MODUL PRAKTIKUM
PENGANTAR PEMROGRAMAN

PROGRAM STUDI SISTEM INFORMASI
DEPARTEMEN MATEMATIKA FAKULTAS MATEMATIKA
DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN

2022

Kata Pengantar

Puji syukur ke hadirat Tuhan Yang Maha Kuasa, yang telah memberikan rahmat-Nya sehingga Pengantar Pemrograman (Python) untuk mahasiswa/i Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin ini dapat diselesaikan dengan sebaik-baiknya.

Modul praktikum ini dibuat sebagai pedoman dalam melakukan kegiatan praktikum Pengantar Pemrograman (Python) yang merupakan kegiatan penunjang mata kuliah Pemrograman Berorientasi Objek pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin. Modul praktikum ini diharapkan dapat membantu mahasiswa/i dalam mempersiapkan dan melaksanakan praktikum dengan lebih baik, terarah, dan terencana. Pada setiap topik telah ditetapkan tujuan pelaksanaan praktikum dan semua kegiatan yang harus dilakukan oleh mahasiswa/i serta teori singkat untuk memperdalam pemahaman mahasiswa/i mengenai materi yang dibahas.

Penyusun menyakini bahwa dalam pembuatan Modul Praktikum Pengantar Pemrograman (Python) ini masih jauh dari sempurna. Oleh karena itu penyusun mengharapkan kritik dan saran yang membangun guna penyempurnaan modul praktikum ini dimasa yang akan datang.

Akhir kata, penyusun mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung.

Makassar, Agustus 2022

Penyusun

DAFTAR ISI

BAB I
PENGENALAN PYTHON

Sebuah program dapat diartikan kumpulan instruksi-instruksi yang dibuat secara terstruktur dan logis untuk menyelesaikan permasalahan. Sebuah masalah memiliki makna keadaan yang tidak sesuai dengan kenyataan. Tanpa permasalahan maka tidak akan ada program. Seorang pembuat program disebut dengan programmer harus memiliki kemampuan membuat program berdasarkan ketentuan masing-masing bahasa pemrograman yang digunakan. Ada beberapa jenis bahasa pemrograman seperti C, php, java, dan python, selain itu bahasa lainnya seperti basic, pascal, cobol, dan lain-lainnya.

Software development merupakan pengembangan sebuah perangkat lunak. Selanjutnya menurut istilah, merupakan proses pengembangan sebuah aplikasi perangkat lunak yang dijalankan secara sistematis sehingga menghasilkan sebuah produk yang baik dan berkualitas.

Python adalah bahasa pemprograman open-source software development yang sangat populer, menawarkan kemampuan kontrol proses yang dikembangkan. Python mampu mengembangkan aplikasi jaringan multi-protokol yang kompleks sambil juga mempertahankan sintaks yang sederhana dan mudah. Platform seperti Google, Instagram, Spotify, dan Reddit semuanya menggunakan Python.

Capaian Pembelajaran

1. Mahasiswa diharapkan mampu menjelaskan konsep bahasa pemrograman
2. Mahasiswa diharapkan mampu membedakan kategori bahasa pemrograman
3. Mahasiswa diharapkan mampu menjelaskan histori bahasa pemrograman python

A. Sejarah Python

Guido van Rossum menjadi pencipta salah satu bahasa pemrograman yang popular yaitu python pada tahun 1990 di Be-landa tepatnya di CWI atau Centrum Wiskunde & Informatica. Penciptaan python sendiri merupakan projek kelanjutan dari bahasa pemrograman yang telah ada sebelumnya, yaitu bahasa pemrograman jenis ABC. Versi 1.2 menjadi versi terakhir python yang dirilis oleh CWI pada tahun 1995, ditahun yang sama tepatnya di Corporation for National Research Initiative (CNRI) dinegara Virginia Amerika Guido masih aktif melakukan projek pengembangan python. Pengembangan python terus dilakukan, dan pada tahun 2001, melalui Python Software Foundation (PSF) sebuah organisasi yang Guido

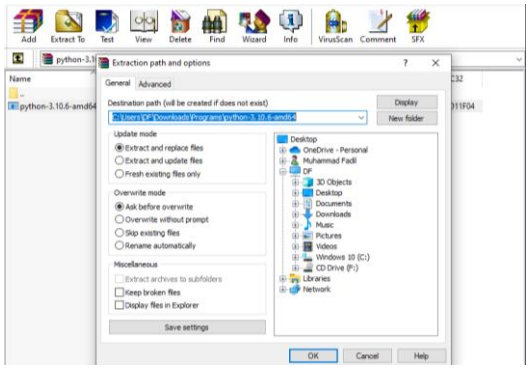
gunakan untuk mengembangkan python, melalui PSF segala hal terkait pengembangan hingga hak intelektual python dilakukan. Sekilah informasi terkait penamaan python, sebenarnya nama python yang dipakai oleh Guido bukan berasal dari nama ular yang kita kenal. Melainkan nama grup komedi dari Negara Inggris yaitu Monty Python.

Python sebagai bahasa pemrograman yang populer dan komprehensif dengan menggabungkan kapabilitas, sintaksis kode yang jelas serta dilengkapi pustaka standar yang mempunyai fungssionalitas sangat besar. Python termasuk dari jajaran bahasa pemograman tingkat tinggi seperti bahasa pemograman C, C++, Java, Perl dan Pascal. Sedangkan bahasa pemograman tingkat rendah adalah bahasa mesin yaitu bahasa pemograman Assembly. Dalam bahasa pemrograman tingkat tinggi, terbagi menjadi dua jenis cara untuk memproses bahasa tingkat tinggi ke bahasa tingkat rendah, yaitu compiler dan interpreter. Jenis pertama adalah interpreter, cara kerjanya cukup mudah, yaitu membaca sebuah program setiap baris yang ditulis dengan bahasa tingkat tinggi. Interpreter nantinya akan memproses langsung per baris untuk mengeluarkan outputnya.

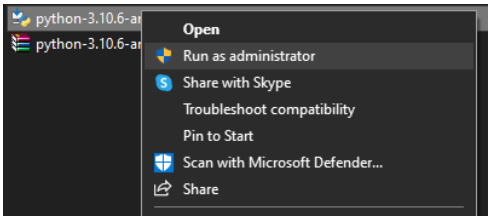
B. Instalisasi Python

Langkah – langkah untuk mengistall python di system operasi windows:

1. Download python:
   System 32bit:
   https://www.python.org/ftp/python/3.10.6/python-3.10.6-embed-win32.zip
   System 64bit:
   https://www.python.org/ftp/python/3.10.6/python-3.10.6-embed-amd64.zip
2. Extrak file yang telah didownload

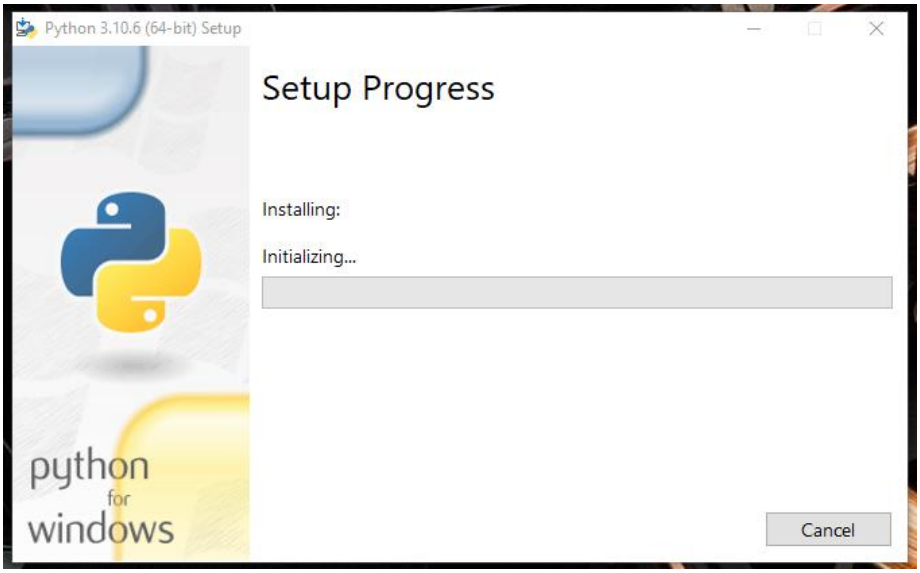


3. Klik kanan file hasil extrak, lalu pilih “Run As Administrator”

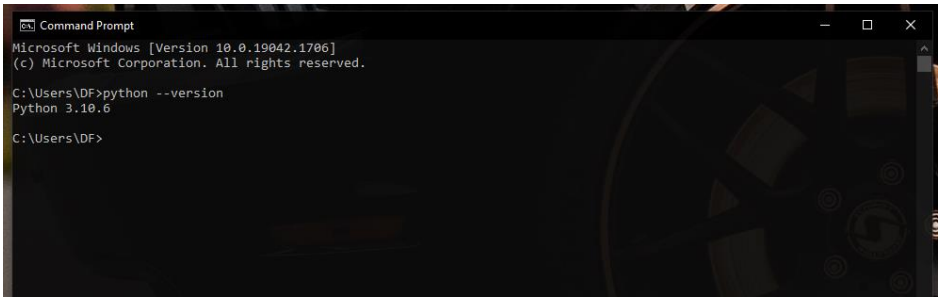


4. Beri ceklis pada “install launcher for all users” dan “Add Python 3.x to PATH”

5. Klik “Install Now”, lalu tunggu sampai proses selesai

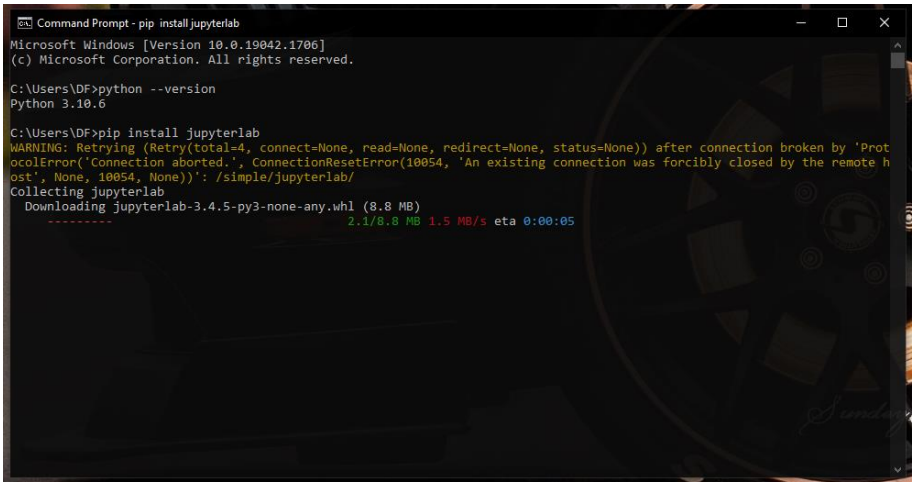


6. Setelah proses install selesai, klik “Close”, kemudian buka CMD lalu ketik python –version kemudian tekan enter.

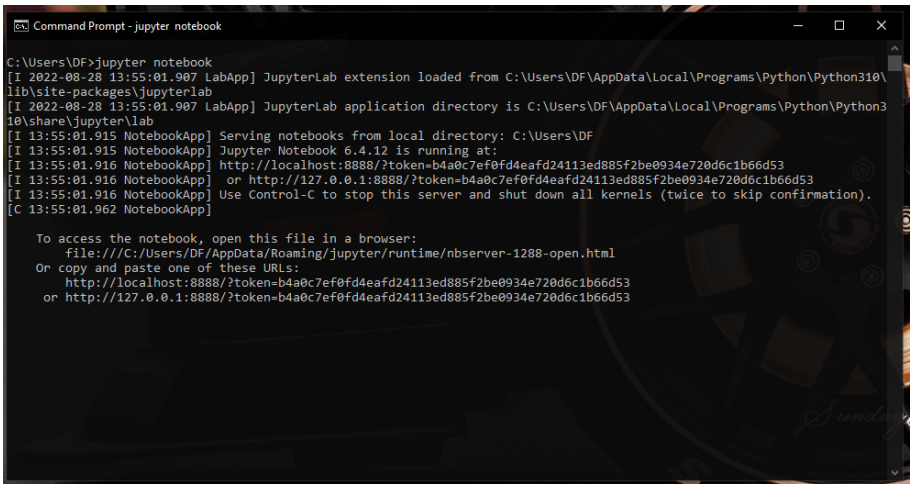


Jika tidak muncul atau tidak mengeluarkan output versi python yang terinstall, lakukan instalisasi ulang dengan memperhatikan langkah langkah yang sebelumnya.

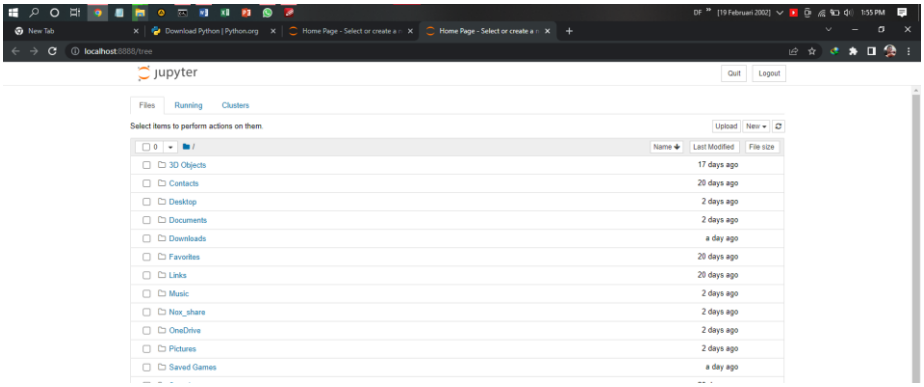
7. Kemudian ketik pip install jupyterlab



8. Untuk menjalankan program python menggunakan jupyter notebook, ketik “jupyter notebook” di cmd kemudian tunggu sampai server jupyter notebook jalan.



9. Setelah browser terbuka, maka python siap digunakan


jupyter
Quit
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
Name
Last Modified
File size
3D Objects 17 days ago
Contacts 20 days ago
Desktop 2 days ago
Documents 2 days ago
Downloads a day ago
Favorites 20 days ago
Links 20 days ago
Music 2 days ago
Nox_share 2 days ago
OneDrive 2 days ago
Pictures 2 days ago
Saved Games a day ago

TUGAS PRAKTIKUM

Download software python menggunakan link yang telah diberikan, kemudian melakukan instalisasi.

BAB II
MENGENAL ATURAN PENULISAN SINTAKS PYTHON

Jupyter Notebook merupakan tool yang populer untuk mengolah data di python. Jupyter Notebook memungkinkan untuk mengintegrasikan antara kode dengan output di dalam satu dokumen secara interaktif. Jupyter Notebook sudah otomatis terinstall ketika kita telah menginstall python dengan anaconda tetapi kita juga bias menginstall Jupyter Notebook tanpa menginstall anaconda.

Capaian Pembelajaran

1. Mahasiswa diharapkan mampu memahami dan menggunakan Identifiers, Keywords, Statements dan Expressions.
2. Mahasiswa diharapkan mampu memahami dan menggunakan Variables, Operators, Precedence dan Associativity.
3. Mahasiswa diharapkan mampu memahami dan menggunakan Data types, Indentation, Comments, Reading Input, Print Output, dan Conversi type data.

A. Identifier

Identifier atau python identifier adalah nama yang digunakan untuk mengidentifikasi atau memberikan identitas untuk variabel, function, class, module, ataupun object lainnya. Identifier bisa diawali dengan huruf A-Z atau a-z atau underscore (_) diikuti huruf, underscore, dan digit (0-9). Adapun aturan penulisan identifier di Python yaitu sebagai berikut:

- Identifier class diawali dengan huruf kapital dan identifier lainnya dengan huruf kecil.
- Identifier yang diawali underscore dimaksudkan untuk penggunaan pribadi.
- Identifier yang diawal dengan dua buah underscore adalah identifier yang sangat pribadi.
- Identifier yang diawali dan diakhiri dengan dua underscore merupakan nama khusus yang telah ditetapkan Python.

B. Keywords

Keywords pada python adalah kata-kata khusus yang memiliki arti dan tujuan tertentu dan tidak dapat digunakan untuk apa pun selain tujuan khusus tersebut. Berikut adalah keywords pada python:

```
In [1]: help("keywords")

Here is a list of the Python keywords.  Enter any keyword to get more help.

False               class               from                or
None                continue            global              pass
True                def                 if                  raise
and                 del                 import              return
as                  elif                in                  try
assert              else                is                  while
async               except              lambda              with
await               finally             nonlocal            yield
break               for                 not
```

C. Statements

Statement adalah pernyataan atau instruksi yang diberikan untuk dieksekusi oleh mesin. Interpreter Python bertugas menginterpretasikan statement menjadi perintah yang sesuai. Penulisan statement di Python tidak diakhiri dengan tanda titik koma (;). Contohnya sebagai berikut:

```
In [2]: a = 5
        b = 4
        c = a + b
        print(c)

        9
```

Dari code tersebut setiap statement dipisahkan oleh karakter atau karakter menandakan suatu baris telah selesai. Namun tanda titik koma (;) bisa digunakan untuk kasus tertentu. Misalnya menuliskan statement dalam satu baris seperti berikut:

```
In [3]: a = 5; b = 4; c = a + b
        print(c)

        9
```

Selain itu ada kalanya penulisan statement bisa sangat panjang jika dituliskan dalam satu baris. Solusinya kita bisa menuliskannya dengan multiple baris menggunakan tanda (;).

D. Expressions

Expressions adalah kombinasi dari operator yang diinterpretasikan untuk menghasilkan beberapa nilai lain. Dalam bahasa pemrograman apa pun, expressions dievaluasi sesuai dengan prioritas operatornya. Sehingga jika ada lebih dari satu operator dalam suatu expressions, maka prioritasnya menentukan operasi mana yang akan dilakukan terlebih dahulu.

Constant Expressions

Constant Expressions adalah ekspresi yang hanya memiliki nilai konstan.

Contoh:

```
x = 2
y = 3.5
z = x + y
print(z)
```

```
5.5
```

Arithmetic Expressions

Arithmetic Expressions adalah kombinasi dari nilai numerik, operator, dan terkadang tanda kurung. Hasil dari jenis ekspresi ini juga merupakan nilai numerik. Operator yang digunakan dalam ekspresi ini adalah operator aritmatika seperti penambahan, pengurangan, dll.

Contoh:

```
x = 5
y = 2

a = x * y
b = x / y

print(a)
print(b)
```

```
10
2.5
```

Integer Expressions

Integer Expressions adalah jenis ekspresi yang hanya menghasilkan hasil bilangan bulat setelah semua perhitungan dan konversi tipe datanya.

Contoh:

```
x = 5
y = int(5.5)
z = x + y

print(z)
```

```
10
```

Floating Expressions

Floating Expressions adalah jenis ekspresi yang menghasilkan angka floating point sebagai hasil setelah semua perhitungan dan konversi tipe datanya.

Contoh:

```
x = 5
y = 2
z = x / y

print(z)
```

```
2.5
```

Relational Expressions

Relational Expressions adalah ekspresi aritmatika ditulis di kedua sisi operator relasional (>, < , >= , <=). Ekspresi aritmatika tersebut dievaluasi terlebih dahulu, kemudian dibandingkan per operator relasional dan pada akhirnya menghasilkan keluaran boolean. Ekspresi ini juga disebut ekspresi Boolean.

Contoh:

```
x = 5
y = 2
a = 3
b = 4

z = (a + b) > (x - y)

print(z)
```

```
True
```

Logical Expressions

Logical Expressions adalah jenis ekspresi yang menghasilkan True atau False. Ini pada dasarnya menentukan satu atau lebih kondisi. Misalnya, (10 == 9) adalah

kondisi jika 10 sama dengan 9. Seperti yang kita ketahui tidak benar, sehingga akan mengembalikan False.

Contoh:

```
a = True
b = False
z = a and b

print(z)
```

```
False
```

Bitwise Expressions

Bitwise Expressions adalah jenis ekspresi dimana perhitungan dilakukan pada tingkat bit.

```
a = 12
x = a << 2
y = a >> 1

print(x,y)
```

```
48 6
```

Combinational Expressions

Combinational Expressions adalah jenis ekspresi dengan menggabungkan berbagai ekspresi kemudain menjadikan satu ekspresi.

Contoh:

```
a = 12
x = (a * 2) + (a << 1)

print(x)
```

```
48
```

E. Variables

Sebuah variabel dapat memiliki nama pendek (seperti x dan y) atau nama yang lebih deskriptif. Aturan untuk variabel Python:

- Nama variabel harus dimulai dengan huruf atau karakter garis bawah
- Nama variabel tidak boleh diawali dengan angka

- Nama variabel hanya boleh berisi karakter alfanumerik dan garis bawah (A-z, 0-9, dan (_)
- Nama variabel sensitif huruf besar/kecil (usia, Usia, dan USIA adalah tiga variabel berbeda)

Contoh:

```
myname = 'Fadil'
Myname = 'Jidil'
MyName = 'Ilham'
myName = 'Yoris'

print(myname, Myname, MyName, myName)
```

```
Fadil Jidil Ilham Yoris
```

F. Operators

Operator pada Pemrograman Python dapat disimbolkan dengan tanda atau karakter seperti +, -, *, /, **, % dan sebagainya. Contoh sederhana seperti operasi penjumlahan dari 1+2=3. Dimana angka 1 dan 2 disebut sebagai operand yaitu nilai yang dioperasikan oleh operator, sedangkan karakter + disebut sebagai operator. Bahasa Pemrograman Python mendukung berbagai macam jenis operator. Secara garis besar, Python memiliki tujuh jenis operator seperti:

Operator Aritmatika

Operator	Deskripsi
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
//	Pembagian (dibulatkan kebawah)
%	Sisa Bagi / Modulo

Operator Pembanding

Operator	Deskripsi
>	Lebih Besar
<	Lebih Kecil
==	Sama Dengan
!=	Tidak Sama dengan
>=	Besar Sama dengan
<=	Lebih Kecil Sama dengan

Operator Penugasan

Operator	Deskripsi
=	Assignment
+=	Penjumlahan
-=	Pengurangan
*=	Perkalian
/=	Pembagian
//=	Pembagian (dibulatkan kebawah)
%=	Sisa Bagi / Modulo

Operator Logical

Operator	Deskripsi
and	dan
or	atau
not	ingkaran/negasi

Operator Keanggotaan

Operator	Deskripsi
in	Menghasilkan nilai TRUE jika nilai yang ditentukan berada dalam objek tertentu
not in	Menghasilkan nilai TRUE jika nilai yang ditentukan tidak ada dalam objek tertentu

Operator Identitas

Operator	Deskripsi
is	Menghasilkan nilai TRUE jika kedua nilai operand memiliki identitas yang sama.
not is	Menghasilkan nilai FALSE jika kedua nilai operand memiliki identitas yang sama.

Operator Bitwise

Operator	Deskripsi
&	and
|	or
^	xor
~	not

<<	left shift
>>	right shift

Operator Precedence

Operator Precedence adalah operator yang membarikan lebih dari satu penugasan dalam satu statement.

Contoh:

```
x,y,z = 2,3,4

print(x + y)
```

```
5
```

Operator Associativity

Operator Associativity adalah operator dalam python yang dapat menggabungkan lebih dari satu operator aritmatika.

Contoh:

```
x = 2
y = (x**2) + (2*x) + 1

print(y)
```

```
9
```

G. Data Types

Dalam pemrograman, tipe data merupakan konsep penting. Variabel dapat menyimpan data dari tipe yang berbeda, dan tipe yang berbeda dapat melakukan hal yang berbeda. Berikut tipe data pada python:

- Text Type : str
- Numeric Types : int, float, complex
- Sequence Types : list, tuple, range
- Mapping Type : dict
- Set Types : set, frozenset
- Boolean Type : bool
- Binary Types : bytes, bytearray, memoryview
- None Type : NoneType

Contoh:

Example	Data Type
`x = "Hello World"`	str
`x = 20`	int
`x = 20.5`	float
`x = 1j`	complex
`x = ["apple", "banana", "cherry"]`	list
`x = ("apple", "banana", "cherry")`	tuple
`x = range(6)`	range
`x = {"name" : "John", "age" : 36}`	dict
`x = {"apple", "banana", "cherry"}`	set
`x = frozenset({"apple", "banana", "cherry"})`	frozenset
`x = True`	bool
`x = b"Hello"`	bytes
`x = bytearray(5)`	bytearray
`x = memoryview(bytes(5))`	memoryview
`x = None`	NoneType

H. Indentation

Indentasi adalah penulisan yang agak menjorok masuk ke dalam. Dalam sebuah artikel biasanya indentasi digunakan untuk menunjukkan paragraf baru atau paragraf selanjutnya. Indentasi pada Python penting dipahami agar tidak terjadi error saat menjalankan program, contohnya pada penulisan code untuk looping atau perulangan seperti berikut:

```
print('Hi....')
print("I'm Python")
    print('Ini Error')
```

```
  Input In [14]
    print('Ini Error')
    ^
IndentationError: unexpected indent
```

Contoh penulisan yang benar:

```
x = 2
if x < 4:
    print('x kurang dari 4')
```

```
x kurang dari 4
```

I. Comments

Penulisan code dalam Python kita juga bisa menambahkan komentar. Komentar adalah sebuah baris code atau statement yang diabaikan interpreter Python. Komentar biasanya dituliskan hanya sebagai catatan atau penjelasan suatu baris code. Penulisan komentar pada Python terdiri dari 2 jenis yaitu satu baris dan multi baris. Komentar satu baris dituliskan dengan tanda pagar (#), sedangkan multi baris dituliskan dengan tanda petik dua sebanyak tiga kali. Contohnya sebagai berikut:

```
# Ini adalah contoh komentar

## Ini Juga masih komentar ##

# Penjumlahan
x = 2
y = 3

a = x + y

print(a) # untuk menampilkan nilai a
```

```
5
```

J. Reading

Reading adalah mode akses untuk mengatur jenis operasi yang mungkin dilakukan dalam file yang dibuka. Ini mengacu pada bagaimana file akan digunakan setelah dibuka. Mode ini juga menentukan lokasi File yang akan dibuka. Ada 6 mode akses di python yaitu:

('r') : Buka file teks untuk membaca. Jika file tidak ada, memunculkan kesalahan I/O. Ini juga merupakan mode default di mana file dibuka.

('r+') : Buka file untuk membaca dan menulis. Menimbulkan kesalahan I/O jika file tidak ada.

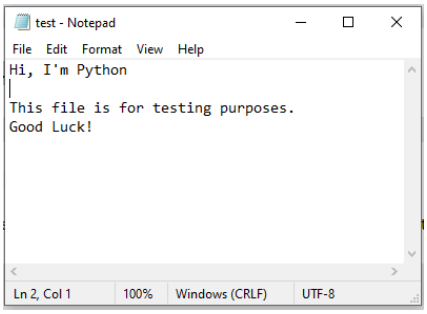
('w') : Buka file untuk menulis. Untuk file yang ada, data dipotong dan ditulis ulang. Membuat file jika file tidak ada.

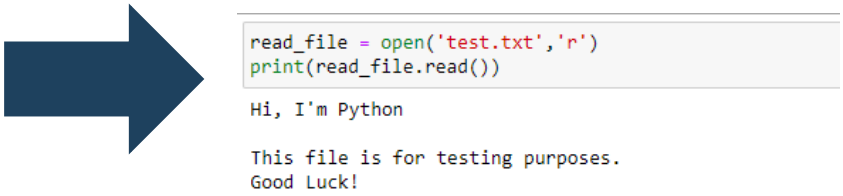
('w+') : Buka file untuk membaca dan menulis. Untuk file yang sudah ada, data dipotong dan ditulis ulang.

('a') : Buka file untuk menulis. File dibuat jika tidak ada. Data yang ditulis akan disisipkan di akhir, setelah data yang ada.

('a+') : Buka file untuk membaca dan menulis. File dibuat jika tidak ada. Data yang ditulis akan disisipkan di akhir, setelah data yang ada.

Contoh:





K. Input

Python sudah menyediakan fungsi 'input()' dan 'raw_input()' untuk mengambil inputan dari keyboard.

Contoh:

```
name = input('Nama   : ')
nim  = input('Nim    : ')
prodi= input('Prodi  : ')

print('\nHasil:')
print('Nama  :', name)
print('Nim   :', nim)
print('Prodi :', prodi)
```

```
Nama   :
```

Dan contoh hasil runingnya:

```
name = input('Nama   : ')
nim  = input('Nim    : ')
prodi= input('Prodi  : ')

print('\nHasil:')
print('Nama  :', name)
print('Nim   :', nim)
print('Prodi :', prodi)
```

```
Nama   : Siti Aisyah Arsella
Nim    : H051171017
Prodi  : Statistika

Hasil:
Nama  : Siti Aisyah Arsella
Nim   : H051171017
Prodi : Statistika
```

L. Conversi Type Data

Conversi type data pada python yaitu melakukan perubahan type data pada suatu vaiabel. Baik itu string ke integer atau pun sebalinya.

Contoh:

```
x = 2.5    # Float
y = int(x) # Float ke Integer

print(type(x))
print(type(y))
```

```
<class 'float'>
<class 'int'>
```

Contoh lain:

```
x = 2      # Integer
y = str(x) # Integer ke String

print(type(x))
print(type(y))
```

```
<class 'int'>
<class 'str'>
```

TUGAS PRAKTIKUM

Buatlah program sederhana yang dapat menginput informasi pribadi mahasiswa, lalu hitung umur mahasiswa tersebut.

Contoh:

```
nama = input('Nama          : ')
nim  = input('Nim           : ')
prodi= input('Prodi         : ')
fak  = input('Fakultas      : ')
ttl  = input('Tanggal Lahir : ')

## Lanjutkan sintaksnya ##
.
.
.
```

```
Nama          :
```

Output:

```
Nama          : Siti Ihza Arsella Kalsim
Nim           : H051171017
Prodi         : Sistem Informasi
Fakultas      : MIPA
Tanggal Lahir : 26-10-1999

Biodata Mahasiswa
Nama          : Siti Ihza Arsella Kalsim
Nim           : H051171017
Prodi         : Sistem Informasi
Fakultas      : MIPA
Tanggal lahir : 26-10-1999 (23 tahun)
```

BAB III
CONTROL FLOW STATEMENTS

Control flow statements adalah urutan eksekusi kode program. Control flow program Python diatur oleh pernyataan kondisional, loop, dan panggilan fungsi.

Capaian Pembelajaran

1. Mahasiswa diharapkan mampu menggunakan statemen if, elif, dan else.
2. Mahasiswa diharapkan mampu menggunakan perulangan while dan for, serta penggunaan break dan continue.
3. Mahasiswa diharapkan mampu menggunakan statement try – exception.

A. Selection/Decision Control Statements

Dalam Bahasa pemprograman python, pernyataan seleksi juga dikenal sebagai pernyataan control flow atau pernyataan percabangan. Pernyataan seleksi memungkinkan program untuk menguji beberapa kondisi dan mengeksekusi instruksi berdasarkan kondisi yang benar.

Percabangan if

Perintah if dalam python berfungsi melakukan eksekusi atau menjalankan statement jika kondisinya benar.

Contoh:

```
x = 7
if (x % 2 != 0):
    print('Bilangan Ganjil')

if (x % 2 == 0):
    print('Bilangan Genap')
```

```
Bilangan Ganjil
```

Percabangan if … else

Percabangan if … else yaitu jika kondisi bernilai benar (true) maka perintah dijalankan, tetapi jika perintah bernilai salah, makan statement yang di jalankan adalah statement yang terdapa pada perintah else.

Contoh:

```
x = 7
if (x % 2 == 0):
    print('Bilangan Ganjil')
else:
    print('Bilangan Genap')
```

```
Bilangan Genap
```

Percabangan if, elif, dan else

Percabangan ini digunakan jika memiliki banyak kemungkinan yang terjadi.

Contoh:

```
x = 8
if (x % 2 == 0):
    print(x,'Habis di bagi 2')
elif (x % 3 == 0):
    print(x,'Habis di bagi 3')
elif (x % 4 == 0):
    print(x,'Habis di bagi 4')
elif (x % 5 == 0):
    print(x,'Habis di bagi 5')
else:
    print(x,'Tidak habis di bagi 2,3,4, atau 5')
```

```
8 Habis di bagi 2
```

B. Repetition

Pernyataan pengulangan digunakan untuk mengulang sekelompok (blok) instruksi atau beberapa statemen. Dalam Python, biasanya memiliki dua loop/perulangan yaitu perulangan for, dan perulangan while.

Perulangan for

Perulangan for digunakan untuk mengulangi urutan yang berupa list, tupel, dictionary, atau set.

Contoh:

```
x = 3
for i in range(0,x):
    print('Hello, I\'m Python')
```

```
Hello, I'm Python
Hello, I'm Python
Hello, I'm Python
```

Perulangan while

Perulangan dengan menggunakan perintah while yaitu perulangan yang dilakukan secara terus menerus sampai kondisi bernilai salah.

Contoh:

```
x = 3
while(x > 0):
    print('Hello, I\'m Python')
    x -= 1
```

```
Hello, I'm Python
Hello, I'm Python
Hello, I'm Python
```

Penggunaan Break dan Continue

Break adalah perintah yang digunakan untuk menghentikan secara paksa suatu perulangan. Sedangkan continue statement berikutnya adakan dieksekusi jika kondisinya benar.

Contoh penggunaan break:

```
x = 3
for i in range(0,3):
    print('Hello, I\'m python')
    break
```

```
Hello, I'm python
```

Contoh penggunaan continue:

```
x = 3
for i in range(0,3):
    if i == 1:
        continue
    print('Hi...')
```

```
Hi...
Hi...
```

Penggunaan statement try-exception

Ketika terjadi kesalahan, atau exception, Python biasanya akan berhenti dan menghasilkan pesan kesalahan. Pengecualian ini dapat ditangani menggunakan pernyataan try:

Contoh:

```
x = 3
y = 'Empat'
try:
    z = x + y
except:
    z = str(x) + y

print(z)
```

```
3Empat
```

Contoh:

```
x = 3
y = 'Empat'
try:
    z = x + y
except:
    z = y + str(x)
else:
    print('Tidak dapat menjumlahkan keduanya')

print(z)
```

```
Empat3
```

Contoh:

```
x = 3
y = 'Empat'
try:
    z = x + y
except:
    z = y + str(x)
finally:
    print('!!! salah satunya dilakukan conversi type data')

print(z)
```

```
!!! salah satunya dilakukan conversi type data
Empat3
```

TUGAS PRAKTIKUM

Buatlah program sederhana yaitu konversi angka ke huruf atau huruf ke angka yaitu 0-9. Contoh programnya:

Jika pilih 1

```
========================================
        Konversi huruf ke angka
========================================
----------------------------------------
1. Konversi angka ke huruf
2. Konversi huruf ke angka
----------------------------------------
Pilih: 1
Input angka:5
----------------------------------------
Angka : 5
Huruf : Lima
----------------------------------------
```

Jika pilih 2

```
========================================
        Konversi huruf ke angka
========================================
----------------------------------------
1. Konversi angka ke huruf
2. Konversi huruf ke angka
----------------------------------------
Pilih: 2
Input huruf:Tiga
----------------------------------------
huruf : Tiga
angka : 3
----------------------------------------
```

Jika salah input:

```
========================================
        Konversi huruf ke angka
========================================
----------------------------------------
1. Konversi angka ke huruf
2. Konversi huruf ke angka
----------------------------------------
Pilih: 3
----------------------------------------
[!] Error: Inputan salah
----------------------------------------
1. Konversi angka ke huruf
2. Konversi huruf ke angka
----------------------------------------
Pilih: p
----------------------------------------
[!] Error: Inputan salah
----------------------------------------
1. Konversi angka ke huruf
2. Konversi huruf ke angka
----------------------------------------
Pilih: -
----------------------------------------
[!] Error: Inputan salah
----------------------------------------
1. Konversi angka ke huruf
2. Konversi huruf ke angka
----------------------------------------

Pilih:
```

BAB IV
FUNCTIONS PADA PYTHON

Fungsi adalah blok kode yang hanya berjalan ketika dipanggil. Anda dapat meneruskan data, yang dikenal sebagai parameter, ke dalam suatu fungsi. Sebuah fungsi dapat mengembalikan data sebagai hasilnya.

Capaian Pembelajaran

1. Mahasiswa diharapkan mampu membuat build in function pada python
2. Mahasiswa diharapkan mampu membuat fungsi dan penggunaannya

A. Creating a Function

Pembuatan function dalam python didefinisikan menggunakan kata kunci def kemudian diikuti dengan nama funtionya. Kemudian pemanggilan function tersebut cukup memanggil nama funtionnya.

Contoh:

```
# Function
def message():
    print('Hello... I\'m python')

# Pemanggilan funtion message
message()
```

```
Hello... I'm python
```

B. Arguments

Informasi dapat diteruskan ke fungsi sebagai argumen. Argumen ditentukan setelah nama fungsi, di dalam tanda kurung. Dapat menambahkan argumen sebanyak yang diinginkan, cukup pisahkan dengan koma (,).

Contoh:

```
def name(argument):
    print('Hi... my name is '+argument)
name("Ihza")
name("Aisyah")
```

```
Hi... my name is Ihza
Hi... my name is Aisyah
```

Contoh:

```
def name(argument, age):
    print('Hi... my name is '+argument+ ' ('+str(age)+' year)')
name("Ihza", 24)
name("Aisyah", 25)
```

```
Hi... my name is Ihza (24 year)
Hi... my name is Aisyah (25 year)
```

C. Arbitrary Arguments

Jika tidak diketahui berapa banyak argumen yang akan diteruskan ke dalam funtion, tambahkan * sebelum nama parameter dalam definisi fungsi. Cara ini fungsi akan menerima tupel argumen, dan dapat mengakses item yang sesuai.

Contoh:

```
def my_name(*name):
    print('Hi... my name is '+name[3])

my_name('Fadl','Ihza', 'Ilham', 'Aisyah', 'Jidil')
```

```
Hi... my name is Aisyah
```

D. Default Parameter Value

Default argument yaitu pemberian nilai default pada argument di dalam function. Jika kita memanggil fungsi tanpa argumen, ia menggunakan nilai default tersebut.

Contoh:

```
def my_name(name = 'Python'):
    print('Hi... I\'m '+name)

my_name()
my_name('Muhammad Fadil')
```

```
Hi... I'm Python
Hi... I'm Muhammad Fadil
```

E. Return Values

Suatu fungsi mengembalikan nilai jika menggunakan pernyataan return.

Contoh:

```
def tambah(a,b):
    return(a+b)

print(tambah(2,3))
```

```
5
```

TUGAS PRAKTIKUM

1. Buatlah program konvers uang rupiah ke dollar sederhana pada python dengan menerapkan konsep function.
2. Buatlah program kalkulator sederhana menggunakan function pada python.

BAB V
FUNGSI STRING

String dalam python dikelilingi oleh tanda kutip tunggal, atau tanda kutip ganda. 'Halo' sama dengan “halo”, Hal tersebut dapat menampilkan string literal dengan fungsi print().

Capaian Pembelajaran

1. Mahasiswa diharapkan mampu menggunakan operasi-operasi string dasar.
2. Mahasiswa diharapkan mampu mencari dan menemukan substring dari suatu string.
3. Mahasiswa diharapkan mampu memanipulasi string.

A. Membuat Variabel String

Menetapkan string ke variabel dilakukan dengan nama variabel diikuti dengan tanda sama dengan dan string (beritanda kutip).

Contoh:

```
nama = 'Siti'
nim  = 'H051171017'
umur = "23"

print(type(nama), type(nim), type(umur), sep='\n')
```

```
<class 'str'>
<class 'str'>
<class 'str'>
```

Ketiga variable diatas merupakan type data string. Jadi semua yang diberi tanda kutip ganda merupakan type data string

B. Multiline Strings

String juga dapat menetapkan string multiline ke variabel dengan menggunakan tiga tanda kutip

Contoh:

```
message = """Hi let's learn python,
come as a beginner
come home as a legend."""
print(message)
```

```
Hi let's learn python,
come as a beginner
come home as a legend.
```

C. String adalah Array

Seperti banyak bahasa pemrograman populer lainnya, string dalam Python adalah type data array yang mewakili karakter unicode. Namun, Python tidak memiliki tipe data karakter, karakter tunggal hanyalah string dengan panjang satu. Untuk mengakses elemen string digunakan tanda kurung siku.

Contoh:

```
message = "Hi.... I'm DF"
print(message[7])
print(message[:5])
print(message[-2:])
```

```
I
Hi...
DF
```

D. String Length

Untuk mendapatkan panjang string, gunakan fungsi len().

Contoh:

```
message = "Hi.... I'm DF"

print(len(message))
print(len(message[4:]))
```

```
13
9
```

TUGAS PRAKTIKUM

Buatlah program berdialog dengan computer. Contoh program:

```
Python: Hi..., saya adalah bahasa pemprograman python,
        nama kamu siapa?


User  : Hi... juga, nama saya muhammad fadil
```

Jika di enter maka muncul:

```
Python: Hi..., saya adalah bahasa pemprograman python,
        nama kamu siapa?
User  : Hi... juga, nama saya muhammad fadil

Python: Hi..., Muhammad Fadil. Nama yang bagus,
        kamu lahir tahun berapa?


User  :
```

Dan seterusnya sampai selesai, contoh:

```
Python: Hi..., saya adalah bahasa pemprograman python,
        nama kamu siapa?
User  : Hi... juga, nama saya muhammad fadil

Python: Hi..., Muhammad Fadil. Nama yang bagus,
        kamu lahir tahun berapa?
User  : saya lahir tahun 1997

Python: Wah sekarang kamu berusia kurang lebih 25 tahun.
        Muhammad Fadil sekarang tinggal dimana?
User  : Sekarang saya tinggal di Makassar

Python: Muhammad Fadil sekarang sudah kerja atau masih kuliah ?
User  : Saya sudah kerja

Python: Wah.. semangat yah kerjanya.
        Senang berkomunikasi dengan muhammad fadil.
        Sampai ketemu lagi di lain waktu
```

BAB VI
LIST

List digunakan untuk menyimpan beberapa item dalam satu variabel. List adalah salah satu dari 4 tipe data bawaan dalam Python yang digunakan untuk menyimpan kumpulan data, 3 lainnya adalah Tuple, Set, dan Dictionary, semuanya dengan kualitas dan penggunaan yang berbeda.

Capaian Pembelajaran

1. Mahasiswa diharapkan mampu membuat dan memanipulasi item dalam List.
2. Mahasiswa diharapkan mampu melakukan pengindeksan dan pemotongan dalam List.
3. Mahasiswa diharapkan mampu mengunakan metode yang terkait dengan List.
4. Mahasiswa diharapkan mampu menggunakan List sebagai argumen dalam Function.
5. Mahasiswa diharapkan mampu mengunakan perulangan for untuk mengakses item individual dalam List.

A. Membuat List

Membuat data bertype list dalam python, sangatlah mudah, cukup memebri nama variable lalu dibuka dan ditutup dengan kurung siku.

Contoh

```
my_list = [] ## List kosong ##
my_list_1 = ['Fadil','Jidil','Ihza','Ilham']
my_list_2 = [1,3,5,7,9,11,12,13,15]
my_list_3 = ['A',100,'B',85,'C',60]

print(my_list_1)
print(my_list_2)
print(my_list_3)
```

```
['Fadil', 'Jidil', 'Ihza', 'Ilham']
[1, 3, 5, 7, 9, 11, 12, 13, 15]
['A', 100, 'B', 85, 'C', 60]
```

Selain dapat membuat list dengan isi list tersebut sekumpulan data yang memiliki type data yang berbeda, kita juga dapat mengetahui panjang atau banyaknya data yang terdapat dalam list tersebut dengan menggunakan perintah len().

Contoh:

```
my_list_1 = ['Fadil','Jidil','Ihza','Ilham']
my_list_2 = [1,3,5,7,9,11,12,13,15]
my_list_3 = ['A',100,'B',85,'C',60]

print(len(my_list_1))
print(len(my_list_2))
print(len(my_list_3))
```

```
4
9
6
```

B. Mengakses Elemen List

Untuk mengakses item list, lihat nomor indeksnya. Gunakan operator indeks [] untuk mengakses item dalam list tersebut. Indeks harus berupa bilangan bulat.

Contoh:

```
my_list = [] ## List kosong ##
my_list_1 = ['Fadil','Jidil','Ihza','Ilham']
my_list_2 = [1,3,5,7,9,11,12,13,15]
my_list_3 = ['A',100,'B',85,'C',60]

print(my_list_1[0])
print(my_list_2[:6])
print(my_list_3[:4])
```

```
Fadil
[1, 3, 5, 7, 9, 11]
['A', 100, 'B', 85]
```

Contoh:

```
my_list_1 = [['Fadil','Jidil','Ihza','Ilham'],['H01','H02','H03']]

print(my_list_1[:4][0])
```

```
['Fadil', 'Jidil', 'Ihza', 'Ilham']
```

C. Menambahkan Elemen ke dalam List

Elemen dapat ditambahkan ke dalam list dengan menggunakan fungsi append() dan fungsi insert(). Hanya satu elemen pada satu waktu yang dapat ditambahkan ke list dengan menggunakan metode append(), untuk penambahan beberapa elemen dengan metode append(), kita dapat menggunakan metode looping atau perulangan.

Contoh:

```
my_list_1 = ['Fadil', 'Jidil','Ilham','Siti','Aisyah']
print(my_list_1)

my_list_1.append('Ihza')
print(my_list_1)
```

```
['Fadil', 'Jidil', 'Ilham', 'Siti', 'Aisyah']
['Fadil', 'Jidil', 'Ilham', 'Siti', 'Aisyah', 'Ihza']
```

Contoh:

```
my_list_1 = ['Fadil', 'Jidil','Ilham','Siti','Aisyah']
print(my_list_1)

my_list_1.insert(1,'Ihza')
print(my_list_1)
```

```
['Fadil', 'Jidil', 'Ilham', 'Siti', 'Aisyah']
['Fadil', 'Ihza', 'Jidil', 'Ilham', 'Siti', 'Aisyah']
```

D. Menghapus Elemen dari List

Elemen dapat dihapus dari list dengan menggunakan fungsi remove(), tetapi kesalahan muncul jika elemen tidak ada dalam daftar. Metode Remove() hanya menghapus satu elemen pada satu waktu, untuk menghapus berbagai elemen, iterator digunakan. Metode remove() menghapus item yang ditentukan.

Contoh:

```
my_list_1 = ['Fadil', 'Jidil','Ilham','Siti','Aisyah']
print(my_list_1)

my_list_1.remove('Ilham')
print(my_list_1)
```

```
['Fadil', 'Jidil', 'Ilham', 'Siti', 'Aisyah']
['Fadil', 'Jidil', 'Siti', 'Aisyah']
```

TUGAS PRAKTIKUM

Buatlah program sederhana untuk menginput data mahasiswa, kemudian data tersebut dapat edit. Terdapat 5 menu pada program, kelima menut tersebut memiliki fungsi yang berbeda.

Output program seperti berikut:

```
=====================================================
                        Menu
-----------------------------------------------------
0.] Exit
1.] Tambah data Mahasiswa
2.] Hapus datat Mahasiswa
3.] Edit data Mahasiswa
4.] Tampilkan daftar Mahasiswa
=====================================================
-----------------------------------------------------
> Pilih:1
-----------------------------------------------------
Nama         : Muhammad Fadil
Nim          : H005
Jenis Kelamin: Laki-Laki
-----------------------------------------------------
             Sukses menambahkan data
-----------------------------------------------------

=====================================================
                        Menu
-----------------------------------------------------
0.] Exit
1.] Tambah data Mahasiswa
2.] Hapus datat Mahasiswa
3.] Edit data Mahasiswa
4.] Tampilkan daftar Mahasiswa
=====================================================
-----------------------------------------------------
> Pilih:1
-----------------------------------------------------
Nama         : Bana Bana
Nim          : H006
Jenis Kelamin: Waria
-----------------------------------------------------
             Sukses menambahkan data
-----------------------------------------------------
```

BAB VII
DICTIONARY

Dictionary adalah kumpulan nilai kunci, yang digunakan untuk menyimpan nilai data key dan value, tidak seperti list yang hanya menyimpan satu nilai sebagai elemen.

Capaian Pembelajaran

1. Mahasiswa diharapkan mampu membuat dan memanipulasi pasangan key:value dalam Dictionary.
2. Mahasiswa diharapkan mampu mengunakan metode yang terkait dengan Dictionary.
3. Mahasiswa diharapkan mampu menggunakan perulangan for untuk mengakses pasangan key:value dalam Dictionary.

A. Membuat Dictionary

Dalam Python , dictionary bisa dibuat dengan menempatkan urutan elemen dalam kurung kurawal, dipisahkan dengan 'koma'. Kamus menampung pasangan nilai, satu menjadi Kunci dan elemen pasangan terkait lainnya menjadi Kunci:nilainya . Nilai dalam kamus dapat berupa tipe data apa pun dan dapat diduplikasi, sedangkan kunci tidak dapat diulang dan tidak dapat diubah .

Catatan – Kunci (key) dictionary peka huruf besar/kecil, nama yang sama tetapi huruf besar Kunci yang berbeda akan diperlakukan secara berbeda.

Contoh:

```
dictionary = {1: 'Matematika', 2: 'Statistika', 3: 'Sistem Informasi'}

print("\nDictionary: ")
print(dictionary)
```

```
Dictionary:
{1: 'Matematika', 2: 'Statistika', 3: 'Sistem Informasi'}
```

B. Menambahkan Elemen Dictionary

Penambahan elemen dapat dilakukan dengan berbagai cara. Satu nilai pada satu waktu dapat ditambahkan ke Kamus dengan mendefinisikan nilai bersama dengan

kuncinya misalnya Dict[Key] = 'Value'. Memperbarui nilai yang ada dalam Kamus dapat dilakukan dengan menggunakan metode update(). Nilai kunci bersarang juga dapat ditambahkan ke Kamus yang ada.

```
dictionary = {1: 'Matematika', 2: 'Statistika', 3: 'Sistem Informasi'}

dictionary[4] = 'Data Science'
dictionary[5] = 'Aktuaria'

print("\nDictionary: ")
print(dictionary)
```

```
Dictionary:
{1: 'Matematika', 2: 'Statistika', 3: 'Sistem Informasi', 4: 'Data Scienc
e', 5: 'Aktuaria'}
```

C. Mengakses Elemen Dictionary

Untuk mengakses item dalam dictionary, lihat nama kuncinya. Kunci (key) dapat digunakan di dalam tanda kurung siku.

Contoh:

```
dictionary = {1: 'Matematika', 2: 'Statistika', 3: 'Sistem Informasi'}

dictionary[4] = 'Data Science'
dictionary[5] = 'Aktuaria'

print("\nDictionary: ")
print(dictionary)

print()
print('Key 1:',dictionary[1])
print('Key 2:',dictionary[2])
```

```
Dictionary:
{1: 'Matematika', 2: 'Statistika', 3: 'Sistem Informasi', 4: 'Data Scienc
e', 5: 'Aktuaria'}

Key 1: Matematika
Key 2: Statistika
```

Mengakses elemen dictionary menggunakan perulangan for

Contoh:

```
dictionary = {1: 'Matematika', 2: 'Statistika', 3: 'Sistem Informasi'}

dictionary[4] = 'Data Science'
dictionary[5] = 'Aktuaria'

for i in dictionary:
    print('Key',i,':',dictionary[i])
```

```
Key 1 : Matematika
Key 2 : Statistika
Key 3 : Sistem Informasi
Key 4 : Data Science
Key 5 : Aktuaria
```

TUGAS PRAKTIKUM

Buatlah program perpusatakaan sederhana menggunakan metode dictonary, dalam program buku dapat dipijam dan memiliki batas peminjaman, serta buku dalam perpustakaan dapat diupdate (tambah, edit, dan delete), dan terdapat daftar denda peminjaman buku.

BAB VIII
TUPLES DAN SETS

Tuple dan Set adalah salah satu dari 4 tipe data bawaan dalam Python yang digunakan untuk menyimpan kumpulan data, 2 lainnya adalah List dan Dictionary semuanya dengan kualitas dan penggunaan yang berbeda.

Capaian Pembelajaran

1. Mahasiswa diharapkan mampu Membuat dan memanipulasi item dalam Tuple.
2. Mahasiswa diharapkan mampu mengunakan for loop untuk mengakses item individual dalam Tuple.
3. Mahasiswa diharapkan mampu menjelaskan hubungan antara Dictionary dan Tuple.
4. Mahasiswa diharapkan mampu menjelaskan hubungan antara List dan Tuple.
5. Mahasiswa diharapkan mampu menerapkan operasi matematika seperti union dan intersection menggunakan Set.

A. Tupel

Tuple digunakan untuk menyimpan beberapa item dalam satu variabel. Tuple adalah kumpulan item dan tidak dapat diubah, penulisan tuple dengan tanda kurung.

Contoh:

```
my_tuple_1 = ('Matematika','Statistika','Sistem Informasi')
my_tuple_2 = (1,'Matematika',False, 'Statistika',True,'Sistem Informasi')

print(my_tuple_1)
print(my_tuple_2)
```

```
('Matematika', 'Statistika', 'Sistem Informasi')
(1, 'Matematika', False, 'Statistika', True, 'Sistem Informasi')
```

Sama halnya dengan list, tuple juga bisa memiliki item yang berbeda type datanya serta juga dapat diketahui panjangnya atau banyaknya item di dalam tuple dengan menggunakan fungsi len()

Contoh:

```
my_tuple = ('Matematika','Statistika','Sistem Informasi')

print('Panjang tuple:',len(my_tuple))
```

```
Panjang tuple: 3
```

B. Manipulasi Tuple

Tuple kumpulan beberapa item dan tidak dapat diubah lagi. Tetapi kita bisa melakukan manipulasi tuple tersebut, dengan cara melakukan konversi type data tuple ke list, lalu data list yang kita ubah, setelah diubah, dilakukan kembali konversi data dari list ke tuple.

Contoh:

```
my_tuple = ('Matematika','Statistika','Sistem Informasi')

conv_list = list(my_tuple)
conv_list.append('Aktuaria')
conv_list.remove('Statistika')
conv_list.insert(1,'Data Science')
my_tuple = tuple(conv_list)

for i in my_tuple:
    print(i)
```

```
Matematika
Data Science
Sistem Informasi
Aktuaria
```

C. Set

Set adalah kumpulan yang tidak berurutan, tidak dapat diubah, dan tidak terindeks dan tidak ada anggota yang terduplikat.

Contoh:

```
my_set = {1, 'Matematika',False, 'Statistika',True,'Sistem Informasi'}
print(my_set)
```

```
{False, 1, 'Matematika', 'Statistika', 'Sistem Informasi'}
```

Sama halnya dengan tuple, set juga bisa memiliki item yang berbeda type datanya serta juga dapat diketahui panjangnya atau banyaknya item di dalam set dengan menggunakan fungsi len()

Contoh:

```
my_set = {1, 'Matematika',False, 'Statistika',True,'Sistem Informasi'}
print('Banyak item:',len(my_set))
```

```
Banyak item: 5
```

D. Manipulasi Set

Sama halnya dengan tuple, set juga kumpulan beberapa item dan tidak dapat diubah lagi. Tetapi kita bisa melakukan manipulasi set tersebut, dengan cara melakukan konversi type data set ke list, lalu data list yang kita ubah, setelah diubah, dilakukan kembali konversi data dari list ke set.

Contoh:

```
my_set = {1, 'Matematika',False, 'Statistika',True,'Sistem Informasi'}
my_list = list(my_set)
my_list.remove(1)
my_list.remove(False)
my_list.append('Aktuaria')
my_list.insert(1,'Data Science')
my_set = set(my_list)

for i in my_set:
    print(i)
```

```
Aktuaria
Matematika
Data Science
Statistika
Sistem Informasi
```

TUGAS PRAKTIKUM

Buatlah program perpusatakaan sederhana menggunakan metode tuples dan set, dalam program buku dapat dipijam dan memiliki batas peminjaman, serta buku dalam perpustakaan dapat diupdate (tambah, edit, dan delete), dan terdapat daftar denda peminjaman buku.

BAB IX
FILE

Python juga mendukung penanganan file dan memungkinkan pengguna untuk menangani file yaitu untuk membaca dan menulis file. Konsep penanganan file telah meluas ke berbagai bahasa pemprograman lain, tetapi implementasinya rumit atau panjang. Tidak seperti konsep Python, konsep ini juga mudah dan singkat. Python memperlakukan file secara berbeda sebagai teks atau biner. Setiap baris kode mencakup urutan karakter dan dapat membentuk file teks. Setiap baris file diakhiri dengan karakter khusus, yang disebut karakter EOL atau End of Line seperti koma {,} atau karakter baris baru untuk mengakhiri baris.

Capaian Pembelajaran

1. Mahasiswa diharapkan mampu memilih file teks atau file biner untuk membaca dan menulis data.
2. Mahasiswa diharapkan mampu mendemonstrasikan penggunaan fungsi bawaan untuk menavigasi sistem file.
3. Mahasiswa diharapkan mampu manfaatkan modul os untuk mengoperasikan tugas Sistem Operasi yang mendasarinya.

A. Penanganan File

Fungsi utama untuk bekerja dengan file dalam Python adalah open(). Fungsi open() membutuhkan dua parameter yaitu nama file, dan mode. Ada empat metode (mode) yang berbeda untuk membuka file dalam python yaitu:

`"r"`: Baca - Nilai default. Membuka file untuk membaca, kesalahan terjadi jika file tidak ada

`"a"`: Tambahkan - Membuka file untuk ditambahkan, membuat file jika tidak ada

`"w"`: Tulis - Membuka file untuk ditulis, membuat file jika tidak ada

`"x"`: Buat - Membuat file yang ditentukan, mengembalikan kesalahan jika file ada

Selain itu, Anda dapat menentukan apakah file harus ditangani sebagai mode biner atau teks

`"t"`: Teks - Nilai default. Modus teks

`"b"`: Biner - Modus biner (misalnya gambar)

B. Membuka File

Untuk membuka file pada python menggunakan fungsi open() atau dapat dilihat pada sintak berikut:

```
file = open("nama_file.type", "mode")
```

Atau

```
with open('nama_file.type','mode')
```

Contoh:

```
nama = open("Nama.txt",'r')
print(nama.read())
```

```
Shaff Shalihin
Fadillah Cheryl Ananda Putri
Muh. Ilham Maulana Ramlan
Muhammad Ihsanul Mumtaz
Khaerurrozikin
Arni Raihanah Rahman
Muhammad Nabil Shadiq
Aan Syawaluddin Adi Putra
```

Contoh:

```
nama = []
with open("Nama.txt") as txt:
    for i in txt:
        nama.append(i[:-1])
        print(i[:-1])
```

```
Shaff Shalihin
Fadillah Cheryl Ananda Putri
Muh. Ilham Maulana Ramlan
Muhammad Ihsanul Mumtaz
Khaerurrozikin
Arni Raihanah Rahman
Muhammad Nabil Shadiq
```

C. Membuat File

Untuk membuat file pada python menggunakan fungsi open() dengan mode ‘w’ atau dapat dilihat pada sintak berikut:

```
nama_variabel = open("Nama File.type",'w')
```

Contoh:

```
nama = open("Nama Mahasiswa.txt",'w')
nama.write("Muhammad Fadil\n")
nama.write("Masjidil Aqsah\n")
nama.write('Ihza Kurniawan\n')
nama.write('Ade Kurniawan\n')
nama.write("Hilmi Abyan\n")
nama.write("Yoris Rombe\n")
nama.write("Ilham\n")
nama.close()
```

Hasil:

TUGAS PRAKTIKUM

Buatlah file daftar nama dan nim mahasiswa, kemudian data tersebut dapat dicetak dan dapat di modifikasi (tambah, edit, dan delete).

Contoh hasil:

```
-------------------------------------------------------
 | Nama                              | NIM          |
-------------------------------------------------------
 | Shaff Shalihin                    | H071221093   |
 | Fadillah Cheryl Ananda Putri      | H071221077   |
 | Muh. Ilham Maulana Ramlan         | H071221035   |
 | Muhammad Ihsanul Mumtaz           | H071221015   |
-------------------------------------------------------
```

BAB X
REGULAR EXPRESSION OPERATIONS

Regular Expression (RegEx) adalah urutan karakter yang membentuk pola pencarian. RegEx dapat digunakan untuk memeriksa apakah string berisi pola pencarian yang ditentukan atau tidak.

Capaian Pembelajaran

1. Mahasiswa diharapkan mampu membuat ekspresi reguler yang cocok dengan pola teks.
2. Mahasiswa diharapkan mampu menerapkan ekspresi reguler ke teks menggunakan metode dari modul re.
3. Mahasiswa diharapkan mampu menggunakan metakarakter dalam membangun ekspresi reguler.
4. Mahasiswa diharapkan mampu menggunakan operasi yang umum digunakan yang melibatkan ekspresi reguler.
5. Mahasiswa diharapkan mampu menggunakan ekspresi reguler untuk pencarian teks atau penggantian string.

A. Modul RegEx

Python memiliki paket atau package bawaan bernama re, yang dapat digunakan untuk bekerja dengan RegEx dengan cara Impor remodel seperti berikut:

```
import re
```

Setelah melakukan impor remodul, barulah dapat mulai menggunakan ekspresi regular

Contoh:

```
import re

txt = "Hi... I'm a legend programmer"
x = re.search("^Hi.*legend*", txt)
print(x)
```

```
<re.Match object; span=(0, 18), match="Hi... I'm a legend">
```

B. Fungsi RegEx

Modul `re` ini menawarkan serangkaian fungsi yang memungkinkan kita mencari string untuk kecocokan:

Findall	:	Returns daftar yang berisi semua kecocokan.
Search	:	Returns objek perbandingan jika ada kecocokan dimana saja pada string.
Split	:	Returns daftar dimana string telah dibagi di setiap perbandingan.
Sub	:	Replaces satu atau banyak kecocokan dengan string.

Contoh fungsi findall:

```
import re

txt = "Hi... I'm a legend programmer"
x = re.findall("I'm", txt)
print(x)
```

```
["I'm"]
```

Contoh fungsi search:

```
import re

txt = "Hi... I'm a legend programmer"
x = re.search("^Hi.*legend*", txt)
print(x)
```

```
<re.Match object; span=(0, 18), match="Hi... I'm a legend">
```

Contoh fungsi split:

```
import re

txt = "Hi... I'm a legend programmer"
x = re.findall("\s", txt)
print(x)
```

```
[' ', ' ', ' ', ' ']
```

Contoh fungsi sub:

```
import re

txt = "Hi... I'm a legend programmer"
x = re.sub("\s","_", txt)
print(x)
```

```
Hi..._I'm_a_legend_programmer
```

C. Metakarakter

Metakarakter adalah karakter dengan arti khusus dalam modul re. berikut daftar karakter tersebut:

Karakter	Deskripsi
[]	Satu set karakter
\	Urutan khusus (dapat juga digunakan untuk menghindari karakter khusus)
.	Semua karakter (kecuali karakter baris baru)
^	Dimulai dengan
$	Berakhir dengan
*	Nol atau lebih kejadian
+	Satu atau lebih kejadian
?	Nol atau satu kejadian
{}	Persis jumlah kejadian yang ditentukan
|	atau
()	Pengelompokan

D. Special Sequences

Special sequaences adalah salah satu kareakter yang diikuti tanda “\” dan memiliki arti khusus, berikut daftar arti kususnya:

Karakter	Deskripsi
\A	Mengembalikan kecocokan jika karakter yang ditentukan berada di awal string
\b	Mengembalikan kecocokan di mana karakter yang ditentukan berada di awal atau di akhir kata ("r" pada awalnya memastikan bahwa string diperlakukan sebagai "string mentah")

\B	Mengembalikan kecocokan di mana karakter yang ditentukan ada, tetapi BUKAN di awal (atau di akhir) kata ("r" pada awalnya memastikan bahwa string diperlakukan sebagai "string mentah")
\d	Mengembalikan kecocokan di mana string berisi angka (angka dari 0-9)
\D	Mengembalikan kecocokan di mana string TIDAK mengandung angka
\s	Mengembalikan kecocokan di mana string berisi karakter spasi
\S	Mengembalikan kecocokan di mana string TIDAK mengandung karakter spasi
\w	Mengembalikan kecocokan di mana string berisi karakter kata apa pun (karakter dari a hingga Z, angka dari 0-9, dan garis bawah _ karakter)
\W	Mengembalikan kecocokan di mana string TIDAK mengandung karakter kata apa pun
\Z	Mengembalikan kecocokan jika karakter yang ditentukan berada di akhir string

E. Sets

Set adalah himpunan karakter di dalam sepasang tanda kurung siku "[]" dengan memiliki arti khusus. Berikut daftar arti kusus tersebut:

Karakter	Deskripsi
[arn]	Mengembalikan kecocokan dimana salah satu karakter yang ditentukan (a, r, atau n) ada
[a-n]	Mengembalikan kecocokan untuk setiap karakter huruf kecil, menurut abjad antara a dan n
[^arn]	Mengembalikan kecocokan untuk karakter apa pun KECUALI a, r, dan n
[0123]	Mengembalikan kecocokan dimana salah satu digit yang ditentukan (0, 1, 2, atau 3) ada
[0-9]	Mengembalikan kecocokan untuk setiap digit antara 0 dan 9
[0-5][0-9]	Mengembalikan kecocokan untuk angka dua digit dari 00 dan 59

[a-zA-Z]	Mengembalikan kecocokan untuk karakter apa pun menurut abjad antara a dan z, huruf kecil ATAU huruf besar
[+]	Dalam set, +, *, ., |, (), $,{} tidak memiliki arti khusus, jadi [+] berarti: mengembalikan kecocokan untuk setiap karakter + dalam string

TUGAS PRAKTIKUM

Carilah text bacaan, serta setarakan sumbernya (baik berupa link atau berupa jurnal atau buku). Kemudian teksi apakah teks bacaan tersebut mengandung hoax atau tidak.

BAB XI
OBJECT ORIENTED PROGRAMMING

Sama halnya dengan Bahasa pembrograman lain seperti java, C, javascript, dll. Python juga memiliki metode object oriented programming atau lebih sering kita dengar dengan istila OOP atau PBO. Python adalah bahasa pemrograman berorientasi objek. Hampir semua yang ada di Python adalah objek, dengan properti dan metodenya.

Capaian Pembelajaran

1. Mahasiswa diharapkan mampu membuat objek.
2. Mahasiswa diharapkan mampu mengenali atribut data dan metode untuk objek yang diberikan.
3. Mahasiswa diharapkan mampu menggunakan notasi titik untuk mengakses atribut data dan metode suatu objek.
4. Mahasiswa diharapkan mampu mendemonstrasikan implementasi variabel instan, metode, dan konstruktor.
5. Mahasiswa diharapkan mampu menggunakan Enkapsulasi, Polimorfisme dan Warisan.

A. Membuat Class

Pembuatan class dalam Bahasa pembrograman python sangatlah mudah.

Contoh:

```
1  class Message:
2      message = 'Hello.... ini adalah pesan class'
3
4  pesan = Message.message
5  print(pesan)
```

```
Hello.... ini adalah pesan class
```

Contoh diatas dapat dilihat. Bahwa line 1-2 adalah class

B. Membuat Object

Pembuatan object dalam Bahasa pemprograman python cukup memanggil nama class-nya titik (.) lalu diikuti dengan nama variable yang ingin kita akses:

Contoh

```
class Message:
    message = 'Hello.... ini adalah pesan class'

## Object ##
pesan = Message.message
print(pesan)
```

```
Hello.... ini adalah pesan class
```

Contoh lain:

```
class Message:
    message = 'Hello.... ini adalah pesan class'

## Object ##
pesan = Message()
print(pesan.message)
```

```
Hello.... ini adalah pesan class
```

C. Function __init__()

Untuk memahami arti kelas, kita harus memahami fungsi __init__(), fungsi ini merupakan function bawaan python. Semua kelas memiliki fungsi yang disebut __init__(), yang selalu dijalankan ketika kelas sedang dimulai. Fungsi __init__() digunakan untuk menetapkan nilai ke properti objek, atau operasi lain yang perlu dilakukan saat objek sedang dibuat.

Contoh:

```
class Person:
    def __init__(self, name, nim):
        self.name = name
        self.nim = nim

Mahasiswa1 = Person("Fadil", "H12115006")

print('Nama :',Mahasiswa1.name)
print('NIM  :',Mahasiswa1.nim)
```

```
Nama : Fadil
NIM  : H12115006
```

D. Membuat Method dalam Class

Class juga dapat berisi metode. Metode dalam class adalah fungsi yang dimiliki clas tersebut.

Contoh:

```python
class Person:
    def __init__(self, name, nim):
        self.name = name
        self.nim = nim

    def info_person(self):
        print("Hello my name is " + self.name)

Mahasiswa1 = Person("Fadil", "H12115006")
Mahasiswa1.info_person()
```

```
Hello my name is Fadil
```

TUGAS PRAKTIKUM

Buatlah daftar nama dan nim mahasiswa dan data tersebut dapat dimodifikasi (tambah, edit, dan delete) menggunakan konsep OOP.

BAB XII
INTRODUCTION TO DATA SCIENCE

Seiring berkembangnya zaman. Bahasa pemprograman python sangat terkenal, selain dari bahasa pemprogramanya yang cukup sederhana dan mudah dipahami, bahasa pemprograman python juga sangat mudah digunakan untuk mengolah data. Maka bahasa pemprograman python sangat cocok digunakan dalam data science.

Capaian Pembelajaran

1. Mahasiswa diharapkan mampu menjelasan konsep pemrograman fungsional.
2. Mahasiswa diharapkan mampu melakukan serialisasi dan deserialisasi objek JSON.
3. Mahasiswa diharapkan mampu mendemonstrasikan penerapan Modul Numpy dan pandas.
4. Mahasiswa diharapkan mampu enampilkan grafik menggunakan perpustakaan visualisasi Altair.

A. Data Scientist

Data scientist adalah pekerjaan dalam bentuk penggabungan ilmu komputer (pemrograman), statistik, dan matematika yang bertujuan untuk mengumpulkan, menafsirkan, dan menganalisis kumpulan data besar yang terstruktur dan tidak terstruktur. Seorang data scientist biasanya bekerja dalam tim untuk mengumpulkan berbagai data dan informasi.

Data tersebut digunakan untuk memprediksi perilaku pelanggan dan mengidentifikasi peluang bisnis yang lebih baru. Pada praktiknya data scientist juga melakukan eksperimen terhadap data-data yang telah dikumpulkan dengan maksud membuktikan dan memberikan solusi yang paling tepat untuk perkembangan sebuah usaha atau bisnis.

Dalam struktur kerja, seorang data scientist umumnya melakukan laporan kerja kepada pemimpin proyek/departemen, Chief Data Officer, atau kepala analytics dalam tim analisis data yang lebih besar.

Tugas dan tanggung jawab data scientist

1. Mengidentifikasi sumber pengumpulan data untuk kebutuhan bisnis.
2. Memproses, membersihkan, dan mengintegrasikan data.
3. Pengumpulan data otomatis dan proses manajemen.

4. Menganalisis data dalam jumlah besar untuk memperkirakan tren dan memberikan laporan dengan rekomendasi.
5. Berkolaborasi dengan tim bisnis, teknik, dan produk.
6. Mengembangkan, mengimplementasikan, dan memelihara database.
7. Menghasilkan informasi dan wawasan dari kumpulan data dan mengidentifikasi tren dan pola.
8. Menyiapkan laporan untuk tim eksekutif dan proyek.

Skill yang dibutuhkan untuk menjadi data scientist

Dalam melakukan berbagai aktivitas yang berhubungan dengan pengelolaan data, kamu membutuhkan beberapa kemampuan berikut untuk dapat menjadi seorang data scientist.

1. Menganalisis data

Kemampuan menganalisis data ini termasuk salah satu aspek utama ketika bekerja sebagai data scientist. Bukan hanya sekedar menganalisis, kamu juga harus dapat menghasilkan visualisasi data (penyajian data dalam format bergambar atau grafik sehingga mudah dipahami dan dianalisis).

2. Kemampuan statistik

Secara sederhana, statistik yang digunakan dalam pekerjaan ini lebih kepada penggunaan sampel dan populasi. Selain itu, data scientist juga menggunakan statistik dasar untuk memberi gambaran pada data yang akan diolah dan dianalisis. Sehingga dapat menghasilkan data yang sesuai dengan perencanaan (dibutuhkan perusahaan).

3. Kemampuan pemrograman

Salah satu kemampuan utama yang dibutuhkan sebagai data scientist adalah dapat melakukan pemrograman dan menggunakan tools yang membantu dalam menganalisis data. Kamu juga harus memiliki pengetahuan dan pengalaman coding dengan beberapa bahasa seperti R, Python, dan lainnya. Adapun pemrograman dengan penggunaan bahasa seperti python ini berfungsi untuk memudahkan para data scientist untuk mengatur atau mengorganisir kumpulan data yang tidak terstruktur. Jika kamu ingin mempelajari dasar-dasar pemrograman tersebut, kamu bisa langsung mengikuti kelasnya dengan mengklik banner di bawah ini.

4. Kemampuan Database, Query (SQL), dan Pengolahan Data

Seorang data scientist membutuhkan SQL (Structured Query Language) untuk menangani data terstruktur yang ada pada database sehingga menjadi lebih mudah. Adapun hal ini juga termasuk pemahaman mengenai SQL commands seperti:

1. Data Query Language
2. Data Manipulation Language
3. Data Definition Language
4. Data Control Language

5. Pemahaman Bisnis

Selain pengetahuan akan teknis data, kamu juga harus memiliki pemahaman mengenai bisnis. Sering kali seorang data scientist berada dalam diskusi atau rapat bisnis yang ditugaskan untuk mengomunikasikan ide-ide kompleks dan membuat keputusan organisasi (perusahaan) berdasarkan data.

B. Mengambil Data API menggunakan Python

Mengambil data dengan API dari Kaggle

Link data json: https://api.covid19india.org/state_district_wise.json

```
import requests
import json
response_API = requests.get('https://api.covid19india.org/
                            'state_district_wise.json')

data = response_API.text
parse_json = json.loads(data)

active_case = parse_json['Andaman and Nicobar Islands']
                        ['districtData']['South Andaman']
                        ['active']
print("Active cases in South Andaman:", active_case)
```

```
Active cases in South Andaman: 19
```

C. Mengolah Data Menggunakan Python

Pengolaan data menggunakan python memiliki dua package yang sangat penting yaitu numpy dan pandas. Jadi pertama-tama kita harus mengimpor kedua package tersebut.

Contoh:

```
import pandas as pd
import numpy as np
```

Setelah mengimport kedua package, kita read file excel type csv ataupun xlsx

Contoh:

```
## Load data csv ##
data_set = pd.read_csv('FuelConsumptionCo2.csv')

## Data Set ##
data_set.head()
```

	MODELYEAR	MAKE	MODEL	VEHICLECLASS	ENGINESIZE	CYLINDERS	TRANSMISSION
0	2014	ACURA	ILX	COMPACT	2.0	4	AS5
1	2014	ACURA	ILX	COMPACT	2.4	4	M6
2	2014	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7
3	2014	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6
4	2014	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6

Saat data berhasil direcord. Sebelum melakukan pengolaan data, terlebih dahulu cek data tersebut apakah data lengkap (tidak ada yang kosong atau nan)

```
## Cek Data Set ##
data_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1067 entries, 0 to 1066
Data columns (total 13 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   MODELYEAR                 1067 non-null   int64
 1   MAKE                      1067 non-null   object
 2   MODEL                     1067 non-null   object
 3   VEHICLECLASS              1067 non-null   object
 4   ENGINESIZE                1067 non-null   float64
 5   CYLINDERS                 1067 non-null   int64
 6   TRANSMISSION              1067 non-null   object
 7   FUELTYPE                  1067 non-null   object
 8   FUELCONSUMPTION_CITY      1067 non-null   float64
 9   FUELCONSUMPTION_HWY       1067 non-null   float64
 10  FUELCONSUMPTION_COMB      1067 non-null   float64
 11  FUELCONSUMPTION_COMB_MPG  1067 non-null   int64
 12  CO2EMISSIONS              1067 non-null   int64
dtypes: float64(4), int64(4), object(5)
memory usage: 108.5+ KB
```

Data yang dapat diolah ada data yang type datanya berupa integer (int64) atau float (float64). Kemudian lakukan seleksi variable. Seleksi variable bertujuan untuk mengetahui variable mana saja yang bisa dilakukan pengolahan.

```
1 data_set.corr()
```

UMPTION_HWY	FUELCONSUMPTION_COMB	FUELCONSUMPTION_COMB_MPG	CO2EMISSIONS
NaN	NaN	NaN	NaN
0.778746	0.819482	-0.808554	0.874154
0.724594	0.776788	-0.770430	0.849685
0.965718	0.995542	-0.935613	0.898039
1.000000	0.985804	-0.893809	0.861748
0.985804	1.000000	-0.927965	0.892129
-0.893809	-0.927965	1.000000	-0.906394
0.861748	0.892129	-0.906394	1.000000

Fokus pada variable CO2EMISSION, variable yang digunakan adalah variable yang nilai korelasinya lebih dari atau sama dengan 0.5 dalam artian variable yang memiliki nilai korelasi atau hubungan korelasi yang kuat. Kemudian mengambil variable yang akan diolah

```
1 data_olah = data_set[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_CON
2 data_olah.head()
```

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244

Kemudian data yang siap diolah di bagi dua, yaitu data training dan data olah.

Data training adalah data yang digunakan untuk membangun model, setelah model diperoleh, jika model yang diperoleh akan di terapkan pada data testing. Jika model tersebut memiliki nilai akurasi yang sukup besar maka model diterpakan untuk data berikutnya dengan kata lain model bisa digunakan untuk memprediksi.

```
1 msk = np.random.rand(len(data_olah)) < 0.8
2 data_train = data_olah[msk]
3 data_test = data_olah[~msk]
```

Kemudian dilakukan pengolahan data

```
from sklearn import linear_model
model = linear_model.LinearRegression()

train_x = np.asanyarray(data_train[['ENGINESIZE']])
train_y = np.asanyarray(data_train[['CO2EMISSIONS']])
model.fit (train_x, train_y)
# The coefficients
print ('Coefficients: ', model.coef_[0][0])
print ('Intercept   : ', model.intercept_[0])
```

```
Coefficients:  39.379834080845235
Intercept   :  125.792624890745
```

Model sudah diperoleh, maka model tersebut dimasukkan ke data testing

```
from sklearn.metrics import r2_score

test_x = np.asanyarray(data_test[['ENGINESIZE']])
test_y = np.asanyarray(data_test[['CO2EMISSIONS']])
test_y_ = model.predict(test_x)

print("MAE      : %.2f" % np.mean(np.absolute(test_y_ - test_y)))
print("MSE      : %.2f" % np.mean((test_y_ - test_y) ** 2))
print("R2-score : %.2f" % r2_score(test_y_ , test_y) )
```

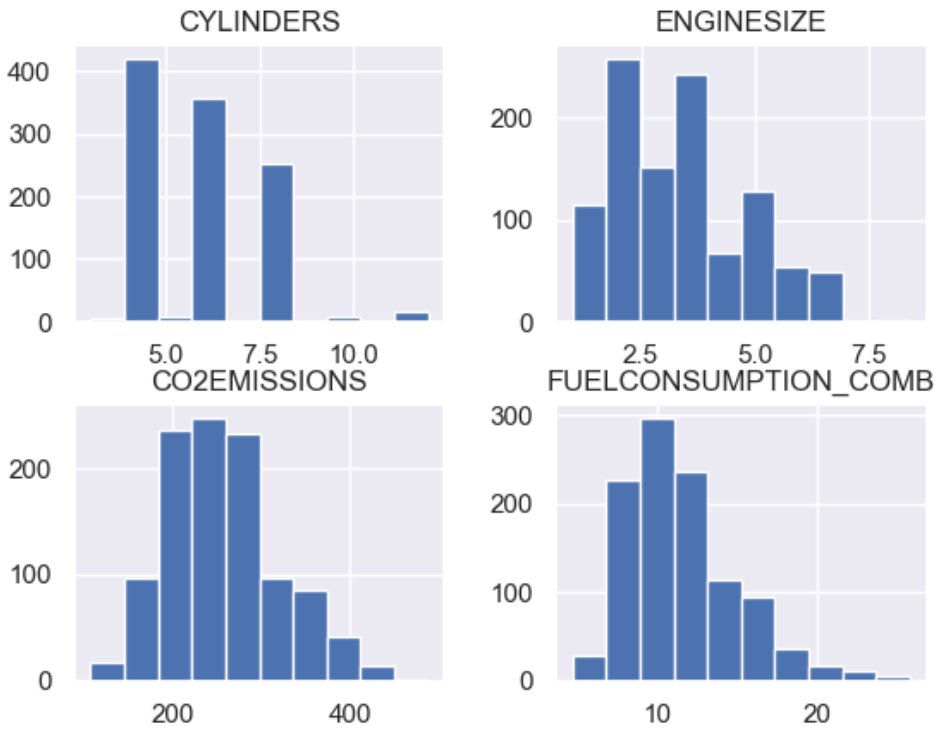
```
MAE      : 23.79
MSE      : 919.70
R2-score : 0.69
```

D. Visualisasi Data

Setelah melakukan atau sebelum melakukan pengolahan data, visualisasi data sangatlah penting sebelum pengambilan keputusan. Visualisasi data menggunakan python sangatlah mudah, ada dua package yang umum digunakan dalam memvisualisasikan data yaitu matplotlib dan seaborn.

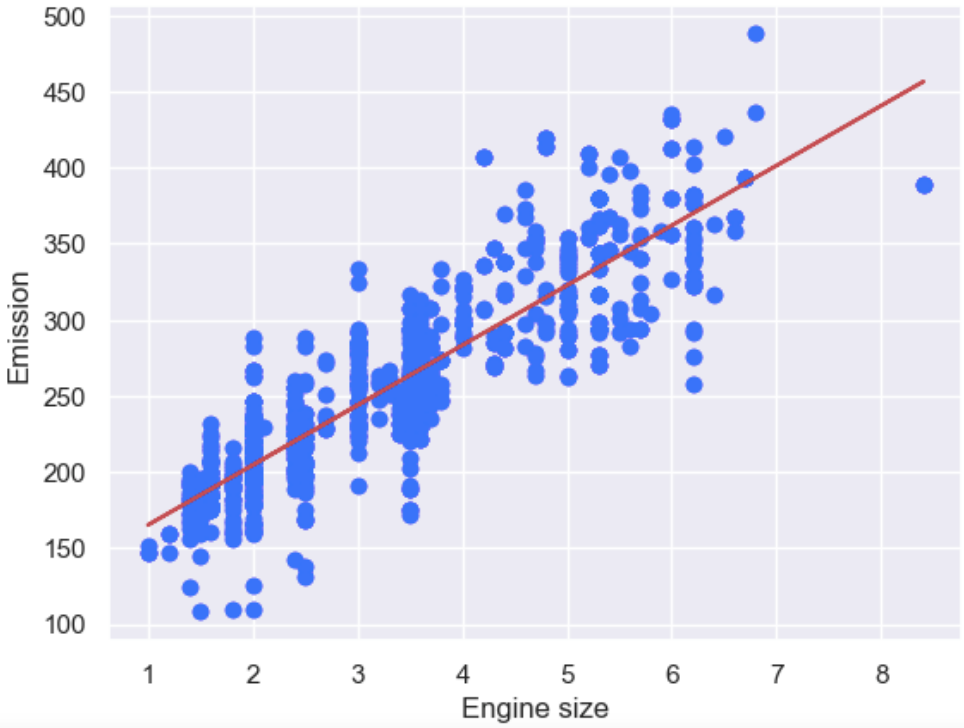
```
import matplotlib.pyplot as plt
import seaborn as sn
%matplotlib inline

sn.set_theme(style='darkgrid')
viz = data_olah[['CYLINDERS','ENGINESIZE','CO2EMISSIONS',
                 'FUELCONSUMPTION_COMB']]
viz.hist()
plt.show()
```




```
plt.scatter(data_train.ENGINESIZE,
            data_train.CO2EMISSIONS,  color='blue')
plt.plot(train_x, model.coef_[0][0]*train_x +
         model.intercept_[0], '-r')
plt.xlabel("Engine size")
plt.ylabel("Emission")
```

```
Text(0, 0.5, 'Emission')
```



TUGAS PRAKTIKUM

Carilah data, lalu lakukan pengolahan dan visualisasikan data tersebut dan berikan kesimpulan yang yang dapat diperoleh dari data tersebut.

DAFTAR PUSTAKA

Chatterjee, Marina. (2022). Top 9 Job Roles in the World of Data Science for 2022. https://www.mygreatlearning.com/blog/different-data-science-jobs-roles-industry/#5 [Daring] (Diakses 20 Januari 2022).

Doyle, Leslie. (2020). What Does a Data Scientist Do?. https://www.northeastern.edu/graduate/blog/what-does-a-data-scientist-do/ [Daring] (Diakses 20 Januari 2022).

Data-flair.training. (2019). What Role does SQL Play in Data Science – Must have Skill for Data Scientists. https://data-flair.training/blogs/sql-in-data-science/ [Daring] (Diakses 20 Januari 2022).

Glassdoor.com. (2020). What is a Data Scientist?. https://www.glassdoor.com/Job-Descriptions/Data-Scientist.htm [Daring]. (Diakses pada 27 Januari 2021).

Mastersindatascience.org. (2020). What is a Data Scientist. https://www.mastersindatascience.org/careers/data-scientist/ [Daring]. (Diakses pada 1 Maret 2021).

Medium.com. (2019). Statistik Dasar untuk Data Scientist — Part 1. https://medium.com/purwadhikaconnect/statistik-dasar-untuk-data-scientist-part-1-af7ee6999911/ [Daring]. (Diakses pada 1 Maret 2021).

https://blog.skillacademy.com/apa-itu-data-scientist

https://www.dqlab.id/

https://www.geeksforgeeks.org/python-programming-language/

https://www.w3schools.com/python/

https://www.askpython.com/python/examples/pull-data-from-an-api