

PERTEMUAN 2

PENGENALAN PEMROGRAMAN PYTHON

A. TUJUAN PEMBELAJARAN

Setelah mempelajari materi ini diharapkan mahasiswa mengenal pemrograman python, installasi perangkat lunak dan mampu membuat program dengan Python.

B. URAIAN MATERI

1. Pemrograman Python

Menemukan struktur data dan algoritma mengharuskan mengomunikasikan instruksi secara rinci ke komputer. Cara terbaik untuk melakukan komunikasi semacam itu adalah menggunakan bahasa komputer tingkat tinggi, seperti Python. Pemrograman Python bahasa ini awalnya dikembangkan oleh Guido van Rossum pada awal 1990-an, dan sejak itu menjadi bahasa yang digunakan secara menonjol dalam industri dan pendidikan. Versi utama kedua dari bahasa, Python 2, dirilis pada tahun 2000, dan yang ketiga versi utama, Python 3, dirilis pada tahun 2008. tercatat bahwa ada kompatibilitas yang signifikan antara Python 2 dan Python 3.

Bahasa pemograman Python termasuk bahasa pemograman yang mudah dibaca dan terstruktur, Namun pertama-tama, akan meninjau beberapa pola untuk pemrograman dan lihat bahasa pemrograman Python untuk memastikan dalam memahami dasar struktur dan sintaks bahasa. Untuk mulai menulis program menggunakan Python, perlu menginstal Python di perangkat komputer yang sesuai dengan kapasitas memori. Pada Bahan ajar ini akan menggunakan contoh menggunakan Python 3. Python 2 tidak kompatibel dengan Python 3 jadi pastikan apabila ingin melakukan menginstal Python 3 atau yang lebih baru di komputer. Saat menulis program dalam bahasa apa pun, Lingkungan Pengembangan Terpadu yang baik (IDE) adalah alat yang berharga, jadi Anda pasti ingin memasang IDE juga.

Python secara formal adalah bahasa yang ditafsirkan. Perintah dieksekusi melalui bagian dari perangkat lunak yang dikenal sebagai julu bahasa Python.

Interpreter menerima perintah, mengevaluasi perintah tersebut, dan melaporkan hasil dari perintah tersebut. Selagi interpreter dapat digunakan secara interaktif (terutama saat debugging), seorang programmer biasanya mendefinisikan serangkaian perintah terlebih dahulu dan menyimpan perintah tersebut dalam file teks biasa yang dikenal sebagai kode sumber atau skrip. Untuk Python, kode sumber secara konvensional disimpan dalam file bermama dengan akhiran .py (mis., demo.py).

Pada sebagian besar sistem operasi, juru bahasa Python dapat dimulai dengan mengetik python dari baris perintah. Secara default, penerjemah dimulai secara interaktif mode dengan ruang kerja yang bersih. Perintah dari skrip yang telah ditentukan sebelumnya disimpan di file (mis., demo.py) dieksekusi dengan memanggil penerjemah dengan nama file sebagai argumen (mis., python demo.py), atau menggunakan flag -i tambahan untuk jalankan skrip dan kemudian masuk ke mode interaktif (mis. python -i demo.py). Banyak lingkungan pengembangan terintegrasi (IDE) menyediakan perangkat lunak yang lebih kaya platform pengembangan untuk Python, termasuk satu bermama IDLE yang disertakan dengan distribusi Python standar. IDLE menyediakan editor teks tertanam dengan dukungan untuk menampilkan dan mengedit kode Python, dan debugger dasar, memungkinkan langkah-demi-langkah eksekusi program sambil memeriksa nilai-nilai variabel kunci. Alasan utama Python untuk dipelajari oleh calon programmer atau progemmer adalah:

- a. Python sangat mudah dipelajari. Sintaksnya mudah dan kodennya sangat mudah dibaca.
- b. Python memiliki banyak aplikasi. Ini digunakan untuk mengembangkan aplikasi web, ilmu data, IoT, pengembangan aplikasi yang cepat, dan sebagainya.
- c. Hal ini memungkinkan Anda untuk menulis program dalam baris kode yang lebih sedikit daripada sebagian besar bahasa pemrograman.
- d. Ini memiliki dukungan komunitas yang sangat besar & forum aktif untuk mendukung pengguna.
- e. Kehadiran Modul Pihak Ketiga membuat bahasa Python lebih kuat.

- f. Perpustakaan pada Python dukungan ekstensif (mis: NumPy untuk perhitungan numerik, Panda untuk analisis data dll) membantu pengguna untuk memecahkan masalah besar dengan mudah.
- g. Ini memiliki struktur data yang sangat user-friendly yang menyederhanakan desain kode dan logika.
- h. Popularitas Python berkembang pesat. Sekarang ini adalah salah satu bahasa pemrograman yang paling populer.

2. Baris Perintah

Program terdiri dari perintah yang memberi tahu komputer apa yang harus dilakukan. Ini perintah disebut pernyataan, yang dijalankan komputer. Bab ini mendeskripsikan pernyataan Python yang paling sederhana dan menunjukkan bagaimana pernyataan tersebut digunakan untuk melakukan aritmatika, yang merupakan salah satu tugas paling umum untuk komputer dan juga tempat yang bagus untuk mulai belajar membuat program. Itu juga dasar dari hampir segala sesuatu yang mengikuti.

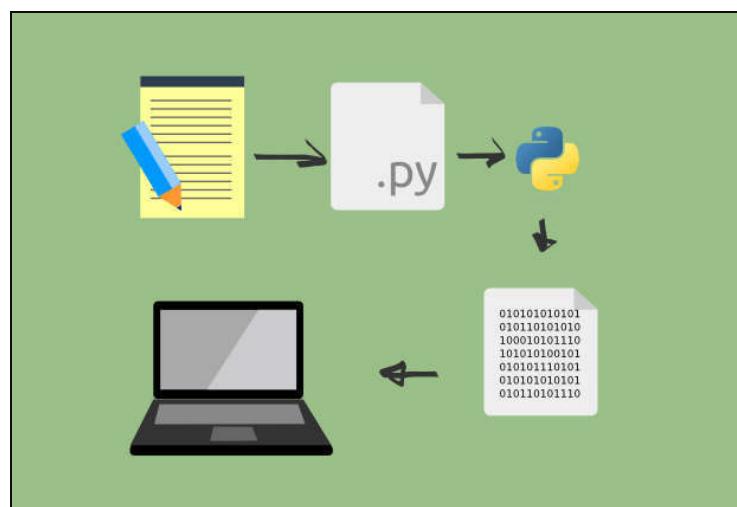
a. Aturan Penulisan

Python berorientasi objek dimana semua item data dalam Python adalah objek. Dengan Python, item data yang dapat dianggap serupa diberi nama berdasarkan tipe atau kelas. Syarat type dan class dalam Python adalah sinonim: mereka adalah dua nama untuk hal yang sama. Jadi ketika membaca tentang tipe dalam Python, baiknya untuk dapat memikirkan kelas atau sebaliknya.

Ada beberapa tipe data bawaan dalam Python termasuk int, float, str, list, dan dict yang merupakan kependekan dari kamus. Jenis data ini dan operasi terkaitnya disertakan dalam lampiran di akhir teks sehingga memiliki referensi cepat jika perlu merujuknya saat memprogram. Selain itu juga bisa mendapatkan bantuan untuk jenis apa pun dengan mengetik help(typename) di shell Python, di mana typename adalah tipe atau kelas di Python. Referensi bahasa yang sangat bagus dapat ditemukan di <http://python.org/doc>, the situs web dokumentasi resmi Python algoritma untuk menangani sejumlah besar data. Tanpa pemahaman yang tepat efisiensi, adalah mungkin untuk menghentikan komputer tercepat sekalipun

ketika bekerja dengan kumpulan data yang besar. Ini telah terjadi sebelumnya, dan akan segera mengerti betapa mudahnya hal itu bisa terjadi. Tapi pertama-tama, akan meninjau beberapa pola untuk pemrograman dan melihat bahasa pemrograman Python untuk memastikan untuk memahami dasarnya struktur dan sintaksis bahasa.

Saat menulis program dalam bahasa apa pun, Integrated Development Environment (IDE) yang baik adalah alat yang berharga sehingga ingin menginstal IDE. Ada beberapa konsep umum tentang Python yang harus ketahui saat membaca teks. Python adalah bahasa yang ditafsirkan. Itu berarti tidak harus pergi melalui langkah tambahan apa pun setelah menulis kode Python sebelum dapat menjalankannya. Kamu bisa cukup tekan tombol debug di Wing IDE dan itu untuk menyimpan program yang dibuat jika belum melakukannya setidaknya sekali, kemudian akan menjalankan program yang dibuat, tidak akan mendapatkan kesalahan apa pun sebelum menjalankan program seperti yang dilakukan dengan beberapa bahasa pemrograman.



Gambar 2. 1 Gambar Alur pembuatan python

Python yang diciptakan oleh Guido van Rossum, dan dirilis pada tahun 1991. Ini adalah open-source dan dapat dengan bebas menggunakan dan mendistribusikan Python, bahkan untuk penggunaan komersial. Ini sangat populer untuk pengembangan web dan dapat membangun hampir semua hal seperti aplikasi seluler, aplikasi web, alat, analisis data, pembelajaran mesin, dll. Ini dirancang untuk menjadi sederhana dan mudah seperti bahasa

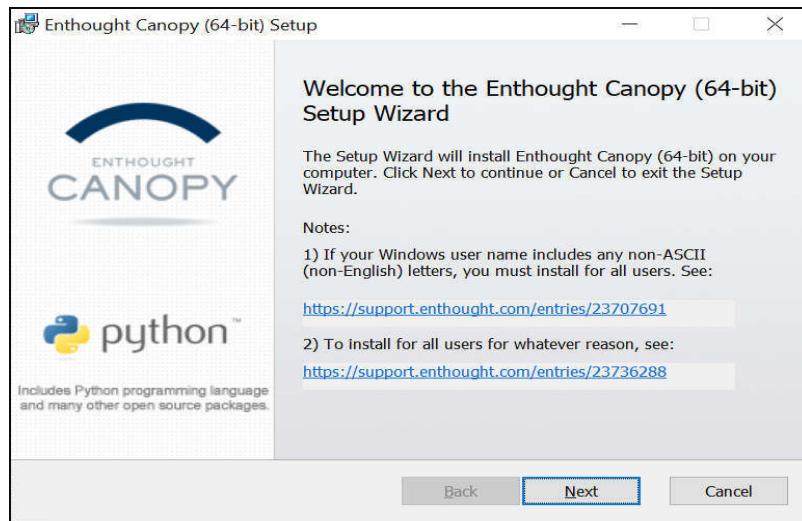
Inggris. Jauh lebih mudah untuk membaca dan menulis program Python dibandingkan dengan bahasa lain seperti C ++, Java, C #. Ini sangat produktif dan efisien yang membuatnya menjadi bahasa pemrograman yang sangat populer. Program-program yang ditulis dalam Python secara khas jauh lebih pendek dibandingkan dengan program-program C atau C++, karena beberapa pertimbangan:

- a. Tipe data tingkat tinggi digunakan untuk menyatakan operasi kompleks dalam suatu statemen tunggal pengelompokan statemen telah selesai dengan indentasi sebagai pengganti dari pengurungan mulai dan akhiran.
- b. Tidak ada deklarasi-deklarasi argumentasi atau variabel yang diperlukan.
- c. Sintaks Python sangat bergantung pada penggunaan spasi. Pernyataan individu biasanya diakhiri dengan karakter baris baru, meskipun perintah dapat diperpanjang ke baris lain, baik dengan karakter garis miring terbalik penutup (\), atau jika pembukaan pembatas belum ditutup, seperti karakter { dalam mendefinisikan peta nilai spasi juga merupakan kunci dalam membatasi badan struktur kontrol dengan Python.

Secara khusus, blok kode diindentasi untuk menetapkannya sebagai badan kontrol struktur, dan struktur kontrol bersarang menggunakan peningkatan jumlah lakukan. Komentar adalah anotasi yang disediakan untuk pembaca manusia, namun diabaikan oleh Penerjemah Python. Sintaks utama untuk komentar dalam Python didasarkan pada penggunaan dari karakter #, yang menunjuk sisa baris sebagai komentar.

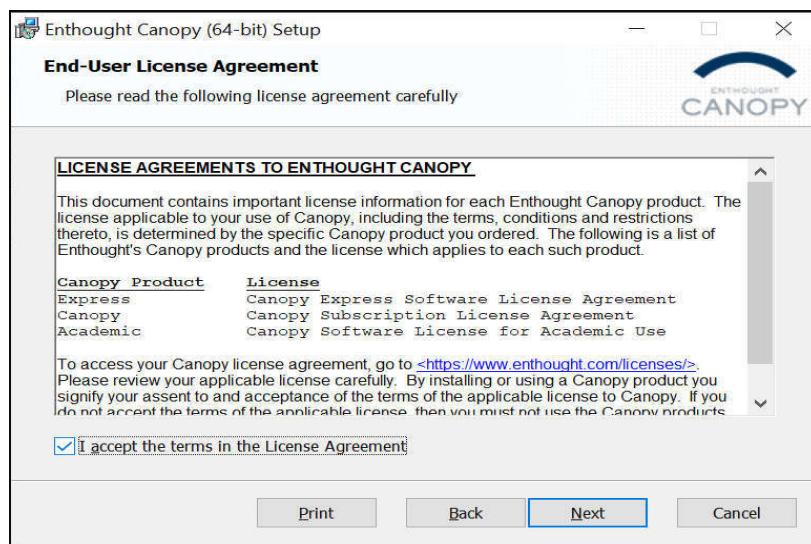
b. Instalasi Pyton

Siapkan software Pemrograman Python dapat di download diinternet, perangkat lunak yang digunakan adalah Enthought Canapy sebagai berikut.

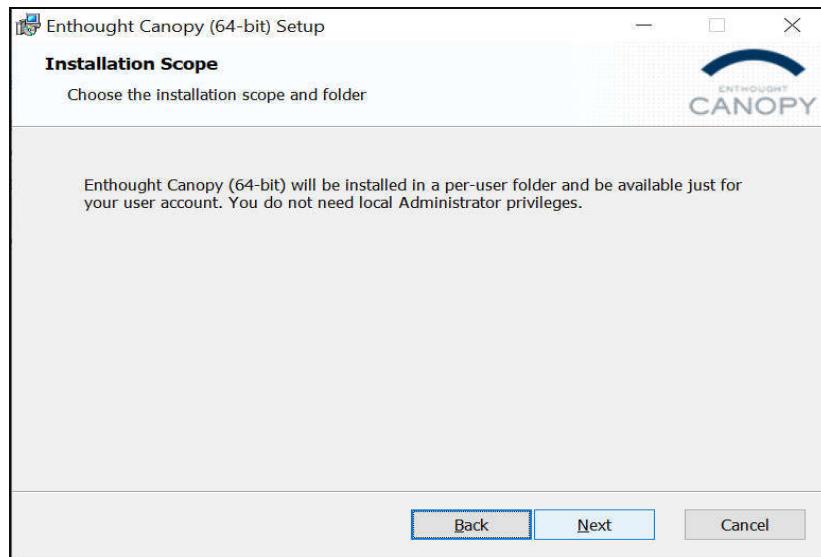


Gambar 2. 2 Installasi Python

Setelah tampil maka klik next sehingga tampil seperti gambar dibawah ini.

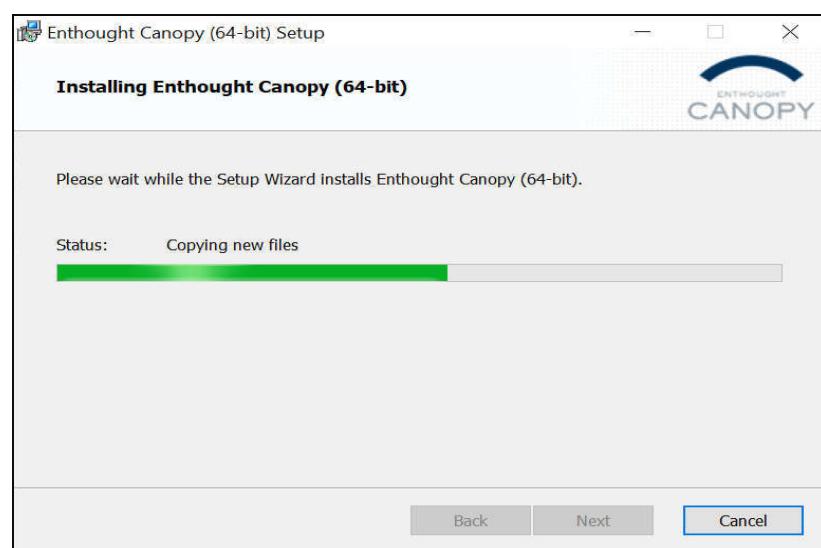


Ceklist dan persejutuan untuk melanjutkan instalasi, dan akan tampil seperti dibawah ini

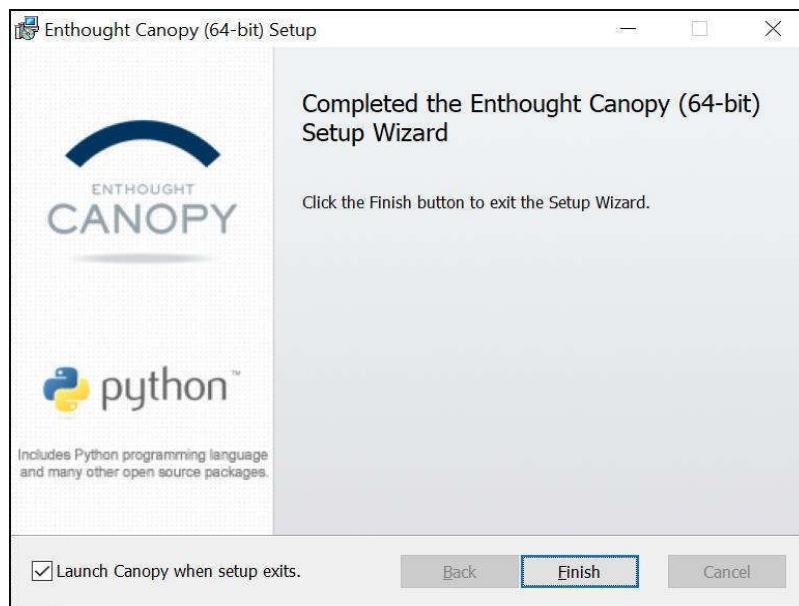


Klik next untuk melanjutkan install sehingga muncul

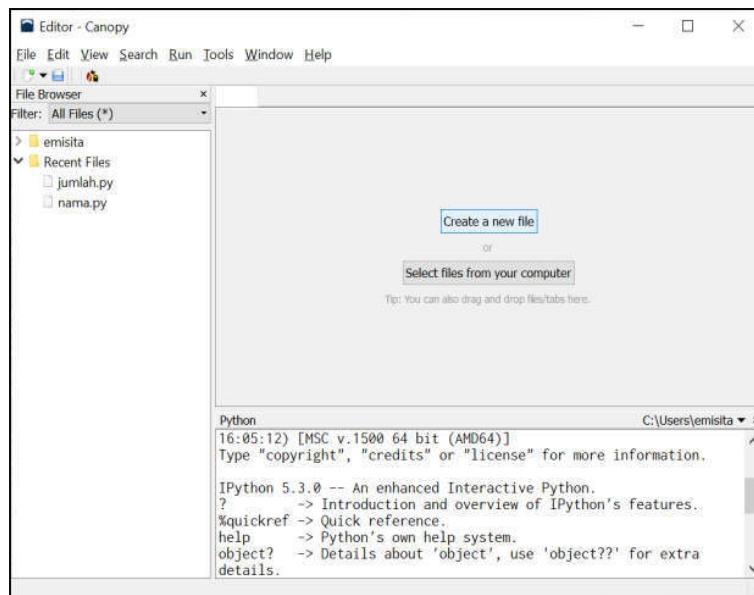
Pilih next lalu akan muncul status dalam copy file seperti dibawah ini.



Setelah 100% maka proses installasi selesai seperti dibawah



Pada tampilan selanjutnya adalah editor Canopy telah tampil seperti dibawah.



Dari tampilan diatas mengartikan bahwa pemrograman Python telah siap di gunakan pada software enthought canopy

c. Memulai Python

Bahasa ini sudah terinstal secara default di CANOPY berikut dengan editornya. Jadi hanya cukup klik icon Canopy di windows maka akan tampil seperti ini :



Untuk menulis program anda tingal klik icon editor maka akan tampil :

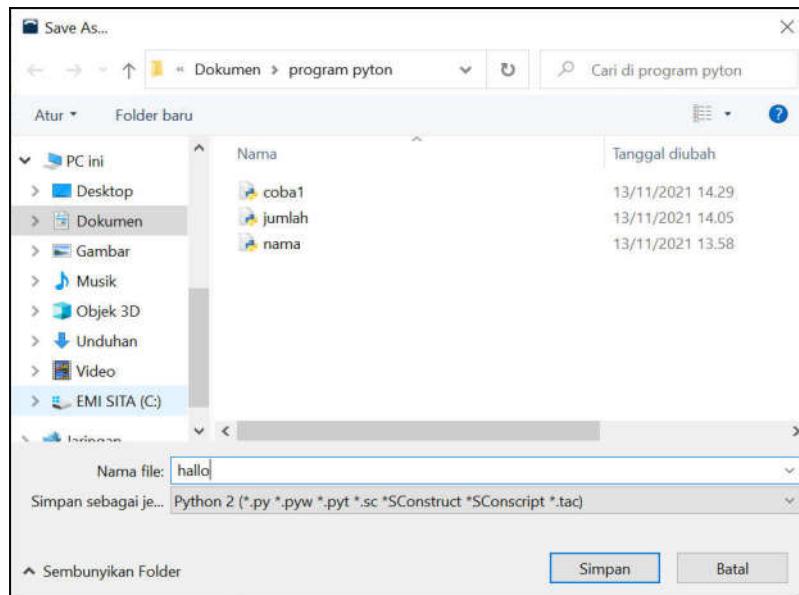
```

1 nama = "emi sita"
2 print("halo siapa namamu : " + nama)

```

The screenshot shows the 'Editor - Canopy' window. On the left is a 'File Browser' pane with a tree view showing 'emisita' and 'Recent Files' with 'jumlah.py' and 'nama.py'. The main area contains a code editor with the above Python script. At the bottom is a 'Python' terminal window showing the output of the script's execution.

Setelah melakukan penulisan sesuai sintax dan source code dapat menyimpan dalam lokasi memory yang dibuat dengan penamaan file dibelakangan .py kemudian save



Python adalah bahasa yang sangat sederhana, dan memiliki sintaks yang sangat mudah. Ini mendorong pemrogram untuk memprogram tanpa kode boilerplate (disiapkan). Direktif paling sederhana di Python adalah perintah "print" ini hanya mencetak sebuah baris (dan juga menyertakan baris baru, tidak seperti di C). Ada dua versi Python utama, Python 2 dan Python 3. Python 2 dan 3 sangat berbeda. Tutorial ini menggunakan Python 3, karena lebih tepat secara semantik dan mendukung fitur yang lebih baru.

Misalnya, satu perbedaan antara Python 2 dan 3 adalah pernyataan cetaknya. Dalam Python 2, pernyataan "print" bukanlah sebuah fungsi, dan oleh karena itu ia dipanggil tanpa tanda kurung. Namun, di Python 3, ini adalah sebuah fungsi, dan harus dipanggil dengan tanda kurung. Untuk mencetak string dengan Python 3, cukup tulis: `print("Selamat dating di Python.")`

1) Variables and Types

Python sepenuhnya berorientasi objek, dan tidak "diketik secara statis", dan tidak perlu mendeklarasikan variabel sebelum menggunakannya, atau mendeklarasikan tipenya. Setiap variabel dalam Python adalah sebuah objek. Tutorial ini akan membahas beberapa tipe dasar variabel.

Angka

Python mendukung dua jenis bilangan - bilangan bulat (bilangan bulat) dan bilangan floating point (desimal). (Ini juga mendukung bilangan kompleks, yang tidak akan dijelaskan dalam tutorial ini).

Untuk mendefinisikan integer, gunakan sintaks berikut:

```
myfloat = 7.0
print(myfloat)
myfloat = float(7)
print(myfloat)
```

2) Lists

List sangat mirip dengan array. Mereka dapat berisi jenis variabel apa pun, dan dapat berisi variabel sebanyak yang Anda inginkan. List juga dapat diulang dengan cara yang sangat sederhana. Berikut ini contoh cara membuat list

```
mylist = []
mylist.append(1)
mylist.append(2)
mylist.append(3)
print(mylist[0]) # prints 1
print(mylist[1]) # prints 2
print(mylist[2]) # prints 3

# prints out 1,2,3
for x in mylist:
    print(x)
```

a) Operator Dasar

Bagian ini menjelaskan cara menggunakan operator dasar dengan Python.

b) Operator Arithmetik

Sama seperti bahasa pemrograman lainnya, operator penjumlahan, pengurangan, perkalian, dan pembagian dapat digunakan dengan angka.

```
number = 1 + 2 * 3 / 4.0
print(number)
```

c) Menggunakan Operator dengan String

Python mendukung string penggabungan menggunakan operator penambahan:

```
helloworld = "hello" + " " + "world"  
print(helloworld)
```

Python juga mendukung perkalian string untuk membentuk string dengan urutan berulang:

```
lotsofhellos = "hello" * 10  
print(lotsofhellos)
```

Menggunakan Operator List

List dapat digabungkan dengan operator tambahan:

```
even_numbers = [2,4,6,8]  
angka_ganjil = [1,3,5,7]  
all_numbers = odd_numbers + even_numbers  
cetak (all_numbers)
```

d) Format String

Python menggunakan pemformatan string gaya-C untuk membuat string baru yang diformat. Operator "%" digunakan untuk memformat sekumpulan variabel yang diapit oleh "tuple" (list ukuran tetap), bersama dengan format string, yang berisi teks normal bersama dengan "penentu argumen", simbol khusus seperti "% s" dan "% d".

Misalkan memiliki variabel yang disebut "nama" dengan nama pengguna di dalamnya, dan kemudian ingin mencetak (salam untuk pengguna tersebut.)

```
# This prints out "Hello, John!"  
  
name = "John"  
  
print("Hello, %s!" % name)
```

Objek apapun yang bukan string dapat diformat menggunakan operator % s juga. String yang dikembalikan dari metode "repr" dari objek tersebut diformat sebagai string. Sebagai contoh:

```
# This prints out: A list: [1, 2, 3]
mylist = [1,2,3]
print("A list: % s" % mylist)
```

e) Kondisi

Python menggunakan logika boolean untuk mengevaluasi kondisi. Nilai boolean True dan False dikembalikan saat ekspresi dibandingkan atau dievaluasi. Sebagai contoh:

```
x = 2
print(x == 2) # prints out True
print(x == 3) # prints out False
print(x < 3) # prints out True
```

Perhatikan bahwa penugasan variabel dilakukan menggunakan operator sama dengan tunggal "=", sedangkan perbandingan antara dua variabel dilakukan menggunakan operator sama dengan ganda "==" . Operator "tidak sama dengan" ditandai sebagai "!=".

f) Operator Boolean

Operator "dan" dan "atau" boolean mengizinkan pembuatan ekspresi boolean yang kompleks, misalnya:

```
name = "John"
age = 23
if name == "John" and age == 23:
    print("Your name is John, and you are also 23 years old.")

if name == "John" or name == "Rick":
    print("Your name is either John or Rick.")
```

g) Operator "dalam"

Operator "dalam" dapat digunakan untuk memeriksa apakah objek tertentu ada dalam wadah objek yang dapat diulang, seperti list:

```
name = "John"
if name in ["John", "Rick"]:
    print("Your name is either John or Rick.")
```

Python menggunakan lekukan untuk menentukan blok kode, bukan tanda kurung. Indentasi Python standar adalah 4 spasi, meskipun tab dan ukuran ruang lainnya akan berfungsi, selama itu konsisten. Perhatikan bahwa blok kode tidak memerlukan penghentian apa pun. Berikut adalah contoh penggunaan pernyataan "if" Python menggunakan blok kode:

```
statement = False
another_statement = True
if statement is True:
    # do something
    pass
elif another_statement is True: # else if
    # do something else
    pass
else:
    # do another thing
    pass
```

Pernyataan dievaluasi sebagai benar jika salah satu dari berikut ini benar:

- (1) Variabel boolean "Benar" diberikan, atau dihitung menggunakan ekspresi, seperti perbandingan aritmatika.
 - (2) Sebuah objek yang tidak dianggap "kosong" dilewatkan. Berikut beberapa contoh objek yang dianggap kosong: 1. String kosong: "" 2. List kosong: [] 3. Angka nol: 0 4. Variabel boolean palsu: Salah.
- h) Operator '='

Tidak seperti operator ganda sama dengan "==" , operator "adalah" tidak cocok dengan nilai variabel, tetapi contoh itu sendiri. Sebagai contoh:

```
x = [1,2,3]
y = [1,2,3]
print(x == y) # Prints out True
print(x is y) # Prints out False
```

i) Operator "bukan"

Menggunakan "tidak" sebelum ekspresi boolean akan membalikkannya:

```
print(not False) # Prints out True
print((not False) == (False)) # Prints out False
```

j) Loop

Untuk loop, lakukan iterasi pada urutan tertentu. Berikut ini contohnya:

```
primes = [2, 3, 5, 7]
```

```
for prime in primes:
```

```
    print(prime)
```

Untuk loop dapat mengulang urutan angka menggunakan fungsi "range" dan "xrange". Perbedaan antara range dan xrange adalah bahwa fungsi range mengembalikan list baru dengan angka dari kisaran yang ditentukan, sedangkan xrange mengembalikan iterator, yang lebih efisien. (Python 3 menggunakan fungsi range, yang bertindak seperti xrange). Perhatikan bahwa fungsi rentang berbasis nol.

Pemrograman dalam bahasa berorientasi objek biasanya berarti mengimplementasikan kelas yang menggambarkan objek yang menyimpan informasi yang dibutuhkan oleh program yang sedang menulis. Objek berisi data dan metode beroperasi pada data tersebut. Sebuah kelas adalah definisi data dan metode untuk jenis objek tertentu. Setiap kelas berisi satu metode khusus yang disebut konstruktur. Konstruktur tugasnya adalah membuat instance objek dengan menempatkan referensi ke data di dalam objek diri. Misalnya, pertimbangkan kelas yang disebut hewan sapi. Seekor sapi memiliki nama, tanggal lahir, dan suara yang dihasilkannya saat mengaum.

3) Pemanggilan Method

Python mendukung fungsi tradisional (lihat Bagian 1.5) yang dipanggil dengan pajak syn seperti diurutkan(data), dalam hal ini data adalah parameter yang dikirim ke fungsi. Kelas Python juga dapat mendefinisikan satu atau lebih metode (juga dikenal sebagai anggota fungsi), yang dipanggil pada instance tertentu dari kelas menggunakan titik (“.”) operator. Misalnya, kelas daftar Python memiliki metode bernama sort yang dapat dipanggil dengan sintaks seperti data.sort(). Metode khusus ini mengatur ulang isi daftar sehingga mereka diurutkan.

Ekspresi di sebelah kiri titik mengidentifikasi objek yang menjadi dasar metode dipanggil. Seringkali, ini akan menjadi pengidentifikasi (misalnya, data), tetapi kita dapat menggunakan operator titik untuk memanggil metode pada hasil langsung dari beberapa operasi lain. Untuk contoh, jika respons mengidentifikasi instance string, sintaks response.lower() .startswith(y) pertama-tama mengevaluasi metode panggilan, response.lower(), yang dengan sendirinya mengembalikan instance string baru, dan kemudian metode startwith(y) dipanggil pada string perantara itu.

Saat menggunakan metode kelas, penting untuk memahami perilakunya. Beberapa metode mengembalikan informasi tentang keadaan suatu objek, tetapi tidak berubah yang menyatakan. Ini dikenal sebagai pengakses. Metode lain, seperti metode sortir dari kelas daftar, lakukan perubahan status objek. Metode-metode ini dikenal sebagai mutator atau metode pembaruan

C. LATIHAN SOAL

1. Jelaskan menurut anda sejarah bahasa pemrograman Python!
2. Jelaskan alasan mengapa programmer untuk mempelajari bahasa Python!
3. Jelaskan 3 perbedaan Python dengan bahasa C!
4. Jelaskan alur penyimpanan file dalam bahasa Python!
5. Jelaskan cara memanggil method pada pemrograman Python!

D. REFERENSI

- Emi Sita Eriana, A. Z. (2021). *Praktikum Algoritma dan Pemrograman*. Tangerang Selatan: Unpam Press.
- Mohamad Aslam Katahman, M. F. (2021). Pembangunan Aplikasi Realiti Terimbuh Untuk Pengenalan Struktur Data. *Information Technology And Computer Science*.
- Nasrullah, A. H. (2021). Implementasi Algoritma Decision Tree Untuk Klasifikasi Produk Laris. *Jurnal Ilmiah Ilmu Komputer I*.
- Peng Qi, Y. Z. (2020). Stanza: A Python Natural Language Processing Toolkit For Many Human Languages.
- Pradana Setialana, T. B. (2017). Pencarian Hubungan Kekerabatan Pada Struktur Data Genealogy Dalam Graph Databases.
- Ranny Meilisa, D. P. (2020). Model Pembelajaran Flipped Classroom Pada Mata Kuliah Algoritma Dan Struktur Data. *Jurnal Ilmiah Pendidikan Dan Pembelajaran (Jipp)*.
- Revanza, M. G. (2020). Struktur Data Dan Bahasa Pemrograman.
- Risah Subariah, E. S. (T.Thn.). *Praktikum Analisis & Perancangan Sistem (Uml)*.
- Sianipar, R. H. (2013). *Pemrograman & Struktur Data C: Belajar Dari Contoh Untuk Programmer Pemula Maupun Berpengalaman*. Penerbit Informatika, 2013.
- Thanaki, J. (2017). *Python Natural Language Processing*. Mambai.
- Zein, A. (2018). Pendekripsi Kantuk Secara Real Time Menggunakan Pustaka Opencv Dan Dlib Python. *Sainstech : Jurnal Penelitian Dan Pengkajian Sains Dan Teknologi*.