

MODUL MATA KULIAH

BAHASA PEMROGRAMAN DASAR

PG168 – 3 SKS



**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

**JAKARTA
SEPTEMBER 2021**

TIM Penyusun

Agus Umar Hamdani, M.Kom
Tri Ika Jaya Kusumawati, M.Kom¹



MODUL PERKULIAHAN #3

OPERASI DASAR DALAM PYTHON

Capaian Pembelajaran	:	Mahasiswa Mampu: <ol style="list-style-type: none">1. Mengenal dan memahami tipe-tipe operator dasar pada bahasa pemrograman Python.2. Mengenal dan memahami derajat operator.
Sub Pokok Bahasan	:	<ol style="list-style-type: none">1. Operator dasar dalam bahasa pemrograman Python, antara lain : aritmatika, perbandingan, penugasan, keanggotaan, identitas, logika, bitwise.2. Derajat operator atau prioritas eksekusi operator dalam bahasa Python.
Daftar Pustaka	:	<ol style="list-style-type: none">1. Zarman, Wendi dan Wicaksono, Mochamad Fajar. <i>"Implementasi Algoritma dalam bahasa Python"</i>. Edisi Pertama. Bandung : Penerbit Informatika. 2020.2. Kurniawati, Arik. <i>"Algoritma dan Pemrograman menggunakan Python"</i>. Edisi Pertama. Yogyakarta : Depublish. 2016.3. Ismah. <i>"Pemrograman Komputer Dasar-dasar Python"</i>. Jakarta : Fakultas Ilmu Pendidikan Universitas Muhammadiyah Jakarta. 2017.4. Irfani, M. Haviz dan Dafid. <i>"Modul Praktikum Dasar Pemrograman dengan bahasa Python"</i>. Palembang : Sekolah Tinggi Manajemen Informatika Global Informatika Multidata Palembang. 2016.5. Fikri, Rijalul. <i>"Praktikum Algoritma dan Pemrograman Komputer"</i>. Surabaya : Program Studi Teknik Komputer dan Telematika Jurusan Teknik Elektro Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember. 2010.6. Sugiana, Owo. <i>Membuat Aplikasi Bisnis menggunakan bahasa Python dan database berbasis SQL</i>. Jakarta. 2003.7. Septian, Ridwan Fadjar. <i>Buku Serial Open Source Belajar Pemrograman Python Dasar</i>. Versi 1. Bandung : POSS-UPI. 2013.8. Hendri. <i>Cepat Mahir Python</i>. Ilmu Komputer.com. 2003.9. Herho, Sandy H.S. <i>Tutorial Pemrograman Python 2 Untuk pemula</i>. Bandung: WCPL Press. 2017.10. <i>Welcome to Python.org</i> (diakses pada 29 September 2021 pukul : 21.00 WIB)11. <i>Programiz.com</i>. Python Operators (https://www.programiz.com/python-programming/operators diakses pada 29 September 2021 pukul 21.32 WIB)

PRAKTIKUM 3

OPERASI DASAR DALAM PEMROGRAMAN PYTHON

2.1. Teori Singkat

Operator di dalam Python adalah symbol khusus yang berfungsi untuk menjalankan suatu operasi tertentu, baik operasi aritmatika maupun operasi logika. Sedangkan nilai yang dioperasikan oleh operator disebut dengan operand [Programiz.com, 2021]. Jenis-jenis operator dalam bahasa Python terbagi atas 7 (tujuh) macam, yaitu : operator aritmatika, operator perbandingan, operator penugasan, operator logika, operator keanggotaan (membership), operator identitas dan operator bitwise.

2.2. Jenis-jenis Operator

a. Operator Aritmatika (Arithmetic Operators)

Tabel 2.3 Operator Aritmatika dalam Python

Operator	Contoh	Penjelasan
Penjumlahan +	$1 + 3 = 4$	Menjumlahkan nilai dari masing-masing operan atau bilangan
Pengurangan -	$4 - 1 = 3$	Mengurangi nilai operan di sebelah kiri menggunakan operan di sebelah kanan
Perkalian *	$2 * 4 = 8$	Mengalikan operan/bilangan
Pembagian /	$10 / 5 = 2$	Untuk membagi operan di sebelah kiri menggunakan operan di sebelah kanan
Sisa Bagi %	$11 \% 2 = 1$	Mendapatkan sisa pembagian dari operan di sebelah kiri operator ketika dibagi oleh operan di sebelah kanan
Pangkat **	$8 ** 2 = 64$	Memangkatkan operan disebelah kiri operator dengan operan di sebelah kanan operator
Pembagian Bulat //	$10 // 3 = 3$	Sama seperti pembagian. Hanya saja angka dibelakang koma dihilangkan

b. Operator Perbandingan

Operator perbandingan (comparison operators) digunakan untuk membandingkan suatu nilai dari masing-masing operan.

Tabel 2.4 Operator Perbandingan

Operator	Contoh	Penjelasan
Sama dengan ==	1 == 1	bernilai True Jika masing-masing operan memiliki nilai yang sama, maka kondisi bernilai benar atau True.
Tidak sama dengan !=	2 != 2	bernilai False Akan menghasilkan nilai kebalikan dari kondisi sebenarnya.
Tidak sama dengan <>	2 <> 2	bernilai False Akan menghasilkan nilai kebalikan dari kondisi sebenarnya.
Lebih besar dari >	5 > 3	bernilai True Jika nilai operan kiri lebih besar dari nilai operan kanan, maka kondisi menjadi benar.
Lebih kecil dari <	5 < 3	bernilai True Jika nilai operan kiri lebih kecil dari nilai operan kanan, maka kondisi menjadi benar.
Lebih besar atau sama dengan >=	5 >= 3	bernilai True Jika nilai operan kiri lebih besar dari nilai operan kanan, atau sama, maka kondisi menjadi benar.
Lebih kecil atau sama dengan <=	5 <= 3	bernilai True Jika nilai operan kiri lebih kecil dari nilai operan kanan, atau sama, maka kondisi menjadi benar.

c. Operator Penugasan

Operator penugasan digunakan untuk memberikan atau memodifikasi nilai ke dalam sebuah variabel.

Table 2.5 Operator Penugasan

Operator	Contoh	Penjelasan
Sama dengan =	a = 1	Memberikan nilai di kanan ke dalam variabel yang berada di sebelah kiri.
Tambah sama dengan +=	a += 2	Memberikan nilai variabel dengan nilai variabel itu sendiri ditambah dengan nilai di sebelah kanan.
Kurang sama dengan -=	a -= 2	Memberikan nilai variabel dengan nilai variabel itu sendiri dikurangi dengan nilai di sebelah kanan.
Kali sama dengan *=	a *= 2	Memberikan nilai variabel dengan nilai variabel itu sendiri dikali dengan nilai di sebelah kanan.
Bagi sama dengan /=	a /= 4	Memberikan nilai variabel dengan nilai variabel itu sendiri dibagi dengan nilai di sebelah kanan.

Sisa bagi sama dengan %=	a %= 3	Memberikan nilai variabel dengan nilai variabel itu sendiri dibagi dengan nilai di sebelah kanan. Yang diambil nantinya adalah sisa baginya.
Pangkat sama dengan **=	a **= 3	Memberikan nilai variabel dengan nilai variabel itu sendiri dipangkatkan dengan nilai di sebelah kanan.
Pembagian bulat sama dengan //=	a //= 3	Membagi bulat operan sebelah kiri operator dengan operan sebelah kanan operator kemudian hasilnya diisikan ke operan sebelah kiri.

d. Operator Logika

Operator logika ini sering juga disebut sebagai **operator boolean**. **Operator logika** adalah operator yang digunakan untuk membuat kesimpulan logis dari 2 kondisi boolean: **true** atau **false**. Dalam bahasa Python terdapat 3 operator logika:

Table 2.7 Operator Logika

Operator	Penjelasan	Contoh	Hasil
and	True jika kedua operand bernilai True	True and True	True
or	True jika salah satu operand bernilai True	True or False	True
not	True jika operand bernilai False	not False	True

e. Operator Bitwise

Bitwise adalah operator khusus untuk menangani operasi logika bilangan biner dalam bentuk bit.

Bilangan biner sendiri merupakan jenis bilangan yang hanya terdiri dari 2 jenis angka, yakni 0 dan 1. Jika nilai asal yang dipakai bukan bilangan biner, akan dikonversi secara otomatis menjadi bilangan biner. Misalnya 7 desimal = 0111 dalam bilangan biner.

Pada penerapannya, operator bitwise tidak terlalu sering dipakai, kecuali anda sedang membuat program yang harus memproses bit-bit komputer. Selain itu operator ini cukup rumit dan harus memiliki pemahaman tentang sistem bilangan biner. Dalam bahasan kali ini saya menganggap anda sudah paham beda antara bilangan biner (basis 2) dan bilangan desimal (basis 10).

Bahasa Python mendukung 6 jenis operator bitwise:

Table 2.8 Operator Bitwise

Operator	Nama	Contoh	Biner	Hasil (biner)	Hasil (decimal)
&	And	10 & 12	1010 & 1100	1000	8
	Or	10 12	1010 1100	1110	14
^	Xor	10 ^ 12	1010 ^ 1100	0110	6
~	Not	~ 10	~1010	0101	-11 (two complement)
<<	Left shift	10 << 1	1010 << 1	10100	20
>>	Right shift	10 >> 1	1010 >> 1	101	5

f. Operator Keanggotaan

Python adalah bahasa pemrograman yang terbilang unik, ia memiliki operator khusus atau juga sering dinamakan sebagai operator spesial. Dinamakan spesial **karena memang operator ini hanya ada di Python** dan tidak ada di bahasa pemrograman lainnya.

Di antara operator spesial tersebut adalah **operator keanggotaan**. Atau di dalam bahasa inggris ia dinamakan *membership operator*.

Operator keanggotaan dalam python **hanya memiliki dua varian**, yaitu in dan not in. Berikut ini tabel lebih lengkapnya:

Table 2.9 Operator Keanggotaan

Operator	Penjelasan
in	Bernilai true jika suatu nilai ada di dalam <i>sequence</i>
not in	Bernilai false jika suatu nilai tidak ada di dalam <i>sequence</i>

g. Operator Identitas

Selain operator keanggotaan, python juga masih memiliki operator spesial lainnya: yaitu **operator identitas**. Operator ini **didefinisikan dengan is** dan **is not**.

Tugasnya adalah **untuk mengetahui apakah dua buah variabel** memiliki **nilai** yang sama dan **posisi** yang sama di dalam memori. Karena **tidak semua** nilai yang sama **memiliki** tempat / posisi yang sama di dalam memori.

Table 2.10 Operator Identitas

Simbol	Tugas
is	Bernilai true jika dua variabel bersifat identik baik dari segi nilai mau pun penempatan lokasi di memory
is not	Bernilai false jika dua variabel tidak identik baik dari segi nilai mau pun penempatan lokasi di memory

2.1.1 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.

2.3. Derajat Operator / Prioritas Eksekusi Operator dalam bahasa Python

Bahasa pemrograman harus memiliki aturan yang jelas untuk urutan ekspresi majemuk, seperti : $5 + 2 * 3$. Urutan formal prioritas untuk operator dalam bahasa Python dapat dilihat pada tabel 3.12. Operator dengan prioritas lebih tinggi akan dikerjakan terlebih dahulu daripada operator dengan prioritas lebih rendah, kecuali jika suatu ekspresi diberi tanda kurung. Oleh karena itu, pada contoh tersebut, operasi perkalian lebih diutamakan daripada operasi penambahan.

Contoh terdapat ekspresi : $x = 5 + 2 * 3$

Maka, Python akan mengevaluasi dan mengeksekusinya menjadi : $5 + (2 * 3)$ dengan hasil akhir untuk variabel $x = 11$. Tetapi, ekspresi yang diberikan tanda kurung $(5 + 2) * 3$ akan menghasilkan nilai $x = 21$, karena derajat atau prioritas tanda kurung lebih tinggi dari pada perkalian.

Operator dalam tipe yang sama biasanya dievaluasi dari kiri ke kanan, misalnya : $5 - 2 + 3$ memiliki hasil akhir adalah nilai 6. Pengecualian pada aturan ini adalah untuk operator unary dan eksponen (pangkat) dievaluasi dari kanan ke kiri.

Python memungkinkan penugasan berantai, seperti : $x = y = 0$, untuk menetapkan isi variabel ke nilai paling kanan. Python juga memungkinkan perangkaian operator perbandingan. Misalkan: ekspresi $1 \leq x + y \leq 10$ dievaluasi sebagai compound atau gabungan $(1 \leq x + y)$ dan $(x + y \leq 10)$, tetapi tanpa menghitung nilai antara $x + y$ dua kali.

Table 2.11 Derajat Operator di Python

No	Tipe	Simbol
1	Member Access	Expr.member
2	Pemanggilan pada fungsi atau method	Expr(...),Expr[...]
3	Eksponen	**
4	Unary operator	+expr, -expr, ~expr
5	Perkalian dan Pembagian	*, /, //, %
6	Penjumlahan, pengurangan	+, -
7	Bitwise shifting (operasi pergesaran bit)	<<, >>
8	Bitwise AND	&
9	Bitwise XOR	^
10	Bitwise OR	
11	Perbandingan	Is, is not, ==, !=, <, <=, >, >=, in, not in
12	Operator logika NOT	Not expr

13	Operator logika AND	AND
14	Operator logika OR	OR
15	Kondisi	value1 IF Condition Else value-2
16	Penugasan (Assignments)	=, +=, -=, *=

Tabel 2.11 diatas sudah diurutkan berdasarkan prioritas tertinggi ke prioritas terendah. Kata "expr" pada tabel tersebut bisa berarti variabel atau hasil dari ekspresi sebelumnya. Semua operator pada tabel 2.11 yang tidak diberi kata "expr" berarti binary operator yang harus memiliki dua operand.

Praktikum

Langkah-langkah Praktikum

1. Buka Editor Python (IDLE / Pycharm / VSCode).
2. Buatlah file baru dengan membuka menu File > New > Source File atau dengan shortcut Ctrl + N.
3. Tulislah Program 2.1 berikut ini :

Program 2.1 : Variable. Py

1. Tuliskan Program 2.1 berikut ini pada editor :

```

1 nama = 'Ahmad Budi'
2 usia = 24
3 sudah_menikah = True
4
5 print('nama:', nama)
6 print('usia:', usia)
7 print('sudah menikah:', sudah_menikah)
8
```

2. Simpan Program ini dengan nama variable.py
3. Jalankan program 2.1 di atas lalu tuliskan apa yang tercetak di layar.

4. Pada program 2.1, bagaimana kita mengetahui tipe data yang digunakan?
Lanjut kan program 2.1 ke dalam 2.2 untuk mengetahui tipe data yang digunakan.

Program 2.2 : Variable2. Py

1. Tuliskan program 2.2 berikut ini pada editor :

```
1 nama = 'Ahmad Budi'
2 usia = 24
3 sudah_menikah = True
4
5 print('nama:', nama)
6 print('usia:', usia)
7 print('sudah menikah:', sudah_menikah)
8
9 print(type(nama))
10 print(type(usia))
11 print(type(sudah_menikah))
12
```

2. Simpan program 2.2 dengan nama variable2.py
3. Jalankan program 2.2 di atas lalu tuliskan apa yang tercetak di layar.

4. Berdasarkan program 2.2 kita sudah mengetahui bagaimana cara memeriksa tipe data apakah yang digunakan dalam sebuah variable yang memiliki nilai atau isi. Berikutnya kita akan coba dengan cara penulisan variable lainnya pada program 2.3

Program 2.3 : Variable3. Py

1. Tuliskan program 2.3 berikut ini pada editor :

```

1 #proses memasukkan data ke dalam variabel
2 nama = "John Doe"
3 #proses mencetak variabel
4 print(nama)
5
6 #nilai dan tipe data dalam variabel dapat diubah
7 umur = 20 #nilai awal
8 print(umur) #mencetak nilai umur
9 type(umur) #mengecek tipe data umur
10 umur = "dua puluh satu" #nilai setelah diubah
11 print(umur) #mencetak nilai umur
12 type(umur) #mengecek tipe data umur
13
14 namaDepan = "Budi"
15 namaBelakang = "Susanto"
16 nama = namaDepan + " " + namaBelakang
17 umur = 22
18 hobi = "Berenang"
19 print("Biodata\n", nama, "\n", umur, "\n", hobi)
20
21 #contoh variabel lainya
22 inivariabel = "Halo"
23 ini_juga_variabel = "Hai"
24 _inivariabeljuga = "Hi"
25 inivariabel222 = "Bye"
26
27 panjang = 10
28 lebar = 5
29 luas = panjang * lebar
30 print(luas)
31

```

2. Simpan program 2.3 dengan nama variable3.py
3. Jalankan program 2.2 di atas lalu tuliskan apa yang tercetak di layar.

4. Berdasarkan program 2.3 analisa apakah ada yang kurang atau tidak tampil dari program tuliskan di bawah ini :

Program 2.4 : typedata1.py

1. Tuliskan program 2.4 berikut ini pada editor :

```

1 #tipe data Boolean
2 print(True)
3
4 #tipe data String
5 print("Ayo belajar Python")
6 print('Belajar Python Sangat Mudah')
7
8 #tipe data Integer
9 print(20)
10
11 #tipe data Float
12 print(3.14)
13
14 #tipe data Complex
15 print(5j)
16
17 #tipe data List
18 print([1,2,3,4,5])
19 print(["satu", "dua", "tiga"])
20
21 #tipe data Tuple
22 print((1,2,3,4,5))
23 print(("satu", "dua", "tiga"))
24
25 #tipe data Dictionary
26 print({"nama":"Budi", 'umur':20})
27 #tipe data Dictionary dimasukan ke dalam variabel biodata
28 biodata = {"nama":"Andi", 'umur':21} #proses inisialisasi variabel biodata
29 print(biodata) #proses pencetakan variabel biodata yang berisi tipe data Dictionary
30 print(type(biodata))
31 #fungsi untuk mengecek jenis tipe data. akan tampil <class 'dict'>
32 #yang berarti dict adalah tipe data dictionary
33

```

2. Simpan program 2.4 dengan nama typedata1.py
3. Jalankan program 2.4 di atas lalu tuliskan apa yang tercetak di layar.

4. Setelah menjalankan program 2.4, masih ada tipe data lainnya yang belum dibuat apa saja? Mari kita lanjutkan praktik pada program 2.5

Program 2.5 : typedata2.py

1. Tuliskan program 2.5 berikut ini pada editor :
2. Simpan program 2.5 dengan nama typedata2.py
3. Jalankan program 2.5 di atas lalu tuliskan apa yang tercetak di layar.

4. Apa yang perlu ditambahkan pada program tersebut? (silahkan diperbaiki)

Program 2.6 : typedata3.py

1. Tuliskan program 2.6 berikut ini pada editor :

```

1 # list kosong
2 list_kosong = []
3
4 # list yang berisi kumpulan string
5 list_buah = ['Pisang', 'Nanas', 'Melon', 'Durian']
6
7 # list yang berisi kumpulan integer
8 list_nilai = [80, 70, 90, 60]
9
10 # list campuran berbagai tipe data
11 list_jawaban = [150, 33.33, 'Presiden Sukarno', False]
12 |

```

2. Simpan program 2.6 dengan nama `tipedata3.py`
3. Jalankan program 2.6 di atas lalu tuliskan apa yang tercetak di layar.

4. Pada **Kode program 1**, kita lihat bahwa **sebuah list didefinisikan menggunakan tanda kurung siku ([])**. Kita juga saksikan bahwa list pada python, bisa berisi berbagaimacam tipe data. Bisa terdiri dari tipe data yang sejenis mau pun dari tipe data yang berbeda-beda. Lalu bagaimana cara menampilkan isi list, mari kita lakukan perubahan pada program ini ke program 2.7

Program 2.7 : `tipedata4.py`

1. Tuliskan program 2.7 berikut ini pada editor :

```

1 # list kosong
2 list_kosong = []
3
4 # list yang berisi kumpulan string
5 list_buah = ['Pisang', 'Nanas', 'Melon', 'Durian']
6
7 # list yang berisi kumpulan integer
8 list_nilai = [80, 70, 90, 60]
9
10 # list campuran berbagai tipe data
11 list_jawaban = [150, 33.33, 'Presiden Sukarno', False]
12
13
14 #cetak isi seluruh list
15 print('list_kosong:', list_kosong)
16 print('list_buah:', list_buah)
17 print('list_nilai:', list_nilai)
18 print('list_jawaban:', list_jawaban)
19 print("\n")
20
21 #cetak isi list tertentu
22 print(list_buah[0])
23 print(list_buah[2])
24 print(list_buah[1])
25 print(list_buah[3])
26 print("\n")
27
28 #cetak isi list menggunakan index negatif untuk mencetak
29 #dari belakang
30 print(list_buah[-1])
31 print(list_buah[-2])
32 print(list_buah[-3])
33 print(list_buah[-4])
34
35 #NB: yang perlu diperhatikan adalah: bahwa indeks negatif
36 #tidak dimulai dari 0, akan tetapi dimulai dari angka 1.
37 |

```

2. Simpan program 2.7 dengan nama `tipedata4.py` (program untuk tipe data list)
3. Jalankan program 2.7 di atas lalu tuliskan apa yang tercetak di layar.
4. Apa fungsi dari `print("\n")` pada program 2.7 ?

Program 2.8 : Operator1.py

1. Tuliskan program 2.7 berikut ini pada editor :

```

1 x = 10
2 y = 12
3
4 print('x berisi angka',x , 'desimal atau',bin(x), 'biner')
5 print('y berisi angka',y , 'desimal atau',bin(y), 'biner')
6
7 print('\n')
8
9 print('x & y   : ',x & y)
10 print('x | y   : ',x | y)
11 print('x ^ y   : ',x ^ y)
12 print('~x      : ',~x)
13 print('x << 1  : ',x << 1)
14 print('x >> 1  : ',x >> 1)
15

```

2. Simpan program 2.8 dengan nama Operator1.py
3. Jalankan program 2.8 (Operator Bitwise)di atas lalu tuliskan apa yang tercetak di layar.

4. Tuliskan maksud atau arti dari hasil program!

Program 2.9 : Operator2.py

1. Tuliskan program 2.9 berikut pada editor :

```

1 print('Hasil dari True and True   :', True and True)
2 print('Hasil dari True and False  :', True and False)
3 print('Hasil dari False and True   :', False and True)
4 print('Hasil dari False and False  :', False and False)
5
6 print('\n')
7
8 print('Hasil dari True or True     :', True or True)
9 print('Hasil dari True or False    :', True or False)
10 print('Hasil dari False or True    :', False or True)
11 print('Hasil dari False or False   :', False or False)
12
13 print('\n')
14
15 print('Hasil dari not True   :', not True)
16 print('Hasil dari not False :', not False)
17
18 #Contoh Operasi
19 hasil = (5 > 6) and (10 <= 8)
20 print(hasil)
21
22 hasil = ('budiluhur' == 'budiluhur') or (10 <= 8)
23 print(hasil)
24
25 hasil = not (10 < 10)
26 print(hasil)
27
28 hasil = ('budiluhur' == 'budiluhur') and (10 <= 8) or (1 != 1)
29 print(hasil)
30

```

2. Simpan program 2.9 dengan nama Operator2.py
3. Jalankan program 2.9 (operator logika) di atas lalu tuliskan apa yang tercetak di layar.

Program 2.10 : Operator3.py

1. Tuliskan program 2.10 berikut pada editor :


```

1 a = 5
2 b = 5
3 c = 6
4 print('a is b :', a is b)
5 print('a is c :', a is c)
6 print('a is not c :', a is not c)
7 print('\n')
8
9 i = 'Budi Luhur'
10 j = 'Budi Luhur'
11 print('i is j :', i is j)
12 print('i is not j :', i is not j)
13 print('\n');
14
15 x = ['a', 'b', 'c']
16 y = ['a', 'b', 'c']
17 print('x is y :', x is y)
18 print('x is not y :', x is not y)
19

```

2. Simpan program 2.10 dengan nama Operator3.py (program untuk tipe data booleans)
3. Jalankan program 2.10 di atas lalu tuliskan apa yang tercetak di layar.

Program 2.11 : Operator4.py

1. Tuliskan program 2.10 berikut pada editor :

```

1 foo = 'BudiLuhur'
2 print('foo :', foo)
3 print('\i\' in foo      :', 'i' in foo)
4 print('\k\' not in foo :', 'k' not in foo)
5 print('\d\' not in foo :', 'd' not in foo)
6 print('\n')
7
8
9 bar = ['a', 'b', 'c']
10 print('bar :', bar)
11 print('\a\' in bar      :', 'a' in bar)
12 print('\a\' not in bar :', 'a' not in bar)
13 print('\d\' not in bar :', 'd' not in bar)
14 print('\n')
15
16 baz = (12, 43, 102, 55)
17 print('baz :', baz)
18 print('102 in baz      :', 102 in baz)
19 print('102 not in baz :', 102 not in baz)
20 print('35 not in baz :', 35 in baz)
21

```

2. Simpan program 2.11 dengan nama Operator4.py
3. Jalankan program 2.11 di atas lalu tuliskan apa yang tercetak di layar.

4. Jelaskan logika dari program di atas :

Program 2.12 : Operator5.py

1. Tuliskan program 2.12 berikut pada editor :

```
1 print (11 == 1)
2
3 print ("CodeSaya" != "Python")
4
5 print (5 >= 0)
6
7 print ("HODOR" == "hodor")
8
9 print (100 < (10*10))
10
```

2. Simpan program 2.12 dengan nama Operator5.py
3. Jalankan program 2.12 di atas lalu tuliskan apa yang tercetak di layar.

4. Jelaskan logika dari program di atas :

Program 2.13 : Operator6.py

1. Tuliskan program 2.13 berikut pada editor :

```

1 # penugasan pertama
2 a = 10
3 print('a = 10 -> ', a)
4 a += 5
5 print('a += 5 -> ', a)
6 a -= 3
7 print('a -= 3 -> ', a)
8 a *= 6
9 print('a *= 6 -> ', a)
10 a /= 8
11 print('a /= 8 -> ', a)
12
13 # karena a jadi float, kita ubah lagi menjadi integer
14 a = int(a)
15
16 a %= 9
17 print('a %= 9 -> ', a)
18 a //= 6
19 print('a //= 6 -> ', a)
20 a **= 1
21 print('a **= 1 -> ', a)
22
23
24 a &= 2
25 print('a &= 2 -> ', a)
26 a |= 3
27 print('a |= 3 -> ', a)
28 a ^= 4
29 print('a ^= 4 -> ', a)
30 a >>= 4
31 print('a >>= 4 -> ', a)
32 a <<= 2
33 print('a <<= 2 -> ', a)
34

```

2. Simpan program 2.13 dengan nama Operator6.py
3. Jalankan program 2.13 di atas lalu tuliskan apa yang tercetak di layar.

4. Jelaskan logika dari program di atas :

Latihan

Setelah membuat program di atas (praktik variable, tipe data dan operator), buatlah duplikasi dari contoh praktik sebagai berikut :

1. Buat sebuah program sederhana dengan python yang jika dirunning akan menghasilkan keluaran seperti ini :

Data Nilai Mahasiswa BPD

```
=====
Nama          : Ahmad Budi
Fakultas      : FTI
Matakuliah    : BPD
Nilai Tugas   : 65
Nilai UTS     : 75
Nilai UAS     : 80
=====
```

```
Nilai Akhir   : 74.0
```

2. Buatlah program sederhana dengan python yang jika dijalankan akan menghasilkan keluaran seperti ini :

```
Profileku
=====
Nama      : Ahmad Budi
<class 'str'>
Fakultas  : FTI
<class 'str'>
Matakuliah : BPD
<class 'str'>
Nilai Akhir : 80
<class 'int'>
Profil    : ('anak ke 2', 165, 60.25)
<class 'tuple'>
```

3. Buatlah program sederhana dengan python yang jika dijalankan akan menghasilkan keluaran seperti ini :

```
Profileku
=====
Nama          : Ahmad Budi
Fakultas      : FTI
Matakuliah    : BPD
Nilai Akhir   : 80
Profil        : ('anak ke 2', 165, 60.25)
Hobi          : ('nonton', 'travelling', 'kuliner')
Kampus        : Budi Luhur

Membandingkan isi variable

nama is kampus :      : False
profil is not hobi :    : True
```

Tugas Mandiri

Kerjakan soal-soal berikut :

1. Diketahui sebuah segitiga memiliki panjang alas = 8 cm dan tinggi = 5 cm. Buatlah sebuah program dalam python untuk menghitung dan menampilkan

luas dari segitiga tersebut!

2. Diketahui sebuah bangun ruang berbentuk bola dan memiliki diameter 15 cm. Buatlah sebuah program dalam Bahasa python untuk menghitung dan menampilkan isi (volume) dari bangun ruang bola tersebut!
3. Diketahui sebuah segitiga siku-siku dengan panjang sisi alas 4 cm dan sisi tinggi 5 cm. Dengan menggunakan rumus Phitagoras, buatlah sebuah program Bahasa python untuk menghitung sisi miring segitiga tersebut dan menampilkannya di layar!



FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS BUDI LUHUR

Jl. Raya Ciledug, Petukangan Utara, Pesanggrahan

Jakarta Selatan, 12260

Telp: 021-5853753 Fax : 021-5853752

<http://fti.budiluhur.ac.id>