


<p>Nama: Aini Rihhadatul Aisy</p> <p>NIM: 064102400024</p>	 <p>Praktikum Algoritma & Pemrograman</p>	<p>MODUL 4</p> <p>Nama Dosen: Binti solihah, S.T, M.KOM</p>
<p>Hari/Tanggal: Jumat, 11 Oktober 2024</p>		<p>Nama Asisten Labratorium:</p> <ol style="list-style-type: none"> 1. Yustianas Rombon - 064002300015 2. Vira 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

```
ulangan = 0

while (ulangan < 10):
    print("Kuulangan ", 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

```



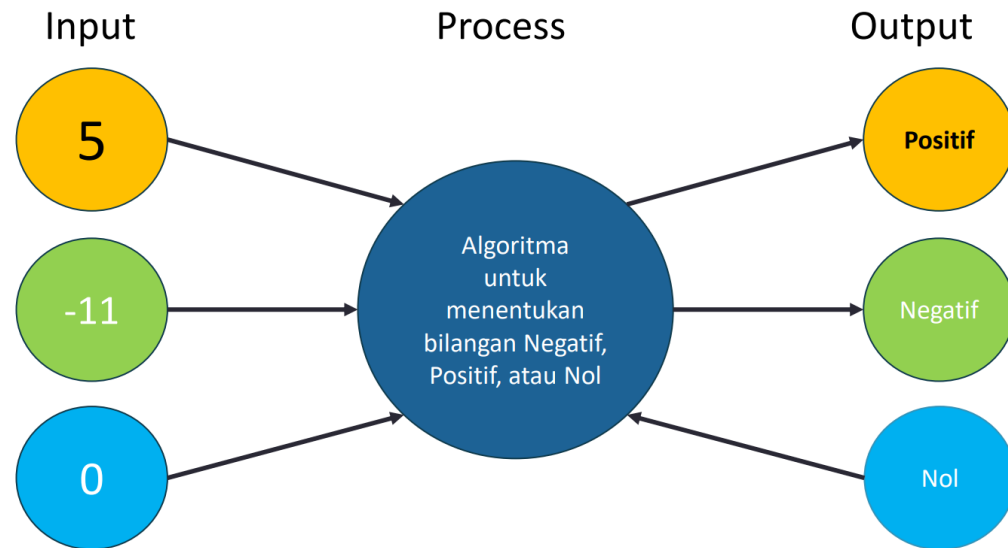
IPO (Input Process Output)

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
6666666
55555
4444
333
22
1
```

Pseudocode



Untuk i(variabel) dengan batas angka awal 7 sampai batas angka akhir 0 dan setiap pengulangan dikurangi 1 angka
Tampilkan/cetak i menjadi tipe data string, dan dikali dengan tiap masing-masing string/angka sebanyak angka tersebut

Input Process Output (IPO)

Input (tidak ada)

Proses

- membuat loop yang menghitung dari 7 hingga 1 (setiap langkah selalu -1).
- Untuk setiap angka i, cetak angka tersebut sebanyak i kali sebagai string.

Output

7777777

666666

55555

4444

333

22

1

Source Code

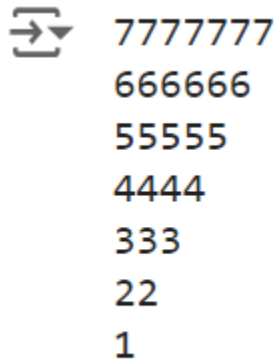




```
for i in range(7, 0, -1):  
    print(str(i) * i)
```

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

```

START

// Meminta input bulan dan tahun dari pengguna
PRINT "Masukkan bulan (1-12):"
READ bulan
PRINT "Masukkan tahun:"
READ tahun

// Inisialisasi jumlah_hari
SET jumlah_hari = 0

// Memulai perulangan untuk validasi dan penghitungan
WHILE TRUE DO
    // Memeriksa validitas bulan

```



```

IF bulan < 1 OR bulan > 12 THEN
    PRINT "Bulan tidak valid. Silakan masukkan bulan antara 1 dan 12."
    PRINT "Masukkan bulan (1-12):"
    READ bulan
    CONTINUE // Kembali ke awal perulangan

// Menentukan jumlah hari berdasarkan bulan
IF bulan IN [1, 3, 5, 7, 8, 10, 12] THEN
    SET jumlah_hari = 31
    BREAK // Keluar dari perulangan
ELSE IF bulan IN [4, 6, 9, 11] THEN
    SET jumlah_hari = 30
    BREAK // Keluar dari perulangan
ELSE IF bulan == 2 THEN
    // Memeriksa tahun kabisat
    IF (tahun MOD 4 == 0 AND tahun MOD 100 != 0) OR (tahun MOD 400 == 0)
    THEN
        SET jumlah_hari = 29
    ELSE
        SET jumlah_hari = 28
    ENDIF
    BREAK // Keluar dari perulangan
ENDIF
ENDWHILE

PRINT "Jumlah hari dalam bulan ", bulan, " tahun ", tahun, " adalah: ", jumlah_hari, "
hari."

END

```

Input Process Output



Input:

1. Bulan (integer, antara 1 hingga 12).
2. Tahun (integer).

Proses:

1. Memeriksa validitas bulan yang dimasukkan:
 - Jika bulan kurang dari 1 atau lebih dari 12, minta pengguna memasukkan bulan yang valid.
2. Menentukan jumlah hari dalam bulan berdasarkan input:
 - Jika bulan adalah 1, 3, 5, 7, 8, 10, atau 12, jumlah hari adalah 31.
 - Jika bulan adalah 4, 6, 9, atau 11, jumlah hari adalah 30.
 - Jika bulan adalah 2, periksa tahun untuk menentukan apakah itu tahun kabisat:
 - Jika tahun kabisat, jumlah hari adalah 29.
 - Jika tidak, jumlah hari adalah 28.

Output:

- Menampilkan jumlah hari dalam bulan dan tahun yang dimasukkan oleh pengguna dalam format yang jelas.

Source Code

```

bulan = int(input("Masukkan bulan (1-12): "))
tahun = int(input("Masukkan tahun: "))
jumlah_hari = 0
while True:
    if bulan < 1 or bulan > 12:
        print("Bulan tidak valid. Silakan masukkan bulan antara 1 dan 12.")
        bulan = int(input("Masukkan bulan (1-12): "))
        continue

    if bulan in [1, 3, 5, 7, 8, 10, 12]:
        jumlah_hari = 31
        break
    elif bulan in [4, 6, 9, 11]:
        jumlah_hari = 30
        break
    elif bulan == 2:
        if (tahun % 4 == 0 and tahun % 100 != 0) or (tahun % 400 == 0):
            jumlah_hari = 29
        else:
            jumlah_hari = 28
        break
    print(f"Jumlah hari dalam bulan {bulan} tahun {tahun} adalah: {jumlah_hari} hari.")
  
```

✓ 14s completed at 9:35PM



Output

```
Masukkan bulan (1-12): 19
Masukkan tahun: 2023
Bulan tidak valid. Silakan masukkan bulan antara 1 dan 12.
Masukkan bulan (1-12): 9
Jumlah hari dalam bulan 9 tahun 2023 adalah: 30 hari.
```

4. File Praktikum

Github Repository:

```
print("Taruh link github repository praktikum kalian disini yaa...")
```

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. Infinite loop dalam perulangan while terjadi ketika kondisi yang diuji dalam pernyataan while selalu bernilai benar (true). biasanya terjadi bisa karena kondisi yang digunakan berubah dalam perulangan yang menyebabkan tidak bisa keluar dari



looping(tidak pernah salah) dan bisa juga karena selalu menggunakan ekspresi yang selalu menghasilkan nilai benar(selalu benar)

2. **Input Bulan dan Tahun** : Program mulai dengan meminta pengguna untuk memasukkan bulan (dalam rentang 1-12) dan tahun. Input ini disimpan dalam variabel bulan dan tahun.

Inisialisasi Variabel: Variabel jumlah_hari diinisialisasi dengan nilai 0. Ini akan digunakan untuk menyimpan jumlah hari dalam bulan yang ditentukan.

Perulangan while:

Program memasuki perulangan while True, yang akan terus berjalan sampai perintah break dijalankan.

Validasi Bulan:

Di dalam perulangan, program pertama-tama memeriksa apakah nilai bulan valid (yaitu antara 1 dan 12).

Jika bulan tidak valid, program akan mencetak pesan kesalahan dan meminta pengguna untuk memasukkan bulan lagi. Setelah input baru dimasukkan, program akan melanjutkan ke iterasi berikutnya dengan menggunakan continue.

Penentuan Jumlah Hari: Jika bulan valid, program kemudian memeriksa bulan yang dimasukkan:

Bulan dengan 31 Hari: Jika bulan adalah 1, 3, 5, 7, 8, 10, atau 12, program akan menetapkan jumlah_hari menjadi 31 dan menjalankan break untuk keluar dari perulangan.

Bulan dengan 30 Hari: Jika bulan adalah 4, 6, 9, atau 11, program akan menetapkan jumlah_hari menjadi 30 dan juga keluar dari perulangan.

Bulan Februari: Jika bulan adalah 2, program perlu memeriksa apakah tahun tersebut adalah tahun kabisat:

Jika tahun kabisat, jumlah_hari diatur menjadi 29.

Jika tidak, jumlah_hari diatur menjadi 28.

Setelah itu, program keluar dari perulangan.

Output Hasil: Setelah keluar dari perulangan, program mencetak jumlah hari dalam bulan dan tahun yang telah dimasukkan oleh pengguna.

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.
- a. Kita dapat mengetahui cara untuk memakai keadaan while dalam praktikum kali ini, dan menurut saya cukup sulit untuk menggunakan keadaan while ini karena harus butuh ketelitian dalam mengerjakannya.



b.

7. Cek List (✓)

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

8. Formulir Umpan Balik

No	Elemen Kompetensi	Waktu Pengerjaan	Kriteria
1.	Latihan Pertama	30 Menit	Menarik
2.	Latihan Kedua	30 Menit	Menarik

Keterangan:

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