penulisan kritik dan tinjauan

Hak cipta milik IB

### KOMPARASI ALGORITMA A STAR, DIJKSTRA, DAN BFS UNTUK PATH FINDING NPC DALAM GAME

Geraldin Everdin Pua

Richard Vinc N. Santoso, S.TI., M.T.I.

Richard Vinc N. Santoso, S.TI., M.T.I.

Richard Vinc N. Santoso, S.TI., M.T.I.

Hak Cipte UT BISNIS dan INFORMATIKA KWIK KIAN GIE Jakarta, Indonesia (Telp(021) 082194562671; E-mail: (57160405@student.kwikkiangie.ac.id, geralepua98@gmail.com)

Abstrak T. Ada banyak komponen and the componen and the component and the component

Dilindungi Urak Bagian atausta Ada banyak komponen yang membentuk sebuah game, salah satunya adalah Non Player Character, atau yang disingkat NPC. NPC dapat dirancang untuk bergerak dari satu posist ke posisi lain, dalam hal ini NPC dapat menggunakan metode *path finding* untuk bergerak dari satu posisi ke posisi lain. Ada Beberapa metode untuk mengimplementasikan path finding untuk NPC misalnya dengan menggunakan algoritma path finding, seperti A Star, Difkstra, dan Breadth First Search. Penelitian ini dirancang untuk menentukan apakah algoritma A Star dapat mengoptimasi path finding NPC dalam game. Peneliti mempelajari cara kerja path finding yang diteliti dan juga melakukan studi pada 5 jurnal terdahulu \_yang memakai algoritma path finding ∃didalamnya. Studi ini dilakukan untuk mengumpulkan data bagaimana cara kerja pergerakan NPC menggunakan path finding dan mempelajari hasil penelitian tersebut. *Metode* path finding yang akan dipelajari ada 3 yakni A Star, Dijkstra, dan Breadth First Search. Peneliti memberikan gambaran bagaimana NPC dapat bergerak dalam sebuah situasi menunjukkan beberapa contoh algoritma path Finding pada NPC dalam bergerak dari sebuah posisi Peneliti lainnya. menggunakan metode waterfall untuk menjalankan penelitiannya. Metode waterfall tersebut memiliki urutan proses yaitu adalah perencanaan, analisis, perancangan pemograman, pengujian dan implementasi. Ada 3 path finding yang digunakan penulis yakni A Star, Dijkstra, dan Breadth First Search. Peneliti

melakukan eksperimen path finding dengan membuat sebuah simulasi yang mencatat performa setiap path finding. Pada simulasi ini setiap path finding melakukan pencarian jalan keluar dalam sebuah maze yang dibuat oleh peneliti, lalu mencatat berapa lama dan berapa yang dibutuhkan proses menyelesaikan setiap maze. Dari data yang dihasilkan dapat dilihat bahwa Algoritma A Star dapat menyelesaikan maze dalam waktu dan banyaknya proses paling sedikit dibandingkan algoritma Dijkstra dan Breadth First Search. Dari penelitian vang telah dilakukan melakukan simulasi *maze* dengan algoritma – algoritma path finding, dapat disimpulkan bahwa algoritma path finding A Star dapat mengoptimasikan path finding **NPC** dibandingkan dengan algoritma path finding lainnya yakni Dijkstra dan BFS.

### Kata Kunci: NPC, path finding, Algoritma A Star, Algoritma Dijkstra, Algoritma Breadth First Search

### 1. Pendahuluan

Ada banyak komponen yang membentuk sebuah game, salah satunya adalah Non Player Character, atau yang disingkat NPC. NPC itu sendiri adalah sebuah karakter di dalam game di mana karakter tersebut tidak dikendalikan oleh para pemain dan hanya bisa dikendalikan oleh komputer. NPC merupakan hasil dari teknologi Artificial Inteligence, disingkat dengan AI. Dengan bantuan AI, NPC dapat dirancang untuk berinteraksi dengan object, maupun NPC dengan pemain.

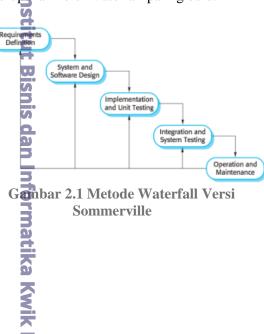


NPC dapat bergerak dari satu posisi ke posisi lain, dalam hal ini NPC dapat menggunakan metode path finding untuk bergerak dari satu posisi ke posisi lain. Ada beberapa metode untuk mengimplementasikan path finding untuk NPC misalnya dengan menggunakan algoritma path finding, seperti Dijkstra, A Star, dan Breadth First Search. Algoritma — algoritma tersebut dapat menghasilkan rute terpendek, akan tetapi ada beberapa masalah dengan algoritma — algoritma ini membutuhkan proses pencarian yang panjang.

Penelitian ini dilakukan dengan batasan masalah berfokus pada path finding yang dilakukan NPC, dan diimplementasikan dengan algoritma A Star, Dijkstra, dan Breadth First Search. Penelitian ini juga bertujuan untuk mengoptimasi path finding NPC dengan menggunakan A \*. Manfaat dari Penelitian ini adalah memberikan kontribusi untuk permasalahan path finding oleh NPC untuk game game ke depannya.

### -2. Metode Penelitian

Metode pengembangan sistem yang digunakan adalah metode Waterfall. Peneliti menggunakan metode Waterfall dikarenakan model metode Waterfall memiliki proses yang peneliharaan, jika ada sesuatu yang salah. Setiap peneliharaan, jika ada sesuatu yang salah setiap peneliharaan, jika ada sesuatu yang salah. Setiap peneliharaan, jika ada sesuatu yang salah setiap peneliharaan, jika ada sesuatu yang salah setiap peneliharaan peneliha



### 1. Requirements Definition

Pada tahap ini, penulis sebagai peneliti menentukan variabel – variabel dan algoritma apa saja yang akan dijadikan landasan dalam simulasi yang akan dilakukan. Peneliti disini juga membuat rancangan bagaimana hasil penelitian tersebut dibuat. Dan Peneliti mulai melakukan pengamatan dan pengumpulan data yang dibutuhkan pada objek yang akan diteliti.

### 2. System and Software Design

Pada tahap ini peneliti menggambaran ke dalam bentuk blueprint software sebelum proses pembuatan coding dimulai. Desain harus dapat mengimplementasikan semua kebutuhan yang telah disebutkan pada tahap sebelumnya.

### 3. Implementation and Unit Testing

Pada tahap ini peneliti melakukan pemrograman untuk membuat sistem sesuai perencanaan yang sebelumnya dibuat. Lalu peneliti melakukan pengujian apakah sistem yang dibuat sudah sesuai dengan data analisis yang telah dilakukan dan juga melakukan mengamati apakah sistem tersebut berjalan seperti kemauan peneliti. Peneliti melakukan pengujian pada algoritma path finding dengan eksperimen aplikasi simulasi yang telah dibuat.

### 4. Integration and System Testing

Pada tahap ini peneliti melakukan implementasi sistem yang telah dibuat kepada objek yang diteliti.

### 5. Operation and Maintenance

Untuk tahap ini penulis hanya akan melakukan perbaikan kekurangan sistem yang tidak terlihat pada tahap – tahap sebelumnya. Dan tidak akan melakukan *maintenance* karena sistem yang dibuat bukanlah sistem yang bekerja untuk digunakan oleh perusahaan

### A. Teknik Pengumpulan Data

Penulis akan menggunakan dua jenis data dalam penelitian ini, yaitu data primer dan data sekunder. Data primer yang akan digunakan penulis adalah data hasil penelitian yang akan dilakukan penulis yang didapatkan dengan menyiapkan beberapa simulasi yang akan dibuat penulis, melalui simulasi ini penulis akan mendapatkan data



- data hasil yang berkaitan dengan optimasi path finding. Sedangkan data sekunder yang penulis gunakan diambil dari jurnal - jurnal terdahulu.

## . Dilarang Studi Pustaka

Studi pustaka dilakukan dengan cara mengumpulkan data dari sumber yang sudah ada dari jurnal, artikel, video tutorial, media internet, buku digital, teks dan yang terkait mengenai penelitian sejenis untuk menjadi dasar dari penelitian <sup>□</sup>yan**g** dilakukan ini.

Dilindungi Undang-Undang Beberapa kata kunci yang peneliti gunakan adalah sebagai berikut:

- 1. **Path Finding** 1. **Path** 1. Path Finding
- 2. **E**A Star (A\*)
- 3. Dijkstra

mengutip sebagian atau seluruh karya tulis

ini tanpa mencantumkan dan menyebutkan sumber:

- 4. Breadth First Search (BFS)
- 5. Heuristic
- 6. **a**Cost
- 7. Open Set 8. Closed Set

### Eksperimen

Peneliti melakukan pengujian dengan membuat program simulasi buatan yang dijadikan alat untuk melakukan testing dengan menggunakan beberapa algoritma untuk mencari cara bagaimana membuat algoritma - algoritma tersebut dipakai berjalan dan sesuai yang diharapkan atau tidak.

Dalam hal ini penulis akan membuat sebuah simulasi untuk path finding yang akan dilakukan NPC, simulasi ini akan menggunakan beberapa metode path finding yakni Breadth First Search (BFS), Dikstra, dan A Star (A\*), simulasi yang diperagakan akan menampilkan cara NPC melakukan path finding dalam beberapa maze yang sudah disiapkan penulis. Dalam similasi ini ada beberapa data yang harus di perhatikan yakni berapa banyak proses dilakukan dan waktu dibutuhkan untuk menyelesaikan sebuah maze. Banyaknya proses yang dilakukan dan waktu akan dihitung dari dimulainya NPC melakukan path finding dari node awal sampai NPC bergerak ke node akhir yang sudah ditentukan.

### B. Teknik Analisis Data

Penulis mendapatkan ilmu pengetahuan yang dapat digunakan pada penelitian yang sedang dijalankan lalu membuat kesimpulan vang telah dikumpulkan. data Kesimpulan tersebut didapatkan dari data yang dikumpulkan saat melakukan studi pustaka tentang implementasi algoritma A\* pada penelitian - penelitian terdahulu. Untuk mengukur manakah metode path finding yang lebih baik, maka data - data dari hasil eksperimen yang perlu diambil adalah waktu dan berapa banyak proses yang dilakukan, data - data ini akan diambil dari ketiga metode path finding yang digunakan dalam tiap *maze*. Dan akan dikomparasikan untuk menentukan manakah metode path finding terbaik.

### 3. Hasil dan Pembahasan

Dalam melaksanakan penelitian ini, peneliti mengikuti alur metode Waterfall dalam pembuatan program yang diinginkan. Pada bagian ini penulis merancang bagaimana sistem dibuat, perancangannya adalah sebagai berikut:

### A. Requirements Definition

Untuk menjalankan simulasi ini penulis membuat program dengan bahasa Javascript dan alat bantu Visual Studio Code serta library P5.js lalu dijalankan menggunakan Google Chrome 64bit versi 83.0.4103.116.

Penulis berencana membuat sebuah maze dalam bentuk grid, penulis juga akan membuat beberapa maze dengan rintangan yang berbeda – beda, saat pelaksanaan penulis juga akan melakukan perekaman sebagai bukti bahwa data hasil yang didapat valid.

Untuk menjalankan simulasi diperlukan algoritma path finding, di sini penulis menyiapkan 3 algortima path finding yakni: A Star (A\*), Dijkstra, dan Breadth First Search (BFS), ketiga algoritma ini akan dimasukkan kedalam maze yang sama agar data hasil dapat di komparasikan.

Cara kerja tiap algoritma dapat dilihat pada gambar berikut:

Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan

mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber

ungi Undang-Undang

Dilarang

а

### a. A Star

Cara kerja dari algoritma ini hampir sama dengan algoritma Dijkstra, akan tetapi dimodifikasi dengan fungsi heuristic.

Algoritma A\* memerlukan dua model antrian yaitu antrian open set dan juga antrian closed set. Modifikasi dari fungsi heuristic dari algoritma A\* dapat melakukan prediksi pada setiap node - node yang dibuat. Langkah ini dilakukan agar memudahkan algoritma tersebut untuk menentukan langkah - langkah selanjutnya yang di harapkan.

Mulai

Inisiasi Mik n' sebagai
Titik Awai dan
Gene List

Menghtung fungsi Cost
((n) = g(n)+ h(n)

No

Mengecak setap sektor dar 'n'
yang tidak termasuk ke
dalam Closed List, dan masuksan indeks
dari titik 'n' yang memiki nilai 't terkecii

Menghtung fungsi
cost Tunks estap

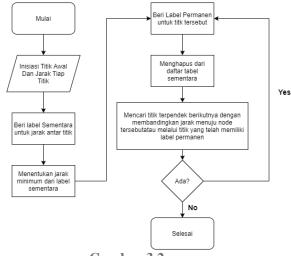
Menghtung fungsi
cost Tunks estap

Gambar Flowchart A Star

Gambar Flowchart A Star

# b. Dijkstra

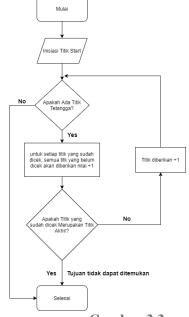
Algoritma Dijkstra akan memilih node sumber dan menetapkan kemungkinan label sementara (dianggap sebagai nilai tak terhingga) ke setiap node lainnya. Langkah berikutnya akan mencoba untuk meminimalkan biaya untuk setiap node. Biaya yang dimaksud disini adalah jarak atau waktu yang diambil untuk mencapai node tujuan dari node sumber.



Gambar 3.2 Gambar Flowchart Dijkstra

### c. Breadth First Search

Algoritma BFS akan melakukan pencarian rute terpendek secara menyeluruh atau dengan mengecek dari node awal ke semua node tetangganya secara terurut.



Gambar 3.3 Gambar Flowchart Breadth First Search

### B. Software and Design

Perancangan yang dibuat penulis adalah sebuah grid berukuran 20 x 20, posisi node awal, node akhir, dan rintangan akan ditentukan oleh penulis. Selama simulasi melakukan path finding dari node awal

Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan

Dilindungi Undang-Undang

sampai node akhir, sebuah path akan tercetak dan selalu update setelah mengunjungi sebuah grid, untuk berapa banyak proses yang dilakukan saat path finding akan tersimpan ke dalam sebuah variabel dan dicetak setelah mencapai node akhir

Penulis menggambarkan desain untuk simulasi seperti berikut: ak Cipta

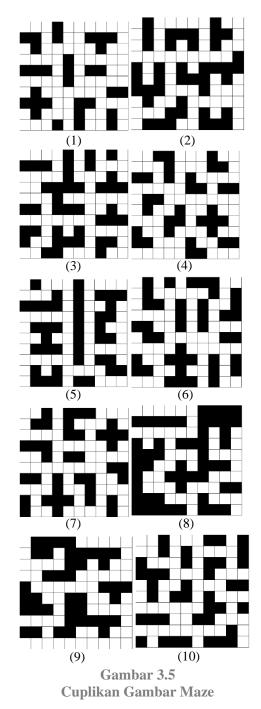
Closed Set Gambar 3.4 Gambar Desain Simulasi Maze Path **Finding** 

Dapat dilihat dari gambar di atas, node start dan akhir akan diletakkan sesuai dengan imputan penulis yakni ujung kiri atas (node awal) dan ujung kanan bawah (node akhir Untuk path akan dilambangkan dengan warna pink dan akan selalu berubah setiap pengecekan path baru, melambangkan path yang akan dicek dan path yang sudah dicek akan dilambangkan dengan warna hijau (open set) dan merah (closed set), dan rintangan dilambangkan dengan warna hitam.

### C. Implementation and Unit Testing

Algoritma A Star (A\*), Dijkstra, dan BFS dimasukkan ke dalam simulasi. Setiap simulasi akan memiliki node start dan node akhip-yang sama. Penulis membuat grid dalam fungsi Setup(). Dalam fungsi Setup() setelah mencetak grid, dia juga mencetak dimana node start dan node akhir dalam maze Saat maze sudah siap, path finding akan dapat dilakukan dengan menggunakan fungsi Spot() yang berfungsi mengecek tetangga grid. Dan penghitungan jalur terpendek akan dikerjakan dengan fungsi Draw() yang mneggunakan algoritma path finding (A\*, Dijkstra, dan BFS).

Pengujian akan dijalankan menggunakan 10 maze yang sudah dibuat oleh penulis, dalam percobaan ini penulis menghitung berapa lama dan berapa banyak langkah yang digunakan dalam path finding maze, maze - maze yang digunakan dapat dilihat sebagai berikut:





tanpa izin IBIKKG



Tabel 3.1

43

38

<u>=</u>0

Rata - Rata

Tabel berisikan data Langkah rata – rata yang dilakukan NPC dalam Maze

60

56

60

56

Maze -	A Star	Dijkstra	~
		Dijkstra	BFS
ije	0.420	0.700	0.700
1	Milidetik	Milidetik	Milidetik
	0.410	0.590	0.590
2	Milidetik	Milidetik	Milidetik
†s	0.370	0.520	0.520
i <b>țu</b> :	Milidetik	Milidetik	Milidetik
	0.380	0.670	0.670
şisr	Milidetik	Milidetik	Milidetik
	0.350	0.440	0.440
50	Milidetik	Milidetik	Milidetik
5	0.400	0.600	0.600
	Milidetik	Milidetik	Milidetik
0	0.420	0.620	0.620
3	Milidetik	Milidetik	Milidetik

8	0.380	0.420	0.420
	Milidetik	Milidetik	Milidetik
9	0.340	0.570	0.570
	Milidetik	Milidetik	Milidetik
10	0.440	0.610	0.610
	Milidetik	Milidetik	Milidetik
Rata	0.391	0.574	0.574
-	Milidetik	Milidetik	Milidetik
Rata			

Tabel 3.2 Tabel berisikan data Waktu rata – rata yang dilakukan NPC dalam maze

Dapat dilihat dari tabel 3.1 dan tabel 3.2 bahwa A\* dapat menyelesaikan maze dalam waktu paling sedikit dibandingkan dengan BFS dan Dijkstra yang hanya mampu menyelesaikan maze dalam waktu yang hampir bersamaan. Hal ini dikarenakan BFS dan Dijkstra memiliki metode pencarian rute yang cukup mirip, dengan perbedaan dimana Dijkstra mampu menetapkan jarak selain 1 nilai untuk setiap langkah dibandingkan BFS. Misalnya, dalam mencari rute jarak (atau bobot) dapat ditentukan kecepatan, biaya, dan preferensi grid yang di Algoritma tersebut kemudian memberikan jalur terpendek dari sumber node ke setiap node tetangga dalam grid yang dilintasi.

Sedangkan BFS pada dasarnya hanya memperluas pencarian dengan satu langkah (biaya) pada setiap iterasi, yang kebetulan memiliki efek menemukan jumlah langkah terkecil yang diperlukan untuk mencapai setiap node yang diberikan dari sumber node.

### D. Integration and System Testing

Setelah masa percobaan simulasi selesai, penulis membuat sebuah simulasi *path finding* dengan membuat *maze* secara acak yang dapat dijadikan sebagai *showcase* cara kerja *path finding*.

Hasil Implementasi dari penulis dapat dilihat dengan gambar berikut :

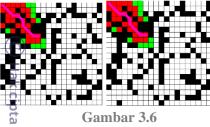
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun . Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber: b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG ۵ . Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan penulisan kritik dan tinjauan suatu masalah.

Dilarang mengutip sebagian atau seluruh karya tulis

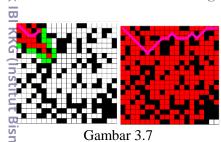
 $\equiv$ 

tanpa mencantumkan dan menyebutkan sumber:

Informatika Kwik Kia



**3Gambar Tangkapan Layar Saat** Berhasil Melakukan Path Finding



Gambar Tangkapan Layar Saat Gagal Melakukan Path Finding

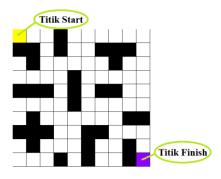
# Hak Cipta Dilindungi Undang-Unda**ng Operation** and Maintenance

Setelah menyelesaikan masa implementation dan integration ada suatu masalah yang sebelumnya terlewat oleh penulis yakni penanda start dan finish, sebelamnya penulis hanya menyatakan bahwa start ada di ujung kiri atas maze dan finishada di ujung kanan bawah *maze*.

Tampilan maze setelah implementation dan integration dapat dilihat sebagai berikut:

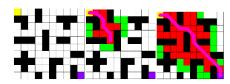


Dan, tampilan setelah diperbaiki penulis dapat dilihat sebagai berikut:



Gambar 3.9 Tampilan Maze Setelah Perbaikan

Pada perbaikan ini penulis melambangkan titik start dengan warna kuning dan titik finish dengan warna ungu, agar dapat mempermudah visualisasi simulasi yang dibuat penulis. Dengan perbaikan ini tampilan simulasi maze yang sudah diperbaiki akan terlihat sebagai berikut:



Gambar 3.10 Tampilan Maze Sebelum, Saat, dan **Sesudah Path Finding** 

### 4. Kesimpulan

Berdasarkan hasil analisis simulasi path finding dalam maze yang dibuat penulis, maka dapat ditarik kesimpulan sebagai berikut:

1. Simulasi ini dapat dijalankan menggunakan browser Google Chrome. Simulasi ini memuat 3 algoritma path finding yakni A Star (A\*), Dijkstra, dan Breadth First Search (BFS). Showcase ini menunjukkan performa algoritma algoritma path finding dalam menentukan rute terpendek untuk menyelesaikan maze, rintangan yang ada di dalam maze di buat secara acak dengan bantuan komputer, hasil performa tiap path finding dicatat dalam bentuk waktu dan berapa banyak proses yang dilakukan untuk menyelesaikan maze.



Dilarang

Hak Cipta

mengutip sebagian atau seluruh karya tulis

 $\equiv$ 

tanpa mencantumkan dan menyebutkan sumber:

2. Dalam percobaan simulasi yang diuji dengan menggunakan 10 maze yang dibuat oleh penulis dapat disimpulkan bahwa path finding algoritma A\* menghasilkan waktu yang ditempuh rata - rata adalah 0,391 milidetik, dengan waktu tercepat 0,340 milidetik pada *maze* tercepat 0,5 % ke 9 dan waktu terlambat 0,440 milidetik pada *maze* ke 10. Tetapi dan BFS, waktu yang ditempuh rata – rata adalah 0,574 milidetik, dengan waktu tercepat 0,420 milidetik pada maze ke 8 dan waktu terlambat 0,700 milidetik pada maze pertama. Hal ini terjadi karena algoritma A\* mencari jalur terpendek menggunakan

Dilindungi Undang-Und bantuan fungsi *heuristic*. Dari penelitian yang telah dilakukan dengan melakukan simulasi maze dengan algoritma – algoritma path finding, dapat disimpulkan bahwa algoritma path finding A Star (A\*) dapat mengoptimasikan path dibandingkan finding **NPC** dengan algoritma *path* finding lainnya yakni Dijkstra dan BFS

### 5. Saran

Dalam penelitian ini, terdapat beberapa kekurangan juga dalam proses pembuatannya. Oleh karena itu, saran yang dapat diberikan untuk penelitian yang selanjutnya adalah:

- 1. Membuktikan apakah algoritma A Star adalah algoritma path finding yang optimal paling dengan mengkomparasikannya dengan algoritma yang sejenisnya selain yang sudah dilakukan penulis.
- Membuktikan apakah ada metode lain intuk membuat rintangan dalam maze.
- Membuktikan apakah algoritma A Star dapat diimplementasikan ke dalam game dengan grafik 3D.
- Mengimplementasikan metode path finding bukan hanya untuk dalam game tetapi bisa juga diimplementasikan ke dalam bidang – bidang yang lainnya.

### 6. Ucapan Terima Kasih

Puji syukur saya ucapkan kepada Allah SWT. berkat karena atas dan pertolongannya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka untuk memenuhi salah satu syarat untuk mencapai gelar sarjana jurusan Teknik Informatika pada Institut Bisnis dan Informatika Kwik Kian Gie. mengerjakan karya skripsi ini banyak sekali rintangan yang harus saya hadapi, tanpa bantuan dari berbagai pihak, sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih yang sebesar-besarnya kepada:

- Ridzal Saiful Zein ayah tercinta, Everdin MP ibu tercinta, Indriadin MP tercinta. Yuli Pramono pamanku tercinta, dan keluarga yang telah memberikan doa dan dukungan penuh selama saya mengerjakan skripsi ini.
- Bapak Richard Vinc N. Santoso, S.TI., M.T.I. selaku dosen pembimbing yang telah menyediakan waktu dan tenaga membimbing saya dalam untuk menyelesaikan skripsi ini.
- Bapak Yunus Fadhilah Soleman. S.Kom., M.Kom. selaku dosen pembimbing pada saat awal saya melakukan bimbingan skripsi.
- Rodo Maxmilano selaku teman dekat dan seperjuangan dalam menyelesaikan memberikan skripsi yang telah dukungan dan bantuan
- Adbert Lijanto selaku teman dekat yang memberikan saaya dukungan dan bantuan saat megerjakan skripsi ini
- Rhein Michael selaku teman dekat dan seperjuangan dalam menyelesaikan skripsi yang telah memberikan dukungan dan bantuan
- Filemon selaku teman dekat dan menyelesaikan seperjuangan dalam skripsi memberikan yang telah dukungan dan bantuan
- Teman-teman dekat yang tidak dapat disebut satu persatu yang selalu memberikan semangat.



Dilarang

۵

- 9. Teman-teman iurusan Teknik Informatika angkatan 2016 yang telah berjuang bersama-sama saat melakukan perkuliahan.
- 10. Teman-teman jurusan lain maupun

- 10. Teman-teman jurusan lain maupun teman luar kampus yang tidak dapat disebut satu persatu yang telah memberikan semangat dan bantuan.

  Hak Senoga Tuhan membalas segala kebaikan yang telah mereka berikan kepada saya, baik berupa moral maupun material.

  DAFTAR PUSTAKA

  Tahundang Undang Pamungkas, E. P. (2014). Penerapan Algoritma A\* (A Star) Pada Game Edukasi The Maze Island Berbasis Android. Jurusan Teknik Informatika, STMIK GI MDP, Palembang.

  Algoritma A Star (A\*) pada Game Algoritma A Star (A\*) pada Game
- penelitian, pege 57 63.

  Binanto, J. (2014). Algoritma A Star (A\*) pada Game Petualangan Labirin Berbasis Android. Jurnal Ilmu Komputer dan Informatika,
- ANALISA **METODE** CLASSIC LIFE CYCLE

  (WATERFALL) UNTUK

  PENGEMBANGAN PERANGKAT

  LUNAK MULTIMEDIA. Universitas

  Sanata Dharma, Yogyakarta, Indonesia.

  Blasius Neri Puspika, A. R. (2012). Implementasi
- penyusunan lapo an Algoritma Dijkstra Daram penentuan Jalur Terpendek Yogyakarta Di Menggunakan GPS Dan Qt Geolocation. Informatika: Jurnal Teknologi Komputer dan Informatika, Page 141 - 149.
  - Kim, S.-M. J.-T.-G. (2006). Verification of FSM using Attributes Definition of NPCs. Dong University Digital Contents Cooperation Research Center, Department of Digital Contents. Dong Shin Unversity, Department of Computer Science, Chonnam National University, Page 168 **-1**74. ika Kwik Kia

- Dalem, I. B. (2018). Penerapan A Star (A\*) Menggunakan Graph Untuk Menghitung Jarak Terpendek. Program Studi Ilmu Komputer, Pasca Sarjana Universitas Pendidikan Ganesha, Singaraja - Bali, Page 41 - 47.
- Erniyati, M. K., & Multayi, M. K. (2019). Pencarian Jalur Terdekat Menuju Rumah Di Kota Bogor Dengan Manggunakan Algoritma A\*. Komputasi (Jurnal Ilmiah Ilmu Komputer dan Maematika), Page 235 - 253.
- Fernando, Y., Mustaqov, M. A., & Megawaty, D. A. (2020). Penerapan Algoritma A-Star Pencarian **Aplikasi** Lokasi Fotografi Di Bandar Lampung Berbasis Android. Jurnal TEKNOINFO, Page 27 -
- Ian Millington, J. F. (2009). Artificial Intelligence for Games 2nd Edition. Amsterdam: Morgan Kaufmann Publishers.
- Lubis, E. S. (2015). SISTEM PENGANTARAN **MAKANAN DENGAN** PENDAYAGUNAAN **VEHICLE** MENGGUNAKAN GEOGRAPHICAL INFORMATION SYSTEM (GIS) DAN **ALGORITMA** Α **STAR** (A\*).AKULTAS ILMU KOMPUTER DAN **TEKNOLOGI INFORMASI** UNIVERSITAS SUMATERA UTARA.
- Muhammad Khoirudin Harahap, N. K. (2017). Pencarian Jalur Tependek Dengan Algoritma Dijkstra. Jurnal & Penelitian Teknik Informatika, Page 18 - 23.
- Pressman, R. S. (2010). Software Engineering: A Practitioner's Approach, 7th Edition. New York: McGraw-Hill.
- Rachmah, N. F. (2008). Aplikasi Algoritma Dijkstra dalam Pencarian Lintasan Terpendek Graf. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika.
- Rakhmat Kurniawan. R., S. M. (2016). Penerapan Algoritma A\* (A Star) Sebagai Solusi Pencarian Rute Terpendek Pada Maze. Program Studi Ilmu Komputer, Fakultas

penulisan kritik dan tinjauan suatu masalah

untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan

Sains dan Teknologi Universitas Islam Negeri Sumatera Utara.

Reddy, H. 2013). PATH FINDING - Dijkstra's 1. Di and A\* Algorithm's. а

Reddy, H. (2013). PATH FINDING - Dijkstra's and A\* Algorithm's.

Right A\* Algorithm's.

Right Tahara Shita, S. (2016). IMPLEMENTASI ALGORITMA BFS (BREADTH-FIRST SEARCH) PADA APLIKASI WEB CRAWLER. Jurnal TELEMATIKA MKOM, Page 127 - 132.

BY DENERAPAN ALGORTIMA A\* DENYELESAIAN RUTE TERPENDEK PENDISTRIBUSIAN BARANG. Jurusan Matematika, Fully, Universitas Negeri Semarang, Indonesia, Page 7 - 12.

BY DALAM PENYELESAIAN RUTE TERPENDEK PENDISTRIBUSIAN BARANG. Jurusan Matematika, Fully, Universitas Negeri Semarang, Indonesia, Page 7 - 12.

BY DALAM PENYELESAIAN RUTE TERPENDEK PENDISTRIBUSIAN BARANG. Jurusan Matematika, Fully, Universitas Negeri Semarang, Indonesia, Page 7 - 12.

BY DALAM PENYELESAIAN RUTE TERPENDEK PENDISTRIBUSIAN BARANG. Jurusan Matematika, Fully, Universitas Negeri Semarang, Indonesia, Page 7 - 12.

BY DALAM PENYELESAIAN RUTE TERPENDEK PENDISTRIBUSIAN BARANG. Jurusan Matematika, Fully, Universitas Negeri Semarang, Indonesia, Page 7 - 12.

Royce, D. W. (1970). MANAGING THE **DEVELOPMENT** OF **LARGE** SOFTWARE SYSTEMS. The Institue of Electrical and Electronic Engineers, Page 328 - 338.

Sandy Purnama, D. A. (2018). PENERAPAN ALGORITMA A STAR (A\*) UNTUK PENENTUAN JARAK TERDEKAT **WISATA KULINER KOTA** BANDARLAMPUNG. Informatika, Universitas Teknokrat Indonesia, Page 28 - 32.

Sommerville, I. (2011).Software Engineering - Ninth Edition. Boston: Addison-Wesley.

# Institut Bisnis dan Informatika Kwik Kian