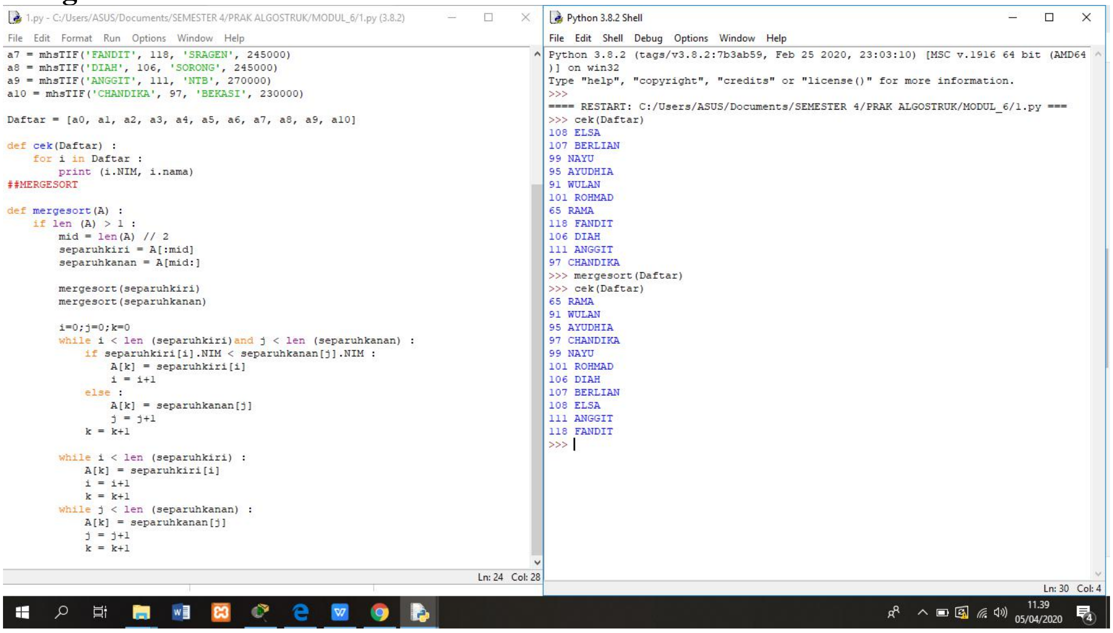
NAMA : ELSA PUTRI ALIYYA
NIM : L200180108
KELAS : D

MODUL 6

1.

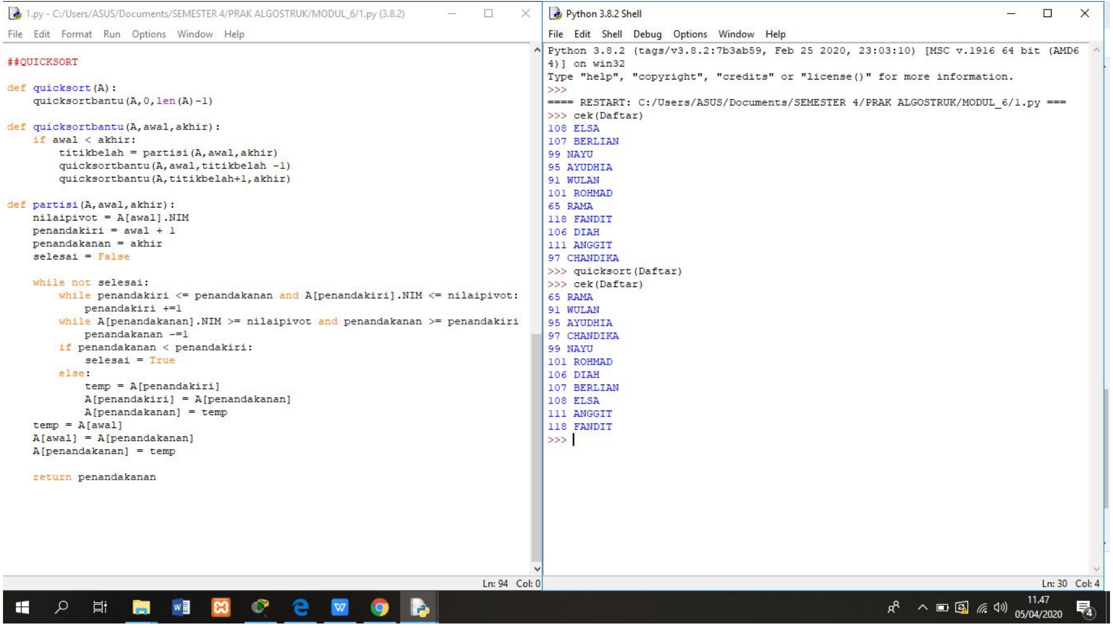
Mergesort



```
a7 = mhsTIF('FANDIT', 118, 'SRAGEN', 245000)
a8 = mhsTIF('DIAH', 106, 'SORONG', 245000)
a9 = mhsTIF('ANGGIT', 111, 'NTB', 270000)
a10 = mhsTIF('CHANDIKA', 97, 'BEKASI', 230000)

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

def cek(Daftar) :
    for i in Daftar :
        print (i.NIM, i.nama)
##MERGESORT

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].NIM < separuhkanan[j].NIM :
                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 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64
)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/1.py ===
>>> cek(Daftar)
108 ELSA
107 BERLIAN
99 NAYU
95 AYUDHIA
91 WULAN
101 ROHMAD
65 RAMA
118 FANDIT
106 DIAH
111 ANGGIT
97 CHANDIKA
>>> mergesort(Daftar)
>>> cek(Daftar)
65 RAMA
91 WULAN
95 AYUDHIA
97 CHANDIKA
99 NAYU
101 ROHMAD
106 DIAH
107 BERLIAN
108 ELSA
111 ANGGIT
118 FANDIT
>>> |
```

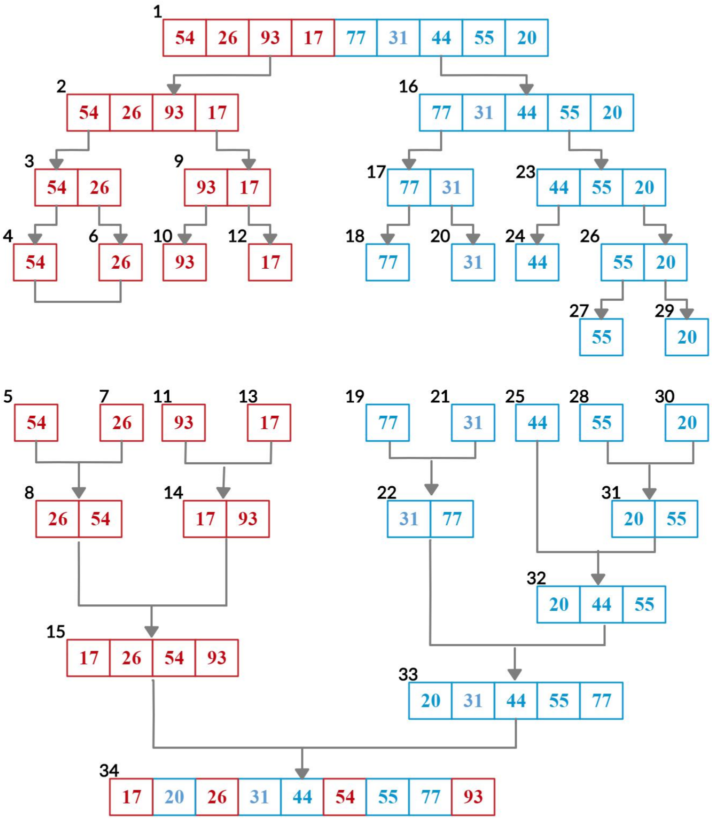
Quicksort



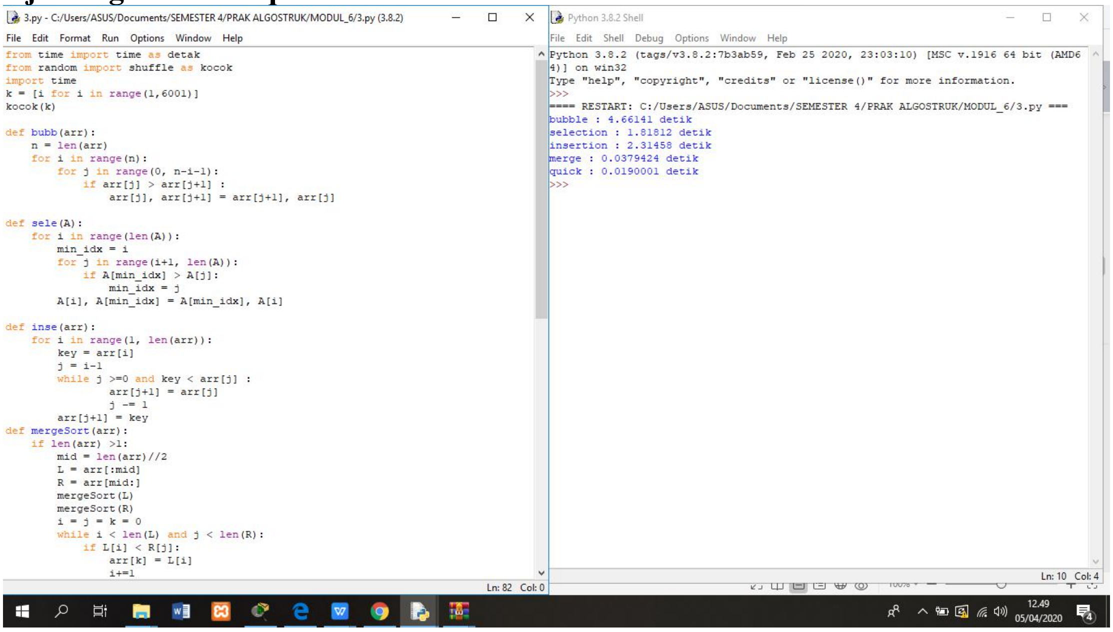
```
##QUICKSORT

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].NIM
    penandakiri = awal + 1
    penandakanan = akhir
    selesai = False

    while not selesai:
        while penandakiri <= penandakanan and A[penandakiri].NIM <= nilaipivot:
            penandakiri +=1
        while A[penandakanan].NIM >= nilaipivot and penandakanan >= penandakiri
            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 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD6
4)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/1.py ===
>>> cek(Daftar)
108 ELSA
107 BERLIAN
99 NAYU
95 AYUDHIA
91 WULAN
101 ROHMAD
65 RAMA
118 FANDIT
106 DIAH
111 ANGGIT
97 CHANDIKA
>>> quicksort(Daftar)
>>> cek(Daftar)
65 RAMA
91 WULAN
95 AYUDHIA
97 CHANDIKA
99 NAYU
101 ROHMAD
106 DIAH
107 BERLIAN
108 ELSA
111 ANGGIT
118 FANDIT
>>> |
```

2. Menandai eksekusi proses pada gambar 6.1 dan 6.2

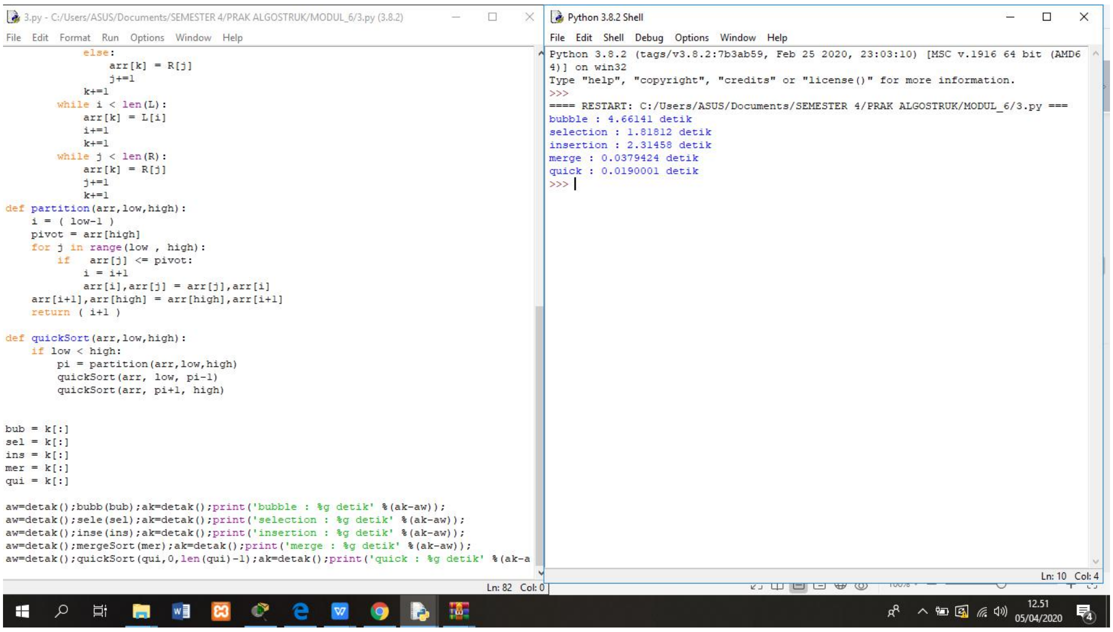


3. Uji mergesort dan quicksort


3.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/3.py (3.8.2)
File Edit Format Run Options Window Help
Ln: 82 Col: 0
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Ln: 10 Col: 4
12.49
05/04/2020

```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
```

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD6
4)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/3.py ===
bubble : 4.66141 detik
selection : 1.81812 detik
insertion : 2.31458 detik
merge : 0.0379424 detik
quick : 0.0190001 detik
>>>
```

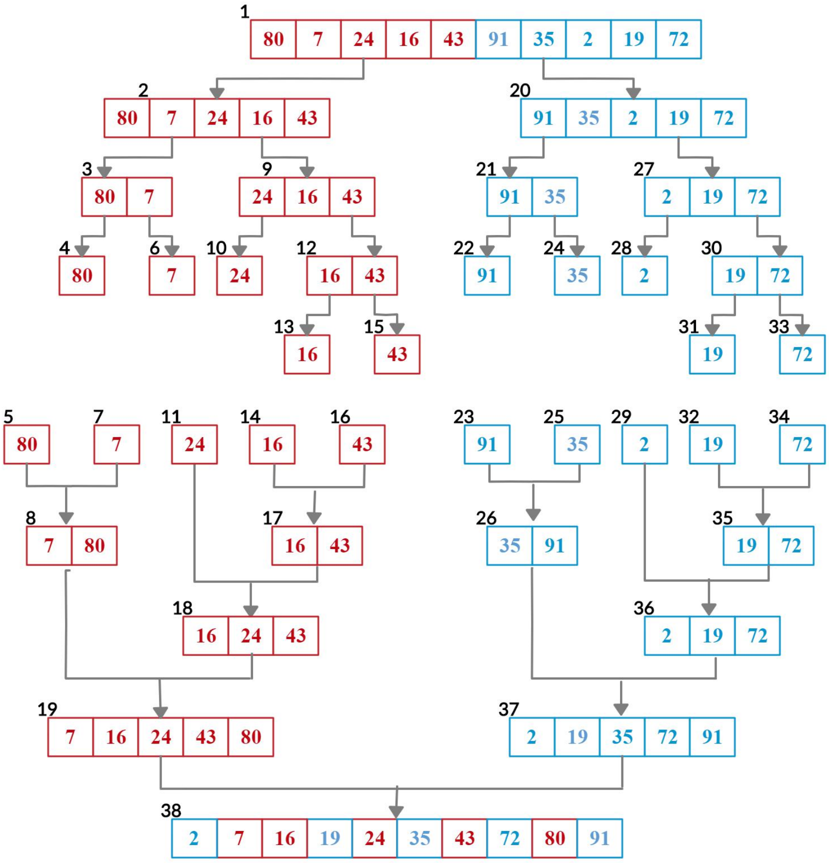

3.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/3.py (3.8.2)
File Edit Format Run Options Window Help
Ln: 82 Col: 0
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Ln: 10 Col: 4
12.51
05/04/2020

```
            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-a
```

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD6
4)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/3.py ===
bubble : 4.66141 detik
selection : 1.81812 detik
insertion : 2.31458 detik
merge : 0.0379424 detik
quick : 0.0190001 detik
>>> 
```

4.

A. mergesort



B. 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

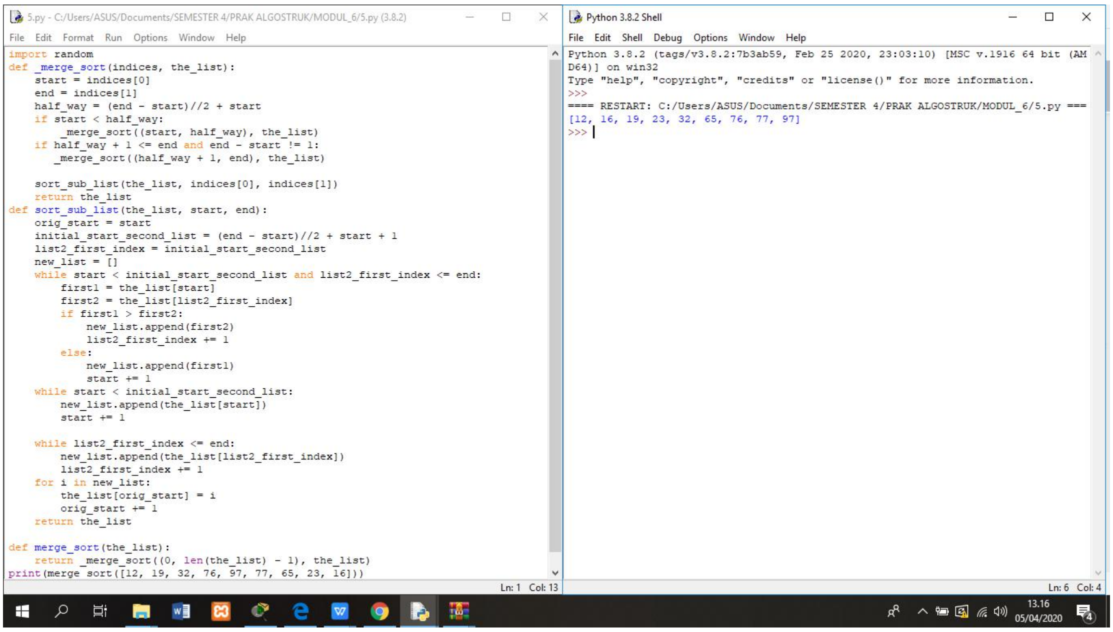
pivot

2	**7**	**16**	**19**	**24**	**35**	**43**	**72**	**80**	**91**

Low High

2	**7**	**16**	**19**	**24**	**35**	**43**	**72**	**80**	**91**

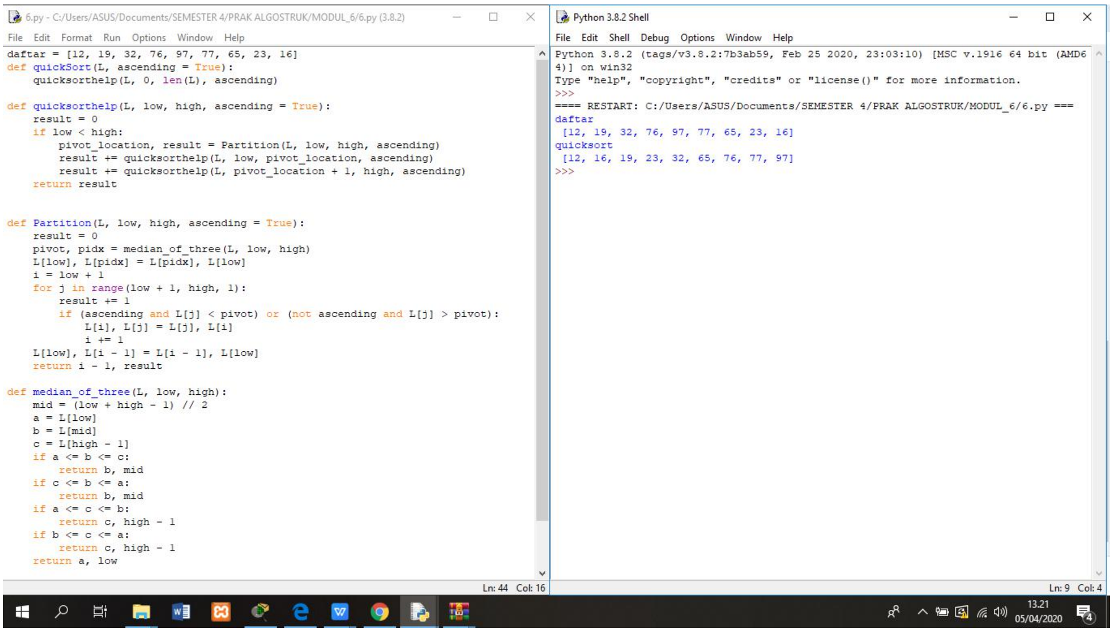
5.


5.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/5.py (3.8.2)
File Edit Format Run Options Window Help
Ln: 1 Col: 13
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Ln: 6 Col: 4
13.16
05/04/2020

```
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([12, 19, 32, 76, 97, 77, 65, 23, 16]))
```

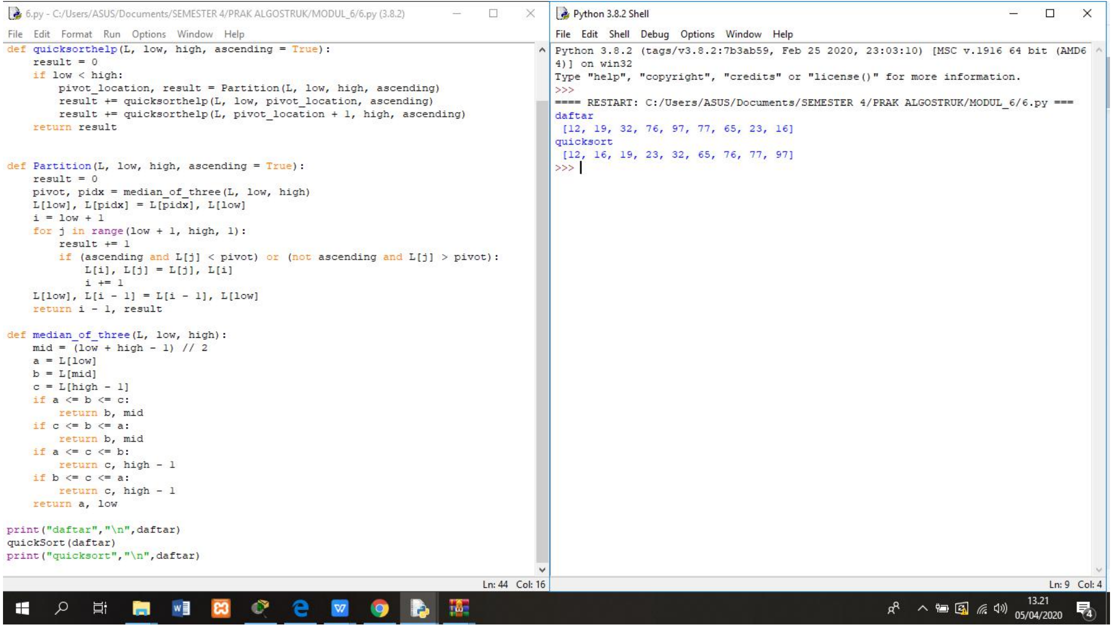
```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/5.py ===
[12, 16, 19, 23, 32, 65, 76, 77, 97]
>>> 
```

6.


6.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/6.py (3.8.2)
File Edit Format Run Options Window Help
Ln: 44 Col: 16
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Ln: 9 Col: 4
13.21
05/04/2020

```
daftar = [12, 19, 32, 76, 97, 77, 65, 23, 16]
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
```

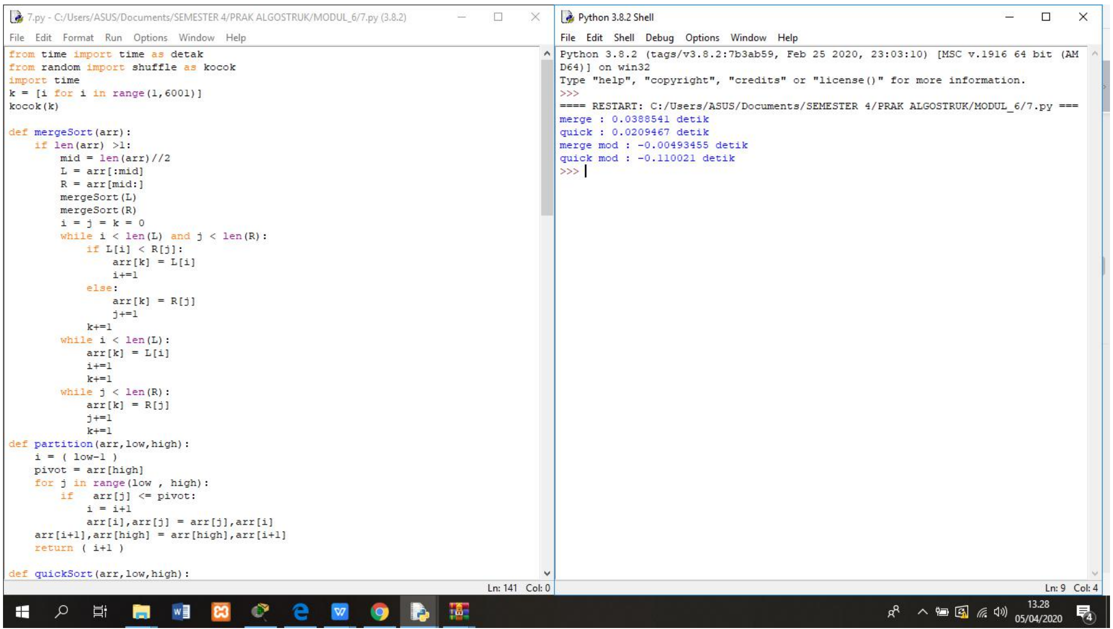
```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD6
4)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/6.py ===
daftar
 [12, 19, 32, 76, 97, 77, 65, 23, 16]
quicksort
 [12, 16, 19, 23, 32, 65, 76, 77, 97]
>>> 
```


6.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/6.py (3.8.2)
File Edit Format Run Options Window Help
Ln: 44 Col: 16
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Ln: 9 Col: 4
13.21
05/04/2020

```
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

print("daftar","\n",daftar)
quickSort(daftar)
print("quicksort","\n",daftar)
```

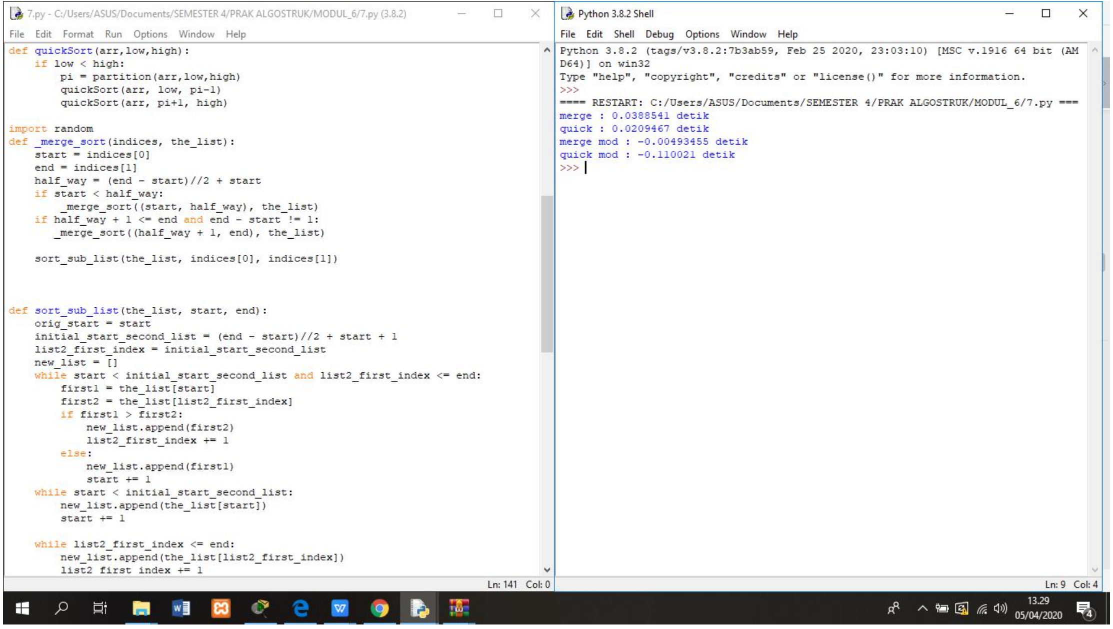
```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD6
4)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/6.py ===
daftar
 [12, 19, 32, 76, 97, 77, 65, 23, 16]
quicksort
 [12, 16, 19, 23, 32, 65, 76, 77, 97]
>>> 
```

7.


7.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/7.py (3.8.2)
File Edit Format Run Options Window Help
Ln: 141 Col: 0
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Ln: 9 Col: 4
13.28
05/04/2020

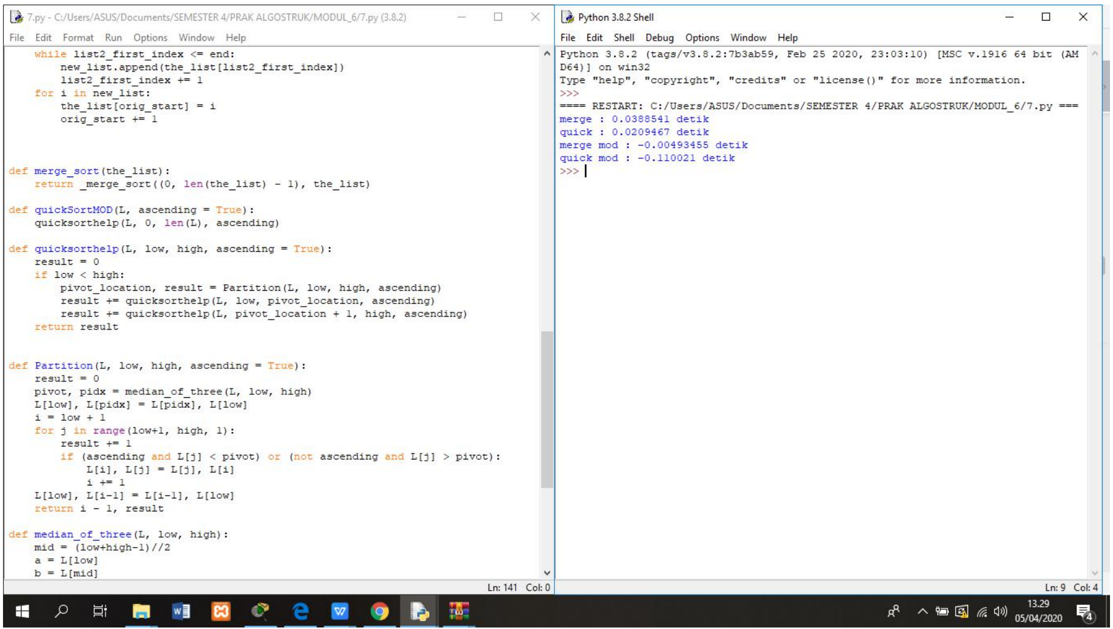
```python
from time import time as detak
from random import shuffle as kocok
import time
k = [i for i in range(1,6001)]
kocok(k)

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
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):
```

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/7.py ===
merge : 0.0388541 detik
quick : 0.0209467 detik
merge mod : -0.00493455 detik
quick mod : -0.110021 detik
>>>
```


7.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/7.py (3.8.2)
File Edit Format Run Options Window Help
Ln: 141 Col: 0
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Ln: 9 Col: 4
13.29
05/04/2020

```python
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])



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
```

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/7.py ===
merge : 0.0388541 detik
quick : 0.0209467 detik
merge mod : -0.00493455 detik
quick mod : -0.110021 detik
>>>
```

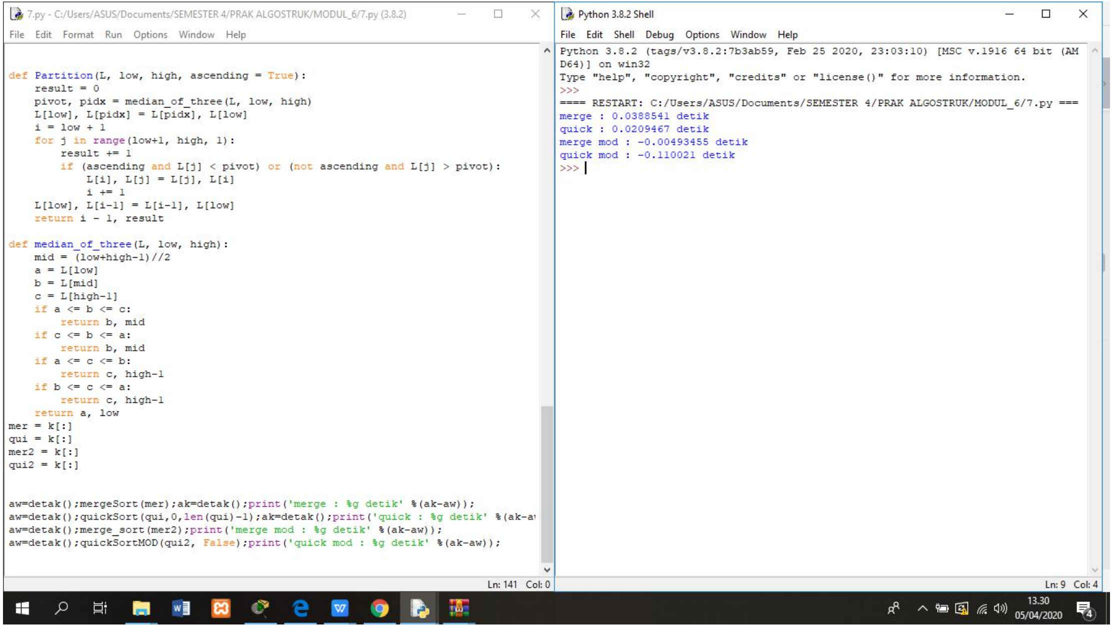


7.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/7.py (3.8.2)
File Edit Format Run Options Window Help

```
    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


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]
```

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/7.py ===
merge : 0.0388541 detik
quick : 0.0209467 detik
merge mod : -0.00493455 detik
quick mod : -0.110021 detik
>>> 
```

Ln: 9 Col: 4
13.29
05/04/2020

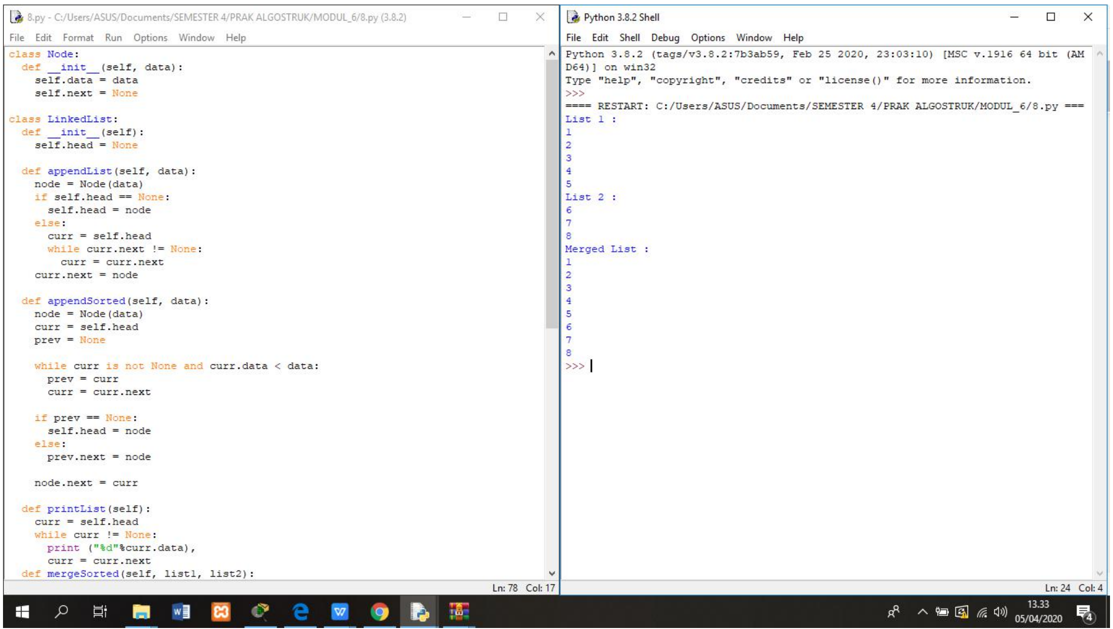


7.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/7.py (3.8.2)
File Edit Format Run Options Window Help

```


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
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-a
aw=detak();merge_sort(mer2);print('merge mod : %g detik' %(ak-aw));
aw=detak();quickSortMOD(qui2, False);print('quick mod : %g detik' %(ak-aw));
```

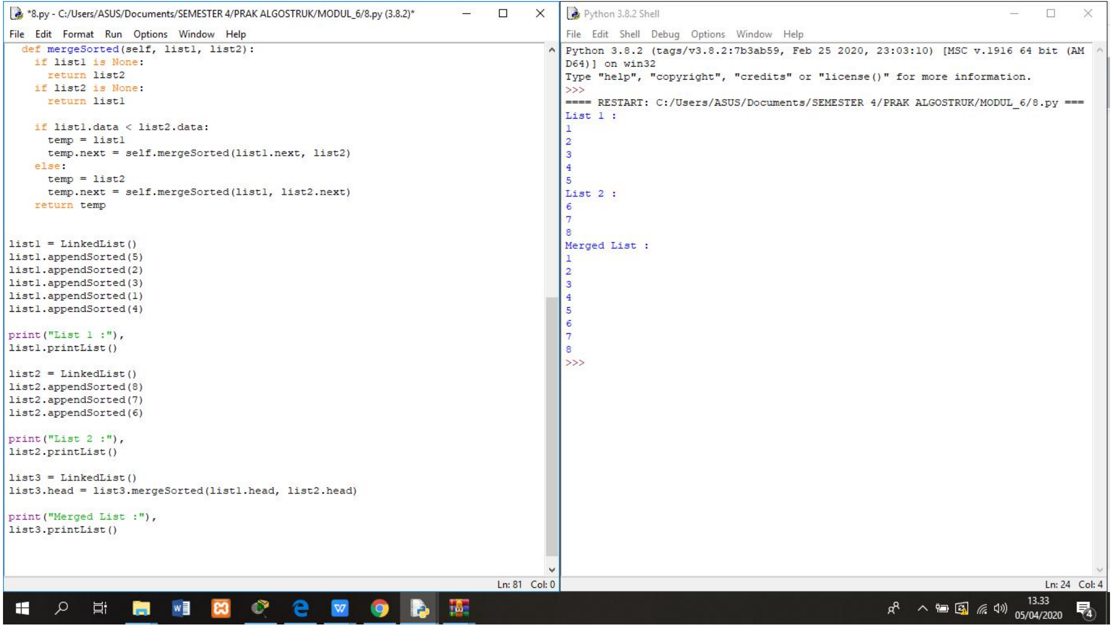
```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/7.py ===
merge : 0.0388541 detik
quick : 0.0209467 detik
merge mod : -0.00493455 detik
quick mod : -0.110021 detik
>>> 
```

8.


8.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/8.py (3.8.2)
File Edit Format Run Options Window Help
Ln: 78 Col: 17
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Ln: 24 Col: 4
13.33
05/04/2020

```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):
```

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/8.py ===
List 1 :
1
2
3
4
5
List 2 :
6
7
8
Merged List :
1
2
3
4
5
6
7
8
>>> 
```


8.py - C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/8.py (3.8.2)
File Edit Format Run Options Window Help
Ln: 81 Col: 0
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Ln: 24 Col: 4
13.33
05/04/2020

```python
  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(5)
list1.appendSorted(2)
list1.appendSorted(3)
list1.appendSorted(1)
list1.appendSorted(4)

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

list2 = LinkedList()
list2.appendSorted(8)
list2.appendSorted(7)
list2.appendSorted(6)

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

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

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

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/ASUS/Documents/SEMESTER 4/PRAK ALGOSTRUK/MODUL_6/8.py ===
List 1 :
1
2
3
4
5
List 2 :
6
7
8
Merged List :
1
2
3
4
5
6
7
8
>>>
```