

<p>Nama: <b>Steven Raja P M</b></p> <p>NIM: <b>064102400018</b></p>	 <p><b>Praktikum Algoritma &amp; Pemrograman</b></p>	<h1><b>MODUL 8</b></h1> <p>Nama Dosen: <b>Binti solihah, S.T, M.KOM</b></p> <p>Nama Asisten Labratorium:</p> <ul style="list-style-type: none"><li>1. <b>Yustianas Rombon - 064002300015</b></li><li>2. <b>Vira Aditya Kurniawan - 065002300012</b></li></ul>
<p>Hari/Tanggal: <b>Hari, Tanggal Bulan 2024</b></p>		

## **Fungsi Rekursif (Recursive Function)**

### **1. Teori Singkat**

#### **Fungsi Rekursif**

Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri secara berulang. Jadi di dalam tubuh fungsi yang dideklarasikan kita memanggil fungsi itu sendiri. Rekursif ini sebenarnya merupakan sebuah perulangan di dalam sebuah program. Namun, perulangan rekursif ini sangat berbeda dengan perulangan pada umumnya, seperti while dan for. Walaupun fungsinya sama yaitu untuk melakukan perulangan atau looping. Letak perbedaannya adalah dari cara kerjanya. Jika ‘for’ dan ‘while’ merupakan sebuah perulangan yang menggunakan sebuah kondisi atau Boolean (true/false), maka pada rekursif ini terjadi pada sebuah fungsi atau metode yang memanggil dirinya sendiri. Dari penjelasan tersebut dapat kita katakan sebagai perulangan yang memanggil dirinya sendiri untuk melakukan sebuah perulangan.

Fungsi rekursif dapat menyelesaikan beberapa persoalan seperti perhitungan bilangan fibbonaci dan faktorial.

$$\text{faktorial}(5) = 5 * \text{faktorial}(4)$$

$$\text{faktorial}(4) = 4 * \text{faktorial}(3)$$

$$\text{faktorial}(3) = 3 * \text{faktorial}(2)$$

$$\text{faktorial}(2) = 2 * \text{faktorial}(1)$$

$$\text{faktorial}(1) = 1$$



Maka faktorial(5) =  $5 * 4 * 3 * 2 * 1$ , akan menghasilkan 120

#### Source Code

```
#FUnksi Rekursif
def rekursif(angka):
    if angka > 0:
        print (angka)
        angka = angka - 1
        rekursif(angka)
    else:
        print(angka)

masukkan = int(input("masukkan angka: "))
rekursif(masukkan)
```

#### Output





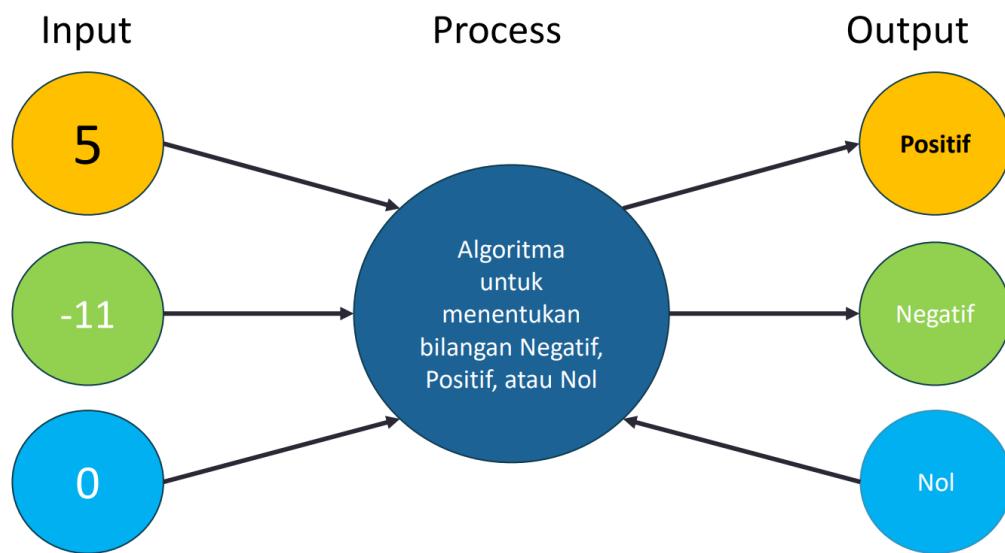
## IPO (Input Process Output)

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)



### Notasi Algoritma Flowchart

1. Flowchart adalah representasi visual atau diagram alir yang digunakan untuk menggambarkan langkah-langkah dan urutan proses suatu algoritma atau program.
2. Flowchart menyajikan langkah-langkah dalam bentuk simbol-simbol grafis yang saling terhubung, membantu dalam memvisualisasikan bagaimana informasi mengalir dan bagaimana proses dilakukan.
3. Dalam kaitannya dengan notasi deskriptif, notasi algoritma yang menggunakan flowchart dapat lebih cepat dibaca dan dilihat alur dan hubungannya.

### Simbol-simbol pada Flowchart

1. Setiap elemen flowchart dihubungkan oleh garis aliran bertanda panah
2. Garis aliran dimulai dari atas symbol dan keluar dari bagian bawah, kecuali symbol keputusan yang alirannya keluar dari bawah atau samping
3. Aliran bergerak dari atas ke bawah
4. Proses awal dan akhir menggunakan symbol terminal.



## ... Simbol-simbol pada Flowchart



**Terminator** yang menandakan *Start* (awal) atau *End* (akhir) program.



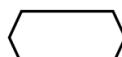
**Flow line** yang digunakan untuk menunjukkan arah aliran pada program.



**Process** menunjukkan proses yang dilakukan pada masukan.



**Input atau output** untuk menunjukkan masukan dan keluaran.



**Preparation** digunakan untuk membuat deklarasi nilai awal.



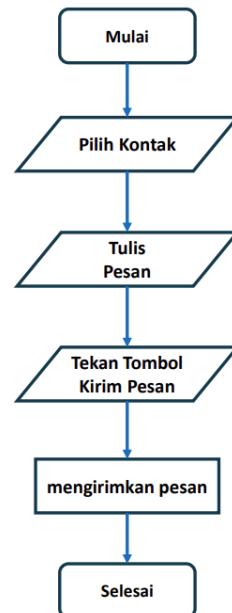
**On Page Connector** digunakan untuk menghubungkan antar *flowchart*.



**Decision** menunjukkan keputusan atau kondisi untuk memilih keputusan.

**Contoh sederhana  
Penggunaan *flowchart*  
untuk menunjukkan algoritma**

**Kasus/Aliran:  
Mengirim pesan WhatsApp**



## 2. Alat dan Bahan

Hardware : Laptop/PC

Software : Spyder (Anaconda Python)

## 3. Elemen Kompetensi

a. Latihan pertama



Buatlah sebuah fungsi penjumlahan berurut menggunakan konsep rekursif (tanpa for/while). User perlu memasukkan angka awal dan bilangan-bilangan yang ingin ditambahkan seperti pada berikut:

Masukkan Jumlah: 3 (inputan user)

Masukkan angka ke-1: 1 (inputan user)

Masukkan angka ke-2: 2 (inputan user)

Masukkan angka ke-3: 3 (inputan user)

IPO (Input Output Process)

```
▶ def rekursif_penjumlahan(n, hasil=0):
    if n == 0:
        return hasil

    angka = int(input(f"Masukkan angka ke-{4 - n}: "))

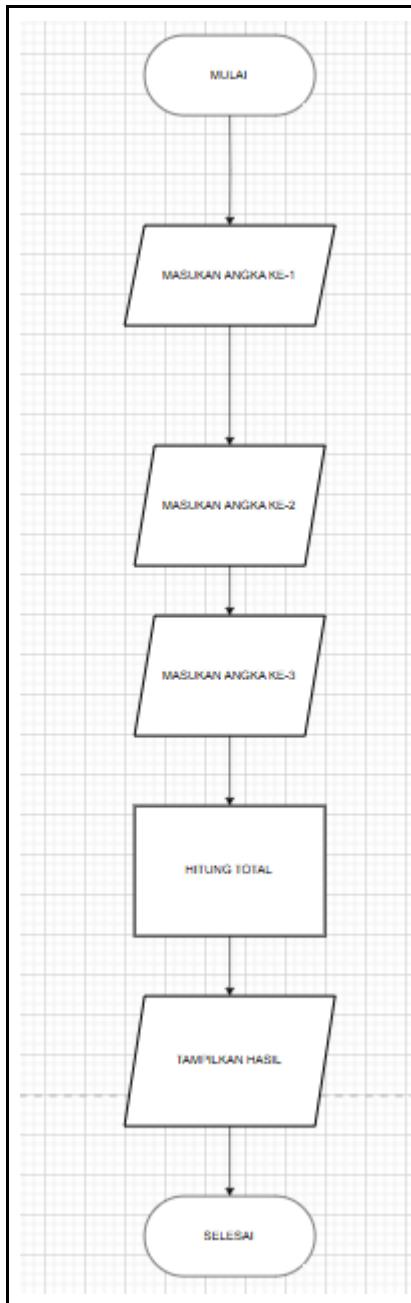
    return rekursif_penjumlahan(n - 1, hasil + angka)

hasil_akhir = rekursif_penjumlahan(3)
print(f"Hasil penjumlahan: {hasil_akhir}")
```

```
→ Masukkan angka ke-1: 11
    Masukkan angka ke-2: 11
    Masukkan angka ke-3: 11
    Hasil penjumlahan: 33
```

Flowchart





### Source Code



```
def rekursif_penjumlahan(n, hasil=0):
    if n == 0:
        return hasil

    angka = int(input(f"Masukkan angka ke-{4 - n}: "))

    return rekursif_penjumlahan(n - 1, hasil + angka)

hasil_akhir = rekursif_penjumlahan(3)
print(f"Hasil penjumlahan: {hasil_akhir}")
```

### Output

→ Masukkan angka ke-1: 11  
Masukkan angka ke-2: 11  
Masukkan angka ke-3: 11  
Hasil penjumlahan: 33

b. Latihan Kedua

Buatlah sebuah fungsi perpangkatan menggunakan konsep rekursif (tanpa for/while). User hanya perlu memasukkan base number (angka awal) dan power (pangkatnya). Hasil akhir berupa perhitungan perpangkatannya.

IPO (Input Output Process)



```
s  ⏎ def perpangkatan(base, power):
...     if power == 0:
...         return 1
...     return base * perpangkatan(base, power - 1)

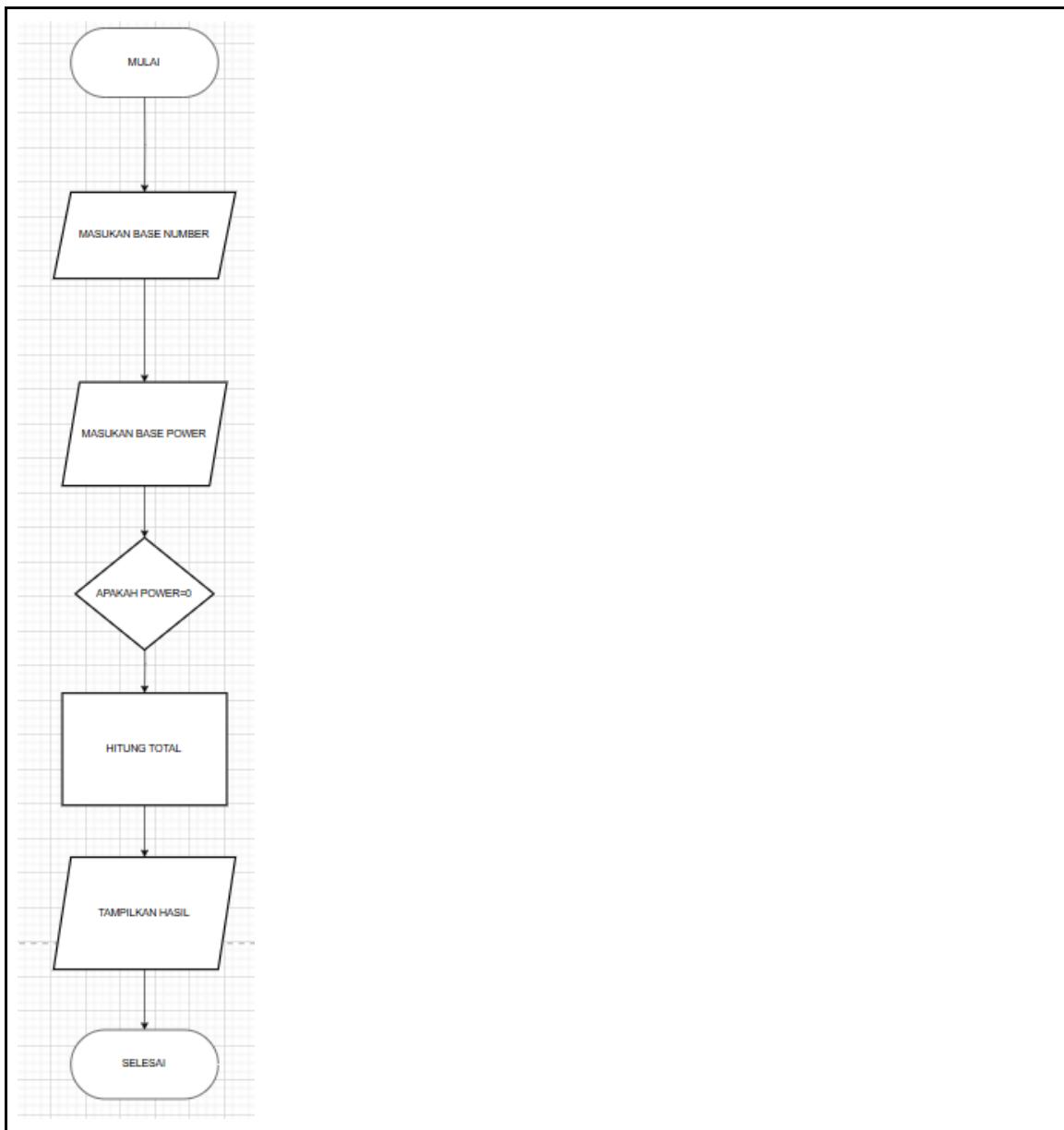
base = int(input("Masukkan base number: "))
power = int(input("Masukkan power: "))

hasil = perpangkatan(base, power)
print(f"Hasil dari {base}^{power} adalah {hasil}")

→ Masukkan base number: 15
Masukkan power: 10
Hasil dari 15^10 adalah 576650390625
```

Flowchart





Source Code



```
def perpangkatan(base, power):
    if power == 0:
        return 1
    return base * perpangkatan(base, power - 1)

base = int(input("Masukkan base number: "))
power = int(input("Masukkan power: "))

hasil = perpangkatan(base, power)
print(f"Hasil dari {base}^{power} adalah {hasil}")
```

### Output

```
→ Masukkan base number: 15
Masukkan power: 10
Hasil dari 15^10 adalah 576650390625
```

### c. Latihan Ketiga (Tidak Wajib)

Buatlah 2 buah fungsi konversi ‘hexadecimal ke decimal’ yang mengkonversi string berbentuk hexadecimal ke integer dan ‘decimal ke hexadecimal’ yang mengkonversi integer berbentuk decimal kedalam bentuk hexadecimal. Setiap fungsi akan mengambil 1



nilai untuk dikonversi (sebagai satu-satunya parameter yang dimilikinya) dan me-return nilai yang sudah terkonversi sebagai satu-satunya hasil akhir. Berikan error message jika inputan yang diberikan invalid. Gunakan konsep rekursi untuk mendapat nilai maksimal di soal bonus ini. (lihat Exercise 98)

#### Source Code

```
print("Isi dengan Source Code kalian ya...")
```

#### Output

```
print("Screenshot hasil keluaran atau output dari source code kalian ya...")
```

#### **4. File Praktikum**



Github Repository:

<https://github.com/Stevenraja/PrakAlgo8.git>

## 5. Soal Latihan

Soal:

1. Sebutkan dan jelaskan perbedaan secara teknis antara perulangan for/while dengan perulangan fungsi rekursif?
2. Deskripsikan serta narasikan jalannya alur source code program yang sebelumnya telah kalian buat pada Elemen Kompetensi Latihan Kedua!

Jawaban:

1. for/while= mengulang blok kode menggunakan kondisi yang jelas  
rekursif= memanggil fungsi sendiri dengan parameter yang diperbarui setiap langkah
2. input data untuk memasukan angka base(angka yang akan dipangkatkan) lalu masukan power(mengkalikan seberapa banyak base), jika power = 0 maka program akan berhenti dan memberikan hasil =1(pangkat 0 =1),lalu hasil dari base^power adalah hasil akhirnya

## 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 bagaimana menjalankan fungsi rekursif untuk menghitung penjumlahan dan perpangkatan dan apa perbedaanya disaat kita menggunakan for/while

## 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	45 Menit	Menarik



<b>2.</b>	Latihan Kedua	40Menit	Menarik
-----------	---------------	---------	---------

Keterangan:

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

