

## IMPLEMENTASI INTELLIGENT DECISION SUPPORT SYSTEM PADA PENGAMBILAN KEPUTUSAN PLAYER DALAM VIDEO GAME

Oleh:

Ian Yo Salli<sup>1\*</sup>, Syaiful Rahman<sup>2</sup>, Baizul Zaman<sup>3</sup>

<sup>1,2</sup>Teknik Informatika, STMIK Kharisma Makassar

**Abstrak:** Tujuan dari penelitian ini adalah untuk menerapkan suatu algoritma yang dikembangkan pada Intelligent Decision Support System (IDSS) agar dapat menjadi data reliable yang dapat dipakai sebagai pertimbangan. Manfaat dari penelitian ini yaitu menjadi referensi tambahan mengenai penggunaan IDSS dalam pengembangan suatu aplikasi. Dalam penelitian ini, penulis membuat suatu video game sederhana berjudul Gold and Fame untuk menguji algoritma yang akan dipakai sebagai bahan pertimbangan dalam IDSS. Video game ini dibuat dengan game engine Unity 5 dalam bahasa pemrograman C#. Dalam penelitian ini, penulis mengembangkan metode pathfinding A\* yang kemudian akan menyediakan data untuk dipertimbangkan pada IDSS. Hasil dari pengujian algoritma oleh penulis, didapatkan bahwa, penggunaan algoritma yang dikembangkan lebih efisien dari standard sehingga dapat dipakai sebagai pengolah data pada IDSS.

Kata kunci : Intelligent Decision Support System, Pathfinding

**Abstract:** The purpose of this research is to apply an algorithm developed in the Intelligent Decision Support System (IDSS) so that it can be reliable data that can be used as a consideration. The benefit of this research is that it becomes an additional reference regarding the use of IDSS in the development of an application. In this study, the authors made a simple video game entitled Gold and Fame to test the algorithm that will be used as a consideration in IDSS. This video game was made with the Unity 5 game engine in the C # programming language. In this study, the authors developed the pathfinding method A \* which then provided data for consideration in the IDSS. The results of the algorithm testing by the author show that the use of the developed algorithm is more efficient than the standard so that it can be used as a data processor in IDSS.

Keywords : Intelligent Decision Support System, Pathfinding

### PENDAHULUAN

*Video game* merupakan salah satu industry yang berkembang pesat pada zaman modern ini, dimana sekarang permintaan untuk game yang realistis semakin meningkat. Untuk meningkatkan tingkat realita sebuah *video game*, pengembangan terhadap *AI* tidak lepas dari konteks tersebut. (Xiao Cui & Hao Shi, 2011).

Salah satu teknik mendasar sebuah *AI* adalah *pathfinding*. Masalah dalam beberapa teknik *pathfinding* dalam penentuan keputusan oleh *AI* seperti yang dicantumkan oleh Xiao Cui & Hao Shi (2011) dalam konteks *video game* adalah *AI* hanya mengkalkulasi jarak

---

\* Corresponding author : Ian Yo Salli (ianyosalli\_17@kharisma.ac.id)

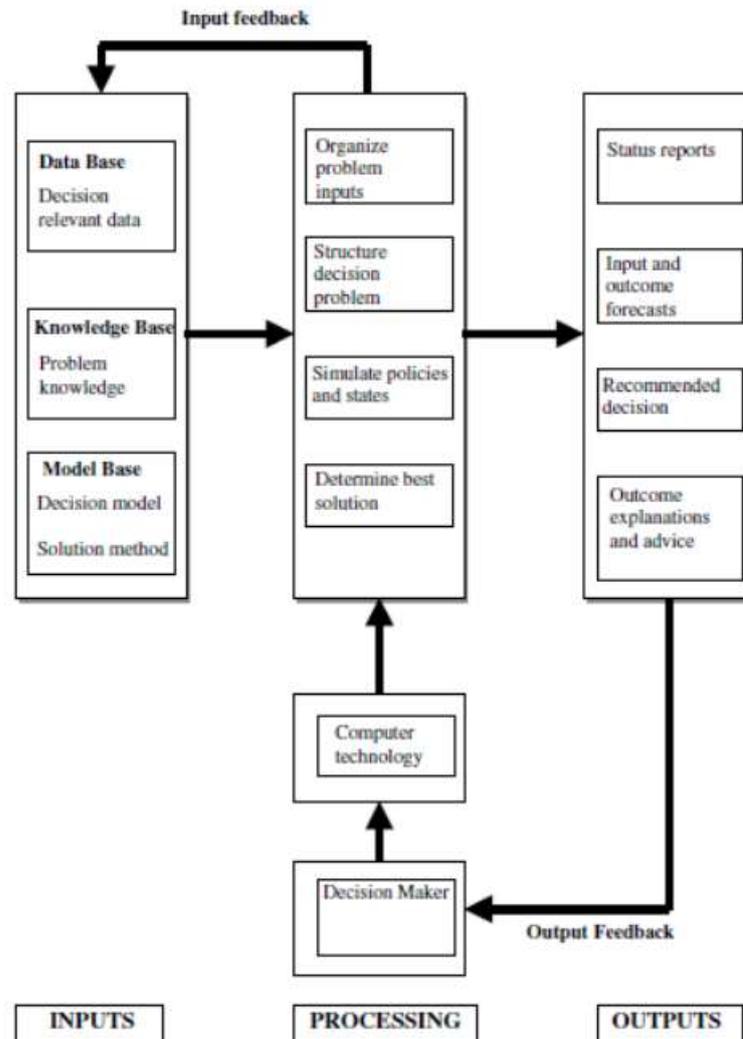
terpendek antara dua titik, sedangkan pada *video game*, *AI* perlu dapat menganalisa keadaan sekitar, seperti ada tidaknya halangan atau bahaya dalam menentukan jalur perjalanan. Untuk mengatasi permasalahan tersebut, peneliti akan menggunakan teknik *pathfinding* yang diusulkan oleh Xiao Cui & Hao Shi (2011) dan *threat analysis* yang diusulkan oleh Bjarnolf, Gustavsson, Brax & Fredin (2008) untuk meningkatkan kemampuan *AI* dalam membuat keputusan yang lebih pintar (*IDSS*) dengan menggunakan *UNITY 5* dalam bahasa *C#* yang akan diuji pada game peneliti berjudul *Gold and Fame* di bagian *battle scene* khususnya dalam penentuan jalur pergerakan karakter. Penelitian Gloria Phillips-Wren (2008) "*Intelligent Decision Making: An AI-Based Approach*" mempelajari tentang *Intelligent Decision Support System (IDSS)* untuk pengembangan penentuan keputusan. *IDSS* merupakan penggunaan teknik-teknik *AI* dalam penentuan keputusan, sehingga peneliti dapat menggabungkan beberapa teknik *AI* untuk mengembangkan penentuan keputusan *AI* agar dapat terlihat lebih pintar.

## LANDASAN TEORI

### 1. Intelligent Decision Support System (IDSS)

Menurut Liang (2005), *Decision Support System (DSS)* awalnya dikenal sebagai sistem untuk membantu dalam pembuatan keputusan pada beberapa masalah yang dihadapi dengan memberikan beberapa situasi untuk dianalisis agar dapat menghasilkan beberapa kesimpulan yang dapat dipakai sebagai pertimbangan dalam menentukan keputusan. Ketika teknik-teknik *Artificial Intelligence (AI)* digunakan dalam pengembangan *DSS*, sistem tersebut disebut *Intelligence Decision Support System (IDSS)*. Penggunaan *AI* pada *DSS* digunakan untuk menghasilkan kesimpulan efektif yang dapat dipakai sebagai pertimbangan penentuan keputusan. (Wren, 2013)

Linger & Burstein (1997) menyediakan dua lapisan pada *framework* mereka pada *IDSS*, lapisan *pragmatic* berhubungan dengan pekerjaan yang akan dilakukan pada masalah, dan lapisan *conceptual* berhubungan dengan proses dan struktur dari masalah. Menggunakan lapisan tersebut, arsitektur *IDSS* dapat dikembangkan. Dapat dilihat pada gambar 2.3, *IDSS* memiliki *database*, *knowledge base*, dan *model base*, dimana beberapa atau semua akan menggunakan teknik *AI*.



Gambar 1. Arsitektur IDSS (Forgionne et al., 2005)

*Database* berisi semua data yang relevan terhadap masalah, termasuk nilai-nilai awal, tindakan-tindakan yang dapat diambil, dan pengukuran performa. *Knowledge base* berisi aturan untuk panduan untuk memilih suatu pilihan. *Model base* berisikan rekaman masalah dan cara pemilihan solusi bersama dengan algoritma dan metodologi untuk memberikan hasil dari model yang telah ditentukan. (Wren, 2009)

## 2. Pathfinding

Berdasarkan Xiao Cui dan Hao Shi pada “*A\*-based Pathfinding in Modern Computer Games*” tahun 2010, *pathfinding* secara umum diartikan sebagai pencarian jarak terpendek antar dua titik. Beberapa contoh dapat dilihat pada penjadwalan transit, *telephone routing*, *maze navigation*, atau *robot path planning*. Semakin meningkatnya industry game, *pathfinding* merupakan masalah populer yang sering dibahas. Beberapa masalah utama yang ditemukan dalam *pathfinding* adalah bagaimana computer dapat menghindari halangan dengan pintar dan bagaimana computer dapat memilih jalur yang paling efisien.

A\* merupakan algoritma pencarian yang dapat dipakai untuk menemukan solusi dari berbagai masalah. Untuk *pathfinding*, algoritma A\* berulang kali memeriksa lokasi belum dijelajahi yang paling menjanjikan. Ketika sebuah lokasi telah diperiksa, algoritma akan selesai apabila lokasi tersebut adalah objek pencariannya. Kalau belum, algoritma akan menandai setiap tetangga dari lokasi sekarangnya (Cui & Shi, 2011). Berikut adalah *pseudo-code* dari algoritma A\* menurut Hart, Nilson, dan Raphael pada tahun 1998:

- a. Pasang lokasi awal pada *open list*.
- b. Ulang langkah berikut:
  - 1) Mencari titik dengan *f* terendah pada *open list*.
  - 2) Ubah menjadi *closed list*.
  - 3) Untuk setiap titik yang dapat dicapai dari titik sekarang
    - a) Kalau dia berada pada *closed list*, abaikan.
    - b) Kalau dia tidak berada pada *open list*, masukkan kedalam *open list*. Jadikan titik sekarang menjadi *parent* dari titik tersebut. Catat setiap *f*, *g*, dan *h* dari titik ini.
    - c) Apabila dia sudah berada pada *open list*, periksa apabila jalur ini adalah jalur yang lebih baik. Kalau ya, ganti *parent* menjadi titik sekarang dan kalkulasi ulang nilai *f* dan *g*.
  - 4) Berhenti ketika
    - a) Apabila objek pencarian sudah masuk kedalam *closed list*.
    - b) Tidak berhasil menemukan objek pencarian dan sudah tidak ada *open list* yang lain.
- c. Mengikuti jejak dari titik akhir hingga titik mulai. Jalur yang terbentuk merupakan jalur yang dicari.

### 3. Artificial Intelligence

*Artificial Intelligence (AI)* pertama diperkenalkan oleh pada John McCarthy pada tahun 1956 dengan ide mesin dapat bekerja seperti manusia sehingga membuat peneliti mencari apabila mesin dapat belajar dan berpikir sendiri seperti sebagaimana yang diperkenalkan oleh Alan Turing. Pengujian Turing menggunakan pendekatan pragmatis untuk mengidentifikasi apabila mesin dapat merespon seperti manusia. Menurut Khaled AISedrah (2017), *AI* adalah bagian studi yang mempelajari mengenai kapasitas pembelajaran mesin seperti manusia dan kemampuan untuk memberikan respon terhadap perilaku tertentu.

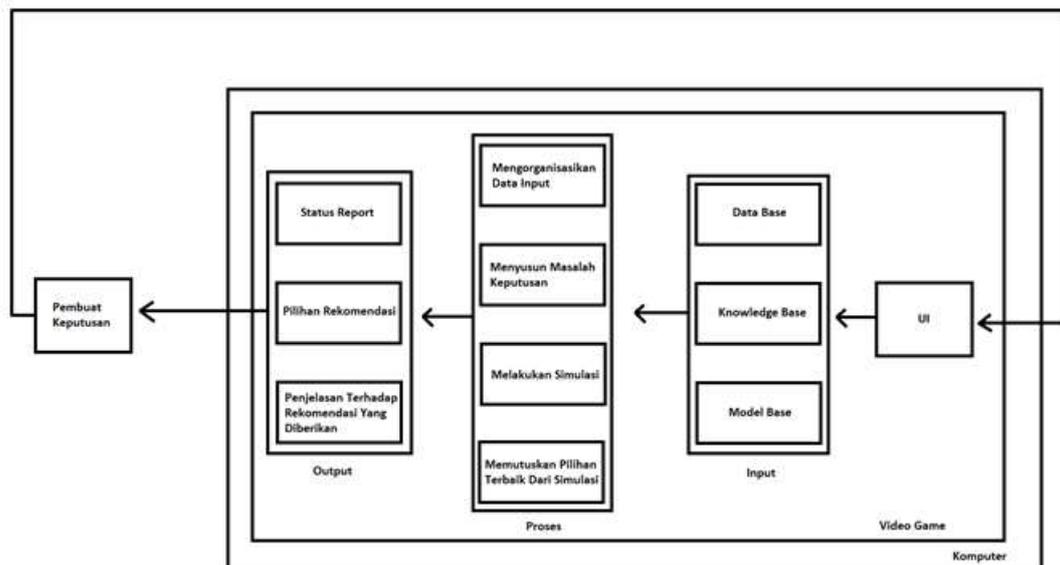
### 4. Teori Pengambilan Keputusan

Menurut Van Der Plight (2001), pengambilan keputusan manusia telah dipelajari oleh berbagai subjek, seperti ekonomi, filosofi, psikologi dan statistika. Studi pengambilan keputusan manusia adalah teori pilihan rasional. Konsep penjelasan yang paling berpengaruh dalam analisis keputusan manusia ketika tidak pasti adalah *Subjective Expected Utility (SEU)* dari alternative yang ada.

## ANALISIS DAN DESAIN SISTEM

Sistem pada kondisi awal belum menerapkan algoritma perkembangan. Terkadang dalam menentukan jalur, diperlukan dua titik agar sistem dapat mendapat jalur terpendek antara dua titik tersebut. Namun, tidak menutup kemungkinan akan ada halangan dalam jalur tersebut. Sehingga dalam pencarian jalur, jalur terpendek tidak selalu merupakan jalur terbaik.

Dari permasalahan yang terjadi perlu adanya solusi yang baik dalam memecahkan permasalahan tersebut, oleh karena itu dikembangkan algoritma *pathfinding* agar tidak hanya dicari jalur terpendek namun mencari jalur paling hemat *cost*. Pada sistem yang akan dibangun, pengguna perlu untuk memilih satu tempat untuk memulai pencarian jalur dan juga satu tempat sebagai target pencarian. Pengguna juga akan perlu untuk meletakkan berbagai halangan agar dapat dipakai sistem mencari jalur yang optimal.



Gambar 2 Arsitektur Sistem

Sebuah *IDSS* memiliki *data base*, *knowledge base*, dan *model base*, dimana beberapa atau semuanya akan menggunakan metode *AI*. Ketika pembuat keputusan (*player*) menjalankan aplikasi, *player* akan berinteraksi dengan *UI*. Ketika dalam proses interaksi dengan *UI* ada membutuhkan data yang perlu diproses, data tersebut akan diproses dengan *IDSS*, dimana *IDSS* terbagi menjadi tiga, input, proses, dan output.

Pada bagian input, data base memiliki semua data yang relevan terhadap masalah dalam menentukan keputusan. *Knowledge base* memegang semua data mengenai penyelesaian permasalahan, seperti panduan dalam menentukan pilihan, atau nasihat mengenai kemungkinan yang dapat terjadi. *Model base* merupakan daftar dalam model masalah keputusan dan berbagai pendekatan dalam membuat hasil akhir dari model tersebut.

Penentu keputusan akan menggunakan computer untuk memproses input menjadi output yang relevan. Beberapa hal yang akan terjadi dalam bagian proses:

1. Mengatur input masalah
2. Mengstrukturkan model masalah penentuan keputusan
3. Menggunakan model untuk mensimulasikan yang akan terjadi
4. Mencari solusi terbaik

Dalam melakukan bagian proses, *IDSS* menggunakan pengetahuan yang ada dalam knowledge base untuk membantu dalam melakukan pencernaan data. Hasil dari bagian proses ini akan memberikan bagian output berupa status report, rekomendasi, dan penjelasan rekomendasi.

Status report akan mengidentifikasi variabel yang relevan dan menunjukkan nilai untuk elemen-elemen masalah ini. Rekomendasi akan mengusulkan nilai yang paling mendekati pengukuran performa. Penjelasan akan membenarkan rekomendasi dan memberikan masukan dalam penentuan keputusan berikutnya. Untuk bagian dari output *IDSS*, hasil proses akan diberikan kepada penentu keputusan dalam bentuk data yang akan disajikan atas beberapa variabel, yaitu *distance*, *health damage*, dan *energy damage*. Variabel-variabel ini kemudian akan menjadi data untuk dipakai penentu keputusan dalam mempertimbangkan keputusannya.

				72 10 82	62 14 76	52 24 76	48 34 82	52 44 96		
				68 0 68	58 10 68	48 20 68	38 30 68	34 40 74	38 50 88	
		58 24 82						24 44 68	28 54 82	
		54 28 82	44 24 68	34 20 54	24 24 48	14 28 42	10 32 48	14 48 62	24 58 82	
		58 38 96	40 34 74	30 30 60	20 34 54	10 38 48	10 38 A	10 52 62	20 62 82	
			44 44 88	34 40 74	24 44 68	14 48 62	10 52 62	14 56 70	24 66 90	

Gambar 3 Pencarian Jalur

*Node* memiliki beberapa variabel yang kemudian akan menjadi pertimbangan bagi algoritma dalam menentukan jalur. Beberapa variabel utama dari *node* adalah *energy damage*, *health damage*, *walkable Boolean*, dan posisi array *node*. Ketika *player* telah menentukan titik mulai dan objek pencarian, sistem membuat variabel *f*, *g*, *h* yang berdasarkan dari perkiraan *cost* dari titik ke objek pencarian. Dimana *g* merupakan *cost* dari titik awal sampai titik sekarang, *h* merupakan perkiraan *cost* dari titik sekarang hingga titik akhir, *f* adalah  $g + h$ .

Berikut adalah contoh cara kerja algoritma yang disajikan oleh peneliti. Dapat dilihat pada gambar 3, A merupakan titik mulai dan B merupakan objek pencarian. Tujuan dari A adalah untuk mencari jalur hingga sampai di B. Algoritma akan mencari titik-titik tetangga dan melihat nilai dari setiap titik tersebut. Tetangga dari titik tersebut akan masuk kedalam *open list* sehingga dapat dilakukan pengecekan. Nilai tetangga terkecil akan dipilih, dan akan terbuka titik yang lain. Algoritma akan coba untuk berpindah ke titik dengan *cost* terendah kemudian memasukkan tetangganya kedalam *open list* dan apabila *cost* dari tetangga yang sudah ada lebih besar daripada *cost* baru dari titik sekarang, maka *cost* tetangga akan dirubah ke yang paling kecil. Algoritma akan melakukan perulangan sehingga sampai pada objek pencarian.

Apabila telah ditemukan objek pencarian, algoritma akan menjelajahi ulang titik-titik yang telah diambil dalam usaha untuk mencari objek pencarian tersebut seperti terlihat pada gambar 3. Setelah selesai dilakukan penjelajahan ulang, jalur yang terbentuk adalah jalur paling optimal (hemat *cost*) yang akan disajikan ke penentu keputusan.

## PENGUJIAN SISTEM

Pengujian dilakukan dengan membandingkan hasil antara algoritma standard dengan algoritma yang ditingkatkan sehingga menjadi bahan perbandingan sebagai penjelasan rekomendasi yang diberikan oleh IDSS. Penelitian Borg dan Gall pada tahun 2007 menyatakan bahwa jumlah sampel yang dibutuhkan untuk pengujian bersifat eksperimen dan komparatif adalah 15-30 data. Oleh karena itu, akan dilakukan pengujian algoritma pada kondisi yang sama. Akan dilakukan 3 kali pengujian setiap ukuran map, tersedia 6 ukuran map sehingga ada 18 total sampel. Hasil pengukuran tersebut dapat dilihat pada tabel 1.

Tabel 1 Hasil Pengukuran

Akumulasi hasil pengujian	IDSS	Standard	Selisih (Standard - IDSS)
Distance	900	555	-345
Processing Time	57ms	54ms	-3
Health Loss	1810	12220	10410
Energy Loss	14034	28846	14812

Berdasarkan tabel 1, dapat terlihat dengan mengimplementasikan algoritma yang dipakai dalam IDSS, kemampuan sistem semakin efisien dalam hal *cost* pergerakan. Sistem mampu mengkalkulasi jalur yang lebih efisien, dengan menggunakan jalur lebih jauh.

## KESIMPULAN

Berdasarkan hasil pengujian serta informasi yang dikumpulkan, maka pada penelitian ini dapat dihasilkan kesimpulan bahwa hasil akumulasi dari 18 pengujian didapatkan bahwa dengan menggunakan algoritma standard, jarak yang ditempuh adalah 555 kotak sedangkan dengan algoritma yang dikembangkan jarak ditempuh adalah 900 kotak, selisih didapatkan 345 kotak. Namun energy point dan health point yang terhabiskan pada algoritma standard

adalah 12220 dan 28846, sedangkan dengan menggunakan algoritma dikembangkan, energy point dan health point yang terhabiskan adalah 1810 dan 14034 dimana selisih antara kedua nilai adalah 10410 dan 14812. Dari hasil ini ditemukan bahwa, penggunaan algoritma yang dikembangkan lebih efisien dari standard sehingga dapat dipakai sebagai pengolah data pada IDSS.

#### DAFTAR PUSTAKA

- [1] Alsedrah. (2017). Artificial Intelligence. American University of The Middle East.
- [2] Borg, Walter R, Meredith D, Gall and Joyce P. Gall.2007. Education Research. New York: Pearson Education, Inc.
- [3] Belciug & Gorunescu. (2020). Intelligent Support Systems – A Journey To A Smarter Health Care, 57-70.
- [4] Coleman. (2008). Fractal Analysis of Stealthy Pathfinding Aesthetics. International Journal of Computer Games Technology.
- [5] Cui & Shi. (2010). A\*-based Pathfinding in Modern Computer Games. International Journal of Computer Science and Network Security volume 11, 125-130.
- [6] Cui & Shi. (2011). Direction Oriented Pathfinding in Video Games. International Journal of Artificial Intelligence & Applications Volume 2.
- [7] Mehta, Verma. (2015). A Review on Algorithms for Pathfinding in Computer Games. IEEE Sponsored 2<sup>nd</sup> International.
- [8] Pligt. (2001). Psychology of Decision Making. International Encyclopedia of the Social & Behavioral Sciences, 3309-3315.
- [9] Schiefer & Silva .(1995). An Evaluation of Expert Systems Technology For Decision Power. IFAC Artificial Intelligence in Agriculture, 279-285.
- [10] Shanteau & Pingenot. (2009). Subjective Expected Utility Theory. Encyclopedia of Medical Decision Making.
- [11] Wong. (2009). Artificial Intelligence for Computer Games. International Journal of Computer Games Technology Volume 2009.
- [12] Wren. (2008). Intelligent Decision Making: An AI-Based Approach.
- [13] Wren, Mora, Forgionne, & Gupta. (2009). An Integrative Evaluation Framework For Intelligent Support Systems. European Journal of Operational Research, 642-652.
- [14] Wren. (2013). Intelligent Decision Support Systems. Multicriteria Decision Aid and Artificial Intelligence, 25-44.
- [15] Thakur, R. N., & Pandey, U. S. 2019 . *The Role of Model-View Controller in Object Oriented Software Development*. Nepal Journal of Multidisciplinary Research Vol. 2, No.2.