

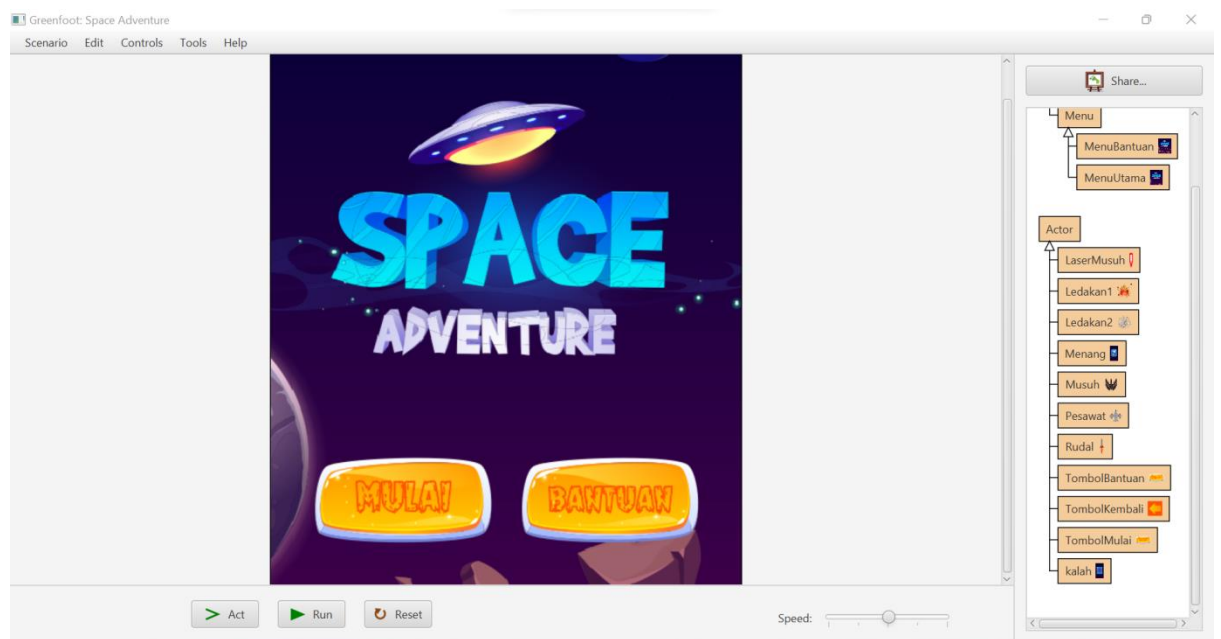
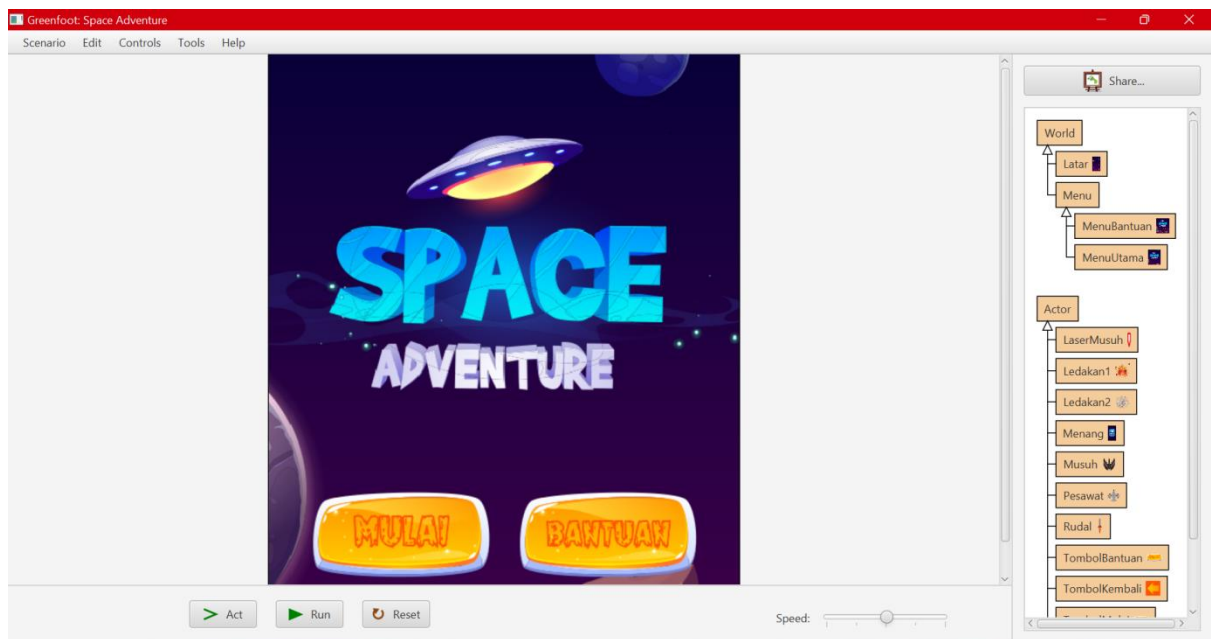
UAS PBO



Anggota Kelompok 9:

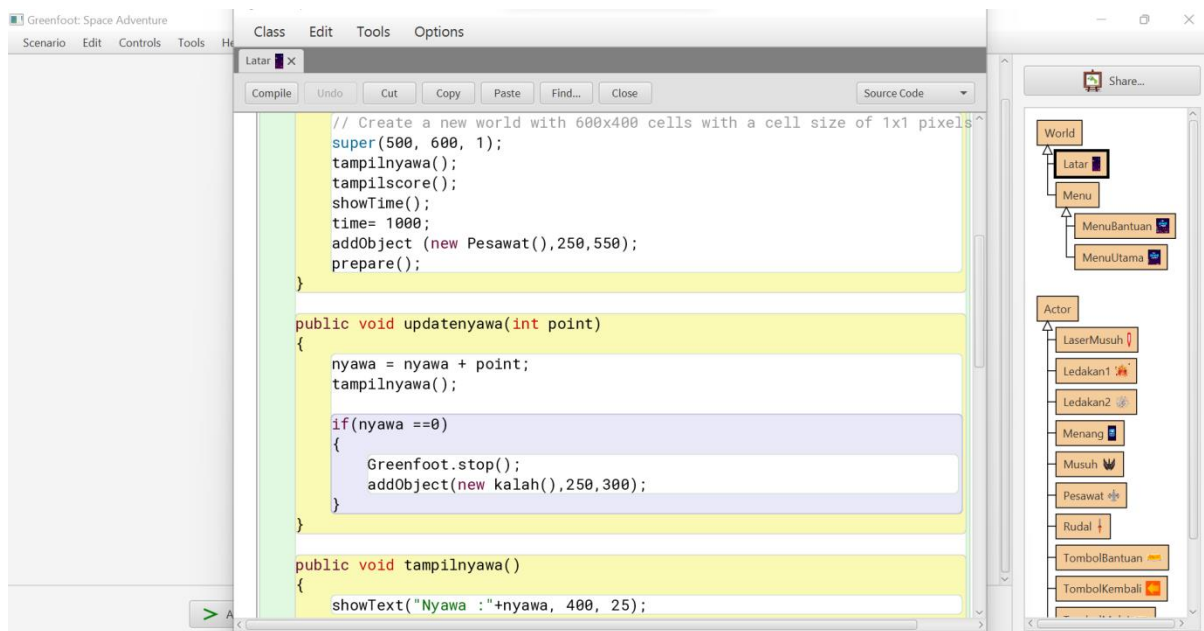
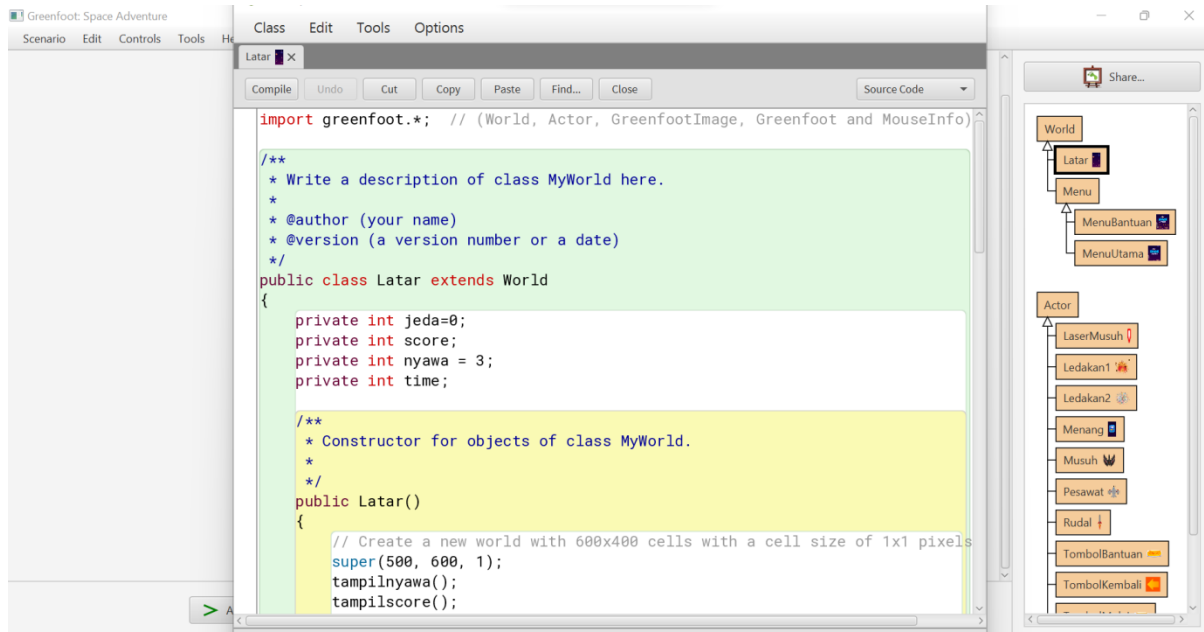
- 1. Salsabilla Ivania Kimalda / 2211102441226**
- 2. Lidya Natasha / 2211102441230**
- 3. Akmal Mujahid / 2211102441021**
- 4. Muhammad Rizky Akbar Rianda / 2211102441224**

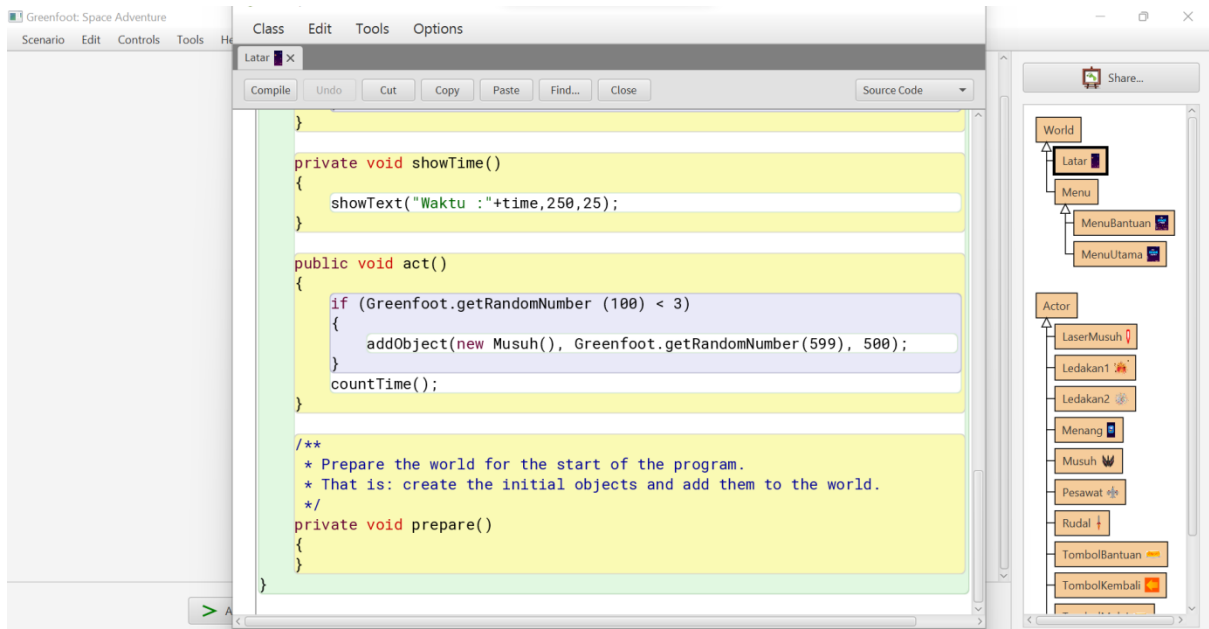
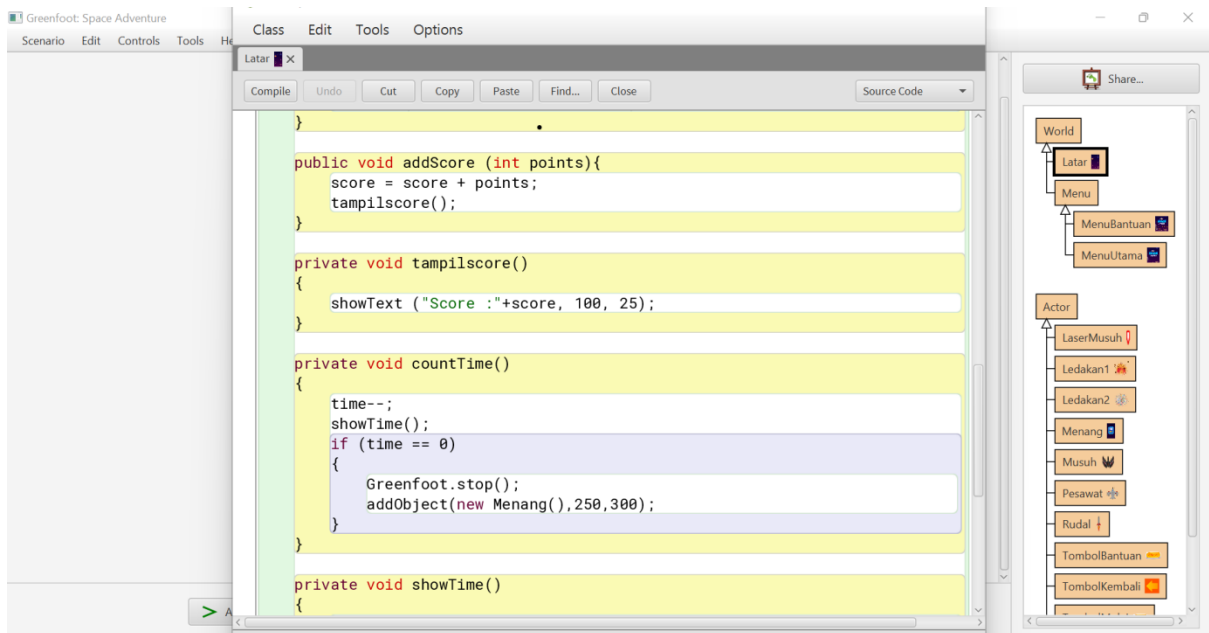
**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR**



1. World

Latar :





Penjelasan:

Impor Pustaka

java

import greenfoot.*;

Baris ini berisi pernyataan impor yang menggunakan kelas-kelas dari pustaka Greenfoot. Pustaka ini digunakan untuk membuat simulasi grafis dan permainan dalam konteks pembelajaran pemrograman dengan Java.

Deklarasi Kelas

```
java
```

```
public class Latar extends World
```

Deklarasi kelas `Latar` yang merupakan turunan dari kelas `World`. Dalam Greenfoot, sebuah dunia (`World`) adalah ruang di mana aktor-aktor berinteraksi.

Variabel Instance

```
java
```

```
private int jeda = 0;private int score;private int nyawa = 3;private int  
time;
```

Variabel instance (variabel anggota kelas) yang digunakan untuk menyimpan dan mengelola data spesifik untuk setiap instance dari kelas `Latar`.

- `jeda` diinisialisasi dengan nilai 0.
- `score` menyimpan skor pemain.
- `nyawa` merepresentasikan jumlah nyawa pemain.
- `time` digunakan sebagai timer mundur.

Konstruktor

```
java
```

```
public Latar()
```

Konstruktor dari kelas `Latar` yang digunakan untuk inisialisasi dunia dan menyiapkan kondisi awal. Ini membuat dunia dengan ukuran 500x600 piksel, dengan ukuran sel sebesar 1x1 piksel. Menginisialisasi `nyawa` menjadi 3, menampilkan skor dan waktu, menambahkan objek pesawat (`Pesawat`), dan memanggil metode `prepare()`.

Metode `updatenyawa(int point)`

Metode ini mengupdate jumlah nyawa pemain dan menampilkannya. Jika nyawa habis, permainan dihentikan dan objek "kalah" (`kalah`) ditambahkan ke dunia.

Metode `tampilnyawa()`

Metode ini menampilkan jumlah nyawa pemain di layar.

Metode `addScore(int points)`

Metode ini menambahkan skor ke skor pemain dan menampilkannya di layar.

Metode `tampilscore()`

Metode ini menampilkan skor pemain di layar.

Metode `countTime()`

Metode ini mengurangi waktu yang tersisa dan menampilkannya. Jika waktu habis, permainan dihentikan dan objek "Menang" (`Menang`) ditambahkan ke dunia.

Metode `showTime()`

Metode ini menampilkan waktu yang tersisa di layar.

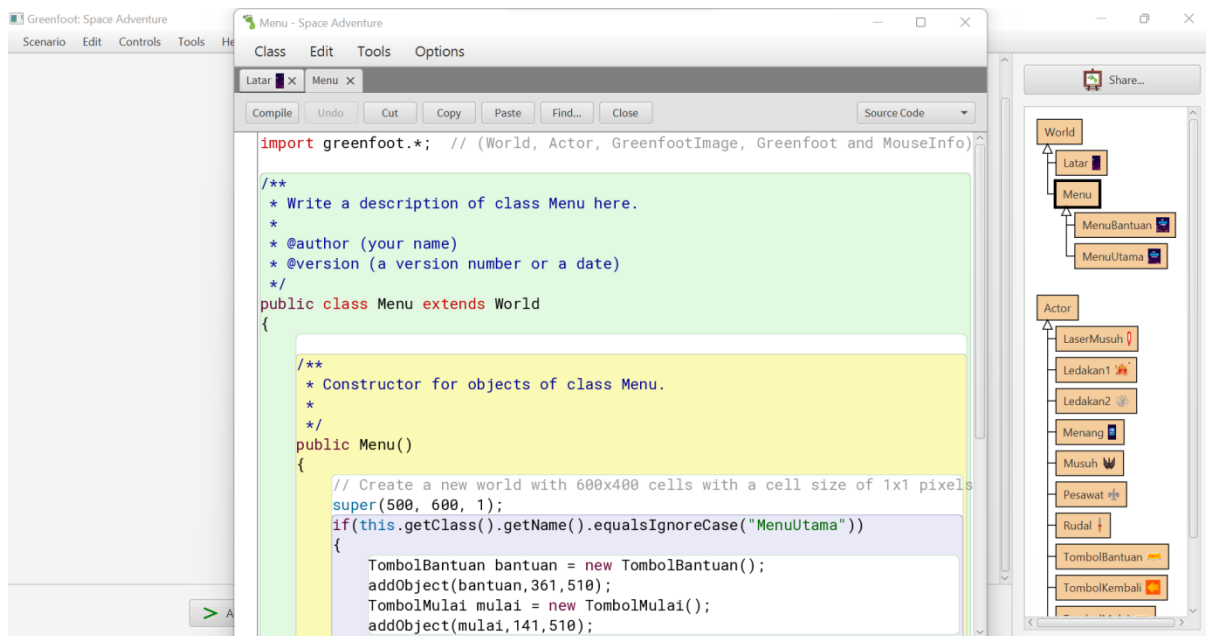
Metode `act()`

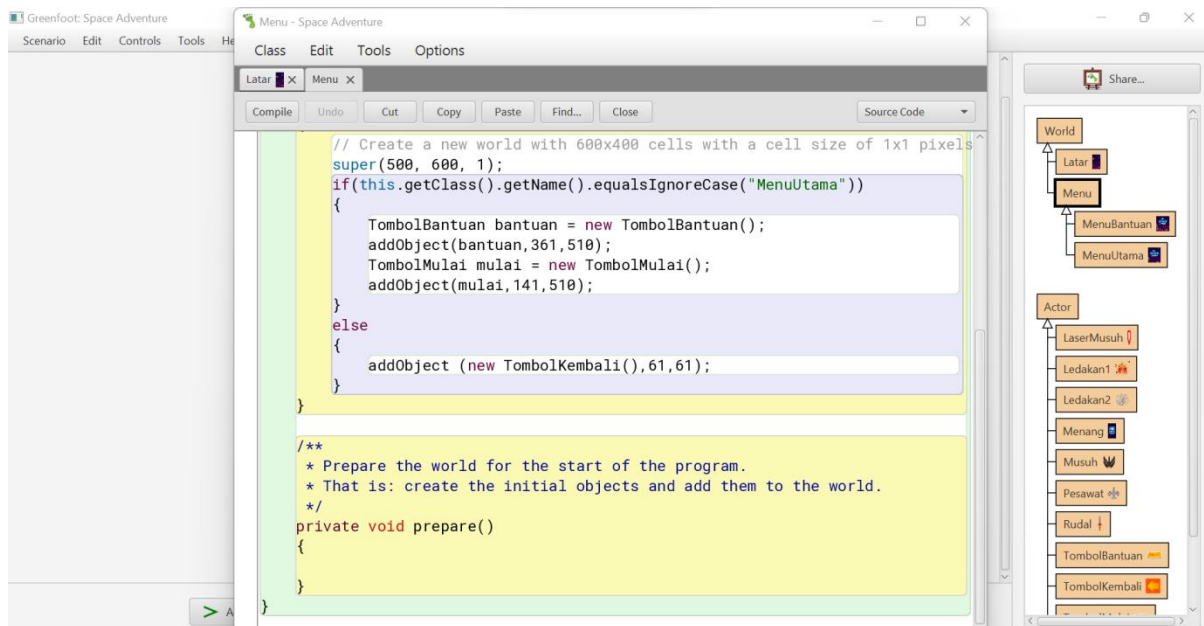
Metode ini adalah metode khusus dalam Greenfoot yang secara otomatis dipanggil oleh framework. Digunakan untuk mendefinisikan perilaku dunia. Di dalamnya, musuh (`Musuh`) ditambahkan secara acak ke dunia, dan metode `countTime()` dipanggil.

Metode `prepare()`

Metode ini adalah metode yang kosong, kemungkinan digunakan untuk menyiapkan dan menginisialisasi objek-objek awal dalam dunia. Saat ini, tidak berisi kode.

Menu :





Penjelasam:

Impor Pustaka

java

import greenfoot.*;

Baris ini mengimpor kelas-kelas yang diperlukan dari pustaka Greenfoot.

Deklarasi Kelas

java

public class Menu extends World

Deklarasi kelas `Menu` yang merupakan turunan dari kelas `World`. Ini mengindikasikan bahwa `Menu` adalah dunia dalam Greenfoot.

Konstruktor

java

public Menu()

Konstruktor dari kelas `Menu` yang digunakan untuk inisialisasi dunia dan menambahkan objek-objek ke dalamnya.

- Membuat dunia dengan ukuran 500x600 piksel, dengan ukuran sel sebesar 1x1 piksel menggunakan `super(500, 600, 1)`.

- Menggunakan kondisi `if` untuk memeriksa apakah kelas yang memanggilnya adalah `MenuUtama`. Jika ya, maka beberapa tombol (objek `TombolBantuan` dan `TombolMulai`) ditambahkan ke dunia pada posisi tertentu.
- Jika bukan `MenuUtama`, maka objek `TombolKembali` ditambahkan ke dunia pada posisi tertentu.

Metode `prepare()`

```
java
```

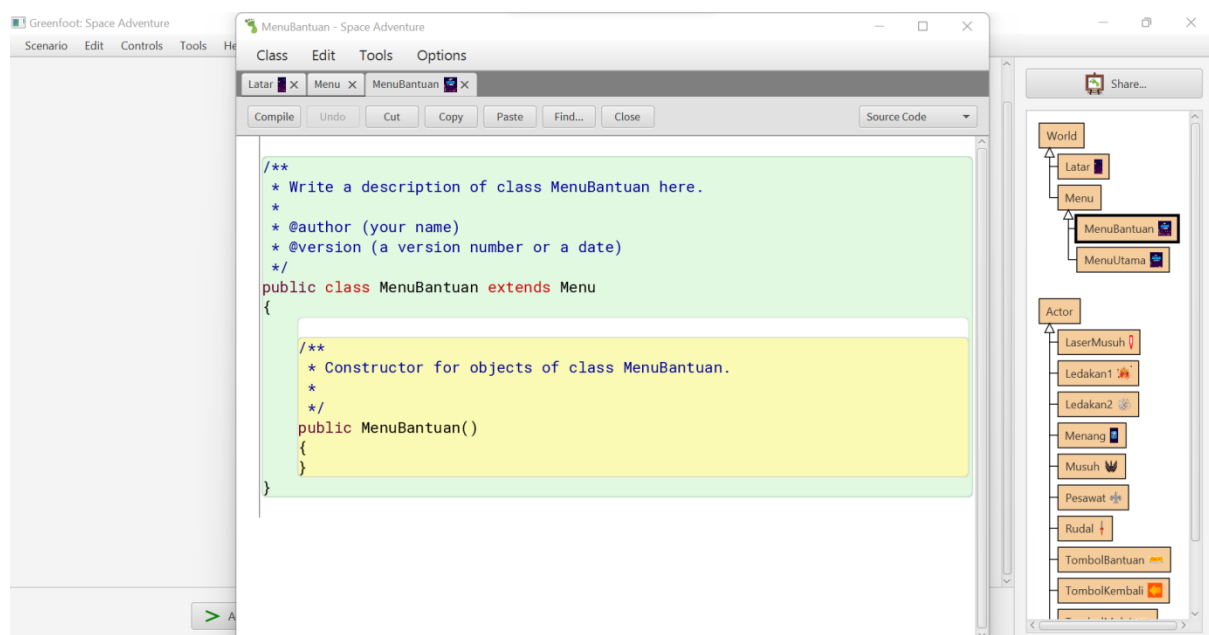
```
private void prepare()
```

```
{
}
```

Ini adalah metode yang kosong dan saat ini tidak berisi kode. Metode ini mungkin ditujukan untuk menyiapkan dan menginisialisasi objek-objek dalam dunia, tetapi dalam implementasi saat ini, tidak ada kode di dalamnya.

Sesuai dengan struktur kelas, sepertinya metode `prepare()` saat ini tidak digunakan dan mungkin bisa dihapus, kecuali jika akan digunakan di masa depan.

Menu Bantuan:



Penjelasan:

Impor Pustaka

```
java
```



```
import greenfoot.*;
```

Baris ini mengimpor kelas-kelas yang diperlukan dari pustaka Greenfoot.

Deklarasi Kelas

```
java
```

```
public class MenuBantuan extends Menu
```

Deklarasi kelas `MenuBantuan` yang merupakan turunan dari kelas `Menu`. Ini menunjukkan bahwa `MenuBantuan` adalah jenis khusus dari `Menu`.

Konstruktor

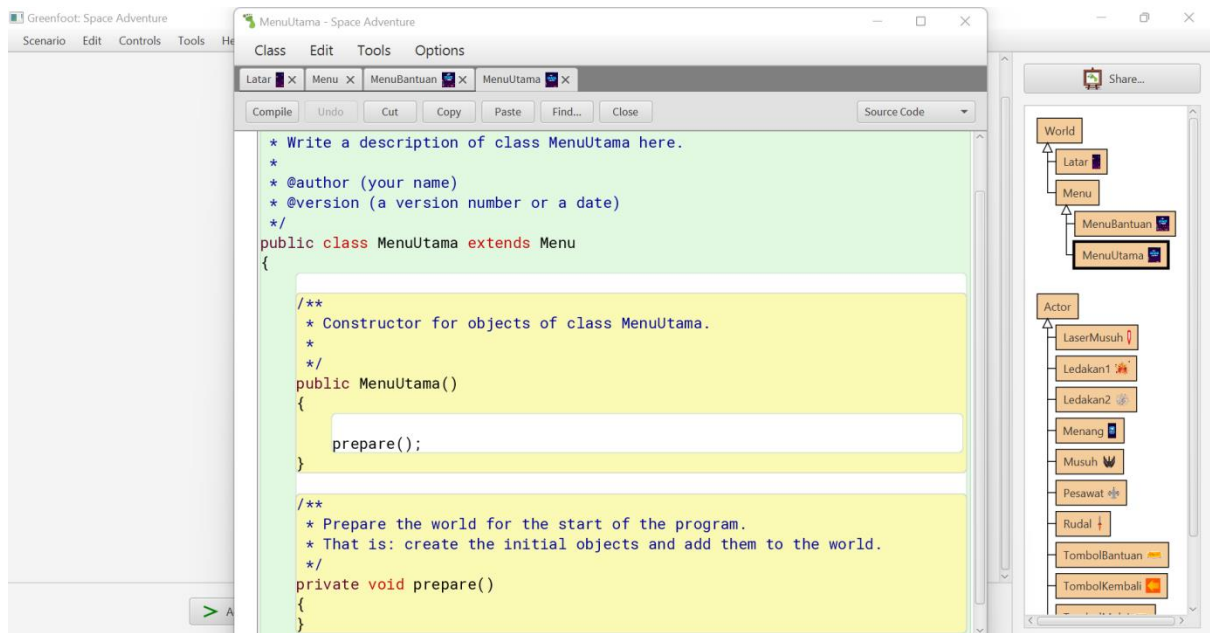
```
java
```

```
public MenuBantuan()
```

```
{  
  
}
```

Konstruktor dari kelas `MenuBantuan`. Saat ini, konstruktor ini tidak berisi kode apapun, yang berarti tidak ada tindakan khusus yang diambil ketika objek `MenuBantuan` dibuat.

Menu Utama:



Penjelasan:

Impor Pustaka

```
java
```

```
import greenfoot.*;
```

Baris ini mengimpor kelas-kelas yang diperlukan dari pustaka Greenfoot.

Deklarasi Kelas

```
java
```

```
public class MenuUtama extends Menu
```

Deklarasi kelas `MenuUtama` yang merupakan turunan dari kelas `Menu`. Ini menunjukkan bahwa `MenuUtama` adalah jenis khusus dari `Menu`.

Konstruktor

```
java
```

```
public MenuUtama()
```

```
{
```

```
    prepare();
```

```
}
```

Konstruktor dari kelas `MenuUtama`. Saat objek `MenuUtama` dibuat, konstruktor ini secara otomatis memanggil metode `prepare()`.

Metode `prepare()`

```
java
```

```
private void prepare()
```

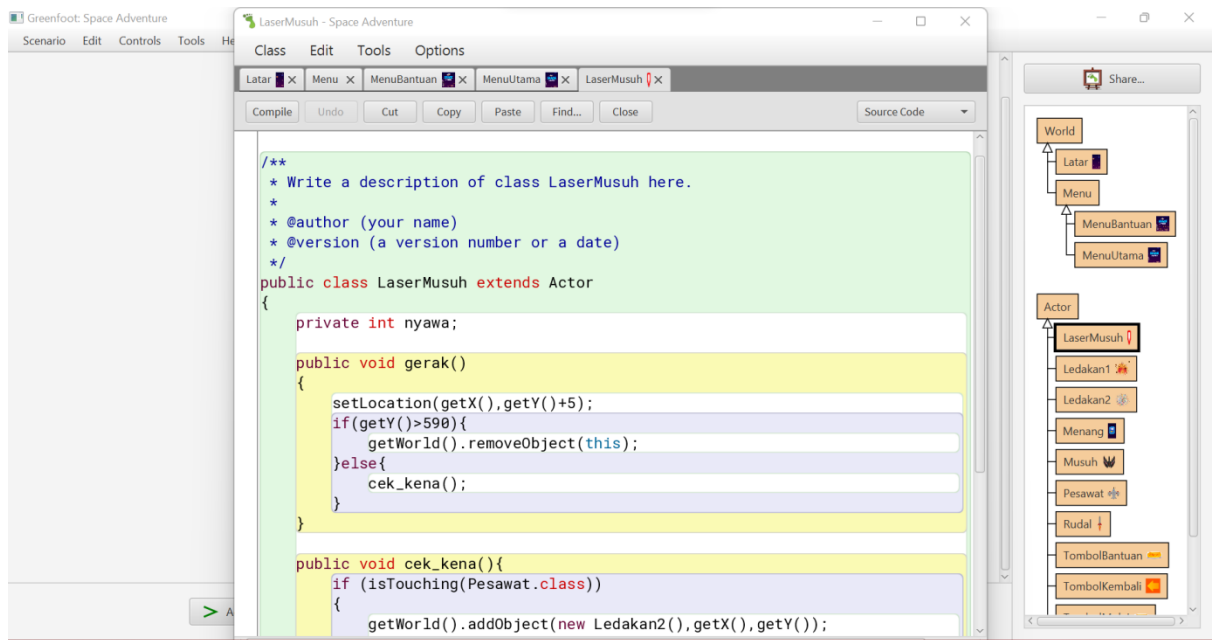
```
{
```

```
}
```

Ini adalah metode yang kosong dan saat ini tidak berisi kode. Metode ini mungkin ditujukan untuk menyiapkan dan menginisialisasi objek-objek dalam dunia, tetapi dalam implementasi saat ini, tidak ada kode di dalamnya

2. Actor

Laser Musuh :



Penjelasan:

Impor Pustaka

java

import greenfoot.*;

Baris ini mengimpor kelas-kelas yang diperlukan dari pustaka Greenfoot.

Deklarasi Kelas

java

public class LaserMusuh extends Actor

Deklarasi kelas `LaserMusuh` yang merupakan turunan dari kelas `Actor`. Kelas ini mewakili objek laser yang ditembakkan oleh musuh.

Variabel Instance

java

```
private int nyawa;
```

Variabel instance (variabel anggota kelas) yang mungkin tadinya dimaksudkan untuk menyimpan jumlah nyawa (life) dari objek `LaserMusuh`. Namun, tampaknya variabel ini tidak digunakan dalam implementasi saat ini.

Metode `gerak()`

java

```
public void gerak()
{
    setLocation(getX(), getY() + 5);
    if (getY() > 590)
    {
        getWorld().removeObject(this);
    }
    else
    {
        cek_kenan();
    }
}
```

Metode ini mengatur pergerakan laser musuh ke bawah. Jika laser mencapai batas bawah (`getY() > 590`), laser dihapus dari dunia. Jika tidak, metode `cek_kenan()` dipanggil untuk memeriksa apakah laser menyentuh objek `Pesawat`.

Metode `cek_kenan()`

java

```
public void cek_kenan()
{
    if (isTouching(Pesawat.class))
```

```

    {
        getWorld().addObject(new Ledakan2(), getX(), getY());
        Latar w = (Latar) getWorld();
        w.updatenyawa(-1);
        getWorld().removeObject(this);
        Greenfoot.playSound("meledak.mp3");
    }
}

```

Metode ini memeriksa apakah laser musuh menyentuh objek Pesawat. Jika iya, itu menambahkan objek ledakan (`Ledakan2`) ke dunia, mengurangi nyawa pemain menggunakan metode `updatenyawa(-1)` dari kelas `Latar`, dan menghapus laser dari dunia. Selain itu, memutar suara ledakan.

Metode `act()`

```

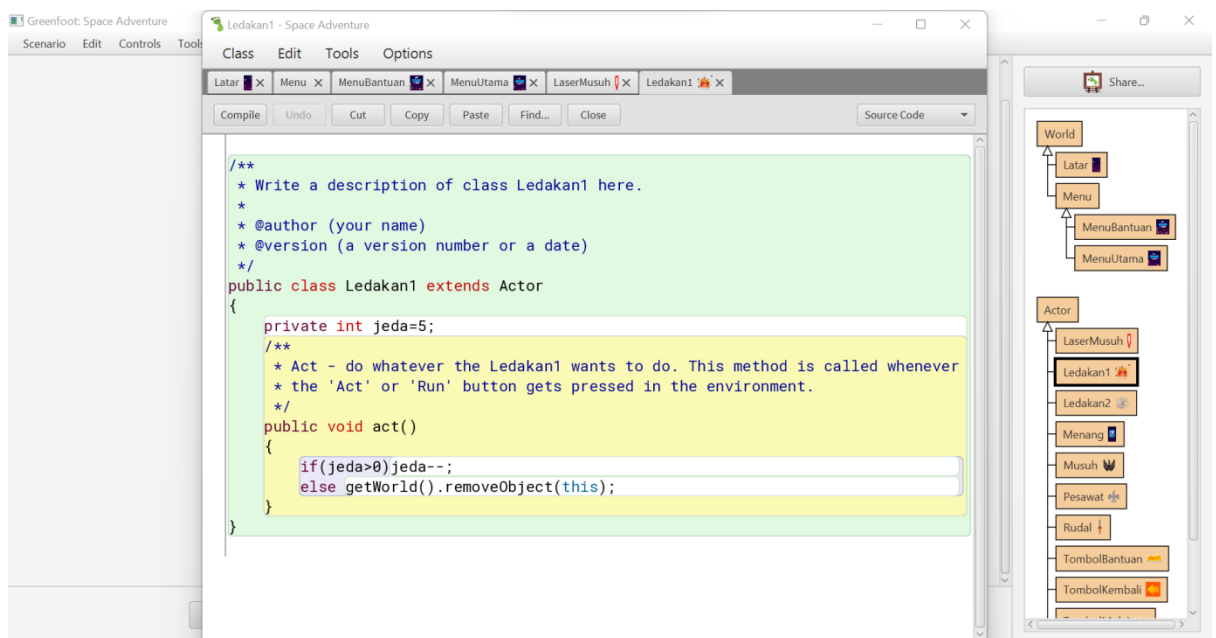
java
public void act()
{
    gerak();
}

```

Metode ini merupakan metode yang dipanggil secara otomatis oleh Greenfoot framework. Dalam hal ini, metode ini hanya memanggil metode `gerak()`.

Kelas ini, secara umum, digunakan untuk mewakili perilaku laser yang ditembakkan oleh musuh. Saat laser bergerak ke bawah, ia memeriksa apakah menyentuh pesawat. Jika iya, itu mengakibatkan ledakan dan mengurangi nyawa pemain.

Ledakan 1 :



Penjelasan:

Impor Pustaka

java

```
import greenfoot.*;
```

Baris ini mengimpor kelas-kelas yang diperlukan dari pustaka Greenfoot.

Deklarasi Kelas

java

```
public class Ledakan1 extends Actor
```

Deklarasi kelas `Ledakan1` yang merupakan turunan dari kelas `Actor`. Kelas ini mewakili objek ledakan pertama.

Variabel Instance

```
java
```

```
private int jeda = 5;
```

Variabel instance (variabel anggota kelas) yang digunakan untuk mengatur jeda sebelum objek ledakan dihapus dari dunia. Nilai awalnya adalah 5.

Metode `act()`

```
java
```

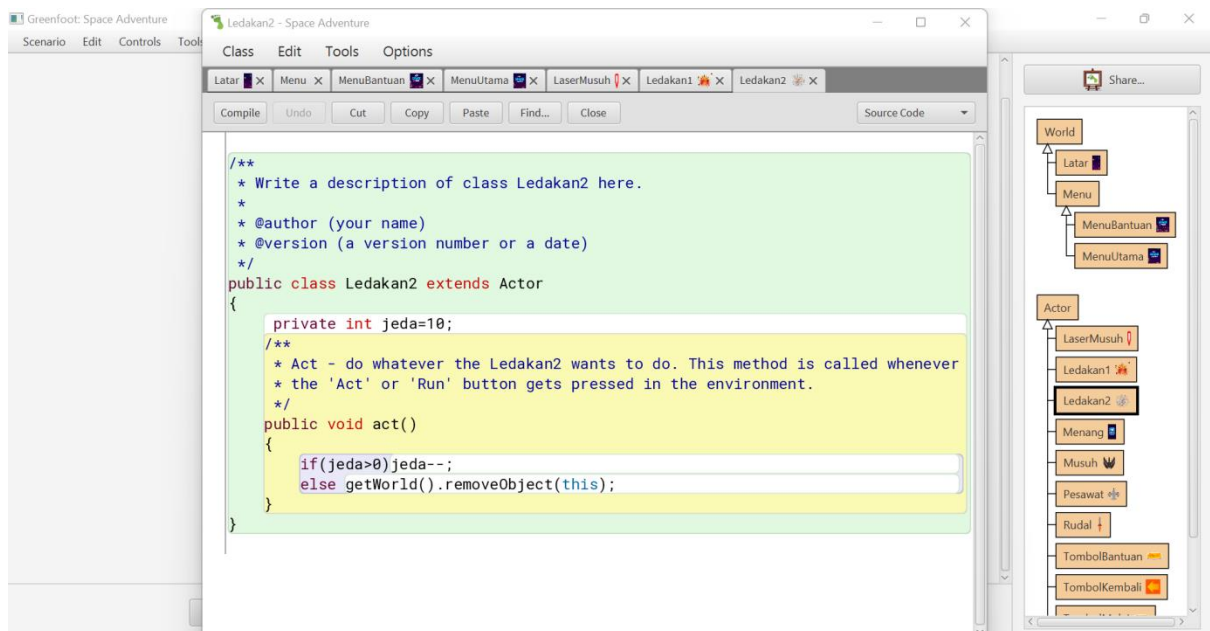
```
public void act()
```

```
{  
    if (jeda > 0)  
    {  
        jeda--;  
    }  
    else  
    {  
        getWorld().removeObject(this);  
    }  
}
```

Metode ini akan dipanggil setiap kali tombol 'Act' atau 'Run' ditekan dalam lingkungan Greenfoot. Pada awalnya, `jeda` akan dikurangi satu setiap kali metode ini dipanggil. Jika `jeda` masih lebih besar dari 0, tidak ada tindakan tambahan yang diambil. Namun, jika `jeda` sudah mencapai 0 atau kurang, objek ledakan (`this`) akan dihapus dari dunia menggunakan `getWorld().removeObject(this)`.

Kelas ini secara khusus digunakan untuk membuat ledakan yang mungkin memiliki efek visual atau permainan yang menyertakan ledakan. Dengan `jeda` yang ditentukan, objek ledakan dapat tampil untuk beberapa saat sebelum dihapus dari dunia.

Ledakan 2 :



Penjelasan:

Impor Pustaka

java


```
import greenfoot.*;
```

Baris ini mengimpor kelas-kelas yang diperlukan dari pustaka Greenfoot.

Deklarasi Kelas

```
java  
  
public class Ledakan2 extends Actor
```

Deklarasi kelas `Ledakan2` yang merupakan turunan dari kelas `Actor`. Kelas ini mewakili objek ledakan kedua.

Variabel Instance

```
java  
  
private int jeda = 10;
```

Variabel instance (variabel anggota kelas) yang digunakan untuk mengatur jeda sebelum objek ledakan dihapus dari dunia. Nilai awalnya adalah 10.

Metode `act()`

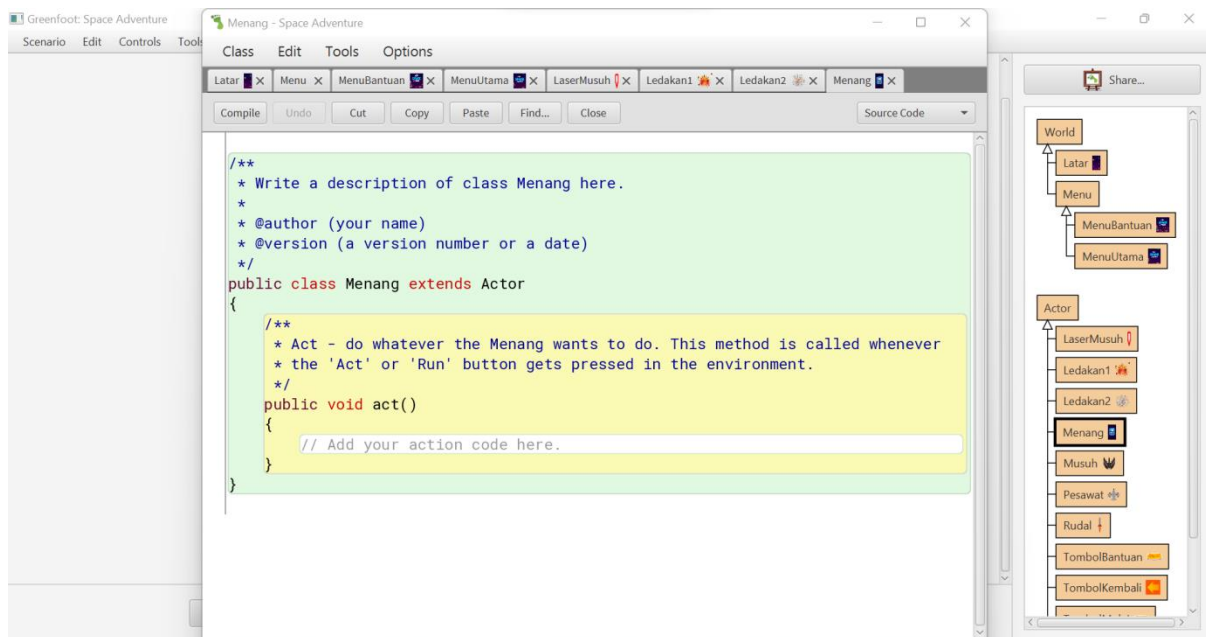
```
java  
  
public void act()  
{  
    if (jeda > 0)  
    {  
        jeda--;  
    }  
    else  
    {  
        getWorld().removeObject(this);  
    }  
}
```

Metode ini akan dipanggil setiap kali tombol 'Act' atau 'Run' ditekan dalam lingkungan Greenfoot. Pada awalnya, `jeda` akan dikurangi satu setiap kali metode ini dipanggil. Jika `jeda` masih lebih besar dari 0, tidak ada tindakan tambahan yang diambil. Namun, jika `jeda`

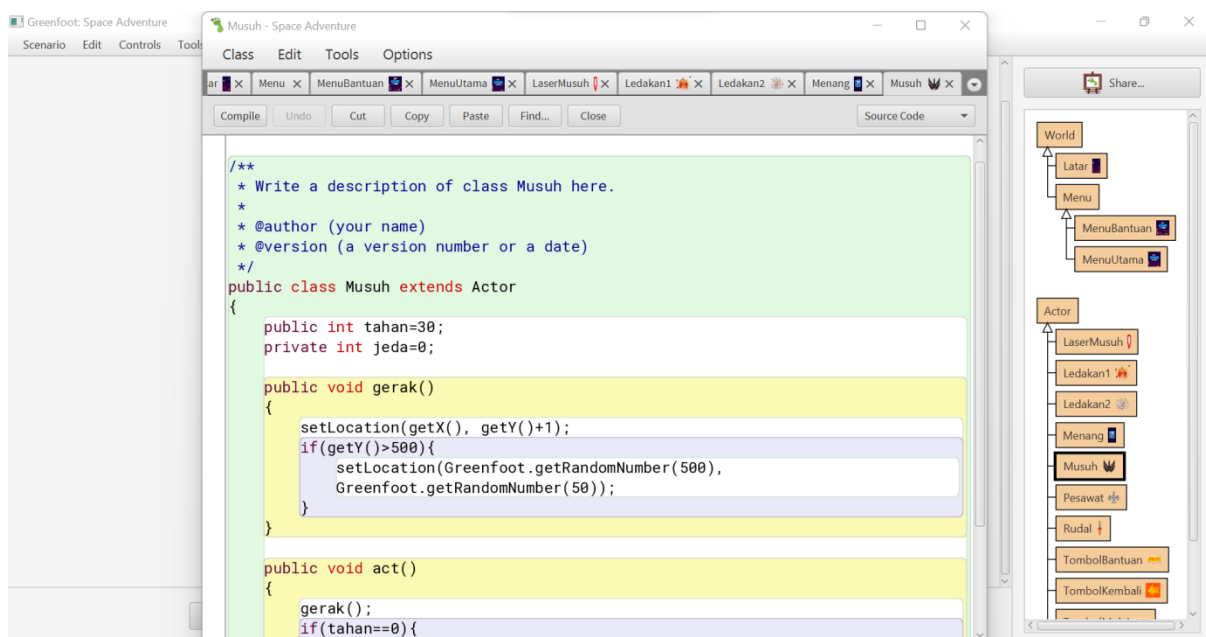
sudah mencapai 0 atau kurang, objek ledakan (`this`) akan dihapus dari dunia menggunakan `getWorld().removeObject(this)`.

Kelas ini, seperti `Ledakan1`, digunakan untuk membuat ledakan dengan jeda yang ditentukan, sehingga objek ledakan dapat tampil untuk beberapa saat sebelum dihapus dari dunia.

Menang :



Musuh :



```

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

```

Penjelasan :

Disini musuh nya akan bergerak secara random dengan kecepatan 1,nanti musuh nya akan otomatis menembakkan laser musuh kearah bawah dengan kecepatan 50,

Pesawat :

The screenshot displays the Greenfoot IDE interface for a project titled 'Pesawat - Space Adventure'. The main editor window shows the source code for the 'Pesawat' class, which extends the 'Actor' class. The code includes comments for documentation and a 'tombol()' method that handles keyboard input to move the spaceship.

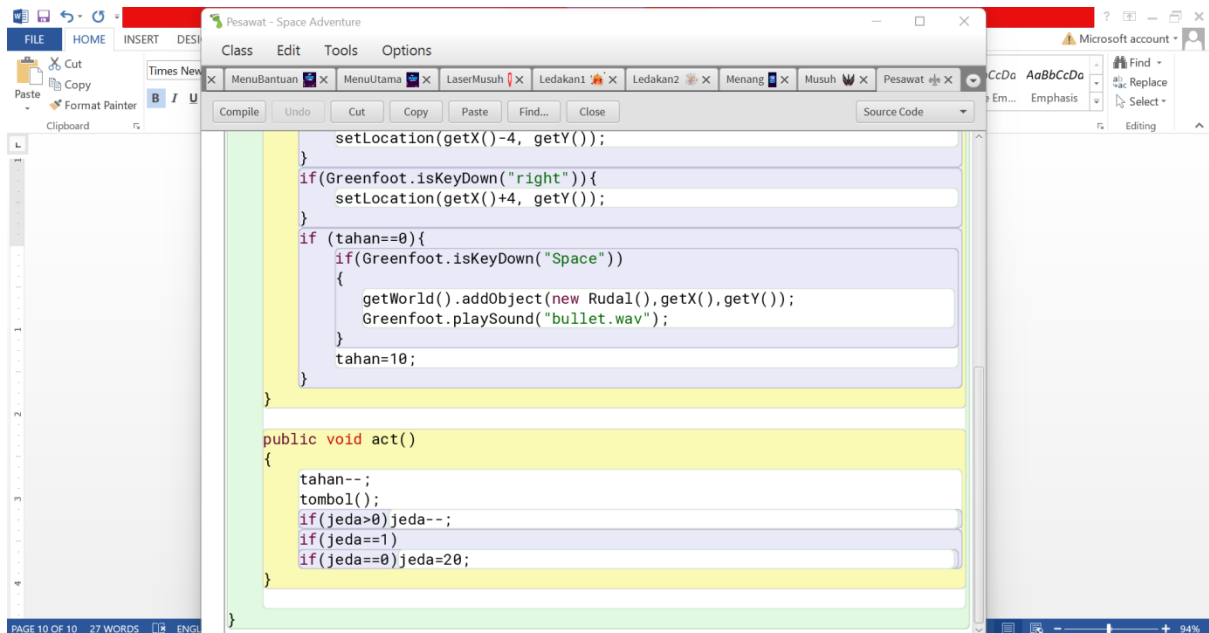
```

/**
 * 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());
        }
    }
}

```

On the right side of the IDE, the 'Project Hierarchy' panel shows the structure of the project. It includes a 'World' package with 'Latar' and 'Menu' objects, and an 'Actor' package containing 'LaserMusuh', 'Ledakan1', 'Ledakan2', 'Menang', 'Musuh', 'Pesawat', 'Rudal', 'TombolBantuan', and 'TombolKembali'.

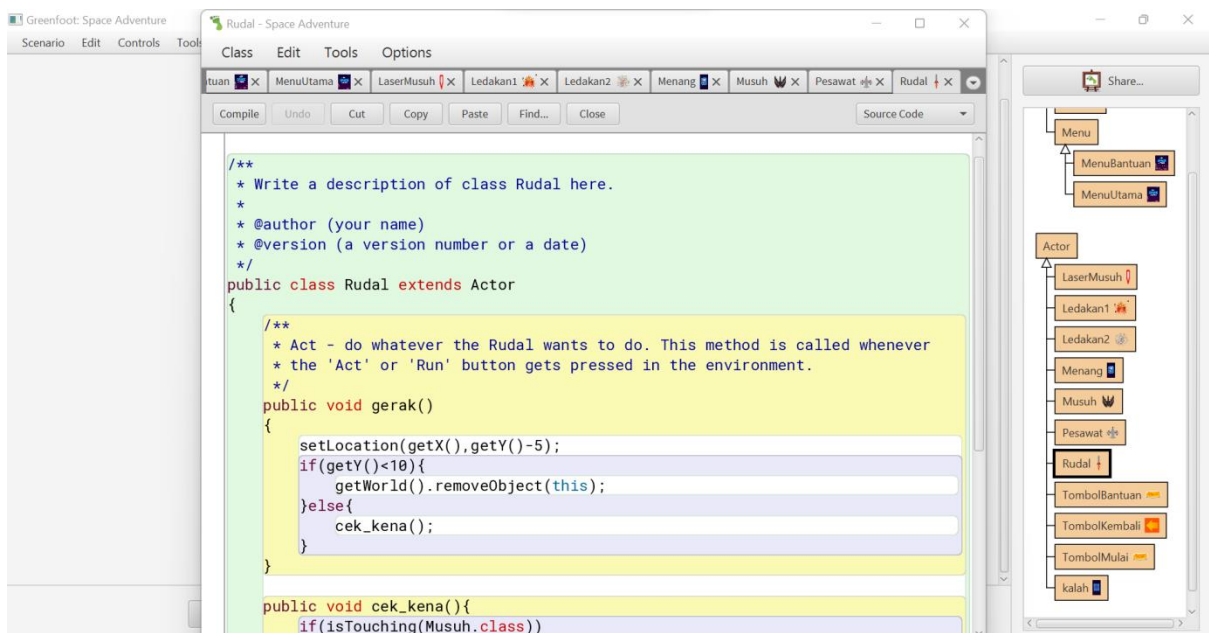


Penjelasan :

Jika kita klik tombol atas, bawah,kiri,kanan maka pesawat akan bergerak sejauh 4

Jika kita menekan atau menahan tombol spasi maka akan mengeluarkan rudal yang di iringi dengan sound bulle wav

Rudal :

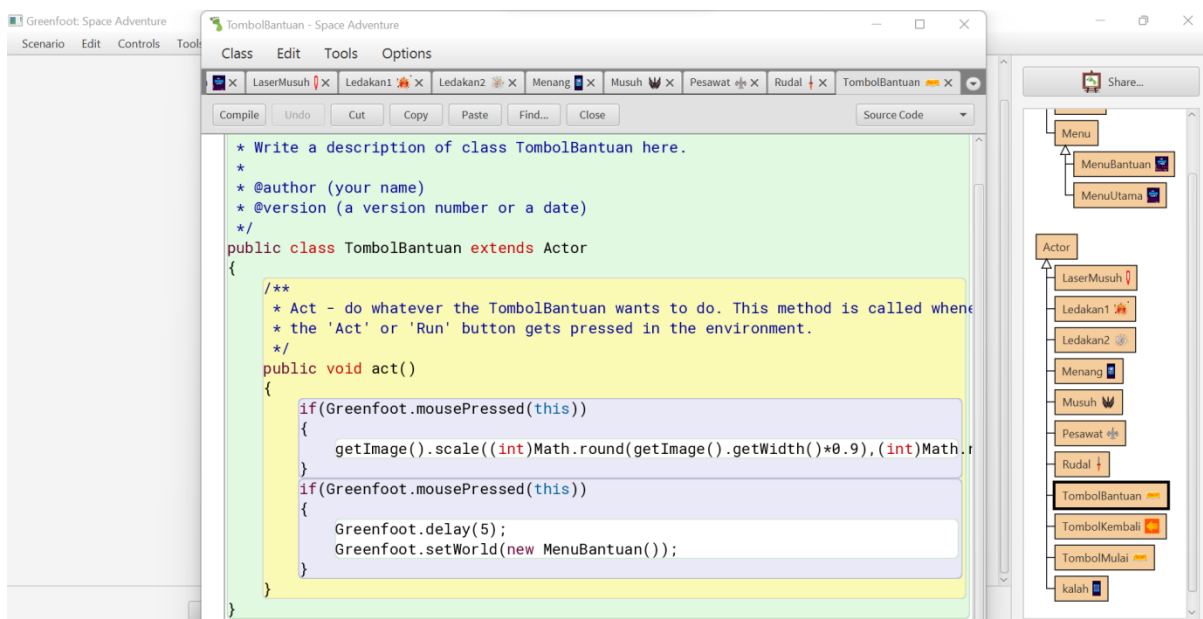




Penjelasan :

rudal nya dia bakal naik keatas dengan kecepatan 5 jika melewati 10 maka rudal nya akan di remove atau di hapus, jika tidak dia akan cek kena yang di maksud cek kena adalah ketika rudal menyentuh musuh dia akan meledak, jika meledak akan menambah score sebanyak 20, dan ketika meledak akan menampilkan suara explosion wav

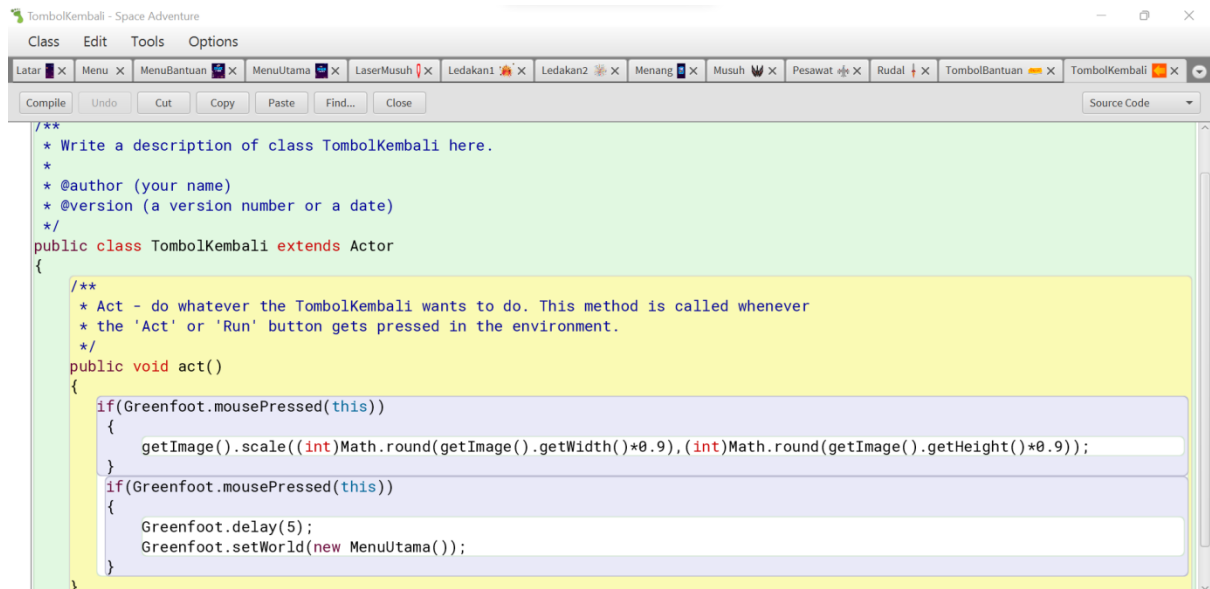
Tombol Bantuan :



Penjelasan :

Pada tombol bantuan di klik akan menampilkan bantuan

Tombol Mulai :

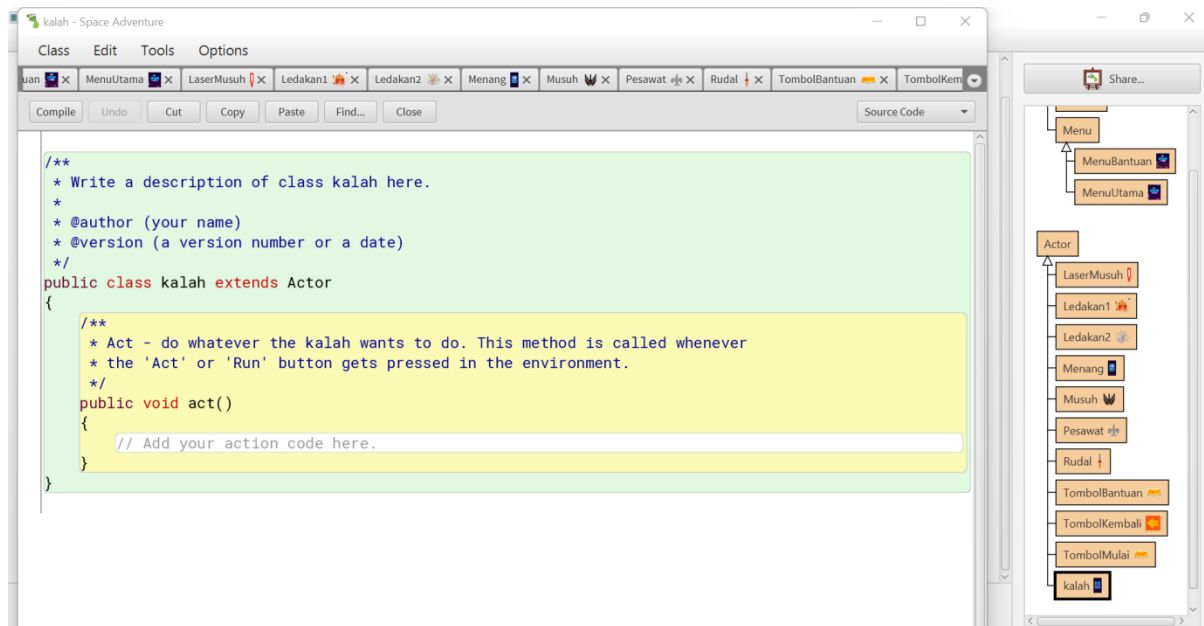


```
/**
 * Write a description of class TombolKembali here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class TombolKembali extends Actor
{
    /**
     * Act - do whatever the TombolKembali wants to do. This method is called whenever
     * the 'Act' or 'Run' button 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 MenuUtama());
        }
    }
}
```

Penjelasan :

Untuk rudal nya dia bakal naik keatas dengan kecepatan 5 jika melewati 10 maka rudal nya akan di remove atau di hapus, jika tidak dia akan cek kena yang di maksud cek kena adalah ketika rudal menyentuh musuh dia akan meledak, jika meledak akan menambah score sebanyak 20, dan ketika meledak akan menampilkan suara explosion wav

Kalah :



Penjelasan :

objek ini akan muncul jika permainan kalah.