Nama : Dimas Kharismawan

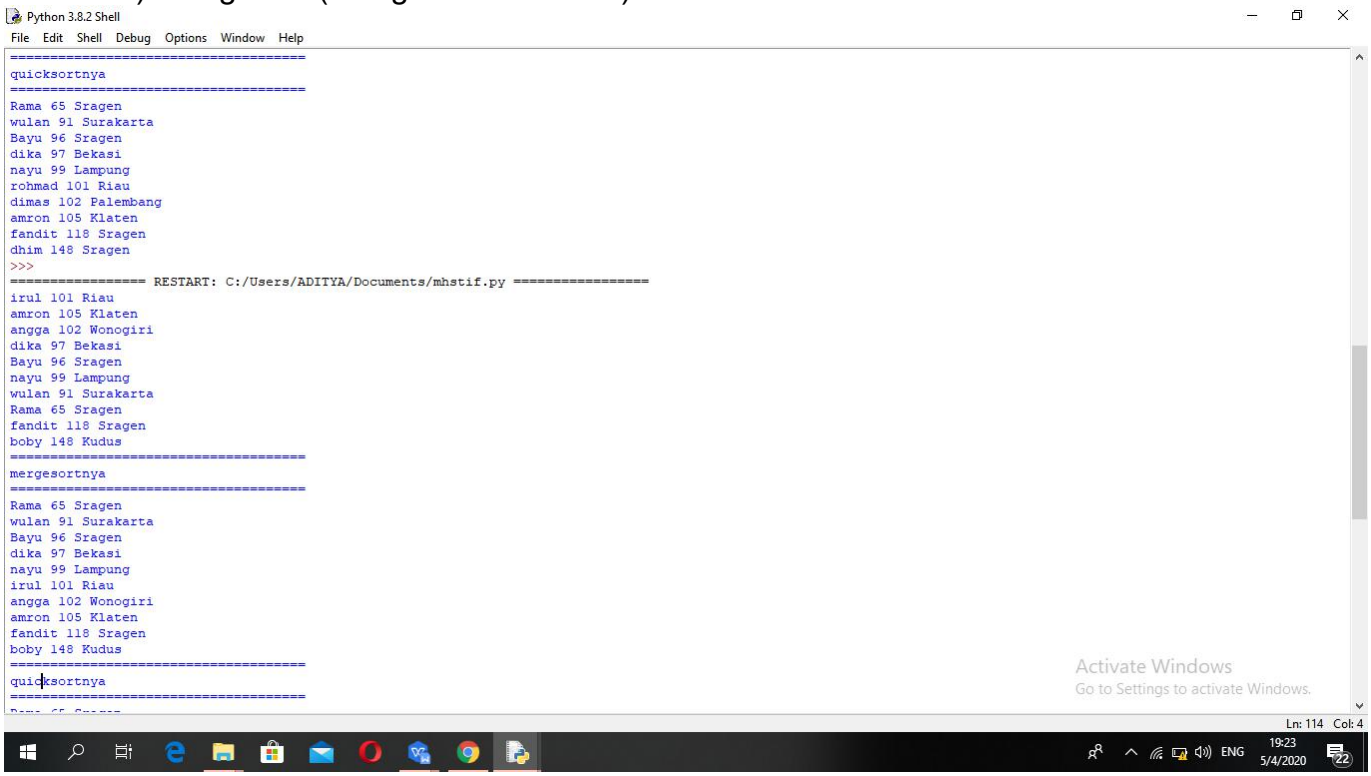
Nim : L200180102

Kelas : D

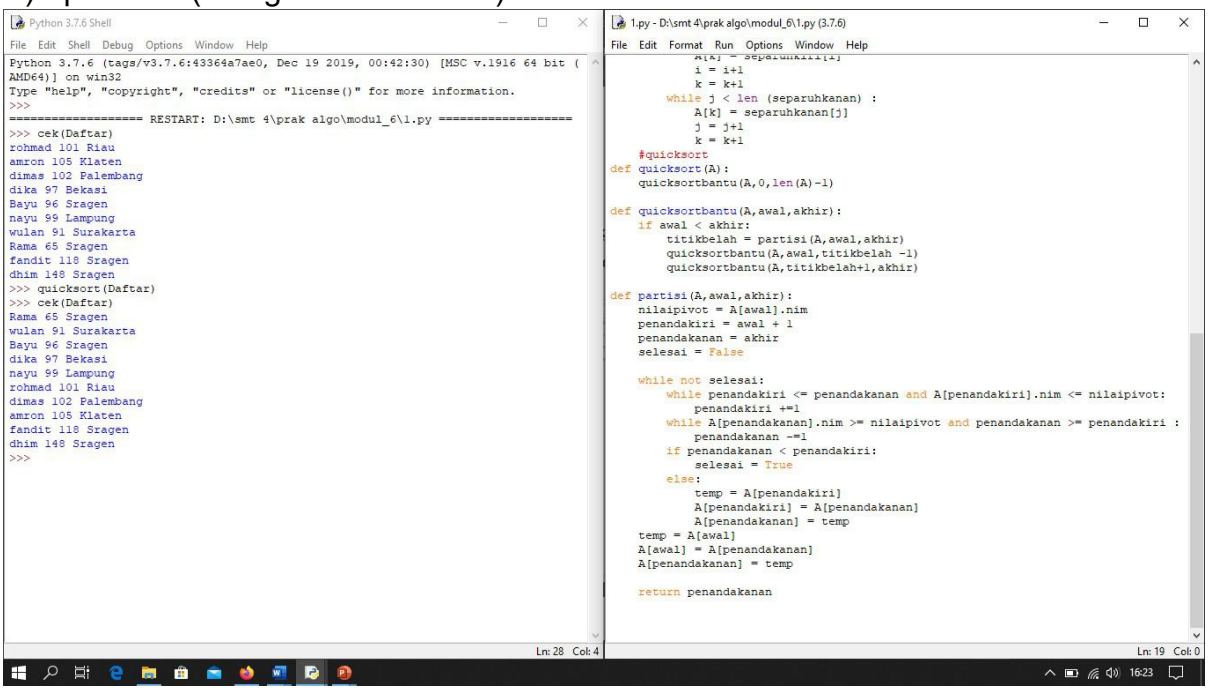
Prak-ASD

Modul 6

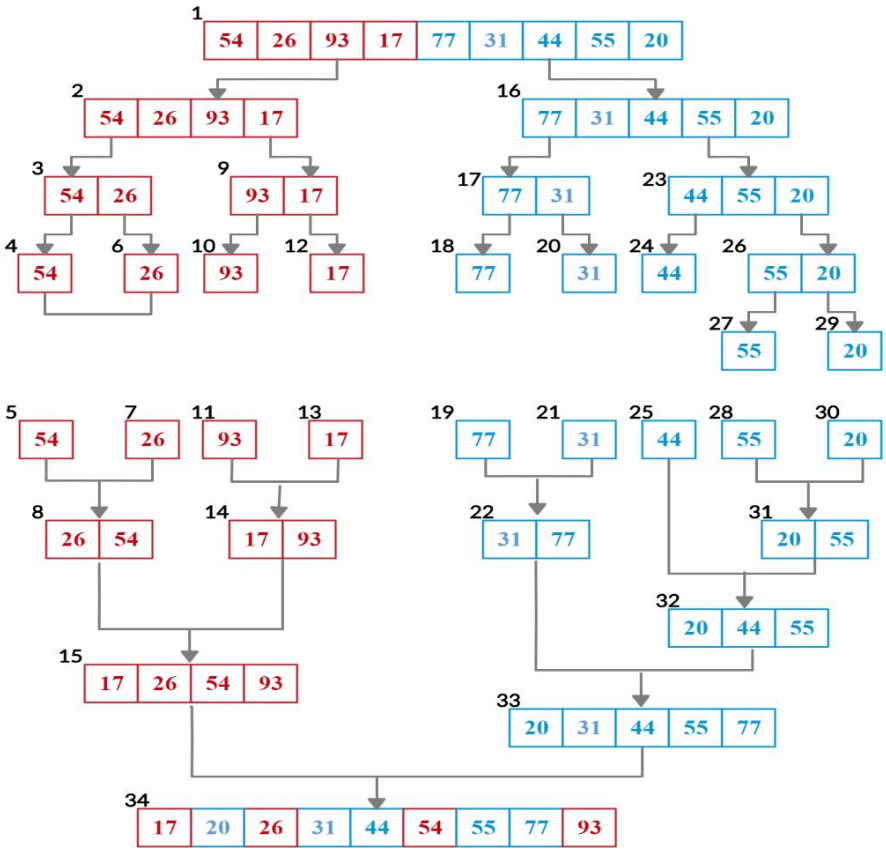
1. A). Mergesort (mengurutkan MhsTif)



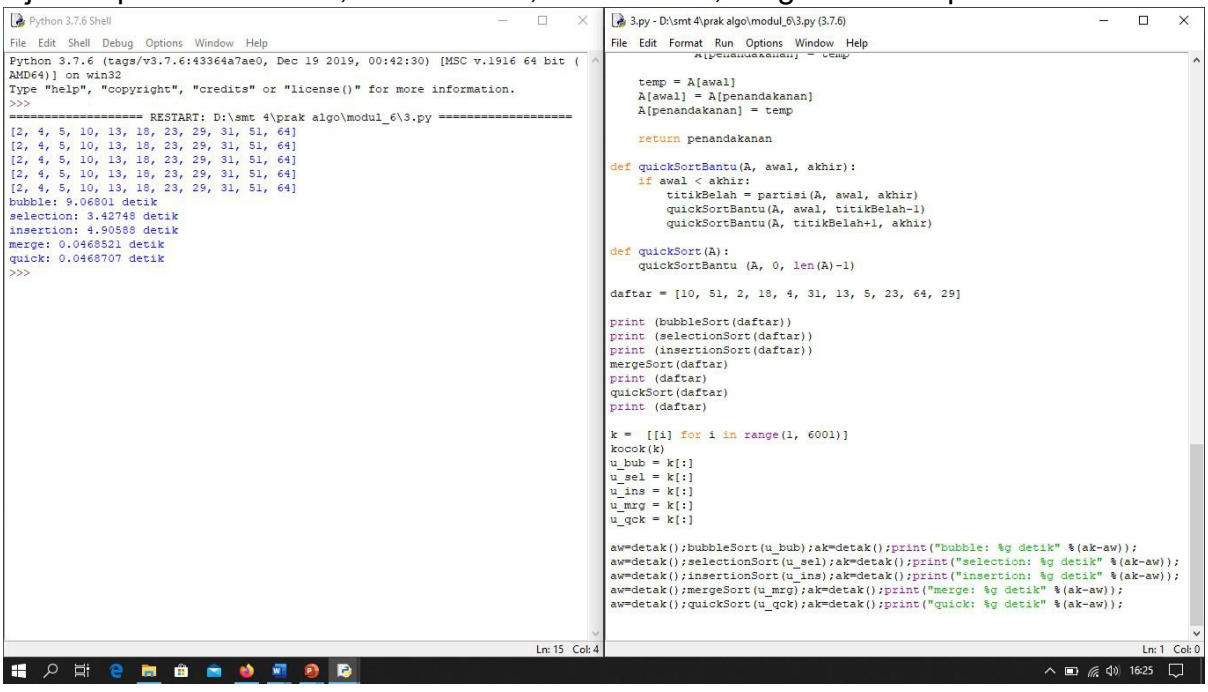
B). quicksort (Mengurutkan MhsTif)



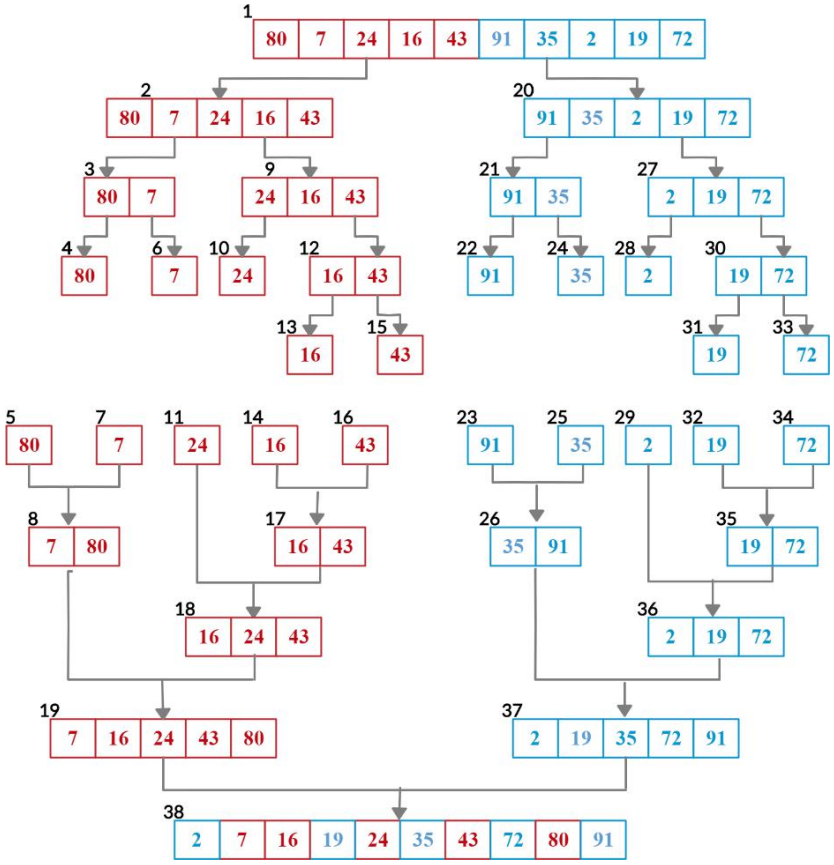
2. Beri nomer urut eksekusi proses gambar 6.1 dan 6.2 mengacu pada output di halaman 59



3. Uji kecepatan bublesort,selectionsort,insertionsort,mergesort dan quicksort



4. A. diberikan List = [80,7,24,16,43,91,35,2,19,72] ,gambarlah trace pengurutan algoritmanya (Merge sort)



4 B. diberikan List = [80,7,24,16,43,91,35,2,19,72] ,gambarlah trace pengurutan algoritmanya (quickSort)

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	

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

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

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

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

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

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

pivot

2	7	19	16	43	24	35	72	80	91

Low High

pivot

2	7	19	16	43	24	35	72	80	91

Low High

pivot

2	7	19	16	24	43	35	72	80	91

Low High

pivot

2	7	19	16	24	43	35	72	80	91

Low High

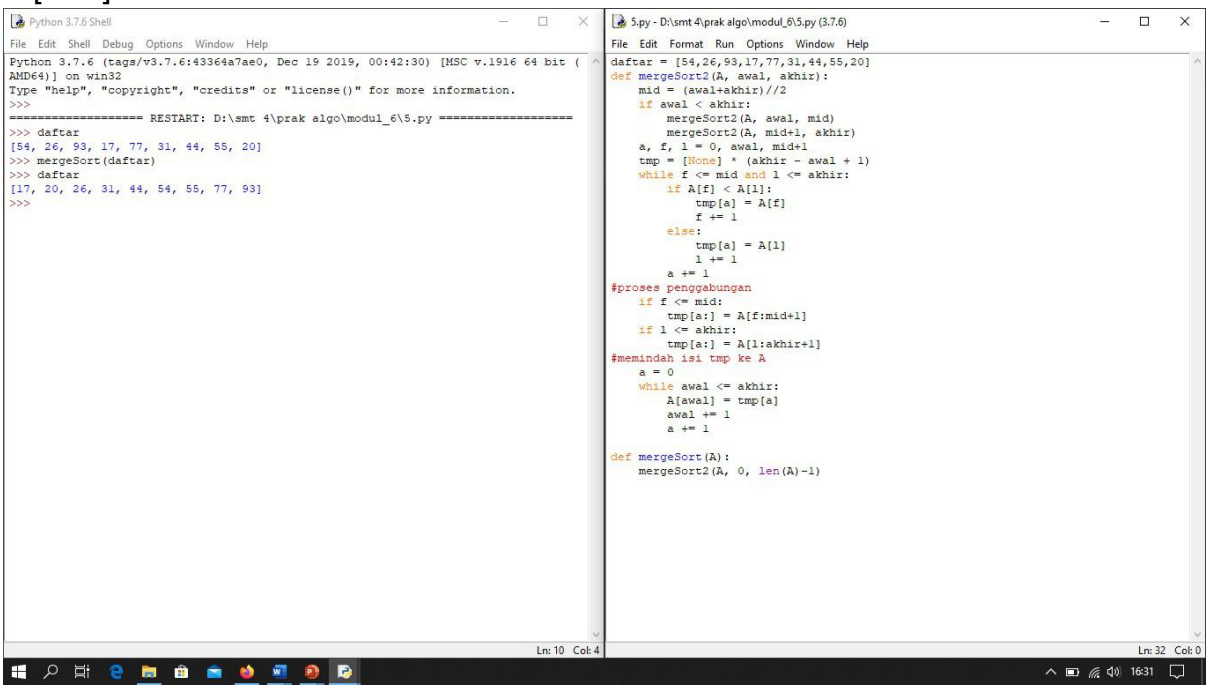
pivot

2	7	16	19	24	35	43	72	80	91

Low High

2	7	16	19	24	35	43	72	80	91

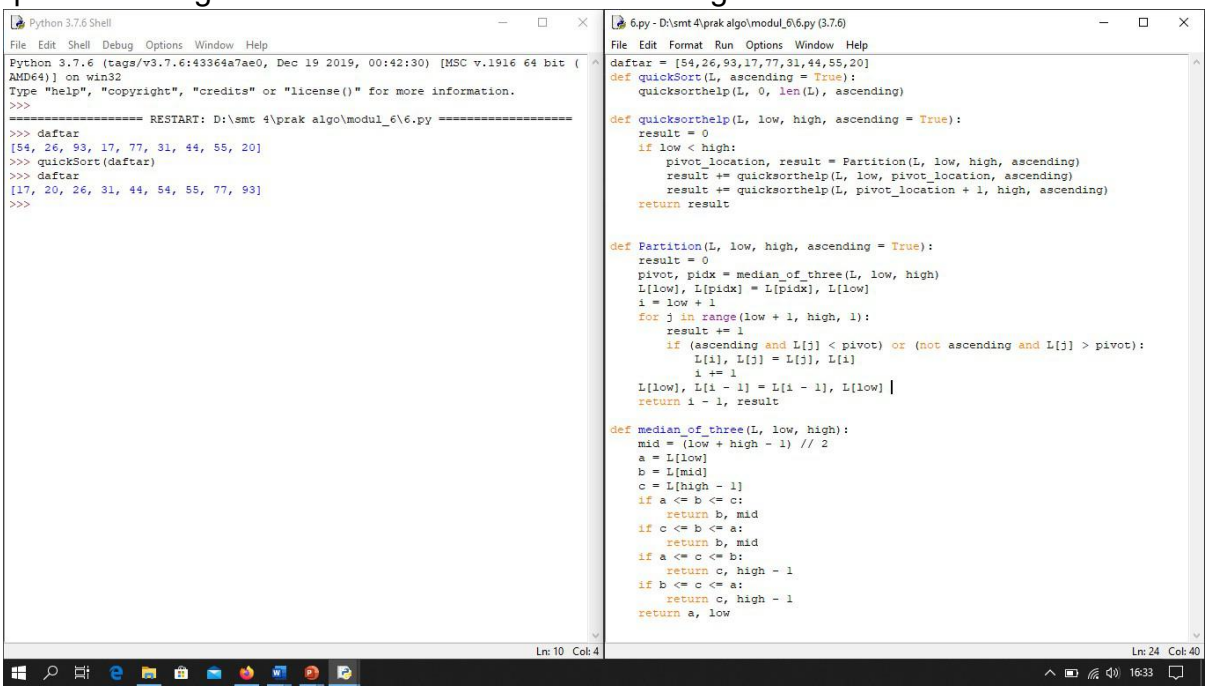
5. Tingkatkan efisien mergesort dengan tidak memakai operator A[:mid] dan A [mid:]



```
Python 3.7.6 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\smt 4\prak algo\modul_6\5.py ==========
>>> daftar
[54, 26, 93, 17, 77, 31, 44, 55, 20]
>>> mergeSort(daftar)
>>> daftar
[17, 20, 26, 31, 44, 54, 55, 77, 93]
>>>
```

```
5.py - D:\smt 4\prak algo\modul_6\5.py (3.7.6)
File  Edit  Format  Run  Options  Window  Help
daftar = [54,26,93,17,77,31,44,55,20]
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
#proses penggabungan
    if f <= mid:
        tmp[a:] = A[f:mid+1]
    if l <= akhir:
        tmp[a:] = A[l:akhir+1]
#memindah isi tmp ke A
    a = 0
    while awal <= akhir:
        A[awal] = tmp[a]
        awal += 1
        a += 1

def mergeSort(A):
    mergeSort2(A, 0, len(A)-1)
```

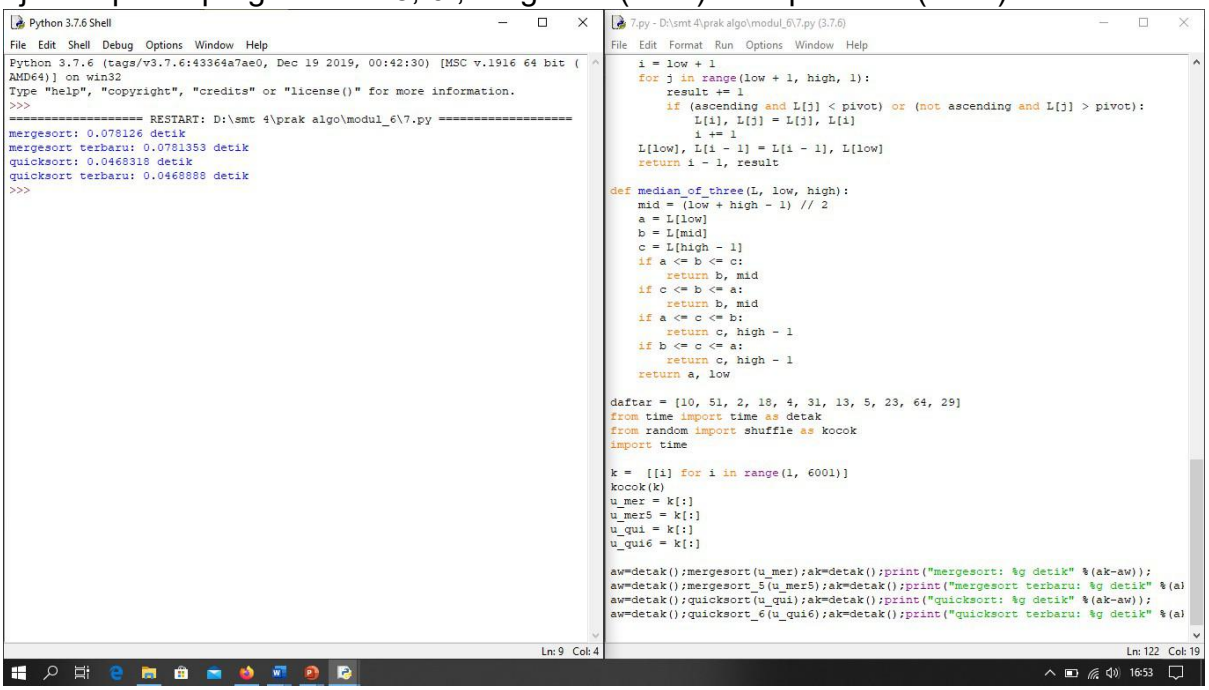
6. quicksort dengan memakai metode median-dari-tiga



```
Python 3.7.6 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\smt 4\prak algo\modul_6\6.py ==========
>>> daftar
[54, 26, 93, 17, 77, 31, 44, 55, 20]
>>> quickSort(daftar)
>>> daftar
[17, 20, 26, 31, 44, 54, 55, 77, 93]
>>>
```

```
6.py - D:\smt 4\prak algo\modul_6\6.py (3.7.6)
File  Edit  Format  Run  Options  Window  Help
daftar = [54,26,93,17,77,31,44,55,20]
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
```

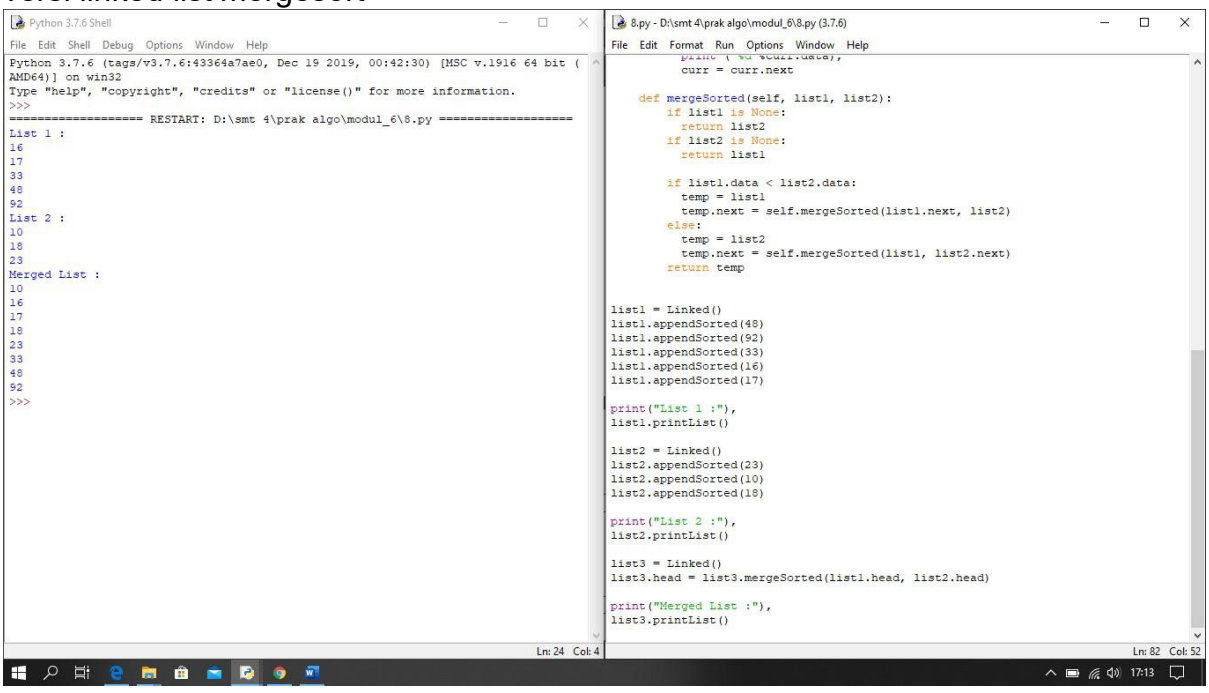
7. uji kecepatan program nmr 5, 6 ,mergesort (awal) dan quicksort (akhir)



```
Python 3.7.6 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\smt 4\prak algo\modul_6\7.py ==========
mergesort: 0.078126 detik
mergesort terbaru: 0.0781353 detik
quicksort: 0.0468318 detik
quicksort terbaru: 0.0468888 detik
>>>
```

```
7.py - D:\smt 4\prak algo\modul_6\7.py (3.7.6)
File  Edit  Format  Run  Options  Window  Help
    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

daftar = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]
from time import time as detak
from random import shuffle as kocok
import time

k =  [[i] for i in range(1, 6001)]
kocok(k)
u_mer = k[:]
u_mer5 = k[:]
u_qui = k[:]
u_qui6 = k[:]

aw=detak();mergesort(u_mer);ak=detak();print("mergesort: %g detik" %(ak-aw));
aw=detak();mergesort_5(u_mer5);ak=detak();print("mergesort terbaru: %g detik" %(a
aw=detak();quicksort(u_qui);ak=detak();print("quicksort: %g detik" %(ak-aw));
aw=detak();quicksort_6(u_qui6);ak=detak();print("quicksort terbaru: %g detik" %(a
```

8. versi linked list mergesort



```
Python 3.7.6 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\smt 4\prak algo\modul_6\8.py ==========
List 1 :
16
17
33
48
92
List 2 :
10
18
23
Merged List :
10
16
17
18
23
33
48
92
>>>
```

```
8.py - D:\smt 4\prak algo\modul_6\8.py (3.7.6)
File  Edit  Format  Run  Options  Window  Help
            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 = Linked()
list1.appendSorted(48)
list1.appendSorted(92)
list1.appendSorted(33)
list1.appendSorted(16)
list1.appendSorted(17)

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

list2 = Linked()
list2.appendSorted(23)
list2.appendSorted(10)
list2.appendSorted(18)

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

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

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