

## Accessing String (Lanjutan)

String merupakan kumpulan beberapa karakter.

## Penggunaan Method

```
In [21]: print(data)
Kelas Algoritma
```

```
In [23]: print(data.upper())
KELAS ALGORITMA
```

```
In [24]: print(data.find('m'))
13
```

```
In [25]: print(data.replace('K', 'T'))
Telas Algoritma
```

## Contoh Aplikasi String: Konversi Biner ke Desimal

$$\begin{aligned} & \text{101} \\ & \rightarrow 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 \\ & \rightarrow 1 + 0 + 4 = 5 \end{aligned}$$

$$\begin{aligned} & \underline{\text{1001}} \\ & \rightarrow 1 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 \\ & \rightarrow 1 + 0 + 0 + 8 = 9 \end{aligned}$$

Kita membaca setiap karakter mulai dari belakang (pangkat terendah)

Langkah2nya:

```
In [26]: binStr='110'
for i in range(len(binStr)):
    print(binStr[i])
```

1. Membaca karakter dari depan

2. Modifikasi for

a. Kondisi awal (no.1)

```
In [28]: binStr='110'
for i in range(0, len(binStr), 1): # (awal, akhir, interval)
    print(binStr[i])
```

b. Membaca karakter dari belakang

```
In [29]: binStr='110'
         for i in range(len(binStr)-1,0,-1):#(awal,akhir, interval)
           print(binStr[i])

0
1
```

```
In [30]: binStr='110'
         for i in range(len(binStr)-1,-1,-1):#(awal,akhir, interval)
           print(binStr[i])

0
1
1
```

c. Mengalikan dg  $2^n$  & dijumlahkan

```
In [31]: binStr='110'
         pangkat=0
         hasil=0
         for i in range(len(binStr)-1,-1,-1):#(awal,akhir, interval)
           print(binStr[i],pangkat)
           hasil=hasil+int(binStr[i])*2**pangkat
           pangkat+=1

0 0
1 1
1 2
```

Karakter yg kita baca 011 dari iterasi ke 1-3, pangkatnya dari 0-2, naik 1 per 1 (dilakukan increment)

```
In [32]: binStr='110'
         pangkat=0
         hasil=0
         for i in range(len(binStr)-1,-1,-1):#(awal,akhir, interval)
           print(binStr[i],pangkat)
           hasil=hasil+int(binStr[i])*2**pangkat
           pangkat+=1
         print(hasil)

0 0
1 1
1 2
6
```

```
In [33]: binStr=input('masukkan biner = ')
          |
          pangkat=0
          hasil=0
          for i in range(len(binStr)-1,-1,-1):#(awal,akhir, interval)
              print(binStr[i],pangkat)
              hasil=hasil+int(binStr[i])*2**pangkat
              pangkat+=1
          print(hasil)

          masukkan biner = 1000
          0 0
          0 1
          0 2
          1 3
          8
```

```
In [35]: binStr=input('masukkan biner = ')
          pangkat=0
          hasil=0
          for i in range(len(binStr)-1,-1,-1):#(awal,akhir, interval)
              print(binStr[i],pangkat)
              hasil=hasil+int(binStr[i])*2**pangkat
              pangkat+=1
          print(hasil)

          masukkan biner = 1010
          0 0
          1 1
          0 2
          1 3
          10
```

# LIST

## ALGORITMA PEMROGRAMAN

List adalah kumpulan dari beberapa data atau value.

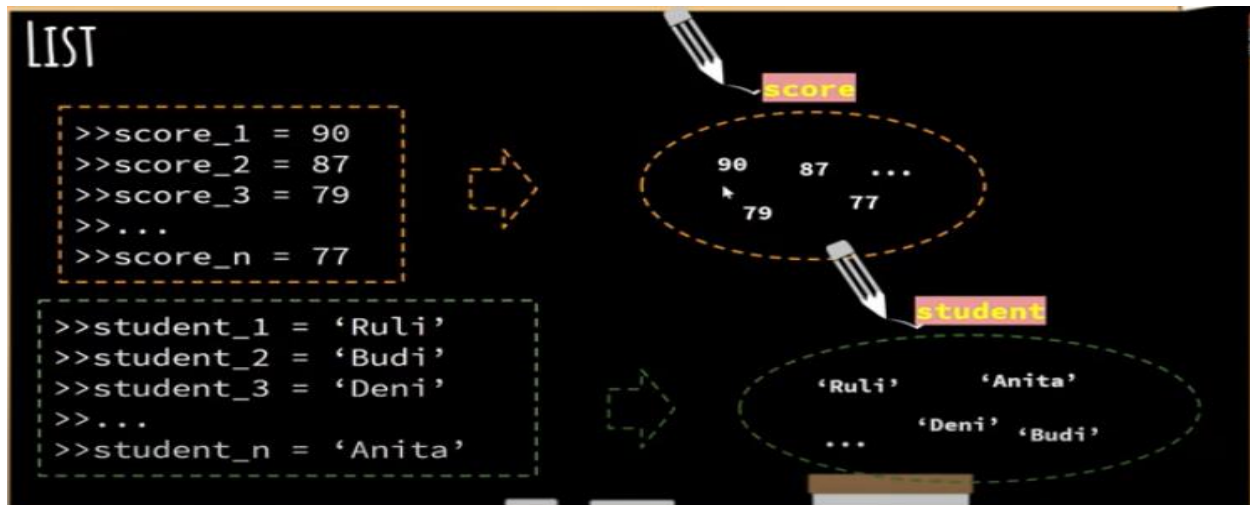
Misal: Untuk sejumlah mahasiswa, kita memiliki variabel untuk menyimpan score (tipe data integer, valuenya 90); selain itu kita juga menyimpan variabel nama mhs dg tipe data string, value nya adalah Ruli (student\_1 = 'Ruli').

```
LIST

>>>score_1 = 90
>>>score_2 = 87
>>>score_3 = 79
>>...
>>>score_n = 77

>>>student_1 = 'Ruli'
>>>student_2 = 'Budi'
>>>student_3 = 'Deni'
>>...
>>>student_n = 'Anita'
```

Penyebutan tersebut jika jumlah data ratusan/lebih maka menjadi tidak efektif. Lebih baik jika kita mengumpulkan variabel dalam 1 wadah yg sama untuk nilai yang sama dan tujuan yang sama. Seperti skor merepresentasikan nilai mahasiswa; student merepresentasikan nama mahasiswa. Tujuan pengumpulan value/data tersebut yaitu supaya lebih rapi, terorganisir, pengolahan data menjadi lebih mudah.

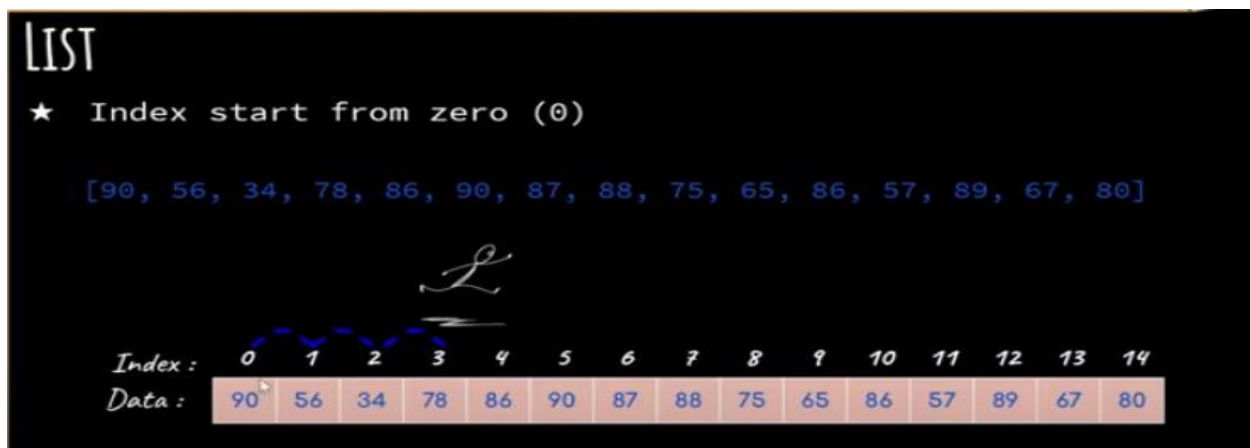


Untuk mengidentifikasi kumpulan data maka kumpulan data juga kita beri nama variable score utk kumpulan nilai & student utk kumpulan nama mhs. Kumpulan data tersebut disebut list.

List dalam Python bias mengumpulkan data dg tipe data yg berbeda.



Untuk mengakses list sama seperti string yaitu dimulai dari 0



## ACCESSING LIST

```
>>listVar[a:b]
```

- ★ Retrieve members from index **a** up to index **b-1**
- ★ `[n]`, `[:]`, `[a:]`, `[:b]`, `[-1]`, `[:-1]`
- ★ **Mutable**

```
In [5]: data[0]
Out[5]: 90

In [6]: #data[a:b]
         data[6:10]
Out[6]: [87, 88, 75, 65]
```

akses anggota indeks ke-0 sampai dengan b-1 :  
namaList[0:b], atau  
namaList[:b]

```
In [7]: #data[0:5]
Out[7]: [90, 56, 34, 78, 86]
```

```
In [8]: #data[0:5]
         data[:5]
Out[8]: [90, 56, 34, 78, 86]
```

```
In [9]: #data[9:15]
         data[9:]
Out[9]: [65, 86, 57, 89, 67, 80]
```

```
In [11]: data[-1]
Out[11]: 80
```

```
In [13]: print(data)
         data[1]=75
         print(data)

[90, 56, 34, 78, 86, 90, 87, 88, 75, 65, 86, 57, 89, 67, 80]
[90, 75, 34, 78, 86, 90, 87, 88, 75, 65, 86, 57, 89, 67, 80]
```

Menunjukkan bahwa list bersifat mutable, artinya data bisa diubah langsung menggunakan assignment (=).

```
★ Using iteration/loop
★ len(strVar)
```

```
>>lsData=[a,b,c,...,z]
>>for i in range(len(lsData)):
    Syntax_1 → lsData[i]
    ...
    Syntax_n → lsData[n]
```

```
>>lsData=[a,b,c,...,z]
>>for dt in lsData:
    Syntax_1 → dt
    ...
    Syntax_n → dt
```

Kita bisa melakukan akses setiap data pada list menggunakan index atau langsung akses tiap data.

```
In [13]: print(data)
data[1]=75
78, 86, 90, 87, 88, 75, 65, 86, 57, 89, 67, 80]
[90, 75, 34, 78, 86, 90, 87, 88, 75, 65, 86, 57, 89, 67, 80]
```

**len(namaList) : banyaknya data, atau jumlah data, bukan indeks terakhir**

```
In [ ]: num=len(data)
for i in range(num):
    print(data[i])
```

```
In [14]: num=len(data)
print(num)
for i in range(num):
    print(data[i])
```

15  
90  
75  
34  
78  
86  
90  
87  
88  
75  
65  
86  
57  
89  
67  
80

mengakses list melalui index

```
In [15]: for nilai in data:
        print(nilai)
```

90  
75  
34  
78  
86  
90  
87  
88  
75  
65  
86  
57  
89  
67  
80

Mengakses list langsung melalui data.

Contoh implementasi list, untuk menghitung jumlah total nilai pada list atau nilai rata2.

```
In [16]: temp=0
for nilai in data:
    temp=temp+nilai
print('total=',temp)
```

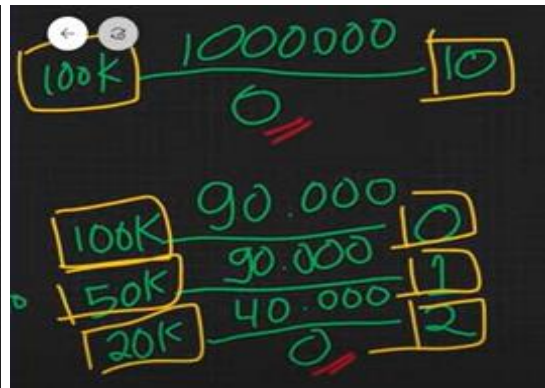
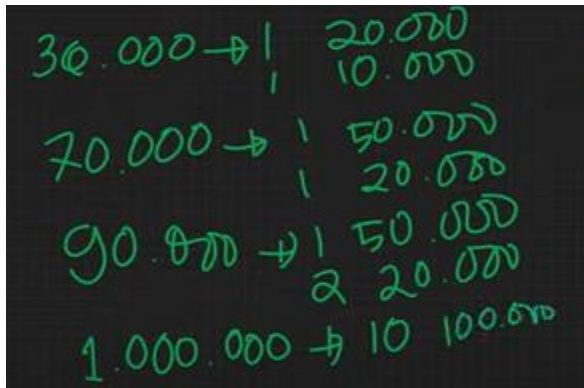
total= 1147

```
In [17]: temp=0
for nilai in data:
    temp=temp+nilai
ratarata=temp/len(data)
print('rata-rata=',ratarata)
```

rata-rata= 76.46666666666667



Contoh lain



```
In [23]: M uang=[100000,50000,20000,10000,5000,2000,1000]
strUang=['seratus ribu', 'lima puluh ribu', 'dua puluh ribu','sepuluh ribu','lima ribu','dua ribu','seribu']

rupiah=int(input('masukkan uang = '))
sisarupiah
stop=False
indUang=0
while not(stop):
    bagi=sisarupiah//uang[indUang]
    print(bagi, ' lembar ', strUang[indUang])
    sisan=sisarupiah%uang[indUang]
    if sisan==0:
        stop=True
        indUang+=1

masukkan uang = 99000
0 lembar seratus ribu
1 lembar lima puluh ribu
2 lembar dua puluh ribu
0 lembar sepuluh ribu
1 lembar lima ribu
2 lembar dua ribu
```

```
In [24]: M uang=[100000,50000,20000,10000,5000,2000,1000]
strUang=['seratus ribu', 'lima puluh ribu', 'dua puluh ribu','sepuluh ribu','lima ribu','dua ribu','seribu']

rupiah=int(input('masukkan uang = '))
sisarupiah
stop=False
indUang=0
while not(stop):
    bagi=sisarupiah//uang[indUang]
    if bagi!=0:
        print(bagi, ' lembar ', strUang[indUang])
    sisan=sisarupiah%uang[indUang]
    if sisan==0:
        stop=True
        indUang+=1

masukkan uang = 99000
1 lembar lima puluh ribu
2 lembar dua puluh ribu
1 lembar lima ribu
2 lembar dua ribu
```

