**1. Bubble Sort**

Yang pertama kita akan membahas bubble sort. Algoritma ini merupakan salah satu algoritma pengurutan yang paling sederhana, baik dalam hal pengertian maupun penerapannya.

Teknik ini menyusun data yang diinginkan secara berurutan dengan membandingkan elemen data yang ada dan terus diulang hingga tidak perlu dilakukan penukaran lagi.

gambaran dari algoritma bubble sort:

*for i:=1 to Jumlah\_data-1 do*

*for j:=i+1 to Jumlah\_data do*

*if Data[i]>Data[j] then*

*begin*

*t:=Data[i];*

*Data[i]:=Data[j];*

*Data[j]:=t;*

*end;*

Kita misalkan memiliki 5 angka yang akan kita simpan kedalam variabel Data (Array).

Dengan masing-masing nilai sebagai berikut:

Data[1] := 3;

Data[2] := 1;

Data[3] := 4;

Data[4] := 2;

Data[5] := 6;

**Cara Kerja** :

**Langkah pertama** :

Data[1] akan dibandingkan dengan Data[2]. Jika Data[1] lebih besar dari Data[2] maka nilai dari kedua variabel tersebut ditukar posisinya.

**Lankah kedua :**

Data[1] akan terus dibandingkan dengan data-data selanjutnya (Data[3], Data[4], dan Data[5]). Hingga akhirnya Data[1] berisi nilai terkecil.

**Langkah ketiga :**

Setelah proses perbandingan Data[1] selesai, selanjutnya kita akan membandingkan Data[2] dengan Data[3], Data[4] dan Data[5] seperti proses sebelumnya.

**Langkah keempat :**

Begitu seterusnya sampai semua data selesai di bandingkan.

Berikut adalah contoh program pascal dengan agoritma buble sort:

*Program Bubble\_Urutan;*

*uses crt;*

*var Data:array[1 .. 5] of integer;*

*i,j,t,Jumlah\_data:integer;*

*Begin*

*Data[1] := 3;*

*Data[2] := 1;*

*Data[3] := 4;*

*Data[4] := 2;*

*Data[5] := 6;*

*Jumlah\_data := 5;*

*writeln('Data Awal:');*

*for i:=1 to Jumlah\_data do*

*begin*

*writeln('Data[',i,'] = ',Data[i]);*

*end;*

*for i:=1 to Jumlah\_data-1 do*

*for j:=i+1 to Jumlah\_data do*

*if Data[i]>Data[j] then*

*begin*

*t:=Data[i];*

*Data[i]:=Data[j];*

*Data[j]:=t;*

*end;*

*writeln('Hasil:');*

*for i:=1 to Jumlah\_data do*

*begin*

*writeln('Data[',i,'] = ',Data[i]);*

*end;*

*End.*

**2.** **Quick Short**

Algoritma quick short ditemukan oleh E. Hoare. Algoritma ini menggunakan metode rekursi sampai habis. Prinsipnya membagi data menjadi dua bagian yang sama (kiri dan kanan).

Dimana data tengah menjadi pivot (pusat operasi). Kemudian kita akan mengumpukan data dengan nilai lebih kecil dari pivot disebelah kiri pivot, dan di kanan untuk yang lebih besar.

Karena dimungkinkan bagian kiri dan kanan pivot tidak sama besarnya. maka dari itu tiap bagian di bagi menjadi dua lagi sehingga mempunyai pivot yang baru.

Source Code

*baca:=0;*

*pusat := A[(awal+akhir) div 2];*

*kiri := awal;*

*kanan := akhir;*

*While kiri <= kanan Do*

*Begin*

*While A[kiri] < pusat Do*

*Inc(kiri);*

*While A[kanan] > pusat Do*

*Dec(kanan);*

*If kiri<=kanan Then*

*Begin*

*Ganti(A[kiri],A[kanan]);*

*Inc(kiri);*

*Dec(kanan);*

*Inc(baca);*

*End;*

*End;*

*If kanan>awal Then*

*Urut(awal,kanan);*

*If akhir>kiri Then*

*Urut(kiri,akhir);*

**3 Shell Sort**

Prinsipnya hampir sama dengan Bubble Sort tetapi dioptmisisasi sehingga lebih cepat.

Ditemukan oleh Donald Shell. prinsipnya adalah membandingkan data dengan jarak tertentu dalam array. Algoritma utamanya adalah sebagai berikut :

*baca:=0;*

*For i:= (m Div 2) Downto 1 Do*

*For j:= 1 To m-i Do*

*If A[j]>A[j+i] Then*

*Begin*

*Ganti(A[j],A[j+i]);*

*Inc(baca);*

*End;*

**4 Selection Sort**

Algoritma utamanya adalah sebagai berikut:

*baca:=0;*

*For i:= 1 To m-1 Do*

*Begin*

*tempat:=i;*

*For j:= i+1 To m Do*

*If A[tempat]>A[j] Then*

*tempat:=j;*

*Ganti(A[i],A[tempat]);*

*Inc(baca);*

*End;*

**5 Insert Sort**

Algoritma utamanya adalah sebagai berikut :

*baca:=0;*

*For i:= 2 To m Do*

*Begin*

*G:=A[i];*

*j:=i-1;*

*A[0]:=G;*

*While G<A[j] Do*

**MENARA HANOI LOGIKA**



