**UNIT 10**

**Fungsi**

**Tujuan:**

1. Mampu Memahami Penggunaan Fungsi pada Python

2. Mampu Memahami Fungsi Matematis menggunakan Python

3. Mampu Menganalisis Persamaan yang dikerjakan

**Dasar Teori:**

Dalam menyelesaikan masalah yang kompleks menggunakan perangkat lunak, kasus utama harus dipecah-pecah menjadi kasus yang lebih kecil. Kemudian kita berkonsentrasi untuk mencari pemecahan yang terbaik dari masing-masing bagian kecil ini. Masing- masing bagian diselesaikan dengan menggunakan algoritma sebaik mungkin. Bagian- bagian kecil ini pada akhirnya bisa dapat digabungkan untuk memberikan jawaban yang optimal terhadap masalah yang ada.

Fungsi dipakai untuk mengumpulkan beberapa perintah yang sering dipakai dalam sebuah program. Dengan memakai fungsi, program yang dibuat menjadi lebih terstruktur. Lebih mudah diikuti oleh orang lain yang membaca program dibuat. Paling penting adalah mempersingkat waktu yang diperlukan untuk mengembangkan suatu perangkat lunak. Karena perangkat lunak yang dibuat, bisa jadi memakai komponen-komponen yang sama. Seperti layaknya sebuah bahasa pemrograman, Python juga memberikan fasilitas pembuatan fungsi yang sangat bagus. Konsep fungsi dalam Python sama dengan bahasa pemrograman C/C++. Python menganggap fungsi dan prosedur adalah sesuatu yang sama, dalam artian cara mendeklarasikan fungsi dan prosedur adalah sama. Hanya bedanya, kalau fungsi mengembalikan suatu nilai setelah proses sedangkan prosedur tidak.

1. Sebuah fungsi diawali dengan statemen **def** kemudian diikuti oleh sebuah nama\_fungsi nya. Pernyataan def dipakai untuk mendeklarasikan fungsi.
2. Sebuah fungsi dapat memiliki daftar argumen (parameter) ataupun tidak.
3. Tanda titik dua ( : ) menandakan awal pendefinisian tubuh dari fungsi yang terdiri dari statemen-statemen
4. Statemen return menandakan akhir dari pemanggilan fungsi dan akan mengirimkan suatu nilai balik kepada program yang memanggil fungsi tersebut. Pernyataan return dipakai untuk mengembalikan suatu nilai kepada bagian program yang memanggil fungsi.

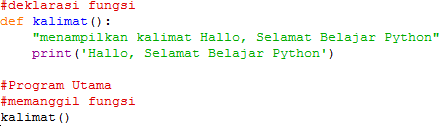
***Pembuatan dan pemanggilan fungsi***

**namaFungsi (parameter)**

Dalam deklarasi fungsi, juga bisa menambahkan komentar-komentar yang memberi penjelasan mengenai fungsi yang dibuat. Secara umum memang bisa menambahkan komentar-komentar di sembarang tempat dalam program yang dibuat. Baris-baris komentar diawali dengan karakter pagar (#). Semua karakter yang mengikuti tanda ini sampai akhir baris dianggap sebagai komentar dan tidak akan mempengaruhi jalannya progam. Akan tetapi terdapat satu gaya pemberian komentar dalam Python yang disebut dengan docstring. Biasanya dipakai untuk memberi penjelasan mengenai fungsi atau objek. *Docstring* diapit dengan tanda petik ganda, komentar jenis ini hanya boleh diberikan tepat satu baris dibawah deklarasi fungsi atau objek yang akan ditunjukkan pada pembahasan selanjutnya. Docstring sangat bermanfaat ketika kita ingin mendokumentasikan semua fungsi dan kelas yang telah kita buat. Karena ada beberapa perangkat lunak yang mampu membuat dokumentasi berdasarkan *docstring* yang ada dalam *source code.*

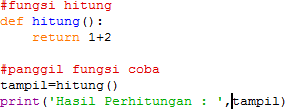
***Fungsi tanpa parameter dan tanpa return***

Perintah dibawah dapat dimaknai sebagai berikut, Pernyataan def mendefinisikan sebuah fungsi dengan nama kalimat. Tidak ada paramater yang akan dilewatkan ke dalam fungsi sehingga di dalam tanda kurung tidak ada yang perlu dituliskan Baris deklarasi fungsi ini diakhiri dengan titik dua (:). Tanda ini memberitahukan pada interpreter Python bahwa baris ini masih berlanjut pada baris-baris berikutnya. Dalam deklarasi diatas terdapat penggunaan komentar yang ditandai dengan tanda pagar (#) yaitu tulisan #deklarasi fungsi dan tulisan #Program Utama serta tulisan #memanggil fungsi dengan maksud untuk memberi keterangan ataupun memperjelas maksud dari kode-kode yang digunakan. Pada perintah diatas juga terdapat docstring yaitu tulisan "menampilkan kalimat Hallo, Selamat Belajar Python" yang diapit dengan tanda petik ganda. Digunakan untuk memberi penjelasan mengenai fungsi dengan nama kalimat diatas. Fungsi diatas tidak memiliki nilai kembalian dengan demikian tidak perlu menggunakan pernyataan return.

****

Setelah perintah di atas dijalankan (run) maka akan tampil seperti berikut ini :

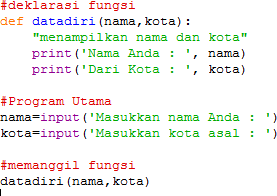


***Fungsi tanpa parameter dengan return***

Pada perintah diatas setelah proses penjumlahan maka hasilnya akan dikembalikan ke fungsi yang memanggil dengan menggunakan pernyataan return, sehubungan ada nilai yang dikembalikan maka diperlukan variabel penampung yang dalam hal ini menggunakan variabel tampil baru selanjutnya hasilnya ditampilkan.

Setelah perintah diatas dijalankan (*run*) maka akan tampil seperti berikut ini :



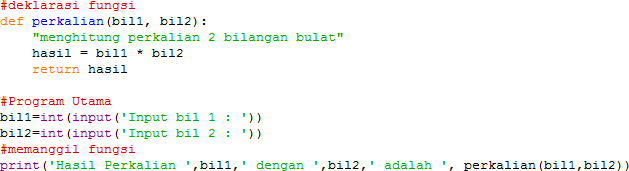
***Fungsi dengan parameter tanpe return***

Pada perintah diatas paramater-parameter yang akan dilewatkan ke dalam fungsi didaftarkan dalam tanda kurung yaitu parameter nama dan kota. Masing-masing paramater dipisahkan oleh koma (,) . Setelah dilewatkan selanjutnya di dalam fungsi akan ditampilkan berdasarkan nilai yang dikirim.

Setelah perintah diatas dijalankan (*run*) maka akan tampil seperti berikut ini :



***Fungsi dengan parameter dengan return***



Pernyataan def mendefinisikan sebuah fungsi dengan nama perkalian. Paramater- parameter yang akan dilewatkan ke dalam fungsi didaftarkan dalam tanda kurung yaitu bil1 dan bil2. Masing-masing paramater dipisahkan oleh koma (,). Baris deklarasi fungsi ini diakhiri dengan titik dua (:). Tanda ini memberitahukan pada interpreter Python bahwa baris ini masih berlanjut pada baris-baris berikutnya.

Perhatikan dua baris pernyataan terindentasi yang mengikutinya. Dalam Python semua pernyataan yang diindentasi dalam satu tingkatan indentasi adalah pernyataan- pernyataan yang satu derajat. Artinya semua pernyataan tersebut akan dieksekusi sesuai dengan urutan penulisannya. Untuk memanggil fungsi yang telah dibuat adalah dengan cara menyebutkan nama fungsi yang bersangkutan beserta daftar parameter yang sebenarnya. Dalam deklarasi fungsi, Anda juga bisa menambahkan komentar-komentar yang memberi penjelasan mengenai fungsi yang dibuat. Secara umum kita memang bisa menambahkan komentar-komentar di sembarang tempat dalam program yang kita buat. Baris-baris komentar diawali dengan karakter pagar (#). Semua karakter yang mengikuti tanda ini sampai akhir baris dianggap sebagai komentar dan tidak akan mempengaruhi jalannya progam.

Setelah perintah diatas dijalankan (run) maka akan tampil seperti berikut ini :



***Variabel Lokal dan Global***

1. Variabel disebut local ketika variabel tersebut didefinisikan didalam sebuah fungsi (def). Artinya, variabel tersebut hanya dapat di gunakan dalam cakupan fungsi tersebut saja.
2. Variabel disebut global jika sebuah variabel didefinisikan diluar fungsi. Artinya, variabel tersebut dapat digunakan oleh fungsi lain atau pun program utamanya

**Contoh Program:**

***Contoh 1: Program menghitung banyak abjad pada teks***

def menghitung\_banyak\_huruf(teks):

hitung\_huruf = 0

# Mengurutkan setiap huruf pada teks

for huruf in teks:

# Jika spasi tidak dihitung

if huruf != " ":

# Menghitung setiap huruf

hitung\_huruf = hitung\_huruf + 1

# Mengubah angka menjadi string untuk dapat di print

return str(hitung\_huruf)

input\_teks = str(input("Masukkan teks : "))

print(f"Ada {menghitung\_banyak\_huruf(input\_teks)} Huruf di dalam teks")

OUTUT:

>>

Masukkan teks : Saya adalah mahasiswa Teknik Elektro

Ada 32 Huruf di dalam teks

***Contoh 2: Program untuk mengklasifikasi apakah bilangan genap atau ganjil***

def genap\_atau\_ganjil(angka):

if angka % 2 == 0:

return "Angka adalah genap"

else:

return "Angka adalah ganjil"

print(genap\_atau\_ganjil(7))

OUTPUT:

>> Angka adalah ganjil

***Contoh 3: Progam untuk mendeteksi nilai terkecil pada list.***

def cari\_terkecil(l):

terkecil = l[0]

for idx in range(1,len(l)):

nilai = l[idx]

if nilai < terkecil:

terkecil = nilai

return terkecil

nilai\_uts = [90,85,70,55,60,67,86,99,100,72,73]

nilai\_terkecil = cari\_terkecil(nilai\_uts)

print('Nilai terkecil pada UTS adalah : {}'.format(nilai\_terkecil))

OUTPUT:  
>>

Nilai terkecil pada UTS adalah : 55

***Contoh 4: Progam untuk membuat himpunan dan melakukan pemetaan untuk menghitung nilai dari Y(x) = x2***

def buat\_himpunan\_interval(angka\_awal, angka\_akhir):

himpunan = []

for i in range(angka\_awal, angka\_akhir + 1):

himpunan.append(i)

return himpunan

def kuadrat(himpunan):

hasil = []

for i in himpunan:

hasil.append(i\*i)

return hasil

print(kuadrat(buat\_himpunan\_interval(1,7)))

OUTPUT:

>> [1, 4, 9, 16, 25, 36, 49]

**Tugas Praktikum**

# Buat lah suatu fungsi yang dapat membuat fungsi himpunan

# dan Fungsi dari Y(x) = x^2 + 2x + 1

# Isilah garis bawah untuk membuat program bekerja

def buat\_himpunan\_interval( \_\_\_, \_\_\_ )

himpunan = []

for i in range(\_\_\_, \_\_\_\_\_ + 1)

himpunan.append()

return \_\_\_\_

def hasil\_fungsi(himpunan):

hasil = []

for i in himpunan:

hasil.append(\_\_\_\_)

return hasil

print(f"Nilai kuadrat fungsi adalah { \_\_\_ }")

**Tugas Unit 10:**

1. Buatlah analisis program untuk contoh 1 dan 2 pada Unit Python List
2. Buatlah sebuah program untuk menghitung nilai akhir dari seorang mahasiswa. Terdapat 4 variabel yang akan user masukkan yaitu nama, tugas, quiz, dan ujian secara berurutan dimana masing-masing dari tugas, quiz dan ujian terdiri dari beberapa nilai sehingga didefinisikan sebagai list yang akan user tentukan sendiri banyak elemennya. Nilai akhir dirumuskan dengan 40% ujian, 35% tugas, dan 25% quiz. Tampilkan juga index huruf dari nilai akhir yang akan program tampilkan bersama dengan nama dan nilai akhir. Buatlah analisis programnya.

|  |  |
| --- | --- |
| Index Huruf | Nilai |
| A | ≥ 90 |
| B | ≥ 80 |
| C | ≥ 70 |
| D | ≥ 60 |
| E | < 60 |