

LAPORAN
Struktur Data Linear
Praktikum 5 : Insertion Sort



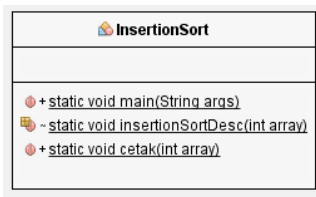
NAMA : Johanes Yogtan WR

NIM : 215314105

Program Studi INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA

1. Insertion Sort

- Diagram UML



- Input (screenshot)

Ascending :

```
package Modul3_Pengurutan;
public class InsertionSort {
    public static void main(String args[]) {
        int array[] = {5, 8, 26, 15, 11, 31};
        System.out.println("DATA SEBELUM DIURUTKAN");
        cetak(array);

        insertionSortAsc(array);
        System.out.println("\nDATA SESUDAH DIURUTKAN");
        cetak(array);
        System.out.println("");
    }

    static void insertionSortAsc(int array[]) {
        int n = array.length;
        for (int j = 1; j < n; j++) {
            int key = array[j];
            int i = j - 1;
            while ((i > -1) && (array[i] > key)) {
                array[i + 1] = array[i];
                i--;
            }
            array[i + 1] = key;
        }
    }

    public static void cetak(int array[]) {
        for (int i = 0; i < array.length; i++) {
            System.out.print(array[i] + " ");
        }
    }
}
```

Descending :

```
package Modul3_Pengurutan;
public class InsertionSort {
    public static void main(String args[]) {
        int array[] = {5, 8, 26, 15, 11, 31};
        System.out.println("DATA SEBELUM DIURUTKAN");
        cetak(array);

        insertionSortDesc(array);
        System.out.println("\nDATA SESUDAH DIURUTKAN");
        cetak(array);
        System.out.println("");
    }

    static void insertionSortDesc(int array[]) {
        int n = array.length;
        for (int j = 1; j < n; j++) {
            int key = array[j];
            int i = j - 1;
            while (i >= 0 && array[i] < key) {
                array[i + 1] = array[i];
                i--;
            }
            array[i + 1] = key;
        }
    }

    public static void cetak(int array[]) {
        for (int i = 0; i < array.length; i++) {
            System.out.print(array[i] + " ");
        }
    }
}
```

- Output (screenshot)
Ascending :

```
run:
DATA SEBELUM DIURUTKAN
5 8 26 15 11 31
DATA SESUDAH DIURUTKAN
5 8 11 15 26 31
BUILD SUCCESSFUL (total time: 0 seconds)
```

Descending :

```
run:
DATA SEBELUM DIURUTKAN
5 8 26 15 11 31
DATA SESUDAH DIURUTKAN
31 26 15 11 8 5
BUILD SUCCESSFUL (total time: 0 seconds)
```

- **Ilustrasi**

Ilustrasi Bubblesort

Iterasi 1

X[0]	Dengan	X[1]	(5 dengan 8)	Tak Berubah
X[1]	Dengan	X[2]	(8 dengan 26)	Tidak Berubah
X[2]	Dengan	X[3]	(26 dengan 15)	Ditukar
X[3]	Dengan	X[4]	(26 dengan 11)	Ditukar
X[4]	Dengan	X[5]	(26 dengan 31)	Tidak Berubah

Iterasi 2

X[0]	Dengan	X[1]	(5 dengan 8)	Tak Berubah
X[1]	Dengan	X[2]	(8 dengan 15)	Tidak Berubah
X[2]	Dengan	X[3]	(15 dengan 11)	Ditukar
X[3]	Dengan	X[4]	(15 dengan 26)	Tidak Berubah
X[4]	Dengan	X[5]	(26 dengan 31)	Tidak Berubah

Iterasi 3 **SAMPAI** Iterasi 6

X[0]	Dengan	X[1]	(5 dengan 8)	Tak Berubah
X[1]	Dengan	X[2]	(8 dengan 11)	Tidak Berubah
X[2]	Dengan	X[3]	(11 dengan 15)	Tidak Berubah
X[3]	Dengan	X[4]	(15 dengan 26)	Tidak Berubah
X[4]	Dengan	X[5]	(26 dengan 31)	Tidak Berubah

Penjelasan

Program akan melakukan perulangan bersangkar, perulangan pertama untuk menjalankan iterasi dan perulangan kedua untuk menjalankan indeks data setiap iterasi. Selama perulangan kedua akan menjalankan fungsi if, apakah array $[j - 1]$ / array 0 lebih besar dari array 1. Jika iya, indeks 0 akan disimpan ke variabel sementara dan data indeks 1 ditukar ke indeks 0, sehingga data indeks 0 ditukar ke indeks 1. Jika tidak, array 1 akan berjalan membandingkan dengan data lainnya.

Seperti iterasi 1 dan 2 diatas, indeks 0 dan 1 dibandingkan, ternyata 3 lebih besar dari 2, sehingga data ditukar, dilanjutkan indeks 1 dan 2, ternyata 3 tidak lebih besar dari 5, sehingga data tidak ditukar, dilanjutkan indeks 2 dan 3, ternyata 5 tidak lebih besar dari 7, sehingga data tidak ditukar, dilanjutkan indeks 2 tadi yang sudah dipindahkan untuk membandingkan, indeks 3 dan 4, ternyata 7 tidak lebih besar dari 8, sehingga data tidak ditukar, dilanjutkan indeks 4 dan 5, ternyata 8 tidak lebih besar dari 9, sehingga data tidak berubah. Begitu seterusnya hingga perulangan 1 selesai dan data tidak ditukar.

Ilustrasi Selectionsort

Iterasi 1

X[0]	Dengan	X[1]	(5 dengan 8)	Tidak Berubah
X[1]	Dengan	X[2]	(5 dengan 26)	Tidak Berubah
X[2]	Dengan	X[3]	(5 dengan 15)	Tidak Berubah
X[3]	Dengan	X[4]	(5 dengan 11)	Tidak Berubah
X[4]	Dengan	X[5]	(5 dengan 31)	Tidak Berubah

Iterasi 2

X[0]	Dengan	X[1]	(8 dengan 26)	Tidak Berubah
X[1]	Dengan	X[2]	(8 dengan 15)	Tidak Berubah
X[2]	Dengan	X[3]	(8 dengan 11)	Tidak Berubah
X[3]	Dengan	X[4]	(8 dengan 31)	Tidak Berubah

Iterasi 3

X[0]	Dengan	X[1]	(26 dengan 15)	Tidak Berubah
X[1]	Dengan	X[2]	(26 dengan 11)	Ditukar
X[2]	Dengan	X[3]	(11 dengan 31)	Tidak Berubah

Iterasi 4 – Data Terurut

Penjelasan

Program akan melakukan perulangan bersangkar, perulangan pertama untuk menjalankan iterasi dan menyimpan indeks yang sedang dijalankan dan perulangan kedua untuk menjalankan indeks data setiap iterasi. Selama perulangan kedua akan menjalankan fungsi if, apakah array [j] / array 0 lebih besar dari array 1. Jika iya, indeks 0 akan disimpan ke variabel sementara, jika tidak, array akan berjalan membandingkan dengan data lainnya.

Seperti iterasi 0 dan 1 diatas, indeks 1 dan 0 dibandingkan, ternyata 7 tidak lebih kecil dari 5, sehingga data tidak ditukar, dilanjutkan indeks 2 dan 0, ternyata 5 tidak lebih kecil dari 3, sehingga data tidak ditukar, dilanjutkan indeks 3 dan 0, ternyata 9 tidak lebih kecil dari 3, sehingga data tidak ditukar”, dilanjutkan indeks 4 dan 0, ternyata 3 tidak lebih kecil dari 2, sehingga data tidak ditukar, dilanjutkan indeks 5 dan 0, ternyata 8 tidak lebih kecil dari 2, sehingga data tidak ditukar. Begitu seterusnya hingga perulangan 1 selesai dan data tidak ditukar.

Ilustrasi Insertionsort

Iterasi 1

X[0]	Dengan	X[1]	(5 dengan 8)	Tidak Berubah
X[1]	Dengan	X[2]	(8 dengan 26)	Tidak Berubah
X[2]	Dengan	X[3]	(26 dengan 15)	Ditukar
X[2]	Dengan	X[3]	(15 dengan 26)	Tidak Berubah
X[4]	Dengan	X[5]	(11 dengan 31)	Tidak Berubah

Iterasi 2

X[1]	Dengan	X[2]	(8 dengan 15)	Tidak Berubah
X[2]	Dengan	X[3]	(15 dengan 26)	Tidak Berubah
X[3]	Dengan	X[4]	(26 dengan 11)	Ditukar
X[3]	Dengan	X[4]	(11 dengan 26)	Tidak Berubah
X[4]	Dengan	X[5]	(26 dengan 31)	Tidak Berubah

Iterasi 3

X[2]	Dengan	X[3]	(15 dengan 11)	Ditukar
X[2]	Dengan	X[3]	(11 dengan 15)	Tidak Berubah
X[3]	Dengan	X[4]	(15 dengan 26)	Tidak Berubah
X[4]	Dengan	X[5]	(26 dengan 31)	Tidak Berubah

Iterasi 4

X[4]	Dengan	X[5]	(26 dengan 31)	Tidak Berubah
------	--------	------	----------------	---------------

Program akan melakukan perulangan , perulangan pertama untuk menjalankan iterasi dan menyimpan indek yang sedang dijalankan dan menyimpan indek dikurangi satu yang sedang dijalanin. Selama perulangan berlangsung akan menjalankan fungsi while, apakah I / array 0 lebih besar dari array -1 dan array[i]/array 0 lebih besar dari 1. Jika iya, maka data array 0 akan disimpan array 1 dan indeknya dikurangkan, jika tidak maka data akan disimpan seperti biasa.

Seperti iterasi 0 dan 1 diatas, indek 0 dan 1 di bandingkan, ternyata 5 tidak ebih besar dari 8, sehingga data tidak ditukar, dilanjutkan indek 1 dan 2, ternyata 8 tidak lebih kecil dari 26, sehingga data tidak ditukar, dilanjutkan indek 2 dan 3, ternyata 26 lebih kecil dari 3, sehingga data tidak ditukar”, dilanjutkan indek 4 dan 0, ternyata 3 tidak lebih besar dari 15, sehingga ditukar, dilanjutkan indek 3 dan 2, ternyata 15 tidak lebih besar dari 26, sehingga data tidak ditukar. Begitu seterusnya hingga perulangan 1 selesai dan data tidak ditukar.

- Tabel Perbandingan Iterasi

Perbandingan Bubblesort

	[0]	[1]	[2]	[3]	[4]	[5]	Jumlah Perbandingan (jumlah pertukaran yang terjadi)
Awal	5	8	26	15	11	31	-
Iterasi 1	5	8	15	11	26	31	2
Iterasi 2	5	8	11	15	26	31	1
Iterasi 3	5	8	11	15	26	31	0
Iterasi 4	5	8	11	15	26	31	0
Iterasi 5	5	8	11	15	26	31	0
Iterasi 6	5	8	11	15	26	31	0

Perbandingan Selectionsort

	[0]	[1]	[2]	[3]	[4]	[5]	Jumlah Perbandingan (jumlah pertukaran yang terjadi)
Awal	5	8	26	15	11	31	-
Iterasi 1	5	8	26	15	11	31	0
Iterasi 2	5	8	26	15	11	31	0
Iterasi 3	5	8	11	15	26	31	1
Iterasi 4	5	8	11	15	26	31	0
Iterasi 5	5	8	11	15	26	31	0

Iterasi 6	5	8	11	15	26	31	0
-----------	---	---	----	----	----	----	---

Perbandingan InsertionSort

	[0]	[1]	[2]	[3]	[4]	[5]	Jumlah Perbandingan (jumlah pertukaran yang terjadi)
Awal	5	8	26	15	11	31	-
Iterasi 1	5	8	15	26	11	31	1
Iterasi 2	5	8	15	11	26	31	1
Iterasi 3	5	8	11	15	26	31	1
Iterasi 4	5	8	11	15	26	31	-

2. Tabel Perbandingan Waktu

Perbandingan Bubblesort

N	Nano Time
1000	865348085031900
10.000	865365834915900
100.000	865401230201300
1.000.000	Too long wait...
10.000.000	Too long wait...

Perbandingan Selectionsort

N	Nano Time
1000	865482859128300
10.000	865497663171600
100.000	865524598022300
1.000.000	Too long wait...
10.000.000	Too long wait...

Perbandingan Insertionsort

N	Nano Time
1000	865541457994400
10.000	865561118434600
100.000	865575032980500
1.000.000	Too long wait...
10.000.000	Too long wait...

