

STRUKTUR DATA

Tugas_3



NAMA: Bagas Diatama Wardoyo

NPM: 140810230061

Dikumpulkan tanggal :

22 Maret 2024

Universitas Padjadjaran

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

Program Studi S-1 Teknik Informatika

2024

1. Buatlah program modular untuk mencari jumlah dari setiap baris dan kolom suatu matriks. Hasil penjumlahan disimpan dalam suatu array 1 dimensi

Source Code

```
/* Nama program : soal_2
   Nama      : Bagas Diatama Wardoyo
   NPM      : 140810230061
   Tanggal buat : 14/03/2024
   Deskripsi  : program modular untuk mencari jumlah dari setiap baris
dan kolom suatu matriks.
*****
*****/

#include <iostream>

using namespace std;

void inputMatriks(int arr[][3]);
void sumMatriks_baris(int arr[][3]);
void sumMatriks_kolom(int arr[][3]);

int main()
{
    int matriks[3][3];
    inputMatriks(matriks);
    cout << endl;
    sumMatriks_baris(matriks);
    cout << endl;
    sumMatriks_kolom(matriks);

    return 0;
}

//Fungsi input matriks
```

```

void inputMatriks(int arr[][3])
{
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            cin >> arr[i][j];
        }
    }
}

//Fungsi untuk menjumlahkan setiap baris matriks
void sumMatriks_baris(int arr[][3])
{
    int sumBaris[] {0, 0, 0};
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            sumBaris[i] += arr[i][j];
        }
    }
    for (int n : sumBaris)
    {
        cout << n << endl;
    }
}

//Fungsi untuk menjumlahkan setiap kolom matriks
void sumMatriks_kolom(int arr[][3])
{
    int sumBaris[] {0, 0, 0};
    for (int i = 0; i < 3; i++)
    {

```

```

        for (int j = 0; j < 3; j++)
        {
            sumBaris[i] += arr[j][i];
        }
    }
    for (int n : sumBaris)
    {
        cout << n << endl;
    }
}

```

Hasil Running

```

1 0 4
5 2 1
2 1 2

5
8
5

8
3
7

```

2. Buatlah program modular untuk melakukan penjumlahan dan perkalian 2 buah matriks

Source Code

```

/* Nama program : soal_3
   Nama          : Bagus Diatama Wardoyo
   NPM           : 140810230061
   Tanggal buat  : 14/03/2024
   Deskripsi     : program modular untuk melakukan
penjumlahan dan perkalian 2 buah matriks.
*****
*****/

#include <iostream>

void inputArray(int arr_a[][3], int arr_b[][3]);

```

```

void perkalianMatriks(int arr_a[][3], int arr_b[][3]);
void penjumlahanMatriks(int arr_a[][3], int arr_b[][3]);
void cetakMatriks(int arr[][3]);

int main()
{
    int arr_a[3][3], arr_b[3][3];
    inputArray(arr_a, arr_b);
    perkalianMatriks(arr_a, arr_b);
    penjumlahanMatriks(arr_a, arr_b);

    return 0;
}

//Fungsi input nilai matriks
void inputArray(int arr_a[][3], int arr_b[][3])
{
    std::cout << "Masukkan array_a (3x3): " << '\n';
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            std::cin >> arr_a[i][j];
        }
    }
    std::cout << "Masukkan array_b (3x3): " << '\n';
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            std::cin >> arr_b[i][j];
        }
    }
}

//Fungsi mencetak matriks
void cetakMatriks(int arr[][3], int operasi)
{
    if (operasi == 1)
    {

```

```

        std::cout << "Hasil Perkalian: " << '\n';
    }
    else if (operasi == 2)
    {
        std::cout << "Hasil penjumlahan: " << '\n';
    }
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            std::cout << arr[i][j] << " ";
        }
        std::cout << std::endl;
    }
}

//Fungsi perkalian matriks
void perkalianMatriks(int arr_a[][3], int arr_b[][3])
{
    int result[3][3];
    int sum = 0;
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            for (int k = 0; k < 3; k++)
            {
                sum += arr_a[i][k] * arr_b[k][j];
            }
            result[i][j] = sum;
            sum = 0;
        }
    }
    cetakMatriks(result, 1);
}

//Fungsi penjumlahan matriks
void penjumlahanMatriks(int arr_a[][3], int arr_b[][3])
{
    int result[3][3];

```

```

for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        result[i][j] = arr_a[i][j] + arr_b[i][j];
    }
}

cetakMatriks(result, 2);
}

```

Hasil Running

```

Masukkan array_a (3x3):
5 8 4
1 2 3
9 6 3
Masukkan array_b (3x3):
1 4 7
2 5 8
3 6 9
Hasil Perkalian:
33 84 135
14 32 50
30 84 138
Hasil penjumlahan:
6 12 11
3 7 11
12 12 12

```

3. Buatlah program modular untuk mengolah suatu kumpulan data bertipe Array of record Mahasiswa (npm, nama, nilai) Fungsi yang dibuat antara lain : inputData, rata-rata, nilaiTertinggi, nilaiTerendah, Sorting, Searching, dan outputData Kunci searching/sorting adalah NPM

Source Code

```

/* Nama program : soal_4
   Nama          : Bagus Diatama Wardoyo
   NPM           : 140810230061
   Tanggal buat  : 14/03/2024
   Deskripsi     : Pengolahan Data mahasiswa menggunakan
struct
*****
*****/
#include <iostream>

```

```

using namespace std;

//Struct mahasiswa berisi nama, npm, dan nilai
struct Mahasiswa
{
    string npm, nama;
    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;
}

//Fungsi 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 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; 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\tNMP\tNama\tNilai\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;
        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 <<
endl;
        }
    }
}

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);
    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;
}

```

Hasil Running

```

Banyak Data: 3
Masukkan data mahasiswa ke- 1
NPM      : 008
Nama     : Bagus
Nilai    : 85
Masukkan data mahasiswa ke- 2
NPM      : 004
Nama     : Diatama
Nilai    : 48
Masukkan data mahasiswa ke- 3
NPM      : 001
Nama     : Wardoyo
Nilai    : 100

Daftar Nilai Mahasiswa Struktur Data
-----
No      NPM   Nama   Nilai
-----
1       001   Wardoyo 100
2       004   Diatama 48
3       008   Bagus   85
-----
Nilai Rata rata      : 77.6667
Nilai Tertinggi      : 100
-----Pencarian-----
Masukkan NPM Pencarian : 001
Ditemukan NPM = 001, Nama = Wardoyo, Nilai = 100

```

4. Buatlah program untuk mengelola selisih waktu. Fungsi : input, mencari selisihWaktu(fungsi) dan output

Source Code

```
/* Nama program : soal_5
   Nama          : Bagas Diatama Wardoyo
   NPM           : 140810230061
   Tanggal buat  : 14/03/2024
   Deskripsi     : Buatlah program untuk mengelola selisih
waktu
*****
*****/
#include <iostream>
using namespace std;

// Struktur untuk merepresentasikan waktu
struct Waktu {
    int jam, menit, detik;
};

// Prototipe fungsi
void masukkanWaktu(Waktu &waktu);
void hitungSelisih(Waktu awal, Waktu akhir, Waktu
&selisih);
void cetakWaktu(Waktu waktu);

int main() {
    Waktu waktuA, waktuB, selisih;

    cout << "Masukkan Waktu A: ";
    masukkanWaktu(waktuA);

    cout << "Masukkan Waktu B: ";
    masukkanWaktu(waktuB);

    hitungSelisih(waktuA, waktuB, selisih);

    cout << "Selisih Waktu: ";
    cetakWaktu(selisih);
}
```

```

        return 0;
    }

    // Fungsi untuk mendapatkan input waktu dari pengguna
    void masukkanWaktu(Waktu &waktu) {
        cout << "Jam: ";
        cin >> waktu.jam;
        cout << "Menit: ";
        cin >> waktu.menit;
        cout << "Detik: ";
        cin >> waktu.detik;
    }

    // Fungsi untuk menghitung selisih antara dua waktu
    void hitungSelisih(Waktu awal, Waktu akhir, Waktu
&selisih) {
        int detikAwal = awal.jam * 3600 + awal.menit * 60 +
awal.detik;
        int detikAkhir = akhir.jam * 3600 + akhir.menit * 60
+ akhir.detik;
        int selisihDetik = detikAkhir - detikAwal;

        // Mengatasi selisih negatif
        if (selisihDetik < 0) {
            selisihDetik += 24 * 3600; // Dalam sehari
        }

        // Mengonversi selisih kembali ke jam, menit, dan
detik
        selisih.jam = selisihDetik / 3600;
        selisih.menit = (selisihDetik % 3600) / 60;
        selisih.detik = (selisihDetik % 3600) % 60;
    }

    // Fungsi untuk mencetak waktu
    void cetakWaktu(Waktu waktu) {
        cout << waktu.jam << " jam, " << waktu.menit << "
menit, " << waktu.detik << " detik" << endl;
    }

```

Hasil Running

```
Masukkan Waktu A:
Jam: 23
Menit: 13
Detik: 55
Masukkan Waktu B:
Jam: 24
Menit: 50
Detik: 10
Selisih Waktu: 1 jam, 36 menit, 15 detik
```

5. Buatlah program untuk mengelola selisih Tanggal. Fungsi : input, mencari selisihTanggal(fungsi &void) dan output

Source Code

```
/* Nama program : soal_6
   Nama          : Bagus Diatama Wardoyo
   NPM           : 140810230061
   Tanggal buat  : 14/03/2024
   Deskripsi     : program untuk mengelola selisih
Tanggal.
*****
*****/
#include <iostream>
using namespace std;

// Struktur untuk merepresentasikan tanggal
struct Tanggal {
    int hari, bulan, tahun;
};

// Fungsi-fungsi
void inputTanggal(Tanggal &tgl);
void selisihTanggal(Tanggal tglAwal, Tanggal tglAkhir,
Tanggal &selisih);
void cetakTanggal(Tanggal tgl);

int main() {
    Tanggal tglAwal, tglAkhir, selisih;

    cout << "Masukkan Tanggal Awal:" << endl;
    inputTanggal(tglAwal);
```

```

        cout << "Masukkan Tanggal Akhir:" << endl;
        inputTanggal(tglAkhir);

        selisihTanggal(tglAwal, tglAkhir, selisih);

        cout << "Selisih Tanggal: ";
        cetakTanggal(selisih);

        return 0;
    }

// Fungsi untuk memasukkan tanggal
void inputTanggal(Tanggal &tgl) {
    cout << "Hari: ";
    cin >> tgl.hari;
    cout << "Bulan: ";
    cin >> tgl.bulan;
    cout << "Tahun: ";
    cin >> tgl.tahun;
}

// Fungsi untuk menghitung selisih antara dua tanggal
void selisihTanggal(Tanggal tglAwal, Tanggal tglAkhir,
Tanggal &selisih) {
    // Menghitung selisih hari, bulan, dan tahun
    selisih.tahun = tglAkhir.tahun - tglAwal.tahun;
    selisih.bulan = tglAkhir.bulan - tglAwal.bulan;
    selisih.hari = tglAkhir.hari - tglAwal.hari;

    // Mengatasi hasil yang negatif
    if (selisih.hari < 0) {
        selisih.hari += 30; // Anggap setiap bulan adalah
30 hari
        selisih.bulan--;
    }
    if (selisih.bulan < 0) {
        selisih.bulan += 12; // 12 bulan dalam setahun
        selisih.tahun--;
    }
}

```

```
}  
  
// Fungsi untuk menampilkan tanggal  
void cetakTanggal(Tanggal tgl) {  
    cout << tgl.hari << " Hari " << tgl.bulan << " Bulan  
" << tgl.tahun << " Tahun" << endl;  
}
```

Hasil Running

```
Masukkan Tanggal Awal:  
Hari: 08  
Bulan: 06  
Tahun: 2005  
Masukkan Tanggal Akhir:  
Hari: 23  
Bulan: 03  
Tahun: 2024  
Selisih Tanggal: 15 Hari 9 Bulan 18 Tahun
```