

METODE PENGURUTAN DATA DALAM BAHASA PEMROGRAMAN JAVA

Disusun untuk memenuhi tugas Praktikum Algoritma dan Struktur Data

MAKALAH

disusun oleh :

Nama : **DEDE ISKANDAR**

NIM : 1127050031

Kelas : IF III/E



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN GUNUNG DJATI
BANDUNG
2013**

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Alloh SWT. karena hanya atas kehendak-Nya lah makalah ini dapat terselesaikan. Tak lupa shalawat serta salam semoga tercurah limpah kepada Nabi rahmatan lil 'alamin, Nabi Muhammad s.a.w kepada para sahabatnya, keluarganya, tabi'in-tabi'itnya, hingga pada kita selaku umatnya di yaumul akhir nanti.

Makalah ini bertujuan untuk memenuhi salah satu tugas mata kuliah Praktikum Algoritma dan Struktur Data. Selain itu juga untuk mempelajari tentang pengurutan dalam bahasa pemrograman Java

Dalam penyelesaian makalah ini, penulis mengalami banyak kesulitan terutama disebabkan oleh kurangnya ilmu pengetahuan. Untuk itu penulis mengharapkan adanya kritik dan saran yang positif agar makalah ini menjadi lebih baik. Harapan penulis, mudah-mudahan makalah ini dapat memberikan pengetahuan yang bermanfaat bagi para pembaca.

Bandung, Oktober 2013

Penulis

DAFTAR ISI

Kata Pengantar	i
Daftar Isi.....	ii
BAB I PENDAHULUAN	1
A. Latar Belakang Masalah	1
B. Rumusan Masalah.....	1
C. Batasan Masalah.....	1
D. Tujuan Penulisan	2
E. Manfaat Penulisan	2
F. Metode Penulisan.....	2
BAB II PEMBAHASAN	3
A. Pengurutan (<i>Sorting</i>).....	3
B. Metode Pengurutan	4
BAB III PENUTUP.....	10
A. Kesimpulan	10

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Pemrograman adalah sebuah seni yang tentu saja membutuhkan logika pemikiran yang tinggi. Meski demikian, tentu saja pemrograman menggunakan metode belajar yang baik dan terstruktur sesungguhnya dapat dipelajari oleh hampir semua orang yang tertarik untuk mendalaminya. Tidak perlu IQ yang sangat tinggi cukup motivasi dan kesabaran untuk terus belajar.

Metode pengurutan sangat layak untuk dipelajari oleh para pemrogram. Pada beberapa kasus yang cukup nyata, banyak pemrogram handal dapat mempelajari banyak sekali konsep-konsep dasar pemrograman hanya dengan mempelajari kasus-kasus pengurutan data.

B. Rumusan Masalah

Berdasarkan latar belakang masalah, penulis merumuskan rumusan masalah sebagai berikut :

1. Apa yang disebut dengan metode pengurutan data?
2. Bagaimana metode pengurutan data dalam bahasa pemrograman?
3. Bagaimana algoritma pengurutan data dalam bahasa pemrograman java?
4. Bagaimana *source code* pengurutan data dalam bahasa pemrograman java?

C. Batasan Masalah

Melihat dari rumusan masalah, pada makalah ini penulis memberikan batasan-batasan pada :

1. Bagaimana metode pengurutan data dalam bahasa pemrograman Java?
2. Bagaimana source code pengurutan data dalam bahasa pemrograman java?

D. Tujuan Penulisan

1. Untuk memenuhi tugas mata kuliah praktikum Algoritma dan Struktur Data
2. Untuk mengetahui bagaimana metode pengurutan data dalam bahasa pemrograman Java

E. Manfaat Penulisan

Makalah ini disusun dengan harapan memberikan kegunaan baik secara teoritis maupun secara praktis. Secara teoritis makalah ini berguna untuk sebagai bahan ajar bagaimana metode pengurutan data dalam bahasa pemrograman Java. Secara praktis makalah ini diharapkan bermanfaat bagi:

1. Penulis, sebagai wahana penambah pengetahuan dan konsep keilmuan khususnya tentang metode pengurutan dalam bahasa pemrograman Java.
2. Pembaca, sebagai media informasi tentang metode pengurutan data dalam bahasa java secara teoretis maupun praktis.

F. Metode Penulisan

Metode yang digunakan dalam penulisan makalah ini adalah metode studi pustaka, yaitu mempelajari dan mengumpulkan data dari pustaka dan baik berupa buku maupun informasi dari berbagai sumber dan juga dengan menjelajah di internet.

BAB II

PEMBAHASAN

A. PENGURUTAN (*SORTING*)

Pengurutan data secara umum bisa didefinisikan sebagai suatu penyusunan ulang (pengorganisasian) himpunan data/objek dengan aturan tertentu. Secara umum ada 2 jenis pengurutan data, yaitu pengurutan secara urut naik (*ascending*), yaitu jika data-data disusun mulai dari nilai yang paling kecil hingga yang paling besar, serta pengurutan secara urut turun (*descending*), yaitu jika data-data disusun mulai nilai yang paling besar hingga yang paling kecil. Dalam hal ini, jika yang diurutkan adalah String atau arakter (*char*), nilai data akan dikatakan lebih kecil atau lebih besar berdasarkan pada urutan relatifnya dalam tabel ASCII (*American Standard Coded for Information Interchange*) yang telah terdefinisi dengan baik sebelumnya.

Keuntungan dari data yang terurut, seperti kita singgung di atas, adalah kemudahannya dalam mencari data tertentu, serta kemudahannya untuk dilakkan perbaikan, disisipi data yang baru, dihapus, dan digabungkan. Selain itu, dalam terapan ilmu komputer, pengurutan juga digunakan dalam mengkompilasi program-program komputer jika tabel-tabel simbol harus dibentuk, serta memegang peran penting untuk mempercepat akses yang sering dilakukan terhadap data tertentu, misalnya pada aplikasi-aplikasi basis data.

Data yang harus diurutkan tentunya sangat bervariasi, baik dari segi banyaknya ataupun jenisnya. Dalam hal ini, tidak ada algoritma yang paling baik untuk semua keadaan, ada yang berkinerja dengan baik saat jumlah data sedikit namun ada yang berkinerja kurang baik saat jumlah data ditingkatkan. Sebaliknya, ada pula yang berkinerja dengan baik dengan data berjumlah sedikit maupun banyak, namun membutuhkan

ruang penyimpanan program di memori yang besar karena bersifat reursif serta sangat kompleks. Demikianlah, tidak ada algoritma yang sungguh-sungguh baik/sempurna, ditinjau dari sei kinerja (penggunaan waktu kerja prosesor) dan penggunaan ruang memori (*memori aloation*).

Pemilihan algoritma sangat ditentukan oleh struktur data yang digunakan. Dengan alasan tersebut, sejumlah metode pengurutan bisa digolongkan sebagai pengurutan internal menggunakan array dan pengurutan tata berkas akses yang bersifat eksternal. Kategori pertama dinamakan pengurutan internal karena berjalan hanya dalam memori (RAM) komputer, sedangkan yang kedua dikatakan sebagai pengurutan eksternal karena melibatkan tempat-tempat penyimpanan di luar memori komputer, seperti cakram, pita magnetik, hardisk, dan sebagainya.

B. Metode Pengurutan Data dalam bahasa pemrograman Java

Terdapat banyak metode pengurutan data dalam bahasa pemrograman java, diantaranya *Bubble sort*, *selection sort*, *insertion sort*, *shell sort*, *quick short*, *exchange sort*, *heap sort*, *tree sort*, *merge sort*, dan radix sort.

Pada makalah ini hanya akan mempelajari tiga metode pengurutan data, yaitu *insertion sort*, *shell sort*, dan *quick short*.

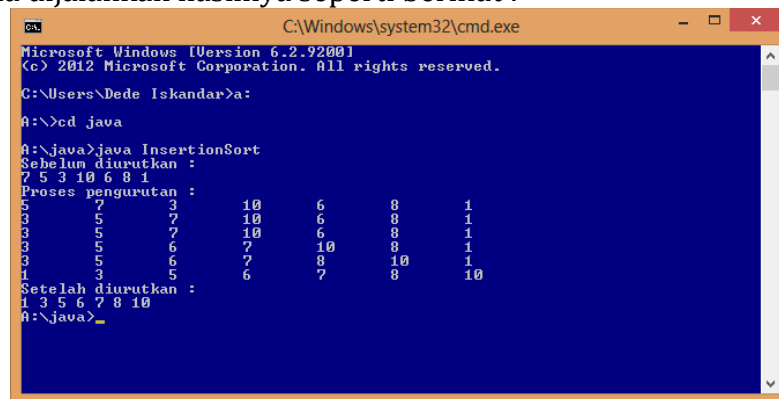
1. *Insertion sort*

insertion sort pada dasarnya memilah data yang akan diurutkan menjadi dua bagian, yang belum diurutkan (Blok A) dan yang sudah diurutkan (Blok B) Elemen pertama diambil dari bagian array yang belum diurutkan dan kemudian diletakkan sesuai posisinya pada bagian lain dari array yang telah diurutkan. Langkah ini dilakukan secara berulang hingga tidak ada lagi elemen yang tersisa pada bagian array yang belum diurutkan.

Contoh program Insertion sort

```
public class InsertionSort{
    public static void main(String[] args){
        int[] angka = {7,5,3,10,6,8,1};
        int i, j, temp;
        System.out.println("Sebelum diurutkan :");
        for(i=0;i < angka.length;i++){
            System.out.print(angka[i]+" ");
        }
        System.out.println("\nProses pengurutan :");
        for(i=1; i < angka.length; i++){
            temp = angka[i];
            j = i-1;
            while(j >= 0 && temp < angka[j]){
                angka[j+1] = angka[j];
                j = j-1;
            }
            angka[j+1] = temp;
        }
        for(int k=0;k < angka.length;k++){
            System.out.print(angka[k]+" ");
        }
        System.out.println();
        System.out.println("Setelah diurutkan :");
        for(i=0;i < angka.length;i++){
            System.out.print(angka[i]+" ");
        }
    }
}
```

Bila dijalankan hasilnya seperti berikut :



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\Dede Iskandar>a:
A:\>cd java
A:\java>java InsertionSort
Sebelum diurutkan :
7 5 3 10 6 8 1
Proses pengurutan :
5      7      3      10      6      8      1
3      5      7      10      6      8      1
3      5      7      10      6      8      1
3      5      6      7      10      8      1
3      5      6      7      8      10      1
1      3      5      6      7      8      10
Setelah diurutkan :
1 3 5 6 7 8 10
A:\java>
```

2. Shell Sort

Metode ini dikembangkan oleh Donald L Shell pada tahun 1959

Dalam metode ini jarak antara dua elemen yang dibandingkan dan ditukarkan tertentu. Secara singkat metode ini dijelaskan sebagai berikut. Pada langkah pertama, kita ambil elemen pertama dan kita bandingkan dengan elemen pada jarak tertentu dari elemen pertama tersebut. Kemudian elemen kedua kita bandingkan dengan elemen

lain dengan jarak yang sama seperti diatas. Demikian seterusnya sampai seluruh elemen dibandingkan. Pada langkah kedua proses diulang dengan langkah yang lebih kecil, pada langkah ketiga jarak tersebut diperkecil lagi seluruh proses dihentikan jika jarak sudah sama dengan satu.

Contoh program Shell Sort :

```
import java.util.Scanner;
public class ShellSort {
    public static void main (String [] args){
        Scanner input = new Scanner (System.in);
        System.out.print("Masukkan jumlah data: ");
        int [] data=new int [input.nextInt()];

        for(int i=0;i<data.length;i++){
            System.out.printf("Masukkan data ke-%d =
", (i+1));
            data[i]=input.nextInt();
        }

        System.out.println("Data belum urut: ");
        for(int i=0;i<data.length;i++){
            System.out.printf("%d ",data[i]);
        }

        int jarak=data.length;
        boolean end;
        while (jarak>1){
            jarak/=2;
            end=false;
            while (end==false){
                end=true;
                for(int j=0;j<data.length-jarak;j++){
                    if (data[j]>data[j+jarak]){
                        int tukar=data[j];
                        data[j]=data[j+jarak];
                        data[j+jarak]=tukar;
                        end=false;
                    }
                }
            }
        }

        System.out.println("\nData setelah urut: ");
        for(int i=0;i<data.length;i++){
            System.out.printf("%d ",data[i]);
        }
    }
}
```

3. Qucik Sort

Metode QuickSort diperkenalkan oleh C.A.R Hoare pada tahun 1960. Metode ini merupakan pengembangan lebih lanjut dari Shell Sort. Pengalaman dan perhitungan para ahli menunjukkan bahwa Quick Sort ini merupakan metode yang sangat efisien dibandingkan metode-metode pengurutan lain. Selain menerapkan teknik rekursif *divide and conquer*, Teknik ini juga didasarkan pada pivot yang menjadi kunci perbandingan.

Salah satu cara menemukan lokasi yang benar untuk nilai pivot dalam suatu data yang tidak terurut adalah dengan cara melakukan penyusunan ulang sedemikian rupa sehingga semua elemen yang lebih kecil dari nilai pivot akan selalu berada di sebelah kiri pivot, sebaliknya semua elemen yang nilainya lebih besar dari nilai pivot akan selalu berada di sebelah kanan pivot. Dan selanjutnya akan bergerak sedemikian rupa sehingga suatu waktu mereka akan bertemu.

Contoh program Quick Sort

```
public class QuickSort {
    int[] num;
    int n;
    public QuickSort(int n){
        this.n=n;
        // p=0;q=n;
        num=new int[n+1];
        for(int i=0;i<n;i++){
            num[i]= (int)
Math.round(Math.random()*n/2+1000);

        }
        num[n]=99999;
        // quickSort(0,n-1);
    }
    public void quickSort(){
        quickSort(0,n-1);
    }
    public void quickSort(int p,int q){
        int j;
        if(p<q){
            j=partition(p,q);
            quickSort(p,j-1);
        }
    }
}
```

```

        quickSort(j+1,q);
    }
}

private int partition(int p, int q) {
    int i=p;
    int j=q+1;
    int pivot=num[p];
    do{
        do{ i++;
        }while(num[i]<pivot);
        do{j--;
        }while(num[j]>pivot);

        if(i<j){
            swap(i,j);
        }while(i<j);
        swap(p,j);

    }

    return j;
}

private void swap(int i, int i0){
    int temp=num[i];
    num[i]=num[i0];
    num[i0]=temp;
}

public void print(){

    for(int i=0;i<n;i++){
        System.out.print(num[i]+" ");
    }
}

public static void main(String []a){
    QuickSort s=new QuickSort(100);
    System.out.println("Before Sort");
    s.print();
    s.quickSort();
    System.out.println();
    System.out.println("after Sort");
    s.print();
}
}

```

Bila dijalankan hasilnya seperti berikut

```
C:\Windows\system32\cmd.exe
A:\java>java QuickSort
Before Sort
1002 1007 1031 1042 1002 1019 1010 1038 1021 1011 1020 1047 1003 1022 1022 1006
1032 1042 1006 1037 1021 1034 1044 1000 1000 1017 1012 1047 1017 1026 1007 1020
1037 1047 1010 1009 1045 1014 1021 1030 1017 1047 1005 1047 1023 1040 1043 1017
1031 1017 1007 1026 1040 1032 1034 1003 1044 1035 1014 1010 1027 1006 1035 1045
1046 1032 1018 1041 1006 1028 1041 1013 1009 1020 1036 1015 1016 1039 1006 1031
1035 1024 1037 1017 1001 1012 1037 1027 1007 1002 1003 1043 1047 1017 1040 1041
1002 1033 1019 1046
after Sort
1000 1001 1002 1002 1002 1002 1003 1003 1003 1005 1006 1006 1006 1006 1006 1007
1007 1007 1007 1008 1009 1009 1010 1010 1010 1011 1012 1012 1013 1014 1014 1015
1016 1017 1017 1017 1017 1017 1017 1018 1019 1019 1020 1020 1020 1021 1021
1021 1022 1023 1024 1026 1026 1027 1027 1028 1029 1030 1031 1031 1031 1032 1032
1032 1033 1034 1034 1035 1035 1035 1036 1037 1037 1037 1037 1038 1039 1040 1040
1041 1041 1041 1042 1042 1043 1043 1044 1044 1045 1045 1046 1046 1047 1047 1047
1047 1047 1047 1048
A:\java>
```

BAB III

PENUTUP

A. Kesimpulan

Dengan melihat pembahasan yang ada, maka dapat diambil beberapa kesimpulan, yaitu :

1. Pengurutan data secara umum bisa didefinisikan sebagai suatu penyusunan ulang (pengorganisasian) himpunan data/objek dengan aturan tertentu.
2. Ada 2 jenis pengurutan data, yaitu pengurutan secara urut naik (ascending), yaitu jika data-data disusun mulai dari nilai yang paling kecil hingga yang paling besar, serta pengurutan secara urut turun (descending), yaitu jika data-data disusun mulai nilai yang paling besar hingga yang paling kecil.
3. Keuntungan dari data yang terurut, seperti kita singgung di atas, adalah kemudahannya dalam mencari data tertentu, serta kemudahannya untuk dilakukan perbaikan, disisipi data yang baru, dihapus, dan digabungkan.
4. metode pengurutan data yang paling populer dalam bahasa pemrograman java, diantaranya *Bubble sort*, *selection sort*, *insertion sort*, *shell sort*, *quick short*, *exchange sort*, *heap sort*, *tree sort*, *merge sort*, dan *radix sort*.
5. Tidak ada metode pengurutan yang sungguh-sungguh baik/ sempurna, ditinjau dari segi kinerja (penggunaan waktu kerja prosesor) dan penggunaan ruang memori (*memori aloation*).

DAFTAR PUSTAKA