



Dosen Pengampu:

Sayekti Harits Suryawan, M.Kom

KELOMPOK 24:

Ahmad Kasim Ramadhan	2211102441221
Syafrizal Sani Pratama	2211102441057
Farhan Difa Pratama	2211102441016

**PRODI S1 TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH
KALIMANTAN TIMUR
SAMARINDA
2023**

Codingan Class

MyWorld

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
```

```
public class Dunia extends World
```

```
{
```

```
    public int ratio = 500;
```

```
    public Dunia()
```

```
{
```

```
    // Create a new world with 600x400 cells with a cell size of 1x1 pixels.
```

```
    super(600, 400, 1);
```

```
    prepare();
```

```
}
```

```
public void act()
```

```
{
```

```
    anggur();
```

```
    apel();
```

```
    apricot();
```

```
    jeruk();
```

```
    lemon();
```

```
    mangga();
```

```
    nanas();
```

```

    peach();

    pear();

    semangka();

    bom();
}

public void anggur()
{
    if(Greenfoot.getRandomNumber(ratio) < 1)
    {
        addObject(new Anggur(), Greenfoot.getRandomNumber(600), 400);
    }
}

public void apel()
{
    if(Greenfoot.getRandomNumber(ratio) < 1)
    {
        addObject(new Apel(), Greenfoot.getRandomNumber(600), 400);
    }
}

public void apricot()
{
    if(Greenfoot.getRandomNumber(ratio) < 1)

```

```
{  
    addObject(new Apricot(), Greenfoot.getRandomNumber(600), 400);  
}  
}
```

```
public void jeruk()  
{  
    if(Greenfoot.getRandomNumber(ratio) < 1)  
    {  
        addObject(new Jeruk(), Greenfoot.getRandomNumber(600), 400);  
    }  
}
```

```
public void lemon()  
{  
    if(Greenfoot.getRandomNumber(ratio) < 1)  
    {  
        addObject(new Lemon(), Greenfoot.getRandomNumber(600), 400);  
    }  
}
```

```
public void mangga()  
{  
    if(Greenfoot.getRandomNumber(ratio) < 1)  
    {
```

```
        addObject(new Mangga(), Greenfoot.getRandomNumber(600), 400);  
    }  
}
```

```
public void nanas()  
{  
    if(Greenfoot.getRandomNumber(ratio) < 1)  
    {  
        addObject(new Nanas(), Greenfoot.getRandomNumber(600), 400);  
    }  
}
```

```
public void peach()  
{  
    if(Greenfoot.getRandomNumber(ratio) < 1)  
    {  
        addObject(new Peach(), Greenfoot.getRandomNumber(600), 400);  
    }  
}
```

```
public void pear()  
{  
  
    if(Greenfoot.getRandomNumber(ratio) < 1)  
    {
```

```

        Counter counter = new Counter();

        addObject(new Pear(counter), Greenfoot.getRandomNumber(600), 400);
    }
}

```

```

public void semangka()
{
    if(Greenfoot.getRandomNumber(ratio) < 1)
    {
        addObject(new Semangka(), Greenfoot.getRandomNumber(600), 400);
    }
}

```

```

public void bom()
{
    if(Greenfoot.getRandomNumber(500) < 1)
    {
        addObject(new Bom(), Greenfoot.getRandomNumber(600), 400);
    }
}

```

```

private void prepare()
{
    Counter skor = new Counter();

    addObject(skor, 55, 23);

    Kunai kunai = new Kunai(skor);
}

```

```
        addObject(kunai, 300, 200);  
  
    }  
  
}
```

Penjelasan:

MyWorld

- Dunia ini dibuat dengan ukuran 600x400 sel dan ukuran sel 1x1 piksel.
- Terdapat variabel **ratio** yang digunakan untuk mengontrol tingkat kemungkinan munculnya buah-buahan. Semakin kecil nilainya, semakin tinggi kemungkinan munculnya buah-buahan.
- Metode **act()** dipanggil setiap frame dan memanggil berbagai metode untuk menambahkan buah-buahan dan bom ke dunia.
- Setiap metode buah (seperti **anggur()**, **apel()**, dll.) memeriksa apakah nilai acak kurang dari **ratio**, dan jika iya, menambahkan buah ke dunia pada posisi acak.
- Terdapat metode untuk setiap jenis buah-buahan (**anggur()**, **apel()**, dll.) dan bom (**bom()**), yang menggunakan nilai acak untuk menentukan apakah objek tersebut akan ditambahkan ke dunia.
- Terdapat metode **prepare()** yang menambahkan pemain (**Kunai**) dan objek **Counter** ke dunia.
- Pemain (**Kunai**) memiliki parameter **Counter** yang digunakan untuk melacak skor permainan.
- Metode **pear()** menambahkan objek **Counter** ke dunia bersamaan dengan objek **Pear**. Hal ini menunjukkan adanya ketergantungan antara objek buah dan objek lainnya.
- Terdapat objek **Counter** yang menampilkan skor permainan dan ditempatkan di posisi (55, 23).

Codingan Class

BOM

```
import greenfoot.*;

public class Bom extends Actor
{
    public void act()
    {
        // Add your action code here.
        setLocation(getX(), getY() - 2);
        if(ujung())
        {
            getWorld().removeObject(this);
        }
        Actor kunai = getOneObjectAtOffset(0, 0, Kunai.class);
        if(kunai != null)
        {
            setImage("ledakan.png");
        }
    }
    public boolean ujung()
    {
        if(getY() < 75)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}
```



```

        return false;
    }
}
}

```

Penjelasan:

BOM

- **Bom** adalah kelas dasar yang merupakan subkelas dari **Actor**. Ini berarti objek dari kelas ini dapat muncul di dunia permainan Greenfoot.
- Terdapat metode **act()**, yang merupakan bagian dari siklus hidup objek dalam Greenfoot. Saat tombol "Act" atau "Run" ditekan, metode ini akan dipanggil.
- Metode ini mengatur beberapa tindakan untuk objek **Bom**:
 - **setLocation(getX(), getY() - 2)**: Menggerakkan objek **Bom** ke atas sejauh 2 piksel pada setiap pemanggilan.
 - **if(ujung())**: Memeriksa apakah objek **Bom** telah mencapai ujung atas dunia permainan (kurang dari 75 piksel). Jika iya, objek **Bom** dihapus dari dunia.
 - **Actor kunai = getOneObjectAtOffset(0, 0, Kunai.class)**: Memeriksa apakah terdapat objek **Kunai** pada posisi yang sama dengan objek **Bom**.
 - **if(kunai != null)**: Jika iya, mengganti gambar objek **Bom** dengan gambar ledakan (**setImage("ledakan.png")**).
- Metode **ujung()** memeriksa apakah objek **Bom** telah mencapai ujung atas dunia permainan (kurang dari 75 piksel).
- Mengembalikan nilai boolean yang menunjukkan apakah objek **Bom** berada pada ujung atas dunia permainan.

Codingan Class

Buah

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
```

```
public class Buah extends Actor
{
    public void act()
    {
        // Add your action code here.
    }

    public void gerak()
    {
        setLocation(getX(), getY() - 2);
    }

    public void gerakBelah()
    {
        setLocation(getX(), getY() + 2);
    }

    public boolean ujungBawah()
    {
        if(getY() > 375)
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public boolean ujung()
    {
        if(getY() < 75)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}
```

```

    }

    public boolean belah()
    {
        Actor kunai = getObjectAtOffset(0, 0, Kunai.class);
        return kunai != null;
    }
}

```

Penjelasan:

Buah

- **Buah** adalah kelas dasar yang merupakan subkelas dari **Actor**. Ini berarti objek dari kelas ini dapat muncul di dunia permainan Greenfoot.
- Terdapat metode **act()**, yang merupakan bagian dari siklus hidup objek dalam Greenfoot. Saat tombol "Act" atau "Run" ditekan, metode ini akan dipanggil. Saat ini, metode **act()** masih kosong dan tidak melakukan tindakan khusus.
- Metode **gerak()** menggerakkan objek **Buah** ke atas (koordinat Y berkurang) sejauh 2 piksel pada setiap pemanggilan.
- Metode **gerakBelah()** menggerakkan objek **Buah** ke bawah (koordinat Y bertambah) sejauh 2 piksel pada setiap pemanggilan.
- Metode-metode ini mungkin digunakan untuk mengatur pergerakan vertikal objek **Buah** dalam dunia permainan.
- Metode **ujungBawah()** memeriksa apakah objek **Buah** telah mencapai ujung bawah dunia permainan (lebih dari 375 piksel).
- Metode **ujung()** memeriksa apakah objek **Buah** telah mencapai ujung atas dunia permainan (kurang dari 75 piksel).
- Kedua metode ini mengembalikan nilai boolean yang menunjukkan apakah objek **Buah** berada di salah satu ujung vertikal dunia permainan.
- Metode **belah()** menggunakan metode **getObjectAtOffset()** untuk memeriksa apakah terdapat objek **Kunai** pada posisi yang sama dengan objek **Buah**.
- Jika iya, metode ini mengembalikan **true**, menandakan bahwa objek **Buah** berada pada posisi yang sama dengan objek **Kunai**.

Codingan Class

Anggur

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
```

```
public class Anggur extends Buah
{
    public void act()
    {
        // Add your action code here.
        gerak();
        if(belah())
        {
            getWorld().addObject(new AnggurBelah(), getX(), getY());
        }
        hilang();
    }

    public void hilang()
    {
        if(ujung())
        {
            getWorld().removeObject(this);
        }
    }
}
```

Anggur Belah

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
```

```
public class AnggurBelah extends Buah
{
    public void act()
    {
        // Add your action code here.
        gerakBelah();
        if(ujungBawah())
        {
            getWorld().removeObject(this);
        }
    }
}
```

Penjelasan:

Anggur:

- **Anggur** adalah kelas turunan yang mengextends kelas **Buah**, menunjukkan bahwa **Anggur** mewarisi sifat dan perilaku umum dari kelas **Buah**.
- Terdapat metode **act()**, yang merupakan bagian dari siklus hidup objek dalam Greenfoot. Saat tombol "Act" atau "Run" ditekan, metode ini akan dipanggil.
- Metode ini melakukan beberapa tindakan untuk objek **Anggur**:
 - Memanggil metode **gerak()** untuk menggerakkan objek **Anggur** ke atas.
 - Jika objek **Anggur** bertabrakan dengan objek **Kunai** (melalui metode **belah()**), maka menambahkan objek baru **AnggurBelah** ke dunia pada posisi yang sama dengan objek **Anggur**.
 - Memanggil metode **hilang()**.
- Metode **hilang()** memeriksa apakah objek **Anggur** telah mencapai ujung atas dunia permainan (melalui metode **ujung()**).
- Jika iya, menghapus objek **Anggur** dari dunia menggunakan **getWorld().removeObject(this)**.

Anggur Belah

- **AnggurBelah** adalah kelas turunan yang mengextends kelas **Buah**, menunjukkan bahwa **AnggurBelah** mewarisi sifat dan perilaku umum dari kelas **Buah**.
- Terdapat metode **act()**, yang merupakan bagian dari siklus hidup objek dalam Greenfoot. Saat tombol "Act" atau "Run" ditekan, metode ini akan dipanggil.
- Metode ini melakukan beberapa tindakan untuk objek **AnggurBelah**:
 - Memanggil metode **gerakBelah()** untuk menggerakkan objek **AnggurBelah** ke bawah.
 - Jika objek **AnggurBelah** mencapai ujung bawah dunia (melalui metode **ujungBawah()**), maka menghapus objek **AnggurBelah** dari dunia menggunakan **getWorld().removeObject(this)**.
- Metode **gerakBelah()** menggerakkan objek **AnggurBelah** ke bawah (koordinat Y bertambah) sejauh 2 piksel pada setiap pemanggilan.

- Metode **ujungBawah()** memeriksa apakah objek **AnggurBelah** telah mencapai ujung bawah dunia permainan (lebih dari 375 piksel).
- Jika iya, mengembalikan **true**, menunjukkan bahwa objek **AnggurBelah** berada pada ujung bawah dunia.

Note: Penjelasan diatas juga termasuk kedalam class turunan dari apel,apricot,jeruk,lemon,mangga,nanas,peach,pear,dan semangka cuman yang membedakan hanya nama subclass nya saja.

Codingan:

Class Counter

import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

```
public class Counter extends Actor
{
    private static final Color transparent = new Color(0,0,0,0);
    private GreenfootImage background;
    private int value;
    private int target;

    public Counter()
    {
        background = getImage();
        value = 0;
        target = 0;
        updateImage();
    }

    public void act()
    {
        if (value < target) {
            value++;
            updateImage();
        }
        else if (value > target) {
            value--;
            updateImage();
        }
    }

    public void add(int score)
    {
```

```

        target += score;
    }

    public int getValue()
    {
        return value;
    }

    public void setValue(int newValue)
    {
        target = newValue;
        value = newValue;
        updateImage();
    }

    private void updateImage()
    {
        GreenfootImage image = new GreenfootImage(background);
        GreenfootImage text = new GreenfootImage("" + value, 22, Color.GRAY, transparent);
        image.drawImage(text, (image.getWidth()-text.getWidth())/2,
                        (image.getHeight()-text.getHeight())/2);
        setImage(image);
    }
}

```

Penjelasan:

- **Counter** adalah kelas yang bertanggung jawab untuk menampilkan dan mengelola nilai skor atau counter dalam permainan.
- **background**: Menyimpan gambar latar belakang dari objek **Counter**.
- **value**: Menyimpan nilai skor saat ini.
- **target**: Menyimpan nilai target yang akan dicapai oleh skor.
- **transparent**: Warna transparan yang digunakan dalam pembuatan gambar teks.
- Terdapat konstruktor **Counter** yang menginisialisasi gambar latar belakang, nilai skor, dan target skor awal. Konstruktor ini juga memanggil metode **updateImage()** untuk memperbarui tampilan gambar.
- Metode **act()** dipanggil setiap frame dan memperbarui nilai skor pada gambar sesuai dengan target skor. Jika nilai skor kurang dari target, nilai skor akan bertambah; jika lebih, nilai skor akan berkurang. Metode juga memanggil **updateImage()** setiap kali nilai skor diubah.

- Metode **add()** digunakan untuk menambahkan nilai tertentu ke target skor.
- Metode **getValue()** mengembalikan nilai skor saat ini.
- Metode **setValue()** mengatur nilai skor dan target skor menjadi nilai baru yang diberikan dan memanggil **updateImage()**.
- Metode **updateImage()** membuat gambar baru dengan latar belakang, menambahkan teks nilai skor pada tengah gambar, dan mengatur gambar sebagai gambar objek **Counter**.

Codingan Class

Kunai

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
```

```
public class Kunai extends Actor
```

```
{
```

```
    private Counter counter;
```

```
    public int jumlahNyawa = 3;
```

```
    public boolean nyentuhBom = false;
```

```
    public Kunai(Counter pointCounter)
```

```
    {
```

```
        counter = pointCounter;
```

```
    }
```

```
    public void act()
```

```
    {
```

```
        // Add your action code here.
```

```
        gerak();
```

```
        addCounter(Pear.class);
```

```
        addCounter(Peach.class);
```



```

        addCounter(Nanas.class);
        addCounter(Jeruk.class);
        addCounter(Apel.class);
        addCounter(Semangka.class);
        addCounter(Anggur.class);
        addCounter(Lemon.class);
        addCounter(Mangga.class);
        addCounter(Apricot.class);
        bom();
    }

    public void gerak()
    {
        if(Greenfoot.mouseMoved(null))
        {
            MouseInfo mouse = Greenfoot.getMouseInfo();
            setLocation(mouse.getX(), mouse.getY());
            turn(15);
        }
    }

    public void addCounter(Class cls)
    {
        Actor actor;

        actor = getOneObjectAtOffset(0,0,cls);

        if(actor != null)
        {

```

```

        Greenfoot.playSound("sword.mp3");
        getWorld().removeObject(actor);
        counter.add(10);
    }
}

public void bom()
{
    Actor bom;
    bom = getOneObjectAtOffset(0, 0, Bom.class);
    if(bom != null)
    {
        Greenfoot.playSound("bom.wav");
        jumlahNyawa--;
        getWorld().removeObject(bom);
        if(jumlahNyawa == 2)
        {
            getWorld().addObject(new Nyawa2(), 525, 24);
        }
        if(jumlahNyawa == 1)
        {
            getWorld().addObject(new Nyawa1(), 525, 24);
        }
        if(jumlahNyawa == 0)
        {
            Greenfoot.playSound("smartfren.wav");
            getWorld().addObject(new GameOver(), 525, 24);
        }
    }
}

```

```

        getWorld().addObject(new ScoreBoard(counter.getValue()), 300, 200);
        Greenfoot.stop();
    }
}
}

public int kurangiNyawa()
{
    return jumlahNyawa;
}
}

```

Penjelasan:

- **Kunai** adalah kelas yang merepresentasikan objek kunai dalam permainan. Objek kunai ini dapat digerakkan oleh pemain menggunakan mouse.
- **counter**: Merupakan objek dari kelas **Counter** yang digunakan untuk melacak nilai skor.
- **jumlahNyawa**: Menyimpan jumlah nyawa yang dimiliki pemain.
- **nyentuhBom**: Menyimpan status apakah kunai menyentuh bom atau tidak.
- Terdapat konstruktor **Kunai** yang menerima parameter objek **Counter** untuk menginisialisasi objek **counter**.
- Terdapat metode **act()** yang dipanggil setiap frame dan melakukan beberapa tindakan:
 - Memanggil metode **gerak()** untuk menggerakkan kunai sesuai dengan pergerakan mouse.
 - Memanggil metode **addCounter()** untuk menambah nilai skor jika kunai menyentuh objek buah tertentu.
 - Memanggil metode **bom()** untuk menangani interaksi dengan bom.
- Metode **gerak()** menggerakkan kunai sesuai dengan pergerakan mouse. Jika mouse bergerak, posisi kunai diatur sesuai dengan posisi mouse.

- Metode **addCounter()** memeriksa apakah kunai menyentuh objek buah tertentu (kelas yang diberikan). Jika iya, nilai skor ditambahkan, objek buah dihapus, dan suara "sword.mp3" diputar.
- Metode **bom()** memeriksa apakah kunai menyentuh objek bom. Jika iya, jumlah nyawa dikurangi, suara "bom.wav" diputar, dan objek bom dihapus. Jika jumlah nyawa mencapai 0, suara "smartfren.wav" diputar, objek GameOver ditambahkan ke dunia, serta objek ScoreBoard dengan nilai skor, dan permainan dihentikan.
- Metode **kurangiNyawa()** mengembalikan nilai jumlah nyawa.

Codingan Class

Score Board

```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
```

```
import java.util.Calendar;
```

```
public class ScoreBoard extends Actor
```

```
{
```

```
    public static final float FONT_SIZE = 48.0f;
```

```
    public static final int WIDTH = 400;
```

```
    public static final int HEIGHT = 300;
```

```
    public ScoreBoard()
```

```
    {
```

```
        this(100);
```

```
    }
```

```
    public ScoreBoard(int score)
```

```
    {
```

```

        makeImage("Game Over", "Score : ", score);
    }

    private void makeImage(String title, String prefix, int score)
    {
        GreenfootImage image = new GreenfootImage(WIDTH, HEIGHT);

        image.setColor(new Color(0, 0, 0, 160));
        image.fillRect(0, 0, WIDTH, HEIGHT);
        image.setColor(new Color(255, 255, 255, 100));
        Font font = image.getFont();
        font = font.deriveFont(FONT_SIZE);
        image.setFont(font);
        image.setColor(Color.WHITE);
        image.drawString(title, 60, 100);
        image.drawString(prefix + score, 60, 200);
        setImage(image);
    }
}

```

Penjelasan:

- **ScoreBoard** adalah kelas yang bertanggung jawab untuk menampilkan papan skor atau informasi tentang akhir permainan, termasuk judul ("Game Over") dan skor pemain.
- **FONT_SIZE**: Konstanta yang menyimpan ukuran font yang digunakan dalam papan skor.
- **WIDTH** dan **HEIGHT**: Konstanta yang menyimpan lebar dan tinggi dari gambar papan skor.
- Terdapat dua konstruktor:

- Konstruktor pertama (**ScoreBoard()**) digunakan untuk membuat papan skor dengan nilai skor default (100).
- Konstruktor kedua (**ScoreBoard(int score)**) menerima parameter nilai skor dan menggunakan metode **makeImage()** untuk membuat gambar papan skor dengan judul "Game Over" dan nilai skor.
- Metode ini digunakan untuk membuat gambar papan skor dengan judul, prefix, dan nilai skor yang diberikan.
- Membuat objek **GreenfootImage** dengan lebar dan tinggi tertentu.
- Mengisi latar belakang gambar dengan warna hitam transparan.
- Mengatur ukuran dan warna font, serta menambahkan judul dan nilai skor ke gambar.
- Mengatur gambar yang dihasilkan sebagai gambar objek **ScoreBoard**.

Penerapan Inheritance (Pewarisan), Polymorphism (Polimorfisme), Overriding (Penggantian), Enkapsulasi, Interaksi Antar Objek, Kelas dan Objek:

- Inheritance (Pewarisan):

```
public class Anggur extends Buah { /* ... */ }
```

```
public class AnggurBelah extends Buah { /* ... */ }
```

Buah, Anggur, dan AnggurBelah adalah contoh penggunaan inheritance. Anggur dan AnggurBelah adalah subclass dari Buah. Mereka mewarisi sifat dan perilaku dari kelas Buah.

- Polymorphism (Polimorfisme):

```
public class Anggur extends Buah { /* ... */ }
```

Di dalam metode act() kelas Dunia, objek Anggur dan AnggurBelah dapat dipanggil dan digunakan bersama-sama, menunjukkan polimorfisme.

- Overriding (Penggantian):

```
public class Anggur extends Buah {
    public void act() {
        // ...
    }
    // ...
}
```

}

Metode act() di kelas Anggur meng-override metode act() di kelas Buah.

- Overloading (Pemuatan Lebih):

Overloading terjadi ketika terdapat beberapa metode dengan nama yang sama tetapi memiliki parameter yang berbeda,

- Enkapsulasi

Counter.java

Konsep enkapsulasi terlihat di kelas Counter. Variabel value, target, dan metode setValue() digunakan untuk mengelola dan mengakses nilai secara terkontrol.

Kunai.java

Variabel counter di kelas Kunai dapat dianggap sebagai implementasi sederhana dari enkapsulasi, karena hanya dapat diakses melalui metode-metode yang disediakan.

- Interaksi Antar Objek:

Dunia.java

Dalam metode act(), terdapat pemanggilan metode-metode seperti anggur(), apel(), dsb., yang menunjukkan interaksi antar objek di dunia permainan.

Kunai.java

Pada metode addCounter(), ada interaksi antara objek Kunai dan objek buah (Pear, Peach, dll.), di mana nilai skor bertambah ketika kunai menyentuh buah.

```
public void addCounter(Class cls) {  
    //  
}
```

- Kelas dan Objek

Semua Kelas:

Setiap kelas yang didefinisikan (Buah, Anggur, AnggurBelah, Kunai, Counter, ScoreBoard, dll.) adalah implementasi dari konsep kelas dan objek.

Contoh Penggunaan Objek:

Dalam metode prepare() di kelas Dunia, objek Counter dan Kunai dibuat dan diatur sebagai bagian dari inisialisasi dunia permainan.

```
private void prepare() {  
    Counter skor = new Counter();  
    addObject(skor, 55, 23);  
    Kunai kunai = new Kunai(skor);  
    addObject(kunai, 300, 200);  
}
```

Contoh Penggunaan Objek Lain:

Dalam metode act() di kelas Kunai, terdapat pemanggilan metode untuk menambah nilai skor dan menangani bom, menunjukkan interaksi antar objek.

```
public void act() {  
    // ...  
    addCounter(Pear.class);  
    addCounter(Peach.class);  
    addCounter(Nanas.class);  
    addCounter(Jeruk.class);  
    addCounter(Apel.class);  
    addCounter(Semangka.class);  
    addCounter(Anggur.class);  
    addCounter(Lemon.class);  
    addCounter(Mangga.class);  
    addCounter(Apricot.class);  
    bom();  
}
```


