**Pengertian Sorting beserta metode-nya**

**Pengurutan Data (Sorting)**   
  
Dalam Bahasa pemograman C++ pengurutan disebut juga dengan “Sorting“. Pengurutan atau “Sorting“ adalah suatu proses menyusun kembali data yang sebelumnya telah disusun dengan suatu pola tertentu, sehingga tersusun secara teratur menurut aturan tertentu ( untuk data yang bertipe numerik atau karakter). Misalkan saja dalam kehidupan sehari-hari mu, kamu tuh pasti sembarangan kan menaruh buku pelajaran yang dipakai pada hari senin sampai dengan hari sabtu, kemudian untuk lebih mudahnya kamu mengurutkan buku pelajaran tersebut mana yang untuk hari senin, selasa dan sampai sabtu, jadi kamu lebih mudah mengambil buku tersebut dan tidak kebingungan. Sama saja dengan Sorting, data acak tersebut data yang sudah terurut akan lebih cepat untuk dicari.   
Pengurutan data dalam pemograman biasanya dan pada umumnya untuk data yang bertipe data numerik ataupun karakter. Pada bahasa pemograman terdapat 2 macam pengurutan yaitu, ascending (urut naik) dan descending urut turun. Ascending (urut naik) merupakan pengurutan dari angka yang nilainya lebih kecil kemudian menuju ke nilainya yang lebih besar. Sedangkan descending (urut turun) adalah sebaliknya, yaitu pengurutan dari nilainya yang lebih besar kemudian menuju ke nilainya yang lebih kecil.   
  
Contoh:   
• Data dipilih secara acak : 14 8 10 12 2 6 4   
• Ascending (urut naik) : 2 4 6 8 10 12 14   
• Descending (urut turun) : 14 12 10 8 6 4 2   
  
  
Berikut ini merupakan Metode Pengurutan Data (Sorting)   
  
*1. Bubble Sort*   
  
Nah, penjelasan awal adalah Metode bubble sort. Buble Sort merupakan metode yang sangat simpel dan mudah untuk melakukan pengurutan data , namun setiap metode tersebut pasti memiliki kelemahan dan keunggulan. Walaupun sangat sederhana namun, Metode ini mempunyai kelemahan yaitu, pada saat mengurutkan data yang sangat besar akan mengalami kekacuan, atau kinerja nya kurang baik. Mungkin kalian bingung ya, knapa sih namanya Buble, knapa nggak Circle Sort atau Soda sort aja (maav ney bukannya ngubah-ngubah metode). Berikut ini penjelasannya, “Bubble” karena proses pengurutan data nya tersebut secara bertahap bergerak/berpindah ke posisinya sesuai urutannya, misalkan saja anda meniup segelas air dengan menggunakan sedotan , tentunya akan mengeluarkan gelembung yang saling berurutan keluar dalam pipet. Pengurutan data Buble Sort dilakukan dengan cara membandingkan elemen sekarang dengan elemen berikutnya. Penukaran tersebut baru dilakukan kalau kriterianya tersebut sudah terpenuhi.   
  
Pengurutan Ascending (urut naik) : Jika elemen sekarang lebih besar dari elemen berikutnya maka kedua elemen tersebut ditukar. Pengurutan Descending ( urut turun): Jika elemen sekarang lebih kecil dari elemen berikutnya, maka kedua elemen tersebut ditukar. Nah, Algoritma ini seolah-olah menggeser satu per satu elemen dari kanan ke kiri atau kiri ke kanan, Sekarang tergantung jenis pengurutannya, ascending (urut naik) atau descending (urut turun). Ketika satu proses telah selesai, maka bubble sort akan mengulangi proses, demikian seterusnya sampai dengan iterasi sebanyak n-1. Bubble Sort berhenti jika seluruh array telah diperiksa dan tidak ada pertukaran lagi yang bisa dilakukan, serta tercapai perurutan yang telah diinginkan.   
  
*2. Exchange Sort*   
  
Pembahasan yang kedua mengenai Metode Exchange Sort. Metode ini merupakan metode pengurutan data yang hampir mirip dengan Bubble Sort ( Mirorr-Nya buble sort), bahkan mungkin juga metode Bubble Sort sama dengan Exchange Sort. Namun setiap metode pasti memiliki perbedaan, perbedaan antara Exchange Sort dan Bubble Sort terletak dalam hal bagaimana membandingkan antar elemen-elemennya.   
Exchange sort membandingkan suatu elemen dengan elemen-elemen lainnya dalam array tersebut, dan melakukan pertukaran elemen jika perlu. Jadi ada elemen yang selalu menjadi elemen pusat (pivot). Sedangkan Bubble sort akan membandingkan elemen pertama/terakhir dengan elemen sebelumnya/sesudahnya, kemudian elemen tersebut itu akan menjadi pusat (pivot) untuk dibandingkan dengan elemen sebelumnya/sesudahnya lagi,dan begitu seterusnya.   
  
*3. Selection Sort*   
  
Metode yang ketiga adalah Selection Sort merupakan metode pengurutan data kombinasi antara sorting dan searching. Kinerja metode ini sebagai berikut ini : Mula – mula suatu petunjuk (diberi nama posAwal ) yang menunjuk lokasi awal pengurutan data diatur agar berisi indeks pertama dalam larik. Selanjutnya dicari bilangan terkecil yang terletak antara posisi sesudah yang ditunjuk oleh penunjuk tersebut element yang terakhir dalam larik. Lokasi bilangan ini di tunjuk oleh posMin. Lalu tukarkan nilai bilangan terkecil tersebut dengan nilai yang di tunjukkan posAwal. Proses seperti ini di ulang dari posAwal bernilai 0 hingga n-2 , dengan n menyatakan jumlah element larik   
  
*4. Insertion Sort*   
  
Kamu pernah bermain kartu gak?? pastilah kamu tau kartu remi kan, pastinya kamu-kamu pernah mengkocok kartunya atau sebagai Bandarnya. Metode insertion sort ini merupakan metode pengurutan data yang mirip dengan cara orang mengurutkan kartu, selembar demi selembar kartu diambil dan disisipkan (insert) ke tempat yang seharusnya sehinggan penambahan kartu tersebut akan membuat semua kartu tetap terurutkan.   
Pengurutan dimulai dari data ke-2 sampai dengan data terakhir, jika ditemukan data yang lebih kecil, maka akan ditempatkan (diinsert) diposisi yang seharusnya. Pada penyisipan elemen, maka elemen-elemen lain akan bergeser ke belakang   
  
**5. Quick Sort**   
  
Pada tahun 1962, C. AR Hoare merupakan orang yang mengemukakan pertama kali Metode Quick sort. Metode ini merupakan metode pengurutan data yang agak rumit namun, sangat mudah jika ada data yang nilainya sangat besar, seperti yang saya katakan sebelumnya, jika metode tersebut simpel pastilah ada juga kekurangannya, begitu juga sebaliknya jika metode ini rumit maka kelebihannya juga ada, kira-kira seperti itu. Metode ini menggunakan strategi “pecah-belah” dengan mekanisme seperti berikut : Larik L[p..r] dengan indeks terkecil adalah p dan indeks terbesar adalah r disusun ulang (dipartisi) menjadi dua buah larik A[p..p] dan A[q+1…r] sehingga setiap elemen dalam A[p..q] selalu bernilai lebih kecil daripada A[q+1…r]. Selanjutnya kedua larik tersebut di urut secara rekursif. Dengan sendirnya kombinasi kedua larik tersebut membentuk larik dengan data yang telah di urut.   
  
 **Berikut ini contoh Buble Sort,beserta dengan metode pengurutan (sorting) yang ada:**   
  
#include   
#include   
  
int data[100],data2[100];   
int n;   
  
void tukar(int a,int b)   
{   
int t;   
t = data[b];   
data[b] = data[a];   
data[a] = t;   
}   
  
void buble\_sort()   
{   
for(int i=1;i  
{   
for(int j=n-1;j>=i;j--)   
{   
if(data [j]  
}   
}   
cout<<"=================="<  
cout<<"Bubble sort selesai..!!!"<  
cout<<"==================="<  
  
}   
  
void exchange\_sort()   
{   
for (int i=0; i  
{   
for(int j=(i+1); j  
{   
if (data [i]>data[j]) tukar(i,j);   
}   
}   
cout<<"==================="<  
cout<<"Exchange sort selesai..!!"<  
cout<<"==================="<  
  
}   
  
void selection\_sort()   
{   
int pos,i,j;   
for(i=0;i  
{   
pos = i;   
for(j = i+1;j  
{   
if(data[j] < data[pos]) pos=j;   
}   
if(pos !=i) tukar (pos,i);   
}   
cout<<"==================="<  
cout<<"Selection sort selesai"<  
cout<<"==================="<  
  
}   
  
void insertion\_sort()   
{   
int temp,i,j;   
for(i=1;i  
{   
temp= data[i];   
j = i-1;   
while(data[j]>temp && j>=0)   
{   
data[j+1] = data[j];   
j--;   
}   
data[j+1] = temp;   
}   
cout<<"==================="<  
  
cout<<"insertion sort selesai...!!!"<  
cout<<"==================="<  
  
}   
  
void QuickSort (int L,int R)   
{   
int i,j;   
int mid;   
  
i= L;   
j= R;   
mid =data[(L+R) / 2];   
  
do   
{   
while (data[i] < mid ) i++;   
while (data[j] > mid ) j--;   
  
if (i <=j)   
{   
tukar (i,j);   
i++;   
j--;   
};   
}   
while ( i  
  
if (L  
if (i  
}   
void input()   
{   
cout<<"masukkan jumlah data= ";cin>>n;   
for(int i=0;i  
{   
cout<<"Masukkan Data ke-"<<(i+1)<<"=";cin>>data[i];   
data2[i] = data[i];   
}   
}   
  
void tampil()   
{   
cout<<"data : "<  
for(int i=0;i  
{   
cout<<data[i]<<" ";=""   
}   
cout<  
}   
  
void AcakLagi()   
{   
for(int i=0;i  
{   
data[i] = data2[i];   
}   
cout<<"data sudah teracak..!!"<  
}   
void main()   
{   
int pil;   
clrscr();   
do   
{   
clrscr();   
cout<<"Program Sorting Complete...!!"<  
cout<<"============================="<  
cout<<"1. Input Data"<  
cout<<"2. Buble Sort"<  
cout<<"3. Exchange Sort"<  
cout<<"4. Selection Sort"<  
cout<<"5. Insertion Sort"<  
cout<<"6. Quick Sort"<  
cout<<"7. Tampilkan Data"<  
cout<<"8. Acak Lagi"<  
cout<<"9. Exit"<  
cout<<"Masukkan Pilihan Anda : ";   
cin>>pil;   
switch(pil)   
{   
case 1:input(); break;   
case 2:buble\_sort(); break;   
case 3:exchange\_sort(); break;   
case 4:selection\_sort(); break;   
case 5:insertion\_sort(); break;   
case 6:QuickSort(0,n-1); break;   
case 7:tampil(); break;   
case 8:AcakLagi(); break;   
  
}   
getch();   
}while(pil!=9);   
}   
</data[i]<<">