

PRAKTIKUM 5

Nama : Fayza Noval Fausta

NIM : G.231.22.0102

Kelas : TI SORE B2

Mata Kuliah : Struktur Data

Penjelasan Algoritma Bubble Sort, Selection Sort, dan Insertion Sort

A. Bubble Sort

CODING

```
def bubSort(array):
    n = len(array)
    swapped = False

    for i in range(n - 1):
        for j in range(0, n - i - 1):
            if array[j] > array[j + 1]:
                swapped = True
                array[j], array[j + 1] = array[j + 1], array[j]

        if not swapped:
            break

MyListArray = [64, 34, 25, 12, 22, 11, 90]
bubSort(MyListArray)

print("The result from sorted array is:")
for i in range(len(MyListArray)):
    print("%d" % MyListArray[i], end=" ")
```

ALGORITMA

1. Melakukan looping untuk mengiterasi sebanyak n-1 kali
2. Melakukan looping untuk membandingkan dan menukar elemen
3. Membandingkan elemen pada indeks j dan j+1
4. Jika diketahui lebih besar, maka akan melakukan pertukaran
5. Lalu cek apakah ada pertukaran selama iterasi
6. Jika tidak ada pertukaran, array sudah terurut, lalu iterasi dihentikan
7. Untuk melakukan pengujian terhadap bubbleSort
8. Menampilkan array yang sudah terurut

B. Selection Sort

CODING

```
import sys

MyListArray = [64, 25, 12, 22, 11]

for i in range(len(MyListArray)):
    min_idx = i
```

```

for j in range(i + 1, len(MyListArray)):
    if MyListArray[min_idx] > MyListArray[j]:
        min_idx = j

MyListArray[i], MyListArray[min_idx] = MyListArray[min_idx], MyListArray[i]

print("The result from sorted array is:")
for i in range(len(MyListArray)):
    print("%d" % MyListArray[i], end=" ")

```

ALGORITMA

1. Melakukan looping untuk mengiterasi sebanyak panjang array
2. Menginisialisasi indeks minimum sebagai i
3. Melakukan looping untuk mencari elemen minimum dari sisa array
4. Membandingkan elemen pada indeks min_idx dan j
5. Jika elemen pada indeks j lebih kecil, update min_idx
6. Melakukan pertukaran elemen untuk menempatkan elemen minimum pada posisi yang benar
7. Menampilkan array yang sudah terurut

C. Insertion Sort

CODING

```

def insertionSort(myArray):
    for i in range(1, len(myArray)):

        key = myArray[i]
        j = i - 1

        while j >= 0 and key < myArray[j]:
            myArray[j + 1] = myArray[j]
            j -= 1

        myArray[j + 1] = key

myArray = [12, 11, 13, 5, 6]
insertionSort(myArray)

for i in range(len(myArray)):
    print("%d" % myArray[i])

```

ALGORITMA

1. Melakukan looping untuk mengiterasi sebanyak panjang array, dimulai dari indeks ke-1
2. Menyimpan nilai saat ini ke dalam key
3. Menginisialisasi indeks j sebagai indeks sebelumnya dari i
4. Melakukan looping untuk memindahkan elemen-elemen yang lebih besar dari key ke kanan
5. Menempatkan key pada posisi yang benar
6. Untuk menguji insertionSort

7. Menampilkan array yang sudah terurut