**MODUL**

**BIG DATA ANALYTIC**

****

**Disusun oleh :**

**Sri Redjeki, S.Si., M.Kom.**

**Maria Mediatrix S. Kom., M.Eng.**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**AKAKOM**

**YOGYAKARTA**

**2019**

# KATA PENGANTAR

# DAFTAR ISI

[KATA PENGANTAR ii](#_Toc15905388)

[DAFTAR ISI iii](#_Toc15905389)

[MODUL 1 NAMA POKOK BAHASAN 1](#_Toc15905390)

# MODUL 1 OVERVIEW PYTHON

|  |  |
| --- | --- |
|  | CAPAIAN PEMBELAJARAN |
| 1. Mampu mempraktekkan penggunaan python (editor, syntax, struktur, variabel dan struktur data, control statement, function) 2. Mampu mengenal dan menggunakan software yang didukung oleh Python | |

|  |  |
| --- | --- |
|  | KEBUTUHAN ALAT/BAHAN/SOFTWARE |
| 1. Python (Anaconda / Miniconda) 2. Jupyter Notebook | |

|  |  |
| --- | --- |
|  | DASAR TEORI |

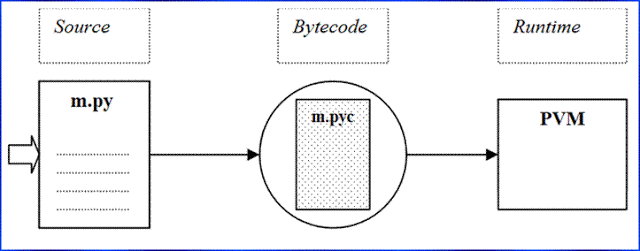
Python adalah Bahasa pemrograman komputer serta peranti penerjemah (interpreter) untuk menjalankan / mengeksekusi *source code* yang dibuat menggunakan Bahasa pemrograman Python. Bahasa pemrograman Python dibuat pertama kali pada tahun 1980-an oleh Guido Van Rossum dan saat ini dikembangkan oleh komunitas di bawah kendali PSF (Python Software Foundation - <https://www.python.org/psf/>). Untuk selanjutnya, istilah Python akan mengacu pada spesifikasi serta software untuk interpreter Python tersebut.

Python merupakan Bahasa pemrograman tingkat tinggi dan memiliki keistimewaan yang membedakannya dengan Bahasa-bahasa pemrograman lain yaitu :

* Mudah dipelajari (*easy to learn*)
* Mudah digunakan (*easy to use*)
* Mudah dalam pengembangan (*easy to develop*)
* Manajemen memori dinamis
* Pemrograman berorientasi objek (*Object Oriented Programming*)
* *Platform independent*
* Bersifat open source dan gratis

Kelebihannya yang bersifat platform independent membuat program yang dibuat menggunakan Python dapat berjalan pada sistem operasi manapun selama di dalamnya terdapat *interpreter* Python.

Secara umum, software Python biasanya bisa diambil dari https://www.python.org meskipun beberapa perusahan maupun komunitas developer juga membuat distribusi Python maupun versi *interpreter* Python untuk platform tertentu. Python dari situs web tersebut dikenal dengan istilah CPython dan merupakan *reference implementation* dari spesifikasi Python. Kode program yang ditulis dalam Python (file .py) dieksekusi menggunakan program Python Interpreter (sama seperti PHP, Perl, dan Ruby). Meskipun demikian, pada saat kode program dieksekusi, secara implisit Python sebenarnya melakukan kompilasi terhadap kode tersebut terlebih dahulu. Hasil kompilasi ini berupa *byte code* (file .pyc). Selanjutnya, yang dieksekusi oleh Python Virtual Machine (PVM) adalah *byte code*, bukan kode programnya. PVM adalah mesin *runtime* untuk Python, yang merupakan bagian dari sistem Python, bukan software terpisah. Secara teknis, sistem Python secara keseluruhan disebut sebagai “Python Interpreter”. Dalam Python, *byte code* disimpan di dalam memori computer dan akan langsung dibuang ketika eksekusi program berakhir. Hal inilah yang menyebabkan proses eksekusi program di dalam Python dapat berjalan dengan cepat. Model eksekusi program dapat digambarkan sebagai berikut.



***Identifier* / Nama**

Saat memprogram menggunakan Python, seorang programmer harus berurusan dengan nama/*identifier*, misalnya nama variabel, nama kelas, dan lain-lain. Berikut adalah ketentuan nama yang sah di Python:

1. Bukan merupakan kata kunci / *keyword* di Python
2. Membedakan huruf besar dan kecil
3. Tidak boleh dimulai dengan digit (0-9)
4. Digit hanya boleh setelah karakter pertama
5. Boleh huruf besar atau kecil
6. Karakter yang diperbolehkan: *underscore* (\_).
7. Tidak boleh menggunakan karakter-karakter yang sudah ada di Python untuk keperluan tertentu (misalnya \* / + -).

Meskipun tidak ada aturan, di kalangan pemrogram Python, biasanya digunakan konvensi berikut ini:

1. Nama modul harus huruf kecil, jika diperlukan bisa menggunakan *underscore*.
2. Nama variabel dan nama fungsi / *method* harus huruf kecil dan menggunakan *underscore* jika terdiri atas 2 kata atau lebih.
3. Nama kelas harus *CamelCase*.
4. Konstanta harus huruf besar semua.

Contoh:

**modul.NamaKelas.nama\_method**

**Komentar**

Baris(-baris) tertentu dalam kode sumber Python biasanya digunakan untuk membuat keterangan atau dokumentasi. Supaya tidak dijalankan, maka harus dimasukkan dalam

komentar:

# komentar satu baris penuh

print(„abc‟) # komentar mulai kolom tertentu

“”” komentar

Lebih dari

satu baris

“””

**Variabel dan Tipe Data Dasar**

Variabel adalah nama yang digunakan untuk menyimpan suatu nilai. Nama ini nantinya akan digunakan dalam proses-proses program selanjutnya. Perintah umumnya adalah:

**Nama\_var = nilai**

Contoh :

var01 = 20

var\_02 = 30

nama\_var = 'Satu dua tiga'

print(var01)

print(var\_02)

print(nama\_var)

# ini salah

var 01 = 21

Bentuk penugasan (pengisian variabel) lainnya:

>>> var1 = var2 = var3 = 4

>>> var1

4

>>> var2

4

>>> var3

4

>>> v1, v2, v3 = 'isi 1', 20, 43

>>> v1

'isi 1'

>>> v2

20

>>> v3

43

>>>

Python adalah bahasa pemrograman yang termasuk dalam kategori *dynamic*

*typing*, artinya tipe data suatu variabel nanti bisa berubah / bersifat dinamis, berbeda

dari apa yang telah dideklarasikan pada awalnya:

>>> var1 = 143

>>> var2 = var1 + 2

>>> var2

145

>>> var1 = 'Python'

>>> var2 = var1 + 2

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: can only concatenate str (not "int") to str

>>>

Variabel juga bisa dihapus:

>>> a = 10

>>> a

10

>>> del a

>>> a

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'a' is not defined

>>>

Ada beberapa tipe data dasar yang bisa disimpan oleh variabel.

**Numerik**

Ada 3 tipe angka: **integer** (bilangan bulat), **float** (bilangan pecahan), serta **complex** (bilangan kompleks).

**String**

String digunakan untuk menyimpan data karakter / huruf / angka yang tidak dimaksudkan untuk operasi matematika.

**Anaconda**

Anaconda adalah platform bahasa pemrograman Python yang bersifat *open-source*. Yang bertujuan untuk menyederhanakan manajemen paket dan penyebaran. Anaconda memiliki package installer yang cukup handal dan memiliki package yang lengkap dan ter*update*. Anaconda sangat direkomendasikan ketika menggunakan Python sebagai bahasa untuk *machine learning*. Fungsi Anaconda yaitu digunakan untuk pemrosesan data skala besar, *predictive analytics*, dan *scientific computing* yang mampu menyederhanakan manajemen *packages* dan *deployment*. Beberapa kelebihan anaconda:

* Mudah dan Cepat dalam menginstall *package* untuk *data science*
* Mudah dalam mengatur *library*, dependensi dan lingkungan dengan Conda
* *Library machine learning* sudah tersedia seperti scikit-learn, TensorFlow, Theano untuk melakukan pengembangan dan pelatihan data
* *Library data analysis* seperti numpy, pandas dan Numba
* *Library* visualisasi yang sudah ada seperti Dash, Matplotlib, Bokeh, Datashader dan lainnya
* *Jupyter Notebook* sebagai lingkungan IDE berbasis web yang mudah digunakan

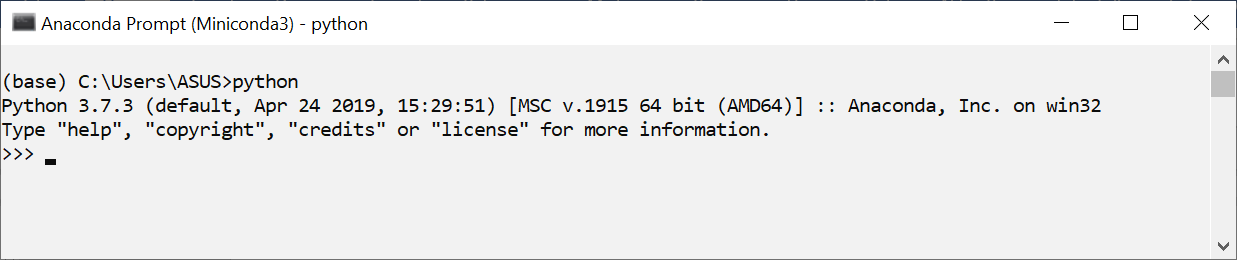
**Jupyter Notebook**

Anaconda memiliki beberapa editor seperti Jupyter Notebook, QT Console, Spyder, VS Code, Glueviz, Orange 3. Salah satu kelebihan Anaconda adalah adanya Jupyter Notebook. Jupyter Notebook merupakan editor dalam bentuk web aplikasi di localhost komputer. Kelebihan utama editor ini yaitu *user friendly* dan mudah digunakan. Editor ini berbasis web dan dijalankan di localhost komputer. Jupyter dapat mendokumentasikan sebuah pekerjaan, dimana *coding* dan dokumentasi bisa dilakukan dalam satu page dan disimpan dalam bentuk presentasi yang menarik.

|  |  |
| --- | --- |
|  | PRAKTIK |

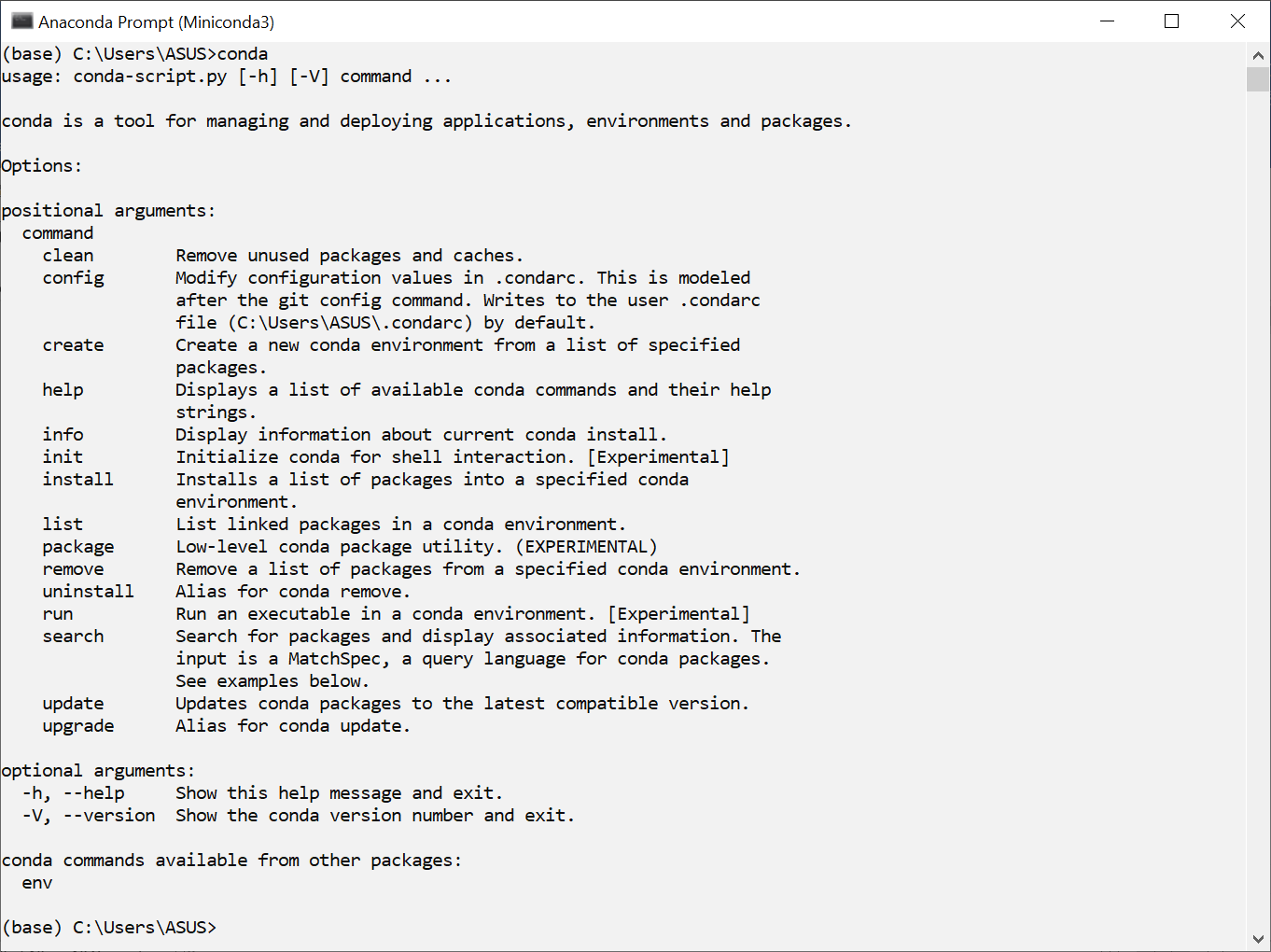
**Cara awal menjalankan python**

1. Langkah awal buka program python yang ada pada start menu komputer/notebook hingga muncul tampilan dibawah ini :



Selain menjalankan interpreter python melalui start menu pada windows, dapat juga dijalankan melalui command prompt atau memanggilnya pada fasilitas run pada windows. Pada interpreter ini kita dapat mengetikkan perintah pada python.

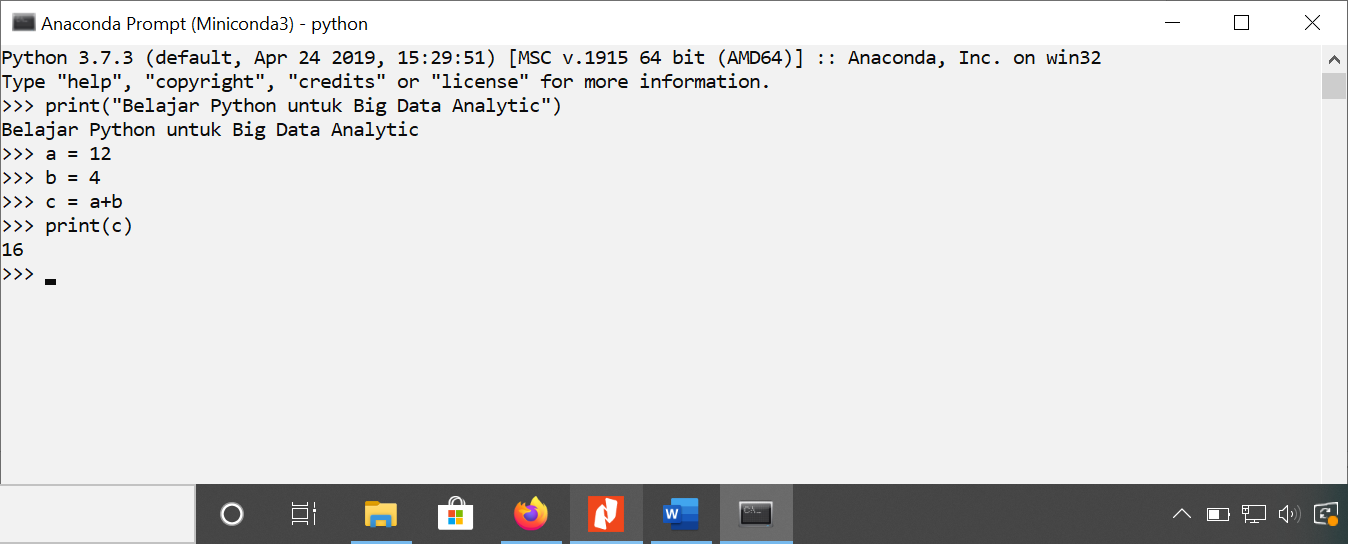
1. Cara menjalankan anaconda dan melihat fungsi apakah conda dapat berjalan dengan baik. Dalam mengupdate conda ketik conda update.



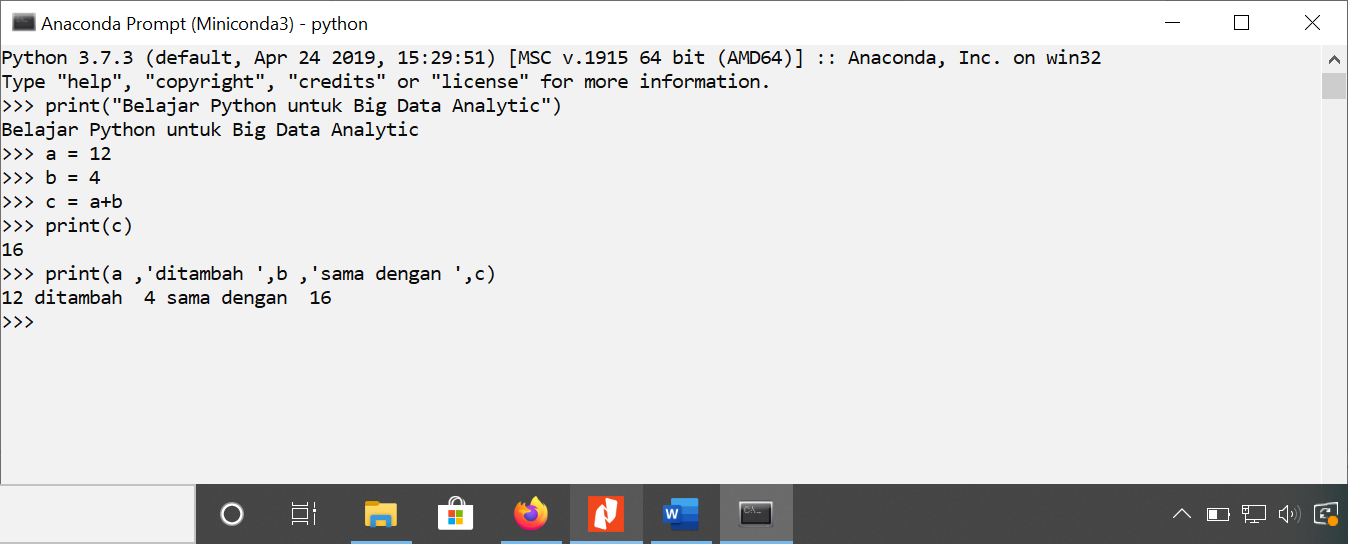
1. Untuk mengetikkan program



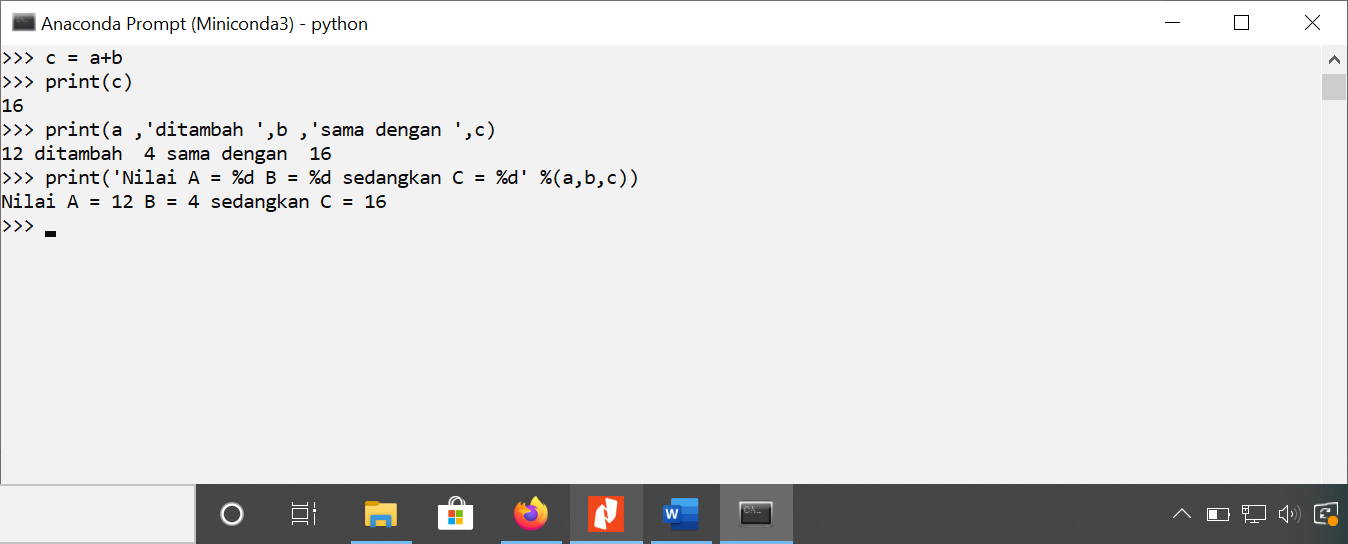
1. Contoh penggunaan Print dan Variabel

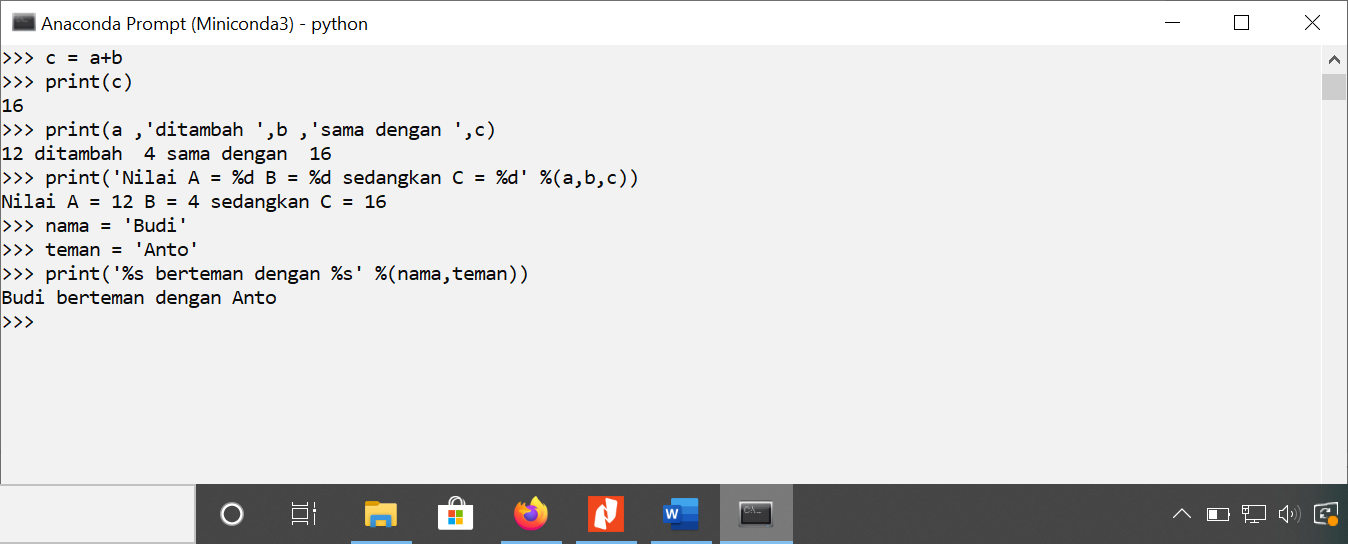


1. Contoh untuk Print penggabungan String dan variable dengan tanda koma



1. Contoh print penggabungan menggunakan symbol %d(integer), %f(float, %.nf n sebagai n angka di belakang koma), %s(string)





1. Operator aritmatika : coba jalankan perintah-perintah berikut

print(21+22)

print(34-14)

print(2\*3)

print(21/2)

print(21.00/2.00)

print(21%2)

print(21//2)

print(21/2)

print(2 > 3)

print(5 < 6)

print(7 != 8)

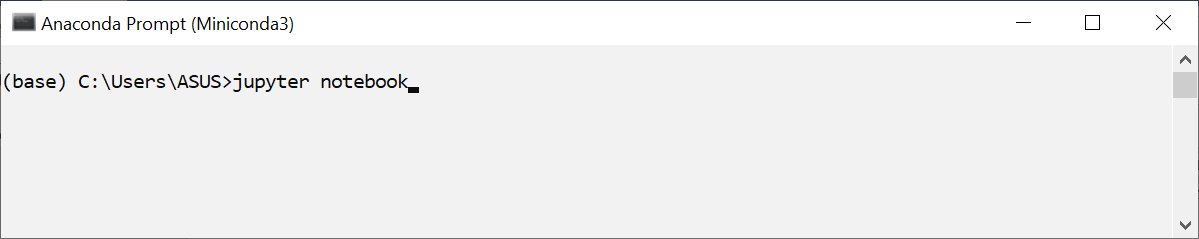
print(7 == 8)

print((2 > 3) & (4 < 7))

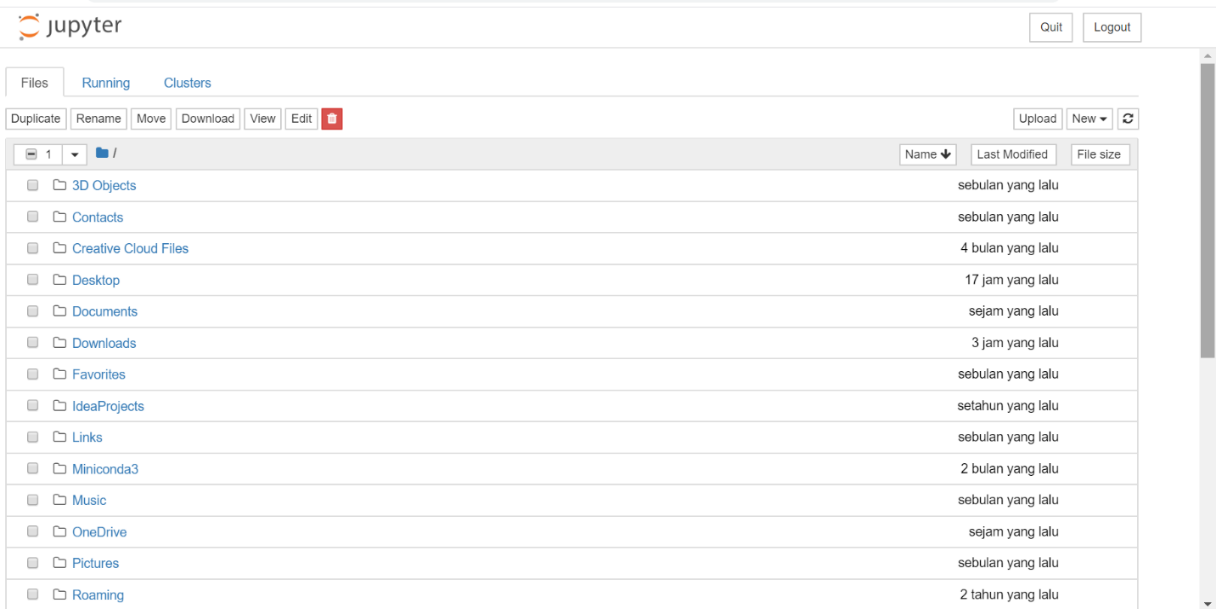
print((2 > 3) | (4 < 7))

ketikkan **exit()** untuk keluar

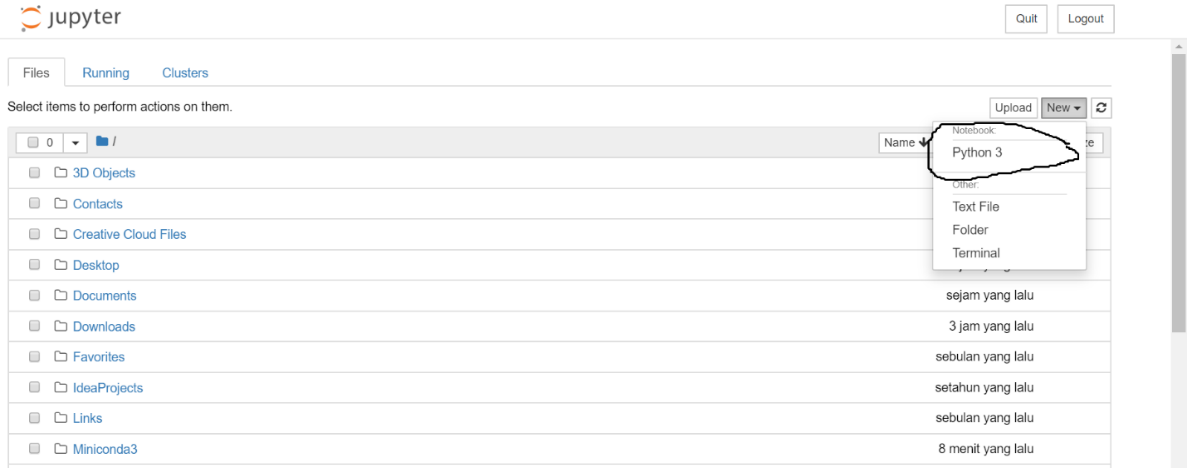
1. Untuk membuka editor Jupyter Notebook



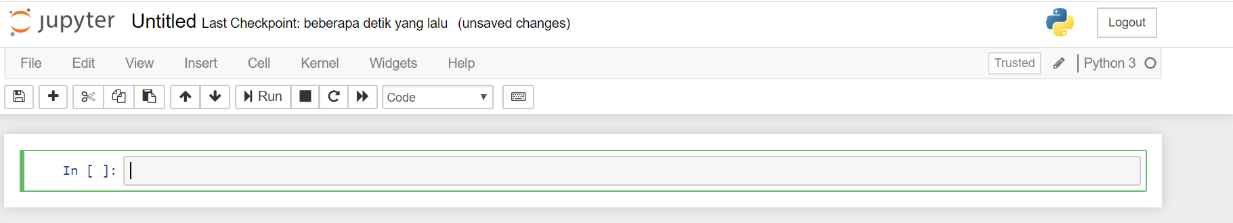
Kemudian akan muncul tampilan sebagai berikut :



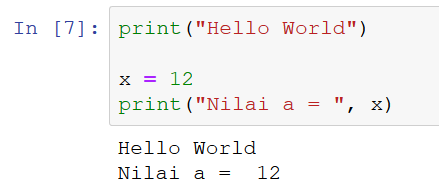
1. Memulai menggunakannya dengan memilih file pada tab ***Files*** atau kita bisa buat baru dengan dropdown ***New*** lalu pilih pilihan dibawah label ***Notebooks*** dan muncul tampilan seperti pada gambar dibawah ini.



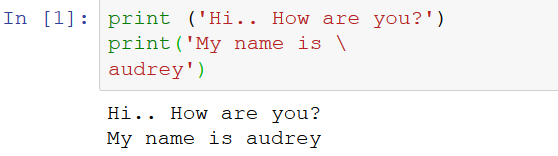
1. Dari gambar diatas akan muncul tampilan untuk melakukan editor program seperti dibawah ini

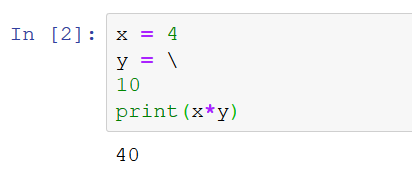


1. Membuat variabel dan melakukan pemanggilan variabel



1. Cobalah mengetikkan perintah-perintah diatas menggunakan Jupyter Notebook kemudian jalankan
2. Contoh penggunaan multiline statements





|  |  |
| --- | --- |
|  | **LATIHAN** |

1. Buatlah program python menggunakan Jupyter Notebook untuk :
2. Menghitung luas persegi Panjang
3. Menghitung luas segitiga

|  |  |
| --- | --- |
| **tugas.png** | **TUGAS** |

|  |  |
| --- | --- |
|  | **REFERENSI** |