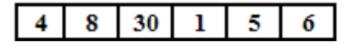
# TEKNIK PENGURUTAN DATA (SORTING)

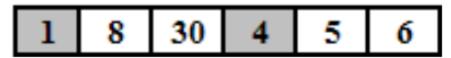
#### Pendahuluan

- Pengurutan data adalah proses menyusun kembali himpunan objek menggunakan aturan tertentu.
- Pengurutan dibagi atas dua yaitu pengurutan kecil ke besar (ascending) dan pengurutan besar ke kecil (descending)
- Berikut ini akan diperkenalkan sebuah teknik pengurutan data yang disebut dengan metode "bubble sort" atau metode gelembung.
- Misalkan larik A berikut akan diurutkan isinya:

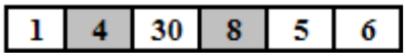


- Cara kerja metode buble sort sebagai berikut:
  - 1. Pertama kali ambil elemen pada indeks 1 sebagai patokan, A[1]=4
  - 2. Bandingkan isi A[1] dengan A[2],..[A6]. Dalam membandingkan isi elemen larik dalam proses pengurutan digunakan operator relational > untuk ascending dan < untuk descending.

- Apakah A[1] > A[2]. Karena 4 tidak lebih besar 8 maka tidak ditukar.
- Apakah A[1] > A[3]. Karena 4 tidak lebih besar 30 maka tidak ditukar.
- Apakah A[1] > A[4]. Karena 4 lebih besar 1, maka ditukar, sehingga isi elemen A menjadi



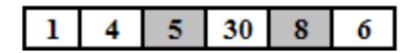
- Apakah A[1] > A[5]. Karena 1 tidak lebih besar 5 maka tidak ditukar.
- Apakah A[1] > A[6]. Karena 1 tidak lebih besar 6 maka tidak ditukar.
- Ambil isi elemen indeks 2 sebagai patokan A[2]=8, kemudian bandingkan dengan A[3].. A[6]:
  - Apakah A[2] > A[3]. Karena 8 tidak lebih besar 30 maka tidak ditukar.
  - Apakah A[2] > A[4]. Karena 8 lebih besar 4 maka ditukar sehingga isi elemen A menjadi



- Apakah A[2] > A[5]. Karena 4 tidak lebih besar 5 maka tidak ditukar.
- Apakah A[2] > A[6]. Karena 4 tidak lebih besar 6 maka tidak ditukar.
- Ambil isi elemen indeks 3 sebagai patokan A[3]=30, kemudian bandingkan dengan A[4].. A[6]:
  - Apakah A[3] > A[4]. Karena 30 lebih besar 8 maka ditukar

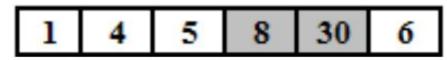


Apakah A[3] > A[5]. Karena 8 lebih besar 5 maka ditukar menjadi:



 Apakah A[3]> A[6]. Karena 5 tidak lebih besar 6 maka tidak ditukar.

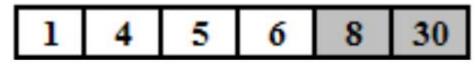
- Ambil indeks 4 sebagai patokan A[4]=30, kemudian bandingkan dengan A[5], A[6]:
  - Apakah A[4] > A[5]. Karena 30 lebih besar 8 maka ditukar menjadi:



Apakah A[4] > A[6]. Karena 8 lebih besar 6 maka ditukar menjadi:



- Ambil indeks 5 sebagai patokan A[5]=30, kemudian bandingkan dengan A[6].
  - Apakah A[5] > A[6]. Karena 30 lebih besar 8 maka ditukar menjadi:



7. Sampai disini proses pengurutan selesai.`

```
Algoritma Urut_buble_sort
Deklarasi
Const N=6
A: Array [1..N] of Integer;
indeks1, indeks2, Temp: Integer;
Deskripsi
{Baca larik A}
For indeks1←1 to N step 1 do
  Write('Masukkan elemen A ke-',indeks1); Read(A[indeks1]);
Endfor
{Urutkan Data}
For indeks1 \leftarrow1 to N-1 step 1 do
   For indeks2 ← Indeks1+1 to N step 1 do
       If(A[indeks1]>A[indeks2]) then
          Temp←A[indeks1];
          A[Indeks1] \leftarrow A[Indeks2]
         A[Indeks2] ←Temp
      Endif
    Endfor
Endfor
```

#### Praktimum 12.1. Mengurutkan Data Secara Ascending

```
//Mengurutkan data secara Ascending
                                                   //Urutkan Data
#include<iostream>
                                                    for(indeks1=0;indeks1<N-1;indeks1++)
#include<iomanip>
                                                    {for (indeks2=indeks1+1;indeks2<N;indeks2++)</pre>
#define N 6
                                                      if(A[indeks1]>A[indeks2])
using namespace std;
                                                       { temp=A[indeks1];
int A[6];
                                                        A[indeks1]=A[indeks2];
int indeks1, indeks2, temp;
                                                        A[indeks2]=temp;
main()
{ //Baca Nilai Vektor
 for (indeks1=0;indeks1<6;indeks1++)
                                                    //cetak data setelah diurutkan
 { cout<<"Masukkan Elemen A Ke:"<<indeks1;
                                                     cout<<endl<<"Data Setelah Diurutkan"<<endl;
   cin>>A[indeks1];
                                                    for (indeks1=0;indeks1<N;indeks1++)
                                                     { cout<<A[indeks1]<<setw(5);
//cetak data sebelum diurutkan
                                                    return 0;
cout<<endl<<"Data Sebelum Diurutkan"<<endl:
  for (indeks1=0;indeks1<N;indeks1++)
   { cout<<A[indeks1]<<setw(5);
```

#### Praktikum 12.2. Mengurutkan Data Secara Descending

```
//Mengurutkan data secara Ascending
#include<iostream>
#include<iomanip>
#define N 6
using namespace std;
int A[6];
int indeks1, indeks2, temp;
main()
{ //Baca Nilai Vektor
 for (indeks1=0;indeks1<6;indeks1++)
 { cout<<"Masukkan Elemen A Ke:"<<indeks1;
   cin>>A[indeks1];
//cetak data sebelum diurutkan
                                                   return 0;
cout<<endl<<"Data Sebelum Diurutkan"<<endl:
 for (indeks1=0;indeks1<N;indeks1++)
   { cout<<A[indeks1]<<setw(5);
```

```
//Urutkan Data
 for(indeks1=0;indeks1<N-1;indeks1++)
 {for (indeks2=indeks1+1;indeks2<N;indeks2++)</pre>
   if(A[indeks1]<A[indeks2])
    { temp=A[indeks1];
      A[indeks1]=A[indeks2];
      A[indeks2]=temp;
 //cetak data setelah diurutkan
  cout<<endl<<"Data Setelah Diurutkan"<<endl;
 for (indeks1=0;indeks1<N;indeks1++)
  { cout<<A[indeks1]<<setw(5);
```

# **TERIMA KASIH**