

PETUNJUK PRAKTIKUM

ALGORITMA DAN LOGIKA INFORMATIKA

DAFTAR ISI

Petunjuk Praktikum #1	
Pengenalan Bahasa Pemrograman Python	3
Petunjuk Praktikum #2	
Komentar, Input dan Output	3
Petunjuk Praktikum #3	
Tipe Data dan Variabel	3
Petunjuk Praktikum #4	
Operator Logika	3
Petunjuk Praktikum #5	
Fungsi Matematika Pada Python	3
Petunjuk Praktikum #6	
String Pada Python	3
Petunjuk Praktikum #7	
Teknik Percabangan	3
Petunjuk Praktikum #8	
UTS	3
Petunjuk Praktikum #9	
Teknik Percabangan	3
Petunjuk Praktikum #10	
Perulangan	3
Petunjuk Praktikum #11	
Perulangan Bersarang	3
Petunjuk Praktikum #12	
Array 1 Dimensi	3
Petunjuk Praktikum #13	
Array 2 Dimensi	3
Petunjuk Praktikum #14	
Procedure	3
Petunjuk Praktikum #15	
Function	3
Petunjuk Praktikum #16	
UAS	3

PETUNJUK PRAKTIKUM #1

PENGENALAN BAHASA PYTHON

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	Algoritma dan Logika Informatika		
	Pertemuan : 1	Komentar, Input dan Output	
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			100 menit
			Hal 1 dari 141

A. Capaian Pembelajaran Mata Kuliah

1. Mampu menjunjung tinggi kebenaran, kebaikan dan keindahan.
2. Menunjukkan sikap bertanggungjawab atas pekerjaan di bidang keahliannya secara mandiri.

B. Sub Capaian Pembelajaran Mata Kuliah

Mahasiswa mampu menjelaskan dasar logika dan notasi algoritma serta bahasa pemrograman dasar.

C. Tujuan

1. Mahasiswa mampu menjelaskan konsep algoritma dan pemrograman dasar.

D. Dasar Teori

[Python](#) adalah bahasa pemrograman interpretatif multiguna. Tidak seperti bahasa lain yang susah untuk dibaca dan dipahami, python lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintaks. Hal ini membuat Python sangat mudah dipelajari baik untuk pemula maupun untuk yang sudah menguasai bahasa pemrograman lain.

Bahasa ini muncul pertama kali pada tahun 1991, dirancang oleh seorang bernama [Guido van Rossum](#). Sampai saat ini Python masih dikembangkan oleh [Python Software Foundation](#). Bahasa Python mendukung hampir semua sistem operasi, bahkan untuk sistem operasi Linux, hampir semua distronya sudah menyertakan Python di dalamnya.

Dengan kode yang simpel dan mudah diimplementasikan, seorang programmer dapat lebih mengutamakan pengembangan aplikasi yang dibuat, bukan malah sibuk mencari syntax error.

```
print("Python sangat simpel")
```

Hanya dengan menuliskan kode print seperti yang diatas, anda sudah bisa mencetak apapun yang anda inginkan di dalam tanda kurung `()`. Dibagian akhir kode pun, anda tidak harus mengakhirinya dengan tanda semicolon `;`

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	Algoritma dan Logika Informatika		
	Pertemuan : 1	Komentar, Input dan Output	
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			100 menit
			Hal 2 dari 141

Karakteristik Python :

- Python case-sensitive, artinya perbedaan huruf besar dan huruf kecil menyebabkan perbedaan makna
- Python sangat memerhatikan indentasi dan pergantian baris. Kesalahan indentasi dapat menyebabkan gagal compile hingga kesalahan program.

E. Alat/Instrument/Aparatus/Bahan

1. Personal Computer (PC)
2. Python 3

F. Keselamatan Kerja

1. Menggunakan alat dan bahan sesuai dengan fungsinya
2. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar.
3. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen.
4. Tidak bercanda atau makan dan minum saat praktik
5. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

G. Lembar Kerja

1. Instalasi Python

Sebelum Anda menggunakan Python, Anda harus menginstalnya terlebih dahulu di sistem operasi komputer Anda. Saat ini Python memiliki 2 versi yang berbeda, yaitu Python versi **3** dan Python versi **2**. Disini kita akan belajar bahasa pemrograman Python menggunakan versi terbaru **3**.

Cara menginstal python sangat mudah, ikuti panduan dibawah ini. Dibawah adalah panduan cara instal python di platform Linux, Windows dan Mac OS.

Linux

1. Buka browser, kunjungi <http://www.python.org/downloads/source/>
2. Download versi terbaru Python berbentuk file zip untuk Unix/Linux
3. Ekstrak file zip yang baru saja di download
4. Edit file Modules/Setup jika Anda ingin kostumisasi Python
5. Jalankan `./configure` script
6. `make`

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	Algoritma dan Logika Informatika		
	Pertemuan : 1	Komentar, Input dan Output	
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			100 menit
			Hal 3 dari 141

7. `make install`

Langkah ini akan menginstal Python di lokasi standar `/usr/local/bin` dan library di `/usr/local/lib/pythonXX` dimana `XX` adalah versi terbaru Python yang anda gunakan.

Untuk beberapa distro (distribution store) dari sistem operasi linux sudah terinstal Python di dalamnya. Jadi Anda tidak perlu menginstalnya lagi.

Windows

1. Buka browser, kunjungi <http://www.python.org/downloads/windows/>
2. ATAU, klik direct link <https://www.python.org/ftp/python/3.8.1/python-3.8.1.exe>
3. Buka (klik 2x) file installer python yang baru saja di download
4. Ikuti langkah instalasi sampai selesai

Mac OS

1. Buka browser, kunjungi <http://www.python.org/download/mac/>
2. Download versi terbaru Python untuk Macintosh
3. Buka file yang baru saja di download
4. Ikuti langkah instalasi sampai selesai

Menjalankan Python (Menggunakan Shell)

1. Buka IDLE (python shell di windows), Anda bisa mencarinya di tombol **START**.
2. Tuliskan script Python Anda, contoh: `print("Selamat datang di Python")`. jika sudah tekan tombol **ENTER**, dan script Python akan dijalankan/eksekusi.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	Algoritma dan Logika Informatika		
	Pertemuan : 1	Komentar, Input dan Output	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			Hal 4 dari 141

```
File Edit Shell Debug Options Window Help
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 18 2019, 23:46:00) [MSC v.1916
32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
>>> print("Selamat datang di Python")
Selamat datang di Python
>>> |
```

Ln: 6 Col: 4

Menjalankan Python (Menggunakan Script Editor)

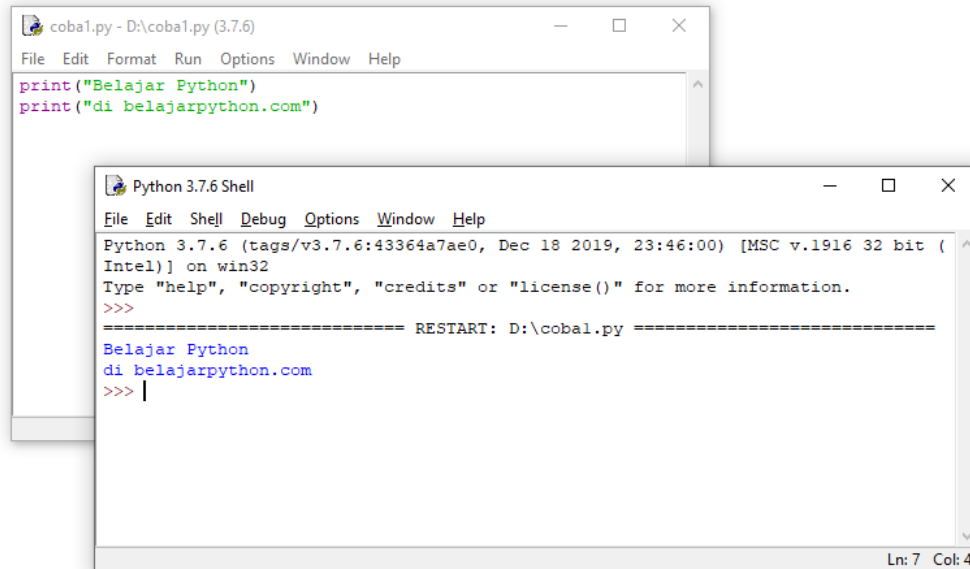
1. Untuk menjalankan script yang disimpan dalam file, buka IDLE (python shell di windows), Anda bisa mencarinya di tombol **START**.
2. Klik menu **File - New File**
3. Tulis script Python pada window yang muncul, contoh:

```
print("Belajar Python")
print("di belajarpython.com")
```

1. Simpan script lewat menu **File - Save**
2. Jalankan program dengan klik menu **Run - Run Module**

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	Algoritma dan Logika Informatika		
	Pertemuan : 1	Komentar, Input dan Output	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 5 dari 141



```

coba1.py - D:\coba1.py (3.7.6)
File Edit Format Run Options Window Help
print("Belajar Python")
print("di belajarpypython.com")

Python 3.7.6 Shell
File Edit Shell Debug Options Window Help
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 18 2019, 23:46:00) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\coba1.py =====
Belajar Python
di belajarpypython.com
>>> |
Ln: 7 Col: 4

```

Selain menggunakan Python IDLE, berikut IDE atau code editor yang dapat digunakan untuk menjalankan kode python :

- PyCharm
- VSCode
- JupyterNotebook
- GoogleColaboratory

H. Tugas Praktikum

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

PETUNJUK PRAKTIKUM #2

KOMENTAR, INPUT DAN OUTPUT

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	Algoritma dan Logika Informatika		
	Pertemuan : 1	Komentar, Input dan Output	
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 1 dari 141

A. Capaian Pembelajaran Mata Kuliah

1. Mampu menerapkan algoritma/metode dalam perangkat lunak berbasis komputer.
2. Mampu menerapkan konsep teoritis dalam bidang algoritma

B. Sub Capaian Pembelajaran Mata Kuliah

Mahasiswa mampu menggunakan komentar, dan mengimplementasikan input dan output secara tepat

C. Tujuan

1. Mahasiswa mampu mengimplementasikan input dan output
2. Mahasiswa mampu mengimplementasikan komentar

D. Dasar Teori

Komentar

Komentar (comment) adalah kode di dalam script Python yang tidak dieksekusi atau tidak dijalankan mesin. Komentar hanya digunakan untuk menandai atau memberikan keterangan tertulis pada script. Komentar biasa digunakan untuk membiarkan orang lain memahami apa yang dilakukan script. atau untuk mengingatkan kepada programmer sendiri jika suatu saat kembali mengedit script tersebut. Untuk menggunakan komentar anda cukup menulis tanda pagar `#`, diikuti dengan komentar Anda.

Input dan Output

Sebuah program dapat menampilkan output ke layar. Hal ini dapat digunakan untuk memberikan keterangan program, selain itu dapat digunakan untuk menampilkan perintah bahwa program memerlukan inputan baik dalam bentuk tipe data integer maupun karakter dari pengguna program. Di Python dapat menggunakan fungsi `print()` untuk mencetak atau menampilkan objek ke perangkat keluaran (layar) atau ke file teks.

Untuk berinteraksi dengan pengguna program atau memberikan inputan, Python memiliki fungsi `input()` yang berfungsi untuk menerima baris input dari user dan mengembalikannya dalam bentuk string.

E. Alat/Instrument/Aparatus/Bahan

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	Algoritma dan Logika Informatika		
	Pertemuan : 1	Komentar, Input dan Output	
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			100 menit
			Hal 2 dari 141

1. Personal Computer (PC)
2. Python 3

F. Keselamatan Kerja

1. Menggunakan alat dan bahan sesuai dengan fungsinya
2. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar.
3. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen.
4. Tidak bercanda atau makan dan minum saat praktik
5. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

G. Lembar Kerja Komentar

Dibawah ini adalah contoh penggunaan komentar pada Python

```
#Ini adalah komentar
# Tulisan ini tidak akan dieksekusi

#komentar dengan tanda pagar hanya bisa digunakan
#untuk
#satu
#baris

"""
Penulisan Komentar lebih dari satu baris yaitu
dengan menggunakan kutip dua 3 kali dan
ditutup dengan kutip dua 3 kali juga
"""

print("Hello World") #ini juga komentar

#print("Welcome")

# komentar bisa berisi spesial karakter !@#$$%^&*(),./; '[]\

#mencetak nama
print("Budi")

#mencetak angka/integer
print(123)
```

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	Algoritma dan Logika Informatika		
	Pertemuan : 1	Komentar, Input dan Output	
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			100 menit
			Hal 3 dari 141

Saat anda menjalankan script diatas, Anda akan melihat output berupa `Hello World`, `Budi` dan `123`, karena tulisan/komentar yang ditulis tidak dieksekusi.

Input dan Output

Dibawah ini adalah contoh penggunaan input dan output dalam bahasa pemrograman Python

```
nama = input("Nama : ")
nim = input("NIM : ")
prodi = input("Prodi : ")

print("Nama : ", nama)
print("NIM : ", nim)
print("Prodi : ", prodi)
```

Jika menginput bilangan, misalnya integer lewat fungsi `input()`, maka hasil inputan tersebut akan dibaca oleh Python sebagai string dan bukan integer atau float. Untuk mengubahnya terlebih dahulu menjadi tipe integer menggunakan fungsi `int()` dan untuk mengubahnya menjadi tipe float menggunakan fungsi `float()`. Seperti contoh ini :

```
a = input('Masukkan angka: ')
b = input('Masukkan angka : ')


hasil = int(a) + int(b)
```

H. Tugas Praktikum

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

PETUNJUK PRAKTIKUM #3

TIPE DATA DAN VARIABEL

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 1 dari 141

A. Capaian Pembelajaran Mata Kuliah

1. Mampu menerapkan algoritma/metode dalam perangkat lunak berbasis komputer.
2. Mampu menerapkan konsep teoritis dalam bidang algoritma

B. Sub Capaian Pembelajaran Mata Kuliah

Mahasiswa mampu menggunakan variable serta tipe data pada pembuatan notasi algoritma serta mengimplementasikan ke bahasa pemrograman secara tepat

C. Tujuan

1. Mahasiswa mampu mengimplementasikan penggunaan variabel
2. Mahasiswa mampu mengklasifikasikan tipe data

D. Dasar Teori

Tipe Data dan Variabel

Tipe data adalah suatu media atau memori pada komputer yang digunakan untuk menampung informasi. Python sendiri mempunyai tipe data yang cukup unik bila kita bandingkan dengan bahasa pemrograman yang lain. Dalam bahasa Python secara umum terdapat empat jenis tipe data yang bisa digunakan untuk sebuah variabel pada program. Python menyediakan empat macam tipe data dasar, yaitu :

1. int(integer), tipe data yang digunakan untuk bilangan bulat
2. float, tipe data yang digunakan untuk bilangan pecahan
3. string, tipe data yang digunakan untuk karakter
4. boolean, tipe data tipe data yang hanya memiliki dua nilai yaitu *True* dan *False*

Ada juga tipe data lainnya yang bisa digunakan dalam Python, yaitu :

1. hexadecimal, tipe data yang menyatakan bilangan dalam format heksa (bilangan berbasis 16)
2. complex, tipe data yang menyatakan pasangan angka real dan imajiner
3. list, tipe data yang menyimpan berbagai tipe data dan isinya bisa diubah-ubah
4. tuple, tipe data yang menyimpan berbagai tipe data tapi isinya tidak bisa diubah

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 2 dari 141

5. dictionary, tipe data yang menyimpan berbagai tipe data berupa pasangan penunjuk dan nilai

Variabel adalah suatu pengenal yang digunakan untuk mewakili suatu nilai tertentu didalam proses program. Variabel biasa digunakan di dalam program dengan tujuan untuk menampung data. Nilai yang terdapat pada variabel sewaktu-waktu dapat diubah. Jumlah variabel yang dibuat dapat tidak terbatas, namun masing-masing variabel tersebut harus bersifat unik dan tidak boleh ada nama variabel yang sama.

Python tidak membutuhkan deklarasi variabel secara khusus sebagaimana beberapa bahasa pemrograman lainnya. Deklarasi cukup dengan memberikan nilai awal seperti contoh berikut:

```
x = 1
```

Yang menunjukkan variabel x bertipe integer (bilangan bulat). Python juga memberikan keluwesan dalam deklarasi variabel ini. Variabel yang sama dapat berubah-ubah tipe datanya meski dalam sebuah script.


```
x = 1
x = "Saya belajar Python."
```

Baris pertama mendeklarasikan x bertipe integer, dan pada baris berikutnya x berubah tipe menjadi string.

Ketentuan mengenai variabel dalam Python :

- Variabel tidak perlu dideklarasikan mempunyai tipe data tertentu
- Jenis data dalam variabel dapat berubah-ubah
- Penulisan variabel harus diawali dengan huruf, dan untuk karakter selanjutnya bisa berupa huruf atau angka
- Dapat berupa huruf Kapital, tetapi bersifat case-sensitive, nama Kapital dengan kapital adalah variabel yang berlainan.
- Penulisan variabel tidak boleh dipisah oleh <spasi>
- Untuk variabel yang terdiri dari 2 suku kata, dapat dipisah dengan simbol underscore (_) seperti nama_saya, nama_variabel_nama.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			Hal 3 dari 141

E. Alat/Instrument/Aparatus/Bahan

1. Personal Computer (PC)
2. Aplikasi PyCharm

F. Keselamatan Kerja

1. Menggunakan alat dan bahan sesuai dengan fungsinya
2. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar.
3. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen.
4. Tidak bercanda atau makan dan minum saat praktik
5. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

G. Lembar Kerja

Tipe Data

Untuk mencoba berbagai macam tipe data, silahkan coba script Python dibawah ini.

```
#tipe data Boolean
print(True)

#tipe data String
print("Ayo belajar Python")
print('Belajar Python Sangat Mudah')

#tipe data Integer
print(20)

#tipe data Float
print(3.14)


#tipe data Hexadecimal
print(9a)

#tipe data Complex
print(5j)

#tipe data List
print([1,2,3,4,5])
print(["satu", "dua", "tiga"])

#tipe data Tuple
```

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			Hal 4 dari 141

```
print((1,2,3,4,5))
print(("satu", "dua", "tiga"))

#tipe data Dictionary
print({"nama":"Budi", 'umur':20})
#tipe data Dictionary dimasukan ke dalam variabel biodata
biodata = {"nama":"Andi", 'umur':21} #proses inisialisasi variabel
biodata
print(biodata) #proses pencetakan variabel biodata yang berisi tipe data
Dictionary
print(type(biodata)) #fungsi untuk mengecek jenis tipe data. akan tampil
<class 'dict'> yang berarti dict adalah tipe data dictionary
```

Variabel

Dibawah ini adalah contoh penggunaan variabel dalam bahasa pemrograman Python

```
#proses memasukan data ke dalam variabel
nama = "John Doe"
#proses mencetak variabel
print(nama)


#nilai dan tipe data dalam variabel dapat diubah
umur = 20 #nilai awal
print(umur) #mencetak nilai umur
type(umur) #mengecek tipe data umur
umur = "dua puluh satu" #nilai setelah diubah
print(umur) #mencetak nilai umur
type(umur) #mengecek tipe data umur

namaDepan = "Budi"
namaBelakang = "Susanto"
nama = namaDepan + " " + namaBelakang
umur = 22
hobi = "Berenang"
print("Biodata\n", nama, "\n", umur, "\n", hobi)

#contoh variabel lainya
inivariabel = "Halo"
ini_juga_variabel = "Hai"
_inivariabeljuga = "Hi"
inivariabel222 = "Bye"

panjang = 10
```

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 5 dari 141

```

lebar = 5
luas = panjang * lebar
print(luas)


```

H. Tugas Praktikum

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

PETUNJUK PRAKTIKUM #4

OPERATOR LOGIKA

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			Hal 1 dari 141

A. Capaian Pembelajaran Mata Kuliah

1. Mampu menerapkan algoritma/metode dalam perangkat lunak berbasis komputer .
2. Mampu menerapkan konsep teoritis dalam bidang algoritma.

B. Sub Capaian Pembelajaran Mata Kuliah

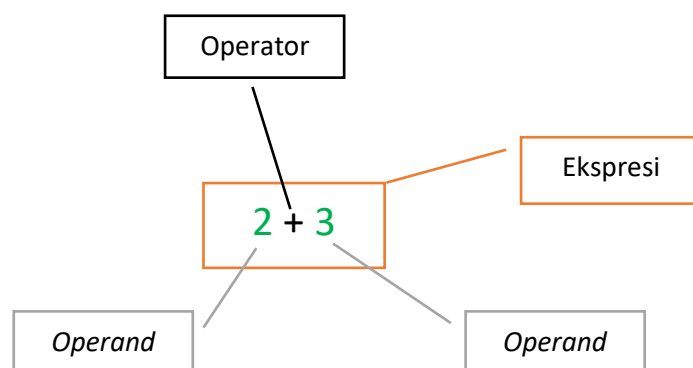
Mahasiswa mampu menggunakan beberapa jenis operator pada notasi algoritma, serta mengimplementasikan operator menggunakan bahasa pemrograman.

C. Tujuan

1. Mahasiswa mampu mengimplementasikan penggunaan operator.
2. Mahasiswa dapat **menguraikan** dan menentukan penggunaan tipe data dan operator – operator yang digunakan dalam bahasa pemrograman.


D. Dasar Teori

Operator adalah simbol atau tanda yang jika diletakkan pada dua buah operand dapat menghasilkan sebuah hasil, operator berupa simbol yang digunakan untuk menyusun suatu ekspresi dengan melibatkan satu atau beberapa *operand*. Contohnya pada matematika dimana tanda tambah ('+') jika diletakkan di antara dua buah angka akan menghasilkan angka lain hasil pertambahan dari dua angka tersebut. Tanda tambah inilah yang disebut dengan operator.



Bahasa pemrograman Python mendukung berbagai macam operator, diantaranya :

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 2 dari 141

- 1.Operator Aritmatika (Arithmetic Operators)
- 2.Operator Perbandingan (Comparison (Relational) Operators)
- 3.Operator Penugasan (Assignment Operators)
- 4.Operator Logika (Logical Operators)
- 5.Operator Bitwise (Bitwise Operators)
- 6.Operator Keanggotaan (Membership Operators)
- 7.Operator Identitas (Identity Operators)


1. Operator Aritmatika

Operator aritmatika adalah operator yang digunakan untuk melakukan perhitungan data.

Operator	Deskripsi	Contoh
+	Penjumlahan	2 + 3 bernilai 5
-	Pengurangan	3 - 2 bernilai 1
*	Perkalian	3 * 2 bernilai 6
/	Pembagian	4 / 2 bernilai 2
//	Pembagian (dibulatkan kebawah)	13 // 5 bernilai 2
%	Sisa bagi / modulus	13 % 5 bernilai 3
**	Pemangkatan	2 ** 2 bernilai 4

Contoh penggunaan operator aritmatika :

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			Hal 3 dari 141

```

a = 4
b = 2

print('4 + 2 = ' , a+b)
print('4 x 2 = ' , a*b)
print('4 / 2 = ' , a/b)
print('4 - 2 = ' , a-b)
print('4 ^ 2 = ' , a**b)
print('4 % 2 = ' , a%b)

```

Hasil yang didapat :

```

4 + 2 = 6
4 x 2 = 8
4 / 2 = 2.0
4 - 2 = 2
4 ^ 2 = 16
4 % 2 = 0

Process finished with exit code 0


```

2. Operator Perbandingan

Operator perbandingan di Python digunakan untuk melakukan suatu perbandingan antar operan. Output yang ditampilkan adalah nilai boolean **True** atau **False**. Operator perbandingan sering digunakan dalam perulangan.

Operator	Deskripsi	Contoh True	Contoh False
==	Sama dengan	2 == 2	2 == 3
!=	Tidak sama dengan	3 != 2	3 != 3
<	Kurang dari	2 < 3	3 < 2
>	Lebih dari	3 > 2	2 > 3
<=	Kurang dari sama dengan	2 <= 2	1 <= 3

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 4 dari 141

>=	Lebih dari sama dengan	6 >= 5	2 >= 5
----	------------------------	--------	--------

Contoh penggunaan operator perbandingan :

```
suhu = input("Masukan suhu: ")
if int(suhu) > 30:
    print("Hari ini panas")
else:
    print("Hari ini tidak panas")
```

Hasil yang didapat :

```
Masukan suhu: 27
Hari ini tidak panas


Process finished with exit code 0
```

3. Operator Penugasan

Operator penugasan adalah operator yang digunakan untuk memasukkan nilai atau modifikasi nilai suatu variabel.

Operator	Deskripsi	Contoh
=	Assignment	N = 30
+=	Penjumlahan	N += 5, N akan ditambah 5
-=	Pengurangan	N -= 2, N akan dikurang 2
*=	Perkalian	N *= 2, N akan dikali 2
/=	Pembagian	N /= 2, N akan dibagi 2
//=	Pembagian (dibulatkan kebawah)	N //= 5, N akan dibagi 5 (dibulatkan kebawah)

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			Hal 5 dari 141

%=	Sisa bagi / modulus	N %= 5, N akan dimodulo 5
**=	Pemangkatan	N **=5, N akan dipangkat 5

Contoh penggunaan operator penugasan :

```
x = int(input("Masukkan angka : "))
x /= 2
print('nilai x dibagi 2 ', x)
```

Hasil yang didapat :

```
Masukkan angka : 10
nilai x dibagi 2 = 5.0


Process finished with exit code 0
```

4. Operator Logika

Operator Logika adalah operator yang digunakan untuk membandingkan 2 kondisi logika, yaitu logika benar (**TRUE**) dan logika salah (**FALSE**). Operator logika sering digunakan untuk kondisi **IF**, atau untuk keluar dari proses perulangan (**looping**).

Operator	Deskripsi	Contoh True	Contoh False
and	Dan	(1 < 2) and (3 == 3)	(1 == 2) and (3 == 2)
or	Atau	(1 < 2) or (4 == 3)	(3 < 2) or (2 == 3)
not	Negasi	not (3 < 2)	not (1 > 2)

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			Hal 6 dari 141

Contoh penggunaan operator logika :

```

skp = input("Jumlah SKP: ")
sks = input("jumlah SKS: ")

if int(skp) >= 100 and int(sks) >= 75:
    print("Anda boleh Kerja Praktek")
else:
    print("Anda belum bisa Kerja Praktek")

```

Hasil yang didapat :

```

Jumlah SKP: 120
jumlah SKS: 75
Anda boleh Kerja Praktek

Process finished with exit code 0


```

5. Operator Bitwise

Operator Bitwise adalah operator untuk melakukan operasi berdasarkan bit/biner.

Nama	Simbo l	Deskripsi
AND	&	Operator biner AND, memeriksa apakah operan di sebelah kiri dan operan sebelah kanan mempunyai angka biner 1 di setiap bit. Jika keduanya bernilai 1 maka bit hasil operasi akan bernilai 1.
OR		Operator biner OR, memeriksa apakah operan di sebelah kiri dan operan sebelah kanan mempunyai angka biner 1 di setiap bit. Jika salah satunya bernilai 1 maka bit hasil operasi akan bernilai 1.
XOR	^	Operator biner XOR, memeriksa apakah operan di sebelah kiri dan operan sebelah kanan mempunyai

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 7 dari 141

		angka biner 1 di setiap bit. Jika keduanya bernilai 1 maka bit hasil operasi akan bernilai 0.
Negasi	~	Memberikan nilai variabel dengan nilai variabel itu sendiri dibagi dengan nilai di sebelah kanan.
Left Shift	<<	Memberikan nilai variabel dengan nilai variabel itu sendiri dibagi dengan nilai di sebelah kanan. Yang diambil nantinya adalah sisa baginya.
Right Shift	>>	Operator penggeser biner ke kiri, deret bit akan digeser ke kiri sebanyak n kali.
Kali sama dengan	*=	Operator biner Negative, membalik nilai bit. Misal dari 1 menjadi 0, dari 0 menjadi 1.

Contoh penggunaan operator logika :

```
x = int(input('Masukkan Angka : '))
y = int(input('Masukkan Angka : '))


print('x berisi angka', x, 'desimal atau',
      bin(x), 'biner')

print('y berisi angka', y, 'desimal atau',
      bin(y), 'biner')

print('\n')
```

Hasil yang didapatkan :

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			Hal 8 dari 141

```

Masukkan Angka : 2
Masukkan Angka : 6
x berisi angka 2 desimal atau 0b10 biner
y berisi angka 6 desimal atau 0b110 biner

x & y : 2
x | y : 6
x ^ y : 4
~x    : -3
x << 1 : 4
x >> 1 : 1

Process finished with exit code 0

```


6. Operator Keanggotaan

Operator keanggotaan adalah operator yang dipakai untuk memeriksa apakah suatu nilai ada di dalam sebuah himpunan atau tidak. Himpunan yang dimaksud terdiri dari tipe data “berbentuk array” seperti string, list, tuple, set dan dictionary.

Nama	Deskripsi	Contoh
in	Memeriksa apakah nilai yang dicari berada pada list atau struktur data python lainnya. Jika nilai tersebut ada maka kondisi akan bernilai True.	sebuah_list = [1, 2, 3,4 ,5] print 5 in sebuah_list
not in	Memeriksa apakah nilai yang dicari tidak ada pada list atau struktur data python lainnya. Jika nilai tersebut tidak ada maka kondisi akan bernilai True.	sebuah_list = [1, 2, 3,4 ,5] print 10 not in sebuah_list

Contoh penggunaan :

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 9 dari 141

```

kalimat = 'stiki'
print('Kalimat :',kalimat)
print('\i\' in kalimat      :', 'i' in kalimat)
print('\z\' not in kalimat :', 'z' not in kalimat)
print('\d\' not in kalimat :', 'd' not in kalimat)

```

Hasil yang didapatkan :

```

Kalimat : stiki
'i' in kalimat      : True
'z' not in kalimat : True
'd' not in kalimat : True

Process finished with exit code 0


```

7. Operator Identitas

Operator identitas adalah operator yang bisa dipakai untuk memeriksa apakah nilai sebuah variabel ada di tempat yang sama (di memory) atau tidak.

Nama	Deskripsi	Contoh
is	Memeriksa apakah nilai di sebelah kiri operan memiliki identitas memori yang sama dengan nilai di sebelah kanan operan. Jika sama maka kondisi bernilai True.	a, b = 10, 10 # hasil akan True print a is b
not is	Memeriksa apakah nilai di sebelah kiri operan memiliki identitas memori yang berbeda dengan nilai di sebelah kanan	a, b = 10, 5 # hasil akan True print a is not b

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			Hal 10 dari 141

	operan. Jika berbeda maka kondisi bernilai True.	
--	--	--

Contoh penggunaan :

```
a = 5
b = 5
c = 6
print('a is b :', a is b)
print('a is c :', a is c)
print('a is not c :', a is not c)
```

Hasil yang didapatkan :

```
a is b : True
a is c : False
a is not c : True

Process finished with exit code 0
```

E. Alat/Instrument/Aparatus/Bahan

1. Personal Computer (PC)
2. Aplikasi PyCharm


F. Keselamatan Kerja

1. Menggunakan alat dan bahan sesuai dengan fungsinya
2. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar.
3. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen.
4. Tidak bercanda atau makan dan minum saat praktik
5. Rapihan dan kembalikan alat dan bahan sesuai tempatnya

G. Lembar Kerja

1. Penggunaan Operator Logika AND

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			Hal 11 dari 141

```

skp = input("Jumlah SKP: ")
sks = input("jumlah SKS: ")

if int(skp) >= 100 and int(sks) >= 75:
    print("Anda boleh Kerja Praktek")
else:
    print("Anda belum bisa Kerja Praktek")

```

Buat file baru dengan nama “logika.py” dengan cara klik *file > new* dan pilih *Python File*. Salin kode diatas dan jalankan program dengan menggunakan *ctrl+shift+f10*, kemudian amati hasilnya. Program diatas menerima dua inputan skp dan sks, dimana jika kondisi variabel skp lebih besar sama dengan 100 dan variabel sks lebih besar sama dengan 75 maka akan menampilkan pernyataan.

2. Penggunaan Operator Perbandingan

```

suhu = input("Masukan suhu: ")
if int(suhu) > 30:
    print("Hari ini panas")
else:
    print("Hari ini tidak panas")


```

Buat file baru dengan nama “perbandingan.py” dengan cara klik *file > new* dan pilih *Python File*. Salin kode diatas dan jalankan program dengan menggunakan *ctrl+shift+f10*, kemudian amati hasilnya dan pahami logikanya. Program diatas menerima inputan yang nantinya disimpan pada variabel suhu. Kemudian jika nilai yang diterima lebih dari 30 maka akan menampilkan pernyataan “Hari ini panas”.

H. Tugas Praktikum

1. Buatlah program yang menerima masukan panjang dan lebar sebuah persegi panjang, dan akan mengeluarkan nilai luas segitiga tersebut!

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Operator Logika	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			Hal 12 dari 141

```
Panjang : 20
Lebar : 5
Hasil adalah 50 cm

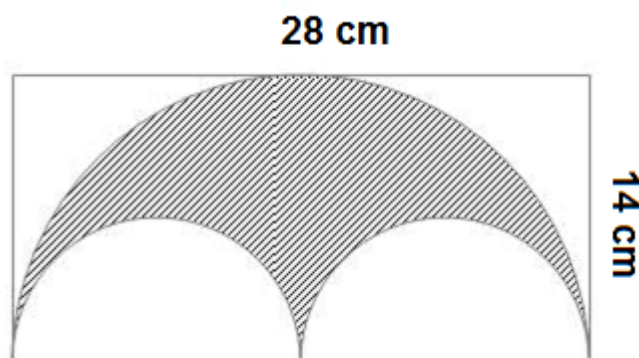
Process finished with exit code 0
```

2. Buatlah program yang menerima masukan Nama, Alamat, Tahun Lahir, dan tampilkan umur dari data yang diinput !

```
Nama : adis
Alamat : denpasar
Tahun Kelahiran : 2000
Nama : adis
Alamat : denpasar
Tahun Kelahiran : 2000
Umur : 20

Process finished with exit code 0
```


3. Hitunglah luas daerah yang diarsir, selesaikan permasalahan berikut dengan menggunakan operator aritmatika pada Python !



Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

PETUNJUK PRAKTIKUM #5

STRING PADA PYTHON

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 5	String pada Python	
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			100 menit
			Hal 1 dari 141

A. Capaian Pembelajaran Mata Kuliah

1. Mampu menerapkan prinsip-prinsip rekayasa untuk perancangan perangkat keras dan perangkat lunak.
2. Mampu menerapkan prinsip perancangan untuk menyelesaikan masalah rekayasa kompleks pada perangkat keras dan perangkat lunak dengan memperhatikan faktor-faktor ekonomi, kesehatan, keselamatan publik, lingkungan, sosial, dan budaya.

B. Sub Capaian Pembelajaran Mata Kuliah

1. Mahasiswa mampu memahami konsep tipe data string dalam suatu kasus.
2. Mahasiswa memahami jenis-jenis operator yang ada dalam tipe data string.

C. Tujuan

Mahasiswa **mengetahui** perintah dasar yang digunakan dalam tipe data string dan **menerapkan** operator string dalam bahasa pemrograman.

D. Dasar Teori

String pada Python adalah tipe data yang memuat satu karakter atau lebih karakter (*sequences of character*) yang diapit oleh tanda petik tunggal (') atau tanda petik ganda (") atau petik ganda tiga kali ("""). Dalam bahasa pemrograman Python, deklarasi suatu string tidak dibedakan penggunaan tanda petik tunggal atau tanda petik ganda. Python memperlakukan tanda petik tunggal sama dengan tanda petik ganda. Membuat string semudah memberi nilai pada sebuah variabel.

Dibawah ini adalah contoh sederhana dari sebuah string pada bahasa pemrograman.

```
print("Hello World")
```

Penulisan dengan menggunakan tanda kutip dua terkadang berguna jika hendak menuliskan kata – kata seperti pada contoh berikut ini:

```
s = "Yoga's Belajar  
Python"
```

Kita juga dapat melakukan penggabungan antara dua buah kata pada data *string* seperti pada contoh berikut ini:

```
s = "Belajar" + "  
Python"
```

Operator perkalian juga dapat digunakan untuk melakukan pengulangan data *string* seperti pada contoh berikut ini :

```
s = "Belajar " * 3  
print(s)
```

Fungsi **len** dapat digunakan untuk menghitung panjang suatu data *string*, atau dalam konteks ini merupakan jumlah karakter yang ditulis termasuk spasi. Contohnya:

```
s = len("Belajar  
Python")
```

Slice digunakan untuk mengambil nilai substring dari sebuah tipe data string. Di python slice didefinisikan dengan tanda kurung siku []. Berikut adalah contoh codingnya:

1. **[:]** akan mengekstrak seluruh sekuensi string dari awal sampai akhir.
2. **[start:]** akan mengekstrak sekuensi string mulai pada index start hingga akhir, dimana nilai juga bernilai negative. Jika start bernilai negative maka index dimulai dari yang paling akhir.
3. **[start:end]** akan mengekstrak string mulai pada index start hingga index end dikurangi 1.

```
name = 'belajar python'  
  
print ("name[0]: ", name[0])  
print ("name[0:3]: ", name[0:3])  
print ("name[1:4]: ", name[1:4])
```

1. Mengupdate String

Anda dapat "memperbarui" string yang ada dengan (kembali) menugaskan variabel ke string lain. Nilai baru dapat dikaitkan dengan nilai sebelumnya atau ke string yang sama sekali berbeda sama sekali. Sebagai contoh :

```
message = 'Hello World'
print ("Updated String :- ", message[:6] +
      'Python')
```

2. Escape Character

Dibawah ini adalah tabel dari daftar karakter escape atau karakter non-printable yang dapat diwakili/ditulis dengan awalan notasi backslash.

Notasi Backslash	Karakter Hexadecimal	Penjelasan
<code>\a</code>	<code>0x07</code>	Bell atau alert
<code>\b</code>	<code>0x08</code>	Backspace
<code>\cx</code>		Control-x
<code>\C-x</code>		Control-x
<code>\e</code>	<code>0x1b</code>	Escape
<code>\f</code>	<code>0x0c</code>	Formfeed
<code>\M-\C-x</code>		Meta-Control-x

<code>\n</code>	<code>0x0a</code>	Newline
<code>\nnn</code>		Octal notation, dimana n berada di range 0-7
<code>\r</code>	<code>0x0d</code>	Carriage return
<code>\s</code>	<code>0x20</code>	Space
<code>\t</code>	<code>0x09</code>	Tab
<code>\v</code>	<code>0x0b</code>	Vertical tab
<code>\x</code>		Character x
<code>\xnn</code>	<code>asda f s d f s d f</code>	Notasi Hexadecimal, dimana n berada di range 0-9, a-f, atau A-F

3. Operator Special String

Asumsikan variabel a adalah 'Belajar' dan variabel b adalah 'Python', lalu dibawah ini adalah operator yang bisa dipakai pada kedua string di variabel tersebut.

a = "Belajar"

b = "Python"

Berikut adalah daftar operator spesial string pada Python :

Operator	Contoh	Penjelasan
+	a + b akan menghasilkan BelajarPython	Concatenation - Menambahkan nilai pada kedua sisi operator
*	a*2 akan menghasilkan BelajarBelajar	Pengulangan - Membuat string baru, menggabungkan beberapa salinan dari string yang sama
[]	a[1] akan menghasilkan e	Slice - Memberikan karakter dari indeks yang diberikan
[:]	a[1:4] akan menghasilkan ela	Range Slice - Memberikan karakter dari kisaran yang diberikan
in	B in a akan menghasilkan 1	Keanggotaan - Mengembalikan nilai true jika ada karakter dalam string yang diberikan
not in	Z not in a akan menghasilkan 1	Keanggotaan - Mengembalikan nilai true jika karakter tidak ada dalam string yang diberikan
r/R	print r'\n' prints \n dan print R'\n'prints \n	Raw String - Menekan arti aktual karakter Escape. Sintaks untuk string mentah sama persis dengan string biasa kecuali operator string mentah, huruf "r", yang mendahului tanda petik. "R" bisa berupa huruf kecil (r) atau huruf besar (R) dan harus ditempatkan tepat sebelum tanda kutip pertama.
%		Format - Melakukan format String

4. Operator Format String

Salah satu fitur yang ada pada Python adalah format string operator %. Operator ini unik untuk string dan membuat paket memiliki fungsi dari keluarga printf C () C.

Berikut adalah daftar lengkap simbol yang bisa digunakan bersamaan dengan % :

Operator	Penjelasan
%c	character
%s	Konversi string melalui str () sebelum memformat
%i	Dianggap sebagai bilangan bulat desimal
%d	Dianggap sebagai bilangan bulat desimal
%u	Unsigned decimal integer
%o	Bilangan bulat oktal
%x	Bilangan bulat heksadesimal (huruf kecil)
%X	Bilangan bulat heksadesimal (huruf besar)
%e	Notasi eksponensial (dengan huruf kecil 'e')
%E	Notasi eksponensial (dengan huruf besar 'E')
%f	Bilangan real floating point
%g	Yang lebih pendek dari% f dan% e
%G	Lebih pendek dari% f dan% E

5. Triple Quote (Tanda Petik tiga kali)

Tanda kutip dua tiga kali juga dapat digunakan dengan membiarkan string untuk ditulis dalam beberapa baris, termasuk kata kerja Newlines, Tabs, dan karakter khusus lainnya.. Tanda ini juga berguna ketika hendak menuliskan kalimat panjang yang terbagi dalam beberapa baris. Berikut adalah contohnya:

```
s = """Dalam bahasa pemrograman \t Python
deklarasi suatu string tidak dibedakan penggunaan tanda petik
tunggal atau tanda petik ganda. \n
Python memperlakukan tanda petik tunggal sama dengan tanda petik
ganda.
Membuat string semudah memberi nilai pada sebuah variabel."""
print(s)
```

6. String Unicode

Pada Python 3, semua string diwakili dalam Unicode. Sedangkan pada Python 2 disimpan secara internal sebagai 8-bit ASCII, maka diperlukan lampiran 'u' untuk membuatnya menjadi Unicode. Tetapi hal ini tidak lagi diperlukan sekarang.

Python menyertakan metode built-in untuk memanipulasi string. berikut adalah metodenya.

Metode	Penjelasan
<code>capitalize()</code>	Meng-kapitalkan huruf pertama string
<code>center(width, fillchar)</code>	Mengembalikan string yang dilapisi dengan fillchar dengan string asli yang dipusatkan pada total width kolom.
<code>count(str, beg = 0, end = len(string))</code>	Menghitung berapa kali str yang terjadi dalam string atau dalam substring string jika memulai indeks beg dan end index end diberikan.
<code>decode(encoding = 'UTF-8', errors = 'strict')</code>	Dekode string menggunakan codec yang terdaftar untuk pengkodean. Encoding default ke pengkodean string default.
<code>encode(encoding = 'UTF-8', errors = 'strict')</code>	Mengembalikan versi string yang dikodekan string; Pada kesalahan, default adalah menaikkan ValueError kecuali jika kesalahan diberikan dengan 'ignore' atau 'replace'.
<code>endswith(suffix, beg = 0, end = len(string))</code>	Menentukan apakah string atau substring string (jika memulai indeks memohon dan mengakhiri akhir indeks diberikan) berakhir dengan akhiran; Mengembalikan nilai true jika benar dan salah.

<code>expandtabs(tabsize = 8)</code>	Memperluas tab dalam string ke banyak ruang; Default ke 8 spasi per tab jika tabsize tidak tersedia.
<code>find(str, beg = 0 end = len(string))</code>	Tentukan jika str terjadi dalam string atau dalam substring string jika memulai indeks beg dan end index end diberikan return index jika ditemukan dan -1 sebaliknya.
<code>index(str, beg = 0, end = len(string))</code>	Sama seperti find (), namun menimbulkan pengecualian jika str tidak ditemukan.
<code>isalnum()</code>	Mengembalikan true jika string memiliki minimal 1 karakter dan semua karakternya alfanumerik dan false sebaliknya.
<code>isalpha()</code>	Mengembalikan true jika string memiliki minimal 1 karakter dan semua karakter adalah abjad dan false sebaliknya.
<code>isdigit()</code>	Mengembalikan true jika string hanya berisi digit dan false sebaliknya.
<code>islower()</code>	Mengembalikan true jika string memiliki setidaknya 1 karakter casing dan semua karakter casing dalam huruf kecil dan false sebaliknya.
<code>isnumeric()</code>	Mengembalikan true jika string Unicode hanya berisi karakter numerik dan false sebaliknya.
<code>isspace()</code>	Mengembalikan true jika string hanya berisi karakterspasi dan false sebaliknya.
<code>istitle()</code>	Mengembalikan true jika string benar "titlecased" dan false sebaliknya.
<code>isupper()</code>	Mengembalikan true jika string memiliki setidaknya satu karakter casing dan semua karakter casing ada dalam huruf besar dan false sebaliknya.

<code>join(seq)</code>	Merges (concatenates) representasi string elemen dalam urutan seq menjadi string, dengan string pemisah.
<code>len(string)</code>	Mengembalikan panjang string
<code>ljust(width[, fillchar])</code>	Mengembalikan string berlapis ruang dengan string asli dibiarkan dibenarkan ke kolom lebar total.
<code>lower()</code>	Mengonversi semua huruf besar dalam bentuk string menjadi huruf kecil.
<code>lstrip()</code>	Menghapus semua spasi utama dalam string.
<code>maketrans()</code>	Mengembalikan tabel terjemahan untuk digunakan dalam fungsi terjemahan.
<code>max(str)</code>	Mengembalikan karakter alfabetik dari string str.
<code>min(str)</code>	Mengembalikan min karakter abjad dari string str.
<code>replace(old, new [, max])</code>	Menggantikan semua kemunculan lama dalam string dengan kejadian baru atau paling maksimal jika max diberikan.
<code>rfind(str, beg = 0, end = len(string))</code>	Sama seperti find (), tapi cari mundur dalam string.
<code>rindex(str, beg = 0, end = len(string))</code>	Sama seperti index (), tapi cari mundur dalam string.
<code>rjust(width, [, fillchar])</code>	Mengembalikan string berlapis ruang dengan senar asli benar-dibenarkan untuk total kolom lebar.
<code>rstrip()</code>	Menghapus semua spasi spasi string.
<code>split(str="", num=string.count(str))</code>	Membagi string sesuai dengan pemisah str (ruang jika tidak disediakan) dan mengembalikan daftar substring;

	Terpecah menjadi paling banyak substring jika diberikan.
<code>splitlines(num=string.count('\n'))</code>	Membagi string sama sekali (atau num) NEWLINEs dan mengembalikan daftar setiap baris dengan NEWLINEs dihapus.
<code>startswith(str, beg=0,end=len(string))</code>	Determines if string or a substring of string (if starting index beg and ending index end are given) starts with substring str; returns true if so and false otherwise.
<code>strip([chars])</code>	Lakukan kedua lstrip () danrstrip () pada string
<code>swapcase()</code>	Kasus invers untuk semua huruf dalam string.
<code>title()</code>	Mengembalikan versi string "titlecased", yaitu, semua kata diawali dengan huruf besar dan sisanya huruf kecil.
<code>translate(table, deletchars="")</code>	Menerjemahkan string sesuai dengan tabel terjemahan str (256 karakter), menghapus string del.
<code>upper()</code>	Mengonversi huruf kecil dalam bentuk string ke huruf besar.
<code>zfill (width)</code>	Mengembalikan string asli yang tertinggal dengan angka nol ke total karakter lebar; Dimaksudkan untuk angka, zfill () mempertahankan tanda apapun yang diberikan (kurang satu nol).
<code>isdecimal()</code>	Mengembalikan nilai true jika string unicode hanya berisi karakter desimal dan false sebaliknya.

E. Alat/Instrument/Aparatus/Bahan

1. Personal Computer (PC)

2. Aplikasi PyCharm

F. Keselamatan Kerja

1. Menggunakan alat dan bahan sesuai dengan fungsinya
2. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar.
3. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen.
4. Tidak bercanda atau makan dan minum saat praktik
5. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

G. Lembar Kerja

Kegiatan Praktikum 1

Berikut adalah contoh program menggunakan oprator penggabungan +, pengulangan *, Mengambil karakter dari sebuah string [], dan Mengambil karakter dari sebuah string berdasarkan range (substring) [:].

```
var1 = 'Belajar Python!'
var2 = ' Sangat menyenangkan!'

#operator concatenation
print ("Hasil penggabungan : ", var1 + var2)

#Operator repetition
print ("Hasil pengulangan : ", var1 * 3)

#Operator []
print ("Hasil : ", var1[3])

#Operator [:]
print ("Hasil : ", var1[0:8])
```

Jika program diatas dijalankan, maka akan menampilkan output sebagai berikut :

```
Hasil penggabungan : Belajar Python! Sangat menyenangkan!
Hasil repetisi : Belajar Python!Belajar Python!Belajar Python!
Hasil : a
Hasil : Belajar

Process finished with exit code 0
```

Kegiatan Praktikum 2

Untuk memformat sebuah string dapat menggunakan tanda persen "%". Di bawah ini adalah contoh program untuk memformat sebuah string.

```
print ("Perkenalkan nama saya %s berumur %d tahun!" %  
('Python', 20))
```

Jika program diatas jalankan, maka akan menampilkan output sebagai berikut :

```
Perkenalkan nama saya Python berumur 20 tahun!  
  
Process finished with exit code 0
```

Kegiatan Praktikum 3

Berikut adalah contoh program menggunakan tanda petik ganda tiga kali dengan membiarkan string untuk ditulis dalam beberapa baris, termasuk kata kerja NEWLINES, TABs, dan karakter khusus lainnya.. Tanda ini juga berguna ketika hendak menuliskan kalimat panjang yang terbagi dalam beberapa baris. Berikut adalah contohnya:

```
s = """Dalam bahasa pemrograman Python  
deklarasi suatu string tidak dibedakan penggunaan  
tanda petik tunggal atau tanda petik ganda. \n  
Python \t memperlakukan tanda petik tunggal sama  
dengan tanda petik ganda.  
Membuat string semudah memberi nilai pada sebuah  
variabel."""  
print(s)
```

Jika program diatas jalankan, maka akan menampilkan output sebagai berikut :

```
Dalam bahasa pemrograman Python
deklarasi suatu string tidak dibedakan penggunaan tanda petik tunggal atau tanda petik ganda.

Python memperlakukan tanda petik tunggal sama dengan tanda petik ganda.
Membuat string semudah memberi nilai pada sebuah variabel.

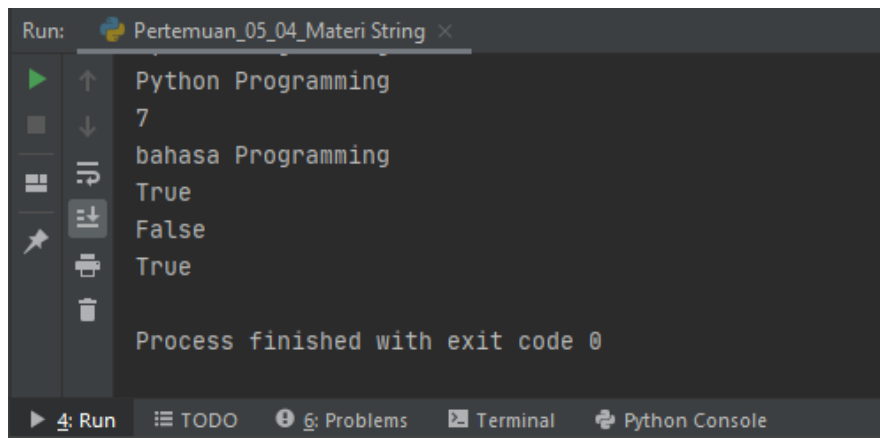
Process finished with exit code 0
```

Kegiatan Praktikum 4

Berikut ini adalah contoh program menggunakan beberapa string method.

```
course = "Python Programming"
print(course.upper())
print(course.lower())
print(course.title())
print(course.rstrip())
print(course.find("Pro"))
print(course.replace("Python", "bahasa"))
print("Pro" in course)
print("Pro" not in course)
print("Swift" not in course)
```

Jika program diatas dijalankan, maka akan menampilkan output sebagai berikut :

A screenshot of a Python IDE window titled 'Run: Pertemuan_05_04_Materi String'. The output console shows the results of the program: 'Python Programming', '7', 'bahasa Programming', 'True', 'False', 'True', and 'Process finished with exit code 0'. The IDE interface includes a sidebar with icons for Run, TODO, Problems, Terminal, and Python Console.

```
Run: Pertemuan_05_04_Materi String ×
Python Programming
7
bahasa Programming
True
False
True
Process finished with exit code 0
```

H. Tugas Praktikum


1. Buatlah program dimana di dalam suatu kelas terdapat 4 anak yang mempunyai nilai matakuliah yang berbeda-beda. Gunakan metode f-string untuk membuat program tersebut. Dengan ketentuan nilai seperti berikut :

toni = 80
rani = 78
jaka = 100
diah = 65

```
nilai toni = 80 , nilai rani = 78 , nilai jaka = 100 dan nilai diah = 65
```

2. Buatlah program untuk menghitung jumlah huruf dengan ketentuan seperti gambar di bawah ini.


```
-----belajar bahasa pemrograman python-----  
berapa jumlah huruf a pada tulisan diatas : 7 huruf ditemukan
```

 INSTIKI <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 5	Fungsi Matematika pada Python	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 0 dari 141

PETUNJUK PRAKTIKUM #6

FUNGSI MATEMATIKA PADA PYTHON

Dilarang memperbanyak sebagian atau seluruh isi dokumen tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	tanpa ijin	Diperiksa oleh :
--	------------	------------------

 INSTIKI <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 5	Fungsi Matematika pada Python	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 1 dari 141

A. Capaian Pembelajaran Mata Kuliah

1. Mampu menerapkan algoritma/metode dalam perangkat lunak berbasis komputer
2. Mampu menerapkan konsep teoritis dalam bidang algoritma.

B. Sub Capaian Pembelajaran Mata Kuliah

Mahasiswa mampu memahami fungsi matematika.


C. Tujuan

1. Mahasiswa mampu **memahami** kegunaan fungsi matematika.
2. Mahasiswa mampu **menggunakan** fungsi matematika dengan tepat.
3. Mahasiswa mampu **mengaplikasikan** fungsi matematika ke dalam bahasa pemrograman.

D. Dasar Teori

Fungsi matematika adalah sebuah modul yang menyediakan akses ke fungsi matematika yang ditentukan oleh standar C. Fungsi ini tidak dapat digunakan dengan bilangan kompleks. Perbedaan antara fungsi yang mendukung bilangan kompleks dan yang tidak adalah sebagian besar pengguna tidak ingin menggunakan fungsi matematika sebanyak yang diperlukan untuk memahami bilangan kompleks. Maka, fungsi matematika digunakan untuk memungkinkan mendeteksi bilangan kompleks yang tak terduga yang digunakan sebagai ukuran, sehingga pemrograman dapat menentukan bagaimana dan mengapa itu dibuat. Semua nilai yang dikembalikan adalah float. Berikut adalah beberapa fungsi yang disediakan oleh modul ini:

Nama	Penggunaan	Penjelasan
Absolute	abs(x)	Nilai absolut dari x:(positive) jarak antara x and 0.
Ceiling	ceil(x)	Ceiling dari x: integer terkecil yang kurang dari x.

 INSTIKI <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 5	Fungsi Matematika pada Python	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 2 dari 141

Cmp	cmp(x, y)	-1 if $x < y$, 0 if $x == y$, or 1 if $x > y$. Tidak berlaku lagi dengan Python 3. Sebaliknya gunakan $\text{return } (x > y) - (x < y)$
Eksponen	exp(x)	Nilai eksponen dari x: e^x
Nama	Penggunaan	Penjelasan
Fabs	fabs(x)	Nilai absolut dari x.
Floor	floor(x)	Nilai dasar dari x: internet terbesar tidak lebih besar dari x.
Log	log(x)	Logaritma dari x, untuk $x > 0$.
Log 10	log10(x)	Basis 10 logaritma dari x, untuk $x > 0$.
Max	max(x1, x2,...)	Argumen terbesar: Nilai terdekat dengan tak terhingga positif
Min	min(x1, x2,...)	Argumen terkecil: nilai yang paling mendekati tak berhingga negatif.
Modf	modf(x)	Bagian pecahan dan bilangan bulat dari x dalam tupel dua item. Kedua bagian memiliki tanda yang sama dengan x. Bagian integer dikembalikan sebagai float.
Pow	pow(x, y)	Nilai $x ** y$.
Round	round(x [,n])	X dibulatkan menjadi n digit dari titik desimal. Putaran Python jauh dari nol sebagai tie-breaker: $\text{round}(0.5)$ adalah 1.0 dan $\text{round}(-0.5)$ adalah -1.0.
Akar Kuadrat	sqrt(x)	Akar kuadrat x untuk $x > 0$.

Dilarang memperbanyak sebagian atau seluruh isi dokumen tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	tanpa ijin	Diperiksa oleh :
--	------------	------------------

 INSTIKI <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 5	Fungsi Matematika pada Python	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 3 dari 141

Fungsi Nomor Acak

Nomor acak digunakan untuk aplikasi permainan, simulasi, pengujian, keamanan, dan privasi. Python mencakup fungsi berikut yang umum digunakan. Berikut adalah daftarnya :

Nama	Penggunaan	Penjelasan
Choice	choice(seq)	Item acak dari list, tuple, atau string.
RandRange	randrange ([start,] stop [,step])	Elemen yang dipilih secara acak dari jangkauan (start, stop, step).
Random	random()	A random float r, sehingga 0 kurang dari atau sama dengan r dan r kurang dari 1
Seed	seed([x])	Menetapkan nilai awal integer yang digunakan dalam menghasilkan bilangan acak. Panggil fungsi ini sebelum memanggil fungsi modul acak lainnya. Tidak ada pengembalian
Shuffle	shuffle(lst)	Mengacak daftar dari daftar di tempat. Tidak ada pengembalian
Floor	floor(x)	The floor of x: the largest

Dilarang memperbanyak sebagian atau seluruh isi dokumen tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	tanpa ijin	Diperiksa oleh :
--	------------	------------------

 INSTIKI <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 5	Fungsi Matematika pada Python	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 4 dari 141


		integer not greater than x.
Uniform	uniform(x, y)	Sebuah float acak r, sedemikian rupa sehingga x kurang dari atau sama dengan r dan r kurang dari y.

Fungsi Trigonometri

Python mencakup fungsi berikut yang melakukan perhitungan trigonometri. Berikut adalah daftarnya :

Nama	Penggunaan	Penjelasan
Acos	acos(x)	Kembalikan kosinus x, di radian.
Asin	asin(x)	Kembalikan busur sinus x, dalam radian.
Atan	atan(x)	Kembalikan busur singgung x, di radian.
Atan 2	atan2(y, x)	Kembali atan (y / x), di radian.
Kosinus	cos(x)	Kembalikan kosinus x radian.
Hypot	hypot(x, y)	Kembalikan norma Euclidean, sqrt (x * x + y * y).
Sin	sin(x)	Kembalikan sinus dari x radian.
Tan	tan(x)	Kembalikan tangen x radian.
Derajat	degrees(x)	Mengonversi sudut x dari radian ke derajat.
Radian	radians(x)	Mengonversi sudut x dari derajat ke radian.

Dilarang memperbanyak sebagian atau seluruh isi dokumen tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	tanpa ijin	Diperiksa oleh :
--	------------	------------------

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 5	Fungsi Matematika pada Python	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021
			Hal 5 dari 141

Konstanta Matematika

Modul ini juga mendefinisikan dua konstanta matematika. Berikut adalah daftarnya:

Nama	Penggunaan	Penjelasan
Pi	pi	Konstanta Pi matematika
e	e	Konstanta e matematika

E. Alat/Instrument/Aparatus/Bahan

1. Personal Computer (PC)
2. Aplikasi PyCharm

F. Keselamatan Kerja

1. Menggunakan alat dan bahan sesuai dengan fungsinya
2. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar.
3. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen.
4. Tidak bercanda atau makan dan minum saat praktik
5. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

G. Lembar Kerja


a) Fungsi abs(x)

1. Fungsi abs(x) digunakan untuk menampilkan nilai absolute dari sebuah bilangan x.

Salinlah kodingan dibawah ini

```
print("abs(-50) : ", abs(-50))
print("abs(10.5) : ", abs(-10.5))
print("abs(100) : ", abs(100))
```

Dilarang memperbanyak sebagian atau seluruh isi dokumen tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	tanpa ijin	Diperiksa oleh :
--	------------	------------------

 INSTIKI <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 5	Fungsi Matematika pada Python	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021
			Hal 6 dari 141

2. Jalankan program diatas, maka akan menghasilkan :

```
abs(-50) : 50
abs(10.5) : 10.5
abs(100) : 100

Process finished with exit code 0
```

b) Fungsi ceil(x)

1. Fungsi abs(x) digunakan untuk membulatkan kebawah sebuah bilangan x.

Salinlah kodingan dibawah ini

```
import math


print("math.ceil(-50.25) : ", math.ceil(-50.25))
print("math.ceil(100.12) : ", math.ceil(100.12))
print("math.ceil(30.87) : ", math.ceil(30.87))
print("math.ceil(80) : ", math.ceil(80))
print("math.ceil(73) : ", math.ceil(73))
```

2. Jalankan program diatas, maka akan menghasilkan :

```
math.ceil(-50.25) : -50
math.ceil(100.12) : 101
math.ceil(30.87) : 31
math.ceil(80) : 80
math.ceil(73) : 73

Process finished with exit code 0
```

Dilarang memperbanyak sebagian atau seluruh isi dokumen tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	tanpa ijin	Diperiksa oleh :
--	------------	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 5	Fungsi Matematika pada Python	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 7 dari 141

c) Fungsi pow(x,y)

1. Fungsi pow(x,y) digunakan untuk menampilkan perpangkatan bilangan x dan membutuhkan parameter.

Salinlah kodingan dibawah ini

```
import math


print("5 pangkat 2 : ", math.pow(5, 2))
print("7 pangkat 4 : ", math.pow(7, 4))
print("8 pangkat 8 : ", math.pow(8, 8))
print("4 pangkat 2 : ", math.pow(4, 2))
print("10 pangkat 4 : ", math.pow(10, 4))
```

2. Jalankan program diatas, maka akan menghasilkan :

```
5 pangkat 2 : 25.0
7 pangkat 4 : 2401.0
8 pangkat 8 : 16777216.0
4 pangkat 2 : 16.0
10 pangkat 4 : 10000.0

Process finished with exit code 0
```

Dilarang memperbanyak sebagian atau seluruh isi dokumen tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	tanpa ijin	Diperiksa oleh :
--	------------	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 5	Fungsi Matematika pada Python	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 8 dari 141

d) Fungsi sqrt(x)

1. Fungsi abs(x) digunakan untuk menghitung akar kuadrat dari suatu bilangan x. Salinlah kodingan dibawah ini

```
import math

print("akar 0 : ", math.sqrt(0))
print("akar 1 : ", math.sqrt(1))
print("akar 4 : ", math.sqrt(4))
print("akar 9 : ", math.sqrt(9))
print("akar 36 : ", math.sqrt(36))
```

2. Jalankan program diatas, maka akan menghasilkan :

```
akar 0 : 0.0
akar 1 : 1.0
akar 4 : 2.0
akar 9 : 3.0
akar 36 : 6.0

Process finished with exit code 0
```


e) Fungsi round(x)

1. Fungsi abs(x) digunakan untuk membulatkan sebuah bilangan x

Salin

```
print("10.5 : ", round(10.5))
print("5.478947 : ", round(5.478947))
print("73.7387483 : ", round(73.7387483))
print("3.673673 : ", round(3.673673))
print("437.378437 : ", round(437.378437))
```

Dilarang memperbanyak sebagian atau seluruh isi dokumen tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	tanpa ijin	Diperiksa oleh :
--	------------	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 5	Fungsi Matematika pada Python	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 9 dari 141

2. Jalankan program diatas, maka akan menghasilkan :

```
10.5 : 10
5.478947 : 5
73.7387483 : 74
3.673673 : 4
437.378437 : 437

Process finished with exit code 0
```


H. Tugas Praktikum

- 34, 20, 67, 91, 29, 42, 77, 39, 84, 43, 94.
Dari bilangan diatas berturut-turut, hitunglah nilai Max dan Min menggunakan fungsi matematika!
- $a = 7$
 $b = 5$
Buatlah program menghitung rumus pitagoras menggunakan fungsi matematika!

Dilarang memperbanyak sebagian atau seluruh isi dokumen tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	tanpa ijin	Diperiksa oleh :
--	------------	------------------

PETUNJUK PRAKTIKUM #7

TEKNIK PERCABANGAN

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 6	Percabangan	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021
			Hal 1 dari 141

A. Capaian Pembelajaran Mata Kuliah

1. Mampu mendefinisikan kebutuhan pengguna atau pasar terhadap kinerja (menganalisis, mengevaluasi dan mengembangkan) algoritma/metode berbasis komputer.
2. Mampu menelaah cara kerja sistem komputer dan mampu merancang dan mengembangkan berbagai algoritma/metode untuk memecahkan masalah.

B. Sub Capaian Pembelajaran Mata Kuliah

Mahasiswa mampu menyelesaikan permasalahan yang berkaitan dengan teknik percabangan pada kasus sederhana.

C. Tujuan

mahasiswa mampu **memecahkan** permasalahan yang terkait dengan percabangan.


D. Dasar Teori

Percabangan *if* merupakan sebuah blok program yang menyatakan bahwa sebuah aksi akan dijalankan jika kondisi percabangan dipenuhi jika tidak dipenuhi maka aksi tidak akan dijalankan. Percabangan *if* biasa digunakan untuk mengerjakan aksi yang memiliki syarat tertentu untuk menjalankannya. Pernyataan *if* diklasifikasikan lagi ke dalam empat bagian, yaitu :

1. Pernyataan *if* dengan satu kondisi (*if tunggal*)
2. Pernyataan *if* dengan dua kondisi (*if - else*)
3. Pernyataan *if* dengan dua kondisi atau lebih (*if - elif*)
4. Pernyataan *if* bersarang (*if* di dalam *if*)

Pernyataan *If tunggal* hanya melibatkan satu kondisi yang akan diperiksa. Apabila kondisi yang diperiksa bernilai benar, maka program akan mengeksekusi bagian yang berada dalam blok. Jika sebaliknya, maka program akan mengabaikan pernyataan di dalam blok dan langsung melanjutkan eksekusi berikutnya.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 INSTIKI <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Percabangan	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 2 dari 141

Bentuk Umum :

```
{terdiri dari satu statemen}
if (kondisi):
    pernyataan

{terdiri atas beberapa statemen}
if (kondisi):
    {
        pernyataan1
        pernyataan2
        ...
    }
```

Contoh penggunaan :


```
panas = True
if panas:
    print("Hari ini panas")
    print("Perbanyak minum air")
print("Nikmati harimu :)")
```

Hasil yang didapat :

```
Hari ini panas
Perbanyak minum air
Nikmati harimu :)|
Process finished with exit code 0
```

Jika panas bernilai False

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 INSTIKI <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Percabangan	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 3 dari 141


```
Nikmati harimu :)

Process finished with exit code 0
```

Pernyataan *if* dengan dua kondisi (*if - else*) dipergunakan untuk menyatakan pernyataan percabangan dua kondisi dimana ada dua blok aksi yang dipilih untuk dikerjakan jika syarat kondisi aksi terpenuhi. Saat pembacaan program sampai pada blok *if-else* maka akan dilakukan pemeriksaan terhadap syarat kondisi percabangan yang ada pada deklarasi *if*, jika syarat dipenuhi maka yang akan dijalankan adalah aksi yang ada di dalam blok *if*, tapi jika syarat tidak dipenuhi maka aksi yang dikerjakan adalah yang ada di dalam blok *else*.

Bentuk Umum :

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 INSTIKI <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Percabangan	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			Hal 4 dari 141

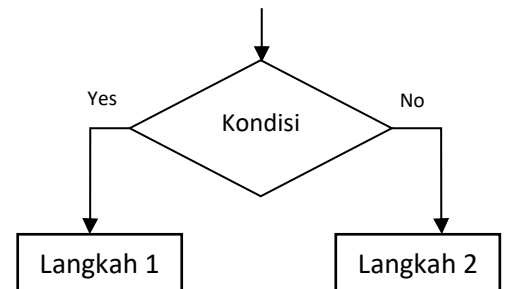
```

if (kondisi):
    pernyataan1
else:
    pernyataan2

atau


if (kondisi):
    {
        pernyataan1a
        pernyataan1b
        ...
    }
else:
    {
        pernyataan2a
        pernyataan2b
        ...
    }

```



Contoh penggunaan :

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 INSTIKI <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Percabangan	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 5 dari 141

```

panas = False
if panas:
    print("Hari ini panas")
    print("Perbanyak minum air")
else:
    print("Hari ini dingin")
    print("Gunakan pakaian hangat")
print("Nikmati harimu :)")

```

Hasil yang didapatkan :

```

Hari ini dingin
Gunakan pakaian hangat
Nikmati harimu :)


Process finished with exit code 0

```

Pengambilan keputusan (kondisi if elif) merupakan lanjutan/percabangan logika dari "kondisi if". Dengan elif kita bisa membuat kode program yang akan menyeleksi beberapa kemungkinan yang bisa terjadi. Hampir sama dengan kondisi "else", bedanya kondisi "elif" bisa banyak dan tidak hanya satu.

Bentuk umum :

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Percabangan	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 6 dari 141

```

if (kondisi):
    pernyataan1
elif (kondisi):
    pernyataan2
elif (kondisi):
    pernyataan3
elif (kondisi):
    pernyataan4

```

Contoh penggunaan :


```

hari_ini = "Minggu"
if(hari_ini == "Senin"):
    print("Saya akan kuliah")
elif(hari_ini == "Selasa"):
    print("Saya akan kuliah")
elif(hari_ini == "Rabu"):
    print("Saya akan kuliah")
elif(hari_ini == "Kamis"):
    print("Saya akan kuliah")
elif(hari_ini == "Jumat"):
    print("Saya akan kuliah")
elif(hari_ini == "Sabtu"):
    print("Saya akan kuliah")
elif(hari_ini == "Minggu"):
    print("Saya akan libur")

```

Hasil yang didapatkan :

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Percabangan	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 7 dari 141

```
Saya akan libur

Process finished with exit code 0
```

E. Alat/Instrument/Aparatus/Bahan

1. Personal Computer (PC)
2. Aplikasi PyCharm

F. Keselamatan Kerja

1. Menggunakan alat dan bahan sesuai dengan fungsinya
2. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar.
3. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen.
4. Tidak bercanda atau makan dan minum saat praktik


7. Lembar Kerja

1. Penentuan nilai menggunakan if

```
nilai = int(input("Nilai : "))
if nilai > 7:
    print("Kamu mendapatkan nilai A")
else:
    print("Kamu medapatkan nilai", nilai)
```

2. Buat file baru dengan nama "nilai.py". jalankan program tersebut dengan menekan **Ctrl+Shift+F10**, kemudian amati hasilnya jika diberikan inputan skor yang berbeda dan pahami logika dari program.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Percabangan	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 8 dari 141

2. Menampilkan nama hari berdasarkan input menggunakan elif

```

hari = int(input("Input Angka : "))
if(hari == 1):
    print("Senin")
elif(hari == 2):
    print("Selasa")
elif(hari == 3):
    print("Rabu")
elif(hari == 4):
    print("Kamis")
elif(hari == 5):
    print("Jumat")
elif(hari == 6):
    print("Sabtu")
elif(hari == 7):
    print("Minggu")


```

Buat file baru dengan nama “bulan.py”. jalankan program tersebut dengan menekan *Ctrl+Shift+F10*, kemudian amati hasilnya jika diberikan inputan nilai yang berbeda dan pahami logika dari program.

8. Tugas Praktikum

1. Buatlah program yang memiliki dua masukan yaitu umur dan tinggi badan, dengan ketentuan umur minimum 17 dan tinggi badan minium 165. Jika syarat dipenuhi, maka akan muncul kalimat “Syarat terpenuhi”. Gunakan percabangan if satu kondisi yang disertai dengan logika and!

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Percabangan	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021 Hal 9 dari 141

```

Umur : 22
Tinggi badan : 165
Syarat terpenuhi!

Process finished with exit code 0

```

2. Buatlah program penentuan bilangan genap atau ganjil yang menerima masukan sebuah bilangan. Kemudian menampilkan apakah bilang tersebut adalah bilangan genap atau ganjil menggunakan percabangan if dua kondisi!

```

Masukkan Bilangan : 11
11 adalah bilangan ganjil

Process finished with exit code 0

```

3. Buatlah program kalkulator sederhana yang memiliki tiga masukan yaitu : pilih operasi, bilangan 1 dan bilangan 2. Pada masukan pilih operasi memiliki nilai : 1 sebagai penjumlahan , 2 sebagai pengurangan, 3 sebagai perkalian dan 4 sebagai pembagian. Kemudian tampilkan hasil dari operasi yang dijalankan.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 INSTIKI <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 3	Percabangan	100 menit
	Kode MK: SKW-002	Rev : 00	Tgl : 23/08/2021
			Hal 10 dari 141

```

Pilih Operasi.
1.Jumlah
2.Kurang
3.Kali
4.Bagi
Masukkan pilihan operasi (1/2/3/4): 2
Masukkan bilangan pertama: 3
Masukkan bilangan kedua: 4
3 - 4 = -1

Process finished with exit code 0

```

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

PETUNJUK PRAKTIKUM #8

UTS

- A. Capaian Pembelajaran Mata Kuliah**
- B. Sub Capaian Pembelajaran Mata Kuliah**
- C. Tujuan**
- D. Dasar Teori**
- E. Alat/Instrument/Aparatus/Bahan**
- F. Keselamatan Kerja**
- G. Lembar Kerja**
- H. Tugas Praktikum**


Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

PETUNJUK PRAKTIKUM #9

TEKNIK PERCABANGAN 2

ALGORITMA DAN LOGIKA INFORMATIKA 0

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 4	Teknik Percabangan 2	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021
			Hal 1 dari 141

A. Capaian Pembelajaran Mata Kuliah

1. Mampu mendefinisikan kebutuhan pengguna atau pasar terhadap kinerja (menganalisis, mengevaluasi dan mengembangkan) algoritma/metode berbasis komputer.
2. Mampu menelaah cara kerja sistem komputer dan mampu merancang dan mengembangkan berbagai algoritma/metode untuk memecahkan masalah.

B. Sub Capaian Pembelajaran Mata Kuliah

Mahasiswa mampu menyelesaikan permasalahan yang berkaitan dengan teknik percabangan pada kasus sederhana.

C. Tujuan

Mahasiswa mampu **memecahkan** permasalahan yang terkait dengan percabangan.


A. Dasar Teori

Teknik Percabangan Bersarang

Selain pernyataan *if* dengan dua kondisi, suatu pernyataan *if* dapat mengandung pernyataan *if* yang lain. Bentuk seperti ini biasa disebut *if* bersarang (*nested if*). Sebuah program mengijinkan blok percabangan *if* di dalam blok percabangan lainnya, dan tidak membatasi jenis percabangan apa yang boleh berada di dalam percabangan lainnya.

Bentuk Umum :

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 4	Teknik Percabangan 2	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 2 dari 141

```


if(kondisi1):
    if(kondisi2):
        .
        .
    if(kondisi n):

    else:
        pernyataan
        .
        .
    else:
        pernyataan
else:
    pernyataan

```

Contoh Penggunaan

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 4	Teknik Percabangan 2	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 3 dari 141

```

gaji = input("Input Gaji : ")
berkeluarga = True
punya_rumah = True

if int(gaji) > 2700000:
    print("Gaji sudah diatas UMR")
    if berkeluarga:
        print("Wajib ikut asuransi dan menabung untuk pensiun")
    else:
        print("Tidak perlu ikut asuransi")
    if punya_rumah:
        print("Wajib bayar pajak rumah")
    else:
        print("Tidak wajib bayar pajak rumah")
else:
    print("Gaji belum UMR")

```

Hasil yang didapatkan :

```


Input Gaji : 2000000
Gaji sudah diatas UMR
Wajib ikut asuransi dan menabung untuk pensiun
Wajib bayar pajak rumah

Process finished with exit code 0

```

Pernyataan *switch case* merupakan alternatif dari pernyataan *if* untuk masalah dengan pilihan berganda. Pada masalah tertentu, *switch case* lebih memberikan kejelasan dari pada *if*. Namun perlu diketahui bahwa semua persoalan yang dapat ditangani *switch case*

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 4	Teknik Percabangan 2	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 4 dari 141

bisa ditangani oleh *if*, tetapi tidak sebaliknya. *Switch case* biasanya digunakan untuk memilih di antara lebih dari 2 pilihan. *Switch case* digunakan untuk menggantikan struktur *if-else-if* dimana kondisinya mengacu pada variabel yang sama.


Pernyataan *switch* akan menyelesaikan kondisi yang diberikan dan kemudian membandingkan hasilnya dengan konstanta-konstanta yang berada di *case*. Jika semua konstanta-konstanta yang dibandingkan tidak ada yang sama, maka pernyataan yang berada pada default yang akan diproses. Bentuk umum penulisan *switch case* pada bahasa pemrograman lain, seperti : java, c, php dll.

```
switch(kondisi){
    case konstanta1:
        pernyataan1a;
        pernyataan1b;
        break;
    case konstanta2:
        pernyataan2a;
        pernyataan2b;
```

Tetapi python tidak memiliki itu. Untuk menggunakan *switch case* pada python, dapat diganti dengan membuat *function* pengganti *switch case* dan menggunakan tipe data *dictionary* untuk menyimpan data-datanya.

Contoh penggunaan :

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 4	Teknik Percabangan 2	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 5 dari 141

```
def hari(i):
    switcher = {
        0: 'Minggu',
        1: 'Senin',
        2: 'Selasa',
        3: 'Rabu',
        4: 'Kamis',
        5: 'Jumat',
        6: 'Sabtu'
    }
    return switcher.get(i, "Angka yang dimasukkan salah!")
```

Hasil :

```
Input Angka : 6
Sabtu

Process finished with exit code 0
```


B. Alat/Instrument/Aparatus/Bahan

1. Personal Computer (PC)
2. Aplikasi PyCharm

C. Keselamatan Kerja

1. Menggunakan alat dan bahan sesuai dengan fungsinya
2. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar
3. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen.
4. Tidak bercanda atau makan dan minum saat praktik
5. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 4	Teknik Percabangan 2	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 6 dari 141

D. Lembar Kerja

1. Menentukan kewajiban pekerja berdasarkan besar gaji

```

gaji = input("Input Gaji : ")
berkeluarga = True
punya_rumah = True


if int(gaji) >= 2700000:
    print("Gaji sudah diatas UMR")
    if berkeluarga:
        print("Wajib ikut asuransi dan menabung untuk pensiun")
    else:
        print("Tidak perlu ikut asuransi")
    if punya_rumah:
        print("Wajib bayar pajak rumah")
    else:
        print("Tidak wajib bayar pajak rumah")
else:
    print("Gaji belum UMR")

```

Buat file baru salin kode diatas dan jalankan program tersebut, kemudian amati hasilnya dan pahami logikanya. Apa yang terjadi jika variabel berkeluarga dan punya_rumah diganti nilainya ? Pahami proses yang terjadi pada program tersebut.

2. Memilih hari menggunakan switch case

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 4	Teknik Percabangan 2	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 7 dari 141

```


i = int(input("Input Angka : "))
def hari(i):
    switcher = {
        0: 'Minggu',
        1: 'Senin',
        2: 'Selasa',
        3: 'Rabu',
        4: 'Kamis',
        5: 'Jumat',
        6: 'Sabtu'
    }
    return switcher.get(i, "Angka yang dimasukkan salah!")

print(hari(i))

```

Buat file baru salin kode diatas dan jalankan program tersebut, kemudian amati hasilnya dan pahami logikanya. Apa yang terjadi jika variabel berkeluarga dan punya_rumah diganti nilainya ? Pahami proses yang terjadi pada program tersebut.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 4	Teknik Percabangan 2	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 8 dari 141

3. Menentukan nilai terbesar dari tiga input

```

a = int(input("Input Angka 1 : "))
b = int(input("Input Angka 2 : "))
c = int(input("Input Angka 3 : "))

if a > b:
    if a > c:
        maks = a
    else:
        if b > c:
            maks = b
        else:
            maks = c
else:
    maks = c

print('Nilai yang terbesar adalah: %d' % maks)


```

Buat file baru dengan nama “maksimal.py”. jalankan program tersebut dengan menekan *Ctrl+Shift+F10*, kemudian amati hasilnya yang diberika dan pahami logika dari program.

E. Tugas Praktikum

1. Buatlah program penentuan potongan harga berdasarkan total belanja, member dan status mahasiswa. Program akan memiliki tiga masukan yaitu total belanja, member dan mahasiswa. Jika total belanja yang dimasukkan diatas 150000 maka akan mendapatkan diskon 5% dan jika memiliki member mendapatkan diskon 15%. Apabila status mahasiswa mendapatkan diskon 10% dan kalau memiliki member mendapatkan diskon 20%.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 4	Teknik Percabangan 2	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 9 dari 141

```

Masukkan Harga : 200000
mahasiswa ? (y/t) : y
member ? (y/t) : t
Potongan harga sebesar 20000.0


Process finished with exit code 0

```

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

PETUNJUK PRAKTIKUM #10

PERULANGAN

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 7	Perulangan	
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021
			100 menit
			Hal 1 dari 141

A. Capaian Pembelajaran Mata Kuliah

1. Mampu mendefinisikan kebutuhan pengguna atau pasar terhadap kinerja (menganalisis, mengevaluasi dan mengembangkan) algoritma/metode berbasis komputer.
2. Mampu menelaah cara kerja sistem komputer dan mampu merancang dan mengembangkan berbagai algoritma/metode untuk memecahkan masalah.

B. Sub Capaian Pembelajaran Mata Kuliah

Mahasiswa mampu memecahkan permasalahan yang berkaitan teknik perulangan untuk contoh kasus sederhana.

C. Tujuan

Mahasiswa mampu **memecahkan** permasalahan yang terkait dengan perulangan.

D. Dasar Teori

Perulangan (looping) adalah suatu bagian yang bertugas melakukan kegiatan mengulang suatu proses sesuai dengan yang diinginkan. Banyak dari aplikasi perangkat lunak yang melakukan pekerjaan berulang-ulang sampai sebuah kondisi yang diinginkan, oleh karena itu perulangan merupakan bagian yang penting dalam pemrograman karena dengan adanya perulangan pembuat program tidak perlu menulis kode program sebanyak perulangan yang diinginkan.

Dalam bahasa menyediakan beberapa konstruksi perintah untuk melakukan proses-proses perulangan, yaitu :

1. FOR
2. While

MENYUSUN PENGULANGAN DENGAN “FOR”

Struktur for digunakan untuk melakukan perulangan yang tidak berkondisi. Artinya jumlah perulangannya telah diketahui dengan pasti.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

Dalam pengulangan for, tidak seperti di bahasa pemrograman C atau Java yang menggunakan nilai incremental untuk melakukan pengulangan. Di Python, for melakukan pengulangan dengan meng-iterasi elemen dari sebuah list. List ini dapat berisi kumpulan karakter, kumpulan string, kumpulan angka, atau kumpulan data jenis lainnya yang disediakan Python. (Untuk lebih lengkapnya di bab berikutnya akan dibahas lebih jauh tentang List di Python).

Bentuk Umum :

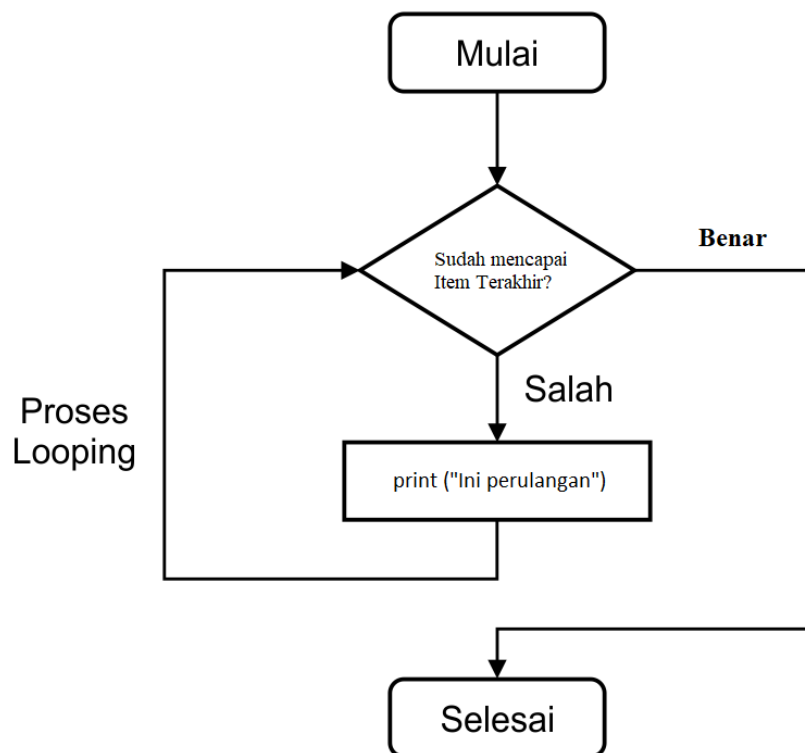
```
for indeks in range(banyak_perulangan):  
    # jalankan kode ini  
    # jalankan juga kode ini  
#kode ini tidak akan diulang karena berada di  
luar for
```

Variabel **indeks** berfungsi untuk menampung indeks, dan fungsi **range()** berfungsi untuk membuat list dengan range dari variable yang nanti kita buat.

Sebagai contoh coba lihat kode berikut. Kode berikut dieksekusi apabila variabel “angka” masih dibawah 8.

```
for i in range(8):  
    print ("Ini perulangan")
```

Flowchart



Kode diatas jika dieksekusi akan tampil seperti pada gambar berikut ini

```
Ini perulangan
Ini perulangan
Ini perulangan
Ini perulangan
Ini perulangan
Ini perulangan
Ini perulangan
Ini perulangan
Ini perulangan

Process finished with exit code 0
```

MENYUSUN PENGULANGAN DENGAN “WHILE”

Perulangan *while* biasa digunakan jika jumlah perulangan tidak diketahui atau memiliki kemungkinan dapat dilakukan kurang dari batas perulangan yang telah ditentukan. Perulangan *while* hanya akan melakukan perulangan selama kondisi perulangan terpenuhi. Perintah-perintah akan dilaksanakan apabila ekspresi masih dalam keadaan *true*. Di dalam *loop*

ada nilai yang mengontrol *loop* dan nilainya harus berubah, sehingga pada akhir program akan keluar dari *loop*.

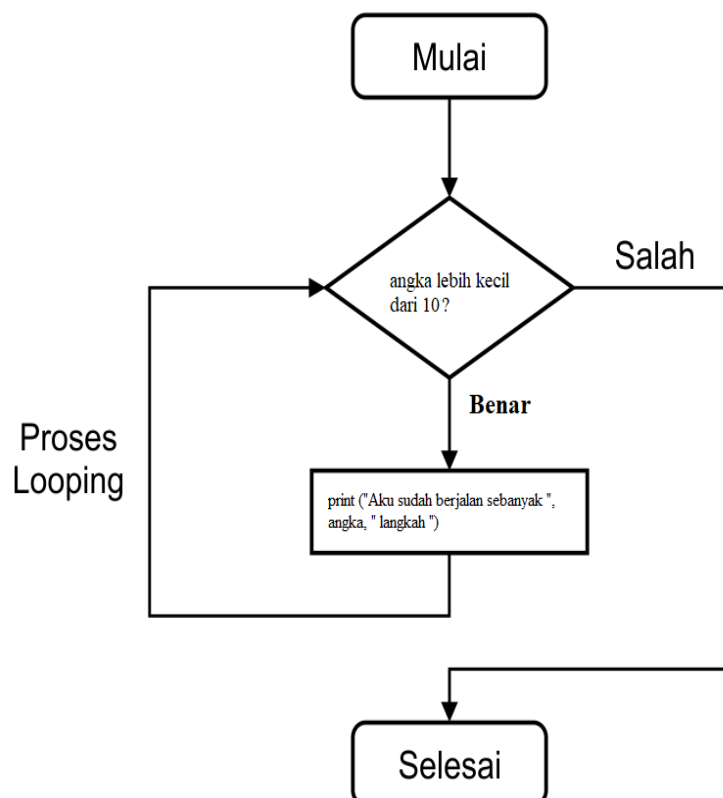
Bentuk Umum :

```
while(True):  
    # jalankan kode ini  
# kode ini berada di luar perulangan while
```

sebagai contoh coba lihat kode berikut. Kode berikut dieksekusi apabila variabel “angka” masih dibawah 10.

```
angka = 0  
while (angka < 10):  
    print ("Aku sudah berjalan sebanyak ", angka, " langkah ")  
    angka += 1
```

Flowchart



Kode diatas jika dieksekusi akan tampil seperti pada gambar berikut ini :

```
Aku sudah berjalan sebanyak 0 langkah
Aku sudah berjalan sebanyak 1 langkah
Aku sudah berjalan sebanyak 2 langkah
Aku sudah berjalan sebanyak 3 langkah
Aku sudah berjalan sebanyak 4 langkah
Aku sudah berjalan sebanyak 5 langkah
Aku sudah berjalan sebanyak 6 langkah
Aku sudah berjalan sebanyak 7 langkah
Aku sudah berjalan sebanyak 8 langkah
Aku sudah berjalan sebanyak 9 langkah

Process finished with exit code 0
```

E. Alat/Instrument/Aparatus/Bahan

1. Personal Computer (PC)
2. Aplikasi PyCharm

F. Keselamatan Kerja

1. Menggunakan alat dan bahan sesuai dengan fungsinya
2. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar.
3. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen.
4. Tidak bercanda atau makan dan minum saat praktik
5. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

G. Lembar Kerja

PRAKTIKUM FOR

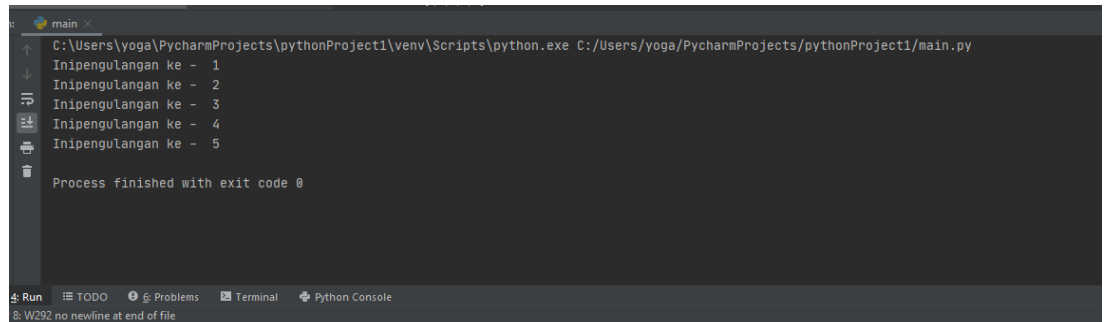
Membuat Perulangan For Dengan List

Disini ada sebuah *list* yang berisi [1, 2, 3, 4, 5], (sebuah list diawali oleh tanda '[' dan ditutup oleh tanda ']'). Banyaknya elemen pada *list* tersebut menentukan banyaknya pengulangan yang akan dilakukan saat melakukan pengulangan. Mari kita lihat implementasinya pada kode dibawah ini :

```
for i in [1, 2, 3, 4, 5]:
    print ("Ini pengulangan ke - ", i)
```

Pada contoh diatas, akan dicetak teks “ini pengulangan ke - “ sebanyak 5 kali. Nilai 'i' pada pengulangan tersebut akan selalu berganti nilainya setiap tahap pengulangan dilakukan. Misal ketika pengulangan pertama, nilai 'i'

akan berisi 1, ketika pengulangan kedua, nilai 'i' akan berisi 2, begitu seterusnya sampai elemen terakhir. Jika kode diatas dieksekusi akan menampilkan hasil seperti pada gambar dibawah ini :



```
main
C:\Users\yoga\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/yoga/PycharmProjects/pythonProject1/main.py
↑
Inipengulangan ke - 1
↓
Inipengulangan ke - 2
Inipengulangan ke - 3
Inipengulangan ke - 4
Inipengulangan ke - 5
Process finished with exit code 0
```

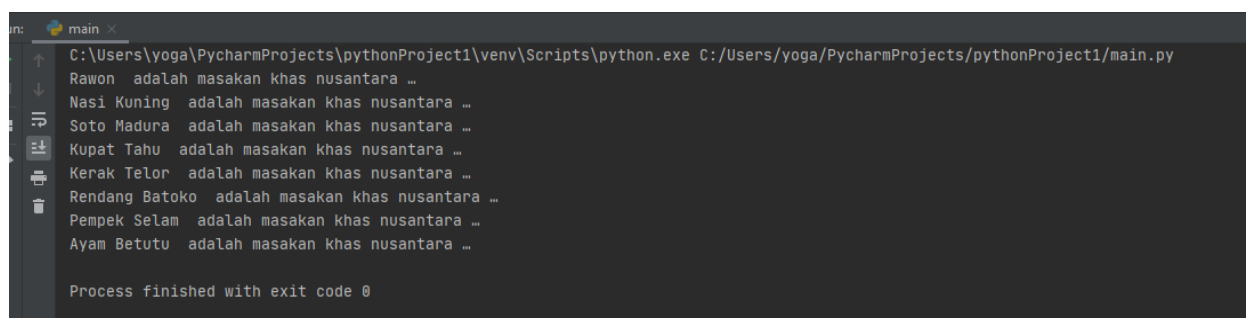
Membuat Perulangan For Dengan List Dengan Isi String

Selain menggunakan list yang berisi angka, List yang berisi string dapat digunakan juga untuk melakukan pengulangan for di Python. Misal terdapat list yang berisi seperti berikut ["Rawon", "Nasi Kuning", "Soto Madura", "Kupat Tahu", "Kerak Telor", "Rendang Batoko", "Pempek Selam", "Ayam Betutu"], dalam list tersebut terdapat elemen sebanyak delapan jenis masakan nusantara. Dengan demikian ketika pengulangan for menggunakan list masakan tadi, pengulangan akan dijalankan sebanyak

```
# pengulangan sebanyak 8 kali
for i in ["Rawon", "Nasi Kuning", "Soto Madura", "Kupat Tahu", "Kerak
Telor", "Rendang Batoko", "Pempek Selam", "Ayam Betutu"] :
    print (i, " adalah masakan khas nusantara ...")
```

delapan kali. Mari Anda lihat implementasinya pada kode dibawah ini :

Kode diatas jika dieksekusi akan terlihat seperti gambar dibawah ini :



```
main
C:\Users\yoga\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/yoga/PycharmProjects/pythonProject1/main.py
↑
Rawon  adalah masakan khas nusantara ...
↓
Nasi Kuning  adalah masakan khas nusantara ...
Soto Madura  adalah masakan khas nusantara ...
Kupat Tahu  adalah masakan khas nusantara ...
Kerak Telor  adalah masakan khas nusantara ...
Rendang Batoko  adalah masakan khas nusantara ...
Pempek Selam  adalah masakan khas nusantara ...
Ayam Betutu  adalah masakan khas nusantara ...
Process finished with exit code 0
```

Membuat Perulangan For Dengan String

String pada dasarnya merupakan *list* karakter. Misal terdapat *string* seperti berikut "abcde". Jika *string* tersebut digunakan pada pengulangan **for**, maka akan terjadi pengulangan sebanyak lima kali. Coba lihat kode dibawah ini :

```
# pengulangan sebanyak 5 kali for i in "abcde":  
    print (i, " adalah alfabet")
```

Kode diatas jika dieksekusi akan terlihat seperti gambar dibawah ini :



PRAKTIKUM WHILE

Membuat Perulangan While

Perulangan While untuk mengurutkan nilai dari 1 sampai 5.

```
i=1  
while i <=5:  
    print(i)  
    i = i + 1  
  
print("program selesai")
```

Kode diatas jika dieksekusi akan terlihat seperti gambar dibawah ini :

```
1
2
3
4
5
program selesai

Process finished with exit code 0
```

Menghitung Rata-rata dari Sejumlah Nilai

```
print("PROGRAM PYTHON MENGHITUNG NILAI RATA-RATA")

banyakdata = 5
i = 0
print()

jum = 0

while i < 5 :
    nilai = int(input("Masukkan data ke-%d: " % (i+1)))
    i = i + 1
    jum = jum + nilai
    rata = jum / banyakdata

print("\nRata-rata  = %0.2f" % rata)
```

Kode diatas jika dieksekusi akan terlihat seperti gambar dibawah ini :

```
PROGRAM PYTHON MENGHITUNG NILAI RATA-RATA

Masukkan data ke-1: 80
Masukkan data ke-2: 70
Masukkan data ke-3: 70
Masukkan data ke-4: 85
Masukkan data ke-5: 90

Rata-rata  = 79.00

Process finished with exit code 0
```

H. Tugas Praktikum

1. Buatlah sebuah program menghitung faktorial dengan hasil seperti dibawah

```
hasil faktorial dari 5 adalah 120
```

2. Buatlah program yang akan menampilkan deretan bilangan ganjil seperti ini : (1 3 5 7 9 11 13 15) menggunakan perulangan *for*!

PETUNJUK PRAKTIKUM #11

PERULANGAN BERSARANG

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 9	Perulangan Bersarang	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021
			Hal 1 dari 141

A. Capaian Pembelajaran Mata Kuliah

1. Mampu mendefinisikan kebutuhan pengguna atau pasar terhadap kinerja (menganalisis, mengevaluasi dan mengembangkan) algoritma/metode berbasis komputer.
2. Mampu menelaah cara kerja sistem komputer dan mampu merancang dan mengembangkan berbagai algoritma/metode untuk memecahkan masalah.

B. Sub Capaian Pembelajaran Mata Kuliah

Mahasiswa mampu memecahkan permasalahan yang berkaitan teknik perulangan untuk contoh kasus sederhana.

C. Tujuan

Mahasiswa mampu **memecahkan** permasalahan yang terkait dengan perulangan.

A. Dasar Teori

Perulangan Bersarang/Nested Loop

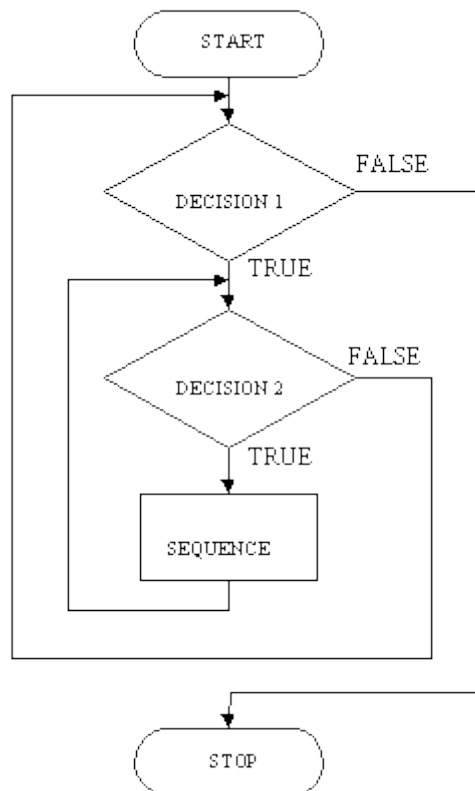
Perulangan bersarang/Nested Loop adalah perulangan yang didalamnya ada perulangan lagi. Perulangan ini digunakan untuk memecahkan suatu masalah yang tidak cukup digunakan 1 perulangan, seperti membuat program untuk menghitung seluruh data yang data-nya diinputkan oleh user, bisa juga untuk menghitung matriks, dan lain-lain.

Bentuk Umum:

```
for indeks in range(banyak_perulangan1):
    For indeks in range(banyak_perulangan2)
        # jalankan kode ini
#kode ini tidak akan diulang karena berada di luar for
```

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

PENJELASAN PERULANGAN BERSARANG DENGAN FLOWCHART



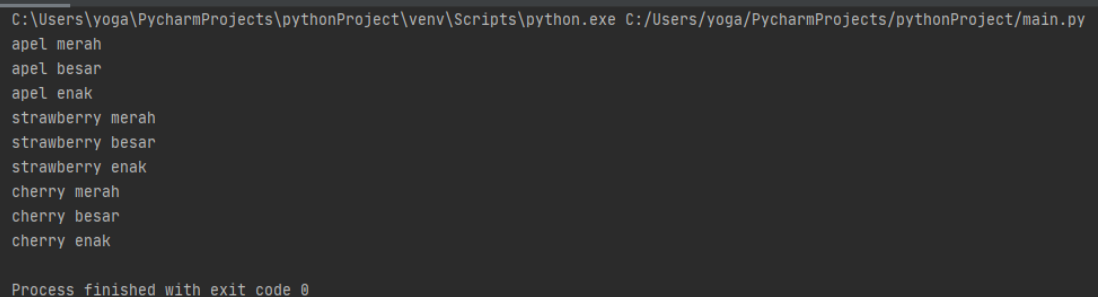
Saat loop Decision 1 bernilai true maka program akan berlanjut ke loop Decision 2 saat decision 2 bernilai true juga maka program akan mengeksekusi coding yang ada dalam statement setelah program selesai dieksekusi maka program akan mengulang ke decision 2 kalau decision 2 bernilai true juga maka akan mengulang seperti tadi jika decision 2 bernilai false maka akan berlanjut ke decision 1 jika decision 1 bernilai true maka akan berulang seperti tadi jika decision 1 bernilai false maka program berakhir.

1. PERULANGAN BERSARANG MENGGUNAKAN FOR

```
buah = ["apel", "strawberry", "cherry"]  
kata = ["merah", "besar", "enak"]  
  
for x in buah: #outer loop  
    for y in kata: #inner loop  
        print(x, y)
```

Dalam coding tersebut outer loop akan di eksekusi sekali lalu di ikuti oleh inner loop sampai habis, lalu lanjut lagi ke outer loop dan hasilnya akan seperti dibawah.

Hasilnya:



```
C:\Users\yoga\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/yoga/PycharmProjects/pythonProject/main.py  
apel merah  
apel besar  
apel enak  
strawberry merah  
strawberry besar  
strawberry enak  
cherry merah  
cherry besar  
cherry enak  
  
Process finished with exit code 0
```

2. Perulangan Bersarang Menggunakan WHILE

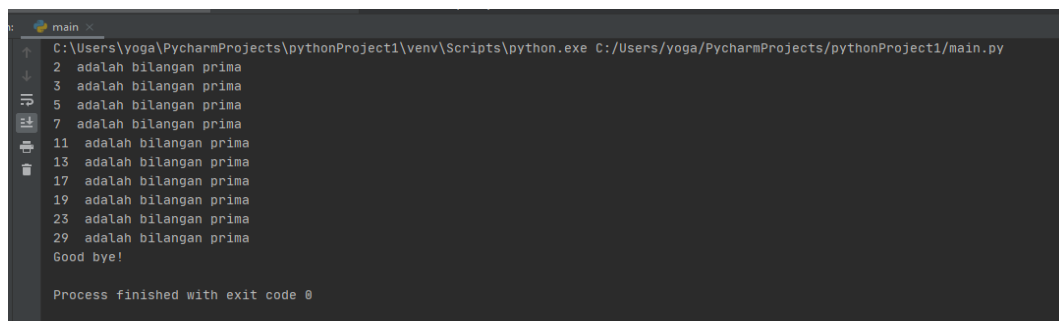
Contoh Program.

Di bawah ini adalah contoh program implementasi Nested Loop untuk mencetak bilangan prima dari 2 sampai 30.

```
# Nama file: nested_loop.py
# Program menampilkan bilangan prima dari 2 s/d 30

i = 2
while(i < 30):
    j = 2
    while(j <= (i/j)):
        if not(i%j): break
        j = j + 1
    if (j > i/j) : print (i, " adalah bilangan prima")
    i = i + 1
print ("Good bye!")
```

Apabila program diatas Anda jalankan, maka akan menampilkan output seperti di bawah ini.



```
main
C:\Users\yoga\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/yoga/PycharmProjects/pythonProject1/main.py
2 adalah bilangan prima
3 adalah bilangan prima
5 adalah bilangan prima
7 adalah bilangan prima
11 adalah bilangan prima
13 adalah bilangan prima
17 adalah bilangan prima
19 adalah bilangan prima
23 adalah bilangan prima
29 adalah bilangan prima
Good bye!
Process finished with exit code 0
```

```

#Deklarasi variabel
var_nilai = 0
var_i = 1

#Perulangan WHILE
while (var_nilai < 10) :
    print("Perulangan pertama Ke ",var_nilai)
    while(var_i < 3) :
        print(" Perulangan ke ", var_nilai," ", "var_i)
        var_i +=1
    #diluar perulangan var_i
    var_i = 1
    var_nilai +=1
#diluar_perulangan var_nilai
print("var_nilai = ",int(var_nilai)," = 10. Bernilai
False")

```

Penjelasan Program:

- *while (var_nilai < 10)*. Artinya adalah ketika var_nilai bernilai kurang dari 10, maka perulangan tetap berjalan
- *while (var_i < 3)*. Artinya ketika var_i bernilai kurang dari 3, maka perulangan terus berjalan.
- *var_i += 1*. Artinya menambahkan nilai var_i. Hal ini diperlukan karena while tidak menyediakan penambahan var_i secara otomatis. Jika tidak ditambahkan, maka nilai dari var_i tetap 1 dan perulangan tidak akan berhenti
- *var_i = 1*. Ketika perulangan kedua berjalan, nilai dari var_i terus berjalan. Disini kita mengembalikan nilai var_i menjadi 1.
- *var_nilai += 1*. artinya menambahkan nilai var_nilai agar bertambah 1.

3. Contoh Perulangan Bersarang Menggunakan FOR dan WHILE

```
#Deklarasi variabel
var_nilai = 0
var_i = 1

#Perulangan FOR
for var_nilai in range (0,10) :
    print("Perulangan pertama Ke ",var_nilai)
    while(var_i < 3) :
        print(" Perulangan ke ", var_nilai," ",var_i)
        var_i +=1
    #diluvar perulangan var_i
    var_i = 1
#diluvar_perulangan var_nilai
print("var_nilai = ",int(var_nilai)+1," = 10. Bernilai False")
```

Penjelasan Program

- *for var_nilai in range (0,10).* artinya selama var_nilai nilainya berada diantara 0 sampai 10, maka program akan terus berjalan.
- *while (var_i < 3).* Artinya ketika var_i bernilai kurang dari 3, maka perulangan terus berjalan.
- *var_i += 1.* Artinya menambahkan nilai var_i. Hal ini diperlukan karena while tidak menyediakan penambahan var_i secara otomatis. Jika tidak ditambahkan, maka nilai dari var_i tetap 1 dan perulangan tidak akan berhenti
- *var_i = 1.* Ketika perulangan kedua berjalan, nilai dari var_i terus berjalan. Disini kita mengembalikan nilai var_i menjadi 1.

Dari Contoh perulangan bersarang diatas, menghasilkan output yang sama, yaitu seperti dibawah ini :

```
main x
C:\Users\yoga\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/yoga/PycharmProjects/pythonProject1/main.py
Perulangan pertama Ke 0
  Perulangan ke 0 , 1
  Perulangan ke 0 , 2
Perulangan pertama Ke 1
  Perulangan ke 1 , 1
  Perulangan ke 1 , 2
Perulangan pertama Ke 2
  Perulangan ke 2 , 1
  Perulangan ke 2 , 2
Perulangan pertama Ke 3
  Perulangan ke 3 , 1
  Perulangan ke 3 , 2
Perulangan pertama Ke 4
  Perulangan ke 4 , 1
  Perulangan ke 4 , 2
Perulangan pertama Ke 5
  Perulangan ke 5 , 1
  Perulangan ke 5 , 2
Perulangan pertama Ke 6
  Perulangan ke 6 , 1
  Perulangan ke 6 , 2
Perulangan pertama Ke 7
  Perulangan ke 7 , 1
  Perulangan ke 7 , 2
Perulangan pertama Ke 8
  Perulangan ke 8 , 1
  Perulangan ke 8 , 2
Perulangan pertama Ke 9
  Perulangan ke 9 , 1
  Perulangan ke 9 , 2
var_nilai = 10 = 10. Bernilai False

Process finished with exit code 0
```

Python juga mendukung penggunaan kontrol di dalam looping. Ada 3 kontrol, yaitu :

break : untuk menghentikan looping ketika terjadi kondisi tertentu.

continue : untuk melanjutkan operasi, ketika pada blok statemen menghasilkan nilai yang diharapkan atau yang dicari.

pass : kontrol ini tidak menghasilkan apa-apa, pass akan berguna untuk mengecek apakah statemen berjalan apa tidak.

B. Alat/Instrument/Aparatus/Bahan

1. Personal Computer (PC)
2. Aplikasi PyCharm

C. Keselamatan Kerja

1. Menggunakan alat dan bahan sesuai dengan fungsinya
2. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar.
3. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen.
4. Tidak bercanda atau makan dan minum saat praktik
5. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

D. Lembar Kerja

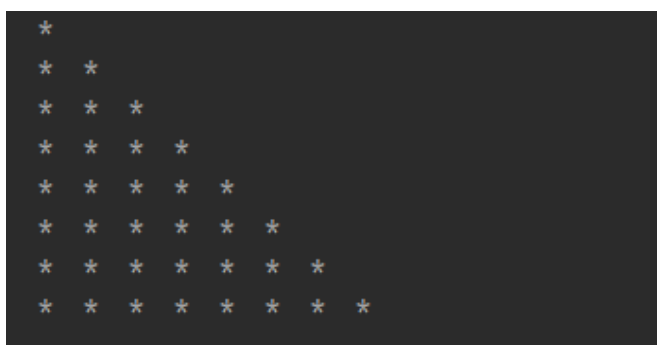
1. MEMBUAT SEGITIGA SIKU SIKU DENGAN NESTED LOOP

```
string = ""
bar = 1
x = 8

while bar <= x:
    kol = bar
    while kol > 0:
        string = string + " * "
        kol = kol - 1

    string = string + "\n"
    bar = bar + 1
print(string)
```

Kode diatas jika dieksekusi akan tampil seperti pada gambar berikut ini



2. MEMBUAT SEGITIGA DENGAN NESTED LOOP

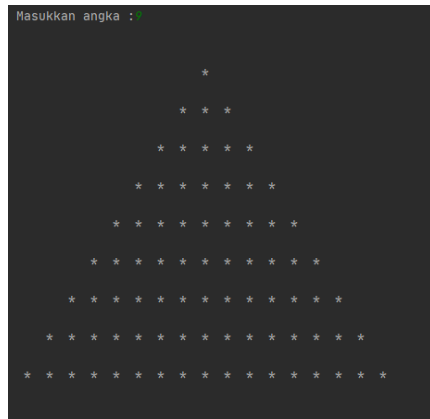
```
string = ""

x = int(input("Masukkan angka :"))
bar = x
# Looping Baris
while bar >= 0:
    # Looping Kolom Spasi Kosong
    kol = bar
    while kol > 0:
        string = string + "  "
        kol = kol - 1
    # Looping Kolom Bintang Sisi Kiri
    kiri = 1
    while kiri < (x - (bar - 1)):
        string = string + " * "
        kiri = kiri + 1
    # Looping Kolom Bintang Sisi Kanan
    kanan = 1
    while kanan < kiri - 1:
        string = string + " * "
        kanan = kanan + 1

    string = string + "\n\n"
    bar = bar - 1

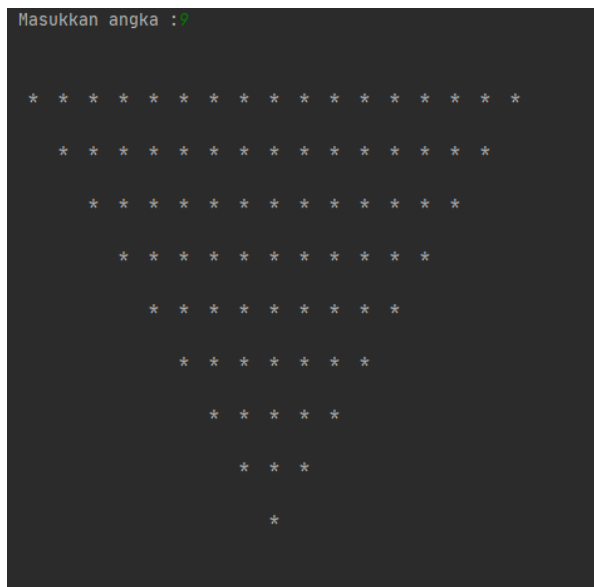
print(string)
```

Kode diatas jika dieksekusi akan tampil seperti pada gambar berikut ini




E. Tugas Praktikum

Buatlah program dengan hasil seperti dibawah



PETUNJUK PRAKTIKUM #12

ARRAY 1 DIMENSI

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 10	Array 1 Dimensi	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021
			Hal 1 dari 141

A. Capaian Pembelajaran Mata Kuliah

1. Mampu menerapkan algoritma/metode dalam perangkat lunak berbasis komputer.
2. Mampu menerapkan konsep teoritis dalam bidang algoritma.

B. Sub Capaian Pembelajaran Mata Kuliah

Mahasiswa mampu menerapkan konsep array 1 dimensi serta mengimplementasikan array 1 dimensi pada bahasa pemrograman.

C. Tujuan

1. mahasiswa mampu **mensimulasikan** array satu dimensi menunjukkan dan mengetahui contoh-contoh permasalahan yang diselesaikan dengan array

D. Dasar Teori

LIST

List adalah struktur data pada python yang mampu menyimpan lebih dari satu data, seperti array. Untuk mendefinisikan sebuah list Anda cukup buat sebuah variabel dengan nama yang diinginkan, kemudian isi variabel tersebut dengan list yang akan dibuat. List diawali dengan tanda '[' dan diakhiri dengan tanda ']'. Isinya dapat beragam, dapat string, number, object, bahkan list lagi. Pemisah antara data yang satu dengan yang lainnya digunakan tanda koma. List dapat ditambah isinya, dirubah data pada elemennya, hapus elemen, dan hapus seluruh list.

a) Cara Membuat List di Python

List dapat kita buat seperti membuat variabel biasa, namun nilai variabelnya diisi dengan tanda kurung siku ([]).

Contoh:

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

```
# Membuat List
kosong
warna = []

# Membuat list
dengan isi 1 item
hobi = ["membaca"]
```

Apabila list-nya memiliki lebih dari satu isi, maka kita bisa memisahkannya dengan tanda koma.

Contoh:

```
buah = ["jeruk", "apel", "mangga", "duren"]
```

List dapat diisi dengan tipe data apa saja, string, integer, float, double, boolean, object, dan sebagainya. Isi didalam list dapat dicampur.

Contoh:

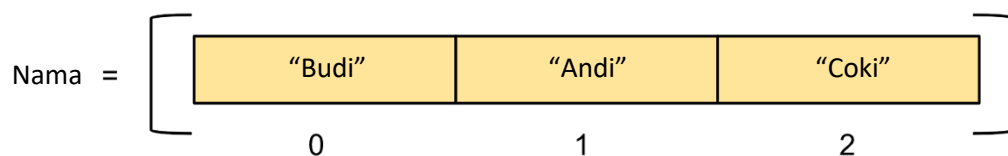
```
laci = ["buku", 21, True, 34.12]
```

Ada empat jenis tipe data pada list **laci**:

1. **"buku"** adalah tipe data string;
2. **21** adalah tipe data integer;
3. **True** adalah tipe data boolean;
4. dan **34.12** adalah tipe data float.

b) Cara Mengambil Nilai Dari List

List sama seperti array, list juga memiliki nomer indeks untuk mengakses data atau isinya.



Nomer indeks *list* selalu dimulai dari nol (0).

Nomer indeks ini yang kita butuhkan untuk mengambil isi (item) dari *list*.

Contoh:

```
# Kita punya list nama-nama buah
buah = ["apel", "anggur", "mangga",
"jeruk"]

# Misanya kita ingin mengambil mangga
# Maka indeksinya adalah 2
print (buah[2])
```

Akan menghasilkan output:

```
main x
C:\Users\yoga\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/yoga/PycharmProjects/pythonProject1/main.py
mangga
Process finished with exit code 0
```

Mengganti Nilai List

List bersifat *mutable*, artinya isinya bisa kita ubah-ubah.

Contoh:

```
# list mula-mula
buah = ["jeruk", "apel", "mangga", "duren"]
# mengubah nilai index ke-2
buah[2] = "kelapa"
```

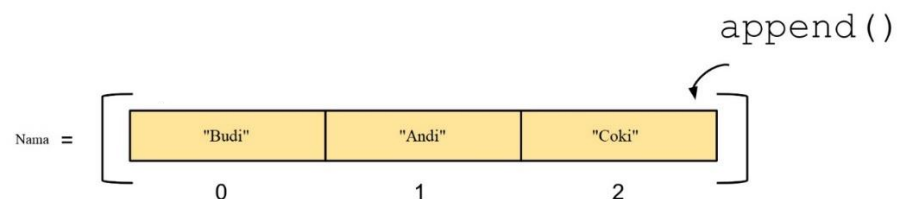
Maka "mangga" akan diganti dengan "kelapa".

```
["jeruk", "apel", "kelapa", "duren"]
```

c) Menambahkan Item List

Terdapat 3 metode (*method*) atau fungsi yang bisa digunakan untuk menambahkan isi atau item ke List:

1. **append(item)** menambahkan item dari belakang.
2. **insert(index, item)** menambahkan item dari indeks tertentu

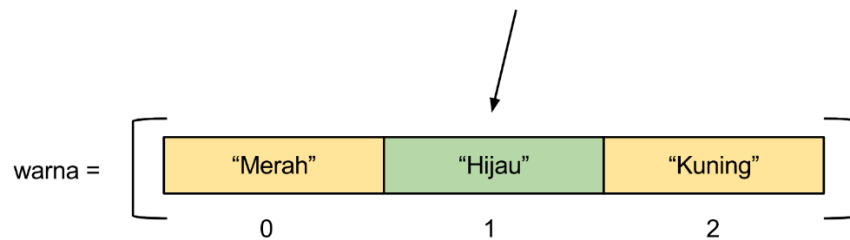


Contoh:

```
#list mula-mula
buah = ["jeruk", "apel", "mangga", "duren"]
# Tambahkan manggis
buah.append("manggis")
```

Selain `append()` kita juga bisa menggunakan method `insert()` untuk menambahkan item pada indeks tertentu.

```
warna.insert(1, "hijau")
```

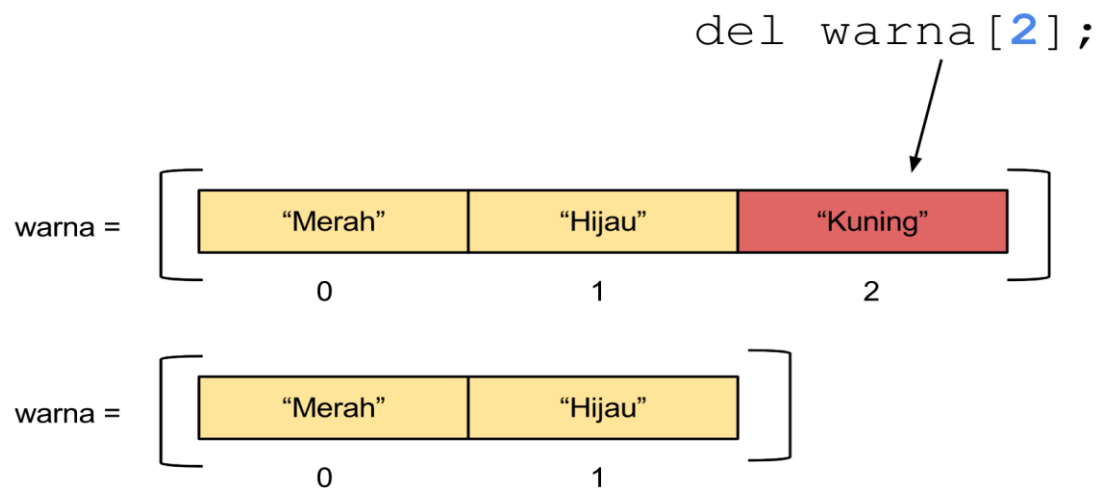


Contoh:

```
#list mula-mula
buah = ["jeruk", "apel", "mangga", "duren"]
buah.insert(2, "duren")
```

d) Menghapus Item Di List

Untuk menghapus salah satu isi dari *List*, kita bisa menggunakan perintah `del`.



Perintah `del` akan menghapus sebuah variabel dari memori.

Contoh:

```
# Membuat List
todo_list = [
    "Belajar Python",
    "Belajar Django",
    "Belajar MongoDB",
    "Belajar Sulap",
    "Belajar Flask"
]

# Misalkan kita ingin menghapus "Belajar
Sulap"
# yang berada di indeks ke-3
del todo_list[3]

print (todo_list)
```

Hasilnya, "Belajar Sulap" akan dihapus:

```
['Belajar Python', 'Belajar Django', 'Belajar MongoDB',  
'Belajar Flask']
```

Selain menggunakan perintah `del`, kita juga bisa menggunakan *method* `remove()` dengan parameter item yang akan dihapus.

Contoh:

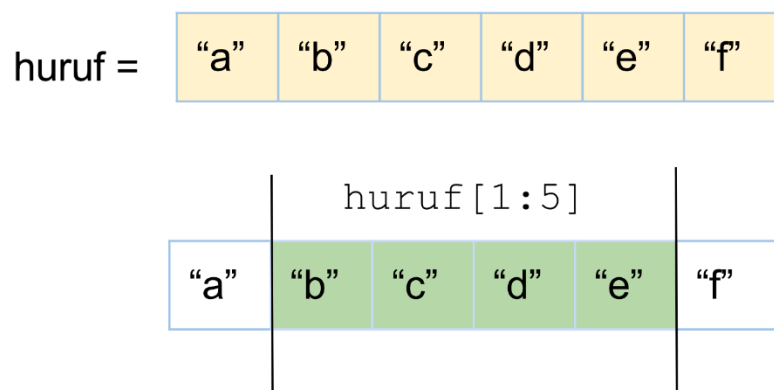
```
# mula-mula kita  
punya list  
a = ["a", "b", "c",  
     "d"]  
# kemudian kita  
hapus b  
a.remove("b")  
  
print(a)
```

Hasilnya:

```
['a', 'c', 'd']
```

e) Memotong List

Seperti string, list juga dapat dipotong-potong.

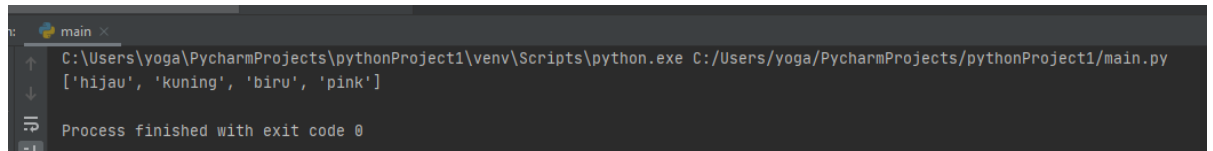


Contoh:

```
# Kita punya list warna
warna = ["merah", "hijau", "kuning", "biru", "pink", "ungu"]

# Kita potong dari indeks ke-1 sampai ke-5
Print( warna[1:5])
```

Hasilnya:



```
main x
C:\Users\yoga\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/yoga/PycharmProjects/pythonProject1/main.py
['hijau', 'kuning', 'biru', 'pink']
Process finished with exit code 0
```

f) Operasi List

Ada beberapa operasi yang bisa dilakukan terhadap List, diantaranya:

- Penggabungan (+)
- Perkalian (*)

Contoh:

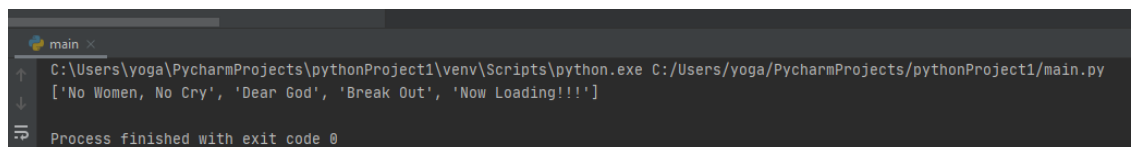
```
# Beberapa list lagu
list_lagu = [
    "No Women, No Cry",
    "Dear God"
]

# playlist lagu favorit
playlist_favorit = [
    "Break Out",
    "Now Loading!!!"
]

# Mari kita gabungkan keduanya
semua_lagu = list_lagu + playlist_favorit

print (semua_lagu)
```

Hasilnya:



```
main x
C:\Users\yoga\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/yoga/PycharmProjects/pythonProject1/main.py
['No Women, No Cry', 'Dear God', 'Break Out', 'Now Loading!!!']
Process finished with exit code 0
```

Sedangkan untuk operasi perkalian hanya dapat dilakukan dengan bilangan.

Contoh:

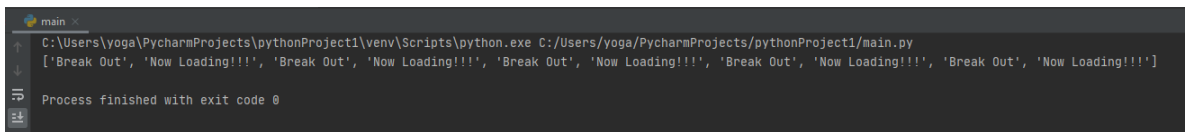
```
# playlist lagu favorit
playlist_favorit = [
    "Break Out",
    "Now Loading!!!"
]

# ulangi sebanyak 5x
ulangi = 5

now_playing = playlist_favorit * ulangi

print (now_playing)
```

Hasilnya:



```
main
C:\Users\yoga\PycharmProjects\pythonProject1\main.py
['Break Out', 'Now Loading!!!', 'Break Out', 'Now Loading!!!', 'Break Out', 'Now Loading!!!', 'Break Out', 'Now Loading!!!', 'Break Out', 'Now Loading!!!']
Process finished with exit code 0
```

E. Alat/Instrument/Aparatus/Bahan

1. Personal Computer (PC)
2. Aplikasi PyCharm

F. Keselamatan Kerja

1. Menggunakan alat dan bahan sesuai dengan fungsinya
2. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar.
3. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen.
4. Tidak bercanda atau makan dan minum saat praktik
5. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

G. Lembar Kerja

1. Membuat Program Dengan List

Coba salin program berikut.

Tujuan

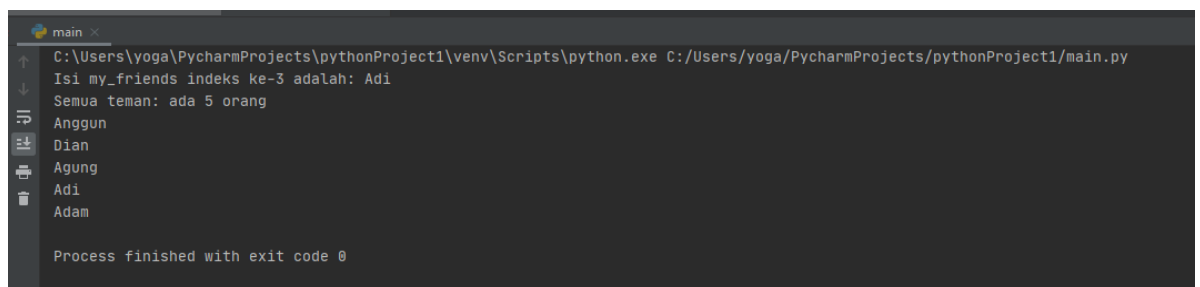
1. Buat sebuah list untuk menyimpan kenalanmu
2. Isi list sebanyak 5
3. Tampilkan isi list indeks nomer 3
4. Tampilkan semua teman dengan perulangan
5. Tampilkan panjang list

```
# Buat list untuk menampung nama-nama teman
my_friends = ["Anggun", "Dian", "Agung", "Adi",
              "Adam"]

# Tampilkan isi list my_friends dengan nomer indeks 3
print ("Isi my_friends indeks ke-3 adalah:
{}".format(my_friends[3]))
# Tampilkan semua daftar teman
print ("Semua teman: ada {}
orang".format(len(my_friends)))
for friend in my_friends:
    print(friend)
```

Pada kode di atas, kita menggunakan fungsi `len()` untuk mengambil panjang *list*.

Hasil outputnya:



```
main x
C:\Users\yoga\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/yoga/PycharmProjects/pythonProject1/main.py
Isi my_friends indeks ke-3 adalah: Adi
Semua teman: ada 5 orang
Anggun
Dian
Agung
Adi
Adam
Process finished with exit code 0
```

2. Membuat Program Dengan Method `append()`.

Salinlah Coding Berikut

```
# Membuat list kosong untuk menampung hobi
hobi = []
stop = False
i = 0

# Mengisi hobi
while (not stop):
    hobi_baru = input("Inputkan hobi yang ke-{}: ".format(i))
    hobi.append(hobi_baru)

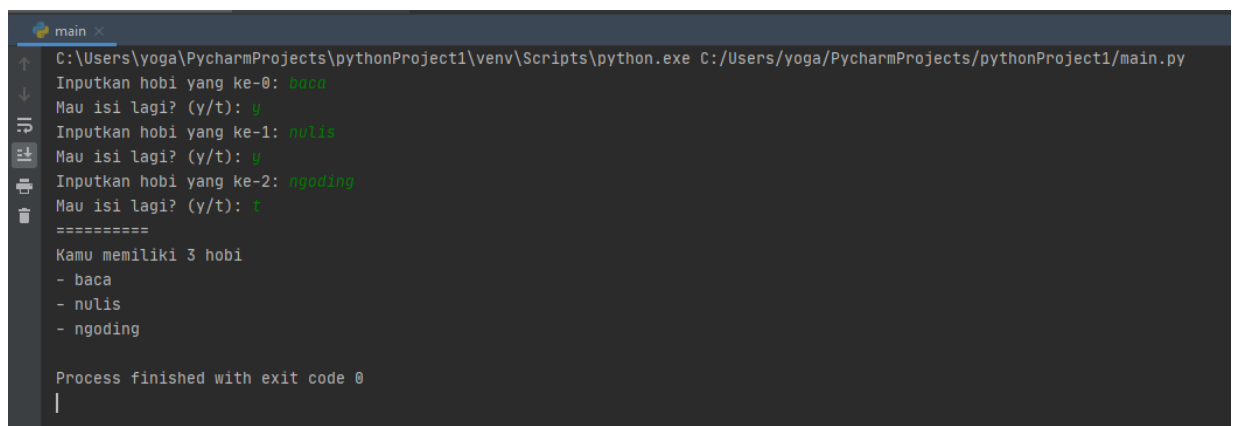
    # Increment i
    i += 1

    tanya = input("Mau isi lagi? (y/t): ")
    if (tanya == "t"):
        stop = True

# Cetak Semua Hobi
print("=" * 10)
print("Kamu memiliki {} hobi".format(len(hobi)))
for hb in hobi:
    print("- {}".format(hb))
```

Memfaatkan *method* `append()`. Silahkan langsung di ketik dan dicoba.

Coba eksekusi dan inputkan sebuah nilai.



```
main x
C:\Users\yoga\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/yoga/PycharmProjects/pythonProject1/main.py
Inputkan hobi yang ke-0: baca
Mau isi lagi? (y/t): y
Inputkan hobi yang ke-1: nulis
Mau isi lagi? (y/t): y
Inputkan hobi yang ke-2: ngoding
Mau isi lagi? (y/t): t
=====
Kamu memiliki 3 hobi
- baca
- nulis
- ngoding

Process finished with exit code 0
|
```

3. Membuat Program Dengan Perintah Del

Salinlah coding Berikut

```
# Membuat List
todo_list = [
    "Balajar membaca",
    "Belajar bicara",
    "Belajar berjalan",
    "Belajar berlari",
    "Belajar berenang"
]

# Misalkan kita ingin menghapus "Belajar
Sulap"
# yang berada di indeks ke-3
del todo_list[3]

print (todo_list)
```

Hasilnya, "Belajar berlari" akan dihapus

```
['Balajar membaca', 'Belajar bicara', 'Belajar berjalan', 'Belajar berenang']
```

4. Membuat Program operasi list

Salinlah coding Berikut

```
# Beberapa list lagu
list_lagu = [
    "anak SMA 1",
    "anak SMP 1"
]

# playlist lagu favorit
playlist_favorit = [
    "Anak Jaksel",
    "Anak STM"
]

# Mari kita gabungkan keduanya
semua_lagu = list_lagu + playlist_favorit

print (semua_lagu)
```

Hasilnya

```
['anak SMA 1', 'anak SMP 1', 'Anak JakseL', 'Anak STM']
```

H. Tugas Praktikum


1. Buatlah Program dengan menggunakan list dengan isian list 10 data lalu tampilkan data dari indeks 2 sampai 5 saja
2. Buatlah Program dengan menggunakan list yang dapat menginputkan data nama teman teman kalian menggunakan method prepend
3. Buatlah program untuk menghitung rata rata dan jumlah total nilai yang di inputkan dengan menggunakan list dengan contoh hasil sebagai berikut

```
Masukkan banyaknya nilai: 4
Masukkan nilai ke-1 : 30
Masukkan nilai ke-2 : 40
Masukkan nilai ke-3 : 50
Masukkan nilai ke-4 : 20

Hasil nilai total adalah : 140.0
Hasil rata-rata adalah : 35.0
```


PETUNJUK PRAKTIKUM #13

ARRAY 2 DIMENSI

	Fakultas Teknologi dan Informatika		
	Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 11	Array 2 Dimensi	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021
			Hal 1 dari 141

A. Capaian Pembelajaran Mata Kuliah

1. Mampu menerapkan algoritma/metode dalam perangkat lunak berbasis komputer.
2. Mampu menerapkan konsep teoritis dalam bidang algoritma.

B. Sub Capaian Pembelajaran Mata Kuliah

Mahasiswa mampu menerapkan konsep array 2 dimensi serta mengimplementasikan array 2 dimensi pada bahasa pemrograman.

C. Tujuan

1. mahasiswa mampu **mensimulasikan** array dua dimensi menunjukkan dan mengetahui contoh-contoh permasalahan yang diselesaikan dengan array.

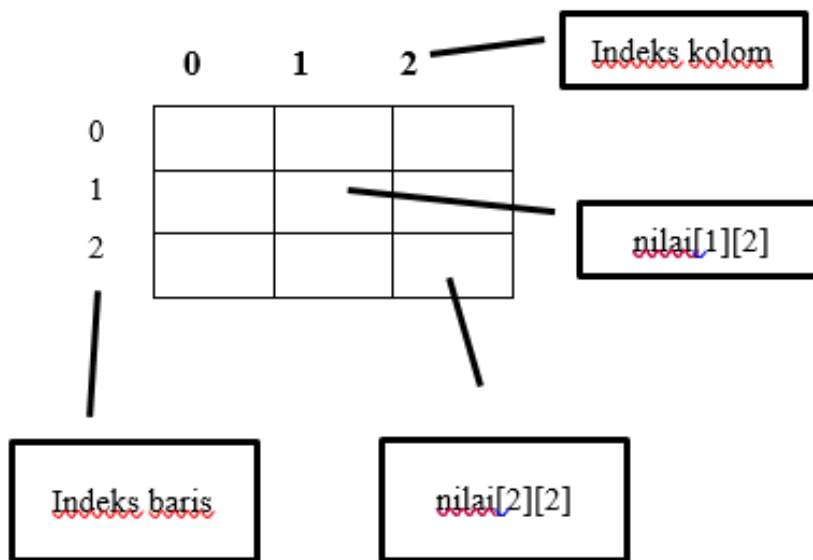
D. Dasar Teori

LIST 2 Dimensi

List dapat juga memiliki lebih dari satu dimensi atau disebut dengan multi dimensi. List 2 dimensi biasanya digunakan untuk menyimpan struktur data yang kompleks seperti tabel, matriks, graph, tree, dsb.

Terdapat 2 jenis list, yaitu list 1 dimensi dan list 2 dimensi. List dengan 1 dimensi dapat digambarkan sebagai sebuah baris, sedangkan list 2 dimensi merupakan dapat digambarkan seperti sebuah matrik. Berikut format pendeklarasian list 2 dimensi:

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------



Contoh Penggunaan

Salinlah kode program di bawah ini:

```
# List minumandengan 2 dimensi
list_minuman = [
    ["Kopi", "Susu", "Teh"],
    ["Jus Apel", "Jus Melon", "Jus Jeruk"],
    ["Es Kopi", "EsCampur", "EsTeler"]
]

# Cara mengakses list multidimensi
# misalkankitainginmengambil "esCampur"
print("kitaakanakseske 2,1")
print(list_minuman[2][1])
print(" ")

# Cara menampilkansemuaminuman
for menu in list_minuman:
    for minuman in menu:
        print (minuman)
```

Fungsi Di Dalam LIST

Berikut macam-macam fungsi bawaan list:

- A. `len(list)`
- B. `max(list)`
- C. `min(list)`
- D. `list(tuples)`

len(list)

Fungsi `len(list)` digunakan untuk menghitung panjang sebuah list.

Contoh Program

```
#Nama File : list_len.py
list1, list2 = [123, 'xyz', 'zara'], [456, 'abc']

print ("Panjang list1 : ", len(list1))
print ("Panjang list2 : ", len(list2))
```

Jika Anda jalankan program diatas, maka akan menampilkan output sebagai berikut:

```
Panjang list1 : 3
Panjang list2 : 2
```

max(list)

Fungsi `max(list)` digunakan untuk menampilkan nilai maksimum dari sebuah list.

Contoh Program

```
#Nama File : list_max.py
values = [-100, 1, 10, 1000]

print("Nilai Maksimum: ", max(values))
```

Jika program diatas Anda jalankan, maka akan menampilkan output sebagai berikut:

```
Nilai Maksimum: 1000
```

min(list)

Fungsi min(list) digunakan untuk menampilkan nilai minimum dari sebuah list.

Contoh Program

```
#Nama File : list_min.py
values = [-100, 1, 10, 1000]
print("Nilai Minimum: ", min(values))
```

Jika program diatas Anda jalankan, maka akan menampilkan output sebagai berikut:

```
Nilai Minimum: -100
```

list(tuples)

Fungsi list(tuples) digunakan untuk mengkonversi dari tuples menjadi list.

Contoh Program

```
#Nama File : list_convert.py

aTuples = (1234, 'abc', 'def', 'ghi', 4321);
aLists = list(aTuples)
```

Jika Anda jalankan program diatas, maka akan menampilkan output sebagai berikut:

```
Elemen dari list : [1234, 'abc', 'def', 'ghi', 4321]
```

E. Alat/Instrument/Aparatus/Bahan

1. Personal Computer (PC)
2. Aplikasi PyCharm

F. Keselamatan Kerja

1. Menggunakan alat dan bahan sesuai dengan fungsinya
2. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar.
3. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen.
4. Tidak bercanda atau makan dan minum saat praktik
5. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

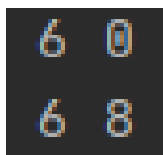
G. Lembar Kerja

1. Melakukan Penjumlahan Matriks

Penjumlahan matriks dilakukan dengan menjumlahkan setiap elemen. Hasil penjumlahan tersebut akan menjadi elemen baru. Masing - masing matriks kita akses setiap elemennya pada koordinat yang sama kemudian kita jumlahkan untuk mendapatkan elemen baru.

```
mat1 = [  
    [5, 0],  
    [2, 6],  
]  
  
mat2 = [  
    [1, 0],  
    [4, 2],  
]  
  
for x in range(0, len(mat1)):  
    for y in range(0, len(mat1[0])):  
        print (mat1[x][y] + mat2[x][y], end=' '),  
        print('')
```

Jika *source code* kita jalankan, maka akan muncul *output* seperti berikut:



```
6 0  
6 8
```

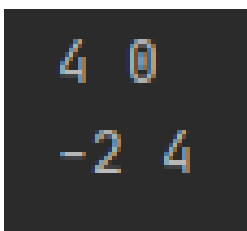
2. Melakukan Pengurangan Matriks

Tidak berbeda jauh dengan penjumlahan matriks, pada pengurangan matriks kita hanya mengganti operatornya saja dengan tanda kurang (-). Maka matriks baru akan terbentuk sebagai hasil dari pengurangan setiap

kedua elemen matriks. Sebagai contoh berikut adalah *source code* untuk melakukan pengurangan matriks:

```
mat1 = [  
    [5, 0],  
    [2, 6],  
]  
  
mat2 = [  
    [1, 0],  
    [4, 2],  
]  
  
for x in range(0, len(mat1)):  
    for y in range(0, len(mat1[0])):  
        print (mat1[x][y] - mat2[x][y], end=' '),  
        print('')
```

Jika *source code* kita jalankan, maka akan muncul *output* seperti berikut:



The output shows the result of subtracting mat2 from mat1. The first row is 4 0 and the second row is -2 4.

3. Melakukan Perkalian Matriks

Perkalian matriks merupakan salah satu operasi dasar yang *tricky*. Karena di dalamnya bukan hanya terdapat operasi perkalian, melainkan juga penjumlahan. Perkalian suatu matriks memang tidak sama dengan bilangan biasa, tidak juga langsung mengalikan setiap elemen. Perkalian matriks dilakukan dengan menjumlahkan hasil perkalian suatu baris matriks

pertama ke kolom matriks kedua. Setiap baris di matriks pertama akan dikalikan ke setiap kolom di matriks kedua.

Di Python, kita akan menggunakan *nested loop* for di dalam nested loop yang kedua. *Looping* ketiga tersebut kita gunakan untuk melakukan proses penjumlahan hasil perkalian baris dan kolom. Hasilnya elemen matriks baru akan ditempatkan pada koordinat tersebut. Sebagai contoh berikut adalah *source code* yang melakukan proses perkalian matriks:

```
mat1 = [  
    [5, 0],  
    [2, 6],  
]  
  
mat2 = [  
    [1, 0],  
    [4, 2],  
]  
  
mat3 = []  
  
for x in range(0, len(mat1)):  
    row = []  
    for y in range(0, len(mat1[0])):  
        total = 0  
        for z in range(0, len(mat1)):  
            total = total + (mat1[x][z] * mat2[z][y])  
        row.append(total)  
    mat3.append(row)  
  
for x in range(0, len(mat3)):  
    for y in range(0, len(mat3[0])):  
        print (mat3[x][y], end=' ')  
    print ()
```

Berikut adalah *output* dari *source code* diatas:


```
5 0  
26 12
```


H. Tugas Praktikum

1. Transpose matrik adalah terjadinya pertukaran tempat antara elemen baris dan elemen kolom dan sebaliknya pada sebuah matrik. Buatlah sebuah program transpose matrik.
2. Buatlah sebuah list dengan data lebih dari 5 , hitunglah panjang ,nilai tertinggi dan nilai terendah list tersebut menggunakan fungsi yang telah dijelaskan diatas.

PETUNJUK PRAKTIKUM #14

PROCEDURE

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 12	Procedure	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 1 dari 141

A. Capaian Pembelajaran Mata Kuliah

1. Mampu mendefinisikan kebutuhan pengguna atau pasar terhadap kinerja (menganalisis, mengevaluasi dan mengembangkan) algoritma/metode berbasis komputer.
2. Mampu menelaah cara kerja sistem komputer dan mampu merancang dan mengembangkan berbagai algoritma/metode untuk memecahkan masalah.

B. Sub Capaian Pembelajaran Mata Kuliah

Mahasiswa mampu mengoperasikan sub program menjadi sebuah prosedur dengan benar pada Bahasa pemrograman.

C. Tujuan

Mahasiswa mampu **menjabarkan** dan **mengimplementasikan** penggunaan value parameter dan variable parameter, ketepatan menyusun algoritma prosedur, ketepatan menyelesaikan contoh kasus dengan prosedur.

D. Dasar Teori

Prosedur merupakan bagian dari program yang berupa serangkaian *statement* yang memiliki nama. Sebuah prosedur dapat terdiri atas satu atau lebih dari satu prosedur. Penggunaan prosedur dapat memberikan beberapa keuntungan, sebagai berikut:

1. Program besar dapat di pisah-pisah menjadi program-program kecil melalui fungsi.
2. Kemudahan dalam mencari kesalahan-kesalahan karena alur logika jelas dan kesalahan dapat dilokalisasi dalam suatu modul tertentu.
3. Memperbaiki atau memodifikasi program dapat dilakukan pada suatu modul tertentu saja tanpa mengganggu keseluruhan program.
4. Dapat digunakan kembali (Reusability) oleh program atau fungsi lain.
5. Meminimalkan penulisan perintah yang sama.

Oleh karena itu, prosedur pada umumnya digunakan pada program yang kompleks dan memiliki subprogram. Seperti layaknya sebuah bahasa pemrograman, Python juga memberikan fasilitas pembuatan prosedur yang

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

sangat bagus. Konsep fungsi dalam Python sama dengan bahasa pemrograman C/C++. Python menganggap prosedur dan fungsi adalah sesuatu yang sama, dalam artian cara mendeklarasikan prosedur dan fungsi adalah sama. Hanya bedanya, kalau fungsi mengembalikan suatu nilai setelah proses sedangkan prosedur tidak.

Dalam penggunaannya, sebuah prosedur dapat digunakan dengan memanggil atau mengetikkan nama prosedur tersebut. Terdapat dua jenis prosedur, yaitu prosedur dengan parameter dan prosedur tanpa parameter. Berikut format penulisan pendeklarasian prosedur yang memiliki parameter:

Prosedur dalam Python menggunakan kata kunci `def` (defenisi). Setelah `def` ada nama prosedur (**salam**) lalu diikuti dengan parameter yang diapit oleh tanda kurung dan diakhiri dengan tanda titik dua (`:`) yang menandakan awal pendefinisian tubuh dari prosedur yang terdiri dari statemen-statemen.

Bentuk Umum:

```
#membuat prosedur
def nama_prosedur(parameter):
    print("hello world")

#pemanggilan prosedur
nama_prosedur()
```

Setiap perintah yang berada dalam prosedur agak masuk kedalam (tidak sejajar dengan **def**). Pernyataan `def` mendefinisikan sebuah prosedur dengan nama prosedur. Nama prosedur tidak memiliki daftar argumen dan tidak meminta nilai kembalian. Pendefinisian nama prosedur diakhiri dengan tanda (`:`), kemudian diikuti oleh statemen `print` yang menjadi isi dari tubuh prosedur. Lalu untuk memanggil nama prosedur kita gunakan perintah `nama_prosedur()`.

E. Alat/Instrument/Aparatus/Bahan

1. Personal Computer (PC)
2. Aplikasi PyCharm

F. Keselamatan Kerja

1. Menggunakan alat dan bahan sesuai dengan fungsinya
2. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar.

3. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen.
4. Tidak bercanda atau makan dan minum saat praktik
5. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

G. Lembar Kerja

a) Prosedur tanpa parameter

Salinlah kodingan dibawah ini

```
def salam():  
    print("hello world")  
  
salam()
```

Jalankan program diatas, maka akan menghasilkan :

```
hello world  
  
Process finished with exit code 0
```

b) Prosedur tanpa parameter menggunakan inputan

Salinlah kodingan dibawah ini

```
def biodata():  
    nama = input("Masukan Nama anda: ")  
    umur = input("Masukan Umur anda: ")  
  
    print("Nama anda ",nama,"dan umur anda  
    ",umur)
```

Jalankan program diatas, maka akan menghasilkan :

```
Masukan Nama anda: Toni  
Masukan Umur anda: 18  
Nama anda Toni dan umur anda 18  
  
Process finished with exit code 0
```

c) Prosedur menggunakan parameter

Salinlah kodingan dibawah ini

```
def salam(ucapan):  
    print(ucapan)  
  
salam("selamat pagi")
```

Dari program diatas, ucapan merukan variabel parameter dari prosedur sedangkan selamat pagi adalah nilai parameter dari prosedur. Setelah dijalankan maka akan menghasilkan:

```
selamat pagi  
  
Process finished with exit code 0
```

d) Menghitung Luas Persegi Panjang menggunakan Prosedur

Salinlah kodingan dibawah ini

```
def luas_persegipanjang(panjang, lebar):  
    luas = panjang * lebar  
    print("Luas Persegi Panjang adalah %2.f" %  
luas)  
  
luas_persegipanjang(4,6)
```

Dari kodingan diatas, untuk memisahkan satu parameter dengan parameter yang lainnya menggunakan tanda baca koma (.). Setelah dijalankan maka akan menghasilkan:

```
Luas Persegi Panjang adalah 24  
  
Process finished with exit code 0
```

e) Menghitung Luas Persegi Panjang Menggunakan Prosedur yang Menggunakan Inputan

Salinlah kodingan dibawah ini

```
def luas_persegipanjang():  
    panjang = float(input("panjang: "))  
    lebar = float(input("lebar: "))  
  
    luas = panjang * lebar  
  
    print("Luas Persegi Panjang adalah %2.f" % luas)
```

Jalankan program diatas, maka akan menghasilkan :

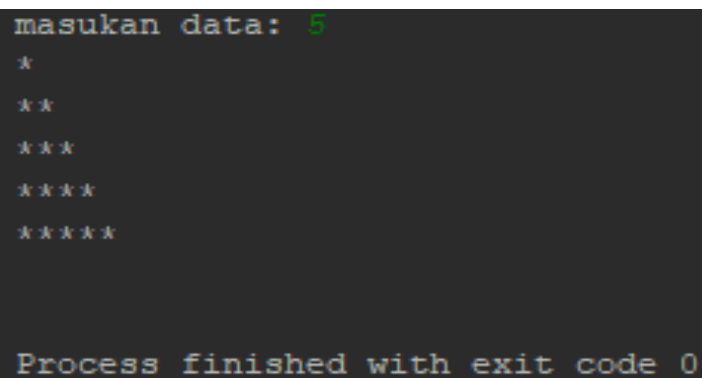
```
panjang: 5  
lebar: 7  
Luas Persegi Panjang adalah 35  
  
Process finished with exit code 0
```

f) Membuat Perulangan Segitiga Siku-Siku Menggunakan Prosedur

Salinlah kodingan dibawah ini

```
def perulangan(x):  
    string = ""  
    bar = 1  
  
    while bar <= x:  
        kol = bar  
        while kol > 0:  
            string = string + "*"   
            kol=kol - 1  
  
        string = string + "\n"  
        bar=bar + 1  
    print(string)  
  
perulangan(float(input("masukan data: ")))
```

Jalankan program diatas, maka akan menghasilkan :



```
masukan data: 5  
*  
**  
***  
****  
*****  
  
Process finished with exit code 0
```


g) Membuat Percabangan Menggunakan Prosedur

Salinlah kodingan dibawah ini

```
def percabangan():
    cuaca = input("Bagaimana Cuaca Hari Ini: ")

    if cuaca == "panas":
        print("Hari ini Panas")
        print("Perbanyak minum air")
    else:
        print("Hari ini dingin")
        print("Gunakan pakaian Hangat")

    print("\nNikmati Harimu :)")

percabangan()
```

Jalankan program diatas, maka akan menghasilkan :

```
Bagaimana Cuaca Hari Ini: dingin
Hari ini dingin
Gunakan pakaian Hangat

Nikmati Harimu :)


Process finished with exit code 0
```

H. Tugas Praktikum

1. Buatlah sebuah program untuk menghitung:
 - a. Luas Persegi dengan Prosedur menggunakan parameter yang berisi inputan.
 - b. Luas Segitiga dengan Prosedur menggunakan parameter yang berisi inputan.
 - c. Luas Lingkaran dengan Prosedur menggunakan parameter yang berisi inputan.

PETUNJUK PRAKTIKUM #15

FUNCTION

 INSTIKI <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 13	Function	
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021
			100 menit
			Hal 1 dari 141

A. Capaian Pembelajaran Mata Kuliah

1. Mampu mendefinisikan kebutuhan pengguna atau pasar terhadap kinerja (menganalisis, mengevaluasi dan mengembangkan) algoritma/metode berbasis komputer.
2. Mampu menelaah cara kerja sistem komputer dan mampu merancang dan mengembangkan berbagai algoritma/metode untuk memecahkan masalah.

B. Sub Capaian Pembelajaran Mata Kuliah

Mahasiswa mampu mengoperasikan sub program menjadi sebuah fungsi dengan benar pada Bahasa pemrograman.

C. Tujuan

1. Mahasiswa mampu **menjelaskan** struktur dasar fungsi, mengoperasikan fungsi ke dalam contoh-contoh masalah sederhana.
2. Mahasiswa mampu **mengimplementasikan** algoritma dalam bentuk pseudocode dan flowchart untuk menyelesaikan masalah dengan fungsi.

D. Dasar Teori

Pada dasarnya fungsi atau *function* maupun prosedur adalah sama. Fungsi maupun prosedur sama-sama merupakan subprogram. Selain itu, fungsi dan prosedur dapat digunakan atau dipanggil berkali-kali atau berulang-ulang dalam sebuah program. Namun, fungsi lebih cenderung digunakan untuk membuat perintah-perintah yang bersifat perhitungan. Selain itu, perbedaan antara fungsi dan prosedur adalah fungsi memiliki pengembalian nilai, sehingga pada saat dipanggil, sebuah fungsi dapat langsung digunakan untuk mengisi ekspresi yang ada. Keuntungan menggunakan fungsi :

1. Program besar dapat di pisah-pisah menjadi program-program kecil melalui fungsi.
2. Kemudahan dalam mencari kesalahan-kesalahan karena alur logika jelas dan kesalahan dapat dilokalisasi dalam suatu modul tertentu.
3. Memperbaiki atau memodifikasi program dapat dilakukan pada suatu modul tertentu saja
4. tanpa mengganggu keseluruhan program.
5. Dapat digunakan kembali (Reusability) oleh program atau fungsi lain.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

6. Meminimalkan penulisan perintah yang sama.

Kategori Fungsi

Standard Library Function adalah fungsi-fungsi yang telah disediakan oleh Interpreter Python dalam file-file atau librarynya. Misalnya: `raw_input()`, `input()`, `print()`, `open()`, `len()`, `max()`, `min()`, `abs()` dll.

1. Fungsi Yang Mengembalikan

Fungsi yang mengembalikan adalah menggunakan kembali hasil proses untuk proses berikutnya. Cara mengembalikan nilai adalah menggunakan kata kunci **return** lalu diikuti dengan nilai yang akan dikembalikan.

Bentuk Umum:

```
def nama_fungsi(parameter):  
    return  
  
nama_fungsi()
```

Statemen **return** dapat diletakkan di bagian mana saja dalam tubuh fungsi. Statemen **return** menandakan akhir dari pemanggilan fungsi dan akan mengirimkan suatu nilai balik kepada program yang memanggil fungsi tersebut. Statemen **return** bersifat opsional, artinya jika sebuah fungsi tidak memiliki statemen **return**, maka sebuah fungsi tidak akan mengembalikan suatu nilai apapun.

Variabel Global dan Variabel Lokal

Variabel Global adalah variabel yang bisa diakses dari semua fungsi, sedangkan variabel local hanya bisa diakses didalam fungsi tempat ia berada saja. Pada Python, urutan pengaksesan variabel (scope) dikenal dengan sebutan **LGB (Local, Global, and Build-in)**. Jadi program

python mulai mencari variabel local terlebih dahulu, jika variabel local ada maka itu yang digunakan terlebih dahulu. Tapi jika variabel local tidak ditemukan, maka pencarian akan terus lanjut ke Global dan Build-in. Variabel Build-in adalah variabel yang sudah ada didalam Python. Untuk lebih jelasnya bisa dilihat dari program dibawah ini.

Bentuk Umum:

```
#variabel global
nama = ""

def data_diri():
    #variabel lokal
    nama = ""
    umur = ""

    #mengakses variabel lokal
    print("Nama Peserta kedua : %s" %
nama)

    print("Umur : %d" % umur)

#mengakses variabel global
print("Nama Peserta pertama : %s" % nama)
#memanggil variabel local
data_diri()
```

Perhatikanlah variabel **nama** yang berada didalam fungsi **data_diri()** dan diluar fungsi **data_diri()**. Variabel nama yang berada di dalam fungsi **data_diri()** adalah variabel local. Jadi, saat kita memanggil fungsi **data_diri()** maka nilai yang akan tampil adalah nilai yang ada didalam fungsi **data_diri()**. Karena Python mulai mencari dari Local, ke Global, dan Build-in, maka yang ditampilkan pertama kali adalah variabel lokal.

E. Alat/Instrument/Aparatus/Bahan

1. Personal Computer (PC)

2. Aplikasi PyCharm

F. Keselamatan Kerja

1. Menggunakan alat dan bahan sesuai dengan fungsinya
2. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar.
3. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen.
4. Tidak bercanda atau makan dan minum saat praktik
5. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

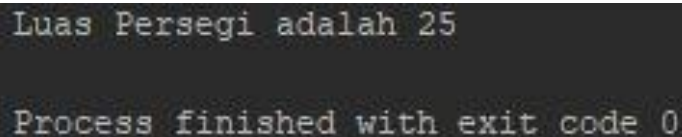
G. Lembar Kerja

a) Fungsi Yang Mengembalikan

Salinlah kodingan dibawah ini

```
def luas_persegi(sisi):  
    luas = sisi * sisi  
    return luas  
  
print("Luas Persegi Panjang adalah %d" % luas_persegi(5))
```

Jalankan program diatas, maka akan menghasilkan :



```
Luas Persegi adalah 25  
Process finished with exit code 0
```

Pada fungsi **luas_persegi()** kita melakukan **print** pada saat pemanggilannya. Jadi, fungsi **luas_persegi()** akan bernilai sesuai dengan hasil yang dikembalikan. Sehingga kita dapat memanfaatkan untuk menghitung volume persegi.

Salinlah Koding dibawah ini

```
def luas_persegi(sisi):  
    luas = sisi * sisi  
    return luas  
  
def volume_persegi(sisi):  
    volume = luas_persegi(sisi) * sisi  
    return volume  
  
print("Luas Persegi adalah %d" % luas_persegi(5))  
print("Volume Persegi adalah %d" % volume_persegi(5))
```

Jalankan program diatas, maka akan menghasilkan

```
Luas Persegi adalah 25  
Volume Persegi adalah 125  
  
Process finished with exit code 0
```


b) Penggunaan Fungsi Variabel Global dan Variabel Global

Salinlah kodingan dibawah ini

```
#variabel Global
nama = "Toni"
umur = 20

def data_diri():
    #variabel lokal
    nama = "Tina"
    umur = 20

    #mengakses variabel lokal
    print("Nama Peserta Pertama : %s" % nama)
    print("Umur : %d" % umur)

#mengakses variabel global
print("Nama peserta Kedua : %s" % nama)
print("Umur : %d" % umur)

#memanggil variabel lokal
data_diri()
```

Jalankan program diatas, maka akan menghasilkan :

```
Nama Peserta pertama : Toni
Umur : 19
Nama Peserta kedua : Tina
Umur : 20


Process finished with exit code 0
```

H. Tugas Praktikum

1. Buatlah sebuah program untuk menghitung:
 - a. Volume balok dengan fungsi dikembalikan.
 - b. Volume Lingkaran dengan fungsi dikembalikan.
 - c. Volume Tabung dengan fungsi dikembalikan.
- 2

PETUNJUK PRAKTIKUM #16

UAS

	Fakultas Teknologi dan Informatika Program Studi Teknik Informatika		
	ALGORITMA DAN LOGIKA INFORMATIKA		
	Pertemuan : 16	UAS	100 menit
	Kode MK: BTIW-001	Rev : 00	Tgl : 23/08/2021 Hal 1 dari 141

- A. Capaian Pembelajaran Mata Kuliah**
- B. Sub Capaian Pembelajaran Mata Kuliah**
- C. Tujuan**
- D. Dasar Teori**
- E. Alat/Instrument/Aparatus/Bahan**
- F. Keselamatan Kerja**
- G. Lembar Kerja**
- H. Tugas Praktikum**

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------



FAKULTAS TEKNOLOGI DAN INFORMATIKA
PROGRAM STUDI TEKNIK INFORMATIKA

Jl. Tukad Pakerisan No. 97 Denpasar