

# Projek UAS Pemrograman Berorientasi Objek memakai Greenfoot

## Pembuatan Game “Car“

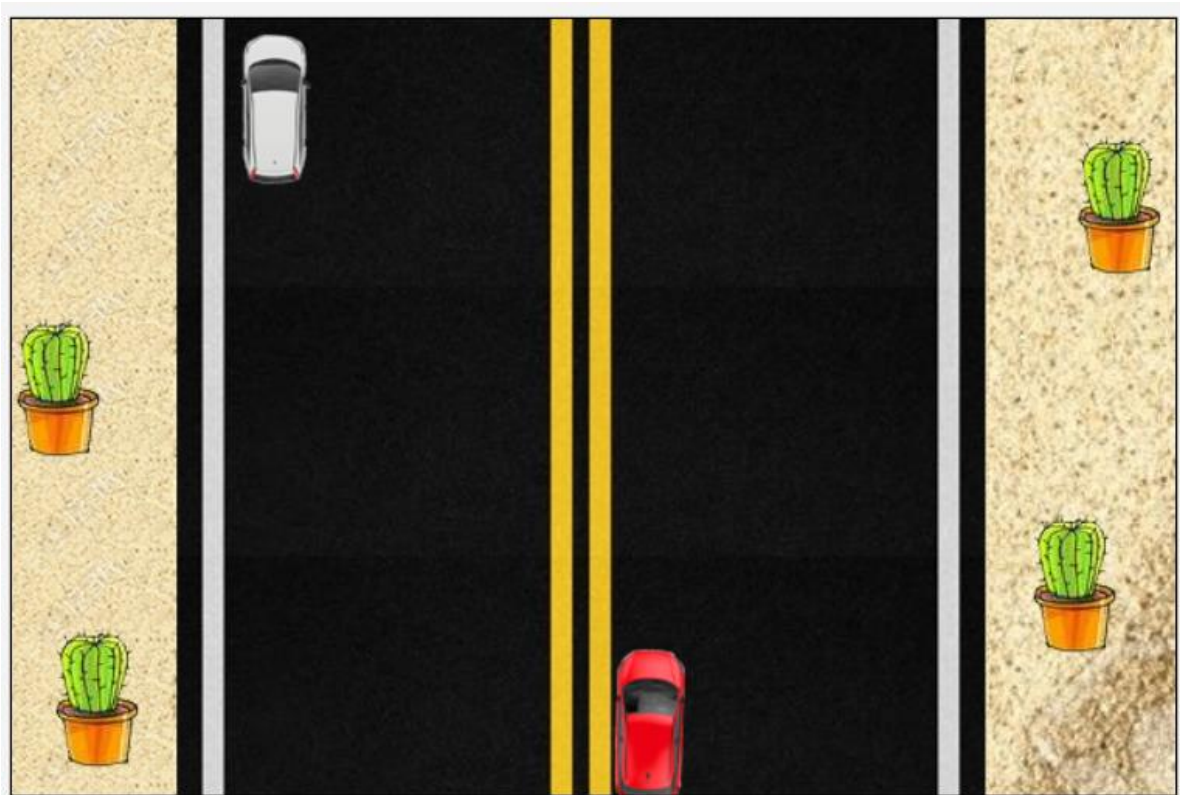
### Kelompok 20:

1. Alwansyah 2211102441026
2. Nabila Salsabila Yusri 2211102441038
3. Nurjanna 2211102441214
4. Selma Alfionita 2211102441208

### Skenario :

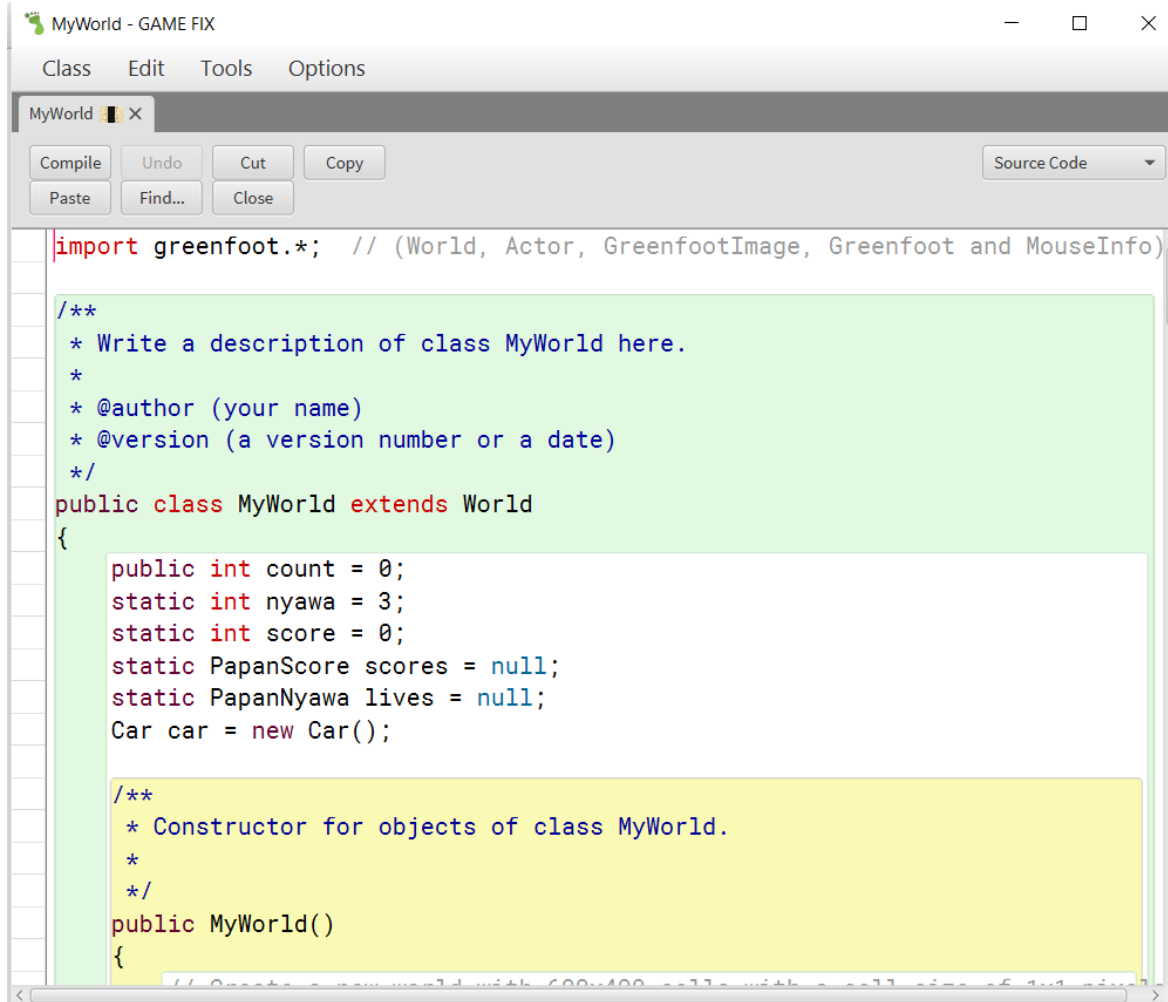
1. Mobil dapat bergerak ke kanan,kiri,maju, dan kebelakang.
2. Mobil menghindari semua rintangan(mobil lain).
3. Jika mobil menabrak mobil lain atau rintangan maka nyawanya akan berkurang sampai nyawanya habis dan kemudian kalah.
4. Rintangan yang disediakan berupa mobil lain dengan warna yang berbeda maupun yang sama
5. Setelah mendapat point 100 dengan cara mengambil bunga atau sejenis tumbuhan maka akan menang dan permainan akan berhenti.

### SS Template:



## Penjelasan Fungsi-Fungsi yang digunakan:

### MyWorld:

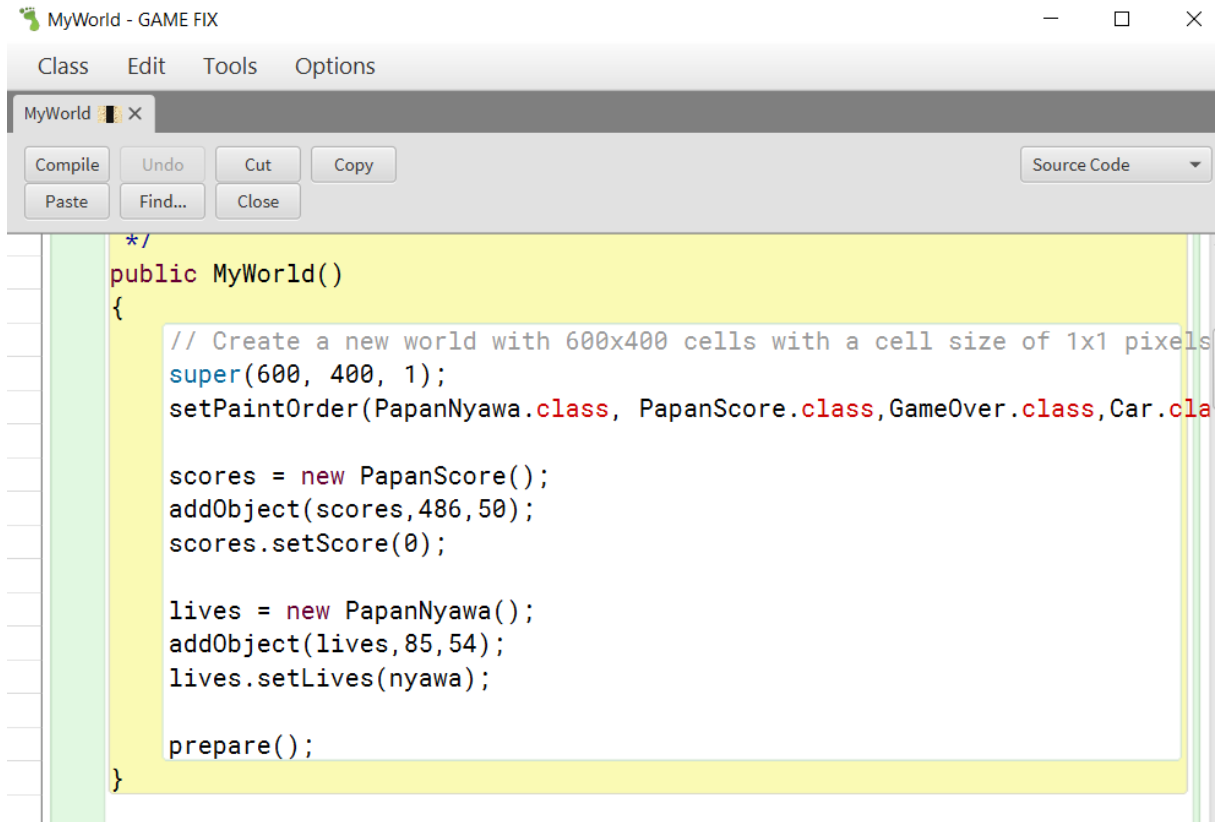


```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/**
 * Write a description of class MyWorld here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class MyWorld extends World
{
    public int count = 0;
    static int nyawa = 3;
    static int score = 0;
    static PapanScore scores = null;
    static PapanNyawa lives = null;
    Car car = new Car();

    /**
     * Constructor for objects of class MyWorld.
     *
     */
    public MyWorld()
    {
        // Create a new world with 600x400 cells with a cell size of 1x1 pixels
    }
}
```

- Untuk bagian kode diatas maksudnya yaitu untuk menginisialisasikan atau menentukan terlebih dahulu nilai dari nyawa,count,score,PapanNyawa, dan juga PapanScore.



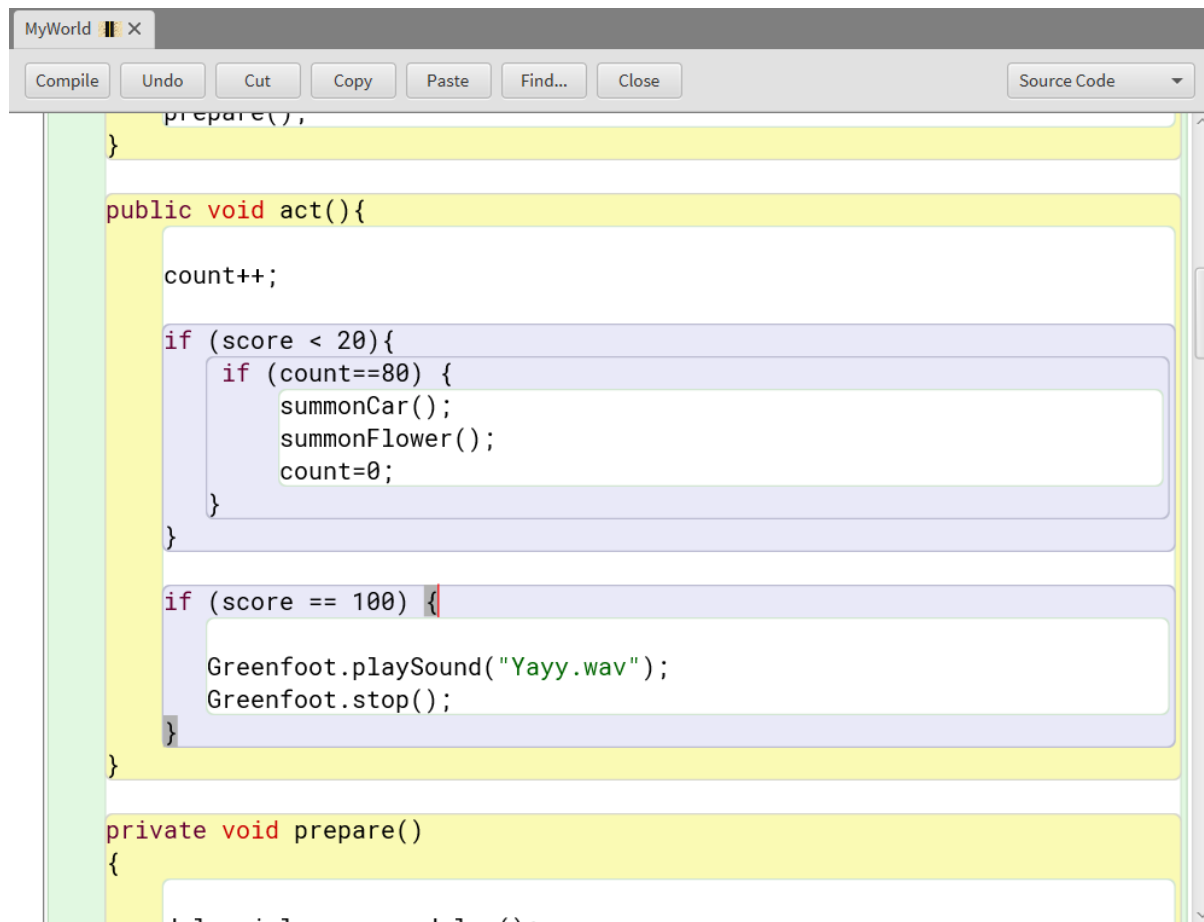
```
public MyWorld()
{
    // Create a new world with 600x400 cells with a cell size of 1x1 pixels
    super(600, 400, 1);
    setPaintOrder(PapanNyawa.class, PapanScore.class,GameOver.class,Car.class);

    scores = new PapanScore();
    addObject(scores,486,50);
    scores.setScore(0);

    lives = new PapanNyawa();
    addObject(lives,85,54);
    lives.setLives(nyawa);

    prepare();
}
```

- Kode diatas maksudnya yaitu pertama setPaintOrder mendefinisikan semua class yang ada, kemudian yang baris selanjutnya yaitu untuk menempatkan letak dari PapanScore.class dan juga menentukan nilai scorenya yaitu '0' dan selanjutnya baris yang terakhir sama seperti baris yang sebelumnya yaitu menentukan letak dari PapanNyawa.class dan juga menentukan nilai nyawa yaitu 3 yang didapat dari yang sudah didefinisikan pertama tadi yaitu "static int nyawa = 3;".



```
MyWorld [X]
Compile Undo Cut Copy Paste Find... Close Source Code
prepare(),
}

public void act(){

    count++;

    if (score < 20){
        if (count==80) {
            summonCar();
            summonFlower();
            count=0;
        }
    }

    if (score == 100) {
        Greenfoot.playSound("Yayy.wav");
        Greenfoot.stop();
    }

}

private void prepare()
{
    // ...
}
```

- Pada kode diatas maksudnya yaitu untuk melakukan sebuah aksi yang pertama yaitu count++ yang artinya yaitu menambahkan nilai dari hasil count terakhir,kemudian ada 'if' yang terakhir yang bertujuan untuk ketika score sudah mencapai nilai 100 maka akan melakukan atau memutar sound yayy.wav dan juga greenfoot akan berhenti.

- Kode diatas digunakan untuk meletakkan semua objek yang ada dan digunakan.

```

        addObject(gameOver, 300, 100);
        removeObject(gameOver);
    }

    public void summonCar(){
        addObject(new Other_Car(), Greenfoot.getRandomNumber(380)+100, 0);
    }

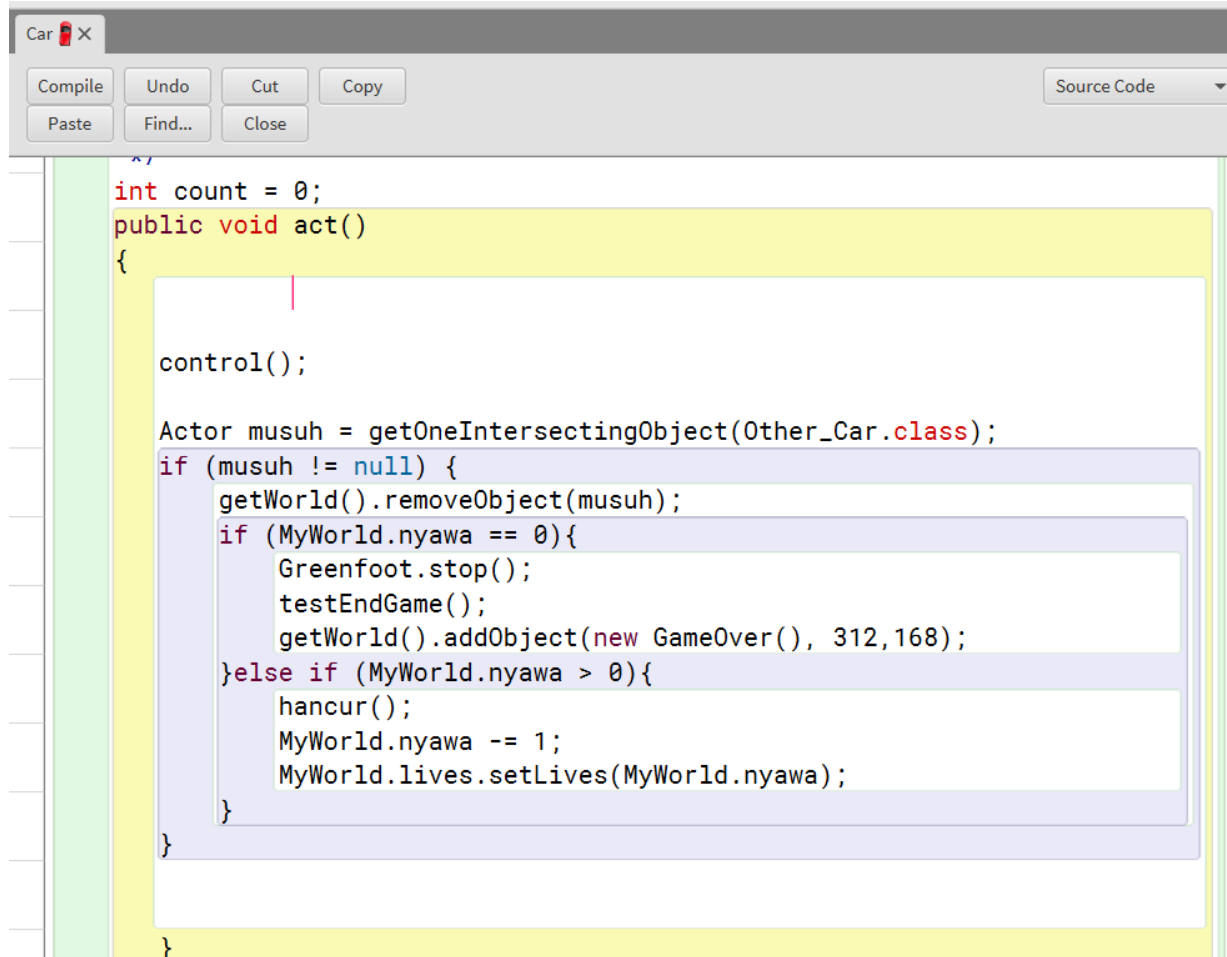
    public void summonFlower(){
        addObject(new Jamur(), Greenfoot.getRandomNumber(380)+100, 0);
    }

    public void die(){
        addObject(car, 300, 363);
    }
}

```

- Public void summonCar() fungsinya yaitu untuk membuat objek Other.car bergerak secara random
- Public void summonFlower() fungsinya digunakan untuk membuat Jamur bergerak secara random
- Public void die() fungsinya digunakan untuk menambahkan object car diposisi 300X dan 363Y

## Car:

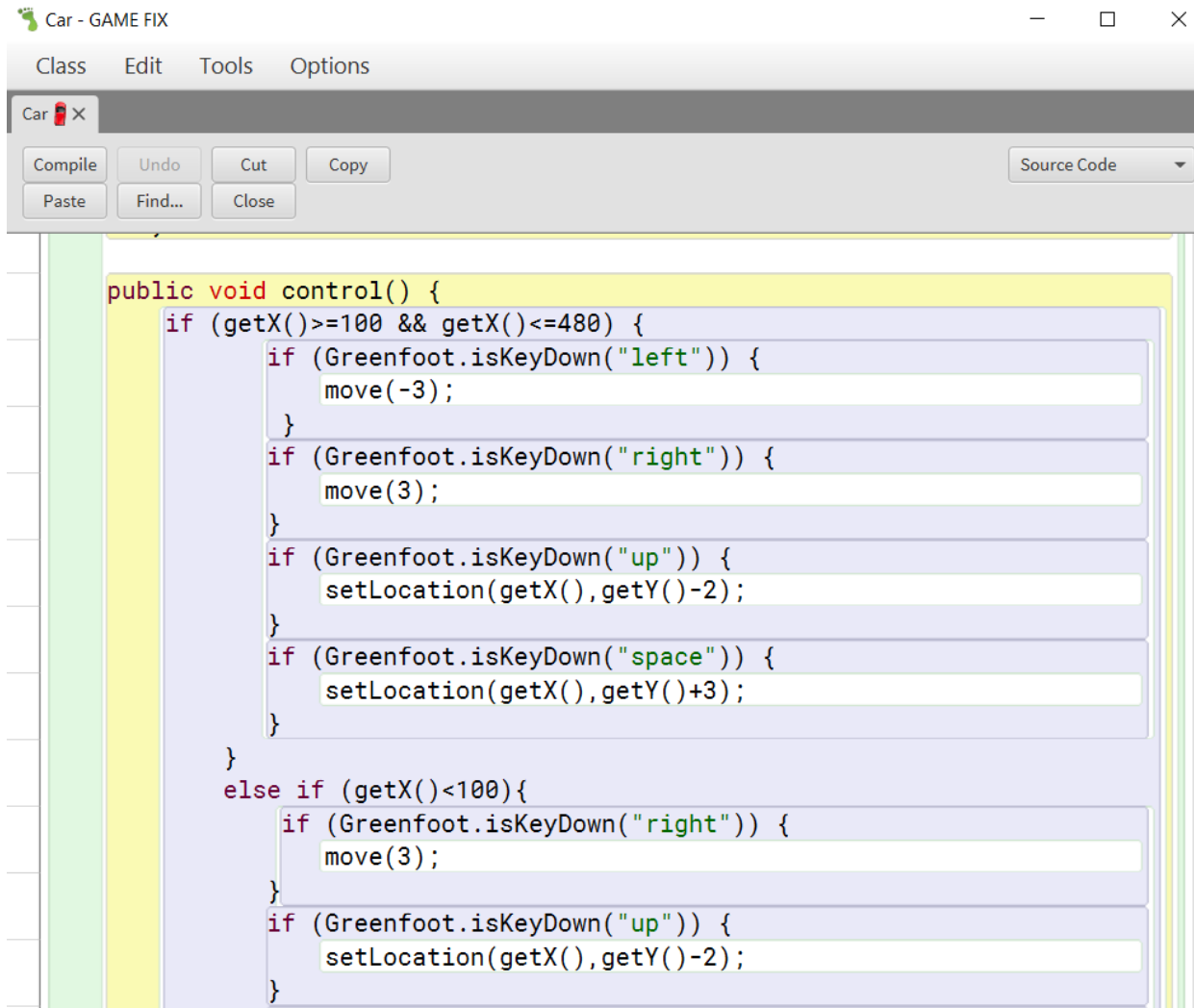


```
int count = 0;
public void act()
{
    control();

    Actor musuh = getOneIntersectingObject(Other_Car.class);
    if (musuh != null) {
        getWorld().removeObject(musuh);
        if (MyWorld.nyawa == 0){
            Greenfoot.stop();
            testEndGame();
            getWorld().addObject(new GameOver(), 312,168);
        }else if (MyWorld.nyawa > 0){
            hancur();
            MyWorld.nyawa -= 1;
            MyWorld.lives.setLives(MyWorld.nyawa);
        }
    }
}
```

Kode diatas artinya yaitu :

- Yang pertama `int count = 0` adalah digunakan untuk mendefinisikan count yang dimulai dari 0
- Kemudian `control()` yaitu mendefinisikan terlebih dahulu baru setelah itu dieksekusi pada baris selanjutnya atau dibawahnya.
- Kemudian ada sebuah kondisi yang dimana ketika nyawa bernilai 0 maka game akan berhenti dan akan memunculkan kata gameover.
- Kemudian jika nyawa masih lebih dari 0 maka nyawa akan terus berkurang ketika menabrak rintangan.



The screenshot shows a Java IDE window titled "Car - GAME FIX". The window has a menu bar with "Class", "Edit", "Tools", and "Options". Below the menu bar is a toolbar with buttons for "Compile", "Undo", "Cut", "Copy", "Paste", "Find...", and "Close". A "Source Code" dropdown menu is also present. The code editor displays the following Java code:

```
public void control() {  
    if (getX() >= 100 && getX() <= 480) {  
        if (Greenfoot.isKeyDown("left")) {  
            move(-3);  
        }  
        if (Greenfoot.isKeyDown("right")) {  
            move(3);  
        }  
        if (Greenfoot.isKeyDown("up")) {  
            setLocation(getX(), getY() - 2);  
        }  
        if (Greenfoot.isKeyDown("space")) {  
            setLocation(getX(), getY() + 3);  
        }  
    }  
    else if (getX() < 100) {  
        if (Greenfoot.isKeyDown("right")) {  
            move(3);  
        }  
        if (Greenfoot.isKeyDown("up")) {  
            setLocation(getX(), getY() - 2);  
        }  
    }  
}
```

- Kode diatas digunakan untuk mengontrol gerak dari mobilnya yang bisa bergerak ke kiri,kanan,bawah dengan menekan space dank e atas.



```

        if (Greenfoot.isKeyDown("space")) {
            setLocation(getX(),getY()+3);
        }
    }
    else {
        if (Greenfoot.isKeyDown("left")) {
            move(-3);
        }
        if (Greenfoot.isKeyDown("up")) {
            setLocation(getX(),getY()-2);
        }
        if (Greenfoot.isKeyDown("space")) {
            setLocation(getX(),getY()+3);
        }
    }
}

private void hancur(){
    getWorld().addObject(new Kuat(), getX(), getY());
    getWorld().removeObject(this);
}

```

- Kode diatas yang paling bawah digunakan untuk melakukan aksi yaitu ketika mobil menabrak rintangan maka mobil akan tetep berjalan pada posisi X dan Y ketika dia menabrak rintangan tersebut.

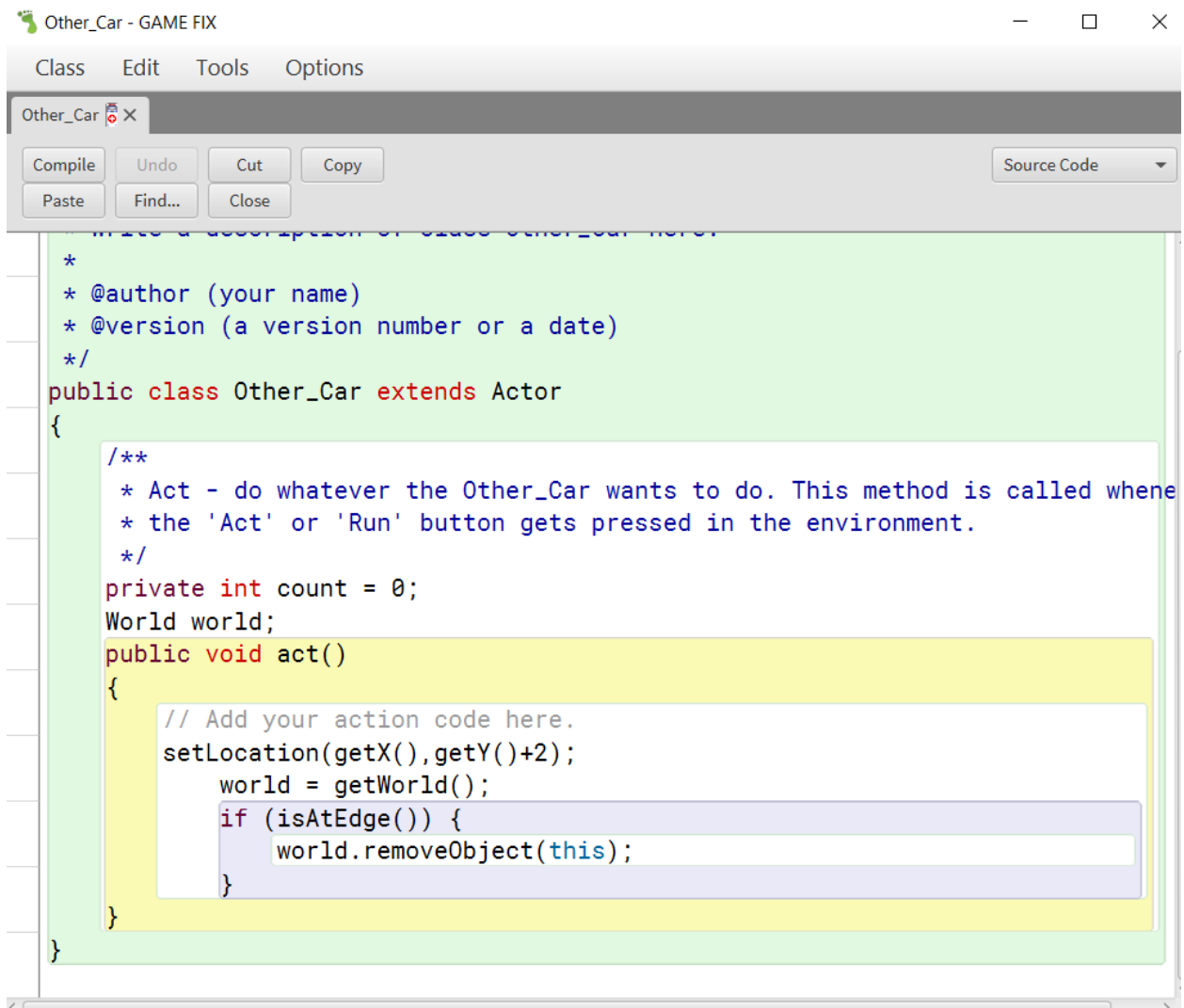
```

public void testEndGame() {
    if (MyWorld.nyawa <= 0) {
        setImage(gameoverimage);
        setLocation(312,368);
        Greenfoot.stop();
    }
}
}

```

- Kode diatas digunakan untuk menghentikan game ketika nyawa kurang dari sama dengan 0 dan mengentikan greennfoot dan juga memunculkan gambar image gameover yang sudah didefinisikan diatasnya yang ditempatkan pada 312,368.

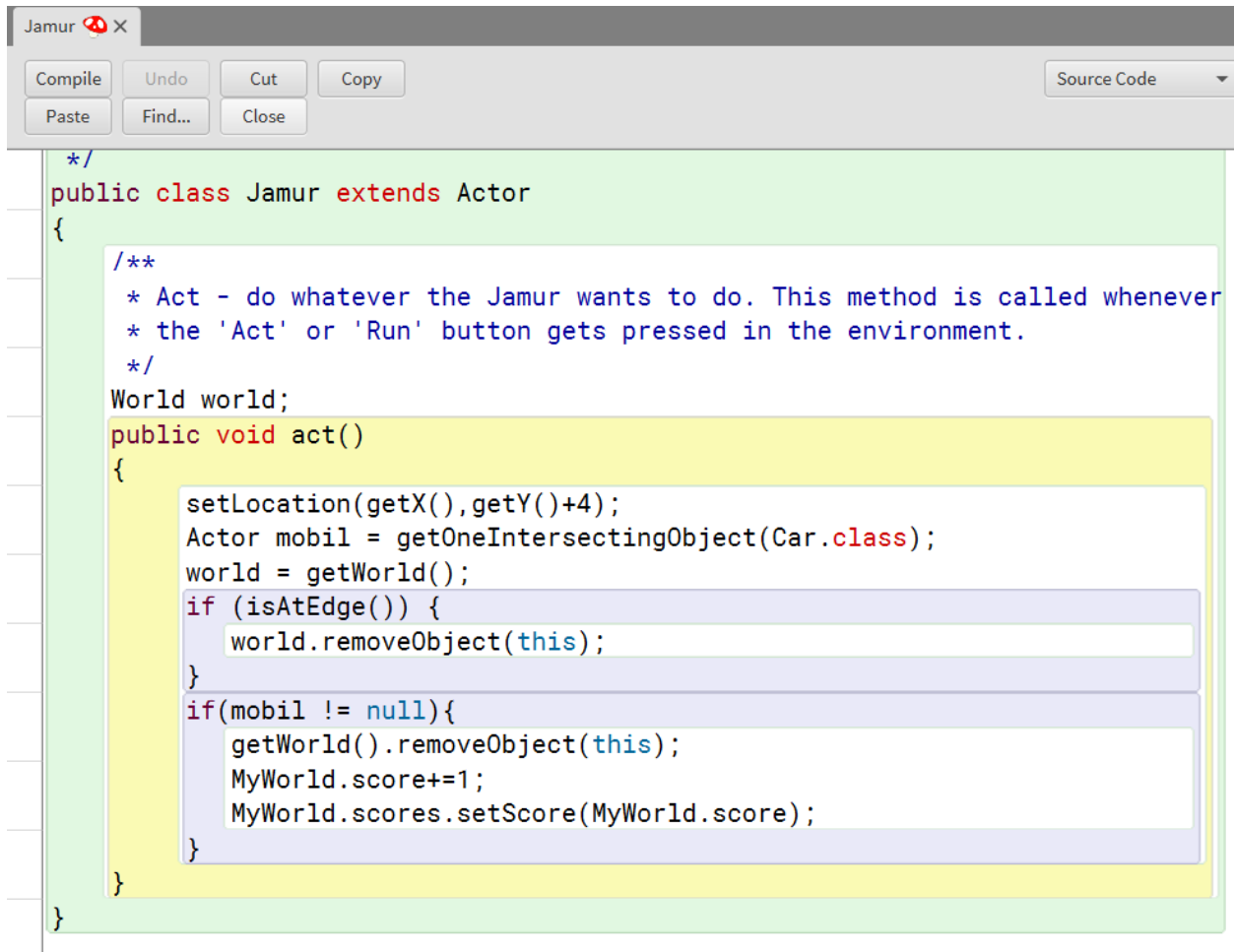
## Other\_Car:



```
/*
 * @author (your name)
 * @version (a version number or a date)
 */
public class Other_Car extends Actor
{
    /**
     * Act - do whatever the Other_Car wants to do. This method is called when
     * the 'Act' or 'Run' button gets pressed in the environment.
     */
    private int count = 0;
    World world;
    public void act()
    {
        // Add your action code here.
        setLocation(getX(),getY()+2);
        world = getWorld();
        if (isAtEdge()) {
            world.removeObject(this);
        }
    }
}
```

- Kode diatas digunakan untuk meletakkan lokasi mobil untuk pertama kali dan ketika program dirun maka akan menghilangkan object yang ada di world.

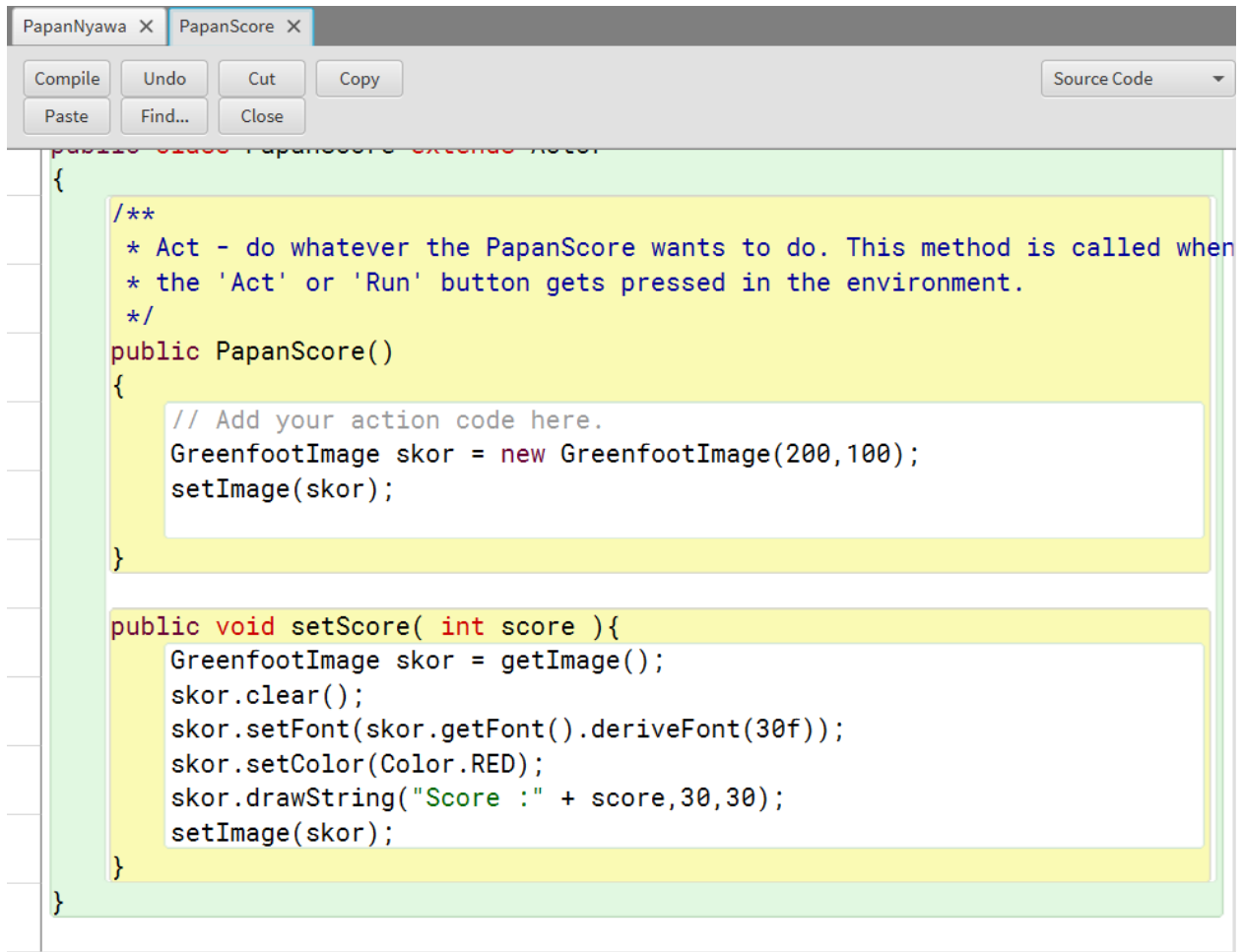
## Jamur:



```
*/
public class Jamur extends Actor
{
    /**
     * Act - do whatever the Jamur wants to do. This method is called whenever
     * the 'Act' or 'Run' button gets pressed in the environment.
     */
    World world;
    public void act()
    {
        setLocation(getX(),getY()+4);
        Actor mobil = getOneIntersectingObject(Car.class);
        world = getWorld();
        if (isAtEdge()) {
            world.removeObject(this);
        }
        if(mobil != null){
            getWorld().removeObject(this);
            MyWorld.score+=1;
            MyWorld.scores.setScore(MyWorld.score);
        }
    }
}
```

- Kode diatas digunakan untuk meletakkan lokasi mobil ketika mobil bersentuhan ke object lain yaitu dari mobil.class maka akan menghilang.
- Dan if yang kedua yaitu maksudnya digunakan untuk menghilangkan mobil dan menambah nilai dengan nilai 1 dari score yang ada di MyWorld dan juga scorenya dimulai dari score yang sudah ditentukan di MyWorld.

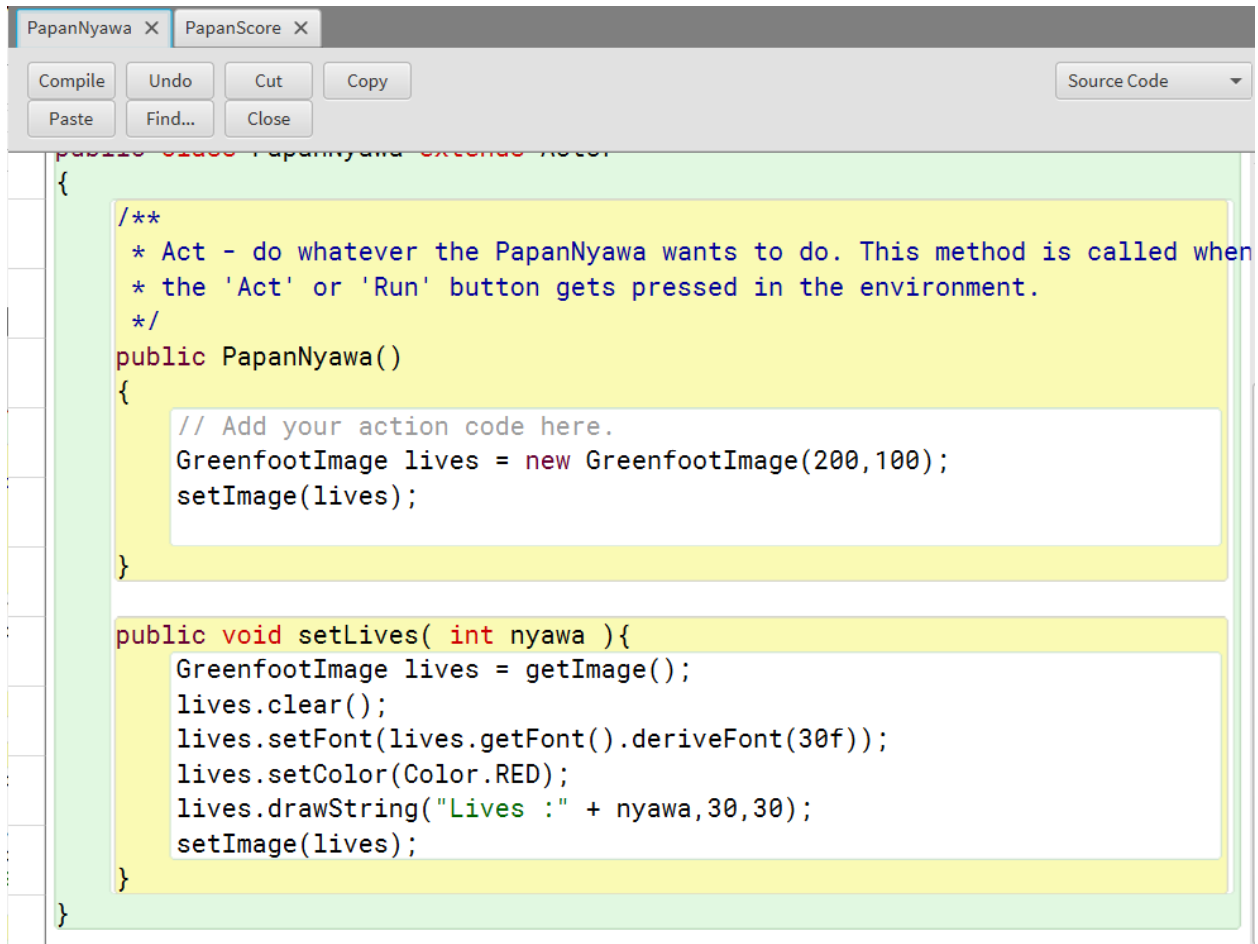
## PapanScore:



```
public class PapanScore extends Node {  
    /**  
     * Act - do whatever the PapanScore wants to do. This method is called when  
     * the 'Act' or 'Run' button gets pressed in the environment.  
     */  
    public PapanScore()  
    {  
        // Add your action code here.  
        GreenfootImage skor = new GreenfootImage(200,100);  
        setImage(skor);  
    }  
  
    public void setScore( int score ){  
        GreenfootImage skor = getImage();  
        skor.clear();  
        skor.setFont(skor.getFont().deriveFont(30f));  
        skor.setColor(Color.RED);  
        skor.drawString("Score :" + score,30,30);  
        setImage(skor);  
    }  
}
```

- Kode diatas maksudnya yaitu untuk yang kedua public void setScore() digunakan untuk membuat tulisan Score dengan warna Merah dengan ukuran 30.
- Untuk kode yang pertama digunakan untuk memunculkan tulisan dari nyawa yang berjumlah 3.

## PapanNyawa:

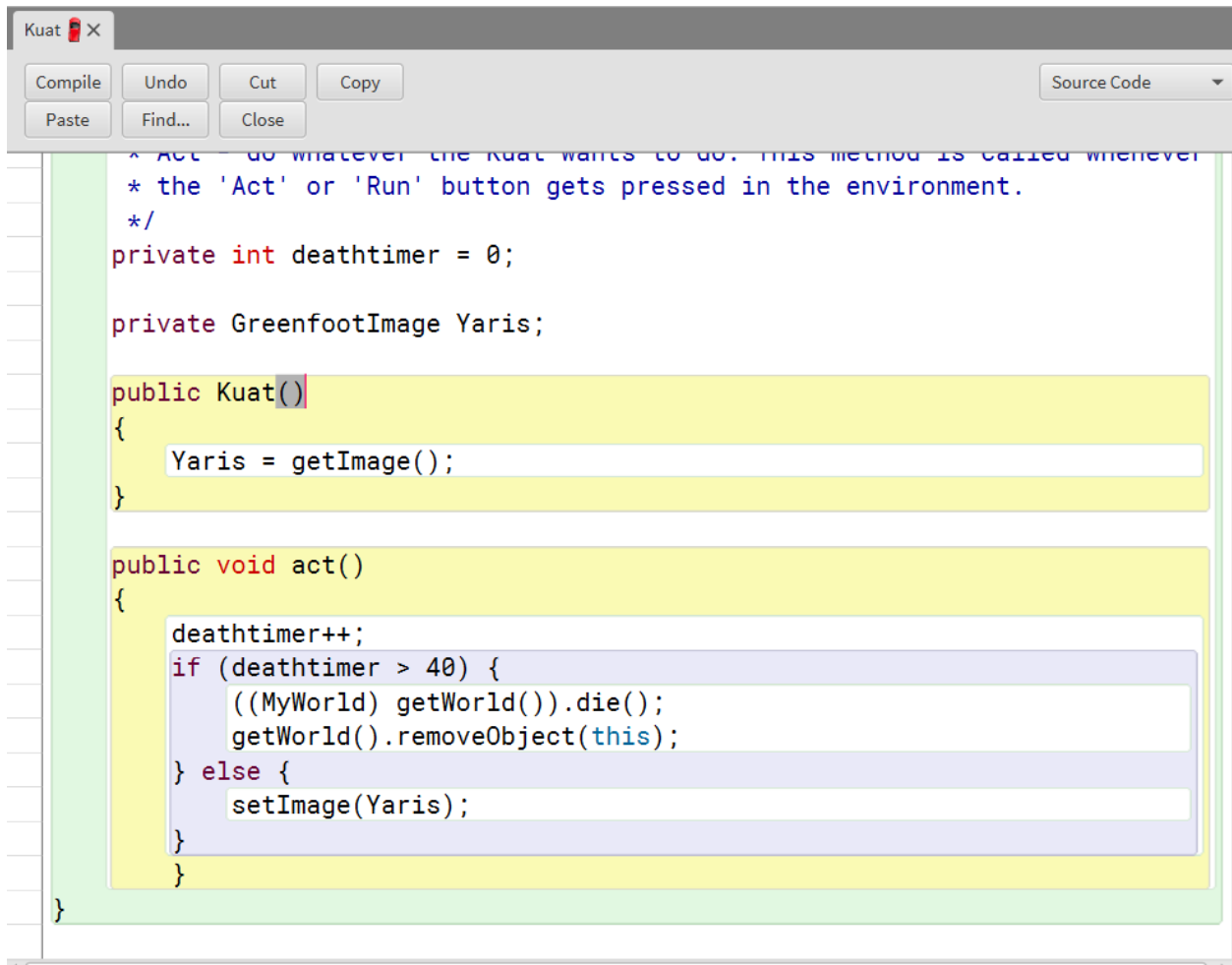


The screenshot shows a code editor window with two tabs: 'PapanNyawa' and 'PapanScore'. The 'PapanNyawa' tab is active, displaying Java code. The code includes a class definition for 'PapanNyawa' with a constructor and a 'setLives' method. The constructor initializes a 'GreenfootImage' object named 'lives' with dimensions (200, 100) and calls 'setImage(lives)'. The 'setLives' method takes an 'int' parameter 'nyawa', gets the current 'lives' image, clears it, sets the font to size 30, sets the color to red, and draws the string 'Lives : ' followed by 'nyawa' at position (30, 30). The code is syntax-highlighted with colors: blue for keywords, red for 'public', green for 'void', and black for other identifiers and literals.

```
public class PapanNyawa extends Actor {  
    /**  
     * Act - do whatever the PapanNyawa wants to do. This method is called when  
     * the 'Act' or 'Run' button gets pressed in the environment.  
     */  
    public PapanNyawa()  
    {  
        // Add your action code here.  
        GreenfootImage lives = new GreenfootImage(200,100);  
        setImage(lives);  
    }  
  
    public void setLives( int nyawa ){  
        GreenfootImage lives = getImage();  
        lives.clear();  
        lives.setFont(lives.getFont().deriveFont(30f));  
        lives.setColor(Color.RED);  
        lives.drawString("Lives :" + nyawa,30,30);  
        setImage(lives);  
    }  
}
```

- Kode diatas yang pertama digunakan untuk menampilkan tulisan score ke layar greenfoot
- Untuk yang kedua digunakan untuk membentuk tulisan Lives dengan warna Merah dengan ukuran font 30.

## Kuat:



```
* Act - do whatever the Kuat wants to do. This method is called whenever
* the 'Act' or 'Run' button gets pressed in the environment.
*/
private int deathtimer = 0;

private GreenfootImage Yaris;

public Kuat()
{
    Yaris = getImage();
}

public void act()
{
    deathtimer++;
    if (deathtimer > 40) {
        ((MyWorld) getWorld()).die();
        getWorld().removeObject(this);
    } else {
        setImage(Yaris);
    }
}
```

- Kode diatas digunakan untuk menginisialisasikan deathtimer dengan nilai pertama yaitu 0 dan juga menginisialisasi gambar Yaris.
- Kemudian yang public Kuat digunakan untuk memunculkan gambar berupa mobil yaris.
- Kemudian yang public void act didalamnya ada deattime++ yang fungsinya untuk menambah 1 untuk setiap waktunya kemudian ada pengkondisian if jika waktunya lebih dari 40 maka akan menghilangkan object mobil yarisnya dan ketika salah atau nilainya kurang dari 40 maka akan tetap memunculkan mobil yarisnya.