

Penerapan Algoritma Dijkstra Pada Game Learning Matematika Berbasis Android

John Adler^{1*}, Bagas Farid Ramadhan²

¹⁾Program Studi Teknik Komputer, Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia
Jl. Dipati Ukur No. 112 - 116, Bandung, Indonesia 40132

²⁾ Program Studi Sistem Komputer, Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia
Jl. Dipati Ukur No. 112 - 116, Bandung, Indonesia 40132

* email: john.adler@email.unikom.ac.id

(Naskah masuk: 10 Februari 2021; diterima untuk diterbitkan: 16 Agustus 2021)

ABSTRAK Sebuah lembaga survey di bidang game. Newzoo menempatkan Indonesia di peringkat ke-16 dunia dalam hal penggunaan game yang cukup diminati oleh rentang usia produktif 10 - 20 tahun. Untuk itulah perlu adanya proses pembelajaran dan pendidikan terhadap game agar anak-anak di usia tersebut tidak kecanduan game yang akan merusak sel otak. Maka dari itu dengan membangun serta merancang game maze yang memiliki nilai edukasi yang sangat baik untuk anak-anak, maka dari itu digunakan algoritma Dijkstra, sebagai sistem kecerdasan buatan untuk mencari jalan tercepat menuju suatu lokasi. Yang akan melatih otak anak-anak untuk berfikir dan berkembang agar nantinya anak-anak tidak tergantung lagi pada permainan yang bersifat merusak. Dengan menggunakan kusioner yang telah dirancang maka dapat diambil kesimpulan bahwasannya dari 18 orang anak dari rentang kelas 5 SD sampai kelas 1 SMP yang memainkan game ini 88.9% diantaranya merasa game ini sudah sesuai dengan materi yang mereka pelajari, kemudian game ini juga dirasa menyenangkan dan mereka ingin memainkannya lagi. Yang artinya game ini dapat digunakan sebagai alternatif pembelajaran disekolah.

Kata Kunci - Algoritma Dijkstra, game maze, Artificial Intelligence, Android Games, Matematika.

Application of Dijkstra Algorithm to Android Based Mathematics Learning Game

ABSTRACT - A survey institute in the field of gaming. Newzoo ranks Indonesia 16th in the world in terms of game usage that is in high demand by the productive age range of 10-20 years. To consider the need for learning and educational processes for games so that children at this age will not like games that will damage brain cells. Therefore by building and supporting maze games that have excellent educational value for children, then using the Dijkstra algorithm, as an artificial intelligence system to find the fastest way to get to a location. Which will train children's brains to think and develop so that children are not dependent anymore on destructive games. By using the black box method, it can be concluded that 18 children from the range of grade 5 to grade 1 of junior high who play this game 88.9% agree this game is in accordance with the material they intended, then this game is also fun and they want to play it again. This meaningful game can be used as an alternative to school learning.

Keywords - Algoritma Dijkstra, Maze Game, Artificial Intelligence, Android Games, Math.

1. PENDAHULUAN

Pada saat ini game sangat digandrungi oleh berbagai kalangan dari anak-anak sampai orang tua. Pada masa sekarang game juga sudah bukan lagi sebuah permainan biasa namun game juga sudah

menjadi sebuah usaha ataupun mata pencaharian. Menurut Newzoo, sebuah lembaga survey game di dunia pada tahun 2017, Indonesia menempati urutan ke-16, dalam hal pembelian game [1]. Hal inilah yang membuat adanya penelitian untuk mejadikan games bukan hanya sebagai permainan yang

didalamnya hanya ada menang dan kalah tetapi juga ada nilai pendidikan didalamnya.

Mengapa menggunakan game sebagai metode pembelajaran, karena menurut Virvou game learning dapat memotivasi pemain untuk belajar, sehingga proses belajar akan terasa lebih menyenangkan [2]. Sama halnya dengan hasil penelitian yang dilakukan oleh Randel pada tahun 1991, tercatat bahwa pemakaian game dapat memberikan manfaat terutama pada materi-materi yang berhubungan dengan hitungan dan bahasa. Bahkan Pivec membuktikan bahwa game learning berhasil diterapkan untuk pendidikan formal khususnya di akademi militer, ilmu kedokteran, dan ilmu fisika [3]. Game learning juga dapat diterapkan pada berbagai genre game lain, dan hal itulah yang membuat game learning dapat digunakan sebagai sarana alternatif pembelajaran.

Selain itu didalam pembangunan sebuah game yang tidak kalah penting untuk diperhatikan adalah penerapan Kecerdasan buatan didalam game tersebut. Ada beberapa kecerdasan buatan yang sering dipakai dalam game yaitu *Algoritma Best- Path Planning*, *Breadth First Search(BFS)*, *A**, dan juga *Dijkstra*. Selain untuk game algoritma-algoritma tadi juga dapat digunakan sebagai kecerdasan untuk mencari jalan tercepat untuk menuju lokasi. Karena Algoritma ini baru berhenti ketika mendapatkan solusi yang terbaik [4]. Menurut Selvia Lorena Br Ginting, Yogie Rinaldy Ginting, dan Tulus Prabudi bila jarak yang ditempuh terlalu jauh maka algoritma ini tidak begitu akurat namun bila pada jarak dekat algoritma ini cukup presisi [5]. Selanjutnya, Menurut Anwari dan Hozairi Algoritma *BFS* mempunyai kelebihan dalam masalah kecepatan dibandingkan dengan algoritma *Dijkstra* namun ketelitiannya tidak lebih baik dari *Dijkstra* karena Algoritma *Dijkstra* dapat menjangkau lebih banyak tempat [6]. Kemudian Menurut Arif Cahyo, dan maful prayoga algoritma *A** memiliki laju yang lebih cepat dan lebih baik dari *dijkstra* [7]. Lalu menurut percobaan yang dilakukan Silvester Dian Handi permana antara algoritma *A**, *BFS*, dan *Dijkstra* untuk diterapkan pada game terutama game bertipe maze lebih baik menggunakan algoritma *A** namun bila bloknnya lebih banyak maka algoritma *Dijkstra* lebih baik dan itu semua tergantung pada aplikasi apa yang akan dipakai [8]. Selain itu Algoritma *Dijkstra* ini juga memiliki sisi pembelajaran secara tidak langsung salah satunya adalah mencari jalan keluar yang melatih otak untuk berfikir. Maka dari itu dipilihlah algoritma *dijkstra* sebagai *Artificial Intelligence* pembantu pada game ini.

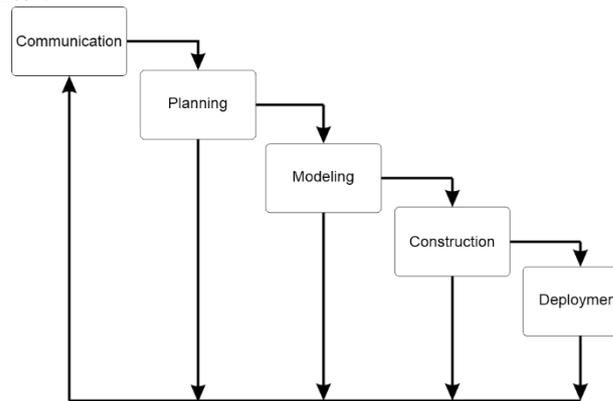
Maka dari itu *game* terutama *game maze* yang didalamnya terdapat Algoritma *Dijkstra* dapat digunakan sebagai sarana alternatif pembelajaran diluar sekolah.

2. METODE DAN BAHAN

Metode

Metode yang digunakan dalam penerapan algoritma *Dijkstra* pada game learning matematika ini menggunakan metode waterfall. Metode waterfall memiliki tahapan-tahapan yang mendasari alur pengembangan aplikasi yang akan dibangun.

Beberapa tahapan dari metode *waterfall* yang diterapkan pada pengembangan algoritma *Dijkstra* pada game learning matematika berbasis android, yaitu :



Gambar 1 Metode Waterfall [9]

1. Communication

Pada tahapan ini dilakukan analisa terhadap keperluan aplikasi, penerapan karakteristik aplikasi dengan melakukan sebuah wawancara dengan calon pengguna.

2. Planning

Tahap ini dilakukan sebelum melakukan pemrograman pada aplikasi. Tahap ini bertujuan agar dapat memberikan gambaran dari sesuatu yang seharusnya dikerjakan dan bagaimana tampilannya. Tahap ini membantu dalam menspesifikasikan kebutuhan hardware dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan..

3. Modeling

Di tahap ini dilakukan penggabungan dari sejumlah objek yang sudah dibuat. Kemudian dilakukan pengujian hal ini dilakukan untuk mengetahui apakah perangkat lunak yang dibuat telah sesuai dengan desainnya.

4. Construction

Seluruh kebutuhan perangkat lunak harus bisa didapatkan pada fase ini, termasuk didalamnya manfaat dari perangkat lunak yang diharapkan sesuai dengan pengguna dan batasan perangkat lunak yang dibangun. Informasi ini dapat diperoleh melalui wawancara, survey atau diskusi. Informasi tersebut kemudian dianalisis untuk mendapatkan hasil yang sesuai dengan kebutuhan pengguna untuk digunakan pada tahap selanjutnya.

5. Deployment

Ini merupakan tahap terakhir dalam model waterfall. Perangkat lunak yang sudah dibangun kemudian dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki *bug* yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

Perancangan Sistem

Dari permasalahan yang dihadapi, maka dibangunlah game dengan implementasi *Artificial intelligence* berupa algoritma dijkstra untuk membantu proses pembelajaran matematika melalui telepon genggam berbasis android. Aplikasi yang dibuat ini diharapkan dapat membantu dan memotivasi belajar terutama pada bidang matematika.

Komponen dan Spesifikasi

Dalam membangun aplikasi ini, dibutuhkan beberapa komponen berupa hardware, software, dan algoritma pembantu yaitu Dijkstra. Komponen untuk membangun game dapat dilihat pada Tabel 1

Tabel.1 Tabel Komponen Pembangun aplikasi

No.	Nama	Fungsi pada Game
1	UNITY 3D	Sebagai wadah untuk membangun game dan mengolah objek yang akan digunakan
2	Android	Sistem operasi untuk menjalankan game tersebut melalui smartphone setelah game terinstall
3	Algoritma Dijkstra	Sebagai kecerdasan buatan pembantu dalam memainkan game ini.

Dalam pengimplementasiannya sistem ini harus menggunakan piranti yang mendukung kemampuan sistem agar dapat berjalan dengan lancar dan sesuai. Berikut ini adalah spesifikasi minimum untuk mendukung kemampuan proses pada game ini :

1. Spesifikasi perangkat lunak

Spesifikasi perangkat lunak yang dapat dipakai untuk mengoperasikan sistem ini adalah Sistem operasi *android 8 Oreo* dengan spesifikasi minimum adalah *android 4.4*

Kitkat.

2. Spesifikasi perangkat keras

Spesifikasi perangkat keras yang dapat digunakan sebagai media pembantu untuk pembuatan sistem ini ialah : Processor intel Core i3 Sandybridge, Harddisk 512 GB, SSD

240GB, RAM 6144 MB, VGA intel HD graphic 4000 1792 MB, LCD monitor dengan resolusi 1920x1080, Keyboard dan mouse.

Algoritma Dijkstra

Algoritma Dijkstra adalah algoritma untuk mencari lintasan terpendek dari suatu lintasan. Algoritma ini menggunakan prinsip *Greedy*, yang menyatakan bahwasannya pada setiap langkah yang diambil akan memilih sisi yang berbobot paling kecil dan memasukkannya ke dalam himpunan solusi. [10]

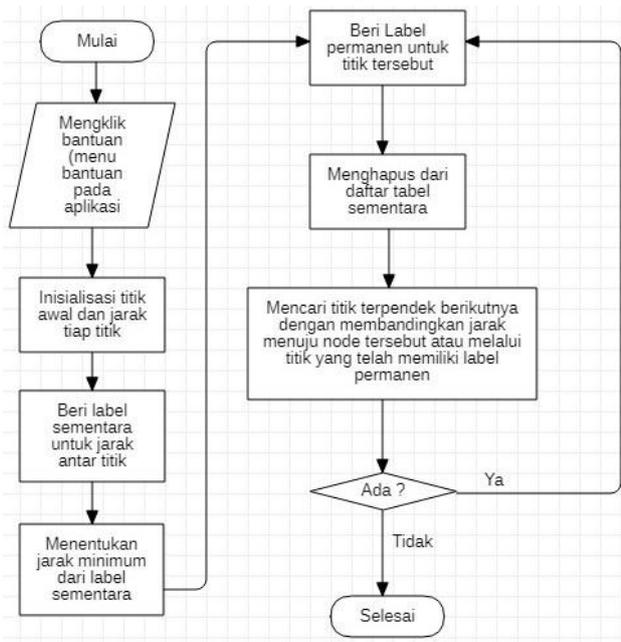
Input algoritma ini adalah sebuah graph berarah yang berbobot (weighted directed graph) G dan sebuah sumber vertex s dalam G dan V adalah himpunan semua vertices dalam graph G [11]

Berikut *Pseudocode* dari algoritma Dijkstra dalam mencari rute terpendek pada sebuah graf [12]:

```

algorithm DijkstraShortestWeightedPath( $G, s$ )
  _pre-cond_:  $G$  is a weighted (directed or undirected) graph, and  $s$  is one of its nodes.
  _post-cond_:  $\pi$  specifies a shortest weighted path from  $s$  to each node of  $G$ , and  $d$  specifies their lengths.
  begin  $d(s) = 0, \pi(s) = \_$ 
  for other  $v, d(v) = \infty$  and  $\pi(v) = \text{nil}$ 
  handled =  $\emptyset$ 
  notHandled = priority queue containing all nodes.
  Priorities given by  $d(v)$ .
  loop
  _loop-invariant_: See above.
  exit when notHandled =  $\emptyset$ 
  let  $u$  be a node from notHandled with smallest  $d(u)$ 
  for each  $v$  connected to  $u$ 
  foundPathLength =  $d(u) + w_{uv}$ 
  if  $d(v) > \text{foundPathLength}$  then
   $d(v) = \text{foundPathLength}$ 
   $\pi(v) = u$ 
  (update the notHandled priority queue)
  end if
  end for
  move  $u$  from notHandled to handled
  end loop
  return  $\_d, \pi\_$ 
end algorithm.
    
```

Untuk flowchart algoritmanya maka akan seperti pada Gambar 2



Gambar 2 Flowchart Algoritma Dijkstra [13]

Rancangan Tampilan Aplikasi

Pada tahap ini dilakukan diperlukan adanya rancangan aplikasi yang bertujuan untuk memberikan representasi dari apa yang seharusnya dikerjakan dan bagaimana tampilannya. Tahap ini akan membantu dalam merincikan kebutuhan hardware dan sistem serta mendefinisikan arsitektur sistem secara utuh [14]. Adapun yang menjadi tampilan perancangan sistem adalah sebagai berikut:

Tabel 2 Rancangan Tampilan Aplikasi

Learning the Power of Number



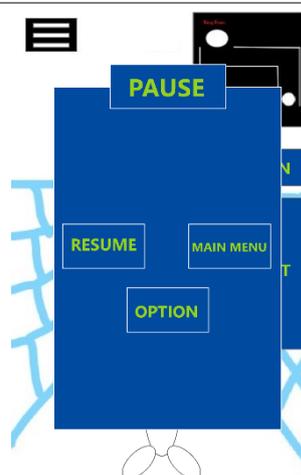
Keterangan :

1. Terdapat judul (Belajar Matematika)
2. Button Bermain
3. Button Credit (Untuk membuka credit scene)
4. Button Keluar



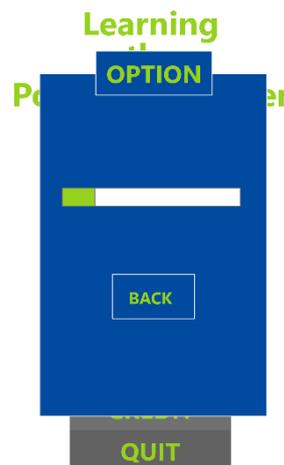
Keterangan :

1. Berisi tampilan saat game dimulai
2. "Gambar garis tiga masuk ke menu Option"
3. Gambar dipojok kanan atas adalah minimap (map kecil yang memperlihatkan a n pemain dari atas)
4. Tombol bantuan
5. Baloon soal



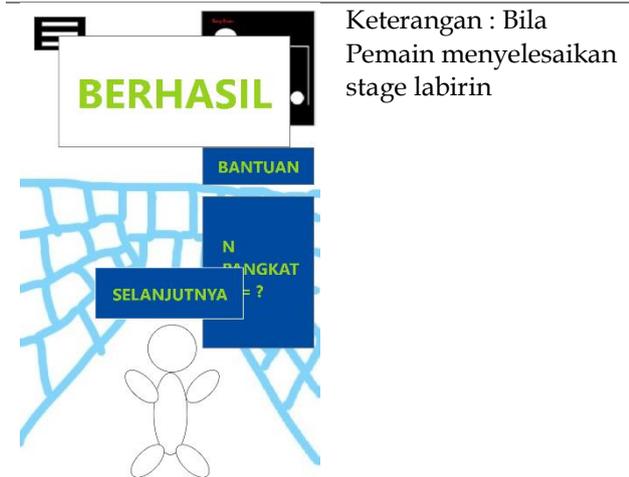
Keterangan :

1. Berisi tampilan menu pause
2. Resume button untuk membuat pemain kembali bermain
3. Option button Main menu button

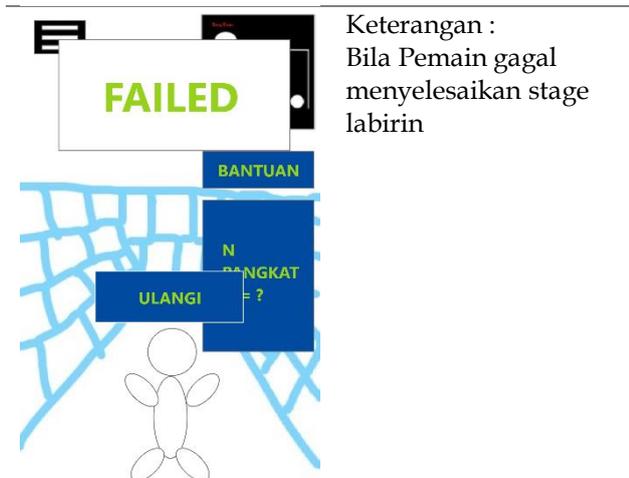


Keterangan :

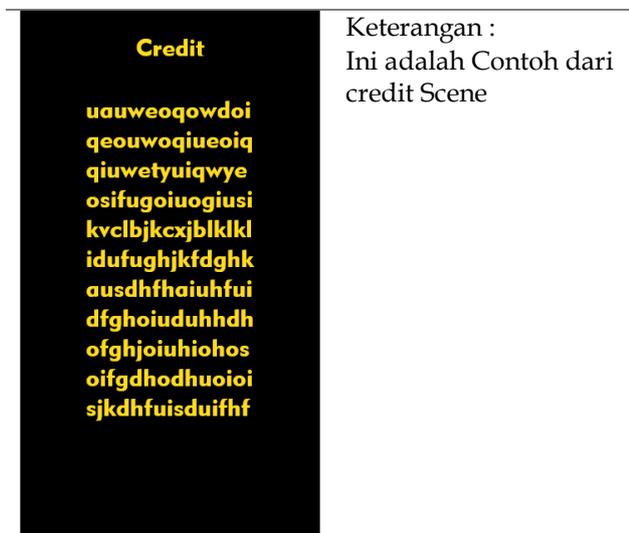
1. Menu Option
2. Slider untuk mengatur volume
3. Button back untuk kembali ke menu pause / main menu



Keterangan : Bila Pemain menyelesaikan stage labirin



Keterangan : Bila Pemain gagal menyelesaikan stage labirin



Keterangan : Ini adalah Contoh dari credit Scene

membuka game tersebut. Tampilan antarmuka menu utama dapat dilihat pada gambar 3



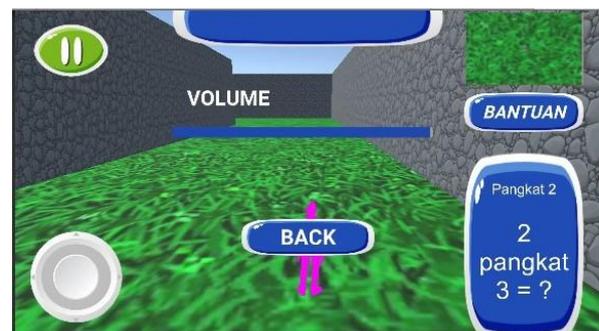
Gambar 3 Antarmuka Menu utama

Antarmuka Menu Option

Antarmuka menu option merupakan sebuah tampilan untuk pemain agar pemain dapat mengatur volume dari suatu permainan. Antarmuka menu option dapat dilihat pada gambar 4 dan 5, di mana pada gambar 4 sistem akan menampilkan menu option pada saat pemain sedang di main menu dan gambar 5 menampilkan menu option ketika pemain sedang berada dalam permainan.



Gambar 4 Antarmuka Menu Option di Main menu



Gambar 5 Antarmuka Menu Option didalam permainan

Antarmuka menu credit

Antarmuka menu credit merupakan sebuah tampilan yang menunjukkan pembuat game dan pembuat assetnya.

3. HASIL DAN PEMBAHASAN

Tampilan Game

Antarmuka menu utama

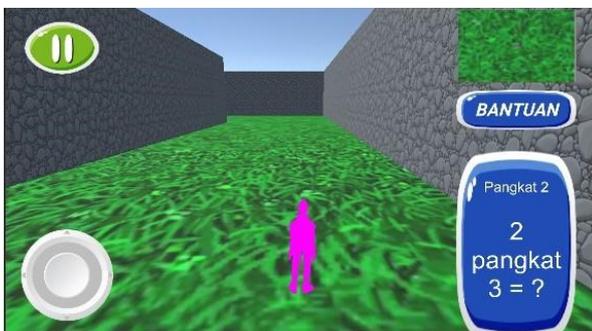
Antarmuka menu utama merupakan suatu tampilan awal pada saat pemain pertama kali



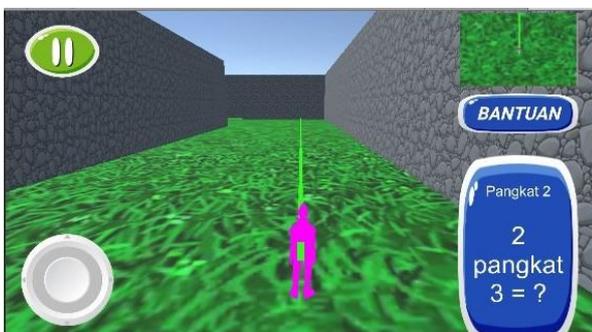
Gambar 6 Antarmuka Menu Kredit

Gameplay

Tugas utama pemain dalam saat bermain game ini adalah dengan cara mencari angka yang sesuai dengan soal yang diberikan. Lalu pemain diberikan 10 level dimana disetiap levelnya memiliki pertanyaan yang berbeda beda. Apabila pemain berhasil menyelesaikan satu level maka akan ada bacaan *Great* yang ada pada gambar 9 yang berarti berarti pemain akan diarahkan ke level selanjutnya, namun apabila pemain gagal di level berapapun maka akan ada tulisan *Failed* yang ada pada gambar 10 yang berarti pemain tersebut akan kembali ke level 1. Kemudian bila pemain tersebut sudah menyelesaikan kesepuluh level tersebut maka akan ada bacaan *THE END* yang ada pada gambar 11 yang berarti game tersebut telah selesai. Selanjutnya pemain bisa memilih untuk tetap bermain (kembali ke level 1) atau masuk ke main menu.



Gambar 7 Tampilan awal menu play



Gambar 8 Tampilan setelah menekan tombol bantuan



Gambar 9 Tampilan saat pemain berhasil menyelesaikan level



Gambar 10 Tampilan saat pemain gagal menyelesaikan level



Gambar 11 Tampilan saat pemain berhasil menyelesaikan permainan

Pengujian

Pengujian ini berfungsi sebagai pencari kesalahan pada program yang tidak berjalan dengan baik.

Pengujian Blackbox

pengujian black box adalah menguji kegunaan khusus yang terdapat pada *software* yang telah dibangun, menguji hasil *output* tanpa perlu melihat proses untuk mendapatkan *output* tersebut dan mengetahui kelebihan dari perangkat lunak dalam memenuhi kebutuhan *user* [15]. Hasil dari pengujian dapat dilihat pada table 3 di bawah ini

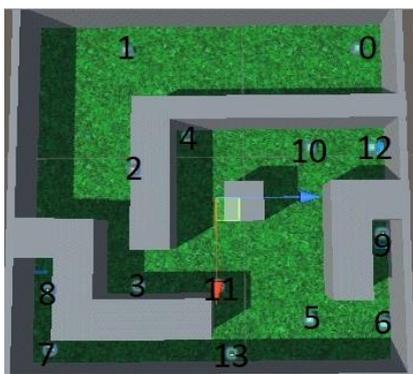
Tabel 3 Tabel Pengujian Black Box

No.	Kelas Uji	Detail Pengujian	Jenis Pengujian	Hasil Pengujian
1	Uji Antarmuka Menu Utama	Menekan Tombol Mulai	Black Box	Berhasil

2	Uji Sound Pada latar	Mengubah setting Volume	Black Box	Berhasil
3	Uji Pergerakan Pemain	Menggerakkan Pemain	Black Box	Berhasil
4	Uji Pertanyaan	Menjawab Pertanyaan	Black Box	Berhasil
5	Uji Tombol Bantuan	Menekan Tombol Bantuan	Black Box	Berhasil
6	Uji Tombol Pause	Menekan Tombol Pause	Black Box	Berhasil
7	Uji Tombol Setting	Menekan Tombol Setting	Black Box	Berhasil

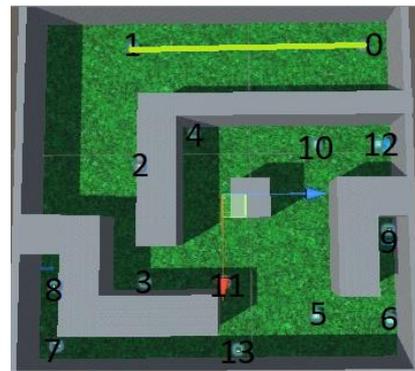
Analisis Algoritma Dijkstra

Analisa adalah penerapan metode sampai perhitungan matematisnya dikaitkan dengan model penelitian untuk memperlihatkan ketertarikan antar variable yang diteliti serta model matematis untuk analisisnya.[16] Kemudian Analisa Algoritma Dijkstra dilakukan pada game the power of number pertama inialisasikan node 0(-20.69,-10.98) sebagai node awal dan 12(-2.9,-9.1) adalah node tujuan. Kemudian cari node terdekat pada gambar 13 node terdekat terdapat pada node 1(-20.69,-51.6). Karena node 1 adalah satu satunya node terdekat



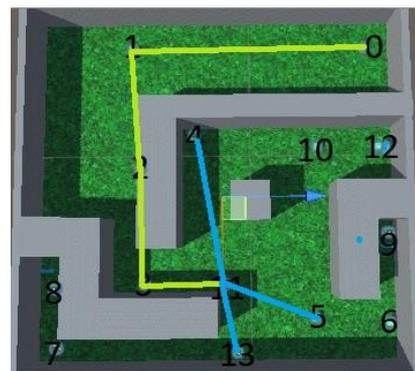
Gambar 12 Tampilan node level 1 pada game

Setelah diketahui bahwa node 1 merupakan note terdekat maka algoritma dijkstra akan langsung memberikan garis permanen ke node 1. Seperti pada gambar 12.



Gambar 13 Garis permanen dari node 0 ke node 1

Kemudian dari node 1 akan mencari node yang paling dekat untuk sampai ke tujuan yaitu node 2(0.6,-50.2). Karena tidak ada perbandingan lagi maka algoritma dijkstra akan memberikan garis permanen dari node 1 ke node 2. Kemudian node 3(20.5,-49.2) dan 11(20.5,-35,7) pun sama seperti node 2 karena tidak ada perbandingan lain maka node 2 akan memberikan garis permanen ke node 3 dan node 3 ke node 11. Dari node 11 ada 3 perbandingan yaitu node 11 ke node 4(-5.1,-40.8), node 5(20.2,-20.4) dan node 13(31.8,-33.5) seperti pada gambar 14.



Gambar 14 Perbandingan tiga node terdekat dari node sebelumnya

Kemudian Ketiga node tadi dibandingkan dengan node sebelumnya dan node tujuan. Selanjutnya dihitung jarak terdekatnya dengan menggunakan rumus jarak. Setelah dihitung jaraknya seperti pada tabel 4

Tabel 4 Tabel perbandingan antara node 4,5 dan 13

Node Selanjutnya	Titik Node	Selisih dengan Node Akhir (node 12(-2.9,-9.1))
4	(-5.1,-40.8)	27.5
5	(20.2,-20.4)	17.5
13	(31.8,-33.5)	37.8

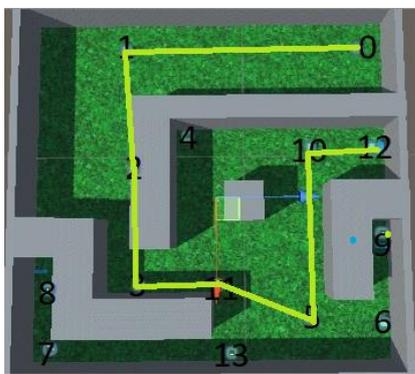
Setelah itu didapatkan hasilnya yaitu node 5 adalah node yang paling dekat bila dibandingkan dengan node tujuan (titiknya lebih mendekati nilai 0 dibanding node lainnya). Kemudian node 5 diberikan

garis permanen. Setelah itu dari node 5 ada dua perbandingan yaitu node 13 dan node 10. Setelah dihitung jaraknya seperti pada tabel 5:

Tabel 5 Tabel perbandingan antara node 13 dan 10

Node Selanjutnya	Titik Node	Selisih dengan Node Tujuan (node 12(-2.9,-9.1))
13	(31.8,-33.5)	42.4
10	(-2.9,-19.6)	10.5

Kemudian didapatkan hasilnya yaitu node 10 lebih dekat ke tujuan. Dari node 10 ada 3 perbandingan yaitu dari node 10 ke node 5, node 4, dan node 13. Karena node 12 adalah tujuannya maka dari node 10 akan ditarik garis permanen ke node 12 seperti pada gambar 14.



Gambar 15 Tampilan saat Algoritma ketika sampai ke node tujuan

Kesimpulannya adalah dari hasil hitungan dengan pengamatan secara visual hasilnya sama. Dimana tiap node yang dihitung dengan menggunakan rumus jarak sama dengan hasil node yang diambil. Oleh karena itu tombol bantuan pada game dapat membantu pemain untuk menemukan jalur tercepat menuju tujuan.

4. KESIMPULAN

Berdasarkan hasil kesimpulan dari penelitian ini, maka penerapan algoritma dijkstra pada game learning matematika berbasis android dapat digunakan sesuai dengan tujuannya, yaitu aplikasi ini dapat difungsikan sebagai saran alternatif pembelajaran.

Adapun masukan yang dapat digunakan sebagai kelanjutan dari penelitian ini adalah sebagai berikut: Game ini sebaiknya ditambahkan tingkat kesulitannya seperti menambah levelnya ke beberapa tingkatan seperti mudah, sedang, dan sulit, tampilan menu utama pada gamenya sebaiknya diberikan animasi agar terlihat lebih menarik, dan dapat dimainkan secara online.

DAFTAR PUSTAKA

- [1] Newzoo. (2017). "The Indonesian Gamer | 2017". Dipetik Februari 9, 2019, dari internet: <https://newzoo.com/insights/infographics/th-e-indonesian-gamer2017/> Diakses 1 Februari 2021.
- [2] M. Vivrou, "Combining Software Games with Education: Evaluation of Its Educational Effectiveness". *Journal Educational Technology and Society*, vol. 8, no. 2, pp. 54-65, 2005.
- [3] C. Solehudin, "Pembuatan Game Edukasi Strategi Bajak laut Untuk Anak-Anak Dengan Menggunakan Macromedia Flash 8.0," Skripsi Sarjana, Surakarta: Universitas Muhammadiyah Surakarta, 2012.
- [4] M. H. H. Ichsan, E. Yudaningsy dan M. A. Muslim, "Solusi Optimal Pencarian Jalur Tercepat dengan Algoritma Hybrid Fuzzy-Dijkstra," *Jurnal EECCIS*, vol. 6, no. 2, pp. 155-160, 2012.
- [5] S. L. B. Ginting, Y. R. Ginting dan T. Prabudi, "Penerapan Algoritma Best-Path Planning untuk Aplikasi Pencarian Rute Transportasi Publik Berbasis Android," *Jurnal Teknologi dan Informasi (JATI)*, vol. 5, no. 2, pp. 49-62, 2015.
- [6] Anwari dan Hozairi, "Perbandingan Algoritma Breadth First Search dan Dijkstra untuk Penentuan Rute Terpendek Pengiriman Barang Unilever," *Jurnal Mnemonic*, vol. 2, no. 1, pp. 67-72, 2019.
- [7] A. C. Prasetyo, M. P. Arnandi, H. S. Hudnanto dan B. Setiaji, "Perbandingan Algoritma Astar dan Dijkstra Dalam Menentukan Rute Terdekat," *Jurnal Ilmiah SISFOTENIKA*, vol. 9, no. 1, pp. 36-46, 2019.
- [8] S. D. H. Permana, K. B. Y. Bintoro, B. Arifitama dan A. Syahputra, "Comparative Analysis of Pathfinding Algorithms A *, Dijkstra, and BFS on Maze Runner Game," *International Journal Of Information System & Technology*, vol. 1, no. 2, pp. 1-8, 2018.
- [9] R. S. Pressman dan B. R. Maxim, *Process Models*, USA: McGraw-Hill, 2010.
- [10] D. T. Salaki, "Penentuan Lintasan Terpendek dari FMIPA ke Rektorat dan Fakultas lain di UNSRAT Manado Menggunakan Algoritma Dijkstra," *Jurnal Ilmiah Sains*, vol. 11, no. 74, pp. 73-76, 2011.
- [11] H. K. Rosen, *Discrete Mathematics and its Applications*, USA: McGraw-Hill, 1999.
- [12] R. Rainaldi, "Aplikasi Petunjuk Lokasi Sarana Umum di Kota Bandung Menggunakan Algoritma Dijkstra Berbasis Android", Skripsi Sarjana, Bandung: Universitas Komputer Indonesia, 2017.
- [13] R. Hasriandi, "Implementasi Algoritma Dijkstra untuk Pencarian Lintasan", Skripsi Sarjana, Jember: Universitas Muhammadiyah Jember, 2015.

- [14] S. Saputri dan A. J. P. Sibarani, "Implementasi Augmented Reality Pada Pembelajaran Matematika Mengenal Bangun Ruang Dengan Metode Marked Based Tracking Berbasis Android," *Komputika: Jurnal Sistem Komputer*, Universitas Komputer Indonesia, Bandung, vol. 9, no. 1, pp. 15-24, 2020.
- [15] G. W. Setiawan, "Pengujian Perangkat Lunak Menggunakan Metode Black Box Studi Kasus Exelsa Universitas Sanata Dharma," Skripsi Sarjana, Yogyakarta: Universitas Sanata Dharma, 2011.
- [16] A. E. Nugraha, "Pembangunan game maze edukasi Ilmu Pengetahuan Sosial (IPS)," Skripsi Sarjana, Bandung: Universitas Komputer Indonesia, 2014.