NAMA : Michael Raphael Ben Ariyanto

NIM : 672021028

1.Algoritma adalah urutan Langkah untuk menyelesaikan masalah secara sistematis dan logis.algoritma menawarkan suatu metode dalam menyelesaikan sebuah permasalahan.algoritma diartikan sebagai urutan Langkah dalam menyelesaikan masalah secara sistematis dan logis.pendekatan sistematis dan logis tersebut,menjadikan proses penyelesaian masalah terjaga kebenarannya karena algoritma hendaklah benar agar dapat menghasilkan solusi yang benar.

Array adalah kumpulan data bertipe sama dan menghasilkan nama yang sama pula.antara satu variable dengan variable lain di dalam array berdasarkan”subscript”.subscript berupa bilangan di dalam kurung (…).

### 2. Array

Array adalah sebuah bentuk dari kumpulan data yang dimana akan terstruktur dengan memiliki sebuah nama variabel yang dimana sama dan juga tipe data yang dianggap sama akan tetapi memiliki sebuah indeks maupun alamat yang dimana akan berbeda-beda. Pada umumnya setiap variabel sendiri akan memiliki berbagai macam bentuk data yang dianggap sama maupun berbeda-beda akan tetapi terdapat pula sebuah data yang haruslah sama. Urutan daripada indeks array sendiri akan dimulai dari hitungan 0,1,2 dan dilakukan seterusnya. Pada tipe data variabel array sendiri pada umumnya adalah dengan menggunakan char, string, int, float dan juga double.

Contoh array :

**#include <stdio.h>**

**#include <conio.h>**

**#include <string.h>**

**#define maks 3**

**struct TMhs**

**{**

**char NIM[9];**

**char Nama[21];**

**int NilaiUTS,NilaiUAS,NilaiQuis;**

**float NilaiAkhir;**

**char index;**

**};**

**main()**

**{**

**TMhs mhs[maks]; // array struct**

**int i;**

**for(i=0;i<maks;i++)**

**{**

**printf("Pengisian Data Mahasiswa Ke-%i\n",i+1);**

**printf("NIM        : ");fflush(stdin);gets(mhs[i].NIM);**

**printf("NAMA       : ");fflush(stdin);gets(mhs[i].Nama);**

**printf("Nilai QUIZ : ");scanf("%d",&mhs[i].NilaiQuis);**

**printf("Nilai UTS  : ");scanf("%d",&mhs[i].NilaiUTS);**

**printf("Nilai UTAS : ");scanf("%d",&mhs[i].NilaiUAS);**

**mhs[i].NilaiAkhir=0.2\*mhs[i].NilaiQuis+0.3\*mhs[i].NilaiUTS+0.5\*mhs[i].NilaiUAS;**

**if(mhs[i].NilaiAkhir>=80) mhs[i].index='A';else**

**if(mhs[i].NilaiAkhir>=60) mhs[i].index='B';else**

**if(mhs[i].NilaiAkhir>=40) mhs[i].index='C';else**

**if(mhs[i].NilaiAkhir>=20) mhs[i].index='D';else**

**if(mhs[i].NilaiAkhir>=0)  mhs[i].index='E';**

**};**

**clrscr();**

**printf("Data yang telah dimasukan adalah : \n");**

**printf("----------------------------------------------------------------------\n");**

**printf("|    NIM     |       NAMA         | QUIS | UTS | UAS |  N A  | INDEX |\n");**

**printf("----------------------------------------------------------------------\n");**

**for(i=0;i<maks;i++)**

**{**

**printf("| %-8s | %-20s |  %3i | %3i | %3i | %6.2f |   %c   |\n",**

**mhs[i].NIM,mhs[i].Nama,mhs[i].NilaiQuis,mhs[i].NilaiUTS,**

**mhs[i].NilaiUAS,mhs[i].NilaiAkhir,mhs[i].index);**

**}**

**printf("----------------------------------------------------------------------\n");**

**getch();**

**return 0;**

**}**

**Struct**

Struct adalah sebuah bentuk dari kumpulan data yang dimana akan memiliki berbagai macam bentuk tipe data yang akan berbeda. Pada secara pendeklarasiannya, sebuah struck akan dianggap berbeda dengan array yang dimana hanya akan memiliki sebanyak satu buah tipe data yang digunakan untuk setiap kumpulannya. Struct sendiri akan digunakan apabila pada data yang ingin dilakukan pengelompokan akan memiliki tipe data yang berbeda-beda.

Contoh struct :

#include <stdio.h>

**struct** Siswa {

**char**  nama[50];

**char**  sekolah[50];

  unsigned **int**  uangSaku;

};

**int** main(**void**)

{

**struct** Siswa siswa01;

**strcpy**(siswa01.nama, "Andi Permana Nugroho");

**strcpy**(siswa01.sekolah, "SMA 1 Lumut Ijo");

  siswa01.uangSaku = 10000;

**printf**("%s bersekolah di %s ", siswa01.nama, siswa01.sekolah);

**printf**("dengan uang saku %i per hari\n", siswa01.uangSaku);

**return** 0;

}

**3**. Sequential search adalah salah satu teknik pencarian data dimana data yang akan dicari dicocokkan / disesuaikan dengan data yang ada pada masing-masing indeks mulai dari indeks ke - 0 hingga indeks n - 1 (jumlah - 1 karena indeks array dimulai dari 0)

Contoh program

#include <stdio.h>

int main(){

int banyakData, i, cari;

int ketemu = 0;

printf("Pencarian Data dengan Algoritma Sequential Search\n\n");

printf("Input Banyak Data\t: ");

scanf("%d", &banyakData);

int data[banyakData];

printf("Input Kumpulan Data\t: ");

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

{

scanf("%d", &data[i]);

}

printf("Input Data yang Dicari : ");

scanf("%d", &cari);

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

{

if(cari == data[i])

{

ketemu = 1;

printf("Hasil Pencarian \t: Data Ditemukan pada Indeks ke - %d\n", i);

break;

}

}

if(ketemu == 0)

printf("Hasil Pencarian \t : Data Tidak Ditemukan\n");

getch();

return 0;

}

Penjelasannya menggunakan algoritma sequential search

membuat sebuah array sepanjang / sebanyak n (int data[n]) yang dimana data yang diinput oleh user nantinya akan disimpan pada tiap-tiap indeks array (indeks ke - 0 hingga indeks n - 1).

Kemudian saya meminta inputan lagi (scanf cari) agar user dapat menginput data yang akan dicari. Lalu saya menggunakan perulangan sebanyak n data untuk mencari data yang akan dicari dengan cara membandingkan / mencocokkan data yang dicari dengan data yang ada pada masing-masing indeks array.

Sebelumnya saya telah menginsiasikan variabel ketemu = 0 (data tidak ditemukan) terlebih dahulu yang dimana variabel ini nantinya akan saya gunakan untuk melakukan pencetakan keluaran. Jika data yang dicari ada pada array / ditemukan, maka nilai ketemu akan berubah menjadi 1, kemudian dilanjutkan dengan melakukan pencetakan keterangan posisi indeks data, lalu program akan berhenti karena adanya fungsi break.

Jika data belum / tidak ditemukan, maka pencarian akan dilakukan hingga indeks terakhir. Apabila semua data telah diperiksa / dicocokkan dan data belum ketemu, maka nilai ketemu akan tetap = 0 yang mana jika ketemu = 0 maka akan dicetak keluaran data tidak ditemukan.

4.Perbedaan Bubble sort, Insertion sort , Selection sort

**Bubble Sort** adalah metode pengurutan algoritma dengan cara melakukan penukaran data secara terus menerus sampai bisa dipastikan dalam suatu iterasi tertentu tidak ada lagi perubahan/penukaran. Algoritma ini menggunakan perbandingan dalam operasi antar elemennya.

Contoh Bubble Sort :

#include <stdio.h>

void bubbleSort(int arr[], int n){

int a, m, tmp;

for(a = 0; a < n; a++){

for(m=0; m < n-a-1; m ++){

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

tmp = arr[m];

arr[m] = arr[m+1];

arr[m+1] = tmp;

}

}

}

}

int main() {

int array[100], x, a, m;

printf("Masukkan banyaknya elemen: ");

scanf("%d", &x);

printf("Masukkan angka: \n");

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

scanf("%d", &array[a]);

}

bubbleSort(array, x);

printf("Hasil pengurutan adalah :\n");

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

printf("%d ", array[a]);

}

printf("\n");

}

**Insertion Sort** merupakan salah satu algoritma pengurutan yang dimana melakukan perbandingkan elemen array dengan elemen array dengan memiliki nilai tertinggi dalam array. Jika elemen array pembanding memiliki nilai lebih tinggi, maka elemen array tersebut akan tukar posisi dengan elemen array yang memiliki nilai lebih rendah dan elemen array yang memiliki nilai lebih rendah akan menjadi elemen pembanding sampai posisi yang tepat ditemukan. Untuk mengetahui proses pengurutan ada dibagian kesimpulan.

Contoh insertion :

#include <stdio.h>

int angka[7]={1,7,5,4,3,2,6};

void main(){

for(int x=1; x<7; x++){

for(int y=x; y>0 && angka[y]<angka[y-1]; y--){

int temp=angka[y-1];

angka[y-1]=angka[y];

angka[y]=temp;

}

}

for(int x=0; x<7; x++){

printf("%d ", angka[x]);

}

getchar();

}

**Selection Sort** merupakan salah satu algoritma sorting yang populer dan cukup sering digunakan. Sesuai namanya, algoritma sorting yang satu ini akan melakukan penyeleksian data dari kumpulan data yang belum di sort lalu memasukkannya ke dalam bagian yang sudah di sort.

Contoh selection :

#include <stdio.h>

int angka[7]={1,7,5,4,3,2,6};

void main(){

for(int x=1; x<7; x++){

for(int y=x; y>0 && angka[y]<angka[y-1]; y--){

int temp=angka[y-1];

angka[y-1]=angka[y];

angka[y]=temp;

}

}

for(int x=0; x<7; x++){

printf("%d ", angka[x]);

}

getchar();

}

5. #include <stdio.h>

#include <conio.h>

#include <string.h>

#define maks 3

struct TMhs

{

char NIM[9];

char Nama[21];

int NilaiQuis,NilaiTTS,NilaiTAS;

float NilaiAkhir;

char index;

};

main()

{

TMhs mhs[maks]; // array struct

int i;

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

{

printf("Pengisian Data Mahasiswa Ke-%i\n",i+1);

printf("NIM : ");fflush(stdin);gets(mhs[i].NIM);

printf("NAMA : ");fflush(stdin);gets(mhs[i].Nama);

printf("Nilai QUIS : ");scanf("%d",&mhs[i].NilaiQuis);

printf("Nilai TTS : ");scanf("%d",&mhs[i].NilaiTTS);

printf("Nilai TAS : ");scanf("%d",&mhs[i].NilaiTAS);

mhs[i].NilaiAkhir=0.2\*mhs[i].NilaiQuis+0.3\*mhs[i].NilaiTTS+0.5\*mhs[i].NilaiTAS;

if(mhs[i].NilaiAkhir>=80) mhs[i].index='A';else

if(mhs[i].NilaiAkhir>=60) mhs[i].index='B';else

if(mhs[i].NilaiAkhir>=50) mhs[i].index='E';else

if(mhs[i].NilaiAkhir>=20) mhs[i].index='F';else

if(mhs[i].NilaiAkhir>=0) mhs[i].index='G';

};

printf("Data yang telah dimasukan adalah : \n");

printf("----------------------------------------------------------------------\n");

printf("| NIM | NAMA | QUIS | TTS | TAS | N A | INDEX |\n");

printf("----------------------------------------------------------------------\n");

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

{

printf("| %-8s | %-20s | %3i | %3i | %3i | %6.2f | %c |\n",

mhs[i].NIM,mhs[i].Nama,mhs[i].NilaiQuis,mhs[i].NilaiTTS,

mhs[i].NilaiTAS,mhs[i].NilaiAkhir,mhs[i].index);

}

printf("----------------------------------------------------------------------\n");

getch();

return 0;

}

