

Sorting Algorithm

Bubble Sort -- Selection Sort -- Insertion Sort -- Exchange Sort



Sorting Algorithm

Salah satu bagian penting dari struktur data adalah **sorting** atau pengurutan data. Ada banyak sekali Algoritma pengurutan data di dunia komputer, yaitu : **bubble sort**, **selection sort**, **insertion sort**, **exchange sort**, quick sort, merge sort, dll.

Setiap algoritma memiliki kelebihan dan kekurangan masing – masing, kita akan mempelajari cukup 4 algoritma saja, yaitu bubble sort, selection sort, insertion sort, dan exchange sort. Bila ingin mempelajari algoritma yang lain silakan berkunjung ke www.wikipedia.org .

Bubble Sort

Perhatikan kode berikut :

```
d:\bubble.cpp
#include "stdio.h"

void main() {

    int  A[5]={3,4,1,2,8},i,j,tampung;

    printf("Sebelum sorting : \n");
    for (i=0;i<5;i++){
        printf("%i ",A[i]);
    }

    for (i=1;i<5;i++){
        for (j=5-1;j>=i;j--){
            if (A[j]<A[j-1]){
                tampung=A[j];
                A[j]=A[j-1];
                A[j-1]=tampung;
            }
        }
    }

    printf("\n\nSetelah sorting : \n");
    for (i=0;i<5;i++){
        printf("%i ",A[i]);
    }
}
```

Task 1

1. Apa yang dilakukan program diatas ?
2. Lakukan untuk pengurutan sebaliknya!
3. Pengurutan diatas dilakukan dari depan atau belakang? Buat program untuk sebaliknya!
4. Buat program agar *user* bisa inputkan data secara dinamis, baik untuk *ascending*, maupun *descending*!

5. Tambahkan kode agar *user* dapat melihat proses pengurutan data!

Note : Ascending adalah pengurutan data dari terkecil menuju terbesar, sedangkan descending adalah pengurutan dari data terbesar menuju terkecil.

Exchange Sort

Perhatikan kode berikut :

```
d:\exchange.cpp
#include "stdio.h"

void main() {

    int  A[5]={3,4,1,2,8},i,j,tampung;

    printf("Sebelum sorting : \n");
    for (i=0;i<5;i++){
        printf("%i ",A[i]);
    }

    for (i=0;i<5-1;i++){
        for (j=i+1;j<5;j++){
            if (A[i]<A[j]){
                tampung=A[i];
                A[i]=A[j];
                A[j]=tampung;
            }
        }
    }

    printf("\n\nSetelah sorting : \n");
    for (i=0;i<5;i++){
        printf("%i ",A[i]);
    }
}
```

Task 2

1. Apa yang dilakukan program diatas ?
 2. Lakukan untuk pengurutan sebaliknya!
 3. Buat program agar *user* bisa inputkan data secara dinamis, baik untuk *ascending*, maupun *descending*!
 4. Tambahkan kode agar *user* dapat melihat proses pengurutan data!
-

5. Apa perbedaan antara exchange sort dan bubble sort!

Selection Sort

Perhatikan kode berikut :

```
d:\selectio.cpp
int  A[5]={3,4,1,2,8},i,j,tampung,pos;

printf("Sebelum sorting : \n");
for (i=0;i<5;i++){
    printf("%i ",A[i]);
}

for (i=0;i<5-1;i++){
    pos=i;
    for(j=i+1;j<5;j++){
        if (A[j]<A[pos]){
            pos=j;
        }
    }
    if (pos != i){
        tampung=A[pos];
        A[pos]=A[i];
        A[i]=tampung;
    }
}

printf("\n\nSetelah sorting : \n");
for (i=0;i<5;i++){
    printf("%i ",A[i]);
}
}
```

Task 3

1. Apa yang dilakukan program diatas ?
2. Lakukan untuk pengurutan sebaliknya!
3. Apa fungsi **pos**?
4. Buat program agar *user* bisa inputkan data secara dinamis, baik untuk *ascending*, maupun *descending*!

5. Tambahkan kode agar *user* dapat melihat proses pengurutan data!

Insertion Sort

Perhatikan kode berikut :

```
d:\insertio.cpp
#include "stdio.h"

void main() {

    int  A[5]={3,4,1,2,8},i,j,tampung;

    printf("Sebelum sorting : \n");
    for (i=0;i<5;i++){
        printf("%i ",A[i]);
    }

    for (i=1;i<5;i++){
        tampung=A[i];
        j=i-1;

        while (A[j]>tampung && j>=0){
            A[j+1] = A[j];
            j--;
        }
        A[j+1]=tampung;
    }

    printf("\n\nSetelah sorting : \n");
    for (i=0;i<5;i++){
        printf("%i ",A[i]);
    }
}
```

Task 4

1. Apa yang dilakukan program diatas ?
2. Lakukan untuk pengurutan sebaliknya!
3. Apa fungsi **tampung**?
4. Buat program agar *user* bisa inputkan data secara dinamis, baik untuk *ascending*, maupun *descending*!

5. Tambahkan kode agar *user* dapat melihat proses pengurutan data!

Task 5

```
#include "stdio.h"
#include "iostream.h"

typedef struct surat{
int kodepos;
char pengirim[20];
char penerima[20];
char kota[20];
};

void main(){

int i,j,tampung;
surat suratku[5];

for (i=0;i<5;i++){
printf("\nSurat %i\n",i+1);
printf("Kode pos : ");
scanf("%i",&suratku[i].kodepos);
printf("Pengirim : ");
cin.getline(suratku[i].pengirim,20);
printf("Penerima : ");
cin.getline(suratku[i].penerima,20);
printf("Kota : ");
cin.getline(suratku[i].kota,20);
}

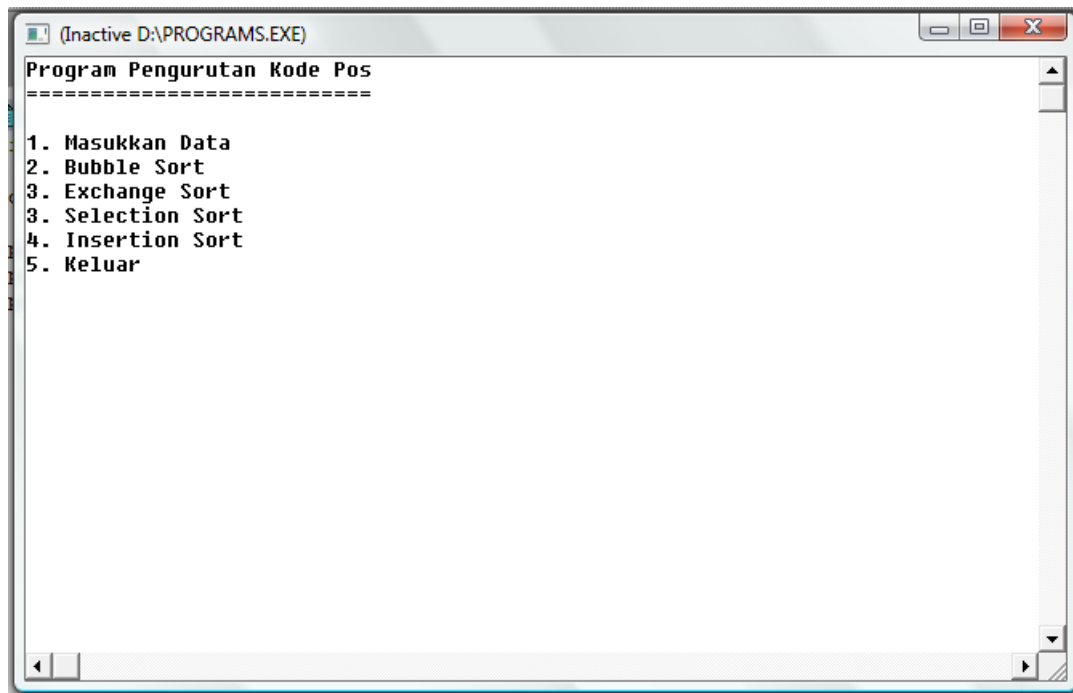
for (i=1;i<5;i++){
for(j=5-1;j>=i;j--){
if (suratku[j].kodepos<suratku[j-1].kodepos){
}
}

printf("\n\nSurat sesudah diurutkan\n");
for (i=0;i<5;i++){
printf("\n\nSurat %i\n",i+1);
printf("\nKode pos : %i ",suratku[i].kodepos);
printf("\nPengirim : %s ",suratku[i].pengirim);
printf("\nPenerima : %s ",suratku[i].penerima);
printf("\nKota : %s ",suratku[i].kota);
}
}
}
```

1. Apa yang menjadi kesalahan pada program diatas?
2. Coba anda perbaiki!

Task 6

1. Gabungkan 4 sorting diatas menjadi 1 program menu pilihan untuk mengurutkan data kode pos.



2. User pertama kali harus memasukkan data terlebih dahulu, apabila tidak ada data, maka pengurutan data tidak bisa berjalan.
 3. Data yang masuk hanya bisa berupa angka.
 4. Semua metode mengurutkan secara ascending.
 5. Pada tiap metode, user dapat melihat proses pengurutan data.
-

TAMBAHAN

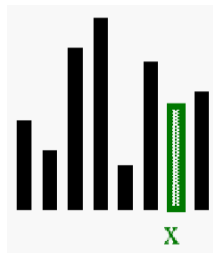
Quick Sort

Quick sort merupakan algoritma kedua tercepat setelah merge sort. Quick sort disebut juga *divide and conquer algorithm*. Quick sort memiliki average case $n \log n$

Langkah-langkah quick sort :

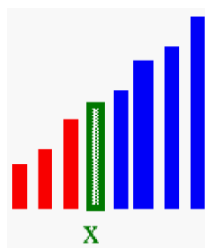
1. SELECT

Pilih sebuah element , elemen ini kita sebut *pivot*



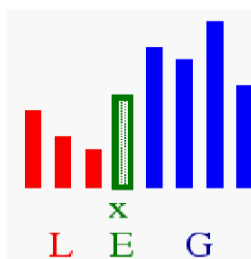
2. DIVIDE

Kemudian pisahkan data menjadi 2 bagian, bagian yang **lebih kecil** dari pivot dan bagian yang **lebih besar** dari pivot



3. RECUR and CONQUER

Kemudian 2 bagian tersebut diurutkan secara rekursif dengan memanggil fungsi itu sendiri, misal **quicksort()**



Perhatikan kode dibawah ini dan silakan di coba deprogram anda masing-masing:

```
#include <stdio.h>

void quicksort (int *a, int lo, int hi)
{
    int m = lo, n = hi, h;
    int x = a[(lo + hi) / 2];

    do {
        while (a[m] < x) {
            ++m;
        }
        while (a[n] > x) {
            --n;
        }
        if (m <= n) {
            h = a[m];
            a[m] = a[n];
            a[n] = h;
            ++m;
            --n;
        }
    } while (m <= n);

    if (lo < n) {
        quicksort(a, lo, n);
    }
    if (m < hi) {
        quicksort(a, m, hi);
    }
}

#define MAX 5

int main (void)
{
    int data[MAX] = { 3, 4, 1, 2, 8 };
    int x;

    printf("sebelum disortir\n");

    for (x = 0; x < MAX; ++x) {
        printf("%d ", data[x]);
        printf("\n");
        quicksort(data, 0, MAX - 1);
        printf("setelah disortir\n");
        for (x = 0; x < MAX; ++x) {
            printf("%d ", data[x]);
        }
        printf("\n");
    }

    return(0);
}
```

Task 7

1. Bagian mana pada program yang menentukan **pivot** ?
2. Bagian mana pada program yang memisahkan array menjadi 2 bagian ?
3. Bagian mana pada program yang melakukan fungsi rekursif ?

SOAL-SOAL

1. Soal Dan Pembahasan

- **Bubble Sort**

```
printf("\n");
for (i=10-1;i>=0;i--)
{
    for (j=0;j<i;j++)
    {
        for (k=0;k<10;k++)
        {
            printf("%i ",A[k]);
        }
        printf("\n");
        if (A[j]>A[j+1])
        {
            tampung=A[j];
            A[j]=A[j+1];
            A[j+1]=tampung;
        }
    }
    printf("\n");
}
```

- **Exchange Sort**

```
printf("\n");
for (i=0;i<9;i++)
{
    for (j=i+1;j<10;j++)
    {
        for (k=0;k<10;k++)
        {
            if (k==i) printf("| ");
            printf("%i",A[k]);
            if (k==i) printf("| ");
            printf(" ");
        }
        printf("\n");
        if (A[i]>A[j])
        {
            tampung=A[i];
            A[i]=A[j];
            A[j]=tampung;
        }
    }
    printf("\n");
}
```

- **Selection Sort**

```
printf("\n");
for (i=0; i<9; i++)
{
    pos=i;
    for (j=i+1; j<10; j++)
    {
        for (k=0; k<10; k++)
        {
            printf("%i ", A[k]);
        }
        printf("\tpos : %i\n", pos);
        if (A[pos]>A[j])
        {
            pos=j;
        }
    }
    printf("\n");
    if (i!=pos)
    {
        tempung=A[i];
        A[i]=A[pos];
        A[pos]=tempung;
    }
}
```

- **Insertion Sort**

```
for (i=0+1; i<10; i++)
{
    v=A[i];
    j=i-1;
    while (A[j]>v && j>=0)
    {
        A[j+1]=A[j];
        j--;
    }
    A[j+1]=v;
}
```

- Sebagai latihan keempat sorting di atas, lakukan:
 - Ubah code jadi sorting DESCENDING
 - Ubah code jadi inputan dinamis!(jumlah data tidak lagi 10)

2. Soal Tanpa Pembahasan

- **Search then Sort**

Aio buat program yang seru-seru!

Input nya sebuah deret angka yang mengandung tanda titik di dalam nya. Nah, kalo titik nya udah ketemu, terus semua angka di belakang nya di sorting jadi ascending. Nih contoh input output nya:

Input :

Banyak data : **10**

4 2 3 5 1 . 8 7 9 6

Output :

4 2 3 5 1 . 6 7 8 9

```
5 2 1 . 3 1 2 4 5 6 0
5 2 1 . 0 1 2 3 4 5 6
```

Note : tanda titik hanya akan ada 1 dalam 1 testcase.

- **2D Sort**

Buatlah program untuk melakukan sorting array 2 dimensi!

Input :

m : 3

n : 2

3 2 1

5 4 3

Output:

1 2 3

3 4 5

```
m : 3
n : 2
4 2 5
8 6 9
Output :
2 4 5
6 8 9
```

3. Soal Take Home (**Take Home Test maybe?)

Seorang yang amat sangat cerdas mendapatkan tugas yang menarik, dia di haruskan membuat sebuah program dengan menu:

a. **Duo Multi Sorting**

Menu pertama adlaah program “duo multi sorting” dengan contoh sorting:

Barisan angka yang belum di sorting: **4 2 5 1 8 7 10 9 6**

Barisan angka hasil multi sorting : **1 10 3 8 5 6 7 4 9 2**

```

DUO MULTI SORTING
Inputkan:
Jumlah bilangan : 10
Bilangan nya : 2 3 1 9 12 31 11 13 12 4
Output : 1 31 3 12 9 11 12 4 13 2

```

Kondisi-kondisi yang menjadi batasan adalah:

- Panjang data tidak akan pernah lebih dari 20
- Algoritma sorting yang digunakan **BEBAS!**

b. Ascending vs Descending

Menu kedua adalah program “Ascending vs Descending” dengan contoh sorting:

Barisan angka yang belum di sorting : **4 2 3 5 1 8 7 10 9 6 12 14 15 13 11**

Barisan angka yang sudah di sorting : **1 2 3 4 5 10 9 8 7 6 11 12 13 14 15**

```

ASCENDING US DESCENDING
Inputkan:
Jumlah bilangan : 10
Bilangan nya : 11 21 31 12 1 5 2 3 4 64
Output : 1 2 3 4 5 64 31 21 12 11

```

Kondisi-kondisi yang menjadi batasan adalah:

- Panjang data tidak akan pernah lebih dari 30
- Setiap 5 digit, terjadi perubahan sorting, **CEK POLA!**
- Algoritma sorting yang digunakan harus **EXCHANGE SORT!**

c. Single Word Sorting

Menu ketiga adalah program “Single Word Sorting” dengan contoh sorting :

Kalimat yang belum disorting : **danny emang tampan sekali**

Kalimat sesudah disorting : **adnny aegmn aamnpt aeikls**

```

SINGLE WORD SORTING
Inputkan:
danny emang tampan sekali yah
Output : adnny aegmn aamnpt aeikls ahy MENU

```

Kondisi-kondisi yang menjadi batasan adalah:

- Panjang kalimat tidak pernah lebih dari 30 suku(sudah termasuk spasi dan kawan” nya)
- Algoritma sorting yang digunakan harus **SELECTION SORT!**

Nah, karena otak nya sudah tidak lagi cerdas, dia meminta bantuan pada kita yang masih lebih cerdas daripada dia...:D

Daftar Pustaka

- http://en.wikipedia.org/wiki/Sorting_algorithm
- *lecturer.eepis-its.edu/~entin/Struktur%20Data/Minggu%2011/Minggu%2011%20Quick.pdf*
- <http://www.informatika.org/~rinaldi/Stmik/2005-2006/Makalah2006/MakalahStmik2006-27.pdf>
- <http://lecturer.ukdw.ac.id/anton/strukdat.php>

