

<p>Nama: <b>(Muhammad Rizki)</b></p> <p>NIM: <b>(064102400020)</b></p>	 <p><b>Praktikum Algoritma &amp; Pemrograman</b></p>	<h1>MODUL 6</h1> <p>Nama Dosen: <b>Binti solihah, S.T, M.KOM</b></p> <p>Nama Asisten Laboratorium:</p> <ul style="list-style-type: none"><li>1. <b>Yustianas Rombon - 064002300015</b></li><li>2. <b>Vira Aditya Kurniawan - 065002300012</b></li></ul>
<p>Hari/Tanggal: <b>Hari, Tanggal Bulan 2022</b></p>		

## Fungsi (Function) pada Pemrograman Python

### 1. Teori Singkat

#### Fungsi

Fungsi merupakan suatu bagian dari program yang dimaksudkan untuk mengerjakan suatu tugas tertentu dan letaknya terpisah dari program yang memanggilnya. Pada pembuatan program yang kompleks dan memiliki banyak fitur, kita diharuskan menggunakan fungsi. Fungsi diperlukan agar mempermudah kita dalam membaca sebuah kode program dan mempermudah untuk merawatnya. Dengan adanya sebuah fungsi kita tidak perlu menulis kode sepanjang gerbang kereta api di program utama dan kita dapat memecah atau membaginya lalu tinggal memanggil fungsinya saja didalam program utama. Dan itu juga akan mempersingkat penulisan dari program utama tersebut nantinya

Fungsi memiliki parameter yang dapat dipergunakan untuk memasukkan atau menampung variabel kedalam sebuah fungsi. Dalam pendeklarasian parameter pada sebuah fungsi kita juga dapat memasukkan *default argument/parameter*.

Fungsi juga dapat mengembalikan nilai dengan cara menggunakan keyword *return* yang di taruh didalam blok program fungsi itu sendiri untuk mengembalikan nilai yang ingin kita kembalikan kedalam program utama.

Membuat Fungsi pada Python

Fungsi pada Python, dibuat dengan kata kunci *def* kemudian diikuti dengan nama fungsinya.



```
# Pendeklarasian sebuah fungsi
def nama_fungsi():
    print("Hello Ini Fungsi")

# Pemanggilan sebuah fungsi
nama_fungsi()
```

### Fungsi dengan parameter pada Python

Memasukkan parameter kedalam sebuah fungsi

```
# pendeklarasian fungsi
def luas_persegi(sisi):
    luas = sisi * sisi
    return luas

# pemanggilan fungsi
print("Luas persegi: ", luas_persegi(6))
```

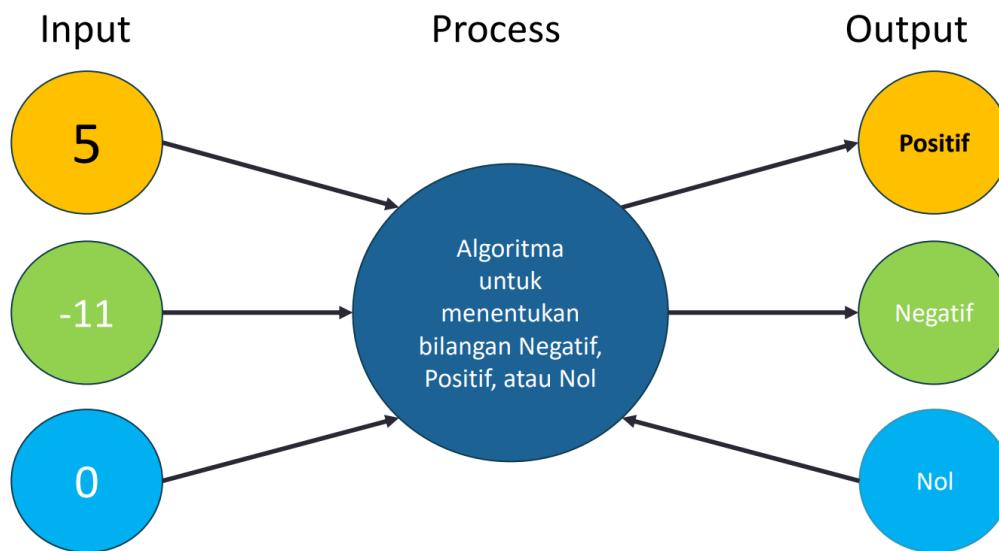


## IPO (Input Process Output)

Konsep Dasar Input, Process, dan Output (IPO)

- Konsep input, process, dan output adalah prinsip dasar dalam pemrograman dan pengembangan algoritma.
- Setiap algoritma melibatkan tiga tahap utama: mengambil data masukan (input), melakukan operasi atau pengolahan data (process), dan menghasilkan hasil akhir (output).
- Konsep ini menggambarkan bagaimana algoritma beroperasi untuk memproses informasi.

## Gambaran IPO (Menentukan Bilangan)



## Notasi Algoritma Flowchart

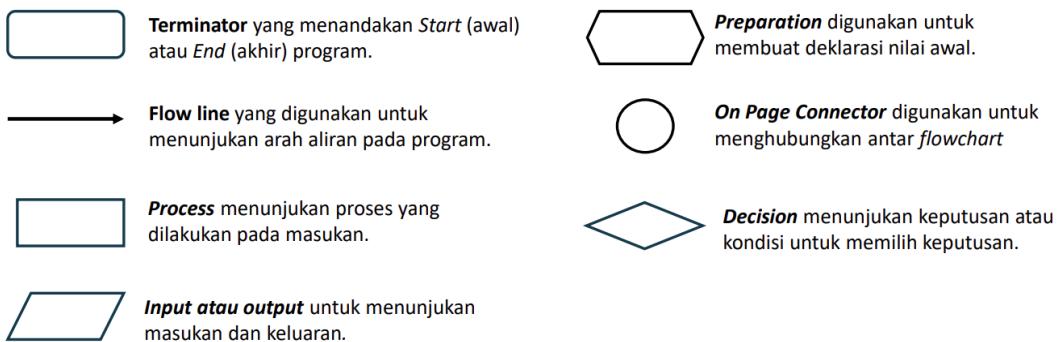
1. Flowchart adalah representasi visual atau diagram alir yang digunakan untuk menggambarkan langkah-langkah dan urutan proses suatu algoritma atau program.
2. Flowchart menyajikan langkah-langkah dalam bentuk simbol-simbol grafis yang saling terhubung, membantu dalam memvisualisasikan bagaimana informasi mengalir dan bagaimana proses dilakukan.
3. Dalam kaitannya dengan notasi deskriptif, notasi algoritma yang menggunakan flowchart dapat lebih cepat dibaca dan dilihat alur dan hubungannya.

## Simbol-simbol pada Flowchart



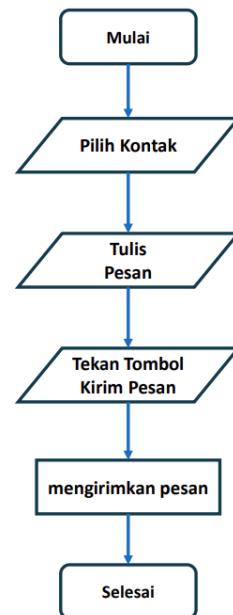
1. Setiap elemen flowchart dihubungkan oleh garis aliran bertanda panah
2. Garis aliran dimulai dari atas symbol dan keluar dari bagian bawah, kecuali symbol keputusan yang alirannya keluar dari bawah atau samping
3. Aliran bergerak dari atas ke bawah
4. Proses awal dan akhir menggunakan symbol terminal.

### ... Simbol-simbol pada Flowchart



**Contoh sederhana  
Penggunaan *flowchart*  
untuk menunjukkan algoritma**

**Kasus/Aliran:  
Mengirim pesan WhatsApp**



## 2. Alat dan Bahan

Hardware : Laptop/PC

Software : Spyder (Anaconda Python)

## 3. Elemen Kompetensi



a. Latihan pertama

Buatlah program yang sebelumnya telah dibuat yaitu program untuk merata-ratakan nilai sesuai dengan kategori huruf yang diinputkan dengan mengimplementasikan fungsi yang sudah dipelajari. Persyaratan program yaitu fungsi menggunakan *default argument/parameter* dan mengimplementasikan pengembalian nilai.

IPO (Input Process Output)

**Input**

1. **jumlah\_mapel:** Jumlah mata pelajaran yang akan dimasukkan oleh pengguna.
2. **nilai\_huruf:** Nilai huruf untuk setiap mata pelajaran, dimasukkan satu per satu (misalnya, "A", "B+", "C-").

**Proses**

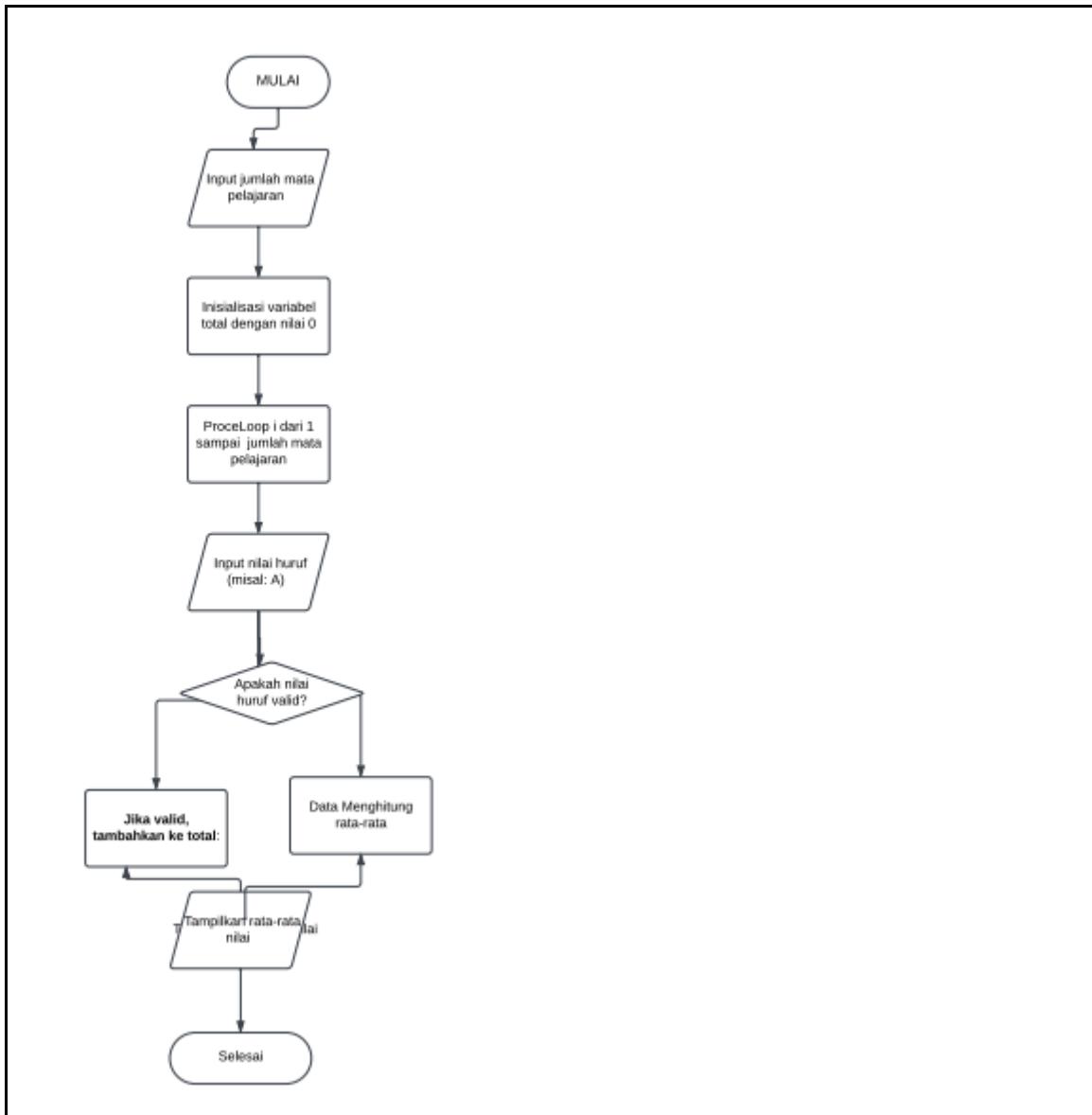
1. **Validasi Nilai Huruf:**
  - o Memeriksa setiap nilai huruf yang dimasukkan apakah ada di dalam dictionary `nilai_huruf`.
  - o Jika nilai tidak valid, program memberi pesan kesalahan dan menghentikan proses input.
2. **Konversi Nilai Huruf ke Nilai Numerik:**
  - o Menggunakan dictionary `nilai_huruf` untuk mengonversi setiap nilai huruf yang valid menjadi nilai numerik.
3. **Menghitung Total Nilai:**
  - o Menjumlahkan semua nilai numerik yang diperoleh dari konversi nilai huruf.
4. **Menghitung Rata-Rata Nilai:**
  - o Membagi total nilai dengan jumlah mata pelajaran (`jumlah_mapel`) untuk mendapatkan rata-rata nilai.

**Output**

- **Rata-rata Nilai:** Menampilkan rata-rata nilai dalam format dua desimal, misalnya, 3.25.

Flowchart





### Source Code



```
nilai_huruf = {
    'A': 4.00,
    'A-': 3.75,
    'B+': 3.50,
    'B': 3.00,
    'B-': 2.75,
    'C+': 2.50,
    'C': 2.00,
    'C-': 1.75,
    'D': 1.50,
    'E': 1.25
}

Fungsi untuk menghitung rata-rata nilai
def hitung_rata_rata_nilai(jumlah_mapel):
    nilai = []      Menyimpan nilai huruf yang dimasukkan pengguna

    for i in range(jumlah_mapel):
        huruf = input(f"Masukkan nilai mata pelajaran ke-{i + 1}:")

        Validasi agar input sesuai dengan nilai yang ada di
        dictionary
        if huruf in nilai_huruf:
            nilai.append(huruf)
        else:
            print("Nilai tidak valid. Harap masukkan nilai yang
sesuai (misalnya: A, B+, C-).")
            return

    Menghitung total nilai berdasarkan konversi dari nilai_huruf
    total = sum(nilai_huruf[huruf] for huruf in nilai) sum itu
    buat nambahin nilai nya

    Menghitung rata-rata nilai
    rata_rata = total / jumlah_mapel
    return rata_rata

Input jumlah mata pelajaran dari pengguna
jumlah_mapel = int(input("Masukkan jumlah mata pelajaran: "))

Memanggil fungsi untuk menghitung rata-rata
rata_rata = hitung_rata_rata_nilai(jumlah_mapel)
```



```
print(f"Rata-rata nilai: {rata_rata:.2f}")
```

Output

```
→ Masukkan jumlah mata pelajaran: 2
Masukkan nilai mata pelajaran ke-1: A
Masukkan nilai mata pelajaran ke-2: B-
Rata-rata nilai: 3.38
```

b. Latihan Kedua

Buatlah program yang sebelumnya telah dibuat yang menentukan jumlah hari dalam suatu bulan sesuai dengan inputan bulan dan tahun yang diinputkan oleh user dengan mengimplementasikannya menggunakan fungsi termasuk memperhatikan tahun kabisat dan non kabisat. Gunakan 2 fungsi beserta implementasikan parameternya.

IPO (input process output)



• **Input:**

- Masukkan nomor bulan (1-12): 2
- Masukkan tahun: 2024

• **Proses:**

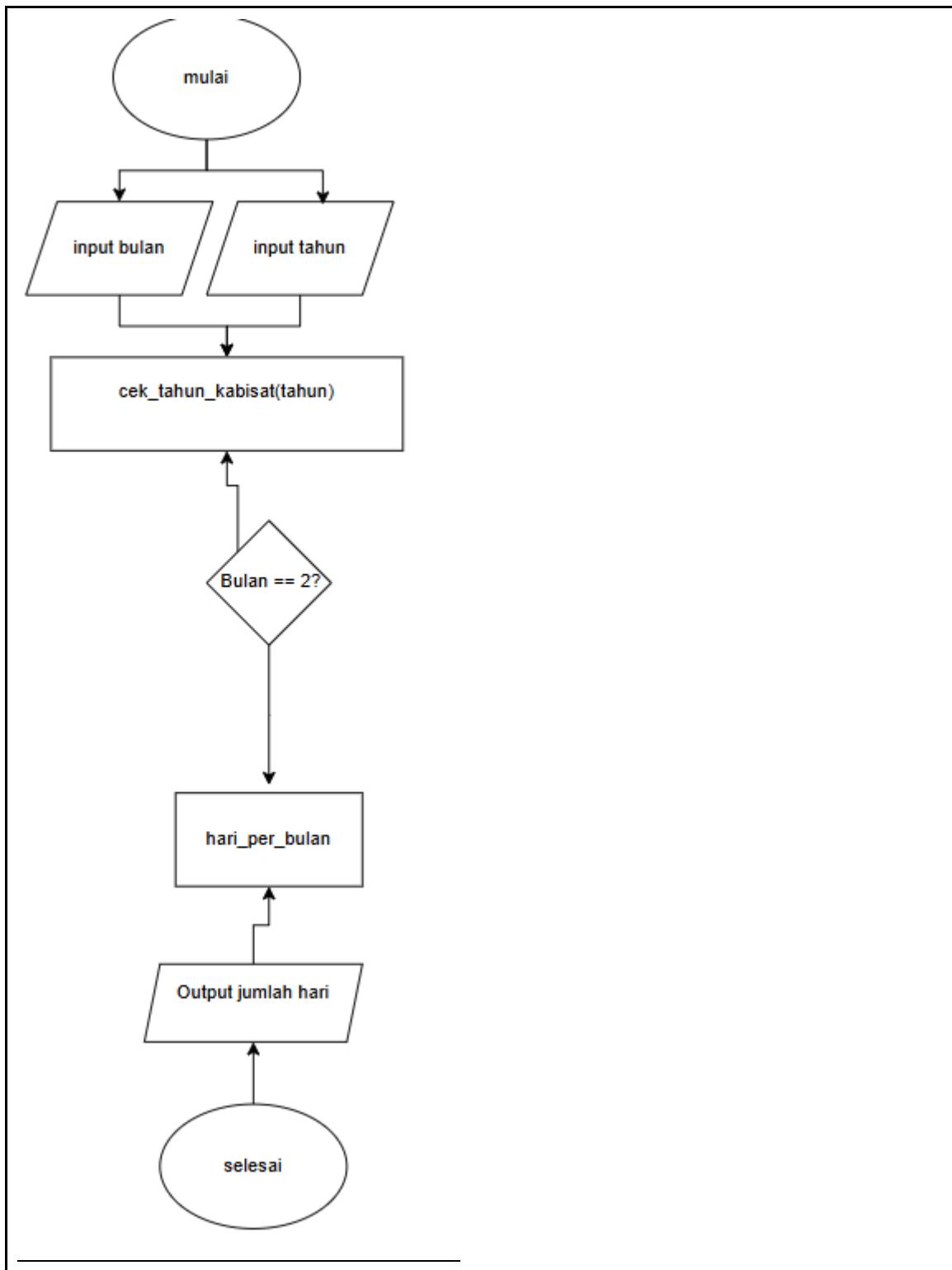
- Memeriksa apakah 2024 adalah tahun kabisat (hasil: `True`).
- Menghitung jumlah hari di bulan 2 (Februari) menjadi 29.

• **Output:**

- Jumlah hari dalam bulan ke-2 pada tahun 2024 adalah: 29

Flowchart





## Source Code

```
Fungsi untuk memeriksa apakah tahun adalah tahun kabisat
def cek_tahun_kabisat(tahun):
    Tahun kabisat adalah tahun yang habis dibagi 4, kecuali habis
    dibagi 100 namun tidak habis dibagi 400
    if (tahun % 4 == 0 and tahun % 100 != 0) or (tahun % 400 == 0):
        return True
    else:
        return False

Fungsi untuk menentukan jumlah hari dalam suatu bulan
def jumlah_hari_bulan(bulan, tahun):
    Daftar jumlah hari dalam tiap bulan pada tahun non-kabisat
    hari_per_bulan = {
        1: 31,      Januari
        2: 28,      Februari (akan disesuaikan jika kabisat)
        3: 31,      Maret
        4: 30,      April
        5: 31,      Mei
        6: 30,      Juni
        7: 31,      Juli
        8: 31,      Agustus
        9: 30,      September
        10: 31,     Oktober
        11: 30,     November
        12: 31,     Desember
    }

    Jika bulan adalah Februari dan tahun kabisat, ubah hari
    menjadi 29
    if bulan == 2 and cek_tahun_kabisat(tahun):
        return 29
    else:
        return hari_per_bulan.get(bulan, "Bulan tidak
valid") Mengembalikan hari atau pesan error

Input dari pengguna
bulan = int(input("Masukkan nomor bulan (1-12): "))
tahun = int(input("Masukkan tahun: "))

Memanggil fungsi untuk mendapatkan jumlah hari
```



```
hari = jumlah_hari_bulan(bulan, tahun)
print(f"Jumlah hari dalam bulan ke-{bulan} pada tahun {tahun}
adalah: {hari}")
```

### Output

```
→ Masukkan nomor bulan (1-12): 2
Masukkan tahun: 2022
Jumlah hari dalam bulan ke-2 pada tahun 2022 adalah: 28
```

#### 4. File Praktikum

Github Repository:

<https://github.com/muhrizki26/modul-6-algoritma.git>



## 5. Soal Latihan

Soal:

1. Jelaskan fungsi utama default *argument/parameter* dalam fungsi dan mengapa perlu menggunakan *default parameter/argument* pada sebuah fungsi?
2. Deskripsikan serta narasikan jalannya alur source code program yang sebelumnya telah kalian buat pada Elemen Kompetensi Latihan Kedua!

Jawaban:

1. Fungsi Utama Default Argument/Parameter

Default argument dalam fungsi memberikan nilai awal yang digunakan jika tidak ada argumen yang diberikan saat fungsi dipanggil. Ini meningkatkan fleksibilitas penggunaan fungsi dan membuat kode lebih bersih.

2. Deskripsi Alur Source Code Program

Program ini menghitung jumlah hari dalam bulan tertentu berdasarkan tahun. Pengguna diminta untuk memasukkan bulan dan tahun, lalu fungsi `cek\_tahun\_kabisat` memeriksa apakah tahun tersebut kabisat. Untuk Februari, jumlah hari ditentukan sesuai dengan status tahun kabisat, sementara untuk bulan lainnya, jumlah hari diambil dari dictionary. Hasil jumlah hari kemudian ditampilkan kepada pengguna.

## 6. Kesimpulan

- a. Dalam penggeraan program dengan bahasa pemrograman Python, kita harus benar-benar teliti dalam menginputkan suatu fungsi untuk menampilkan suatu keluaran pada layar dengan sesuai.
- b. Kita dapat mengetahui... Dalam pemrograman Python, ketelitian dalam menginputkan data sangat penting untuk memastikan hasil keluaran sesuai harapan, karena kesalahan kecil dalam argumen atau logika dapat mengakibatkan hasil yang salah atau program tidak berjalan. Memahami sintaksis dan logika, serta melakukan pengujian menyeluruh terhadap fungsi dan hasil keluaran, adalah langkah krusial untuk validasi. Selain itu, dokumentasi yang baik dan komentar dalam kode membantu pemahaman program, baik bagi pengembang maupun pengguna. Oleh karena itu, penerapan praktik terbaik seperti refactoring dan penanganan error juga sangat dianjurkan untuk meningkatkan kualitas dan efisiensi kode yang ditulis.



### 7. Cek List (✓)

No	Elemen Kompetensi	Penyelesaian	
		Selesai	Tidak Selesai
1.	Latihan Pertama	✓	
2.	Latihan Kedua	✓	

### 8. Formulir Umpang Balik

No	Elemen Kompetensi	Waktu Penggerjaan	Kriteria
1.	Latihan Pertama	80 Menit	menarik
2.	Latihan Kedua	60 Menit	cukup

Keterangan:

1. Menarik
2. Baik
3. Cukup
4. Kurang

