


<p><b>Nama:</b> (MUHAMMAD RIZKI)</p> <p><b>NIM:</b> (064102400020)</p>	 <p><b>Praktikum Algoritma &amp; Pemrograman</b></p>	<p><b>MODUL 2</b></p> <p><b>Nama Dosen:</b> Anung B. Ariwibowo, M. Kom</p>
<p><b>Hari/Tanggal:</b> Jum'at, 27 september 2024</p>		<p><b>Nama Asisten Labratorium:</b></p> <ol style="list-style-type: none"> <li>1. Yustianas Rombon - 064002300015</li> <li>2. Vira Aditya Kurniawan - 065002300012</li> </ol>

## Operasi Aritmatika pada Python

### 1. Teori Singkat

#### Operasi Aritmatika


Operasi Aritmatika adalah bagian dari pengolahan bilangan dari sebuah komputer untuk melakukan operasi hitung. Selain melakukan operasi hitung, operasi aritmatika juga bisa dilakukan untuk operasi logika. Dasar melakukan operasi hitung dalam aritmatika komputer adalah penjumlahan atau yang disebut adder

Berikut ini adalah Operator Aritmatika pada Bahasa Pemrograman Python

Operator	Simbol
Penjumlahan	+
Pengurangan	-
Perkalian	*
Pembagian	/
Sisa Bagi	%
Pemangkatan	**

Berikut ini adalah contoh sederhana program operasi aritmatika:

#### Anaconda Prompt



```
(base) C:\Users\Azhar Rizky Zulma>python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC
v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license"
for more information.
>>> 3 + 2
5
>>> 18 % 5
3
>>> abs(-7)
7
>>> float(9)
9.0
>>> int(5.3)
5
>>> complex(1, 2)
(1+2j)
>>> 2 ** 8
256
```

Operator matematik berfungsi dengan normal di Python seperti dalam bahasa pemrograman yang lain. Ada beberapa catatan yang harus diperhatikan.

- Operasi pengisian beberapa variabel dengan nilai yang sama dapat dilakukan sekali jalan.
- Tanda `()` dipakai untuk mengelompokkan operasi yang harus dilakukan terlebih dahulu.



- Pembagian bilangan integer dengan bilangan integer akan dibulatkan ke bawah.
- Bilangan integer akan dikonversikan menjadi bilangan floating point dalam operasi yang melibatkan bilangan integer dan bilangan floating point.
- Kita tidak dapat mengkonversikan bilangan kompleks ke bilangan real (floating point atau integer); hanya bilangan mutlaknya yang bisa kita dapatkan.

## Operator Penugasan

Seperti namanya, operator ini digunakan untuk memberikan tugas pada variabel. Misalnya:

umur = 18

Maka variabel umur telah kita berikan tugas untuk menyimpan angka 18. Selain menyimpan atau pengisian nilai, ada juga menjumlahkan, mengurangi, perkalian, pembagian, dsb. Selengkapnya bisa dilihat di tabel berikut.

Operator	Simbol
Pengisian	=
Penjumlahan	+=
Pengurangan	-=
Perkalian	*=
Pembagian	/=
Sisa Bagi	%=
Pemangkatan	**=

Karena pada python tidak terdapat operator increment & decrement maka operator tersebut digantikan oleh operator penugasan dengan menginputkan seperti ini  $i += 1$ .

## Python Package

Package merupakan sebuah bundle atau pengelompokan dari banyak fungsi serta kelas (source code) menjadi satu kesatuan unit tunggal dalam library yang dapat digunakan serta dipanggil pada source code yang sedang anda kembangkan untuk mendapatkan sebuah fungsi tanpa harus mengetikkan source codenya secara berulang. Python sendiri menyediakan package untuk menyediakan operasi-operasi standar. Untuk operasi-operasi yang lebih khusus, perlu menggunakan fungsi dari package lain. Dalam praktikum kali ini kita akan belajar operasi



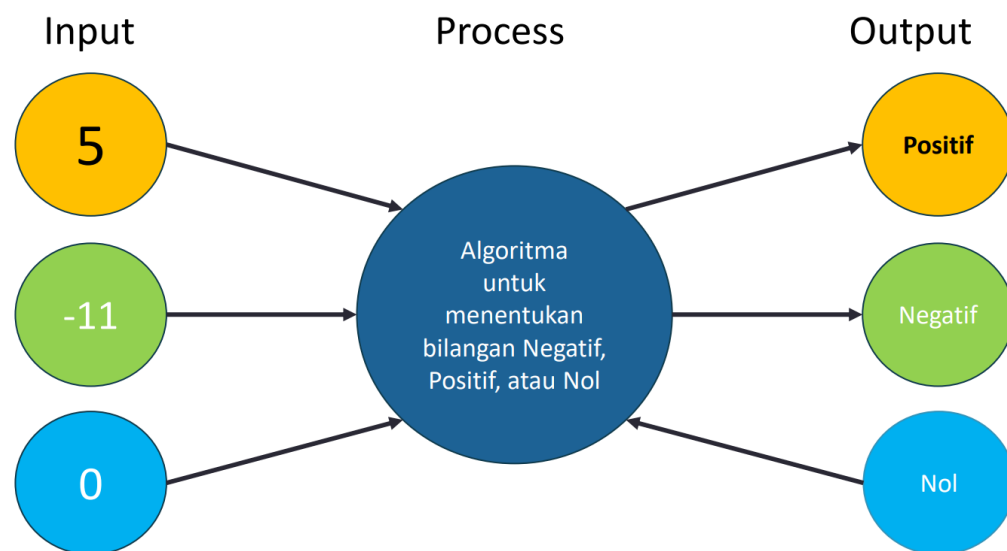
aritmatika dan belajar tentang menggunakan package untuk memanggil fungsi operasi trigonometri yang packagenya sudah disediakan pada Python yaitu dalam package *Math*:

## 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 sebuah program yang menerima dua bilangan bulat a dan b dari user. Program anda harus menampilkan hasil perhitungan:

- Jumlah a dan b
- Selisih antara b dengan a
- Hasil kali a dan b
- Sisa pembagian a dengan b
- Pembagian a dengan b
- Hasil dari log(a)
- a pangkat b

Gunakan fungsi log10() dari package math

Input Process Output (IPO) :



### INPUT

```
import math
a = int(input("Masukkan bilangan pertama (a): "))
b = int(input("Masukkan bilangan kedua (b): "))

jumlah = a + b
selisih = b - a
hasil_kali = a * b
sisipembagian = a % b
pembagian = a / b
log_a = math.log10(a)
pangkat = a ** b
```

### PROSES

```
print("Jumlah a dan b: {jumlah}")
print("Selisih antara b dengan a: {selisih}")
print("Hasil kali a dan b: {hasil_kali}")
print("Sisa pembagian a dengan b: {sisapembagian}")
print("Pembagian a dengan b: {pembagian}")
print("Hasil dari log(a): {log_a}")
print("a pangkat b: {pangkat}")
```

### OUTPUT

```
Masukkan bilangan pertama (a): 12
Masukkan bilangan kedua (b): 3
Jumlah a dan b: {jumlah}
Selisih antara b dengan a: -9
Hasil kali a dan b: 36
Sisa pembagian a dengan b: 0
Pembagian a dengan b: 4.0
Hasil dari log(a): 1.0791812460476249
a pangkat b: 1728
```

Pseudocode :



- Tampilkan input (“Masukkan bilangan pertama (a): “))
- Save input ke variable **a**
- Tampilkan input (“Masukkan bilangan kedua (b): “))
- Save input ke variable **b**
- Masukan penjumlahan antara variable a + variable b
- Masukan selisih antara variable b - variable a
- Masukan hasil kali antara variable a \* variable b
- Masukan sisa\_pembagian antara variable a % variable b
- Masukan pembagian antara variable a / variable b
- Masukan hasil log\_a ke math.log10(a)
- Masukan perpangkatan antara variable a \*\* variable b
- Tampilkan hasil penjumlahan antara variable a + variable b
- Tampilkan hasil Masukan selisih antara variable b - variable a
- Tampilkan hasil hasil kali antara variable a \* variable b
- Tampilkan hasil sisa\_pembagian antara variable a % variable b
- Tampilkan hasil pembagian antara variable a / variable b
- Tampilkan hasil hasil log\_a ke math.log10(a)
- Tampilkan hasil perpangkatan antara variable a \*\* variable b

### Source Code



```
import math
a = int(input("Masukkan bilangan pertama (a): "))
b = int(input("Masukkan bilangan kedua (b): "))

jumlah = a + b
selisih = b - a
hasil_kali = a * b
sisipembagian = a % b
pembagian = a / b
log_a = math.log10(a)
pangkat = a ** b

print(f"Jumlah a dan b: {jumlah}")
print(f"Selisih antara b dengan a: {selisih}")
print(f"Hasil kali a dan b: {hasil_kali}")
print(f"Sisa pembagian a dengan b: {sisipembagian}")
print(f"Pembagian a dengan b: {pembagian}")
print(f"Hasil dari log(a): {log_a}")
print(f"a pangkat b: {pangkat}")
```

### Output





```
➔ Masukkan bilangan pertama (a): 12
Masukkan bilangan kedua (b): 3
Jumlah a dan b: 15
Selisih antara b dengan a: -9
Hasil kali a dan b: 36
Sisa pembagian a dengan b: 0
Pembagian a dengan b: 4.0
Hasil dari log(a): 1.0791812460476249
a pangkat b: 1728
Muhammad Rizki 064102400020
```

b. Latihan Kedua

Buatlah program untuk menghitung jarak antara dua titik di permukaan bumi menggunakan rumus dan fungsi trigonometri pada package math pada python.

Source Code

```
import math

def haversine_distance(lat1, lon1, lat2, lon2):
    """
    Menghitung jarak antara dua titik di permukaan bumi
    menggunakan rumus Haversine.

    Parameters:
    lat1, lon1 : float - Lintang dan bujur titik pertama dalam
    derajat
    lat2, lon2 : float - Lintang dan bujur titik kedua dalam
    derajat

    Returns:
    float - Jarak antara dua titik dalam kilometer
    """
```



```
lat1, lon1, lat2, lon2 = map(math.radians, [lat1, lon1, lat2, lon2])

dlat = lat2 - lat1
dlon = lon2 - lon1

a = math.sin(dlat / 2)**2 + math.cos(lat1) * math.cos(lat2) *
math.sin(dlon / 2)**2
c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))

radius = 6371.0

distance = radius * c
return distance

#Contoh Koordinat dua titik
lat1 = -6.200000    # Koordinat Jakarta
lon1 = 106.816666
lat2 = 51.507222    # Koordinat London
lon2 = -0.1275

jarak = haversine_distance(lat1, lon1, lat2, lon2)
print(f"Jarak antara dua titik adalah {jarak:.2f} kilometer.")
```

### Pseudocode

- Tampilkan input ("Masukkan lintang titik pertama (lat1): ")
- Save input ke variabel lat1
- Tampilkan input ("Masukkan bujur titik pertama (lon1): ")
- Save input ke variabel lon1
- Tampilkan input ("Masukkan lintang titik kedua (lat2): ")
- Save input ke variabel lat2
- Tampilkan input ("Masukkan bujur titik kedua (lon2): ")
- Save input ke variabel lon2



- Konversi lat1, lon1, lat2, lon2 dari derajat ke radian menggunakan `math.radians()`
- Hitung selisih lintang `dlat = lat2 - lat1`
- Hitung selisih bujur `dlon = lon2 - lon1`
- Hitung `a = math.sin(dlat / 2)**2 + math.cos(lat1) * math.cos(lat2) * math.sin(dlon / 2)**2`
- Hitung `c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))`
- Tetapkan jari-jari bumi `radius = 6371.0` (dalam kilometer)
- Hitung `jarak = radius * c`
- Tampilkan hasil jarak antara dua titik dalam kilometer

### Output

```
lon2 = -0.1275

jarak = haversine_distance(lat1, lon1, lat2, lon2)
print(f"Jarak antara dua titik adalah {jarak:.2f} kilometer.")
```

Jarak antara dua titik adalah 11715.38 kilometer.

## 4. File Praktikum

Github Repository:

<https://github.com/muhrizki26/praktek-algoritma-modul-2-Muhammad-Rizki-064102400020.git>

## 5. Soal Latihan

Soal:

1. Dalam operasi aritmatika, presedensi yang tertinggi adalah ekspresi dalam tanda kurung. Jelaskan serta deskripsikan apa yang dimaksud dari presedensi yang tertinggi tersebut?
2. Deskripsikan serta narasikan jalannya alur source code program yang sebelumnya telah kalian buat pada Elemen Kompetensi Latihan Kedua!

Jawaban:

1. Dalam operasi aritmatika, preseden tertinggi mengacu pada urutan atau prioritas pelaksanaan operasi matematis di mana ekspresi dalam tanda kurung dieksekusi terlebih dahulu. Ini memastikan bahwa perhitungan dalam tanda kurung diselesaikan sebelum operasi seperti penjumlahan, pengurangan, perkalian, atau pembagian. Oleh karena itu, hasil akhir dari ekspresi matematis dapat dipengaruhi oleh penggunaan tanda kurung.



2. deskripsi dan alur program yang menggunakan rumus Haversine untuk menghitung jarak antara dua titik di permukaan bumi:

1. Impor Modul: Program dimulai dengan mengimpor modul "math", yang menyediakan berbagai fungsi matematika yang diperlukan untuk perhitungan, seperti fungsi trigonometri dan konversi radian.

2. Definisi Fungsi: Fungsi "haversine\_distance" didefinisikan oleh empat parameter: "lat1", "lon1", "lat2", dan "lon2", yang masing-masing menunjukkan lintang dan bujur dua titik derajat. Selain itu, terdapat docstring di dalam fungsi yang menjelaskan tujuan fungsi, parameter yang diterima, dan nilai yang dikembalikan.

3. Konversi Derajat ke Radian: - Menggunakan "math.radians()", dapat mengubah koordinat lintang dan bujur dari derajat ke radian. Ini penting karena fungsi trigonometri dalam Python bekerja dengan radian.

4. Metode untuk Menghitung Selisih Koordinat:

Hubungan antara kedua titik dihitung dengan menghitung jarak lintang (dlat) dan bujur (dlon). Rumus Haversine menggunakan perbedaan ini:

5. Rumus Haversine dapat digunakan untuk menghitung nilai a sebagai berikut: menggunakan  $\text{math.sin}(dlat/2)^2$  untuk menghitung kuadrat sinus setengah selisih lintang, mengalikan  $\text{math.cos}(lat1)$  dan  $\text{math.cos}(lat2)$  untuk mendapatkan komponen kosinus lintang, yang kemudian dikalikan dengan kuadrat sinus setengah selisih bujur. Selanjutnya, nilai c dihitung dengan menggunakan  $\text{math.atan2}()$  untuk menghitung jarak sudut antara kedua titik.

6. Metode untuk Menghitung Jarak:

Dengan jari-jari bumi 6371 kilometer, jarak antara dua titik dapat dihitung dengan

7. mengalikan jari-jari bumi dengan nilai c. Nilai pengembalian:

Nilai jarak, yaitu jarak dalam kilometer, dikembalikan oleh fungsi haversine\_distance.

## 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 suatu bentuk source code baru yang dimana kita mendata suatu rumus matematika dan menjalankan atau menampilkan ke dalam programnya = dengan menggunakan spyder maupun googlecollab. Modul 2 ini banyak mengajarkan untuk lebih teliti pada masing-masing code yang akan dijalankan seperti contohnya f'' ini agar bias terbaca variabelnya dan juga program akan berjalan.



## 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	20 Menit	menarik
2.	Latihan Kedua	25 Menit	Cukup menarik

Keterangan:

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