float(input("Masukkan Tinggi Badan (cm): "))
berat_badan = float(input("Masukkan Berat Badan (kg): "))

#penentuan untuk tolak ukurnya
berat_ideal = tinggi_badan - 100
is_kelebihan = berat_badan > berat_ideal
is_kekurangan = berat_badan < (berat_ideal - 10)

# Kekurangan = 0, Kelebihan = 2, Ideal = 1
kategori_berat = int(is_kelebihan) * 2 + int(not is_kekurangan and not is_kelebihan) * 1

# Daftar status: kekurangan, ideal, atau kelebihan
status_list = ["Kekurangan Berat Badan", "Berat Badan Ideal", "Kelebihan Berat Badan"]

#keterangan statusnya
status = status_list[kategori_berat]

print("=" * 85)
print("|          HASIL CEK BERAT BADAN          |")
print("=" * 85)
print(f"| Nama Pasien      : {nama_pasien:<52}      |")
print(f"| Tinggi Badan    : {tinggi_badan:<5.1f} cm{' ' * 52} |")
print(f"| Berat Badan     : {berat_badan:<5.1f} kg{' ' * 52} |")
print(f"| Berat Ideal     : {berat_ideal:<5.1f} kg{' ' * 52} |")
print(f"| Status          : {status:<52}          |")
print("=" * 85)

print("\nTerima kasih telah menggunakan layanan kesehatan Daffa!")
```

4. Hasil Output

```
=====
|                                     |
|          LAYANAN KESEHATAN DASAR DAFFA          |
|          PROGRAM CEK BERAT BADAN          |
|=====|
Masukkan Nama Pasien: Seokjin
Masukkan Tinggi Badan (cm): 179
Masukkan Berat Badan (kg): 63.4
=====
|                                     |
|          HASIL CEK BERAT BADAN          |
|=====|
| Nama Pasien      : Seokjin              |
| Tinggi Badan    : 179.0 cm              |
| Berat Badan     : 63.4 kg               |
| Berat Ideal     : 79.0 kg               |
| Status          : Kekurangan Berat Badan |
|=====|
Terima kasih telah menggunakan layanan kesehatan Daffa!
```

Gambar 4.1 Output berat badan pasien kurang

Output ini tercipta dari inputan nama pasien bernama Seokis dengan tinggi badan 179 cm dan berat 63 kg sehingga menghasilkan status kekurangan berat badan setelah diproses oleh program sederhana ini.

```
=====
|                                     |
|          LAYANAN KESEHATAN DASAR DAFFA          |
|          PROGRAM CEK BERAT BADAN          |
|=====|
Masukkan Nama Pasien: Jungkook
Masukkan Tinggi Badan (cm): 178
Masukkan Berat Badan (kg): 70
=====
|                                     |
|          HASIL CEK BERAT BADAN          |
|=====|
| Nama Pasien      : Jungkook              |
| Tinggi Badan    : 178.0 cm              |
| Berat Badan     : 70.0 kg               |
| Berat Ideal     : 78.0 kg               |
| Status          : Berat Badan Ideal      |
|=====|
Terima kasih telah menggunakan layanan kesehatan Daffa!
```

Gambar 4.2 Output berat badan pasien ideal

Output ini tercipta dari inputan nama pasien bernama Jungkook dengan tinggi badan 178 cm dan berat 70 kg sehingga menghasilkan status berat badan ideal setelah diproses oleh program sederhana ini.

```
=====
|                                     |
|               LAYANAN KESEHATAN DASAR DAFFA               |
|               PROGRAM CEK BERAT BADAN                     |
|=====
Masukkan Nama Pasien: Taeil jelek
Masukkan Tinggi Badan (cm): 155
Masukkan Berat Badan (kg): 65.8
=====
|                                     |
|               HASIL CEK BERAT BADAN                       |
|=====
| Nama Pasien      : Taeil jelek                             |
| Tinggi Badan    : 155.0 cm                                 |
| Berat Badan     : 65.8 kg                                  |
| Berat Ideal     : 55.0 kg                                  |
| Status          : Kelebihan Berat Badan                    |
|=====
Terima kasih telah menggunakan layanan kesehatan Daffa!
```

Gambar 4.3 Output berat badan pasien berlebih

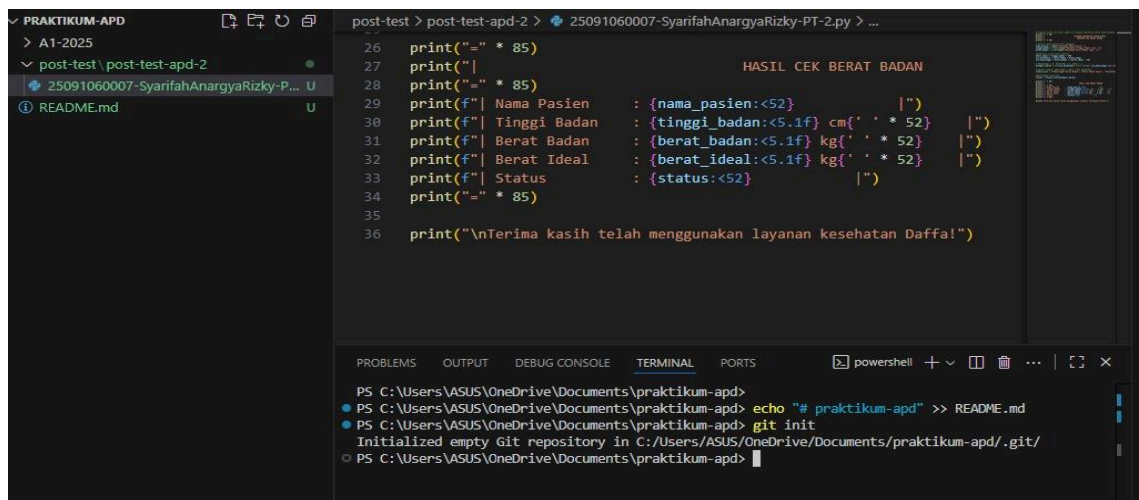
Output ini tercipta dari inputan nama pasien bernama Taeil jelek dengan tinggi badan 155cm dan berat 65,8 kg sehingga menghasilkan status berat badan ideal setelah diproses oleh program sederhana ini.

5. Langkah-langkah GIT

Git adalah sistem kontrol versi yang membantu kita melacak perubahan pada kode atau proyek kita, jadi dengan adanya git ini kita itu bisa ngeliat semua perubahan yang kita buat dari awal gitu, bahkan kita bisa kembali ke versi sebelumnya tanpa kehilangan jejak dari proyek yang sudah kita buat sebelumnya.

5.1 GIT Init

Git init ini berfungsi untuk membuat folder proyek kita ini menjadi repository git, jadi git ini bakal ngelacak perubahan file di folder ini.



The screenshot shows a VS Code editor with a file explorer on the left displaying a project structure: 'PRAKTIKUM-APD' > 'A1-2025' > 'post-test \ post-test-apd-2' > '25091060007-SyarifahAnargyaRizky-PT-2.py'. The main editor shows a Python script with a function 'HASIL_CEK_BERAT_BADAN' that takes patient data and returns a status. The terminal at the bottom shows the command 'git init' being executed, resulting in 'Initialized empty Git repository in C:/Users/ASUS/OneDrive/Documents/praktikum-apd/.git/'.

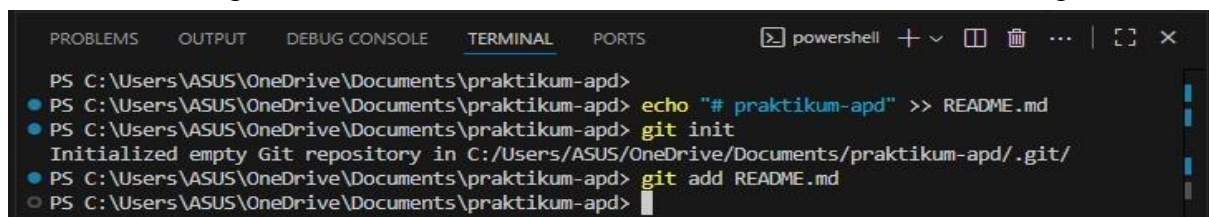
```
26 print("=" * 85)
27 print("|
28 print("=" * 85)
29 print(f"| Nama Pasien      : {nama_pasien:<52}      |")
30 print(f"| Tinggi Badan       : {tinggi_badan:<5.1f} cm(' ' * 52) |")
31 print(f"| Berat Badan        : {berat_badan:<5.1f} kg(' ' * 52) |")
32 print(f"| Berat Ideal        : {berat_ideal:<5.1f} kg(' ' * 52) |")
33 print(f"| Status              : {status:<52}              |")
34 print("=" * 85)
35
36 print("\nTerima kasih telah menggunakan layanan kesehatan Daffa!")
```

```
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd>
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd> echo "# praktikum-apd" >> README.md
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd> git init
Initialized empty Git repository in C:/Users/ASUS/OneDrive/Documents/praktikum-apd/.git/
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd>
```

Gambar 5.1.1 Mengubah foldernya menjadi repository git

5.2 GIT Add

Git add digunakan untuk menambahkan file dan memasukan file ke daftar perubahan.



The screenshot shows a VS Code terminal window with the following commands and output:

```
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd>
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd> echo "# praktikum-apd" >> README.md
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd> git init
Initialized empty Git repository in C:/Users/ASUS/OneDrive/Documents/praktikum-apd/.git/
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd> git add README.md
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd>
```

Gambar 5.2.1 Memindahkan file dari tempat asalnya ke tempat dimana file siap di commit

5.3 GIT Commit

Git commit digunakan untuk menyimpan perubahan secara permanen.

```
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd> git commit -m "first commit"
[master (root-commit) 0c278a3] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd> █
```

Gambar 5.3.1 Menyimpan perubahan

5.4 GIT Remote

Git remote digunakan untuk menghubungkan ke server git.

```
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd> git branch -M main
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd> git remote add origin https://github.com/seokjinjuseyyo/praktikum-apd.git
```

Gambar 5.4.1 Menghubungkan folder proyek kita ke server git

5.5 GIT Push

Git push digunakan untuk mengirim commit kita ke github nanti, jadi disambungkan gitu repository lokal kita ke github.

```
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd> git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 246 bytes | 61.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/seokjinjuseyyo/praktikum-apd.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Gambar 5.5.1 Menghubungkan repository lokal ke github