

Algoritma dan Pemrograman Lanjut

Pertemuan Ke-8 Pengurutan (*Sorting*) 1



Disusun Oleh :
Wilis Kaswidjanti, S.Si.,M.Kom.

**Jurusan Teknik Informatika
Fakultas Teknologi Industri
Universitas Pembangunan Nasional “Veteran”
Yogyakarta**

Algoritma dan Pemrograman Lanjut

Judul Materi : Pengurutan (*Sorting*) 1

Deskripsi Materi : Materi ini membahas metode sorting langsung Metode Penyisipan Langsung (*Straight Insertion Sort*), Metode Seleksi (*Straight Selection Sort*) dan Metode Penukaran (*Exchange selection*) / Gelembung (*Bubble Sort*) menggunakan tipe data array

Tujuan Instruksional Khusus :

1. Memahami dan membandingkan metode sorting menggunakan tipe data array dan algoritma rekursif
2. Mengimplemetasikan penggunaan tipe data array
3. Mendeskripsikan berbagai metode sorting
4. Mengilustrasikan berbagai metode sorting menggunakan tipe data array

Referensi :

- Buku Teks
Munir, Rinaldi (2005), *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*, Buku 2, Edisi Ketiga, Penerbit Informatika Bandung, Bab 2, hal 35-76.
Charibaldi, N. (2004), *Modul Kuliah Algoritma Pemrograman II*, Edisi Kedua, Yogyakarta
- Buku Acuan/Referensi
Brassard, Gilles (1999), *Fundamentals of algorithma*, PrinteceHall.
Jarne, Stroustrup B. (1997), *C++ Programming language*, AT &T.
Kristanto, Andri (2003), *Algoritma pemrograman C++*, Graha Ilmu.
Schildt,Herbert (2000), *The Complete Reference C++*, McGraw-Hill.
Sedgewick, R. (2000), *Algoritma Third edition In C part 5*, Addison Wesley.

PENGURUTAN (*SORTING*) 1

PENDAHULUAN

Sorting adalah suatu proses pengurutan data yang sebelumnya disusun secara acak atau tidak teratur menjadi urut dan teratur menurut suatu aturan tertentu.

Biasanya pengurutan terbagi menjadi dua yaitu :

- ascending (pengurutan dari karakter/angka kecil ke karakter/angka besar).
- descending (pengurutan dari karakter/angka besar ke karakter/angka kecil).

ISI

Untuk melakukan proses pengurutan dapat menggunakan beberapa metode antara lain :

A. Metode pengurutan langsung :

1. Metode Penukaran (*Exchange selection*) / Gelembung (*Bubble Sort*)
2. Metode Seleksi (*Straight Selection Sort*)
3. Metode Penyisipan Langsung (*Straight Insertion Sort*)

B. Metode pengurutan tidak langsung :

1. Shell Sort
2. Quick Sort
3. Merge Sort

A. Metode pengurutan langsung :

1. Metode Penukaran (*Exchange selection*) / Gelembung (*Bubble Sort*)

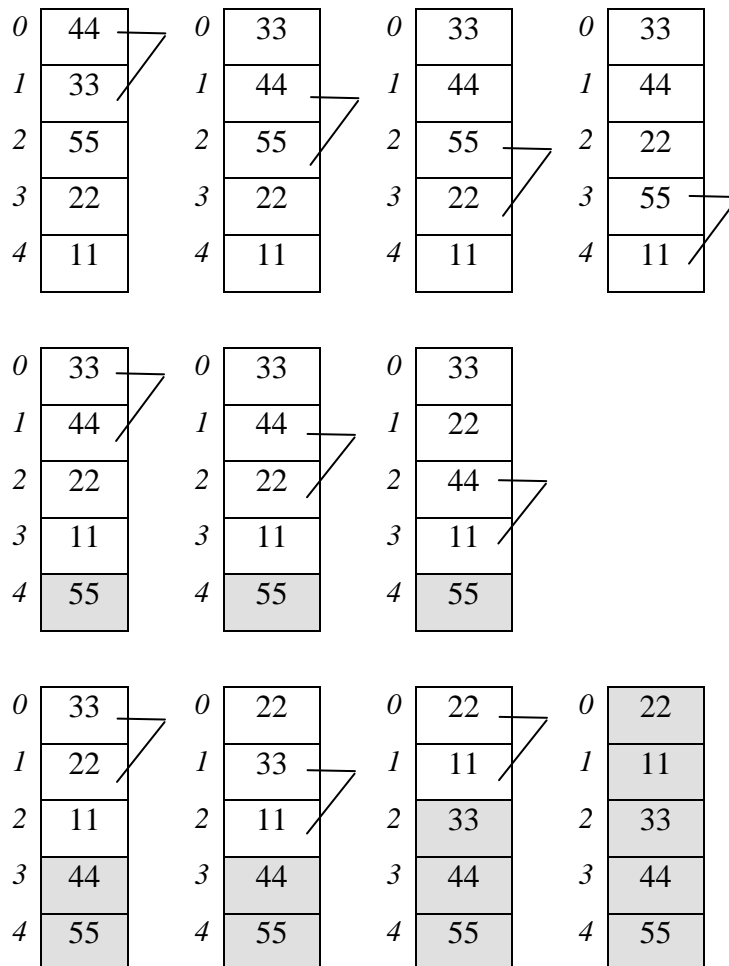
- metode pertama yang paling banyak dipelajari pemrogram.
- Sederhana →
 - a. bubble sort tidak efisien dan menyita banyak waktu prosessor lebih banyak daripada teknik sorting yang lain.
 - b. tidak lebih dari 30 atau kurang dari 30 elemen, penggunaan bubble sort masih sangat baik

- Metode gelembung / penukaran adalah metode yang mendasarkan penukaran 2 buah elemen untuk mencapai keadaan urut yang diinginkan

Langkah-langkah :

- : Baca array elemen yang diurutkan (N)
- : Kerjakan langkah 3 untuk I=1 s/d N-1
- : Kerjakan langkah 4 untuk J=1 s/d N-1
- : Cek apakah $A[J] > A[J+1]$
- : Selesai

Ilustrasi Bubble Sort



Program berikut memakai bubble sort untuk mengurutkan data array berisi 30 nilai acak :

```
/* Bubble Sort */
#include <iostream.h>
#include <stdlib.h>
#include <conio.h>

void bubble_sort(int array[], int size)
{
    int temp, i, j;
    for (i=0; i<size-1; i++)
        for (j=0; j<size-1-i; j++)
            if (array[j] > array[j+1])
                {
                    temp= array[j];
                    array[j]= array[j+1];
                    array[j+1]= temp;
                }
}
```

2. Metode Penyisipan Langsung (*Straight Insertion Sort*)

Dapat dibagi menjadi 2 bagian

- Bagian sebelah kiri data sudah terurut (tujuan)
- Bagian sebelah kanan data belum terurut (sumber)

Langkah-langkah :

- 1 : Baca array elemen yang akan diurutkan (n)
- 2 : Kerjakan langkah 3 sampai langkah 6 untuk $i : 1 \text{ s/d } n-1$
- 3 : Tentukan elemen yang akan disisipkan ($\text{Temp} = A [i]$;
 $j = i-1$;))
- 4 : Kerjakan langkah 5 selama $\text{temp} < A [j]$ dan $j \geq 0$;
- 5 : $A [j+1] = A [j]$; $j = j-1$;
- 6 : Tempatkan elemen $A [j+1] = \text{Temp}$;
- 7 : Selesai

Algoritma Straight Insertion Sort

Deklarasi

I,J,K,N : Integer

Temp : real

A : array [1..20] of real

Deskripsi

Input(N) {maksimal N=20}

K traversal [1..N]

Input (Af) {masukkan data sebanyak N}

I traversal [2..N]

Temp \leftarrow A1

J \leftarrow I-1

While (temp < A_j) and (J >= 1) do

A_{j+1} \leftarrow A_j

J \leftarrow J-1

Endwhile

A_{j+1} \leftarrow Temp

Ilustrasi Insertion Sort

i	j=i-1	Temp = A[i]	A [0]	A [1]	A [2]	A [3]	A [4]	A [5]
			4	7	9	5	8	6
1	0	7	4	<u>7</u>	9	5	8	6
2	1	9	4	7	<u>9</u>	5	8	6
3	2	5	4	7	<u>5</u>	9	8	6
	1	5	4	<u>5</u>	7	9	8	6
4	3	8	4	5	7	<u>8</u>	9	6
5	4	6	4	5	7	8	<u>6</u>	9
	3	6	4	5	7	<u>6</u>	8	9
	2	6	4	5	<u>6</u>	7	8	9

3. Metode Seleksi (*Straight Selection Sort*)

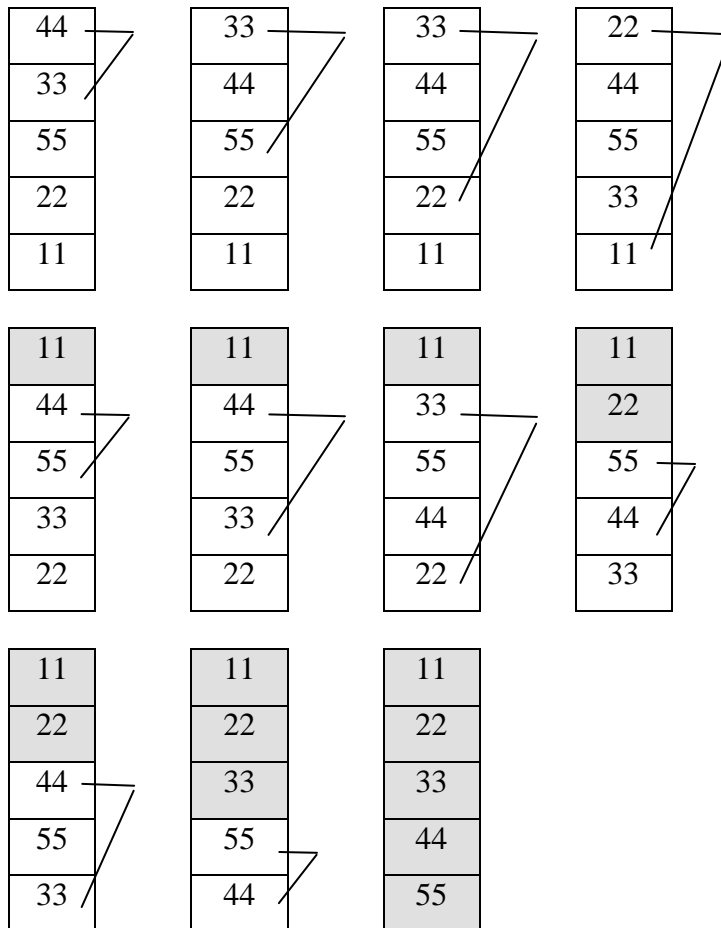
Selection sort dimulai dengan menyelesaikan elemen array (misalnya elemen pertama). Kemudian sorting mencari keseluruhan array hingga menemukan nilai yang

terkecil. Sorting menempatkan nilai terkecil pada elemen tersebut, memilih elemen kedua dan mencari elemen terkecil kedua.

Langkah-langkah Straight Selection Sort

- 1 : Baca array elemen yang diurutkan (n)
- 2 : Kerjakan langkah-langkah 3 sampai langkah 5
untuk $i=1$ s/d $n-1$
- 3 : Tentukan lokasi awal data terkecil Mindeks =1;
kerjakan langkah 4 untuk $j=i+1$ s/d n
- 4 : Cari data terkecil dan catat lokasinya. Test
apakah $A_{Mindeks} > A_j$?
Jika ya, catat Mindeks = j
- 5 : Tukar nilai $A_{mindeks}$ dengan A_j
- 6 : Selesai

Ilustrasi Straight Selection Sort



4. Metode Penyisipan Langsung (*Straight Insertion Sort*)

Dapat dibagi menjadi 2 bagian

- Bagian sebelah kiri data sudah terurut (tujuan)
- Bagian sebelah kanan data belum terurut (sumber)

Langkah-langkah :

- 1 : Baca array elemen yang akan diurutkan (n)
- 2 : Kerjakan langkah 3 sampai langkah 6 untuk $i : 1$ s/d $n-1$
- 3 : Tentukan elemen yang akan disisipkan ($Temp = A [i]$;
 $j = i-1$;))
- 4 : Kerjakan langkah 5 selama $temp < A [j]$ dan $j \geq 0$;
- 5 : $A [j+1] = A[j]$; $j = j-1$;
- 6 : Tempatkan elemen $A [j+1] = Temp$;
- 7 : Selesai

Algoritma Straight Insertion Sort

Deklarasi

I,J,K,N : Integer

Temp : real

A : array [1..20] of real

Deskripsi

Input(N) {maksimal N=20}

K traversal [1..N]

Input (Af) {masukkan data sebanyak N}

I traversal [2..N]

Temp \leftarrow A1

J \leftarrow I-1

While (temp $<$ A_j) and (J \geq 1) do

A_{j+1} \leftarrow A_j

J \leftarrow J-1

Endwhile

A_{j+1} \leftarrow Temp

Ilustrasi Insertion Sort

i	j=i-1	Temp = A[i]	A [0]	A [1]	A [2]	A [3]	A [4]	A [5]
			4	7	9	5	8	6
1	0	7	4	<u>7</u>	9	5	8	6
2	1	9	4	7	<u>9</u>	5	8	6
3	2	5	4	7	<u>5</u>	9	8	6
	1	5	4	<u>5</u>	7	9	8	6
4	3	8	4	5	7	<u>8</u>	9	6
5	4	6	4	5	7	8	<u>6</u>	9
	3	6	4	5	7	<u>6</u>	8	9
	2	6	4	5	<u>6</u>	7	8	9

PENUTUP

Ketiga metode pengurutan langsung yang dibahas pada pertemuan ini hanya beberapa dari metode pengurutan langsung lainnya. Banyak metode lain untuk mengurutkan data. Metode-metode tersebut bertujuan sama yaitu mengurutkan data, yang berbeda hanya cara yang mempengaruhi kecepatan untuk jumlah data tertentu.

SOAL-SOAL

1. Pergunakan ketiga metode pengurutan langsung di atas untuk menampilkan data dalam pengolahan data nilai suatu mata kuliah.

Buatlah menu untuk memilih metode yang dipakai, dan pilihan field yang akan dipakai sebagai key juga harus bisa dipilih (misal berdasarkan NIM atau Nilai Akhir).

Struktur Data yang dipakai sebagai berikut :

Deklarasi

TYPE

DataMhs = Record

```
< NIM : String
  Nama : String
  UTS,UAS,Prakt,NA : real {0 s/d 100}
>
```

```
BykMhs, i, j : integer
Mahasiswa : Array[1..100] of DataMhs
Temp : DataMhs
```

Nilai Akhir (NA) dihitung dengan rumus : $NA = 25\% * UTS + 50\% * UAS + 25\% * Prakt$

Proses yang dilakukan dalam program ini :

- memasukkan data,
- sorting, dan
- tampilkan data.

2. Carilah metode pengurutan langsung lainnya selain ketiga metode yang dibahas pada pertemuan ini.