

# **MODUL PRAKTIKUM**

**Dasar-Dasar Pemrograman**



**Penyusun : Eko Verianto, S.Kom., M.Cs.**

# **DAFTAR ISI**

## **BAB 1**

PERTEMUAN 1 Pengantar Perkuliahan .	<b>Error! Bookmark not defined.</b>
PERTEMUAN 2 Pengenalan Alice .....	8

# PENGANTAR

# PERKULIAHAN

## Tujuan Pembelajaran:

1. Menjelaskan materi perkuliahan untuk satu semester
2. Menjelaskan tujuan pembelajaran akhir
3. Menjelaskan mengenai Oracle Academy sebagai kurikulum penunjang kebutuhan pembelajaran
4. Menjelaskan peta pembelajaran
5. Menjelaskan program aplikasi yang digunakan selama perkuliahan
6. Pengenalan Alice 3

## Materi Perkuliahan

1. Pengantar Perkuliahan
2. Memulai dengan Alice
3. Memulai dengan Greenfoot
4. Dasar Pemrograman Java
5. Struktur dalam Pemrograman Java
6. Array dan Exception
7. Class pada Java

## Tujuan Pembelajaran Akhir

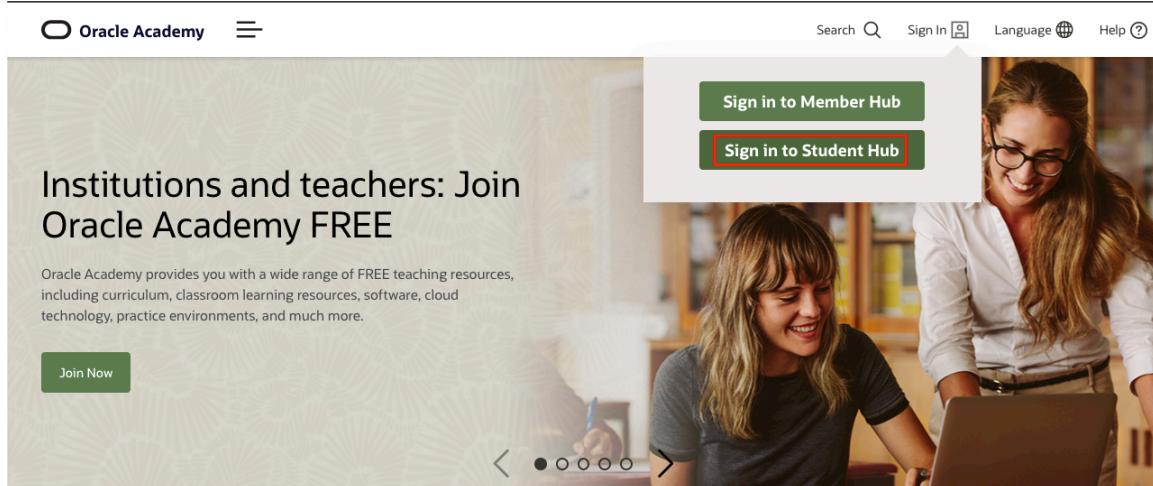
Mahasiswa akan mempelajari mengenai elemen dasar pada program komputer dan juga bahasa pemrograman java untuk memulai menulis kode program pada aplikasi yang akan dikembangkan. Di akhir perkuliahan diharapkan mahasiswa dapat:

1. Membuat animasi dan permainan sederhana
2. Mendemonstrasikan pengetahuan yang didapatkan terkait teknologi java dan juga bahasa pemrograman java
3. Menggunakan bahasa pemrograman java untuk membuat aplikasi
4. Mengintegrasikan struktur percabangan, perulangan dan teknik lanjutan lainnya untuk membuat aplikasi

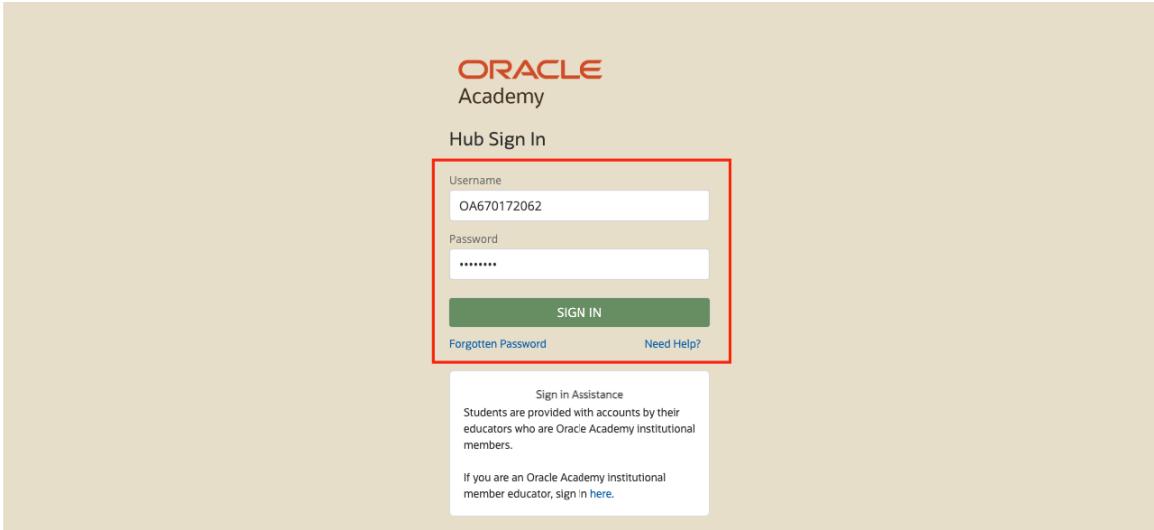
## Kurikulum Oracle Academy

Selama satu semester ini, kita akan menggunakan kurikulum dari oracle academy untuk menunjang proses pembelajaran kita. Anda dapat mengikuti langkah-langkah berikut ini untuk membuka akun oracle academy anda.

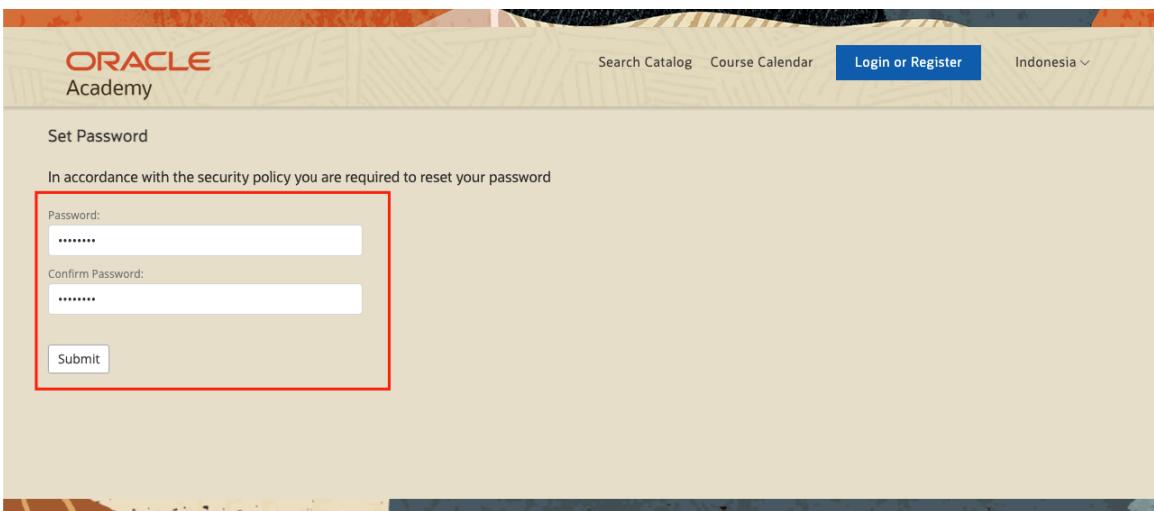
1. Silahkan buka tautan berikut ini <https://academy.oracle.com/>
2. Pilih menu sign in kemudian pilih sign in to student hub



3. Masukan username dan password sesuai dengan akun anda (cek pada daftar)



4. Setelah berhasil sign in, anda diminta untuk mengubah password anda, silahkan lakukan perubahan pada password (gunakan password yang biasa digunakan)



5. Setelah selesai mengubah password, anda akan diarahkan ke halaman utama dashboard. Pada halaman ini terdapat satu channel anda dengan nama "praktikum dasar-dasar pemrograman". Klik pada channel dengan nama "praktikum dasar-dasar pemrograman" tersebut.

The screenshot shows the Oracle Academy homepage. At the top right, there is a user profile for 'Eko Verianto 72220000' labeled 'Learner'. The main navigation bar includes 'Home', 'Career Center', 'My Learning', and 'My Reports'. A banner on the left says 'Explore careers!' and 'Career Center'. On the right, there is a 'More Help' sidebar with links for 'Redeem Certification Exam Vouchers Demonstration' and 'Member Hub Learner Guide'. Below the banner, under 'My Channels', there is a list with one item: 'Praktikum Dasar-Dasar Pemrograman' with a 'Details' button. This list is highlighted with a red border.

6. Setelah memilih channel, anda akan diarahkan ke halaman learning path. Pada bagian ini anda memiliki satu learning path dengan nama "JF Java Fundamentals Learner – English". Silahkan klik pada learning path dengan nama "JF Java Fundamentals Learner – English" tersebut

The screenshot shows the Oracle Academy homepage again. The 'Learning Paths' section is highlighted with a red border. It displays a single learning path titled 'Java Fundamentals Learner' with a small thumbnail image. Below the thumbnail, it says 'Learning Path - Full Course' and 'JF Java Fundamentals Learner - English'. There is also a 'Details' button.

7. Selanjutnya anda akan masuk ke halaman course yang berisi materi pembelajaran, soal kuis serta soal TTS dan TAS. Pada halaman ini anda dapat melihat dan mengunduh materi.

The screenshot shows the Oracle Academy homepage. At the top right, there is a user profile for "Eko Verianto 72220000" with the status "Learner". The main navigation bar includes links for "Home", "Career Center", "My Learning", and "My Reports". Below the navigation, there is a vertical list of course sections: "Section 0 - Course Resources", "Section 1 - Introduction", "Section 2 - Alice 3", and "Section 3 - Greenfoot".

- a. Untuk melihat materi, anda dapat memilih course terlebih dahulu, kemudian klik play

The image contains three screenshots of the Oracle Academy interface:

- Screenshot 1:** Shows the "Section 1 - Introduction" section selected. A specific video thumbnail titled "JF 1-1: Introduction" is highlighted with a red box.
- Screenshot 2:** Shows the "JF 1-1: Introduction" video details page. The "Play" button is highlighted with a red box.
- Screenshot 3:** Shows the video player interface for "JF 1-1: Introduction". The video title "JF 1-1: Introduction" is visible, along with the "Online" status and "Credits (HH:MM):00:48".

b. Untuk mengunduh materi, anda dapat memilih course terlebih dahulu, kemudian klik nama dokumen pada bagian reference material

The screenshot shows the Oracle Academy interface. At the top, there's a navigation bar with links for Home, Career Center, My Learning, and My Reports. On the right, a user profile for 'Eko Verianto 72220000 Learner' is visible. Below the navigation, a course card for 'JF 1-1: Introduction' is displayed. The card includes a thumbnail of a person riding a motorcycle, the title 'Java Fundamentals Learner', status 'Online', credits '(HH:MM):00:48', and a 'Play' button. To the right of the card, there's a 'Details' section and another card for 'JF 1-1: Introduction' with a red box around the PDF download link 'JF 1-1: Introduction (SG PDF) (466.8 KB)'.

c. Untuk mengerjakan kuis, TTS dan TAS, anda dapat memilihnya *quiz*, *midterm exam* atau *final exam* terlebih dahulu, kemudian klik play

The screenshot shows the Oracle Academy interface with a list of quizzes and an exam detail page. At the top, there's a navigation bar with links for Home, Career Center, My Learning, and My Reports. Below the navigation, three cards are listed: 'JF Section 2 Quiz 1 - L1-L7' (Exam, Credits (HH:MM):00:30), 'JF 2-8: IF and WHILE Control Structures' (Online, Credits (HH:MM):00:36), and 'JF 2-9: Expressions' (Online, Credits (HH:MM):00:36). The first card has a red box around its 'Play' button. Below this list, a 'Channel > Learning Path Details' section is shown for 'JF Section 2 Quiz 1 - L1-L7'. This section includes a thumbnail of a person riding a motorcycle, the title 'Java Fundamentals Learner', status 'Exam', credits '(HH:MM):00:30', and a 'Play' button. To the right of this, there's a 'Details' section and another card for 'JF Section 2 Quiz 1 - L1-L7' with a red box around the PDF download link 'JF Section 2 Quiz 1 - L1-L7 (SG PDF) (466.8 KB)'.

## **Peta Pembelajaran**

1. Pengantar Perkuliahan, Pengenalan Alice 3, Menambahkan & Memposisikan Objek, Procedure & Argument
2. Rotation & Randomization, Deklarasi Procedure, Control Statement, Function
3. Kuis, Struktur Kontrol IF & WHILE, Ekspresi, Variabel, Kontrol Keyboard
4. Membuat animasi sederhana, Variabel & Tipe Data, Class & Method, Quiz
5. Pengenalan GreenFoot, Method, Variabel & Parameter, Source Code & Documentation
6. Mengembangkan dan Menguji Aplikasi, Randomization & Memahami Notasi Dot, Konstruktor, Quiz
7. Mendefinisikan Metode, Suara dan Kontrol Keyboard, World Animation, Abstraction, Perulangan, Variabel dan Array, Quiz
8. Midterm Exam
9. Pengenalan Eclipse, Driver Class & Object Class, Tipe Data & Operator, String, Quiz
10. Scanner & Conditional Statement, Control Statement, Quiz
11. Array, Handling Error, Quiz
12. Classes, Object, & Method, Static Modifier
13. Nested Classes, Inheritance
14. Polymorphism, Quiz

## **Program Aplikasi**



# PENGENALAN ALICE

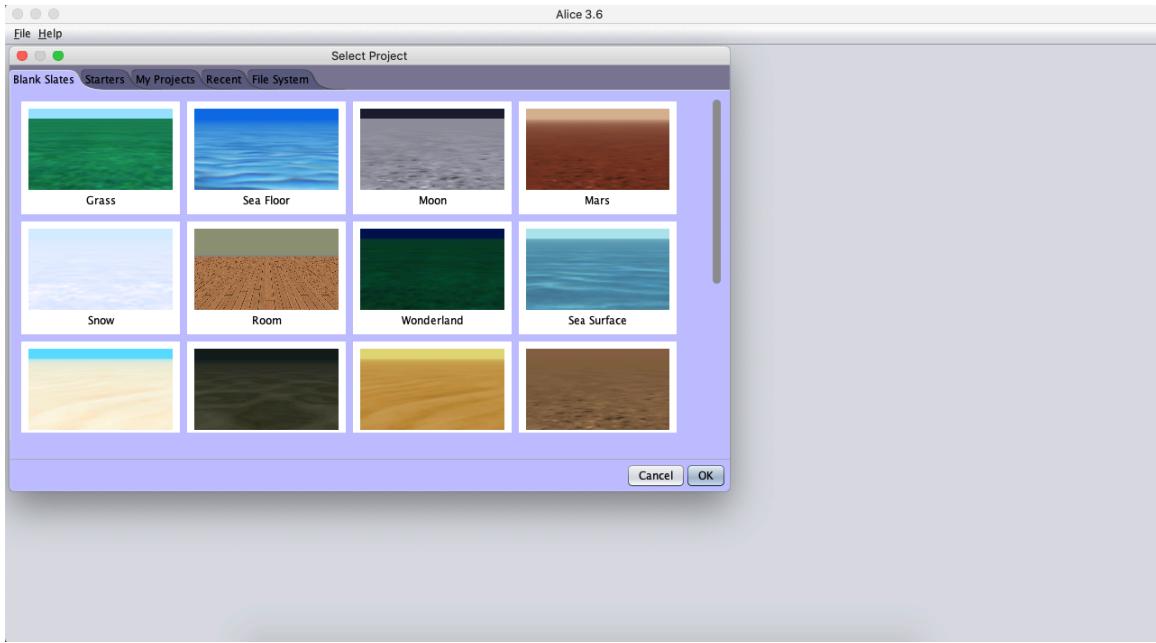
## Tujuan Pembelajaran:

1. Menjelaskan materi perkuliahan untuk satu semester
2. Menjelaskan tujuan pembelajaran akhir
3. Menjelaskan mengenai Oracle Academy sebagai kurikulum penunjang kebutuhan pembelajaran
4. Menjelaskan peta pembelajaran
5. Menjelaskan program aplikasi yang digunakan selama perkuliahan

## Pengenalan Alice 3

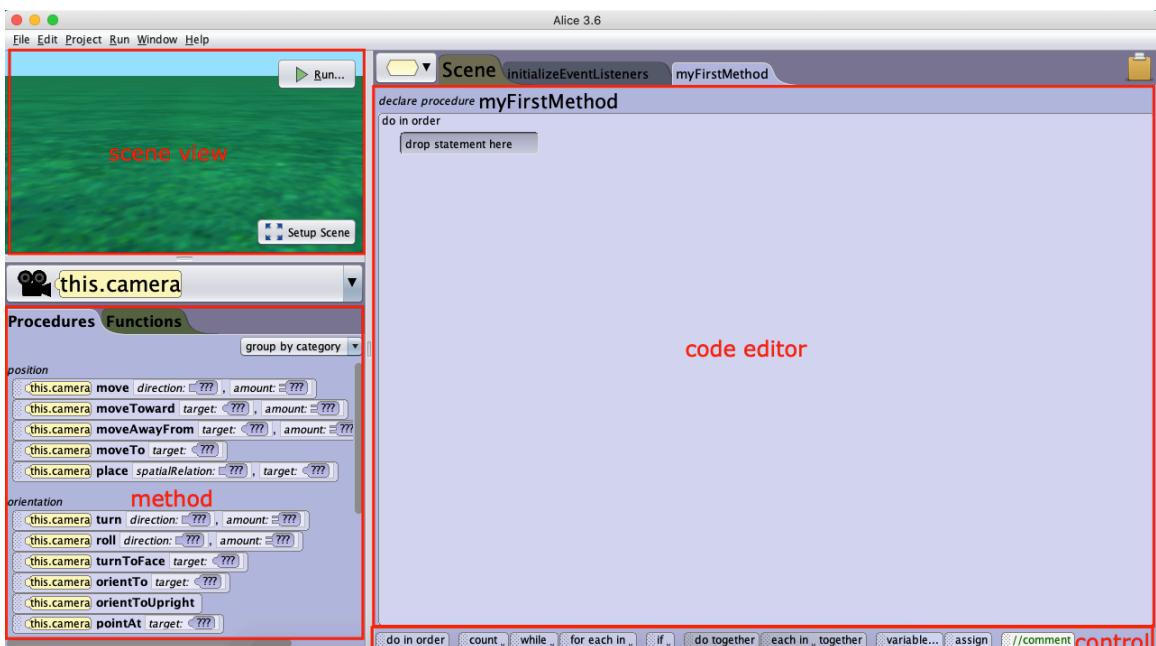
Alice 3 (tiga) merupakan pengembangan dari versi sebelumnya, ini merupakan pemrograman 3D dan freeware dari bahasa pemrograman berbasis objek yang digunakan untuk membuat prototipe environment dalam membangun dunia virtual yang sederhana. Alice juga digunakan untuk media pembelajaran pemrograman berorientasi objek. Lakukan percobaan berikut ini:

1. Silahkan buka aplikasi Alice 3, jika berhasil maka tampilan awalnya seperti berikut ini:

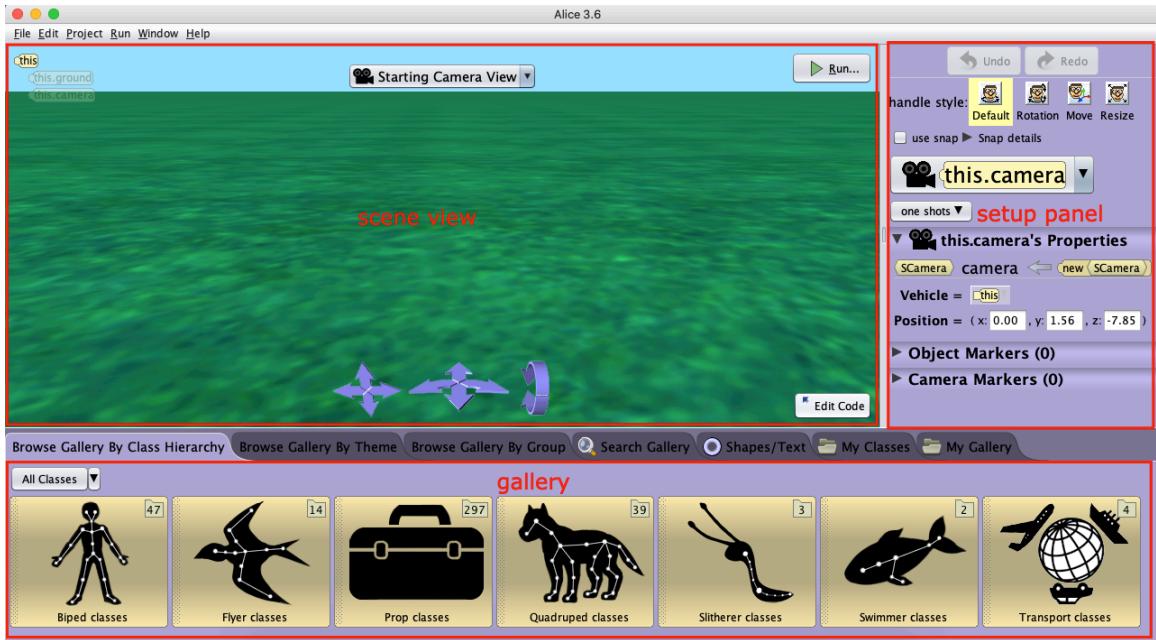


Pada bagian ini terdapat beberapa latar belakang yang dapat anda gunakan, selanjutnya pilih salah satu latar belakang yang anda inginkan kemudian klik ok.

## 2. Tampilan Editor

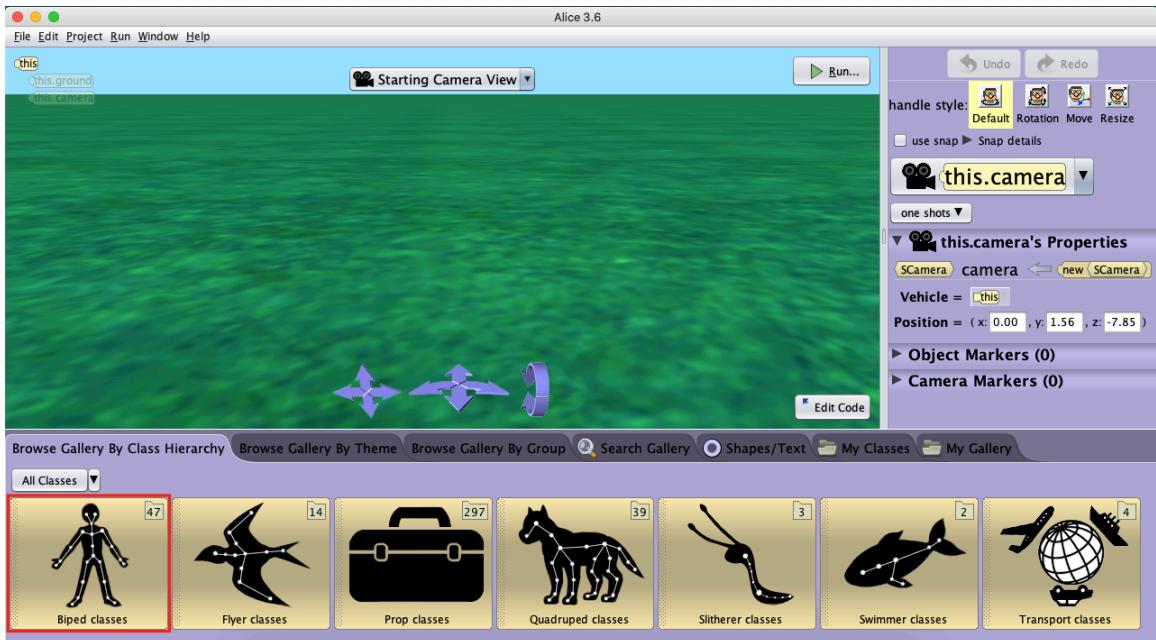


## 3. Tampilan Setup Scene



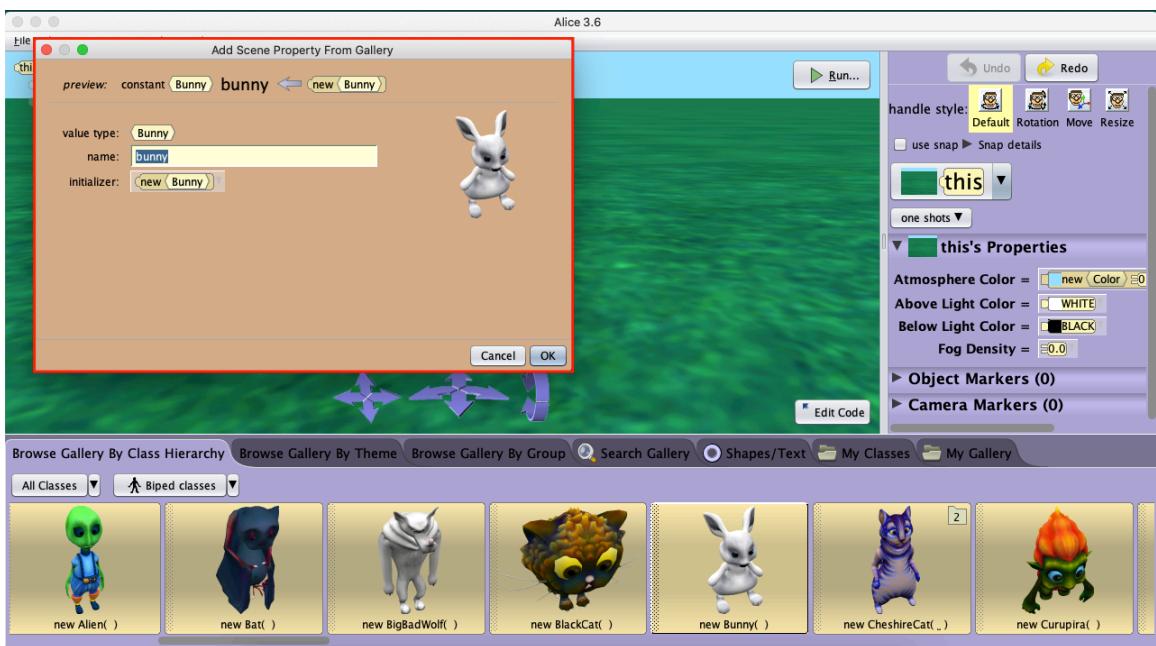
#### 4. Menambahkan Objek

Pilih salah satu kategori class pada galeri yang disediakan, misalkan saja kita memilih biped, kemudian klik class yang diinginkan misalkan saja bunny, atau drag and drop class bunny ke bagian scene view





5. Jika berhasil maka akan ada jendela pop up untuk mengatur properti pada class



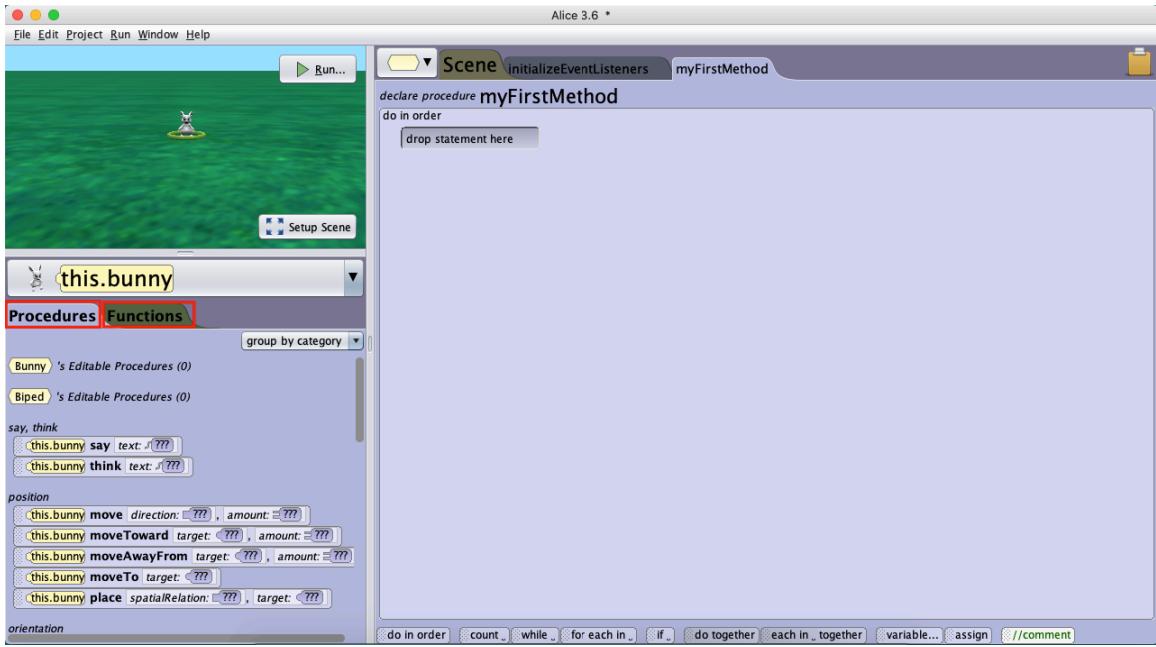
6. Jika sudah selesai mengelola properti class maka akan tercipta sebuah objek baru pada scene view



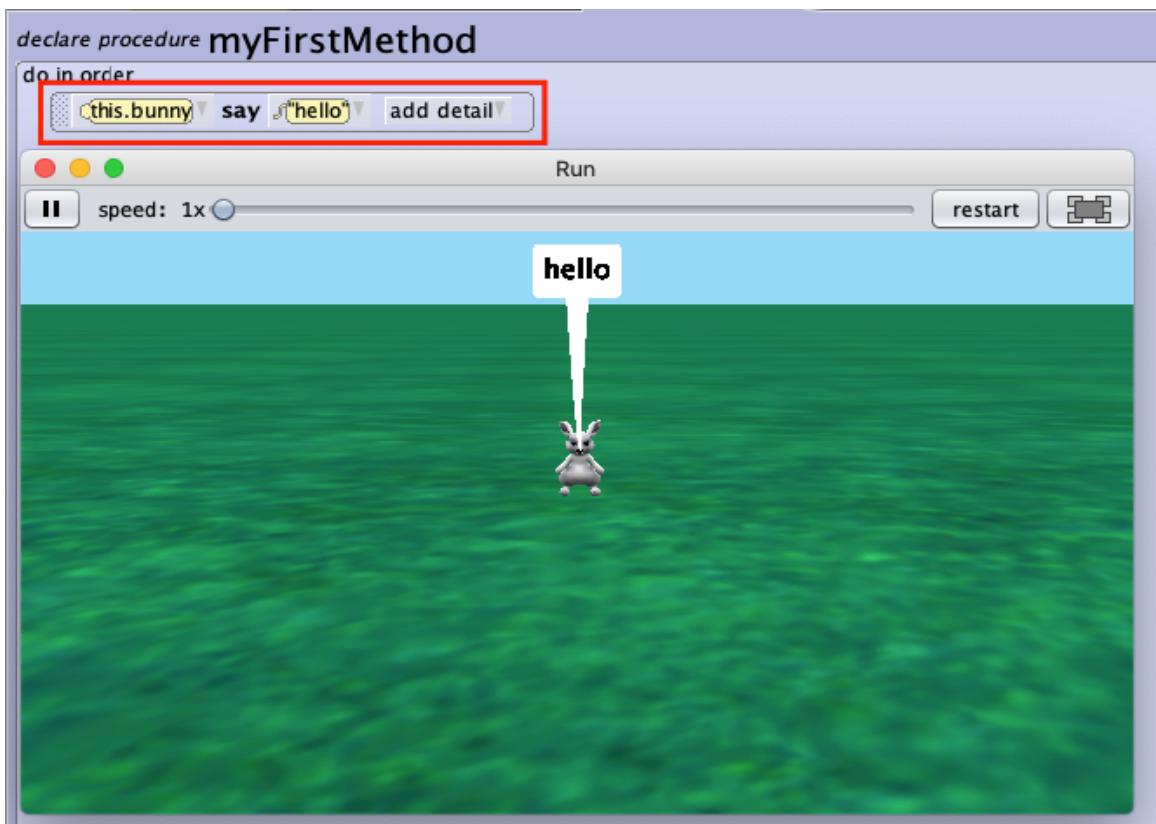
7. Kembali ke code editor dengan klik edit code untuk menambahkan beberapa instruksi



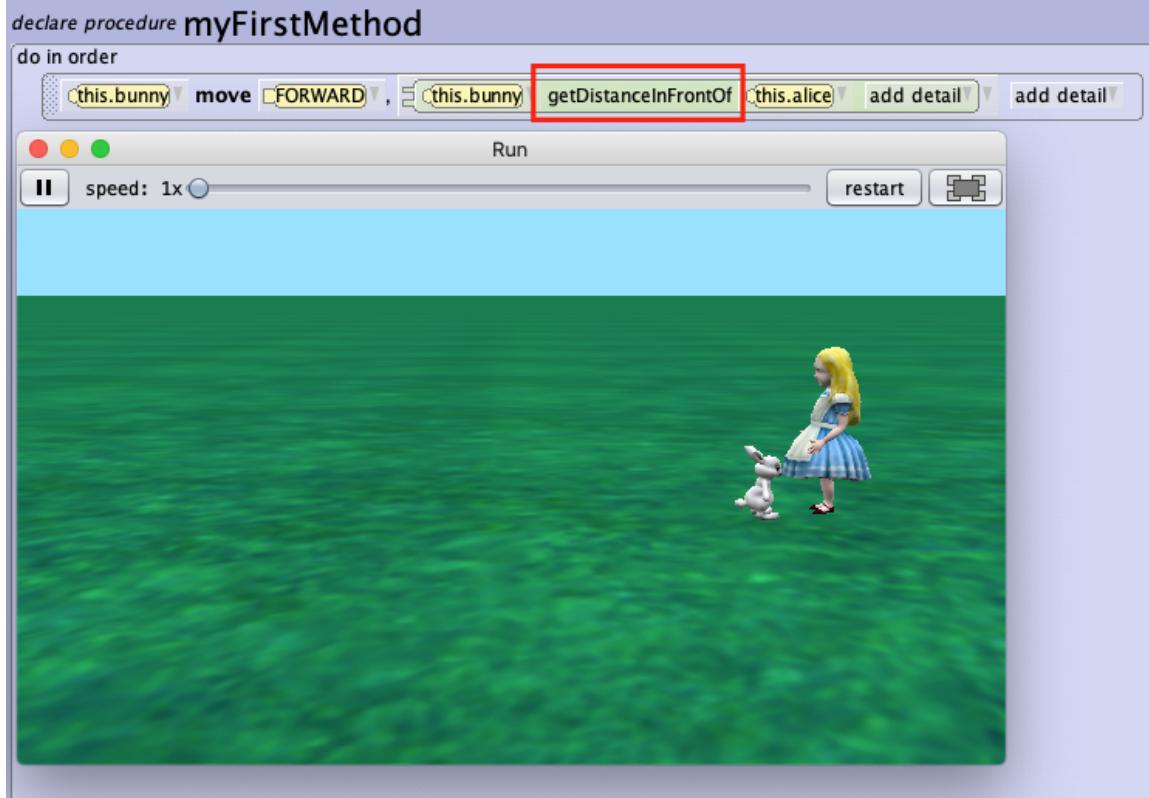
8. Terdapat dua metode yang dapat digunakan yang tersedia pada panel metode, diantaranya adalah procedures dan functions. Procedures merupakan metode yang tidak dapat mengembalikan nilai, sedangkan functions adalah metode yang mengembalikan nilai.



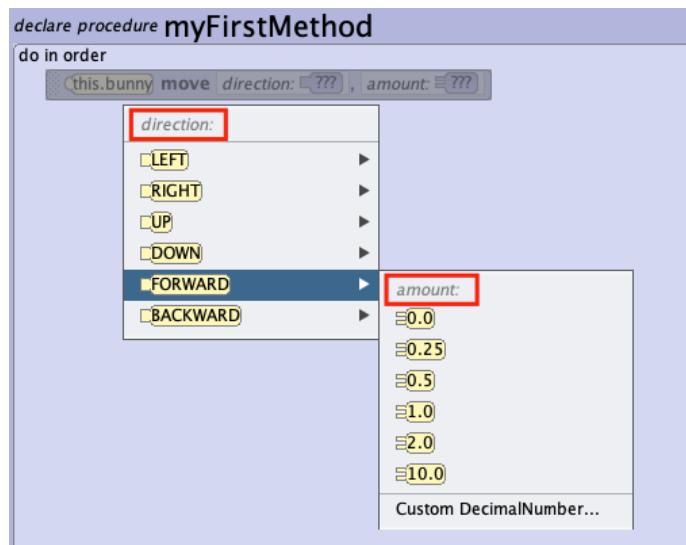
9. Penggunaan procedure akan langsung dieksekusi sesuai dengan instruksi dari metode tersebut, sebagai contoh jika procedure say ditambahkan pada code editor, ketika program dijalankan maka instruksi dari procedure tersebut langsung dieksekusi



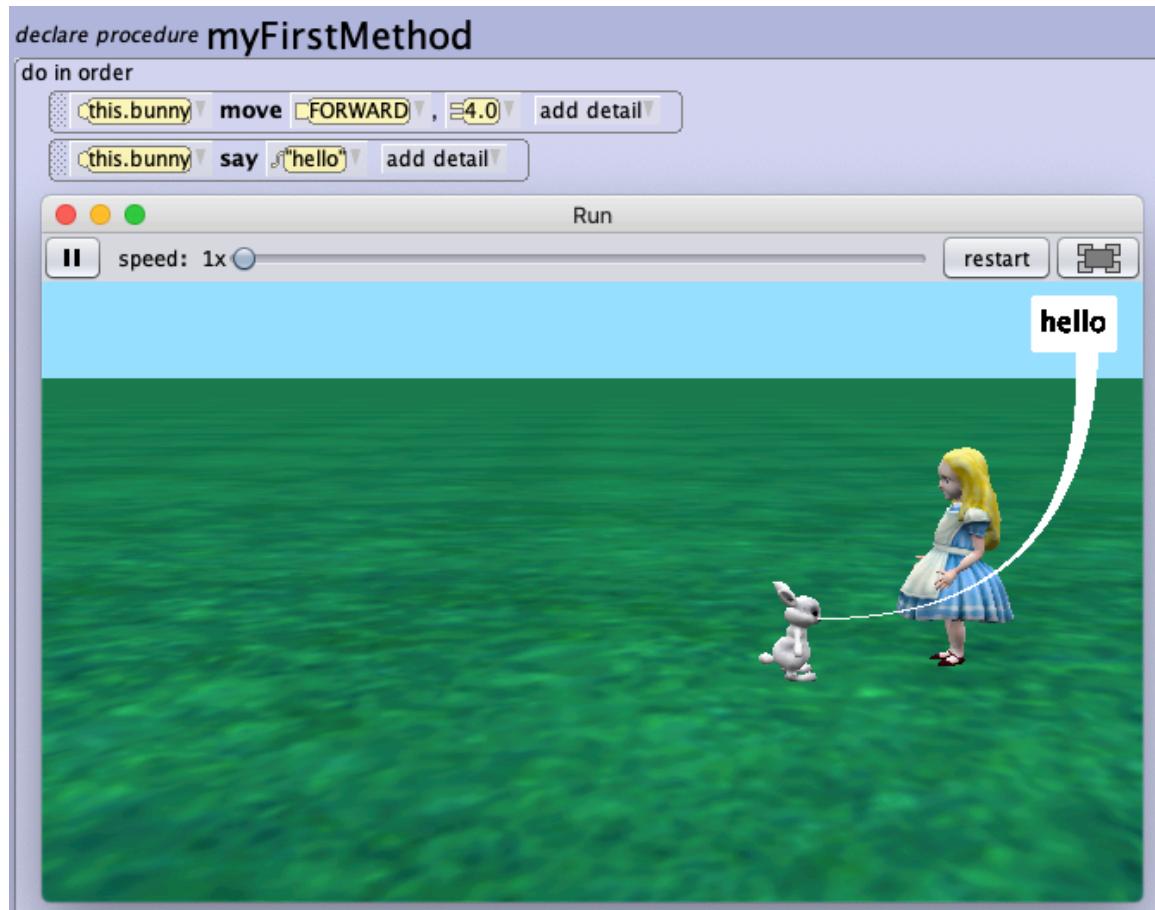
10. Penggunaan function tidak dapat langsung ditambahkan pada code editor, anda harus menyisipkan function pada sebuah procedure. Hal ini dikarenakan sifat dari function yang mengembalikan nilai



11. Pada alice terdapat argumen, dalam pemrograman argumen merupakan nilai yang diberikan. Pada alice terdapat beberapa tipe argumen diantaranya adalah direction, amount, duration, text.



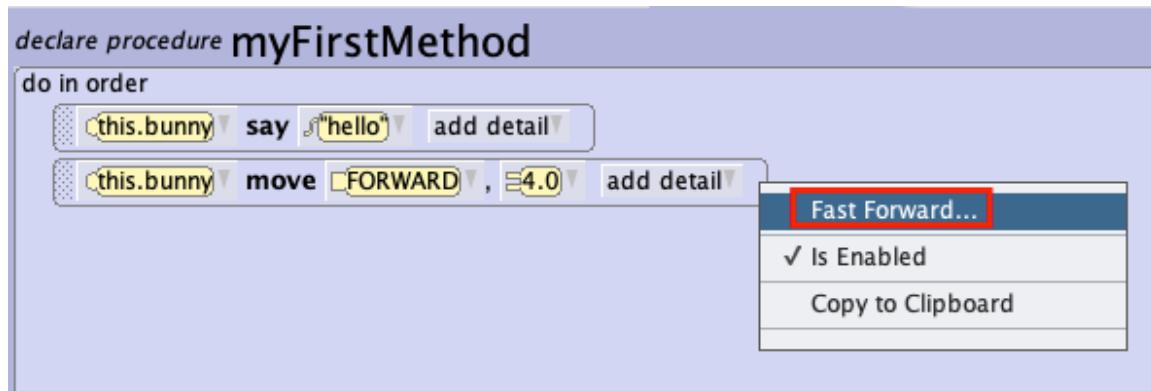
12. Menambahkan lebih dari satu instruksi secara terarah dengan menggunakan kontrol do in order



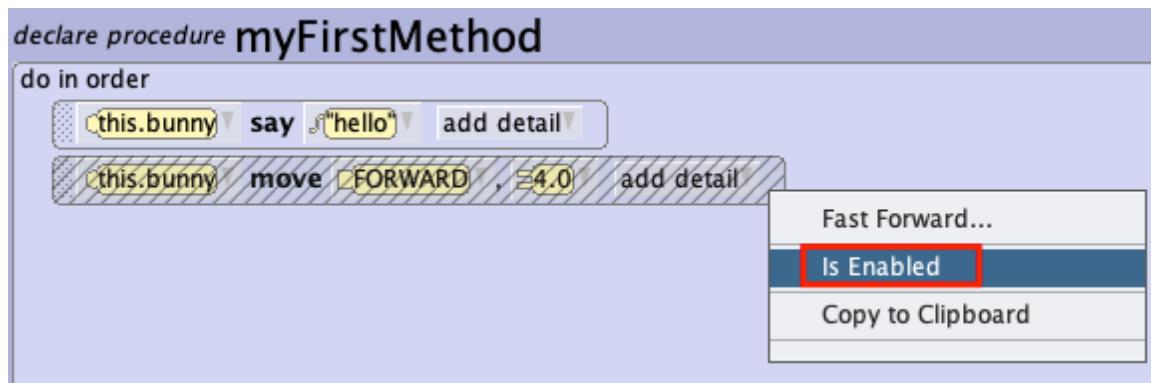
13. Mengubah urutan procedure pada code editor



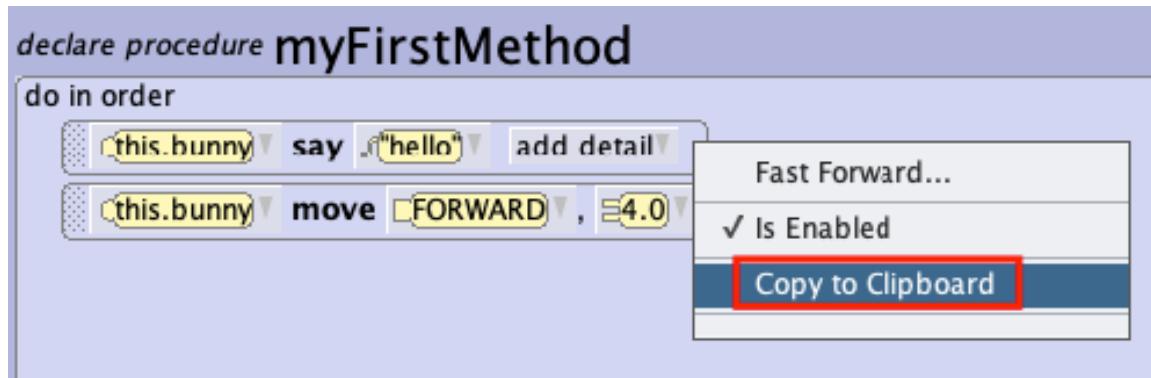
14. Menjalankan animasi dari instruksi atau procedure tertentu



15. Enable dan disable instruksi atau procedure

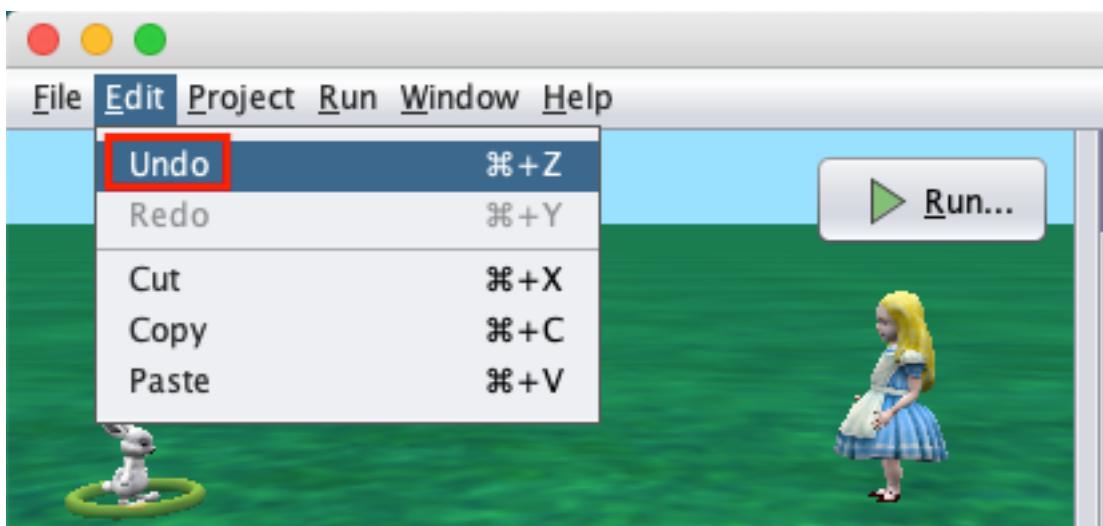


16. Menyalin instruksi pada clipboard

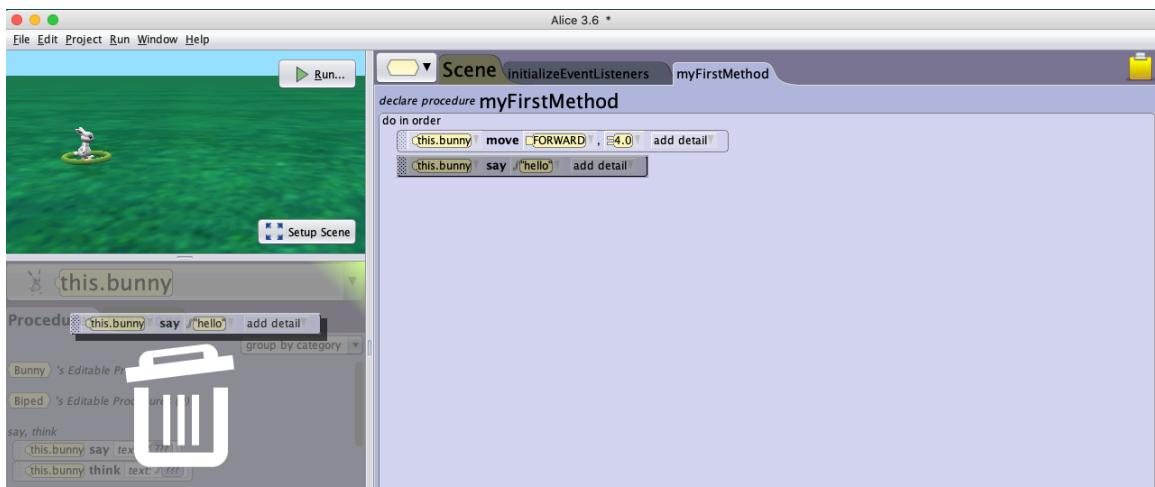


17. Membatalkan perintah

Instruksi ataupun aktivitas sebelumnya dapat dibatalkan dengan undo



### 18. Menghapus instruksi



### 19. Menambahkan Komentar



### 20. Melakukan testing dan debugging

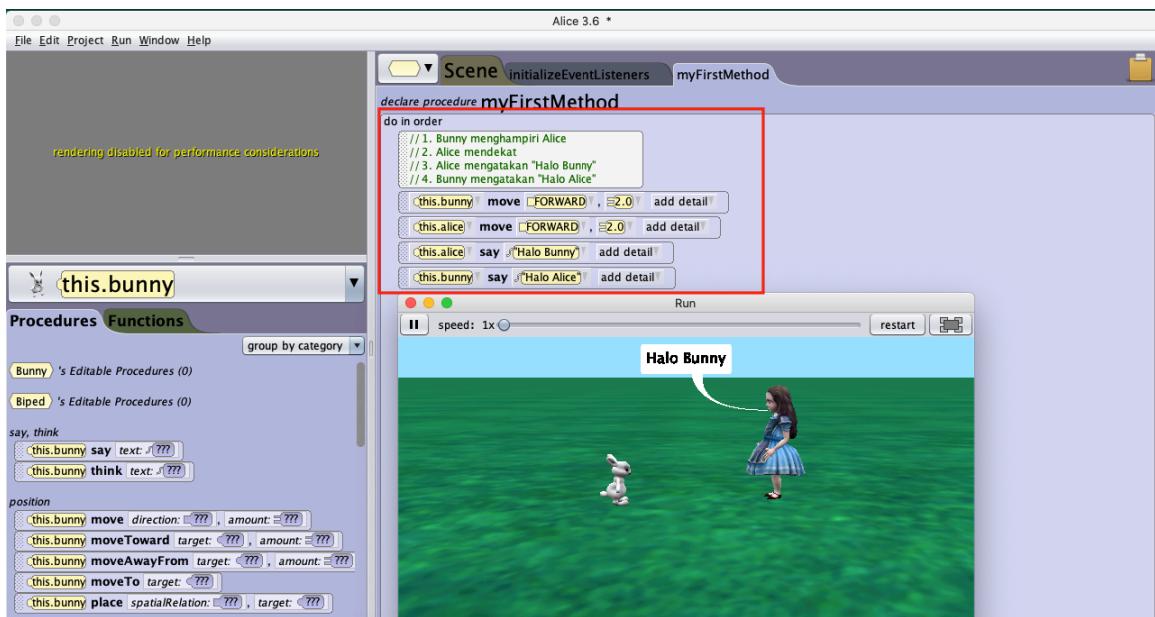
Beberapa teknik yang dapat digunakan diantaranya sesuaikan argumen, sesuaikan ekspresi matematika, perbaiki metode atau instruksi, dan mengatasi masalah lainnya

21. Simpan proyek dengan klik file kemudian klik save, tentukan nama proyek dan direktori tempat anda menyimpan proyek ini, selanjutnya klik save

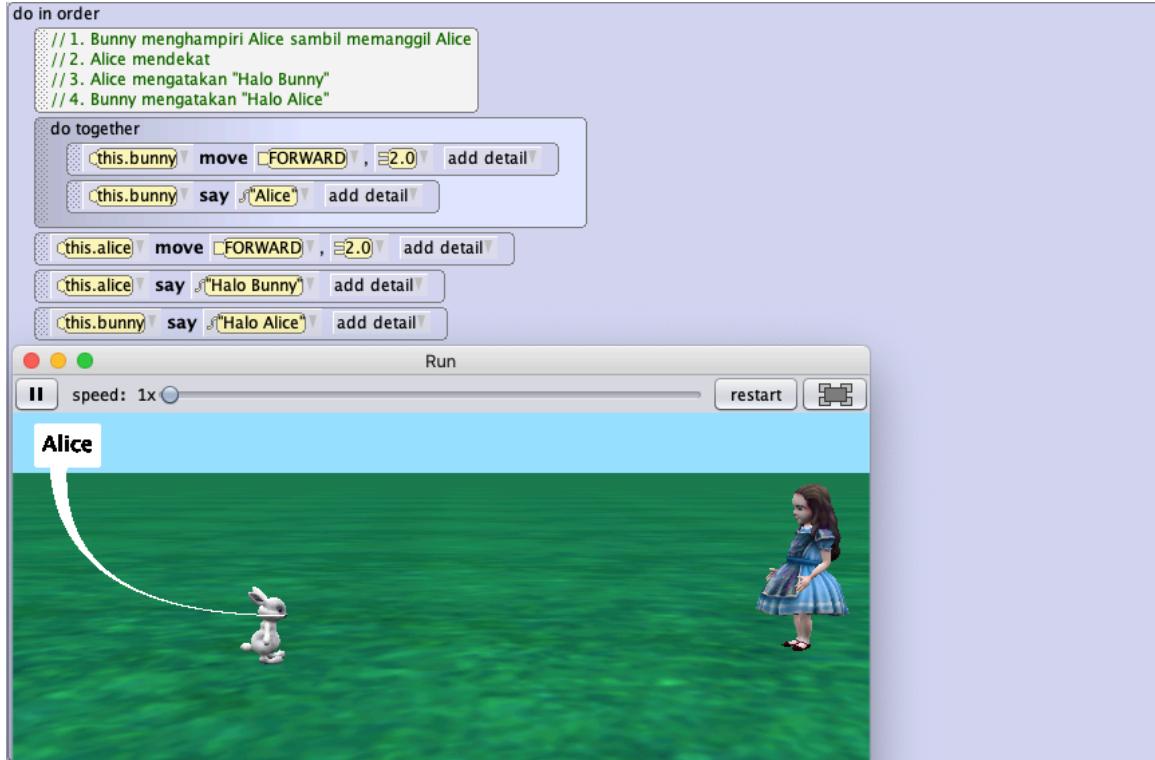


22. Proses pengembangan program dapat dilakukan dengan menggunakan pendekatan top-down, artinya programmer bekerja dengan tujuan. Langkah untuk menggunakan pendekatan top-down adalah dengan membuat dokumen langkah demi langkah dalam bentuk teks (storyboard: Modul JF 2.5 SG). Berikut ini merupakan contoh pembuatan storyboard pada alice 3 dalam bentuk teks.

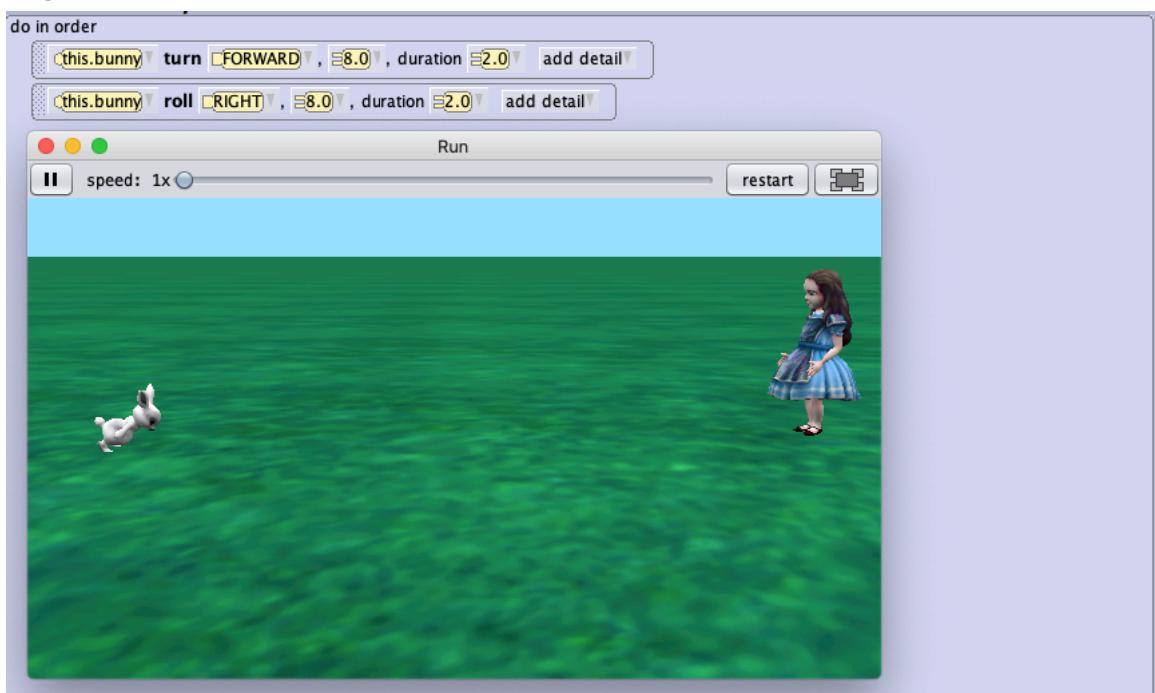
- 1) Bunny menghampiri alice
- 2) Alice mendekat
- 3) Alice mengatakan "Halo Bunny"
- 4) Bunny mengatakan "Halo Alice"



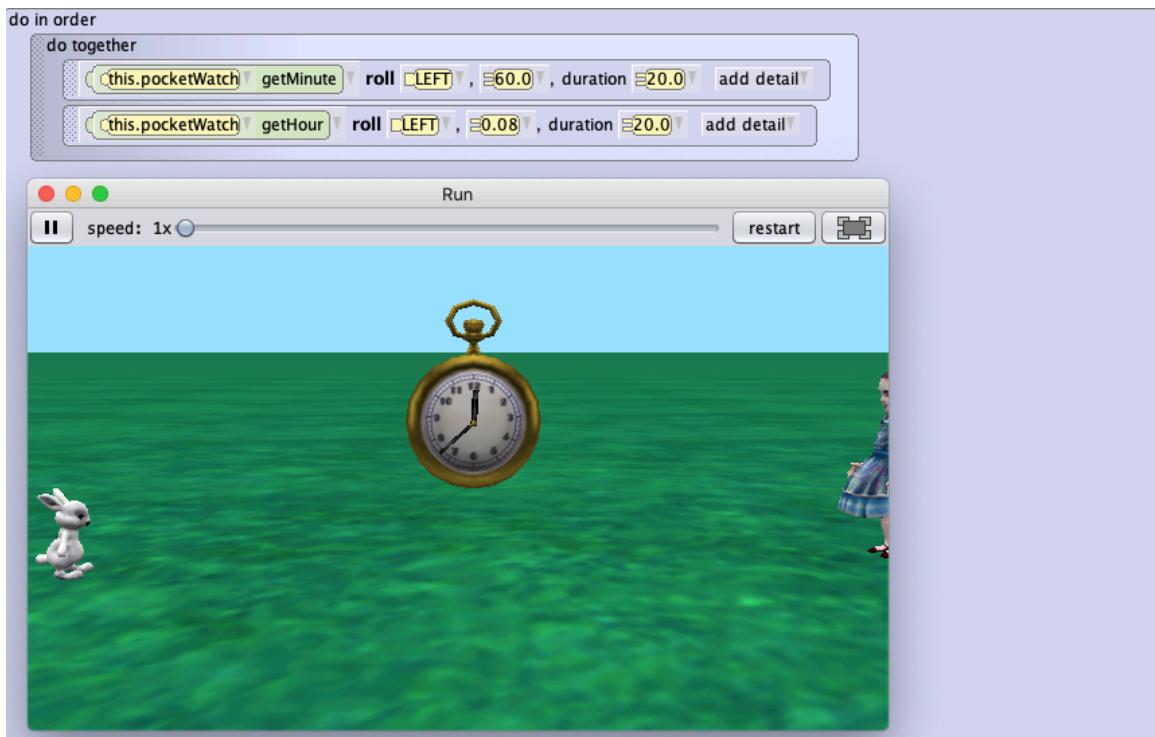
Menggunakan control do together pada program alice



23. Perbedaan procedure turn dan roll, rotasi objek pada procedure turn ke arah kiri, kanan, depan dan belakang dari titik pusat objek, sedangkan roll objek bergulir ke arah kiri dan kanan dari titik pusat objek.



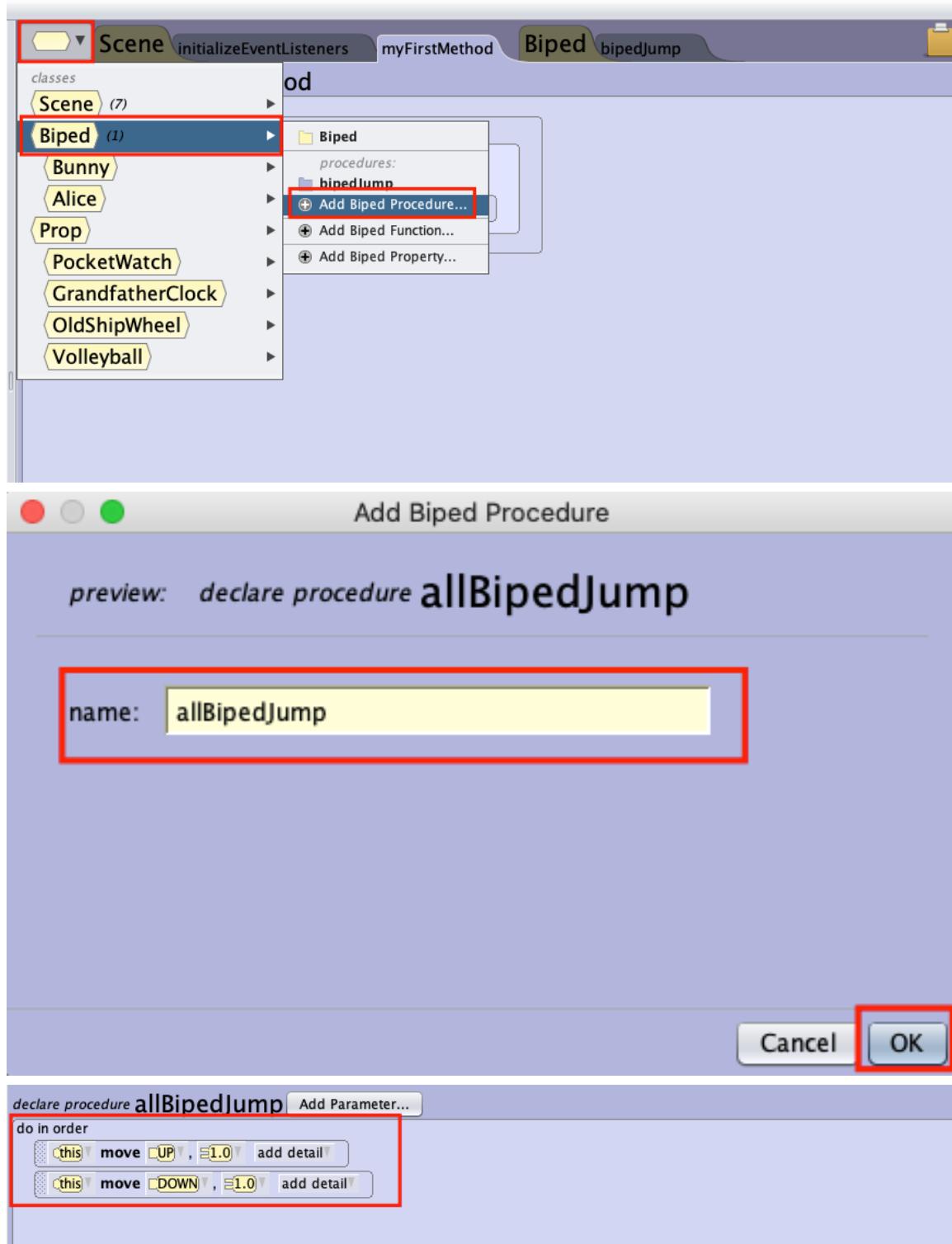
## 24. Mengendalikan sub-part pada objek

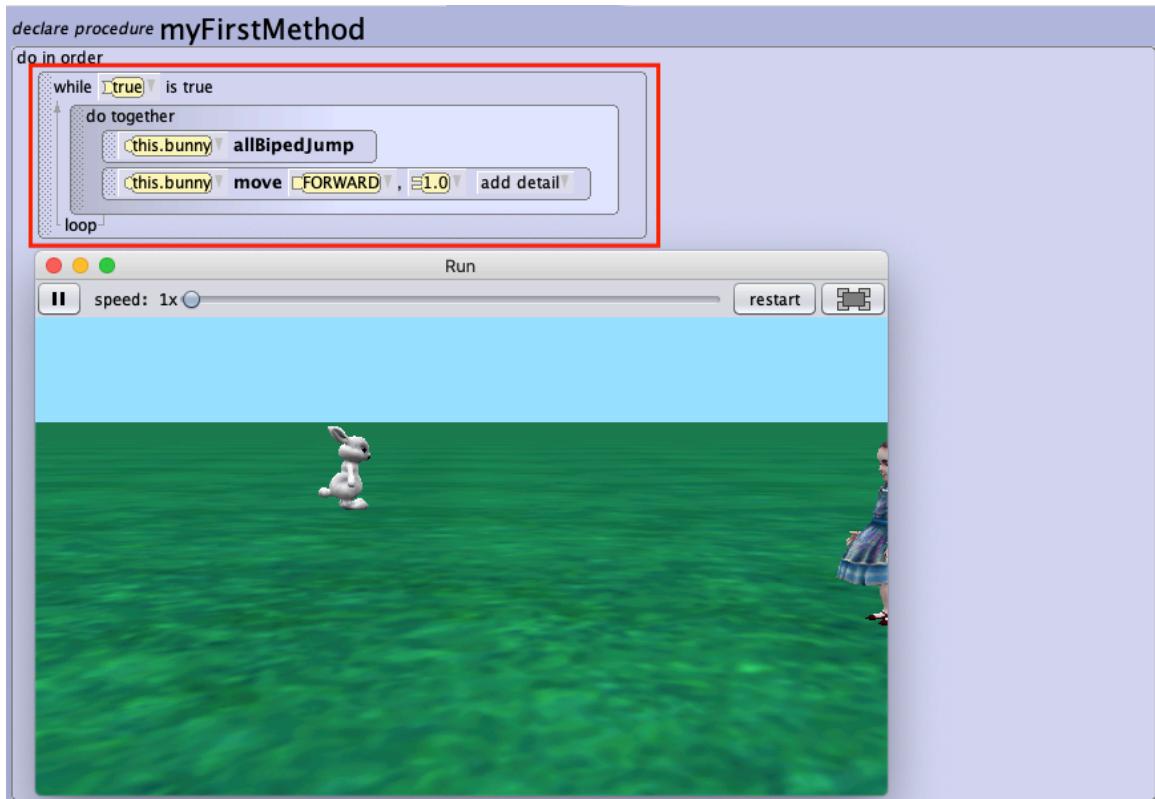


## 25. Menambahkan procedure pada control statement & nested do together



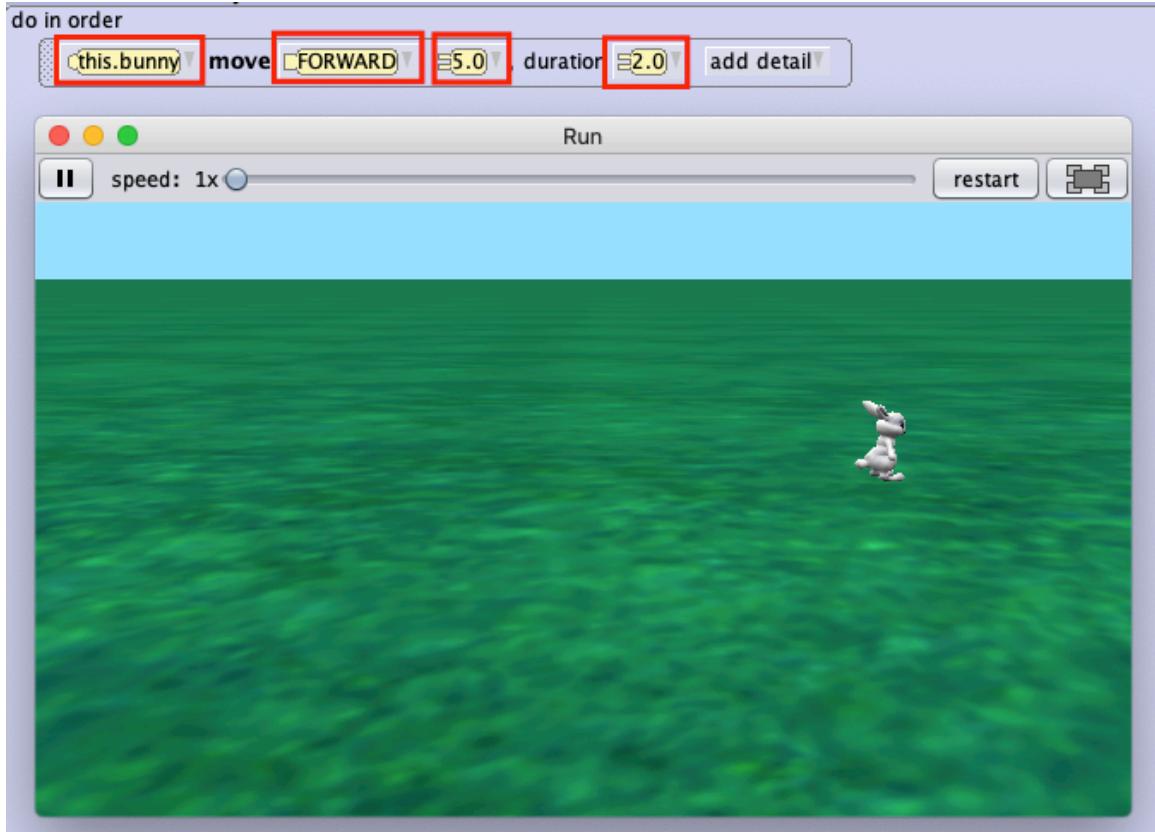
26. Deklarasi Procedure (membuat metode procedure). Procedure dibuat ketika terjadi procedure abstraction (instruksi berulang/bagian dari instruksi yang dilakukan secara berulang)



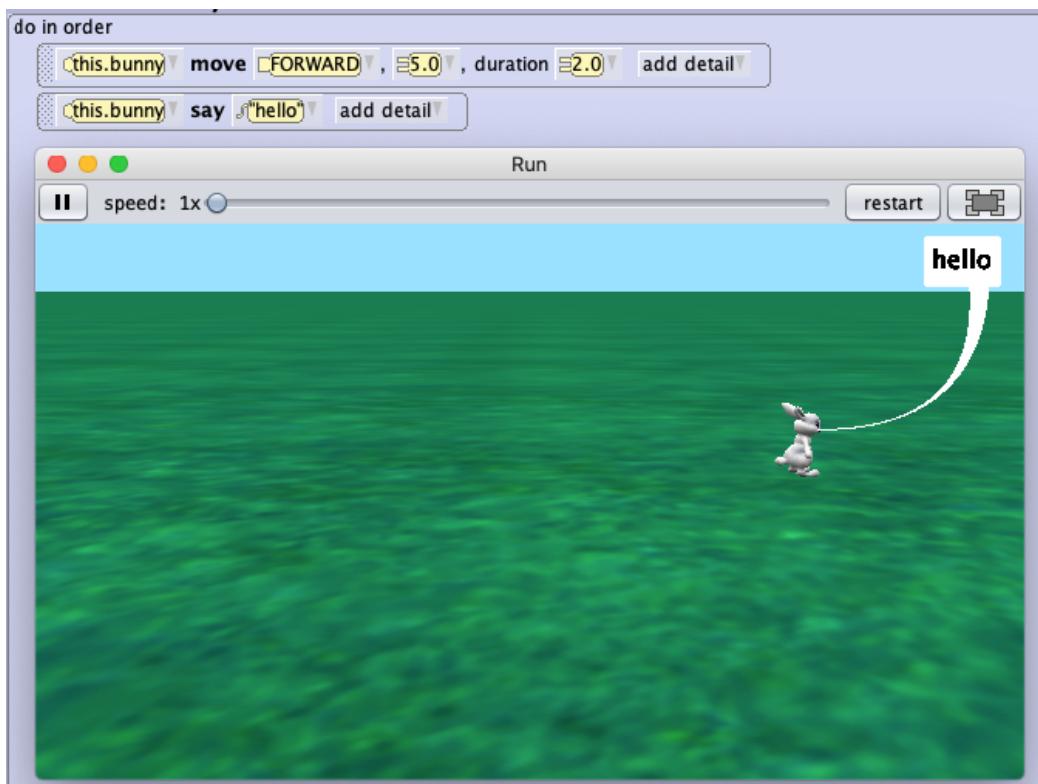


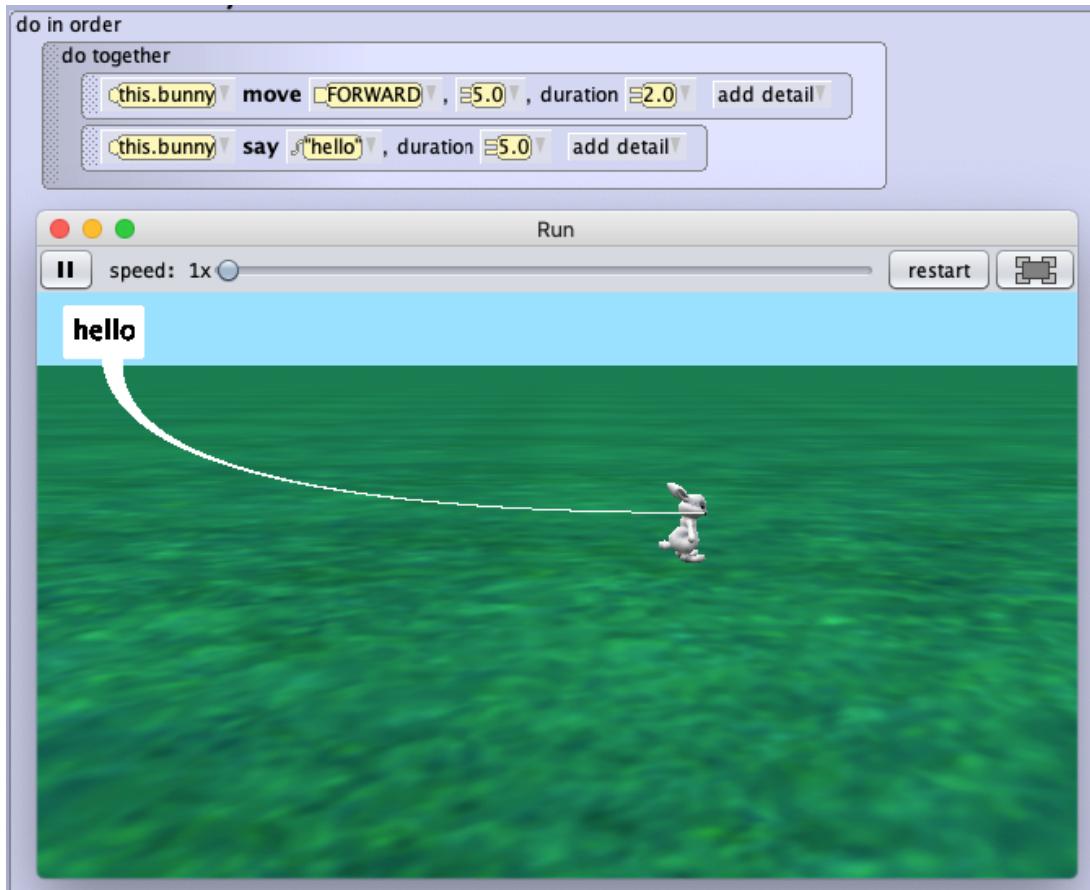
Procedure bipedJump dapat digunakan pada semua class biped, ini disebut sebagai pewarisan (inheritance)

27. Argumen pada Alice 3 dapat diubah sesuai dengan kebutuhan, adapun argumen tersebut terdiri dari object, direction, direction amount, time duration

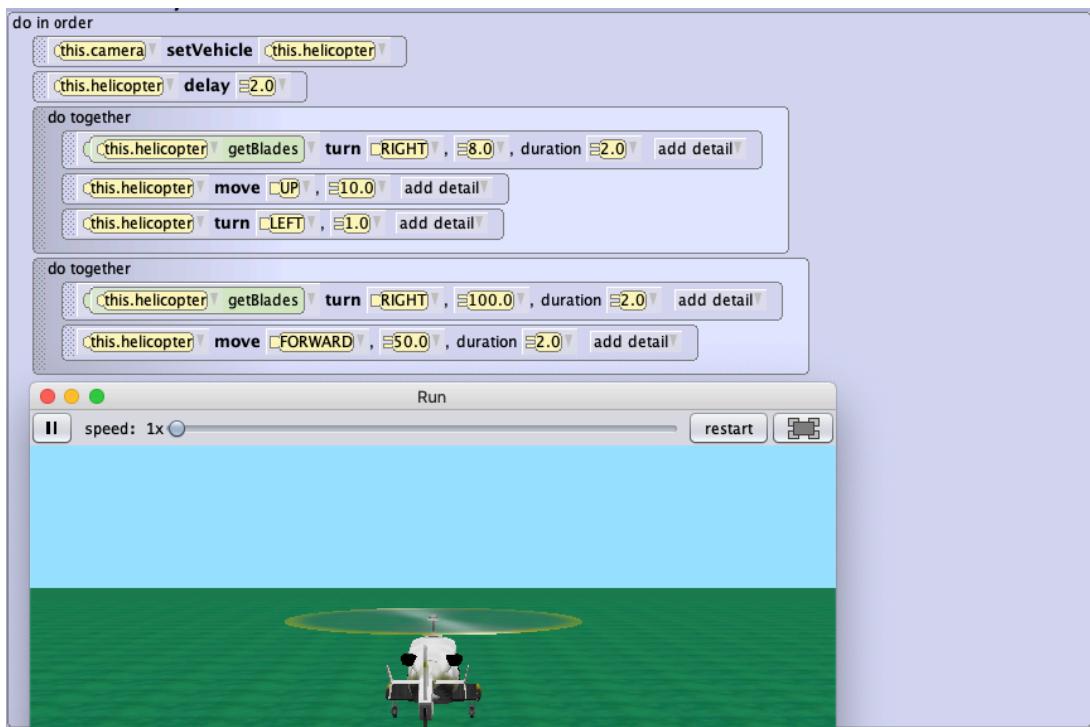


## 28. Control statement do in order dan do together

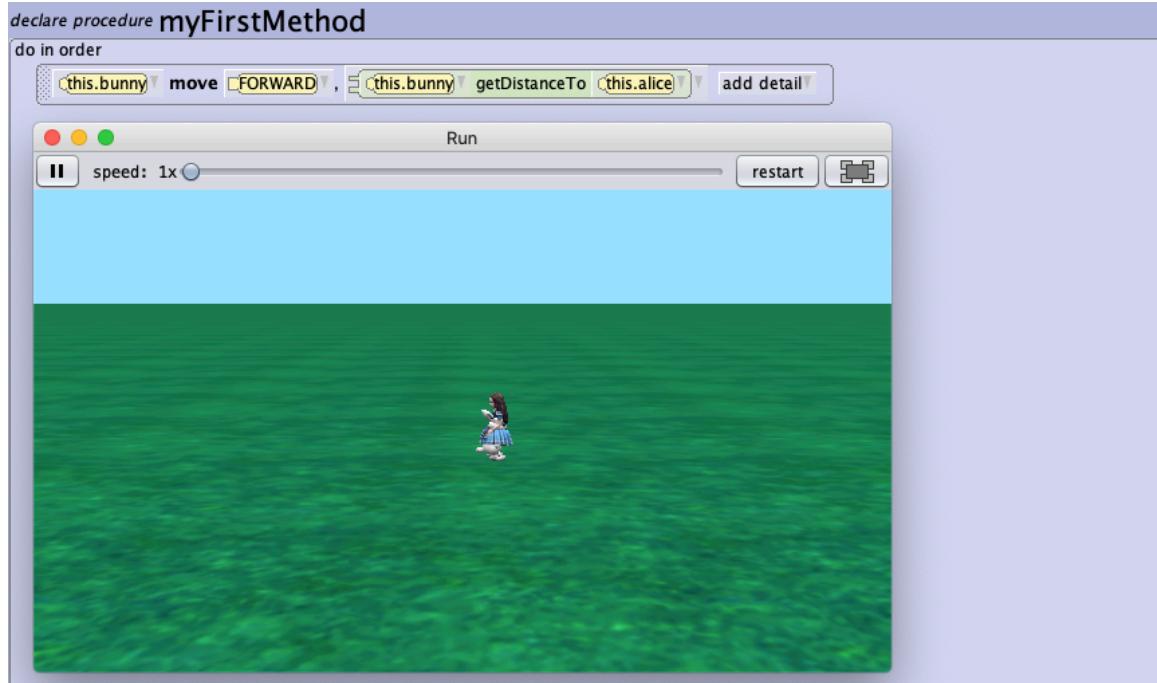




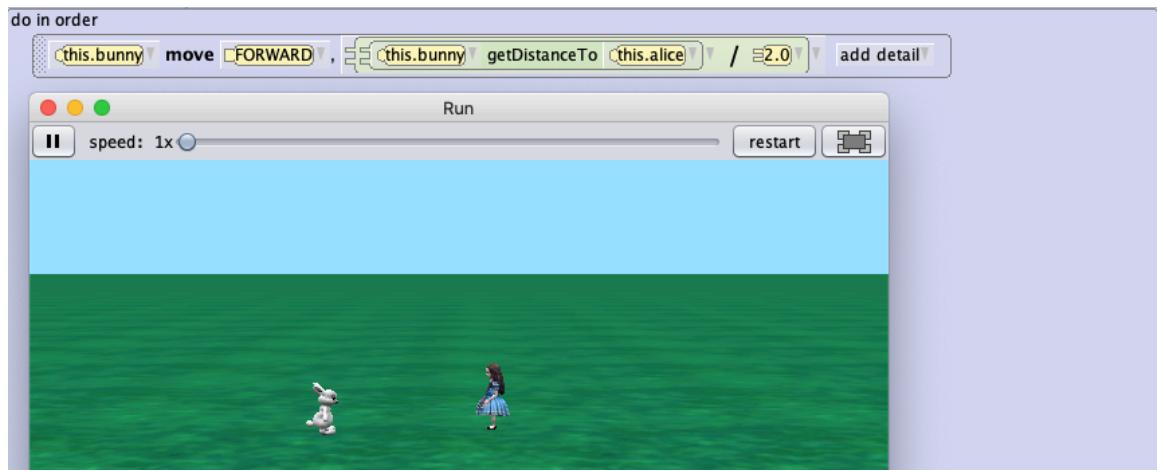
## 29. Menggunakan procedure setVehicle



30. Seperti yang sudah dijelaskan sebelumnya bahwa function adalah metode yang mengembalikan nilai. Pada Alice 3 function tidak dapat digunakan secara langsung, function hanya mengembalikan nilai. Berikut ini contoh penggunaan function.



31. Menambahkan operator pada function



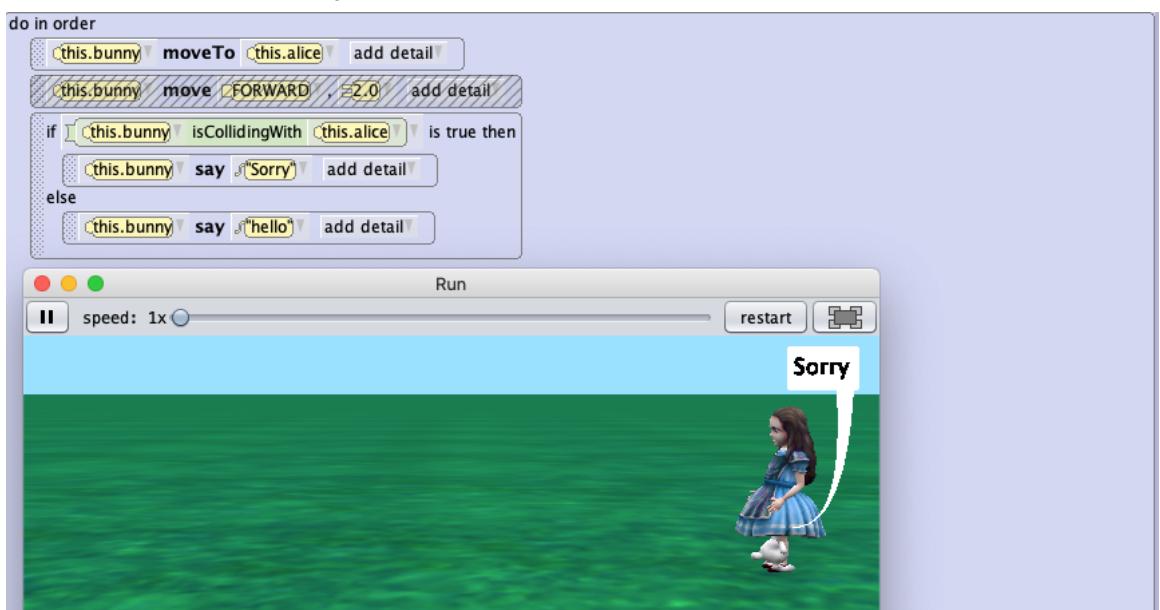
32. Menambahkan function lain untuk menagani operator sebelumnya



*nb:cek nilai depth pada setup scene*

### 33. Kuis

### 34. Kontrol statement IF pada Alice

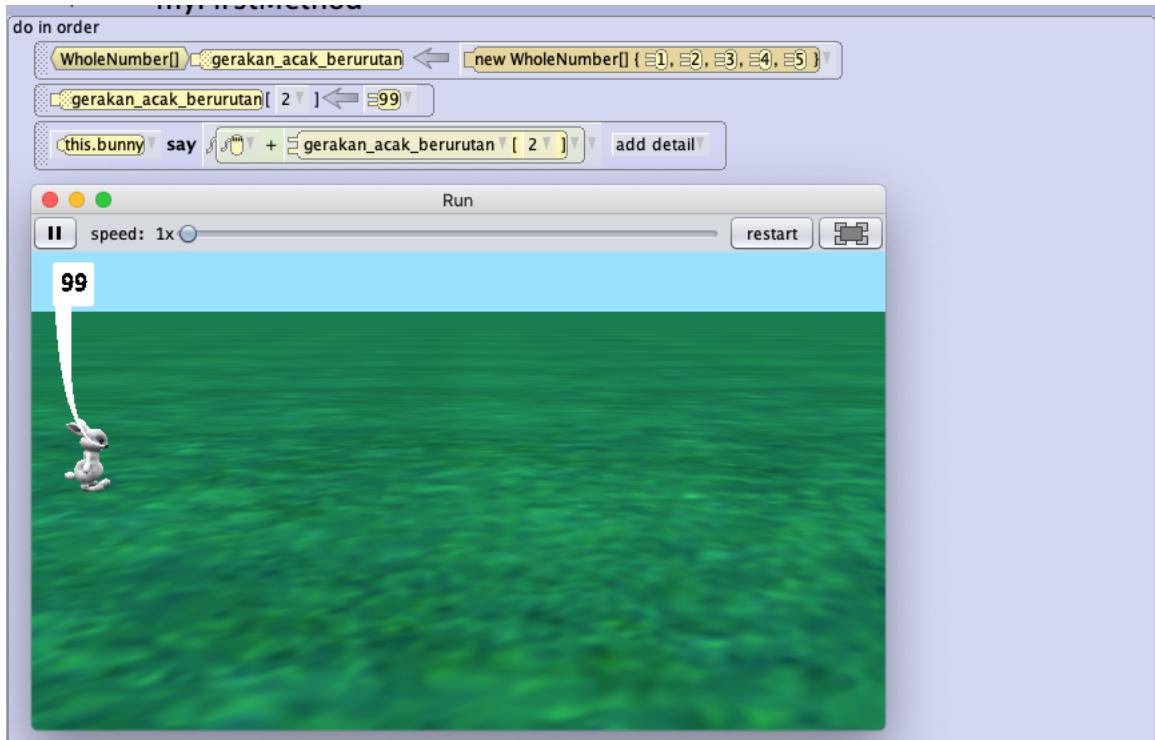


### 35. Kontrol statement WHILE pada Alice

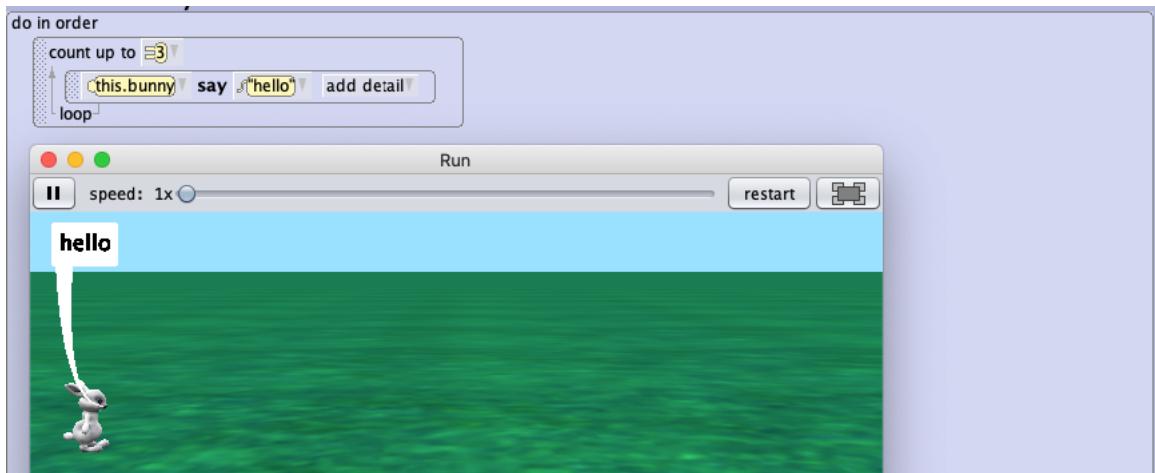


### 36. Variabel pada Alice (Cek tipe data pada slide JF 2 10)

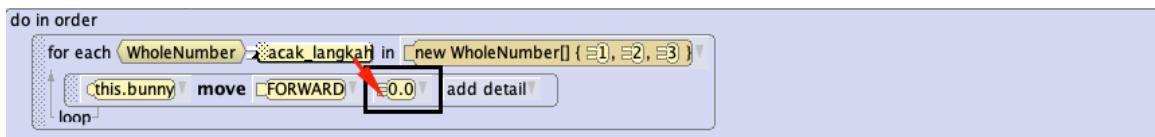




### 37. Kontrol statement COUNT



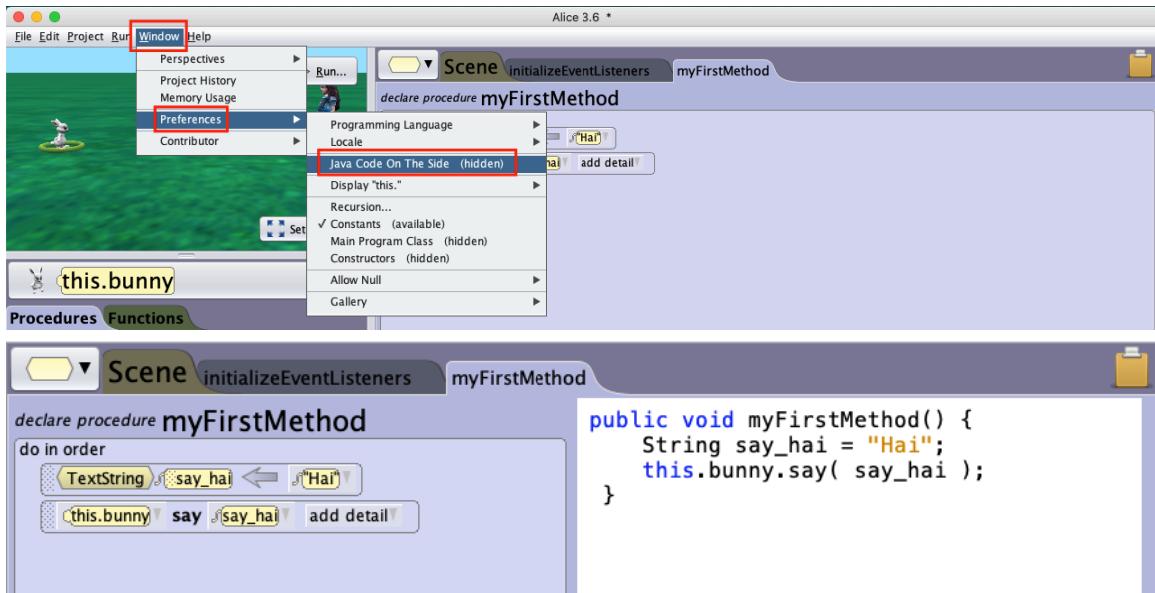
### 38. Kontrol Statement FOR EACH IN



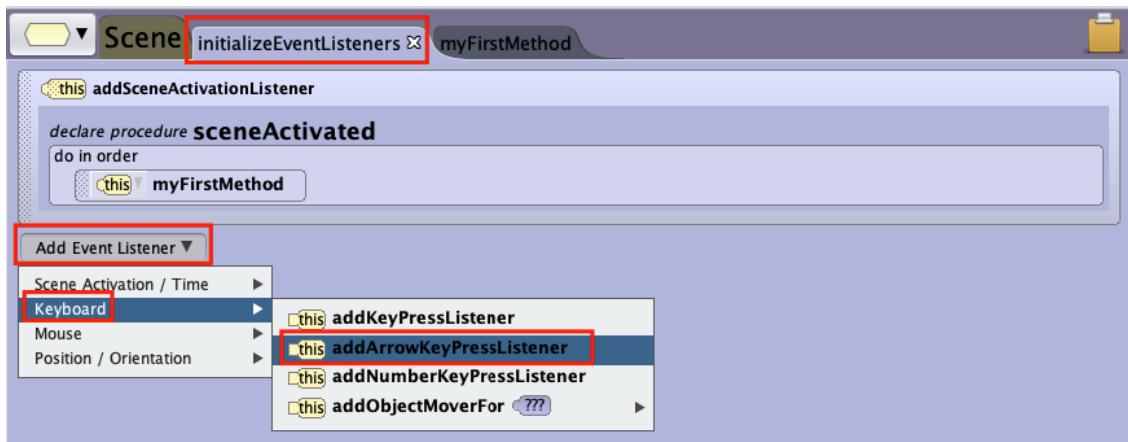
### 39. Kontrol Statement EACH IN TOGETHER

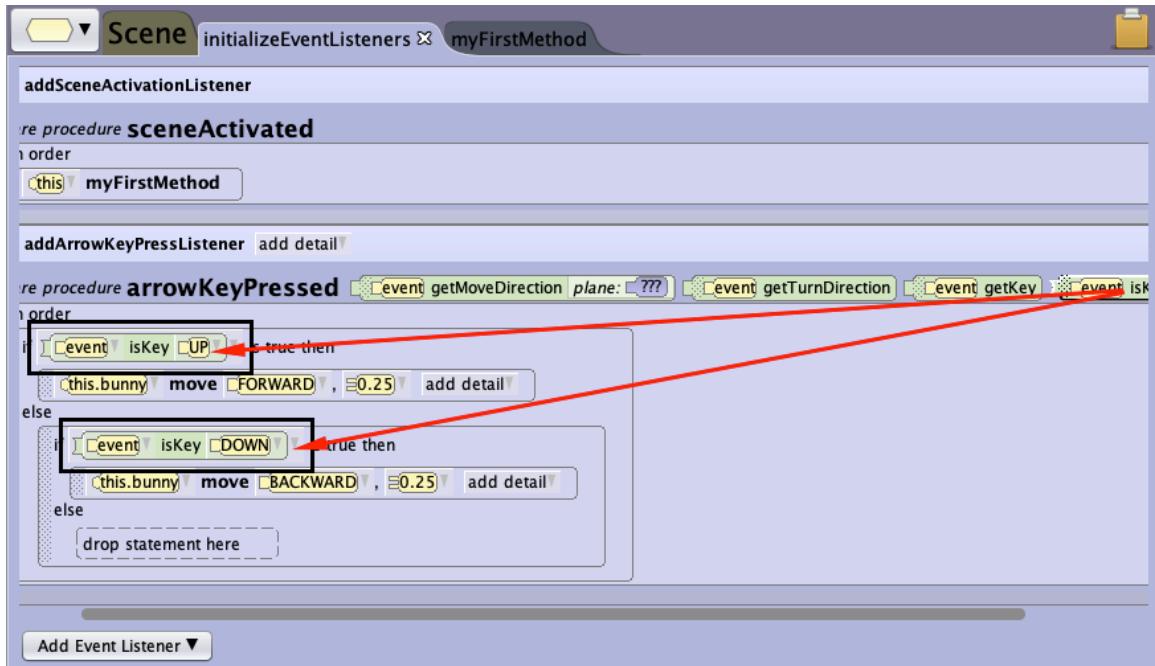


### 40. Melihat kode java pada Alice

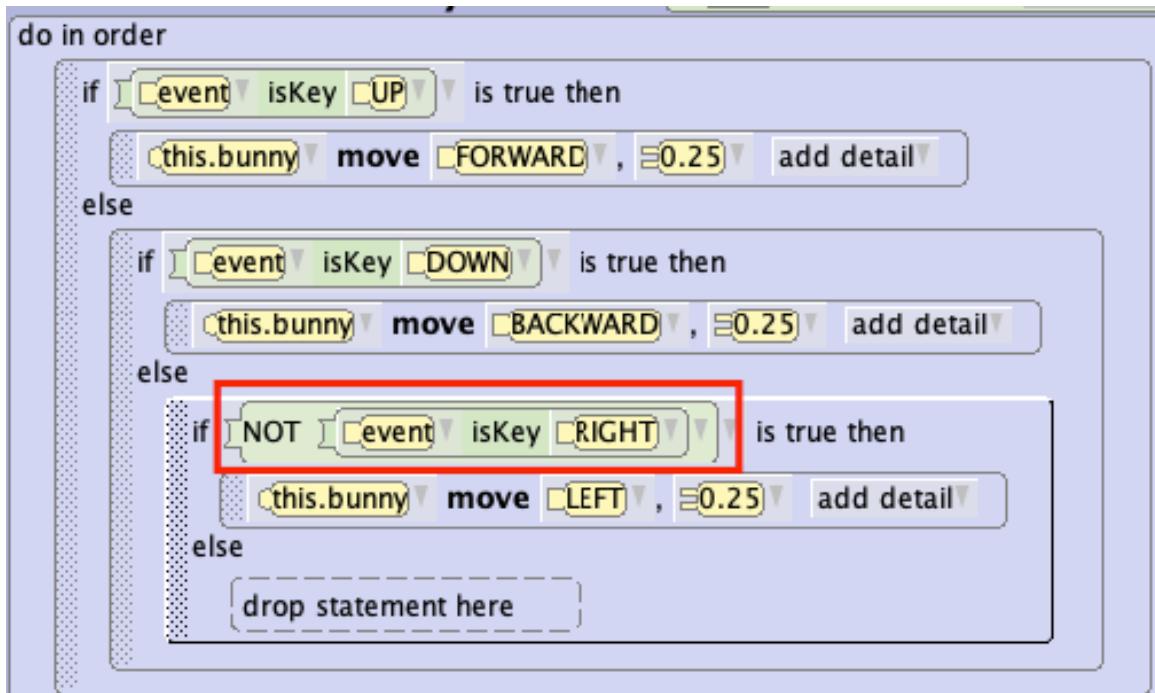


### 41. Kontrol melalui keyboard

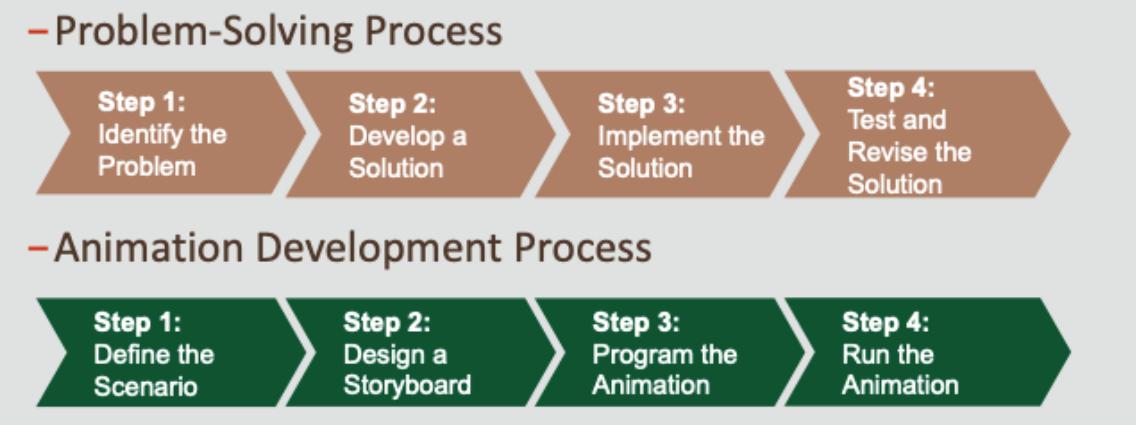




Bagaimana dengan ini!



42. Buatlah animasi lengkap dengan proses sebagai berikut!



43. Perbandingan variabel pada Alice dan Java

The screenshot shows a comparison between Alice script and Java code:

**Alice Script (Top):**

```
declare procedure myFirstMethod
do in order
    [TextString] hello_world ← ["Hello World"]
    [this.alice] say [hello_world] add detail
```

**Java Code (Bottom):**

```
1 public class utama {
2     public static void main(String[] args) {
3         // TODO Auto-generated method stub
4         String hello_world = "Hello World";
5         System.out.println(hello_world);
6     }
7 }
```

44. Relational Operator

The screenshot shows Alice script utilizing relational operators:

```
declare procedure myFirstMethod
do in order
    [this.bunny] move FORWARD, [this.bunny] getDistanceInFrontOf [this.alice] add detail add detail
    if [this.bunny] getHeight < [this.alice] getHeight is true then
        [this.alice] getHead turn FORWARD, 0.05 add detail
    else
        [drop statement here]
```

```
utama.java X

1
2 public class utama {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         // Asumsi jarak antara bunny dan alice adalah 5 meter
7         double bunny = 0.59;
8         double alice = 1.42;
9
10        if (bunny < alice) [
11            System.out.println("Alice getHead Turn Forward 0.05");
12        }
13
14    }
15
16 }
17
```



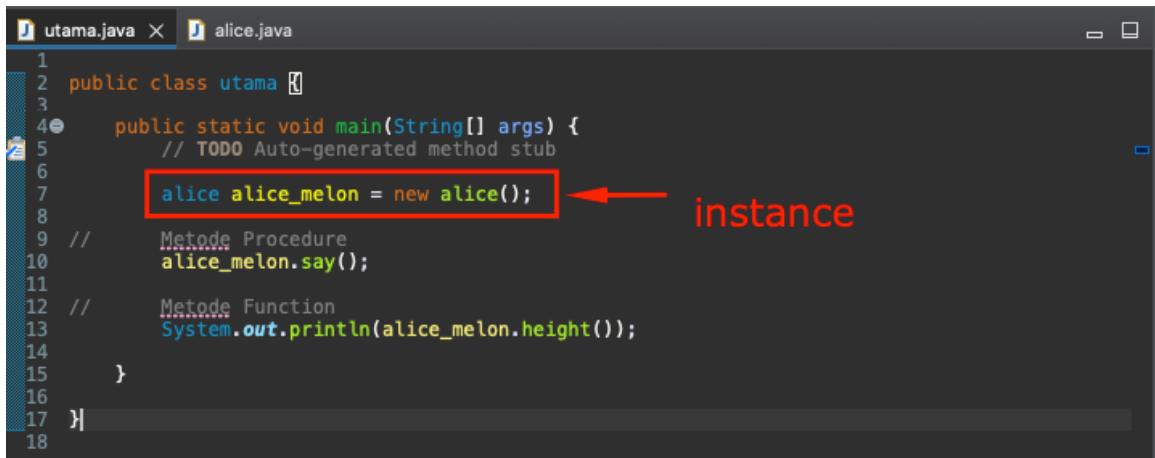
```
1 public class utama {
2     public static void main(String[] args) {
3         // TODO Auto-generated method stub
4         // Asumsi jarak antara bunny dan alice adalah 5 meter
5         String bunny = "behindAlice";
6         String alice = "behindBunny";
7
8         if ((bunny == "behindAlice") && (alice == "behindBunny")) {
9             System.out.println("Kelewatan");
10        }
11    }
12 }
13
14 }
```

## 45. Assignment Operator



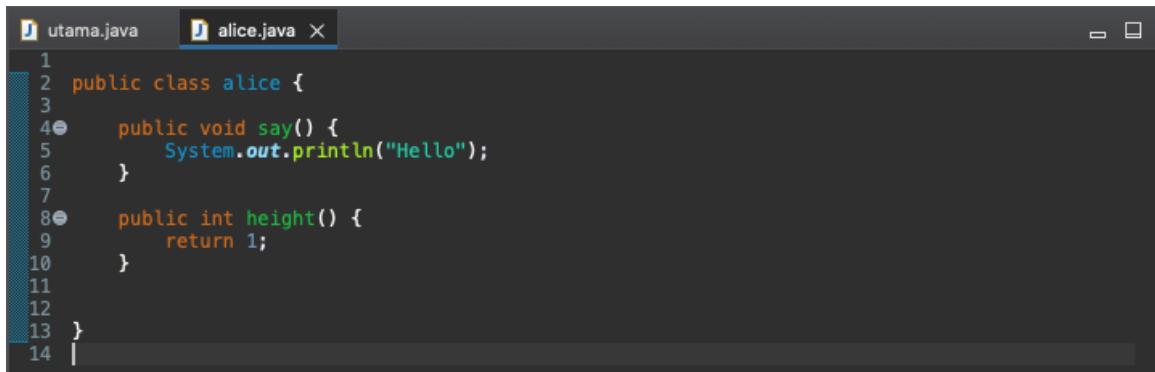
```
1 public class utama {
2
3
4    public static void main(String[] args) {
5        // TODO Auto-generated method stub
6        // Asumsi jarak antara bunny dan alice adalah 5 meter
7        int tiga_meter = 3;
8        int empat_meter = tiga_meter;
9
10       System.out.println(empat_meter);
11
12    }
13
14 }
15
```

#### 46. Mengenal class, objek dan instance pada Java



```
utama.java X alice.java
1
2 public class utama {
3
4    public static void main(String[] args) {
5        // TODO Auto-generated method stub
6
7        alice alice_melon = new alice(); ← instance
8
9        // Metode Procedure
10       alice_melon.say();
11
12      // Metode Function
13      System.out.println(alice_melon.height());
14
15    }
16
17 }
18
```

#### 47. Mengenal Metode Procedure dan Function pada Java



```
utama.java X alice.java
1
2 public class alice {
3
4    public void say() {
5        System.out.println("Hello");
6    }
7
8    public int height() {
9        return 1;
10    }
11
12
13 }
14
```

# PENGENALAN GREENFOOT

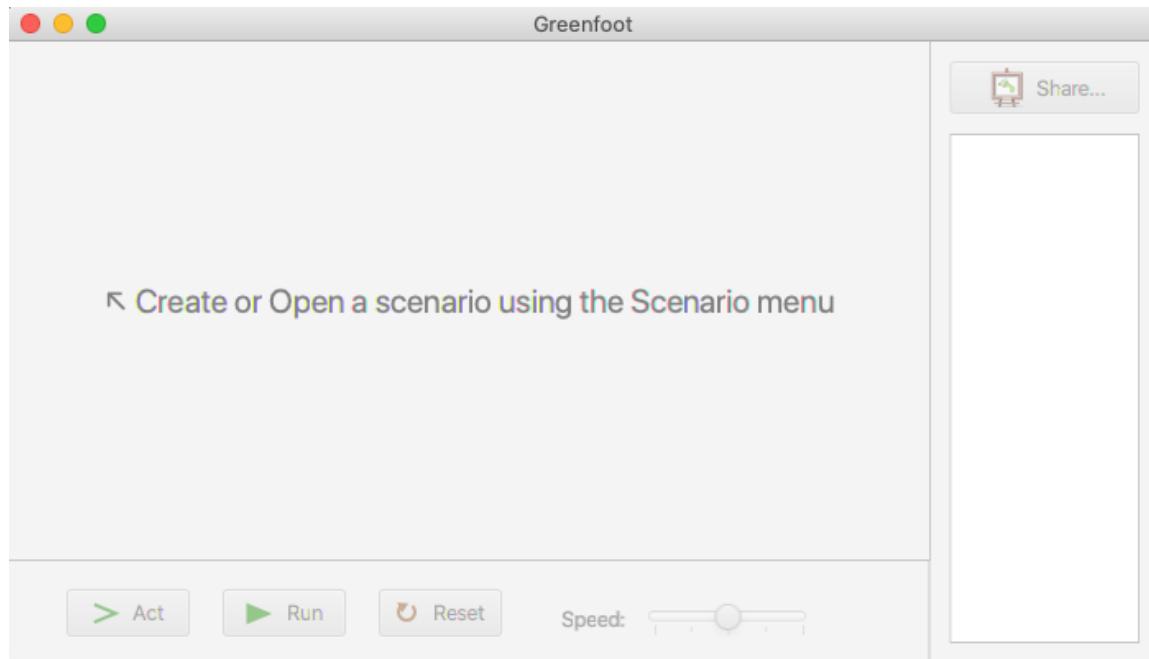
## Tujuan Pembelajaran:

1. Mengenal Greenfoot
2. Memulai dengan Greenfoot
3. Menggunakan metode, variabel dan parameter
4. Memanfaatkan source code dan documentation
5. Mengembangkan dan menguji aplikasi

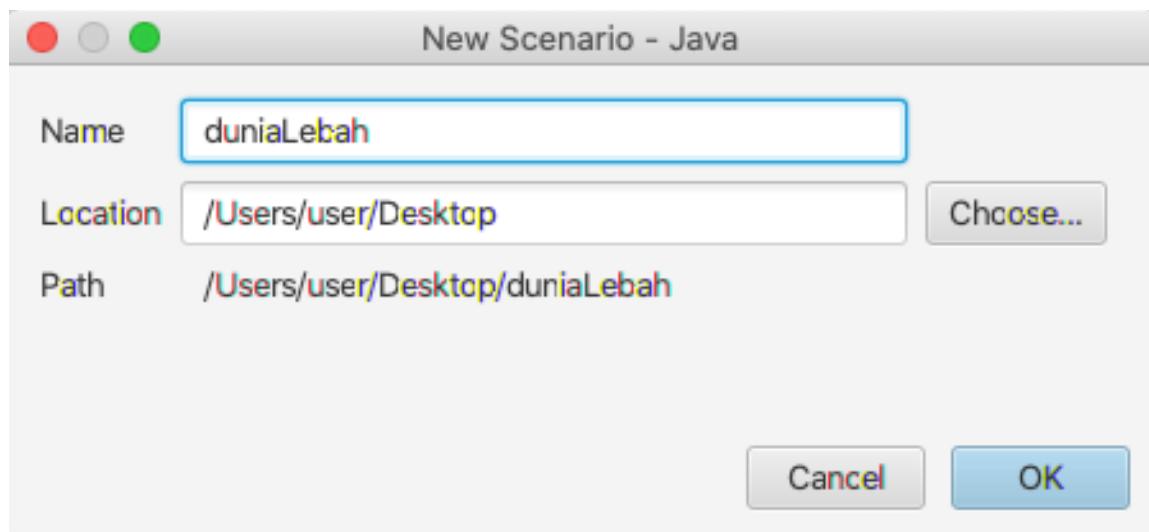
## Pengenalan Greenfoot

Greenfoot adalah perangkat lunak yang di desain untuk pemula agar terbiasa dengan pemrograman berorientasi objek (OOP), yang mendukung pengembangan aplikasi visual dengan menggunakan bahasa pemrograman java.

Memulai program greenfoot dapat dilakukan dengan membuka aplikasi greenfoot terlebih dahulu.



Membuat scenario dapat dilakukan dengan memilih menu scenario kemudian pilih new java scenario kemudian buat nama scenario.

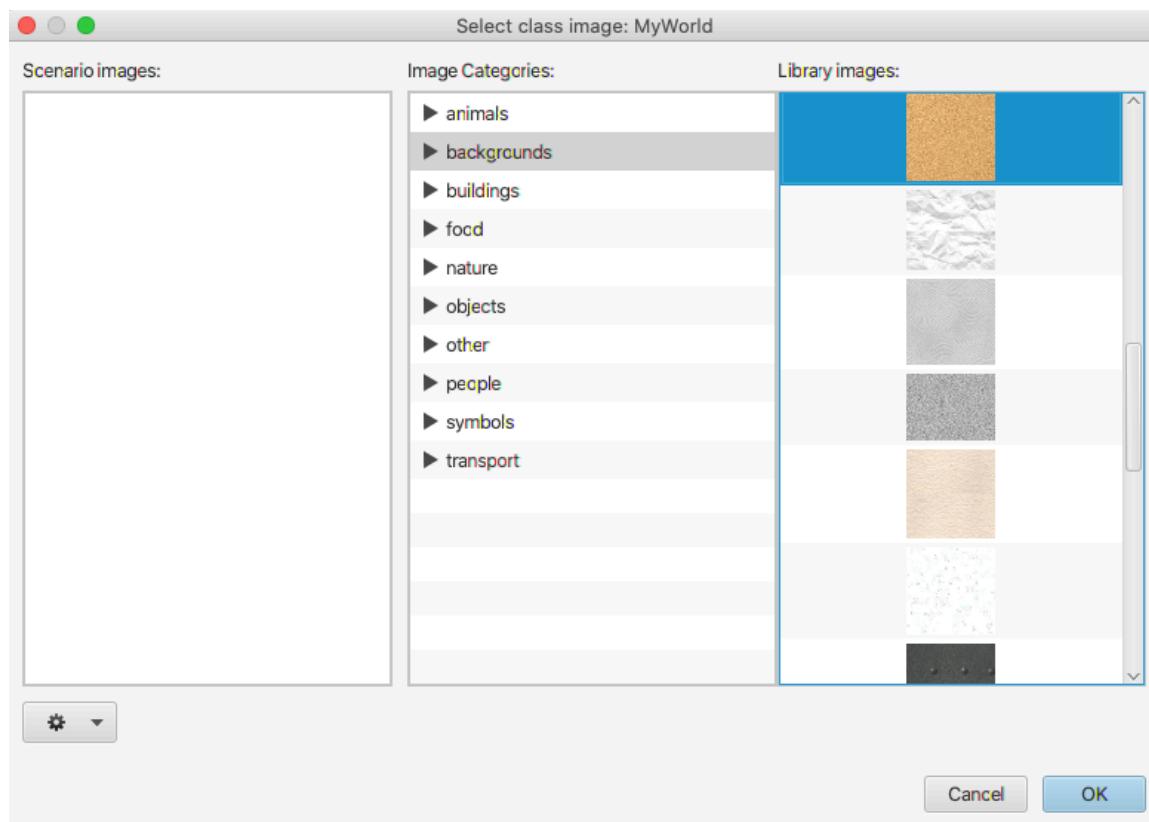


Jika sudah pilih OK kemudian berikut ini tampilan dari scenario yang telah dibuat.



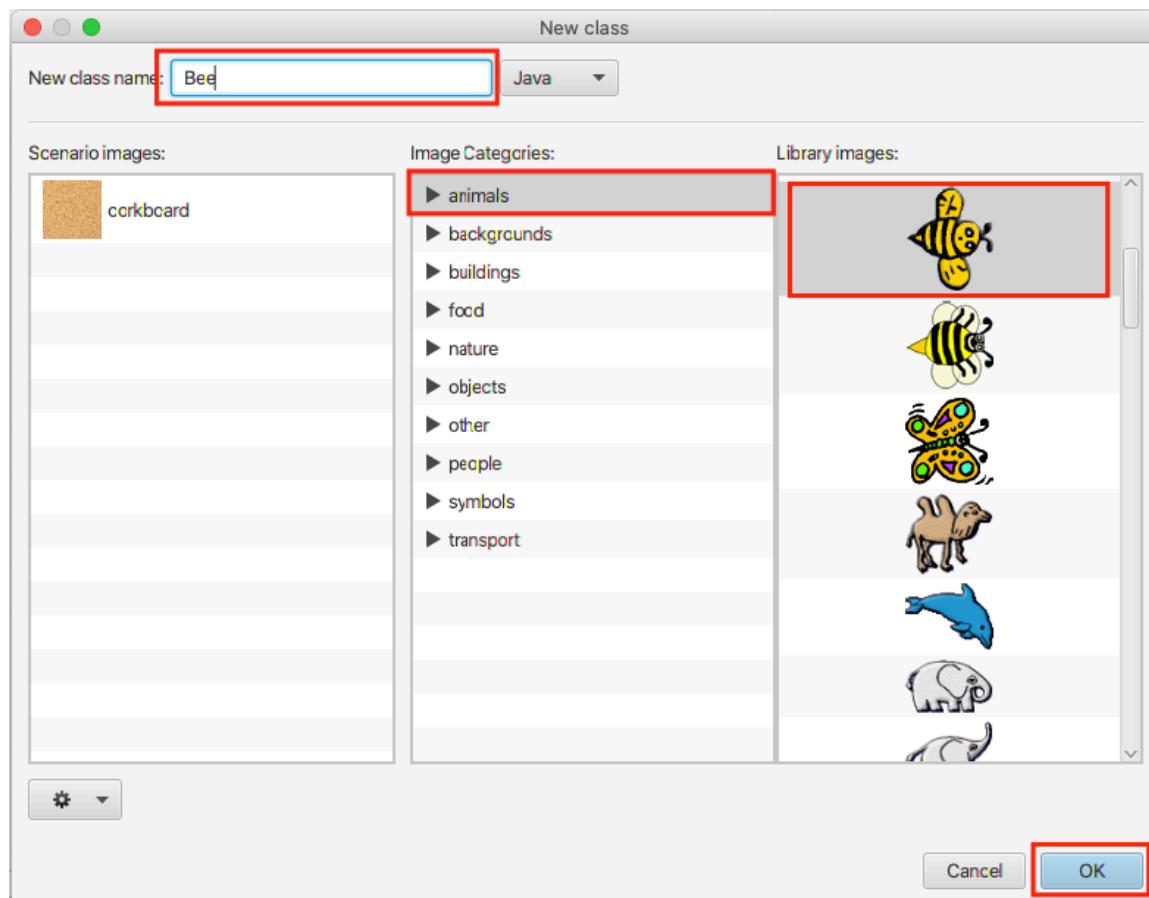
Pengguna dapat menambahkan gambar pada world yang telah tersedia dengan cara klik kanan pada MyWorld kemudian pilih set image,

selanjutnya pilih backgrounds. Pengguna dapat memilih latar belakang yang ingin digunakan.

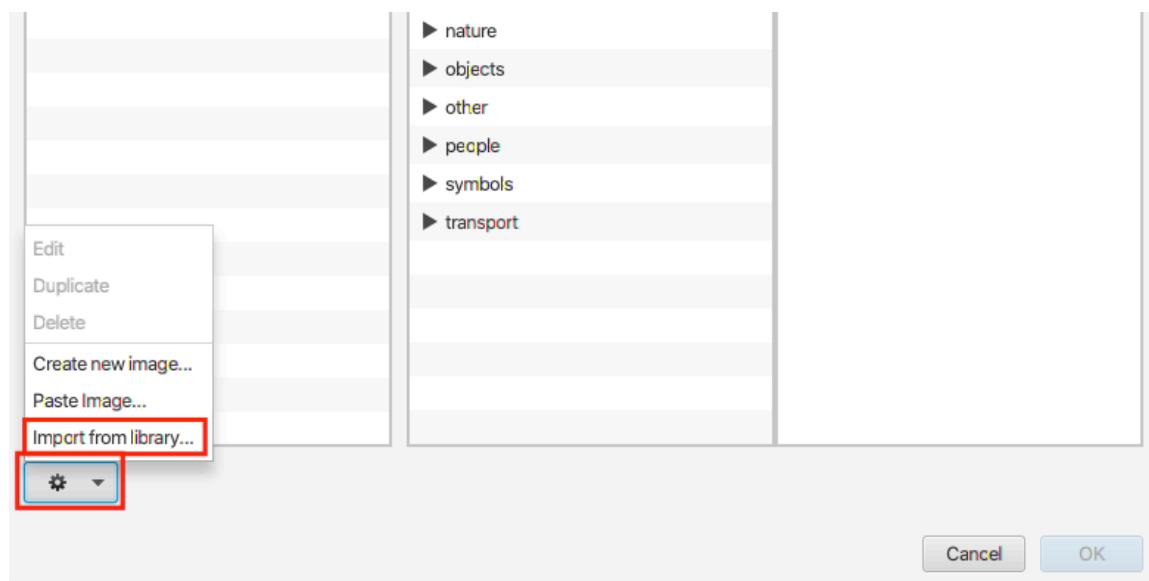


Jika sudah menentukan pilihan untuk latar belakang kemudian pilih OK. Agar latar belakang dapat diterapkan oleh world, klik kanan pada MyWorld kemudian pilih newMyWorld().

Greenfoot mengizinkan pengguna untuk menambahkan subclass melalui superclass yang sudah dipersiapkan. Misalkan saja menambah subclass baru dengan nama Bee pada superclass Actor dapat dilakukan dengan cara klik kanan pada superclass Actor kemudian pilih new subclass, pada jendela tambahkan nama class kemudian pilih gambar yang akan digunakan.



Selain menggunakan gambar yang tersedia, pengguna juga dapat menambahkan gambar lain dengan cara klik pada icon gear kemudian pilih import from library.



Scenario yang telah mengalami perubahan dapat disimpan dengan memilih scenario kemudian pilih save. Subclass Bee belum terlihat pada MyWorld, untuk dapat menampilkannya klik kanan pada subclass Bee kemudian pilih newBee() selanjutnya tempatkan pada MyWorld. Lakukan ujicoba dengan memilih tombol reset.



Semua Bee akan hilang atau kembali ke bentuk awal, agar Bee yang ditempatkan pada MyWorld tidak hilang dapat dengan tools kemudian pilih save the world. Lakukan ujicoba dengan memilih tombol reset, subclass Bee tidak akan hilang, perintah save the world juga secara otomatis menambahkan kode program pada MyWorld. Klik kanan pada subclass kemudian pilih open editor maka jendela source code untuk MyWorld akan terbuka.

Pada jendela source code terdapat beberapa control yang dapat digunakan untuk mengelola kode program, seperti compile untuk memeriksa kesalahan pada kode program, undo untuk kembali ke perubahan sebelumnya, cut untuk memindahkan (source code), copy untuk menempelkan (source code), find untuk mencari kata kunci dan close untuk menutup jendela source code.

The screenshot shows the 'MyWorld - duniaLebah' application window. The title bar has three buttons (red, yellow, green) and the window title. Below the title bar is a toolbar with buttons for 'Compile', 'Undo', 'Cut', 'Copy', 'Paste', 'Find...', and 'Close'. To the right of the toolbar is a dropdown menu labeled 'Source Code'. The main area contains Java-like source code:

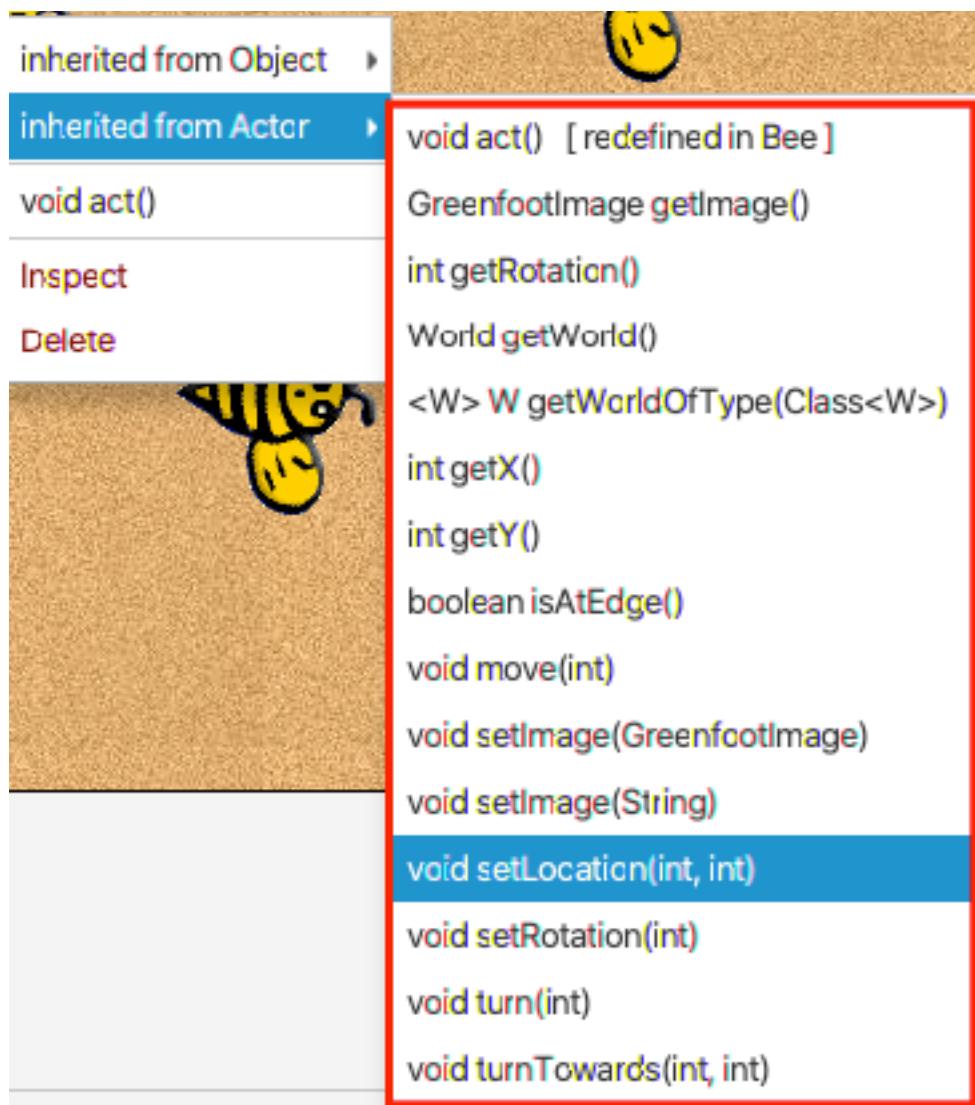
```
*  
*/  
public MyWorld()  
{  
    // Create a new world with 600x400 cells with a cell size of 1x1 pixels  
    super(600, 400, 1);  
    prepare();  
}  
/**  
 * Prepare the world for the start of the program.  
 * That is: create the initial objects and add them to the world.  
 */  
private void prepare()  
{  
    Bee bee = new Bee();  
    addObject(bee, 145, 106);  
    Bee bee2 = new Bee();  
    addObject(bee2, 259, 252);  
    Bee bee3 = new Bee();  
    addObject(bee3, 393, 89);  
    Bee bee4 = new Bee();  
    addObject(bee4, 520, 251);  
    Bee bee5 = new Bee();  
    addObject(bee5, 43, 256);  
}  
}
```

At the bottom of the code editor, there is a status bar with the message 'Class compiled - no syntax errors' and a 'saved' indicator.

Terdapat kontrol tambahan untuk melihat source code dan documentation pada sisi kanan atas jendela source code. Pada bagian bawah jendela source code digunakan untuk menampilkan pesan kesalahan atau error.

Dalam pemrograman juga terdapat istilah pewarisan atau inheritance, yang berarti metode yang ada pada superclass (parent) dapat diturunkan kepada subclass (child).

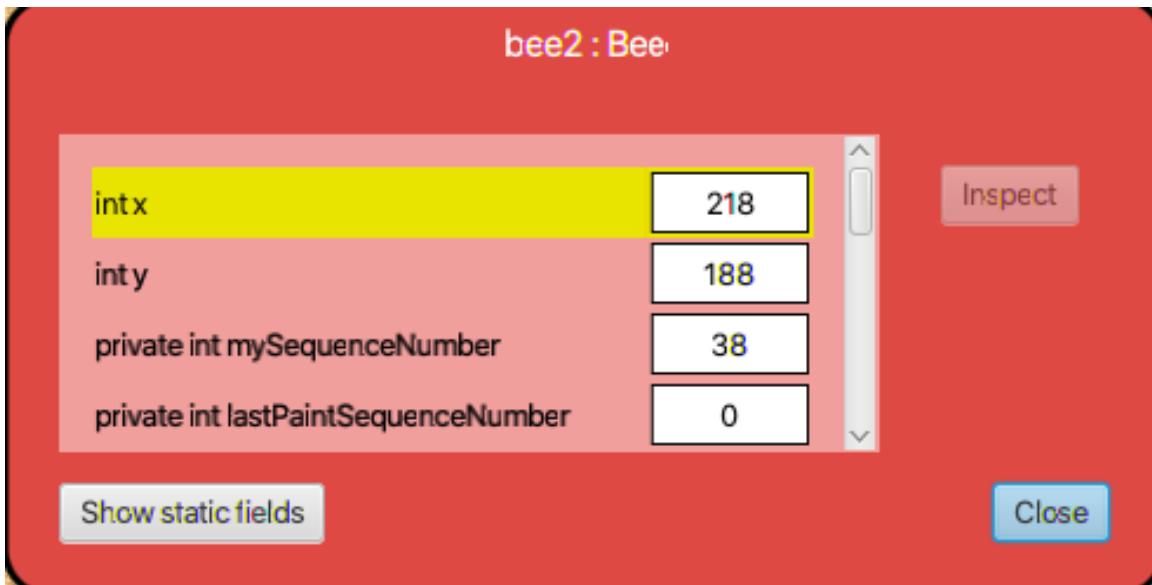
Pengguna dapat melihat metode yang diturunkan oleh superclass Actor kepada subclass Bee dengan cara klik kanan pada instance kemudian pilih inherited from actor. Terdapat cukup banyak metode yang diturunkan dari superclass. Metode tersebut juga terdiri dari procedure dan function.



Metode dalam pemrograman merupakan kumpulan operasi, metode dapat melakukan suatu tugas seperti berputar, berpindah atau metode yang secara khusus mendapatkan nilai. Beberapa komponen yang ada pada metode seperti tipe pengembalian (return type), nama metode dan parameter (contoh void move(3), int getX()), pada greenfoot juga terdapat metode dengan tipe pengembalian void dan metode dengan tipe pengembalian non-void (integer, boolean).

Pada pemrograman terdapat variabel yang digunakan untuk menampung data atau informasi yang nantinya akan digunakan kembali. Selanjutnya adalah object yang ditambahkan pada MyWorld memiliki properti yang disebut object properties. Object property ini

terdiri dari size, color, range of movement. Adapun cara yang dapat digunakan untuk melihatnya dengan cara klik kanan pada object kemudian pilih inspect.



## Metode act()

Metode act() merupakan metode yang akan dijalankan pertama kali dari semua metode yang ada. Tambahkan metode act() pada object hewan kemudian tambahkan metode berikut beserta parameternya.

```
public void act()
{
    // Add your action code here.
    move(5);
    turn(5);
}
```

Lakukan debugging untuk mengetahui kesalahan yang terjadi pada kode program.

```
public void act()
{
    // Add your action code here.
    move(5);
    turn(5);
}
```

Source code pada greenfoot merupakan blueprint atau peta yang mendefinisikan objek dan fungsi program, source code dapat di kelola melalui code editor, melalui code editor pengembang dapat menuliskan dan memodifikasi source code, meninjau metode pewarisan dan propertinya serta metode yang dituliskan secara spesifik.

Apa saja yang terdapat pada source code? Source code pada greenfoot terdiri dari class, metode act(), method signature (metode dengan nama metode dan parameternya, invoking method), komentar.

Greenfoot menyediakan dokumentasi yang bisa dijadikan sebagai acuan untuk belajar menggunakan metode-metode yang telah disediakan.

Metode sequential pada greenfoot berarti terdapat beberapa metode atau instruksi yang dituliskan secara urutan. Pada greenfoot juga terdapat statement if yang dapat digunakan untuk menggerakkan objek melalui keyboard.

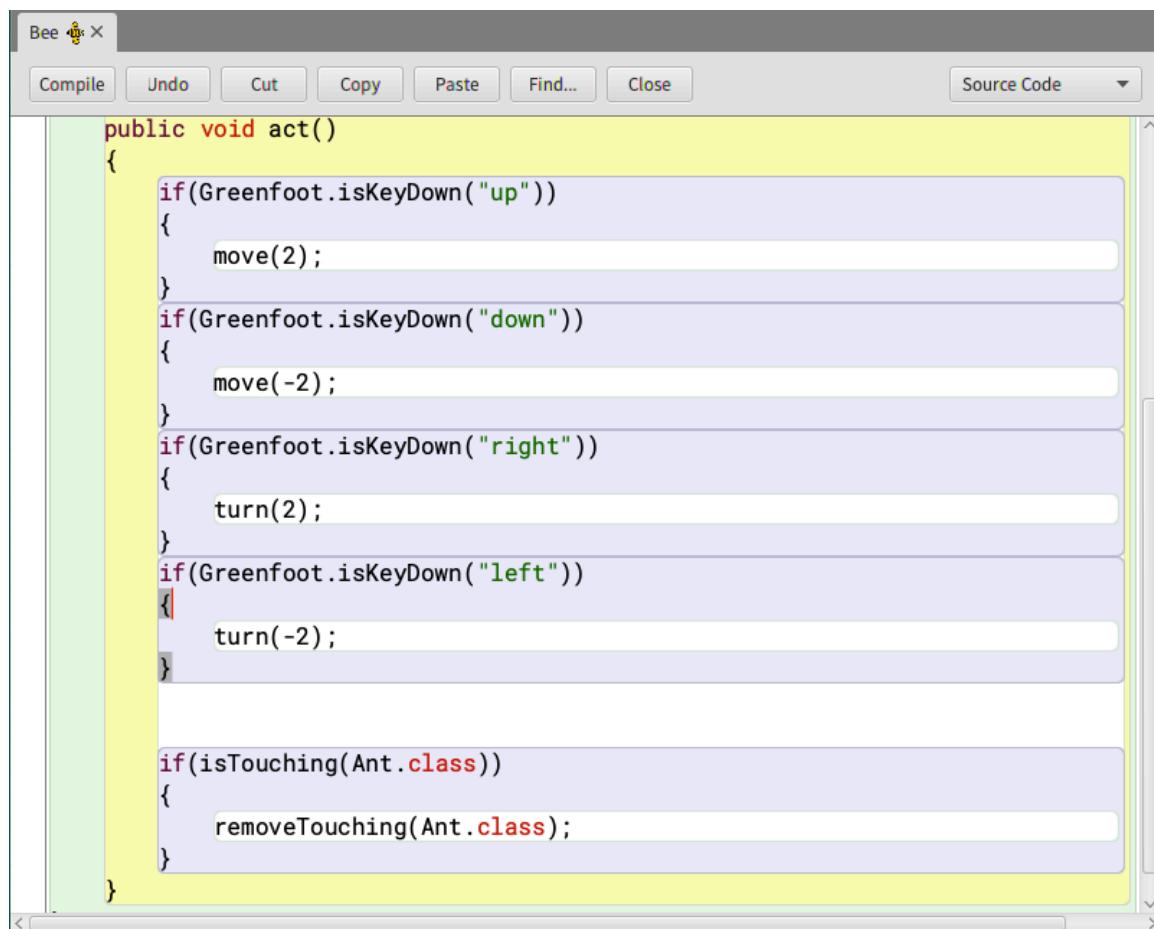
```
public void act()
{
    if(Greenfoot.isKeyDown("up"))
    {
        move(2);
    }
    if(Greenfoot.isKeyDown("down"))
    {
        move(-2);
    }
    if(Greenfoot.isKeyDown("right"))
    {
        turn(2);
    }
    if(Greenfoot.isKeyDown("left"))
    {
        turn(-2);
    }
}
```

Langkah dalam melakukan debugging program dapat dilakukan dengan compile, pengembang juga dapat melihat kesalahan yang terjadi dengan auto-layout (pada menu edit).

Terdapat beberapa fase yang dapat dilakukan untuk mengembangkan program dengan greenfoot, diantaranya:

1. Menganalisa permasalahan untuk dipecahkan
2. Merancang solusi
3. Mengembangkan game
4. Menguji game (sesuai dengan rancangan sebelumnya)
5. Evaluasi (Mengembangkan kembali dengan alur yang sama)

Membuat program sederhana dengan greenfoot:

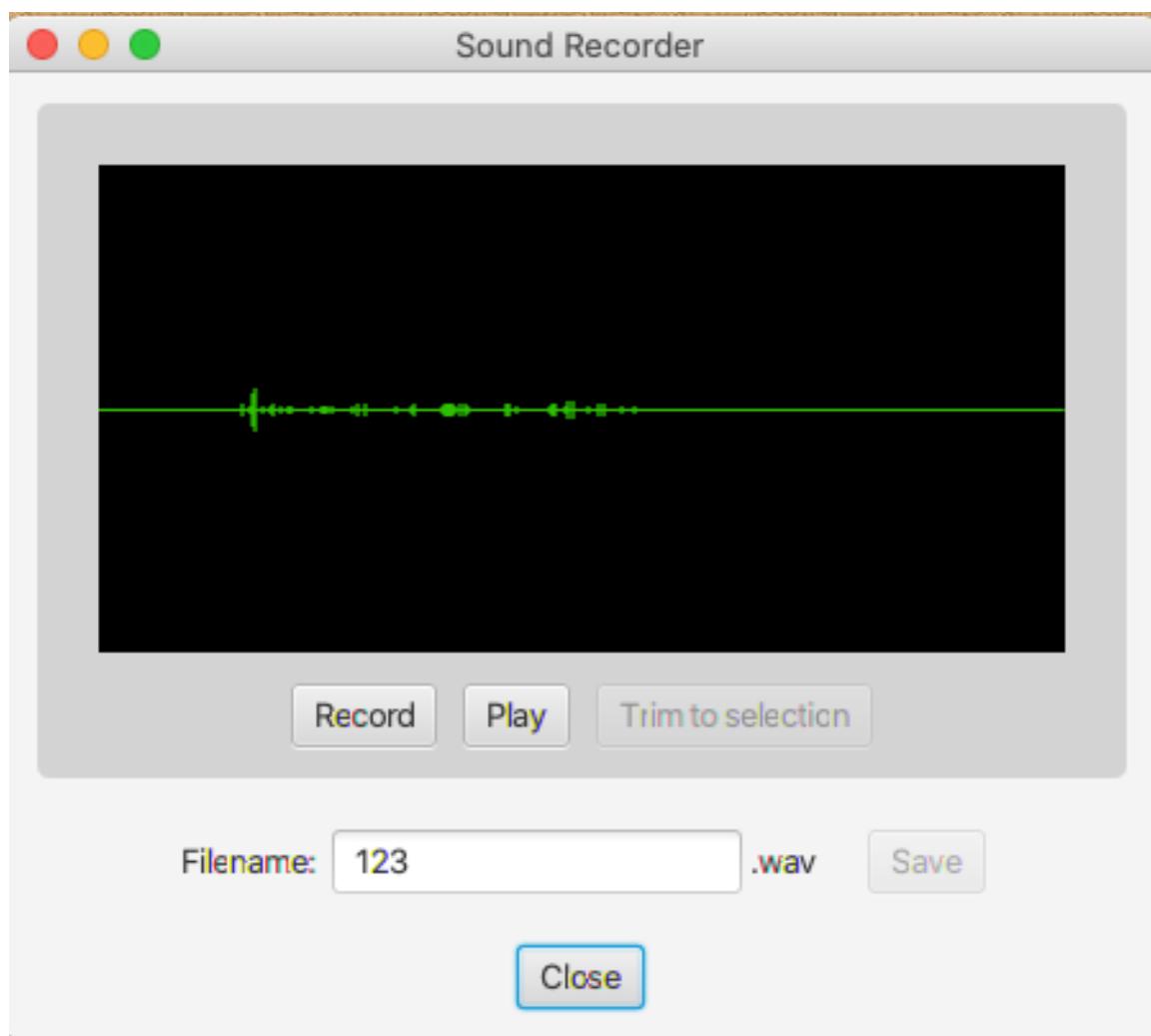


The screenshot shows the Greenfoot IDE interface with a script titled "Bee". The code in the editor is as follows:

```
public void act()
{
    if(Greenfoot.isKeyDown( "up" ))
    {
        move(2);
    }
    if(Greenfoot.isKeyDown( "down" ))
    {
        move(-2);
    }
    if(Greenfoot.isKeyDown( "right" ))
    {
        turn(2);
    }
    if(Greenfoot.isKeyDown( "left" ))
    {
        turn(-2);
    }

    if(isTouching(Ant.class))
    {
        removeTouching(Ant.class);
    }
}
```

Menambahkan suara pada permainan dapat dengan memilih menu tools kemudian pilih show sound recorder.



Lakukan perekaman suara dengan klik tombol record dan selesai merekam dengan klik tombol stop. Kemudian tambahkan nama file rekaman dan simpan rekaman. Agar dapat menggunakan rekaman tersebut tambahkan instruksi berikut.

```
if(isTouching(Ant.class))
{
    Greenfoot.playSound("123.wav");
    removeTouching(Ant.class);
}
```

Menambahkan nilai random dengan membuat salah satu objek melakukan rotasi dengan nilai yang diacak dapat dilakukan pada greenfoot dengan menambahkan perintah berikut.

```
public void act()
{
    // Add your action code here.
    move(10);
    if(isAtEdge())
    {
        turn(Greenfoot.getRandomNumber(90));
    }
}
```

getRandomNumber(90) berarti program akan mengacak angka dengan rentang antara 0 sampai dengan 89.

Pada greenfoot juga dapat digunakan operator pembanding, berikut ini contoh penggunaan operator pembanding pada greenfoot.

```
public void act()
{
    // Add your action code here.
    if(Greenfoot.getRandomNumber(100) < 20)
    {
        move(10);
        turn(20);
    }
}
```

Instruksi di atas menunjukan bahwa jika bilangan random kurang dari 20, maka akan ada perpindahan posisi dan juga rotasi yang terjadi pada objek. Selain operator pembanding juga terdapat operator logika seperti berikut ini.

```
public void act()
{
    // Add your action code here.
    if((getX() > getWorld().getWidth()/2) &&
    (getY() < getWorld().getHeight()/2))
    {
        move(10);
    }
}
```

Pemrogram juga dapat menambahkan logikan or (||) atau menambahkan not (!). Greenfoot juga mendukung penggunaan percabangan dalam instruksiannya.

```
public void act()
{
    // Add your action code here.
    move(10);
    if(Greenfoot.getRandomNumber(90) < 20)
    {
        turn(90);
    }
    else
    {
        move(5);
    }
}
```

Sama halnya dengan java, greenfoot juga memiliki konstruktor pada class myWorld, adapun bentuk konstruktornya adalah sebagai berikut.

```
public class MyWorld extends World
{
    /**
     * Constructor for objects of class MyWorld.
     *
     */
    public MyWorld()
    {
        // Create a new world with 600x400 cells with a cell size of 1x1
        super(800, 600, 1);
        prepare();
    }
}
```

Konstruktor adalah sebuah metode yang namanya mirip dengan nama class tempat metode tersebut berada. Pemrogram dapat menambahkan objek secara manual ke world yang tersedia dengan menambahkan instruksi berikut ini pada metode MyWorld.

```
public MyWorld()
{
    // Create a new world with 600x400 cells with a cell size of 1x1
    super(800, 600, 1);
    prepare();
    addObject(new Ant(), 400, 300);
}
```

Penggunaan kata kunci new menunjukkan adanya instansiasi class, sehingga class blueprint (class object) dapat digunakan.

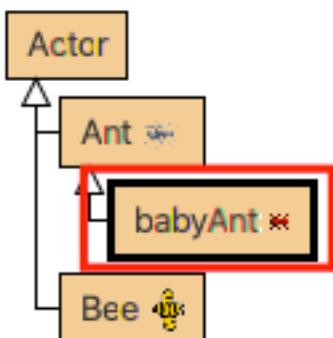
Pemrogramm dapat membuka dokumentasi untuk melihat beberapa metode built-in yang dapat digunakan untuk merekayasa animasi, misalkan saja menggunakan salah satu metode yaitu setLocation.

```

public void act()
{
    // Add your action code here.
    move(10);
    if(isAtEdge())
    {
        setLocation(200, 300);
    }
}

```

Menjadikan subclass sebagai superclass, dapat dilakukan pada greenfoot dengan menambahkan class seperti biasa.



Menambahkan konsep inheritance pada greenfoot dapat dilakukan dengan menambahkan sebuah metode procedure pada class Ant kemudian nanti metode ini dapat digunakan oleh babyAnt.

Tambahkan kode berikut pada class Ant:

```

public void teleportasi()
{
    move(10);
    if(isAtEdge())
    {
        setLocation(Greenfoot.getRandomNumber(200),
                    Greenfoot.getRandomNumber(200));
    }
}

```

Kemudian panggil metode tersebut pada class babyAnt:

```
public void act()
{
    // Add your action code here.
    teleportasi();
}
```

Memberikan batasan pada tepi kanan pada objek dapat dilakukan dengan menambahkan metode fungsi pada superclass Ant.

```
public boolean atRightEdge()
{
    if(getX() > getWorld().getWidth() - 20)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

Panggil metode atRightEdge tersebut pada class babyAnt:

```
public void act()
{
    // Add your action code here.
    move(10);
    if(atRightEdge())
    {
        setLocation(400, 300);
    }
}
```

Metode ask pada greenfoot memberikan inputan dari pengguna, ini dapat dilakukan dengan menambahkan instruksi berikut ini.

```
String nama = Greenfoot.ask("Nama Anda: ");
int n = Integer.parseInt(nama);
if(Greenfoot.isKeyDown("UP"))
{
    move(n);
}
```

Menjadikan instruksi bergerak dengan keyboard dan gerakan menghilangkan object sebagai metode.

```
public void gerakanBee()
{
    if(Greenfoot.isKeyDown("UP"))
    {
        move(5);
    }

    if(Greenfoot.isKeyDown("DOWN"))
    {
        move(-5);
    }

    if(Greenfoot.isKeyDown("RIGHT"))
    {
        turn(5);
    }

    if(Greenfoot.isKeyDown("LEFT"))
    {
        turn(-5);
    }
}
```

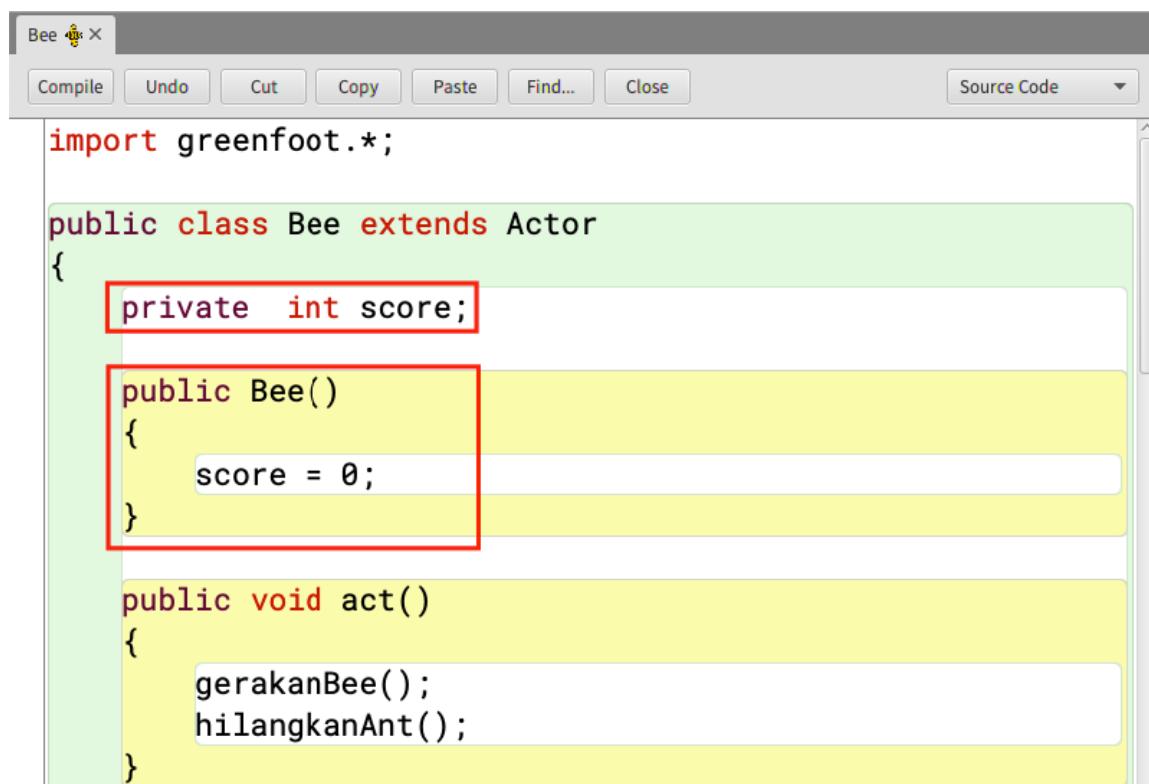
```
public void hilangkanAnt()
{
    if(isTouching(Ant.class))
    {
        //Greenfoot.playSound("hai.wav");
        removeTouching(Ant.class);
    }
}
```

Lakukan pemanggilan nama metode pada class utama (class act) agar dapat menggunakan instruksi.

```
public void act()
{
    gerakanBee();
    hilangkanAnt();
}
```

Abstraction pada greenfoot pada kasus ini menggunakan contoh pembuatan skor permainan.

Tambahkan property pada class Bee dan juga konstruktor seperti berikut ini:



The screenshot shows the Greenfoot code editor with the class 'Bee' selected. The code is as follows:

```
import greenfoot.*;

public class Bee extends Actor
{
    private int score;

    public Bee()
    {
        score = 0;
    }

    public void act()
    {
        gerakanBee();
        hilangkanAnt();
    }
}
```

A red box highlights the constructor definition: `public Bee()`. Another red box highlights the variable declaration: `private int score;`.

Kemudian tambahkan instruksi atau perintah berikut pada metode hilangkan Ant.

```
public void hilangkanAnt()
{
    if(isTouching(Ant.class))
    {
        //Greenfoot.playSound("hai.wav");
        removeTouching(Ant.class);
        score = score + 20;
        getWorld().showText("Score: "+score,80,25);
    }
}
```

Agar instruksi ini dapat digunakan berkali-kali perlu dipisahkan menjadi sebuah metode, ini ditujukan untuk menghilangkan penggunaan kode secara berulang, sehingga cukup memanggil metodenya saja.

Tambahkan sebuah metode baru dengan instruksi yang sama dengan sebelumnya.

```
private void addScore()
{
    score = score + 20;
}
```

Kemudian lakukan pemanggilan metode addScore pada metode hilangkanAnt.

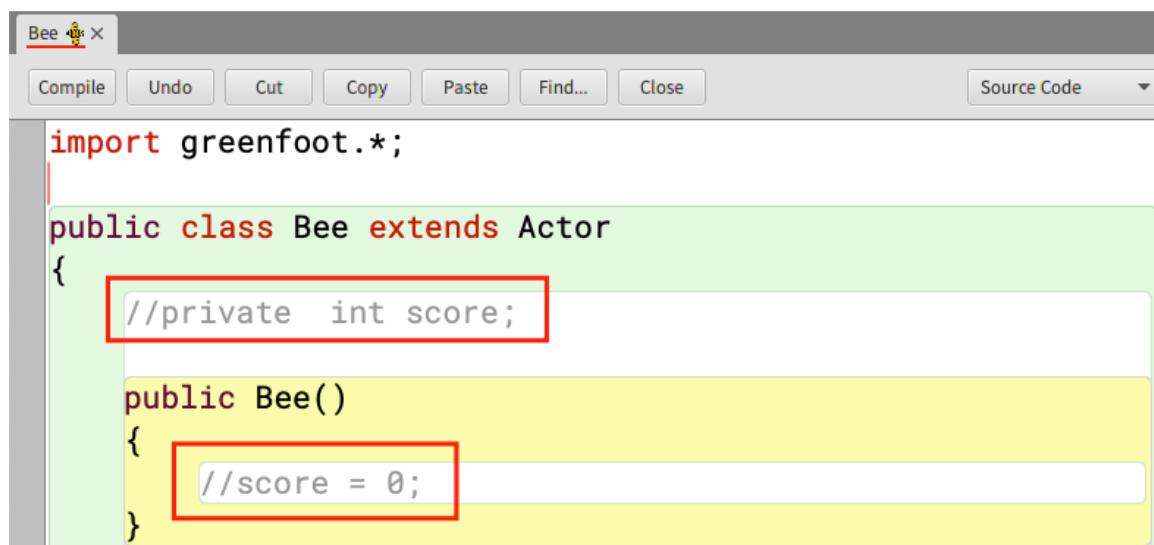
```
public void hilangkanAnt()
{
    if(isTouching(Ant.class))
    {
        //Greenfoot.playSound("hai.wav");
        removeTouching(Ant.class);
        //score = score + 20;
        addScore();
        getWorld().showText("Score: "+score,80,25);
    }
}
```

Instruksi ini masih dapat dimodifikasi dengan menambahkan parameter pada metode addScore dan pada proses pemanggilan metode addScore ditambahkan argument.

```
public void hilangkanAnt()
{
    if(isTouching(Ant.class))
    {
        //Greenfoot.playSound("hai.wav");
        removeTouching(Ant.class);
        //score = score + 20;
        addScore(20);
        getWorld().showText("Score: "+score, 80, 25);
    }
}

private void addScore(int point)
{
    score = score + point;
}
```

Selanjutnya ada casting, untuk dapat menggunakan casting pada greenfoot lakukan penghapusan pada metode addScore, property private int score dan juga score = 0 pada konstruktor serta .



The screenshot shows the Greenfoot IDE interface with the title bar "Bee". Below the title bar is a toolbar with buttons for "Compile", "Undo", "Cut", "Copy", "Paste", "Find...", "Close", and "Source Code". The main code editor area contains the following Java code:

```
import greenfoot.*;

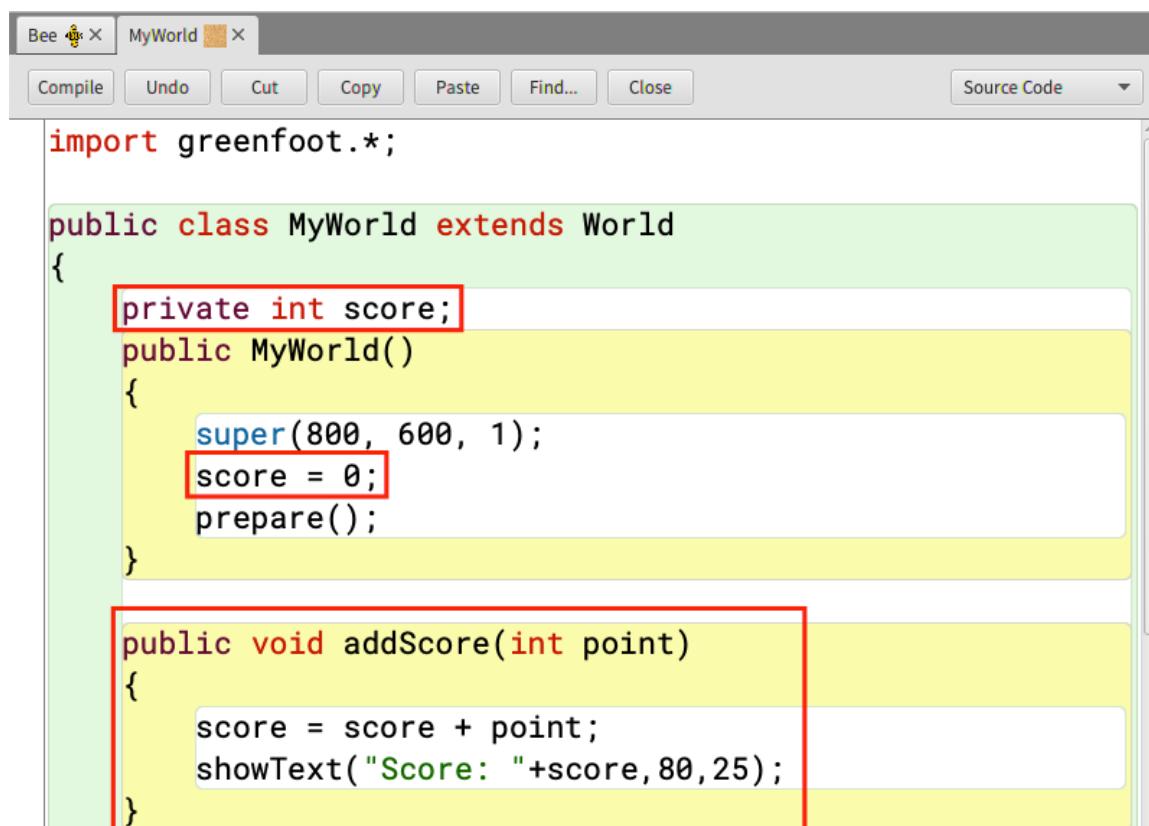
public class Bee extends Actor
{
    //private int score;

    public Bee()
    {
        //score = 0;
    }
}
```

The code is syntax-highlighted, with "greenfoot.\*;" in red, "Bee" and "Actor" in blue, and other keywords in black. Two specific lines of code are highlighted with red boxes: the declaration of the private integer variable "score" and its initialization to 0 in the constructor. These lines are commented out with double slashes.

```
/**private void addScore(int point)
{
    score = score + point;
}*/
```

Selanjutnya tambahkan perintah berikut pada class MyWorld:



The screenshot shows the Greenfoot IDE interface with two tabs open: 'Bee' and 'MyWorld'. The 'MyWorld' tab is active, displaying the following Java code:

```
import greenfoot.*;

public class MyWorld extends World
{
    private int score;
    public MyWorld()
    {
        super(800, 600, 1);
        score = 0;
        prepare();
    }

    public void addScore(int point)
    {
        score = score + point;
        showText("Score: "+score, 80, 25);
    }
}
```

The code is color-coded: 'import' and 'public' are red, 'MyWorld' and 'World' are blue, and variable names like 'score' and method parameters like 'point' are black. The declaration of 'score' and its assignment in the constructor, as well as the entire definition of the 'addScore' method, are highlighted with a red rectangular box.

Berikutnya tambahkan instruksi untuk melakukan casting pada metode hilangkanAnt di dalam class Bee seperti berikut:

```
public void hilangkanAnt()
{
    if(isTouching(Ant.class))
    {
        //Greenfoot.playSound("hai.wav");
        removeTouching(Ant.class);
        //score = score + 20;
        //addScore(20);
        //getWorld().showText("Score: "+score,80,25);
        MyWorld myworld = (MyWorld)getWorld();
        myworld.addScore(20);
    }
}
```

Contoh penerapan perulangan pada Greenfoot dapat dilakukan dengan menambahkan instruksi berikut pada class world.

```
private void prepare()
{
    Bee bee = new Bee();
    addObject(bee,600,200);

    int i = 0;
    while(i < 100)
    {
        int koor_x = Greenfoot.getRandomNumber(400);
        int koor_y = Greenfoot.getRandomNumber(600);
        addObject(new Ant(), koor_x, koor_y);
        i = i + 1;
    }
}
```

Membuat variabel pada Greenfoot seperti berikut ini:

```
import greenfoot.*;  
  
public class Bee extends Actor  
{  
    private GreenfootImage imageright;  
    private GreenfootImage imageleft;  
    //private int score;  
    boolean isTurning = false;  
  
    public Bee()  
    {  
        imageright = new GreenfootImage("bee_right.png");  
        imageleft = new GreenfootImage("bee_left.png");  
    }  
}
```

The code editor window titled "Bee" shows the class definition. A red box highlights the declaration of two private variables: "imageright" and "imageleft". Another red box highlights the constructor "Bee()" which initializes these variables with images named "bee\_right.png" and "bee\_left.png".

\*import gambarnya terlebih dahulu

```
if(Greenfoot.isKeyDown("RIGHT"))  
{  
    turn(5);  
    isTurning = true;  
    setImage(imageleft);  
}  
  
if(Greenfoot.isKeyDown("LEFT"))  
{  
    turn(-5);  
    isTurning = true;  
    setImage(imageright);  
}
```

The code editor window shows the logic for handling keyboard events. It checks if the right key is down, turns the actor 5 degrees, sets the "isTurning" flag to true, and changes the image to "imageleft". It does the same for the left key, turning -5 degrees, setting "isTurning" to true, and changing the image to "imageright". Both sections of code are enclosed in red boxes.

Menggunakan array pada greenfoot dapat dilakukan dengan membuat variabel array dan menambahkan instruksi pada metode konstruktor seperti berikut ini:

```
import greenfoot.*;  
  
public class Bee extends Actor  
{  
    private GreenfootImage[] images = new GreenfootImage[4];  
    private int current_image;  
  
    public Bee()  
    {  
        //images[0] = new GreenfootImage("bee1.png");  
        //images[1] = new GreenfootImage("bee2.png");  
        //images[2] = new GreenfootImage("bee3.png");  
        //images[3] = new GreenfootImage("bee4.png");  
        //ubah menjadi seperti ini  
        int i = 0;  
        while(i < 4)  
        {  
            images[i] = new GreenfootImage("bee"+(i+1)+".png");  
            i = i+1;  
        }  
        current_image = 0;  
    }  
}
```

Kemudian tambahkan metode untuk membuat animasi gerakan sayap (terbang) pada Bee seperti berikut ini:

```
private void animasiBee()  
{  
    if(current_image == 3)  
    {  
        current_image = 0;  
    }  
    else  
    {  
        current_image++;  
    }  
    setImage(images[current_image]);  
}  
  
public void act()  
{  
    animasiBee();  
    if(Greenfoot.isKeyDown("UP"))  
    {  
        move(5);  
    }  
  
    if(Greenfoot.isKeyDown("DOWN"))  
    {  
        move(-5);  
    }  
}
```