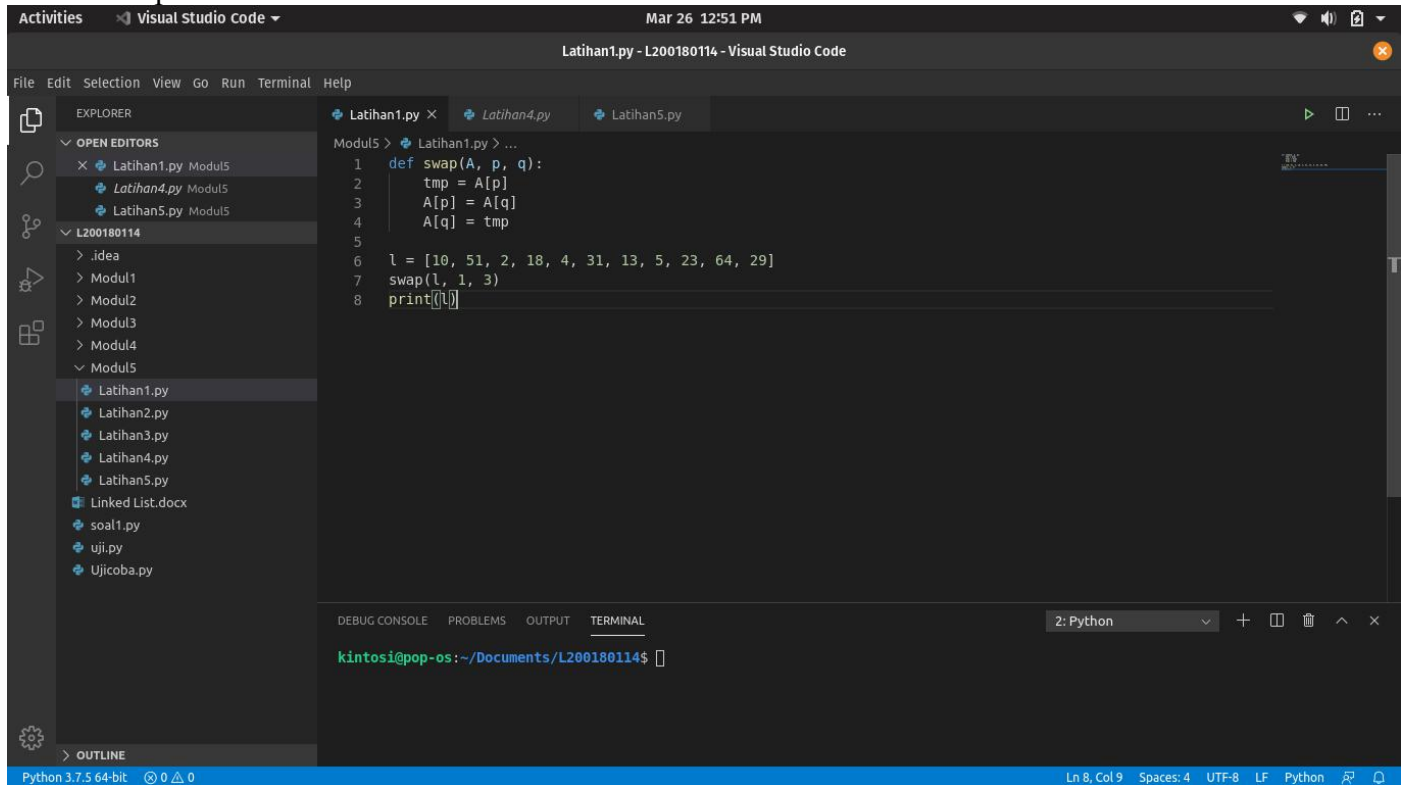


Nama : Arga Dwi Ardinata  
NIM : L200180114  
Kelas : E

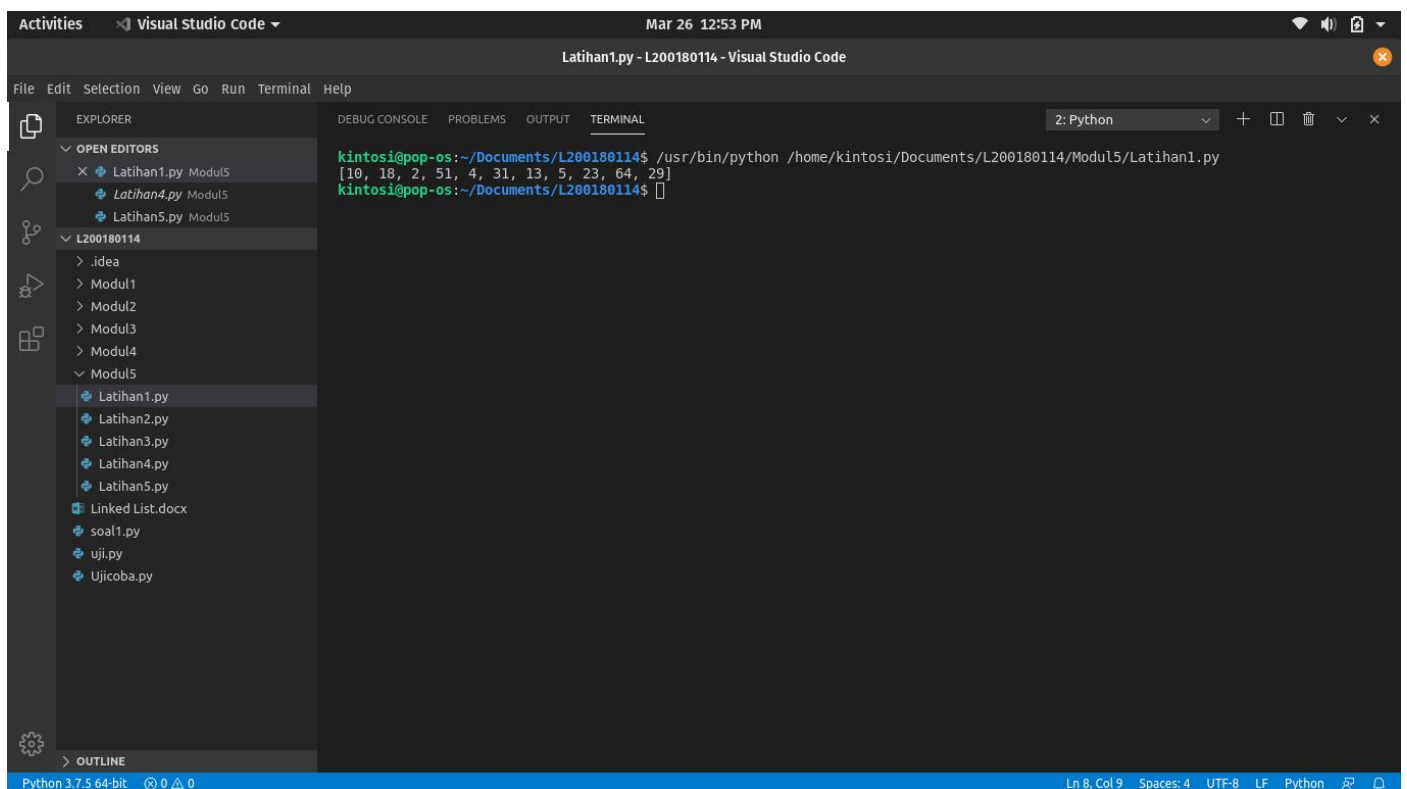
## routine swap



The screenshot shows the Visual Studio Code interface with the file 'Latihan1.py' open. The code defines a swap function and applies it to a list. The Explorer sidebar shows the project structure for 'L200180114'.

```
def swap(A, p, q):  
    tmp = A[p]  
    A[p] = A[q]  
    A[q] = tmp  
  
l = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]  
swap(l, 1, 3)  
print(l)
```

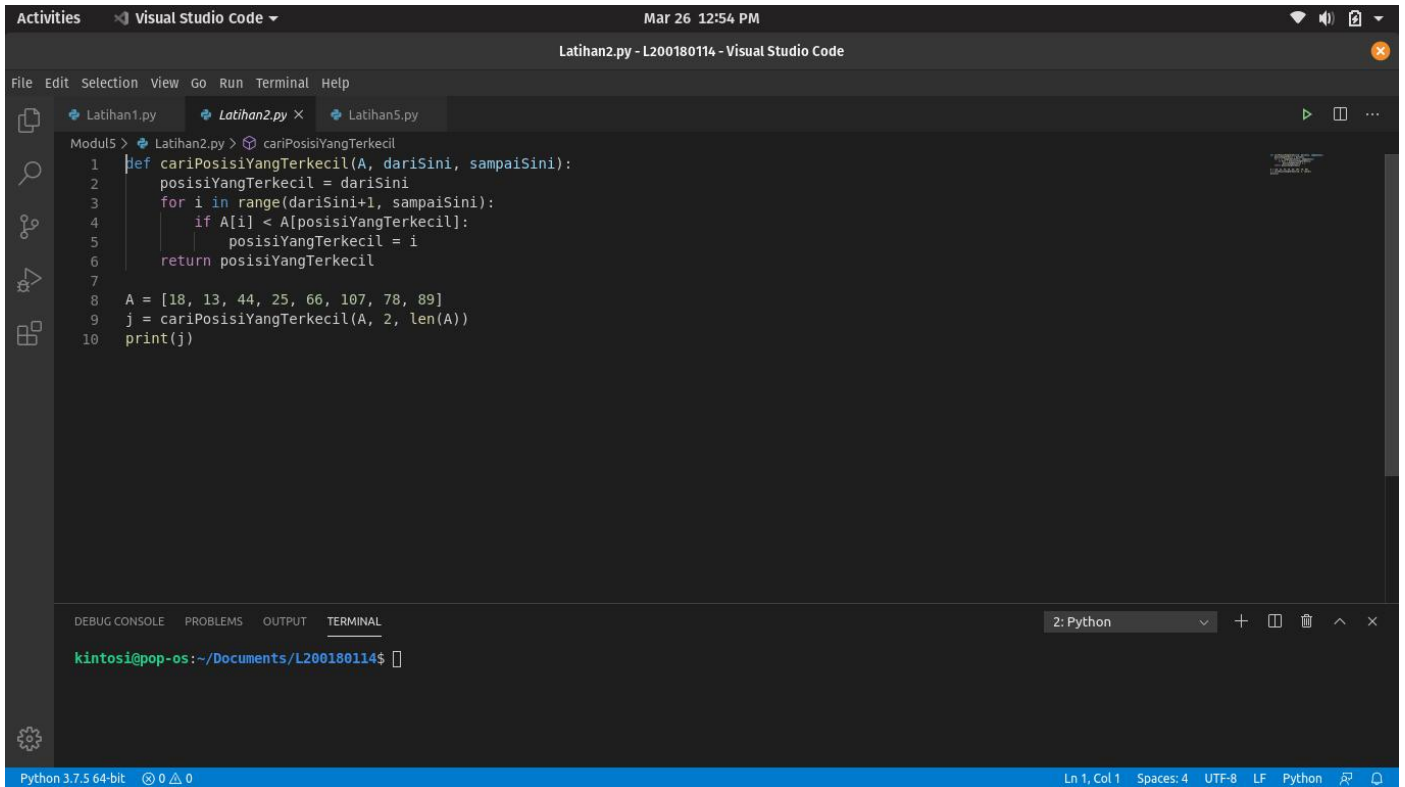
The terminal at the bottom shows the command prompt: `kintosi@pop-os:~/Documents/L200180114$`.



The screenshot shows the same Visual Studio Code interface, but the terminal now displays the output of the Python script. The command executed is `/usr/bin/python /home/kintosi/Documents/L200180114/Modul5/Latihan1.py`, and the output is the modified list: `[10, 18, 2, 51, 4, 31, 13, 5, 23, 64, 29]`.

```
kintosi@pop-os:~/Documents/L200180114$ /usr/bin/python /home/kintosi/Documents/L200180114/Modul5/Latihan1.py  
[10, 18, 2, 51, 4, 31, 13, 5, 23, 64, 29]  
kintosi@pop-os:~/Documents/L200180114$
```

## Routinw untuk mencari nilai terkecil



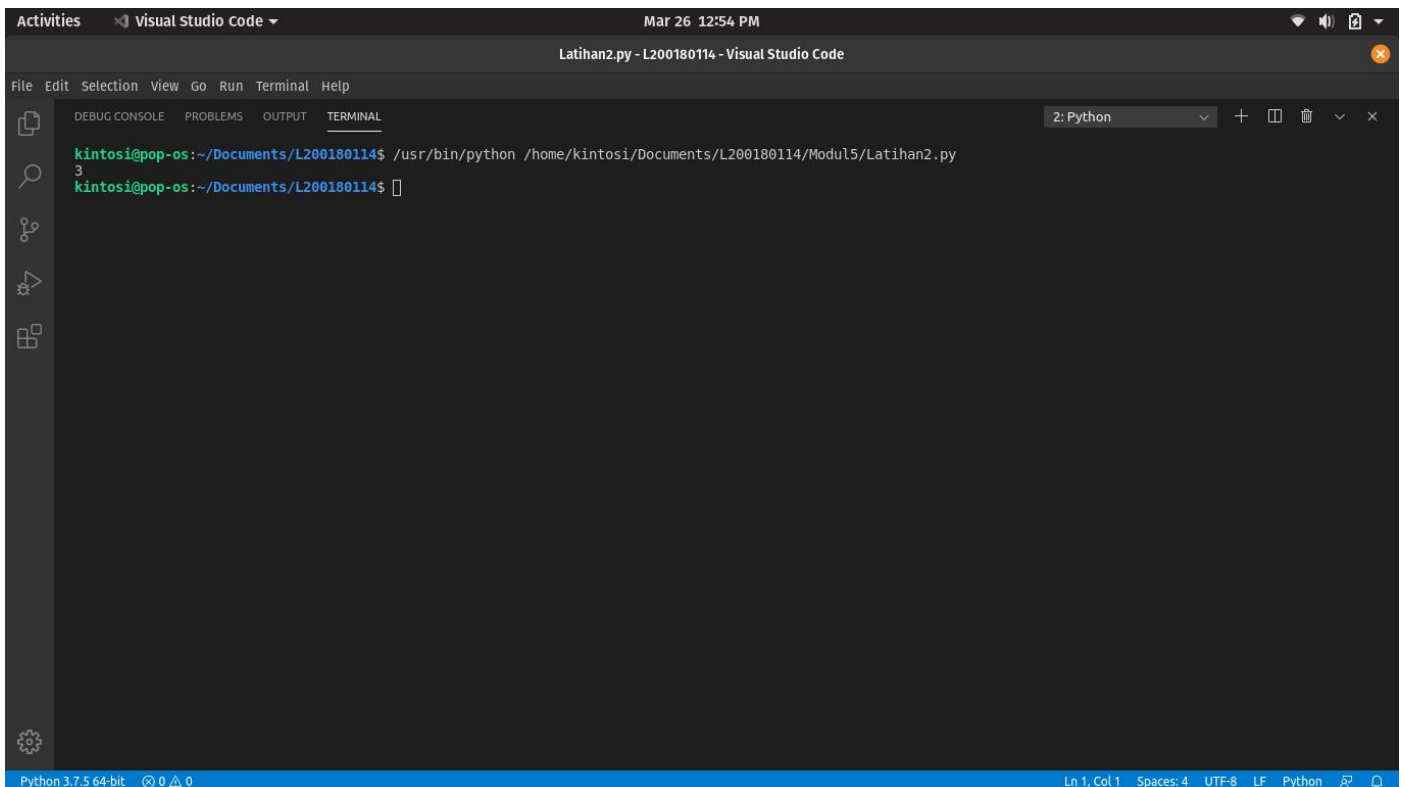
The screenshot shows the Visual Studio Code interface with a file named 'Latihan2.py' open. The editor displays a Python function 'cariPosisiYangTerkecil' that takes a list 'A', a starting index 'dariSini', and an ending index 'sampaiSini'. The function iterates through the list from 'dariSini' to 'sampaiSini' to find the minimum value and its index. Below the function, a list 'A' is defined with values [18, 13, 44, 25, 66, 107, 78, 89], and the function is called with 'j = cariPosisiYangTerkecil(A, 2, len(A))'. The terminal at the bottom shows the command 'kintosi@pop-os:~/Documents/L200180114\$'.

```
Modul5 > Latihan2.py > cariPosisiYangTerkecil
1 def cariPosisiYangTerkecil(A, dariSini, sampaiSini):
2     posisiYangTerkecil = dariSini
3     for i in range(dariSini+1, sampaiSini):
4         if A[i] < A[posisiYangTerkecil]:
5             posisiYangTerkecil = i
6     return posisiYangTerkecil
7
8 A = [18, 13, 44, 25, 66, 107, 78, 89]
9 j = cariPosisiYangTerkecil(A, 2, len(A))
10 print(j)
```

DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL 2: Python

kintosi@pop-os:~/Documents/L200180114\$

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

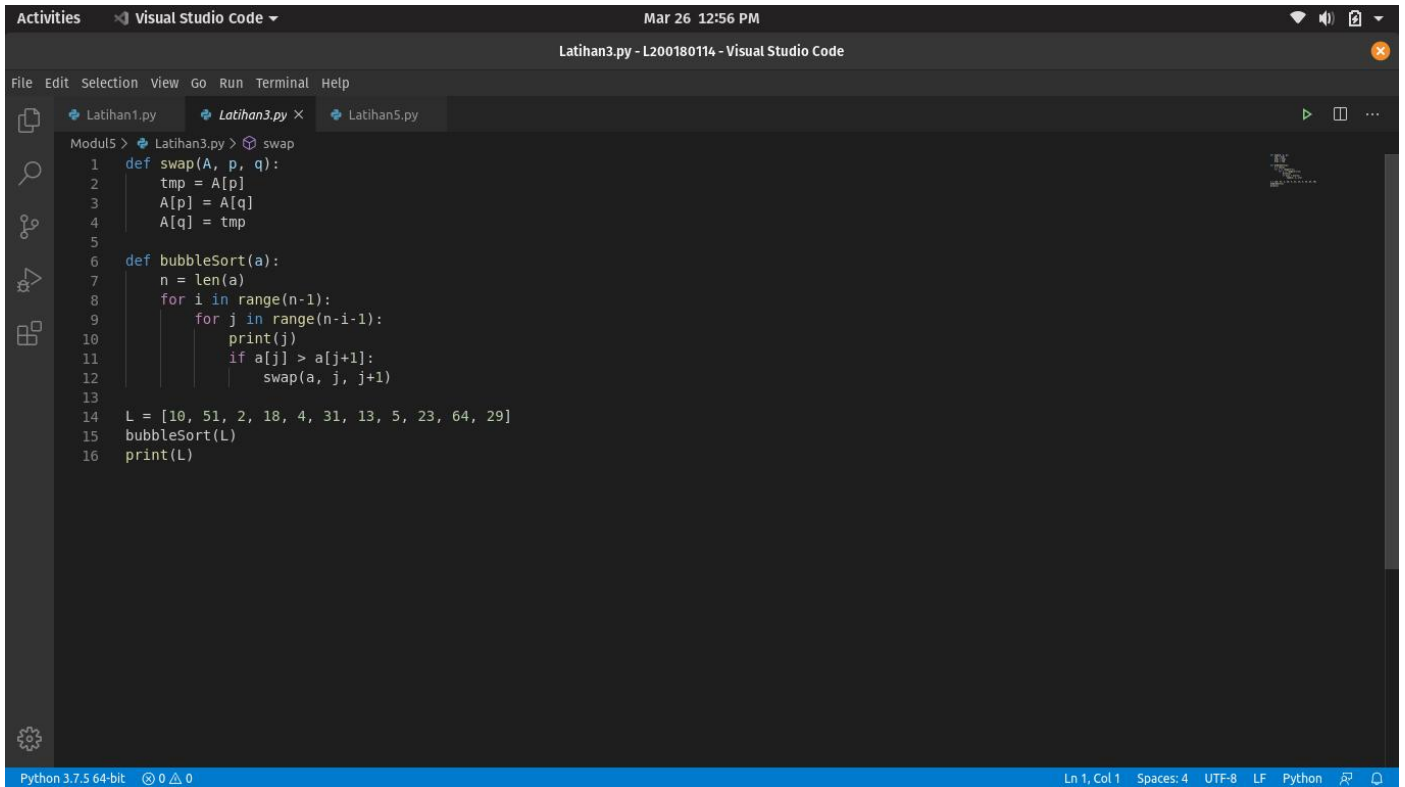


The screenshot shows the Visual Studio Code interface with the terminal window active. The terminal displays the command to run the Python script: '/usr/bin/python /home/kintosi/Documents/L200180114/Modul5/Latihan2.py'. The output of the script is '3', which is the index of the minimum value in the list. The terminal prompt is 'kintosi@pop-os:~/Documents/L200180114\$'.

```
DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL 2: Python
kintosi@pop-os:~/Documents/L200180114$ /usr/bin/python /home/kintosi/Documents/L200180114/Modul5/Latihan2.py
3
kintosi@pop-os:~/Documents/L200180114$
```

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

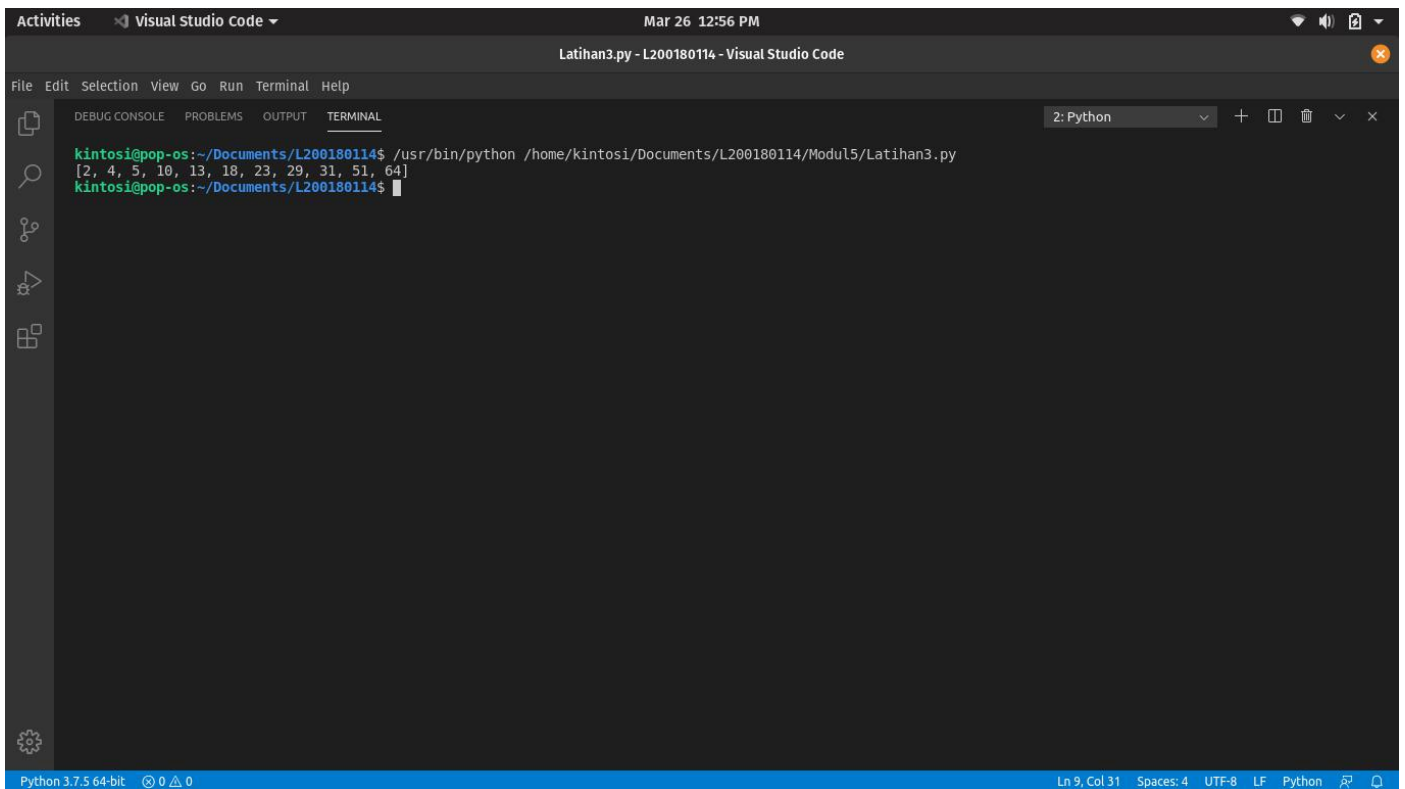
# Bubble Sort



The screenshot shows the Visual Studio Code editor with a file named 'Latihan3.py' open. The code is a Python script implementing the Bubble Sort algorithm. It includes a 'swap' function to exchange elements in a list and a 'bubbleSort' function that iterates through the list, comparing adjacent elements and swapping them if they are in the wrong order. The list 'L' is initialized with the values [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29].

```
Modul5 > Latihan3.py > swap
1 def swap(A, p, q):
2     tmp = A[p]
3     A[p] = A[q]
4     A[q] = tmp
5
6 def bubbleSort(a):
7     n = len(a)
8     for i in range(n-1):
9         for j in range(n-i-1):
10             print(j)
11             if a[j] > a[j+1]:
12                 swap(a, j, j+1)
13
14 L = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]
15 bubbleSort(L)
16 print(L)
```

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

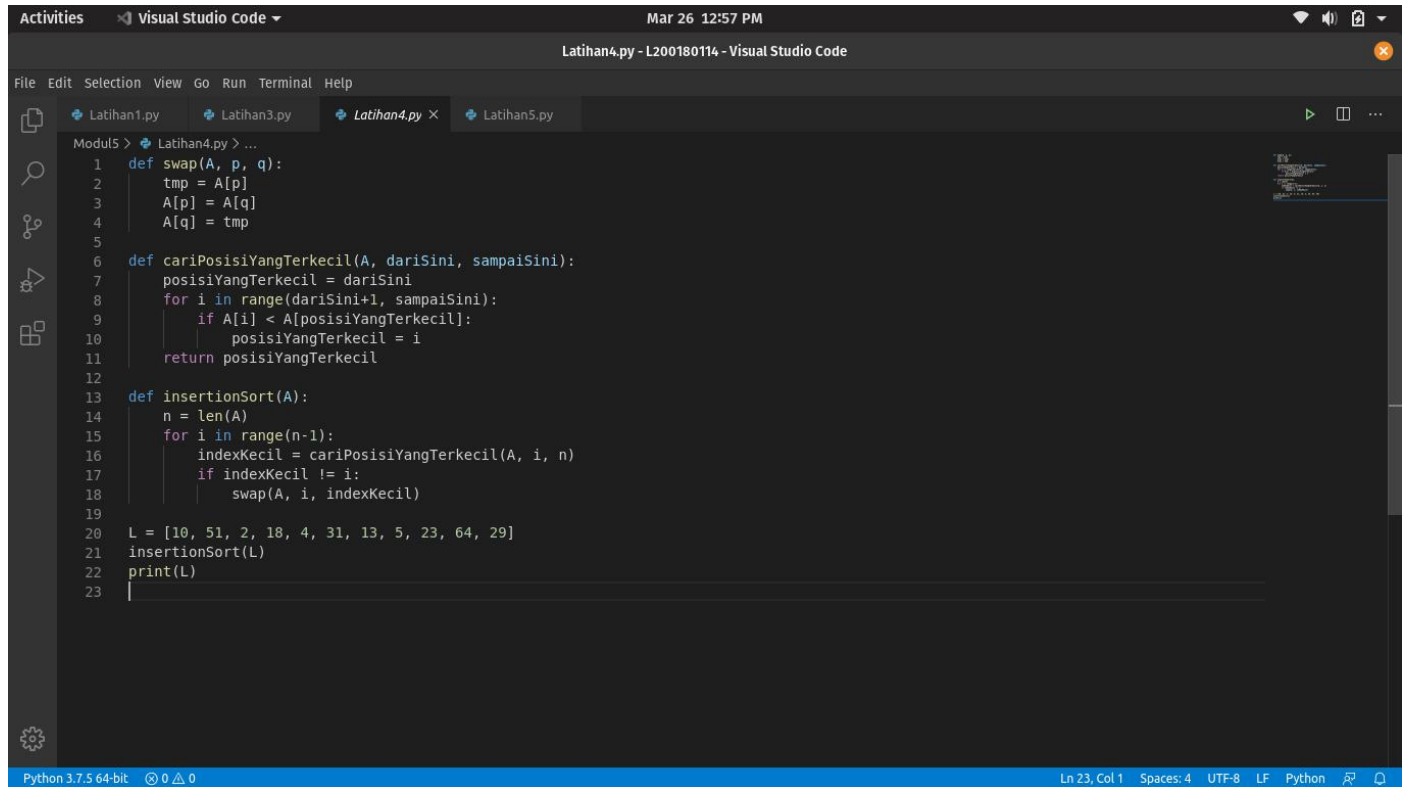


The screenshot shows the Visual Studio Code terminal window with the command prompt. The command executed is `/usr/bin/python /home/kintosi/Documents/L200180114/Modul5/Latihan3.py`. The output of the script is displayed in the terminal, showing the sorted list: [2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64].

```
DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL
kintosi@pop-os:~/Documents/L200180114$ /usr/bin/python /home/kintosi/Documents/L200180114/Modul5/Latihan3.py
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
kintosi@pop-os:~/Documents/L200180114$
```

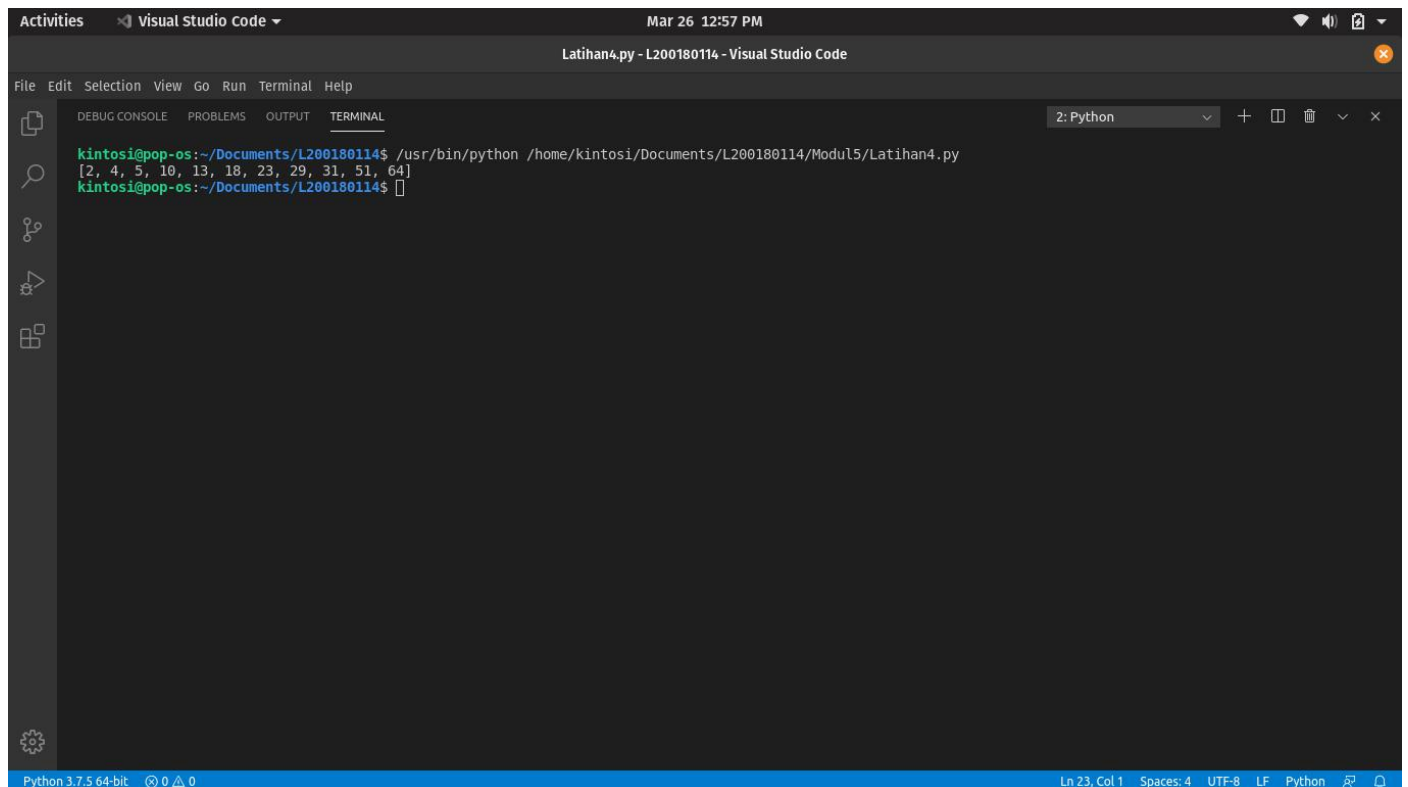
2: Python Ln 9, Col 31 Spaces: 4 UTF-8 LF Python

# Selection Sort



The screenshot shows the Visual Studio Code editor with a file named 'Latihan4.py'. The code implements a Selection Sort algorithm. It includes a 'swap' function, a 'cariPosisiYangTerkecil' function to find the minimum element in a subarray, and an 'insertionSort' function that uses the 'cariPosisiYangTerkecil' function to perform the sorting. The main part of the code initializes a list 'L' with the values [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29] and calls the 'insertionSort' function on it, followed by printing the sorted list.

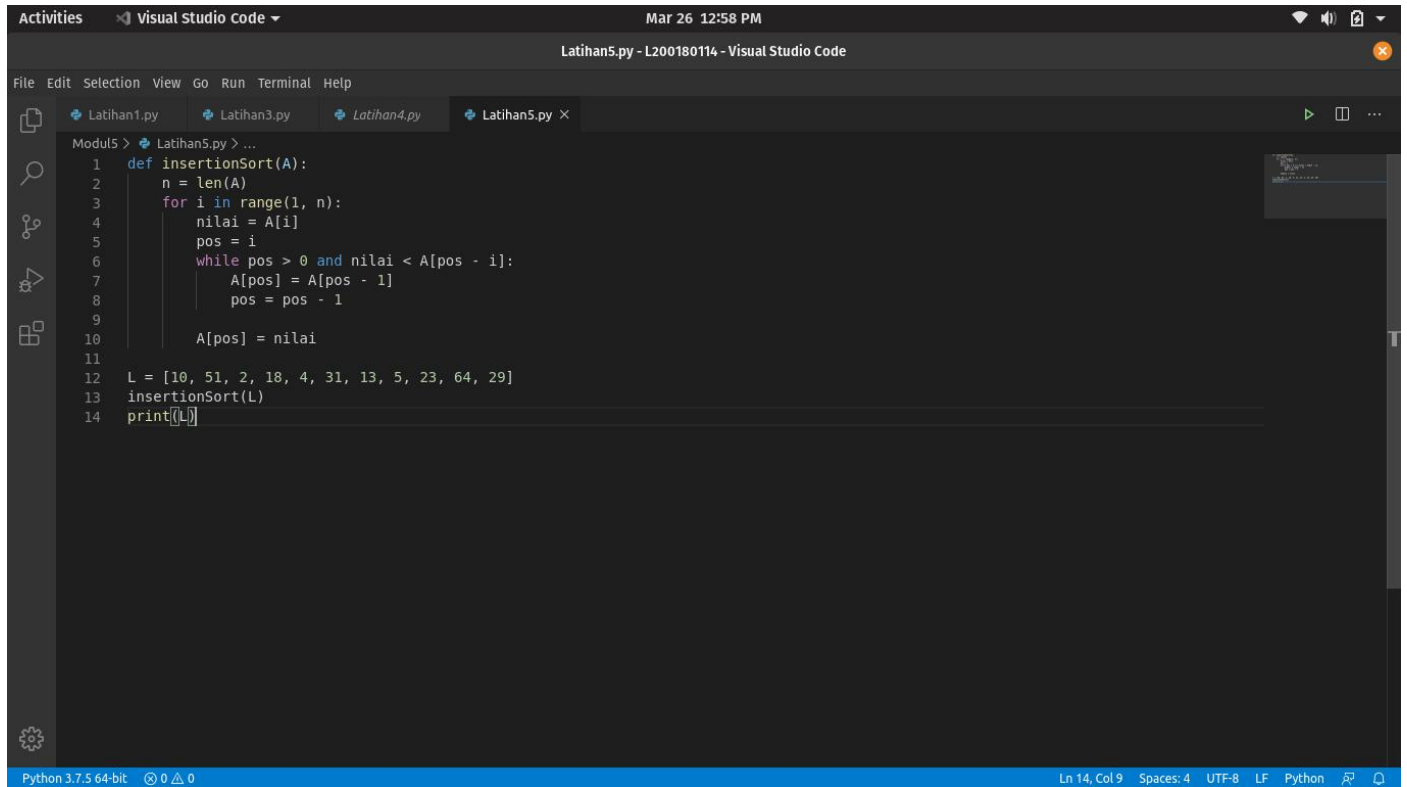
```
Modul5 > Latihan4.py > ...
1  def swap(A, p, q):
2      tmp = A[p]
3      A[p] = A[q]
4      A[q] = tmp
5
6  def cariPosisiYangTerkecil(A, dariSini, sampaiSini):
7      posisiYangTerkecil = dariSini
8      for i in range(dariSini+1, sampaiSini):
9          if A[i] < A[posisiYangTerkecil]:
10             posisiYangTerkecil = i
11      return posisiYangTerkecil
12
13 def insertionSort(A):
14     n = len(A)
15     for i in range(n-1):
16         indexKecil = cariPosisiYangTerkecil(A, i, n)
17         if indexKecil != i:
18             swap(A, i, indexKecil)
19
20 L = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]
21 insertionSort(L)
22 print(L)
23
```



The screenshot shows the terminal window in Visual Studio Code. The command executed is `/usr/bin/python /home/kintosi/Documents/L200180114/Modul5/Latihan4.py`. The output of the program is the sorted list: `[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]`.

```
DEBUG CONSOLE  PROBLEMS  OUTPUT  TERMINAL
kintosi@pop-os:~/Documents/L200180114$ /usr/bin/python /home/kintosi/Documents/L200180114/Modul5/Latihan4.py
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
kintosi@pop-os:~/Documents/L200180114$
```

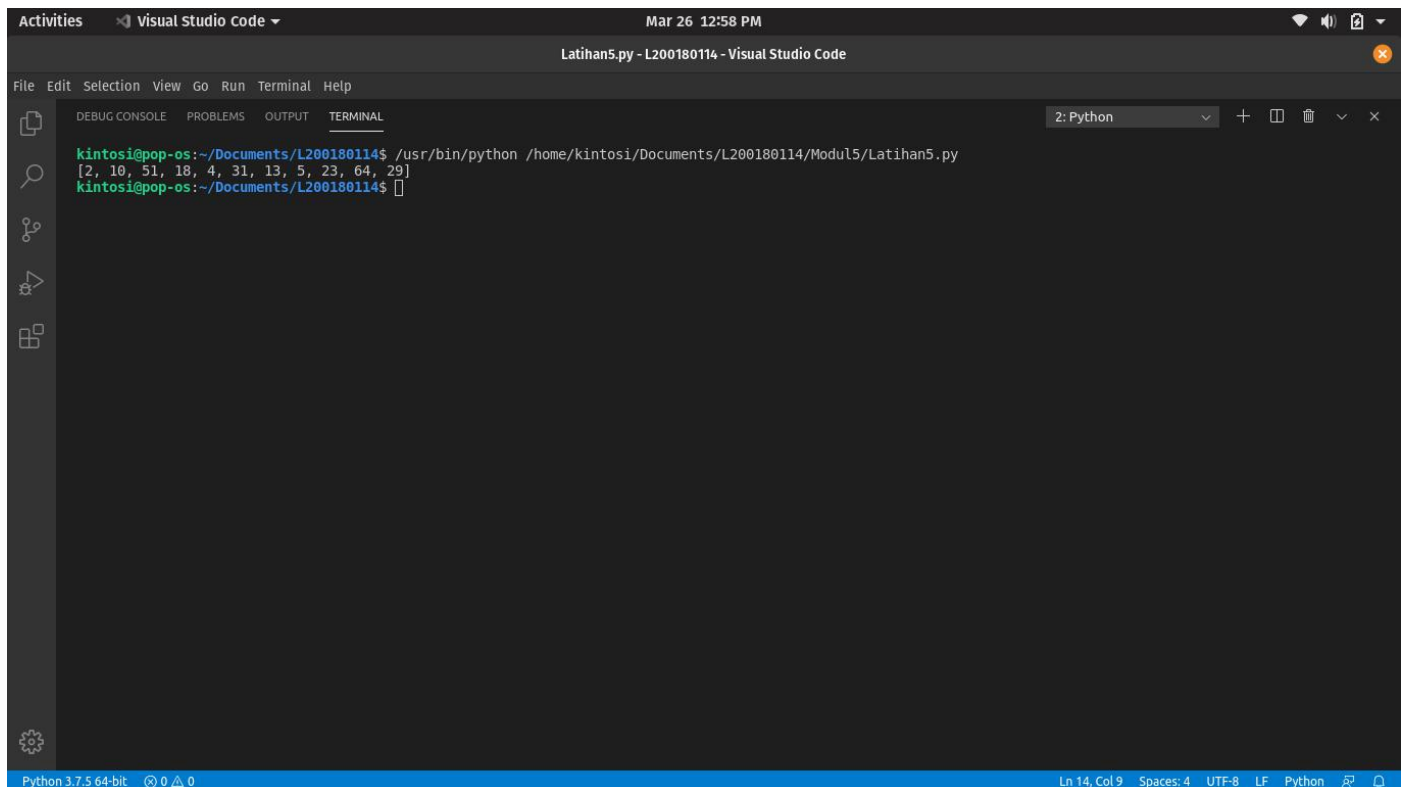
# Insertion Sort



The screenshot shows the Visual Studio Code editor with a file named 'Latihan5.py' open. The code defines an 'insertionSort' function and applies it to a list 'L'. The function iterates through the list, comparing each element with the previous ones and shifting them to the right until the correct position is found. The list 'L' is initialized with the values [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29].

```
Modul5 > Latihan5.py > ...
1  def insertionSort(A):
2      n = len(A)
3      for i in range(1, n):
4          nilai = A[i]
5          pos = i
6          while pos > 0 and nilai < A[pos - i]:
7              A[pos] = A[pos - 1]
8              pos = pos - 1
9
10         A[pos] = nilai
11
12 L = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]
13 insertionSort(L)
14 print(L)
```

The status bar at the bottom indicates 'Python 3.7.5 64-bit' and 'Ln 14, Col 9'.



The screenshot shows the terminal window in Visual Studio Code. The command executed is 'python /home/kintosi/Documents/L200180114/Modul5/Latihan5.py'. The output shows the sorted list: '[2, 10, 51, 18, 4, 31, 13, 5, 23, 64, 29]'. The terminal title is '2: Python'.

```
DEBUG CONSOLE  PROBLEMS  OUTPUT  TERMINAL
kintosi@pop-os:~/Documents/L200180114$ /usr/bin/python /home/kintosi/Documents/L200180114/Modul5/Latihan5.py
[2, 10, 51, 18, 4, 31, 13, 5, 23, 64, 29]
kintosi@pop-os:~/Documents/L200180114$
```

The status bar at the bottom indicates 'Python 3.7.5 64-bit' and 'Ln 14, Col 9'.