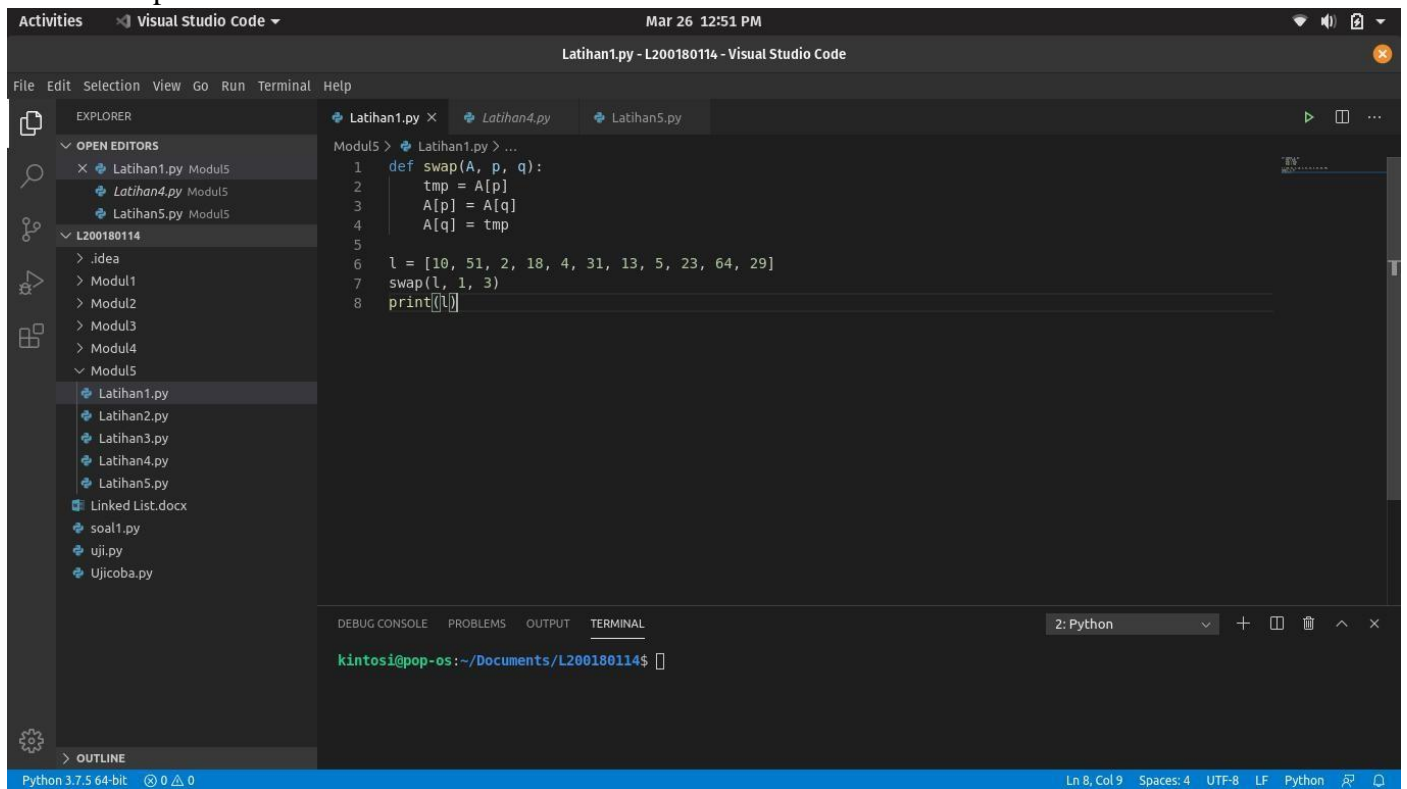


Nama : Karina Muslimah
NIM : L200180138
Kelas : E

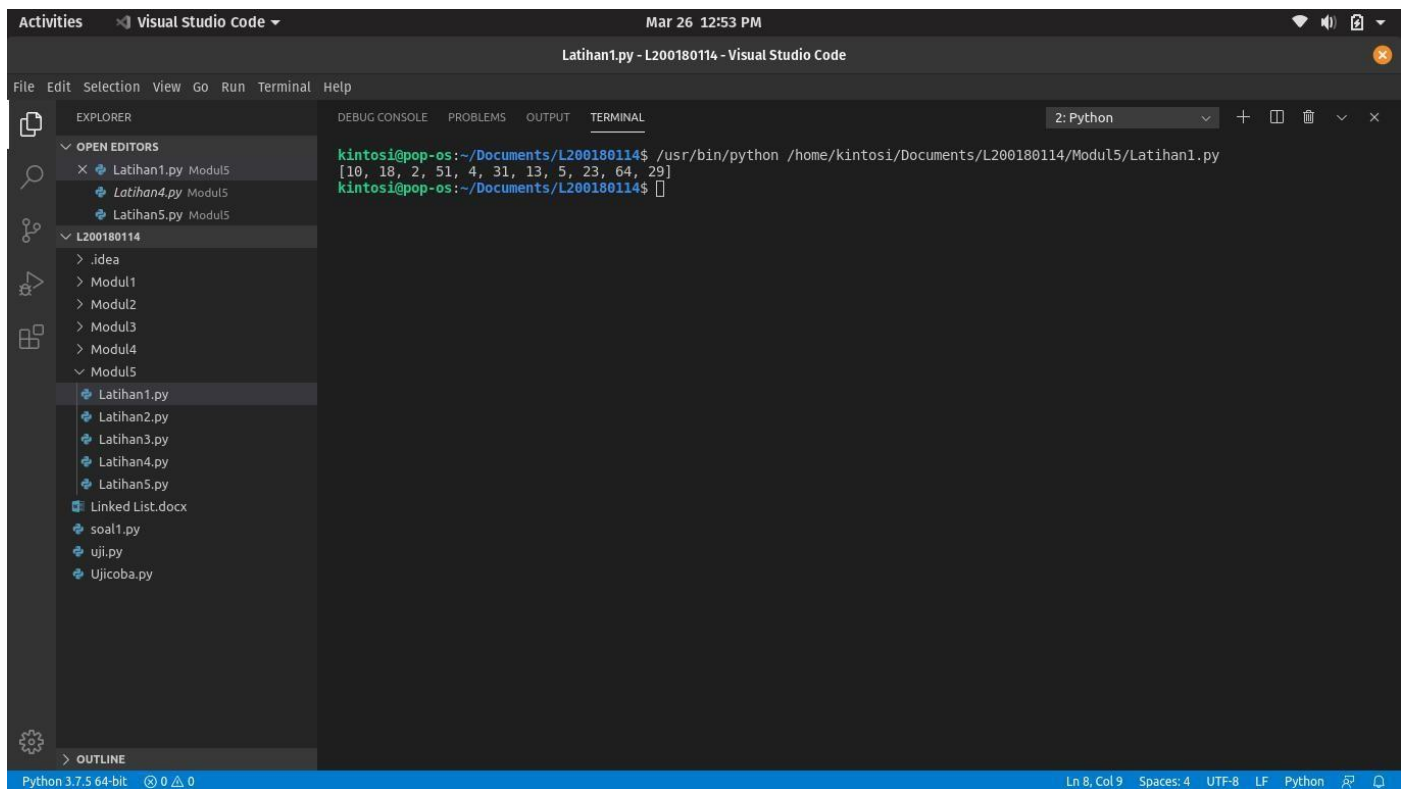
routine swap



The screenshot shows the Visual Studio Code interface with the file explorer on the left and the code editor in the center. The file explorer shows a project named 'L200180114' with a folder 'Modul5' containing files 'Latihan1.py', 'Latihan2.py', 'Latihan3.py', 'Latihan4.py', and 'Latihan5.py'. The code editor displays the content of 'Latihan1.py'.

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

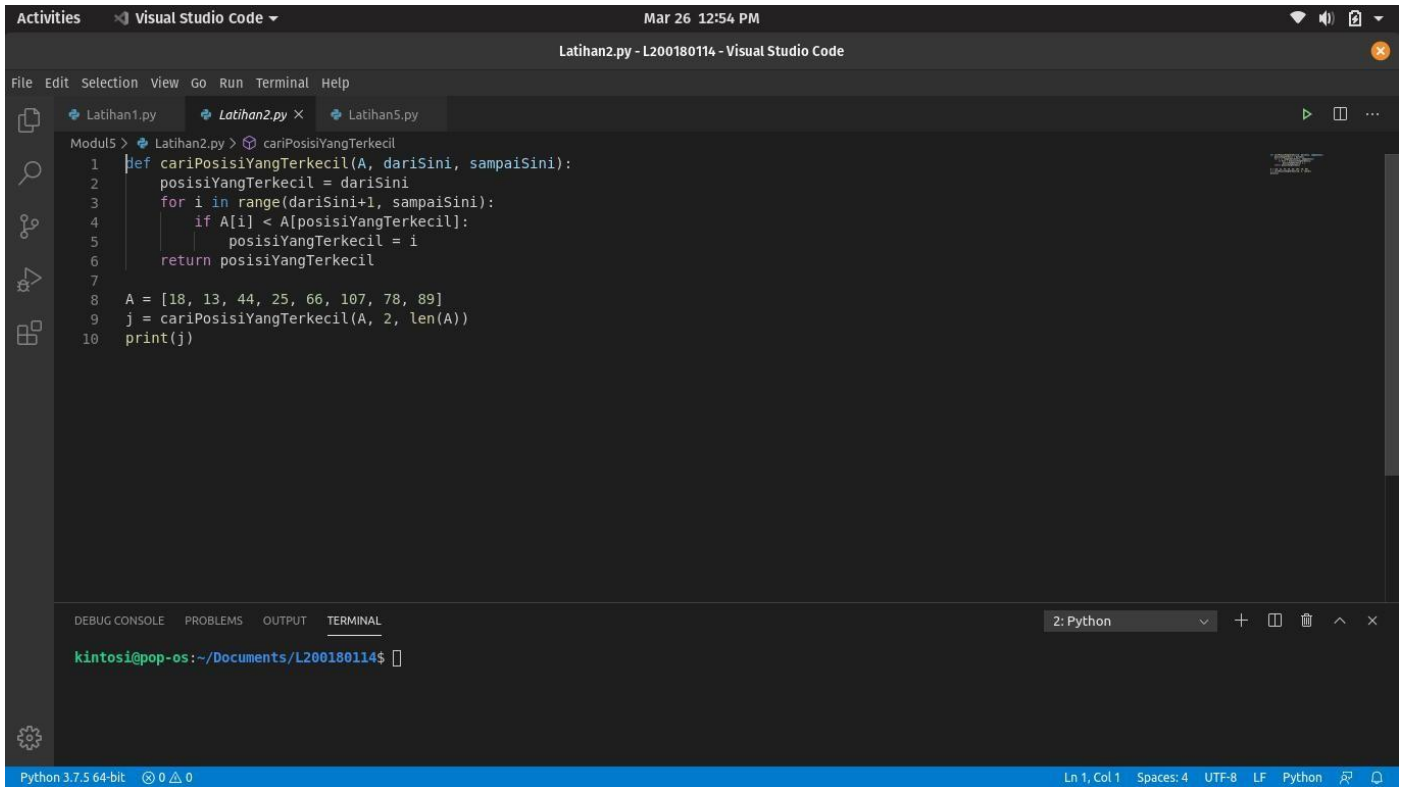
The terminal at the bottom shows the command prompt 'kintosi@pop-os:~/Documents/L200180114\$'.



The screenshot shows the Visual Studio Code interface with the terminal window open. The terminal displays the command to run the Python script and its output.

```
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$
```

Routine untuk mencari nilai terkecil



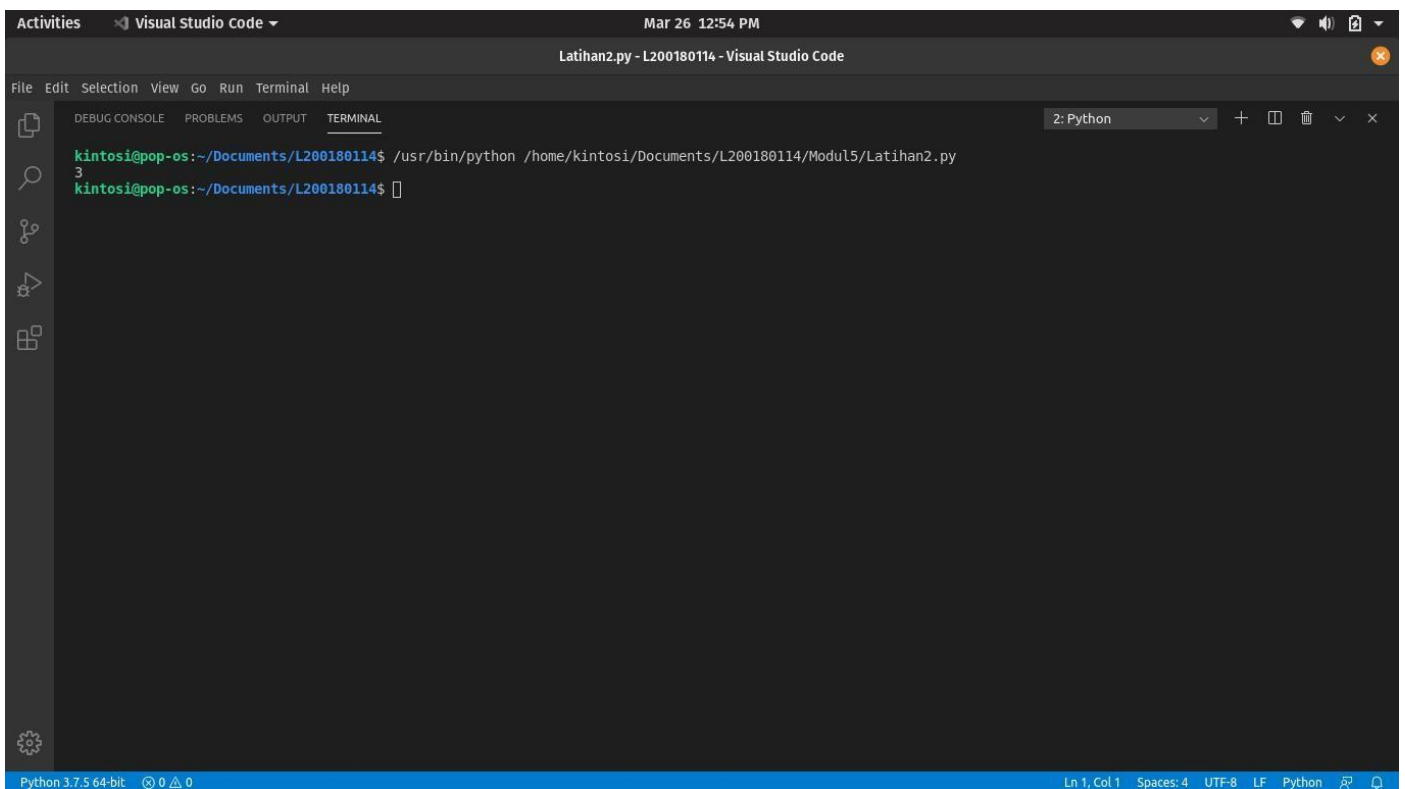
The screenshot shows the Visual Studio Code interface with a Python file named `Latihan2.py` open. The code defines a function `cariPosisiYangTerkecil` that finds the index of the minimum element in a list `A` starting from index `dariSini` up to `sampaiSini`. The list `A` is defined as `[18, 13, 44, 25, 66, 107, 78, 89]`, and the function is called with `cariPosisiYangTerkecil(A, 2, len(A))`. The terminal shows the command prompt `kintosi@pop-os:~/Documents/L200180114$`.

```
Modul5 > 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

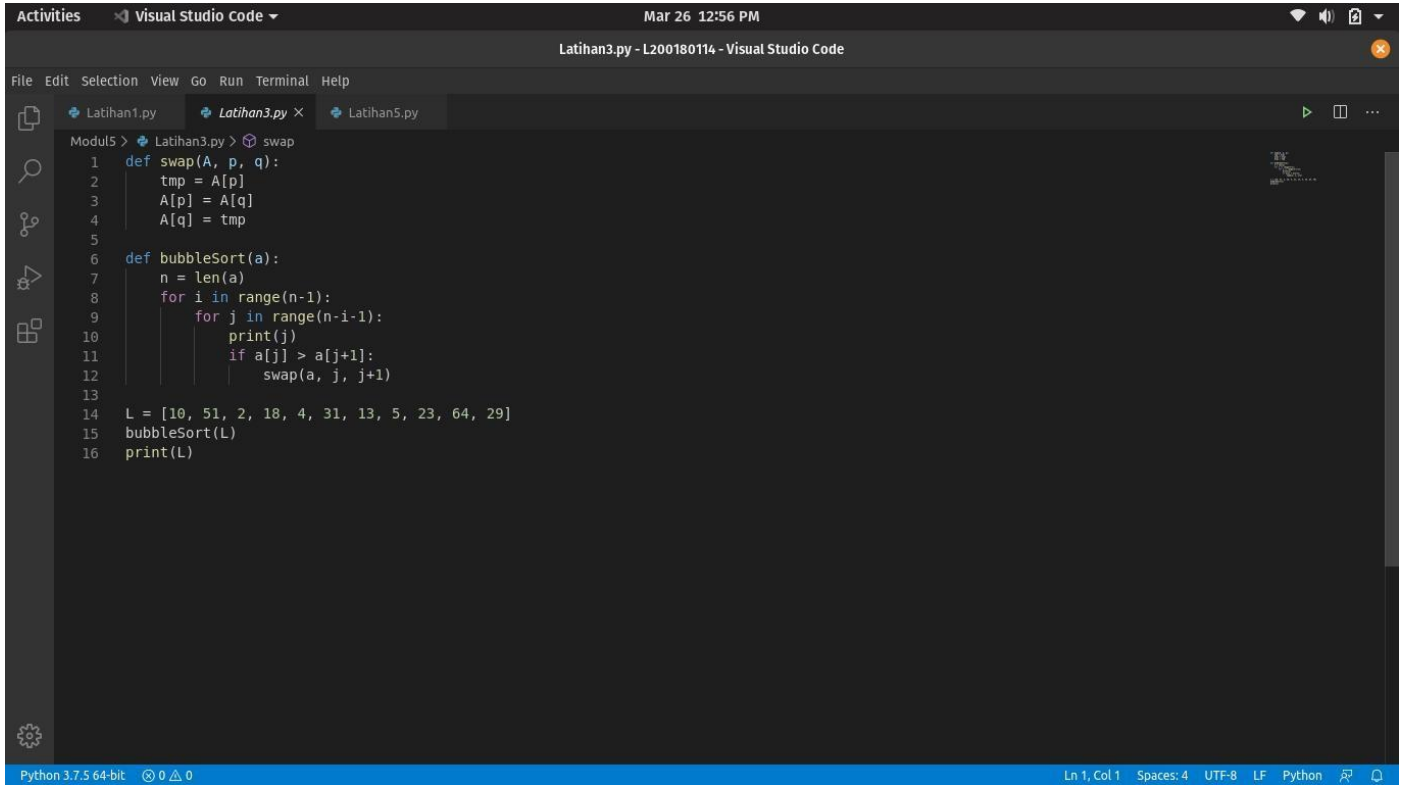


The screenshot shows the Visual Studio Code interface with the terminal open. The command `/usr/bin/python /home/kintosi/Documents/L200180114/Modul5/Latihan2.py` has been executed, and the output `3` is displayed. The terminal shows the command prompt `kintosi@pop-os:~/Documents/L200180114$`.

```
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

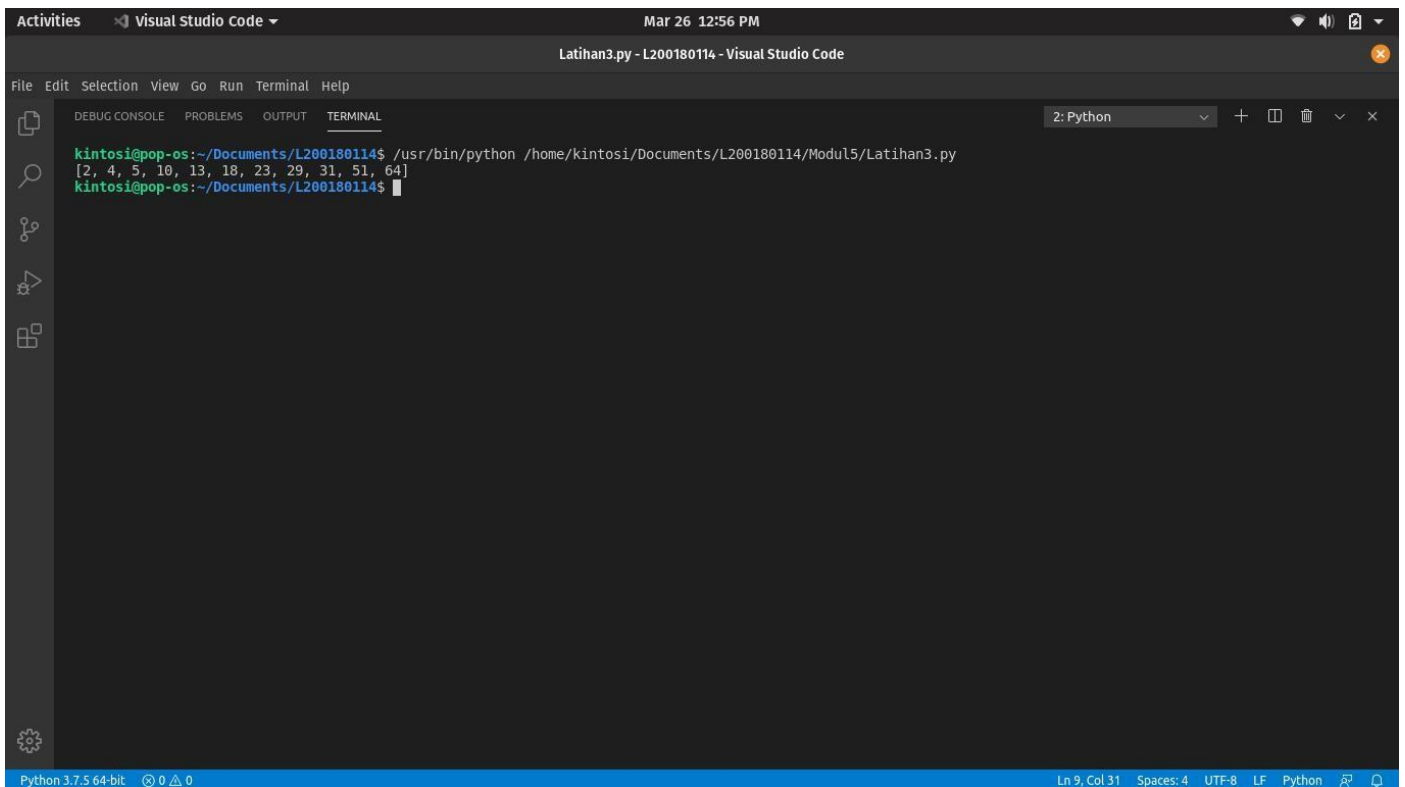
Bubble Sort



The screenshot shows the Visual Studio Code editor with a file named 'Latihan3.py'. The code implements a bubble sort algorithm. It includes a 'swap' function to exchange elements at indices 'p' and 'q' in a list 'A'. The 'bubbleSort' function 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

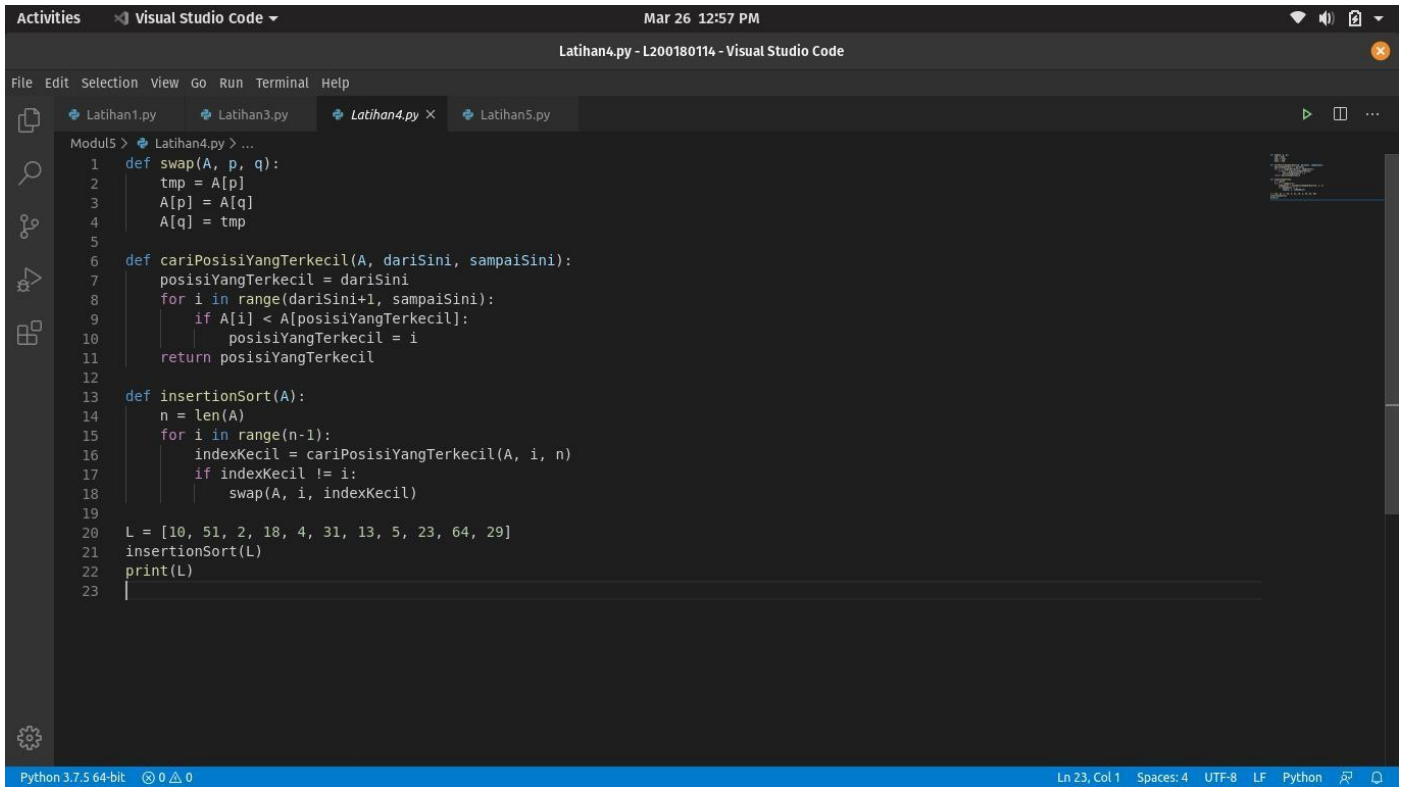


The screenshot shows the terminal output of the Python script. The command executed is `/usr/bin/python /home/kintosi/Documents/L200180114/Modul5/Latihan3.py`. The output displays the list of numbers after the first pass of the bubble sort: `[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]`.

```
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$
```

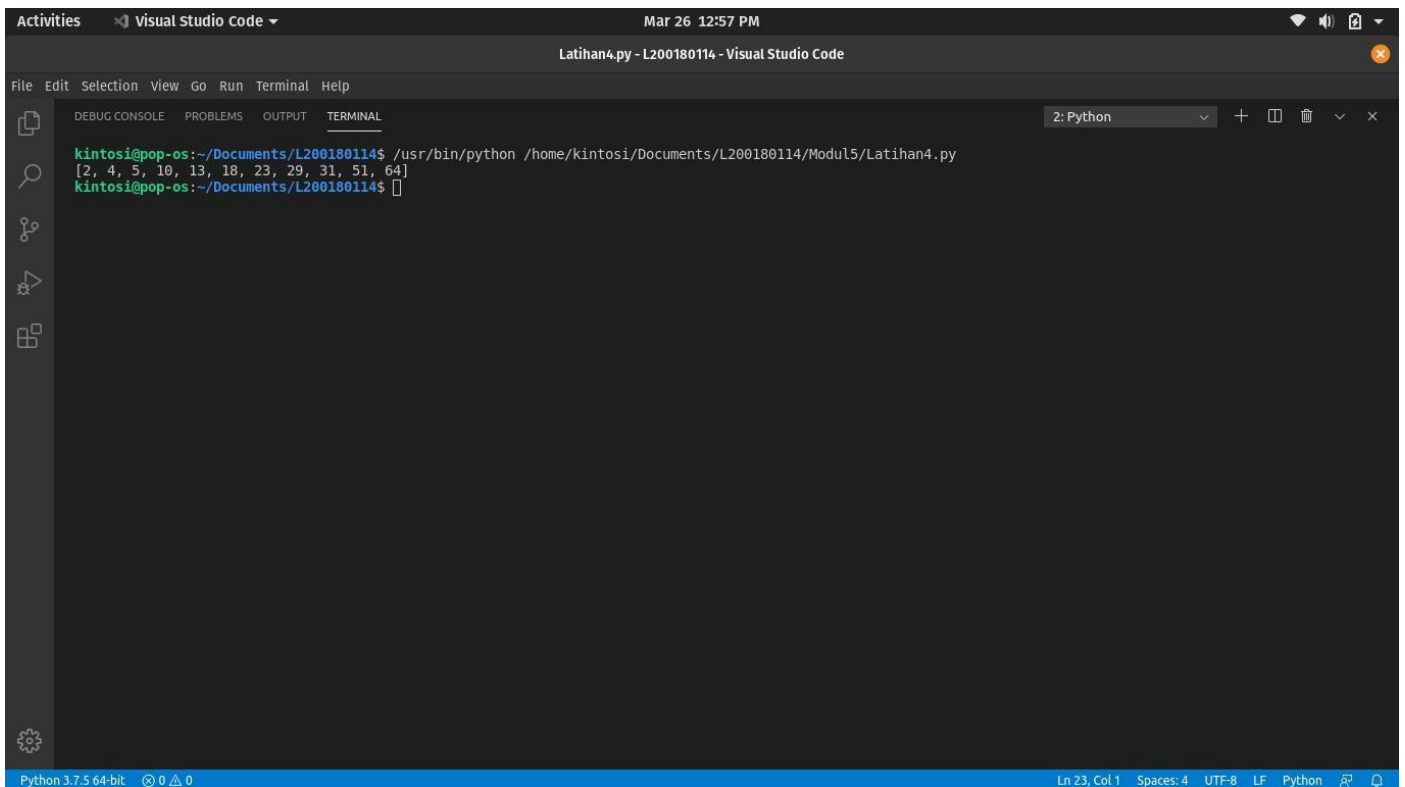
Python 3.7.5 64-bit

Selection Sort



The screenshot shows the Visual Studio Code editor with a Python file named `Latihan4.py`. The code implements a Selection Sort algorithm. It includes a `swap` function to exchange elements, a `cariPosisiYangTerkecil` function to find the minimum element in a subarray, and an `insertionSort` function that uses these to sort the array. The array `L` is initialized with the values `[10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]`, and the `insertionSort` function is called on it. The final sorted array is printed.

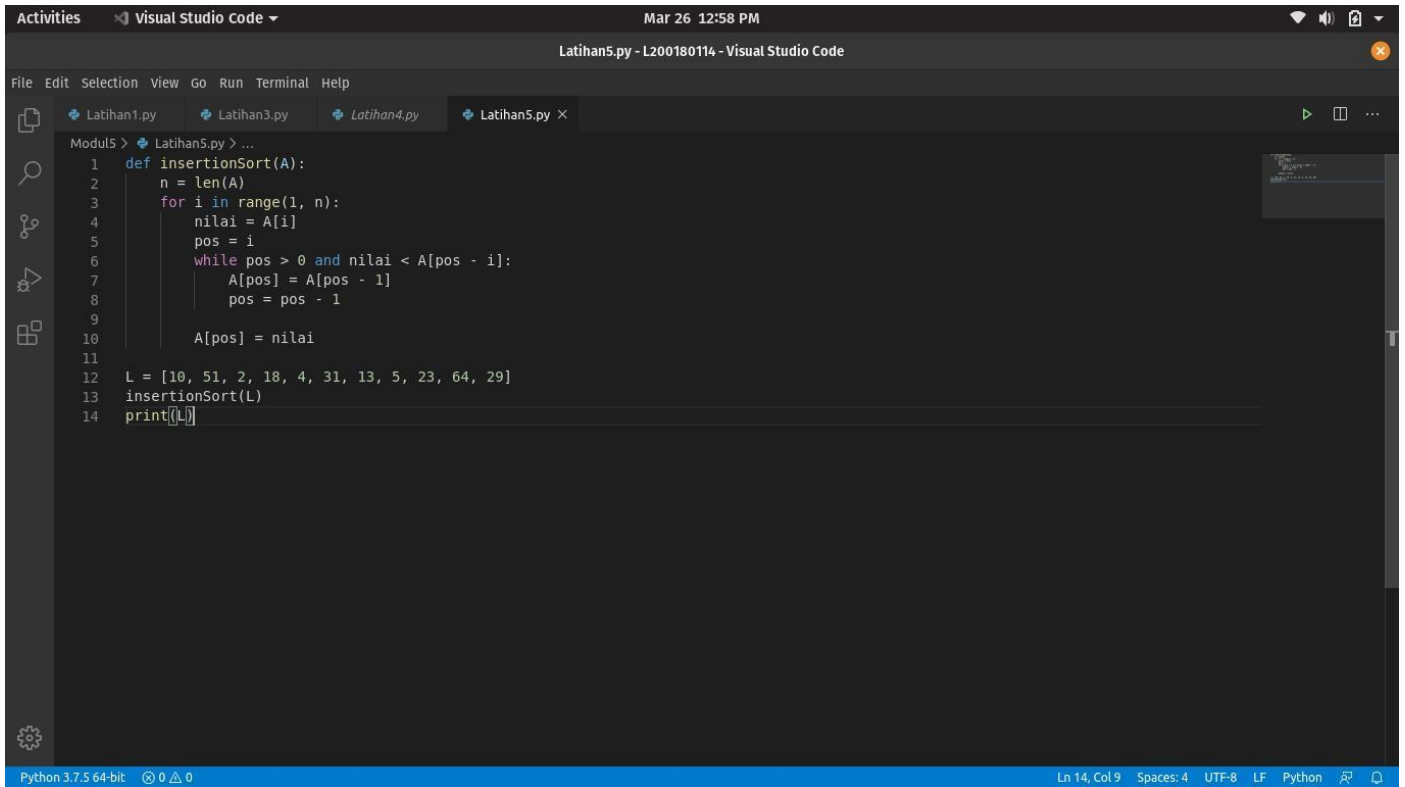
```
Modul5 > Latihan4.py > ...
1  def swap(A, p, q):
2      tmp = A[p]
3      A[p] = A[q]
4      A[q] = tmp
5
6  def cariPosisiYangTerkecil(A, dariSini, sampaiSini):
7      posisiYangTerkecil = dariSini
8      for i in range(dariSini+1, sampaiSini):
9          if A[i] < A[posisiYangTerkecil]:
10             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. It displays the command to run the Python script and the output, which is the sorted array `[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$
```

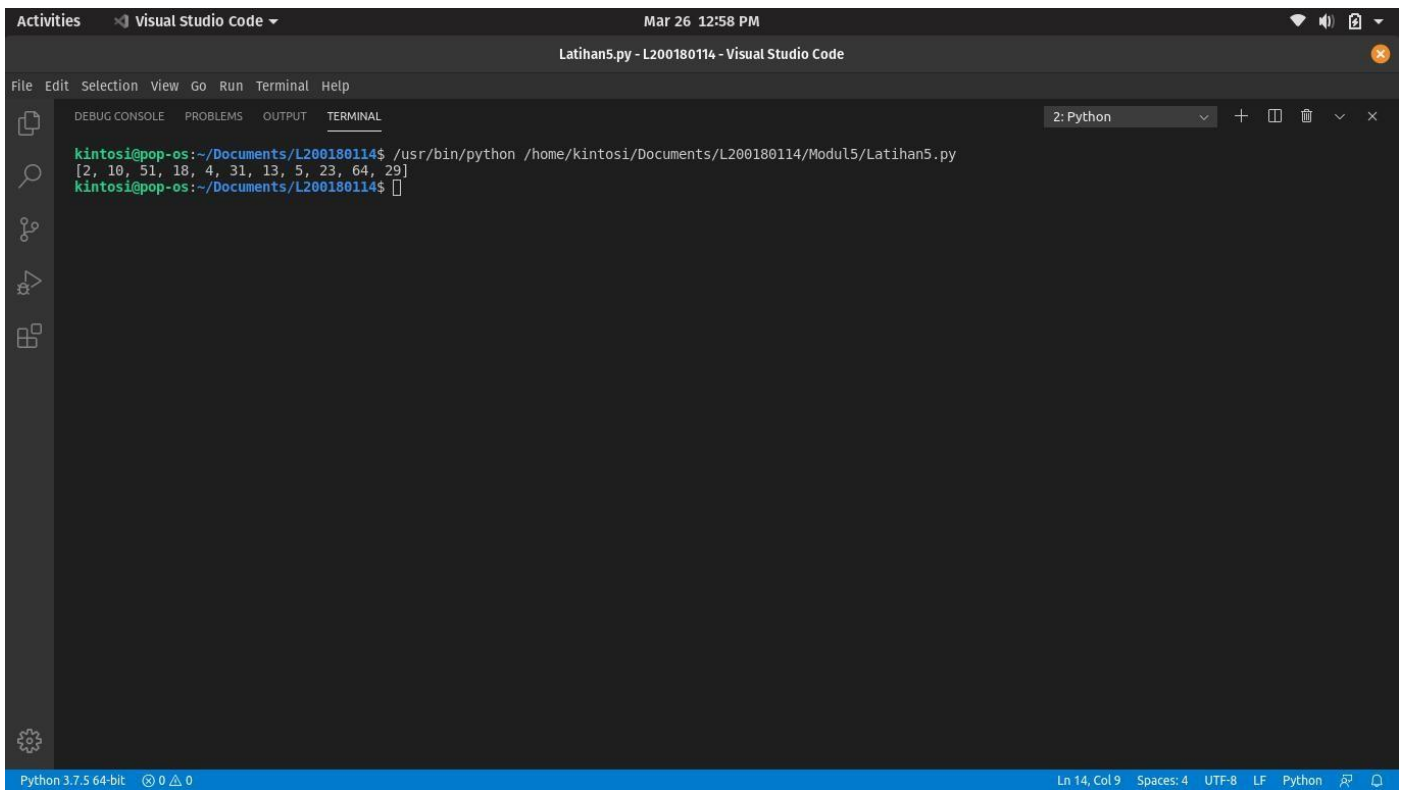
Insertion Sort



The screenshot shows the Visual Studio Code editor with a Python file named 'Latihan5.py'. The code implements the Insertion Sort algorithm. The function 'insertionSort(A)' takes a list 'A' and sorts it in ascending order. It uses a nested loop: an outer loop for 'i' from 1 to 'n-1', and an inner loop for 'pos' from 'i' down to 0, shifting elements to the right until the correct position for 'A[i]' is found. The list 'L' is initialized with the values [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29], and the function is called on it. The final sorted list 'L' is printed.

```
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 - 1]:
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)
```

Python 3.7.5 64-bit | 0 0 0 | Ln 14, Col 9 | Spaces: 4 | UTF-8 | LF | Python



The screenshot shows the terminal window in Visual Studio Code. The command to run the Python script is executed, and the output shows the sorted list [2, 10, 51, 18, 4, 31, 13, 5, 23, 64, 29].

```
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$
```

2: Python | + | - | x |

Python 3.7.5 64-bit | 0 0 0 | Ln 14, Col 9 | Spaces: 4 | UTF-8 | LF | Python