

**LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN
KOMPUTER
MODUL 4
OPERATOR**

Laporan ini disusun untuk memenuhi Tugas Mata Kuliah
Praktikum Basis Data



Disusun Oleh :

AHSAN MAULANA RIZQI

104062400071

S1 BD-04-B

**PROGRAM STUDI S1 BISNIS DIGITAL
TELKOM UNIVERSITY PURWOKERTO
2025**

1. Dasar Teori

1.1. Pengantar Operator

Operator adalah simbol yang memerintahkan komputer untuk melakukan satu tugas sederhana. *Operand* adalah data, ekspresi, atau nilai yang menjadi dasar tindakan atau pekerjaan operator. Operator Python adalah simbol khusus yang melakukan operasi tertentu pada nilai atau variabel yang disebut *operand*, seperti operator aritmetika (+), operator perbandingan (==), operator logika (*and*), operator keanggotaan (*in*), operator penugasan (*+=*), dan operator identitas (*is*). Beberapa operator yang terdapat pada bahasa pemrograman python adalah:

- a) Operator Aritmetika (*Arithmetic Operators*).
- b) Operator Perbandingan (*Comparison / Relational Operators*).
- c) Operator Penugasan (*Assignment Operators*).
- d) Operator Logika (*Logical Operators*).
- e) Operator Keanggotaan (*Membership Operators*).
- f) Operator Identitas (*Identity Operators*).

1.2. Jenis-Jenis Operator

- a) Operator Aritmetika (*Arithmetic Operators*).

Operator aritmetika adalah operator umum yang biasa digunakan untuk melakukan perhitungan aritmetika. Operator ini terdiri dari:

Table 1. Jenis-jenis operator aritmetika

<i>Operator</i>	<i>Symbol</i>	Contoh	Keterangan
Penjumlahan	+	$2 + 5 = 7$	Menjumlahkan nilai dari masing masing <i>operand</i>
Pengurangan	-	$5 - 2 = 3$	Mengurangi nilai <i>operand</i> sebelah kiri dengan nilai <i>operand</i> sebelah kanan
Perkalian	*	$2 * 5 = 10$	Mengalikan nilai <i>operand</i>
Pembagian	/	$10 / 5 = 2$	Membagi nilai <i>operand</i> di sebelah kiri dengan nilai <i>operand</i> di sebelah kanan
Sisa bagi	%	$11 \% 2 = 1$	Sisa bagi dari nilai <i>operand</i> di sebelah kiri ketika dibagi dengan nilai <i>operand</i> di sebelah kanan

Pangkat	**	$8 ** 2 = 64$	Memangkatkan nilai <i>operand</i> sebelah kiri dengan nilai <i>operand</i> di sebelah kanan
Pembagian Bulat	//	$10 // 3 = 3$	Seperti halnya operator pembagian tetapi dibulatkan ke bawah atau angka di belakang koma dihilangkan

b) Operator Perbandingan (*Comparison / Relational Operators*).

Perbandingan adalah operator yang digunakan untuk membandingkan nilai dari masing-masing operand. Hasil dari operasi menggunakan operator perbandingan hanya ada 2 yaitu *True* dan *False*. Operator ini terdiri dari:

Table 2. jenis-jenis operator perbandingan

<i>Operator</i>	<i>Symbol</i>	Contoh	Keterangan
Kurang dari	<	$2 < 5$ (<i>False</i>)	Bernilai <i>False</i> karena nilai <i>operand</i> di sebelah kiri lebih besar dari nilai <i>operand</i> di sebelah kanan sedangkan operator yang digunakan adalah <
Lebih dari	>	$2 > 5$ (<i>True</i>)	Bernilai <i>True</i> karena nilai <i>operand</i> di sebelah kiri lebih besar dari nilai <i>operand</i> di sebelah kanan dan operator yang digunakan adalah >
Kurang dari atau sama dengan	<=	$2 <= 5$ (<i>True</i>)	Bernilai <i>True</i> karena nilai <i>operand</i> di sebelah kiri kurang dari atau sama dengan nilai <i>operand</i> di sebelah kanan dan operator yang digunakan <=
Lebih dari atau sama dengan	>=	$5 >= 10$ (<i>False</i>)	Bernilai <i>False</i> karena nilai <i>operand</i> di sebelah kiri lebih kecil dari atau sama dengan nilai <i>operand</i> di sebelah kanan dan operator yang digunakan >=
Sama dengan	==	$10 == 5$	Bernilai <i>False</i> karena nilai operand di sebelah kiri tidak sama dengan nilai <i>operand</i> di sebelah kanan dan operator yang digunakan adalah ==

Tidak sama dengan	<code>!=</code>	<code>8 != 2</code>	Bernilai True karena nilai operand di sebelah kiri tidak sama dengan nilai operand di sebelah kanan dan operator yang digunakan adalah <code>!=</code>
-------------------	-----------------	---------------------	--

c) Operator Penugasan (*Assignment Operators*).

Operator penugasan adalah operator yang digunakan untuk memberikan atau memodifikasi sebuah nilai dalam sebuah variabel. Operator ini terdiri dari:

Table 3. jenis-jenis operator penugasan

<i>Operator</i>	<i>Symbol</i>	Contoh	Keterangan
Sama dengan	<code>=</code>	<code>a = 5</code>	Memberikan nilai 5 pada variabel a
Tambah sama dengan	<code>+=</code>	<code>a += 5</code>	Memberikan nilai kepada variabel a dengan nilai variabel itu sendiri ditambah dengan nilai 5 atau (<code>a = a + 5</code>)
Kurang sama dengan	<code>-=</code>	<code>a -= 5</code>	Memberikan nilai kepada variabel a dengan nilai variabel itu sendiri dikurang dengan nilai 5 atau (<code>a = a - 5</code>)
Kali sama dengan	<code>*=</code>	<code>a *= 5</code>	Memberikan nilai kepada variabel a dengan nilai variabel itu sendiri dikali dengan nilai 10 atau (<code>a = a * 10</code>)
Bagi sama dengan	<code>/=</code>	<code>a /= 2</code>	Memberikan nilai kepada variabel a dengan nilai variabel itu sendiri dibagi dengan nilai 2 atau (<code>a = a / 2</code>)
Sisa bagi sama dengan	<code>%=</code>	<code>a %= 2</code>	Memberikan nilai kepada variabel a dengan nilai variabel itu sendiri dibagi dengan nilai 2 atau (<code>a = a % 2</code>) tetapi hasil yang ditampilkan adalah sisa bagi
Pangkat sama dengan	<code>**=</code>	<code>a **= 3</code>	Memberikan nilai kepada variabel a dengan nilai variabel itu sendiri dipangkatkan dengan nilai 3 atau (<code>a = a ** 3</code>)

Pembagian bulat sama dengan	//=	a //= 3	Memberikan nilai kepada variabel a dengan nilai variabel itu sendiri dibagi dengan nilai 3 atau (a = a // 3) tetapi hasil yang ditampilkan adalah nilai pembagian yang dibulatkan
-----------------------------	-----	---------	---

d) Operator Logika (*Logical Operators*).

Operator Logika digunakan untuk membandingkan dua *Operand* atau dua nilai yang bertipe *Boolean* dan akan menghasilkan nilai *Boolean* yaitu *True* atau *False*. Berikut beberapa jenis operatornya:

Table 4. jenis-jenis operator logika

Operator	Contoh	Keterangan
and	7 > 2 and 2 > 1 (False)	Bernilai <i>True</i> karena kedua <i>operand</i> bernilai <i>True</i> . 7 > 2 = <i>True</i> , 2 > 1 = <i>True</i> . Jika salah satu atau kedua <i>operand</i> tersebut bernilai <i>False</i> maka hasilnya <i>False</i>
or	7 < 2 or 2 > 1 (False)	Bernilai <i>True</i> karena salah satu <i>operand</i> bernilai <i>True</i> . 7 < 2 = <i>False</i> , 2 > 1 = <i>True</i> . Jika salah satu atau kedua <i>operand</i> tersebut bernilai <i>True</i> maka hasilnya <i>True</i> . Hasil akan bernilai <i>False</i> ketika kedua <i>operand</i> bernilai <i>False</i> .
not	a = not True (False)	Bernilai <i>False</i> karena nilai dari <i>operand</i> bernilai <i>True</i> . Jika nilai <i>operand</i> awalnya bernilai <i>False</i> maka hasilnya akan bernilai <i>True</i> , begitupun sebaliknya

e) Operator Keanggotaan (*Membership Operators*).

Operator keanggotaan adalah operator yang bisa digunakan untuk memeriksa apakah ada sebuah nilai yang menjadi bagian keanggotaan dari sebuah *sequence*. *Sequence* bisa terdiri dari sebuah *list*, *string* atau *tuple*. Hasil dari operasi menggunakan operator keanggotaan hanya ada 2 yaitu *True* dan *False*. Operator ini terdiri dari:

Table 5. jenis-jenis operator keanggotaan

Operator	Contoh	Keterangan
----------	--------	------------

in	list_data = [2,3,4,5] 5 in list_data (True)	Dilakukan pemeriksaan terhadap sebuah list apakah ada nilai 5 di dalam list dengan nama variabel list_data. Karena nilai 5 terdapat dalam list tersebut, hasil yang ditampilkan adalah True
not in	list_data = [2,4,5,8] 8 not in list_data (False)	Dilakukan pemeriksaan terhadap sebuah list apakah nilai 8 tidak ada di dalam list dengan nama variabel list_data. Karena nilai 8 ada di dalam list tersebut, hasil yang ditampilkan adalah False

f) Operator Identitas (*Identity Operators*).

Operator identitas adalah operator yang digunakan untuk memeriksa identitas dari nilai apakah memiliki identitas memori yang sama atau tidak. Hasil dari operasi menggunakan operator keanggotaan hanya ada 2 yaitu *True* dan *False*. Operator ini terdiri dari:

Table 6. jenis-jenis operator identitas

Operator	Contoh	Keterangan
is	a = 10 b = 910 a is b (False)	Dilakukan pemeriksaan apakah nilai <i>operand</i> di sebelah kiri memiliki identitas memori yang sama dengan nilai <i>operand</i> yang berada di sebelah kanan. Jika sama maka akan bernilai <i>True</i> , jika tidak akan bernilai <i>False</i> .
is not	a = 10 b = 10 a is not b (True)	Dilakukan pemeriksaan apakah nilai <i>operand</i> di sebelah kiri memiliki identitas memori yang berbeda dengan nilai <i>operand</i> yang berada di sebelah kanan. Jika sama maka akan bernilai <i>True</i> , jika tidak akan bernilai <i>False</i> .

2. Pembahasan Tugas Guided

2.1. Operator Aritmetika.

Pada gambar 1 terdapat kode program berupa operasi aritmetika dengan dua variabel yaitu variabel x dengan nilai 9 dan y dengan nilai 2. Setiap operasi aritmetika

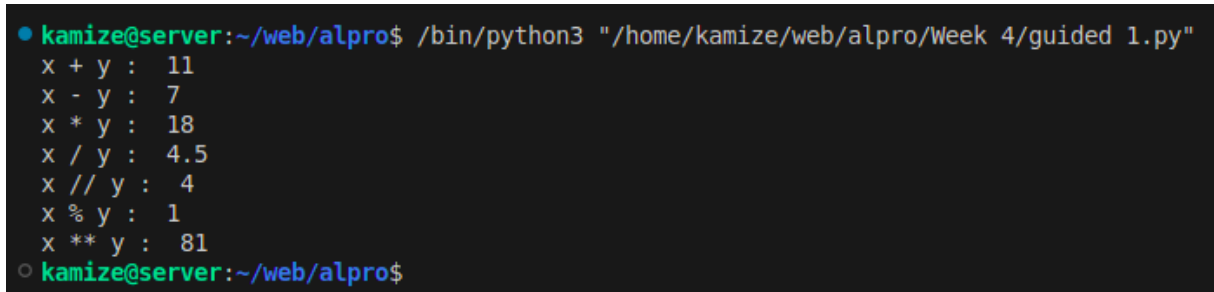
yang dilakukan dengan operand variabel x dan variabel y akan ditampilkan ke layar menggunakan fungsi print().

A screenshot of a code editor window titled 'alpro - guided 1.py'. The editor has a dark background with syntax-highlighted Python code. The code consists of 10 lines, each performing an arithmetic operation on variables x and y and printing the result with a descriptive comment in Indonesian. The operations are: addition, subtraction, multiplication, division (floating-point), integer division, modulus, and exponentiation.

```
1 x = 9
2 y = 2
3
4 print ("x + y : ", x + y) # penjumlahan
5 print ("x - y : ", x - y) # pengurangan
6 print ("x * y : ", x * y) # perkalian
7 print ("x / y : ", x / y) # pembagian (menghasilkan bilangan pecahan)
8 print ("x // y : ", x // y) # pembagian (menghasilkan bilangan bulat)
9 print ("x % y : ", x % y) # modulus (siswa pembagian)
10 print ("x ** y : ", x ** y) # pemangkatan
```

Gambar 1. Code operator aritmetika

Operasi penjumlahan $x + y$ menghasilkan nilai 11. Operasi pengurangan $x - y$ menghasilkan nilai 7. Operasi perkalian $x * y$ menghasilkan nilai 18. Operasi pembagian x / y menghasilkan nilai 4,5. Operasi pembagian bulat $x // y$ menghasilkan nilai 4. Operasi modulus $x \% y$ menghasilkan nilai 1. Operasi perpangkatan $x ** y$ menghasilkan nilai 81. Hasil Output sesuai dengan gambar 2.

A screenshot of a terminal window showing the execution of the Python script from Gambar 1. The prompt is 'kamize@server:~/web/alpro\$'. The output displays the results of the arithmetic operations: x + y : 11, x - y : 7, x * y : 18, x / y : 4.5, x // y : 4, x % y : 1, and x ** y : 81. The terminal ends with the prompt 'kamize@server:~/web/alpro\$' again.

```
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 4/guided 1.py"
x + y : 11
x - y : 7
x * y : 18
x / y : 4.5
x // y : 4
x % y : 1
x ** y : 81
○ kamize@server:~/web/alpro$
```

Gambar 2. Hasil output operator aritmetika

2.2. Operator Perbandingan (Comparison / Relational Operators).

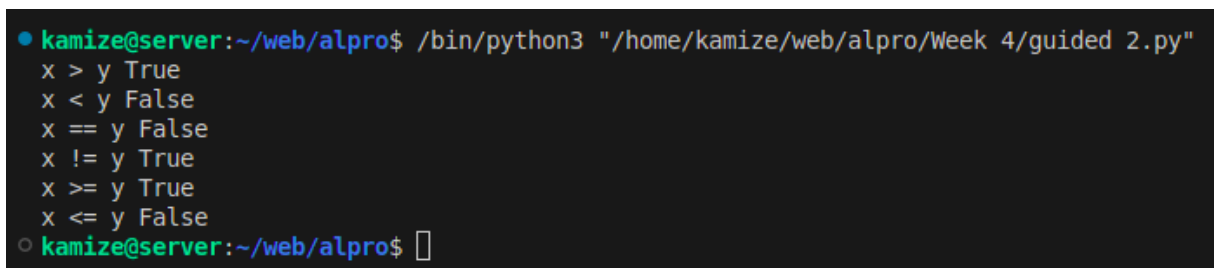
Pada gambar 3 terdapat kode program berupa operasi perbandingan dengan dua variabel yaitu variabel x dengan nilai 5 dan y dengan nilai 4. Setiap operasi perbandingan yang dilakukan dengan operand variabel x dan variabel y akan menghasilkan tipe data Boolean yang berupa *True* atau *False*.



```
1  # variabel
2  x = 5
3  y = 4
4
5  # operator perbandingan
6  print ("x > y", x > y) #lebih besar
7  print ("x < y", x < y) # lebih kecil
8  print ("x == y", x == y) # sama dengan
9  print ("x != y", x != y) # tidak sama dengan
10 print ("x >= y", x >= y) # Lebih besar sama dengan
11 print ("x <= y", x <= y) # lebih kecil sama dengan
```

Gambar 3. Code operator perbandingan

Operator lebih besar pada code $x > y$ menghasilkan nilai *True*, karena 5 lebih besar dari 4. Operator kurang pada code $x < y$ menghasilkan nilai *False*, karena 5 tidak lebih kecil dari 4. operator sama dengan pada code $x = y$ menghasilkan nilai *False*, karena 5 tidak sama dengan 4. operator tidak sama dengan pada kode $x != y$ menghasilkan nilai *True*, karena 5 tidak sama dengan 4. operator lebih besar sama dengan pada kode $x >= y$ menghasilkan nilai *True*, karena 5 lebih dari sama dengan 4. Operator lebih kuranga dari sama dengan pada kode $x <= y$ menghasilkan nilai *False*, karena 5 tidak kurang dari sama dengan 4. Pada terminal akan menghasilkan output sesuai gambar 4.



```
• kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 4/guided 2.py"
x > y True
x < y False
x == y False
x != y True
x >= y True
x <= y False
○ kamize@server:~/web/alpro$
```

Gambar 4. Hasil output operator perbandingan

2.3. Operator Penugasan (Assignment Operators).

Kode pada gambar 5 merupakan implementasi penggunaan operator penugasan (assignment operators) dalam Python. Operator ini digunakan untuk melakukan operasi aritmetika sekaligus dengan memperbarui nilai dari variabel dalam satu langkah.


```
alpro - guided 3.py
1 # menginputkan nilai a
2 a = 8 # nilai awal a
3
4 a+=2 # menambahkan dengan 2
5 print(a) # menampilkan hasil nilai a ditambah dengan 2, menghasilkan nilai 10
6 a-=3 # mengurangi dengan 3
7 print(a) # menampilkan hasil nilai a dikurangi dengan 3, menghasilkan nilai 7
8 a*=10 # mengalikan dengan 10
9 print(a) # menampilkan hasil nilai a dikali dengan 10, menghasilkan nilai 70
10 a/= 4 # membagi dengan 4
11 print(a) # menampilkan hasil nilai a dibagi dengan 4, menghasilkan nilai 17.5
12 a**= 2 # memangkatkan dengan 2
13 print(a) # menampilkan hasil nilai a dipangkatkan dengan 2, menghasilkan nilai 306.25
14
15 # berapa nilai a sekarang?
16 print("hasil nilai a adalah : ", a) # nilai akhir a
```

Gambar 5. Code operator penugasan

Variabel *a* yang bernilai 8 sebagai nilai awal untuk melakukan operasi aritmetika secara berurutan. Nilai ini akan dimodifikasi secara bertahap menggunakan operator penugasan. Operator `+=` menambahkan nilai 2 ke *a* dengan nilai *a* yang awalnya 8 di tambahkan dengan 2 menghasilkan nilai baru menjadi 10. Operator `-=` mengurangi nilai *a* yang baru itu 10 dengan 3 menghasilkan nilai baru menjadi 7. Operator `*=` mengkalikan nilai *a* yang baru yaitu 7 dengan 10 akan menghasilkan nilai baru menjadi 70. Operator `/=` membagi nilai *a* yang baru yaitu 70 dengan 4 akan menghasilkan nilai baru yaitu 17,5. Operator `**=` memangkatkan nilai *a* baru yaitu 17,5 dengan 2 akan menghasilkan nilai akhir menjadi 306,25.

```
kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 4/guided 3.py"
10
7
70
17.5
306.25
hasil nilai a adalah : 306.25
kamize@server:~/web/alpro$
```

Gambar 6. Hasil output operator penugasan

2.4. Operator Logika (Logical Operators).

Kode pada gambar 7 merupakan implementasi dari operator logika (*and*, *or*, dan *not*) dalam Bahasa pemrograman Python. Operator ini digunakan untuk membandingkan dua nilai boolean. Menghasilkan nilai *output* berupa *True* atau *False*



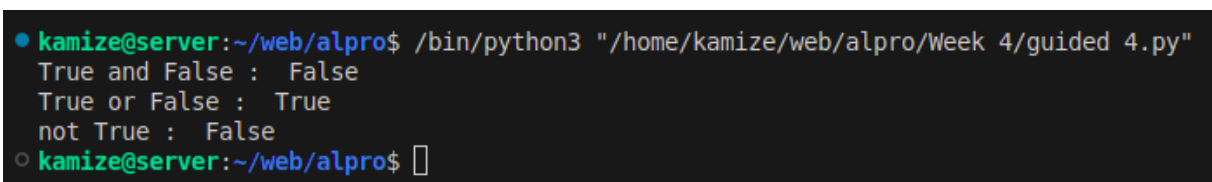
```

1 # operator logika
2
3 x = True
4 y = False
5
6 # logika and
7 z = x and y # true dan false hasilnya false karena kedua duanya harus true untuk menghasilkan nilai true
8 print ("True and False : ", z)
9
10 # logika or
11 z = x or y
12 print ("True or False : ", z)
13
14 # logika not
15 z = not x
16 print ("not True : ", z)

```

Gambar 7. Code operator logika

Variabel *x* yang bernilai *True* dan variabel *y* yang bernilai *False*. Operator *and* digunakan untuk memvalidasi apakah kedua nilai bernilai *True*. Karena *x* bernilai *True* dan *y* bernilai *False* maka *x and y* menghasilkan nilai *False* karena kedua nilai harus *True* untuk menghasilkan nilai *True*. Operator *or* digunakan untuk memvalidasi apakah ada salah satu yang nilai bernilai *True*. Karena *x* bernilai *True* dan *y* bernilai *False* maka *x or y* menghasilkan nilai *True* karena ada salah satu yang bernilai *True*. Operator *not* digunakan untuk menghasilkan kebalikan dari nilai *True* atau *False*. Karena *x* bernilai *True* maka *not x* menghasilkan nilai *False*.



```

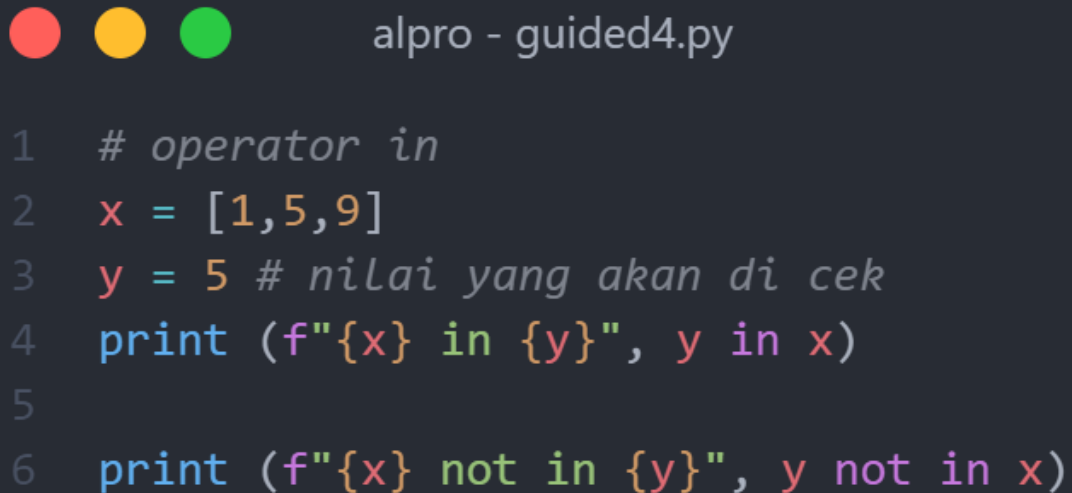
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 4/guided 4.py"
True and False : False
True or False : True
not True : False
○ kamize@server:~/web/alpro$ 

```

Gambar 8. Hasil output operator

2.5. Operator Keanggotaan (Membership Operators).

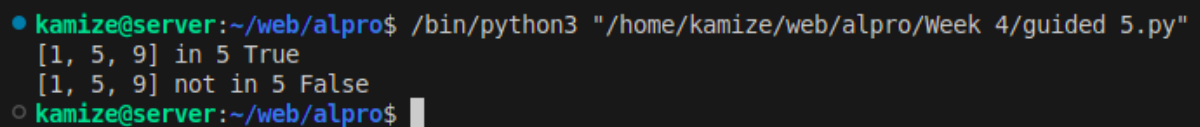
Kode pada gambar 9 menunjukkan implementasi dari penggunaan operator keanggotaan (*in* dan *not in*) dalam Bahasa pemrograman python. Operator ini digunakan memeriksa apakah ada sebuah nilai yang menjadi bagian keanggotaan dari sebuah *sequence*. *Sequence* bisa terdiri dari sebuah *list*, *string* atau *tuple*. Menghasilkan nilai *output* berupa *True* atau *False*.



```
1  # operator in
2  x = [1,5,9]
3  y = 5 # nilai yang akan di cek
4  print (f"{x} in {y}", y in x)
5
6  print (f"{x} not in {y}", y not in x)
```

Gambar 9. Code operator logika

Variabel x merupakan *list* yang beranggotakan [1,5,9] dan variabel y yang bernilai 5. Operator *in* digunakan untuk memeriksa apakah y termasuk kedalam anggota variabel x. Kode *y in x* akan menghasilkan nilai *True*, karena y termasuk anggota x. operator *not in* digunakan untuk memeriksa apakah y tidak termasuk kedalam anggota variabel x. Kode *y not in x* menghasilkan nilai *False*, karena y termasuk anggota x.



```
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 4/guided 5.py"
[1, 5, 9] in 5 True
[1, 5, 9] not in 5 False
○ kamize@server:~/web/alpro$
```

Gambar 10. Hasil output operator logika

2.6. Operator Identitas (Identity Operators).

Kode pada gambar 11 menunjukkan implementasi penggunaan operator identitas (*is* dan *is not*) dalam python. Operator ini digunakan untuk memeriksa identitas dari nilai apakah memiliki identitas memori yang sama atau tidak. Menghasilkan *output* berupa nilai *True* atau *False*.

```
alpro - guided5.py

1  # operator is
2  x = 4
3  y = x
4  z = y is x # menghasilkan nilai true karena y sama dengan x
5  print ("y adalah x", z)
6
7  # operator is not
8  x = 10 # nilai yang divalidasi
9  y = 4 # nilai untuk memvalidasi
10 # mengecek apakah y bukan x
11 z = y is not x
12 print ("y is not x", z) # menghasilkan nilai true karena y sama dengan x
```

Gambar 11. Code Identity Operators

Variabel x bernilai 4 dan variabel y yang diinisialisasi nilai x, sehingga y memiliki nilai 4. Operator is untuk memeriksa variabel x dan y memiliki identitas memori yang sama. Kode `x is y` menghasilkan nilai *True*, karena y memiliki identitas memori yang sama.

Variabel x bernilai 10 dan variabel y bernilai 4, nilai x digunakan untuk nilai yang divadidasi dan nilai y digunakan untuk nilai yang memvadidasi. Operator *is not* untuk memeriksa identitas memori yang tidak sama. Kode `y is not x` menghasilkan nilai *True*, karena y tidak memiliki identitas memori yang sama.

```
kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 4/guided 6.py"
y adalah x True
y is not x True
kamize@server:~/web/alpro$
```

Gambar 12. Hasil Output Identity Operators

3. Pembahasan Tugas Unguided

3.1. Jika nilai a adalah 9 dan b adalah 5, buatlah kodingan yang menjawab pertanyaan berikut:

- a) Apakah a sama dengan b?
- b) Apakah $a < b$?
- c) Apakah $a > b$?
- d) Apakah $a \leq b$?
- e) Apakah $a \geq b$?
- f) Apakah $a \neq b$?

```

1 # Nilai a dan b
2 a = 9
3 b = 5
4
5 # Pertanyaan
6 print("a) Apakah a = b? ", a == b) # menghasilkan nilai False karena 9 tidak sama dengan 5, bernilai True jika a sama dengan 5
7 print("b) Apakah a < b? ", a < b) # menghasilkan nilai False karena 9 lebih besar dari 5, bernilai True jika a lebih kecil dari 5
8 print("c) Apakah a > b? ", a > b) # menghasilkan nilai True karena 9 lebih besar dari 5, bernilai True jika a lebih besar dari 5
9 print("d) Apakah a <= b? ", a <= b) # menghasilkan nilai False karena 9 lebih besar dari 5, bernilai True jika a lebih kecil sama dengan 5
10 print("e) Apakah a >= b? ", a >= b) # menghasilkan nilai True karena 9 lebih besar dari 5, bernilai True jika a lebih besar sama dengan 5
11 print("f) Apakah a != b? ", a != b) # menghasilkan nilai True karena 9 tidak sama dengan 5, bernilai True jika a tidak sama dengan 5

```

Gambar 13. Code unguided 1

Operator sama dengan pada code $a = b$ menghasilkan nilai *False*, karena 9 tidak sama dengan 5. Operator kurang dari pada code $a < b$ menghasilkan nilai *False*, karena 9 tidak kurang dari 5. Operator lebih dari pada code $a > b$ menghasilkan nilai *True*, karena 9 lebih dari 5. operator kurang dari sama dengan pada kode $a <= b$ menghasilkan nilai *False*, karena 9 lebih dari sama dengan 5. operator lebih dari sama dengan pada kode $a >= b$ menghasilkan nilai *True*, karena 9 lebih dari sama dengan 5. Operator tidak sama dengan pada kode $a != b$ menghasilkan nilai *True*, karena 9 tidak sama dengan 5. Pada terminal akan menghasilkan *output* sesuai gambar 14.

```

kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 4/unguided 1.py"
a) Apakah a = b? False
b) Apakah a < b? False
c) Apakah a > b? True
d) Apakah a <= b? False
e) Apakah a >= b? True
f) Apakah a != b? True
kamize@server:~/web/alpro$

```

Gambar 14. Hasil output unguided 1

3.2. Jika nilai a adalah 5 dan b adalah 6, buat kodingan untuk menjawab pertanyaan berikut:

- $a + b = ?$
- $b - a = ?$
- $a * b = ?$
- $a / b = ?$

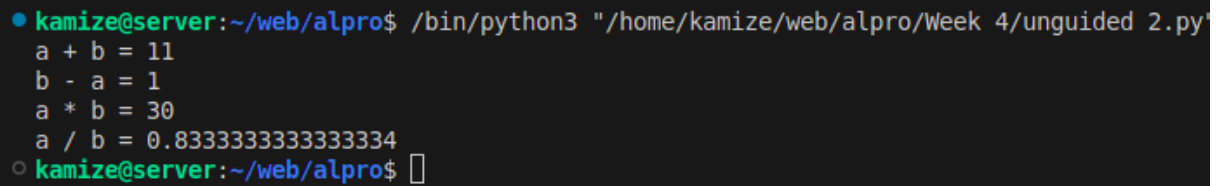
```

1 # Nilai a dan b
2 a = 5
3 b = 6
4
5 # Operasi matematika
6 print("a + b =", a + b) # hasil penjumlahan a dan b adalah 11
7 print("b - a =", b - a) # hasil pengurangan b dan a adalah 1
8 print("a * b =", a * b) # hasil perkalian a dan b adalah 30
9 print("a / b =", a / b) # hasil pembagian a dan b adalah 0.8333333333333334

```

Gambar 15. Code unguided 2

Variabel *a* bernilai 5 dan variabel *b* bernilai 6. Operasi penjumlahan $a + b$ menghasilkan nilai 11. Operasi pengurangan $b - a$ menghasilkan nilai 1. Operasi perkalian $x * y$ menghasilkan nilai 30. Operasi pembagian x / y menghasilkan nilai 0,833...4. Hasil *Output* sesuai dengan gambar 16.



```
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/Week 4/unguided 2.py"
a + b = 11
b - a = 1
a * b = 30
a / b = 0.8333333333333334
○ kamize@server:~/web/alpro$
```

Gambar 16. Hasil output unguided 2

4. Ringkasan

Laporan praktikum ini membahas dasar pemrograman python tentang berbagai jenis operator yang digunakan pada operasi aritmatika, perbandingan, penugasan, logika, keanggotaan, dan identitas. Operator ini berperan penting dalam membuat alur program secara efisien dan terstruktur.

Yang pertama terdapat yang menghasilkan nilai real. Seperti pada operator aritmatika (+, -, *, /, %, **, dan //) yang digunakan untuk melakukan operasi matematika. Misalnya, kode $3 + 2$ menghasilkan nilai 5. Operator penugasan (=, +=, -=, *=, /=, %=, //=, **=) digunakan untuk memberikan nilai ke variabel dan melakukan operasi matematika secara langsung dalam proses penugasan. Misalnya, $a += 5$ setara dengan $a = a + 5$, yang memungkinkan penulisan kode lebih ringkas.

Yang kedua terdapat operator yang menghasilkan nilai berupa *True* atau *False*. Seperti operator perbandingan (==, !=, >, <, >=, <=) memungkinkan program membandingkan dua nilai. Dalam operator logika (*and* dan *or not*) untuk membandingkan dua nilai *Boolean*. Operator keanggotaan (*in* dan *not in*) memeriksa keberadaan sebuah nilai dalam *sequence*. Sedangkan Operator identitas (*is* dan *is not*) digunakan untuk mengecek apakah dua variabel terletak pada memori yang sama.

Laporan ini juga berisi tugas praktik (*guided*) yang mencakup implementasi berbagai jenis operator dalam Python. Salah satu contoh latihan adalah penggunaan operator aritmatika dalam operasi perhitungan, operator perbandingan untuk mengevaluasi ekspresi logis, serta penggunaan operator logika. Dalam bagian tugas mandiri (*unguided*), diberikan studi kasus seperti pembuatan kalkulator sederhana dengan operator aritmatika, pengujian kondisi dengan operator perbandingan.

Dari laporan ini, dapat disimpulkan bahwa pemahaman tentang konsep dasar operator dalam Python sangat penting dalam pemrograman. Dengan menguasai operator ini,

programmer dapat menuliskan kode yang lebih efisien, mudah dipahami, dan terstruktur dengan baik, sehingga dapat mendukung pengembangan kode ke yang lebih kompleks.