

LAPORAN TUGAS ALGORITMA DAN PEMROGRAMAN LANJUT
ARRAY, PROSEDUR DAN FUNGSI



Disusun oleh:

Ahmad Rafi' Irsyad Nugraha
(2509106034)

Muhammad Bakil Amru
(2509106044)

Kelas (A'25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA

2025

DAFTAR ISI

DAFTAR ISI	1
BAB I PEMBAHASAN SOAL 1-5	2
1.1. Pembahasan Soal Nomor 1	2
1.2. Pembahasan Soal Nomor 2	8
1.3. Pembahasan Soal Nomor 3	11
1.4. Pembahasan Soal Nomor 4	15
1.5. Pembahasan Soal Nomor 5	18
BAB 2 PEMBAHASAN TUGAS A	20
2.1. Langkah-langkah Pseudocode.....	20
2.2. Flowchart	21
2.3. Program.....	21
2.4. Algoritma	22
BAB 3 PEMBAHASAN TUGAS B.....	23
3.1. Langkah-langkah Pseudocode.....	23
3.2. Flowchart	24
3.3. Program.....	24
3.4. Algoritma	26
BAB 4 PEMBAHASAN TUGAS C	27
4.1. Langkah-langkah Pseudocode.....	27
4.2. Flowchart	28
4.3. Program.....	29
4.4. Algoritma	29
BAB 5 KESIMPULAN.....	31
DAFTAR PUSTAKA	32

BAB I PEMBAHASAN SOAL 1-5

1.1. Pembahasan Soal Nomor 1

Pengerja: Ahmad Rafi' Irsyad Nugraha (2509106034)

Perintah pada soal:

“Buatlah pseudocode, algoritma, flowchart dan program untuk menampilkan nilai min, max dan rata-rata dalam suatu matriks. Contoh bila ada matrik berikut maka nilai terkecil adalah 2, nilai maksimum adalah 8 dan rata-rata adalah 5.”

6	7	8
1	5	3

Langkah-langkah Algoritma:

- Mulai
- Deklarasi matriks 2 x 3 dengan nilai [[6,7,8],[1,5,3]]

UNTUK MENGHITUNG RATA-RATA (MEAN)

- Inisiasi sum dengan 0
- Inisiasi length dengan 0
- Untuk setiap baris matriks:
 - Untuk Setiap kolom matriks:
 - Tambahkan nilai sum dengan nilai elemen matriks
 - Tambahkan nilai length dengan 1
- Inisiasi mean dengan menghitung hasil pembagian dari sum dengan length
- Tampilkan mean

UNTUK MENCARI NILAI MAX

- Inisiasi nilai max dengan elemen baris pertama dan kolom pertama matriks
- Untuk setiap baris matriks:
 - Untuk Setiap kolom matriks:
 - Jika elemen matriks saat ini lebih besar dari nilai max, maka ganti nilai max dengan elemen matriks saat ini
 - Jika tidak, maka nilai max tetap

- Tampilkan nilai max

UNTUK MENCARI NILAI MIN

- Inisiasi nilai min dengan elemen baris pertama dan kolom pertama matriks
- Untuk setiap baris matriks:
 - Untuk Setiap kolom matriks:
 - Jika elemen matriks saat ini lebih kecil dari nilai min, maka ganti nilai min dengan elemen matriks saat ini
 - Jika tidak, maka nilai min tetap
- Tampilkan nilai min
- Selesai

Langkah-Langkah Pseudocode:

Program: Menghitung Rata-rata, Nilai Maksimum, dan Minimum Matriks

Kamus:

```
matriks[2][3] : array integer
sum : integer
length : integer
mean : float
max : integer
min : integer
i, j : integer
```

Algoritma:

Start

Initialize matriks

```
sum ← 0
```

```
length ← 0
```

```
For i = 0 to 1
```

```
  For j = 0 to 2
```

```
    length ← length + 1
```

```
    sum ← sum + matriks[i][j]
```

```
  End For
```

```
End For
```

```
mean ← sum / length
```

```
max ← matriks[0][0]

For i = 0 to 1
  For j = 0 to 2
    If matriks[i][j] > max Then
      max ← matriks[i][j]
    End If
  End For
End For

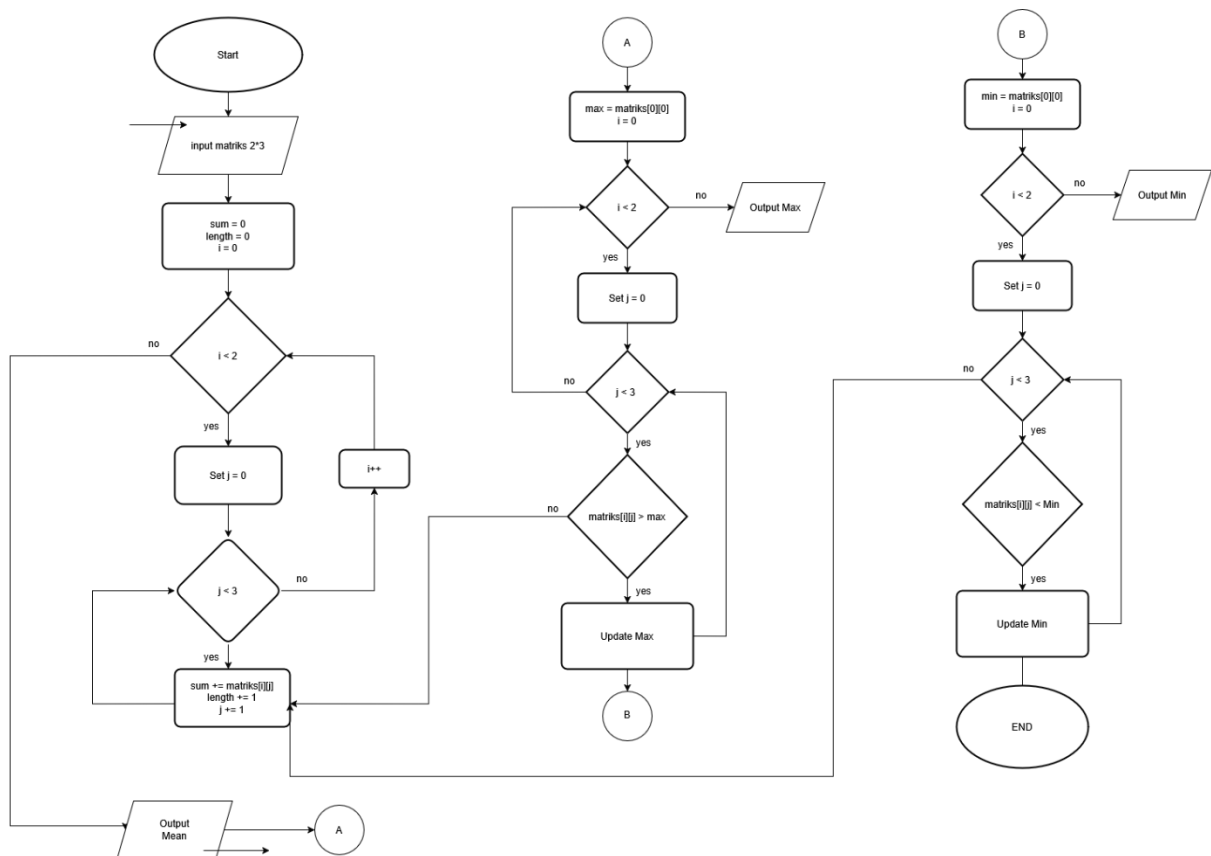
min ← matriks[0][0]

For i = 0 to 1
  For j = 0 to 2
    If matriks[i][j] < min Then
      min ← matriks[i][j]
    End If
  End For
End For

Output mean, max, min

End
```

Flowchart:



Program:

```

#include <iostream>
using namespace std;

int main()
{
    int matriks[2][3] = {{6, 7, 8}, {1, 5, 3}};

    // Menghitung rata-rata

```

```

int sum = 0;
int length = 0;
for (int i = 0; i < 2; i++)
{
    for (int j = 0; j < 3; j++)
    {
        length += 1;
        sum += matriks[i][j];
    }
}

float mean = sum / length;

cout << "Rata-rata matriks = " << mean << endl;

// Menghitung nilai max

// menginisiasi nilai max dengan elemen pertama pada
matriks
int max = matriks[0][0];

for (int i = 0; i < 2; i++)
{
    for (int j = 0; j < 3; j++)
    {
        if (matriks[i][j] > max)
        {
            max = matriks[i][j];
        }
    }
}

cout << "Nilai max matriks = " << max << endl;

// Menghitung nilai min

// menginisiasi nilai min dengan elemen pertama pada
matriks
int min = matriks[0][0];

for (int i = 0; i < 2; i++)
{
    for (int j = 0; j < 3; j++)
    {
        if (matriks[i][j] < min)
        {
            min = matriks[i][j];
        }
    }
}

```

```

    }
}

cout << "Nilai min matriks = " << min << endl;
return 0;
}

```

1.2. Pembahasan Soal Nomor 2

Pengerja: Ahmad Rafi' Irsyad Nugraha (2509106034)

Perintah pada soal:

“Buatlah pseudocode, algoritma, flowchart dan program untuk menghitung invers matriks berordo 2x2.”

Langkah-langkah Algoritma:

- Mulai
- Deklarasi matriks 2x2
- Untuk setiap baris matriks:
 - Untuk Setiap kolom matriks:
 - Input elemen matriks
- Hitung determinan dengan rumus: $\text{determinan} = (\text{matriks}[0][0] \times \text{matriks}[1][1]) - (\text{matriks}[0][1] \times \text{matriks}[1][0])$
- Hitung adjoin matriks dengan aturan:
 - $\text{adjoin}[0][0] = \text{matriks}[1][1]$
 - $\text{adjoin}[0][1] = -\text{matriks}[0][1]$
 - $\text{adjoin}[1][0] = -\text{matriks}[1][0]$
 - $\text{adjoin}[1][1] = \text{matriks}[0][0]$
- Deklarasi matriks
- Untuk setiap baris matriks:
 - Untuk Setiap kolom matriks:
 - Hitung invers matriks dengan rumus: $\text{invers}[i][j] = (1 / \text{determinan}) \times \text{adjoin}[i][j]$
- Tampilkan invers
- Selesai

Langkah-Langkah Pseudocode:

Program: Menghitung Invers Matriks Ordo 2x2

Kamus:

matriks[2][2] : array integer
determinan : float
adjoin[2][2] : array integer
invers[2][2] : array float
i, j : integer

Algoritma:

Start

Input matriks

determinan \leftarrow (ad) - (bc)

Compute adjoin matrix

For i = 0 to 1

For j = 0 to 1

invers[i][j] \leftarrow (1 / determinan) * adjoin[i][j]

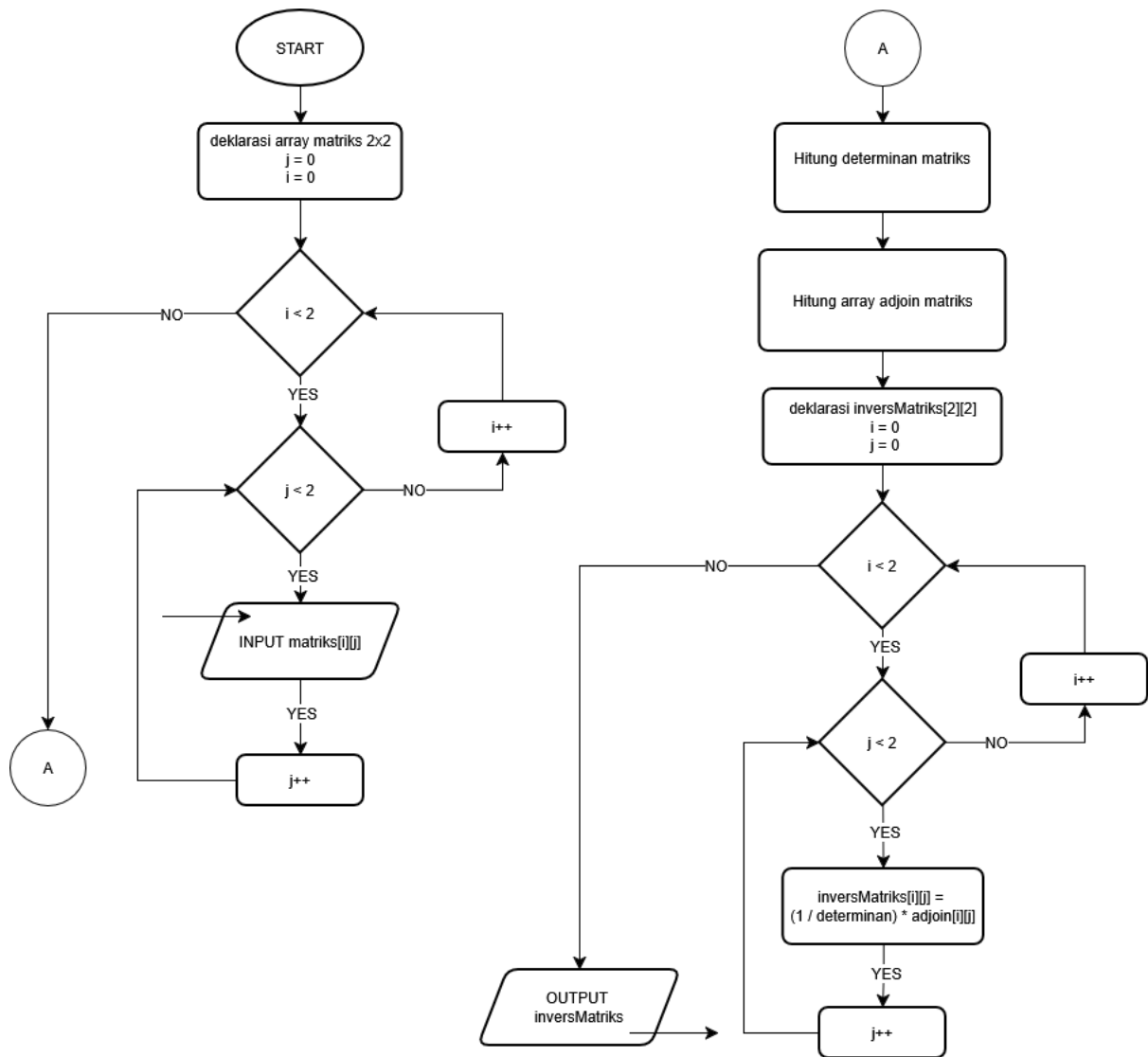
End For

End For

Output invers matrix

End

Flowchart:



Program:

```
#include <iostream>
using namespace std;

int main()
{
    int matriks[2][2];

    // User Menginput Matriks

    cout << "Buat Matriks Terlebih Dahulu\n"
         << endl;

    for (int i = 0; i < 2; i++)
    {
```

```

        for (int j = 0; j < 2; j++)
        {
            cout << "Input Ordo " << i + 1 << " Baris " << j
+ 1 << ": ";
            cin >> matriks[i][j];
        }
    }

    cout << "\nMatriks Anda: " << endl;
    for (int i = 0; i < 2; i++)
    {
        cout << "\n";
        for (int j = 0; j < 2; j++)
        {
            cout << matriks[i][j] << " ";
        }
    }

    // Deklarasi nilai determinan dan adjoin

    float determinan = (matriks[0][0] * matriks[1][1]) -
(matriks[0][1] * matriks[1][0]);

    int adjoin[2][2] = {{matriks[1][1], -(matriks[0][1])},
                        {-(matriks[1][0]), matriks[0][0]}};

    float inversMatriks[2][2];

    // Menghitung Invers

    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 2; j++)
        {
            inversMatriks[i][j] = (1 / determinan) *
adjoin[i][j];
        }
    }

    cout << "\n\nHasil Invers Matriks Anda:" << endl;
    for (int i = 0; i < 2; i++)
    {
        cout << "\n";
        for (int j = 0; j < 2; j++)
        {
            cout << inversMatriks[i][j] << " ";
        }
    }
}

```

```
}    return 0;
```

1.3. Pembahasan Soal Nomor 3

Pengerja: Ahmad Rafi' Irsyad Nugraha (2509106034)

Perintah pada soal:

“Buatlah pseudocode, algoritma, flowchart dan program untuk menampilkan array multidimensi dan menampilkan nilai rata-rata elemen yang ada didalamnya.”

Langkah-langkah Algoritma:

- Mulai
- Deklarasi matriks 3x3
- Untuk setiap baris matriks:
 - Untuk Setiap kolom matriks:
 - Input elemen matriks
- Untuk setiap baris matriks:
 - Untuk Setiap kolom matriks:
 - Tampilkan elemen matriks

MENGHITUNG RATA-RATA (MEAN)

- Inisiasi sum dengan 0
- Inisiasi length dengan 0
- Untuk setiap baris matriks:
 - Untuk Setiap kolom matriks:
 - Tambahkan nilai sum dengan nilai elemen matriks
 - Tambahkan nilai length dengan 1
- Inisiasi mean dengan menghitung hasil pembagian dari sum dengan length
- Tampilkan mean
- Selesai

Langkah-Langkah Pseudocode:

Program: Menampilkan Elemen Matriks 3x3 dan Menghitung nilai Rata-ratanya

Kamus:

```
matriks[3][3] : array integer
sum : integer
length : integer
mean : float
i, j : integer
```

Algoritma:

Start

```
For i = 0 to 2
  For j = 0 to 2
    Input matriks[i][j]
  End For
End Forsum ← 0
```

```
For i = 0 to 2
  For j = 0 to 2
    Output matriks[i][j]
  End For
End Forsum ← 0
```

length ← 0

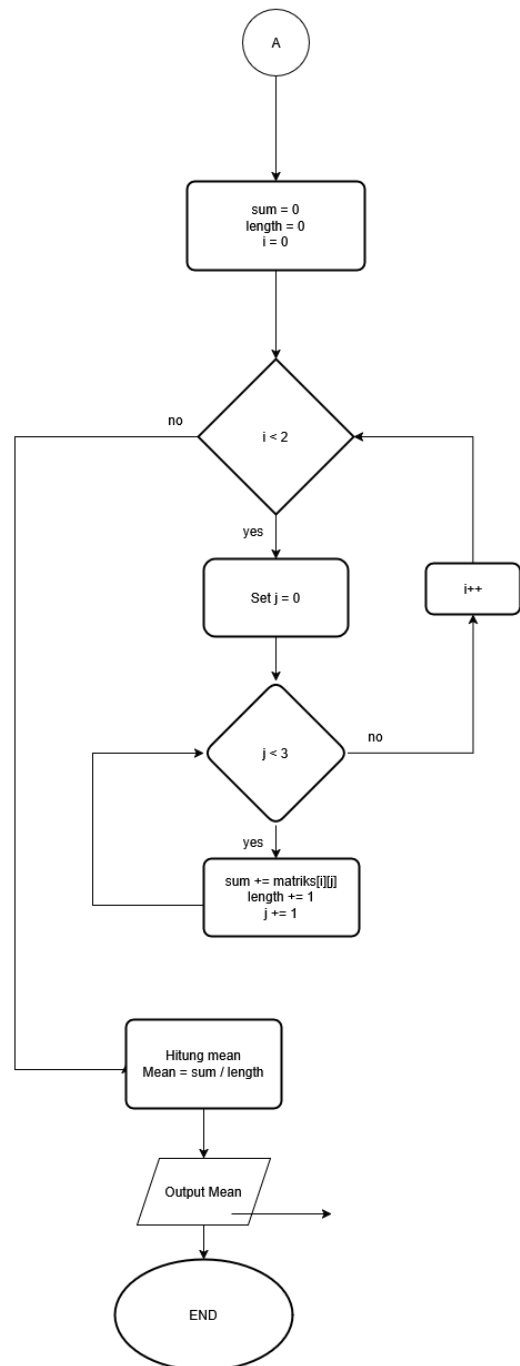
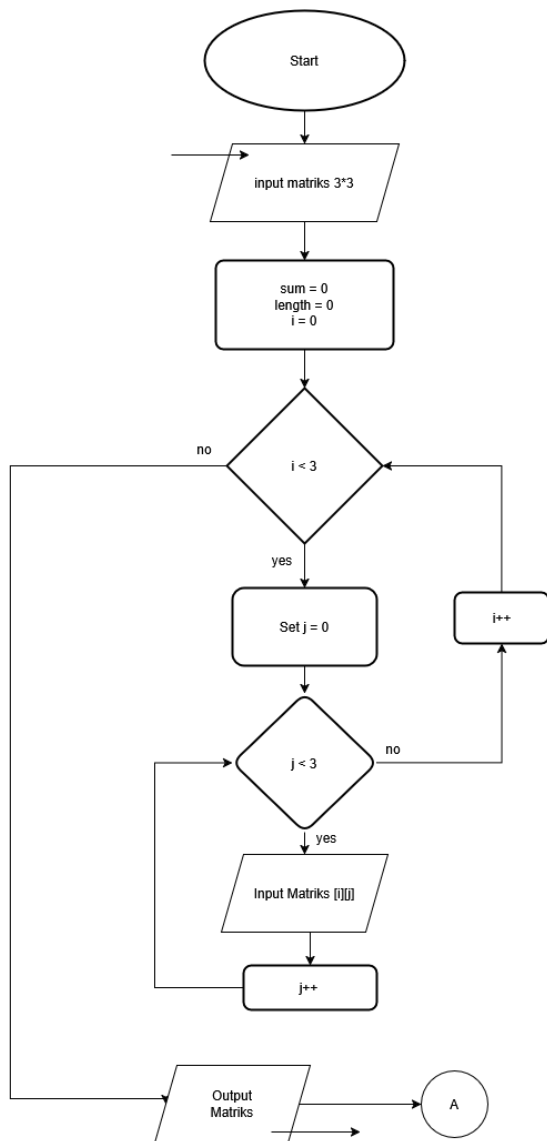
```
For i = 0 to 2
  For j = 0 to 2
    sum ← sum + matriks[i][j]
    length ← length + 1
  End For
End For
```

mean ← sum / length

Output mean

End

Flowchart:



Program:

```

#include <iostream>
using namespace std;

int main()
{
    // Input Matriks
  
```

```

int matriks[3][3];

cout << "Buat Matriks Terlebih Dahulu\n"
      << endl;

for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        cout << "Input Ordo " << i + 1 << " Baris " << j
+ 1 << ": ";
        cin >> matriks[i][j];
    }
}

cout << "\nMatriks Anda: " << endl;
for (int i = 0; i < 3; i++)
{
    cout << "\n";
    for (int j = 0; j < 3; j++)
    {
        cout << matriks[i][j] << " ";
    }
}

// Hitung Rata-rata (mean)

int sum = 0;
int length = 0;
for (int i = 0; i < 2; i++)
{
    for (int j = 0; j < 3; j++)
    {
        length += 1;
        sum += matriks[i][j];
    }
}

float mean = sum / length;

cout << "\nRata-rata matriks anda = " << mean << endl;
return 0;
}

```

1.4. Pembahasan Soal Nomor 4

Pengerja: Ahmad Rafi' Irsyad Nugraha (2509106034)

Perintah Soal:

“Buatlah program menggunakan pointer.”

Program:

```
#include <iostream>
using namespace std;

int main()
{
    struct Buku
    {
        int id;
        string judul;
        string pengarang;
        float rating;
    };

    // Menginput Jumlah Buku

    int jumlahBuku = 0;

    while (jumlahBuku <= 0)
    {

        cout << "Input Jumlah Buku: ";
        cin >> jumlahBuku;

        if (jumlahBuku <= 0)
        {
            cin.ignore(1000, '\n');
            cin.clear();

            cout << "Jumlah buku harus berupa angka dan
tidak boleh kurang dari 0\n"
                << endl;
        }
    }

    // Mendeklarasi Pointer listBuku dengan tipe data Array
    struct Buku[]

    Buku *listBuku = new Buku[jumlahBuku];
```

```

// Menginput data buku menggunakan pointer traversal

Buku *ptrInput = listBuku;
for (int i = 0; i < jumlahBuku; i++)
{
    cin.ignore();
    ptrInput->id = i + 1;
    cout << "Input Judul Buku: ";
    getline(cin, ptrInput->judul);
    cout << "Input Pengarang Buku: ";
    getline(cin, ptrInput->pengarang);

    cout << "Input Rating Buku (Skala 0-5): ";
    cin >> ptrInput->rating;
    while (ptrInput->rating < 0 || ptrInput->rating > 5)
    {
        cout << "Rating harus berdasarkan skala 0-5\n"
               << endl;

        cout << "Input Rating Buku (Skala 0-5): ";
        cin >> ptrInput->rating;
    }

    cout << "\nBuku Berhasil Ditambahkan!\n"
           << endl;

    ptrInput += 1;
}

// Menampilkan Buku Menggunakan Pointer Traversal

Buku *ptrOutput = listBuku;

for (int i = 0; i < jumlahBuku; i++)
{
    cout << "\nBuku ke-" << i + 1 << endl;
    cout << "ID          : " << ptrOutput->id << endl;
    cout << "Judul         : " << ptrOutput->judul << endl;
    cout << "Pengarang    : " << ptrOutput->pengarang <<
endl;
    cout << "Rating       : " << ptrOutput->rating << endl;

    ptrOutput += 1;
}

return 0;
}

```

Pembahasan:

Pada Program ini saya membuat sebuah Create dan Read sederhana dengan menginput dan menampilkan data buku menggunakan struct, pointer, dan alokasi memori dinamis pada C++.

Struct Buku digunakan untuk menyimpan data berupa id, judul, pengarang, dan rating. Program terlebih dahulu meminta user untuk menginput jumlah buku dengan error handling dan validasi agar tidak kurang dari atau sama dengan 0 serta untuk memastikan bahwa input hanya berupa angka. Setelah itu, memori dialokasikan secara dinamis menggunakan pointer sesuai jumlah buku yang diinput.

```
Buku *listBuku = new Buku[jumlahBuku];
```

Kemudian program akan meminta user untuk mengisi data buku, kecuali pada bagian id yang akan terisi otomatis oleh program. Masing-masing data yang telah diinput akan disimpan ke masing-masing elemen array menggunakan pointer traversal dengan operator ->. Program juga memvalidasi nilai rating agar berada pada rentang 0–5.

Pada saat menampilkan data buku, program juga menggunakan pointer traversal untuk berpindah dari satu elemen ke elemen lainnya dalam setiap perulangan dan menampilkan masing masing data elemen

Secara keseluruhan, program ini menerapkan konsep struct, pointer, alokasi memori dinamis, serta validasi input dalam pengelolaan data buku.

1.5. Pembahasan Soal Nomor 5

Pengerja: Ahmad Rafi' Irsyad Nugraha (2509106034)

Perintah Soal:

“Buatlah program untuk menampilkan matriks segitiga atas dan bawah seperti ini:”

9	9	9	9	0
9	9	9	0	1
9	9	0	1	1
9	0	1	1	1
0	1	1	1	1

Program:

```
#include <iostream>
using namespace std;

int main()
{
    int matriksSegitiga[5][5] = {{9, 9, 9, 9, 0},
                                   {9, 9, 9, 0, 1},
                                   {9, 9, 0, 1, 1},
                                   {9, 0, 1, 1, 1},
                                   {0, 1, 1, 1, 1}};
    cout << "Matriks Segitiga Atas dan Bawah: " << endl;

    for (int i = 0; i < 5; i++)
```

```
{
    cout << "\n";
    for (int j = 0; j < 5; j++)
    {
        cout << matriksSegitiga[i][j] << " ";
    }
}
return 0;
}
```

Pembahasan:

Mula-mula program akan mendeklarasi array 2 dimensi yang berbentuk seperti matriks segitga atas dan bawah sesuai dengan perintah soal

Kemudian program akan menampilkan Matriks tersebut menggunakan nested loop agar dapat mengakses masing-masing elemen dan menampilkannya secara dinamis

BAB 2

PEMBAHASAN TUGAS A

Pengerja: Muhammad Bakil Amru (2509106044)

2.1 Langkah-Langkah Pseudocode:

Program: Menentukan Jenis Segitiga

Kamus:

sisiA, sisiB, sisiC : integer

Algoritma:

Start

Input sisiA, sisiB, sisiC

If tidak memenuhi syarat segitiga Then
 Output "Bukan Triangle"

Else If memenuhi rumus Pythagoras Then
 Output "Segitiga siku siku"

Else If dua sisi sama Then
 Output " Segitiga Samakaki"

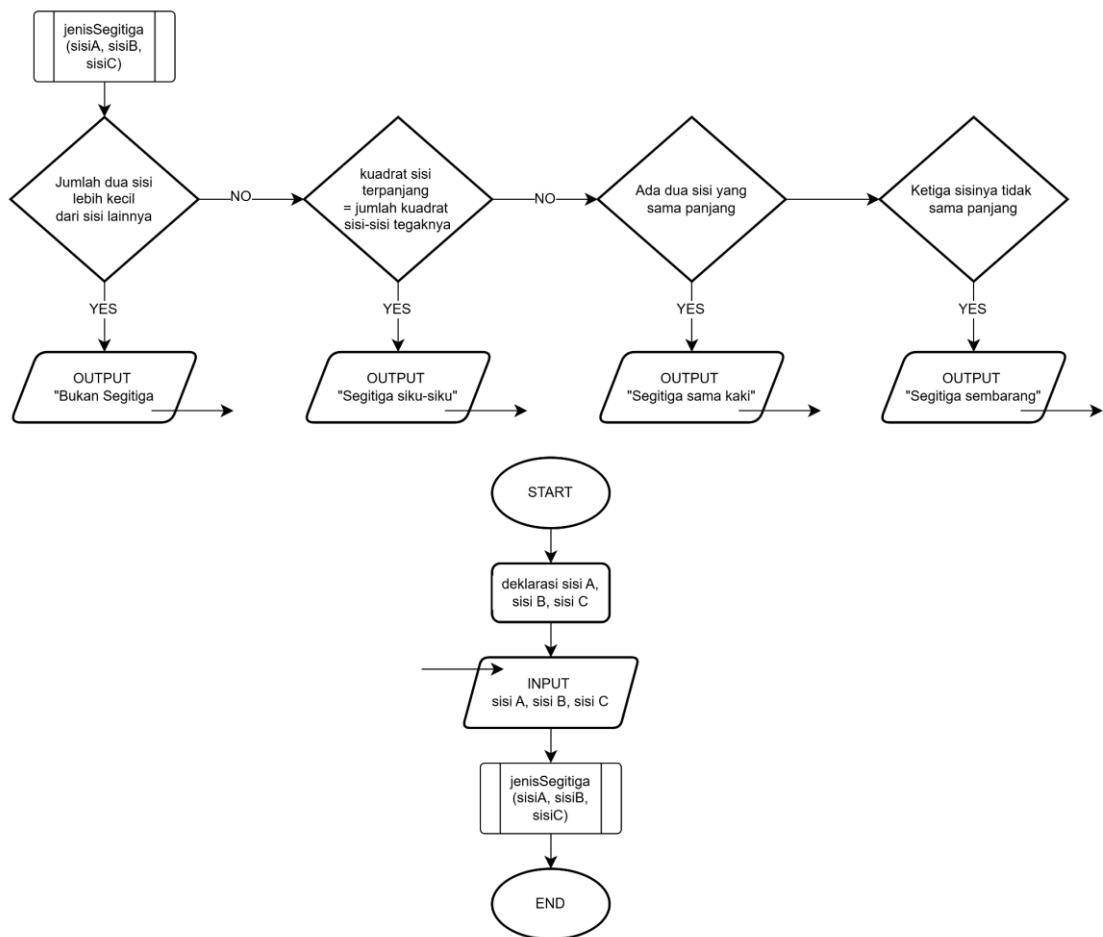
Else
 Output "Segitiga Sembarang"
End If

EndJudul

Output mean, max, min

End

2.2 Flowchart:



2.3 Program:

```

#include <iostream>
#include <cmath>
using namespace std;

void jenisSegitiga(int sisiA, int sisiB, int sisiC)
{
    if (sisiA + sisiB < sisiC || sisiB + sisiC < sisiA ||
    sisiA + sisiC < sisiB)
    {
        cout << "Bukan Segitiga" << endl;
    }
    else if (sisiA * sisiA + sisiB * sisiB == sisiC * sisiC
    ||
        sisiB * sisiB + sisiC * sisiC == sisiA * sisiA
    ||
        sisiC * sisiC + sisiA * sisiA == sisiB * sisiB)
    {
        cout << "Segitiga Siku Siku" << endl;
    }
}
  
```

```

    }
    else if (
        (sisiA == sisiB && sisiC != sisiA && sisiC != sisiB)
    ||
        (sisiB == sisiC && sisiA != sisiB && sisiA != sisiC)
    ||
        (sisiC == sisiA && sisiB != sisiC && sisiB != sisiA)
    )
    {
        cout << "Segitiga Sama Kaki" << endl;
    }
    else if (sisiA != sisiB && sisiB != sisiC && sisiA !=
sisiC)
    {
        cout << "Segitiga Sembarang" << endl;
    }
}

int main()
{
    long sisiA;
    long sisiB;
    long sisiC;

    cout << "Input Sisi A: ";
    cin >> sisiA;
    cout << "Input Sisi B: ";
    cin >> sisiB;
    cout << "Input Sisi C: ";
    cin >> sisiC;

    jenisSegitiga(sisiA, sisiB, sisiC);

    return 0;
}

```

Algoritma:

Pada Tugas A ini, Kami membuat program untuk menentukan sebuah segitiga dengan user diminta untuk menginput ketiga sisinya dengan syarat :

1. *Jika jumlah dua sisi lebih kecil dari sisi ketiga maka tidak bisa disebut sebagai segitiga dan akan menghasilkan output “bukan segitiga”.*
2. *Jika memenuhi syarat pythagoras yaitu $A^2 + B^2 = C^2$ maka akan menghasilkan output “segitiga siku-siku”.*
3. *Jika ada 2 sisi sama panjang, dan sisi lainnya berbeda akan menghasilkan output “segitiga sama kaki”*
4. *Jika semua sisi berbeda akan menghasilkan output “segitiga sembarang”.*

BAB 3

PEMBAHASAN TUGAS B

Pengerja: Muhammad Bakil Amru (2509106044)

3.1 Langkah-Langkah Pseudocode:

Program: Cek Triple Pythagoras

Kamus:

sisi[3] : array integer
nilaiPythagoras : integer
nilaiHypotenuse : integer
i, j : integer

Algoritma:

Start

Input sisi[0], sisi[1], sisi[2]

Sort array sisi ascending (Bubble Sort)

nilaiPythagoras \leftarrow sisi[0]² + sisi[1]²
nilaiHypotenuse \leftarrow sisi[2]²

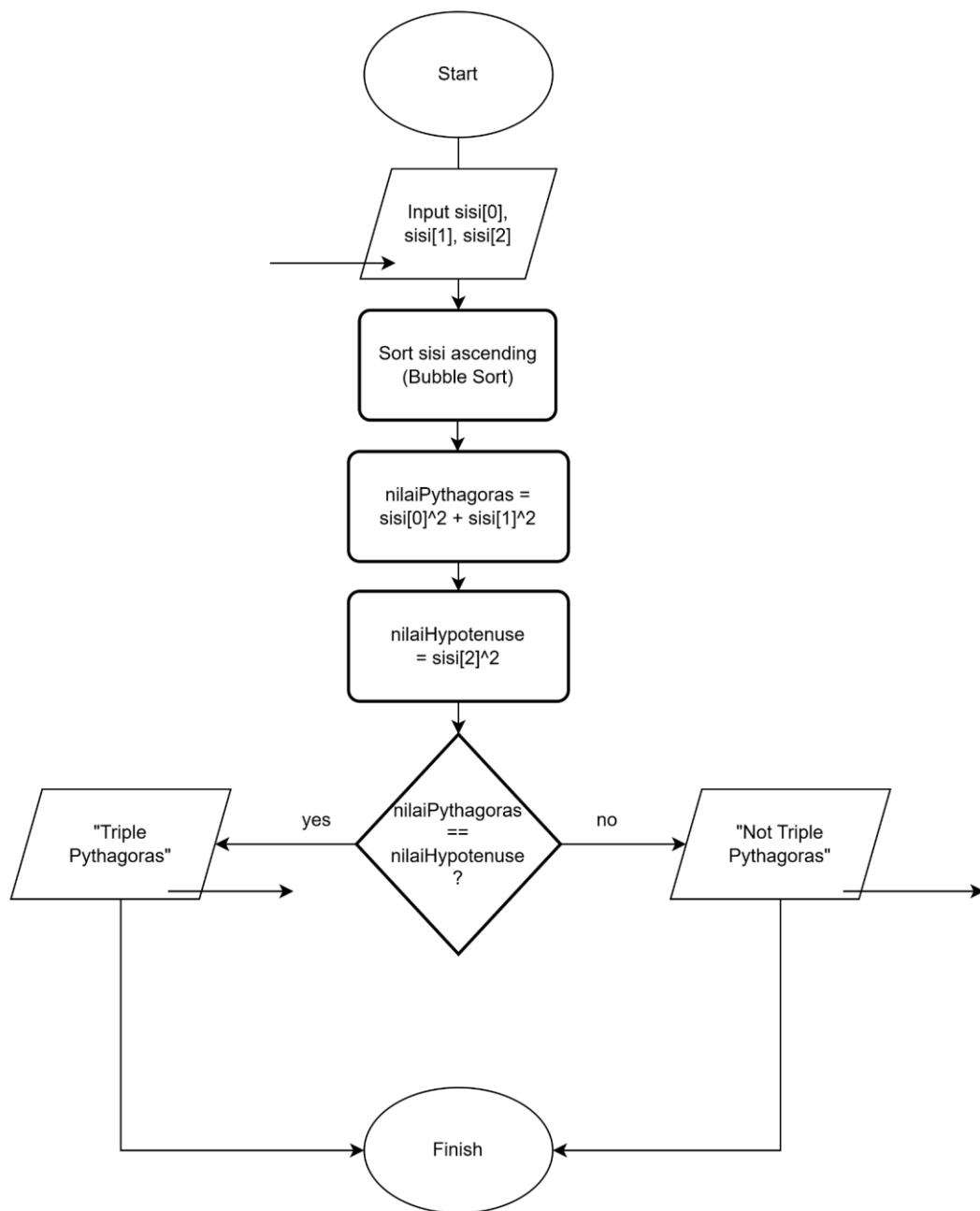
If nilaiPythagoras == nilaiHypotenuse Then
 Output "Triple Pythagoras"

Else
 Output "Not Triple Pythagoras"

End If

End

3.2 Flowchart:



3.3 Program:

```
#include <iostream>
#include <cmath>
using namespace std;

bool isPythagoras(int sisiA, int sisiB, int sisiC)
{
```

```

// Menggunakan algoritma Bubble sort untuk mencari Sisi
Terbesar/Sisi Miring (sisi[2])

int sisi[3] = {sisiA, sisiB, sisiC};
int n = sizeof(sisi) / sizeof(sisi[0]);

for (int i = 0; i < n - 1; i++)
{
    for (int j = 0; j < n - i - 1; j++)
    {
        if (sisi[j] > sisi[j + 1])
        {
            int temp = sisi[j];
            sisi[j] = sisi[j + 1];
            sisi[j + 1] = temp;
        }
    }
}

for (int i = 0; i < n; i++)
{
    cout << sisi[i] << endl;
}

int nilaiPythagoras = (sisi[0] * sisi[0]) + (sisi[1] *
sisi[1]);
int nilaiHypotenuse = sisi[2] * sisi[2];

cout << "Nilai Pythagoras (a^2 + b^2) = " <<
nilaiPythagoras << endl;
cout << "Nilai c^2 = " << nilaiHypotenuse << endl;

return (nilaiPythagoras == nilaiHypotenuse);
}

int main()
{
    int sisiA;
    int sisiB;
    int sisiC;

    cout << "Input Sisi A: ";
    cin >> sisiA;
    cout << "Input Sisi B: ";
    cin >> sisiB;
    cout << "Input Sisi C: ";
    cin >> sisiC;

    if (isPythagoras(sisiA, sisiB, sisiC))

```

```

{
    cout << "Termasuk Triple Pythagoras" << endl;
}
else
{
    cout << "Tidak Termasuk Triple Pythagoras" << endl;
}

return 0;
}

```

3.4 Algoritma :

Jadi pada tugas B, Kami membuat program untuk mengecek Triple pythagoras yang berupa bilangan bulat A,B,dan C. yang dimana jika nilaiPhytagoras == $A^2 + B^2$ sama dengan nilaiHypotenuse == C^2 . maka bisa dinyatakan itu adalah sebuah pythagorasselain itu jika nilaiPhytagoras tidak sama dengan nilaiHypotenuse maka tidak bisa dinyatakan sebagai pythagoras/bukan pythagoras.

BAB 4

PEMBAHASAN TUGAS C

Pengerja: Muhammad Bakil Amru (2509106044)

4.1 Langkah-Langkah Pseudocode:

Program: Menghitung Nilai Pangkat Bilangan Menggunakan Fungsi Rekursif

Kamus :

bilangan : integer
pangkat : integer
hasilPangkat : integer

Function hitungPangkat(bilangan, pangkat) : integer

Algoritma:

Program Utama

Start

Input bilangan
Input pangkat

hasilPangkat ← hitungPangkat(bilangan, pangkat)

Output hasilPangkat

End

Function hitungPangkat(bilangan, pangkat)

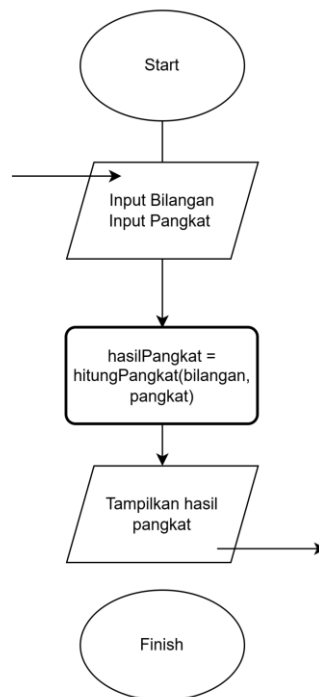
Start

```
If pangkat == 0 Then
  Return 1
Else
  Return bilangan * hitungPangkat(bilangan, pangkat - 1)
End If

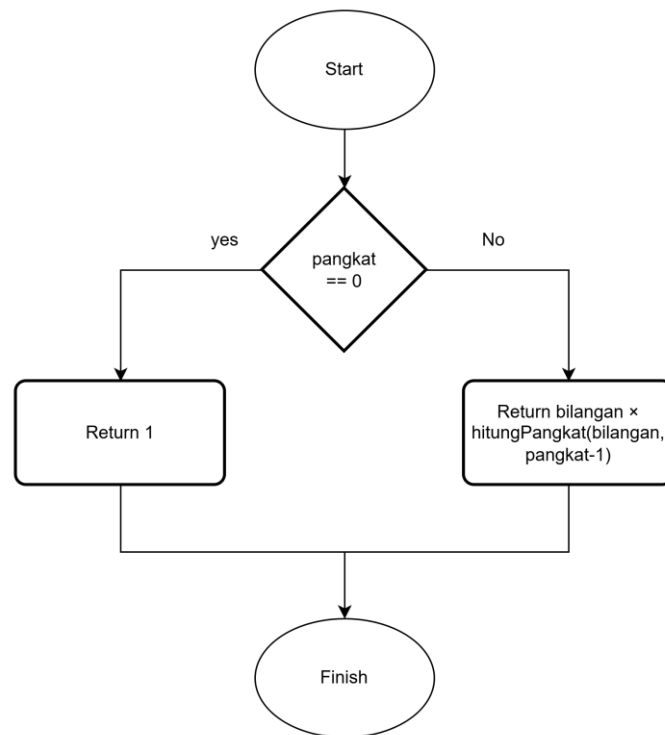
End
```

4.2 Flowchart

Program Utama :



Flowchart Fungsi Rekursif :



4.3 Program:

```

#include <iostream>
#include <cmath>
using namespace std;

int hitungPangkat(int bilangan, int pangkat)
{
    if (pangkat == 0)
    {
        return 1;
    }
    return bilangan * hitungPangkat(bilangan, pangkat - 1);
}

int main()
{
    int bilangan;
    int pangkat;

    cout << "Input Bilangan: ";
    cin >> bilangan;
    cout << "Input Pangkat: ";
    cin >> pangkat;

    int hasilPangkat = hitungPangkat(bilangan, pangkat);
  
```

```
        cout << "Hasil dari " << bilangan << " Pangkat " <<
pangkat << " = " << hasilPangkat << endl;

        return 0;
    }
```

4.4 Algoritma:

Jadi pada Tugas C, kami membuat program untuk menghitung hasil pangkat dari suatu bilangan menggunakan fungsi rekursif. Pertama-tama program akan meminta input berupa bilangan dan pangkat dari user. Setelah itu program akan menghitung nilai pangkat menggunakan fungsi hitung pangkat. Jika pangkat bernilai 0 maka hasilnya adalah 1, karena bilangan apapun jika dipangkatkan 0 hasilnya 1. Jika pangkat tidak sama dengan 0 maka program akan mengalikan bilangan dengan hasil pemanggilan fungsi rekursif secara berulang sampai nilai pangkat menjadi 0. Setelah proses perhitungan selesai, hasil pangkat akan ditampilkan ke layar sebagai output.

KESIMPULAN

Berdasarkan seluruh pembahasan pada laporan ini, kami menyimpulkan bahwa pemahaman mengenai array, prosedur, fungsi, serta pointer sangat penting dalam proses pengembangan program. Melalui penyusunan algoritma, pseudocode, dan flowchart sebelum implementasi kode, proses pembuatan program menjadi lebih terarah, sistematis, dan terstruktur.

Pada pembahasan soal 1,2,3 dan 5, kami mempelajari bagaimana penggunaan array multidimensi dalam sebuah program guna menyusun sebuah matriks, serta cara menghitung nilai minimum, maksimum, dan rata-ratanya. Kami juga belajar cara mencari invers dari sebuah matriks melalui serangkaian algoritma, serta menampilkan bentuk matriks tertentu. Selain itu, pada tugas penggunaan pointer (soal 4), kami memahami konsep alokasi memori dinamis dan bagaimana pointer dapat digunakan untuk mengakses serta mengelola data secara lebih fleksibel.

Dalam Tugas A,B, dan C kami menerapkan konsep Modular programming seperti fungsi dan prosedur guna memisahkan logika program agar lebih mudah dipahami dan digunakan ulang. Penggunaan fungsi membantu meningkatkan efisiensi penulisan kode serta memudahkan proses pengujian program.

Secara keseluruhan, melalui tugas ini kami tidak hanya belajar menulis program, tetapi juga melatih kemampuan berpikir logis, sistematis, dan analitis dalam menyelesaikan permasalahan menggunakan bahasa pemrograman C++. Konsep-konsep dasar yang dipelajari pada praktikum ini menjadi fondasi penting untuk pengembangan program yang lebih kompleks di masa mendatang.

DAFTAR PUSTAKA

Aayushi Agrawal (2025). *How to Sort an Array in C++?*. *geeksforgeeks.org*.
<https://www.geeksforgeeks.org/cpp/how-to-sort-an-array-in-cpp/>