



ALGORITMA PEMROGRAMAN

ALGORITMA SORTING (SELECTION SORT & BUBBLE SORT)

TIM PENYUSUN: - DOSEN
- MOCH. DAFFA SHAFWAN CHAIRULLAH

PRESENTED BY: LAB. TEKNIK INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

ALGORITMA PEMROGRAMAN

CAPAIAN PEMBELAJARAN MATA KULIAH

1. Mahasiswa dapat menjelaskan dan menerapkan beberapa Algoritma Sorting dalam menyelesaikan masalah.
2. Mahasiswa mampu menganalisa kompleksitas dari beberapa Algoritma Sorting.

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

1. Mahasiswa mampu menjelaskan dan menganalisa Selection Sort dan Bubble Sort.
2. Mahasiswa mampu mengidentifikasi perbedaan Selection Sort dan Bubble Sort.
3. Mahasiswa mampu membuat program dari algoritma dan flowchart dari Algoritma Selection Sort dan Bubble Sort menggunakan bahasa pemrograman Java.
4. Mahasiswa mampu menyelesaikan permasalahan dengan algoritma sorting.

KEBUTUHAN HARDWARE & SOFTWARE

- Laptop/ PC
- Eclipse/ Netbeans/ dsb.

MATERI PRAKTIKUM

1. Selection Sort

Merupakan metode yang mengkombinasikan antara sorting dan searching. Untuk setiap proses, akan dicari elemen-elemen yang belum diurutkan yang memiliki nilai terkecil atau terbesar akan ditukarkan ke posisi yang tepat di dalam array. Misalnya untuk putaran pertama, akan dicari dengan nilai terkecil dan data ini akan ditempatkan di indeks terkecil (data [0]), pada putaran kedua akan dicari data kedua terkecil, dan akan ditempatkan di indeks kedua (data [1]). Selama proses, perbandingan dan pengubahan hanya dilakukan pada indeks perbandingan saja, pertukaran data secara fisik terjadi pada akhir proses.

Untuk materi mengenai visualisasi proses algoritma selection sort dapat dipelajari pada video berikut : <https://www.youtube.com/watch?v=Ns4TPTC8whw> . Kita juga dapat membuat visualisasi sendiri proses pengurutan algoritma selection sort pada link berikut : <https://visualgo.net/en/sorting>

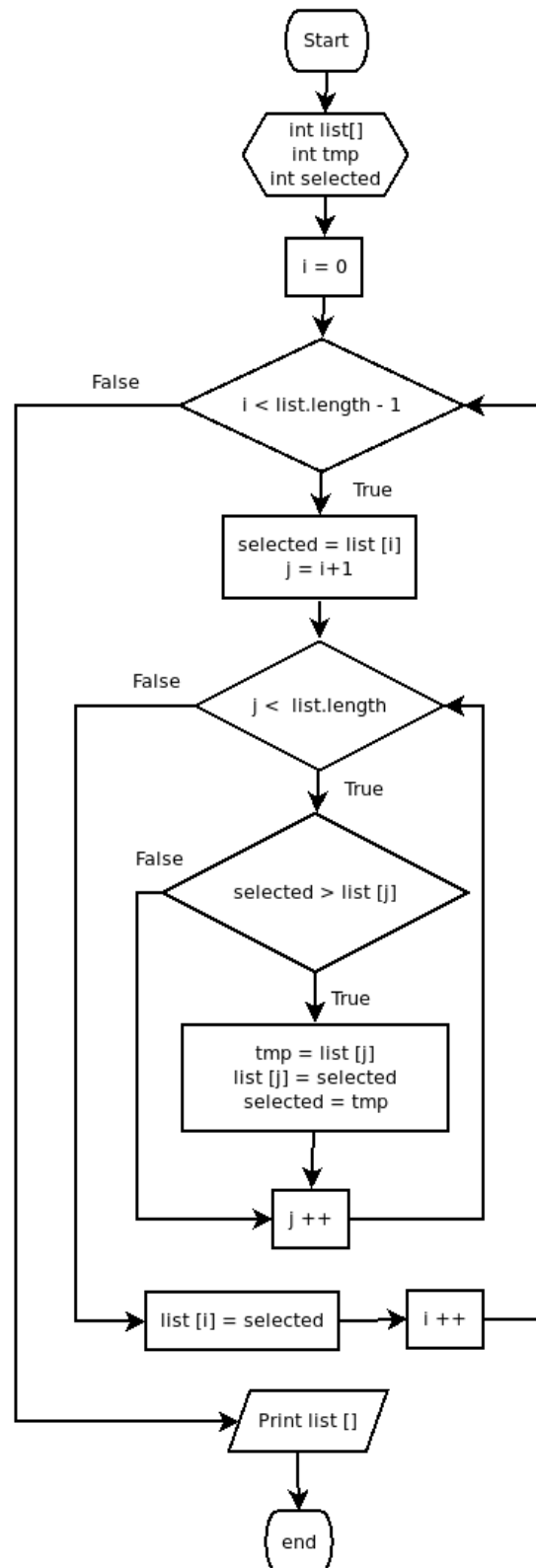
Algoritma :

- **Step 1** – Atur array indeks 0 (elemen pertama) sebagai lokasi/nilai minimum
- **Step 2** – Cari elemen paling kecil yang ada di dalam list
- **Step 3** – Tukar elemen terkecil tersebut ke lokasi/nilai minimum
- **Step 4** – Atur elemen selanjutnya (sebelah kanannya) sebagai lokasi/nilai minimum
- **Step 5** – Ulangi sampai list elemen-elemen kita berhasil terurut semua

Contoh :

5	7	2	$\frac{1}{5}$	3		Terdapat array yang belum terurut.
5	7	2	$\frac{1}{5}$	3	Step 1	Mencari nilai terkecil dari urutan dengan membandingkan satu persatu tiap elemen, dan ditemukan nilai terkecil yaitu 2 .
2	7	5	$\frac{1}{5}$	3	Step 2	Swap nilai terkecil yang ditemukan (2) ke indeks [0].
2	7	5	$\frac{1}{5}$	3	Step 3	Mencari nilai terkecil setelah 2 , dan ditemukan nilai terkecil yaitu 3 .
2	3	5	$\frac{1}{5}$	7	Step 4	Swap nilai terkecil yang ditemukan (3) ke indeks [1].
2	3	5	$\frac{1}{5}$	7	Step 5	Mencari nilai terkecil setelah 3 , dan ditemukan nilai terkecil yaitu 5 . Tidak ada proses swiipe, karena nilai 5 sudah berada di indeks [2].
2	3	5	$\frac{1}{5}$	7	Step 6	Mencari nilai terkecil setelah 5 , dan ditemukan nilai terkecil yaitu 7 .
2	3	5	7	$\frac{1}{5}$	Step 7	Swap nilai terkecil yang ditemukan (7) ke indeks [3].
2	3	5	7	$\frac{1}{5}$		Array sudah terurut dari nilai terkecil hingga terbesar.

Flowchart :



Program :

```

public class SelectionSort {
    ➤ public static void main(String[] args) {
        int[] arr = {5, 7, 2, 15, 3};
        int indexMin = 0; //indeks untuk nilai terkecil

        for(int i = 0; i < arr.length; i++) {
            indexMin = i;
            for(int j = i + 1; j < arr.length; j++) {

                //mengatur sebagai nilai terkecil ketika menemukan nilai yang lebih kecil
                if(arr[j] < arr[indexMin]) {
                    indexMin = j;
                }
            }
            //proses swipe nilai setelah mendapat indeks dari nilai terkecil
            int temp = arr[i]; //variabel temp untuk menyimpan nilai sementara
            arr[i] = arr[indexMin];
            arr[indexMin] = temp;
        }
        for(int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
    }
}

```

2. Bubble Sort

Merupakan metode sorting termudah, diberi nama “Bubble” karena proses pengurutan secara berangsur-angsur bergerak / berpindah ke posisinya yang tepat, seperti gelembung yang keluar dari sebuah gelas bersoda. Bubble sort mengurutkan data dengan cara membandingkan elemen sekarang dengan elemen berikutnya. Jika elemen sekarang lebih besar dari elemen berikutnya maka kedua elemen tersebut ditukar, jika pengurutan ascending. Jika elemen sekarang lebih kecil dari elemen berikutnya, maka kedua elemen tersebut ditukar, jika pengurutan descending. Algoritma ini seolah olah menggeser satu per satu elemen dari kanan ke kiri atau dari kiri ke kanan, tergantung jenis pengurutannya.

Ketika satu proses telah selesai, maka bubble sort akan mengulangi proses, demikian seterusnya. Kapan berhenti? Bubble sort berhenti jika seluruh array telah diperiksa dan tidak ada pertukaran lagi yang bisa dilakukan, serta tercapai perurutan yang telah diinginkan.

Untuk materi mengenai visualisasi proses algoritma bubble sort dapat dipelajari pada video berikut : <http://www.youtube.com/watch?v=lyZQPjUT5B4> . Kita juga dapat membuat visualisasi sendiri proses pengurutan algoritma bubble sort pada link berikut : <https://visualgo.net/en/sorting>

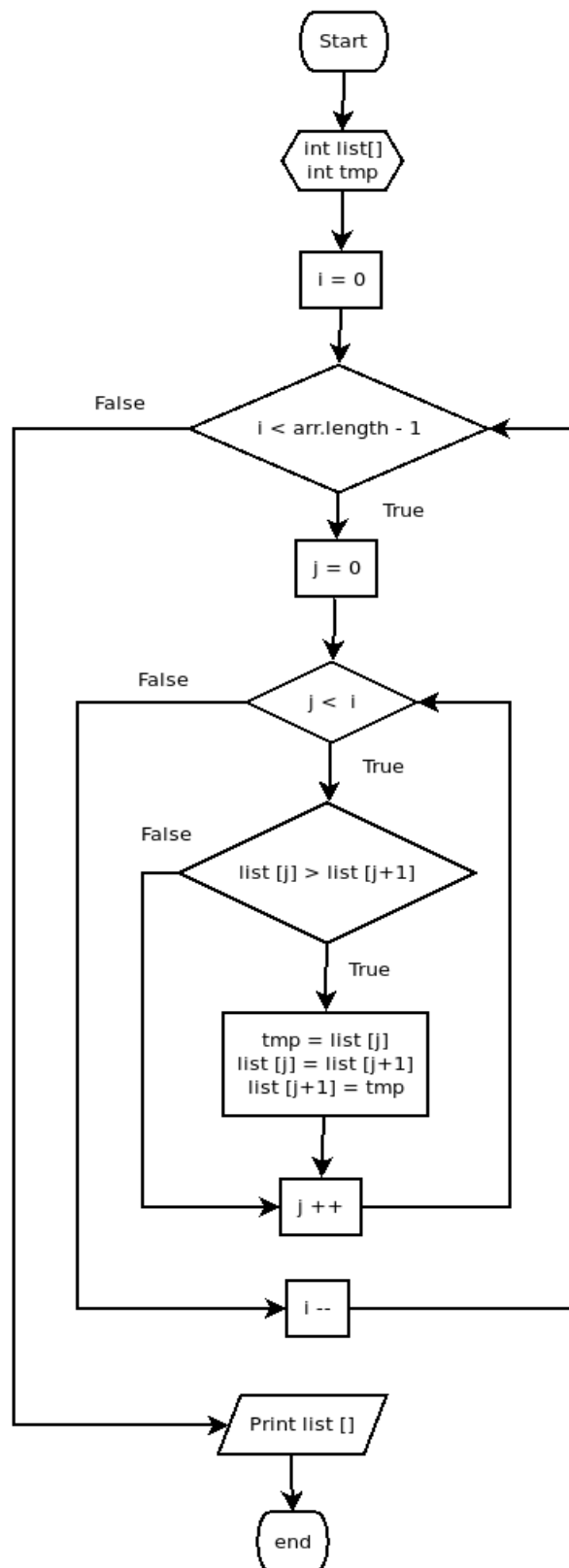
Algoritma :

- **Step 1** - Membandingkan elemen sekarang (indeks [0]) dengan elemen berikutnya (indeks [1])
- **Step 2** - Proses penukaran elemen (ascending / descending)
- **Step 3** - Melanjutkan step 1 dan step 2 sampai proses perbandingan pertama selesai
- **Step 4** - Memulai proses perbandingan selanjutnya dengan mengulang step 1,2, dan 3
- **Step 5** - Melakukan pengecekan ketika sudah tidak ada proses penukaran dalam urutan
- **Step 6** - Selesai

Contoh :

Proses 1	5	7	2	1 5	3	menja di	5	2	7	1 5	3
	5	2	7	1 5	3	manja di	5	2	7	3	1 5
Proses 2	5	2	7	3	1 5	manja di	2	5	7	3	1 5
	2	5	7	3	1 5	manja di	2	5	3	7	1 5
Proses 3	2	5	3	7	1 5	manja di	2	3	5	7	1 5
	2	3	5	7	1 5	teruru t	2	3	5	7	1 5

Flowchart :



Program :

```

1  public class BubbleSort {
2
3  public static void main(String args[]) {
4      int list[] = { 5, 7, 2, 15, 3 };
5
6      for (int i = 0; i < list.length - 1; i++) {
7          for (int j = 0; j < list.length - i - 1; j++) {
8              if (list[j] > list[j + 1]) {
9                  // proses swap list[j+1] dan list[i]
10                 int temp = list[j];
11                 list[j] = list[j + 1];
12                 list[j + 1] = temp;
13             }
14         }
15     }
16     for (int i = 0; i < list.length; ++i)
17         System.out.print(list[i] + " ");
18     System.out.println();
19 }
20 }

```

LEMBAR KERJA

TUGAS PRAKTIKUM

1. Membuat array String dua dimensi yang menyimpan daftar nama mahasiswa dan umurnya yang ada di kelas masing-masing (ada dua kolom, kolom nama dan umur). Urutkan array tersebut sesuai abjad nama dan umur dengan menggunakan algoritma selection sort dan bubble sort. Program harus dibuat secara dinamis (inputan dari user) dan menggunakan OOP.
2. Menganalisis perbedaan kompleksitas waktu dari kedua algoritma tersebut dan dijelaskan pada saat praktikum kepada asisten masing-masing.

RUBRIK PENILAIAN

1. Menjelaskan langkah penerapan algoritma yang telah dipelajari. (30 poin)
2. Menyelesaikan tugas praktikum sesuai dengan ketentuan. (30 poin)
3. Menjelaskan program yang telah dibuat. (20 poin)
4. Menjelaskan hasil analisis perbedaan kompleksitas waktu kedua algoritma. (15 poin)
5. Kerapian program yang dibuat. (5 poin)