Digital Transformation Technology (Digitech) | e-ISSN : 2807-9000
Volume 3, **Number** 1, Maret 2023
https://doi.org/10.47709/digitech.v3i1.2745

Penerapan Line of Sight dan Finit State Machine pada Game Platformer "RUN!"

Andi Yusika Rangan[1*], Hendy[2]
[1,2]STMIK Widya Cipta Dharma, Indonesia
[1]Andi@wicida.ac.id, [2]hendy.stark16@gmail.com

Histori Artikel:
Diajukan: 20 Agustus 2023
Disetujui: 21 Agustus 2023
Dipublikasi: 21 Agustus 2023

Kata Kunci:
Component; Finite State Machine; Game; Line Of Sight; Platformer; Usability Test.

Abstrak
Pada saat ini, game telah memperluas fungsinya lebih dari sekadar fasilitas hiburan; Beberapa digunakan sebagai sarana belajar, lahan bisnis, dan bahkan sebagai olahraga oleh para profesional. "RUN" adalah game platformer gender. Dalam permainan ini, pemain mengontrol karakter utama, yang dapat berjalan, melompat, berlari, dan memanjat untuk menghindari karakter atau musuh non-pemain, serta mengambil item, menghindari rintangan, dan menerapkan mesin keadaan terbatas dan metode garis pandang untuk karakter musuh atau karakter non-pemain. Pengembangan aplikasi adalah metode pengembangan multimedia yang terdiri dari konsep, desain, pengumpulan material, perakitan, pengujian, dan distribusi. Pada tahap konsep, yang dilakukan adalah menentukan target pengguna, spesifikasi umum, dan aturan desain dasar. Pada tahap desain, perancangan yang berkaitan dengan struktur permainan meliputi penggunaan case diagram, activity diagram, sequence diagram, dan flowchart. Pada tahap pengumpulan materi, gambar dan audio pendukung juga dibutuhkan untuk membuat game. Pada tahap perakitan, pembuatan game dilakukan berdasarkan desain yang telah dibuat dan bahan yang telah dikumpulkan. Menggunakan GameMaker Studio sebagai aplikasi pembuatannya, dimainkan pada *desktop* baik PC atau *Laptop*. Pengujian dilakukan dengan menggunakan usability test, dengan hasil menunjukkan bahwa 92% aplikasi dapat berjalan dengan baik. Pada tahap akhir, distribusi game "RUN!" telah dikemas ke dalam sebuah aplikasi yang dapat dimainkan oleh semua kalangan sebagai hiburan bagi para penggemar game platformer klasik

PENDAHULUAN

Penggunan game saat ini tidak hanya sebagai sarana hiburan tetapi juga sebagai sarana pembelajaran, lahan bisnis dan salah satu cabang olah raga. Game dengan genre *platformer* adalah *genre* dari jenis game yang masih banyak peminatnya. *Genre platformer* adalah sebuah *genre game* yang menitikberatkan kontrol pada karakter seperti berlari, meloncat dan memanjat untuk menghindari rintangan. Ada beberapa penelitian yang berkaitan dengan *genre game* ini :

a. pada penelitian dengan judul Membangun *Side Scrolling Game* "Petualangan Yada Mencari Benda Pusaka" Menggunakan Metode Algoritma A* (A *Star*) Dan Algoritma *Line of Sight* dihasilkan game dibuat menggunakan contsruct 2 dan dilakukan pengujian black box dan white box serta memanfaatkan emulator android (Prabowo, 2020)
b. Penelitan dengan judul membangun "RPG (Role Playing Game) "Adventure Of Ren : Edenweid Island Dan Pengembangan Game Agent Berbasis Finite State Machine" ini game menerapkan FSM digunakan pada game agent dan dibuat menggunakan RPG maker, pengujian yang digunakan ada beta testing (Sanjaya, 2017).
c. penelitian dengan judul Implementasi Pathfinding A-Star Dan Line Of Sight Pada Game Edukasi Pembelajaran Online Berbasis Android Untuk Mengurangi Kejenuhan Belajar, menerapkan algoritma A-Star Pathfinding dan Line of Sight dengan metode pengujian black box dan beta testing (Rizal Dwi Saputro et al., 2022)

Berdasarkan hasil penelitian diatas *geme Genre platformer* dapat dibuat dengan megunakan berbagai software pembuat game seperti construk dan RPG dan untuk karakter agen game menjadi lebih hidup maka karater tersebut bisa diimplementasikan algoritma seperti *Finite State Machine* (FSM). Berdasarkan latar belakang tersebut maka

dibuatlah rumusan masalah bagaimana membuat aplikasi *game* "RUN!" dan menerapkan metode *Line of Sight* dan *Finite State Machine* pada *game* tersebut. Game akan dibuat mengunakan GameMaker Studio 2.

STUDI LITERATUR

1. Game

Game atau permainan adalah suatu cara belajar dengan menganalisa dengan sekelompok pemain maupun individual dengan menggunakan strategi-strategi yang rasional (Agustina, 2015). Menurut (Henry, 2013) Berikut jenis-jenis game :

a) Role Playing Game (RPG)

Game role-playing (RPG) adalah salah satu game yang menggabungkan pengalaman atau level ke dalam gameplay mereka. Biasanya, pencipta game ini memiliki kebebasan untuk menjelajahi dunia game dan, dalam beberapa kasus, untuk menentukan kesimpulan game

b) Strategy

Strategi adalah genre permainan di mana tujuannya adalah untuk menyerang markas musuh dengan memimpin unit atau pasukan. Biasanya, dalam jenis permainan ini, perlu untuk mendapatkan emas untuk mendanai tim.

c) Simulation

Simulation adalah genre yang mementingkan realisme. Segala faktor pada*game* ini sangat diperhatikan agar semirip mungkin dengan dunia nyata. Segala nilai, material, referensi, dan faktor lainnya adalah berdasarkan dari dunia nyata. Cara memainkannya juga berbeda, karena biasanya kontrol yang dimiliki cukup rumit.

d) Racing

Game balap adalah jenis game balap di mana pemain mengontrol kendaraan dalam upaya memenangkan perlombaan.

e) Action Adventure

Action Adventure adalah game di mana petualangan penuh aksi dari salah satu karakter berlanjut sampai akhir game.

f) Arcade

Game arcade adalah subgenre video game yang tidak memiliki fokus naratif dan dimainkan "hanya untuk bersenang-senang" atau untuk mengejar skor tinggi.

g) Fighting Game

Game pertarungan adalah game pertarungan. Mirip dengan game arcade, pemain melepaskan teknik yang kuat dalam pertempuran. Biasanya, genre ini dimainkan di arena kecil melawan satu lawan.

Game juga diklasifikasikan berdasarjan jenis platformnya yang diunakan sebagai berikut :

a) *Arcade Game*s
*Arcade game*s yaitu sering disebut ding-dong di Indonesia, biasanya berada di daerah/tempat khusus dan memiliki *box* atau mesin yang memang khusus didesain untuk jenis *video game*s tertentu dan tidak jarang bahkan memiliki fitur yang dapat membuat pemainnya merasa masuk dan menikmati, seperti pistol, kursi khusus, sensor gerakan, sensor injakan, dan setik mobil (beserta transmisi tentunya).

b) PC games
PC *Game*s atau yang juga dikenal dengan *Desktop game*s adalah *video game* yang dimainkan dengan menggunakan *Personal Compute*

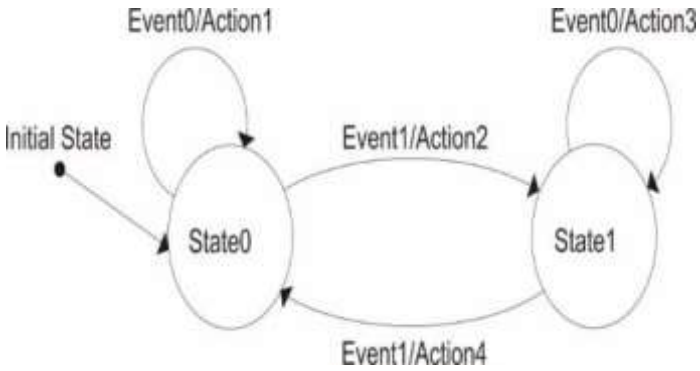
c) Console Games
*Console game*s adalah video *game* yang dimainkan menggunakan *console* tertentu, seperti Playstation 2, Playstation 3, XBOX 360, dan Nintendo Wii

d) *Handheld Game*s
*Handheld game*s adalah *video game* yang dimainkan di *console* khusus *video game* yang dapat dibawa kemana saja, contoh Nintendo DS dan Sony PSP

e) *Mobile Game*s
*Mobile game*s adalah *video game* yang dapat dimainkan atau khususuntuk *mobile phone* atau PDA

2. *Finite State Machine*

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). Pada satu saat dalam periode waktu yang cukup signifikan, sistem akan berada pada salah satu state yang aktif (Rahadian et al., 2017). Sebuah *state* hanya bisa terjadi dalam sekali waktu dan FSM digunakan untuk mengurai perilaku/sifat dari objek untuk bisa mudah diolah menjadi state atau bagian (Gambar 1).

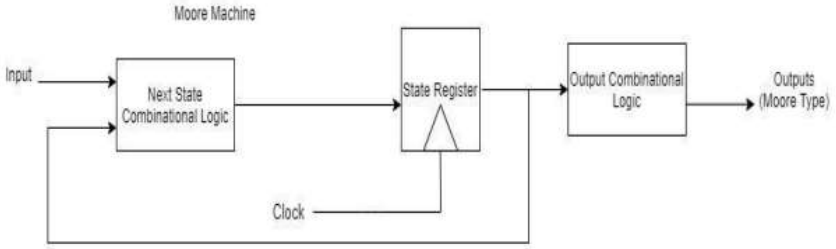


Gambar 1. *Finite State Machine*

Gambar daias menjelaskan prilaku objek agar lebih mudah dikelola menjadi sebuah potongan atau *state.* Sedangkan menurut (Asmiatun & Putri, 2017), *Finite State Machine* adalah perangkat atau model perangkat yang memiliki jumlah *state* atau kondisi terbatas bisa dalam pada waktu tertentu dan dapat beroperasi pada masukkan baik untuk membuat transisi dari satu *state* ke *state* yang lain atau membuat sebuah *output* perilaku atau tindakan. Sebuah state yang terbatas hanya bisa terjadi dalam satu waktu.

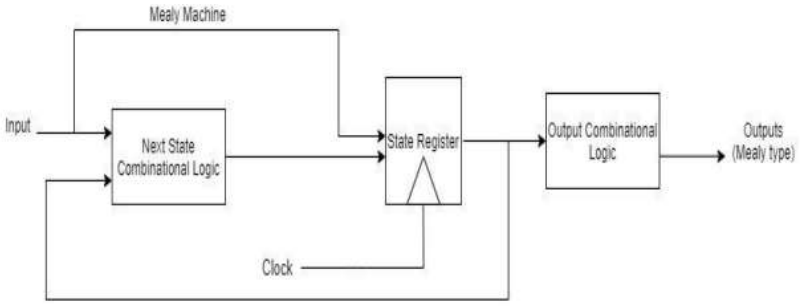
FSM diklasifikasikan menjadi dua jenis yang mereka beri nama mesin Moore dan Mealy yang banyak digunakan untuk aplikasi automata dan control (Garapati & Musala, 2020). Ada jenis FSM lain yang disebut FSM deterministik dan FSM Non-deterministik. FSM deterministik adalah yang untuk setiap simbol input, seseorang dapat menentukan keadaan mesin berikutnya dan memiliki jumlah keadaan terbatas di mana seperti dalam FSM non-deterministik seseorang tidak dapat menentukan keadaan mesin berikutnya :

a) Mesin Moore
   Mesin Moore Mesin keadaan terbatas dikatakan sebagai mesin Moore jika outputnya hanya bergantung pada keadaan sekarang dan tidak memiliki hubungan dengan input. Untuk menghasilkan output, ia mempertimbangkan jam seperti yang ditunjukkan pada Gambar 2.



Gambar 2. Moore Machine umum Block Diagram

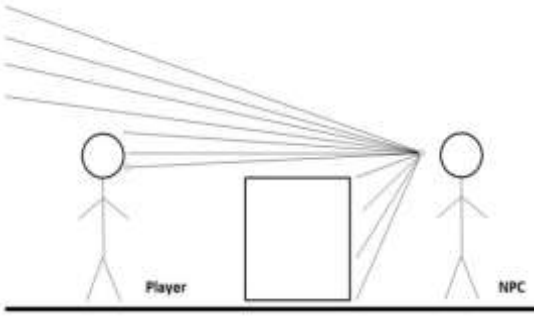
b) Mesin Mealy
   Mesin Finite state dikatakan sebagai mesin Mealy jika outputnya hanya bergantung pada input present state dan present seperti yang ditunjukkan pada Gambar 3. Untuk input yang diterapkan, dapat menghasilkan berbagai pola output yang berbeda dalam keadaan yang sama. Ini ditentukan sebagai fungsi dari input dan sinyal clock



Gambar 3. Mealy Machine umum Block Diagram

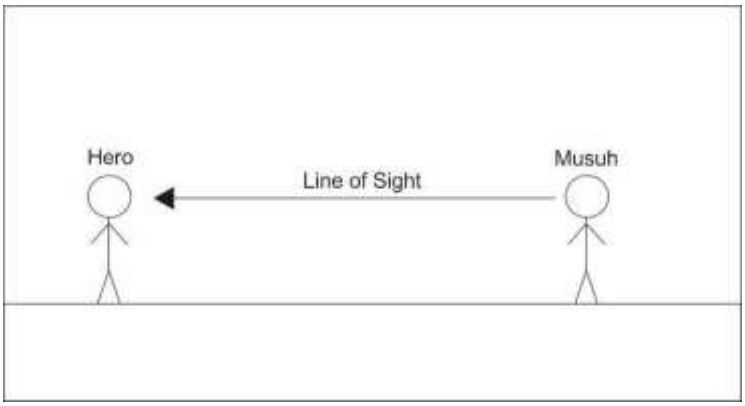
3. Line of Sight

Algoritma Line of Sight bekerja dengan membuat garis lurus dari NPC (Non-Player Character) ke player, jika garis berhasil menyentuh hero tanpa mengenai objek, maka Line of Sight berhasil dan itu bertanda NPC dapat melihat player. Tetapi jika garis terhalang oleh objek lain seperti tembok (Komarudin & Yuniarti, 2016)

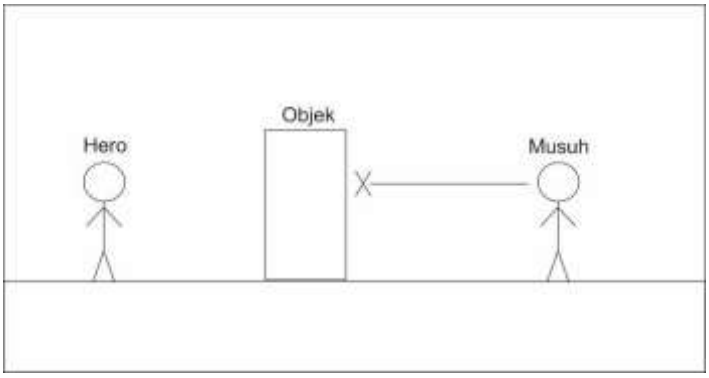


Gambar 4. *Line of Sight*

Dengan menghasilkan garis dengan tampilan acak, algoritma masih dapat berfungsi secara efektif. Atau, dengan menggambar garis langsung ke pahlawan NPC musuh, memastikan bahwa garis tersebut berada dalam bidang pandang musuh, dan kemudian menentukan apakah ada penghalang di sepanjang garis antara NPC musuh dan pahlawan. Agar NPC musuh dapat melihat pahlawan, garis pandang diperlukan dalam permainan. Proses menonton ini juga dipengaruhi oleh ada atau tidak adanya penghalang terhadap pandangan NPC musuh tentang pahlawan. Gambar 5 dan 6 menampilkan contoh ilustrasi.



Gambar 5. Musuh melihat Hero



Gambar 6. Musuh tidak melihat hero

Cara kerja algoritma Line of Sight adalah dengan membuat garis lurus dari musuh ke hero, jika garis berhasil menyentuh hero tanpa mengenai objek seperti gambar 5 maka Line of Sight berhasil dan menandakan bahwa musuh dapat melihat hero tersebut. Namun jika garis terhalang oleh benda lain seperti dinding pada gambar 6, itu menandakan bahwa musuh tidak bisa melihat hero tersebut.

4. *Platformer*

Game platformer merupakan subgenre dari genre game aksi. Hal yang paling umum pada genre ini adalah tombol untuk melompat. Kesulitasn dalam game platformer dapat disesuaikan karena hanya dipengaruhi dimensi kecepatan dan kerumitan lompatan. Tantangan utama dari game ini adalah melompat ke platform sambil menghindari lubang yang memisahkan antar platform. (Bahtiar et al., 2019). *Game Platformer* mengharuskan pemainnya berjalan dari kiri ke kanan layar secara linear untuk mencapai tujuan permainan. Ketika pemain bergerak layar akan bergerak mengikuti pemain atau juga bisa disebut *side-scrolling*, dengan cara ini ukuran dari sebuah *level* permainan akan bisa dibuat lebih besar daripada ukuran tampilan dari permainan tersebut (Iskandar et al., 2020).

5. Unified Modelling Language (UML)

Unified Modelling Language (UML**)** pemodelan secara visual yang digunakan untuk standarisasi dari spesifikasi, pemngabaran atau pendokumentasian dari sistem perangkat lunak atau software yang dibangun (Pakaya et al., 2020). Beberapa fungsi-fungsi yang ada di dalam UML adalah sebagai berikut

a) ***Use Case Diagram***

Use case digunakan untuk mengambarkan behavior dari sistem yang dibuat, juga mendeskripsikan interaksi aktor-aktor, funsi serta hak akses dari masing-masing aktor yang ada di di dalam sistem. (Kurniawan, 2018).

b) ***Class Diagram***

Class diagram mengambarkan struktur dari suatu objek. Diagram ini juga menggabarkan hubungan *class objek* yang menyususn sistem dan juga hubungan dengan *class objek* lainnya (Subhiyakto & Astuti, 2020).

c) ***Activity Diagram***

Activity diagram menggambarkan alur kerja atau aktivitas suatu proses bisnis yang ada di dalam sistem atau aplikasi yang dibangun(Aliman, 2021). *Ini adalah simbol yang ditemukan pada diagram aktivitas:*

6. Gamemaker Studio

GameMaker Studio merupakan aplikasi untuk membantu pengguna untukmembuat *game* baru dan inovatif, serta membuat ide prototype secepat dan seintuitif mungkin untuk target multi-platform. Penggunaan utamanya ditujukan untukpembuatan *game* 2D, tapi tidak menutup kemungkinanuntuk membuat *game* 3D (Kusuma et al., 2019). GameMaker Studio dilengkapidengan berbagai macam *tool* dan *editor* untuk membantupengguna mewujudkan idenya, dengan memungkinkan hasil akhir yang bisa dieksport ke berbagai *platform.*

7. Usability Test

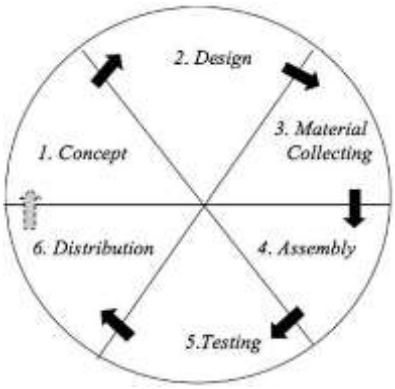
Usability Test adalah pengujian yang melakukan analisis kuantitatif yang menentukan seberapa mudah pengguna menggunakan antarmuka suatu aplikasi (Wedayanti et al., 2019). Suatu aplikasi disebut useable jika fungsi-fungsinya dapat dijalankan secara efektif, efisien, dan memuaskan. Usability sendiri

di definisikan dengan lima komponen :

a) *Learnability*, seberapa mudah pengguna dalam berinteraksi saat pertama kali mereka mencoba aplikasi?
b) *Efficiency*, setelah mempelajari desain aplikasi, seberapa cepat pengguna dapat menggunakan aplikasi?
c) *Memorability*, ketika pengguna kembali menggunakan aplikasi setelah jangka waktu tertentu, seberapa mudah mereka mengingat cara berinteraksi dengan aplikasi?
d) *Errors*, berapa banyak kesalahan yang dibuat olehpengguna, seberapa besar kesalahan tersebut, dan bagaimana mereka bisa menyelesaikan masalah tersebut?
e) *Satisfaction*, seberapa memuaskan dalam menggunakan aplikasi?

METODE

Motode pengembanga multimedia digunakan untuk mengembangkan aplikasi game "RUN!". Pada gambar 7 mendeskripsikan tahapan-tahan pengembangan sistem sebagai berikut :



Gambar 7. Metode Pengambangan multimedia

Adapau tahapan-tahapannya dapat dijelaskan sebagai berikut :

1. *Concept*
   Tahap ini menentukan tujuan, termasuk identifikasi audiens, jenis aplikasi (presentasi, interaktif, dll.), Dan spesifikasi umum, serta tujuan aplikasi (informasi, hiburan, pelatihan, dll.) dan spesifikasi umum. Pada tahap ini, aturan dasar desain, seperti ukuran aplikasi yang dimaksudkan, juga ditentukan.
2. *Design*
   Membuat spesifikasi secara rinci mengenai arsitektur proyek, gaya, dan kebutuhan material untuk proyek. Spesifikasi dibuat cukup rinci sehingga pada tahap berikut , yaitu material collecting dan assembly tidak diperlukan keputusan baru, tetapi menggunakan apa yang sudah ditentukan pada tahap design.
3. *Material Collecting*
   Mengumpulkan materi seperti gambar clipart, audio, dan animasi, serta membuat gambar grafis, foto, dan audio untuk tahap selanjutnya. Bahan yang diperlukan untuk multimedia dapat diperoleh dari sumber-sumber seperti perpustakaan, bahan yang sudah ada sebelumnya dari pihak ketiga, atau manufaktur khusus yang dilakukan oleh pihak ketiga.
4. *Assembly*
   Seluruh proyek multimedia dibuat pada tahap ini. Membuat berdasarkan storyboard yang dihasilkan selama fase desain. Pembuatan dilakukan dengan memasukkan data yang digunakan untuk berbagai tampilan dan menemukan layar dalam urutan yang tepat.
5. Testing
   Pentingnya pengujian perangkat lunak dan implikasinya yang mengacu pada kualitas perangkat lunaktidak selalu ditekan karena melibatkan sederetan aktivitasproduksi dimana peluang terjadinya kesalahan manusia sangat besar dan karena ketidakmampuan manusia untuk melakukan dan berkomuinikasi dengan sempurna maka pengembangan perangkat lunak diiringi dengan aktivitas jaminan kualitas.
6. Distribution

Setelah pengujian selesai, pada tahapan ini akan dilakukan proses pendistribusian perangkat lunak yang telah dibuat, bisa menggunakan berbagai media seperti CD/DVD, *Flash drive*, atau secara *Online*.

HASIL

Berikut adalah implementasi metode pengembangan multimedia :

1. **Pada tahapan concept**

   Menentukan alur game, penentuan aturan skor seperti Pemain dapat mengumpulkan *item* dalam permainan yang akan memberi pemain skor dan atau nyawa. Koin Kuning +100 skor, Koin Ungu +200 skor, Nyawa +400 skor dan +1 nyawa. Skor pemain akan meningkat 100 ketika mereka mengalahkan musuh, tetapi jika mereka terkena serangan musuh, hidup mereka akan berkurang 1 dan skor mereka akan berkurang 100. Jika pemain kehilangan seluruh hidup mereka, mereka harus memulai dari awal. plot permainan dengan menggambarkan penculikan karakter bernama Tony oleh sekelompok penyihir dari dunia lain. Tony diculik sebagai pengorbanan untuk meningkatkan kekuatan Zargothrax, pemimpin kelompok penyihir. Tony dipenjara, tetapi dia melarikan diri karena penjaga yang ceroboh. Tony bertemu dengan beberapa musuh, termasuk Slime, Mage, dan Fire Shooter, saat ia mencari portal yang menghubungkan dunia tempat ia dibawa ke dunianya sendiri. Ketika Tony menemukan portal, ia juga bertemu Zargothrax, yang mungkin menculik teman Tony, Andy. Zargothrax kembali ke planet asal Tony untuk menculik lebih banyak orang. Sebelum berangkat, Zargothrax memerintahkan bawahannya, Great Mage, untuk mengalahkan Tony dan mengunci Tony dan Andy di markas masing-masing, namun Tony mampu mengalahkan Great Mage. Tony berhasil menyelamatkan Andy, dan mereka kemudian kembali melalui portal ke dunia asal mereka.

2. **Pada tahapan Desain**

   Pada tahapan ini membuat membuat desain dari konsep yang sudah dibuat dengan mengunakan UML. Dimana juga penerapan algoritma *Line of Sight* dan *Finite StateMachine* pada *Game Platformer* "RUN!" dijelaskan .

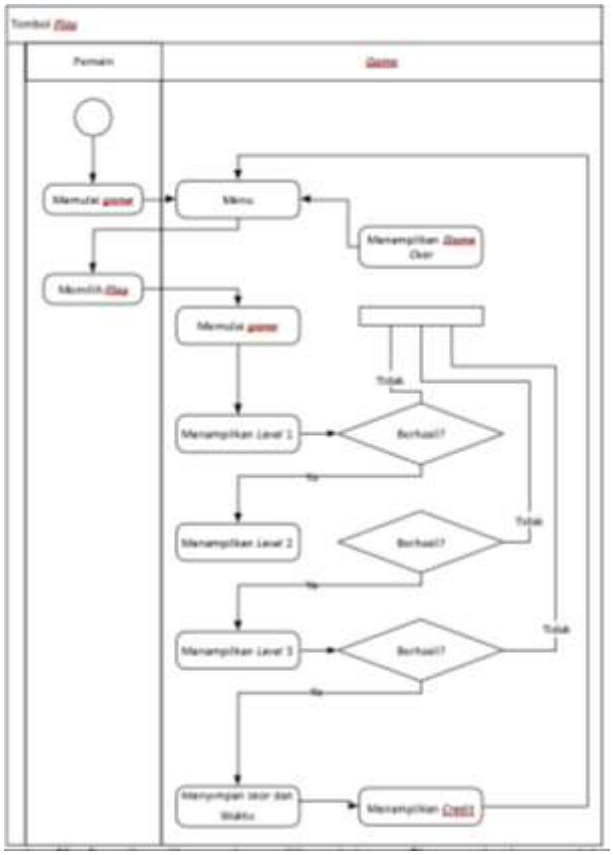
   a) Use Case Diagram

   Perancangan *Use Case Diagram* digunakan untuk menggambarkan apa yang akan terjadi ketika pemain berinteraksi dengan *Use Case* yang ada pada aplikasi game. Berdasarkan *Use Case* diatas jika pemain memilihtombol menu *Play*, pemain akan memulai *game*, jikapemain memilih tombol menu *Highscore* pemain akan diarahkan ke papan *Highscore*. Tombol menu *Setting* akan menampilkan pengaturan untuk volume suara, tombol menu *credit* akan menampilkan pesan dari pembuat *game*, dan tombol menu *Exit* akan menutup aplikasi *game*.

Gambar 8. *Use Case Diagram* "RUN!"

Berdasarkan *Use* Case diatas jika pemain memilih tombol menu *Play*, pemain akan memulai *game*, jika pemain memilih tombol menu *Highscore* pemainakan diarahkan ke papan *Highscore*. Tombol menu *Setting* akan menampilkan pengaturan untuk volume suara, tombol menu *credit* akan menampilkan pesan dari pembuat *game*, dan tombol menu *Exit* akan menutup aplikasi *game*.

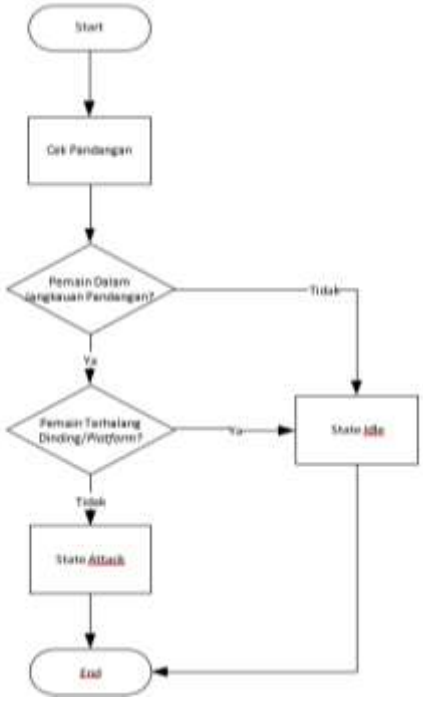
b) Actifity Diagram



Gambar 9. *Activity Diagram* "RUN!"

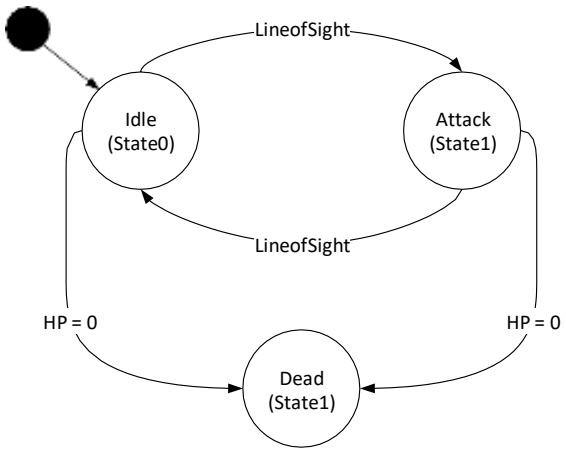
c) Desain *Flowchart Line of Sight*

Berikut adalah gambaran flowchart yang menggambarkan bagaimana proses alur *Lineof Sight* yang diterapkan pada *game* "RUN!".



Gambar 10. Desain *Flowchat Line of Sight*

d) Diagram *Finite State Machine* NPC Musuh



Gambar 11. Diagram *Finite State Machine* NPC Musuh

Gambar 11 adalah diagram *Finite State Machine* yang menggambarkan bagaimana *Finite State Machine* pada *game* "RUN!" ini bekerja, perubahan *state* akan terjadi tergantung dari *output Line of Sight* atau saat kondisi terpenuhi.

3. Pada tahapan *Material Collection*

Pada tahap ini melakukan pengumpulan bahan yang diperlukan dalampembuatan "RUN!" seperti berikut :

a) Gambar yang akan digunakan untuk membuat game, seperti paku, digunakan untuk mewakili karakter, ubin digunakan untuk mewakili lingkungan, seperti tanah dan rumput, dan sprite digunakan untuk mewakili musuh.
b) Audio adalah suara, yang akan digunakan sebagai media suara dalam game ini, seperti musik latar, dan efek suara adalah suara yang dihasilkan saat pemain berjalan, melompat, atau menyerang musuh.

4. Pada Tahapan Assambly

Pada tahapan ini, peneliti melakukan coding program untuk membuat game, berikut tampilan game yang sdauh selesai dibuat. Gambar 12 tampilan game pada saat di level 1 permaianan, ada 3 level permaiana yang dibuat dengan perbedaan tingkat kesulitan.

Gambar 12. Tampilan Game

5. Pada Tahapan *Testing*

Pengujian *Usability Test* dilakukan untuk menguji sisi fungsional aplikasi *game* dari sisi pemain. Pengujian dilakukan secara objektif dan *game* diuji oleh 30orang responden, Data yang diperoleh dari responden kemudian dianalisis menggunakan analisis interval Agar data dapat dihitung dalam bentuk kuantitatif, hasil kuesioner dari yang diterima dari responden akan diberi skor likert lalu dihitung dengan rumus perhitungan dibawah ini :

Skor Maksimum = Jumlah Responden x Skor tertinggi likertIndeks (%) =
(Total Skor / Skor Maksimum) x 100

Berikut adalah tabel untuk menentukan hasil dari perhitungan indeks :

Tabel 2 Interval Penilaian

Indeks	Interval Penilaian
0% - 19.99%	Sangat Tidak Setuju
20% - 39.99%	Tidak Setuju
40% - 59.99%	Netral
60% - 79.99%	Setuju
80% - 100%	Sangat Setuju

Secara keseluruhan *game* berfungsi sebagai mestinya sebuah *game*.
Skor Maksimum = 150
indeks (%) = (138 / 150) x 100 i
Indeks yang didapatkan (%) = 92 %

6. **Pada Tahapan *Distribution***
   Pada tahapan ini *game* sudah selesai dibuat dan diuji, dan *game* sudah siap dikemas dan didistribusikan baik secara fisik (CD/DVD) atau *online* (pengunduhan mandiri).

PEMBAHASAN

pada gambar 13 Penerapan *Line of Sight* terlihat screenshot coding game dimana tahapan kerja dari penerapan ini adalah pertama akandilakukan pengecekan apakah pemain ada dalam pandangan NPC musuh, jika tidak NPC musuh akan masuk ke *state idle*, tapi jika ya akan dilakukan pembuatan garis dari NPC musuh ke *hero*, jika di garis antara NPC musuh dan *hero* ada objek dinding atau platform yang menghalangi, maka NPC musuh akan masuk ke *state idle*, tapi jika tidak ada yang menghalangi,maka NPC musuh akan masuk ke *state attack* dan akan terus berada di *state attack* selama *hero* masih ada dalam *Line of Sight* dari NPC musuh.

Gambar13. *Code Line of Sight*

Pada gambar 14 adalah screenshot penerapan *code Finite State Machine* yang digunakan pada NPC Musuh. Disini NPC musuh memiliki 2 state yaitu *state attack* dan *state idle*, perubahan state akan dilakukan sesuai dengan outputatau keluaran dari algoritma *Line of Sight* yang juga berjalan dalam NPC musuh. Jika *Line of Sight* NPC musuh dapat melihat pemain, maka NPC musuh akan masuk ke dalam *state attack* dan mulai

menyerang pemain, dan sebaliknya jika *Line of Sight* NPC musuh tidak dapat melihat pemain, maka NPC musuh akan berada di dalam *state idle* dan akan berpatroli di areanya.

```
LineofSight();
slimeanimation();
slimecollision();

switch (state)
{
    case e_state.idle:
    {
        hsp = walksp;
        vsp = (min(7, vsp + 0.05));
        show_debug_message("idle");
        break;
    }

    case e_state.attack:
    {
        dir = sign(oPlayer.x - x);
        hsp = dir * 4;
        vsp = (min(7, vsp + 0.05));
        show_debug_message("attack");
        break;
    }

}
```

Gambar 14. *Code Finite State Machine*

KESIMPULAN

Berdasarkan uarain diatas dapat ditarik kesimpulan *Game* Platformer "RUN!" ini menerapkan metode *Line of Sight* dan berhasil dalam penerapannya. NPC(*Non-Playable Character*) yang mengimplementasikan metode ini dapat mendeteksi *hero* selama masih dalam jarak pandang dan tidak terhalang objek dinding atau *platform*. Menerapkan metode *Finite State Machine* dan berhasil dalam penerapannya. NPC (*Non-Playable Character*) yang menerapkan metode ini dapat berganti *state* dari *idle* ke *attack* atau sebaliknya dengan *output* dari metode *Line of Sight* dan hasil pengujian usability testing *Game Platformer* "RUN!" ini dapat dapat menjalankan fungsi-fungsi komponennya dengan baik dan responden dapat memahami antar muka aplikasi *game* ini dengan mudah.

REFERENSI

Agustina, C. (2015). Aplikasi game pendidikan berbasis android untuk memperkenalkan pakaian adat Indonesia. *Indonesian Journal on Software Engineering (IJSE)*, *1*(1), 1–8.

Aliman, W. (2021). Perancangan perangkat lunak untuk menggambar diagram berbasis android. *Syntax Literate; Jurnal Ilmiah Indonesia*, *6*(6), 3091–3098.

Asmiatun, S., & Putri, A. N. (2017). *Belajar Membuat Game 2D dan 3D Menggunakan Unity*. Deepublish.

Bahtiar, A., Muhima, R. R., & Rachman, A. (2019). Penerapan Model Spiral Pada Rancang Bangun Game Platformer. *Prosiding Seminar Nasional Sains Dan Teknologi Terapan*, *1*(1), 601–606.

Garapati, P., & Musala, S. (2020). Moore and Mealy Negative Edge detector A VHDL Example for Finite State Machine. *2020 International Conference on Communication and Signal Processing (ICCSP)*, 1159–1161. https://doi.org/10.1109/ICCSP48568.2020.9182310

Henry, S. (2013). *Cerdas Dengan Games*. Gramedia Pustaka Utama.

Iskandar, U. A. S., Diah, N. M., & Ismail, M. (2020). Identifying artificial intelligence Pathfinding algorithms for Platformer games. *2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, 74–80.

Kurniawan, T. A. (2018). Pemodelan use case (UML): evaluasi terhadap beberapa kesalahan dalam praktik. *J. Teknol. Inf. Dan Ilmu Komput*, *5*(1), 77.

Kusuma, R. T., Mulyani, A., & Rianto, H. (2019). Rancang Bangun Game "Legends of Spaceship" Menggunakan

Game Maker Studio. *JISICOM (Journal of Information System, Informatics and Computing)*, *3*(2), 37–44.

Pakaya, R., Tapate, A. R., & Suleman, S. (2020). Perancangan aplikasi penjualan hewan ternak untuk qurban dan aqiqah dengan metode Unified Modeling Language (UML). *Jurnal Technopreneur (Jtech)*, *8*(1), 31–40.

Prabowo, P. (2020). *Membangun Side Scrolling Game" Petualangan Yada Mencari Benda Pusaka" Menggunakan Metode Algoritma A*(A Star) Dan Algoritma Line Of Sight*. STMIK WIDYA CIPTA DHARMA.

Rahadian, M. F., Suyatno, A., & Maharani, S. (2017). *Penerapan metode finite state machine pada game "The Relationship."*

Rizal Dwi Saputro, R. D. S., Patmi Kasih, P. K., & Siti Rochana, S. R. (2022). *Implementasi Pathfinding A-Star Dan Line Of Sight Pada Game Edukasi Pembelajaran Online Berbasis Android Untuk Mengurangi Kejenuhan Belajar*. Universita Nusantara PGRI Kediri.

Sanjaya, T. (2017). *Membangun "RPG (Role Playing Game)"Adventure Of Ren: Edenweid Island Dan Pengembangan Game Agent Berbasis Finite State Machine."* Teknik Informatika.

Subhiyakto, E. R., & Astuti, Y. P. (2020). Aplikasi Pembelajaran Class Diagram Berbasis Web Untuk Pendidikan Rekayasa Perangkat Lunak. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, *11*(1), 143–150.

Wedayanti, N. L. P. A., Wirdiani, N. K. A., & Purnawan, I. K. A. (2019). Evaluasi Aspek usability pada aplikasi Simalu menggunakan metode usability testing. *J. Ilm. Merpati (Menara Penelit. Akad. Teknol. Informasi)*, *7*(2), 113.