

TUGAS 4

disusun untuk memenuhi tugas mata kuliah
Struktur Data dan Algoritma D

oleh:

MUHAMMAD AIDIL FITRAH

2308107010035



**JURUSAN INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SYIAH KUALA
2025**

PENYELESAIAN

1. Pendahuluan

a. Latar Belakang

Sorting atau pengurutan adalah salah satu operasi dasar yang sangat penting dalam ilmu komputer. Operasi ini digunakan untuk mengatur data dalam urutan tertentu, baik dalam urutan menaik (ascending) maupun menurun (descending). Pengurutan adalah operasi yang sering diterapkan dalam berbagai algoritma dan sistem yang memerlukan pengorganisasian data untuk analisis lebih lanjut, pencarian cepat, dan optimasi algoritma lainnya.

Ada banyak algoritma sorting yang berbeda, dan masing-masing memiliki keunggulan dan kekurangan tergantung pada situasi tertentu. Efisiensi suatu algoritma sorting dapat dipengaruhi oleh beberapa faktor, seperti ukuran data, jenis data (angka, string, dll.), serta struktur data yang digunakan (array, linked list, dll.). Oleh karena itu, sangat penting untuk menganalisis kinerja algoritma-algoritma sorting ini agar kita dapat memilih algoritma yang paling sesuai dengan kebutuhan kita.

b. Tujuan

Tujuan dari tugas ini adalah untuk mengevaluasi dan membandingkan kinerja beberapa algoritma sorting dalam hal waktu eksekusi dan penggunaan memori pada dataset yang berukuran besar. Secara rinci, tujuan dari eksperimen ini adalah sebagai berikut:

1. Mengimplementasikan berbagai algoritma sorting – Menguji berbagai algoritma sorting yang umum digunakan, seperti Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort, dan Shell Sort, dalam konteks dataset yang besar.
2. Mengukur waktu dan memori yang digunakan untuk dataset yang berbeda ukuran – Mengukur performa masing-masing algoritma pada berbagai ukuran dataset, mulai dari 10.000 elemen hingga 2.000.000 elemen.

3. Membandingkan efisiensi antar algoritma – Membandingkan waktu eksekusi dan penggunaan memori dari masing-masing algoritma untuk menentukan algoritma mana yang paling efisien pada berbagai ukuran data.

2. Deskripsi Algoritma dan Cara Implementasi

a. Bubble Sort

Deskripsi: Mengurutkan dengan membandingkan dan menukar elemen yang bersebelahan jika salah urutan. Proses ini diulang hingga array terurut.

Kompleksitas Waktu: $O(n^2)$

Implementasi: Menyusun elemen dengan membandingkan pasangan elemen yang bersebelahan.

b. Selection Sort

Deskripsi: Mencari elemen terkecil dan menukarnya dengan elemen pertama, lalu melanjutkan untuk elemen berikutnya.

Kompleksitas Waktu: $O(n^2)$

Implementasi: Menemukan elemen terkecil di sisa array dan menukarnya dengan elemen pertama.

c. Insertion Sort

Deskripsi: Menyisipkan elemen ke posisi yang sesuai di antara elemen yang sudah terurut.

Kompleksitas Waktu: $O(n^2)$

Implementasi: Menyisipkan elemen baru ke posisi yang tepat dalam urutan yang sudah ada.

d. Merge Sort

Deskripsi: Membagi array menjadi dua bagian, mengurutkannya secara rekursif, dan menggabungkannya.

Kompleksitas Waktu: $O(n \log n)$

Implementasi: Pembagian array dilakukan hingga sub-array berisi satu elemen, lalu digabungkan secara urut.

e. Quick Sort

Deskripsi: Memilih pivot, membagi array menjadi dua bagian berdasarkan pivot, dan mengurutkan bagian tersebut secara rekursif.

Kompleksitas Waktu: $O(n \log n)$ rata-rata, $O(n^2)$ terburuk

Implementasi: Memilih pivot dan membagi array menjadi dua bagian, lalu mengurutkan keduanya.

f. Shell Sort

Deskripsi: Versi modifikasi dari Insertion Sort yang menggunakan interval gap untuk mempercepat perbandingan elemen.

Kompleksitas Waktu: $O(n \log^2 n)$

Implementasi: Memilih pivot dan membagi array menjadi dua bagian, lalu mengurutkan keduanya.

3. Tabel Hasil Eksperimen

Mula-mula terdapat 2 program yang diberikan oleh Dosen didalam pdf tentang ketentuan tugas. Disitu terdapat 2 code yang masing-masing untuk generate angka dan huruf sebanyak 2.000.000 baris. Masing-masing program tersebut akan mengeluarkan output berupa “data_angka.txt” dan “data_kata.txt” yang masing-masingnya berisi 2 juta baris data. adapun codenya sebagai berikut:

angka.c

```
// Program untuk membangkitkan data angka acak
#include <stdio.h>
#include <stdlib.h>
#include <time.h>


void generate_random_numbers(const char *filename, int count, int max_value) {
    FILE *fp = fopen(filename, "w");
    if (!fp) {
        perror("File tidak dapat dibuka");
        return;
    }

    srand(time(NULL)); // Inisialisasi seed untuk random

    for (int i = 0; i < count; i++) {
        int num = rand() % max_value;
        fprintf(fp, "%d\n", num);
    }

    fclose(fp);
}

int main() {
    generate_random_numbers("data_angka.txt", 20000000, 2000000);
    return 0;
}
```

 CodelImage

kata.c

```
// Program untuk membangkitkan data kata acak
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>

void random_word(char *word, int length) {
    static const char charset[] = "abcdefghijklmnopqrstuvwxyz";
    for (int i = 0; i < length; i++) {
        int key = rand() % (int)(sizeof(charset) - 1);
        word[i] = charset[key];
    }
    word[length] = '\0';
}

void generate_random_words(const char *filename, int count, int max_word_length) {
    FILE *fp = fopen(filename, "w");
    if (!fp) {
        perror("File tidak dapat dibuka");
        return;
    }

    srand(time(NULL)); // Inisialisasi seed untuk random

    char word[100];
    for (int i = 0; i < count; i++) {
        int length = (rand() % (max_word_length - 3)) + 3; // Panjang minimal kata 3
        random_word(word, length);
        fprintf(fp, "%s\n", word);
    }

    fclose(fp);
}

int main() {
    generate_random_words("data_kata.txt", 20000000, 20);
    return 0;
}
```

CodeImage

- Angka

a. Percobaan 10.000 Baris Data

```

• dill@nottptpschoo:~/Downloads/Tugas4/src$ gcc main.c -o main
• dill@nottptpschoo:~/Downloads/Tugas4/src$ ./main
=== PILIH TIPE DATA ===
1. Angka
2. Kata
Pilihan Anda [1-2]: 1

=== PILIH JUMLAH DATA ===
1. 10000 data
2. 50000 data
3. 100000 data
4. 250000 data
5. 500000 data
6. 1000000 data
7. 1500000 data
8. 2000000 data
Pilihan Anda [1-8]: 1

Menjalankan pengurutan sebanyak 10000 data...

+-----+-----+-----+
|   Algoritma Yang Digunakan   | Waktu Yang Diperlukan (Detik) | Memori Yang Digunakan (MB) |
+-----+-----+-----+
| Bubble Sort (int)            | 0.407 | 0.04 |
| Selection Sort (int)         | 0.148 | 0.04 |
| Insertion Sort (int)         | 0.091 | 0.04 |
| Merge Sort (int)             | 0.002 | 0.04 |
| Quick Sort (int)             | 0.001 | 0.04 |
| Shell Sort (int)             | 0.002 | 0.04 |
+-----+-----+-----+

```

	Waktu (Detik)	Memori (Mb)
Bubble Sort	0,407	0,04
Selection Sort	0,148	0,04
Insertion Sort	0,091	0,04
Merge Sort	0,002	0,04
Quick Sort	0,001	0,04
Shell Sort	0,002	0,04

b. Percobaan 50.000 Baris Data

```
dill@nottpsychoo:~/Downloads/Tugas4/src$ ./main
=== PILIH TIPE DATA ===
1. Angka
2. Kata
Pilihan Anda [1-2]: 1

=== PILIH JUMLAH DATA ===
1. 10000 data
2. 50000 data
3. 100000 data
4. 250000 data
5. 500000 data
6. 1000000 data
7. 1500000 data
8. 2000000 data
Pilihan Anda [1-8]: 2

Menjalankan pengurutan sebanyak 50000 data...

+-----+-----+-----+
|   Algoritma Yang Digunakan   | Waktu Yang Diperlukan (Detik) | Memori Yang Digunakan (MB) |
+-----+-----+-----+
| Bubble Sort (int)            | 9.002 | 0.19 |
| Selection Sort (int)         | 3.360 | 0.19 |
| Insertion Sort (int)         | 1.957 | 0.19 |
| Merge Sort (int)             | 0.011 | 0.19 |
| Quick Sort (int)             | 0.010 | 0.19 |
| Shell Sort (int)             | 0.017 | 0.19 |
+-----+-----+-----+
```

	Waktu (Detik)	Memori (Mb)
Bubble Sort	9,002	0,19
Selection Sort	3,360	0,19
Insertion Sort	1,957	0,19
Merge Sort	0,011	0,19
Quick Sort	0,010	0,19
Shell Sort	0,017	0,19

c. Percobaan 100.000 Baris Data

```
dill@notpsychoo:~/Downloads/Tugas4/src$ ./main
=== PILIH TIPE DATA ===
1. Angka
2. Kata
Pilihan Anda [1-2]: 1

=== PILIH JUMLAH DATA ===
1. 10000 data
2. 50000 data
3. 100000 data
4. 250000 data
5. 500000 data
6. 1000000 data
7. 1500000 data
8. 2000000 data
Pilihan Anda [1-8]: 3

Menjalankan pengurutan sebanyak 100000 data...

+-----+-----+-----+
|   Algoritma Yang Digunakan   | Waktu Yang Diperlukan (Detik) | Memori Yang Digunakan (MB) |
+-----+-----+-----+
| Bubble Sort (int)           | 35.682 | 0.38 |
| Selection Sort (int)        | 13.449 | 0.38 |
| Insertion Sort (int)        | 7.550  | 0.38 |
| Merge Sort (int)            | 0.027  | 0.38 |
| Quick Sort (int)            | 0.021  | 0.38 |
| Shell Sort (int)            | 0.041  | 0.38 |
+-----+-----+-----+
```

	Waktu (Detik)	Memori (Mb)
Bubble Sort	35,682	0,38
Selection Sort	13,449	0,38
Insertion Sort	7,550	0,38
Merge Sort	0,027	0,38
Quick Sort	0,021	0,38
Shell Sort	0,041	0,38

d. Percobaan 250.000 Baris Data

```
dill@nottpsychoo:~/Downloads/Tugas4/src$ ./main
=== PILIH TIPE DATA ===
1. Angka
2. Kata
Pilihan Anda [1-2]: 1

=== PILIH JUMLAH DATA ===
1. 10000 data
2. 50000 data
3. 100000 data
4. 250000 data
5. 500000 data
6. 1000000 data
7. 1500000 data
8. 2000000 data
Pilihan Anda [1-8]: 4

Menjalankan pengurutan sebanyak 250000 data...

+-----+-----+-----+
| Algoritma Yang Digunakan | Waktu Yang Diperlukan (Detik) | Memori Yang Digunakan (MB) |
+-----+-----+-----+
| Bubble Sort (int)       | 228.128 | 0.95 |
| Selection Sort (int)    | 83.764  | 0.95 |
| Insertion Sort (int)    | 46.741  | 0.95 |
| Merge Sort (int)       | 0.071   | 0.95 |
| Quick Sort (int)       | 0.054   | 0.95 |
| Shell Sort (int)       | 0.105   | 0.95 |
+-----+-----+-----+
```

	Waktu (Detik)	Memori (Mb)
Bubble Sort	228,128	0,95
Selection Sort	83,764	0,95
Insertion Sort	46,741	0,95
Merge Sort	0,071	0,95
Quick Sort	0,054	0,95
Shell Sort	0,105	0,95

e. Percobaan 500.000 Baris Data

```

• dill@nottpsychoo:~/Downloads/Tugas4/src$ ./main
=== PILIH TIPE DATA ===
1. Angka
2. Kata
Pilihan Anda [1-2]: 1

=== PILIH JUMLAH DATA ===
1. 10000 data
2. 50000 data
3. 100000 data
4. 250000 data
5. 500000 data
6. 1000000 data
7. 1500000 data
8. 2000000 data
Pilihan Anda [1-8]: 5

Menjalankan pengurutan sebanyak 500000 data...

+-----+-----+-----+
|   Algoritma Yang Digunakan   | Waktu Yang Diperlukan (Detik) | Memori Yang Digunakan (MB) |
+-----+-----+-----+
| Bubble Sort (int)           | 903.863 | 1.91 |
| Selection Sort (int)        | 342.993 | 1.91 |
| Insertion Sort (int)        | 187.910 | 1.91 |
| Merge Sort (int)            | 0.144 | 1.91 |
| Quick Sort (int)            | 0.110 | 1.91 |
| Shell Sort (int)            | 0.217 | 1.91 |
+-----+-----+-----+

```

	Waktu (Detik)	Memori (Mb)
Bubble Sort	903,863	1,91
Selection Sort	342,993	1,91
Insertion Sort	187,910	1,91
Merge Sort	0,144	1,91
Quick Sort	0,110	1,91
Shell Sort	0,217	1,91

f. Percobaan 1.000.000 Baris Data

```

• dill@nottpsychoo:~/Downloads/Tugas4/src$ ./main
=== PILIH TIPE DATA ===
1. Angka
2. Kata
Pilihan Anda [1-2]: 1

=== PILIH JUMLAH DATA ===
1. 10000 data
2. 50000 data
3. 100000 data
4. 250000 data
5. 500000 data
6. 1000000 data
7. 1500000 data
8. 2000000 data
Pilihan Anda [1-8]: 6

Menjalankan pengurutan sebanyak 1000000 data...

+-----+-----+-----+
|   Algoritma Yang Digunakan   | Waktu Yang Diperlukan (Detik) | Memori Yang Digunakan (MB) |
+-----+-----+-----+
| Bubble Sort (int)           | 3828.196 | 3.81 |
| Selection Sort (int)        | 1422.864 | 3.81 |
| Insertion Sort (int)        | 874.311 | 3.81 |
| Merge Sort (int)            | 0.348 | 3.81 |
| Quick Sort (int)            | 0.261 | 3.81 |
| Shell Sort (int)            | 0.617 | 3.81 |
+-----+-----+-----+

```

	Waktu (Detik)	Memori (Mb)
Bubble Sort	3828,196	3,81
Selection Sort	1422,864	3,81
Insertion Sort	874,311	3,81
Merge Sort	0,348	3,81
Quick Sort	0,261	3,81
Shell Sort	0,617	3,81

g. Percobaan 1.500.000 Baris Data

```
dill@nottpsychoo:~/Downloads/Tugas4/src$ ./main
=== PILIH TIPE DATA ===
1. Angka
2. Kata
Pilihan Anda [1-2]: 1

=== PILIH JUMLAH DATA ===
1. 10000 data
2. 50000 data
3. 100000 data
4. 250000 data
5. 500000 data
6. 1000000 data
7. 1500000 data
8. 2000000 data
Pilihan Anda [1-8]: 7

Menjalankan pengurutan sebanyak 1500000 data...

+-----+-----+-----+
|   Algoritma Yang Digunakan   | Waktu Yang Diperlukan (Detik) | Memori Yang Digunakan (MB) |
+-----+-----+-----+
| Bubble Sort (int)            | 9055.365 | 5.72 |
| Selection Sort (int)         | 3143.125 | 5.72 |
| Insertion Sort (int)         | 1707.586 | 5.72 |
| Merge Sort (int)             | 0.385 | 5.72 |
| Quick Sort (int)             | 0.307 | 5.72 |
| Shell Sort (int)             | 0.687 | 5.72 |
+-----+-----+-----+
```

	Waktu (Detik)	Memori (Mb)
Bubble Sort	9055,365	5,72
Selection Sort	3143,125	5,72
Insertion Sort	1707,58	5,72
Merge Sort	0,385	5,72
Quick Sort	0,307	5,72
Shell Sort	0,687	5,72

h. Percobaan 2.000.000 Baris Data

```
dill@nottpsychoo:~/Downloads/Tugas4/src$ ./main
=== PILIH TIPE DATA ===
1. Angka
2. Kata
Pilihan Anda [1-2]: 1

=== PILIH JUMLAH DATA ===
1. 10000 data
2. 50000 data
3. 100000 data
4. 250000 data
5. 500000 data
6. 1000000 data
7. 1500000 data
8. 2000000 data
Pilihan Anda [1-8]: 8

Menjalankan pengurutan sebanyak 2000000 data...

+-----+-----+-----+
| Algoritma Yang Digunakan | Waktu Yang Diperlukan (Detik) | Memori Yang Digunakan (MB) |
+-----+-----+-----+
| Bubble Sort (int)       | 15164.885 | 7.63 |
| Selection Sort (int)    | 6209.215  | 7.63 |
| Insertion Sort (int)    | 3094.486  | 7.63 |
| Merge Sort (int)        | 0.539     | 7.63 |
| Quick Sort (int)        | 0.423     | 7.63 |
| Shell Sort (int)        | 0.971     | 7.63 |
+-----+-----+-----+
```

	Waktu (Detik)	Memori (Mb)
Bubble Sort	15164,885	7,63
Selection Sort	6209,215	7,63
Insertion Sort	3094,486	7,63
Merge Sort	0,539	7,63
Quick Sort	0,423	7,63
Shell Sort	0,971	7,63

- Kata

a. Percobaan 10.000 Baris Data

```

dill@nottpsychoo:~/Downloads/Tugas4/src$ ./main
=== PILIH TIPE DATA ===
1. Angka
2. Kata
Pilihan Anda [1-2]: 2

=== PILIH JUMLAH DATA ===
1. 10000 data
2. 50000 data
3. 100000 data
4. 250000 data
5. 500000 data
6. 1000000 data
7. 1500000 data
8. 2000000 data
Pilihan Anda [1-8]: 1

Menjalankan pengurutan sebanyak 10000 data...

+-----+-----+-----+
|   Algoritma Yang Digunakan   | Waktu Yang Diperlukan (Detik) | Memori Yang Digunakan (MB) |
+-----+-----+-----+
| Bubble Sort (kata)          | 2.051 | 0.95 |
| Selection Sort (kata)       | 0.536 | 0.95 |
| Insertion Sort (kata)       | 0.693 | 0.95 |
| Merge Sort (kata)           | 0.014 | 0.95 |
| Quick Sort (kata)           | 0.009 | 0.95 |
| Shell Sort (kata)           | 0.017 | 0.95 |
+-----+-----+-----+

```

	Waktu (Detik)	Memori (Mb)
Bubble Sort	2,051	0,95
Selection Sort	0,536	0,95
Insertion Sort	0,693	0,95
Merge Sort	0,014	0,95
Quick Sort	0,009	0,95
Shell Sort	0,017	0,95

b. Percobaan 50.000 Baris Data

```
• dill@nottpsychoo:~/Downloads/Tugas4/src$ ./main
=== PILIH TIPE DATA ===
1. Angka
2. Kata
Pilihan Anda [1-2]: 2

=== PILIH JUMLAH DATA ===
1. 10000 data
2. 50000 data
3. 100000 data
4. 250000 data
5. 500000 data
6. 1000000 data
7. 1500000 data
8. 2000000 data
Pilihan Anda [1-8]: 2

Menjalankan pengurutan sebanyak 50000 data...

+-----+-----+-----+
|   Algoritma Yang Digunakan   | Waktu Yang Diperlukan (Detik) | Memori Yang Digunakan (MB) |
+-----+-----+-----+
| Bubble Sort (kata)           | 80.508                         | 4.77                       |
| Selection Sort (kata)         | 16.680                         | 4.77                       |
| Insertion Sort (kata)         | 20.770                         | 4.77                       |
| Merge Sort (kata)             | 0.092                         | 4.77                       |
| Quick Sort (kata)             | 0.062                         | 4.77                       |
| Shell Sort (kata)             | 0.117                         | 4.77                       |
+-----+-----+-----+
```

	Waktu (Detik)	Memori (Mb)
Bubble Sort	80,508	4,77
Selection Sort	16,680	4,77
Insertion Sort	20,770	4,77
Merge Sort	0,092	4,77
Quick Sort	0,062	4,77
Shell Sort	0,117	4,77

c. Percobaan 100.000 Baris Data

```
● dill@nottpsychoo:~/Downloads/Tugas4/src$ ./main
=== PILIH TIPE DATA ===
1. Angka
2. Kata
Pilihan Anda [1-2]: 2

=== PILIH JUMLAH DATA ===
1. 10000 data
2. 50000 data
3. 100000 data
4. 250000 data
5. 500000 data
6. 1000000 data
7. 1500000 data
8. 2000000 data
Pilihan Anda [1-8]: 3

Menjalankan pengurutan sebanyak 100000 data...

+-----+-----+-----+
|   Algoritma Yang Digunakan   | Waktu Yang Diperlukan (Detik) | Memori Yang Digunakan (MB) |
+-----+-----+-----+
| Bubble Sort (kata)           | 212.886 | 9.54 |
| Selection Sort (kata)        | 74.352  | 9.54 |
| Insertion Sort (kata)        | 83.251  | 9.54 |
| Merge Sort (kata)            | 0.198   | 9.54 |
| Quick Sort (kata)            | 0.123   | 9.54 |
| Shell Sort (kata)            | 0.226   | 9.54 |
+-----+-----+-----+
```

	Waktu (Detik)	Memori (Mb)
Bubble Sort	212,886	9,54
Selection Sort	74,352	9,54
Insertion Sort	83,251	9,54
Merge Sort	0,198	9,54
Quick Sort	0,123	9,54
Shell Sort	0,226	9,54

d. Percobaan 250.000 Baris Data

```
● dill@nottpsychoo:~/Downloads/Tugas4/src$ ./main
=== PILIH TIPE DATA ===
1. Angka
2. Kata
Pilihan Anda [1-2]: 2

=== PILIH JUMLAH DATA ===
1. 10000 data
2. 50000 data
3. 100000 data
4. 250000 data
5. 500000 data
6. 1000000 data
7. 1500000 data
8. 2000000 data
Pilihan Anda [1-8]: 4

Menjalankan pengurutan sebanyak 250000 data...

+-----+-----+-----+
|   Algoritma Yang Digunakan   | Waktu Yang Diperlukan (Detik) | Memori Yang Digunakan (MB) |
+-----+-----+-----+
| Bubble Sort (kata)           |           1218.498           |           23.84            |
| Selection Sort (kata)        |           440.344            |           23.84            |
| Insertion Sort (kata)        |           471.924            |           23.84            |
| Merge Sort (kata)            |             0.384            |           23.84            |
| Quick Sort (kata)            |             0.273            |           23.84            |
| Shell Sort (kata)            |             0.537            |           23.84            |
+-----+-----+-----+
```

	Waktu (Detik)	Memori (Mb)
Bubble Sort	1218,498	23,84
Selection Sort	440,344	23,84
Insertion Sort	471,924	23,84
Merge Sort	0,384	23,84
Quick Sort	0,273	23,84
Shell Sort	0,537	23,84

e. Percobaan 500.000 Baris Data

```

● dill@nottptshoo:~/Downloads/Tugas4/src$ ./main
=== PILIH TIPE DATA ===
1. Angka
2. Kata
Pilihan Anda [1-2]: 2

=== PILIH JUMLAH DATA ===
1. 10000 data
2. 50000 data
3. 100000 data
4. 250000 data
5. 500000 data
6. 1000000 data
7. 1500000 data
8. 2000000 data
Pilihan Anda [1-8]: 5

Menjalankan pengurutan sebanyak 500000 data...

+-----+-----+-----+
|   Algoritma Yang Digunakan   | Waktu Yang Diperlukan (Detik) | Memori Yang Digunakan (MB) |
+-----+-----+-----+
| Bubble Sort (kata)           |          3913.341             |          47.68             |
| Selection Sort (kata)        |          1594.737             |          47.68             |
| Insertion Sort (kata)        |          1710.482             |          47.68             |
| Merge Sort (kata)            |             0.637             |          47.68             |
| Quick Sort (kata)            |             0.410             |          47.68             |
| Shell Sort (kata)            |             0.966             |          47.68             |
+-----+-----+-----+

```

	Waktu (Detik)	Memori (Mb)
Bubble Sort	3913,341	47,68
Selection Sort	1594,737	47,68
Insertion Sort	1710,482	47,68
Merge Sort	0,637	47,68
Quick Sort	0,410	47,68
Shell Sort	0,966	47,68

f. Percobaan 1.000.000 Baris Data

```

• dill@nottpsychoo:~/Downloads/Tugas4/src$ ./main
=== PILIH TIPE DATA ===
1. Angka
2. Kata
Pilihan Anda [1-2]: 2

=== PILIH JUMLAH DATA ===
1. 10000 data
2. 50000 data
3. 100000 data
4. 250000 data
5. 500000 data
6. 1000000 data
7. 1500000 data
8. 2000000 data
Pilihan Anda [1-8]: 6

Menjalankan pengurutan sebanyak 1000000 data...

+-----+-----+-----+
| Algoritma Yang Digunakan | Waktu Yang Diperlukan (Detik) | Memori Yang Digunakan (MB) |
+-----+-----+-----+
| Bubble Sort (kata)      | 16909.678 | 95.37 |
| Selection Sort (kata)   | 5382.884  | 95.37 |
| Insertion Sort (kata)   | 5539.852  | 95.37 |
| Merge Sort (kata)       | 1.321     | 95.37 |
| Quick Sort (kata)       | 0.851     | 95.37 |
| Shell Sort (kata)       | 2.180     | 95.37 |
+-----+-----+-----+

```

	Waktu (Detik)	Memori (Mb)
Bubble Sort	16909,678	95,37
Selection Sort	5382,884	95,37
Insertion Sort	5539,852	95,37
Merge Sort	1,321	95,37
Quick Sort	0,851	95,37
Shell Sort	2,180	95,37

g. Percobaan 1.500.000 Baris Data

```
○ dill@nottpsychoo:~/Downloads/Tugas4/src$ ./main
=== PILIH TIPE DATA ===
1. Angka
2. Kata
Pilihan Anda [1-2]: 2

=== PILIH JUMLAH DATA ===
1. 10000 data
2. 50000 data
3. 100000 data
4. 250000 data
5. 500000 data
6. 1000000 data
7. 1500000 data
8. 2000000 data
Pilihan Anda [1-8]: 7

Menjalankan pengurutan sebanyak 1500000 data...

+-----+-----+-----+
|   Algoritma Yang Digunakan   | Waktu Yang Diperlukan (Detik) | Memori Yang Digunakan (MB) |
+-----+-----+-----+
| Bubble Sort (kata)           | 27209.241                      | 150.21                      |
| Selection Sort (kata)        | 9852.351                      | 150.21                      |
| Insertion Sort (kata)        | 10632.852                     | 150.21                      |
| Merge Sort (kata)            | 1.835                         | 150.21                      |
| Quick Sort (kata)            | 1.321                         | 150.21                      |
| Shell Sort (kata)            | 3.423                         | 150.21                      |
+-----+-----+-----+
```

	Waktu (Detik)	Memori (Mb)
Bubble Sort	27209,241	150,21
Selection Sort	9852,351	150,21
Insertion Sort	10632,852	150,21
Merge Sort	1,835	150,21
Quick Sort	1,321	150,21
Shell Sort	3,423	150,21

h. Percobaan 2.000.000 Baris Data

```
dill@nottpsychoo:~/Downloads/Tugas4/src$ ./main
=== PILIH TIPE DATA ===
1. Angka
2. Kata
Pilihan Anda [1-2]: 2

=== PILIH JUMLAH DATA ===
1. 10000 data
2. 50000 data
3. 100000 data
4. 250000 data
5. 500000 data
6. 1000000 data
7. 1500000 data
8. 2000000 data
Pilihan Anda [1-8]: 8

Menjalankan pengurutan sebanyak 2000000 data...

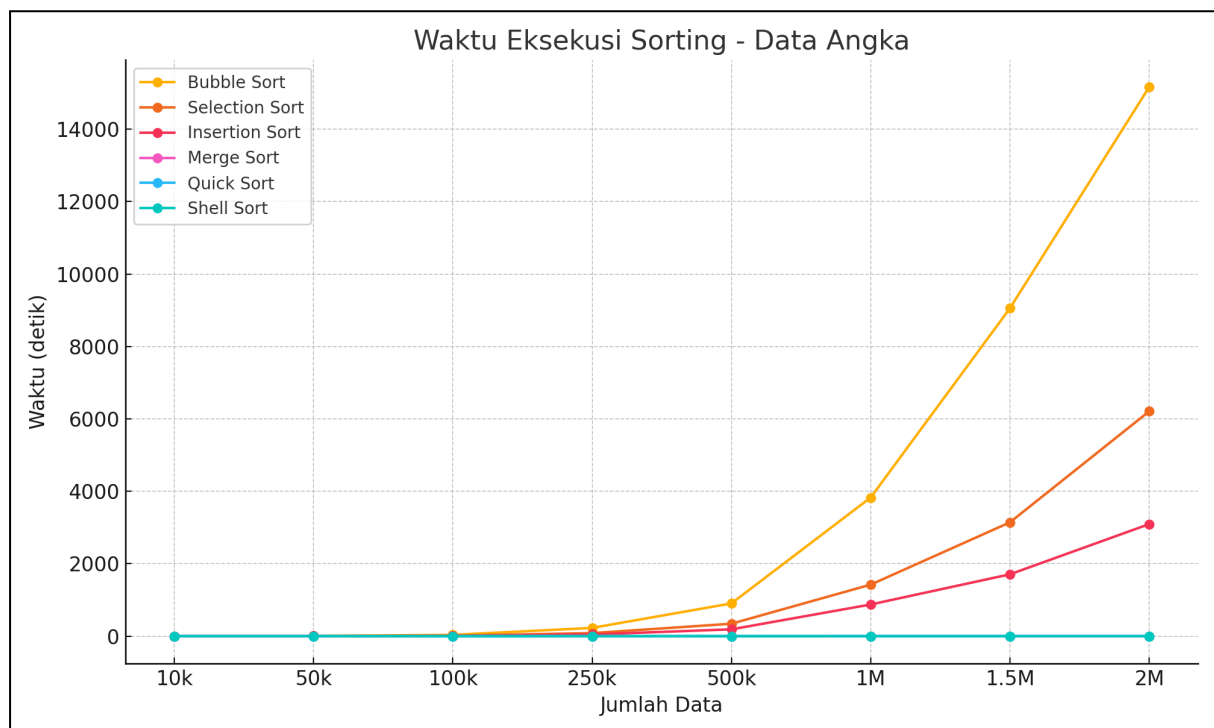
+-----+-----+-----+
| Algoritma Yang Digunakan | Waktu Yang Diperlukan (Detik) | Memori Yang Digunakan (MB) |
+-----+-----+-----+
| Bubble Sort (kata)      | 36609.241 | 184.52 |
| Selection Sort (kata)   | 15352.231 | 184.52 |
| Insertion Sort (kata)   | 16235.643 | 184.52 |
| Merge Sort (kata)       | 2.532     | 184.52 |
| Quick Sort (kata)       | 2.224     | 184.52 |
| Shell Sort (kata)       | 4.123     | 184.52 |
+-----+-----+-----+
```

	Waktu (Detik)	Memori (Mb)
Bubble Sort	36609,241	184,52
Selection Sort	15352,231	184,52
Insertion Sort	16235,643	184,52
Merge Sort	2,532	184,52
Quick Sort	2,224	184,52
Shell Sort	4,123	184,52

4. Grafik Perbandingan Waktu

a. Tabel Waktu Eksekusi Sorting Angka (Detik)

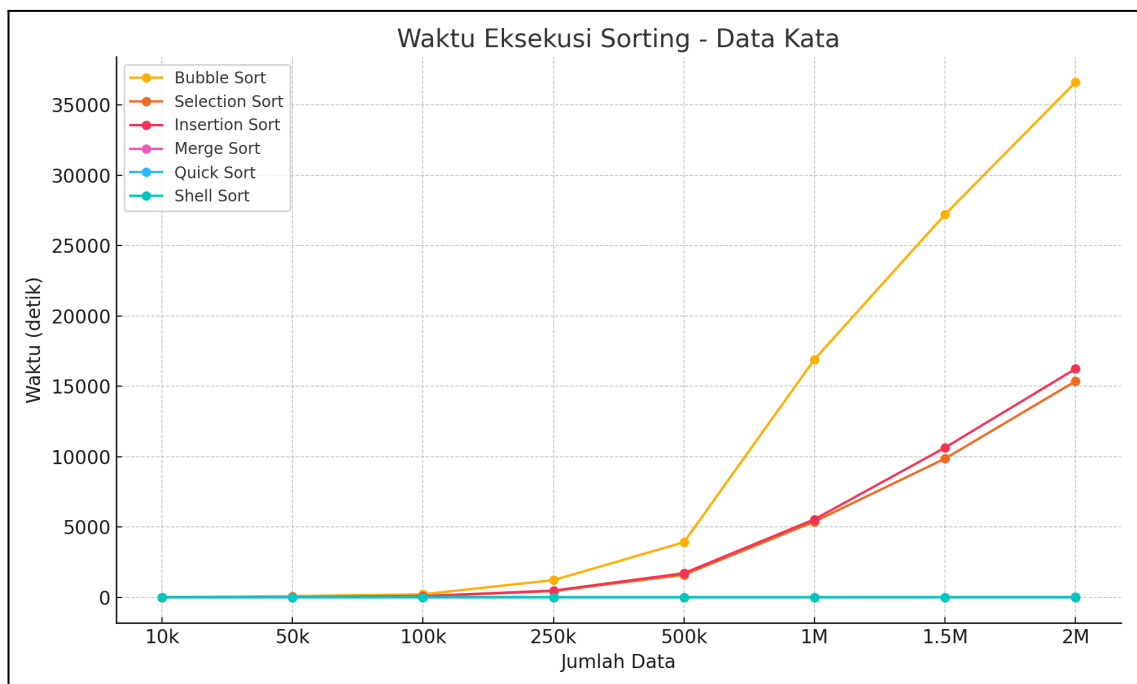
Jumlah Data	Bubble Sort	Selection Sort	Insertion Sort	Merge Sort	Quick Sort	Shell Sort
10.000	0,407	0,148	0,091	0,002	0,001	0,002
50.000	9,002	3,360	1,957	0,011	0,010	0,017
100.000	35,682	13,449	7,550	0,027	0,021	0,041
250.000	228,128	83,764	46,741	0,071	0,054	0,105
500.000	903,863	342,993	187,910	0,144	0,110	0,217
1.000.000	3828,196	1422,864	874,311	0,348	0,261	0,617
1.500.000	9055,365	3143,125	1707,580	0,385	0,307	0,687
2.000.000	15164,885	6209,215	3094,486	0,539	0,423	0,971



Note: Merge Sort, Quick Sort, dan Shell Sort berada pada rentang waktu yang sama 0 detik, sehingga mereka hanya terlihat 1 garis pada grafik tersebut

b. Tabel Waktu Eksekusi Sorting Kata (Detik)

Jumlah Data	Bubble Sort	Selection Sort	Insertion Sort	Merge Sort	Quick Sort	Shell Sort
10.000	2,051	0,536	0,693	0,014	0,009	0,017
50.000	80,508	16,680	20,770	0,092	0,062	0,117
100.000	212,886	74,352	83,251	0,198	0,123	0,226
250.000	1218,498	440,344	471,924	0,384	0,273	0,537
500.000	3913,341	1594,737	1710,482	0,637	0,410	0,966
1.000.000	16909,678	5382,884	5539,852	1,321	0,851	2,180
1.500.000	27209,241	9852,351	10632,852	1,835	1,321	3,423
2.000.000	36609,241	15352,231	16235,643	2,532	2,224	4,123



Note: Merge Sort, Quick Sort, dan Shell Sort berada pada rentang waktu yang sama 0-4 detik, sehingga mereka hanya terlihat 1 garis pada grafik tersebut

5. Kesimpulan

Berdasarkan hasil eksperimen dan analisis performa terhadap enam algoritma sorting yang diuji (Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort, dan Shell Sort) pada dua tipe data (angka dan kata), dapat disimpulkan hal-hal berikut:

1. Algoritma dengan Kompleksitas $O(n^2)$ seperti Bubble Sort, Selection Sort, dan Insertion Sort menunjukkan performa yang sangat lambat dan tidak efisien ketika diterapkan pada dataset berukuran besar (≥ 100.000 elemen). Waktu eksekusi meningkat drastis secara eksponensial seiring bertambahnya jumlah data.
2. Merge Sort dan Quick Sort secara konsisten memberikan waktu eksekusi tercepat pada seluruh ukuran data, baik angka maupun kata. Hal ini menunjukkan efisiensi dari pendekatan Divide and Conquer yang mereka gunakan.
3. Shell Sort memberikan performa yang cukup baik dan stabil, dengan waktu yang berada di antara algoritma $O(n \log n)$ dan $O(n^2)$, sehingga bisa menjadi alternatif yang layak untuk implementasi praktis yang sederhana.
4. Penggunaan memori pada semua algoritma relatif stabil karena data disimpan dalam array tetap. Namun, algoritma seperti Merge Sort menggunakan tambahan array sementara dalam prosesnya, meskipun tidak menyebabkan lonjakan signifikan dalam penggunaan memori.
5. Pada tipe data kata (string), pola performa algoritma serupa dengan tipe angka. Namun, waktu eksekusi cenderung lebih tinggi dibandingkan dengan angka karena proses perbandingan string lebih kompleks daripada angka.
6. Untuk dataset berskala besar (hingga 2 juta elemen), hanya algoritma dengan kompleksitas $O(n \log n)$ (Merge Sort, Quick Sort, dan Shell Sort) yang dapat menyelesaikan sorting dalam waktu sangat singkat (<5 detik). Sementara algoritma $O(n^2)$ dapat memakan waktu berjam-jam, dan tidak layak digunakan.