

<p>Nama: (MUHAMMAD RIZKI)</p> <p>NIM: (064102400020)</p>	 <p>Praktikum Algoritma & Pemrograman</p>	<h2>MODUL 4</h2> <p>Nama Dosen: Binti solihah, S.T, M.KOM</p> <p>Nama Asisten Laboratorium:</p> <ol style="list-style-type: none">Yustianas Rombon - 064002300015Vira Aditya Kurniawan - 065002300012
--	---	--

Struktur Kendali II (Control Structure II)

1. Teori Singkat

Operasi Perulangan

Perulangan dalam bahasa pemrograman berfungsi menugaskan komputer untuk melakukan sesuatu secara berulang-ulang. Terdapat dua jenis perulangan dalam bahasa pemrograman python, yaitu perulangan dengan *while* dan *for*. Perulangan *for* disebut counted loop (perulangan yang terhitung), sementara perulangan *while* disebut uncounted loop (perulangan yang tak terhitung). Perbedaannya adalah perulangan *for* biasanya digunakan untuk mengulangi kode yang sudah diketahui banyak perulangannya. Sementara *while* untuk perulangan yang memiliki syarat dan tidak tentu berapa banyak perulangannya. Contohnya:



Berikut contoh program yang menggunakan perulangan *while*:

Source Code

```
ulangi = 0

while (ulangi < 10):
    print("Kuulangi ", ulangi, " kali")
    ulangi = ulangi + 1
```

Output





Berikut contoh program yang menggunakan perulangan *for*:

[Source Code](#)



```
● ● ●  
  
for i in range(5):  
    print("Nilai i adalah ", i)  
  
for i in range(3,10):  
    print("Kuulangi sebanyak ", i,  
"kali")  
  
for i in range(30,20,-2):  
    print("Aku mundur dari 30 ", i)
```

Output

```
● ● ●  
  
Nilai i adalah 0  
Nilai i adalah 1  
Nilai i adalah 2  
Nilai i adalah 3  
Nilai i adalah 4  
Kuulangi sebanyak 3 kali  
Kuulangi sebanyak 4 kali  
Kuulangi sebanyak 5 kali  
Kuulangi sebanyak 6 kali  
Kuulangi sebanyak 7 kali  
Kuulangi sebanyak 8 kali  
Kuulangi sebanyak 9 kali  
Aku mundur dari 30 30  
Aku mundur dari 30 28  
Aku mundur dari 30 26  
Aku mundur dari 30 24  
Aku mundur dari 30 22
```



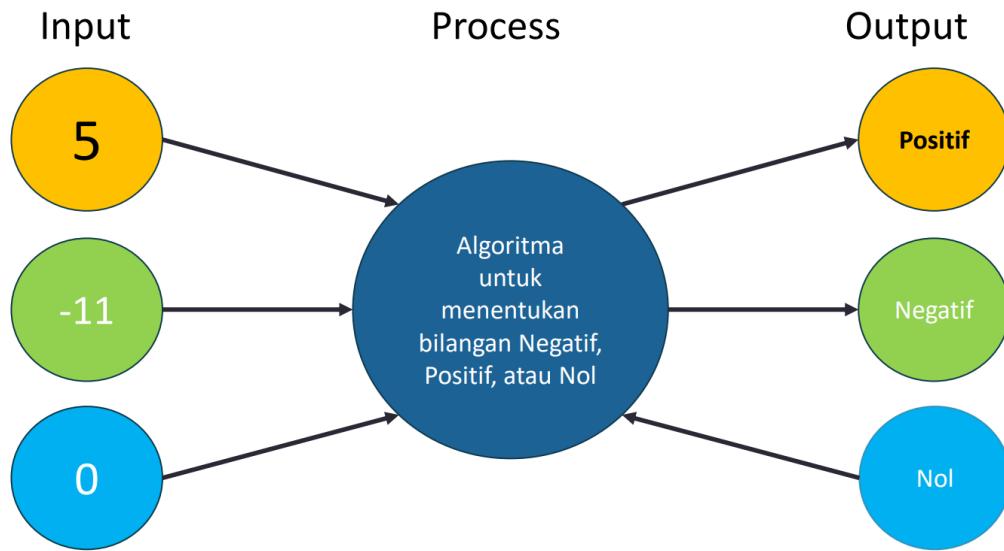
IPO (Input Process Output)

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)



Pseudocode

Pseudocode adalah suatu bentuk deskripsi informal yang mirip dengan bahasa manusia dan digunakan untuk menggambarkan algoritma atau proses secara naratif. Ini tidak terikat pada bahasa pemrograman tertentu, tetapi memberikan panduan tentang langkah-langkah yang harus diambil dalam suatu algoritma dengan bahasa yang lebih mudah dimengerti.



Contoh PseudoCode

Inisiasi Variabel:

```
N      = 0  
total = 0.0
```

Pengulangan:

```
UNTUK i DARI 1 SAMPAI 10 LANGKAH 2  
    CETAK i  
END UNTUK
```

Pengkondisional (Conditional):

```
JIKA nilai > 10  
    CETAK "Nilai lebih dari 10"  
SELAINNYA JIKA nilai = 10  
    CETAK "Nilai sama dengan 10"  
SELAINNYA  
    CETAK "Nilai kurang dari 10"  
AKHIR JIKA
```

Fungsi atau Prosedur:

```
FUNGSI tambah(a, b)  
    KEMBALIKAN a + b  
AKHIR FUNGSI
```

Contoh Lengkap:

```
DEKLARASI variabel n, bilangan, total, rata_rata FLOAT  
MINTA "Masukkan jumlah bilangan: " SIMPAN  
total = 0.0
```

```
UNTUK i DARI 1 SAMPAI n  
    MINTA "Masukkan bilangan ke- " + i + ": " SIMPAN bilangan  
    total = total + bilangan  
END UNTUK  
  
rata_rata = total / n  
CETAK "Rata-rata adalah: " + rata_rata
```

2. Alat dan Bahan

Hardware : Laptop/PC

Software : Spyder (Anaconda Python)

3. Elemen Kompetensi

a. Latihan pertama

Buatlah program dengan keluaran output seperti dibawah ini menggunakan perulangan *for*

```
7777777  
666666  
55555  
4444  
333  
22  
1
```

Pseudocode



Jadi masukkan fungsi looping yaitu for
Setelah itu masukkan looping
For i in range(7,0,-1):
 Tampilkan (str(i)*i)

Input Process Output (IPO)

INPUT
For i in range(7,0,-1)
PROSES
Print(str(i) *i)
OUTPUT
7777777
666666
55555
4444
333
22
1

Source Code



```
▶ print("muhammad rizki")

for i in range(7,0,-1):
    print(str(i) *i)

→ muhammad rizki
7777777
666666
55555
4444
333
22
1
```

Output



```
→ muhammad rizki
7777777
666666
55555
4444
333
22
1
```

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 menggunakan implementasi perulangan *while*.

Pseudocode

- User diminta memasukkan bulan (misalnya, 2 untuk Februari).
- User diminta memasukkan tahun (misalnya, 2024).
- Jika bulan dan tahun valid:
 - Cek apakah tahun 2024 kabisat (ya, karena 2024 dibagi 4).
 - Karena Februari di tahun kabisat, tampilkan: "Bulan 2 pada tahun 2024 memiliki 29 hari."



Input Process Output

Input

1. Input Bulan:

- o Tipe: Integer
- o Rentang: 1 hingga 12 (1 untuk Januari, 2 untuk Februari, dan seterusnya)
- o Validasi: Jika input tidak dalam rentang tersebut, tampilkan pesan kesalahan dan minta input ulang.

2. Input Tahun:

- o Tipe: Integer
- o Tidak ada batasan khusus pada tahun, tetapi diharapkan input yang valid.
- o Validasi: Jika input bukan angka, tampilkan pesan kesalahan dan minta input ulang.

Proses

1. Cek Validitas Input:

- o Gunakan perulangan `while` untuk meminta input bulan dan tahun sampai pengguna memberikan input yang valid.

2. Cek Tahun Kabisat:

- o Implementasikan logika untuk menentukan apakah tahun yang dimasukkan adalah tahun kabisat menggunakan fungsi.

3. Tentukan Jumlah Hari dalam Bulan:

- o Gunakan fungsi yang memeriksa bulan dan tahun untuk menentukan jumlah hari dalam bulan tersebut berdasarkan kriteria:
 - Jika bulan adalah 1, 3, 5, 7, 8, 10, atau 12, maka jumlah hari = 31.
 - Jika bulan adalah 4, 6, 9, atau 11, maka jumlah hari = 30.
 - Jika bulan adalah 2, periksa tahun kabisat untuk menentukan apakah jumlah hari = 28 atau 29.

Output

1. Pesan Hasil:

- o Tampilkan jumlah hari dalam bulan yang diminta dengan format:
 - "Bulan [nomor bulan] pada tahun [tahun] memiliki [jumlah hari] hari."
- o Contoh output:
 - Jika pengguna memasukkan bulan 2 dan tahun 2024:
 - Output: "Bulan 2 pada tahun 2024 memiliki 29 hari."



Source Code

```
def is_leap_year(year):
    return (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)

def get_days_in_month(month, year):
    while True:
        if month in [1, 3, 5, 7, 8, 10, 12]:
            return 31
        elif month in [4, 6, 9, 11]:
            return 30
        elif month == 2:
            if is_leap_year(year):
                return 29
            else:
                return 28
        else:
            return None

while True:
    try:
        month = int(input("Masukkan nomor bulan (1-12): "))
        if month < 1 or month > 12:
            print("Nomor bulan harus antara 1 dan 12.")
            continue
        year = int(input("Masukkan tahun: "))
        break
    except ValueError:
        print("Input tidak valid. Masukkan angka yang benar untuk bulan dan tahun.")

days = get_days_in_month(month, year)

# Menampilkan hasil
if days:
    print(f"Bulan {month} pada tahun {year} memiliki {days} hari.")
else:
    print("Input bulan tidak valid.")
```



Output

```
↓ Masukkan nomor bulan (1-12): 11
↓ Masukkan tahun: 2000
Bulan 11 pada tahun 2000 memiliki 30 hari.
```

4. File Praktikum

Github Repository:

```
https://github.com/muhrizki26/praktikum-algo-modul-4.git
```

5. Soal Latihan

Soal:

1. Dalam perulangan while kondisi apakah yang menyebabkan *infinite loop*? Jelaskan dan berikan contohnya!
2. Deskripsikan serta narasikan jalannya alur source code program yang sebelumnya telah kalian buat pada Elemen Kompetensi Latihan Kedua!

Jawaban:



- Infinite Loop dalam Perulangan while:** Infinite loop pada perulangan `while` terjadi ketika kondisi yang memicu penghentian loop tidak pernah terpenuhi, sehingga loop terus berjalan tanpa akhir. Hal ini biasanya disebabkan oleh kondisi yang selalu bernilai benar (`True`) atau jika variabel yang digunakan untuk menghentikan loop tidak diperbarui dengan benar.

```
python
i = 0
while i < 5:
    print("Looping terus")
# Tidak ada perubahan nilai i, menyebabkan loop tidak berhenti
```

Ini nomor 2.Fungsi Tahun Kabisat:

- `is_leap_year(year)`: Mengecek apakah tahun yang dimasukkan adalah tahun kabisat (mengembalikan nilai benar atau salah).
- Fungsi Jumlah Hari:**
 - `get_days_in_month(month, year)`: Menentukan jumlah hari dalam bulan berdasarkan bulan dan tahun.
 - 31 hari untuk bulan Januari, Maret, Mei, Juli, Agustus, Oktober, Desember.
 - 30 hari untuk bulan April, Juni, September, November.
 - 28 atau 29 hari untuk bulan Februari, tergantung apakah tahun tersebut kabisat.
 - Input dari Pengguna:**
 - Program meminta pengguna memasukkan nomor bulan dan tahun.
 - Memeriksa validitas input; jika tidak valid, tampilkan pesan kesalahan dan ulangi.
 - Penghitungan dan Output:**
 - Setelah input valid, program memanggil `get_days_in_month`.
 - Menampilkan hasil: "Bulan [nomor bulan] pada tahun [tahun] memiliki [jumlah hari] hari."

Narasi Alur Program

Program dimulai dengan meminta pengguna memasukkan nomor bulan (misalnya, 2 untuk Februari) dan tahun (misalnya, 2024). Setelah memvalidasi input, program akan menentukan apakah tahun tersebut kabisat. Jika kabisat, Februari akan memiliki 29 hari. Program kemudian mencetak hasilnya: "Bulan 2 pada tahun 2024 memiliki 29 hari." Jika pengguna memasukkan bulan yang tidak valid, program akan memberi tahu dan meminta input ulang. Program ini memastikan hasil yang akurat dan memberikan pengalaman pengguna yang baik.

6. Kesimpulan



- a. Dalam pengerjaan program dengan bahasa pemrograman Python, kita harus benar-benar teliti dalam menginputkan suatu fungsi untuk menampilkan suatu keluaran pada layar dengan sesuai.

Kita dapat mengetahui fungsi looping yaitu pake for dan while jadi kita mengidentifikasi suatu angka

Looping **for** digunakan ketika jumlah iterasi sudah diketahui, sedangkan **while** cocok untuk iterasi yang bergantung pada kondisi tertentu. **For** lebih efisien untuk iterasi terdefinisi, sedangkan **while** memberikan fleksibilitas lebih, tapi harus diwaspadai agar tidak terjadi infinite loop.

1. Cek List (✓)

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

2. Formulir Umpam Balik

No	Elemen Kompetensi	Waktu Pengerjaan	Kriteria
1.	Latihan Pertama	15 Menit	menarik
2.	Latihan Kedua	20 Menit	baik

Keterangan:

1. Menarik
2. Baik
3. Cukup
4. Kurang
- 5.

