



MODUL PRAKTIKUM

Membuat Class Sendiri

Versi 1.4

Modul Praktikum Membuat Class Sendiri

1. Tujuan

- Membuat kelas mereka sendiri
- Mendeklarasikan atribut dan *method* pada *class*
- Menggunakan referensi *this* untuk mengakses *instance data*
- Membuat dan memanggil *overloaded method*
- Mengimport dan membuat *package*
- Menggunakan *access modifiers* untuk mengendalikan akses terhadap *class member*

2. Latar Belakang

Setelah kita mempelajari penggunaan *class* dari Java Class Library, kita akan mempelajari bagaimana menuliskan sebuah *class* sendiri. Pada bagian ini, untuk mempermudah pemahaman pembuatan *class*, kita akan membuat contoh *class* dimana akan ditambahkan beberapa data dan fungsi – fungsi lain.

Kita akan membuat *class* yang mengandung informasi dari Siswa dan operasi – operasi yang dibutuhkan pada *record* siswa.

Beberapa hal yang perlu diperhatikan pada *syntax* yang digunakan pada bab ini dan bagian lainnya :

- | | |
|---------------|--|
| * | - Menandakan bahwa terjadi lebih dari satu kejadian dimana elemen tersebut diimplementasikan |
| <description> | - Menandakan bahwa Anda harus memberikan nilai pasti pada bagian ini |
| [] | - Indikasi bagian optional |

Modul Praktikum Membuat Class Sendiri

3. Percobaan

Percobaan 1 Membuat Class Student Record:

Modul Praktikum Membuat Class Sendiri

```
public class StudentRecord {  
    private String name;  
    private String address;  
    private int age;  
    private double mathGrade;  
    private double englishGrade;  
    private double scienceGrade;  
    private double average;  
    private static int studentCount;  
    /**  
     * Menghasilkan nama dari Siswa  
     */  
    public String getName(){  
        return name;  
    }  
    /**  
     * Mengubah nama siswa  
     */  
    public void setName( String temp ){  
        name = temp;  
    }  
    // area penulisan kode lain  
    /**  
     * Menghitung rata - rata nilai Matematik, Bahasa Inggris, ** Ilmu Pasti  
     */  
    public double getAverage(){  
        double result = 0;  
        result = ( mathGrade+englishGrade+scienceGrade )/3;  
        return result;  
    }  
    /**  
     * Menghasilkan jumlah instance StudentRecord  
     */  
    public static int getStudentCount(){  
        return studentCount;  
    }  
}
```

Modul Praktikum Membuat Class Sendiri

```
public class StudentRecordExample {  
    public static void main( String[] args ){  
        //membuat 3 object StudentRecord  
        StudentRecord annaRecord = new StudentRecord();  
        StudentRecord beahRecord = new StudentRecord();  
        StudentRecord crisRecord = new StudentRecord();  
        //Memberi nama siswa  
        annaRecord.setName( "Anna" );  
        beahRecord.setName( "Beah" );  
        crisRecord.setName( "Cris" );  
        //Menampilkan nama siswa "Anna"  
        System.out.println( annaRecord.getName() );  
        //Menampilkan jumlah siswa  
        System.out.println( "Count="+StudentRecord.getStudentCount() );  
    }  
}
```

Percobaan 2 Membuat Class Student Record Example :

Modul Praktikum Membuat Class Sendiri

```
public class StudentRecord {
    private String name;
    private String address;
    private int age;
    private double mathGrade;
    private double englishGrade;
    private double scienceGrade;
    private double average;
    private static int studentCount;
    public StudentRecord(){ //constructor default
        studentCount++;
    }
    public StudentRecord(String temp){
        this.name = temp;
        studentCount++;
    }
    public StudentRecord(String name, String Address){
        this.name = name;
        this.address = Address;
        studentCount++;
    }
    public StudentRecord(double mGrade,double eGrade,double sGrade){
        mathGrade = mGrade;
        englishGrade = eGrade;
        scienceGrade = sGrade;
        studentCount++;
    }
    /**
     * Menghasilkan nama dari siswa
     */
    public String getName(){
        return name;
    }
    /**
     * mengubah nama siswa
     */
}
```

Percobaan 3 Membuat Class Student Record yang telah dimodifikasi:

Modul Praktikum Membuat Class Sendiri

```
public void setName(String temp){
    name = temp;
}
public String getAddress(){
    return address;
}
public void setAddress(String temp){
    address = temp;
}
public int getAge(){
    return age;
}
public void setAge(int temp){
    age = temp;
}
//area penulisan kode lain
/**
 * Menghitung rata-rata nilai matematika, bahasa inggris, ilmu pengetahuan
 */
public double getAverage(){
    double result = 0;
    result = (mathGrade+englishGrade+scienceGrade)/3;
    return result;
}
public static int getStudentRecord(){
    return studentCount;
}
public void print(String temp){
    System.out.println("Name : "+name );
    System.out.println("Address : "+address);
    System.out.println("Age : "+age);
}
public void print(double eGrade, double mGrade, double sGrade){
    System.out.println("Name : "+name);
    System.out.println("Math Grade : "+mGrade);
    System.out.println("English Grade : "+eGrade);
    System.out.println("Science Grade : "+sGrade);
}
```

Modul Praktikum Membuat Class Sendiri

```
public double getEnglishGrade() {  
    return englishGrade;  
}  
public void setEnglishGrade(double englishGrade) {  
    this.englishGrade = englishGrade;  
}  
public double getMathGrade() {  
    return mathGrade;  
}  
public void setMathGrade(double mathGrade) {  
    this.mathGrade = mathGrade;  
}  
public double getScienceGrade() {  
    return scienceGrade;  
}  
public void setScienceGrade(double scienceGrade) {  
    this.scienceGrade = scienceGrade;  
}  
}
```


Modul Praktikum Membuat Class Sendiri

Percobaan 4 Membuat Class Student Record Example yang telah

```
public class StudentRecordExample {
    public static void main(String[] args) {
        //membuat 3 object StudentRecord
        StudentRecord annaRecord = new StudentRecord();
        StudentRecord beahRecord = new StudentRecord();
        StudentRecord crisRecord = new StudentRecord();
        //versi baru
        StudentRecord recordAnna = new StudentRecord("Candra");
        StudentRecord recordBeah = new StudentRecord("Bagus", "Malang");
        StudentRecord recordCris = new StudentRecord(80, 90, 100);
        //memberi nama siswa
        annaRecord.setName("Anna");
        beahRecord.setName("Beah");
        crisRecord.setName("Cris");
        //menampilkan nama siswa "Anna"
        System.out.println(annaRecord.getName());
        //menampilkan jumlah siswa
        System.out.println("Count="+StudentRecord.getStudentRecord());
        StudentRecord anna2Record = new StudentRecord();
        anna2Record.setName("Anna");
        anna2Record.setAddress("Philipina");
        anna2Record.setAge(15);
        anna2Record.setMathGrade(80);
        anna2Record.setEnglishGrade(95.5);
        anna2Record.setScienceGrade(100);
        //overload method
        anna2Record.print(anna2Record.getName());
        anna2Record.print(anna2Record.getEnglishGrade(),
            anna2Record.getMathGrade(),
            anna2Record.getScienceGrade());
        recordAnna.print(recordAnna.getName());
    }
}
```

dimodifikasi :

Modul Praktikum Membuat Class Sendiri

```
package schoolClasses;

public class StudentRecord {
    private String name;
    private String address;
    private int age;
}
```

Percobaan 5 Package :

```
public class PublicElevator {
    public boolean bukaPintu = false;
    public int lantaiSkrng = 1;
    public int berat = 0;
    public final int KAPASITAS = 1000;
    public final int LANTAI_ATAS = 5;
    public final int LANTAI_BAWAH = 1;
}
```

Percobaan 6 Modifier Public :

Modul Praktikum Membuat Class Sendiri

```
public class PublicElevatorTest {
    public static void main(String args[]){
        PublicElevator pubElevator = new PublicElevator();
        pubElevator.bukaPintu = true; //penumpang masuk
        pubElevator.bukaPintu = false; //pintu ditutup
        //pergi ke lantai 0 dibawah gedung
        pubElevator.lantaiSkrng--;
        pubElevator.lantaiSkrng++;
        //lompat ke lantai 7 (hanya ada 5 lantai dalam gedung)
        pubElevator.lantaiSkrng = 7;
        pubElevator.bukaPintu = true; //penumpang masuk/keluar
        pubElevator.bukaPintu = false;
        pubElevator.lantaiSkrng = 1; //menuju lantai pertama
        pubElevator.bukaPintu = true; //penumpang masuk/keluar
        pubElevator.lantaiSkrng++; //elevator bergerak tanpa menutup pintu
        pubElevator.bukaPintu = false;
        pubElevator.lantaiSkrng--;
        pubElevator.lantaiSkrng--;
    }
}
```

Percobaan 7 Modifier Private:

```
public class PrivateElevator1 {
    private boolean bukaPintu = false;
    private int lantaiSkrng = 1;
    private int berat = 0;
    private final int KAPASITAS = 1000;
    private final int LANTAI_ATAS = 5;
    private final int LANTAI_BAWAH = 1;
}
```

Modul Praktikum Membuat Class Sendiri

```
public class PrivateElevator1Test {  
    public static void main(String args[]){  
        PrivateElevator1 privElevator = new PrivateElevator1();  
        /*****  
        * Baris kode dibawah ini tidak akan di-compile *  
        * karena berusaha untuk mengakses variabel    *  
        * private *  
        *****/  
        privElevator.bukaPintu = true; //penumpang masuk  
        privElevator.bukaPintu = false; //pintu ditutup  
        //pergi ke lantai 0 dibawah gedung  
        privElevator.lantaiSkrng--;  
        privElevator.lantaiSkrng++;  
        //lompat ke lantai 7 (hanya ada 5 lantai dalam gedung)  
        privElevator.lantaiSkrng = 7;  
        privElevator.bukaPintu = true; //penumpang masuk/keluar  
        privElevator.bukaPintu = false;  
        privElevator.lantaiSkrng = 1; //menuju lantai pertama  
        privElevator.bukaPintu = true; //penumpang masuk/keluar  
        privElevator.lantaiSkrng++; //elevator bergerak tanpa menutup pintu  
        privElevator.bukaPintu = false;  
        privElevator.lantaiSkrng--;  
        privElevator.lantaiSkrng--;  
    }  
}
```

Modul Praktikum Membuat Class Sendiri

```
public class PrivateShirt1{
    private int idBaju = 0; //ID default untuk baju
    private String keterangan = "-Keterangan Diperlukan-"; //default
    //Kode warna R=Merah, G=Hijau, B=Biru, U=Tidak Ditentukan
    private char kodeWarna = 'U';
    private double harga = 0.0; //Harga default untuk semua barang
    private int jmlStok = 0; //Default untuk jumlah barang
    public char getKodeWarna() {
        return kodeWarna;
    }
    public void setKodeWarna(char kode) {
        kodeWarna = kode;
    }
    // Method get dan set untuk idBaju, keterangan
    // harga, dan jmlStok akan mengikuti
} //akhir kelas
```

Percobaan 8 Method Accessor dan Method Mutator:

```
public class PrivateShirt1Test {
    public static void main(String[] args) {
        PrivateShirt1 privShirt = new PrivateShirt1();
        char kodeWarna;
        // Mengisi kode warna yang valid
        privShirt.setKodeWarna('R');
        kodeWarna = privShirt.getKodeWarna();
        // Kelas PrivateShirt1Test bisa mengisi sebuah kodeWarna yang valid
        System.out.println("Kode Warna: " + kodeWarna);
        // Mengisi kode warna yang salah
        privShirt.setKodeWarna('Z');
        kodeWarna = privShirt.getKodeWarna();
        // Kelas PrivateShirt1Test bisa mengisi sebuah kodeWarna yang salah
        System.out.println("Kode Warna: " + kodeWarna);
    }
}
```

Modul Praktikum Membuat Class Sendiri

```
public class PrivateShirt2 {
    private int idBaju = 0; //ID default untuk baju
    private String keterangan = "-Keterangan Diperlukan-"; //default
    //Kode warna R=Merah, G=Hijau, B=Biru, U=Tidak Ditentukan
    private char kodeWarna = 'U';
    private double harga = 0.0; //Harga default untuk semua barang
    private int jmlStok = 0; //Default untuk jumlah barang

    public char getKodeWarna() {
        return kodeWarna;
    }
    public void setKodeWarna(char kode) {
        switch (kode) {
            case 'R':
            case 'G':
            case 'B':
                kodeWarna = kode;
                break;
            default:
                System.out.println("kodeWarna salah, gunakan R, G, atau B");
        }
    }
    // Method get dan set untuk idBaju, keterangan
    // harga, dan jmlStok akan mengikuti
} //akhir kelas
```

Percobaan 9 Method Accessor dan Method Mutator:

```
public class PrivateShirt2Test {
    public static void main(String[] args) {
        PrivateShirt2 privShirt = new PrivateShirt2();
        char kodeWarna;
        // Mengisi kode warna yang valid
        privShirt.setKodeWarna('R');
        kodeWarna = privShirt.getKodeWarna();
        // Kelas PrivateShirt2Test bisa mengisi sebuah kodeWarna yang valid
        System.out.println("Kode Warna: " + kodeWarna);
        // Mengisi kode warna yang salah
        privShirt.setKodeWarna('Z');
        kodeWarna = privShirt.getKodeWarna();
        // Kelas PrivateShirt2Test bisa mengisi sebuah kodeWarna yang salah
        // Kode warna masih R
        System.out.println("Kode Warna: " + kodeWarna);
    }
}
```

Modul Praktikum Membuat Class Sendiri

```
public class ConstructorShirt1 {  
    private int idBaju = 0; //ID default untuk baju  
    private String keterangan = "-Keterangan Diperlukan-"; //default  
    //Kode warna R=Merah, G=Hijau, B=Biru, U=Tidak Ditentukan  
    private char kodeWarna = 'U';  
    private double harga = 0.0; //Harga default untuk semua barang  
    private int jmlStok = 0; //Default untuk jumlah barang  
  
    public ConstructorShirt1(char kodeAwal){  
        switch (kodeAwal) {  
            case 'R':  
            case 'G':  
            case 'B':  
                kodeWarna = kodeAwal;  
                break;  
            default:  
                System.out.println("kodeWarna salah, gunakan R, G, atau B");  
        }  
    }  
    public char getKodeWarna() {  
        return kodeWarna;  
    }  
} //akhir kelas
```

Percobaan 10 Constructor:

```
public class ConstructorShirt1Test {  
    public static void main(String[] args) {  
        ConstructorShirt1 constShirt = new ConstructorShirt1('R');  
        char kodeWarna;  
        kodeWarna = constShirt.getKodeWarna();  
        System.out.println("Kode Warna: " + kodeWarna);  
    }  
}
```

Modul Praktikum Membuat Class Sendiri

```
public class DefaultShirt {  
    private int idBaju = 0; //ID default untuk baju  
    private String keterangan = "-Keterangan Diperlukan-"; //default  
    //Kode warna R=Merah, G=Hijau, B=Biru, U=Tidak Ditentukan  
    private char kodeWarna = 'U';  
    private double harga = 0.0; //Harga default untuk semua barang  
    private int jmlStok = 0; //Default untuk jumlah barang  
  
    public DefaultShirt(){  
        kodeWarna = 'R';  
    }  
  
    public char getKodeWarna() {  
        return kodeWarna;  
    }  
} //akhir kelas
```

Percobaan 11 Constructor Default:

```
public class DefaultShirtTest {  
    public static void main(String[] args) {  
        DefaultShirt defShirt = new DefaultShirt();  
        char kodeWarna;  
        kodeWarna = defShirt.getKodeWarna();  
        System.out.println("Kode Warna: " + kodeWarna);  
    }  
}
```


Modul Praktikum Membuat Class Sendiri

```

public class ConstructorShirt2 {
    private int idBaju = 0; //ID default untuk baju
    private String keterangan = "-Keterangan Diperlukan-"; //default
    private char kodeWarna = 'U' ;//Kode warna R=Merah, G=Hijau, B=Biru, U=Tidak Ditentukan
    private double harga = 0.0; //Harga default untuk semua barang
    private int jmlStok = 0; //Default untuk jumlah barang
    public ConstructorShirt2() {
        kodeWarna = 'R';
    }
    public ConstructorShirt2(char kodeAwal) {
        switch (kodeAwal) {
            case 'R':
            case 'G':
            case 'B': kodeWarna = kodeAwal; break;
            default : System.out.println("kodeWarna salah, gunakan R, G, atau B");
        }
    }
    public ConstructorShirt2(char kodeAwal, int jmlAwal) {
        switch (kodeAwal) {
            case 'R':
            case 'G':
            case 'B': kodeWarna = kodeAwal; break;
            default : System.out.println("kodeWarna salah, gunakan R, G, atau B");
        }
        if (jmlAwal > 0 && jmlAwal < 2000) {
            jmlStok = jmlAwal;
        } else {
            System.out.println("Jumlah salah. harus diantara 0 - 2000");
        }
    }
    public char getKodeWarna() {
        return kodeWarna;
    }
    public int getJmlStok() {
        return jmlStok;
    } //akhir kelas
}

```

Percobaan 12 Constructor Overloading:

Modul Praktikum Membuat Class Sendiri

```
public class ConstructorShirt2Test {  
    public static void main(String[] args) {  
        ConstructorShirt2 constShirtSatu = new ConstructorShirt2('R');  
        ConstructorShirt2 constShirtDua = new ConstructorShirt2('G');  
        ConstructorShirt2 constShirtTiga = new ConstructorShirt2('B', 1000);  
        char kodeWarna;  
        int jumlah;  
        kodeWarna = constShirtSatu.getKodeWarna();  
        System.out.println("Kode Warna Object 1: " + kodeWarna);  
        kodeWarna = constShirtDua.getKodeWarna();  
        System.out.println("Kode Warna Object 2: " + kodeWarna);  
        kodeWarna = constShirtTiga.getKodeWarna();  
        jumlah = constShirtTiga.getJmlStok();  
        System.out.println("Kode Warna Object 3: " + kodeWarna);  
        System.out.println("Jumlah Stok Object 3: " + jumlah);  
    }  
}
```

Percobaan 13 Method Overloading:

```
public class Calculator {  
    public int sum(int numberOne, int numberTwo){  
        System.out.println("Method One");  
        return numberOne + numberTwo;  
    }  
  
    public float sum(float numberOne, float numberTwo) {  
        System.out.println("Method Two");  
        return numberOne + numberTwo;  
    }  
  
    public float sum(int numberOne, float numberTwo) {  
        System.out.println("Method Three");  
        return numberOne + numberTwo;  
    }  
}
```

Modul Praktikum Membuat Class Sendiri

```
public class CalculatorTest {  
    public static void main(String[] args) {  
        Calculator myCalculator = new Calculator();  
        int totalOne = myCalculator.sum(2,3);  
        System.out.println(totalOne);  
        float totalTwo = myCalculator.sum(15.99F, 12.85F);  
        System.out.println(totalTwo);  
        float totalThree = myCalculator.sum(2, 12.85F);  
        System.out.println(totalThree);  
    }  
}
```

Modul Praktikum Membuat Class Sendiri

Percobaan 14 Method Overloading:

```
public class ShirtTwo {
    public int shirtID = 0; //Default ID for the shirt
    public String description = "-description required-"; //default
    public char colorCode = 'U'; //The color codes are R=Red, B=Blue, U=Unset
    public double price = 0.0; //Default price for all shirts
    public int quantityInStock = 0; //Default quantity for all shirts

    public void setShirtInfo(int ID, String desc, double cost) {
        shirtID = ID;
        description = desc;
        price = cost;
    }
    public void setShirtInfo(int ID, String desc, double cost, char color) {
        shirtID = ID;
        description = desc;
        price = cost;
        colorCode = color;
    }
    public void setShirtInfo(int ID, String desc, double cost, char color, int quantity) {
        shirtID = ID;
        description = desc;
        price = cost;
        colorCode = color;
        quantityInStock = quantity;
    }
    //This method displays the values for an item
    public void display() {
        System.out.println("Item ID: " + shirtID);
        System.out.println("Item description: " + description);
        System.out.println("Color Code: " + colorCode);
        System.out.println("Item price: " + price);
        System.out.println("Quantity in stock: " + quantityInStock);
    } //end of display method
} //end of class
```

4. Latihan

4.1 Entry Buku Alamat

Tugas Anda adalah membuat sebuah *class* yang memuat data-data pada buku alamat. Tabel berikut mendefinisikan informasi yang dimiliki oleh buku alamat.

Modul Praktikum Membuat Class Sendiri

Attribut	Deskripsi
Nama	Nama Lengkap perseorangan
Alamat	Alamat Lengkap
Nomor Telepon	Nomor telepon personal
Alamat E-Mail	Alamat E-Mail personal

Tabel 1: Atribut dan Deskripsi Atribut

Buat implementasi dari *method* sebagai berikut :

- Menyediakan *accessor* dan *mutator method* terhadap seluruh atribut
- Constructor

4.1 Buku Alamat

Buat sebuah *class* buku alamat yang dapat menampung 100 data. Gunakan *class* yang telah dibuat pada nomor pertama. Anda harus mengimplementasikan *method* berikut pada buku alamat :

1. Memasukkan data
2. Menghapus data
3. Menampilkan seluruh data
4. Update data