

VERSION 1.0
SEPTEMBER 4, 2021



ALGORITMA PEMROGRAMAN

ALGORITMA SORTING (SELECTION SORT & BUBBLE SORT) | MODUL 2

DISUSUN OLEH:

- FARKHAN HAMZAH FIRDAUS
- GILLY HUGA ANARGYA

DIAUDIT OLEH:

- DIDIH RIZKI CHANDRANEGARA, S.KOM., M.KOM
- CHRISTIAN SRI KUSUMA, S.KOM, M.KOM

PRESENTED BY: TIM LAB-IT

UNIVERSITAS MUHAMMADIYAH MALANG

ALGORITMA PEMROGRAMAN

PERSIAPAN MATERI

Algoritma Sorting

Pengurutan (**Sorting**) adalah proses menyusun kembali data yang sebelumnya telah disusun dengan suatu pola tertentu, sehingga tersusun secara teratur menurut aturan tertentu. Pengurutan dapat dilakukan secara ascending (menaik) dan descending (menurun). Contoh:

Data Acak: 8 4 7 1 2 93 3 11

Ascending: 1 2 3 4 7 8 11 93

Descending: 93 11 8 7 4 3 2 1

Terdapat beberapa algoritma yang cukup populer dalam mengurutkan data. Seperti misalnya insertion sort, selection sort, merge sort, heap sort, quick sort, bubble sort, shell sort, comb sort, counting sort, bucket sort, radix sort. Tentunya pengaplikasian algoritma sorting memiliki banyak keuntungannya. Hal yang paling utama dalam penggunaannya yakni untuk mempersingkat dalam penggunaan waktu dalam pembuatan proses program. Berikut merupakan macam-macam algoritma sorting yang akan kita pelajari pada modul ini :

1. Selection Sort

Merupakan metode yang mengkombinasikan antara sorting dan searching. 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>

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 : <https://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>

TUJUAN

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

TARGET MODUL

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.

PERSIAPAN SOFTWARE/APLIKASI

Eclipse/ Netbeans/IntelliJ IDEA/ dsb (IDE Bahasa Java)

KEGIATAN PERCOBAAN

PERCOBAAN 1

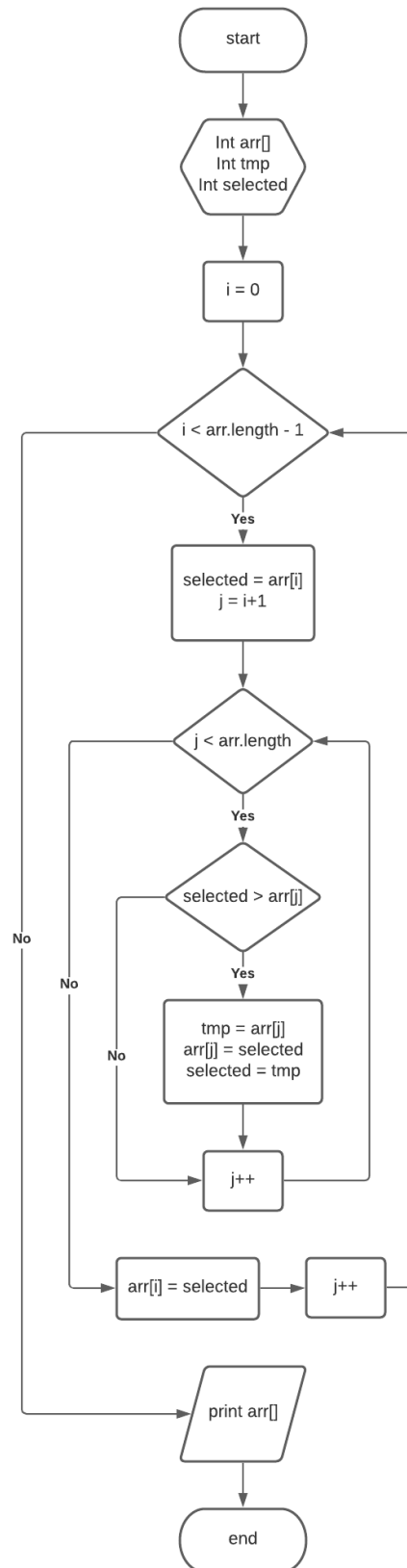
Algoritma Selection Sort :

- **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	9	3		Terdapat array yang belum terurut.
5	7	2	9	3	Step 1	Mencari nilai terkecil dari urutan dengan membandingkan satu persatu tiap elemen, dan ditemukan nilai terkecil yaitu 2.
2	7	5	9	3	Step 2	Swap nilai terkecil yang ditemukan (2) ke indeks [0].
2	7	5	9	3	Step 3	Mencari nilai terkecil setelah 2, dan ditemukan nilai terkecil yaitu 3.
2	3	5	9	7	Step 4	Swap nilai terkecil yang ditemukan (3) ke indeks [1].
2	3	5	9	7	Step 5	Mencari nilai terkecil setelah 3, dan ditemukan nilai terkecil yaitu 5. Tidak ada proses swipe, karena nilai 5 sudah berada di indeks [2].
2	3	5	9	7	Step 6	Mencari nilai terkecil setelah 5, dan ditemukan nilai terkecil yaitu 7.
2	3	5	7	9	Step 7	Swap nilai terkecil yang ditemukan (7) ke indeks [3].
2	3	5	7	9		Array sudah terurut dari nilai terkecil hingga terbesar.

Flowchart :



Program :

```

1 public class SelectionShort{
2
3     public static void main(String[] args){
4         int[] arr = {5, 7, 2, 9, 3};
5         int indexMin = 0; //index untuk nilai terkecil
6
7         for(int i = 0; i < arr.length; i++){
8             indexMin = i;
9             for(int j = i + 1; j < arr.length; j++){
10
11                 //mengatur sebagai nilai terkecil ketika menemukan nilai yang lebih kecil
12                 if(arr[j] < arr[indexMin]){
13                     indexMin = j;
14                 }
15             }
16
17             //proses swipe nilai setelah mendapat indeks dari nilai tekecil
18             int temp = arr[i]; //variabel temp untuk menyimpan nilai sementara
19             arr[i] = arr[indexMin];
20             arr[indexMin] = temp;
21         }
22         for(int i = 0; i < arr.length; i++){
23             System.out.print(arr[i] + " ");
24         }
25     }
26 }

```

PERCOBAAN 2

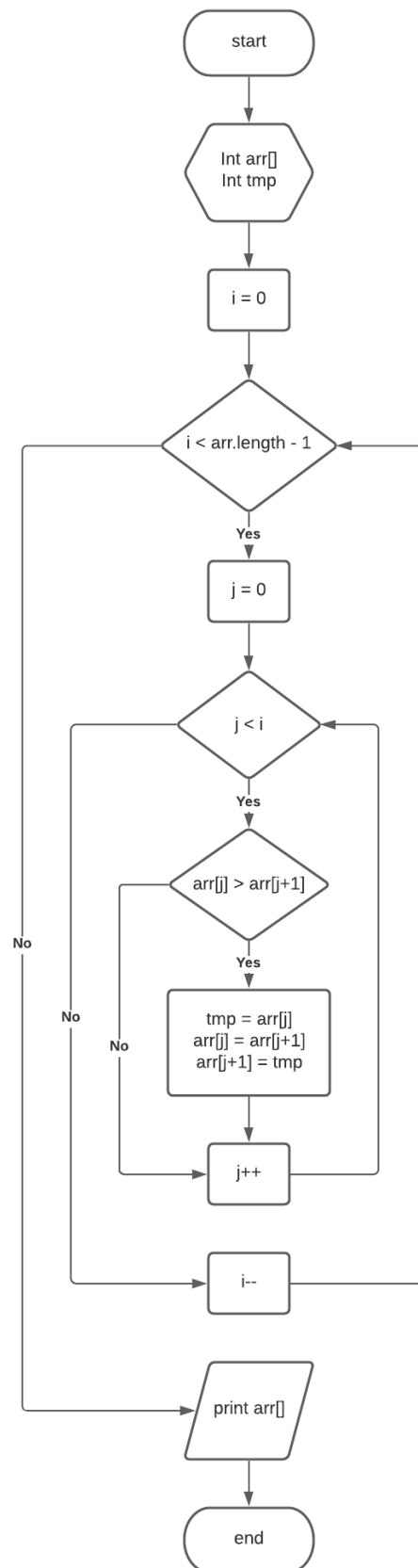
Algoritma Bubble Sort :

- **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	9	3	menjadi	5	2	7	9	3
	5	2	7	9	3	menjadi	5	2	7	3	9
Proses 2	5	2	7	3	9	menjadi	2	5	7	3	9
	2	5	7	3	9	menjadi	2	5	3	7	9
Proses 3	2	5	3	7	9	menjadi	2	3	5	7	9
	2	3	5	7	9	terurut	2	3	5	7	9

Flowchart :



Program :

```

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

```

TUGAS

1. Buatlah 2 buah array String dan masing masing array menyimpan nama orang. Kemudian gabungkan kedua array tersebut menjadi satu array. Urutkan array yang telah digabungkan tersebut sesuai abjad nama dengan menggunakan algoritma selection sort dan bubble sort. Hasil yang diharapkan dari program :

```

Penumpang Mobil A: Lala, Zahra, Dede, Bobo
Penumpang Mobil B: Jarvis, Daniel, Qibo, Aldo

Daftar penumpang: Lala, Zahra, Dede, Bobo, Jarvis, Daniel, Qibo, Aldo

Setelah diurutkan: Aldo, Bobo, Daniel, Dede, Jarvis, Lala, Qibo, Zahra

```

2. Menganalisis perbedaan kompleksitas waktu dari kedua algoritma tersebut dan dijelaskan pada saat praktikum kepada asisten masing-masing.

DETAIL PENILAIAN TUGAS

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