

210411100085_Modul5_Sorting

May 10, 2022

1 Konsep

1.1 Selection Sort (Kode Soal = 1)

Selection sort merupakan algoritma pengurutan yang prosesnya adalah mencari data yang memenuhi syarat yang sudah ditentukan kemudian data tersebut akan diletakkan pada indeks yang telah ditentukan.

Langkah selection sort :

- Maka data bisa dicari data terkecil terlebih dahulu dengan cara membandingkan kedua buah data.
- Setelah itu data akan dipindah ke indeks minimal bila menggunakan ascending sort.
- Proses selanjutnya adalah mengulangi proses diatas tetapi indeks minimal akan ditambah 1 karena indeks minimal sebelumnya digunakan untuk menampung hasil dari nilai terkecil.
- Ulangi langkah diatas sampai data terurut.

"""

*Misal ada 3 data di dalam list
yang akan diurutkan menggunakan selection sort ascending*

1 -> [3,6,5]

2 -> [3,5,6]

1.) Cari data terkecil dengan membandingkan indeks 0 dan 1 yaitu 3 dan 6.

2.) Kemudian tampung indeks dari nilai terkecil.

3.) Bandingkan indeks dari nilai terkecil yaitu 0 dengan indeks ke 2.

4.) Setelah selesai kemudian pindah nilai dari indeks nilai terkecil ke indeks minimal, sesudah

5.) Setelah ditemukan indeks dari nilai terkecil yaitu indeks ke 2, kemudian pindah nilai dari

6.) Data sudah terurut karena iterasinya sudah selesai.

"""

2 Implementasi-1

2.1 Kode Soal 1

```
[4]: def bubbleSortDescending(nums):
    print("\n\nData yang akan diurutkan = ", nums)
    no = 1
    for i in range(len(nums) - 1, 0, -1):

        print("\nOuter Loop #", no)
        for j in range(len(nums) - 1, len(nums) - i - 1, -1):
            check = nums[j] > nums[j - 1]
            if check:
                nums[j], nums[j - 1] = nums[j - 1], nums[j]
            print("is data[%d] > data[%d] ? %s , Data -> %s " %
                  (j, j - 1, check, nums))

        no += 1

    return nums

lstNums = [10, 9, 1, 3, 8, 7, 11]
print("\n\nSorted Data ", bubbleSortDescending(lstNums))
```

Data yang akan diurutkan = [10, 9, 1, 3, 8, 7, 11]

Outer Loop # 1

```
is data[6] > data[5] ? True , Data -> [10, 9, 1, 3, 8, 11, 7]
is data[5] > data[4] ? True , Data -> [10, 9, 1, 3, 11, 8, 7]
is data[4] > data[3] ? True , Data -> [10, 9, 1, 11, 3, 8, 7]
is data[3] > data[2] ? True , Data -> [10, 9, 11, 1, 3, 8, 7]
is data[2] > data[1] ? True , Data -> [10, 11, 9, 1, 3, 8, 7]
is data[1] > data[0] ? True , Data -> [11, 10, 9, 1, 3, 8, 7]
```

Outer Loop # 2

```
is data[6] > data[5] ? False , Data -> [11, 10, 9, 1, 3, 8, 7]
is data[5] > data[4] ? True , Data -> [11, 10, 9, 1, 8, 3, 7]
is data[4] > data[3] ? True , Data -> [11, 10, 9, 8, 1, 3, 7]
is data[3] > data[2] ? False , Data -> [11, 10, 9, 8, 1, 3, 7]
is data[2] > data[1] ? False , Data -> [11, 10, 9, 8, 1, 3, 7]
```

Outer Loop # 3

```
is data[6] > data[5] ? True , Data -> [11, 10, 9, 8, 1, 7, 3]
is data[5] > data[4] ? True , Data -> [11, 10, 9, 8, 7, 1, 3]
is data[4] > data[3] ? False , Data -> [11, 10, 9, 8, 7, 1, 3]
```

```
is data[3] > data[2] ? False , Data -> [11, 10, 9, 8, 7, 1, 3]
```

Outer Loop # 4

```
is data[6] > data[5] ? True , Data -> [11, 10, 9, 8, 7, 3, 1]
is data[5] > data[4] ? False , Data -> [11, 10, 9, 8, 7, 3, 1]
is data[4] > data[3] ? False , Data -> [11, 10, 9, 8, 7, 3, 1]
```

Outer Loop # 5

```
is data[6] > data[5] ? False , Data -> [11, 10, 9, 8, 7, 3, 1]
is data[5] > data[4] ? False , Data -> [11, 10, 9, 8, 7, 3, 1]
```

Outer Loop # 6

```
is data[6] > data[5] ? False , Data -> [11, 10, 9, 8, 7, 3, 1]
```

Sorted Data [11, 10, 9, 8, 7, 3, 1]

3 Implementasi-2

```
[43]: def bubbleSortModified(nums):
    print("\n\nData yang akan diurutkan = ", nums)
    no = 1

    for i in range(len(nums) - 1):
        numsSorted = 0
        totalIteration = len(nums) - 1 - i

        print("\nIterasi ke- %d jumlah iterasi %d" % (no, totalIteration))

        for j in range(totalIteration):
            check = nums[j] > nums[j + 1]
            if check:
                nums[j], nums[j + 1] = nums[j + 1], nums[j]
                numsSorted += 1
            print("%d = %s " % (j + 1, nums))

        if numsSorted == 0:
            break

        no += 1

    return nums

lstNums = [10, 2, 5, 8, 1, 20, 2, 2, 4]
print("\n\nData Urut ", bubbleSortModified(lstNums))
```

Data yang akan diurutkan = [10, 2, 5, 8, 1, 20, 2, 2, 4]

Iterasi ke- 1 jumlah iterasi 8

1 = [2, 10, 5, 8, 1, 20, 2, 2, 4]
2 = [2, 5, 10, 8, 1, 20, 2, 2, 4]
3 = [2, 5, 8, 10, 1, 20, 2, 2, 4]
4 = [2, 5, 8, 1, 10, 20, 2, 2, 4]
5 = [2, 5, 8, 1, 10, 20, 2, 2, 4]
6 = [2, 5, 8, 1, 10, 2, 20, 2, 4]
7 = [2, 5, 8, 1, 10, 2, 2, 20, 4]
8 = [2, 5, 8, 1, 10, 2, 2, 4, 20]

Iterasi ke- 2 jumlah iterasi 7

1 = [2, 5, 8, 1, 10, 2, 2, 4, 20]
2 = [2, 5, 8, 1, 10, 2, 2, 4, 20]
3 = [2, 5, 1, 8, 10, 2, 2, 4, 20]
4 = [2, 5, 1, 8, 10, 2, 2, 4, 20]
5 = [2, 5, 1, 8, 2, 10, 2, 4, 20]
6 = [2, 5, 1, 8, 2, 2, 10, 4, 20]
7 = [2, 5, 1, 8, 2, 2, 4, 10, 20]

Iterasi ke- 3 jumlah iterasi 6

1 = [2, 5, 1, 8, 2, 2, 4, 10, 20]
2 = [2, 1, 5, 8, 2, 2, 4, 10, 20]
3 = [2, 1, 5, 8, 2, 2, 4, 10, 20]
4 = [2, 1, 5, 2, 8, 2, 4, 10, 20]
5 = [2, 1, 5, 2, 2, 8, 4, 10, 20]
6 = [2, 1, 5, 2, 2, 4, 8, 10, 20]

Iterasi ke- 4 jumlah iterasi 5

1 = [1, 2, 5, 2, 2, 4, 8, 10, 20]
2 = [1, 2, 5, 2, 2, 4, 8, 10, 20]
3 = [1, 2, 2, 5, 2, 4, 8, 10, 20]
4 = [1, 2, 2, 2, 5, 4, 8, 10, 20]
5 = [1, 2, 2, 2, 4, 5, 8, 10, 20]

Iterasi ke- 5 jumlah iterasi 4

1 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
2 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
3 = [1, 2, 2, 2, 4, 5, 8, 10, 20]
4 = [1, 2, 2, 2, 4, 5, 8, 10, 20]

Data Urut [1, 2, 2, 2, 4, 5, 8, 10, 20]