PENERAPAN ALGORITMA *DIJSKTRA* UNTUK *GAME* HIJAIYAH

SKRIPSI

Oleh :

NUR ZAIDAH

NIM. 09650196

JURUSAN TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM

MALANG

2016

PENERAPAN ALGORITMA *DIJSKTRA* UNTUK *GAME* HIJAYIAH

SKRIPSI

Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN)
Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :

Nur Zaidah

NIM. 09650196

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK BRAHIM MALANG
2016

PENERAPAN ALGORITMA DIJSKTRA
UNTUK GAME HIJAIYAH

SKRIPSI

Oleh :

NUR ZAIDAH
NIM. 09650196

Telah disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Dr. Muhammad Faisal, M.T	**Dr. Suhartono, M.Kom**
NIP. 19740510 200501 1 007	**NIP. 19680519 200312 1 001**

Tanggal, 8 Juni 2016

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Cahyo Crysdian

NIP. 19740424 200901 1 008

PENERAPAN ALGORITMA DIJSKTRA UNTUK GAME HIJAIYAH

SKRIPSI

Oleh :

NUR ZAIDAH
NIM. 09650196

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal 23 Juni 2016

Susunan Dewan Penguji		Tanda Tangan
1. Penguji Utama	: Hani Nurhayati, M.T NIP. 19780625 200801 2 006	( )
2. Ketua	: Fresy Nugroho, M.T NIP. 19710722 201101 1 001	( )
3. Sekretaris	: Dr. Muhammad Faisal, M.T NIP. 19740510 200501 1 007	( )
4. Anggota	: Dr. Suhartono, M.Kom NIP. 19680519 200312 1 001	( )

Mengetahui dan Mengesahkan
Ketua Jurusan Teknik Informatika

Dr.Cahyo Crysdian, M.CS
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Nur Zaidah

NIM : 09650196

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Penelitian : **PENERAPAN ALGORITMA DIJKSTRA UNTUK GAME HIJAIYAH**

Menyatakan dengan sebenar-benarnya bahwa hasil penelitian saya ini tidak terdapat unsur-unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata hasil penelitian ini terbukti terdapat unsur-unsur penjiplakan, maka saya bersedia untuk mempertanggungjawabkan, serta diproses sesuai peraturan yang berlaku.

Malang, 8 Juni 2016

Yang membuat pernyataan

Nur Zaidah

NIM. 09650196

MOTO

Hasil Terindah, Terbaik dan Paling Mengesankan

Adalah Apa Saja yang Selesai

"Allah Tidak Akan Merubah Keadaan Suatu Kaum,

Hingga Mereka Mau Merubahnya Sendiri"

HALAMAN PERSEMBAHAN

The Almighty and The Prophet

Segala puji bagi Allah SWT. Kita memuji-Nya memohon petolongan dan ampunan kepada-Nya. Segala puji syukur kepada Allah atas limpahan rahmad-Nya. Kepada Rosulullah Muhammad SAW yang telah menjadi penuntun dan panutan bagi seluruh umah manusia.

Lovely Family

Terimakasih kepada Bapak, Ibu, Suami dan Anak-anak ku yang memberikan kasih sayang yang luar biasa, dukungan, do'a, kerja keras serta kesabarannya. Sheryl dan Hafizh terimakasih yang selalu memberikan semangat kepada mama.

Friends

Para pejuang dakwah IKATAN Mahasiswa Muhammadiyah (IMM) UIN Maliki Malang khususnya IMM komisariat Revivalis, terima kasih kerja samanya dalam bahu membahu mencari ilmu. Semoga semoga apa yang kita harapkan semuanya tercapai dan di berkahi oleh Allah SWT.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji bagi Allah SWT tuhan semesta alam, karena atas segala rahmat dan karunia-Nya sehingga penulis mampu menyelesaikan skripsi dengan judul "Permainan Hai Hijaiyyah Untuk Media Pembelajaran Huruf Hijaiyyah Menggunakan Dijkstra Algorithm Sebagai Pathfinding Pada Agen Musuh" dengan baik dan lancar. Shalawat serta salam selalu tercurah kepada tauladan terbaik Nabi Agung Muhammad SAW yang telah membimbing umatnya dari zaman kebodohan menuju Islam yang *rahmatan lil alamiin.*

Dalam penyelesaian skripsi ini, banyak pihak yang telah memberikan bantuan baik secara moril, nasihat dan semangat maupun materiil. Atas segala bantuan yang telah diberikan, penulis ingin menyampaikan doa dan ucapan terimakasih yang sedalam-dalamnya kepada :

1. Prof. DR. H. Mudjia Raharjo, M.Si, selaku rektor UIN Maulana Malik Ibrahim Malang beserta seluruh staf. Bakti Bapak dan Ibu sekalian terhadap UIN Maliki Malang turut membesarkan dan mencerdaskan penulis.
2. Dr. Hj. Bayyinatul M., drh., M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang beserta seluruh staf. Bapak dan ibu sekalian sangat berjasa memupuk dan menumbuhkan semangat untuk maju kepada penulis.
3. Bapak Dr. Cahyo Crysdian, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang, yang sudah memberi banyak pengetahuan, inspirasi dan pengalaman yang berharga.
4. Bapak Dr. M Faisal, M.T selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, memotivasi, mengarahkan dan memberi masukan kepada penulis dalam pengerjaan skripsi ini hingga akhir.
5. Bapak Dr. Suhartono,M.Kom selaku dosen pembimbing II yang juga senantiasa memberi masukan dan nasihat serta petunjuk dalam penyusunan skripsi ini.

6. Ayah, Ibu, Suami, Anak-Anak dan Kakak Adik serta keluarga besar tercinta yang selalu memberi dukungan yang tak terhingga serta doa yang senantiasa mengiringi setiap langkah penulis.
7. Segenap Dosen Teknik Informatika yang telah memberikan bimbingan keilmuan kepada penulis selama masa studi.
8. Teman – teman seperjuangan Teknik Informatika 2009
9. Para peneliti yang telah mengembangkan Game dengan Engine *Unity3d* yang menjadi acuan penulis dalam pembuatan skripsi ini. Serta semua pihak yang telah membantu yang tidak bisa disebutkan satu satu. Terimakasih banyak.

Berbagai kekurangan dan kesalahan mungkin pembaca temukan dalam penulisan skripsi ini, untuk itu penulis menerima segala kritik dan saran yang membangun dari pembaca sekalian. Semoga apa yang menjadi kekurangan bisa disempurnakan oleh peneliti selanjutnya dan semoga karya ini senantiasa dapat memberi manfaat. Amin. *Wassalamualaikum Wr. Wb.*

Malang, 8 Juni 2016

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Zaidah, Nur. 2016. **Penerapan Algoritma Dijsktra untuk Game Hijaiah.** Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing : (I) Dr. M. Faisal, M.T, (II) Dr. Suhartono, M.Kom

Kata Kunci: *Pencarian Rute, Agen Musuh, Permainan Edukasi, Dijkstra*

Huruf hijaiah merupakan dasar untuk mempelajari bahasa al Quran. Dalam memahami al-Qur'an, setiap umat islam wajib mengetahui huruf hijaiah sebagai dasar dalam mempelajari al-Qur'an dan kandungannya. Namun dewasa ini anak-anak semakin kesulitan dalam mempelajari huruf hijaiah ditengah kesibukan mempelajari ilmu umum di sekolah. Padahal al-Qur'an merupakan pedoman utama untuk mengarungi kehidupan di dunia.

Demi mendukung melestarikan Bahasa Arab, peneliti bermaksud menghadirkan salah satu solusi menyenangkan untuk mempermudahkan pengenalan huruf hijaiah ke dalam sebuah *game*. Permainan tersebut bergenre edukasi. Penelitian ini menjelaskan bagaimana merancang algoritma *pathfinding* pada agen musuh.

Permainan Hai Hijaiyah untuk Media Pembelajaran Huruf Hijaiah Menggunakan *Dijkstra Algorithm* sebagai Pathfinding Pada Agen Musuh berbasis desktop yang dibangun dengan memanfaatkan Unity3d. Pemain bertugas untuk menyelesaikan misi yang sarat akan nilai pengenalan terhadap huruf-huruf hijaiah. Agen musuh mengimplementasikan sistem kecerdasan buatan yang akan mencari keberadaan pemain.

Implementasi kecerdasan buatan pada penelitian ini memanfaatkan metode Dijkstra. Metode Dijkstra digunakan sebagai pencarian rute terdekat. Penelitian ini difokuskan pada platform desktop

ABSTRACT

Zaidah, Nur. 2016. **Implementation of Dijkstra Algorithm for Hijaiyyah Game**. Thesis. Department of Informatics Engineering Faculty of Science and Technology of the State Islamic University of Maulana Malik Ibrahim Malang

Supervisor: (I) Dr. M. Faisal, M.T, (II) Dr. Suhartono, M.Kom

---

Keyword: Pathfinding, NPC, Educational Games, Dijkstra

Hijaiah letter is the basis for studying the language of the Koran. In understanding the Qur'an, all Muslims must know hijaiah letter as a basis for studying the Koran and its contents. But today's kids are getting difficulty in learning letters hijaiah amid a bustle studying general science in school. In fact, the Koran is the main guideline for living life in the world.

In favor of preserving the Arabic language, researchers intend to present a fun solution to facilitate the introduction of the letter hijaiah into a game. The genre of educational games. This paper describes how to design pathfinding algorithm at enemy agents.

Games Hai Hijaiyah for Media Learning Letter Hijaiah Using Dijkstra Algorithm as pathfinding In Enemy Agent-based desktop that is built by using Unity3D. Players commissioned to complete the mission will be full recognition of the value of the letters hijaiah. Implementing an enemy agent artificial intelligence system that will look for the presence of players.

Implementation of artificial intelligence in this study utilizing Dijkstra method. Dijkstra method is used as a search for the shortest route. This study focused on the desktop platform.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Al – Qur'an adalah rangkuman firman Allah yang diturunkan sebagai rahmat untuk seluruh alam melalui Nabi Muhammad SAW. Membaca Al-Qur'an merupakan tuntunan ibadah kepada Allah SWT sekaligus wahana memahami Al-Qur'an sebagai pedoman dalam mengarungi kehidupan. Perintah membaca Al – Qur'an sendiri telah jelas secara gamblang tertuang dalam surat Al-Alaq ayat 1-5:

Artinya :

Bacalah dengan (menyebut) nama Tuhanmu Yang menciptakan, Dia telah menciptakan manusia dari segumpal darah, Bacalah, dan Tuhanmulah Yang Maha Pemurah, Yang mengajar (manusia) dengan perantaraan kalam, Dia mengajarkan kepada manusia apa yang tidak diketahuinya.

Dalam memahami al-Qur'an, setiap umat islam wajib mengetahui huruf hijaiah sebagai dasar dalam mempelajari al-Qur'an dan kandungannya. Namun dewasa ini anak-anak semakin kesulitan dalam mempelajari huruf hijaiah ditengah kesibukan mempelajari ilmu umum di sekolah. Padahal al-Qur'an merupakan pedoman utama untuk mengarungi kehidupan di dunia. *Game* mampu mengajarkan banyak keterampilan dan dapat dijadikan sebagai alternaftif menarik

dalam media pembelajaran. Bermain *game* merupakan sebuah terobosan baru dalam bidang pendidikan yang mampu menstimulus perkembangan anak. Hal tersebut melatar belakangi perlunya peningkatan konten *game* sebagai media pembelajaran.

Dari beberapa permasalahan tersebut, peneliti bermaksud menghadirkan salah satu solusi menyenangkan untuk mempermudahkan pengenalan huruf hijaiah ke dalam sebuah *game*. Tujuannya adalah memperkenalkan huruf hijaiah agar anak-anak semakin tertarik mempelajari al-Qur'an. Pada *game* ini, pemain diharuskan mengumpulkan item yang akan memandu pemain untuk mempelajari lebih dalam berkenaan dengan pembelajaran huruf hijaiah. Untuk menjadikan game lebih menantang dan menarik digunakan *artificial inteligence* untuk menjadi motor penggerak NPC. AI yang digunakan adalah Algoritma Dijkstra yang diterapkan sebagai pembangkit perilaku pencarian (*pathfinding*) pada NPC.

1.2 Identifikasi Masalah

1. Bagaimana membangun sebuah *game* 3D bergenre edukasi sebagai media pengenalan huruf hijaiah.
2. Bagaimana mengimplementasikan metode *Dijkstra* sebagai pathfinding pada *NPC*?

1.3 Tujuan Penelitian

1. Membangun sebuah game 3D sebagai media pengenalan huruf hijaiah.
2. Mengimplementasikan metode Dijkstra sebagai pathfinding *NPC* (*Non Player Character*) dalam menemukan lokasi pemain.

1.4 Batasan Masalah

Batasan masalah dari penelitian ini adalah:

a. Game dibangun menggunakan Unity 5 Personal (*Free*)
b. *Game* dimainkan oleh *Single Player*.
c. Berjenis edukasi dibangun untuk platform windows pada *Personal Computer*

1.5 Manfaat Penelitian

Sebagai game edukatif untuk membantu anak belajar huruf hijaiah secara menyenangkan dan tidak membosankan.

BAB II
KAJIAN PUSTAKA

2.1 Landasan Teori

2.2.1 Permainan

Game atau permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius dengan tujuan *refreshing*. Dewasa ini bermain game telah menjadi gaya hidup masyarakat modern. Game merupakan hiburan berbagai usia yang mampu membawa nuansa menyenangkan (Anggara, 2008)

Teori permainan pertama kali dikemukakan oleh sekelompok ahli Matematika pada tahun 1944. John von Neumann and Oskar Morgenstern menyatakan bahwa permainan terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri atau pun untuk meminimalkan kemenangan lawan (Neumann dkk, 1953).

Berdasarkan pendapat seorang ahli, teori permainan adalah bagian dari ilmu pengetahuan yang berkaitan dengan pembuatan keputusan pada saat ada dua pihak atau lebih berada dalam kondisi bersaing. Pihak-pihak yang bersaing ini diasumsikan bersifat rasional dan cerdas. Hal ini diartikan bahwa masing-masing pihak akan melakukan strategi rasional demi tujuan memenangkan persaingan. Masing-masing pihak juga berusaha mengetahui strategi lawan. Selanjutnya pihak ini disebut pemain. (Dimyati 1992).

Teori permainan merupakan teori yang menggunakan pendekatan matematis dalam merumuskan situasi persaingan dan konflik antara berbagai kepentingan. Teori ini dikembangkan untuk menganalisa proses pengambilan keputusan yaitu strategi optimum dari situasi-situasi persaingan yang berbeda-beda dan melibatkan dua atau lebih kepentingan (Kartono 1994).

Tujuan teori ini adalah menganalisis proses pengambilan keputusan dari persaingan yang melibatkan dua atau lebih pemain. Kegunaan dari teori permainan adalah metodologi yang disediakan untuk menstruktur dan menganalisis masalah pemilihan strategi. Menggunakan teori permainan, maka langkah pertama adalah menentukan secara eksplisit pemain, strategi yang ada, dan juga menentukan preferensi serta reaksi dari setiap pemain.

Terdapat dua jenis strategi permainan yang dapat digunakan, yakni *pure strategy* dimana setiap pemain mempergunakan strategi tunggal dan *mixed strategy*. *Mixed staretegy* memiliki ciri setiap pemain menggunakan strategi campuran yang berbeda. *Pure strategy* digunakan untuk jenis permainan yang hasil optimalnya mempunyai *saddle point* (titik keseimbangan antara nilai permainan kedua pemain). Sedangkan *mixed strategy* digunakan untuk mencari solusi optimal dari kasus yang tidak memiliki *saddle point*.

2.1.2 Unsur-unsur Dasar Teori Permainan

Ada beberapa unsur dasar yang sangat penting dalam penyelesaian setiap kasus dengan teori permainan. Berikut penjelassan detilnya :

a) Jumlah Pemain

Permainan diklasifikasikan menurut jumlah tujuan yang ada dalam permainan tersebut. Dalam hal ini pengertian "jumlah pemain" tidak selalu sama artinya dengan "jumlah orang" yang terlibat dalam permainan. Jumlah pemain berarti jumlah kelompok pemain berdasarkan masing-masing kepentingan atau tujuannya. Dengan demikian dua orang atau lebih yang mempunyai kepentingan yang sama dapat diperhitungkan sebagai satu kelompok pemain.

b) Ganjaran / Pay-off

Ganjaran adalah hasil akhir yang terjadi pada akhir permainan. Berkenaan dengan ganjaran ini, permainan digolongkan menjadi 2 macam kategori, yaitu permainan jumlah nol (*zero-sum games*) dan permainan jumlah bukan nol (*non-zero-sum games*). Permainan jumlah nol terjadi jika jumlah ganjaran dari seluruh pemain adalah nol. Ganjaran dinilai dengan memperhitungkan setiap keuntungan sebagai bilangan positif dan setiap kerugian sebagai bilangan negatif.

Permainan jumlah bukan-nol setiap kemenangan bagi suatu pihak pemain merupakan kekalahan bagi pihak pemain lain. Kedua kategori permainan berdasarkan ganjaran ini memiliki karakter utama bahwa permainan jumlah nol adalah suatu sistem yang tertutup. Sedangkan permainan jumlah-bukan nol adalah sistem terbuka. Mayoritas permainan pada dasarnya merupakan permainan jumlah nol. Berbagai situasi dapat dianalisis sebagai permainan jumlah nol.

c) Strategi Permainan

Strategi permainan adalah suatu rencana tertentu dari seorang pemain, sebagai reaksi atas aksi yang mungkin dilakukan oleh pemain yang menjadi saingannya. Permainan diklasifikasikan menurut jumlah strategi yang tersedia bagi masing-masing pemain. Jika pemain pertama memiliki **m** kemungkinan strategi dan pemain kedua memiliki **n** kemungkinan strategi, maka permainan tersebut dinamakan permainan **m x n**.

Perbedaan jenis permainan berdasarkan jumlah strategi ini adalah bahwa permainan dibedakan menjadi permainan berhingga dan permainan tak berhingga. Permainan berhingga terjadi apabila jumlah terbesar dari strategi yang dimiliki oleh setiap pemain berhingga atau tertentu, sedang permainan tak berhingga terjadi jika setidak-tidaknya seorang pemain memiliki jumlah strategi yang tak berhingga.

d) Matriks Permainan

Setiap permainan yang dianalisis dengan teori permainan selalu dapat disajikan dalam bentuk sebuah matriks permainan. Matriks permainan disebut juga matriks ganjaran yaitu sebuah matriks yang semua unsur berupa ganjaran dari para pemain yang terlibat dalam permainan tersebut. Baris-barisnya melambangkan strategi –strategi yang dimiliki pemain pertama. Kolom-kolomnya melambangkan strategi-strategi yang dimiliki pemain lain.

Dengan demikian, permainan berstrategi **m x n** dilambangkan dengan matriks permainan **m x n**. Teori permainan berasumsi bahwa strategi yang tersedia bagi masing-masing pemain dapat dihitung dan ganjaran yang berkaitan dengannya dapat dinyatakan dalam unit, meski tidak selalu harus dalam unit moneter. Hal ini penting bagi penyelesaian permainan, sebagai media menentukan pilihan strategi yang akan dijalankan oleh masing-masing pemain. Dengan asumsi bahwa masing masing pemain berusaha memaksimalkan keuntungannya atau meminimalkan kerugiannya. Nilai dari suatu permainan adalah ganjaran yang diharapkan dari sepanjang rangkaian permainan. Kedua pemain selalu berusaha memainkan strateginya yang optimum.

Secara konvensional, nilai permainan dilihat dari pihak pemain yang strategi dilambangkan oleh baris-baris matriks ganjaran, dengan kata lain dilihat dari sudut pandang pemain tertentu. Pemain dikatakan adil (fair) apabila nilainya nol, dimana takseorang pemain pun yang memperoleh keuntungan atau kemenangan dalam permainan yang tidak adil (unfair). Seorang pemain akan memperoleh kemenangan atas pemain lain, yaitu jika nilai permainan tersebut bukan nol, dalam hal ini nilai pemain adalah positif jika pemain pertama memperoleh kemenangan, sebaliknya nilai permainan negatif jika pemain lain memperoleh kemenangan.

e) Titik Pelana (*Saddle Poin*)

Titik pelana adalah suatu unsur di dalam matriks permainan yang sekaligus sebagai maksimin baris dan minimaks kolom. Permainan dikatakan bersaing ketat (*Strictly determined*) jika matriksnya memiliki titik pelana. Strategi yang optimum bagi masing-masing pemain adalah strategi pada baris dan kolom yang mengandung titik pelana tersebut. Dalam hal ini baris yang mengandung titik pelana merupakan strategi optimum bagi pemain pertama, sedangkan kolom yang mengandung titik pelana merupakan strategi optimum bagi pemain lain.

Langkah pertama penyelesaian sebuah matriks permainan adalah memeriksa ada atau tidaknya titik pelana. Bila terdapat titik pelana permainan dapat segera dianalisis untuk diselesaikan. Untuk menentukan titik pelana biasanya dilakukan dengan menuliskan nilai-nilai minimum dan maksimum masing-masing kolom, kemudian menentukan maksimun diantara minimum baris dan minimum diantara maksimum kolom. Jika unsur maksimum dari minimum baris sama dengan unsur minimum dari maksimum kolom, atau jika maksimin = minimaks, berarti unsur tersebut merupakan titik pelana.

Teori permainan dapat diterapkan dalam berbagai bidang, meliputi teknologi, kemiliteran, bisnis, social, ekonomi dan ekologi. Sebagai contoh pada dunia bisnis, seorang direktur suatu perusahaan didalam memperkenalkan sebuah produk baru berusaha mengetahui kemungkinan strategi paling baik atau suatu kombinasi strategi untuk merebut pasar yang lebih besar. Sementara saingannya juga mencoba meperkenalkan produk sejenis dengan strategi yang berbeda dengan direktur pemasaran tersebut.

Berdasarkan teori Dimyati (2006), di sinilah peranan teori permainan untuk menentukan strategi mana yang akan diputuskan oleh direktur pemasaran tersebut demi kepentingan merebut pasar. Persaingan yang di contohkan di atas dapat diidentifikasi untuk menjelaskan konsep teori permainan yang terdiri dari beberapa unsur-unsur dasar, yaitu:

1. Angka-angka dalam matriks *pay-off*, atau biasa disebut matriks permainan, menunjukkan hasil-hasil dari strategi–strategi permainan yang berbeda-beda, hasil-hasil ini dinyatakan dalam suatu bentuk ukuran efektifitas seperti uang, persentase *market share*, atau utilitas.

2. *Maximizing player* adalah pemain yang berada di baris dan yang memenangkan/memperoleh keuntungan permainan, sedangkan minimizing player adalah pemain yang berada di kolom dan yang menderita kekalahan / kerugian.

3. Strategi permainan adalah rangkaian kegiatan atau rencana yang menyeluruh dari seorang pemain, sebagai reaksi atas perilaku pesaingnya. Dalam hal ini, strategi atau rencana tidak dapat dirusak oleh pesaing lainya.

4. Aturan-aturan permainan adalah pola dimana para pemain memilih strategi.

5. Nilai permainan adalah hasil pay-off yang diperkirakan oleh pemain sepanjang rangkaian permainan dimana masing-masing pemain menggunakan strategi terbaiknya. Permainan dikatakan adil apabila nilai permainan sama dengan nol dan sebaliknya.

6. Dominan adalah kondisi dimana pemain dengan setiap pay-offnya dalam strategi superior terhadap setiap pay-off yang berhubungan dalam suatu strategi alternative. Aturan dominan digunakan untuk mengurangi ukuran matriks pay-off dan upaya perhitungan.

7. Strategi optimal adalah kondisi dimana dalam rangkaian kegiatan permainan seorang pemain berada dalam posisi yang paling menguntungkan tanpa menghiraukan kondisi pesaingnya.

8. Tujuan dari model adalah mengidentifikasi strategi atau rencana optimal untuk setiap pemain.

2.1.3 Jenis-jenis Permainan :

Game dikategorikan dalam beberapa genre. *Andrew Rollings and Ernest Adams* menjelaskan kategori *game* sebagai berikut (Rolling dkk, 2003):

a. *Action Game*

Game yang mengutamakan gerak. Permainan jenis ini membutuhkan ketangkasan/ respon yang cepat dari pemain. Permainan aksi (Action games) mengajak pemain untuk menggunakan refleks, akurasi serta waktu yang tepat untuk menyelesaikan sebuah tantangan. Merupakan genre dasar dari sebuah permainan dan digunakan di hampir semua permainan yang ada. Dalam permainan aksi, biasanya terdapat pertempuran. Terdapat banyak sub-genre dari permainan aksi, seperti *Fighting games* dan *First-person shooter*.

Gambar 2.1 GTA contoh *action game*

Sumber : www.techtudo.com.br

b. Strategy Game

Asal-usul *genre* ini berasal dari *board games* biasanya memiliki banyak aturan yang merangsang strategi dan perencanaan. Real-time strategy (RTS), yaitu genre permainan video yang mengandung unsur strategi. Permainan video RTS memerlukan kecakapan pemain untuk memimpin suatu kelompok, mengelola sumber daya, dan melancarkan serangan untuk memperluas wilayah kekuasaan. Contoh permainan video jenis ini adalah serial *Age of Empires*, *Civilization*, *Rise of Nations*, dan *Warcraft*. *Turn-based strategy* (TBS), yaitu jenis permainan video strategi yang pemainnya saling menunggu giliran untuk bergerak.

Gambar 2.2 Contoh *strategy game*

Sumber : www.svisati.com

c. Role-Playing Game

Permainan ini biasanya memiliki ciri khas dengan memiliki alur cerita yang kuat dan meningkatkan pengalaman pemain. Permainan peran (Role-playing games), merupakan permainan video yang menempatkan pemain sebagai tokoh dalam permainan tersebut untuk memecahkan suatu misteri dengan menyelesaikan berbagai macam *puzzle* dan *quest*. Jenis ini biasanya memiliki alur cerita yang kompleks. Contoh permainan video dalam ragam ini adalah serial *Final Fantasy*, *Star Ocean*, dan *Suikoden*.

Gambar 2.3 Final Fantasy

Sumber : lusipurr.com

d. Sport Game

Pertandingan olahraga seperti di dunia nyata yang dibawa ke dalam sebuah *game*. Permainan olahraga (Sports games), yaitu jenis permainan video yang

menuntut keterampilan pemain untuk melakukan pertandingan olahraga secara virtual, seperti pertandingan sepak bola, basket, dan sebagainya. Contoh permainan video dalam ragam ini adalah serial *Pro Evolution Soccer* dan *Madden NFL*.

Gambar 2.4 *Football Game*

Sumber : www.psu.com

e. Vehicle Simulation Game

Game ini mencoba menciptakan bagaimana rasa mengemudi atau menerbangkan suatu kendaraan sebagai media simulasi. Game simulasi kendaraan adalah genre video game yang mencoba untuk memberikan pengalaman pemain dengan interpretasi realistis operasi berbagai jenis kendaraan. Kendaraan yang dimaksud termasuk mobil, pesawat, perahu, pesawat ruang angkasa, kendaraan militer dan berbagai kendaraan lain. Tantangan utama game ini adalah untuk menguasai dan mengemudi

kendaraan dari perspektif pilot atau driver. Kebanyakan game menambahkan tantangan lain seperti balap atau kendaraan tempur saingan. Game sering ditambahkan pula efek *real*, dengan beberapa game termasuk fisika dan tantangan seperti manajemen bahan bakar yang lebih realistis .

Gambar 2.5 Simulasi Pesawat Terbang

Sumber : en.wikipedia.org

f. Construction and Management Simulation Game

Jenis *game* ini bertujuan untuk membuat sesuatu dalam konteks proses yang berkelanjutan. Semakin baik pula hasil yang diperoleh. Konstruksi dan simulasi manajemen (CMS) adalah jenis simulasi permainan di mana pemain membangun, memperluas atau mengelola komunitas fiksi atau proyek dengan sumber daya yang terbatas. Strategi video game kadang-kadang

menggabungkan aspek CMS ke dalam perekonomian. Pemain harus mengelola sumber daya sekaligus memperluas proyek mereka. Permainan CMS murni berbeda dari game strategi. Tujuan pemain adalah tidak untuk mengalahkan musuh, tetapi untuk membangun sesuatu dalam konteks yang berkelanjutan. Permainan dalam kategori ini kadang-kadang juga disebut "permainan manajemen". SimCity merupakan contoh awal dari keberhasilan besar genre game ini.

Gambar 2.6 Contoh CMS Game

Sumber : macgamestore.com

g. Adventure Game

Game petualangan dengan sebuah cerita interaktif tentang karakter yang dikendalikan oleh pemain. Permainan petualangan aksi (Action-adventure

games), yaitu jenis permainan video yang menempatkan pemain sebagai tokoh dalam permainan tersebut, mirip dengan permainan peran, tetapi ditambah dengan unsur-unsur aksi, misalnya perkelahian dan tembak-menembak. Contoh permainan video dalam ragam ini adalah serial *Prince of Persia* dan *Devil May Cry*.

Gambar 2.7 Game Play Prince of Persia

Sumber : newgamenetwork.com

h. Artificial Life

Jenis ini melibatkan proses pemodelan biologis dan seringkali untuk mensimulasikan siklus kehidupan makhluk hidup. Game simulasi kehidupan sekitar, memelihara dan menumbuhkan populasi. Pemain diberikan kekuatan untuk mengontrol kehidupan makhluk otonom atau orang-orang. *Artificial Life* berhubungan dengan penelitian ilmu komputer di kehidupan buatan.

Gambar 2.8 Game Play The Sims

Sumber : sims.wikia.com

i. Puzzle Game

Permainan mengenai pemecahan teka-teki. Permainan jenis ini menarik secara visual dan menyenangkan untuk dimainkan.

Gambar 2.9 Contoh Game Puzzle

Sumber : gametop.com

j. Education Game

Permainan yang mengandung konten pembelajaran terhadap suatu bidang tertentu.

Gambar 2.10 Game Bernuansa Pembelajaran

Sumber : itunes.apple.com

2.1.4 Bahasa Arab

Bahasa Arab (*al-lughah al-'Arabīyyah*) secara ringkas adalah salah satu bahasa Semit Tengah. Bahasa Arab termasuk dalam rumpun bahasa Semit, berkerabat dengan bahasa Ibrani dan bahasa-bahasa Neo Arami. Bahasa Arab memiliki lebih banyak penutur daripada bahasa-bahasa lainnya dalam rumpun bahasa Semit. Dituturkan oleh lebih dari 280 juta orang sebagai bahasa pertama, penutur sebagian besar tinggal di Timur Tengah dan Afrika Utara.

Bahasa ini adalah bahasa resmi dari 25 negara. Merupakan bahasa peribadatan Islam sekaligus merupakan bahasa yang dipakai dalam Al-Qur'an. Berdasarkan penyebaran geografisnya, percakapan bahasa Arab memiliki banyak variasi (dialek). Bahasa Arab modern telah diklasifikasikan sebagai satu makrobahasa dengan 27 sub-bahasa dalam ISO 639-3. Bahasa Arab Baku (kadang-kadang disebut Bahasa Arab Sastra) diajarkan secara luas di sekolah dan universitas, serta digunakan di tempat kerja, pemerintahan, dan media massa.

Bahasa Arab Baku berasal dari Bahasa Arab Klasik, satu-satunya anggota rumpun bahasa Arab Utara Kuna yang saat ini masih digunakan, sebagaimana terlihat dalam inskripsi peninggalan Arab pra-Islam yang berasal dari abad ke-4. Bahasa Arab Klasik juga telah menjadi bahasa kesusasteraan dan bahasa peribadatan Islam sejak lebih kurang abad ke-6. Abjad Arab ditulis dari kanan ke kiri.

Bahasa Arab telah memberi banyak kosakata kepada bahasa lain dari dunia Islam, sama seperti peranan Latin kepada kebanyakan bahasa Eropa.

Semasa Abad Pertengahan bahasa Arab juga merupakan alat utama budaya, terutamanya dalam sains, matematik adan filsafah, yang menyebabkan banyak bahasa Eropa turut meminjam banyak kosakata dari bahasa Arab.

Seperti dengan bahasa Eropa lain, banyak kata-kata Inggris diserap dari bahasa Arab, pada umumnya melalui bahasa Eropa lainnya, terutama dari Spanyol dan Italia, di antaranya adalah kosakata yang digunakan sehari-hari seperti "gula" (sukkar), "kapas" (quṭn) atau "majalah" (makhzen). Kata-kata lain yang sangat terkenal misalnya "aljabar", "alkohol" dan "zenith".

Pengaruh Arab telah menjadi paling mendalam pada negara-negara yang dikuasai oleh islam. Arab adalah sumber kosa kata utama untuk bahasa yang berbagai seperti bahasa Berber, Kurdi, Persia, Swahili, Urdu, Hindi, Turki, Melayu dan Indonesia, baik juga seperti bahasa lain di negara di mana bahasa ini dituturkan. Contohnya perkataan Arab untuk buku /kita:b/ digunakan dalam semua bahasa di atas, kecual pada bahasa Melayu dan Indonesia (di mana ia spesifiknya bermaksud "buku agama").

Tabel 2.1 Huruf Hijaiah

No.	Huruf	Pengucapan	Internasional
1.	ا	Alif	Alif
2.	ب	Ba	bāʾ
3.	ت	Ta	tāʾ

4.	ث	Tsa	ṯāʾ
5.	ج	Jim	ǧīm
6.	ح	Ha	ḥāʾ
7.	خ	Kha	ḫāʾ
8.	د	Dal	dāl
9.	ذ	Dzal	ḏāl
10.	ر	Ra	r āʾ
11.	ز	Zai	z ā y
12.	س	Sin	sīn
13.	ش	Syin	Šīn
13.	ص	Shad	ṣād
15.	ض	Dhad	ḍād
16.	ط	Tha	ṭāʾ
17.	ظ	zha'	ẓāʾ
18.	ع	'ain	'ain
19.	غ	ghain	ġain
20.	ف	Fa	fāʾ

21.	ق	Qaf	qāf
22.	ك	Kaf	kāf
23.	ل	Lam	lām
24.	م	Mim	mīm
25.	ن	Nun	nūn
26.	ه	Ha	hāʾ
27.	و	Wau	wāw
28.	ي	Ya	yāʾ
29.	hamzah	hamzah	hamzah

Sistem ortografi bahasa 'Arab memakai sistem Abjad. Sistem Abjad yaitu sistem tulisan yang huruf-hurufnya melambangkan bunyi Konsonan sedangkan bunyi vokal dilambangkan dengan Harokat.

Huruf Hijaiah terdiri dari 29 huruf Abjad: 26 berupa konsonan murni dan 3 berupa konsonan semi-vokal yaitu huruf "Alif", "Waw" dan "Ya'". Bunyi vokal tidak dilambangkan dengan Abjad tetapi dengan Harokat. Ada 3 harokat dalam bahasa 'Arab: "Fathah" melambangkan bunyi "a" (dan pada beberapa Abjad: bunyi "o"), "Kasrah" melambangkan bunyi "i", dan "Dhammah" melambangkan bunyi "u".

Bahasa Arab juga memiliki penekanan, disebut sebagai tasydid. Penekanan tasydid hanya terjadi di konsonan. Sementara itu, penekanan pada huruf vokal juga terjadi, disebut harakat panjang. Seperti misalnya pada kata KAA-tib (penulis), terjadi penekanan pada huruf vokal, yaitu pemanjangan harakat. Lalu, contoh lainnya yaitu, ma-JAL-LA (majalah), terjadi penekanan pada huruf "La" di mana la merupakan konsonan, dan mendapat penekanan tasydid yakni konsonan L ganda.

2.1.4.1 Tata bahasa

Kosakata bahasa Arab dibagi dalam tiga kelompok, Ism (kata benda), Fi'l (kata kerja), dan Harf (partikel fungsional). Bahasa Arab termasuk bahasa infleksional. Struktur kalimatnya berupa konstruksi topik-komentar atau dikenal juga sebagai Mubtada' wa Khobar. Ada dua macam frasa dalam bahasa Arab, yaitu Jumlatu-l-ismiyyah (frasa nominal) dan Jumlatu-l-fi'liyyah (frasa aktif).

Ada dua macam gender pada Ism dan Fi'l yaitu Mudzakkar (maskulin) dan Mu-annats (feminin). Tiga macam jumlah untuk Ism dan Fi'l yaitu Mufrad (tunggal), Mutsanna (dwi), dan Jama' (jamak). Jumlah jamak terbagi tiga kategori, yaitu Jama' Mudzakkar Salim (jamak biasa maskulin), Jama' Mu-annats Salim (jamak biasa feminin) dan Jama' Taksir (jamak tak beraturan). Khusus untuk Ism ada dua macam artikel, yaitu Ma'ruf (definit) dan Nakirah (non-definit).

Ism ada tiga tingkat peran *Grammatical Case*, yaitu nominatif, akusatif, dan genitif. Ism nominatif berperan sebagai subjek kalimat, Ism akusatif berperan

sebagai obyek (langsung/tidak langsung), Ism genitif berperan sebagai obyek preposisional atau pemilik.

2.1.5 Huruf Hijaiah

Abjad Arab disebut huruf hijaiah, berasal dari aksara Aramaik (dari bahasa Syria dan Nabatea), di mana abjad Arab terlihat kemiripannya dengan abjad Koptik dan Yunani. Dalam bahasa arab, huruf hijaiah dikenal sebagai huruf-huruf yang digunakan dalam pembentukan kata. Huruf hijaiah terdapat berjumlah 29 huruf.

Gambar 2.11 Susunan Huruf Hijaiah

2.1.6 Pathfinding

P*athfinding* merupakan algoritma yang dimanfaatkan sebagai metode untuk menemukan rute terpendek. *Pathfinding* digunakan untuk menentukan arah pergerakan suatu objek dari satu tempat ke tempat lain berdasarkan keadaan peta dan objek lainnya. Dalam pemecahan *pathfinding* akan dibutuhkan algoritma yang dapat dengan cepat memproses dan menghasilkan solusi paling optimal untuk

mencapai suatu lokasi target. Algoritma yang digunakan untuk *pathfinding* sudah banyak yang ditemukan. Misalnya Bellman–Ford, Dijkstra, Floyd–Warshall, dan A star.

Pathfinding bisa juga dikatakan proses penentuan rute terpendek antara dua titik. Varian yang lebih praktis dari pemecahan labirin. Bidang penelitian ini biasanya didasarkan pada algoritma Dijkstra sebagai penentu jalan terpendek pada graf berbobot. Pada intinya, pathfinding adalah metode penemuan rute pada sebuah graf dengan memulai pada satu titik dan menjelajahi node yang berdekatan sampai node tujuan tercapai. Umumnya dengan maksud untuk menemukan rute terpendek dan tercepat.

Penemuan rute memiliki beragam metode, ada yang memanfaatkan pencarian luas, kemudian menentukan rute terdekat dan cenderung memakan waktu lama. Metode lain yang lebih cepat adalah, "menjelajah". Sebagaimana analogi orang yang berjalan pada suatu ruang; daripada memeriksa setiap kemungkinan rute di awal, orang umumnya akan berjalan ke arah tujuan dan hanya menyimpang dari jalan untuk menghindari halangan.

Dua masalah utama pada pathfinding adalah

(1) Menemukan jalan antara dua node dalam graf;

(2) Menentukan lintasan terpendek yang optimal.

Algoritma dasar bekerja seperti pencarian luas dengan memeriksa semua kemungkinan jalur. Mulai dari node awal yang diinisiasi, kemudian diiterasi ke semua jalur potensial sampai mencapai node tujuan. Masalah lebih rumit adalah menemukan jalur yang optimal. Pendekatan lengkap mengenai kasus tersebut kemudian dikenal sebagai algoritma Bellman-Ford. Algoritma tersebut menghasilkan waktu yang lama.

Namun, tidak perlu untuk memeriksa semua jalur yang mungkin untuk menemukan satu yang optimal. Algoritma seperti A * dan algoritma Dijkstra adalah contoh algoritma paling strategis dalam penemuan jalur, baik melalui fungsi heuristik maupun melalui pemrograman dinamis. Kedua algoritma tersebut bekerja dengan menghilangkan jalur yang dianggap mustahil. Algoritma ini tervalidasi mencapai kompleksitas waktu lebih rendah dibandingkan Bellman-Ford.

A star (A*) dan djikstra adalah algoritma yang paling sering digunakan dalam hal *pathfinding* pada sebuah *game*, Algoritma A* ditemukan oleh Peter Hart, Nils Nilsson, dan Bertram Raphael di 1968 dan terus dikembangkan hingga kini. A* sendiri menggunakan *best first search* dan bobot minimal yang diberikan dari simpul awal ke simpul tujuan. Sedangkan djikstra adalah sebuah algoritma yang dikembangkan oleh Edsger Dijkstra, seorang ilmuwan komputer dari Belanda.

Willy Setiawan (2010) mengungkapkan bahwa A* menggunakan jarak ditambah biaya yang dinyatakan dengan f(x) dari fungsi heuristik untuk menentukan urutan pencarian mencapai simpul mana pada sebuah pohon,

sedangkan algoritma Djikstra adalah sebuah algoritma yang menyelesaikan pencarian jalur terpendek pada graf dengan nilai non negatif untuk bobot setiap simpul, menghasilkan pohon jalur terpendek.

2.1.7 Algoritma Dijkstra

Sudah banyak algoritma yang bisa digunakan untuk tujuan pencarian rute terpendek. Tidak bisa di pungkiri Djikstra masih menjadi salah satu yang populer dari sekian banyak algoritma tersebut (Setiawan, 2015).

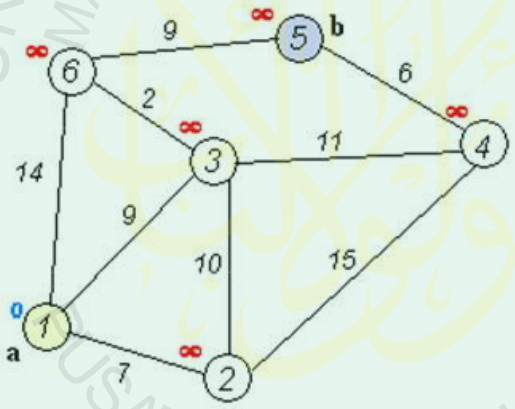
Algoritma Dijkstra, dinamakan sesuai dengan nama penemunya, seorang ilmuwan komputer berkebangsaan Belanda yang bernama Edsger Dijkstra, adalah algoritma yang digunakan untuk mencari lintasan terpendek pada sebuah graf berarah. Ian Millington (2006) pada bukunya yang berjudul *Artificial Inteligence for Games* menyebut algoritma dijkstra sebagai versi ringkas algoritma a star. Keringkasan yang dimaksud adalah bahwa keduanya memiliki kesamaan dalam proses pencarian rute terdekat, hanya saja pada algoritma Dijkstra fungsi heuristik tidak diperhitungkan sebagaimana yang dilakukan oleh A star.

Cara kerja algoritma Dijkstra memakai stategi greedy, dimana pada setiap langkah dipilih sisi dengan bobot paling kecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih.

Djikstra merupakan salah satu varian bentuk algoritma popular dalam pemecahan persoalan terkait masalah optimasi pencarian lintasan terpendek. Sebuah lintasan yang mempunyai panjang minimum dari verteks a ke z dalam *graph* berbobot. Bobot tersebut adalah bilangan positif jadi tidak dapat dilalui

oleh node negatif. Namun jika terjadi demikian, maka penyelesaian yang diberikan adalah infiniti (Tak Hingga). Pada algoritma Dijkstra, node digunakan karena algoritma Dijkstra menggunakan graph berarah untuk penentuan rute listasan terpendek.

Algoritma Djikstra bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalkan titik mengambarkan gedung dan garis menggambarkan jalan, maka algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik.

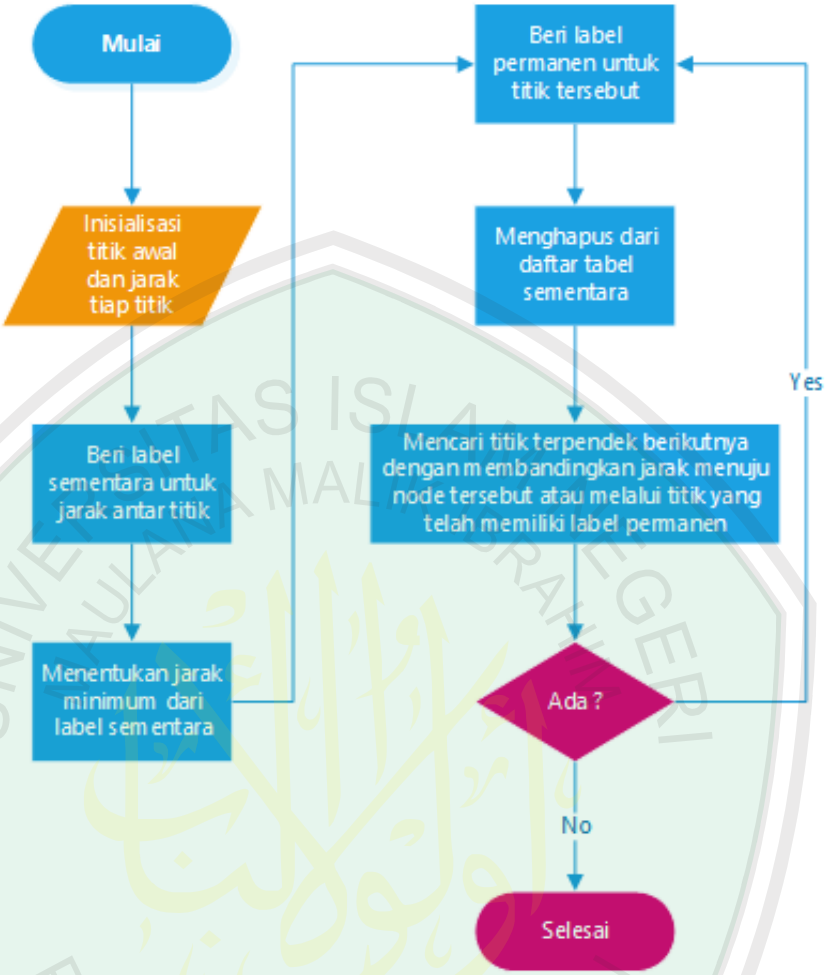


Gambar 2.12 Contoh Ketersambungan Titik dalam Algoritma Dijkstra

Pertama-tama tentukan titik mana yang akan menjadi node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu per satu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap. Inilah urutan logika dari algoritma Dijkstra (Setiawan, 2015):

1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi)
2. Set semua node “Belum terjamah” dan set node awal sebagai “Node keberangkatan”.
3. Dari node keberangkatan, pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke node C berja

   rak 2, maka jarak ke C melewati B menjadi 6+2=8. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
4. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai “Node terjamah”. Node terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. Set “Node belum terjamah” dengan jarak terkecil (dari node keberangkatan) sebagai “Node Keberangkatan” selanjutnya dan lanjutkan dengan kembali ke step 3.



Gambar 2.13 Flowchart Algoritma Dijkstra

Berikut adalah *pseudo-code* algoritma Djikstra

```
function Dijkstra(Graph, source):
     for each vertex v in Graph:                          // Initializations
         dist[v] := infinity ;                            // Unknown distance function from
                                                          // source to v
         previous[v] := undefined ;                       // Previous node in optimal path
     end for                                              // from source

     dist[source] := 0 ;                                  // Distance from source to source
     Q := the set of all nodes in Graph ;                 // All nodes in the graph are
                                                          // unoptimized - thus are in Q
    while Q is not empty:                                 // The main loop
        u := vertex in Q with smallest distance in dist[] ; // Start node in first case
        remove u from Q ;
        if dist[u] = infinity:
            break ;                                       // all remaining vertices are
        end if                                            // inaccessible from source

        for each neighbor v of u:                         // where v has not yet been
                                                          // removed from Q.
            alt := dist[u] + dist_between(u, v) ;
            if alt < dist[v]:                             // Relax (u,v,a)
                dist[v] := alt ;
                previous[v] := u ;
                decrease-key v in Q;                      // Reorder v in the Queue
            end if
        end for
    end while
return dist;
```

Gambar 2.14 Pseudo-code Dijkstra

2.1.8 Agen Musuh (NPC)

Yunifa Miftahul Arif (2010) dalam penelitiannya yang berjudul Strategi Menyerang pada *Game FPS* Menggunakan *Hierarchy Finite State Machine* dan *Logika Fuzzy* menjelaskan bahwa NPC adalah :

> *"Jenis otonomous agent yang ditujukan untuk penggunaan komputer animasi dan media interaktif seperti games dan virtual reality. Agen ini mewakili tokoh dalam cerita atau permainan dan memiliki kemampuan untuk improvisasi tindakan mereka. Ini adalah kebalikan dari seorang tokoh dalam sebuah film animasi, yang tindakannya ditulis di muka, dan untuk "avatar" dalam sebuah permainan atau virtual reality, tindakan yang diarahkan secara real time oleh pemain. Dalam permainan, karakter otonom biasanya disebut Non-Player Character (NPC)".*

Berdasarkan teori yang dikemukakan oleh Brent Ellison (2008), ia menjelaskan bahwa NPC merupakan sebuah karakter non-pemain (NPC), kadang-kadang dikenal sebagai karakter non-orang atau karakter yang tidak dikendalikan oleh pemain. Dalam video game, biasanya berarti karakter yang dikendalikan oleh komputer melalui kecerdasan buatan. Pada *role-playing game* tradisional istilah ini berlaku untuk karakter yang dikendalikan oleh *gamemaster*, bukan pemain lain.

Dalam *role-playing game* tradisional seperti Dungeons & Dragons, NPC adalah karakter fiksi yang diperankan oleh *gamemaster*. Jika karakter pemain berperan sebagai tokoh protagonis cerita, karakter non-player dapat dianggap

sebagai "pemain pendukung" atau "ekstra" dari narasi *Roleplaying*. Karakter non-pemain mengisi dunia fiksi dari permainan, dan dapat mengisi peran apapun yang tidak dikendalikan oleh pemain. Karakter non-pemain mungkin sekutu, para pengamat atau musuh. NPC juga bisa pedagang yang menjual peralatan atau perlengkapan. NPC bervariasi dalam game yang kompleks.

Genre permainan apapun, hampir seluruhnya berinteraksi dengan karakter *nonplayer*, termasuk novel visual seperti Ace Attorney ataupun The sims. Umumnya menampilkan dialog percabangan yang kompleks dan sering menyajikan kemungkinan tanggapan pemain kata demi kata sebagai reaksi atas pilihan yang disetujui. Secara sederhana maka NPC dapat diartikan sebagai karakter yang sepenuhnya dikendalikan komputer dan tidak dapat dimainkan oleh pemain. Pengendalian NPC umumnya menggunakan bidang ilmu kecerdasan buatan.

Kecerdasan buatan dimasukkan dengan tujuan agar NPC dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer, logika fuzzy, jaringan syaraf tiruan dan robotika. Kecerdasan buatan menjadikan NPC memiliki gerak variatif, mampu memainkan perannya sebagai penyempurna permainan dan menjadikan *gameplay* lebih asik dan menantang.

2.1.9 Unity

Unity merupakan perangkat lunak untuk pengembangaan game *multiplatform*. Editor pada Unity dibuat dengan *user interface* yang sederhana dan mudah dipahami oleh *game developer* pemula sekalipun. Grafis pada unity dibuat dengan grafis tingkat tinggi untuk OpenGL dan directX. Unity mendukung semua format file, terutamanya format umum. Unity cocok dengan versi 64-bit. Dapat beroperasi pada Mac OS X dan Windows dan dapat menghasilkan *game* untuk Mac, Windows, Wii, iPhone, iPad maupun Android.

Unity merupakan salah satu *game engine* paling terkemuka dewasa ini. Unity adalah sebuah *software development* yang terintegrasi untuk menciptakan *video game* atau konten lainnya seperti visualisasi arsitektur atau real-time animasi baik yang bernuansa 2D maupun 3D. Unity dapat digunakan pada microsoft Windows dan Mac OS X. Permainan yang dihasilkan dapat dijalankan secara *multiplatform*. Unity juga dapat menghasilkan permainan untuk browser dengan menggunakan *plugin Unity Web Player*.

Unity secara lebih rinci dapat dimanfaatkan untuk pengembangan *3D video game*, *real time* animasi 3D dan visualisasi arsitektur maupun konten interaktif serupa lainnya. Editor Unity dapat menggunakan plugin untuk *web player* dan menghasilkan *game browser* yang didukung oleh Windows maupun Mac. Plugin web player dapat juga dipakai untuk widgets Mac. Unity juga mendukung *console* terbaru seperti PlayStation 3 dan Xbox 360. Tahun 2009 Unity Technology menjadi 5 perusahaan game terbesar di dunia setelah tahun

sebelumnya di tahun 2006, menjadi juara dua pada *Apple Design Awards*. Hingga puncaknya di tahun 2010 Unity berhasil memperoleh *Technology Innovation Award* yang diberikan oleh *Wall Street Journal*.

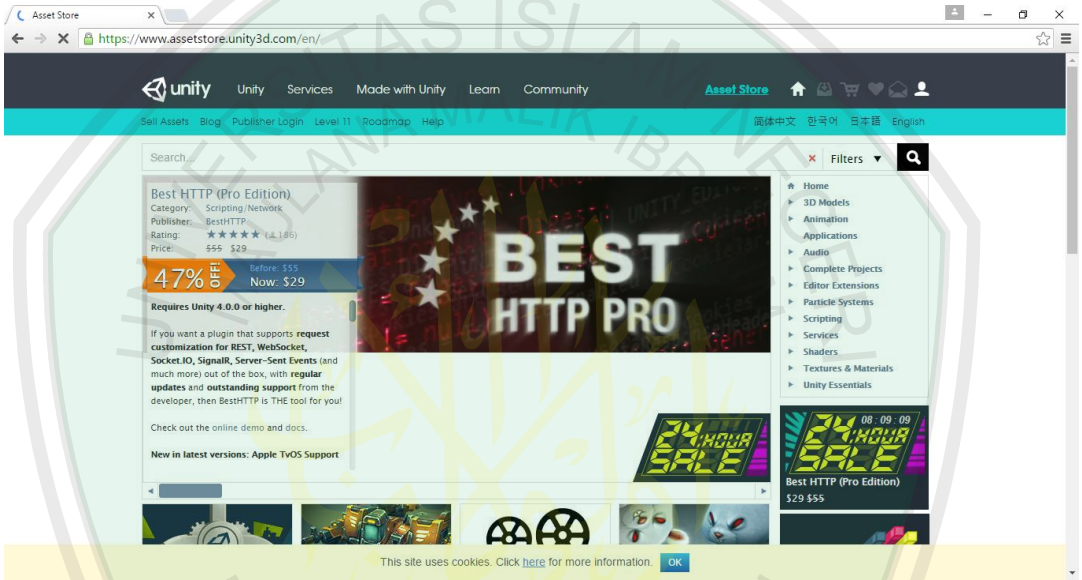
Unity Technology yang merupakan pengembang resmi Unity juga menyediakan *asset store*. *Asset store* terdiri dari berbagai konten, *model*, *prefab*, *script*, *sound* dan kebutuhan lain yang diperlukan dalam pembangunan sebuah permainan. Editor Unity dapat menyimpan metadata. Editor Unity dapat diperbaharui dengan sesegera mungkin seperti file yang telah dimodifikasi. Server aset Unity juga berjalan pada Mac, Windows, Linux dan juga berjalan pada PostgreSQL, database server opensource.

Unity memiliki dua lisensi yakni Unity dan Unity Pro. Versi Unity tersedia dalam bentuk gratis, sedang versi Unity Pro didistribusikan secara berbayar. Versi Unity Pro memiliki berbagai fitur bawaan seperti efek *post processing*, *render*, efek tekstur serta berbagai fitur spesial yang tidak tersedia di versi gratis. Unity dan Unity Pro menyediakan berbagai tutorial, konten, *completed project*, wiki, dukungan melalui forum dan pembaruan ke depannya.

Fitur-fitur Unity

1. Asset Store

Diluncurkan November 2010. Unity Asset Store adalah sebuah resource yang hadir di Unity editor. Asset store terdiri dari koleksi lebih dari 4.400 asset packages, beserta 3D *models, textures, materials, particle system*. Juga dilengkapi dengan musik, efek suara, *tutorial, complete project, scripting package, editor extensions* dan servis online.



Gambar 2.15 Halaman Asset Store

2. Asset Tracking

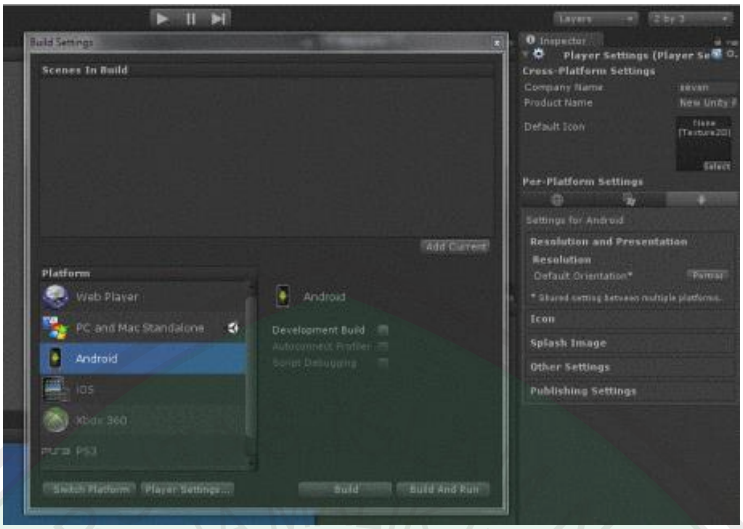
Unity juga menyertakan Server Unity Asset yang merupakan sebuah solusi terkontrol untuk game asset developer dan script. Server tersebut menggunakan PostgreSQL sebagai backend. Sistem audio dibuat menggunakan FMOD library yang memiliki kemampuan untuk memutar Ogg Vorbis compressed audio. Video playback menggunakan Theora codec. Terrain engine dan vegetasi (dimana mensuport tree billboarding, occlusion

culling dengan Umbra), built-in lightmapping dan global illumination dengan Beast. Multiplayer networking menggunakan RakNet dan navigasi mesh sebagai pencari jalur built-in.

3. Platforms

Unity mendukung pengembangan ke berbagai platform. Di dalam project, developer memiliki kontrol untuk mengembangkan permainan ke berbagai perangkat mobile, web browser, desktop, dan console. Unity juga mengijinkan spesifikasi kompresi tekstur dan pengaturan resolusi di setiap platform yang didukung.

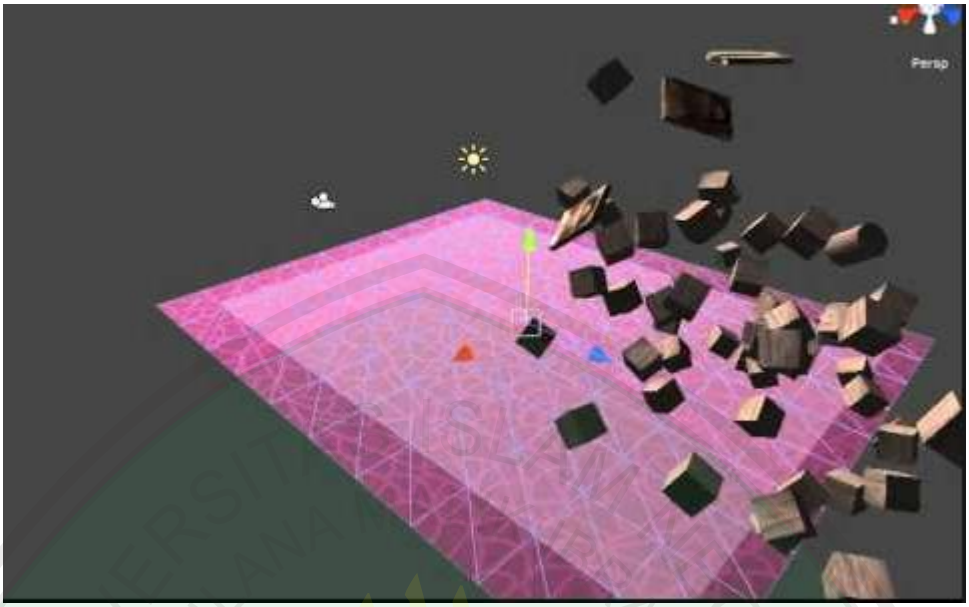
Saat ini platform yang didukung adalah BlackBerry 10, Windows 8, Windows Phone 8, Windows, Mac, Linux, Android, iOS, Unity Web Player, Adobe Flash, PlayStation 3, Xbox 360, Wii U and Wii. Meskipun tidak semua terkonfirmasi secara resmi, Unity juga mendukung PlayStation Vita yang dapat dilihat pada game Escape Plan dan Oddworld: New ‘n’ Tasty. Rencana platform berikutnya adalah PlayStation 4 dan Xbox One. Bahkan Unity telah mendukung HTML dan plug-in Adobe baru dimana akan disubtitusikan ke Flash Player.



Gambar 2.16 Platform yang didukung unity

4. Physics

Unity juga memiliki support built-in untuk PhysX physics engine (sejak Unity 3.0) dari Nvidia (sebelumnya Ageia) dengan penambahan kemampuan untuk simulasi real-time cloth pada arbitrary, skinned meshes, thick ray cast, dan collision layers.



Gambar 2.17 Physics

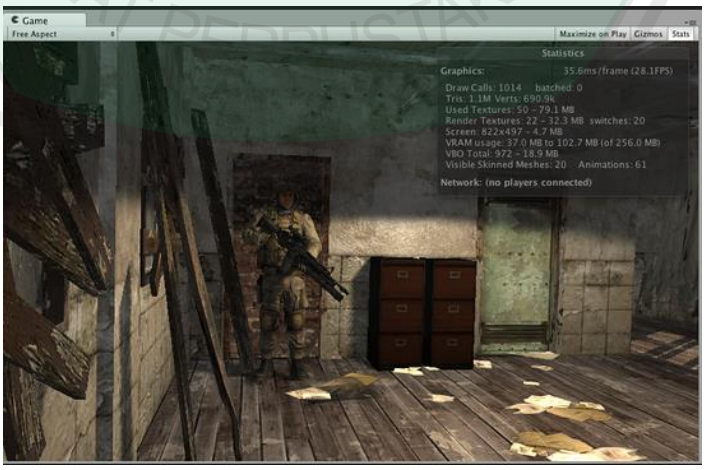
5. Rendering

Graphics engine yang digunakan adalah Direct3D (Windows, Xbox 360), OpenGL (Mac, Windows, Linux, PS3), OpenGL ES (Android, iOS), dan proprietary APIs (Wii). Ada pula kemampuan untuk bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), dynamic shadows using shadow maps, render-to-texture and full-screen post-processing effects.

Unity dapat mengambil format desain dari 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks and Allegorithmic Substance. Asset tersebut dapat ditambahkan ke game project dan diatur melalui graphical user interface Unity.

ShaderLab adalah bahasa yang digunakan untuk shaders, bertujuan agar mampu memberikan deklaratif “programming” dari fixed-function pipeline dan program shader ditulis dalam GLSL atau Cg. Sebuah shader dapat menyertakan banyak varian dan sebuah spesifikasi fallback declarative. Hal tersebut membuat Unity dapat mendeteksi berbagai macam video card terbaik saat ini, dan jika tidak ada yang kompatibel, maka akan dilempar menggunakan shader alternatif yang mungkin dapat menurunkan fitur dan performa.

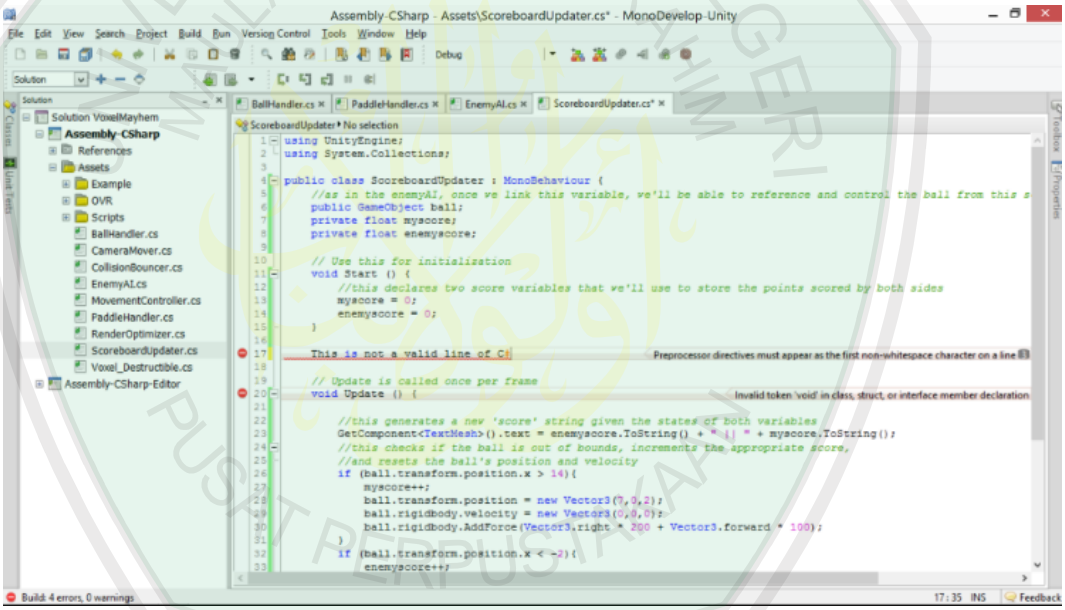
Pada 3 Agustus 2013, seiring dengan diluncurkannya versi 4.2, Unity mengijinkan developer indie mengunakan realtime shadows untuk directional lights dan juga menambahkan kemampuan dari DirectX11 yang memberikan shadows dengan resolusi pixel yang lebih sempurna. Tekstur untuk membuat objek 3D dari grayscale menjadi lebih mudah serta pengembangan animasi yang lebih halus dan mempercepat FPS.



Gambar 2.18 Rendering Statistics

6. Scripting

Script game engine dibuat dengan Mono Develop, sebuah implementasi open-source dari .NET Framework. Programmer dapat menggunakan UnityScript (bahasa terkustomisasi yang terinspirasi dari sintax ECMAScript, dalam bentuk JavaScript), C#, atau Boo (terinspirasi dari sintaks bahasa pemrograman phyton). Dimulai dengan dirilisnya versi 3.0, Unity menyertakan versi MonoDevelop yang terkustomisasi untuk debug script.



Gambar 2.19 Mono Develop

2.2 Penelitian Terkait

2.2.1 *Rancang Bangun Permainan "Werkudara" Menggunakan Dijkstra Pada Agen Musuh*

Penelitian yang memfokuskan dalam dunia *game 3D ini* bernuansa tradisional dan mengangkat cerita Wayang Werkudara sebagai tokoh utama. Penelitian ini telah berhasil diselesaikan oleh Saiful Yahya dkk dari Sekolah Tinggi Informatika dan Komputer Indonesia Malang. Selain mengangkat budaya daerah, penelitian ini juga memanfaatkan algoritma Dijkstra yang diterapkan pada agen musuh.

2.2.2 *Pencarian Rute Terpendek dalam Dunia 3 Dimensi Berdasarkan Algoritma Dijkstra*

Ahmad Hoirul dan Andi Tenriawaru dari Institut Sepuluh Nopember mengembangkan penelitian yang berkaitan dengan dunia tiga dimensi. Pada penelitian tersebut, algoritma Dijkstra dimanfaatkan sebagai metode pencarian rute terpendek.

2.2.3 *A Dijkstra Algorithm For Dynamic Games*

Penelitian ini dikembangkan oleh Martino Bardi dari Italia dan Juan Pablo Maldonado Lopez dari Perancis. Penelitian ini memfokuskan dalam pengimplementasian algoritma dijkstra dalam permainan yang dinamis. Maksud dari permainan yang dinamis adalah pemain bergerak secara simultan. Penelitian ini menggunakan algoritma Dijkstra sebagai *shortest path*

BAB III
DESAIN DAN PERANCANGAN GAME

3.1 Analisis dan Perancangan Game

Permainan ini adalah *game* yang bergenre *education game* dan dimainkan secara *single player*. Pada *game* terdapat karakter sebagai pemain utama yang akan dijalankan oleh pengguna, karakter musuh yang merupakan karakter lawan akan dijalankan secara otomatis oleh komputer dengan menginisiasi algoritma pathfinding *Dijkstra*. Objek penelitian dalam permainan ini adalah algoritma *Dijkstra* yang digunakan NPC untuk mencari keberadaan pemain.

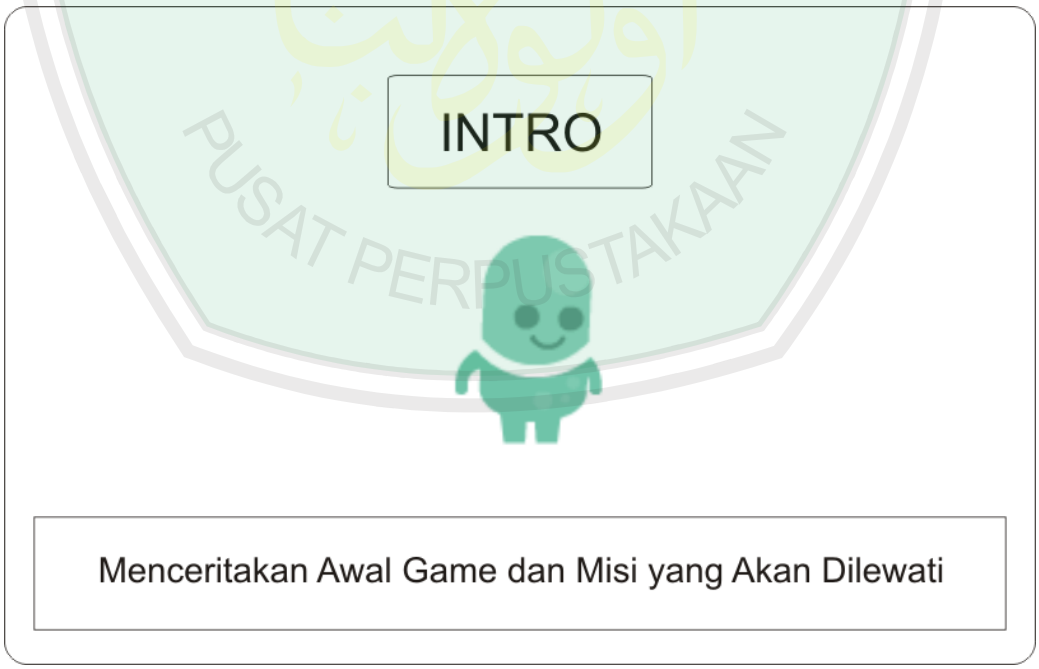
3.1.1 Keterangan Umum Game

Permainan ini bernama “Hai Hijaiyah”. Hai Hijaiyah merupakan permainan edukasi berbasis dekstop yang dijadikan media untuk pengenalan huruf hijaiyah. Sistem kemenangan akan ditentukan dengan penyelesaian misi. Game ini berisi misi-misi tertentu yang akan memandu pemain untuk lebih mengenal wawasan tentang dasar-dasar huruf hijaiah. Ketika berhasil menyelesaikan misi,pemain akan dibaa ke level selanjutnya.

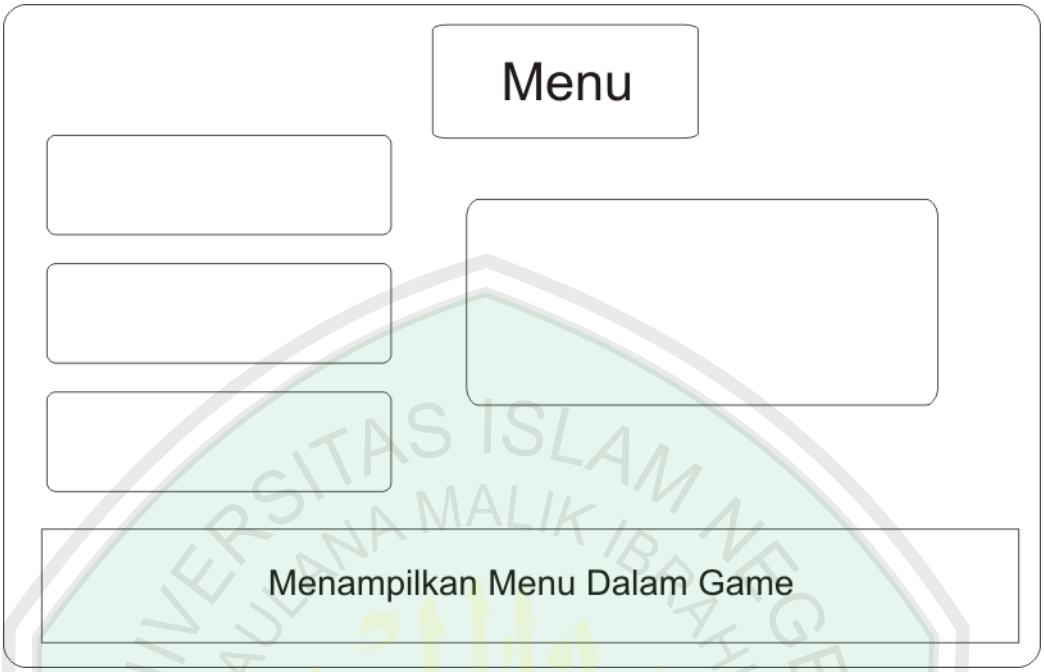
Pada game ini ada terdapat 3 level, dimana setiap level akan mempunyai misi yaitu mengumpulkan huruf-huruf hijaiah. Setiap huruf hijaiah mewakili bonus poin dan memiliki petunjuk yang akan membantu pemain untuk lebih mengenal tentang huruf hijaiah.

Pada *stage* pertama, pemain dibawa ke sebuah ruangan dan diharuskan untuk menemukan item-item huruf hijaiah yang tersebar di seluruh ruangan. Huruf yang menjadi item adalah 10 huruf pertama. Pada *stage* kedua, pemain masih dituntut untuk mengumpulkan item huruf hijaiah yakni 10 huruf hijaiah lanjutan dari huruf di *stage* pertama, hanya saja pada *stage* kedua terdapat halangan yang menjadikan permainan lebih menarik. Di *stage* terakhir atau level final, pemain diharuskan untuk mengu 40 kan item sembari mengalahkan NPC yang bertindak sebagai musuh. NPC inilah yang geraknya ditentukan oleh inisiasi algoritma Dijkstra.

3.1.2 Story Board Permainan

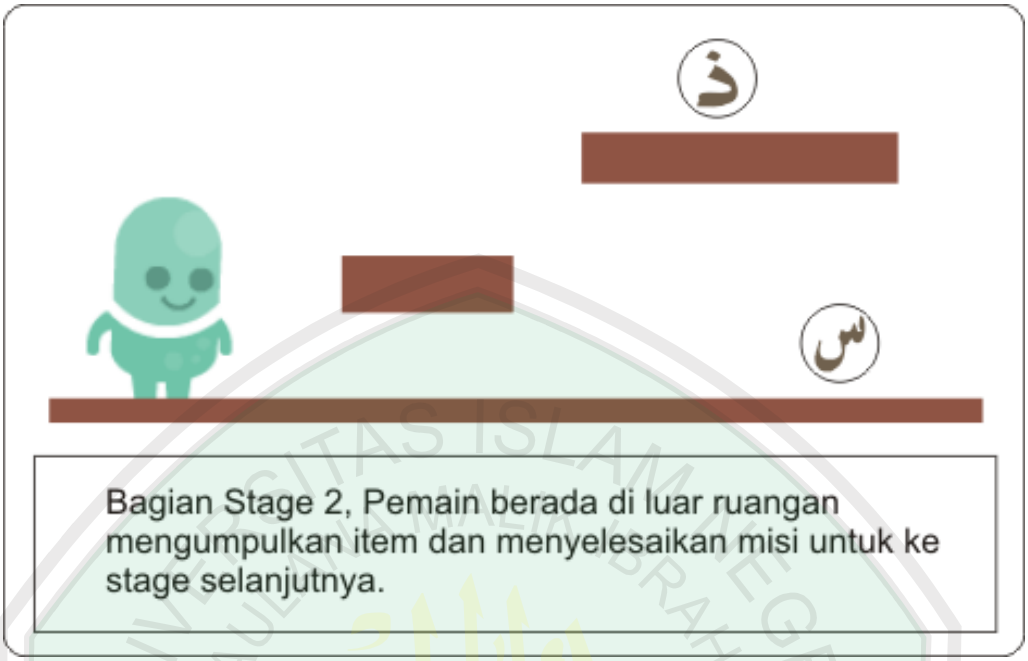


Gambar 3.1 *Story Board* Halaman Intro



Gambar 3.2 *Story Board* Halaman Menu

Gambar 3.3 *Story Board* Stage Pertama



Gambar 3.4 *Story Board* Stage Kedua

Gambar 3.5 *Story Board* Stage Ketiga

Gambar 3.6 *Story Board* Halaman Keterangan Item

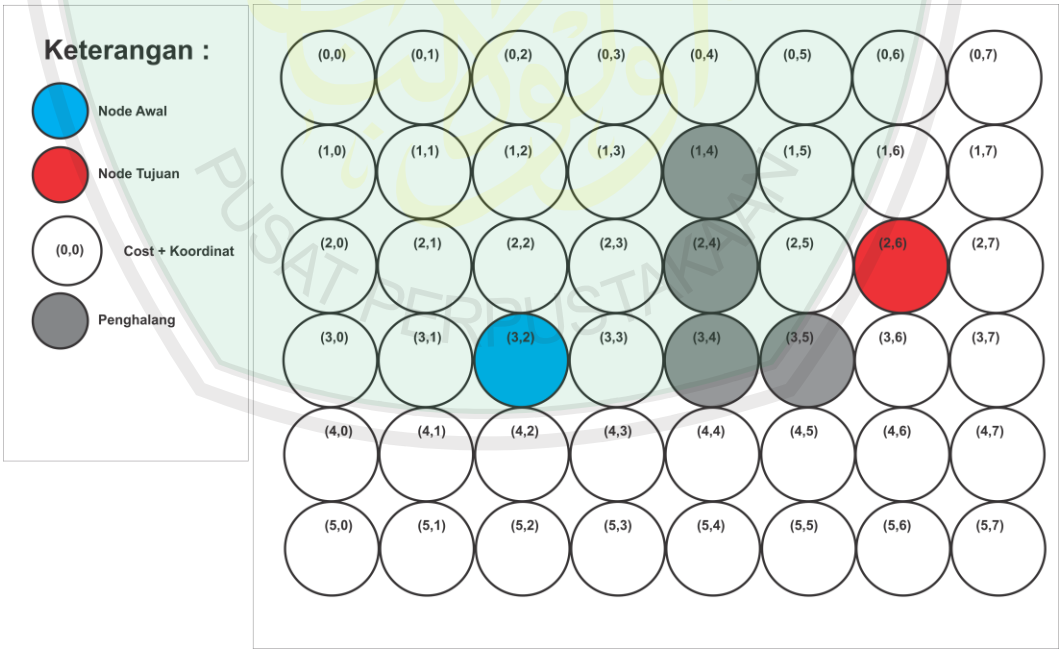
3.1.3 Simulasi Algoritma Dijkstra

Algoritma Dijkstra memiliki sifat membangkitkan simpul tetangga dan mencari *cost* minimum dari setiap simpul/node yang dibangkitkan.

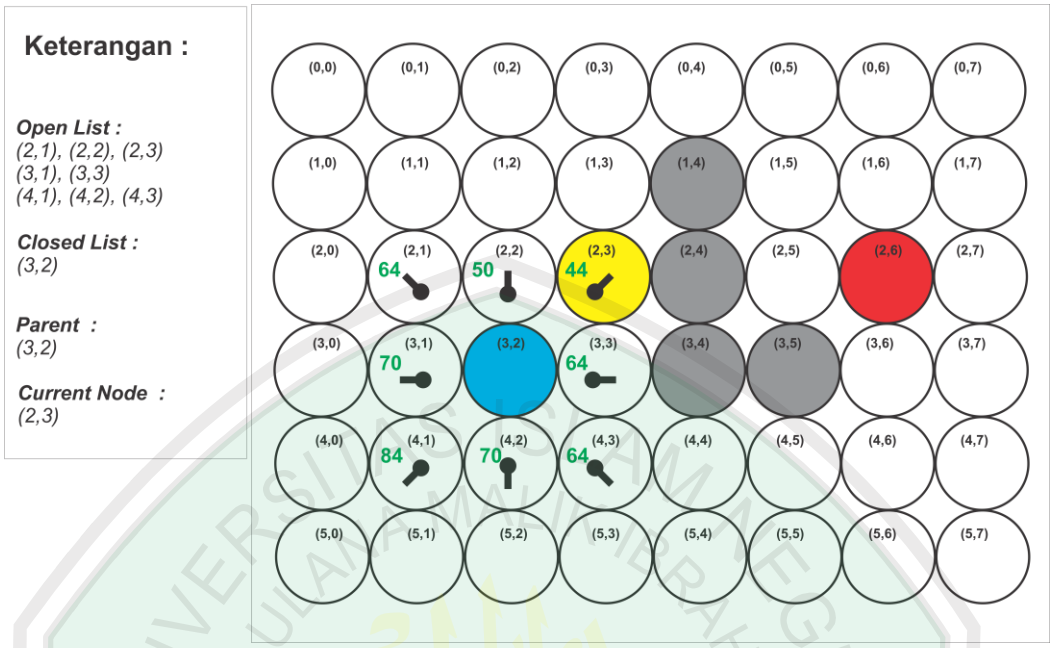
Berikut adalah proses simulasi perhitungan manual langkah dijkstra.

1. Beri bobot untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi)
2. Set semua node sebagai open node “Belum terjamah” dan set node awal sebagai “Node awal”.

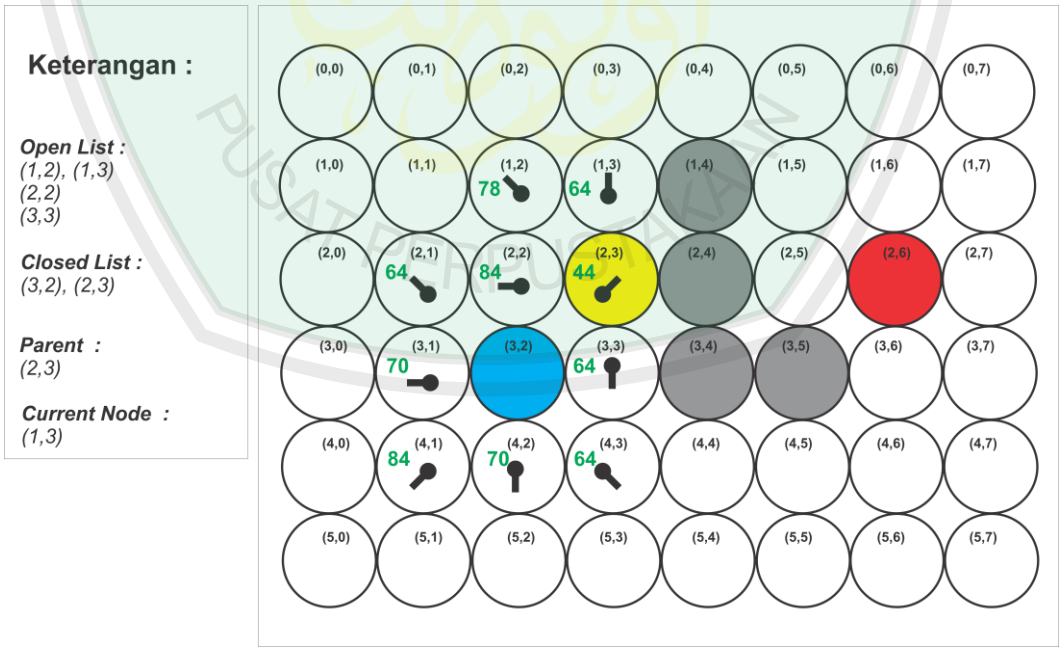
3. Dari node awal, pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
4. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai "Closed Node". Node terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. Set "Open Node" dengan jarak terkecil (dari node awal) sebagai "Node Keberangkatan" selanjutnya dan lanjutkan dengan kembali ke step 3.



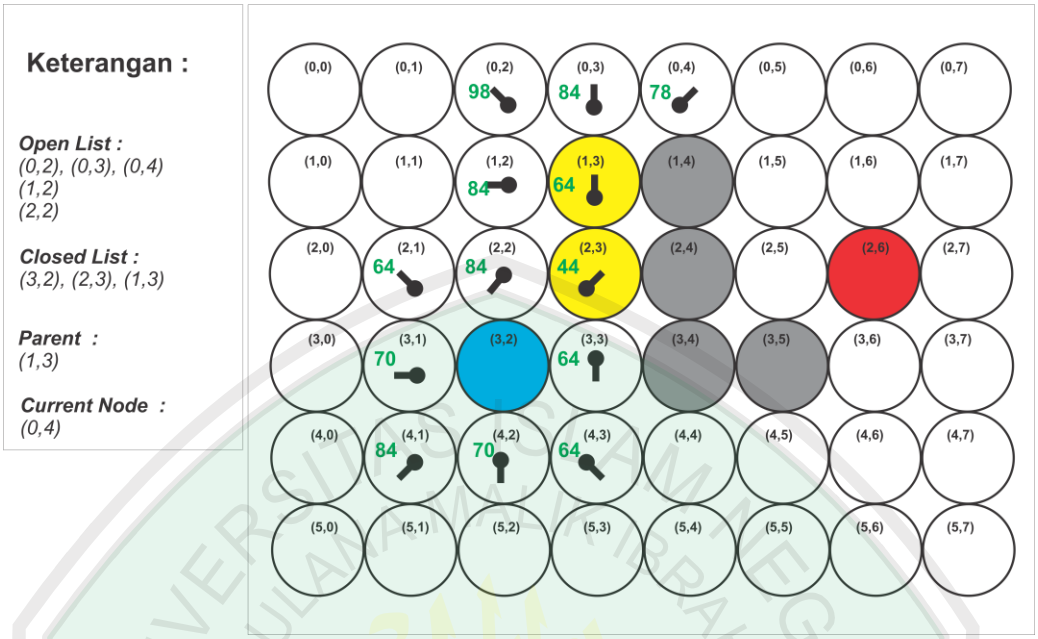
Gambar 3.7 Simulasi Algoritma Dijkstra



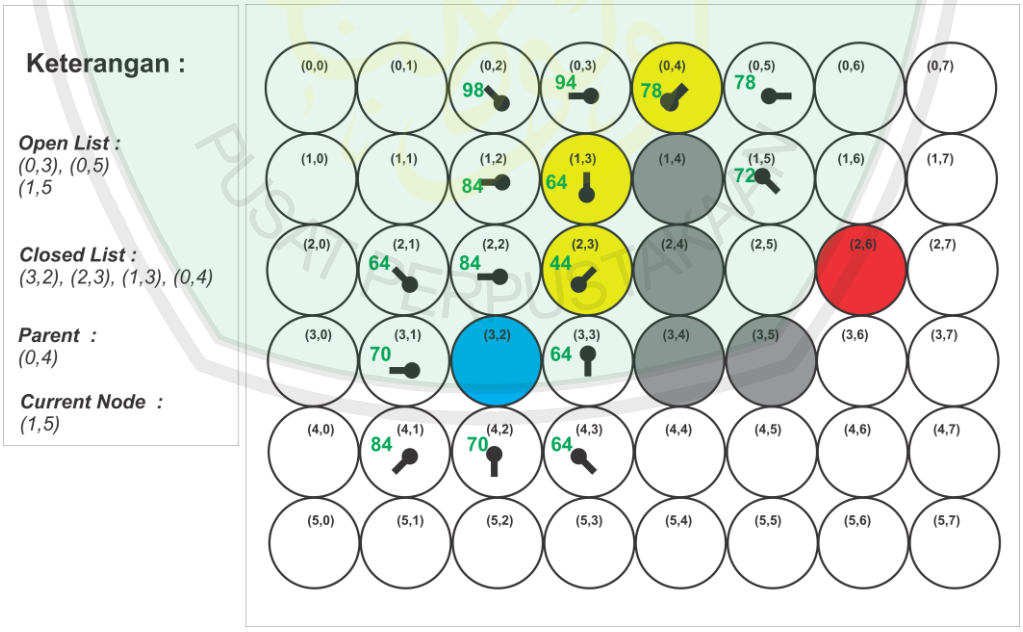
Gambar 3.8 Simulasi Algoritma Dijkstra langkah Pertama



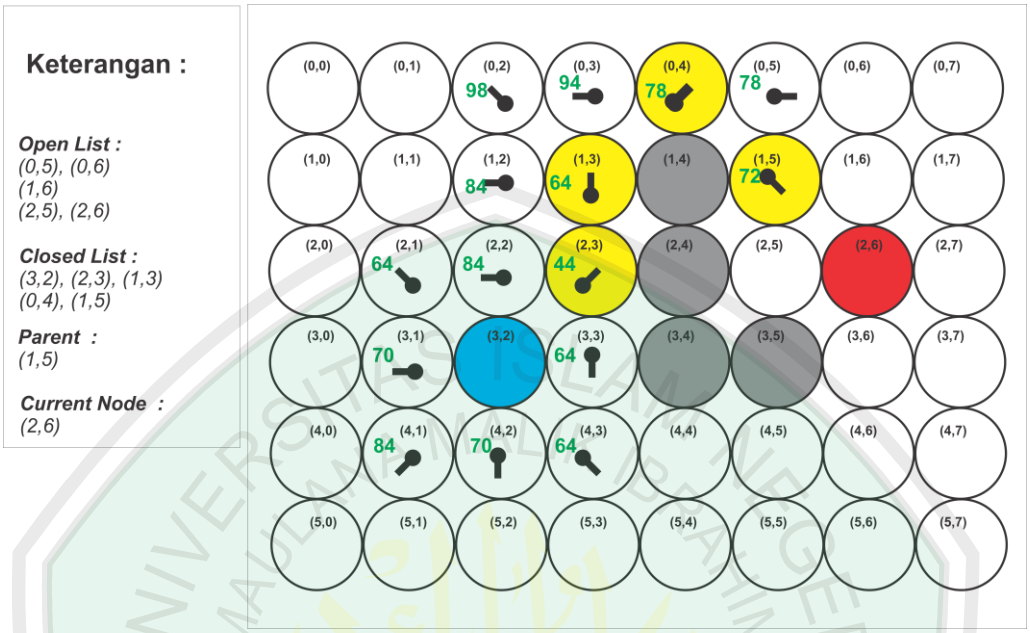
Gambar 3.9 Simulasi Algoritma Dijkstra Langkah Kedua



Gambar 3.10 Simulasi Algoritma Dijkstra Langkah Ketiga



Gambar 3.11 Simulasi Algoritma Dijkstra Langkah Keempat



Gambar 3.12 Simulasi Algoritma Dijkstra Langkah Kelima

BAB IV
HASIL DAN PEMBAHASAN

4.1 Implementasi

Bab ini membahas mengenai hasil implementasi dari perencanaan yang telah dibahas sebelumnya. Selain itu pada bab ini dilakukan pengujian untuk mengetahui apakah *game* dapat berjalan sesuai dengan tujuan penelitian yang ingin dicapai.

4.1.1 Kebutuhan Perangkat Keras

Perangkat keras yang diperlukan untuk mengimplementasikan perangkat lunak dari aplikasi *game* ini, sebagai berikut:

Tabel 4.1 Kebutuhan Perangkat Keras

No.	Perangkat Keras	Spesifikasi
1.	Processor	Intel (R) Celeron (R) 1.0Ghz
2.	RAM	2 GB
3.	VGA	Intel (R) HD Graphics
4.	HDD	500 GB
5.	Monitor	12’
6.	Speaker	On
7.	Mouse & Keyboard	On

4.1.2 Kebutuhan Perangkat Lunak

Perangkat keras yang diperlukan untuk mengimplementasikan perangkat lunak dari aplikasi *game* ini, sebagai berikut:

Tabel 4.2 Kebutuhan Perangkat Lunak

No.	Perangkat Lunak	Spesifikasi
1.	Sistem Operasi	Windows 10 64 Bit
2.	*Game Engine*	Unity3D 5.3.4f1
3.	Konsep desain 2D	Corel Draw X7
4.	Desain 3D	Asset Store
5.	*Script Writer*	Mono Develop

4.2 Implementasi Algoritma Dijkstra pada Agen Musuh

Implementasi adalah proses penerapan komponen sistem utama yang dibangun berdasarkan rancang bangun yang telah diajukan dan ditetapkan sebelumnya. Implementasi kecerdasan buatan pada penelitian ini diterapkan di perilaku pencarian pada NPC dengan memanfaatkan algoritma Dijkstra.

Metode Dijkstra digunakan sebagai algoritma pencarian NPC. Posisi NPC dijadikan sebagai *starting point* dan *player* diinisiasi sebagai tujuan. Perhitungan Dijkstra diupdate secara berkala sehingga proses pencarian berjalan secara dinamis berdasarkan posisi target. Jika target berubah maka NPC akan bergerak secara otomatis mencari posisi target berada.

4.2.1 Algoritma Dijkstra Sebagai Metode Pencarian

Pada bagian ini membahas tentang penerapan metode pencarian *Dijkstra* pada NPC. Algoritma *Dijkstra* pada NPC bekerja pada saat *game* pertama dimulai. NPC memiliki kecerdasan untuk menemukan posisi pemain. Ketika NPC sampai di posisi pemain maka NPC telah menemukan lokasi target. Pemain bertugas memenuhi misi sebagai tugas dan menyelesaikan *game*. Algoritma Dijkstra dalam *game* ini akan diimplementasikan menggunakan bahasa C# dengan *source code* sebagai berikut :

Tabel 4.3 Keterangan Algoritma Dijkstra

No	Method / Fungsi	Keterangan

1.	`public class DijkstraAlgorythm`
	`{`
	`public static Stack<GameObject>`
	`Dijkstra(GameObject[] Graph,`
	`GameObject source, GameObject target)`
	`{`
		Inisiasi class, dan penetapan start point dan end node

2.	`Dictionary<GameObject,float> dist =`
	`new Dictionary<GameObject,float>();`
	
	`Dictionary<GameObject,GameObject>`
	`previous = new`
	
	`Dictionary<GameObject,GameObject>();`
	
	`List<GameObject> Q = new`
	`List<GameObject>();`
		Pembuatan objek

3.	`foreach(GameObject v in Graph)`
	`{`
	`dist[v] = Mathf.Infinity;`
	`previous[v] = null;`
	`Q.Add(v);`
	`}`
	`dist[source] = 0;`
		Perulangan untuk menghitung jarak tiap node

4.	while(Q.Count > 0)
	{ float shortestDistance =
	Mathf.Infinity;
	
	GameObject shortestDistanceNode =
	null;
	
	foreach(GameObject obj in Q)
	{ if(dist[obj] < shortestDistance)
	{
	shortestDistance = dist[obj];
	
	shortestDistanceNode = obj;
	}
	}
	
	GameObject u = shortestDistanceNode;
	Q.Remove(u);
		Pencarian node terpendek

5.	if(u == target)
	{
	Stack<GameObject> S = new
	Stack<GameObject>();
	
	while(previous[u] != null)
	{
	S.Push (u);
	u = previous[u];
	}
	return S;
	}
		Proses penemuan target

6.	if(dist[u] == Mathf.Infinity)
	{
	break;
	}
	
	foreach(GameObject v in
	u.GetComponent<Node>().neighbors)
	{
	
	float alt = dist[u]
	+(u.transform.position -
	v.transform.position).magnitude;
		Pemilihan node-node tetangga

7.	if(alt < dist[v])
	{ dist[v] = alt;
	previous[v] = u;
	}
	}
	}
	return null;
	}	

4.3 Implementasi Pada Game

Implementasi merupakan proses pembangunan komponen-komponen pokok suatu sistem. Komponen tersebut dibangun berdasarkan desain dan rancangan yang telah dibuat sebelumnya.

4.3.1 Pembangunan Terrain

Gambar 4.1 Tampilan *Terain*

4.3.2 Halaman *Splashscreen*

Gambar 4.2 Halaman *Splashscreen*

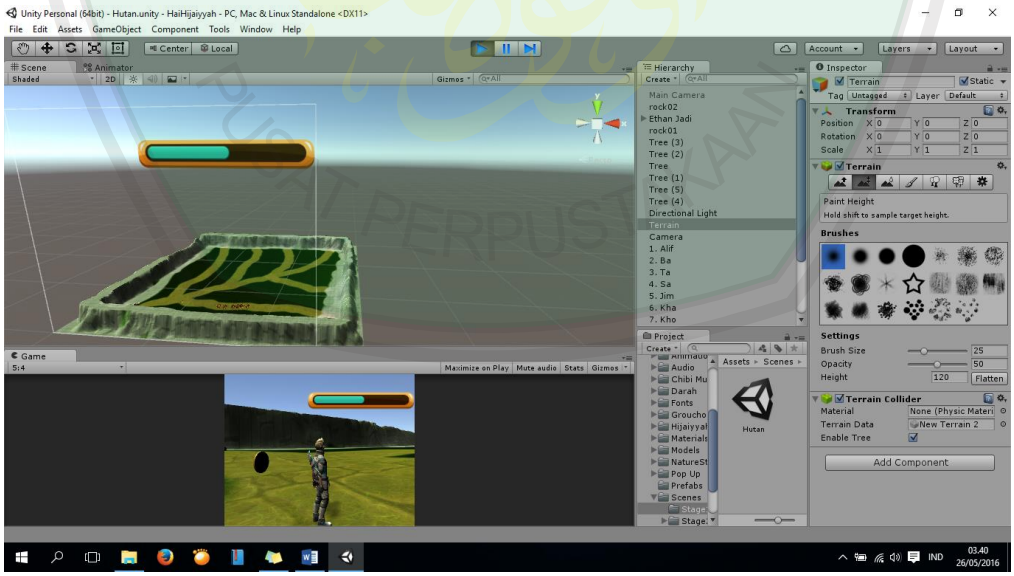
4.3.3 Antarmuka Menu

Gambar 4.3 Tampilan Menu *Screen*

4.3.4 Menu Petunjuk

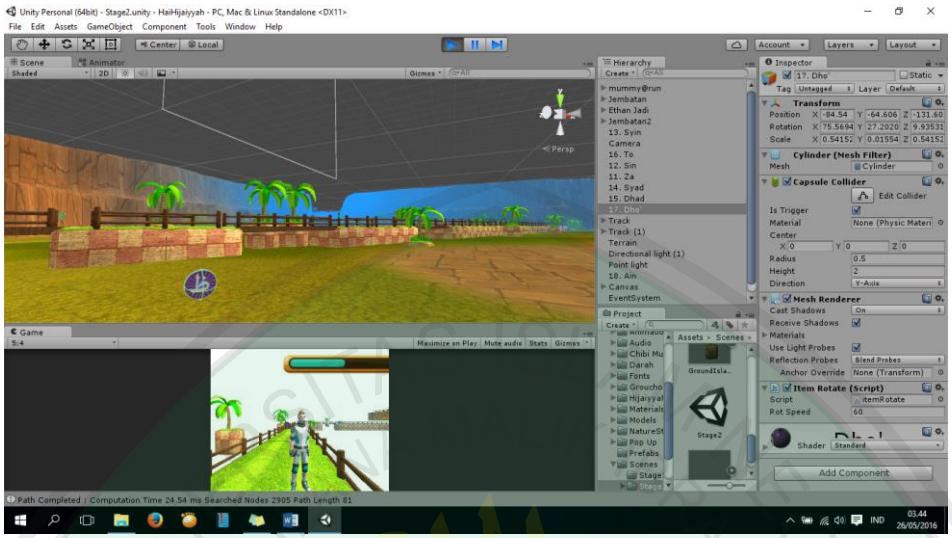
Gambar 4.4 Halaman Petunjuk

4.3.5 Pengaturan Terrain Stage 1



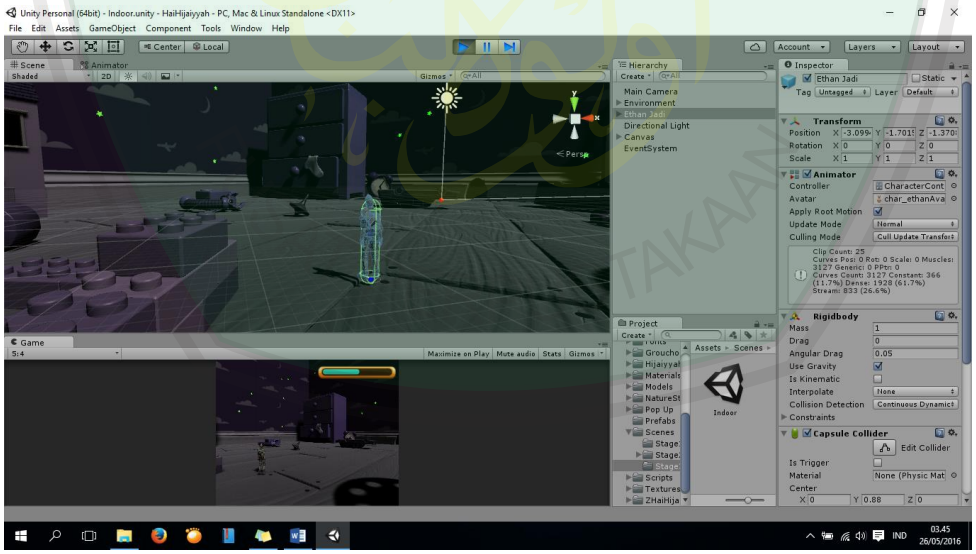
Gambar 4.5 Pengaturan *Terrain* Stage 1

4.3.6 Pengaturan Terrain Stage 2



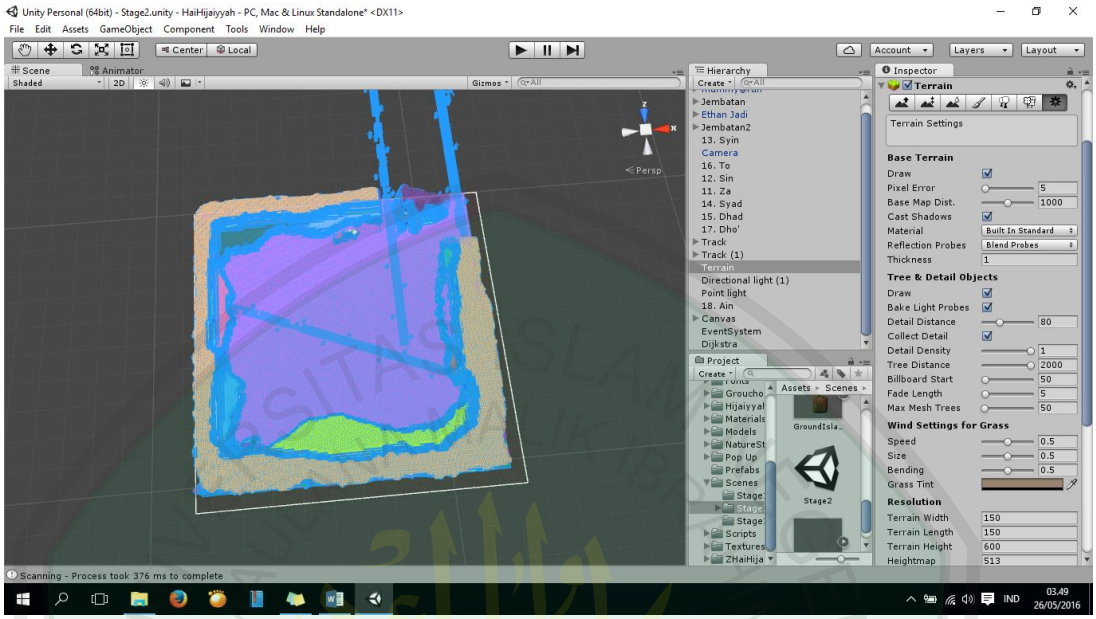
Gambar 4.6 Pengaturan *Terrain* Stage 2

4.3.7 Pengaturan Terrain Stage 3



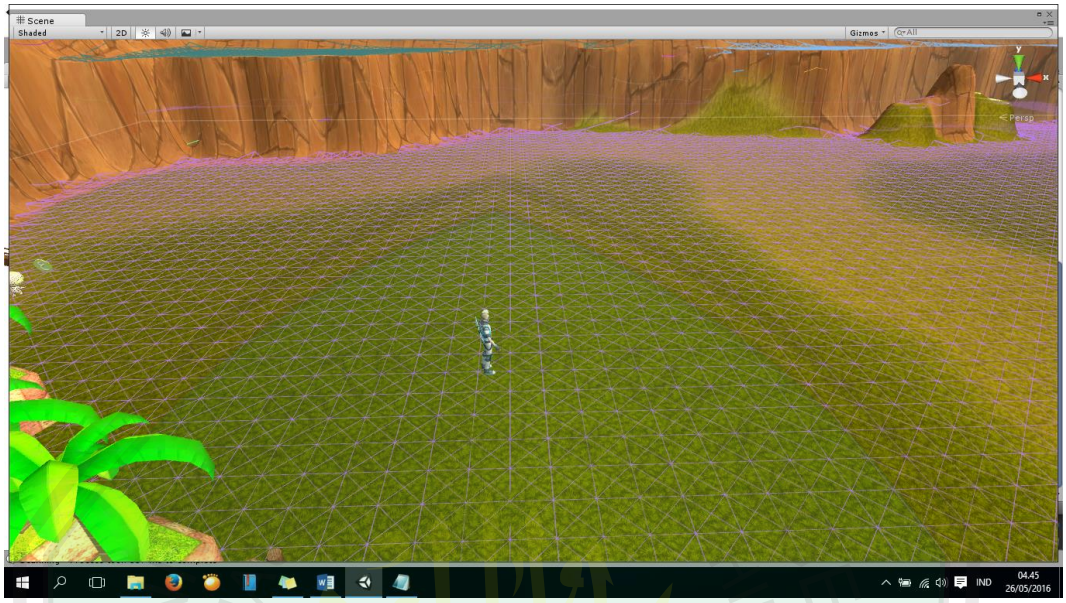
Gambar 4.7 Pengaturan *Terrain* Stage 3

4.3.8 Pembuatan Batasan Lintasan Agen Musuh



Gambar 4.8 Pengaturan Lintasan

4.3.9 Node-node Lintasan

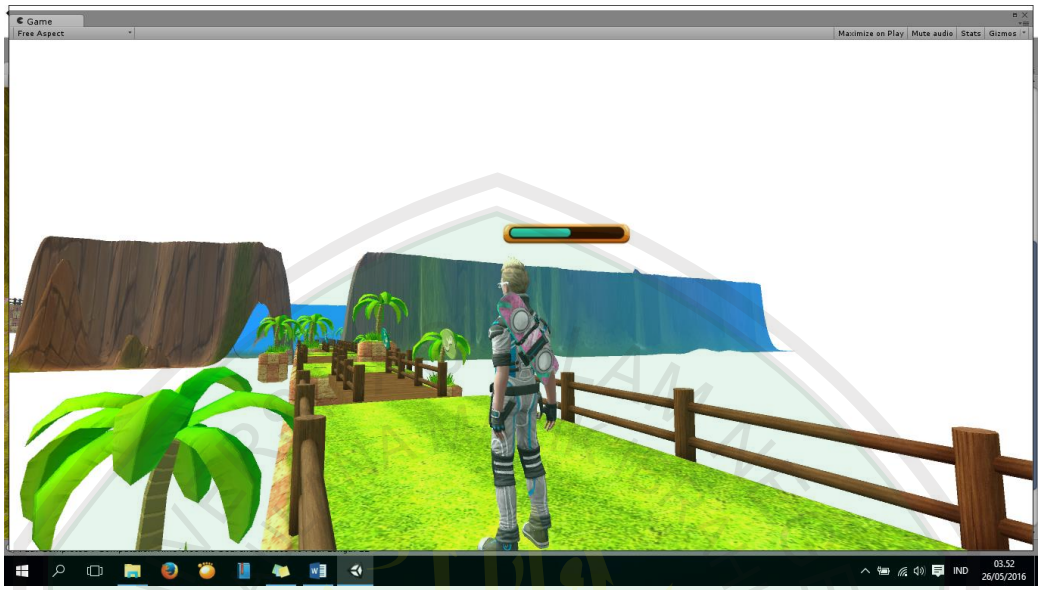


Gambar 4.9 *Node* Lintasan

4.3.10 *Scene* Game pada *Stage* Pertama

Gambar 4.10 Tampilan *Game* Pada *Stage* Pertama

4.3.11 *Scene* Game pada *Stage* Kedua

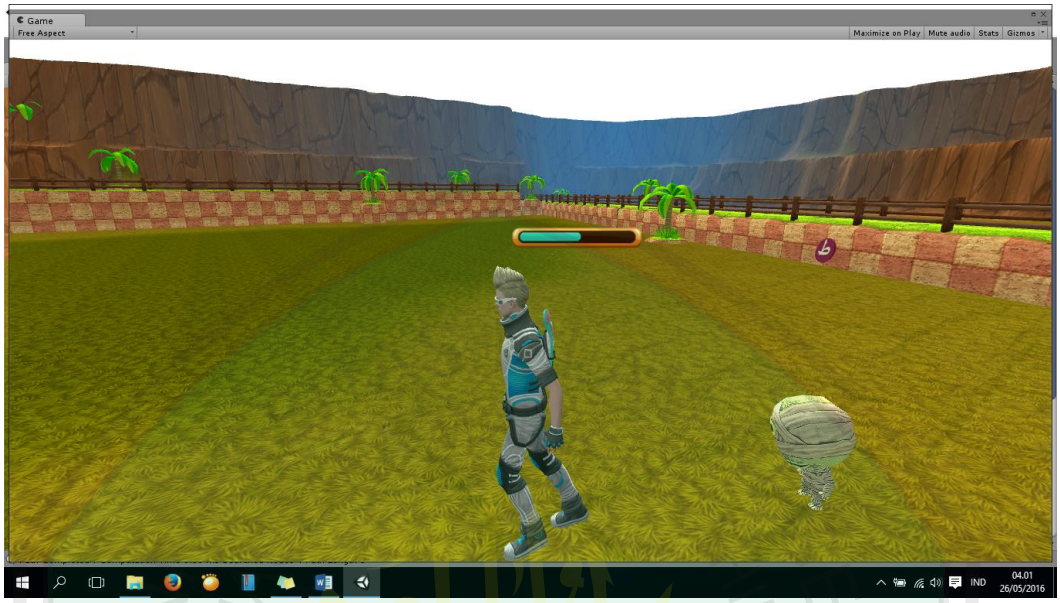


Gambar 4.11 Tampilan *Game* Pada *Stage* Kedua

4.3.12 *Scene* Game pada *Stage* Ketiga

Gambar 4.12 Tampilan *Game* Pada *Stage* Ketiga

4.3.13 *Scene Game* Saat NPC Mengejar

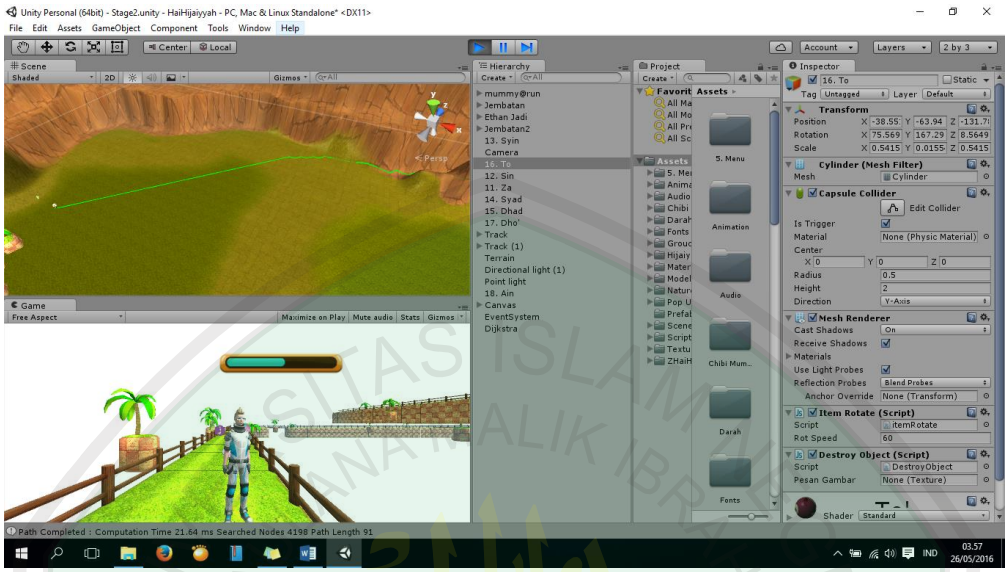


Gambar 4.13 Tampilan Game Saat NPC Mengejar

4.3.14 Scene Game Saat NPC Menyerang

Gambar 4.14 Tampilan *Game* Saat NPC Menyerang

4.3.15 Lintasan NPC



Gambar 4.15 Tampilan *Graph* Lintasan yang Bisa Dilalui NPC

4.3.16 Item Hijaiah

Gambar 4.16 Item Hijaiah

4.3.17 Keterangan Item *Game*

Gambar 4.17 Tampilan Keterangan Item Hijaiah

4.4 Uji Coba

Pada subbab ini membahas tentang uji coba yang telah dilakukan. Terdapat dua uji coba yakni uji coba imlementasi algoritma *Dijkstra* dan uji coba pada *game*. Berikut penjelasan terkait uji coba yang dilakukan yang dimaksud.

4.4.1 Uji Coba Algoritma

Uji coba Dijkstra dilakukan untuk mengetahui kinerja algoritma dalam menemukan rute. Proses ini melakukan uji coba dengan mengambil nilai koordinat musuh dan pemain.

Posisi Pemain :

(X: -80.3, Y: -65.1, Z: -114.8)

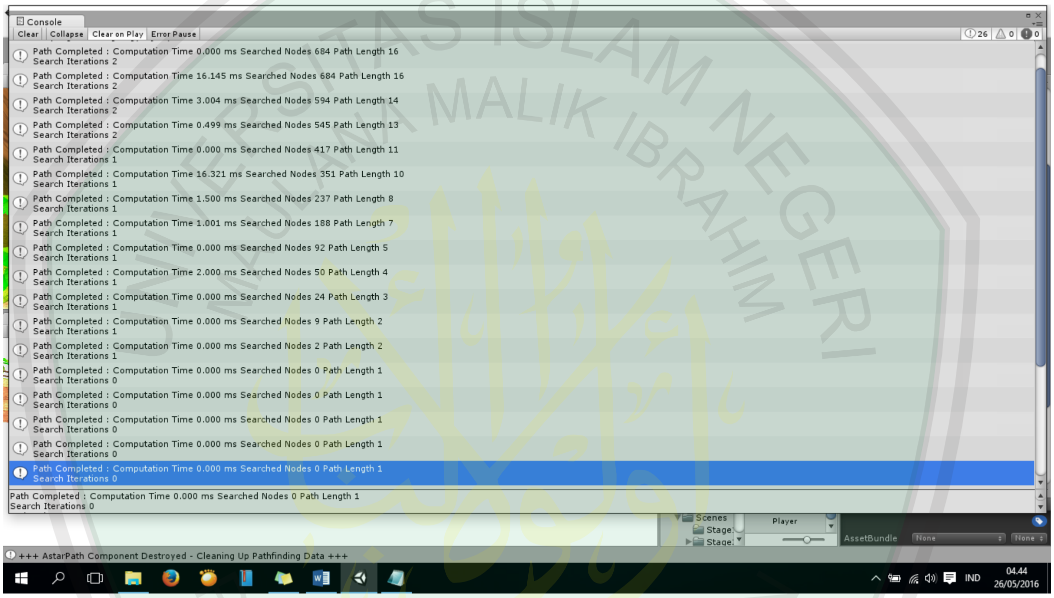
Posisi NPC :

(X: -64.6, Y: -65.1, Z: -119.0)

Tabel 4.4 Hasil Uji Coba Algoritma *Dijkstra*

Langkah	Nilai	Koordinat Start Node	Koordinat End Node
1	**F: 17070**	**Point: (-80.3, -65.1, -114.8)**	**(-64.6, -65.1, -119.0)**
2	F: 17070	Point: (-79.8, -65.1, -114.7)	(-64.6, -65.1, -119.0)
3	F: 15070	Point: (-78.3, -65.1, -114.4)	(-64.6, -65.1, -119.0)
4	F: 14070	Point: (-76.7, -65.1, -114.4)	(-64.6, -65.1, -119.0)
5	F: 12070	Point: (-75.2, -65.1, -114.4)	(-64.6, -65.1, -119.0)
6	F: 11070	Point: (-73.6, -65.1, -114.4)	(-64.6, -65.1, -119.0)
7	F: 9070	Point: (-72.0, -65.1, -114.4)	(-64.6, -65.1, -119.0)
8	F: 8070	Point: (-70.5, -65.1, -114.4)	(-64.6, -65.1, -119.0)
9	F: 5252	Point: (-69.2, -65.1, -115.2)	(-64.6, -65.1, -119.0)
10	F: 4242	Point: (-68.1, -65.1, -116.3)	(-64.6, -65.1, -119.0)

11	F: 2828	Point: (-67.1, -65.1, -117.4)	(-64.6, -65.1, -119.0)
12	F: 1414	Point: (-65.9, -65.1, -118.4)	(-64.6, -65.1, -119.0)
13	F: 1000	Point: (-65.4, -65.1, -118.8)	(-64.6, -65.1, -119.0)
14	**F: 0**	**Point: (-65.4, -65.1, -119.0)**	**(-64.6, -65.1, -119.0)**



Gambar 4.18 Tampilan Uji Coba

4.4.2 Uji Coba *Game*

Uji coba ini dilakukan untuk mengetahui apakah *game* yang telah dibuat dapat diimplementasikan di PC. Berikut hasil pengujian yang disajikan dalam tabel

Tabel 4.5 Uji Coba *Game*

No.	Versi OS	Layar	Spesifikasi	RAM	Keterangan
1.	Windows 7	14”	Acer Aspire 4740 Processor Intel (R) Core (TM) 2,13Ghz (4 CPU) VGA Intel HD Graphic Media Accelerator	4 GB	Tampilan menu berjalan dengan baik. Tombol berfungsi dengan baik. Tampilan *game* berjalan dengan baik.
2.	Windows 8	14”	Asus A43SA Processor Intel Core i3-2330M, VGA ATI 6730 (2 GB)	2 GB	Tampilan menu berjalan dengan baik. Tombol berfungsi dengan baik. Tampilan *game* berjalan dengan baik.
3.	Windows 8.1	14”	Asus NV46J Processor Quadcore 2.5 Ghz VGA Nvidia 2 GB	4 GB	Tampilan menu berjalan dengan baik. Tombol berfungsi dengan baik. Tampilan *game* berjalan dengan baik.

4.	Windows 10	14”	Acer Aspire V5 Prosesor Intel (R) Celeron (R) 1.0 GHz VGA Intel (R) HD Graphics	2 GB	Tampilan menu berjalan dengan baik. Tombol berfungsi dengan baik. Tampilan *game* berjalan dengan baik.

Dari pengujian yang dilakukan sebanyak 4 kali pada berbagai platform windows yang memiliki sistem operasi dan spesifikasi berbeda. Dapat diketahui prosentase pengujian pada **Tabel 4.6** di bawah ini.

Tabel 4.6 Persentase Hasil Pengujian Game

No	Jenis Pengujian	Baik	
		Jumlah	%
1	Sistem	4	(4/4) x 100 = 100%
2	Tombol	12	(12/12) x 100 = 100%
3	Tampilan	4	(4/4) x 100 = 100%

Keterangan:

Tabel di atas merupakan tabel yang berisi hasil pengujian *game* yang diterapkan pada 4 sistem operasi windows desktop. Hasil penerapan sistem telah dijelaskan pada **Tabel 4.5**. Hasil persentase yang didapatkan dari pengujian adalah **100%** *game* dapat berjalan dengan baik pada *device* tersebut.

4.5 Integrasi Dalam Islam

Pembelajaran merupakan hal yang sangat penting bagi umat manusia khususnya umat muslim. Manusia berkewajiban untuk senantiasa belajar segala sesuatu terkait hubungan manusia secara vertikal (*hablum minallah*) dan hubungan manusia secara horizontal (*hablum minannas*). Kajian terkait *hablum minallah* termasuk di dalamnya perwujudan cinta terhadap ajaran islam dalam hal ini tentu berkaitan dengan cinta Al-Quran.

Terkait dengan al-Qur'an, bahasa Arab menjadi komponen utama. Maka belajar huruf hijaiah menjadi hal yang sangat urgen. Demi cita-cita untuk mengetahui kedalaman kandungan isi dari kitab al-Quran yang merupakan perwujudan dari pedoman hidup di dunia serta tujuan untuk menggapai ridho Allah. Belajar huruf hijaiah merupakan gerbang awal untuk melestarikan kandungan ajaran Qur'an.

Al-Quranul kariim juga memiliki tujuan yang sama terkait pelestarian dan pengkajian ilmu-ilmu Allah. Allah *Subhanahu wa Ta'ala* menghaturkan lantunan indah ayat-ayatNya agar manusia dapat menjadikannya sebagai bahan pertimbangan untuk senantiasa berlomba dalam kebaikan, *fastabikhul khoirat*. Berlomba meraih keberuntungan dan amal shaleh bagi orang-orang yang beriman serta turut menyebarkan kebaikan kepada saudara-saudaranya.

Sejarah mencatat banyak sekali nilai historis dan ilmu yang dapat dijadikan teladan. Fungsi proses pembelajaran atau menuntut ilmu adalah dinaikkan

derajatnya oleh Allah Swt. Pada *Al-Quran* surat *Al-Mujaadilah* ayat 11 yang berbunyi :

يَٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُوٓاْ إِذَا قِيلَ لَكُمۡ تَفَسَّحُواْ فِي ٱلۡمَجَٰلِسِ فَٱفۡسَحُواْ يَفۡسَحِ ٱللَّهُ لَكُمۡۖ وَإِذَا قِيلَ ٱنشُزُواْ فَٱنشُزُواْ يَرۡفَعِ ٱللَّهُ ٱلَّذِينَ ءَامَنُواْ مِنكُمۡ وَٱلَّذِينَ أُوتُواْ ٱلۡعِلۡمَ دَرَجَٰتٖۚ وَٱللَّهُ بِمَا تَعۡمَلُونَ خَبِيرٞ ١١

Terjemahan ayat : *Hai orang-orang beriman apabila kamu dikatakan kepadamu: "Berlapang-lapanglah dalam majlis", maka lapangkanlah niscaya Allah akan memberi kelapangan untukmu. Dan apabila dikatakan: "Berdirilah kamu", maka berdirilah, niscaya Allah akan meninggikan orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat. Dan Allah Maha Mengetahui apa yang kamu kerjakan. (QS. al-Mujaadilah ayat 11)*

Berdasar tafsir *Jalalin* mengenai QS al Mujaadilah ayat 11. *Yaitu berupayalah dengan sungguh-sungguh walau dengan memaksakan diri untuk memberi tempat orang lain dalam majlis-majlis yakni satu tempat, baik tempat duduk maupun bukan tempat duduk, apabila diminta kepada kamu agar melakukan itu maka lapangkanlah tempat untuk orang lain itu dengan suka rela. Jika kamu melakukan hal tersebut, niscaya Allah akan melapangkan segala sesuatu buat kamu dalam hidup ini. Dan apabila di katakan: ”Berdirilah kamu ketempat yang lain, atau untuk diduduk tempatmu buat orang yang lebih wajar, atau bangkitlah melakukan sesuatu seperti untuk shalat dan berjihad, maka berdiri dan bangkit-lah, Allah akan meninggikan orang-orang yang beriman di*

antara kamu wahai yang memperkenankan tuntunan ini.dan orang-orang yang diberi ilmu pengetahuan beberapa derajat kemudian di dunia dan di akhirat dan Allah terhadap apa-apa yang kamu kerjakan sekarang dan masa akan datang Maha Mengetahui. (M. Quraish Shihab, 2006)

Tuntunan untuk senantiasa menjaga dan melestarikan juga dapat disarikan melalui QS. ar-Rum ayat 41, Allah SWT berfirman:

ظَهَرَ ٱلْفَسَادُ فِى ٱلْبَرِّ وَٱلْبَحْرِ بِمَا كَسَبَتْ أَيْدِى ٱلنَّاسِ لِيُذِيقَهُم بَعْضَ ٱلَّذِى عَمِلُوا۟ لَعَلَّهُمْ يَرْجِعُونَ ﴿٤١﴾

Terjemahan ayat : *Telah nampak kerusakan di darat dan di laut disebabkan karena perbuatan tangan manusia, supaya Allah merasakan kepada mereka sebahagian dari (akibat) perbuatan mereka, agar mereka kembali (ke jalan yang benar). (QS. ar-Rum ayat 41)*

Melalui permainan ini, pengguna diharapkan mampu memahami huruf hijaiah secara menyeluruh. Mengambil tauladan yang dapat dijadikan khasanah dalam mengarungi kehidupan di masa mendatang. Dasar tentang huruf hijaiah dan bahasa Arab disarikan dari berbagai sumber yang tervalidasi keabsahannya.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari implementasi dan pengujian yang dilakukan peneliti, maka diambil kesimpulan sebagai berikut :

1. Pembuatan *game* berbasis dekstop bergenre edukasi sebagai media pengenalan huruf hijaiah dapat menggunakan Unity 5 Free (Personal Edition) sebagai *game engine* dan *asset store* sebagai sumber *asset*.
2. Penggunaan Algoritma Dijkstra berhasil diterapkan sebagai penggerak perilaku pencarian pada NPC. Ditunjukkan dengan uji coba pada **Tabel 4.5**. Dalam proses pencarian, NPC mampu melewati halangan yang ada dan berhasil menemukan keberadaan lokasi *player*. *Game Hai Hijaiah* telah diuji cobakan ke 4 sistem operasi berbeda pada platform windows desktop, pada berbagai *device* tersebut game ini menunjukkan tingkat keberhasilan 100% pada uji coba sistem, tampilan dan tombol.

5.2 Saran

Peneliti yakin dengan penuh kesadaran bahwa dalam pembuatan permainan ini masih ada begitu banyak kekurangan. Kekurangan yang sangat perlu dilakukan pengembangan demi sumbangsih terhadap ilmu pengetahuan, diantaranya :

1. Menambah jumlah level permainan serta aturan untuk kenaikan level sehingga permainan menjadi lebih menantang.
2. Menambah ragam NPC dengan perilaku yang bervariasi dan disertai animasi yang menarik

3. Permainan ini tidak hanya disajikan dalam *platform desktop* saja, namun juga bisa dikembangkan pada *platform smartphone* agar pemahaman terkait huruf hijaiah semakin kian diminati.
4. Mengingat *genre* dari game ini adalah *game adventure* yang diterapkan sebagai media pengenalan huruf hijaiah, diharapkan dalam pengembangan ke depan *game* ini bisa dimainkan oleh siswa-siswa SD/ MI sampai SMA/ MA dan generasi selanjutnya.

DAFTAR PUSTAKA

Agustinus Nilwan. 1998. *Pemrograman Animasi dan Game Profesional 4*. Jakarta : Elex Media Komputindo.

Anggara. 2008. *Memahami Teknik Dasar Pembuatan Game Berbasis Flash*. Gave Media: Yogyakarta.

Arif, Yunifa Miftachul dan Ririen Kusumawati. 2012. *Attack Strategy For NPC In FPS Game Using Fuzzy Sugeno*. Basic Science International Conference.

Apperley, Thomas H. 2008. *Genre and Game Studies: Toward a Critical Approach to Video Game Genres*. University of Melbourne

Basori, Ahmad Hoirul dkk. 2008. Pencarian Rute Terpendek dalam Dunia 3 Dimensi Berdasarkan Algoritma Dijkstra. JUTI : Vol 7 No.2 hal 81-86

Brent Ellison .2008.*"Defining Dialogue Systems*. Gamasutra.

Dimyati A, 2006. *Operations Research, Model-Model Pengambilan Keputusan*, Sinar Baru Algensindo : Bandung

John Von Neumann and Oskar Morgenstern. 1953. *Theory of Games And Economic Behavior*. Princenton. Princenton University Press.

Kartono. 1994. *Teori Permainan (Game Theory).* Andi Offset : Yogyakarta)

Martino Bardi, Juan Pablo Maldonado Lopes. 2015. A Djikstra-type for dynamic games <hal-00881218v2>

M. Quraish Shihab. 2006. *Tafsir Al-Misbah, Pesan Kesan dan Keserasian al-Qur'an*, Volume XIV, Jakarta: Lentera Hati. hlm., 77

Millington, Ian. 2006. *Artificial Intelligence for Games.* USA : Elsevier.Inc

Mclean, Alex. 2002. *Hunting Down the Player in a Convincing Manner*. Pivotal Games

Rolling, Andrew dan Ernest Adams. 2003. *Game Design*. USA: New Riders Publishing.

Safaat, Nazaruddin. 2012. Pemrograman Aplikasi Mobile SmartPhone dan Tablet PC Berbasis Android. Informatika: Bandung.

Sutojo, T; Mulyanto, Edi; Suhartono; Vincent. 2011. *Kecerdasan Buatan*. Yogyakarta: ANDI.

Yahya, Saiful dkk. 2014. Rancang Bangun Permainan "Werkudara" Menggunakan Dijkstra Pada Agen Musuh. Malang : The 1st Internatiional Conference for Arts and Arts Education on Indonesia (ICAAE)

Yunifa, Fachrul, Fressy. 2011. *Desain Perubahan Perilaku pada NPC Game Menggunakan Logika Fuzzy*. Malang : Seminar on electrical, informatics, and ITS education.

Setiawan, Wira. 2015. *Tentang Algoritma Dijkstra*. https://wirasetiawan29.wordpress.com/2015/04/02/tentang-algoritma-dijkstra/ diakses pada 2 Desember 2015

http://www.asmaul-husna.com/2015/07/belajar-huruf-hijaiah.html diakses pada 2 Desember 2015