

SORT (pengurutan data)

Struktur data pascal



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 Pengurutan Data



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

Metode Pengurutan Data



Untuk melakukan proses pengurutan tersebut dapat digunakan berbagai macam cara / metoda. Beberapa metoda diantaranya :

- Buble / Exchange Sort
- Selection Sort
- Insertion Sort
- Quick Sort

Bubble / Exchange Sort



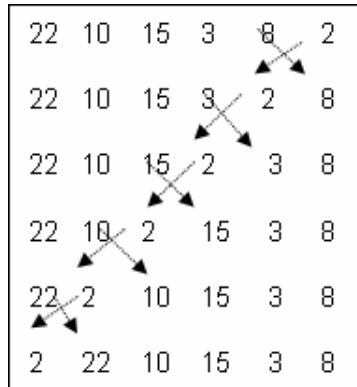
Memindahkan elemen yang sekarang dengan elemen yang berikutnya, jika elemen sekarang $>$ elemen berikutnya, maka tukar

Proses Pengurutan

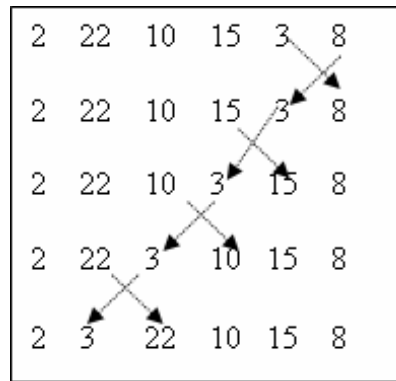


Data paling akhir dibandingkan dengan data di depannya, jika ternyata lebih kecil maka tukar. Dan pengecekan yang sama dilakukan terhadap data yang selanjutnya sampai dengan data yang paling awal.

Proses Pengurutan



Proses 1



Proses 2

Procedure Tukar Data



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

Contoh Procedure Bubble



```
Procedure Asc_Bubble;  
Var i,j,k : integer;  
  Begin  
    For i:= 2 to jmldata do  
      begin  
        For j:= jmldata downto 1 do  
          If data[j] < data[j-1] then  
            Tukardata (data[j], data[j-1]);  
          end;  
        end;  
      end;  
    end;
```

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 Selection Sort



```
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
```

Proses 1

```
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
```

Proses 2

Procedure Selection



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

Insertion Sort



Pengurutan dilakukan dengan cara membandingkan data ke-1 (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 Insertion Sort



```
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
```

Proses 1

```
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
```

Proses 2

Prosedure Insert Sort

```
procedure asc_insert;
var temp,k:integer;
begin
  For i := 2 to jmlldata 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;
```

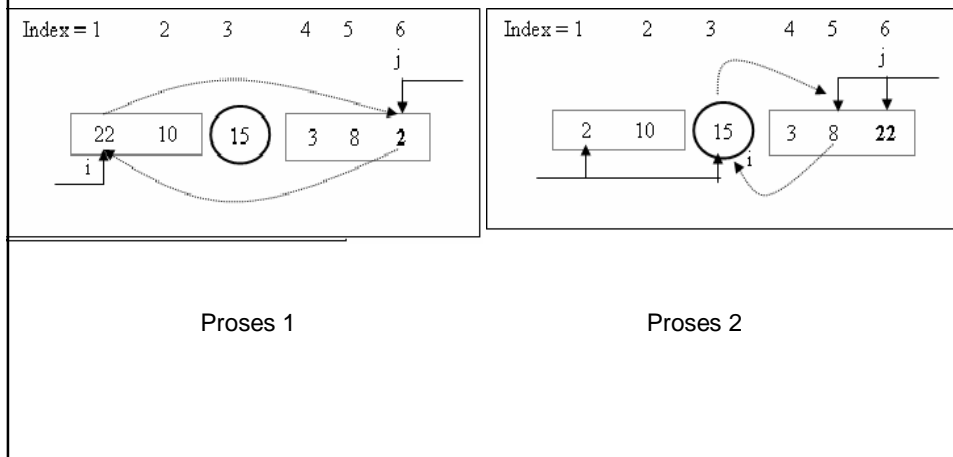


QUICK SORT

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 Quick Sort



Procedure quick sort



```

Procedure Asc_Quick(L,R : Integer);
Var i, j,k: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;
    for k:=1 to jmlldata do
      write(data[k],' ');writeln;
  End;
  
```