

MIDTERM

OBJECT ORIENTED PROGRAM



ANANDA AZ HARUDDIN SALIMA

2241720071

2 I

Question 1: Class writing.

Based on the example ClassA class below, explain whether the source code in the example class is correct. If not, what needs to be fixed?

```
public class ClassA { float
    f1 = 0.15f;
    float hitung() { float x
        = 2f * f1;
    }
}
```

Jawab no 1:

```
public class
    ClassA { float
        f1 = 0.15f;

        float
            hitung(
                ) { float
                    x = 2f *
                    f1;
                    return x;
            }
    }
```

Question 2: Calculation of the number of elements of a 2-dimensional array.

In the SoalArray1 class, there is a 2-dimensional array with a size of 3x3. Write Java code to calculate the total number of elements of that array by using loops.

```
public class SoalArray1 { public static void main(String[]
    args) { int[][] arrayInt = {{1, 1, 4}, {2, 1, 2}, {3, 2,
    1}};
        // Count the number of elements of a 2-dimensional array
        use recurrence
    }
}
```

Answer no 2: public class SoalArray1 { public static void
main(String[] args) { int[][] arrayInt = {{1, 1, 4}, {2, 1, 2},
{3, 2, 1}}; int total = 0; // Variables to store the total
number of elements

```
    for (int i = 0; i < arrayInt.length;
        i++) { for (int j = 0; j <
            arrayInt[i].length; j++) {
                total += arrayInt[i][j]; // Add element values to
                totals
            }
    }
```

```
}
```

```
System.out.println("Total number of array elements:  
" + total);
```

```
}
```

```
}
```

It uses two nested loops to access all elements in a 2-dimensional array and sum them into a total variable.

Question 3: Inheritance of attribute and methods.

In the given source code, the class `ClassY` is an instance of the class `Class`.
specifies what attributes and methods are inherited by `ClassY` from its parent class (class `Class`).

Also explain what the output of the code written in the `ClassY` class is and how the value is obtained.

```
public class Class {  
    int a = 2; int x  
    = 0;  
    int hitung() { x = x +  
        5 * a; return x;  
    }  
}  
public class ClassY extends Class  
{ int b = 5;
```

Answer no 3:

In the given source code, the `ClassY` class is an instance of the `Class` class. Therefore, `ClassY` inherits attributes and methods from its parent class, `Class`. Here are the attributes and methods inherited by `ClassY` from `Class`:

Inherited attributes:

`int a` - This attribute is inherited from the `Class`. The initial value is 2. `int x` - This attribute is also inherited from the `Class`. The initial value is 0.

Method yang diwarisi:

`int count()` - This method is inherited from the `Class`. This method calculates the value of `x` with the formula $x = x + 5 * a$, and returns the value of `x` after calculation.

Now let's look at the output of the code written on the `ClassY` class and how that value is obtained:

```
public static void main(String[] args) {  
    ClassY cy = new ClassY();  
    System.out.println(cy.hitungY());  
}
```

In the main method, the cy object is created as an instance of the ClassY class. Then, the countY() method is called on the cy object. In the countY() method, there is a command `y = count() * b;`.

count() is called from a Class class because that method is inherited by ClassY. It will calculate the value of x based on the given formula ($x = x + 5 * a$). After that, the value of x will be $x = 0 + 5 * 2 = 10$.

Then, the calculated x value (i.e. 10) is multiplied by the b value of the ClassY class, which is 5. Thus, $y = 10 * 5 = 50$.

The calculated y value is then returned by the Y() calculation method.

The final result is printed to the screen with the `System.out.println(cy.hitungY())` command, so the output output is 50.

So, the output of that code is 50, and this value is obtained through calculations performed in the countY() method, which uses the count() method inherited from the Class class.

```
int y = 0;

int hitungY() { y =
    hitung() * b;
    return y;
}
public static void main(String[] args)
{
    ClassY cy = new ClassY();
    System.out.println(cy.hitungY());
}
}
```

Question 4: Class Student with Constructor.

In the `Student` class, complete the code with:

Added constructor to populate attribute

Create a student object and populate the attributes through the constructor.

```
Nim, name, address and jenisKelamin
. Nim, nam , address and jenisKelamin
e
```

```

public class Mahasiswa {
    String nim, nama, alamat;
    char jenisKelamin;

    // a. Tambahkan constructor
    // Gunakan constructor untuk
    // mengisi atribut nim, nama, alamat, jenisKelamin
    public static void main(String[] args) {
        // b. Buat objek mahasiswa
        // Isi atribut nim, nama, alamat, jenisKelamin
        // lewat constructor
    }
}

```

Answer no 4:

```

public class
Student {
    String nim,
    name,
    address; char
    gender;

```

```

// Constructor to fill in the attributes nim, name, address, and gender
public Student(String nim, String name, String address, char
gender) { this.nim = nim; this.name = name; this.address = address;
    this.ennee = NSAS;
}

```

```

public static void main(String[] args) {
    // Create a student object and fill in the nim, name, address, gender attributes via the
    constructor
    Student student1 = new Student ("12345", "Nath", "Jalan SEDAP", 'L');

    // Example of using attributes on student objects1
    System.out.println("NIM: " + student1.nim);
    System.out.println("Name: " + student1.name);
    System.out.println("Adress: " + student1.adress);
    System.out.println("gender: " + student1.gender);
}
}

```

Question 5: OOP Book -> Writer's.

Take a look at the following salted class and create Source code in java language based on that diagram class

Answer no 5:

```
public class Buku { private String ISBN; private String judul; private
Penulis writers;
private int price;

public String getISBN() {
    return ISBN;
}

public void setISBN(String ISBN) {
    this.ISBN = ISBN;
}

public String getTitle() {
    return judul;
}

public void setTitle(String title) {
    this.title = title;
}

public int getPrice() {
    return price;
}

public void setPrice(int price) {
    this.price = price;
}

public Writers getWriters() {
    return Writers;
}

public void set Writers WritersWriters) {
    this. writers = Writers;
}
}

public class
Writers {
private
String name;
private String address;

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getAddress() {
    return address;
}

public void setAddrees(String address) {
    this.address = address;
}
}
```

Class Diagram - Buku and Penulis

