



ALGORITMA PENGURUTAN

Oleh : S. Thya Safitri, MT



Definisi

- Sorting merupakan suatu proses untuk menyusun kembali himpunan obyek menggunakan aturan tertentu.
- Sorting disebut juga sebagai suatu algoritma untuk meletakkan kumpulan elemen data kedalam urutan tertentu berdasarkan satu atau beberapa kunci dalam tiap-tiap elemen.
- Pengurutan data dalam struktur data sangat penting untuk data yang bertipe data numerik ataupun karakter.



Alasan ???

- Mengapa harus melakukan sorting data?
Ada banyak alasan dan keuntungan dengan mengurutkan data.
1. Data yang terurut mudah untuk dicari, mudah untuk diperiksa, dan mudah untuk dibetulkan jika terdapat kesalahan.
 2. Data yang terurut dengan baik juga mudah untuk dihapus jika sewaktu-waktu data tersebut tidak diperlukan lagi.
 3. Selain itu, dengan mengurutkan data maka kita semakin mudah untuk menyisipkan data ataupun melakukan penggabungan data.



Macam Sorting

- Pada dasarnya ada dua macam urutan yang biasa digunakan dalam suatu proses sorting:
 - Urut Naik (ascending)
Mengurutkan dari data yang mempunyai nilai paling kecil sampai paling besar
 - Urut Turun (descending)
Mengurutkan dari data yang mempunyai nilai paling besar sampai paling kecil.
- **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



Metode Pengurutan Data

- Pengurutan berdasarkan perbandingan (*comparison-based sorting*)
 - Bubble sort, exchange sort
- Pengurutan berdasarkan prioritas (*priority queue sorting method*)
 - Selection sort, heap sort (menggunakan tree)
- Pengurutan berdasarkan penyisipan dan penjagaan terurut (*insert and keep sorted method*)
 - Insertion sort, tree sort
- Pengurutan berdasarkan pembagian dan penguasaan (*divide and conquer method*)
 - Quick sort, merge sort
- Pengurutan berkurang menurun (*diminishing increment sort method*)
 - Shell sort (pengembangan insertion)



- Algoritma sorting dasar:
 - Bubble Sort
 - Insertion Sort
 - Selection Sort
- Algoritma sorting lanjutan:
 - Merge Sort
 - Quick Sort
 - Bucket Sort
 - Shell Sort
 - Radix Sort
 - External Sort



Macam Algoritma Pengurutan yang akan dibahas

- Insertion Sort
- Selection Sort
- Bubble Sort





Insertion Sort



Insertion Sort

- Mirip dengan cara orang **mengurutkan** kartu, selembat demi selembat kartu diambil dan **disisipkan** (insert) ke tempat yang seharusnya.
- Pengurutan dimulai dari data ke-2 sampai dengan data terakhir, jika ditemukan data yang **lebih kecil**, maka akan ditempatkan (**diinsert**) diposisi yang seharusnya.
- Pada penyisipan elemen, maka elemen-elemen lain akan bergeser ke belakang





Proses Insertion Sort

- Proses pengurutan dengan metode penyisipan langsung dapat dijelaskan sebagai berikut :

Data dicek satu per satu mulai dari yang kedua sampai dengan yang terakhir. Apabila ditemukan data yang lebih kecil daripada data sebelumnya, maka data tersebut disisipkan pada posisi yang sesuai. Akan lebih mudah apabila membayangkan pengurutan kartu. Pertama-tama anda meletakkan kartu-kartu tersebut di atas meja, kemudian melihatnya dari kiri ke kanan. Apabila kartu di sebelah kanan lebih kecil daripada kartu di sebelah kiri, maka ambil kartu tersebut dan sisipkan di tempat yang sesuai.



Insertion Sort Example

Proses 1

0	1	2	3	4	5
22	10	15	3	8	2

Temp	Cek	Geser
10	Temp<22?	Data ke-0 ke posisi 1

Temp menempati posisi ke -0

0	1	2	3	4	5
10	22	15	3	8	2

Proses 2

0	1	2	3	4	5
10	22	15	3	8	2

Temp	Cek	Geser
15	Temp<22	Data ke-1 ke posisi 2
15	Temp>10	-

Temp menempati posisi ke-1

0	1	2	3	4	5
10	15	22	3	8	2

Proses 3

0	1	2	3	4	5
10	15	22	3	8	2

Temp	Cek	Geser
3	Temp<22	Data ke-2 ke posisi 3
3	Temp<15	Data ke-1 ke posisi 2
3	Temp<10	Data ke-0 ke posisi 1

Temp menempati posisi ke-0

0	1	2	3	4	5
3	10	15	22	8	2

Proses 4

0	1	2	3	4	5
3	10	15	22	8	2

Temp	Cek	Geser
8	Temp<22	Data ke-3 ke posisi 4
8	Temp<15	Data ke-2 ke posisi 3
8	Temp<10	Data ke-1 ke posisi 2
8	Temp>3	-

Temp menempati posisi ke-1

0	1	2	3	4	5
3	8	10	15	22	2

Insertion Sort Example

Proses 5

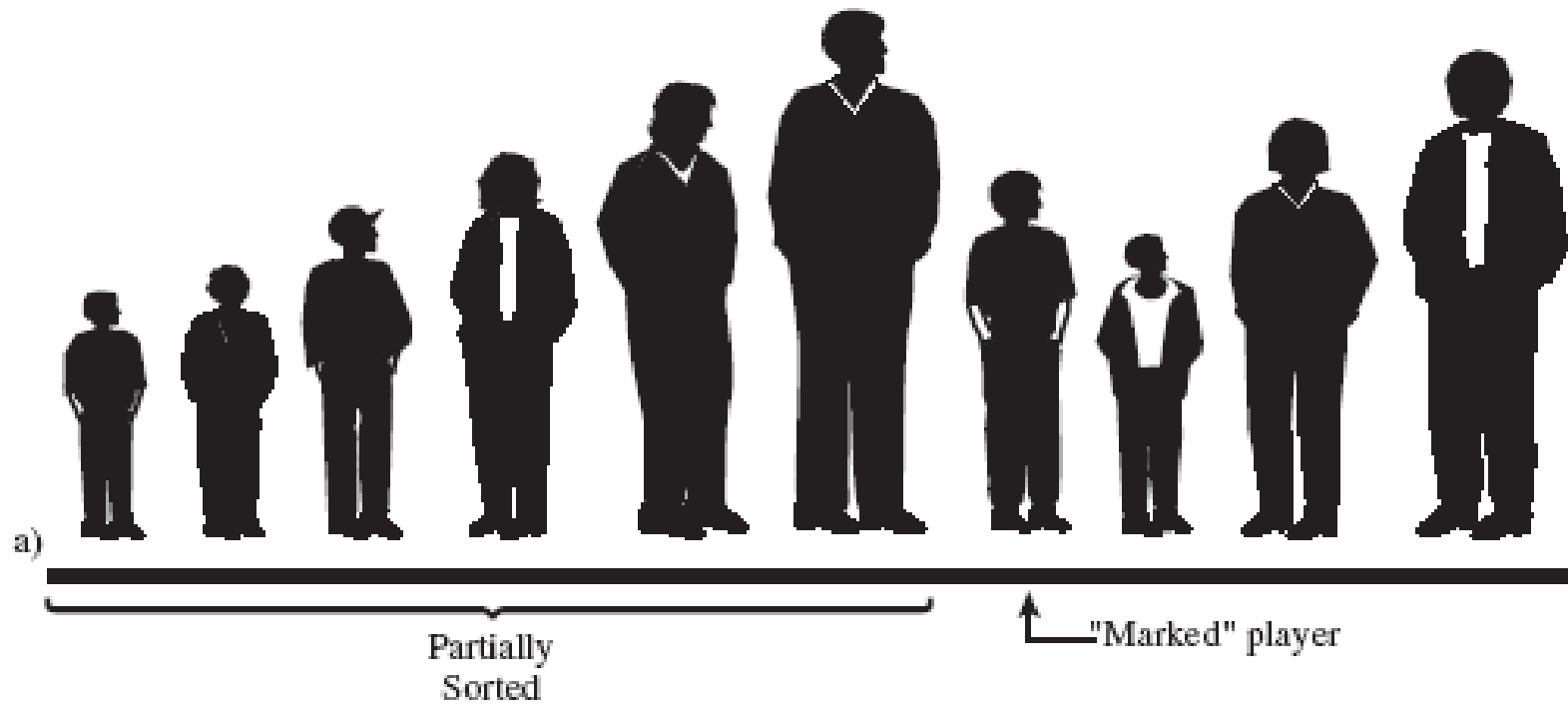
0	1	2	3	4	5
3	8	10	15	22	2

Temp	Cek	Geser
2	Temp<22	Data ke-4 ke posisi 5
2	Temp<15	Data ke-3 ke posisi 4
2	Temp<10	Data ke-2 ke posisi 3
2	Temp<8	Data ke-1 ke posisi 2
2	Temp<3	Data ke-0 ke posisi 1

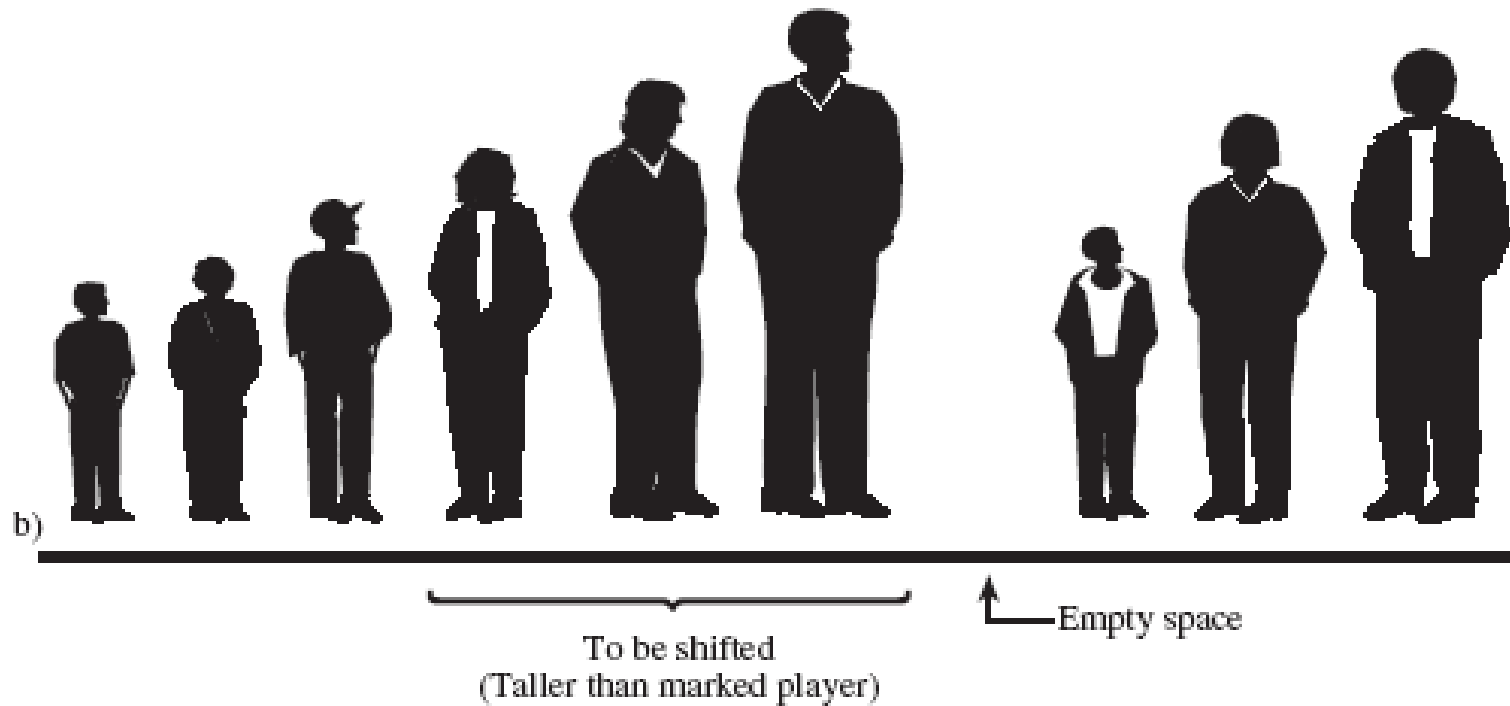
Temp menempati posisi ke-0

0	1	2	3	4	5
2	3	8	10	15	22

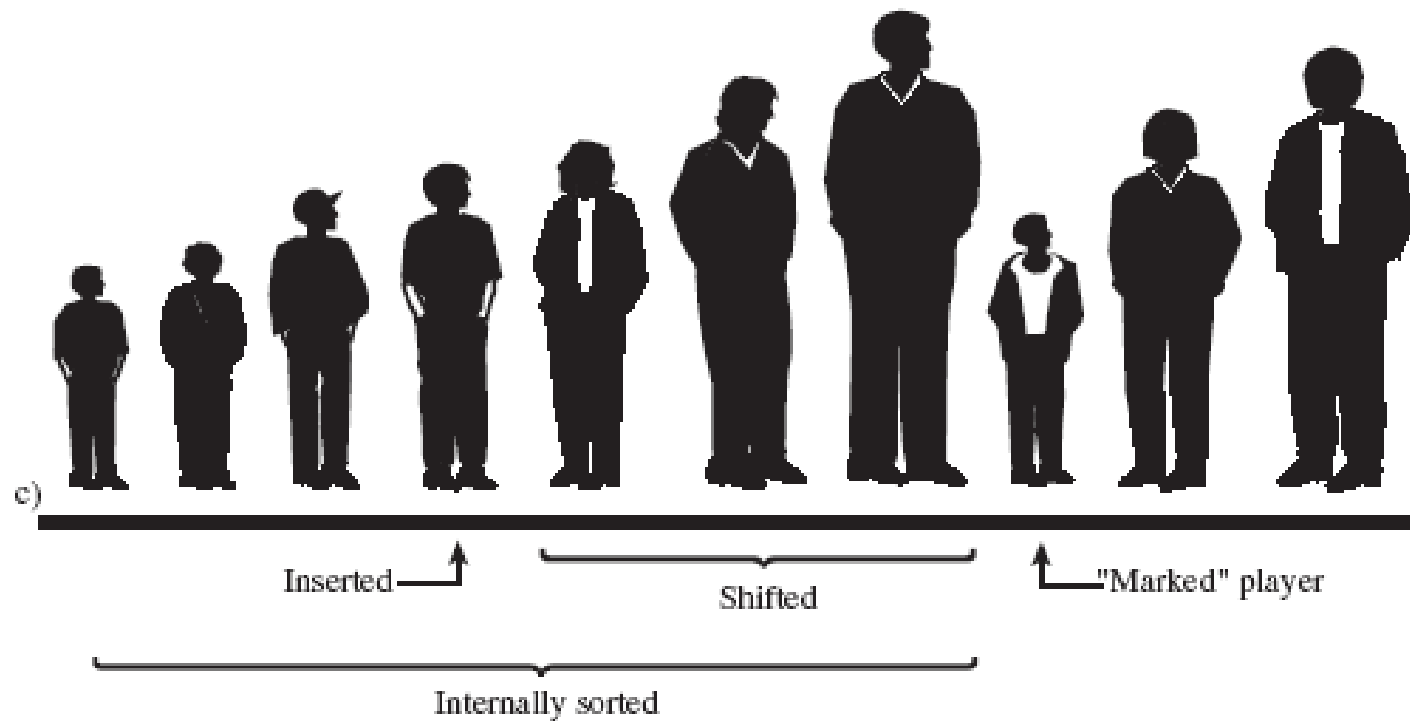
Ilustrasi (2)



Ilustrasi (2)



Ilustrasi (2)



Ilustrasi (3)

Proses	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]
K = 1:	77	33	44	11	88	22	66	55
K = 2:	77	33	44	11	88	22	66	55
K = 3:	33	77	44	11	88	22	66	55
K = 4:	33	44	77	11	88	22	66	55
K = 5:	11	33	44	77	88	22	66	55
K = 6:	11	33	44	77	88	22	66	55
K = 7:	11	22	33	44	77	88	66	55
K = 8:	11	22	33	44	66	77	88	55
Urutan :	11	22	33	44	55	66	77	88



Algoritma dan Procedure Insertion

- Algoritma penyisipan langsung dapat dituliskan sebagai berikut :

1. $i = 1$
2. selama ($i < N$) kerjakan baris 3 sampai dengan 9
3. $x = \text{Data}[i]$
4. $j = i - 1$
5. selama ($x < \text{Data}[j]$) kerjakan baris 6 dan 7
6. $\text{Data}[j + 1] = \text{Data}[j]$
7. $j = j - 1$
8. $\text{Data}[j+1] = x$
9. $i = i + 1$



Algoritma dan Procedure Insertion

- Prosedur yang menggunakan metode penyisipan langsung:

```
void StraighInsertSort()  
{  
    int i, j, x;  
    for(i=1; i<Max; i++)  
    {  
        x = Data[i];  
        j = i - 1;  
        while (x < Data[j])  
        {  
            Data[j+1] = Data[j];  
            j--;  
        } Data[j+1] = x;  
    }  
}
```



Latihan (1)

- Data :

7, 19, 4, 8, 20, 1

Urutkan secara ascending dengan menggunakan algoritma insertion sort!



Selection Sort



Selection Sort

- Metode seleksi melakukan pengurutan dengan cara mencari data yang terkecil kemudian menukarkannya dengan data yang digunakan sebagai acuan atau sering dinamakan pivot.
- Selama proses, perbandingan dan pengubahan **hanya dilakukan** pada **indeks** perbandingan saja, pertukaran data secara fisik terjadi pada **akhir** proses.
- Merupakan kombinasi antara sorting dan searching.
- Untuk setiap proses, akan dicari elemen-elemen yang belum diurutkan yang memiliki nilai terkecil (ascending) atau terbesar (descending) kemudian akan dipertukarkan ke posisi yang tepat di dalam array.



Proses Selection Sort

- Proses pengurutan dengan metode seleksi dapat dijelaskan sebagai berikut :

Langkah pertama dicari data terkecil dari data pertama sampai data terakhir. Kemudian data terkecil ditukar dengan data pertama. Dengan demikian, data pertama sekarang mempunyai nilai paling kecil dibanding data yang lain. Langkah kedua, data terkecil kita cari mulai dari data kedua sampai terakhir. Data terkecil yang kita peroleh ditukar dengan data kedua dan demikian seterusnya sampai semua elemen dalam keadaan terurutkan.



Selection Sort Example

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



Algoritma dan Procedure Selection

- Algoritma seleksi dapat dituliskan sebagai berikut :

1. $i = 0$
2. selama ($i < N-1$) kerjakan baris 3 sampai dengan 9
3. $k = i$
4. $j = i + 1$
5. Selama ($j < N$) kerjakan baris 6 dan 7
6. Jika ($Data[k] > Data[j]$) maka $k = j$
7. $j = j + 1$
8. Tukar $Data[i]$ dengan $Data[k]$
9. $i = i + 1$



Algoritma dan Procedure Selection

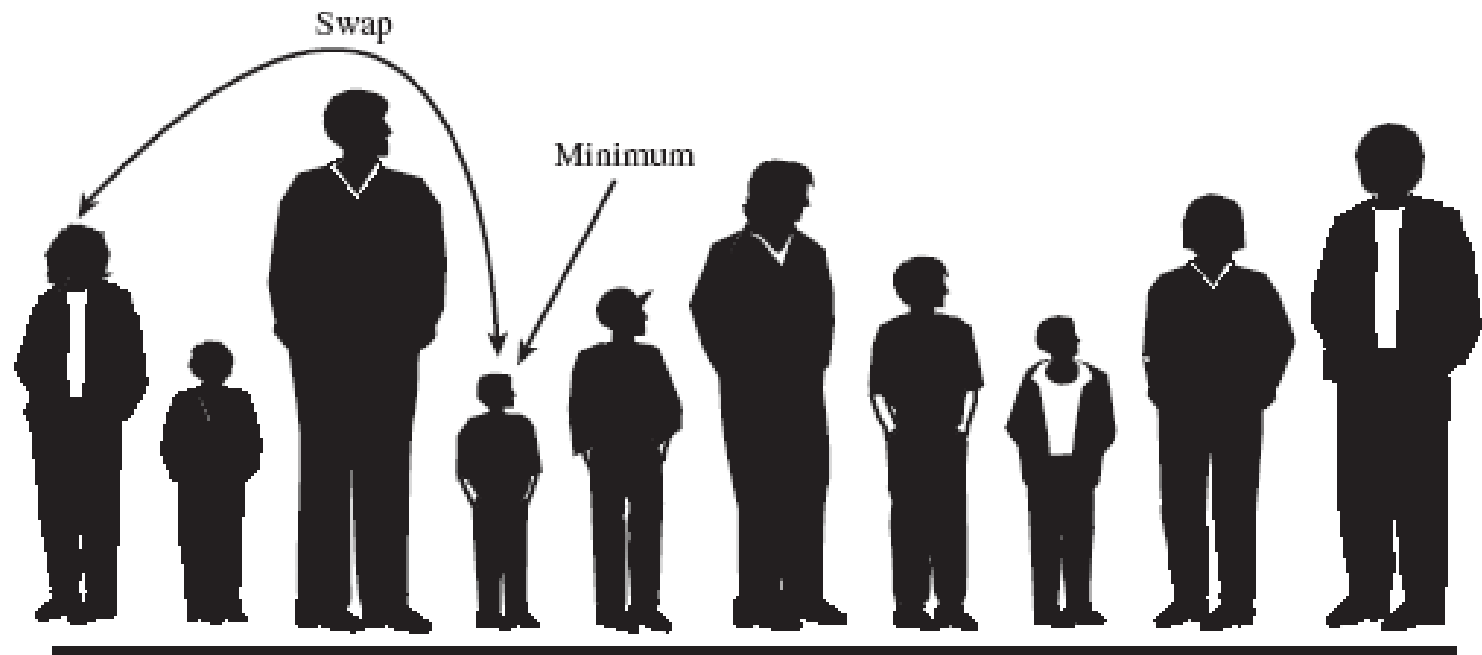
- Di bawah ini merupakan prosedur yang menggunakan metode seleksi:

```
void SelectionSort()  
{  
    int i, j, k;  
    for(i=0; i<Max-1;i++)  
    {  
        k = i;  
        for (j=i+1; j<Max; j++)  
            if(Data[k] > Data[j])  
                k = j;  
        Tukar(&Data[i], &Data[k]);  
    }  
}
```

Ilustrasi (1)

Proses	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]
K = 1 ; LOC = 4	77	33	44	11	88	22	66	55
K = 2 ; LOC = 6	11	33	44	77	88	22	66	55
K = 3 ; LOC = 6	11	22	44	77	88	33	66	55
K = 4 ; LOC = 6	11	22	33	77	88	44	66	55
K = 5 ; LOC = 8	11	22	33	44	88	77	66	55
K = 6 ; LOC = 7	11	22	33	44	55	77	66	88
K = 7 ; LOC = 7	11	22	33	44	55	66	77	88
Urutan :	11	22	33	44	55	66	77	88

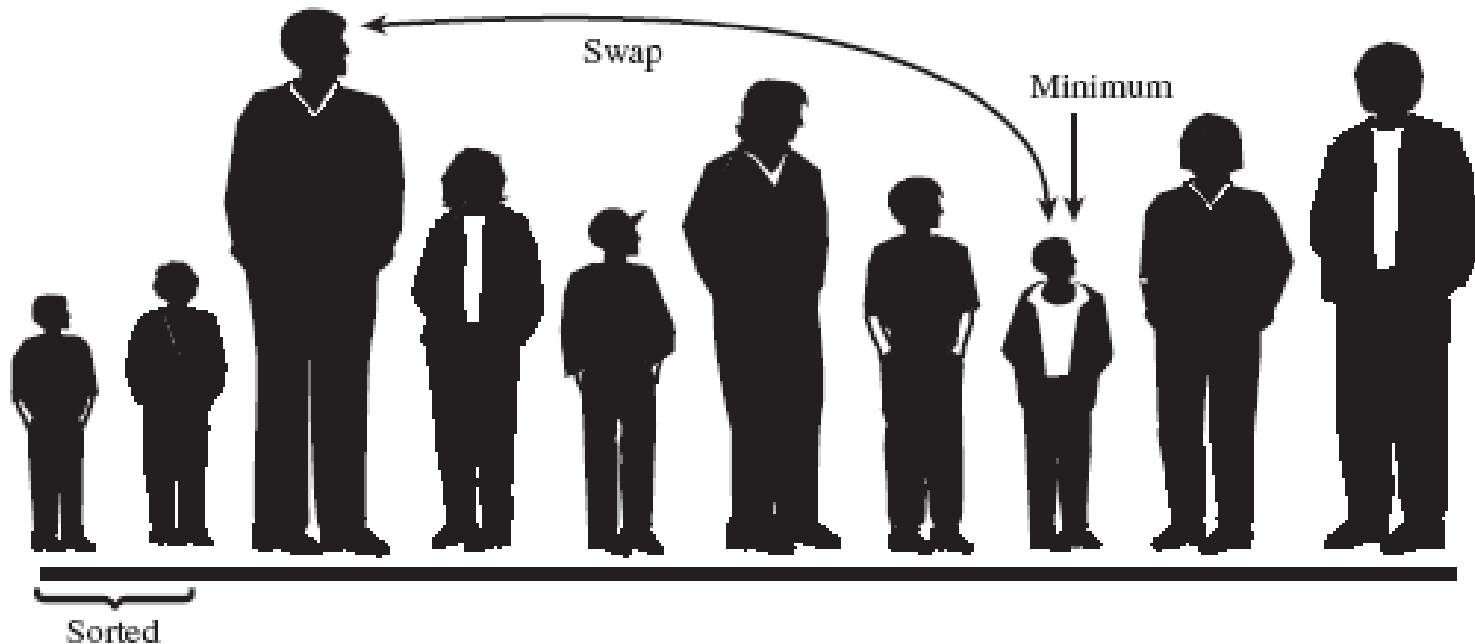
Ilustrasi (2)



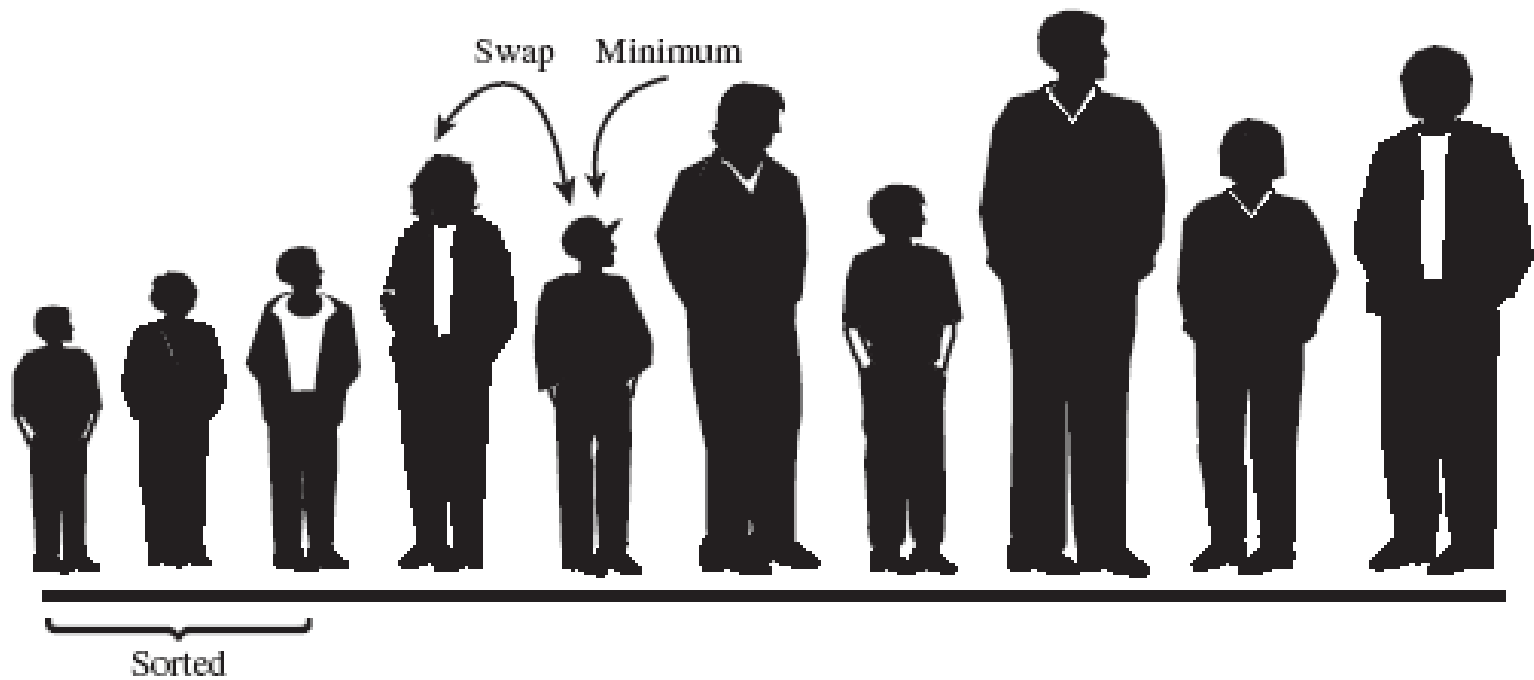
Ilustrasi (2)



Ilustrasi (2)



Ilustrasi (2)





Latihan (2)

- Data :

7, 19, 4, 8, 20, 1

Urutkan secara ascending dengan menggunakan algoritma selection sort!



Bubble Sort



Bubble Sort

- 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**.
- 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 Example

4 2 5 3 9

(4 2 5 3 9) menjadi (2 4 5 3 9)

(2 4 5 3 9) menjadi (2 4 5 3 9)

(2 4 5 3 9) menjadi (2 4 3 5 9)

(2 4 3 5 9) menjadi (2 4 3 5 9)

Pass kedua

(2 4 3 5 9) menjadi (2 4 3 5 9)

(2 4 3 5 9) menjadi (2 3 4 5 9)

(2 3 4 5 9) menjadi (2 3 4 5 9)

(2 3 4 5 9) menjadi (2 3 4 5 9)

Pass ketiga

(2 3 4 5 9) menjadi (2 3 4 5 9)

(2 3 4 5 9) menjadi (2 3 4 5 9)

(2 3 4 5 9) menjadi (2 3 4 5 9)

(2 3 4 5 9) menjadi (2 3 4 5 9)



Algoritma dan Procedure Bubble

Algoritma bubble dapat dituliskan sebagai berikut (ascending):

1. $i = 0$
2. selama ($i < N-1$) kerjakan baris 3 sampai 7
3. $j = N - 1$
4. Selama ($j \geq i$) kerjakan baris 5 sampai 7
5. Jika ($Data[j-1] > Data[j]$) maka tukar $Data[j-1]$ dengan $Data[j]$
6. $j = j - 1$
7. $i = i + 1$



Algoritma dan Procedure Bubble

- Prosedur yang menggunakan metode gelembung:

```
void BubbleSort()  
{  
    int i, j;  
    for(i=1; i<Max-1; i++)  
        for(j=Max-1; j>=i; j--)  
            if(Data[j-1] > Data[j])  
                Tukar(&Data[j-1], &Data[j]);  
}
```



Bubble Sort

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

```
if(Data[j-1] > Data[j])  
    Tukar(&Data[j-1], &Data[j]);
```

Menjadi:

```
if(Data[j-1] < Data[j])  
    Tukar(&Data[j-1], &Data[j]);
```

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



Perbandingan 3 Metode

- Tabel Perbandingan Kecepatan Metode Pengurutan Data

Untuk data sejumlah 10.000 data pada komputer Pentium II 450 MHz

Metode	Waktu (detik)		
	Data Acak	Data Ascending	Data Descending
Bubble Sort	11,2	1,32	19,77
Insertion Sort	1,09	0,00	2,25
Selection Sort	1,32	1,32	19,77



Bubble Sort

- Merupakan algoritma sorting yang paling mudah.
- Proses pengurutan secara berangsur-angsur bergerak/berpindah ke posisinya yang tepat, seperti gelembung.





Bubble Sort

- Bubble Sort mengurutkan data dengan cara membandingkan elemen sekarang dengan elemen berikutnya.
 - Jika elemen sekarang **lebih besar** dari elemen berikutnya maka kedua elemen tersebut **ditukar**, jika pengurutan **ascending**.
 - Jika elemen sekarang **lebih kecil** dari elemen berikutnya, maka kedua elemen tersebut **ditukar**, jika pengurutan **descending**



Bubble Sort

- Algoritma ini seolah-olah menggeser satu per satu elemen dari kanan ke kiri atau kiri ke kanan, tergantung jenis pengurutannya.
- Ketika satu proses telah selesai, maka bubble sort akan mengulangi proses, demikian seterusnya.
- Kapan berhentinya? Bubble sort berhenti jika seluruh array telah diperiksa dan tidak ada pertukaran lagi yang bisa dilakukan, serta tercapai perurutan yang telah diinginkan.

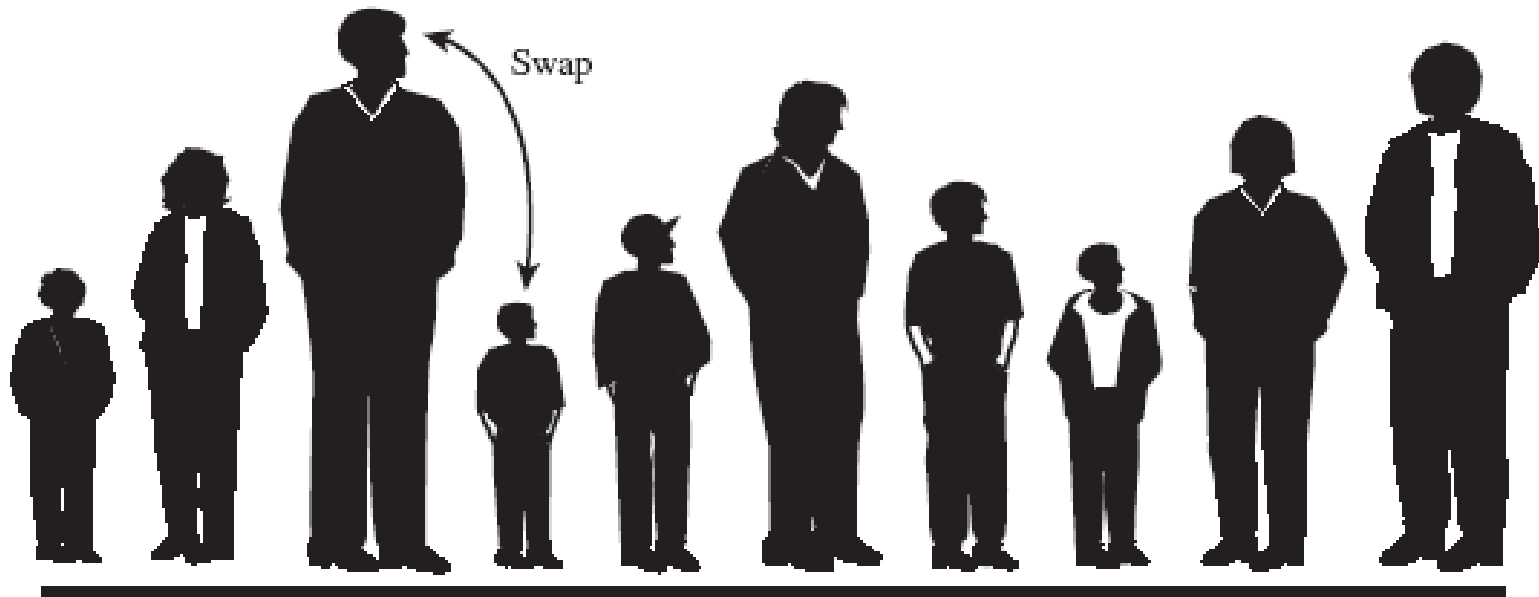
Ilustrasi Bubble Sort (1)



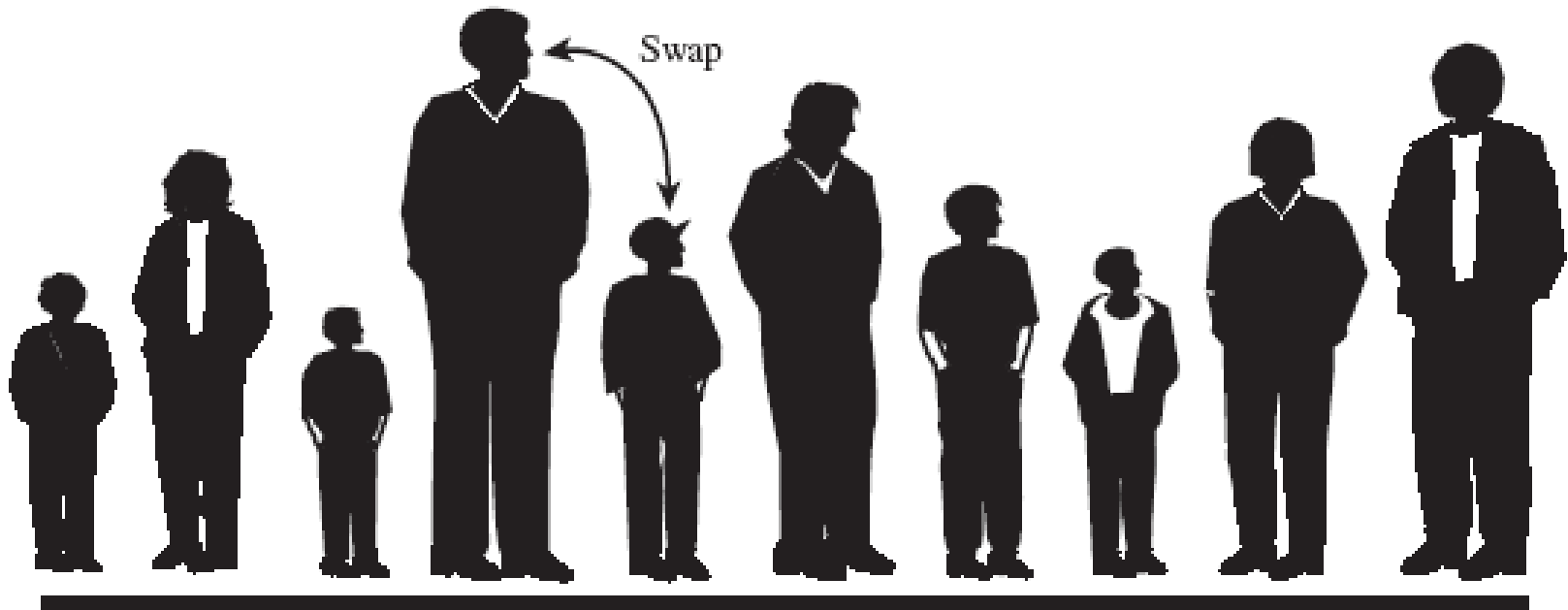
Ilustrasi Bubble Sort (1)



Ilustrasi Bubble Sort (1)



Ilustrasi Bubble Sort (1)





Algoritma Bubble Sort

```
ch = true;
for (i=0; i < n-2 && ch; i++) {
    ch = false;
    for (j=n-1; j > i; j--) {
        if (x[j] < x[j-1]) {
            tmp = x[j];
            x[j] = x[j-1];
            x[j-1] = tmp;
            ch = true;
        }
    }
}
```



Latihan (3)

- Data :

7, 19, 4, 8, 20, 1

Urutkan secara ascending dengan menggunakan algoritma insert, select dan bubble sort!





Pustaka

- Mitchell Waite, “Data Structures & Algorithms in Java”, SAMS, 2001