UNIT 6

Pengantar Bahasa Pemrograman Python

Tujuan:

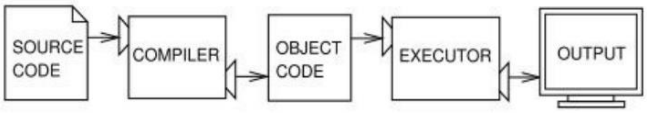
1. Mampu mengerti bahasa pemrograman python dengan baik.
2. Mampu memahami sistematika penulisan bahasa pemograman Pyhton beserta tipe datanya.
3. Mampu memahami bentuk umum algoritma dan implementasinya dalam bahasa python.
4. Mampu membuat program sederhana yang melibatkan input/output, assignment, variable, konstnta, ekspresi, dan tipe data bentukan.

Dasar Teori:

Bahasa Pemrograman Python

Python merupakan bahasa pemrograman dinamis yang mendukung pemrograman berbasis objek. File python selalu diakhiri dengan *.py* yang mengindikasikan bahwa file tersebut adalah *python program*.

interpreter memproses program lebih cepat,secara bergantian membaca kode dan memperlihatkan hasilnya



kompiler menerjemahkan kode ke kode objek, lalu dijalankan oleh eksekutor perangkat keras

File python selalu diakhiri dengan *.py* yang mengindikasikan bahwa file tersebut adalah *python program*. Editor akan menjalankan file tersebut melalui *python interpreter*, yang mana akan membaca program dan menentukan arti dari tiap kata pada program tersebut.

Fitur fitur pada Pyhton

Beberapa **fitur** yang dimiliki Python adalah :

1. Memiliki kepustakaan yang luas; dalam distribusi Python telah disediakan modul-modul siap pakai untuk berbagai keperluan.
2. Memiliki tata bahasa yang jernih dan mudah dipelajari.
3. Memiliki aturan *layout* kode sumber yang memudahkan pengecekan, pembacaankembali dan penulisan ulang kode sumber. berorientasi obyek.
4. Memiliki sistem pengelolaan memori otomatis (garbage collection, seperti java)modular, mudah dikembangkan dengan menciptakan modul-modul baru; modul-modul tersebut dapat dibangun dengan bahasa Python maupun C/C++.

5. Memiliki fasilitas pengumpulan sampah otomatis, seperti halnya pada bahasapemrograman Java, python memiliki fasilitas pengaturan penggunaan ingatankomputer sehingga para pemrogram tidak perlu melakukan pengaturan ingatankomputer secara langsung.

Kelebihan Kekurangan Pada Pyhton

Beberapa kelebihan bahasa Python antara lain :

1. Tidak ada tahapan kompilasi dan penyambungan (link) sehingga kecepatan perubahan pada masa pembuatan system aplikasi meningkat.
2. Tidak ada deklarasi tipe sehingga program menjadi lebih sederhana, singkat, dan fleksible.
3. Manajemen memori otomatis yaitu kumpulan sampah memori sehingga dapat menghindari pencatatan kode
4. Tipe data dan operasi tingkat tinggi yaitu kecepatan pembuatan system aplikasimenggunakan tipe objek yang telah ada
5. Pemrograman berorientasi objek
6. Pelekatan dan perluasan dalam C
7. Pengenalan Python7
8. Terdapat kelas, modul, eksepsi sehingga terdapat dukungan pemrograman skala besarsecara modular
9. Pemuatan dinamis modul C sehingga ekstensi menjadi sederhana dan berkas biner yangkecil
10. Pemuatan kembali secara dinamis modul phyton seperti memodifikasi aplikasi tanpa menghentikannya
11. Model objek universal kelas Satu
12. Konstruksi pada saat aplikasi berjalan
13. Interaktif, dinamis dan alamiah
14. Akses hingga informasi interpreter
15. Portabilitas secara luas seperti pemrograman antar platform tanpa ports
16. Kompilasi untuk portable kode byte sehingga kecepatan eksekusi bertambah dan melindungi kode sumber
17. Antarmuka terpasang untuk pelayanan keluar seperti perangkat Bantu system, GUI,persistence, database, dll

Beberapa kekurangan bahasa Python antara lain :

1. Beberapa penugasan terdapat diluar dari jangkauan python, seperti bahasa pemrograman dinamis lainnya, python tidak secepat atau efisien sebagai statis, tidak seperti bahasa pemrograman kompilasi seperti bahasa C.
2. Disebabkan python merupakan interpreter, python bukan merupakan perangkat bantu terbaik untuk pengantar komponen performa kritis.

3. Python tidak dapat digunakan sebagai dasar bahasa pemrograman implementasi untuk beberapa komponen, tetapi dapat bekerja dengan baik sebagai bagian depan skrip antarmuka untuk mereka.
4. Python memberikan efisiensi dan fleksibilitas tradeoff by dengan tidak memberikannya secara menyeluruh.

Identitas Python

Bahasa pemograman Python adalah bahasa pemograman yang mudah dibaca dan terstruktur, hal ini karena digunakannya sistem indentasi. Yaitu memisahkan blok - blok program dengan susunan indentasi. Jadi untuk memasukan sub - sub program dalam suatu blok, sub - sub program tersebut diletakkan satu atau lebih spasi dari kolom suatu blok program.

Python memiliki sedikit perbedaan pada cara penulisan program dengan bahasa pemrograman yang lain seperti C/Java. Kalau pada C/Java menggunakan tanda kurung sebagai pemisah blok program, di Python kita hanya menggunakan spasi sebagai pemisah blok program yang biasa disebut sebagai Indentasi. Karena Python menjalankan perintah secara berurutan, maka kita harus pintar menyusun perintah agar mendapatkan hasil seperti yang diinginkan.

Python mempunyai 28 kata kunci yang mendefinisikan aturan - aturan dan struktur bahasa, dan mereka tidak dapat digunakan sebagai nama variabel.

Variabel dan Konstanta

Variabel

Sama seperti bahasa c, variabel pada python merupakan tempat untuk menyimpan suatu nilai yang mana nilai tersebut bisa berubah-ubah sejalan dengan program. Pada bahasa python, variabel bisa langsung digunakan tanpa harus dideklarasikan terlebih dahulu.

```
>>> a = "belajar Python"
>>> b = 5
>>> phi = 3.14
```

Konstanta

Konstanta adalah tipe dari variable yang nilainya tidak dapat diubah. Pada Python, konstanta biasanya di deklarasikan dan ditetapkan pada sebuah modul. Modul adalah file baru yang berisi variabel, fungsi, *dll.* yang dapat diimport kedalam file utama. Didalam modul, konstanta ditulis dengan huruf kapital dan underscore untuk memisahkan kata.

Contoh Mendeklarasikan dan Menetapkan Nilai Pada Konstanta:

Membuat file dengan nama **konstanta.py**	`PHI = 3.14` `GRAVITASI = 9.8`

Membuat sebuah file dengan nama **main.py**	`import konstanta`
	
	`print(konstanta.PHI)`
	`print(konstanta.GRAFITASI)`

Maka Output	`3.14` `9.8`

Untuk penamaan dari variabel dan konstanta harus mengikuti beberapa aturan berikut:

- Nama dari suatu variabel hanya bisa terdiri dari huruf, angka dan garis bawah (underscore)
- Bisa diawali dengan huruf ataupun garis bawah tapi tidak diperbolehkan menggunakan awalan angka. Sebagai contoh, variabel bisa dinamai dengan nama_1 tapi tidak dengan 1_nama.
- Hindari menggunakan keyword atau sebuah fungsi sebagai nama dari suatu variabel.
- Nama dari suatu variabel harus simpel tapi bersifat deskriptif
- Hati-hati ketika menggunakan huruf kecil dari l dan huruf kapital dari O karena bisa disalah artikan sebagai angka 1 dan 0
- Statemen yang tidak boleh dijadikan nama variabel adalah keywords pada Python.

Tipe Data

Tipe Data	Fungsi	Format
str()	Untuk konversi type data ke String	%d
int()	Untuk konversi type data ke Interger	%f
float()	Untuk konversi type data ke Float	%s

Number

Tipe data Number merepresentasikan nilai-nilai berupa angka. Python menggolongkan beberapa tipe data umum seperti integer (bilangan bulat) dan floating-point (bilangan desimal) ke dalam tipe data number.

String

Selain angka, python juga mampu melakukan manipulasi string, yang dapat diekspresikan dengan beberapa cara. Penulisan nilai string pada python menggunakan tandapetik satu (') atau tanda petik dua (“).

Operator String literal juga dapat menggabungkan beberapa baris dalam berbagai cara. Dengan menggunakan operator (\n\ ) di akhir kalimat untuk menyambung kalimat selanjutnya yang berada di baris selanjutnya

Penulisan string untuk multiple line juga dapat dilakukan dengan menggunakan tanda petik dua atau satu sebanyak 3 kali, (“ “ “ atau ' ' '). Untuk menggabungkan dua buah string atau lebih dapat dilakukan dengan dua cara. Pertama, dengan menulis langsung dua buah string yang diapit dengan tanda kutip atau dengan penggunaan operator tambah (+).

```
>>>"Belajar" "python"
'Belajarpython'
>>> 'Pulang' + 'Pergi'
'PulangPergi'
```

Sebuah string, setiap karakternya dapat diindex, seperti pengindexan pada bahasa C.Karakter pertama pada sebuah string berindex 0, karakter ke-dua berindex 1 dan seterusnya.

```
>>> kata = "Veteran"
>>> kata[0]
'V'
>>> kata[4:7]
'eran'
>>> kata[:6]
'Vetera'
>>> kata[5:]
'ran'
```

Python merupakan bahasa pemrograman yang fleksibel, python memiliki beberapa bentuk dalam melakukan proses print, yakni:

```
angka = 5
#Menggunakan format seperti C
print("1 | Nilai dari angka adalah : %d" %(angka))

#Menggunakan concatenation
#Karena proses print dalam bentuk string
#maka fungsi str() digunakan untuk mengubah integer menjadi string
print("2 | Nilai dari angka adalah : " + str(angka))
```

```
#Menggunakan pemisahan koma
print("3 | Nilai dari angka adalah :" ,angka)


#Menggunakan bentuk format()
print("4 | Nilai dari angka adalah : {}".format(angka))


#Menggunakan f-string
print(f"5 | Nilai dari angka adalah : {angka}")
OUTPUT :
>>
1 | Nilai dari angka adalah : 5
2 | Nilai dari angka adalah : 5
3 | Nilai dari angka adalah : 5
4 | Nilai dari angka adalah : 5
5 | Nilai dari angka adalah : 5
```

Selain itu tipe data variabel string memiliki beberapa fungsi khusus, yang digunakan untuk mengubah dan memanipulasi teks, seperti dibawah berikut:

Nama	Keterangan
upper()	Mengonversi huruf kecil dalam bentuk string ke huruf besar.
lower()	Hapus item pada list sesuai dengan nilai yang didefinisikan
isalpha()	Mengembalikan true jika string memiliki minimal 1 karakter dan semua karakter adalah abjad dan false sebaliknya.
isnumeric ()	Mengembalikan true jika string unicode hanya berisi karakter numerik dan false sebaliknya.
strip()	Mengubah kata pada suatu kalimat menjadi *list*

Berikut merupakan program dari setiap fungsi diatas yang diawali dengan deklarasi multi line string petik tiga (‘’’), program dibawah bertujuan untuk mendeteksi dan mengubah teks yang telah di deklarasikan, yakni:

1. Mendeteksi apakah seluruh teks tersusun atas huruf alphabet
2. Mendeteksi apakah seluruh teks tersusun bilangan numerik
3. Mengubah seluruh teks menjadi huruf kecil
4. Mengubah seluruh teks menjadi huruf besar
5. Mengubah seluruh teks menjadi list

```
# mendeklarasikan string multi line dengan petik 3
teks = '''
Saya adalah mahasiswa Teknik Elektro angkatan 2021
dengan NIM 2110314069
'''
# Mengecek apakah seluruh teks adalah abjad
# JIka benar akan mengembalikan nilai True
print(f"Apakah Teks berbentuk abjad semua ? : {teks.isalpha()}")

# Mengecek apakah seluruh teks adalah angka
# JIka benar akan mengembalikan nilai True
print(f"Apakah Teks berbentuk angka semua ? : {teks.isnumeric()}")

#mengubah teks kedalam huruf kecil semua
print(f"\nMengubah teks menjadi huruf kecil : {teks.lower()}")

#mengubah teks kedalam huruf kapital semua
print(f"Mengubah teks menjadi huruf kapital : {teks.upper()}")

#Mengubah teks ke dalam bentuk list
print(f"Mengubah kedalam bentuk list : \n{teks.split()}")
```

OUTPUT

>> Apakah Teks berbentuk abjad semua ? : False

>> Apakah Teks berbentuk angka semua ? : False

>>

Mengubah teks menjadi huruf kecil :

saya adalah mahasiswa teknik elektro angkatan 2021

Mengubah teks menjadi huruf kapital :

SAYA ADALAH MAHASISWA TEKNIK ELEKTRO ANGKATAN 2021

DENGAN NIM 2110314069

#Mengubah teks ke dalam bentuk list

print(f"Mengubah kedalam bentuk list : \n{teks.split()}")

['Saya', 'adalah', 'mahasiswa', 'Teknik', 'Elektro', 'angkatan', '2021', 'dengan', 'NIM', '2110314069']

Format Penulisan Integer:

Format	Keterangan
%f	Akan menampilkan 6 digit angka dibelakang koma
%xf	Akan menampilkan sebanyak x digit dengan spasi sebagai pemisah
%0xf	Akan menampilkan angka sebanyak x digit dengan 0 sebagai pemisah
%.xf	Akan menampilkan x digit dibelakang koma
*catatan x diisi dengan angka yang diinginkan	

Operator

Operator Matematika dan Operator Penugasan

Operator	Keterangan
*	Perkalian
/	Pembagian
%	Modulo
+	Penjumlahan
-	Pengurangan

Operator	Contoh	Sama dengan
*=	x *= 50	x = x * 50
/=	x /= 50	x = x / 50
%=	x %= 50	x = x % 50
+=	x += 50	x = x + 50
-=	x -= 50	x = x - 50

Operaator Perbandingan

Operator	Keterangan	Operator	Keterangan
>=	Lebih besar atau sama dengan	<	Lebih kecil
<=	Lebih kecil atau sama dengan	>	Lebih besar
!=	Tidak sama dengan	==	Sama dengan

Contoh Program:

Contoh 1: Mengukur Luas Lingkaran

```
# Mengukur luas lingkaran
pi = 3.14159
jari_jari = input("Masukkan nilai jari-jari : ")
# Mengubah jari2 ke dalam float
jari_jari = float(jari_jari)
luas_lingkaran = pi * jari_jari * jari_jari
print(f"Luas dari lingakaran adalah : {luas_lingkaran}")
OUTPUT:
>> Masukkan nilai jari-jari : 7
>> Luas dari lingakaran adalah : 153.93791
```

Contoh 2: Membuat Kalkulator Sederhana

```
#Contoh Program#
#MenghitungBilanganSederhana#

a = float(input("Nilai a : "))
b = float(input("Nilai b : "))
c = float(input("Nilai c : "))
d = float(input("Nilai d : "))
jumlah= a+b+c+d
kali= a*b*c*d

print("Jumlah Semua Bilangan : ",jumlah)
print("Kali Semua Bilangan : ",kali)
OUTPUT:
>>
Nilai a : 1
Nilai b : 2
Nilai c : 3
Nilai d : 4
Jumlah Semua Bilangan :  10.0
Kali Semua Bilangan :  24.0
```

Tugas Praktikum

```
# Buatlah suatu program yang dapat menghitung Volume dari tabung
# Dengan jari jari dan tinggi merupakan input dari pengguna
# Dan teks yang menunjukkan
# Isilah garus bawah yang kosong untuk membuat program bekerja
pi = 3.14159
tinggi = _____
jari_jari = _____

volume = _________
print(f"Nilai dari volume adalah : {volume}")
```

Tugas:

1. Membuat contoh program memasukkan Data Diri (isi data diri Anda).Dengan Format Sebagai Berikut.
   Nama:
   NIM:
   Jurusan:
   Tempat/Tanggal Lahir:
   Alamat Tempat Tinggal:
   Email:
   Nomor Telefon:

2. Membuat Operasi Hitung dari daftar Pembelian dengan Data sebagai Berikut
   Ayam 1 kilo= Rp.50000
   Tomat 1 kilo =Rp. 16000
   Wortel 1 kilo = Rp. 18000
   Timun ½ kilo = Rp.6000
   Nugget = Rp.25000

   Seorang Ibu membeli beberapa bahan makanan sebanyak 2 kilo ayam, ½ kilo Tomat, 3 kilo Wortel , 1 kilo Timun, 1 Bungkus Nugget. Buatlah harga total jumlah belanja ibu menggunakan program operasi hitung dengan input manual dari Harga telah dicantumkan.

3. Berikan penje;asan Analisis pada setiap program yang dibuat pada Tugas unit 6

UNIT 7

Percabangan

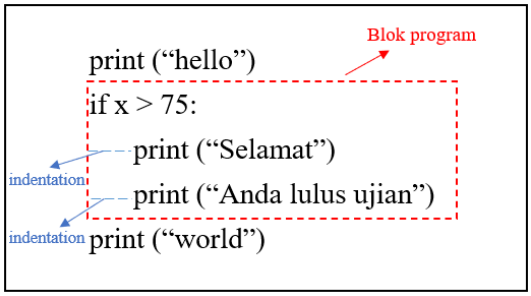
Tujuan:

1. Mampu memahami struktur penulisan percabangan menggunakan if, if-else, if-elif-else serta if bersarang
2. Mampu membuat program menggunakan fungsi kendali percabangan
3. Mampu menganalisis program yang memiliki percabangan di dalamnya.

Dasar Teori:

Pada algoritma, percabangan merupakan suatu fungsi kendali pada program. Percabangan hanya mengenal keadaan benar atau salah, true atau false, 1 atau 0 atau yang dikenal dengan istilah Boolean. Pada percabangan, jika suatu pernyataan menunjukkan nilai benar, maka suatu kondisi akan dieksekusi, namun jika menunjukkan nilai yang salah maka akan dieksekusi kondisi yang lainnya.

Perlu diingat Python menggunakan indentation untuk menyatakan satu blok. Maka pernyataan dari suatu blok percabangan perlu diberikan indentation, yang menunjukan bahwa pernyataan tersebut adalah bagian dari suatu blok percabangan.



Python membaca program dari atas ke bawah yang berarti pada kasus percabangan jika kondisi awal yang bernilai benar maka akan di eksekusi dan untuk kondisi di bawahnya tidak tereksekusi meskipun kondisi di bawah juga bernilai benar, oleh karena hal tersebut penting untuk memperhatikan urutan dari tiap kondisi. Python memiliki 3 jenis percabangan yaitu if, if else, dan if elif else serta di tambahan if bersarang.

Percabangan if digunakan untuk memilih apakah sebuah pernyataan akan dijalankan atau tidak, sesuai kondisi yang diberikan. Sintaks umum pada percabangan if

```
if kondisi :
        pernyataan jika kondisi benar
```

Percabangan if-else digunakan untuk memilih pernyataan mana yang akan dijalankan dari 2 pernyataan sesuai kondisi yang diberikan

```
if kondisi_1 :
        pernyataan jika kondisi_1 benar
else :
        pernyataan jika kondisi_1 salah
```

Percabangan if-elif-else digunakan untuk memilih pernyataan mana yang akan dijalankan dengan beberapa kondisi pengecekan.

```
if kondisi_1 :
        pernyataan jika kondisi_1 benar
elif kondisi_2:
        pernyataan jika kondisi_2 benar
elif kondisi_3:
        pernyataan jika kondisi_3 benar
.
.
.
elif kondisi_n:
        pernyataan jika kondisi_n benar
else :
        pernyataan jika seluruh kondisi salah
```

Percabangan if bersarang digunakan jika kondisi if yang paling luar bernilai benar maka percabangan if yang berada di dalamnya baru akan dicek

```
if kondisi_1 :
        if kondisi_A:
                pernyataan jika kondisi_A benar
        else :
                pernyataan jika kondisi_A salah
elif kondisi_2:
        if kondisi_B:
                pernyataan jika kondisi_B benar
        else :
                pernyataan jika kondisi_B salah
else :
        pernyataan jika seluruh kondisi salah
```

Contoh program:

Contoh 1: if - Program untuk menentukan kelulusan.

```
nilai=float(input("Masukan nilai anda : "))
if (nilai < 75) and (nilai > 65):
  print("Mendapat nilai B")
```

```
elif (nilai > 75 ) and (nilai <80):
  print("Mendapat nilai A-")
elif(nilai >80 ):
  print("Mendapat nilai A")
else:
  print("Anda Mengulang Mata Kuliah")

OUTPUT:
>>
Masukan nilai anda : 50
Anda Mengulang Mata Kuliah
```

Contoh 2: if bersarang - Program untuk mengetahui tarif parkir.

```
print("Tarif parkir mobil untuk 5 jam pertama dikenai Rp5000 perjam setelah
5 jam menjadi Rp4000")
print("Tarif parkir motor untuk 5 jam pertama dikenai Rp3000 perjam setelah
5 jam menjadi Rp2000")

mobil=5000
motor=3000
mobil_setelah5jam=4000
motor_setelah5jam=2000

print("Jenis Kendaraan")
print("1. Mobil")
print("2. Motor")
kendaraan=str(input("Pilih No. Jenis kendaraan:"))
parkir=int(input("Lamanya Kendaraan Diparkirkan...Jam:"))
if kendaraan=="1":
   if parkir>5:
      print("\nTarif Parkir RP",mobil*5+((parkir-5)*mobil_setelah5jam))
   else :
      print("\nTarif Parkir RP",parkir*mobil)
elif kendaraan=="2":
   if parkir>5:
      print("\nTarif Parkir RP",motor*5+((parkir-5)*motor_setelah5jam))
   else :
      print("\nTarif Parkir RP",parkir*motor)
else:
   print("\nInput jenis kendaraan salah")
```

OUTPUT:
>>
Tarif parkir mobil untuk 5 jam pertama dikenai Rp5000 perjam setelah 5 jam menjadi Rp4000
Tarif parkir motor untuk 5 jam pertama dikenai Rp3000 perjam setelah 5 jam menjadi Rp2000
Jenis Kendaraan

1. Mobil

2. Motor

Pilih No. Jenis kendaraan:1

Lamanya Kendaraan Diparkirkan...Jam:5

Tarif Parkir RP 25000

Tugas Praktikum

```
# Program untuk mencari nilai terbesar dari 3 nilai input
# Isilah garis bawah kosong untuk membuat program bekerja

nilai_1 = int(input("Masukkan nilai pertama :"))
nilai_2 = int(input("Masukkan nilai kedua : "))
nilai_3 = int(input("Masukkan nilai ketiga : "))

if nilai_1 > nilai_2:
  if ____ > _____:
    print("Nilai terbesar adalah nilai-1 ")
  elif ____ < ____:
    print("Nilai terbesar adalah nilai-3 ")
  else:
    print("Nilai-1 sama besarnya dengan nilai-3 ")
elif ____ < ____:
  if ____ > ____:
    print("Nilai terbesar adalah nilai-2 ")
  elif ____ < ____:
    print("Nilai terbesar adalah nilai-3 ")
  else:
    print("Nilai-2 sama besarnya dengan nilai-3")
else:
  print("Nilai-1 sama besarnya dengan nilai-2")
```

Tugas Unit 7:

1. Buatlah program kalkulator sederhana menggunakan Python. User akan memasukkan dua buah bilangan riil dan satu buah operator aritmatika (+, -, *, atau /), kemudian program akan mengoperasikan dua bilangan tersebut dengan operator yang sesuai. Contoh tampilan program:

-Masukan angka pertama: -Pilih jenis operator: -Masukan angka kedua: -Tampilkan hasil

2. Toko Karisma menjual pakaian yaitu baju dengan harga Rp55.000 dan celana dengan harga Rp 45.000. Toko Karisma memberikan diskon kepada pembeli dengan ketentuan diskon sebagai berikut jika total pembelian pakaian Rp 300.000 atau lebih maka mendapatkan diskon sebesar 10% lalu jika pembeli memiliki kartu member diskon ditambah 5% menjadi 15%, jika total pembelian pakaian mencapai Rp 500.000 atau lebih maka diskon sebesar 15% lalu jika pembeli memiliki kartu member diskon ditambah 5% menjadi 20%, dan Toko karisma masih memberikan diskon walaupun pembelian pakaian dibawah Rp 300.000 jika pembeli memiliki kartu member yaitu sebesar 5%. Buat program agar user mengetahui harga akhir dari pembelian pakaian, dengan hal yang akan di tampilkan.

-Jumlah baju yang dibeli: -Jumlah celana yang dibeli: -Apakah Anda menggunakan kartu member: -Tampilkan diskon yang didapatkan -Tampilkan harga awal -Tampilkan harga akhir setelah diskon

UNIT 8

Perulangan

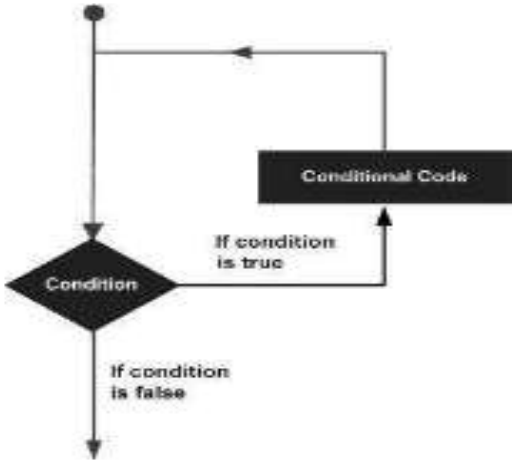
Tujuan:

1. Mampu memahami struktur penulisan perulangan menggunakan for, while dan range.
2. Mampu membuat program perulangan menggunakan for, while dan range.
3. Mampu menganalisis program yang memiliki perulangan di dalamnya.

Dasar Teori:

Struktur Pengulangan For

Tujuan utama dari perulangan adalah agar tidak menuliskan perintah yang sama dan akan diekesekusi secara berulang-ulang. Perintah for dalam python mempunyai ciri khas tersendiri, statement for bekerja untuk mengulang tipe data yang sekuensial seperti List, String, dan Tuple.



For Tunggal

Struktur penulisan:

```
for variabel in iterable :
    pernyataan 1
    pernyataan 2
    dan seterusnya
```

Range

Struktur penulisan :

- range(nilai_awal, nilai_akhir, pencacah)
- range(nilai_awal, nilai_akhir)
- range(nilai_akhir)

Perintah Perulangan dengan For Bertingkat

Struktur penulisan :

```
for variabel1 in iterable :
    for variable2 in iterable :
        …
        for variable in iterable :
            pernyataan 1
            pernyataan 2
            dan seterusnya
```

While

Ketika menggunakan sebuah perulangan while terlebih dahulu harus mengatur variabel perulangan seperti berikan nilai awal, cara uji kondisi untuk menentukan penyelesaian, lalu pastikan untuk memastikan perulangan akan berhenti setelah eksekusi selesai.
Berbeda dengan perulangan for yang digunakan jika diketahui jumlah maksimal putaran yang diperlukan untuk mengeksekusi body.

```
While conditions:
        Statement 1
        Statement 2
        …..
        Statement n
```

Contoh Program:

Contoh 1: For Loop – Program untuk menampilkan bilangan dengan batas tertentu.

```
nilai_awal = int(input('Masukkan nilai awal = '))
nilai_akhir = int(input('Masukkan nilai akhir = '))


sum = 0
for i in range(nilai_awal, nilai_akhir+1) :
  sum = i + sum
print(f"Nilai total dari sum adalah : {sum}")OUTPUT
>>
Masukkan nilai awal = 1
Masukkan nilai akhir = 5
Nilai total dari sum adalah : 15
```

Contoh 2: For Bertingkat - Program untuk menghitung nilai rata-rata rata-rata

```
banyakMahasiswa = int(input('banyak mahasiswa: '))

for M in range(banyakMahasiswa) :
   print('Mahasiswa ke-',M+1)
   banyakMK = int(input('banyak matakuliah yang diambil: '))
   totalnilai=0

   for N in range(banyakMK) :
      nilai = int(input('input nilai ke %d :' %(N+1)))
      totalnilai=totalnilai+nilai

   rerata=totalnilai/banyakMK
   print('rata-rata : ',rerata)
```

Contoh 3: While Loop – Program games angka sederhana yang menggunakan hukum matematika trichotomy

```
permutasi = int(input("Masukkan nilai permutasi : "))

i = 1

hasil = 1

while i  < permutasi + 1:

  hasil = hasil * i

  i = i + 1


print(f"Permutasi dari {permutasi}! : {hasil}")


OUTPUT

>>

Masukkan nilai permutasi : 5

Permutasi dari 5! : 120
```

Tugas Praktikum

```
# Menghitung jumlah dari barisan aritmatik ( Un )
# Nilai a (nilai awal), b (beda) dan n (nilai ke-n) dari input pengguna


a = int(input("Masukkan nilai a : "))
b = int(input("Masukkan nilai b: "))
n = int(input("Masukkan nilai n : "))


for i in range(n+1):
    # Rumus dari Un aritmatika adalah Un = a + b(n-1)
    Un = a + b*(i-1)
    i = i + 1


print(f" Nilai Un adalah : {Un}")
```

Tugas Unit 8:

1. Buatlah program yang menampilkan dari bilangan tertentu sampai bilangan tertentu dan menghitung banyaknya bilangan serta menghitung penjumlahan dari seluruh bilangan yang ada. Note: user dapat memasukkan input berupa nilai awal dan nilai akhir yang berupa bilangan riil, kemudian program akan mengoperasikan dua bilangan tersebut dan menghitung penjumlahan dari seluruh bilangan yang ada. Contoh tampilan program:

   ```
   Batas bawah bilangan :
   Batas atas bilangan :
   Bilangan antara (bilangan atas) dan (bilangan bawah)
   Hasil semua bilangan
   ```

2. Buatlah program yang menampilkan penjumlahan matrix A dan matrix B, dimana Matrix A $= \begin{bmatrix} 8 & 8 \\ 12 & 21 \end{bmatrix}$ dan Matrix B $= \begin{bmatrix} 31 & 13 \\ 21 & 12 \end{bmatrix}$ menggunakan perintah perulangan for.

3. Ammar adalah seorang mahasiswa Teknik Elektro UPN “Veteran” Jakarta yang kost di sekitar wilayah UPNVJ Limo. Karena sering kali kehabisan uang di tanggal muda padahal baru saja diberikan uang bulanan oleh orang tuanya, dia berinisiatif untuk membuat sebuah program yang dapat menghitung keuangannya setelah mengeluarkan beberapa uang untuk pengeluaran sehari – harinya. Note: User dapat menginput besaran pengeluaran berkali – kali dan dapat memasukkan input untuk keluar dan melihat besaran sisa saldo dan jika pengeluaran lebih besar daripada saldo awal, output yang keluar “Saldo tidak cukup”. Contoh tampilan program:

```
Jumlah saldo awal :
Pengeluaran pertama :
Pengeluaran ke-2 :
…
Pengeluaran ke-n :
Sisa Saldo
```

   a. Buatlah program pengulangan *while* yang dapat memenuhi soal diatas.
   b. Buatlah analisis dari program diatas

UNIT 9

List, Tuple dan Dictionary

Tujuan:

1. Mampu memahami struktur penulisan list pada python
2. Mampu membuat program menggunakan list atau array
3. Mampu menganalisis program yang memiliki larik atau array didalamnya

Dasar Teori:

List adalah sebuah kumpulan item dalam susunan tertentu. List dapat disusun dari kata, angka hingga nama orang. Tanda kurung siku mengindikasikan sebuah list, tiap-tiap elemen dari list dipisahkan oleh tanda koma.

```
komponen_elektronika = [‘resistor’, ’kapasitor’, ‘induktor’, ‘transistor’]
```

Ketika mendefinisikan sebuah list perlu diingat bahwa posisi index dimulai dari 0, index 1 milik item kedua dan seterusnya. Pada contoh list diatas, resistor memiliki index 0 dan kapasitor memiliki index 1. Python memiliki syntax khusus untuk mengakses elemen terakhir pada sebuah list. Dengan menggunakan index -1, program akan mengembalikan nilai dari item terakhir pada sebuah list.

```
komponen_elektronika = [‘resistor’, ’kapasitor’, ‘induktor’, ‘transistor’]
print(komponen_elektronika[0])
# menampilkan nilai dari item pertama (resistor) pada list komponen_elektronika
print(komponen_elektronika[-2])
# menampilkan nilai item kedua dari terakhir (induktor) pada list komponen_elektronika
```

Array Multidimensi

Dengan bantuan list, array 2 dimensi dan 3 dimensi dapat dibuat pada python. Cukup dengan mengatur letak dari tanda kurung siku, array 2 dimensi ataupun 3 dimensi dapat dibuat dengan mudah.

```
# mendefinisikan array 3 dimensi dengan elemen *
array = [[ ['*' for col in range(6)] for col in range(4)] for row in range(3)]
print(array)
```

Melakukan loop pada list

Mengulangi perintah untuk setiap elemen pada list dilakukan dengan menggunakan perintah 'for *variabel* in *nama_list*'. Sama seperti perulangan, syntax ini memungkinkan program melakukan perintah yang sama untuk setiap elemen dalan list.

```
komponen_elektronika = ['resistor', 'kapasitor', 'induktor', 'transistor']

for x in komponen_elektronika:

        print(x)

# menampilkan tiap elemen pada list komponen_elektronika
```

Syntax	Tujuan Penggunaan	Spesifikasi
.append()	Menambahkan elemen baru	Menambahkan elemen baru pada akhir list
.insert()	Menambahkan elemen baru	Menambahkan elemen baru pada index tertentu
del statement	Menghapus elemen	Menghapus elemen pada index tertentu dari sebuah list
.pop()	Menghapus elemen	Menghapus elemen dari list dan mengembalikan nilainya jika diperlukan
.remove()	Menghapus elemen	Menghapus elemen dari list jika tidak diketahui index elemen tersebut
.sort()	Menyusun elemen	Menyusun elemen sesuai alphabet
sorted()	Menyusun elemen	Menyusun elemen sesuai alphabet namun hanya sementara
.reverse()	Menyusun elemen	Me-reverse susunan elemen
range()	Menghasilkan angka	Menghasilkan angka dengan rentang tertentu
list()	Membuat sebuah list	Mengubah suatu atau beberapa nilai menjadi list
nama_list[a:b]	Memotong list	Memotong list dengan rentang dari index a ke b

min()	Mencari nilai minimum	Mencari nilai minimum pada list terdiri dari angka

```
x = [1,2,3,4,5,6,7,8,9,10]
y = []
for i in x:
  #Program untuk menghitung luas fungsi kuadrat y = x^2
  #dengan memetakan masing-masig nilai x
  y.append(i*i)

print(f"Nilai y setelah dimetakan adalah : {y}")

#Mencoba fungsi remove()
#Menghilangkan nilai 100
y.remove(100)
print(f"List setelah enghilangkan nilai 100 pada list : {y}")

#Mencoba fungsi insert()
#Memasukkan nilai 100 kembai ke list
# 9 adalah index dari list, index dimulai dari 0
y.insert(9, 100)
print(f"List setelah memasukkan nilai x kembali : {y}")
```

```
#Mencoba menggunakan fungsi count()
#Menghitung banyaknya angka 1 di dalam list
print(f"Menghitung banyak nilai y : {y.count(1)}")


#mencoba menggunakan fungsi pop()
#Menghapus item terakhir pada list yakni angka 100
y.pop()
print(f"Menghilangkan nilai terakhir y : {y}")
OUTPUT:
>>
Nilai y setelah dimetakan adalah : [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
List setelah enghilangkan nilai 100 pada list : [1, 4, 9, 16, 25, 36, 49, 64, 81]
List setelah memasukkan nilai x kembali : [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
Menghitung banyak nilai y : 1
Menghilangkan nilai terakhir y : [1, 4, 9, 16, 25, 36, 49, 64, 81]
```

Tuple

Tuple merupakan list yang tidak dapat diubah. Berbeda dengan list yang menggunakan tanda kurung siku [ ], untuk mendefinisikan suatu tuple gunakan tanda kurung ( ) setelah nama tuple.

```
komponen_elektronika = (‘resistor’, ’kapasitor’, ‘induktor’, ‘transistor’)
```

Dictionary

Dictionary Python berbeda dengan List ataupun Tuple. Karena setiap urutanya berisi key dan value. Setiap key dipisahkan dari value-nya oleh titik dua (:), item dipisahkan oleh koma, dan semuanya tertutup dalam kurung kurawal. Dictionary kosong tanpa barang ditulis hanya dengan dua kurung kurawal, seperti ini: {}.

```
dictionary = {‘Nama’ :‘Ahmad Budi', ‘Jurusan’ : 'teknik elekro’, ‘angkatan’ : 2021}
print(dictionary)
```

OUTPUT:

```
>> {'Nama': 'Ahmad Budi', 'Jurusan': 'teknik elekro', 'angkatan': 2021}
```

```
dictionary = {'Nama' :'Ahmad Budi', 'Jurusan' : 'teknik elektro', 'angkatan' : 2021}
# Menambahkan data
dictionary['NIM'] = 2110314069
# Membuat copy dari dictionary
dictionary_copy = dictionary.copy()
print(f"Copy : {dictionary_copy}")
# Melihat items dari dictionary
print(f"Items dari dictionary : {dictionary.items()}")
# Melihat keys dari dictionary
print(f"Keys dari dictionary : {dictionary.keys()}")
# Melihat values dari dictionary
print(f"Values dari dictionary : {dictionary.values()}")
```

OUTPUT:

```
>>
Copy : {'Nama': 'Ahmad Budi', 'Jurusan': 'teknik elektro', 'angkatan': 2021, 'NIM': 2110314069}
Items dari dictionary : dict_items([('Nama', 'Ahmad Budi'), ('Jurusan', 'teknik elektro'), ('angkatan', 2021), ('NIM', 2110314069)])
Keys dari dictionary : dict_keys(['Nama', 'Jurusan', 'angkatan', 'NIM'])
Values dari dictionary : dict_values(['Ahmad Budi', 'teknik elektro', 2021, 2110314069])
```

copy()	Mengcopy dictionary kedalam suatu variabel
items()	Melihat keys dan values dari suatu dictionary
keys()	Melihat nilai keys dari suatu dictionary
values()	Melihat values dari suatu dictionary

Contoh Program:

Contoh 1: Program membuat matriks berdasarkan masukkan dari user.

```
baris = int(input("Masukkan banyak baris : "))
kolom = int(input("Masukkan banyak kolom : "))
matriks = []
for i in range(baris):
        sementara = []
          for j in range(kolom):
                sementara.append(int(input(f"Masukkan angka : ")))
          matriks.append(sementara)
print(matriks)
```

OUTPUT:

```
>>
Masukkan banyak baris : 2
Masukkan banyak kolom : 2
Masukkan angka : 1
Masukkan angka : 2
Masukkan angka : 3
Masukkan angka : 4
[[1, 2], [3, 4]]
```

Contoh 2: Program untuk memetakan nilai x kedalam fungsi Y(x) = X^2

```
x = [1,2,3,4,5,6,7,8,9,10]
y = []
for i in x:
  #Program untuk menghitung luas fungsi kuadrat y = x^2
  #dengan memetakan nilai x
  y.append(i*i)
```

Contoh 3: Progam klasifikasi jurusan berdasarkan deskripsi teks.

```
fakultas_teknik = ["TE", "TI", "TM", "TP"]
identitas = '''
Saya adalah mahasiswa TE dengan NIM 2110314069 angkatan
2021
'''
# Memisahkan teks indentitas ke dalam bentuk list
for i in identitas.split():
  # Mengurutkan setiap nilai pada List Fakultas Teknik
  for j in fakultas_teknik:
    if i == j:
      # Mengidentifikasi apabilai jurusan mahasiswa ada di dalam jurusan fakultas
      print(f"Merupakan mahasiswa teknik dengan jurusan {i}")
OUTPUT:
>> Merupakan mahasiswa teknik dengan jurusan TE
```

Tugas Praktikum

```
# Program Membuat nilai himpunan Y dengan
# Memetakan nilai x pada interval [1,10] ke dalam fungsi Y(X) = (x^3) + 1


# Membuat list kosong berisi himpunan 1-10
x = [ _, _, _, _, _, _, _, _, _, _]


# Membuat list kosong dari Y
y = __


# Melakukan iterasi terhadap list pada nilai X
for nilai_x in __ :
  # Memasukkan persamaan fungsi
  # Memasukkan nilai ke dalam list Y
  y.append( _________ )


print(f"Himpunan nilai Y(X) = (x^3) + 1 adalah : {y} ")
```

Tugas Unit 10:

Buatlah analisis program untuk contoh 1 dan 2 pada Unit Python List !

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.

```
#deklarasi fungsi
def kalimat():
    "menampilkan kalimat Hallo, Selamat Belajar Python"
    print('Hallo, Selamat Belajar Python')

#Program Utama
#memanggil fungsi
kalimat()
```

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

```
Hallo, Selamat Belajar Python
```

Fungsi tanpa parameter dengan return

```
#fungsi hitung
def hitung():
    return 1+2

#panggil fungsi coba
tampil=hitung()
print('Hasil Perhitungan : ',tampil)
```

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 :

```
Hasil Perhitungan :  3
```

Fungsi dengan parameter tanpe return

```
#deklarasi fungsi
def datadiri(nama,kota):
    "menampilkan nama dan kota"
    print('Nama Anda : ', nama)
    print('Dari Kota : ', kota)

#Program Utama
nama=input('Masukkan nama Anda : ')
kota=input('Masukkan kota asal : ')

#memanggil fungsi
datadiri(nama,kota)
```

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 :

```
Masukkan nama Anda : saya
Masukkan kota asal : palembang
Nama Anda :  saya
Dari Kota :  palembang
```

Fungsi dengan parameter dengan return

```
#deklarasi fungsi
def perkalian(bil1, bil2):
    "menghitung perkalian 2 bilangan bulat"
    hasil = bil1 * bil2
    return hasil

#Program Utama
bil1=int(input('Input bil 1 : '))
bil2=int(input('Input bil 2 : '))
#memanggil fungsi
print('Hasil Perkalian ',bil1,' dengan ',bil2,' adalah ', perkalian(bil1,bil2))
```

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 :

```
Input bil 1 : 5
Input bil 2 : 7
Hasil Perkalian  5  dengan  7  adalah  35
```

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) = x^2

```
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:

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