

PERBANDINGAN BUBBLE SORT DENGAN INSERTION SORT PADA BAHASA PEMROGRAMAN C DAN FORTRAN

Reina; Josef Bernadi Gautama

Computer Science Department, School of Computer Science, Binus University
Jl. K.H. Syahdan No. 9, Palmerah, Jakarta Barat 11480
reina@binus.edu; jbernadi@binus.edu

ABSTRACT

Sorting is a basic algorithm studied by students of computer science major. Sorting algorithm is the basis of other algorithms such as searching algorithm, pattern matching algorithm. Bubble sort is a popular basic sorting algorithm due to its easiness to be implemented. Besides bubble sort, there is insertion sort. It is less popular than bubble sort because it has more difficult algorithm. This paper discusses about process time between insertion sort and bubble sort with two kinds of data. First is randomized data, and the second is data of descending list. Comparison of process time has been done in two kinds of programming language that is C programming language and FORTRAN programming language. The result shows that bubble sort needs more time than insertion sort does.

Keywords: *algorithm, sorting, bubble sort, insertion sort, C, FORTRAN*

ABSTRAK

Sorting merupakan algoritma dasar yang dipelajari oleh mahasiswa fakultas ilmu komputer. Algoritma sorting menjadi dasar bagi algoritma lain dalam melakukan pencarian, patern matching dan proses lainnya. Algoritma sorting dasar yang populer digunakan adalah bubble sort. Bubble sort menjadi salah satu algoritma sorting yang mudah dimengerti dan mudah diimplementasikan dalam bahasa program. Selain bubble sort, algoritma lain yang dapat digunakan untuk melakukan pengurutan adalah insertion sort. Insertion kurang populer daripada bubble sort karena memiliki algoritma yang lebih rumit dalam melakukan pengurutan. Paper ini membahas kecepatan proses antara bubble sort dan insertion sort menggunakan dua jenis data, yaitu data random dan data yang terurut secara descending. Penelitian ini juga membandingkan hasil sorting antara bahasa program C dan bahasa program FORTRAN. Hasil dari penelitian ini menunjukkan bahwa waktu mengurutkan yang dibutuhkan bubble sort lebih lama dibandingkan pengurutan dengan menggunakan insertion sort.

Kata kunci: *algoritma, sorting, bubble sort, insertion sort, C, FORTRAN*

PENDAHULUAN

Algoritma sorting adalah algoritma untuk mengurutkan sejumlah data yang menghasilkan urutan dari kecil ke besar (*ascending*) maupun dari besar ke kecil (*descending*). Algoritma sorting merupakan salah satu algoritma yang dipelajari oleh mahasiswa tingkat dasar pada fakultas ilmu komputer. Algoritma ini menjadi dasar dari algoritma lainnya seperti pencarian, *patern matching*, dan lain-lain seperti yang dikatakan oleh Sunny Y. Wang (1996) “*Sort Algorithm is one of the fundamental technique in computer science because of the following reasons. First, it is the basis of many other algorithms such as searching, pattern matching, digital filter etc., and many application have been found in database system*”.

Salah satu algoritma yang dapat digunakan untuk melakukan sorting adalah bubble sort. Algoritma bubble sort yang mudah diingat, menjadikan algoritma ini paling banyak dipilih oleh mahasiswa dalam membuat program sorting. Selain bubble sort ada algoritma lain yaitu insertion sort. Algoritma insertion sort agak sulit dipahami, namun diduga lebih cepat dalam melakukan sorting dibandingkan dengan bubble sort. Hal ini selaras dengan yang dikatakan oleh Astrachan (2003), “*The bubble sort algorithm is not very useful in practice, since it runs more slowly than insertion sort and selection sort, yet is more complicated to program*”.

Bubble sort memiliki tingkat popularitas yang lebih tinggi pada penelitian di tahun 2000 dan 2002. Seperti yang ditunjukkan oleh Tabel 1, pada tahun 2000 lebih banyak orang yang mengakses kata kunci bubble sort sebanyak 12.400 hits, dibandingkan dengan insertion sort yang hanya 8450 hits. Dua tahun kemudian pada tahun 2002, jumlah orang yang mengakses kata kunci bubble sort juga lebih besar dari pada insertion sort yaitu 33.800 hits, sedangkan insertion sort hanya 21.870, sehingga dapat ditarik kesimpulan bahwa bubble sort lebih populer dibandingkan dengan insertion sort.

Tabel 1 *Tingkat Popularitas Algoritma Sorting*

| Sort | Hits 2000 | Hits 2002 |
|-----------|-----------|-----------|
| Quick | 26,780 | 80,200 |
| Merge | 13,330 | 33,500 |
| Heap | 9,830 | 22,960 |
| Bubble | 12,400 | 33,800 |
| Insertion | 8,450 | 21,870 |
| Selection | 6,720 | 20,600 |
| Shell | 4,540 | 8,620 |

Untuk itu dibutuhkan suatu penelitian yang membuktikan perbandingan kecepatan implementasi kedua algoritma sorting tersebut dengan jumlah data di atas sepuluh ribu. Bahasa pemrograman yang paling banyak digunakan untuk implementasi algoritma pada fakultas ilmu komputer adalah bahasa pemrograman C, yang merupakan bahasa yang paling banyak digunakan dalam pembuatan aplikasi. Peneliti memilih dua bahasa pemrograman yang digunakan untuk penerapan kedua algoritma tersebut, yaitu bahasa pemrograman C dan bahasa pemrograman FORTRAN.

METODE

Dalam penelitian ini disiapkan dua jenis data, yaitu lima puluh ribu data random berupa bilangan bulat antara 1 sampai 10000 dan lima puluh ribu data yang terurut secara descending dari

50000 sampai 1 yang masing-masing disimpan dalam sebuah *file*. Peneliti membuat algoritma yang sama untuk diterapkan dalam bahasa pemrograman C dan bahasa pemrograman FORTRAN untuk membaca data dari file, menyimpan dalam array satu dimensi, mencatat waktu proses sorting yang dilakukan baik dengan bubble maupun insertion sort. Hasil pencatatan disimpan dalam file yang merupakan hasil dari penelitian ini.

Algoritma bubble sort yang digunakan dalam penelitian ini adalah bubble sort yang akan berhenti melakukan pengurutan dalam sekali putaran data jika data sudah terurut. Pseudocode bubble sort yang digunakan adalah sebagai berikut:

```
Start
//membaca isi file
initialize arr to array of 50000
initialize a to 0
open File 1
while not end of file "file 1"
    input arr index a from 1 line of file 1
    add 1 to a
end while

//mencetak waktu mulai
initialize time_start to time now
print time_start

//bubble sort
initialize temp to 0
initialize flag to 0
initialize n to 50000
initialize i to 1
initialize j to n-1
do
    set flag to 0
    do
        if arr index j-1 > arr index j
            then
                set temp to arr index j
                set arr index j to arr index j-1
                set arr index j-1 to temp
                set flag to 1
            end if
        minus 1 to j
    while j>=i
    add 1 to i
while i<n and flag <> 1

//mencetak waktu selesai
initialize time_end to time now
print time_end
initialize time_need to time end - time_start
print time_need
end
```

Sedangkan konsep insertion sort yang digunakan adalah insertion yang umum digunakan. Pseudocode insertion yang digunakan adalah sebagai berikut:

```
Start
//membaca isi file
initialize arr to array of 50000
initialize a to 0
open File 1
while not end of file "file 1"
    input arr index a from 1 line of file 1
    add 1 to a
end while

//mencetak waktu mulai
initialize time_start to time now
print time_start

//bubble sort
initialize temp to 0
initialize n to 50000
initialize i to 0
initialize k to 0
initialize y to 0

do
    set y to arr index k
    set i to k-1
    do
        set arr index i+1 to arr index i
        minus 1 to j
        while i>=0 and arr index i > y
            set arr index i+1 to arr i
            add 1 to k
    while k<n

//mencetak waktu selesai
initialize time_end to time now
print time_end
initialize time_need to time end - time_start
print time_need

end
```

Untuk mendapatkan hasil, digunakan komputer *notebook* yang digunakan untuk mengkompilasi program C dan FORTRAN. Komputer yang digunakan untuk menjalankan program memiliki spesifikasi sebagai berikut:

Jenis : Notebook
Type : Acer Travelmate 4740
Processor : Intel Core i3 M330 2,13 Ghz
Memory RAM : 2 GB

HASIL PEMBAHASAN

Untuk menghasilkan data random yang digunakan sebagai masukan dari aplikasi, dilakukan pengacakan angka yang dibantu oleh bahasa program C. Penggalan program yang dipakai untuk data random adalah:

```
for(int i=0; i<50000; i++)
{
    arr[i]=rand()% 10000;
    fprintf(file,"%d",arr[i]);
}
```

Variasi data masukan lainnya yang digunakan adalah data terurut dari besar ke kecil. Pembuatan data ini juga dibantu oleh aplikasi bahasa program C, dengan penggalan program sebagai berikut:

```
for(int i=50000; i>0; i--)
{
    fprintf(file,"%d",i);
}
```

Dari pseudocode yang sudah disiapkan, diterjemahkan ke bahasa program C dan FORTRAN, sehingga menghasilkan program-program sebagai berikut :

Program bubble sort yang disiapkan:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void main()
{
    long a=0,arr[50000];
    FILE *fp;
    fp=fopen("data2.txt","r");
    FILE *fw;
    fw=fopen("hasil.txt","a");
    if(fp==NULL){
        printf("File can't be opened\n"); exit(1);
    }
    while(!feof(fp)){
        fscanf(fp,"%d",&arr[a]);
        a++;
    }

    //cetak waktu sekarang
    time_t now;
    time(&now);
```

```

printf("%s", ctime(&now));
fprintf(fw,"%s", ctime(&now));
clock_t startClock = clock();

//bubble sort
long temp=0,flag=0,n=50000;
for(int i=1; i<n; i++)
{
    flag=0;
    for(int j=n-1; j>=i; j--)
    {
        if(arr[j-1] > arr[j])
        {
            temp=arr[j]; //swaping
            arr[j]=arr[j-1];
            arr[j-1]=temp;

            flag=1;
            //printf("t");
        }
    }
    if(flag==0)break;
}
//cetak waktu selesai
time(&now);
printf("%s", ctime(&now));
fprintf(fw,"%s", ctime(&now));
clock_t endClock = clock();
printf("%ld detik\n\n", (endClock - startClock) / CLOCKS_PER_SEC);
fprintf(fw,"%ld detik\n\n", (endClock - startClock) / CLOCKS_PER_SEC);
fclose(fp);
fclose(fw);
}

```

Sedangkan program code yang dipakai untuk melakukan algoritma insertion sort dengan bahasa program C adalah sebagai berikut:

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void main()
{
    long a=0,arr[50000];
    FILE *fp;
    fp=fopen("data2.txt","r");
    FILE *fw;
    fw=fopen("hasil.txt","a");
    if(fp==NULL){
        printf("File can't be opened\n"); exit(1);
    }
    while(!feof(fp)){
        fscanf(fp,"%d",&arr[a]);
    }
}

```

```

        a++;
    }

    //cetak waktu sekarang
    time_t now;
    time(&now);
    printf("%s", ctime(&now));
    fprintf(fw,"%s", ctime(&now));
    clock_t startClock = clock();

    //insertion sort
    long n=50000;
    long i, k, y;
    for(k=0; k < n; k++) {
        y = arr[k];
        for(i=k-1; i >= 0 && y < arr[i]; i--)
            arr[i+1] = arr[i];
        arr[i+1] = y;
    }

    //cetak waktu selesai
    time(&now);
    printf("%s", ctime(&now));
    fprintf(fw,"%s", ctime(&now));
    clock_t endClock = clock();
    printf("%ld detik\n\n", (endClock - startClock) / CLOCKS_PER_SEC);
    fprintf(fw,"%ld detik\n\n", (endClock - startClock) / CLOCKS_PER_SEC);
    fclose(fp);
    fclose(fw);
    getchar();
}

```

Algoritma buble sort pada bahasa program FORTRAN sebagai berikut:

```

Integer data(50000)
Character (Len=20) :: FDate@
Character (Len=8) :: Time@

Open(5,File='data1.txt')
Open(10,File='hasilbubble2.txt')
Open(20,File='hasilsort.txt')

Do 10 i = 1,50000
10     Read(5,*) data(i)

Write(10,*) FDate@(), Time@()
Call Clock@(Time1)

i = 1
Do While (i .LT. 50000)
    kflag = 0
    j = 50000

```

```

        Do While (j .GT. 1)
            If (data(j-1) .GT. data(j)) then
                kTemp = data(j)
                data(j) = data(j-1)
                data(j-1) = kTemp
                kflag = 1
            EndIf
            j = j - 1
        End Do
        i = i + 1
        If (kflag .EQ. 0) Goto 60
    End Do
60 Write(10,*) FDate@(), Time@()
    Call Clock@(Time2)
    Write(10,*) Time2-Time1

    Close(5)
    Close(10)

    Do 40 i = 1,50000
40 Write(20,*) i,data(i)
    Close(20)
End

```

Sedangkan algoritma insertion pada bahasa FORTRAN adalah sebagai berikut:

```

Integer data(50000)
Character (Len=20) :: FDate@
Character (Len=8) :: Time@

Open(5,File='data1.txt')
Open(10,File='hasilinsertion.txt')
Open(20,File='hasilsortinsertion.txt')

Do 10 i = 1,50000
10 Read(5,*) data(i)

Write(10,*) FDate@(), Time@()
Call Clock@(Time1)

k = 1
Do While (k .LE. 50000)
y = data(k)
i = k - 1
Do While ((i .GT. 0) .AND. (y .LT. data(i)))
    data(i+1) = data(i)
    i = i - 1
End DO
data(i+1) = y
k = k + 1
End Do
Write(10,*) FDate@(), Time@()

```



```

Call Clock@(Time2)
Write(10,*) Time2-Time1

Close(5)
Close(10)

Do 40 i = 1,50000
40 Write(20,*) i,data(i)

Close(20)

End

```

Untuk mengeksekusi program ini, digunakan *compiler* Visual studio express edition 2010 untuk C, dan Silverfrost FTN95 untuk FORTRAN.

Dari hasil penelitian didapatkan hasil seperti yang ditunjukkan pada Tabel 2, yaitu waktu yang dibutuhkan dalam mengurutkan menggunakan bahasa C dengan menggunakan algoritma bubble sort dengan data acak yaitu 22 detik. Sedangkan untuk data descending bubble sort membutuhkan waktu 26 detik. Sedangkan jika menggunakan algoritma insertion sort untuk data acak membutuhkan waktu 8 detik, dan 16 detik untuk data yang terurut descending.

Adanya perbedaan waktu antara algoritma bubble sort dan insertion sort juga ditunjukkan oleh table 3 dengan menggunakan bahasa FORTRAN, didapat rata-rata waktu untuk mengurutkan dengan algoritma bubble sort adalah 128 detik untuk data acak dan 151 detik untuk data terurut descending. Sedangkan mengurutkan dengan algoritma insertion sort adalah 21 detik untuk data acak, dan 43 detik untuk data terurut secara descending.

Tabel 2 Hasil Penelitian dengan Bahasa C

| Bahasa Program C | | |
|------------------|-----------|-----------------|
| | Data Acak | Data Descending |
| Bubble Sort | 22 | 26 |
| Insertion Sort | 8 | 16 |

Tabel 3 Hasil Penelitian dengan Bahasa FORTRAN

| Bahasa Program FORTRAN | | |
|------------------------|-----------|-----------------|
| | Data Acak | Data Descending |
| Bubble Sort | 128 | 151 |
| Insertion Sort | 21 | 43 |

Dari hasil yang didapat, penelitian ini menunjukkan bahwa dugaan algoritma insertion sort mengurutkan data lebih cepat dibandingkan dengan bubble sort adalah benar. Kecepatan waktu proses ini berpengaruh kepada jumlah perulangan yang dilakukan oleh kedua jenis sorting ini. Bubble sort menggunakan iterasi sebanyak n^2 , hal ini mengakibatkan pengaksesan data yang sama dilakukan berulang ulang. Sedangkan pada insertion sort, iterasi yang dilakukan hanya 1 kali kepada keseluruhan data, karena cara yang dilakukan insertion sort adalah menemukan posisi yang tepat pada setiap *key*-nya. Kecepatan eksekusi ini terbukti baik dalam bahasa pemrograman C maupun bahasa pemrograman

FORTRAN. Proses pengurutan yang lebih cepat akan membantu pengguna dalam efisiensi waktu dan penggunaan memori.

PENUTUP

Penelitian ini menyimpulkan bahwa sorting menggunakan algoritma insertion sort akan membutuhkan waktu lebih singkat dari pada algoritma bubble sort walaupun algoritma bubble sort yang lebih populer. Kecepatan proses ini dipengaruhi oleh perbedaan cara kerja setiap sort. Jumlah pengaksesan setiap *key* pada bubble sort lebih banyak dibandingkan dengan jumlah akses pada insertion sort. Hal ini mengakibatkan waktu proses akan semakin lambat, dan berpengaruh kepada keseluruhan proses sorting. Kecepatan proses ini terbukti tidak hanya di bahasa program C, tetapi juga terbukti pada bahasa program FORTRAN.

DAFTAR PUSTAKA

- Astrachan, Owen. (2003). Bubble sort: an archaeological algorithmic analysis. *SIGCSE Bull*, 35(1), 1 – 5.
- Nyhoff, L., & Leestma, S. (1983). *FORTRAN 77 for Engineers and Scientists*. Canada: Macmillan Publishing Company.
- Wang, Sunny Y. (1996). A New Sort Algorithm: Self-Indexed Sort. *SIGPLAN Notices*, 31, 28 – 36. DOI:10.1145/227717.227725