



# **PROJECT UAS PEMROGRAMAN BERORIENTASI OBJEK (PBO)**

**INF2093/2103**

Kelompok 31

Disusun Oleh :

Frandi Ferdyan / 2211102441011

Ryfyky Putra Ramadhan / 2211102441195

Vicky Drajat Fatahilallah / 2211102441015

Teknik Informatika  
Fakultas Sains & Teknologi  
Universitas Muhammadiyah Kalimantan Timur  
Samarinda, 2023

# 1.LATAR

```
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 Latar extends World
{
    private int jeda=0;
    private int score;
    private int nyawa = 3;
    private int time;

    /**
     * Constructor for objects of class MyWorld.
     *
     */
    public Latar()
    {
        // Create a new world with 600x400 cells with a cell size of 1x1 pixels.
        super(500, 600, 1);
        tampilNyawa();
        tampilScore();
        showTime();
        time= 1000;
        addObject (new Pesawat(),250,550);
        prepare();
    }

    public void updateNyawa(int point)
    {
        nyawa = nyawa + point;
        tampilNyawa();

        if(nyawa ==0)
        {
            Greenfoot.stop();
            addObject(new kalah(),250,300);
        }
    }

    public void tampilNyawa()
    {
        showText("Nyawa :"+nyawa, 400, 25);
    }

    public void addScore (int points){
        score = score + points;
        tampilScore();
    }

    private void tampilScore()
    {
        showText ("Score :"+score, 100, 25);
    }

    private void countTime()
    {

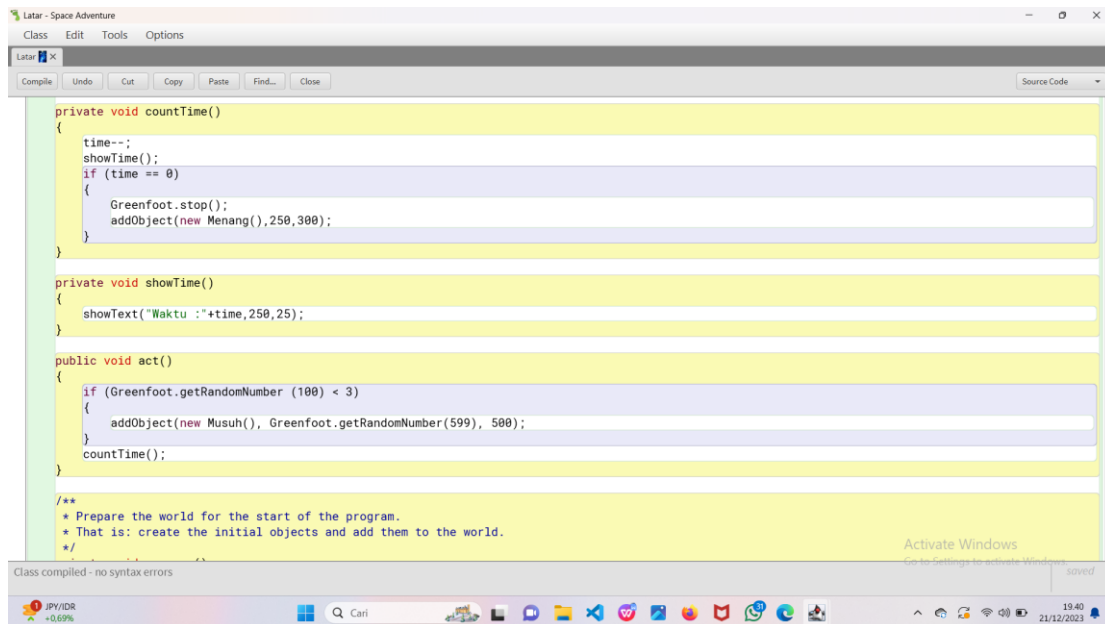
```

Class compiled - no syntax errors

Activate Windows  
Go to Settings to activate Windows.

19:39  
21/12/2023

19:40  
21/12/2023



```
private void countTime()
{
    time--;
    showTime();
    if (time == 0)
    {
        Greenfoot.stop();
        addObject(new Menang(), 250, 300);
    }
}

private void showTime()
{
    showText("Waktu : "+time, 250, 25);
}

public void act()
{
    if (Greenfoot.getRandomNumber (100) < 3)
    {
        addObject(new Musuh(), Greenfoot.getRandomNumber(599), 500);
    }
    countTime();
}

/**
 * Prepare the world for the start of the program.
 * That is: create the initial objects and add them to the world.
 */
```

### Latar (MyWorld) Class:

**Konstruktor (Latar()):** Membuat dunia permainan dengan ukuran 500x600 sel, menetapkan jumlah nyawa awal (3), waktu awal (1000), dan menyiapkan tampilan skor serta nyawa. Selain itu, menambahkan objek pesawat ke posisi (250, 550) dan mempersiapkan dunia (metode prepare()).

**updatenyawa(int point):** Menangani pembaruan nyawa. Jika pemain mendapatkan poin, nyawa akan ditambahkan sesuai dengan poin tersebut. Jika nyawa habis (0), permainan dihentikan dan ditambahkan objek "kalah".

**tampilnyawa():** Menampilkan jumlah nyawa di sudut kanan atas layar.

**addScore(int points):** Menambah skor berdasarkan poin yang diperoleh oleh pemain.

**tampilscore():** Menampilkan skor di sudut kiri atas layar.

**countTime():** Mengurangi waktu setiap siklus. Jika waktu habis, permainan dihentikan dan ditambahkan objek "Menang".

**showTime():** Menampilkan waktu tersisa di bagian atas layar.

**act():** Metode ini dipanggil setiap siklus permainan. Menambahkan objek musuh secara acak dengan probabilitas 3%. Selain itu, memanggil countTime() untuk menghitung waktu mundur.

#### Metode prepare():

Tidak diimplementasikan (kosong).

#### Variabel Kelas:

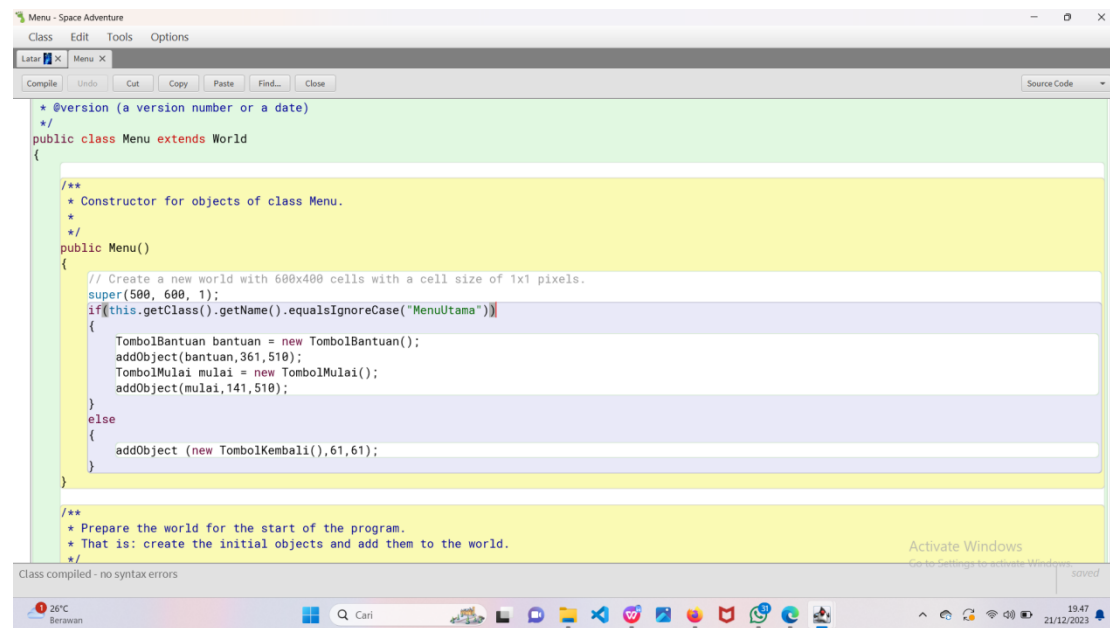
jeda: Variabel yang tidak digunakan dalam implementasi ini.

score: Menyimpan skor pemain.

nyawa: Menyimpan jumlah nyawa pemain.

time: Menyimpan waktu tersisa dalam detik.

## 1. MENU



```
* @version (a version number or a date)
*/
public class Menu extends World
{
    /**
     * Constructor for objects of class Menu.
     */
    public Menu()
    {
        // Create a new world with 600x400 cells with a cell size of 1x1 pixels.
        super(500, 600, 1);
        if(this.getClass().getName().equalsIgnoreCase("MenuUtama"))
        {
            TombolBantuan bantuan = new TombolBantuan();
            addObject(bantuan, 361, 510);
            TombolMulai mulai = new TombolMulai();
            addObject(mulai, 141, 510);
        }
        else
        {
            addObject (new TombolKembali(), 61, 61);
        }
    }

    /**
     * Prepare the world for the start of the program.
     * That is: create the initial objects and add them to the world.
     */
}
```

### konstruktor (Menu()):

Membuat dunia permainan dengan ukuran 500x600 sel dan ukuran sel 1x1 pixel menggunakan `super(500, 600, 1);`.

Memeriksa apakah nama kelas saat ini adalah "MenuUtama" menggunakan `this.getClass().getName().equalsIgnoreCase("MenuUtama")`.

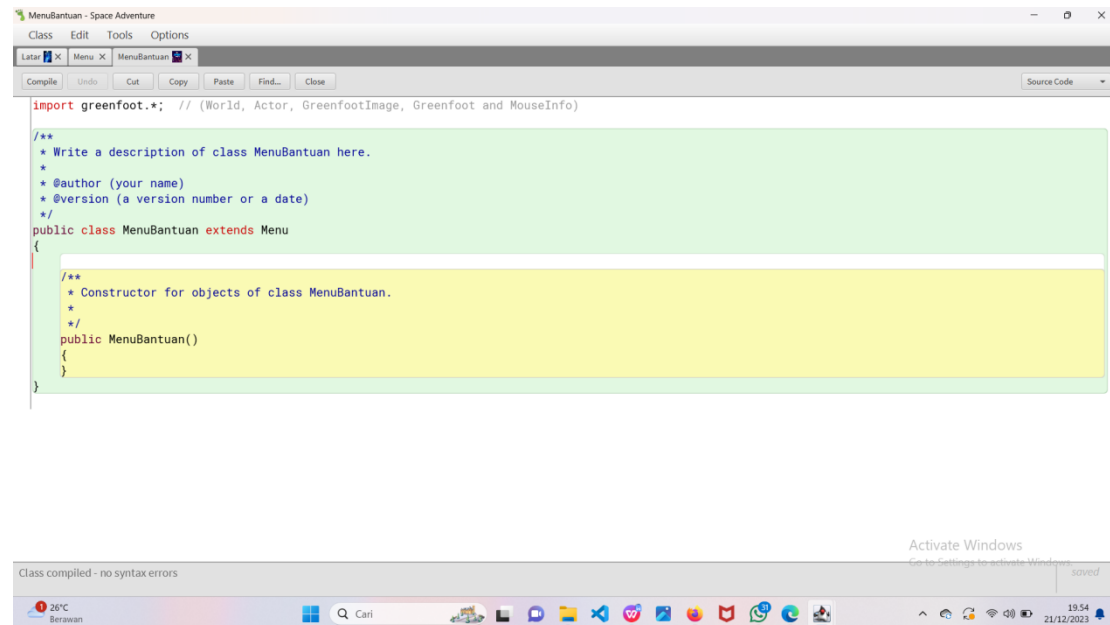
Jika kelas ini adalah "MenuUtama," tambahkan dua objek ke dunia:

TombolBantuan pada koordinat (361, 510).

TombolMulai pada koordinat (141, 510).

Jika nama kelas bukan "MenuUtama," tambahkan objek TombolKembali pada koordinat (61, 61).

## 2. MenuBantuan



The screenshot shows a Java IDE window titled "MenuBantuan - Space Adventure". The menu bar includes "Class", "Edit", "Tools", and "Options". The toolbar contains buttons for "Compile", "Undo", "Cut", "Copy", "Paste", "Find...", "Close", and a "Source Code" dropdown. The code editor displays the following Java code:

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

/**
 * Write a description of class MenuBantuan here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class MenuBantuan extends Menu
{
    /**
     * Constructor for objects of class MenuBantuan.
     *
     */
    public MenuBantuan()
    {
    }
}
```

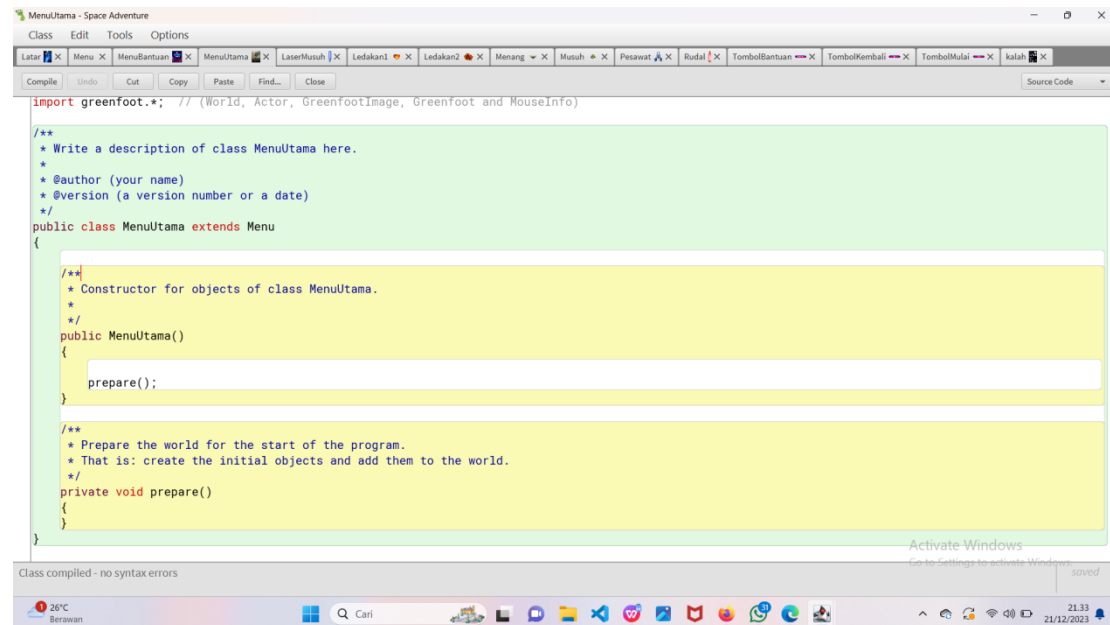
At the bottom of the IDE, a status bar indicates "Class compiled - no syntax errors". The Windows taskbar is visible at the bottom, showing the system clock as 19:54 on 21/12/2023.

### Konstruktor (MenuBantuan()):

Konstruktor ini digunakan untuk membuat objek dari kelas MenuBantuan.

Meskipun konstruktor ini tidak memiliki implementasi yang terlihat, itu secara implisit mewarisi konstruktor dari kelas induknya (Menu). Konstruktor ini tidak melakukan tindakan khusus pada pembuatannya.

### 3. Menu Utama



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

/**
 * Write a description of class MenuUtama here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class MenuUtama extends Menu
{
    /**
     * Constructor for objects of class MenuUtama.
     *
     */
    public MenuUtama()
    {
        prepare();
    }

    /**
     * Prepare the world for the start of the program.
     * That is: create the initial objects and add them to the world.
     */
    private void prepare()
    {
    }
}
```

#### Konstruktor (MenuUtama()):

Konstruktor ini digunakan untuk membuat objek dari kelas MenuUtama.

Memanggil metode prepare() yang mungkin digunakan untuk menyiapkan dan menambahkan objek-objek awal ke dalam dunia permainan. Namun, dalam kode yang diberikan, metode prepare() kosong.

#### Metode prepare():

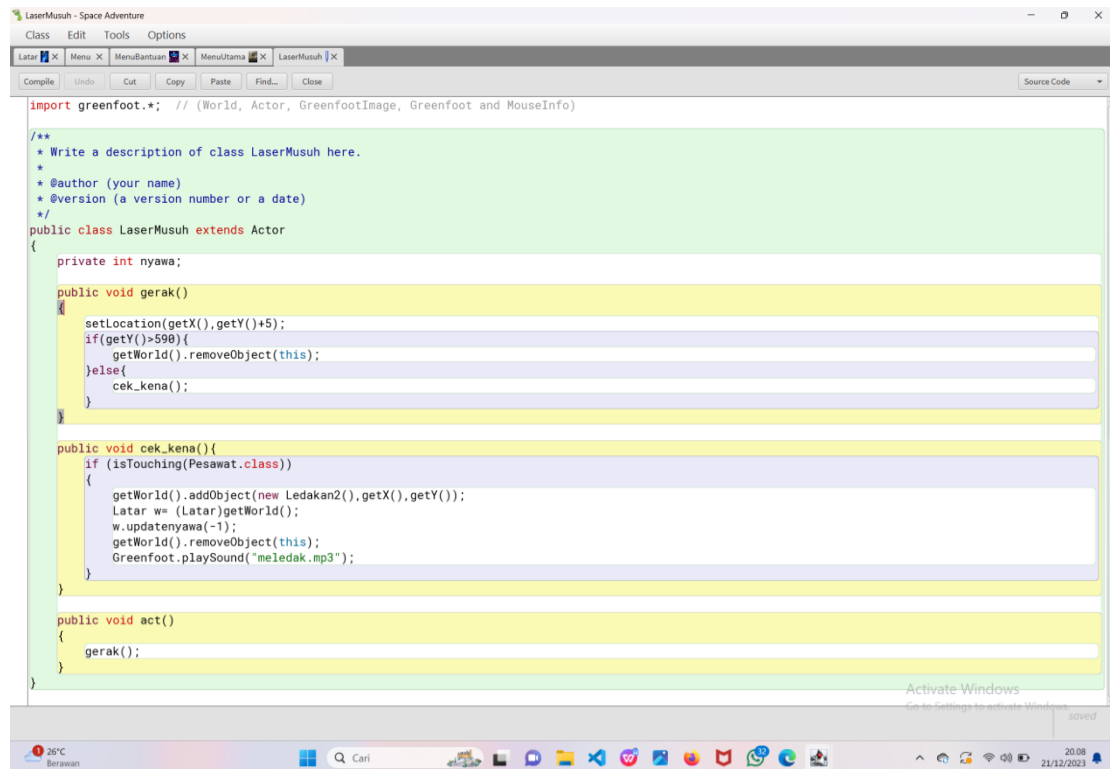
Metode ini disebutkan dalam konstruktor tetapi tidak memiliki implementasi yang terlihat.

Potensialnya, metode ini dapat diimplementasikan untuk menyiapkan dan menambahkan objek-objek awal ke dalam dunia permainan. Namun, dalam kode yang diberikan, metode ini tidak melakukan tindakan khusus.

#### Deklarasi Kelas (MenuUtama):

Pewarisan (extends Menu): Menandakan bahwa MenuUtama adalah turunan dari kelas Menu, mewarisi sifat dan metode dari kelas Menu

## 5. Laser Musuh



### Deklarasi Kelas (LaserMusuh):

Pewarisan (extends Actor): Menandakan bahwa LaserMusuh adalah turunan dari kelas Actor, yang merupakan kelas dasar untuk objek yang dapat berinteraksi dalam dunia Greenfoot.

### Variabel Kelas (private int nyawa):

Variabel nyawa digunakan untuk menyimpan jumlah nyawa. Namun, pada implementasi ini, tampaknya variabel ini tidak digunakan dan tidak diinisialisasi.

### Metode gerak():

Metode ini mengatur pergerakan objek LaserMusuh. Objek bergerak ke bawah dengan perubahan posisi Y sebesar 5.

Jika posisi Y melebihi 590, objek dihapus dari dunia (`getWorld().removeObject(this)`).

Jika belum mencapai batas bawah, metode `cek_kena()` dipanggil.

### Metode cek\_kena():

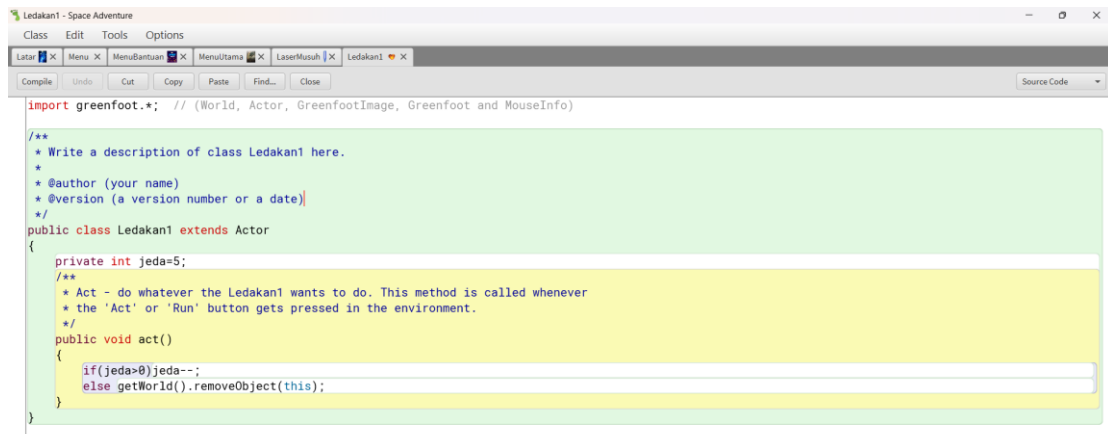
Metode ini memeriksa apakah objek LaserMusuh menyentuh objek dari kelas Pesawat (`isTouching(Pesawat.class)`).

Jika objek LaserMusuh menyentuh pesawat, objek `Ledakan2` ditambahkan ke dunia di posisi objek LaserMusuh dan pembaruan nyawa (`updatenyawa(-1)`) dilakukan menggunakan metode dari kelas Latar. Selanjutnya, objek LaserMusuh dihapus dari dunia, dan suara meledak diputar (`Greenfoot.playSound("meledak.mp3")`).

### Metode act():

Metode ini dipanggil setiap siklus permainan dan hanya memanggil metode `gerak()`.

## 6. Ledakan 1



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

/**
 * Write a description of class Ledakan1 here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Ledakan1 extends Actor
{
    private int jeda=5;
    /**
     * Act - do whatever the Ledakan1 wants to do. This method is called whenever
     * the 'Act' or 'Run' button gets pressed in the environment.
     */
    public void act()
    {
        if(jeda>0)jeda--;
        else getWorld().removeObject(this);
    }
}
```

### Variabel Kelas (**private int jeda=5**):

Variabel jeda digunakan untuk menentukan waktu tunda sebelum objek Ledakan1 dihapus dari dunia. Pada konstruktor, nilai awal jeda diatur menjadi 5.

### Metode **act()**:

Metode ini dipanggil setiap siklus permainan.

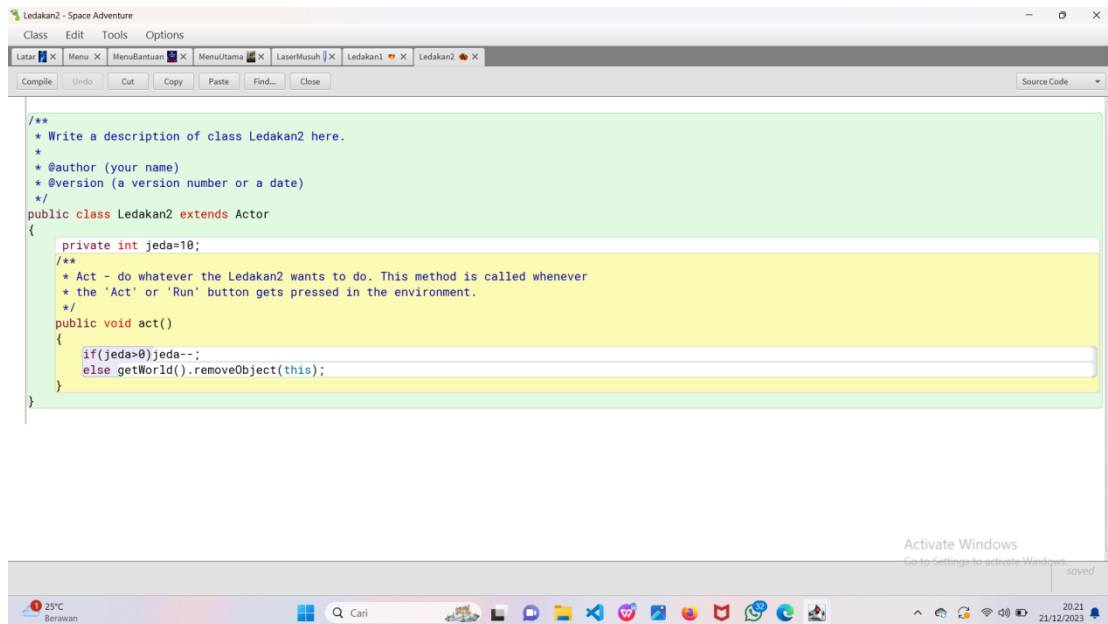
Pada setiap siklus, nilai jeda dikurangi satu.

Jika nilai jeda masih lebih besar dari 0, tidak ada tindakan lebih lanjut yang diambil.

Jika nilai jeda sudah mencapai 0, objek Ledakan1 dihapus dari dunia menggunakan `getWorld().removeObject(this)`.



## 7. ledakan2



```
/**
 * Write a description of class Ledakan2 here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Ledakan2 extends Actor
{
    private int jeda=10;
    /**
     * Act - do whatever the Ledakan2 wants to do. This method is called whenever
     * the 'Act' or 'Run' button gets pressed in the environment.
     */
    public void act()
    {
        if(jeda>0)jeda--;
        else getWorld().removeObject(this);
    }
}
```

### Variabel Kelas (**private int jeda=10**):

Variabel jeda digunakan untuk menentukan waktu tunda sebelum objek Ledakan2 dihapus dari dunia. Pada konstruktor, nilai awal jeda diatur menjadi 10.

### Metode **act()**:

Metode ini dipanggil setiap siklus permainan.

Pada setiap siklus, nilai jeda dikurangi satu.

Jika nilai jeda masih lebih besar dari 0, tidak ada tindakan lebih lanjut yang diambil.

Jika nilai jeda sudah mencapai 0, objek Ledakan2 dihapus dari dunia menggunakan `getWorld().removeObject(this)`.

## 8. Menang

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

/**
 * Write a description of class Menang here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Menang extends Actor
{
    /**
     * Act - do whatever the Menang wants to do. This method is called whenever
     * the 'Act' or 'Run' button gets pressed in the environment.
     */
    public void act()
    {
        // Add your action code here.
    }
}
```

### Deklarasi Kelas (Menang):

Pewarisan (extends Actor): Menandakan bahwa Menang adalah turunan dari kelas Actor, yang merupakan kelas dasar untuk objek yang dapat berinteraksi dalam dunia Greenfoot.

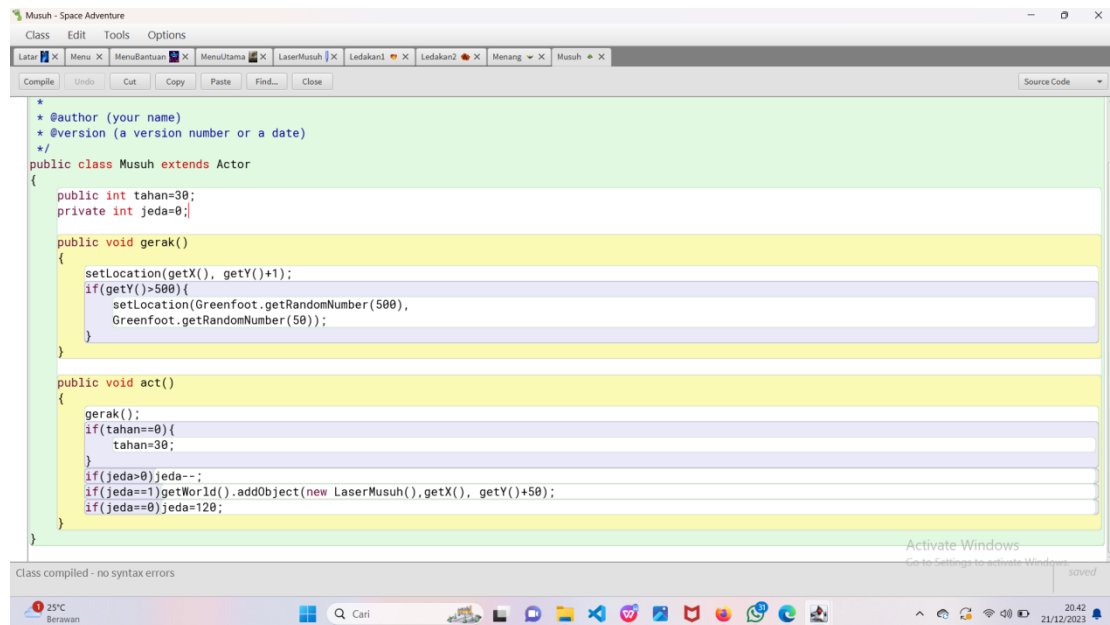
### Metode act():

Metode ini dipanggil setiap siklus permainan.

Saat ini, metode ini tidak memiliki implementasi atau tindakan (// Add your action code here.).

Jika diperlukan, implementasi dapat ditambahkan di sini untuk memberikan perilaku tertentu ketika objek Menang berinteraksi dalam dunia permainan. Misalnya, menambahkan efek visual atau suara yang menunjukkan bahwa pemain telah memenangkan permainan.

## 9. Musuh



```

* @author (your name)
* @version (a version number or a date)
*/
public class Musuh extends Actor
{
    public int tahan=30;
    private int jeda=0;

    public void gerak()
    {
        setLocation(getX(), getY()+1);
        if(getY()>500){
            setLocation(Greenfoot.getRandomNumber(500),
                Greenfoot.getRandomNumber(50));
        }
    }

    public void act()
    {
        gerak();
        if(tahan==0){
            tahan=30;
        }
        if(jeda>0)jeda--;
        if(jeda==1)getWorld().addObject(new LaserMusuh(),getX(), getY()+50);
        if(jeda==0)jeda=120;
    }
}

```

### Variabel Kelas (public int tahan=30 dan private int jeda=0):

Variabel tahan: Digunakan untuk menentukan periode waktu sebelum musuh dapat menembak lagi.

Variabel jeda: Digunakan sebagai penghitung untuk menentukan kapan musuh dapat menembak.

### Metode gerak():

Metode ini mengatur pergerakan objek Musuh. Objek bergerak ke bawah dengan perubahan posisi Y sebesar 1.

Jika posisi Y melebihi 500, objek dipindahkan ke posisi acak di atas layar.

### Metode act():

Metode ini dipanggil setiap siklus permainan.

Memanggil metode gerak() untuk mengatur pergerakan musuh.

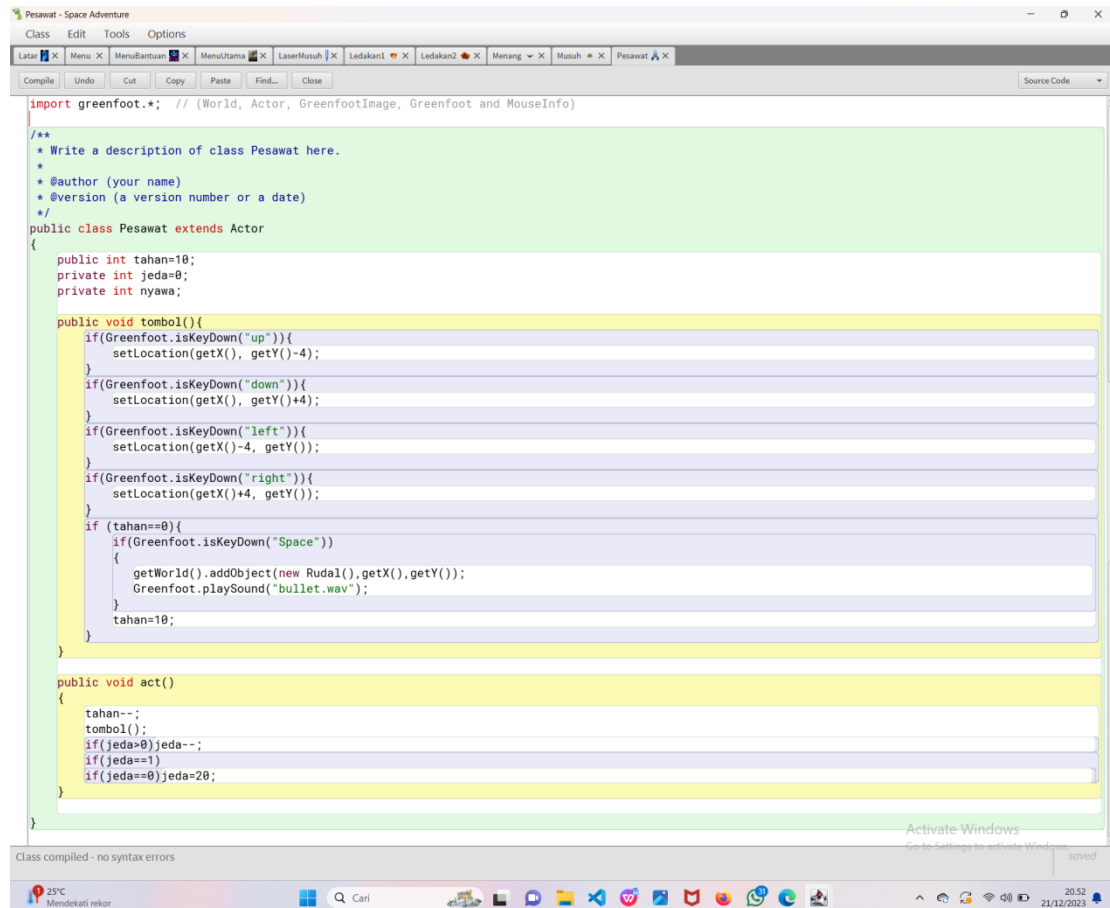
Jika nilai tahan adalah 0, setel kembali ke 30. Nilai tahan digunakan untuk mengatur periode waktu sebelum musuh dapat menembak lagi.

Memeriksa jika nilai jeda lebih besar dari 0, maka kurangkan nilai jeda.

Jika nilai jeda adalah 1, tambahkan objek LaserMusuh ke dunia pada posisi musuh dengan pergeseran Y sebesar 50.

Jika nilai jeda adalah 0, setel kembali nilai jeda ke 120.

## 10. Pesawat



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

/**
 * Write a description of class Pesawat here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Pesawat extends Actor
{
    public int tahan=10;
    private int jeda=0;
    private int nyawa;

    public void tombol(){
        if(Greenfoot.isKeyDown("up")){
            setLocation(getX(), getY()-4);
        }
        if(Greenfoot.isKeyDown("down")){
            setLocation(getX(), getY()+4);
        }
        if(Greenfoot.isKeyDown("left")){
            setLocation(getX()-4, getY());
        }
        if(Greenfoot.isKeyDown("right")){
            setLocation(getX()+4, getY());
        }
        if (tahan==0){
            if(Greenfoot.isKeyDown("Space"))
            {
                getWorld().addObject(new Rudal(),getX(),getY());
                Greenfoot.playSound("bullet.wav");
            }
            tahan=10;
        }
    }

    public void act()
    {
        tahan--;
        tombol();
        if(jeda>0)jeda--;
        if(jeda==1)
        if(jeda==0)jeda=20;
    }
}
```

### Variabel Kelas (public int tahan=10, private int jeda=0, dan private int nyawa):

Variabel tahan: Digunakan untuk menentukan periode waktu sebelum pesawat dapat menembak lagi.

Variabel jeda: Digunakan sebagai penghitung untuk menentukan interval waktu tertentu.

Variabel nyawa: Digunakan untuk menyimpan jumlah nyawa pesawat. Namun, pada implementasi ini, tampaknya variabel ini tidak digunakan dan tidak diinisialisasi.

### Metode tombol():

Metode ini mengatur respons terhadap input tombol dari pemain.

Jika tombol "up" ditekan, pesawat bergerak ke atas.

Jika tombol "down" ditekan, pesawat bergerak ke bawah.

Jika tombol "left" ditekan, pesawat bergerak ke kiri.

Jika tombol "right" ditekan, pesawat bergerak ke kanan.

Jika tombol "Space" ditekan dan tahan sudah mencapai 0, objek Rudal ditambahkan ke dunia pada posisi pesawat, dan suara tembakan diputar. Selanjutnya, tahan diatur kembali menjadi 10.

### Metode act():

Metode ini dipanggil setiap siklus permainan.

Mengurangkan nilai tahan untuk mengatur frekuensi tembakan pesawat.

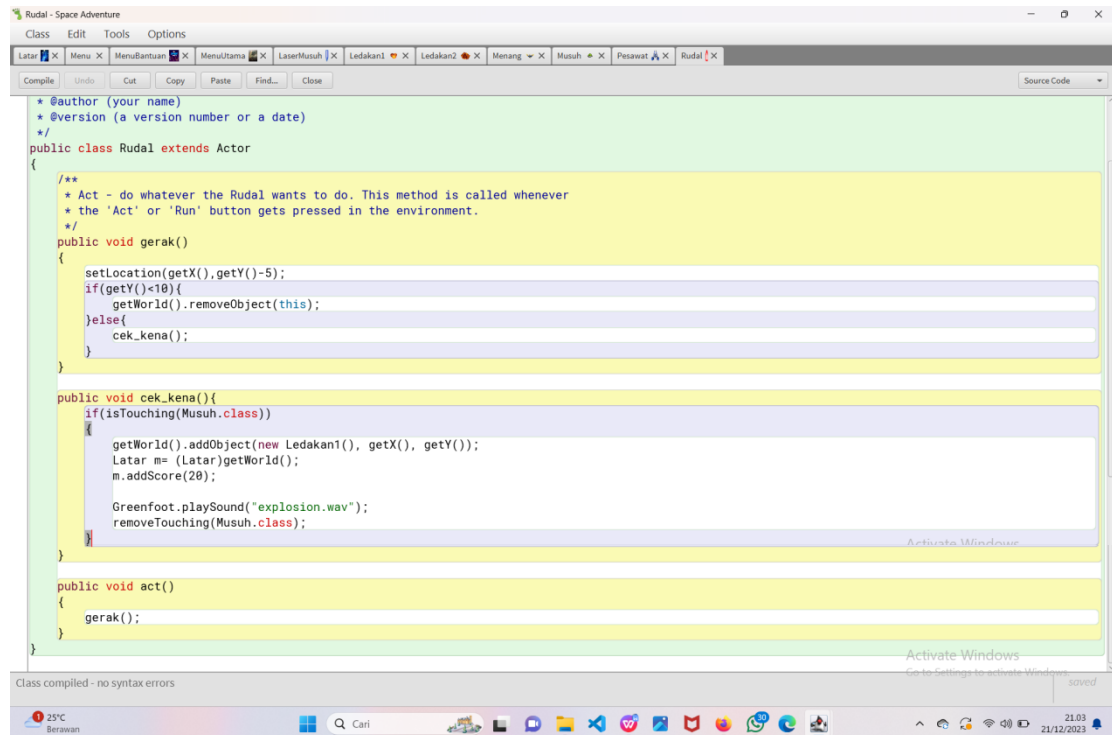
Memanggil metode tombol() untuk menanggapi input tombol dari pemain.

Mengurangkan nilai jeda jika nilainya lebih besar dari 0.

Jika nilai jeda adalah 1 (tanpa kondisi yang sesuai), kondisi if selanjutnya tidak memiliki tindakan.

Jika nilai jeda adalah 0, nilai jeda diatur kembali menjadi 20.

## 11. Rudal



```

* @author (your name)
* @version (a version number or a date)
*/
public class Rudal extends Actor
{
    /**
     * Act - do whatever the Rudal wants to do. This method is called whenever
     * the 'Act' or 'Run' button gets pressed in the environment.
     */
    public void gerak()
    {
        setLocation(getX(), getY()-5);
        if(getY() < 10){
            getWorld().removeObject(this);
        }else{
            cek_kena();
        }
    }

    public void cek_kena(){
        if(isTouching(Musuh.class))
        {
            getWorld().addObject(new Ledakan1(), getX(), getY());
            Latar m = (Latar) getWorld();
            m.addScore(20);

            Greenfoot.playSound("explosion.wav");
            removeTouching(Musuh.class);
        }
    }

    public void act()
    {
        gerak();
    }
}

```

### Metode gerak():

Metode ini mengatur pergerakan objek Rudal. Objek bergerak ke atas dengan perubahan posisi Y sebesar -5.

Jika posisi Y kurang dari 10, objek Rudal dihapus dari dunia (`getWorld().removeObject(this)`).

Jika belum mencapai batas atas, metode `cek_kena()` dipanggil.

### Metode cek\_kena():

Metode ini memeriksa apakah objek Rudal menyentuh objek dari kelas Musuh (`isTouching(Musuh.class)`).

Jika objek Rudal menyentuh musuh, objek `Ledakan1` ditambahkan ke dunia pada posisi objek Rudal, dan skor pemain diupdate menggunakan metode `addScore(20)` dari kelas `Latar`.

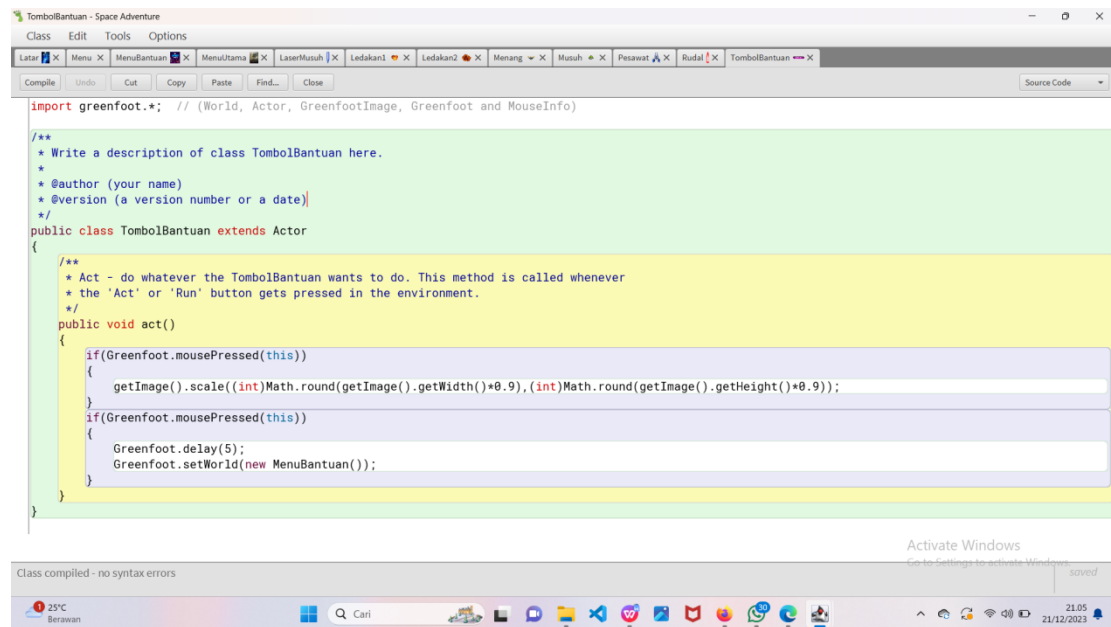
Suara ledakan diputar menggunakan `Greenfoot.playSound("explosion.wav")`.

Objek Musuh yang bersentuhan dengan rudal dihapus dari dunia menggunakan `removeTouching(Musuh.class)`.

### Metode act():

Metode ini dipanggil setiap siklus permainan dan hanya memanggil metode `gerak()`.

## 12. Tombol Bantuan



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

/**
 * Write a description of class TombolBantuan here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class TombolBantuan extends Actor
{
    /**
     * Act - do whatever the TombolBantuan wants to do. This method is called whenever
     * the 'Act' or 'Run' button gets pressed in the environment.
     */
    public void act()
    {
        if(Greenfoot.mousePressed(this))
        {
            getImage().scale((int)Math.round(getImage().getWidth()*0.9), (int)Math.round(getImage().getHeight()*0.9));
        }
        if(Greenfoot.mousePressed(this))
        {
            Greenfoot.delay(5);
            Greenfoot.setWorld(new MenuBantuan());
        }
    }
}
```

Class compiled - no syntax errors

Activate Windows  
Go to Settings to activate Windows

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

saved

21.05  
21/12/2023

25°C  
Berawan

Cari

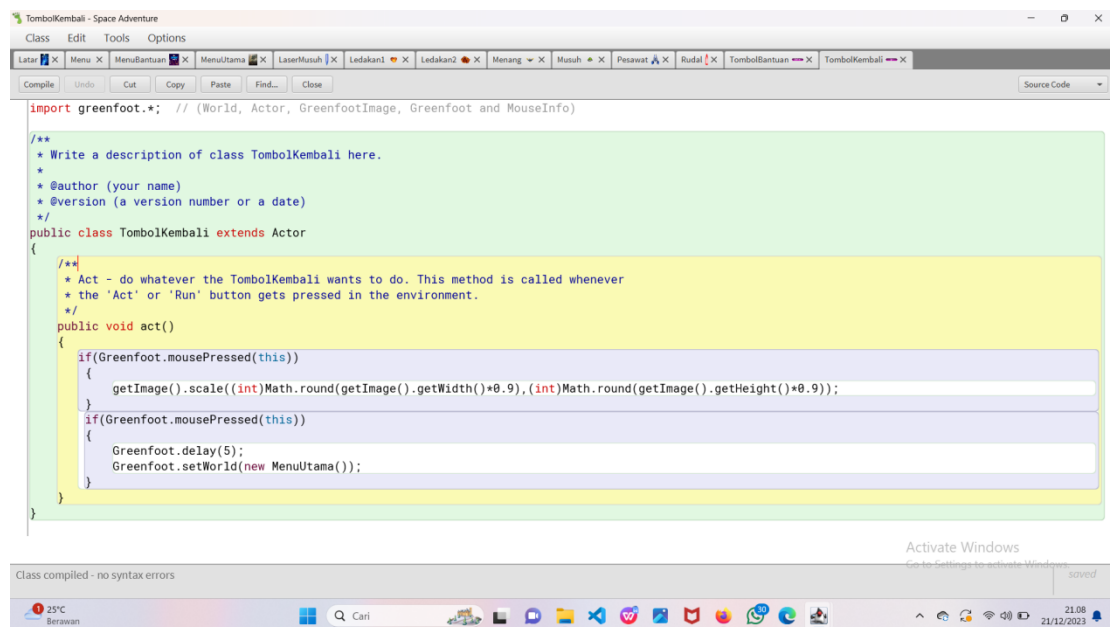
saved

21.05  
21/12/2023

25°C  
Berawan

Cari

### 13. Tombol Kembali



#### Metode act():

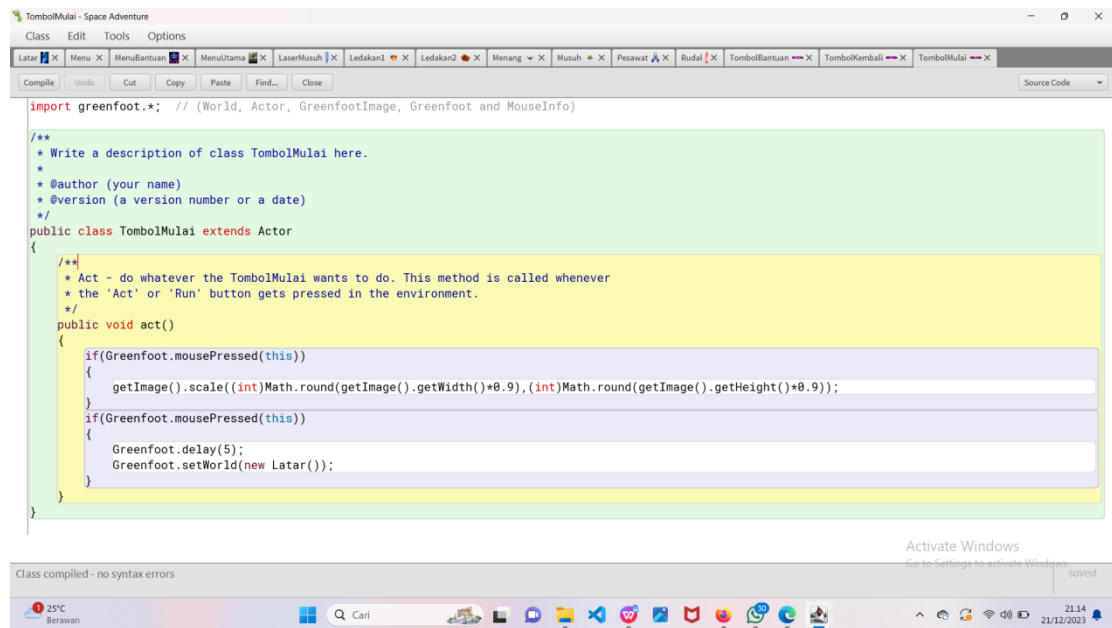
Metode ini dipanggil setiap siklus permainan.

Jika tombol mouse ditekan di atas objek TombolKembali (`Greenfoot.mousePressed(this)`), ukuran gambar tombol tersebut dikurangi sebesar 10% menggunakan `getImage().scale()`.

Selanjutnya, jika tombol mouse ditekan lagi di atas objek TombolKembali, terdapat penundaan (delay) selama 5 siklus permainan (`Greenfoot.delay(5)`).

Setelah penundaan, dunia permainan diubah menjadi objek dari kelas `MenuUtama` menggunakan `Greenfoot.setWorld(new MenuUtama())`.

## 14. Tombol Mulai



### Metode act():

Metode ini dipanggil setiap siklus permainan.

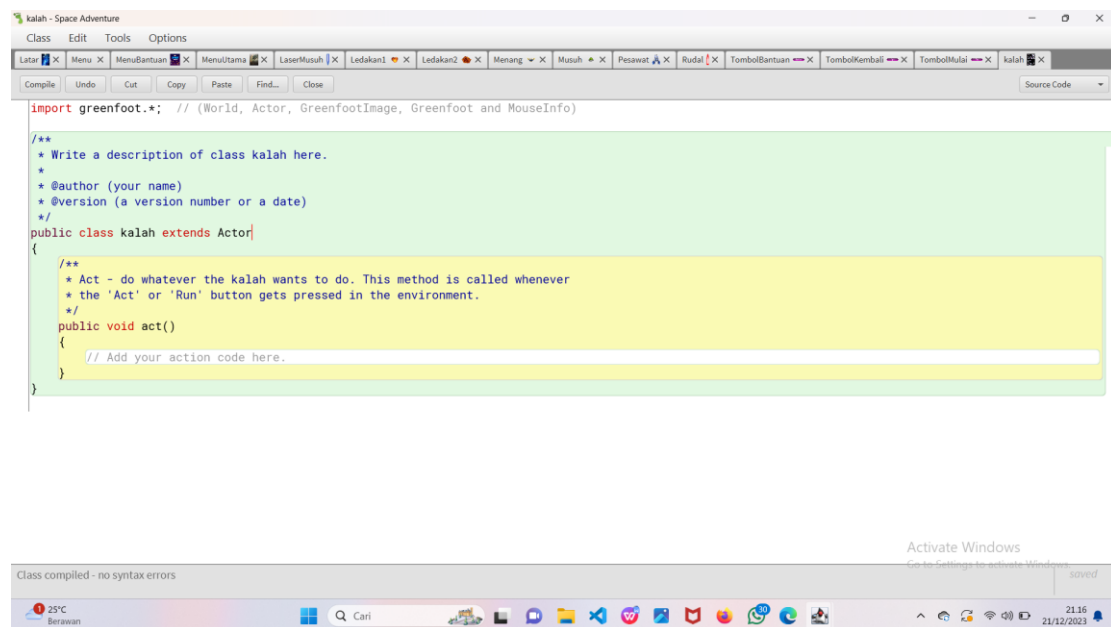
Jika tombol mouse ditekan di atas objek TombolMulai (`Greenfoot.mousePressed(this)`), ukuran gambar tombol tersebut dikurangi sebesar 10% menggunakan `getImage().scale()`.

Selanjutnya, jika tombol mouse ditekan lagi di atas objek TombolMulai, terdapat penundaan (delay) selama 5 siklus permainan (`Greenfoot.delay(5)`).

Setelah penundaan, dunia permainan diubah menjadi objek dari kelas Latar menggunakan `Greenfoot.setWorld(new Latar())`.



## 15. Kalah



Program di atas adalah implementasi kelas kalah yang merupakan turunan dari kelas Actor pada lingkungan pengembangan Greenfoot. Namun, dalam kode yang diberikan, tidak ada aksi khusus yang diimplementasikan di dalam metode act(). Dengan kata lain, kelas kalah ini tidak memiliki perilaku tertentu yang akan dieksekusi saat objeknya aktif dalam dunia permainan.