

MODUL 4

CLASS BAGIAN II

1. Tujuan
 - a. Mahasiswa memahami pengembangan class
 - b. Mahasiswa mampu menerapkan pengoperasian class, object, dan method yang berkaitan dengan alokasi memori.
2. Latihan praktikum

Buat file .java di editor masing-masing, dan lakukan latihan pemrograman yang ditunjukkan setiap nomor.

 - a. Kode contoh penggunaan variabel dan konstanta

```
import java.text.*;

class luasLingkaran{
    public static void main(String[] args) {
        final double PI = 3.14159;
        final string TAB = "\t";
        final string NEWLINE = "\n"

        double jarijari,luas, keliling;

        DecimalFormat df = new DecimalFormat("0.000");

        jarijari = 2.35

        luas = PI*jarijari*jarijari;
        keliling = 2.0 *PI*jarijari;

        System.out.println("Untuk jari-jari dengan panjang " + TAB +
df.format(jarijari) + " cm"+ NEWLINE + "dihasilkan Luas " + TAB + df.format(luas) +
NEWLINE + "dan kelililng " + TAB + df.format(keliling) );

    }
}
```

- b. Kode ini digunakan merupakan penerapan pengembalian sebuah objek dari sebuah method

```
class Question {
    private int one;
    public void myMethod( int one ) {
        this.one = one;
        one = 12;
    }
}
```

```
class Test {
    public static void main(String[] arg) {
```

```
        int one = 30;
        Question q = new Question();
        q.myMethod(one);
        System.out.println(one);
    }
}
```

c. Pemanggilan melalui pelewatan nilai parameter

```
class Tester {
    public void myMethod(int one, double two ) {
        one = 25;
        two = 35.4;
    }
}
```

Berikan kode program berikut untuk method main

```
Tester tester;
int x, y;
tester = new Tester();
x = 10;
y = 20;
tester.myMethod( x, y );
System.out.println( x + " " + y );
```

3. Tugas Praktikum

- a. Diketahui sebuah class Fraction dengan kode di bawah ini, buatlah class main untuk menampilkan hasil numerator dan denominator berdasarkan class-class lainnya dimana class-class tersebut merupakan class penentu nilai numerator dan denominator!

```
class Fraction {
    private int numerator;
    private int denominator;
    public Fraction(int num, int denom) {
        setNumerator(num);
        setDenominator(denom);
    }
    public int getDenominator() {
        return denominator;
    }
    public int getNumerator() {
        return numerator;
    }
    public void setDenominator(int denom) {
        if (denom == 0) {
            //Fatal error
            System.err.println("Fatal Error");
            System.exit(1);
        }
        denominator = denom;
    }
    public void setNumerator(int num) {
        numerator = num;
    }
    public String toString() {
        return getNumerator() + "/" + getDenominator();
    }
}
```

```
}  
    //untuk method yang baru tambahkan di baris baru  
}
```

Class-class yang diperlukan:

a) method simplify pada Class fraction.

```
public void simplify( ) {  
    int num = getNumerator();  
    int denom = getDenominator();  
    int gcd = gcd(num, denom);  
    setNumerator(num/gcd);  
    setDenominator(denom/gcd);  
}
```

b) Tambahkan method gcd

```
public int gcd(int m, int n) {  
  
    int r = n % m;  
    while (r !=0) {  
        n = m;  
        m = r;  
        r = n % m;  
    }  
    return m;  
}
```

c) Tambahkan method add

```
public Fraction add( Fraction frac) {  
    int a, b, c, d;  
    Fraction sum;  
    a = this.getNumerator(); //get the receiving  
    b = this.getDenominator(); //object's num and denom  
    c = frac.getNumerator(); //get frac's num  
    d = frac.getDenominator(); //and denom  
    sum = new Fraction(a*d + b*c, b*d);  
    return sum;  
}
```

d) Tambahkan method divide

```
public Fraction divide(Fraction frac) {  
    int a, b, c, d;  
    Fraction quotient;  
    a = this.getNumerator();  
    b = this.getDenominator();  
    c = frac.getNumerator();  
    d = frac.getDenominator();  
    quotient = new Fraction(a*d, b*c);  
    return quotient;  
}
```

e) Tambahkan method multiply

```
public Fraction multiply(Fraction frac) {  
    int a, b, c, d;
```

```
Fraction product;  
a = this.getNumerator();  
b = this.getDenominator();  
c = frac.getNumerator();  
d = frac.getDenominator();  
product = new Fraction(a*c, b*d);  
return product;  
}
```

f) Tambahkan method subtract

```
public Fraction subtract(Fraction frac) {  
    int a, b, c, d;  
    Fraction diff;  
    a = this.getNumerator();  
    b = this.getDenominator();  
    c = frac.getNumerator();  
    d = frac.getDenominator();  
    diff = new Fraction(a*d - b*c, b*d);  
    return diff;  
}
```

- b. Modifikasi operasi method gcd dan bandingkan hasil sebelum modifikasi dengan setelah modifikasi method gcd!

```
Fraction f1 = new Fraction(24, 36);  
f1.simplify();
```

- c. Lakukan pemanggilan constructor menggunakan this!

```
public Fraction() {  
    this(0, 1);  
}  
public Fraction(int number) {  
    this(number, 1);  
}  
public Fraction(Fraction frac) {  
    this( frac.getNumerator(),  
    frac.getDenominator() );  
}  
public Fraction(int num, int denom) {  
    setNumerator(num);  
    setDenominator(denom);  
}
```

- d. Nilai f1 pada method simplify masih berubah dikarenakan masih terdapat dua alokasi memori untuk class tersebut, lakukan modifikasi pada method simplify dengan cara eliminasi alokasi memori yang tidak diperlukan pada akhir eksekusi object simplify melalui pengarah nilai dari salah satu lokasi memori ke lokasi memori yang dituju untuk penyimpanan hasil!