

LAPORAN PEMBUATAN APLIKASI CALCULATOR MATRIKS

Laporan Ini Dibuat untuk Memenuhi Tugas Ujian Akhir Semester Mata Kuliah Aljabar Geometri

Dosen Pengampu: Ahmad Zamakhsyari Sidiq, M.T.



Kelompok 7

Disusun oleh:

Deden Khoerul Fahmi	(10222002)
Rifa Maulana Ibrahim	(10222008)
Rian Abdul Aziz	(10222037)
Wahdah Siti Nurfauziyah	(10222046)
Siti Rubiah Adawiah	(10222148)
Pipin Sopiah	(10222195)

SEKOLAH TINGGI TEKNOLOGI CIPASUNG

TASIKMALAYA

2023/2024

KATA PENGANTAR

Puji syukur kita panjatkan kehadiran Tuhan Yang Maha Esa atas rahmat-Nya sehingga kami dapat menyelesaikan laporan ini. Laporan ini disusun sebagai hasil dari pengembangan dan implementasi aplikasi kalkulator matriks. Kalkulator matriks ini dirancang dengan tujuan untuk mempermudah pengguna dalam melakukan operasi-operasi matriks secara efisien.

Dalam laporan ini, kami akan menjelaskan secara detail langkah-langkah pembuatan aplikasi kalkulator matriks, termasuk konsep desain, pemrograman, dan implementasi dari pengujian kode program utama yang diterapkan.

Semoga laporan ini dapat memberikan gambaran yang detail dan bermanfaat bagi pembaca. Terima kasih kepada semua pihak yang telah memberikan dukungan dan bantuan selama proses pengembangan aplikasi ini. Kami juga berharap kritik dan saran yang sifatnya membangun di masa mendatang.

Tasikmalaya, 11 Desember 2023

Penyusun

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI	iii
BAB I DESKRIPSI MASALAH	1
BAB II TEORI SINGKAT.....	2
A. Sistem Persamaan Linear	2
B. Determinan.....	4
C. Matriks Balikan/Invers.....	5
D. Matriks Tranpose	5
E. Penjumlahan Matriks	5
BAB III IMPLEMENTASI PROGRAM.....	6
A. Include.....	6
B. Menentukan jumlah maksimal kolom dan baris yang dapat di isi	6
C. Menampilkan menu dan sub menu	7
D. Menjalankan program	8
BAB IV PENGUJIAN.....	17
BAB V PENUTUP	20
A. Kesimpulan	20
B. Saran	20
C. Refleksi	21

BAB I

DESKRIPSI MASALAH

Matriks merupakan susunan bilangan (elemen) yang disusun menurut baris dan kolom sehingga berbentuk persegi panjang dan banyak manfaat, diantaranya dapat digunakan untuk mengenkriptografi file dan menghitung image. Apabila akan dilakukan operasi matriks nantinya membutuhkan waktu dan perhitungan yang lumayan panjang, tergantung pada ordo yang akan dimasukkan. Kalkulator yang merupakan alat bantu untuk menghitung bilangan menjadi sangat penting dan membantu apabila terdapat bilangan yang sulit dikerjakan atau mengalami perhitungan yang panjang untuk memperoleh suatu nilai.. Oleh sebab itulah diperlukan suatu aplikasi yang dapat mempermudah dan mempersingkat waktu dalam melakukan perhitungan matriks tersebut, yaitu dengan menggunakan kalkulator matriks.

Kalkulator matriks adalah aplikasi hitung yang digunakan untuk menghitung operasi dasar matriks meliputi penjumlahan, pengurangan, dan perkalian, kemudian aplikasi ini juga dapat mengubah menjadi bentuk transpose matriks, menghitung determinan, dan invers matriks.

Kalkulator matriks ini dibuat menggunakan bahasa pemrograman C++. Bahasa pemrograman C++ adalah salah satu bahasa pemrograman komputer yang paling populer di dunia dan dapat digunakan untuk membuat berbagai aplikasi berkinerja tinggi (general-purpose programming language) seperti aplikasi games, browsers, pengolah gambar dan masih banyak lagi.

BAB II

TEORI SINGKAT

A. Sistem Persamaan Linear

Sistem persamaan linear atau sering disebut SPL adalah Sistem persamaan linear ini dapat diselesaikan dengan banyak cara salah satunya dengan menggunakan eliminasi Gauss- Gauss Jordan. Elemanisi Gauss dapat digunakan untuk memperoleh matriks eselon baris, sedangkan eleminasi Gauss-Jordan digunakan untuk memperoleh matriks eselon baris tereduksi.

Adapun sifat-sifat matriks eselon baris- baris tereduksi adalah sebagai berikut:

1. Jika baris tidak seluruhnya dari 0, maka bilangan tak nol pertama baris tersebut adalah 1 (disebut satu utama).
2. Jika terdapat baris seluruhnya terdiri dari 0, maka semua baris seperti itu dikelompokkan bersama sama dibawah matriks
3. Dalam sembarang dua baris yng berurutan yang seluruhnya tidak terdiri dari nol maka, satu utama dalam baris yang lebih rendah terdapat lebih jauh ke kanan dari satu utama dalam baris yang lebih tinggi. **Sifat-sifat yang dimiliki matriks eselon baris tereduksi** ialah sifat-sifat 1, 2, 3 serta sifat sifat berikut:
4. Masing- masing kolom yang mengandung satu utama mempunyai nol di tempat lain.

Pertama yakni eleminasi Gauss. Dalam eliminasi Gauss terdapat 3 solusi, yakni solusi unik/tunggal, banyak solusi, dan tidak ada solusi. Berikut ini contoh dari eleminasi Gauss:

1. Solusi unik/tunggal

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ 2 & 3 & 1 & 1 \\ 3 & 1 & 2 & 1 \end{array} \right] \xrightarrow{\text{Eliminasi Gauss}} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 1 & 1 \end{array} \right]$$

Solusi: $x_1 = 1, x_2 = 0, x_3 = -1$

2. Solusi banyak/tidak terhingga

$$\left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 2 & -1 & 1 & 2 \\ 1 & 2 & 3 & 6 \end{array} \right] \xrightarrow[\text{Gauss}]{\text{Eliminasi}} \left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 \end{array} \right]$$

Perhatikan hasil eliminasi Gauss pada baris terakhir. Persamaan yang bersesuaian dengan baris terakhir tersebut adalah

$$0x_1 + 0x_2 + 0x_3 = 0$$

yang dipenuhi oleh banyak nilai x . Solusinya diberikan dalam bentuk parameter:

$$\text{Misalkan } x_3 = k,$$

$$\text{maka } x_2 = 2 - k \text{ dan } x_1 = 4 - x_2 - 2x_3 = 4 - (2 - k) - 2k = 2 - k,$$

dengan $k \in \mathbb{R}$. Terdapat tidak berhingga nilai k .

3. Tidak ada solusi

$$\left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 2 & -1 & 1 & 2 \\ 1 & 2 & 3 & 7 \end{array} \right] \xrightarrow[\text{Gauss}]{\text{Eliminasi}} \left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{array} \right]$$

Perhatikan hasil eliminasi Gauss pada baris terakhir. Persamaan yang bersesuaian dengan baris terakhir tersebut adalah

$$0x_1 + 0x_2 + 0x_3 = 1$$

yang dalam hal ini, tidak nilai x_i yang memenuhi, $i = 1, 2, 3$

Selanjutnya eliminasi Gauss-jordan, berikut contohnya:

Contoh 1: Selesaikan SPL berikut dengan eliminasi Gauss-Jordan

$$\begin{aligned} x_1 - x_2 + 2x_3 - x_4 &= -1 \\ 2x_1 + x_2 - 2x_3 - 2x_4 &= -2 \\ -x_1 + 2x_2 - 4x_3 + x_4 &= 1 \\ 3x_1 &= -3 \end{aligned}$$

Penyelesaian:

$$\left[\begin{array}{cccc|c} 1 & -1 & 2 & -1 & -1 \\ 2 & 1 & -2 & -2 & -2 \\ -1 & 2 & -4 & 1 & 1 \\ 3 & 0 & 0 & -3 & -3 \end{array} \right] \xrightarrow[\substack{R2 \sim -2R1 \\ R3 \sim R1 \\ R4 \sim -3R1}]{\substack{R2/3 \\ R3-R2 \\ R4-3R2}} \left[\begin{array}{cccc|c} 1 & -1 & 2 & -1 & -1 \\ 0 & 3 & -6 & 0 & 0 \\ 0 & 1 & -2 & 0 & 0 \\ 0 & 3 & -6 & 0 & 0 \end{array} \right]$$

$$\left[\begin{array}{cccc|c} 1 & -1 & 2 & -1 & -1 \\ 0 & 1 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] \xrightarrow{R1+R2} \left[\begin{array}{cccc|c} 1 & 0 & 0 & -1 & -1 \\ 0 & 1 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right]$$

Persamaan yang diperoleh:

$$x_1 - x_4 = -1 \quad (\text{i})$$

$$x_2 - 2x_3 = 0 \quad (\text{ii})$$

Matriks eselon baris tereduksi

B. Determinan

Matrik determinan adalah selisish antara perkalian elemen-elemen pada diagonal utama dengan perkalian elemen-elemen pada diagonal sekunder. Determinan matriks hanya dapat di tentukan pada matriks persegi. Determinan matriks terbagi menjadi dua yakni determinan matriks ordo 2x2 dan determinan matriks ordo 3x3.

1. Determinan ordo 2x2

Determinan matriks ordo 2x2 dapat dihitung dengan cara berikut

$$\det(A) = a_{11}a_{22} - a_{12}a_{21}$$

2. Determinan ordo 3x3

Determinan matriks ordo 3x3 dapat dihitung dengan cara berikut

$$\det(A) = (a_{11}a_{22}a_{33} + a_{12}a_{21}a_{31} + a_{13}a_{21}a_{32}) - (a_{13}a_{22}a_{31} + a_{11}a_{23}a_{32} + a_{12}a_{21}a_{33})$$

3. Determinan matriks segitiga atas

Bisa didapat dengan cara berikut:

$$\det(A) = a_{11}a_{22}a_{33}a_{44}$$



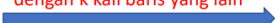
4. Determinan matriks segitiga bawah

Bisa didapat dengan cara berikut:

$$\det(A) = a_{11}a_{22}a_{33}a_{44}$$

Aturan determinan:

Misal A adalah matriks nxn yang didapat dari memanipulasi matriks, maka:

- A  B , maka $\det(B) = k \det(A)$
Kalikan sebuah baris dengan k
- A  B , maka $\det(B) = -\det(A)$
Pertukarkan dua baris
- A  B , maka $\det(B) = \det(A)$
Sebuah baris ditambahkan dengan k kali baris yang lain

Determinan matriks-matriks diatas dapat dihitung juga dengan OBE dengan memperhatikan aturan diatas, dan mengubah matriks menjadi matriks segitiga.

C. Matriks Balikan/Invers

Matriks baklikan disimbolkan dengan A^{-1} . Memiliki rumus untuk ordo 2×2 sebagai berikut:

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Dengan syarat $ad-bc$ tidak sama dengan 0. Jika $ad-bc$ tersebut menghasilkan 0 maka matriks tersebut tidak memiliki balikan (*not invertible*)

D. Matriks Tranpose

Matriks Transpose adalah matriks baru yang elemen baris dan kolomnya merupakan elemen kolom dan baris matriks sbelumnya. Misalnya matriks ordo 2×3 maka transposenya akan menjadi matriks ordo 3×2 . Bentuk penulisan matriks transpose adalah menggunakan pangkat huruf T.

Adapun sifat-sifat matriks Transpose adalah sebagai berikut:

$$\begin{aligned} (a) \quad (A^T)^T &= A \\ (b) \quad (A + B)^T &= A^T + B^T \\ (c) \quad (A - B)^T &= A^T - B^T \\ (d) \quad (kA)^T &= kA^T \\ (e) \quad (AB)^T &= B^T A^T \end{aligned}$$

E. Penjumlahan Matriks

Aturan matriks penjumlahan

1. Matriks bisa dijumlahkan/dioperasikan dengan matriks yang berordo sama
2. Memiliki sifat sebagai berikut:

Kumulatif sehingga $A + B = B + A$

Asosiatif sehingga $(A + B) + C = A + (B + C)$

unsur identitas O sehingga $A + O = O + A = A$

$A + (-A) = -A + A = O$

BAB III

IMPLEMENTASI PROGRAM

A. Include

Include adalah bagian dari struktur bahasa C. Tujuannya untuk mengimpor fungsi-fungsi yang sudah didefinisikan pada header file. Header file sendiri merupakan file yang berisi definisi fungsi yang sudah dibuat. Header file bertujuan agar bisa digunakan pada program C yang lain.

```
1    #include <iostream>
2    #include <fstream> // Header ini untuk membuat file
3    #include <iomanip> // Header ini untuk setw
4    #include <cmath> // Header ini untuk fungsi matematika
5    using namespace std;
6
```

1. **#include <iostream>**: adalah perintah untuk menyertakan *file header iostream* yang berisi fungsi-fungsi dasar untuk masukan dan keluaran, seperti *cout* dan *cin*.
2. **#include <fstream>**, program ini bisa membaca isi file sekaligus memberikan output ke dalam file tersebut. Contoh yang paling dekat adalah penulisan pada file teks yang biasanya berformat (.txt).
3. **#include <iomanip>**. Header <iomanip> terdiri dari fungsi yang digunakan untuk memanipulasi output program C++. Kita dapat membuat keluaran program apa pun menjadi lebih rapi dan rapi berdasarkan di mana kita ingin menampilkannya atau siapa yang akan menggunakannya.
4. **#include <cmath>** merupakan header untuk fungsi matematika.

B. Menentukan jumlah maksimal kolom dan baris yang dapat di isi

```
~
7    const int size = 3; // Change the size according to your matrix size
8    const int baris = 2;
9    const int kolom = 3;
10   double matriks[baris][kolom];
11
```

“const int” adalah cara untuk mendeklarasikan sebuah variabel integer yang memiliki nilai tetap atau konstan. Ini berarti bahwa setelah nilai diberikan pada variabel “const int,” nilai tersebut tidak dapat diubah selama masa eksekusi program.

C. Menampilkan menu dan sub menu

Untuk menampilkan sebuah menu dan sub menu, kita menggunakan perintah cout. adalah tipe data yang tidak bertipe karena mempunyai ukuran 0 byte biasanya digunakan untuk tipe data kosong seperti membuat fungsi.

Tipe data digunakan adalah tipe data void. Void merupakan tipe data yang tidak bertipe karena mempunyai ukuran 0 byte biasanya digunakan untuk tipe data kosong seperti membuat fungsi.

```
11
12 // Fungsi untuk menampilkan Menu Utama
13 void tampilan_Menu_Utama()
14 {
15     cout << "==== Menu Utama =====> << endl;
16     cout << "1. Pertambahan dan Pengurangan Matriks (2x2)" << endl;
17     cout << "2. Transpose Matriks" << endl;
18     cout << "3. inverse Matriks (2x2)" << endl;
19     cout << "4. Determinan" << endl;
20     cout << "5. Sistem Persamaan Linier (2x3)" << endl;
21     cout << "6. Keluar" << endl;
22     cout << "=====> << endl;
23 }
..
```

```
25 // Fungsi untuk menampilkan submenu Pertambahan dan Pengurangan
26 void tampilan_Sub_Menu_tambah_kurang()
27 {
28     cout << "----- Pertambahan dan Pengurangan ----- " << endl;
29     cout << "A. Pertambahan" << endl;
30     cout << "B. Pengurangan" << endl;
31     cout << "C. Kembali ke Menu Utama" << endl;
32     cout << "-----> << endl;
33 }
34
35 // Fungsi untuk menampilkan submenu Transpose
36 void tampilan_Sub_Menu_Transpose()
37 {
38     cout << "----- Perkalian dan Pembagian ----- " << endl;
39     cout << "A. 2x2" << endl;
40     cout << "B. 3x3" << endl;
41     cout << "C. Kembali ke Menu Utama" << endl;
42     cout << "-----> << endl;
43 }
..
```

```

45 // Fungsi untuk menampilkan submenu Determinan
46 void tampilan_Sub_Menu_Determinan()
47 {
48     cout << "----- Perkalian dan Pembagian ----- " << endl;
49     cout << "A. 2x2" << endl;
50     cout << "B. 3x3" << endl;
51     cout << "C. Kembali ke Menu Utama" << endl;
52     cout << "-----" << endl;
53 }
54
55 // Fungsi untuk menampilkan submenu SPL
56 void tampilan_Sub_Menu_SPL() {
57     cout << "----- Sistem Persamaan Linier (2x3) -----" << endl;
58     cout << "A. Input Matriks A (2x3)" << endl;
59     cout << "B. Hitung Solusi" << endl;
60     cout << "C. Kembali ke Menu Utama" << endl;
61     cout << "-----" << endl;
62 }
63

```

Output :

```

1. Pertambahan dan Pengurangan Matriks (2x2)
2. Transpose Matriks
3. inverse Matriks (2x2)
4. Determinan
5. Sistem Persamaan Linier (2x3)
6. Keluar
=====

```

D. Menjalankan program

Selanjutnya adalah variabel untuk program pilihan. Untuk variabel pilihan menggunakan instruksi switch case. Switch case merupakan program percabangan yang dirancang khusus untuk menangani pengambilan keputusan yang melibatkan sejumlah atau banyak alternatif. Karena di menu utama terdapat 6 pilihan, maka case yang di simpan berjumlah 6.

1. Case 1 (Penjumlahan dan Pengurangan Matriks (2x2))

Didalam case 1 terdapat case lainnya, case itu dibuat untuk membuat 3 pilihan yang dimana kita dapat memilih antara melakukan penjumlahan atau pengurangan atau kembali ke menu utama,

Case 1 menggunakan pernyataan while do (perulangan), while do tersebut digunakan agar kita dapat mengulangi proses perhitungan sebanyak yang kita inginkan.

```
// Fungsi untuk menangani pilihan di Menu Utama
void prosesMenuUtama(int pilihan)
{
    switch (pilihan)
    {
        case 1:
            // Pilih menu Pertambahan dan Pengurangan
            while (true)
            {
                tampilan_Sub_Menu_tambah_kurang();
                char submenu;
                cout << "Pilih: ";
                cin >> submenu;
                int A[size][size], B[size][size], result[size][size];
                int size = 2;
                switch (submenu)

                switch (submenu)
                {
                    case 'A':
                    case 'a':
                        cout << "Masukkan elemen matriks A:\n";
                        for (int i = 0; i < size; ++i)
                        {
                            for (int j = 0; j < size; ++j)
                            {
                                cout << "A[" << i + 1 << "][" << j + 1 << "]: ";
                                cin >> A[i][j];
                            }
                        }
                        // Input matriks B
                        cout << "Masukkan elemen matriks B:\n";
                        for (int i = 0; i < size; ++i)
                        {
                            for (int j = 0; j < size; ++j)
                            {
                                cout << "B[" << i + 1 << "][" << j + 1 << "]: ";
                                cin >> B[i][j];
                                result[i][j] = A[i][j] + B[i][j];
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```

,
// Menampilkan hasil Pertambahan
cout << "Hasil pertambahan matrix A dan B:\n";
for (int i = 0; i < size; ++i)
{
    for (int j = 0; j < size; ++j)
    {
        cout << result[i][j] << " ";
    }
    cout << endl;
}
break;
case 'B':
case 'b':
    // Input matriks A
    cout << "Masukkan elemen matriks A:\n";
    for (int i = 0; i < size; ++i)
    {
        for (int j = 0; j < size; ++j)
        {
            cout << "A[" << i + 1 << "][" << j + 1 << "]: ";
            cin >> A[i][j];
        }

        result[i][j] = A[i][j] - B[i][j];
    }
    // Menampilkan hasil Pengurangan
    cout << "Hasil pengurangan matrix A dan B:\n";
    for (int i = 0; i < size; ++i)
    {
        for (int j = 0; j < size; ++j)
        {
            cout << result[i][j] << " ";
        }
        cout << endl;
    }
    break;
case 'C':
case 'c':
    // Kembali ke menu utama
    return;
default:
    cout << "Pilihan tidak valid." << endl;
}
}
break;

```

2. Case 2 (Transpose matriks)

Didalam case 2 juga terdapat switch case lainnya, switch case ini dibuat untuk membuat suatu pilihan ordo mana yang ingin kita transpose antara transpose ordo 2x2 atau ordo 3x3.

Case 2 juga menggunakan pernyataan while do (perulangan), while do tersebut memiliki fungsi sama seperti case 1, kita dapat mengulangi proses transpose matriks ataupun mengakhirinya. Pada case 2 ini dimasukkan sebuah perintah agar output yang dihasilkan berupa matriks sebelum dan sesudah transpose.

```
case 2:
while (true)
{
    tampilan_Sub_Menu_Transpose();
    char submenu;
    cout << "Choose: ";
    cin >> submenu;
    int A[2][2], B[3][3], result[size][size];
    switch (submenu)
    {
        case 'A':
        case 'a':
            // Input matrix
            cout << "Masukkan elemen matriks A:\n";
            for (int i = 0; i < 2; ++i)
            {
                for (int j = 0; j < 2; ++j)
                {
                    std::cout << "A[" << i + 1 << "][" << j + 1 << "]: ";
                    std::cin >> A[i][j];
                }
            }

            // Transpose matriks
            for (int i = 0; i < 2; ++i)
            {
                for (int j = 0; j < 2; ++j)
                {
                    result[j][i] = A[i][j];
                }
            }

            // Output matriks sebelum dan sesudah transpose
            std::cout << "Matriks sebelum transpose:\n";
            for (int i = 0; i < 2; ++i)
            {
                for (int j = 0; j < 2; ++j)
                {
                    std::cout << A[i][j] << " ";
                }
            }
            std::cout << "\n";
        }
    }
```

```

,
std::cout << "Matriks setelah transpose:\n";
for (int i = 0; i < 2; ++i)
{
    for (int j = 0; j < 2; ++j)
    {
        std::cout << result[i][j] << " ";
    }
    std::cout << "\n";
}

break;
case 'B':
case 'b':
    // Input matrix
    cout << "Masukkan elemen matriks A:\n";
    for (int i = 0; i < size; ++i)
    {
        for (int j = 0; j < size; ++j)
        {
            std::cout << "B[" << i + 1 << "][" << j + 1 << "]: ";
            std::cin >> B[i][j];
        }
    }

// Transpose matriks
for (int i = 0; i < size; ++i)
{
    for (int j = 0; j < size; ++j)
    {
        result[j][i] = B[i][j];
    }
}

// Output matriks sebelum dan sesudah transpose
std::cout << "Matriks sebelum transpose:\n";
for (int i = 0; i < size; ++i)
{
    for (int j = 0; j < size; ++j)
    {
        std::cout << B[i][j] << " ";
    }
    std::cout << "\n";
}
,

```

```

    ,
    std::cout << "Matriks setelah transpose:\n";
    for (int i = 0; i < size; ++i)
    {
        for (int j = 0; j < size; ++j)
        {
            std::cout << result[i][j] << " ";
        }
        std::cout << "\n";
    }
    break;
    case 'C':
    case 'c':
        // Kembali ke menu utama
        return;
    default:
        cout << "Pilihan tidak valid." << endl;
    }
}
break;

```

```

Pilih: 2
----- Perkalian dan Pembagian -----
A. 2x2
B. 3x3
C. Kembali ke Menu Utama
-----
Choose: a
Masukkan elemen matriks A:
A[1][1]: 3
A[1][2]: 6
A[2][1]: 8
A[2][2]: 4
Matriks sebelum transpose:
3 6
8 4
Matriks setelah transpose:
3 8
6 4
----- Perkalian dan Pembagian -----
A. 2x2
B. 3x3
C. Kembali ke Menu Utama
-----

Choose: b
Masukkan elemen matriks A:
B[1][1]: 8
B[1][2]: 6
B[1][3]: 3
B[2][1]: 6
B[2][2]: 8
B[2][3]: 3
B[3][1]: 9
B[3][2]: 8
B[3][3]: 8
Matriks sebelum transpose:
8 6 3
6 8 3
9 8 8
Matriks setelah transpose:
8 6 9
6 8 8
3 3 8
----- Perkalian dan Pembagian -----
A. 2x2
B. 3x3
C. Kembali ke Menu Utama
-----

```


3. Case 3 (Invers Matiks (2x2))

Case 3 hanya menggunakan pernyataan while do, tidak di masukkan switch case karena di menu pilihan ini hanya terdapat satu proses perhitungan.

```
case 3:
    while (true)
    {
        // Pilih submenu ukuran Matriks
        int A[2][2], B[3][3], result[size][size];
        cout << "Masukkan elemen matriks A (2x2):\n";
        for (int i = 0; i < 2; ++i)
        {
            for (int j = 0; j < 2; ++j)
            {
                cout << "A[" << i + 1 << "][" << j + 1 << "]: ";
                cin >> A[i][j];
            }
        }
        // Menghitung Matriks Determinan ukuran 2x2
        int det = A[0][0] * A[1][1] - A[0][1] * A[1][0];
        if (det == 0)
        {
            cout << "Determinant is zero. Inverse does not exist." << endl;
            return;
        }

        // Menghitung Matriks Inverse ukuran 2x2
        result[0][0] = A[1][1] / det;
        result[0][1] = -A[0][1] / det;
        result[1][0] = -A[1][0] / det;
        result[1][1] = A[0][0] / det;
        // Menampilkan Matriks Inverse ukuran 2x2
        cout << "Matriks inverse A (2x2):\n";
        for (int i = 0; i < 2; ++i)
        {
            for (int j = 0; j < 2; ++j)
            {
                cout << result[i][j] << " ";
            }
            cout << "\n";
        }
        cout << "\n";
        return;
    }
    break;
```

4. Case 4 (Determinan)

Untuk case 4 sama dengan case 1 dan 2. Pada case ini di berikan sebuah case untuk memilih ordo yang ingin di lakukan proses perhitungan dan pernyataan while do untuk memilih mengulangi perhitungan atau mengakhirinya.

```
case 4:
while (true)
{
    tampilan_Sub_Menu_Determinan();
    char submenu;
    cout << "Pilih: ";
    cin >> submenu;
    int A[size][size]; // Matriks A dideklarasikan di luar blok switch
    switch (submenu)
    {
        case 'A':
        case 'a':
            // Input matriks Determinan 2x2
            cout << "Masukkan elemen matriks A (2x2):\n";
            for (int i = 0; i < 2; ++i)
            {
                for (int j = 0; j < 2; ++j)
                {
                    cout << "A[" << i + 1 << "][" << j + 1 << "]: ";
                    cin >> A[i][j];
                }

                '\n';
            }
            // Hitung Matriks Determinan dan tampilkan hasil
            hitung_Determinan(A, 3);
            break;
        case 'C':
        case 'c':
            // Kembali ke menu utama
            return;
        default:
            cout << "Pilihan tidak valid." << endl;
    }
}
break;
```

5. Case 5 (Sistem Persamaan Linier (2x3))

Untuk case 5 output yang di hasilkan memiliki 2 opsi:

Opsi pertama ada 3 macam yaitu matriks koefisien yang di masukkan, matriks setelah eliminasi gauss dan solusi sistem persamaan linier.

Opsi yang kedua yaitu system persamaan linier tidak memiliki solusi unik.

6. Case 6 (Keluar)

Case ini adalah case untuk menutup atau mengakhiri program kalkulator matriks.

```
    }  
    case 6:  
        // Keluar dari program  
        cout << "Program selesai." << endl;  
        cout << "    Terima kasih!" << endl;  
        exit(0);  
    default:  
        cout << "Pilihan tidak valid." << endl;
```

BAB IV

PENGUJIAN

A. Menu Pilihan

```
C:\Users\rianz\OneDrive\Docu X + v

===== Menu Utama =====
1. Pertambahan dan Pengurangan Matriks (2x2)
2. Transpose Matriks
3. inverse Matriks (2x2)
4. Determinan
5. Sistem Persamaan Linier (2x3)
6. Keluar
=====
Pilih:
```

B. Pertambahan dan Pengurangan Matriks (2x2)

```
Pilih: 1
----- Pertambahan dan Pengurangan -----
A. Pertambahan
B. Pengurangan
C. Kembali ke Menu Utama
-----
Pilih: a
Masukkan elemen matriks A:
A[1][1]: 7
A[1][2]: 4
A[2][1]: 3
A[2][2]: 6
Masukkan elemen matriks B:
B[1][1]: 7
B[1][2]: 3
B[2][1]: 7
B[2][2]: 4
Hasil pertambahan matrix A dan B:
14 7
10 10
----- Pertambahan dan Pengurangan -----
A. Pertambahan
B. Pengurangan
C. Kembali ke Menu Utama
-----
Pilih: |
```

```
Pilih: 1
----- Pertambahan dan Pengurangan -----
A. Pertambahan
B. Pengurangan
C. Kembali ke Menu Utama
-----
Pilih: b
Masukkan elemen matriks A:
A[1][1]: 9
A[1][2]: 4
A[2][1]: 7
A[2][2]: 2
Masukkan elemen matriks B:
B[1][1]: 8
B[1][2]: 4
B[2][1]: 9
B[2][2]: 7
Hasil pengurangan matrix A dan B:
1 0
-2 -5
----- Pertambahan dan Pengurangan -----
A. Pertambahan
B. Pengurangan
C. Kembali ke Menu Utama
-----
Pilih: |
```

C. Transpose Matriks

```
=====
Pilih: 2
----- Perkalian dan Pembagian -----
A. 2x2
B. 3x3
C. Kembali ke Menu Utama
-----
Choose: a
Masukkan elemen matriks A:
A[1][1]: 9
A[1][2]: 4
A[2][1]: 2
A[2][2]: 7
Matriks sebelum transpose:
9 4
2 7
Matriks setelah transpose:
9 2
4 7
----- Perkalian dan Pembagian -----
A. 2x2
B. 3x3
C. Kembali ke Menu Utama
=====
```

```
=====
Choose: b
Masukkan elemen matriks A:
B[1][1]: 9
B[1][2]: 9
B[1][3]: 6
B[2][1]: 6
B[2][2]: 4
B[2][3]: 8
B[3][1]: 4
B[3][2]: 6
B[3][3]: 7
Matriks sebelum transpose:
9 9 6
6 4 8
4 6 7
Matriks setelah transpose:
9 6 4
9 4 6
6 8 7
----- Perkalian dan Pembagian -----
A. 2x2
B. 3x3
C. Kembali ke Menu Utama
=====
Choose:
```

D. Invers Matriks

```
=====
Pilih: 3
Masukkan elemen matriks A (2x2):
A[1][1]: 3
A[1][2]: -6
A[2][1]: -5
A[2][2]: 9
Matriks inverse A (2x2):
-3 -2
-1 -1

===== Menu Utama =====
1. Pertambahan dan Pengurangan Matriks (2x2)
2. Transpose Matriks
3. inverse Matriks (2x2)
4. Determinan
5. Sistem Persamaan Linier (2x3)
6. Keluar
=====
Pilih: |
```

E. Determinan

```
=====
Pilih: 4
----- Perkalian dan Pembagian -----
A. 2x2
B. 3x3
C. Kembali ke Menu Utama
-----
Pilih: a
Masukkan elemen matriks A (2x2):
A[1][1]: 9
A[1][2]: 4
A[2][1]: 7
A[2][2]: 4
Matriks A:
    9  4
    7  4
Determinan matriks A (2x2): 8
----- Perkalian dan Pembagian -----
A. 2x2
B. 3x3
C. Kembali ke Menu Utama
-----
```

```
Pilih: b
Masukkan elemen matriks A (3x3):
A[1][1]: 9
A[1][2]: 9
A[1][3]: 7
A[2][1]: 5
A[2][2]: 7
A[2][3]: 3
A[3][1]: 7
A[3][2]: 7
A[3][3]: 5
Matriks A:
    9  9  7
    5  7  3
    7  7  5
Determinan matriks A (3x3): -8
----- Perkalian dan Pembagian -----
A. 2x2
B. 3x3
C. Kembali ke Menu Utama
-----
Pilih: |
```

F. Sistem Persamaan Linier

```
=====
Pilih: 5
Masukkan matriks koefisien (2x3):
Masukkan elemen matriks[1][1]: 8
Masukkan elemen matriks[1][2]: 8
Masukkan elemen matriks[1][3]: 6
Masukkan elemen matriks[2][1]: 5
Masukkan elemen matriks[2][2]: 7
Masukkan elemen matriks[2][3]: 5

Matriks koefisien yang dimasukkan:
    8    8    6
    5    7    5

Matriks setelah eliminasi Gauss:
    8    8    6
    0    2    1.25

Solusi sistem persamaan linier:
x1 = 0.125
x2 = 0.625
```

```
2. Transpose Matriks
3. inverse Matriks (2x2)
4. Determinan
5. Sistem Persamaan Linier (2x3)
6. Keluar
=====
Pilih: 5
Masukkan matriks koefisien (2x3):
Masukkan elemen matriks[1][1]: 7
Masukkan elemen matriks[1][2]: 7
Masukkan elemen matriks[1][3]: 6
Masukkan elemen matriks[2][1]: 5
Masukkan elemen matriks[2][2]: 5
Masukkan elemen matriks[2][3]: 8

Matriks koefisien yang dimasukkan:
    7    7    6
    5    5    8

Sistem persamaan linier tidak memiliki solusi unik.
```

BAB V

PENUTUP

A. Kesimpulan

Jadi program kalkulator matriks ini dibuat untuk mengurangi beban manusia dalam perhitungan matriks, menawarkan solusi yang praktis. Program matriks ini menggunakan bahasa pemrograman C++ yang dirancang dalam waktu kurang lebih 1 bulan. Program ini masih berbentuk terminal, dan belum memiliki design User Interface.

Kalkulator matriks ini terdiri dari kalkulator Sistem persamaan linear, determinan matriks, tranpose matriks, invers matriks, dan oenjumlahan matriks. Kalkulatir ini dibuat berdasarkan rumus yang telah kami pelajari di semester 2 tahun 2023 mata kuliah Aljabar Geometri.

B. Saran

Saran yang kami miliki untuk tugas ini terdapat 2 aspek yakni aspek tugas dan pengembangan program dan aspek kekompakan kelompok

1. Tugas dan pengembangan program
 - a. Saat merancang program harus dari jauh jauh hari agar tidak tergesa gesa sehingga menimbulkan banyak kekeliruan.
 - b. Fahami terlebih dahulu materi yang akan diterapkan dalam program
 - c. Fahami bahasa program yang akan digunakan
2. Kekompakan kelompok
 - a. Hindari kerja dadakan.
 - b. Jangan menjadi beban kelompok
 - c. Selalu samakan persepsi dan jadwal dengan anggota kelompok, namun jika sudah diberi kesempatan untuk menyamakan jadwal, jangan menjadi orang yang egois.
 - d. Jika tidak bisa membantu minimal tawarkan apa yang bisa ditawarkan dari diri terhadap teman yang sedang berjuang meluluskan anda
 - e. Siapkan hati yang tenang saat melaksanakan kerja kelompok, dengan begitu kerja kelompok berjalan dengan baik walaupun dilaksanakan dengan santai mungkin.

C. Refleksi

Kekompakan kelompok adalah tonggak utama sukses nya tugas, namun selain itu yang utama bagi kelompok kami apa yang harus direfleksikan selanjutnya adalah dalam segi pengerjaan tugas yang terlalu mendadak. Maka dari itu, untuk kedepannya pengerjaan tugas yang mendadak harus dihindari, karena sebagus apapun kerjasama dan pembagian tugas, serta kontribusi tersonal terhadap kelompok, akan kurang sempurna jika segalanya dilakukan secara mendadak