Konsep Sorting dalam Pemrograman

Saniman dan Muhammad Fathoni

Abstrak

Sort adalah proses pengurutan data yang sebelumnya disusun secara acak sehingga menjadi tersusun secara teratur menurut suatu aturan tertentu. Empat teknik pengurutan yaitu: (1) Bubble Sort; (2) Selection Sort; (3) Insertion Sort; dan (4) Quick Sort. Metode Bubble Sort mengurutkan elemen dengan memindahkan elemen yang sekarang dengan elemen yang berikutnya, jika elemen sekarang nilainya lebih besar (>) dari elemen berikutnya, maka tukar posisi. Metode Selection Sort adalah teknik yang membandingkan elemen yang sekarang dengan elemen yang berikutnya sampai dengan elemen yang terakhir. Jika ditemukan elemen lain yang lebih kecil dari elemen sekarang maka dicatat posisinya dan kemudian ditukar. Dan begitu seterusnya. Sedangkan pada metode Insertion Sort pengurutan dilakukan dengan cara membandingkan data ke-I (dimana I dimulai dari data ke-2 sampai dengan data terakhir) dengan data berikutnya. Jika ditemukan data yang lebih kecil maka data tersebut disisipkan ke depan sesuai posisi yang seharusnya. Sedangkan Quick Sort akan membandingkan suatu elemen (pivot) dengan elemen yang lain sehingga elemen-elemen lain yang lebih kecil dari pivot tersebut terletak di sebelah kirinya dan elemen-elemen lain yang lebih besar dari pivot tersebut terletak di sebelah kanannya.

Kata Kunci: sorting, bubble sort, selection sort, insertion sort, quick sort

A. DEFINISI SORT

Sort adalah proses pengurutan data yang sebelumnya disusun secara acak sehingga menjadi tersusun secara teratur menurut suatu aturan tertentu.

Pada umumnya terdapat 2 jenis pengurutan :

- Ascending (Naik)
- Descending (Turun)

Contoh :

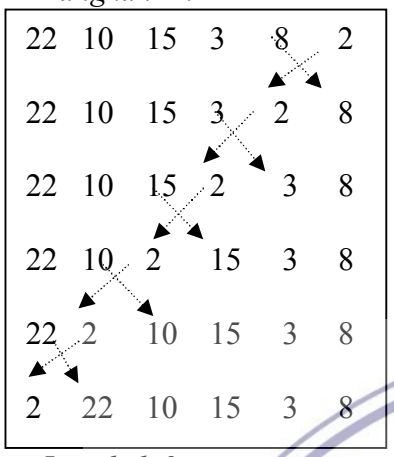
Data Acak	: 5 6 8 1 3 25 10
Terurut Ascending	: 1 3 5 6 8 10 25
Terurut Descending	: 25 10 8 6 5 3 1

B. BUBBLE / EXCHANGE SORT

Memindahkan elemen yang sekarang dengan elemen yang berikutnya, jika elemen sekarang nilainya lebih besar (>) dari elemen berikutnya, maka tukar posisi.

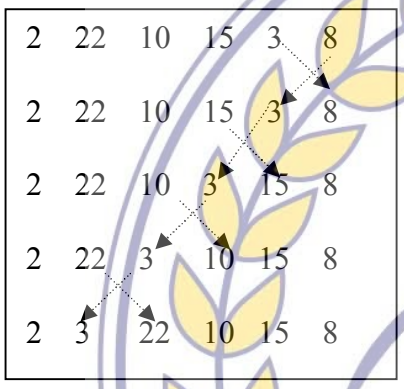
Contoh proses pengurutannya diberikan sebagai berikut:

Langkah 1 :



Pengecekan dapat dimulai dari data paling awal atau paling akhir. Pada contoh di samping ini pengecekan di mulai dari data yang paling akhir. Data paling akhir dibandingkan dengan data di depannya, jika ternyata lebih kecil maka tukar. Dan pengecekan yang sama dilakukan terhadap data yang selanjutnya sampai dengan data yang paling awal.

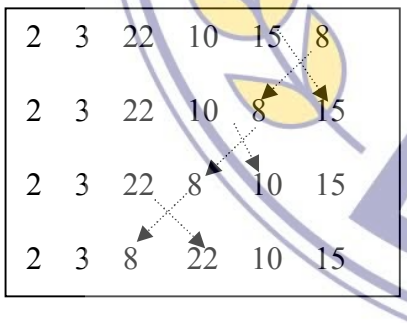
Langkah 2 :



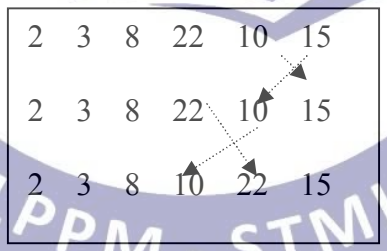
Kembalinya data paling akhir dibandingkan dengan data didepannya jika ternyata lebih kecil maka tukar, tetapi kali ini pengecekan tidak dilakukan sampai dengan data paling awal yaitu 2 karena data tersebut pasti merupakan data terkecil (didapatkan dari hasil pengurutan pada langkah 1).

Proses pengecekan pada langkah 3 dst. Sama dengan langkah sebelumnya.

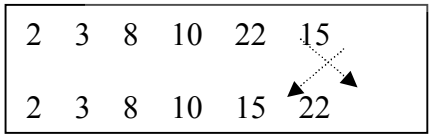
Langkah 3 :



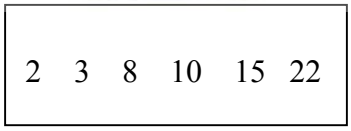
Langkah 4 :



Langkah 5 :



Terurut :



Proses di atas adalah pengurutan data dengan metoda bubble ascending. Untuk yang descending adalah kebalikan dari proses diatas. Berikut penggalan listing program Procedure TukarData dan Procedure Bubble Sort.

Procedure TukarData

```
Procedure TukarData(var a,b : word);
Var c : word;
Begin
      c:=a;
      a:=b;
      b:=c;
End;
```

Procedure Bubble Sort Ascending

```
Procedure Asc_Bubble(var data:array; jmldata:integer);
Var i,j : integer;
    Begin
      For i:= 2 to JmlData do
        For j:= JmlData DownTo i Do
          If data[j] < data[j-1] Then
            Tukardata (data[j], data[j-1]);
    End;
```

Untuk pengurutan secara descending anda hanya perlu menggantikan baris ke-6 dengan berikut ini :

```
If data[j] > data[j-1] then
```

C. SELECTION SORT

Membandingkan elemen yang sekarang dengan elemen yang berikutnya sampai dengan elemen yang terakhir. Jika ditemukan elemen lain yang lebih kecil dari elemen sekarang maka dicatat posisinya dan kemudian ditukar. Dan begitu seterusnya.

Proses :

Langkah 1:

i = 1	2	3	4	5	6
22	10	15	3	8	2

pembanding	Posisi
22 > 10	2
10 < 15	2
10 > 3	4
3 < 8	4
3 > 2	6

Posisi data ke-1(22) = 6

Tukar data ke-1 dengan data ke-6

2 10 15 3 8 22

Langkah 2 :

i = 1	2	3	4	5	6
2	10	15	3	8	22

pembanding	Posisi
10 < 15	2
10 > 3	4
3 < 8	4
3 < 22	4

Posisi data ke-2(10) = 4

Tukar data ke-2 dengan data ke-4

2 3 15 10 8 22

Langkah 3 :

i = 1	2	3	4	5	6
	2	3	15	10	8	22

pembanding	Posisi
15 > 10	4
10 > 8	5
8 < 22	5

Posisi data ke-3(15) = 5
Tukar data ke-3 dengan data ke-5

Langkah 4 :

i = 1	2	3	4	5	6
	2	3	8	10	15	22

pembanding	Posisi
10 < 15	4
10 < 22	4

Posisi data ke-4 tetap
Pada posisinya = 4 (tidak berubah)

Langkah 5 :

i = 1	2	3	4	5	6
	2	3	8	10	15	22

pembanding	Posisi
15 < 20	5

posisi data ke-5 tetap
pada posisinya = 5 (tidak

Terurut :

2	3	8	10	15	22

Proses pengurutan di atas adalah dengan metoda selection Ascending. Untuk descending hanyalah kebalikan dari proses di atas. Berikut penggalan listing program Procedure Selection Sort secara ascending

Procedure Selection Sort Ascending

```
Procedure Asc_Selection;
  Var min, pos : byte;
   Begin
      For i:= 1 to max-1 do
         Begin
           Pos:=i;
           For j:= i+1 to max do
              If data[j] < data[pos] then
                 pos:=j;
           If  i <> pos then
                 tukardata(data[i],data[pos]);
         End;
   End;
```

Untuk pengurutan secara descending, anda hanya perlu mengganti baris ke-8 sbb :

```
if data[pos] < data[j] then pos:=j;
```

D. INSERTION SORT

Pengurutan yang dilakukan dengan cara membandingkan data ke-I (dimana I dimulai dari data ke-2 sampai dengan data terakhir) dengan data berikutnya. Jika ditemukan data yang lebih

kecil maka data tersebut disisipkan ke depan sesuai posisi yang seharusnya.

Proses :

Langkah 1:

i = 1 2 3 4 5 6
22 10 15 3 8 2

temp	cek	geser
10	temp<22	data ke-1→ posisi 2

temp menempati posisi ke-1.

10 22 15 3 8 2

Langkah 2 :

i = 1 2 3 4 5 6
10 22 15 3 8 2

temp	cek	geser
15	temp<22	data ke-2→ posisi 3
	temp>10	

temp menempati posisi ke-2.

10 15 22 3 8 2

Langkah 3 :

i = 1 2 3 4 5 6
10 15 22 3 8 2

temp	cek	geser
3	temp < 22	data ke-3→ posisi 4
	temp < 15	data ke-2→ posisi 3
	temp < 10	data ke-1→posisi 2

temp menempati posisi ke-1.
3 10 15 22 8 2

Langkah 4:

i = 1 2 3 4 5 6
3 10 15 22 8 2

temp	cek	geser
8	temp < 22	data ke-4→posisi 5
	temp < 15	data ke-3→posisi 4
	temp < 10	data ke-2→posisi 3
	temp > 3	

temp menempati posisi ke-2.
3 8 10 15 22 2

Langkah 5 :

i = 1 2 3 4 5 6
3 8 10 15 22 2

temp	cek	geser
2	temp < 22	data ke-5→posisi 6
	temp < 15	data ke-4→posisi 5
	temp < 10	data ke-3→posisi 4
	temp < 8	data ke-2→posisi 3
	temp < 3	data ke-1→posisi 2

temp menempati posisi ke-1.

Langkah 6 :

2 3 8 10 15 22

```
Procedure Asc_Insert;
 Begin
 For i := 2 to max do
   Begin
     Temp :=data[i];
      j := i-1;
```

```
        while (data[j] > temp) and (j>0) do
           begin
              data[j+1] := data[j];
              dec(j);
           end;
        data[j+1]:=temp;
     end;
  end;
```

Untuk pengurutan secara descending anda tinggal mengganti baris ke 8 dengan baris berikut ini :

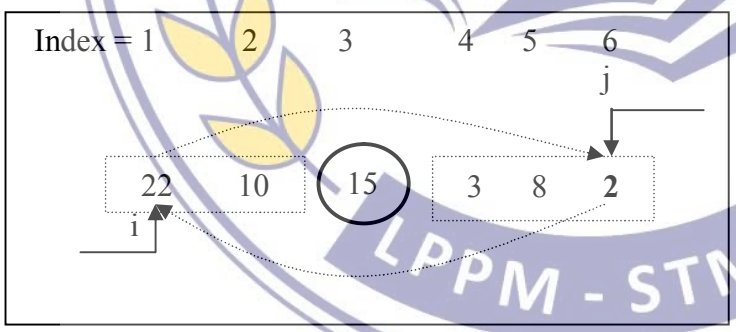
```
While(data[j] < temp)and(j>0)do
```

E. QUICK SORT

Membandingkan suatu elemen (disebut pivot) dengan elemen yang lain dan menyusunnya sedemikian rupa sehingga elemen-elemen lain yang lebih kecil daripada pivot tersebut terletak di sebelah kirinya dan elemen-elemen lain yang lebih besar daripada pivot tersebut terletak di sebelah kanannya. Sehingga dengan demikian telah terbentuk dua sublist, yang terletak di sebelah kiri dan kanan dari pivot. Lalu pada sublist kiri dan sublist kanan kita anggap sebuah list baru dan kita kerjakan proses yang sama seperti sebelumnya. Demikian seterusnya sampai tidak terdapat sublist lagi. Sehingga didalamnya telah terjadi proses Rekursif.

Proses :
Bilangan yang di dalam kurung merupakan pivot
Persegi panjang yang digambarkan dengan garis terputus-putus menunjukkan sublist.
i bergerak dari sudut kiri ke kanan sampai mendapatkan nilai yang >= pivot.
j bergerak dari sudut kanan ke kiri sampai menemukan nilai yang < pivot.

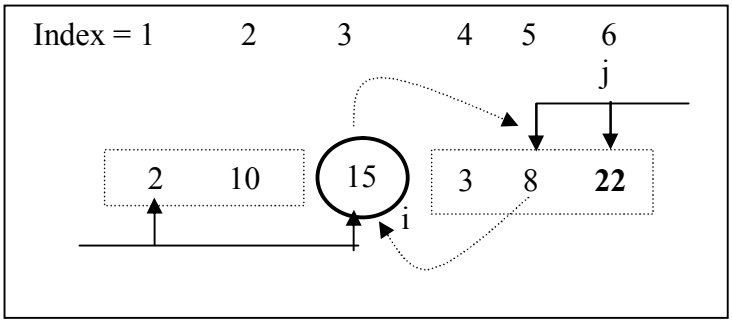
Langkah 1 :



i berhenti pada index ke-1 karena langsung mendapatkan nilai yang > dari pivot (15).
j Berhenti pada index ke-6 karena juga langsung mendapatkan nilai yang < dari pivot.
Karena i < j maka data yang ditunjuk olh I ditukar dengan data yang ditunjuk oleh j sehingga menjadi :

2 10 15 3 8 22

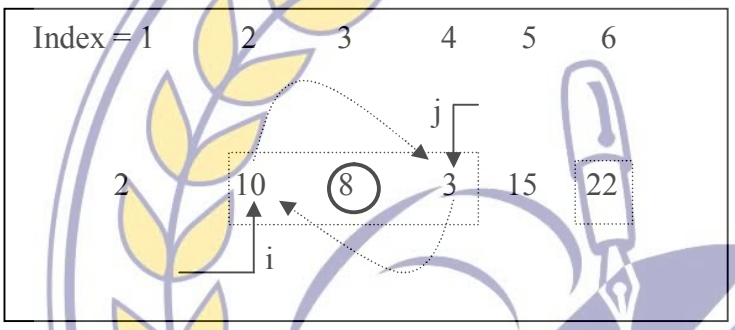
Langkah 2 :



i berhenti pada index ke-3 (pivot) karena tidak menemukan bilangan yang > dari pivot.
j berhenti pada index k-5 menunjuk pada nilai yang < dari pivot.
Karena i < j maka data yang ditunjuk oleh i (pivot) ditukar dengan data yang ditunjuk oleh j sehingga menjadi :

2 10 8 3 15 22

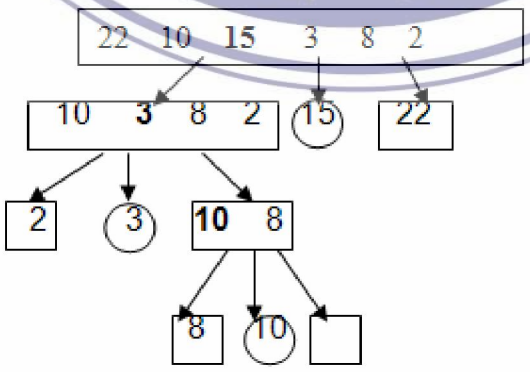
Langkah 3 :



Proses yang sama seperti sebelumnya dilakukan terhadap 2 buah sublist yang baru (ditandai dengan persegi panjang dengan garis terputus-putus).

2 3 8 10 15 22

Atau dapat juga digambarkan dalam bentuk tree seperti di bawah ini dengan pivot yang ditandai dengan huruf tebal. Kemudian setelah terurut dibaca inorder.



Procedure Quicksort dengan nilai paling kiri sebagai pembanding (pivot):

```
Procedure Asc_Quick(L,R : Integer);
Var i, j:integer;
 Begin
   If L<R then
    Begin
      i := L;  j := R+1;
       repeat
          repeat inc(i) until data[i] >= data[1];
          repeat dec(j) until data[j] <= data[1];
          if i < j then tukardata (data[i], data[j]);
       until  i > j;
      tukardata (data[1], data[j]);
      Asc_Quick(L,j-1);
      Asc_Quick(j+1,R);
    End;
End;
```

Untuk pengurutan secara descending anda tinggal mengganti tanda aritmatik pada baris ke 8 dan 9 sehingga menjadi seperti baris berikut :

```
        repeat inc(i) until data[i] >= data[1];
        repeat dec(j) until data[j] <= data[1];
```

Procedure Quick Sort dengan nilai tengah sebagai pembanding (pivot).

```
Procedure Asc_Quick(L,R : Integer);
Var mid, i, j : integer;
Begin
    i:= L; j:=R;
    mid := data[(L+R) div 2];
    repeat
       while data[i] < mid Do inc(i);
       while data[j] > mid do dec(j);
       if  i < j then
          begin
            change(data[i],data[j]);
            inc(i); dec(j);
          end;
    until  i > j;
    if L < j then Asc_Quick(L , j);
    if i > R then Asc_Quick(i , R);
end;
```

Untuk pengurutan secara descending, anda hanya perlu mengganti baris ke-6 & 7 sbb:

```
                    while data[j] < mid do inc(j);
                    while data[k] > mid do dec(k);
```

Contoh:
Anda diminta membuat sebuah program sorting dengan metode bublsort. Mintalah user untuk memasukkan 10 angka. Lalu tampilkan angka-angka tersebut setelah disort baik secara ascending maupun descending.
Layar 1:

```
Masukkan 10 data
= = = = = = = = = =
Data ke-1 = 5          Data ke-6 = 45
Data ke-2 = 2          Data ke-7 = 8
Data ke-3 = 67         Data ke-8 = 23
Data ke-4 = 43         Data ke-9 = 39
Data ke-5 = 90         Data ke-10 = 7
```

{ket : tampilan ketika mengiput 10 angka}

Layar 2 :

```
5  2  67  43  90  45  8  23  39  7
Data yang telah diurutkan :
* * * * * * * * * * * * * * * * * * *
Ascending  : 2  5  7  8  23  39  43  45  67  90
Descending : 90  67  45  43  39  23  8  7  5  2
```

{ket : tampilan setelah dilakukan bubble sort}

Jawaban :
contoh: sortir buble.

```
uses crt;
const    max = 10;
Type     arr = array[1..max] of byte;
Var      i   : byte;
        Data : arr;

Procedure Input;
 begin
   Clrscr;
   Writeln ('Masukkan 10 data');
   Writeln ('= = = = = = = = = =');
   For i := 1 to max do  {input 10 data}
         begin
       write('Data ke-', i ,'='); readln(data[i]);
    end;
   Clrscr;
   For i := 1 to max do
    Write(data[i],' ');
   Writeln;
   Writeln (' * * * * * * * * * * * * * * * * * *');
   Writeln ('Data yang telah diurutkan :');
 end;
Procedure Change (var a,b :byte); {procedure untuk menukar data}
```

```
Var c : byte;
 Begin
    c := a;  a := b;  b := c;
 end;
Procedure Asc_Buble; {pengurutan secara ascending}
Var p,q  : byte;
       flaq : boolean;
 begin
    flaq:=false;
    p:=2;
    while (p<max) and (not flaq) do
       begin
          flaq:=true;
       for  q := max downto p do
    if data[q] < data[q-1] then
       begin
          change (data[q], data[q-1]);
          flaq:=false;
             end;
                inc(i);
       end;
    write('Ascending :');
    end;
Procedure Desc_Buble; {pengurutan secara descending}
Var p, q  : byte;
       Flaq : boolean;
    Begin
     flaq:=false;
     p:=2;
     while (p<max) and (not flaq) do
       begin
         flaq:=true;
       for  q := max downto p do
    if data[q] < data[q-1] then
       begin
          change (data[q], data[q-1]);
          flaq:=false;
             end;
                inc(i);
       end;
    write('Descending :');
    end;

Procedure Output;
Begin
    For i := 1 to max do
         Write(data[i],'');
    Writeln;
end;
Begin {program utama}
    Input;
    Asc_buble; output;
    Desc_buble; output;
    Readkey;
```

```
  end.
```

Contoh Program sort

```
PROGRAM SORTING_BUBLE;
USES CRT;
     VAR DERET : array[1..100] of integer;
         loop,nested,banyak, tampung : integer;
         begin
     clrscr;
  write('Masukan berapa bilangan yang anda inginkan: ');
     readln(banyak);
     for loop :=1 to banyak  do begin
     write('bilangan ke', loop:3,'=');
     readln(deret[loop]);
     end;  {proses pengurutan ascending}
     for loop := 1 to banyak -1 do
     for nested :=loop + 1 to banyak do
     if(deret[nested]<deret[loop]) then begin
      tampung := deret[nested];
      deret[nested] :=deret [loop];
      deret[loop] :=tampung;
      end;
     {cetak hasil secara ascending}
     writeln;
     writeln('hasil sort secara Ascending');
     for loop := 1 to banyak do
      writeln('Data Ke: ',loop:3, '= ', deret[loop]);
     {cetak hasil Descending}
     writeln('Hasil sort secara Descending');
     for loop :=  banyak downto 1 do
           writeln('Data Ke: ',(banyak-loop+1):3, '= ', deret[loop]);
      readkey;
      end.
```

Contoh : Program Sort Max

```
PROGRAM Pilihmaks;
USES CRT;
     VAR DERET : array[1..100] of integer;
         loop,nested,banyak, tampung,imaks : integer;
         begin
    clrscr;
     writeln('METODE SELECTION SORT MAKSIMUM');
  write('Masukan berapa bilangan yang anda inginkan: ');
     readln(banyak);
     for loop := 1 to banyak do begin
     write('bilangan ke', loop:3,'=');
     readln(deret[loop]);
     end;
     {proses pengurutan ascending}
     for loop :=  banyak  downto 2 do begin
       imaks:=1;
       for nested := 2 to loop do
     if(deret[nested] > deret[imaks]) then
        imaks:=nested;
```

```
        tampung := deret[imaks];
      deret[imaks] := deret [loop];
      deret[loop] :=tampung;
      end;
     {cetak hasil secara ascending}
     writeln;
     writeln('hasil sort secara Ascending');
     for loop := 1 to banyak do
      writeln('Data Ke: ',loop:3, '= ', deret[loop]);
      WRITELN('====================================');
     {cetak hasil Descending}
     writeln('Hasil sort secara Descending');
     for loop :=  banyak downto 1 do
      writeln('Data Ke: ',(banyak-loop+1):3, '= ', deret[loop]);
   WRITELN('====================================');
      WRITELN('THANKS FOR YOU');
      readln;
      end.
```

Program Insert sort

```
PROGRAM Insertion_Sort;
{USES CRT;}
     VAR Larik : array[1..100] of integer;
         i,j,N, X : integer;
         ketemu : boolean;
         begin
     {clrscr;}
     writeln('Metode Pengurutan sisip max dan min');
     writeln('====================================');
     write('Masukan Banyak Data: ');
     readln(N);
        for i :=1 to N  do begin
        write('bilangan ke', i:3,'=');
        readln(Larik[i]);
       end;
     {proses pengurutan ascending}
     for I := 2 to N  do begin
       X := Larik[I];
       J  := I -1;
       ketemu :=false;
       while (J >= 1) and (not ketemu) do   begin
          if X < Larik[J] then begin
          Larik[J + 1] := Larik[J];
          J := J - 1;
          end
          else ketemu := true;
          end;
          Larik[J+1] := X;
      end;
     {cetak hasil secara ascending}
     writeln;
     writeln('hasil sort secara Ascending');
     writeln('vvvvvvvvvvvvvvvvvvvvvvvvvvv');
     for i := 1 to N do
```

```
 writeln('Data Ke: ',i:3, '= ', Larik[i]);
 writeln;
{cetak hasil Descending}
writeln('Hasil sort secara Descending');
writeln('VVVVVVVVVVVVVVVVVVVVVVVVVVVVV');
for i :=  N downto 1 do
 writeln('Data Ke: ',(N - i + 1):3, '= ', Larik[i ]);
 readln;
 end.
```

F. DAFTAR PUSTAKA

Budi Sutedjo, S.Kom., MM dan Michael AN, S.Kom., 2004, **Algoritma dan Teknik Pemrograman Konsep, Implementasi dan Aplikasi**, Yogyakarta: ANDI.

Fathul Wahid, 2004, **Dasar-Dasar Algoritma dan Pemrograman**, Yogyakarta: ANDI.

Heri Sismoro, 2005, **Pengantar Logika Informatika, Algoritma dan Pemrograman Komputer**, Yogyakarta: ANDI.

Jogiyanto H.M., 1997, **Teori dan Aplikasi Program Komputer Bahasa Pascal Jilid 1**, Yogyakarta: ANDI Offset.

Yulikuspartono, S.Kom., 2004, **Pengantar Logika dan Algoritma**, Yogyakarta: ANDI.