

MODUL 14 PENGURUTAN

1. Deskripsi Singkat

Istilah pengurutan adalah proses menyusun kembali data dengan aturan tertentu. pada modul ini akan dibahas tentang pengurutan data pada sekumpulan data yang bertipe sama (array). Pada modul 14 ini akan dibahas tentang algoritma pengurutan data dasar, dan juga penggunaannya dalam program.

2. Tujuan Praktikum

Setelah praktikum pada modul 14 ini diharapkan:

1. mahasiswa memahami dan menguasai konsep pengurutan menggunakan bubble sort, selection sort, dan insertion sort
2. mahasiswa menguasai teknik pengurutan dan dapat menggunakannya dalam program

3. Material Praktikum

Kegiatan pada modul 14 ini memerlukan material berupa program Pascal

4. Kegiatan Praktikum

Pada modul ini kita akan membahas tentang pengurutan data menggunakan metode bubble, selection, dan insertion dengan pengurutan secara ascending.

A. Bubble Sort

Pada bubble sort, proses pengurutan dilakukan dengan cara membandingkan satu data dengan data berikutnya, jika lebih kecil maka akan ditukar urutannya.

Berikut adalah fungsi pengurutan dengan bubble sort

```
Type Larik = array [1..100] of integer;
Procedure BubbleSort(var data: Larik; n:integer);
var
    i,j,temp:integer;
Begin
    for i:=1 to n-1 do
        for j:=n downto i+1 do
            if (data[j]<data[j-1]) then begin
                temp := data[j];
```

```

    data[j] := data[j-1];
    data[j-1] := temp;
end;
End;

```

Ketik ulang prosedur tersebut pada compiler dan lengkapi dengan bagian main program. Simpan dengan nama **praktikum14Bubble_Ascending.pas**. Jalankan program, dan cobalah urutkan isi sebuah array menggunakan program tersebut. Pastikan hasilnya sudah benar.

Simpan ulang praktikum14Bubble_Ascending.pas menjadi **praktikum14Bubble_Descending.pas**. Lakukan modifikasi pada Procedure BubbleSort agar dapat mengurutkan isi array secara menurun dari besar ke kecil.

B. Selection sort

Pada selection sort, proses pengurutan dilakukan dengan cara mencari nilai data terkecil atau terbesar pada setiap perulangan dan menempatkannya pada posisi yang sesuai. Selection sort merupakan kombinasi dari searching dan sorting. Berikut adalah prosedur untuk melakukan selection sort:

```

Type Larik = array [1..100] of integer;
Procedure SelectionSort(var Data:Larik; n:integer);
var
    i, j, min, temp:integer;
Begin
    For i:=1 to n-1 do
        begin
            Min:=i;
            For j:= i+1 to n do
                If Data[j] < Data[min] then Min:=j;
            Temp:=Data[i];
            Data[i]:=Data[min];
            Data[min]:=temp;
        End;
    End;
End;

```

Ketik ulang prosedur SelectionSort di atas pada compiler dan lengkapi dengan bagian main program. Simpan dengan nama **praktikum14Selection_Ascending.pas**. Jalankan program, dan cobalah urutkan isi sebuah array menggunakan program tersebut. Pastikan hasilnya sudah benar.

Simpan ulang praktikum14Selection_Ascending.pas menjadi **praktikum14Selection_Descending.pas**. Lakukan modifikasi pada Procedure SelectionSort agar dapat mengurutkan isi array secara menurun dari besar ke kecil.

C. Insertion Sort

Pada insertion sort, proses pengurutan dilakukan seperti mengurutkan kartu pada satu tangan. Untuk menemukan posisi yang banar, maka satu persatu kartu yang ada di tangan harus dibandingkan secara berurutan. Berikut adalah prosedur untuk melakukan selection sort:

```
Type Larik = array [1..100] of integer;
Procedure InsertionSort(var data:larik; n:integer);
var
    i,j,key:integer;
begin
    for i:=2 to n do
    begin
        key:=data[i];
        j:=i-1;
        while (j>0)and(data[j]>key) do begin
            data[j+1]:=data[j];
            j:=j-1;
        end;
        data[j+1]:=key;
    end;
End;
```

Berikut adalah contoh listing program lengkapnya

```
program Urutin;
Type Larik = array [1..100] of integer;
```

```

Procedure BubbleSort(var data: Larik; n:integer);
var
    i,j,temp:integer;
Begin
    for i:=1 to n-1 do
        for j:=n downto i+1 do
            if (data[j]<data[j-1]) then begin
                temp := data[j];
                data[j] := data[j-1];
                data[j-1] := temp;
            end;
        end;
    end;
End;

Procedure SelectionSort(var Data:Larik; n:integer);
var
    i, j, min, temp:integer;
begin
    For i:=1 to n-1 do
        begin
            Min:=i;
            For j:= i+1 to n do
                begin
                    If Data[j] < Data[min] then
                        Min:=j;
                end;
            End;
            Temp:=Data[i];
            Data[i]:=Data[min];
            Data[min]:=temp;
        end;
    end;
End;

```

```

Procedure InsertionSort(var data:larik; n:integer);
var
    i,j,key:integer;
begin
    for i:=2 to n do
    begin
        key:=data[i];
        j:=i-1;
        while (j>0)and(data[j]>key) do begin
            data[j+1]:=data[j];
            j:=j-1;
        end;

        data[j+1]:=key;
    end;
End;

var
    data:larik;
    i,n:integer;
begin
    n:=5;
    data[1]:=5;
    data[2]:=8;
    data[3]:=6;
    data[4]:=9;
    data[5]:=1;
    Insertionsort(data,n);
    {Bubblesort(data,n);}
    {Selectionsort(data,n);}
    for i:=1 to n do writeln(data[i]);

```

end.

Secara performa, ketiga prosedur pengurutan tersebut memiliki kompleksitas yang sama atau setara. Tidak ada yang lebih unggul atau lebih cepat dari yang lainnya.