Macam-macam dan Contoh dari Sorting C++ - belajar C++

Macam-macam Sorting C++

Selection Sort

Tehnik pengurutan dengan cara pemilihan elemen atau proses kerja denganmemilih elemen data terkecil untuk kemudian dibandingkan & ditukarkan denganelemen pada data awal, dst sampai seluruh elemen shg akan menghasilkan pola data yg telah disort.

Prinsip kerja dari teknik ini adalah :

Pengecekan dimulai data ke-1 sampai dengan data ke-n

Tentukan bilangan dengan Index terkecil dari data bilangan tersebut

Tukar bilangan dengan Index terkecil tersebut dengan bilangan pertama ( I = 1 ) dari data bilangan tersebut

Lakukan langkah 2 dan 3 untuk bilangan berikutnya ( I= I+1 ) sampai didapatkan urutan yg optimal.

Baca juga : Pengertian dan Macam-macam Array C++

Contoh

#include<conio.h>

#include<stdio.h>

int main()

{

int i, j, iMin;

int n, Urut;

int Tmp, code;

int Arr[100];

printf("\nInputkan banyak data yang akan diurutkan : ");

scanf("%i", &n);

Urut = 1;

for(i = 0; i < n; i++)

{

printf("Masukan data ke %i : ", i + 1);

scanf("%i", &Arr[i]);

}

for(i = 0; i < n - 1; i++)

{

iMin = i;

for(j = Urut; j < n; j++)

{

if(Arr[j] < Arr[iMin])

{

iMin = j;

if(Arr[i] != Arr[iMin])

{

Tmp = Arr[i];

if(Arr[i] > Arr[iMin])

{

Arr[i] = Arr[iMin];

Arr[iMin] = Tmp;

}

}

}

}

Urut = Urut + 1;

}

printf("\nSetelah Pengurutan\n");

for(i = 0; i < n; i++)

{

printf("Elemen ke %i : %i\n", i + 1, Arr[i]);

}

getch();

}

Buble Sort

Prinsip Kerja dari Bubble Sort adalah :

Pengecekan mulai dari data ke-1 sampai data ke-n

Bandingkan data ke-n dengan data sebelumnya (n-1)

Jika lebih kecil maka pindahkan bilangan tersebut dengan bilangan yg ada didepannya ( sebelumnya ) satu persatu (n-1,n-2,n-3,....dst)

Jika lebih besar maka tidak terjadi pemindahan

Ulangi langkah 2 dan 3 s/d sort optimal.

Baca juga : Pengertian dan Macam-macam Preprocessor Directives C++

Contoh

#include<conio.h>

#include<stdio.h>

int main()

{

int i, j, iMin;

int n, Urut;

int Tmp, code;

int Arr[100];

printf("\nInputkan banyak data yang akan diurutkan : ");

scanf("%i", &n);

for(i = 0; i < n; i++)

{

printf("Masukan data ke %i : ", i + 1);

scanf("%i", &Arr[i]);

}

for(i = 1; i < n; i++)

{

for(j = 0; j < n - 1; j++)

{

if(Arr[j] > Arr[j + 1])

{

Tmp = Arr[j];

Arr[j] = Arr[j + 1];

Arr[j + 1] = Tmp;

}

}

}

printf("\nSetelah Pengurutan\n");

for(i = 0; i < n; i++)

{

printf("Elemen ke %i : %i\n", i + 1, Arr[i]);

}

getch();

}

Insertion Short

Prinsip dasar Insertion adalah secara berulang-ulang menyisipkan / memasukan setiap elemen. ke dlm posisinya / tempatnya yg benar.

Prinsip Kerja Insertion Sort adalah

Pengecekan mulai dari data ke-1 sampai data ke-n

Bandingkan data ke-I ( I = data ke-2 s/d data ke-n )

Bandingkan data ke-I tersebut dengan data sebelumnya (I-1), Jika lebih kecil maka data tersebut dapat disisipkan ke data awal sesuai dgn posisisi yg seharusnya

Lakukan langkah 2 dan 3 untuk bilangan berikutnya ( I= I+1 ) sampai didapatkan urutan yg optimal.

Baca juga : Macam-macam dan Contoh Fungsi Manipulasi String C++

Contoh

#include <conio.h>

#include <stdio.h>

int main( )

{

int i, j, iMin;

int n, Urut;

int Tmp, code;

int Arr[100];

printf("\nInputkan banyak data yang akan diurutkan : ");

scanf("%i", &n);

for(i = 0; i < n; i++)

{

printf("Masukan data ke %i : ", i + 1);

scanf("%i", &Arr[i]);

}

for(i = 1; i < n; i++)

{

Tmp = Arr[i];

j = i - 1;

while(Arr[j] >= Tmp && j > 0)

{

Arr[j + 1] = Arr[j];

j = j - 1;

}

if(Tmp >= Arr[j])

{

Arr[j + 1] = Tmp;

}

else

{

Arr[j + 1] = Arr[j];

Arr[j] = Tmp;

}

}

printf("\nSetelah Pengurutan\n");

for(i = 0; i < n; i++)

{

printf("Elemen ke %i : %i\n", i + 1, Arr[i]);

}

getch( );

}

Cukup sekian dari belajarcpp.com semoga artikel ini bisa bermanfaat untuk kita semua. Mohon maaf atas kekuranganya dan Terima Kasih atas dukungan dan kunjungan anda ke BelajarCPP. Have a nice day.

Last update : 4 Februari 2016