

Nama : Matthew Aldhino Sirait

NPM : 4522210087

I. Pecahan.java

- Source Code

```
class Pecahan {
    private int numerator;
    private int denominator;

    public Pecahan(int numerator, int denominator) {
        this.numerator = numerator;
        this.denominator = denominator;
    }

    public Pecahan add(Pecahan other) {
        int newNumerator = this.numerator * other.denominator +
other.numerator * this.denominator;
        int newDenominator = this.denominator * other.denominator;
        return new Pecahan(newNumerator, newDenominator);
    }

    public Pecahan subtract(Pecahan other) {
        int newNumerator = this.numerator * other.denominator -
other.numerator * this.denominator;
        int newDenominator = this.denominator * other.denominator;
        return new Pecahan(newNumerator, newDenominator);
    }

    public Pecahan multiply(Pecahan other) {
        int newNumerator = this.numerator * other.numerator;
        int newDenominator = this.denominator * other.denominator;
        return new Pecahan(newNumerator, newDenominator);
    }

    public Pecahan divide(Pecahan other) {
        int newNumerator = this.numerator * other.denominator;
        int newDenominator = this.denominator * other.numerator;
        return new Pecahan(newNumerator, newDenominator);
    }

    public void simplify() {
        int gcd = gcd(this.numerator, this.denominator);
        this.numerator /= gcd;
        this.denominator /= gcd;
    }
}
```

```

private int gcd(int a, int b) {
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}

@Override
public String toString() {
    return numerator + "/" + denominator;
}

public static void main(String[] args) {
    int numerator1 = 3;
    int denominator1 = 4;
    int numerator2 = 1;
    int denominator2 = 2;

    Pecahan pecahan1 = new Pecahan(numerator1, denominator1);
    Pecahan pecahan2 = new Pecahan(numerator2, denominator2);

    Pecahan result = pecahan1.add(pecahan2);
    System.out.println("Penjumlahan: " + result);

    result = pecahan1.subtract(pecahan2);
    System.out.println("Pengurangan: " + result);

    result = pecahan1.multiply(pecahan2);
    System.out.println("Perkalian: " + result);

    result = pecahan1.divide(pecahan2);
    System.out.println("Pembagian: " + result);
}
}

```

II. Matrix.java

- Source Code

```

public class Matrix {
    private int[][] matrix;

    public Matrix(int[][] matrix) {
        this.matrix = matrix;
    }
}

```

```
public Matrix add(Matrix other) {
    int rows = matrix.length;
    int cols = matrix[0].length;
    int[][] result = new int[rows][cols];

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            result[i][j] = matrix[i][j] + other.matrix[i][j];
        }
    }

    return new Matrix(result);
}

public Matrix subtract(Matrix other) {
    int rows = matrix.length;
    int cols = matrix[0].length;
    int[][] result = new int[rows][cols];

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            result[i][j] = matrix[i][j] - other.matrix[i][j];
        }
    }

    return new Matrix(result);
}

public Matrix dot(Matrix other) {
    int rows = matrix.length;
    int cols = other.matrix[0].length;
    int[][] result = new int[rows][cols];

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            int sum = 0;
            for (int k = 0; k < matrix[0].length; k++) {
                sum += matrix[i][k] * other.matrix[k][j];
            }
            result[i][j] = sum;
        }
    }

    return new Matrix(result);
}

public Matrix transpose() {
    int rows = matrix[0].length;
```

```

        int cols = matrix.length;
        int[][] result = new int[rows][cols];

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                result[i][j] = matrix[j][i];
            }
        }

        return new Matrix(result);
    }

    public Matrix inverse() {
        // Implementasi invers matriks di sini

        // Sementara ini mengembalikan matriks asli sebagai contoh
        return this;
    }

    @Override
    public String toString() {
        StringBuilder sb = new StringBuilder();
        int rows = matrix.length;
        int cols = matrix[0].length;

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                sb.append(matrix[i][j]);
                sb.append(" ");
            }
            sb.append("\n");
        }

        return sb.toString();
    }

    public static void main(String[] args) {
        int[][] data1 = {{1, 2}, {3, 4}};
        int[][] data2 = {{5, 6}, {7, 8}};

        Matrix matrix1 = new Matrix(data1);
        Matrix matrix2 = new Matrix(data2);

        // Operasi penjumlahan
        Matrix additionResult = matrix1.add(matrix2);
        System.out.println("Hasil Penjumlahan:");
        System.out.println(additionResult);
    }

```

```

        // Operasi pengurangan
        Matrix subtractionResult = matrix1.subtract(matrix2);
        System.out.println("Hasil Pengurangan:");
        System.out.println(subtractionResult);

        // Operasi dot matriks
        Matrix dotProductResult = matrix1.dot(matrix2);
        System.out.println("Hasil Dot Matriks:");
        System.out.println(dotProductResult);

        // Operasi transpose
        Matrix transposeResult = matrix1.transpose();
        System.out.println("Hasil Transpose:");
        System.out.println(transposeResult);

        // Operasi invers
        Matrix inverseResult = matrix1.inverse();
        System.out.println("Hasil Invers:");
        System.out.println(inverseResult);
    }
}

```

III. Main.java

- Source Code

```

public class Main {
    public static class Pecahan {
        private int numerator;
        private int denominator;

        public Pecahan(int numerator, int denominator) {
            this.numerator = numerator;
            this.denominator = denominator;
        }

        public Pecahan add(Pecahan other) {
            int newNumerator = this.numerator * other.denominator +
other.numerator * this.denominator;
            int newDenominator = this.denominator * other.denominator;
            return new Pecahan(newNumerator, newDenominator);
        }

        public Pecahan subtract(Pecahan other) {
            int newNumerator = this.numerator * other.denominator -
other.numerator * this.denominator;
            int newDenominator = this.denominator * other.denominator;
            return new Pecahan(newNumerator, newDenominator);
        }
    }
}

```

```

public Pecahan multiply(Pecahan other) {
    int newNumerator = this.numerator * other.numerator;
    int newDenominator = this.denominator * other.denominator;
    return new Pecahan(newNumerator, newDenominator);
}

public Pecahan divide(Pecahan other) {
    int newNumerator = this.numerator * other.denominator;
    int newDenominator = this.denominator * other.numerator;
    return new Pecahan(newNumerator, newDenominator);
}

@Override
public String toString() {
    return numerator + "/" + denominator;
}
}

public static class Matrix {
    private int[][] matrix;

    public Matrix(int[][] matrix) {
        this.matrix = matrix;
    }

    public Matrix add(Matrix other) {
        int rows = matrix.length;
        int cols = matrix[0].length;
        int[][] newMatrix = new int[rows][cols];
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                newMatrix[i][j] = this.matrix[i][j] + other.matrix[i][j];
            }
        }
        return new Matrix(newMatrix);
    }

    public Matrix subtract(Matrix other) {
        int rows = matrix.length;
        int cols = matrix[0].length;
        int[][] newMatrix = new int[rows][cols];
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                newMatrix[i][j] = this.matrix[i][j] - other.matrix[i][j];
            }
        }
        return new Matrix(newMatrix);
    }
}

```

```

    }

    public Matrix dot(Matrix other) {
        int rows = matrix.length;
        int cols = other.matrix[0].length;
        int[][] newMatrix = new int[rows][cols];
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                for (int k = 0; k < matrix[0].length; k++) {
                    newMatrix[i][j] += this.matrix[i][k] *
other.matrix[k][j];
                }
            }
        }
        return new Matrix(newMatrix);
    }

    public Matrix transpose() {
        int rows = matrix.length;
        int cols = matrix[0].length;
        int[][] newMatrix = new int[cols][rows];
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                newMatrix[j][i] = this.matrix[i][j];
            }
        }
        return new Matrix(newMatrix);
    }

    public Matrix inverse() {
        int rows = matrix.length;
        int cols = matrix[0].length;

        // Kode untuk menghitung invers matriks menggunakan metode adjoint
        // Implementasi metode adjoint di sini
        // ...

        int[][] newMatrix = new int[cols][rows]; // Ganti dengan hasil
invers yang benar
        return new Matrix(newMatrix);
    }

    @Override
    public String toString() {
        StringBuilder sb = new StringBuilder();
        int rows = matrix.length;
        int cols = matrix[0].length;

```

```

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                sb.append(matrix[i][j]);
                sb.append(" ");
            }
            sb.append("\n");
        }

        return sb.toString();
    }
}

public static void main(String[] args) {
    // Operasi Pecahan
    Pecahan pecahan1 = new Pecahan(3, 4);
    Pecahan pecahan2 = new Pecahan(1, 2);

    Pecahan result = pecahan1.add(pecahan2);
    System.out.println("Penjumlahan Pecahan: " + result);

    result = pecahan1.subtract(pecahan2);
    System.out.println("Pengurangan Pecahan: " + result);

    result = pecahan1.multiply(pecahan2);
    System.out.println("Perkalian Pecahan: " + result);

    result = pecahan1.divide(pecahan2);
    System.out.println("Pembagian Pecahan: " + result);

    // Operasi Matrix
    int[][] data1 = {{1, 2}, {3, 4}};
    int[][] data2 = {{5, 6}, {7, 8}};

    Matrix matrix1 = new Matrix(data1);
    Matrix matrix2 = new Matrix(data2);

    Matrix additionResult = matrix1.add(matrix2);
    System.out.println("Hasil Penjumlahan Matriks:");
    System.out.println(additionResult);

    Matrix subtractionResult = matrix1.subtract(matrix2);
    System.out.println("Hasil Pengurangan Matriks:");
    System.out.println(subtractionResult);

    Matrix dotProductResult = matrix1.dot(matrix2);
    System.out.println("Hasil Dot Product Matriks:");
    System.out.println(dotProductResult);
}

```



```

Matrix transposeResult = matrix1.transpose();
System.out.println("Hasil Transpose Matriks:");
System.out.println(transposeResult);

Matrix inverseResult = matrix1.inverse();
System.out.println("Hasil Invers Matriks:");
System.out.println(inverseResult);
}
}

```

- Hasil Running

```

PS C:\Users\328ma.LAPTOP-IADCH63F\OneDrive\Documents\BELAJAR\java\Teori 1> & 'C:\Program Files\Java\jre-1.8\bin\java
.exe' -cp 'C:\Users\328ma.LAPTOP-IADCH63F\AppData\Roaming\Code\User\workspaceStorage\973d8932915425228617ea967cae5d
4a\redhat.java\jdt_ws\Teori_1_2f365584\bin' 'Main'

Penjumlahan Pecahan: 10/8
Pengurangan Pecahan: 2/8
Perkalian Pecahan: 3/8
Pembagian Pecahan: 6/4
Hasil Penjumlahan Matriks:
6 8
10 12

Hasil Pengurangan Matriks:
-4 -4
-4 -4

Hasil Dot Product Matriks:
19 22
43 50

Hasil Transpose Matriks:
1 3
2 4

Hasil Invers Matriks:
0 0
0 0

```

IV. GitHub

- Link: https://github.com/MatthewAldhinoSirait/4522210087_MatthewAldhinoSirait