


<p><b>Nama:</b> <b>ROBIN FELIX HAMA</b></p> <p><b>NIM:</b> <b>064102400028</b></p>	 <p><b>Praktikum Algoritma &amp; Pemrograman</b></p>	<p><b>MODUL 5</b></p> <p><b>Nama Dosen:</b> <b>Binti solihah, S.T, M.KOM</b></p>
<p><b>Hari/Tanggal: (Isi Tanggal)</b> <b>Jumat, 18 Oktober 2024</b></p>		<p><b>Nama Asisten Labratorium:</b></p> <ol style="list-style-type: none"> <li><b>Yustianas Rombon - 064002300015</b></li> <li><b>Vira Aditya Kurniawan - 065002300012</b></li> </ol>

## Latihan – Struktur Kendali

### 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). Perbedaanannya 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

```

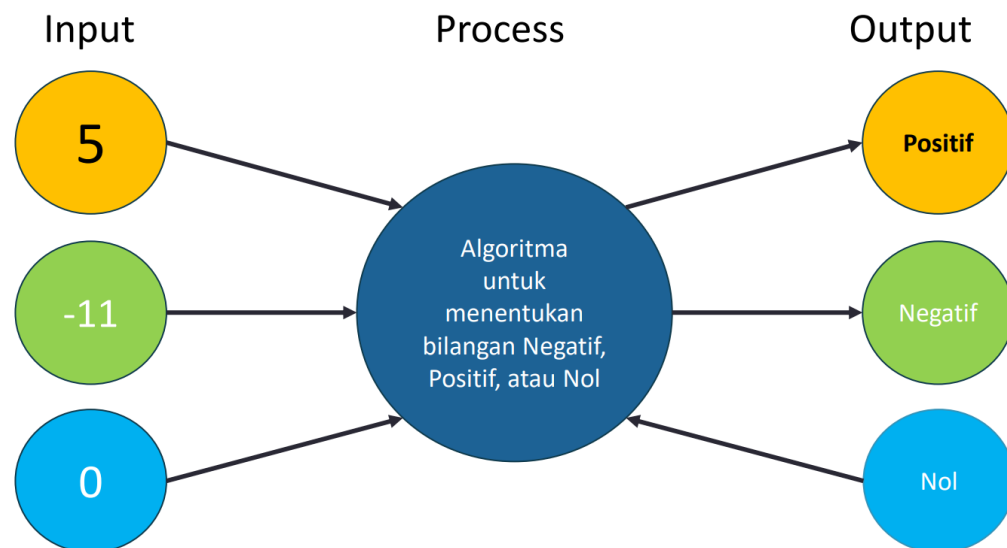


## IPO (Input Process Output)

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 fungsi untuk merata-ratakan nilai sesuai dengan kategori huruf yang diinputkan dimana aturannya adalah sebagai berikut: (Exercise 66)

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

Implementasikan apa yang kalian pelajari pada materi-materi di modul-modul sebelumnya

Input Process Output



```

nilai = 0
counter = 0

while True :
    Implementasi = input("Grade :")

    if Implementasi == "A":
        counter += 1
        nilai += 4.00
        continue
    elif Implementasi == "A-":
        counter += 1
        nilai += 3.75
        continue
    elif Implementasi == "B+":
        counter += 1
        nilai += 3.50
        continue
    elif Implementasi == "B":
        counter += 1
        nilai += 3.00
        continue
    elif Implementasi == "B-":
        counter += 1
        nilai += 2.75
        continue
    elif Implementasi == "C+":
        counter += 1
        nilai += 2.50
        continue
    elif Implementasi == "C":
        counter += 1
        nilai += 2.00
        continue
    elif Implementasi == "C-":
        counter += 1
        nilai += 1.75
        continue
    elif Implementasi == "D":
        counter += 1
        nilai += 1.50
        continue

```



```
elif Implementasi == "E":
    counter += 1
    nilai += 1.25
    continue
else:
    rata_rata = nilai / counter
    print("Rata-rata nilai:", rata_rata)
    break
```

#### Pseudocode

- Masukan Nilai untuk menyimpan nilai implementasi
- Nilai = 0
- Masukan Counter untuk menyimpan nilai implementasi
- Counter = 0
- Masukan While Supaya loop akan terus berjalan
- While
- Masukan if untuk memeriksa nilai input(grade)
- if
- Masukan elif untuk memeriksa nilai input(grade)
- Elif
- Masukan else untuk menangani kondisi ketika input yang diberikan tidak cocok dengan grade yang terdaftar
- Masukan print untuk mengeluarkan hasil loop
- print("Rata-rata nilai:", rata\_rata)
- Masukan break agar menghentikan hasil loop





### Source Code

```
nilai = 0  
counter = 0  
  
while True :  
    Implementasi = input("Grade :")  
  
    if Implementasi == "A":  
        counter += 1  
        nilai += 4.00  
        continue  
    elif Implementasi == "A-":  
        counter += 1  
        nilai += 3.75  
        continue  
    elif Implementasi == "B+":  
        counter += 1  
        nilai += 3.50  
        continue  
    elif Implementasi == "B":  
        counter += 1  
        nilai += 3.00  
        continue
```



```
elif Implementasi == "B-":
```

```
    counter += 1
```

```
    nilai += 2.75
```

```
    continue
```

```
elif Implementasi == "C+":
```

```
    counter += 1
```

```
    nilai += 2.50
```

```
    continue
```

```
elif Implementasi == "C":
```

```
    counter += 1
```

```
    nilai += 2.00
```

```
    continue
```

```
elif Implementasi == "C-":
```

```
    counter += 1
```

```
    nilai += 1.75
```

```
    continue
```

```
elif Implementasi == "D":
```

```
    counter += 1
```

```
    nilai += 1.50
```

```
    continue
```

```
elif Implementasi == "E":
```

```
    counter += 1
```

```
    nilai += 1.25
```

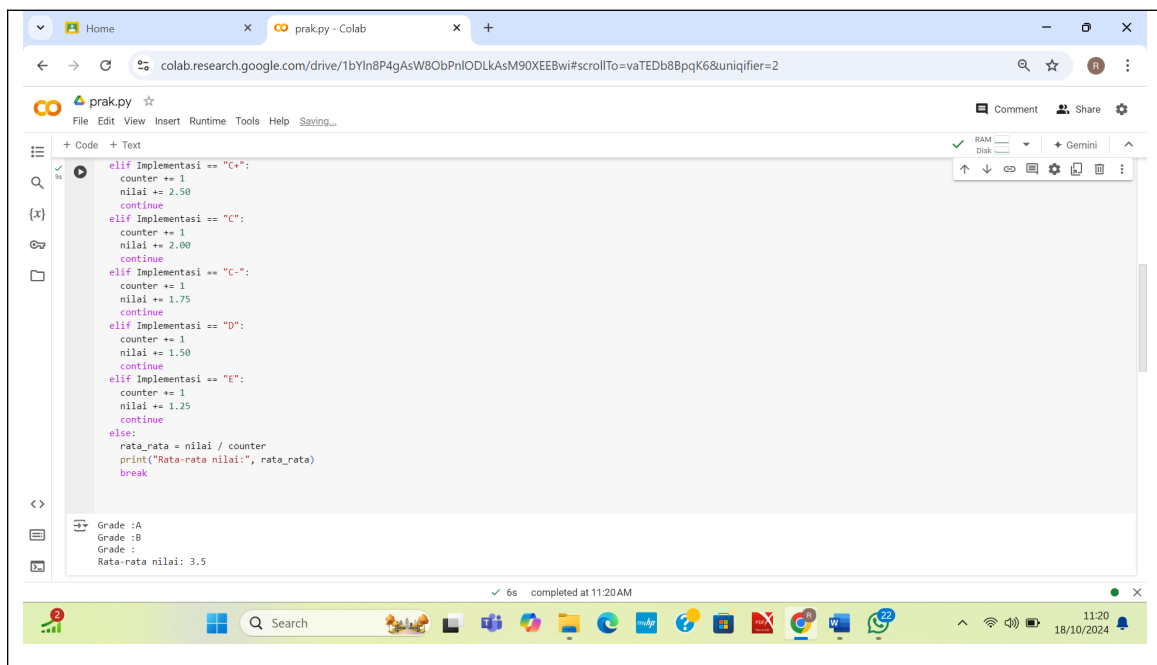
```
    continue
```

```
else:
```



```
rata_rata = nilai / counter  
  
print("Rata-rata nilai:", rata_rata)  
  
break
```

## Output



The screenshot shows a Google Colab notebook interface. The code cell contains a loop that calculates the average of grades based on letter grades. The output cell shows the results of the loop.

```
elif Implementasi == "C+":  
    counter += 1  
    nilai += 2.50  
    continue  
elif Implementasi == "C":  
    counter += 1  
    nilai += 2.00  
    continue  
elif Implementasi == "C-":  
    counter += 1  
    nilai += 1.75  
    continue  
elif Implementasi == "D":  
    counter += 1  
    nilai += 1.50  
    continue  
elif Implementasi == "D-":  
    counter += 1  
    nilai += 1.25  
    continue  
else:  
    rata_rata = nilai / counter  
    print("Rata-rata nilai:", rata_rata)  
    break
```

Grade :A  
Grade :B  
Grade :  
Rata-rata nilai: 3.5

6s completed at 11:20 AM

## b. Latihan Kedua



Buatlah program untuk menghitung Tiket Masuk Kebun Binatang Berdasarkan Umur beserta Pembayarannya dengan aturan sebagai berikut: (Exercise 67)

- Umur yang kurang dari atau sama dengan 2 tahun digratiskan
- Umur yang lebih dari atau sama dengan 3 tahun hingga umur yang kurang dari atau sama dengan 12 tahun seharga 14 dollar
- Umur lebih dari atau sama dengan 65 tahun seharga 18 dollar
- Dan selain ketiga kategori diatas harganya normal yaitu 23 dollar

Program harus menggunakan perulangan untuk menghitung total keseluruhan harga yang diinputkan secara berulang, dan juga jika uang yang diinputkan berlebih maka program wajib mengembalikan nilai kembalian uang tersebut.

Input Process Output

```
def hitung_tiket(umur):
    if umur <= 2:
        return 0
    elif 3 <= umur <= 12:
        return 14
    elif umur >= 65:
        return 18
    else:
        return 23

def main():
    total_harga = 0
    while True:
        try:
            umur = int(input("Masukkan umur (atau -1 untuk keluar): "))
            if umur == -1:
                break

            harga = hitung_tiket(umur)
            total_harga += harga
            print(f'Harga tiket untuk umur {umur} adalah ${harga}.')
        except ValueError:
            print("Input tidak valid, silakan masukkan angka yang benar.")

    if total_harga > 0:
```



```
print(f'Total harga tiket adalah ${total_harga}.')
while True:
    try:
        uang = float(input("Masukkan uang yang dibayarkan: "))
        if uang < total_harga:
            print("Uang yang dimasukkan tidak cukup.")
        else:
            kembalian = uang - total_harga
            print(f'Kembalian Anda adalah ${kembalian:.2f}.')
            break
    except ValueError:
        print("Input tidak valid, silakan masukkan angka yang benar.")

if __name__ == "__main__":
    main()
```

#### Pseudocode

- Mendefinisikan harga tiket berdasarkan umur
- def hitung\_tiket(umur)
- Memakai if untuk mengecek apakah umur pengguna kurang dari atau sama dengan 2 tahun. Jika ya, fungsi mengembalikan harga tiket 0.
- if umur <=2
- Memakai elif jika kondisi pertama tidak terpenuhi, program memeriksa apakah umur berada di antara 3 hingga 12 tahun. Jika ya, harga tiket yang dikembalikan adalah \$14.
- elif 3<= umur <=12:
- Memakai else jika semua kondisi sebelumnya tidak terpenuhi, berarti umur berada dalam rentang 13 hingga 64 tahun. Dalam kasus ini, fungsi mengembalikan harga tiket \$23.
- else : return 23
- Memakai while digunakan untuk membuat loop tak hingga yang terus meminta input pengguna sampai kondisi tertentu terpenuhi (yaitu, pengguna memasukkan -1 untuk keluar).
- While True
- Print untuk mengeluarkan hasil

#### Source Code



```
def hitung_tiket(umur):  
    if umur <= 2:  
        return 0  
    elif 3 <= umur <= 12:  
        return 14  
    elif umur >= 65:  
        return 18  
    else:  
        return 23  
  
def main():  
    total_harga = 0  
    while True:  
        try:  
            umur = int(input("Masukkan umur (atau -1 untuk keluar): "))  
            if umur == -1:  
                break  
            harga = hitung_tiket(umur)  
            total_harga += harga  
            print(f"Harga tiket untuk umur {umur} adalah ${harga}.")  
        except ValueError:  
            print("Input tidak valid, silakan masukkan angka yang benar.")  
  
    if total_harga > 0:
```



```

_____ print(f'Total harga tiket adalah ${total_harga}.')
_____ while True:
_____     try:
_____         uang = float(input("Masukkan uang yang dibayarkan: "))
_____         if uang < total_harga:
_____             print("Uang yang dimasukkan tidak cukup.")
_____         else:
_____             kembalian = uang - total_harga
_____             print(f'Kembalian Anda adalah ${kembalian:.2f}.')
_____             break
_____     except ValueError:
_____         print("Input tidak valid, silakan masukkan angka yang benar.")

if __name__ == "__main__":
    __main__()

```

### Output



```
def main():
    total_harga = 0
    while True:
        try:
            umur = int(input("Masukkan umur (atau -1 untuk keluar): "))
            if umur == -1:
                break
            harga = hitung_tiket(umur)
            total_harga += harga
            print(f"Harga tiket untuk umur {umur} adalah ${harga}.")
        except ValueError:
            print("Input tidak valid, silakan masukkan angka yang benar.")

    if total_harga > 0:
        print(f"Total harga tiket adalah ${total_harga}.")
        while True:
            try:
                uang = float(input("Masukkan uang yang dibayarkan: "))
                if uang < total_harga:
                    print("Uang yang dimasukkan tidak cukup.")
                else:
                    kembalian = uang - total_harga
                    print(f"Kembalian Anda adalah ${kembalian:.2f}.")
                    break
            except ValueError:
                print("Input tidak valid, silakan masukkan angka yang benar.")

    if __name__ == "__main__":
        main()

*** Masukkan umur (atau -1 untuk keluar): 17
Harga tiket untuk umur 17 adalah $23.
```

#### 4. File Praktikum

Github Repository:

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

#### 5. Soal Latihan

Soal:

1. Apa yang dimaksud dengan kondisi percabangan bersarang dan seperti apa contoh implementasinya?
2. Deskripsikan serta narasikan jalannya alur source code program yang sebelumnya telah kalian buat pada Elemen Kompetensi Latihan Kedua!

Jawaban:

1. Kondisi percabangan adalah adalah struktur kontrol dalam pemrograman di mana terdapat percabangan (if) di dalam percabangan lain. Contoh implementasinya Misalnya, kita ingin menentukan status kelulusan seorang siswa berdasarkan nilai ujian. Jika nilai ujian lebih dari atau sama dengan 75, kita perlu memeriksa apakah siswa tersebut juga telah menyelesaikan semua tugas.
2. Program ini adalah sebuah aplikasi sederhana untuk menghitung harga tiket berdasarkan umur pengunjung dan mengelola pembayaran.

#### 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.
- b. Kita dapat mengetahui... (Tolong Isi lebih dari dua baris!)

## 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	90 Menit	Menarik
2.	Latihan Kedua	90 Menit	Menarik

Keterangan:

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