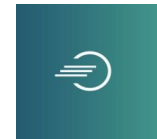
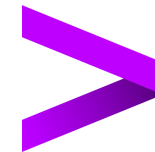


Fiqry Revadiansyah

A data scientist with 5+ years of experience

Currently working at  ajaib

Past Experiences:



 bukalapak

Fiqry Revadiansyah
Data Scientist

Profile & Portfolios



LINKEDIN



SPEAKERDECK

Python Intro & Data Types

(List and NumPy)



List & NumPy

☐ Pengenalan Tipe Data List

☐ Operasi pada List

☐ Pengenalan Tipe Data
Tuple dan Set

☐ Operasi pada Tuple dan Set

☐ Pengenalan NumPy

☐ Operasi pada NumPy

List & NumPy



Pengenalan Tipe Data List



Operasi pada List



Pengenalan Tipe Data
Tuple dan Set



Operasi pada Tuple dan Set



Pengenalan NumPy



Operasi pada NumPy

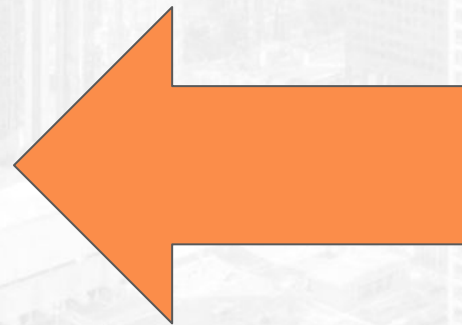
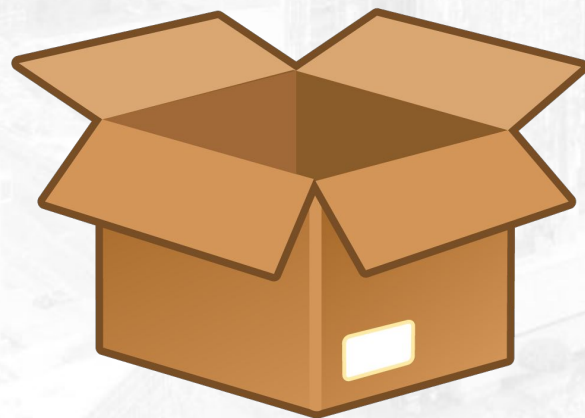
Hands-On Required :

Hands - On : 2. List & NumPy.ipynb

[Klik disini untuk
mengakses folder
Hands-On](#)

List

Salah satu tipe data pada python yang mampu menyimpan lebih dari satu data dengan berbagai tipe data. Nilai nya bersifat *mutable* dapat diubah-ubah.



Integer

Boolean

String

Float

List

Kenapa harus List?



Kita ingin menyimpan seluruh nama kota di Indonesia, sekitar ~514 nama kota dan kabupaten. Apa yang kamu lakukan?

A. Menuliskan setiap nama kota menjadi 1 variabel yang berbeda

B. Membuat kumpulan nama dalam 1 variabel saja

Membuat list

```
1 # list kosong
2 my_list = []
3
4 # list integer
5 my_list = [1, 2, 3]
6
7 # list dengan berbagai tipe data
8 my_list = [1, "Surabaya", 3.4]
```

List dapat diisi dengan berbagai jumlah tipe data (integer, float, string, list, Boolean)

Membuat list - Multi dimensi

Contoh #1

```
# List profesi dengan multi dimensi
list_profesi1 = [
    ['Dokter', 'Perawat', 'Dosen' ],
    ['Polisi', 'Dokter', 'Barista'],
    ['Dokter', 'Polisi', 'Koki' ]
]
```

Contoh #2

```
#list profesi di dalam list ada list
list_profesi2 = [
    ['Dosen',
    'Perawat',
    'Barista',
    ['Barista', 'Perawat']
    ],
    ['Polisi',
    'Koki']
]
```

Pada tipe data **list** memungkinkan untuk memiliki list di dalam list

Struktur List

profesi =

Objek	Dokter	Perawat	Dosen
Indeks	0	1	2

Nomor indeks list selalu dimulai dari nol (0).

Nomor indeks ini kita butuhkan untuk mengambil isi (item) dari list

Mengambil sebagian item pada list #1

```
list_profesi1
```

```
[['Dokter', 'Perawat', 'Dosen'],  
 ['Polisi', 'Dokter', 'Barista'],  
 ['Dokter', 'Polisi', 'Koki']]
```

```
list_profesi1[0]
```

```
['Dokter', 'Perawat', 'Dosen']
```

```
list_profesi1[0][1]
```

```
'Perawat'
```

Nomor indeks list selalu dimulai dari nol (0).

Nomor indeks ini kita butuhkan untuk mengambil isi (item) dari list

Mengambil sebagian item pada list #2

(List Slicing)

listku =

Index	0	1	2	3	4	5
Object	NLP	Computer Vision	Reinforcement Learning	Machine Learning	Data Science	Business Intelligence

		listku[1 : 4]				
Index	0	1	2	3	4	5
Object	NLP	Computer Vision	Reinforcement Learning	Machine Learning	Data Science	Business Intelligence

listku[1:4] artinya akan mengambil index ke 1 sampai index ke 3. Index ke-empat sebagai akhir tidak diikutsertakan.

Mengambil sebagian item pada list #2

(List Slicing Formula)

Simple Slicing Python : `[start:end]`

Slicing	Makna
<code>[:]</code>	Mengambil seluruh index, tiada batasan
<code>[n :]</code>	Mengambil dari index ke-n sampai index terakhir
<code>[: m]</code>	Mengambil dari index pertama sampai index sebelum ke-m
<code>[n : m]</code>	Mengambil dari index ke-n hingga index sebelum ke-m
<code>[-1]</code>	Mengambil index terakhir
<code>[: -1]</code>	Mengambil index dari pertama sampai sebelum terakhir (selain index -1)
<code>[-1 :]</code>	Mengambil index dari dari belakang (satu index)

Double Slice Python : `[start:end:step]`

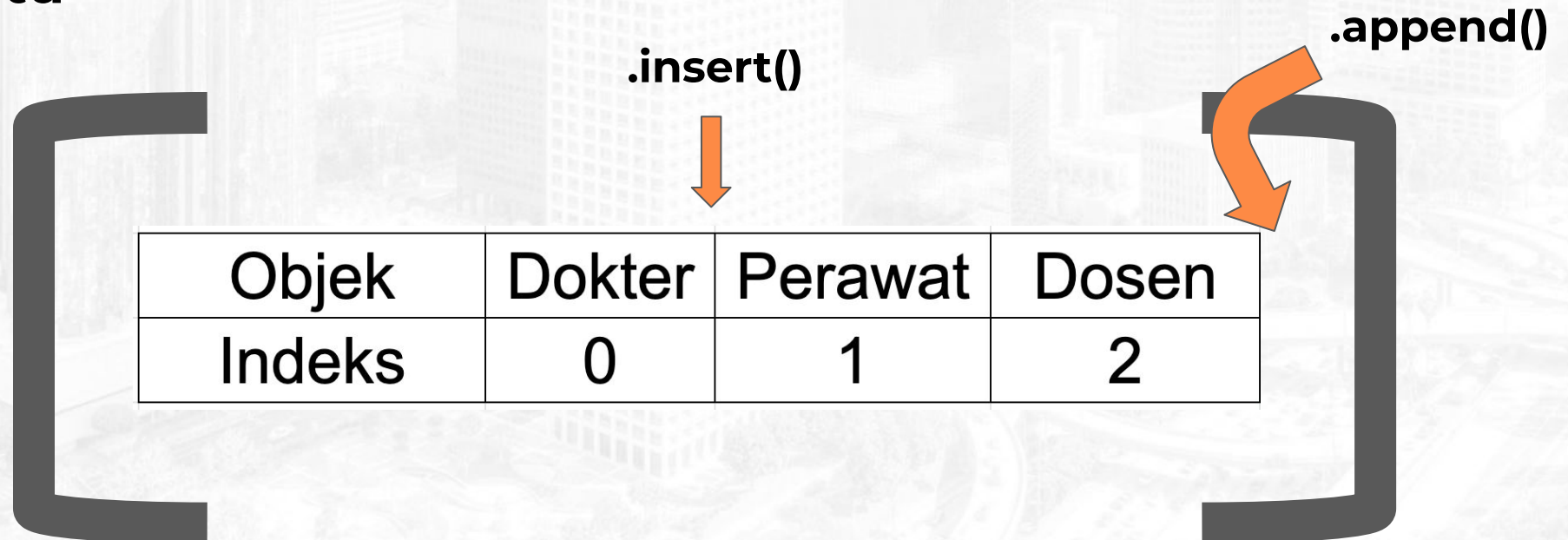
Slicing	Makna
<code>[::]</code>	Mengambil seluruh index, tiada batasan
<code>[n ::]</code>	Mengambil dari index ke-n sampai index terakhir
<code>[:: o]</code>	Mengambil dari index pertama dengan lompat setiap o index
<code>[n :: o]</code>	Mengambil dari index ke-n dengan lompat setiap o index
<code>[n : m : o]</code>	Mengambil dari index ke-n hingga sebelum ke-m, lompat setiap o index
<code>[:: -1]</code>	Mengambil seluruh value dengan mengurutkan dari belakang

Menambahkan item pada list

Menambahkan

- **append (item)** menambahkan item dari belakang
- **insert(index, item)** menambahkan item dari indeks tertentu

profesi =



Objek	Dokter	Perawat	Dosen
Indeks	0	1	2

Menambahkan item pada list

Append

```
profesi = ['Dokter', 'Perawat', 'Dosen']  
profesi.append('Polisi')  
profesi
```

```
['Dokter', 'Perawat', 'Dosen', 'Polisi']
```

Insert



```
profesi.insert(2, 'Barista')  
profesi
```

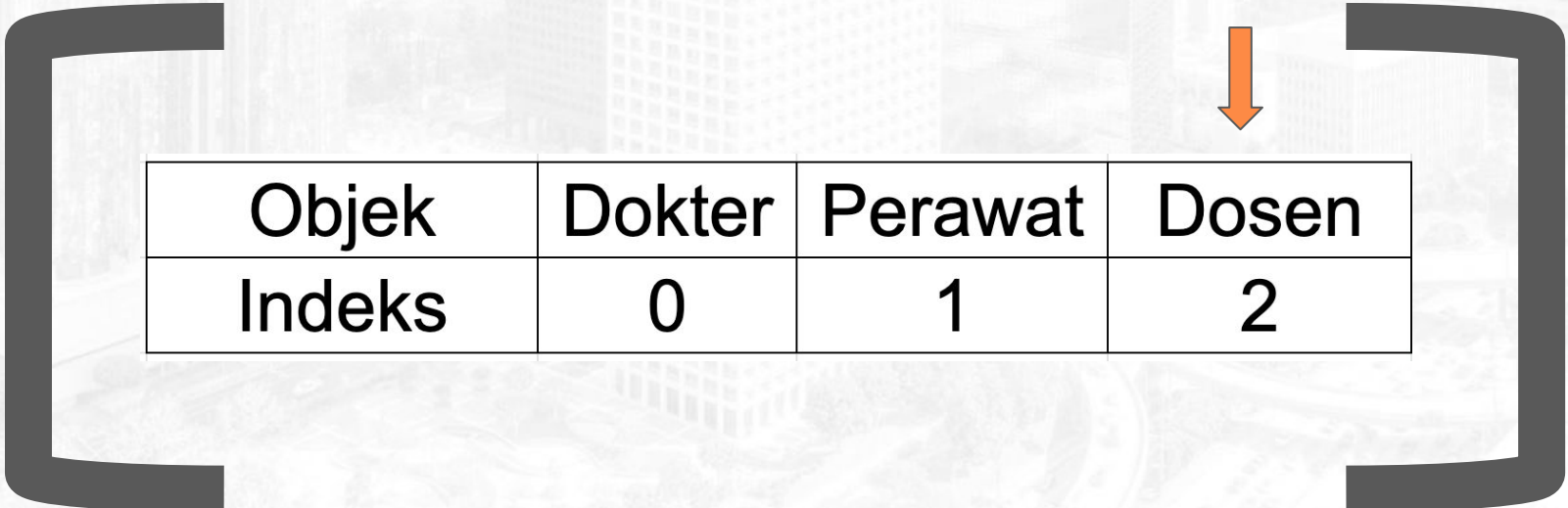
```
['Dokter', 'Perawat', 'Barista', 'Dosen', 'Polisi']
```

Menghapus item pada list

Untuk menghapus salah satu dari isi *list*, kita bisa menggunakan perintah **del <object>** dan **remove()**

del profesi[2]
atau
profesi.remove("Dosen")

profesi =



Objek	Dokter	Perawat	Dosen
Indeks	0	1	2

Menghapus item pada list

del

```
['Dokter', 'Perawat', 'Barista', 'Dosen', 'Polisi']
```

```
del profesi[2]  
profesi
```

```
['Dokter', 'Perawat', 'Dosen', 'Polisi']
```

Menghilangkan object pada index **ke-2**

remove()

```
['Dokter', 'Perawat', 'Dosen', 'Polisi']
```

```
profesi.remove('Dosen')  
profesi
```

```
['Dokter', 'Perawat', 'Polisi']
```

Menghilangkan item dengan value **Dosen**

Challenge Time

Diberikan sebuah list

`umur_user = [20, 22, 18, 21, 19]`

1. Tambahkan item baru pada list umur yaitu umur dari user baru: **23**
2. Hapus item pada list umur yaitu umur dari **user pertama**

Challenge Time

Diberikan sebuah list
`customer_id = ["A123", "A124", "B144", "C112", "B154", "C133"]`

1. Ada customer baru, tambahkan customer id tersebut yaitu **"C135"** ke dalam list `customer_id` (pada bagian belakang)
2. Hapus customer id **"A123"** karena sudah tidak aktif

List & NumPy



Pengenalan Tipe Data List



Operasi pada List



Pengenalan Tipe Data
Tuple dan Set



Operasi pada Tuple dan Set



Pengenalan NumPy



Operasi pada NumPy

Operasi List - Satuan dan Multipel

Terdapat beberapa operasi pada list, diantaranya

- Penggabungan (+)
- Perkalian (*)

```
# 2 Lists
supervised_ml = ["Logistic Regression", "SVM", "Random Forest"]
unsupervised_ml = ["KNN", "Clustering", "GMM"]

# Combination: list 1 + list 2
machine_learning = supervised_ml + unsupervised_ml

# Result
print(machine_learning)

['Logistic Regression', 'SVM', 'Random Forest', 'KNN', 'Clustering', 'GMM']
```

Penggabungan (+)

```
# List 1
user_type = ["Fraud", "Normal"]

# List Operation with *
print(user_type * 3)

['Fraud', 'Normal', 'Fraud', 'Normal', 'Fraud', 'Normal']
```

Perkalian (*)

Operasi List - Aritmatika Dasar

Objektif	Fungsi	List	Output / Hasil
Mendapatkan nilai terkecil	min()	[3,1,2,10,4]	1
Mendapatkan nilai terbesar	max()		10
Mendapatkan total jumlah dari semua item pada list	sum()		15
Mendapatkan jumlah item dalam list	len()		5
Mengurutkan item berdasarkan nilainya (terendah -> tertinggi)	sorted()		[1,2,3,4,10]

List & NumPy



Pengenalan Tipe Data List



Operasi pada List



Pengenalan Tipe Data
Tuple dan Set



Operasi pada Tuple dan Set



Pengenalan NumPy



Operasi pada NumPy

Tipe data **Tuple**

Tuple merupakan tipe data yang sama dengan **List**,
namun bersifat **immutable** (tidak bisa diubah)
sehingga waktu proses jauh lebih cepat.
Menggunakan kurung **()** pada tuple

```
bilangan_genap = (2, 4, 6, 8, 10)
```

```
print(bilangan_genap)
```

```
(2, 4, 6, 8, 10)
```


Performa **Tuple**

```
%%time
```

```
a = (1,2,3,4,5,6) * 10000 * 10000
```

```
CPU times: user 1.07 s, sys: 774 ms, total: 1.84 s
Wall time: 1.84 s
```

```
%%time
```

```
a = [1,2,3,4,5,6] * 10000 * 10000
```

```
CPU times: user 2.12 s, sys: 1.41 s, total: 3.53 s
Wall time: 3.68 s
```

***2x Lebih Cepat**



Tipe data **Set**

Salah satu tipe data pada python yang mampu menyimpan lebih dari satu data dengan berbagai tipe data unik (distinct) dan tidak memiliki index

Menggunakan kurung kurawal { }

```
bilangan_ganjil = {1,5,3,9,11,7,1,1,1}
```

```
bilangan_ganjil
```

```
{1, 3, 5, 7, 9, 11}
```

Fitur pada **Set**

```
negara = ["belanda", "jepang", "qatar", "belanda", "belanda",  
          "jepang", "jepang", "saudi arabia", "argentina", "chile",  
          "malaysia", "belanda", "belanda", "jepang", "belanda"]
```

```
# convert menjadi set
```

```
negara_2 = set(negara)
```

```
# kembalikan ke list
```

```
negara_final = list(negara_2)
```

```
print(negara_final)
```

```
['qatar', 'malaysia', 'saudi arabia', 'chile', 'belanda', 'jepang', 'argentina']
```

Bisa mentransformasikan list menjadi seperti distinct value melalui set

List & NumPy



Pengenalan Tipe Data List



Operasi pada List



Pengenalan Tipe Data
Tuple dan Set



Operasi pada Tuple dan Set



Pengenalan NumPy



Operasi pada NumPy

Operasi pada **Tuple**

Objektif	Fungsi / Operasi	Tuple	Output / Hasil
Mendapatkan jumlah objek	len()	(1,2,3,4)	4
Menyatukan dua tuple atau lebih	+ (5,6)		(1,2,3,4,5,6)
Melakukan repetisi elemen pada tuple	* 2		(1,2,3,4,1,2,3,4)
Mengetahui apakah sebuah elemen ada didalam tuple atau tidak	2 in <tuple>		True
Memilih berdasarkan indeks/slice	[1:3]		(2,3)

Operasi pada **Set**

1. Menambahkan item pada set

Untuk menambahkan satu dari isi set, kita bisa menggunakan perintah **add()** dan **update()**

`makanan.add("Rendang")`
atau
`makanan.update(['Rendang'])`

makanan =



"Bakso"	"Soto"	"Mie Ayam"	"Rendang"
---------	--------	------------	-----------

Operasi pada **Set**

2. Menghapus elemen pada set

Untuk menghapus salah satu dari isi set, kita bisa menggunakan perintah **remove()**

```
makanan.remove("Mie Ayam")
```

makanan =



Challenge Time

Diberikan set nama_user = {'Adi', 'Budi', 'Andi'}

1. Tambahkan item baru pada set nama_user yaitu **'Anton'**
2. Hapus item pada set nama_user yaitu **'Budi'**

List & NumPy



Pengenalan Tipe Data List



Operasi pada List



Pengenalan Tipe Data
Tuple dan Set



Operasi pada Tuple dan Set



Pengenalan NumPy



Operasi pada NumPy

Numpy

NumPy (Numerical Python) adalah library Python yang fokus pada scientific computing.

NumPy memiliki kemampuan untuk membentuk objek N-dimensional array, yang mirip dengan list pada Python.

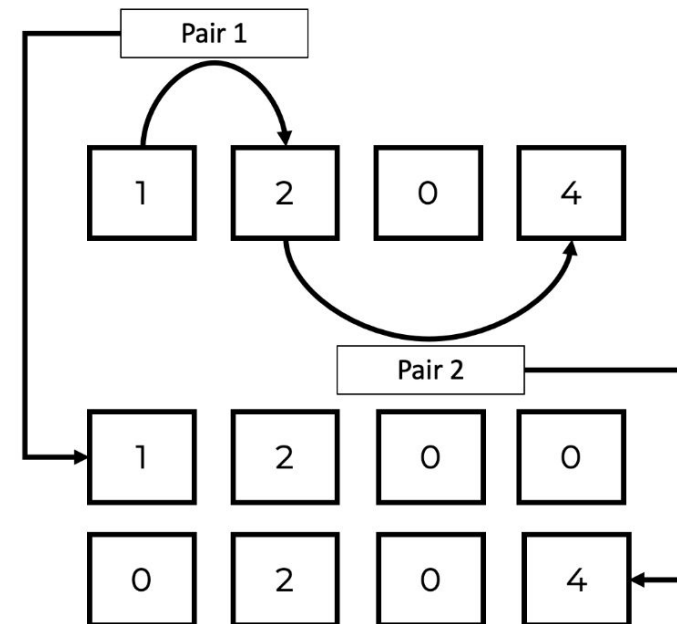


Mengapa Penting?

Numpy memudahkan seorang data scientist untuk bentuk data berupa vektor dan matrix

$$\begin{bmatrix} 1 & 4 & -2 & 5 \\ 0 & -5 & 2 & 7 \\ 0 & 0 & -3 & 9 \\ 0 & 0 & -8 & 8 \end{bmatrix}$$

Contoh persamaan linear pada matrix



Contoh coding test dari vektor dan matrix menggunakan numpy

Membuat Array

Pada Python

Format Code :

Import numpy as np

np.array(value)

Membuat Array

Pada Python

Format Code :

Import numpy as np

np.array(value)

Bisa berupa :

- Single value (integer, string, dan lainnya)
- List, Tuple, Set
- Multidimensional List

Membuat Array

Pada Python

Contoh Code 1 : Single Value

Import numpy as np

```
np.array(50)
```

```
array(50)
```

Array berupa
integer

```
np.array('ade irawan')
```

```
array('ade irawan', dtype='<U10')
```

Array berupa
string

Membuat Array

Pada Python

Contoh Code 1 : Single Value

Import numpy as np

```
np.array(50)
```

```
array(50)
```

Array berupa integer

```
np.array('ade irawan')
```

```
array('ade irawan', dtype='<U10')
```

Array berupa string

Jarang terjadi pada kasus nyata, karena **numpy** biasanya digunakan pada data numeric

Membuat Array

Pada Python

Contoh Code 2 : List

Import numpy as np

```
bilangan_genap = np.array([2, 4, 6, 8, 10])  
bilangan_genap
```

```
array([ 2,  4,  6,  8, 10])
```


Membuat Array

Pada Python

Contoh Code 2 : List

Import numpy as np

```
bilangan_genap = np.array([2, 4, 6, 8, 10])  
bilangan_genap
```

```
array([ 2,  4,  6,  8, 10])
```

Array berisikan List

Membuat Array

Pada Python

Contoh Code 3 : Multidimensional List

Import numpy as np

```
▼ bilangan = np.array([[1, 3, 5, 7, 9],  
                       [2, 4, 6, 8, 10]])  
bilangan  
  
array([[ 1,  3,  5,  7,  9],  
       [ 2,  4,  6,  8, 10]])
```

Membuat Array

Pada Python

Contoh Code 3 : Multi List atau dimensional data

Import numpy as np

```
bilangan = np.array([[1, 3, 5, 7, 9],  
                    [2, 4, 6, 8, 10]])  
bilangan  
  
array([[ 1,  3,  5,  7,  9],  
       [ 2,  4,  6,  8, 10]])
```

Value array dimulai tanda
buka kurung siku “[“

Value array ditutup tanda
tutup kurung siku “]“

Membuat Array

Pada Python

Contoh Code 3 : Multi List atau dimensional data

Import numpy as np

Kedua List dipisah oleh
tanda koma “,”

```
bilangan = np.array([[1, 3, 5, 7, 9],  
                    [2, 4, 6, 8, 10]])
```

bilangan

```
array([[ 1,  3,  5,  7,  9],  
       [ 2,  4,  6,  8, 10]])
```


Membuat Array

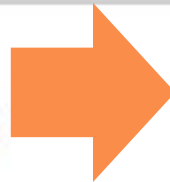
Pada Python

Contoh Code 3 : Multi List atau dimensional data

Import numpy as np

```
▼ bilangan = np.array([[1, 3, 5, 7, 9],  
                       [2, 4, 6, 8, 10]])  
bilangan
```

```
array([[ 1,  3,  5,  7,  9],  
       [ 2,  4,  6,  8, 10]])
```



Multi dimensional
array 2 x 5

Challenge Time

Buatlah array 2 x 5 dari list berikut :

umur = [20, 23, 18, 19, 30]
nomor_sepatu = [41, 42, 44, 39, 40]

Simpan array tersebut ke dalam object bernama
challenge_array

print(challenge_array)

List & NumPy



Pengenalan Tipe Data List



Operasi pada List



Pengenalan Tipe Data
Tuple dan Set



Operasi pada Tuple dan Set



Pengenalan NumPy



Operasi pada NumPy

Operasi NumPy - Function

```
1 #membuat array
2 list_nilai_kelas = np.array([65,90,89,76,78,92,94,90,100])
```

Operasi	Deskripsi	Output
np.sum(nama_array)	Menjumlahkan semua nilai pada elemen array	774
np.max(nama_array)	Mengambil nilai terbesar pada array	100
np.min(nama_array)	Mengambil nilai terkecil pada array	65
np.mean(nama_array)	Mengambil rata-rata nilai pada elemen array	86.0
np.median(nama_array)	Mengambil nilai tengah pada elemen array	90.0

Operasi NumPy - Function

```
1 #membuat array
2 list_nilai_kelas = np.array([65,90,89,76,78,92,94,90,100])
```

Operasi	Deskripsi	Output
np.log10(nama_array)	Melakukan log10 pada semua nilai pada elemen array	[1.81291336 1.95424251 1.94939001 1.88081359 1.8920946 1.96378783 1.97312785 1.95424251 2.]
np.argmax(nama_array)	Mengambil index dengan nilai terbesar	8
np.argmin(nama_array)	Mengambil index dengan nilai terkecil	0
np.sort(nama_array)	Mengurutkan item	[65 76 78 89 90 90 92 94 100]

Operasi NumPy - Dimension

Format Code :

`nama_array.reshape((n, m))`



Mengubah array
menjadi n x m dimensi

```
1 a = ['Ammar', 'Heru', 'Adit', 'Pras', 'Putri', 'Dara']
```

```
1 a = np.array(a)
```

Operasi NumPy - Dimension

Contoh Code :

```
1 a = ['Ammar', 'Heru', 'Adit', 'Pras', 'Putri', 'Dara']
```

```
1 a = np.array(a)
```

**** array dengan 1 dimensi**

Operasi NumPy - Dimension

Contoh Code :

```
1 a = ['Ammar', 'Heru', 'Adit', 'Pras', 'Putri', 'Dara']
```

```
1 a = np.array(a)
```

```
1 a = a.reshape((3, 2))
```

```
2 a
```

```
array([[ 'Ammar', 'Heru'],  
       [ 'Adit', 'Pras'],  
       [ 'Putri', 'Dara']], dtype='<U5')
```



Dimensi
menjadi 3 x 2

Operasi NumPy - Single Value

** Operasi matematika dilakukan pada setiap elemen pada List

```
#membuat array
list_nilai = np.array([65,90,89,76,78,92,94,100])

print(list_nilai * 3) #mengalikan semua element pada list nilai dengan 3
print(list_nilai / 2) #membagi semua element pada list nilai dengan 2
print(list_nilai + 10) #menjumlahkan semua element pada list nilai dengan 10
print(list_nilai - 2) #mengurangkan semua element pada list nilai dengan 2
```

```
[195 270 267 228 234 276 282 300]
[32.5 45.  44.5 38.  39.  46.  47.  50. ]
[ 75 100  99  86  88 102 104 110]
[63 88 87 74 76 90 92 98]
```

Operasi NumPy - Array vs Array

** Operasi matematika dilakukan pada setiap index yang sama dari kedua array

```
#membuat array
list_nilai1 = np.array([65,90,89,76,78,92,94,100])
list_nilai2 = np.array([1,2,3,4,5,6,7,8])

print(list_nilai1 * list_nilai2) #mengalikan semua element pada list_nilai1 dengan list_nilai2
print(list_nilai1 / list_nilai2) #membagi semua element pada list_nilai1 dengan list_nilai2
print(list_nilai1 + list_nilai2) #menjumlahkan semua element pada list_nilai1 dengan list_nilai2
print(list_nilai1 - list_nilai2) #mengurangkan semua element pada list_nilai1 dengan list_nilai2
```

```
[ 65 180 267 304 390 552 658 800]
[65.          45.          29.66666667 19.          15.6          15.33333333
 13.42857143 12.5          ]
[ 66  92  92  80  83  98 101 108]
[64 88 86 72 73 86 87 92]
```

Challenge Time

Buatlah array 3 x 3 dari list berikut :

```
nilai_1    = [ 80, 87, 91 ]  
nilai_2    = [ 77, 65, 75 ]  
nilai_3    = [ 59, 60, 47 ]
```

Simpan array tersebut ke dalam object bernama **nilai**

Dari array bernama **nilai** tersebut, ubahlah dimensinya menjadi dimensi 1.

Kemudian cek apakah **mean** (rata-rata) nya lebih besar dari pada **median** (nilai tengah) ?

Hint: Gunakan np.mean dan np.median
Gunakan kondisional untuk mengecek (boolean)

Challenge Time

Di toko A, dicatat pemasukan dan pengeluaran dari hari ke-1 hingga ke -5 dengan array2 ini:

Array **item_terjual** yang berisi [3,4,5,2,7],
Array **biaya_pengeluaran** (sudah termasuk modal) yang berisi [20000, 22000, 25000, 15000, 46000]

Lalu harga jual per item adalah 10000

Soal:

1. Carilah hari ke berapa toko tersebut mendapatkan keuntungan terbesar?
2. Cari total profit dari kelima hari hasil penjualan toko tersebut!

List & NumPy



Pengenalan Tipe Data List



Operasi pada List



Pengenalan Tipe Data
Tuple dan Set



Operasi pada Tuple dan Set



Pengenalan NumPy



Operasi pada NumPy



Terima Kasih!