

# Achmad Arrosyidi - jsisfotek

*by Achmad Arrosyidi*

---

**Submission date:** 15-Dec-2022 06:51AM (UTC+0700)

**Submission ID:** 1981519636

**File name:** Achmad\_Arrosyidi\_-\_jsisfotek.pdf (1.19M)

**Word count:** 4672

**Character count:** 29423



## Perbandingan Algoritma Simple Sorting Antara Menggunakan Variabel Temporary dan Tanpa Variabel Temporary

Achmad Arrosyidi<sup>1</sup>, Didiet Anindita Armandy<sup>2</sup>

<sup>1</sup>Universitas Dinamika

<sup>2</sup>Universitas Dinamika

[achmad@dinamika.ac.id](mailto:achmad@dinamika.ac.id)

### Abstract

Temporary variables used in swapping technique **31** sorting algorithms. There are alternative swapping techniques uses pairs of + and -, x and / operators in the sorting process. This study aims to determine the duration of the sorting process uses the + and - operator pairs, the x and / operator, and the uses of temporary variables. By knowing the duration, it can determine the performance of each swapping technique. Important of this research it can contribute for advanced research or applied research.

Data generated input algorithms in descending and ascending sequence. The input was sorted by 18 types of Bubble, Selection, and Insertion sort algorithm. Output sequence contrary from the data input. Duration sorting process obtained from the end processed time subtract by the start processed time. Sorting duration recapitulated to get the average duration of the sorting process so can be compared between swapping techniques. The test method has run 18 types of Simple sorting algorithms from 10, 50, and 100 data variants, frequency of each variant 10 times tests. The total data used 28,800 of integer types.

Result of the average duration test that swapping uses x and / operators 306,236 milliseconds uses + and - operators 294,998 milliseconds, and swapping technique with temporary variables 294,557 milliseconds. The best performance showed by swapping technique uses temporary variables instead of uses pair of + and - operators, also used pair of x and / operators

Keywords: algorithm, sorting, variable, temporary, comparison

### Abstrak

Teknik swapping dalam proses *sorting* menggunakan variabel *temporary*. Terdapat alternatif teknik *swapping* yaitu menggunakan pasangan operator + dan -, serta pasangan operator x dan / dalam proses *sorting*. Penelitian ini bertujuan untuk mengetahui durasi proses *sorting* dan penggunaan pasangan operator + dan -, pasangan operator x dan /, serta perbandingan variabel *temporary*. Dengan mengetahui durasi tersebut maka dapat mengetahui performa setiap teknik *swapping*, sehingga penelitian ini penting untuk dilakukan karena dapat berkontribusi dalam penelitian tingkat lanjut ataupun penelitian terapan.

**37** disusun oleh algoritma *input* secara *descending*, dan *ascending*. *Input* kemudian diproses oleh 18 jenis komposisi algoritma Bubble sort, Selection sort, dan Insertion sort. *Output* yang dihasilkan kebalikan dari *input*-nya. Durasi proses *sorting* didapat dan waktu akhir dikurangi waktu mulai. Durasi *sorting* direkapitulasi sehingga diperoleh rata-rata durasi proses *sorting* untuk dibandingkan antar teknik *swapping*. Metode pengujian yakni menjalankan 18 jenis algoritma Simple sorting dengan varian data 10, 50, dan 100, dengan frekuensi setiap pengujian sebanyak 10 kali. Total data yang digunakan sebanyak 28.800 bertipe bilangan bulat.

Hasil rata-rata durasi *sorting* menggunakan teknik *swapping* dengan operator x dan / 306.236 mil detik, teknik *swapping* dengan operator + dan - 294.998 mil detik, dan teknik *swapping* dengan variabel *temporary* 294.557 mil detik. Performa terbaik proses *sorting* ditunjukkan oleh algoritma Simple sorting yaitu Bubble sort, Selection sort, dan Insertion sort menggunakan teknik *swapping* dengan variabel *temporary*.

Kata kunci: algoritma, *simple sorting*, *temporary*, perbandingan.

JSISFOTEK is licensed under a Creative Commons 4.0 International License.



### 1. Pendahuluan

Kecepatan adalah salah satu parameter performansi algoritma *sorting* (Hiwe Hiwe Aung, 2019), (Jmasa et al., 2019), (Bustami et al., 2019), (M. Rablu et al., 2022), (K & J, 2018), (Ekowati et al., 2022). Variabel *temporary* digunakan sebagai penyimpanan sementara

saat proses *sorting*. Terdapat alternatif *sorting*, yaitu *swapping* menggunakan pasangan operator + dan -, serta x dan /. *Sorting* menggunakan variabel *temporary* akan menambah beban pada memori komputer (Raghavanshi, 2018), (Anggrani et al., 2020), (Zhao et al., 2016), (Abdel-Hafeez & Gordon-Ross, 2017)

**25**

Diterima: xx-xx-20xx | Revisi: xx-xx-20xx | Diterbitkan: xx-xx-20xx | doi: 10.37034/jsisfotek.v4i2.1

dibandingkan *sorting* tanpa menggunakan variabel *temporary*.

### 1.1. Algoritma Sorting

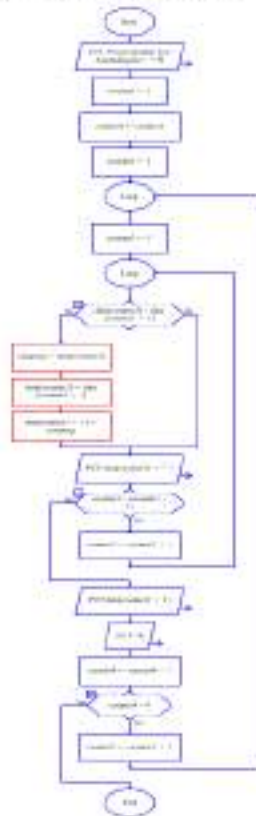
#### a. Tipe Algoritma Sorting

Sorting internal: semua elemen dapat dibaca ke dalam memori pada saat yang sama dan penyortiran dilakukan dalam memori. Didalamnya terdapat Simple, Advanced, dan Radix Sort. Sorting eksternal: dalam hal ini, dataset begitu besar sehingga tidak muat seluruh dataset ke dalam memori (Rajasepal & Thilakavalli, 2016) sehingga pengurutan dilakukan dalam potongan.

#### b. Algoritma Simple Sorting

1) Bubble Sort, Pengurutan dengan cara pertukaran data dengan data disebelahnya secara terus menerus sampai tidak ada lagi perubahan. Secara umum algoritma Bubble sort menggunakan variabel *temporary*. Bubble sort adalah algoritma pengurutan yang sederhana (Sari et al., 2022).

Gambar 1 adalah *flowchart* dari algoritma Bubble Sort yang menggunakan variabel *temporary*.

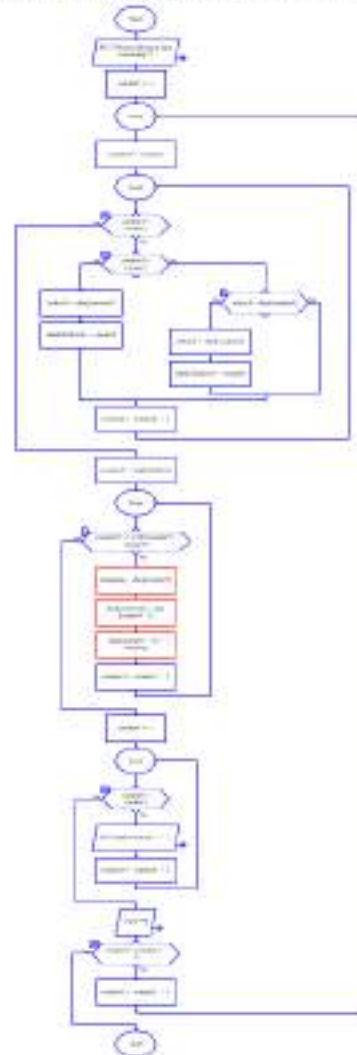


Gambar 1. Algoritma Bubble Sort

40 Selection Sort, pengurutan menggunakan algoritma selection sort adalah pengurutan dengan cara memilih atau mencari data terkecil atau terbesar dari setiap data (Pasetto & Akhriev, 2011). Saat data ditemukan maka data akan diletakkan di posisi pertama kemudian dilanjutkan dengan urutan kedua dan seterusnya sampai dengan data terakhir.

Secara umum algoritma Selection sort menggunakan variabel *temporary* sebagai variabel untuk menyimpan data sementara saat pertukaran dalam pengurutan data.

Gambar 2 adalah *flowchart* dari algoritma Selection Sort yang menggunakan variabel *temporary*.

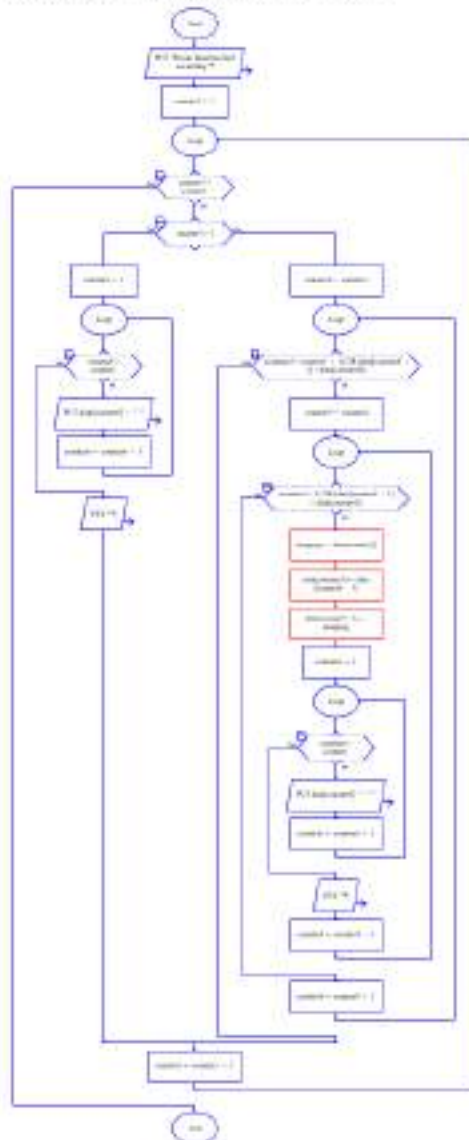


Gambar 2. Algoritma Selektiva Sort

3) Insertion Sort, pengurutan data dengan cara menyisipkan data baru diantara data yang sudah ada

sebelumnya dengan menggunakan kriteria tertentu (*ascending / descending*). Penyisipan data dilakukan secara terus menerus sampai tidak ada data yang tersisa lagi untuk disisipkan. Secara umum algoritma Insertion sort menggunakan variabel *temporary* sebagai variabel untuk menyimpan data sementara saat pertukaran dalam pengurutan data (Kavdikar, 2021).

Gambar 3 adalah *flowchart* dari algoritma Insertion Sort yang menggunakan variabel *temporary*.



Gambar 3. Algoritma Insertion Sort

## 1.2. Teknik Pertukaran Data

### a. Menggunakan Variabel Temporary

Menggunakan variabel *temporary* untuk menyimpan nilai *x* untuk sementara. Kemudian memberi nilai *y* di *x* dan kemudian *temporary* di *y*. Dengan cara ini, nilai ditukar. Algoritma 1 adalah contoh penerapan pertukaran data menggunakan variabel *temp*.

#### Algoritma 1. Proses Pertukaran Data Menggunakan Variabel Temporary

```

x = 5
y = 10
temp = x
x = y
y = temp
Print('The value of x after swapping: {}',
format(x))
Print('The value of y after swapping: {}',
format(y))
    
```

### b. Tanpa Menggunakan Variabel Temporary

Dengan menggunakan operasi aritmatika untuk melakukan pertukaran. Pertukaran data tanpa variabel *temporary* dapat menggunakan Pasangan Operator Penjumlahan-Pengurangan dapat menggantikan penggunaan variabel *temporary* dalam proses pertukaran data. Algoritma 2 adalah contoh penerapan pertukaran data menggunakan operator penjumlahan dan pengurangan sebagai pengganti variabel *temporary*.

#### Algoritma 2. Proses Pertukaran Data Menggunakan Pasangan Operator Penjumlahan-Pengurangan

```

x = x + y
y = x - y
x = x - y
    
```

### 1) Pasangan Operator Perkalian-Pembagian

Penggunaan operator perkalian dan pembagian dapat menggantikan penggunaan variabel *temporary* dalam proses pertukaran data. Algoritma 3 adalah contoh penerapan pertukaran data menggunakan operator perkalian dan pembagian sebagai pengganti variabel *temporary*.

#### Algoritma 3. Proses Pertukaran Data Menggunakan Pasangan Operator Perkalian-Pembagian

```

x = x * y
y = x / y
x = x / y
    
```

## 1.3. Software Penelitian

*Software* yang digunakan untuk membuat algoritma dalam penelitian ini menggunakan *software* simulasi Raptor. RAPTOR adalah bahasa pemrograman visual berdasarkan diagram alir yang dapat dieksekusi, serta mempunyai fitur sintaks pemrograman dengan menggunakan enam simbol grafis. Fitur-fitur ini memungkinkan peneliti untuk lebih fokus pada

penalaran algoritmik, sehingga lebih presisi pada bagian algoritma yang diteliti.

Alasan diadakan penelitian ini.

Penelitian ini menguji kecepatan *simple sorting* Bubble sort, Insertion sort, dan Selection sort, yang masing-masing menggunakan variabel temporary (Sunandar, 2019), (Setiawan, 2017), serta tanpa variabel temporary.

Tujuan dari penelitian ini adalah untuk menghasilkan informasi (Sonita & Nurtaneo, 2016) perbandingan antar algoritma *sorting* menggunakan Bubble, Insertion, dan Selection sort yang proses didalamnya terdapat proses pertukaran data menggunakan variabel temporary, dan tanpa menggunakan variabel temporary. Dengan mengetahui durasinya maka bermanfaat untuk memberikan kontribusi terhadap terapan (Sunandar & Indrianto, 2020), (Betschart et al., 2022).

Manfaat dari penelitian ini adalah membantu berkontribusi terhadap penelitian tingkat lanjut maupun penelitian terapan yang menggunakan proses *sorting* agar performanya dapat lebih optimal. Menambah kepustakaan keilmuan di bidang komputer khususnya dalam bidang algoritma Simple sorting. Sehingga perlu untuk mengetahui bagaimana menguji performa kecepatan algoritma Simple Sorting antara menggunakan variabel temporary dan tanpa variabel temporary?

2. Metodologi Penelitian

a. Persiapan

Telah membuat algoritma untuk penyediaan data secara *ascending* dan *descending* yang valid yang siap diuji. Dari setiap algoritma Bubble, Selection dan Insertion sort akan menghasilkan dua jenis urutan output yaitu *ascending*, dan *descending*. Setiap proses pengurutan memanfaatkan teknik menggunakan variabel temporary, dan tanpa variabel temporary. Penggunaan tanpa variabel temporary menggunakan pasangan operator + dan -, serta pasangan operator perkalian x dan /. Kombinasi diatas akan menghasilkan 18 (delapan belas) algoritma yaitu:

- 1) Bubble sort ascending dari input descending, menggunakan variabel temporary.
- 2) Bubble sort descending dari input ascending, menggunakan variabel temporary.
- 3) Bubble sort ascending dari input descending, menggunakan pasangan operator + dan -.
- 4) Bubble sort descending dari input ascending, menggunakan pasangan operator + dan -.
- 5) Bubble sort ascending dari input descending, menggunakan pasangan operator x dan /.

- 6) Bubble sort descending dari input ascending, menggunakan pasangan operator x dan /.
- 7) Selection sort ascending dari input descending, menggunakan variabel temporary.
- 8) Selection sort descending dari input ascending, menggunakan variabel temporary.
- 9) Selection sort ascending dari input descending, menggunakan pasangan operator + dan -.
- 10) Selection sort descending dari input ascending, menggunakan pasangan operator + dan -.
- 11) Selection sort ascending dari input descending, menggunakan pasangan operator x dan /.
- 12) Selection sort descending dari input ascending, menggunakan pasangan operator x dan /.
- 13) Insertion sort ascending dari input descending, menggunakan variabel temporary.
- 14) Insertion sort descending dari input ascending, menggunakan variabel temporary.
- 15) Insertion sort ascending dari - input descending, menggunakan pasangan operator + dan -.
- 16) Insertion sort descending dari input ascending, menggunakan pasangan operator + dan -.
- 17) Insertion sort ascending dari input descending, menggunakan pasangan operator x dan /.
- 18) Insertion sort descending dari input ascending, menggunakan pasangan operator x dan /.

Secara rinci setiap percobaan dengan menggunakan algoritma 18 *sorting* yang berbeda. Masing-masing algoritma memproses input berupa data *descending* menghasilkan output yang urut secara *ascending*, dan juga sebaliknya yaitu input data *ascending* menghasilkan output yang urut secara *descending*. Secara rinci algoritma *sorting* yang digunakan pengujian dapat disajikan pada Tabel 1 berikut ini.

Tabel 1. Rincian Algoritma Sorting

No	SUSUNAN INPUT	JUMLAH DATA				PERUBAHAN	WAKTU TAK	PROSES		SUSUNAN OUTPUT
		1	2	3	4			ALGORITMA	TEKNIK MANIPULASI	
1	Descending	20	20	100	100	1	1.000	Temporari	Operator + dan -	Ascending
2	Descending	20	20	100	100	1	1.000	Temporari	Operator x dan /	Descending
3	Descending	20	20	100	100	1	1.000	Temporari	Operator + dan -	Ascending
4	Descending	20	20	100	100	1	1.000	Temporari	Operator x dan /	Descending
5	Descending	20	20	100	100	1	1.000	Temporari	Operator + dan -	Ascending
6	Descending	20	20	100	100	1	1.000	Temporari	Operator x dan /	Descending
7	Descending	20	20	100	100	1	1.000	Temporari	Operator + dan -	Ascending
8	Descending	20	20	100	100	1	1.000	Temporari	Operator x dan /	Descending
9	Descending	20	20	100	100	1	1.000	Temporari	Operator + dan -	Ascending
10	Descending	20	20	100	100	1	1.000	Temporari	Operator x dan /	Descending
11	Descending	20	20	100	100	1	1.000	Temporari	Operator + dan -	Ascending
12	Descending	20	20	100	100	1	1.000	Temporari	Operator x dan /	Descending
13	Descending	20	20	100	100	1	1.000	Temporari	Operator + dan -	Ascending
14	Descending	20	20	100	100	1	1.000	Temporari	Operator x dan /	Descending
15	Descending	20	20	100	100	1	1.000	Temporari	Operator + dan -	Ascending
16	Descending	20	20	100	100	1	1.000	Temporari	Operator x dan /	Descending
17	Descending	20	20	100	100	1	1.000	Temporari	Operator + dan -	Ascending
18	Descending	20	20	100	100	1	1.000	Temporari	Operator x dan /	Descending

b. Teknik Karakterisasi

Telah membuat tiga varian data yaitu 10, 50, dan 100 (Jones & Ackley, 2014). Data diuji menggunakan frekuensi sebanyak 10 kali. Sehingga terdapat total data sebanyak 28.800 data yang telah diuji.

Tabel 2. Rincian Jumlah Data

No	JENISAN DATA	Jumlah Data			Frekuensi	DURASI TOTAL	PROSES		DURASI OUTPUT
		1	2	3			ALGO BUBBLE	TEKNIK SAMPEL	
1	Ascending	10	50	100	100	1.000	Ascending	Ascending	
2	Ascending	10	50	100	100		Bubble Sort	Ascending	
3	Ascending	10	50	100	100			Descending	
4	Ascending	10	50	100	100			Descending	
5	Ascending	10	50	100	100			Descending	
6	Ascending	10	50	100	100			Descending	
7	Ascending	10	50	100	100			Descending	
8	Ascending	10	50	100	100			Descending	
9	Ascending	10	50	100	100			Descending	
10	Ascending	10	50	100	100			Descending	
11	Ascending	10	50	100	100	Selection Sort		Ascending	
12	Ascending	10	50	100	100		Descending		
13	Ascending	10	50	100	100		Descending		
14	Ascending	10	50	100	100		Descending		
15	Ascending	10	50	100	100		Descending		
16	Ascending	10	50	100	100		Descending		
17	Ascending	10	50	100	100		Descending		
18	Ascending	10	50	100	100		Descending		
19	Ascending	10	50	100	100		Descending		
20	Ascending	10	50	100	100		Descending		
21	Ascending	10	50	100	100	Insertion Sort	Ascending		
22	Ascending	10	50	100	100		Descending		
23	Ascending	10	50	100	100		Descending		
24	Ascending	10	50	100	100		Descending		
25	Ascending	10	50	100	100		Descending		
26	Ascending	10	50	100	100		Descending		
27	Ascending	10	50	100	100		Descending		
28	Ascending	10	50	100	100		Descending		
29	Ascending	10	50	100	100		Descending		
30	Ascending	10	50	100	100		Descending		

c. Proses menguji setiap algoritma

Telah dilakukan pengurutan secara terbalik antara *input* dibandingkan *output*-nya. Setiap pengujian algoritma mampu melakukan rekapitulasi durasi pengurutan

d. Pengujian Algoritma

Telah dijalankan 18 (delapan belas) jenis algoritma *sorting* yang dibuat sebelumnya. Frekuensi yang digunakan adalah sebanyak 10 kali pengujian. Pengujian akan dilakukan dengan menggunakan *sampling* jenuh, yaitu sebuah teknik penentuan sampel, karena semua anggota populasi digunakan sebagai sampel. Sampel jenuh disebut juga dengan istilah *sensus*, karena semua anggota populasi dijadikan sampel. Algoritma akan diarahkan proses menghitung durasi proses setelah *sorting*. Proses perhitungan ini dilakukan diluar *sorting*.

Selain akan menghasilkan durasi yang didapat dari selisih waktu akhir dikurangi dengan waktu awal, juga melakukan proses perhitungan rata-rata dari setiap dari setiap data yang diuji, serta menghasilkan waktu rata-rata keseluruhan proses *sorting*. Setiap 18 algoritma, telah mengurutkan data sebanyak 3 varian (10, 50, dan 100), sehingga terdapat 54 hasil pengujian. Hasil setiap pengujian yang telah dijalankan, tersaji pada Tabel 3 s/d 56 yang terurut mulai dari Bubble, Selection, dan Insertion sort.

Tabel 3. Hasil Pengujian Bubble Sort Ascending Dari Input Descending, Menggunakan Variabel Temporary Dengan Jumlah Data 10

JENISAN DATA	Jenis Algoritma	DURASI			Frekuensi	DURASI TOTAL	PROSES		DURASI OUTPUT
		1	2	3			ALGO BUBBLE	TEKNIK SAMPEL	
Ascending	Bubble Sort	10	50	100	100	1.000	Ascending	Ascending	
Ascending	Bubble Sort	10	50	100	100		Bubble Sort	Ascending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100	Descending			

Tabel 4. Hasil Pengujian Bubble Sort Ascending Dari Input Descending, Menggunakan Variabel Temporary Dengan Jumlah Data 50

JENISAN DATA	Jenis Algoritma	DURASI			Frekuensi	DURASI TOTAL	PROSES		DURASI OUTPUT
		1	2	3			ALGO BUBBLE	TEKNIK SAMPEL	
Ascending	Bubble Sort	10	50	100	100	1.000	Ascending	Ascending	
Ascending	Bubble Sort	10	50	100	100		Bubble Sort	Ascending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100	Descending			

Tabel 5. Hasil Pengujian Bubble Sort Ascending Dari Input Descending, Menggunakan Variabel Temporary Dengan Jumlah Data 100

JENISAN DATA	Jenis Algoritma	DURASI			Frekuensi	DURASI TOTAL	PROSES		DURASI OUTPUT
		1	2	3			ALGO BUBBLE	TEKNIK SAMPEL	
Ascending	Bubble Sort	10	50	100	100	1.000	Ascending	Ascending	
Ascending	Bubble Sort	10	50	100	100		Bubble Sort	Ascending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100	Descending			

Tabel 6. Hasil Pengujian Bubble Sort Descending Dari Input Ascending, Menggunakan Variabel Temporary Dengan Jumlah Data 10

JENISAN DATA	Jenis Algoritma	DURASI			Frekuensi	DURASI TOTAL	PROSES		DURASI OUTPUT
		1	2	3			ALGO BUBBLE	TEKNIK SAMPEL	
Ascending	Bubble Sort	10	50	100	100	1.000	Ascending	Ascending	
Ascending	Bubble Sort	10	50	100	100		Bubble Sort	Ascending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100	Descending			

Tabel 7. Hasil Pengujian Bubble Sort Descending Dari Input Ascending, Menggunakan Variabel Temporary Dengan Jumlah Data 50

JENISAN DATA	Jenis Algoritma	DURASI			Frekuensi	DURASI TOTAL	PROSES		DURASI OUTPUT
		1	2	3			ALGO BUBBLE	TEKNIK SAMPEL	
Ascending	Bubble Sort	10	50	100	100	1.000	Ascending	Ascending	
Ascending	Bubble Sort	10	50	100	100		Bubble Sort	Ascending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100			Descending	
Ascending	Bubble Sort	10	50	100	100	Descending			

Tabel 8. Hasil Pengujian Bubble Sort Descending Dari Input Ascending, Menggunakan Variabel Temporary Dengan Jumlah Data 100

Tabel 9. Bubble Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator + dan - Dengan Jumlah Data 10

Tabel 10. Bubble Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator + dan - Dengan Jumlah Data 50

Tabel 11. Bubble Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator + dan - Dengan Jumlah Data 100

Tabel 12. Bubble Sort Descending Dari Input Ascending, Menggunakan Pasangan Operator + dan - Dengan Jumlah Data 10

Tabel 13. Bubble Sort Descending Dari Input Ascending, Menggunakan Pasangan Operator + dan - Dengan Jumlah Data 50

Tabel 14. Bubble Sort Descending Dari Input Ascending, Menggunakan Pasangan Operator + dan - Dengan Jumlah Data 100

Tabel 15. Bubble Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator x dan / Dengan Jumlah Data 10

Tabel 16. Bubble Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator x dan / Dengan Jumlah Data 50

Tabel 17. Bubble Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator x dan / Dengan Jumlah Data 100





Tabel 28. Selection Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator + dan - Dengan Jumlah Data 50

Urutan	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	
1	100	99	98	97	96	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50

Tabel 29. Selection Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator + dan - Dengan Jumlah Data 100

Urutan	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	100	99	98	97	96	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

Tabel 30. Selection Sort Descending Dari Input Ascending, Menggunakan Pasangan Operator + dan - Dengan Jumlah Data 10

Urutan	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10

Tabel 31. Selection Sort Descending Dari Input Ascending, Menggunakan Pasangan Operator + dan - Dengan Jumlah Data 50

Urutan	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Tabel 32. Selection Sort Descending Dari Input Ascending, Menggunakan Pasangan Operator + dan - Dengan Jumlah Data 100

Urutan	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

Tabel 33. Selection Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator x dan / Dengan Jumlah Data 10

Urutan	1	2	3	4	5	6	7	8	9	10
1	100	99	98	97	96	95	94	93	92	91

Tabel 34. Selection Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator x dan / Dengan Jumlah Data 50

Urutan	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	
1	100	99	98	97	96	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50

Tabel 35. Selection Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator x dan / Dengan Jumlah Data 100

Urutan	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	100	99	98	97	96	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10									

Tabel 36. Selection Sort Descending Dari Input Ascending, Menggunakan Pasangan Operator x dan / Dengan Jumlah Data 10

Urutan	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10

Tabel 37. Selection Sort Descending Dari Input Ascending, Menggunakan Pasangan Operator x dan / Dengan Jumlah Data 50

Urutan	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Tabel 38. Selection Sort Descending Dari Input Ascending, Menggunakan Pasangan Operator  $\times$  dan  $\div$  Dengan Jumlah Data 100

Iteration	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

Tabel 43. Insertion Sort Descending Dari Input Ascending, Menggunakan Variabel Temporary Dengan Jumlah Data 50

Iteration	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Tabel 39. Insertion Sort Ascending Dari Input Descending, Menggunakan Variabel Temporary Dengan Jumlah Data 10

Iteration	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10

Tabel 44. Insertion Sort Descending Dari Input Ascending, Menggunakan Variabel Temporary Dengan Jumlah Data 100

Iteration	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

Tabel 40. Insertion Sort Ascending Dari Input Descending, Menggunakan Variabel Temporary Dengan Jumlah Data 50

Iteration	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Tabel 45. Insertion Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator  $+$  dan  $-$  Dengan Jumlah Data 10

Iteration	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10

Tabel 41. Insertion Sort Ascending Dari Input Descending, Menggunakan Variabel Temporary Dengan Jumlah Data 100

Iteration	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

Tabel 46. Insertion Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator  $+$  dan  $-$  Dengan Jumlah Data 50

Iteration	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Tabel 42. Insertion Sort Descending Dari Input Ascending, Menggunakan Variabel Temporary Dengan Jumlah Data 10

Iteration	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10

Tabel 47. Insertion Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator  $+$  dan  $-$  Dengan Jumlah Data 100

Iteration	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

Tabel 48. Insertion Sort Descending Dari Input Ascending, Menggunakan Pasangan Operator + dan - Dengan Jumlah Data 10

Table with 12 columns: Iterasi, Data Awal, Data Akhir, Waktu, and other performance metrics for Insertion Sort with + and - operators on 10 data points.

Tabel 49. Insertion Sort Descending Dari Input Ascending, Menggunakan Pasangan Operator + dan - Dengan Jumlah Data 50

Table with 12 columns: Iterasi, Data Awal, Data Akhir, Waktu, and other performance metrics for Insertion Sort with + and - operators on 50 data points.

Tabel 50. Insertion Sort Descending Dari Input Ascending, Menggunakan Pasangan Operator + dan - Dengan Jumlah Data 100

Table with 12 columns: Iterasi, Data Awal, Data Akhir, Waktu, and other performance metrics for Insertion Sort with + and - operators on 100 data points.

Tabel 51. Insertion Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator x dan / Dengan Jumlah Data 10

Table with 12 columns: Iterasi, Data Awal, Data Akhir, Waktu, and other performance metrics for Insertion Sort with x and / operators on 10 data points.

Tabel 52. Insertion Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator x dan / Dengan Jumlah Data 50

Table with 12 columns: Iterasi, Data Awal, Data Akhir, Waktu, and other performance metrics for Insertion Sort with x and / operators on 50 data points.

Tabel 53. Insertion Sort Ascending Dari Input Descending, Menggunakan Pasangan Operator x dan / Dengan Jumlah Data 100

Table with 12 columns: Iterasi, Data Awal, Data Akhir, Waktu, and other performance metrics for Insertion Sort with x and / operators on 100 data points.

Tabel 54. Insertion Sort Descending Dari Input Ascending, Menggunakan Pasangan Operator x dan / Dengan Jumlah Data 10

Table with 12 columns: Iterasi, Data Awal, Data Akhir, Waktu, and other performance metrics for Insertion Sort with x and / operators on 10 data points.

Tabel 55. Insertion Sort Descending Dari Input Ascending, Menggunakan Pasangan Operator x dan / Dengan Jumlah Data 50

Table with 12 columns: Iterasi, Data Awal, Data Akhir, Waktu, and other performance metrics for Insertion Sort with x and / operators on 50 data points.

Tabel 56. Insertion Sort Descending Dan Input Ascending, Menggunakan Pasangan Operator x dan / Dengan Jumlah Data 100

Table with 12 columns: Iterasi, Data Awal, Data Akhir, Waktu, and other performance metrics for Insertion Sort with x and / operators on 100 data points.

c. Penyusunan Hasil Pengujian

Tabel 57 menunjukkan, hasil akhir rekapitulasi rata-rata pengujian algoritma Bubble, Selection, dan Insertion sort dan penggabungan input, output data secara ascending dan descending, durasi menggunakan variabel temporary sebesar 1.669 milidetik, pasangan operator + dan - sebesar 1.626.6667 milidetik, serta pasangan operator x dan / sebesar 1.650.333333 milidetik dengan jumlah data 10. Pengujian dilakukan sebanyak 10 kali.

Tabel 57. Rekapitulasi Perbandingan Algoritma Dengan Jumlah Data 10

ALGORITMA	JENIS DATA	WAKTU (MILIDETIK)																		
		ASCEENDING					DESCENDING													
		INPUT	OUTPUT	TEMPORARY	OPERATOR + DAN -	OPERATOR X DAN /	INPUT	OUTPUT	TEMPORARY	OPERATOR + DAN -	OPERATOR X DAN /									
10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

Tabel 58 menunjukkan, hasil akhir rekapitulasi rata-rata pengujian algoritma Bubble, Selection, dan Insertion sort dan penggabungan *input*, *output* data secara *ascending* dan *descending*, durasi menggunakan variabel *temporary* sebesar 241.995 milidetik, pasangan operator + dan - sebesar 243.456 milidetik, serta pasangan operator x dan / sebesar 101.869,3333 milidetik dengan jumlah data 50. Pengujian dilakukan sebanyak 10 kali.

Tabel 58. Rekapitulasi Perbandingan Algoritma Dengan Jumlah Data 50

ALGORITMA	JENIS DATA	WAKTU (MILIDETIK)																		
		ASCEENDING					DESCENDING													
		INPUT	OUTPUT	TEMPORARY	OPERATOR + DAN -	OPERATOR X DAN /	INPUT	OUTPUT	TEMPORARY	OPERATOR + DAN -	OPERATOR X DAN /									
50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50

Tabel 59 menunjukkan, hasil akhir rekapitulasi rata-rata pengujian algoritma Bubble, Selection, dan Insertion sort dan penggabungan *input*, *output* data secara *ascending* dan *descending*, durasi menggunakan variabel *temporary* sebesar 2.060.704 milidetik, pasangan operator + dan - sebesar 2.158.849 milidetik, serta pasangan operator x dan / sebesar 780.158,3333 milidetik dengan jumlah data 100. Pengujian dilakukan sebanyak 10 kali.

Tabel 59. Rekapitulasi Perbandingan Algoritma Dengan Jumlah Data 100

ALGORITMA	JENIS DATA	WAKTU (MILIDETIK)																		
		ASCEENDING					DESCENDING													
		INPUT	OUTPUT	TEMPORARY	OPERATOR + DAN -	OPERATOR X DAN /	INPUT	OUTPUT	TEMPORARY	OPERATOR + DAN -	OPERATOR X DAN /									
100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Tabel 59 menunjukkan, hasil akhir rekapitulasi rata-rata penggabungan *input*, *output* data secara *ascending* dan *descending* untuk pengujian algoritma Bubble, Selection, dan Insertion sort.

Tabel 60. Rekapitulasi Rata-rata Perbandingan Algoritma Variasi Jumlah Data 10, 50, dan 100

VARIASI JUMLAH DATA	INPUT	OUTPUT	RATA-RATA (MILIDETIK)			
			TEMPORARY	OPERATOR + DAN -	OPERATOR X DAN /	OPERATOR + DAN -
10	10	10	241.995	243.456	101.869,3333	101.869,3333
50	50	50	2.060.704	2.158.849	780.158,3333	780.158,3333
100	100	100	20.607.040	21.588.490	7.801.583,3333	7.801.583,3333

Dari rekapitulasi rata-rata Tabel 60 diatas didapatkan, durasi menggunakan variabel *temporary* menggunakan varian data 10 yakni 1.643.666667 milidetik, menggunakan varian data 50 adalah 101.869,3333 milidetik, menggunakan varian data 100 selama 780.158,3333 milidetik. Durasi menggunakan operator + dan - menggunakan varian data 10 adalah 1.627,333 milidetik, menggunakan varian data 50 yaitu 101.800,3 milidetik, menggunakan varian data 100 adalah 781.556 milidetik. Durasi menggunakan operator x dan / menggunakan varian data 10 yakni 1.651,5 milidetik, menggunakan varian data 50 adalah 102.328,83 milidetik, menggunakan varian data 100 adalah 814.727,17 milidetik. Sehingga dapat disusun rekapitulasi pada Tabel 61.

Tabel 61. Rekapitulasi Perbandingan Algoritma

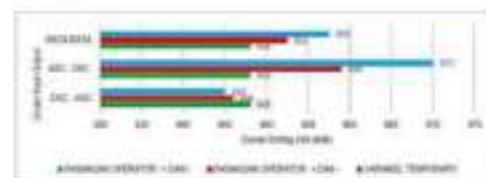
VARIASI JUMLAH DATA	TEMPORARY	OPERATOR + DAN -	OPERATOR X DAN /
10	1.643.666667	1.627.333	1.651.5
50	101.869,3333	101.800,3	102.328,83
100	780.158,3333	781.556	814.727,17

### 3. Hasil dan Pembahasan

#### Analisis Hasil Pengujian

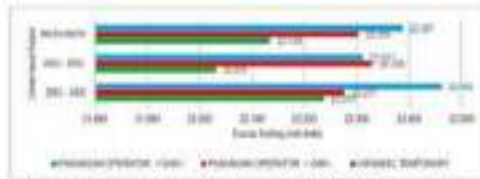
##### 3.1. Analisis Perbandingan Bubble Sort

1) Analisis Perbandingan Bubble Sort dengan Variasi Data 10. Dari Tabel 60 dapat disajikan dalam bentuk grafik pada Gambar 4.



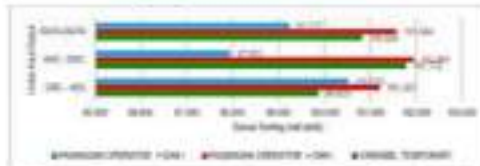
Gambar 4. Perbandingan Algoritma Bubble Sort Dengan Variasi Data 10

Terdapat hasil konsisten pada *swapping* menggunakan variabel *temporary*, namun tidak pada *swapping* menggunakan operator + dan -, begitu juga dengan *swapping* menggunakan operator x dan /. Sedangkan rata-rata didapatkan durasi tercepat adalah *swapping* menggunakan variabel *temporary*. 2) Analisis Perbandingan Bubble Sort dengan Variasi Data 50. Dari Tabel 60 dapat disajikan dalam bentuk grafik pada Gambar 5.



Gambar 5. Perbandingan Algoritma Bubble Sort Dengan Variasi Data 50

Terdapat hasil tidak konsisten pada semua teknik *swapping*. Sedangkan rata-rata didapatkan durasi tercepat adalah *swapping* menggunakan variabel *temporary*. 3) Analisis **5** perbandingan Bubble Sort dengan Variasi Data 100. Dari Tabel 60 dapat disajikan dalam bentuk grafik pada Gambar 6;



Gambar 6. Perbandingan Algoritma Bubble Sort Dengan Variasi Data 100

Terdapat hasil tidak konsisten pada semua teknik *swapping*. Sedangkan rata-rata didapatkan durasi tercepat adalah *swapping* menggunakan operator  $\times$  dan  $!$ .

### 3.2. Analisis Perbandingan Selection Sort

1) Analisis Perbandingan **4** Selection Sort dengan Variasi Data 10. Dari Tabel 60 dapat disajikan dalam bentuk grafik pada Gambar 7;



Gambar 7. Perbandingan Algoritma Selection Sort Dengan Variasi Data 100

Terdapat hasil tidak konsisten pada semua teknik *swapping*. Sedangkan rata-rata didapatkan durasi tercepat adalah *swapping* menggunakan variabel *temporary*. 2) Analisis Perbandingan **4** Selection Sort dengan Variasi Data 50. Dari Tabel 60 dapat disajikan dalam bentuk grafik pada Gambar 8;



Gambar 8. Perbandingan Algoritma Selection Sort Dengan Variasi Data 50

Terdapat hasil tidak konsisten pada semua teknik *swapping*. Sedangkan rata-rata didapatkan durasi tercepat adalah *swapping* menggunakan variabel *temporary*. 3) Analisis **5** perbandingan Selection Sort dengan Variasi Data 100. Dari Tabel 60 dapat disajikan dalam bentuk grafik pada Gambar 9;

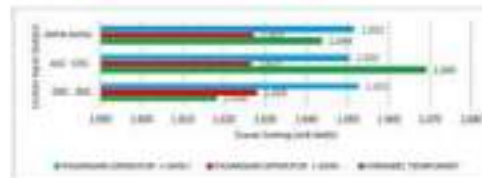


Gambar 9. Perbandingan Algoritma Selection Sort Dengan Variasi Data 100

Terdapat hasil konsisten pada semua teknik *swapping*. Konsistensi tersebut pada durasi *sorting descending* ke *ascending* lebih cepat daripada durasi *sorting ascending* ke *descending*. Sedangkan rata-rata didapatkan durasi tercepat adalah *swapping* menggunakan variabel *temporary*.

### 3.3. Analisis Perbandingan Insertion Sort

1) Analisis **5** Perbandingan Insertion Sort Dengan Variasi Data 10. Dari Tabel 60 dapat disajikan dalam bentuk grafik pada Gambar 10;



Gambar 10. Perbandingan Algoritma Insertion Sort Dengan Variasi Data 100

Terdapat hasil tidak konsisten pada semua teknik *swapping*. Sedangkan rata-rata didapatkan durasi tercepat adalah *swapping* menggunakan operator  $+$  dan  $-$ . 2) Analisis **5** perbandingan Insertion Sort Dengan Variasi Data 50. Dari Tabel 60 dapat disajikan dalam bentuk grafik pada Gambar 11;



Gambar 11. Perbandingan Algoritma Insertion Sort Dengan Variasi Data 50

Terdapat hasil tidak konsisten pada semua teknik *swapping*. Sedangkan rata-rata didapatkan durasi tercepat adalah *swapping* menggunakan operator  $+$  dan  $-$ . 3) Analisis Perbandingan **4** Insertion Sort Dengan Variasi Data 100. Dari Tabel 60 dapat disajikan dalam bentuk grafik pada Gambar 12;

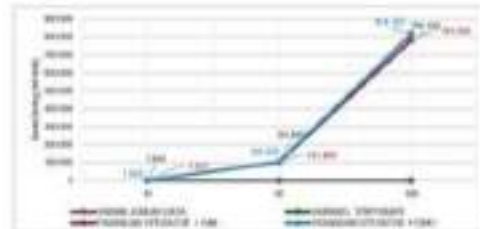


Gambar 12. Perbandingan Algoritma Insertion Sort Dengan Variasi Data 100

Terdapat hasil konsisten pada semua teknik *swapping*. Konsistensi tersebut pada durasi *sorting descending* ke *ascending* lebih cepat daripada durasi *sorting ascending* ke *descending*. Sedangkan rata-rata didapatkan durasi tercepat adalah *swapping* menggunakan variabel *temporary*.

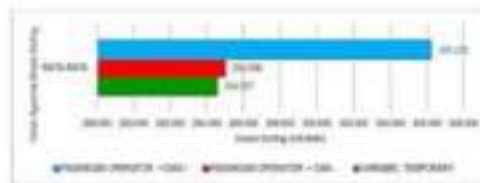
### 3.4. Analisis Perbandingan Bubble, Selection, dan Insertion Sort

Dari Tabel 60 dapat disajikan dalam bentuk grafik perbandingan hasil *sorting* antara algoritma Bubble, Selection, dan Insertion sort menggunakan variabel *temporary*, pasangan + dan -, serta pasangan x dan / yang terdapat pada Gambar 13.



Gambar 13. Perbandingan Algoritma Bubble, Selection, dan Selection Sort dengan Variasi Data 10, 50, dan 100

Gambar 13 menunjukkan, terdapat kondisi semakin banyak data yang diurutkan, baik secara *descending* ke *ascending* ataupun dari *ascending* ke *descending* membutuhkan waktu semakin lama. Terdapat perbedaan durasi *sorting* hasil dari setiap varian jumlah data. Dari hasil analisis perbandingan setiap algoritma dapat disajikan dalam bentuk grafik pada Gambar 14.



Gambar 14. Perbandingan Rata-rata Durasi Sorting Menggunakan Algoritma Bubble, Selection, dan Insertion Sort

Gambar 14 menunjukkan, hasil pengujian rata-rata *sorting* baik menggunakan algoritma Bubble, Selection, dan Insertion sort dengan varian data yang diproses 10, 50, dan 100, serta frekuensi pengujian sebanyak 10 kali pada setiap proses *sorting*. Rata-rata durasi terlama hingga ke tercepat tersusun yaitu, durasi

menggunakan teknik *swapping* menggunakan operator x dan / yaitu 306.236 mili detik, sedangkan durasi menggunakan teknik *swapping* menggunakan operator + dan - yaitu 294.998 mili detik, dan durasi menggunakan teknik *swapping* menggunakan variabel *temporary* yaitu 294.557 mili detik.

## 4. Kesimpulan

*Sorting* tercepat didapatkan oleh teknik *swapping* menggunakan variabel *temporary*, daripada *sorting* menggunakan tanpa variabel *temporary*.

## Ucapan Terimakasih

Terima kasih kepada Pusat Penelitian dan Pengabdian Kepada Masyarakat Universitas Dinamika, yang telah memfasilitasi dalam bentuk dana, dan persetujuannya agar penelitian ini dapat terlaksana.

## Daftar Rujukan

Abdel-Hafeez, S., & Gardo-Ross, A. (2017). An efficient O(N) comparison-free sorting algorithm. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(6), 1930-1942. <https://doi.org/10.1109/TVLSI.2017.26617345>

Anggrani, D., Wibawa, A. P., Primawati, P., & Hermat, H. (2020). Perbandingan Efisiensi Algoritma Sorting dalam Pengujian Bandwidth. *JADM: Jurnal Desain*, 23(2), 96-103. <https://doi.org/10.29999/jkompy1212.538.96-103>

Bonchar, R. O., Thidry, A., Aguilera-Garcia, D., Zochal, M., Moch, H., Tveret, R., Zeller, T., Blankenberg, S., & Ziegler, A. (2022). Comparison of calling pipelines for whole genome sequencing: an empirical study demonstrating the importance of mapping and alignment. *BioRx*, 2022.09.18.508404. <https://doi.org/10.1101/2022.09.18.508404>

Bustami, B., Fadhilah, P., & Alfiansyah, G. (2019). Comparison of Simple Algorithm Data (Sorting) Control Methods on Selection and Bubble Sort. *TECHN - Jurnal Teknik Informatika*, 23(2), 289. <https://doi.org/10.29103/teknis1112.1601>

Ekowati, M. A., Nurdyatama, Z. P., Wilianto, W., & Danarti, K. (2022). Comparative Analysis of the Speed of the Sorting Method on Google Translate Indonesian-English Using Binary Search. *International Journal of Global Operations Research*, 3(2), 108-115. <https://doi.org/10.11944/ijgor.v3i2.167>

Hise Hwe Aung. (2019). Analysis and Comparative of Sorting Algorithms. *Published in International Journal of Trend in Scientific Research and Development*, 3(5), 1049-1053. <https://doi.org/https://doi.org/10.31124/ijtsrd26578>

Imaz, Y., Ben, Atallah, R., Ben, Davyier, D., & Jemou, M. Ben. (2010). A comparative study of sorting algorithms with FPGA acceleration by high level synthesis. *Computacion y Sistemas*, 23(1), 213-230. <https://doi.org/10.13063/CyS-23-1-2099>

Jones, T. B., & Ackley, D. H. (2014). Comparison of sorting algorithms. *Proceedings of the International Conference on Dependable Systems and Networks*, 726-731. <https://doi.org/10.1109/DSN.2014.74>

K, M. R., & I, M. R. (2010). A Comparative Study of Sorting and Searching Algorithms. *International Research Journal of Engineering and Technology (IRJET)*, 06(01), 1412-1416. <https://doi.org/10.5539/irjet.v1i2.p143>

Kavita. (2021). Comparative study of sorting algorithms. *Apel*. <https://doi.org/10.13140/RG.2.2.30552.62724>

- M. Rabi, A. J. (19), E. Y. Baha, B. M. Malyw, Y., & Douda, M. (2022). Performance Comparison of these Sorting Algorithms Using Shared Data and Consistency Mechanism in Java. *Arab-Zone Journal of Basic & Applied Research*, 8(1), 55–64. <https://doi.org/10.55639/03761x>
- Fastio, D., & Akhriv. (17) (2011). A comparative study of parallel sort algorithms. *SPLASHTI'11: Companion - Proceedings of OOPSLA'11, OOPSLA' 2011, GPCE'11, DGS'11, and SPLASHTI'11 Companion*, 203–204. <https://doi.org/10.1145/2048147.2048207>
- Raghibatshi, D. (2018). Data Structure: Theoretical Approach. *International Journal of Trend in Scientific Research and Development*, Volume-3(Issue-11), 268–273. <https://doi.org/10.31144/ijtsrd18077>
- Rajagopal, D., & Thalakavali, K. (2016). Different Sorting Algorithm's Comparison based Upon the Time Complexity. *International Journal of U- and e- Service, Science and Techno*, 9(1), 287–296. <https://doi.org/10.14257/ijunesr.2016.9.8.24>
- Sari, N., Gurawan, W. A., Sari, P. K., Zeki, I., & Syahputra, A. (2022). Analisis Algoritma Bubble Sort Secara Ascending Dan Descending Serta Implementasinya Dengan Menggunakan Bahasa Pemrograman Java. *ADW Bios Digital*, 1(1), 16–25. <https://doi.org/10.34306/adwds/341625>
- Setiawan, R. (2017). Comparing sorting algorithm complexity based on control flow structure. *Proceedings of 2016 International Conference on Information Management and Technology, ICIMTech 2016*, November, 224–228. <https://doi.org/10.1109/ICIMTech.2016.7990334>
- Sonita, A., & Nuraino, F. (2016). Analisis Perbandingan Algoritma Bubble Sort, Merge Sort, Dan Quick Sort Dalam Proses Pengurutan Kombinasi Angka Dan Huruf. *Penelitian*, 2(2), 75–80. <https://doi.org/10.33169/pseniocode.2.2.75-80>
- Surandar, E. (2019). Perbandingan Metode Selection Sort dan Insertion Sort Dalam Pengurutan Data Menggunakan Bahasa Program Java. *Febr*, 13(2), 172–178. <https://doi.org/10.33322/febr.v13i2.485>
- Surandar, E., & Indrianto, I. (2020). Implementasi Algoritma Bubble Sort Terhadap 2 Basis Model Varian Pengurutan Data Menggunakan Bahasa Program Java. *Febr*, 13(2), 255–265. <https://doi.org/10.33322/febr.v13i2.1088>
- Zhao, L., Liu, X., & Shao, X. (2016). Comparative Analysis of Several Sorting Algorithms in Java Language. *Acwive*, 415–419. <https://doi.org/10.2990/acwive-16.2016.68>

## ORIGINALITY REPORT

---

23%

SIMILARITY INDEX

22%

INTERNET SOURCES

9%

PUBLICATIONS

14%

STUDENT PAPERS

---

## PRIMARY SOURCES

---

1	Submitted to Universitas Putera Indonesia YPTK Padang Student Paper	2%
2	<a href="http://www.coursehero.com">www.coursehero.com</a> Internet Source	1%
3	<a href="http://stt-pln.e-journal.id">stt-pln.e-journal.id</a> Internet Source	1%
4	<a href="http://eprints.uns.ac.id">eprints.uns.ac.id</a> Internet Source	1%
5	<a href="http://www.scribd.com">www.scribd.com</a> Internet Source	1%
6	<a href="http://www.adi-journal.org">www.adi-journal.org</a> Internet Source	1%
7	<a href="http://eprints.ums.ac.id">eprints.ums.ac.id</a> Internet Source	1%
8	Submitted to Universitas Pamulang Student Paper	1%
9	<a href="http://www.iieta.org">www.iieta.org</a> Internet Source	1%

---



10	<a href="http://www.biorxiv.org">www.biorxiv.org</a> Internet Source	1 %
11	Submitted to Unicaf University Student Paper	1 %
12	Submitted to Westcliff University Student Paper	1 %
13	Submitted to Flinders University Student Paper	1 %
14	<a href="http://m.earticle.net">m.earticle.net</a> Internet Source	1 %
15	<a href="http://repo.darmajaya.ac.id">repo.darmajaya.ac.id</a> Internet Source	1 %
16	<a href="http://repository.iainpalopo.ac.id">repository.iainpalopo.ac.id</a> Internet Source	1 %
17	<a href="http://cris.technion.ac.il">cris.technion.ac.il</a> Internet Source	<1 %
18	<a href="http://iorajournal.org">iorajournal.org</a> Internet Source	<1 %
19	<a href="http://azjournalbar.com">azjournalbar.com</a> Internet Source	<1 %
20	<a href="http://eprints.binus.ac.id">eprints.binus.ac.id</a> Internet Source	<1 %
21	<a href="http://europub.co.uk">europub.co.uk</a> Internet Source	<1 %

22	<a href="http://jsisfotek.org">jsisfotek.org</a> Internet Source	<1 %
23	<a href="http://www.scielo.org.mx">www.scielo.org.mx</a> Internet Source	<1 %
24	Submitted to Cyprus Academic Library Consortium Student Paper	<1 %
25	<a href="http://www.infeb.org">www.infeb.org</a> Internet Source	<1 %
26	Submitted to Kent Institute of Business and Technology Student Paper	<1 %
27	<a href="http://ieeexplore.ieee.org">ieeexplore.ieee.org</a> Internet Source	<1 %
28	Submitted to Institut Teknologi Kalimantan Student Paper	<1 %
29	<a href="http://seminar.iaii.or.id">seminar.iaii.or.id</a> Internet Source	<1 %
30	Submitted to Colorado Technical University Online Student Paper	<1 %
31	<a href="http://dspace.uui.ac.id">dspace.uui.ac.id</a> Internet Source	<1 %
32	<a href="http://oa.upm.es">oa.upm.es</a> Internet Source	<1 %

33	<a href="http://sikarsa.um.ac.id">sikarsa.um.ac.id</a> Internet Source	<1 %
34	<a href="http://suwokoindustries.blogspot.com">suwokoindustries.blogspot.com</a> Internet Source	<1 %
35	Submitted to University of Sheffield Student Paper	<1 %
36	<a href="http://article.nadiapub.com">article.nadiapub.com</a> Internet Source	<1 %
37	<a href="http://blogbugabagi.blogspot.com">blogbugabagi.blogspot.com</a> Internet Source	<1 %
38	<a href="http://es.scribd.com">es.scribd.com</a> Internet Source	<1 %
39	<a href="http://www.atlantis-press.com">www.atlantis-press.com</a> Internet Source	<1 %
40	<a href="http://123dok.com">123dok.com</a> Internet Source	<1 %
41	<a href="http://a-research.upi.edu">a-research.upi.edu</a> Internet Source	<1 %
42	<a href="http://bg.copernicus.org">bg.copernicus.org</a> Internet Source	<1 %
43	<a href="http://digilib.uinsby.ac.id">digilib.uinsby.ac.id</a> Internet Source	<1 %
44	<a href="http://repositorio.iscte-iul.pt">repositorio.iscte-iul.pt</a> Internet Source	<1 %

45 repository.um.ac.id

Internet Source

<1 %

---

46 Saad AL-Azzam, Mohammad Qatawneh.  
"Parallel Processing of Sorting and Searching  
Algorithms Comparative Study", Modern  
Applied Science, 2018

Publication

<1 %

---

47 Peeyush Kumar, Ayushe Gangal, Sunita  
Kumari. "Recombinant Sort: N-Dimensional  
Cartesian Spaced Algorithm Designed from  
Synergetic Combination of Hashing, Bucket,  
Counting and Radix Sort", Ingénierie des  
systèmes d information, 2020

Publication

<1 %

---

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off