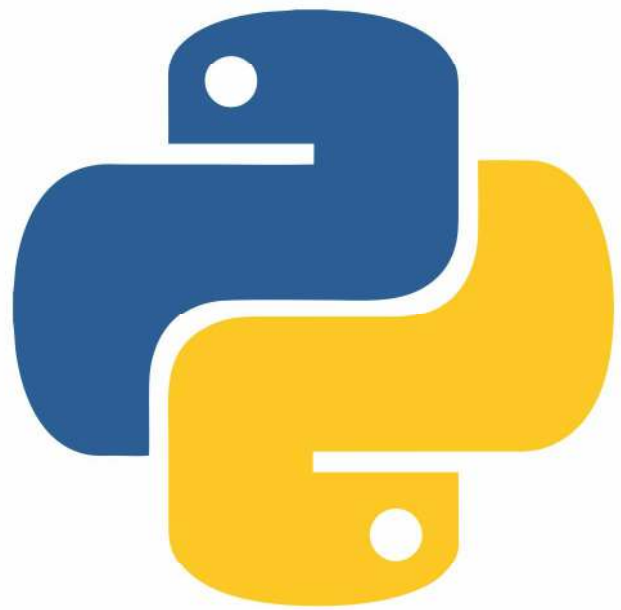


PYTHON

(FOR DATA SCIENCE)
Pemrograman V



PYTHON

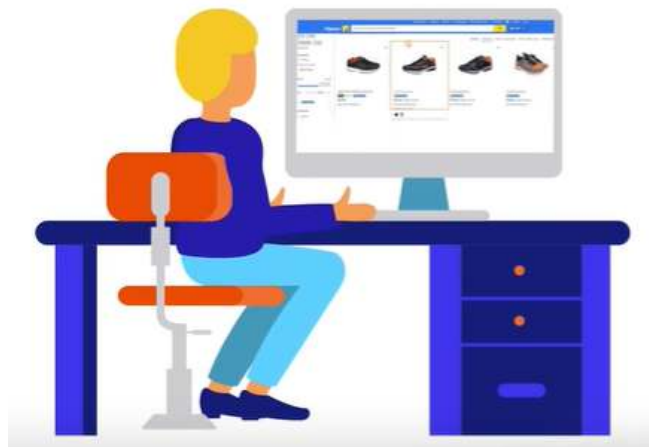
Data Science

Data science adalah salah satu disiplin ilmu yang secara khusus mempelajari soal data terutama data kuantitatif atau data numerik. Saat ini, ilmu yang satu ini mulai menjelma menjadi suatu profesi baru di bidang teknologi yang banyak dicari oleh berbagai jenis perusahaan.

Secara umum data science adalah penggalian atau bisa juga disebut mengekstrak data agar dapat difilter serta didapatkan data yang benar untuk menghasilkan produk data yang sebenar-benarnya.

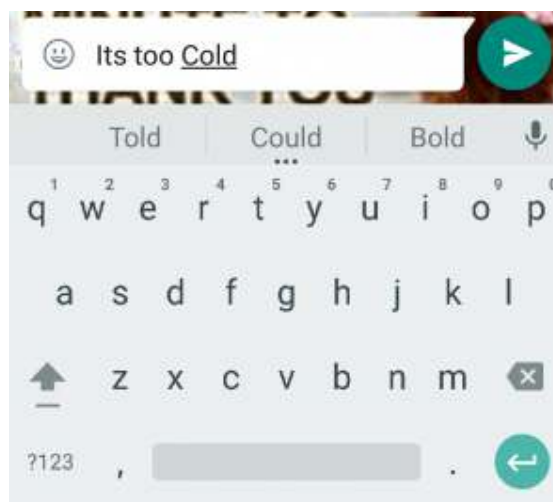
1.1 Contoh Implementasi Data Science

- Rekomendasi Barang Online Shop



Gambar 1 Online Shop

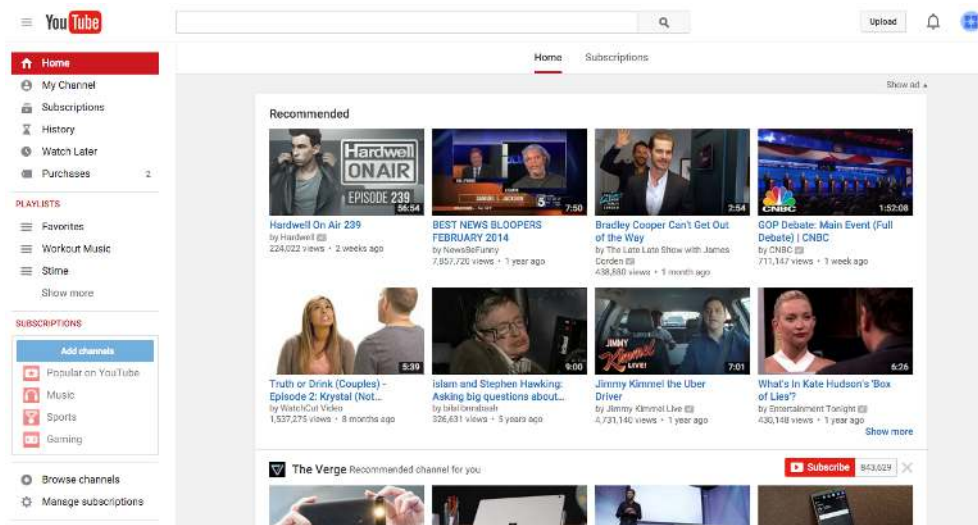
- Prediksi Kata pada Keyboard Smartphone



Gambar 2 Smartphone Keyboard

PYTHON

- Rekomendasi Video Youtube, dll



Gambar 3 Rekomendasi Video Youtube

2.1 Pandas DataFrame

Pandas berasal dari kata Python Data Analysis Library, turunan dari kata Panel Data. Mendukung data multi-dimensi yang artinya elemen-elemen pada data diakses dengan menggunakan 2 buah index. Sedangkan data satu dimensi adalah elemen pada data dapat diakses hanya dengan 1 buah index.

	Age	Location	Name
0	24	New York	John
1	13	Paris	Anna
2	53	Berlin	Peter
3	33	London	Linda

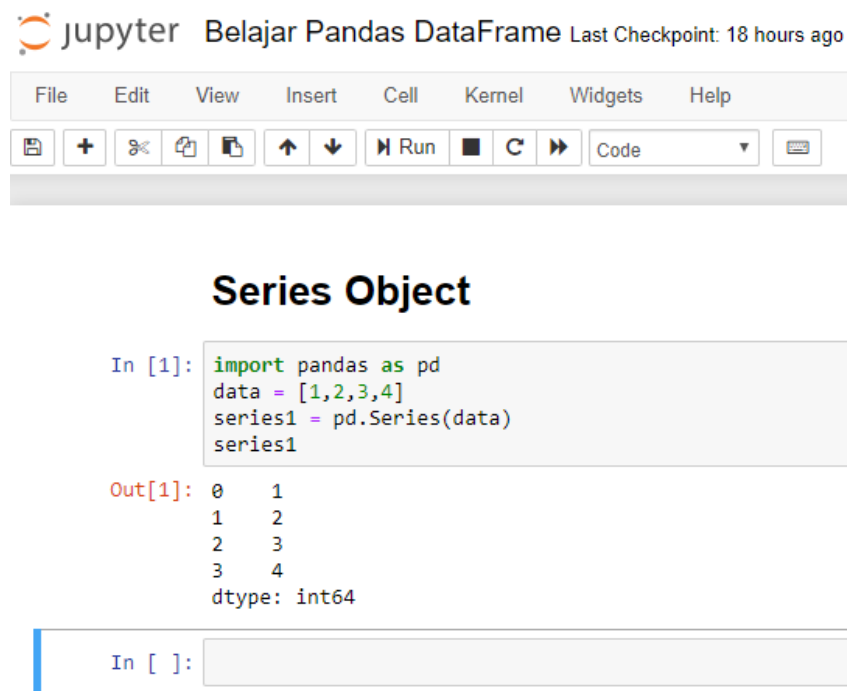
Gambar 4 Data Multi-Dimensi

2.1.1 Series

Series merupakan struktur data dasar dalam Pandas. Series adalah data satu dimensi yang dapat berisi tipe data seperti integer, string, dll. Dan mendukung tipe data sama atau campuran. Contoh series object.

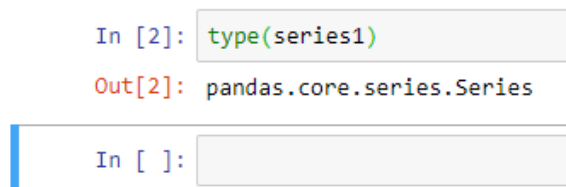
PYTHON

1. Membuat Series



Gambar 5 Series Object

2. Cek Tipe Struktur Data Series atau Bukan



Gambar 6 Cek Tipe Struktur Data Series atau Bukan

3. Ubah Nama Index

Ubah Nama Index

```
In [3]: series1 = pd.Series(data, index = ['a','b','c','d'])
series1
```

```
Out[3]: a    1
        b    2
        c    3
        d    4
        dtype: int64
```

Gambar 7 Ubah Nama Index

PYTHON

2.1.2 DataFrame

DataFrame merupakan array dua dimensi dengan baris dan kolom. Struktur data ini merupakan cara paling standar untuk menyimpan data. Secara sederhana, DataFrame merupakan tabel/data tabular. Setiap kolom pada DataFrame merupakan objek dari Series, dan baris terdiri dari elemen yang ada pada Series. Contoh DataFrame.

1. DataFrame Menggunakan List

DataFrame

```
In [4]: import pandas as pd  
data = [1,2,3,4]  
df = pd.DataFrame(data)  
df
```

```
Out[4]:
```

	0
0	1
1	2
2	3
3	4

Gambar 8 DataFrame Menggunakan List

2. DataFrame Menggunakan Dictionary

```
In [7]: dictionary = {'buah ':['Apel','Jeruk','Lemon'],'jumlah':[10,5,12]}  
df = pd.DataFrame(dictionary)  
df
```

```
Out[7]:
```

	buah	jumlah
0	Apel	10
1	Jeruk	5
2	Lemon	12

```
In [ ]:
```

Gambar 9 DataFrame Menggunakan Dictionary

PYTHON

3. DataFrame Menggunakan List dengan Tipe Data Campuran

DataFrame List Tipe Data Campuran

```
In [2]: import pandas as pd
data = [['Berti', 90, 85, 95, 90.5],
        ['Qorygore', 80, 85, 90, 86.6],
        ['Bimo', 70, 75, 80, 78.5]]
index = [0, 1, 2]
kolom = ['Nama', 'Tugas', 'UTS', 'UAS', 'Rata-Rata']
df = pd.DataFrame(data, index, kolom)
df
```

```
Out[2]:
```

	Nama	Tugas	UTS	UAS	Rata-Rata
0	Berti	90	85	95	90.5
1	Qorygore	80	85	90	86.6
2	Bimo	70	75	80	78.5

Gambar 10 DataFrame Menggunakan List dengan Tipe Data Campuran

4. DataFrame Menggunakan List & Dictionary dengan Tipe Data Campuran

```
In [4]: # atau dengan Dictionary
import pandas as pd
nama = ['Berti', 'Qorygore', 'Bimo']
tugas = [90, 80, 70]
uts = [85, 85, 75]
uas = [95, 90, 80]
ratarata = [90.5, 86.6, 78.5]
df2 = pd.DataFrame({'Nama': nama, 'Tugas': tugas, 'UTS': uts, 'UAS': uas, 'Rata-Rata': ratarata})
df2
```

```
Out[4]:
```

	Nama	Tugas	UTS	UAS	Rata-Rata
0	Berti	90	85	95	90.5
1	Qorygore	80	85	90	86.6
2	Bimo	70	75	80	78.5

Gambar 11 DataFrame Menggunakan List & Dict dengan Tipe Data Campuran

2.2 Merge, Join, & Concatenate

Ketiga fungsi ini pengertiannya adalah operasi penggabungan. Perbedaannya adalah sebagai berikut.

PYTHON

2.2.1 Merge

Merge adalah operasi penggabungan antara DataFrame Objects.

1. Siapkan 2 Data

```
In [19]: # DATA PERTAMA
import pandas as pd
nama = ['Berti', 'Ryndes', 'Arin']
tugas = [95, 90, 75]
jurusan = ['IF', 'SI', 'KA']
df3 = pd.DataFrame({'Nama': nama, 'Tugas': tugas, 'Jurusan': jurusan})
df3
```

```
Out[19]:
```

	Nama	Tugas	Jurusan
0	Berti	95	IF
1	Ryndes	90	SI
2	Arin	75	KA

```
In [20]: # DATA KEDUA
nama = ['Berti', 'Ryndes', 'Rylo']
uts = [85, 84, 70]
jurusan = ['IF', 'SI', 'SI']
df4 = pd.DataFrame({'Nama': nama, 'UTS': uts, 'Jurusan': jurusan})
df4
```

```
Out[20]:
```

	Nama	UTS	Jurusan
0	Berti	85	IF
1	Ryndes	84	SI
2	Rylo	70	SI

Gambar 12 Dua DataFrame untuk di-Merge

2. Inner Merge

```
In [21]: # INNER MERGE
df3.merge(df4)
```

```
Out[21]:
```

	Nama	Tugas	Jurusan	UTS
0	Berti	95	IF	85
1	Ryndes	90	SI	84

Gambar 13 Inner Merge

PYTHON

3. Left Merge

```
In [23]: # LEFT MERGE
df3.merge(df4, on='Nama', how='left')
```

```
Out[23]:
```

	Nama	Tugas	Jurusan_x	UTS	Jurusan_y
0	Berti	95	IF	85.0	IF
1	Ryndes	90	SI	84.0	SI
2	Arin	75	KA	NaN	NaN

Gambar 14 Left Merge

4. Right Merge

```
In [28]: # RIGHT MERGE
df3.merge(df4, on='Nama', how='right')
```

```
Out[28]:
```

	Nama	Tugas	Jurusan_x	UTS	Jurusan_y
0	Berti	95.0	IF	85	IF
1	Ryndes	90.0	SI	84	SI
2	Rylo	NaN	NaN	70	SI

Gambar 15 Right Merge

5. Outer Merge

```
In [29]: # OUTER MERGE
df3.merge(df4, on='Nama', how='outer')
```

```
Out[29]:
```

	Nama	Tugas	Jurusan_x	UTS	Jurusan_y
0	Berti	95.0	IF	85.0	IF
1	Ryndes	90.0	SI	84.0	SI
2	Arin	75.0	KA	NaN	NaN
3	Rylo	NaN	NaN	70.0	SI

Gambar 16 Outer Merge

PYTHON

2.2.2 Join

Join adalah operasi penggabungan dengan menggunakan index.

1. Siapkan 2 Data

```
In [30]: # DATA PERTAMA
nama = ['Berti', 'Ryndes', 'Arin']
tugas = [95, 90, 75]
jurusan = ['IF', 'SI', 'KA']
df3 = pd.DataFrame({'Nama': nama, 'Tugas': tugas, 'Jurusan': jurusan}, index=['L1', 'L2', 'L3'])
df3
```

```
Out[30]:
```

	Nama	Tugas	Jurusan
L1	Berti	95	IF
L2	Ryndes	90	SI
L3	Arin	75	KA

```
In [31]: # DATA KEDUA
nama = ['Berti', 'Ryndes', 'Rylo']
uts = [85, 84, 70]
jurusan = ['IF', 'SI', 'SI']
df4 = pd.DataFrame({'Nama B': nama, 'UTS': uts, 'Jurusan B': jurusan}, index=['L2', 'L3', 'L4'])
df4
```

```
Out[31]:
```

	Nama B	UTS	Jurusan B
L2	Berti	85	IF
L3	Ryndes	84	SI
L4	Rylo	70	SI

Gambar 17 Dua DataFrame untuk di-Join

2. Inner Join

```
In [33]: # INNER JOIN
df3.join(df4, how='inner')
```

```
Out[33]:
```

	Nama	Tugas	Jurusan	Nama B	UTS	Jurusan B
L2	Ryndes	90	SI	Berti	85	IF
L3	Arin	75	KA	Ryndes	84	SI

Gambar 18 Inner Join

PYTHON

3. Left Join

```
In [34]: # LEFT JOIN
df3.join(df4, how='left')
```

Out[34]:

	Nama	Tugas	Jurusan	Nama B	UTS	Jurusan B
L1	Berti	95	IF	NaN	NaN	NaN
L2	Ryndes	90	SI	Berti	85.0	IF
L3	Arin	75	KA	Ryndes	84.0	SI

Gambar 19 Left Join

4. Right Join

```
In [35]: # RIGHT JOIN
df3.join(df4, how='right')
```

Out[35]:

	Nama	Tugas	Jurusan	Nama B	UTS	Jurusan B
L2	Ryndes	90.0	SI	Berti	85	IF
L3	Arin	75.0	KA	Ryndes	84	SI
L4	NaN	NaN	NaN	Rylo	70	SI

Gambar 20 Right Join

5. Outer Join

```
In [36]: # OUTER JOIN
df3.join(df4, how='outer')
```

Out[36]:

	Nama	Tugas	Jurusan	Nama B	UTS	Jurusan B
L1	Berti	95.0	IF	NaN	NaN	NaN
L2	Ryndes	90.0	SI	Berti	85.0	IF
L3	Arin	75.0	KA	Ryndes	84.0	SI
L4	NaN	NaN	NaN	Rylo	70.0	SI

Gambar 21 Outer Join

PYTHON

2.2.3 Concatenate

Concatenate adalah operasi penggabungan objek DataFrame secara vertical.

```
In [42]: pd.concat([df3,df4], sort=False')
```

Out[42]:

	Jurusan	Jurusan B	Nama	Nama B	Tugas	UTS
L1	IF	NaN	Berti	NaN	95.0	NaN
L2	SI	NaN	Ryndes	NaN	90.0	NaN
L3	KA	NaN	Arin	NaN	75.0	NaN
L2	NaN	IF	NaN	Berti	NaN	85.0
L3	NaN	SI	NaN	Ryndes	NaN	84.0
L4	NaN	SI	NaN	Rylo	NaN	70.0

Gambar 22 Concatenate