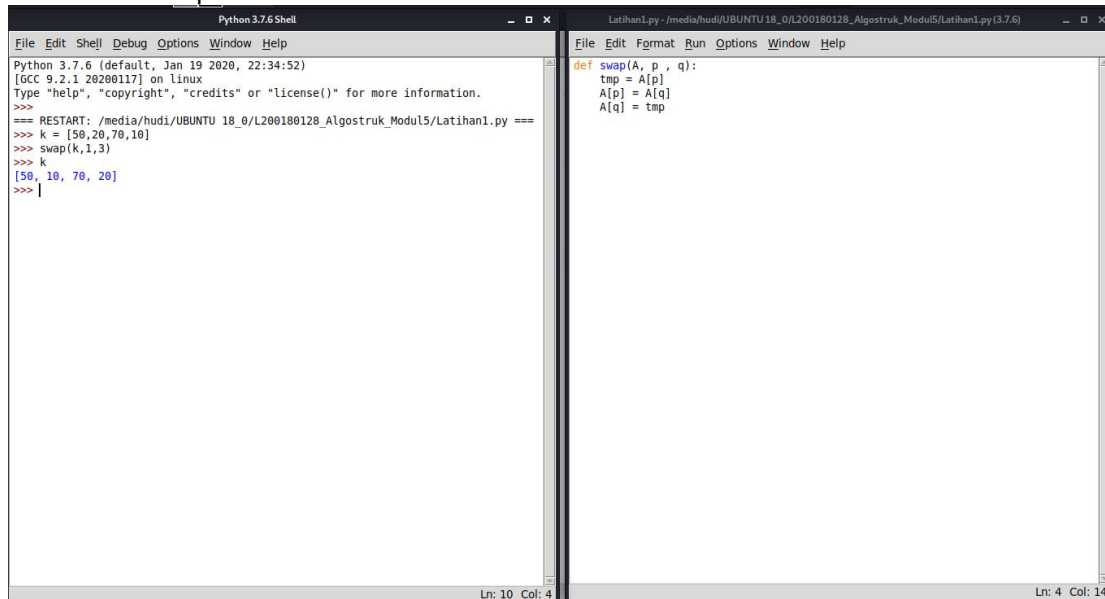


Nama : Hudi Pradjanu
NIM : L200180128
Kelas : E

Latihan 1 Swap

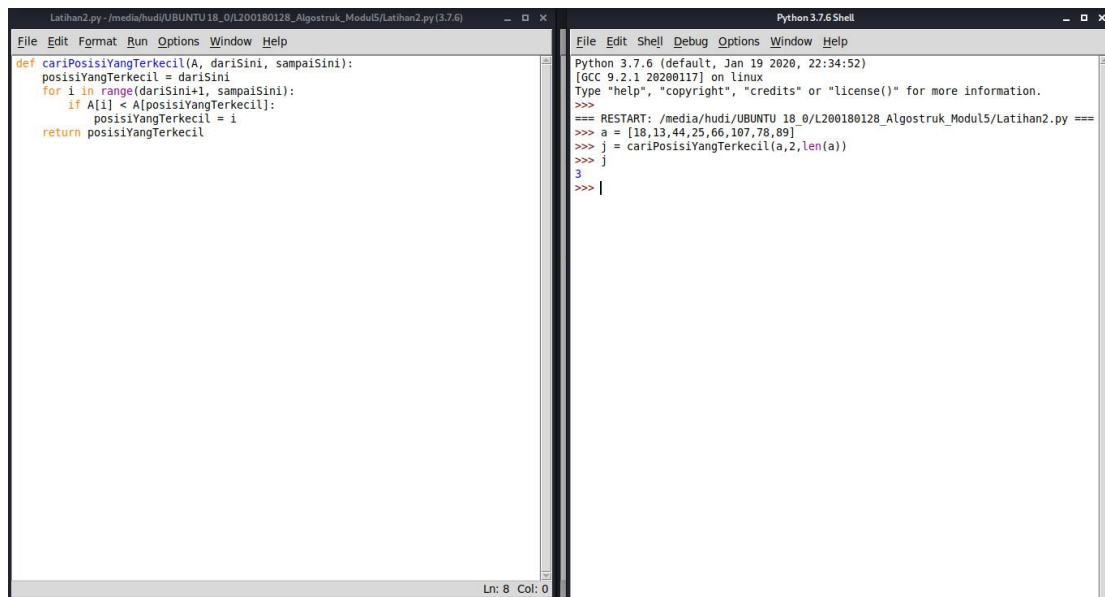


The image shows two side-by-side Python 3.7.6 Shell windows. The left window displays the execution of a script named 'Latihan1.py'. It shows the definition of a swap function and its use to swap elements at indices 1 and 3 of a list [50, 20, 70, 10]. The output shows the list after the swap: [50, 10, 70, 20]. The right window shows the source code of the 'Latihan1.py' file, which defines a swap function that takes a list and two indices, swaps the elements at those indices, and returns the modified list.

```
Python 3.7.6 Shell
File Edit Shell Debug Options Window Help
Python 3.7.6 (default, Jan 19 2020, 22:34:52)
[GCC 9.2.1 20200117] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: /media/hudi/UBUNTU 18_0/L200180128_Algostruk_Modul5/Latihan1.py ===
>>> k = [50,20,70,10]
>>> swap(k,1,3)
>>> k
[50, 10, 70, 20]
>>> |
Ln: 10 Col: 4
```

```
Latihan1.py - /media/hudi/UBUNTU 18_0/L200180128_Algostruk_Modul5/Latihan1.py (3.7.6)
File Edit Format Run Options Window Help
def swap(A, p, q):
    tmp = A[p]
    A[p] = A[q]
    A[q] = tmp
Ln: 4 Col: 14
```

Latihan 2 Mencari data terkecil

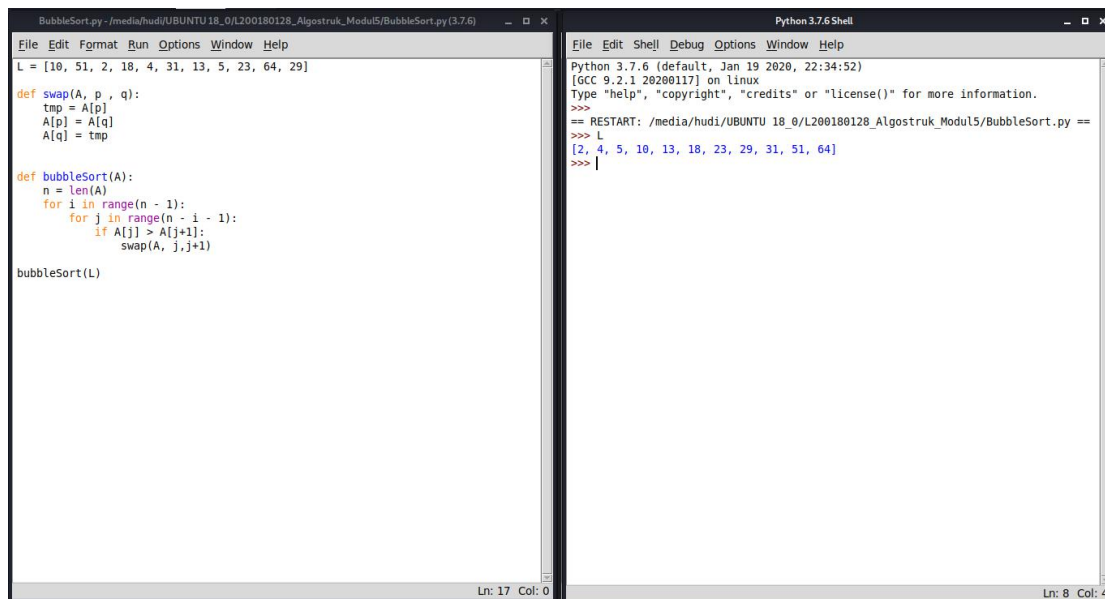


The image shows two side-by-side Python 3.7.6 Shell windows. The left window displays the source code of a function named 'cariPosisiYangTerkecil' which finds the index of the minimum element in a list. The right window shows the execution of this function on a list [18, 13, 44, 25, 66, 107, 78, 89]. The output shows the index of the minimum element, which is 2.

```
Latihan2.py - /media/hudi/UBUNTU 18_0/L200180128_Algostruk_Modul5/Latihan2.py (3.7.6)
File Edit Format Run Options Window Help
def cariPosisiYangTerkecil(A, dariSini, sampaiSini):
    posisiYangTerkecil = dariSini
    for i in range(dariSini+1, sampaiSini):
        if A[i] < A[posisiYangTerkecil]:
            posisiYangTerkecil = i
    return posisiYangTerkecil
Ln: 8 Col: 0
```

```
Python 3.7.6 Shell
File Edit Shell Debug Options Window Help
Python 3.7.6 (default, Jan 19 2020, 22:34:52)
[GCC 9.2.1 20200117] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: /media/hudi/UBUNTU 18_0/L200180128_Algostruk_Modul5/Latihan2.py ===
>>> a = [18,13,44,25,66,107,78,89]
>>> j = cariPosisiYangTerkecil(a,2,len(a))
>>> j
3
>>> |
Ln: 4 Col: 14
```

BubbleSort



The image shows a Python IDE with two windows. The left window, titled 'BubbleSort.py', contains the following code:

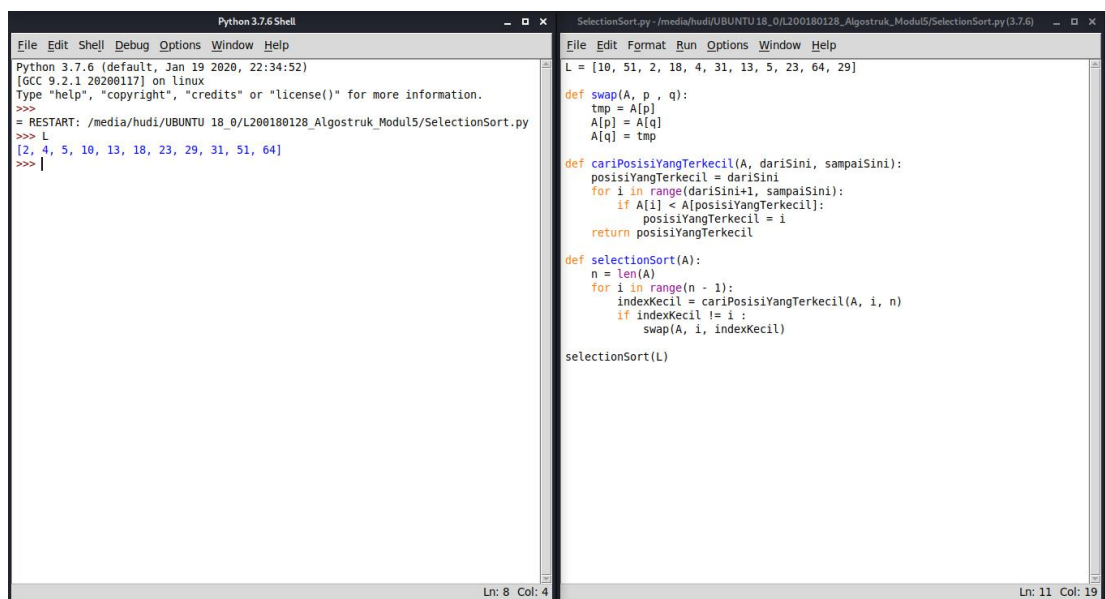
```
def swap(p, q):  
    tmp = A[p]  
    A[p] = A[q]  
    A[q] = tmp  
  
def bubbleSort(A):  
    n = len(A)  
    for i in range(n - 1):  
        for j in range(n - i - 1):  
            if A[j] > A[j+1]:  
                swap(A, j, j+1)  
  
bubbleSort(L)
```

The right window, titled 'Python 3.7.6 Shell', shows the execution of the code:

```
Python 3.7.6 (default, Jan 19 2020, 22:34:52)  
[GCC 9.2.1 20200117] on linux  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
== RESTART: /media/hudi/UBUNTU 18_0/L200180128_Algostruk_Modul5/BubbleSort.py ==  
>>> L  
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]  
>>> |
```

At the bottom of each window, the cursor position is shown: 'Ln: 17 Col: 0' for the left and 'Ln: 8 Col: 4' for the right.

Selectioin Sort



The image shows a Python IDE with two windows. The left window, titled 'Python 3.7.6 Shell', shows the execution of the code:

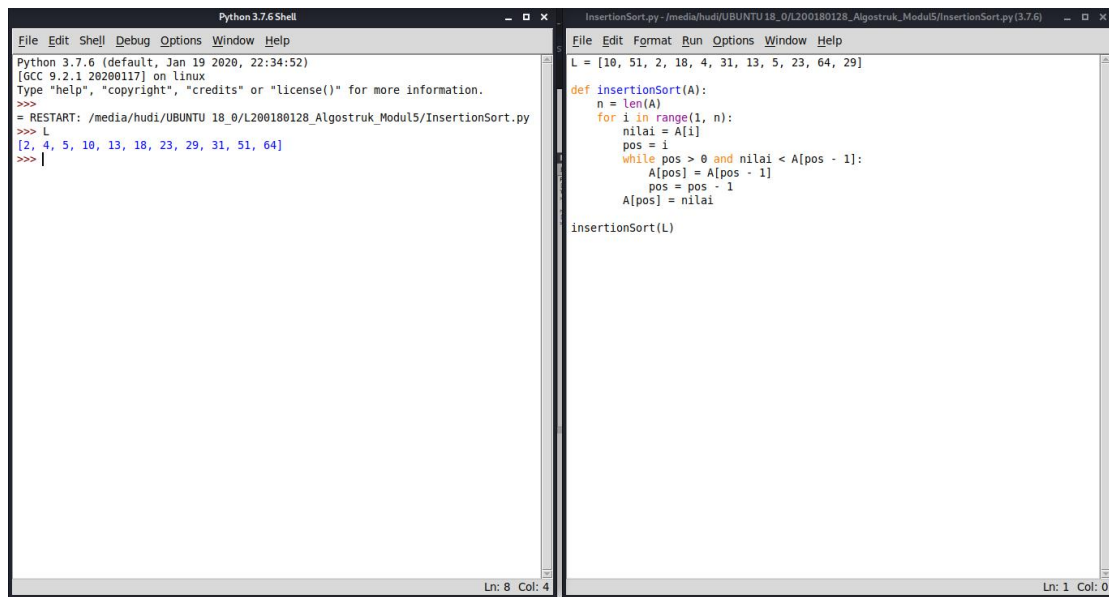
```
Python 3.7.6 (default, Jan 19 2020, 22:34:52)  
[GCC 9.2.1 20200117] on linux  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
== RESTART: /media/hudi/UBUNTU 18_0/L200180128_Algostruk_Modul5/SelectionSort.py ==  
>>> L  
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]  
>>> |
```

The right window, titled 'SelectionSort.py', contains the following code:

```
def cariPosisiYangTerkecil(A, dariSini, sampaiSini):  
    posisiYangTerkecil = dariSini  
    for i in range(dariSini+1, sampaiSini):  
        if A[i] < A[posisiYangTerkecil]:  
            posisiYangTerkecil = i  
    return posisiYangTerkecil  
  
def selectionSort(A):  
    n = len(A)  
    for i in range(n - 1):  
        indexKecil = cariPosisiYangTerkecil(A, i, n)  
        if indexKecil != i:  
            swap(A, i, indexKecil)  
  
selectionSort(L)
```

At the bottom of each window, the cursor position is shown: 'Ln: 8 Col: 4' for the left and 'Ln: 11 Col: 19' for the right.

Insertion Sort



The image shows two side-by-side windows from a Linux environment. The left window is a terminal titled 'Python 3.7.6 Shell' showing the execution of a Python script. The right window is a text editor titled 'InsertionSort.py' showing the source code of the script.

```
Python 3.7.6 Shell
Python 3.7.6 (default, Jan 19 2020, 22:34:52)
[GCC 9.2.1 20200117] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: /media/hudi/UBUNTU 18_0/L200180128_Algostruk_Modul5/InsertionSort.py
>>> L
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
>>> |

Ln: 8 Col: 4
```

```
InsertionSort.py - /media/hudi/UBUNTU 18_0/L200180128_Algostruk_Modul5/InsertionSort.py (3.7.6)
File Edit Format Run Options Window Help
L = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]

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

insertionSort(L)

Ln: 1 Col: 0
```

Tugas 1

```
Python 3.7.6 Shell
File Edit Shell Debug Options Window Help
Python 3.7.6 (default, Jan 19 2020, 22:34:52)
[GCC 9.2.1 20200117] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: /media/hudi/UBUNTU 18_0/L200180128_Algostruk_Modul5/Tugas1.py ====
>>> checkNIM(Daftar)
NIM : 10 Nama : Ika Kota Tinggal : Sukoharjo
NIM : 51 Nama : Budi Kota Tinggal : Sragen
NIM : 2 Nama : Ahmad Kota Tinggal : Surakarta
NIM : 18 Nama : Chandra Kota Tinggal : Surakarta
NIM : 4 Nama : Eka Kota Tinggal : Boyolali
NIM : 31 Nama : Fandi Kota Tinggal : Salatiga
NIM : 13 Nama : Deni Kota Tinggal : Klaten
NIM : 5 Nama : Galuh Kota Tinggal : Wonogiri
NIM : 23 Nama : Janto Kota Tinggal : Klaten
NIM : 64 Nama : Hasan Kota Tinggal : Karanganyar
NIM : 29 Nama : Khalid Kota Tinggal : Purwodadi
>>> sortNIM(Daftar)
>>> checkNIM(Daftar)
NIM : 2 Nama : Ahmad Kota Tinggal : Surakarta
NIM : 4 Nama : Eka Kota Tinggal : Boyolali
NIM : 5 Nama : Galuh Kota Tinggal : Wonogiri
NIM : 10 Nama : Ika Kota Tinggal : Sukoharjo
NIM : 13 Nama : Deni Kota Tinggal : Klaten
NIM : 18 Nama : Chandra Kota Tinggal : Surakarta
NIM : 23 Nama : Janto Kota Tinggal : Klaten
NIM : 29 Nama : Khalid Kota Tinggal : Purwodadi
NIM : 31 Nama : Fandi Kota Tinggal : Salatiga
NIM : 51 Nama : Budi Kota Tinggal : Sragen
NIM : 64 Nama : Hasan Kota Tinggal : Karanganyar
>>> |

Tugas1.py - /media/hudi/UBUNTU 18_0/L200180128_Algostruk_Modul5/Tugas1.py (3.7.6)
File Edit Format Run Options Window Help
self.nama = nama
self.NIM = NIM
self.kotaTinggal = kota
self.uangSaku = us

c0 = MhsTIF('Ika',10,'Sukoharjo', 240000)
c1 = MhsTIF('Budi',51,'Sragen', 230000)
c2 = MhsTIF('Ahmad',2,'Surakarta', 250000)
c3 = MhsTIF('Chandra',18,'Surakarta', 235000)
c4 = MhsTIF('Eka',4,'Boyolali', 240000)
c5 = MhsTIF('Fandi',31,'Salatiga', 250000)
c6 = MhsTIF('Deni',13,'Klaten', 245000)
c7 = MhsTIF('Galuh',5,'Wonogiri', 245000)
c8 = MhsTIF('Janto',23,'Klaten', 245000)
c9 = MhsTIF('Hasan',64,'Karanganyar', 270000)
c10 = MhsTIF('Khalid',29,'Purwodadi', 265000)

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

#Nomor 1
def swap(A, p, q):
    tmp = A[p]
    A[p] = A[q]
    A[q] = tmp

def sortNIM(daftar):
    n = len(daftar)
    for i in range(n-1):
        for j in range(n-i-1):
            if daftar[j].NIM > daftar[j+1].NIM:
                swap(daftar, j, j+1)

def checkNIM(a):
    n = len(a)
    for i in a:
        print('NIM : {} Nama : {} Kota Tinggal : {}'.format(i.NIM, i.nama, i.kota))

Ln: 31 Col: 4
Ln: 16 Col: 17
```

Tugas 2

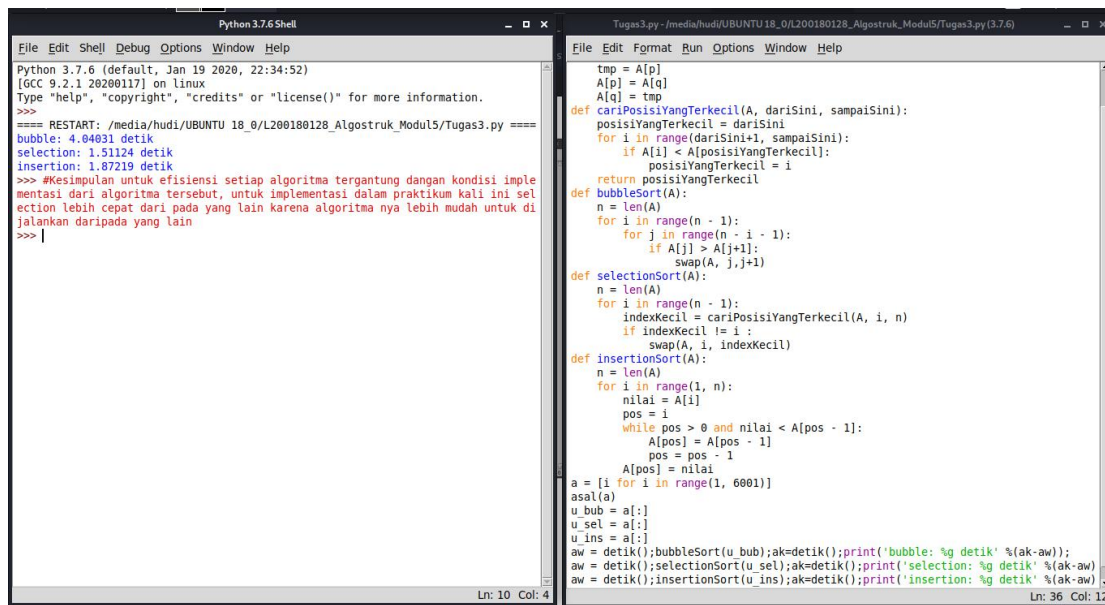
```
Python 3.7.6 Shell
File Edit Shell Debug Options Window Help
Python 3.7.6 (default, Jan 19 2020, 22:34:52)
[GCC 9.2.1 20200117] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: /media/hudi/UBUNTU 18_0/L200180128_Algostruk_Modul5/Tugas2.py ====
>>> A
[2, 34, 56, 78, 89, 190]
>>> B
[1, 4, 12, 23, 36, 76, 120]
>>> C = sortToC(A,B)
>>> C
[1, 2, 4, 12, 23, 34, 36, 56, 76, 78, 89, 120, 190]
>>> |

Tugas2.py - /media/hudi/UBUNTU 18_0/L200180128_Algostruk_Modul5/Tugas2.py (3.7.6)
File Edit Format Run Options Window Help
A = [2,34,56,78,89,190]
B = [1,4,12,23,36,76,120]

def sortToC(a, b):
    c = a+b
    for i in range(1, len(c)):
        nilai = c[i]
        pos = i
        while pos > 0 and nilai < c[pos - 1]:
            c[pos] = c[pos-1]
            pos -= 1
        c[pos] = nilai
    return c

Ln: 13 Col: 4
Ln: 11 Col: 19
```

Tugas 3



The image shows two side-by-side terminal windows. The left window is a Python 3.7.6 Shell, and the right window is a file editor showing the code for 'Tugas3.py'.

Left Window (Python 3.7.6 Shell):

```
Python 3.7.6 Shell
File Edit Shell Debug Options Window Help
Python 3.7.6 (default, Jan 19 2020, 22:34:52)
[GCC 9.2.1 20200117] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: /media/hudi/UBUNTU 18_0/L200180128_Algostruk_Modul5/Tugas3.py ====
bubble: 4.04031 detik
selection: 1.51124 detik
insertion: 1.87219 detik
>>> #Kesimpulan untuk efisiensi setiap algoritma tergantung dengan kondisi imple
mentasi dari algoritma tersebut, untuk implementasi dalam praktikum kali ini sel
ection lebih cepat dari pada yang lain karena algoritma nya lebih mudah untuk di
jalankan daripada yang lain
>>> |
```

Right Window (Tugas3.py):

```
Tugas3.py - /media/hudi/UBUNTU 18_0/L200180128_Algostruk_Modul5/Tugas3.py (3.7.6)
File Edit Format Run Options Window Help

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(A):
    n = len(A)
    for i in range(n - 1):
        for j in range(n - i - 1):
            if A[j] > A[j+1]:
                swap(A, j, j+1)
def selectionSort(A):
    n = len(A)
    for i in range(n - 1):
        indexKecil = cariPosisiYangTerkecil(A, i, n)
        if indexKecil != i:
            swap(A, i, indexKecil)
def insertionSort(A):
    n = len(A)
    for i in range(1, n):
        nilai = A[i]
        pos = i
        while pos > 0 and nilai < A[pos - 1]:
            A[pos] = A[pos - 1]
            pos = pos - 1
        A[pos] = nilai
a = [i for i in range(1, 6001)]
asal(a)
u_bub = a[:]
u_sel = a[:]
u_ins = a[:]
aw = detik();bubbleSort(u_bub);ak=detik();print('bubble: %g detik' %(ak-aw));
aw = detik();selectionSort(u_sel);ak=detik();print('selection: %g detik' %(ak-aw));
aw = detik();insertionSort(u_ins);ak=detik();print('insertion: %g detik' %(ak-aw))
```