Jurnal Matematika dan Pengetahuan Alam ISSN 0853-0033
Volume 6, Nomor 1, Tahun 1997

METODE BUBBLE SORT GANDA

I Made Arnawa, Budi Rudianto, dan Narwen

Jurusan Matematika FMIPA Universitas Andalas

ABSTRACT

One of the simplest sorting algoritms is the bubble sort.
Bubble sort consists of n-1 passes. For pass I, where I=1 through n-1 bubble sort ensures that the elements in positions 1 through I are in sorted order. Bubble sort make use of the fact that elements in positions 1 through I-1 are already known to be in sorted order. **Metode Bubble Sort Ganda** combines aseending and descending scheme of bubble sort, that consists of n/2 or (n-1)/2 passed, that depend on even or odd amount of data.

PENDAHULUAN

Salah satu proses yang sering dijumpai dalam pengolahan data dengan komputer adalah proses pengurutan (sorting). Proses ini dilakukan karena :

(i) Sebagai syarat dalam pengolahan data lebih lanjut, yaitu suatu pengolahan data yang menghendaki data mempunyai urutan tertentu.

(ii) Sebagai tujuan dari proses pengurutan data itu sendiri, ini biasanya dipakai untuk memberikan tampilan laporan yang sistematis, misalnya kalau data yang diolah adalah data kepegawaian maka tampilan laporan yang dapat diberikan adalah pegawai-pegawai yang terurut menurut nomor induk pegawai (NIP).

Yang dimaksud dengan urut dalam hal ini adalah urutan fisik antara record yang ditentukan oleh salah satu field (medan). Untuk record kepegawaian misalnya, yang menjadi fieldnya adalah : Nomor Induk Pegawai (NIP), masa kerja, golongan gaji, dan lain-lainnya. Dengan demikian record kepegawaian dapat diurut menurut NIP, masa kerja, golongan gaji, dan lain-lainnya. Field yang menentukan urutan record disebut sebagai sortkey.

Metode pengurutan (Sorting) secara garis besarnya dikelompokkan atas metode maju dan metode mundur. Metode Bubble Sort yang merupakan salah satu metode Sorting memerlukan **n-1** langkah untuk mengurutkan **n** data (Edy Purnomo, 1986). Disamping itu metode Bubble Sort juga merupakan salah satu metode sorting yang mempunyai algoritma sangat sederhana (Mark Allen Weiss, 1993). Metode sorting yang baik adalah metode yang mempunyai algoritma sederhana dan memerlukan jumlah langkah yang sedikit. Cepat atau lambatnya proses sorting suatu metode bergantung kepada jumlah langkah atau jumlah perbandingan yang dilakukan oleh metode tersebut. Makin sedikit jumlah langkahnya atau jumlah perbandingannya makin cepat proses sortingnya.

Penelitian ini berkaitan dengan perumusan suatu metode sorting yang mempunyai algoritma cukup sederhana dan memerlukan jumlah langkah relatif lebih sedikit dari metode bubble sort, yaitu rumusan yang menggabungkan metode maju dan metode mundur. Rumusannya dinamakan Metode Bubble Sort Ganda.

RUMUSAN METODE BUBBLE SORT GANDA

Rumusan Metode Bubble Sort Ganda dinyatakan dalam bentuk diagram alir yang disajikan pada Gambar 1, dimana

$N^* = n/2$ untuk n genap dan
$N^* = (n-1)/2$ untuk n ganjil.

METODE PENELITIAN

Untuk membandingkan metode bubble sort ganda dengan metode bubble sort, dilakukan langkah-langkah sebagai berikut

1. Menentukan jumlah langkah yang diperlukan, masing-masing untuk metode Bubble Sort dan metode Bubble Sort Ganda.
2. Menentukan jumlah perbandingan yang diperlukan, masing-masing untuk metode Bubble Sort dan metode Bubble Sort Ganda.

Jika metode Bubble Sort Ganda, memerlukan jumlah langkah dan jumlah perbandingan yang lebih sedikit dari Bubble Sort, maka proses Sorting metode Bubble Sort Ganda lebih cepat dari metode Bubble Sort.

HASIL DAN PEMBAHASAN

Sebagai uji coba, metode Bubble Sort dan metode Bubble Sort Ganda diterapkan untuk mensorting data numerik : 351, 3,2,5,4,7,6,9,8,11,10,13,14,15,16,341,2,3,99,77,66,111,56,45, 14,15,16,13,,14,15. Program dalam bahasa pascal disajikan pada Gambar 2 dan hasil pada setiap langkah untuk masing-masing metode disajikan dalam Gambar 3 dan Gambar 4.

```
{ ***********************************************************
  * Program ini dibuat dengan menggunakan PASCAL 7          *
  *              Oleh  : I Made Arnawa                      *
  *                      Budi Rudianto                      *
  *                      N o r w e n                        *
  * Jurusan Matematika FMIPA Unand                          *
  *********************************************************** }
Program coba;
uses crt;const n=30;
type larik=array[1..n] of word;
var tipefile : text;
    i,j : word;
    A : larik;
    x : char;
Procedure Tulis_Data;
begin
 assign(tipefile,'c:\data.dat');
 rewrite(tipefile);
 for i:=1 to n do
   begin
    write('Masukan data Ke-[',i,']= ');
    readln( A[i] );
    write(tipefile,a[i]:6);
    if i mod 10=0 then writeln(tipefile);
   end;
 close(tipefile);
end;
Procedure Baca_Data;
begin
 assign(tipefile,'c:\data.dat');
 reset(tipefile);
  writeln('Keadaan Data mula-mula');
  for i:=1 to n do
   begin
    read(tipefile, A[i] );
    write(A[i]:6);
    if not eof(tipefile) then
    if i mod 10 = 0 then writeln;
```

```
  end;
close(tipefile);
 Delay(10000);
end;

Procedure Bubble_Sort;
var hold : word;
begin
 writeln('Hasil Sorting Metode Bubble Sort :');
 writeln('-------------------------------');
 for i:=1 to n-1 do
 begin
  for j:= i+1 to n do
   begin
    if a[i] > a[j] then
     begin
       hold:=a[i];
       a[i]:=a[j];
       a[j]:=hold;
     end;
    end;
    writeln('langkah ke-[',i,']');
    for j:=1 to n do write(A[j]:4);
    writeln; repeat x:=readkey; until upcase(x)='T'
end;
end;

Procedure Bubble_Sort_Ganda;
var hold, big, small,k,l,m : word;
begin
  writeln('Hasil Sorting Metode Bubble Sort Ganda :');
  writeln('---------------------------------------');
  for i:=1 to 15 do
  begin
   big:=a[i];
   l:=i;
   for j:= i+1 to (n+1-i) do
     if big < a[j] then
```

```
    begin
      big:=a[j];
      l:=j;
    end;
  hold:=a[i];
  a[i]:=big;
  a[l]:=hold;
  small:=a[i+1];
  m:=i+1;
  for k:=i+2 to ( n+1-i) do
  if small > a[k] then
    begin
      m:=k;
    end;
  hold:=a[n+1-i];
  a[n+1-i]:=small;
  a[m]:=hold;
  writeln('langkah ke-',i,':');
  for j:=1 to n do write(A[j]:4);
  writeln;
  repeat x:=readkey; until upcase(x)='T'
 end;
end;
Begin
 clrscr;
 {Tulis_data;}
 Baca_data;
 clrscr;
 {bubble_sort;}
 clrscr;
 bubble_sort_ganda;
end.
```

Gambar 2. Metode Bubble Sort dan Metode Bubble Sort Ganda dalam Bahasa PASCAL

Hasil Sortung Metode Bubble Sort :

---

langkah ke-[1]
351 3 2 5 4 7 6 9 8 11 10 13 14 15 16
341 2 3 99 77 66 111 56 45 14 15 16 13 14 15

langkah ke-[2]
351 341 2 3 4 5 6 7 8 9 10 11 13 14 15
16 2 3 99 77 66 111 56 45 14 15 16 13 14 15

langkah ke-[3]
351 341 111 2 3 4 5 6 7 8 9 10 11 13 14
15 2 3 16 77 66 99 56 45 14 15 16 13 14 15

langkah ke-[4]
351 341 111 99 2 3 4 5 6 7 8 9 10 11 13
14 2 3 15 16 66 77 56 45 14 15 16 13 14 15

langkah ke-[5]
351 341 111 99 77 2 3 4 5 6 7 8 9 10 11
13 2 3 14 15 16 66 56 45 14 15 16 13 14 15

langkah ke-[6]
351 341 111 99 77 66 2 3 4 5 6 7 8 9 10
11 2 3 13 14 15 16 56 45 14 15 16 13 14 15

langkah ke-[7]
351 341 111 99 77 66 56 2 3 4 5 6 7 8 9
10 2 3 11 13 14 15 16 45 14 15 16 13 14 15

langkah ke-[8]
351 341 111 99 77 66 56 45 2 3 4 5 6 7 8
9 2 3 10 11 13 14 15 16 14 15 16 13 14 15

langkah ke-[9]
351 341 111 99 77 66 56 45 16 2 3 4 5 6 7
8 2 3 9 10 11 13 14 15 14 15 16 13 14 15

langkah ke-[10]
351 341 111 99 77 66 56 45 16 16 2 3 4 5 6
7 2 3 8 9 10 11 13 14 14 15 15 13 14 15

langkah ke-[11]
351 341 111 99 77 66 56 45 16 16 15 2 3 4 5
6 2 3 7 8 9 10 11 13 14 14 15 13 14 15

langkah ke-[12]
351 341 111 99 77 66 56 45 16 16 15 15 2 3 4
5 2 3 6 7 8 9 10 11 13 14 14 13 14 15

langkah ke-[13]
351 341 111 99 77 66 56 45 16 16 15 15 15 2 3
4 2 3 5 6 7 8 9 10 11 13 14 13 14 14

langkah ke-[14]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 2
3 2 3 4 5 6 7 8 9 10 11 13 13 14 14

langkah ke-[15]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
2 2 3 3 4 5 6 7 8 9 10 11 13 13 14

langkah ke-[16]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
14 2 2 3 3 4 5 6 7 8 9 10 11 13 13

langkah ke-[17]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
14 13 2 2 3 3 4 5 6 7 8 9 10 11 13

langkah ke-[18]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
14 13 13 2 2 3 3 4 5 6 7 8 9 10 11

langkah ke-[19]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
14 13 13 11 2 2 3 3 4 5 6 7 8 9 10

langkah ke-[20]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
14 13 13 11 10 2 2 3 3 4 5 6 7 8 9

langkah ke-[21]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
14 13 13 11 10 9 2 2 3 3 4 5 6 7 8

langkah ke-[22]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
14 13 13 11 10 9 8 2 2 3 3 4 5 6 7

langkah ke-[23]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
14 13 13 11 10 9 8 7 2 2 3 3 4 5 6

langkah ke-[24]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
14 13 13 11 10 9 8 7 6 2 2 3 3 4 5

langkah ke-[25]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
14 13 13 11 10 9 8 7 6 5 2 2 3 3 4

langkah ke-[26]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
14 13 13 11 10 9 8 7 6 5 4 2 2 3 3

langkah ke-[27]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
14 13 13 11 10 9 8 7 6 5 4 3 2 2 3

langkah ke-[28]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
14 13 13 11 10 9 8 7 6 5 4 3 3 2 2

langkah ke-[29]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
14 13 13 11 10 9 8 7 6 5 4 3 3 2 2

Gambar 3. Hasil Sorting Dengan Metoda Bubble Sort

Hasil Sorting Metode Bubble Sort Ganda :

---

langkah ke-[1]
351 3 15 5 4 7 6 9 8 11 10 13 14 15 16 34
2 3 99 77 66 111 56 45 14 15 16 13 14 2

langkah ke-[2]
351 341 15 5 4 7 6 9 8 11 10 13 14 15 16 3 14
99 77 66 111 56 45 14 15 16 13 2 2

langkah ke-[3]
351 341 111 5 4 7 6 9 8 11 10 13 14 15 16 13 14
99 77 66 15 56 45 14 15 16 3 2 2

langkah ke-[4]
351 341 111 99 4 7 6 9 8 11 10 13 14 15 16 13 14 1
5 77 66 15 56 45 14 15 3 3 2 2

langkah ke-[5]
351 341 111 99 77 7 6 9 8 11 10 13 14 15 16 13 14 1
5 15 66 15 56 45 14 4 3 3 2 2

langkah ke-[6]
351 341 111 99 77 66 6 9 8 11 10 13 14 15 16 13 14 1
14 15 7 15 56 45 5 4 3 3 2 2

langkah ke-[7]
351 341 111 99 77 66 56 9 8 11 10 13 14 15 16 13 14
16 14 15 7 15 45 6 5 4 3 3 2 2

langkah ke-[8]
351 341 111 99 77 66 56 45 8 11 10 13 14 15 16 13 14
16 14 15 9 15 7 6 5 4 3 3 2 2

langkah ke-[9]
351 341 111 99 77 66 56 45 16 11 10 13 14 15 15
13 14 16 14 15 9 8 7 6 5 4 3 3 2 2

langkah ke-[10]
351 341 111 99 77 66 56 45 16 16 10 13 14 15 15
13 14 11 14 15 9 8 7 6 5 4 3 3 2 2

langkah ke-[11]
351 341 111 99 77 66 56 45 16 16 15 13 14 15 15
13 14 11 14 10 9 8 7 6 5 4 3 3 2 2

langkah ke-[12]
351 341 111 99 77 66 56 45 16 16 15 15 14 13 15
13 14 14 11 10 9 8 7 6 5 4 3 3 2 2

langkah ke-[13]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
13 14 13 11 10 9 8 7 6 5 4 3 3 2 2

langkah ke-[14]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
14 13 13 11 10 9 8 7 6 5 4 3 3 2 2

langkah ke-[15]
351 341 111 99 77 66 56 45 16 16 15 15 15 14 14
14 13 13 11 10 9 8 7 6 5 4 3 3 2 2

Gambar 4. Hasil Sorting dengan Metode Bubble Sort Ganda

Pembahasan dari sorting data meliputi :

1. Perbandingan langkah-langkah yang dilakukan metode Bubble Sort dan metode Bubble Sort Ganda.
2. Proses pembandingan antar Sortkey dalam masing-masing metode.

Untuk mensorting 30 data, metode Bubble Sort dan metode Bubble Sort Ganda membutuhkan masing-masing 29 langkah dan 15 langkah. Proses pembandingan antar Sortkey masing-masing metode melakukan proses 29^2 kali untuk metode Bubble Sort dan (30x29)/2 kali. Jumlah pembandingan yang dilakukan oleh masing-masing metode dapat diperoleh dengan menggunakan rumus permutasi.

Untuk mensorting n data ($n > 2$) metode Bubble Sort membutuhkan n-1 langkah dan melakukan $(n-1)^2$ proses pembandingan. Metode Bubble Sort Ganda membutuhkan n/2 langkah jika n genap dan (n-1)/2 langkah jika n ganjil. Sedangkan proses pembandingan antar sortkey membutuhkan n (n-1)/2 proses pembandingan. Dengan menggunakan sifat ketaksamaan kuadrat dapat ditunjukkan bahwa $n(n-1)/2 < (n-1)^2$.

KESIMPULAN

Metode Bubble Sort Ganda mempunyai algoritma cukup sederhana, sehingga mudah dipahami dan mudah dalam penyusunan programnya. Untuk mensorting n data ($n > 2$) metode Bubble Sort ganda membutuhkan jumlah langkah dan proses pembandingan antar sortkey yang lebih sedikit dari metode Bubble Sort, sehingga proses sorting metode Bubble Sort Ganda lebih cepat.

DAFTAR PUSTAKA

Conte, Samuel Daniel. 1982. *Elementary Numerical Analysis* 3 rd. ed. Mc Graw - Hill. London.

Gottfried, B.S. 1982. *The Theory and Problems of Programming with BASIC*. 2nd ed. Mc Graw - Hill International Book Company. Singapore.

Purcell, E.J. 1984. *Calculus With Analytic Geometry*. 4th ed. Prentice Hall, Inc. Arizona.

Purnomo, Edy. 1986. *Pemrograman Komputer dalam Bahasa Basic*. Penerbit Binacipta. Bandung.

Weiss, Mark Allen. 1994. *Data Structures and Algorithm Analysis in C++*. The Benyamin / Cummings Publishing Company, Inc. Redwood City. California.