

PRAKTIKUM DASAR PEMROGRAMAN

PERTEMUAN KE-9

LIST OF LIST

TUJUAN PRAKTIKUM

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu merealisasikan set ke dalam bahasa pemrograman Python serta mengaplikasikan masalah/kasus yang perlu diselesaikan dengan menggunakan set.

TOOLS

Tools yang diperlukan untuk melakukan praktikum ini adalah interpreter Python yang telah terinstall di komputer.

MATERI PRAKTIKUM:

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). Untuk itu, dalam list of list diberikan predikat tambahan `IsAtom` dan `IsList` untuk mengecek apakah sebuah elemen berupa list atau atom.

Meskipun bahasa Python memiliki type bawaan list, namun set dalam praktikum ini akan tetap direalisasikan (dalam paradigma fungsional/ secara rekursif) sebagai list of list sebagaimana yang dipelajari dalam kuliah atau Diktat.

Berikut adalah langkah-langkah pada praktikum ini:

1. Buatlah sebuah file python bernama `lol_<nim>.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:

```
from list import *
```

File list dan lol berada dalam folder yang sama.

4. Buatlah fungsi predikat khusus dalam list of list seperti berikut:

```
8  #DEFINISI DAN SPESIFIKASI PREDIKAT KHUSUS LIST
9  # isEmpty: list of list -> boolean
10 # isEmpty(S) benar jika S adalah list of list kosong.
11 def isEmpty(S):
12 |     return S == []
13
14 # IsAtom: list of list -> boolean
15 # IsAtom(S) benar jika S adalah sebuah atom.
16 def isAtom(S):
17 |     return type(S) != list
18
19 # IsList: list of list -> boolean
20 # IsList(S) benar jika S adalah sebuah list.
21 def isList(S):
22 |     return type(S) == list
```

5. Buatlah konstruktor list of list seperti berikut:

```
24 #DEFINISI DAN SPESIFIKASI KONSTRUKTOR
25 # KonsLo: list, list of list -> list of list
26 # KonsLo(L,S) membentuk list baru dengan list (L) yang diberikan sebagai elemen
27 # pertama list of list (S).
28 def konsLo(L,S):
29 |     return [L] + S
30
31 # KonsLi: list of list, list -> list of list
32 # KonsLi(S,L) membentuk list baru dengan list (L) yang diberikan sebagai elemen
33 # terakhir list of list (S).
34 def konsLi(S,L):
35 |     return S + [L]
36
```

6. Buatlah selektor khusus list of list seperti berikut:

```
37 #DEFINISI DAN SPESIFIKASI SELEKTOR
38 # FirstList: list of list tidak kosong -> list
39 # FirstList(S) menghasilkan elemen pertama list (mungkin sebuah list atau atom)
40 def firstList(S):
41 |     return S[0]
42
43 # LastList: list of list tidak kosong -> list
44 # LastList(S) menghasilkan elemen terakhir list (mungkin sebuah list atau atom)
45 def lastList(S):
46 |     return S[-1]
47
48 # TailList: list of list tidak kosong -> list of list
49 # LastList(S) menghasilkan sisa list of list S tanpa elemen pertamanya.
50 def tailList(S):
51 |     return S[1:]
52
53 # HeadList: list of list tidak kosong -> list of list
54 # HeadList(S) menghasilkan sisa list of list S tanpa elemen terakhirnya.
55 def headList(S):
56 |     return S[:-1]
```

7. Berikut adalah contoh realisasi sebuah fungsi IsEqS(S1,S2) dan IsMemberLS(L,S) dalam bahasa Python.

```
79 # IsMemberLS: list, list of list-> boolean
80 # IsMemberLS(L,S) mengembalikan true jika list L ada di dalam list of list S
81 def isMemberLS(L,S):
82 |     if(isEmpty(S)):
83 |         return False
84 |     else:
85 |         if(isAtom(firstList(S))):
86 |             return isMemberLS(L,tailList(S))
87 |         else:
88 |             if(isEqual(firstList(S),L)):
89 |                 return True
90 |             else:
91 |                 return isMemberLS(L,tailList(S))
```

```

58 #DEFINISI DAN SPESIFIKASI FUNGSI YANG MENGOPERASIKAN LIST
59 # IsEqS : 2 list of list -> boolean
60 # IsEqS(S1,S2) benar jika S1 memiliki elemen dengan nilai dan urutan yang sama dengan S2
61 def isEqS(S1,S2):
62     if(isEmpty(S1) and isEmpty(S2)):
63         return True
64     elif(isEmpty(S1) and not isEmpty(S2)):
65         return False
66     elif(not isEmpty(S1) and isEmpty(S2)):
67         return False
68     else:
69         if(isAtom(firstList(S1)) and isAtom(firstList(S2))):
70             if(firstList(S1) == firstList(S2)):
71                 return isEqS(tailList(S1),tailList(S2))
72             else:
73                 return False
74         elif(isList(firstList(S1)) and isList(firstList(S2))):
75             return isEqS(firstList(S1),firstList(S2)) and isEqS(tailList(S1),tailList(S2))
76         else:
77             return False

```

8. Setiap kali merealisasikan sebuah fungsi, aplikasikan fungsi tersebut, sehingga dapat diketahui apakah realisasi sudah sesuai dengan spesifikasi yang diharapkan.

Contoh aplikasi fungsi berikut dipanggil di dalam fungsi `print()` agar hasil aplikasi atau output dapat langsung ditampilkan pada output program.

Aplikasi	Output
94 <code>print(konsLo([2],[3]))</code>	<code>[[2], 3]</code>
95 <code>print(konsLi([3,4,5],[6]))</code>	<code>[3, 4, 5, [6]]</code>
96 <code>print(firstList([[3],4,5,6,7]))</code>	<code>[3]</code>
97 <code>print(lastList([3,4,5,6,[7]]))</code>	<code>[7]</code>
98 <code>print(tailList([3,[4,5,6,7]]))</code>	<code>[[4, 5, 6, 7]]</code>
99 <code>print(headList([[3,4,5,6],7]))</code>	<code>[[3, 4, 5, 6]]</code>
100 <code>print(isEmpty([]))</code>	<code>True</code>
101 <code>print(isEmpty([[]]))</code>	<code>False</code>
102 <code>print(isAtom(firstList([3],4,5,6,7)))</code>	<code>False</code>
103 <code>print(isList(firstList([3],4,5,6,7)))</code>	<code>True</code>
104 <code>print(NBE1mt([3,4,[5,6],7]))</code>	<code>4</code>
105 <code>print(isEqS([3,[4,5],6,7],[3,[4,5],6,7]))</code>	<code>True</code>
106 <code>print(isEqS([3,4,5,6,7],[3,[4,5],6,7]))</code>	<code>False</code>
107 <code>print(isEqS([3,[4,5],[6],7],[3,[4,5],6,7]))</code>	<code>False</code>
108 <code>print(isMemberLS([6],[3,[4,5],6,[6],7]))</code>	<code>True</code>
109 <code>print(isMemberLS([4,5],[3,[4,5],6,[6],7]))</code>	<code>True</code>

9. Kerjakan fungsi-fungsi berikutnya yang ada pada bagian latihan.

10. Pada akhir sesi praktikum, laporkan hasil praktikum pada link berikut <https://bit.ly/daspro24lap>.

TUGAS/LATIHAN

Buatlah realisasi python untuk fungsi-fungsi berikut yang telah diberikan definisi dan spesifikasinya. Untuk setiap fungsi yang telah direalisasikan buatlah aplikasinya dengan berbagai variasi input untuk menguji apakah output fungsi sudah sesuai dengan yang seharusnya.

```
# 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}

# 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}

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

# MaxSumElt: list of list -> integer
# MaxSumElt(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}
```

*****Selamat Mengerjakan dan Berlatih *****