**STRUKTUR DATA**

**(Tugas 4 Maret 2024)**

****

**NAMA: Bagas Diatama Wardoyo**

**NPM: 140810230061**

**Dikumpulkan tanggal :**

**4 Maret 2024**

**Universitas Padjadjaran**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**Program Studi S-1 Teknik Informatika**

**2024**

1. **Source Code**

/\* Nama program : soal\_1

   Nama         : Bagas Diatama Wardoyo

   NPM          : 140810230061

   Tanggal buat : 01/03/2024

   Deskripsi    : Pengolahan Data mahasiswa menggunakan struct

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

using namespace std;

//Struct mahasiswa berisi nama, npm, status, dan nilai

struct Mahasiswa

{

    string npm, nama, status;

    int nilai;

};

//Menginisialkan struct Mahasiswa sebagai dataMhs[10]

typedef Mahasiswa dataMhs[10];

//Menginisiasi berapa banyak data yang akan di olah

void banyakData(int &n)

{

    cout << "Banyak Data: ";

    cin >> n;

}

//Fungsu untuk menginput data mahasiswa

void input(dataMhs mhs, int n)

{

    for (int i = 0; i < n; i++)

    {

        cout << "Masukkan data mahasiswa ke- " << (i + 1) << endl;

        cout << "NPM\t: ";

        cin >> mhs[i].npm; //input npm

        cin.ignore();

        cout << "Nama\t: ";

        getline(cin, mhs[i].nama);// input nama

        cout << "Nilai\t: ";

        cin >> mhs[i].nilai;// input nilai

    }

}

//Fungsi yang menentukan status kelulusan mahasiswa berdasarkan nilai

void MutuStatus(dataMhs mhs, int n)

{

    for (int i = 0; i < n; i++)

    {

        if (mhs[i].nilai >= 60)

        {

            mhs[i].status = "Lulus";

        }

        else

        {

            mhs[i].status = "Gagal";

        }

    }

}

//Fungsi menukar index data mahasiswa sesuai npm

void swap(Mahasiswa &a, Mahasiswa &b)

{

    Mahasiswa temp = a;

    a = b;

    b = temp;

}

//Fungsi memngurutkan data mahasiswa sesuai npm

void ascendingSort(dataMhs a, int n)

{

    for (int i = n - 1; i > 0; i--)

    {

        for (int j = 0; j < i; j++)

        {

            if (a[j].npm > a[j + 1].npm)

            {

                swap(a[j], a[j + 1]);

            }

        }

    }

}

//Fungsi untuk menghitung rata rata dari seluruh nilai mahasiswa

void average(dataMhs a, int n, float &avg) {

    float total = 0;

    for (int i = 0; i < n; i++) {

        total += a[i].nilai;

    }

    avg = total / n;

}

//Fungsi unutk menentukan nilai maksimal dari seluruh data nilai mahasiswa

void findMax(dataMhs a, int n, float &maxValue) {

    maxValue = a[0].nilai;

    for (int i = 1; i < n - 1; i++) {

        if (a[i].nilai > maxValue) {

            maxValue = a[i].nilai;

        }

    }

}

//Fungsi untuk mencetak seluruh data

void cetakMahasiswa(dataMhs mhs, int n){

    cout << "\nDaftar Nilai Mahasiswa Struktur Data " << endl;

    cout << "----------------------------------------------------------------------------" << endl;

    cout << "No\t " << "NMP\t" << "Nama\t" << "Nilai\t" << "Status\_Lulus\t" << endl;

    cout << "----------------------------------------------------------------------------" << endl;

    for (int i = 0; i < n; i++) {

        cout << i + 1 << "\t" << mhs[i].npm << "\t" << mhs[i].nama << "\t" << mhs[i].nilai << "\t" << mhs[i].status;

        cout << endl;

    }

    cout << "----------------------------------------------------------------------------" << endl;

}

//Fungsi untuk mencari data mahasiswa berdasarkan NPM

void cariData(string npm, dataMhs data, int n){

    for (int  i = 0; i < n; i++)

    {

        if (npm == data[i].npm)

        {

            cout << "Ditemukan NPM = " << npm << ", Nama = " << data[i].nama << ", Nilai = " << data[i].nilai << ", Status = " << data[i].status << endl;

        }else{

            cout << "Data Tidak Ditemukkan";

        }

    }

}

int main()

{

    dataMhs mhs; //deklarasi struct sebagai mhs

    int n; //deklarasi n sebagai banyak data, dan max sebagai nilai maksimum

    float avg, max; //deklarasi avg sebagai rata rata seluruh nilai

    string npm; //deklarasi npm sebagai input npm

    banyakData(n);

    input(mhs, n);

    MutuStatus(mhs, n);

    ascendingSort(mhs, n);

    cetakMahasiswa(mhs, n);

    average(mhs, n, avg);

    findMax(mhs, n, max);

    cout << "Nilai Rata rata    : " << avg << endl;

    cout << "Nilai Tertinggi    : " << max << endl;

    cout << "Pencarian: " << endl;

    cout << "Masukkan NPM Pencarian : "; cin >> npm;

    cariData(npm, mhs, n);

    return 0;

}

1. Hasil Running

