LAPORAN PRAKTIKUM

Jobsheet 10 – Abstract

Disusun sebagai

Mata Kuliah :

Pemrograman Berbasis Object

Oleh :

Siska Nuri Aprilia

Sib 2F

2341760038

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

POLITEKNIK NEGERI MALANG

TAHUN 2024/2025

JOBSHEET W08

ABSTRACT CLASS

1. KOMPETENSI

1. Memahami konsep dasar dan tujuan abstract class
2. Mampu menerapkan abstract class dalam suatu kode program
3. Mampu membuat subclass yang meng-extend abstract class dengan mengimplementasikan seluruh abstract method-nya

2. PENDAHULUAN

Abstract class merupakan class yang **tidak dapat diinstansiasi namun dapat di-extend**. Umumnya abstract class digunakan sebagai **generalisasi** atau **guideline** dari subclass dan hanya bisa digunakan lebih lanjut setelah **di-extend** oleh **concrete class** (class pada umumnya)

Abstract class memiliki karakteristik sebagai berikut:

1. Selalu dideklarasikan dengan menggunakan keyword "**abstract class**"
2. Dapat memiliki atribut dan methods (yang bukan abstract method) seperti concrete class
3. Umumnya memiliki **abstract method**, yaitu method yang hanya dideklarasikan tetapi tidak memiliki implementasi (body)
    - Abstract method mendifinisikan apa saja yang bisa dilakukan oleh sebuah class namun tidak ada detail bagaimana cara melakukannya
    - Untuk membuat abstract method, hanya menuliskan deklarasi method tanpa body dan menggunakan keyword abstract

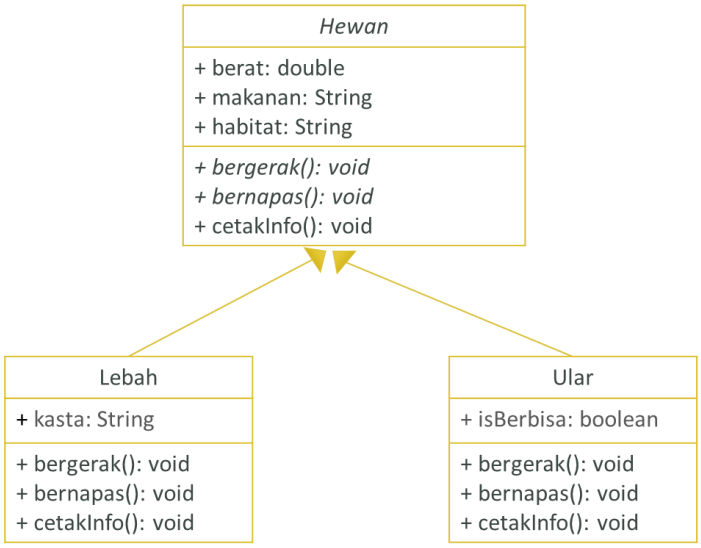
Untuk mendeklarasikan abstract class:
public abstract class <NamaClass> Untuk
mendeklarasikan method abstract:

public abstract <return_type> <namaMethod>();

Contoh:

```
public abstract class Hewan {
    public abstract void bergerak();
    public abstract void bernapas();
}
```

Secara umum, notasi class diagram untuk abstract class sama dengan concrete class, namun nama kelas dicetak miring atau ditambah anotasi <<abstract>> di atas nama kelas. Di samping itu abstract method harus dicetak miring juga seperti pada contoh berikut.



Cara menggunaan abstract class:

- Abstract class tidak dapat diinstansiasi (tidak dapat dibuat objectnya). Baris kode berikut akan memunculkan *compilation error* "Hewan is abstract; cannot be instantiated"

  Hewan hewan1 = new Hewan();

- Untuk menggunakan abstract class, dibuat concrete class yang meng-extend abstract class tersebut

  o concrete class menggunakan extends keyword

  o concrete class harus mengimplementasi semua abstract method

- Class yang menge-extend abstract class tetapi tidak mengimplementasi seluruh abstract method nya maka harus dideklarasikan sebagai abstract class juga

Fungsi abstract class:

- Mencegah suatu class diinstansiasi atau dibuat objeknya
- Sebagai generalisasi/superclass pada class hierarki
- Sebagai guideline untuk subclass dengan cara memaksa subclass untuk mengimplementasikan abstract method

3. PERCOBAAN A. PERCOBAAN 1

1. Buatlah project baru dengan nama Praktikum08 kemudian buat class baru dengan nama Hewan. Method bernapas dan bergerak tidak memiliki statement atau baris kode.

```
public class Hewan {
    public double berat;
    public String makanan;
    public String habitat;

    public Hewan(double berat, String makanan, String habitat) {
        this.berat = berat;
        this.makanan = makanan;
        this.habitat = habitat;
    }

    public void bergerak(){
        //
    }

    public void bernapas(){
        //
    }

    public void cetakInfo(){
        System.out.println("Berat   : " + this.berat);
        System.out.println("Makanan : " + this.makanan);
        System.out.println("Habitat : " + this.habitat);
    }
}
```

2. Buat class Lebah sebagai subclass dari class Hewan sebagai berikut

```
public class Lebah extends Hewan{
    public String kasta;

    public Lebah(String kasta, double berat, String makanan, String habitat) {
        super(berat, makanan, habitat);
        this.kasta = kasta;
    }
}
```

3. Buat class main dengan nama AbstractClassDemo lalu instansiasi objek dari class Hewan dan class Lebah. Run program kemudian amati hasilnya.

```
public class AbstractClassDemo {
    public static void main(String[] args) {
        Hewan hewan1 = new Hewan(10, "Rumput", "Savana");
        hewan1.cetakInfo();
        hewan1.bergerak();
        hewan1.bernapas();

        Lebah lebah1 = new Lebah("Ratu", 0.05, "Nektar", "Hutan");
        lebah1.cetakInfo();
        lebah1.bergerak();
        lebah1.bernapas();
    }
}
```

Jawaban :

Class Hewan

```
public class Hewan {
    public double berat;
    public String makanan;
    public String habitat;

    public Hewan(double berat, String makanan, String habitat) {
        this.berat = berat;
        this.makanan = makanan;
        this.habitat = habitat;
    }

    public void bergerak(){
        //
    }

    public void bernapas (){
        //
    }

    public void cetakInfo(){
        System.out.println("Berat : " + this.berat);
        System.out.println("Makanan : " + this.makanan);
        System.out.println("Habitat : " + this.habitat);
    }
}
```

Class Lebah

```
public class Lebah extends Hewan {
    public String kasta;

    public Lebah(String kasta, double berat, String makanan,String habitat) {
        super(berat, makanan, habitat);
        this.kasta = kasta;
    }
}
```

AbstractClassDemo

```
public class AbstractClassDemo {
    Run | Debug
    public static void main(String[] args) {
        Hewan hewan1 = new Hewan(berat:10,  makanan:"Rumput", habitat:"Savana");
        hewan1.cetakInfo();
        hewan1.bergerak();
        hewan1.bernapas();

        Lebah lebah1 = new Lebah(kasta:"Ratu", berat:0.05, makanan:"Nektar", habitat:"Hutan");
        lebah1.cetakInfo();
        lebah1.bergerak();
        lebah1.bernapas();
    }

}
```

Output

```
Berat   : 10.0
Makanan : Rumput
Habitat : Savana
Berat   : 0.05
Makanan : Nektar
Habitat : Hutan
```

B. PERTANYAAN

1. Bagaimana hasil pada langkah 3? Apakah objek hewan1 dan lebah1 berhasil diinstansiasi?

   Jawaban :

   objek hewan1 dan lebah1 berhasil diinstansiasi

2. Menurut Anda, mengapa tidak ada baris program pada method bergerak() dan bernapas() pada class Hewan()?

   Jawaban :

   karena abstract ini tidak jelas, method bergerak() dan bernafas ini pada tiap hewan akan berbeda setiap jenisnya. Setiap jenis hewan memiliki cara bergerak dan bernapas yang unik (beda) misalnya, ikan bergerak dengan cara berenang, sementara burung bergerak dengan cara terbang. method ini tidak ada isi agar kelas turunan, seperti Lebah, dapat

menentukan sendiri cara mereka bergerak dan bernapas, sesuai karakteristik spesifik mereka sendiri

3. Class Lebah tidak memiliki method bergerak(), bernapas(), dan cetakInfo(), mengapa tidak terjadi error pada AbstractClassDemo?

   Jawaban :

   Tidak terjadi error pada AbstractClassDemo, meskipun pada kelas Lebah tidak memiliki method bergerak(), bernapas(), dan cetakInfo(). Ini terjadi karena Lebah mewarisi method tersebut dari kelas induknya (super class) , yaitu Hewan. Karena Hewan mengimplementasi method bergerak(), bernapas(), dan cetakInfo(), kelas Lebah otomatis memiliki akses ke method bergerak(), bernapas(), dan cetakInfo() tersebut, sehingga tidak harus mendefinisikannya kembali.

C. PERCOBAAN 2

1. Ubah method bergerak dan bernapas menjadi abstract method.

```
public class Hewan {
    public double berat;
    public String makanan;
    public String habitat;

    public Hewan(double berat, String makanan, String habitat) {
        this.berat = berat;
        this.makanan = makanan;
        this.habitat = habitat;
    }

    public abstract void bergerak();
    public abstract void bernapas();

    public void cetakInfo(){
        System.out.println("Berat   : " + this.berat);
        System.out.println("Makanan : " + this.makanan);
        System.out.println("Habitat : " + this.habitat);
    }
}
```

Jawaban :

```
public abstract void bergerak();
public abstract void bernapas();
```

2. Akan muncul error sebagai berikut

```
public class Hewan {
    public doubl
    public Strin
    public Strin
```

Hewan is not abstract and does not override abstract method bernapas() in Hewan
---
(Alt-Enter shows hints)

Jawaban :

```
public class Hewan {
    public do
    public St
    public St
    public Hewan(double berat, String makanan, String habitat) {
```

The type Hewan must be an abstract class to define abstract methods Java(16777549)

Hewan

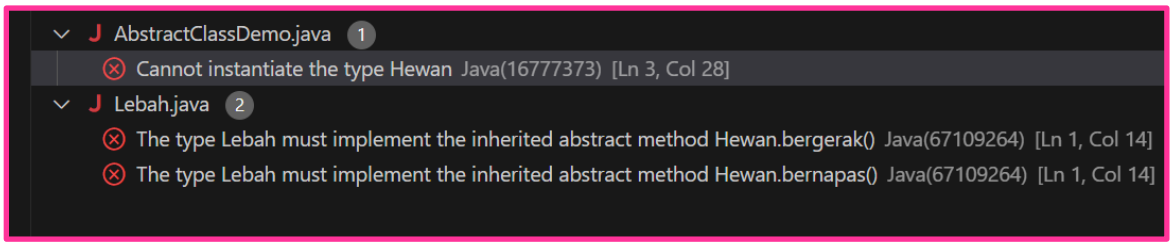
View Problem (Alt+F8) Quick Fix... (Ctrl+.)

3. Ubah class Hewan menjadi abstract Class. Jalankan program kemudian amati hasilnya.

```
public abstract class Hewan {
    public double berat;
    public String makanan;
    public String habitat;

    public Hewan(double berat, String makanan, String habitat) {
        this.berat = berat;
        this.makanan = makanan;
        this.habitat = habitat;
    }

    public abstract void bergerak();
    public abstract void bernapas();

    public void cetakInfo(){
        System.out.println("Berat   : " + this.berat);
        System.out.println("Makanan : " + this.makanan);
        System.out.println("Habitat : " + this.habitat);
    }
}
```

Jawaban :

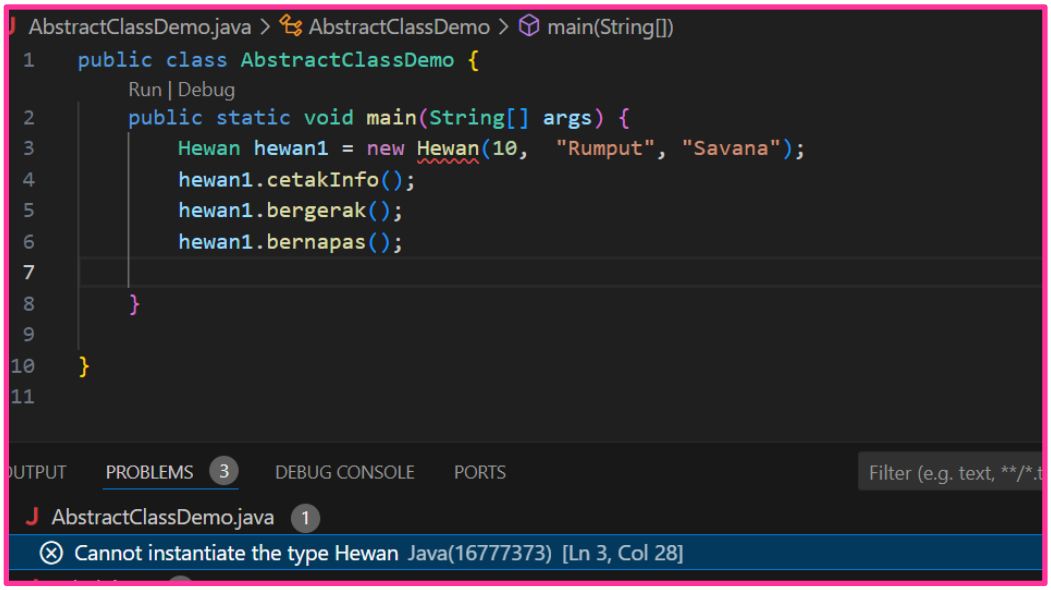
```java
public abstract class Hewan {
    public double berat;
    public String makanan;
    public String habitat;

    public Hewan(double berat, String makanan, String habitat) {
        this.berat = berat;
        this.makanan = makanan;
        this.habitat = habitat;
    }

    public abstract void bergerak();
    public abstract void bernapas();

    public void cetakInfo(){
        System.out.println("Berat   : " + this.berat);
        System.out.println("Makanan : " + this.makanan);
        System.out.println("Habitat : " + this.habitat);
    }
}
```



4. Ubah class demo sebagai berikut. Run program kemudian amati hasilnya

```java
public class AbstractClassDemo {
    public static void main(String[] args) {
        Hewan hewan1 = new Hewan(10, "Rumput", "Savana");
        hewan1.cetakInfo();
        hewan1.bergerak();
        hewan1.bernapas();
    }
}
```

Jawaban :



5. Ubah class demo sebagai berikut. Run program kemudian amati hasilnya

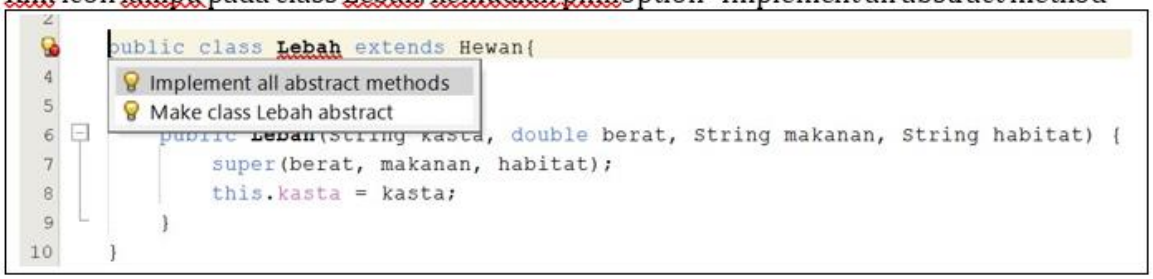
```
public class AbstractClassDemo {
    public static void main(String[] args) {
        Lebah lebah1 = new Lebah("Ratu", 0.05, "Nektar", "Hutan");
        lebah1.cetakInfo();
        lebah1.bergerak();
        lebah1.bernapas();
    }
}
```

Jawaban :

```
public class AbstractClassDemo {
    Run | Debug
    public static void main(String[] args) {
        Lebah lebah1 = new Lebah(kasta:"Ratu",  berat:0.05, makanan:"nextar", habitat:"Hutan");
        lebah1.cetakInfo();
        lebah1.bergerak();
        lebah1.bernapas();

    }

}
```

```
Berat   : 0.05
Makanan : nextar
Habitat : Hutan
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
        The type Lebah must implement the inherited abstract method Hewan.bergerak()

        at Lebah.bergerak(Lebah.java:1)
        at AbstractClassDemo.main(AbstractClassDemo.java:5)
```

6. Klik icon lampu pada class Lebah, kemudian pilihoption "Implement all abstract method"



Jawaban :

```
public class Lebah extends Hewan {
    public String kasta;

    public Lebah(String kasta, double berat, String makanan,String habitat) {
        super(berat, makanan, habitat);
        this.kasta = kasta;
    }

    @Override
    public void bergerak() {
        // TODO Auto-generated method stub
        throw new UnsupportedOperationException(message:"Unimplemented method 'bergerak'");
    }

    @Override
    public void bernapas() {
        // TODO Auto-generated method stub
        throw new UnsupportedOperationException(message:"Unimplemented method 'bernapas'");
    }
}
```

7. Implementasi method bergerak dan bernapas pada class Lebah sebagai berikut. Run program kemudian amati hasilnya.

```
@Override
public void bernapas(){
    System.out.println("Otot perut mengendur, udara masuk melalui lubang di segmen tubuh");
    System.out.println("Trakea mengirimkan oksigen");
    System.out.println("Otot perut berkontraksi, udara dikeluarkan");
}

@Override
public void bergerak(){
    System.out.println("Mengepakkan sayap ke depan");
    System.out.println("Memutar sayap hampir 90 derajat");
    System.out.println("Mengepakkan sayap ke belakang");
}
```

Jawaban :

```
@Override
public void bernapas() {
    // TODO Auto-generated method stub
    System.out.println(x:"Otot perut mengenur, udara masuk melalui lubang isegmen tubuh");
    System.out.println(x:"Trakea mengirimkan oksigen");
    System.out.println(x:"Otot perut berkontraksi, udara dikeluarkan");
}

@Override
public void bergerak() {
    // TODO Auto-generated method stub
    System.out.println(x:"Mengepakkan sayap kedepan");
    System.out.println(x:"memutar sayap hampir 90 derajat ");
    System.out.println(x:"Mengepakkan sayap kebelakang");
}
```

8. Tambahkan method cetakInfo() pada class Lebah. Run program kemudian amati hasilnya.

```
@Override
public void cetakInfo() {
    super.cetakInfo();
    System.out.println("Kasta    :" + this.kasta);
}
```

Jawaban :

```java
@Override
public void cetakInfo() {
    // TODO Auto-generated method stub
    super.cetakInfo();
    System.out.println("Kasta : " + this.kasta);
}
```

```
Berat   : 0.05
Makanan : nextar
Habitat : Hutan
Kasta   : Ratu
Mengepakkan sayap kedepan
memutar sayap hampir 90 derajat
Mengepakkan sayap kebelakang
Otot perut mengenur, udara masuk melalui lubang segmen tubuh
Trakea mengirimkan oksigen
Otot perut berkontraksi, udara dikeluarkan
```

9. Buat class Ular kemudian sebagai berikut. Instansiasi objek bertipe Ular pada class AbstractClassDemo. Ekseksusi ketiga method untuk object tersebut.

```java
public class Ular extends Hewan{
    public boolean isBerbisa;

    public Ular(boolean isBerbisa, double berat, String makanan, String habitat) {
        super(berat, makanan, habitat);
        this.isBerbisa = isBerbisa;
    }

    @Override
    public void bergerak(){
        System.out.println("Mengerutkan otot dari segala sisi hingga membentuk lengkungan");
        System.out.println("Menemukan titik penahan seperti batu atau pohon");
        System.out.println("Menggunakan sisik untuk mendorong tubah ke depan");
    }

    @Override
    public void bernapas(){
        System.out.println("Otot tulang rusuk kontraksi, udara masuk lewat hidung");
        System.out.println("Trakea mengirimkan udara ke paru-paru");
        System.out.println("Otot tulang rusuk relaksasi, udara dikeluarkan lewat hidung");
    }

    @Override
    public void cetakInfo() {
        super.cetakInfo();
        System.out.println("Berbisa    :" + (this.isBerbisa ? "Ya" : "Tidak"));
    }
}
```

Jawaban :

```java
public class Ular extends Hewan {
    public boolean isBerbisa;

    public Ular(boolean isBerbisa, double berat, String makanan, String habitat ) {
        super(berat, makanan, habitat);
        this.isBerbisa = isBerbisa;
    }

    @Override
    public void bergerak() {
        System.out.println(x:"Mengerutkan otot dari segala sisi hingga membentuk lengkungan");
        System.out.println(x:"Menemukan titik penahan seperti batu atau pohon");
        System.out.println(x:"Menggunakan sisik untuk mendorong tubuh ke depan");
    }

    @Override
    public void bernapas() {
        System.out.println(x:"Otot tulang rusuk konstraksi, udara masuk lewat hidung");
        System.out.println(x:"Trakea mengirimkan udara ke paru-paru");
        System.out.println(x:"Otot tulang relaksi, udara dikeluarkan lewat hidung");
    }

    @Override
    public void cetakInfo() {
        super.cetakInfo();
        System.out.println("Berbisa : " + (this.isBerbisa ? "Ya" : "Tidak"));
    }

}
```

```java
public class AbstractClassDemo {
    Run | Debug
    public static void main(String[] args) {
        Lebah lebah1 = new Lebah(kasta:"Ratu",  berat:0.05, makanan:"nextar", habitat:"Hutan");
        lebah1.cetakInfo();
        lebah1.bergerak();
        lebah1.bernapas();

        Ular ular1 = new Ular(isBerbisa:true, berat:0.05, makanan:"nextar", habitat:"Hutan");
        ular1.cetakInfo();
        ular1.bergerak();
        ular1.bernapas();

    }

}
```

```
Makanan : nextar
Habitat : Hutan
Kasta   : Ratu
Mengepakkan sayap kedepan
memutar sayap hampir 90 derajat
Mengepakkan sayap kebelakang
Otot perut mengenur, udara masuk melalui lubang segmen tubuh
Trakea mengirimkan oksigen
Otot perut berkontraksi, udara dikeluarkan
Berat   : 0.05
Makanan : nextar
Habitat : Hutan
Berbisa : Ya
Mengerutkan otot dari segala sisi hingga membentuk lengkungan
Menemukan titik penahan seperti batu atau pohon
Menggunakan sisik untuk mendorong tubuh ke depan
Otot tulang rusuk konstraksi, udara masuk lewat hidung
Trakea mengirimkan udara ke paru-paru
Otot tulang relaksi, udara dikeluarkan lewat hidung
```

```java
public class Ular extends Hewan {
    public boolean isBerbisa;

    public Ular(boolean isBerbisa, double berat, String makanan, String habitat ) {
        super(berat, makanan, habitat);
        this.isBerbisa = isBerbisa;
    }

    @Override
    public void bergerak() {
        System.out.println(x:"Mengerutkan otot dari segala sisi hingga membentuk lengkungan");
        System.out.println(x:"Menemukan titik penahan seperti batu atau pohon");
        System.out.println(x:"Menggunakan sisik untuk mendorong tubuh ke depan");
    }

    @Override
    public void bernapas() {
        System.out.println(x:"Otot tulang rusuk konstraksi, udara masuk lewat hidung");
        System.out.println(x:"Trakea mengirimkan udara ke paru-paru");
        System.out.println(x:"Otot tulang relaksi, udara dikeluarkan lewat hidung");
    }

    @Override
    public void cetakInfo() {
        super.cetakInfo();
        System.out.println("Berbisa : " + (this.isBerbisa ? "Ya" : "Tidak"));
    }

}
```

D. PERTANYAAN

1. Pada langkah 1, mengapa sebaiknya method bergerak() dan bernapas() dideklarasikan sebagai abstract method?

   Jawaban :

   Pada ada langkah 1, method bergerak() dan bernapas() sebaiknya dideklarasikan sebagai method abstrak (abstract) karena cara bergerak dan bernapas dilakukan secara berbeda-beda untuk setiap jenis hewan. Kelas Hewan adalah kelas umum yang berarti hanya mendefinisikan atribut dan perilaku - perilaku yang dimiliki semua hewan, tetapi tidak bisa menentukan cara pastinya bagaimana setiap hewan bergerak dan bernapas. Misalnya:

   1. Burung bergerak dengan terbang, sedangkan ikan bergerak dengan berenang.
   2. Ikan bernapas melalui insang, sementara mamalia bernapas menggunakan paru-paru.

2. Mengapa pada langkah 2 muncul error?

   Jawaban :

   Pada langkah 2, error terjadi karena kode pemrograman modifikasi tersebut mencoba untuk mendeklarasikan kelas Hewan sebagai kelas biasa (bukan abstract) tetapi memiliki metode bergerak() dan bernapas() yang dideklarasikan sebagai metode abstract. Hal itu lah yang menyebabkan pada langkah 2 ini eror

3. Apakah sebuah class yang memiliki abstract method harus dideklarasikan sebagai abstract class?

   Jawaban :

   iyaa, sebuah kelas yang memiliki abstract method harus dideklarasikan sebagai *abstract class*. Ini karena metode abstrak tidak memiliki implementasi, dan tujuannya adalah agar kelas turunan yang mengonkretkan spesifiknya atau Class yang menge-extend abstract class tetapi tidak mengimplementasi seluruh abstract method nya maka harus dideklarasikan sebagai abstract class juga

4. Sebaliknya, apakah abstract class harus memiliki abstract method?

   Jawaban :

   Kelas abstrak tidak diharuskan memiliki method abstract di dalamnya . Namun, kelas apa pun yang memiliki method abstract di dalamnya atau yang tidak menyediakan implementasi untuk method abstract apa pun yang dideklarasikan di superkelasnya harus dideklarasikan sebagai kelas abstract

5. Mengapa muncul error pada langkah 4?

   Jawaban :

   Error pada langkah 4 terjadi karena kelas Lebah belum mengimplementasikan method abstract bergerak() dan bernapas() yang ada di kelas induknya (superclass), yaitu Hewan. Ketika sebuah kelas turunan mewarisi kelas abstract yang memiliki method abstract, kelas turunan tersebut harus mengimplementasikan method – method abstract tersebut

6. Apakah abstract class dapat memiliki constructor?

   Jawaban :

   iyaaa, kelas Abstract selalu memiliki konstruktor

7. Apakah constructor abstract class dapat dipanggil?

   Jawaban :

   iyaaa, konstruktor di dalam kelas abstract dapat dipanggil, tetapi tidak secara langsung. Artinya, kita tidak bisa membuat objek dari kelas abstract itu sendiri. Namun, ketika kita membuat kelas turunan (kelas yang mewarisi dari kelas abstrak), konstruktor di kelas turunan itu bisa memanggil konstruktor di kelas abstrak

8. Pada langkah 6-8, mengapa method bergerak() dan bernapas() **harus** di-override, namun method cetakInfo() **tidak harus** di-override?

   Jawaban :

   Pada langkah 6-8, metho bergerak() dan bernapas() harus di-override karena keduanya dideklarasikan sebagai method abstract pada kelas Hewan. Artinya, setiap kelas yang diturunkan dari Hewan harus dimplementasi agar bisa menjelaskan cara hewan tersebut

bergerak dan bernapas. Jika tidak di-override,kode program tidak akan dapat dikompilasi karena ia menganggap tidak lengkap. Kemudian pada method cetakInfo() tidak perlu di-override karena sudah memiliki implementasi default dalam kelas Hewan

9. Simpulkan kegunaan dari abstract method

Jawaban :

Abstract method itu seperti aturan yang harus diikuti oleh kelas-kelas turunanya. Artinya setiap subclass harus memiliki cara sendiri untuk melakukan suatu tugas, tetapi tidak menjelaskan bagaimana cara melakukannya (pada praktikum ini misanya yaitu cara bernafas dan bergerak pada hewan lebah dan ular). sehingga semua subclass memiliki cara sendiri dan sesuai untuk menjalankan method tersebut

10. Simpulkan kegunaan dari abstract class

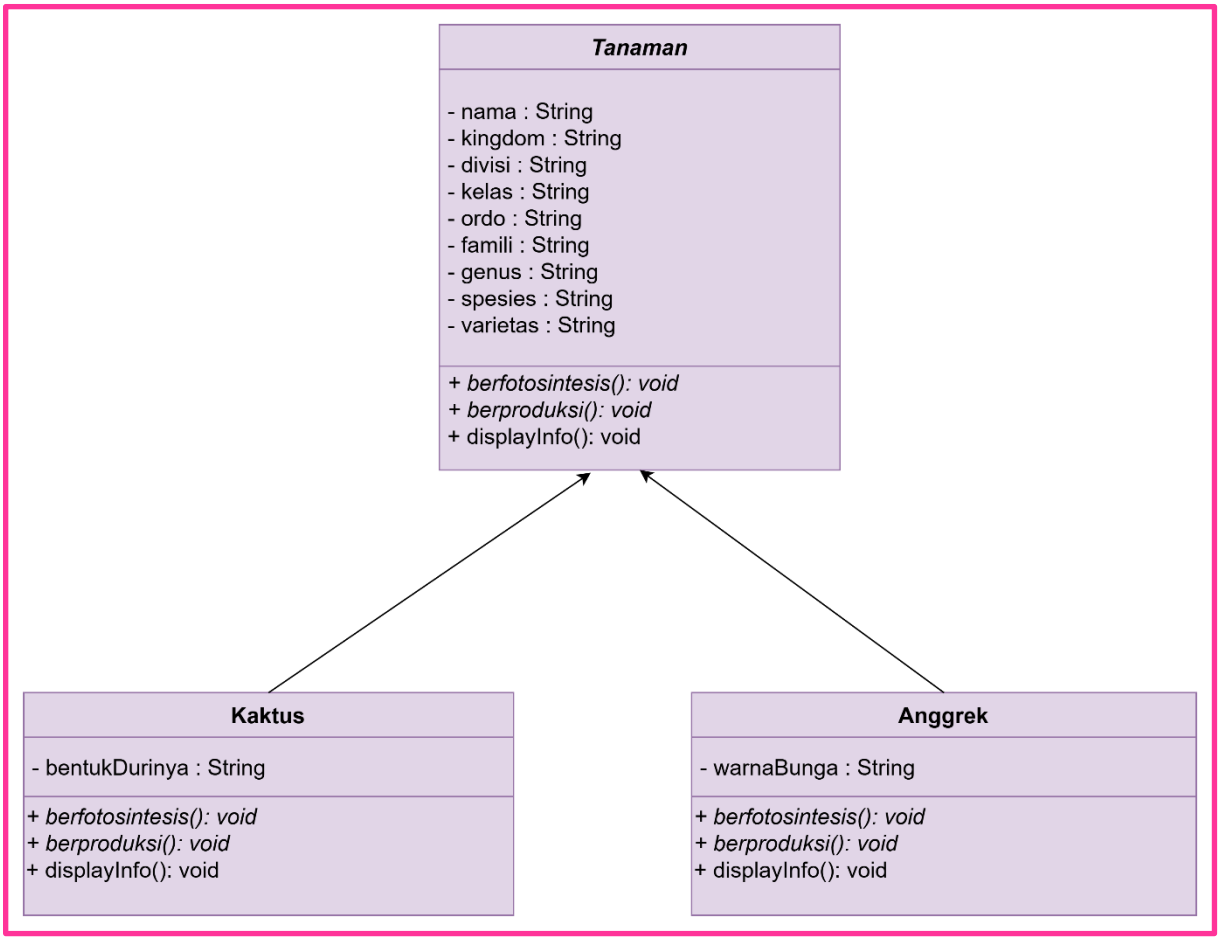
Jawaban :

Abstract class digunakan sebagai generalisasi atau guideline dari subclass dan hanya bisa digunakan lebih lanjut setelah di-extend oleh concrete class (class pada umumnya). Di dalam abstract class, kita juga bisa memiliki abstract method yang harus diisi oleh subclass, sehingga semua subclass mengikuti aturan yang sama

4. TUGAS

Implementasikan class diagram yang telah dirancang pada tugas PBO Teori ke dalam kode program. Selanjutnya buatlah instansiasi objek dari masing-masing subclass kemudian coba eksekusi method-method yang dimiliki.

1. Class Diagram



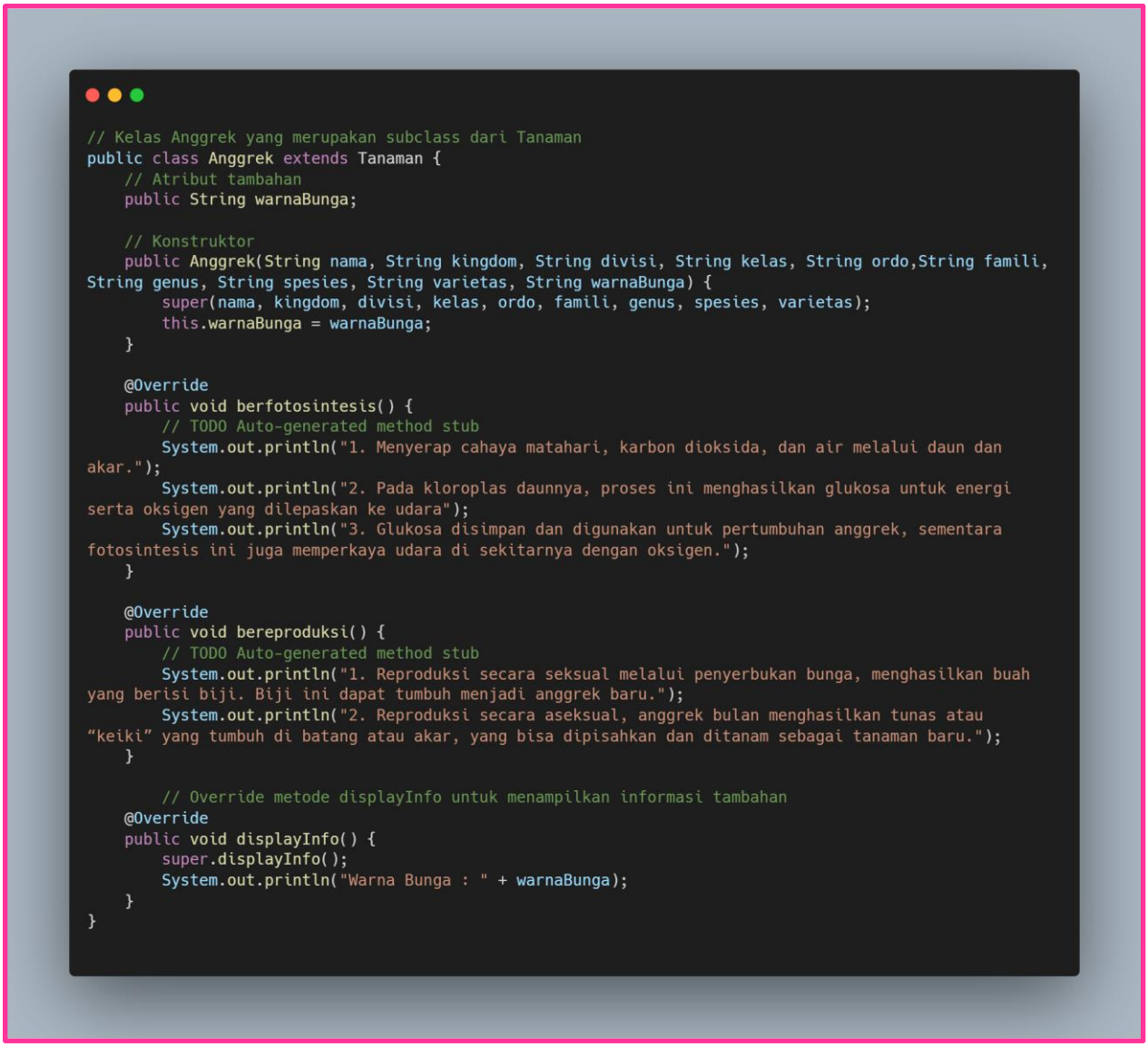
2. Class Tumbuhan

```java
// Kelas Tanaman sebagai superclass
public abstract class Tanaman {
    // Atribut
    public String nama;
    public String kingdom;
    public String divisi;
    public String kelas;
    public String ordo;
    public String famili;
    public String genus;
    public String spesies;
    public String varietas;

    // Konstruktor
    public Tanaman(String nama, String kingdom, String divisi, String kelas, String ordo,String famili,
String genus, String spesies, String varietas) {
        this.nama = nama;
        this.kingdom = kingdom;
        this.divisi = divisi;
        this.kelas = kelas;
        this.ordo = ordo;
        this.famili = famili;
        this.genus = genus;
        this.spesies = spesies;
        this.varietas = varietas;
    }

    // Metode berfotosintesis
    public abstract void berfotosintesis();

    // Metode berproduksi
    public abstract void bereproduksi();



    // Metode untuk menampilkan informasi
    public void displayInfo() {
        System.out.println("Nama        : " + nama);
        System.out.println("Kingdom     : " + kingdom);
        System.out.println("Divisi      : " + divisi);
        System.out.println("Kelas       : " + kelas);
        System.out.println("Ordo        : " + ordo);
        System.out.println("Famili      : " + famili);
        System.out.println("Genus       : " + genus);
        System.out.println("Spesies     : " + spesies);
        System.out.println("Varietas    : " + varietas);
    }
}
```

3. Class Kaktus

```java
// Kelas Kaktus yang merupakan subclass dari Tanaman
public class Kaktus extends Tanaman {
    // Atribut tambahan
    public String bentukDurinya;

    // Konstruktor
    public Kaktus(String nama, String kingdom, String divisi, String kelas, String ordo,String famili,
String genus, String spesies, String varietas, String bentukDurinya) {
        super(nama, kingdom, divisi, kelas, ordo, famili, genus, spesies, varietas);
        this.bentukDurinya = bentukDurinya;
    }

    @Override
    public void berfotosintesis() {
        System.out.println("1. Fotosintesis pada kaktus dimulai dengan penyerapan cahaya matahari oleh
klorofil yang terdapat di batangnya.");
        System.err.println("2. Melalui stomata yang terbuka pada malam hari.");
        System.out.println("3. Setelah penyerapan cahaya matahari selesai,kaktus mengambil karbon
dioksida dari udara dan menggunakan air yang disimpan dalam jaringan batang.");
    }

    @Override
    public void bereproduksi() {
       System.out.println("1. Reproduksi secara seksual dengan menghasilkan bunga yang diserbuki oleh
serangga, burung, atau angin, lalu membentuk buah berisi biji yang dapat tumbuh menjadi kaktus baru.");
       System.out.println("2. Reproduksi secara aseksual, kaktus menghasilkan tunas di batangnya atau
diperbanyak dengan memotong batang dan menanamnya, di mana potongan tersebut akan mengembangkan akar
dan tumbuh sebagai kaktus baru.");

    }
       // Override metode displayInfo untuk menampilkan informasi tambahan
    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Bentuk Durinya : " + bentukDurinya);

    }
}
```

4. Class Anggrek

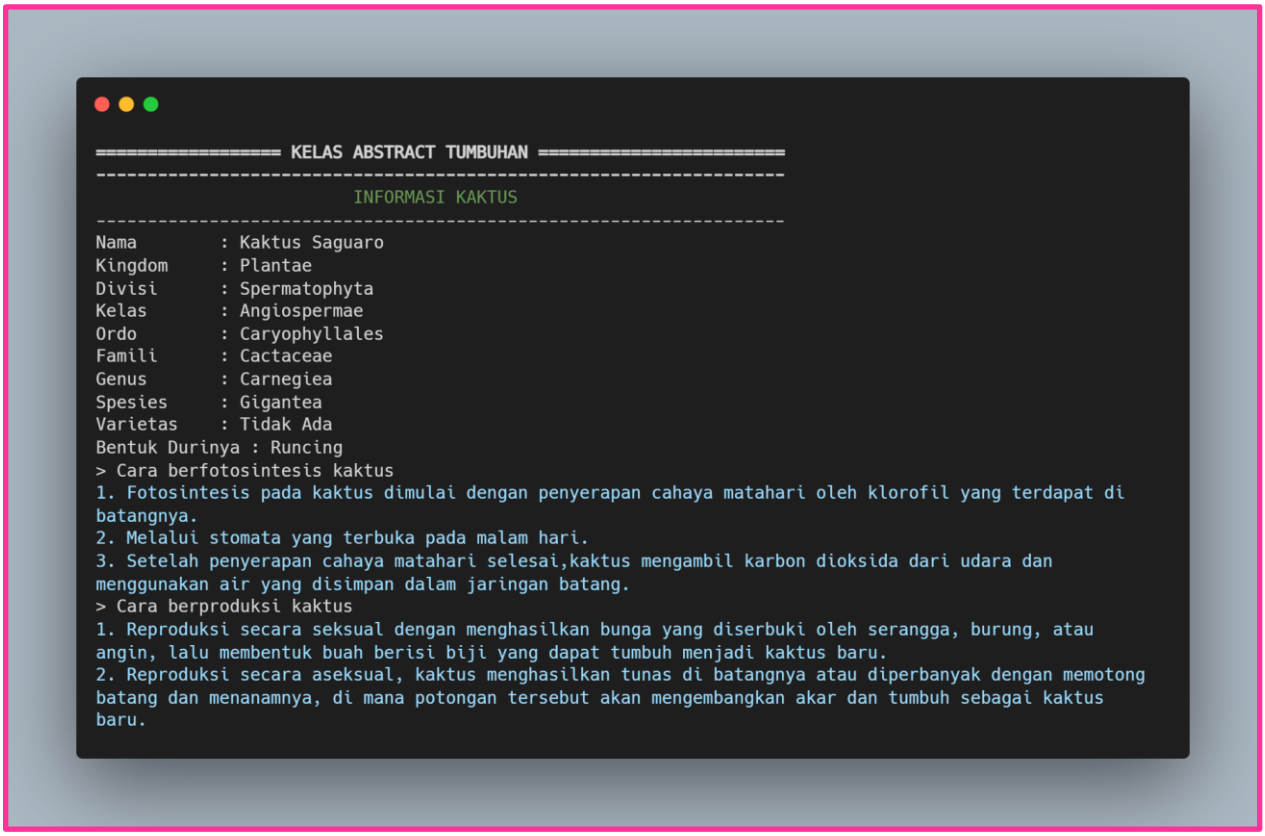


```java
// Kelas Anggrek yang merupakan subclass dari Tanaman
public class Anggrek extends Tanaman {
    // Atribut tambahan
    public String warnaBunga;

    // Konstruktor
    public Anggrek(String nama, String kingdom, String divisi, String kelas, String ordo,String famili,
String genus, String spesies, String varietas, String warnaBunga) {
        super(nama, kingdom, divisi, kelas, ordo, famili, genus, spesies, varietas);
        this.warnaBunga = warnaBunga;
    }

    @Override
    public void berfotosintesis() {
        // TODO Auto-generated method stub
        System.out.println("1. Menyerap cahaya matahari, karbon dioksida, dan air melalui daun dan
akar.");
        System.out.println("2. Pada kloroplas daunnya, proses ini menghasilkan glukosa untuk energi
serta oksigen yang dilepaskan ke udara");
        System.out.println("3. Glukosa disimpan dan digunakan untuk pertumbuhan anggrek, sementara
fotosintesis ini juga memperkaya udara di sekitarnya dengan oksigen.");
    }

    @Override
    public void bereproduksi() {
        // TODO Auto-generated method stub
        System.out.println("1. Reproduksi secara seksual melalui penyerbukan bunga, menghasilkan buah
yang berisi biji. Biji ini dapat tumbuh menjadi anggrek baru.");
        System.out.println("2. Reproduksi secara aseksual, anggrek bulan menghasilkan tunas atau
“keiki” yang tumbuh di batang atau akar, yang bisa dipisahkan dan ditanam sebagai tanaman baru.");
    }

        // Override metode displayInfo untuk menampilkan informasi tambahan
    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Warna Bunga : " + warnaBunga);
    }
}
```

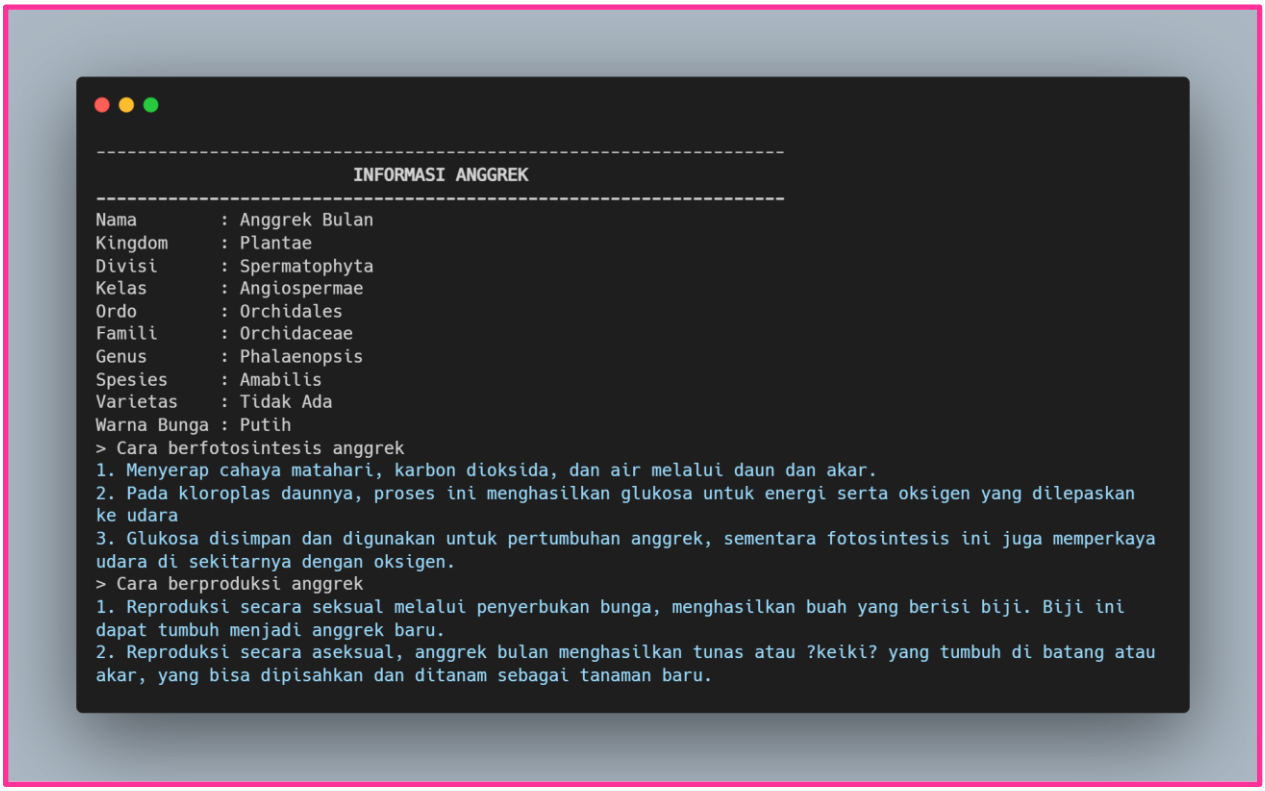
5. Class Main

```java
public class Main {
    public static void main(String[] args) {
        System.out.println("================== KELAS ABSTRACT TUMBUHAN ========================");

        // Membuat objek Kaktus
        Kaktus kaktus = new Kaktus("Kaktus Saguaro", "Plantae", "Spermatophyta", "Angiospermae",
        "Caryophyllales", "Cactaceae", "Carnegiea", "Gigantea","Tidak Ada", "Runcing");

        System.out.println("-------------------------------------------------------------------");
        System.out.println("                        INFORMASI KAKTUS                           ");
        System.out.println("-------------------------------------------------------------------");
        kaktus.displayInfo();
        System.out.println("> Cara berfotosintesis kaktus  ");
        kaktus.berfotosintesis();
        System.out.println("> Cara berproduksi kaktus      ");
        kaktus.bereproduksi();


        // Membuat objek Anggrek
        Anggrek anggrek = new Anggrek("Anggrek Bulan", "Plantae", "Spermatophyta", "Angiospermae",
        "Orchidales", "Orchidaceae", "Phalaenopsis", "Amabilis", "Tidak Ada", "Putih");

        System.out.println("-------------------------------------------------------------------");
        System.out.println("                        INFORMASI ANGGREK                          ");
        System.out.println("-------------------------------------------------------------------");
        anggrek.displayInfo();
        System.out.println("> Cara berfotosintesis anggrek ");
        anggrek.berfotosintesis();
        System.out.println("> Cara berproduksi anggrek     ");
        anggrek.bereproduksi();

    }
}
```

6. Output



```
================== KELAS ABSTRACT TUMBUHAN ======================
------------------------------------------------------------------
                        INFORMASI KAKTUS
------------------------------------------------------------------
Nama         : Kaktus Saguaro
Kingdom      : Plantae
Divisi       : Spermatophyta
Kelas        : Angiospermae
Ordo         : Caryophyllales
Famili       : Cactaceae
Genus        : Carnegiea
Spesies      : Gigantea
Varietas     : Tidak Ada
Bentuk Durinya : Runcing
> Cara berfotosintesis kaktus
1. Fotosintesis pada kaktus dimulai dengan penyerapan cahaya matahari oleh klorofil yang terdapat di
batangnya.
2. Melalui stomata yang terbuka pada malam hari.
3. Setelah penyerapan cahaya matahari selesai,kaktus mengambil karbon dioksida dari udara dan
menggunakan air yang disimpan dalam jaringan batang.
> Cara berproduksi kaktus
1. Reproduksi secara seksual dengan menghasilkan bunga yang diserbuki oleh serangga, burung, atau
angin, lalu membentuk buah berisi biji yang dapat tumbuh menjadi kaktus baru.
2. Reproduksi secara aseksual, kaktus menghasilkan tunas di batangnya atau diperbanyak dengan memotong
batang dan menanamnya, di mana potongan tersebut akan mengembangkan akar dan tumbuh sebagai kaktus
baru.
```


INFORMASI ANGGREK
Nama : Anggrek Bulan
Kingdom : Plantae
Divisi : Spermatophyta
Kelas : Angiospermae
Ordo : Orchidales
Famili : Orchidaceae
Genus : Phalaenopsis
Spesies : Amabilis
Varietas : Tidak Ada
Warna Bunga : Putih
> Cara berfotosintesis anggrek
1. Menyerap cahaya matahari, karbon dioksida, dan air melalui daun dan akar.
2. Pada kloroplas daunnya, proses ini menghasilkan glukosa untuk energi serta oksigen yang dilepaskan ke udara
3. Glukosa disimpan dan digunakan untuk pertumbuhan anggrek, sementara fotosintesis ini juga memperkaya udara di sekitarnya dengan oksigen.
> Cara berproduksi anggrek
1. Reproduksi secara seksual melalui penyerbukan bunga, menghasilkan buah yang berisi biji. Biji ini dapat tumbuh menjadi anggrek baru.
2. Reproduksi secara aseksual, anggrek bulan menghasilkan tunas atau ?keiki? yang tumbuh di batang atau akar, yang bisa dipisahkan dan ditanam sebagai tanaman baru.