Nama : Sukma Nindi Lisyarini

NIM : L200170147

Kelas : D

Laporan Praktikum

Algoritma dan Struktur Data

Modul 6

Nomer 1

```
File  Edit  Format  Run  Options  Window  Help
class Mahasiswa(object):
    def __init__(self, nama, NIM, kota, us):
        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kota
        self.uangSaku = us

a0 = Mahasiswa('Sukma', 147, 'Yogyakarta', 500000)
a1 = Mahasiswa('Nindi', 100, 'Sragen', 700000)
a2 = Mahasiswa('Cantik', 200, 'Surakarta', 250000)
a3 = Mahasiswa('Nurma', 180, 'Surakarta', 500000)
a4 = Mahasiswa('Fitra', 112, 'Boyolali', 600000)
a5 = Mahasiswa('Retno', 110, 'Salatiga', 250000)
a6 = Mahasiswa('Muhibah', 131, 'Klaten', 500000)
a7 = Mahasiswa('Zahra', 201, 'Wonogiri', 245000)
a8 = Mahasiswa('Rizka', 231, 'Klaten', 400000)
a9 = Mahasiswa('Iis', 143, 'Karanganyar', 700000)
a10 = Mahasiswa('Ayasha', 129, 'Purwodadi', 100000)

Daftar = [a0, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10]

def urutkanNIM(a):
    baru = {}
    for i in range(len(a)):
        baru[a[i].nama] = a[i].NIM
    listofTuples = sorted(baru.items(), key = lambda x: x[1])
    for elem in listofTuples:
        print (elem[0], ':', elem[1])
urutkanNIM(Daftar)
```

```
File  Edit  Shell  Debug  Options  Window  Help
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 20
17, 20:19:30) [MSC v.1500 32 bit (Intel)] on
win32
Type "copyright", "credits" or "license()" f
or more information.
>>>
 RESTART: C:\Users\TOSHIBA\Downloads\BEM KAB
INET JUANG\May Day\Modul6_D_147\Nomor 1.py
('Nindi', ':', 100)
('Retno', ':', 110)
('Fitra', ':', 112)
('Ayasha', ':', 129)
('Muhibah', ':', 131)
('Iis', ':', 143)
('Sukma', ':', 147)
('Nurma', ':', 180)
('Cantik', ':', 200)
('Zahra', ':', 201)
('Rizka', ':', 231)
>>>
```

Nomer 2

```
File  Edit  Format  Run  Options  Window  Help
def bubblesort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
    return arr

def gabung(a,b):
    c = []
    c = a+b
    n = len(c)
    for i in range(n):
        for j in range(0, n-i-1):
            if c[j] > c[j+1]:
                c[j], c[j+1] = c[j+1], c[j]
    return c

a = [5,45,12,32,6,10,2]
b = [26,8,20,14,40]
a,b = bubblesort(a),bubblesort(b)

print(a,'\n',b)
print()
print(gabung(a,b))
```

```
File  Edit  Shell  Debug  Options  Window  Help
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017,
20:19:30) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for
more information.
>>>
 RESTART: C:\Users\TOSHIBA\Downloads\BEM KABINE
T JUANG\May Day\Modul6_D_147\Nomor 2.py
([2, 5, 6, 10, 12, 32, 45], '\n', [8, 14, 20, 2
6, 40])
()
[2, 5, 6, 8, 10, 12, 14, 20, 26, 32, 40, 45]
>>>
```

Nomer 3 dan Nomer 4

```
File  Edit  Format  Run  Options  Window  Help
from time import time as detak
from random import shuffle as kocok
import time

k = [i for i in range(1,6001)]
kocok(k)

def bubb(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1] :
                arr[j], arr[j+1] = arr[j+1], arr[j]

def sele(A):
    for i in range(len(A)):
        min_idx = i
        for j in range(i+1, len(A)):
            if A[min_idx] > A[j]:
                min_idx = j
        A[i], A[min_idx] = A[min_idx], A[i]

def inse(arr):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i-1
        while j >=0 and key < arr[j] :
                arr[j+1] = arr[j]
                j -= 1
        arr[j+1] = key
def mergeSort(arr):
    if len(arr) >1:
        mid = len(arr)//2
        L = arr[:mid]
        R = arr[mid:]
        mergeSort(L)
        mergeSort(R)
        i = j = k = 0
        while i < len(L) and j < len(R):
```

```
File  Edit  Shell  Debug  Options  Window  Help
Python 2.7.14 (v2.7.14:84471935ed, Sep
tel)] on win32
Type "copyright", "credits" or "licens
>>>
 RESTART: C:\Users\TOSHIBA\Downloads\B
r 3 dan 4.py
bubble : 9.971 detik
selection : 3.384 detik
insertion : 4.364 detik
merge : 0.099 detik
quick : 0.059 detik
>>>
```

Nomer 5

```
File  Edit  Format  Run  Options  Window  Help
import random

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 <= en
        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
```

```
File  Edit  Shell  Debug  Options  Window  Help
Python 2.7.14 (v2.7.14:84471935ed, Sep
20:19:30) [MSC v.1500 32 bit (Intel)]
Type "copyright", "credits" or "licens
ore information.
>>>
 RESTART: C:\Users\TOSHIBA\Downloads\B
JUANG\May Day\Modul6_D_147\Nomor 5.py
[12, 13, 45]
>>>
```

Nomer 6

```
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
```

```
Python 2.7.14 (v2.7.14:84471935ed, Sep 1
6 2017, 20:19:30) [MSC v.1500 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license(
)" for more information.
>>>
 RESTART: C:\Users\TOSHIBA\Downloads\BEM
 KABINET JUANG\May Day\Modul6_D_147\Nomo
r 6.py
('sorted :', [124, 123, 15, 12, 4])
>>>
```

Nomer 7

```
File  Edit  Format  Run  Options  Window  Help
from time import time as detak
from random import shuffle as kocok
import time

k = [i for i in range(1,6001)]
kocok(k)

def mergeSort(arr):
    if len(arr) >1:
        mid = len(arr)//2
        L = arr[:mid]
        R = arr[mid:]
        mergeSort(L)
        mergeSort(R)
        i = j = k = 0
        while i < len(L) and j < len(R):
            if L[i] < R[j]:
                arr[k] = L[i]
                i+=1
            else:
                arr[k] = R[j]
                j+=1
            k+=1
        while i < len(L):
            arr[k] = L[i]
            i+=1
            k+=1
        while j < len(R):
            arr[k] = R[j]
            j+=1
            k+=1

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

```
File  Edit  Shell  Debug  Options  Window  Help
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017,
20:19:30) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for m
ore information.
>>>
 RESTART: C:\Users\TOSHIBA\Downloads\BEM KABINET
JUANG\May Day\Modul6_D_147\Nomor 7.py
merge : 0.115 detik
quick : 0.0579998 detik
merge mod : -0.062 detik
quick mod : -0.184 detik
>>>
```

Nomer 8

File Edit Format Run Options Window Help

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

class LinkedList:
  def __init__(self):
    self.head = None

  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

  def printList(self):
    curr = self.head
    while curr != None:
      print ("%d"%curr.data),
```

File Edit Shell Debug Options Window Help

```
Python 2.7.14 (v2.7.14:84471935ed, Se
p 16 2017, 20:19:30) [MSC v.1500 32 b
it (Intel)] on win32
Type "copyright", "credits" or "licen
se()" for more information.
>>>
 RESTART: C:\Users\TOSHIBA\Downloads\
BEM KABINET JUANG\May Day\Modul6_D_14
7\Nomor 8.py
List 1 : 3 7 12 13 16 List 2 : 1 9 10
Merged List : 1 3 7 9 10 12 13 16
>>>
```