Nama : W. Faisal Hari Dewanto
NIM : L200180046
Kelas : B

Modul 6

Kelas Mahasiswa

mahasiswa.py - C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\M...

File Edit Format Run Options Window Help

```
class mahasiswa():
    """ class mahsiswa yang dibangun dai class manusia """
    def __init__(self,nama,NIM,kota,us):
        self.Nama=nama
        self.NIM=NIM
        self.kota=kota
        self.uang=us
    def __str__(self):
        s=self.nama+',NIM '+str(self.NIM)\
           +'. tinggal di '+self.kota\
           +'. uang saku Rp '+str(self.uang)\
           +'. tiap bulan'
        return s
    def ambilin(self):
        return self.Nama
    def ambilnim(self):
        return self.NIM
    def ambiluang(self):
        return self.uang
    def makan(self,s):
        print ("saya makan",s)
        self.keadaan='kenyang'
    def pkota(self):
        return self.kota
    def perbarui(self,x):
        self.kota=x
    def tambah(self,x):
        self.uang=self.uang+x

class mhsTIF(mahasiswa):
    def katakan(self):
        print('hallo')
```

1.

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL - 06\1.py
35000
40000
40000
45000
45000
50000
50000
==========================================
35000
40000
40000
45000
45000
50000
50000
>>>
```

Ln: 20 Col: 4

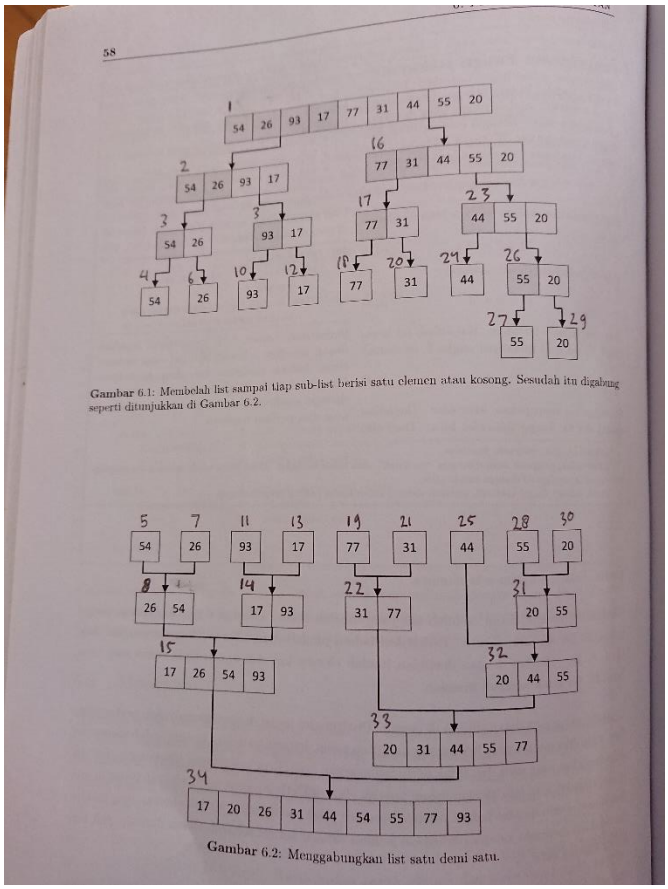
1.py - C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL - 06\1.py (3.8.1)

File Edit Format Run Options Window Help

```
import mahasiswa as mhs

c1=mhs.mhsTIF('Faisal',10,'Klaten',240000)
c2=mhs.mhsTIF('Luqman',2,'Tangerang',250000)
c3=mhs.mhsTIF('Anna',18,'Jakxel',235000)
c4=mhs.mhsTIF('Sofia',4,'Semarang',240000)
c5=mhs.mhsTIF('Nabila',31,'Brebees',250000)
c6=mhs.mhsTIF('Claire',13,'Badung',245000)
c7=mhs.mhsTIF('Armstrong',23,'Denpasar',245000)

x = [c1, c2, c3, c4, c5, c6, c7]


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)
```

Ln: 9 Col: 47

2.

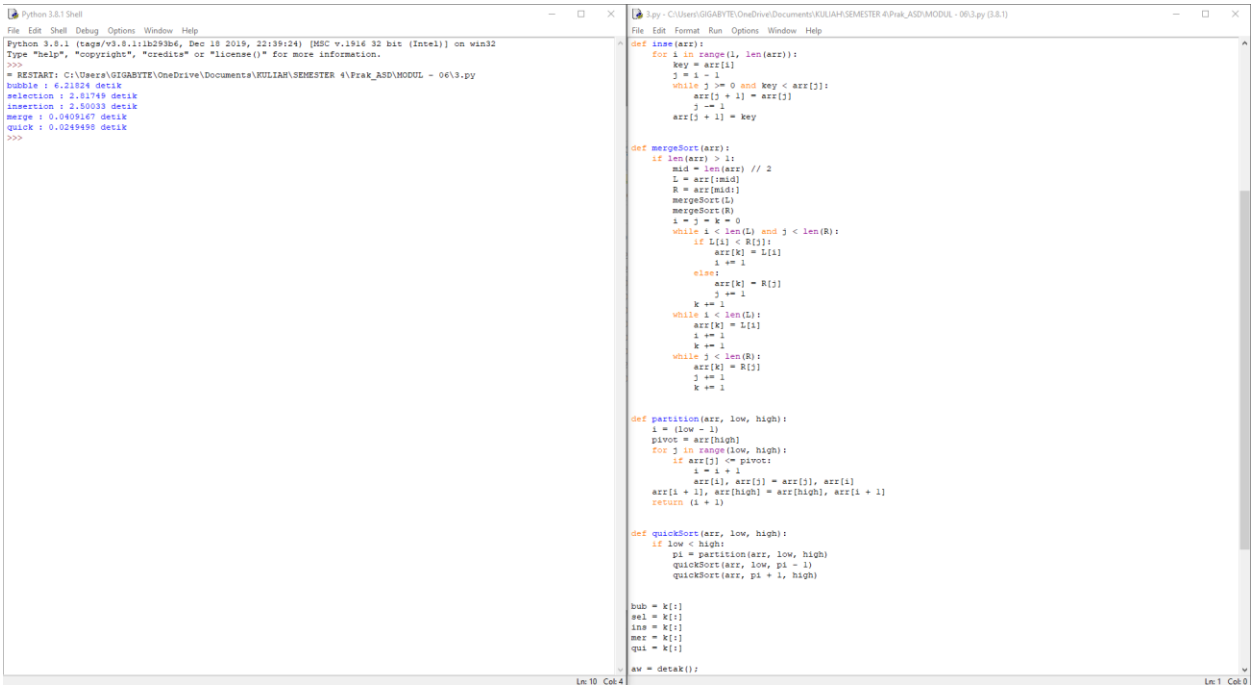


58

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.

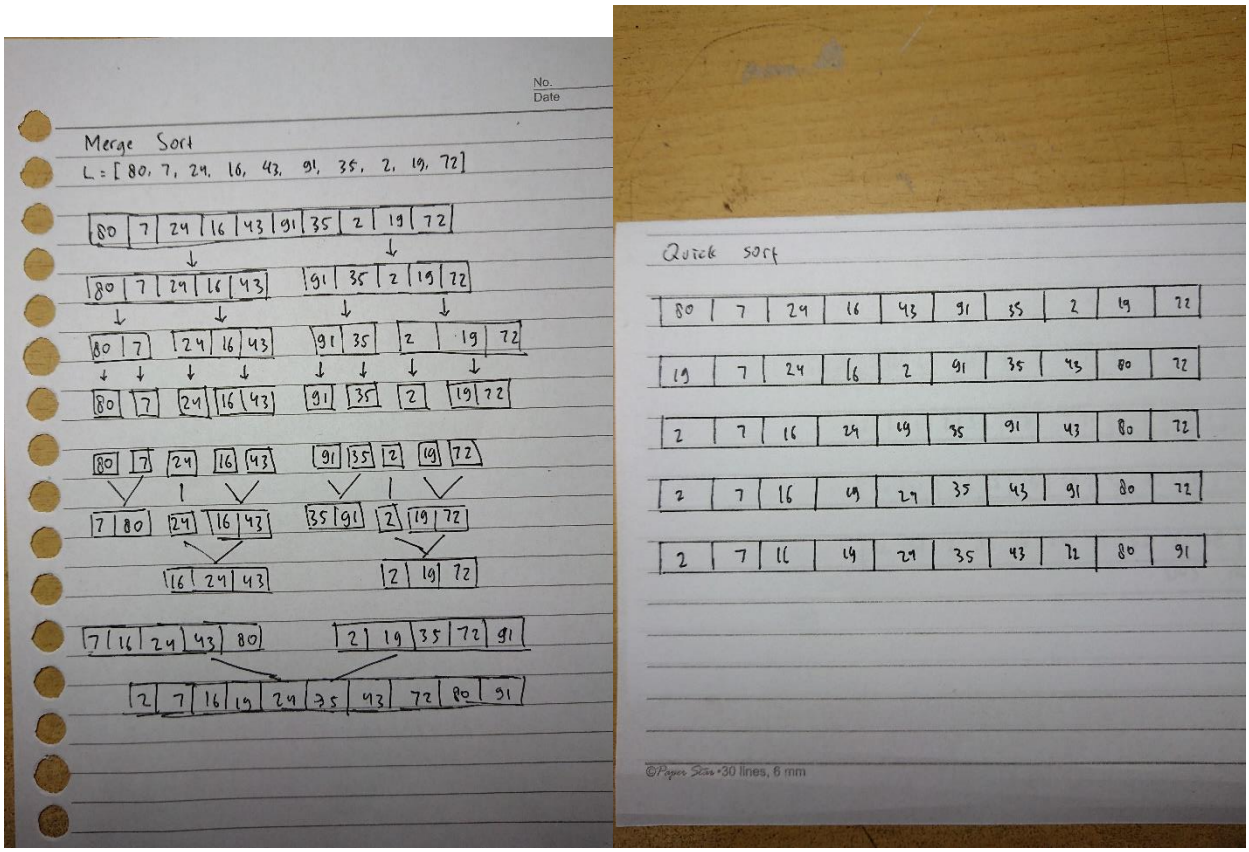
3.



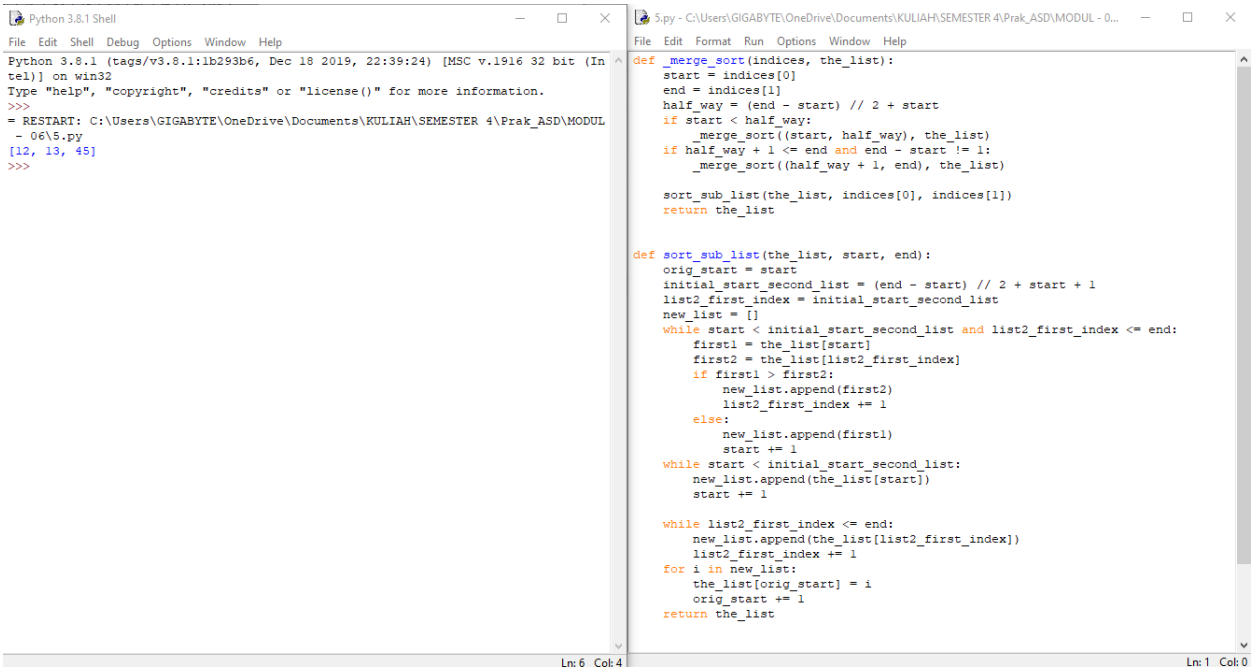
```
Python 3.8.1 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL - 06\3.py
bubble : 6.21824 detik
selection : 2.81749 detik
insertion : 2.50033 detik
merge : 0.0409167 detik
quick : 0.0249498 detik
>>>
```

```
3.py - C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL - 06\3.py (3.8.1)
File  Edit  Format  Run  Options  Window  Help
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();
```

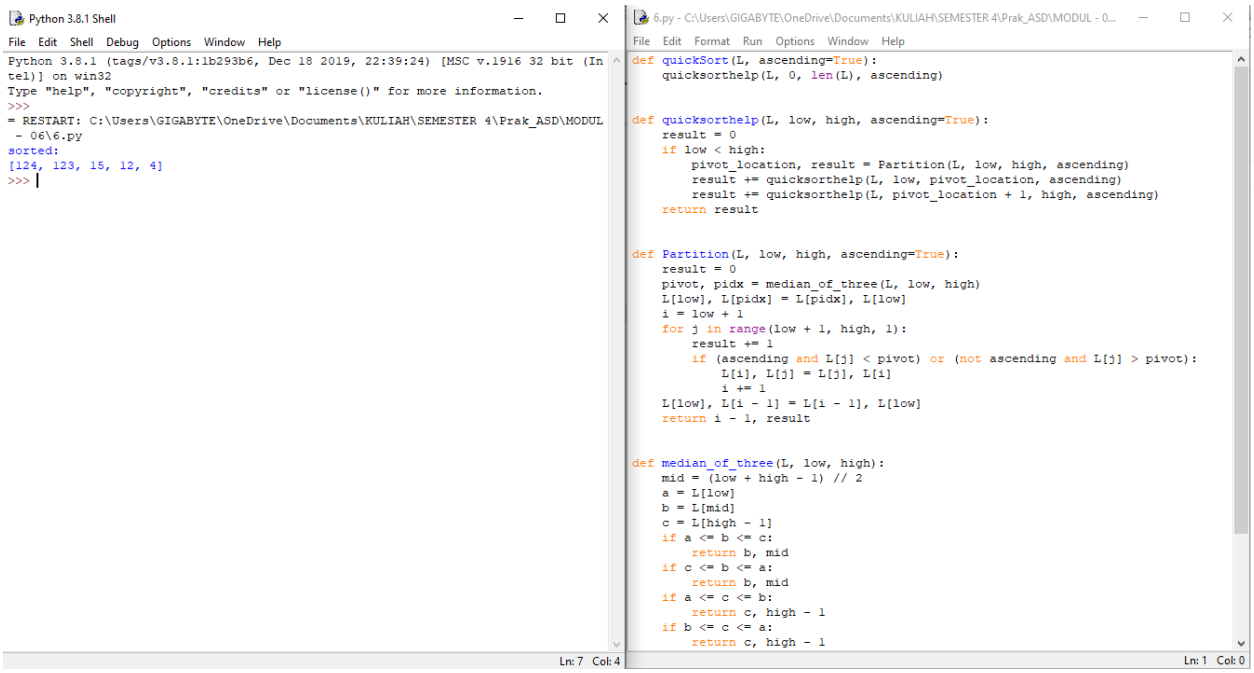
4.


Merge Sort
L = [80, 7, 24, 16, 43, 91, 35, 2, 19, 72]
Quick sort
©Paper Star •30 lines, 6 mm

5.


Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL - 06\5.py
[12, 13, 45]
>>>
Ln: 6 Col: 4
5.py - C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL - 0...
File Edit Format Run Options Window Help
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
Ln: 1 Col: 0

6.



Python 3.8.1 Shell

```
File  Edit  Shell  Debug  Options  Window  Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL
 - 06\6.py
sorted:
[124, 123, 15, 12, 4]
>>>
```

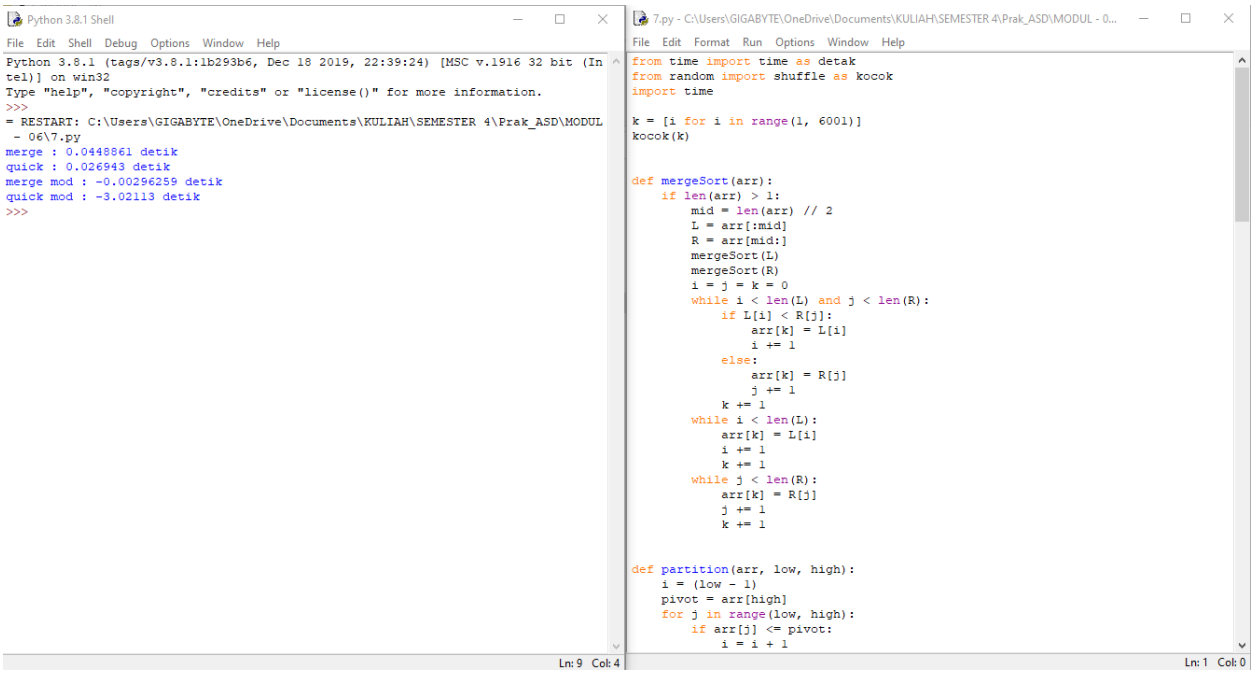
Ln: 7 Col: 4

6.py - C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL - 0...

```
File  Edit  Format  Run  Options  Window  Help
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
```

Ln: 1 Col: 0

7.



Python 3.8.1 Shell

```
File  Edit  Shell  Debug  Options  Window  Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL
 - 06\7.py
merge : 0.0448861 detik
quick : 0.026943 detik
merge mod : -0.00296259 detik
quick mod : -3.02113 detik
>>>
```

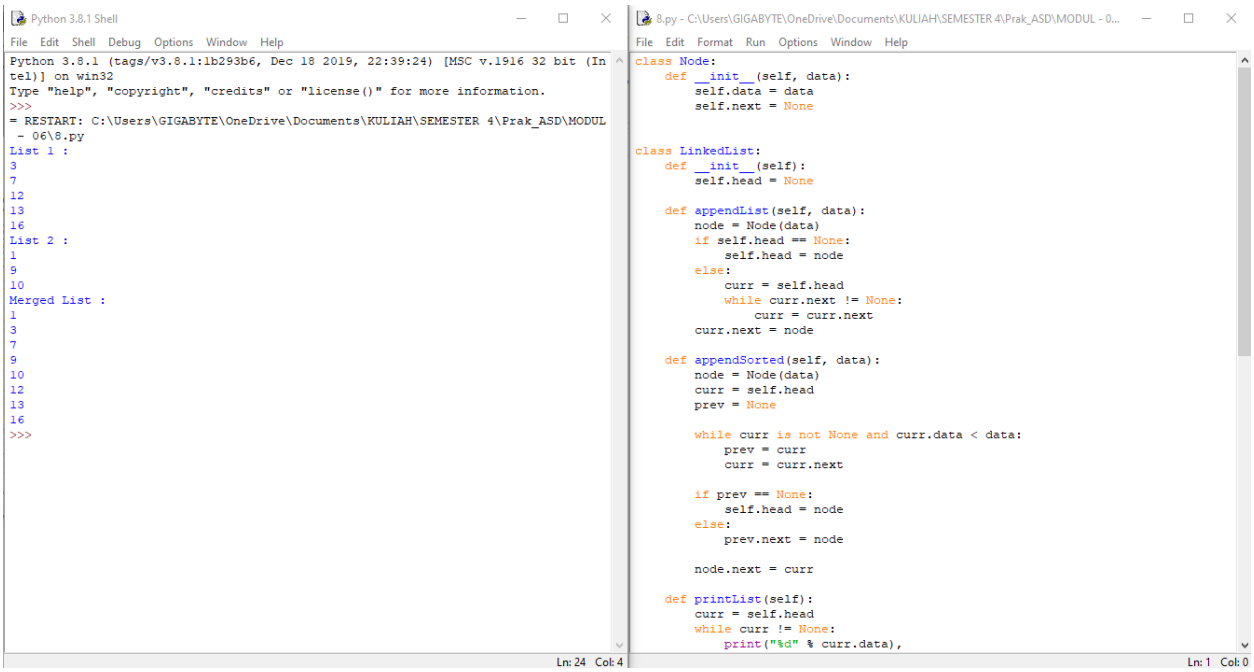
Ln: 9 Col: 4

7.py - C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL - 0...

```
File  Edit  Format  Run  Options  Window  Help
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
```

Ln: 1 Col: 0

8.



Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL
 - 06\8.py
List 1 :
3
7
12
13
16
List 2 :
1
9
10
Merged List :
1
3
7
9
10
12
13
16
>>>
```

Ln: 24 Col: 4

8.py - C:\Users\GIGABYTE\OneDrive\Documents\KULIAH\SEMESTER 4\Prak_ASD\MODUL - 0...

File Edit Format Run Options Window Help

```
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),
```

Ln: 1 Col: 0