Implementasi Model *Linear Programming* Pada Game RPG

Muh. Fahmi Husein

Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
Email : muhfahmihusein@yahoo.com

ABSTRAK

Game yang akan dibangun ini ber-*genre* Role Playing Game (RPG). *Storyline game* ini berlatar petualangan di sebuah Desa. *Gameplay game* ini pemain di diminta untuk menyelesaikan permasalahan yang ada di sebuah Desa. Setiap warga desa memiliki masalah berbeda yang menjadi sebuah misi bagi pemain. Misi tersebut dapat berupa pencarian barang atau *item* dan membunuh musuh *No-player character* (*NPC*) di suatu tempat. Apabila misi tersebut dapat diselesaikan, pemain akan diberi hadiah berupa koin. Material bahan dapat digunakan pemain untuk membuat sebuah *item*, misalnya penggabungan bahan besi, nikel dan logam bisa dibuat menjadi pedang. Penggabungan jumlah besi, nikel, dan logam harus dileburkan menjadi pedang agar pedang yang dihasilkan memiliki kekuatan maksimal dan dapat digunakan untuk membunuh musuh. Permainan akan selesai jika pemain telah mengalahkan semua musuh, misalnya musuh mudah, musuh sedang, musuh sulit dan menyelesaikan setiap misi misalnya membunuh musuh yang menghalangi jalan, mencari sebuah item misalnya mengambil besi, nikel, dan logam mulia. Adapun metode yang digunakan dalam *linear programming* yaitu metode *simpleks*.

Setelah tahap implementasi selesai, maka dilakukan tahap pengujian terhadap aplikasi Implementasi model *linear programming* pada *game* RPG. Pengujian ditujukan pada tingkat kesesuaian nilai model *linear programming* yang sudah ditetapkan. Setelah dilakukan pengujian didapatkan kesimpulan bahwa rule – rule yang di uji sesuai dengan perancangan dan *linear programming* dapat mengoptimalkan penggunaan *item* dengan baik.

Kata kunci : RPG *game*, *linear programming*, keputusan

1. PENDAHULUAN

Game RPG merupakan game yang para pemainnya memainkan peran tokoh-tokoh-tokoh khayalan dan berkolaborasi untuk tujuan yang sama. Para pemain memilih aksi tokoh mereka berdasarkan karakteristik tokoh tersebut. Keberhasilan aksi mereka tergantung dari sistem peraturan permainan yang telah ditentukan. Asal tetap mengikuti peraturan yang ditentukan, para pemain bisa berimprovisasi membentuk arah dan hasil akhir permainan ini.[1]

Salah satu contoh game yang menjadi acuan yaitu *game* Digimon. Pemain yang mengatur jalannya permainan sehingga keinginan atau target tercapai. Permainan *game* ini berpetualang untuk mengumpulkan komponen-komponen sebuah *item*. Setiap karakter utama memiliki kemampuan *item* yang dapat digunakan untuk salah satu game *RPG*. Dalam game RPG pemain diharuskan menyelesaikan permasalahan yang ada di Desa tersebut. Setiap warga memiliki masalah yang berbeda yang menjadi sebuah misi bagi pemain. Apabila misi tersebut dapat diselesaikan, pemain akan di beri hadiah koin. Sedangkan material bahan dapat digunakan pemain untuk membuat sebuah item, misalnya penggabungan bahan besi, nikel, dan logam bisa dibuat menjadi pedang. Permainan akan selesai jika pemain telah mengalahkan semua musuh, misalnya musuh mudah, musuh sedang, musuh sulit dan menyelesaikan setiap misi misalnya membunuh musuh yang menghalangi jalan.

Metode yang digunakan dalam *linear programming* yaitu metode *simplek*. Kesesuaian *linear programming* antara nilai *input* dan nilai *output* sehingga dapat mengoptimalkan penggunaan *item*.

Berdasarkan uraian yang sudah dijelaskan, maka penelitian ini akan mengambil berjudul "**Implementasi Model Linear Programming Pada Game RPG".**

2. ISI PENELITIAN

2.1 Kecerdasan Buatan

Kecerdasan buatan berasal dari bahasa Inggris "*Artificial Intelligence*" atau disingkat AI, yaitu *intelligence* adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan. Kecerdasan yang dimaksud disini merajuk pada mesin yang mampu berpikir, menimbang tindakan yang diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia.

Menurut Alan Turing, ahli matematika berkebangsaan Inggris yang dijuluki bapak komputer modern dan pembongkar sandi Nazi

dalam era Perang Dunia II 1950, menetapkan definisi *Artificial Intelligence*: "Jika komputer tidak dapat dibedakan dengan manusia saat berbicara melalui terminal komputer, maka bisa dikatakan komputer itu cerdas, mempunyai kecerdasan".

Menurut John Carthy dari stanford mendefinisikan kecerdasan sebagai " kemampuan untuk mencapai sukses dalam menyelesaikan suatu permasalahan".

Menurut Herbert Alexander Simon (June 15, 1916-February 9, 2001): " Kecerdasan buatan (artificial intelligence) merupakan kawasan penelitian, aplikasi, dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas"

Menurut Rich and Knight (1991): " Kecerdasan buatan (AI) merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia"

Dari sini dapat dikatakan bahwa :

Cerdas adalah memiliki pengetahuan, pengalaman, dan penalaran untuk membuat keputusan dan mengambil tindakan. Jadi, agar mesin bisa cerdas (bertindak seperti manusia) maka harus diberi bekal pengetahuan dan diberi kemampuan untuk menalar.[2].

2.2 Klasifikasi *Game*

Game biasanya digunakan untuk mengkatagorikan sebuah *game* berdasarkan dari interaksi *game*play daripada perbedaan tampilan atau narasi. Sebuah jenis *game* dapat didefinisikan dengan kumpulan dari sebuah tipe permainan dari *game* tersebut. *Game* dapat diklasifikasikan menjadi dua, yaitu : [4]

2.2.1 Berdasarkan Jenis Platform yang Digunakan

Tiga kategori utama platform pada *game* adalah personal computer (PC), console, dan mobile : [4]

1. Personal Computer (PC)

   PC *game* adalah *game* yang dibuat untuk computer baik berbasis Windows, Mac, ataupun Linux. PC menyediakan kekuatan grafis dan pemrosesan yang kuat yang memungkinkan pengembangan untuk membuat *game* yang mutakhir. Tetapi kelemahan PC adalah mahal karena pengguna harus menggunakan hardware yang up-to-date untuk memainkan *game* PC dengan haik. Selain itu, kelemahan *game* PC bagi pengembang adalah banyaknya varian dari konfigurasi PC membuat sulit bagi pengembangan untuk memastikan *game* tersebut berjalan dengan benar pada semua setup PC
2. Console

   Console adalah hardware yang dibuat oleh pihak ketiga seperti Sony, Microsoft, dan Nintendo. Console terhubung ke televisi dan tujuan utamanya adalah untuk bermain *game*. *Game* Console sangat menarik bagi *game* pengembang karena mereka hanya perlu memikirkan satu konfigurasi hardoware ketika membuat software untuk konsol. Sangat kontras dengan PC yang memiliki opsi konfigurasi yang tak terbatas.
3. Mobile

   Mobile platform terdiri atas sesuatu yang portable dan bisa digenggam, termasuk ponsel, PDA, ipod, dan handle *game* seperti Nintendo Dsi atau Sony PSP. *Game* monile memiliki kontrol yang sederhana (terutama jika dibandingkan dengan PC)

2.2.1 Berdasarkan Genre Permainan

Genre *game* bisa dibagi menjadi 9, yaitu :

1. *Action*

   Action game adalah *game* dimana kebanyakan dari tantangan yang disajikan merupakan dari tes *physical skill* dan koordinasi pemain. Salah satu sub-genre action *game* adalah *shooters game*, baik yang 2D maupun 3D seperti *First Person, Shooters* (FPS)
2. *Strategy*

   Strategy game menantang pemain untuk mencapai kemenangan dengan perencanaan, khususnya melalui perencanaan serangkaian tindakan yang dilakukan melawan satu lawan atau lebih. Kemenangan diraih dengan perencanaan matang dan pengambilan keputusan yang optimal.
3. *Role Playing Game* (RPG)

   RPG adalah *game* dimana pemain mengontrol satu atau lebih karakter yang biasanya di desain oleh pemain itu sendiri, dan memandu mereka melewati berbagai rintangan yang diatur oleh komputer. Perkembangan karakter dalam hal kekuatan dan kemampuannya adalah kunci dari *game* jenis ini.
4. *Sports*

   Sports game mensimulasikan berbagai aspek dari olahraga atletik nyata maupun imajiner, apakah itu memainkan pertandingan, me-*manage* tim dan karir atau keduanya. Salah satu contoh *game* jenis ini adalah *Pro Evolution Soccer 2012* (PES 2012), dimana pemain bisa memainkan pertandingan, menjadi manajer tim, maupun menjadi pemain dan mengembangkan karirnya sendiri

5. *Vehicle Simulation*

   Vechicle Simulation membuat feeling mengedarai kendaraan, baik kendaraan nyata maupun kendaraan imajiner. Performa dan karakteristik kasar mesin harus menyerupai kenyataan, kecuali jika yang didesain adalah kendaraan imajiner
6. Construction and Management Simulation

   CMS *game* adalah *game* tentang proses. Tujuan pemain bukan untuk mengalahkan musuh, tetapi membangun sesuatu dengan konteks proses yang sedang berjalan. Semakin pemain mengerti dan mengontrol proses, semakin sukses sesuatu yang ia bangun. *Game* seperti ini biasanya menyediakan dua jenis permainan, yaitu mode bebas dimana pemain bebas membangun sesuatu dan mode misi dimana terdapat skenario hal apa yang harus dibangun oleh pemain.
7. Adventure

   Adventure *game* adalah cerita interaktif tentang karakter protagonist yang dimainkan oleh pemain. Penyampaian cerita dan eksplorasi adalah elemen inti dari *game* ini. Penyelesaian teka-teki dan tantangan konseptual adalah bagian besar dari permainan.
8. Artificial Life and Puzzle *Game*

   Artificial Life *game* adalah *game* yang membuat tiruan dari kehidupan sebenarnya. Biasanya ada dua jenis *game* ini, tiruan kehidupan manusia, contohnya The SIMS, dan tiruan kehidupan binatang, contohnya Tamagochi
9. Online *Game*

   Istilah online *game* disini mengacu kepada multiplayer *game* dimana mesin dari para pemain terhubung dengan jaringan
   tantangan konseptual adalah bagian besar dari permainan.
   1. *Artificial Life and Puzzle Game*
      Artificial Life game adalah *game* yang membuat tiruan dari kehidupan sebenarnya. Biasanya ada dua jenis *game* ini, tiruan kehidupan manusia, contohnya The SIMS, dan tiruan kehidupan binatang, contohnya Tamagochi.
   2. *Online Game*
      Istilah *online game* disini mengacu kepada *multiplayer game* dimana mesin dari para pemain terhubung dengan jaringan.

2.2.2 Sudut Pandang Permainan (*Game View Point*)

Sebuah *game* biasanya mempunyai sudut pandang permainan tersendiri disesuaikan berdasarkan *genre game* yang diambil. Berikut beberapa macam sudut pandang permainan yang biasa digunakan : [3]

1. *Side Scrolling*
   Adalah sudut pandang permainan yang terlihat dari samping dan memungkinkan karakter utama untuk bergerak dari kiri ke kanan serta memungkinkan background pada *game* seolah-olah bergeser mengikuti pergerakan karakter utama.
2. *Top Down*
   Adalah sudut pandang permainan yang memungkinkan karakter utama bermanuver ke empat arah namun cara permainannya sendiri bergeser dari bawah ke atas, dan biasanya *game* yang menggunakan sudut pandang permainan jenis ini adalah *shooter game*.
3. *Isometric*
   Adalah sudut pandang permainan yang memungkinkan permainan terlihat diantara sisi *Side Scrolling* dan juga *Top Down*, dan biasanya diterapkan pada *game* dengan *genre RTS* (*Real Time Strategy*).
   1]

2.3 *Linear Programming*

Linear programming adalah salah satu pendekatan matematika yang paling sering dipergunakan dan diterapkan dalam keputusan-keputusan manajerial. Tujuan dari penggunaan linear programming adalah untuk menyusun suatu model yang dapat dipergunakan untuk membantu pengambilan keputusan dalam menentukan alokasi yang optimal dari sumber daya ke berbagai alternatif. Penggunaan model *linear programming* dalam hal ini adalah mengalokasikan sumber daya tersebut sedemikian rupa sehingga akan maksimum atau alternatif minimum.[5]
Ada empat kondisi utama yang diperlukan dalam penerapannya :

1. Harus adanya sumber daya yang terbatas
2. Ada suatu fungsi tujuan seperti memaksimalkan atau meminimalkan
3. Harus ada linearitas
4. Harus ada keseragaman

2.4 Analisis *Game*

Game yang akan dibangun ini Game yang dibangun merupakan *game* 3D yang ber-*genre Role Playing Game* (RPG). Permainan ini berlatar belakang petualangan di Desa, pemain diminta untuk menyelesaikan permasalahan yang ada di desa tersebut. Dengan cara mengalahkan semua musuh.

2.5 **Gameplay**

Permainan ini berlatar belakang petualangan di Desa, pemain diminta untuk menyelesaikan permasalahan yang ada di desa tersebut. Setiap warga desa memiliki masalah berbeda yang menjadi sebuah misi bagi pemain. Misi tersebut dapat berupa pencarian sebuah barang atau *item* dan membunuh musuh (*NPC*) di suatu

tempat. Apabila misi tersebut dapat diselesaikan, pemain akan diberi hadiah berupa koin.. Sedangkan material bahan dapat digunakan pemain untuk membuat sebuah item, misalnya penggabungan bahan besi, nikel dan logam bisa dibuat menjadi pedang. Penggabungan jumlah besi, nikel, dan logam harus dileburkan menjadi pedang agar pedang yang dihasilkan memiliki kekuatan maksimal dan dapat digunakan untuk membunuh musuh. Permainan akan selesai jika pemain telah mengalahkan semua musuh, misalnya musuh mudah, musuh sedang, musuh sulit dan menyelesaikan setiap misi misalnya membunuh musuh yang menghalangi jalan, mencari sebuah item misalnya mengambil besi, nikel, dan logam mulia.

2.6 Analisis Model *Linear Programming*

Model *linear programming* yang dapat di selesaikan dengan beberapa metode yang salah satunya adalah metode *simpleks* akan diterapkan pada saat mengoptimalkan penggunaan *item* dan untuk menaikan level karakter.

Contoh kasus dalam game yang akan dibangun :

1. Untuk menambah damage pedang menggunakan 3 variabel yaitu : A (Besi), B (Nikel), C (Logam). Untuk meleburkan setiap variabel membutuhkan biaya. Tetapi setiap meleburkan variable membutuhkan biaya yang berbeda-beda. Untuk meleburkan Besi membutuhkan biaya 8 koin, meleburkan Nikel membutuhkan biaya 6 koin dan meleburkan Logam membutuhkan biaya 10 koin. Sedangkan koin yang disediakan hanya 1200 koin. Hitung maksimasi kekuatan yang didapat bila 1 variabel Besi menambah 12 damage, Nikel 10 damage, dan Logam 15 damage. Bila setiap menambah damage pedang hanya memuat 180 pcs dari jumlah keseluruhan variable?

Maka penyelesaianya sebagai berikut :
Variabel Keputusan :
A = menyatakan jumlah besi yang harus dikumpulkan
B = menyatakan jumlah nikel yang harus dikumpulkan
C = menyatakan jumlah logam yang harus dikumpulkan

Formulasi Matematika Awal :
Fungsi Tujuan
Maksimasi $Z = 12\,A + 10\,B + 15\,C$
Fungsi Pembatas
$8\,A + 6\,B + 10\,C \leq 1200$
$A + B + C \leq 180$
$A, B, C \geq 0$

Formulasi Bentuk Standar *Simpleks*:
Fungsi Tujuan
Maksimasi $Z = 12\,A + 10\,B + 15\,C$
Atau
$Z - 12\,A - 10\,B - 15\,C$
Fungsi Pembatas
$8\,A + 6\,B + 10\,C = 1200$
$A + B + C = 180$

Formulasi Bentuk Standar Programa Linear :
Persamaan
(0) $Z - 12\,A - 10\,B - 15\,C = 0$
(1) $8\,A + 6\,B + 10\,C + S1 = 0$
(2) $A + B + C + S2 = 0$

Formulasi Bentuk Standar *Simpleks* :
Persamaan
(0) $Z - 12\,A - 10\,B - 15\,C = 0$
(1) $8\,A + 6\,B + 10\,C + S1 = 0$
(2) $A + B + C + S2 = 0$

Tabel *Simplexs* Iterasi 1 :

Cj	Basic Variables	12 X1	10 X2	15 X3	0 slack 1	0 slack 2	Quantity
Iteration 1							
0	slack 1	8	6	10	1	0	1.200
0	slack 2	1	1	1	0	1	180
	zj	0	0	0	0	0	0
	cj-zj	12	10	15	0	0	
Iteration 2							
15	X3	0,8	0,6	1	0,1	0	120
0	slack 2	0,2	0,4	0	-0,1	1	60
	zj	12	9	15	1,5	0	1.800
	cj-zj	0	1,0	0	-1,5	0	
Iteration 3							
15	X3	0,5	0	1	0,25	-1,5	30,0
10	X2	0,5	1	0	-0,25	2,5	150
	zj	12,5	10	15	1,25	2,5	1.950
	cj-zj	-0,5	0	0	-1,25	-2,5	

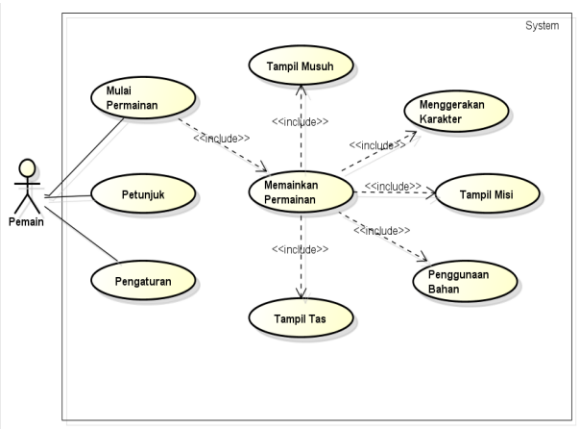
Kesimpulan :
Untuk memaksimalkan kekuatan, maka dibutuhkan variabel Logam sebanyak 30pcs dan variabel Nikel sebanyak 150pcs, setelah dijumlahkan variabel keseluruhan telah mencapai 180pcs, dan dari kedua variabel tersebut menghasilkan sebanyak 1.950 kekuatan.

2.7 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik sesuai kebutuhan. Analisis kebutuhan fungsional dimodelkan dengan menggunakan UML *(Unified Modeling Language)*.

2.8 Use Case Diagram

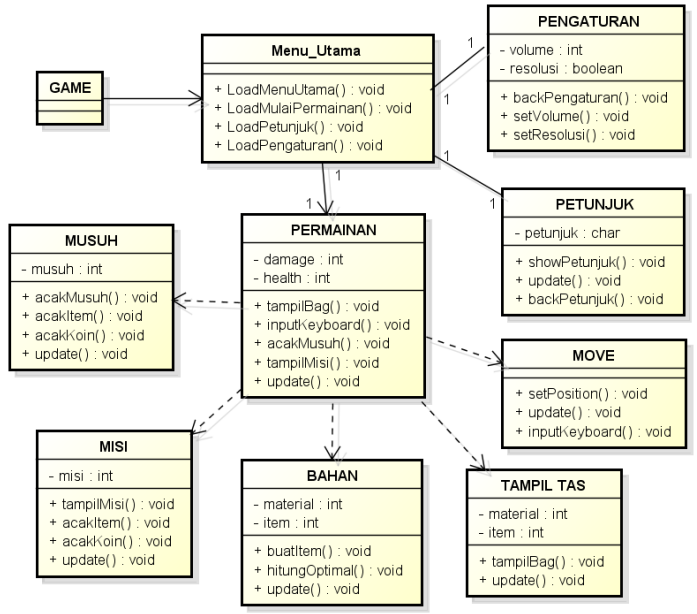
Diagram *use case* menggambarkan hubungan antara aktor dan sistem yang dibangun. Perancangan proses-proses yang terdapat pada *game* yang akan dibangun akan digambarkan dengan diagram *use case*. Interaksi antara aktor dengan aplikasi *game* yang akan dibangun digambarkan pada *diagram use case* pada Gambar 3.3



Gambar 3.3 *Use case Diagram*

2.9 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas.



Gambar 3.22 *Class Diagram*

2.10 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan setiap tampilan program yang dibangun. Antarmuka yang disajikan berikut merupakan antarmuka yang mewakili keseluruhan program.

1. Implementasi Antarmuka Menu Utama

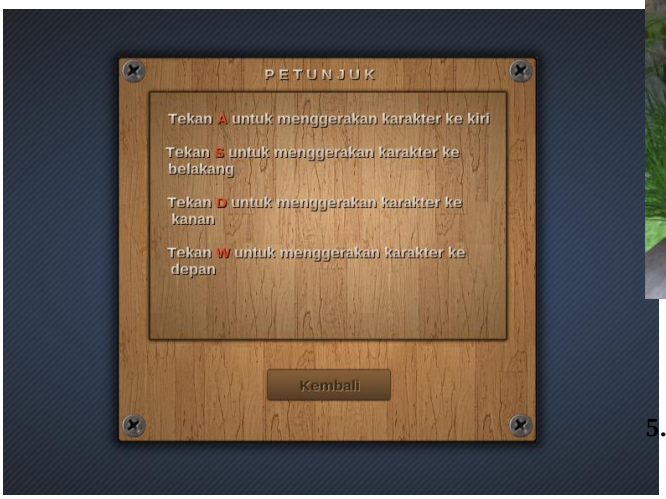
Gambar 4.1 menunjukan antarmuka menu utama dimana halaman tersebut menampilkan tombol mulai, tombol petunjuk, tombol pengaturan. Dan tombol keluar.



Gambar 4.1 Tampilan Menu Utama

2. Implementasi Antarmuka Petunjuk

Gambar 4.2 menunjukan antarmuka petunjuk dimana halaman tersebut menampilkan petunjuk.



Gambar 4.2 Tampilan Petunjuk

3. Implementasi Antarmuka Pengaturan

Gambar 4.3 menunjukan antarmuka pengaturan dimana halaman tersebut menampilkan tombol lihat level, tombol suara, dan tombol kembali.



Gambar 4.3 Tampilan Pengaturan

4. Implementasi Antarmuka Saat Mulai Permainan

Gambar 4.4 menunjukan antarmuka saat mulai permainan dimana halaman tersebut menampilkan darah, koin, tombol memperbesar ukuran peta,tombol memperkecil ukuran peta, tombol lihat peta, dan tombol Kunci.



Gambar 4.4 Tampilan Saat Mulai Permainan

5. Implementasi Antarmuka Saat Melawan Musuh

Gambar 4.5 menunjukan antarmuka saat melawan musuh dimana halaman tersebut menampilkan tampilan saat melawan musuh.



Gambar 4.5 Tampilan Saat Melawan Musuh

6. Implementasi Antarmuka Saat di Build Item

Gambar 4.6 menunjukan antarmuka saat menambah kekuatan pedang dimana halaman tersebut menampilkan tampilan saat melihat hasil.



Gambar 4.6 Tampilan Saat di Build Item

7. Implementasi Antarmuka Saat di Store

Gambar 4.7 menunjukan antarmuka saat di Store dimana halaman tersebut menampilkan tampilan saat melihat membeli *item*.



Gambar 4.7 Tampilan Saat di Store

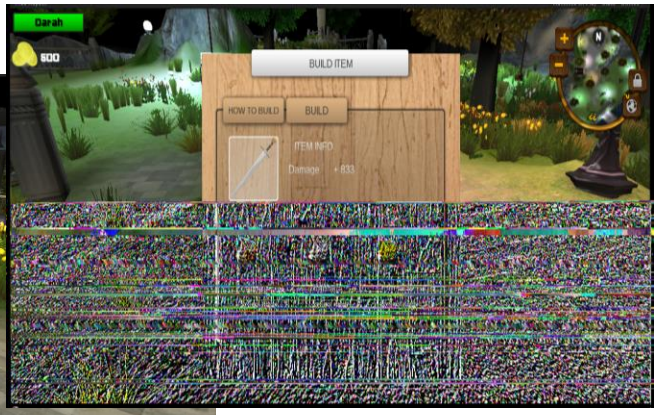
Setelah memainkan permainan, pemain akan mendapatkan nilai yang didapat dengan solusi yang berbeda tergantung dengan nilai pemain yang didapat sehingga nilai optimal berbeda.

1. Pengujian nilai 0 yang didapat akan menghasilkan sebanyak 0 Damage.
2.



Gambar 4.13 Pengujian nilai optimal dengan *NPC*

3. Pengujian nilai 500 yang didapat akan menghasilkan sebanyak 833 Damage.



Gambar 4.14 Pengujian nilai optimal dengan *NPC*

Kesimpulan Pengujian

Setelah dilakukan pengujian didapatkan kesimpulan bahwa rule – rule yang di uji sesuai dengan perancangan dan *linear programming* memiliki akurasi yang baik antara nilai *input* dan nilai *output*.

PENUTUP

Berdasarkan hasil yang didapatkan dalam penelitian dan penyusunan skripsi ini serta disesuiakan dengan tujuan maka diperoleh kesimpulan bahwa *linear programming* memiliki akurasi yang baik antara nilai *input* dan *output* sehingga dapat mengoptimalkan penggunaan *item* pada *game* RPG.

DAFTAR PUSTAKA

[1] Komputer. Wahana, *Membuat Game RPG dengan RPG Maker*, Andi Publishing, 2013.

[2] T. Edi Mulyanto, Vincent Suharto. 2010, *Kecerdasan Buatan*, Jakarta : Andi Offset.

[3] Suryanto, ST, MSc, *Artificial Intelligence Searching, Reasoning, Planning and Learning,* Penerbit Informatika, Bandung, 2007.

[4] Von Neuman, John, Oskar Moegenstern. 1953, *Theory of Game And Economic Behavior*. Princenton: Pricenton University Press.

[5] Siswanto, *Operations Research jilid 1*, Penerbit Erlanggan, Jakarta : 2007.

[6] S. Rosa dan Shalahuddin, M, *Rekayasa Perangkat Lunak*, Penerbit Informatika, 2013.

[7] "Unity Technologies," 2014. [Online]. Available: http://unity3d.com/.[Accessed 10 October 2013].

[8] Gumelar, Asep. 2007. *Memanipulasi & Editing Image dengan Adobe*[Online] diakses pada 21 September 2014 jam 09.00 Asep Gumelar.(2007).http://pfi3pdata.litbang.deptan.go.id/laporan/one/24/.

[9] TP. P. Widodo dan Heriawati, *Menggunakan UML*, Bandung: Informatika, 2011.

[10] R. S. Pressman, *Software Engineering A Practitioner's Approach 7th Edition*, New York: McGraw-Hill, 2010.