

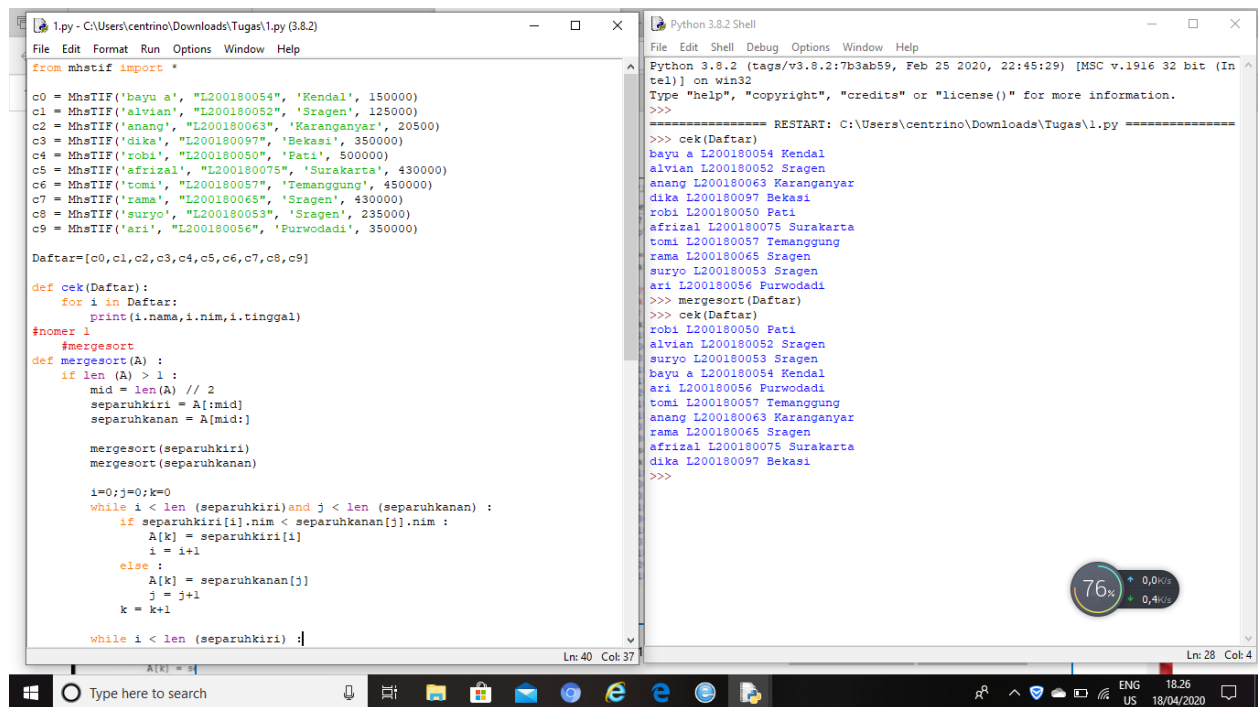
Nama : Ari wibowo

Nim : L200180056

Kelas : C

Tugas

1. A) Mergesort (mengurutkan mhstif)



The screenshot shows a Python IDE with two windows. The left window displays the Mergesort algorithm implementation, and the right window shows the output of the program.

```
from mhstif import *

c0 = MhsTIF('bayu a', 'L200180054', 'Kendal', 150000)
c1 = MhsTIF('alvian', 'L200180052', 'Sragen', 125000)
c2 = MhsTIF('anang', 'L200180063', 'Karanganyar', 205000)
c3 = MhsTIF('dika', 'L200180097', 'Bekasi', 350000)
c4 = MhsTIF('robi', 'L200180050', 'Pati', 500000)
c5 = MhsTIF('afrizal', 'L200180075', 'Surakarta', 430000)
c6 = MhsTIF('tomi', 'L200180057', 'Temanggung', 450000)
c7 = MhsTIF('rama', 'L200180065', 'Sragen', 430000)
c8 = MhsTIF('suryo', 'L200180053', 'Sragen', 235000)
c9 = MhsTIF('ari', 'L200180056', 'Purwodadi', 350000)

Daftar=[c0,c1,c2,c3,c4,c5,c6,c7,c8,c9]

def cek(Daftar):
    for i in Daftar:
        print(i.nama,i.nim,i.tinggal)

#nomer 1
#mergesort
def mergesort(A):
    if len(A) > 1:
        mid = len(A) // 2
        separuhkiri = A[:mid]
        separuhkanan = A[mid:]

        mergesort(separuhkiri)
        mergesort(separuhkanan)

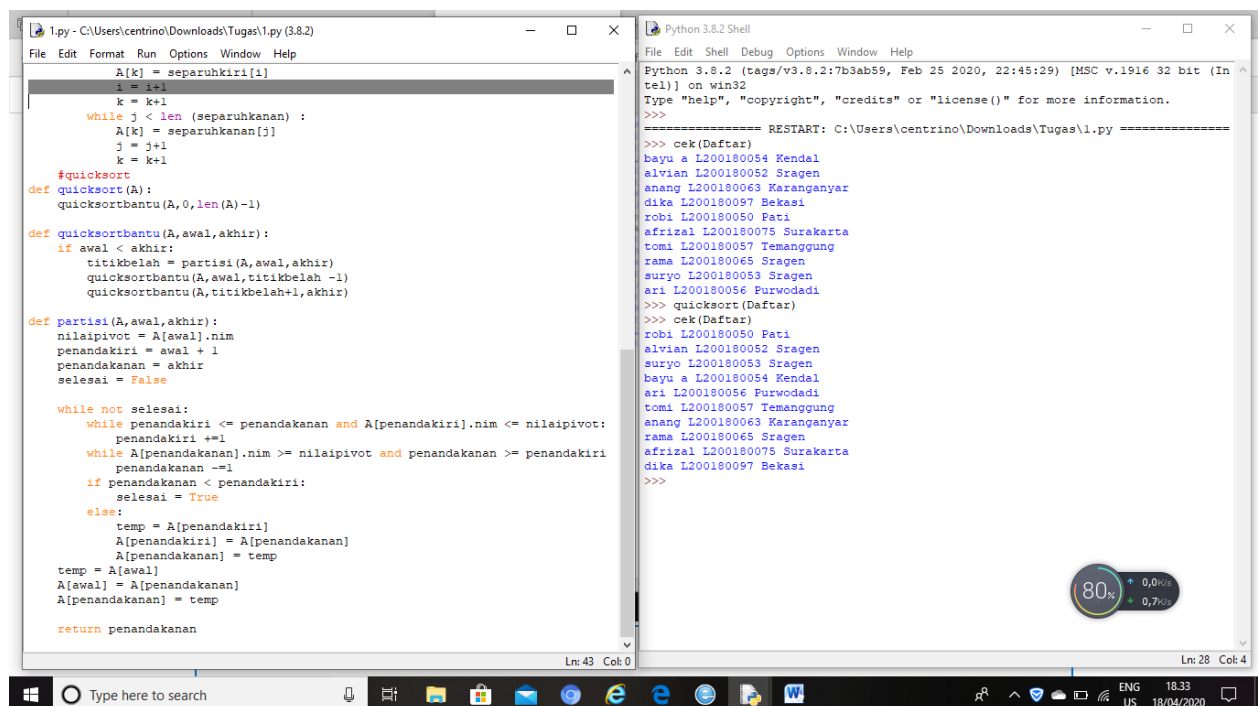
        i=0;j=0;k=0
        while i < len(separuhkiri) and j < len(separuhkanan):
            if separuhkiri[i].nim < separuhkanan[j].nim:
                A[k] = separuhkiri[i]
                i = i+1
            else:
                A[k] = separuhkanan[j]
                j = j+1
            k = k+1
        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

    return A
```

The output window shows the following results:

```
>>> cek(Daftar)
bayu a L200180054 Kendal
alvian L200180052 Sragen
anang L200180063 Karanganyar
dika L200180097 Bekasi
robi L200180050 Pati
afrizal L200180075 Surakarta
tomi L200180057 Temanggung
rama L200180065 Sragen
suryo L200180053 Sragen
ari L200180056 Purwodadi
>>> mergesort(Daftar)
>>> cek(Daftar)
robi L200180050 Pati
alvian L200180052 Sragen
suryo L200180053 Sragen
bayu a L200180054 Kendal
ari L200180056 Purwodadi
tomi L200180057 Temanggung
anang L200180063 Karanganyar
rama L200180065 Sragen
afrizal L200180075 Surakarta
dika L200180097 Bekasi
>>>
```

B) Quicksort (mengurutkan mhstif)



The screenshot shows a Python IDE with two windows. The left window displays the Quicksort algorithm implementation, and the right window shows the output of the program.

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

#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

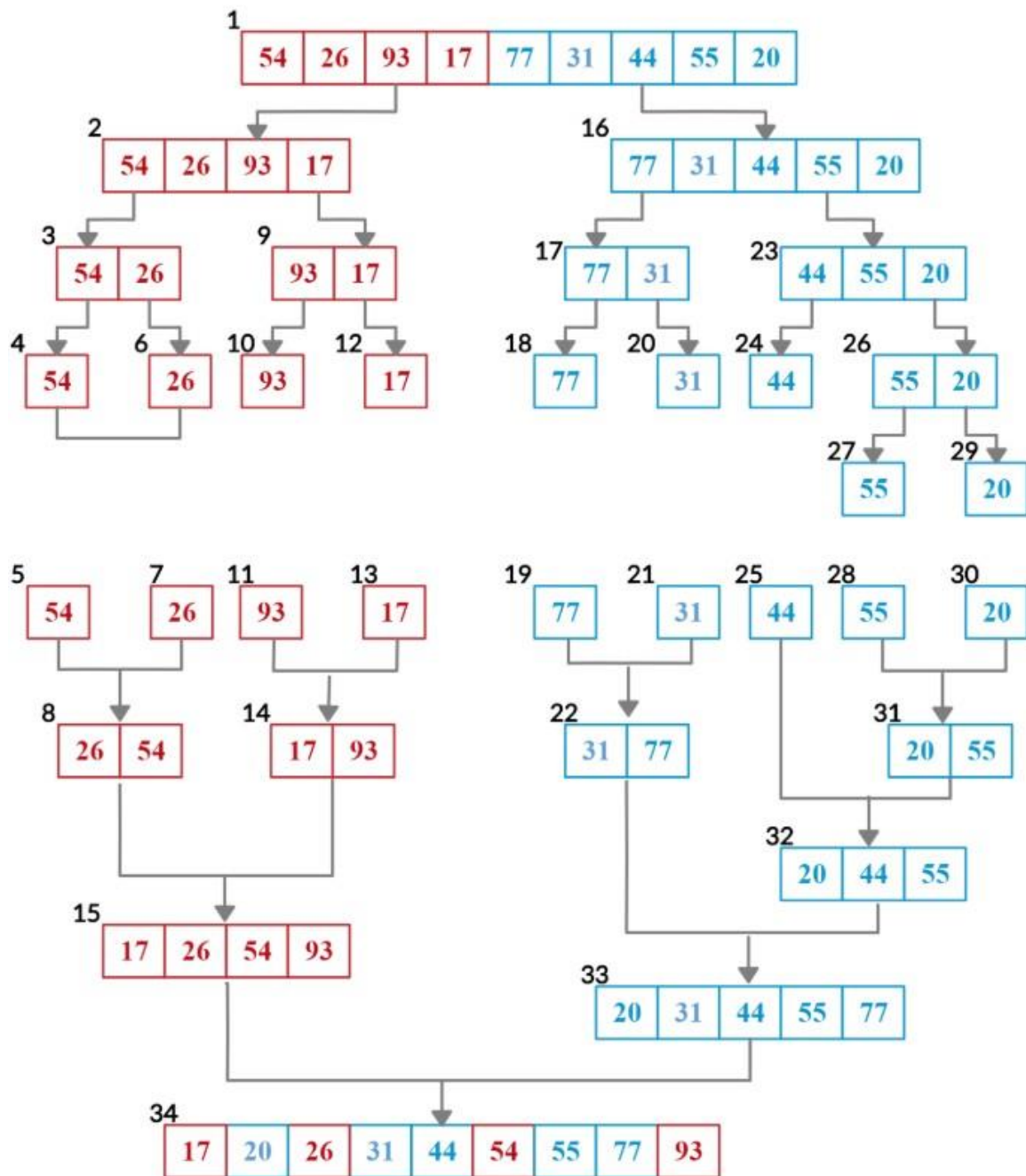
    return penandakanan
```

The output window shows the following results:

```
>>> cek(Daftar)
bayu a L200180054 Kendal
alvian L200180052 Sragen
anang L200180063 Karanganyar
dika L200180097 Bekasi
robi L200180050 Pati
afrizal L200180075 Surakarta
tomi L200180057 Temanggung
rama L200180065 Sragen
suryo L200180053 Sragen
ari L200180056 Purwodadi
>>> quicksort(Daftar)
>>> cek(Daftar)
robi L200180050 Pati
alvian L200180052 Sragen
suryo L200180053 Sragen
bayu a L200180054 Kendal
ari L200180056 Purwodadi
tomi L200180057 Temanggung
anang L200180063 Karanganyar
rama L200180065 Sragen
afrizal L200180075 Surakarta
dika L200180097 Bekasi
>>>
```

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

halaman 58



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

The image shows a Python 3.8.2 IDE with two windows. The left window displays a Python script for a sorting algorithm, and the right window shows the execution output.

Python Script (Left Window):

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

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

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[:]

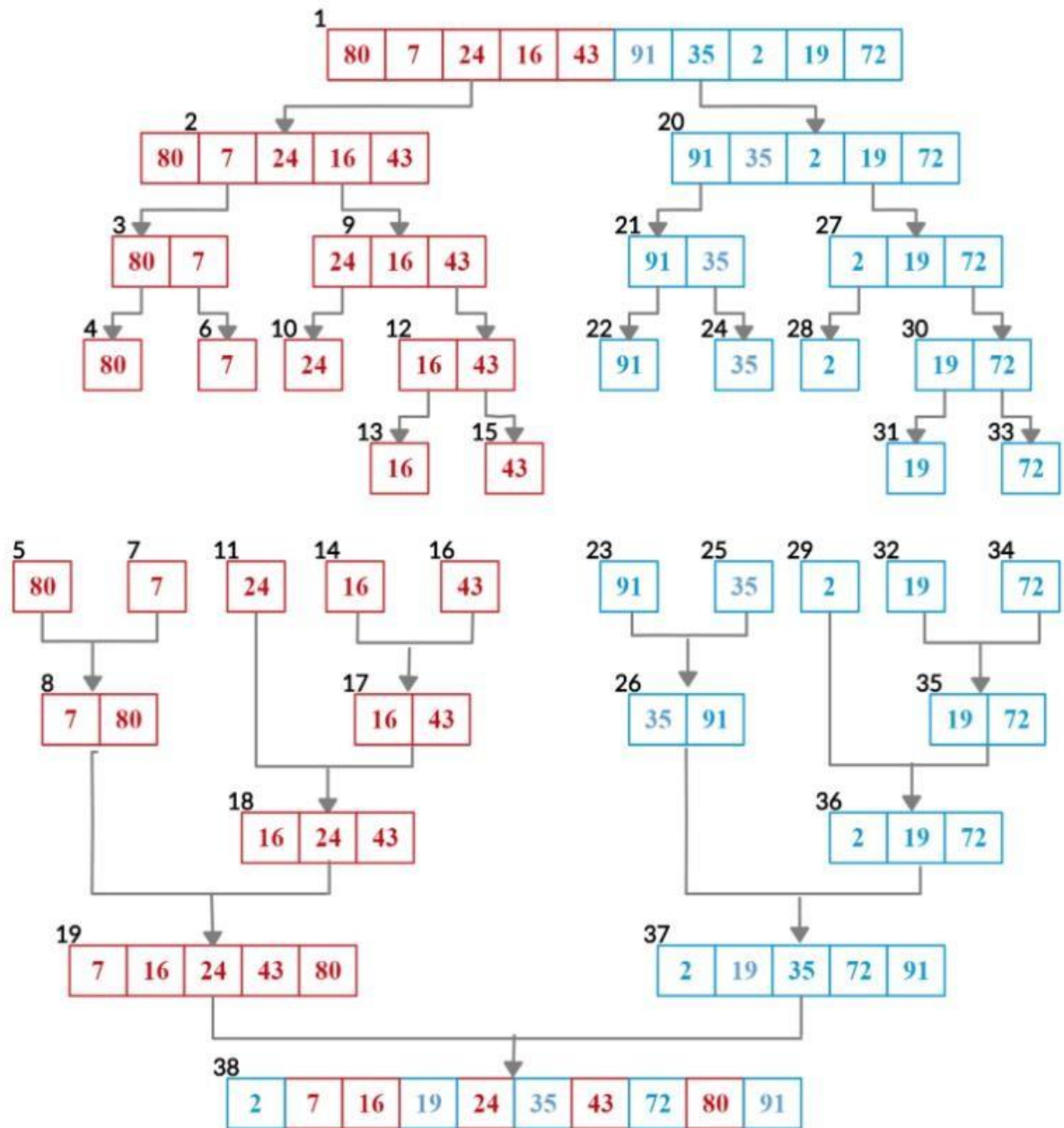
aw=detak();bubbleSort(u_bub);ak=detak();print("bubble: %g detik" %(ak-aw));
aw=detak();selectionSort(u_sel);ak=detak();print("selection: %g detik" %(ak-aw));
aw=detak();insertionSort(u_ins);ak=detak();print("insertion: %g detik" %(ak-aw));
aw=detak();mergeSort(u_mrg);ak=detak();print("merge: %g detik" %(ak-aw));
aw=detak();quickSort(u_qck);ak=detak();print("quick: %g detik" %(ak-aw));
```

Execution Output (Right Window):

```
Python 3.8.2 Shell
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\centrino\Downloads\Tugas\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: 25.5383 detik
selection: 10.0493 detik
insertion: 10.2093 detik
merge: 0.129998 detik
quick: 0.0999982 detik
>>>
===== RESTART: C:\Users\centrino\Downloads\Tugas\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: 25.8382 detik
selection: 9.88933 detik
insertion: 10.1093 detik
merge: 0.129999 detik
quick: 0.0999916 detik
>>>
```

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

nomer 4 . merge sort



a. Diberikan List = [80,7,24,16,43,91,35,2,19,72] ,gambarlah trace pengurutan algoritmanya (quicksort)

QuickSort

List = [80,7,24,16,43,91,35,2,19,72]

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

pivot

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

Low High

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				

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

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

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

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

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 megersort dengan tidak memakai operator A[:mid] dan A[mid:]

The screenshot shows two windows from a Windows desktop. The left window is a 'Python 3.8.2 Shell' with the following content:

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
daftar===== RESTART: C:\Users\centrino\Downloads\Tugas\5.py =====
>>> daftar
[54, 26, 93, 17, 77, 31, 44, 55, 20]
>>> mergeSort(daftar)
>>> daftar
[17, 20, 26, 31, 44, 54, 55, 77, 93]
>>>
```

The right window is a text editor showing the implementation of a merge sort algorithm in Python:

```
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)
```

The taskbar at the bottom shows the Windows Start button, a search bar, and several application icons. A system tray in the bottom right corner displays a battery level of 89%, network status, and the date 18/04/2020.

6. Quicksort dengan memakai metode median-dari-tiga


```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
daftarae===== RESTART: C:\Users\centrino\Downloads\Tugas\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 - C:\Users\centrino\Downloads\Tugas\6.py (3.8.2)
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 nomor 5 dan 6 mergesort (awal) dan quicksort (akhir)

```
7.py - C:\Users\centrino\Downloads\Tugas\7.py (3.8.2)
File Edit Format Run Options Window Help
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 += 1
            else:
                A[k] = separuhkanan[j]
                j += 1
            k += 1
        while i < len(separuhkiri):
            A[k] = separuhkiri[i]
            i += 1
            k += 1
        while j < len(separuhkanan):
            A[k] = separuhkanan[j]
            j += 1
            k += 1
    alist = [54,26,93,17,77,31,44,55,20]

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 += 1
        while A[penandakanan] >= nilaipivot and penandakanan >= penandakiri:
            penandakanan -= 1
        if penandakanan < penandakiri:
            selesai = True
        else:
            A[penandakiri], A[penandakanan] = A[penandakanan], A[penandakiri]
            penandakiri += 1
            penandakanan -= 1

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\centrino\Downloads\Tugas\7.py =====
mergesort : 0.159989 detik
mergesort terbaru : 0.199983 detik
quicksort : 0.0899861 detik
quicksort terbaru : 0.119991 detik
>>>
```

8. Versi linked list mergesort

```
8.py - C:\Users\centrino\Downloads\Tugas\8.py (3.8.2)
File Edit Format Run Options Window Help

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

class Linked():
    def __init__(self,head=None):
        self.head = head

    def cetak(self):
        cur = self.head
        while cur != None:
            print(cur.data)
            cur = cur.next

    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

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\centrino\Downloads\Tugas\8.py =====
List 1 :
16
17
33
48
92
List 2 :
10
18
23
Mergesort Linked list :
10
16
17
18
23
33
48
92
>>> |

83% 1.0% 161,1%
19.09
18/04/2020
```