

**PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**

**MODUL 6**

**PENGURUTAN LANJUTAN**



**Oleh : Fariz Taufiqul Hafidz**

**Nim : L200210192**

**Kelas : F**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS KOMUNIKASI DAN INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**2023**

## A. Soal-soal mahasiswa

1. Ubahlah kode mergeSort dan quickSort di atas agar bisa mengurutkan list yang berisi object-object mhsTIF yang sudah kamu buat di Modul 2.

Uji programmu secukupnya.

- Kode mergeSort() dan quickSort()

```
1 class mhsTIF():
2     def __init__(self,x,y,z,v):
3         self.nama = x
4         self.nim = y
5         self.kotaTinggal = z
6         self.UangSaku = v
7
8 c0 = mhsTIF('Ika',102,'Sukoharjo',240000)
9 c1 = mhsTIF('Budi',104,'Sragen',230000)
10 c2 = mhsTIF('Ahmad',101,'Surakarta',250000)
11 c3 = mhsTIF('Chandra',107,'Surakarta',235000)
12 c4 = mhsTIF('Eka',105,'Boyolali',240000)
13 c5 = mhsTIF('Fandi',109,'Salatiga',250000)
14 c6 = mhsTIF('Deni',106,'Klaten',245000)
15 c7 = mhsTIF('Galuh',108,'Wonogiri',245000)
16 c8 = mhsTIF('Janto',103,'Klaten',245000)
17 c9 = mhsTIF('Hasan',120,'Karanganyar',270000)
18 c10 = mhsTIF('Khalid',114,'Purwodadi',265000)
19
20 daftar = [c0.nim, c1.nim, c2.nim, c3.nim, c4.nim, c5.nim, c6.nim, c7.nim, c8.nim, c9.nim, c10.nim]
21
```

```
1 def mergeSort(A):
2     print("Membelah", A)
3     if len(A) > 1:
4         mid = len(A) // 2
5         separuhKiri = A[:mid]
6         separuhKanan = A[mid:]
7
8         mergeSort(separuhKiri)
9         mergeSort(separuhKanan)
10
11         i = 0
12         j = 0
13         k = 0
14
15         while i < len(separuhKiri) and j < len(separuhKanan):
16             if separuhKiri[i] < separuhKanan[j]:
17                 A[k] = separuhKiri[i]
18                 i = i + 1
19             else:
20                 A[k] = separuhKanan[j]
21                 j = j + 1
22             k = k+1
23
24         while i < len(separuhKiri):
25             A[k] = separuhKiri[i]
26             i = i+1
27             k = k+1
28
29         while j < len(separuhKanan):
30             A[k] = separuhKanan[j]
31             j = j+1
32             k = k+1
33     print('Menggabungkan', A)
```

```

1  def quickSort(A):
2      quickSortBantu(A, 0, len(A) - 1)
3
4  def quickSortBantu(A, awal, akhir):
5      if awal < akhir:
6          titikBelah = partisi(A, awal, akhir)
7          quickSortBantu(A, awal, titikBelah - 1)
8          quickSortBantu(A, titikBelah + 1, akhir)
9
10 def partisi(A, awal, akhir):
11     nilaiPivot = A[awal]
12
13     penandaKiri = awal + 1
14     penandaKanan = akhir
15
16     selesai = False
17     while not selesai:
18
19         while penandaKiri <= penandaKanan and A[penandaKiri] <= nilaiPivot:
20             penandaKiri = penandaKiri + 1
21
22         while A[penandaKanan] >= nilaiPivot and penandaKanan >= penandaKiri:
23             penandaKanan = penandaKanan - 1
24
25         if penandaKanan < penandaKiri:
26             selesai = True
27         else:
28             temp = A[penandaKiri]
29             A[penandaKiri] = A[penandaKanan]
30             A[penandaKanan] = temp
31
32     temp = A[awal]
33     A[awal] = A[penandaKanan]
34     A[penandaKanan] = temp
35
36     return penandaKanan
37
38
39 ## Memanggil fungsi mergeSort()
40 mergeSort(daftar)
41
42 ## Memanggil fungsi quickSort()
43 quickSort(daftar)
44 print("\n"+"Hasil QuickSort")
45 print(daftar)

```

- Output

```

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\Kuliah\Semester 4\Praktikum ASD\Modul 6> & 'C:\Users\Asus\AppData\Local\Micr
nFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '63204' '--' 'E:\Kuliah\S
Membelah [102, 104, 101, 107, 105, 109, 106, 108, 103, 120, 114]
Membelah [102, 104, 101, 107, 105]
Membelah [102, 104]
Membelah [102]
Menggabungkan [102]
Membelah [104]
Menggabungkan [104]
Menggabungkan [102, 104]
Membelah [101, 107, 105]

```

PROBLEMS

OUTPUT

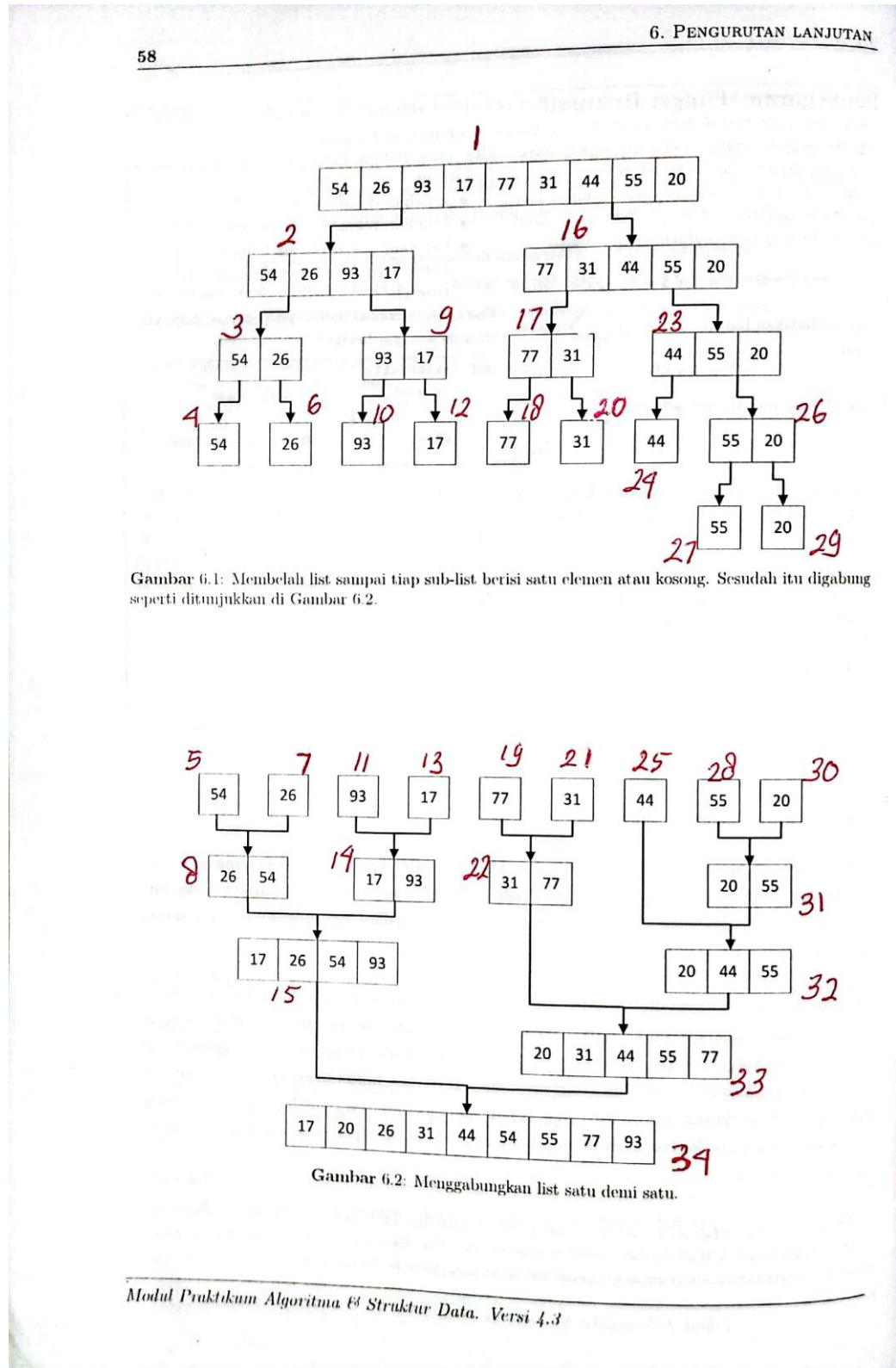
TERMINAL

DEBUG CONSOLE

```
Membelah [101]
Menggabungkan [101]
Membelah [107, 105]
Membelah [107]
Menggabungkan [107]
Membelah [105]
Menggabungkan [105]
Menggabungkan [105, 107]
Menggabungkan [101, 105, 107]
Menggabungkan [101, 102, 104, 105, 107]
Membelah [109, 106, 108, 103, 120, 114]
Membelah [109, 106, 108]
Membelah [109]
Menggabungkan [109]
Membelah [106, 108]
Membelah [106]
Menggabungkan [106]
Membelah [108]
Menggabungkan [108]
Menggabungkan [106, 108]
Menggabungkan [106, 109, 108]
Membelah [103, 120, 114]
Membelah [103]
Menggabungkan [103]
Membelah [120, 114]
Membelah [120]
Menggabungkan [120]
Membelah [114]
Menggabungkan [114]
Menggabungkan [114, 120]
Menggabungkan [103, 114, 120]
Menggabungkan [103, 106, 109, 108, 114, 120]
Menggabungkan [101, 102, 104, 105, 107, 103, 106, 109, 108, 114, 120]

Hasil QuickSort
[101, 102, 103, 104, 105, 106, 107, 108, 109, 114, 120]
PS E:\Kuliah\Semester 4\Praktikum ASD\Modul 6>
```

2. Memakai bolpen merah atau biru, tandai dan beri nomer urut eksekusi proses pada Gambar 6.1 dan 6.2, dengan mengacu pada output di halaman 59.



3. Uji kecepatan. Ujilah mergeSort dan quickSort di atas (bersama metode sort yang kamu pelajari sebelumnya) dengan kode di bawah ini.

- Kode

```
1  from time import time as detik
2  from random import shuffle as kocok
3  import time
4  k = [i for i in range(1,6001)]
5  kocok(k)
6
7  def bubb(arr):
8      n = len(arr)
9      for i in range(n):
10         for j in range(0, n-i-1):
11             if arr[j] > arr[j+1]:
12                 arr[j], arr[j+1] = arr[j+1], arr[j]
13
14  def sele(A):
15      for i in range(len(A)):
16         min_idx = i
17         for j in range(i+1, len(A)):
18             if A[min_idx] > A[j]:
19                 min_idx = j
20         A[i], A[min_idx] = A[min_idx], A[i]
21
22  def inse(arr):
23      for i in range(1, len(arr)):
24         key = arr[i]
25         j = i-1
26         while j >=0 and key < arr[j]:
27             arr[j+1] = arr[j]
28             j -= 1
29         arr[j+1] = key
```

```

1  def mergeSort(arr):
2      if len(arr) > 1:
3          mid = len(arr)//2
4          L = arr[:mid]
5          R = arr[mid:]
6          mergeSort(L)
7          mergeSort(R)
8          i = j = k = 0
9          while i < len(L) and j < len(R):
10             if L[i] < R[j]:
11                 arr[k] = L[i]
12                 i+=1
13             else:
14                 arr[k] = R[j]
15                 j+=1
16             k+=1
17             while i < len(L):
18                 arr[k] = L[i]
19                 i+=1
20                 k+=1
21             while j < len(R):
22                 arr[k] = R[j]
23                 j+=1
24                 k+=1
25
26 def partition(arr,Low,high):
27     i = ( Low-1 )
28     pivot = arr[high]
29     for j in range(Low , high):
30         if arr[j] <= pivot:
31             i = i+1
32             arr[i],arr[j] = arr[j],arr[i]
33     arr[i+1],arr[high] = arr[high],arr[i+1]
34     return ( i+1 )
35
36 def quickSort(arr,Low,high):
37     if Low < high:
38         pi = partition(arr,Low,high)
39         quickSort(arr, Low, pi-1)
40         quickSort(arr, pi+1, high)
41
42
43 bub = k[: ]
44 sel = k[: ]
45 ins = k[: ]
46 mer = k[: ]
47 qui = k[: ]
48
49 aw=detak();bubb(bub);ak=detak();print('bubble : %g detik' %(ak-aw));
50 aw=detak();sele(sel);ak=detak();print('selection : %g detik' %(ak-aw));
51 aw=detak();inse(ins);ak=detak();print('insertion : %g detik' %(ak-aw));
52 aw=detak();mergeSort(mer);ak=detak();print('merge : %g detik' %(ak-aw));
53 aw=detak();quickSort(qui,0,len(qui)-1);ak=detak();print('quick : %g detik' %(ak-aw));

```

- Output

```

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

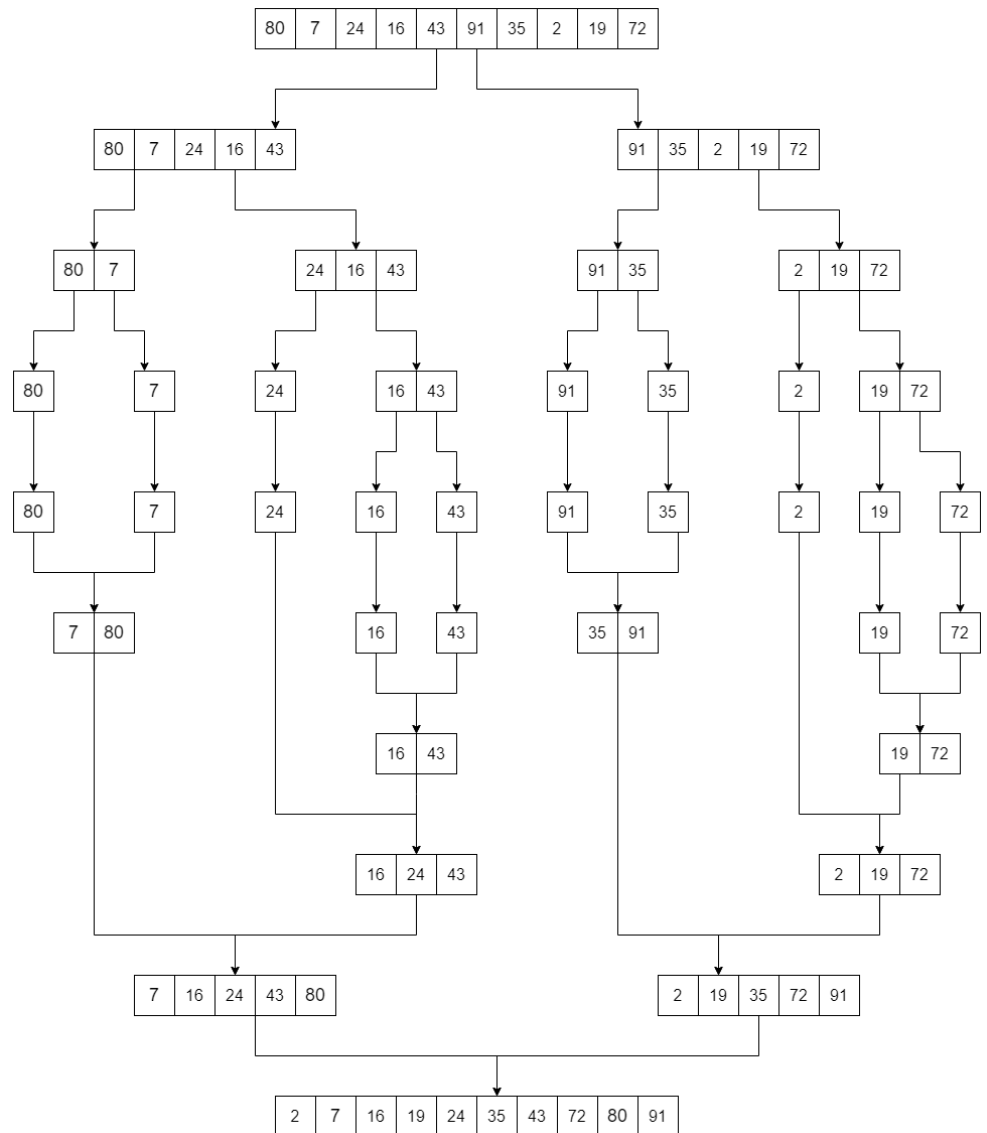
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\Kuliah\Semester 4\Praktikum ASD\Modul 6> & 'C:\Users\Asus\AppData
nFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '63259' '-.'
bubble : 9.21126 detik
selection : 4.33185 detik
insertion : 5.04885 detik
merge : 0.0877986 detik
quick : 0.0568175 detik
PS E:\Kuliah\Semester 4\Praktikum ASD\Modul 6>

```

4. Diberikan list  $L = [80, 7, 24, 16, 43, 91, 35, 2, 19, 72]$ , gambarlah trace pengurutan untuk algoritma
- a. merge sort





b. quick sort

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	80	35	2	19	91
----	---	----	----	----	----	----	---	----	----

Low

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	19	35	43	72	80	91
---	---	----	----	----	----	----	----	----	----

Low High

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