Nama : Aviza Ayuni Wulan

NIM : L200180187 / G

Praktikum Algostruk dan Struktur Data Modul 6

- 1. Ubah kode mergeSort dan quickSort agar bias mengurutkan list yang berisi object-object mhsTIF
 - mergeSort

```
1_mergeSort.py - E:/INFORMATICS/ASD/1_mergeSort.py (2.7.15)
                                                                                                         ×
3_mergeSort.py - E:/INFORMATICS/ASD/1_mergeSort
                                          File Edit Format Run Options Window Help
File Edit Format Run Options Window Help
class MhsTIF():
                                                  i = 0; j=0; k=0
   def __init__(self, nim):
                                                  while i < len(separuhkiri) and j < len(separuhkanan):</pre>
        self.nim = nim
                                                     if separuhkiri[i] < separuhkanan[j]:</pre>
                                                         A[k] = separuhkiri[i]
    def __str__(self):
                                                          i = i + 1
        return str(self.nim)
                                                      else:
                                                         A[k] = separuhkanan[j]
m0 = MhsTIF(10)
                                                          j = j + 1
ml = MhsTIF(51)
                                                      k=k+1
m2 = MhsTIF(2)
m3 = MhsTIF(18)
                                                  while i < len(separuhkiri):</pre>
m4 = MhsTIF(4)
                                                     A[k] = separuhkiri[i]
m5 = MhsTIF(31)
                                                      i = i + 1
m6 = MhsTIF(13)
                                                      k=k+1
m7 = MhsTIF(5)
m8 = MhsTIF(23)
                                                  while j < len(separuhkanan):</pre>
m9 = MhsTIF(64)
                                                      A[k] = separuhkanan[j]
                                                      j = j + 1
m10 = MhsTIF(29)
                                                      k=k+1
m0.next = m1
                                              #print("Menggabungkan", A)
ml.next = m2
                                          def convert(arr, obj):
m2.next = m3
                                             hasil=[]
m3.next = m4
                                              for x in range (len(arr)):
m4.next = m5
                                                  for i in range (len(arr)):
m5.next = m6
                                                     if arr[x] == obj[i].nim:
m6.next = m7
                                                         hasil.append(obj[i])
m7.next = m8
                                              return hasil
m8.next = m9
m9.next = m10
                                          Daftar = [m0, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10]
                                          A = []
def mergeSort(A):
                                          for x in Daftar:
    #print("Membelah
                                             A.append(x.nim)
    if len(A) > 1:
        mid = len(A) // 2
                                          print("MERGE SORT")
        separuhkiri = A[:mid]
                                          mergeSort(A)
        separuhkanan = A[mid:]
                                          for x in convert(A, Daftar):
                                             print (x.nim)
        mergeSort(separuhkiri)
        mergeSort (separuhkanan)
```

```
Python 2.7.15 Shell
                                                                        \times
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 (In -
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
======= RESTART: E:/INFORMATICS/ASD/1_mergeSort.py =========
MERGE SORT
4
5
10
13
18
23
29
31
51
64
>>>
```

QuickSort

```
1_QuickSort.py - E:/INFORMATICS/prak ASD/Modul6/1_QuickSort.py (2.7.15)
File Edit Format Run Options Window Help
class MhsTIF():
    def __init__(self, nim):
    self.nim = nim
   def __str__(self):
    return str(self.nim)
m0 = MhsTIF(10)
ml = MhsTIF(51)
m2 = MhsTIF(2)
m3 = MhsTIF(18)
m4 = MhsTIF(4)
m5 = MhsTIF(31)
m6 = MhsTIF(13)
m7 = MhsTIF(5)
m8 = MhsTIF(23)
m9 = MhsTIF(64)
m10 = MhsTIF(29)
m0.next = ml
ml.next = m2
m2.next = m3
m3.next = m4
m4.next = m5
m5.next = m6
m6.next = m7
m7.next = m8
m8.next = m9
m9.next = m10
def partisi(A, awal, akhir):
    nilaipivot = A[awal]
    penandakiri = awal + 1
    penandakanan = akhir
    selesai = False
    while not selesai:
         while penandakiri <= penandakanan and A[penandakiri] <= nilaipivot:
    penandakiri = penandakiri + 1
                                                                                  Ln: 85 Col: 0
```

```
*1_QuickSort.py - E:/INFORMATICS/prak ASD/Modul6/1_QuickSort.py (2.7.15)*
File Edit Format Run Options Window Help
            penandakiri = penandakiri + 1
        while penandakanan >= penandakiri and A[penandakanan] >= nilaipivot
            penandakanan = penandakanan - 1
        if penandakanan < penandakiri:</pre>
            selesai = True
        else:
             temp = A[penandakiri]
            A[penandakiri] = A[penandakanan]
A[penandakanan] = temp
    temp = A[awal]
    A[awal] = A[penandakanan]
    A[penandakanan] = temp
    return penandakanan
def quickSortBantu(A, awal, akhir):
    if awal < akhir:</pre>
       titikBelah = partisi(A, awal, akhir)
        quickSortBantu(A, awal, titikBelah-1)
        quickSortBantu(A, titikBelah+1, akhir)
def quickSort(A):
    quickSortBantu (A, 0, len(A)-1)
def convert(arr, obj):
   hasil=[]
    for x in range (len(arr)):
        for i in range (len(arr)):
    if arr[x] == obj[i].nim:
                hasil.append(obj[i])
    return hasil
Daftar = [m0, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10]
A = []
for x in Daftar:
   A.append(x.nim)
print("QUICK SORT")
quickSort(A)
for x in convert(A, Daftar):
    print (x.nim)
                                                                          Ln: 54 Col: 26
```

```
Python 2.7.15 Shell
                                                                          X
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 (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
====== RESTART: E:/INFORMATICS/prak ASD/Modul6/1_QuickSort.py ========
QUICK SORT
2
4
5
10
13
18
23
29
31
51
64
>>>
```

2. Menulis pakai bolpen merah dan biru

3. Uji kecepatan

```
No3.py - E:/INFORMATICS/prak ASD/Modul6/No3.py (2.7.15)
                                                                  No3.py - E:/INFORMATICS/prak ASD/Modul6/No3.py (2.7.15)
File Edit Format Run Options Window Help
                                                                  File Edit Format Run Options Window Help
from time import time as detak
                                                                       return S
from random import shuffle as kocok
import time
                                                                  def mergeSort(A):
                                                                       #print("Membelah
                                                                       if len(A) > 1:
mid = len(A) // 2
def swap(A, p, q):
    tmp = A[p]
A[p] = A[q]
A[q] = tmp
                                                                            separuhkiri = A[:mid]
separuhkanan = A[mid:]
def cariPosisiYangTerkecil(A, dariSini, sampaiSini):
                                                                           mergeSort(separuhkiri)
    posisiYangTerkecil = dariSini
                                                                            mergeSort (separuhkanan)
    for i in range(dariSini+1, sampaiSini):
    if A[i] < A[posisiYangTerkecil]:</pre>
                                                                            i = 0; j=0; k=0
             posisiYangTerkecil = i
                                                                            while i < len(separuhkiri) and j < len(separuhkanan):</pre>
    return posisiYangTerkecil
                                                                                if separuhkiri[i] < separuhkanan[j]:</pre>
                                                                                    A[k] = separuhkiri[i]
i = i + 1
def bubbleSort(S):
    n = len(S)
                                                                                else:
     for i in range (n-1):
                                                                                    A[k] = separuhkanan[j]
         for j in range (n-i-1):
    if S[j] > S[j+1]:
                                                                                     j = j + 1
                                                                                k=k+1
                   swap(S,j,j+1)
    return S
                                                                            while i < len(separuhkiri):</pre>
                                                                                A[k] = separuhkiri[i]
def selectionSort(S):
                                                                                i = i + 1
                                                                                k=k+1
    n = len(S)
         indexKecil = cariPosisiYangTerkecil(S, i, n)
if indexKecil != i:
                                                                            while j < len(separuhkanan):
                                                                                A[k] = separuhkanan[j]
                                                                                j = j + 1
              swap(S, i, indexKecil)
    return S
                                                                                 k=k+1
                                                                       #print("Menggabungkan", A)
def insertionSort(S):
                                                                  def partisi(A, awal, akhir):
    nilaipivot = A[awal]
    n = len(S)
    for i in range(1, n):
        nilai = S[i]
         pos = i
while pos > 0 and nilai < S[pos -1]:
    S[pos] = S[pos-1]</pre>
                                                                       penandakiri = awal + 1
                                                                      penandakanan = akhir
         pos = pos - 1
S[pos] = nilai
                                                                       selesai = False
                                                                       while not selesai:
```

```
No3.py - E:/INFORMATICS/prak ASD/Modul6/No3.py (2.7.15)
File Edit Format Run Options Window Help
       while penandakiri <= penandakanan and A[penandakiri] <= nilaipivot:
          penandakiri = penandakiri + 1
       while penandakanan >= penandakiri and A[penandakanan] >= nilaipivot:
          penandakanan = penandakanan - 1
       if penandakanan < penandakiri:</pre>
          selesai = True
       else:
          temp = A[penandakiri]
          A[penandakiri] = A[penandakanan]
          A[penandakanan] = temp
   temp = A[awal]
   A[awal] = A[penandakanan]
   A[penandakanan] = temp
   return penandakanan
def quickSortBantu(A, awal, akhir):
   if awal < akhir:</pre>
       titikBelah = partisi(A, awal, akhir)
       quickSortBantu(A, awal, titikBelah-1)
       quickSortBantu(A, titikBelah+1, akhir)
def quickSort(A):
   quickSortBantu (A, 0, len(A)-1)
daftar = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]
print (bubbleSort(daftar))
print (selectionSort(daftar))
print (insertionSort(daftar))
mergeSort(daftar)
print (daftar)
quickSort (daftar)
print (daftar)
k = [[i] for i in range(1, 6001)]
kocok(k)
u bub = k[:]
u sel = k[:]
u_ins = k[:]
u mrg = k[:]
u \neq k[:]
aw=detak();bubbleSort(u bub);ak=detak();print("bubble: %g detik" %(ak-aw));
aw=detak();selectionSort(u_sel);ak=detak();print("selection: %g detik" %(ak-aw))
aw=detak();insertionSort(u ins);ak=detak();print("insertion: %g detik" %(ak-aw))
aw=detak(); mergeSort(u mrg); ak=detak(); print("merge: %g detik" %(ak-aw));
aw=detak();quickSort(u_qck);ak=detak();print("quick: %g detik" %(ak-aw));
                                                                                           Ln: 135 Col: 0
```

```
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
bubble: 5.261 detik
selection: 2.129 detik
insertion: 2.279 detik
merge: 0.03 detik
quick: 0.027 detik
>>> |
```

- 4. Diberikan list L
- 5. Tingkatkan efisiensi mergeSort dengan tidak menggunakan operator Slice dan lalu mempass indek awal dan index akhir bersama list-nya saat kita memanggil mergeSort secara rekursif.

```
No5.py - E:/INFORMATICS/prak ASD/Modul6/No5.py (2.7.15)
                                                                     No5.py - E:/INFORMATICS/prak ASD/Modul6/No5.py (2.7.15)
File Edit Format Run Options Window Help
                                                                     File Edit Format Run Options Window Help
                                                                    m9 = MhsTIF("Winda", 81, "Pontianak", 250000)
ml0 = MhsTIF("Qina", 43, "Lombok", 550000)
class MhsTIF():
     def init (self, nama, nim, kota, us):
          self.nama = nama
          self.nim = nim
                                                                     daftar = [m0, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10]
          self.kota = kota
                                                                     def cetak(A):
         self.us = us
                                                                         for i in A:
                                                                              print (i)
           str (self):
          s = self.nama +', NIM '+str(self.nim)\
              +'. Tinggal di '+ self.kota \
                                                                     def mergeSort2(A, awal, akhir):
               +'. Uang saku Rp. '+ str(self.us)\
                                                                         mid = (awal+akhir)//2
               +' tiap bulannya.'
                                                                          if awal < akhir:
                                                                              mergeSort2(A, awal, mid)
                                                                              mergeSort2(A, mid+1, akhir)
     def ambilNama(self):
                                                                         a, f, l = 0, awal, mid+1
          return self.nama
                                                                                [None] * (akhir - awal + 1)
     def ambilNim(self):
                                                                          while f <= mid and l <= akhir:
         return self.nim
                                                                                \begin{tabular}{ll} if & A[f].ambilUangSaku() < & A[l].ambilUangSaku() : \\ \end{tabular} 
     def ambilUangSaku(self):
                                                                                    tmp[a] = A[f]
         return self.us
                                                                                   f += 1
m0 = MhsTIF("Alfa", 76, "Banyuwangi", 249000)
                                                                               else:
m0 - MnsTIF("Alia", 76, "Banyuwangi", 249000)
m1 = MnsTIF("Pita", 53, "Purwokerto", 234000)
m2 = MnsTIF("Octa", 37, "Purworejo", 220000)
m3 = MnsTIF("Ila", 49, "Surakarta", 232000)
m4 = MnsTIF("Uni", 46, "Demak", 300000)
m5 = MnsTIF("Yeri", 31, "Cilacap", 250000)
                                                                                   tmp[a] = A[1]
                                                                         if f <= mid:
m6 = MhsTIF("Tisa", 60, "Kutai", 245000)
                                                                               tmp[a:] = A[f:mid+1]
m7 = MhsTIF("Roro", 91, "Lembang", 231000)
                                                                         if 1 <= akhir:
m8 = MhsTIF("Elvi", 15, "Bogor", 289000)
m9 = MhsTIF("Winda", 81, "Pontianak", 250000)
ml0 = MhsTIF("Qina", 43, "Lombok", 550000)
                                                                               tmp[a:] = A[1:akhir+1]
                                                                          while awal <= akhir:
daftar = [m0, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10]
                                                                              A[awal] = tmp[a]
                                                                               awal += 1
def cetak(A):
    for i in A:
                                                                               a += 1
         print (i)
                                                                     def mergeSort(A):
                                                                          mergeSort2(A, 0, len(A)-1)
def mergeSort2(A, awal, akhir):
     mid = (awal+akhir)//2
```

```
======= RESTART: E:/INFORMATICS/prak ASD/Modul6/No5.py ===========
>>> cetak(daftar)
Alfa, NIM 76. Tinggal di Banyuwangi. Uang saku Rp. 249000 tiap bulannya.
Pita, NIM 53. Tinggal di Purwokerto. Uang saku Rp. 234000 tiap bulannya.
Octa, NIM 37. Tinggal di Purworejo. Uang saku Rp. 220000 tiap bulannya.
Ila, NIM 49. Tinggal di Surakarta. Uang saku Rp. 232000 tiap bulannya.
Uni, NIM 46. Tinggal di Demak. Uang saku Rp. 300000 tiap bulannya.
Yeri, NIM 31. Tinggal di Cilacap. Uang saku Rp. 250000 tiap bulannya.
Tisa, NIM 60. Tinggal di Kutai. Uang saku Rp. 245000 tiap bulannya.
Roro, NIM 91. Tinggal di Lembang. Uang saku Rp. 231000 tiap bulannya.
Elvi, NIM 15. Tinggal di Bogor. Uang saku Rp. 289000 tiap bulannya.
Winda, NIM 81. Tinggal di Pontianak. Uang saku Rp. 250000 tiap bulannya.
Qina, NIM 43. Tinggal di Lombok. Uang saku Rp. 550000 tiap bulannya.
>>> mergeSort(daftar)
>>> cetak(daftar)
Octa, NIM 37. Tinggal di Purworejo. Uang saku Rp. 220000 tiap bulannya.
Roro, NIM 91. Tinggal di Lembang. Uang saku Rp. 231000 tiap bulannya.
Ila, NIM 49. Tinggal di Surakarta. Uang saku Rp. 232000 tiap bulannya.
Pita, NIM 53. Tinggal di Purwokerto. Uang saku Rp. 234000 tiap bulannya.
Tisa, NIM 60. Tinggal di Kutai. Uang saku Rp. 245000 tiap bulannya.
Alfa, NIM 76. Tinggal di Banyuwangi. Uang saku Rp. 249000 tiap bulannya.
Winda, NIM 81. Tinggal di Pontianak. Uang saku Rp. 250000 tiap bulannya.
Yeri, NIM 31. Tinggal di Cilacap. Uang saku Rp. 250000 tiap bulannya.
Elvi, NIM 15. Tinggal di Bogor. Uang saku Rp. 289000 tiap bulannya.
Uni, NIM 46. Tinggal di Demak. Uang saku Rp. 300000 tiap bulannya.
Qina, NIM 43. Tinggal di Lombok. Uang saku Rp. 550000 tiap bulannya.
>>>
```

 Meningkatkan efisiensi program quicksort dengan memakai metode median dari tiga untuk memilih pivot.

```
No6.py - E:/INFORMATICS/prak ASD/Modul6/No6.py (2.7.15)
 No6.py - E:/INFORMATICS/prak ASD/Modul6/No6.py (2.7.15)
                                                                                 File Edit Format Run Options Window Help
 File Edit Format Run Options Window Help
                                                                                 m7 = MhsTIF("Roro", 91, "Lembang", 231000)
 class MhsTIF():
                                                                                 m8 = MhsTIF("Elvi", 15, "Bogor", 289000)

m9 = MhsTIF("Winda", 81, "Pontianak", 250000)

m10 = MhsTIF("Qina", 43, "Lombok", 550000)
      def __init__(self, nama, nim, kota, us):
            self.nama = nama
            self.nim = nim
            self.kota = kota
                                                                                 daftar = [m0, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10]
            self.us = us
                                                                                 A = []
      def __str__(self):
            s = self.nama +', NIM '+str(self.nim)\
                                                                                 for i in daftar:
                 +'. Tinggal di '+ self.kota \
                                                                                      A.append(i.nama)
                  +'. Uang saku Rp. '+ str(self.us)\
                  +' tiap bulannya.'
                                                                                 def cetak():
                                                                                      for i in A:
            return s
                                                                                            print(i)
      def ambilNama(self):
                                                                                 def quickSort(arr):
            return self.nama
      def ambilNim(self):
                                                                                       kurang = []
                                                                                       pivotList = []
            return self.nim
      def ambilUangSaku(self):
                                                                                       lebih = []
            return self.us
                                                                                       if len(arr) <= 1:</pre>
                                                                                            return arr
m0 = MhsTIF("Alfa", 76, "Banyuwangi", 249000)
ml = MhsTIF("Pita", 53, "Purwokerto", 234000)
m2 = MhsTIF("Octa", 37, "Purworejo", 220000)
m3 = MhsTIF("Ila", 49, "Surakarta", 232000)
m4 = MhsTIF("Uni", 46, "Demak", 300000)
m5 = MhsTIF("Yeri", 31, "Cilacap", 250000)
m6 = MhsTIF("Tisa", 60, "Kutai", 245000)
m7 = MhsTIF("Roro", 91, "Lembang", 231000)
m8 = MhsTIF("Elvi", 15, "Bogor", 289000)
m9 = MhsTIF("Winda", 81, "Pontianak", 250000)
m10 = MhsTIF("Qina", 43, "Lombok", 550000)
                                                                                       else:
                                                                                            pivot = arr[0]
                                                                                             for i in arr:
                                                                                                 if i < pivot:
                                                                                                        kurang.append(i)
                                                                                                  elif i > pivot:
                                                                                                       lebih.append(i)
                                                                                                  else:
                                                                                            pivotList.append(i)
kurang = quickSort(kurang)
                                                                                            lebih = quickSort(lebih)
                                                                                             return kurang + pivotList + lebih
 daftar = [m0, m1, m2, m3, m4, m5, m6, m7, m8, m9, m10]
                                                                                  print("Sebelum diurutkan")
 A = []
                                                                                 cetak()
                                                                                  orint("\nSetelah diurutkan")
 for i in daftar:
                                                                                 quickSort(A)
      A.append(i.nama)
                                                                                 cetak()
 def cetak():
```

```
======= RESTART: E:/INFORMATICS/prak ASD/Modul6/No6.py ==========
Sebelum diurutkan
Alfa
Pita
Octa
Ila
Uni
Yeri
Tisa
Roro
Elvi
Winda
Qina
Setelah diurutkan
Alfa
Pita
Octa
Ila
Uni
Yeri
Tisa
Roro
Elvi
Winda
Oina
>>>
```

7. Uji kecepatan keduanya dan perbandingkan juga dgn kode awalnya.

```
No7.py - E:/INFORMATICS/prak ASD/Modul6/No7.py (2.7.15)
 File Edit Format Run Options Window Help
 from time import time as detak
 from random import shuffle as kocok
 import time
 def mergeSort(A):
      #print("Membelah
      if len(A) > 1:
    mid = len(A) // 2
    separuhkiri = A[:mid]
           separuhkanan = A[mid:]
          mergeSort(separuhkiri)
          mergeSort (separuhkanan)
           i = 0; j=0; k=0
           while i < len(separuhkiri) and j < len(separuhkanan):
                if separuhkiri[i] < separuhkanan[j]:
   A[k] = separuhkiri[i]
   i = i + 1</pre>
                    A[k] = separuhkanan[j]
                j = j + 1
k=k+1
           while i < len(separuhkiri):</pre>
                A[k] = separuhkiri[i]
i = i + 1
k=k+1
           while j < len(separuhkanan):</pre>
               A[k] = separuhkanan[j]
j = j + 1
                k=k+1
      #print("Menggabungkan", A)
 def partisi(A, awal, akhir):
    nilaipivot = A[awal]
      penandakiri = awal + 1
penandakanan = akhir
```

```
selesai = False
while not selesai:
          while penandakiri <= penandakanan and A[penandakiri] <= nilaipivot: penandakiri = penandakiri + 1
          while penandakanan >= penandakiri and A[penandakanan] >= nilaipivot:
              penandakanan = penandakanan - 1
          if penandakanan < penandakiri:</pre>
               selesai = True
          else:
               temp = A[penandakiri]
               A[penandakiri] = A[penandakanan]
A[penandakanan] = temp
    temp = A[awal]
A[awal] = A[penandakanan]
A[penandakanan] = temp
     return penandakanan
def quickSortBantu(A, awal, akhir):
     if awal < akhir:
         titikBelah = partisi(A, awal, akhir)
          quickSortBantu(A, awal, titikBelah-1)
quickSortBantu(A, titikBelah+1, akhir)
def quickSort(A):
     quickSortBantu (A, 0, len(A)-1)
def mergeSort2(A, awal, akhir):
    mid = (awal+akhir)//2
if awal < akhir:
          mergeSort2(A, awal, mid)
          mergeSort2(A, mid+1, akhir)
     a, f, 1 = 0, awal, mid+1 
tmp = [None] * (akhir - awal + 1)
                                                                                              Ln: 5 Col: 17
```

```
No7.py - E:/INFORMATICS/prak ASD/Modul6/No7.py (2.7.15)
File Edit Format Run Options Window Help
    a, f, 1 = 0, awal, mid+1
tmp = [None] * (akhir - awal + 1)
while f <= mid and 1 <= akhir:</pre>
        if A[f] < A[1]:
            tmp[a] = A[f]
             f += 1
        else:
             tmp[a] = A[1]
        1 += 1
a += 1
    if f <= mid:</pre>
         tmp[a:] = A[f:mid+1]
    if 1 <= akhir:</pre>
         tmp[a:] = A[1:akhir+1]
    while awal <= akhir:
        A[awal] = tmp[a]
        awal += 1
        a += 1
def mergeSortNew(A):
    mergeSort2(A, 0, len(A)-1)
def quickSortNew(arr):
   kurang = []
    pivotList = []
    lebih = []
    if len(arr) <= 1:</pre>
         return arr
    else:
        pivot = arr[0]
         for i in arr:
            if i < pivot:
                 kurang.append(i)
             elif i > pivot:
                 lebih.append(i)
                                                                                    Ln: 54 Col: 33
```

```
No7.py - E:/INFORMATICS/prak ASD/Modul6/No7.py (2.7.15)
                                                                            X
 File Edit Format Run Options Window Help
    pivotList = []
    lebih = []
    if len(arr) <= 1:
        return arr
    else:
        pivot = arr[0]
        for i in arr:
           if i < pivot:</pre>
               kurang.append(i)
            elif i > pivot:
               lebih.append(i)
            else:
               pivotList.append(i)
        kurang = quickSortNew(kurang)
        lebih = quickSortNew(lebih)
        return kurang + pivotList + lebih
daftar = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]
mergeSort(daftar)
print (daftar)
quickSort (daftar)
print (daftar)
mergeSortNew(daftar)
print (daftar)
quickSortNew(daftar)
print (daftar)
k = [[i] for i in range(1, 6001)]
kocok(k)
u_mrg = k[:]
u_qck = k[:]
u_mrgNew = k[:]
u_qckNew = k[:]
aw=detak();mergeSort(u_mrg);ak=detak();print("merge: %g detik" %(ak-aw));
aw=detak();quickSort(u qck);ak=detak();print("quick: %g detik" %(ak-aw));
aw=detak(); mergeSortNew(u mrgNew); ak=detak(); print("merge New: %g detik" %(ak-aw
aw=detak();quickSortNew(u_qckNew);ak=detak();print("quick New: %g detik" %(ak-aw
                                                                      Ln: 84 Col: 18
Hasil:
            ====== RESTART: E:/INFORMATICS/prak ASD/Modul6/No7.py =========
 [2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
 [2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
 [2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
 [2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
merge: 0.0340002 detik
```

8. Buat versi linked list untuk program mergeSort di atas

quick: 0.0180001 detik merge New: 0.049 detik quick New: 0.026 detik

>>>

```
🕞 No8.py - E:/INFORMATICS/prak ASD/Modul6/No8.py (2.7.15 🍃 No8.py - E:/INFORMATICS/prak ASD/Modul6/No8.py (2.7.15)
File Edit Format Run Options Window Help
                                              File Edit Format Run Options Window Help
class Node():
    def __init__(self, data, tautan=None):
                                                       A = A
        self.data = data
                                                   if len(A) > 1:
        self.tautan = tautan
                                                       mid = len(A) // 2
                                                       separuhkiri = A[:mid]
def cetak(head):
                                                       separuhkanan = A[mid:]
    curr = head
    while curr is not None:
                                                       mergeSortLL(separuhkiri)
        try:
                                                       {\tt mergeSortLL}\,({\tt separuhkanan})
           print (curr.data)
            curr = curr.tautan
                                                       i = 0; j=0; k=0
        except:
                                                       while i < len(separuhkiri) and j < len(separuhkanan):</pre>
            pass
                                                           if separuhkiri[i] < separuhkanan[j]:</pre>
                                                              A[k] = separuhkiri[i]
a = Node(1)
                                                               i = i + 1
b = Node(3)
c = Node(5)
                                                              A[k] = separuhkanan[j]
d = Node(7)
                                                               j = j + 1
e = Node(2)
f = Node(4)
                                                           k=k+1
g = Node(6)
                                                       while i < len(separuhkiri):</pre>
                                                           A[k] = separuhkiri[i]
a.tautan = b
                                                           i = i + 1
b.tautan = c
                                                           k=k+1
c.tautan = d
d.tautan = e
                                                       while j < len(separuhkanan):</pre>
e.tautan = f
                                                           A[k] = separuhkanan[j]
f.tautan = g
                                                           j = j + 1
def mergeSortLL(A):
    linked = A
                                                  for x in A:
    try:
                                                       try:
        daftar = []
                                                          linked.data = x
        curr = A
                                                           linked = linked.tautan
        while curr:
                                                       except:
            daftar.append(curr.data)
                                                           pass
             curr = curr.tautan
        A = daftar
                                              mergeSortLL(a)
    except:
                                              cetak(a)
        A = A
```