

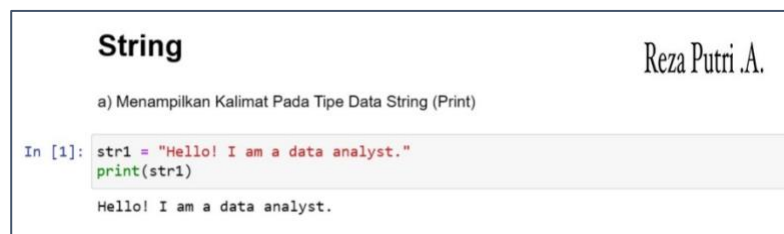
A. EXERCISE WEEK 2

1. Penerapan Metode Pada String, List, dan Dictionary

➤ String

String merupakan salah satu tipe data yang terdapat dalam python untuk menyatakan dan menyimpan karakter berupa huruf baik dalam bentuk kata ataupun kalimat. Terdapat beberapa metode yang dapat di terapkan pada tipe data string, di antaranya yakni :

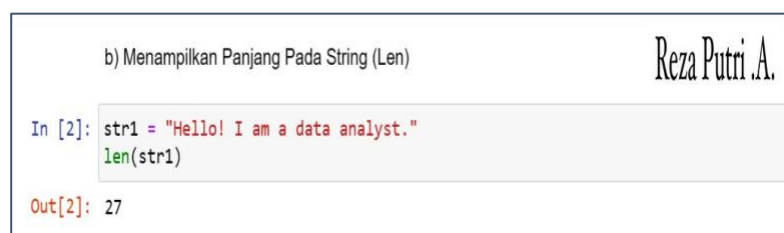
a) Menampilkan Kalimat Pada Tipe Data String (Print)



```
String Reza Putri A.  
a) Menampilkan Kalimat Pada Tipe Data String (Print)  
  
In [1]: str1 = "Hello! I am a data analyst."  
        print(str1)  
  
Hello! I am a data analyst.
```

Pada kode script ini terdapat variabel `str1` yang di pergunakan untuk menyimpan string yang di inputkan. Di inputkan string “Hello! I am a data analyst.”. Dan fungsi `print` di pergunakan untuk mencetak string yang telah di inputkan. Maka, akan di cetak string Hello! I am a data analyst.

b) Menampilkan Panjang Pada String (Len)



```
b) Menampilkan Panjang Pada String (Len) Reza Putri A.  
  
In [2]: str1 = "Hello! I am a data analyst."  
        len(str1)  
  
Out[2]: 27
```

Pada kode script ini terdapat metode string fungsi python `len` yang di pergunakan untuk mengetahui panjang string dengan menggunakan `len()`. Terdapat variabel `str1` yang telah di inputkan. Maka, akan di tampilkan panjang string sebesar 27.

c) Mengakses Index Pada String (String Slices)

➤ Pada Index Tertentu

```
In [3]: str1 = "Hello! I am a data analyst."  
str1[5]  
  
Out[3]: '!'
```

Pada kode script ini terdapat metode string fungsi python untuk mengakses huruf atau karakter pada index tertentu di dalam variabel str1 dengan menggunakan []. Di inputkan index string ke-5. Perlu di ingat bahwa index python di mulai dengan 0. Maka, akan di tampilkan '!'.

➤ Pada Index Pada Rentang Tertentu

```
In [4]: str1 = "Hello! I am a data analyst."  
str1[0:3]  
  
Out[4]: 'Hel'
```

Pada kode script ini terdapat metode string fungsi python untuk mengakses huruf atau karakter pada index rentang tertentu di dalam variabel str1 dengan menggunakan [a:b]. Dengan a adalah index awal dan b adalah index akhir yang akan di tampilkan. Di inputkan a 0 dan b 3. Maka, akan di tampilkan 'Hel'.

➤ Pada Index Rentang Awal Di Tentukan

```
In [5]: str1 = "Hello! I am a data analyst."  
str1[15:]  
  
Out[5]: 'ata analyst.'
```

Pada kode script ini terdapat metode string fungsi python untuk mengakses huruf atau karakter pada index yang rentang awalnya di tentukan hingga ke index rentang akhir di dalam variabel str1 dengan menggunakan [a:]. Dengan a adalah index awal. Di inputkan a 15. Maka, akan di tampilkan 'ata analyst.'.

➤ Pada Index Rentang Akhir Di Tentukan

```
In [6]: str1 = "Hello! I am a data analyst."
        str1[:7]

Out[6]: 'Hello! '
```

Reza Putri A.

Pada kode script ini terdapat metode string fungsi python untuk mengakses huruf atau karakter pada index yang di mulai dari index awal hingga index yang rentang akhirnya di tentukan di dalam variabel str1 dengan menggunakan [:b]. Dengan b adalah index akhir. Di inputkan b 7. Maka, akan di tampilkan 'Hello! '.

d) Mengubah Karakter Menjadi Karakter Lain Pada String (Replace)

```
In [7]: str1 = "Hello! I am a data analyst."
        str1.replace("analyst", "scientist")

Out[7]: 'Hello! I am a data scientist.'
```

d) Mengubah Karakter Menjadi Karakter Lain Pada String (Replace)

Reza Putri A.

Pada kode script ini terdapat metode string fungsi python replace yang di pergunakan untuk mengubah suatu karakter mejadi karakter lain yang di inginkan dengan menggunakan (a,b). Dengan a adalah karakter awal yang ingin di ubah menjadi b karakter yang di inginkan. Di inputkan a analyst dan b scientist. Maka, akan di ubah karakter analyst menjadi scientist, dan di tampilkan 'Hello! I am a data scientist'.

e) Mencari Suatu Karakter Pada String (Find)

```
In [8]: str1 = "Hello! I am a data analyst."
        str1.find("H")

Out[8]: 0
```

e) Mencari Suatu Karakter Pada String (Find)

Reza Putri A.

Pada kode script ini terdapat metode string fungsi python find yang di pergunakan untuk mencari suatu karakter dengan menggunakan find("a"). Dengan a adalah karakter yang di cari. Di inputkan katakter "H" yang akan

di cari pada variabel str1. Maka, akan di tampilkan bahwa karakter “H” terletak pada index ke 0.

f) Membagi Karakter Pada String Oleh Pemisah Yang Di Tentukan (Split)

```
f) Membagi Karakter Pada Data String Oleh Pemisah Yang Di Tentukan (Split) Reza Putri .A.  
In [9]: str2 = "Langkah-Langkah Data Wrangling"  
        str2.split()  
Out[9]: ['Langkah-Langkah', 'Data', 'Wrangling']
```

Pada kode script ini terdapat metode string fungsi python split yang di pergunakan untuk membagi karakter oleh pemisah yang di tentukan dengan menggunakan split(). Di inputkan str2 “Langkah-Langkah Data Wrangling” yang akan di pisah jika terdapat spasi antara satu karakter dengan karakter lainnya. Maka, akan di tampilkan ['Langkah-Langkah', 'Data', 'Wrangling'].

g) Menggabungkan Karakter Pada String (Join)

```
g) Menggabungkan Karakter Pada String (Join) Reza Putri .A.  
In [10]: str2 = "Langkah-Langkah", "Data", "Wrangling"  
         j = "-"  
         j.join(str2)  
Out[10]: 'Langkah-Langkah-Data-Wrangling'
```

Pada kode script ini terdapat metode string fungsi python join yang di pergunakan untuk menggabungkan karakter dengan tanda karakter yang di tentukan dengan menggunakan join(). Terdapat 2 variabel yakni variabel str2 untuk menginputkan data string yang akan di gabungkan dan j adalah karakter untuk menggabungkan data string. Di inputkan str2 “Langkah-Langkah”, “Data”, “Wrangling” yang akan di gabungkan dengan karakter “-”. Maka, akan di tampilkan ‘Langkah-Langkah-Data-Wrangling’.

h) Menampilkan Bearapa Kali Karakter Muncul Pada String (Count)

h) Menampilkan Bearapa Kali Karakter Muncul Pada String (Count) Reza Putri .A.

```
In [11]: str2 = "Langkah-Langkah Data Wrangling"
str2.count("Langkah")

Out[11]: 2
```

Pada kode script ini terdapat metode string fungsi python count yang di pergunakan untuk menampilkan berapa kali karakter muncul dengan menggunakan count("a"). Dengan a adalah karakter yang akan di hitung berapa kali muncul. Terdapat variabel str2 "Langkah-Langkah Data Wrangling". Di inputkan a "Langkah". Maka, akan di tampilkan bahwa karakter "Langkah" muncul sebanyak 2 kali.

i) Mengubah Karakter Pertama Menjadi Huruf Besar Pada String (Capitalize)

i) Mengubah Karakter Pertama Menjadi Huruf Besar Pada String (Capitalize) Reza Putri .A.

```
In [12]: str2 = "langkah-langkah data wrangling"
str2.capitalize()

Out[12]: 'Langkah-langkah data wrangling'
```

Pada kode script ini terdapat metode string fungsi python capitalize yang di pergunakan untuk mengubah karakter pertama menjadi huruf besar dengan menggunakan capitalize(). Terdapat variabel str2 "langkah-langkah data wrangling" yang akan di ubah karakter pertamanya menjadi huruf besar. Maka, akan di tampilkan 'Langkah-langkah data wrangling'.

j) Mengubah Karakter Pertama Setiap Kata Menjadi Huruf Besar Pada String (Title)

j) Mengubah Karakter Pertama Setiap Kata Menjadi Huruf Besar Pada String Reza Putri .A.

```
In [13]: str2 = "langkah-langkah data wrangling"
str2.title()

Out[13]: 'Langkah-Langkah Data Wrangling'
```

Pada kode script ini terdapat metode string fungsi python title yang di pergunakan untuk mengubah karakter pertama setiap kata menjadi huruf besar dengan menggunakan title(). Terdapat variabel str2 "langkah-langkah

data wrangling” yang akan di ubah karakter pertamanya pada setiap kata menjadi huruf besar. Maka, akan di tampilkan ‘Langkah-Langkah Data Wrangling’.

k) Mengubah Setiap Karakter Menjadi Huruf Besar Pada String (Upper)

```
k) Mengubah Setiap Karakter Menjadi Huruf Besar Pada String (Upper) Reza Putri .A.  
  
In [14]: str2 = "langkah-langkah data wrangling"  
         str2.upper()  
  
Out[14]: 'LANGKAH-LANGKAH DATA WRANGLING'
```

Pada kode script ini terdapat metode string fungsi python upper yang di pergunakan untuk mengubah setiap karakter menjadi huruf besar dengan menggunakan upper(). Terdapat variabel str2 “langkah-langkah data wrangling” yang akan di ubah setiap karakternya menjadi huruf besar. Maka, akan di tampilkan “LANGKAH-LANGKAH DATA WRANGLING”.

l) Mengubah Setiap Karakter Menjadi Huruf Kecil Pada String (Lower)

```
l) Mengubah Karakter Menjadi Huruf Kecil Pada String (Lower) Reza Putri .A.  
  
In [15]: str2 = "langkah-langkah data wrangling"  
         str2.lower()  
  
Out[15]: 'langkah-langkah data wrangling'
```

Pada kode script ini terdapat metode string fungsi python lower yang di pergunakan untuk mengubah setiap karakter menjadi huruf kecil dengan menggunakan lower(). Terdapat variabel str2 “langkah-langkah data wrangling” yang akan di ubah karakternya menjadi huruf kecil. Maka, akan di tampilkan ‘langkah-langkah data wrangling’.

m) Mengubah Setiap Karakter Huruf Besar Menjadi Huruf Kecil Dan Sebaliknya Pada String (Swapcase)

```
m) Mengubah Setiap Karakter Huruf Besar Menjadi Huruf Kecil Dan Sebaliknya Pada String (Swapcase)
```

```
In [16]: str2 = "Langkah-Langkah Data Wrangling"
str2.swapcase()
```

```
Out[16]: 'LANGKAH-LANGKAH dATA wRANGLING'
```

Reza Putri A.

Pada kode script ini terdapat metode string fungsi python swapcase yang di pergunakan untuk mengubah setiap karakter huruf besar menjadi huruf kecil dan sebaliknya dengan menggunakan swapcase(). Terdapat variabel str2 “Langkah-Langkah Data Wrangling” yang akan di ubah karakter huruf kecilnya menjadi huruf besar dan sebaliknya. Maka, akan di tampilkan ‘LANGKAH-LANGKAH dATA wRANGLING’.

n) Menghilangkan Spasi Karakter Pada String

- Menghilangkan Spasi Di Kanan Atau Kiri Karakter Pada String (Strip)

```
n) Menghilangkan Spasi Pada String
```

```
In [17]: str3 = "  Pengolahan Data Terstruktur  "
str3.strip()
```

```
Out[17]: 'Pengolahan Data Terstruktur'
```

Reza Putri A.

Pada kode script ini terdapat metode string fungsi python strip yang di pergunakan untuk menghilangkan atau menghapus spasi di kanan atau kiri karakter dengan menggunakan strip(). Terdapat variabel str3 “ Pengolahan Data Terstruktur ” yang akan di hapus spasi di sebelah kanan dan kiri karakternya. Maka, akan di tampilkan ‘Pengolahan Data Terstruktur’.

- Menghilangkan Spasi Di Kanan Karakter Pada String (Rstrip)

```
In [18]: str3 = "  Pengolahan Data Terstruktur  "
str3.rstrip()
```

```
Out[18]: '  Pengolahan Data Terstruktur'
```

Reza Putri A.

Pada kode script ini terdapat metode string fungsi python `rstrip` yang di pergunakan untuk menghilangkan atau menghapus spasi di kanan karakter dengan menggunakan `rstrip()`. Terdapat variabel `str3` “ Pengolahan Data Terstruktur ” yang akan di hapus spasi di sebelah kanan karakternya. Maka, akan di tampilkan ‘ Pengolahan Data Terstruktur’.

➤ Menghilangkan Spasi Di Kiri Karakter Pada String (`Lstrip`)

```
In [19]: str3 = "Pengolahan Data Terstruktur "
str3.lstrip()

Out[19]: 'Pengolahan Data Terstruktur'
```

Reza Putri .A.

Pada kode script ini terdapat metode string fungsi python `lstrip` yang di pergunakan untuk menghilangkan atau menghapus spasi di kiri karakter dengan menggunakan `lstrip()`. Terdapat variabel `str3` “ P engolahan Data Terstruktur ” yang akan di hapus spasi di sebelah kiri karakternya. Maka, akan di tampilkan ‘Pengolahan Data Terstruktur’.

o) Membandingkan Dua String (`==`)

```
o) Membandingkan Dua String

In [20]: str4 = "Pengolahan Data Terstruktur"
str5 = "Pengolahan Data Tidak Struktun"
str6 = "Pengolahan Data Terstruktur"
(str4==str5)

Out[20]: False

In [21]: str4 = "Pengolahan Data Terstruktur"
str5 = "Pengolahan Data Tidak Struktun"
str6 = "Pengolahan Data Terstruktur"
(str4==str6)

Out[21]: True
```

Reza Putri .A.

Pada kode script ini terdapat metode string fungsi python yang di pergunakan untuk membandingkan dua string apakah sama atau berbeda, dengan kondisi jika sama akan menampilkan “True” dan jika berbeda akan menampilkan “False”.

Pada kondisi pertama variabel `str4` di bandingkan dengan `str5` dan di tampilkan “False” karena, isi dari kedua variabel tersebut berbeda. Pada kondisi

kedua variabel str4 di bandingkan dengan str6 dan di tampilkan “True” karena, isi dari kedua variabel tersebut sama.

➤ List

List merupakan salah satu tipe data yang terdapat pada python untuk menyatakan dan menyimpan karakter data yang memiliki tipe data integer (bilangan bulat), float (bilangan desimal), string (huruf), dan boolean, ataupun campuran dari tipe data yang berbeda. Terdapat beberapa metode yang dapat di terapkan pada tipe data list, di antaranya yakni :

a) Menampilkan List (Print)

```
a) Menampilkan List (Print) Reza Putri A.  
  
In [22]: list1 = [89, 90, 91, 92, 97, 98]  
         print(list1)  
  
[89, 90, 91, 92, 97, 98]
```

Pada kode script ini terdapat variabel list1 yang di pergunakan untuk menyimpan list yang di inputkan. Di inputkan list1 = [89, 90, 91, 92, 97, 98]. Dan fungsi print di pergunakan untuk mencetak list yang di inputkan. Maka, akan di cetak list [89, 90, 91, 92, 97, 98].

b) Menampilkan Panjang Pada List (Len)

```
b) Menampilkan Panjang Pada List (Len) Reza Putri A.  
  
In [23]: list1 = [89, 90, 91, 92, 97, 98]  
         len(list1)  
  
Out[23]: 6
```

Pada kode script ini terdapat metode list fungsi python len yang di pergunakan untuk mengetahui panjang list dengan menggunakan len(). Terdapat variabel list1 yang telah di inputkan. Maka, akan di tampilkan panjang list sebesar 6.

c) Mengakses Index Pada List (List Slices)

➤ Pada Index Tertentu

```
In [24]: list1 = [89, 90, 91, 92, 97, 98]
list1[0]

Out[24]: 89
```

Pada kode script ini terdapat metode list fungsi python untuk mengakses nilai dari list pada index tertentu dengan menggunakan []. Terdapat variabel list1 yang telah di inputkan dan inputkan index ke 0. Maka, akan di tampilkan sebesar 89.

➤ Pada Index Rentang Tertentu

```
In [25]: list1 = [89, 90, 91, 92, 97, 98]
list1[1:3]

Out[25]: [90, 91]
```

Pada kode script ini terdapat metode list fungsi python untuk mengakses nilai dari list pada index rentang tertentu dengan menggunakan [a:b]. Dengan a adalah index awal dan b adalah index akhir. Terdapat variabel list1 yang telah di inputkan dan di inputkan index awal 1 dan index akhir 3. Maka, akan di tampilkan sebesar [90, 91].

➤ Pada Index Rentang Awal Di Tentukan

```
In [26]: list1 = [89, 90, 91, 92, 97, 98]
list1[2:]

Out[26]: [91, 92, 97, 98]
```

Pada kode script ini terdapat metode list fungsi python untuk mengakses nilai dari list pada index rentang awal di tentukan dengan menggunakan [a:]. Dengan a adalah index awal. Terdapat variabel list1 yang telah di inputkan dan di inputkan a 2. Maka, akan di tampilkan sebesar [91, 92, 97, 98].

➤ Pada Index Rentang Akhir Di Tentukan

```
In [27]: list1 = [89, 90, 91, 92, 97, 98]
list1[:4]

Out[27]: [89, 90, 91, 92]
```

Reza Putri .A.

Pada kode script ini terdapat metode list fungsi python untuk mengakses nilai dari list pada index rentang akhir di tentukan dengan menggunakan [:b]. Dengan b adalah index akhir. Terdapat variabel list1 yang telah di inputkan dan di inputkan b 4. Maka, akan di tampilkan sebesar [89, 90, 91, 92].

d) Menambahkan Suatu Nilai Pada List (Append)

```
d) Menambahkan Suatu Nilai Pada List (Append)

In [28]: list1 = [89, 90, 91, 92, 97, 98]
list1.append(99)
list1

Out[28]: [89, 90, 91, 92, 97, 98, 99]
```

Reza Putri .A.

Pada kode script ini terdapat metode list fungsi python untuk menambah suatu nilai dari list pada index terakhir dengan menggunakan append(). Terdapat variabel list1 yang telah di inputkan dan di inputkan append sebesar 99 yang akan di tampilkan pada index terakhir. Maka, akan di tampilkan nilai dari list1 sebesar [89, 90, 91, 92, 97, 98, 99].

e) Mengganti Suatu Nilai Pada List (Update)

```
e) Mengganti Suatu Nilai Pada List (Update)

In [29]: list1 = [89, 90, 91, 92, 97, 98]
list1[1] = 85
list1

Out[29]: [89, 85, 91, 92, 97, 98]
```

Reza Putri .A.

Pada kode script ini terdapat metode list fungsi python untuk mengganti suatu nilai dari list pada index tertentu dengan menggunakan list[a] = b. Dengan a adalah nilai dari index yang ingin di ganti dan b adalah nilai yang di inginkan. Terdapat variabel list1 yang telah di inputkan dan di inputkan a 1 dan b 85. Maka, akan di tampilkan sebesar [89, 85, 91, 92, 97, 98].

f) Menyisipkan Suatu Nilai Di Index Tertentu Pada List (Insert)

```
f) Menyisipkan Suatu Nilai Di Index Tertentu Pada List (Insert) Reza Putri .A.  
In [30]: list1 = [89, 90, 91, 92, 97, 98]  
list1.insert(1, 95)  
list1  
Out[30]: [89, 95, 90, 91, 92, 97, 98]
```

Pada kode script ini terdapat metode list fungsi python untuk menyisipkan suatu nilai dari list pada index tertentu dengan menggunakan insert(a, b). Dengan a adalah posisi index dan b adalah nilainya. Terdapat variabel list1 yang telah di inputkan dan di inputkan index 1 dan nilai 95. Maka, akan di tampilkan sebesar [89, 95, 90, 91, 92, 97, 98].

g) Menghapus Suatu Nilai Di Index Tertentu Pada List (Pop)

```
g) Menghapus Suatu Nilai Di Index Tertentu Pada List (Pop) Reza Putri .A.  
In [31]: list1 = [89, 90, 91, 92, 97, 98]  
list1.pop(3)  
list1  
Out[31]: [89, 90, 91, 97, 98]
```

Pada kode script ini terdapat metode list fungsi python untuk menghapus suatu nilai di index tertentu dengan menggunakan pop(a). Dengan a adalah index dari nilai yang ingin di hapus. Terdapat variabel list1 yang telah di inputkan dan di inputkan a 3, nilai index 3 akan di hapus. Maka, akan di tampilkan sebesar [89, 90, 91, 92, 97, 98].

h) Melakukan Sorting Pada List (Sort)

```
h) Melakukan Sorting Pada List (Sort) Reza Putri .A.  
In [32]: list1 = [89, 90, 91, 92, 97, 98]  
list1.sort()  
list1  
Out[32]: [89, 90, 91, 92, 97, 98]
```

Pada kode script ini terdapat metode list fungsi python untuk melakukan sorting dengan menggunakan sort(). Terdapat variabel list1 yang telah di inputkan dan di lakukan sorting dari nilai-nilai pada variabel list1. Maka, akan di tampilkan [89, 90, 91, 92, 97, 98].

i) Menampilkan Berapa Kali Suatu Nilai Muncul Pada List (Count)

```
i) Menampilkan Berapa Kali Suatu Nilai Muncul Pada List (Count) Reza Putri .A.  
In [33]: list2 = [99, "Aya", 97, "Niona", 92, "Aya"]  
         list2.count("Aya")  
Out[33]: 2
```

Pada kode script ini terdapat metode list fungsi python untuk menampilkan berapa kali suatu nilai muncul dengan menggunakan count(a). Dengan a adalah nilai yang ingin di cari. Terdapat variabel list2 yang telah di inputkan dan di inputkan a adalah “Aya”. Maka, akan di tampilkan sebesar 2.

j) Membalikkan Urutan Pada List (Reversed)

```
j) Membalikkan Urutan Pada List (Reversed) Reza Putri .A.  
In [34]: list3 = [99, 97, 96, 95, 94, 92, 91]  
         for n in reversed (list3):  
             print(n)  
91  
92  
94  
95  
96  
97  
99
```

Pada kode script ini terdapat metode list fungsi python untuk membalikkan urutan pada list dengan menggunakan reversed dan looping for. Terdapat variabel list3 yang telah di inputkan. Maka, akan di tampilkan reversed dari list3 sebesar [91, 92, 94, 95, 96, 97, 99].

➤ **Dictionary**

Dictionary merupakan salah satu tipe data pada python untuk menyimpan kumpulan data atau nilai yang setiap urutannya berisi urutan dengan pendekatan “key-value”. Dictionary bisa menyimpan berbagai macam tipe data, mulai dari tipe data asli maupun tipe data objek. Terdapat beberapa metode yang dapat di terapkan pada tipe data dictionary, di antaranya yakni :

a) Menampilkan Dictionary (Print)

```
a) Menampilkan Dictionary (Print) Reza Putri .A.  
In [35]: dic1 = {"A" : "85", "B" : "80", "C" : "75"}  
         dic1  
Out[35]: {'A': '85', 'B': '80', 'C': '75'}
```

Pada kode script ini terdapat variabel dic1 yang di pergunakan untuk menyimpan key dan value yang di inputkan. Maka, akan di tampilkan {'A' : '85', 'B' : '80', 'C' : '75'}.

b) Menampilkan Panjang Pada Dictionary (Len)

```
b) Menampilkan Panjang Pada Dictionary (Len) Reza Putri A.  
In [36]: dic1 = {"A" : "85", "B" : "80", "C" : "75"}  
         len(dic1)  
Out[36]: 3
```

Pada kode script ini terdapat metode dictionary fungsi python len yang di pergunakan untuk mengetahui panjang dictionary dengan menggunakan len(). Terdapat variabel dic1 yang telah di inputkan. Maka, akan di tampilkan nilai panjang dari dic1 sebesar 3.

c) Mengakses Index Pada Dictionary (Get)

```
c) Mengakses Index Pada Dictionary (Get) Reza Putri A.  
In [37]: dic1 = {"A" : "85", "B" : "80", "C" : "75"}  
         dic1.get("A")  
Out[37]: '85'
```

Pada kode script ini terdapat metode dictionary fungsi python get yang di pergunakan untuk mengakses nilai index tertentu dengan menggunakan get(a). Dengan a adalah key yang akan di cari. Terdapat variabel dic1 yang telah di inputkan dan di inputkan a "A". Maka, akan di tampilkan sebesar 85.

d) Mengubah Suatu Nilai Dari Key Pada Dictionary (Get)

```
d) Mengubah Suatu Nilai Dari Key Pada Dictionary (Get) Reza Putri A.  
In [38]: dic1 = {"A" : "85", "B" : "80", "C" : "75"}  
         dic1["C"] = "79"  
         dic1.get("C")  
Out[38]: '79'
```

Pada kode script ini terdapat metode dictionary fungsi python [] dan get yang di pergunakan untuk mengubah suatu nilai dari key. Dengan nilai dari c akan di

ubah menjadi 79. Terdapat variabel dic1 yang telah di inputkan. Maka, akan di tampilkan nilai dari c sebesar 79.

e) Menambahkan Item Pada Dictionary

```
e) Menambahkan Item Pada Dictionary Reza Putri .A.  
In [39]: dic1 = {"A" : "85", "B" : "80", "C" : "75"}  
         dic1["D"] = "70"  
         dic1  
Out[39]: {'A': '85', 'B': '80', 'C': '75', 'D': '70'}
```

Pada kode script ini terdapat metode dictionary fungsi python [] yang di pergunakan untuk menambahkan item di index paling akhir. Terdapat variabel dic1 yang telah di inputkan dan di inputkan item “D” dengan nilai item 70. Maka, akan di tampilkan {‘A’ : ‘85’, ‘B’ : ‘80’, ‘C’ : ‘75’, ‘D’ : ‘70’}.

f) Menghapus Item Pada Dictionary (Delete)

```
f) Menghapus Item Pada Dictionary (Delete) Reza Putri .A.  
In [40]: dic2 = {"Nama" : "Lyodra", "Usia" : "19", "Asal" : "Sukabumi"}  
         del dic2["Nama"]  
         dic2  
Out[40]: {'Usia': '19', 'Asal': 'Sukabumi'}
```

Pada kode script ini terdapat metode dictionary fungsi python delete yang di pergunakan untuk menghapus item pada key tertentu dengan menggunakan del. Terdapat variabel dic2 yang telah di inputkan dan di inputkan del dic2 dengan key “Nama”. Maka, akan di hapus key dan nilai dari nama, dan di tampilkan {‘Usia’ : ‘19’, ‘Asal’ : ‘Sukabumi’}.

g) Menggunakan Perulangan Untuk Mencetak Dictionary (Loop For)

```
g) Menggunakan Perulangan Untuk Mencetak Dictionary (Loop For) Reza Putri .A.  
In [41]: dic2 = {"Nama" : "Lyodra", "Usia" : "19", "Asal" : "Sukabumi"}  
         for n in dic2:  
             print(dic2[n])  
Lyodra  
19  
Sukabumi
```

Pada kode script ini terdapat metode dictionary fungsi python yang di pergunakan untuk mencetak dictionary menggunakan perulangan loop for.

Terdapat variabel dic2 yang telah di inputkan dan di lakukan perulangan pada dic2. Maka, akan di tampilkan Lyodra, 19, Sukabumi.

2. Membuat List 100 Nomor Random

2. Membuat List 100 Nomor Random

Reza Putri .A.

```
In [80]: import random as rd

len_list = []
for n in range(10):
    number = []
    divide = []

    for k in range(100):
        random = rd.randint(1,100)
        number.append(random)
        if random % 3 == 0:
            divide.append(random)

    div_list = len(number) - len(divide)
    len_list.append(div_list)

average = sum(len_list) / len(len_list)

print("Di peroleh nilai - nilai dari selisih sebesar : ", (len_list))
print("Di peroleh nilai rata - rata sebesar : ", (average))

Di peroleh nilai - nilai dari selisih sebesar : [72, 64, 73, 70, 71, 66, 80, 69, 68, 69]
Di peroleh nilai rata - rata sebesar : 70.2
```

Penjelasan Deskriptif :

Pada kode script ini menggunakan library random dari python dengan cara import random as rd. Terdapat 3 list yakni, len_list untuk menyimpan selisih dari semua bilangan, number untuk menyimpan 100 bilangan acak, dan divide untuk menyimpan bilangan acak dari number yang habis di bagi 3. Dan di lakukan looping sebanyak 10 kali.

Lalu, di berikan looping di dalam looping sebelumnya sebesar 100 untuk memberikan nilai dari bilangan acak. Dengan random sebagai variabel untuk menuliskan proses rumus random dari angka bilangan 1 sampai 100. Setelahnya, bilangan random tersebut akan di masukkan ke dalam list number. Dan di lakukan pengecekan, jika bilangan habis di bagi 3, maka bilangan akan masuk ke list divide.

Terdapat variabel div_list untuk menuliskan proses rumus menghitung selisih dari panjang kedua list bilangan tersebut. Nilai selisih tersebut akan di simpan pada list len_list. Terdapat variabel average pula yang di pergunakan untuk menuliskan proses rumus mencari rata - rata dari bilangan yang telah di simpan di len_list. Terdapat print len_list di pergunakan untuk menampilkan nilai-nilai dari selisih. Dan print average di pergunakan untuk menampilkan nilai rata – rata. Maka, akan di tampilkan nilai len_list

sebanyak sepuluh bilangan, yakni [72, 64, 73, 70, 71, 66, 80, 69, 68,69]. Dan di tampilkan nilai rata – rata dari len_list tersebut sebesar 70,2.

B. EXERCISE WEEK 3

1. Menampilkan Data Dari HTML Ke Dataframe Sesuai Dengan Yang Di Inginkan

a) Membaca Dan Menampilkan HTML

```
a) Membaca Dan Menampilkan HTML
```

```
In [1]: import pandas as pd

data_html = "https://en.wikipedia.org/wiki/List_of_2022_American_television_debuts"
dfs = pd.read_html(data_html)
dfs
```

```
Out[1]: [
      List of years in American television:
0      2019 2020 2021 2022 2023 2024 2025
1      List of 2022 American television debuts
2      2021-22 United States network television schedule
3      2022-23 United States network television schedule
4      List of American television programs currently...
      First aired      Title      Channel      Source
0      January 1      We Baby Bears      Cartoon Network      [1]
1      January 2      Alex vs. America      Food Network      [2]
2      January 2      Next Level Chef      Fox      [3]
3      January 3      The Cleaning Lady      Fox      [4]
4      January 3      Ugliest Home in America      HGTV      [5][6]
...
301     December 15      Love for the Ages      Peacock      [340]
302     December 15      Sonic Prime      Netflix [344][339] [345]
303     December 16      The Recruit      Netflix      [346]
304     December 18      1923      Paramount+ [347][348]
305     December 19      The Wheel      NBC      [349][350]
```

Pada kode script ini, menggunakan library pandas untuk melakukan pembacaan data dengan format HTML. Dengan melakukan import pandas as pd. Lalu, url dari HTML di simpan ke dalam variabel data_html. Variabel dfs di pergunakan untuk menuliskan proses membaca HTML dengan pd.read_html(). Maka, akan di tampilkan data dari HTML.

b) Menampilkan Panjang HTML

```
b) Menampilkan Panjang HTML
```

```
In [2]: print("Di peroleh panjang HTML : ", (len(data_html)))

Di peroleh panjang HTML : 69
```

Pada kode script ini, menggunakan python len yang di pergunakan untuk menampilkan panjang HTML dari url pada data_html yang di inputkan. Maka, akan di tampilkan panjang HTML 69.

c) Menampilkan Tabel Miniseries

c) Menampilkan Tabel Miniseries

Reza Putri .A.

```
In [3]: dfs[3]
```

Out[3]:

	First aired	Title	Channel	Source
0	January 6	Women of the Movement	ABC	[401][402]
1	January 6	Let the World See	ABC	[403]
2	January 6	Joe Montana: Cool Under Pressure	Peacock	[404]
3	January 6	Indivisible: Healing Hate	Paramount+	[405]
4	January 7	El Deafo	Apple TV+	[406]
...
57	December 4	George & Tammy	Showtime/Paramount Network	[460]
58	December 8	Harry & Meghan	Netflix	[461]
59	December 22	The Best Man: The Final Chapters	Peacock	[263][462]
60	December 25	The Witcher: Blood Origin	Netflix	[463][464]
61	December 27	Chef Dynasty: House of Fang	Food Network	[465]

Pada kode script ini menggunakan python `dfs[a]` untuk menampilkan tabel dataframe yang di inginkan. Dengan `a` adalah index dari tabel dataframe yang ingin di tampilkan. Di inputkan `a` adalah 3, maka akan di tampilkan dataframe tabel miniseries.

Pada dataframe tabel ini berisi 4 kolom dengan header (judul) first aired, title, channel, dan source, dan berisi 62 baris. Pada dasarnya, python memiliki fungsi bawaan default menampilkan data 5 teratas dan data 5 terbawah. Maka, akan di tampilkan tabel dataframe dai miniseries 5 teratas dan 5 terbawah.

d) Menampilkan Ukuran Dataframe Tabel Miniseries

d) Menampilkan Ukuran Dataframe Tabel Miniseries

Reza Putri .A.

```
In [4]: dfs[3].shape
```

Out[4]: (62, 4)

Pada kode script ini menggunakan python library pandas `shape` untuk menampilkan ukuran dataframe dari tabel miniseries dengan menggunakan `dfs[3].shape`. Maka, akan di tampilkan ukuran dataframe dari tabel miniseries sebesar 62 baris dan 4 kolom.

2. Menampilkan Data Tabel Dari PDF Ke Dataframe Sesuai Dengan Yang Di Inginkan

a) Membaca Dan Menampilkan PDF

```
a) Membaca Dan Menampilkan PDF Reza Putri .A.  
  
In [1]: import tabula as tb  
data_pdf = "Housing_data.pdf"  
dfs = tb.read_pdf(data_pdf, pages = "all")  
dfs  
  
Out[1]: [ 0.17004 12.5 7.87 0 0.524 6.004 85.9 6.5921 5 311  
0 0.22489 12.5 7.87 0 0.524 6.377 94.3 6.3467 5 311  
1 0.11747 12.5 7.87 0 0.524 6.009 82.9 6.2267 5 311  
2 0.09378 12.5 7.87 0 0.524 5.889 39.0 5.4509 5 311,  
15.2 386.71 17.1 18.9  
0 15.2 392.52 20.45 15.0  
1 15.2 396.90 13.27 18.9  
2 15.2 390.50 15.71 21.7]
```

Pada kode script ini, menggunakan java dan library python tabula dengan mengimport tabula as tb. Dengan di lakukan penggunaan penulisan java di command prompt "java -version". Lalu, file dari pdf di simpan ke dalam variabel data_pdf. Variabel dfs di pergunakan untuk menuliskan proses membaca pdf dengan tb.read_pdf(data_pdf, pages = "a"). Dengan a adalah jumlah halaman yang ingin di tampilkan. Di inputkan a all. Maka, akan di tampilkan data dari semua halaman yang ada PDF.

b) Menampilkan Tabel Indeks 0

```
b) Menampilkan Tabel Indeks 0 Reza Putri .A.  
  
In [2]: dfs[0]  
  
Out[2]:  
0.17004 12.5 7.87 0 0.524 6.004 85.9 6.5921 5 311  
0 0.22489 12.5 7.87 0 0.524 6.377 94.3 6.3467 5 311  
1 0.11747 12.5 7.87 0 0.524 6.009 82.9 6.2267 5 311  
2 0.09378 12.5 7.87 0 0.524 5.889 39.0 5.4509 5 311
```

Pada kode script ini, menggunakan python dfs[a] untuk menampilkan tabel dataframe yang di inginkan. Dengan a adalah index dari tabel dataframe yang ingin di tampilkan. Di inputkan a sebesar 0. Maka, akan di tampilkan tabel dataframe dari indeks ke 0.

Pada tabel dataframe ini berisi 10 kolom dengan header (judul) 0.17004, 12.5, 7.87, 0, 0.524, 6.004, 85.9, 6.5921, 5, 311, dan berisi 3 baris.

c) Menampilkan Tabel Indeks 1

```
c) Menampilkan Tabel Indeks 1 Reza Putri A.  
  
In [3]: dfs[1]  
Out[3]:
```

	15.2	386.71	17.1	18.9
0	15.2	392.52	20.45	15.0
1	15.2	396.90	13.27	18.9
2	15.2	390.50	15.71	21.7

Pada kode script ini, menggunakan python `dfs[a]` untuk menampilkan tabel dataframe yang di inginkan. Dengan a adalah index dari tabel dataframe yang ingin di tampilkan. Di inputkan a sebesar 1. Maka, akan di tampilkan tabel dataframe dari indeks ke 1.

Pada tabel dataframe ini berisi 4 kolom dengan header (judul) 15.2, 386.71, 17.1, 18.9, dan berisi 3 baris.

3. Penjelasan Mengenai BeautifulSoup dan Web Scraping

Beautiful Soup merupakan salah satu library python yang di pergunakan untuk mengakses dan melakukan ekstrasi file dengan format XML atau HTML dan mengambil data. Bisa di katakan pula bahwa beautiful soup merupakan library yang di pergunakan untuk melakukan parsing elemen HTML. Memiliki fungsi sebagai parser (penguraian) untuk memisahkan komponen – komponen XML atau HTML menjadi rangkaian yang lebih mudah untuk di baca.

Sedangkan, Web Scraping merupakan sebuah proses atau sistem yang di pergunakan untuk melakukan ekstrasi data dari data semi-terstruktur, seperti website. Dapat di katakan pula bahwa web scraping adalah sebuah cara untuk melakukan pengambilan suatu data atau informasi tertentu dengan jumlah besar dalam waktu yang singkat. Web scraping memiliki dua metode, yakni metode manual dan otomatis. Dalam prosesnya, web scraping memiliki beberapa tahapan, di antaranya yakni :

- Memasukkan link url untuk di lakukan scraping.
- Scraper akan memuat seluruh kode HTML untuk halaman tersebut.
- Scraper akan mulai mengekstrak data pada halaman tertentu yang di pilih.

- Di lakukan ekstraksi untuk semua data dan di tampilkan data dengan format yang lebih mudah di pahami oleh pengguna.

Antara beautiful soup dengan web scraping tentunya saling berkaitan, library beautiful soup merupakan library yang sering di gunakan untuk web scraping, hal ini di karenakan library beautiful soup relatif lebih mudah di gunakan.