

<p>Nama: Steven Raja P M NIM: 064102400018</p>	 <p>Praktikum Algoritma & Pemrograman</p>	<h1>MODUL 4</h1> <p>Nama Dosen: Binti solihah, S.T, M.KOM</p> <p>Nama Asisten Labratorium:</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
```



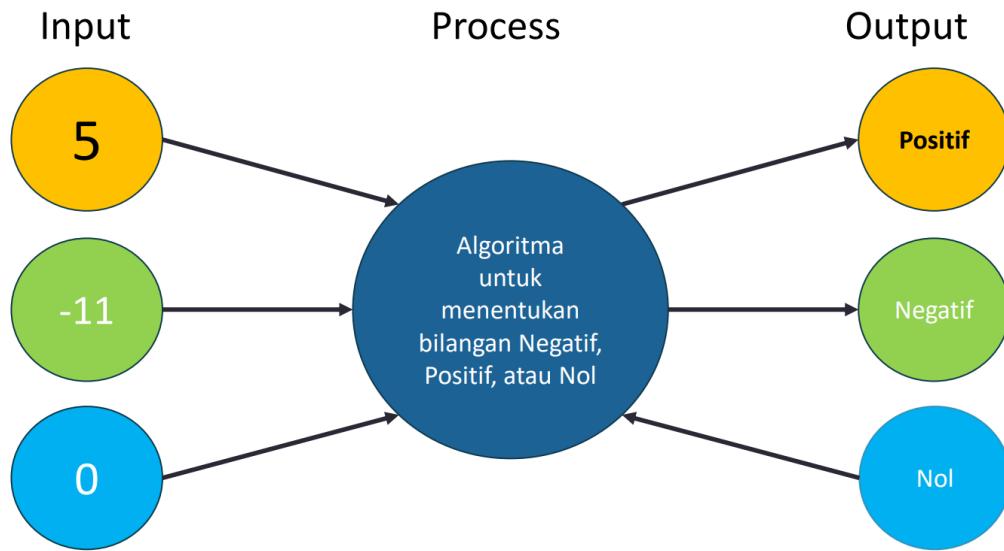
IPO (Input Process Output)

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



for i in range itu untuk looping(pengulangan)
mulai dari 7 sampai 1 0 tidak termasuk
-1 itu untuk mengurangi setiap satu kali
Str(i) untuk mengubah angka i menjadi string
Misalnya i nya adalah 7 berarti mulai dari 7777777 dan seterusnya
Print untuk mencetak hasil ke layar

Input Process Output (IPO)

```
/s ➔ for i in range(7, 0, -1):  
    print(str(i) * i)
```

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

Source Code



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

Output



```
→ 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

```
Buat daftar hari per bulan dari januari sampai desember,jika tahun kabisat ubah februari menjadi 29 setelah itu buat variabel untuk memasukan data bulan 1 sampai 12 dan tahun,jika hasilnya valid maka ditampilkan hari pada bulan tersebut dan jika tidak valid maka akan menampilkan bulan tidak valid,untuk y/n itu untuk memilih jika y maka dilakukan pengulangan dan jika n maka program selesai
```



Input Process Output

```
def jumlah_hari(bulan, tahun):
    hari_per_bulan = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]

    if (tahun % 4 == 0 and tahun % 100 != 0) or (tahun % 400 == 0):
        hari_per_bulan[1] = 29 # Februari pada tahun kabisat

    if 1 <= bulan <= 12:
        return hari_per_bulan[bulan - 1]
    else:
        return None

while True:
    try:
        bulan = int(input("Masukkan bulan (1-12): "))
        tahun = int(input("Masukkan tahun: "))

        hari = jumlah_hari(bulan, tahun)

        if hari is not None:
            print(f"Bulan {bulan} tahun {tahun} memiliki {hari} hari.")
        else:
            print("Bulan tidak valid. Silakan masukkan bulan antara 1 dan 12.")

        keluar = input("Apakah Anda ingin mengulangi? (y/n): ")
        .lower()
        if keluar != 'y':
            break
    except ValueError:
        print("Input tidak valid. Silakan masukkan angka.")
```

```
→ Masukkan bulan (1-12): 10
Masukkan tahun: 2004
Bulan 10 tahun 2004 memiliki 31 hari.
Apakah Anda ingin mengulangi? (y/n): y
Masukkan bulan (1-12): 11
Masukkan tahun: 2005
Bulan 11 tahun 2005 memiliki 30 hari.
Apakah Anda ingin mengulangi? (y/n): n
```



Source Code

```
▶ def jumlah_hari(bulan, tahun):
    hari_per_bulan = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]

    if (tahun % 4 == 0 and tahun % 100 != 0) or (tahun % 400 == 0):
        hari_per_bulan[1] = 29 # Februari pada tahun kabisat

    if 1 <= bulan <= 12:
        return hari_per_bulan[bulan - 1]
    else:
        return None

while True:
    try:
        bulan = int(input("Masukkan bulan (1-12): "))
        tahun = int(input("Masukkan tahun: "))

        hari = jumlah_hari(bulan, tahun)

        if hari is not None:
            print(f"Bulan {bulan} tahun {tahun} memiliki {hari} hari.")
        else:
            print("Bulan tidak valid. Silakan masukkan bulan antara 1 dan 12.")

        keluar = input("Apakah Anda ingin mengulangi? (y/n): ").lower()
        if keluar != 'y':
            break
    except ValueError:
        print("Input tidak valid. Silakan masukkan angka.")
```

Output



```
→ Masukkan bulan (1-12): 10
Masukkan tahun: 2004
Bulan 10 tahun 2004 memiliki 31 hari.
Apakah Anda ingin mengulangi? (y/n): y
Masukkan bulan (1-12): 11
Masukkan tahun: 2005
Bulan 11 tahun 2005 memiliki 30 hari.
Apakah Anda ingin mengulangi? (y/n): n
```

4. File Praktikum

Github Repository:

<https://github.com/Stevenraja/Prak4Algoritma.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:

1. Ketika kondisi selalu True dan tidak ada mekanisme untuk mengubah menjadi False

Contoh: x = 0

while x < 5:

 print(x)

Tidak ada perubahan pada x

2. Buat daftar hari per bulan dari januari sampai desember,jika tahun kabisat ubah februari menjadi 29 setelah itu buat variabel untuk memasukan data bulan 1 sampai 12 dan tahun,jika hasilnya valid maka ditampilkan hari pada bulan tersebut dan jika tidak valid maka akan menampilkan bulan tidak valid,untuk y/n itu untuk memilih jika y maka dilakukan pengulangan dan jika n maka program selesai



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 bagaimana cara menggunakan looping dan program ini efektif dalam menyelesaikan tugas yang diberikan dan menunjukkan penerapan konsep pemrograman yang baik, seperti penggunaan fungsi, pengelolaan input, dan perulangan

7. Cek List (✓)

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

8. Formulir Umpam Balik

No	Elemen Kompetensi	Waktu Pengerjaan	Kriteria
1.	Latihan Pertama	20 Menit	Menarik
2.	Latihan Kedua	20 Menit	Menarik

Keterangan:

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

