

Analisis Algoritma Bubble Sort Secara Ascending Dan Descending Serta Implementasinya Menggunakan Bahasa Pemrograman Java

Nilasari¹, Welnaldo Abdi Gunawan², Putri Khofipah Sari³, Ismal Zikri⁴, Andrian Syahputra⁵

^{1,2,3,4,5}Fakultas Teknologi dan Ilmu Komputer, Universitas Potensi Utama, Medan, Indonesia

Abstrak

Banyak algoritma pengurutan data pada pemrograman tentu saja menjadi hal yang membingungkan bagi sebagian orang untuk menentukan algoritma pengurutan data mana yang cocok dipakai dalam membuat programnya terutama bagi yang baru memasuki ranah dunia pemrograman. Dengan demikian, maka diperlukannya informasi setiap algoritma yang ada. Pada pembahasan ini, penulis memilih algoritma Bubble Sort sebagai objek bahasan. Tujuan dari penelitian ini untuk mengetahui efisiensi dari algoritma Bubble Sort dalam melakukan sorting data sehingga penulis berharap dapat menjadi sumber informasi untuk umum maupun penulis mengenai algoritma Bubble Sort sehingga kita mampu menyesuaikan saat apa algoritma ini dipakai. Menggunakan metode algoritma Bubble Sort dengan dua varian pengurutan data yaitu Ascending dan Descending, proses sorting algoritma Bubble Sort tidak memiliki banyak perbedaan yang spesifik.

Kata Kunci: : CIPP, Program Magang, Siswa Menengah Kejuruan (SMK)

1. Pendahuluan

1.1 Latar Belakang

Pengurutan data atau “Sorting” adalah suatu proses menyusun kembali data yang sebelumnya telah tersusun secara acak hingga tersusun secara teratur menurut aturan tertentu[1]. Data ini biasanya bertipe numerik dan karakter.

Pengurutan data memiliki banyak fungsi dalam sebuah sistem. Data yang terurut mudah untuk dicari, mudah untuk diperiksa, dan mudah untuk diperbaiki jika terjadi kesalahan. Data yang terurut dengan baik juga mudah untuk dihapus jika sewaktu-waktu data tersebut tidak diperlukan lagi. Selain itu, dengan mengurutkan data maka kita semakin mudah untuk menyisipkan data ataupun melakukan penggabungan. Pengurutan data adalah bagian integral dari analisis data. Setiap algoritma dalam perancangan perangkat lunak pasti membutuhkan sebuah data, baik data analog maupun data digital[2], yang mana data tersebut banyak berupa numerik bahkan huruf yang akan dikelola sehingga menjadi hasil yang diinginkan.

Kita mungkin ingin menyusun daftar nama dalam urutan abjad, menyusun daftar tingkatan inventaris produk dari tertinggi ke terendah, atau mengurutkan baris menurut warna atau ikon. Pengurutan data membantu memvisualisasikan dengan cepat dan memahami data kita dengan lebih baik, menata dan menemukan data yang kita inginkan dan akhirnya membuat keputusan yang lebih efektif.

Pengurutan data dapat dilakukan dengan 2 model varian pengurutan data yaitu secara Ascending dan Descending[3]. Kita dapat mengurutkan data menurut teks dari A ke Z atau Z ke A, data angka dari terkecil ke terbesar atau terbesar ke terkecil, serta data tanggal dan waktu dari terlama ke terbaru atau terbaru ke terlama dalam satu atau beberapa kolom.

Terdapat macam-macam algoritma yaitu bubble sort, insertion sort, selection sort, shell sort, merge sort, quick sort, dan heap sort. Pada penelitian ini hanya akan dibahas 1 metode pengurutan data yaitu algoritma bubble sort untuk membatasi luasnya pembahasan. Hal ini dikarenakan algoritma bubble sort adalah algoritma pengurutan paling dasar. Algoritma ini memiliki metode pengurutan paling sederhana daripada algoritma pengurutan yang lain sehingga mudah dipahami terutama bagi yang baru memasuki ranah pemrograman. Selain itu, jumlah pengulangan akan tetap sama jumlahnya meskipun data sudah cukup terurut. Sehingga, penulis memutuskan untuk memilih algoritma bubble sort sebagai objek penelitian.

1.2 Tujuan Penelitian

Dari pembahasan di atas, tujuan yang ingin dicapai adalah dari penelitian ini adalah sebagai berikut :

- Menerapkan algoritma bubble sort dalam pengurutan data berupa angka menggunakan bahasa pemrograman java.
- Memahami fungsi syntax yang diterapkan dalam kode program.
- Memahami proses pengurutan data terdapat 2 model varian pengurutan data yaitu ascending dan descending.

1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut :

- Algoritma bubble sort digunakan untuk sorting angka secara ascending dan descending.
- Implementasi algoritma menggunakan aplikasi Netbeans pada bahasa pemrograman Java, Pengujian dalam pengurutan ini hanya menggunakan algoritma bubble sort.

[†]E-mail: nilasari03120@gmail.com

welnaldoabdiGunawan02@gmail.com

zikriismal42@gmail.com

putrikhofipah02@gmail.com

andriansyahputra4@gmail.com

2. Metode Penelitian

Metode pada penelitian ini yaitu menggunakan Algoritma Bubble Sort terhadap dua buah varian model pengurutan data. Pertama secara Ascending dari data terkecil ke data terbesar dan Descending dari data terbesar ke data terkecil, dengan menggunakan deretan data Array pada masing-masing tabel (tertera pada setiap pembahasan).

2.1 Algoritma Bubble Sort

Bubble Sort adalah metode pengurutan data dengan cara melakukan penukaran data dari data pertama dengan data di sebelahnya secara terus menerus sampai bisa dipastikan dalam suatu iterasi tertentu tidak ada lagi perubahan atau penukaran. Algoritma ini menggunakan perbandingan dalam operasi antar elemennya.

Algoritma Bubble Sort adalah algoritma pengurutan paling dasar serta memiliki metode pengurutan paling sederhana daripada algoritma pengurutan yang lain[4]. Proses pencarian solusi dilakukan secara brute force, langsung ke intinya yaitu membandingkan elemen-elemen dalam tabel.

Algoritma Bubble Sort merupakan proses pengurutan yang secara berangsur-angsur memindahkan data ke posisi yang tepat. Karena itulah algoritma ini dinamakan “Bubble” atau yang jika diterjemahkan ke dalam Bahasa Indonesia artinya gelembung[5]. Fungsi algoritma ini adalah untuk mengurutkan data dari yang terkecil ke yang terbesar (Ascending) atau sebaliknya (Descending).

Sebelum kita masuk untuk membuat program, berikut adalah syarat dan langkah-langkah yang harus diperhatikan pada metode Bubble Sort.

- Jumlah iterasi sama dengan banyaknya bilangan dikurang 1.
- Di setiap iterasi, jumlah pertukaran bilangannya sama dengan jumlah banyaknya bilangan.
- Dalam algoritma Bubble Sort, meskipun deretan bilangan tersebut sudah terurut, proses sorting akan tetap dilakukan.
- Tidak ada perbedaan cara yang berarti untuk teknik algoritma Bubble Sort Ascending dan Descending.

Untuk mempelajari algoritma Bubble Sort ini kita hanya perlu memahami cara yang digunakan untuk mengurutkan data. Logika sederhananya, algoritma ini menggunakan perbandingan dalam operasi antar elemennya. Algoritma Bubble Sort ini mempunyai kelebihan dan kekurangan. Dua hal inilah yang menjadi pertimbangan programmer ketika membuat program. Berikut beberapa kelebihan yang dimiliki oleh algoritma Bubble Sort.

- Algoritma ini adalah metode paling sederhana untuk mengurutkan data.
- Bubble Sort adalah algoritma yang mudah dipahami.
- Sedangkan kekurangan dari algoritma Bubble Sort adalah sebagai berikut.
- Tingkat efisiensinya yang kurang. Bubble Sort ini merupakan metode pengurutan yang tidak efisien, khususnya ketika menangani data yang jumlahnya besar. Hal ini karena ketika mengurutkan data yang sangat besar akan sangat lambat prosesnya.
- Jumlah pengulangan yang dilakukan oleh algoritma ini akan tetap sama jumlahnya meskipun data yang diurutkan sudah cukup terurut[6].

Dengan beberapa faktor di atas, kita dapat menyesuaikan saat kapan sebaiknya algoritma Bubble Sort ini digunakan.

2.2 Cara Kerja Algoritma Bubble Sort

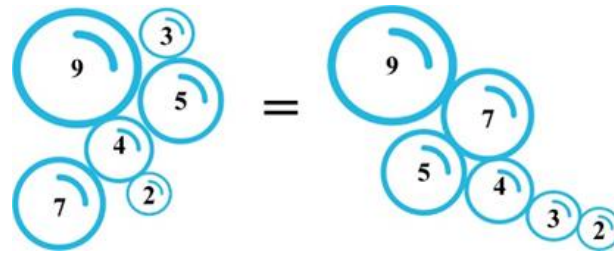
Cara kerja Bubble Sort hampir sama dengan Selection Sort, yang membedakan adalah data yang dibandingkan. Pada Algoritma Bubble Sort, data yang dibandingkan adalah data yang berdekatan. Dasar algoritma Bubble Sort ini sangat mudah, setiap data di dalam koleksi dibandingkan dengan data berikutnya. Misalnya data pada indeks ke-0 dibandingkan dengan data pada indeks ke-1 atau data indeks ke-1 dibandingkan dengan data indeks ke-2, begitu seterusnya sampai data tidak bisa dibandingkan lagi. Tujuan membandingkan data-data tersebut adalah untuk memastikan urutan di antara kedua data sudah sesuai atau tidak.

Pada data yang bertipe char, nilai data dikatakan lebih kecil atau lebih besar dari yang lain didasarkan pada urutan relatif (collating sequence) seperti dinyatakan dalam tabel ASCII.

ASCII control characters				ASCII printable characters				Extended ASCII characters			
00	NULL	(Null character)		32	space	64	@	96	-	128	C
01	SOH	(Start of Header)		33	!	65	A	97	a	129	D
02	STX	(Start of Text)		34	"	66	B	98	b	130	E
03	ETX	(End of Text)		35	#	67	C	99	c	131	F
04	EOT	(End of Trans.)		36	\$	68	D	100	d	132	G
05	ENQ	(Enquiry)		37	%	69	E	101	e	133	H
06	ACK	(Acknowledgement)		38	&	70	F	102	f	134	I
07	BEL	(Bell)		39	'	71	G	103	g	135	J
08	BS	(Backspace)		40	(72	H	104	h	136	K
09	HT	(Horizontal Tab)		41)	73	I	105	i	137	L
10	LF	(Line feed)		42	*	74	J	106	j	138	M
11	VT	(Vertical Tab)		43	+	75	K	107	k	139	N
12	FF	(Form feed)		44	,	76	L	108	l	140	O
13	CR	(Carriage return)		45	-	77	M	109	m	141	Q
14	SO	(Shift Out)		46	.	78	N	110	n	142	R
15	SI	(Shift In)		47	/	79	O	111	o	143	S
16	DLE	(Data link escape)		48	0	80	P	112	p	144	T
17	DC1	(Device control 1)		49	1	81	Q	113	q	145	U
18	DC2	(Device control 2)		50	2	82	R	114	r	146	V
19	DC3	(Device control 3)		51	3	83	S	115	s	147	W
20	DC4	(Device control 4)		52	4	84	T	116	t	148	X
21	NAK	(Negative acknowl.)		53	5	85	U	117	u	149	Y
22	SYN	(Synchronous idle)		54	6	86	V	118	v	150	Z
23	ETB	(End of trans. block)		55	7	87	W	119	w	151	[
24	CAN	(Cancel)		56	8	88	X	120	x	152	\
25	EM	(End of medium)		57	9	89	Y	121	y	153]
26	SUB	(Substitute)		58	:	90	Z	122	z	154	^
27	ESC	(Escape)		59	;	91	[123	{	155	_
28	FS	(File separator)		60	<	92	\	124		156	`
29	GS	(Group separator)		61	=	93]	125	}	157	~
30	RS	(Record separator)		62	>	94	^	126	~	158	
31	US	(Unit separator)		63	?	95	_	127	DEL	159	
127	DEL	(Delete)									

Gambar 1. ASCII

Ide dari metode pengurutan gelembung (Bubble Sort) ini terinspirasi oleh gelembung sabun yang berada di permukaan air. Karena berat jenis gelembung sabun yang lebih ringan ketimbang berat jenis air, maka gelembung sabun akan selalu terapung diatas permukaan. Prinsip inilah yang dipakai pada algoritma pengurutan gelembung



Gambar 2. Ilustrasi Gelembung Bubble Sort

Berikut adalah pseudocode Algoritma Bubble Sort dalam mengurutkan bilangan acak dari bilangan terkecil ke bilangan terbesar dengan metode bubble sort.

Deklarasi :

- A = Array [1...n] of integer {Data dalam larik ke-1 sampai n}.
- i : integer {Pencacah jumlah tahapan}.
- j : integer {Pencacah untuk langkah pada tiap tahapan}.
- temp : integer {Variabel sementara untuk pertukaran}.

Deskripsi :

```
for(int i=0; i<A.length-1; i++){
    for(int j=0; j<A.length-1; j++){
        if(A[j]>A[j+1]){
            {lakukan pertukaran}
                int temp=A[j+1];
                A[j+1]=A[j];
                A[j]=temp;
            end if
        }
    }
end for
```

Mirip dengan pseudocode dari selection sort, bubble sort juga menggunakan looping for bertingkat. Yang membedakan yaitu dimana yang ditukar (swap) adalah A[j] dan A[j + 1] (Array j dan Array j + 1) dengan kata lain array yang bersebelahan[7].

2.2.1 Pengurutan secara Ascending

Ascending atau biasa disebut urutan naik yaitu pengurutan data dari data terkecil ke data terbesar. Elemen yang lebih kecil ditukar posisinya dengan elemen yang lebih besar bila posisi elemen yang lebih kecil ada di bawah elemen yang lebih besar. Jika tabel belum terurut proses diulang kembali sampai elemen paling kecil berada di posisi teratas dan elemen lainnya sudah terurut. Contoh, misalnya terdapat 6 data acak yaitu 9, 4, 2, 5, 3 dan 7 dapat diurutkan menjadi 2, 3, 4, 5, 7, dan 9.

Berikut adalah langkah-langkah pengurutan data algoritma Bubble Sort dengan secara Ascending :

- Bandingkan nilai data ke-1 dan data ke-2.
- Jika data ke-1 lebih besar dari data ke-2 maka tukar posisinya.
- Kemudian data yang lebih besar tadi dibandingkan dengan data ke-3.
- Lakukan langkah kedua hingga selesai[8].

Ilustrasi konsep pertukaran data dari Algoritma Bubble Sort ini adalah sebagai berikut:

Iterasi 1 :

9	4	2	5	3	7
4	9	2	5	3	7
4	2	9	5	3	7
4	2	5	9	3	7
4	2	5	3	9	7
4	2	5	3	7	9

Iterasi 2 :

4	2	5	3	7	9
2	4	5	3	7	9
2	4	5	3	7	9
2	4	3	5	7	9
2	4	3	5	7	9
2	4	3	5	7	9

Iterasi 3 :

2	4	5	3	7	9
2	4	5	3	7	9
2	4	5	3	7	9
2	4	3	5	7	9
2	4	3	5	7	9
2	4	3	5	7	9

Iterasi 4 :

2	4	3	5	7	9
2	4	3	5	7	9
2	3	4	5	7	9
2	3	4	5	7	9
2	3	4	5	7	9
2	3	4	5	7	9

Iterasi 5 :						Hasil :					
2	3	4	5	7	9	2	3	4	5	7	9
2	3	4	5	7	9	2	3	4	5	7	9
2	3	4	5	7	9	2	3	4	5	7	9
2	3	4	5	7	9	2	3	4	5	7	9
2	3	4	5	7	9	2	3	4	5	7	9
2	3	4	5	7	9	2	3	4	5	7	9

Gambar 3. Pertukaran Data secara Ascending

Dapat dilihat pada Gambar 3 diatas proses pertukaran data secara Ascending.

2.2.2 Pengurutan secara Descending

Descending atau biasa disebut urutan turun yaitu pengurutan data dari data terbesar ke data terkecil. Elemen yang lebih besar ditukar posisinya dengan elemen yang lebih kecil bila posisi elemen yang lebih besar ada di bawah elemen yang lebih kecil. Jika tabel belum terurut proses diulang kembali sampai elemen paling besar berada di posisi teratas dan elemen lainnya sudah terurut. Contoh, misalnya terdapat 6 data acak yaitu 9, 4, 2, 5, 3 dan 7 dapat diurutkan menjadi 9, 7, 5, 4, 3, dan 2.

Berikut adalah langkah-langkah pengurutan data algoritma Bubble Sort dengan secara Descending :

- Bandingkan nilai data ke-1 dan data ke-2.
- Jika data ke-1 lebih kecil dari data ke-2 maka tukar posisinya.
- Kemudian data yang lebih kecil tadi dibandingkan dengan data ke-3.
- Lakukan langkah kedua hingga selesai.

Ilustrasi konsep pertukaran data dari Algoritma Bubble Sort ini adalah sebagai berikut

Iterasi 1 :						Iterasi 2 :					
9	4	2	5	3	7	9	4	5	3	7	2
9	4	2	5	3	7	9	4	5	3	7	2
9	4	2	5	3	7	9	5	4	3	7	2
9	4	5	2	3	7	9	5	4	3	7	2
9	4	5	3	2	7	9	5	4	7	3	2
9	4	5	3	7	2	9	5	4	7	3	2

Iterasi 3 :						Iterasi 4 :					
9	5	4	7	3	2	9	5	7	4	3	2
9	5	4	7	3	2	9	5	7	4	3	2
9	5	4	7	3	2	9	7	5	4	3	2
9	5	7	4	3	2	9	7	5	4	3	2
9	5	7	4	3	2	9	7	5	4	3	2
9	5	7	4	3	2	9	7	5	4	3	2

Iterasi 5 :						Hasil :					
9	7	5	4	3	2	9	7	5	4	3	2
9	7	5	4	3	2	9	7	5	4	3	2
9	7	5	4	3	2	9	7	5	4	3	2
9	7	5	4	3	2	9	7	5	4	3	2
9	7	5	4	3	2	9	7	5	4	3	2
9	7	5	4	3	2	9	7	5	4	3	2

Gambar 4. Pertukaran Data secara Descending

2.3 Kompleksitas Algoritma Bubble Sort

Kompleksitas sebuah algoritma Bubble Sort dapat dilihat dari beberapa jenis kasus, yaitu Worst-case, Average-case, dan Best-case.

2.3.1 Kondisi Worst-case

Worst-case (kondisi terburuk) yaitu kompleksitas dengan jumlah paling besar (T_{max}), adalah waktu terlama yang diperlukan suatu algoritma untuk memecahkan masalah pada n jumlah inputan. Analisis *worst case* memberikan batas atas atau *upper bound* dari *running time*[9].

Worst-case merupakan dasar dari analisis kompleksitas algoritma. *Worst-case* dinyatakan dalam notasi Big-O (O), karena *worst-case* merupakan upper bound dari sebuah algoritma. Suatu algoritma tidak mungkin dieksekusi dengan kompleksitas

waktu lebih dari *worst-case*. Pada kondisi ini, iterasi dilakukan sebanyak data $n-1$.

Contoh:

Iterasi 1 :

9	5	4	2
5	9	4	2
5	4	9	2
5	4	2	9

Iterasi 2 :

5	4	2	9
5	4	2	9
5	2	4	9
5	2	4	9

Iterasi 3 :

5	2	4	9
2	5	4	9
2	4	5	9
2	4	5	9

Gambar 5. Kondisi *Worst-case* (iterasi terjadi sebanyak 3 kali)

2.3.2 Kondisi *Average-case*

Average-case yaitu kompleksitas dengan jumlah rata-rata keseluruhan kemungkinan adalah waktu rata-rata yang diperlukan suatu algoritma dalam memecahkan masalah pada jumlah n jumlah inputan. *Average-case* dapat dinyatakan dalam notasi Big-O (O), karena *Average-case* terletak di antara *lower bound* dan *upper bound*. Biasanya nilai dari *average-case* mendekati *upper bound*-nya, tapi juga bisa lebih dekat dengan *lower bound*.

Contoh:

Iterasi 1 :

9	4	2	5
4	9	2	5
4	2	9	5
4	2	5	9

Iterasi 2 :

4	2	5	9
2	4	5	9
2	4	5	9
2	4	5	9

Gambar 6. Kondisi *Average-case* (iterasi terjadi sebanyak 2 kali)

2.3.3 Kondisi *Best-case*

Best-case (kondisi terbaik) kompleksitas dengan jumlah paling kecil (T_{min}), adalah waktu tercepat yang diperlukan suatu algoritma dalam memecahkan masalah pada n jumlah inputan. Hasil dari analisis ini adalah *lower bound* atau batas bawah *running time* suatu algoritma.

Base-case dapat dinyatakan dalam notasi Big-Omega(Ω) walaupun sering juga dinyatakan dalam notasi Big-O(O), karena *Best-case* merupakan *lower bound* dari kompleksitas waktu sebuah algoritma [10]. Suatu algoritma tidak mungkin dieksekusi dengan kompleksitas waktu kurang dari waktu best case. Pada kondisi ini, iterasi dilakukan hanya sekali.

Contoh :

Iterasi 1 :

2	4	5	9
2	4	5	9
2	4	5	9
2	4	5	9

Gambar 7. Kondisi *Average-case* (iterasi terjadi sebanyak 1 kali)

2.4 Implementasi Algoritma Bubble Sort

Berikut adalah implementasi algoritma Bubble Sort dengan menggunakan bahasa pemrograman Java dengan menggunakan Netbeans IDE 8.2 RC.

2.4.1 Ascending

Berikut adalah implementasi algoritma Bubble Sort secara Ascending. Gunakan method untuk menampilkan data yang telah di-sorting dengan menggunakan algoritma Bubble Sort seperti berikut [11].

```
public static void main(String[] args) {
    //deklarasi variabel
    int[] angka={9,4,2,5,3,7};

    //data sebelum di-sorting
    System.out.println("Angka sebelum di-sorting "+Arrays.toString(angka));
    bubblesort(angka);

    System.out.print("Angka setelah di-sorting ");
    print(angka);
}
```


Setelah selesai pada kode program diatas, kemudian buat method untuk menampung proses algoritma Bubble Sort seperti berikut [12].

```
//function bubblesort
public static int[] bubblesort(int[] A){
    //proses mee bubble sort
    for(int i=0; i<A.length-1; i++){
        System.out.println("Iterasi "+(i+1));
        for(int j=0; j<A.length-1; j++){
            if(A[j]>A[j+1]){
                int temp=A[j+1];
                A[j+1]=A[j];
                A[j]=temp;
            }
            System.out.println(Arrays.toString(A));
        }
        System.out.println();
    }
    return A;
}
```

Kemudian untuk langkah terakhir gunakan method print untuk menampilkan proses iterasi data yang di-sorting seperti berikut [13].

2.4.2 Descending

Implementasi algoritma Bubble Sort secara Descending hampir sama dengan Ascending. Perbedaannya terletak pada proses data yang dibandingkan [14].

Jika pada Ascending $\text{if}(A[j]>A[j+1])$, maka pada Descending kebalikan dari Ascending yaitu $\text{if}(A[j]<A[j+1])$. Dapat dilihat pada kode program berikut [15].

```
//function bubblesort
public static int[] bubblesort(int[] A){
    //proses mee bubble sort
    for(int i=0; i<A.length-1; i++){
        System.out.println("Iterasi "+(i+1));
        for(int j=0; j<A.length-1; j++){
            if(A[j]<A[j+1]){
                int temp=A[j+1];
                A[j+1]=A[j];
                A[j]=temp;
            }
            System.out.println(Arrays.toString(A));
        }
        System.out.println();
    }
    return A;
}
```

3. Hasil dan Pembahasan

Proses pengujian algoritma Bubble Sort ini menggunakan 6 data angka random yang sudah ditampung ke dalam array. Hanya menggunakan 6 data agar proses iterasi yang ditampilkan tidak terlalu banyak [16].

Berikut adalah hasil pengujian sorting data secara *Ascending*, iterasi terjadi sebanyak 5 kali iterasi hingga data terurut secara *Ascending* [17] [18].

```
run:
Angka sebelum di-sorting [9, 4, 2, 5, 3, 7]
Iterasi 1
[4, 9, 2, 5, 3, 7]
[4, 2, 9, 5, 3, 7]
[4, 2, 5, 9, 3, 7]
[4, 2, 5, 3, 9, 7]
[4, 2, 5, 3, 7, 9]

Iterasi 2
[2, 4, 5, 3, 7, 9]
[2, 4, 5, 3, 7, 9]
[2, 4, 3, 5, 7, 9]
[2, 4, 3, 5, 7, 9]
[2, 4, 3, 5, 7, 9]
```

```
Iterasi 3
[2, 4, 3, 5, 7, 9]
[2, 3, 4, 5, 7, 9]
[2, 3, 4, 5, 7, 9]
[2, 3, 4, 5, 7, 9]
[2, 3, 4, 5, 7, 9]
```

```
Iterasi 4
[2, 3, 4, 5, 7, 9]
[2, 3, 4, 5, 7, 9]
[2, 3, 4, 5, 7, 9]
[2, 3, 4, 5, 7, 9]
[2, 3, 4, 5, 7, 9]
```

```
Iterasi 5
[2, 3, 4, 5, 7, 9]
[2, 3, 4, 5, 7, 9]
[2, 3, 4, 5, 7, 9]
[2, 3, 4, 5, 7, 9]
[2, 3, 4, 5, 7, 9]
```

Angka setelah di-sorting 2 3 4 5 7 9

Berikut adalah hasil pengujian sorting data secara Descending dengan data random yang sama seperti sebelumnya dan iterasi tetap terjadi sebanyak 5 kali iterasi [19] [20].

Output - BubleSort (run) ✕

```
run:
Angka sebelum di-sorting [9, 4, 2, 5, 3, 7]
Iterasi 1
[9, 4, 2, 5, 3, 7]
[9, 4, 2, 5, 3, 7]
[9, 4, 5, 2, 3, 7]
[9, 4, 5, 3, 2, 7]
[9, 4, 5, 3, 7, 2]
```

```
Iterasi 2
[9, 4, 5, 3, 7, 2]
[9, 5, 4, 3, 7, 2]
[9, 5, 4, 3, 7, 2]
[9, 5, 4, 7, 3, 2]
[9, 5, 4, 7, 3, 2]
```

```
Iterasi 3
[9, 5, 4, 7, 3, 2]
[9, 5, 4, 7, 3, 2]
[9, 5, 7, 4, 3, 2]
[9, 5, 7, 4, 3, 2]
[9, 5, 7, 4, 3, 2]
```

```
Iterasi 4
[9, 5, 7, 4, 3, 2]
[9, 7, 5, 4, 3, 2]
[9, 7, 5, 4, 3, 2]
[9, 7, 5, 4, 3, 2]
[9, 7, 5, 4, 3, 2]
```

```
Iterasi 5
[9, 7, 5, 4, 3, 2]
[9, 7, 5, 4, 3, 2]
[9, 7, 5, 4, 3, 2]
[9, 7, 5, 4, 3, 2]
[9, 7, 5, 4, 3, 2]
```

Angka setelah di-sorting 9 7 5 4 3 2

Kesimpulan

Dari pembahasan ini dapat disimpulkan algoritma Bubble Sort adalah algoritma dengan metode paling sederhana untuk mengurutkan data dan merupakan algoritma yang mudah dipahami.

Bubble Sort ini merupakan metode pengurutan yang tidak efisien, khususnya ketika menangani data yang jumlahnya besar. Hal ini karena ketika mengurutkan data yang sangat besar akan sangat lambat prosesnya. Pada sorting data secara Ascending dan Descending perbedaan hanya terletak pada ukuran data yang diurutkan, apakah itu lebih besar atau lebih kecil dengan syntax programnya. Jika pada Ascending $\text{if}(A[j] > A[j+1])$, maka pada Descending kebalikan dari Ascending yaitu $\text{if}(A[j] < A[j+1])$.

Ucapan Terimakasih

Oleh karena itu, peneliti mengucapkan terima kasih kepada Bapak Andrian Syahputra M.kom, selaku Dosen Mata Kuliah Perancangan dan Analisis Algoritma yang telah menyediakan waktu, tenaga dan pikirannya, dan kepada rekan-rekan penulis yang telah banyak membantu dalam menyelesaikan jurnal ini.

Daftar Pustaka

- [1] A. H. Saptadi and D. W. Sari, "Analisis Algoritma Insertion Sort, Merge Sort Dan Implementasinya Dalam Bahasa Pemrograman C++," J. INFOTEL - Inform. Telekomun. Elektron., vol. 4, no. 2, p. 10, 2012, doi: 10.20895/infotel.v4i2.103.
- [2] H. Satria Tambunan, I. Gunawan, E. Irawan, and S. Tunas Bangsa, "Optimasi Algoritma Shell Sort Dalam Pengurutan Data Huruf Dan Angka," Junal Sist. Inf. Ilmu Komput. Prima, vol. 2, no. 1, pp. 23–27, 2018.
- [3] E. Sunandar and I. Indrianto, "Implementasi Algoritma Bubble Sort Terhadap 2 Buah Model Varian Pengurutan Data Menggunakan Bahasa Program Java," Petir, vol. 13, no. 2, pp. 255–265, 2020, doi: 10.33322/petir.v13i2.1008.
- [4] R. R. Nim and K. W. Asimptotik, "Analisis Algoritma Bubble Sort," no. 1, pp. 1–5, 2009.
- [5] Maysarah, "Sorting (Buble dan Quick Sort)," 2019. <https://maysarahmay.blogspot.com/2019/05/sorting-buble-dan-quick-sort.html> (accessed Jan. 12, 2022).
- [6] M. T. . Sutiono S.Kom., M.Kom., "Apa itu Algoritma Bubble Sort?," 2018. <https://dosenit.com/kuliah-it/rpl/algoritma-bubble-sort> (accessed Jan. 12, 2022).
- [7] M. K. Mira Suryani, S.Pd., "ARRAY/LARIK," 2018. https://lmsspada.kemdikbud.go.id/pluginfile.php/51109/mod_resource/content/1/6.Array.pdf (accessed Jan. 12, 2022).
- [8] William Hartanto, "Implementasi Algoritma Bubble Sort dengan Bahasa Pemrograman Python," binus.ac.id, 2019. <https://binus.ac.id/bandung/2019/12/implementasi-algoritma-bubble-sort-dengan-bahasa-pemrograman-python/> (accessed Jan. 11, 2022).
- [9] MUCHAMAD HENDRIX PRISTYAWAN, "ANALISIS ALGORITMA SEQUENTIAL SEARCH UNTUK MENCARI DATA MAHASISWA TEKNIK INFORMATIKA PADA SEKOLAH TINGGI TEKNOLOGI PELITA BANGSA," 2018. <https://ecampus.pelitabangsa.ac.id/pb/AmbilLampiran?ref=22916&jurusan=&jenis=Item&usingId=false&download=false&clazz=ais.database.model.file.LampiranLain>.
- [10] ATB, "Asymptotic Notation, Best Case, Average Case, dan Worst Case," Medium.com, 2020. <https://medium.com/@atb1161/asymptotic-notation-best-case-average-case-dan-worst-case-2457e579d9d> (accessed Jan. 12, 2022).
- [11] Husnadi, T. C., Marianti, T., & Ramadhan, T. (2022). Determination of shareholders' welfare with financing quality as a moderating variable. APTISI Transactions on Management (ATM), 6(2), 191-208.
- [12] Agustin, F., Oganda, F. P., Lutfiani, N., & Harahap, E. P. (2020). Manajemen Pembelajaran Daring Menggunakan Education Smart Courses. Technomedia Journal, 5(1 Agustus), 40-53.
- [13] Rahardja, U., Handayani, I., Lutfiani, N., & Oganda, F. P. (2020). An Interactive Content Media on Information System iLearning+. IJCCS (Indonesian Journal of Computing and Cybernetics Systems), 14(1), 57-68.
- [14] Oganda, F. P., Rahardja, U., Aini, Q., Hardini, M., & Bist, A. S. (2020). Blockchain: Visualization of the Bitcoin Formula. PalArch's Journal of Archaeology of Egypt/Egyptology, 17(6), 308-321.
- [15] Santoso, R. E., Oganda, F. P., Harahap, E. P., & Permadi, N. I. (2021). Pemanfaatan Penggunaan Hyperlocal Marketing bagi Startup Bidang Kuliner di Tangerang. ADI Bisnis Digital Interdisiplin Jurnal, 2(2), 60-65.
- [16] Prawiyogi, A. G., Anwar, A. S., Yusup, M., Lutfiani, N., & Ramadhan, T. (2021). Pengembangan Program Studi Bisnis digital bagi pengusaha dengan perangkat lunak lean. ADI Bisnis Digital Interdisiplin Jurnal, 2(2), 52-59.
- [17] Oganda, F. P., Hardini, M., & Ramadhan, T. (2021). Pengaruh Penggunaan kontrak cerdas pada Cyberpreneurship Sebagai Media Pemasaran dalam Dunia Bisnis. ADI Bisnis Digital Interdisiplin Jurnal, 2(1), 55-64.
- [18] Ramadhan, T., Aini, Q., Santoso, S., Badrianto, A., & Supriati, R. (2021). Analysis of the potential context of Blockchain on the usability of Gamification with Game-Based Learning. International Journal of Cyber and IT Service Management, 1(1), 84-100.
- [19] Saraswati, M., Lutfiani, N., & Ramadhan, T. (2021). Kolaborasi Integrasi Inkubator Bersama Perguruan Tinggi Sebagai Bentuk Pengabdian Terhadap Masyarakat Dalam Perkembangan Iptek. ADI Pengabdian Kepada Masyarakat, 1(2), 23-31.

-
- [20] Elmanda, V., Purba, A. E., Sanjaya, Y. P. A., & Julianingsih, D. (2022). Efektivitas Program Magang Siswa SMK di Kota Serang Dengan Menggunakan Metode CIPP di Era Adaptasi New Normal Pandemi Covid-19. ADI Bisnis Digital Interdisiplin Jurnal, 3(1), 5-15.