Nama : Ulin Nuha Trisiana

NIM : L200180190

Praktikum ASD Kelas G

Soal-soal untuk Mahasiswa Modul 6

1. Merge Sort dan Quick Sort

Modul6_1.py - E:/Prak_Algostruk/Modul6_1.py (3.6.5)

File Edit Format Run Options Window Help

```
#Nomer 1
#Merge Sort

class MhsTIF():
    def __init__(self, nim):
        self.nim = nim

    def __str__(self):
        return str(self.nim)

a0 = MhsTIF(13)
a1 = MhsTIF(4)
a2 = MhsTIF(6)
a3 = MhsTIF(7)
a4 = MhsTIF(24)
a5 = MhsTIF(15)
a6 = MhsTIF(29)
a7 = MhsTIF(23)
a8 = MhsTIF(11)
a9 = MhsTIF(1)
a10 = MhsTIF(3)

a0.next = a1
a1.next = a2
a2.next = a3
a3.next = a4
a4.next = a5
a5.next = a6
a6.next = a7
a7.next = a8
a8.next = a9
a9.next = a10

def mergeSort(A):
    print("Membelah      ",A)
    if len(A) > 1:
        mid = len(A) // 2
        separuhkiri = A[:mid]
        separuhkanan = A[mid:]
        mergeSort(separuhkiri)
        mergeSort(separuhkanan)
```

Modul6_1.py - E:/Prak_Algostruk/Modul6_1.py (3.6.5)

File Edit Format Run Options Window Help

```
        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
            else:
                A[k] = separuhkanan[j]
                j = j + 1
            k = k + 1
        while i < len(separuhkiri):
            A[k] = separuhkiri[i]
            i = i + 1
            k = k + 1
        while j < len(separuhkanan):
            A[k] = separuhkanan[j]
            j = j + 1
            k = k + 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 = [a0, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10]
A = []

for x in Daftar:
    A.append(x.nim)

print("MERGE SORT")
mergeSort(A)
for x in convert(A, Daftar):
    print(x.nim)
```

Modul6_1.py - E:/Prak_Algostruk/Modul6_1.py (3.6.5)

File Edit Format Run Options Window Help

```python
#Quick Sort
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:
            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)

Daftar = [a0, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10]
A = []

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

for x in Daftar:
    A.append(x.nim)

print("QUICK SORT")
quickSort(A)
for x in convert(A, Daftar):
    print(x.nim)
```

Ln: 82 Col: 28

Output

Python 3.6.5 Shell

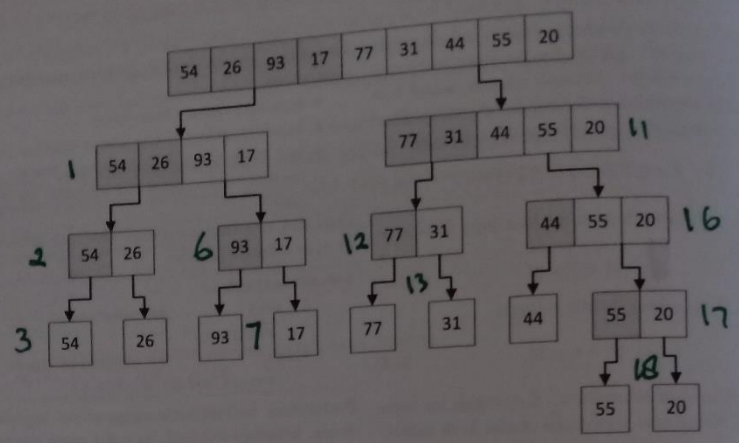
File Edit Shell Debug Options Window Help

```
==================== RESTART: E:/Prak_Algostruk/Modul6_1.py ====================
MERGE SORT
Membelah        [13, 4, 6, 7, 24, 15, 29, 23, 11, 1, 3]
Membelah        [13, 4, 6, 7, 24]
Membelah        [13, 4]
Membelah        [13]
Membelah        [4]
Membelah        [6, 7, 24]
Membelah        [6]
Membelah        [7, 24]
Membelah        [7]
Membelah        [24]
Membelah        [15, 29, 23, 11, 1, 3]
Membelah        [15, 29, 23]
Membelah        [15]
Membelah        [29, 23]
Membelah        [29]
Membelah        [23]
Membelah        [11, 1, 3]
Membelah        [11]
Membelah        [1, 3]
Membelah        [1]
Membelah        [3]
1
3
4
6
7
11
13
15
23
24
29
QUICK SORT
1
3
4
6
7
11
```

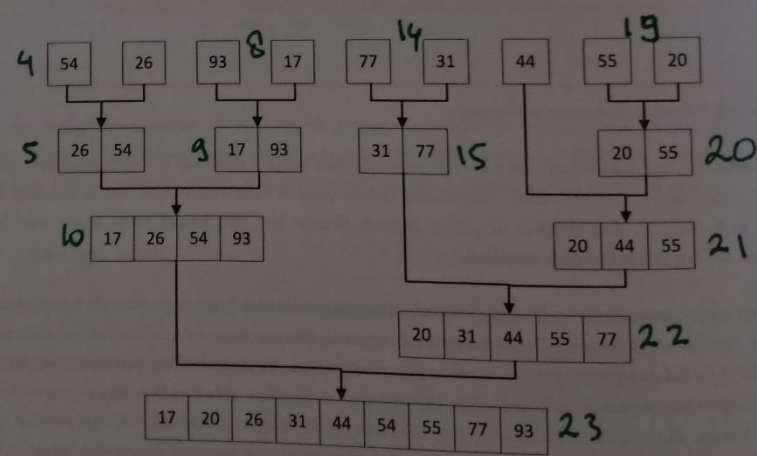
```
13
15
23
24
29
QUICK SORT
1
3
4
6
7
11
13
15
23
24
29
>>>
```

Ln: 145 Col: 4

2. Eksekusi proses pada gambar 6.1 dan 6.2



Gambar 6.1: Membelah list sampai tiap sub-list berisi satu elemen atau kosong. Sesudah itu digabung seperti ditunjukkan di Gambar 6.2.



Gambar 6.2: Menggabungkan list satu demi satu.

3. Uji kecepatan Merge Sort dan Quick Sort

Modul6_3.py - E:/Prak_Algostruk/Modul6_3.py (3.6.5)

File Edit Format Run Options Window Help

```python
#Nomer 3

from time import time as detak
from random import shuffle as kocok
import time

def swap(A, p, q):
    tmp = A[p]
    A[p] = A[q]
    A[q] = tmp

def cariPosisiYangTerkecil(A, dariSini, sampaiSini):
    posisiYangTerkecil = dariSini
    for i in range(dariSini+1, sampaiSini):
        if A[i] < A[posisiYangTerkecil]:
            posisiYangTerkecil = i
    return posisiYangTerkecil

def bubbleSort(S):
    n = len(S)
    for i in range (n-1):
        for j in range (n-i-1):
            if S[j] > S[j+1]:
                swap(S,j,j+1)
    return S

def selectionSort(S):
    n = len(S)
    for i in range(n-1):
        indexKecil = cariPosisiYangTerkecil(S, i, n)
        if indexKecil != i:
            swap(S, i, indexKecil)
    return S

def insertionSort(S):
    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]
```

Ln: 1 Col: 8

Modul6_3.py - E:/Prak_Algostruk/Modul6_3.py (3.6.5)

File Edit Format Run Options Window Help

```
            pos = pos - 1
        S[pos] = nilai
    return S

def mergeSort(A):
    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
            else:
                A[k] = separuhkanan[j]
                j = j + 1
            k=k+1
        while i < len(separuhkiri):
            A[k] = separuhkiri[i]
            i = i + 1
            k=k+1
        while j < len(separuhkanan):
            A[k] = separuhkanan[j]
            j = j + 1
            k=k+1

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:
            selesai = True
```

Ln: 1 Col: 8

Modul6_3.py - E:/Prak_Algostruk/Modul6_3.py (3.6.5)

File Edit Format Run Options Window Help

```
        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)

d = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]

print (bubbleSort(d))
print (selectionSort(d))
print (insertionSort(d))
mergeSort(d)
print (d)
quickSort(d)
print (d)

k =  [[i] for i in range(1, 6001)]
kocok(k)
u_bub = k[:]
u_sel = k[:]
u_ins = k[:]
u_mrg = k[:]
u_qck = 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));
```

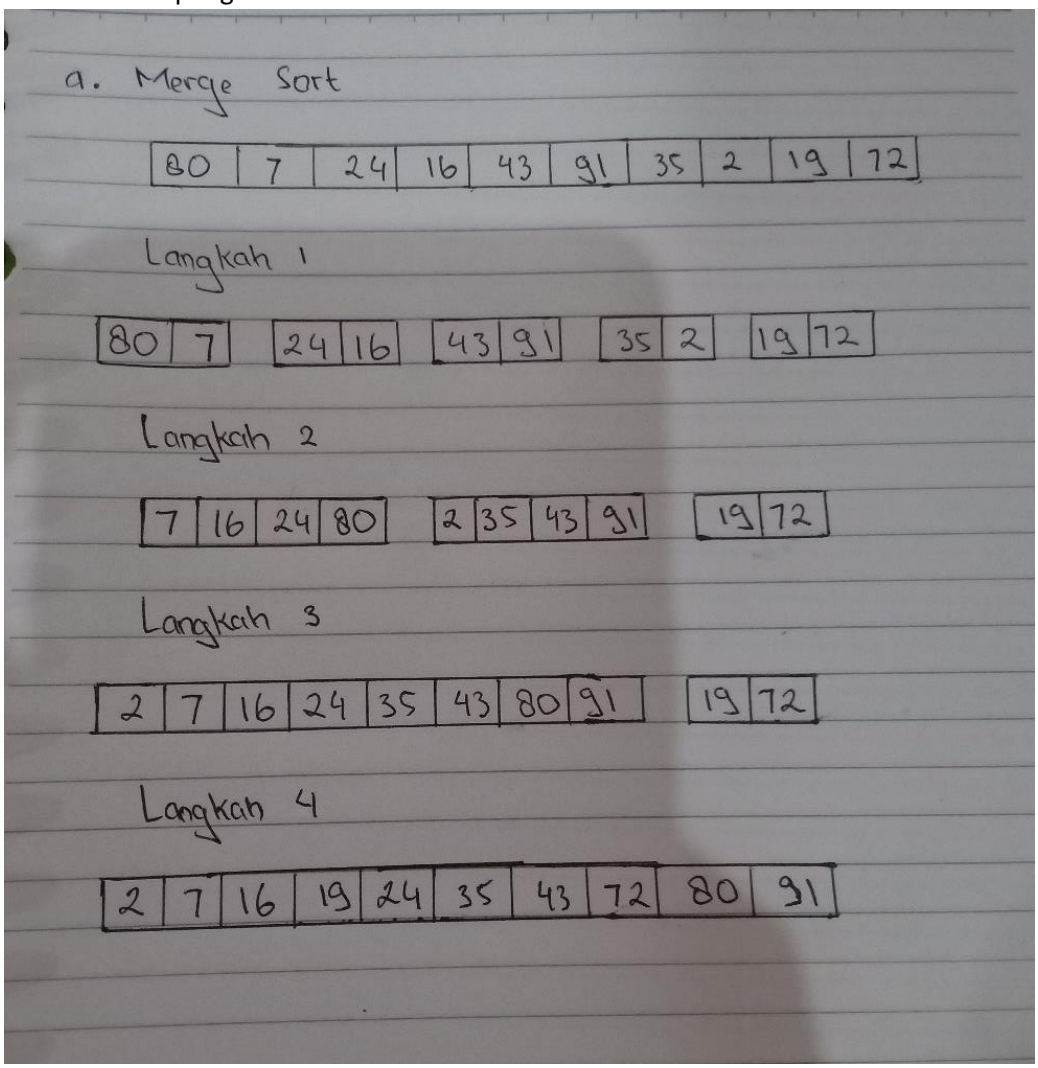
Ln: 1 Col: 8

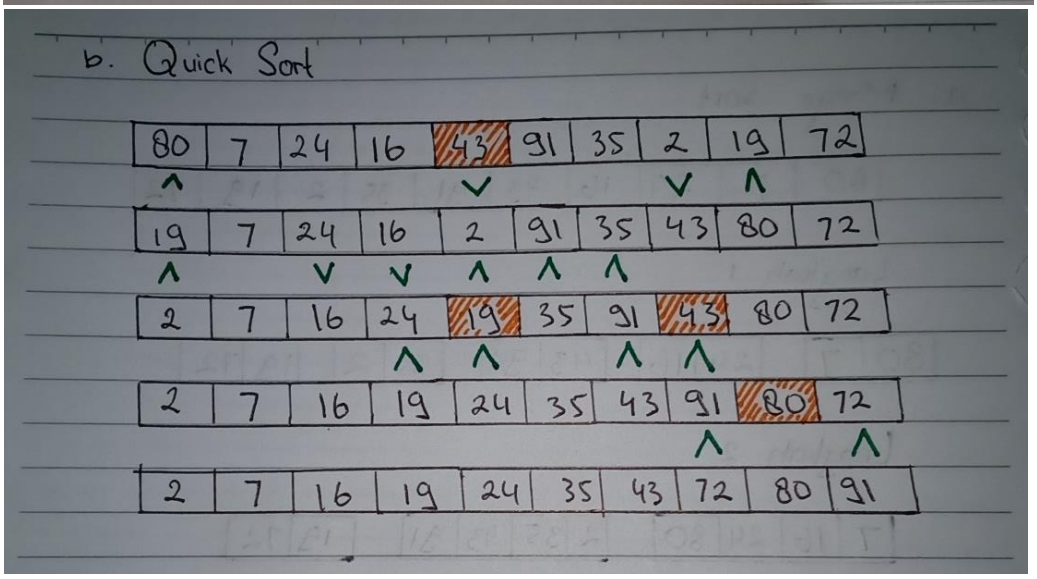
Output

```
=================== RESTART: E:/Prak_Algostruk/Modul6_3.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]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
bubble: 10.9094 detik
selection: 4.06581 detik
insertion: 5.07707 detik
merge: 0.0880663 detik
quick: 0.0628734 detik

>>>
```

4. Gambar trace pengurutan





5. Program Merge Sort tanpa operasi Slicing

Modul 6_5a.py - E:\Prak_Algostruk\Modul 6_5a.py (3.6.5)

File Edit Format Run Options Window Help

```python
#Nomer 5

import random
def _merge_sort(indices, the_list):
    start = indices[0]
    end = indices[1]
    half_way = (end - start)//2 + start
    if start < half_way:
        _merge_sort((start, half_way), the_list)
    if half_way + 1 <= end and end - start != 1:
       _merge_sort((half_way + 1, end), the_list)

    sort_sub_list(the_list, indices[0], indices[1])
    return the_list

def sort_sub_list(the_list, start, end):
    orig_start = start
    initial_start_second_list = (end - start)//2 + start + 1
    list2_first_index = initial_start_second_list
    new_list = []
    while start < initial_start_second_list and list2_first_index <= end:
        first1 = the_list[start]
        first2 = the_list[list2_first_index]
        if first1 > first2:
            new_list.append(first2)
            list2_first_index += 1
        else:
            new_list.append(first1)
            start += 1
    while start < initial_start_second_list:
        new_list.append(the_list[start])
        start += 1
    while list2_first_index <= end:
        new_list.append(the_list[list2_first_index])
        list2_first_index += 1
    for i in new_list:
        the_list[orig_start] = i
        orig_start += 1
    return the_list


def merge_sort(the_list):
    return _merge_sort((0, len(the_list) - 1), the_list)

print(merge_sort([13,45,12,3,10,2]))
```

Ln: 39 Col: 19

Output

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD6
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: E:\Prak_Algostruk\Modul 6_5a.py ==================
[2, 3, 10, 12, 13, 45]
>>>
```

6. Kode Quick Sort dengan memakai metode median-dari-tiga untuk memilih pivot

Modul6_6.py - E:/Prak_Algostruk/Modul6_6.py (3.6.5)

File Edit Format Run Options Window Help

```python
#Nomer 6

class MhsTIF():
    def __init__(self, nama, nim, kota, us):
        self.nama = nama
        self.nim = nim
        self.kota = kota
        self.us = us

    def __str__(self):
        s = self.nama + ', NIM ' + str(self.nim) \
            + '. Tinggal di ' + self.kota \
            + '. Uang saku Rp. ' + str(self.us) \
            + ' tiap bulannya.'
        return s

    def ambilNama(self):
        return self.nama

    def ambilNim(self):
        return self.nim

    def ambilUangSaku(self):
        return self.us

a1=MhsTIF("Anna",190,"Ngawi",250000)
a2=MhsTIF("Noer",207,"Surakarta",550000)
a3=MhsTIF("Kinan",167,"Ngawi",50000)
a4=MhsTIF("Nafiza",104,"Jakarta",100000)
a5=MhsTIF("Sari",132,"Jakarta",750000)
a6=MhsTIF("Andri",209,"Sragen",650000)
a7=MhsTIF("Fahrur",134,"Ngawi",8250000)
a8=MhsTIF("Sia",202,"Salatiga",400000)
a9=MhsTIF("Arif",213,"Ngawi",480000)
a10=MhsTIF("Supri",160,"Sragen",950000)
a11=MhsTIF("Erwan",215,"Salatiga",365000)

Daftar = [a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11]
A = []
```

```python
for i in Daftar:
    A.append(i.nama)

def cetak():
    for i in A:
        print(i)

def quickSort(arr):
    kurang = []
    pivotList = []
    lebih = []
    if len(arr) <= 1:
        return arr
    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

print("Sebelum diurutkan")
cetak()
print("\nSetelah diurutkan")
quickSort(A)
cetak()
```

Ln: 17 Col: 16

Output

```
=================== RESTART: E:/Prak_Algostruk/Modul6_6.py ===================
Sebelum diurutkan
Anna
Noer
Kinan
Nafiza
Sari
Andri
Fahrur
Sia
Arif
Supri
Erwan

Setelah diurutkan
Anna
Noer
Kinan
Nafiza
Sari
Andri
Fahrur
Sia
Arif
Supri
Erwan
>>>
```

7. Uji kecepatan

Modul6_7.py - E:/Prak_Algostruk/Modul6_7.py (3.6.5)

File Edit Format Run Options Window Help

```
#Nomer 7

from time import time as detak
from random import shuffle as kocok
import time

def mergeSort(A):
    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
            else:
                A[k] = separuhkanan[j]
                j = j + 1
            k = k + 1
        while i < len(separuhkiri):
            A[k] = separuhkiri[i]
            i = i + 1
            k = k + 1
        while j < len(separuhkanan):
            A[k] = separuhkanan[j]
            j = j + 1
            k = k + 1

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: 27 Col: 0

Modul6_7.py - E:/Prak_Algostruk/Modul6_7.py (3.6.5)

File Edit Format Run Options Window Help

```
        while penandakanan >= penandakiri and A[penandakanan] >= nilaipivot:
            penandakanan = penandakanan - 1
        if penandakanan < penandakiri:
            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, l = 0, awal, mid + 1
    tmp = [None] * (akhir - awal + 1)
    while f <= mid and l <= akhir:
        if A[f] < A[l]:
            tmp[a] = A[f]
            f += 1
        else:
            tmp[a] = A[l]
            l += 1
        a += 1
    if f <= mid:
        tmp[a:] = A[f:mid + 1]
```

Modul6_7.py - E:/Prak_Algostruk/Modul6_7.py (3.6.5)

File Edit Format Run Options Window Help

```
    if l <= akhir:
        tmp[a:] = A[l:akhir + 1]
    a = 0
    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:
        return arr
    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 = quickSortNew(kurang)
        lebih = quickSortNew(lebih)
        return kurang + pivotList + lebih

d = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]

mergeSort(d)
print(d)
quickSort(d)
print(d)
mergeSortNew(d)
print(d)
quickSortNew(d)
print(d)
```

Ln: 27 Col: 0

```
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: 27 Col: 0

Output

Python 3.6.5 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64
)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==================== RESTART: E:/Prak_Algostruk/Modul6_7.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: 0.118789 detik
quick: 0.0688512 detik
merge New: 0.14271 detik
quick New: 0.0598071 detik
>>>
```

8. Linked List untuk program Merge Sort

Modul6_8.py - E:/Prak_Algostruk/Modul6_8.py (3.6.5)

File Edit Format Run Options Window Help

```
#Nomer 8

class Node():
    def __init__(self, data, tautan=None):
        self.data = data
        self.tautan = tautan

def cetak(head):
    curr = head
    while curr is not None:
        try:
            print (curr.data)
            curr = curr.tautan
        except:
            pass

a = Node(13)
b = Node(4)
c = Node(6)
d = Node(7)
e = Node(24)
f = Node(15)
g = Node(29)
h = Node(23)

a.tautan = b
b.tautan = c
c.tautan = d
d.tautan = e
e.tautan = f
f.tautan = g
g.tautan = h

def mergeSortLL(A):
    linked = A
    try:
        daftar = []
```

Modul6_8.py - E:/Prak_Algostruk/Modul6_8.py (3.6.5)

File Edit Format Run Options Window Help

```
        curr = A
        while curr:
            daftar.append(curr.data)
            curr = curr.tautan
        A = daftar
    except:
        A = A

    if len(A) > 1:
        mid = len(A) // 2
        separuhkiri = A[:mid]
        separuhkanan = A[mid:]
        mergeSortLL(separuhkiri)
        mergeSortLL(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
            else:
                A[k] = separuhkanan[j]
                j = j + 1
            k=k+1
        while i < len(separuhkiri):
            A[k] = separuhkiri[i]
            i = i + 1
            k=k+1
        while j < len(separuhkanan):
            A[k] = separuhkanan[j]
            j = j + 1
            k=k+1
    for x in A:
        try:
            linked.data = x
            linked = linked.tautan
        except:
            pass

mergeSortLL(a)
cetak(a)
```

Ln: 45 Col: 0

Output

Python 3.6.5 Shell

File Edit Shell Debug Options Window Help

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD6
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=================== RESTART: E:/Prak_Algostruk/Modul6_8.py ===================
4
6
7
13
15
23
24
29
>>>
```