


<b>Nama : armyta mecky</b> <b>NIM : 065002100039</b>	 <b>Algoritma dan Pemrograman Dasar</b>	<b>Modul 4</b>  <b>Nama Dosen:</b> Abdul Rochman
<b>Hari/Tanggal:</b>  Rabu, 13 Oktober 2021		<b>Nama Aslab:</b>  1. Grace Ester A. (064001900014) 2. Rifdah Amelia (064001900019)

## MODUL 4 : STRUKTUR PENGULANGAN (LOOPING) PART 2

**Deskripsi Modul :** Membuat program dengan menggunakan struktur pengulangan (looping)

No.	Elemen Kompetensi	Indikator Kinerja	Halaman
1.	Mampu memahami penggunaan struktur pengulangan (looping) pada Python	Memahami penggunaan struktur pengulangan (looping) pada Python	3

### TEORI SINGKAT

Python adalah bahasa pemrograman tujuan umum yang ditafsirkan, tingkat tinggi. Dibuat oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, filosofi desain Python menekankan keterbacaan kode dengan penggunaan spasi putih yang signifikan. Konstruksi bahasanya dan pendekatan berorientasi objek bertujuan untuk membantu programmer menulis kode yang jelas dan logis untuk proyek skala kecil dan besar.

Python diketik secara dinamis dan pengumpulan sampah. Ini mendukung beberapa paradigma pemrograman, termasuk pemrograman terstruktur (terutama, prosedural), berorientasi objek, dan fungsional. Python sering dideskripsikan sebagai bahasa "termasuk baterai" karena perpustakaan standarnya yang komprehensif.

Python dikandung pada akhir 1980-an sebagai penerus bahasa ABC. Python 2.0, dirilis pada tahun 2000, memperkenalkan fitur-fitur seperti pemahaman daftar dan sistem pengumpulan sampah dengan penghitungan referensi.

Python 3.0, dirilis pada tahun 2008, adalah revisi utama dari bahasa yang tidak sepenuhnya kompatibel dengan versi sebelumnya, dan banyak kode Python 2 yang tidak berjalan tanpa modifikasi pada Python 3.

Penerjemah Python tersedia untuk banyak sistem operasi. Komunitas programmer global mengembangkan dan memelihara CPython, implementasi referensi yang gratis dan bersumber terbuka. Sebuah organisasi nirlaba, Python Software Foundation, mengelola dan mengarahkan sumber daya untuk pengembangan Python dan CPython.

## DAFTAR PERTANYAAN

1. Apa perbedaan antara seleksi (conditional statement) dengan pengulangan (looping)? Jelaskan!

*Jawab:*

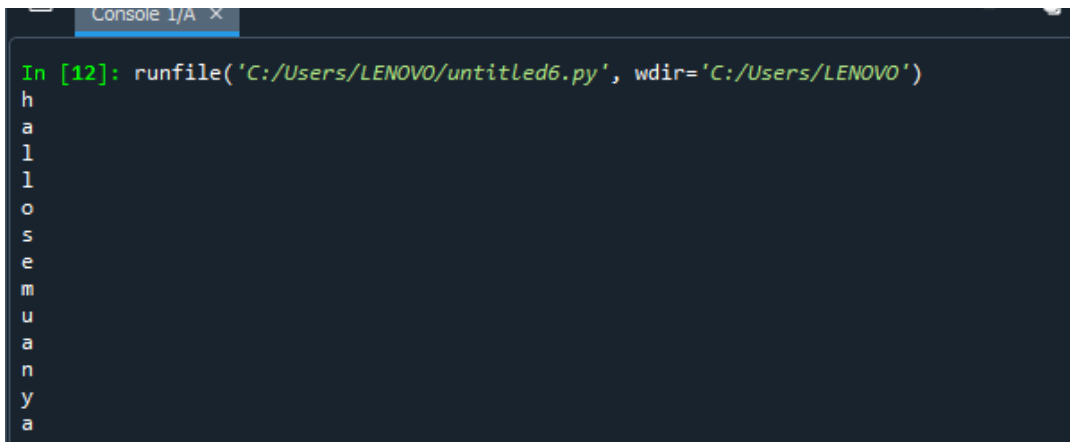
**>>Seleksi kondisi adalah proses penentuan langkah berikutnya berdasarkan proses yang terjadi sebelumnya. Seleksi kondisi ini sangat penting dalam pemrograman sebab dengan adanya seleksi kondisi, program dapat menentukan proses apa yang harus dilakukan selanjutnya berdasarkan keadaan sebelumnya. Sehingga nampak seolaholah program dapat berpikir dan mengambil keputusan. Disinilah letak kekurangan komputer yaitu tidak mampu berpikir sendiri, semua hal yang dilakukan adalah berdasarkan perintah.**

**>> Perulangan (iterasi) adalah proses yang berulang. Iterasi selalu ada dalam bahasa pemrograman apapun, karena disinilah letak kelebihan komputer dibanding manusia, yaitu mampu melakukan hal yang sama berulang kali tanpa kesalahan akibat bosan atau lelah.**

2. Apa keluaran dari potongan program berikut? Dan berikan penjelasannya!

```
a = 'halosemuanya'
for b in a:
    print(b)
```

*Jawab:*



```
Console 1/A x
In [12]: runfile('C:/Users/LENOVO/untitled6.py', wdir='C:/Users/LENOVO')
h
a
l
l
o
s
e
m
u
a
n
y
a
```

**Hasil running dari program yang telah diberikan yaitu kata hallo semuanya dengan kata yang sama tetapi dengan gaya yang memanjang**

3. Dari program di atas (program soal nomor 2) terdapat berapa variabel? Apa fungsi dari 'b' pada `for b in a`?

*Jawab:* **terdapat 2 variabel yaitu: variabel a dan variabel `for b in a`**  
**Fungsi b yaitu : untuk menyimpan data/hasil dari variabel a**

## LAB SETUP

Hal yang harus disiapkan dan dilakukan oleh praktikan untuk menjalankan praktikum modul ini, antara lain:

1. Menyiapkan IDE untuk membangun program python (Spyder, Sublime, dll);
2. Python sudah terinstal dan dapat berjalan dengan baik di laptop masing-masing;
3. Menyimpan semua dokumentasi hasil praktikum pada laporan yang sudah disediakan.

## ELEMEN KOMPETENSI I

**Deskripsi** : Membuat program konversi bilangan desimal-biner dan biner-desimal

**Kompetensi Dasar** : Memahami penggunaan looping dalam program konversi bilangan

1. Buatlah sebuah program yang dapat mengkonversi bilangan desimal ke biner dan biner ke desimal. Dengan hasil running sebagai berikut

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

D:\G\Uni\LAB ASSISTANT\AP\4>python P4-1.py

@@@@@ @@@@@ @@@ @@@@ @@@@@
@      @  @ @  @  @      @
@      @@@@@ @  @  @      @
@ @@@ @@@ @@@@@ @      @@@
@  @ @  @ @  @  @  @      @
@  @ @  @ @  @  @  @      @
@@@@@ @  @ @  @  @  @@@@@

---PROGRAM KONVERSI BILANGAN---
1 -> Desimal ke Biner
2 -> Biner ke Desimal
3 -> Exit
Silahkan pilih menu : 1
Masukkan bilangan desimal: 13
Nilai binernya adalah 1101

---PROGRAM KONVERSI BILANGAN---
1 -> Desimal ke Biner
2 -> Biner ke Desimal
3 -> Exit
Silahkan pilih menu : 2
Masukkan bilangan biner: 1101
Nilai desimalnya adalah 13

---PROGRAM KONVERSI BILANGAN---
1 -> Desimal ke Biner
2 -> Biner ke Desimal
3 -> Exit
Silahkan pilih menu : 3
Terima kasih telah menggunakan program ini!

D:\G\Uni\LAB ASSISTANT\AP\4>
```

2. Cantumkan hasil running program.

### Screenshot

The screenshot shows the Spyder Python IDE interface. The editor on the left contains a Python script for converting between decimal and binary. The console on the right shows the output of the program.

```

3 Created on Tue Oct 19 10:29:30 2021
4
5 @author: LENOVO
6 """
7
8 import sys
9 armyta=23
10 while armyta ==23:
11     print("==program koversi bilangan")
12     print("1.desimal ke biner")
13     print("2.biner ke desimal")
14     print("3.keluar")
15     armyta4=int(input("silahkan pilih menu: "))
16     if armyta4==1:
17         desimal = int(input("masukkan desimal:"))
18         if desimal ==0:
19             print(0)
20         else:
21             print("hasil bagi sisa biner")
22             bitsring=""
23             while desimal > 0:
24                 sisa= desimal %2
25                 desimal = desimal//2
26                 bitsring= str(sisa) + bitsring
27             print("nilai binernya adalah: ",bitsring)
28     elif armyta4==2:
29         bit=input("masukkan str biner: ")
30         desimal=0
31         eks = len(bit)-1
32         for digit in bit:
33             desimal += int(digit)*2**eks
34             eks-= 1
35         print ("nilai desimal adalah :",desimal)
36     elif armyta4==3:
37         sys.exit(0)
38
39

```

Console Output:

```

In [1]: runfile('C:/Users/LENOVO/untitled3.py', wdir='C:/Users/LENOVO')
==program koversi bilangan
1.desimal ke biner
2.biner ke desimal
3.keluar

silahkan pilih menu: 2

masukkan str biner: 23
nilai desimal adalah : 4
nilai desimal adalah : 7
==program koversi bilangan
1.desimal ke biner
2.biner ke desimal
3.keluar

silahkan pilih menu:

```

## ELEMEN KOMPETENSI II

**Deskripsi** : Membuat program mengecek angka genap dalam list angka

**Kompetensi Dasar** : Memahami penggunaan looping dalam program mengecek angka genap

1. Buatlah sebuah program yang dapat mengecek apakah terdapat angka genap dalam list angka.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

D:\G\Uni\LAB ASSISTANT\AP\4>python P4-2.py
@@@@@  @@@@@  @@@  @@@@  @@@@@
@      @  @  @  @  @  @      @
@      @@@@@  @  @  @      @
@ @@@  @@      @@@@@  @      @@@
@  @  @  @  @  @  @  @      @
@  @  @  @  @  @  @  @      @
@@@@@  @  @  @  @  @@@@@  @@@@@

Isi list angka (integer): 1 2 3 5 7
List angka memiliki angka genap

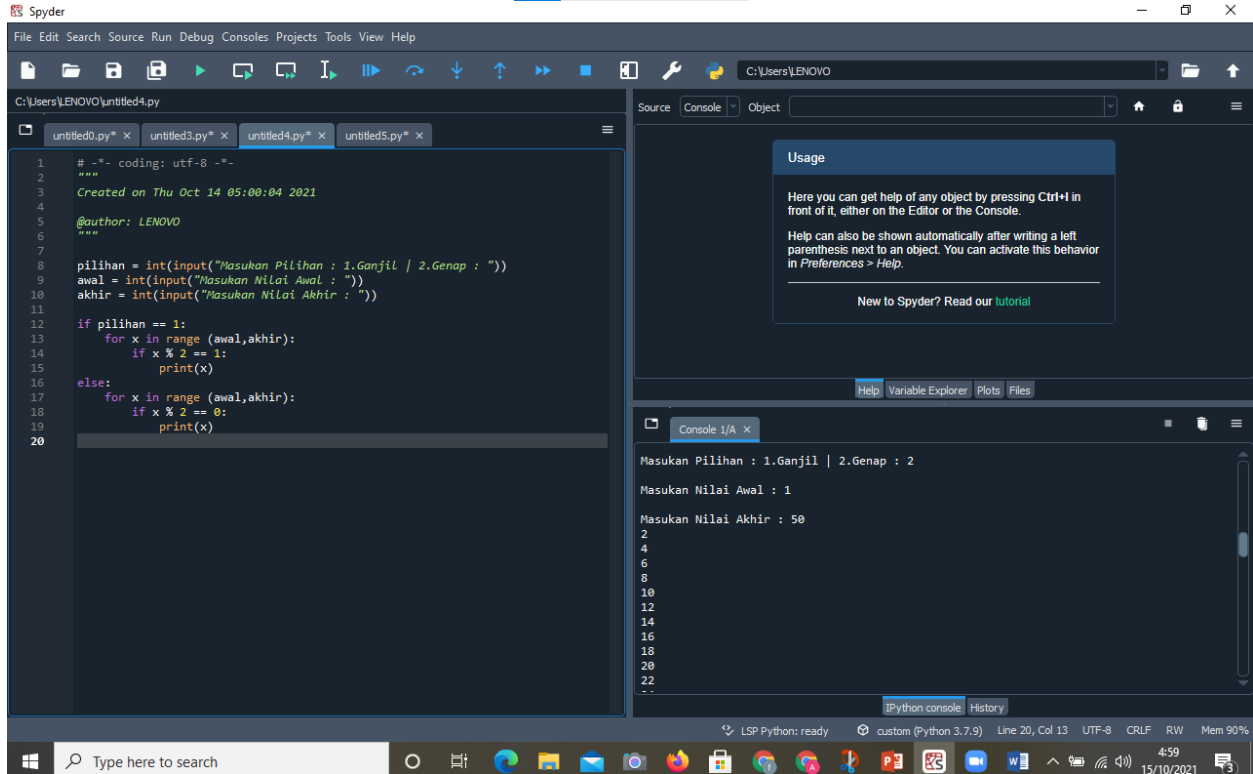
D:\G\Uni\LAB ASSISTANT\AP\4>python P4-2.py
@@@@@  @@@@@  @@@  @@@@  @@@@@
@      @  @  @  @  @  @      @
@      @@@@@  @  @  @      @
@ @@@  @@      @@@@@  @      @@@
@  @  @  @  @  @  @  @      @
@  @  @  @  @  @  @  @      @
@@@@@  @  @  @  @  @@@@@  @@@@@

Isi list angka (integer): 3 5 7 9
List angka tidak memiliki angka genap

D:\G\Uni\LAB ASSISTANT\AP\4>
```

Cantumkan hasil running program

## Screenshot



**Link Github :**

**KESIMPULAN** (*minimal 3 baris*)

>>Seleksi kondisi adalah proses penentuan langkah berikutnya berdasarkan proses yang terjadi sebelumnya. Seleksi kondisi ini sangat penting dalam pemrograman sebab dengan adanya seleksi kondisi, program dapat menentukan proses apa yang harus dilakukan selanjutnya berdasarkan keadaan sebelumnya. Sehingga nampak seolaholah program dapat berpikir dan mengambil keputusan. Disinilah letak kekurangan komputer yaitu tidak mampu berpikir sendiri, semua hal yang dilakukan adalah berdasarkan perintah.

>> Perulangan (iterasi) adalah proses yang berulang. Iterasi selalu ada dalam bahasa pemrograman apapun, karena disinilah letak kelebihan komputer dibanding manusia, yaitu mampu melakukan hal yang sama berulang kali tanpa kesalahan akibat bosan atau lelah.

**CEKLIST**

1. Membuat program menggunakan struktur looping

(√)