

PELATIHAN PEMBUATAN GAME MENGGUNAKAN GREENFOOT

¹⁾ Joseph Dedy Irawan ²⁾ Sonny Prasetyo ³⁾ Suryo Adi Wibowo ⁴⁾ Yosep Agus Pranoto

^{1,2,3,4)} Prodi Teknik Informatika, Fakultas Teknologi Industri, Institut Teknologi Nasional Malang

ABSTRAK

Banyak sekali murid dan guru di SMK khususnya dalam bidang teknologi informasi dan komunikasi mengeluhkan sulitnya mempelajari pemrograman komputer, karena banyak sekali perintah-perintah yang harus dipelajari, sehingga mereka merasa sulit dan menjadi semakin malas mempelajarinya.

Teknologi pemrograman sangat berkembang dengan pesat, salah satu diantaranya adalah Greefoot, dimana pemrograman ini dapat digunakan untuk pembuatan game dengan cara yang sangat mudah karena hanya memerlukan sedikit perintah sehingga akan sangat memudahkan murid dan guru mempelajarinya.

Dengan mengadakan pelatihan pembuatan game menggunakan greenfoot, diharapkan para murid dan guru dapat mempelajari pemrograman dengan mudah serta minat untuk mempelajari pemrograman menjadi tinggi sehingga proses pembelajaran dapat berjalan dengan lebih mudah dan lancar.

Kata kunci : Kelompok Pembuatan Game, Greenfoot, Pemrograman Komputer

Pemrograman adalah salah satu pelajaran yang dianggap sulit oleh para murid di SMK, sehingga banyak siswa yang mengalami kesulitan dan malas untuk mempelajarinya, hal ini disebabkan pada pemrograman dibutuhkan banyak sekali perintah-perintah yang harus dituliskan, sehingga membuat para murid menjadi kesulitan.

Dengan menggunakan greenfoot, pembuatan program dalam hal ini pembuatan game dapat dilakukan dengan mudah, karena pada pemrograman ini hanya dibutuhkan sedikit sekali perintah-perintah yang ditambahkan untuk membuat sebuah game, sehingga para murid akan dengan lebih mudah dapat memahami pemrograman dengan lebih baik.

Pengertian Greenfoot

Greenfoot adalah perangkat lunak yang didesain untuk pemula agar dapat terbiasa dengan Pemrograman Berorientasi Objek (Object-Oriented Programming), yang mendukung pengembangan aplikasi bergambar dengan memakai bahasa pemrograman Java™.

Greenfoot didesain dan telah diimplentasikan di Universitas Kent, Inggris dan Universitas Deakin, Melbourne – Australia

Keuntungan Greenfoot

- Merupakan software yang mudah untuk semua kalangan
- Software open source
- Menghemat waktu proses belajar mengajar
- Membuat siswa menyukai pemrograman

Kerugian Greenfoot

- Game yang dihasilkan 2 dimensi tidak bisa 3 dimensi
- Perlu waktu mempelajari khususnya untuk siswa yang belum pernah belajar Java

Perkembangan Greenfoot

Greenfoot adalah pengembangan lingkungan Java interaktif yang dirancang terutama untuk tujuan pendidikan di sekolah tinggi dan tingkat sarjana. Hal ini memungkinkan mudah pengembangan aplikasi grafis dua dimensi, seperti simulasi dan permainan interaktif.

Greenfoot sedang dikembangkan dan dipelihara di University of Kent dan La Trobe University, dengan dukungan dari Oracle. Ini adalah perangkat lunak bebas, dirilis di bawah lisensi GPL. Greenfoot yang tersedia untuk Microsoft Windows, Mac OS X, Linux, Sun Solaris, dan setiap baru JVM.

Proyek Greenfoot diprakarsai oleh Michael Kolling pada tahun 2003, dan prototipe pertama dibangun oleh Poul Henriksen (Masters mahasiswa) dan Michael Kolling (supervisor) tahun 2003/2004. Dari pengembangan 2005 terus melibatkan anggota lain dari BlueJ Group di University of Kent dan Deakin University.

Rilis penuh pertama, Greenfoot versi 1.0, diterbitkan pada tanggal 31 Mei 2006, dengan rilis lebih lanjut berikut sesekali setelahnya.

Pada bulan Mei 2007, proyek Greenfoot dianugerahi "Choice Award Duke" dalam kategori "Teknologi Java dalam Pendidikan", dan pada tahun 2010 memenangkan "Premier

Award untuk Keunggulan dalam Pendidikan Teknik Courseware".

Pada bulan Maret 2009, proyek Greenfoot menjadi Bebas dan Open Source Software, dan berlisensi di bawah GNU GPL dengan pengecualian Classpath

Strategi Pembelajaran Greenfoot

Pembelajaran pembuatan game menggunakan Greenfoot adalah dengan memulai pembuatan game-game sederhana, sehingga para siswa dan guru dapat merasakan betapa mudahnya membuat game tanpa harus bersusah payah menuliskan program. Setelah para peserta sudah menguasai tingkat dasar, maka materi akan terus ditingkatkan tingkat kesulitannya sehingga tanpa terasa peserta pelatihan akan menjadi mahir menggunakan Greenfoot.

Tujuan dari kegiatan ini adalah supaya siswa dan guru lebih mencintai bahasa pemrograman, dan dapat merasakan bahwa membuat program untuk game tidak harus sulit tetapi bisa juga dilaksanakan dengan mudah dan cepat.

METODE

Kerangka Pemecahan Masalah

Permasalahan guru dan siswa SMK yang kesulitan dalam mempelajari pemrograman, meskipun pemrograman merupakan pelajaran yang sangat penting yang harus diketahui khususnya pada jurusan TKJ dan sejenisnya sehingga akan diberikan pelatihan pemrograman dengan menggunakan software open source GreenFoot, karena program ini sangat mudah dioperasikan dan dipelajari

Realisasi Pemecahan Masalah

Pelatihan diberikan dalam bentuk teori dan praktek membuat game, disertai tugas-tugas yang harus dikerjakan pada saat pelatihan agar para peserta lebih memahami cara pembuatan program menggunakan Greenfoot.

Khalayak Sasaran

1. Siswa SMK
2. Guru-guru yang mengajar pemrograman di SMK

Metode yang Digunakan

1. Pelatihan dilaksanakan secara kontinyu
2. Pelatihan dilakukan dengan memberikan buku panduan dimana diberikan teori langsung dilanjutkan praktek

3. Pemberian tugas-tugas untuk melatih peserta terbiasa dengan penggunaan Greenfoot

HASIL DAN PEMBAHASAN

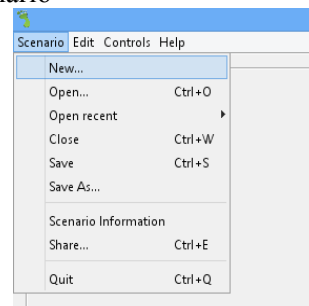
Melalui pelatihan ini, maka hasil yang dapat diperoleh adalah siswa dan guru dapat membuat game menggunakan Greenfoot, dimana game dapat dibuat dengan mudah dan cepat sehingga siswa dan guru akan menyukai pemrograman dan tidak menghindarinya lagi.

Keterbatasan dari pembuatan game menggunakan greenfoot adalah game yang dihasilkan merupakan game 2 dimensi, sehingga tampilan kurang menarik, akan tetapi dengan kreatifitas game 2 dimensi ini dapat dibuat semenarik mungkin dengan menambahkan karakter-karakter yang menarik.

Materi dari pelatihan pembuatan game menggunakan greenfoot dapat dilihat berikut ini:

Membuat scenario baru

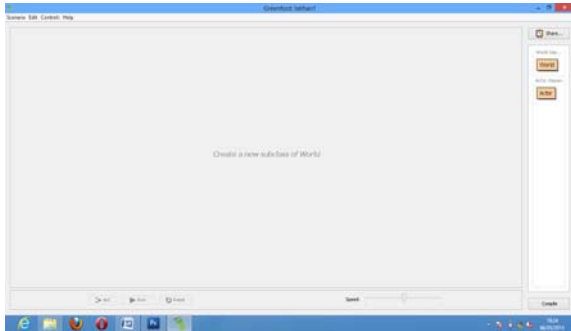
Membuat scenario baru pada greenfoot dapat ditunjukkan dalam gambar 3. pada greenfoot scenario digunakan untuk membuat alur cerita project baru yang akan di buat. Buat scenarionya dengan mengklik menu scenario lalu new lanjutkan dengan memberi nama scenario



Gambar 1. Membuat scenario baru

Dalam folder yang dibuat 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.

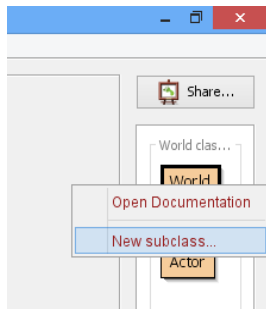
Tampilan greenfoot pada halaman awal seperti ditunjukkan dalam gambar 2 dalam halaman tersebut memiliki menu button world dan actor



Gambar 2. Dalam halaman awal memiliki menu button world dan actor

Subclass Land (World)

Buatlah sebuah sub class di class world dengan cara klik kanan lalu pilih new subclass seperti gambar 3. Lalu akan muncul jendela baru untuk memasukkan nama kelas seperti gambar 4 misalnya beri nama Latar (perhatikan huruf besar dan kecilnya, karena pada pemrograman java bersifat case sensitive artinya huruf besar dan kecil berpengaruh).

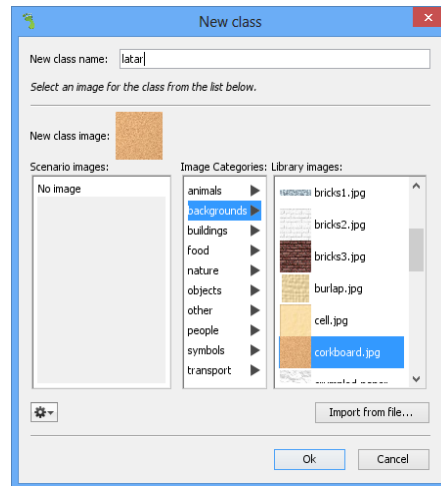


Gambar 3. Membuat world

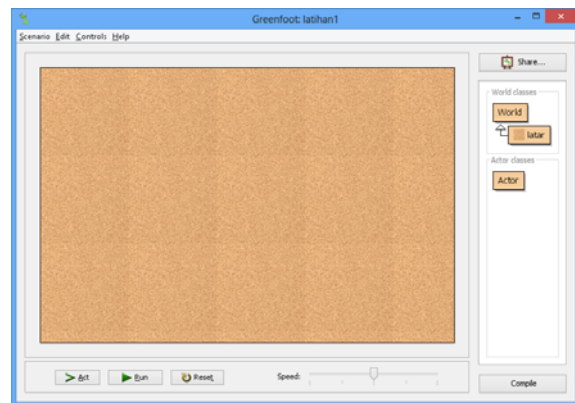
Pada proses simpan halaman new class anda dapat memilih berbagai macam pilihan latar belakang yang digunakan dalam game yang akan dibuat seperti dalam gambar 4 setelah selesai anda dapat meng klik OK.

Ketika proses pemilihan world telah anda lakukan maka akan muncul tampilan jendela seperti dalam gambar 5. untuk proses berikutnya anda diharuskan untuk melakukan Klik compile button yang ada di halaman kanan bawah.

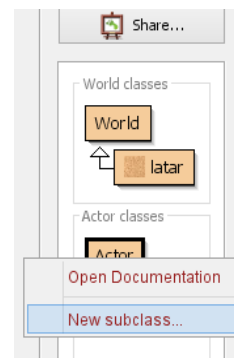
Lanjutkan dengan membuat actor yang bermain dalam scenario tersebut. Buat actor baru (subclass actor) dengan cara klik kanan di actor (seperti gambar 6 namun di tombol Actor) lalu pilih new subclass kemudian pilih gambarnya dan beri nama subclass tersebut misalnya Lebah seperti gambar 7 lalu klik OK.



Gambar 4. proses simpan dan editing world

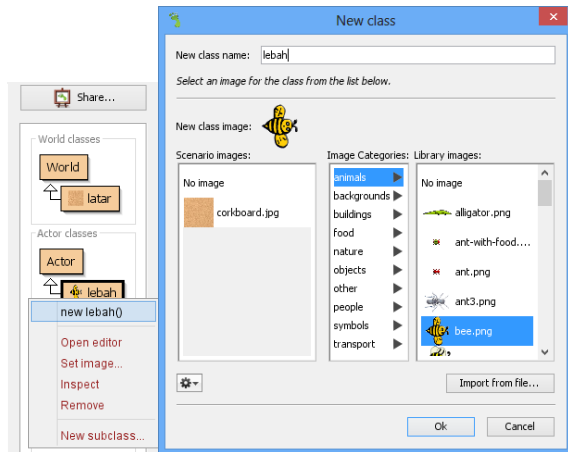


Gambar 5. Hasil dari world



Gambar 6. Membuat actor yang bermain dalam scenario

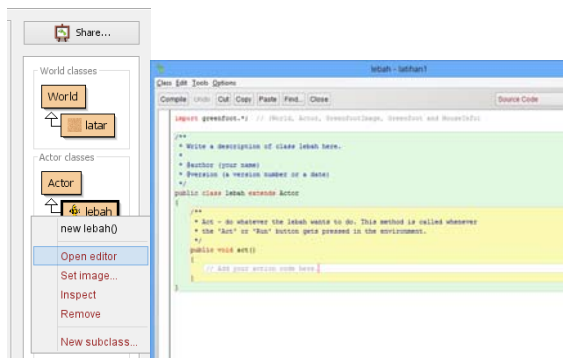
Tambahkan class actor sesuai dengan nama yang diinginkan (lebah) serta pilih actor yang sesuai seperti dalam gambar 7 lalu klik OK.



Gambar 7. pilih actor yang sesuai

Memulai Perintah Dasar

Cara memberikan perintah adalah dengan menggunakan menu editor dengan cara membuka menu editor seperti dalam gambar 8 dan untuk menampilkan tampilan editor dapat dilakukan dengan perintah klik kanan dan pilih open editor sehingga akan tampil jendela tampilan editor yang digunakan untuk memberi input berupa source code pada objek yang di tuju.



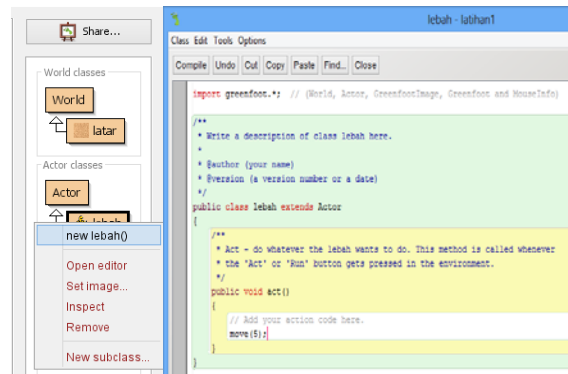
Gambar 8. Tampilan editor

Perintah move

Perintah move digunakan untuk mengkonfigurasi object actor agar dapat bergerak ke kiri dan ke kanan pada land/word/latar

| | |
|------|---|
| void | move (int distance) Move this actor the specified distance in the direction it is currently facing. |
|------|---|

Perintah **move** dapat di masukkan dalam menu editor pada aktor yang di tunjukkan dalam gambar 9 selanjutnya klik compile



Gambar 9. menu editor pada actor

Sehingga akan tampil tampilan jendela seperti di gambar 10 berikut.



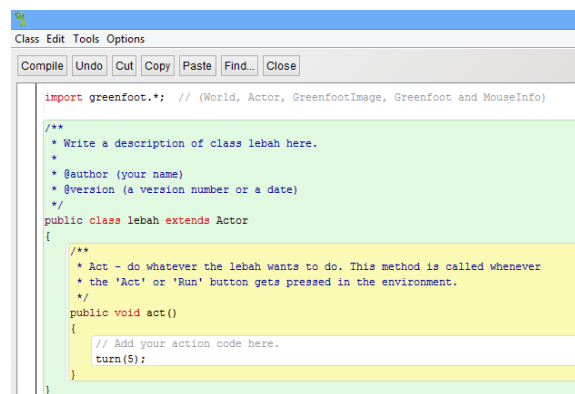
Gambar 10. gambar pergerakan actor

Perintah turn

Perintah **turn** digunakan untuk mengkonfigurasi object actor agar dapat berputar berdasarkan kecepatan

| | |
|------|--|
| void | turn (int amount) Turn this actor by the specified amount (in degrees) |
|------|--|

Perintah **turn** dapat di masukkan dalam menu editor pada aktor yang di tunjukkan dalam gambar 11 selanjutnya klik compile



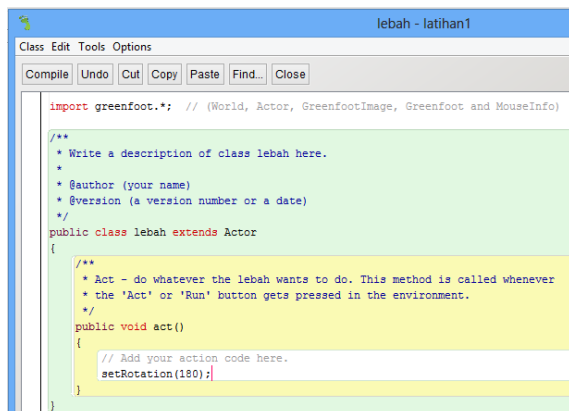
Gambar 11. perintah turn pada actor

Perintah setRotation

Perintah **setRotation** digunakan untuk mengkonfigurasi object actor untuk mengeset sudut putaran yang dilakukan actor

| | |
|------|---|
| void | setRotation (int rotation) Set the rotation of this actor |
|------|---|

Perintah **setRotation** dapat di masukkan dalam menu editor pada aktor yang di tunjukkan dalam gambar 12 selanjutnya klik compile



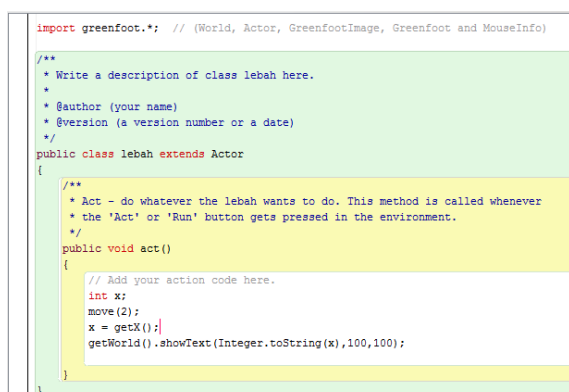
Gambar 12. perintah setRotation pada actor

Perintah getX()

Perintah **getX()** digunakan untuk mengkonfigurasi object actor agar mendapatkan posisi aktor pada kordinat x(...)

| | |
|-----|--|
| int | getX () Return the x-coordinate of the object's current location |
|-----|--|

Perintah **getX()** dapat di masukkan dalam menu editor pada aktor yang di tunjukkan dalam gambar 13 selanjutnya klik compile



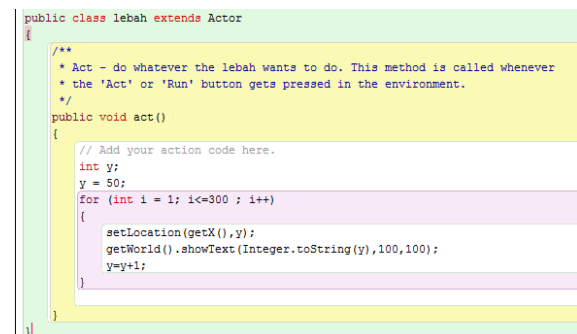
Gambar 13. Perintah getX()

Perintah getY()

Perintah **getY()** digunakan untuk mengkonfigurasi object actor agar mendapatkan posisi aktor pada kordinat Y(...)

| | |
|-----|--|
| int | getY () Return the y-coordinate of the object's current location |
|-----|--|

Perintah **getY()** dapat di masukkan dalam menu editor pada aktor yang di tunjukkan dalam gambar 14 selanjutnya klik compile



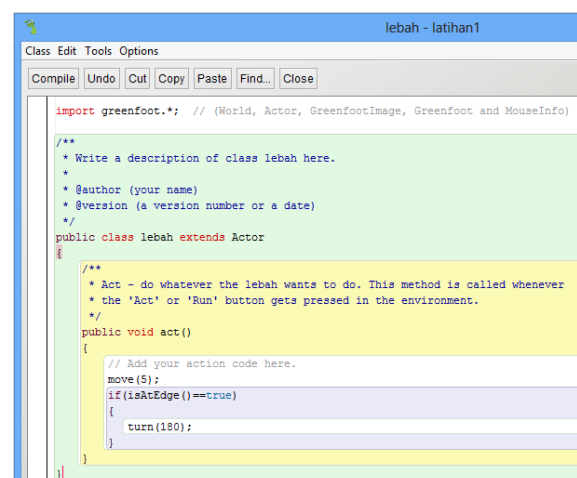
Gambar 14. Perintah getY()

Perintah isAtEdge() – batas tepi world

Perintah **isAtEdge()** digunakan untuk mengkonfigurasi object actor agar mendeteksi tepi world/land/latar.

| | |
|---------|--|
| boolean | isAtEdge() Detect whether the actor has reached the edge of the world. |
|---------|--|

Perintah **isAtEdge()** dapat di masukkan dalam menu editor pada aktor yang di tunjukkan dalam gambar 15 selanjutnya klik compile



Gambar 15. Perintah isAtEdge()

Perintah **isKeyDown()**

Perintah **isKeyDown()** digunakan untuk mendeteksi tekanan tombol keyboard. Biasanya digunakan untuk melakukan control terhadap game atau animasi

| | |
|----------------|--|
| Static boolean | isKeyDown (java.lang.String keyName) Check whether a given key is currently pressed down |
|----------------|--|

Perintah **setLocation()**

Perintah **setLocation** digunakan untuk menentukan atau meletakkan actor pada posisi tertentu berdasarkan koordinat sumbu x dan sumbu y.

| | |
|------|---|
| void | setLocation (int x, int y) Assign a new location for this actor |
|------|---|

Contoh perintah Perintah **isKeyDown()** dan **setLocation** dapat di masukkan dalam menu editor pada aktor yang di tunjukkan dalam gambar 16 selanjutnya klik compile

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

/**
 * Write a description of class lebah here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class lebah extends Actor
{
    /**
     * Act - do whatever the lebah 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.
        if (Greenfoot.isKeyDown("A")==true)
        {
            setLocation(getX()-5, getY());
        }
    }
}
```

Gambar 16. Perintah **isKeyDown()** dan **setLocation**

Perintah **getOneIntersectingObject()**

Perintah **getOneIntersectingObject()** digunakan untuk mendeteksi singgungan (tabrakan) sebuah objek dengan objek yang lain.

| | |
|-----------------|--|
| Protected Actor | getOneIntersectingObject (java.lang.Class cls) Return an object that intersects this object. |
|-----------------|--|

Contoh perintah Perintah **getOneIntersectingObject()** dapat di masukkan dalam menu editor pada aktor yang di tunjukkan dalam gambar 17 selanjutnya klik compile

```
public void act()
{
    // Add your action code here.
    if (Greenfoot.isKeyDown("W")==true)
    {
        setLocation(getX(), getY()-5);
    }

    if (Greenfoot.isKeyDown("Z")==true)
    {
        setLocation(getX(), getY()+5);
    }

    Actor tabrakan = getOneIntersectingObject(bom.class);
    if (tabrakan!=null)
    {
        Greenfoot.stop();
    }
}
```

Gambar 17. Perintah **getOneIntersectingObject()**

Perintah **addObject()**

Perintah **addObject()** digunakan untuk menambahkan objek baru pada world. Perintah **addObject** ini biasanya digunakan pada saat runtime.

| | |
|------|--|
| void | addObject(Actor object, int x, int y) Add an Object to the world |
|------|--|

Contoh perintah Perintah **addObject()** dapat di masukkan dalam menu editor pada aktor yang di tunjukkan dalam gambar 18 selanjutnya klik compile

```
if (Greenfoot.isKeyDown("A")==true)
{
    getWorld().addObject(new apple_merah(), Greenfoot.getRandomNumber(550), Greenfoot.getRandomNumber(200));
}
```

Gambar 18. Perintah **addObject()**

Perintah **removeObject()**

Perintah **removeObject()** digunakan untuk menghapus objek dari world. Perintah **removeObject** ini biasanya digunakan pada saat runtime.

| | |
|------|--|
| void | removeObject(Actor object) Remove an Object from the world |
|------|--|

Contoh perintah Perintah **removeObject()** dapat di masukkan dalam menu editor pada aktor yang di tunjukkan dalam gambar 19 selanjutnya klik compile

```
Actor makan = getOneIntersectingObject(apple_merah.class);
if (makan!=null)
{
    getWorld().removeObject(makan);
}
```

Gambar 19. Perintah **removeObject()**

Perintah playSound()

Perintah **playSound()** digunakan memutar file suara pada Greenfoot

| | |
|-------------|---|
| Static void | playSound(java.lang.String soundfile) Play sound from a file. |
|-------------|---|

Contoh perintah Perintah **playSound()** dapat di masukkan dalam menu editor pada aktor yang di tunjukkan dalam gambar 20 selanjutnya klik compile

```
Actor tabrakan = getOneIntersectingObject(bom.class);
if(tabrakan!=null)
{
    Greenfoot.stop();
    Greenfoot.playSound("sad-trombone.wav");
}

Actor makan = getOneIntersectingObject(apple_merah.class);
if(makan!=null)
{
    getWorld().removeObject(makan);
    Greenfoot.playSound("hooray.wav");
}
```

Gambar 20. Perintah playSound()

Perintah showText()

Perintah **showText()** Menampilkan beberapa teks dengan posisi yang telah ditentukan pada world

| | |
|------|--|
| void | showText(java.lang.String text, int x, int y) Show some text centred at the given position in the world. |
|------|--|

Contoh perintah Perintah **showText()** dapat di masukkan dalam menu editor pada aktor yang di tunjukkan dalam gambar 21 selanjutnya klik compile

```
public int skor_bom = 0;
public int skor_lebah = 0;

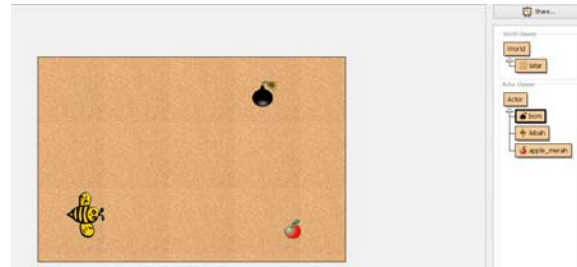
public void act()
{
    // Add your action code here.
    getWorld().showText("SKOR LEBAH = "+Integer.toString(skor_lebah), 75, 25);
}
```

Gambar 21. Perintah showText()

Contoh scenario game :

“Terdapat objek lebah dan objek bom. Player mengendalikan objek lebah. Kedua objek bersaing untuk menghabiskan objek buah apel. Jika skor objek bom mencapai angka 50 lebih dulu dari player, maka kalah, sedangkan jika player mencapai skor 50 terlebih dulu,

maka menang. Jika player menabrak bom, maka gameover.”



Gambar 22. Proses pembuatan game



Gambar 23. Tampilan Game

KESIMPULAN DAN SARAN**Kesimpulan**

Kesimpulan dari pelatihan pembuatan game menggunakan Greenfoot pada guru dan siswa SMK adalah guru-guru dan siswa sangat menyukai pemrograman menggunakan Greenfoot karena dalam membuat game bisa dilakukan dengan sangat mudah dan cepat sehingga mereka menjadi menyukai bahasa pemrograman.

Saran

Diharapkan guru-guru dapat menggunakan Greenfoot sebagai salah satu bahasa pemrograman dalam mengajar murid-muridnya, sehingga para murid terpancing untuk mempelajari bahas pemrograman karena kemudahannya.

DAFTAR PUSTAKA

- Michael Kölling, 2009, *Introduction to Programming with Greenfoot*, Pearson Education
- Michael Kölling, Poul Henriksen, *greenfoot: Combining Object Visualisation with Interaction*
- <http://www.greenfoot.org>