

Nama : Garin Hanggario

NPM : 4517210015

Pemrograman Berorientasi Objek - A

SOAL 1

Source code untuk class Pecahan yang terdapat operasi penjumlahan, pengurangan, perkalian, dan pembagian :

```
public class Pecahan {
    private int pembilang;
    private int penyebut;

    public Pecahan(int pembilang, int penyebut) {
        if (penyebut == 0) {
            throw new IllegalArgumentException("Penyebut tidak boleh nol");
        }
        this.pembilang = pembilang;
        this.penyebut = penyebut;
    }

    public Pecahan tambah(Pecahan other) {
        int newPenyebut = this.penyebut * other.penyebut;
        int newPembilang = (this.pembilang * other.penyebut) + (other.pembilang *
this.penyebut);
        return sederhana(newPembilang, newPenyebut);
    }

    public Pecahan kurang(Pecahan other) {
        int newPenyebut = this.penyebut * other.penyebut;
        int newPembilang = (this.pembilang * other.penyebut) - (other.pembilang *
this.penyebut);
        return sederhana(newPembilang, newPenyebut);
    }

    public Pecahan kali(Pecahan other) {
        int newPembilang = this.pembilang * other.pembilang;
        int newPenyebut = this.penyebut * other.penyebut;
        return sederhana(newPembilang, newPenyebut);
    }

    public Pecahan bagi(Pecahan other) {
        if (other.pembilang == 0) {
            throw new IllegalArgumentException("Pembilang pembagi tidak boleh
nol");
        }
    }
}
```

```

        int newPembilang = this.pembilang * other.penyebut;
        int newPenyebut = this.penyebut * other.pembilang;
        return sederhana(newPembilang, newPenyebut);
    }

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

    private Pecahan sederhana(int pembilang, int penyebut) {
        int gcd = gcd(pembilang, penyebut);
        return new Pecahan(pembilang / gcd, penyebut / gcd);
    }

    @Override
    public String toString() {
        return pembilang + "/" + penyebut;
    }

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

        Pecahan hasilTambah = pecahan1.tambah(pecahan2);
        Pecahan hasilKurang = pecahan1.kurang(pecahan2);
        Pecahan hasilKali = pecahan1.kali(pecahan2);
        Pecahan hasilBagi = pecahan1.bagi(pecahan2);

        System.out.println("Hasil dari penjumlahan: " + pecahan1 + " + " +
            pecahan2 + " = " + hasilTambah);
        System.out.println("Hasil dari pengurangan: " + pecahan1 + " - " +
            pecahan2 + " = " + hasilKurang);
        System.out.println("Hasil dari perkalian: " + pecahan1 + " * " + pecahan2
            + " = " + hasilKali);
        System.out.println("Hasil dari pembagian: " + pecahan1 + " / " + pecahan2
            + " = " + hasilBagi);
    }
}

```

Screenshot hasil run :

The screenshot displays an IDE with a Java file named `Pecahan.java` and its execution output in the terminal.

Java Code (Pecahan.java):

```
54 @Override
55 public String toString() {
56     return pembilang + "/" + penyebut;
57 }
58
59 Run | Debug
60 public static void main(String[] args) {
61     Pecahan pecahan1 = new Pecahan(pembilang:1, penyebut:2);
62     Pecahan pecahan2 = new Pecahan(pembilang:3, penyebut:4);
63
64     Pecahan hasilTambah = pecahan1.tambah(pecahan2);
65     Pecahan hasilKurang = pecahan1.kurang(pecahan2);
66     Pecahan hasilKali = pecahan1.kali(pecahan2);
67     Pecahan hasilBagi = pecahan1.bagi(pecahan2);
68
69     System.out.println("Hasil dari penjumlahan: " + pecahan1 + " + " + pecahan2 + " = " + hasilTambah);
70     System.out.println("Hasil dari pengurangan: " + pecahan1 + " - " + pecahan2 + " = " + hasilKurang);
71     System.out.println("Hasil dari perkalian: " + pecahan1 + " * " + pecahan2 + " = " + hasilKali);
72     System.out.println("Hasil dari pembagian: " + pecahan1 + " / " + pecahan2 + " = " + hasilBagi);
73 }
74 }
```

Terminal Output:

```
PS G:\Garin\TUGAS KAMPUS\PRO CODE> & "C:\Program Files\Java\jre-1.8\bin\java.exe" -cp ".\C:\Users\hargu\AppData\Localing\Code\User\workspace\storage\3b86542aadd4494651dfe3a2c728f3\redhot_java\src_ws\PRO CODE_db\src\2\bin" -P
ecahan
Hasil dari penjumlahan: 1/2 + 3/4 = 5/4
Hasil dari pengurangan: 1/2 - 3/4 = 1/4
Hasil dari perkalian: 1/2 * 3/4 = 3/8
Hasil dari pembagian: 1/2 / 3/4 = 2/3
PS G:\Garin\TUGAS KAMPUS\PRO CODE>
```

SOAL 2

Source code untuk class Pecahan.java :

```
class Pecahan{
    private int pembilang;
    private int penyebut;

    public Pecahan(int pembilang, int penyebut) {
        if (penyebut == 0) {
            throw new IllegalArgumentException("Penyebut tidak boleh nol");
        }
        this.pembilang = pembilang;
        this.penyebut = penyebut;
    }

    public int getPembilang() {
        return pembilang;
    }

    public int getPenyebut() {
        return penyebut;
    }

    @Override
    public String toString() {
        return pembilang + "/" + penyebut;
    }
}
```

Source code untuk class Matriks.java :

```
public class Matriks {
    private Pecahan[][] data;
    private int baris;
    private int kolom;

    public Matriks(int baris, int kolom) {
        this.baris = baris;
        this.kolom = kolom;
        data = new Pecahan[baris][kolom];
    }

    public Matriks tambah(Matriks other) {
        if (baris != other.baris || kolom != other.kolom) {
```

```

        throw new IllegalArgumentException("Ukuran matriks tidak cocok");
    }
    Matriks hasil = new Matriks(baris, kolom);
    for (int i = 0; i < baris; i++) {
        for (int j = 0; j < kolom; j++) {
            int newPembilang = this.data[i][j].getPembilang() *
other.data[i][j].getPenyebut()
                + other.data[i][j].getPembilang() *
this.data[i][j].getPenyebut();
            int newPenyebut = this.data[i][j].getPenyebut() *
other.data[i][j].getPenyebut();
            hasil.data[i][j] = new Pecahan(newPembilang, newPenyebut);
        }
    }
    return hasil;
}

public Matriks kurang(Matriks other) {
    if (baris != other.baris || kolom != other.kolom) {
        throw new IllegalArgumentException("Ukuran matriks tidak cocok");
    }
    Matriks hasil = new Matriks(baris, kolom);
    for (int i = 0; i < baris; i++) {
        for (int j = 0; j < kolom; j++) {
            int newPembilang = this.data[i][j].getPembilang() *
other.data[i][j].getPenyebut()
                - other.data[i][j].getPembilang() *
this.data[i][j].getPenyebut();
            int newPenyebut = this.data[i][j].getPenyebut() *
other.data[i][j].getPenyebut();
            hasil.data[i][j] = new Pecahan(newPembilang, newPenyebut);
        }
    }
    return hasil;
}

public Matriks dot(Matriks other) {
    if (baris != other.baris || kolom != other.kolom) {
        throw new IllegalArgumentException("Ukuran matriks tidak cocok");
    }
    Matriks hasil = new Matriks(baris, kolom);
    for (int i = 0; i < baris; i++) {
        for (int j = 0; j < kolom; j++) {
            int newPembilang = this.data[i][j].getPembilang() *
other.data[i][j].getPembilang();

```

```

        int newPenyebut = this.data[i][j].getPenyebut() *
other.data[i][j].getPenyebut();
        hasil.data[i][j] = new Pecahan(newPembilang, newPenyebut);
    }
}
return hasil;
}

public Matriks transpose() {
    Matriks hasil = new Matriks(kolom, baris);
    for (int i = 0; i < baris; i++) {
        for (int j = 0; j < kolom; j++) {
            hasil.data[j][i] = this.data[i][j];
        }
    }
    return hasil;
}

public static Matriks identitas(int ukuran) {
    Matriks identitas = new Matriks(ukuran, ukuran);
    for (int i = 0; i < ukuran; i++) {
        for (int j = 0; j < ukuran; j++) {
            if (i == j) {
                identitas.data[i][j] = new Pecahan(1, 1);
            } else {
                identitas.data[i][j] = new Pecahan(0, 1);
            }
        }
    }
    return identitas;
}

// Metode untuk mencetak matriks
@Override
public String toString() {
    StringBuilder builder = new StringBuilder();
    for (int i = 0; i < baris; i++) {
        for (int j = 0; j < kolom; j++) {
            builder.append(data[i][j]).append("\t");
        }
        builder.append("\n");
    }
    return builder.toString();
}
}

```

```
public static void main(String[] args) {
    Matriks matriks1 = new Matriks(2, 2);
    Matriks matriks2 = new Matriks(2, 2);

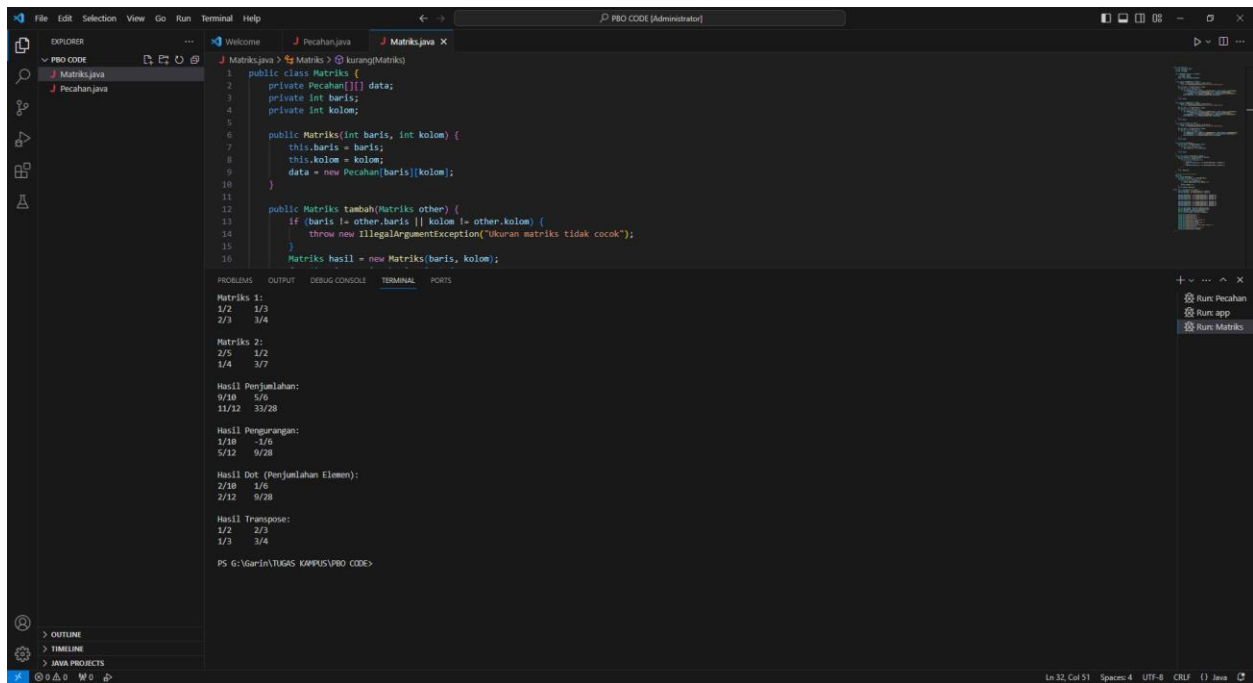
    matriks1.data[0][0] = new Pecahan(1, 2);
    matriks1.data[0][1] = new Pecahan(1, 3);
    matriks1.data[1][0] = new Pecahan(2, 3);
    matriks1.data[1][1] = new Pecahan(3, 4);

    matriks2.data[0][0] = new Pecahan(2, 5);
    matriks2.data[0][1] = new Pecahan(1, 2);
    matriks2.data[1][0] = new Pecahan(1, 4);
    matriks2.data[1][1] = new Pecahan(3, 7);

    Matriks hasilTambah = matriks1.tambah(matriks2);
    Matriks hasilKurang = matriks1.kurang(matriks2);
    Matriks hasilDot = matriks1.dot(matriks2);
    Matriks hasilTranspose = matriks1.transpose();

    System.out.println("Matriks 1:");
    System.out.println(matriks1);
    System.out.println("Matriks 2:");
    System.out.println(matriks2);
    System.out.println("Hasil Penjumlahan:");
    System.out.println(hasilTambah);
    System.out.println("Hasil Pengurangan:");
    System.out.println(hasilKurang);
    System.out.println("Hasil Dot (Penjumlahan Elemen:");
    System.out.println(hasilDot);
    System.out.println("Hasil Transpose:");
    System.out.println(hasilTranspose);
}
}
```

Screenshot hasil output :



The screenshot shows an IDE with a Java project named 'PRO CODE'. The main editor displays the 'Matrks.java' file, which contains a 'Matrks' class. The class has private attributes 'data' (a 2D array of 'Pecahan' objects), 'baris' (rows), and 'kolom' (columns). It includes a constructor 'Matrks(int baris, int kolom)' and a method 'tambah(Matrics other)' that adds two matrices. The 'tambah' method checks if the matrices have the same dimensions and then creates a new 'Matrks' object with the result.

```
1 public class Matrks {
2     private Pecahan[][] data;
3     private int baris;
4     private int kolom;
5
6     public Matrks(int baris, int kolom) {
7         this.baris = baris;
8         this.kolom = kolom;
9         data = new Pecahan[baris][kolom];
10    }
11
12    public Matrks tambah(Matrics other) {
13        if (baris != other.baris || kolom != other.kolom) {
14            throw new IllegalArgumentException("Ukuran matrks tidak cocok");
15        }
16        Matrks hasil = new Matrks(baris, kolom);
```

The terminal output shows the results of the program execution. It displays two matrices, their sum, their difference, the dot product of their elements, and the transpose of the first matrix.

```
Matrks 1:
1/2  1/3
2/3  3/4

Matrks 2:
2/5  1/2
1/4  3/7

Hasil Penjumlahan:
9/10  5/6
11/12  33/28

Hasil Pengurangan:
1/10  -1/6
5/12  9/28

Hasil Dot (Penjumlahan Elemen):
2/10  1/6
2/12  9/28

Hasil Transpose:
1/2  2/3
1/3  3/4

PS G:\Garin\TUGAS KAMPUS\PRO CODE>
```