

Subject :**- Abstract Class & Interface****Objective :**

- o Memahami Abstract Class & Interface

Tugas praktikum :**Praktikum Soal 1**

1. Buatlah file Hewan.java.
2. Edit file Hewan.java dengan sintaks seperti ini :

```
abstract class Hewan {
    protected String nama;
    protected int jumKaki;
    protected boolean bisaTerbang = false;

    public Hewan(String nama, int kaki, boolean terbang) {
        this.nama = nama;
        jumKaki = kaki;
        bisaTerbang = terbang;
    }
    public abstract void bersuara();

    public static void makan() {
        System.out.println("nyam, nyam, nyam");
    }

    public void isHewan() {
        System.out.println("nama : " + nama);
        System.out.println("jumlah kaki : " + jumKaki);
        System.out.println("bisa terbang : " + bisaTerbang);
    }
}
```

3. Buatlah file Manusia.java.
4. Edit file Manusia.java dengan sintaks seperti ini :

```
interface Manusia {
    public void menyanyi();
    public void ketawa();
}
```

5. Buatlah file Perkutut.java.
6. Edit file Perkutut.java dengan sintaks seperti ini :

```
class Perkutut extends Hewan {
    public Perkutut()
```

```

        super("perkutut",2,true);
    }

    public void bersuara() {
        System.out.println("\ncuit, cuit, cuit");
    }

    public static void main(String[] args) {
        Perkutut p = new Perkutut();
        p.isHewan();
        p.bersuara();
    }
}

```

7. Buatlah file Sapi.java.

8. Edit file Sapi.java dengan sintaks seperti ini :

```

class Sapi extends Hewan {
    public Sapi() {
        super("sapi", 4, false);
    }

    public void bersuara() {
        System.out.println("\nemoh...,emoh...");
    }

    public static void main(String[] args) {
        Sapi s = new Sapi();
        s.isHewan();
        s.bersuara();
    }
}

```

9. Buatlah file SpongeBob.java.

10. Edit file SPongEBob.java dengan sintaks seperti ini :

```

class SpongeBob extends Hewan implements Manusia {
    public SpongeBob() {
        super("sponge bob", 2, false);
    }

    public void bersuara() {
        System.out.println("\nhallo patrickt....");
    }

    public void menyanyi() {
        System.out.println("nye, nye, nye, wik, wik, wik");
    }

    public void ketawa() {
        System.out.println("kek, kek, kek");
    }

    public static void makan() {

```

```

        System.out.println("uenak tenan...");
    }
    public void makan2() {
        super.makan();
    }
    public static void main(String[] args) {
        SpongeBob s = new SpongeBob();
        s.isHewan();
        s.bersuara();
        s.menyanyi();
        s.ketawa();
        s.makan2();
        Hewan.makan();
        makan();
    }
}

```

11. Compile file SpongeBob.java menggunakan java compiler (javac) dengan -cp :

```
$ javac -cp ./;. SpongeBob.java
```

12. Setelah kompilasi berhasil, running file SpongeBob.java menggunakan perintah seperti berikut :

```
$ java -cp ./;. SpongeBob
```

13. Setelah running berhasil, hasil output akan tampil seperti dibawah ini :

```

nama : sponge bob
jumlah kaki : 2
bisa terbang : false

hallo patrick...
nye, nye, nye, wik, wik, wik
kek, kek, kek
nyam, nyam, nyam
nyam, nyam, nyam
uenak tenan...

```

1. Buatlah file A.java.
2. Edit file A.java dengan sintaks seperti ini :

```
abstract class A {  
    abstract public void method2();  
    public void method1() {  
        System.out.println("method konkrit dari class A");  
    }  
}
```

3. Buatlah file B.java.
4. Edit file B.java dengan sintaks seperti ini :

```
class B extends A{  
    public void method2() {  
        System.out.println("method abstrak yang sudah  
        menjadi konkrit dalam class B");  
    }  
}
```

5. Buatlah file Latjava.java.
6. Edit file Latjava.java dengan sintaks seperti ini :

```
class Latjava {  
    public static void main(String[] args) {  
        B ob= new B() ;  
        ob.method1();  
        ob.method2();  
    }  
}
```

7. Compile file Latjava.java menggunakan java compiler (javac) dengan -cp :

```
$ javac -cp ./;. Latjava.java
```

8. Setelah kompilasi berhasil, running file Latjava.java menggunakan perintah seperti berikut :

```
$ java -cp ./;. Latjava
```

9. Setelah running berhasil, hasil output akan tampil seperti dibawah ini :

```
method konkrit dari class A  
method abstrak yang sudah menjadi konkrit dalam class B
```

Praktikum Soal 3

1. Buatlah file GraphicObject.java.
2. Edit file GraphicObject.java dengan sintaks seperti ini :

```
abstract class GraphicObject {
    int x,y;
    void moveTo(int newValueX, int newValueY) {
        x = newValueX;
        y = newValueY;
    }
    abstract void drawShape();
    abstract void resize();
}
```

3. Buatlah file Cicle.java.
4. Edit file Cicle.java dengan sintaks seperti ini :

```
public class Circle extends GraphicObject {
    void resize() {
        System.out.println("Circle Resize");
    }
    void drawShape() {
        System.out.println("Circle Shape");
    }
    void position() {
        System.out.println(this.x + this.y);
    }
}
```

5. Buatlah file Main.java.
6. Edit file Main.java dengan sintaks seperti ini :

```
public class Main {
    public static void main(String[] args) {
        Circle Lingkaran = new Circle();
        Lingkaran.moveTo(5,5);
        Lingkaran.drawShape();
        Lingkaran.resize();
        Lingkaran.position();
    }
}
```

7. Compile file Main.java menggunakan java compiler (javac) dengan -cp :

```
$ javac -cp ./;. Main.java
```

8. Setelah kompilasi berhasil, running file Main.java

menggunakan perintah seperti berikut :

```
$ java -cp ./;. Main
```

9. Setelah running berhasil, hasil output akan tampil seperti dibawah ini :

```
Circle Shape  
Circle Resize  
10
```

Praktikum Soal 4

1. Buatlah file Vehicle.java.
2. Edit file Vehicle.java dengan sintaks seperti ini :

```
abstract class Vehicle {  
    int numofGears;  
    String color;  
    abstract boolean hasDiskBrake();  
    abstract int getNoofGears();  
}
```

3. Buatlah file Shape.java.
4. Edit file Shape.java dengan sintaks seperti ini :

```
abstract class Shape {  
    public String color;  
    public Shape() {  
    }  
  
    public void setColor(String c) {  
        color = c;  
    }  
  
    public String getColor() {  
        return color;  
    }  
  
    abstract public double area();  
}
```

5. Buatlah file Point.java.
6. Edit file Point.java dengan sintaks seperti ini :

```
public class Point extends Shape {  
    static int x, y;  
    public Point() {  
        x = 0;
```

```

        y = 0;
    }

    public double area() {
        return 0;
    }

    public double perimeter() {
        return 0;
    }

    public static void print() {
        System.out.println("point: " + x + "," + y);
    }

    public static void main(String args[]) {
        Point p = new Point();
        p.print();
    }
}

```

7. Compile file Point.java menggunakan java compiler (javac) dengan -cp :

```
$ javac -cp ./;. Point.java
```

8. Setelah kompilasi berhasil, running file Point.java menggunakan perintah seperti berikut :

```
$ java -cp ./;. Point
```

9. Setelah running berhasil, hasil output akan tampil seperti dibawah ini :

```
point: 0,0
```

Praktikum Soal 5

1. Buatlah file Mobil.java.
2. Edit file Mobil.java dengan sintaks seperti ini :

```

abstract public class Mobil {
    public void injakPedalGas() {

    }

    public void injakRem() {
        System.out.println("Mobil berhenti!");
    }
}

```

```
}
```

3. Buatlah file Kijang.java.

4. Edit file Kijang.java dengan sintaks seperti ini :

```
public class Kijang extends Mobil{
    public void injakPedalGas(){
        System.out.println("Mobil Melaju dengan kecepatan 80
        Km/jam...");
    }
}
```

5. Buatlah file BMW.java.

6. Edit file BMW.java dengan sintaks seperti ini :

```
public class BMW extends Mobil {
    public void injakPedalGas(){
        System.out.println("Mobil Melaju dengan kecepatan
        100 Km/jam...");
    }
}
```

7. Buatlah file TestMobil.java.

8. Edit file TestMobil.java dengan sintaks seperti ini :

```
public class TestMobil {
    public static void main(String[] args){
        Mobil mobil = new Mobil(){
    public void injakPedalGas(){
        System.out.println("Mobil berjalan...");
    }
};

        Kijang kijang = new Kijang();
        BMW bmw = new BMW();

        mobil.injakPedalGas();

        mobil = kijang;
        mobil.injakPedalGas();

        mobil = bmw;
        mobil.injakPedalGas();
    }
}
```

9. Compile file TestMobil.java menggunakan java compiler (javac) dengan -cp :


```
$ javac -cp ./;. TestMobil.java
```

10. Setelah kompilasi berhasil, running file TestMobil.java menggunakan perintah seperti berikut :

```
$ java -cp ./;. TestMobil
```

11. Setelah proses running berhasil, lihat hasil running, jika berhasil akan muncul tampilan seperti di bawah ini :

```
Mobil berjalan...  
Mobil Melaju dengan kecepatan 80 Km/jam...  
Mobil Melaju dengan kecepatan 100 Km/jam...
```