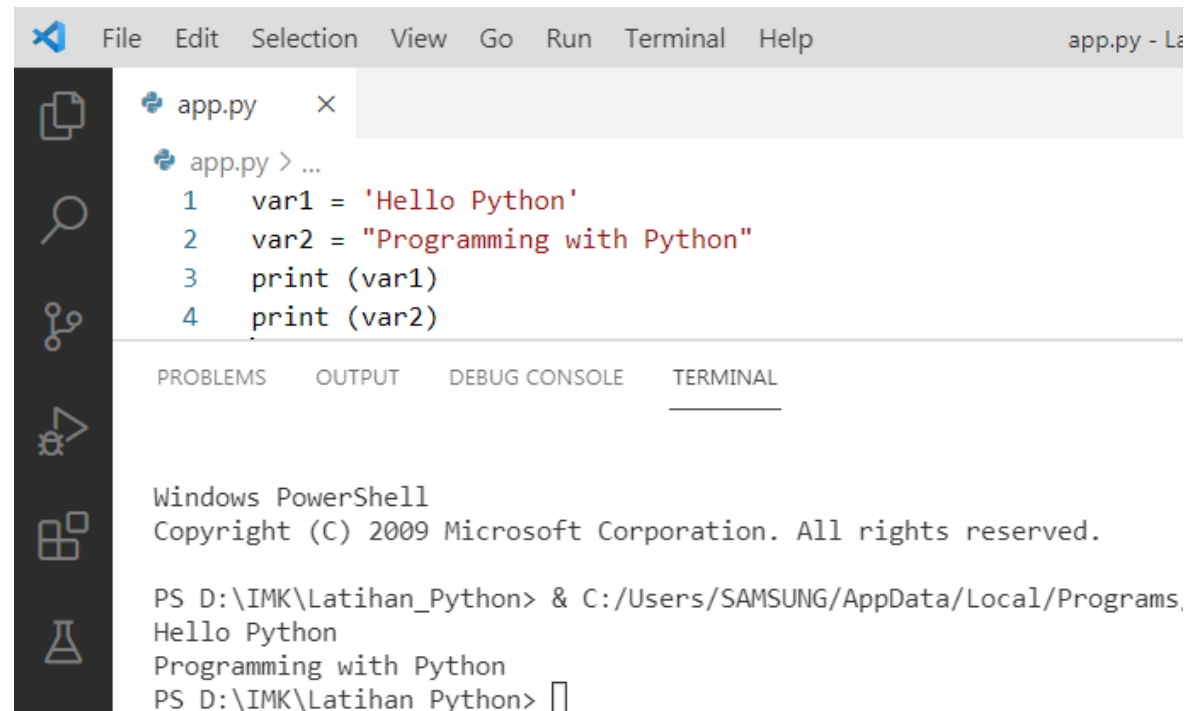


Pertemuan 3

String, Bilangan & Operator

String

String adalah tipe data yang paling sering digunakan di Python. Kita bisa membuat string dengan meletakkan karakter di dalam tanda kutip. Tanda kutipnya bisa kutip tunggal maupun kutip ganda. Contohnya adalah sebagai berikut:

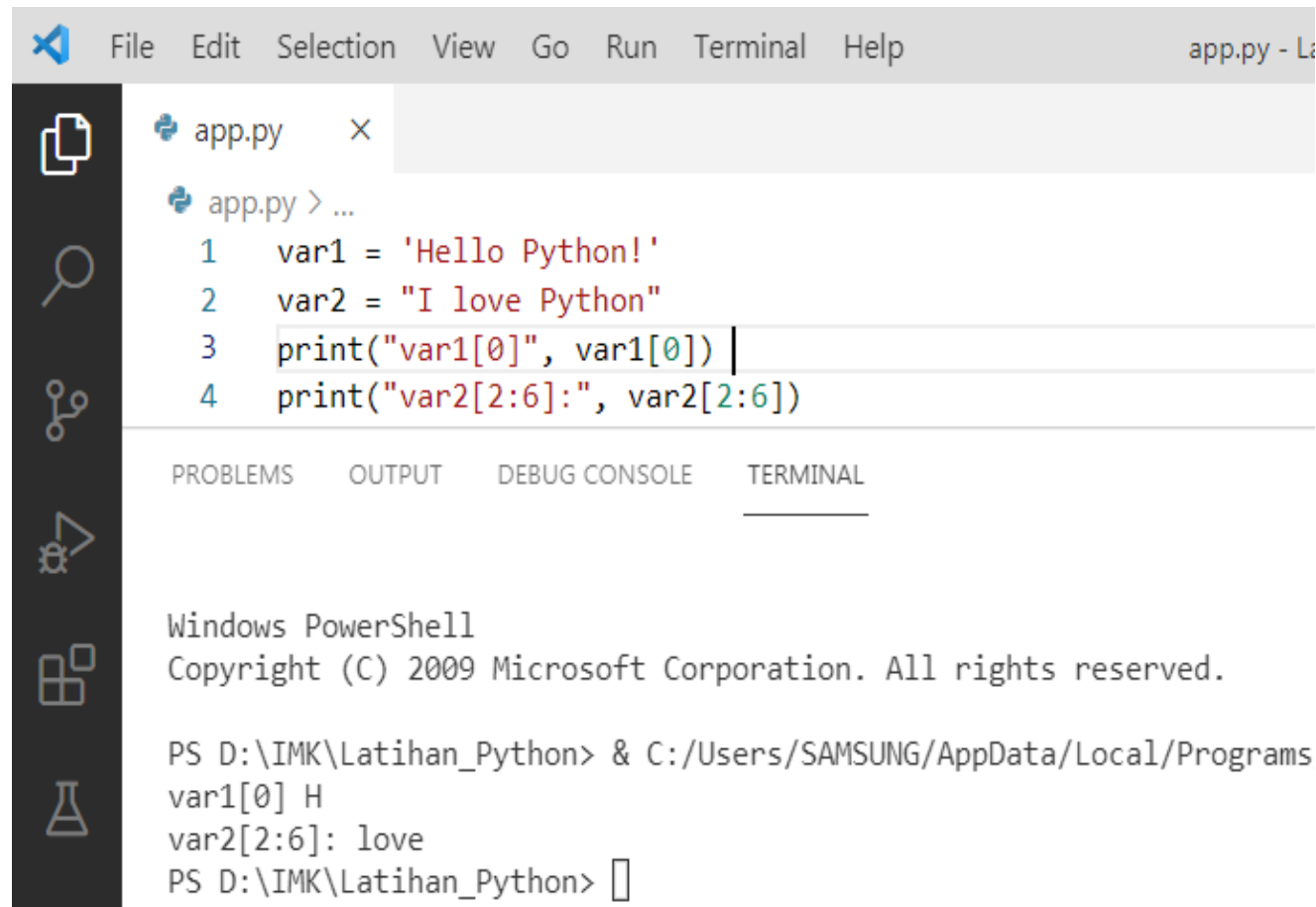


```
File Edit Selection View Go Run Terminal Help app.py - La
app.py x
app.py > ...
1 var1 = 'Hello Python'
2 var2 = "Programming with Python"
3 print (var1)
4 print (var2)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs,
Hello Python
Programming with Python
PS D:\IMK\Latihan Python> █
```

Mengakses Nilai String

Untuk mengakses nilai atau substring dari string, kita menggunakan indeks dalam tanda [].



```
File Edit Selection View Go Run Terminal Help app.py - La
app.py x
app.py > ...
1 var1 = 'Hello Python!'
2 var2 = "I love Python"
3 print("var1[0]", var1[0])
4 print("var2[2:6]:", var2[2:6])

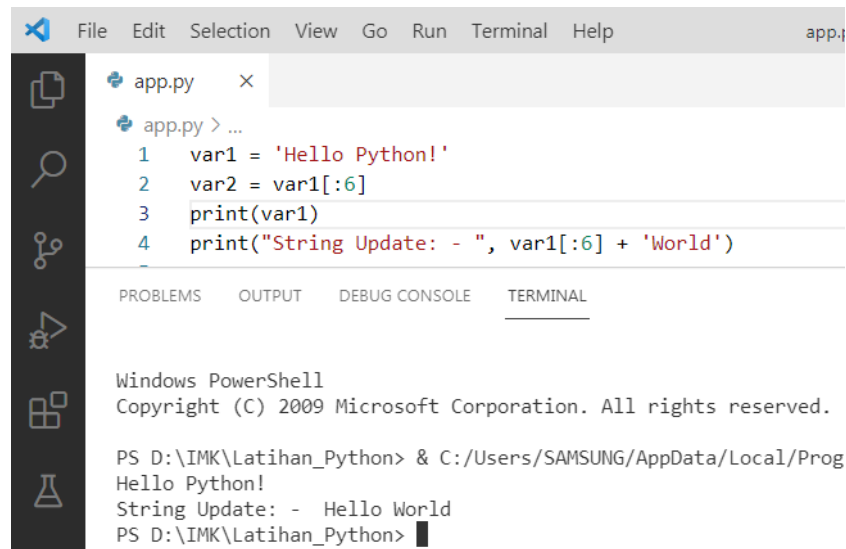
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs
var1[0] H
var2[2:6]: love
PS D:\IMK\Latihan_Python> 
```

Mengupdate String

String adalah tipe data immutable, artinya tidak bisa diubah. Untuk mengupdate string, kita perlu memberikan nilai variabel string lama ke string yang baru. Nilai yang baru adalah nilai string lama yang sudah diupdate.



```
File Edit Selection View Go Run Terminal Help app.py
app.py x
app.py > ...
1 var1 = 'Hello Python!'
2 var2 = var1[:6]
3 print(var1)
4 print("String Update: - ", var1[:6] + 'World')

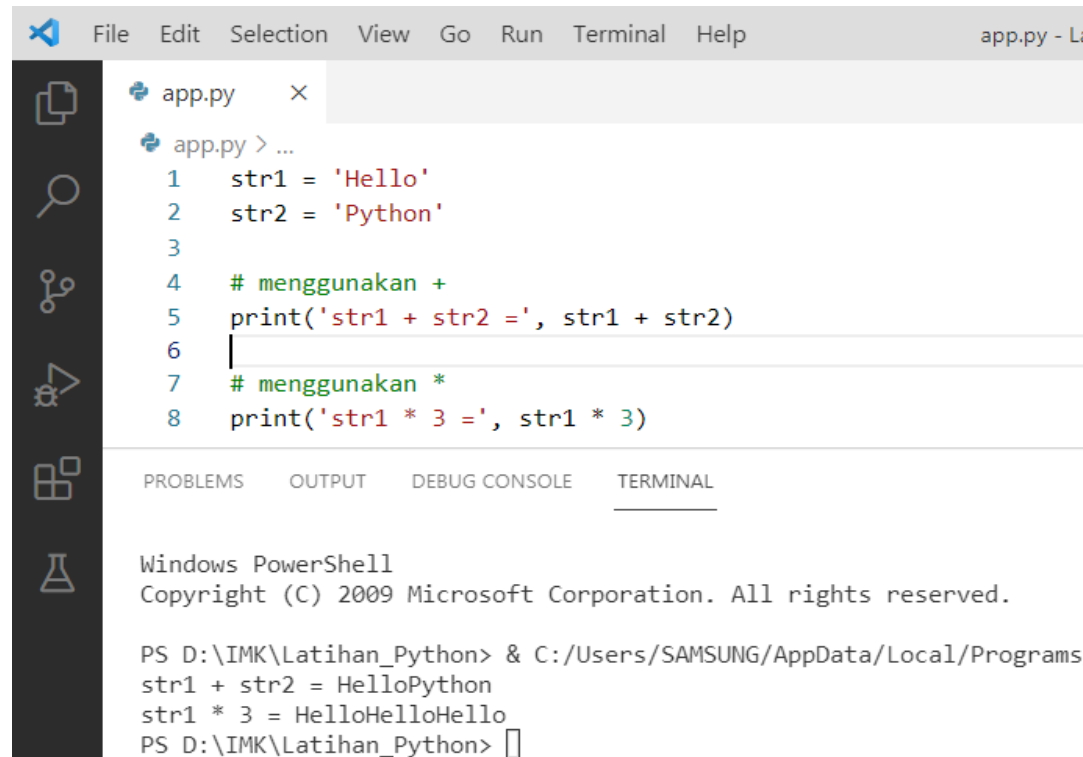
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Prog
Hello Python!
String Update: - Hello World
PS D:\IMK\Latihan_Python>
```

Menggabung String

Kita bisa menggabungkan dua atau lebih string menjadi satu dengan menggunakan operator `+`. Selain itu kita juga bisa melipatgandakan string menggunakan operator `*`.



```
File Edit Selection View Go Run Terminal Help app.py - La
app.py x
app.py > ...
1 str1 = 'Hello'
2 str2 = 'Python'
3
4 # menggunakan +
5 print('str1 + str2 =', str1 + str2)
6
7 # menggunakan *
8 print('str1 * 3 =', str1 * 3)

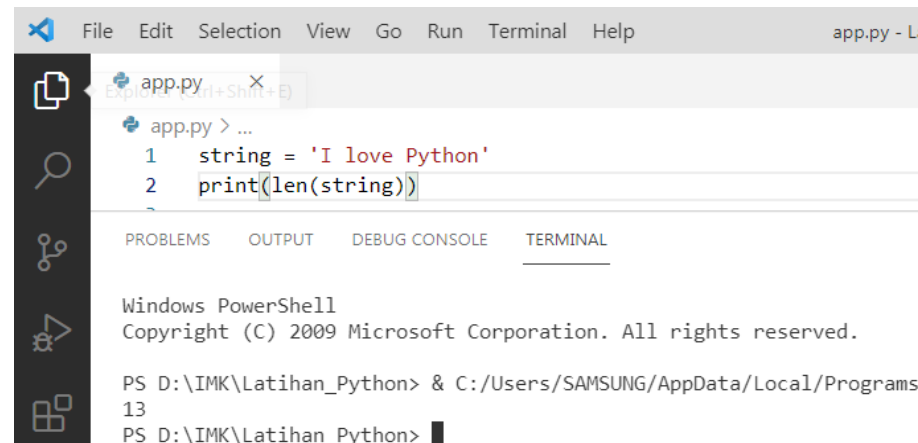
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs,
str1 + str2 = HelloPython
str1 * 3 = HelloHelloHello
PS D:\IMK\Latihan_Python> 
```

Mengetahui Panjang String

Untuk mengetahui panjang dari string, kita bisa menggunakan fungsi `len()`.



```
File Edit Selection View Go Run Terminal Help app.py - La
app.py x
app.py > ...
1 string = 'I love Python'
2 print(len(string))

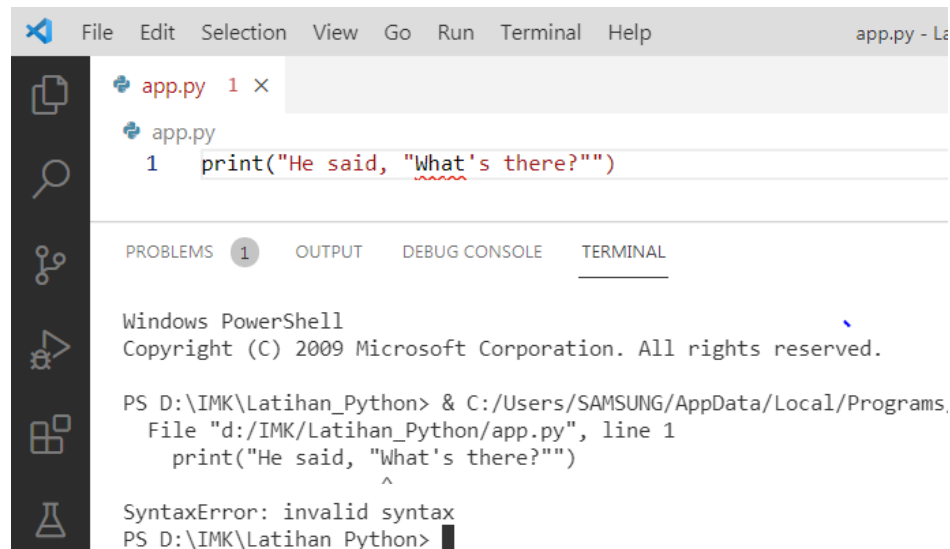
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs,
13
PS D:\IMK\Latihan_Python> 
```

Karakter Escape

Kalau kita hendak mencetak string: He said, "What's there?" kita tidak bisa menggunakan tanda kutip tunggal maupun ganda. Bila kita melakukannya, akan muncul pesan error `SyntaxError` karena teks berisi kutip tunggal dan ganda.



```
File Edit Selection View Go Run Terminal Help app.py - Le
app.py 1 x
app.py
1 print("He said, "What's there?")

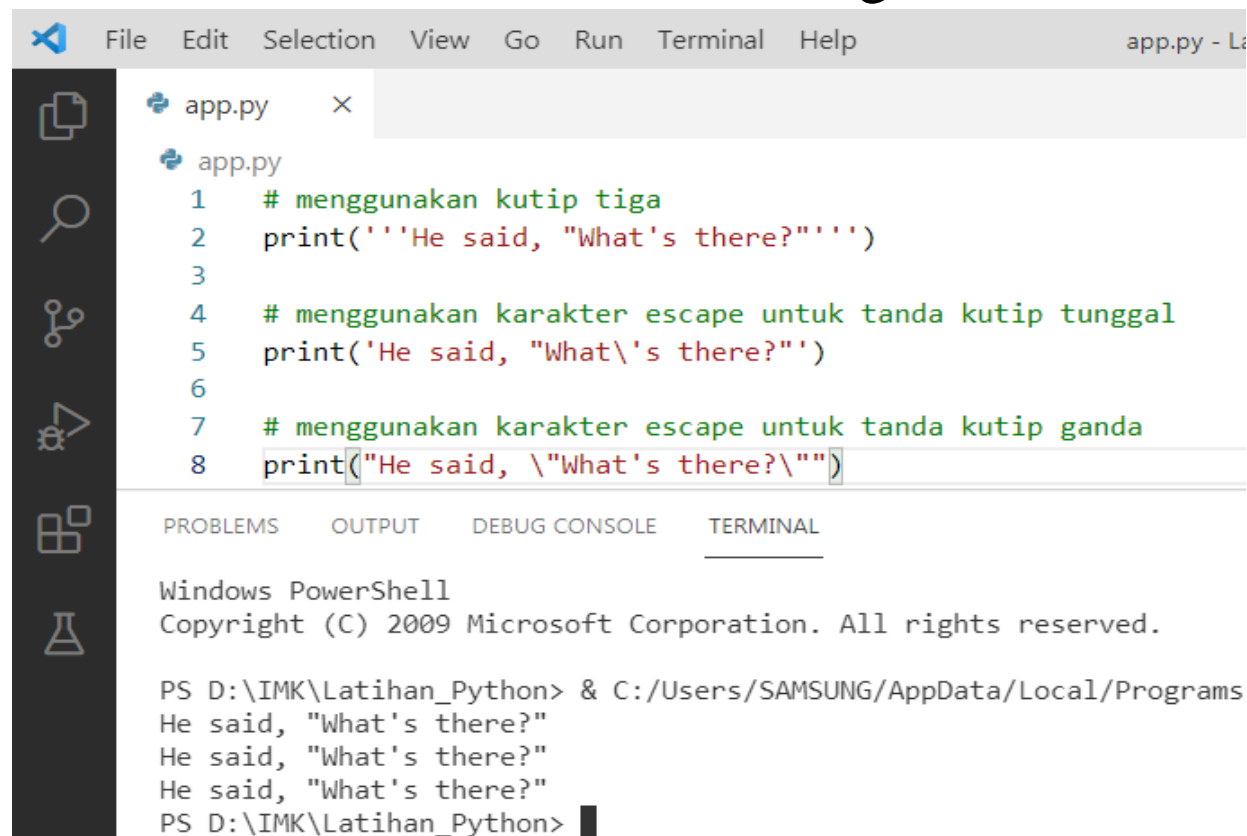
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs.
File "d:/IMK/Latihan_Python/app.py", line 1
    print("He said, "What's there?")
                    ^
SyntaxError: invalid syntax
PS D:\IMK\Latihan_Python>
```

Untuk hal seperti ini kita bisa menggunakan tanda kutip tiga atau menggunakan karakter escape.

Karakter escape dimulai dengan tanda backslash \. Interpreter akan menerjemahkannya dengan cara berbeda dengan string biasa. Solusi untuk error di atas adalah sebagai berikut:



```
File Edit Selection View Go Run Terminal Help app.py - La
app.py
1 # menggunakan kutip tiga
2 print(''He said, "What's there?''')
3
4 # menggunakan karakter escape untuk tanda kutip tunggal
5 print('He said, "What\'s there?")
6
7 # menggunakan karakter escape untuk tanda kutip ganda
8 print("He said, \"What's there?\")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

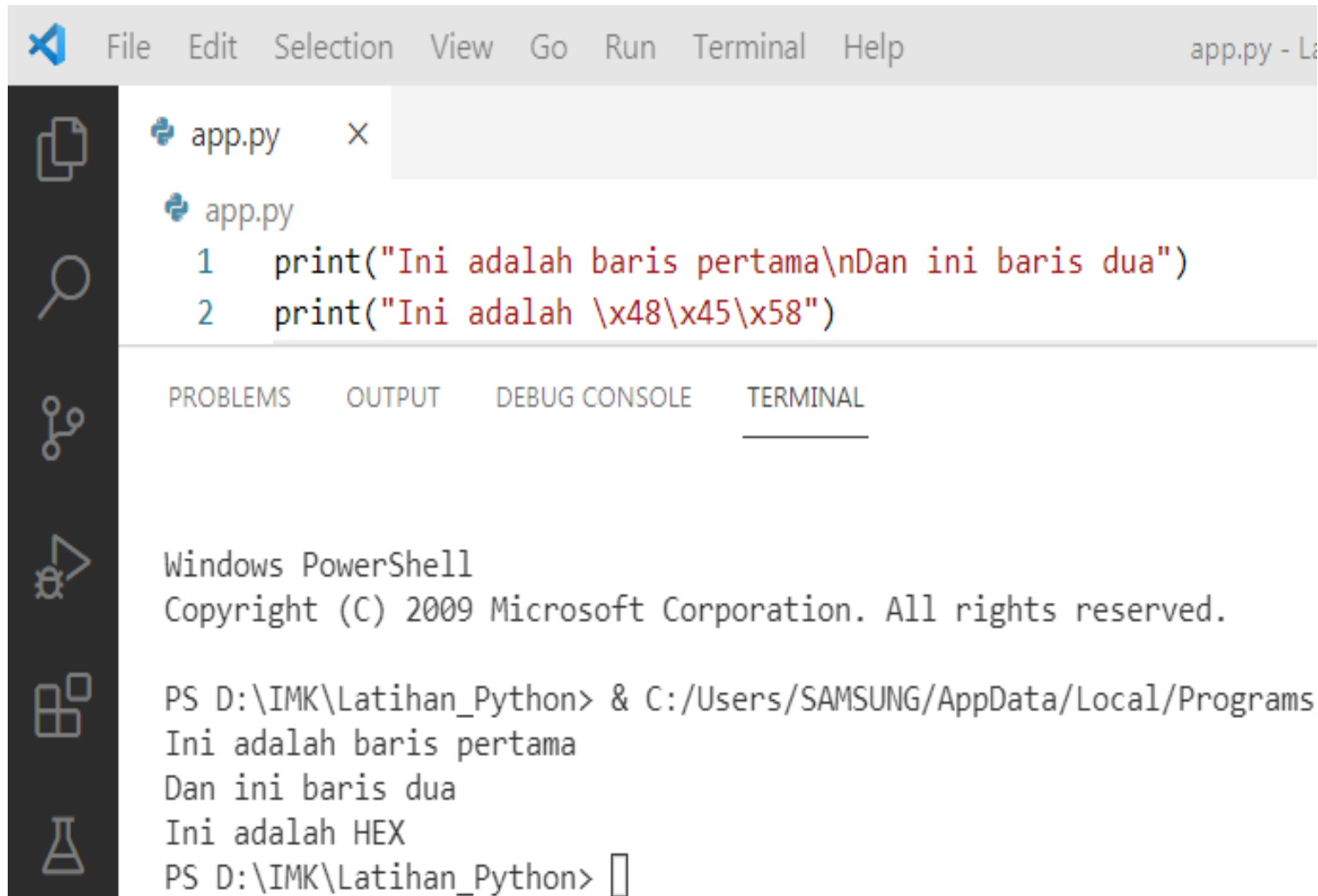
PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs
He said, "What's there?"
He said, "What's there?"
He said, "What's there?"
PS D:\IMK\Latihan_Python>
```

Berikut adalah daftar karakter escape yang didukung oleh Python.

Karakter Escape	Deskripsi
<code>\newline</code>	Backslash dan newline diabaikan
<code>\\</code>	Backslash
<code>\'</code>	Kutip tunggal
<code>\"</code>	Kutip ganda
<code>\a</code>	ASCII bel

<code>\b</code>	ASCII backspace
<code>\f</code>	ASCII formfeed
<code>\n</code>	ASCII linefeed
<code>\r</code>	ASCII carriage return
<code>\t</code>	ASCII tab horizontal
<code>\v</code>	ASCII tab horizontal
<code>\ooo</code>	karakter dengan nilai oktal oo
<code>\xHH</code>	karakter dengan nilai heksadesimal HH

Berikut ini adalah beberapa contohnya:



```
File Edit Selection View Go Run Terminal Help app.py - La
```

```
app.py X
```

```
app.py
```

```
1 print("Ini adalah baris pertama\nDan ini baris dua")
```

```
2 print("Ini adalah \x48\x45\x58")
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
```

```
Windows PowerShell
```

```
Copyright (C) 2009 Microsoft Corporation. All rights reserved.
```

```
PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs
```

```
Ini adalah baris pertama
```

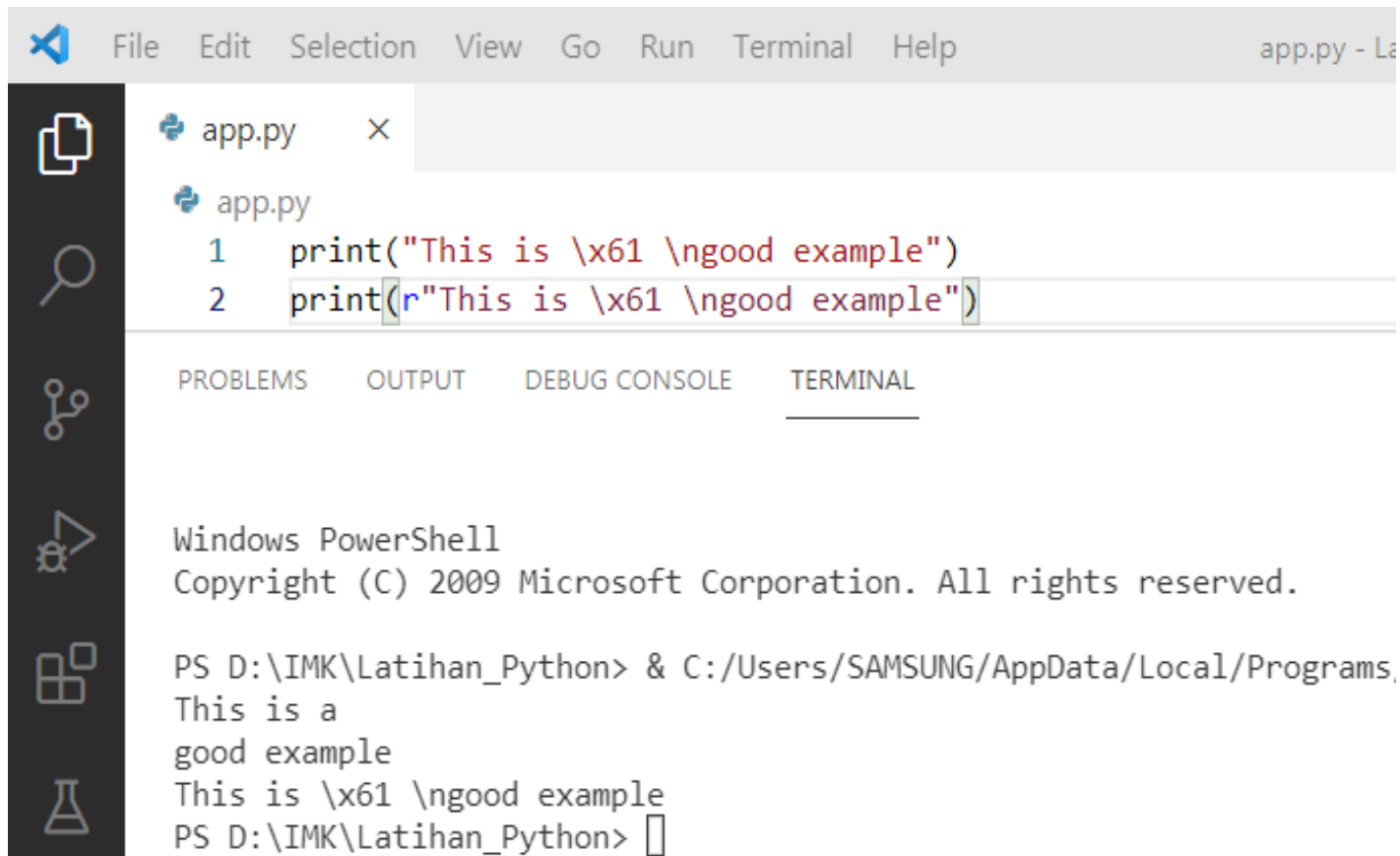
```
Dan ini baris dua
```

```
Ini adalah HEX
```

```
PS D:\IMK\Latihan_Python> 
```

Raw String untuk Mengabaikan Karakter Escape

Kadang kala kita perlu untuk mengabaikan karakter escape yang ada dalam string. Kita bisa melakukannya dengan meletakkan huruf r atau R sebelum tanda kutip string.



```
File Edit Selection View Go Run Terminal Help app.py - La
app.py x
app.py
1 print("This is \x61 \ngood example")
2 print(r"This is \x61 \ngood example")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs.
This is a
good example
This is \x61 \ngood example
PS D:\IMK\Latihan_Python> 
```

Mengatur Format String

Ada dua cara melakukan format pada string. Pertama dengan menggunakan fungsi `format()`, dan kedua dengan menggunakan cara lama (menggunakan `%`).

Metode `format()`

Memformat string dengan fungsi `format()` dibuat dengan menggunakan tanda `{ }` sebagai placeholder atau posisi substring yang akan digantikan. Kita biasa menggunakan argumen posisi atau kata kunci untuk menunjukkan urutan dari substring.

```
app.py x
app.py > ...
1  # menggunakan posisi default
2  default_order = "{} , {} dan {}".format("Budi", "Galih","Ratna")
3  print('\n--- Urutan default ---')
4  print(default_order)
5
6  # menggunakan argument posisi
7  positional_order = "{1}, {0} dan {2}".format("Budi", "Galih","Ratna")
8  print('\n--- Urutan berdasarkan posisi ---')
9  print(positional_order)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell

Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs/Python/Python38-

--- Urutan default ---

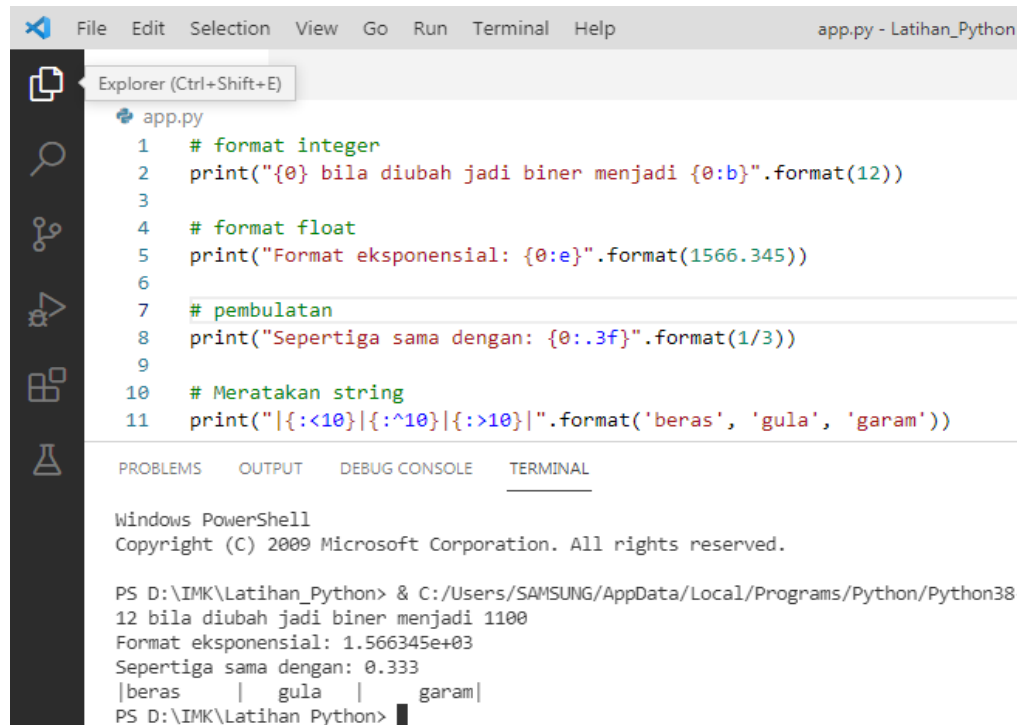
Budi, Galih dan Ratna

--- Urutan berdasarkan posisi ---

Galih, Budi dan Ratna

PS D:\IMK\Latihan_Python> █

Metode `format()` dapat memiliki spesifikasi format opsional. Misalnya, kita bisa menggunakan tanda `<` untuk rata kiri, `>` untuk rata kanan, `^` untuk rata tengah, dan sebagainya.



```

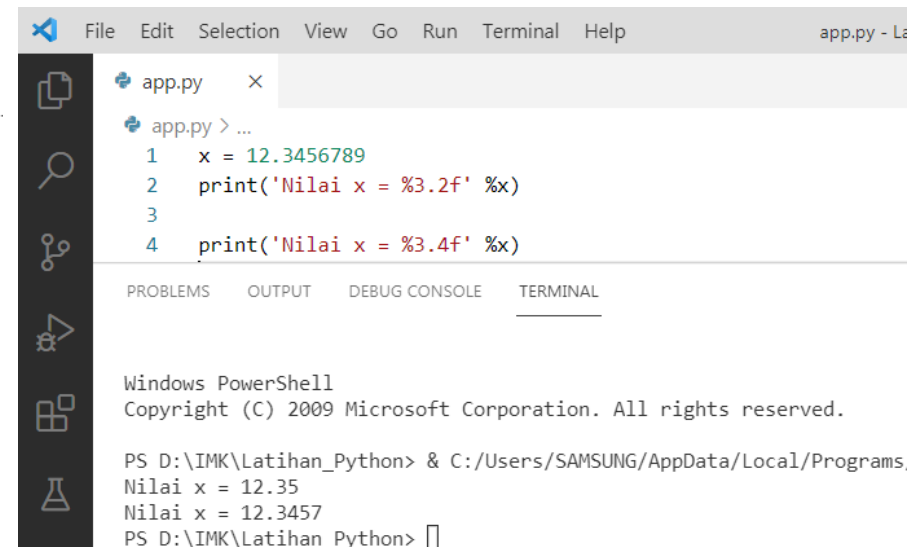
1  # format integer
2  print("{0} bila diubah jadi biner menjadi {0:b}".format(12))
3
4  # format float
5  print("Format eksponensial: {0:e}".format(1566.345))
6
7  # pembulatan
8  print("Sepertiga sama dengan: {0:.3f}".format(1/3))
9
10 # Meratakan string
11 print("|{:<10}|{: ^10}|{:>10}|".format('beras', 'gula', 'garam'))

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs/Python/Python38-
12 bila diubah jadi biner menjadi 1100
Format eksponensial: 1.566345e+03
Sepertiga sama dengan: 0.333
|beras | gula | garam|
PS D:\IMK\Latihan_Python>



```

1  x = 12.3456789
2  print('Nilai x = %3.2f' %x)
3
4  print('Nilai x = %3.4f' %x)

```

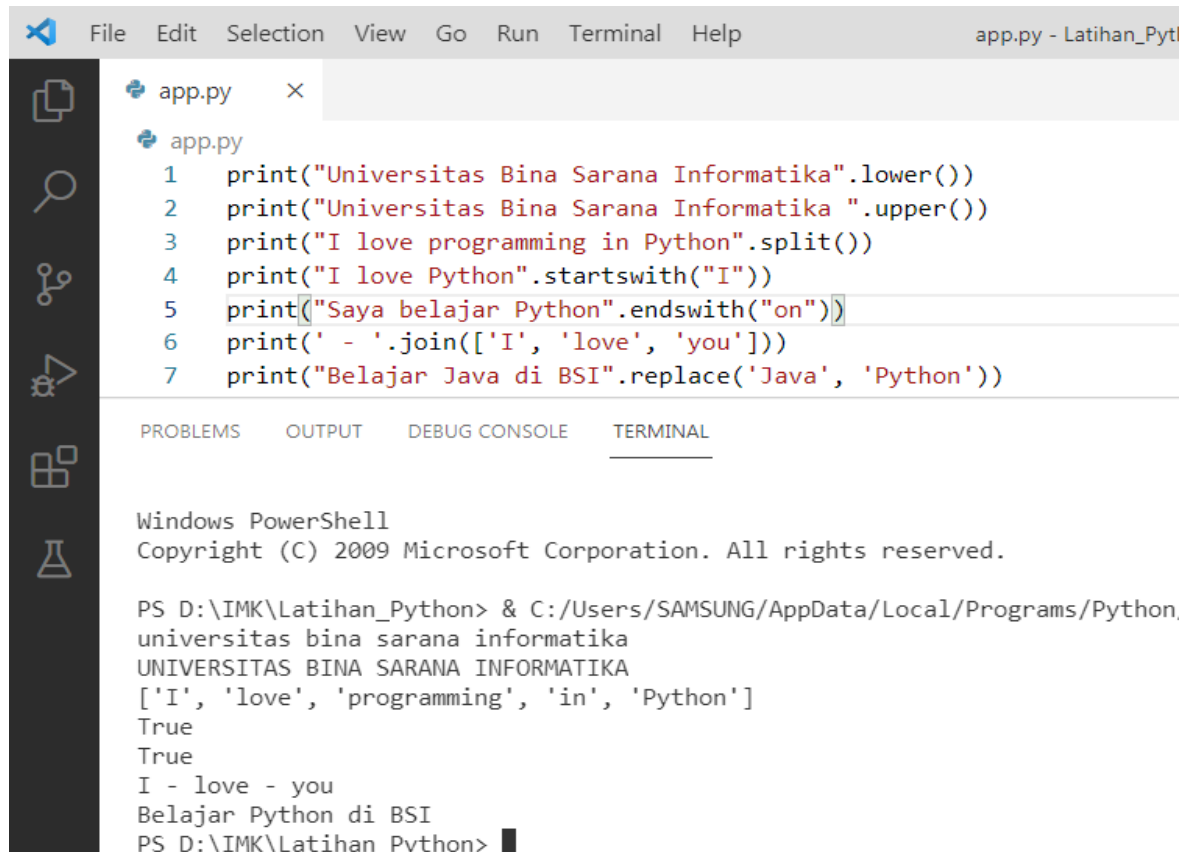
PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs,
Nilai x = 12.35
Nilai x = 12.3457
PS D:\IMK\Latihan_Python>

Metode / Fungsi Bawaan String

String memiliki banyak fungsi bawaan. `format()` yang kita bahas di atas hanya salah satu darinya. Fungsi atau metode lainnya yang sering dipakai adalah `join()`, `lower()`, `upper()`, `split()`, `startswith()`, `endswith()`, `replace()` dan lain sebagainya.



```
File Edit Selection View Go Run Terminal Help app.py - Latihan_Pyt
app.py x
app.py
1 print("Universitas Bina Sarana Informatika".lower())
2 print("Universitas Bina Sarana Informatika ".upper())
3 print("I love programming in Python".split())
4 print("I love Python".startswith("I"))
5 print("Saya belajar Python".endswith("on"))
6 print(' - '.join(['I', 'love', 'you']))
7 print("Belajar Java di BSI".replace('Java', 'Python'))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs/Python,
universitas bina sarana informatika
UNIVERSITAS BINA SARANA INFORMATIKA
['I', 'love', 'programming', 'in', 'Python']
True
True
I - love - you
Belajar Python di BSI
PS D:\IMK\Latihan_Python>
```

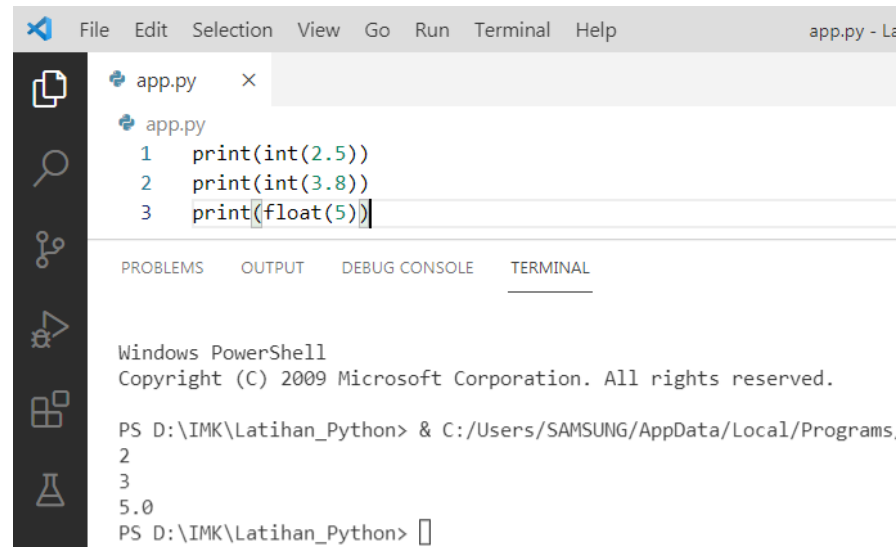

Bilangan (Number)

Bilangan (number) adalah salah satu tipe data dasar di Python. Python mendukung bilangan bulat (integer), bilangan pecahan (float), dan bilangan kompleks (complex). Masing – masing diwakili oleh kelas int, float, dan complex. Integer adalah bilangan bulat, yaitu bilangan yang tidak mempunyai koma. Contohnya 1, 2, 100, -30, -5, 99999, dan lain sebagainya. Panjang integer di python tidak dibatasi jumlah digitnya. Selama memori masih cukup, maka sepanjang itulah jumlah digit yang akan ditampilkan.

Float adalah bilangan pecahan atau bilangan berkoma. Contohnya adalah 5.5, 3.9, 72.8, -1.5, -0.7878999, dan lain sebagainya. Panjang angka di belakang koma untuk float ini adalah 15 digit. Bilangan kompleks (complex) adalah bilangan yang terdiri dari dua bagian, yaitu bagian yang real dan bagian yang imajiner. Contohnya adalah $3 + 2j$, $9 - 5j$, dan lain sebagainya.

Konversi Jenis Bilangan

Kita bisa mengubah jenis bilangan dari int ke float, atau sebaliknya. Mengubah bilangan integer ke float bisa menggunakan fungsi `int(num)` dimana `num` adalah bilangan float.



```
File Edit Selection View Go Run Terminal Help app.py - La
app.py x
app.py
1 print(int(2.5))
2 print(int(3.8))
3 print(float(5))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

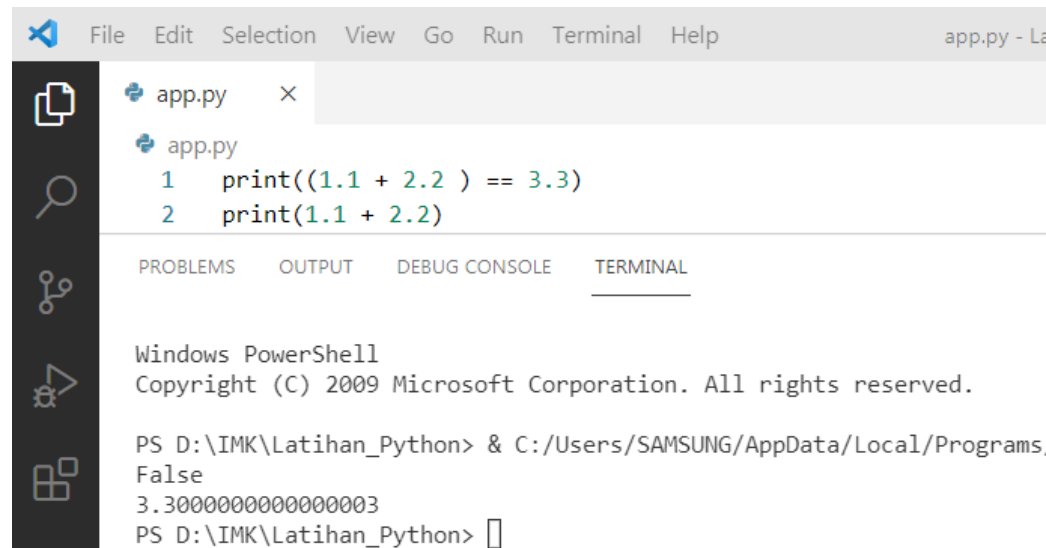
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs,
2
3
5.0
PS D:\IMK\Latihan_Python> 
```

Pada saat kita mengubah float ke integer, bilangan dibulatkan ke bawah. Sebaliknya saat kita mengubah integer ke float, maka bilangan bulat akan menjadi bilangan berkoma.

Python Decimal

Ada kalanya perhitungan menggunakan float di Python membuat kita terkejut. Kita tahu bahwa 1.1 ditambah 2.2 hasilnya adalah 3.3. Tapi pada saat kita lakukan dengan Python, maka hasilnya berbeda.



```
File Edit Selection View Go Run Terminal Help app.py - Le
app.py x
app.py
1 print((1.1 + 2.2) == 3.3)
2 print(1.1 + 2.2)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

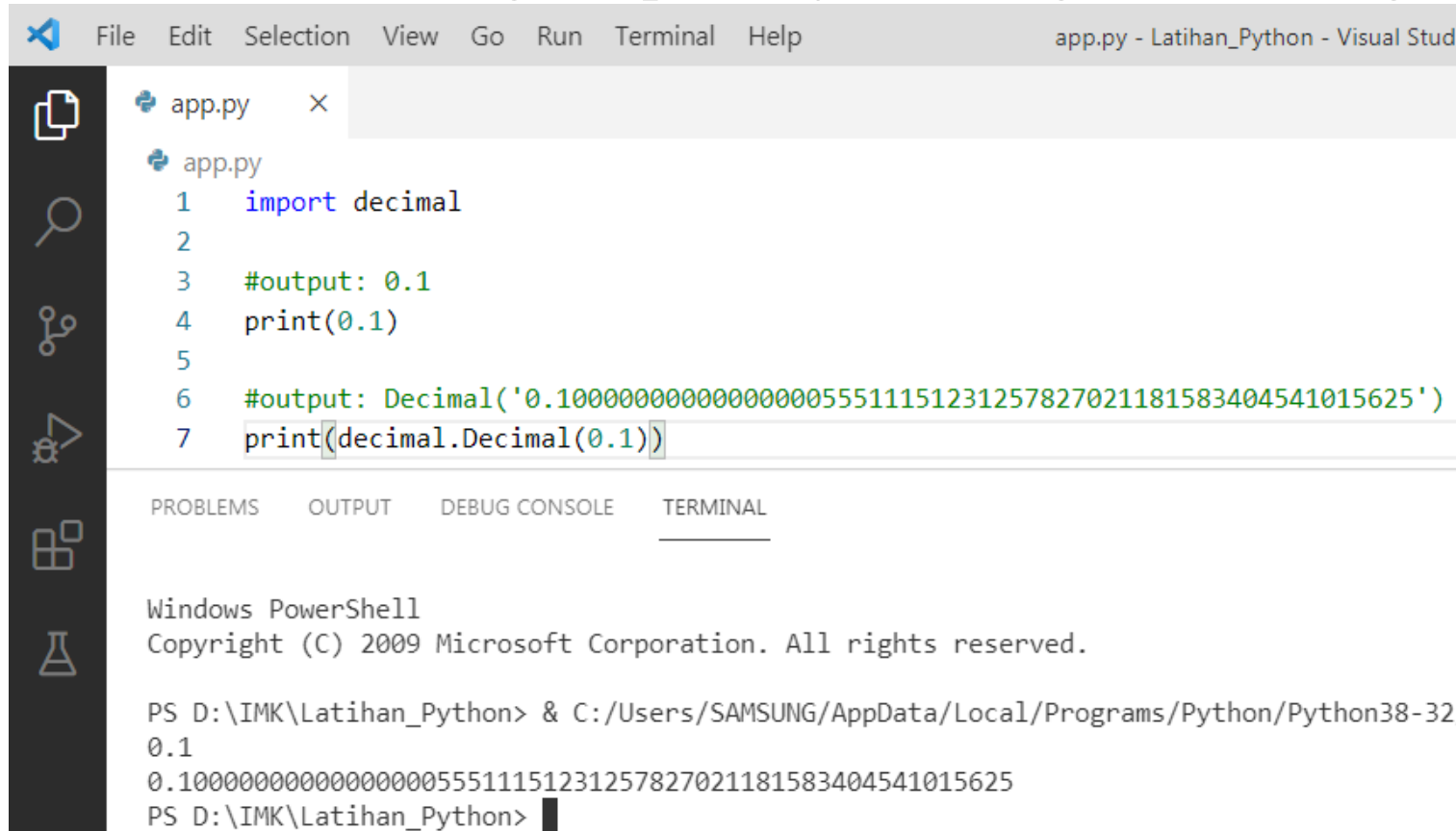
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs,
False
3.3000000000000003
PS D:\IMK\Latihan_Python> 
```

Mengapa terjadi demikian?

Hal ini terjadi karena bilangan dalam komputer disimpan dalam bentuk digit 0 atau 1. Bila padanan digitnya tidak sesuai, maka bilangan float seperti 0.1 dalam bilangan biner akan menjadi pecahan yang sangat panjang yaitu 0.000110011001100110011... dan komputer kita hanya akan menyimpan panjang yang terbatas. Hal inilah yang menyebabkan terjadinya masalah seperti pada contoh di atas.

Untuk menangani hal seperti itu, kita bisa menggunakan modul bawaan Python yaitu modul decimal. Float hanya memiliki presisi sampai 15 digit di belakang koma, sementara dengan modul decimal kita bisa mengatur presisi jumlah digit di belakang koma.



```
File Edit Selection View Go Run Terminal Help app.py - Latihan_Python - Visual Stud

app.py x
app.py
1 import decimal
2
3 #output: 0.1
4 print(0.1)
5
6 #output: Decimal('0.100000000000000055511151231257827021181583404541015625')
7 print(decimal.Decimal(0.1))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs/Python/Python38-32.
0.1
0.100000000000000055511151231257827021181583404541015625
PS D:\IMK\Latihan_Python> |
```

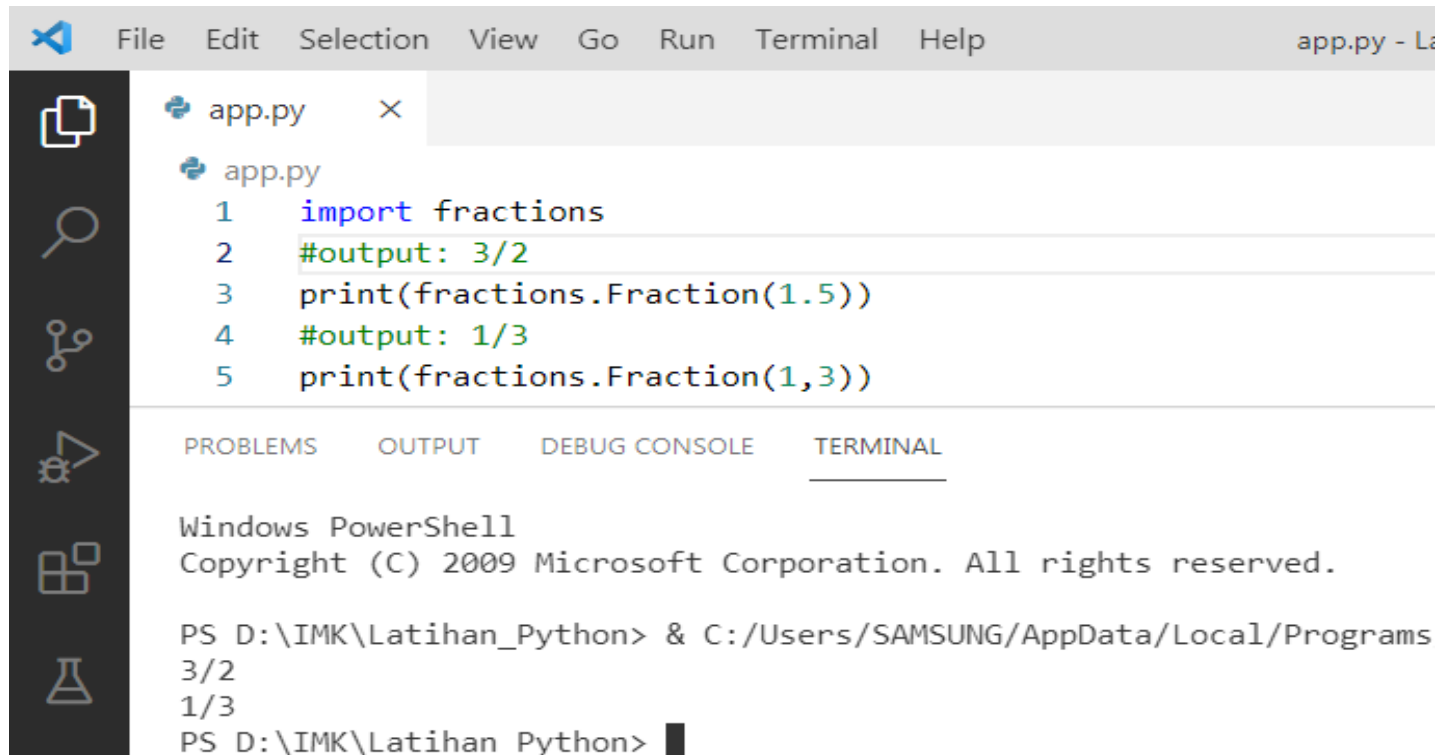
Modul ini juga membuat kita bisa melakukan perhitungan seperti di sekolah.

Kapan Saatnya Menggunakan Decimal Dibanding Float? Kita lebih baik menggunakan Decimal dalam kasus:

- Saat kita ingin membuat aplikasi keuangan yang membutuhkan presisi desimal yang pasti
- Saat kita ingin mengontrol tingkat presisi yang diperlukan
- Saat kita ingin menerapkan perkiraan berapa digit decimal yang signifikan
- Saat kita ingin melakukan operasi perhitungan sama persis dengan yang kita lakukan di sekolah

Bilangan Pecahan

Python menyediakan modul fractions untuk mengoperasikan bilangan pecahan. Pecahan adalah bilangan yang memiliki pembilang dan penyebut, misalnya $3/2$. Perhatikan contoh berikut:

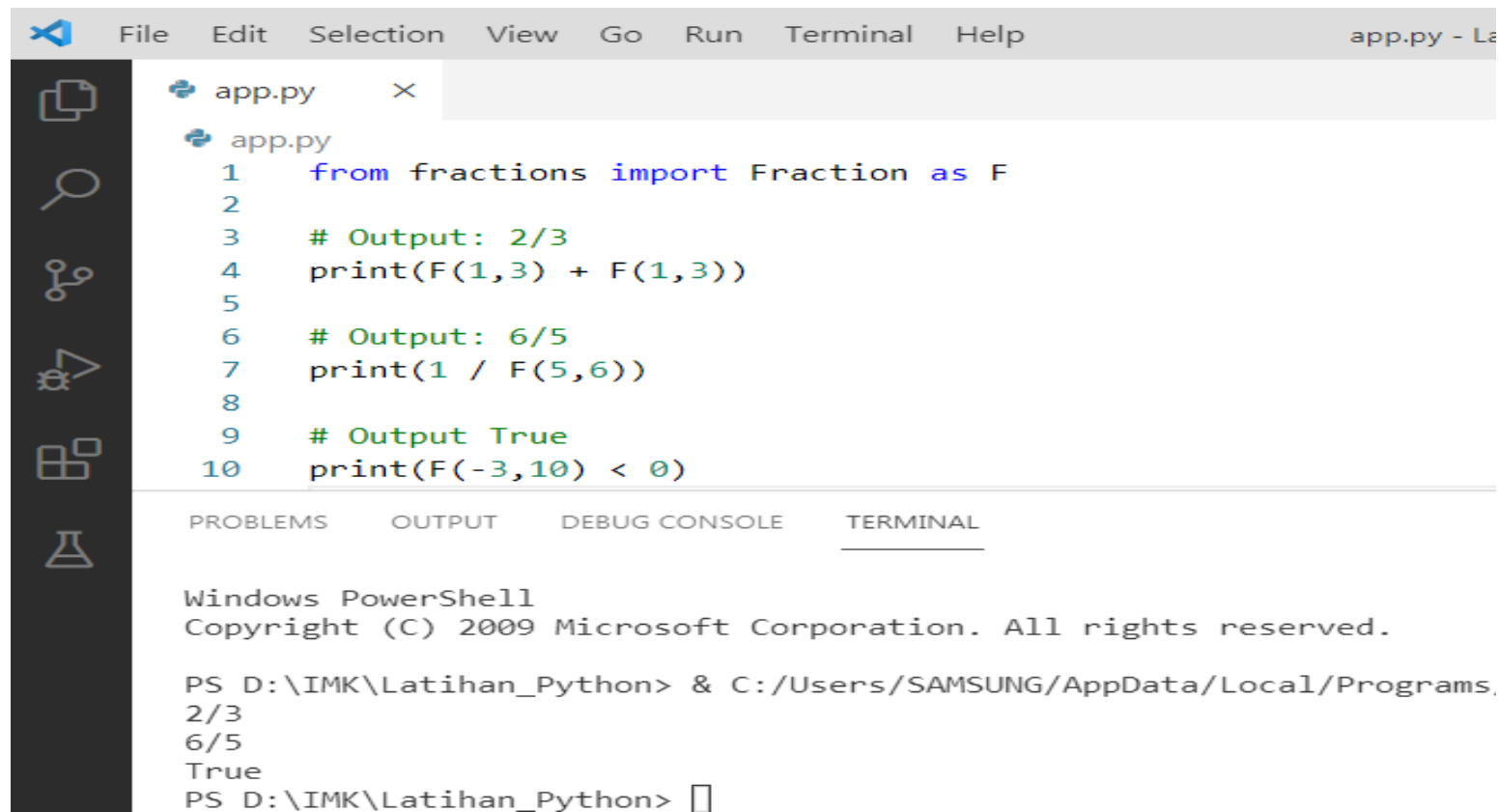


```
File Edit Selection View Go Run Terminal Help app.py - La
app.py x
app.py
1 import fractions
2 #output: 3/2
3 print(fractions.Fraction(1.5))
4 #output: 1/3
5 print(fractions.Fraction(1,3))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs
3/2
1/3
PS D:\IMK\Latihan_Python> 
```

Operasi dasar seperti penjumlahan atau pembagian pecahan juga bisa dilakukan dengan modul fractions ini



```
File Edit Selection View Go Run Terminal Help app.py - La
app.py
app.py
1  from fractions import Fraction as F
2
3  # Output: 2/3
4  print(F(1,3) + F(1,3))
5
6  # Output: 6/5
7  print(1 / F(5,6))
8
9  # Output True
10 print(F(-3,10) < 0)

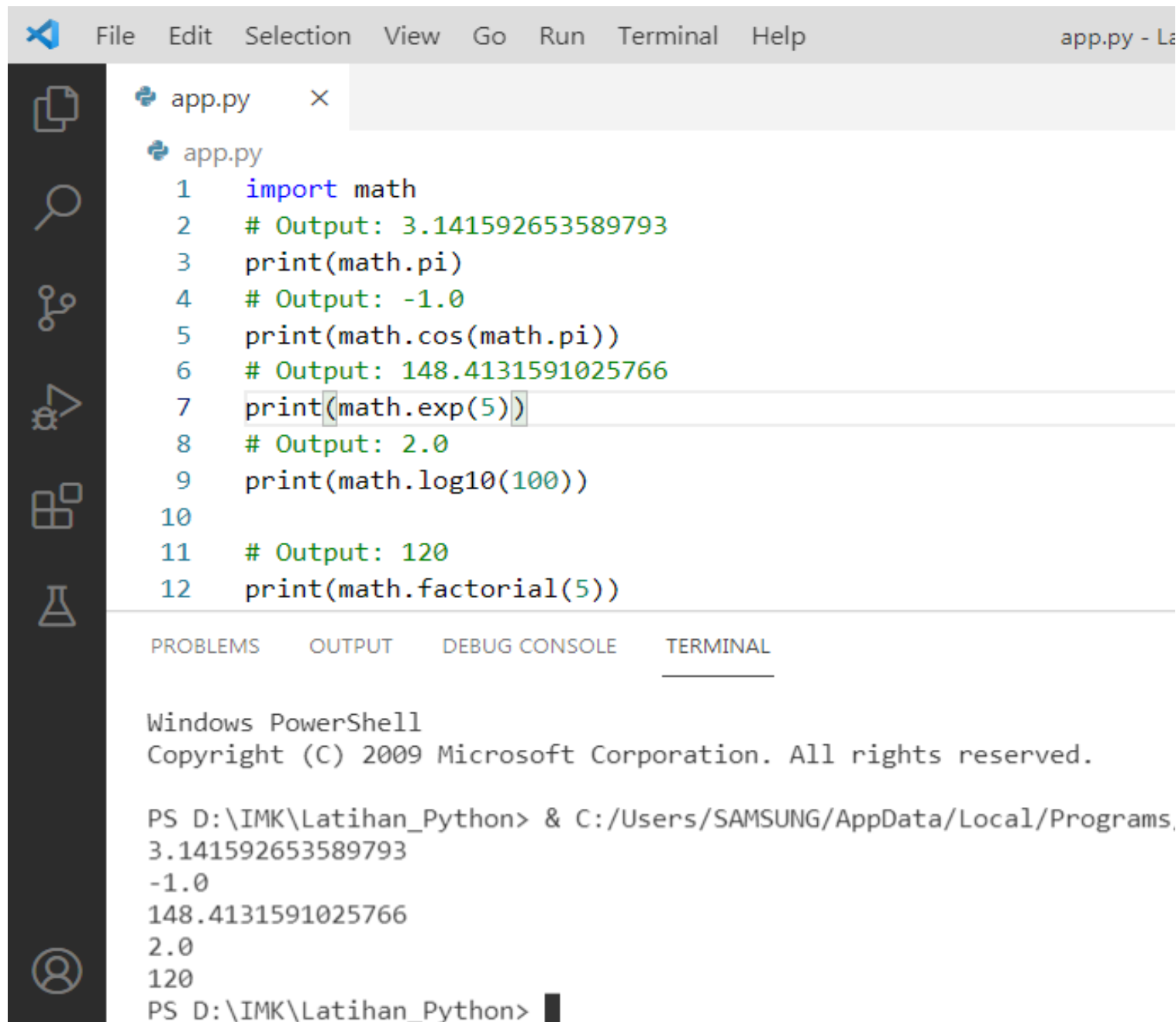
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs,
2/3
6/5
True
PS D:\IMK\Latihan_Python> █
```

Matematika dengan Python

Python menyediakan modul math melakukan hal yang berbau matematis seperti trigonometri, logaritma, probabilitas, statistik, dan lain – lain.



The screenshot shows a Visual Studio Code editor window with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The right side of the window shows the file name 'app.py - La'. The editor area displays the following Python code in 'app.py':

```
1 import math
2 # Output: 3.141592653589793
3 print(math.pi)
4 # Output: -1.0
5 print(math.cos(math.pi))
6 # Output: 148.4131591025766
7 print(math.exp(5))
8 # Output: 2.0
9 print(math.log10(100))
10
11 # Output: 120
12 print(math.factorial(5))
```

Below the editor, the 'TERMINAL' tab is active, showing the output of the program in a Windows PowerShell environment:

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs,
3.141592653589793
-1.0
148.4131591025766
2.0
120
PS D:\IMK\Latihan_Python> █
```


Operator

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.

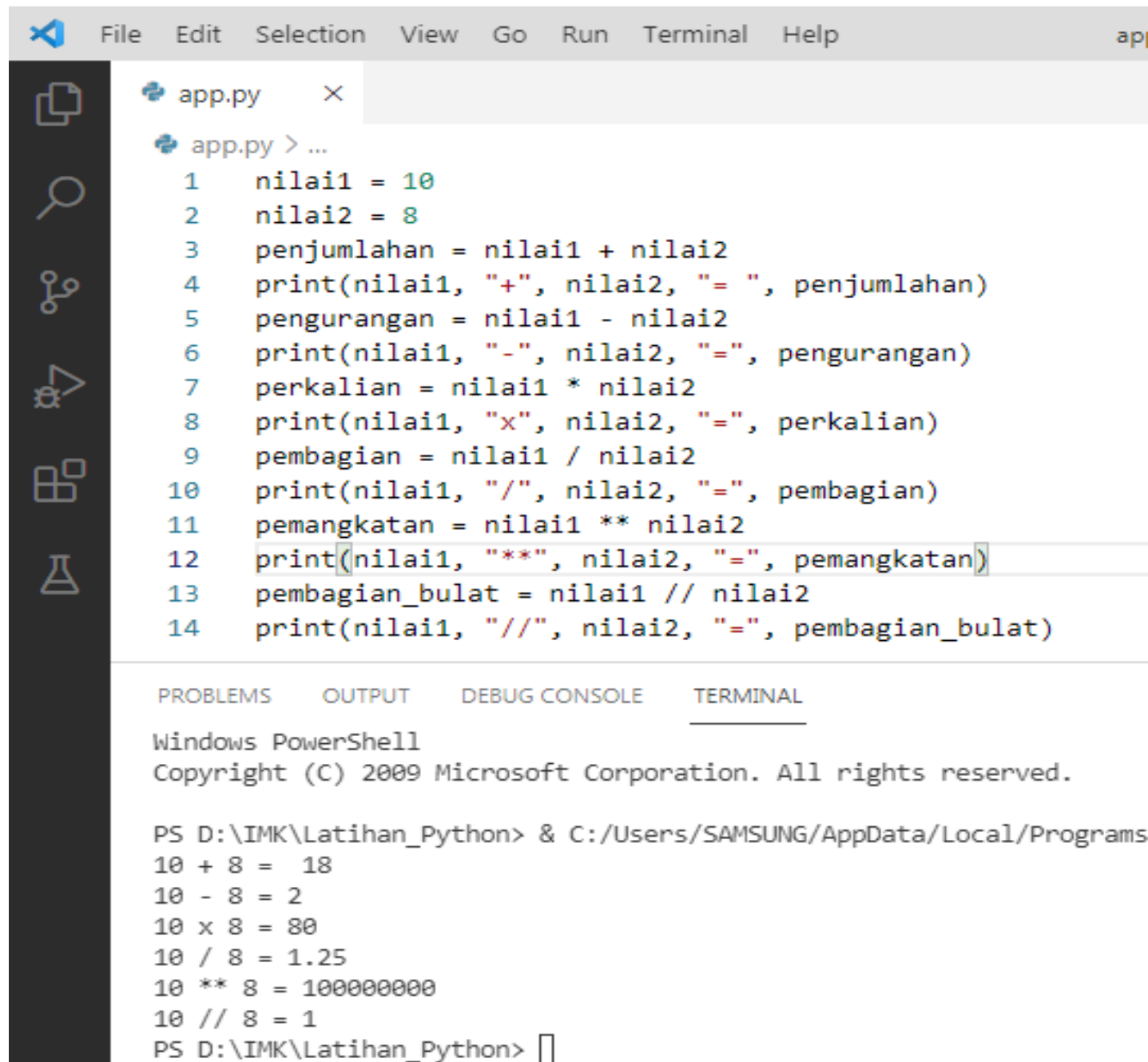
Python memiliki sejumlah operator, yaitu:

- Operator Aritmatika
- Operator Perbandingan
- Operator Penugasan
- Operator Logika
- Operator Bitwise
- Operator Identitas
- Operator Keanggotaan

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.

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

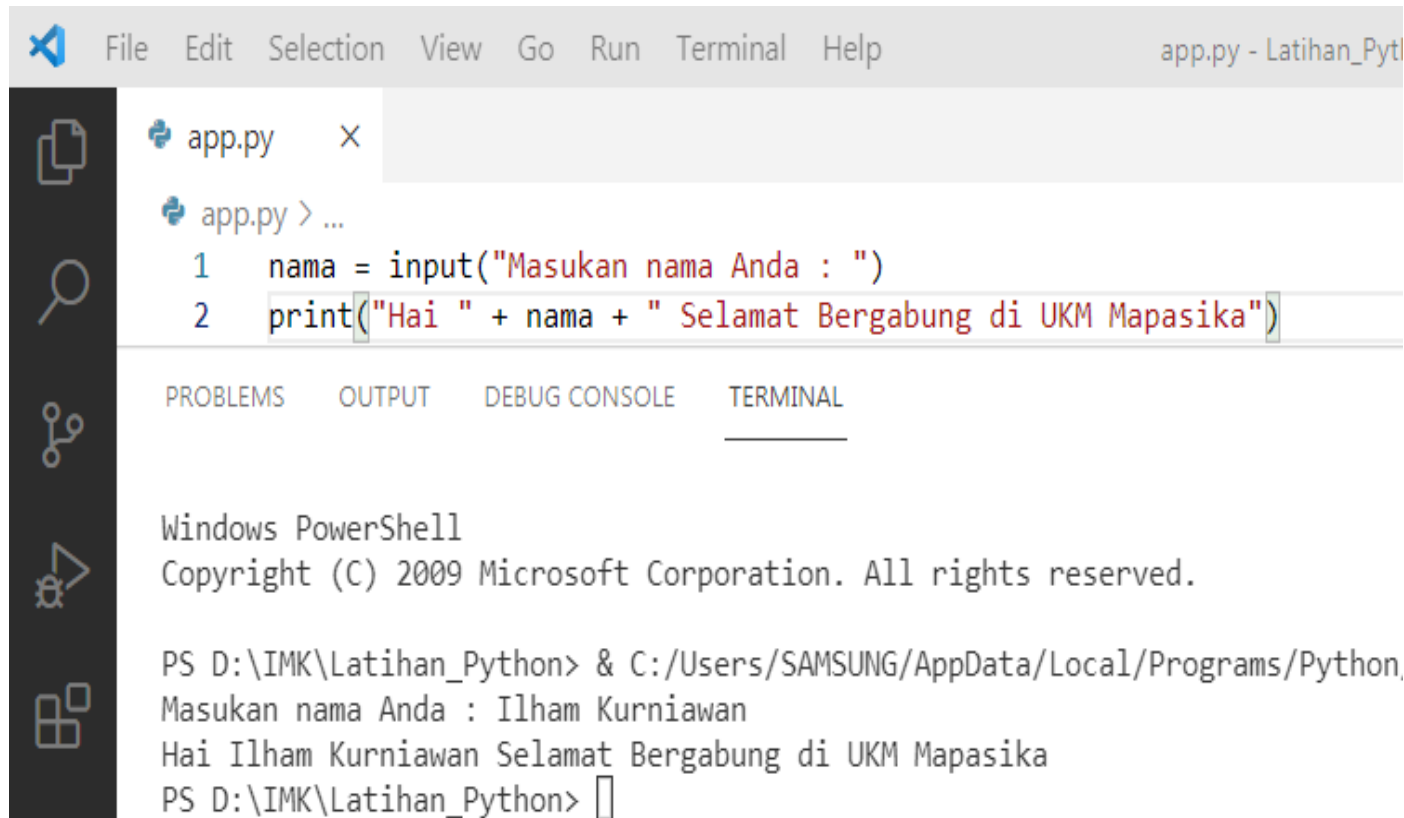


```
File Edit Selection View Go Run Terminal Help app.py
app.py ×
app.py > ...
1  nilai1 = 10
2  nilai2 = 8
3  penjumlahan = nilai1 + nilai2
4  print(nilai1, "+", nilai2, "=", penjumlahan)
5  pengurangan = nilai1 - nilai2
6  print(nilai1, "-", nilai2, "=", pengurangan)
7  perkalian = nilai1 * nilai2
8  print(nilai1, "x", nilai2, "=", perkalian)
9  pembagian = nilai1 / nilai2
10 print(nilai1, "/", nilai2, "=", pembagian)
11 pemangkatan = nilai1 ** nilai2
12 print(nilai1, "**", nilai2, "=", pemangkatan)
13 pembagian_bulat = nilai1 // nilai2
14 print(nilai1, "//", nilai2, "=", pembagian_bulat)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs
10 + 8 = 18
10 - 8 = 2
10 x 8 = 80
10 / 8 = 1.25
10 ** 8 = 100000000
10 // 8 = 1
PS D:\IMK\Latihan_Python> 
```

Kita juga bisa menentukan suatu variabel sesuai dengan keinginan kita dengan menggunakan fungsi **input()** seperti contoh berikut :



```
File Edit Selection View Go Run Terminal Help app.py - Latihan_Pyt

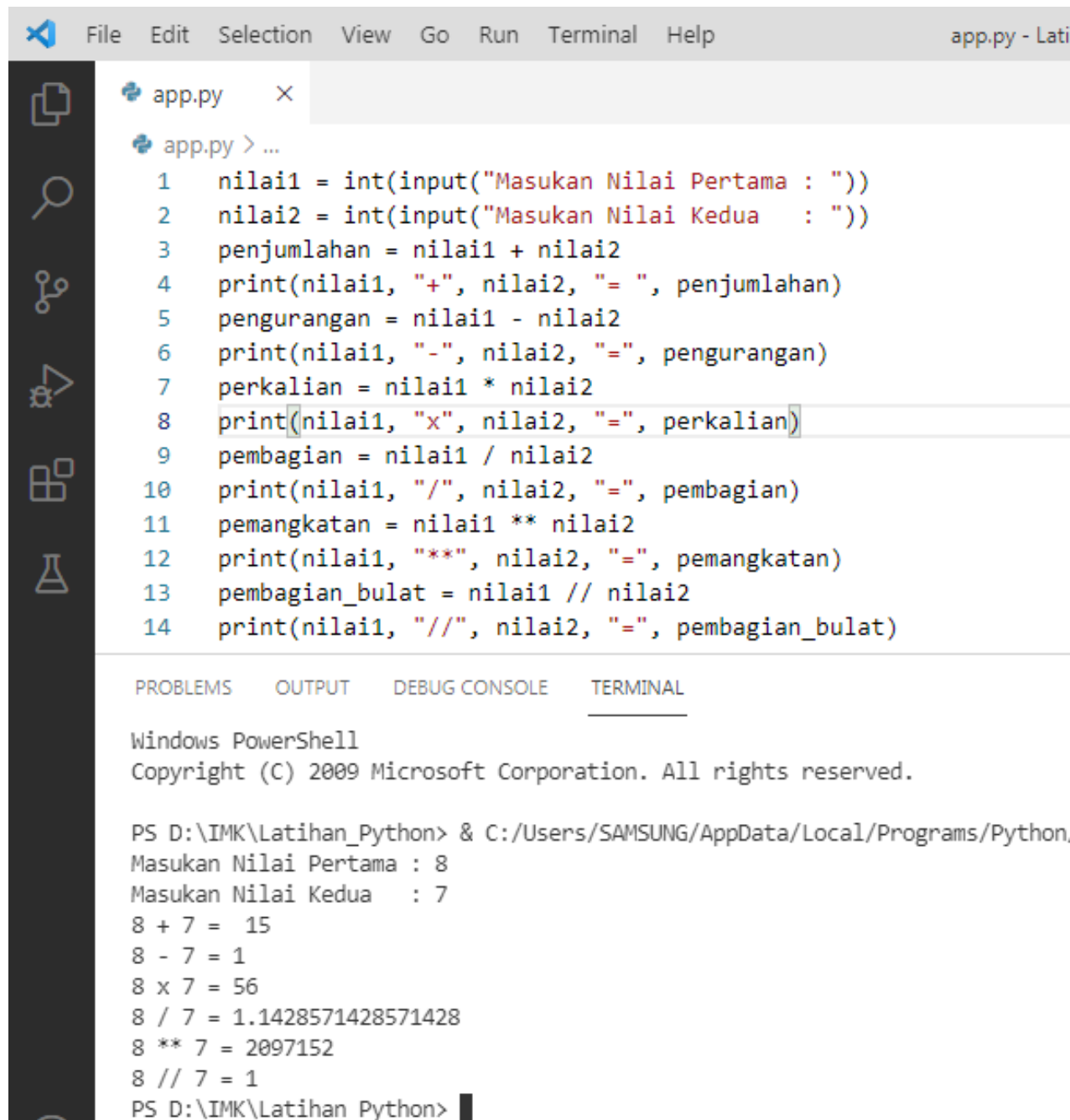
app.py x
app.py > ...
1 nama = input("Masukan nama Anda : ")
2 print("Hai " + nama + " Selamat Bergabung di UKM Mapasika")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs/Python,
Masukan nama Anda : Ilham Kurniawan
Hai Ilham Kurniawan Selamat Bergabung di UKM Mapasika
PS D:\IMK\Latihan_Python> 
```

Jika karakter yang akan kita input merupakan sebuah integer maka kita juga harus menambahkan fungsi **int()** seperti pada contoh berikut :



The screenshot shows a Visual Studio Code editor window with a file named 'app.py' open. The code in the editor performs basic arithmetic operations on two input numbers. Below the editor, the 'TERMINAL' tab is active, showing the command prompt output of the script.

```
File Edit Selection View Go Run Terminal Help app.py - Lati

app.py ×
app.py > ...
1  nilai1 = int(input("Masukan Nilai Pertama : "))
2  nilai2 = int(input("Masukan Nilai Kedua  : "))
3  penjumlahan = nilai1 + nilai2
4  print(nilai1, "+", nilai2, "=", penjumlahan)
5  pengurangan = nilai1 - nilai2
6  print(nilai1, "-", nilai2, "=", pengurangan)
7  perkalian = nilai1 * nilai2
8  print(nilai1, "x", nilai2, "=", perkalian)
9  pembagian = nilai1 / nilai2
10 print(nilai1, "/", nilai2, "=", pembagian)
11 pemangkatan = nilai1 ** nilai2
12 print(nilai1, "**", nilai2, "=", pemangkatan)
13 pembagian_bulat = nilai1 // nilai2
14 print(nilai1, "//", nilai2, "=", pembagian_bulat)

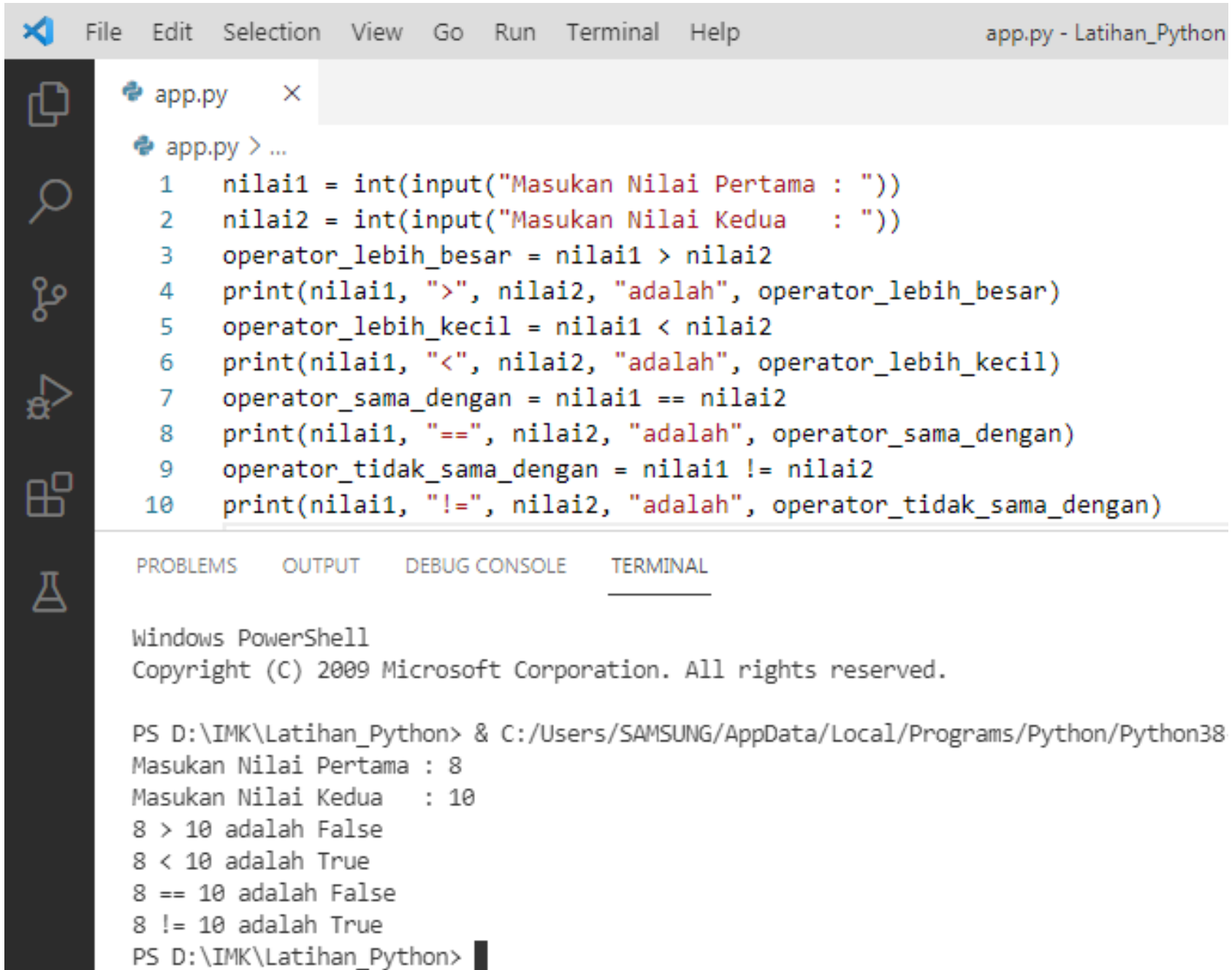
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs/Python,
Masukan Nilai Pertama : 8
Masukan Nilai Kedua  : 7
8 + 7 = 15
8 - 7 = 1
8 x 7 = 56
8 / 7 = 1.1428571428571428
8 ** 7 = 2097152
8 // 7 = 1
PS D:\IMK\Latihan_Python> |
```

Operator Perbandingan

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

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



The screenshot shows a Visual Studio Code editor window with the title bar "app.py - Latihan_Python". The editor has a menu bar with "File", "Edit", "Selection", "View", "Go", "Run", "Terminal", and "Help". On the left, there is a sidebar with icons for Explorer, Search, Source Control, Run and Debug, Extensions, and Testing. The main editor area shows a Python file named "app.py" with the following code:

```
1  nilai1 = int(input("Masukan Nilai Pertama : "))
2  nilai2 = int(input("Masukan Nilai Kedua : "))
3  operator_lebih_besar = nilai1 > nilai2
4  print(nilai1, ">", nilai2, "adalah", operator_lebih_besar)
5  operator_lebih_kecil = nilai1 < nilai2
6  print(nilai1, "<", nilai2, "adalah", operator_lebih_kecil)
7  operator_sama_dengan = nilai1 == nilai2
8  print(nilai1, "==", nilai2, "adalah", operator_sama_dengan)
9  operator_tidak_sama_dengan = nilai1 != nilai2
10 print(nilai1, "!=" , nilai2, "adalah", operator_tidak_sama_dengan)
```

Below the editor, there is a panel with tabs for "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", and "TERMINAL". The "TERMINAL" tab is selected, showing a Windows PowerShell prompt. The output of the script is as follows:

```
PS D:\IMK\Latihan_Python> & C:/Users/SAMSUNG/AppData/Local/Programs/Python/Python38
Masukan Nilai Pertama : 8
Masukan Nilai Kedua : 10
8 > 10 adalah False
8 < 10 adalah True
8 == 10 adalah False
8 != 10 adalah True
PS D:\IMK\Latihan_Python>
```


Operator Penugasan

Operator penugasan adalah operator yang digunakan untuk memberi nilai ke variabel. $a = 7$ adalah contoh operator penugasan yang memberi nilai 7 di kanan ke variabel a yang ada di kiri.

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 $c = c \% a$

Operator Logika

Operator logika adalah operator yang digunakan untuk melakukan operasi logika

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

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 $a = 10$ (0000 1010) dalam biner dan $b = 4$ (0000 0100) dalam biner.

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

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)	a is True
is not	True jika kedua operand tidak identik (tidak merujuk ke objek yang sama)	a is not True

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)

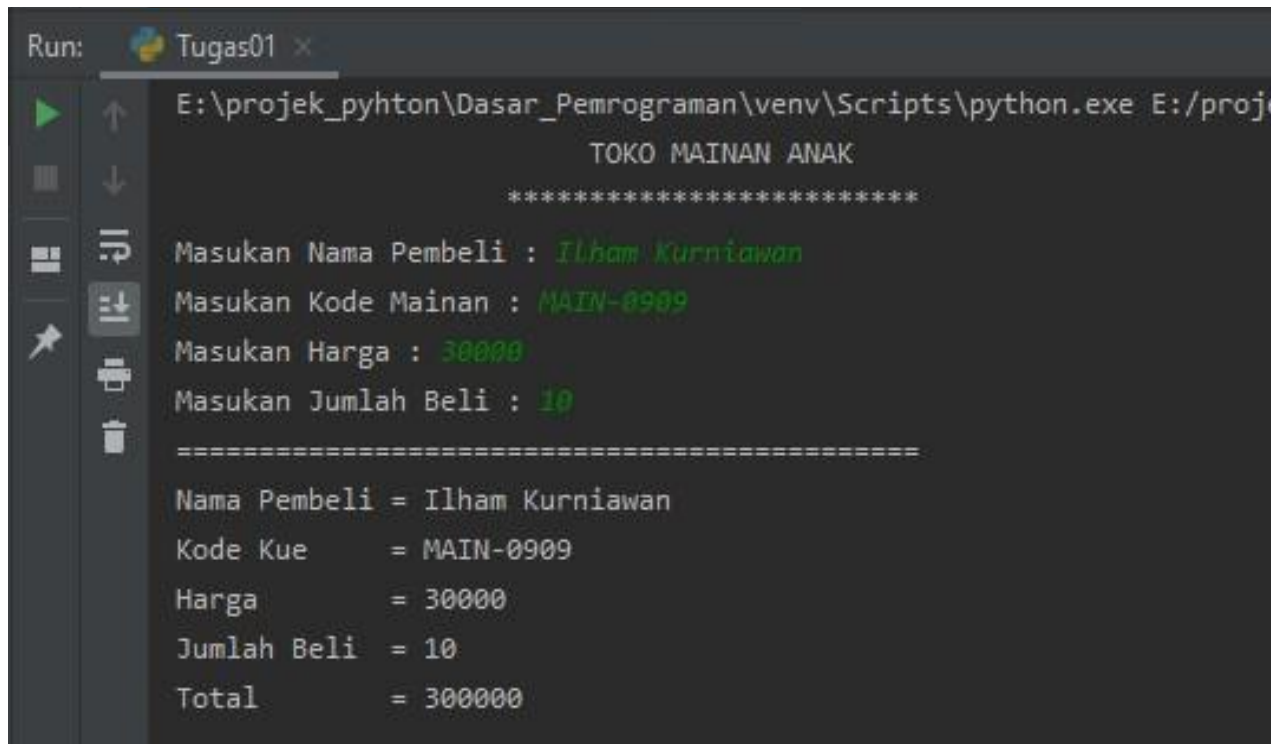
Operator	Penjelasan	Contoh
In	True jika nilai/variabel ditemukan di dalam data	5 in a
not in	True jika nilai/variabel tidak ada di dalam data	5 not in a

Latihan 1

- Buatlah 1 Contoh Operator Penugasan
- Buatlah 1 Contoh Operator Logika
- Buatlah 1 Contoh Operator Bitwise
- Buatlah 1 Contoh Operator Identitas
- Buatlah 1 Contoh Operator Keanggotaan

Latihan 2

- Buatlah program seperti gambar dibawah ini



```
Run: Tugas01 x
E:\projek_pyhton\Dasar_Pemrograman\venv\Scripts\python.exe E:/projek_pyhton\Dasar_Pemrograman\Tugas01.py
TOKO MAINAN ANAK
*****
Masukan Nama Pembeli : Ilham Kurniawan
Masukan Kode Mainan : MAIN-0909
Masukan Harga : 30000
Masukan Jumlah Beli : 10
=====
Nama Pembeli = Ilham Kurniawan
Kode Kue     = MAIN-0909
Harga        = 30000
Jumlah Beli  = 10
Total        = 300000
```

Uji Kompetensi

1. **Individu**
2. **Final Project** (Berkelompok)

Uji Kompetensi(Individu)

1. Untuk Matakuliah **Dasar Pemrograman Tidak Ada UTS Dan UAS**, digantikan **Uji kompetensi (Individu dan Final Project)**
2. Uji Kompetensi Individu dilaksanakan pada pertemuan **12**. Masing-masing mahasiswa diminta mengerjakan soal yang sudah ditentukan.wajib membawa laptop

Uji Kompetensi (Final Projek)

1. Final Project dilakukan di **pertemuan 13-15** dengan ketentuan sebagai berikut :
 - a. Jumlah anggota setiap kelompok adalah 5 anggota (optional) tergantung jumlah mahasiswa pada kelas tersebut.
 - b. Isi dari final project :
Nilai Running Program diambil berdasarkan: (Logika Program, Debuging, penulisan Kode Program, Tampilan output Program)
Alur logika dari proses project yang dibuat
 - b. Masing-masing kelompok membuat paper laporan pembuatan final project
 - c. Program, Paper dan Presentasi di Burning Kedalam CD
 - d. Masing-masing kelompok mempresentasikan hasil final projectnya.
 - e. Presentasi disajikan dengan media presentasi yang isinya berupa alur logika program dan eksekusi running program
 - f. Penilaian di tentukan oleh dosen pengajar diruang kelas
2. Tema Projek di serahkan ke dosen pengajar di **Pertemuan ke 2**
3. Projek sudah bisa di kerjakan setelah di lakukan penyerahan **tema kepada dosen pengajar**
4. **Penilaian dilakukan oleh dosen pengajar ketika presentasi**

Tema Projek UAS :

1. Berbasis Bisnis (Optional):Contoh
 - Penjualan dan Pembelian
 - Pengadaan barang
2. Berbasis Science
 - Science (Bid.Matematika,Fisika,Kimia atau IPA)
- Animasi Edukasi
 - Berbasis Kesehatan(Diagnosa Penyakit)
- 3.Kreatifitas tampilan
- 4.Tema Harus Menarik
- 5.Penilai di Tentukan oleh Dosen Pengajar