Nama : Nur Fitria Melani
NIM : L200180012
Kelas : A

MODUL 6.
PENGURUTAN LANJUTAN

LATIHAN.

lt1.py - D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6\lt1.py (3

File Edit Format Run Options Window Help

```python
def gabungkanDuaListUrut(A,B):
    la = len(A);lb = len(B)
    C = list()  #C adalah list baru
    i = 0; j = 0

    #gabungkan keduanya sampai salah satu kosong
    while i < la and j < lb:
        if A[i] < B[i]:
            C.append(A[i])
            i += 1
        else:
            C.append(B[j])
            j += 1

    while i < la:        #jika A mempunyai sisa
        C.append(A[i])   #tumpukkan ke list baru itu
        i += 1           #satu demi satu

    while i < lb:        #jika B mempunyai sisa
        C.append(B[i])   #tumpukkan ke list baru itu
        j += 1           #satu demi satu

    return C
```

Python 3.7.2 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
 RESTART: D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6\lt1
.py
>>> P = [2,8,15,23,37]
>>> Q = [4,6,15,20]
>>> R = gabungkanDuaListUrut(P,Q)
>>> print(R)
[2, 4, 6, 15, 20, 8, 15, 23, 37]
>>>
```

Ln: 10 Col: 4

Ln: 1 Col: 0

lt2.py - D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DA...

File Edit Format Run Options Window Help

```python
from lt1 import gabungkanDuaListUrut
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
```

Python 3.7.2 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
 RESTART: D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6\lt2
.py
>>> alist = [54,26,93,17,77,31,44,55,20]
>>> mergeSort(alist)
>>> print(alist)
[17, 20, 26, 31, 44, 54, 55, 77, 93]
>>>
```

lt3.py - D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6\lt3.py (3.7.2)

File Edit Format Run Options Window Help

```python
def quickSort(A):
    quickSortBantu(A,0,len(A)-1)
def quickSortBantu(A, awal, akhir):
    if awal < akhir:
        titikBelah=partisi(A,awal,akhir)
        quickSortBantu(A,awal,titikBelah-1)
        quickSortBantu(A,titikBelah+1,akhir)
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 A[penandaKanan] >= nilaiPivot and penandaKanan >= penandaKiri:
            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
```

Python 3.7.2 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
 RESTART: D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6\lt3
.py
>>> alist = [54,26,93,17,77,31,44,55,20]
>>> quickSort(alist)
>>> print(alist)
[17, 20, 26, 31, 44, 54, 55, 77, 93]
>>>
```

Ln: 9 Col:

Ln: 1 Col: 0

TUGAS

Nomor 1.

no1.py - D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6\no1.py (3.7.2)

File Edit Format Run Options Window Help

```python
import mahasiswa as mhs

def merge(a):
    if len(a) > 1:
        mid = len(a) // 2
        kiri = a[:mid]
        kanan = a[mid:]
        merge(kiri)
        merge(kanan)
        i = 0;
        j = 0;
        k = 0
        while (i < len(kiri) and j < len(kanan)):
            if kiri[i].uang < kanan[j].uang:
                a[k] = kiri[i]
                i = i + 1
            else:
                a[k] = kanan[j]
                j = j + 1
            k = k + 1
        while i < len(kiri):
            a[k] = kiri[i]
            i = i + 1
            k = k + 1
        while j < len(kanan):
            a[k] = kanan[j]
            j = j + 1
            k = k + 1

def partition( arr, low, high):
    i = (low - 1)
    pivot = arr[high].uang
    for j in range(low, high):
        if arr[j].uang <= pivot:
            i = i + 1
            arr[i].uang, arr[j].uang = arr[j].uang, arr[i].uang
    arr[i + 1].uang, arr[high].uang = arr[high].uang, arr[i + 1].uang
    return (i + 1)

def quickSort(arr, low, high):
    if low < high:
        pi = partition(arr, low, high)
        quickSort(arr, low, pi - 1)
        quickSort(arr, pi + 1, high)

c1 = mhs.mhsTIF("Rafi",11,"Surakarta",245000)
c2 = mhs.mhsTIF("Fitria",12,"Sragen",230000)
c3 = mhs.mhsTIF("Nata",23,"Magelang",250000)
c4 = mhs.mhsTIF("Dodo",26,"Semarang",240000)
c5 = mhs.mhsTIF("Hana",27,"Yogyakarta",235000)

x = [c1, c2, c3, c4, c5]

merge(x)
for i in x:
    print(i.uang)

print("-----------------------------------")
quickSort(x,0,len(x)-1)
for i in x:
    print(i.uang)
```

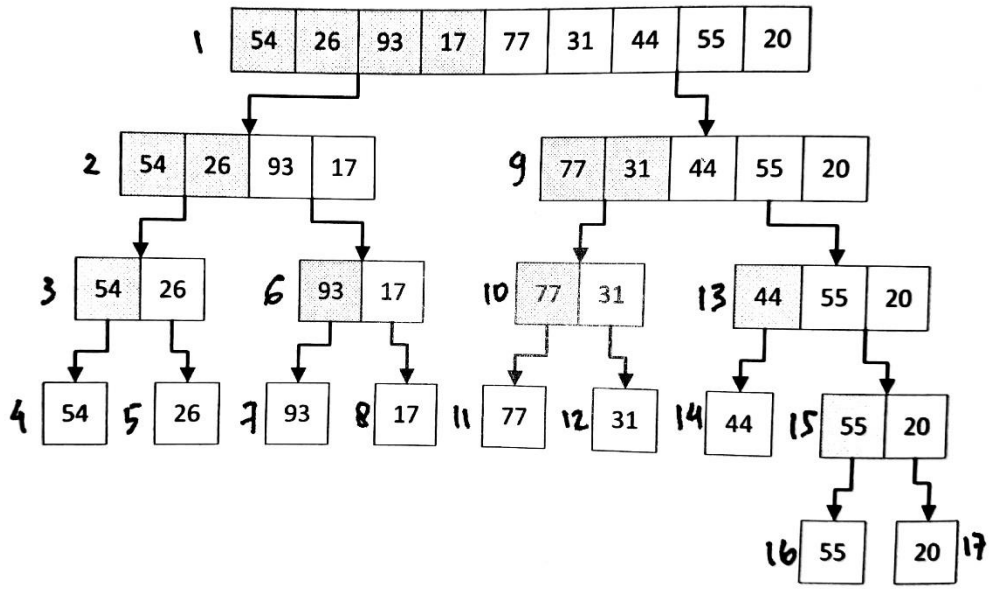
Ln: 40 Col: 30

Python 3.7.2 Shell

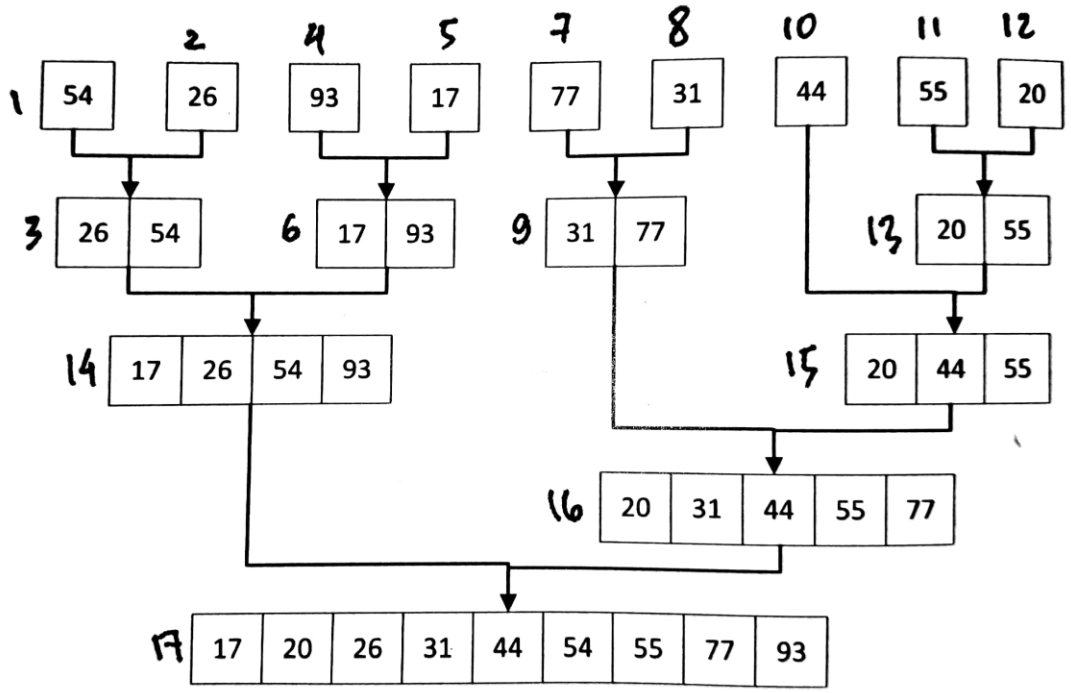
File Edit Shell Debug Options Window Help

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32
bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
 RESTART: D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6
\no1.py
230000
235000
240000
245000
250000
------------------------------------
230000
235000
240000
245000
250000
>>>
```

Nomor 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.

Nomor 3.

no3.py - D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6\no3.py (3.7.2)

File Edit Format Run Options Window Help

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

k = [i for i in range(1, 6001)]
kocok(k)

def bubb(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n - i - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]

def sele(A):
    for i in range(len(A)):
        min_idx = i
        for j in range(i + 1, len(A)):
            if A[min_idx] > A[j]:
                min_idx = j
        A[i], A[min_idx] = A[min_idx], A[i]

def inse(arr):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i - 1
        while j >= 0 and key < arr[j]:
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key

def mergeSort(arr):
    if len(arr) > 1:
        mid = len(arr) // 2
        L = arr[:mid]
        R = arr[mid:]
        mergeSort(L)
        mergeSort(R)
        i = j = k = 0
        while i < len(L) and j < len(R):
            if L[i] < R[j]:
                arr[k] = L[i]
                i += 1
            else:
                arr[k] = R[j]
                j += 1
            k += 1
        while i < len(L):
            arr[k] = L[i]
            i += 1
            k += 1
        while j < len(R):
            arr[k] = R[j]
            j += 1
            k += 1

def partition(arr, low, high):
    i = (low - 1)
    pivot = arr[high]
    for j in range(low, high):
        if arr[j] <= pivot:
            i = i + 1
            arr[i], arr[j] = arr[j], arr[i]
    arr[i + 1], arr[high] = arr[high], arr[i + 1]
    return (i + 1)

def quickSort(arr, low, high):
    if low < high:
        pi = partition(arr, low, high)
        quickSort(arr, low, pi - 1)
        quickSort(arr, pi + 1, high)

bub = k[:]
sel = k[:]
ins = k[:]
mer = k[:]
qui = k[:]

aw = detak();
bubb(bub);
ak = detak();
print('bubble : %g detik' % (ak - aw));
aw = detak();
sele(sel);
ak = detak();
print('selection : %g detik' % (ak - aw));
aw = detak();
inse(ins);
ak = detak();
print('insertion : %g detik' % (ak - aw));
aw = detak();
mergeSort(mer);
ak = detak();
print('merge : %g detik' % (ak - aw));
aw = detak();
quickSort(qui, 0, len(qui) - 1);
ak = detak();
print('quick : %g detik' % (ak - aw));
```

Ln: 71 Col: 36

Python 3.7.2 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
 RESTART: D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6\no3
.py
bubble : 8.49807 detik
selection : 3.42502 detik
insertion : 2.70218 detik
merge : 0.061116 detik
quick : 0.0601828 detik
>>>
```

Ln: 10 Col: 4

Nomor 4.

L : [80, 7, 24, 16, 43, 91, 35, 2, 19, 72]

→ Merge Sort

80	7	24	16	43	91	35	2	19	72

Proses 1

7	80	16	24	43	91	2	35	19	72

Proses 2

7	16	24	80	2	35	43	91	19	72

Proses 3

2	7	16	24	35	43	80	91	19	72

Proses 4

2	7	16	19	24	35	43	72	80	91

Quick Sort

80	7	24	16	43	91	35	2	19	72

pivot									
80	7	24	16	43	91	35	2	19	72
Low									High

									pivot
72	7	24	16	43	91	35	2	19	80
Low									High

									pivot
72	7	24	16	43	91	35	2	19	80
					Low				High

					pivot				
72	7	24	16	43	80	35	2	19	91
					Low				High

								pivot	
72	7	24	16	43	19	35	2	80	91
					Low			High	

Nomor 5.

no5.py - D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6\no5.py (3.7.2)

File Edit Format Run Options Window Help

```python
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([24, 8, 57, 15]))
```

Ln: 41 Col: 29

Python 3.7.2 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
 RESTART: D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6\no5
.py
[8, 15, 24, 57]
>>>
```

Ln: 6 Col: 4

Nomor 6.

no6.py - D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6\no6.py (3.7.2)

File Edit Format Run Options Window Help

```python
def quickSort(L, ascending=True):
    quicksorthelp(L, 0, len(L), ascending)
def quicksorthelp(L, low, high, ascending=True):
    result = 0
    if low < high:
        pivot_location, result = Partition(L, low, high, ascending)
        result += quicksorthelp(L, low, pivot_location, ascending)
        result += quicksorthelp(L, pivot_location + 1, high, ascending)
    return result

def Partition(L, low, high, ascending=True):
    result = 0
    pivot, pidx = median_of_three(L, low, high)
    L[low], L[pidx] = L[pidx], L[low]
    i = low + 1
    for j in range(low + 1, high, 1):
        result += 1
        if (ascending and L[j] < pivot) or (not ascending and L[j] > pivot):
            L[i], L[j] = L[j], L[i]
            i += 1
    L[low], L[i - 1] = L[i - 1], L[low]
    return i - 1, result

def median_of_three(L, low, high):
    mid = (low + high - 1) // 2
    a = L[low]
    b = L[mid]
    c = L[high - 1]
    if a <= b <= c:
        return b, mid
    if c <= b <= a:
        return b, mid
    if a <= c <= b:
        return c, high - 1
    if b <= c <= a:
        return c, high - 1
    return a, low

listt = list([64, 29, 4, 112, 154])
quickSort(listt, False)  #descending order
print('sorted:')
print(listt)
```

Ln: 39 Col: 35

Python 3.7.2 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
 RESTART: D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6\no6
.py
sorted:
[154, 112, 64, 29, 4]
>>>
```

Ln: 7 Col:

Nomor 7.

no7.py - D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6\no7.py (3.7.2)

File Edit Format Run Options Window Help

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

k = [i for i in range(1, 6001)]
kocok(k)

def mergeSort(arr):
    if len(arr) > 1:
        mid = len(arr) // 2
        L = arr[:mid]
        R = arr[mid:]
        mergeSort(L)
        mergeSort(R)
        i = j = k = 0
        while i < len(L) and j < len(R):
            if L[i] < R[j]:
                arr[k] = L[i]
                i += 1
            else:
                arr[k] = R[j]
                j += 1
            k += 1
        while i < len(L):
            arr[k] = L[i]
            i += 1
            k += 1
        while j < len(R):
            arr[k] = R[j]
            j += 1
            k += 1
```

```python
def partition(arr, low, high):
    i = (low - 1)
    pivot = arr[high]
    for j in range(low, high):
        if arr[j] <= pivot:
            i = i + 1
            arr[i], arr[j] = arr[j], arr[i]
    arr[i + 1], arr[high] = arr[high], arr[i + 1]
    return (i + 1)

def quickSort(arr, low, high):
    if low < high:
        pi = partition(arr, low, high)
        quickSort(arr, low, pi - 1)
        quickSort(arr, pi + 1, high)

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])
```

```python
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

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

def quickSortMOD(L, ascending=True):
    quicksorthelp(L, 0, len(L), ascending)
def quicksorthelp(L, low, high, ascending=True):
    result = 0
    if low < high:
        pivot_location, result = Partition(L, low, high, ascending)
        result += quicksorthelp(L, low, pivot_location, ascending)
        result += quicksorthelp(L, pivot_location + 1, high, ascending)
    return result
```

```python
def Partition(L, low, high, ascending=True):
    result = 0
    pivot, pidx = median_of_three(L, low, high)
    L[low], L[pidx] = L[pidx], L[low]
    i = low + 1
    for j in range(low + 1, high, 1):
        result += 1
        if (ascending and L[j] < pivot) or (not ascending and L[j] > pivot):
            L[i], L[j] = L[j], L[i]
            i += 1
    L[low], L[i - 1] = L[i - 1], L[low]
    return i - 1, result

def median_of_three(L, low, high):
    mid = (low + high - 1) // 2
    a = L[low]
    b = L[mid]
    c = L[high - 1]
    if a <= b <= c:
        return b, mid
    if c <= b <= a:
        return b, mid
    if a <= c <= b:
        return c, high - 1
    if b <= c <= a:
        return c, high - 1
    return a, low
```

```python
mer = k[:]
qui = k[:]
mer2 = k[:]
qui2 = k[:]

aw = detak();
mergeSort(mer);
ak = detak();
print('merge : %g detik' % (ak - aw));
aw = detak();
quickSort(qui, 0, len(qui) - 1);
ak = detak();
print('quick : %g detik' % (ak - aw));
aw = detak();
merge_sort(mer2);
print('merge mod : %g detik' % (ak - aw));
aw = detak();
quickSortMOD(qui2, False);
print('quick mod : %g detik' % (ak - aw));
```

Ln: 1 Col: 30

Python 3.7.2 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
 RESTART: D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6\no7
.py
merge : 0.12745 detik
quick : 0.0662491 detik
merge mod : -0.0684826 detik
quick mod : -0.191195 detik
>>>
```

Ln: 9 Col: 4

Nomor 8.

no8.py - D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6\no8.py (3.7.2)

File Edit Format Run Options Window Help

```python
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

class LinkedList:
    def __init__(self):
        self.head = None

    def appendList(self, data):
        node = Node(data)
        if self.head == None:
            self.head = node
        else:
            curr = self.head
            while curr.next != None:
                curr = curr.next
        curr.next = node

    def appendSorted(self, data):
        node = Node(data)
        curr = self.head
        prev = None

        while curr is not None and curr.data < data:
            prev = curr
            curr = curr.next

        if prev == None:
            self.head = node
        else:
            prev.next = node
        node.next = curr

    def printList(self):
        curr = self.head
        while curr != None:
            print("%d" % curr.data),
            curr = curr.next

    def mergeSorted(self, list1, list2):
        if list1 is None:
            return list2
        if list2 is None:
            return list1

        if list1.data < list2.data:
            temp = list1
            temp.next = self.mergeSorted(list1.next, list2)
        else:
            temp = list2
            temp.next = self.mergeSorted(list1, list2.next)
        return temp

list1 = LinkedList()
list1.appendSorted(12)
list1.appendSorted(37)
list1.appendSorted(8)
list1.appendSorted(65)
list1.appendSorted(42)

print("List 1 :"),
list1.printList()

list2 = LinkedList()
list2.appendSorted(4)
list2.appendSorted(72)
list2.appendSorted(57)

print("List 2 :"),
list2.printList()

list3 = LinkedList()
list3.head = list3.mergeSorted(list1.head, list2.head)

print("Merged List :"),
list3.printList()
```

Ln: 77 Col: 17

Python 3.7.2 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
 RESTART: D:\KULIAH\SEMESTER 4\PRAKTIKUM ALGORITMA DAN STRUKTUR DATA\MODUL_6\no8
.py
List 1 :
8
12
37
42
65
List 2 :
4
57
72
Merged List :
4
8
12
37
42
57
65
72
>>>
```

Ln: 24 Col: 4