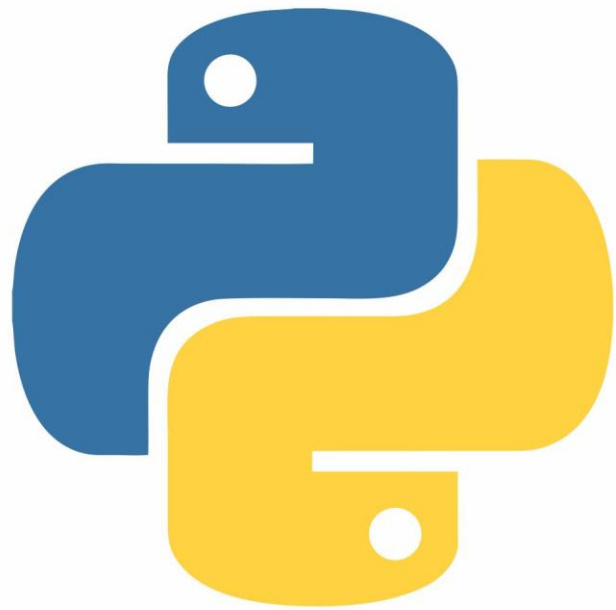


PYTHON

Pemrograman V



PYTHON

Sintaks Dasar

1.1 Identifier

Identifier adalah identitas atau nama yang telah diberikan kepada function, variabel, obyek, class, namespace dan lain-lain.

- Identifier bisa terdiri dari kombinasi huruf kecil (a-z) atau huruf besar (A-Z), angka (0-9), dan underscore (_).
- Identifier tidak bisa dimulai menggunakan angka, misalnya 1variabel.
- Kata kunci tidak bisa digunakan sebagai identifier.

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

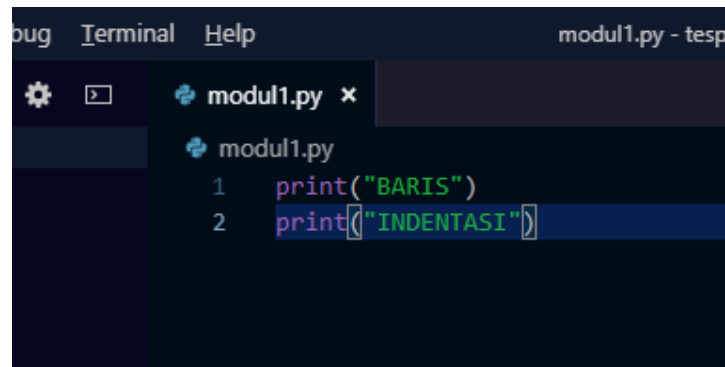
Gambar 1 Kata Kunci

- Tidak bisa menggunakan karakter spesial seperti @, \$, and %.
- Python bersifat case sensitive, variable dan Variabel adalah 2 identifier berbeda.

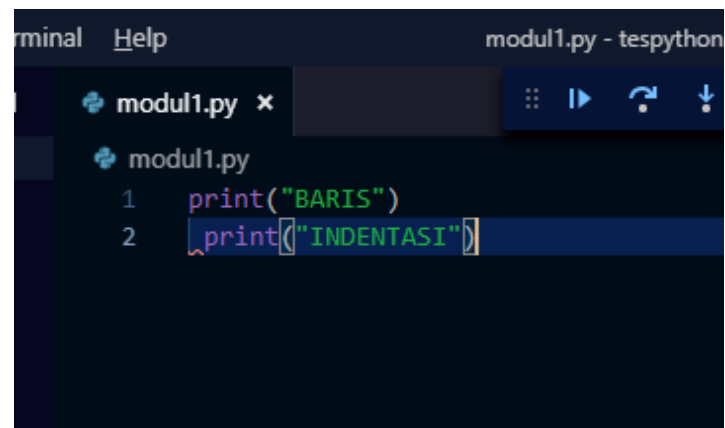
1.2 Baris dan Indentasi

Python tidak menggunakan tanda { } untuk menandai blok/grup kode. Blok kode di python menggunakan tanda indentasi (spasi). Jumlah spasi untuk setiap baris yang ada dalam satu blok kode harus sama.

PYTHON



Gambar 2 Contoh Benar



Gambar 3 Contoh Salah

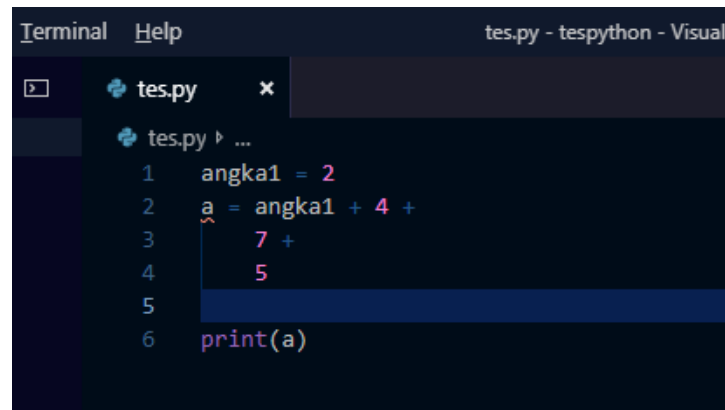
1.3 Pernyataan Multibaris

Di Python, akhir dari sebuah statement adalah karakter baris baru (newline). Kita dapat membuat sebuah statement terdiri dari beberapa baris dengan menggunakan tanda backslash (\).



Gambar 4 Contoh Benar

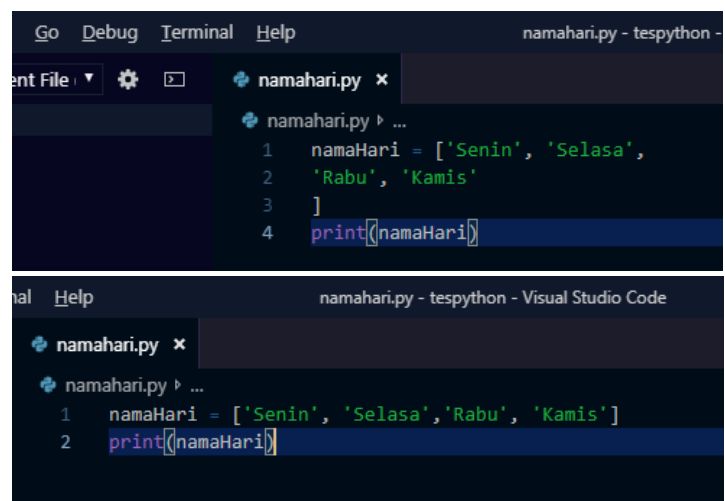
PYTHON



```
tes.py
tes.py ▶ ...
1  angka1 = 2
2  a = angka1 + 4 +
3    7 +
4    5
5
6  print(a)
```

Gambar 5 Contoh Salah

Statement yang ada di dalam tanda kurung [], { }, dan () tidak memerlukan tanda \. Contoh :



```
namahari.py
namahari.py ▶ ...
1  namaHari = ['Senin', 'Selasa',
2             'Rabu', 'Kamis'
3             ]
4  print(namaHari)

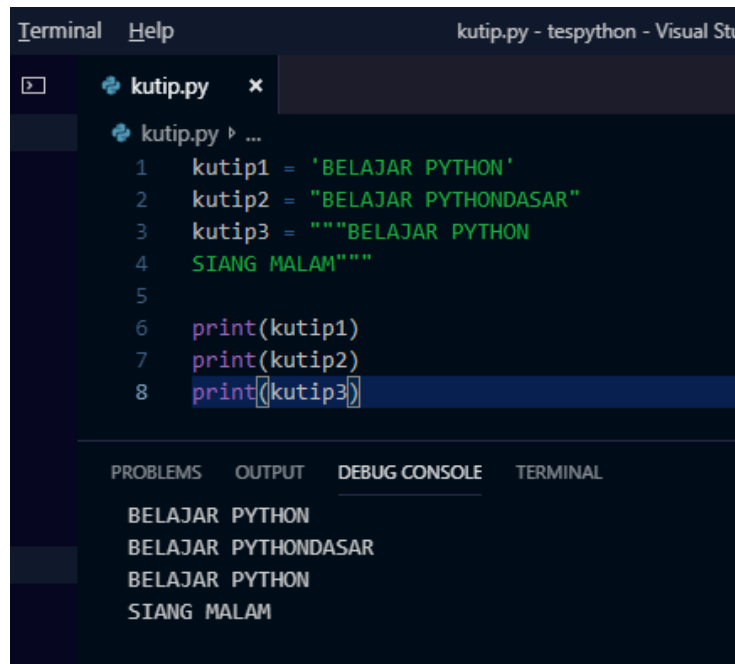
namahari.py
namahari.py ▶ ...
1  namaHari = ['Senin', 'Selasa', 'Rabu', 'Kamis']
2  print(namaHari)
```

Gambar 6 Contoh Statement Dalam Tanda Kurung

1.4 Tanda Kutip

Python menggunakan tanda kutip tunggal ('), ganda ("), maupun triple (""" atau """) untuk menandai string, sepanjang stringnya diawali oleh tanda kutip yang sama di awal dan akhir string. Tanda kutip tiga digunakan untuk string multibaris.

PYTHON



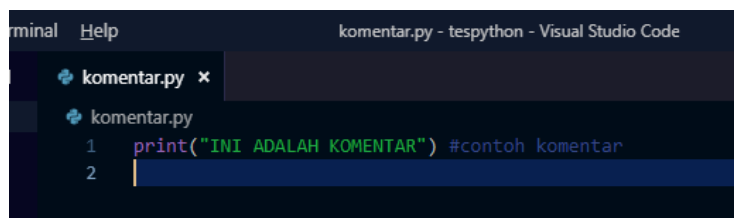
```
Terminal Help kutip.py - tespython - Visual Studio Code
kutip.py x
kutip.py > ...
1 kutip1 = 'BELAJAR PYTHON'
2 kutip2 = "BELAJAR PYTHONDASAR"
3 kutip3 = """BELAJAR PYTHON
4 SIANG MALAM"""
5
6 print(kutip1)
7 print(kutip2)
8 print(kutip3)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
BELAJAR PYTHON
BELAJAR PYTHONDASAR
BELAJAR PYTHON
SIANG MALAM
```

Gambar 7 Penggunaan Tanda Kutip

1.5 Komentar

Tanda pagar (#) digunakan untuk menandai komentar di python. Komentar tidak akan diproses oleh interpreter Python. Komentar hanya berguna untuk programmer untuk memudahkan memahami maksud dari kode.

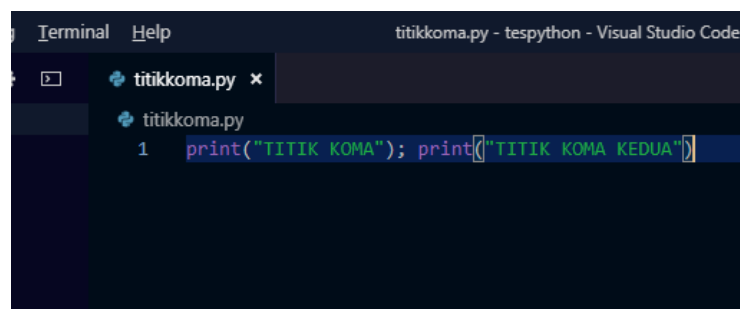


```
Terminal Help komentar.py - tespython - Visual Studio Code
komentar.py x
komentar.py
1 print("INI ADALAH KOMENTAR") #contoh komentar
2
```

Gambar 8 Contoh Komentar

1.6 Dua Pernyataan Dalam Satu Baris

Titik koma dapat digunakan ketika terdapat 2 pernyataan dalam 1 baris kode.



```
Terminal Help titikcoma.py - tespython - Visual Studio Code
titikcoma.py x
titikcoma.py
1 print("TITIK KOMA"); print("TITIK KOMA KEDUA")
```

Gambar 9 Dua Pernyataan Dalam Satu Baris

PYTHON

Tipe Data

Beberapa tipe data built-in atau bawaan yang cukup lengkap dan tidak sulit untuk digunakan. Beberapa tipe data yang wajib Anda ketahui di Python3 antara lain:

- **int**, tipe data yang dapat Anda isi dengan bilangan bulat.
- **float**, tipe data yang dapat Anda isi dengan bilangan koma.
- **string**, tipe data yang dapat Anda isi dengan sebuah karakter atau kalimat.
- **complex**, tipe data bilangan kompleks atau bilangan imajiner, seperti 5j, 54j, 1j.
- **long**, tipe data yang dapat Anda isi dengan bilangan yang sangat besar. Bisa lebih dari 1 milyar.
- **boolean**, tipe data yang dapat Anda isi untuk mendukung operasi bool. Nilainya hanya True dan False.
- **List**, adalah tipe data yang berisi item yang berurut. List bisa berisi anggota dengan tipe yang sama maupun berbeda. Untuk mendeklarasikan list, digunakan tanda kurung [] dan masing-masing anggotanya dipisahkan oleh tanda koma.
- **Tuple**, adalah jenis data lain yang mirip dengan list. Perbedaannya dengan list adalah anggotanya tidak bisa diubah (immutable). Tuple dideklarasikan dengan menggunakan tanda kurung (). dan anggotanya dipisahkan oleh tanda koma.
- **Set**, adalah salah satu tipe data di Python yang tidak berurut (unordered). Set memiliki anggota yang unik (tidak ada duplikasi). Jadi misalnya kalau kita meletakkan dua anggota yang sama di dalam set, maka otomatis set akan menghilangkan yang salah satunya. Set dibuat dengan meletakkan anggota – anggotanya di dalam tanda kurung kurawal { }, dipisahkan menggunakan tanda koma.
- **Dictionary**, adalah tipe data yang tiap anggotanya terdiri dari pasangan kunci-nilai (key-value). Dictionary dideklarasikan dengan menggunakan tanda kurung kurawal { }, dimana anggotanya memiliki bentuk **kunci:nilai** atau **key:value** dan tiap anggota dipisah tanda koma. Kunci dan nilainya bisa memiliki tipe sembarang.

Kita bisa menggunakan fungsi **type()** untuk mengetahui tipe data suatu objek di python. Contoh :

```
x = 5
print(x, "tipenya adalah ", type(x))
x = 2.0
print(x, "tipenya adalah ", type(x))
```

PYTHON

Variabel, Operator, dan Ekspresi

3.1 Variabel

Variabel adalah lokasi di memori yang digunakan untuk menyimpan nilai. Pada saat kita membuat sebuah variabel, kita 'memesan' tempat di dalam memori. Tempat tersebut bisa diisi dengan data atau objek, baik itu bilangan bulat (integer), pecahan (float), karakter (string), dan lain - lain.

3.1.1 Memberi Nilai Variabel

Di python, variabel tidak perlu dideklarasikan secara eksplisit. Deklarasi atau pembuatan variabel terjadi secara otomatis pada saat kita memberi (menugaskan) suatu nilai ke variabel. Tanda sama dengan (=) digunakan untuk memberikan nilai ke variabel.

3.1.2 Multi penugasan

Kita bisa memberi nilai ke beberapa variabel secara bersamaan seperti berikut.

```
x = y = z = 3
```

Kita juga bisa menugaskan masing - masing variabel ke objek yang berbeda dalam sebaris perintah. Sebagai contoh :

```
a, b, c = 1, 3.4, "Umar"
```

3.2 Operator & Ekspresi

Hampir semua statemen (baris logis) yang Anda tulis akan mengandung ekspresi. Contoh sederhana dari ekspresi adalah $2+3$. Sebuah ekspresi dapat diturunkan menjadi operator dan operand. Operator adalah simbol tertentu yang digunakan untuk melakukan operasi aritmatika maupun logika. Nilai yang padanya dilakukan operasi disebut operand. Misalnya adalah $2 + 3$. Di sini tanda + adalah operator penjumlahan. 2 dan 3 adalah operand.

Python memiliki sejumlah operator, yaitu :

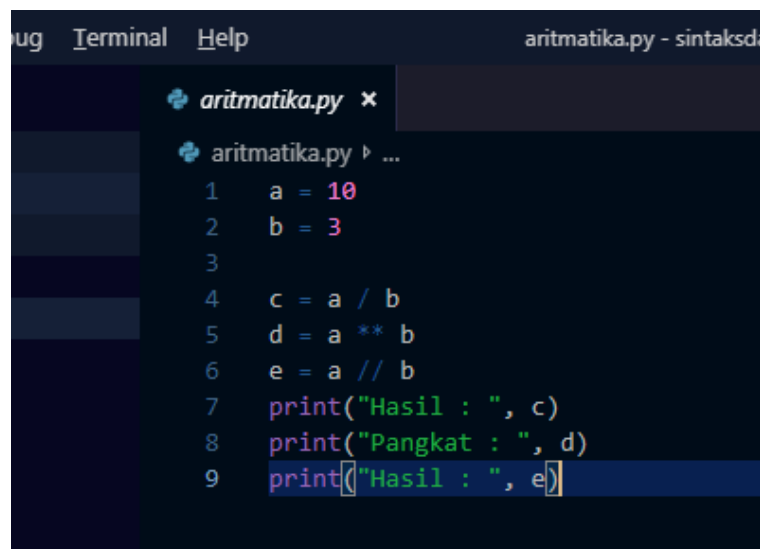
3.2.1 Operator Aritmatika

Operator aritmatika adalah operator yang digunakan untuk melakukan operasi matematika, seperti penjumlahan, pengurangan, perkalian, pembagian, dan sebagainya. Tabel berikut menunjukkan jenis operator aritmatika.

PYTHON

Operator	Nama dan Fungsi	Contoh
+	Penjumlahan, menjumlahkan 2 buah operand	$x + y$
-	Pengurangan, mengurangi 2 buah operand	$x - y$
*	Perkalian, mengalikan 2 buah operand	$x * y$
/	Pembagian, membagi 2 buah operand	x / y
**	Pemangkatan, memangkatkan bilangan	$x ** y$
//	Pembagian bulat, menghasilkan hasil bagi tanpa koma	$x // y$
%	Modulus, menghasilkan sisa pembagian 2 bilangan	$x \% y$

Gambar 10 Operator Aritmatika



```
aritmatika.py x
aritmatika.py > ...
1  a = 10
2  b = 3
3
4  c = a / b
5  d = a ** b
6  e = a // b
7  print("Hasil : ", c)
8  print("Pangkat : ", d)
9  print("Hasil : ", e)
```

Gambar 11 Contoh Kode Operator Aritmatika

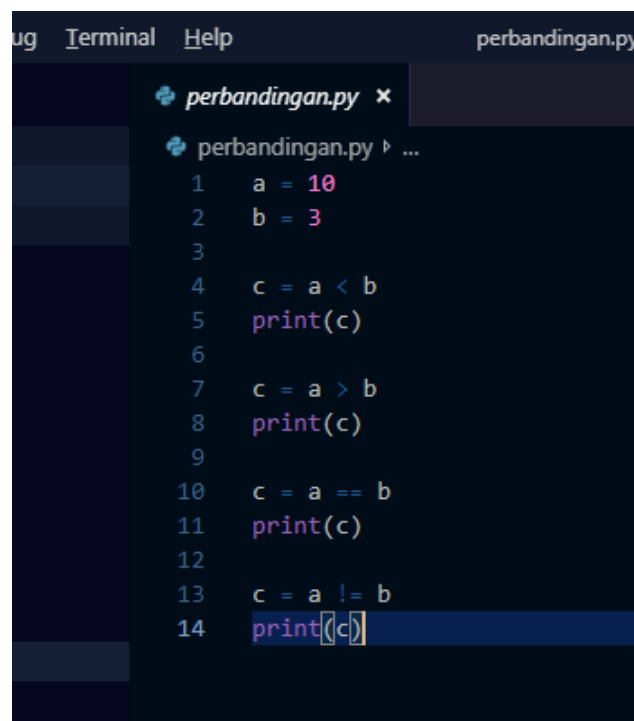
3.2.2 Operator Perbandingan

Operator perbandingan adalah operator yang digunakan untuk membandingkan 2 buah nilai. Hasil perbandingannya adalah True atau False tergantung kondisi.

PYTHON

Operator	Nama dan Fungsi	Contoh
>	Lebih besar dari – Hasilnya True jika nilai sebelah kiri lebih besar dari nilai sebelah kanan	x > y
<	Lebih kecil dari – Hasilnya True jika nilai sebelah kiri lebih kecil dari nilai sebelah kanan	x < y
==	Sama dengan – Hasilnya True jika nilai sebelah kiri sama dengan nilai sebelah kanan	x == y
!=	Tidak sama dengan – Hasilnya True jika nilai sebelah kiri tidak sama dengan nilai sebelah kanan	x != y
>=	Lebih besar atau sama dengan – Hasilnya True jika nilai sebelah kiri lebih besar atau sama dengan nilai sebelah kanan	x >= y
<=	Lebih kecil atau sama dengan – Hasilnya True jika nilai sebelah kiri lebih kecil atau sama dengan nilai sebelah kanan	x <= y

Gambar 12 Operator Perbandingan



```
perbandingan.py
perbandingan.py x
perbandingan.py ▶ ...
1  a = 10
2  b = 3
3
4  c = a < b
5  print(c)
6
7  c = a > b
8  print(c)
9
10 c = a == b
11 print(c)
12
13 c = a != b
14 print(c)
```

Gambar 13 Contoh Kode Operator Perbandingan

3.2.3 Operator Penugasan

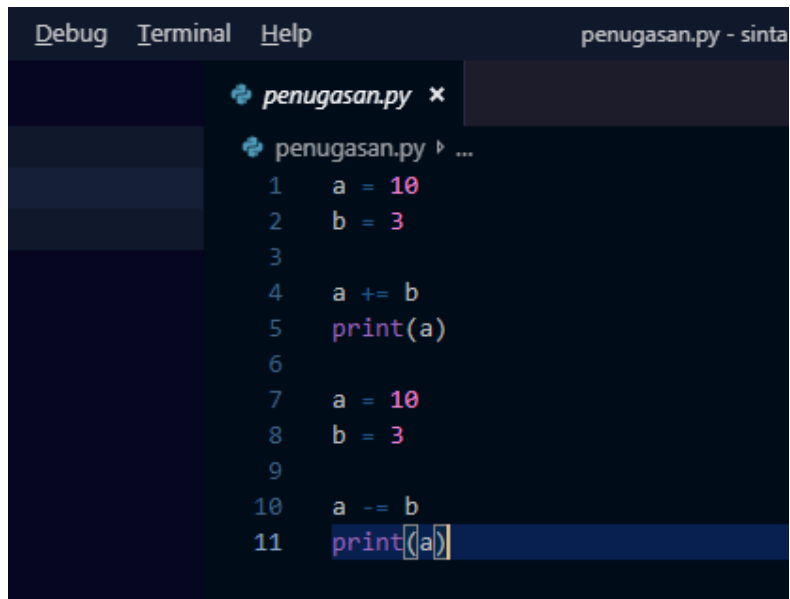
Operator penugasan adalah operator yang digunakan untuk memberi nilai ke variabel.

PYTHON

Operator	Penjelasan	Contoh
=	Menugaskan nilai yang ada di kanan ke operand yang ada di sebelah kiri	$c = a + b$ menugaskan $a + b$ ke c
+=	Menambahkan operand yang di kanan dengan operand yang ada di kiri dan hasilnya di tugaskan ke operand yang di kiri	$c += a$ sama dengan $c = c + a$
-=	Mengurangi operand yang di kanan dengan operand yang ada di kiri dan hasilnya di tugaskan ke operand yang di kiri	$c -= a$ sama dengan $c = c - a$
*=	Mengalikan operand yang di kanan dengan operand yang ada di kiri dan hasilnya di tugaskan ke operand yang di kiri	$c *= a$ sama dengan $c = c * a$
/=	Membagi operand yang di kanan dengan operand yang ada di kiri dan hasilnya di tugaskan ke operand yang di kiri	$c /= a$ sama dengan $c = c / a$
**=	Memangkatkan operand yang di kanan dengan operand yang ada di kiri dan hasilnya ditugaskan ke operand yang di kiri	$c **= a$ sama dengan $c = c ** a$
//=	Melakukan pembagian bulat operand di kanan terhadap operand di kiri dan hasilnya disimpan di operand yang di kiri	$c //= a$ sama dengan $c = c // a$
%=	Melakukan operasi sisa bagi operand di kanan dengan operand di kiri dan hasilnya di simpan di operand yang di kiri	$c \% = a$ sama dengan $c = c \% a$

Gambar 14 Operator Penugasan

PYTHON

A screenshot of a Python IDE window titled 'penugasan.py - sinta'. The window has tabs for 'Debug', 'Terminal', and 'Help'. The main editor area shows a file named 'penugasan.py' with the following code:

```
1 a = 10
2 b = 3
3
4 a += b
5 print(a)
6
7 a = 10
8 b = 3
9
10 a -= b
11 print(a)
```

Gambar 15 Contoh Kode Operator Penugasan

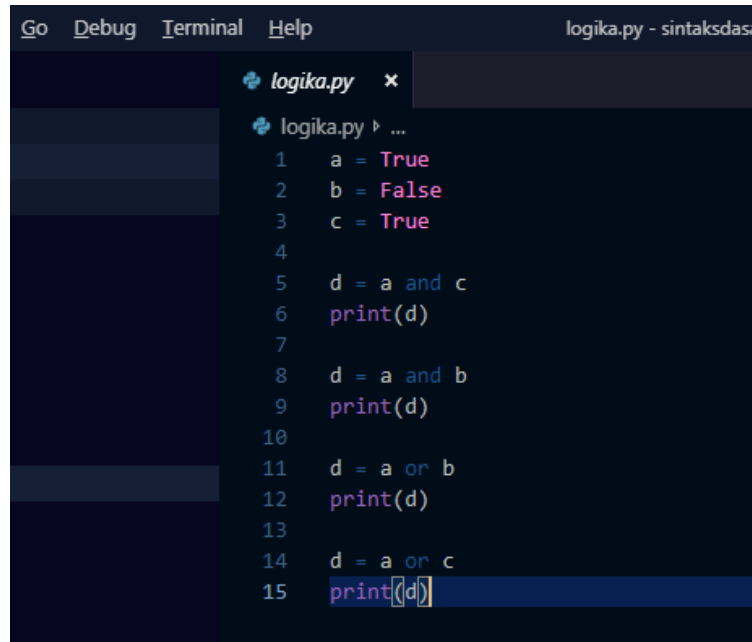
3.2.4 Operator Logika

Operator logika adalah operator yang digunakan untuk melakukan operasi logika.

Operator	Penjelasan	Contoh
and	Hasilnya adalah True jika kedua operandnya bernilai benar	x and y
or	Hasilnya adalah True jika salah satu atau kedua operandnya bernilai benar	x or y
not	Hasilnya adalah True jika operandnya bernilai salah (kebalikan nilai)	not x

Gambar 16 Operator Logika

PYTHON

A screenshot of a Python IDE window titled 'logika.py - sintaksdasar'. The code in the editor is as follows:

```
1 a = True
2 b = False
3 c = True
4
5 d = a and c
6 print(d)
7
8 d = a and b
9 print(d)
10
11 d = a or b
12 print(d)
13
14 d = a or c
15 print(d)
```

Gambar 17 Contoh Kode Operator Logika

3.2.5 Operator Bitwise

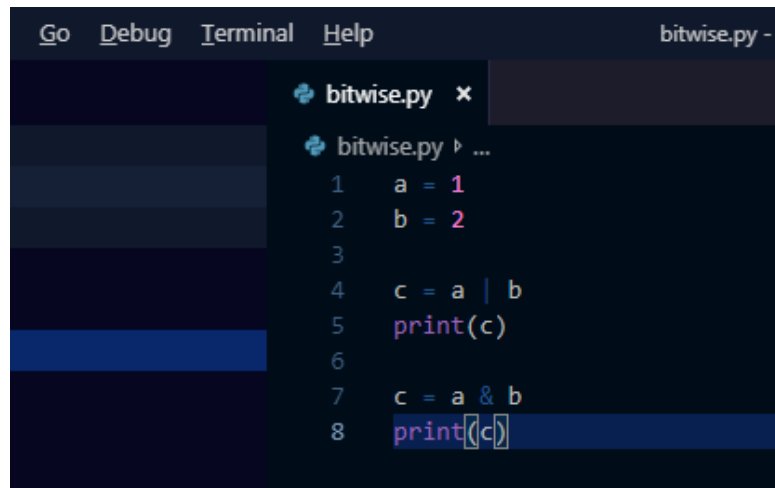
Operator bitwise adalah operator yang melakukan operasi bit terhadap operand. Operator ini beroperasi bit per bit sesuai dengan namanya. Sebagai misal, angka 2 dalam bit ditulis 10 dalam notasi biner dan angka 7 ditulis 111.

Pada tabel di bawah ini, misalkan $x = 10$ (0000 1010) dalam biner dan $y = 4$ (0000 0100) dalam biner.

Operator	Nama	Contoh
&	Bitwise AND	$x \& y = 0$ (0000 0000)
	Bitwise OR	$x y = 14$ (0000 1110)
~	Bitwise NOT	$\sim x = -11$ (1111 0101)
^	Bitwise XOR	$x \wedge y = 14$ (0000 1110)
>>	Bitwise right shift	$x >> 2 = 2$ (0000 0010)
<<	Bitwise left shift	$x << 2 = 40$ (0010 1000)

Gambar 18 Operator Bitwise

PYTHON



```
Go Debug Terminal Help bitwise.py - s
bitwise.py x
bitwise.py ▶ ...
1 a = 1
2 b = 2
3
4 c = a | b
5 print(c)
6
7 c = a & b
8 print(c)
```


Gambar 19 Contoh Kode Operator Bitwise

3.2.6 Operator Identitas

Operator identitas adalah operator yang memeriksa apakah dua buah nilai (atau variabel) berada pada lokasi memori yang sama.

Operator	Penjelasan	Contoh
is	True jika kedua operand identik (menunjuk ke objek yang sama)	x is True
is not	True jika kedua operand tidak identik (tidak merujuk ke objek yang sama)	x is not True

Gambar 20 Operator Identitas



```
ug Terminal Help identitas.py - sintaksdasar
bitwise.py identitas.py x keanggotas
identitas.py ▶ ...
1 a = 1
2 b = 2
3
4 print(1 is a)
5 print(2 is b)
6 print(3 is a)
7 print(1 is not a)
8 print(2 is not b)
9
10 print(" ")
11
12 print(type(a) is int)
13 print(type(b) is float)
```

Gambar 21 Contoh Kode Operator Identitas

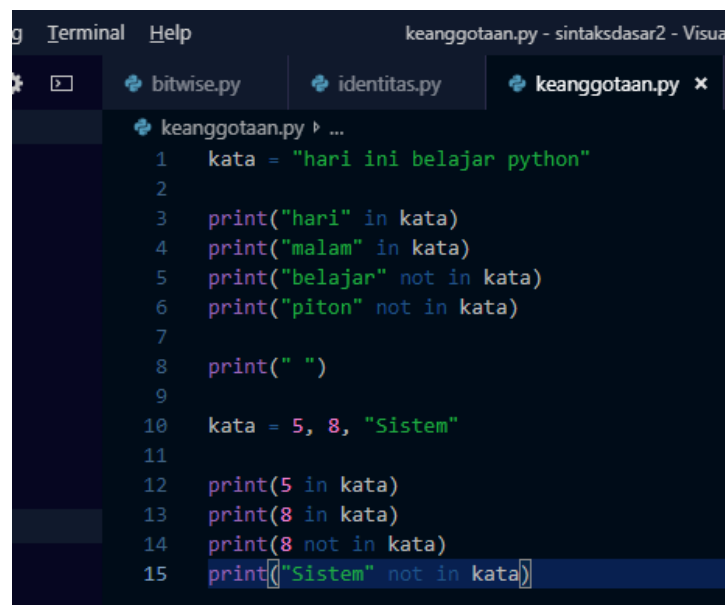
PYTHON

3.2.7 Operator Keanggotaan

Operator keanggotaan adalah operator yang digunakan untuk memeriksa apakah suatu nilai atau variabel merupakan anggota atau ditemukan di dalam suatu data (string, list, tuple, set, dan dictionary).

Operator	Penjelasan	Contoh
in	True jika nilai/variabel ditemukan di dalam data	5 in x
not in	True jika nilai/variabel tidak ada di dalam data	5 not in x

Gambar 22 Operator Keanggotaan



```
g Terminal Help keanggotaan.py - sintaksdasar2 - Visual Studio Code
# bitwise.py identitas.py keanggotaan.py x
keanggotaan.py ▶ ...
1 kata = "hari ini belajar python"
2
3 print("hari" in kata)
4 print("malam" in kata)
5 print("belajar" not in kata)
6 print("piton" not in kata)
7
8 print(" ")
9
10 kata = 5, 8, "Sistem"
11
12 print(5 in kata)
13 print(8 in kata)
14 print(8 not in kata)
15 print("Sistem" not in kata)
```

Gambar 23 Contoh Kode Operator Keanggotaan