

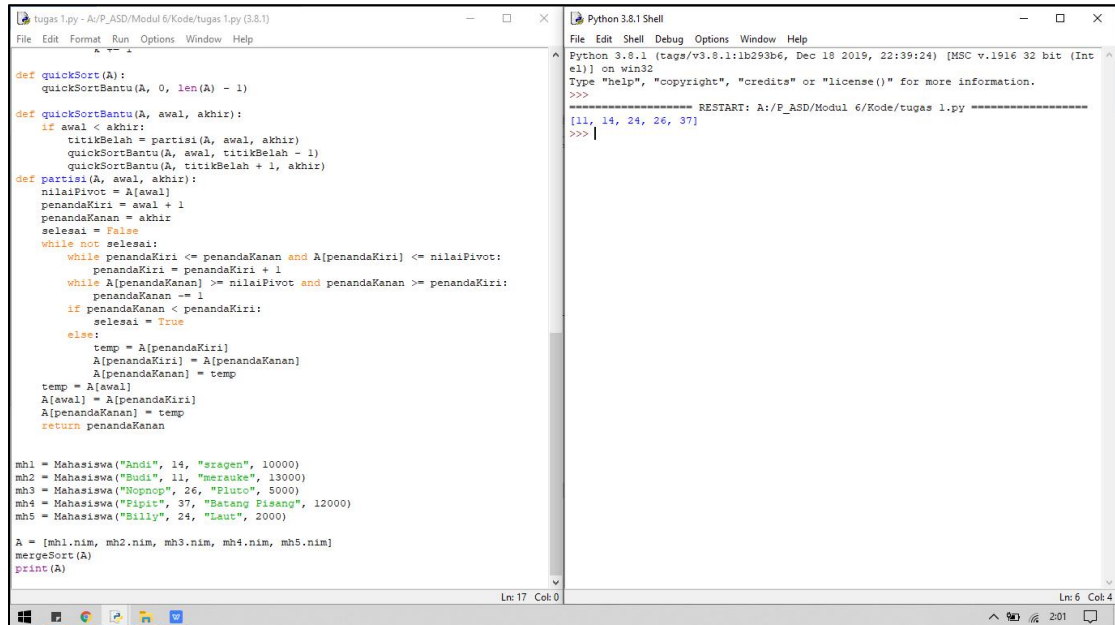
TUGAS PRAKTIKUM

ALGORITMA DAN STRUKTUR DATA

MODUL 6. PENGURUTAN LANJUT

6.4 Soal-soal untuk Mahasiswa

1.



```
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]
    penandaKiri = awal + 1
    penandaKanan = akhir
    selesai = False
    while not selesai:
        while penandaKiri <= penandaKanan and A[penandaKiri] <= nilaiPivot:
            penandaKiri = penandaKiri + 1
        while A[penandaKanan] >= nilaiPivot and penandaKanan >= penandaKiri:
            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[penandaKiri]
            A[penandaKanan] = temp
    return penandaKanan

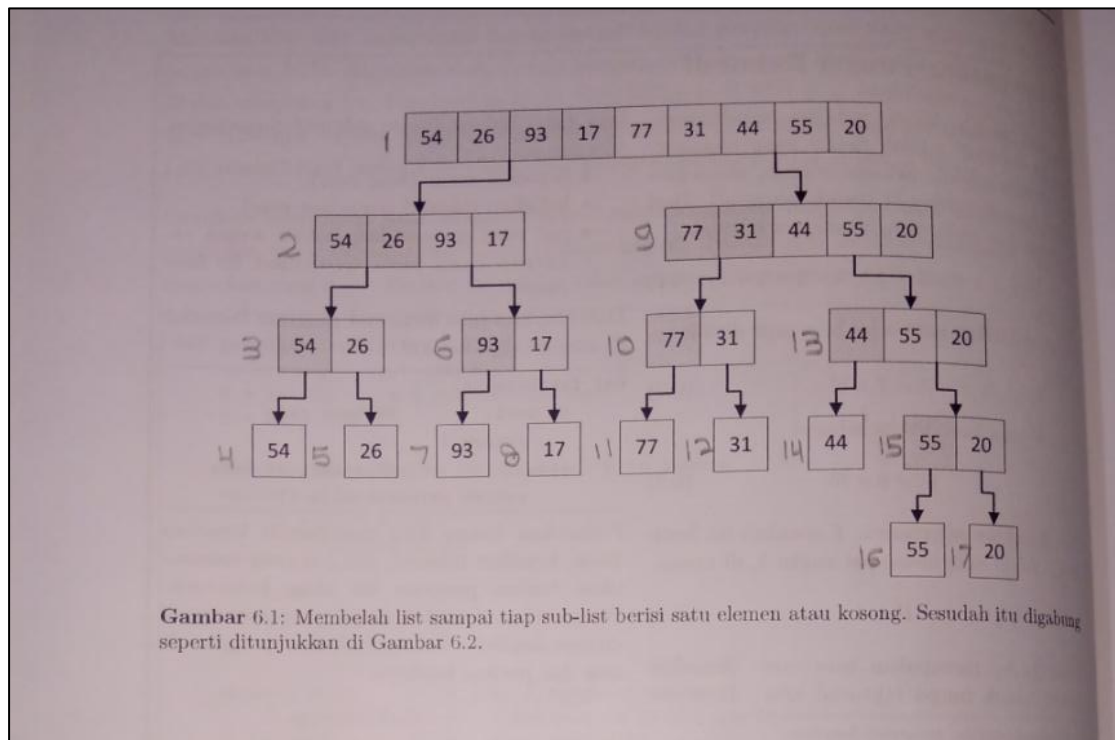
mh1 = Mahasiswa("Andi", 14, "sragen", 10000)
mh2 = Mahasiswa("Budi", 11, "merauke", 13000)
mh3 = Mahasiswa("Hopop", 26, "Pluto", 5000)
mh4 = Mahasiswa("Pipit", 37, "Batang Pissang", 12000)
mh5 = Mahasiswa("Billy", 24, "Laut", 2000)

A = [mh1.nim, mh2.nim, mh3.nim, mh4.nim, mh5.nim]
mergeSort(A)
print(A)
```

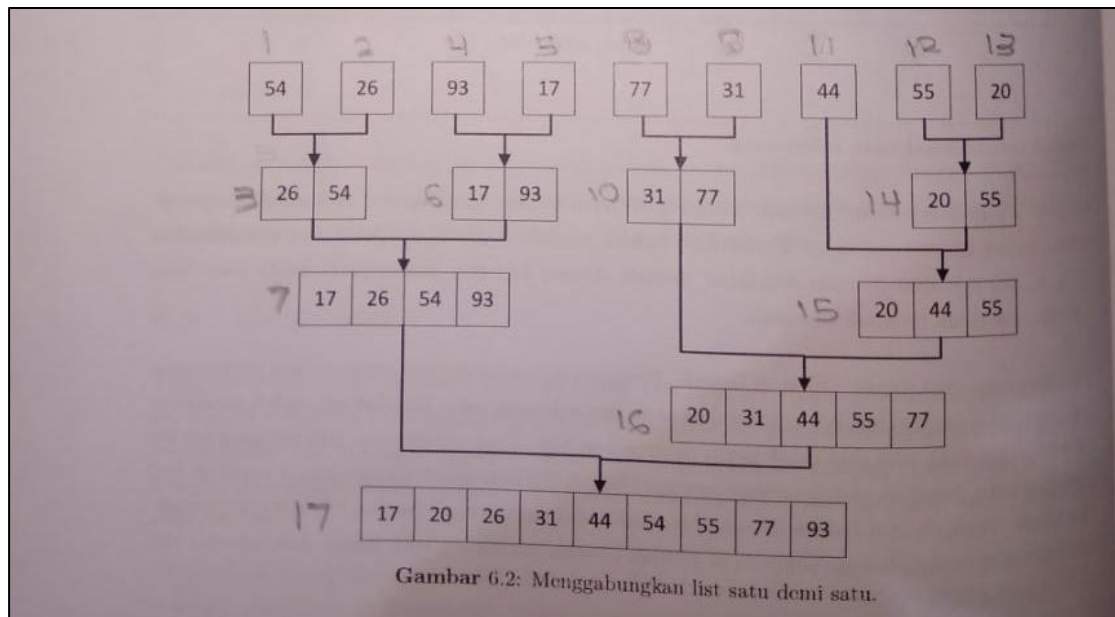
Python 3.8.1 Shell

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: A:/P_ASD/Modul 6/Kode/tugas 1.py =====
>>> [11, 14, 24, 26, 37]
>>>

2.



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



3.

tugas 3.py - A:/P_ASD/Modul 6/Kode/tugas 3.py (3.8.1)
File Edit Format Run Options Window Help

```

A[k] = A[j]
j += 1
while i < len(L):
    A[k] = L[i]
    i += 1
    k += 1
while j < len(R):
    A[k] = R[j]
    j += 1
    k += 1

def partition(A, low, high):
    i = (low - 1)
    pivot = A[high]
    for j in range(low, high):
        if A[j] <= pivot:
            i = i + 1
            A[i], A[j] = A[j], A[i]
    A[i + 1], A[high] = A[high], A[i + 1]
    return i + 1

def quickSortBantu(A, low, high):
    if low < high:
        pi = partition(A, low, high)
        quickSortBantu(A, low, pi - 1)
        quickSortBantu(A, pi + 1, high)

def quickSort(A):
    quickSortBantu(A, 0, len(A)-1)

bub = k[:];
sel = k[:];
ins = k[:];
mer = k[:];
qui = k[:];

aw = detak(); bubbleSort(bub); ak = detak(); print('bubble : %g detik' % (ak-aw))
aw = detak(); selectionSort(sel); ak = detak(); print('selection : %g detik' % (ak-aw))
aw = detak(); insertionSort(ins); ak = detak(); print('insertion : %g detik' % (ak-aw))
aw = detak(); mergeSort(mer); ak = detak(); print('merge : %g detik' % (ak-aw))
aw = detak(); quickSort(qui); ak = detak(); print('quick : %g detik' % (ak-aw))

```

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: A:/P_ASD/Modul 6/Kode/tugas 3.py =====
bubble : 11.6075 detik
selection : 4.71803 detik
insertion : 5.78196 detik
merge : 0.0781145 detik
quick : 0.0311964 detik
>>>

4.

a.

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

b.

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

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	91	35	2	19	72
----	---	----	----	----	----	----	---	----	----

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

5.

```

tugas 5.py - A:\P_ASD\Modul 6\Kode\tugas 5.py (3.8.1)
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

def merge_sort(the_list):
    return _merge_sort((0, len(the_list) - 1), the_list)

print(merge_sort([13, 45, 12, 1, 59]))

```

```

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: A:\P_ASD\Modul 6\Kode\tugas 5.py =====
[1, 12, 13, 45, 59]
>>>

```

6.

The screenshot shows a Python IDE with two windows. The left window, titled 'tugas 6.py', contains the following code:

```
def quicksorthelp(A, low, high):
    result = 0
    if low < high:
        pivot_location, result = Partition(A, low, high)
        result += quicksorthelp(A, low, pivot_location)
        result += quicksorthelp(A, pivot_location + 1, high)
    return result

def Partition(A, low, high):
    result = 0
    pivot, pidx = median_of_three(A, low, high)
    A[low], A[pidx] = A[pidx], A[low]
    i = low + 1
    for j in range(low + 1, high, 1):
        if A[j] < pivot:
            result += 1
            A[i], A[j] = A[j], A[i]
            i += 1
    A[low], A[i - 1] = A[i - 1], A[low]
    return i - 1, result

def median_of_three(A, low, high):
    mid = (low + high - 1) // 2
    a = A[low]
    b = A[mid]
    c = A[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

daftar = [12, 4, 10, 124, 14, 123, 26]
quicksort(daftar)
print(daftar)
```

The right window, titled 'Python 3.8.1 Shell', shows the output of the program:

```
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: A:/P_ASD/Modul 6/Kode/tugas 6.py =====
[4, 10, 12, 14, 26, 123, 124]
>>>
```

7.

The screenshot shows a Python IDE with two windows. The left window, titled 'tugas 7.py', contains the following code:

```
from time import time as detik
from random import shuffle as kocok
import tugas5 # mergeSort baru
import tugas6 # quickSort baru
import tugas3 # mergeSort dan quickSort awal
k = [i for i in range(1, 6000)]
kocok(k)

merA = k[:]
merB = k[:]
quiA = k[:]
quiB = k[:]

# merge Sort baru
aw = detik(); tugas5.merge_sort(merB); ak = detik(); print('merge sort baru : %g detik' % (ak - aw))
# Quick Sort baru
aw = detik(); tugas6.quickSort(quiB); ak = detik(); print('quick sort baru : %g detik' % (ak - aw))

# Merge Sort dan Quick Sort awal
aw = detik(); tugas3.mergeSort(merA); ak = detik(); print('merge sort awal : %g detik' % (ak - aw))
aw = detik(); tugas3.quickSort(quiA); ak = detik(); print('quick sort awal : %g detik' % (ak - aw))
```

The right window, titled 'Python 3.8.1 Shell', shows the output of the program:

```
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: A:/P_ASD/Modul 6/Kode/tugas 7.py =====
[1, 12, 13, 45, 59]
[4, 10, 12, 14, 26, 123, 124]
bubble : 12.4536 detik
selection : 4.80893 detik
insertion : 5.96636 detik
merge : 0.0624895 detik
quick : 0.0468156 detik
merge sort baru : 0.0937345 detik
quick sort baru : 0.0624204 detik
merge sort awal : 0.0760456 detik
quick sort awal : 0.0312173 detik
>>>
```

8.

```

tugas 8.py - A:/P_ASD/Modul 6/Kode/tugas 8.py (3.8.1)
File Edit Format Run Options Window Help
while curr != None:
    print("%d" % curr.data),
    curr = curr.next

def mergeSorted(self, list1, list2):
    if list1 is None:
        return list2
    if list2 is None:
        return list1

    if list1.data < list2.data:
        temp = list1
        temp.next = self.mergeSorted(list1.next, list2)
    else:
        temp = list2
        temp.next = self.mergeSorted(list1, list2.next)
    return temp

list1 = LinkedList()
list1.appendSorted(13)
list1.appendSorted(12)
list1.appendSorted(3)
list1.appendSorted(14)
list1.appendSorted(7)

print("List 1 :"),
list1.printList()

list2 = LinkedList()
list2.appendSorted(26)
list2.appendSorted(10)
list2.appendSorted(1)

print("List 2 :"),
list2.printList()

list3 = LinkedList()
list3.head = list3.mergeSorted(list1.head, list2.head)

print("Merged List :"),
list3.printList()

```

```

Python 3.8.1 Shell
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Int
el)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: A:/P_ASD/Modul 6/Kode/tugas 8.py =====
List 1 :
3
7
12
13
14
List 2 :
1
10
26
Merged List :
1
3
7
10
12
13
14
26
>>>

```