

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"

```
def helloPython():  
    """  
    Fungsi untuk mencetak pesan "Welcome in Python Language"  
  
    Returns:  
        None  
    """  
    print("Welcome in Python Language")  
  
# Memanggil fungsi  
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"

```
def fullName(firstname, lastname):  
    """  
    Prints the full name constructed from the provided first and last name.  
  
    Args:  
        firstname: The person's first name (string).  
        lastname: The person's last name (string).  
    """  
    print(f"{firstname} {lastname}")  
  
# Example usage  
fullName("Dadang", "Gumilang") # Output: Dadang Gumilang
```

 Dadang Gumilang

Parameter fungsi diperbolehkan menggunakan lebih dari 1, dimana paramater 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 maxValu yang mempunyai parameter "val_1","val_2" dan "val_3" yang bertujuan untuk mencari nilai terbesar dari 3 nilai tersebut

```
def maxValu(val_1, val_2, val_3=None):  
    """  
    Mencari nilai maksimum dari 3 nilai yang diberikan.  
  
    Args:  
        val_1: Nilai pertama (wajib).  
        val_2: Nilai kedua (wajib).  
        val_3: Nilai ketiga (opsional, default None). Jika tidak disediakan,  
              fungsi hanya akan membandingkan val_1 dan val_2.  
  
    Returns:  
        Nilai maksimum dari 3 nilai.  
    """  
    if val_3 is None:  
        # Hanya bandingkan val_1 dan val_2  
        result = max(val_1, val_2)  
    else:  
        # Bandingkan val_1, val_2, dan val_3  
        result = max(val_1, val_2, val_3)  
    return result  
  
# Contoh penggunaan  
max_value = maxValu(10, 20 , 40) # Output: 20  
  
print(max_value)
```

 40

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

```
def countCircleArea(diameter, phi=3.14):  
    """  
    Menghitung luas lingkaran.  
  
    Args:  
        diameter: Diameter lingkaran (wajib).  
        phi (default 3.14): Nilai phi (π).  
  
    Returns:  
        Luas lingkaran.  
    """  
    radius = diameter / 2  
    area = phi * radius * radius  
    return area  
  
# Contoh penggunaan  
luas_lingkaran1 = countCircleArea(10) # Output: 78.53979999999999  
luas_lingkaran2 = countCircleArea(15, 3.15) # Output: 176.69399999999998  
print(luas_lingkaran2)
```

 177.1875

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)
```

```
#Cobalah fungsi tersebut pada area kode disini
def info(suhu, daerah='Sukabumi', satuan='Celcius'):
    """
    Menampilkan informasi suhu beserta daerah dan satuannya.

    Args:
        suhu: Nilai suhu (wajib).
        daerah: Nama daerah (opsional, default 'Sukabumi').
        satuan: Satuan suhu (opsional, default 'Celcius').

    Returns:
        None
    """
    print(f"Suhu sekarang di {daerah} : {suhu} {satuan}")
```

```
# Contoh penggunaan
info(30) # Output: Suhu sekarang di Sukabumi : 30 Celcius
info(25, 'Bandung') # Output: Suhu sekarang di Bandung : 25 Celcius
info(37.5, satuan='Fahrenheit') # Output: Suhu sekarang di Sukabumi : 37.5 Fahrenheit
```

```
Suhu sekarang di Sukabumi : 30 Celcius
Suhu sekarang di Bandung : 25 Celcius
Suhu sekarang di Sukabumi : 37.5 Fahrenheit
```

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

```
def checkNumberType(number):
    """
    Mengecek jenis bilangan (negatif, netral, atau positif).

    Args:
        number: Bilangan yang akan dicek (wajib).

    Returns:
        String yang menunjukkan jenis bilangan:
        - "Negatif" jika bilangannya negatif
        - "Netral" jika bilangannya 0
        - "Positif" jika bilangannya positif
    """
    if number < 0:
        return "Negatif"
    elif number == 0:
        return "Netral"
    else:
        return "Positif"
```

```
# Contoh penggunaan
bilangan = 10
jenis_bilangan = checkNumberType(bilangan)
print(f"Bilangan {bilangan} termasuk: {jenis_bilangan}")
```

```
Bilangan 10 termasuk: 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

```
#Soal 1
def sum_list(numbers):
    """
    Menghitung total nilai dari elemen-elemen dalam list.

    Args:
        numbers: List yang berisi nilai-nilai (bisa berupa angka, string, dll.).

    Returns:
        Total nilai dari elemen-elemen dalam list.
    """
    total = 0
    for number in numbers:
        total += number
    return total

# Contoh penggunaan
numbers = [10, 5, 2, 7, 1]
total_sum = sum_list(numbers)
print(f"Total nilai dari list: {total_sum}")
```


 Total nilai dari list: 25

```
#Soal 2
def sum_list(numbers):
    """
    Menghitung total nilai dari elemen-elemen dalam list.

    Args:
        numbers: List yang berisi nilai-nilai (bisa berupa angka, string, dll.).

    Returns:
        Total nilai dari elemen-elemen dalam list.
    """
    total = 0
    for number in numbers:
        total += number
    return total

# Contoh penggunaan
numbers = [10, 5, 2, 7, 1]
total_sum = sum_list(numbers)
print(f"Total nilai dari list: {total_sum}")
```


 Total nilai dari list: 25

```
#Soal 3
def sum_lists(list1, list2):
    """
    Menjumlahkan elemen-elemen dari 2 list dengan indeks yang sama dan mengembalikan hasilnya dalam list baru.

    Args:
        list1: List pertama yang berisi nilai-nilai.
        list2: List kedua yang berisi nilai-nilai.

    Returns:
        List baru yang berisi hasil penjumlahan elemen-elemen dari list1 dan list2 dengan indeks yang sama.
    """
    if len(list1) != len(list2):
        raise ValueError("List harus memiliki panjang yang sama") # Menangani panjang list yang berbeda
    result_list = []
    for i in range(len(list1)):
        result_list.append(list1[i] + list2[i])
    return result_list

# Contoh penggunaan
list1 = [10, 5, 2]
list2 = [7, 1, 4]
summed_list = sum_lists(list1, list2)
print(f"Hasil penjumlahan list: {summed_list}")
```

 Hasil penjumlahan list: [17, 6, 6]

