Bahan Ajar

DASAR PEMROGRAMAN BASIC PYTHON

Interface: Google Colaboratory

Penulis

Christin Erniati Panjaitan, Sri Wahyuni Tarigan,
Dini M Hutagalung, Olnes Yosefa Hutajulu,
Muhammad Dominique Mendoza

Dasar Pemrograman Basic Python
Interface Google Colaboratory

PENULIS

CHRISTIN ERNIATI PANJAITAN
SRI WAHYUNI TARIGAN
DINI M HUTAGALUNG
OLNES YOSEFA HUTAJULU
MUHAMMAD DOMINIQUE MENDOZA

PENERBIT

UNPRI PRESS
ANGGOTA IKAPI

UNIVERSITAS
PRIMA INDONESIA
UNPRI PRESS

Dasar Pemrograman Basic Python
Interface Google Colaboratory

Penulis : ***CHRISTIN ERNIATI PANJAITAN***
SRI WAHYUNI TARIGAN
DINI M HUTAGALUNG
OLNES YOSEFA HUTAJULU
MUHAMMAD DOMINIQUE MENDOZA

Penerbit :
UNPRI PRESS
ISBN: 978-623-8299-28-7
(ANGGOTA IKAPI)

Alamat Redaksi
Kampus 2
Jl. Sampul No. 4 Medan

Kata Pengantar

Pemrograman Python sangat dibutuhkan saat ini karena merupakan Bahasa yang akan dipergunakan untuk Machine Learning, Deep Learning dan Artificial Intelligence. Pemrograman Python melalui Google Colaboratory merupakan ilmu yang sangat menyenangkan untuk dipelajari karena anda tidak perlu menginstall interface tersendiri di komputer. Jadi anda bisa tetap belajar dengan menggunakan gadget seperti tab, smartphone dan laptop asalkan gadget tersebut tetap terkoneksi ke internet.

Buku ini membahas tentang dasar-dasar pemrograman Python di Google Colaboratory. Dimulai dengan settingan awal di google kemudian dilanjutkan dengan memulai kodingan awal ('Hello World'). Langkah selanjutnya adalah mengenal variable yakni teks atau string dan angka baik integer ataupun float. Adapun struktur data ada beberapa tipe yakni List yang mutable atau dapat diubah. Isi dalam list dapat ditambahkan, dihapus dan diubah dan disortir. Ada juga tipe data dictionaries yang dapat menampung banyak informasi seperti biodata mahasiswa yakni nama, gender, usia, nomor telepon, dan email. Lalu ada Tupple yang bersifat immutable yakni tidak dapat diubah-ubah dan set yang mutable. Dalam kondisi bersyarat ada beberapa hal yang perlu dipelajari dimulai dari kondisi Boolean (1 dan 0) dan dengan pemahaman Boolean dapat lebih mudah memahami fungsi logika yakni AND, OR dan NOT. Kemudian dilanjutkan dengan 1 kondisi yakni penggunaan if dan 2 kondisi pilihan dengan if-else. Lalu kondisi pilihan ini naik ke Tingkat yang lebih tinggi dengan menawarkan banyak pilihan atau sering disebut dengan Chained Condition. Salah satu aplikasi Chained Condition yakni penentuan grade dari nilai akademik. Ada juga Nested Condition yang disebut dengan kondisi bersarang yang dimana dalam satu kondisi ada beberapa kondisi lagi yang ditawarkan di dalamnya. Lalu ada while loops yang memastikan program berjalan saat suatu kondisi dipenuhi dan ada gabungan break+Continue dan for+continue. Setiap variable akan disimpan dalam memori yang berbeda selagi informasinya berbeda tapi jikalau informasinya sama maka alamat penyimpanannya akan sama karena alamat penyimpanan memandang kepada isi bukan ke penamaan variabel. Kemudian untuk membangun suatu program yang besar dan comples maka diperlukan pemahaman tentang class dan object sehingga program yang dibangun akan disesuaikan dengan fungsinya. Yang paling penting juga adalah mengimplementasikan kondisi-kondisi dengan

beberapa pilihan, kemudian bekerja dengan file .Txt dan CSV dan merepresentasikan data baik line, scatter dan bar chart. Terakhir kemampuan yang akan sering digunakan di dunia akadamik adalah teknik merepresentasikan data. Pada Python, library untuk representasi datanya sudah sangat lengkap dan anda bisa mempresentasikan data dengan begitu menarik. Beberapa fitur untuk mempertebal/mempertipis ukuran grafik, menggantik warna, dan beberapa fitur untuk marker(penanda). Representasi data yang dibahas di buku ini ada tiga yakni line graph, scatter plot dan bar chart yang dimana ketiga visualisasi data akan sering dipergunakan di kegiatan akademik baik penulisan laporan dan artikel ilmiah.

Mempelajari Python di Google Colaboratory akan sangat membantu anda berlatih sekaligus memiliki catatan pribadi tentang apa yang anda pelajari. Catatan dan program yang anda bangun akan otomatis tersimpan di akun anda. Sehingga anda dapat mengakses catatan dan program anda dimanapun anda berada. Dengan mempelajari dasar-dasar Python pada Google Colaboratory tidak tertutup kemungkinan anda dapat menggunakan aplikasi offline untuk menggunakan Python seperti Pycharm, Anaconda, dll. Dengan memahami pola pemrograman di Python maka secara tidak langsung anda akan dibekali untuk memahami Bahasa pemrograman lainnya.

Salam Hangat,

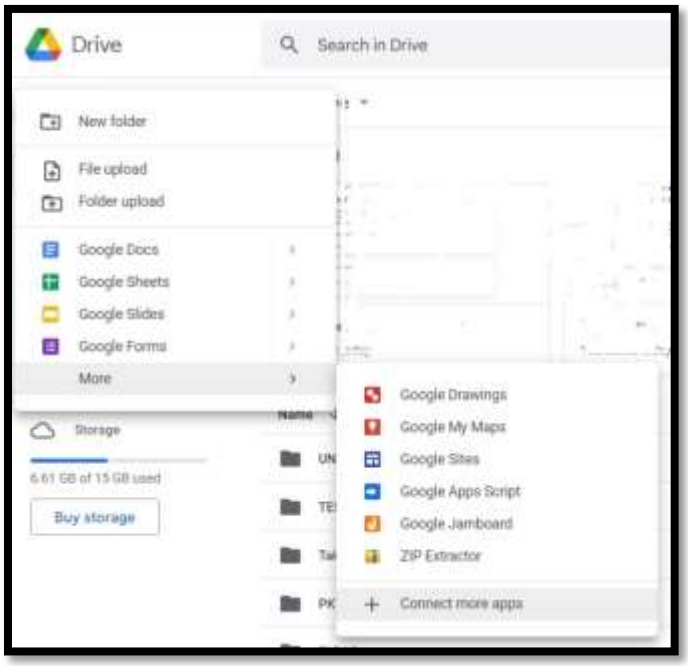
Penulis

Daftar Isi

Bab 1 Pendahuluan

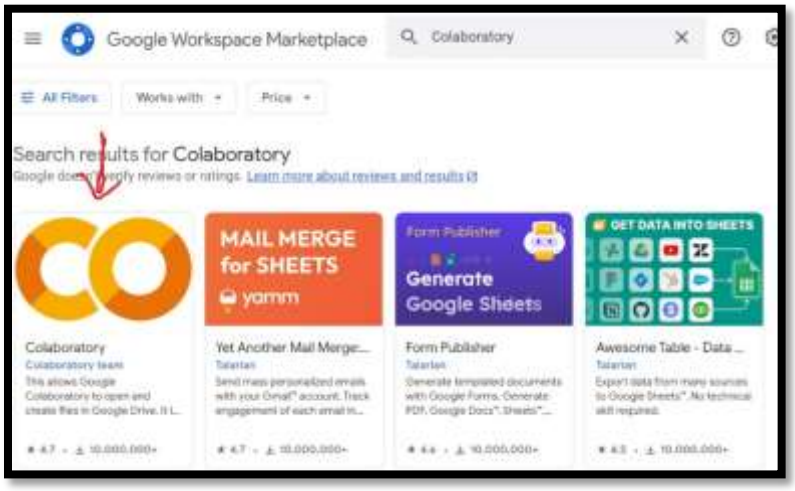
a) Instalasi Google Colaboratory

Google Colaboratory akan menjadi tempat untuk menjalankan program python. Google colab merupakan platform untuk mengeksekusi program python melalui web browser sehingga lebih fleksibel dan tanpa biaya. Adapun beberapa settingan awal di akun gmail yang perlu diperhatikan adalah sebagai berikut:



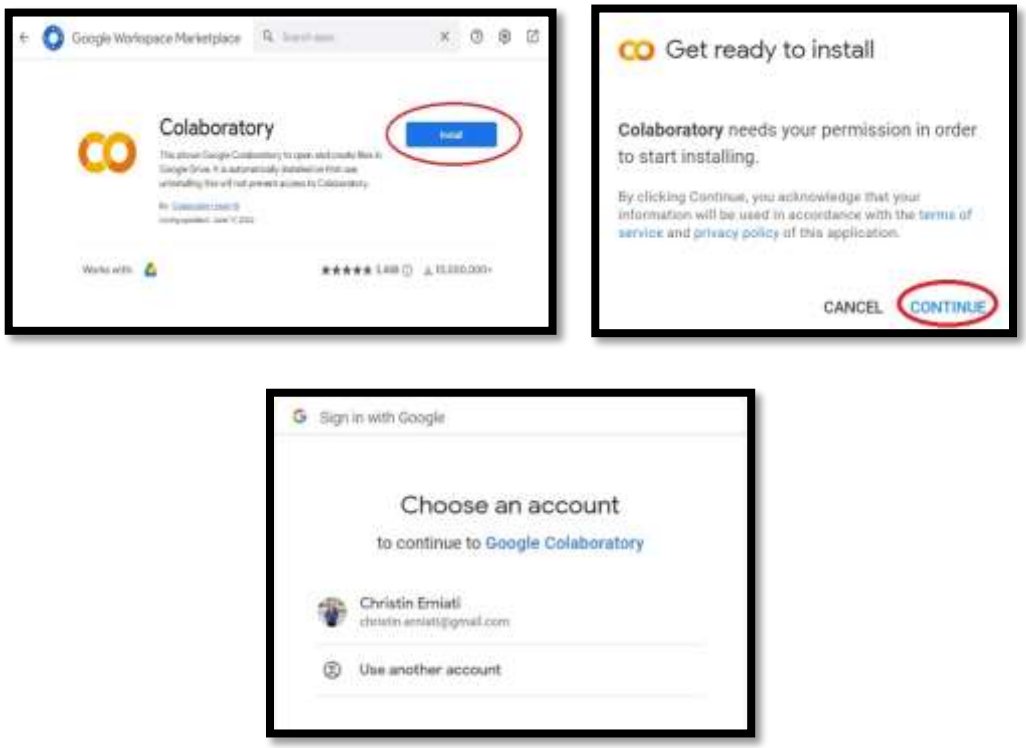
Gambar 1. Tampilan Gdrive

Sebelumnya anda harus memiliki akun gmail terlebih dahulu. Gambar 1 merupakan tampilan dari gdrive yang bisa anda akses dari gmail. Dari gambar 1 terlebih kalau google colab belum terinstal di akun, maka klik *connect more apps*.



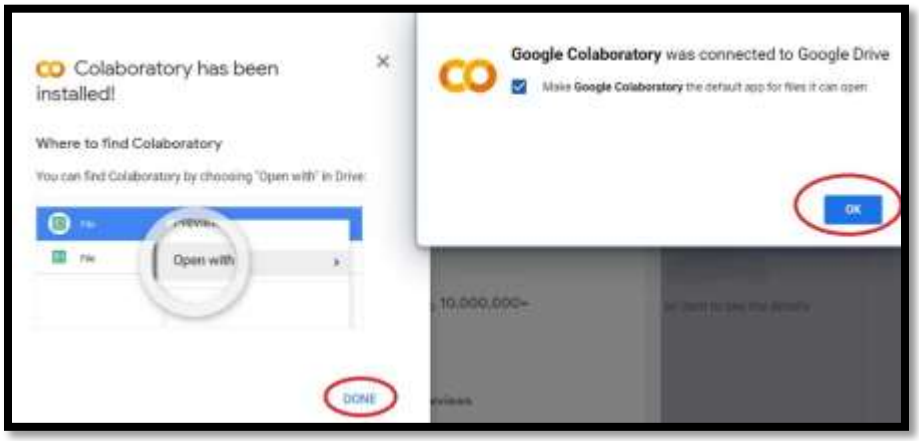
Gambar 2. Google Workspace Marketplace

Kemudian tampilan selanjutnya adalah seperti yang tampak di gambar 2 yakni *Google Workspace Marketplace*. Maka ada beberapa tawaran program dari google. Maka silahkan dipilih *Google Colaboratory.*



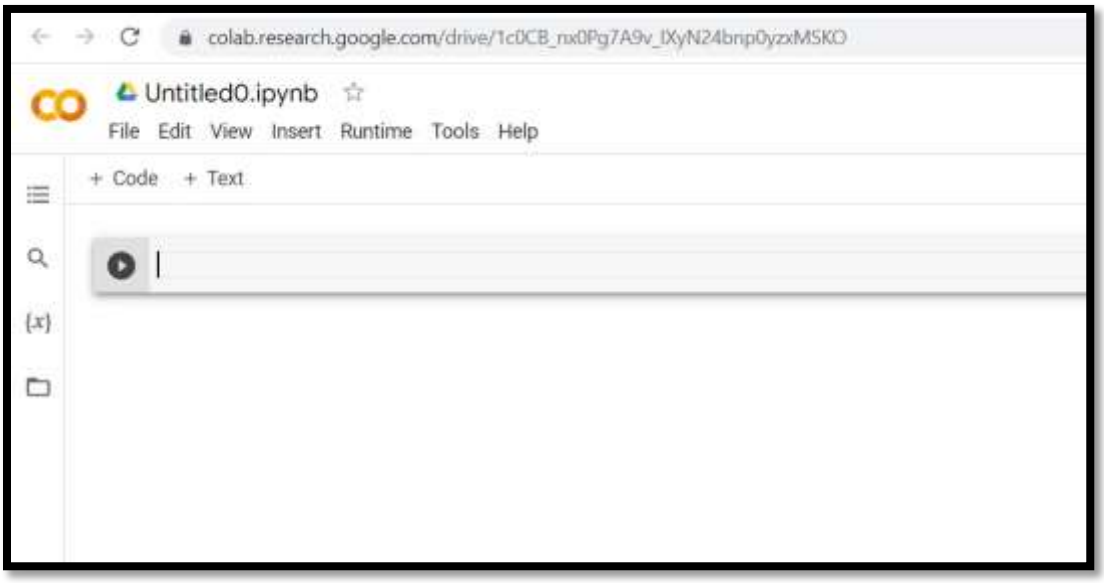
Gambar 3. Instalasi Google Colaboratory

Lalu klik *install* seperti yang tampak pada gambar 3 untuk menambahkan google colaboratory ke akun gmail dan silahkan diarahkan ke akun gmail yang dituju.



Gambar 4. Menyelesaikan instalasi Google Colaboratory

Setelah diarahkan ke akun gmail, maka klik *done* dan *ok* untuk menyelesaikan proses instalasi seperti yang ada di gambar 4.



Gambar 5. Tampilan awal

Setelah proses instalasi dilakukan maka akan keluar tampilan seperti di gambar 5. Maka, dengan demikian setiap perangkat elektronik yang terhubung ke internet baik laptop, dan tab, kita bisa mengeksekusi program python.

b) Awal Pemrograman Python di Google Colab

Hal yang lumrah untuk memulai menjalankan program di awal maka kita perlu menjalankan *Hello World*.

Gambar 6. Tampilan *hello world*

Di google colab ada 2 menu di atas, yang sebelah kiri untuk kodingan dan sebelah kanan untuk text seperti yang tampak di gambar 6. Kodingan dan text dapat kita embed, sehingga kelaknya bisa memiliki dokumentasi tersendiri untuk program python. Pada menu code, nomor 1 mengindikasi tempat untuk meletakkan program dan nomor 2 merupakan tempat menampilkan hasil dari program.

Gambar 7. Tampilan *text*

Gambar 7 menampilkan menu text, yang dimana user bisa meletakkan text dan text tersebut dapat diedit sesuai menu yang tersedia digoogle colab.

Gambar 8. Tampilan gabungan text dan code

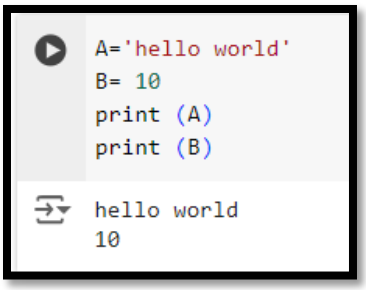
Gambar 8 menampilkan gabungan text dan code. Sehingga kelaknya, user bisa memberikan catatan sebelum atau sesudah program.

Bab 2 Variable & Data

a) Variables

Variable merupakan tempat menyimpan data baik string dan angka seperti pada gambar 9 di bawah ini. Adapun beberapa persyaratan dalam pembuatan nama variable adalah sebagai berikut:

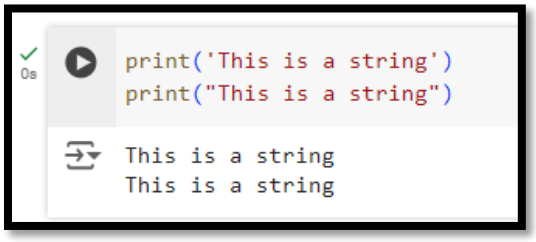
a) Penamaan variable hanya dapat mengandung huruf, angka dan garis Bawah (underscore). Contoh: Data, Data_01
b) Jarak (space) tidak diperboleh dalam pembuatan variable.
c) Hindari menggunakan kata-kata yang sering diakses Python, seperti: *False, None, True, And, As, Assert, Class, Continue, Def, Del, Elif, Else, Except, Finally, For, From, Global, If, Import, In, Is, Lambda, Nonlocal, not, or, pass, raise, return, try, while, with, yield.*
d) Penamaan variable cukup singkat dan dapat mendeskripsikan data di dalamnya.
e) Perlu diperhatikan saat menggunakan huruf kecil (l) dan huruf besar (O), karena sering disalah artikan dengan angka 1 dan 0.



Gambar 9. Variable A & B

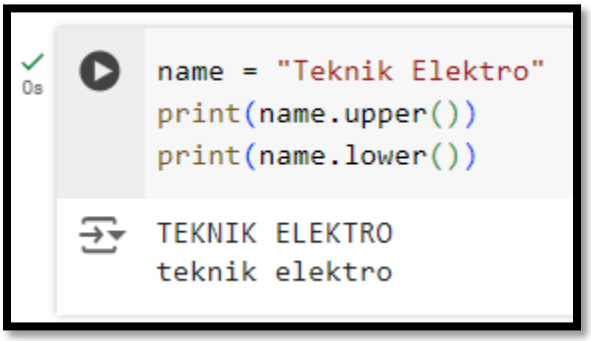
b) Teks (*Strings*)

Segala sesuatu yang masuk ke dalam tanda petik ('), ("") akan didefinisikan sebagai string seperti yang tampak pada gambar 10.



Gambar 10. String

Salah hal yang paling mudah untuk melakukan variasi pada string adalah dengan mengganti input menjadi HURUF BESAR atau huruf kecil seperti yang tampak pada gambar 11.

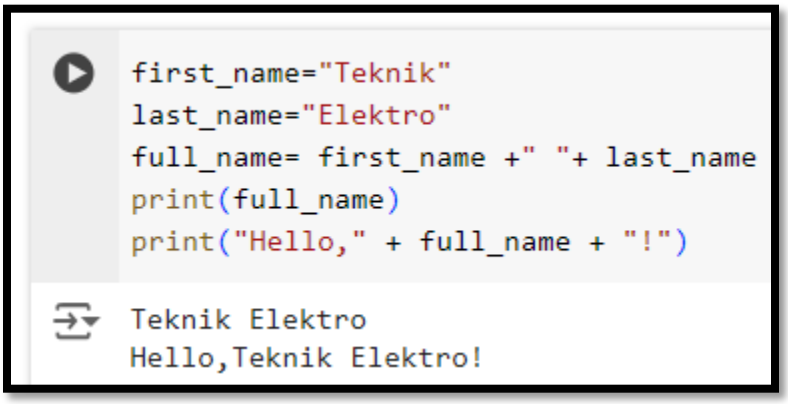


Gambar 11. Huruf Besar & Kecil pada string

Contoh pada gambar 11, dimana string disimpan dalam variable name. Tanda titik (.) setelah name.upper(), memberitahukan Python untuk mengeksekusi upper() di variable name begitu juga dengan name.lower(). Setiap metode di string harus diikuti dengan tanda kurung ().

Menggabungkan String

String yang berada dalam variable yang berbeda dapat digabungkan seperti pada gambar 12 di bawah ini.



Gambar 12 Gabungan dua string pada variable berbeda

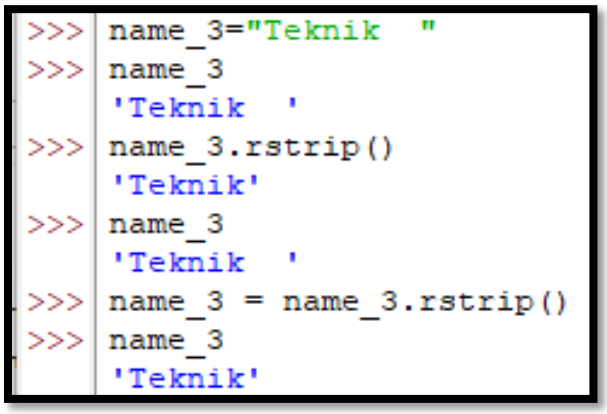
Untuk menggabungkan kedua string tersebut di *full_name* maka diperlukan (“ “) yang bertujuan memberikan jarak (*space*) antara string.

Menambahkan Whitespace

Di dalam pemrograman, whitespace mengarah ke karakter yang tidak kelihatan seperti Jarak antar kata (spaces), mengarahkan awal kalimat paragraph menjorok ke dalam (tabs), dan akhir symbol (end-of-line symbols). Untuk menambahkan tab, maka menggunakan \t dan untuk menambahkan baris baru di string maka gunakan kombinasi karakter \n seperti yang tampak pada 13.

Gambar 13. Menambahkan tab (\t) dan baris baru (\n)

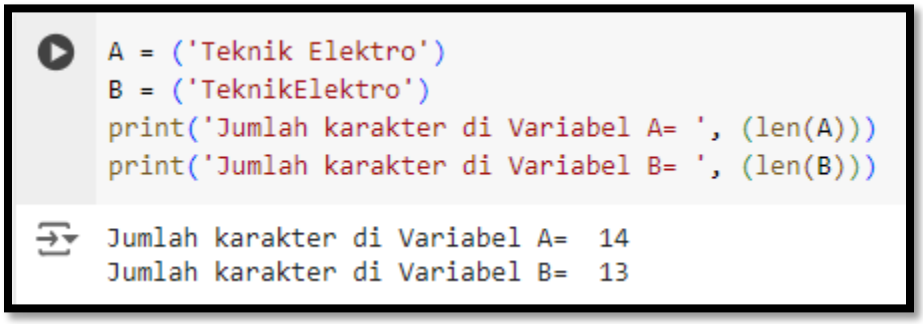
Di string seringkali ada whitespace yang terhiraukan dan bisa menganggu operasi yang lain. Maka untuk menghilangkan ekstra whitespace tersebut dapat menggunakan *variable.rstrip()* seperti yang tampak pada gambar 14.



Gambar 14. Menghilangkan ekstra whitespace

Menghitung Jumlah Karakter (len())

Fungsing len() dapat digunakan untuk menghitung jumlah karakter di dalam variable seperti tampak pada gambar 15 di bawah ini.

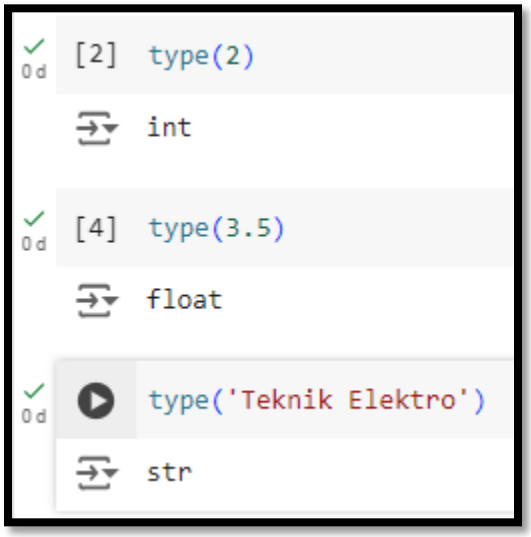


Gambar 15. Menggunakan fungsi len()

Dari gambar 15 ada dua variable yang memiliki informasi yang sama hanya saja variable A memiliki spasi antar kata sedangkan variable B tidak memiliki spasi. Sehingga saat menggunakan fungsi len(), maka spasi tetap dihitung sebagai karakter. Jumlah karakter di variable A berjumlah 14 dan di variable B berjumlah 13.

Melihat Jenis Tipe Data

Di program Python, jenis tipe data dapa dilihat seperti tampak pada gambar 16 di bawah ini.



Gambar 16. Melihat jenis data

c) Angka

Melalui angka (*numbers*) maka operasi matematika sederhana dan kompleks dapat dilakukan. Python dapat melakukan simulasi dari operasi tersebut. Tabel 1 merupakan operasi matematika dari order yang tertinggi.

Tabel 1. Operasi Matematika berdasarkan order tertinggi

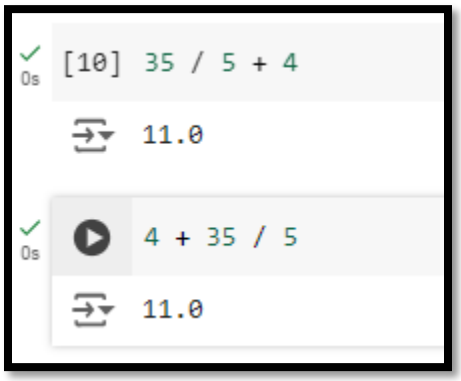
Operasi	Penjelasan	Contoh
**	Pangkat (Exponent)	3**2 = 9
%	Sisa Bagi (Modulus / remainder)	21%9 = 3
//	Pembagian (integer division)	21//9 = 2
/	Pembagian (float division)	21/9 = 2.3333
*	Perkalian (Multiplication)	2*3 = 6
-	Pengurangan (subtraction)	6-4 = 2
+	Penjumlahan (addition)	6+4 = 10

```
A = 3**2 # Exponent
B = 21%9 # Modulus (remainder)
C = 21//9 # Integer Division
D = 21/9 # Float Division
E = 2*3 # Multiplication
F = 6-4 # Subtraction
G = 6 + 4 # Addition
print (A)
print (B)
print (C)
print (D)
print (E)
print (F)
print (G)
```

```
9
3
2
2.3333333333333335
6
2
10
```

Gambar 17. Tampilan operasi dasar matematika di Python

Gambar 17 merupakan tampilan eksekusi operasi dasar matematika di Python. Gambar 18 menunjukkan urutan operator matematika yang didahulukan berdasar table 1. Jikalau kita lihat dari gambar 18 maka perhitungan tersebut tidak menggunakan tanda kurung, sehingga operator yang didahulukan berdasarkan ordernya.



Gambar 18 Hasil operasi matematika sederhana di Python

Memberikan Komentar (#)

Anda dapat juga memberikan koment tambahan di dalam program dengan menambahkan (#) seperti yang tampak pada gambar 17.

Integer dan Floating

Dari definisi matematika, maka bilangan dimulai dari *natural number* yang merupakan angka paling sederhana. Natural number ini dimulai dari 1 dan seterusnya. Kemudian kalau ditambahkan 0 dan bilangan negatif maka bilangan tersebut sudah didefinisikan dengan *integer*. Dan jikalau bilangan tersebut dapat dituliskan dalam formula *m/n* maka bilangan tersebut dikategorikan float. Penjelasan di atas sudah dirangkum pada table 2.

Table 2 Definisi Bilangan

Nama Bilangan	Penjelasan
Natural Number	1, 2, 3, 4, …
Integer Number	…, -3, -2, -1, 0, 1, 2, 3, …
Float Number	Rational: 3/7 = 0.428571428571428571

	Irrational: $\sqrt{2}$ = 1.42142135

Gambar 19 menampilkan pembagian bilangan integer dan floating. Pembagian floating akan menampilkan angka sesuai hasilnya sedangkan pembagian integer hanya menampilkan bilangan bulat saja.

```
A = 35 // 9 # Integer Division
B = 35 / 9 # Float Division
print (A)
print (B)
```

```
3
3.888888888888889
```

Gambar 19 Bilangan Integer dan Floating

```
age = 79
message = "Happy " + str(age) + "th of Independence Day !"
print(message)
```

```
Happy 79th of Independence Day !
```

Gambar 20 Gabungan string dan non-string

Gambar 20 menampilkan gabungan antara string dan bilangan integer. Variable yang mengandung bilangan integer dapat digabungan ke dalam variable yang mengandung string dengan menambahkan *str(variable(non-string))*. \

Bab 3 Struktur Data

a) List

List merupakan suatu nilai yang mengandung beragam informasi dalam urutan. Suatu list dimulai dengan [dan diakhiri dengan]. Informasi yang berada di dalam list merupakan *items*. Item satu dan yang lain dipisahkan dengan koma (,). Gambar 21 merupakan penggunaan list, dimana list digunakan dalam kelompok data yang sama baik angka ataupun string atau bisa gabungan.

Gambar 21. Penggunaan list

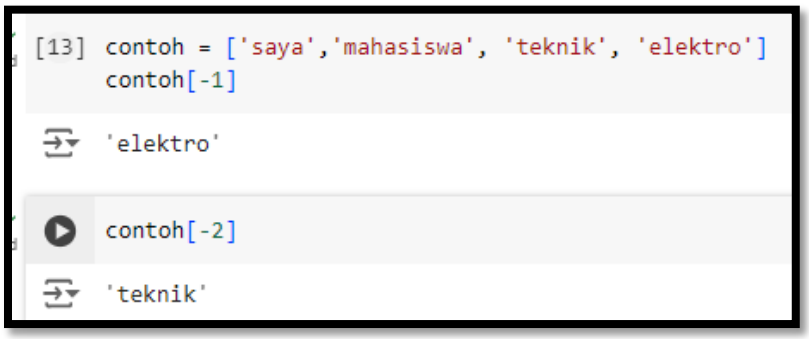
Indeks

Nilai – nilai yang tersimpan di dalam variable memiliki order yang disebut dengan *indeks* Adapun indeks dimulai dari 0 dan posisi sebelah kiri. Dari gambar 22, terdapat variable *contoh* yang berikan informasi string. Maka dari 4 kata di dalam satu variable, contoh[0] adalah saya, contoh[1] adalah mahasiswa dan seterusnya.

Gambar 22. Urutan indeks pada suatu list

Negatif Indeks

Dari penjelasan sebelum, indeks dapat dimulai dari awal informasi di dalam suatu list, maka negatif list memulai informasi dari paling akhir dan dimulai dengan variable(-1). Dari gambar 23, dengan variable yang sama yakni *contoh* maka negatif indeks dimulai dari akhir data, yaknik contoh[-1] adalah elektro, contoh[-2] adalah teknik dan seterusnya.



Gambar 23. Negatif Indeks

Sublist

Beberapa informasi dapat diambil dari list dan proses ini disebut dengan *sublist*. Proses sublist ada pada gambar 24. Dalam suatu variable, ada tiga proses sublist dilakukan. Sublist pertama dan kedua dimulai dari indeks awal hingga indek akhir dimana satu indeks ditambahkan. Sublist ketiga gabungan dari positif dan negative indeks yang dimulai dari 0 dan diakhir dengan indeks negative.

```
A = ['Saya', 'Mahasiswa', 'Teknik', 'Elektro']
print(A[0:4])
print(A[0:3])
print(A[0:-2])
```

```
['Saya', 'Mahasiswa', 'Teknik', 'Elektro']
['Saya', 'Mahasiswa', 'Teknik']
['Saya', 'Mahasiswa']
```

Gambar 24. Sublist

Mengganti Nilai di List

Nilai tertentu pada list dapat diupdate, sehingga variable yang membawa list tersebut akan ikut terupdate juga proses ini dapat dilihat pada gambar 25.

```
A = ['Saya', 'Mahasiswa', 'Teknik', 'Elektro']
A[3] = 'Arsitektur'
print(A)
```

```
['Saya', 'Mahasiswa', 'Teknik', 'Arsitektur']
```

Gambar 25. Update string di list

Menambahkan element (*Append & Insert*)

Selain mengganti nilai di dalam list, elemen di list dapat ditambahkan dengan yang baru seperti yang tampak pada gambar 26 di bawah ini. Fungsi *append* akan meletakkan informasi tersebut paling akhir.

```
A = ['Saya', 'Mahasiswa', 'Teknik', 'Elektro']
A.append('UNPRI')
print(A)
```

```
['Saya', 'Mahasiswa', 'Teknik', 'Elektro', 'UNPRI']
```

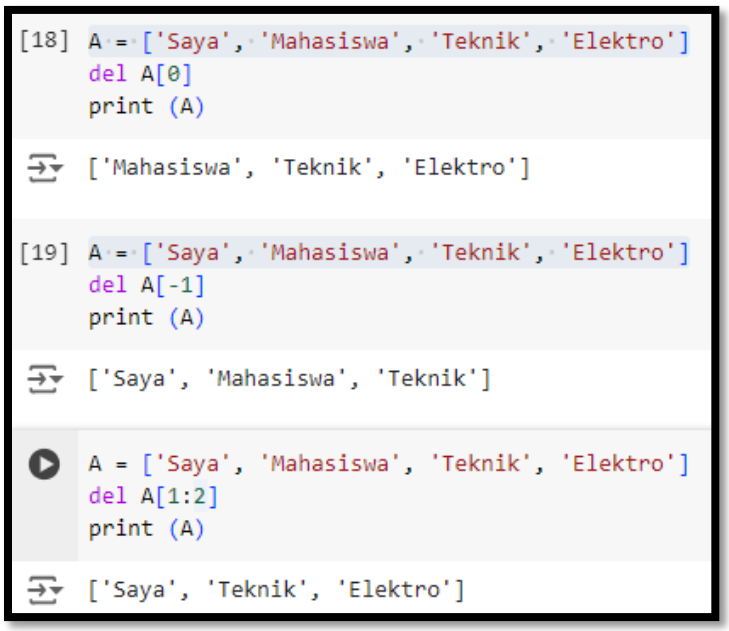
Gambar 26. Append

Penambahan informasi di list juga dapat ditambahkan sesuai urutan indeks. Informasi tambahan akan diletakkan sebelum indek tujuan seperti yang tampak pada gambar 27.

Gambar 27. Menambahkan elemen baru sesuai urutan indeks

Menghapus element (*del, pop, remove*)

Element dalam list dapat dihapus sesuai dengan indeks mana yang mau dihilangkan. Dari gambar 28, element list dapat dihilangkan baik satu element atau beberapa element. Dengan menggunakan fungsi *del* maka tinggal diletakkan urutan indeksnya.



Gambar 28. Menghapus element menggunakan list

Fungsi *pop* juga dapat digunakan untuk menghapus element pada list. Hanya saja, fungsi *pop* akan langsung menghapus element paling akhir seperti yang tampak pada gambar 29. Variable C berisikan 4 element string, setelah menggunakan fungsi *pop* maka element akhir hilang,

```
C = ['Saya', 'Mahasiswa', 'Teknik', 'Elektro']
print(C) #Tampilan utuh
C2 = C.pop()
print (C2) #element akhir dibuang
print (C) #Tampilan setelah ada element yang dibuang
```

```
['Saya', 'Mahasiswa', 'Teknik', 'Elektro']
Elektro
['Saya', 'Mahasiswa', 'Teknik']
```

Gambar 29. Menghapus element menggunakan pop

Element pada list juga dapat dihapus dengan langsung menyebutkan informasi yang mau dihapus seperti tampak pada gambar 30. Penggunaan fungsi *remove* dapat dilakukan dengan menyembutkan variable tujuan dan disertai dengan element yang mau ditiadakan.

```
D = ['Teknik', 'Elektro', 'Fakultas', 'Sains', 'Dan', 'Teknologi']
print(D)
D.remove('Dan')
print(D)
```

```
['Teknik', 'Elektro', 'Fakultas', 'Sains', 'Dan', 'Teknologi']
['Teknik', 'Elektro', 'Fakultas', 'Sains', 'Teknologi']
```

Gambar 25. Fungsi *Remove*

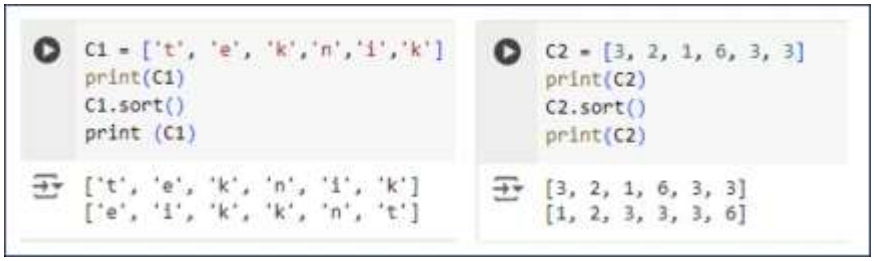
Operasi pada List

Pada gambar 31, ada dua variable yakni X dan Y, dan kedua variable tersebut digabungkan dan outputnya disimpan di variable Z. Penambahan element di list juga bisa dengan melakukan perkalian.

Gambar 31. Menggabungkan dua variable

Mengurutkan Element (*Sort*)

Gambar 32 menampilkan proses pengurutan element pada list. Sebelah kiri menampikan karakter huruf dan sebelah kanan menampilkan karakter angka.



Gambar 32. Mengurutkan element pada list

b) Dictionaries

Di *dictionaries*, informasi yang ditampung lebih beragam seperti menyimpan nama, nomor handphone, tanggal lahir dan lain-lain. Seperti yang tampak pada gambar 33, dimana data mahasiswa dapat dirincikan dengan detail di dalam dictionaries.

```
mahasiswa_01= {'nama':'Tomi', 'Gender':'Pria', 'Tahun Kelahiran': 2002 }
print(mahasiswa_01)
```

```
{'nama': 'Tomi', 'Gender': 'Pria', 'Tahun Kelahiran': 2002}
```

Gambar 33. Data mahasiswa

```
mahasiswa_01= {'nama':'Tomi', 'Gender':'Pria', 'Tahun Kelahiran': 2002 }
print(mahasiswa_01)
mahasiswa_01['tinggi_badan'] = '165 CM'
mahasiswa_01['berat_badan'] = '70 Kg'
print(mahasiswa_01)
```

```
{'nama': 'Tomi', 'Gender': 'Pria', 'Tahun Kelahiran': 2002}
{'nama': 'Tomi', 'Gender': 'Pria', 'Tahun Kelahiran': 2002, 'tinggi_badan': '165 CM', 'berat_badan': '70 Kg'}
```

Gambar 34. Penambahan element di *dictionaries*

Pada gambar 34 terlihat penambahan element di dictionaries sebelumnya. Penambahan element akan diletakkan paling akhir. Elemet yang ditambahkan dimulai dari tinggi badan dan berat badan, maka kedua element baru ini akan ditambahkan sesuai urutannya di variable mahasiswa_01.

Dictionaries Kosong (Empty Dictionaries)

Dictionaries juga dapat dimulai dari kosong dan diisi secara bertahap seperti yang tampak pada gambar 35. Variable Mahasiswa_02 masih kosong, dan kemudia diisi bertahap dengan nama dan gender.

```
[10] mahasiswa_02 = {}
     mahasiswa_02['nama'] = 'Jenifer'
     mahasiswa_02['Gender'] = 'Wanita'
     print(mahasiswa_02)
```

```
{'nama': 'Jenifer', 'Gender': 'Wanita'}
```

Gambar 35. Empty Dictionaries

Mengganti nilai di dictionaries

Adapun nilai pada *dictionaries* dapat diganti seperti yang tampak pada gambar 36. Variable mahasiswa_03 berisikan informasi nama dan usia. Kemudia ditahapan selanjutkan usianya diganti dari 18 menjadi 19 tahun.

```
mahasiswa_03 = {'nama' : 'Budi', 'usia' : '18 tahun'}
print(mahasiswa_03)
mahasiswa_03['usia'] = '19 tahun'
print(mahasiswa_03)
```

```
{'nama': 'Budi', 'usia': '18 tahun'}
{'nama': 'Budi', 'usia': '19 tahun'}
```

Gambar 36. Mengganti nilai

Menghapus elemet pada dictionaries

Elemet pada dictionaries dapat dihapus seperti yang tampak pada gambar 37. Variable mahasiswa_04 memiliki 3 informasi dan salah satu informasinya akan dihapus yakni gender. Sehingga Ketika variable tersebut ditampilkan Kembali sudah tidak memiliki element gender.

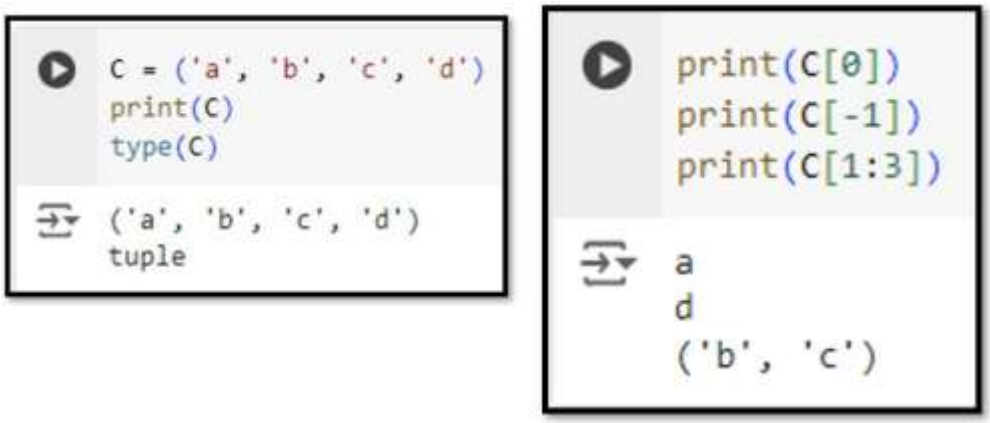
```
mahasiswa_04 = {'nama' : 'Joko', 'usia' : '18 tahun', 'gender' : 'Pria'}
print(mahasiswa_04)
del mahasiswa_04['gender']
print(mahasiswa_04)
```

```
{'nama': 'Joko', 'usia': '18 tahun', 'gender': 'Pria'}
{'nama': 'Joko', 'usia': '18 tahun'}
```

Gambar 37. Menghapus element

c) Tupples

Pada tupples, informasi juga bisa diinput baik string dan angka, hanya saja informasi yang sudah diinput ke tupple tidak bisa diubah (*immutable*). Gambar 38 menampilkan variable C yang berisikan string dengan fungsi tupple.



Gambar 38. Tupple

Untuk melihat order pada tupple dapat juga dilakukan dengan cara yang sama dengan list seperti yang tampak pada gambar 38. Indeks 0 dimulai dari element paling kiri, sedangkan indeks -1 dimulai dari paling kanan.

d) Set

Set tidak bisa menyimpan informasi secara berurut dan elemen di dalamnya tidak dapat diganti tapi dapat dihilangkan seperti yang tampak pada gambar 39 di bawah ini.

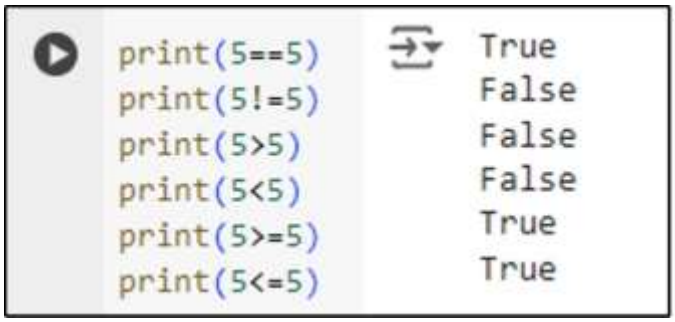
Gambar 39. Contoh variable set

Bab 4 Kondisi Bersyarat (Conditional Statement)

Dalam menjalankan suatu program maka tidak mungkin kondisi yang dirancang akan berjalan dalam satu jalan (one way). Ada kemungkinan program memiliki kondisi pilihan atau yang disebut dengan conditional statement. Adapun kondisi pilihan ini dalam konsep sederhana dapat dilihat, semisalnya si Budi merencanakan akan menonton di Bioskop nanti sore, hanya saja cuaca kurang mendukung, maka si Budi harus tetap berada di rumah.

a) Fungsi Boolean

Kondisi Boolean memiliki ekspresi yakni TRUE or FALSE. Untuk membuat kondisinya dibutuhkan tanda (==, !=, >, <, >=, <=) yang menandakan untuk membandingkan kondisi kiri dan kanan. TRUE apabila sama dan FALSE apabila berbeda seperti yang tampak pada gambar 40.



Gambar 40. Boolean dengan berbagai kondisi

b) Fungsi Logika (*And, Or, Not)*

Kondisi logika menggunakan 3 fungsi yakni *and, or* dan *not*. Adapun ketiga fungsi ini akan digunakan untuk membandingkan kondisi kiri dan kanan atau sebelum dan sesudah.

Tabel 3 Fungsi Logika

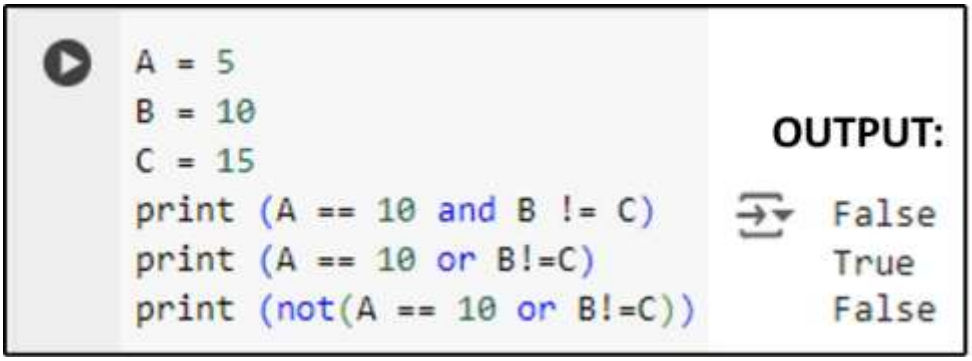
AND	RESULT
1 AND 1	TRUE
1 AND 0	FALSE
0 AND 1	FALSE
0 AND 0	TRUE

OR	RESULT
1 AND 1	TRUE
1 AND 0	TRUE
0 AND 1	TRUE
0 AND 0	FALSE

INVERTER	RESULT
1	FALSE
0	TRUE

1 = TRUE CONDITION
0 = FALSE CONDITION

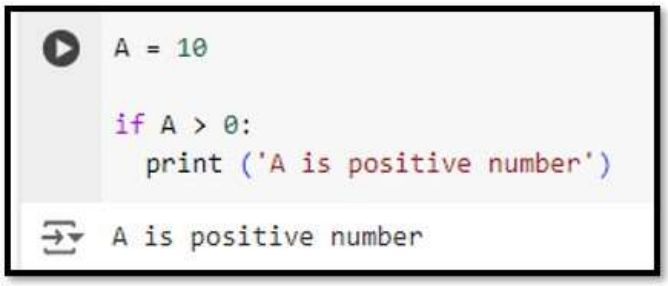
Tabel 3 menyatakan sifat dari ketiga fungsi logika. Sifat dari fungsi AND adalah TRUE jikalau kedua kondisi kira dan kanan memiliki informasi yang sama dan FALSE jikalau salah satunya berbeda. Sedangkan kondisi OR akan menghasilkan TRUE jikalau salah satu kondisi atau keduanya memiliki kondisi yang benar dan FALSE jikalau keduanya tidak mengandung informasi yang benar. Sedangkan INVERTER atau NOT akan menghasilkan kondisi yang terbalik dari inputannya. Pada gambar 41 terlihata ada 3 variable dengan nilai yang berbeda. Kemudia ketiga variable tersebut menggunakan fungsi logika. Kondisi pertama, menggunakan fungsi AND yang sebelah kiri bernilai FALSE dan sebelah kanan bernilai TRUE maka outputnya adalah FALSE. Lalu kondisi kedua, menggunakan fungsi OR yang sebelah kiri bernilai FALSE dan sebelah kanan bernilai TRUE maka outputnya adalah TRUE. Dan kondisi ketiga merupakan kebalikan dari keluaran kondisi kedua yakni FALSE.



Gambar 41 Kondisi Logika

c) Conditional Execution

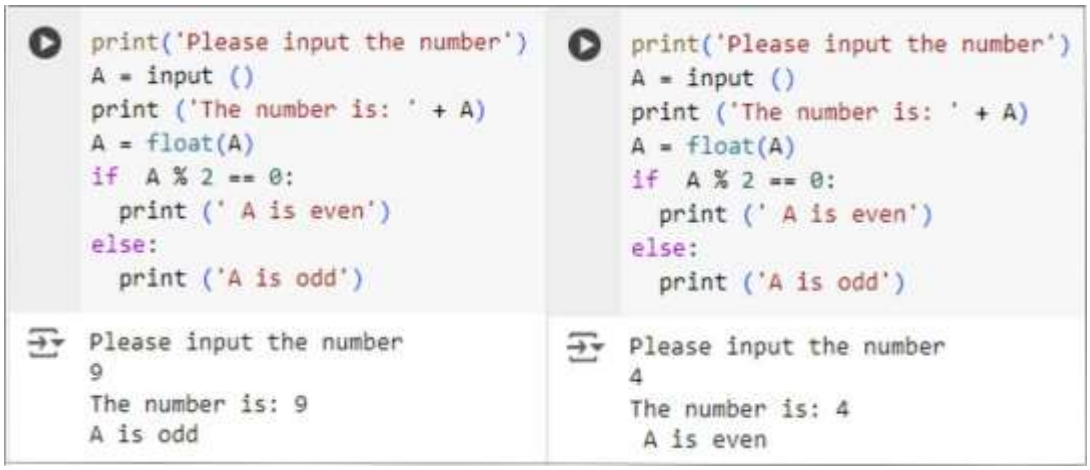
Bentuk kondisi pilihan paling sederhana adalah ***if*** yang dimana dapat dilihat seperti pada gambar 42 di bawah ini. Pada gambar 42, variable A bernilai 10, lalu bertemu dengan kondisi ***if*** yang bertuliskan jikalau isi di variable A lebih dari 0 maka variable A dikategorikan bilangan positif.



Gambar 42. If Condition

d) Alternatif Eksekusi (*If – Else*)

Kondisi pilihan kedua adalah ***if – else*** yang dimana jika kondisi pertama tidak dapat dipenuhi maka tersedia pilihan untuk menuju kondisi kedua. Gambar 43 merupakan contoh dari if-else yang berisikan seleksi angka genap dan ganjil. Sebelah kiri, variabel A berisikan nilai 9 maka variabel A dikategorikan ODD dan sebelah kanan, variabel A dikategorikan EVEN.



```
print('Please input the number')
A = input ()
print ('The number is: ' + A)
A = float(A)
if  A % 2 == 0:
  print (' A is even')
else:
  print ('A is odd')
```

```
Please input the number
9
The number is: 9
A is odd
```

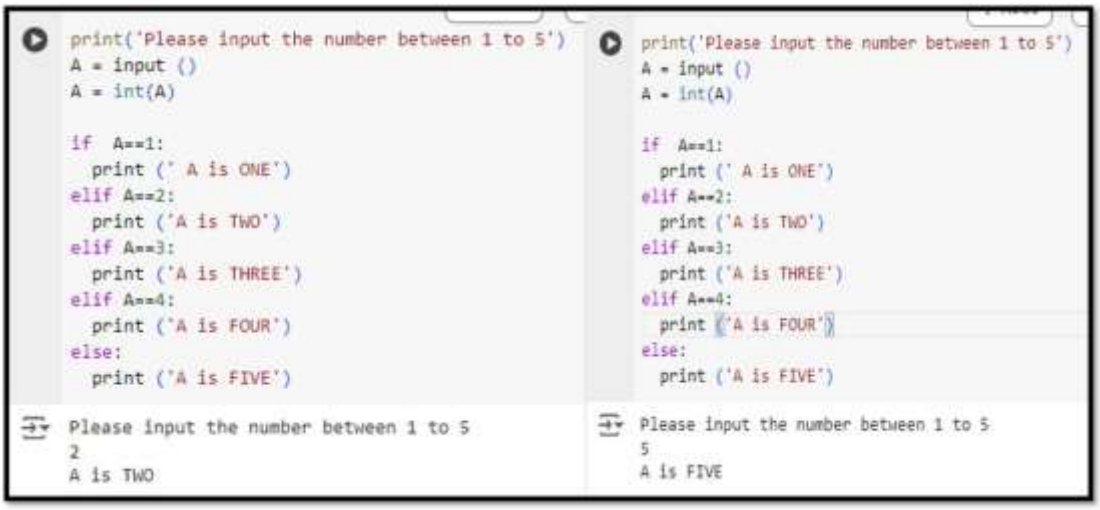
```
print('Please input the number')
A = input ()
print ('The number is: ' + A)
A = float(A)
if  A % 2 == 0:
  print (' A is even')
else:
  print ('A is odd')
```

```
Please input the number
4
The number is: 4
 A is even
```

Gambar 43. Kondisi If – Else

e) Kondisi Pilihan yang berantai (*Chained Condition, If – Elif – Else*)

Kondisi pilihan ketiga adalah If – Elif – else yang menyediakan beberapa pilihan untuk suatu kondisi. Gambar 44 merupakan contoh dari if-elif-else yang mendeskripsikan nilai variable A ke beberapa pilihan. Sebelah kiri, variabel A berisikan 2 dan didefinisikan TWO sedangkan sebelah kanan variable A berisikan 5 dan didefinisikan FIVE.



```
print('Please input the number between 1 to 5')
A = input ()
A = int(A)

if  A==1:
  print (' A is ONE')
elif A==2:
  print ('A is TWO')
elif A==3:
  print ('A is THREE')
elif A==4:
  print ('A is FOUR')
else:
  print ('A is FIVE')
```

```
Please input the number between 1 to 5
2
A is TWO
```

```
print('Please input the number between 1 to 5')
A = input ()
A = int(A)

if  A==1:
  print (' A is ONE')
elif A==2:
  print ('A is TWO')
elif A==3:
  print ('A is THREE')
elif A==4:
  print ('A is FOUR')
else:
  print ('A is FIVE')
```

```
Please input the number between 1 to 5
5
A is FIVE
```

Gambar 44. Kondisi If-Elif – Else

f) Kondisi Bersarang (*Nested Condition*)

Ada satu kondisi yang dimana terkait dengan yang lain. Gambar 45 di Bawah ini akan memperlihatkan salah satu contoh nested condition. Pada gambar 45, variabel A meminta user untuk memasukkan angka yang berupa usia, kemudia angka yang dimasukkan akan dicheck dan diklasifikasikan dewasa, memiliki pekerjaan atau sudah seharusnya pension.

Gambar 45. Nested Condition

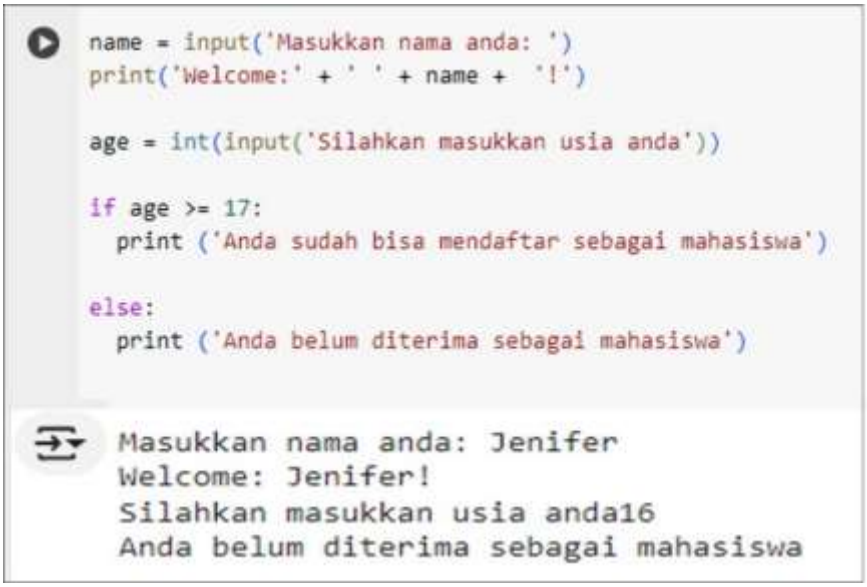
g) Klasifikasi Kondisi Dengan Operasi Logika

Ada juga kondisi yang dimana operasi logika dibutuhkan untuk membantu mendefinisikan suatu kondisi. Gambar 46 menampilkan contoh dari kondisi yang memakai operasi logika. Pada gambar 46, variabel grade akan menampung nilai daspro yang diinput, kemudian nilai tersebut akan diseleksi berdasarkan kategori A, B, C, D dan mengulang. Saat seleksi kategori maka digunakan operator AND untuk membantu mendapatkan range nilai B, C dan D.

Gambar 46. Kondisi dengan Operasi Logika

h) Kerja suatu fungsi (*Function Works*)

Jikalau suatu nilai / text akan dipakai berulang di dalam pemrograman, maka alangkah lebih baiknya nilai/text tersebut dijadikan sebuah fungsi sehingga jikalau dibutuhkan tinggal memanggil fungsi tersebut. Gambar 47 merupakan salah satu contoh fungsi teks yang dimana if else digunakan di dalam sistem yang dibangun. If-else digunakan untuk klasifikasi usia calon mahasiswa dengan ambang batas 17 tahun.



```
name = input('Masukkan nama anda: ')
print('Welcome:' + ' ' + name +  '!')

age = int(input('Silahkan masukkan usia anda'))

if age >= 17:
  print ('Anda sudah bisa mendaftar sebagai mahasiswa')

else:
  print ('Anda belum diterima sebagai mahasiswa')
```

```
Masukkan nama anda: Jenifer
Welcome: Jenifer!
Silahkan masukkan usia anda16
Anda belum diterima sebagai mahasiswa
```

Gambar 47. Function age dan name

```
angka = input('Silahkan masukkan angka: ')
angka = int(angka)

if angka % 2 == 0:
  print('Angka ' + str(angka) + ' adalah bilangan genap')

else:
  print('Angka ' + str(angka) + ' adalah bilangan ganjil')
```

Output

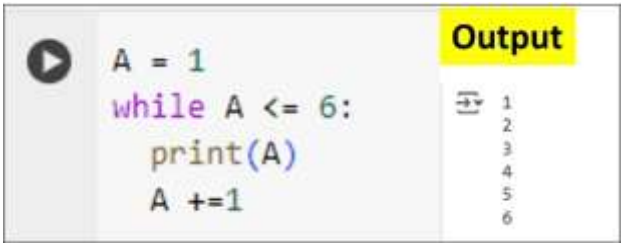
```
Silahkan masukkan angka: 4
Angka 4 adalah bilangan genap

Silahkan masukkan angka: 3
Angka 3 adalah bilangan ganjil
```

Gambar 48 Function Angka

Gambar 48 juga memiliki deskripsi yang hampir mirip dengan gambar sebelumnya, dimana dalam satu variabel akan diinput suatu data yang awal mulanya string dikonversi menjadi int. Kemudian variabel tersebut akan diklasifikasi menggunakan fungsi if-else yakni menentukan bilangan ganjil atau genap.

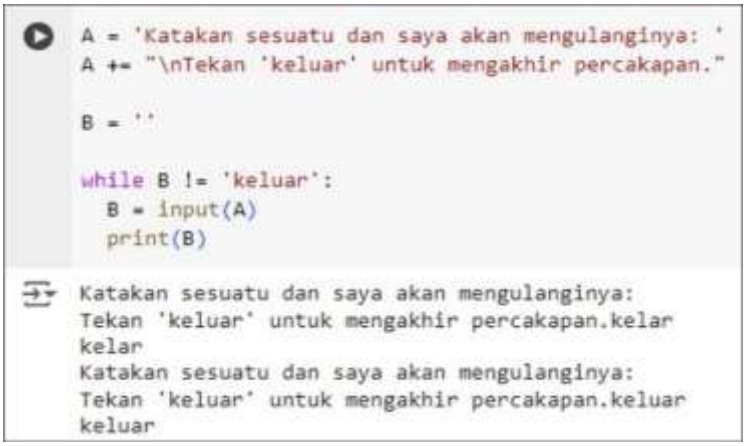
i) While Loops

Salah satu kondisi yang tetap menjalankan suatu program selama (*while*) beberapa kondisi masih TRUE. Gambar 49 merupakan contoh dari penggunaan while, yang dimana variabel A diisi dengan angka, kemudian ada kondisi A<=6 maka variabel A akan ditambah dengan 1.



Gambar 49 While

Gambar 50 juga merupakan salah satu contoh penggunaan while. Variable A akan menampung suatu kata yang dapat diulang dan variabel B akan mengikuti isian variabel A kecuali kata *keluar*. Jikalau kata *keluar* dimasukkan maka sistem akan berhenti.



Gambar 50. While kata keluar

j) Break + Continue

Break dan continue juga digunakan untuk menghentikan atau melanjutkan suatu kondisi. Gambar 51 menampilkan salah satu contoh penggunaan fungsi break yakni variabel A akan diisi oleh suatu

```
while True:
  A = int(input("Masukkan suatu angka genap: "))
  if A % 2 == 0:
    break
  print('Anda belum memasukkan angka genap')
```

```
Masukkan suatu angka genap: 5
Anda belum memasukkan angka genap
Masukkan suatu angka genap: 7
Anda belum memasukkan angka genap
Masukkan suatu angka genap: 4
```

Gambar 51. Penggunaan While – Break

angka tipe integer, kemudian angka tersebut akan diseleksi apakah bilangan genap atau bukan. Jikalau bukan bilangan genap maka user akan diminta untuk terus memasukkan angka hingga nantinya angka genap yang diinput akan menghentikan sistem.

k) For + Continue

Fungsi continue juga dapat dipakai untuk meminta suatu kondisi berpindah ke kondisi yang lain seperti yang tampak pada gambar 52 di bawah ini. Pada gambar 48 ada suatu range yang ditentukan dari 0 – 14. Kemudian, jikalau dalam range tersebut ada nilai yang di bawah 5 atau di atas 9 maka perulangan dihentikan dan yang ditampilkan selain niliai tersebut.

```
for i in range(15):
    if i < 5 or i > 9:
        continue
    print(i)
```

```
5
6
7
8
9
```

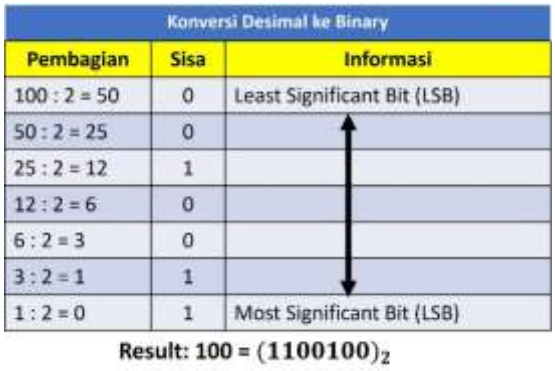
Gambar 52. Penggunaan For – Continue

Bab 5 Bilangan Pada Python

Pada bab 2 ada pembahasan tentang angka, hanya saja belum dibukakan beberapa tipe bilangan digital. Bilangan digital perlu dipelajari karena informasi analog akan dikonversikan ke bilangan digital untuk diproses di perangkat digital. Adapun 3 tipe bilangan yang akan dipelajari ke depannya adalah binary, octal dan hexadecimal.

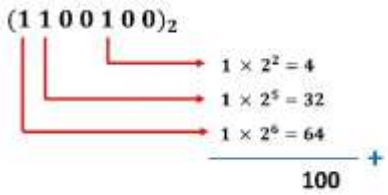
a) Binary (0b)

Bilangan decimal harus terlebih dahulu dikonversikan ke bilangan binary sebelum dikonversikan ke tipe bilangan digital lainnya. Adapun cara mengkonversikan bilangan decimal ke bilangan digital cukup dengan membagi dua seperti yang tampak pada gambar 53.



Gambar 53. Konversi bilangan digital ke bilangan binary

Pada gambar 53 setelah 100 dibagi 2 maka sisa bagi akan diurutkan dan posisi bit paling kiri ada Most Significant Bit (MSB) dan paling kanan adalah Least Significant Bit (LSB) yang menandakan bit mana yang paling berpengaruh. Kemudian untuk mengembalikan bilangan biner (binary) ke bilangan decimal maka setiap bit 1 akan dikalikan dengan 2^n yang dimana n adalah urutan bit dari paling kanan. Gambar 54 merupakan deskripsi konversi bilangan biner ke bilangan desimal.



Gambar 54. Konversi biner ke desimal

Maka pada python dapat dijalankan seperti pada gambar 55 di bawah ini. Pada gambar 55 sebelah kiri menampilkan proses konversi bilangan desimal dengan menggunakan *bin(angka desimal)* maka keluarannya berupa *0b(bilangan biner)*. Pada gambar 55 sebelah kanan menampilkan proses konversi biner ke desimal yang dimana binary 11011 merupakan 27 dalam angka desimal.

Gambar 55. Konversi Desimal – Binary di Python

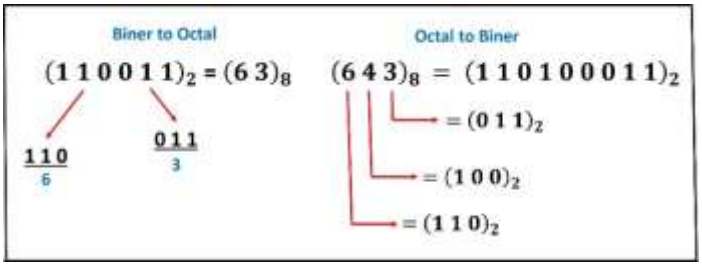
b) Octal (0o)

Bilangan biner akan lebih mudah dikonversikan ke beberapa bilangan digital dan salah satunya adalah bilangan octal. Tabel 4 merupakan penjelasan bilangan octal dari 0-7 bilangan desimal. Satu bilangan octal akan mewakili 3 bit bilangan biner.

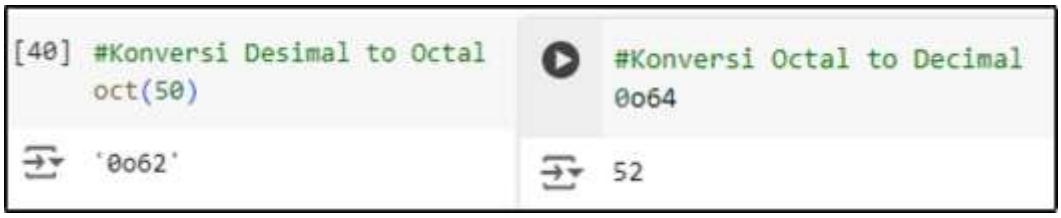
Tabel 4. Bilangan Octal

Desimal	0	1	2	3	4	5	6	7
Octal	000	001	010	011	100	101	110	111

Gambar 56 di bawah merupakan contoh konversi bilangan biner ke octal dan sebaliknya. Pada bilangan biner maka 3 bit akan diambil mulai dari LSB. Sedangkan konversi octal to biner maka semua bilangan octal harus dipecah ke bilangan biner terlebih dahulu sehingga bilangan desimal dapat dihitung seperti cara sebelumnya.



Gambar 56. Konversi Biner - Octal



Gambar 57 Konversi Desimal – Octal di Python

Gambar 57 merupakan proses konversi bilangan desimal ke python dan sebaliknya. Gambar sebelah kiri menggunakan *oct(bilangan desimal)* untuk melakukan konversi desimal ke octal dan gambar sebelah kanan menggunakan *0o(bilangan octal)* untuk konversi octal ke desimal.

c) Hexadecimal (0x)

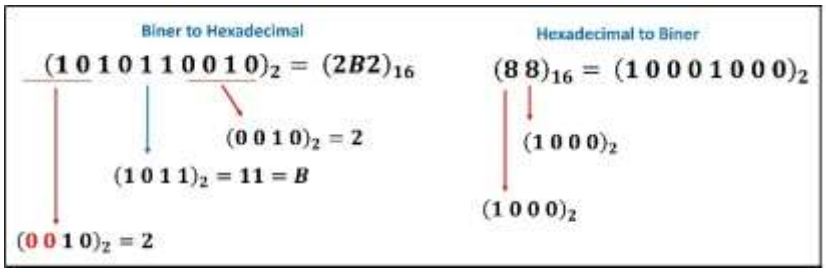
Satu lagi bilangan digital yang perlu diketahui adalah bilangan hexadecimal yang dimana 1 digitnya direpresentasikan oleh 4 bit bilangan biner seperti yang tampak pada tabel 5 di bawah ini.

Tabel 5. Bilangan Hexadecimal

Desimal	0	1	2	3	4	5	6	7
Hexa decimal	0000	0001	0010	0011	0100	0101	0110	0111
Desimal	8	9	10	11	12	13	14	15
Hexa decimal	1000	1001	1010 (A)	1011 (B)	1100 (C)	1101 (D)	1110 (E)	1111 (F)

Gambar 58 di bawah merupakan contoh dari konversi bilangan biner ke hexadecimal dan sebaliknya. Gambar sebelah kiri merupakan konversi bilang biner ke hexadecimal yang dimana 4 bit bilangan biner diambil mulai dari LSB untuk dikonversikan. Jikalau di akhir kekurangan bit maka dapat ditambahkan dengan 0. Gambar sebelah kanan merupakan konversi bilangan

hexadecimal ke biner yang dimana setelah semua bilang hexadecimal dikonversikan akan dikumpulkan dan dapat dikonversikan ke bilangan desimal.



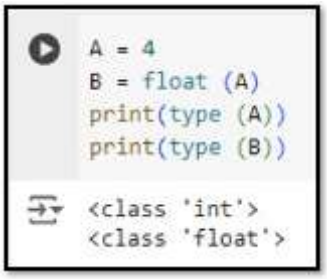
Gambar 58 Konversi Biner – Hexadecimal

Mengkonversi bilangan biner ke hexadecimal dapat dilakukan pada python seperti yang tampak pada gambar 59 di bawah ini. Gambar sebelah kiri mengkoversikan bilangan desimal sebesar 500 ke bilangan hexa menggunakan *hex(bilangan desimal)* dan gambar sebelah kanan mengkoversikan hexadecimal ke desimal menggunakan *0x(bilangan hexadecimal).*

Gambar 59 Konversi Desimal – Hexadecimal di Python

d) Floating Point

Floating point merupakan bilangan pecah dan tipe bilangan ini yang dapat diikutsertakan dalam operasi matematika. Gambar 60 menampilkan 2 variable yang memiliki angka yang sama, hanya saja tiap angka tersebut memiliki tipe data yang berbeda yakni yang satu *integer* dan yang satu lagi *floating point*.



Gambar 60 Tipe angka integer - floating

e) Pembulatan (*Rounding*)

Pembulatan berlaku di tipe angka floating point untuk efisiensi suatu sistem. Pembulatan dapat dilakukan baik mengambil 2 ataupun 3 angka di belakang koma. Gambar 61 memperlihatkan contoh pembulatan floating point di Python.

Gambar 61. Rounding

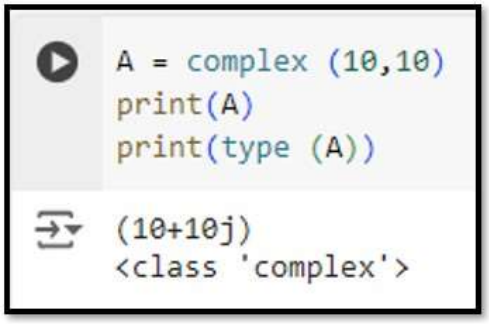
f) Bilangan Kompleks (*Complex Number*)

Bilangan kompleks diperlukan untuk melakukan perhitungan pada hal-hal yang tidak pasti karena di dalam bilangan kompleks tersusun bilangan real dan imajiner. Gambar 62 di bawah ini merupakan contoh bilangan kompleks dan ada fungsi kompleks. yang secara otomatis akan mengkoversikan bilangan di variabel A menjadi bilangan kompleks.

Bilangan kompleks dapat ditulis sebagai berikut:

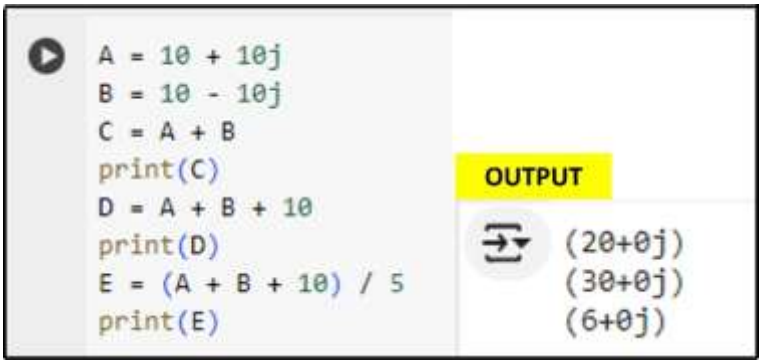
$$z = x + yj$$

X adalah bilangan real dan Y adalah bilangan imajiner dan $j = \sqrt{-1}$



Gambar 62. Complex Number

Dalam bilangan kompleks juga bisa dilakukan operasi aritmatika seperti gambar 63 di bawah ini. Maka sistem akan secara otomatis mengoperasikan bilangan yang sejenis baik real ataupun imajiner.

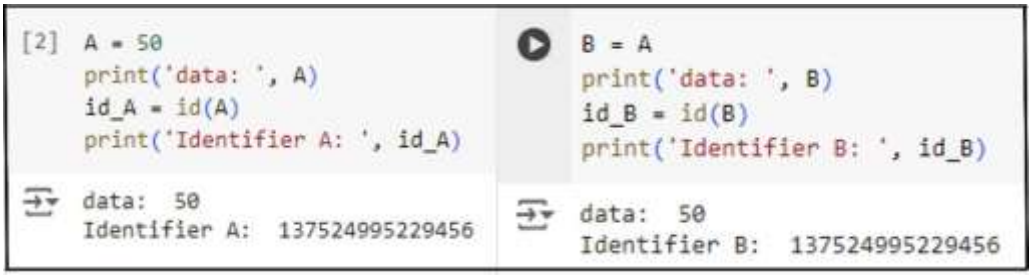


Gambar 63. Operasi Artimatika pada Bilangan Kompleks.

Bab 6 Identifier dan References

a) Identifier

Object dan data di dalam python memiliki identifier (*ID*). Untuk melihat ID pada suatu variabel maka cukup dengan menggunakan *id(variabel)*. Gambar 64 di bawah ini menampilkan contoh identifier pada dua variabel dengan nilai yang sama. Sehingga kedua variabel tersebut memiliki ID yang sama.



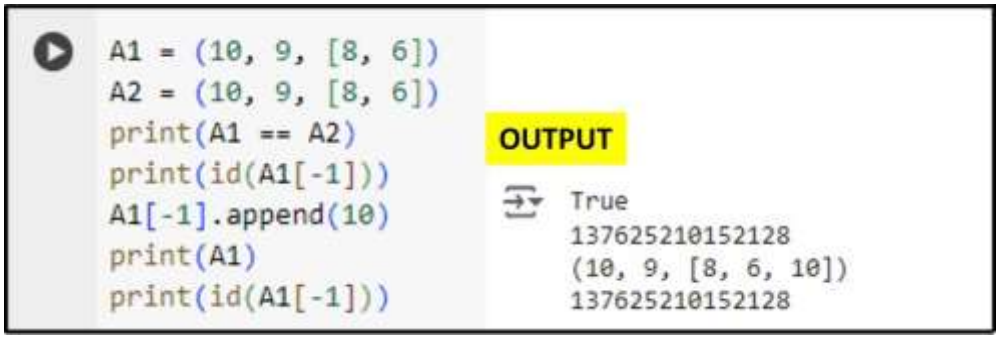
Gambar 64 Identifier pada dua variabel

b) Equality

Gambar 65 di bawah ini juga merupakan aplikasi ID yang dimana tipe data dictionary disimpan dalam satu variabel yang dimana variabel lain berisikan data yang sama. Sehingga *is* dapat digunakan untuk menyatakan kesamaan dengan variabel sebelumnya dan IDnya merujuk ke angka yang sama.

Gambar 65. Equality

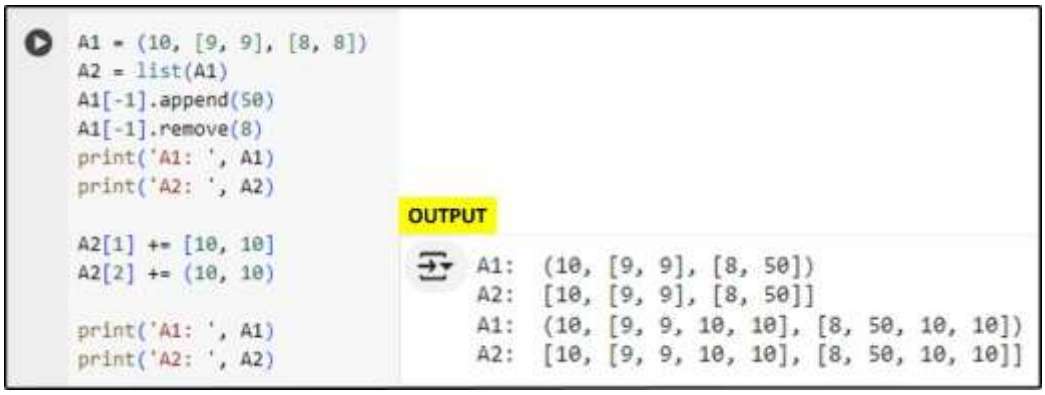
Gambar 66 di bawah ini memiliki dua variable yang berisikan data yang sama sehingga Ketika dicheck equality nya merujuk ke TRUE. Kemudian, pada variable A1 ditambahkan 10 di urutan data paling akhir sehingga Ketika dilakukan pengechekan alamat memory sudah tidak merujuk ke alamat yang sebelumnya.



Gambar 66 Append – ID

c) Shallow Copy

Pada gambar 67 di bawah ini menunjukkan *shallow copy* yang dimana variabel A1 dicopy ulang di variabel A2. Pada variabel A1, angka 50 ditambahkan (*append*) di urutan paling akhir dan angka 8 dihapus (*remove*). Kemudian pada variabel A2, setelah data urutan ke 1 akan ditambahkan [10, 10] dan setelah urutan ke 2 akan ditambahkan (10, 10). Secara keseluruhan, variabel 2 menampikan bentuk tipe list.

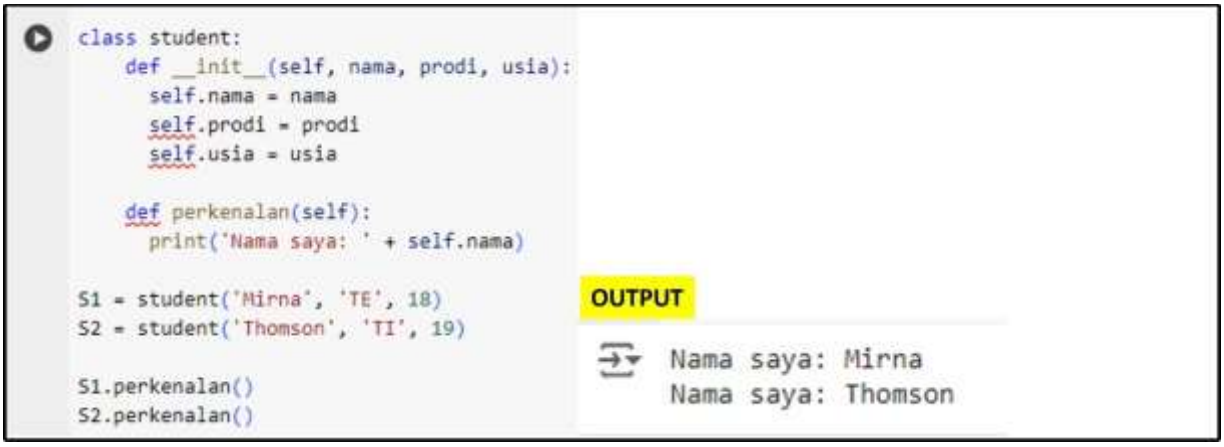


Gambar 67. Shallow Copy

Bab 7 Class dan Object

a) Class

Ketika mendefinisikan suatu class pada program, maka class tersebut udah harus mewakili sifat dan karakteristik seluruh obyek di dalamnya. Gambar 68 di bawah ini merupakan contoh untuk mendefinisikan class yang dimana obyek di dalamnya berupa data mahasiswa.



Gambar 68. Mendefinisikan *Class*

b) Function

Function merupakan kode tersendiri yang dapat dioperasikan ketika dibutuhkan. Function dapat dibangun dengan keyword def dan disertai dengan namanya. Gambar 69 ini merupakan contoh dari penggunaan sederhana function def yang berisikan print(‘welcome’), sehingga ketika dibutuhkan tinggal memanggil welcome().

Gambar 69. Function – Def

Gambar 70 di bawah ini merupakan suatu contoh penggunaan fungsi def untuk mencari keliling lingkaran. Sehingga formula keliling dideskripsikan di function dan function tersebut tinggal

dipanggil saat dibutuhkan. Pada gambar 70 di bawah ini ada dua buah keliling lingkaran yang ingin dicari dengan masing-masing *r* nya adalah 10 dan 20.

```
def lingkaran(message: str, r = float):
    keliling_lingkaran = 2 * 3.14 * r
    print(message, keliling_lingkaran)

lingkaran('Keliling Lingkaran-1: ', 10)
lingkaran('Keliling Lingkaran-2: ', 20)
```

```
Keliling Lingkaran-1:  62.800000000000004
Keliling Lingkaran-2:  125.60000000000001
```

Gambar 70. Function – Def – Keliling Lingkaran

Pada gambar 71 di bawah ini ada dua fungsi digunakan untuk mendeskripsikan formula keliling dan luas lingkaran. Hanya saja dalam fungsi, tugas mempresentasikan output sudah keluar dari bagian function dan berdiri sendiri dengan program gabungannya. Lalu dua variabel yakni X1 dan X2 menampung program yang memanggil ke dua fungsi. Tiap-tiap akhir function diberikan *return value* untuk mengembalikan nilai pada fungsi pemanggilnya.

```
def keliling_lingkaran( r = float):
    KL = round( 2 * 3.14 * r, 2)
    return KL

def luas_lingkaran(r = float):
    LL = 3.14 * r * r
    return LL

X1 = keliling_lingkaran(10)
X2 = luas_lingkaran(10)
print('Keliling Lingkaran: ' , X1)
print('Keliling Lingkaran: ' , X2)
```

```
Keliling Lingkaran:  62.8
Keliling Lingkaran:  314.0
```

Gambar 71 Def-Return Value

c) Posisi Argumen (Positional Argument)

Pada positional argument, argument yang diisi harus berurut dan saat dipanggil harus sesuai dengan urutannya. Gambar 72 merupakan contoh dari *Positional Argument* yang dimana parameternya sudah diurut dan saat pemanggilan juga sudah sesuai dengan urutan parameternya.

```
def student(name, age, subject, year):
  return { 'name' : name, 'age' : age, 'subject' : subject, 'year': year}

S1 = student('Mirna', 19, 'Teknik Elektro', '2022')
print(S1)
```

```
{'name': 'Mirna', 'age': 19, 'subject': 'Teknik Elektro', 'year': '2022'}
```

Gambar 72. Positional Argument

d) *Args

Fungsi *args memampukan satu parameter untuk menerima banyak positional argument. Gambar 73 memperlihatkan perbedaan penggunaan fungsi secara manual dan *args. Gambar sebelah kiri memperlihatkan ada satu parameter menampung lima positional argument dengan operasi dasarnya adalah penjumlahan. Gambar sebelah kanan memperlihatkan penggunaan *args dengan operasi yang sama yakni penjumlahan yang dimana positional argument yang dimasukkan bisa lebih dari 5.

```
def r(r1, r2, r3, r4, r5):
  total = r1 + r2 + r3 + r4 + r5
  print(total)

r(1, 1, 1, 1, 1)
```

```
5
```

```
def penjumlahan(*rs):
  total = 0
  for A in rs:
    total = total + A
  print(total)

penjumlahan(1, 1, 1, 1, 1)
```

```
5
```

Gambar 73. Manual Vs *Args

e) **kwargs

**kwargs merupakan penulisan parameter yang bisa menampung banyak keyword argument dalam satu parameter. Gambar 74 di bawah ini merupakan salah satu contoh dari penerapan *kwargs yang dimana nama-nama pulau Indonesia ditampung dalam beberapa variabel.

```
def pulau(**nama):
  hasil = " "
  for A in nama.values():
    hasil += A
  return hasil
print(pulau(a ='Sumatera', b=' Jawa', c=' Kalimantan', d=' Nias', e=' Sulawesi'))
```

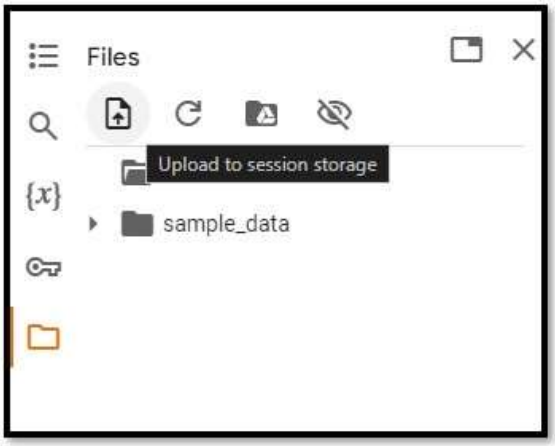
```
 Sumatera Jawa Kalimantan Nias Sulawesi
```

Gambar 74. Penerapan **kwargs

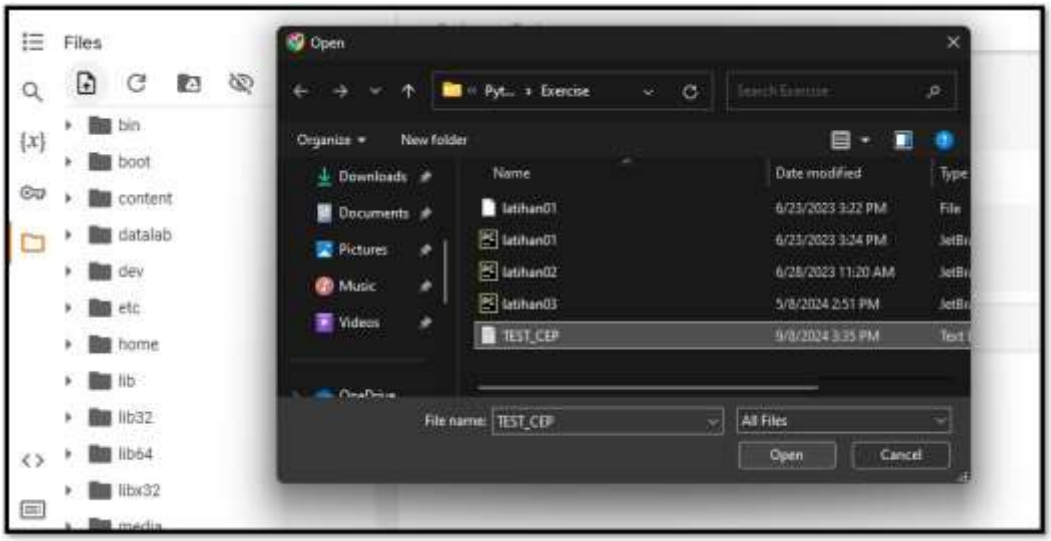
Bab 8 Mengoperasikan File (*.txt dan CSV)*

a) Menulis, Menambahkan dan Menghapus Isi File

Pada python user bisa bekerja dengan sebuah file yang dimana file tersebut dapat kita isi dan hapus. Pertama, user membuat sebuah file.txt di komputer dan kemudian diupload ke files di google colabs seperti yang terlihat pada gambar 75 di bawah ini.

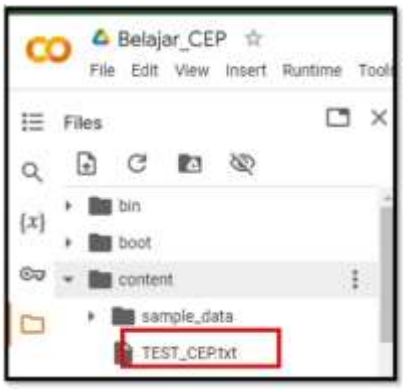


Gambar 75. Files di google colabs



Gambar 76. Upload file.txt

File.txt yang sudah disimpan di computer akan diupload ke google colab seperti yang tampak pada gambar 76. Lalu pada gambar 77 di bawah ini menampilkan bahwa file.txt yang sudah diupload akan tersimpan di files google colab.

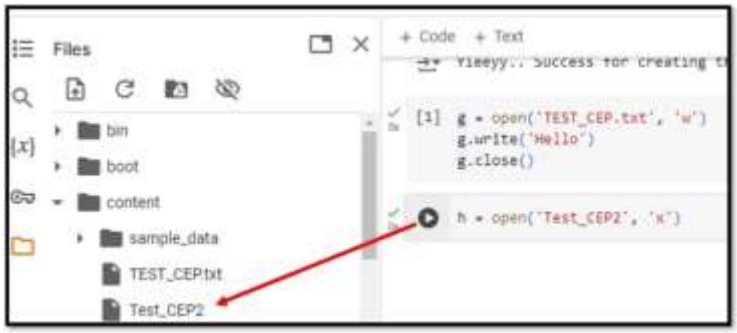


Gambar 77. Files di Google Colab

Untuk mulai mengisi file.txt tersebut maka gunakan fungsi *open(nama_file.txt, aksi)*. Untuk aksi yang digunaka bisa beberapa yakni *w* untuk menulis file, *a* untuk menambah isi *x* untuk menambah file dan selalu akhiri dengan *variable.close()* agar file yang sudah dibaca dan ditulis ditutup otomatis.

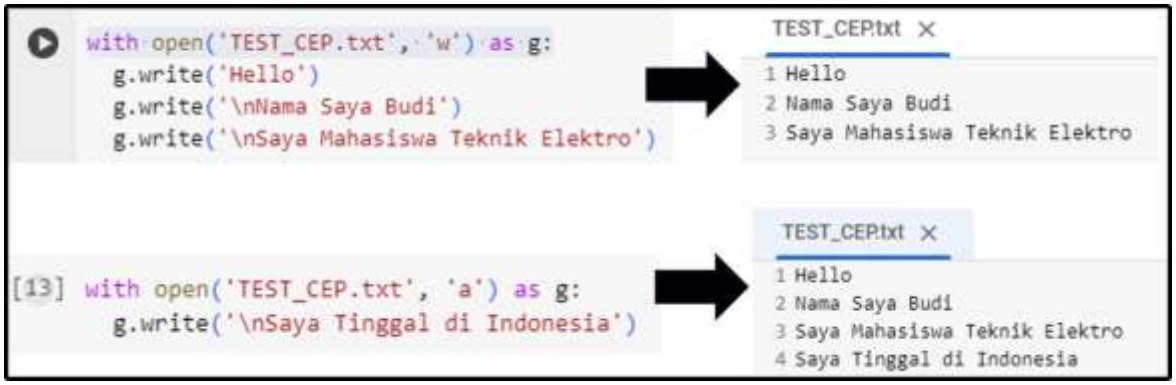
Gambar 78. Write File

Gambar 78 menampilkan suatu file bernama TEST_CEP.txt yang sedang diisi dengan tulisan 'Hello'. Maka user dapat melihat file tersebut sudah terisi dengan cara mengklik file google colab sebelah kiri dan tampilannya akan muncul di sebelah kanan. Lalu untuk menambahkan file lagi tanpa harus mengupload seperti cara sebelumnya dapat menggunakan *x*. Gambar 79 di bawah menampilkan file yang dibuat dan posisinya akan tersimpan di bawah file sebelumnya.



Gambar 79. Menambahkan file dengan *x*

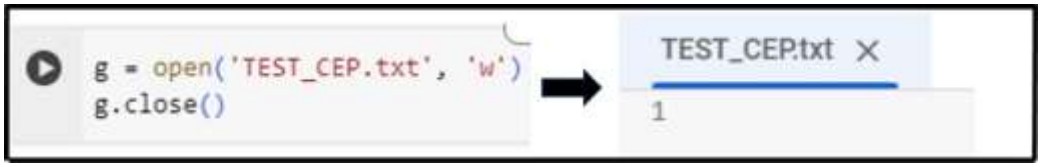
Kemudian ada teknik yang lebif efektif untuk membuka dan menutup file yakni dengan menggunakan *with* dan *open*. Gambar 80 di bawah ini menggunakan fungsi with dan open dan kemudian menulis beberapa line di file.txt. Kemudian jikalau ingin menambahkan tulisan tanpa ingin menimpa maka user dapat menggunakan *a* yang merupakan singkatan dari append. Maka dapat dilihat dari gambar 80, satu tulisan menggunakan fungsi *a* akan ditambahkan otomatis ke bawah tulisa yang sebelumnya.



Gambar 80. Fungsi *with-open* dan *a*

b) Menghapus File dan Direktori

Dan untuk menghapus isi file di TEST_CEP.txt cukup dengan membuka dan menutup kembali file tersebut. Kemudian filte tersebut akan menjadi kosong. Gambar 81 menampilkan kelanjutan deskripsi dari gambar 82.

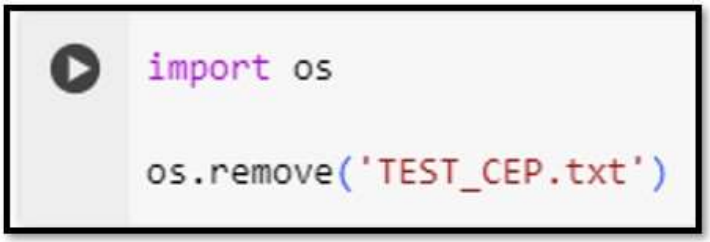


Gambar 81. Menghapus isi file.txt

Gambar 82. Menghapus isi file melalui truncate()

Gambar 82 menjelaskan cara menghapus isi file menggunakan fungsi *truncate()*. Sebelumnya isi file sesua dengan deskripsi di gambar 81, kemudia isi file tersebut dihapus dengan menggunakan fungsi *truncate()*.

Kemudian untuk menghapus file.txt dari file google colaboratory maka menggunakan *os.remove()* dengan terlebih dahulu mengimport os.



Gambar 83. Menghapus File

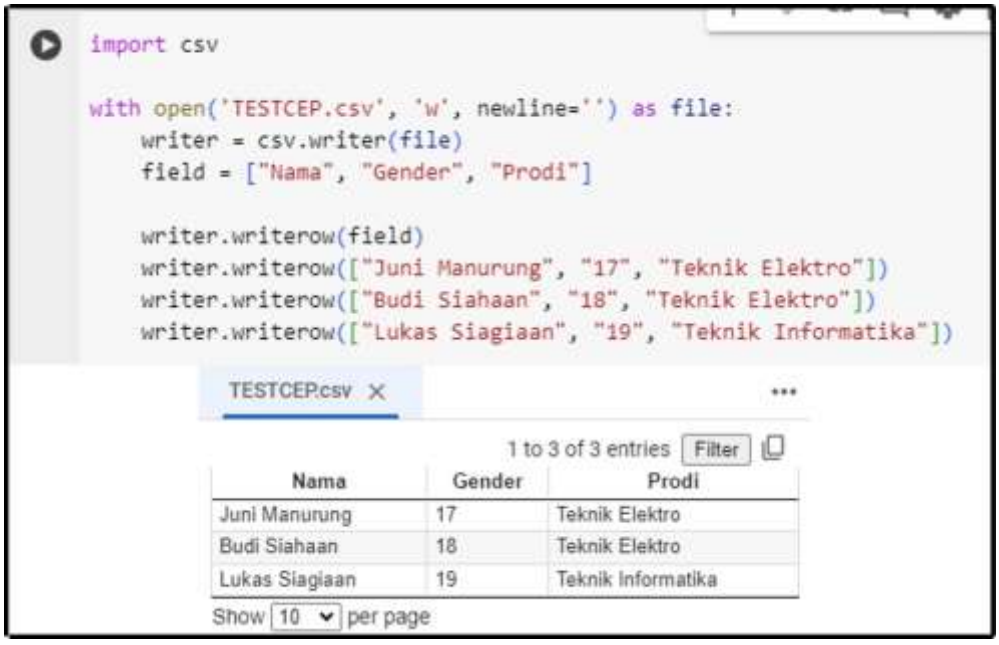
Gambar 83 merupakan proses menghapus file, dimana *os.remove*(namafile.txt), maka secara otomatis file tersebut akan dihapus dan bisa langsung dicheck di file google colab. Gambar 84 di bawah ini menjelaskan proses menghapus direktori di file google colab.

Gambar 84. Menghapus direktori (os.rmdir)

Pada gambar 84 semisalnya user memiliki suatu folder dengan nama TEST_CEP, dan user ingin menghapus folder tersebut maka user cukup memasukkan alamat folder ke os.rmdir.

c) Menulis dan Membaca File CSV

CSV merupakan singkatan ari Comma-Separated Values yang merupakan format paling popular untuk menyimpan data yang bersifat tabular. Gambar 85 merupakan contoh membuat file CSV yang Bernama TESTCEP.csv. Kemudian file tersebut akan diisi dengan tiga kolom yakni Nama, Gender dan Prodi. Lalu, masing-masing baris akan diisid dengan data mahasiswa. Untuk saat ini, hanya tiga data mahasiswa yang ditambahkan.



Gambar 85. File CSV di Python

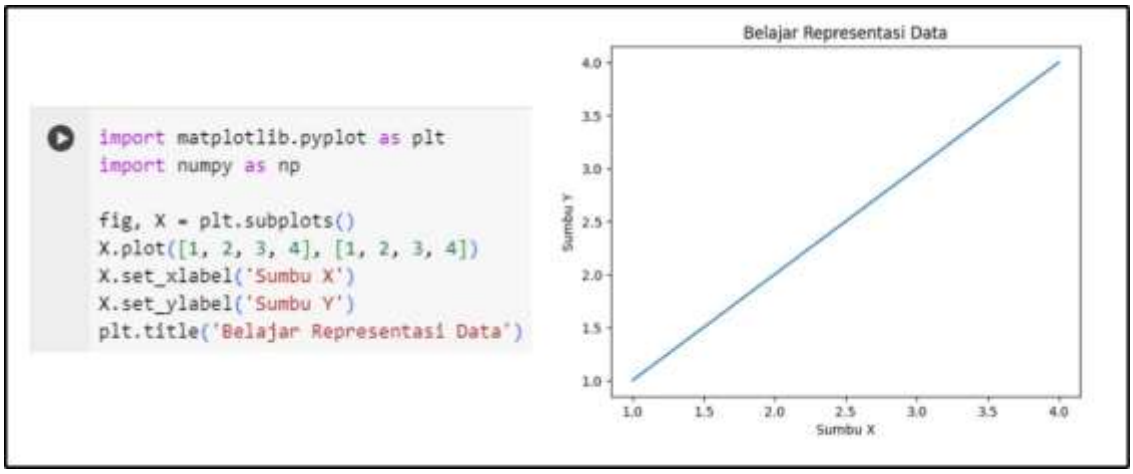
Gambar 86 di bawah ini adalah cara untuk membaca file CSV yang sudah ada. Pertama perlu diimport library CSV dan kemudian membuat suatu variabel yang berisikan fungsi csv.reader. Kemudian satu per satu informasi di CSV akan dibaca dan ditampilkan.

Gambar 86. Membaca file CSV.

Bab 9 Visualisasi Data

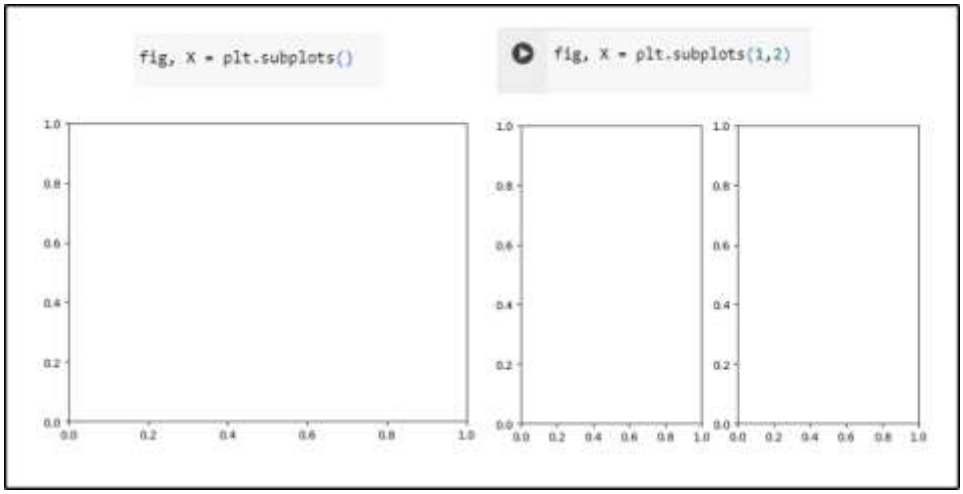
a) Plotting

Untuk mempresentasikan suatu data dalam grafik maka user harus mengimport matplotlib terlebih dahulu ke dalam programnya. Seperti yang tampak pada gambar 87 di bawah ini. Gambar di bawah ini merepresentasikan plot linear yang masing-masing x dan y bernilai sama.



Gambar 87. Representasi Data

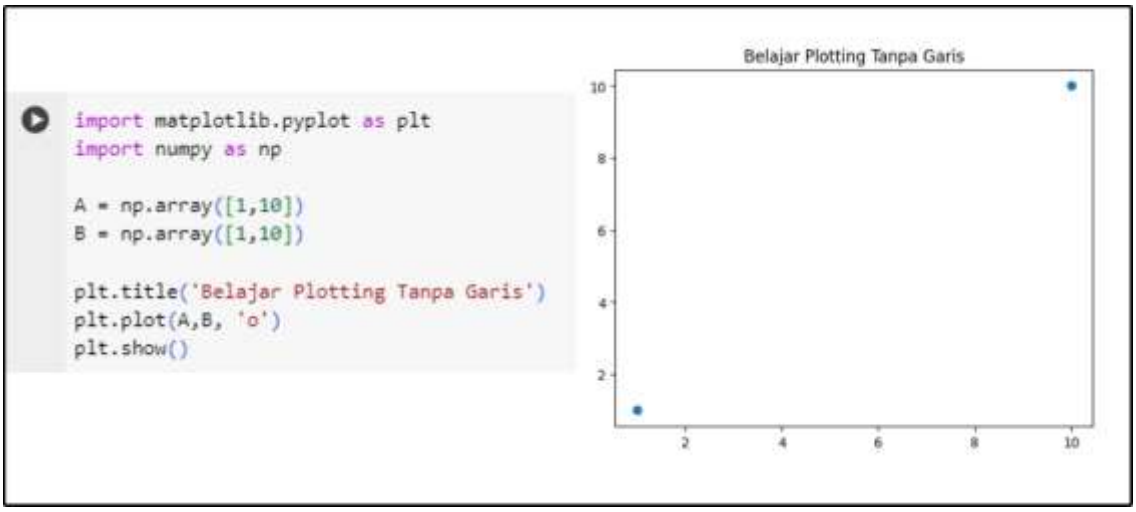
Gambar 88 di bawah menjelaskan fungsi subplot yang dimana jikalau angka tidak dimasukkan maka figure yang keluar ada 1 tetapi jikalau user memasukkan angka 1, 2 maka 1 merepresentasikan baris dan 2 merepresentasikan angka.



Gambar 88. Subplot

b) Plotting Tanpa Garis

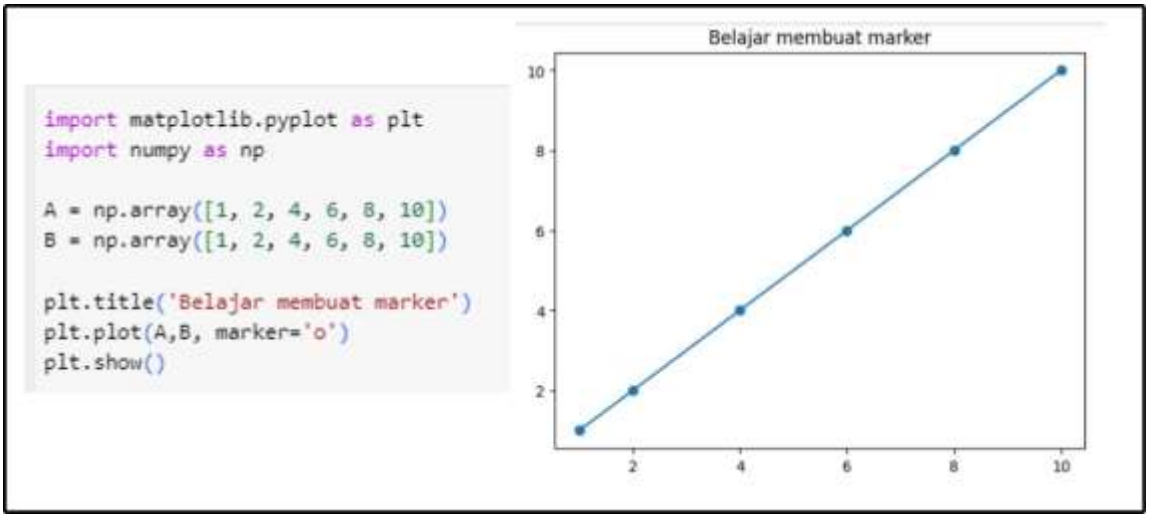
Plotting tanpa garis juga dapat dilakukan seperti gambar 89 di bawah ini. Sehingga hanya menampilkan titik awal dan akhir saja.



Gambar 89. Plotting tanpa garis

c) Tanda (Marker)

Marker ini menjadi sangat penting di dalam grafik untuk melihat nilai-nilai yang dianggap penting. Gambar 90 di bawah ini menampilkan contoh penggunaan marker pada grafik. Sumbu X yang berisikan variabel A dan sumbu Y berisikan variabel B.



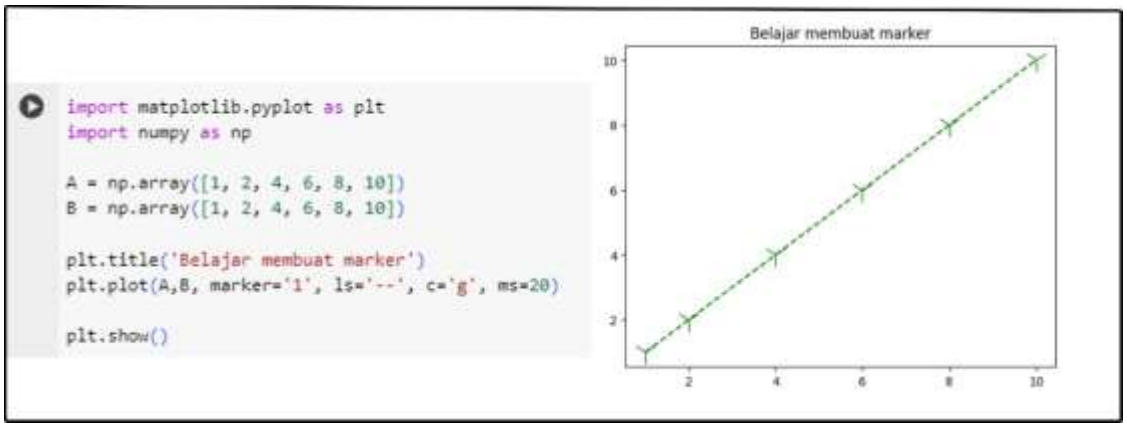
Gambar 90. Membuat Marker

Ada beberapa fitur pada marker yang dapat digunakan sesuai kebutuhan.

Tabel 6. Deskripsi Fitur

No	Fitur	Deskripsi
1	Color (C)	Ada beberapa pilihan warna yang tersedia yakni: 'b' = Biru (Blue) 'g' = Hijau (Green) 'r' = Merah (Red) 'c' = Cyan 'm' = Magenta 'y' = Kuning (Yellow) 'k' = Hitam (Black) 'w' = Putih (White)
2	Linestyle (ls)	Ada beberapa tipe pilihan garis yang tersedia yakni: '-' = garis sambung (solid line style) '--' = garis putus-putus (dashed line style) '-.' = garis putus (dash-dot line style) ':' = garis titik (dotted line style)
3	Marker	Penanda juga memiliki beberapa pilihan seperti di bawah ini: '.' = Point Marker ',' = Pixel Marker 'o' = Circle Marker 'v' = Triangle-Down Marker '^' = Triangle_Up Marker '<' = Triangel_Left Marker '>' = Triangel_Right Marker '1' = Tri_Down Marker '2' = Tri_Up Marker '3' = Tri_Left Marker '4' = Tri_Right Marker

		'8' = Octagon Marker 's' = Square Marker 'p' = Pentagor Marker 'p' = Plus (filled) Marker '*' = Star Marker 'h' = Hexagon1 Marker 'H' = Hexagon2 Marker '+' = Plus Marker 'x' = X Marker 'X' = x (filled) Marker 'D' = Diamond Marker 'd' = Thin_Diamond Marker '|' = Vline Marker '_' = hline Marker
4	Markersize (ms)	Untuk memberikan ukuran marker dan diisi dengan bilangan float.
5	Linewidth (lw)	Untuk memberikan ukuran garis dan diisi dengan bilangan float.

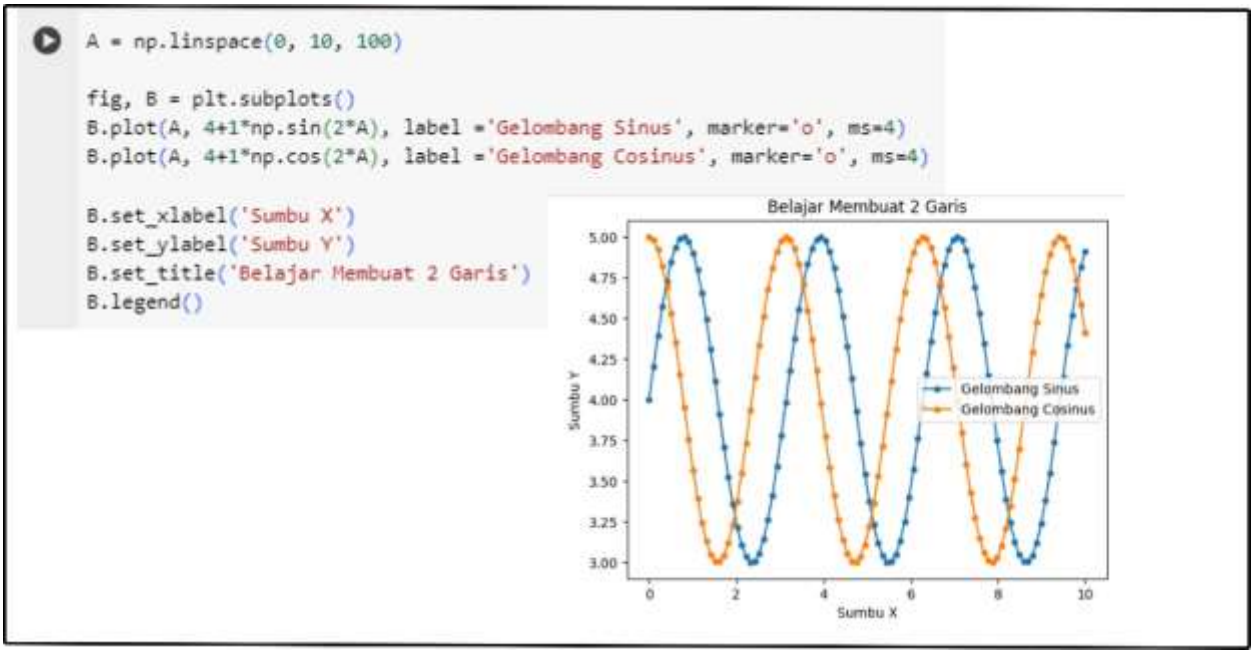


Gambar 91. Variasi Fitur Line dan Marker

Gambar 91 di atas merupakan contoh dalam menggunakan fitur-fitur yang tersedia dengan informasi garis yang sama dengan gambar 90.

d) Visualisasi dua garis dalam satu grafik

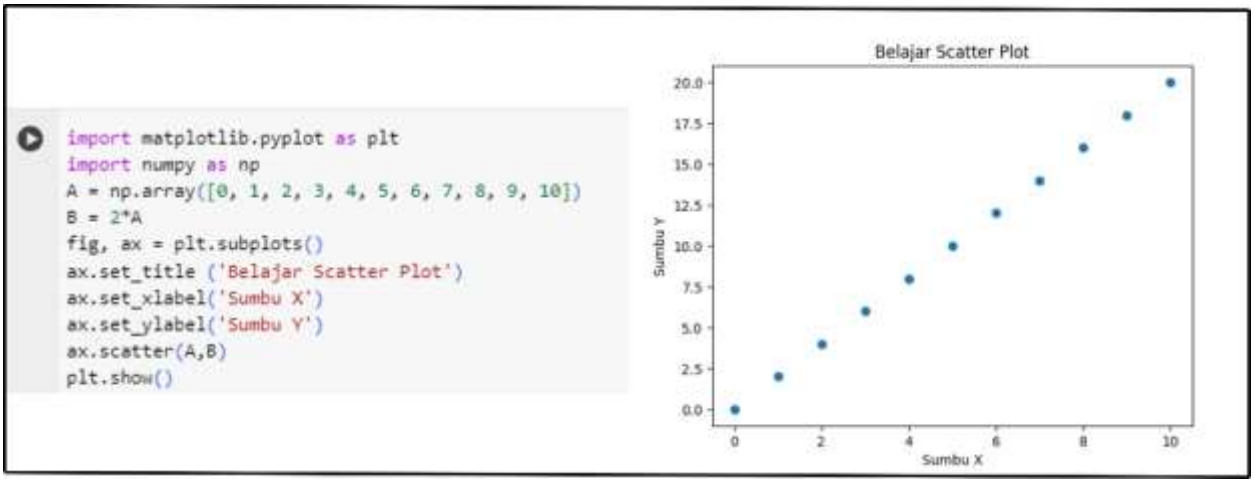
Gambar 92 menjelaskan cara menggabungkan 2 garis dalam 1 grafik dengan menggunakan fitur yang disebutkan sebelumnya. Kali ini dalam grafik, karena datanya sudah mulai banyak maka ada legenda yang bisa membantu *reader* memahami line graph tersebut.



Gambar 92. Dua Line Graph

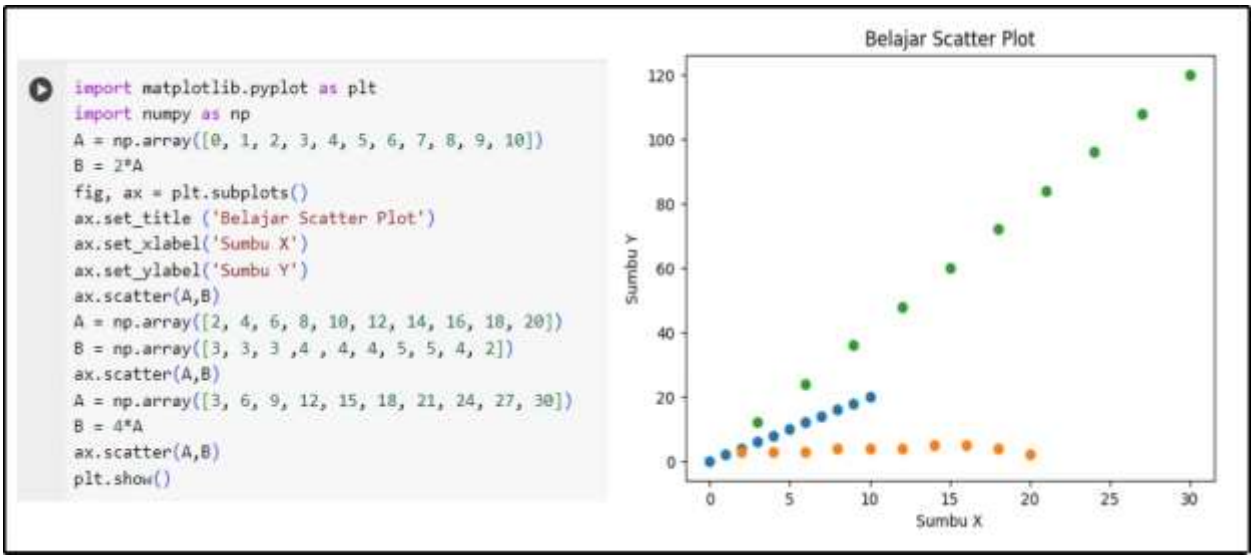
e) Scatter Plot

Selain dengan line graph, data juga dapat direpresentasikan dengan scatter plot. Gambar 93 di bawah ini menampilkan contoh penggunaan scatter plot.



Gambar 93. Scatter Plot

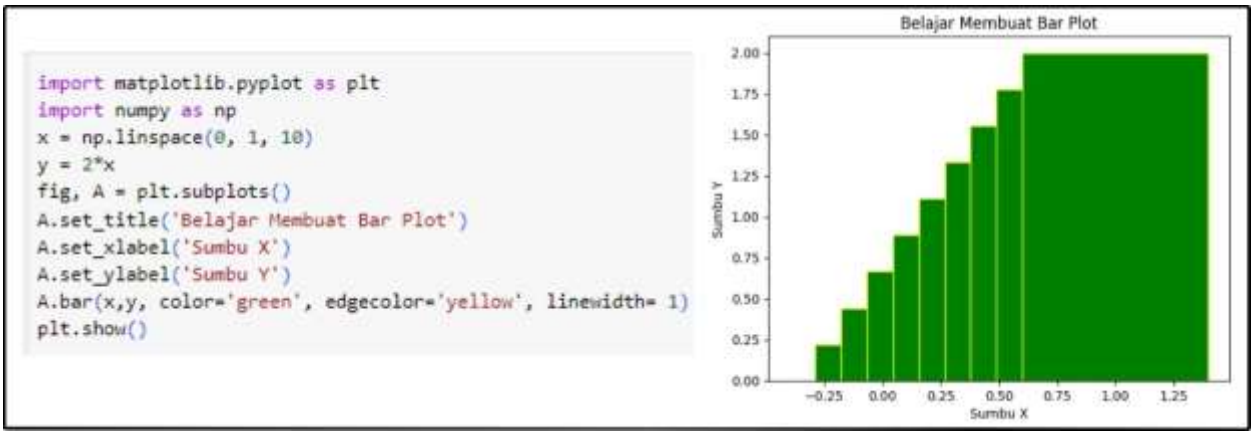
Gambar 94 di bawah ini juga menampilkan scatter plot yang dimana datanya ada tiga inputan dengan ukuran yang sama. Data tersebut direpresentasikan dalam bentuk scatter point dan dengan tiga warna yang berbeda.



Gambar 94. Scatter Plot Tiga Input

f) Bar

Selain dengan line dan scatter plot, representasi data juga dapat dilakukan dengan menggunakan bar-plot seperti gambar 95 di bawah ini. Gambar 95 di bawah ini merepresentasikan input x dan y dalama bar chart yang dimana warna bar adalah hijau dan tepi barnya berwarna kuning.



Gambar 95. Bar Plot

Biografi Penulis 1

Christin Erniati Panjaitan, dilahirkan di Sungai Pakning dan menyelesaikan pendidikan SD, dan SMP di Sungai Pakning. Penulis menyelesaikan pendidikan menengah atas di SMA Kartika I-2 Medan. Mengambil D3 Teknik Telekomunikasi dari Politeknik Caltex Riau (PCR) tahun 2005, lalu melanjutkan ke jenjang sarjana (S1) dengan jurusan Teknik Telekomunikasi dari Institut Teknologi Telkom (sekarang: Telkom University) tahun 2008, dan lulusan S2 dari National Taiwan University of Science and Technology (NTUST) tahun 2016. Tahun 2016-2018, aktif mengajar di Institut Teknologi Del (IT Del) di prodi Teknik Elektro. Tahun 2019, Penulis aktif mengajar di Universitas Prima Indonesia (UNPRI) di prodi Teknik Elektro dan di PU-PT Kesehatan Berbasis IoT & Energi Terbarukan. Penulis aktif menulis dan telah memiliki beberapa publikasi internasional dan nasional. Penulis juga memiliki pengalaman mendapatkan hibah penelitian internal dan nasional. Penulis juga aktif mengikuti program Kemdikbud baik Kampus Mengajar, Praktisi Mengajar, Sertifikasi Kompetensi, Peningkatan Kemampuan Bahasa Inggris (PKBI), Talent Scouting dan lain-lain.

Penulis aktif melakukan kegiatan Tri Darma dan untuk korespondensi dengan penulis, dapat menghubungi: christinpanjaitan@unprimdn.ac.id atau christin.erniati@gmail.

Biografi Penulis 2

Sri Wahyuni Tarigan, memulai kiprah mengajar pada Tahun 2008 di salah satu perusahaan kontraktor di Medan sebelum bergabung di Universitas Prima Indonesia pada Tahun 2014 hingga kini sebagai dosen tetap di Fakultas Sains dan Teknologi Program Studi Teknik Industri. Penulis memiliki hobby traveling dan menulis yang sudah di geluti sejak Tahun 1990, kemudian menuangkannya ke dalam tulisan tangan dan puisi. Latar belakang pendidikan penulis adalah sarjana strata 1 alumni Institut Teknologi Medan Program Studi Teknik Geologi dan pasca sarjana alumni Universitas Negeri Medan Program Studi Pendidikan Kimia. Perpaduan antara latar belakang pendidikan disiplin ilmu yang berbeda membawa penulis melakukan riset geo kimia dan kimia industri untuk di kembangkan dalam berbagai penelitian lapangan serta laboratorium yang menghasilkan penelitian terpadu. Penulis lahir di Medan pada bulan Februari 1973 dan bertempat tinggal di perumahan Terjun Indah di Kecamatan Medan Marelan Kelurahan terjun.

Biografi Penulis 3

Dini M Hutagalung, dilahirkan di Medan dan menyelesaikan pendidikan SD, dan SMP di Medan. Penulis menyelesaikan pendidikan Sekolah Menengah atas di SMA ST. Thomas Medan. Mengambil S1 Fak Pertanian Universitas Sumatera Utara tahun 1995, lalu melanjutkan ke jenjang Magister (S2) dengan jurusan Teknik Informasi (Information Technology) di University of East London tahun 2000-2002. Tahun 2013 sampai sekarang , aktif mengajar di Universitas Sari Mutiara Indonesia.

Penulis aktif melakukan kegiatan Tri Darma dan untuk korespondensi dengan penulis, dapat menghubungi: mhdini@gmail.com.

Biografi Penulis 4

Olnes Yosefa Hutajulu menyelesaikan pendidikan dasar dan menengah di Bandar Huluan dan Pematang Siantar, serta melanjutkan studi S1 di Pendidikan Teknik Elektro Universitas Negeri Medan, lulus tahun 2011. Ia kemudian melanjutkan pendidikan S2 di Universitas Gadjah Mada, Yogyakarta, dan meraih gelar Magister Teknik (M.Eng) pada tahun 2015 dengan beasiswa Beasiswa Pendidikan Pascasarjana Dalam Negeri (BPPDN). Pada tahun 2021, ia memperoleh sertifikasi sebagai Insinyur (Ir) melalui program Pendidikan Profesi Insinyur di Universitas Negeri Medan. Karier akademiknya dimulai sebagai pengajar di Institut Teknologi Del pada tahun 2016 hingga 2017, dan dilanjutkan di Tanri Abeng University dari 2017 hingga 2019. Saat ini, ia adalah dosen ASN di Universitas Negeri Medan sejak tahun 2019. Sebagai akademisi, Olnes memiliki sejumlah publikasi bereputasi di tingkat internasional dan nasional, serta memiliki paten sederhana dan beberapa buku yang telah diterbitkan. Ia aktif menerima hibah penelitian dan pengabdian dari berbagai sumber, baik internal perguruan tinggi maupun nasional. Selain itu, ia juga mendukung program pemerintah melalui peran sebagai Dosen Pembimbing Lapangan (DPL) dalam program Kampus Mengajar dan dalam kegiatan sertifikasi kompetensi. Olnes juga berperan sebagai asesor BNSP dan Tim Penilai Ahli di Kementerian PUPR bidang MEP. Penulis dapat dihubungi melalui email olnes.hutajulu@unimed.ac.id atau nestajulu@gmail.com.

Biografi Penulis 5

Muhammad Dominique Mendoza adalah seorang akademisi dan penulis yang memiliki latar belakang pendidikan dan pengalaman yang luas di bidang teknologi informasi dan manajemen. Beliau menempuh pendidikan SMA di SMA Sutomo 1 Medan sebelum melanjutkan studi S1 di Universitas Bina Nusantara, dengan program double degree dalam Sistem Informasi Komputer dan Manajemen. Kemudian, beliau melanjutkan studi S2 di Universitas Sumatera Utara.

Saat ini, Muhammad Dominique Mendoza adalah dosen di Universitas Negeri Medan, pada program studi Pendidikan Teknologi Informatika dan Komputer, Fakultas Teknik. Beliau mengajar berbagai mata kuliah, termasuk E-Business, Internet of Things (IoT), Manajemen Proyek Teknologi Informatika, serta Rekayasa Industri di bidang pemrograman, jaringan, dan teknologi pendidikan.

Selain sebagai dosen, Muhammad Dominique juga aktif dalam dunia penulisan dan penelitian. Beberapa buku yang telah diterbitkannya antara lain:

1. IoT dalam Pembelajaran
2. Desain Grafis 101
3. Media Pembelajaran Berbantukan Chatbot

Publikasi penelitian Muhammad Dominique dapat diakses melalui akun Google Scholar di tautan berikut: Publikasi Penelitian.

Melalui kontribusinya di bidang pendidikan dan penelitian, Muhammad Dominique terus berusaha mengembangkan dan menyebarkan ilmu pengetahuan, khususnya di bidang teknologi informasi dan pendidikan berbasis teknologi.

Sinopsis

Bahasa Python akan diperlukan karena merupakan dasar bahasa pemrograman yang digunakan di Artificial Intelligence. Google Colaboratory merupakan interface online yang anda bisa pelajari menggunakan perangkat seperti tablet, smartphone, atau laptop, asalkan terhubung dengan internet. Menggunakan Google Colaboratory sangat ringan karena semua library sudah ada, dan tinggal dipanggil saja jikalau dibutuhkan.

Buku ini membahas dasar pemrograman Python di Google Colaboratory, dimulai dengan pengaturan awal di Google dan contoh kode sederhana seperti 'Hello World'. Selanjutnya, dijelaskan tentang variabel,yang dapat diisi dengan teks atau string, serta angka berupa integer dan float. Di dalam buku ini juga membahas berbagai tipe struktur data, seperti List yang dapat diubah (mutable), Dictionary yang mampu menyimpan informasi kompleks, Tuple yang tidak dapat diubah (immutable), dan Set yang mutable.

Dalam pembahasan kondisi, dimulai pemahaman logika Boolean (1 dan 0) dan operator logika seperti AND, OR, dan NOT. Untuk memahami logika, anda perlu memahami konsep digital terlebih dahulu. Kemudian dilanjutkan dengan beberapa tipe *conditional statement* if, if-else, hingga kondisi bertingkat (Chained Condition) dan kondisi bersarang (Nested Condition). Lalu dibahas penggunaan perulangan seperti while loop dan kombinasi break+continue serta for + continue.

Dengan memahami dasar Python akan mudah mempelajari bahasa pemrograman lain dan menggunakan aplikasi offline seperti Pycharm atau Anaconda Navigator.

Daftar Pustaka

1. https://colab.research.google.com/
2. https://revou.co/kosakata/google-colab
3. https://matplotlib.org/
4. Eric Matthes, "*Python Crash Course*", No Starch Press, San Fransisco, 2016.
5. Al Sweigart,"*Automate The Boring Stuff With Python*", No Starch Press, 2015.
6. V., P., Rigdon, "*Calculus 9th Edition*," Pearson New International Edition, 2014.
7. A. Downey, "*Think Python, How to Think Like a Computer Scientist*", 2nd Edition, Green Tea Press, Massachusetts, 2015.
8. L. Ramalho, "*Fluent Python*", O.Reilly Media, 2015.
9. A. Croft, R. Davidson, M. Hargreaves, J. Flint, "*Engineering Mathematics, A Foundation for Electronic, Electrical, Communication and Systems Engineers*", Person Education Limited, 2017.
10. Dasar Pemrograman Python, Novalagung.
11. C. E.Panjaitan, Y.Panjaitan, D.Sitanggang, S.W.Tarigan, "*Image Processing For Detection of Dengue Virus",* Jurnal Sistem Informasi dan Ilmu Komputer Prima, Vol.7 No.2, February 2024.
12. C.Panjaitan, A. Silaban, M.Napitupulu, J.W.Simatupang, "*Comparison K-Nearest Neighbors (K-NN) and Artificial Neural Network (ANN) in Real Time Entrants Recognition"*, IEEE International Seminar on Research of Information and Intelligent System (ISRITI), 2018.
13. Sri Wahyuni Tarigan, "*Use of Computer-Based Educational Media to Improve The Learning Ability of Industrial Engineering Students in Chemistry*", Unpri Press, Medan, 2023.
14. Baharuddin, Olnes Yosefa Hutajulu, Muhammad Dominique Mendoza, Hesti Fibriasari, Muhammad Dani Solihin, "*Monograf inovasi teknologi chatbot berbasis artificial intelligence sebagai learning management system*", Pena Persada Kerta Utama, 2023.
15. Marsangkap Silitonga, Olnes Yosefa Hutajulu, Muhammad Dominique Mendoza, "*Pembelajaran Daring Dengan IoT*", Megalitera, 2021.

16. Reni Rahmadani, Muhammad Dominique Mendoza, Olnes Yosefa Hutajulu, Tansa Trisna Astono Putri, Devi Silvia Panjaitan, Azqal Azqia, "*Membuat dan mengelola sumber daya desain graphic design 101*", Pena Persada Kerta Utama, 2022
17. D.M.Hutagalung, C.E.Panjaitan, "*Use of the SAW (Simple Additive Weighting) Decision Making System in Determining the OSIS Board of Senior High Schools*", Journal of Computer Engineering, System and Science, e-ISSN: 2502-714X, 2023.