

Nama : resky budi nugroho

NIM : L200180018

Kelas : A

Modul 6

1. No 1

```

        else:
            temp = A[penandakiri]
            A[penandakiri] = A[penandakanan]
            A[penandakanan] = temp

    temp = A[awal]
    A[awal] = A[penandakanan]
    A[penandakanan] = temp

    return penandakanan

def convert(arr, obj):
    hasil=[]
    for x in range (len(arr)):
        for i in range (len(arr)):
            if arr[x] == obj[i].nim:
                hasil.append(obj[i])
    return hasil

def urutkanQuick(arr):
    A = []
    for x in Daftar:
        A.append(x.nim)

    print("Quick Sort")
    quickSort(A)
    for x in convert(A, Daftar):
        print (x.nim)

def urutkanMerge(arr):
    A = []
    for x in Daftar:
        A.append(x.nim)

    print("Merge Sort")
    mergeSort(A)
    for x in convert(A, Daftar):
        print (x.nim)

urutkanMerge(Daftar)
urutkanQuick(Daftar)

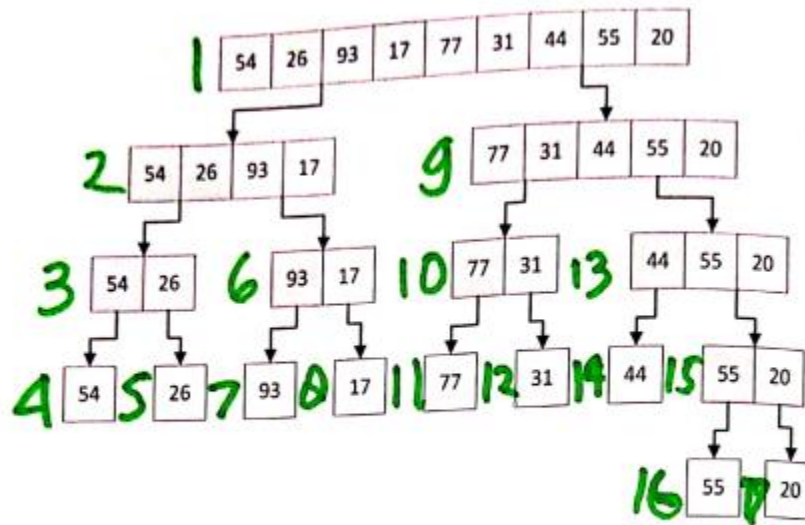
```

```

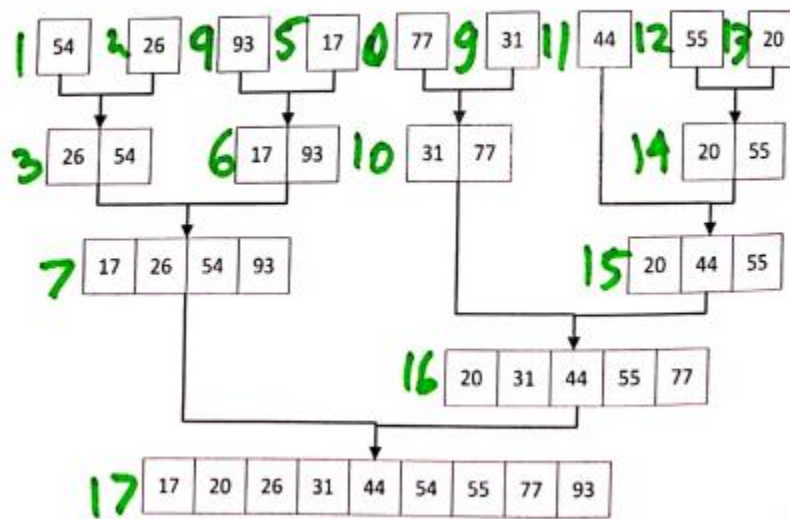
Python 3.8.2 Shell
File Edit Shell Debug Option
Python 3.8.2 (tags/v3.8
D64)] on win32
Type "help", "copyright'
>>>
===== RESTART: C:/User:
Merge Sort
2
4
5
10
13
18
23
29
31
51
64
Quick Sort
2
4
5
10
13
18
23
29
31
51
64
>>> |

```

2. 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.

```

        k=k+1

        while j < len(separuhkanan):
            A[k] = separuhkanan[j]
            j = j + 1
            k=k+1
        #print("Menggabungkan",A)

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)

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

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));

```

```

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
>>>
===== RESTART: C:/Users/ACER/Documents/GitHub/
bubble: 4.8426 detik
selection: 1.95263 detik
insertion: 2.19349 detik
merge: 0.0620854 detik
quick: 0.0156293 detik
>>>

```

4. No 4

4. a) mergeSort

List 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 |
|---|---|----|----|----|----|----|----|----|----|

5. No 5

```
#nomor 5
def cetak(A):
    for i in A:
        print (i)

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].ambilUangSaku() < A[l].ambilUangSaku():
            tmp[a] = A[f]
            f += 1
        else:
            tmp[a] = A[l]
            l += 1
        a += 1

    if f <= mid:
        tmp[a:] = A[f:mid+1]

    if l <= akhir:
        tmp[a:] = A[l:akhir+1]

    a = 0
    while awal <= akhir:
        A[awal] = tmp[a]
        awal += 1
        a += 1

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

print("Sebelum diurutkan")
cetak(Daftar)
mergeSort(Daftar)
print("\nSetelah diurutkan")
cetak(Daftar)
```

```
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
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/ACER/Documents/GitHub/algostruk/MODUL_6/tugas.py ==
Sebelum diurutkan
Ika, nim 10. Tinggal di Sukoharjo. Uang saku Rp 240000. tiap bulannya.
Budi, nim 51. Tinggal di Sragen. Uang saku Rp 230000. tiap bulannya.
Ahmad, nim 2. Tinggal di Surakarta. Uang saku Rp 250000. tiap bulannya.
Chandra, nim 18. Tinggal di Surakarta. Uang saku Rp 235000. tiap bulannya.
Eka, nim 4. Tinggal di Boyolali. Uang saku Rp 240000. tiap bulannya.
Fandi, nim 31. Tinggal di Salatiga. Uang saku Rp 250000. tiap bulannya.
Deni, nim 13. Tinggal di Klaten. Uang saku Rp 245000. tiap bulannya.
Galuh, nim 5. Tinggal di Wonogiri. Uang saku Rp 245000. tiap bulannya.
Janto, nim 23. Tinggal di Klaten. Uang saku Rp 245000. tiap bulannya.
Hasan, nim 64. Tinggal di Karanganyar. Uang saku Rp 270000. tiap bulannya.
Khalid, nim 29. Tinggal di Purwodadi. Uang saku Rp 265000. tiap bulannya.

Setelah diurutkan
Budi, nim 51. Tinggal di Sragen. Uang saku Rp 230000. tiap bulannya.
Chandra, nim 18. Tinggal di Surakarta. Uang saku Rp 235000. tiap bulannya.
Eka, nim 4. Tinggal di Boyolali. Uang saku Rp 240000. tiap bulannya.
Ika, nim 10. Tinggal di Sukoharjo. Uang saku Rp 240000. tiap bulannya.
Janto, nim 23. Tinggal di Klaten. Uang saku Rp 245000. tiap bulannya.
Galuh, nim 5. Tinggal di Wonogiri. Uang saku Rp 245000. tiap bulannya.
Deni, nim 13. Tinggal di Klaten. Uang saku Rp 245000. tiap bulannya.
Fandi, nim 31. Tinggal di Salatiga. Uang saku Rp 250000. tiap bulannya.
Ahmad, nim 2. Tinggal di Surakarta. Uang saku Rp 250000. tiap bulannya.
Khalid, nim 29. Tinggal di Purwodadi. Uang saku Rp 265000. tiap bulannya.
Hasan, nim 64. Tinggal di Karanganyar. Uang saku Rp 270000. tiap bulannya.
>>>
```

6. No 6

```
#nomoer 6
A = []
for i in Daftar:
    A.append(i.nama)

def cetak():
    for i in A:
        print(i)

def quickSort(arr):
    kurang = []
    pivotList = []
    lebih = []
    if len(arr) <= 1:
        return arr
    else:
        pivot = arr[0]
        for i in arr:
            if i < pivot:
                kurang.append(i)
            elif i > pivot:
                lebih.append(i)
            else:
                pivotList.append(i)
        kurang = quickSort(kurang)
        lebih = quickSort(lebih)
        return kurang + pivotList + lebih

print("Sebelum diurutkan")
cetak()
print("\nSetelah diurutkan")
quickSort(A)
cetak()
```

```
===== RESTART: C:/User:
Sebelum diurutkan
Ika
Budi
Ahmad
Chandra
Eka
Fandi
Deni
Galuh
Janto
Hasan
Khalid

Setelah diurutkan
Ika
Budi
Ahmad
Chandra
Eka
Fandi
Deni
Galuh
Janto
Hasan
Khalid
>>> |
```

7. No 7

```

        tmp[a] = A[f]
        f += 1
    else:
        tmp[a] = A[l]
        l += 1
    a += 1

    if f <= mid:
        tmp[a:] = A[f:mid+1]

    if l <= akhir:
        tmp[a:] = A[l:akhir+1]

    a = 0
    while awal <= akhir:
        A[awal] = tmp[a]
        awal += 1
        a += 1

def mergeSortNew(A):
    mergeSort2(A, 0, len(A)-1)

def quickSortNew(arr):
    kurang = []
    pivotList = []
    lebih = []
    if len(arr) <= 1:
        return arr
    else:
        pivot = arr[0]
        for i in arr:
            if i < pivot:
                kurang.append(i)
            elif i > pivot:
                lebih.append(i)
            else:
                pivotList.append(i)
        kurang = quickSortNew(kurang)
        lebih = quickSortNew(lebih)
        return kurang + pivotList + lebih

```

```

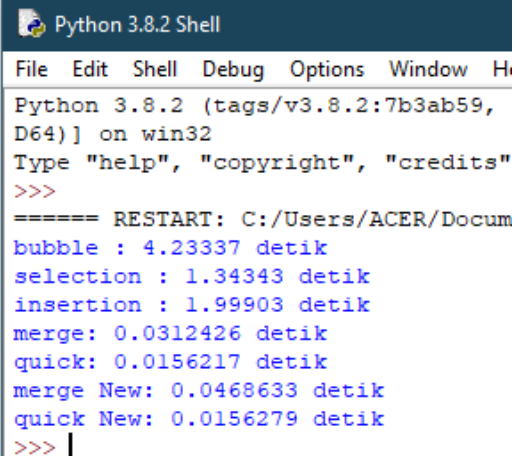
k = list(range(6000))
kocok(k)
u_bub = k[:]
u_sel = k[:]
u_ins = k[:]
u_mrg = k[:]
u_qck = k[:]
u_mrgNew = k[:]
u_qckNew = k[:]

```

```

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));
aw=detak();mergeSortNew(u_mrgNew);ak=detak();print("merge New: %g detik" %(ak-aw));
aw=detak();quickSortNew(u_qckNew);ak=detak();print("quick New: %g detik" %(ak-aw));

```



Python 3.8.2 Shell

| File | Edit | Shell | Debug | Options | Window | Help |
|--|------|-------|-------|---------|--------|------|
| Python 3.8.2 (tags/v3.8.2:7b3ab59, D64) on win32 | | | | | | |
| Type "help", "copyright", "credits" >>> | | | | | | |
| ===== RESTART: C:/Users/ACER/Docum | | | | | | |
| bubble : 4.23337 detik | | | | | | |
| selection : 1.34343 detik | | | | | | |
| insertion : 1.99903 detik | | | | | | |
| merge: 0.0312426 detik | | | | | | |
| quick: 0.0156217 detik | | | | | | |
| merge New: 0.0468633 detik | | | | | | |
| quick New: 0.0156279 detik | | | | | | |
| >>> | | | | | | |

8. No 8


```

d = Node(7)
e = Node(2)
f = Node(4)
g = Node(6)

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:]

        mergeSortLL(separuhkiri)
        mergeSortLL(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

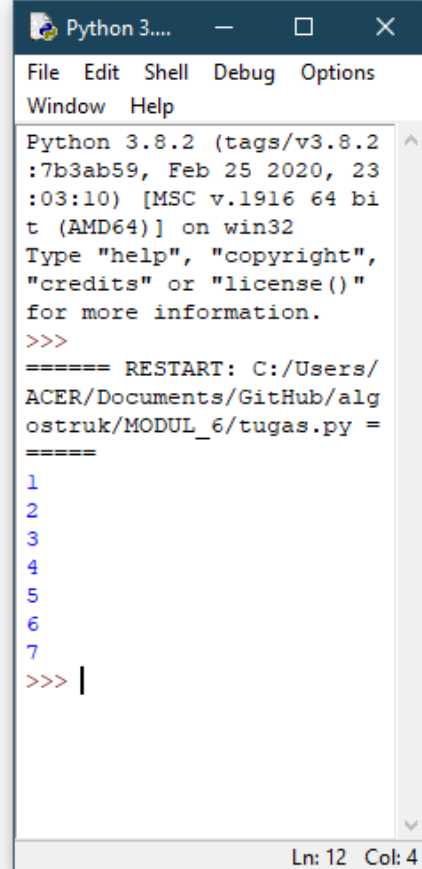
        while i < len(separuhkiri):
            A[k] = separuhkiri[i]
            i = i + 1
            k=k+1

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

    for x in A:
        try:
            linked.data = x
            linked = linked.tautan
        except:
            pass

mergeSortLL(a)

```



```

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/ACER/Documents/GitHub/algostruk/MODUL_6/tugas.py =====
1
2
3
4
5
6
7
>>> |

```

Ln: 12 Col: 4