**BAB IV**

**BUBBLE SORT**

1. **Pengertian Bubble Sort**

Pengurutan merupakan proses dasar yang ada dalam algoritma dan stuktur data. Terdapat banyak algoritma pengurutan yang sering digunakan, namun pada tulisan kali ini akan dibahas mengenai dasar algoritma Bubble Sort. Algortima ini merupakan algortima pengurutan sederhana dan biasanya dipelajari sebagai pokok bahasan seputar pengurutan.

Algoritma Bubble Sort ini merupakan proses pengurutan yang secara berangsur-angsur berpindah ke posisi yang tepat karena itulah dinamakan Bubble yang artinya gelembung. Algoritma ini akan mengurutkan data dari yang terbesar ke yang terkecil (ascending) atau sebaliknya (descending).

Secara sederhana, bisa didefenisikan algoritma Bubble Sort adalah pengurutan dengan cara pertukaran data dengan data disebelahnya secara terus menerus sampai dalam satu iterasi tertentu tidak ada lagi perubahan.

Bubble Sort adalah salah satu algoritma untuk sorting data, atau kata lainnya mengurutkan data dari yang terbesar ke yang terkecil atau sebaliknya (Ascending atau Descending).

Bubble sort (metode gelembung) adalah metode/algoritma pengurutan dengan dengan cara melakukan penukaran data dengan tepat disebelahnya secara terus menerus sampai bisa dipastikan dalam satu iterasi tertentu tidak ada lagi perubahan. Jika tidak ada perubahan berarti data sudah terurut. Disebut pengurutan gelembung karena masing-masing kunci akan dengan lambat menggelembung ke posisinya yang tepat.

                Metode pengurutan gelembung (Bubble Sort) diinspirasikan oleh gelembung sabun yang berada dipermukaan air. Karena berat jenis gelembung sabun lebih ringan daripada berat jenis air, maka gelembung sabun selalu terapung ke atas permukaan. Prinsip di atas dipakai pada pengurutan gelembung.

Algoritma bubble sort adalah salah satu algoritma pengurutan yang paling simple, baik dalam hal pengertian maupun penerapannya. Ide dari algoritma ini adalah mengulang proses pembandingan antara tiap-tiap elemen

array dan menukarnya apabila urutannya salah. Pembandingan elemen-elemen ini akan terus diulang hingga tidak perlu dilakukan penukaran lagi. Algoritma

ini termasuk dalam golongan algoritma comparison sort, karena menggunakan perbandingan dalam operasi antar elemennya. Berikut ini adalah gambaran dari algoritma bubble sort. Misalkan kita mempunyai sebuah array dengan.  Elemen-elemen “4 2 5 3 9”. Proses yang akan terjadi apabila digunakan algoritma bubblesort adalah sebagai berikut.

Pass pertama

(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)

Dapat dilihat pada proses di atas, sebenarnya pada pass kedua, langkah kedua, array telah terurut. Namun algoritma tetap dilanjutkan hingga pass kedua berakhir. Pass ketiga dilakukan karena definisi terurut dalam algoritma bubblesort adalah tidak ada satupun penukaran pada suatu pass, sehingga pass ketiga dibutuhkan untuk memverifikasi keurutan array tersebut.

1. **Algoritma Bubble Sort**

1. Membandingkan data ke-i dengan data ke-(i+1) (tepat bersebelahan). Jika tidak sesuai maka tukar (data ke-i = data ke-(i+1) dan data ke-(i+1) = data ke-i). Apa maksudnya tidak sesuai ? Jika kita menginginkan algoritme menghasilkan data dengan urutan ascending (A-Z) kondisi tidak sesuai adalah data ke-i > data ke-i+1, dan sebaliknya untuk urutan descending (A-Z).

2. Membandingkan data ke-(i+1) dengan data ke-(i+2). Kita melakukan pembandingan ini sampai data terakhir. Contoh: 1 dgn 2; 2 dgn 3; 3 dgn 4; 4 dgn 5 … ; n-1 dgn n.

3. Selesai satu iterasi, adalah jika kita sudah selesai membandingkan antara (n-1) dgn n. Setelah selesai satu iterasi kita lanjutkan lagi iterasi berikutnya sesuai dengan aturan ke-1. mulai dari data ke-1 dgn data ke-2, dst.

4. Proses akan berhenti jika tidak ada pertukaran dalam satu iterasi.

**Contoh Kasus Bubble Sort :**

Sekarang saatnya untuk contoh kasus. Pada contoh kasus ini, saya akan menjelaskan untuk kasus Ascending dan Descending

1. **Ascending**

Berikut diberikan sebuah deretan bilangan seperti berikut:

5, 12, 3, 19, 1, 47

Langkah Bubble Sort:

Iterasi 1:

5, 12, 3, 19, 1, 47 --> Tidak ada pertukaran. (5 < 12 == true)

5, 3, 12, 19, 1, 47 --> Ada pertukaran. (12 < 3 == false)

5, 3, 12, 19, 1, 47 --> Tidak ada pertukaran. (12 < 19 == true)

5, 3, 12, 1, 19, 47 --> Ada pertukaran. (19 < 1 == false)

5, 3, 12, 1, 19, 47 --> Tidak ada pertukaran. (19 < 47 == true)

Iterasi 2:

3, 5, 12, 1, 19, 47 --> Ada petukaran. (5 < 3 == false)

3, 5, 12, 1, 19, 47 --> Tidak ada pertukaran. (5 < 12 == true)

3, 5, 1, 12, 19, 47 --> Ada pertukaran. (12 < 1 == false)

3, 5, 1, 12, 19, 47 --> Tidak ada pertukaran. (12 < 19 == true)

3, 5, 1, 12, 19, 47 --> Tidak ada pertukaran. (19 < 47 == true)

Iterasi 3:

3, 5, 1, 12, 19, 47 --> Tidak ada pertukaran. (3 < 5 == true)

3, 1, 5, 12, 19, 47 --> Ada pertukaran. (5 < 1 == false)

3, 1, 5, 12, 19, 47 --> Tidak ada pertukaran. (5 < 12 == true)

3, 1, 5, 12, 19, 47 --> Tidak ada pertukaran. (12 < 19 == true)

3, 1, 5, 12, 19, 47 --> Tidak ada pertukaran. (19 < 47 == true)

Iterasi 4:

1, 3, 5, 12, 19, 47 --> Ada pertukaran. (3 < 1 == false)

1, 3, 5, 12, 19, 47 --> Tidak ada pertukaran. (3 < 5 == true)

1, 3, 5, 12, 19, 47 --> Tidak ada pertukaran. (5 < 12 == true)

1, 3, 5, 12, 19, 47 --> Tidak ada pertukaran. (12 < 19 == true)

1, 3, 5, 12, 19, 47 --> Tidak ada pertukaran. (19 < 47 == true)

Iterasi 5:

1, 3, 5, 12, 19, 47 --> Tidak ada pertukaran. (1 < 3 == true)

1, 3, 5, 12, 19, 47 --> Tidak ada pertukaran. (3 < 5 == true)

1, 3, 5, 12, 19, 47 --> Tidak ada pertukaran. (5 < 12 == true)

1, 3, 5, 12, 19, 47 --> Tidak ada pertukaran. (12 < 19 == true)

1, 3, 5, 12, 19, 47 --> Tidak ada pertukaran. (19 < 47 == true)

Iterasi 6:

1, 3, 5, 12, 19, 47 --> Tidak ada pertukaran. (1 < 3 == true)

1, 3, 5, 12, 19, 47 --> Tidak ada pertukaran. (3 < 5 == true)

1, 3, 5, 12, 19, 47 --> Tidak ada pertukaran. (5 < 12 == true)

1, 3, 5, 12, 19, 47 --> Tidak ada pertukaran. (12 < 19 == true)

1, 3, 5, 12, 19, 47 --> Tidak ada pertukaran. (19 < 47 == true)

Jadi, hasil akhir deretan bilangan diatas setelah di Bubble Sort secara Ascending ialah

1, 3, 5, 12, 19, 47

1. **Descending**

Masih menggunakan deretan bilangan yang seperti contoh diatas.

5, 12, 3, 19, 1, 47

Langkah Bubble Sort:

Iterasi 1:

12, 5, 3, 19, 1, 47 --> Ada pertukaran. (5 > 12 == false)

12, 5, 3, 19, 1, 47 --> Tidak ada pertukaran. (5 > 3 == true)

12, 5, 19, 3, 1, 47 --> Ada pertukaran. (3 > 19 == false)

12, 5, 19, 3, 1, 47 --> Tidak ada pertukaran. (3 > 1 == true)

12, 5, 19, 3, 47, 1 --> Ada pertukaran. (1 > 47 == false)

Iterasi 2:

12, 5, 19, 3, 47, 1 --> Tidak ada pertukaran. (12 > 5 == true)

12, 19, 5, 3, 47, 1 --> Ada pertukaran. (5 > 19 == false)

12, 19, 5, 3, 47, 1 --> Tidak ada pertukaran. (5 > 3 == true)

12, 19, 5, 47, 3, 1 --> Ada pertukaran. (3 > 47 == false)

12, 19, 5, 47, 3, 1 --> Tidak ada pertukaran. (3 > 1 == true)

Iterasi 3:

19, 12, 5, 47, 3, 1 --> Ada pertukaran. (12 > 19 == false)

19, 12, 5, 47, 3, 1 --> Tidak ada pertukaran. (12 > 5 == true)

19, 12, 47, 5, 3, 1 --> Ada pertukaran. (5 > 47 == false)

19, 12, 47, 5, 3, 1 --> Tidak ada pertukaran. (5 > 3 == true)

19, 12, 47, 5, 3, 1 --> Tidak ada pertukaran. (3 > 1 == true)

Iterasi 4:

19, 12, 47, 5, 3, 1 --> Tidak ada pertukaran. (19 > 12 == true)

19, 47, 12, 5, 3, 1 --> Ada pertukaran. (12 > 47 == false)

19, 47, 12, 5, 3, 1 --> Tidak ada pertukaran. (12 > 5 == true)

19, 47, 12, 5, 3, 1 --> Tidak ada pertukaran. (5 > 3 == true)

19, 47, 12, 5, 3, 1 --> Tidak ada pertukaran. (3 > 1 == true)

Iterasi 5:

47, 19, 12, 5, 3, 1 --> Ada pertukaran. (19 > 47 == false)

47, 19, 12, 5, 3, 1 --> Tidak ada pertukaran. (19 > 12 == true)

47, 19, 12, 5, 3, 1 --> Tidak ada pertukaran. (12 > 5 ==true)

47, 19, 12, 5, 3, 1 --> Tidak ada pertukaran. (5 > 3 == true)

47, 19, 12, 5, 3, 1 --> Tidak ada pertukaran. (3 > 1 == true)

Iterasi 6:

47, 19, 12, 5, 3, 1 --> Tidak ada pertukaran. (47 > 19 == true)

47, 19, 12, 5, 3, 1 --> Tidak ada pertukaran. (19 > 12 == true)

47, 19, 12, 5, 3, 1 --> Tidak ada pertukaran. (12 > 5 == true)

47, 19, 12, 5, 3, 1 --> Tidak ada pertukaran. (5 > 3 == true)

47, 19, 12, 5, 3, 1 --> Tidak ada pertukaran. (3 > 1 == true)

Jadi, hasil akhir deretan bilangan diatas setelah di Bubble Sort secara Descending ialah

47, 19, 12, 5, 3, 1

1. **Pendeklarasian Bubble Sort**

Berikut adalah Contoh Pendeklarasian Bubble Sort pada Java :

public class BubbleSort {

public static void main(String args[]){

int[] data={21,13,36,12,18,9,59,24};

for (int outer = data.length - 1; outer > 0; outer--) {

for (int inter = 0; inter < outer; inter++) {

if (data[inter] > data[inter + 1]) {

int temp = data[inter];

data[inter] = data[inter + 1];

data[inter + 1] = temp;

}

}

}

for (int i=0;i<data.length;i++)

System.out.print(data[i]+" ");

}

}

1. **Kelebihan dan Kelemahan Bubble Sort**

Kelebihan :

·      Metode Bubble Sort merupakan metode yang paling simpel

·      Metode Bubble Sort mudah dipahami algoritmanya

Kelemahan:

Meskipun simpel metode Bubble sort  merupakan metode pengurutan yang paling tidak efisien.  *Kelemahan buble sort* adalah pada saat mengurutkan data yang sangat besar akan mengalami kelambatan luar biasa, atau dengan kata lain kinerja memburuk cukup signifikan ketika data yang diolah jika  data cukup banyak. Kelemahan lain adalah jumlah pengulangan akan tetap sama jumlahnya walaupun data sesungguhnya sudah cukup terurut. Hal ini disebabkan setiap data dibandingkan dengan setiap data yang lain untuk menentukan posisinya.

**DAFTAR PUSTAKA**

<https://www.codepolitan.com/tutorial/algoritma-bubble-sort-dengan-java>

<https://teknojurnal.com/pengertian-algoritma-bubble-sort/>

<http://buublesort.blogspot.co.id/>