

LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN LANJUT

PERTEMUAN KE 14

SORTING



Disusun Oleh

Nama: Erlindawati Nduru

Nim : 233210007

Prodi : Sistem Informasi Akuntansi

UNIVERSITAS TEKNOLOGI DAN DIGITAL INDONESIA

YOGYAKARTA

2024/2025

A.TUJUAN

Mahasiswa dapat mengurutkan data dengan metode bubble sort, selection sort dan mengimplementasikannya dalam program.

B.DASAR TEORI

SORTING (PENGURUTAN)

Dalam banyak aplikasi, pengurutan menjadi algoritma yang sering banyak digunakan. Kalau kita berhubungan dengan data yang jumlahnya besar, maka data tersebut akan mudah kita kelola kalau dalam keadaan terurut dengan suatu kunci pengurutan tertentu. Dengan data yang sudah terurut, kita akan dengan mudah mencari data, mengelompokkan data, dan lain-lain. Ada banyak metode pengurutan. Beberapa diantaranya adalah sebagai berikut.

A.BUBBLE SORT

Metode Sorting Bubble sort

adalah pengurutan dengan membandingkan elemen berikutnya jika elemen sekarang lebih besar dari elemen berikutnya maka elemen tersebut ditukar Contoh :

Data : 22, 10, 15, 3, 8, 2

Langkah 1:

22	10	15	3	8	2
22	10	15	3	2	8
22	10	15	2	3	8
22	10	2	15	3	8
22	2	10	15	3	8
2	22	10	15	3	8

Langkah 2:

2	22	10	15	3	8
2	22	10	15	3	8
2	22	10	3	15	8
2	22	3	10	15	8
2	3	22	10	15	8

Langkah 3:

2	3	22	10	15	8
2	3	22	10	8	15
2	3	22	8	10	15
2	3	8	22	10	15

Langkah 4:

2	3	8	22	10	15
2	3	8	10	22	15
2	3	8	10	22	15

Langkah 5:

2	3	8	10	22	15
2	3	8	10	15	22

Hasil pengurutannya adalah : 2, 3, 8, 10, 15, 22

b. SELECTION SORT

Metode pengurutan Selection Sort adalah

Pengurutan dengan mencari dari elemen yang berikutnya sampai dengan elemen terakhir jika ditemukan elemen lain yang lebih kecil dari elemen sekarang maka elemen yang bersangkutan akan ditukar. Oleh karena itu setiap langkah selalu memilih satu baris dari baris berikutnya.

Contoh :

Data : 22, 10, 15, 3, 8, 2

Penyelesaian :

Langkah 1 :

22, 10, 15, 3, 8, 2

Pembandingan || Posisi data terkecil

22 > 10 2

10 < 15 2

10 > 3 4

3 < 8 4

3 > 2 6

Posisi ke1=22, ditukar posisi ke-6

Hasil : 2, 10, 15, 3, 8, 22

Langkah 2:

2,10,15,3,8,22

Pembandingan || Posisi data terkecil

10 < 15 2

10 > 3 4

3 < 8 4

3 < 22 4

Posisi ke2=10, ditukar posisi ke-4

Hasil : 2, 3, 15, 10, 8, 22

Langkah 3:

2, 3, 15, 10, 8, 22

Pembandingan || posisi data terkecil

15 > 10 4

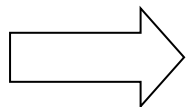
10 > 8 5

8 < 22 5

keterangan :

posisi ke-3 =15, tukar posisi ke-4

hasil : 2, 3, 8, 10, 15, 22



Langkah 4:

2, 3, 8, 10, 15, 22

Pembandingan || posisi data terkecil

10 < 15 5

10 < 22 5

keterangan :

tidak ada pertukaran data

hasil : 2, 3, 8, 10, 15, 22



Langkah 5:

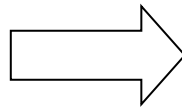
2, 3, 8, 10, 15, 22

Pembandingan || posisi data terkecil

keterangan :

15 < 22

5



tidak ada pertukaran data

hasil : 2, 3, 8, 10, 15, 22

Hasil pengurutannya : 2, 3, 8, 10, 15, 22

C.PRAKTIK

Praktik 1. Program bubble sort.

```
import java.util.Scanner;
public class BubbleSort{
public void bubbleSort(float larik2[])
{
    for (int i=0;i<larik2.length;i++)
    {
        for (int elemen=0;elemen<larik2.length-1;elemen++)
        {
            if (larik2[elemen]>larik2[elemen+1])
                tukar(larik2, elemen,elemen+1);
        }
    }
}
public void tukar(float larik3[], int satu, int dua)
{
    float temp;
    temp = larik3[satu];
    larik3[satu] = larik3[dua];
    larik3[dua] = temp;
}

public static void main(String args[]){
    Scanner masuk = new Scanner(System.in);
    BubbleSort lrk = new BubbleSort();
    float nilai[]= new float[5];
    System.out.println("Masukan 5 buat data nilai");
    for (int i = 0; i < 5; i++)
    {
        System.out.print( (i + 1 )+" : ");
        nilai[i]=masuk.nextFloat();
    }
    System.out.println("Data nilai yang dimasukan");
```

```

for (int i = 0; i < 5; i++)
    System.out.println(nilai[i]);
System.out.println("Data hasil pengurutan  ");
lrk.bubbleSort(nilai);
for (int i = 0; i < 5; i++)
    System.out.println(nilai[i]);
}
}

```

```

import java.util.Scanner;
public class BubbleSort{
    public void bubbleSort(float larik2[])
    {
        for (int i=0;i<larik2.length;i++)
        {
            for (int elemen=0;elemen<larik2.length-1;elemen++)
            {
                if (larik2[elemen]>larik2[elemen+1])
                    tukar(larik2, elemen,elemen+1);
            }
        }
    }
    public void tukar(float larik3[], int satu, int dua)
    {
        float temp;
        temp = larik3[satu];
        larik3[satu] = larik3[dua];
        larik3[dua] = temp;
    }
    public static void main(String args[]){
        Scanner masuk = new Scanner(System.in);
        BubbleSort lrk = new BubbleSort();
        float nilai[] = new float[5];
        System.out.println("Masukan 5 buat data nilai");
        for (int i = 0; i < 5; i++)
        {
            System.out.print( (i + 1 )+" : ");
            nilai[i]=masuk.nextFloat();
        }
        System.out.println("Data nilai yang dimasukan");
        for (int i = 0; i < 5; i++)
            System.out.println(nilai[i]);
    }
}

```

```

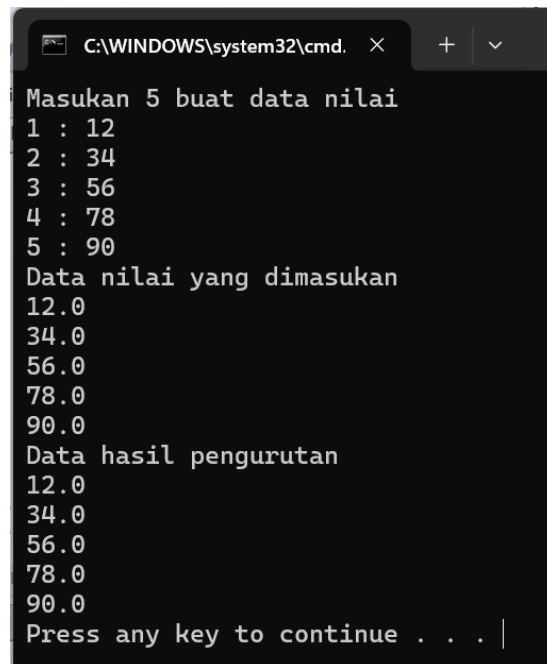
        System.out.println(nilai[i]);
        System.out.println("Data hasil pengurutan  ");
        lrk.bubbleSort(nilai);
        for (int i = 0; i < 5; i++)
            System.out.println(nilai[i]);
    }
}

```

Tool Output

Tool completed successfully

Output:



```
C:\WINDOWS\system32\cmd.  X  +  v

Masukan 5 buat data nilai
1 : 12
2 : 34
3 : 56
4 : 78
5 : 90
Data nilai yang dimasukan
12.0
34.0
56.0
78.0
90.0
Data hasil pengurutan
12.0
34.0
56.0
78.0
90.0
Press any key to continue . . . |
```

Penjelasan:

Program kelas BubbleSort:memiliki dua metode: bubbleSort dan tukar.

Metode bubbleSort:

- Metode menerima sebuah array tipe float dan mengurutkannya menggunakan algoritma bubble sort.Proses pengurutan dilakukan dengan dua loop bersarang. Loop luar mengiterasi seluruh elemen array, sementara loop dalam melakukan perbandingan dan pertukaran elemen jika elemen saat ini lebih besar dari elemen berikutnya.

Metode tukar:

- Metode ini menerima sebuah array dan dua indeks. Metode ini menukar posisi dua elemen dalam array.

Metode main:

- Metode ini adalah titik masuk utama program.Program meminta pengguna untuk memasukkan 5 nilai float.Nilai yang dimasukkan ditampilkan.Array nilai kemudian diurutkan menggunakan metode bubbleSort.Hasil pengurutan ditampilkan.

Praktik 2. Program Selection sort.

```
import java.util.Scanner;
public class SelectionSort{
public void selectionSort(float larik2[])
{
    for (int i=0;i<larik2.length;i++)
    {
        int min =i;
        for (int elemen=i+1;elemen<larik2.length;elemen++)
        {
            if (larik2[min]>larik2[elemen])
                min = elemen;
        }
        tukar(larik2, min,i);
    }
}
public void tukar(float larik3[], int satu, int dua)
{
    float temp;
    temp = larik3[satu];
    larik3[satu] = larik3[dua];
    larik3[dua] = temp;
}
public static void main(String args[]){
    Scanner masuk = new Scanner(System.in);
    SelectionSort lrk = new SelectionSort();
    float nilai[]= new float[5];
    System.out.println("Masukan 5 buat data nilai");
    for (int i = 0; i < 5; i++)
    {
        System.out.print( (i + 1 )+" : ");
        nilai[i]=masuk.nextFloat();
    }
    System.out.println("Data nilai yang dimasukan");
    for (int i = 0; i < 5; i++)
        System.out.println(nilai[i]);
    System.out.println("Data hasil pengurutan   ");
    lrk.selectionSort(nilai);
    for (int i = 0; i < 5; i++)
        System.out.println(nilai[i]);
}
}
```



```

import java.util.Scanner;
public class SelectionSort{
public void selectionSort(float larik2[])
{
    for (int i=0;i<larik2.length;i++)
    {
        int min =i;
        for (int elemen=i+1;elemen<larik2.length;elemen++)
        {
            if (larik2[min]>larik2[elemen])
                min = elemen;
        }
        tukar(larik2, min,i);
    }
}
public void tukar(float larik3[], int satu, int dua)
{
    float temp;
    temp = larik3[satu];
    larik3[satu] = larik3[dua];
    larik3[dua] = temp;
}
public static void main(String args[]){
    Scanner masuk = new Scanner(System.in);
    SelectionSort lrk = new SelectionSort();
    float nilai[]= new float[5];
    System.out.println("Masukan 5 buat data nilai");
    for (int i = 0; i < 5; i++)
    {
        System.out.print( (i + 1) + " : ");
        nilai[i]=masuk.nextFloat();
    }
    System.out.println("Data nilai yang dimasukan");

```

```

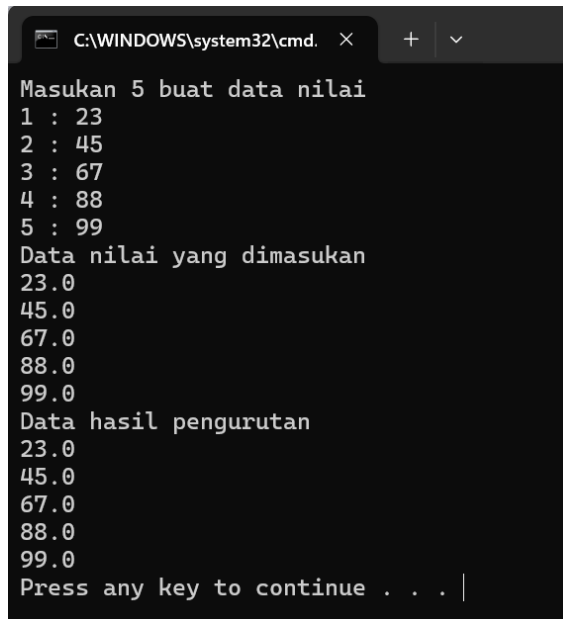
        for (int i = 0; i < 5; i++)
            System.out.println(nilai[i]);
        System.out.println("Data hasil pengurutan ");
        lrk.selectionSort(nilai);
        for (int i = 0; i < 5; i++)
            System.out.println(nilai[i]);
    }
}

```

Tool Output

Tool completed successfully

Output:



```
C:\WINDOWS\system32\cmd.  X  +  v
Masukan 5 buat data nilai
1 : 23
2 : 45
3 : 67
4 : 88
5 : 99
Data nilai yang dimasukan
23.0
45.0
67.0
88.0
99.0
Data hasil pengurutan
23.0
45.0
67.0
88.0
99.0
Press any key to continue . . . |
```

Penjelasan:

Program kelas SelectionSort:Kelas memiliki dua metode: selectionSort dan tukar.

Metode selectionSort:

- Metode menerima sebuah array tipe float dan mengurutkannya menggunakan algoritma selection sort. Proses pengurutan dilakukan dengan dua loop bersarang. Loop luar mengiterasi setiap elemen array. Untuk setiap iterasi, elemen terkecil dalam subarray yang tidak diurutkan ditemukan dengan loop dalam. Setelah elemen terkecil ditemukan, metode tukar dipanggil untuk menukar elemen terkecil dengan elemen pertama dari subarray yang tidak diurutkan.

Metode tukar:

- Metode ini menerima sebuah array dan dua indeks. Metode ini menukar posisi dua elemen dalam array.

Metode main:

- Metode ini adalah titik masuk utama program.
- Program meminta pengguna untuk memasukkan 5 nilai float.
- Nilai yang dimasukkan ditampilkan sebelum diurutkan.

- Array nilai kemudian diurutkan menggunakan metode selectionSort.
- Hasil pengurutan ditampilkan.

Praktik tambahan

Modifikasi program 1

```
import java.util.Scanner;
public class PrakTambahan {
    public void bubbleSort(float larik2[]) {
        for (int i = 0; i < larik2.length; i++) {
            for (int elemen = 0; elemen < larik2.length - 1; elemen++) {
                if (larik2[elemen] > larik2[elemen + 1]) {
                    System.out.println("Tukar : " + larik2[elemen] + " dan " + larik2[elemen + 1]);
                    tukar(larik2, elemen, elemen + 1);
                    printArray(larik2);
                }
            }
        }
    }
    public void tukar(float larik3[], int satu, int dua) {
        float temp;
        temp = larik3[satu];
        larik3[satu] = larik3[dua];
        larik3[dua] = temp;
    }
    public void printArray(float[] array) {
        for (float value : array) {
            System.out.print(value + " ");
        }
        System.out.println();
    }
    public static void main(String args[]) {
        Scanner masuk = new Scanner(System.in);
        PrakTambahan lrk = new PrakTambahan();
        float nilai[] = new float[5];
        System.out.println("Masukan 5 buat data nilai");
        for (int i = 0; i < 5; i++) {
            System.out.print((i + 1) + " : ");
            nilai[i] = masuk.nextFloat();
        }
        System.out.println("Data nilai yang dimasukan");
        for (int i = 0; i < 5; i++)
            System.out.println(nilai[i]);
        System.out.println("Data hasil pengurutan  ");
        lrk.bubbleSort(nilai);
        for (int i = 0; i < 5; i++)
            System.out.println(nilai[i]);
    }
}
```

```
}  
}
```

```
import java.util.Scanner;  
public class PrakTambahan {  
    public void bubbleSort(float larik2[]) {  
        for (int i = 0; i < larik2.length; i++) {  
            for (int elemen = 0; elemen < larik2.length - 1; elemen++) {  
                if (larik2[elemen] > larik2[elemen + 1]) {  
                    System.out.println("Tukar : " + larik2[elemen] + " dan " + larik2[elemen + 1]);  
                    tukar(larik2, elemen, elemen + 1);  
                    printArray(larik2);  
                }  
            }  
        }  
    }  
    public void tukar(float larik3[], int satu, int dua) {  
        float temp;  
        temp = larik3[satu];  
        larik3[satu] = larik3[dua];  
        larik3[dua] = temp;  
    }  
    public void printArray(float[] array) {  
        for (float value : array) {  
            System.out.print(value + " ");  
        }  
        System.out.println();  
    }  
    public static void main(String args[]) {  
        Scanner masuk = new Scanner(System.in);  
        PrakTambahan lrk = new PrakTambahan();  
        float nilai[] = new float[5];  
        System.out.println("Masukan 5 buat data nilai");  
        for (int i = 0; i < 5; i++) {  
            System.out.print((i + 1) + " : ");  
            nilai[i] = masuk.nextFloat();  
            nilai[i+1] = masuk.nextFloat();  
        }  
        System.out.println("Data nilai yang dimasukan");  
        for (int i = 0; i < 5; i++)  
            System.out.println(nilai[i]);  
        System.out.println("Data hasil pengurutan");  
        lrk.bubbleSort(nilai);  
        for (int i = 0; i < 5; i++)  
            System.out.println(nilai[i]);  
    }  
}
```

Tool Output

Tool completed successfully

Output:

```
C:\WINDOWS\system32\cmd. X + v
3 : 7
4 : 9
5 : 1
Data nilai yang dimasukan
2.0
5.0
7.0
9.0
1.0
Data hasil pengurutan
Tukar : 9.0 dan 1.0
2.0 5.0 7.0 1.0 9.0
Tukar : 7.0 dan 1.0
2.0 5.0 1.0 7.0 9.0
Tukar : 5.0 dan 1.0
2.0 1.0 5.0 7.0 9.0
Tukar : 2.0 dan 1.0
1.0 2.0 5.0 7.0 9.0
1.0
2.0
5.0
7.0
9.0
Press any key to continue . . . |
```

Penjelasan:

Program kelas PrakTambahan:Kelas memiliki tiga metode: bubbleSort, tukar, dan printArray.

Metode bubbleSort:

- Metode ini menerima sebuah array tipe float dan mengurutkannya menggunakan algoritma bubble sort. Proses pengurutan dilakukan dengan dua loop bersarang. Loop luar mengiterasi seluruh elemen array, sementara loop dalam melakukan perbandingan dan pertukaran elemen jika elemen saat ini lebih besar dari elemen berikutnya. Setiap kali ada pertukaran elemen, pesan "Tukar: [nilai elemen] dan [nilai elemen berikutnya]" dicetak ke layar, dan metode printArray dipanggil untuk menampilkan keadaan array saat ini.

Metode tukar:

- Metode ini menerima sebuah array dan dua indeks. Metode ini menukar posisi dua elemen dalam array.

Metode printArray:

- Metode mencetak semua elemen dalam array ke layar dengan format yang mudah dibaca.

Metode main:

- Metode ini adalah titik masuk utama program.
- Program meminta pengguna untuk memasukkan 5 nilai float.
- Nilai yang dimasukkan ditampilkan sebelum diurutkan.
- Array nilai kemudian diurutkan menggunakan metode bubbleSort, dengan menampilkan setiap langkah pertukaran. Hasil pengurutan akhir ditampilkan.

Modifikasi program 2

```
import java.util.Scanner;

public class prakTambahan2 {

    public void selectionSort(float larik2[]) {
        for (int i = 0; i < larik2.length; i++) {
            int min = i;
            for (int elemen = i + 1; elemen < larik2.length; elemen++) {
                if (larik2[min] > larik2[elemen])
                    min = elemen;
            }
            tukar(larik2, min, i);
            System.out.println("Setelah Di Tukar : ");
            printArray(larik2);
        }
    }

    public void tukar(float larik3[], int satu, int dua) {
        float temp;
        temp = larik3[satu];
        larik3[satu] = larik3[dua];
        larik3[dua] = temp;
    }

    public void printArray(float[] array) {
        for (float value : array) {
            System.out.print(value + " ");
        }
        System.out.println();
    }

    public static void main(String args[]) {
        Scanner masuk = new Scanner(System.in);
        prakTambahan2 lrk = new prakTambahan2();
        float nilai[] = new float[5];
        System.out.println("Masukan 5 buat data nilai");
```

```

for (int i = 0; i < 5; i++) {
    System.out.print((i + 1) + " : ");
    nilai[i] = masuk.nextFloat();
}
System.out.println("Data nilai yang dimasukan");
lrk.printArray(nilai);
System.out.println("Data hasil pengurutan  ");
lrk.selectionSort(nilai);
lrk.printArray(nilai);
}
}

```

```

import java.util.Scanner;
public class prakTambahan2 {
    public void selectionSort(float larik2[]) {
        for (int i = 0; i < larik2.length; i++) {
            int min = i;
            for (int elemen = i + 1; elemen < larik2.length; elemen++) {
                if (larik2[min] > larik2[elemen])
                    min = elemen;
            }
            tukar(larik2, min, i);
            System.out.println("Setelah Di Tukar : ");
            printArray(larik2);
        }
    }
    public void tukar(float larik3[], int satu, int dua) {
        float temp;
        temp = larik3[satu];
        larik3[satu] = larik3[dua];
        larik3[dua] = temp;
    }
    public void printArray(float[] array) {
        for (float value : array) {
            System.out.print(value + " ");
        }
        System.out.println();
    }
    public static void main(String args[]) {
        Scanner masuk = new Scanner(System.in);
        prakTambahan2 lrk = new prakTambahan2();
        float nilai[] = new float[5];
        System.out.println("Masukan 5 buat data nilai");
        for (int i = 0; i < 5; i++) {
            System.out.print((i + 1) + " : ");

```

```

            System.out.print((i + 1) + " : ");
            nilai[i] = masuk.nextFloat();
        }
        System.out.println("Data nilai yang dimasukan");
        lrk.printArray(nilai);
        System.out.println("Data hasil pengurutan  ");
        lrk.selectionSort(nilai);
        lrk.printArray(nilai);
    }
}

```

Tool Output

Tool completed successfully

Output:

```
Masukan 5 buat data nilai
1 : 1
2 : 45
3 : 78
4 : 90
5 : 56
Data nilai yang dimasukan
1.0 45.0 78.0 90.0 56.0
Data hasil pengurutan
Setelah Di Tukar :
1.0 45.0 78.0 90.0 56.0
Setelah Di Tukar :
1.0 45.0 78.0 90.0 56.0
Setelah Di Tukar :
1.0 45.0 56.0 90.0 78.0
Setelah Di Tukar :
1.0 45.0 56.0 78.0 90.0
Setelah Di Tukar :
1.0 45.0 56.0 78.0 90.0
1.0 45.0 56.0 78.0 90.0
Press any key to continue . . .
```

Penjelasan:

Kelas prakTambahan2:Kelas memiliki tiga metode: selectionSort, tukar, dan printArray.

Metode selectionSort:

- Metode menerima sebuah array tipe float dan mengurutkannya menggunakan algoritma selection sort. Proses pengurutan dilakukan dengan dua loop bersarang. Loop luar mengiterasi setiap elemen array. Untuk setiap iterasi, elemen terkecil dalam subarray yang tidak diurutkan ditemukan dengan loop dalam. Setelah elemen terkecil ditemukan, metode tukar dipanggil untuk menukar elemen terkecil dengan elemen pertama dari subarray yang tidak diurutkan. Setelah setiap pertukaran, array saat ini ditampilkan dengan memanggil metode printArray dan mencetak pesan "Setelah Di Tukar:".

Metode tukar:

- Metode ini menerima sebuah array dan dua indeks. Metode ini menukar posisi dua elemen dalam array.

Metode printArray:

- Metode ini mencetak semua elemen dalam array ke layar dengan format yang mudah dibaca.

Metode main:

- Metode ini adalah titik masuk utama program.
- Program meminta pengguna untuk memasukkan 5 nilai float.
- Nilai yang dimasukkan ditampilkan sebelum diurutkan.
- Array nilai kemudian diurutkan menggunakan metode selectionSort, dengan menampilkan setiap langkah pertukaran.
- Hasil pengurutan akhir ditampilkan.

D.LATIHAN

Kita bisa melakukan sorting dengan menggunakan kelas Collection. Jalankan program berikut, dan jelaskan.

1.Sorting ascending

```
// Collections method sort.
import java.util.List;
import java.util.Arrays;
import java.util.Collections;
public class Sort1
{
    public static void main( String[] args )
    {
        String[] suits = { "Hearts", "Diamonds", "Clubs", "Spades" };

        // Create and display a list containing the suits array elements
        List< String > list = Arrays.asList( suits ); // create List
        System.out.printf( "Unsorted array elements: %s\n", list );

        Collections.sort( list ); // sort ArrayList

        // output list
        System.out.printf( "Sorted array elements: %s\n", list );
    } // end main
} // end class Sort1
```

```
// Collections method sort.
import java.util.List;
import java.util.Arrays;
import java.util.Collections;
public class Sort1
{
    public static void main( String[] args )
    {
        String[] suits = { "Hearts", "Diamonds", "Clubs", "Spades" };

        // Create and display a list containing the suits array elements
        List< String > list = Arrays.asList( suits ); // create List
        System.out.printf( "Unsorted array elements: %s\n", list );

        Collections.sort( list ); // sort ArrayList

        // output list
        System.out.printf( "Sorted array elements: %s\n", list );
    } // end main
} // end class Sort1
```

Tool Output

Tool completed successfully

Output:

```
C:\WINDOWS\system32\cmd. x + v
Unsorted array elements: [Hearts, Diamonds, Clubs, Spades]
Sorted array elements: [Clubs, Diamonds, Hearts, Spades]
Press any key to continue . . . |
```

Penjelasan:

Program Sort1 menggunakan metode Collections.sort untuk mengurutkan elemen-elemen dalam sebuah daftar (list).

Inisialisasi Array:

- Program dimulai dengan mendeklarasikan dan menginisialisasi sebuah array suits yang berisi empat string: "Hearts", "Diamonds", "Clubs", dan "Spades".

Konversi Array ke List:

- Array suits kemudian dikonversi menjadi sebuah daftar (list) menggunakan metode Arrays.asList. Metode mengembalikan daftar yang didukung oleh array yang telah dikonversi. Daftar disimpan dalam variabel list.

Menampilkan Elemen Array yang Belum Diurutkan:

- Program menampilkan elemen-elemen dalam daftar yang belum diurutkan menggunakan System.out.printf dan menampilkan hasilnya dengan format Unsorted array elements: [Hearts, Diamonds, Clubs, Spades].

Mengurutkan Daftar:

- Metode Collections.sort digunakan untuk mengurutkan elemen-elemen dalam daftar list dalam urutan alfanumerik.

Menampilkan Elemen Array yang Sudah Diurutkan:

- Program kemudian menampilkan elemen-elemen dalam daftar yang sudah diurutkan menggunakan System.out.printf dan menampilkan hasilnya dengan format Sorted array elements: [Clubs, Diamonds, Hearts, Spades].

2.Sorting descending

```
// Using a Comparator object with method sort.
import java.util.List;
import java.util.Arrays;
import java.util.Collections;

public class Sort2
{
    public static void main( String[] args )
    {
        String[] suits = { "Hearts", "Diamonds", "Clubs", "Spades" };

        // Create and display a list containing the suits array elements
        List< String > list = Arrays.asList( suits ); // create List
        System.out.printf( "Unsorted array elements: %s\n", list );

        // sort in descending order using a comparator
        Collections.sort( list, Collections.reverseOrder() );

        // output List elements
        System.out.printf( "Sorted list elements: %s\n", list );
    } // end main
} // end class Sort2
```

```
// Using a Comparator object with method sort.
import java.util.List;
import java.util.Arrays;
import java.util.Collections;

public class Sort2
{
    public static void main( String[] args )
    {
        String[] suits = { "Hearts", "Diamonds", "Clubs", "Spades" };

        // Create and display a list containing the suits array elements
        List< String > list = Arrays.asList( suits ); // create List
        System.out.printf( "Unsorted array elements: %s\n", list );

        // sort in descending order using a comparator
        Collections.sort( list, Collections.reverseOrder() );

        // output List elements
        System.out.printf( "Sorted list elements: %s\n", list );
    } // end main
} // end class Sort2
```

Tool Output

Tool completed successfully

Output:

```
C:\WINDOWS\system32\cmd.  ×  +  v
Unsorted array elements: [Hearts, Diamonds, Clubs, Spades]
Sorted list elements: [Spades, Hearts, Diamonds, Clubs]
Press any key to continue . . . |
```

Penjelasan:

Program Sort2 menggunakan objek Comparator bersama dengan metode Collections.sort untuk mengurutkan elemen-elemen dalam sebuah daftar (list) dalam urutan menurun (descending order).

Inisialisasi Array:

- Program dimulai dengan mendeklarasikan dan menginisialisasi sebuah array suits yang berisi empat string: "Hearts", "Diamonds", "Clubs", dan "Spades".

Konversi Array ke List:

- Array suits kemudian dikonversi menjadi sebuah daftar (list) menggunakan metode `Arrays.asList`. Metode ini mengembalikan daftar yang didukung oleh array yang telah dikonversi. Daftar disimpan dalam variabel list.

Menampilkan Elemen Array yang Belum Diurutkan:

- Program menampilkan elemen-elemen dalam daftar yang belum diurutkan menggunakan `System.out.printf` dan menampilkan hasilnya dengan format `Unsorted array elements: [Hearts, Diamonds, Clubs, Spades]`.

Mengurutkan Daftar dalam Urutan Menurun:

- Metode `Collections.sort` digunakan untuk mengurutkan elemen-elemen dalam daftar list dalam urutan menurun. Untuk melakukan ini, metode `Collections.reverseOrder` digunakan untuk mendapatkan objek `Comparator` yang mengatur urutan pengurutan menjadi menurun.

Menampilkan Elemen Array yang Sudah Diurutkan:

- Program kemudian menampilkan elemen-elemen dalam daftar yang sudah diurutkan menggunakan `System.out.printf` dan menampilkan hasilnya dengan format `Sorted list elements: [Spades, Hearts, Diamonds, Clubs]`.

E.TUGAS

1.Jelaskan tentang metode sorting Insertion sort, Merge sort dan Quick sort.

1.Insertion Sort

Insertion sort adalah algoritma sorting yang sederhana dan efektif untuk mengurutkan array dengan ukuran kecil. Cara kerja insertion sort adalah dengan membandingkan setiap elemen dalam array dengan elemen sebelumnya dan memindahkan elemen yang lebih besar ke posisi yang tepat. Proses ini dilakukan secara iteratif, dimulai dari elemen pertama, hingga semua elemen dalam array telah diurutkan.

Kelebihan insertion sort adalah:

Stable: Insertion sort mempertahankan urutan elemen yang memiliki nilai sama.

Effektif untuk array kecil: Insertion sort lebih cepat dan lebih efisien untuk mengurutkan array dengan ukuran kecil.

Kekurangan insertion sort adalah:

Worst-case performance: Insertion sort memiliki performa yang buruk jika array yang akan diurutkan dalam urutan terbalik, dengan waktu kompleksitas $O(n^2)$.

2.Merge Sort

Merge sort adalah algoritma sorting yang berbasis rekursif dan memiliki waktu kompleksitas $O(n \log n)$. Cara kerja merge sort adalah dengan membagi array menjadi dua bagian yang lebih kecil, kemudian mengurutkan masing-masing bagian, dan akhirnya menggabungkan bagian-bagian yang telah diurutkan menjadi satu array yang terurut.

Kelebihan merge sort adalah:

Waktu kompleksitas: Merge sort memiliki waktu kompleksitas $O(n \log n)$, yang lebih baik daripada insertion sort.

Stable: Merge sort juga mempertahankan urutan elemen yang memiliki nilai sama.

Kekurangan merge sort adalah:

Memerlukan ruang tambahan: Merge sort memerlukan ruang tambahan untuk menyimpan array yang akan diurutkan, sehingga dapat menjadi tidak efisien jika ruang yang tersedia terbatas.

3.Quick Sort

Quick sort adalah algoritma sorting yang berbasis rekursif dan memiliki waktu kompleksitas rata-rata $O(n \log n)$. Cara kerja quick sort adalah dengan memilih elemen acak sebagai "pivot" dan kemudian membagi array menjadi dua bagian: elemen yang lebih kecil dari pivot dan elemen yang lebih besar dari pivot. Proses ini dilakukan secara rekursif untuk bagian-bagian yang lebih kecil.

Kelebihan quick sort adalah:

Waktu kompleksitas: Quick sort memiliki waktu kompleksitas rata-rata $O(n \log n)$, yang lebih baik daripada insertion sort.

Efisien untuk array besar: Quick sort lebih efisien untuk mengurutkan array dengan ukuran besar.

Kekurangan quick sort adalah:

Worst-case performance: Quick sort memiliki performa yang buruk jika array yang akan diurutkan dalam urutan terbalik, dengan waktu kompleksitas $O(n^2)$.

Tidak stabil: Quick sort tidak mempertahankan urutan elemen yang memiliki nilai sama.

KESIMPULAN

Sorting adalah proses mengatur elemen-elemen dalam suatu struktur data agar diurutkan berdasarkan suatu urutan tertentu (misalnya, urutan naik atau turun). Pengurutan data mempermudah pencarian, penggabungan, dan analisis data, serta meningkatkan efisiensi berbagai algoritma lainnya.