

BAB V SORT

Sort adalah suatu proses pengurutan data yang sebelumnya disusun secara acak atau tidak teratur menjadi urut dan teratur menurut suatu aturan tertentu.

Biasanya pengurutan terbagi menjadi 2 yaitu :

- Ascending (pengurutan dari karakter / angka kecil ke karakter / angka besar)
- Descending (pengurutan dari karakter / angka besar ke karakter / angka kecil)

Ada banyak cara yang dapat dilakukan untuk melakukan proses pengurutan dari paling tinggi ke paling rendah atau sebaliknya. Untuk masalah pengurutan pada array kita tidak dapat langsung menukar isi dari variabel yang ada, tetapi menggunakan metode penukaran (swap).

Contoh :

Data yang terdiri dari nilai dengan array, `nilai[1] = 10` dan `nilai[2] = 8` akan ditukar isi nilainya sehingga akan menghasilkan `nilai[1] = 8` dan `nilai[2] = 10`.

Proses penukaran tidak dapat langsung dilakukan dengan cara :

```
nilai[1] = nilai[2];  
nilai[2] = nilai[1];
```

Cara yang tepat adalah :

```
temp = nilai[1];  
nilai[1] = nilai[2];  
nilai[2] = temp;
```

Untuk melakukan proses pengurutan dapat menggunakan beberapa metode antara lain :

1. Bubble Sort
2. Selection Sort
3. Quick Sort

METODE SORTING

1. Bubble Sort

Bubble sort adalah suatu metode pengurutan yang membandingkan elemen yang sekarang dengan elemen berikutnya. Apabila elemen yang sekarang lebih besar dari elemen yang berikutnya, maka posisinya akan ditukar, bila tidak maka posisi akan tetap.

Misalkan data sebagai berikut :

5, 34, 32, 25, 75, 42, 22, 2

Pengurutan akan dilakukan dengan mengurutkan 8 data dari yang terkecil sampai yang terbesar.

- Dimulai dengan dua angka pertama yaitu angka 5 dan 34, untuk pengurutan dari kecil ke besar maka akan diperoleh posisi tidak berubah karena $5 < 34$.
- Langkah berikutnya akan membandingkan angka 34 dan 32. Karena $34 > 32$ maka akan terjadi pertukaran posisi, sehingga hasil urutan datanya menjadi 5, 32, 34, 25, 75, 42, 22, 2.
- $34 > 25 = 25, 34$ (berubah)
- $34 < 75 = 34, 75$ (tetap)
- $75 > 42 = 42, 75$ (berubah)
- $75 > 22 = 22, 75$ (berubah)
- $75 > 2 = 2, 75$ (berubah)

Hasil sementara menjadi 5, 32, 25, 34, 42, 22, 2, 75

Langkah kedua :

- $5 < 32 = 5, 32$ (tetap)
- $32 > 25 = 25, 32$ (berubah)
- $32 < 34 = 32, 34$ (tetap)
- $34 < 42 = 34, 42$ (tetap)
- $42 > 22 = 22, 42$ (berubah)
- $42 > 2 = 2, 42$ (berubah)
- $42 < 75 = 42, 75$ (tetap)

Hasil sementara menjadi 5, 25, 32, 34, 22, 2, 42, 75

Langkah ketiga :

- $5 < 25 = 5, 25$ (tetap)
- $25 < 32 = 25, 32$ (tetap)
- $32 < 34 = 32, 34$ (tetap)
- $34 > 22 = 22, 34$ (berubah)
- $34 > 2 = 2, 34$ (berubah)
- $34 < 42 = 34, 42$ (tetap)
- $42 < 75 = 42, 75$ (tetap)

Hasil sementara menjadi 5, 25, 32, 22, 2, 34, 42, 75

Sampai langkah terakhir yaitu langkah ketujuh :

- $5 > 2 = 2, 5$ (berubah)
- $5 < 22 = 5, 22$ (tetap)
- $22 < 25 = 22, 25$ (tetap)
- $25 < 32 = 25, 32$ (tetap)
- $32 < 34 = 32, 34$ (tetap)
- $34 < 42 = 34, 42$ (tetap)
- $42 < 75 = 42, 75$ (tetap)

Hasil sementara menjadi 2, 5, 22, 25, 32, 34, 42, 75

Proses pengurutan data tersebut membutuhkan tujuh langkah atau tujuh kali pengulangan.

Contoh :

//Program:bubble.cpp

#include <iostream.h>

#include <iomanip.h>

void main()

{

int NumList[8] = {5, 34, 32, 25, 75, 42, 22, 2};

```

int Swap;
cout<<"Data sebelum diurutkan: \n";
for(int ctr=0; ctr<8; ctr++)
{
    cout<< setw( 3 ) << NumList[ctr];
}
cout<<"\n\n";
for(int i=0; i<7; i++)
    for(int ii=0; ii<7; ii++)
        if (NumList[ii] > NumList[ii + 1])
        {
            Swap = NumList[ii];
            NumList[ii] = NumList[ii + 1];
            NumList[ii + 1] = Swap;
        }
cout<<"Data setelah diurutkan: \n";

for (int iii=0; iii<8; iii++)
    cout<< setw( 3 ) << NumList[iii];
cout<< endl <<endl;
}

```

Bila program tersebut dijalankan maka :

Data sebelum diurutkan :

5, 34, 32, 25, 75, 42, 22, 2

Data setelah diurutkan :

2, 5, 22, 25, 32, 34, 42, 75

Data pada program diatas akan diurutkan menurut ascending atau dari kecil ke besar.

2. Selection Sort

Selection sort adalah suatu metode pengurutan yang membandingkan elemen yang sekarang dengan elemen berikutnya sampai ke elemen yang terakhir. Jika ditemukan elemen lain yang lebih kecil dari elemen sekarang maka dicatat posisinya dan langsung ditukar.

Misalkan data sebagai berikut :

5, 34, 32, 25, 75, 42, 22, 2

Data tersebut akan diurutkan dengan ascending atau dari kecil ke besar.

Langkah pertama :

Posisi : 1 2 3 4 5 6 7 8

Data : 5 34 32 25 75 42 22 2

Pembandingan Posisi

- 5 < 34 1
- 5 < 32 1
- 5 < 25 1
- 5 < 75 1
- 5 < 42 1
- 5 < 22 1
- 5 > 2 8

Tukar data pada posisi 1 dengan data posisi 8.

Hasil sementara menjadi 2, 34, 32, 25, 75, 42, 22, 5

Langkah kedua :

Posisi	:	1	2	3	4	5	6	7	8
Data	:	2	34	32	25	75	42	22	5

Pembandingan Posisi

- 34 > 32 3
- 32 > 25 4
- 25 < 75 4
- 25 < 42 4
- 25 > 22 7
- 22 > 2 8

Tukar data pada posisi 2 dengan data posisi 8.

Hasil sementara menjadi 2, 5, 32, 25, 75, 42, 22, 34

Langkah ketiga :

Posisi	:	1	2	3	4	5	6	7	8
Data	:	2	5	32	25	75	42	22	34

Pembandingan Posisi

- 32 > 25 4
- 25 > 75 4
- 25 < 42 4
- 25 > 22 7
- 22 < 34 7

Tukar data pada posisi 3 dengan data posisi 7.

Hasil sementara menjadi 2, 5, 22, 25, 75, 42, 32, 34

Langkah keenam (langkah terakhir) :

Posisi	:	1	2	3	4	5	6	7	8
Data	:	2	5	22	25	32	34	75	42

Pembandingan Posisi

- 75 > 42 8

Tukar data pada posisi 7 dengan data posisi 8.

Hasil sementara menjadi 2, 5, 22, 25, 32, 34, 42, 75

Proses pengurutan data tersebut membutuhkan enam langkah atau enam kali perulangan.

Contoh :

```
void SelectionSort (int Array[ ], const int Size)
```

```
{
    int i, j, smallest, temp;
    for (i=0; i<Size; i++)
    {
        smallest=i;
        for (j=i; j<Size; j++)
        {
            if (array[smallest]>array[j])
            {
                smallest=j;
            }
        }
        temp=array[i];
        array[i]=array[smallest];
        array[smallest]=temp;
    }
}
```

3. Quick Sort

Quick sort adalah suatu metode pengurutan yang membandingkan suatu elemen (pivot) dengan elemen yang lain dan menyusunnya sedemikian rupa sehingga elemen yang lain yang lebih kecil daripada pivot terletak disebelah kiri pivot sedangkan elemen yang lebih besar dari pivot diletakkan di sebelah kanan pivot.

Sehingga akan terbentuk dua sub list yaitu yang teletak disebelah kiri pivot dan sebelah kanan pivot.

List yang sebelah kiri pivot juga diterapkan aturan seperti pivot, yaitu membandingkan dengan elemen yang lain. Jika lebih kecil akan diletakkan di sebelah kiri, jika lebih besar akan diletakkan di sebelah kanan.

Contoh :

```
void QuickSort (array A, int L, int N)
{
    if L<N
        M:=Partition (A, L, N)
        QuickSort (A, L, M-1)
        QuickSort (A, M+1, N)
    endif
}
void Partition (array A, int L, int N)
{
    select M, where L <= M <=N
    reorder A(L) ... A(N) so that L<M implies A(L) <= A(M), and L>M
    implies A(L) >= A(M)
    return M
}
```

Contoh :

Data yang akan diurutkan adalah : 20, 10, 15, 5, 8, 3

Bilangan yang terletak diantara kurung buka dan kurung tutup adalah pivot

i bergerak dari kiri ke kanan sampai mendapat nilai \geq pivot

j bergerak dari kanan ke kiri sampai mendapat nilai $<$ pivot

Langkah 1

posisi	1	2	3	4	5	6
data	20	10	(15)	5	8	3
	i					j

i berhenti pada posisi 1 karena langsung mendapatkan nilai yang lebih besar dari pivot (15) yaitu 20

j berhenti pada posisi 6 karena langsung mendapatkan nilai yang lebih kecil dari pivot (15) yaitu 3

karena $i < j$ maka data yang ditunjuk oleh i ditukar dengan data yang ditunjuk j, sehingga data berubah menjadi :

3 10 15 5 8 20

Langkah 2

posisi	1	2	3	4	5	6
data	3	10	(15)	5	8	20
			i		j	

i berhenti pada posisi 3 karena tidak dapat menemukan nilai yang lebih besar dari pivot (15)

j berhenti pada posisi 5 karena langsung mendapatkan nilai yang lebih kecil dari pivot (15) yaitu 8

karena $i < j$ maka data yang ditunjuk oleh i ditukar dengan data yang ditunjuk j, sehingga data berubah menjadi :

3 10 8 5 15 20

Langkah 3

posisi	1	2	3	4	5	6
data	3	10	(8)	5	15	20
		i		j		

i berhenti pada posisi 2 karena langsung mendapatkan nilai yang lebih besar dari pivot (8) yaitu 10

j berhenti pada posisi 4 karena langsung mendapatkan nilai yang lebih kecil dari pivot (8) yaitu 5

karena $i < j$ maka data yang ditunjuk oleh i ditukar dengan data yang ditunjuk j, sehingga data berubah menjadi :

3 5 8 10 15 20

Hasil akhirnya adalah : 3 5 8 10 15 20

CONTOH SOAL :

Soal 1

Buatlah program untuk melakukan pengurutan data secara menurun (dari besar ke kecil) dengan menukar data berikut menggunakan bubble sort.

Data : 21, 83, 42, 11, 10, 9, 3, 20, 102, 27, 15, 92, 2

Bila program dijalankan maka :

102, 92, 83, 42, 27, 21, 20, 15, 11, 10, 9, 3, 2