



Daspro B2

Setelah SET Terbitlah LOL

Pertemuan 9



Apa Itu **LOL**?

List of list adalah sebuah list kosong atau sebuah list yang elemennya dapat berupa atom maupun list. Karena elemen list of list dapat berupa atom maupun list, maka saat pemrosesan elemen biasanya memerlukan penanganan yang berbeda sesuai dengan jenisnya (atom atau list).

Contohnya ?

Ini List :

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Ini List Of List :

[1, 2, 3, 4, [5, 6], 7, [8, 9, 10]]

Kamu perlu tau ini !

Dalam **List Of List**, kita menggunakan istilah baru yaitu Atom
Apa itu Atom ?

Atom pada **LOL** merujuk pada elemen yang berada di dalam sebuah List. Bisa dikatakan atom jika elemen tersebut tunggal atau bukan berupa list. Misalkan sesuai contoh :

Ini List Of List :

[1, 2, 3, 4, [5, 6], 7, [8, 9, 10]]

Elemen 1, 2, 3, 4, 7 merupakan sebuah Atom, lalu di dalam list diatas terdapat list yang di dalamnya terdapat list juga yaitu [5, 6] dan [8, 9, 10] sehingga disebut List Of List.



Kamu perlu tau ini !

Ini List Of List :

[1, 2, 3, 4, [5, 6], 7, [8, 9, 10]]

List of List diatas memiliki 7 Elemen yaitu 1, 2, 3, 4, [5, 6], 7, [8, 9, 10]

Hal ini juga berlaku untuk indexnya, jika dalam python elemen 1 pada LOL diatas memiliki index 0.

Dalam **List Of List**, kita dapat mengecek apakah sebuah elemen dalam List of List berupa Atom atau List dengan menggunakan fungsi predikat **IsAtom?** dan **IsList?**.

Mari buka senjata kita, Open IDE favoritmu, lalu ketik kode-kode indahmu itu !




Bukan praktikum namanya jika tidak ngoding?

**Mari patuhi dan lakukan
langkah berikut !**



Berikut adalah langkah-langkah pada praktikum ini:

1. Buatlah sebuah file python bernama LOL_.py, contohnya: LOL_24060119110023.py.
2. Tuliskan nama file, deskripsi, pembuat, dan tanggal pada bagian awal file Anda sebagai komentar.
3. Agar dapat menggunakan konstruktor, selektor, dan fungsi yang telah dibuat pada list, lakukan import dengan menambahkan baris kode seperti berikut:

```
 LOL_24060123120020.py  
1  from list import *
```

Latihan Kita

 LOL_24060123120020.py > ...

```
1  # Nama File : LOL_24060123120020.py
2  # Deskripsi : Program yang berisi implemtasi operasi-operasi List Of List
3  # Pembuat   : Uzumaki Naruto
4  # Tanggal   : Rabu, 13 November 2024
5
6  from list import *
7
8  #DEIFINISI DAN SPESIFIKASI TYPE PREDIKAT KHUSUS LIST
9  # IsEmpty : list of list -> boolean
10 # IsEmpty(L) true jika list of list kosong, false jika tidak
11 def IsEmpty(L) :
12     | return L == []
13
14 # IsAtom : list of list -> boolean
15 # IsAtom(L) true jika list of list L adalah sebuah Atom, false jika tidak
16 def IsAtom(L) :
17     | return type(L) != list
18
19 # IsList : list of list -> boolean
20 # IsList(L) true jika list of list L adalah sebuah List, false jika tidak
21 def IsList(L) :
22     | return type(L) == list
23
```

Latihan Kita

```
35  #DEIFINISI DAN SPESIFIKASI SELEKTOR
36  # FirstList : list of list tidak kosong -> list
37  # FirstList(S) mengembalikan elemen pertama dari list of list S (mungkin sebuah atom atau list).
38  def FirstList(S) :
39  |     return S[0]
40
41  # LastList : list of list tidak kosong -> list
42  # LastList(S) mengembalikan elemen terakhir dari list of list S (mungkin sebuah atom atau list).
43  def LastList(S) :
44  |     return S[-1]
45
46  # TailList : list of list tidak kosong -> list of list
47  # TailList(S) mengembalikan list of list yang elemen pertamanya dihapus.
48  def TailList(S) :
49  |     return S[1:]
50
51  # HeadList : list of list tidak kosong -> list of list
52  # HeadList(S) mengembalikan list of list yang elemen terakhirnya dihapus.
53  def HeadList(S) :
54  |     return S[:-1]
55
```


Latihan Kita

```
56  # Aplikasi
57  print(KonsLo([1, 2, 3], [4, 5, 6])) # [[1, 2, 3], 4, 5, 6]
58  print(KonsLi([1, 2, 3], [4, 5, 6])) # [1, 2, 3, [4, 5, 6]]
59  print(FirstList([[1, 2, 3], 4, 5, 6])) # [1, 2, 3]
60  print>LastList([[1, 2, 3], 4, 5, 6])) # 6
61  print(TailList([[1, 2, 3], 4, 5, 6])) # [4, 5, 6]
62  print(HeadList([[1, 2, 3], 4, 5, 6])) # [1, 2, 3]
63  print(IsEmpty([])) # True
64  print(IsEmpty([1])) # False
65  print(IsAtom(1)) # True
66  print(IsAtom([1, 2, 3])) # False
67  print(IsList([1, 2, 3])) # True
68  print(IsList(1)) # False
69
```

Latihan Kita

*** Jangan lupa setiap fungsi dibuatkan Aplikasinya yaa**

```
56 # Aplikasi
57 print(KonsLo([1, 2, 3], [4, 5, 6])) # [[1, 2, 3], 4, 5, 6]
58 print(KonsLi([1, 2, 3], [4, 5, 6])) # [1, 2, 3, [4, 5, 6]]
59 print(FirstList([[1, 2, 3], 4, 5, 6])) # [1, 2, 3]
60 print>LastList([[1, 2, 3], 4, 5, 6])) # 6
61 print>TailList([[1, 2, 3], 4, 5, 6])) # [4, 5, 6]
62 print>HeadList([[1, 2, 3], 4, 5, 6])) # [1, 2, 3]
63 print(IsEmpty([])) # True
64 print(IsEmpty([1])) # False
65 print(IsAtom(1)) # True
66 print(IsAtom([1, 2, 3])) # False
67 print(IsList([1, 2, 3])) # True
68 print(IsList(1)) # False
69
```

SEPERTI BIASA HEHE!

```
# IsMemberS: elemen, list of list-> boolean
# IsMemberS(x,S) mengembalikan true jika elemen x ada di dalam list of list S
# Contoh aplikasi:
# IsMemberS(3, []) {menghasilkan false}
# IsMemberS(3, [2, 4, 3, [1], [4,5]]) {menghasilkan true}
# IsMemberS(3, [2, 4, 7, [1], [3,5]]) {menghasilkan true}

# Rember: elemen, list of list -> list of list
# Rember(x,S) menghapus semua elemen x yang ada di list of list S
# Contoh aplikasi:
# Rember*(3, []) {menghasilkan []}
# Rember*(3, [4, 3, [2,4], 3]) {menghasilkan [4, [2,4]]}
# Rember*(3, [2, 4, [3,6,9], 5, 3]) {menghasilkan [2, 4, [6,9], 5]}

# Max: list of list -> elemen
# Max(S) mengembalikan elemen maksimum di dalam list of list S
# Contoh aplikasi:
# Max([4, 5, 6, [8,9,10], [12,0], 8]) {menghasilkan 12}
# Max([4, 15, 6, [8,9,10], [12,0], 8]) {menghasilkan 15}
```

SEPERTI BIASA HEHE!

```
# NBEltAtom: list of list -> integer
# NBEltAtom(S) mengembalikan banyaknya elemen list of list S yang berupa atom
# Contoh aplikasi:
# NBEltAtom([4, 5, 6, [8,9,10], [12,0], 8]) {menghasilkan 4}
# NBEltAtom([4, 15, 6, [8,9], 10, [12], 8]) {menghasilkan 5}
# NBEltAtom([[8,9,10]]) {menghasilkan 0}

# NBEltList: list of list -> integer
# NBEltList(S) mengembalikan banyaknya elemen list of list S yang berupa list
# Contoh aplikasi:
# NBEltList([4, 5, 6, [8,9,10], [12,0], 8]) {menghasilkan 2}
# NBEltList([[4, 15], 6, [8,9], 10, [12], 8]) {menghasilkan 3}
# NBEltList([[8,9,10]]) {menghasilkan 1}

# SumLoL: list of list -> integer
# SumLoL(S) mengembalikan jumlah semua elemen dalam list of list S
# Contoh aplikasi:
# SumLoL([[]]) {menghasilkan 0}
# SumLoL([4, 5, 6, [2,3,1]]) {menghasilkan 21}
# SumLoL([[2,3,4]]) {menghasilkan 9}
```

SEPERTI BIASA HEHE!

```
# MaxNBElmtList: list of list -> integer
# MaxNBElmtList(S) mengembalikan banyaknya elemen list maksimum yang ada pada list of list S
# Contoh aplikasi:
# MaxNBElmtList([[4,5,6,7], [8,9,10], [12,0], 8]) {menghasilkan 4}
# MaxNBElmtList([[4,15], 6, [8,9], 10, [12], 8]) {menghasilkan 2}
# MaxNBElmtList([8,9,10]) {menghasilkan 0}

# MaxSumElmt: list of list -> integer
# MaxSumElmt(S) mengembalikan elemen maksimum pada list of list S
# jika elemen berupa list, maka dihitung jumlahan elemen dalam list tersebut
# jika elemen atom, maka nilainya adalah elemen tersebut
# Contoh aplikasi:
# MaxSumElmt([1,2], 9, [4,5,6], 8) {menghasilkan 15}
# MaxSumElmt([1,2], 90, [4,5,6], 8) {menghasilkan 90}
# MaxSumElmt([8,9,10]) {menghasilkan 10}
# MaxSumElmt([[8,9,10]]) {menghasilkan 27}
```



**“ Bukan tentang seberapa cepat kamu berhasil,
Tetapi tentang seberapa banyak kamu mencoba
untuk berhasil. ”**



**Terimakasih untuk hari ini,
Semoga harimu menyenangkan.**