

Cara membuat game berbasis Java menggunakan Greenfoot bag. 1

Muhidin

naufal_mr@yahoo.com

<http://muhidins.blogspot.com>

Lisensi Dokumen:

Copyright © 2003-2007 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan 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 IlmuKomputer.Com.

Pendahuluan

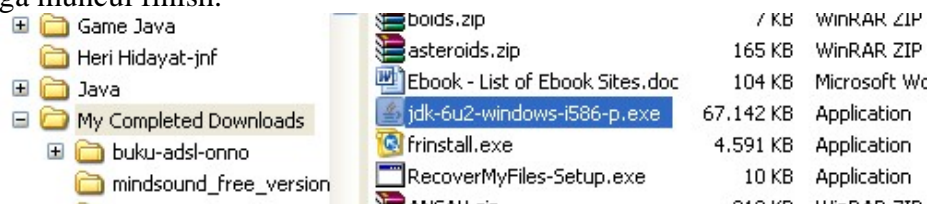
Game berbasis Java sekarang ini sudah banyak sekali beredar di masyarakat luas, baik game yang ada di Handphone maupun menggunakan komputer/laptop. Artikel ini diterbitkan karena penulis merasa begitu kurangnya informasi tentang greenfoot ketika kami ingin mengikuti lomba Grand Java Night Festival 2007 di Kampus Gunadarma Depok (www.gunadarma.ac.id/javanight/)

Syarat Program

Untuk dapat menjalankan program greenfoot dikomputer anda harus sudah ada program Java 5 (JDK 1.5) atau diatasnya disini penulis menggunakan Java 6 (jdk-6u2-windows-i586-p.exe) untuk program java dapat didownload di <http://java.sun.com/javase/downloads/index.jsp>. Sedangkan greenfoot dapat didownload di <http://www.greenfoot.org/download/>.

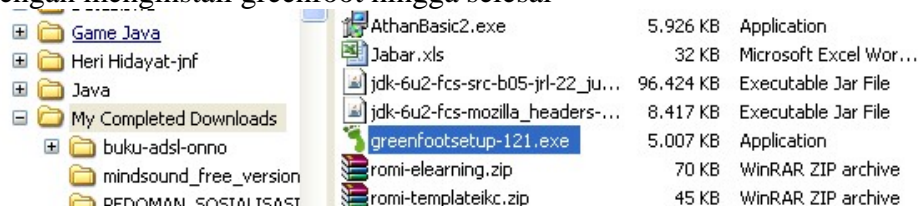
Cara Instalasi

Tahap awal install-lah Java 6 dengan cara double klik pada file hasil download dan seperti kebanyakan program di windows lainnya cara instalasinyaapun sangat mudah, ikuti instruksi dilayar hingga muncul finish.



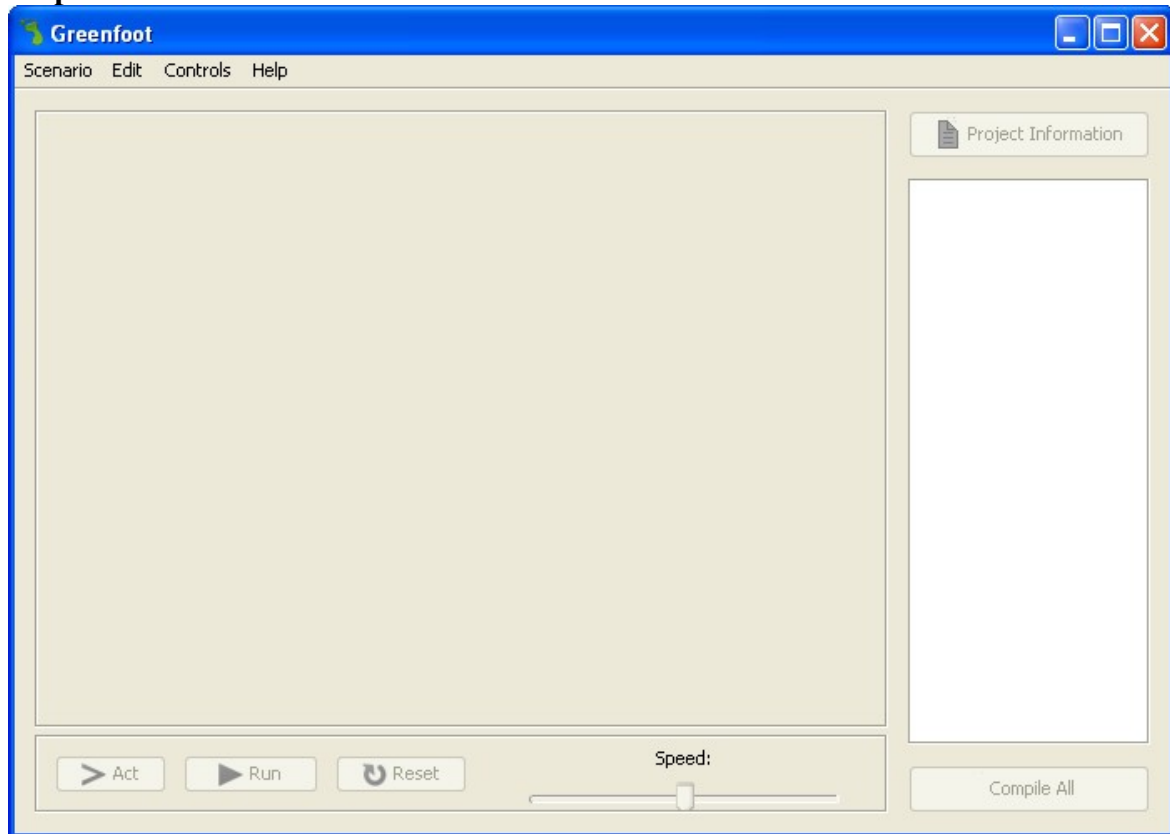
Gambar 1. Double klik file java dan ikuti instruksi layar hingga selesai

Lanjutkan dengan menginstall greenfoot hingga selesai



Gambar 2. Install file Greenfoot hingga finish

Tampilan awal Greenfoot

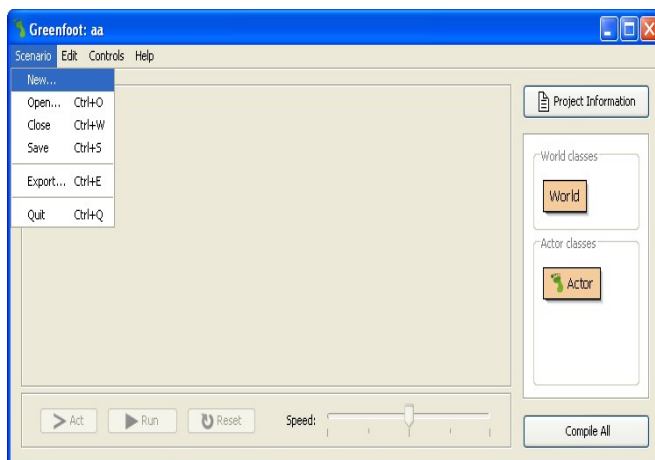


Gambar 3. Tampilan awal greenfoot

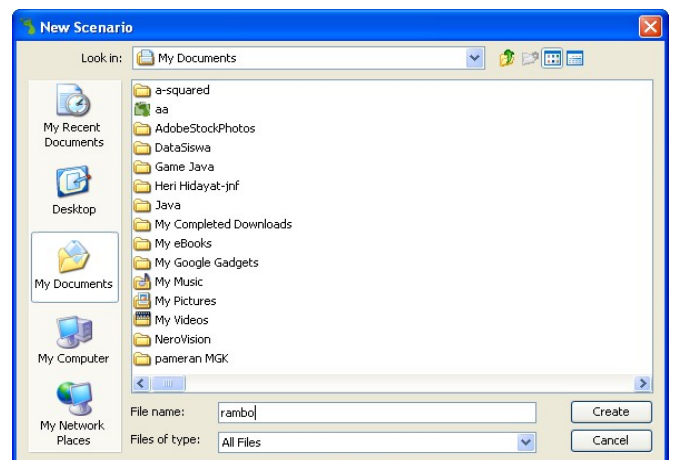
Langkah awal untuk membuat game di greenfoot tentukan skenarionya seperti apa untuk bahasan kali ini nya adalah :

“Objek jagoan akan mengambil diamond yang dijaga ketat oleh ular-ular dan beruang yang bunuh dengan peluru yang berbeda-beda, dan disana ada makanan yang jika diambil akan menambah nilai”

Buat scenarionya dengan menklik menu *scenario* lalu *new* lanjutkan dengan memberi nama scenario. (misalnya *Rambo*) seperti gambar 4. Lalu akan muncul isian seperti gambar 5

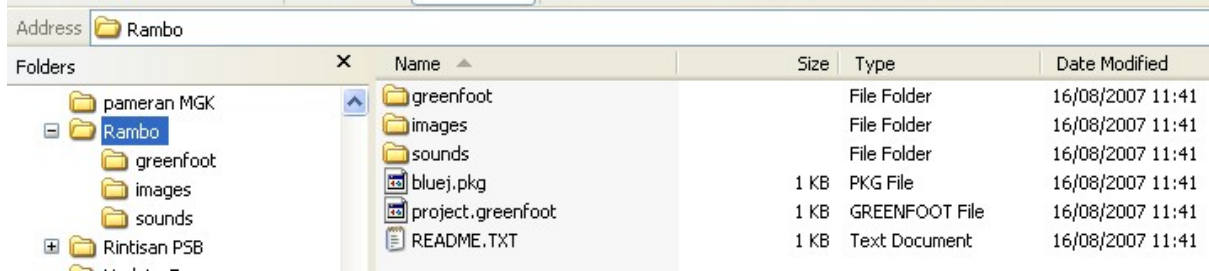


Gambar 4 Membuat scenario baru



Gambar 5. Beri nama scenario

Jika dilihat di windows explorer maka hasil yang kita lakukan akan terlihat seperti gambar 6

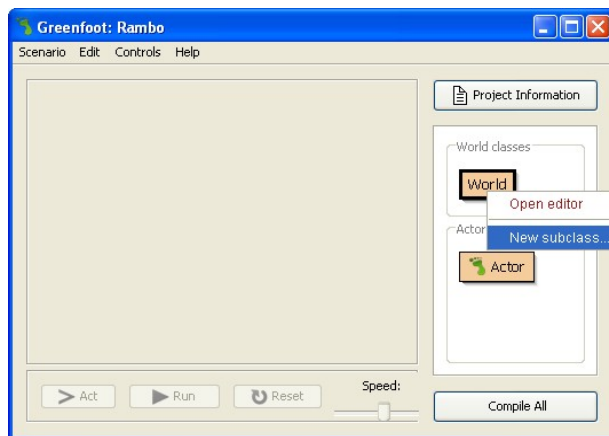


Gambar 6. Hasil pembuatan scenario jika dilihat menggunakan windows explorer

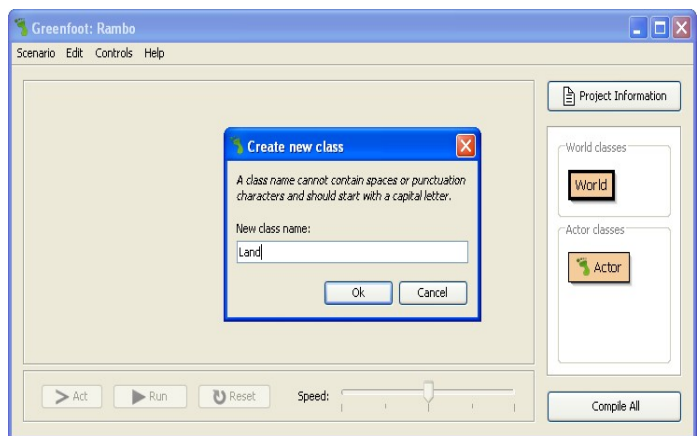
Disana terlihat scenario yang kita buat berbentuk folder dan didalamnya ada 3 folder yang otomatis terbentuk, yaitu : *greenfoot*, *images* dan *sound*. Fungsi folder images adalah untuk menyimpan gambar yang akan kita gunakan, sedangkan folder sound untuk menyimpan file suara yang kita pakai untuk game buatan kita. Folder images dan sound awalnya masih kosong.

Subclass Land (World)

Buatlah sebuah *sub class* di class *world* dengan cara klik kanan lalu pilih *new subclass* seperti gambar 7. Lalu akan muncul jendela baru untuk memasukan nama kelas spt gb. 8 misalnya beri nama *Land* (perhatikan huruf besar dan kecilnya, karena pada pemrograman java bersifat case sensitive artinya huruf besar dan kecil berpengaruh)

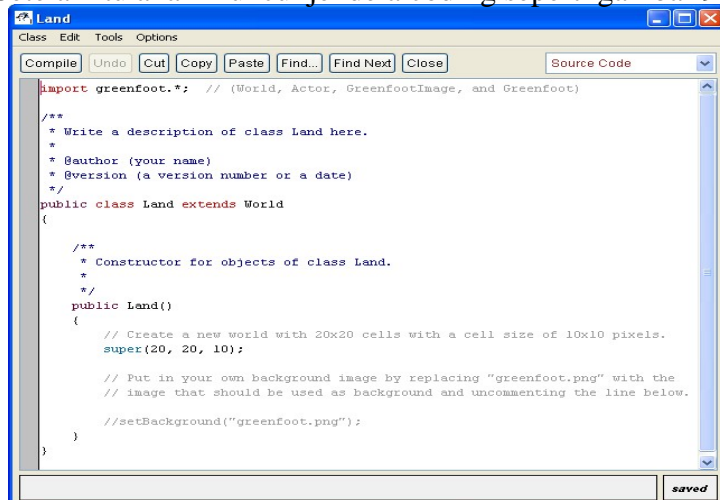


Gambar 7. Cara membuat subclass



Gambar 8. beri nama subclass

Setelah itu akan muncul jendela coding seperti gambar 9.



Gambar 9. Jendela coding

Perhatikan dibawah *Land* (subclass Land) ada perintah

super(20, 20, 10);

perintah ini untuk membuat ukuran layar lebar 20 x tinggi 20 dengan ukuran cell 10x10

super(wide, high, cell);

untuk keperluan kali ini buat ukuran (80,20,20)

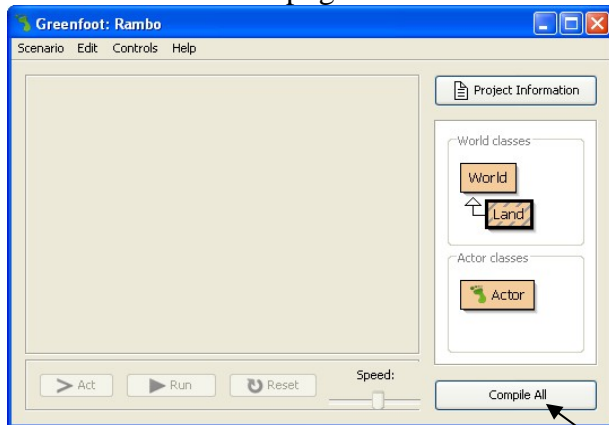
lalu masukan gambar backgroundnya di folder images (misalnya nama file *Walpaper.jpg*)

lalu set dengan perintah :

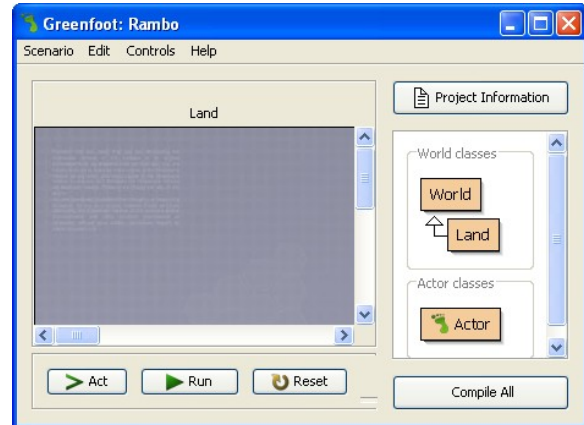
setBackground("Wallpaper.jpg");

test program yang dibuat dengan mengklik tombol *compile all* di pojok kanan bawah spt gbr.

10 dan akan muncul spt gbr. 11

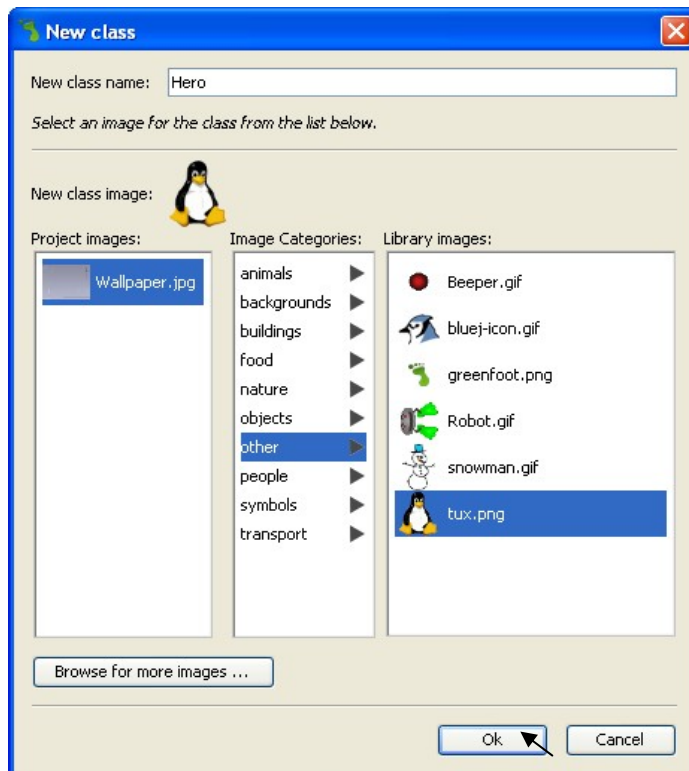


Gambar 10. Test program



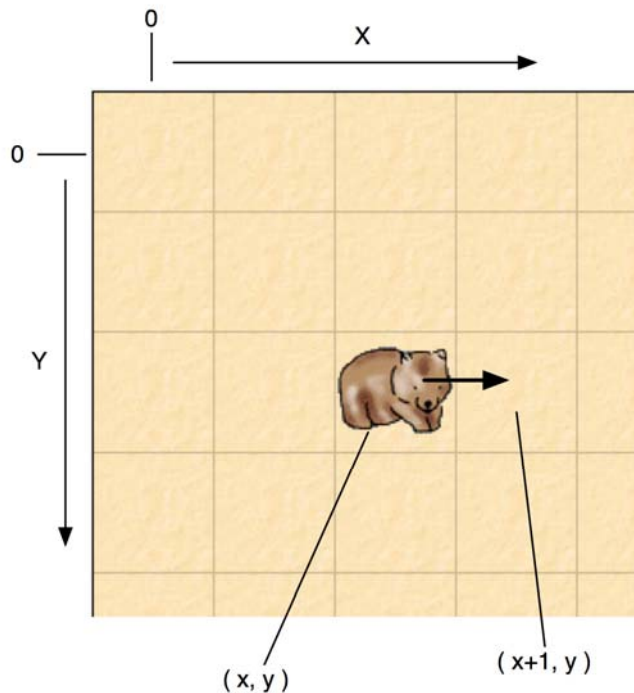
gambar 11. Hasil test

Subclass Hero (Actor)



Gbr. 12. Pilih gambar dan beri nama subclasss

Lanjutkan dengan membuat actor yang bermain dalam scenario tersebut. Buat actor baru (subclass actor) dengan cara klik kanan di actor (spt gambar 7 namun di tombol Actor) lalu pilih new subclass kemudian pilih gambarnya dan beri nama subclass tersebut misalnya Hero seperti gambar 12 lalu klik OK



Gbr. 13 Pemahaman sumbu X dan Y

Untuk melakukan pergerakan perhatikan gambar 13 (**Ubah Lokasi**)

Instruksi dasar pada perintah ini adalah

```
public void act()
{
    int x = getX();
    int y = getY();
    setLocation(x + 1, y);
}
```

Dapat diartikan jika *Act* ditekan maka objek akan bergerak secara horizontal (sumbu x) kearah kanan (positif) dan perintah tersebut dapat ditulis dengan lebih ringkas menjadi:

```
public void act()
{
    setLocation(getX() + 1, getY());
}
```

Buka jendela coding untuk subclass Hero dan tambahkan kode sbb:

```
//CONTROL
if (Greenfoot.isKeyDown("right")) {
    setLocation(getX() + 2, getY());
}
if (Greenfoot.isKeyDown("left")) {
    setLocation(getX() - 2, getY());
}
if (Greenfoot.isKeyDown("up")) {
    setLocation(getX(), getY() - 2);
}
if (Greenfoot.isKeyDown("down")) {
    setLocation(getX(), getY() + 2);
}
```

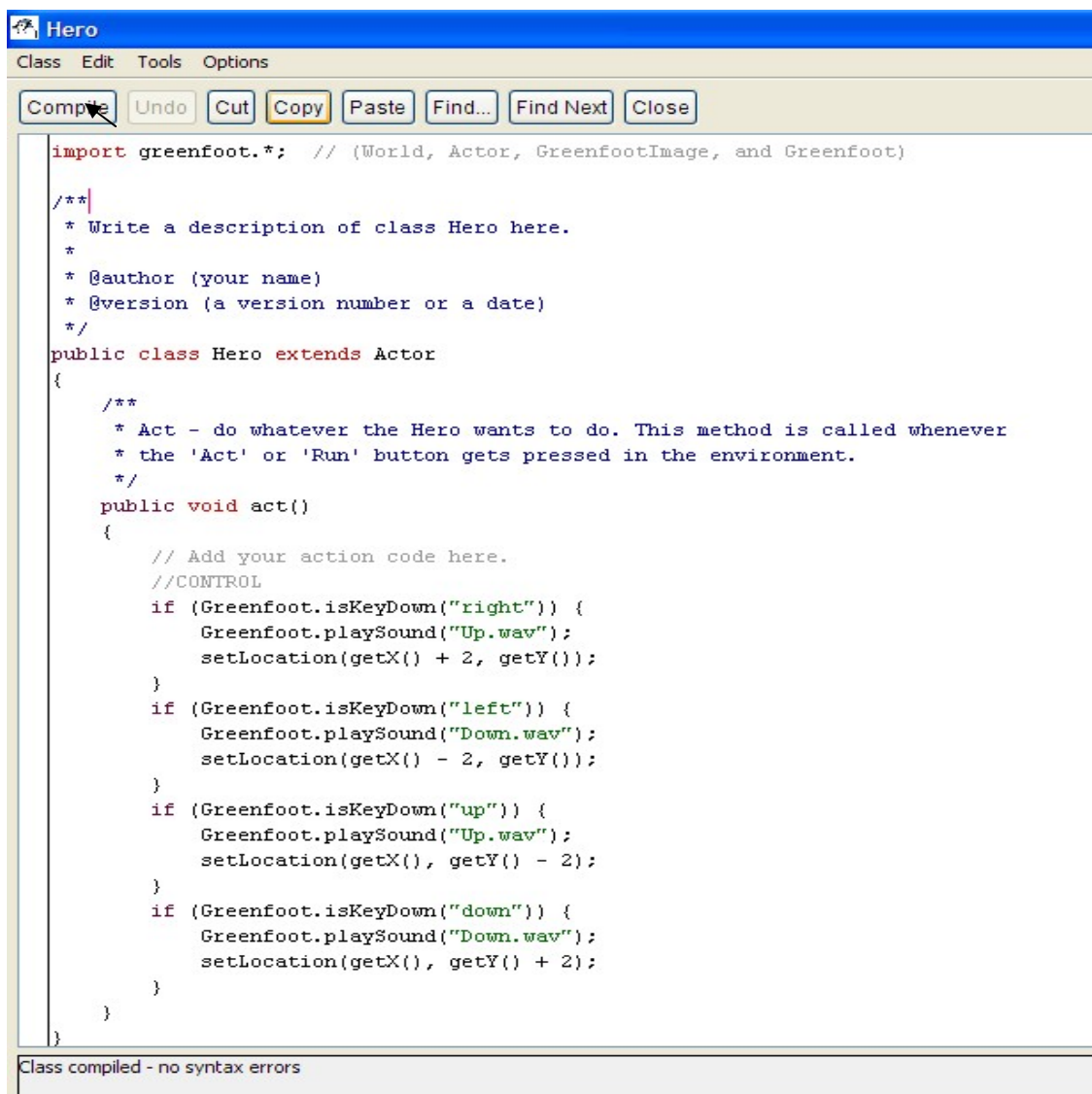
Dasar itulah maka perintah diatas dibuat, yang dapat diartikan sebagai berikut:

Jika panah kanan ditekan maka *Hero* akan berpindah lokasi kekanan 2 pixel (x+2), jika panah kir ditekan maka *Hero* akan berpindah lokasi kekiri 2 pixel (x-2), jika panah atas ditekan maka *Hero* akan berpindah lokasi keatas 2 pixel (y+2) dan jika panah bawah ditekan maka *Hero* akan berpindah lokasi kebawah 2 pixel (y-2).

Tekan *Compile* seperti yang terlihat gambar 15 dan pastikan coding yang anda buat sudah sesuai, jika tak ada yang salah akan muncul tulisan *Class compiled – no syntax error*. Lalu kembali ke jendela *Greenfoot: Rambo* dan test dengan cara tekan *compile all* lalu klik kanan pada subclass *Hero* (spt gb. 14) lalu pilih *run* dan perhatikan actor Hero pergerakannya sudah sesuai dengan harapan kita atau belum ?



Gambar 14. pilihan untuk menampilkan subclass

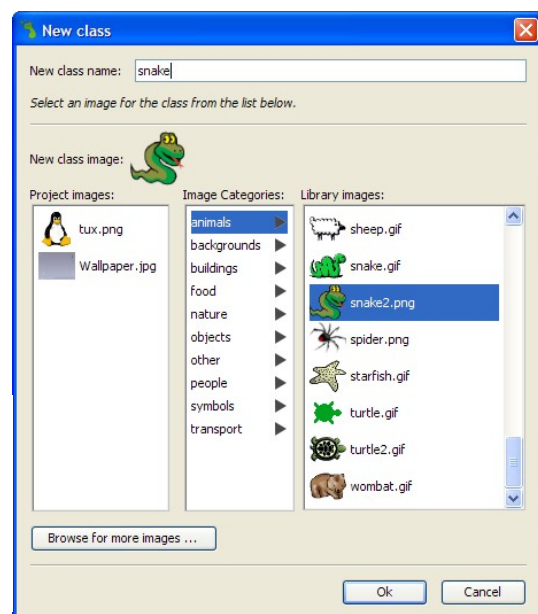


Gambar 15. Jendela coding subclass *Hero*

Lanjutkan dengan membuat actor (subclass) lawannya misal namanya *Snake* dengan cara klik kanan pada actor dan pilih New subclass spt gbr 16 lanjutkan dengan pemberian nama subclass (snake) spt gambar 17. lalu tekan OK

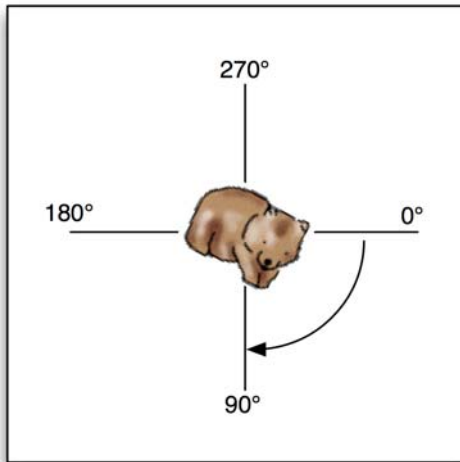


Gambar 16. Buat actor baru



Gambar 17. Pilih gambar yang dipakai dan beri nama subclass

Merubah Rotasi



Gambar 18. Rotasi

Untuk merubah rotasi perintah dasarnya adalah:

```
public void act()  
{  
    int rot = getRotation() + 1;  
    if(rot == 360) {  
        rot = 0;  
    }  
    setRotation(rot);  
}
```

Metode ini berarti mengambil object (gambar) pada rotasi sekarang dan akan selalu bertambah 1. Efeknya objek akan bergerak perlahan kearah panah.

Angka yang benar pada rotasi adalah angka 0-359. Untuk mensiasati ini dibuat pernyataan `if(rot == 360) {` yang dapat diartikan jika sudah mencapai rotasi 360⁰ maka `rot=0`.

Buka jendela coding (double klik pada actor snake) lalu buat pergerakan snake dengan aturan snake akan selalu bergerak sambil berotasi (berputar-putar).

```
import greenfoot.*; // (World, Actor, GreenfootImage, and Greenfoot)  
  
/**  
 * Write a description of class snake here.  
 *  
 * @author (your name)  
 * @version (a version number or a date)  
 */  
public class snake extends Actor  
{  
    /**  
     * Act - do whatever the snake wants to do. This method is called whenever  
     * the 'Act' or 'Run' button gets pressed in the environment.  
     */  
    public void act()  
    {  
        /**  
         * Add your action code here.  
         * //pergerakan snake  
         int rot = getRotation() + 10;  
         if (rot < 90) {  
             setLocation(getX() + 2, getY());  
         }  
         else if (rot < 180) {  
             setLocation(getX(), getY() + 2 );  
         }  
         else if (rot < 270) {  
             setLocation(getX() - 2, getY());  
         }  
         else if (rot < 360) {  
             setLocation(getX(), getY() - 2);  
         }  
         if (rot > 360) {  
             rot = 0;  
         }  
         setRotation(rot);  
    }  
}
```

Subclass Beruang (actor)

Lanjutkan dengan membuat subclass *beruang* dengan cara seperti diatas, dengan aturan sama namun subclass *beruang* jangkauannya lebih kecil (Bukan + 2 tapi + 1 dan rotasi + 15) lihat codingnya dibawah.

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

/**
 * Write a description of class beruang here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class beruang extends Actor
{
    /**
     * Act - do whatever the beruang wants to do. This method is
     called whenever
     the 'Act' or 'Run' button gets pressed in the environment.
     */
    public void act()
    {
        // Add your action code here.
        int rot = getRotation() + 15;
        if (rot < 90) {
            setLocation(getX() + 1, getY());
        }
        else if (rot < 180) {
            setLocation(getX(), getY() + 1 );
        }
        else if (rot < 270) {
            setLocation(getX() - 1, getY());
        }
        else if (rot < 360) {
            setLocation(getX(), getY() - 1);
        }
        if (rot > 360) {
            rot = 0;
        }
        setRotation(rot);
    }
}
```

Subclass Peluru (actor)

Yang selanjutnya harus dipikirkan adalah untuk melumpuhkan lawan-lawan yang telah dibuat tadi dengan senjata apa ? misalkan si *Hero* akan melumpuhkan *snake* dengan menggunakan *Peluru* sedangkan untuk membunuh *Beruang* harus menggunakan 2 *peluru* atau 1 *bom*. Tombol *F1* untuk *Peluru* dan *F2* untuk *Bom*. Baiklah kita lanjutkan dengan membuat subclass *Peluru*. Masukan file suara (suara tembakan ke folder sounds – disini penulis pakai file *EnergyGun.wav*) dengan cara spt diatas dan pilih gambar peluru (bisa ambil di internet atau buat sendiri dengan pengolah gambar – penulis pakai file *bullet.png* diambil dari file *asteroid* (greenfoot) hasil download).

Baiklah tutorial kali ini kita lanjutkan ke bagian ke-2. Bersambung

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 sekarang diamanahi sebagai System Support di Departemen Teknologi Informasi - SMA Plus PGRI Cibinong yaitu departemen yang mengemban ABUDATA (Amanah Bangun Umat cerDAs berTAqwa). Juga diberi amanah sebagai Pembina KOPASUS-IT (KelOmpok PembinaAn khuSUS-Information Teknologi) divisi RPL.