Nama : Hafidh Putra Andhika

NIM : L200180085

Kelas : D

No 1

TugasModulke6.py - F:/Tugas/UMS/Semester 4/Praktikum Algostruk/Modul 6/TugasModulk...

File Edit Format Run Options Window Help

```
print("=====================================")
print("Nomor 1")
print("=====================================")
class MhsTIF(object):
    def __init__(self,nama,nim,kota,uangsaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangsaku

c0 = MhsTIF('Andhika', 85, 'Solo', 35000)
c1 = MhsTIF('Hafidh', 91, 'Salatiga', 30000)
c2 = MhsTIF('Putra', 100, 'Surakarta', 13000)
c3 = MhsTIF('Hesti', 119, 'Solo', 14000)
c4 = MhsTIF('Retno', 74, 'Boyolali', 15000)
c5 = MhsTIF('Sari', 23, 'Boyolali', 16000)
c6 = MhsTIF('Kesya', 113, 'Klaten', 37000)
c7 = MhsTIF('Diwa', 95, 'Wonogiri', 18000)
c8 = MhsTIF('Ariela', 88, 'Karanganyar', 29000)
c9 = MhsTIF('inez', 114, 'Surakart', 20000)
c10 = MhsTIF('Johan', 27, 'Purwodadi', 21000)

Daftar=[c0,c1,c2,c3,c4,c5,c6,c7,c8,c9,c10]

def cek(Daftar):
    for i in Daftar:
        print(i.nama,i.nim,i.kotaTinggal)

################################nomor 1##############################################
    #mergesort
def mergesort(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].nim < separuhkanan[j].nim :
                A[k] = separuhkiri[i]
                i = i+1
            else :
                A[k] = separuhkanan[j]
                j = j+1
```

Ln: 214 Col: 33

TugasModulke6.py - F:/Tugas/UMS/Semester 4/Praktikum Algostruk/Modul 6/TugasModulk...

File Edit Format Run Options Window Help

```
            k = k+1

    #quicksort
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].nim
    penandakiri = awal + 1
    penandakanan = akhir
    selesai = False

    while not selesai:
        while penandakiri <= penandakanan and A[penandakiri].nim <= nilaipivot:
            penandakiri +=1
        while A[penandakanan].nim >= nilaipivot and penandakanan >= penandakiri
            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

cek(Daftar)
print("=====================================")
print("MergeSort")
print("=====================================")
mergesort(Daftar)
cek(Daftar)
print("=====================================")
print("QuickSort")
quicksort(Daftar)
print("=====================================")
cek(Daftar)
```

Ln: 214 Col: 33

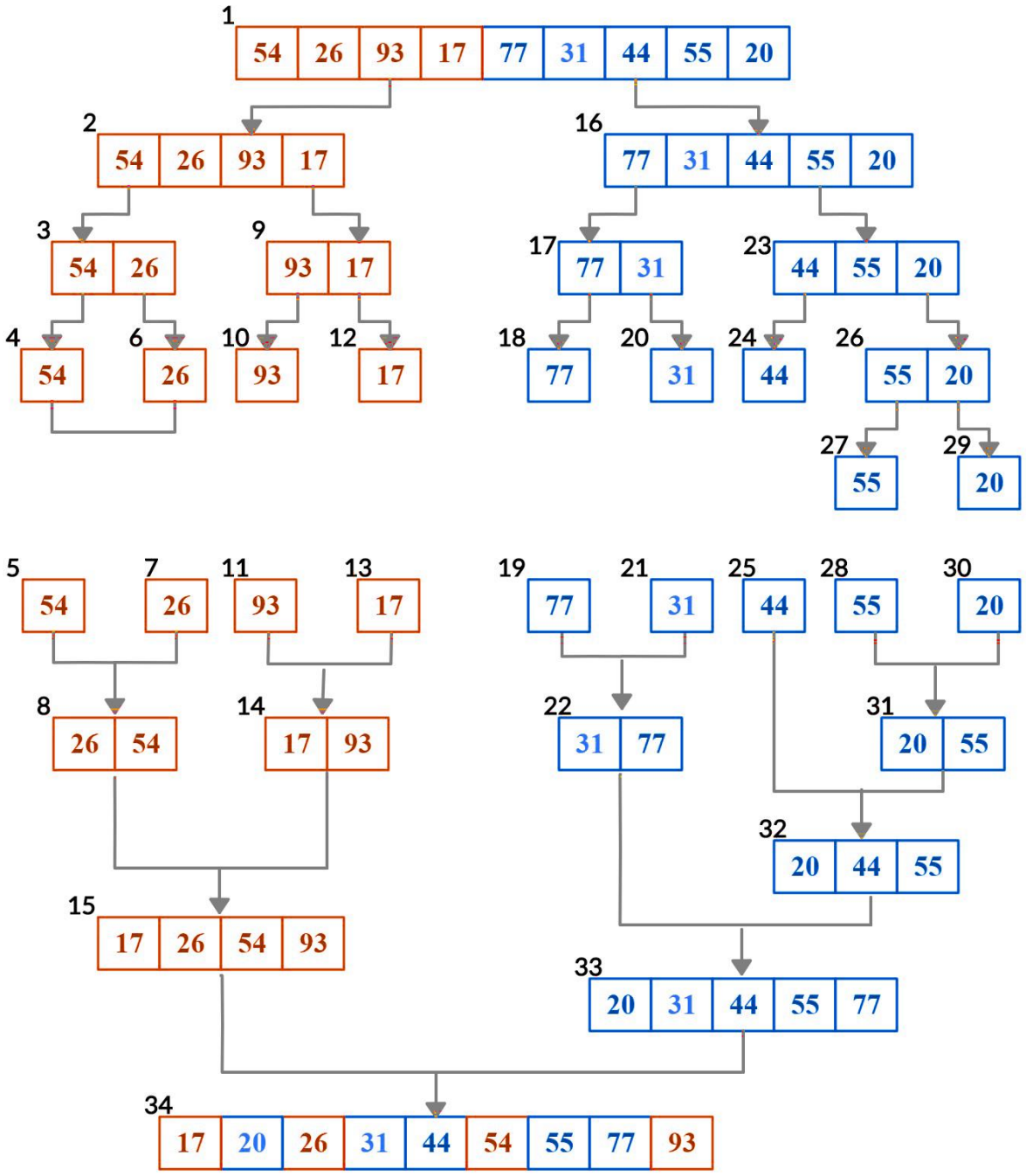
Python 3.8.2 Shell

File Edit Shell Debug Options Window Help

```
=====================================
Nomor 1
=====================================
Andhika 85 Solo
Hafidh 91 Salatiga
Putra 100 Surakarta
Hesti 119 Solo
Retno 74 Boyolali
Sari 23 Boyolali
Kesya 113 Klaten
Diwa 95 Wonogiri
Ariela 88 Karanganyar
inez 114 Surakart
Johan 27 Purwodadi
=====================================
MergeSort
=====================================
Sari 23 Boyolali
Johan 27 Purwodadi
Retno 74 Boyolali
Andhika 85 Solo
Ariela 88 Karanganyar
Hafidh 91 Salatiga
Diwa 95 Wonogiri
Putra 100 Surakarta
Kesya 113 Klaten
inez 114 Surakart
Hesti 119 Solo
=====================================
QuickSort
=====================================
Sari 23 Boyolali
Johan 27 Purwodadi
Retno 74 Boyolali
Andhika 85 Solo
Ariela 88 Karanganyar
Hafidh 91 Salatiga
Diwa 95 Wonogiri
Putra 100 Surakarta
Kesya 113 Klaten
inez 114 Surakart
Hesti 119 Solo
=====================================
```

Ln: 103 Col: 4

No 2


1
54 26 93 17 77 31 44 55 20
2
54 26 93 17
16
77 31 44 55 20
3
54 26
9
93 17
17
77 31
23
44 55 20
4
54
6
26
10
93
12
17
18
77
20
31
24
44
26
55 20
27
55
29
20
5
54
7
26
11
93
13
17
19
77
21
31
25
44
28
55
30
20
8
26 54
14
17 93
22
31 77
31
20 55
32
20 44 55
15
17 26 54 93
33
20 31 44 55 77
34
17 20 26 31 44 54 55 77 93

No 3

TugasModulke6.py - F:/Tugas/UMS/Semester 4/Praktikum Algostruk/Modul 6/TugasModulk...

File Edit Format Run Options Window Help

```python
print("=====================================")
print("Nomor 3")
print("=====================================")
from time import time as detak
from random import shuffle as kocok
import time

def swap(A, p, q):
    tmp = A[p]
    A[p] = A[q]
    A[q] = tmp

def cariPosisiYangTerkecil(A, dariSini, sampaiSini):
    posisiYangTerkecil = dariSini
    for i in range(dariSini+1, sampaiSini):
        if A[i] < A[posisiYangTerkecil]:
            posisiYangTerkecil = i
    return posisiYangTerkecil

def bubbleSort(S):
    n = len(S)
    for i in range (n-1):
        for j in range (n-i-1):
            if S[j] > S[j+1]:
                swap(S,j,j+1)
    return S

def selectionSort(S):
    n = len(S)
    for i in range(n-1):
        indexKecil = cariPosisiYangTerkecil(S, i, n)
        if indexKecil != i:
            swap(S, i, indexKecil)
    return S

def insertionSort(S):
    n = len(S)
    for i in range(1, n):
        nilai = S[i]
        pos = i
        while pos > 0 and nilai < S[pos -1]:
            S[pos] = S[pos-1]
            pos = pos - 1
        S[pos] = nilai
    return S
```

Ln: 214 Col: 33

TugasModulke6.py - F:/Tugas/UMS/Semester 4/Praktikum Algostruk/Modul 6/TugasModulk...

File Edit Format Run Options Window Help

```
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 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:
```

Ln: 214 Col: 33

TugasModulke6.py - F:/Tugas/UMS/Semester 4/Praktikum Algostruk/Modul 6/TugasModulk...

File Edit Format Run Options Window Help

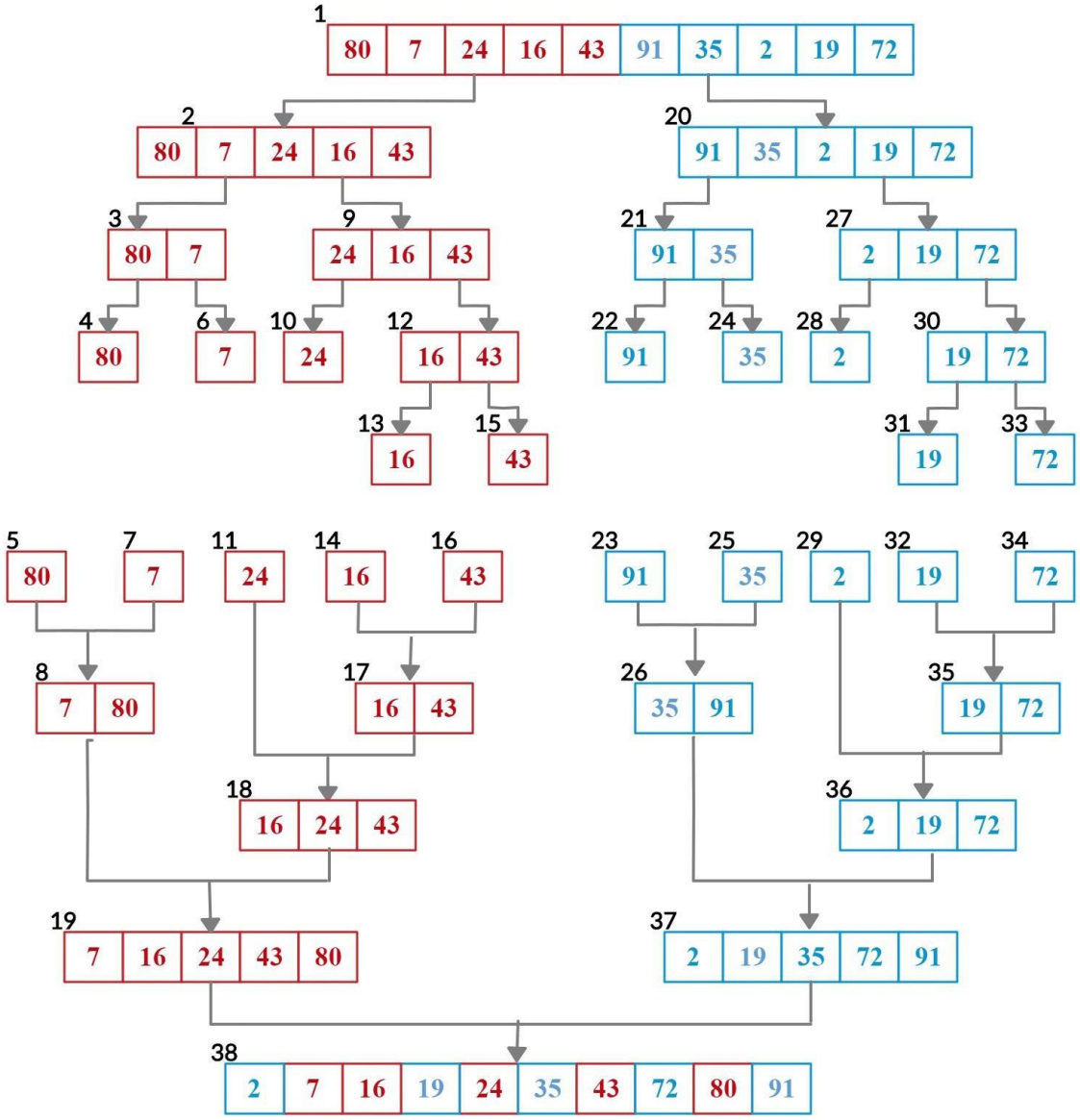
```
        if penandakanan < penandakiri:
            selesai = True
        else:
            temp = A[penandakiri]
            A[penandakiri] = A[penandakanan]
            A[penandakanan] = temp

    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]

print (bubbleSort(daftar))
print (selectionSort(daftar))
print (insertionSort(daftar))
mergeSort(daftar)
print (daftar)
quickSort(daftar)
print (daftar)

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));
```

Ln: 214 Col: 33

```
Python 3.8.2 Shell
File  Edit  Shell  Debug  Options  Window  Help
Nomor 3
=====================================
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
bubble: 7.33182 detik
selection: 3.23014 detik
insertion: 3.47302 detik
merge: 0.0429645 detik
quick: 0.0279849 detik
=====================================
                                        Ln: 103  Col: 4
```

No 4

a. Merge Sort




b. Quick Sort

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

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 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 | 80 |
| | | | | | 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 | |

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 | 43 | 19 | 35 | 72 | 80 | 91 |
| Low | | | | | | High | | | |

pivot

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 7 | 24 | 16 | 43 | 19 | 35 | 72 | 80 | 91 |
| | Low | | | | | High | | | |

pivot

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 7 | 24 | 16 | 43 | 19 | 35 | 72 | 80 | 91 |
| | | Low | | | | High | | | |

pivot

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 7 | 24 | 16 | 43 | 19 | 35 | 72 | 80 | 91 |
| | | Low | | | High | | | | |

pivot

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 7 | 19 | 16 | 43 | 24 | 35 | 72 | 80 | 91 |
| | | Low | | | High | | | | |

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

TugasModulke6.py - F:/Tugas/UMS/Semester 4/Praktikum Algostruk/Modul 6/TugasModulk...

File Edit Format Run Options Window Help

```
##############################nomor 5##########################################

print("=====================================")
print("Nomor 5")
print("=====================================")
daftar = [54,26,93,17,77,31,44,55,20]
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] < A[l]:
            tmp[a] = A[f]
            f += 1
        else:
            tmp[a] = A[l]
            l += 1
        a += 1
#proses penggabungan
    if f <= mid:
        tmp[a:] = A[f:mid+1]
    if l <= akhir:
        tmp[a:] = A[l:akhir+1]
#memindah isi tmp ke A
    a = 0
    while awal <= akhir:
        A[awal] = tmp[a]
        awal += 1
        a += 1

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

print("sebelum","\n",daftar)
mergeSort(daftar)
print("sesudah","\n",daftar)
```

Ln: 214 Col: 33

Python 3.8.2 Shell

File Edit Shell Debug Options Window Help

```
Nomor 5
=====================================
sebelum
 [54, 26, 93, 17, 77, 31, 44, 55, 20]
sesudah
 [17, 20, 26, 31, 44, 54, 55, 77, 93]
```

Ln: 103 Col: 4

No 6

TugasModulke6.py - F:/Tugas/UMS/Semester 4/Praktikum Algostruk/Modul 6/TugasModulk...

File Edit Format Run Options Window Help

```
##############################nomor 6#########################################

print("=====================================")
print("Nomor 6")
print("=====================================")
daftar = [54,26,93,17,77,31,44,55,20]
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:
```

Ln: 214 Col: 33

TugasModulke6.py - F:/Tugas/UMS/Semester 4/Praktikum Algostruk/Modul 6/TugasModulk...

File Edit Format Run Options Window Help

```python
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

print("sebelum","\n",daftar)
quickSort(daftar)
print("sesudah","\n",daftar)
```

Ln: 214 Col: 33

Python 3.8.2 Shell

File Edit Shell Debug Options Window Help

```
Nomor 6
=====================================
sebelum
 [54, 26, 93, 17, 77, 31, 44, 55, 20]
sesudah
 [17, 20, 26, 31, 44, 54, 55, 77, 93]
```

Ln: 103 Col: 4

No 7

TugasModulke6.py - F:/Tugas/UMS/Semester 4/Praktikum Algostruk/Modul 6/TugasModulk...

File Edit Format Run Options Window Help

```
print("=====================================")
print("Nomor 7")
print("=====================================")
def mergesort(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+=1
            else:
                A[k] = separuhkanan[j]
                j+=1
            k+=1
        while i < len(separuhkiri):
            A[k] = separuhkiri[i]
            i+=1
            k+=1
        while j< len(separuhkanan):
            A[k] = separuhkanan[j]
            j+=1
            k+=1
alist = [54,26,93,17,77,31,44,55,20]

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 +=1
        while A[penandakanan] >= nilaipivot and penandakanan >= penandakiri :
            penandakanan -=1
        if penandakanan < penandakiri:
```

Ln: 214 Col: 33

TugasModulke6.py - F:/Tugas/UMS/Semester 4/Praktikum Algostruk/Modul 6/TugasModulk...

File Edit Format Run Options Window Help

```
    while not selesai:
        while penandakiri <= penandakanan and A[penandakiri] <= nilaipivot:
            penandakiri +=1
        while A[penandakanan] >= nilaipivot and penandakanan >= penandakiri :
            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)


#merge sort terbaru
def mergesort2_5(A, awal, akhir):
    mid = (awal+akhir)//2
    if awal < akhir:
        mergesort2_5(A, awal, mid)
        mergesort2_5(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] < A[l]:
            tmp[a] = A[f]
            f += 1
        else:
```

Ln: 214 Col: 33

TugasModulke6.py - F:/Tugas/UMS/Semester 4/Praktikum Algostruk/Modul 6/TugasModulk...

File Edit Format Run Options Window Help

```
#merge sort terbaru
def mergesort2_5(A, awal, akhir):
    mid = (awal+akhir)//2
    if awal < akhir:
        mergesort2_5(A, awal, mid)
        mergesort2_5(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] < A[l]:
            tmp[a] = A[f]
            f += 1
        else:
            tmp[a] = A[l]
            l += 1
        a += 1
#proses penggabungan
    if f <= mid:
        tmp[a:] = A[f:mid+1]
    if l <= akhir:
        tmp[a:] = A[l:akhir+1]
#memindah isi tmp ke A
    a = 0
    while awal <= akhir:
        A[awal] = tmp[a]
        awal += 1
        a += 1

def mergesort_5(A):
    mergesort2_5(A, 0, len(A)-1)

#quick sort terbaru
def quicksort_6(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)
```

Ln: 214 Col: 33

TugasModulke6.py - F:/Tugas/UMS/Semester 4/Praktikum Algostruk/Modul 6/TugasModulk...

File Edit Format Run Options Window Help

```
    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

daftar = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]
from time import time as detak
from random import shuffle as kocok
import time

k =  [[i] for i in range(1, 6001)]
kocok(k)
u_mer = k[:]
u_mer5 = k[:]
u_qui = k[:]
u_qui6 = k[:]

aw=detak();mergesort(u_mer);ak=detak();print("mergesort            : %g detik" %(
aw=detak();mergesort_5(u_mer5);ak=detak();print("mergesort terbaru  : %g detik"
aw=detak();quicksort(u_qui);ak=detak();print("quicksort            : %g detik" %(
aw=detak();quicksort_6(u_qui6);ak=detak();print("quicksort terbaru  : %g detik"
```

Ln: 214 Col: 33

Python 3.8.2 Shell

File Edit Shell Debug Options Window Help

```
=====================================
Nomor 7
=====================================
mergesort            : 0.045974 detik
mergesort terbaru  : 0.0729561 detik
quicksort            : 0.0469778 detik
quicksort terbaru  : 0.0379901 detik
```

Ln: 103 Col: 4

No 8

TugasModulke6.py - F:/Tugas/UMS/Semester 4/Praktikum Algostruk/Modul 6/TugasModulk...

File Edit Format Run Options Window Help

```python
################################nomor 8##########################################

print("=====================================")
print("Nomor 8")
print("=====================================")
class Node():
    def __init__(self,data,next= None,prev = None):
        self.data = data
        self.next = next
        self.prev = prev

class Linked():
    def __init__(self,head = None):
        self.head = head

    def cetak(self):
        cur = self.head
        while cur != None:
            print(cur.data)
            cur = cur.next
    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):
```

Ln: 214 Col: 33

TugasModulke6.py - F:/Tugas/UMS/Semester 4/Praktikum Algostruk/Modul 6/TugasModulk...

File Edit Format Run Options Window Help

```
        node.next = curr

    def printList(self):
        curr = self.head
        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 = Linked()
list1.appendSorted(48)
list1.appendSorted(92)
list1.appendSorted(33)
list1.appendSorted(16)
list1.appendSorted(17)

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

list2 = Linked()
list2.appendSorted(23)
list2.appendSorted(10)
list2.appendSorted(18)

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

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

print("Mergesort Linked list :"),
list3.printList()
```

Ln: 214 Col: 33

Python 3.8.2 Shell

File Edit Shell Debug Options Window Help

```
Nomor 8
=====================================
List 1 :
16
17
33
48
92
List 2 :
10
18
23
Mergesort Linked list :
10
16
17
18
23
33
48
92
>>>
```

Ln: 103 Col: 4