

# Sorting Algorithms

# SORTING

- Pengurutan data dalam struktur data sangat penting untuk data yang beripe data numerik ataupun karakter.
- Pengurutan dapat dilakukan secara ascending (urut naik) dan descending (urut turun)
- Pengurutan (Sorting) adalah proses menyusun kembali data yang sebelumnya telah disusun dengan suatu pola tertentu, sehingga tersusun secara teratur menurut aturan tertentu.
- **Contoh:**
  - Data Acak : 5 6 8 1 3 25 10
  - Ascending : 1 3 5 6 8 10 25
  - Descending : 25 10 8 6 5 3 1

# Beberapa Algoritma Sorting

1. **Bubble Sort**
2. Selection Sort
3. Insertion Sort
4. Merge Sort
5. Quick Sort

# BUBBLE SORT

- Metode sorting termudah
- Diberi nama “Bubble” karena proses pengurutan secara berangsur-angsur bergerak/berpindah ke posisinya yang tepat, seperti gelembung yang keluar dari sebuah gelas bersoda.
- Bubble Sort mengurutkan data dengan cara membandingkan elemen sekarang dengan elemen berikutnya.

# BUBBLE SORT

- Pengurutan Ascending :Jika elemen sekarang **lebih besar** dari elemen berikutnya maka kedua elemen tersebut **ditukar**.
- Pengurutan Descending: Jika elemen sekarang **lebih kecil** dari elemen berikutnya, maka kedua elemen tersebut **ditukar**.
- Algoritma ini seolah-olah menggeser satu per satu elemen dari kanan ke kiri atau kiri ke kanan, tergantung jenis pengurutannya, asc atau desc.
- Ketika satu proses telah selesai, maka bubble sort akan mengulangi proses, demikian seterusnya sampai dengan iterasi sebanyak  $n-1$ .
- Kapan berhentinya? Bubble sort berhenti jika seluruh array telah diperiksa dan tidak ada pertukaran lagi yang bisa dilakukan, serta tercapai perurutan yang telah diinginkan.

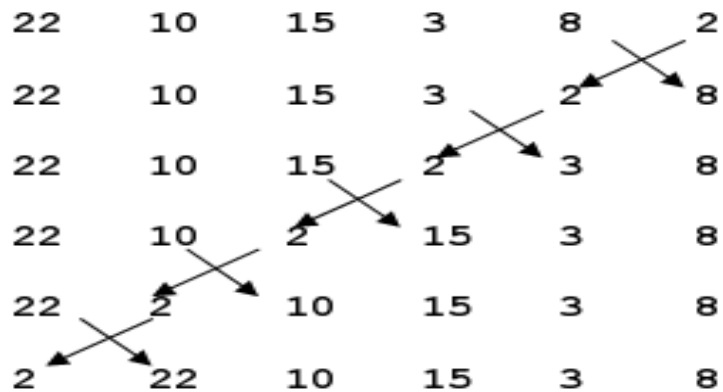
# Bubble Sort: pseudocode

BUBBLESORT(A)

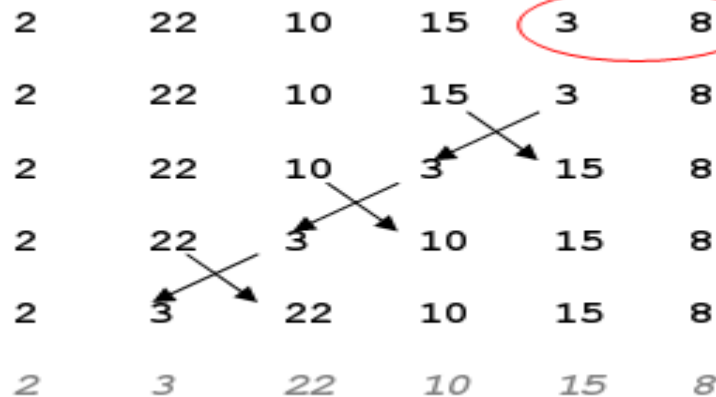
```
1  for  $i \leftarrow 1$  to  $length[A]$ 
2      do for  $j \leftarrow length[A]$  downto  $i+1$ 
3          do if  $A[j] < A[j-1]$ 
4              then exchange  $A[j] \leftrightarrow A[j-1]$ 
```

# Bubble Sort

## Proses 1



## Proses 2



Tidak ada penukaran,  
karena  $3 < 8$

Pegurutan berhenti di sini!

# Bubble Sort

## Proses 3

2	3	22	10	15	8
2	3	22	10	8	15
2	3	22	8	10	15
2	3	8	22	10	15
2	3	8	22	10	15
2	3	8	22	10	15

→ Pegurutan berhenti di sini!

## Proses 4

2	3	8	22	10	15
2	3	8	22	10	15
2	3	8	10	22	15
2	3	8	10	22	15
2	3	8	10	22	15
2	3	8	10	22	15

→ Tidak ada penukaran, karena  $10 < 15$

→ Pegurutan berhenti di sini!



# Bubble Sort

## Proses 5

2      3      8      10      22      15

2      3      8      10      15      22

*2      3      8      10      15      22*

*2      3      8      10      15      22*

*2      3      8      10      15      22*

*2      3      8      10      15      22*

→ Pegurutan berhenti di sini!

# Bubble Sort

- **Versi 1**

```
void bubble_sort(int data[]){  
    for(int i=1;i<n;i++){  
        for(int j=n-1;j>=i;j--){  
            if(data[j]<data[j-1]) tukar(&data[j],&data[j-1]); //ascending  
        }  
    }  
}
```

- **Versi 2**

```
void bubblesort2(int data[]){  
    for(i=1;i<6;i++){  
        for(int j=0;j<6-i;j++){  
            if(data[j]>data[j+1])  
                tukar(&data[j],&data[j+1]); //descending  
        }  
    }  
}
```

# Bubble Sort

- Dengan prosedur diatas, data terurut naik (ascending), untuk urut turun (descending) silahkan ubah bagian:

```
if (data[j]<data[j-1]) tukar(&data[j],&data[j-1]);
```

Menjadi:

```
if (data[j]>data[j-1]) tukar(&data[j],&data[j-1]);
```

- “The bubble sort is an easy algorithm to program, but it is slower than many other sorts”

## Bubble Sort-Desc: tahap demi tahap

Awal                      7            4            5            8            10

## Bubble Sort: tahap demi tahap

Awal	7	4	5	8	10
------	---	---	---	---	----

Step-1	7	4	5	8	10
--------	---	---	---	---	----

# Bubble Sort: tahap demi tahap

Awal                7        4        5        8        10

Step-1            7        4        5        10        8

## Bubble Sort: tahap demi tahap

Awal                7        4        5        8        10

Step-1            7        4        10        5        8

## Bubble Sort: tahap demi tahap

Awal	7	4	5	8	10
------	---	---	---	---	----

Step-1	7	10	4	5	8
--------	---	----	---	---	---



## Bubble Sort: tahap demi tahap

Awal                7        4        5        8        10

Step-1            10        7        4        5        8

## Bubble Sort: tahap demi tahap

Awal	7	4	5	8	10
Step-1	10	7	4	5	8
Step-2	10	7	4	5	8

# Bubble Sort: tahap demi tahap

Awal	7	4	5	8	10
Step-1	10	7	4	5	8
Step-2	10	7	4	8	5

# Bubble Sort: tahap demi tahap

Awal	7	4	5	8	10
Step-1	10	7	4	5	8
Step-2	10	7	8	4	5

## Bubble Sort: tahap demi tahap

Awal	7	4	5	8	10
Step-1	10	7	4	5	8
Step-2	10	8	7	4	5

## Bubble Sort: tahap demi tahap

Awal	7	4	5	8	10
Step-1	10	7	4	5	8
Step-2	10	8	7	4	5
Step-3	10	8	7	4	5

# Bubble Sort: tahap demi tahap

Awal	7	4	5	8	10
Step-1	10	7	4	5	8
Step-2	10	8	7	4	5
Step-3	10	8	7	5	4

## Bubble Sort: tahap demi tahap

Awal	7	4	5	8	10
Step-1	10	7	4	5	8
Step-2	10	8	7	4	5
Step-3	10	8	7	5	4



## Bubble Sort: tahap demi tahap

Awal	7	4	5	8	10
Step-1	10	7	4	5	8
Step-2	10	8	7	4	5
Step-3	10	8	7	5	4
Step-4	10	8	7	5	4

# Beberapa Algoritma Sorting

1. Bubble Sort
- 2. Selection Sort**
3. Insertion Sort
4. Merge Sort
5. Quick Sort

# Selection sort

- Algoritma sorting dengan memilih elemen terbesar ataupun terkecil kemudian menempatkan pada posisinya, pemilihan elemen terbesar dilakukan pada proses pengurutan data secara ascending dari posisi data paling akhir, sedangkan pemilihan elemen terkecil dilakukan pada proses pengurutan data secara ascending dari posisi data paling awal.
- Pengurutan dilakukan dengan memilih element terbesar dan menempatkan pada posisinya
- Perulangan dilakukan terus sampai tinggal 1 data terakhir, dan dipastikan itu merupakan data terkecil

# SELECTION SORT

- Untuk setiap proses, akan dicari elemen-elemen yang belum diurutkan yang memiliki nilai terkecil atau terbesar 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]`), pada putaran kedua akan dicari data kedua terkecil, dan akan ditempatkan di indeks kedua (`data[1]`).
- Selama proses, perbandingan dan pengubahan **hanya dilakukan** pada **indeks** perbandingan saja, pertukaran data secara fisik terjadi pada **akhir** proses.

# SELECTION SORT

## Proses 1

0	1	2	3	4	5
32	75	69	58	21	40

Pembanding	Posisi
32 < 75	0
32 < 69	0
32 < 58	0
32 > 21 (tukar idx)	4
21 < 40	4

Tukar data ke-0 (32) dengan data ke-4 (21)

0	1	2	3	4	5
21	75	69	58	32	40

## Proses 2

0	1	2	3	4	5
21	75	69	58	32	40

Pembanding	Posisi
75 > 69 (tukar idx)	2
69 > 58 (tukar idx)	3
58 > 32 (tukar idx)	4
32 < 40	4

Tukar data ke-1 (75) dengan data ke-4 (32)

0	1	2	3	4	5
21	32	69	58	75	40

## Proses 3

0	1	2	3	4	5
21	32	69	58	75	40

Pembanding	Posisi
69 > 58 (tukar idx)	3
58 < 75	3
58 > 40	5

Tukar data ke-2 (69) dengan data ke-5 (40)

0	1	2	3	4	5
21	32	40	58	75	69

## Proses 4

0	1	2	3	4	5
21	32	40	58	75	69

Pembanding	Posisi
58 < 75	3
58 < 69	3

Tukar data ke-3 (58) dengan data ke-3 (58)

0	1	2	3	4	5
21	32	40	58	75	69

## Proses 5

0	1	2	3	4	5
21	32	40	58	75	69

Pembanding	Posisi
75 > 69	5

Tukar data ke-4 (75) dengan data ke-5 (69)

0	1	2	3	4	5
21	32	40	58	69	75

# SELECTION SORT

- **Prosedur Selection Sort**

```
void selection_sort(int data[]){  
    for(int i=0;i<n-1;i++){  
        pos = i;  
        for(int j=i+1;j<n;j++){  
            if(data[j] < data[pos]) pos = j; //ascending  
        }  
        if(pos != i) tukar(&data[pos],&data[i]);  
    }  
}
```

# Selection Sort: contoh

	1	2	3	4	5	6
1	5	2	4	6	1	3
	1	2	3	4	5	6
2	1	2	4	6	5	3
	1	2	3	4	5	6
3	1	2	4	6	5	3
	1	2	3	4	5	6
4	1	2	3	6	5	4
	1	2	3	4	5	6
5	1	2	3	4	5	6
	1	2	3	4	5	6
6	1	2	3	4	5	6

Carilah elemen terkecil & tukar dengan “5”


1 fixed. Carilah elemen terkecil & tukar dengan “2”

1,2 fixed. Carilah elemen terkecil & tukar dengan “4”

1,2,3 fixed. Carilah elemen terkecil & tukar dengan “6”

1,2,3,4 fixed. Carilah elemen terkecil & tukar dengan “5”

1,2,3,4,5 fixed, otomatis elemen terakhir sudah pada posisi yang benar



Urutkan dengan algoritma bubble sort dan selection sort dari data berikut

5 10 4 2 8 19 7