

Pengertian

1. Fungsi pada python adalah kumpulan perintah atau baris kode yang dikelompokkan menjadi satu kesatuan untuk kemudian bisa dipanggil atau digunakan berkali-kali.
2. Sebuah fungsi dapat menerima parameter, dapat mengembalikan suatu nilai, dan dapat dipanggil berkali-kali secara independen.

Keuntungan

1. Membagi kode program menjadi bagian-bagian kecil dengan tugasnya masing-masing.
2. Membuat kode program menjadi lebih "reusable" dan lebih terstruktur.

Sintaks

```
def function_name():  
    Logic Program
```

Cara memanggil fungsi dengan menuliskan function_name diikuti kurung buka tutup () beserta parameter jika ada

```
function_name()
```

Buatlah fungsi dengan nama "helloPython" yang berfungsi untuk mencetak "Welcome in Python Language"

```
In [18]: def function_hellopython():  
         print("Welcome in Python Language")
```

```
In [19]: function_hellopython()
```

Welcome in Python Language

Fungsi dengan parameter

Sebuah fungsi dapat menerima parameter ataupun argumen yang merupakan suatu nilai/variabel yang dilemparkan ke dalam fungsi untuk diproses lebih lanjut.

Sintaks

```
def function_name(param):  
    Logic Program
```

Cara memanggil fungsi dengan menuliskan function_name diikuti kurung buka tutup () beserta parameter

```
function_name(param)
```

Buatlah fungsi dengan nama "fullName" yang mempunyai parameter "firstname" dan "lastname" digunakan untuk mencetak "firstname" dan "lastname"

```
In [35]: def function_nama(firstname, lastname):  
         print(firstname, lastname)
```

```
In [36]: function_nama("tegar", "Maulana")
```

tegar Maulana

Parameter fungsi diperbolehkan menggunakan lebih dari 1, dimana parameter tersebut ada yang wajib harus diisi dan ada yang tidak harus diisi

```
def function_name(param_1, param_2, param_3,...):  
    Logic Program
```

Cara memanggil fungsi dengan menuliskan function_name diikuti kurung buka tutup () beserta parameter

```
function_name(param_1, param_2, param_3)
```

Buatlah fungsi dengan nama maxValue yang mempunyai parameter "val_1","val_2" dan "val_3" yang bertujuan untuk mencari nilai terbesar dari 3 nilai tersebut

```
In [32]: def function_nama(val_1, val_2, val_3):  
         maxvalue = max(val_1, val_2, val_3)  
         return maxvalue  
result = max(4, 9, 22)  
print(result)
```

22

Parameter opsional digunakan pada fungsi dengan cara memberikan nilai default, artinya nilai parameter sudah diberikan terlebih dahulu tanpa dipanggil

```
def function_name(param_1, param_2, param_3 = 'Nilai'):  
    Logic Program
```

Buatlah fungsi "countCircleArea" dengan 2 parameter yaitu "phi" dan "diameter" dimana parameter phi mempunyai nilai default 3.14

```
In [52]: def countCircleArea(diameter, phi=3.14):  
        angka = diameter / 2  
        angka2 = phi * (angka ** 2)  
        return angka2  
result = countCircleArea(23, 3.14)  
print(result)
```

415.26500000000004

Fungsi dengan 2 Parameter opsional juga dapat dilakukan pada Python.

```
def info(suhu, daerah='Sukabumi', satuan = 'Celcius'):  
    print(f"Suhu sekarang di {daerah} : {suhu} {satuan} ")
```

Adapun Cara memanggilnya

```
info(30)
```

```
In [54]: #Cobalah fungsi tersebut pada area kode disini  
def info(suhu, daerah='Sukabumi', satuan = 'Celcius'):  
    print(f"Suhu sekarang di {daerah} : {suhu} {satuan} ")  
info(30)
```

Suhu sekarang di Sukabumi : 30 Celcius

Fungsi dengan Return Value

Yaitu fungsi dimana akhir dari programnya adalah nilai kembalian atau nilai balik. Artinya nilai dalam fungsi dapat ditampung lagi ke variabel lain untuk digunakan operasi lebih lanjut.

Buatlah fungsi dengan return value yang digunakan untuk mengecek sebuah bilangan termasuk bilangan negatif, bilangan netral(0) dan bilangan positif

```
In [58]: def cek_bilangan(bilangan):  
    if bilangan < 0:  
        return "Negatif"  
    elif bilangan == 0:  
        return "Netral"  
    else:  
        return "Positif"  
bil1 = -33  
bil2 = 0  
bil3 = 4423  
hasil1 = cek_bilangan(bil1)  
hasil2 = cek_bilangan(bil2)  
hasil3 = cek_bilangan(bil3)  
print(f"Bilangan {bil1} adalah bilangan {hasil1}.")  
print(f"Bilangan {bil2} adalah bilangan {hasil2}.")  
print(f"Bilangan {bil3} adalah bilangan {hasil3}.")
```

Bilangan -33 adalah bilangan Negatif.
Bilangan 0 adalah bilangan Netral.
Bilangan 4423 adalah bilangan Positif.

Soal Latihan

1. Buatlah fungsi untuk menjumlahkan total nilai dari list
2. Buatlah fungsi untuk mencari nilai terbesar dari sekumpulan list
3. Buatlah fungsi untuk menjumlahkan 2 buah list

```
In [60]: #Soal 1  
def jumlah_semuanya(nilai_list):  
    total = 0  
    for nilai in nilai_list:  
        total += nilai  
    return total  
nilai_list = [31, 440, 50]  
total_nilai = jumlah_semuanya(nilai_list)  
print(f"jumlah: {total_nilai}")
```

jumlah: 521

```
In [62]: #Soal 2  
def nilai(nilai_list):  
    for nilai in nilai_list:  
        total = max(nilai_list)  
    return total
```

```
In [63]: nilai_list = [33, 50, 220]  
hasil = max(nilai_list)  
print(f"nilai terbesar: {hasil}")
```

nilai terbesar: 220

```
In [76]: #Soal 3
def jumlahkan_list(list1, list2):
    panjang_min = min(len(list1), len(list2))
    hasil = []

    for i in range(panjang_min):
        hasil.append(list1[i] + list2[i])
    return hasil
list1 = [2, 5, 7, 9, 2]
list2 = [100, 2000, 30000, 400000, 500000]
hasil_penjumlahan = jumlahkan_list(list1, list2)

print(f"Hasil dari list1 dan list2 adalah: {hasil_penjumlahan}")
```

Hasil dari list1 dan list2 adalah: [102, 2005, 30007, 400009, 500002]

In [67]:

```
-----
NameError                                Traceback (most recent call last)
Cell In[67], line 1
----> 1 hasil_penjumlahan = jumlahkan_list(list1, list2)
      3 print(f"Hasil dari list1 dan list2 adalah: {hasil_penjumlahan}")

NameError: name 'list1' is not defined
```

In []: