

Nama : Dwi Prabowo
NIM : A11.2021.13761
Klpk : A11.43UG2

Penjelasan Coding Program Matrix

- **Kelas Matrix**

Saya membuat kelas bernama matrix yang didalamnya terdapat :

- 5 variabel anggota : baris, kolom, scalar, matrix, dan result.
- Default constructor : fungsi matrix() akan terkeksekusi jika ada kelas matrix yang terbentuk.
- Operator overload : untuk mengenalkan operator pada saat terjadi operasi perhitungan antar kelas matrix.

Didalamnya juga terdapat fungsi-fungsi berbagai operasi matriks.

```
class Matrix
{
public:
    int baris, kolom, scalar;
    float matrix[20][20];
    int result[10][10];

public:
    // Default Constructor.
    Matrix()
    {
        baris = 0;
        kolom = 0;
        scalar = 0;
        for (int i = 0; i < baris; i++)
        {
            for (int j = 0; j < kolom; j++)
            {
                matrix[i][j] = 0;
                result[i][j] = 0;
            }
        }
    }

    // Operator overload.
    float &operator()(int baris, int kolom)
    {
        return matrix[baris][kolom]; // digunakan untuk mengenalkan operator pada
saat terjadi operasi perhitungan antar kelas matrik
    }
}
```

FUNGSI-FUNGSI YANG ADA DIDALAM KELAS MATRIX :

- **Fungsi getInput(char namaMatrix)**

Fungsi ini untuk menyimpan data inputan dari matrix yang ingin dibuat. Maksimal ordo matriks yang dapat diinputkan adalah 10x10. Untuk matriks baris 1, kolom 1 akan disimpan dalam variabel array 2 dimensi bernama matrix elemen ke-[1][1]. Jika inputan melebihi ordo 10x10 maka akan memberikan output "Inputan tidak valid".

```
// Input Matrix
void getInput(char namaMatrix)
{
    do
    {
        cout << "Masukkan Banyak Baris Matrix " << namaMatrix << " = ";
        cin >> baris;
        cout << "Masukkan Banyak Kolom Matrix " << namaMatrix << " = ";
        cin >> kolom;
        if (baris < 1 || kolom < 1 || baris > 10 || kolom > 10)
        {
            cout << "Inputan tidak valid. Min size = 1x1 and max size = 10x10.\n";
        }
    } while (baris < 1 || kolom < 1 || baris > 10 || kolom > 10);

    cout << "Urutan input matrix dari kiri atas ke kanan bawah\n";
    for (int i = 1; i <= baris; i++)
    {
        for (int j = 1; j <= kolom; j++)
        {
            cout << "Matrix Input (baris ke-" << i << " , kolom ke-" << j << ") = ";
            cin >> matrix[i][j];
        }
    }
}
```

- **Fungsi displayMatrix()**

Untuk melakukan print output dari matrix.

```
// Print/Output matrix
void displayMatrix()
{
    for (int i = 1; i <= baris; i++)
    {
        cout << "| ";
        for (int j = 1; j <= kolom; j++)
        {
            cout << matrix[i][j];
            cout << setw(5);
        }
        cout << " |";
        cout << endl;
    }
}
```

- **Fungsi displayResultMatrix(int baris, int kolom)**

Untuk melakukan print output matrix hasil dari perkalian dan transpose dengan parameter baris dan kolom.

```
// Print hasil Operasi Matrix
void displayResultMatrix(int baris, int kolom)
{
    for (int i = 1; i <= baris; i++)
    {
        cout << "|   ";
        for (int j = 1; j <= kolom; j++)
        {
            cout << result[i][j];
            cout << setw(5);
        }
        cout << " |";
        cout << endl;
    }
}
```

- **Fungsi perkalianSkalar()**

Untuk melakukan operasi perkalian matrix yang diinputkan sebelumnya dengan bilangan skalar. Dengan perulangan, Bilangan skalar akan dikalikan pada setiap elemen yang terdapat nilai sesuai dengan jumlah baris dan kolom matrix.

```
// perkalian skalar
void perkalianSkalar()
{
    cout << "Masukkan nilai skalar untuk mengalikan matrix: ";
    cin >> scalar;
    for (int i = 1; i <= baris; i++)
    {
        for (int j = 1; j <= kolom; j++)
        {
            matrix[i][j] *= scalar;
        }
    }
    cout << "hasil perkalian matrix dengan skalar " << scalar << " : " << endl;
    displayMatrix();
}
```

- **Fungsi penjumlahanMatrix(Matrix b)**

Untuk melakukan operasi penjumlahan antar matriks a sebagai matrix pertama dengan parameter matriks b sebagai matriks kedua. Ada 2 kondisi :

- Jika jumlah baris matriks a == jumlah baris matriks b dan jumlah kolom matriks a == jumlah kolom matriks b, maka akan melakukan 2 perulangan, perulangan pertama dilakukan pada baris matriks sedangkan perulangan kedua(didalam perulangan pertama) dilakukan pada kolom matriks sekaligus untuk menyimpan hasil dari penjumlahan kedua matriks.
- Jika tidak memenuhi kondisi, maka akan memberikan output "Ordo tidak sama".

```
// penjumlahan Matrix
void penjumlahanMatrix(Matrix b)
{
    if (baris == b.baris && kolom == b.kolom)
    {
        for (int i = 1; i <= baris; i++)
        {
            for (int j = 1; j <= kolom; j++)
            {
                matrix[i][j] += b(i, j);
            }
        }
        cout << "hasil penjumlahan : " << endl;
        displayMatrix();
    }
    else
    {
        cout << "Ordo tidak sama. Operasi penjumlahan tidak bisa dilakukan antar matriks\n";
    }
}
```

- **Fungsi penguranganMatrix(Matrix b)**

Untuk melakukan operasi pengurangan antar matriks dengan parameter matriks b sebagai matriks ke-2. Ada 2 kondisi :

- Jika jumlah baris matriks a == jumlah baris matriks b dan jumlah kolom matriks a == jumlah kolom matriks b, maka akan melakukan 2 perulangan, perulangan pertama dilakukan pada baris matriks sedangkan perulangan kedua(didalam perulangan pertama) dilakukan pada kolom matriks sekaligus untuk menyimpan hasil dari pengurangan kedua matriks.
- Jika tidak memenuhi kondisi, maka akan memberikan output "Ordo tidak sama".

```
// Pengurangan Matrix
void penguranganMatrix(Matrix b)
{
    if (baris == b.baris && kolom == b.kolom)
    {
        for (int i = 1; i <= baris; i++)
        {
            for (int j = 1; j <= kolom; j++)
            {
                matrix[i][j] -= b(i, j);
            }
        }
        cout << "hasil pengurangan : " << endl;
        displayMatrix();
    }
    else
    {
        cout << "Ordo tidak sama. Operasi pengurangan tidak bisa dilakukan antar matriks.\n\n";
    }
}
```

- **Fungsi perkalianMatrix(Matrix b)**

Untuk melakukan operasi perkalian antar matriks dengan parameter matriks b sebagai matriks ke-2. Ada 2 kondisi :

- Jika jumlah kolom matriks a == jumlah baris matriks b, maka akan melakukan 3 perulangan :
 - ❖ perulangan pertama dilakukan pada baris matriks.
 - ❖ perulangan kedua(didalam perulangan pertama) dilakukan pada kolom matriks matriks b.
 - ❖ Perulangan ketiga(didalam perulangan kedua) dilakukan pada baris matriks b sekaligus untuk menyimpan hasil dari perkalian kedua matriks kedalam variabel kelas matriks yaitu result.
- Jika tidak memenuhi kondisi, maka akan memberikan output "Matriks tidak bisa dikalikan satu sama lain".

Hasil dari operasi perkalian akan diprint melalui fungsi displayResultMatrix().

```

// Perkalian Matrix
void perkalianMatrix(Matrix b)
{
    if (kolom == b.baris)
    {
        for (int i = 1; i <= baris; i++)
        {
            for (int j = 1; j <= b.kolom; j++)
            {
                result[i][j] = 0;
                for (int k = 1; k <= b.baris; k++)
                {
                    result[i][j] += matrix[i][k] * b(k, j); // mengalikan matrix
dari kelas dengan matrix inputan
                }
            }
        }
        cout << "hasil perkalian : " << endl;
        displayResultMatrix(baris, b.kolom);
    }
    else
    {
        cout << "Matriks tidak bisa dikalikan satu sama lain (kolom matriks A !=
baris Matriks B)\n\n";
    }
}

```

- **Fungsi transpose()**

Untuk melakukan transpose pada matrix yang diinputkan sebelumnya. Variable temp berguna sebagai variabel sementara untuk memindah data jumlah baris menjadi data jumlah kolom dan sebaliknya. Ada 2 perulangan:

- Perulangan pertama dilakukan sebanyak jumlah kolom matriks.
- Perulangan kedua(didalam perulangan pertama) dilakukan sebanyak jumlah baris matriks sekaligus menyimpan hasil transpose matriks.

Hasil dari operasi perkalian akan diprint melalui fungsi displayResultMatrix().

```

// Transpose matrix
void transpose()
{
    int temp = baris;
    baris = kolom;
    kolom = temp;
    for (int i = 1; i <= baris; i++)
    {
        for (int j = 1; j <= kolom; j++)
        {
            result[i][j] = matrix[j][i]; // baris = i, kolom = j
        }
    }
}

```

```

    }
    cout << "hasil transpose : " << endl;

    displayResultMatrix(baris, kolom);
}

```

- **Fungsi carideterminan()**

Saya membatasi maksimal matriks ordo 4x4 Untuk mencari determinan dari matriks yang sudah diinputkan sebelumnya. Terdapat 2 kondisi :

- Jika jumlah baris == jumlah kolom maka akan melakukan pengecekan kondisi :
 - ❖ Apakah jumlah baris <= 4 ? jika iya, maka akan melakukan pengecekan kondisi sbb :
 - a) Jika jumlah baris == 1, maka hanya memiliki satu nilai jadi determinannya merupakan elemen pertama yaitu [1][1].
 - b) Jika jumlah baris == 2, maka determinan matriks merupakan hasil dari $(\text{matrix}[1][1] * \text{matrix}[2][2]) - (\text{matrix}[2][1] * \text{matrix}[1][2])$ atau $ad - bc$.
 - c) Jika jumlah baris == 3, maka akan menentukan determinan matriks menggunakan metode ekspansi baris pertama.
 - d) Jika jumlah baris == 4, maka akan menentukan determinan matriks menggunakan metode ekspansi baris pertama kemudian menggunakan metode saurus.
 - ❖ Jika tidak maka akan memberikan output "operasi tidak dapat dilakukan. Ordo matriks > 4x4".
- Jika jumlah baris != jumlah kolom maka akan memberika output "baris != kolom. Matriks tidak memiliki determinan!".

```

// Cari Determinan Matrix
int carideterminan()
{
    float determinan = 0;
    float hasil1, hasil2, hasil3, hasil4;
    if (baris == kolom)
    {
        if (baris <= 4)
        {
            if (baris == 1) // nilai hanya 1
            {
                determinan = matrix[1][1];
            }
            else if (baris == 2) // ad-bc
            {
                determinan = (matrix[1][1] * matrix[2][2]) - (matrix[2][1] *
matrix[1][2]);
            }
            else if (baris == 3) // metode ekspansi baris 1
            {
                determinan = matrix[1][1] * (matrix[2][2] * matrix[3][3] - matrix[2][3] *
matrix[3][2]) - matrix[1][2] * (matrix[2][2][1] * matrix[3][3] - matrix[2][3] *

```

```

matrix[3][1]) + matrix[1][3] * (matrix[2][1] * matrix[3][2] - matrix[2][2] *
matrix[3][1]));
    }
    else if (baris == 4) // metode ekspansi baris 1
    {
        for (int i = 1; i <= baris; i++)
        {
            for (int j = 1; j <= kolom; j++) // metode saurus
            {
                hasil1 = (matrix[2][2] * matrix[3][3] * matrix[4][4]) +
(matrix[2][3] * matrix[3][4] * matrix[4][2]) + (matrix[2][4] * matrix[3][2] *
matrix[4][3]) - (matrix[4][2] * matrix[3][3] * matrix[2][4]) - (matrix[4][3] *
matrix[3][4] * matrix[2][2]) - (matrix[4][4] * matrix[3][2] * matrix[2][3]);
                hasil2 = (matrix[2][1] * matrix[3][3] * matrix[4][4]) +
(matrix[2][3] * matrix[3][4] * matrix[4][1]) + (matrix[2][4] * matrix[3][1] *
matrix[4][3]) - (matrix[4][1] * matrix[3][3] * matrix[2][4]) - (matrix[4][3] *
matrix[3][4] * matrix[2][1]) - (matrix[4][4] * matrix[3][1] * matrix[2][3]);
                hasil3 = (matrix[2][1] * matrix[3][2] * matrix[4][4]) +
(matrix[2][2] * matrix[3][4] * matrix[4][1]) + (matrix[2][4] * matrix[3][1] *
matrix[4][2]) - (matrix[3][2] * matrix[2][4] * matrix[4][1]) - (matrix[4][2] *
matrix[3][4] * matrix[2][1]) - (matrix[4][4] * matrix[3][1] * matrix[2][2]);
                hasil4 = (matrix[2][1] * matrix[3][2] * matrix[4][3]) +
(matrix[2][2] * matrix[3][3] * matrix[4][1]) + (matrix[2][3] * matrix[3][1] *
matrix[4][2]) - (matrix[4][1] * matrix[3][2] * matrix[2][3]) - (matrix[4][2] *
matrix[3][3] * matrix[2][1]) - (matrix[4][3] * matrix[3][1] * matrix[2][2]);
            }
            // hitung determinan matriks
            determinan = (hasil1 * matrix[1][1] * 1) + (hasil2 * matrix[1][2] * -
1) + (hasil3 * matrix[1][3] * 1) + (hasil4 * matrix[1][4] * -1);
        }
    }
    cout << "Determinan matrix : " << determinan;
}
else
{
    cout << "Operasi tidak dapat dilakukan. ordo matriks > 4x4" << endl;
}
}
else
{
    cout << "baris != kolom. Matriks tidak memiliki determinan!" << endl;
}
cout << '\n';
return determinan;
}

```


- **Fungsi inversMatrix()**

Untuk mencari invers dari matriks yang diinputkan. Ada 2 kondisi :

- Jika jumlah baris == jumlah kolom, maka akan melakukan pengecekan kondisi :
 - ❖ Apakah determinan dari matriks == 0 ? . Jika iya maka akan memberikan output "Determinan = 0. Matriks tidak memiliki invers!".
 - ❖ Jika tidak memenuhi kondisi maka dapat dipastikan determinan tidak sama dengan 0 dan akan mencari invers matriks dengan transformasi baris elementer.
- Jika tidak memenuhi kondisi maka bisa dipastikan jumlah baris tidak sama dengan jumlah kolom dan akan memberikan output "ordo tidak sama. Matriks tidak memiliki invers!".

```
void inversMatrix()
{
    int i, j, k, l;
    if (baris == kolom)
    {
        if (baris <= 4)
        {
            if (carideterminan() == 0)
            {
                cout << "Determinan = 0. Matriks tidak memiliki invers!" << endl;
            }
            else
            {
                // Matriks sisi kanan
                for (j = baris + 1; j <= baris + baris; j++)
                {
                    i = j - baris;
                    matrix[i][j] = 1;
                }
                for (j = baris + 1; j <= baris + baris; j++)
                {
                    for (i = 1; i <= baris; i++)

                        if (i != (j - baris))
                            matrix[i][j] = 0;
                }
                // Proses penginversan
                for (i = 1; i <= baris; i++)
                {
                    for (j = 1; j <= baris + baris; j++)
                    {
                        if (i != j)
                            matrix[i][j] = matrix[i][j] / matrix[i][i];
                    }
                    for (j = 1; j <= baris + baris; j++)
                    {
                        if (i == j)
                            matrix[i][j] = 1;
                    }
                }
            }
        }
    }
}
```

```

    }
    // Penjumlahan kesatu baris element
    for (l = 1; l <= baris; l++)
    {
        if (i != l)
        {
            for (j = i + 1; j <= baris + baris; j++)
            {
                matrix[l][j] = matrix[l][j] - (matrix[i][j] *
matrix[l][i]);
            }
        }
    }
    // Pembuat nol disekitar matriks kiri
    for (k = 1; k <= baris; k++)
    {
        if (i != k)
        {
            matrix[k][i] = 0;
        }
    }
}
cout.precision(4);
cout.setf(ios::fixed);
// Print
for (i = 1; i <= baris; i++)
{
    for (j = baris + 1; j <= baris + baris; j++)
        cout << "    " << matrix[i][j];
    cout << endl;
}
}
else
{
    cout << "Operasi tidak dapat dilakukan. ordo matriks > 4x4" << endl;
}
}
else
{
    cout << "baris != kolom. Matriks tidak memiliki invers!" << endl;
}
}
};

```

- **Interface Program**

Tampilan antarmuka dari program. Terdapat 8 menu dalam tampilan menu utama yang dapat dipilih, jika user menginputkan nilai selain angka 1-8 maka akan memberikan output "Inputan tidak valid, pilihan tidak tersedia!". Silahkan input kembali!" kemudian akan kembali ke menu utama.

Setelah user melakukan operasi matriks, akan diberikan pilihan apakah user ingin kembali ke menu utama untuk menginputkan matriks lagi ? jika iya, user perlu menginputkan kata "iya" atau "IYA. Jika tidak, user perlu menginputkan kata "tidak" atau "Tidak" untuk mengakhiri program.

```
int main()
{
    int pilihan, baris, kolom;
    char A = 'A', B = 'B';
    string lanjut;
    Matrix a, b;
    do
    {
        system("cls");
        cout << "=====\\n"
            << "|                Matrix Program by Dwi Prabowo                |\\n"
            << "|                                Main Menu                                |\\n"
            << "=====\\n";
        cout << "-----\\n";
        cout << "|  Pilih 1 untuk perkalian skalar                |\\n"
            << "|  Pilih 2 untuk penjumlahan matrix              |\\n"
            << "|  Pilih 3 untuk pengurangan matrix              |\\n"
            << "|  Pilih 4 untuk perkalian matrix                |\\n"
            << "|  Pilih 5 untuk transpose matrix                |\\n"
            << "|  Pilih 6 untuk mencari determinan matrix (max ordo 4x4) |\\n"
            << "|  Pilih 7 untuk mencari invers matrix            |\\n"
            << "-----\\n";

        cout << "Masukkan Pilihan Anda: ";
        cin >> pilihan;
        if (pilihan == 1)
        {
            system("cls");
            a.getInput(A);
            cout << "Matrix A : " << endl;
            a.displayMatrix();
            cout << endl;
            a.perkalianSkalar();
        }
        else if (pilihan == 2)
        {
            system("cls");
            a.getInput(A);
            cout << "Matrix A : " << endl;
            a.displayMatrix();
            cout << endl;
        }
    }
}
```

```

        b.getInput(B);
        cout << "Matrix B : " << endl;
        b.displayMatrix();
        cout << endl;
        a.penjumlahanMatrix(b);
    }
    else if (pilihan == 3)
    {
        system("cls");
        a.getInput(A);
        cout << "Matrix A : " << endl;
        a.displayMatrix();
        cout << endl;
        b.getInput(B);
        cout << "Matrix B : " << endl;
        b.displayMatrix();
        cout << endl;
        a.penguranganMatrix(b);
    }

    else if (pilihan == 4)
    {
        system("cls");
        a.getInput(A);
        cout << "Matrix A : " << endl;
        a.displayMatrix();
        cout << endl;
        b.getInput(B);
        cout << "Matrix B : " << endl;
        b.displayMatrix();
        cout << endl;
        a.perkalianMatrix(b);
    }

    else if (pilihan == 5)
    {
        system("cls");
        a.getInput(A);
        cout << "Matrix A : " << endl;
        a.displayMatrix();
        cout << endl;
        a.transpose();
    }

    else if (pilihan == 6)
    {
        system("cls");
        a.getInput(A);
        cout << "Matrix A : " << endl;
        a.displayMatrix();
        cout << endl;
        a.carideterminan();
    }

```

```

        cout << endl;
    }
    else if (pilihan == 7)
    {
        system("cls");
        a.getInput(A);
        cout << "Matrix A : " << endl;
        a.displayMatrix();
        cout << endl;
        a.inversMatrix();
        cout << endl;
    }
    else
    {
        cout << "Inputan tidak valid, Pilihan Tidak tersedia!. Silahkan Input kembali!\n";
    }
    cout << "ingin melanjutkan input matriks? (Ketik iya atau tidak) ";
    cin.ignore();
    getline(cin, lanjut);
} while (lanjut == "iya" || lanjut == "IYA");

system("pause");
return EXIT_SUCCESS;
}

```

Link Github : <https://github.com/bensarkiel/calMatrix.git>