

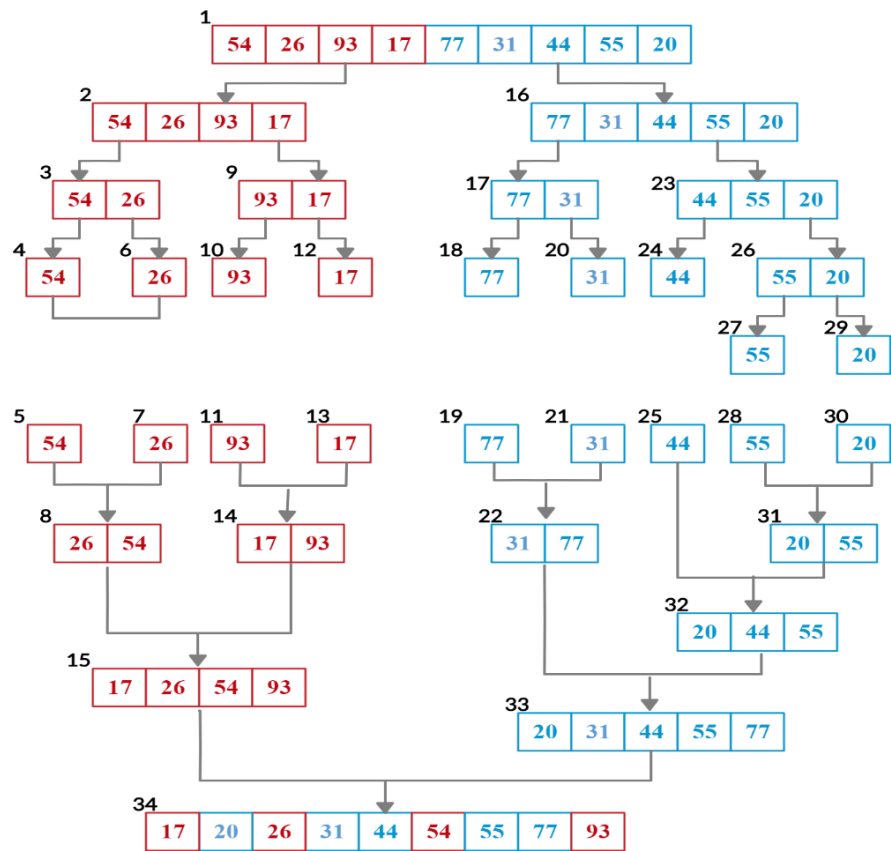
Nama : Berlian Vidia Puspa
NIM : L200180107
Kelas : D

Modul 6 Pengurutan Lanjutan

1. Ubahlah kode mergeSort dan quickSort diatas agar bisa mengurutkan list yang berisi object-object mhsTIF yang sudah kamu buat di modul 2. Uji programmu secukupnya.

<pre>satu.py - C:\Users\ACER\Downloads\URUSAN BERLIN BUKAN I File Edit Format Run Options Window Help def cek(Daftar): for i in Daftar: print(i.nama,i.no,i.kota) #nomer 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].no < separuhkanan[j].no: 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 #quicksort def quicksort(A): quicksortbantu(A,0,len(A)-1) def quicksortbantu(A,awal,akhir):</pre>	<pre>Python 3.7.4 Shell File Edit Shell Debug Options Window Help Ayudhia 95 Surakarta Wulan 91 Kartasura Nayu 99 Pangkalan Bun Tata 61 Pati Irul 101 Riau Caca 97 Banten Diah 106 Sorong Anggit 111 NTT Amron 105 Kudus ===== mergesortnya ===== Tata 61 Pati Wulan 91 Kartasura Ayudhia 95 Surakarta Caca 97 Banten Nayu 99 Pangkalan Bun Irul 101 Riau Amron 105 Kudus Diah 106 Sorong Berlian 107 Pati Elsa 108 Mojolaban Anggit 111 NTT ===== quicksortnya ===== Tata 61 Pati Wulan 91 Kartasura Ayudhia 95 Surakarta Caca 97 Banten Nayu 99 Pangkalan Bun Irul 101 Riau Amron 105 Kudus Diah 106 Sorong Berlian 107 Pati Elsa 108 Mojolaban Anggit 111 NTT</pre>
---	---

2. Memakai bolpoin 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 diatas (bersama metode sort yang kamu pelajari sebelumnya) dengan kode dibawah ini.

```
tiga.py - C:\Users\ACER\Downloads\URUSAN BERLIN BUKAN URUSANMU\berlin\P.ALGOSTR...
File Edit Format Run Options Window Help

from time import time as detik
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]

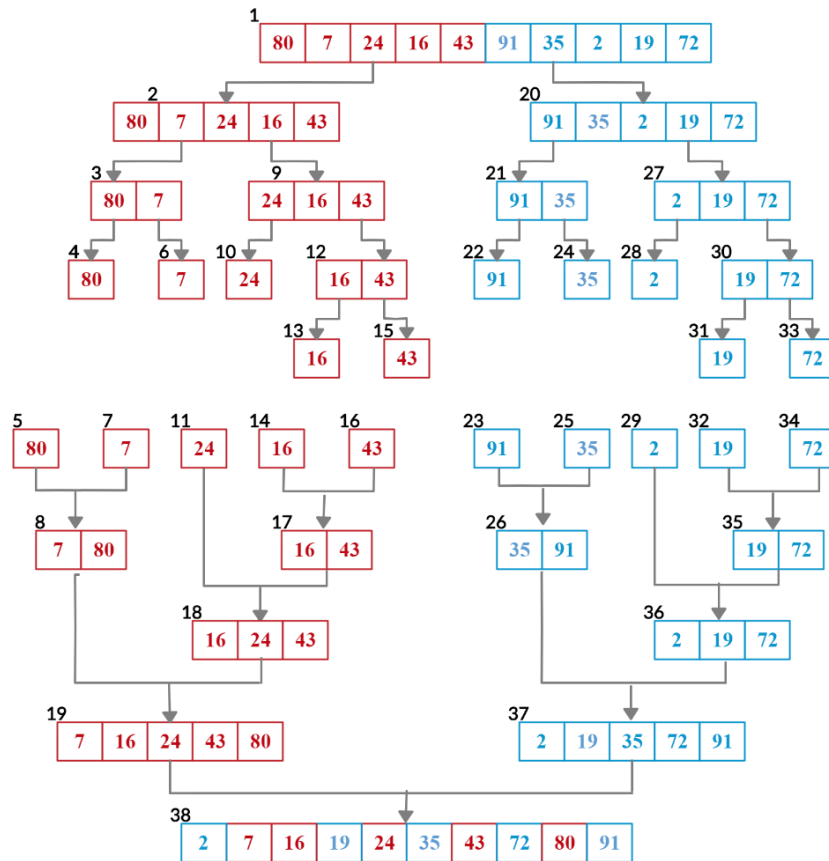
def inse(arr):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i - 1
        while j >= 0 and key < arr[j]:
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key

def mergeSort(arr):
    if len(arr) > 1:
```

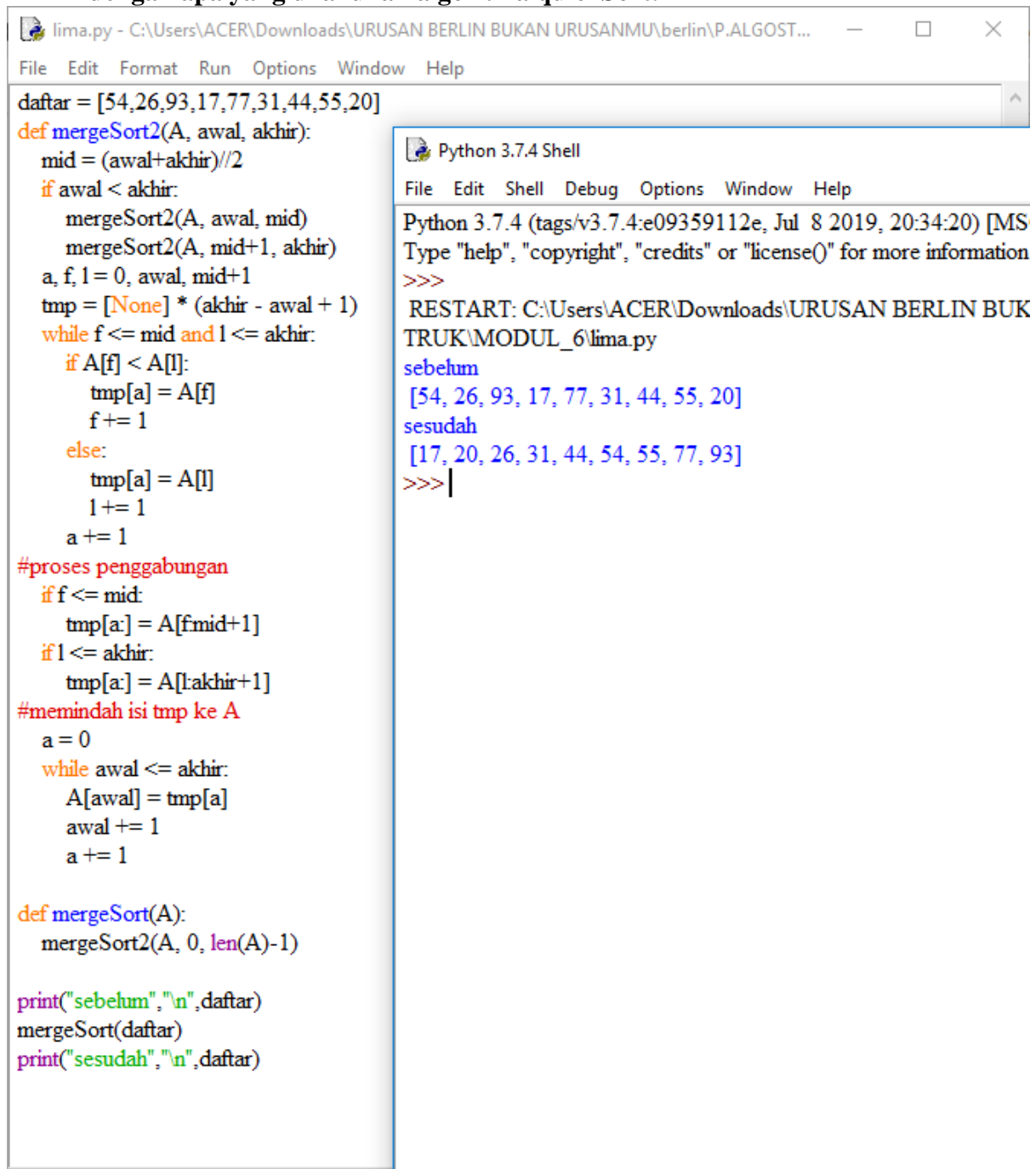
```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help

Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more i
nformation.
>>>
RESTART: C:\Users\ACER\Downloads\URUSAN BE
RLIN BUKAN URUSANMU\berlin\P.ALGOSTRUK\
MODUL_6\tiga.py
bubble : 3.40189 detik
selection : 1.2427 detik
insertion : 1.52491 detik
merge : 0.0289216 detik
quick : 0.0139618 detik
>>> |
```

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



5. Tingkatkan efisiensi program mergeSort dengan tidak memakai operator slice (seperti `A[:mid]` dan `A[mid:]`), dan lalu mem-pass index awal dan index akhir bersama listnya saat kita memanggil mergeSort secara rekursif. Kamu akan perlu memisah fungsi mergeSort itu menjadi beberapa fungsi, mirip halnya dengan apa yang dilakukan algoritma quickSort.



The screenshot shows a Python IDE with a file named `lima.py` and a Python 3.7.4 Shell window. The code in `lima.py` implements a recursive merge sort algorithm. It starts with a list `daftar = [54, 26, 93, 17, 77, 31, 44, 55, 20]`. The `mergeSort2` function recursively splits the list into halves until single elements are reached, then merges them back in sorted order. The `mergeSort` function is a wrapper that calls `mergeSort2` with the initial index 0 and the length of the list minus one. The script prints the list 'sebelum' (before) and 'sesudah' (after) sorting.

```
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)
```

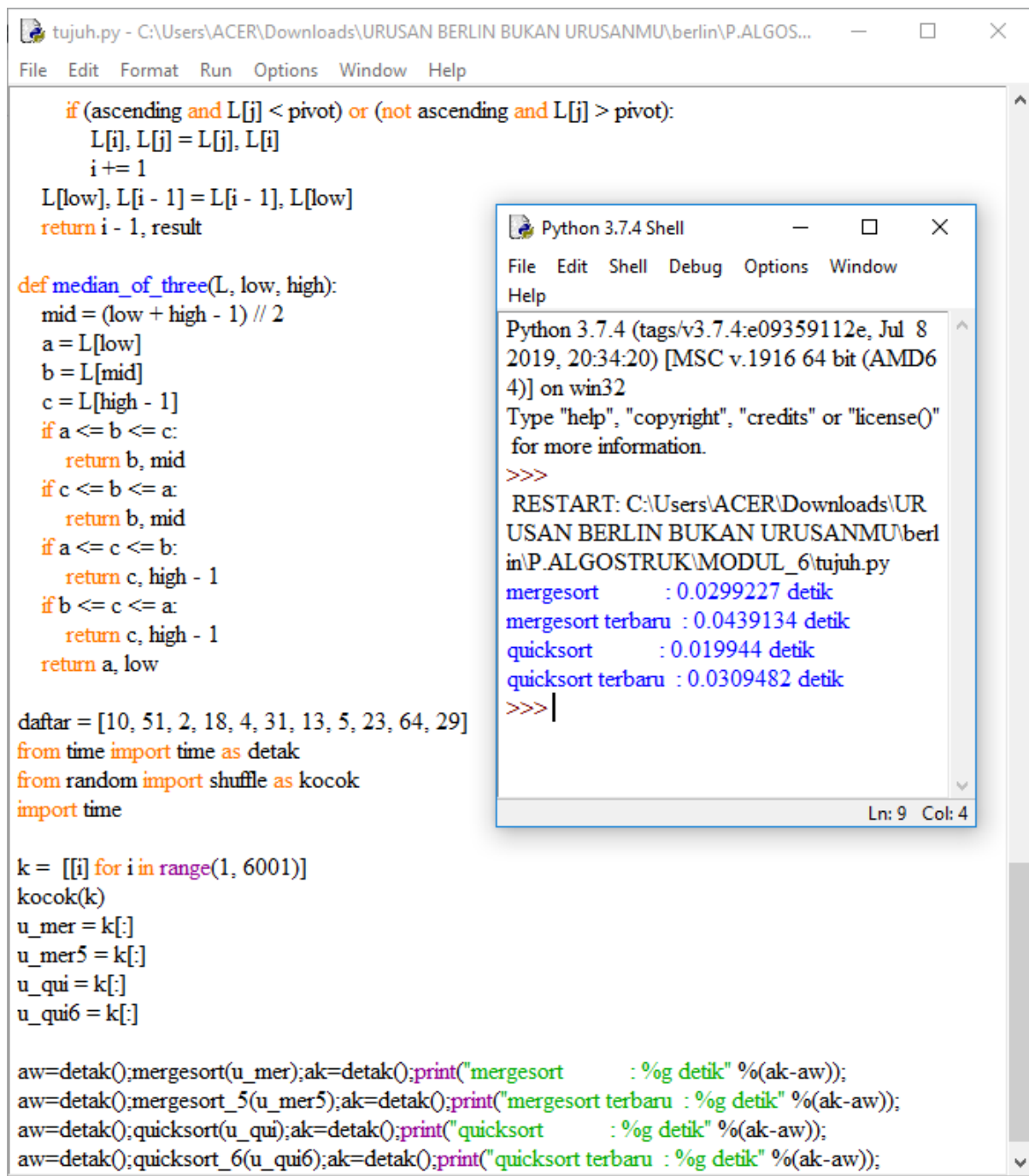
The Python 3.7.4 Shell window shows the execution output:

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MS
Type "help", "copyright", "credits" or "license()" for more information
>>>
RESTART: C:\Users\ACER\Downloads\URUSAN BERLIN BUK
TRUK\MODUL_6\lima.py
sebelum
[54, 26, 93, 17, 77, 31, 44, 55, 20]
sesudah
[17, 20, 26, 31, 44, 54, 55, 77, 93]
>>> |
```

6. Apakah kita bisa meningkatkan efisiensi program quickSort dengan memakai metode median-dari-tiga untuk memilih pivotnya? Ubahlah kodenya dan ujilah.

<pre>enam.py - C:\Users\ACER\Downloads\URUSAN BERLIN BUKAN URUSAN File Edit Format Run Options Window Help 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: return c, high - 1</pre>	<pre>Python 3.7.4 Shell File Edit Shell Debug Options Window Python 3.7.4 (tags/v3.7.4:e09359112e, J 34:20) [MSC v.1916 64 bit (AMD64)] c Type "help", "copyright", "credits" or "lice information. >>> RESTART: C:\Users\ACER\Downloads\ RLIN BUKAN URUSANMU\berlin\P.A MODUL_6\enam.py sebelum [54, 26, 93, 17, 77, 31, 44, 55, 20] sesudah [17, 20, 26, 31, 44, 54, 55, 77, 93] >>> </pre>
---	--

7. Uji-kecepatan keduanya dan perbandingan juga dengan kode awalnya.



The image shows a Python IDE window titled 'tujuh.py' with a menu bar (File, Edit, Format, Run, Options, Window, Help) and a code editor. The code defines a function 'median_of_three' and a list 'daftar'. It then imports 'time' and 'random' modules, and defines 'kocok' function. The script generates a list 'k' of 6001 random numbers and splits it into six parts. It then measures the execution time of 'mergesort' and 'quicksort' on these parts. A terminal window titled 'Python 3.7.4 Shell' is open, showing the output of the script, which compares the execution times of 'mergesort' and 'quicksort' on the generated list and its parts.

```
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 detik
from random import shuffle as kocok
import time

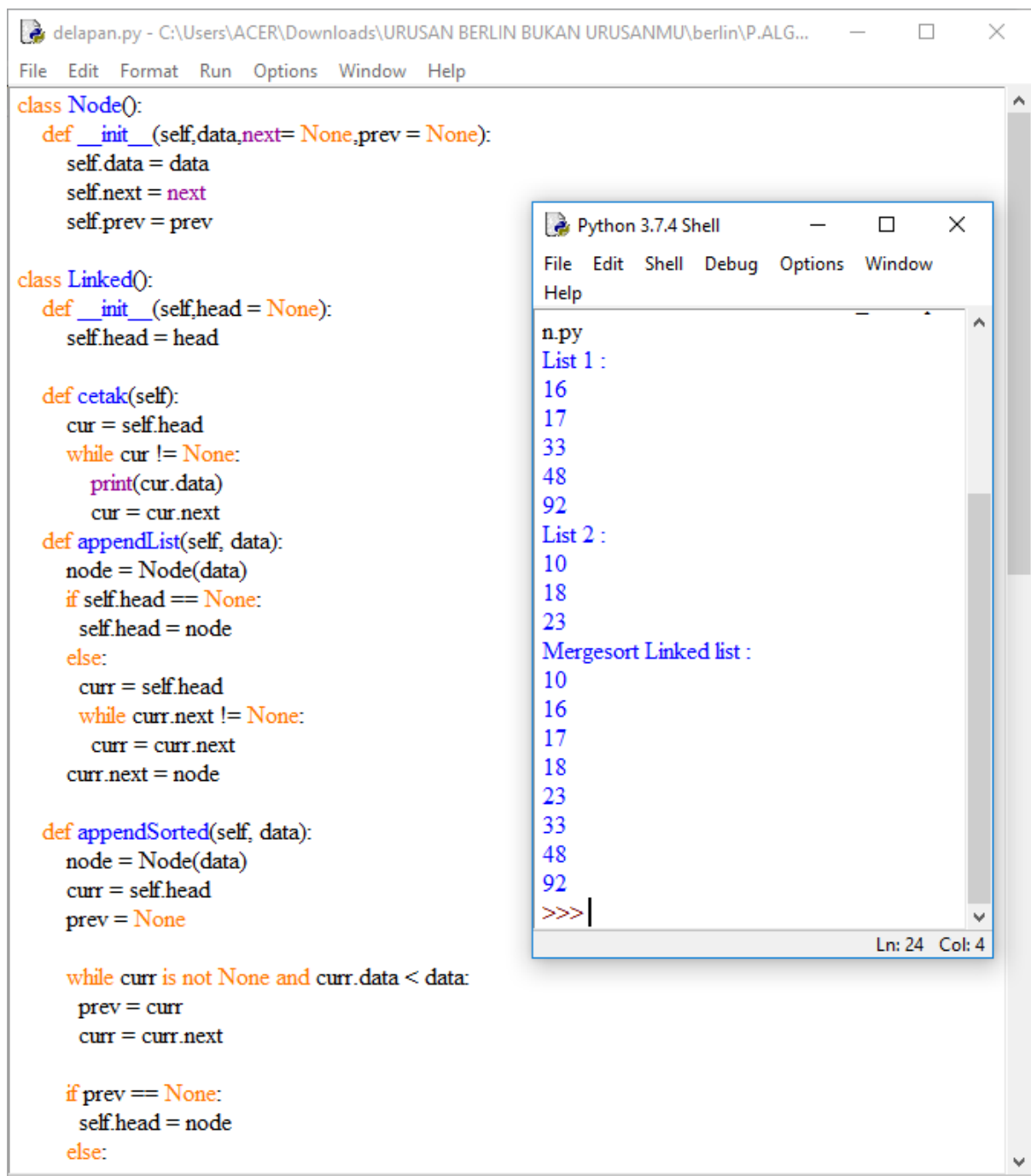
k = [[i for i in range(1, 6001)]
kocok(k)
u_mer = k[:]
u_mer5 = k[:]
u_qui = k[:]
u_qui6 = k[:]

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

Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\ACER\Downloads\URUSAN BERLIN BUKAN URUSANMU\berlin\ALGOSTRUK\MODUL_6\tujuh.py
mergesort : 0.0299227 detik
mergesort terbaru : 0.0439134 detik
quicksort : 0.019944 detik
quicksort terbaru : 0.0309482 detik
>>> |

Ln: 9 Col: 4

8. Buatlah versi linked-list untuk program mergeSort diatas.



The image shows a Python IDE window titled 'delapan.py' with a menu bar (File, Edit, Format, Run, Options, Window, Help). The code defines two classes: `Node` and `Linked`. The `Node` class has an `__init__` method that takes `data`, `next` (default `None`), and `prev` (default `None`) as arguments. The `Linked` class has an `__init__` method that takes `head` (default `None`) as an argument. It also has methods `cetak` (to print the list), `appendList` (to append a new node to the end), `appendSorted` (to insert a new node in sorted order), and a `while` loop for merging two sorted lists. A `Python 3.7.4 Shell` window is open, showing the output of the program. It displays two lists: 'List 1 : 16 17 33 48 92' and 'List 2 : 10 18 23'. Below these, it shows the 'Mergesort Linked list : 10 16 17 18 23 33 48 92'. The shell window has a menu bar (File, Edit, Shell, Debug, Options, Window, Help) and a status bar at the bottom indicating 'Ln: 24 Col: 4'.

```
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:
```

Python 3.7.4 Shell

File Edit Shell Debug Options Window Help

n.py
List 1 :
16
17
33
48
92
List 2 :
10
18
23
Mergesort Linked list :
10
16
17
18
23
33
48
92
>>> |

Ln: 24 Col: 4