Laporan Praktikum

Algoritma Struktur Data

Bubble Sort dan Shell Sort



Oleh :

Marcell Bintang Setiawan

3120521027

D3 Teknik Informatika PSDKU Lamongan

Politeknik Elektronika Negeri Surabaya

2020 / 2021

Percobaan 1 : Bubble sort secara ascending

1. Listing Program

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define MAX 20

void BubbleSort(int arr[])

{

int i, j, temp;

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

{

for (j = 0; j < MAX - i - 1; j++)

{

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

{

temp = arr[j + 1];

arr[j + 1] = arr[j];

arr[j] = temp;

}

}

}

}

int main()

{

int data\_awal[MAX], data\_urut[MAX];

int i;

long k1, k2;

printf("Sebelum pengurutan : \n");

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

{

srand(time(NULL) \* (i + 1));

data\_awal[i] = rand() % 100 + 1;

printf("%d ", data\_awal[i]);

}

printf("\nSetelah pengurutan : \n");

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

data\_urut[i] = data\_awal[i];

BubbleSort(data\_urut);

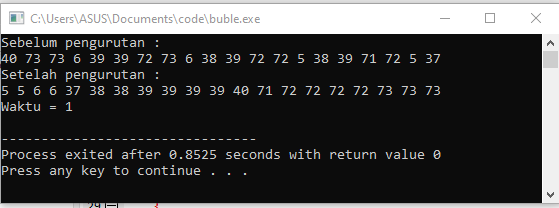
for (i = 0; i < MAX; i++)

printf("%d ", data\_urut[i]);

printf("\nWaktu = %ld\n", k2 - k1);

}

1. Output



1. Analisis

Percobaan diatas merupakan pengurutan sebuah data bilangan dengan menggunakan metode bubble sort secara ascending. Menggunakan fungsi srand untuk membuat data bilangan acak. Kemudian array acak tersebut (data\_awal) masuk ke dalam array baru (data\_urut) untuk dilakukan sorting bubble. Setelah array selesai diurutkan, cetak array tersebut menggunakan looping.

Percobaan 2 : Bubble sort secara ascending dengan flag

1. Listing Program

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#define MAX 20

void BubbleSortFlag(int arr[])

{

int i=0, j, temp;

bool did\_swap=true;

while(i < MAX-1 && did\_swap){

for(j=0; j<MAX-i-1; j++){

did\_swap=false;

if(arr[j] > arr[j+1]){

temp = arr[j+1];

arr[j+1] = arr[j];

arr[j]= temp;

did\_swap=true;

}

}

i++;

}

}

main()

{

int data\_awal[MAX], data\_urut[MAX];

int i;

long k1, k2;

printf("Sebelum pengurutan : \n");

for(i=0; i<MAX; i++){

srand(time(NULL) \* (i+1));

data\_awal[i] = rand() % 100 + 1;

printf("%d ", data\_awal[i]);

}

printf("\nSetelah pengurutan : \n");

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

data\_urut[i] = data\_awal[i];

BubbleSortFlag(data\_urut);

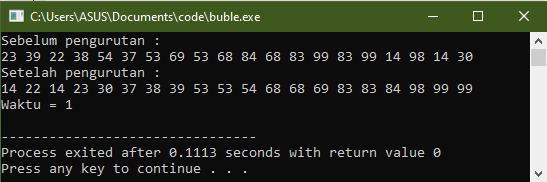
for(i=0; i<MAX; i++)

printf("%d ", data\_urut[i]);

printf("\nWaktu = %ld\n", k2-k1);

}

1. Output



1. Analisis

Fungsi dari flag disini adalah untuk mempermudah algoritma sort bubble. Flag dapat digunakan untuk mengecek apakah elemen ditukar atau tidak dalam loop for. Jika pada tidak terjadi pertukaran, itu berarti array telah diurutkan dan dapat keluar dari loop for.

E. LATIHAN

1.Dari percobaan 1 tambahkan fungsi untuk melakukan pengurutan bubble sort secara descending.

a. Listing Program

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#define MAX 20

void BubbleSortAsc(int arr[])

{

int x, y, temp;

for(x=0; x<MAX-1; x++){

for(y=0; y<MAX-x-1; y++){

if(arr[y] > arr[y+1]){

temp = arr[y+1];

arr[y+1] = arr[y];

arr[y]= temp;

}

}

}

}

void BubbleSortDesc(int arr[])

{

int x, y, temp;

for(x=0; x<MAX-1; x++){

for(y=0; y<MAX-x-1; y++){

if(arr[y] < arr[y+1]){

temp = arr[y+1];

arr[y+1] = arr[y];

arr[y]= temp;

}

}

}

}

int main()

{

int data\_awal[MAX], data\_urut[MAX];

int x, pil;

long k1, k2;

printf("Sebelum pengurutan : \n");

for(x=0; x<MAX; x++){

srand(time(NULL) \* (x+1));

data\_awal[x] = rand() % 100 + 1;

printf("%d ", data\_awal[x]);

}

printf("\n");

do{

printf("1. Ascending\n2. Descending\n3. Keluar\n" );

printf("\nMasukkan Pilihan : \n");

scanf("%d",&pil);

switch(pil){

case 1 :

printf("\nSetelah pengurutan : \n");

for(x=0; x<MAX; x++)

data\_urut[x] = data\_awal[x];

BubbleSortAsc(data\_urut);

for(x=0; x<MAX; x++)

printf("%d ", data\_urut[x]);

printf("\n");

break;

case 2 :

printf("\nSetelah pengurutan : \n");

for(x=0; x<MAX; x++)

data\_urut[x] = data\_awal[x];

BubbleSortDesc(data\_urut);

for(x=0; x<MAX; x++)

printf("%d ", data\_urut[x]);

printf("\n");

break;

case 3 :

exit(0);

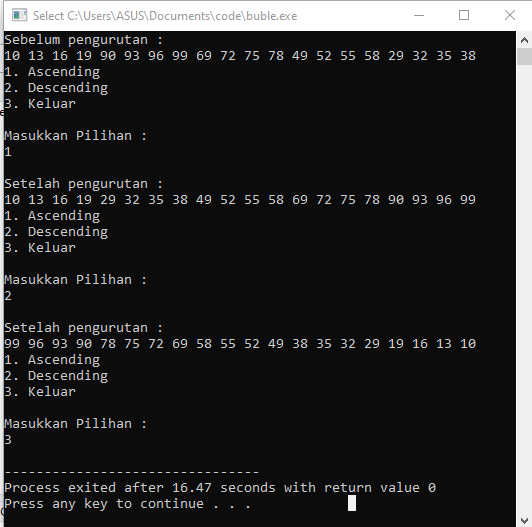
}

}while( pil !=3 );

return (0);

}

b. Output



c. Analisis

untuk membuat program sperti di atas kita hanya perlu memberikan beberapa modifikasi, untuk pengurutan bubble sort di nomor satu kita masukkan ke dalam fungsi BubbleSortAsc dan menambahkan pula fungsi BubbleSortDesc yang isinya sama namun pengulangan for – nya kita ubah menjadi iterasi ke bawah atau berkurang. Kemudian fungsi – fungsi tersebut kita panggil dengan menu yang di buat dengan do while

2. Dari percobaan 2 tambahkan fungsi untuk melakukan pengurutan bubble sort

dengan flag secara descending.

1. Listing Program

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define MAX 20

void BubbleSortFlagAsc(int arr[])

{

int x = 0, y, temp;

bool did\_swap = true;

while (x < MAX - 1 && did\_swap)

{

for (y = 0; y < MAX - x - 1; y++)

{

did\_swap = false;

if (arr[y] > arr[y + 1])

{

temp = arr[y + 1];

arr[y + 1] = arr[y];

arr[y] = temp;

did\_swap = true;

}

}

x++;

}

}

void BubbleSortFlagDesc(int arr[])

{

int x = 0, y, temp;

bool did\_swap = true;

while (x < MAX - 1 && did\_swap)

{

for (y = 0; y < MAX - x - 1; y++)

{

did\_swap = false;

if (arr[x] < arr[y + 1])

{

temp = arr[y + 1];

arr[y + 1] = arr[y];

arr[y] = temp;

did\_swap = true;

}

}

x++;

}

}

main()

{

int data\_awal[MAX], data\_urut[MAX];

int p, pil;

long k1, k2;

printf("Sebelum pengurutan : \n");

for (p = 0; p < MAX; p++)

{

srand(time(NULL) \* (p + 1));

data\_awal[p] = rand() % 100 + 1;

printf("%d ", data\_awal[p]);

}

printf("\n");

do

{

printf("1. Ascending\n2. descending\n3. Exit\n");

printf("\nMasukkan Pilihan : \n");

scanf("%d", &pil);

switch (pil)

{

case 1:

printf("\nSetelah pengurutan : \n");

for (p = 0; p < MAX; p++)

data\_urut[p] = data\_awal[p];

BubbleSortFlagAsc(data\_urut);

for (p = 0; p < MAX; p++)

printf("%d ", data\_urut[p]);

printf("\n");

break;

case 2:

printf("\nSetelah pengurutan : \n");

for (p = 0; p < MAX; p++)

data\_urut[p] = data\_awal[p];

BubbleSortFlagDesc(data\_urut);

for (p = 0; p < MAX; p++)

printf("%d ", data\_urut[p]);

printf("\n");

break;

case 3:

exit(0);

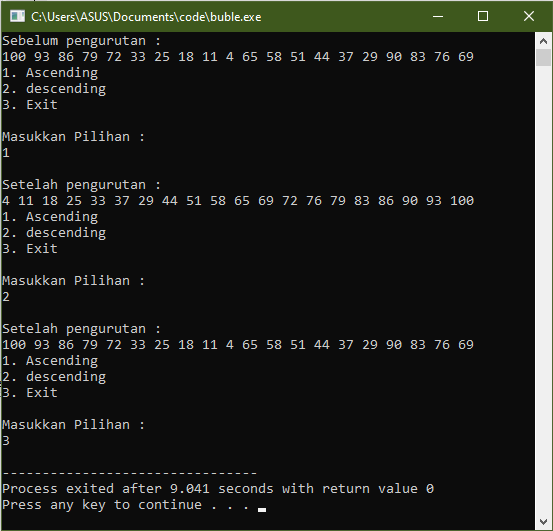
}

} while (pil != 3);

return (0);

}

1. Output



1. Analisis

Program ini sama seperti program sebelumnya, disini metode yang digunakan adalah bubble sort dengan flag yang sudah dipraktekkkan di percobaan nomor 2, berangkat dari program nomor 2, metode tersebut kita masukkan ke dalam fungsi BubbleSortFlagAsc dan fungsi BubbleSortFlagDesc yang isinya berupa metode bubble sort dengan flag secara ascending dan descending. Kemudia fungsi – fungsi tersebut dipanggil dalam bentu menu menggunakan do while dan ditampilkan seoerti pada gambar.