



Sorting Algorithms

1. Insertion
2. Selection
3. Bubble
4. Shell
5. Quick
6. Merge



Selection Sort

(one of the simplest sorting algorithms)

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

Cek seluruh elemen array, temukan nilai terkecil (1) dan 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
---	----	---	---	---	---	---	---	---	---

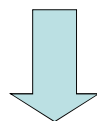
1	2	4	6	8	9	7	10	3	5
---	---	---	---	---	---	---	----	---	---




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

Sekarang, ulangi 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
---	---	---	---	---	---	---	----	---	---

Arna Fariza

Algoritma dan Struktur Data

5



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
---	---	---	---	---	---	---	---	---	----

Arna Fariza

Algoritma dan Struktur Data

6



Algoritma Metode Seleksi

```
1.  $i \leftarrow 0$ 
2. selama ( $i < N-1$ ) kerjakan baris 3 sd 14
3.  $min \leftarrow A[i]$ 
4.  $min\_id \leftarrow i$ 
5.  $j \leftarrow i + 1$ 
6. Selama ( $j < N$ ) kerjakan baris 7 sd 10
7. Jika ( $A[j] < min$ ) kerjakan 8 & 9
8.  $min \leftarrow A[j]$ 
9.  $min\_id \leftarrow j$ 
10.  $j \leftarrow j + 1$ 
11.  $temp \leftarrow A[i]$ 
12.  $A[i] \leftarrow A[min\_id]$ 
13.  $A[min\_id] \leftarrow temp$ 
14.  $i \leftarrow i + 1$ 
```



Pseudo Code

```
SelectionSort(A, n) {
  for i = 0 to n-1 {
    min = i //Assign indeks i sebagai min
    //bandingkan elemen pd indeks j dgn indeks min
    for j = i + 1 to n
      if  $A[j] < A[min]$  //jika elemen j lbh kecil dr elemen min
        min = j //update nilai min menjadi j
      j = j + 1 //Ulangi sampai nilai j sama dgn n
    temp = A[i] //menukarkan 2 elemen :
    A[i] = A[min] // A[i] dengan A[min]
    A[min] = temp
  } //Ulangi sampai nilai i sama dgn n-1
}
```



Selection Sort → Analysis

- Secara umum, yang dilakukan dalam metode seleksi adalah perbandingan key (elemen pada posisi min) serta penukaran elemen
- Sehingga untuk menganalisis metode ini harus dihitung jumlah pembandingannya serta jumlah penukaran elemennya.



Selection Sort → Analysis

- Pada algoritma di atas, loop for terluar dilakukan sebanyak $n-1$ kali
- Pada setiap iterasi, dilakukan satu kali penukaran elemen, sehingga :
 - Total penukaran/swap = $n-1$
 - Total pergeseran = $3 * n-1$
(pada setiap penukaran terjadi 3 x pergeseran)
- Jumlah pembandingan pada metode ini adalah=
 $1 + 2 + \dots + n-1 = n*(n-1)/2$



Selection Sort → Analysis

- Dalam metode ini, jumlah perbandingan untuk best case & worst case-nya sama
- Memindahkan dari kanan ke kiri, meletakkan elemen ke posisi akhirnya tanpa merevisi lagi posisi tsb
- Menghabiskan sebagian besar waktu untuk mencari elemen terkecil pada sisi array yang belum terurut