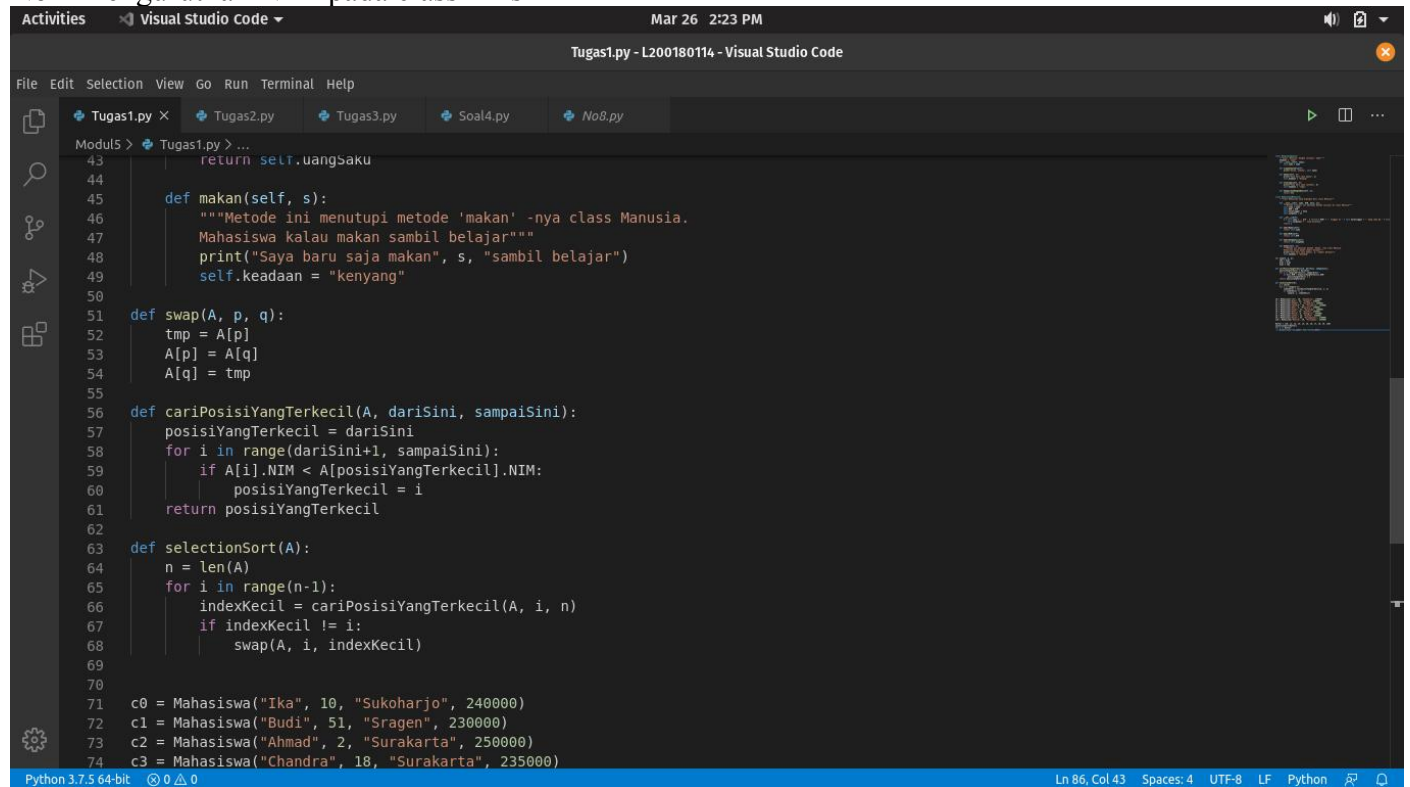


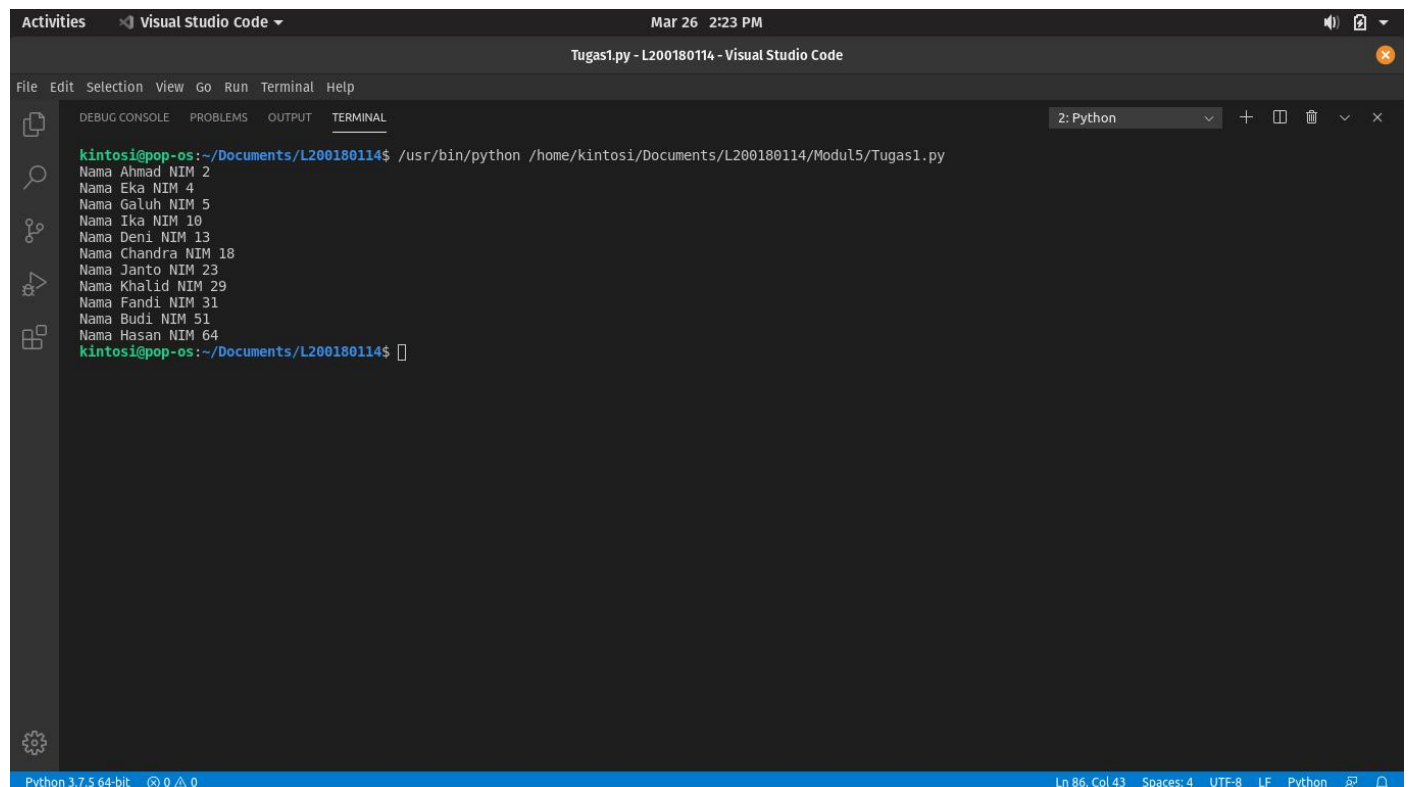
Nama : Arga Dwi Ardinata
NIM : L200180114
Kelas : E

Tugas Mahasiswa

No 1 mengurutkan NIM pada class MhsTIF

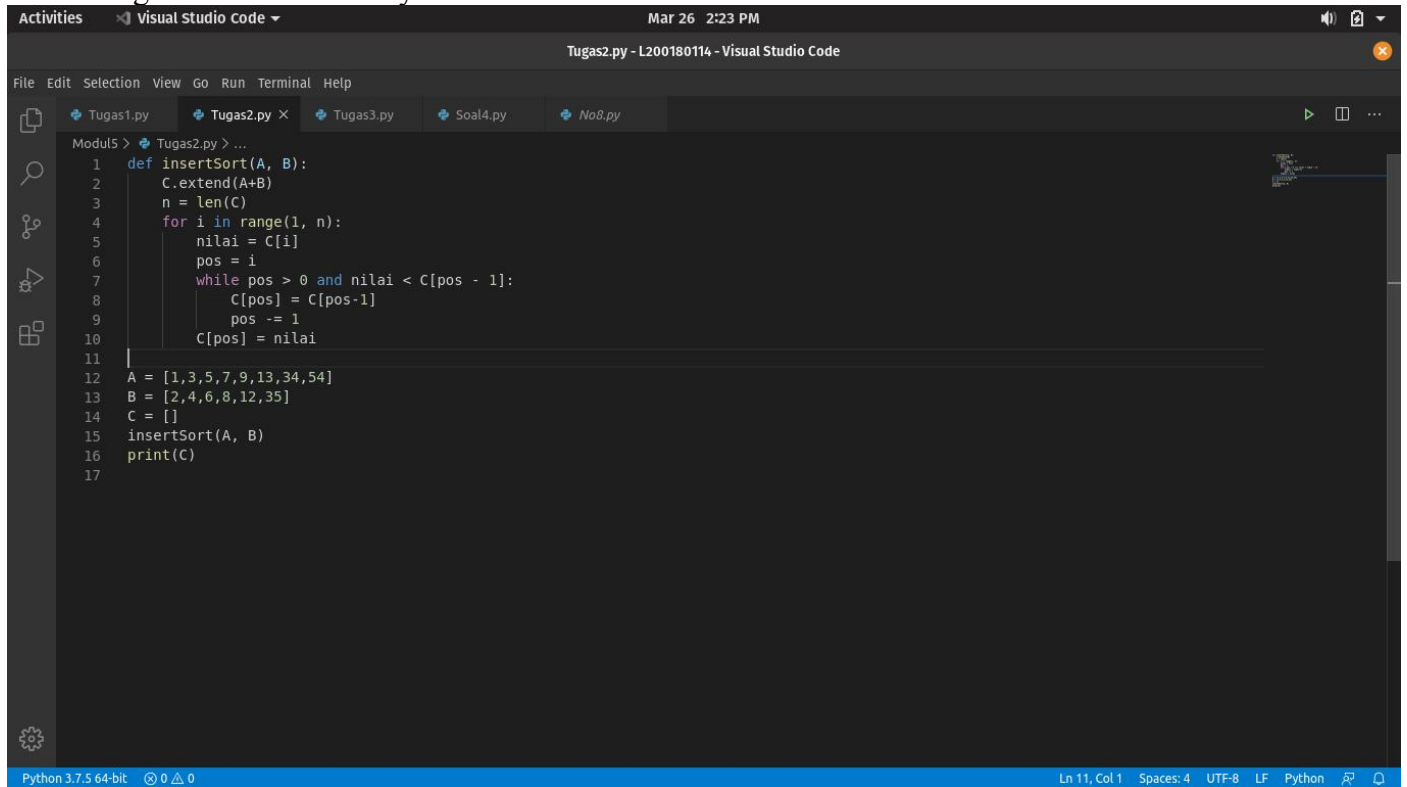


```
43         return self.uangSaku
44
45     def makan(self, s):
46         """Metode ini menutupi metode 'makan' -nya class Manusia.
47         Mahasiswa kalau makan sambil belajar"""
48         print("Saya baru saja makan", s, "sambil belajar")
49         self.keadaan = "kenyang"
50
51     def swap(A, p, q):
52         tmp = A[p]
53         A[p] = A[q]
54         A[q] = tmp
55
56     def cariPosisiYangTerkecil(A, dariSini, sampaiSini):
57         posisiYangTerkecil = dariSini
58         for i in range(dariSini+1, sampaiSini):
59             if A[i].NIM < A[posisiYangTerkecil].NIM:
60                 posisiYangTerkecil = i
61         return posisiYangTerkecil
62
63     def selectionSort(A):
64         n = len(A)
65         for i in range(n-1):
66             indexKecil = cariPosisiYangTerkecil(A, i, n)
67             if indexKecil != i:
68                 swap(A, i, indexKecil)
69
70
71 c0 = Mahasiswa("Ika", 10, "Sukoharjo", 240000)
72 c1 = Mahasiswa("Budi", 51, "Sragen", 230000)
73 c2 = Mahasiswa("Ahmad", 2, "Surakarta", 250000)
74 c3 = Mahasiswa("Chandra", 18, "Surakarta", 235000)
```



```
kintosi@pop-os:~/Documents/L200180114$ /usr/bin/python /home/kintosi/Documents/L200180114/Modul5/Tugast1.py
Nama Ahmad NIM 2
Nama Eka NIM 4
Nama Galuh NIM 5
Nama Ika NIM 10
Nama Deni NIM 13
Nama Chandra NIM 18
Nama Janto NIM 23
Nama Khalid NIM 29
Nama Fandi NIM 31
Nama Budi NIM 51
Nama Hasan NIM 64
kintosi@pop-os:~/Documents/L200180114$
```

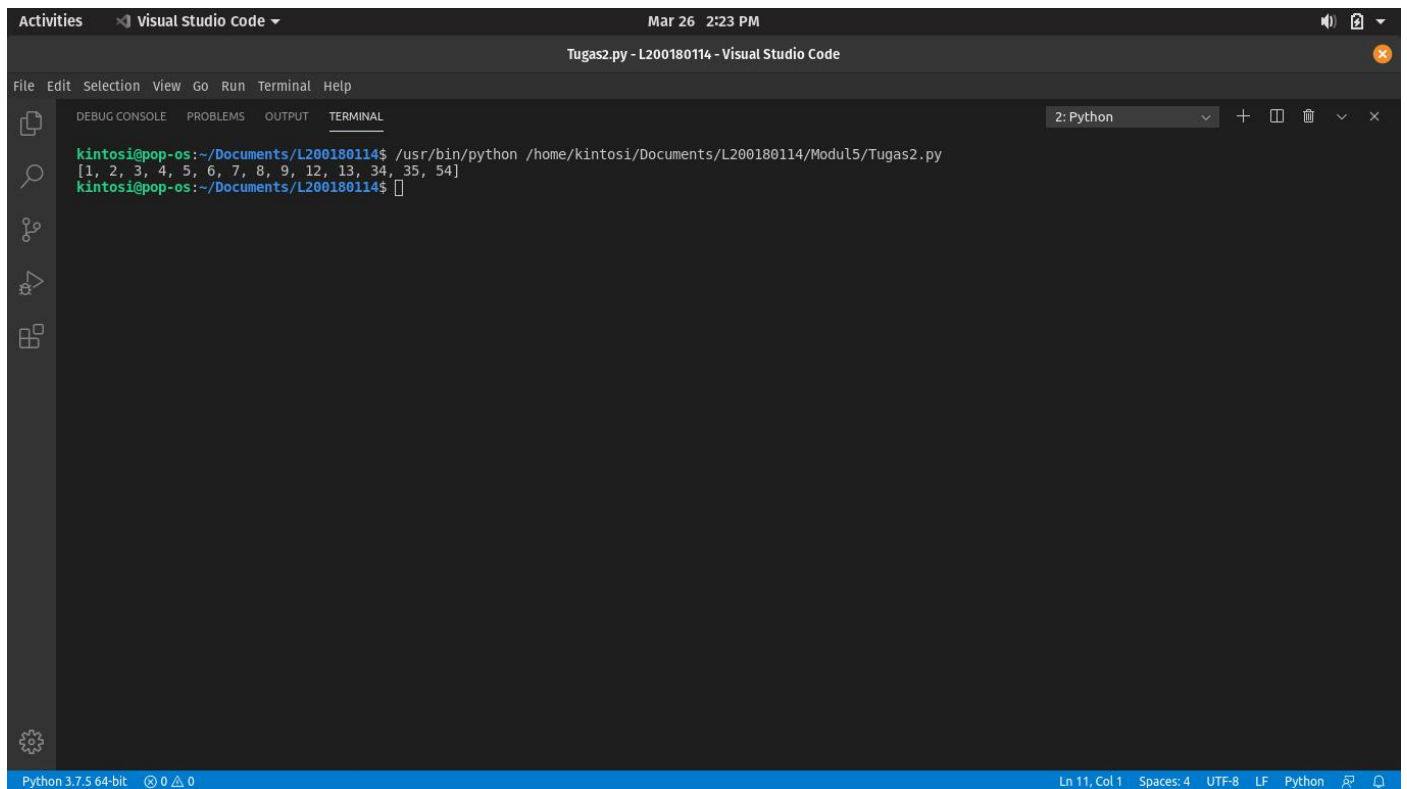
No 2 mengurutkan 2 buah array



The screenshot shows the Visual Studio Code editor with a file named 'Tugas2.py'. The code defines an 'insertSort' function that takes two arrays, A and B, and inserts the elements of B into A in sorted order. The arrays A and B are initialized with specific values, and the function is called to sort the combined elements. The status bar at the bottom indicates 'Python 3.7.5 64-bit'.

```
Modul5 > Tugas2.py > ...
1 def insertSort(A, B):
2     C.extend(A+B)
3     n = len(C)
4     for i in range(1, n):
5         nilai = C[i]
6         pos = i
7         while pos > 0 and nilai < C[pos - 1]:
8             C[pos] = C[pos-1]
9             pos -= 1
10        C[pos] = nilai
11
12 A = [1,3,5,7,9,13,34,54]
13 B = [2,4,6,8,12,35]
14 C = []
15 insertSort(A, B)
16 print(C)
17
```

Python 3.7.5 64-bit 0 0 0 Ln 11, Col 1 Spaces: 4 UTF-8 LF Python

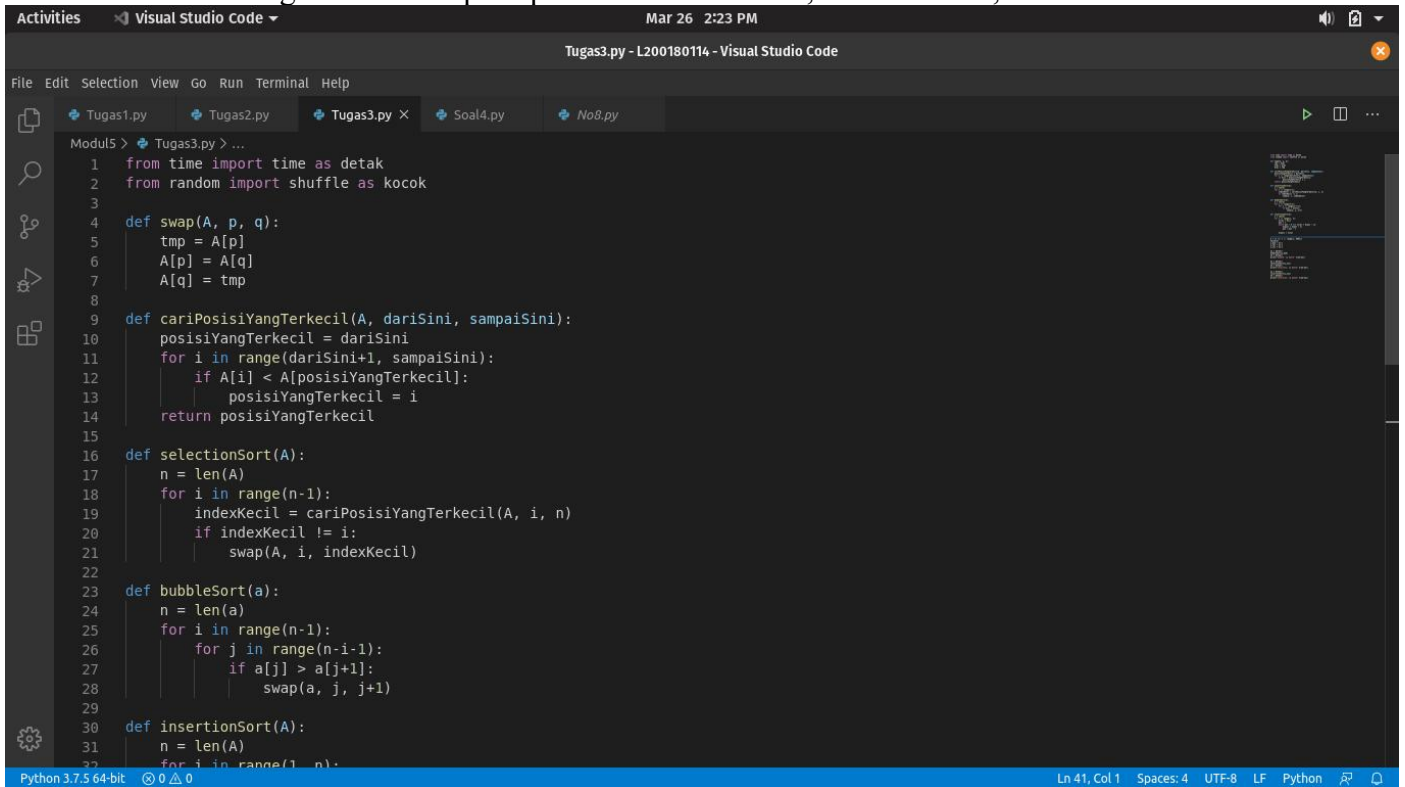


The screenshot shows the terminal window in Visual Studio Code. The command to run the script is entered, and the output shows the sorted array [1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 34, 35, 54]. The status bar at the bottom indicates 'Python 3.7.5 64-bit'.

```
DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL
kintosi@pop-os:~/Documents/L200180114$ /usr/bin/python /home/kintosi/Documents/L200180114/Modul5/Tugas2.py
[1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 34, 35, 54]
kintosi@pop-os:~/Documents/L200180114$
```

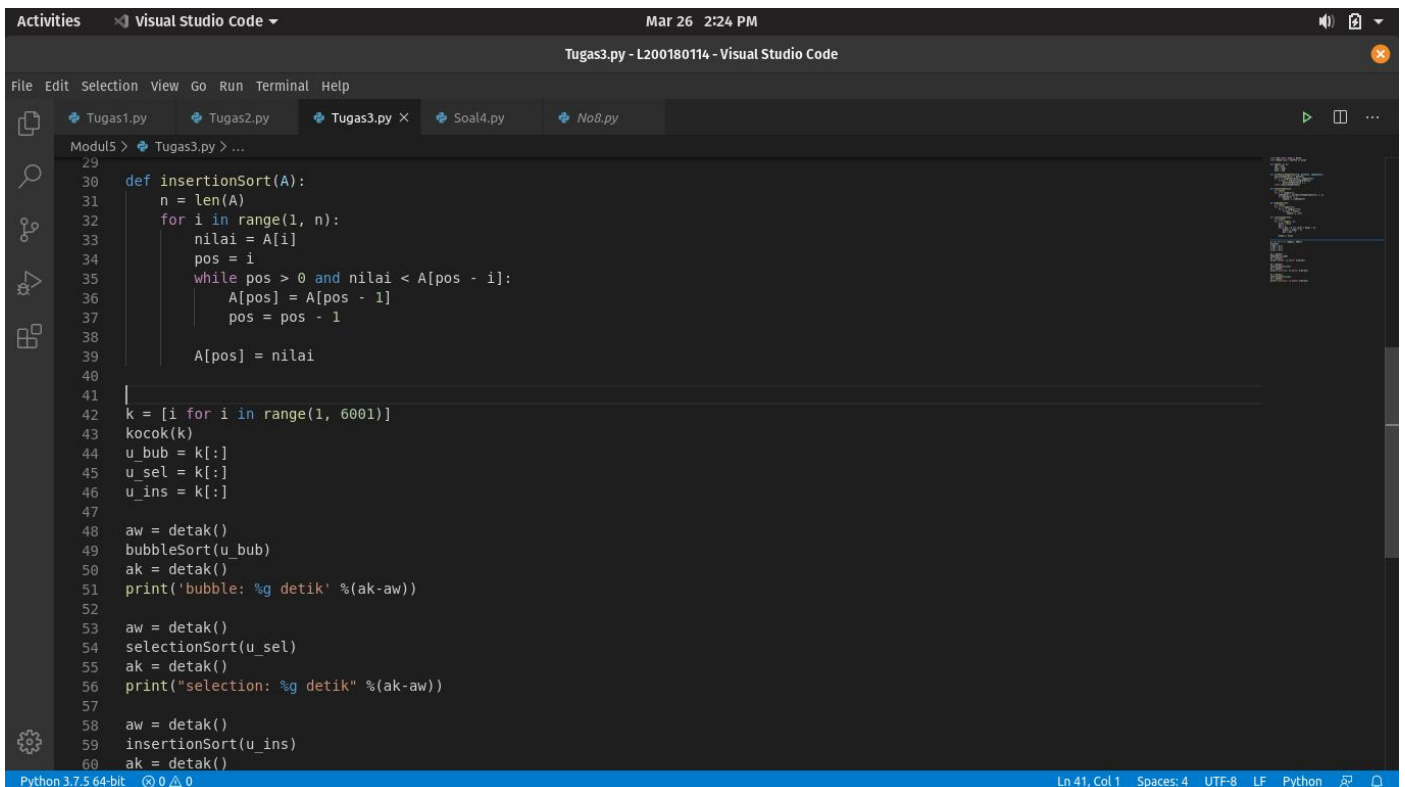
Python 3.7.5 64-bit 0 0 0 Ln 11, Col 1 Spaces: 4 UTF-8 LF Python

No 3 melakukan mengetahui seberapa cepat antara bubble sort, selection sort, dan insertion sort



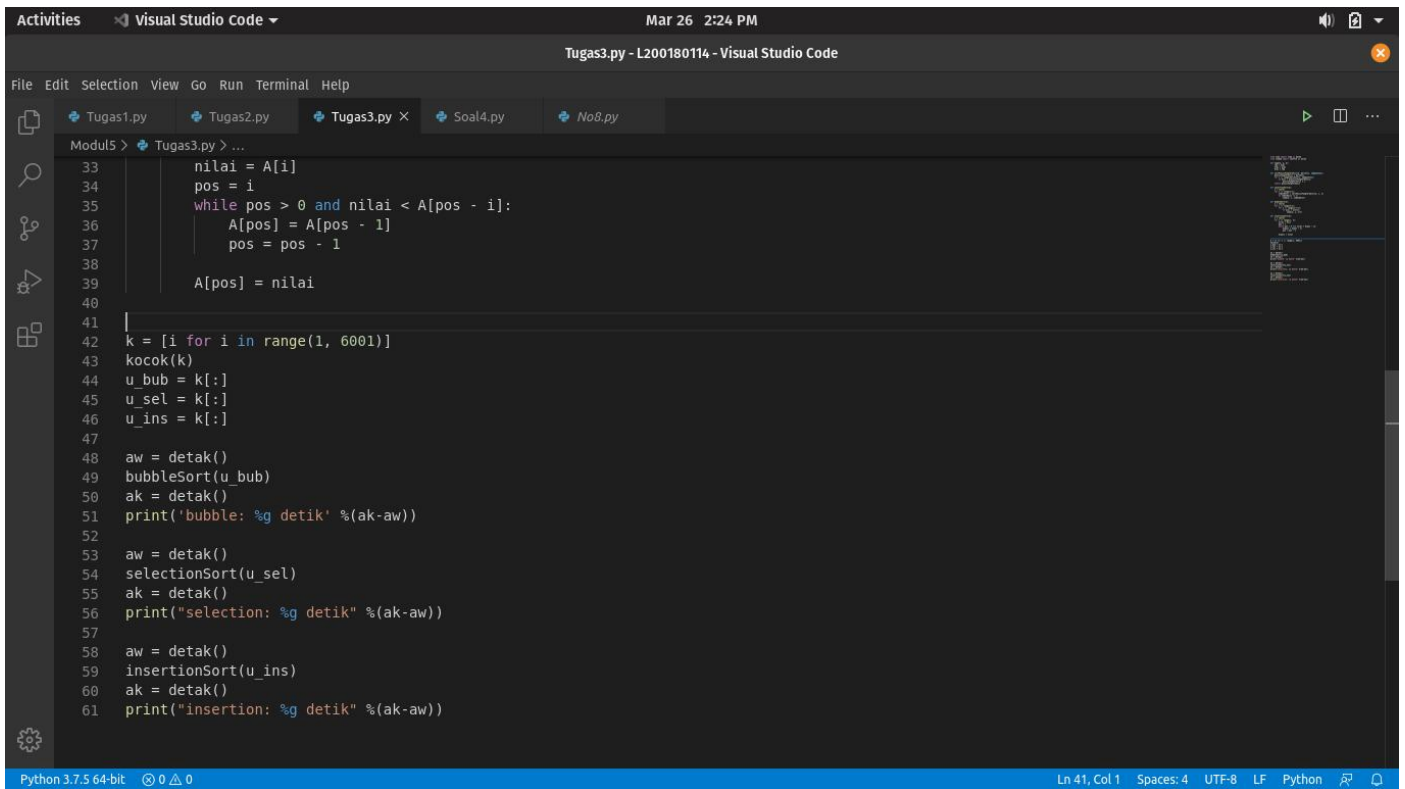
The screenshot shows the Visual Studio Code editor with the file 'Tugas3.py' open. The code defines several functions for sorting: a swap function, a cariPosisiYangTerkecil function, a selectionSort function, a bubbleSort function, and an insertionSort function. The code is written in Python 3.7.5 64-bit. The status bar at the bottom indicates 'Ln 41, Col 1'.

```
1 from time import time as detik
2 from random import shuffle as kocok
3
4 def swap(A, p, q):
5     tmp = A[p]
6     A[p] = A[q]
7     A[q] = tmp
8
9 def cariPosisiYangTerkecil(A, dariSini, sampaiSini):
10     posisiYangTerkecil = dariSini
11     for i in range(dariSini+1, sampaiSini):
12         if A[i] < A[posisiYangTerkecil]:
13             posisiYangTerkecil = i
14     return posisiYangTerkecil
15
16 def selectionSort(A):
17     n = len(A)
18     for i in range(n-1):
19         indexKecil = cariPosisiYangTerkecil(A, i, n)
20         if indexKecil != i:
21             swap(A, i, indexKecil)
22
23 def bubbleSort(a):
24     n = len(a)
25     for i in range(n-1):
26         for j in range(n-i-1):
27             if a[j] > a[j+1]:
28                 swap(a, j, j+1)
29
30 def insertionSort(A):
31     n = len(A)
32     for i in range(1, n):
```

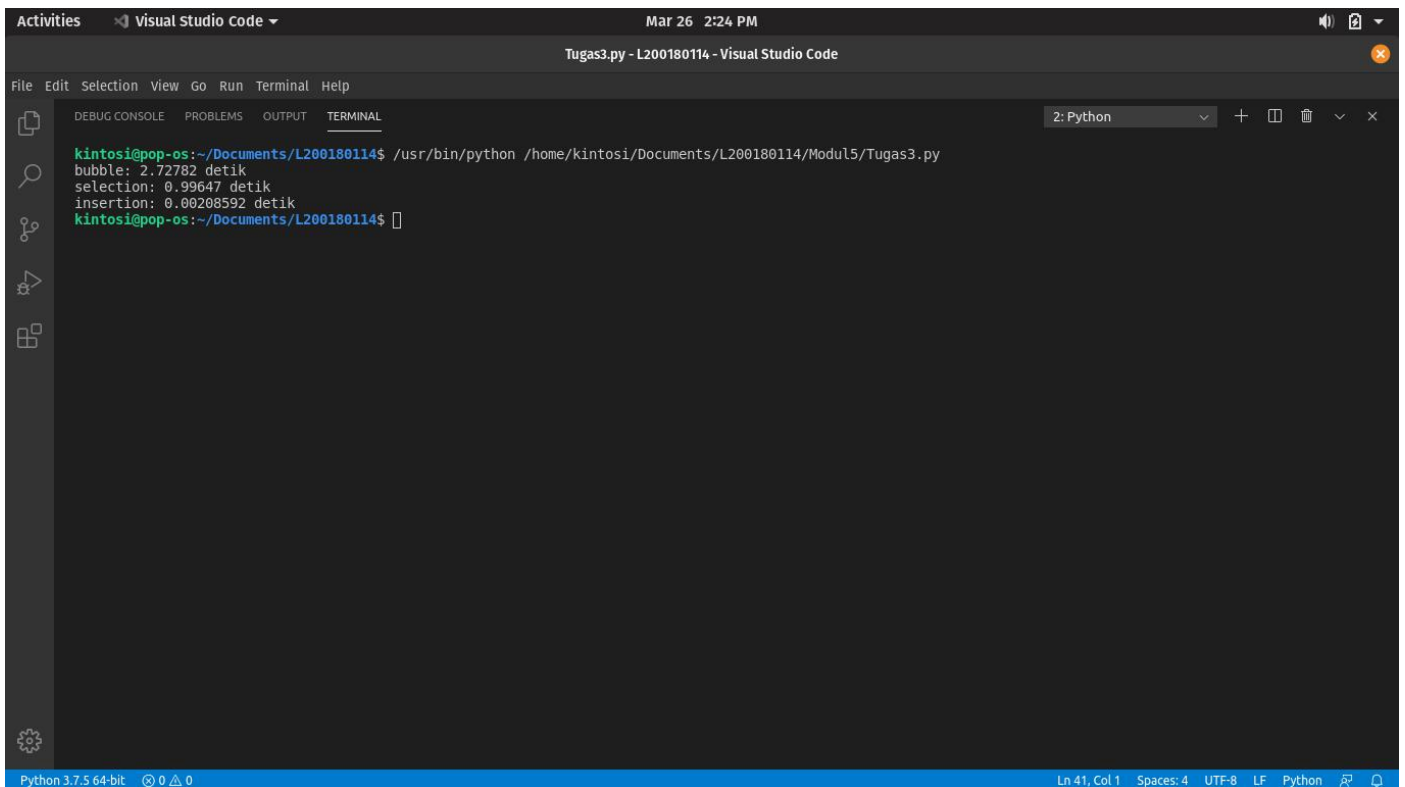


The screenshot shows the continuation of the Python script in Visual Studio Code. It includes the insertionSort function and a main execution block that generates a random list, sorts it using bubble, selection, and insertion sort, and prints the time taken for each. The code is written in Python 3.7.5 64-bit. The status bar at the bottom indicates 'Ln 41, Col 1'.

```
33     nilai = A[i]
34     pos = i
35     while pos > 0 and nilai < A[pos - 1]:
36         A[pos] = A[pos - 1]
37         pos = pos - 1
38     A[pos] = nilai
39
40
41
42 k = [i for i in range(1, 6001)]
43 kocok(k)
44 u_bub = k[:]
45 u_sel = k[:]
46 u_ins = k[:]
47
48 aw = detik()
49 bubbleSort(u_bub)
50 ak = detik()
51 print('bubble: %g detik' %(ak-aw))
52
53 aw = detik()
54 selectionSort(u_sel)
55 ak = detik()
56 print("selection: %g detik" %(ak-aw))
57
58 aw = detik()
59 insertionSort(u_ins)
60 ak = detik()
```



```
33     nilai = A[i]
34     pos = i
35     while pos > 0 and nilai < A[pos - 1]:
36         A[pos] = A[pos - 1]
37         pos = pos - 1
38     A[pos] = nilai
39
40
41
42 k = [i for i in range(1, 6001)]
43 kocok(k)
44 u_bub = k[:]
45 u_sel = k[:]
46 u_ins = k[:]
47
48 aw = detak()
49 bubbleSort(u_bub)
50 ak = detak()
51 print('bubble: %g detik' %(ak-aw))
52
53 aw = detak()
54 selectionSort(u_sel)
55 ak = detak()
56 print("selection: %g detik" %(ak-aw))
57
58 aw = detak()
59 insertionSort(u_ins)
60 ak = detak()
61 print("insertion: %g detik" %(ak-aw))
```



```
kintosi@pop-os:~/Documents/L200180114$ /usr/bin/python /home/kintosi/Documents/L200180114/Modul5/Tugas3.py
bubble: 2.72782 detik
selection: 0.99647 detik
insertion: 0.00208592 detik
kintosi@pop-os:~/Documents/L200180114$
```

Pada ujicoba ini menunjukan bahwa insertion sort yang paling cepat. Sedangkan, bubble sort adalah yang paling lambat diantara ketiga cara tersebut. Bubble sort mencatatkan waktu 2.7 detik. Kemudian, selection sort mencatatkan waktu 0.9 detik. Sedangkan, insertion sort mencatatkan waktu 0.002 detik.