PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika
Vol. 13, No. 2, September 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i2.1008

Implementasi Algoritma Bubble Sort Terhadap 2 Buah Model Varian Pengurutan Data Menggunakan Bahasa Program Java

Endang Sunandar[1]; Indrianto[2]

[1]Jurusan Sistem Komputer Universitas Raharja
Jalan Jendral Sudirman No 40 Modernland Cikokol Tangerang
[2]Jurusan Teknik Informatika Institut Teknologi PLN
Jalan Lingkar Luar Barat Duri Kosambi, Cengkareng Jakarta Barat
[1]endang.sunandar@raharja.info
[2]indrianto@itpln.ac.id

ABSTRACT

In the field of computer science the efforts made in solving a problem are to use systematic, logical, and strategic flow and steps. These steps are known as algorithms. So many algorithms exist for solving a particular problem. In connection with this paper the algorithm discussed is about the data sorting algorithm, among the few known are: Merge, Selection, Insertion, Bubble, Quick, Heap, Shell. All of these algorithms have advantages and disadvantages of each. One interesting algorithm to be implemented on 2 models of data sorting variants is the Bubble Sort algorithm, the reason is that this algorithm has a fairly long and detailed process flow to produce sequential data sequences from previously unsordered data sequences. And the purpose of implementing them using the language of the program is to be able to provide a translation picture of a complicated and long Bubble Sort algorithm that becomes easier and concise to solve using the program language, which in this case uses the Java program language. For ascending sequencing moving from left to right the method is by shifting the larger values to the right position in sequence, so that finally the smaller values are on the left, whereas for descending sorting moving from left to right the method is by shifting the smaller values to the right position in sequence, so that finally the larger values are on the left. And the programming technique differs only in relation operators that are used against the data being compared.

Keywords: *Bubble Sort, Ascending, Descending, Variants, Java*

ABSTRAK

Dalam bidang ilmu komputer upaya yang dilakukan dalam pemecahan suatu masalah adalah dengan menggunakan alur dan langkah-langkah yang sistematis, logis, dan strategis. Langkah-langkah ini dikenal dengan nama algoritma. Begitu banyak algoritma yang ada untuk pemecahan suatu masalah tertentu. Berkaitan dengan tulisan ini algoritma yang dibahas adalah tentang algoritma pengurutan data, diantara beberapa yang dikenal adalah: Merge, Selection, Insertion, Bubble, Quick, Heap, Shell.Seluruh algoritma tersebut memiliki kelebihan dan kekurangan masing-masing. Salah satu algoritma yang menarik untuk diimplementasikan terhadap 2 buah model varian pengurutan data adalah algoritma Bubble Sort, alasannya adalah bahwa algoritma ini memiliki alur proses yang cukup panjang dan detail untuk menghasilkan deretan data terurut dari deretan data tidak terurut sebelumnya.Dan tujuan dari peng-implementasiannya menggunakan bahasa program adalah agar dapat memberikan gambaran penterjemahan dari suatu algoritma Bubble Sort yang rumit dan panjang menjadi lebih mudah dan ringkas penyelesaiannya menggunakan bahasa program, yang dalam hal ini menggunakan bahasa program Java. Untuk pengurutan secara Ascending bergerak dari kiri ke kanan metodenya adalah dengan cara menggeser nilai-nilai yang lebih besar ke posisi sebelah kanan secara berurutan, sehingga akhirnya nilai yang lebih kecil berada di sebelah kiri, Sedangkan untuk pengurutan secara Descending bergerak dari kiri ke kanan metodenya adalah dengan cara menggeser nilai-nilai yang lebih kecil ke posisi sebelah kanan secara berurutan, sehingga akhirnya nilai yang lebih besar berada di sebelah kiri. Dan dalam teknik pemrogramannya hanya berbeda pada operator relasi yang digunakan terhadap data-data yang dibandingkan.

Kata kunci: *Bubble Sort, Ascending, Descending, Varian, Java*

Vol. 13, No. 2, September 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i2.1008

1. PENDAHULUAN

Berbagai pola algoritma pengurutan data yang ada memiliki kelebihan dan kekurangan masing-masing. Penggunaan masing-masing algoritma tersebut sangat tergantung dari kebutuhan dan kondisi yang ada. Umumnya pemilihan suatu algoritma pengurutan data didasarkan pada kesederhanaan / keringkasan alurnya dan kecepatan proses eksekusinya. Beberapa metode pengurutan data yang ada diantaranya adalah Selection Sort, Insertion Sort, Bubble Sort, Merge Sort, Quick Sort, Shell Sort, Heap Sort, dan Radix Sort. Dalam penelitian ini kajian yang dilakukan adalah tentang implementasi algoritma Bubble Sort terhadap 2 buah model varian pengurutan data. Pertimbangan memilih algoritma Bubble Sort adalah karena algoritma ini adalah salah satu algoritma pengurutan data yang memiliki alur cukup panjang dan detail dalam setiap langkahnya untuk menghasilkan deretan data terurut dari deretan data tidak terurut sebelumnya. Kemudian dalam algoritma ini dikenal dengan apa yang namanya Tahap dan Langkah. Tahap adalah fase pengurutan data pada suatu periode tertentu, sedangkan Langkah adalah urutan-urutan pengurutan yang berada pada setiap Tahap. Ilustrasinya adalah sbb:

Tahap 1 :
- Langkah 1 :
- Langkah 2 :
- …….
- Langkah (n-1) :

Tahap 2 :
- Langkah 1 :
- Langkah 2 :
- …….
- Langkah (n-2) :

Tahap m :
- Langkah 1 :
- Langkah 2 :
- …….
- Langkah (n-m) :

Hasil yang diharapkan / keunikan dari penelitian ini adalah dapat terlihat dengan jelas gambaran alur logika dari 2 buah model varian pengurutan data (Ascending dan Descending) menggunakan algoritma Bubble Sort yang diterjemahkan menggunakan bahasa program Java, bahwa perbedaan dalam teknik pemrogramannya sangat sederhana, dibalik alur logika dari 2 buah model varian pengurutan data yang rumit.

Penelitian sebelumnya yang pernah ada adalah “Penggunaan Algoritma Sorting Bubble Sort Untuk Penentuan Nilai Prestasi Siswa” [1]. Dalam penelitian ini digambarkan bagaimana kumpulan data nilai dari setiap siswa dikumpulkan dan diurutkan secara Descending dengan algoritma Bubble Sort untuk menentukan urutan ranking prestasi siswa, menggunakan bahasa program C++. Penelitian selanjutnya adalah “Analisis Perbandingan Algoritma Bubble Sort, Merge Sort, dan Quick Sort dalam Proses Pengurutan Kombinasi Angka dan Huruf” [2]. Dalam penelitian ini menjelaskan tentang perbandingan dari ke-tiga algoritma pengurutan data tersebut dalam hal mengurutkan kombinasi angka dan huruf. Selanjutnya adalah penelitian dengan judul

Vol. 13, No. 2, September 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i2.1008

"Perbandingan Kecepatan Gabungan Algoritma Utama Quick Sort dan Merge Sort dengan Algoritma Tambahan Insertion Sort, Bubble Sort dan Selection Sort" [3]. Dalam pembahasan ini dijelaskan bahwa algoritma Bubble Sort memiliki kelebihan sederhana dan mudah diimplementasikan, dan algoritma ini memiliki kompleksitas waktu Average Case dan Worst Case yaitu O (n^2).

2. METODE / PERANCANGAN PENELITIAN

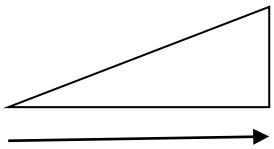
Pada penelitian ini, digunakan metode algoritma Bubble Sort terhadap 2 buah model varian pengurutan data yaitu pengurutan data secara : Ascending bergerak dari Kiri ke Kanan, dan Descending bergerak dari Kiri ke Kanan. Untuk memudahkan gambaran tentang alur proses dari algoritma Bubble Sort terhadap 2 buah model varian pengurutan data ini, kita menggunaan deretan data array seperti di bawah. Dan deretan data yang digunakan hanya 6 buah saja dengan alasan bahwa jika deretan datanya lebih banyak maka penjelasan alur proses dari algoritma Bubble Sort ini akan menjadi lebih banyak dan lebih panjang. Berikut deretan datanya:

Index	0	1	2	3	4	5
Urutan	1	2	3	4	5	6
Data Awal	32	15	3	88	41	37

Dan pembahasan algoritma Bubble Sort dari masing-masing varian akan dibahas pada setiap sub-bab di bawah ini:

2.1. Ascending dari Kiri ke Kanan

Algoritma Bubble Sort dalam metode pengurutan data varian model ini adalah bahwa alur proses akan mendorong nilai data yang paling besar ke posisi paling kanan, demikian seterusnya untuk nilai data yang besar berikutnya diletakkan di posisi sebelah kiri dari nilai data terbesar sebelumnya yang sudah diletakan di posisi sebelah kanan. Gambaran ilustrasinya adalah seperti ini :



<u>Tahap 1 : (sumber datanya adalah Data Awal)</u>

1. Data (1) dan data (2) dibandingkan, jika data (2) < dari data (1) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
2. Data (2) dan data (3) dibandingkan, jika data (3) < dari data (2) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
3. Data (3) dan data (4) dibandingkan, jika data (4) < dari data 3) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
4. Data (4) dan data (5) dibandingkan, jika data (5) < dari data 4) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.

Vol. 13, No. 2, September 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i2.1008

5. Data (5) dan data (6) dibandingkan, jika data (6) < dari data 5) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.

Hasil dari pengurutan Tahap 1 adalah :

Urutan	1	2	3	4	5	6
Hasil Tahap 1	15	3	32	41	37	88

Hasil dari algoritma Tahap 1 ini adalah bahwa proses algoritma menempatkan / mendorong nilai yang paling besar ke posisi paling kanan.

Tahap 2 : (sumber datanya adalah Hasil Tahap 1)

1. Data (1) dan data (2) dibandingkan, jika data (2) < dari data (1) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
2. Data (2) dan data (3) dibandingkan, jika data (3) < dari data (2) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
3. Data (3) dan data (4) dibandingkan, jika data (4) < dari data 3) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
4. Data (4) dan data (5) dibandingkan, jika data (5) < dari data 4) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.

Hasil dari pengurutan Tahap 2 adalah :

Urutan	1	2	3	4	5	6
Hasil Tahap 2	3	15	32	37	41	88

Tahap 3 : (sumber datanya adalah Hasil Tahap 2)

1. Data (1) dan data (2) dibandingkan, jika data (2) < dari data (1) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
2. Data (2) dan data (3) dibandingkan, jika data (3) < dari data (2) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
3. Data (3) dan data (4) dibandingkan, jika data (4) < dari data 3) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.

Hasil dari pengurutan Tahap 3 adalah :

Urutan	1	2	3	4	5	6
Hasil Tahap 3	3	15	32	37	41	88

Tahap 4 : (sumber datanya adalah Hasil Tahap 3)

Vol. 13, No. 2, September 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i2.1008

1. Data (1) dan data (2) dibandingkan, jika data (2) < dari data (1) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
2. Data (2) dan data (3) dibandingkan, jika data (3) < dari data (2) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.

Hasil dari pengurutan Tahap 4 adalah :

Urutan	1	2	3	4	5	6
Hasil Tahap 4	3	15	32	37	41	88

Tahap 5 : (sumber datanya adalah Hasil Tahap 4)

1. Data (1) dan data (2) dibandingkan, jika data (2) < dari data (1) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.

Hasil dari pengurutan Tahap 5 adalah :

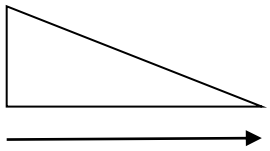
Urutan	1	2	3	4	5	6
Hasil Tahap 5	3	15	32	37	41	88

Sampai dengan Tahap 5 ini (terakhir) proses pengurutan data Ascending yang bergerak dari Kiri ke Kanan selesai. Disini bisa dianalisa kaitan antara Tahap, Langkah, dan Jumlah Data. Jika misal variable Tahap = T, variable Langkah = H, dan variable Jumlah Data = N, maka dapat dibuat persamaan matematika sebagai berikut :

$N = T + H$, dan $T = N - H$, dan $H = N - T$

2.2. Descending dari Kiri ke Kanan

Algoritma Bubble Sort dalam metode pengurutan data varian model ini adalah kita akan mendorong nilai data yang paling kecil ke posisi paling kanan, demikian seterusnya untuk nilai data yang kecil berikutnya diletakkan di posisi sebelah kiri dari nilai terkecil sebelumnya yang sudah diletakkan di posisi sebelah kanan. Gambaran ilustrasinya adalah seperti ini :



Tahap 1 : (sumber datanya adalah Data Awal)

1. Data (1) dan data (2) dibandingkan, jika data (2) > dari data (1) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
2. Data (2) dan data (3) dibandingkan, jika data (3) > dari data (2) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
3. Data (3) dan data (4) dibandingkan, jika data (4) > dari data (3) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.

Vol. 13, No. 2, September 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i2.1008

4. Data (4) dan data (5) dibandingkan, jika data (5) > dari data (4) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
5. Data (5) dan data (6) dibandingkan, jika data (6) > dari data (5) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.

Hasil dari pengurutan Tahap 1 adalah :

Urutan	1	2	3	4	5	6
Hasil Tahap 1	32	15	88	41	37	3

Tahap 2 : (sumber datanya adalah Hasil Tahap 1)

1. Data (1) dan data (2) dibandingkan, jika data (2) > dari data (1) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
2. Data (2) dan data (3) dibandingkan, jika data (3) > dari data (2) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
3. Data (3) dan data (4) dibandingkan, jika data (4) > dari data (3) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
4. Data (4) dan data (5) dibandingkan, jika data (5) > dari data (4) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.

Hasil dari pengurutan Tahap 2 adalah :

Urutan	1	2	3	4	5	6
Hasil Tahap 2	32	88	41	37	15	3

Tahap 3 : (sumber datanya adalah Hasil Tahap 2)

1. Data (1) dan data (2) dibandingkan, jika data (2) > dari data (1) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
2. Data (2) dan data (3) dibandingkan, jika data (3) > dari data (2) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
3. Data (3) dan data (4) dibandingkan, jika data (4) > dari data (3) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.

Hasil dari pengurutan Tahap 3 adalah :

Urutan	1	2	3	4	5	6
Hasil Tahap 3	88	41	37	32	15	3

Vol. 13, No. 2, September 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i2.1008

Tahap 4 : (sumber datanya adalah Hasil Tahap 3)

1. Data (1) dan data (2) dibandingkan, jika data (2) > dari data (1) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.
2. Data (2) dan data (3) dibandingkan, jika data (3) > dari data (2) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.

Hasil dari pengurutan Tahap 4 adalah :

Urutan	1	2	3	4	5	6
Hasil Tahap 4	88	41	37	32	15	3

Tahap 5 : (sumber datanya adalah Hasil Tahap 4)

1. Data (1) dan data (2) dibandingkan, jika data (2) > dari data (1) maka saling ditukarkan posisinya, jika tidak maka posisinya tetap tidak ada pertukaran. Posisi data lainnya tetap, tidak ada pertukaran.

Hasil dari pengurutan Tahap 5 adalah :

Urutan	1	2	3	4	5	6
Hasil Tahap 5	88	41	37	32	15	3

Sampai dengan Tahap 5 ini (terakhir) proses pengurutan data Descending yang bergerak dari Kiri ke Kanan selesai. Disini bisa dianalisa kaitan antara Tahap, Langkah, dan Jumlah Data. Jika misal variable Tahap = T, variable Langkah = H, dan variable Jumlah Data = N, maka dapat dibuat persamaan matematika sebagai berikut :

N = T + H, dan T = N – H, dan H = N - T

3. HASIL DAN PEMBAHASAN

Untuk penggambaran lebih jelas mengenai analisis algoritma Bubble Sort yang dimaksud, dapat dilihat dari listing program Java di bawah ini, dimana terdapat 2 buah model varian pengurutan datanya, yaitu pengurutan data secara Ascending bergerak dari Kiri ke Kanan, dan pengurutan data secara Descending bergerak dari Kiri ke Kanan. Berikut listing programnya :

3.1. Ascending Kiri ke Kanan

❖ **Program Java – Pengurutan Data secara Ascending bergerak dari Kiri ke Kanan :**

```
package analisa_algoritma;
/** Program Pengurutan Data Metode BUBBLE SORT
 * Bandung - 28 Mei 2020
 * @author Ir. Endang Sunandar, M.Kom
 */
public class BubbleSort_Ascending {
   public static void main(String args[]){
      int data[]={32,15,3,88,41,37};
      int pd=data.length;int buffer=0;int lp=0;
      System.out.println("Algoritma BUBBLE SORT - Pengurutan ASCENDING");
      System.out.println("Data Awal adalah : 32,15,3,88,41,37");
      System.out.println("Panjang data array adalah : "+pd+" data");
```

Vol. 13, No. 2, September 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i2.1008

```
      System.out.println("--------------------------------------------");
      for (int r=0;r<(pd-1);r++){
        lp=r+1;System.out.println("Hasil Tahap ke - "+(lp)+" : ");
        for (byte i=0;i<(pd-lp);i++){
          if (data[i]>data[i+1]){
            buffer=data[i];
            data[i]=data[i+1];
            data[i+1]=buffer;
           }//akhir if
          System.out.print(" Langkah ke - "+(i+1)+" ---> ");
          for (int buf=0;buf<pd;buf++){
            System.out.print(data[buf]+"  ");
          }//akhir for buf
          System.out.println("");
         }//akhir for i
        System.out.println("");
      }//akhir looping r
      System.out.println("--------------------------------------------");
    }//akhir public static void
  }//akhir public class
```

❖ **<u>Hasil Output dari eksekusi program diatas adalah :</u>**

```
Algoritma BUBBLE SORT - Pengurutan ASCENDING
Data Awal adalah : 32,15,3,88,41,37
Panjang data array adalah : 6 data
--------------------------------------------
Hasil Tahap ke - 1 :
 Langkah ke - 1 ---> 15  32  3  88  41  37
 Langkah ke - 2 ---> 15  3  32  88  41  37
 Langkah ke - 3 ---> 15  3  32  88  41  37
 Langkah ke - 4 ---> 15  3  32  41  88  37
 Langkah ke - 5 ---> 15  3  32  41  37  88

Hasil Tahap ke - 2 :
 Langkah ke - 1 ---> 3  15  32  41  37  88
 Langkah ke - 2 ---> 3  15  32  41  37  88
 Langkah ke - 3 ---> 3  15  32  41  37  88
 Langkah ke - 4 ---> 3  15  32  37  41  88

Hasil Tahap ke - 3 :
 Langkah ke - 1 ---> 3  15  32  37  41  88
 Langkah ke - 2 ---> 3  15  32  37  41  88
 Langkah ke - 3 ---> 3  15  32  37  41  88

Hasil Tahap ke - 4 :
 Langkah ke - 1 ---> 3  15  32  37  41  88
 Langkah ke - 2 ---> 3  15  32  37  41  88

Hasil Tahap ke - 5 :
 Langkah ke - 1 ---> 3  15  32  37  41  88

--------------------------------------------
BUILD SUCCESSFUL (total time: 0 seconds)
```

Vol. 13, No. 2, September 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i2.1008

3.2. Descending Kiri ke Kanan

❖ **Program Java – Pengurutan Data secara Descending bergerak dari Kiri ke Kanan :**

```
package analisa_algoritma;
/** Program Pengurutan Data Metode BUBBLE SORT
 * Bandung - 28 Mei 2020
 * @author Ir. Endang Sunandar, M.Kom
 */
public class BubbleSort_Ascending {
   public static void main(String args[]){
      int data[]={32,15,3,88,41,37};
      int pd=data.length;int buffer=0;int lp=0;
      System.out.println("Algoritma BUBBLE SORT - Pengurutan DESCENDING");
      System.out.println("Data Awal adalah : 32,15,3,88,41,37");
      System.out.println("Panjang data array adalah : "+pd+" data");
      System.out.println("--------------------------------------------");
      for (int r=0;r<(pd-1);r++){
        lp=r+1;System.out.println("Hasil Tahap ke - "+(lp)+" : ");
        for (byte i=0;i<(pd-lp);i++){
          if (data[i]<data[i+1]){
            buffer=data[i];
            data[i]=data[i+1];
            data[i+1]=buffer;
           }//akhir if
          System.out.print(" Langkah ke - "+(i+1)+" ---> ");
          for (int buf=0;buf<pd;buf++){
            System.out.print(data[buf]+"  ");
          }//akhir for buf
          System.out.println("");
         }//akhir for i
        System.out.println("");
      }//akhir looping r
      System.out.println("--------------------------------------------");
   }//akhir public static void
  }//akhir public class
```

❖ **Hasil Output dari eksekusi program diatas adalah :**

```
Algoritma BUBBLE SORT - Pengurutan DESCENDING
Data Awal adalah : 32,15,3,88,41,37
Panjang data array adalah : 6 data
--------------------------------------------
Hasil Tahap ke - 1 :
 Langkah ke - 1 ---> 32  15  3  88  41  37
 Langkah ke - 2 ---> 32  15  3  88  41  37
 Langkah ke - 3 ---> 32  15  88  3  41  37
 Langkah ke - 4 ---> 32  15  88  41  3  37
 Langkah ke - 5 ---> 32  15  88  41  37  3

Hasil Tahap ke - 2 :
 Langkah ke - 1 ---> 32  15  88  41  37  3
 Langkah ke - 2 ---> 32  88  15  41  37  3
 Langkah ke - 3 ---> 32  88  41  15  37  3
 Langkah ke - 4 ---> 32  88  41  37  15  3
```

Vol. 13, No. 2, September 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i2.1008

```
Hasil Tahap ke - 3 :
 Langkah ke - 1 ---> 88  32  41  37  15  3
 Langkah ke - 2 ---> 88  41  32  37  15  3
 Langkah ke - 3 ---> 88  41  37  32  15  3

Hasil Tahap ke - 4 :
 Langkah ke - 1 ---> 88  41  37  32  15  3
 Langkah ke - 2 ---> 88  41  37  32  15  3

Hasil Tahap ke - 5 :
 Langkah ke - 1 ---> 88  41  37  32  15  3

---------------------------------------------
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. KESIMPULAN DAN SARAN

Berdasarkan hasil analisis yang dilakukan pada penggunaan algoritma Bubble Sort terhadap 2 buah model varian pengurutan data ini dapat disimpulkan beberapa hal sebagai berikut:

Pada model pengurutan data secara Ascending yang bergerak dari kiri ke kanan, alur proses yang dilakukan adalah secara bertahap melakukan proses penempatan nilai data terbesar ke posisi sebelah kanan. Demikian seterusnya hingga nilai data terkecil berada pada posisi paling kiri.

Pada model pengurutan data secara Descending yang bergerak dari kiri ke kanan, alur proses yang dilakukan adalah secara bertahap melakukan proses penempatan nilai data terkecil ke posisi sebelah kanan. Demikian seterusnya hingga nilai data terbesar berada pada posisi paling kiri.

Teknik pemrograman dalam bahasa Java untuk 2 buah model varian pengurutan data ini hampir persis sama, hanya ada beda sedikit pada penggunaan operation relationanl < atau > pada suatu data pada posisi [i] yang dibandingkan dengan data yang pada posisi [i+1].

Masih ada 2 buah model varian pengurutan data lainnya yaitu: Ascending bergerak dari Kanan ke Kiri, dan Descending bergerak dari Kanan ke Kiri. Jika melihat ilustrasi obyek masing-masing pengurutan maka model varian pengurutan data Ascending yang bergerak dari Kiri ke Kanan = model varian pengurutan data Descending yang bergerak dari Kanan ke Kiri. Dan model varian pengurutan data Descending yang bergerak dari Kiri ke Kanan = model varian pengurutan data Ascending yang bergerak dari Kanan ke Kiri. Dan dalam teknik pemrograman komputernya hanya dibedakan pada penggunaan operator relasi (< atau >) dan posisi start pergerakan pembandingan data (dari kiri atau dari kanan).

DAFTAR PUSTAKA

Beberapa sumber pustaka untuk penelitian ini adalah:

[1] Gunawan Indra, Sumarno, Tambunan Heru Satria, *"Penggunaan Algoritma Sorting Bubble Sort Untuk Penentuan Nilai Prestasi Siswa",* Jurnal Sistem Informasi, STIKOM Tunas Bangsa Pematang Siantar. Vol.8, No.2, Mei 2019.

[2] Sonita Anisya, Nurtaneo Febrian, *"Analisis Perbandingan Algoritma Bubble Sort, Merge Sort, dan Quick Sort dalam Proses Pengurutan Kombinasi Angka dan Huruf"*, Jurnal Pseudocode, Program Studi Teknik Informatika, Fakultas Teknik Universitas Bengkulu Vol. 2, No.2, 2015.

Vol. 13, No. 2, September 2020, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v13i2.1008

[3] Al Rivan Muhammad Ezar, *"Perbandingan Kecepatan Gabungan Algoritma Utama Quick Sort dan Merge Sort dengan Algoritma Tambahan Insertion Sort, Bubble Sort dan Selection Sort"*, Jurnal Teknik Informatika dan Sistem Informasi, Vol.3 No.2 Agustus 2017.

[4] Reina, Gautama Josef Bernadi, *"Perbandingan Bubble Sort Dengan Insertion Sort Pada Bahasa Pemrograman C Dan Fortran"*, Jurnal ComTech, Vol.4, No.2, Desember 2013.

[5] Prayitno Prayitno, Basukesti Agus, Indrianingsih Yuliani, "*Load Balancing Menggunakan Metode Bubble Sort Pada Sistem Database Dengan Multi Server"*, Compiler, Vol.2, No.2, 2013.

[6] Sitepu Roma Rio, Yusman Machudor, Febriansyah Febi Eka, "*Implementasi Algoritma Bubble Sort dan Selection Sort Menggunakan Arraylist Multidimensi Pada Pengurutan Data Multi Prioritas*", Jurnal Komputasi Ilmu Komputer Unila, Vol.5 No.1 2017.

[7] Rahayuningsih, *"Analisis Perbandingan Kompleksitas Algoritma Pengurutan Nilai (Sorting)"*, Evolusi : Jurnal Sains dan Manajemen, Universitas Bina Sarana Informatika, Vol. 4 No. 2, 2016.

[8] Shabaz Mohammad, Kumar Ashok, *"SA Sorting: A Novel Sorting Technique for Large-Scale Data"*, Hindawi Journal of Computer Networks and Communications, Vol. 2019,

[9] Akhter Naeem, Idrees Muhammad, Rehman-Furqan-Ur, *"Sorting Algorithms – A Comparative Study"*, International Journal of Computer Science and Information Security (IJCSIS), Vol.14, No.12, December 2016.

[10] Singh Shashank, Sing Rajesh Shyam, Mandoria Hardwari Lal, *"Left-Right Sort : A Novel Sorting Algorithm And Comparison With Insertion Sort, Bubble Sort And Selection Sort"*, International Journal of Engineering & Scientific Research, Vol.3, Issue 10, October 2015.

[11] Rohil Harish, Manisha, *"Run Time Bubble Sort – An Enhancement of Bubble Sort"*, International Journal of Computer Trends and Technology (IJCTT), Vol.14, No.1-Aug 2014.

[12] Saadeh Reem, Qatawneh Mohammad, *"Performance Evaluation of Parallel Bubble Sort Algorihtm On Supercomputer IMan1"*, International Journal of Computer Science & Information Technology (IJCSIT), Vol.11, No.3, June 2019.

[13] Wang Shiying, Wang Mujiangshan, "*The Strong Connectivity of Bubble-Sort Star Graphs",* The Computer Journal, Vol.62, Issue 5, 15 August 2018.

[14] Swain Debabrata, Ramkrishna G., Mahapatra Hitesh, Patro Pramoda, Dhanrao Pravin M.*,"A Novel Sorting Technique to Sort Elements in Ascending Order",* International Journal of Engineering and Advanced Technology (IJEAT), Volume-3, Issue-1, October 2013,

[15] Ms. Mundra Jyoti, Mr. Pal B. L.,*" Minimizing Execution Time of Bubble Sort Algorithm"*, International Journal of Computer Science and Mobile Computing, Vol.4, Issue.9, 2015.