

NAMA : DANANG AJI NUGROHO  
NIM : L200180015  
KELAS : A

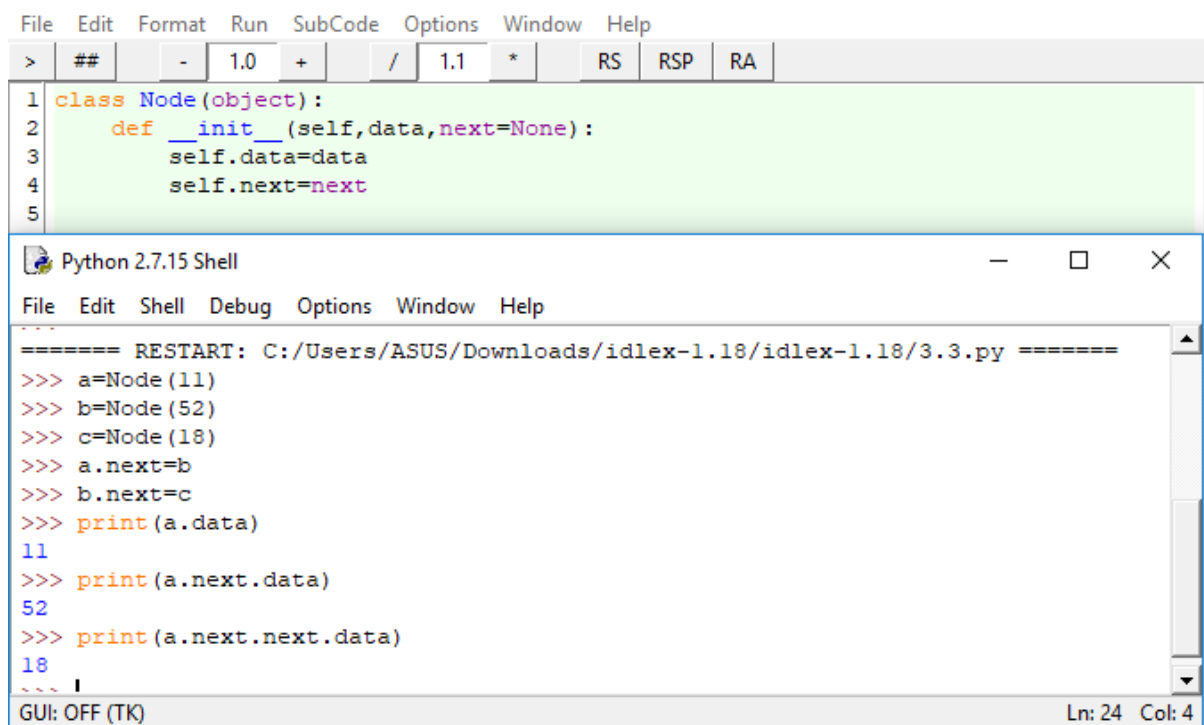
### LATIHAN 3.1

```
>>> A= [ [2,3], [5,7] ]  
>>> A[0][1]  
3  
>>> A[1][1]  
7
```

### LATIHAN 3.2

```
>>> B= [ [0 for j in range(3)] for i in range(3)]  
>>> B  
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]  
>>> |
```

### LATIHAN 3.3



The screenshot shows the Python IDLE interface. The top part is the editor window with a menu bar (File, Edit, Format, Run, SubCode, Options, Window, Help) and a toolbar. The code in the editor is:

```
1 class Node(object):  
2     def __init__(self, data, next=None):  
3         self.data=data  
4         self.next=next  
5
```

The bottom part is the Python 2.7.15 Shell window, which has a menu bar (File, Edit, Shell, Debug, Options, Window, Help). It shows the execution of the code from the editor, with line numbers on the left:

```
===== RESTART: C:/Users/ASUS/Downloads/idlex-1.18/idlex-1.18/3.3.py =====  
>>> a=Node(11)  
>>> b=Node(52)  
>>> c=Node(18)  
>>> a.next=b  
>>> b.next=c  
>>> print(a.data)  
11  
>>> print(a.next.data)  
52  
>>> print(a.next.next.data)  
18  
... |
```

The status bar at the bottom indicates "GUI: OFF (TK)" on the left and "Ln: 24 Col: 4" on the right.

```
File Edit Format Run SubCode Options Window Help
> ## - 1.0 + / 1.1 * RS RSP RA
1 class Node(object):
2     def __init__(self,data,next=None):
3         self.data=data
4         self.next=next
5
6 def kunjungi(head):
7     curNode=head
8     while curNode is not None:
9         print(curNode.data)
10        curNode=curNode.next
11
```

```
Python 2.7.15 Shell
File Edit Shell Debug Options Window Help
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:22:17) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ASUS/Downloads/idlex-1.18/idlex-1.18/3.3.py =====
>>> a=Node(11)
>>> b=Node(52)
>>> c=Node(18)
>>> a.next=b
>>> b.next=c
>>> kunjungi(a)
11
52
18
```

```

File Edit Format Run SubCode Options Window Help
> > - 1.0 + / 1.1 * RS RSP RA
1 class DNode(object):
2     def __init__(self,data):
3         self.data=data
4         self.next=None
5         self.prev=None
6
Python 2.7.15 Shell
File Edit Shell Debug Options Window Help
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:22:17) on win32
Type "copyright", "credits" or "license()" for more info
>>>
===== RESTART: C:/Users/ASUS/Downloads/idlex-1.18/idlex
>>> a=DNode(11)
>>> b=DNode(52)
>>> c=DNode(18)
>>> a.next=b
>>> c.prev=b
>>> print(a.data)
11
>>> print(a.next.data)
52
>>> print(c.prev.data)
52
>>> |

```

## SOAL-SOAL UNTUK MAHASISWA

### 1. ARRAY DUA DIMENSI, MATRIX YANG BERISI ANGKA-ANGKA.

#### A. FUNGSI CEK ISI DAN UKURAN MATRIX

```

2 def matrik(n):
3     panjang=len(n)
4     hasil=True
5     for x in n:
6         lebar=len(x)
7         if lebar != panjang:
8             hasil= False
9             break
10    for i in x:
11        if type(i) != int:
12            hasil = False
13            break
14    return hasil
15
16 def cek(n):
17     x = 0
18     y = 0
19     for i in n:
20         for j in i:
21             y+=1
22             if (str(j).isdigit()==False):
23                 print("Tidak semua isi matriks adalah angka")
24                 break
25             else:
26                 x+=1
27     if(x==y):
28         print("semua isi matriks adalah angka")
29
Python 2.7.15 Shell
File Edit Shell Debug Options Window Help
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:22:17) on win32
Type "copyright", "credits" or "license()" for more info
>>>
===== RESTART: C:/Users/ASUS/Downloads/idlex-1.18/idlex
>>> m1=[[2,3],[2,3]]
>>> m2=[[1,2,'yes'],[1,2.3]]
>>> matrik(m1)
True
>>> matrik(m2)
False
>>> cek(m1)
semua isi matriks adalah angka
>>> cek(m2)
Tidak semua isi matriks adalah angka
Tidak semua isi matriks adalah angka
>>> |

```

```
30 def ordo(n):
31     x,y = 0,0
32     for i in range(len(n)):
33         x+=1
34         y = len(n[i])
35     print("mempunyai ordo "+str(x)+"x"+str(y))
36
```

Code Browser Ln: 33 Col

```
>>> ordo(m1)
mempunyai ordo 2x2
```

**B. FUNGSI MENGAMBIL UKURAN MATRIX**

**C. FUNGSI MENJUMLAHKAN MATRIX**

```
37 def jumlah(n,m):
38     x,y = 0,0
39     for i in range(len(n)):
40         x+=1
41         y = len(n[i])
42     xy = [[0 for j in range(x)] for i in range(y)]
43
44     z = 0
45     if(len(n)==len(m)):
46         for i in range(len(n)):
47             if(len(n[i]) == len(m[i])):
48                 z+=1
49     if(z==len(n) and z==len(m)):
50         print("ukuran sama")
51         for i in range(len(n)):
52             for j in range(len(n[i])):
53                 xy[i][j] = n[i][j] + m[i][j]
54         print(xy)
55     else:
56         print("ukuran beda")
57
58 def kali(n,m):
```

Code Browser Ln: 38

```
>>> m3=[[1,2],[3,4]]
>>> m4=[[5,6],[7,8]]
>>> jumlah(m3,m4)
ukuran sama
[[6, 8], [10, 12]]
```

```

81 def det(A, total=0):
82     x = len(A[0])
83     z = 0
84     for i in range(len(A)):
85         if (len(A[i]) == x):
86             z+=1
87     if(z == len(A)):
88         if(x==len(A)):
89             indices = list(range(len(A)))
90             if len(A) == 2 and len(A[0]) == 2:
91                 val = A[0][0] * A[1][1] - A[1][0] * A[0][1]
92                 return val
93             for fc in indices:
94                 As = A
95                 As = As[1:]
96                 height = len(As)
97                 for i in range(height):
98                     As[i] = As[i][0:fc] + As[i][fc+1:]
99                 sign = (-1) ** (fc % 2)
100                 sub_det = determHitung(As)
101                 total += sign * A[0][fc] * sub_det
102             else:
103                 return "tidak bisa dihitung determinan, bukan matrix bujursangkar"
104         else:
105             return "tidak bisa dihitung determinan, bukan matrix bujursangkar"
106     return total
107

```

Code Browser Ln: 89

```

>>> det(m3)
-2
>>> det(m4)
-2
>>>

```

#### D. FUNGSI MENHITUNG DETERMINAN MATRIX

### 2. TERKAIT MATRIX DAN LIST COMPREHENSION

```

110 def buatNol(n,m=None):
111     if(m==None):
112         m=n
113     print("membuat matriks 0 dengan ordo "+str(n)+"x"+str(m))
114     print([[0 for j in range(m)] for i in range(n)])
115
116 def buatIden(n):
117     print("membuat matriks identitas dengan ordo"+str(n)+"x"+str(n))
118     print([[1 if j==i else 0 for j in range(n)] for i in range(n)])

```

Code Browser Ln: 116

```

>>> buatNol(2,3)
membuat matriks 0 dengan ordo 2x3
[[0, 0, 0], [0, 0, 0]]
>>> buatIden(3)
membuat matriks identitas dengan ordo3x3
[[1, 0, 0], [0, 1, 0], [0, 0, 1]]
>>>

```

```
125 class LinkedList:
126     def __init__(self):
127         self.head = None
128     def tambahDepan(self, new_data):
129         new_node = Node(new_data)
130         new_node.next = self.head
131         self.head = new_node
132     def tambahAkhir(self, data):
133         if (self.head == None):
134             self.head = Node(data)
135         else:
136             current = self.head
137             while (current.next != None):
138                 current = current.next
139             current.next = Node(data)
140         return self.head
141     def tambah(self, data, pos):
142         node = Node(data)
143         if not self.head:
144             self.head = node
145         elif pos==0:
146             node.next = self.head
147             self.head = node
148         else:
149             prev = None
150             current = self.head
151             current_pos = 0
152             while (current_pos < pos) and current.next:
153                 prev = current
154                 current = current.next
155                 current_pos +=1
156             prev.next = node
157             node.next = current
158         return self.head
159     def hapus(self, position):
160         if self.head == None:
161             return
162         temp = self.head
163         if position == 0:
164             self.head = temp.next
165             temp = None
166             return
167         for i in range(position -1 ):
168             temp = temp.next
169             if temp is None:
170                 break
171         if temp is None:
172             return
173         if temp.next is None:
174             return
175         next = temp.next.next
176         temp.next = None
177         temp.next = next
178     def cari(self, x):
179         current = self.head
180         while current != None:
181             if current.data == x:
182                 return "True"
183             current = current.next
184         return "False"
185     def display(self):
186         current = self.head
187         while current is not None:
188             current = current.next
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

### 3. TERKAIT LINKED LIST

```

191 class Node:
192     def __init__(self, data):
193         self.data = data
194         self.prev = None
195 class DoublyLinkedList:
196     def __init__(self):
197         self.head = None
198     def awal(self, new_data):
199         print("menambah pada awal", new_data)
200         new_node = Node(new_data)
201         new_node.next = self.head
202         if self.head is not None:
203             self.head.prev = new_node
204         self.head = new_node
205     def akhir(self, new_data):
206         print("menambah pada akhir", new_data)
207         new_node = Node(new_data)
208         new_node.next = None
209         if self.head is None:
210             new_node.prev = None
211             self.head = new_node
212             return
213         last = self.head
214         while(last.next is not None):
215             last = last.next
216         last.next = new_node
217         new_node.prev = last
218         return
219     def printList(self, node):
220         print("\nDari Depan :")
221         while(node is not None):
222             print(" % d" %(node.data))
223             last = node
224             node = node.next
225         print("\nDari Belakang :")
226         while(last is not None):
227             print(" % d" %(last.data))
228             last = last.prev
229
'True'
>>>
===== RESTART: C:/Users/ASUS/Do
>>> list=DoublyLinkedList()
>>> list.awal(1)
('menambah pada awal', 1)
>>> list.awal(2)
('menambah pada awal', 2)
>>> list.awal(3)
('menambah pada awal', 3)
>>> list.akhir(9)
('menambah pada akhir', 9)
>>> list.akhir(8)
('menambah pada akhir', 8)
>>> list.printList(list.head)

Dari Depan :
3
2
1
9
8

Dari Belakang :
8
9
1
2
3
>>>
GUI: OFF (TK)

```

#### 4. TERKAIT DOUBLY LINKED LIST