

Hi All!



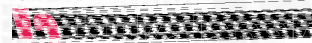
M Ammar Fauzan

Business Intelligence Mgr

EVERM*S



Sistem Informasi
(2014-2018)



Data Scientist
(Sept 2018- August 2020)



Business Intelligence .Mgr
(Sept 2020- Present)



Muhammad Ammar Fauzan

<https://www.linkedin.com/in/muhammad-ammar-fauzan-748883117/>

Python Programming



Topik *Python Programming*

Topik 1 : *Conditional Statement*

Topik 2 : *Loop and Iteration*

Topik 3 : *Function and Error Handling*



Control Flow adalah urutan bagaimana suatu kode program dijalankan / di *execute*. ***Control Flow*** pada Bahasa Pemrograman Python diatur oleh ***Conditional Statement***, ***Iteration***, dan ***Function***

Topik *Python Programming*

Topik 1 : *Conditional Statement*

Topik 2 : *Loop and Iteration*

Topik 3 : *Function and Error Handling*

Hands-On Requirement:

Hands - On :

Python Programming 1 - Rakamin Academy.ipynb

Dataset :

monthly_rakamin_customer_order.csv

**Klik disini untuk mengakses
folder Hands-On dan
Dataset**

Conditional Statement

- ☐ Pengenal Conditional Statement
- ☐ One liner dan Multi Conditional Statement
- ☐ Pembahasan Mini Quiz #1
- ☐ Pengenalan Nested Conditional Statement

- ☐ Nested Conditional Statement Part 2
- ☐ Pembahasan Mini Quiz #2
- ☐ Penerapan Conditional Statement pada Daraframe
- ☐ Pembahasan Mini Quiz #3

Conditional Statement



Pengenal Conditional Statement



One liner dan Multi Conditional Statement



Pembahasan Mini Quiz #1



Pengenalan Nested Conditional Statement



Nested Conditional Statement Part 2



Pembahasan Mini Quiz #2



Penerapan Conditional Statement pada Daraframe



Pembahasan Mini Quiz #3

Pengenalan *Conditional Statement*

Contoh DataFrame

Customer Rakamin Store

	order_month	full_name	user_id	phone	city	province	email	total_order	total_gmv	total_visit
332	2022-06-01	Faruq Adi	10000030	628567812319	Makassar	Sulawesi Selatan	faruq@yahoo.com	201	5490000	773
333	2022-06-01	Abrar Adi	10000053	628223411218	Surabaya	Jawa Timur	abrar.adi@yahoo.co.id	110	2987500	276
334	2022-06-01	Johan Imawan	10000054	628223412323	Balikpapan	Kalimantan Timur	johan.imawan@yahoo.com	95	2705000	258
335	2022-06-01	Muhammad Ammar	10000055	628223412111	Surabaya	Jawa Timur	muh.ammar@gmail.com	30	760000	103
336	2022-05-01	Dodo Putro Aji	10000023	628567812312	Surabaya	Jawa Timur	dodo@gmail.com	15	605000	83

Contoh kasus: Dari data **total order bulanan customer** team bisnis ingin membaginya menjadi 2 group / segment

Contoh DataFrame

Customer Rakamin Store



Contoh DataFrame

Customer Rakamin Store

order_month	full_name	user_id	phone	city	province	email	total_order	total_gmv	total_visit	group_order
2022-06-01	Faruq Adi	10000030	628567812319	Makassar	Sulawesi Selatan	faruq@yahoo.com	201	5490000	773	High
2022-06-01	Abrar Adi	10000053	628223411218	Surabaya	Jawa Timur	abrar.adi@yahoo.co.id	110	2987500	276	High
2022-06-01	Johan Imawan	10000054	628223412323	Balikpapan	Kalimantan Timur	johan.imawan@yahoo.com	95	2705000	258	High
2022-06-01	Muhammad Ammar	10000055	628223412111	Surabaya	Jawa Timur	muh.ammar@gmail.com	30	760000	103	Low
2022-05-01	Dodo Putro Aji	10000023	628567812312	Surabaya	Jawa Timur	dodo@gmail.com	15	605000	83	Low

Dengan implementasi **Conditional Statement**, kita bisa membuat 1 Kolom baru berdasarkan **kategori total order bulanan** based on kriteria / persyaratan team bisnis.

“Conditional Statement adalah suatu logic yang menjalankan suatu tindakan jika suatu kondisi (expressions) terpenuhi. Logic ini menjadi penting ketika kita ingin mengolah data berdasarkan beberapa kondisi / persyaratan.”

Implementasi Conditional Statement

Contoh Code:

```

1  if expression1:
2      statement1(s)
3  elif expression2:
4      statement2(s)
5  elif expression3:
6      statement3(s)
7  ...
8  else:
9      statement4(s)

```

Implementasi source code Conditional Statement

- Header line pada *conditional statement* diawali dengan *syntax if* diikuti dengan *expression* umumnya ber type *boolean* dan *titik dua* :

Implementasi Conditional Statement

Contoh Code:

```

1  if expression1:
2      statement1(s)
3  elif expression2:
4      statement2(s)
5  elif expression3:
6      statement3(s)
7  ...
8  else:
9      statement4(s)

```

Implementasi source code Conditional Statement

- Header line pada *conditional statement* diawali dengan *syntax if* diikuti dengan **expression umumnya** ber type **boolean** dan *titik dua* :
- Jika kondisi terpenuhi pada expression1 maka akan melakukan statement1, **Namun jika tidak terpenuhi akan masuk ke expression2 (elif)** dan seterusnya (atas ke bawah) hingga pengecekan terakhir pada **else**

Implementasi Conditional Statement

Contoh Code:

```

1  if expression1:
2      statement1(s)
3  elif expression2:
4      statement2(s)
5  elif expression3:
6      statement3(s)
7  ...
8  else:
9      statement4(s)

```

Implementasi source code Conditional Statement

- Header line pada *conditional statement* diawali dengan *syntax if* diikuti dengan **expression umumnya** ber type **boolean** dan *titik dua* :
- Jika kondisi terpenuhi pada expression1 maka akan melakukan statement1, **Namun jika tidak terpenuhi akan masuk ke expression2 (elif)** dan seterusnya (atas ke bawah) hingga pengecekan terakhir pada **else**
- Jika **sudah memenuhi suatu expression** tertentu maka kode program akan selesai (keluar dari conditional statement)
- **Statement(s)** bisa lebih dari 1

Implementasi Conditional Statement

Contoh Code:

```
1 if expression1:  
2     statement1(s)  
3 elif expression2:  
4     statement2(s)  
5 elif expression3:  
6     statement3(s)  
7 ...  
8 else:  
9     statement4(s)
```

Elif dan Else
Sifat nya
Optional

Contoh Penggunaan IF 1 kondisi #1

Misalkan dalam **transaksi online**, ketika pesanan sedang dikirim ke pelanggan maka statusnya adalah on-delivery. Kemudian status akan berubah menjadi delivered jika pembeli telah menerima barang.



Contoh Penggunaan IF 1 kondisi #1

From this

order_id	status	barang_diterima
1	on-delivery	False
2	on-delivery	True
3	on-delivery	False
4	on-delivery	True



To this

order_id	status
1	on-delivery
2	delivered
3	on-delivery
4	delivered

[Pengiriman] Implementasi IF - 1 Kondisi

Contoh Code:

```
1 current_status = 'on-delivery'
2 barang_diterima = True
3
4 if barang_diterima==True: #expression 'jika' berwujud nilai boolean
5     current_status='delivered' #jika true maka. . .
6
7 print(current_status)
```

Pada contoh Conditional Statement di atas:

- Kita membuat conditional statement 1 kondisi, yang hanya merubah status jika **barang_diterima** bernilai True, **namun jika tidak memenuhi** tidak ada yang diharapkan terjadi.
- Operasi **==** artinya membandingkan (TRUE/FALSE), **berbeda dengan =**

[Pengiriman] Implementasi IF - 1 Kondisi

Contoh Code:

```

1 current_status = 'on-delivery'
2 barang_diterima = True
3
4 if barang_diterima==True: #expression 'jika' berwujud nilai boolean
5     current_status='delivered' #jika true maka. . .
6
7 print(current_status)

```

delivered

Pada contoh Conditional Statement di atas:

- Kita membuat conditional statement 1 kondisi, yang hanya merubah status jika **barang_diterima** bernilai True, **namun jika tidak memenuhi** tidak ada yang diharapkan terjadi.
- Operasi **==** artinya membandingkan (TRUE/FALSE), **berbeda dengan =**

[Pengiriman] Implementasi IF - 1 Kondisi

Contoh Code:

```
1 current_status = 'on-delivery'
2 barang_diterima = True
3
4 if barang_diterima==True: #expression 'jika' berwujud nilai boolean
5     current_status='delivered' #jika true maka. . .
6
7 print(current_status)
```

delivered



```
1 #implementasi if pada python programming
2 current_status = 'on-delivery'
3 barang_diterima = True
4
5 if barang_diterima==True: #expression
6     current_status='delivered' #jika true maka.
7 else:
8     pass
9
10 print(current_status)
```

delivered

Contoh Penggunaan IF 2 kondisi #2

Mengkategorikan visitor e-commerce dengan persyaratan, **jika jumlah kunjungan (visit) lebih dari 30 kali setiap bulan** maka dikategorikan **potential buyer**, jika kurang dari itu maka **not potential buyer**



Contoh Penggunaan IF 2 kondisi #2

From this

user_id	jumlah_visit
1	32
2	24
3	50
4	8



To this

user_id	potential_buyer
1	True
2	False
3	True
4	False

[Potential Buyer] Implementasi IF - 2 Kondisi

Contoh Code:

```

1 total_visit = 34
2
3 if total_visit > 30: #expression melakukan pengecekan dengan value True / False, jika True akan masuk ke kondisi maka
4     category = 'Potential Buyer' #jika True maka category akan bernilai Potential Buyer
5 else: #jika expression awal False maka akan masuk ke kondisi selanjutnya / akhir
6     category = 'Not Potential Buyer'
7
8 print(category)

```

Pada contoh Conditional Statement di atas:

Kita membuat **conditional statement 2 kondisi (if-else)**, yang akan mengkategorikan user **jika total visit nya lebih dari (>) 30** maka digolongkan **Potential Buyer** namun jika tidak **Not Potential Buyer**

[Potential Buyer] Implementasi IF - 2 Kondisi

Contoh Code:

```

1 total_visit = 34
2
3 if total_visit > 30: #expression melakukan pengecekan dengan value True / False, jika True akan masuk ke kondisi maka
4     category = 'Potential Buyer' #jika True maka category akan bernilai Potential Buyer
5 else: #jika expression awal False maka akan masuk ke kondisi selanjutnya / akhir
6     category = 'Not Potential Buyer'
7
8 print(category)

```

Potential Buyer

Pada contoh Conditional Statement di atas:

Kita membuat **conditional statement 2 kondisi (if-else)**, yang akan mengkategorikan user **jika total visit nya lebih dari (>) 30** maka digolongkan **Potential Buyer** namun jika tidak **Not Potential Buyer**

Conditional Statement



Pengenal Conditional Statement



One liner dan Multi Conditional Statement



Pembahasan Mini Quiz #1



Pengenalan Nested Conditional Statement



Nested Conditional Statement Part 2



Pembahasan Mini Quiz #2



Penerapan Conditional Statement pada Daraframe



Pembahasan Mini Quiz #3

Conditional Statement



Pengenal Conditional Statement



One liner dan Multi Conditional Statement



Pembahasan Mini Quiz #1



Pengenalan Nested Conditional Statement



Nested Conditional Statement Part 2



Pembahasan Mini Quiz #2



Penerapan Conditional Statement pada Daraframe



Pembahasan Mini Quiz #3



Kira-kira bisa **lebih dipersingkat** lagi gak ya syntax nya?

```
1 rating = 3
2
3 if rating >= 4:
4     result = 'Puas'
5 else:
6     result = 'Tidak Puas'
7
8 print(result)
```

Tidak Puas



Implementasi **IF ELSE ONE LINER** pada Python (*Python's Ternary Operator*)

Contoh Code

```
1 rating = 2
2
3 result = 'Puas' if rating >= 4 else 'Tidak Puas'
4 print(result)
```

Tidak Puas

=

```
1 rating = 2
2
3 if rating >= 4:
4     result = 'Puas'
5 else:
6     result = 'Tidak Puas'
7
8 print(result)
```

Tidak Puas

Implementasi **IF ELSE ONE LINER** pada Python (*Python's Ternary Operator*)

Contoh Code

```
1 rating = 2
2
3 result = 'Puas' if rating >= 4 else 'Tidak Puas'
4 print(result)
```

Tidak Puas

=

```
1 rating = 2
2
3 if rating >= 4:
4     result = 'Puas'
5 else:
6     result = 'Tidak Puas'
7
8 print(result)
```

Tidak Puas



Notes:

Kelebihan *One liner conditional statement* **lebih pendek source codenya** namun ada kekurangan dari sisi **tingkat readability (kemudahan dibaca) semakin rendah**.

Implementasi **IF ELSE ONE LINER** pada Python (*Python's Ternary Operator*)

Contoh Code

```
1  #one liner conditional statement
2
3  result = <Nilai True> if expression else <Nilai False>
```

Implementasi **IF ELSE ONE LINER** pada Python

Contoh Code

```

1 rating = 2
2
3 result = 'Puas' if rating >= 4 else 'Tidak Puas'
4 print(result)

```

Tidak Puas

True Value

False Value

Implementasi Condition Statement

Banyak Kondisi

Contoh Kasus: Mengelompokkan usia menjadi beberapa kelompok



Implementasi Condition Statement

Banyak Kondisi

Mengelompokkan usia menjadi beberapa kelompok

usia	kelompok
0-5	Balita
6-12	Anak-anak
13-22	Remaja
23-60	Dewasa
>60	Lansia

Menggunakan konsep pernyataan kondisi Jika ... Maka ...
berganda

Implementasi Condition Statement

Banyak Kondisi

Contoh Code: if - elif - else

```
1  #dalam kumpulan data kependudukan kita ingin membagi menjadi beberapa kategori umur
2  usia=20
3
4  if usia<6:
5      kelompok='balita'
6  elif usia>=6 and usia<=12:
7      kelompok='anak-anak'
8  elif usia>=13 and usia<=22:
9      kelompok='remaja'
10 elif usia>=23 and usia<=60:
11     kelompok='dewasa'
12 else:
13     kelompok='lansia'
14
15 print(kelompok)
```

Implementasi Condition Statement

Banyak Kondisi

Contoh Code: if - elif - else

```
1  #dalam kumpulan data kependudukan kita ingin membagi menjadi beberapa kategori umur
2  usia=20
3
4  if usia<6:
5      kelompok='balita'
6  elif usia>=6 and usia<=12:
7      kelompok='anak-anak'
8  elif usia>=13 and usia<=22:
9      kelompok='remaja'
10 elif usia>=23 and usia<=60:
11     kelompok='dewasa'
12 else:
13     kelompok='lansia'
14
15 print(kelompok)
```

remaja

Implementasi Condition Statement

Banyak Kondisi

Contoh Code: if - elif - else

```

1  #dalam kumpulan data kependudukan kita ingin membagi menjadi beberapa kategori umur
2  usia=20
3
4  if usia<6:
5      kelompok='balita'
6  elif usia>=6 and usia<=12:
7      kelompok='anak-anak'
8  elif usia>=13 and usia<=22:
9      kelompok='remaja'
10 elif usia>=23 and usia<=60:
11     kelompok='dewasa'
12 else:
13     kelompok='lansia'
14
15 print(kelompok)

```

remaja

Notes:

- Jika sudah memenuhi suatu kondisi, maka kode program conditional statement sudah **selesai** dijalankan
- Dikarenakan sesuai dengan kondisi usia >= 13 dan usia <= 22 Maka value variabel kelompok menjadi **'remaja'**

Implementasi Condition Statement

Banyak Kondisi

Perbandingan logic expression namun memiliki hasil yang sama

Versi 1

```

3  usia=20
4
5  if usia<6:
6      kelompok='balita'
7  elif usia>=6 and usia<=12:
8      kelompok='anak-anak'
9  elif usia>=13 and usia<=22:
10     kelompok='remaja'
11  elif usia>=23 and usia<=60:
12     kelompok='dewasa'
13  else:
14     kelompok='lansia'
15
16  print(kelompok)

```

remaja

Versi 2

```

3  usia=20
4
5  if usia<6:
6      kelompok='balita'
7  elif usia<=12:
8      kelompok='anak-anak'
9  elif usia<=22:
10     kelompok='remaja'
11  elif usia<=60:
12     kelompok='dewasa'
13  else:
14     kelompok='lansia'
15
16  print(kelompok)

```

remaja

Pentingnya Menggunakan “ELIF”

Mengapa hasilnya berbeda ?

```

1  usia=5
2
3  if usia<6:
4      kelompok_usia='balita'
5  elif usia<=12:
6      kelompok_usia='anak-anak'
7
8  kelompok_usia

```

'balita'

VS

```

1  usia=5
2
3  if usia<6:
4      kelompok_usia='balita'
5  if usia<=12:
6      kelompok_usia='anak-anak'
7
8  kelompok_usia

```

'anak-anak'

Notes: elif merupakan kependekan dari **else if**

Pentingnya Menggunakan “ELIF”

Mengapa hasilnya berbeda ?

```

1  usia=5
2
3  if usia<6:
4      kelompok_usia='balita'
5  elif usia<=12:
6      kelompok_usia='anak-anak'
7
8  kelompok_usia

```

'balita'

If elif **satu kesatuan**

VS

```

1  usia=5
2
3  if usia<6:
4      kelompok_usia='balita'
5  if usia<=12:
6      kelompok_usia='anak-anak'
7
8  kelompok_usia

```

'anak-anak'

If pada baris kode ke 3 dan 4 **terpisah** dengan baris kode ke 5 dan 6.

Sehingga hasil pada baris kode ke 4 ter replace dengan baris kode ke 6

Conditional Statement



Pengenal Conditional Statement



One liner dan Multi Conditional Statement



Pembahasan Mini Quiz #1



Pengenalan Nested Conditional Statement



Nested Conditional Statement Part 2



Pembahasan Mini Quiz #2



Penerapan Conditional Statement pada Daraframe



Pembahasan Mini Quiz #3

Conditional Statement



Pengenal Conditional Statement



One liner dan Multi Conditional Statement



Pembahasan Mini Quiz #1



Pengenalan Nested Conditional Statement



Nested Conditional Statement Part 2



Pembahasan Mini Quiz #2



Penerapan Conditional Statement pada Daraframe



Pembahasan Mini Quiz #3

1. Buatlah **syntax IF** untuk melakukan pengecekan apakah **plat nomor kendaraan B 1256 AE masuk plat genap atau ganjil !**
2. Buatlah **syntax IF** untuk melakukan pengecekan apakah **variabel kategori1 ada di dalam list kategori**, dimana
 - **kategori1** = 'Fashion Pria'
 - **list_kategori** = ['Handphone', 'Sepeda', 'Makanan', 'Fashion Wanita']

Conditional Statement



Pengenal Conditional Statement



One liner dan Multi Conditional Statement



Pembahasan Mini Quiz #1



Pengenalan Nested Conditional Statement



Nested Conditional Statement Part 2



Pembahasan Mini Quiz #2



Penerapan Conditional Statement pada Daraframe



Pembahasan Mini Quiz #3

Penerapan ***Nested Conditional Statement***



Nested Conditional Statement adalah pernyataan yang memuat beberapa ***persyaratan bertingkat (if di dalam if)***



Nested Conditional Statement adalah pernyataan yang memuat beberapa ***persyaratan bertingkat (if di dalam if)***

Biasanya nested conditional statement digunakan ketika kita memiliki 2 atau banyak context kondisi (expression) yang berbeda namun saling berkaitan atau memiliki tujuan tertentu.



Nested Conditional Statement adalah pernyataan yang memuat beberapa ***persyaratan bertingkat (if di dalam if)***

```

1 #Concept Nested Conditional Statements
2 #contoh if di dalam if
3 if expression1:
4     if expression11:
5 else:
6     statement2(s)
7

```

Contoh kasus: Pemberian Point Java dan Non Java

Contoh dataset

user_id	is_java	Order
1	TRUE	101
2	TRUE	10
3	FALSE	51
4	FALSE	25
5	TRUE	700
6	FALSE	60
7	TRUE	120
8	FALSE	25

Logic Bisnis

is_java	Order	Action
TRUE	> 100	maka tambah 3
	<= 100	maka tambah 1
FALSE	> 50	maka tambah 3
	<= 50	maka tambah 1

- **Terdapat 2 dasar pemberian point:**
 - Based on wilayah (java / non java)
 - Based on performance total order

Contoh kasus: Pemberian Point Java dan Non Java

user_id	is_java	Order	Action
1	TRUE	101	3
2	TRUE	10	1
3	FALSE	51	3
4	FALSE	25	1
5	TRUE	700	3
6	FALSE	60	3
7	TRUE	120	3
8	FALSE	25	1

Logic Bisnis

is_java	Order	Action
TRUE	> 100	maka tambah 3
	<= 100	maka tambah 1
FALSE	> 50	maka tambah 3
	<= 50	maka tambah 1

Contoh kasus: Pemberian Point Java dan Non Java

Contoh Code:

```

1  #nested conditional statement - pemberian point
2  is_java = True
3  total_order = 101
4
5  if is_java==True: #context pembagian java dan non java
6      if total_order > 100: #context performance order
7          action = 3
8      else:
9          action = 1
10 else: #context pembagian java dan non java
11     if total_order > 50: #context performance order
12         action = 3
13     else:
14         action = 1
15
16 print("Action penambahan", action, "point")

```

Action penambahan 3 point

Contoh kasus: Pemberian Point Java dan Non Java

Contoh Code:

```

1  #nested conditional statement - pemberian point
2  is_java = True
3  total_order = 101
4
5  if is_java==True: #context pembagian java dan non java
6      if total_order > 100: #context performance order
7          action = 3
8      else:
9          action = 1
10 else: #context pembagian java dan non java
11     if total_order > 50: #context performance order
12         action = 3
13     else:
14         action = 1
15
16 print("Action penambahan", action, "point")

```

Action penambahan 3 point



Salah satu manfaat penggunaan nested conditional statement adalah agar mempermudah kode program dipahami alur prosesnya (readability)

Conditional Statement



Pengenal Conditional Statement



One liner dan Multi Conditional Statement



Pembahasan Mini Quiz #1



Pengenalan Nested Conditional Statement



Nested Conditional Statement Part 2



Pembahasan Mini Quiz #2



Penerapan Conditional Statement pada Daraframe



Pembahasan Mini Quiz #3

Conditional Statement



Pengenal Conditional Statement



One liner dan Multi Conditional Statement



Pembahasan Mini Quiz #1



Pengenalan Nested Conditional Statement



Nested Conditional Statement Part 2



Pembahasan Mini Quiz #2



Penerapan Conditional Statement pada Daraframe



Pembahasan Mini Quiz #3

Contoh kasus 2: ketika kita memiliki kondisi kelulusan ujian berdasarkan 2 persyaratan (syarat 1 dan syarat 2)

Persyaratan lolos Ujian :

- **Syarat 1** : total nilai lebih dari 140
- **Syarat 2** : tidak ada nilai dibawah 50



user_id	nilai X1	nilai X2
1	80	90
2	100	45
3	75	60
4	70	70

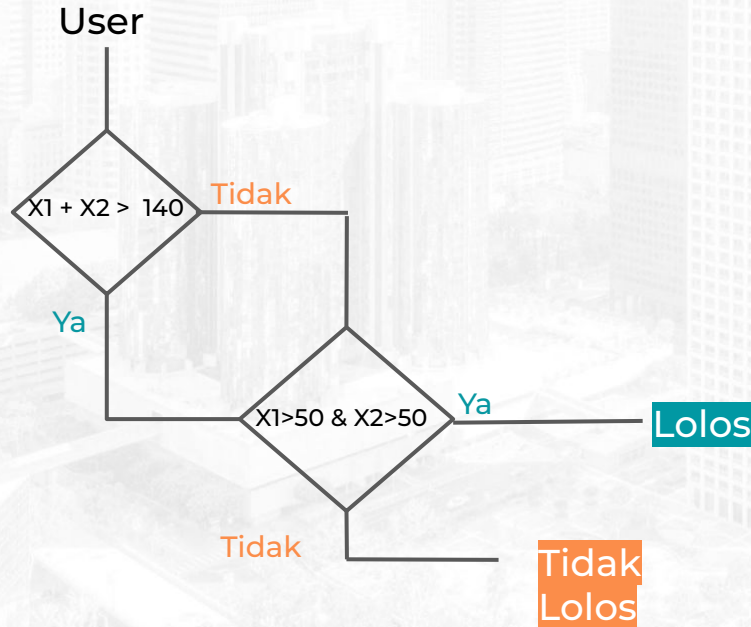
Contoh kasus 2: ketika kita memiliki kondisi kelulusan ujian berdasarkan 2 persyaratan (syarat 1 dan syarat 2)

Contoh Menilai Kelulusan:

user_id	nilai X1	nilai X2	syarat 1	syarat 2	kesimpulan
1	80	90	lolos	lolos	lolos
2	100	45	lolos	tidak lolos	tidak lolos
3	70	65	tidak lolos	lolos	tidak lolos
4	75	70	lolos	lolos	lolos

Contoh kasus 2: ketika kita memiliki kondisi kelulusan ujian berdasarkan 2 persyaratan (syarat 1 dan syarat 2)

Flow Menilai Kelulusan




```

1 nilai_x1=100
2 nilai_x2=50
3
4 total_nilai = nilai_x1 + nilai_x2 #syarat1
5 x1_pass = nilai_x1 > 50 #syarat2 part1
6 x2_pass = nilai_x2 > 50 #syarat2 part2
7
8 if total_nilai>140:
9     syarat1 = 'lolos'
10    if x1_pass and x2_pass:
11        syarat2 = 'lolos'
12        result = 'lolos'
13    else:
14        syarat2 = 'tidak lolos'
15        result = 'tidak lolos'
16 else:
17    syarat1 = 'tidak lolos'
18    if x1_pass and x2_pass:
19        syarat2 = 'lolos'
20        result = 'tidak lolos'
21    else:
22        syarat2 = 'tidak lolos'
23        result = 'tidak lolos'
24
25 print('syarat 1 =', syarat1)
26 print('syarat 2 =', syarat2)
27 print('Hasil =', result)

```

```

syarat 1 = lolos
syarat 2 = tidak lolos
Hasil = tidak lolos

```

Nested Conditional Statement

Menggunakan **If di dalam If** pada Python

Salah satu manfaat *nested condition statement* adalah kita **bisa melakukan pengecekan setiap syarat nya**

```

1 nilai_x1=100
2 nilai_x2=50
3
4 total_nilai = nilai_x1 + nilai_x2 #syarat1
5 x1_pass = nilai_x1 > 50 #syarat2 part1
6 x2_pass = nilai_x2 > 50 #syarat2 part2
7
8 if total_nilai>140:
9     syarat1 = 'lolos'
10    if x1_pass and x2_pass:
11        syarat2 = 'lolos'
12        result = 'lolos'
13    else:
14        syarat2 = 'tidak lolos'
15        result = 'tidak lolos'
16 else:
17    syarat1 = 'tidak lolos'
18    if x1_pass and x2_pass:
19        syarat2 = 'lolos'
20        result = 'tidak lolos'
21    else:
22        syarat2 = 'tidak lolos'
23        result = 'tidak lolos'
24
25 print('syarat 1 =', syarat1)
26 print('syarat 2 =', syarat2)
27 print('Hasil =', result)

```

Context / Syarat 1

Nested Conditional Statement

Menggunakan If di dalam If pada Python

Salah satu manfaat *nested condition statement* adalah kita bisa melakukan pengecekan setiap syarat nya

```

syarat 1 = lolos
syarat 2 = tidak lolos
Hasil = tidak lolos

```

```

1 nilai_x1=100
2 nilai_x2=50
3
4 total_nilai = nilai_x1 + nilai_x2 #syarat1
5 x1_pass = nilai_x1 > 50 #syarat2 part1
6 x2_pass = nilai_x2 > 50 #syarat2 part2
7
8 if total_nilai>140:
9     syarat1 = 'lolos'
10    if x1_pass and x2_pass:
11        syarat2 = 'lolos'
12        result = 'lolos'
13    else:
14        syarat2 = 'tidak lolos'
15        result = 'tidak lolos'
16 else:
17    syarat1 = 'tidak lolos'
18    if x1_pass and x2_pass:
19        syarat2 = 'lolos'
20        result = 'tidak lolos'
21    else:
22        syarat2 = 'tidak lolos'
23        result = 'tidak lolos'
24
25 print('syarat 1 =', syarat1)
26 print('syarat 2 =', syarat2)
27 print('Hasil =', result)

```

Pengecekan
Syarat 2 jika
Syarat 1 lolos

Pengecekan
Syarat 2 jika
Syarat 1 tidak
lolos

syarat 1 = lolos
syarat 2 = tidak lolos
Hasil = tidak lolos

Nested Conditional Statement

Menggunakan If di dalam If pada Python

Salah satu manfaat *nested condition statement* adalah kita **bisa melakukan pengecekan setiap syarat nya**

Alternatif - Expression Lebih Dari 1 Kondisi

Jika hanya mendapatkan result nya saja, kita bisa menggunakan 2 kondisi sekaligus dalam expression awal (**and**). Namun pada contoh dibawah kita tidak bisa mendapatkan hasil dari masing-masing syarat.

Contoh Code:

```

1  #kondisi jika langsung kita tidak bisa mengetahui syarat apa yang membuat tidak lolos
2
3  nilai_x1=100
4  nilai_x2=50
5
6  if (nilai_x1+nilai_x2>140) and (nilai_x1>50 and nilai_x2>50):
7      result = 'lolos'
8  else:
9      result = 'tidak lolos'
10
11 print('Hasil = '+result)

```

Hasil = tidak lolos

Pada expression lebih dari satu kondisi bisa menggunakan **“and”** atau **“or”**


```
3 nilai_x1=100
4 nilai_x2=50
5
6 total_nilai = nilai_x1 + nilai_x2
7 x1_pass = nilai_x1 > 50
8 x2_pass = nilai_x2 > 50
9
10 #pengecekan syarat 1
11 if total_nilai>140:
12     syarat1 = 'lolos'
13 else:
14     syarat1 = 'tidak lolos'
15
16 #pengecekan syarat 2
17 if x1_pass>50 and x2_pass>50:
18     syarat2 = 'lolos'
19 else:
20     syarat2 = 'tidak lolos'
21
22 #melakukan pengecekan hasil akhir
23 if syarat1 == 'lolos' and syarat2 =='lolos':
24     hasil = 'lolos'
25 else:
26     hasil ='tidak lolos'
27
28 print('syarat 1 =', syarat1)
29 print('syarat 2 =', syarat2)
30 print('Hasil =', hasil)|
```

syarat 1 = lolos
syarat 2 = tidak lolos
Hasil = tidak lolos

ONE BY ONE IF ELSE - Alternatif

Jika **tidak menggunakan nested conditional statement**, maka kita bisa menggunakan alternatif syntax one by one if else.

Hal ini mengharuskan setiap syarat dibuat syntax ***conditional statement* nya masing-masing**, begitu juga dengan hasil akhir


```
3 nilai_x1=100
4 nilai_x2=50
5
6 total_nilai = nilai_x1 + nilai_x2
7 x1_pass = nilai_x1 > 50
8 x2_pass = nilai_x2 > 50
9
10 #pengecekan syarat 1
11 if total_nilai>140:
12     syarat1 = 'lolos'
13 else:
14     syarat1 = 'tidak lolos'
15
16 #pengecekan syarat 2
17 if x1_pass>50 and x2_pass>50:
18     syarat2 = 'lolos'
19 else:
20     syarat2 = 'tidak lolos'
21
22 #melakukan pengecekan hasil akhir
23 if syarat1 == 'lolos' and syarat2 =='lolos':
24     hasil = 'lolos'
25 else:
26     hasil ='tidak lolos'
27
28 print('syarat 1 =', syarat1)
29 print('syarat 2 =', syarat2)
30 print('Hasil =', hasil)|
```

syarat 1 = lolos
syarat 2 = tidak lolos
Hasil = tidak lolos

ONE BY ONE IF ELSE - Alternatif

Jika **tidak menggunakan nested conditional statement**, maka kita bisa menggunakan alternatif syntax one by one if else.

Hal ini mengharuskan setiap syarat dibuat syntax ***conditional statement* nya masing-masing**, begitu juga dengan hasil akhir

Conditional Statement



Pengenal Conditional Statement



One liner dan Multi Conditional Statement



Pembahasan Mini Quiz #1



Pengenalan Nested Conditional Statement



Nested Conditional Statement Part 2



Pembahasan Mini Quiz #2



Penerapan Conditional Statement pada Daraframe



Pembahasan Mini Quiz #3

Conditional Statement



Pengenal Conditional Statement



One liner dan Multi Conditional Statement



Pembahasan Mini Quiz #1



Pengenalan Nested Conditional Statement



Nested Conditional Statement Part 2



Pembahasan Mini Quiz #2



Penerapan Conditional Statement pada Daraframe



Pembahasan Mini Quiz #3



Untuk menjadi **seller premium** seorang seller perlu memenuhi syarat-syarat berikut:

- **Berhasil menjual 1000 produk**
- **memiliki feedback positif > 600**

Buatlah syntax **nested conditional statement (if di dalam if)** yang sesuai dengan kasus tersebut. Kemudian uji coba hasil nya dengan case sebagai berikut:

- **`sold`** = 1020
- **`pos_feed`** = 575

Conditional Statement



Pengenal Conditional Statement



One liner dan Multi Conditional Statement



Pembahasan Mini Quiz #1



Pengenalan Nested Conditional Statement



Nested Conditional Statement Part 2



Pembahasan Mini Quiz #2



Penerapan Conditional Statement pada Daraframe



Pembahasan Mini Quiz #3

Penerapan *Conditional Statement* pada Dataframe

In Real Case, dataset **memiliki banyak baris dan kolom**, maka **tidak mungkin kita memberlakukan syntax If else satu persatu secara manual**. Ada beberapa metode untuk mengatasi kasus tersebut, salah satunya adalah menggunakan function **where() pada package Numpy**



Penggunaan np.where()

Misalkan kita ingin menggolongkan user kedalam kelompok **high buyer** (suka belanja) dan **low buyer** berdasarkan **total order dalam sebulan**.

Mengelompokkan user
berdasarkan Total Order
Bulanan



Contoh Kasus

Customer Rakamin Store

User dengan Total Order lebih dari 30 dalam sebulan adalah kelompok **High Buyer**, sedangkan yang dibawah itu **Low Buyer**.



Contoh Kasus

Customer Rakamin Store

```
1 import numpy as np
2 import pandas as pd
3
4 df_summarize = pd.read_csv('monthly_rakamin_customer_order.csv')
5 df_summarize.tail()
```

order_month	full_name	user_id	phone	city	province	email	total_order	total_gmv	total_visit
2022-06-01	Faruq Adi	10000030	628567812319	Makassar	Sulawesi Selatan	faruq@yahoo.com	201	5490000	773
2022-06-01	Abrar Adi	10000053	628223411218	Surabaya	Jawa Timur	abrar.adi@yahoo.co.id	110	2987500	276
2022-06-01	Johan Imawan	10000054	628223412323	Balikpapan	Kalimantan Timur	johan.imawan@yahoo.com	95	2705000	258
2022-06-01	Muhammad Ammar	10000055	628223412111	Surabaya	Jawa Timur	muh.ammar@gmail.com	30	760000	103
2022-05-01	Dodo Putro Aji	10000023	628567812312	Surabaya	Jawa Timur	dodo@gmail.com	15	605000	83

Menggunakan np.where 1 Kondisi

User dengan Total Order lebih dari 30 dalam sebulan adalah kelompok *High Buyer*, sedangkan yang dibawah itu *Low Buyer*.

Contoh Code:

```
1 #menggunakan fungsi where pada package numpy
2 df_summarize['group_order'] = np.where(df_summarize['total_order'] > 30, 'High', 'Low')
```

Menggunakan np.where 1 Kondisi

User dengan Total Order lebih dari 30 dalam sebulan adalah kelompok **High Buyer**, sedangkan yang dibawah itu **Low Buyer**.

Contoh Code:

```
1 #menggunakan fungsi where pada package numpy
2 df_summarize['group_order'] = np.where(df_summarize['total_order'] > 30, 'High', 'Low')
```

```
1 df_summarize['column_name'] = np.where(expression, value True, value False)
```

Menggunakan np.where 1 Kondisi

User dengan Total Order lebih dari 30 dalam sebulan adalah kelompok **High Buyer**, sedangkan yang dibawah itu **Low Buyer**.

order_month	full_name	user_id	phone	city	province	email	total_order	total_gmv	total_visit	group_order
2018-01-01	Ahmad Budiman	10000011	628785222211	Jakarta Pusat	DKI Jakarta	ahmad@gmail.com	125	3490000	230	High
2018-01-01	Azzam Haqq	10000012	628785222212	Surabaya	Jawa Timur	azzam@gmail.co.id	173	4757500	304	High
2018-01-01	Amrina Putri	10000013	628785222213	Jakarta Pusat	DKI Jakarta	amrina@yahoo.co.id	20	550000	62	Low
2018-01-01	Aan Utama	10000014	628785222214	Medan	Sumatra Utara	aan@yahoo.com	18	495000	100	Low
2018-01-01	Adam Adi	10000015	628785222215	Jakarta Timur	DKI Jakarta	adam@gmail.com	159	4372500	524	High

Menggunakan np.where 2 Kondisi

Pada beberapa kasus, **kita memiliki lebih dari 2 kondisi**, maka function **np.where()** ini juga memungkinkan untuk kita manfaatkan.

Contoh Code:

```
1 #menggunakan fungsi where pada package numpy multi
2 df_summarize['group_order_multi'] = np.where(df_summarize['total_order'] > 100, 'High',
3                                             np.where(df_summarize['total_order'] > 30, 'Mid', 'Low'))
```

Idenya adalah kita **menggunakan np.where didalam np.where (nested)**

Menggunakan np.where 2 Kondisi

Pada beberapa kasus, **kita memiliki lebih dari 2 kondisi**, maka function **np.where()** ini juga memungkinkan untuk kita manfaatkan.

Contoh Code:

```

1  #menggunakan fungsi where pada package numpy multi
2  df_summarize['group_order_multi'] = np.where(df_summarize['total_order'] > 100, 'High',
3                                              np.where(df_summarize['total_order'] > 30, 'Mid', 'Low'))

```

True

'High'

Idenya adalah kita **menggunakan np.where didalam np.where (nested)**

Menggunakan np.where 2 Kondisi

Pada beberapa kasus, **kita memiliki lebih dari 2 kondisi**, maka function **np.where()** ini juga memungkinkan untuk kita manfaatkan.

Contoh Code:

```
1 #menggunakan fungsi where pada package numpy multi
2 df_summarize['group_order_multi'] = np.where(df_summarize['total_order'] > 100, 'High',
3 np.where(df_summarize['total_order'] > 30, 'Mid', 'Low'))
```

True

'High'

False

Idenya adalah kita **menggunakan np.where didalam np.where (nested)**

Menggunakan np.where 2 Kondisi

Pada beberapa kasus, **kita memiliki lebih dari 2 kondisi**, maka function **np.where()** ini juga memungkinkan untuk kita manfaatkan.

full_name	user_id	phone	city	province	email	total_order	total_gmv	total_visit	group_order_multi
Ahmad Budiman	10000011	628785222211	Jakarta Pusat	DKI Jakarta	ahmad@gmail.com	125	3490000	230	High
Azzam Haqq	10000012	628785222212	Surabaya	Jawa Timur	azzam@gmail.co.id	173	4757500	304	High
Amrina Putri	10000013	628785222213	Jakarta Pusat	DKI Jakarta	amrina@yahoo.co.id	20	550000	62	Low
Aan Utama	10000014	628785222214	Medan	Sumatra Utara	aan@yahoo.com	18	495000	100	Low
Adam Adi	10000015	628785222215	Jakarta Timur	DKI Jakarta	adam@gmail.com	159	4372500	524	High
Budi Utomo	10000016	628785222216	Bandung	Jawa Barat	budi@gmail.co.id	138	3795000	345	High
Bayu Aji	10000017	628785222217	Semarang	Jawa Tengah	bayu@yahoo.co.id	78	2145000	194	Mid

Conditional Statement



Pengenal Conditional Statement



One liner dan Multi Conditional Statement



Pembahasan Mini Quiz #1



Pengenalan Nested Conditional Statement



Nested Conditional Statement Part 2



Pembahasan Mini Quiz #2



Penerapan Conditional Statement pada Daraframe



Pembahasan Mini Quiz #3

Conditional Statement



Pengenal Conditional Statement



One liner dan Multi Conditional Statement



Pembahasan Mini Quiz #1



Pengenalan Nested Conditional Statement



Nested Conditional Statement Part 2



Pembahasan Mini Quiz #2



Penerapan Conditional Statement pada Daraframe



Pembahasan Mini Quiz #3



Menggunakan dataset **monthly_rakamin_customer_order**,
tambahkan kolom **is_higher_gmv** dengan logik jika

- **Total GMV** lebih besar dari rata-rata GMV bulanan customer maka **True** namun jika tidak maka **False**

Hint: Buat dataframe baru terlebih dulu dengan menggunakan fungsi Group by kemudian di merge / gunakan fungsi .transform()



Danke Schön!



Muhammad Ammar Fauzan

<https://www.linkedin.com/in/muhammad-ammar-fauzan-748883117/>