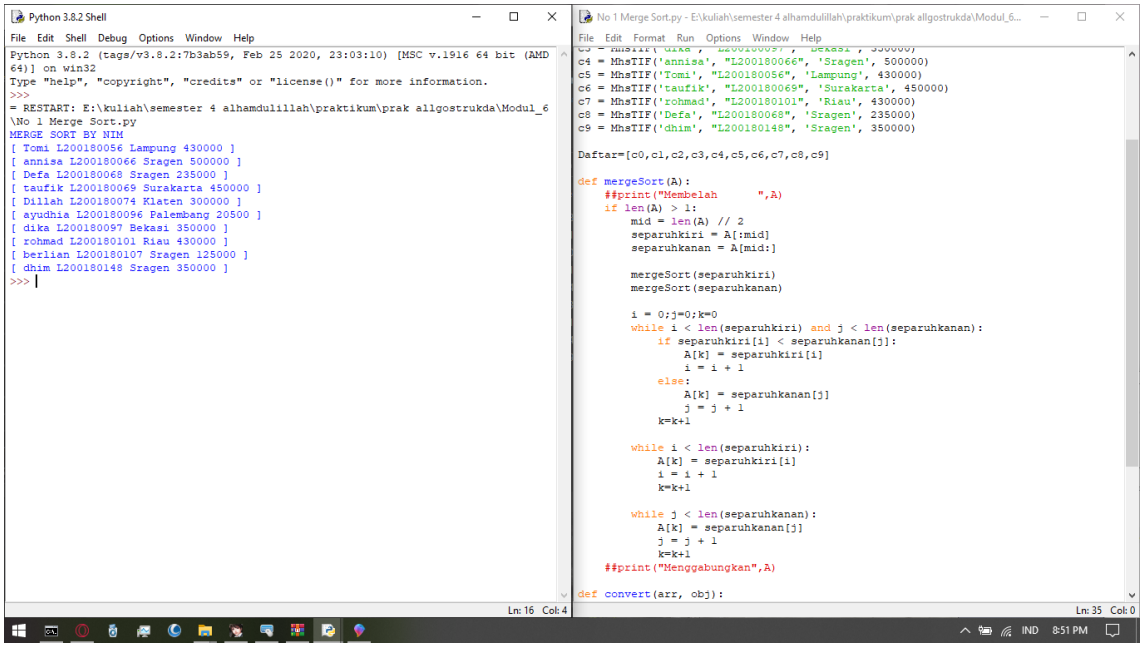
Nama : Abdillah Ahmad
NIM : L200180074
Kelas : C

TUGAS PRAKTIKUM ALGORITMA STRUKTUR DATA MODUL 6

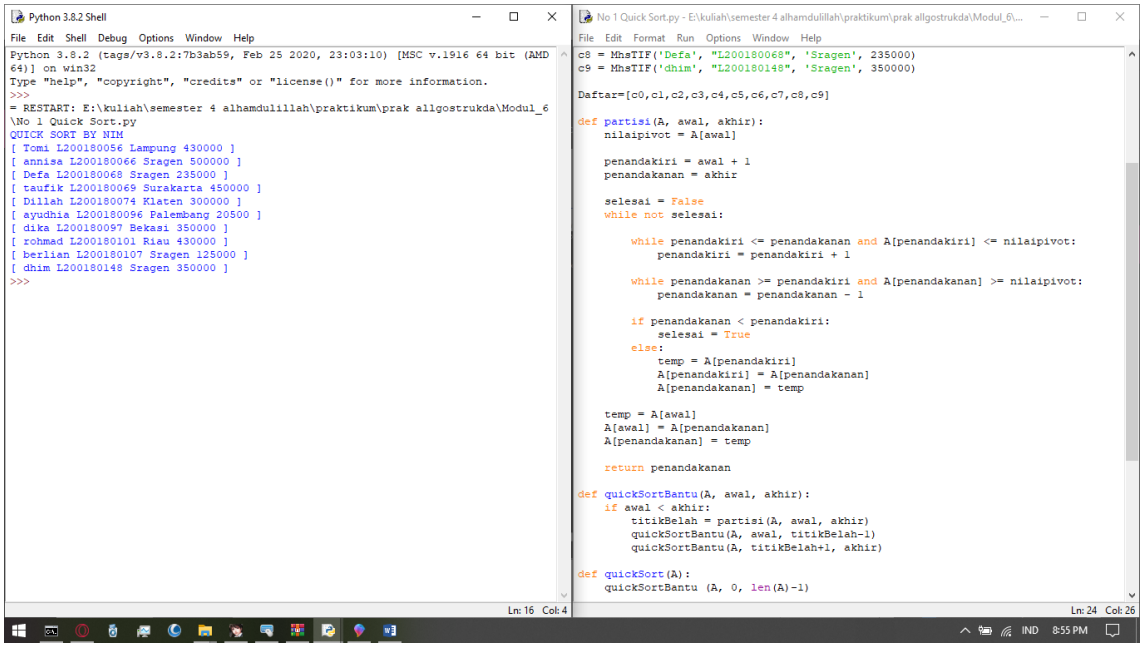
No 1
Merge Sort



```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD
64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\kuliah\semester 4 alhamdulillah\praktikum\prak allgostrukda\Modul_6
\No 1 Merge Sort.py
MERGE SORT BY NIM
[ Tomi L200180056 Lampung 430000 ]
[ annisa L200180066 Sragen 500000 ]
[ Defa L200180068 Sragen 235000 ]
[ taufik L200180069 Surakarta 450000 ]
[ Dillah L200180074 Klaten 300000 ]
[ ayudhia L200180096 Palembang 20500 ]
[ dika L200180097 Bekasi 350000 ]
[ rohmad L200180101 Riau 430000 ]
[ berlian L200180107 Sragen 125000 ]
[ dhim L200180148 Sragen 350000 ]
>>>
```

```
No 1 Merge Sort.py - E:\kuliah\semester 4 alhamdulillah\praktikum\prak allgostrukda\Modul_6...
File Edit Format Run Options Window Help
c4 = MhsTIF('annisa', "L200180066", 'Sragen', 500000)
c5 = MhsTIF('Tomi', "L200180056", 'Lampung', 430000)
c6 = MhsTIF('taufik', "L200180069", 'Surakarta', 450000)
c7 = MhsTIF('rohmad', "L200180101", 'Riau', 430000)
c8 = MhsTIF('Defa', "L200180068", 'Sragen', 235000)
c9 = MhsTIF('dhim', "L200180148", 'Sragen', 350000)

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

def mergeSort(A):
    ##print("Membelah      ",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
    ##print("Menggabungkan",A)

def convert(arr, obj):
```

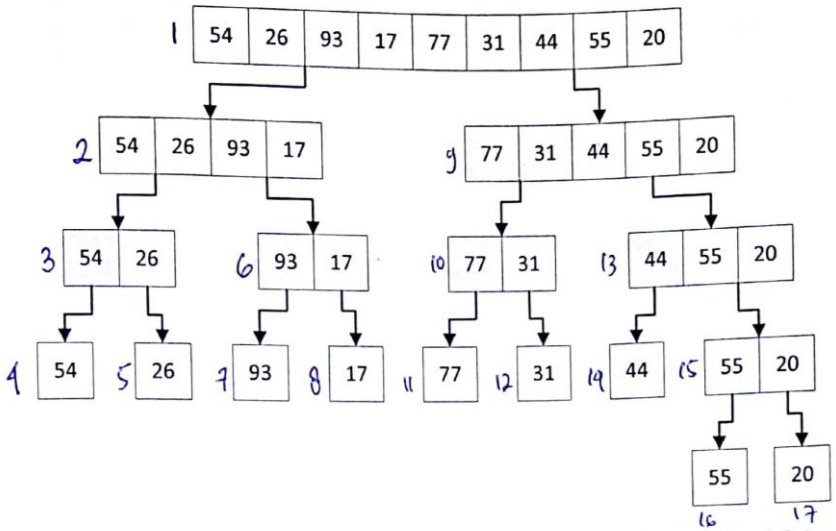
Quick Sort



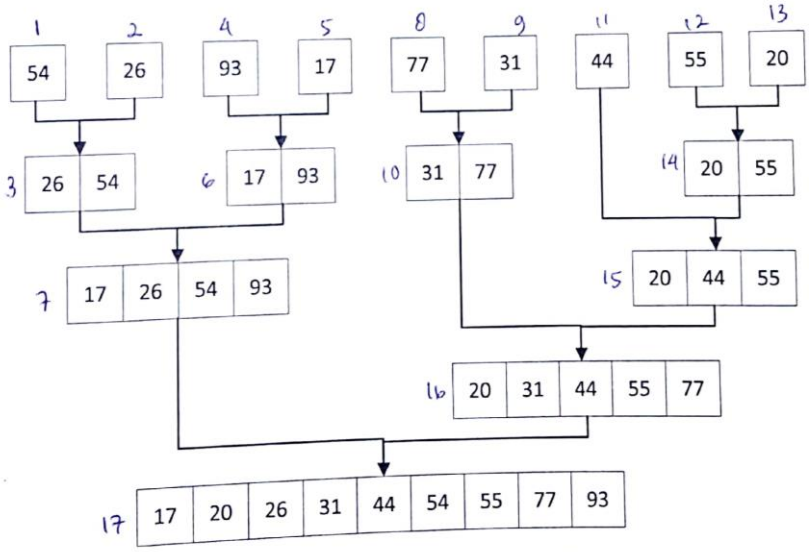
```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD
64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\kuliah\semester 4 alhamdulillah\praktikum\prak allgostrukda\Modul_6
\No 1 Quick Sort.py
QUICK SORT BY NIM
[ Tomi L200180056 Lampung 430000 ]
[ annisa L200180066 Sragen 500000 ]
[ Defa L200180068 Sragen 235000 ]
[ taufik L200180069 Surakarta 450000 ]
[ Dillah L200180074 Klaten 300000 ]
[ ayudhia L200180096 Palembang 20500 ]
[ dika L200180097 Bekasi 350000 ]
[ rohmad L200180101 Riau 430000 ]
[ berlian L200180107 Sragen 125000 ]
[ dhim L200180148 Sragen 350000 ]
>>>
```

```
No 1 Quick Sort.py - E:\kuliah\semester 4 alhamdulillah\praktikum\prak allgostrukda\Modul_6\...
File Edit Format Run Options Window Help
c8 = MhsTIF('Defa', "L200180068", 'Sragen', 235000)
c9 = MhsTIF('dhim', "L200180148", 'Sragen', 350000)

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

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 penandakanan >= penandakiri and A[penandakanan] >= nilaipivot:
            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

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)
```

No 2

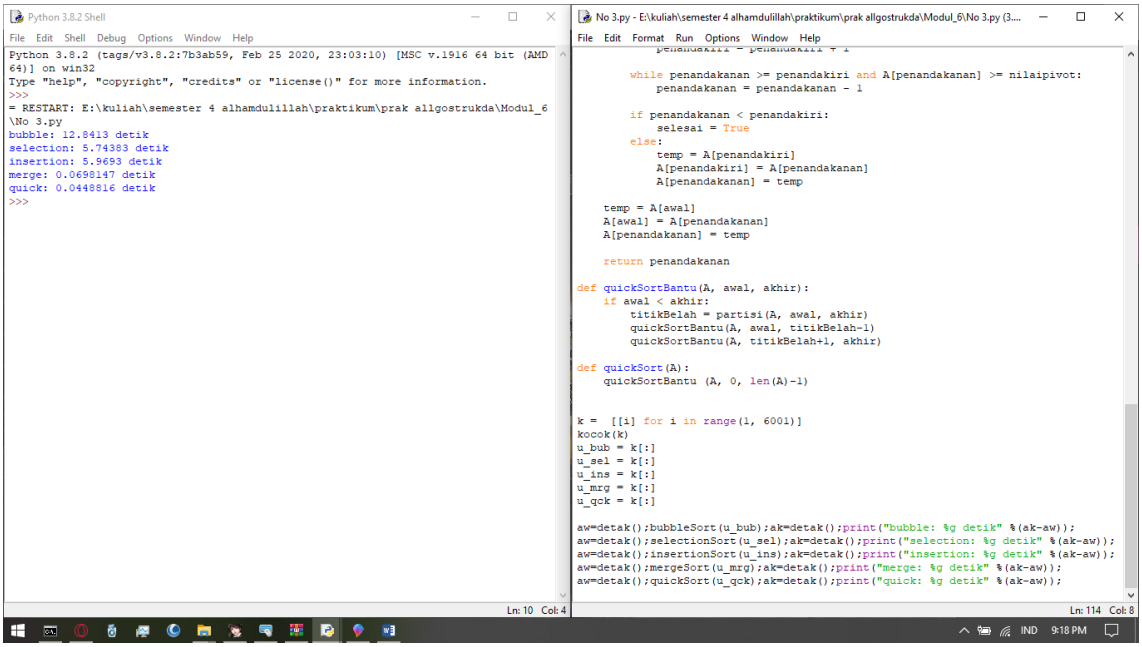


Gambar 6.1: Membelah list sampai tiap sub-list berisi satu elemen atau kosong. Sesudah itu digabung seperti ditunjukkan di Gambar 6.2.



Gambar 6.2: Menggabungkan list satu demi satu.

No 3



No 4

Merge Sort:

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

80	7	24	16	43	91	35	2	19	72

Proses 1

7	80	26	24	43	91	2	35	19	72

Proses 2

7	16	24	80	2	35	43	91	19	72

Proses 3

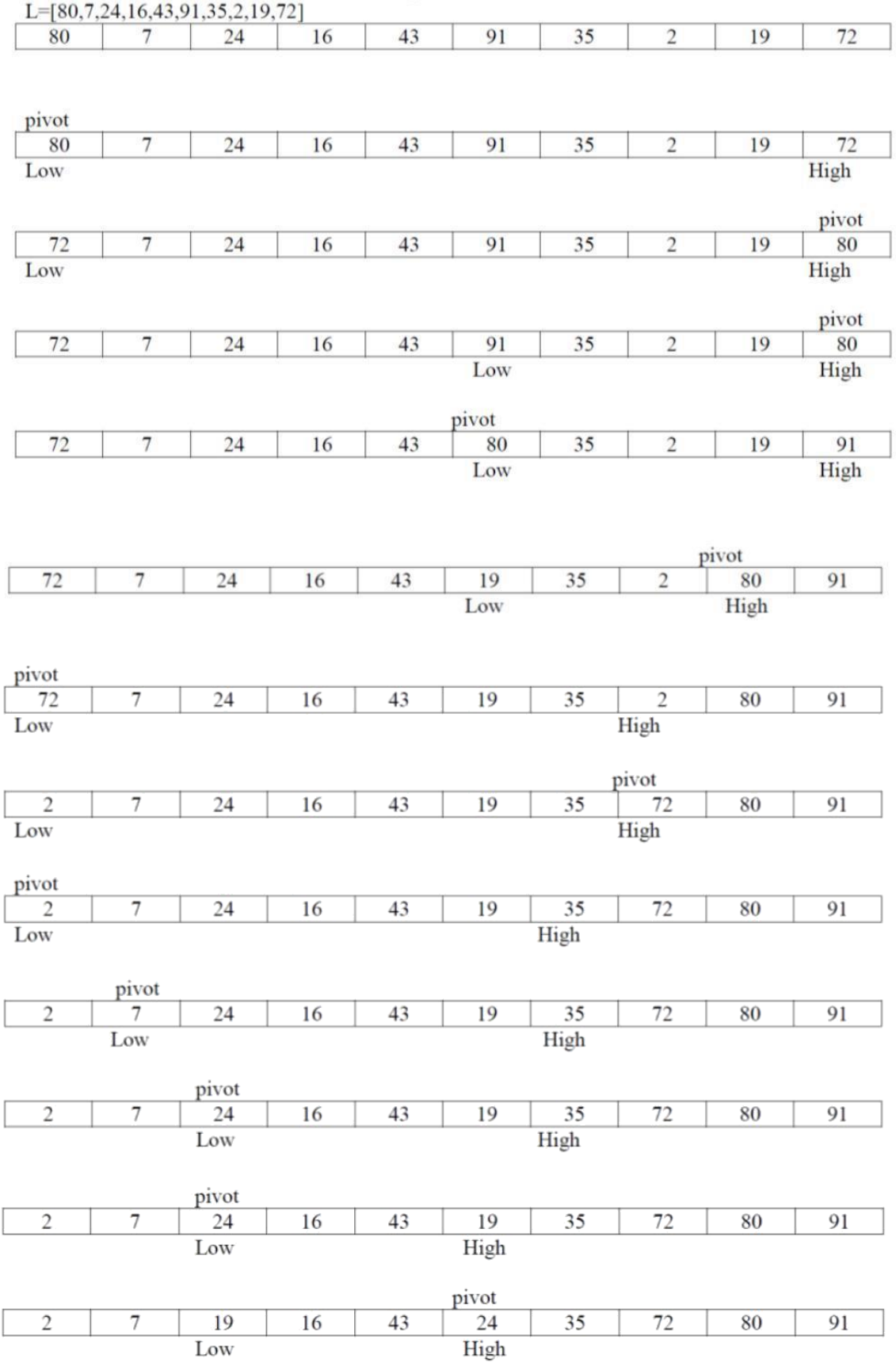
2	7	16	24	35	43	80	91	19	72

Proses 4

2	7	16	19	24	35	43	72	80	91

Quick Sort

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



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

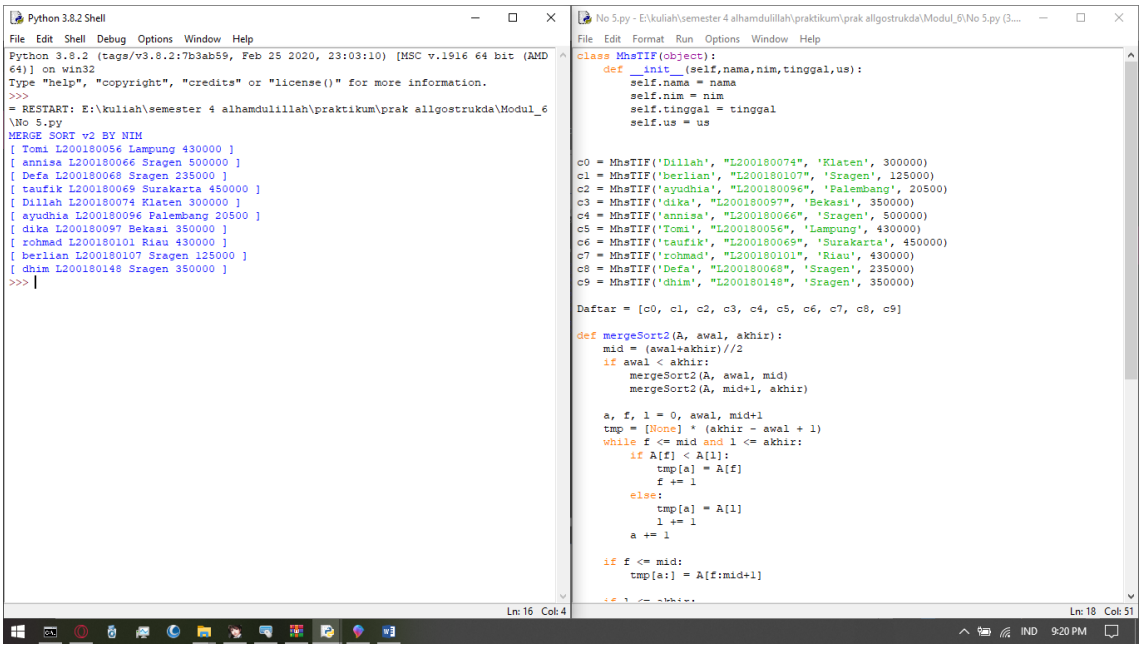
pivot

2	7	16	19	24	35	43	72	80	91

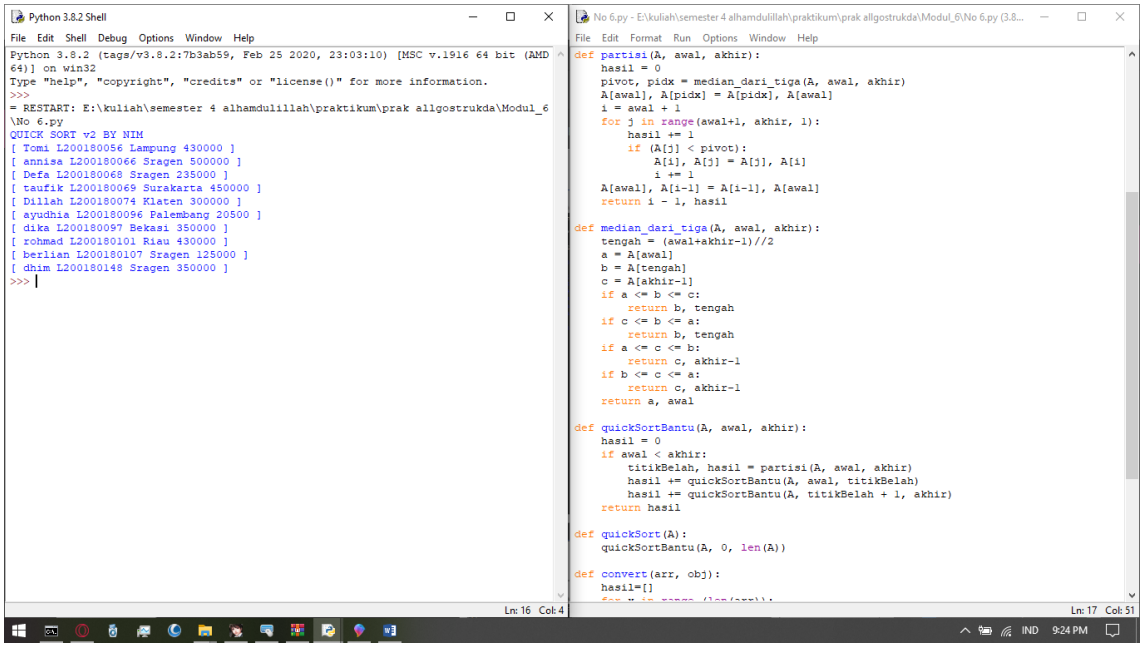
Low High

2	7	16	19	24	35	43	72	80	91

No 5



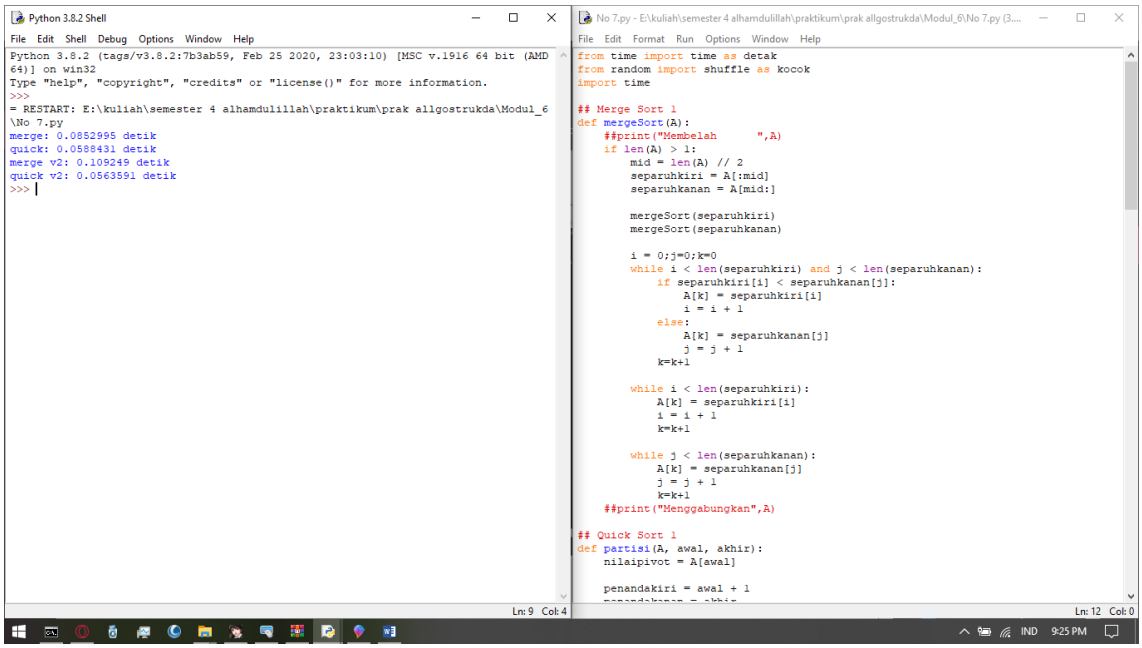
No 6



```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD
64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\kuliah\semester 4 alhamdulillah\praktikum\prak allgostrukda\Modul_6
\No 6.py
QUICK SORT v2 BY NIM
[ Tomi L200180056 Lampung 430000 ]
[ annisa L200180066 Sragen 500000 ]
[ Defa L200180068 Sragen 235000 ]
[ taufik L200180069 Surakarta 450000 ]
[ Dillah L200180074 Klaten 300000 ]
[ ayudhia L200180096 Palembang 20500 ]
[ dika L200180097 Bekasi 350000 ]
[ rohmad L200180101 Riau 430000 ]
[ berlian L200180107 Sragen 125000 ]
[ dhim L200180148 Sragen 350000 ]
>>>
```

```
def partisi(A, awal, akhir):
    hasil = 0
    pivot, pidx = median_dari_tiga(A, awal, akhir)
    A[awal], A[pidx] = A[pidx], A[awal]
    i = awal + 1
    for j in range(awal+1, akhir, 1):
        hasil += 1
        if (A[j] < pivot):
            A[i], A[j] = A[j], A[i]
            i += 1
    A[awal], A[i-1] = A[i-1], A[awal]
    return i - 1, hasil

def median_dari_tiga(A, awal, akhir):
    tengah = (awal+akhir-1)//2
    a = A[awal]
    b = A[tengah]
    c = A[akhir-1]
    if a <= b <= c:
        return b, tengah
    if c <= b <= a:
        return b, tengah
    if a <= c <= b:
        return c, akhir-1
    if b <= c <= a:
        return c, akhir-1
    return a, awal

def quickSortBantu(A, awal, akhir):
    hasil = 0
    if awal < akhir:
        titikBelah, hasil = partisi(A, awal, akhir)
        hasil += quickSortBantu(A, awal, titikBelah)
        hasil += quickSortBantu(A, titikBelah + 1, akhir)
    return hasil

def quickSort(A):
    quickSortBantu(A, 0, len(A))

def convert(arr, obj):
    hasil=[]
```

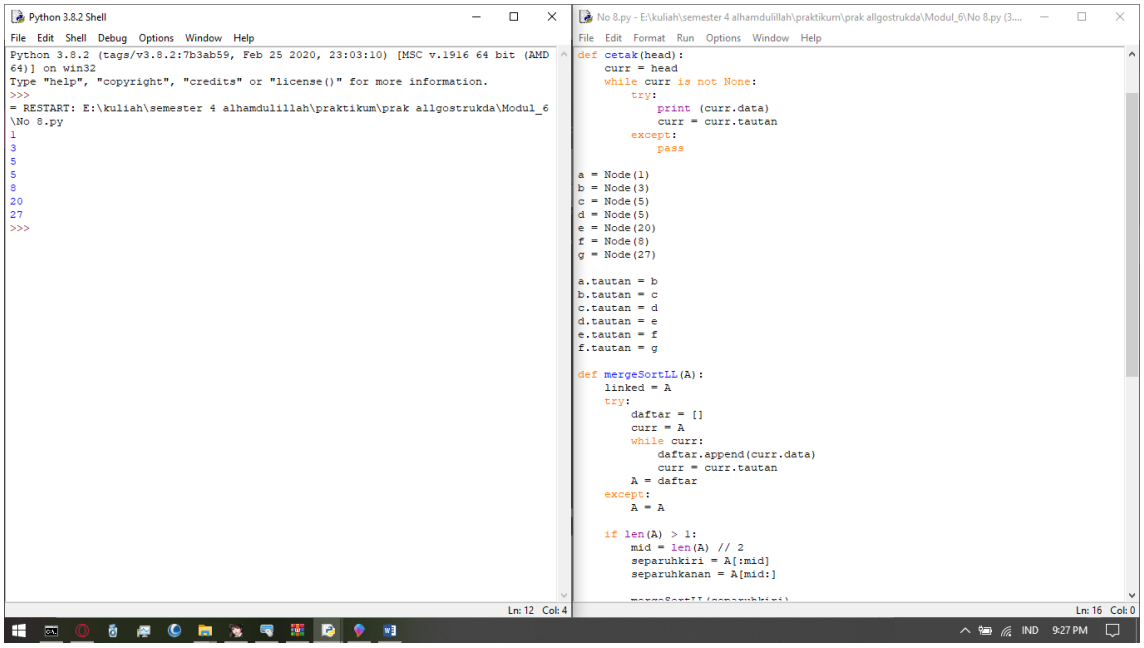
No 7



```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD
64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\kuliah\semester 4 alhamdulillah\praktikum\prak allgostrukda\Modul_6
\No 7.py
merge: 0.0852995 detik
quick: 0.0588431 detik
merge v2: 0.109249 detik
quick v2: 0.0563591 detik
>>>
```

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

## Merge Sort 1
def mergeSort(A):
    ##print("Membelah      ",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
    ##print("Menggabungkan",A)

## Quick Sort 1
def partisi(A, awal, akhir):
    nilaipivot = A[awal]

    penandakiri = awal + 1
```

No 8


Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD 64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\kuliah\semester 4 alhamdulillah\praktikum\prak allgostrukda\Modul_6\No 8.py
1
3
5
5
8
20
27
>>>
Ln: 12 Col: 4
No 8.py - E:\kuliah\semester 4 alhamdulillah\praktikum\prak allgostrukda\Modul_6\No 8.py (3....
File Edit Format Run Options Window Help
def cetak(head):
curr = head
while curr is not None:
try:
print (curr.data)
curr = curr.tautan
except:
pass
a = Node(1)
b = Node(3)
c = Node(5)
d = Node(5)
e = Node(20)
f = Node(8)
g = Node(27)
a.tautan = b
b.tautan = c
c.tautan = d
d.tautan = e
e.tautan = f
f.tautan = g
def mergeSortLL(A):
linked = A
try:
daftar = []
curr = A
while curr:
daftar.append(curr.data)
curr = curr.tautan
A = daftar
except:
A = A
if len(A) > 1:
mid = len(A) // 2
separuhkiri = A[:mid]
separuhkanan = A[mid:]
Ln: 16 Col: 0
IND 9:27 PM