

Nama : Naufal Syuja Luthfi

NPM : 0620101018

UAS PBO

1. Sebutkan dan jelaskan menurut bahasa Anda sendiri istilah – istilah di bawah ini :
(Knowledge Test : 30)

- a. Access Modifier (public, private, protected)
- b. Inheritance
- c. Polymorphsm
- d. Interface
- e. Exception Handling (try catch)

2. Buatlah sebuah program dengan algoritma dibawah ini (Logic Test : 25)

Input Array 1 : [1,3,5,7,8,9] Array 2 : [1,2,4,5,8] Output : 1,5,8

3. Perhatikan potongan code dibawah ini, lengkapi titik titik yang ada sebutkan apa saja konsep OOP yang digunakan diikuti nomor baris. (OOP Test : 45)

<pre>public class DataClass { int numberData; String wordsData; public DataClass(int numberData, String ...){ this.numberData = ; = wordsData; } Public int getNumberData(){ } public String getWordsData(){ } } public interface ParentInterface { public DataClass initData(int number, String words); } public class ParentClass implements { public DataClass initData(int number, String words){ return new DataClass(number, words); } public String getDataString() { return “” ; }</pre>

```

}
public class childClass extends ... {
    DataClass childData ;
    public void setData() {
        childData = initData(1, "firstChild");
    }
    @Override
    public String getDataString() { return childData.getWordsData(); }
}
public class secondChild extends ... {
    DataClass secondChildData ;
    public void setData() {
        secondChildData = initData(2, "secondChild");
    }
    @Override
    public String getDataString() { return ..... .getWordsData(); }
}
public MainClass {
    public static void main (String [] args){
        // do polymorphism here
        .... Objek ;
        objek = new childClass();
        objek.setData();
        System.out.println(objek.getDataString());
        objek = ..... ; // change to secondChild
        objek.setData();
        System.out.println(objek.getDataString());
    }
}
}

```

JAWABAN

1. Access Modifier adalah salah satu fitur penting dalam Object Oriented Programming (OOP) untuk melakukan Data Hiding.
 - a. Public adalah label yang berfungsi untuk menentukan sifat akses ke semua member yang dapat di akses dari dalam class itu sendiri, dari anak class dan juga dari luar class.
 - b. Private adalah label yang berfungsi untuk menentukan sifat akses ke semua member yang mengikutinya menjadi memiliki sifat yang tidak dapat di akses dari manapun kecuali melalui friend function dan dari dalam class.
 - c. Protected adalah label yang berfungsi untuk menentukan sifat akses semua member yang mengikutinya, sehingga memiliki sifat yang tidak dapat diakses dari luar class tapi masih dapat di akses dari dalam class maupun anak class (derived class).

2. Program dengan algoritma

```
import java.util.Scanner;

public class himp{
Scanner input = new Scanner(System.in);

int[] Array1= new int[100];
int[] Array2= new int[100];
int JumlahArray1;
int JumlahArray2;

public void dataAnggota() {
    System.out.print("Masukkan Jumlah Array 1 : ");
    JumlahArray1 = input.nextInt();
    System.out.println("Anggota Array 1");
    for(int i=1;i<=JumlahArray1;i++){
        System.out.print("Anggota "+i+" : ");
        Array1[i]=input.nextInt();
    }
    System.out.print("Masukkan Jumlah Array 2 : ");
    JumlahArray2 = input.nextInt();
    System.out.println("Anggota Array 2");
    for(int i=1;i<=JumlahArray2;i++){
        System.out.print("Anggota "+i+" : ");
        Array2[i]=input.nextInt();
    }
}

public void tampilAnggota(){
    System.out.print("Anggota Array 1 = { ");
    for(int i=1;i<=JumlahArray1;i++){
        System.out.print(Array1[i]+" ");
    }
    System.out.println("}");
}
```

```

        System.out.print("Anggota Array 2 = { ");
        for(int i=1;i<=JumlahArray2;i++){
            System.out.print(Array2[i]+ " ");
        }
        System.out.println("{}");
    }

    public void himpunanIrisan(){
        System.out.print("Himpunan Irisan Array 1 dan 2 : { ");
        for( int i=1;i<=JumlahArray2;i++){
            boolean adaSama = false;
            for(int j=1;j<=JumlahArray1;j++){
                if(Array2[i]==Array1[j])
                    adaSama=true;
            }
            if(adaSama==true){
                System.out.print(Array2[i]+" ");
            }
        }
        System.out.println(" }");
    }

    public static void main(String args[]){
        himp bilangan = new himp();
        bilangan.dataAnggota();
        bilangan.tampilAnggota();
        bilangan.himpunanIrisan();
    }
}

```

3. Pengisian potongan code

```
public class DataClass {

    int numberData; String wordsData;

    public DataClass(int numberData, String wordsData ){

        this.numberData = numberData;

        this.wordsData = wordsData;

    }

    Public int getNumberData() { int | numberData; }

    public String getWordsData() { String | wordsData; }

}

public interface ParentInterface {

    public DataClass initData(int number, String words);

    return new DataClass(number, words);

}

public class ParentClass implements ParentInterface{

    public DataClass initData(int number, String words){

        return new DataClass(number, words);

    }

    public String getDataString() { return “” ; }

}

public class childClass extends ParentClass{

    DataClass childData ;

    public void setData() {

        childData = initData(1, “firstChild”);

    }

}
```

```

    }

    @override

    public String getDataString() { return childData.getWordsData(); }

}

public class secondChild extends ParentClass{

    DataClass secondChildData ;

    public void setData() {

        secondChildData = initData(2, "secondChild");
    }

    @override

    public String getDataString() { return secondChildData.getWordsData(); }

}

public MainClass{

    public static void main (String [] args){

        // do polymorphsm here

        childClass.objek ;

        objek = new childClass();

        objek.setData();

        System.out.println(objek.getDataString());

        objek = new secondChild(); // change to secondChild

        objek.setData();

        System.out.println(objek.getDataString());

    }

}

```