



JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER

USULAN TUGAS AKHIR

1. IDENTITAS PENGUSUL

NAMA : Nabilah
NRP : 5110100708
DOSEN WALI : Dr. Eng. Nanik Suciati, S.Kom., M.Kom.
DOSEN PEMBIMBING : 1. Imam Kuswardayan, S.Kom., M.T.
2. Abdul Munif, S.Kom., M.Sc.

2. JUDUL TUGAS AKHIR

“Implementasi Modul *Random Dungeon Route* Generator pada *Game Sosial Heart Meister*”

3. LATAR BELAKANG

Role-Playing Game (RPG) merupakan permainan di mana para pemainnya memainkan peran dari tokoh-tokoh khayalan. Para pemain dapat memilih aksi tokoh-tokoh berdasarkan karakteristik dari tokoh yang diperankan. Keberhasilan aksi pemain tergantung dari sistem permainan yang telah ditentukan [1].

Dungeon merupakan suatu arena dimana pemain dapat bertarung melawan musuh. *Dungeon* biasa ditemui dalam *game* bergenre *Role-Playing Game* (RPG). Setiap permainan memiliki lebih dari satu *dungeon*, dimana setiap *dungeon* memiliki tingkat kesulitan yang berbeda-beda.

Game sosial *Heart Meister* yang bergenre *Role-Playing Game* ini memiliki fitur *epic battle*, di mana dalam *epic battle* terdapat rute *dungeon map* yang selalu berubah-ubah mengikuti level *pet* tertinggi yang dimiliki pemain pada saat itu, sehingga dibutuhkan

random dungeon route generator untuk membuat variasi rute *dungeon* serta kesulitan yang berbeda-beda pada tiap levelnya.

4. RUMUSAN MASALAH

Beberapa permasalahan yang digunakan sebagai rumusan masalah adalah sebagai berikut.

- a) Bagaimana mengimplementasikan algoritma Dijkstra ke dalam *random dungeon route* generator pada *game* sosial *Heart Meister*.
- b) Bagaimana membuat rute *dungeon* dengan hasil acak, namun memungkinkan pemain untuk tetap bisa menang.
- c) Bagaimana mengintegrasikan modul ke dalam sistem *game Heart Meister*.

5. BATASAN MASALAH

Batasan lingkup masalah yang dibahas pada tugas akhir ini adalah sebagai berikut.

- a) Algoritma yang digunakan untuk mencari jalur dari *start point* ke *final point* (*boss dungeon*) yaitu algoritma Dijkstra.
- b) Tingkat kesulitan *dungeon* bergantung pada level tertinggi dari *party pet* yang dimiliki pemain.

6. TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut.

- a) Mengimplementasikan algoritma Dijkstra dalam pembuatan *dungeon* generator.
- b) Menghasilkan *random dungeon route* generator yang dapat mengacak *dungeon route* yang sesuai dengan level pemain.

7. MANFAAT TUGAS AKHIR

Manfaat yang diharapkan dari pengerjaan tugas akhir ini adalah untuk membantu mengacak *dungeon route* dalam *epic battle*, sehingga memberikan variasi rute *map dungeon* yang harus dilewati oleh pemain.

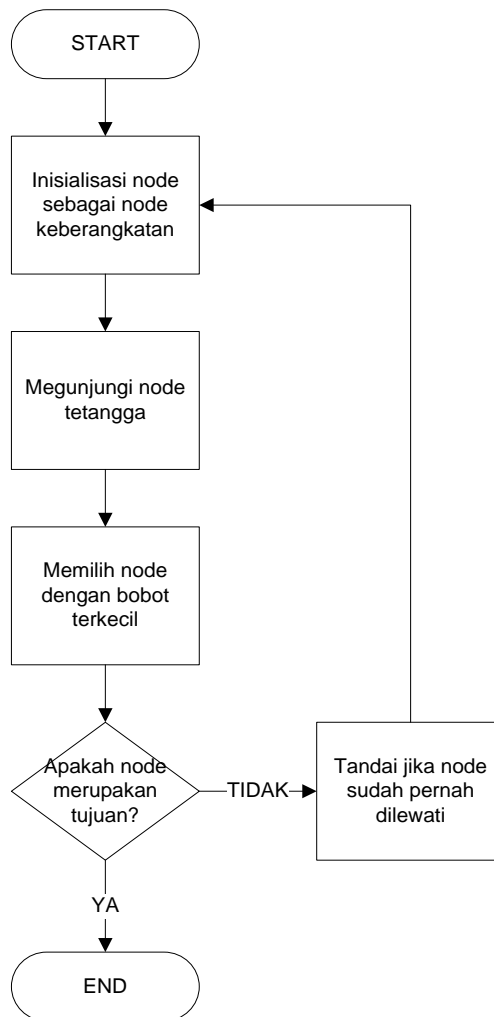
8. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan teori-teori serta pustaka yang dipakai pada waktu penelitian

1. Algoritma Dijkstra

Algoritma Dijkstra bekerja dengan cara mengunjungi simpul-simpul yang ada pada graf, dimulai dengan simpul sumber. Kemudian secara berulang memilih simpul-simpul yang lokasinya terdekat dan menghitung total bobot semua sisi

yang dilewati untuk mencapai simpul yang dituju [2]. *Flowchart* dari algoritma Dijkstra dapat dilihat pada Gambar 1.



Gambar 1. *Flowchart* algoritma Dijkstra

2. C#

C# (dibaca: *see-sharp*) merupakan bahasa pemrograman yang bersifat *general purpose*, berorientasi objek, yang dapat digunakan untuk membuat program di atas arsitektur Microsoft .NET Framework. C# dapat disebut sebagai bahasa pemrograman yang paling mencerminkan dasar *Common Language Runtime* (CLR) dimana semua program-program .NET berjalan [3].

3. Unity

Unity adalah sebuah ekosistem pengembangan *game* yang terintegrasi dengan kumpulan alat intuitif yang sangat berguna untuk membangun permainan interaktif seperti pencahayaan, efek khusus, dan animasi. Unity dapat digunakan untuk membangun *game* dua dimensi ataupun tiga dimensi. Unity mendukung

pengembangan *game* untuk iOS, Android, Windows, BlackBerry 10, OS X, Linux, Web Player, PS3, Xbox 360 dan Wii U [4].

9. RINGKASAN ISI TUGAS AKHIR

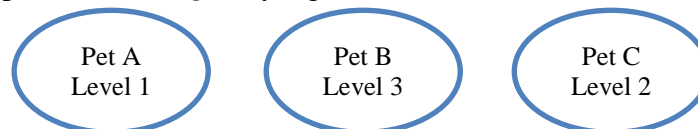
Heart Meister adalah sebuah *social role-playing game* ber-platform Android. Di dalam *game* ini pemain dapat memiliki satu atau lebih *heart*, yaitu peliharaan yang memiliki tiga tipe berbeda dan dapat digunakan untuk bertarung melawan musuh dalam *dungeon* yang ada, *boss* pada *epic battle*, maupun pemain lain dalam PvP *mode*.

Tugas Akhir ini akan membangun *random dungeon route generator* untuk *game* sosial *Heart Meister* dengan implementasi algoritma Dijkstra di dalamnya. *Random dungeon route generator* ini nantinya akan digunakan untuk mengacak *dungeon route* yang terdapat di dalam fitur *epic battle*.

Di dalam *epic battle*, *map dungeon* sudah ditentukan sebelumnya berdasarkan level *pet* tertinggi milik pemain. Level *pet* diklasifikasikan ke dalam beberapa tingkatan. Tingkat tersebut akan menentukan bentuk dasar dari *map*.

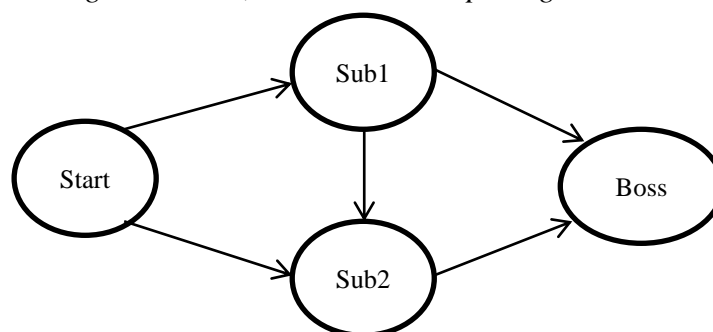
Contoh penentuan rute dalam proses *generate* adalah sebagai berikut.

1. Mengecek level tertinggi dari *party pet* pemain. Level tersebut akan menentukan *difficulty* dari *dungeon*. Jika pemain memiliki *party* yang berisi *pet* a, b dan c dengan penentuan level seperti pada Gambar 2, maka level yang akan dijadikan sebagai penentuan *dungeon* yang akan dimasuki adalah level 3



Gambar 2. Contoh *party pet* pemain

Dalam *dungeon* level 3, bentuk dasar *map dungeon* bisa dilihat pada Gambar 3.



Gambar 3. Bentuk dasar *map dungeon* pada level 3

2. Dengan menggunakan algoritma Dijkstra dimulai pencarian rute yang akan dilewati. Langkah algoritma Dijkstra yang digunakan adalah sebagai berikut.

1. Setiap rute sebelumnya telah diberi bobot nilai yang didapat dari jumlah *stats point* yang dimiliki musuh. Kombinasi musuh di dalam *sub dungeon* ditentukan secara acak. Misal setelah pengacakan kombinasi musuh untuk Sub1 adalah musuh A dan musuh B, sedangkan untuk Sub2 kombinasi musuh adalah musuh C dan musuh D. untuk *stats point* musuh dan *boss* dapat dilihat pada Tabel 1.

Tabel 1. Stats Point Musuh

Stats Point	Musuh A	Musuh B	Musuh C	Musuh D	Boss
Attack	10	13	8	15	20
Defense	10	10	15	15	10
Speed	10	12	3	8	20
Special Attack	10	12	8	5	20
Special Defense	10	10	15	5	10
Health Point	10	8	17	12	50
Jumlah	60	65	66	60	130

Dari Tabel 1 dihitung jumlah kombinasi *stats pet* untuk dijadikan bobot *path* di dalam *map*.

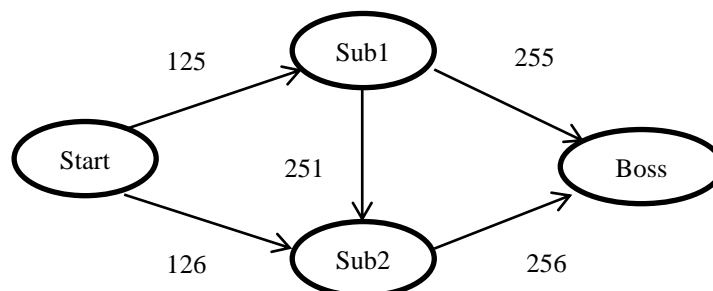
Start → Sub1 = 60 + 65 = 125

Start → Sub2 = 66 + 60 = 126

Sub1 → Sub2 = 125 + 126 = 251

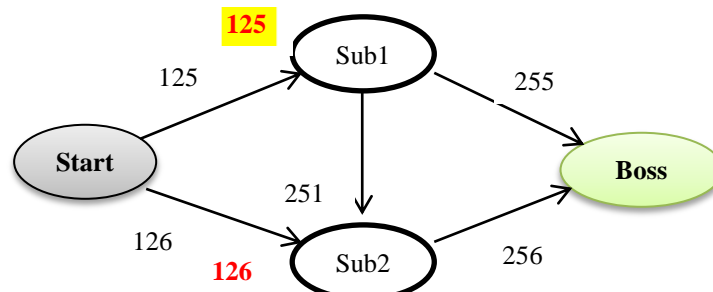
Sub1 → Boss = 125 + 130 = 255

Sub2 → Boss = 126 + 130 = 256



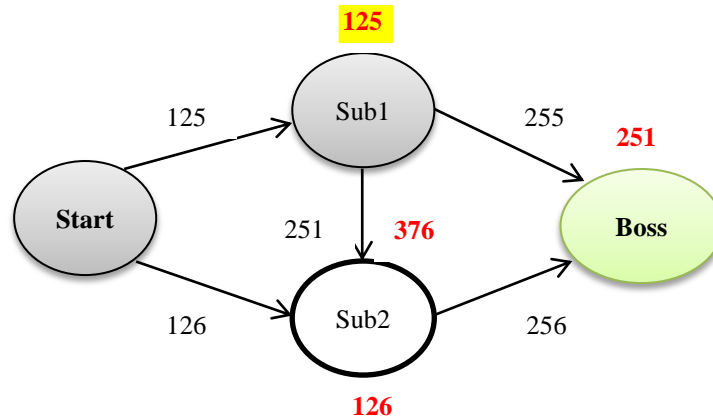
Gambar 4. Pembobotan *path*

2. Dijkstra melakukan kalkulasi terhadap *node* tetangga yang berhubungan langsung dengan *node* keberangkatan (Start). Hasil yang didapat adalah *node* Sub1 karena bobot yang didapat di *node* Sub1 lebih kecil dari *node* Sub2.



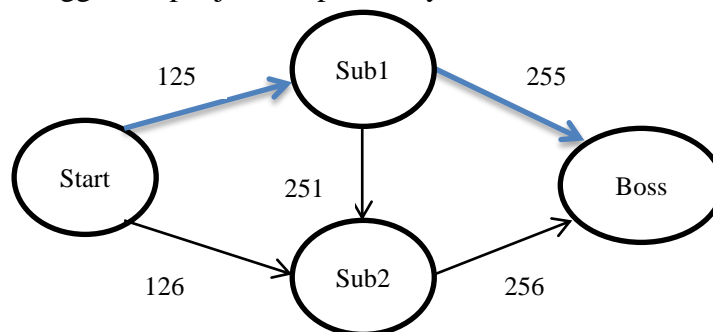
Gambar 5. Mengunjungi *node tetangga* dari *node* keberangkatan

3. *Node* Sub1 diatur menjadi *node* keberangkatan dan ditandai sebagai *node* yang pernah didatangi. Kemudian kalkulasi dilakukan kembali terhadap *node* tetangga yang terhubung langsung dengan *node* yang telah didatangi. Dari hasil kalkulasi didapat *node* Boss (*node* tujuan).



Gambar 6. Mengunjungi *node* tetangga dari Sub1

4. Jika *node* tujuan telah tercapai maka kalkulasi Dijkstra dinyatakan selesai, sehingga didapat jalur terpendeknya adalah Start - Sub1 - Boss.



Gambar 7. Hasil akhir rute yang didapat dari algoritma Dijkstra

10.METODOLOGI

a. Penyusunan proposal Tugas Akhir

Proposal tugas akhir ditulis untuk mengajukan ide atas pengajuan tugas akhir. Proposal ini juga mengandung proyeksi dari ide tugas akhir yang diajukan.

b. Studi literatur

Pada tahap ini dilakukan pencarian informasi dan studi literatur yang diperlukan untuk proses perancangan serta implementasi aplikasi yang akan dibangun. Literatur yang digunakan adalah terkait dengan implementasi algoritma Dijkstra dalam pencarian rute.

c. Analisis dan desain perangkat lunak

Pada tahap ini, dilakukan analisis serta rancangan algoritma yang akan dibangun dengan mengacu pada proposal yang telah dibuat dan hasil studi literatur yang telah dilakukan sebelumnya.

d. Implementasi perangkat lunak

Pada tahap ini akan dilakukan pembangunan aplikasi berdasarkan bentuk implementasi dari hasil analisis dan desain algoritma yang telah dibuat sebelumnya.

e. Pengujian dan evaluasi

Pada tahap ini dilakukan uji coba dengan memasukkan inputan level tertinggi *party pet* pemain yang berbeda-beda, kemudian evaluasi dilakukan dengan mengidentifikasi apakah hasil akhir dari *map dungeon* yang dikeluarkan sesuai dengan level tiap pemain.

f. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain sebagai berikut.

1. Pendahuluan
 - a. Latar Belakang
 - b. Rumusan Masalah
 - c. Batasan Tugas Akhir
 - d. Tujuan
 - e. Metodologi
 - f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

11. JADWAL KEGIATAN

Jadwal kegiatan dalam pengerjaan tugas akhir ini dapat dilihat pada Tabel 2.

Tabel 2. Jadwal Kegiatan Pengerjaan Tugas Akhir

Tahapan	2014																							
	Februari				Maret				April				Mei				Juni							
Penyusunan Proposal																								
Studi Literatur																								
Perancangan sistem																								
Implementasi																								
Pengujian dan evaluasi																								
Penyusunan buku																								

12. DAFTAR PUSTAKA

- [1] Ellebart, Pengertian RPG, 1 September 2010. [Online]. Available: <http://s4.zetaboar.ds.com/Ellebart/topic/8002133/1/>. [Accessed 17 Februari 2014].
- [2] Dibi Khairurrazi Budiarsyah, "ALGORITMA DIJKSTRA, BELLMAN-FORD, DAN FLOYD-WARSHALL UNTUK MENCARI RUTE TERPENDEK DARI SUATU GRAF," p. 15, Desember 2010.
- [3] Wikibooks, Belajar Bahasa C#, 21 Januari 2014. [Online]. Available: http://id.wikibooks.org/wiki/Belajar_Bahasa_C_sharp/Pendahuluan. [Accessed 17 Februari 2014].
- [4] Unity, Unity - Game engine, tools and multiplatform, 2014. [Online]. Available: <http://unity3d.com/unity>. [Accessed 17 Februari 2014].
- [5] Ajurna, Pengertian dan Teori Algoritma Dijkstra, April 2013. [Online]. Available: <http://www.ajurna.net/2013/04/pengertian-dan-teori-algoritma-dijkstra.html>. [Accessed 5 Maret 2014].
- [6] Ernest Adams, *Fundamental of Game Design*, 2nd ed., Christopher Weaver, Ed. Berkeley, United States of America: New Risers, 2010.