

Pengenalan Bahasa Pemrograman Python

1	Bahasa Pemrograman Python	1
1.1	Tujuan Praktikum	
1.2	Alat dan Bahan	
1.3	Materi	
1.3.1	Kenapa Python?	
1.3.2	Menginstall Python 3	
1.3.3	Menjalankan Python Mode Interaktif	
1.3.4	Editor untuk Python	
1.3.5	Menjalankan Script Python di Terminal/Console	
1.3.6	Mencari Bug dan Memperbaikinya (debugging)	
1.4	Kegiatan Praktikum	
1.4.1	Menginstall Package Jupyter Notebook	
1.4.2	Eksplorasi Python mode Interaktif	
1.5	Latihan Mandiri	
2	Variable, Expression dan Statements	19
2.1	Tujuan Praktikum	
2.2	Alat dan Bahan	
2.3	Materi	
2.3.1	Values dan type	
2.3.2	Variabel	
2.3.3	Nama Variabel dan Keywords	
2.3.4	Statements	
2.3.5	Operator dan Operand	
2.3.6	Expressions	
2.3.7	Urutan Operasi	
2.3.8	Operator Modulus dan String	
2.3.9	Menangani Input dari Pengguna	
2.4	Komentar	
2.5	Kegiatan Praktikum	
2.5.1	Membuat Variabel	
2.5.2	Memberikan nilai dalam variabel	
2.5.3	Mencetak nilai dalam variabel	
2.5.4	Separator, tipe data dan fungsi type	
2.6	Latihan Mandiri	

1. Bahasa Pemrograman Python

1.1 Tujuan Praktikum

Setelah mempelajari Bab ini, mahasiswa diharapkan dapat:

1. Dapat menginstall Python pada sistem operasi yang digunakan.
2. Dapat menjalankan perintah-perintah Python dalam mode interaktif.
3. Mengenal beberapa pilihan editor untuk Python dan dapat menggunakan salah satunya untuk menjalankan script Python.
4. Dapat menjalankan script Python di Terminal/Console maupun di Web (dengan Jupyter Notebook).
5. Dapat menginstall maupun mengupdate package.
6. Dapat menemukan kesalahan pada script Python berdasarkan informasi dari interpreter Python.

1.2 Alat dan Bahan

Praktikum ini membutuhkan perangkat komputer yang memiliki spesifikasi minimum sebagai berikut:

1. Terkoneksi ke Internet dan dapat mengunduh package-package Python.
2. Mampu menjalankan sistem operasi Windows 10 atau Ubuntu Linux.

Perangkat lunak yang diperlukan untuk mendukung praktikum ini adalah sebagai berikut:

1. Python 3.7 atau 3.8 yang terinstall menggunakan Anaconda atau Installer Python lainnya.
2. Web Browser (Mozilla Firefox, Microsoft Edge atau Google Chrome).
3. Command Prompt (jika menggunakan Windows).
4. Terminal (jika menggunakan Linux).
5. Editor Python (Visual Studio Code, PyCharm, Spyder atau editor-editor lainnya yang mendukung Python).

1.3 Materi

1.3.1 Kenapa Python?

Python adalah bahasa pemrograman level tinggi yang *interpreted*, mendukung Object Oriented Programming (OOP) dan memiliki sifat *dynamic semantics*. Menurut sebuah survey yang diselenggarakan oleh Stackoverflow (<https://insights.stackoverflow.com/survey/2019#technology>), Python merupakan salah satu bahasa pemrograman yang paling banyak dipakai setelah Javascript. Secara umum Python merupakan salah satu bahasa pemrograman yang paling populer di dunia karena aturan dan sintaksnya yang sederhana untuk dipelajari bagi pemula. Logo dari Python dapat dilihat pada Gambar 1.1.



Gambar 1.1: Logo Python (diambil dari <https://www.python.org/>)

Kesederhanaan bentuk dan sintaks Python dapat dilihat dari perbandingan source code program Hello World jika dibandingkan dengan Java dan C.

Pada bahasa pemrograman Java:

```
1 public class Main {  
2     public static void main(String[] args){  
3         System.out.println("Hello World!");  
4     }  
5 }
```

Pada bahasa pemrograman C:

```
1 #include <stdio.h>  
2  
3 int main(){  
4     printf("Hello World");  
5     return 0;  
6 }
```

Pada bahasa pemrograman Python:

```
1 print("Hello World!");
```

Beberapa kelebihan dari bahasa pemrograman Python antara lain:

- Dukungan pustaka pihak ketiga yang sangat kaya dan beragam. Pustaka-pustaka tersebut menyebabkan Python dapat digunakan untuk menyelesaikan masalah di berbagai macam bidang. Sebagai contoh Python banyak digunakan di bidang data science karena ada banyak sekali pustaka-pustaka pendukung seperti pandas, numpy, tensorflow, keras dan library-library machine learning lainnya.

- Pustaka bawaan dari Python sendiri sudah sangat beragam dan mencakup banyak sekali aspek-aspek dasar yang umumnya dibutuhkan oleh programmer, antara lain dukungan terhadap basis data, pengaksesan jaringan, pengaksesan fitur-fitur di sistem operasi dan masih banyak lainnya.
- Python memiliki lisensi Open Source sehingga dapat digunakan secara bebas bahkan dapat digunakan untuk keperluan komersial tanpa perlu membayar lisensi. Selain itu Python juga aktif dikembangkan sehingga fitur-fitur dan kemampuannya selalu bertambah.
- Python relatif mudah dipelajari untuk pemula karena sintaksnya yang sederhana dan sangat menyerupai bahasa Inggris. Selain itu juga tersedia banyak dokumentasi, tutorial dan bahkan online course tentang Python yang memudahkan seorang pemula untuk mempelajarinya.

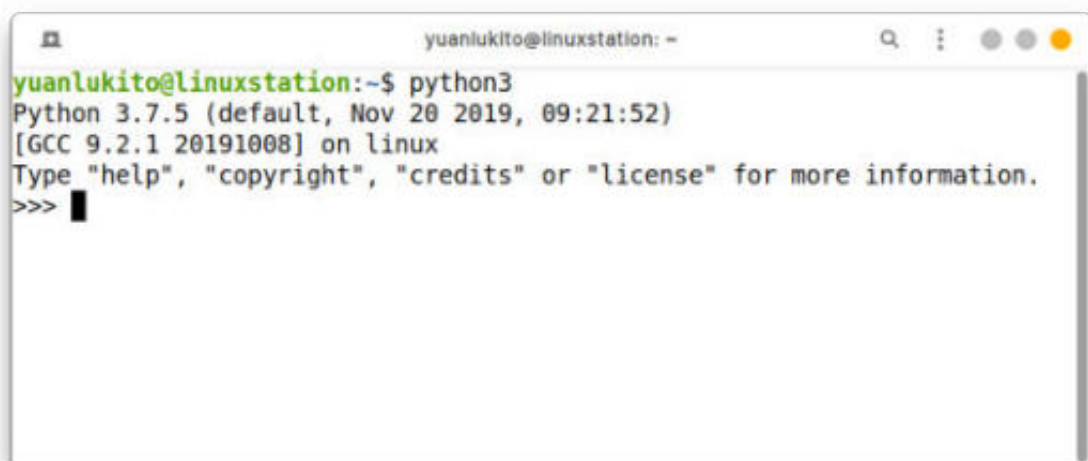
Walaupun memiliki banyak kelebihan, Python juga memiliki beberapa kekurangan sebagai berikut:

- Saat ini Python belum mendukung untuk pembuatan aplikasi di platform mobile seperti Android atau iOS.
- Konsumsi memory yang relatif besar sehingga tidak cocok digunakan untuk kasus-kasus yang membutuhkan memory dalam jumlah sangat besar.
- Kecepatan proses dari Python relatif lebih lambat jika dibandingkan dengan bahasa pemrograman seperti C.

Sebagai pemula, anda tidak perlu terlalu memikirkan tentang kelemahan-kelemahan Python karena kemungkinan besar anda tidak akan menjumpai masalah-masalah tersebut. Dengan bahasa pemrograman Python, diharapkan anda dapat belajar algoritma dan pemrograman dengan lebih cepat dan lebih mudah jika dibandingkan menggunakan bahasa pemrograman lainnya.

1.3.2 Menginstall Python 3

Untuk saat ini terdapat dua macam versi Python yang banyak digunakan, yaitu Python versi 2 dan versi 3. Untuk keperluan mata kuliah ini digunakan Python 3 yang merupakan versi yang lebih baru. Bagi anda yang menggunakan sistem operasi Linux maupun macOS, biasanya Python versi 3 sudah terinstall dan siap digunakan. Pada Ubuntu, Python dapat dijalankan pada terminal menggunakan perintah **python3** seperti yang ditunjukkan pada Gambar 1.2.



```
yuanlukito@linuxstation: ~$ python3
Python 3.7.5 (default, Nov 20 2019, 09:21:52)
[GCC 9.2.1 20191008] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Gambar 1.2: Python versi 3 di Ubuntu Linux. Terinstall Python versi 3.7.5

Bagi anda yang menggunakan Windows, anda bisa menginstall distribusi Anaconda yang relatif mudah untuk diinstall. Pilihlah Python versi 3.7 dan disarankan memilih opsi 64-Bit Graphical

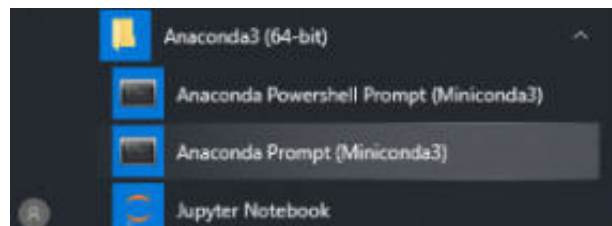
Installer, seperti pada Gambar 1.3.



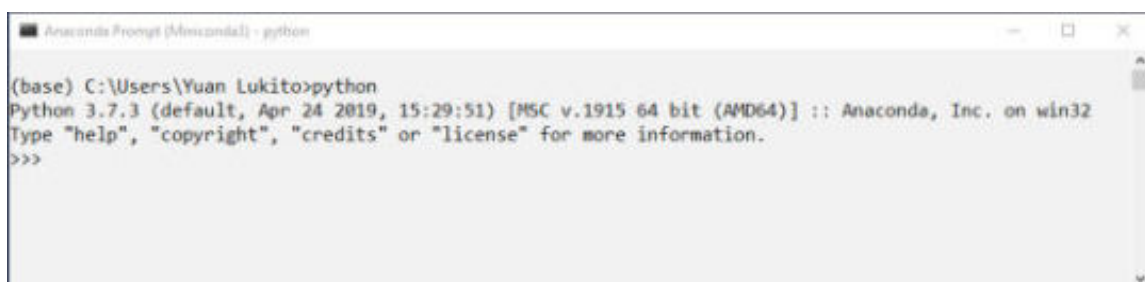
Gambar 1.3: Distribusi Anaconda untuk Windows.

1.3.3 Menjalankan Python Mode Interaktif

Untuk masuk mode interaktif, jalankan perintah **python3** pada terminal di Ubuntu. Jika anda menggunakan Windows, jalankan Anaconda Prompt, kemudian ketikkan perintah **python** seperti yang ditunjukkan pada Gambar 1.4 dan Gambar 1.5.



Gambar 1.4: Menu Anaconda Prompt setelah terinstall di Windows.



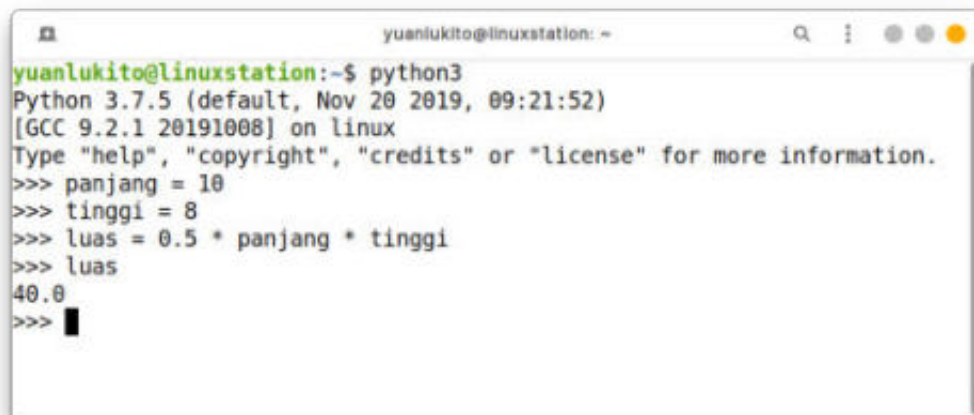
Gambar 1.5: Python 3.7 pada Anaconda prompt di Windows.

Penggunaan mode interaktif ini memungkinkan anda memasukkan perintah satu-persatu dan langsung diproses oleh interpreter Python. Sebagai contoh, hitunglah luas segitiga yang alasnya memiliki panjang 10 cm dan tinggi 8 cm. Untuk menghitung luas segitiga tersebut, ketikkan perintah-perintah berikut ini secara berurutan (diakhiri dengan menekan tombol Enter):

- `panjang = 10`
- `tinggi = 8`
- `luas = 0.5 * panjang * tinggi`

- luas

Hasilnya akan seperti yang ditunjukkan pada Gambar 1.6.



```

yuanlukito@linuxstation: ~
yuanlukito@linuxstation:~$ python3
Python 3.7.5 (default, Nov 20 2019, 09:21:52)
[GCC 9.2.1 20191008] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> panjang = 10
>>> tinggi = 8
>>> luas = 0.5 * panjang * tinggi
>>> luas
40.0
>>> █

```

Gambar 1.6: Menghitung Luas Segitiga dengan bantuan Python

Perintah pertama dan kedua adalah perintah *assignment*, yaitu mengisi nilai variabel panjang dengan nilai 10 dan variabel tinggi diisi dengan nilai 8. Sedangkan perintah ketiga mengisi variabel luas dengan hasil perhitungan $0.5 * panjang * tinggi$. Perintah terakhir adalah menampilkan isi dari variabel luas.

Variabel adalah tempat kita menyimpan nilai yang nantinya akan dipakai untuk proses-proses berikutnya. Pembahasan variabel akan dilakukan pada Bab 2.

Untuk mengakhiri sesi interaktif, ketikkan perintah **exit()** yang diakhiri dengan tombol Enter.

1.3.4 Editor untuk Python

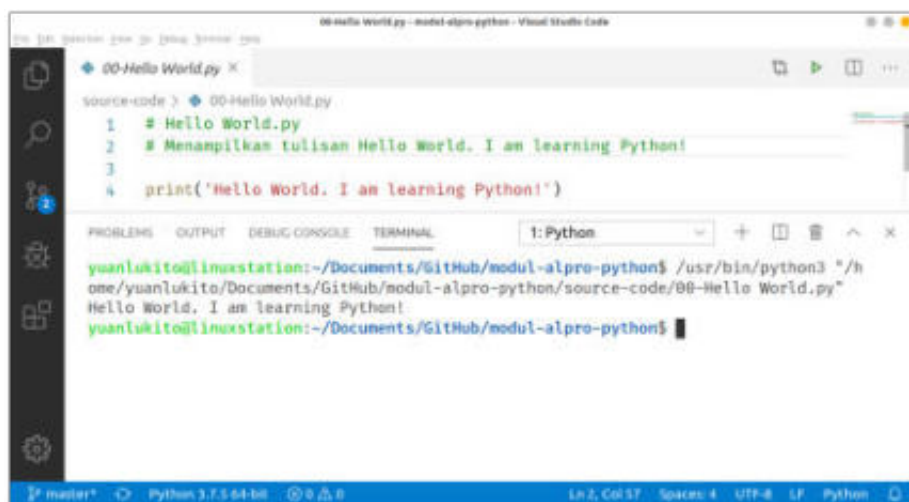
Untuk menulis suatu program, anda memerlukan editor untuk menuliskan source code yang diinginkan. Semakin canggih suatu editor, biasanya akan sangat mempermudah dalam pembuatan program. Beberapa editor yang biasanya dipakai untuk membuat program dalam bahasa pemrograman Python antara lain:

- Visual Studio Code + Python Extension for Visual Studio Code (<https://code.visualstudio.com/>).
- PyCharm (<https://www.jetbrains.com/pycharm/>).
- Spyder (<https://www.spyder-ide.org/>).
- ActivePython (<https://www.activestate.com/products/activepython/>).
- IDLE (<https://docs.python.org/3/library/idle.html>).

Contoh tampilan Visual Studio Code yang sudah terpasang Python Extension dapat dilihat pada Gambar 1.7 dan Gambar 1.8. Editor lain seperti PyCharm punya fitur yang lebih lengkap dan dapat digunakan untuk berbagai macam bentuk project Python. Tampilan PyCharm dapat dilihat pada Gambar 1.9. Modul ini menggunakan Visual Studio Code sebagai editor utama dan PyCharm sebagai editor penunjang. Anda lebih baik menginstall keduanya sekaligus untuk mempelajari kelebihan dan kekurangannya masing-masing. Khusus untuk PyCharm, pilihlah versi Community Edition yang berlisensi gratis.



Gambar 1.7: Python Extension untuk Visual Studio Code.



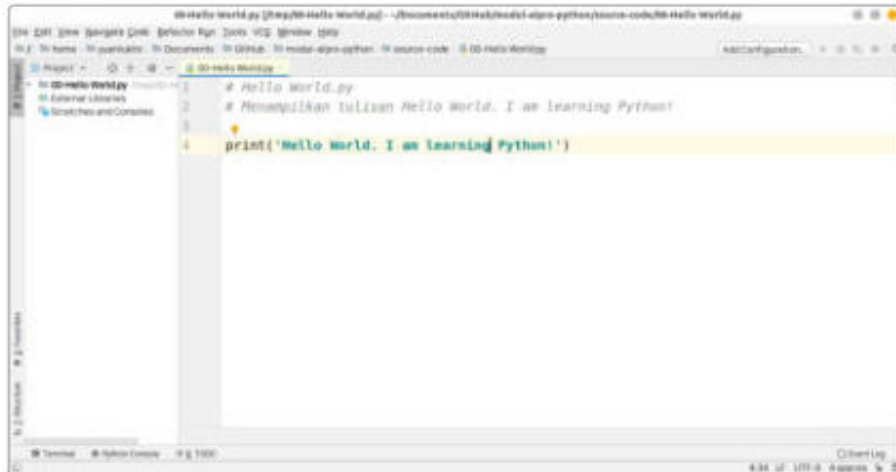
Gambar 1.8: Tampilan Visual Studio Code saat menjalankan script Python.

1.3.5 Menjalankan Script Python di Terminal/Console

Anda sebelumnya sudah mencoba menjalankan Python dalam mode interaktif (seperti pada Gambar 1.6). Pada mode interaktif anda harus mengetik perintah-perintah yang diinginkan satu-persatu. Setiap perintah yang anda masukkan langsung dijalankan oleh interpreter Python. Mode interaktif ini cocok digunakan untuk mencoba-coba fungsi atau perintah-perintah baru karena hasilnya akan langsung didapatkan. Kekurangan dari mode interaktif adalah anda harus mengetik perintahnya satu-persatu dan harus mengulanginya lagi jika anda sudah terlanjur keluar dari mode interaktif.

Interpreter Python juga mendukung mode script, yaitu menjalankan daftar perintah-perintah (yang disimpan dalam file .py) secara langsung tanpa anda harus mengetikkannya lagi satu-persatu. Untuk menjalankan script Python menggunakan Terminal/Command Prompt, bukalah Terminal (pada Ubuntu) atau Anaconda Prompt (pada Windows). Ketikkan perintah **python3 namafile.py** (Ubuntu) atau **python namafile.py** pada Windows.

Untuk mencoba, buatlah sebuah file baru bernama **nilaitukardollar.py**, ketiklah kode program berikut di dalam file tersebut, kemudian simpan di suatu tempat.



Gambar 1.9: Tampilan PyCharm saat mengedit file Python.

```

1  # nilai kurs 1 US$ ke IDR
2  kursud = 13950
3
4  # informasi program
5  print('Program konversi US$ ke IDR')
6  print('Kurs saat ini 1 US$ = ',kursud, 'Rupiah')
7  # input jumlah US$ yang mau ditukar
8  jumlahusd = float(input('Masukkan jumlah uang yang mau ditukar ke Rupiah: '))
9  # hitung nilainya dalam Rupiah
10 dalamrupiah = jumlahusd * kursud
11 # tampilkan hasilnya
12 print('Hasil konversi = Rp. ', dalamrupiah)

```

Untuk menjalankan script tersebut, gunakan perintah **python3 nilaitukardollar.py** (Ubuntu) atau **python nilaitukardollar.py** (Windows). Pada Ubuntu, tampilan hasilnya dapat dilihat pada Gambar 1.10.

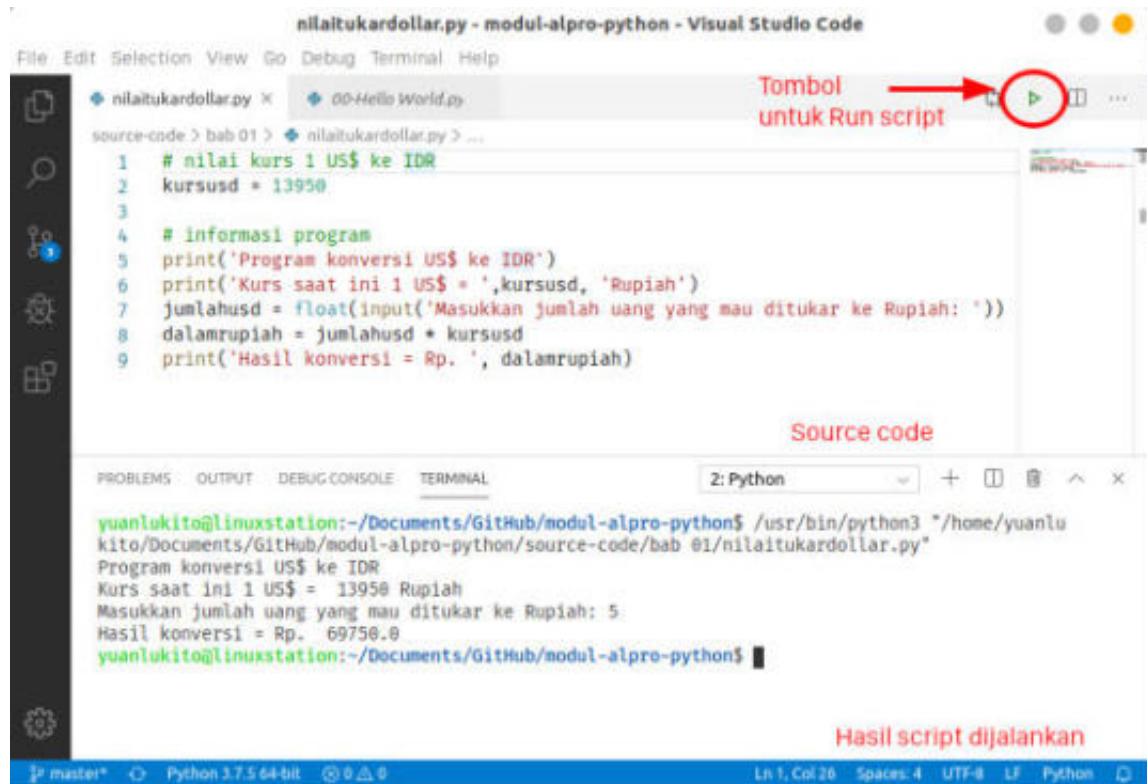


Gambar 1.10: Menjalankan script Python di Terminal Ubuntu.

- # Baris yang diawali dengan tanda # adalah baris komentar. Komentar digunakan untuk memberikan informasi atau keterangan mengenai baris-baris program dan akan diabaikan oleh interpreter Python.

Jika anda menggunakan editor seperti Visual Studio Code, anda tidak perlu repot menjalankan

terminal dan mengetik perintahnya. Pada Visual Studio Code sudah ada tombol Run untuk menjalankan script tersebut. Anda juga dapat berinteraksi dengan hasil program melalui terminal yang disediakan. Tampilan Visual Studio Code saat menjalankan script `nilaitukardollar.py` dapat dilihat pada Gambar 1.11.



Gambar 1.11: Menjalankan script Python pada Visual Studio Code.

1.3.6 Mencari Bug dan Memperbaikinya (debugging)

Bug merupakan istilah dalam pemrograman yang berarti ada kesalahan dalam program. Kesalahan dalam membuat program secara umum dapat digolongkan menjadi dua macam, yaitu:

- Kesalahan dalam menulis program. Biasa disebut sebagai *syntax error* atau *compile error*. Kesalahan jenis ini biasanya diakibatkan oleh kesalahan dalam mengetik (typo). Jenis kesalahan ini mudah sekali untuk ditemukan dengan bantuan interpreter Python maupun editor yang dipakai.
- Kesalahan saat program berjalan. Biasa disebut sebagai *runtime error*. Jenis kesalahan ini lebih sulit ditemukan dan diperbaiki dibanding jenis kesalahan yang pertama.

Python merupakan bahasa pemrograman yang menggunakan sistem interpreter, artinya kode program anda akan dibaca baris-perbaris untuk dicek apakah ada kesalahan dalam penulisan perintah. Jika baris yang sedang dicek memenuhi aturan penulisan program (sintaks), maka baris tersebut langsung dijalankan. Jika terjadi kesalahan pada baris tersebut saat dijalankan, program Python anda akan langsung berhenti dan tidak melanjutkan ke baris berikutnya. Sebagai contoh, perhatikan kode program berikut ini:

```
1 #jumlah belanja (dalam rupiah)
2 belanja = 100000
3
```

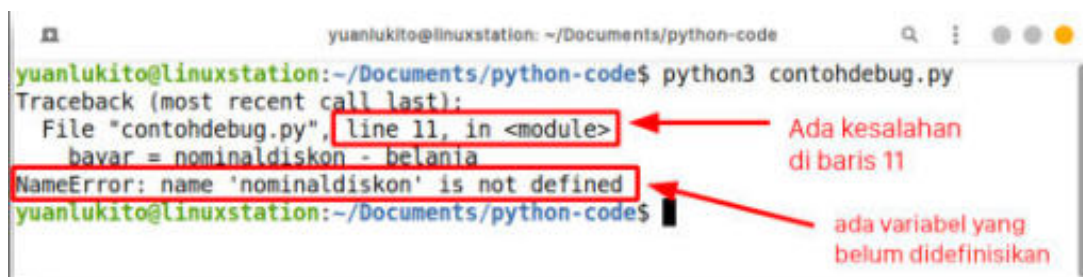
```

4  #besarnya diskon (dalam persen)
5  diskon = 30
6
7  #besarnya diskon
8  nominal_diskon = (diskon/100) * belanja
9
10 #hitung jumlah yang harus dibayar
11 bayar = nominaldiskon - belanja
12
13 #tampilkan hasilnya
14 print('Anda harus membayar: Rp. ', bayar)

```

Program tersebut menghitung jumlah yang harus dibayarkan jika belanja sebesar Rp. 100.000 dan mendapat diskon 30%. Jika script tersebut dijalankan, hasilnya akan seperti pada Gambar 1.12. Saat script dijalankan, tidak ada hasil yang dikeluarkan karena terjadi kesalahan pada baris ke-11. Ada keterangan jenis kesalahan, yaitu **NameError**. Kesalahan ini merupakan kesalahan sintaks karena ada aturan penulisan yang tidak terpenuhi. Kesalahan terjadi pada variabel bernama **nominaldiskon** yang belum didefinisikan tetapi nilainya sudah digunakan.

Jika anda perhatikan kode programnya, pada baris **ke-8** ada variabel bernama **nominal_diskon**, tetapi pada baris **ke-11** dituliskan sebagai **nominaldiskon**. Terjadi kesalahan pengetikan (typo) yang menyebabkan script tidak bisa dijalankan.



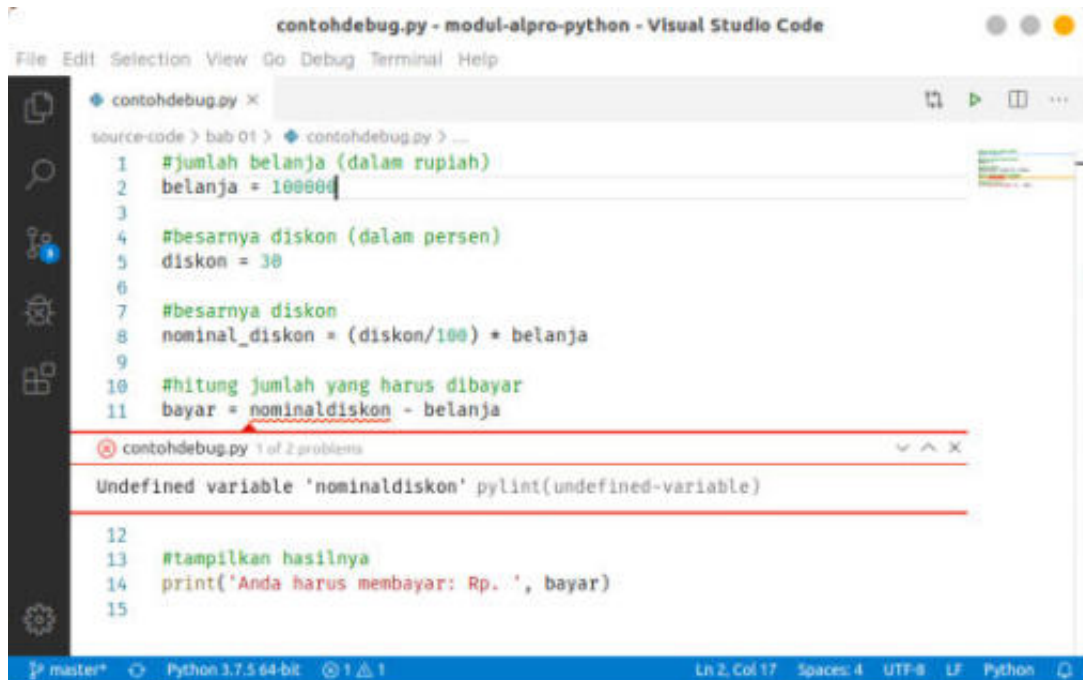
Gambar 1.12: Kesalahan yang muncul saat script dijalankan.

Jika anda menggunakan editor yang mendukung Python, biasanya sebelum dijalankan sudah akan dilakukan pengecekan dan apabila ada kesalahan sintaks dapat diketahui lebih awal. Gambar 1.13 adalah contoh tampilan di Visual Studio Code untuk script tersebut. Untuk memperbaiki kesalahan tersebut, ganti baris ke-8 menjadi **nominaldiskon**, seperti pada kode program berikut ini:

```

1  #jumlah belanja (dalam rupiah)
2  belanja = 100000
3
4  #besarnya diskon (dalam persen)
5  diskon = 30
6
7  #besarnya diskon
8  nominaldiskon = (diskon/100) * belanja
9
10 #hitung jumlah yang harus dibayar
11 bayar = nominaldiskon - belanja

```



Gambar 1.13: Visual Studio Code menemukan kesalahan sintaks sebelum script dijalankan.

```

12
13 #tampilkan hasilnya
14 print('Anda harus membayar: Rp. ', bayar)

```

Setelah diperbaiki, maka sekarang script dapat dijalankan dengan baik sampai selesai seperti yang terlihat pada Gambar 1.14.



Gambar 1.14: Script sudah bisa dijalankan sampai selesai dengan baik.

Jika anda perhatikan dengan baik, script tersebut berjalan dijalankan dengan baik sampai selesai, tetapi hasilnya tidak sesuai dengan yang diharapkan. Jika kita membeli barang Rp. 100.000 dan mendapat diskon 30%, maka seharusnya yang harus dibayarkan adalah Rp. 70.000. Tetapi script tersebut menghasilkan output -70000. Kesalahan ini adalah kesalahan jenis **runtime error**, yaitu kesalahan yang muncul saat program dijalankan. Kesalahan jenis ini biasanya diakibatkan oleh kesalahan algoritma, kesalahan logika atau kesalahan-kesalahan dalam mengatur langkah-langkah dalam program.

Kesalahan terjadi pada baris ke-11, yaitu pada bagian perhitungan jumlah yang harus dibayar, seharusnya jumlah yang harus dibayar adalah harga barang dikurangi diskon. Pada script tersebut perhitungannya terbalik sehingga menjadi salah. Perbaiki script tersebut menjadi seperti berikut ini:

```

1  #jumlah belanja (dalam rupiah)
2  belanja = 100000
3
4  #besarnya diskon (dalam persen)
5  diskon = 30
6
7  #besarnya diskon
8  nominaldiskon = (diskon/100) * belanja
9
10 #hitung jumlah yang harus dibayar
11 # INI SALAH: bayar = nominaldiskon - belanja
12 bayar = belanja - nominaldiskon
13
14 #tampilkan hasilnya
15 print('Anda harus membayar: Rp. ', bayar)

```

Setelah diperbaiki, script tersebut akan menghasilkan hasil yang benar sesuai dengan yang diharapkan. Hasilnya dapat dilihat pada Gambar 1.15.

```

yuanlukito@linuxstation: ~/Documents/python-code
yuanlukito@linuxstation:~/Documents/python-code$ python3 contohdebug.py
Anda harus membayar: Rp. 70000.0
yuanlukito@linuxstation:~/Documents/python-code$

```

Script sudah berjalan dengan baik dan hasilnya juga sesuai dengan harapan

Gambar 1.15: Hasil sudah sesuai dengan yang diharapkan.

1.4 Kegiatan Praktikum

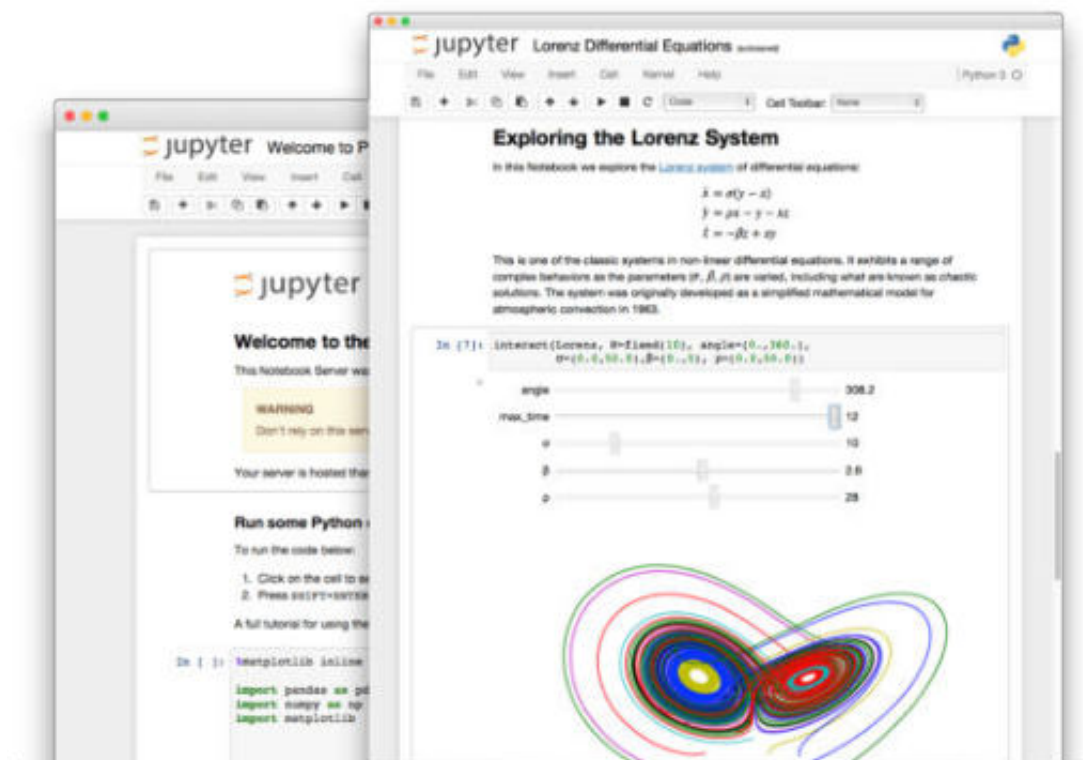
Kegiatan praktikum yang akan dilakukan adalah sebagai berikut:

1. Menginstall Package Jupyter Notebook dan menjalankan script menggunakan Jupyter Notebook.
2. Eksplorasi Python mode interaktif untuk perhitungan-perhitungan sederhana.

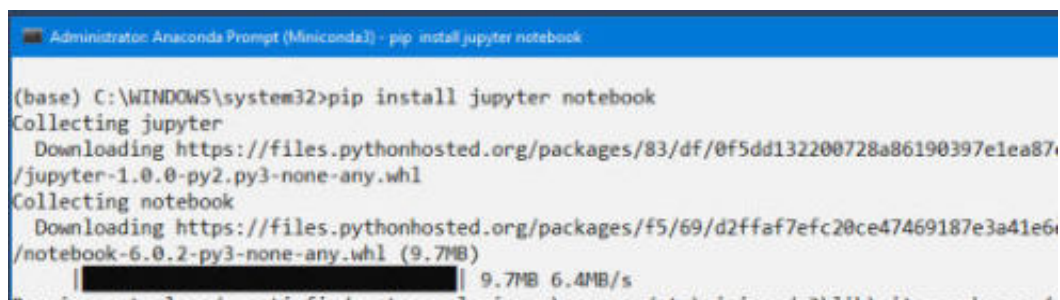
1.4.1 Menginstall Package Jupyter Notebook

Untuk menginstall package Python kita akan menggunakan perintah **pip**. Pip merupakan singkatan dari **package installer for Python**. Pada bagian ini dibahas mengenai instalasi package Jupyter Notebook. Jupyter Notebook adalah aplikasi web yang dapat bertindak sebagai editor Python, menampilkan visualisasi data maupun menampilkan hasil dari menjalankan script Python. Jupyter Notebook banyak dipakai karena hanya dibutuhkan browser untuk mengaksesnya. Tampilan Jupyter Notebook dapat dilihat pada Gambar 1.16.

Untuk menginstall Jupyter Notebook, jalankan Anaconda Prompt (atau Terminal bila anda menggunakan Linux). Kemudian ketik perintah **pip install jupyter notebook**. Pada Ubuntu, perintah yang digunakan adalah **pip3 install jupyter notebook**. Jika anda mengetikkan perintah tersebut dengan benar, pip akan mendownload package-package yang diperlukan dan sekaligus menginstallnya. Tampilan pip saat dijalankan dapat dilihat pada Gambar 1.17.



Gambar 1.16: Contoh tampilan Jupyter Notebook.



Gambar 1.17: Instalasi Jupyter Notebook menggunakan pip.

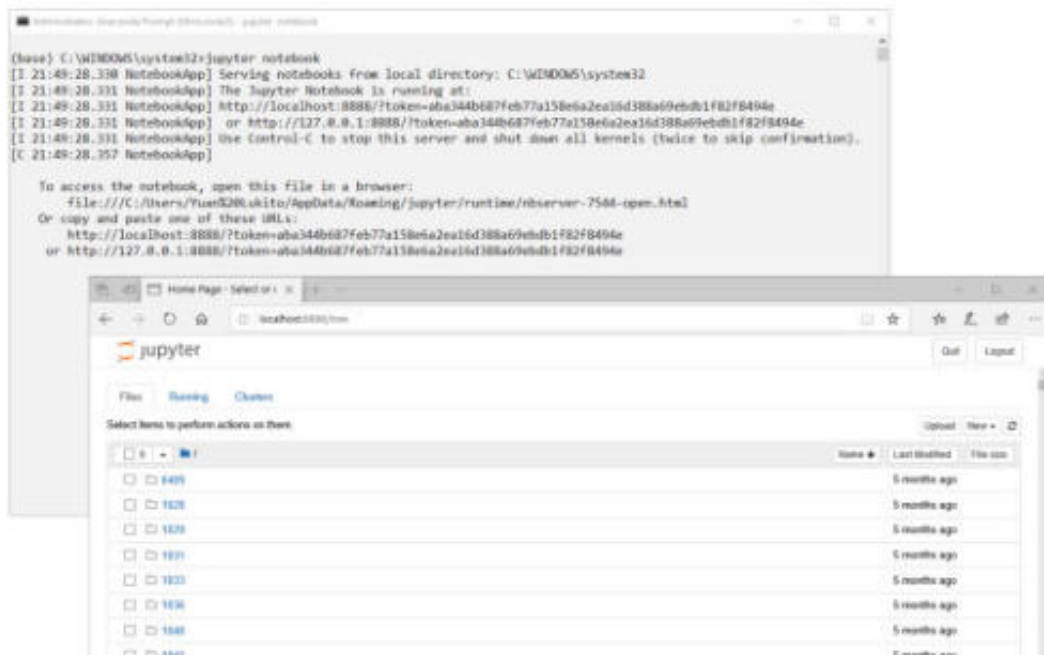
Untuk instalasi package gunakan perintah **pip install <nama package>**

Setelah proses instalasi berhasil, untuk menjalankannya ketikkan perintah **jupyter notebook**. Biasanya akan terbuka browser yang mengarah langsung ke halaman pertama Jupyter Notebook, seperti yang terlihat pada Gambar 1.18.

Untuk menjalankan Jupyter Notebook, gunakan perintah **jupyter notebook** pada Anaconda Prompt atau Terminal.

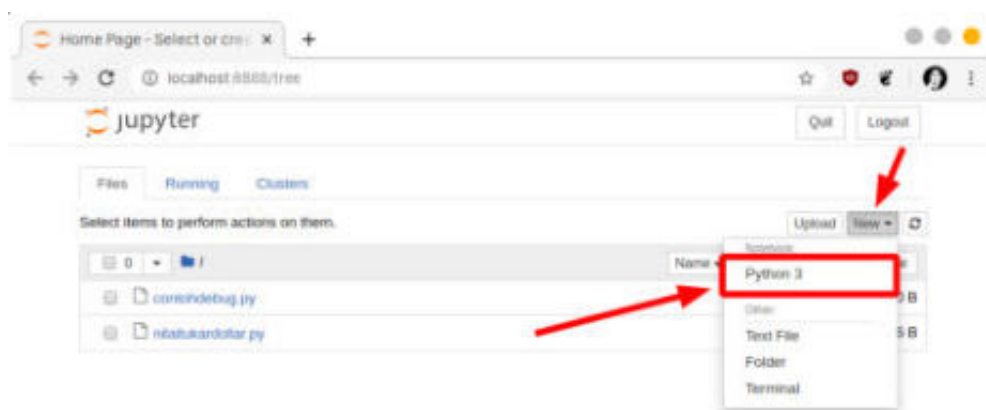
Untuk mengakhiri penggunaan Jupyter Notebook, tekan tombol **Ctrl-C** pada Anaconda Prompt atau Terminal.

Script Python disimpan dalam sebuah file berekstensi *.pynb (Python Notebook). Untuk membuat sebuah notebook baru, pilih menu New > Notebook Python 3. Selain Python Notebook, Jupyter Notebook juga menyediakan fasilitas jika anda ingin membuat file teks atau mengorganisasi



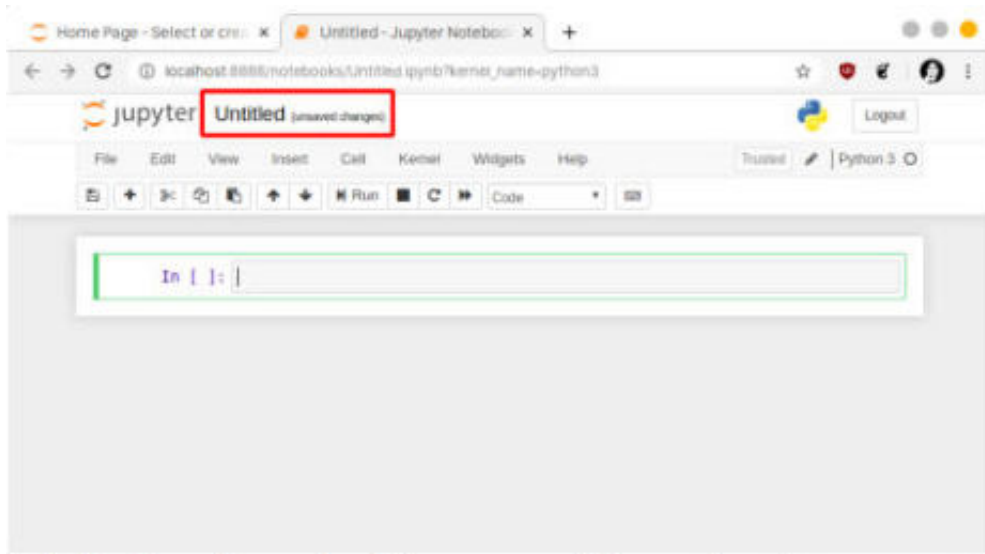
Gambar 1.18: Tampilan awal Jupyter Notebook.

notebook-notebook dalam suatu folder. Menu untuk membuat notebook baru dapat dilihat pada Gambar 1.19.



Gambar 1.19: Menu untuk membuat notebook baru.

Sebuah notebook baru memiliki nama Untitled. Anda dapat mengubahnya sesuai dengan keinginan anda. Kemudian anda diberi 1 Cell untuk menuliskan script Python anda. Setiap Cell dapat dijalankan secara terpisah, maupun juga bisa dijalankan seluruh Cell yang ada sekaligus. Tampilan awal sebuah notebook baru dapat dilihat pada Gambar



Gambar 1.20: Tampilan awal notebook baru di Jupyter Notebook.

1.4.2 Eksplorasi Python mode Interaktif

Pada bagian ini kita akan mencoba menggunakan Python metode interaktif untuk menjawab beberapa pertanyaan-pertanyaan berikut ini:

■ Contoh 1.1 Menghitung Usia

Jika diketahui tahun lahir seseorang, kita bisa menentukan usianya sekarang. Misalnya Andi lahir tahun 1990 dan sekarang adalah tahun 2020, maka usianya adalah 30 tahun. Gunakan Python untuk menjawab beberapa masalah berikut ini (diasumsikan sekarang adalah tahun 2020):

- Trump dilahirkan tahun 1965. Berapa usianya sekarang?
- Pada tahun berapa Trump akan berusia 80 tahun?

Untuk menjawab pertanyaan-pertanyaan tersebut, langkah-langkahnya adalah sebagai berikut:

1. Tahun lahir = 1965.
2. Usia sekarang (2020) = $2020 - 1965 = 55$ tahun.
3. Usia Trump 80 saat = $1965 + 80 = 2045$.

Langkah-langkah penyelesaiannya dalam Python dapat dilihat pada Gambar 1.21.

```
yuanlukito@linuxstation: ~
yuanlukito@linuxstation:~$ python3
Python 3.7.5 (default, Nov 20 2019, 09:21:52)
[GCC 9.2.1 20191008] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> tahun_lahir = 1965
>>> usia_sekarang = 2020 - 1965
>>> usia_sekarang
55
>>> usia_80 = 1965 + 80
>>> usia_80
2045
>>> █
```

Gambar 1.21: Penyelesaian untuk masalah usia Trump.

■ Contoh 1.2 Compound Interest

Compound interest dalam bahasa Indonesia adalah "Bunga ber-bunga". Istilah ini biasanya digunakan dalam dunia keuangan, misalnya perbankan, asuransi maupun investasi. Suatu Bank bernama **Bank Bank Toet** menyediakan produk deposito bernama **Pasti Cuan** dengan bunga 10% per-tahun. Artinya jika anda menyimpan uang anda di deposito tersebut, uang anda akan bertambah sebanyak 10% setiap tahunnya. Gunakan Python untuk mencari jawaban dari contoh masalah berikut ini:

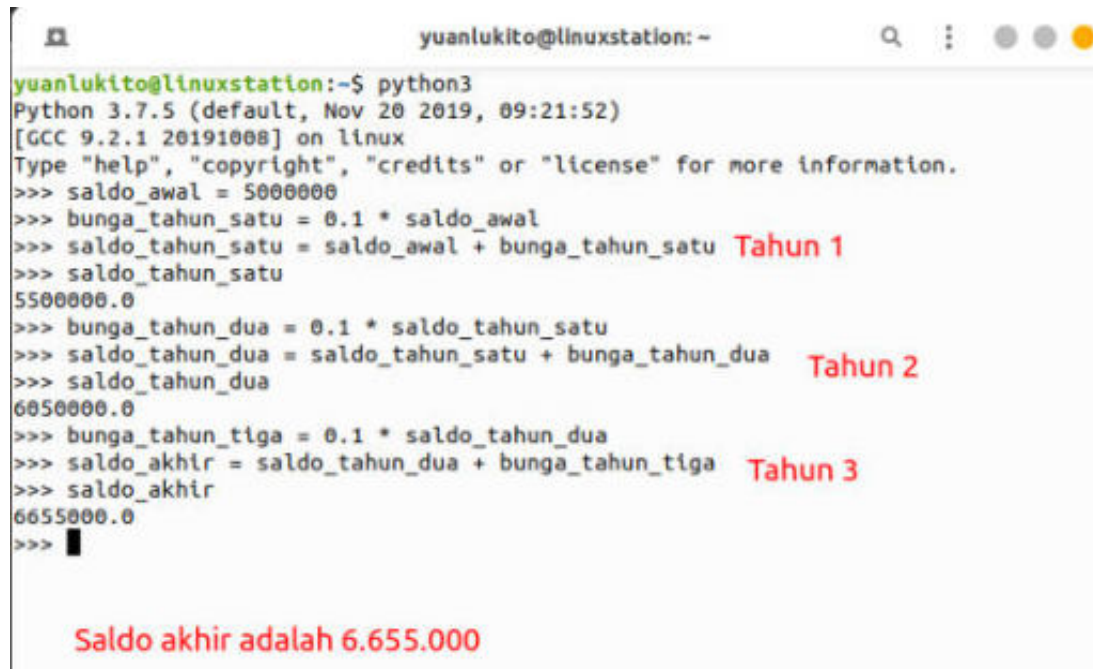
- Jika setoran awal sebanyak Rp. 5 juta rupiah dan disimpan selama 3 tahun tanpa pernah diambil, berapa jumlah uang di akhir tahun ketiga?

Untuk menyelesaikan masalah yang pertama, langkah-langkah yang perlu dilakukan adalah sebagai berikut:

1. Saldo awal Rp. 5 juta.
2. Di akhir tahun pertama, bunga yang didapat adalah $10\% * 5.000.000 = 500.000$.
3. Sehingga saldo di akhir tahun pertama adalah $5.000.000 + 500.000 = 5.500.000$.
4. Di akhir tahun kedua, bunga yang didapat adalah $10\% * 5.500.000 = 550.000$. Ini yang dinamakan sebagai compound interest.
5. Sehingga saldo di akhir tahun kedua adalah $5.500.000 + 550.000 = 6.050.000$.
6. Di akhir tahun ketiga, bunga yang didapat adalah $10\% * 6.050.000 = 605.000$.
7. Sehingga saldo di akhir tahun ketiga adalah $6.050.000 + 605.000 = 6.655.000$.

Setelah anda memahami permasalahan dan dapat menyusun langkah-langkah penyelesaiannya, barulah anda dapat mulai menuliskan programnya menggunakan Python.

Jalankan Python mode interaktif, kemudian masukkan perintah-perintah seperti pada Gambar 1.22.



```
yuanlukito@linuxstation: ~  
yuanlukito@linuxstation:~$ python3  
Python 3.7.5 (default, Nov 20 2019, 09:21:52)  
[GCC 9.2.1 20191008] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> saldo_awal = 5000000  
>>> bunga_tahun_satu = 0.1 * saldo_awal  
>>> saldo_tahun_satu = saldo_awal + bunga_tahun_satu Tahun 1  
>>> saldo_tahun_satu  
5500000.0  
>>> bunga_tahun_dua = 0.1 * saldo_tahun_satu  
>>> saldo_tahun_dua = saldo_tahun_satu + bunga_tahun_dua Tahun 2  
>>> saldo_tahun_dua  
6050000.0  
>>> bunga_tahun_tiga = 0.1 * saldo_tahun_dua  
>>> saldo_akhir = saldo_tahun_dua + bunga_tahun_tiga Tahun 3  
>>> saldo_akhir  
6655000.0  
>>> █  
  
Saldo akhir adalah 6.655.000
```

Gambar 1.22: Hasil perhitungan compound interest selama tiga tahun.

■ Contoh 1.3 Formula Compound Interest

Perhitungan compound interest pada contoh sebelumnya menghitung saldo di akhir tahun ketiga. Bagaimana jika dibutuhkan menghitung berapa saldo di akhir tahun ke-dua puluh? Tentunya

akan sangat melelahkan memasukkan perintah-perintah tersebut secara berulang-ulang untuk menghitung simpanan selama dua puluh tahun. Cara lain yang dapat kita gunakan adalah menggunakan formula compound interest seperti berikut ini:

$$A = P\left(1 + \frac{r}{n}\right)^{nt} \quad (1.1)$$

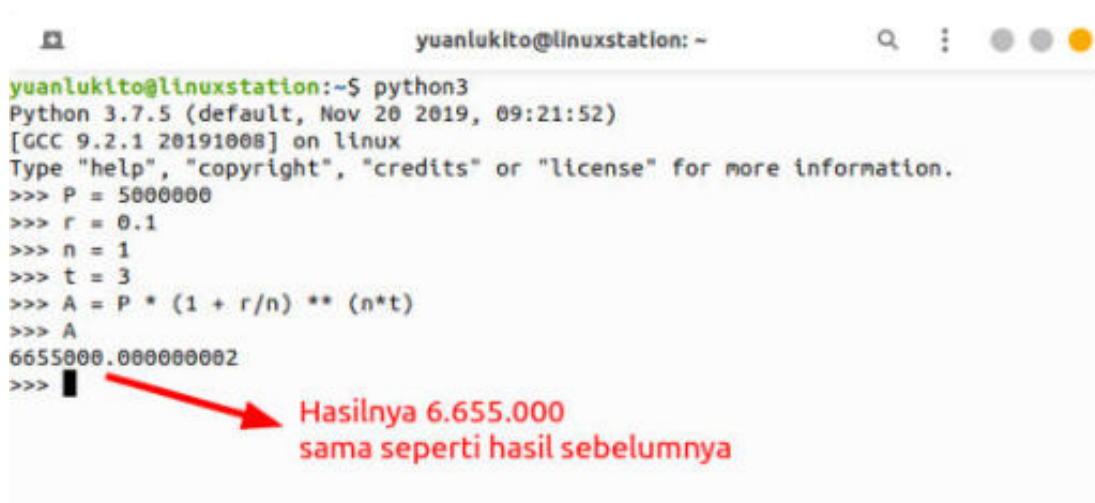
- A = saldo akhir.
- P = saldo awal.
- r = besarnya bunga.
- n = jumlah perhitungan bunga yang dilakukan (dalam 1 tahun berapa kali).
- t = jumlah periode.

Tuliskan formula dan perhitungannya menggunakan Python! ■

Dengan adanya rumus tersebut, langkah-langkah yang diperlukan akan lebih singkat, yaitu:

1. Tentukan saldo awal ($P = 5.000.000$).
2. Tentukan besarnya bunga ($r = 10\% = 0.1$).
3. Tentukan berapa kali bunga didapatkan ($n = 1$, karena bunga dihitung satu kali per-tahun).
4. Tentukan periodenya ($t = 3$ tahun).
5. Masukkan dalam formula compound interest, kita dapatkan saldo akhirnya.

Secara umum dalam membuat program anda perlu terlebih dahulu menentukan langkah-langkah yang diperlukan untuk memecahkan masalah. Jangan terburu-buru untuk langsung membuat programnya. Perhitungan dengan formula compound interest dapat anda lihat pada Gambar 1.23.



```
yuanlukito@linuxstation: ~  
yuanlukito@linuxstation:~$ python3  
Python 3.7.5 (default, Nov 20 2019, 09:21:52)  
[GCC 9.2.1 20191008] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> P = 5000000  
>>> r = 0.1  
>>> n = 1  
>>> t = 3  
>>> A = P * (1 + r/n) ** (n*t)  
>>> A  
6655000.0000000002  
>>> ■
```

Hasilnya 6.655.000
sama seperti hasil sebelumnya

Gambar 1.23: Hasil perhitungan compound interest menggunakan formula.

- # Sebelum membuat program dalam Python, pastikan anda sudah menyusun langkah-langkah penyelesaiannya terlebih dahulu. Langkah-langkah penyelesaian suatu masalah biasa disebut sebagai **algoritma**.

1.5 Latihan Mandiri

Latihan 1.1 Kerjakan Contoh 1.1, 1.2 dan 1.3 menggunakan Jupyter Notebook! ■

Latihan 1.2 Gerard membeli emas 25 gram dengan harga Rp. 650.000/gram. Jika sekarang harga emas menjadi Rp. 685.000/gram,

- Berapa keuntungan yang didapat oleh Gerard (dalam Rp dan dalam %)?
- Jika Gerard kemudian membeli lagi 15 gram emas dengan harga Rp. 685.000, maka Gerard sekarang memiliki total 40 gram emas. Jika kemudian harga emas naik lagi menjadi Rp. 715.000, berapa keuntungan yang didapat oleh Gerard (dalam Rp dan dalam %)?

Latihan 1.3 Berkaitan dengan compound interest pada Contoh 1.2 dan 1.3, jika Erika memiliki uang 200 juta rupiah dan ingin disimpan di deposito Pasti Cuan sampai uangnya menjadi **minimal** 400 juta, berapa lama waktu yang dibutuhkan?
catatan: bunga 10% per-tahun.