

Nama: Muhammad Haidar Abdul Jabbar

Kelas : TK-44-G6

Nim:1103202071

Tugas 1

```
1) print("Hello world!") # mode skip
   ✓ 0.0s Python
2) Hello world!

3) height = 1.84
   ✓ 0.0s Python
4) tall = True
   ✓ 0.0s Python
5) height1 = 1.84
   ✓ 0.0s Python
6) height2 = 1.79
   ✓ 0.0s Python
7) height3 = 1.82
   ✓ 0.0s Python
8) height4 = 1.90
   ✓ 0.0s Python
```

Kode berupa inisialisasi nilai pada variable

```
1) [1.84, 1.79, 1.82, 1.90, 1.80]
   ✓ 0.0s Python
2) [1.84, 1.79, 1.82, 1.9, 1.8]

3) height = [1.84, 1.79, 1.82, 1.90, 1.80]
   ✓ 0.0s Python
4) height
   ✓ 0.0s Python
5) [1.84, 1.79, 1.82, 1.9, 1.8]

6) famz = ["Abe", 1.84, "Beb", 1.79, "Cory", 1.82, "Dad", 1.90]
   ✓ 0.0s Python
7) famz
   ✓ 0.0s Python
8) ["Abe", 1.84, "Beb", 1.79, "Cory", 1.82, "Dad", 1.9]

9) weight = [66.5, 60.3, 64.7, 89.5, 69.8]
   ✓ 0.0s Python
10) weight
   ✓ 0.0s Python
11) [66.5, 60.3, 64.7, 89.5, 69.8]
```

Kode berupa inisialisasi nilai list

```
1) # weight / height ** 2
   ✓ 0.0s
```

List tidak dapat melakukan operasi matematik terhadap list lain

```

import numpy as np
✓ 0.1s Python

np_height = np.array(height)
✓ 0.0s Python

np_height
✓ 0.0s Python
array([1.84, 1.79, 1.82, 1.9 , 1.8 ])

np_weight = np.array(weight)
✓ 0.0s Python

np_weight
✓ 0.0s Python
array([66.5, 60.3, 64.7, 89.5, 69.8])

bmi = np_weight / np_height ** 2
✓ 0.0s Python

bmi
✓ 0.0s Python
array([19.64201323, 18.81963734, 19.53266514, 24.79224377, 21.54320988])

```

Penggunaan numpy sebagai alternatif penggunaan list, yang lebih hemat memori dan mudah dalam melakukan operasi matematika

```

# np.
✓ 0.0s Python

np_height = np.array([1.84, 1.79, 1.82, 1.9, 1.8])
✓ 0.0s Python

np_weight = np.array([66.5, 60.3, 64.7, 89.5, 69.8])
✓ 0.0s Python

type(np_height)
✓ 0.0s Python
numpy.ndarray

type(np_weight)
✓ 0.0s Python
numpy.ndarray

np_2d = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])
✓ 0.0s Python

np_2d
✓ 0.0s Python
array([[ 1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10]])

np_2d.shape
✓ 0.0s Python
Cell 6 of 54

```

```

# series
np.array([1, 2, 3, 4, 5])
✓ 0.0s Python
array([1, 2, 3, 4, 5])

# DataFrame
np.array([[1, 2], [3, 4]])
✓ 0.0s Python
array([[1, 2],
       [3, 4]])

```

Pembuatan numpy array dan dimensinya

```
import pandas as pd

df_water_quality = pd.read_csv("dataset/water_quality_.csv")

df_water_quality
```

	ph	Hardness	Solids	Chloramines	Sulfate	Organic_carbon	Trihalomethanes	Turbidity	Check
0	NaN	204.890456	20791.31898	7.300212	368.516441	10.379783	86.990970	2.963135	0
1	3.716080	129.422921	18630.05786	6.635246	NaN	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.54173	9.275884	NaN	16.868637	66.420093	3.055934	0
3	8.316766	214.373394	22018.41744	8.059332	356.886136	18.436525	100.341674	4.628771	0
4	9.092223	181.101509	17978.98634	6.546600	310.135738	11.558279	31.997993	4.075075	0
...
3271	4.668102	193.681736	47580.99160	7.166639	359.948574	13.894419	66.687695	4.435821	1
3272	7.808856	193.553212	17329.80216	8.061362	NaN	19.903225	NaN	2.798243	1
3273	9.419510	175.762646	33155.57822	7.350233	NaN	11.039070	69.845400	3.298875	1
3274	5.126763	230.603758	11983.86938	6.303357	NaN	11.168946	77.488213	4.708658	1
3275	7.874671	195.102299	17404.17706	7.509306	NaN	16.140368	78.698446	2.309149	1

3276 rows x 9 columns

Melakukan import pandas library untuk manipulasi file dan pemanggilan dataset

```
df_water_quality["ph"]

df_water_quality.Hardness
```

42] ✓ 0.0s

```
0      NaN
1    3.716080
2    8.099124
3    8.316766
4    9.092223
...
3271   4.668102
3272   7.808856
3273   9.419510
3274   5.126763
3275   7.874671
Name: ph, Length: 3276, dtype: float64
```

43] ✓ 0.0s

```
0      204.890456
1      129.422921
2      224.236259
3      214.373394
4      181.101509
...
3271   193.681736
3272   193.553212
3273   175.762646
3274   230.603758
3275   195.102299
Name: Hardness, Length: 3276, dtype: float64
```

+ Code + Markdown

Melakukan pemanggilan data column