

Cara membuat game berbasis Java menggunakan Greenfoot bag.2

Muhidin Saimin

muhidin@smapluspgri.net

muhidins@gmail.com

<http://muhidin.co.cc>

ym: naufal_mr

Lisensi Dokumen:

Copyright © 2008 smapluspgri.net

Seluruh dokumen di smapluspgri.net dapat digunakan, dimodifikasi dan disebarluaskan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari smapluspgri.net

Pendahuluan

Melanjutkan materi kita sebelumnya yang tertunda. Untuk greenfoot dapat dijalankan dengan baik di system operasi windows maupun linux. Materi kali saya ulang mulai dari awal namun dengan menggunakan linux dan kali ini menggunakan PesatNux 1.2 (hasil remastering dari Ubuntu 8.04.1, UbuntuME dan BlankOn 4.0). Perlu diingat java adalah program yang case sensitif artinya beda perintah antara huruf Kapital dan bukan sangat berpengaruh.

Syarat Program

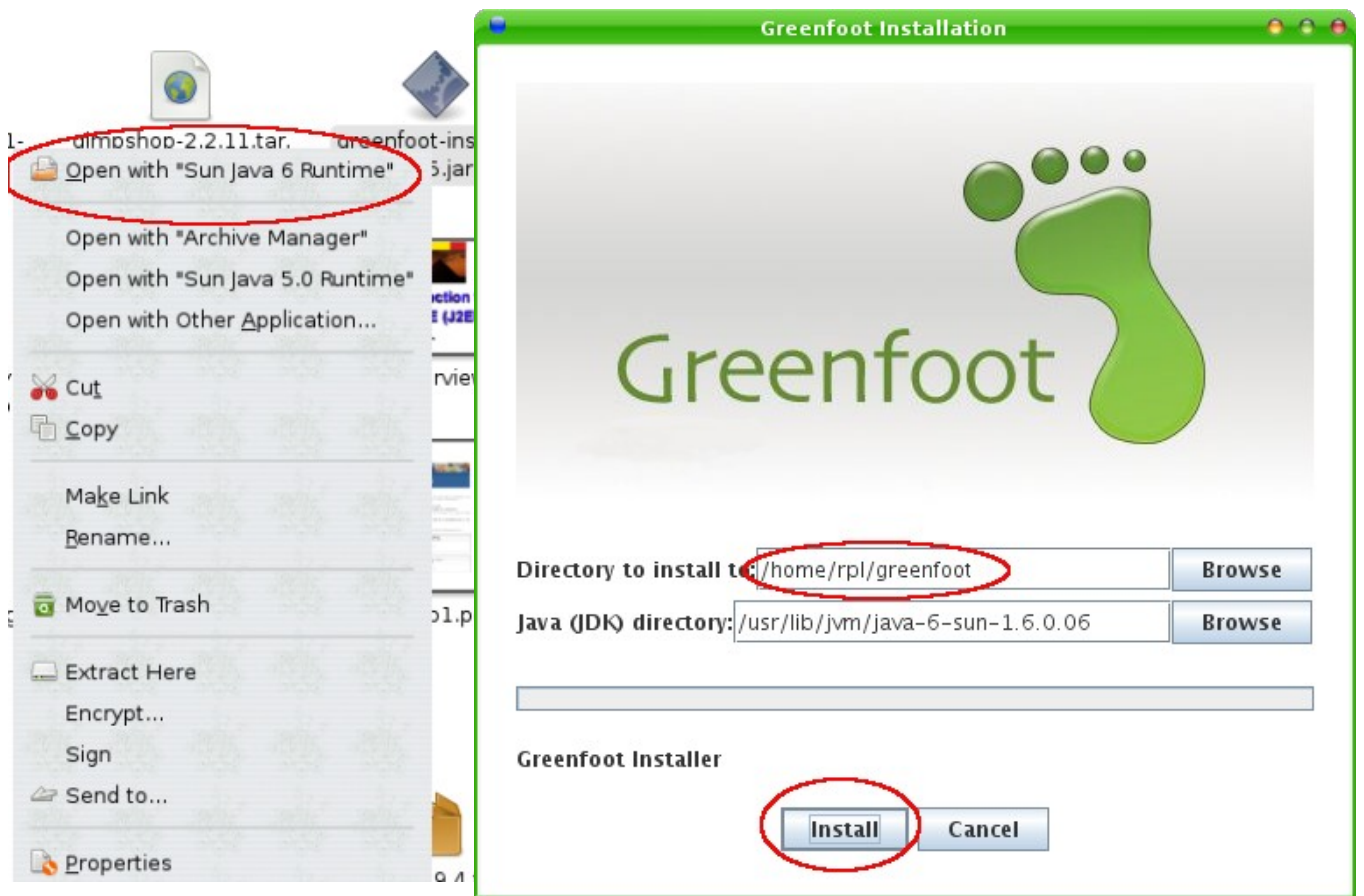
Untuk dapat menjalankan program greenfoot dikomputer anda harus sudah ada program Java 5 (JDK 1.5) atau diatasnya bisa dengan sun-java atau open-java (open-jdk). Kali ini penulis menggunakan Sun Java 6. Sedangkan program greenfoot kali ini sudah mencapai versi 1.4.6 dan dapat diunduh pada <http://greenfoot.org/download/>

Cara Instalasi

Instal java menggunakan synaptic atau jika diconsole (jika menggunakan debian base) – syarat komputer tersambung ke-internet atau memiliki DVD repo

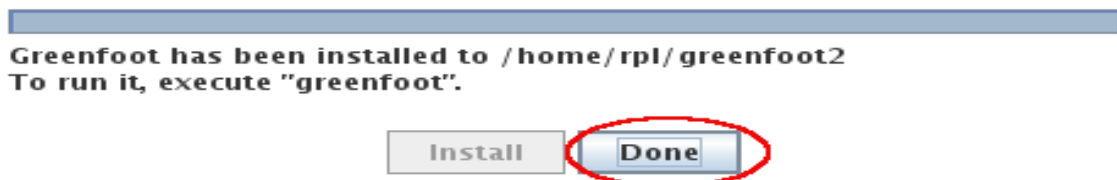
```
$ apt-get install sun-java6-bin sun-java6-jre sun-java6-jdk
```

Untuk greenfoot klik kanan di nama file lalu pilih Sun Java 6 Runtime seperti gambar 1. Lanjutkan dengan memilih tempat instalasi diletakan dan versi java yang digunakan, klik install seperti gambar 2 tunggu hingga selesai lalu pilih done seperti gambar 2a.



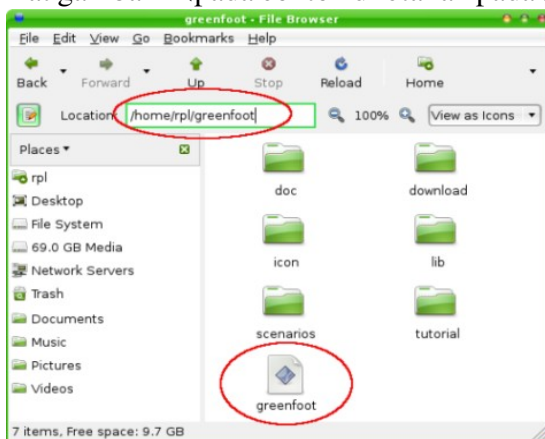
Gambar 1. Klik Kanan pada file pilih Open with Sun Java 6 Runtime

Gambar 2. Pilih folder tempat menginstall greenfoot dan dimana JDK terinstall



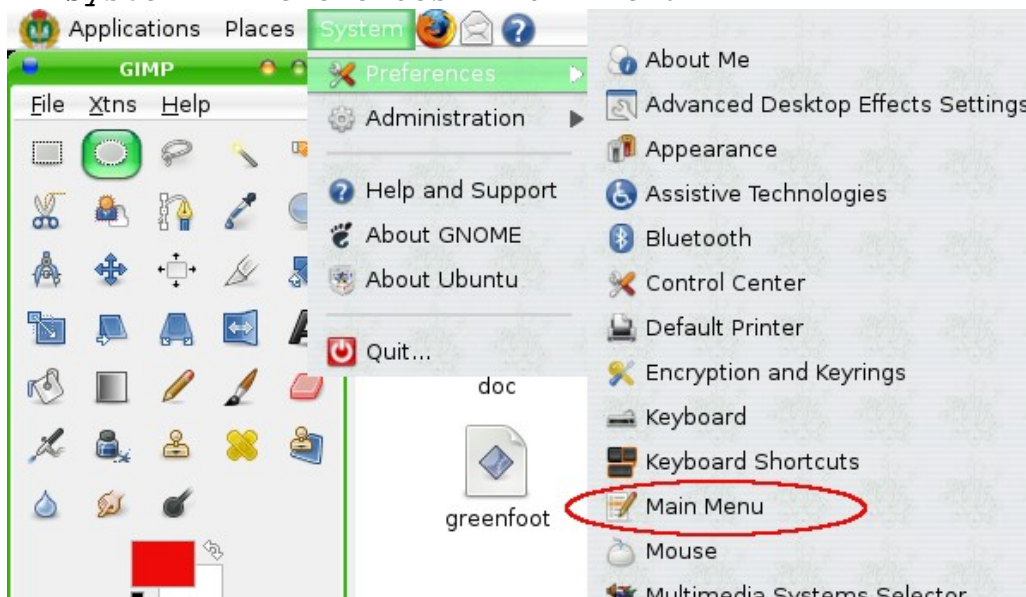
Gambar 2a. Pilih Done

Proses instalasi telah selesai dan greenfoot bisa kita jalankan sesuai tempat instalasi yg dipilih lihat gambar 3 (pada contoh diletakan pada /home/rpl/greenfoot)



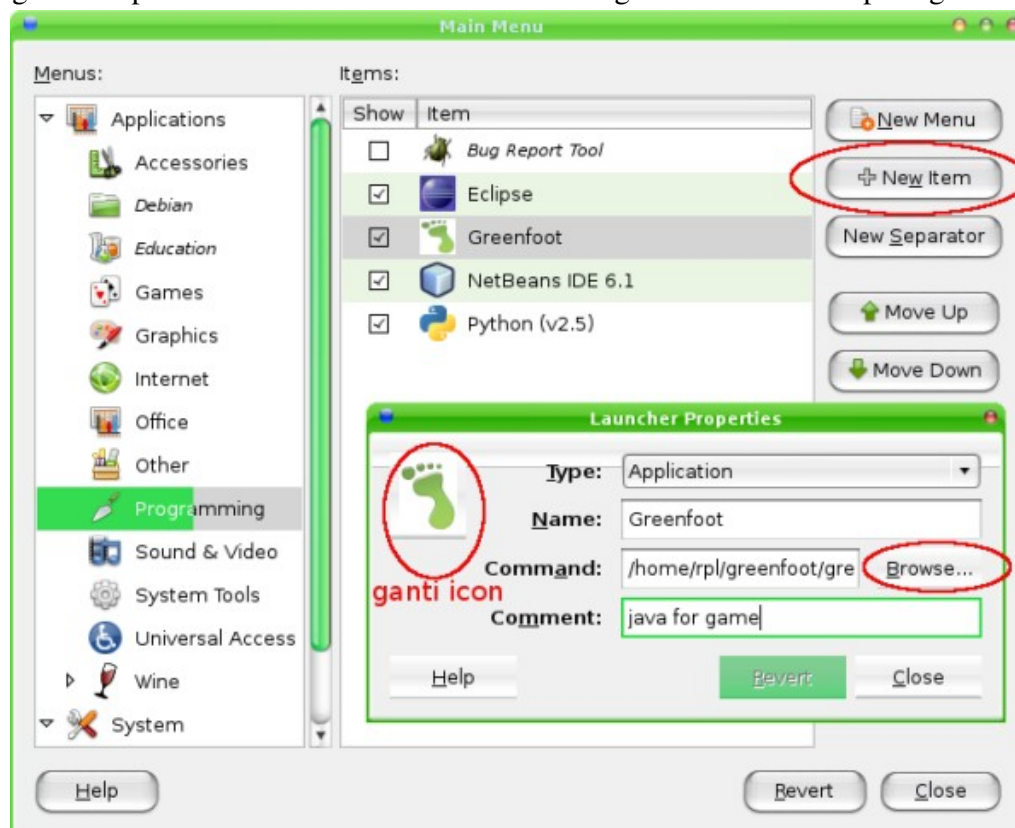
Gambar 3. Folder hasil instalasi greenfoot.

Untuk memudahkan kita dalam menjalankan program greenfoot buatlah link pada menu utama misalnya di Applications > Programming > Greenfoot langkah-langkahnya adalah klik System > Preferences > Main Menu



Gambar 4. Klik Main Menu

Letakan di menu Programming, lanjutkan dengan mengklik New Item dan isi dengan tulisan yang ingin ditampilkan (Greenfoot) lalu isi dimana file greenfoot berada seperti gambar 5



Gambar 5. Berinama menu dan pilih file eksekusi

Jalankan greenfoot sesuai dengan menu yang dibuat spt gbr. 6 lanjutkan dengan memilih new scenario 6a

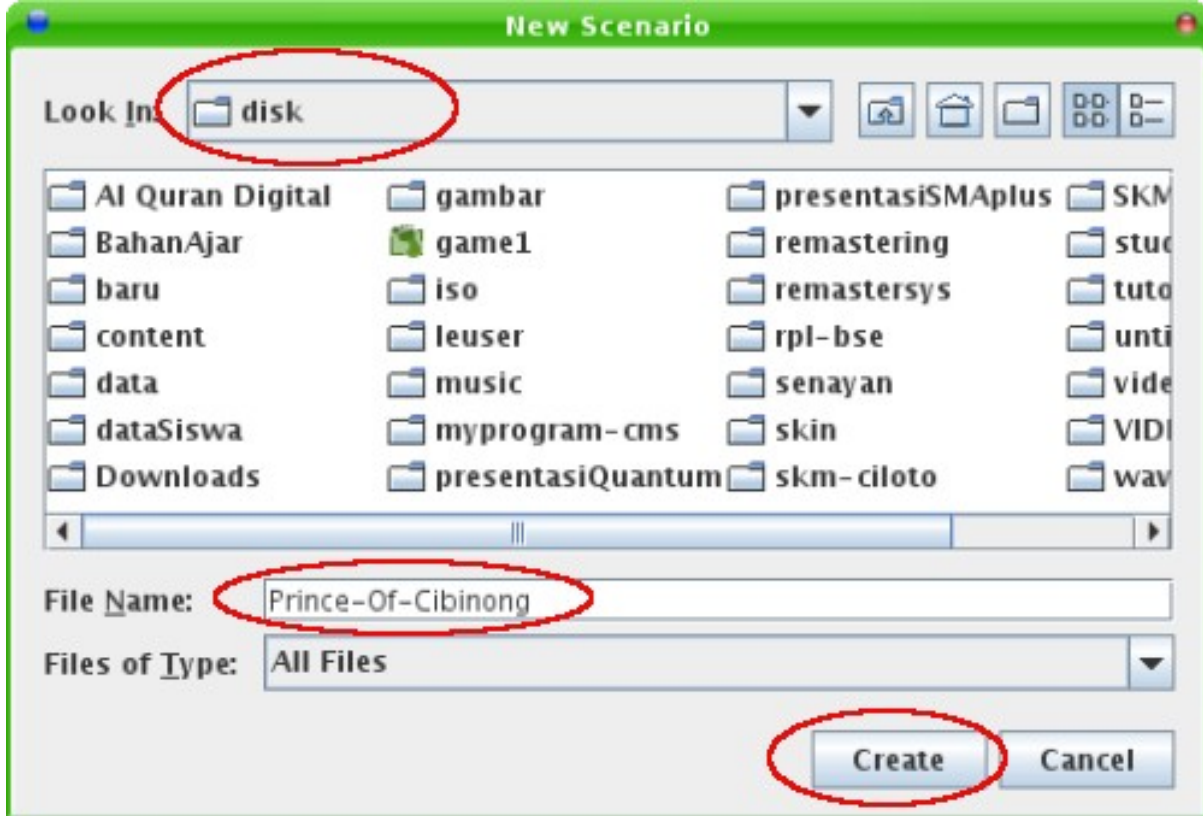


Gambar 6. Menu Greenfoot



Gambar 6a. Gambar menu greenfoot jika pertama kali dijalankan

Beri nama skenario baru misalnya: Prince-of-Cibinong seperti gambar 7 lalu klik tombol Create. Perhatikan dimana filenya dimana kita letakkan. Contoh kali ini diletakan pada partisi khusus data (folder media) di /media/disk/Prince-Of-Cibinong

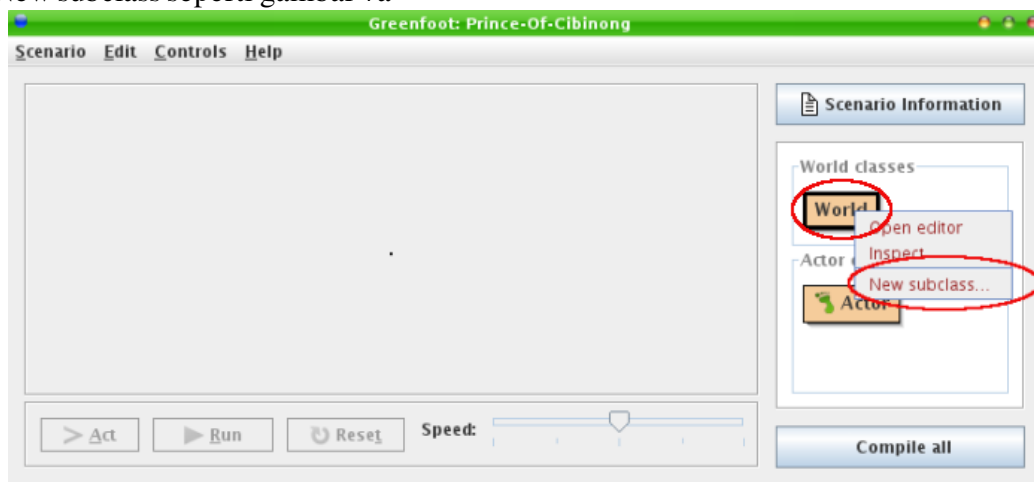


Gambar 7. Beri nama sesuai keinginan (jangan lupa letaknya difolder apa)

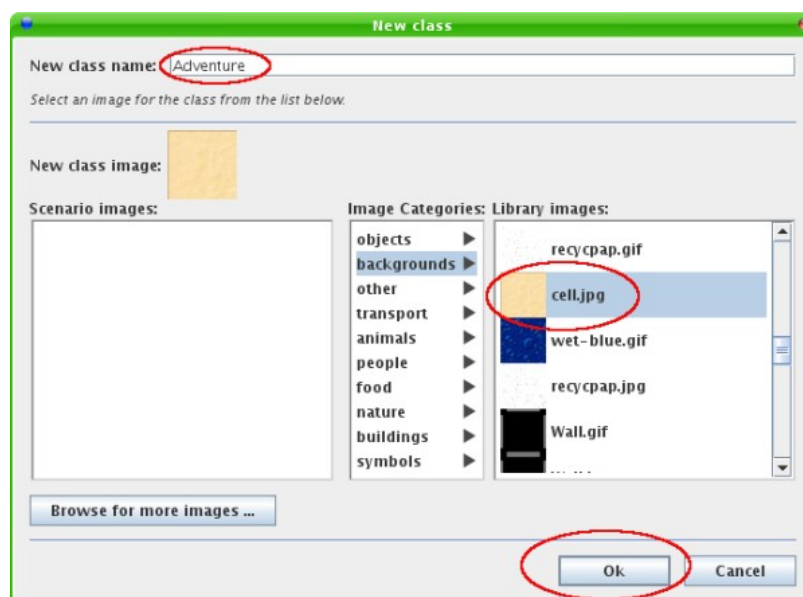
Dan akan muncul tampilan greenfoot, lanjutkan dengan membuat skenario baru. tentukan skenarionya seperti apa untuk bahasan kali ini adalah :

“Sang Pangeran (Prince) berusaha untuk menyelamatkan sang Putri (Princess) yang tertawan namun mempunyai tantangan untuk bisa melawan 5 macam musuh. Pangeran dapat digerakan dengan tombol panah dan dapat menembak dengan tombol x dan z. Musuh1 bergerak dari kiri kekanan dan terus berputar sambil turun, Musuh2 bergerak dari kanan ke kiri dan terus bergerak ke atas. Musuh3 bergerak dari atas ke bawah dan terus bergeser ke kanan. Musuh4 bergerak dari bawah ke atas dan terus bergeser ke kiri. Musuh5 selalu menjaga sang Putri di pojok atas.”

Lanjutkan dengan membuat setting lokasi (World) caranya klik kanan di tombol World lalu pilih New subclass seperti gambar 7a

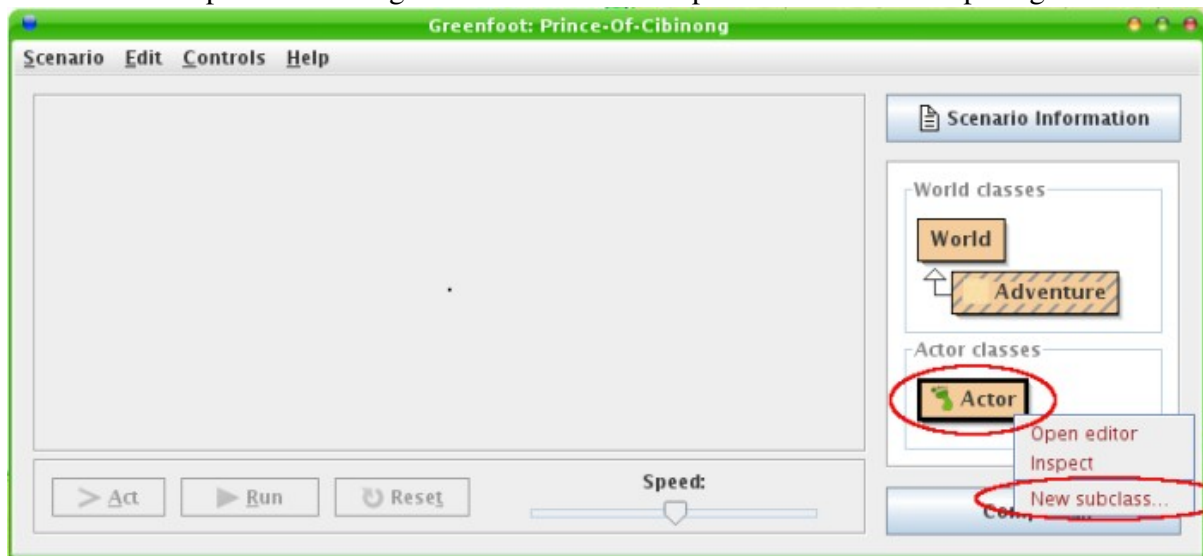


Gambar 7a. Buat kelas baru pada World (untuk menentukan background dan lebar layar)

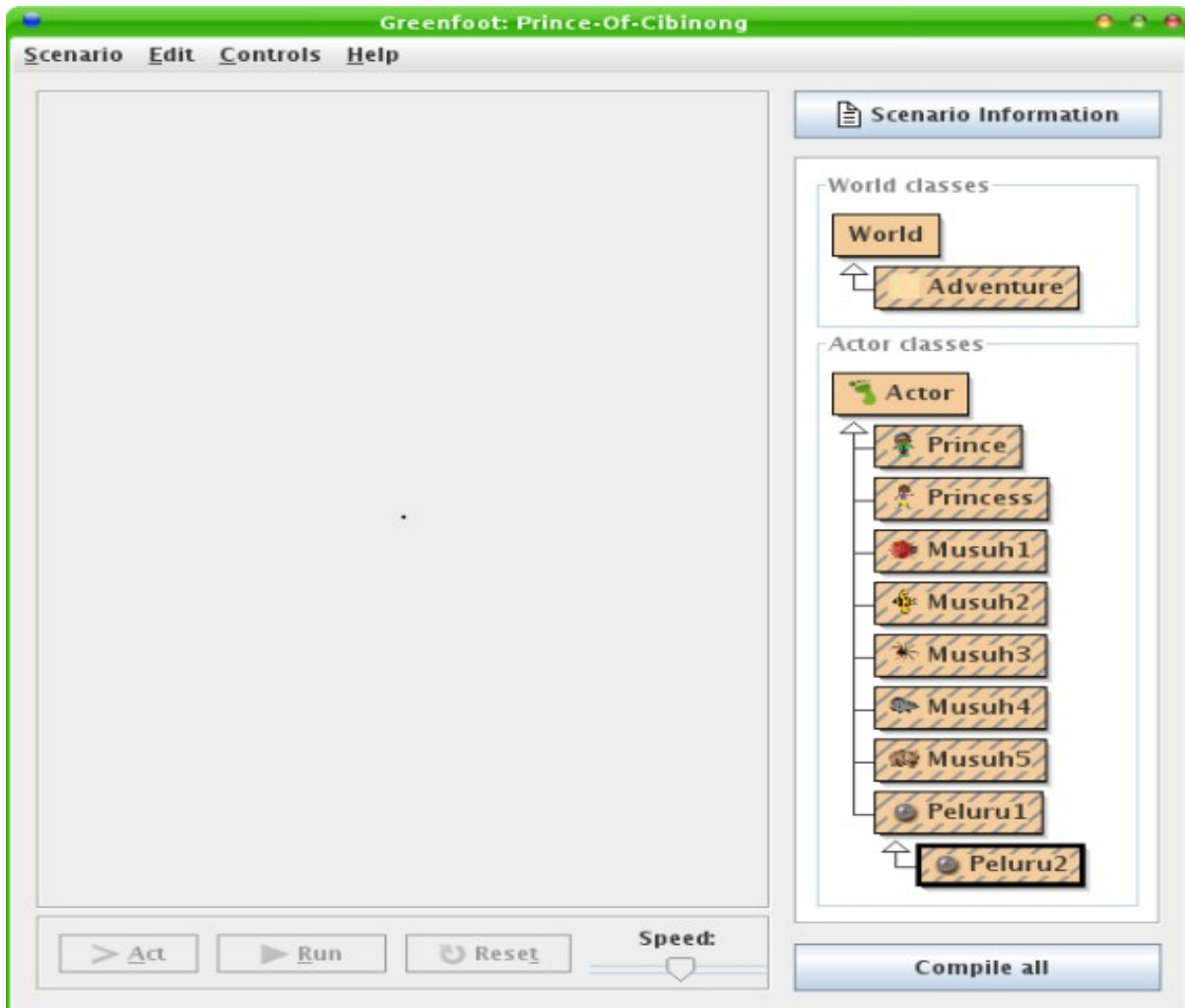


Gambar 8. Beri nama kelas baru dan pilih background yang dipakai

Buat kelas baru pada actor dengan cara klik kanan lalu pilih New subclass seperti gambar 9



Gambar 9. Buat kelas baru pada Actor



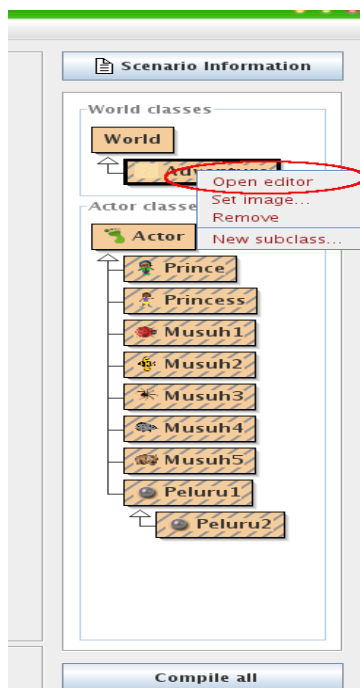
Gambar 10. Buat 9 object seperti gambar
Peluru2 dibuat dengan cara klik kanan pada actor Peluru 1

Jika kita lihat pada file manager, folder dan file yang terbentuk disana terlihat scenario yang kita buat berbentuk folder dan didalamnya ada 3 folder yang otomatis terbentuk, yaitu : *greenfoot*, *images* dan *sound*. Fungsi folder images adalah untuk menyimpan gambar yang akan kita gunakan, sedangkan folder sound untuk menyimpan file suara yang kita pakai untuk game buatan kita. Folder images dan sound awalnya masih kosong. Awal file yang terbentuk adalah file yang berkhiran .java seperti gambar 11

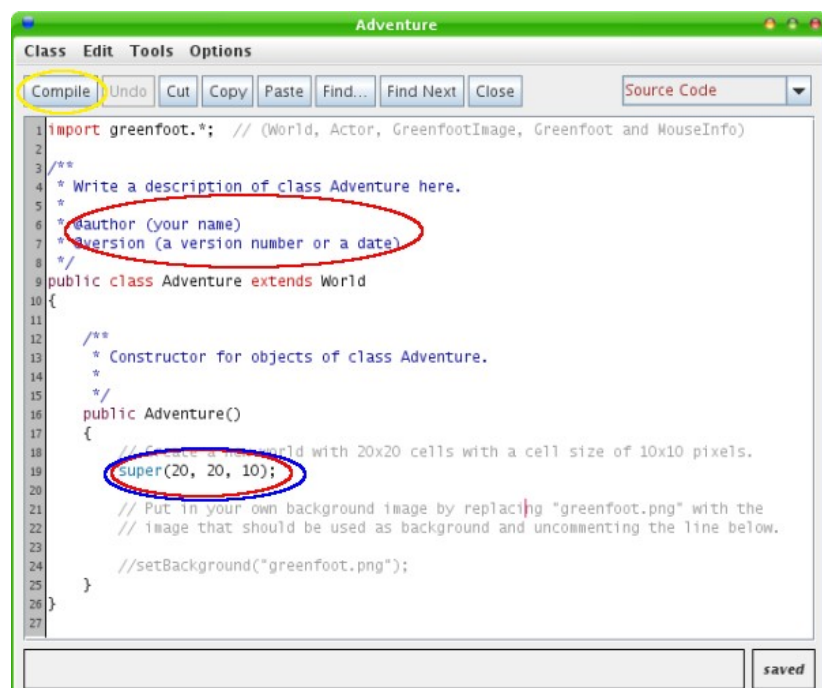


Gambar 11. Folder dan file yang terbentuk pada Pembuatan game greenfoot

Ok, kita mulai untuk membuat codingnya, caranya klik kanan di Adventure lalu pilih Open editor seperti gambar 12 dan akan muncul layar coding seperti gambar 13.



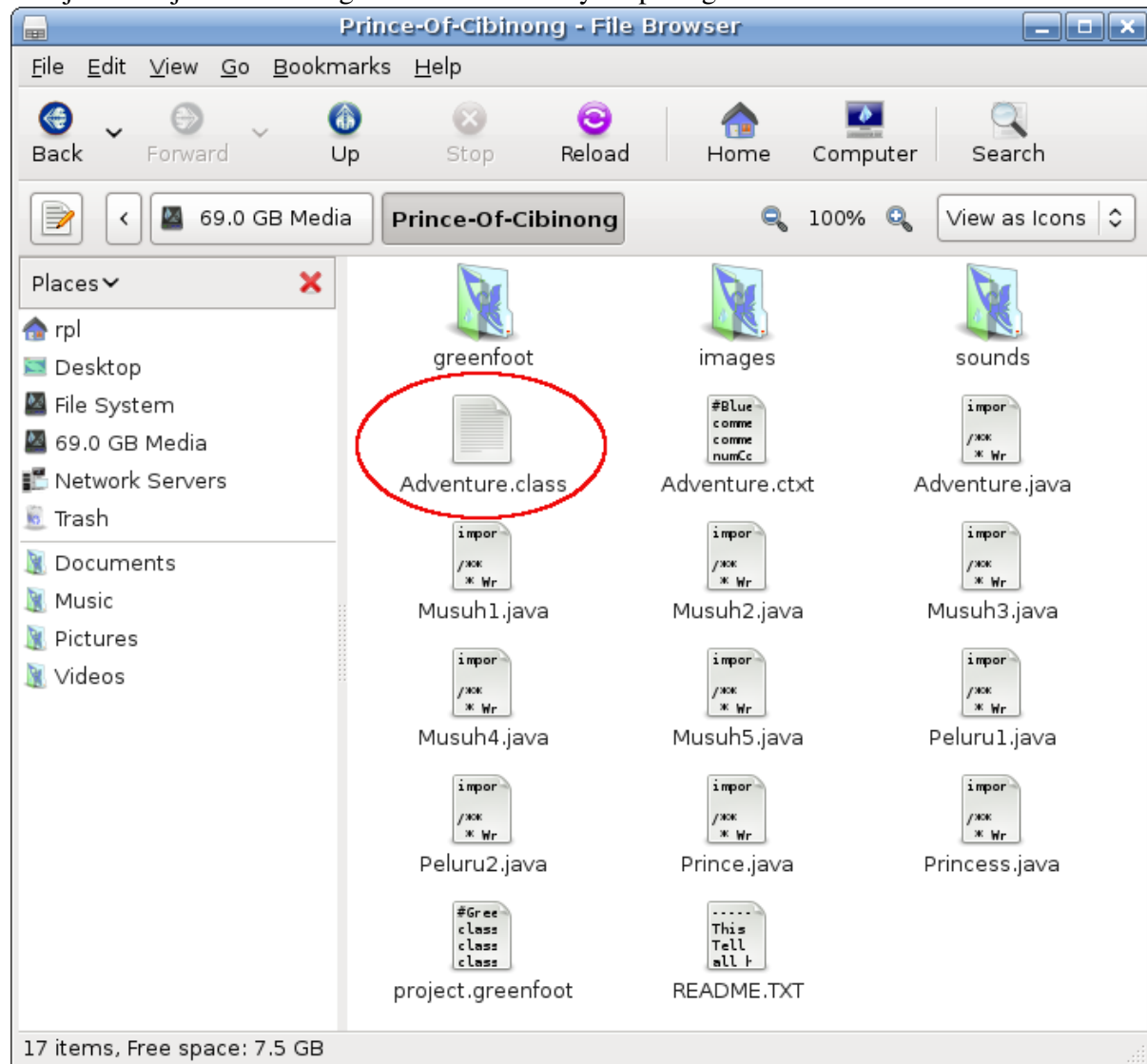
Gambar12. Buka pengedit kode pada *Class World*



Gambar 13. Layar kode

Biasakan untuk mengisi data pembuat contoh @author (Muhidin S) version (1.0.0). Tanda /** adalah tanda dimulainya baris komentar, baris dibawahnya jika langsung diter akan otomatis muncul tanda bintang (*) didepannya dan diakhir baris komentar ditutup dengan tanda */. Selain cara diatas untuk membuat baris komentar bisa dengan tanda // (garis miring 2 kali) maka baris tersebut dianggap sebagai baris komentar dengan warna tulisan redup.

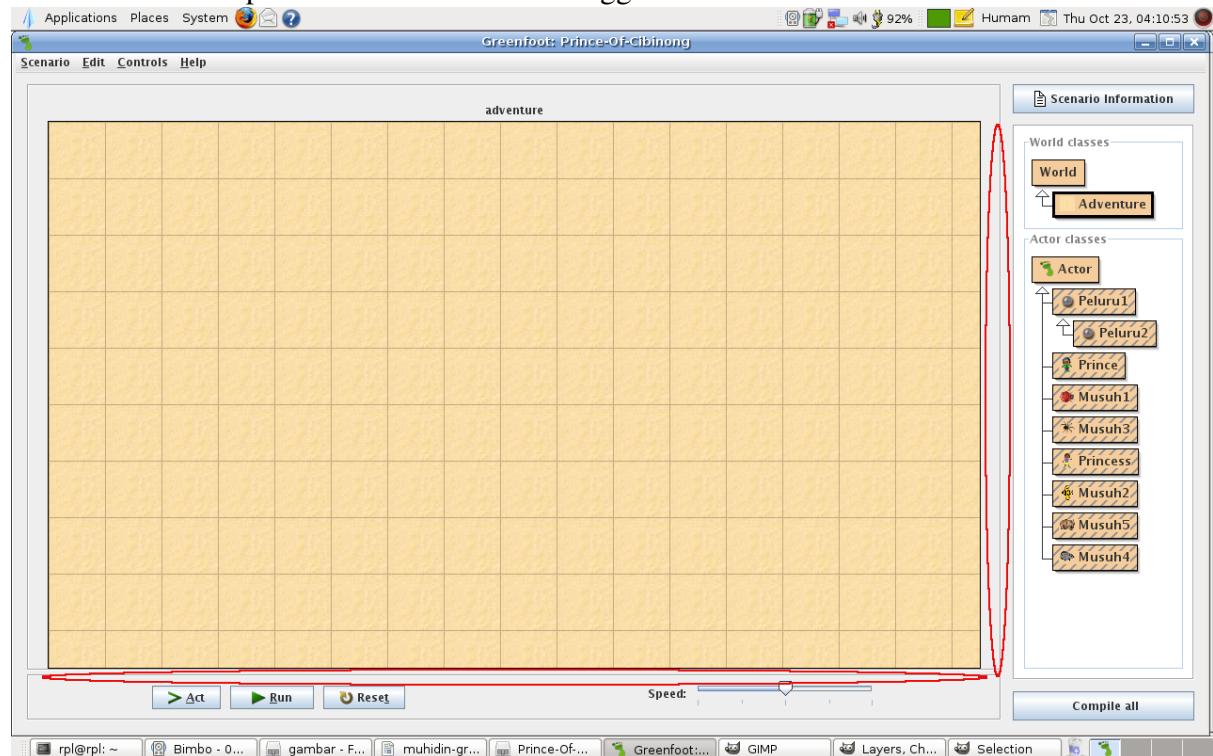
Lanjutkan dengan langsung mengklik Compile (dilingkari warna kuning) lalu check perubahan file apa yg bertambah pada file manager. Button Compile diklik maka greenfoot akan membuat file .java menjadi .class dengan menambahkannya seperti gambar 14



Gambar 14. Penambahan file Adventure.class setelah Compile diklik.

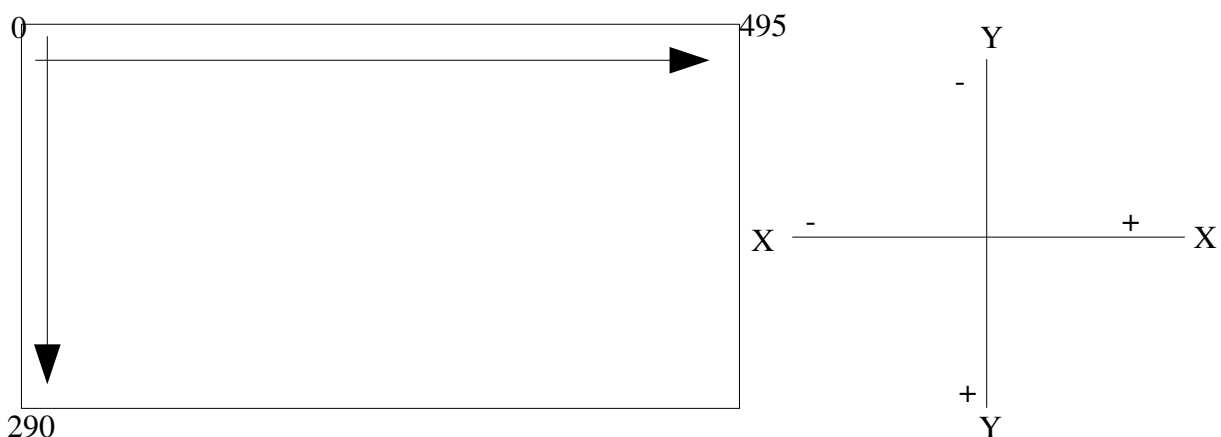
Kembali ke gambar 13, buka editor untuk Adventure dan edit coding `super (20 , 10 , 10) ;` menjadi `super (495 , 290 , 2) ;` ; angka 495 adalah lebar layar pada sumbu x (kanan-kiri) 290 tinggi (atas-bawah) dan 2 adalah lebar cell. Bisa diibaratkan ruangan ukuran 5mx3m dengan lebar keramik ukuran 20cmx20cm (ini cell). Makin kecil ukuran cell makin halus pergerakan Actor dan makin besar angka untuk lebar dan tingginya (sumbu X dan sumbu Y). Isi cell dengan 2 dan sesuaikan angka untuk lebar (X) dan tinggi (Y) dengan ukuran resolusi

monitor ditempat anda (coba dan sesuaikan) hingga seperti gambar 15, tidak membentuk scroll keatas-bawah maupun kiri-kanan. Penulis menggunakan resolusi 1280x800.



Gambar 15. Gambar penuh sehingga nyaman untuk mencoba dan mengedit

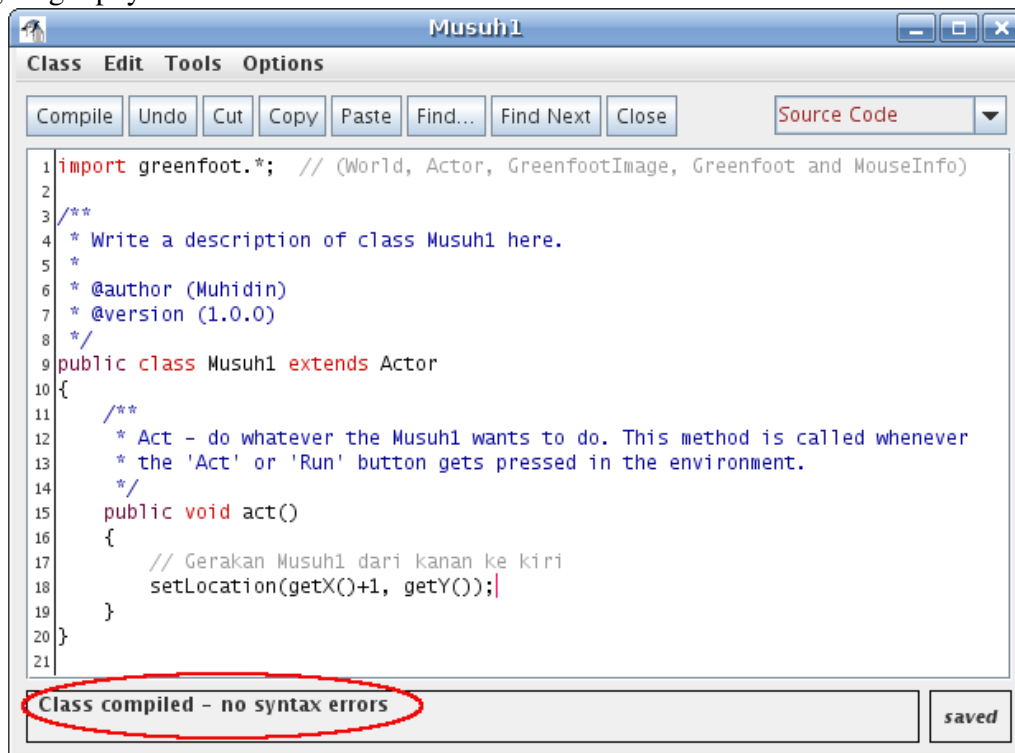
Lanjutkan dengan membuat coding untuk Musuh1 yang akan bergerak dari kiri kekanan. Perintah untuk menyatakan perpindahan lokasi adalah `setLocation(getX(),getY());` `getX()` = sumbu X (kiri-kanan) `getY()` = sumbu Y (atas-bawah). Lihat gambar 16 analogi sumbu X dan sumbu Y.



Gambar 16. Analogi sumbu X dan sumbu Y pada program greenfoot

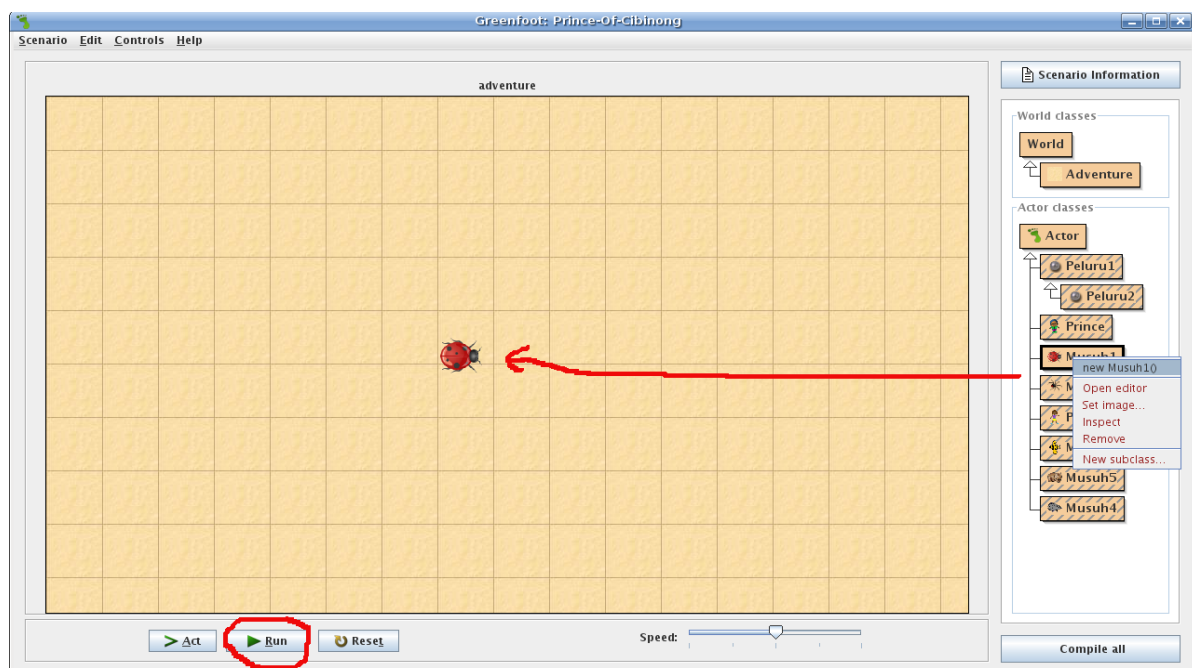
Greenfoot memandang titik awal (0,0) berada pada kiri atas gambar makin kekanan maka nilai sumbu X makin bertambah, semakin kebawah nilai sumbu Y makin bertambah dan sebaliknya. Dengan kata lain sumbu X ditambah akan bergerak kekanan dikurangi bergerak kekiri. Sumbu Y ditambah bergerak kebawah dikurangi bergerak keatas. Maka coding untuk menggerakkan Musuh1 adalah :

setLocation(getX()+1,getY());
coding lengkapnya adalah



Gambar 17. Coding Musuh1 no syntax errors

Klik button Compile jika coding telah benar maka akan muncul tanda dikiri bawah seperti gambar 17 Class compiled – no syntax errors (file class telah terbentuk dan tak ada perintah yang salah). Lanjutkan dengan proses testing. Kembali ke layar utama greenfoot lalu klik kanan pada class Actor Musuh1 dan pilih new Musuh1 () kemudian pindahkan object Musuh1 ketengah-tengah. Lanjutkan dengan menekan tombol Run seperti gambar 18.



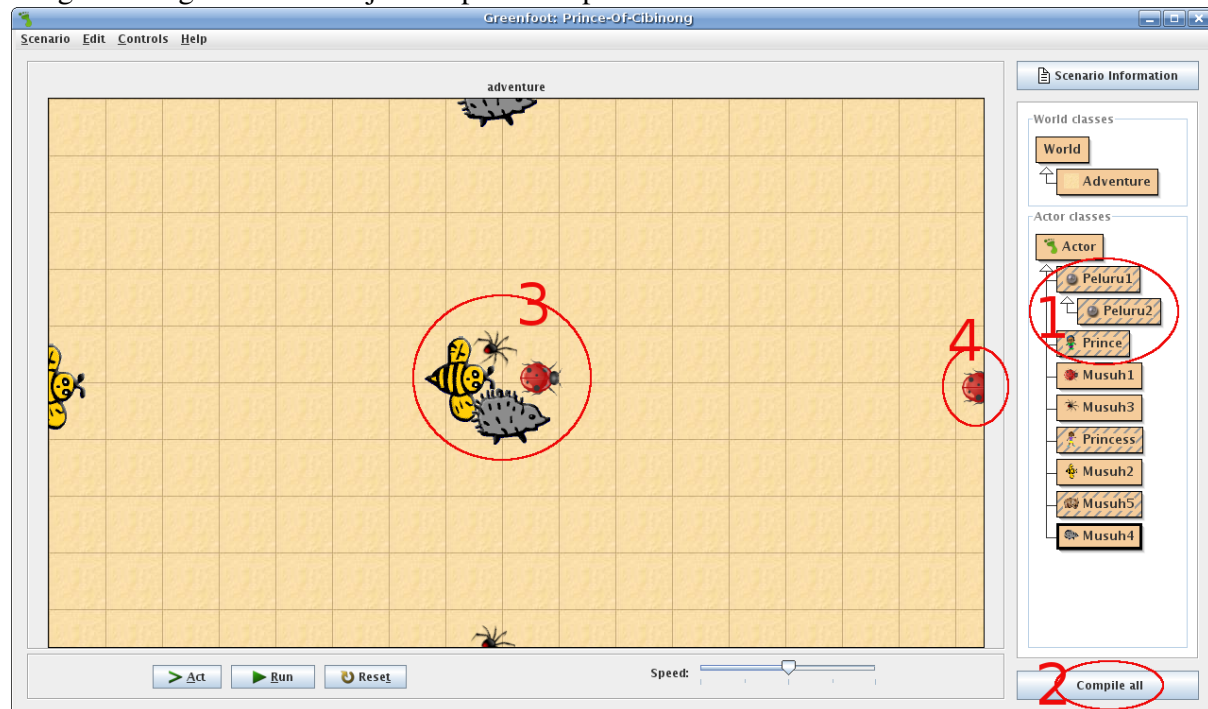
Gambar 18. Testing coding Musuh1

Act = Action menjalankan program sesuai coding act (step by step)

Run = Memulai menjalan program hingga ada instruksi selesai atau Reset ditekan

Reset = Menghentikan program

Lanjutkan membuat coding untuk Musuh2 yang bergerak dari kiri kekanan hanya sumbu X yang dirubah sumbu Y tetap `setLocation (getX() -1, getY())`. Musuh3 bergerak dari atas kebawah sumbuX tetap sumbu Y ditambah `setLocation (getX() , getY() +1) ;`. Musuh4 bergerak dari bawah keatas `setLocation (getX() , getY() -1) ;`. Lanjutkan dengan testing sudahkah berjalan seperti harapan ?



Gambar 19 Testing gerakan Musuh

Apabila diklik kanan new Musuh1() tidak ada kemungkinannya file belum dicompile ada tanda arsiran pada class yang belum dicompile spt gbr 19 no. 1 lakukan Compile All spt gbr 19 no.2. Lakukan testing dengan cara meletakkan file Musuh1-4 ditengah-tengah seperti gbr 19 no.3 dan perhatikan gerakannya, jika Musuh1 telah mencapai angka maksimal, ia berhenti. Bagaimana caranya agar Musuh1 bisa menembus (jika mencapai angka maksimal kembali ke angka minimal). Coding yang perlu ditambahkan adalah :

```
setLocation (getX() +1, getY() );  
if (getX() ==494)  
{  
    setLocation (0, getY() );  
}
```

Keterangan:

`if (getX() ==494)` → apabila sumbu X mencapai angka maksimal (495-1) maka `setLocation (0, getY());` → kembalikan nilai pada sumbu X ke angka terkecil (0)

Untuk Musuh2 caranya sama hanya pada jika mencapai angka terkecil kembalikan ke angka terbesar, menjadi:

```
setLocation(getX()-1,getY());
if(getX()==0)
{
    setLocation(495,getY());
}
```

Untuk Musuh3 dan Musuh4 silahkan kerjakan dan lanjutkan dengan testing sudah benar semua atau belum ?, jika belum perbaiki hingga benar dan mengerti baru lanjutkan. Jangan dilanjutkan jika belum paham.

Lanjutkan dengan membuat gerakan pada Prince (sang pangeran), dengan aturan jika ditekan tombol panah kanan akan bergerak kekanan panah kiri bergerak ke kiri panah atas bergerak keatas dan panah bawah bergerak kebawah. Kerjakan satu persatu lalu test.

Jika menekan tombol Panah Kanan Actor Prince bergerak kekanan, codingnya

```
if(Greenfoot.isKeyDown("right"))
{
    setLocation(getX()+1,getY());
}
```

Keterangan:

`if(Greenfoot.isKeyDown("right"))` → jika tombol “panah kanan” ditekan maka ...
`setLocation(getX()+1,getY());` → gerakan Actor ke arah kanan 1 cell

buatkan untuk penekanan tombol kiri (left), atas (up) dan bawah (down) lalu testing hasilnya. Jika dibutuhkan Actor Prince bisa juga dibuat tembus.

Ketika tombol Reset ditekan Actor selalu hilang, buat agar Reset ditekan actor langsung kembali ketempatnya masing-masing. File yang perlu diedit berada pada World Class yaitu file Adventure. Letakan coding dibawah perintah super, coding yang diperlukan :

```
addObject(new Prince,7,145);
```

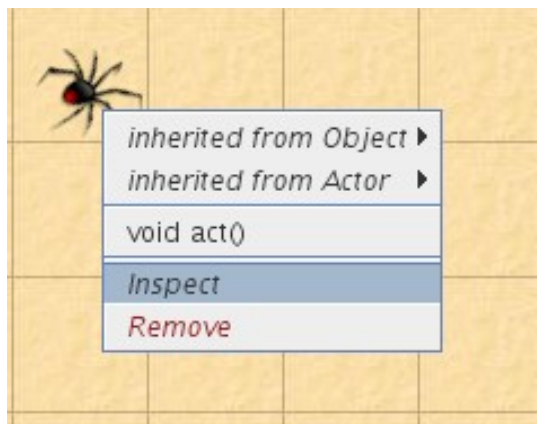
Keterangan:

`addObject(new` → tambahkan object baru (perintah dasar untuk menambahkan objek baru
`Prince` → Actor Prince (ganti Prince dengan nama actor yang lain jika diperlukan)

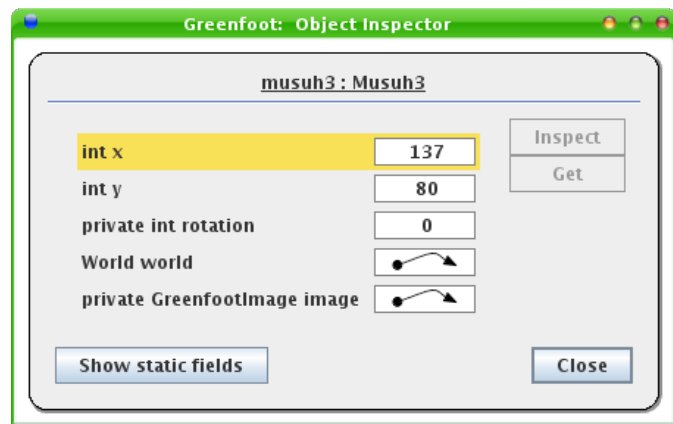
`7,145` → terletak pada sumbuX koordinat 7 (dari kiri maju 7 langkah) dan sumbu Y koordinat 145 (tengah)

Tambahkan seluruh Actor yang diperlukan, atur tata letak agar permainan lebih menarik. Untuk membuat agar Actor berada tepat ditempat yang kita inginkan seret Actor dengan dengan mengklik dan seret ketempat yang sesuai, lanjutkan dengan mengklik kanan pada aktor dan pilih inspect, maka akan muncul letak dari aktor tersebut berapa x dan y - nya terlihat jelas, buka Adventure dan tambahkan addObject untuk Actor yang dibutuhkan. Lihat gambar 20 (memunculkan pilihan inspect) dan 21 (hasil inspect).

Lalu tambahkan semua Actor ke World, agar jalannya Actor sesuai gambar edit gambarnya (Musuh2 gerakan ke kiri berarti gambarnya mesti ke arah kiri) dengan gimp atau program pengolah gambar lainnya dan sesuaikan ukuran, arah atau warna gambarnya (difolder images).



Gambar 20 klik kanan *Inspect*



Gambar 21 hasil perintah *Inspect*

Edit gambar dan tambahkan coding di World hingga seperti gambar 22



Gambar 22 hasil akhir gambar untuk Actor

Hasil file Adventure:

```
public Adventure()
{
    super(495, 290, 2);
    addObject(new Prince(), 7, 145);
    addObject(new Musuh1(), 138, 227);
    addObject(new Musuh2(), 414, 80);
    addObject(new Musuh3(), 137, 80);
    addObject(new Musuh4(), 407, 227);
    addObject(new Musuh5(), 353, 281);
    addObject(new Princess(), 467, 270);
}
```

Lanjutkan dengan coding untuk Musuh5 dengan aturan selalu selalu bergerak serong keatas dengan tujuan menghalangi Prince untuk menemui Princess.

Buka coding Actor Musuh5

```
        setLocation(getX()+2,getY()-1);
buat jika mencapai angka maksimal dari sumbu X akan kembali ke posisi semula
        if(getX()==494)
        {
            setLocation(353,281);
        }
```

Testing dan lihat hasilnya.

Jika berhasil lanjutkan dengan membuat coding di Prince dengan aturan jika ditekan panah Kanan dia akan bergerak kekanan, panah kiri bergerak ke kiri Atas gerak ke atas dan Panah bawah bergerak kebawah. Buka coding Prince lalu buat, dibawah Act().

```
        // Menggerakan sang pangeran dengan cara menekan tombol
        // panah kanan maka Prince bergerak kearah kanan
        if(Greenfoot.isKeyDown("right"))
        {
            setLocation(getX()+1,getY());
        }
        //Tekan Panah Kiri bergerak ke kiri
        if(Greenfoot.isKeyDown("left"))
        {
            setLocation(getX()-1,getY());
        }
        //Tekan Panah bawah bergerak ke bawah
        if(Greenfoot.isKeyDown("down"))
        {
            setLocation(getX(),getY()+1);
        }
        //Tekan Panah atas bergerak ke atas
        if(Greenfoot.isKeyDown("up"))
        {
            setLocation(getX(),getY()-1);
        }
    }
```

Testing dan lihat hasilnya, jika berhasil lanjutkan, jika belum cari apa permasalahannya selesaikan JANGAN LANJUTKAN, jika belum selesai.

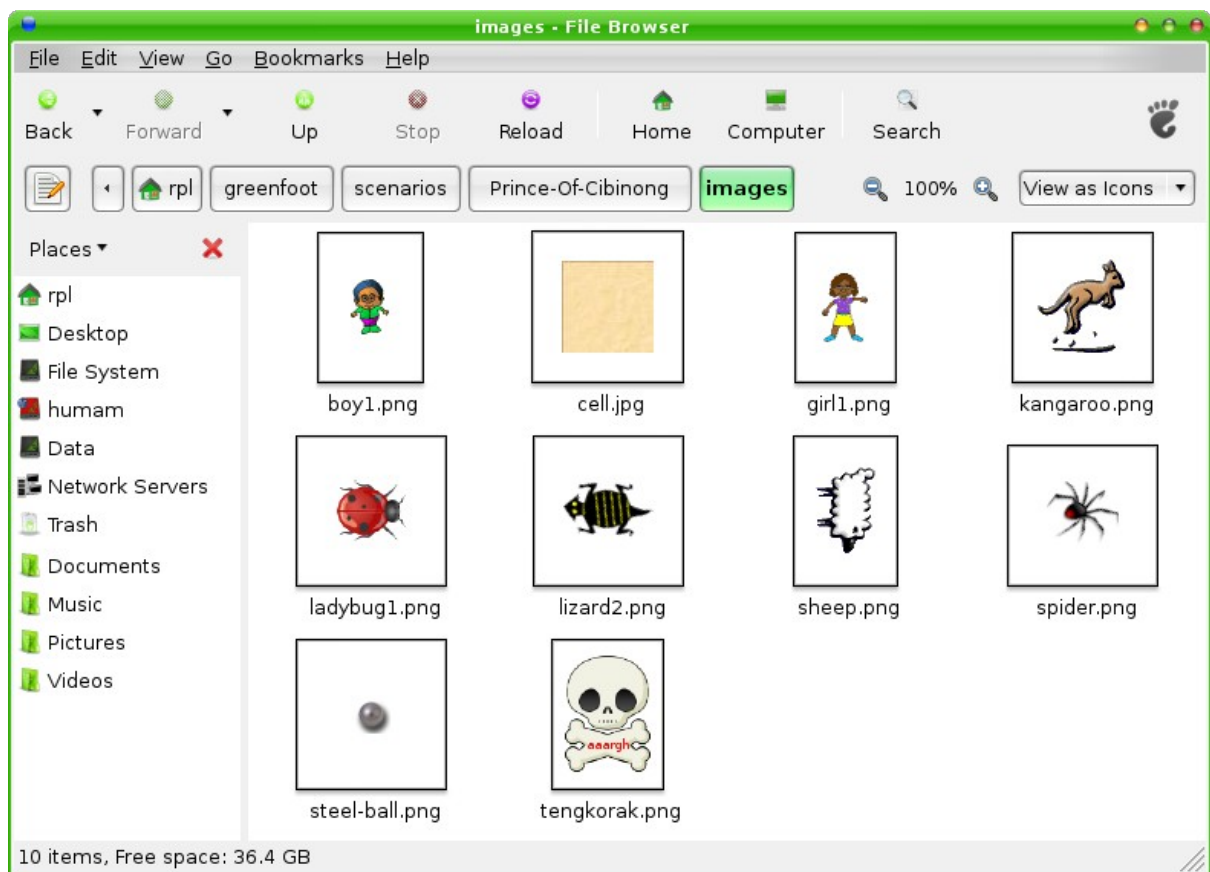
Masih di Prince: Jika Prince nabrak Musuh maka gambar Prince berubah jadi tengkorak, ada suara sirene dan permainan berhenti. Siapkan suara kalah.wav masukan di folder sounds dan gambar tengkorak dg nama tengkorak.png masukan ke folder images seperti gambar lalu buat coding dan letakan dibawah perintah menggerakan tombol tapi masih dalam class void Act.

```
        Musuh1 a=(Musuh1) getOneIntersectingObject(Musuh1.class);
```

Musuh1 = nama Actor

a = nama variabel boleh apa saja hanya untuk mengingat Musuh1 variabel a Musuh2 variabel b.

getOneIntersectingObject = object bersinggungan dengan karena ini ada dikelas Prince maka dapat diartikan jika Object/Actor Prince bertemu Musuh1



Gambar 23 tambahkan file tengkorak.png pada folder images

```
if (a!=null)
{
    setImage("tengkorak.png");
    Greenfoot.playSound("kalah.wav");
    getWorld().removeObject(a);
    Greenfoot.stopSimulation();
    return;
}
setImage("tengkorak.png"); = ganti gambar class ini (Prince) menjadi gambar tengkorak.png)
syarat gbr harus sdh ada.
Greenfoot.playSound("kalah.wav"); = Greenfoot akan memainkan suara kalah.wav (file harus
sudah ada di folder sounds, terkadang memunculkan error jika suara sedang dipakai
misalnya sedang mendengarkan lagu.
getWorld().removeObject(a); = hilangkan object yang ada pada variabel a (Musuh1)
Greenfoot.stopSimulation(); = Hentikan Permainan
```

Testing dengan menabrakan Actor Prince ke Musuh1 dan lihat hasilnya, jika berhasil lanjutkan untuk Musuh2 sampai Musuh5 dengan cara yang sama, jangan lupa testing satu-satu.

Peluru

Setelah semua selesai lanjutkan dengan membuat Senjata/Peluru untuk Prince, caranya buka coding pada Actor yang akan mengeluarkan Peluru dalam hal ini Prince.

Tambahkan coding diatas public void act()

```
private Adventure pelor;  
public void addedToWorld(World dunia)  
{  
    pelor = (Adventure) dunia;  
}
```

private = variabel ini hanya dikenali di class yang bersangkutan (Prince)

Adventure = nama class World

pelor = nama variabel (terserah tapi tidak boleh sama dengan nama Actor

public void addedToWorld(World dunia) = Tambahkan class pelor ke World

dunia = nama variabel baru

Lalu buat di dalam act (boleh paling atas atau paling bawah, yang akan memicu keluarnya peluru, yaitu jika ditekan tombol x (jangan lupa cari suara dengan nama fire.wav)

```
if(Greenfoot.isKeyDown("x"))  
{  
    pelor.addObject(new Peluru1(), getX(), getY());  
    Greenfoot.playSound("fire.wav");  
}
```

if(Greenfoot.isKeyDown("x")) = jika ditekan tombol x

pelor.addObject(new Peluru1(), getX(), getY()); = akan keluar Object baru yaitu Peluru1

letakkan sama dengan Actor/Object aktif (Prince) berada.

Greenfoot.playSound("fire.wav"); = munculkan suara fire.wav (file harus ada difolder sounds)

Setelah peluru bisa keluar atur agar pelurunya langsung bergerak, misal dari kanan ke kiri, buat codingnya di Actor Peluru1

```
setLocation(getX()+1, getY());
```

Ternyata ada sedikit masalah yaitu peluru akan meninggalkan sisa dibelakang akhir background untuk mengatasinya gunakan coding berikut:

```
if(getX()==494)  
{  
    getWorld().removeObject(this);  
    return;  
}
```

getWorld().removeObject(this): = hilangkan object ini

return: = kembali keawal (perintah sangat penting dipakai jika perintah diatasnya

menghilangkan object, jangan sampai ada perintah lain yang menggunakan Object yang telah dihilangkan, jika itu yang terjadi akan muncul pesan error).

Peluru yang baru saja dibuat belum berfungsi apa-apa karena jika Musuh tertembak tidak mati/hilang buat jika peluru terkena Musuh1-5 maka Musuh hilang dan peluru hilang dengan bunyi ah, buat coding di Peluru1 atau Musuh1 contoh disini dibuat di Peluru1, masukan file die1.wav

```
Musuh2 b=(Musuh2) getOneIntersectingObject(Musuh2.class);  
if(b!=null)  
{  
    Greenfoot.playSound("die1.wav");  
    getWorld().removeObject(b);  
    getWorld().removeObject(this);  
    return;  
}
```


testing dan jalankan (untuk tes awal jangan coba dengan Musuh1 karena Musuh1 bergerak ke kanan jadi peluru gak bisa mengejarannya kecuali kecepatan peluru dirubah misalnya jadi +2), jika berhasil kerjakan untuk semua Musuh (1-5) lalu test hasilnya.

Setelah semua bisa, bagaimana jika sang Princess tertembak ?, buat sang Princess mati dan permainan berakhir. Buat coding di Princess dan isi dengan:

```
Peluru1 f=(Peluru1)findOneIntersectingObject(Peluru1.class);
if(f!=null)
{
    Greenfoot.playSound("diel.wav");
    setImage("kalah.png");
    getWorld().removeObject(f);
    Greenfoot.stopSimulation();
}
```

bagaimana untuk mengalahkan Musuh1 ? Buat Peluru2 jika menekan tombol z buat coding tambahan di Prince, letakan dibawah perintah penekanan tombol x

```
else if(Greenfoot.isKeyDown("z"))
{
    pelor.addObject(new Peluru2(), getX(),getY());
    Greenfoot.playSound("fire.wav");
}
```

Buat coding pada Peluru2 salin dari file Peluru1 dan jangan lupa ganti arah geraknya jadi kekiri

```
.....
// Peluru2 bergerak dari kiri ke kanan
setLocation(getX()-1,getY());
if(getX()==0)
{
    getWorld().removeObject(this);
    return;
}
//jika Musuh1 terkena Peluru2
Musuh1 a=(Musuh1)findOneIntersectingObject(Musuh1.class);
if(a!=null)
{
    Greenfoot.playSound("diel.wav");
    getWorld().removeObject(a);
    getWorld().removeObject(this);
    return;
}
.....
```

Membuat Skor

Agar permainan lebih menarik buat skor, dengan ketentuan setiap membunuh 1 musuh maka nilai akan bertambah 20 dengan pengurangan pada peluru 1 peluru -1 nilai

Langkah-langkahnya:

1. Buat Actor Nilai dengan gambar angka 0 (ambil dari folder contoh di balloons tepatnya di greenfoot/scenarios/balloons/images/Counter.png)
2. Masuk ke jendela coding pada Actor Nilai dan buat coding diatas Public void act

```
private int angka = 0;
private int target = 0;
private String huruf;
private int panjangHuruf;

public Nilai(String depan)
{
    huruf=depan;
    panjangHuruf = (huruf.length() + 2) * 16;

    setImage(new GreenfootImage(panjangHuruf, 24));
    GreenfootImage gambar = getImage();
    Font jenisHuruf = gambar.getFont();
    gambar.setFont(jenisHuruf.deriveFont(24.0F));

    gantiGambar();
}
private void gantiGambar()
{
    GreenfootImage gambar = getImage();
    gambar.clear();

    gambar.setColor(Color.RED);

    gambar.drawString(huruf + angka, 1, 18);
}
```

private int angka = 0; ==> variabel angka

Font jenisHuruf = gambar.getFont(); ==> ambil jenis huruf karena perintah ini di atasnya harus ada import java.awt.Font

gambar.setFont(jenisHuruf.deriveFont(24.0F)); ==> gunakan huruf terbesar yaitu 24 dengan jenis font derive. 0F dibaca nol ef

gantiGambar(); ==> jalankan class gantiGambar

private void gantiGambar() ==> membuat class gantiGambar

gambar.clear(); = kosongkan gambar (angka 0 diawal)

gambar.setColor(Color.RED); ==> pakai tulisan warna Merah. Pilihan lainnya BLACK, WHITE, RED, GREEN (penulisan harus huruf besar semua). Karena perintah ini memanggil perintah java di atasnya harus dibuat import java.awt.Color

gambar.fillRect(20, 4, 24, 16); ==> background tulisan disini tidak dipakai

gambar.drawString(huruf + angka, 1, 18); ==> tata letaknya tulisan dulu baru angkanya

3. Tambahkan dibaris ke-dua dari atas dibawah import greenfoot.*;

```
import java.awt.Font;
import java.awt.Color;
```

4. Buka class World(Adventure) lalu buat variabel diatas Public Adventure ()

```
Nilai skor = new Nilai("Skor: ");
```

5. Tambahkan object dibawah perintah menambahkan object yang lain

```
addObject(skor, 35, 280);
```

6. Lihat hasilnya sudah munculkah tulisan Skornya ? jika sudah lanjut agar skor bertambah jika musuh mati

buka kembali layar coding Nilai dan tambahkan coding yang setingkat dengan public/private void.

```
public void tambah(int skor)
{
    target += skor;
}
```

→ nilai target bertambah dan masukan ke skor. Tambahkan pada class public void act

```
if(angka < target)
{
    angka++;
    gantiGambar();
}
```

→ jika angka kurang dari target maka skor selalu bertambah.

Setelah selesai dengan file Nilai, kita beralih ke file classWorld Adventure, buka dan tambahkan coding setingkat public void

```
public void tambahNilai()
{
    skor.tambah(20);
}
```

→ buat variabel public (dapat dibuka pada file lain) dengan penambahan nilai 20. beda Musuh dibuat beda nilaipun bisa caranya buat beberapa Public void spt ini dan nilainya.

Lanjutkan untuk menampilkan perubahan nilai ketika peluru mengenai musuh. Buka coding Peluru1 testing pada Musuh2 tambahkan coding dibawah if.....=Null

```
((Adventure) getWorld()).tambahNilai();
```

→ jika Peluru1 mengenai Musuh2 tambahkan nilai

coding lengkapnya:

```
Musuh2 b=(Musuh2) getOneIntersectingObject(Musuh2.class);
if(b!=null)
{
    ((Adventure) getWorld()).tambahNilai();
    Greenfoot.playSound("diel.wav");
    getWorld().removeObject(b);
    getWorld().removeObject(this);
    return;
}
```

Lakukan hal yang sama untuk Musuh1-Musuh5 lanjutkan pada Peluru2 dengan Musuh1-5.

Testing dan lihat hasilnya.

Lanjutkan untuk pengurangan nilai, jika peluru dikeluarkan kurangi nilai 1, caranya buka file Nilai tambahkan coding dibawah public void tambah.

```
public void kurang(int skor)
{
    target -= skor;
}
```

pada public void act tambahkan coding berikut setelah perintah if selesai

```
else if(angka > target)
{
    angka--;
    gantiGambar();
}
```

lanjutkan dengan menambahkan coding pada World Adventure, tambahkan dibawah coding public void tambahNilai()

```
public void kurangiNilai()
{
    skor.kurang(1);
}
```

Lakukan pemicunya pada actor Prince (pemicu yang mengeluarkan Peluru) syaratnya peluru ditekan nilai langsung berkurang 1, buat coding di bawah playSound

```
((Adventure) getWorld()).kurangiNilai();
```

→ panggil aksi kurangiNilai yg berada di World Adventure.

Lakukan hal yang sama pada penekanan tombol z, coding lengkapnya:

```
if(Greenfoot.isKeyDown("x"))
{
    pelor.addObject(new Peluru1(), getX(),getY());
    Greenfoot.playSound("fire.wav");
    ((Adventure) getWorld()).kurangiNilai();
}
else if(Greenfoot.isKeyDown("z"))
{
    pelor.addObject(new Peluru2(), getX(),getY());
    Greenfoot.playSound("fire.wav");
    ((Adventure) getWorld()).kurangiNilai();
}
```

Pemunculan Papan Skor

Satu urusan lagi, papan skor dengan rancangan jika Prince bertemu Princess papan skor akan muncul dengan ucapan SELAMAT dengan nilai dan permainan berakhir.

Tambahkan Actor PapanSkor lalu buat codingnya, coding lengkapnya:

```
import greenfoot.*;
import java.awt.Color;
import java.awt.Font;
import java.util.Calendar;
/**
 * @author (Muhidin S)
 * @version (1.0.0)
 */
public class PapanSkor extends Actor
{
    public static final float besarHuruf = 48.0f;
    public static final int lebar = 500;
    public static final int tinggi = 300;
    //Tulisan yang muncul dilayar
    public PapanSkor(int skor)
    {
        bikinGambar("SELAMAT", "Skor Akhir: ", skor);
    }
    // Membuat gambar papan skor
    private void bikinGambar(String judul, String depan, int skor)
    {
        GreenfootImage gambar = new GreenfootImage(lebar, tinggi);

        gambar.setColor(new Color(25, 5, 74, 160));
        gambar.fillRect(0, 0, lebar, tinggi);
    }
}
```

```
gambar.setColor(new Color(255, 255, 255, 50));
gambar.fillRect(5, 5, lebar-10, tinggi-10);
Font font = gambar.getFont();
font = font.deriveFont(besarHuruf);
gambar.setFont(font);
gambar.setColor(Color.GREEN);
gambar.drawString(judul, 60, 100);
gambar.drawString(depan + skor, 60, 200);
setImage(gambar);
}
}
```

Di Actor Nilai tambahkan coding yang setingkat dengan public void act()

```
public int nilaiAkhir()
{
    return angka;
}
```

Di World Adventure buat coding setingkat public void seperti ini:

```
public void selesai()
{
    addObject(new PapanSkor(skor.nilaiAkhir()*5), 247,145);
}
```

→ Nilai akhir didapat dari jumlah skor dikalikan 5 dan letakan di tengah-tengah

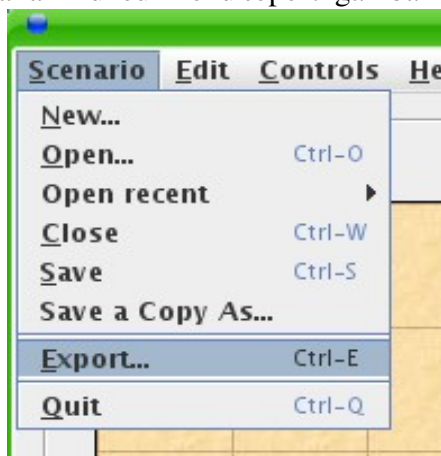
Buat pemicunya di Actor Prince atau Princess (pilih salah satu), disini pakai Princess

```
//Jika Princess terselamatkan oleh Prince
Prince h=(Prince)getOneIntersectingObject(Prince.class);
if(h!=null)
{
    ((Adventure) getWorld()).selesai();
    Greenfoot.playSound("login.wav");
    Greenfoot.stopSimulation();
}
```

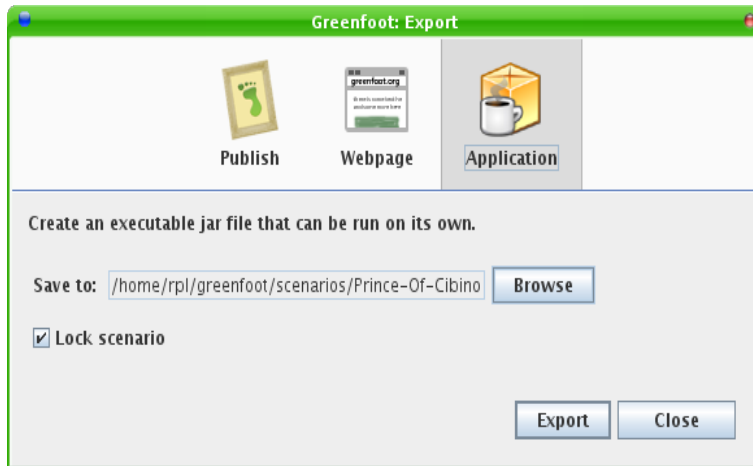
Final

Langkah terakhir adalah jadikan game kita dapat dijalankan di komputer yang tidak ada Greenfootnya (Java tetap wajib ada)

Pada game klik menu Scenario > Export atau tekan tombol Ctrl-E seperti gb 24 maka akan muncul menu seperti gambar 25



Gambar 24 pilih menu Export



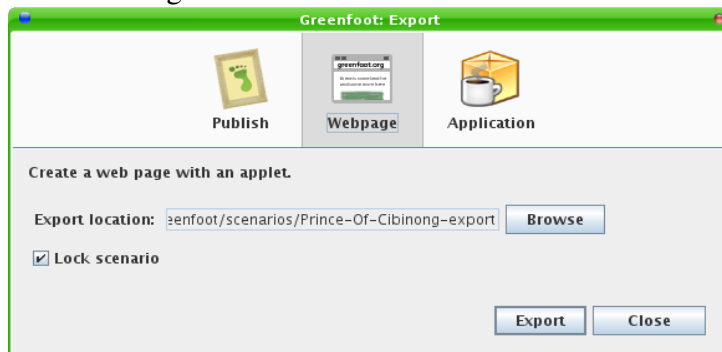
Gambar 25 Menu pilihan Export (Application)

ada tiga pilihan:

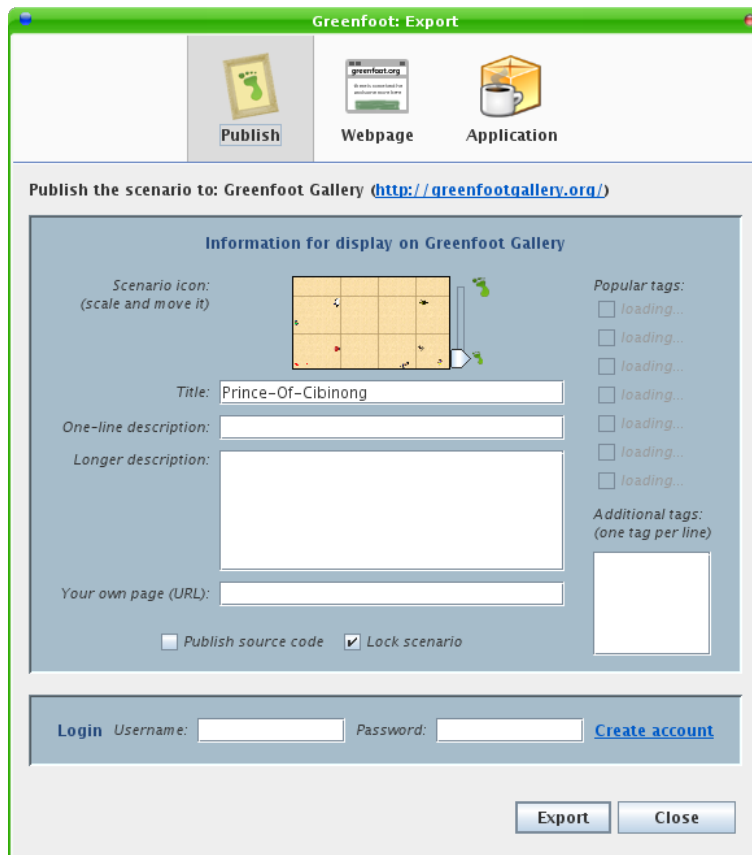
Application : Jadikan sebagai file Aplikasi (menjadi file .jar) gambar.25

Webpage : Jadikan sebagai file .html (buat 1 folder didalamnya langsung terbentuk 2 file, html dan jar). Gambar 26

Publish : Mempublikasikan hasil game buatan kita ke situs <http://greenfootgallery.org> gambar.27



gambar 26 Export ke web



gambar 27 Export source code ke www.greenfootgallery.org

Penutup

Ternyata untuk membuat game berbasis Java menggunakan Greenfoot bukanlah hal yang sulit. Ada kemauan pasti ada jalan.

Referensi

<http://www.greenfoot.org/doc/tutorial/tutorial.html> (Greenfoot Tutorial), Michael Kölling (University of Kent)

Biografi



Muhidin Saimin, sekarang dipercaya sebagai system support di SMA Plus PGRI Cibinong dan Guru Student day Komputer - Java On Linux