

Selection Sort

Dikatakan selection sort karena algoritma ini mencoba memilih satu per satu elemen data dari posisi awal, untuk mencari data paling kecil dengan mencatat posisi index-nya saja, lalu dilakukan pertukaran hanya sekali pada akhir setiap tahapan.

- Untuk setiap proses, akan dicari elemen-elemen yang belum diurutkan yang memiliki nilai terkecil (Ascending) atau terbesar (Descending) akan dipertukarkan ke posisi yang tepat di dalam array.
- Misalnya untuk putaran pertama, akan dicari data dengan nilai terkecil dan data ini akan ditempatkan di indeks terkecil (`data[0]`) / data pertama, pada putaran kedua akan dicari data kedua terkecil, dan akan ditempatkan di indeks kedua (`data[1]`) / data kedua atau selanjutnya.
- Selama proses, perbandingan dan pengubahan **hanya dilakukan** pada **indeks** perbandingan saja, pertukaran data secara fisik terjadi pada **akhir** proses

Ilustrasi

| | | | | | | | | | |
|---|----|---|---|---|---|---|---|---|---|
| 3 | 10 | 4 | 6 | 8 | 9 | 7 | 2 | 1 | 5 |
|---|----|---|---|---|---|---|---|---|---|

1 2 3 4 5 6 7 8 9 10

Cek seluruh elemen array,

temukan nilai terkecil (1)

tukarkan posisinya dengan posisi nilai yang tersimpan pada posisi pertama dari array (3)

| | | | | | | | | | |
|---|----|---|---|---|---|---|---|---|---|
| 3 | 10 | 4 | 6 | 8 | 9 | 7 | 2 | 1 | 5 |
|---|----|---|---|---|---|---|---|---|---|

| | | | | | | | | | |
|---|----|---|---|---|---|---|---|---|---|
| 1 | 10 | 4 | 6 | 8 | 9 | 7 | 2 | 3 | 5 |
|---|----|---|---|---|---|---|---|---|---|

Temukan nilai terkecil kedua (2), dan tukarkan posisinya dengan nilai yang berada pada posisi kedua (10).

| | | | | | | | | | |
|---|----|---|---|---|---|---|---|---|---|
| 1 | 10 | 4 | 6 | 8 | 9 | 7 | 2 | 3 | 5 |
|---|----|---|---|---|---|---|---|---|---|

| | | | | | | | | | |
|---|---|---|---|---|---|---|----|---|---|
| 1 | 2 | 4 | 6 | 8 | 9 | 7 | 10 | 3 | 5 |
|---|---|---|---|---|---|---|----|---|---|

Dua elemen biru pertama tidak akan berubah lagi sebab mereka sudah merupakan nilai terkecil pertama dan kedua dalam array tsb.

ulangi dengan cara/proses “pilih dan tukar”

| | | | | | | | | | |
|---|---|---|---|---|---|---|----|---|---|
| 1 | 2 | 4 | 6 | 8 | 9 | 7 | 10 | 3 | 5 |
| 1 | 2 | 3 | 6 | 8 | 9 | 7 | 10 | 4 | 5 |
| 1 | 2 | 3 | 6 | 8 | 9 | 7 | 10 | 4 | 5 |
| 1 | 2 | 3 | 4 | 8 | 9 | 7 | 10 | 6 | 5 |
| 1 | 2 | 3 | 4 | 8 | 9 | 7 | 10 | 6 | 5 |
| 1 | 2 | 3 | 4 | 5 | 9 | 7 | 10 | 6 | 8 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|----|---|----|
| 1 | 2 | 3 | 4 | 5 | 9 | 7 | 10 | 6 | 8 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 10 | 9 | 8 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 10 | 9 | 8 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 10 | 9 | 8 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Indek key = 1:9

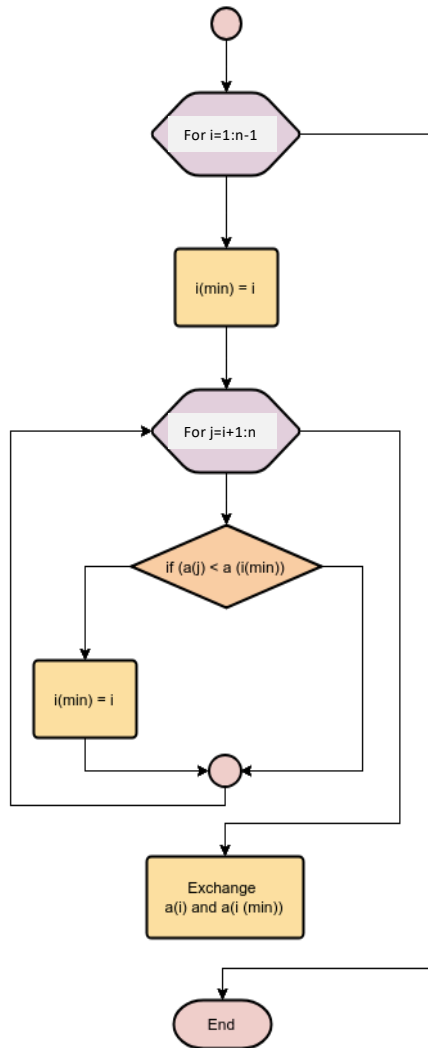
Pembanding : indek setelah indek key

Misal ada n data

indek key = $i=1:n-1$

Pembanding adalah $j=i+1:n$

Source : <https://sisinform-aaf1231072.blogspot.com/2013/02/selection-sort.html>



| | | | | | |
|-----|-------|-------|-------|--|--|
| 1 | Min 1 | | | | |
| 2 | ok | Min 2 | | | |
| 3 | ok | ok 2 | Min 3 | | |
| N-1 | | | | | |

Algoritma Selection Sort (A : array [1..N] of integer) {Diberikan N data kemudian diurutkan naik }

Deklarasi

i, j, k : integer

Deskripsi

for i 1 to N-1 do

k=1

for j : i+1 to N do

if A[j] < A[k] then

k =j

end of

Tukar(A[i], A[k])

end for

Flowchart dan Algoritma

Ingat kembali proses penukaran

Kode Program

Algoritma Selection Sort (A : array [1..N] of integer) {Diberikan N data kemudian diurutkan naik }

Deklarasi

i, j, k : integer

Deskripsi

```
for i 1 to N-1 do
    k=1
    for j : i+1 to N do
        if A[j] < A[k] then
            k =j
        end fof
    Tukar(A[i], A[k])
end for
```

```
def Selection_Sort(A):
    for i in range(len(A)-1):
        min_idx = i
        for j in range(i+1, len(A)):
            if A[min_idx] > A[j]:
                min_idx = j
        # penukaran
        temp=A[i]
        A[i]=A[min_idx]
        A[min_idx]=temp

    print('iterasi', i ,A)
```

Contoh : A=[4,3,5,6,2,78,98]; Jumlah element 7 dengan index 0-6; Sehingga iterasi *i* mulai dari 0-5 dan loop 2 : 1-6

Loop for i=0

Min_index=i=0

Loop for 1 to 6

J=1; cek A[0]:4 > A[1] : 3 True

Min_index=j : 1

J=2; cek A[1]:3 > A[2] : 5 False

J=3; cek A[1]:3 > A[3] : 6 False

J=4; cek A[1]:3 > A[4] : 2 True

Min_index=j : 4

J=5; cek A[4]:2 > A[5] : 78 False

J=6; cek A[4]:2 > A[6] : 98 False

proses penukaran

Temp= A[i]=4

A[i]=A[Min_index]=A[4]=2

A[Min_index]=temp= 4

Hasil iterasi =[2, 3, 5, 6, 4, 78, 98]

Loop for i=1

Min_index=i=1

Loop for 2 to 6

J=2; cek A[1]:3 > A[2] : 5 False

J=3; cek A[1]:3 > A[3] : 6 False

J=4; cek A[1]:3 > A[4] : 4 False

J=5; cek A[1]:3 > A[5] : 78 False

J=6; cek A[1]:3 > A[6] : 98 False

proses penukaran

Temp= A[i]=3

A[i]=A[Min_index]=A[1]=3

A[Min_index]=temp= 3

Hasil iterasi =[2, 3, 5, 6, 4, 78, 98]

Loop for i=2

Min_index=i=2

Loop for 3 to 6

J=3; cek A[2]:5 > A[3] : 6 False

J=4; cek A[2]:5 > A[4] : 4 True

Min_index=j : 4

J=5; cek A[4]:4 > A[5] : 78 False

J=6; cek A[4]:4 > A[6] : 98 False

proses penukaran

Temp= A[i]=5

A[i]=A[Min_index]=A[4]=4

A[Min_index]=temp= 5

Hasil iterasi =[2, 3, 4, 6, 5, 78, 98]

Loop for i=3

Min_index=i=3

Loop for 4 to 6

J=4; cek A[3]:6 > A[4] : 5 True

Min_index=j : 4

J=5; cek A[4]:5 > A[5] : 78 False

J=6; cek A[4]:5 > A[6] : 98 False

proses penukaran

Temp= A[i]=6

A[i]=A[Min_index]=A[4]=5

A[Min_index]=temp= 6

Hasil iterasi =[2, 3, 4, 5, 6, 78, 98]

Loop for i=4

Min_index=i=4

Loop for 5 to 6

J=5; cek A[4]:6 > A[5] : 78 False

J=6; cek A[4]:6 > A[6] : 98 False

proses penukaran

Temp= A[i]=6

A[i]=A[Min_index]=A[4]=6

A[Min_index]=temp= 6

Hasil iterasi =[2, 3, 4, 5, 6, 78, 98]

Loop for i=5

Min_index=i=5

Loop for 5 to 6

J=6; cek A[5]:78 > A[6] : 98 False

proses penukaran

Temp= A[i]=6

A[i]=A[Min_index]=A[6]=98

A[Min_index]=temp= 6

Hasil iterasi =[2, 3, 4, 5, 6, 78, 98]