

Nama : Hadi Suprayitno

Nim : 21091397032

Kelas : B

Shell sort

Shell sort adalah metode membandingkan suatu data dengan data yang lain dengan jarak tertentu. Lalu akan ditukarkan bila perlu. Proses pengurutan shell sort adalah menentukan jarak perbandingan data. Jaraknya adalah $N/2$ atau jumlah data dibagi 2. Apabila data pertama lebih besar dari data ke $N/2$ maka dilakukan pertukaran. Begitu seterusnya sampai data terakhir.

Setelah itu dilakukan perbandingan lagi dengan jarak $(N/2)/2$ atau $N/4$. Setelah itu dilakukan perbandingan dan ditukar apabila diperlukan. Setelah selesai penyortiran maka langkah selanjutnya yaitu melakukan perbandingan lagi dengan jarak $(N/4)/2$ atau $N/8$. Dan begitu seterusnya sampai jarak perbandingannya adalah 1

CONTOH:

Jumlah angka array : 5

Input data array : 4 20 3 9 13

Output hasil array yang telah disorting : 3 4 9 13 20

Maka coding yang digunakan seperti ini:

```

1 #include<iostream>
2
3 using namespace std;
4
5 // fungsi dari penyortiran shell.
6 void ShellSort(int a[], int n)
7 {
8     int selisih_indeks, banyak_indeks, tinggi_indeks, temp;
9     // jarak antar indeks data yang disorting, setiap dari indeks yang telah disorting akan dibagi 2
10    for(selisih_indeks = n/2; selisih_indeks > 0; selisih_indeks = selisih_indeks/2)
11    {
12        for(banyak_indeks = selisih_indeks; banyak_indeks < n; banyak_indeks++)
13        {
14            for(tinggi_indeks = banyak_indeks-selisih_indeks; tinggi_indeks >= 0; tinggi_indeks = tinggi_indeks -selisih_indeks)
15            {
16                // jika nilaipada tinggi indeks lebih besar maka looping berhenti
17                if(a[tinggi_indeks+selisih_indeks] >= a[tinggi_indeks])
18                    break;
19                // tinggi ditukarkan dengan sebaliknya.
20                else
21                {
22                    temp = a[tinggi_indeks];
23                    a[tinggi_indeks] = a[tinggi_indeks+selisih_indeks];
24                    a[tinggi_indeks+selisih_indeks] = temp;
25                }
26            }
27        }
28    }
29 }
30
31 int main()
32 {
33     // input banyaknya data yang disorting
34     int n, i;
35     cout<<"\nnasukkan jumlah data yang ingin di sorting: ";
36     cin>>n;
37
38     int arr[n];
39     for(i = 0; i < n; i++)
40     {
41         cout<<"nasukkan data "<<i+1<<" ";
42         cin>>arr[i];
43     }
44
45     ShellSort(arr, n);
46
47     // mencetak hasil sorting data.
48     cout<<"\nhasil sorting ";
49     for (i = 0; i < n; i++)
50     {
51         cout<<" " <<arr[i];
52     }
53     return 0;
54 }
```

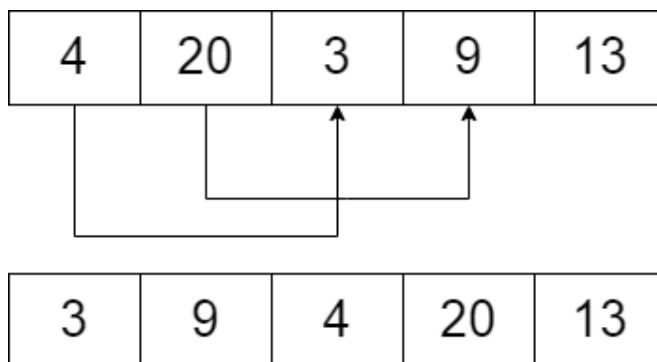
Dan akan menghasilkan output seperti ini

```
C:\Users\LaptopKU\Downloads\shell sort.exe
masukkan jumlah data yang ingin di sorting: 5
masukkan data 1: 4
masukkan data 2: 20
masukkan data 3: 9
masukkan data 4: 13
masukkan data 5: 3

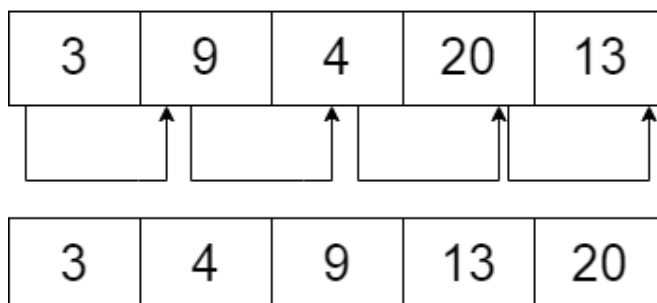
hasil sorting  3  4  9  13  20
-----
Process exited after 34.28 seconds with return value 0
Press any key to continue . . .
```

Proses yang terjadi pada sorting diatas adalah seperti berikut

Langkah pertama yaitu melakukan sorting awal dengan jarak $N/2$. Karena data yang digunakan berjumlah 5. Maka jarak yang digunakan adalah $5/2 = 2$ sehingga dihasilkan sorting awal sebagai berikut :



Setelah itu dilakukan sorting selanjutnya yaitu $N/4$ sehingga dihasilkan jarak $5/4 = 1$ dikarenakan jarak yang digunakan sudah mencapai 1 maka ini merupakan proses sorting terakhir. Sehingga hasil sortingnya sebagai berikut :



KELEBIHAN SHELL SORT

1. Algoritma ini mudah diimplementasikan
2. Operasi pertukarannya hanya beberapa kali saja
3. Waktu pengurutannya dapat lebih ditekan
4. Mudah digabungkan kembali

KEKURANGAN SHELL SORT

1. tidak efektif untuk data yang berjumlah banyak
2. sulit untuk membagi masalah
3. membutuhkan method tambahan