

PENERAPAN ALGORITMA *FINITE STATE MACHINE* DALAM GAME MAZE *QUIS* PENGENALAN HURUF ANAK USIA DINI

Abi Dinar Ramadhan^{1*}, Reva Ragam Santika²

^{1,2} Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta, Indonesia

Email: ^{1*}1911500500@student.budiluhur.ac.id., ²reva.ragam@budiluhur.ac.id

Abstrak- Pengenalan huruf untuk anak usia dini mungkin hal yang cukup sulit dicerna. Namun dengan perkembangan zaman dan kemajuan ilmu pendidikan dan teknologi, memaksa anak-anak untuk dapat mengenal jenis-jenis huruf ajlabar. Dengan seiring berkembangnya zaman dibidang teknologi yang menjangkau berbagai lapisan kehidupan masyarakat dari segala usia. Menyebabkan masyarakat terutama anak-anak menjadi malas untuk belajar dan lebih menggunakan fitur teknologi untuk menonton *video* di youtube dan bermain *video game* yang dapat menyebabkan kecanduan. Dengan ini penulis ingin membuat *game* edukasi yang menyenangkan sehingga anak-anak tidak cepat jenuh atau bosan. Aplikasi *game* ini dibuat menggunakan *game engine* Unity3D. Untuk membuat tantangan dalam permainan lebih menarik, karakter musuh harus dibuat menjaidi lebih reaktif dengan menggunakan algoritma *finite state machine* (FSM) yang terdiri dari dua keadaan yaitu *state* patroli dan *state idle*. Dalam pengujian program ini menggunakan metode pengujian *game development life cycle*(GDLC) metode pengujian ini menentukan apakah aplikasi yang di buat dapat berjalan dengan baik dan sesuai harapan. Berdasarkan hasil wawancara yang melibatkan terhadap beberapa aspek. Aspek *difficulties* 69% (cukup setuju) hasil ini didapat dari pertanyaan pertama dan kedua, aspek *easy to use* 77% (cukup setuju) hasil ini didapat dari pertanyaan ketiga dan keempat, dan aspek *usefulness* 80%(Sangat setuju) hasil ini didapat dari pertanyaan kelima dan keenam. Hasil analisis di atas membawa kita ke kesimpulan bahwa algoritma *finite state machine* dapat digunakan untuk membuat *game maze quis*. Fokus *game* ini adalah untuk mengenalkan bentuk-bentuk huruf secara menyenangkan dan bermain sambil belajar.

Kata Kunci: Game Edukasi, *Unity2d Engine*, *Maze*, *Quis*, Anak-anak, *Finite State Machine*.

THE APPLICATION OF FINITE STATE MACHINE ALGORITHM IN THE MAZE *QUIS* GAME INTRODUCTION TO EARLY CHILDREN'S LETTER INTRODUCTION

Abstract- Letter recognition for early childhood may be quite difficult to digest. However, with the times and advances in education and technology, forcing children to be able to recognize the types of ajlabar letters. Along with the development of the era in the field of technology that reaches various layers of people's lives of all ages. Causing people, especially children, to be lazy to learn and to use more technological features to watch videos on YouTube and play video games which can be addictive. With this the author wants to make an educational game developed using a finite state machine (FSM) algorithm which aims to produce an educational and fun game so that children don't get bored or bored quickly. This game application is made using the Unity3D game engine. To make the challenges in the game more interesting, enemy characters must be made more reactive by using a finite state machine (FSM) algorithm which consists of two states, namely the patrol state and the idle state. In testing this program using the game development life cycle (GDLC) testing method, this test method determines whether the application created can run properly and as expected. Based on the results of interviews involving 32 respondents to several aspects. Difficulty aspect 69% (quite agree) this result is obtained from the first and second questions, easy to use aspect 77% (quite agree) this result is obtained from the third and fourth questions, and usefulness aspect 80% (Strongly agree) this result is obtained from the question fifth and sixth. The results of the analysis above lead us to the conclusion that the finite state machine algorithm can be used to create a maze quis game. The focus of this game is to introduce letter shapes in a fun and playful way while learning

Keywords: Educational Game, *Unity2d Engine*, *Maze*, *Quiz*, Kids, *Finite State Machine*

1. PENDAHULUAN

Pengenalan huruf untuk anak usia dini mungkin hal yang cukup sulit dicerna. Namun dengan perkembangan zaman dan kemajuan ilmu pendidikan dan teknologi, memaksa anak-anak untuk harus bisa mengenal jenis-jenis

huruf ajlalar. Menurut Hariyanto (2009) mengungkapkan bahwa dengan strategi pengenalan huruf sejak usia dini sangat bermanfaat bagi perkembangan bahasa anak, karena membantu mempersiapkan anak untuk dapat membaca dengan mudah. [1]

Permainan atau yang sering disebut dengan game merupakan suatu sarana hiburan yang diminati oleh banyak orang, baik dari kalangan anak-anak, remaja maupun orang dewasa, platform yang digunakan untuk bermain *game* saat ini pun pada saat ini tidak hanya sebatas pada *konsol gaming* saja, mulai dari *computer* hingga *smartphone* bisa digunakan sebagai media memainkan *game*.

Game digemari oleh berbagai usia dari anak – anak sampai orang dewasa sebagai sarana hiburan yang mampu menyegarkan kembali dan melepas jenuh setelah aktivitas sehari – hari yang melelahkan, namun seiring berkembangnya industri *game* kini *game* tidak hanya digunakan semata sebagai hiburan tetapi dapat dibuat untuk lain hal seperti pada bidang bisnis, media pembelajaran atau edukasi. [2] [3]

Pesatnya perkembangan industri *game* dan teknologi semakin membuat anak-anak menjadi malas untuk belajar membuat penulis termotivasi untuk membuat sebuah *game* dan bagaimana *game* ini dapat membantu anak-anak usia dini memahami bentuk huruf dengan menerapkan metode *finite state machine* (FSM) pada *game* edukasi *maze quis* pengenalan huruf untuk anak usia dini.

Penerapan *game* untuk media pendidikan atau yang disebut *game* edukasi bermula dari perkembangan *video game* yang sangat pesat dan menjadikannya sebagai media alternatif untuk kegiatan pembelajaran. *Game* edukasi ini perlu dikembangkan dan seharusnya *game* tidak hanya menyenangkan tapi juga mendidik. [4]

Dan seiring berkembangnya *teknologi game* menjadi sangat *complex* dan memiliki banyak *genre* atau tipe *game* yang dapat dimainkan seperti *adventure*, *maze*, *shooting*, *action*, dan masih banyak lagi. Penulis sampai pada kesimpulan bahwa *game* adalah media yang digunakan untuk tujuan tertentu, seperti hiburan dan pembelajaran. Para ahli juga menemukan bahwa beberapa *game* memiliki efek *negatif* bagi pemain, seperti menjadi kecanduan bermain *game* dan menjadi malas untuk belajar hal baru.

Oleh karna itu penulis ingin membuat *game* edukasi menggunakan metode *finite state machine*(FSM), harapannya anak-anak yang memainkan *game* ini dapat memahami pelajaran dan manfaat serta merubah gaya pembelajaran supaya tidak cepat jenuh dan membosankan. Pada proses pembuatan *game* edukasi ini penulis menggunakan aplikasi Unity sebagai *game enginanya* dan bahasa C# sebagai bahasa pemrogramannya.

2. METODE PENELITIAN

2.1 Game

Permainan merupakan kegiatan terstruktur yang pada umumnya memiliki tujuan sebagai sarana hiburan. Selain sebagai hiburan, *game* juga dapat digunakan sebagai sarana edukasi. Kegiatan ini digemari oleh banyak kalangan karena *game* memiliki karakteristik yang membuat pemainnya senang dan kolaboratif. [5]

Game merupakan salah satu hiburan karena mampu mengurangi tingkat kepenatan seseorang dari rutinitas pekerjaan setiap hari. Menurut jonne arjoranta (2019) *game* adalah mode interaksi antara pemain, mesin dengan tampilan visual elektronik, dan mungkin pemain lain, yang dimediasi oleh konteks fiksi yang bermakna, dan ditopang oleh ikatan emosional antara pemain dan hasil dari tindakannya dalam konteks fiksi ini. [6].

Game pada dasarnya adalah hiburan karena pengguna merasa senang saat memainkannya. Di zaman modern ini, *game* disajikan dengan kualitas visualisasi yang cukup menuntut, karena *game* tersebut didukung oleh teknologi sehingga pemain menjadi lebih interaktif sesuai keinginannya dan pemain merasa hidup dalam *game* tersebut. Dengan kata lain, *game* berkembang seiring dengan teknologi. [7].

Dapat dikatakan bahwa *game* berkembang beriringan dengan perkembangan teknologi. Saat ini kualitas visual pada *game* sudah canggih, hal ini didukung oleh teknologi sehingga pemain dapat berinteraksi sesuai keinginannya dan memberikan rasa realistis dalam *game*. [7]

2.2 Finite State Machine

Game ini dibuat dengan metode algoritma *finite state machine* dalam *game* ini pemain harus mengumpulkan beberapa huruf yang telah diletakan secara acak, dan apabila pemain berhasil mengumpulkan huruf maka pemain akan masuk ke *level quis*. *Dilevel quis* terdapat gambar buah yang harus pemain tebak, dan bila pemain berhasil menjawab *quis* dengan benar maka pemain akan masuk ke-menu untuk memilih *level* selanjutnya.

Menurut Arif (2020) *finite state machine* adalah metode yang dapat digunakan untuk memodelkan perilaku karakter di dalam *game* berdasarkan keadaan yang sudah ditentukan. Transisi keadaan pada *finite state machine* lebih rumit dan urutannya menunjukkan tidak ada pola yang sederhana dan teratur, seperti pada *counter* atau *shift register*. Logika keadaan berikutnya harus dibangun dari awal dan kadang-kadang dikenal sebagai logika acak. [8]

Menurut Firdaus (2019) *finite state machine* (FSM) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: State

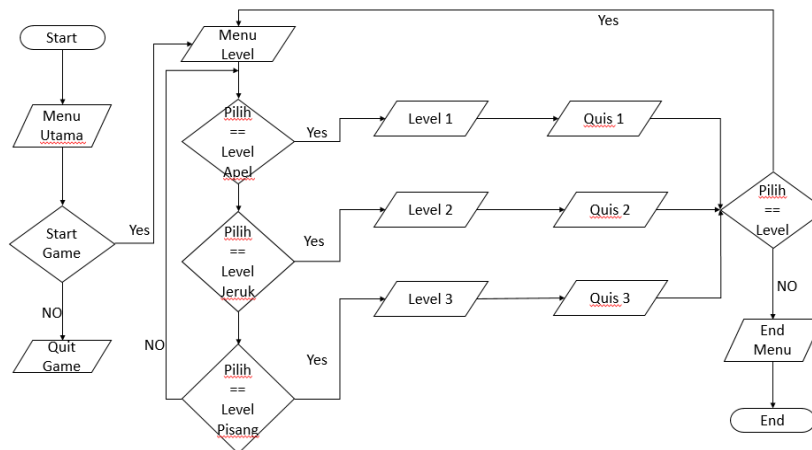
(Keadaan), Event (kejadian) dan action (aksi). [9]. *State machine* bekerja dengan cara menempati satu keadaan (state) kemudian sistem akan berpindah menuju keadaan (state) lain jika mendapatkan event atau masukan tertentu. State akan terus melakukan aksi yang sedang dilakukannya sampai mendapatkan masukan atau event baru yang berasal dari komponen sistem itu sendiri maupun dari perangkat luar. Terdapat transisi menghubungkan setiap state yang ada. Transisi state biasanya disertai dengan aksi yang dilakukan sistem jika mendapat masukan tertentu. [10]

Pada permainan ini progres FSM berlangsung dengan menerima perintah *patrol* & mengejar, saat memulai permainan musuh akan berada dalam keadaan *patrol* dan bila pemain mendekati musuh maka musuh akan masuk kedalam keadaan mengejar.

Pada topik sebelumnya ada sebuah karya ilmiah yang mengangkat topik dengan metode FSM yang berjudul rancang bangun game labirin hijaiyah dengan unity menggunakan metode finite state machine karya Khoiril Solekhan Arif dan menjadi inspirasi penulis untuk membuat topik dengan metode yang sama yaitu finite state machine.

2.3 Rancangan Menu

Gambar dibawah ini menunjukkan proses pembuatan menu untuk game yang dilakukan dalam bentuk *flowchart*. Struktur menu ini berisi semua menu yang ada di dalam *game*. Perancangan menu terdiri dari 9 *scene* yaitu menu utama, menu pemilihan *level*, labirin *level 1*, labirin *level 2*, labirin *level 3*, *quis level 1*, *quis level 2*, *quis level 3* dan menu akses permainan. Saat pertama kali membuka *game*, menu utama akan muncul, pemain dapat langsung memulai *game*. Untuk mulai bermain, pemain dapat memilih menu mulai yang mengarah ke menu pemilihan *level*. Setelah memilih *level* yang akan dimainkan, adegan dari *level* yang dipilih akan muncul dan seterusnya. *Flowchart* dapat dilihat pada gambar 1 dibawah ini.



Gambar 1 Flowchart Rancangan Menu

Berikut merupakan penjelasan dari setiap *menu* dalam *flowchart* di atas:

- Saat pemain membuka permainan, pemain akan dihadapkan dengan menu utama dan ini merupakan tampilan awal permainan untuk memulai permainan pemain dapat menekan tombol *start*.
- Setelah menekan tombol *start* pemain akan masuk ke menu pilih *level* untuk memilih *level* yang ada serta *player* dapat memilih *level* dengan bebas.
- Saat pemain telah memilih *level 1 – 3* pemain akan masuk *scene level* yang ada dan bila pemain berhasil menyelesaikan *level* maka akan masuk ke *stage quis* sesuai dengan *level* yang pemain pilih.
- Saat pemain berhasil menjawab *quis* dengan benar pemain akan masuk ke menu pilih *level* untuk memilih *level* selanjutnya dan apabila pemain ingin menyudahi permainan pemain bisa menekan tombol *end* yang akan membuka *end menu*.
- End Menu* akan muncul setelah pemain menyelesaikan *quis* dan untuk mengakhiri permainan menekan tombol *quit*.

2.4 Rancangan Pengujian

Pengujian aplikasi ini ditunjukkan untuk mengetahui faktor kemudahan, kesulitan, kemanfaatan, dan pengaruh dibidang pembelajaran terutama dalam mengenal huruf. Berikut merupakan data nama individu yang telah memberikan pendapat serta penilaian terhadap *game maze quis* yang penulis buat, dapat dilihat pada table 1 dibawah ini.

Table 1 Data Individu Wawancara

Nama	Jenis Kelamin	Profesi	Umur
Pak Ari	Pria	Guru	-
Pak Cecep	Pria	Guru	-
Pak Udin	Pria	Guru	-
Bu Eli	Wanita	Guru	-
Nata	Wanita	Pelajar	7
Nadin	Wanita	Pelajar	6
Sarah	Wanita	Pelajar	10
Farez	Pria	Pelajar	9
Farhan	Pria	Pelajar	8
Niken	Wanita	Pelajar	11
Dimas	Pria	Pelajar	5
Wisnu	Pria	Pelajar	15

2.5 Game Development Life Cycle (GDLC)

Dalam pengembangan *game* ini, digunakan metode pengembangan sistem yaitu GDLC (*Game Development Life Cycle*). Ada 6 unsur atau tahapan yang digunakan dalam metode pengembangan GDLC. [11]

1. *Initiation*
2. *Pre-Production*
3. *Production*
4. *Testing*
5. *Beta*
6. *Release*

Tahap Inisiasi (*Initiation*)

Pada tahap ini penulis akan mempersiapkan kebutuhan yang akan di gunakan untuk membuat *game*, mulai dari objek penelitian, *tools* atau alat yang di gunakan dalam membuat sebuah *game* seperti *software game engine*, *software* pendukung, *asset art game*, serta *hardware* penunjang proses pembuatan dan pengujian *game*.

Tahap Persiapan (*Pre-Production*)

Pada tahap ini penulis melakukan pekerjaan awal *game* dan memikirkan konsep *game* yang akan di buat. Tahap ini juga berfokus pada

1. Mendefinisikan jenis *game*
2. *Flowchart* sistem
3. *Gameplay Mechanics*
4. Tantangan (*Challenge*)

Tahap Pengerjaan (*Production*)

Pada tahap ini penulis mulai mengerjakan bagian inti dari pembuatan *game* yang melibatkan :

1. Pengumpulan material *game*
2. Membuat map
3. Membuat *script player* dan musuh
4. Membuat objek yang akan menjadi target *player*.

Tahap Uji Coba (*Testing*)

Pada tahap ini penulis akan menguji beberapa aspek yang ada di *game* ini seperti *desain*, tingkat kesulitan, dan *gameplay*. Penulis akan menguji *game* ini dengan saudara, teman, dan penulis sendiri. Tahap ini terbagi menjadi 2 bagian yaitu:

1. *Desain Testing*

Pada pengujian ini berfokus pada kemudahan *akses* dalam *game* dan kemudahan mengerjakan *player* dalam *game*.

2. *Difficulty Testing*

Pada pengujian ini berfokus pada *level* yang ada dan tingkat kesulitan dari setiap *level*.

Tahap Beta

Pada tahap ini, aplikasi atau *game* yang dibuat akan diberikan kepada penguji dari pihak ketiga atau penguji dari luar. Penguji ini akan menentukan hasil dari pengerjaan *game* ini mulai dari kualitas, kemudahan dalam bermain atau *control*, tingkat kesulitan pada *game*, pencarian *bug*, dan lain-lain. Penguji pada tahap beta ini adalah anak usia dini, remaja, dan orang tua.

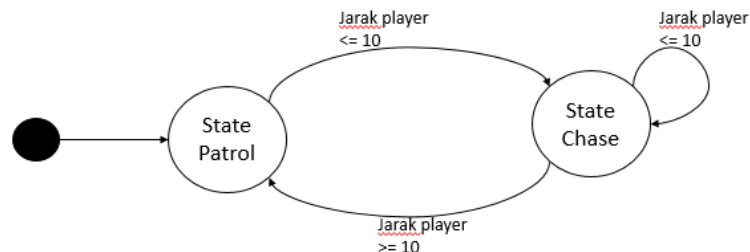
Tahap Perilisan (*Release*)

Pada tahap akhir ini, *game* telah diuji oleh pihak *internal* maupun beta maka *game* akan siap di rilis secara resmi setelah perbaikan selesai dilakukan.

3. IMPLEMENTASI DAN HASIL

3.1 Penerapan Finite State machine

Salah satu mesin pemerintah yang akan digunakan terhadap musuh yang ada digambarkan pada diagram *finite state machine* pada musuh, yang terdiri dari dua keadaan, *state patrol* dan *state chase*. Berikut tampilan FSM pada musuh pada gambar 2 dibawah ini.



Gambar 2 Contoh Diagram *Finite State Machine* Pada Musuh.

Berikut adalah deskripsi tentang keadaan dari gambar 2:

State : *Patrol*

Deskripsi : keadaan *patrol* adalah keadaan awal dari *finite state machine*, pada keadaan ini musuh akan bergerak di *point patrol* yang telah di tentukan disekitar pintu atau *objek* yang *player* harus kumpulkan.

State : *Chase*

Deskripsi : Musuh akan mulai mengejar *player* apabila *player* berada didalam radius yang telah di tentukan dan apabila *player* sudah berada di luar radius, maka musuh akan kembali kedalam *state idle*.

3.2 Tampilan Game

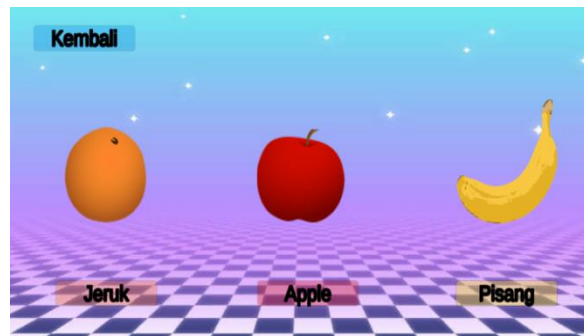
Berikut merupakan tampilan pada *game* yang penulis ambil melalui *library asset store* yang disediakan oleh Unity 3D dan OpenGameArt.

Saat pertama kali *player* membuka aplikasi *game* ini, maka *scene* pertama yang *player* akan temui adalah menu utama. Didalam menu utama terdapat 2 tombol yaitu tombol *start* yang apabila di klik akan memulai *game* dan langsung masuk ke menu pilih *level*, yang kedua adalah tombol *quit* yang apabila di klik maka program akan ditutup secara langsung. Tampilan dapat dilihat pada gambar 3 dibawah ini.



Gambar 3 Tampilan Menu Pilih Level

Menu ini akan muncul jika player menekan tombol *start* pada menu utama dan di menu ini terdapat 3 *level* yaitu *level jeruk*, *level apel*, dan *level pisang*. Setiap *level* memiliki kesulitan yang berbeda dan *player* dapat dengan bebas memilih *level* yang *player* inginkan. Tampilan dapat dilihat pada gambar 4 dibawah ini



Gambar 4 Tampilan Layar Utama

Berikut adalah tampilan *interface player* tampilan ini memberikan informasi dimana huruf terdekat dan musuh terdekat serta jalan yang harus di lalui. Dan sedikit deskripsi tentang berapa banyak huruf yang harus di kumpulkan dan yang telah di kumpulkan. Tampilan dapat dilihat pada gambar 5 dibawah ini



Gambar 5 Tampilan Player

Jika *player* berhasil menyelesaikan *level* apel maka *player* akan masuk ke *level quis 1*, *interface* di *level quis* ini memiliki gambar dan pertanyaan serta 4 jawaban jika *player* salah menjawab maka akan muncul tampilan yang menyatakan bahwa jawabannya salah dan jika benar maka akan masuk ke menu pilih *level* untuk memilih *level* berikutnya. Tampilan dapat dilihat pada gambar 6 dibawah ini



Gambar 6 Tampilan *Quiz*

Setelah menyelesaikan *game*, *player* akan diperlihatkan tampilan *credit scene* yang berisikan nama pembuat dan motivasi untuk *player*. Tampilan dapat dilihat pada gambar 7 dibawah ini



Gambar 7 Tampilan *Credit Scene*

3.3 Hasil Wawancara Tertutup

Pada tabel 2 dibawah ini adalah hasil dari wawancara tertutup.

Table 2 Hasil Wawancara Tertutup

Nama	Pertanyaan 1	Pertanyaan 2	Pertanyaan 3	Pertanyaan 4	Pertanyaan 5	Pertanyaan 6
Pak ari	Red	Yellow	Red	Green	Green	Green
Bu eli	Green	Yellow	Yellow	Yellow	Green	Green
Pak cecep	Red	Red	Yellow	Green	Green	Green
Pak nana	Yellow	Yellow	Yellow	Green	Green	Green
Pak udin	Yellow	Yellow	Yellow	Green	Green	Green
Wisnu	Yellow	Red	Yellow	Yellow	Yellow	Yellow
Dimas	Green	Yellow	Red	Red	Yellow	Yellow
Farhan	Red	Green	Red	Red	Yellow	Yellow
Farez	Yellow	Yellow	Yellow	Yellow	Green	Yellow
Sarah	Yellow	Green	Yellow	Yellow	Red	Red
Niken	Yellow	Red	Green	Yellow	Red	Red
Nata	Green	Red	Green	Green	Green	Yellow
Nadin	Yellow	Green	Yellow	Green	Green	Yellow

Hasil wawancara yang telah dilakukan ke-beberapa individu sebanyak 60 individu sangat setuju , sebanyak 84 individu cukup setuju, dan sebanyak 48 individu kurang setuju.

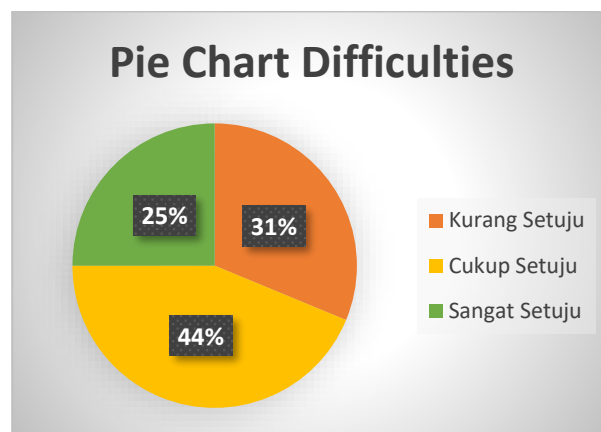
3.4 Hasil Variabel Difficulties

Pada tabel 3 dibawah ini adalah hasil dari perhitungan dan persentase dari uji coba kesulitan (*difficulties*).

Table 1 Hasil Uji Variabel Difficulties

No	Pertanyaan	Sangat Setuju	Cukup Setuju	Kurang Setuju	Respon
1	Setelah memainkan <i>game</i> ini kira-kira menurut anda apakah <i>game</i> ini terlalu sulit/mudah?	10	13	9	32
2	Menurut anda rintangan yang ada di <i>game</i> ini terlalu sulit/mudah?	6	15	11	32
	Total	16	28	20	

Variabel ini bertujuan untuk mengetahui kesulitan serta kerumitan aplikasi *game* “*Maze Quis*”, data pada variabel ini memiliki 2 pertanyaan dan menggunakan 3 alternatif jawaban. Berdasarkan pertanyaan apakah *game* ini terlalu sulit baik dari segi rintangan dan *level*, sebanyak 44% individu mengatakan cukup setuju, 31% individu kurang setuju, dan 25% individu sangat setuju. Presentase digambar dalam pie chart pada gambar 8.


Gambar 8 Pie Chart Variabel *Difficulties*

Dari hasil presentase pie chart pada gambar 8 dapat disimpulkan presentase sangat setuju + cukup setuju yaitu 69% perihal kesulitan yang ada pada *game maze quis*.

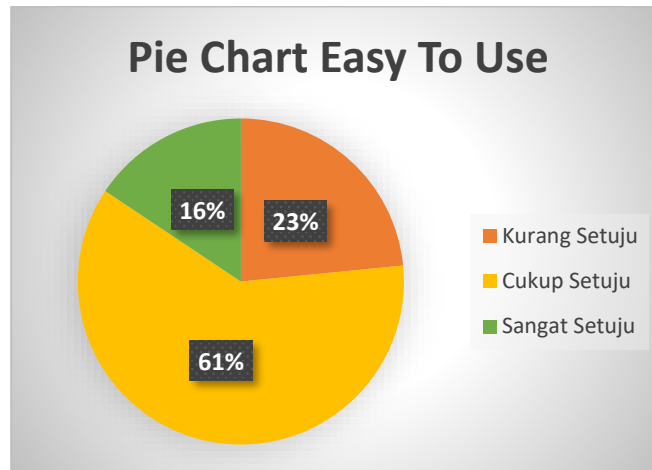
3.5 Hasil Variabel Easy To Use

Pada tabel 4 dibawah ini adalah hasil dari perhitungan dan persentase dari uji coba keandalan (easy to use).

Table 2 Hasil Uji Variabel Easy To Use

No	Pertanyaan	Sangat Setuju	Cukup Setuju	Kurang Setuju	Respon
1	Menurut anda desain pada <i>game</i> ini sudah bagus/kurang bagus?	2	21	9	32
2	Setelah memainkan <i>game</i> ini kira-kira dari segi kontrolnya sudah bagus/kurang bagus?	8	18	6	32
	Total	10	39	15	

Variabel ini bertujuan untuk mengetahui kemudahan *control* pada aplikasi *game* “*Maze Quis*”, data pada variabel ini memiliki 2 pertanyaan dan menggunakan 3 alternatif jawaban. Berdasarkan pertanyaan diatas, sebanyak 61% individu mengatakan cukup setuju, 23% individu kurang setuju, dan 16% individu sangat setuju. Presentase digambar dalam pie chart pada gambar 9.



Gambar 9 Pie Chart Variabel *Easy To Use*

Dari hasil presentase pie chart pada gambar 9 dapat disimpulkan presentase sangat setuju + cukup setuju yaitu 77% perihal kemudahan kontrol yang ada pada *game maze quis*.

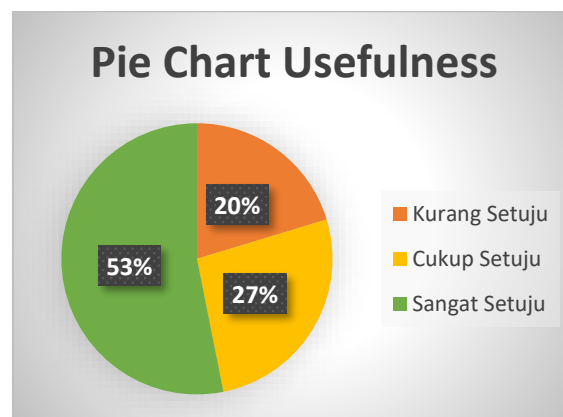
3.6 Hasil Variabel Usefulness

Pada tabel 5 dibawah ini adalah hasil dari perhitungan dan persentase dari uji coba keandalan (usefulness).

Table 3 Hasil Uji Variabel Usefulness

No	Pertanyaan	Sangat Setuju	Cukup Setuju	Kurang Setuju	Respon
1	Setelah memainkan <i>game</i> ini apakah anda akan merekomdesikannya ke orang lain?	15	8	9	32
2	Menurut anda apakah <i>game</i> ini memiliki dampak untuk pembelajaran terutama pembelajaran tentang pengenalan huruf?	19	9	4	32
Total		34	17	13	

Variabel ini bertujuan untuk mengetahui kegunaan serta dampak dari aplikasi *game "Maze Quis"* pada pembelajaran pengenalan huruf. Data pada variabel ini memiliki 2 pertanyaan dan menggunakan 3 alternatif jawaban. Berdasarkan pertanyaan menurut anda apakah *game* ini memiliki dampak untuk pembelajaran terutama pembelajaran tentang pengenalan huruf?, sebanyak 27% individu mengatakan cukup setuju, 20% individu kurang setuju, dan 53% individu sangat setuju. Presentase digambar dalam pie chart pada gambar 10.



Gambar 10 Pie Chart Variabel *Usefulness*

Dari hasil presentase pie chart pada gambar 10 dapat disimpulkan presentase sangat setuju + cukup setuju yaitu 80% perihal manfaat yang ada pada *game maze quis*.

3.7 Hasil Keseluruhan

Pada table 6 dibawah ini adalah hasil yang diperoleh dari perhitungan dan persentase seluruh tabel. Kesulitan *game* (difficulties), kemudahan kontrol (easy to use), manfaat (usefulness).

Table 4 Table Hasil Keseluruhan

No.	Hasil Uji	Sangat Setuju	Cukup Setuju	Kurang Setuju	Presentase
1	Variabel Kesulitan (difficulties)	16	28	20	69%
2	Variabel kemudahan kontrol (easy to use)	10	39	15	77%
3	Variabel Manfaat (usefulness)	34	17	13	80%

Tabel di atas menunjukkan skor kesulitan. (*difficulties*) 69% , kemudahan kontrol (*easy to use*) 77%, manfaat (*usefulness*) 80%, hasil analisis diatas menyimpulkan bahwa algoritma finite state machine dapat digunakan dalam membuat *game maze quis*.

4. KESIMPULAN

Ada kesimpulan yang dapat diambil dari analisis yang telah dilakukan mengenai masalah yang ada dan rekomendasi yang mungkin diperlukan untuk mendukung pengembangan aplikasi *game maze quis*.

Hasil analisis di atas menunjukkan bahwa ada masalah dan solusi yang dikembangkan. Dari data ini, penulis dapat mengambil kesimpulan berikut:

- Algoritma *finite state machine* dapat digunakan untuk membuat *game maze quis*.
- Game maze quis* terbukti dapat bermanfaat untuk media pembelajaran terutama dalam pengenalan huruf.
- Dari wawancara yang telah dilakukan dapat melihat apa yang orang pikirkan tentang kesulitan (difficulties) 69% (cukup baik), kemudahan (easy to use) 77% (baik), manfaat (usefulness) 80% (sangat baik).
- Game Edukasi dapat menjadi sarana pembelajaran secara santai dan menyenangkan.

DAFTAR PUSTAKA

- [1] A. Hariyanto, *Membuat Anak Anda Cepat Pintar Membaca! : panduan dan metode penerapannya*, Yogyakarta: DIVA Press, 2009.
- [2] R. Robianto, "Penerapan Algoritma Finite State Machine Pada Game Edukasi Mencocokkan Satwa Untuk Anak Usia Dini", *Prosiding Seminar Nasional Mahasiswa Fakultas Teknologi Informasi*, Vol. 1 , No.1, pp. 600-606, 2022.
- [3] N. F. R. Widiadhani, "PENERAPAN METODE FINITE STATE MACHINE UNTUK", *Seminar Nasional Mahasiswa Fakultas Teknologi Informasi*, Vol. 1, pp. 819-828, 2022.
- [4] S. Setiawan, "GAME "EDUKASI MATEMATIKA" MENGGUNAKAN METODE PATH", *Jurnal Mahasiswa Teknik Informatika*, Vol. 3, No.1, pp. 411-415, 2019.
- [5] D. ARI, S. A. Wibowo and Y. A. Pranoto, "PENERAPAN METODE FINITE STATE MACHINE DAN FUZZY PADA GAME "BLACK WARRIOR"", *Jurnal Mahasiswa Teknik Informatika*, Vol. 5, pp. 708-709, 2021.
- [6] J. Arjoranta, "How to Define Games and Why We Need to", *The Computer Games Journal*, Vol. 8, pp. 109-120, 2019.
- [7] G. Mau, "RANCANG BANGUN GAME 2D SHOOTER PLATFORMER MENGGUNAKAN METODE FINITE STATE MACHINE", *Jurnal Mahasiswa Teknik Informatika*, Vol. 3, No.1, 2019.
- [8] K. S. Arif, "Rancang Bangun Game Labirin Hijaiyah Dengan Unity Menggunakan Metode", *Journal of Computer and Information Technology*, Vol. 4, No.1, pp. 49-53, 2020.
- [9] M. Firdaus, "PENERAPAN METODE FINITE STATE MACHINE PADA GAME ADVENTURE "TRAPPED MINERS"", *Jurnal Mahasiswa Teknik Informatika*, Vol. 3, No.1, pp. 158-164, 2019.
- [10] E. Yulsilviana and H. Ekawati, "PENERAPAN METODE FINITE STATE MACHINE (FSM) PADA GAME AGENT LEGENDA ANAK BORNEO", *SEBATIK*, Vol. 23, pp. 106-123, 2019.
- [11] N. O. Otto, V. Amrizal and F. E. M. Agustin, "Rancang bangun game edukasi quiz interaktif bahasa Inggris dengan menerapkan gamifikasi berbasis android," Skripsi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah, Jakarta, Indonesia, 2020.