RANCANG BANGUN GAME BERGENRE ROGUELIKE ADVENTURE "TALES OF SELEINE"

TUGAS AKHIR

Disusun Untuk Memenuhi Syarat Kelulusan Program Strata I Pada
Sekolah Tinggi Manajemen Informatika Dan Komputer
(STMIK) Palangkaraya

OLEH :
RIFA'I AL AMIN
NIM C1655201015
PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
2021

RANCANG BANGUN GAME BERGENRE ROGUELIKE ADVENTURE "TALES OF SELEINE"

TUGAS AKHIR

Disusun Untuk Memenuhi Syarat Kelulusan Program Strata I Pada
Sekolah Tinggi Manajemen Informatika Dan Komputer
(STMIK) Palangkaraya

OLEH :
RIFA'I AL AMIN
NIM C1655201015
PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
2021

LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama Mahasiswa : **RIFA'I AL AMIN**

NIM : **C1655201015**

Menyatakan bahwa tugas akhir dengan judul :

RANCANG BANGUN GAME BERGENRE ROGUELIKE ADVENTURE "TALES OF SELEINE"

Adalah hasil karya saya dan bukan merupakan duplikasi sebagian atau seluruhnya dari karya orang lain, kecuali bagian yang sumber informasi dicantumkan.

Pernyataan ini dibuat dengan sebenar-benarnya secara sadar dan bertanggungjawab dan saya bersedia menerima sanksi pembatalan tugas akhir apabila terbukti melakukan duplikat terhadap tugas akhir atau karya ilmiah lain yang sudah ada.

Palangka Raya, 28 Januari 2021

ıg membuat Pernyataan,



RIFA'I AL AMIN

PERSETUJUAN

RANCANG BANGUN GAME BERGENRE ROGUELIKE ADVENTURE "TALES OF SELEINE"

Tugas Akhir ini telah disetujui untuk diujikan
pada Tanggal 28 Januari 2021

Pembimbing I,	Pembimbing II,
Lili Rusdiana, M.Kom.	Susi Hendartie, M.Kom.
NIK. 198707282011007	NIK. 197803202008001

Mengetahui
Ketua STMIK Palangkaraya,

Suparno, M.Kom
NIK. 196901041995105

PENGESAHAN

RANCANG BANGUN GAME BERGENRE ROGUELIKE ADVENTURE "TALES OF SELEINE"

Tugas Akhir ini telah Diuji, Dinilai dan Disahkan
Oleh Tim Penguji pada Tanggal 30 Januari 2021

Tim Penguji Tugas Akhir :

1. Herkules, S.Kom., M.Cs. ..... 
   Ketua
2. Rudini, M.Pd. ..... 
   Sekretaris
3. Elia Zakaria, M.T. .....
   Anggota
4. Lili Rusdiana, M.Kom. ..... 
   Anggota
5. Susi Hendartie, M.Kom. ..... 
   Anggota

MOTTO DAN PERSEMBAHAN

Disiplin adalah kunci.

Ku persembahkan untuk :

- Ayah dan Ibu yang telah mengisi dunia saya dengan begitu banyak kebahagiaan sehingga seumur hidup tidak cukup untuk menikmati semuanya. Terima kasih atas semua cinta yang telah ayah dan ibu berikan kepada saya.
- Adik dan keluarga besar yang telah memberikan dukungan dan doa sehingga tugas akhir ini dapat terselesaikan
- Teman – teman nongkrong coffee terimakasih karena selalu mengingatkan untuk mengerjakan skripsi dan membantu dalam proses pengerjaan skripsi
- Dosen – dosen STMIK Palangka Raya, karena telah membimbing dari awal kuliah hingga sekarang ini.

INTISARI

Rifa'i Al Amin, C1655201015, 2021. Rancang bangun *game bergenre roguelike adventure tales of seleine*, Pembimbing I Lili Rusdiana, M.Kom., Pembimbing II Susi Hendartie, M.Kom.

Game roguelike adventure Tales of Seleine merupakan permainan yang bercerita tentang *Ancient Mage* bernama Seleine, seorang penyihir ambisius yang meninggalkan rumahnya untuk pergi ke *Thelkan Cave* untuk menemukan *Kwolok's Ring*, yang dimana di dalam perjalanannya mencari *Kwolok's Ring* tersebut sudah pasti banyak masalah menunggu, musuh-musuh yang menjaga *Kwolok's Throne Room*, Lantai *Dungeon* yang selalu berubah-ubah dan banyaknya jebakan didalam *Dungeon*.

Tujuan penelitian ini yaitu membuat rancang bangun *game* bergenre *roguelike adventure Tales of Seleine* yang menyenangkan tetapi juga sulit untuk ditaklukan. Aplikasi ini dibuat menggunakan *Unity* 2020 dengan bahasa program *C#.*

Metode yang digunakan pada tugas akhir ini adalah metode pengumpulan data studi pustaka dan kuisioner dan metode pengembangan perangkat lunak model MDLC, serta metode analisis kelemahan sistem menggunakan metode SWOT.

Dari pengujian yang telah dilakukan hasil dari aplikasi yaitu dapat dijalankan pada *Windows* maupun *Mac Os* dan mudah digunakan oleh pengguna, dengan tombol pada semua *scene* yang jelas dan mudah dimengerti. Setiap fitur pada aplikasi berfungsi dengan benar dan desain objek dalam aplikasi menarik. Hasil responden pengguna adalah 86,74%.

Kata Kunci: Aplikasi, *Game, Roguelike, Adventure, Tales of Seleine, Unity*

ABSTRACT

Rifa'i Al Amin, C1655201015, 2021. Design and build a game genre roguelike adventure tales of seleine, Advisor I Lili Rusdiana, M.Kom., Advisor II Susi Hendartie, M.Kom.

The game roguelike adventure Tales of Seleine is a game that tells the story of an Ancient Mage named Seleine, an ambitious witch who left his home to go to Thelkan Cave to find Kwolok's Ring, where on his way to find the Kwolok's Ring , there are definitely many problems waiting, enemies guarding Kwolok's Throne Room, the Dungeon Floor which is always changing and many traps in the dungeon.

The purpose of this research is to design and build a game genre roguelike adventure tales of seleine which is fun but also difficult to conquer. This application was created using Unity 2020 with the program language C#.

The methods used in this final project are literature study data collection methods and questionnaires and MDLC model software development methods, as well as systems weakness analysis methods using the SWOT method.

From the tests that have been done, the results of the application can be run on Windows and Mac Os and are easy to use by users, with buttons on all scenes that are clear and easy to understand. Every feature in the application works properly and the object design in the application is interesting. The result of user respondents is 86,74%

Keywords: Application, Game, Roguelike, Adventure, Tales of Seleine, Unity

KATA PENGANTAR

Puji syukur Alhamdulillah penulis panjatkan kehadirat Allah SWT atas berkat, rahmat, taufik, dan hidayah-Nya, penyusunan skripsi yang berjudul "Rancang Bangun Game Bergenre Roguelike Adventure "Tales of Seleine" dapat diselesaikan dengan baik. Shalawat serta salam senantiasa tercurah kepada junjungan Nabi besar Muhammad SAW beserta keluarga, sahabat, kerabat, dan pengikut beliau hingga akhir zaman.

Tugas akhir ini disusun sebagai memenuhi syarat dalam memperoleh gelar Sarjana Komputer di Sekolah Tinggi Managemen Informatika dan Komputer (STMIK) Palangkaraya.

Selanjutnya ucapan terimakasih sebesar-besarnya penulis sampaikan kepada semua pihak yang telah membantu, membimbing dan memotivasi dalam menyelesaikan penulisan tugas akhir ini, yaitu kepada:

1. Suparno, M.Kom selaku Ketua Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya.
2. Lili Rusdiana, M.Kom selaku Pembimbing I yang telah meluangkan waktunya untuk memberikan bimbingan dalam penyelesaian tugas akhir ini.
3. Susi Hendartie, M.Kom selaku dosen Pembimbing II dalam penulisan tugas akhir ini, yang juga banyak memberikan saran dan masukan dalam penulisan tugas akhir ini.
4. Kedua orang tua, adik dan keluarga atas dukungannya sehingga penulis dapat menyelesaikan penulisan tugas akhir.

5. Seluruh responden yang berpartisipasi dalam penelitian yang dilaksanakan.

Penulis mengharapkan masukan, saran dan kritik yang membangun untuk menyempurnakan tugas akhir ini. Semoga tugas akhir ini bermanfaat bagi penulis dan pembaca khususnya untuk menambah ilmu.

Palangka Raya, 28 Januari 2021

Penulis

DAFTAR ISI

LEMBAR PERNYATAAN ii
PERSETUJUAN iii
PENGESAHAN iv
MOTTO DAN PERSEMBAHAN v
INTISARI vi
ABSTRACT vii
KATA PENGANTAR viii
DAFTAR ISI x
DAFTAR TABEL xiii
DAFTAR GAMBAR xiv
BAB I PENDAHULUAN 1
1.1 Latar Belakang Masalah 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan dan Manfaat 4
1.4.1 Tujuan 4
1.4.2 Manfaat 4
1.5 Sistematika Penulisan 5
BAB II LANDASAN TEORI 6
2.1 Tinjauan Pustaka 6
2.1.1 Penelitian Yang Relevan 6
2.2 Kajian Teori 8
2.2.1 Pengertian Game 8
2.2.2 Jenis-jenis Genre Game 8

2.2.3 Roguelikes 11
2.2.4 Elemen Games 12
2.2.5 Kecerdasan Buatan (Artificial Intellegence) 14
2.2.6 Finite State Machines (FSM) 15
2.2.7 Multimedia Development Life Cycle (MDLC) 16
2.2.8 C# 18
2.2.9 UML (Unified Modeling Language) 19
BAB III METODE PENELITIAN 23
3.1 Teknik Pengumpulan Data 23
3.2 Analisis 23
3.2.1 Analisis Proses 23
3.2.2 Analisis Kelemahan Sistem 25
3.2.3 Analisis Kebutuhan 26
3.2.4 Analisa Kelayakan Sistem 28
3.3 Desain Sistem 29
3.3.1 Storyline 29
3.3.2 Desain Proses 29
3.3.3 Desain Perangkat Lunak 37
BAB IV HASIL DAN PEMBAHASAN 40
4.1 Hasil 40
4.1.1 Implementasi Program 40
4.1.2 Pengujian Sistem dan Uji Coba 45
4.1.3 Manual Program 49
4.1.4 Manual Instalasi 55
4.2 Pembahasan 56

4.2.1 Pembahasan Listing Program..... 56
4.2.2 Pembahasan Hasil Respon Pengguna..... 73
BAB V KESIMPULAN DAN SARAN..... 77
5.1 Kesimpulan 77
5.2 Saran..... 78
DAFTAR PUSTAKA 79
DAFTAR LAMPIRAN..... 82

DAFTAR TABEL

Tabel 1. Penelitian Yang Relevan 6

Tabel 2. Use Case Diagram (Hendini, 2016) 21

Tabel 3. Activity Diagram (Hendini, 2016) 22

Tabel 4. Kebutuhan Perangkat Keras 27

Tabel 5. Kebutuhan Perangkat Lunak 27

Tabel 6. Pengujian buka aplikasi 45

Tabel 7. Pengujian halaman menu utama 45

Tabel 8. Pengujian halaman play 46

Tabel 9. Pengujian Halaman About 47

Tabel 10. Pengujian Halaman Settings 47

Tabel 11. Pengujian halaman credits 48

Tabel 12. Pengujian win scene 48

Tabel 13. Pengujian lost scene 48

Tabel 14. Skor Pilihan Jawaban 73

Tabel 15. Jawaban Responden 74

DAFTAR GAMBAR

Gambar 1. Konsep The Magic Circle (Adams, 2013) 13

Gambar 2. Diagram State Sederhana (Setiawan, 2006)..... 16

Gambar 3. MDLC (Setiawan, et al., 2017) 17

Gambar 4. Use Case Diagram..... 30

Gambar 5. *Diagram Activity Start* 30

Gambar 6. *Diagram Activity About*..... 31

Gambar 7. *Diagram Activity Settings*..... 32

Gambar 8. Diagram Activity Credits 33

Gambar 9. *Diagram Activity Exit*..... 34

Gambar 10. Storyboard halaman 1 35

Gambar 11. Storyboard halaman 2 35

Gambar 12. Storyboard halaman 3 36

Gambar 13. Tampilan Splash Screen..... 37

Gambar 14. Tampilan Menu Utama..... 38

Gambar 15. Tampilan Gameplay 38

Gambar 16. Tampilan Settings..... 39

Gambar 17. Proses pembuatan sprites pohon pada Adobe Photoshop CC 41

Gambar 18. Proses pembuatan splash screen di Adobe After Effects 41

Gambar 19. Proses render splash screen..... 42

Gambar 20. proses pembuatan musik di FL Studio 42

Gambar 21. proses render musik..... 43

Gambar 22. Worksheet Unity 44

Gambar 23. Tampilan splash screen 50
Gambar 24. Tampilan menu utama 50
Gambar 25. Tampilan halaman about 51
Gambar 26. Tampilan menu settings 52
Gambar 27. Tampilan credits 53
Gambar 28. Tampilan gameplay Tales of Seleine 53
Gambar 29. Tampilan victory scene 54
Gambar 30. Tampilan lost scene 54
Gambar 31. Page Itch.io Tales of Seleine 55
Gambar 32. Listing Program Menu 57
Gambar 33. Listing program suara 58
Gambar 34. Listing program grafik 59
Gambar 35. Listing program fullscreen 59
Gambar 36. Listing program resolusi 60
Gambar 37. Variabel player move 60
Gambar 38. Variabel dash 61
Gambar 39. Void start player 61
Gambar 40. Void update player 61
Gambar 41. Void fixed update player 62
Gambar 42. Variabel darah player 62
Gambar 43. Void mengurangi darah player 62
Gambar 44. Merubah sprites hearth player 63
Gambar 45. Menyembuhkan player dengan item 63

Gambar 46. Merubah senjata dari drop item 63
Gambar 47. Spawn random objects 64
Gambar 48. Program mengejar player 64
Gambar 49. Darah musuh berkurang jika kena serangan player 65
Gambar 50. Variabel dan void update melee enemy 65
Gambar 51. Program melee enemy menyerang 66
Gambar 52. Program pergerakan range enemy 66
Gambar 53. Fungsi menyerang range enemy 66
Gambar 54. Variabel summoner dan minion 67
Gambar 55.Void start program summoner 67
Gambar 56. Bergerak ke tempat random 67
Gambar 57. Menyerang player jika dalam jangkauan 68
Gambar 58. Program menyerang player 68
Gambar 59. Memanggil minion 68
Gambar 60. State machine boss 68
Gambar 61. State Machine Behaviour Boss 69
Gambar 62. Variabel patrol behaviour 69
Gambar 63. Void OnStateEnter patrol behaviour 69
Gambar 64. Void OnStateUpdate patrol behaviour 69
Gambar 65. Variabel chase behaviour 70
Gambar 66. Void OnStateEnter chase behaviour 70
Gambar 67. Void OnStateUpdate chase behaviour 70
Gambar 68. Variabel senjata 70

Gambar 69. Void start senjata 71
Gambar 70. Void update senjata 71
Gambar 71. Variabel peluru 71
Gambar 72. Void start peluru 71
Gambar 73. Void update peluru 72
Gambar 74. Menimbulkan efek ledakan 72
Gambar 75. Mendeteksi objek dengan tag enemy dan boss 72
Gambar 76. Hasil Kuesioner 76

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Sebagai sesuatu yang dimulai lebih dari setengah abad yang lalu dan sekarang telah menjadi salah satu industri terbesar di dunia. *Video game* telah mendapatkan tempat tersendiri di dalam bidang hiburan. Pendapatan *game* digital mencapai $ 109,4 miliar pada tahun 2019, naik 3% dari tahun ke tahun (SuperData, 2019). Ini memberi tahu orang-orang tentang seberapa besar industri ini telah berkembang. *Video game* pertama adalah pertandingan tenis yang sangat sederhana diciptakan pada bulan oktober tahun 1958 oleh Fisikawan William Higinbotham. *Video game* menjadi sarana popular oleh masyarakat dunia terutama dikalangan remaja dan dewasa muda untuk menghibur diri guna mengurangi stress dari beratnya tuntutan dalam berbagai aspek kehidupan.

Berdasarkan Quantic Foundry sebuah perusahaan riset pasar yang berfokus pada motivasi *gamer*, membagi motivasi *gamer* menjadi 6 kelompok utama, dengan masing-masing kelompok memiliki dua submotivasi serupa. Enam kelompok utama tersebut adalah *Action, Social, Mastery, Achievement, Immersion, Creativity*. Keemam kategori ini mewakili motivasi paling umum dari pengguna yang bermain *game* digital (Yee, 2015).

Sebuah studi berskala besar di perguruan tinggi keadaan psikologis Columbia University meneliti korelasi pada anak-anak usia 6 sampai 11 tahun. Selama survei ini, 3195 anak-anak dan orang tua mereka menghitung rentang jam yang sama

perminggu ketika anak-anak suka bermain *game*, kemudian para pakar dan akademis mengisi kuesioner yang berkaitan dengan fungsi intelektual, sosial dan emosional setiap anak. Hasilnya adalah bahwa penggunaan *video game* tidak dikaitkan dengan peningkatan resiko masalah kesehatan mental. Sebaliknya, data yang disajikan menunjukan bahwa *video game* adalah faktor pelindung, terutama masalah hubungan teman sebaya bagi anak-anak yang paling terlibat dalam *video game*. Akhirnya, *video game* tampaknya terhubung dengan fungsi intelektual dan prestasi akademik yang lebih baik (Kovess-Masfety, et al., 2016). Dengan adanya survei ini bisa dikatakan *game* memberikan manfaat pengembangan diri untuk para penggunanya.

Seiring dengan perkembangan industri *game*, memungkinkan para *developers game* untuk menciptakan *game* apapun yang mereka inginkan. Untuk itu, dalam beberapa tahun terakhir, *genre* telah diciptakan untuk mengakomodasi *game* yang berada diluar klasifikasi *game* tradisional. Fakta bahwa batas *genre* itu sendiri selalu berubah selama karya baru diproduksi (Wolf, 2008).

Roguelike *game* adalah *game RPG* pemain tunggal, berbasis giliran, tingkat kesulitan yang tinggi, menampilkan lingkungan acak dan kematian permanen. Pemain mengambil peran seorang *hero*, yang memasuki lingkungan permainan (seringkali *dungeon*) dengan tujuan yang sangat sulit. Mencapai tujuan membutuhkan banyak keterampilan, pengalaman *game* dan mungkin sedikit keberuntungan (Cerny & Dechterenko, 2015). Roguelike sendiri merupakan *subgenre* dari *Role-Playing Games*.

Berdasarkan data-data diatas, penulis memilih untuk mengembangkan *game bergenre Roguelike Adventure* dengan judul "Tales of Seleine". Dimana pengguna akan bermain sebagai seorang penyihir bernama Seleine untuk menemukan Kwolok's Ring. Diharapkan dengan adanya *game* ini para pengguna memiliki aplikasi *game* tambahan untuk dimainkan dan memiliki motivasi untuk menyelesaikan *game* ini serta untuk pengembangan diri, mengurangi stress dan menjadi sarana hiburan. *Game* "Tales of Seleine" memiliki *storyline* yang terstruktur, serta karakter, musik dan *interface* yang menarik.

1.2 Rumusan Masalah

Masalah yang dapat dirumuskan berdasarkan latar belakang tersebut, adalah bagaimana cara membangun *game bergenre roguelike adventure* "tales of seleine"?

1.3 Batasan Masalah

Batasan-batasan yang ada pada perancangan *game* ini adalah:

1. *Game* dirancang untuk dapat dijalankan pada sistem komputer atau berbasis *desktop* dan tidak dapat dijalankan pada *smartphone*.
2. *Game* dirancang dengan menggunakan grafik 2D.
3. Game Engine yang digunakan adalah Unity dan menggunakan Bahasa pemrograman C#.
4. Tidak memerlukan koneksi internet atau menggunakan sistem *offline*.
5. *Gameplay* bersifat *single-player* dan *permanent death*.
6. Pemain hanya bisa menyimpan permainan sebanyak 1 permainan.
7. Survei game dilakukan menggunakan kuesioner online.

1.4 Tujuan dan Manfaat

1.4.1 *Tujuan*

Adapun yang menjadi tujuan pembuatan tugas akhir ini adalah untuk membangun sebuah *game* "Tales of Seleine" yang menyenangkan tetapi juga sulit untuk ditaklukan.

1.4.2 *Manfaat*

Manfaat dari pembuatan tugas akhir ini adalah:

1. Manfaat Bagi Penulis

   Manfaat yang didapat penulis adalah bertambahnya wawasan dan pengalaman langsung tentang pembuatan *game*, serta syarat kelulusan Strata 1 (S1) Program Studi Teknik Informatian di STMIK Palangka Raya.

2. Manfaat bagi pengguna

   Bertambahnya aplikasi *game* pilihan untuk dimainkan pada sistem komputer, serta memberikan manfaat pengembangan diri dengan melatih resource management dalam bermain *game* ini dan sebagai sarana hiburan.

3. Manfaat bagi STMIK Palangka Raya

   Sebagai penambah literatur pustaka pada perpustakan STMIK Palangka Raya serta dapat digunakan sebagai salah satu referensi atau kajian untuk mahasiswa STMIK Palangka Raya yang ingin mengembangkan *game bergenre roguelike* ataupun *genre* lain.

1.5 Sistematika Penulisan

BAB I PENDAHULUAN

Pada bab ini dijelaskan mengenai latar belakang penulisan secara umum dan ruang lingkup dari yang ditulis dan dirancang, rumusan masalah, batasan masalah, tujuan dan manfaat yang hendak dicapai, serta sistematika yang digunakan penulis.

BAB II LANDASAN TEORI

Pada bab ini dijelaskan teori-teori yang digunakan dan mendukung dalam penelitian dan perancangan.

BAB III METODE PENELITIAN

Pada bab ini dijelaskan mengenai implementasi dan cara pengoperasian program aplikasi, juga evaluasi aplikasi tersebut.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini dijelaskan pembahasan hasil dari penelitian yang dilakukan peneliti tentang aplikasi ini. Sebagai laporan observasi tentang penelitian terhadap sesuatu. Hasil pembahasan sebagai pertimbangan atau acuan, untuk dijadikan sebagai sebuah teori.

BAB V KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan dari hasil penelitian yang telah dilakukan secara keseluruhan dan juga diberikan saran-saran untuk pengembangan selanjutnya.

BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 *Penelitian Yang Relevan*

Menguraikan tentang penelitian-penelitian sebelumnya yang memiliki topik serupa dan perbedaan antara penelitian sebelumnya dengan penelitian yang akan dibuat.

Tabel 1. Penelitian Yang Relevan

No	Penulis Tahun	Topik Penelitian	Metode	Hasil	Keterangan
1	Mongi, Lourent S. Lumenta, Arie S.M. Sambul, Alwin M. / 2018	Rancang Bangun *Game* Adventure of Unsrat Menggunakan *Game* Engine Unity	XP (Extreme Programming)	Hasil pada penelitian ini adalah *game* yang berkonsep simulasi dengan menyampaikan informasi letak lokasi gedung-gedung yang ada di Universitas dengan konsep *game* dan menggunakan objek 3D.	Perbedaan pada penelitian ini adalah *game* yang berkonsep *RPG* atau memainkan peran yang sesuai dengan karakter yang ada pada *game* dan menjelajah dunia yang ada pada *game* dan *game* ini menggunakan objek 2D.
2	Hermawan, Hendi Setiyani, Hari. / 2019	Implementasi Algoritma A-Star Pada Permainan Komputer Roguelike Berbasis Unity.	Prototype	Hasil pada penelitian ini adalah membuktikan pengimplementasian algoritma A-star pada *Artificial Intelligence* (AI) berhasil.	Penelitian ini menggunakan Finite State Machine Behaviour yang di implementasikan pada AI.

No	Penulis Tahun	Topik Penelitian	Metode	Hasil	Keterangan
3	Kurniati, Agnes Nadia Tanzil, Fidelson Purnomo, Fredy. / 2015	*Game* Development "Tales of Mamochi" with Role Playing *Game* Concept Based on Android.	Scrum	Hasilnya adalah sebuah *game* yang diberi nama "Tales of Mamochi" dengan grafis 2D dan memiliki beberapa fitur, seperti pemilihan karakter, buku koleksi, pertempuran (PVE), kerajinan, mamochi evolusi, pertanian, memasak, pencarian.	Perbedaan terletak pada fitur dan platform yaitu pada penelitian ini memiliki fitur level yang terbentuk secara acak dan platformnya adalah PC/Komputer.
4	Singkoh, Robert Theophani Lumenta, Arie S M Tulenan, Virginia. / 2016	Perancangan *Game* FPS (First Person Shooter) Police Personal Training	Waterfall	Hasil pada penelitian ini adalah *game* berhasil dibuat dengan perspective *First-Person* dan menggunakan objek 3D.	Pada penelitian ini menggunakan perspective *Top-Down* dan menggunakan objek 2D.
5	Irwandi, Praja Erlansari, Aan Effendi, Rusdi. / 2016	Perancangan *Game* First Person Shooter (FPS) “Boar Hunter” Berbasis Virtual Reality	Waterfall	Menghasilkan *game* ber*genre* FPS, untuk Smartphone android yang memiliki *Gyroscope* Sensor guna mendukung fitur Virtual Reality (VR)	Pada penelitian ini ber*genre roguelike* dan ditujukan untuk platform komputer dan tidak mendukung Virtual Reality (VR)

2.2 Kajian Teori

2.2.1 *Pengertian Game*

Menurut (Ridoi, 2018), definisi *game* menurut para ahli adalah sebagai berikut:

a. JOHN C BECK & MITCHELL WADE

Game adalah penarik perhatian yang telah terbukti. *Game* adalah lingkungan pelatihan yang baik bagi dunia nyata dalam organisasi yang menuntut pemecahan masalah secara kolaborasi.

b. IVAN C. SIBERO

Game merupakan aplikasi yang paling banyak digunakan dan dinikmati para pengguna media elektronik saat ini.

Kesimpulan dari definisi di atas bahwa *game* merupakan suatu alat yang sangat berpengaruh bagi kehidupan dan perkembangan teknologi yang ada. *Game* merupakan sebagian yang tidak terpisahkan bagi kehidupan seorang anak khususnya, merupakan hal yang dapat menyebabkan kecanduan kepada orang yang memainkannya, serta dapat menjadi alat refreshing penghilang penat dari segala rutinitas keseharian yang telah kita lalui.

2.2.2 *Jenis-jenis Genre Game*

Menurut (Vince, 2018) terdapat beberapa *genre game* yaitu sebagai berikut:

a. Action Games

Jenis permainan aksi adalah permainan dimana pemain yang mengendalikan dan menjadi pusat aksi, yang terutama terdiri dari tantangan fisik yang harus diatasi oleh pemain. *Subgenre* dari *Action Games* adalah *Platformer, Shooter, Fighting, Beat-em up, Stealth, Survival, Rhythm.*

b. Action-Adventure Games

Genre yang menggabungkan dua mekanisme *game*, *quest* atau rintangan sepanjang *game* yang harus ditaklukkan dengan menggunakan alat atau *item* yang dikumpulkan, serta elemen aksi pada item digunakan. *Subgenre* dari *Action-Adventure Games* adalah *Survival Horror, Metroidvania.*

c. Adventure Games

Dalam Adventure *Games*, pemain biasanya berinteraksi dengan lingkungan mereka dan karakter lain untuk menyelesaikan teka-teki dengan petunjuk untuk memajukan cerita atau *gameplay*. *Subgenrenya* yaitu *Text adventures, Graphic Adventures, Visual Novels, Interactive movie, Real-time 3D*

d. Role-Playing Games (RPG)

Dalam *game RPG* kita akan memainkan seorang tokoh yang menjadi pusat dalam *game* tersebut, dan kita di tuntut untuk mengembangkan karakter tersebut supaya menjadi lebih kuat atau memperkuat senjata kita agar kita bisa menyelesaikan *game* tersebut.

Subgenre dari *RPG* adalah *Action RPG, MMORPG, Roguelikes, Tactical RPG, Sandbox RPG, First-person party-based RPG.*

e. Simulation Games

Semuanya dirancang untuk meniru kenyataan nyata atau fiksi, untuk mensimulasikan situasi atau peristiwa nyata. *Subgenrenya* adalah *Construction and management simulation, Life simulation, Vehicle simulation.*

f. Strategy Games

Permainan ini mengharuskan pemain untuk menggunakan strategi dan taktik yang dikembangkan dengan cermat untuk mengatasi tantangan. *Subgenre* untuk *genre* ini adalah *4X (eXplore, eXpand, eXploit, dan eXterminate), Artillery, Real-time strategy (RTS), Real-time tactics (RTT), Multiplayer online battle arena (MOBA), Tower defense, Turn-based strategy (TBS), Turn-based tactics (TBT), Wargame, Grand strategy wargame.*

g. Sports Games

Game olahraga mensimulasikan olahraga seperti golf, sepak bola, bola basket, baseball, sepak bola, ski, dan bahkan olahraga pub seperti panah dan kolam renang. *Subgenre sports games* adalah *Racing, Team sports, Competitive, Sports-based fighting.*

h. Puzzle Games

Teka-teki atau *puzzle games* biasanya terjadi pada satu layar atau lapangan bermain dan mengharuskan pemain untuk memecahkan masalah. *Subgenrenya* adalah *Logic game, Trivia game*

i. Idle Games

Terakhir adalah *idle games*, sebuah *genre game* yang terbilang baru. *Subgenrenya* adalah *Idle gaming, Casual game, Party game, Programming game, Board game/card game, Massive multiplayer online (MMO), Advergame, Art game, Educational game, Exergame.*

2.2.3 *Roguelikes*

Roguelike *game* adalah *game RPG* pemain tunggal, berbasis giliran, tingkat kesulitan yang tinggi, menampilkan lingkungan acak dan kematian permanen. Pemain mengambil peran seorang *hero*, yang memasuki lingkungan permainan (seringkali *dungeon*) dengan tujuan yang sangat sulit. Mencapai tujuan membutuhkan banyak keterampilan, pengalaman *game* dan mungkin sedikit keberuntungan (Cerny & Dechterenko, 2015).

Menurut (King, 2015), istilah *roguelike* secara sempit mengacu pada permainan yang ada seperti Rogue, sebuah *crawler dungeon* Unix tahun 1980 yang menjadi arketipe dari *genre* yang dinamai. *Roguelike* digunakan untuk menunjukkan bahwa sebuah permainan menggabungkan fitur mekanik utama yang dimiliki Rogue. Estetika dari Rogue (seorang petualang yang mencoba mencapai dasar penjara

bawah tanah) secara luas ikut disalin. Sejak Rogue dan turunan langsungnya adalah *RPG hack-and-slash*, elemen desain unik dari permainan cenderung digabungkan dengan aspek *genre RPG* yang seharusnya mereka sukai.

Dengan banyaknya game yang mirip dengan Rogue, Berlin Interpretation, yang mana diciptakan pada International Roguelike Development Conference (Roguebasin, 2008) kemudian memberikan dua faktor yaitu *high value* dan *low value* untuk menentukan *game roguelike*.

Namun, seiring waktu beberapa *game roguelike* kemudian mulai bergeser dari beberapa karakteristik tersebut, tapi masih menggunakan dua di antaranya. Dua karakteristik tersebut adalah *Random environment generation* dan *Permadeath* yang kemudian menjadi ciri khas dari *game roguelike*.

2.2.4 *Elemen Games*

Elemen-elemen penting dari sebuah game adalah *play, pretending, a goal, the rules* (Adams, 2013). Adapun definisi dari elemen-elemen tersebut adalah:

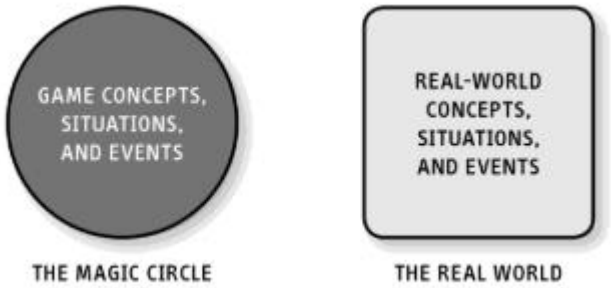
1. *Play*

   Bermain adalah salah satu bentuk hiburan yang membutuhkan keikutsertaan (komunikasi dua arah), dimana buku, film, dan teater adalah bentuk hiburan yang berupa pertunjukkan (komunikasi dua

arah). Ketika membaca buku, penulis yang menghibur pembaca, tetapi ketika bermain, pemainlah yung menghibur dirinya sendiri. Sebuah buku tidak dapat berubah meskipun sudah sering dibaca, namun ketika bermain, pemain dapat membuat keputusan yang mempengaruhi kejadian dalam permainan tersebut.

2. *Pretending*

Berpura-pura adalah kegiatan untuk menciptakan sebuah realitas khayalan di dalam pikiran. Nama lain dari realitas yang diciptakan dari berpura-pura adalah magic circle. Istilah ini pertama kali digunakan oleh sejarahwan Belanda bernama Johan Huizinga dalam bukunya berjudul ”Homo Ludens” (Huizinga, 1971) yang berhubungan dengan dunia imajinasi dalam sebuah drama atau novel fiksi. Dalam *game*, *magic circle* mengacu pada batas yang memisahkan ide dan aktifitas yang berarti yang berada dalam *game* dari ide dan aktifitas yang berarti dalam dunia nyata. Dengan kata lain batas antara dunia nyata dan khayalan (*make believe*).



Gambar 1. Konsep The Magic Circle (Adams, 2013)

3. *A Goal*

Sebuah *game* harus memiliki satu atau lebih tujuan. Tujuan dari *game* tersebut oleh peraturan dan bentuk *game* itu sendiri karena pembuat *game* dapat menentukannya sesuai dengan keinginanya. Dalam mencapai tujuan dalam *game*, harus ada tantangan. Meskipun kesulitan dari tantangan itu tergantung dari persepsi masing-masing pemain.

4. *The Rules*

Peraturan adalan definisi dan instruksi dimana pemain harus mengiktuinya selama *game* berlangsung. Peraturan memiliki beberapa fungsi yaitu membuat objek *game* dan arti dari setiap aktifitas dan kejadian yang berbeda yang berlangsung di dalam *circle magic*. Peraturan juga membuat pemain mengetahui aktifitas apa yang diperbolehkan dan juga mengevaluasi setiap tindakan yang akan membuat pemain mencapai tujuan dari *game* tersebut.

2.2.5 *Kecerdasan Buatan (Artificial Intellegence)*

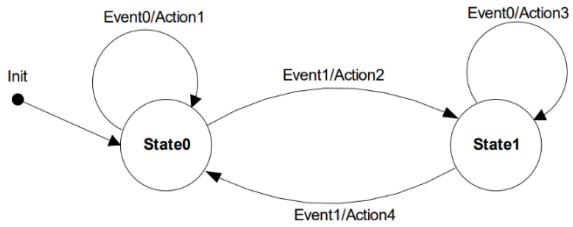
Kecerdasan Buatan (*Artificial Intelligence*) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia. Manusia cerdas (pandai) dalam menyelesaikan permasalahan karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki

tentu akan lebih mampu menyelesaikan permasalahan. Tapi bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran,mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang dimiliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Demikian juga dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik (Dahria, 2008).

2.2.6 *Finite State Machines (FSM)*

Metode yang digunakan adalah *Finite State Machines (FSM)* yaitu 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. Sistem dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau *event* tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri (misal interupsi timer). Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relative kompleks (Setiawan, 2006). Dalam bahasa pemrograman

prosedural seperti bahasa C, FSM ini umumnya direalisasikan dengan menggunakan statemen kontrol *switch case* atau/dan *if..then*. Dengan menggunakan statemen-statemen kontrol ini, aliran program secara praktis akan mudah dipahami dan dilacak jika terjadi kesalahan logika (Setiawan, 2006). Contoh diagram state sederhana ditunjukan pada gambar



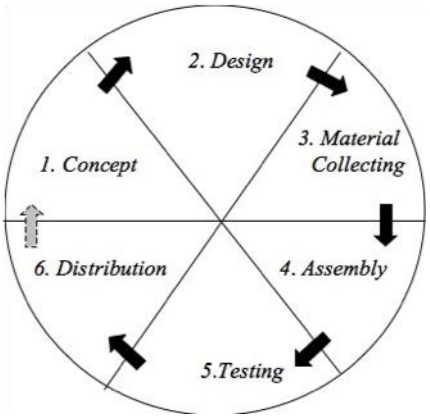
Gambar 2. Diagram State Sederhana (Setiawan, 2006)

Diagram tersebut memperlihatkan *FSM* dengan dua buah *state* dan dua buah input serta empat buah aksi output yang berbeda : seperti terlihat pada gambar, ketika sistem mulai dihidupkan, sistem akan bertransisi menuju *state0*, pada keadaan ini sistem akan menghasilkan *Action1* jika terjadi masukan *Event0*, sedangkan jika terjadi *Event1* maka *Action2* akan dieksekusi kemudian sistem selanjutnya bertransisi ke keadaan *State1* dan seterusnya.

2.2.7 *Multimedia Development Life Cycle (MDLC)*

Menurut Sutopo, dalam (Setiawan, et al., 2017), Pengembangan metode multimedia ini dilakukan berdasarkan enam tahap, yaitu *concept* (pengonsepan), *design* (perancangan), *material collecting* (pengumpulan

bahan), *assembly* (pembuatan), *testing* (pengujian), dan *distribution* (pendistribusian). Tahapan ini dapat dilihat pada gambar



Gambar 3. MDLC (Setiawan, et al., 2017)

Adapun untuk penjelasan tahapan-tahapan diatas sebagai berikut:

1) *Concept* (Konsep)

Tahap *concept* (konsep) adalah tahap untuk menentukan tujuan dan siapa pengguna program (identifikasi audience). Selain itu menentukan macam aplikasi (presentasi, interaktif, dll) dan tujuan aplikasi (hiburan, pelatihan, pembelajaran, dll).

2) *Design* (Desain)

Design (perancangan) adalah tahap membuat spesifikasi mengenai arsitektur program, gaya, tampilan dan kebutuhan material/bahan untuk program.

3) *Material Collecting* (Pengumpulan Materi)

Material Collecting adalah tahap dimana pengumpulan bahan yang sesuai dengan kebutuhan dilakukan. Tahap ini dapat

dikerjakan paralel dengan tahap *assembly*. Pada beberap kasus, tahap *Material Collecting* dan tahap *Assembly* akan dikerjakan secara linear tidak paralel.

4) *Assembly* (Pembuatan)

Tahap *assembly* (pembuatan) adalah tahap dimana semua objek atau bahan multimedia dibuat. Pembuatan aplikasi didasarkan pada tahap *design*.

5) *Testing* (Pengujian)

Dilakukan setelah selesai tahap pembuatan (*assembly*) dengan menjalankan aplikasi/ program dan dilihat apakah ada kesalahan atau tidak. Tahap ini disebut juga sebagai tahap pengujian *alpha* (*alpha test*) dimana pengujian dilakukan oleh pembuat atau lingkungan pembuatnya sendiri.

*6) Distribution (*Pendistribusian)

Tahapan dimana aplikasi disimpan dalam suatu media penyimpanan. Pada tahap ini jika media penyimpanan tidak cukup untuk menampung aplikasinya, maka dilakukan kompresi terhadap aplikasi tersebut.

2.2.8 *C#*

C# adalah salah satu dari banyak bahasa yang bisa dipakai untuk pemrograman .NET. Kelebihan utama bahasa ini adalah sintaksnya yang mirip C, namun lebih mudah dan lebih bersih (Rachmatullah, 2012).

Microsoft membuat C# seiring dengan pembuatan Framework .NET. Chief Architect dalam pembuatan C# adalah Anders Hejlsberg yang sebelumnya berperan dalam pembuatan Borland Delphi dan Turbo Pascal. C# menjanjikan produktifitas dan kemudahan yang ada di Visual Basic dengan kemampuan dan fleksibilitas yang ada di C/C++ (Rachmatullah, 2012).

Apa sebenarnya tujuan bahasa C#? Menurut spesifikasi bahasanya, "C# (pronounced "C Sharp") is a simple, modern, object oriented, and type-safe programming language. It will immediately be familiar to C and C++ programmers. C# combines the high productivity of Rapid Application Development (RAD) languages and the raw power of C++.". Untuk mencapai produktifitas tinggi ini konsep-konsep sulit C++ disederhanakan dan fitur-fitur baru ditambahkan. Hal ini mungkin terasa mirip dengan Java, karena itulah C# bisa dianggap sebagai sepupu Java (Rachmatullah, 2012).

2.2.9 *UML (Unified Modeling Language*)

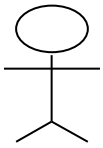
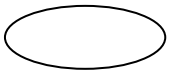
UML (*Unified Modeling Language**)*** adalah sebuah bahasa untuk menentukan, visualisasi, kontruksi dan mendokumentasikan *artifac* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak, dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak seperti pada permodelan bisnis dan sistem non perangkat lunak lainya.

Menurut Grace dalam penelitian (Hendini, 2016), *Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang di pergunakan untuk mendokumentasikan, menspesifikasikan dan membanngun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

1) Use Case Diagram

Use case diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case* Diagram.

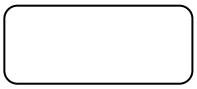
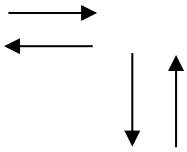
Tabel 2. Use Case Diagram (Hendini, 2016)

No	Simbol	Nama	Keterangan
1		*Actor*	*Actor* atau Aktor adalah *Abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Perlu dicatat bahwa aktor berinteraksi dengan *Use Case*, tetapi tidak memiliki kontrol terhadap *use case*
2		*Generalization*	Mengindikasikan bila actor berinteraksi secara pasif dengan sistem.
3		*Association*	Mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukanya mengindikasi data.
4		*Use Case*	*Use Case* menggambarkan fungsionalitas yang disediakansistem sebagai unit-unit yang bertukar pesan antar unit dengan aktir, yang dinyatakan dengan menggunakan kata kerja.

2) *Activity* Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity* Diagram.

Tabel 3. Activity Diagram (Hendini, 2016)

No	Simbol	Nama	Keterangan
1		Initial Node	Simbol ini menunjukkan bagaimana objek dibentuk atau diawali.
2		*Activity Final Mode*	Menunjukkan bagaimana objek dibentuk atau diakhiri.
3		*Activity*	Menggambarkan suatu proses/kegiatan bisnis
4		*Line Connector*	Simbol ini digunakan untuk menghubungkan satu simbol dengan simbol lainya.

BAB III
METODE PENELITIAN

3.1 Teknik Pengumpulan Data

Teknik pengumpulan data yang penulis gunakan dalam penelitian ini adalah teknik pengumpulan data studi pustaka dan kuisioner. Pengumpulan data studi pustaka dilakukan dengan cara mengumpulkan buku-buku, ada pula mengumpulkan dari teknologi yang sudah ada saat ini, literatur, jurnal, internet dan bacaan-bacaan sebagai sumber referensi yang berhubungan dengan aplikasi yang akan dibuat, serta teori yang berhubungan dengan perancangan *game roguelike* dan kuisioner online yang penulis gunakan untuk mengukur tingkat ketertarikan pengguna pada *game Tales of Seleine* ini.

3.2 Analisis

3.2.1 *Analisis Proses*

Multimedia Development Life Cycle (MDLC)

1. *Concept* (Pengonsepan)

   Tujuan dari pembuatan aplikasi *game* ini adalah sebagai media hiburan didalam ruang lingkup dunia *game* dengan target pengguna berumur dari 13 – 25.

2. *Design* (Perancangan)

   Tahap ini menggunakan perancangan perangkat lunak untuk menggambarkan rencana tampilan aplikasi *game* ini nantinya.

3. *Material Collecting* (Pengumpulan Bahan)

   Bahan – bahan yang digunakan pada *game* ini diperoleh dengan membuat sendiri dan juga pemesanan kepada pihak lain sesuai rancangan yang telah dibuat.

4. *Assembly* (Pembuatan)

   Pada tahap ini semua bahan yang sudah dikumpulkan akan dibuat sesuai dengan perancangan. Pembuatan *sprite character* dari *game* ini penulis buat dengan menggunakan *Adobe Photoshop* dan untuk musik penulis buat menggunakan *Fl Studio* sedangkan untuk video penulis buat menggunakan *Adobe After Effects* dan untuk pemrograman penulis lakukan pada *game engine* Unity.

5. *Testing* (Pengujian)

   Pengujian yang pertama dilakukan adalah *Alpha Test* dimana pengujian ini dilakukan oleh penulis sendiri atau orang – orang dilingkungan penulis. *Alpha Testing* ini terdiri dari pengujian pada fitur aplikasi, dan pengujian terhadap fitur game mechanic dan juga tampilan, apakah sudah berjalan sesuai yang diharapkan atau tidak.

   Setelah *Alpha Testing* dilakukan dan mendapat hasil yang sesuai dengan skenario pengujian, maka tahap *Beta Testing* siap dilakukan, pada tahap ini, digunakan kuesioner online untuk mengukur tingkat ketertarikan pengguna terhadap *game Tales of Seleine* dan fitur - fitur yang telah di uji di *Alpha Testing.*

6. *Distribution* (Pendistribusian)

   Aplikasi *game* ini didistribusikan melalui *itch.io* dengan tujuan *platform Windows* dan *Mac.*

3.2.2 *Analisis Kelemahan Sistem*

Berikut adalah hal-hal yang dapat di analisis menggunakan SWOT :

1) *Strength* (kekuatan) :

   Strength adalah kekuatan yang dijadikan modal dasar dalam membangun *game genre roguelike*. Adapun kekuatan yang dimaksud yakni berbicara tentang kualitas, terlebih mekanisme game yang cukup mudah di mainkan, music yang dibuat dengan nuansa *EDM, dan post-processing game* yang enak dipandang mata sehingga cukup mudah diterima masyarakat.

2) *Weakness* (kelemahan) :

   Dalam kelemahan sistem pada *game* yang dibuat penulis adalah *game* ini hanya berjalan pada sistem komputer dengan minimum requirement processor Intel core 2 Duo E5200, RAM 2 GB dan Graphics dengan memory 1 GB.

3) *Opportunity* (peluang) :

   Kesempatan atau peluang yang dapat diperolah dalam rancang bangun *game bergenre roguelike adventure* “Tales of Seleine” yaitu menarik minat orang-orang yang suka memainkan *game* dengan *genre roguelike* untuk memainkan game Tales of

Seleine ini dan juga bahkan orang-orang yang memiliki motivasi *gamer destruction, completion, fantasy, story*, dan *discovery*.

4) *Threat* (ancaman) :

Ancaman yang memungkinkan dalam *game* ini telah banyak *game bergenre roguelike* yang dikembangkan oleh *developer* besar dan lebih menarik baik dari segi tampilan, fitur, *gameplay*, dan hal-hal lainnya yang dapat lebih menarik pemain (*gamer*) untuk memainkannya. Hal ini memungkinkan terjadi persaingan yang ketat dalam pembuatan game bagi para pembuat *game.*

3.2.3 *Analisis Kebutuhan*

Kebutuhan sistem haruslah sesuai dengan kondisi dan kemampuan pengguna, maka dari itu penulis yang juga adalah sebagai pembuat atau membangun program ikut serta melibatkan pengguna dalam mencari dan menganalisis kebutuhan-kebutuhan sistem yang menunjang dalam proses perancangan serta membangun aplikasi *game roguelike adventure* "Tales of Seleine". Adapun kebutuhan sistem yang diperlukan itu sebagai berikut:

1) Kebutuhan Perangkat Keras

Dalam kebutuhan perangkat keras (*Hardware)* yang penulis gunakan dalam pembuatan aplikasi sebagai berikut :

Tabel 4. Kebutuhan Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Type	MacBook Pro (13-inch, 2017)
2	Processor	2,3 GHz Dual-Core Intel Core i5
3	Memory	8 GB 2133 MHz LPDDR3
4	Disk	128 GB SSD
5	Graphics	Intel Iris Plus Graphics 640 1536 MB

2) Kebutuhan Perangkat Lunak

Dalam kebutuhan Perangkat Lunak (*software)* dalam aplikasi ini yang digunakan seperti berikut :

Tabel 5. Kebutuhan Perangkat Lunak

No	Perangkat Lunak	Spesifikasi
1	Unity	Sebagai perangkat lunak dalam pembuatan game
2	Adobe Photoshop	Sebagai perangkat lunak untuk pembuatan sprite dan animation
3	FL Studio	Sebagai perangkat lunak untuk pembuatan music dan sound fx

4	After Effects	Sebagai perangkat lunak untuk pembuatan cutscene
5	Mac OS Catalina	Digunakan sebagai OS dalam menjalankan aplikasi perangkat lunak

3.2.4 *Analisa Kelayakan Sistem*

1) Kelayakan Teknologi

Kelayakan teknologi yang diberikan aplikasi ini adalah sebagai perangkat lunak yang dapat memberikan pilihan permainan *roguelike* untuk pencinta game jenis *genre* ini.

2) Kelayakan Hukum

Untuk kelayakan hukum aplikasi ini menggunakan perangkat lunak yang gratis dan berbayar yang dimana penulis memiliki license untuk aplikasi berbayar yang penulis gunakan sehingga tidak ada masalah dengan pelanggaran hukum atau *software* bajakan. Perangkat lunak yang dimaksud dalam pembuatan aplikasi ini seperti Unity, Adobe Photoshop, Adobe After Effects dan FL Studio.

3) **Kelayakan Operasional**

Mengenai kelayakan operasional aplikasi yang dibangun, dirancang sesederhana dan semenarik mungkin agar pengguna dapat menikmati aplikasi yang penulis buat.

3.3 Desain Sistem

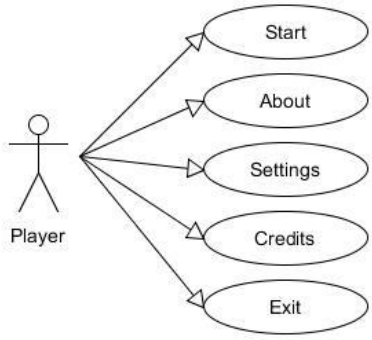
3.3.1 *Storyline*

Bercerita tentang *Ancient Mage* bernama Seleine, seorang penyihir ambisius yang meninggalkan rumahnya untuk pergi ke *Thelkan Cave* untuk menemukan Kwolok's Ring, yang dimana di dalam perjalanannya mencari *Kwolok's Ring* tersebut sudah pasti banyak masalah menunggu, musuh-musuh yang menjaga *Kwolok's Throne Room*, Lantai *Dungeon* yang selalu berubah-ubah dan banyaknya jebakan didalam *Dungeon*. *Player* akan melewati 4 *stage game* dengan setiap *stage* memiliki tingkat kesulitan yang berbeda, seperti jumlah musuh yang ada akan terus bertambah seiring meningkatnya *stage game.* Game akan selesai jika player berhasil mengalahkan boss stage akhir dan mengambil cincin yang terjatuh setelah boss menghilang.

3.3.2 *Desain Proses*

1) Use Case Diagram

Use Case Diagram merupakan gambaran dari *User* yang menggunakan sistem dan perilaku *User* Terhadap sistem.



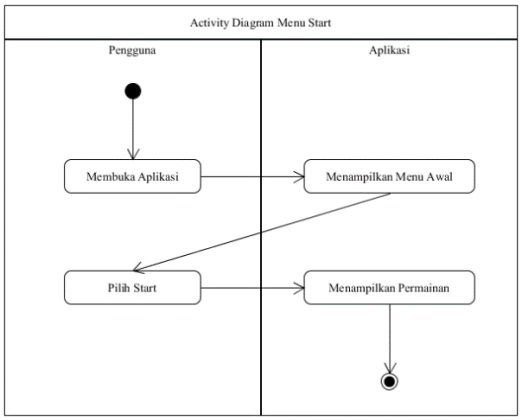
Gambar 4. Use Case Diagram

Dijelaskan bahwa dalam *Use Case* Diagram aplikasi ini terdapat tampilan menu-menu yang ada pada aplikasi *Game* Tales of Seleine.

2) Activity Diagram

Activity Diagram merupakan gambaran alur proses atau cara kerja sistem. Pada diagram ini digambarkan aktivitas-aktivitas apa saja yang dikerjakan oleh sebuah sistem.

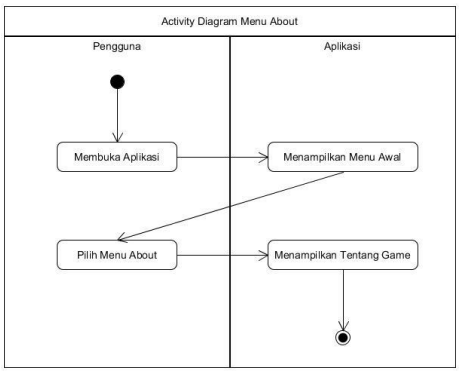
1) Diagram Activity Start



Gambar 5. *Diagram Activity Start*

Pada gambar diatas dijelaskan bahwa saat pengguna membuka aplikasi, kemudian akan ada tampilan menu awal. Pada menu awal aplikasi pengguna memilih tombol "*Start*" yang berfungsi untuk menampilkan permainan yang ada di dalam *button* tersebut.

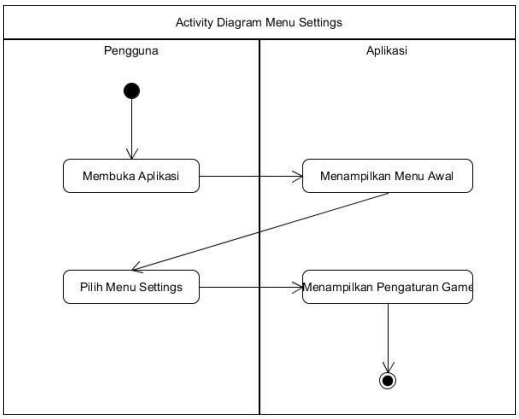
2) Diagram Activity About



Gambar 6. *Diagram Activity About*

Pada gambar diatas dijelaskan bahwa pada saat pengguna memilih menu *About* maka *scene* akan berpindah ke *scene* about.

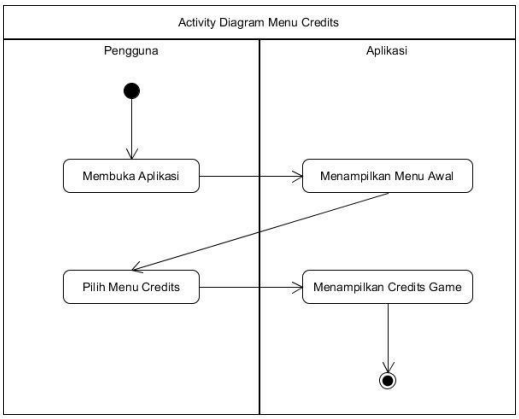
3) Diagram Activity Settings



Gambar 7. *Diagram Activity Settings*

Pada gambar diatas dijelaskan bahwa aplikasi menampilkan informasi kepada pengguna tentang pengaturan pada *game* yaitu *volume*, *graphics*, mode *fullscreen.*

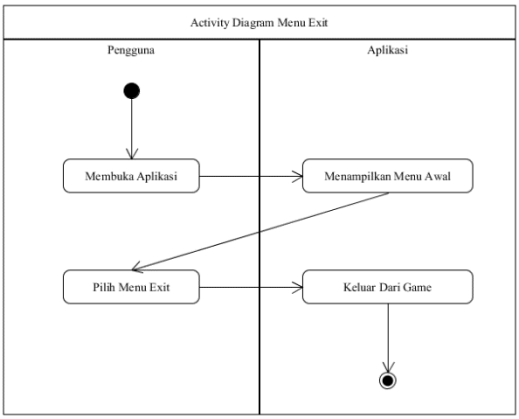
4) Diagram Activity Credits



Gambar 8. *Diagram Activity Credits*

Pada gambar diatas dijelaskan bahwa pada saat pengguna memilih menu *Credits* maka *scene* akan berpindah ke *scene credits*.

5) *Diagram Activity Exit*

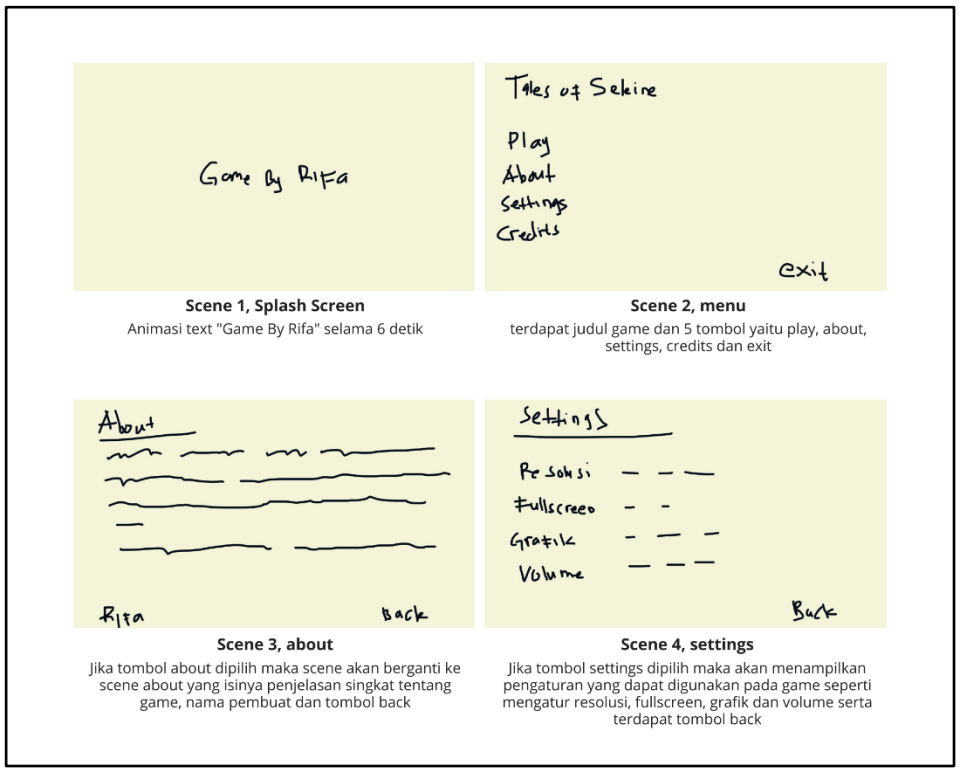


Gambar 9. *Diagram Activity Exit*

Pada gambar diatas dijelaskan bahwa aplikasi menampilkan informasi kepada pengguna jika tombol "*Exit*" dipilih makan akan keluar dari *game*.

3) *Storyboard*

Storyboard pada *Game* sedikit berbeda dari *storyboard* animasi, dimana pada *storyboard game* terdapat *goal* / mission yang akan dibuat pada *game* itu sendiri. Kemudian *storyboard* pada *game* menjelakaskan tentang alur permaianan itu sendiri seperti apa tergantung dari jenis *game*. Hasil desain storyboard game Tales of Seleine merupakan sketsa gambar yang menceritakan alur dari game ini sendiri yang akan ditunjukan pada gambar dibawah ini.



Scene 1, Splash Screen

Animasi text "Game By Rifa" selama 6 detik

Scene 2, menu

terdapat judul game dan 5 tombol yaitu play, about, settings, credits dan exit

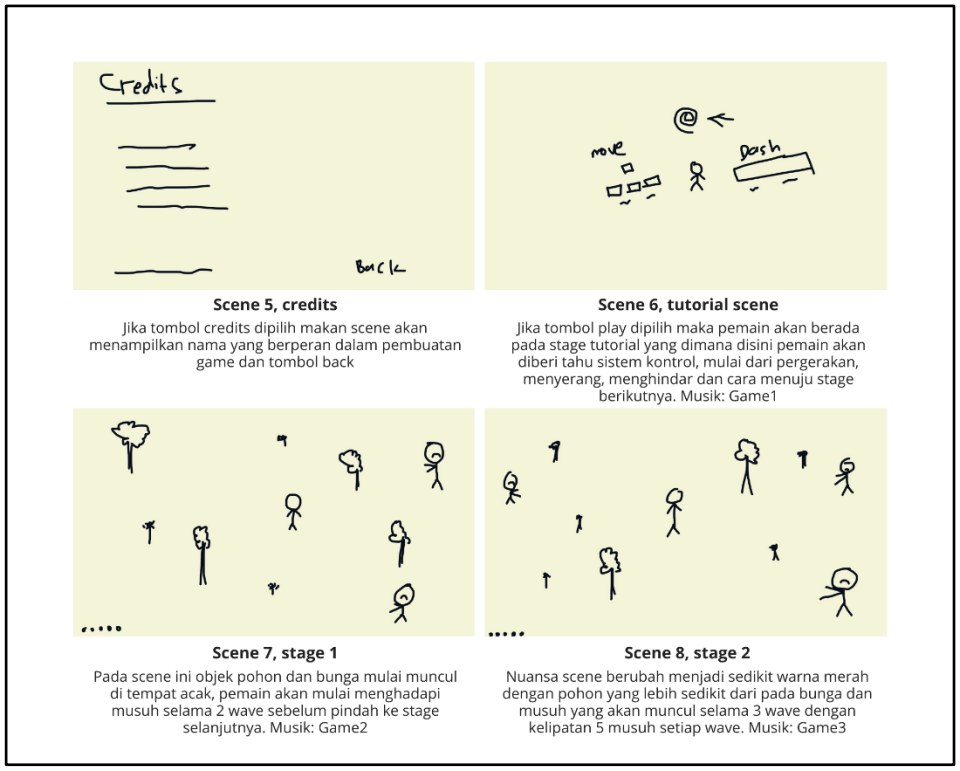
Scene 3, about

Jika tombol about dipilih maka scene akan berganti ke scene about yang isinya penjelasan singkat tentang game, nama pembuat dan tombol back

Scene 4, settings

Jika tombol settings dipilih maka akan menampilkan pengaturan yang dapat digunakan pada game seperti mengatur resolusi, fullscreen, grafik dan volume serta terdapat tombol back

Gambar 10. Storyboard halaman 1



Scene 5, credits

Jika tombol credits dipilih makan scene akan menampilkan nama yang berperan dalam pembuatan game dan tombol back

Scene 6, tutorial scene

Jika tombol play dipilih maka pemain akan berada pada stage tutorial yang dimana disini pemain akan diberi tahu sistem kontrol, mulai dari pergerakan, menyerang, menghindar dan cara menuju stage berikutnya. Musik: Game1

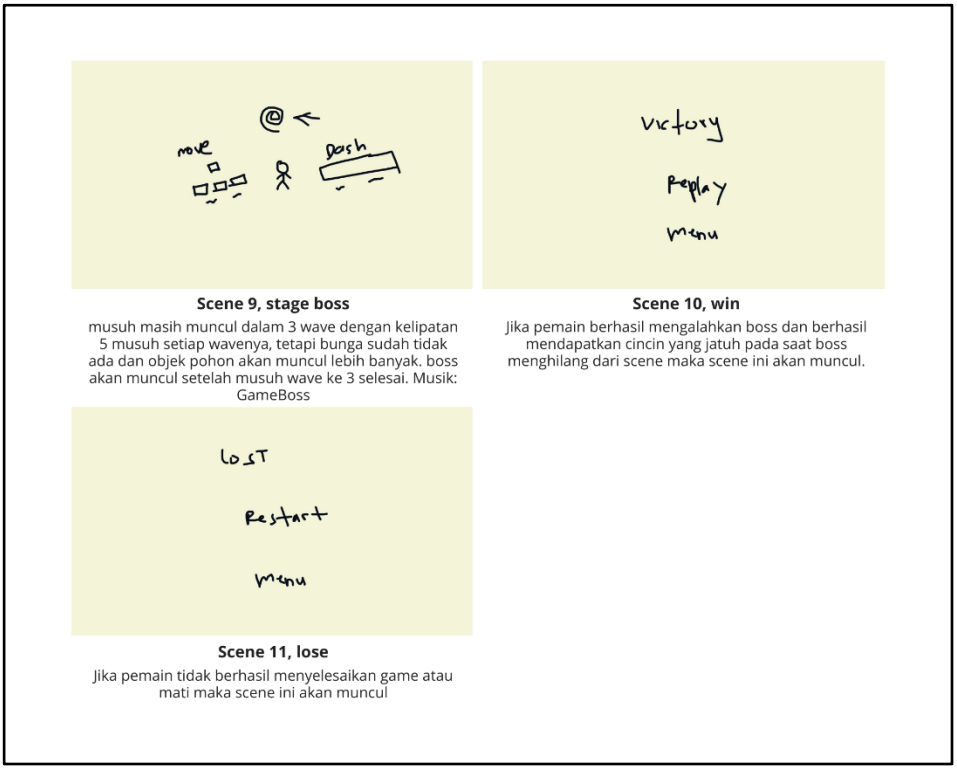
Scene 7, stage 1

Pada scene ini objek pohon dan bunga mulai muncul di tempat acak, pemain akan mulai menghadapi musuh selama 2 wave sebelum pindah ke stage selanjutnya. Musik: Game2

Scene 8, stage 2

Nuansa scene berubah menjadi sedikit warna merah dengan pohon yang lebih sedikit dari pada bunga dan musuh yang akan muncul selama 3 wave dengan kelipatan 5 musuh setiap wave. Musik: Game3

Gambar 11. Storyboard halaman 2



Gambar 12. Storyboard halaman 3

3.3.3 *Desain Perangkat Lunak*

1) Splash Screen

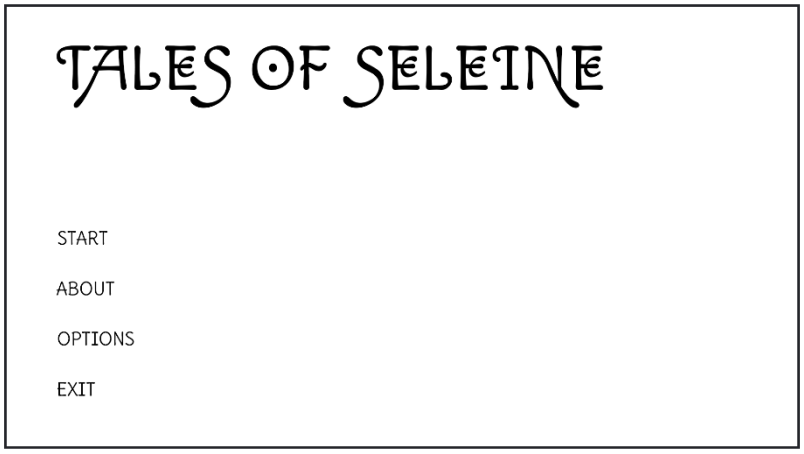
Tampilan awal dari game adalah *splash screen* sambutan dengan *nickname* penulis yang bertransisi ke scene selanjutnya.



Gambar 13. Tampilan Splash Screen

2) Menu Utama

Tampilan menu utama terdapat judul dari *game*, dan tombol "*Start*" untuk memulai permainan, "*About*" untuk menampilkan sekilas tentang game Tales of Seleine, "*Options*" yang berfungsi menampilkan pengaturan *game* dan "*Exit*" berfungsi untuk keluar dari *game*.



Gambar 14. Tampilan Menu Utama

3) Gameplay

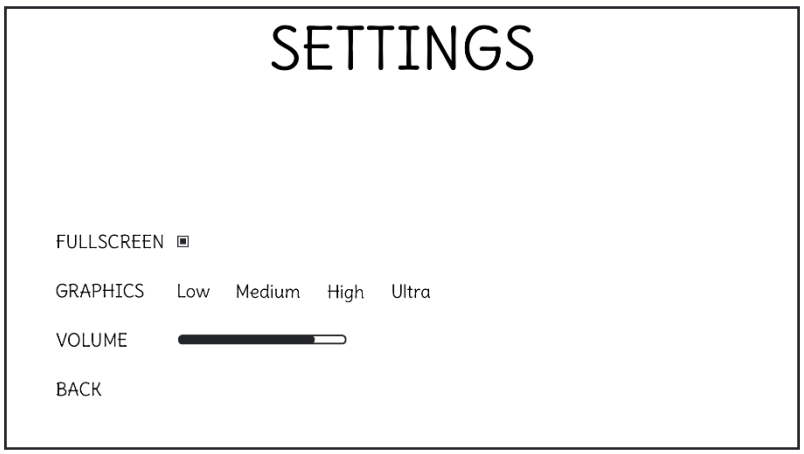
Menu ini berisi tentang gambaran gameplay permainan kedepannya, dimana sebuah area gameplay yang menjadi arena dari pengguna untuk di explore dan memiliki Heads up Display (HUD) seperti tampilan sisa health, gold, mini map, senjata yang digunakan, dan menampilkan tata letak dari pintu yang muncul secara acak yang berguna menuju ruang selanjutnya.



Gambar 15. Tampilan Gameplay

4) Pengaturan

Menu ini menampilkan pengaturan *game* seperti mode *fullscreen,* tingkat grafik, besar kecil *volume* dan tombol kembali ke menu awal.



Gambar 16. Tampilan Settings

BAB IV
HASIL DAN PEMBAHASAN

4.1 Hasil

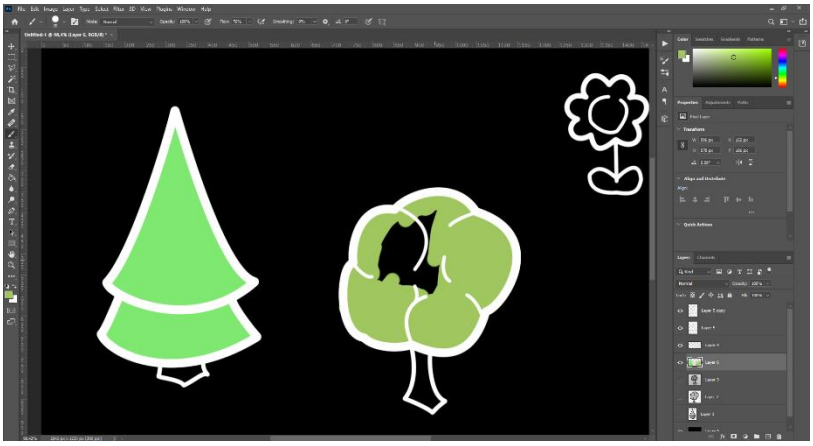
4.1.1 Implementasi Program

Dalam implementasi program *software* yang digunakan penulis untuk membuat *game* Tales of Seleine adalah *Unity* 2020 dengan Bahasa program C#. Untuk perancangan desain dalam *game* penulis menggunakan *Adobe Photoshop* sebagai aplikasi membuat dan mengedit *sprites, object dan background* pada *game* Tales of Seleine. Untuk membuat *splash screen* penulis menggunakan aplikasi *Adobe After Effects*, dan untuk musik serta *sound effects* penulis menggunakan aplikasi FL Studio.

Pada bagian ini, penulis akan memperlihatkan tampilan pada program yang penulis gunakan dalam membuat *game* Tales of Seleine serta pembuatan apa saja yang dikerjakan dan yang digunakan.

a. Pembuatan *sprites, object pick up* dan *background.*

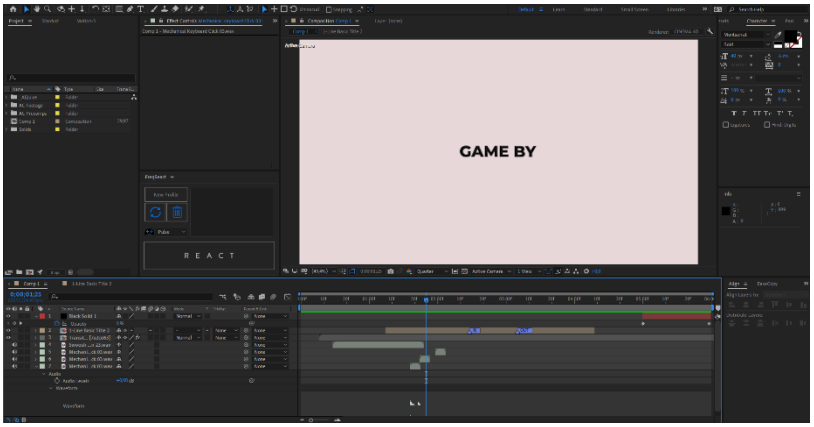
   Pada pembuatan *sprites, object pick up* dan *background* dalam *game* Tales of Seleine ini menggunakan aplikasi *Adobe Photoshop CC. Sprites* adalah suatu kumpulan gambar dengan berbagai bentuk dengan sesuai kebutuhan, bisa berupa karakter atau gambar apapun kemudian disusun sedemikian rupa sehingga bisa menghasilkan suatu gambar animasi. Pada *game* Tales of Seleine ini *sprites* yang digunakan disimpan dengan format *png.



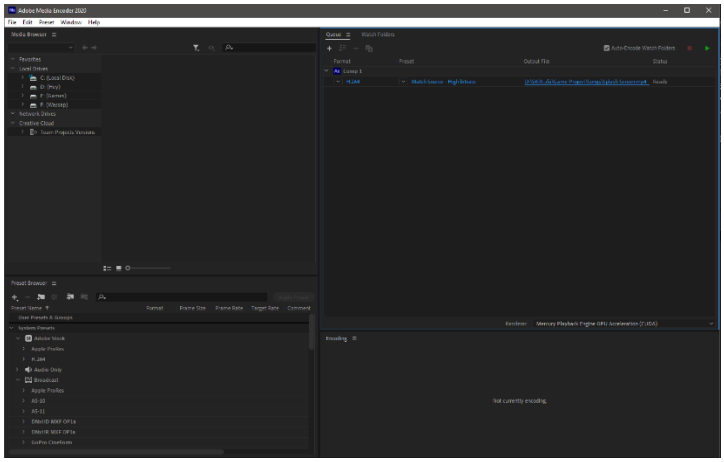
Gambar 17. Proses pembuatan *sprites* pohon pada *Adobe Photoshop CC*

b. Pembuatan *splash screen*

Splash screen adalah layer awal yang dibuka sebelum aplikasi utama dijalankan. Pembuatan *splash screen* sendiri penulis lakukan menggunakan *adobe after effects* sehingga dapat dibuat dengan *motion graphics.* Format dari *splash screen* yang penulis gunakan yaitu *mp4 yang kemudian penulis gunakan didalam *unity*.



Gambar 18. Proses pembuatan *splash screen* di *Adobe After Effects*



Gambar 19. Proses render splash screen

c. Pembuatan musik dan *sound effects*

Musik dan *sound effects* dalam suatu game adalah suatu aspek yang penting perannya dalam membangun suasana game itu sendiri. Dalam game Tales of Seleine penulis menggunakan aplikasi FL Studio dalam proses pembuatan musik dan memproses sound effects yang kemudian nantinya akan di simpan dalam format *wav yang dimana kualitas *wav itu sangat mudah disunting dan memiliki kualitas yang bagus disbanding *mp3.



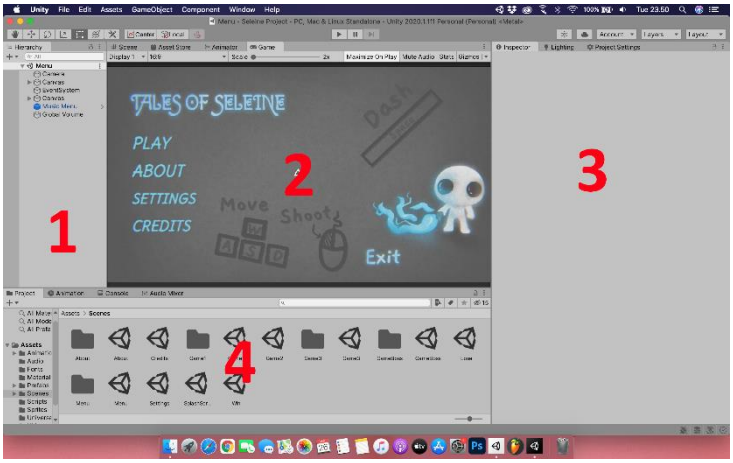
Gambar 20. proses pembuatan musik di FL Studio



Gambar 21. proses render musik

d. *Worksheet Unity 2020*

Unity merupakan salah satu game engine yang digunakan untuk mengembangkan *game multi-platform* yang didesain untuk mudah digunakan. Editor pada *unity* dibuat dengan *user interface* yang sederhana. *Unity* cocok dengan versi 64-bit dan dapat beroperasi pada Mac Os X dan Windows dan dapat menghasilkan game untuk Mac, Windows, Wii, Iphone, Iphone, Ipad, Android dan konsol *game* lainnya.



Gambar 22. *Worksheet Unity*

Keterangan:

1. Panel *Hierarchy*, panel ini berfungsi untuk menampilkan list *game* objek apa saja yang digunakan pada *scene*. Dalam *hierarchy* ini juga dapat membuat game objek 2D, 3D, UI serta Lighting dll.
2. Panel *Game,* panel ini tempat menampilkan hasil dari panel *scene* yang sedang dalam keadaan aktif pada *unity*.
3. Panel *Inspector*, panel ini berfungsi untuk menampilkan property dan komponen yang dimiliki sebuah *game* objek.
4. Panel Project, panel ini berfungsi sebagai tempat menampilkan *assets* projek dari *game* kita. *Assets* yang dimaksud yaitu berupa sebuah gambar (*JPG*, *PNG*, dll), *audio*, tekstur, *prefabs*, animasi dll.

4.1.2 Pengujian Sistem dan Uji Coba

Pengujian merupakan bagian penting dari pengembangan perangkat lunak. Pengujian bertujuan untuk menemukan kesalahan pada sistem dan memastikan bahwa sistem yang dibangun sesuai dengan rencana sebelumnya. Pengujian system yang penulis gunakan yaitu menggunakan *black box testing*. Pengujian sistem ini memfokuskan pada kebutuhan fungsional sistem aplikasi. Berikut ini adalah hasil pengujian *game* Rancang Bangun *Game* Bergenre *Roguelike Adventure* "Tales of Seleine".

a. Pengujian buka aplikasi

Tabel 6. Pengujian buka aplikasi

Pengujian	**Skenario Uji**	**Hasil yang diharapkan**	**Hasil pengujian**
Membuka dan memulai aplikasi	Buka aplikasi Rancang Bangun Game Bergenre Rogulike Adventure "Tales Of Seleine"	Aplikasi terbuka dan menampilkan *splash screen* kemudian masuk ke menu utama	Sesuai

b. Pengujian Halaman Menu Utama

Tabel 7. Pengujian halaman menu utama

Pengujian	**Skenario Uji**	**Hasil yang diharapkan**	**Hasil pengujian**
Button Play	Sentuh *button play* pada menu utama	Aplikasi akan membuka *scene gameplay*	Sesuai

Button About	Sentuh *button about* pada menu utama	Aplikasi akan membuka *scene about*	Sesuai
Button Settings	Sentuh *button settings* pada menu utama	Aplikasi akan membuka *scene settings game*	Sesuai
Button Credits	Sentuh *button credits* pada menu utama	Aplikasi akan membuka *scene credits game*	Sesuai

c. Pengujian Halaman Play

Tabel 8. Pengujian halaman *play*

Pengujian	**Skenario Uji**	**Hasil yang diharapkan**	**Hasil pengujian**
Pergerakan *Player*	Gerakan *player* dengan W,A,S,D	*Player* bergerak ke segala arah	Sesuai
Player Shots	Menekan tombol klik kiri pada mouse	Senjata player menembakan *projectile* kearah kursor mouse	Sesuai
Pickup Items	Ambil *item* yang jatuh dari musuh	Player mendapatkan senjata baru atau darah terisi 1 hati	Sesuai
Dash	Tekan tombol *space* pada keyboard	Kecepatan *player* akan bertambah sesaat	Sesuai
Random Objek	Masuk ke lingkaran ungu dan lihat objek di *stage*	Objek pohon dan bunga akan selalu berubah posisi setiap *game* di *load*	Sesuai

d. Pengujian Halaman About

Tabel 9. Pengujian Halaman About

Pengujian	**Skenario Uji**	**Hasil yang diharapkan**	**Hasil pengujian**
Button Back	Tekan *button back* dengan klik kiri pada mouse	Aplikasi akan Kembali ke halaman menu utama	Sesuai

e. Pengujian Halaman Settings

Tabel 10. Pengujian Halaman Settings

Pengujian	**Skenario Uji**	**Hasil yang diharapkan**	**Hasil pengujian**
Button resolution	Tekan *button size resolution*	Aplikasi akan mengubah ke resolusi yang dipilih	Sesuai
Button fullscreen	Tekan *button Full On* atau *Full Off*	Aplikasi akan berubah ke *mode fullscreen* jika *on* dan ke *mode windowed* jika *off*	Sesuai
Button graphics	Tekan *button high, medium* atau *low*	Aplikasi akan mengubah kualitas *game* ke *setting* yang di pilih	Sesuai
Button volume	Tekan *button* 0, 50 atau 100	Aplikasi akan tingkat keras suara *game* ke angka tersebut	Sesuai
Button Back	Tekan *button back* dengan klik kiri pada mouse	Aplikasi akan Kembali ke halaman menu utama	Sesuai

f. Pengujian Halaman Credits

Tabel 11. Pengujian halaman *credits*

Pengujian	**Skenario Uji**	**Hasil yang diharapkan**	**Hasil pengujian**
Button Back	Tekan *button back* dengan klik kiri pada mouse	Aplikasi akan Kembali ke halaman menu utama	Sesuai

g. Pengujian Halaman Win Scene

Tabel 12. Pengujian *win scene*

Pengujian	**Skenario Uji**	**Hasil yang diharapkan**	**Hasil pengujian**
Button replay	Tekan *button replay*	Aplikasi akan kembali ke *scene* gameplay	Sesuai
Button menu	Tekan *button menu* dengan klik kiri pada mouse	Aplikasi akan Kembali ke halaman menu utama	Sesuai

h. Pengujian Halaman Lost Scene

Tabel 13. Pengujian lost scene

Pengujian	**Skenario Uji**	**Hasil yang diharapkan**	**Hasil pengujian**
Button restart	Tekan *button restart*	Aplikasi akan kembali ke *scene* gameplay	Sesuai
Button menu	Tekan *button menu* dengan klik kiri pada mouse	Aplikasi akan Kembali ke halaman menu utama	Sesuai

Terlihat dari pengujian yang telah dilakukan bahwa aplikasi berjalan dengan baik dan memberikan keluaran atau hasil yang benar secara fungsional, dan sistem berjalan sesuai dengan yang diharapkan.

Setelah melakukan *Black* box *testing*, langkah selanjutnya adalah merancang kuesioner untuk melihat umpan balik pengguna. Kuesioner berisi beberapa pernyataan, yang akan diberikan kepada pengguna dan jawaban akan dievaluasi. Aspek yang dinyatakan dalam kuesioner adalah:

a. Tampilan antarmuka dari aplikasi *game* ini menarik.
b. Petunjuk untuk memainkan *game* Tales of Seleine mudah dipahami.
c. Tampilan objek *sprites* seperti *player*, musuh, pohon dll cukup menarik.
d. Objek musik dan efek suara menarik.
e. Segi konfigurasi mudah dipahami.
f. Game ini menyenangkan dan juga sulit untuk ditaklukan.

4.1.3 Manual Program

Aplikasi *game* Tales of Seleine merupakan *game* yang dijalankan pada sistem berbasis desktop. Penulis akan menjelaskan cara menggunakannya, yaitu sebagai berikut:

1. Buka aplikasi *game* Tales of Seleine di PC atau Laptop pengguna.

2. Setelah terbuka, maka game akan menunjukan *splash screen* seperti pada gambar dibawah.



Gambar 23. Tampilan splash screen

3. Setelah *splash screen* selesai, maka *scene* menu utama akan tampil seperti pada gambar dibawah.

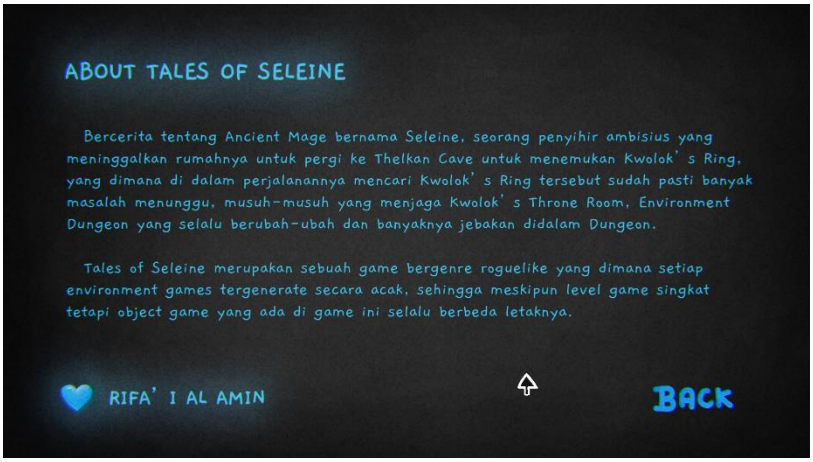


Gambar 24. Tampilan menu utama

Pada menu utama ini terdapat *button play* yang nantinya jika di tekan akan menampilkan *scene gameplay*, kemudian *button about* menampikan sekilas tentang *game* Tales of

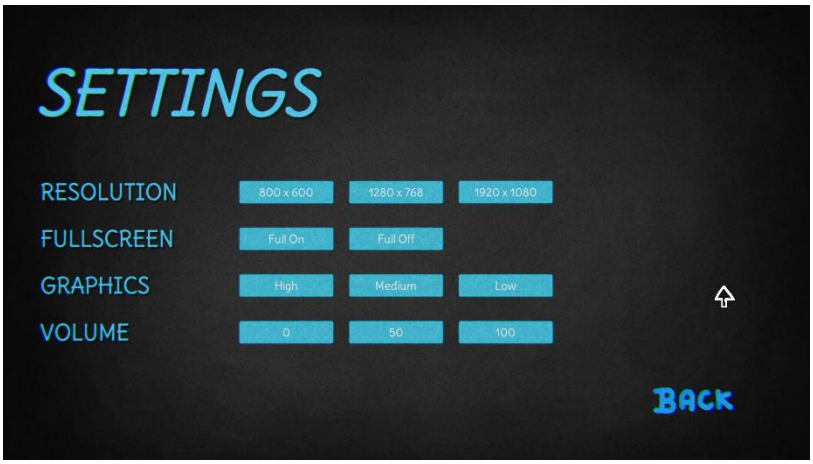
Seleine, *button settings* untuk menampilkan pengaturan *game* seperti resolusi, *fullscreen*, *graphics* dan *volume*, *button credits* untuk menampilkan nama yang berkontribusi pada *game* ini, terakhir *button exit* untuk menutup *game*.

4. Ketika *button about* ditekan maka *scene about* akan tampil dan jika *button back* ditekan maka *scene* menu utama akan tampil kembali.



Gambar 25. Tampilan halaman about

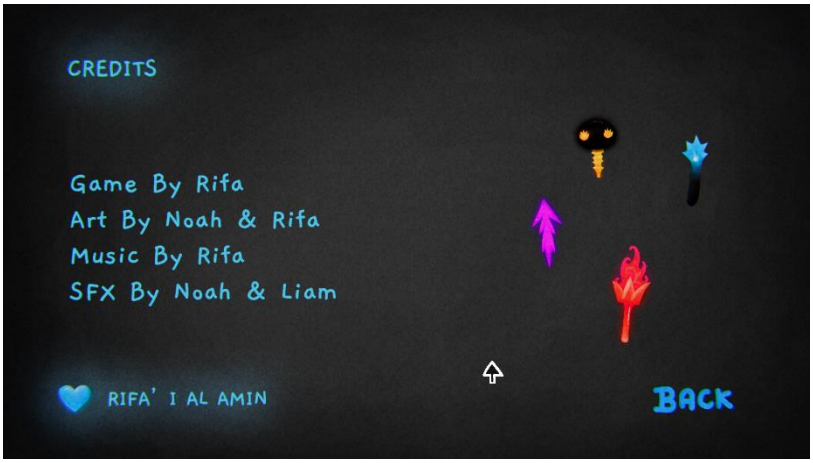
5. Setelah kembali ke menu utama dan button settings di tekan makan scene settings akan tampil seperti gambar dibawah ini.



Gambar 26. Tampilan menu settings

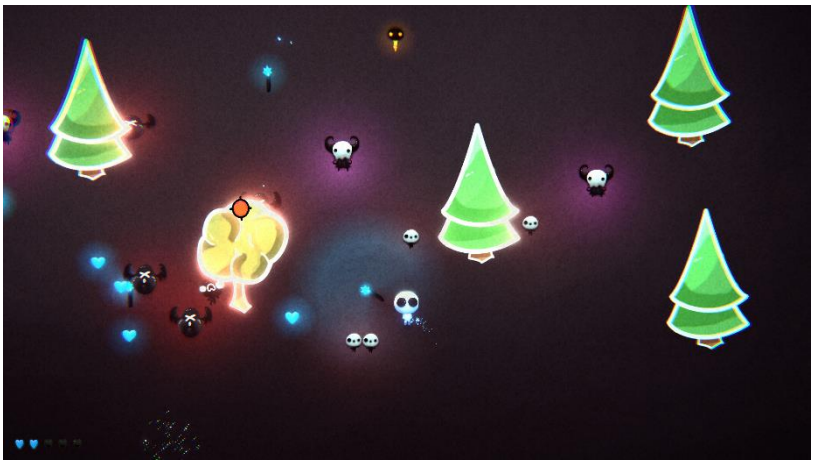
Untuk mengubah ukuran resolusi aplikasi dapat menggunakan *button size* yang ada pada baris *resolution*, untuk mengubah *game* dari *fullscreen* ke *windowed* dapat menggunakan *button* yang ada pada baris *fullscreen*, untuk mengubah *graphics* dapat menggunakan button pada baris *graphics*, untuk mengubah besar kecilnya suara dapat menggunakan *button* yang ada pada baris *volume* dan terakhir *button back* berguna untuk kembali ke menu utama.

6. Ketika sudah kembali ke menu utama dan button credits ditekan maka scene credits akan tampil yang dimana menunjukan nama yang berkontribusi dalam pembuatan game ini.



Gambar 27. Tampilan credits

7. Setelah menekan button play pada menu utama maka scene gameplay ini akan tampil.



Gambar 28. Tampilan gameplay Tales of Seleine

8. Ketika pengguna berhasil mendapatkan cincin di stage terakhir maka scene victory akan tampil dan jika button replay ditekan maka scene gameplay akan tampil kembali sedangkan jika button menu ditekan maka scene menu utama akan tampil.



Gambar 29. Tampilan victory scene

9. Ketika player mati maka scene lost akan tampil dan jika button restart ditekan maka scene gameplay akan tampil sedangkan jika button menu ditekan maka scene menu utama akan tampil.



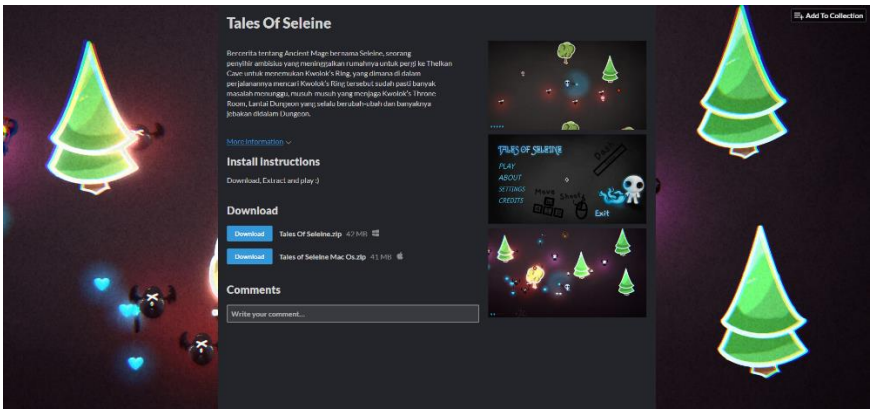
Gambar 30. Tampilan lost scene

10. Ketika button exit di tekan makan aplikasi akan tertutup.

4.1.4 Manual Instalasi

Pada bagian ini penulis akan menjelaskan Langkah – Langkah dalam melakukan download sampai dengan instalasi *game* Tales of Seleine dengan benar, Adapun langkah – langkahnya sebagai berikut:

1. *Game* Tales of Seleine dapat di download pada link https://twinklestar4798.itch.io/tales-of-seleine, kemudian pilih OS komputer atau laptop yang sesuai dengan pengguna dan pengguna dapat mulai mendownload filenya.



Gambar 31. Page Itch.io Tales of Seleine

2. Jika file sudah selesai di download maka selanjutnya *extract* zip file yang sudah pengguna download tadi.
3. Kemudian game siap langsung dimainkan, pengguna dapat membuka folder game yang sudah di extract dan sentuh file berekstensi .exe untuk windows dan .app untuk mac os.

4.2 Pembahasan

4.2.1 Pembahasan Listing Program

Pembahasan listing program merupakan tahapan yang menginterpretasikan listing yang digunakan dalam program sesuai dengan fungsi dan tujuan pada sistem yang dibuat. Berikut ini penulis akan menampilkan kode program untuk membuat *game* Tales of Seleine dengan menggunakan *Unity* dan *Visual Studio Code*.

a. Listing program pada menu utama

Pada pembuatan listing program menu utama ini hanya diletakan pada satu GameObject yang nantinya fungsi dari listing program ini dapat di gunakan di semua button yang ada pada menu utama maupun scene lainnya.

```
using UnityEngine.SceneManagement;

public class SceneTransition : MonoBehaviour
{

    private Animator transitionAnim;

    // Start is called before the first frame update
    void Start()
    {
        transitionAnim = GetComponent<Animator>();
    }

    // Update is called once per frame
    void Update()
    {

    }

    //fungsi load scene
    public void LoadScene (string sceneName)
    {
        StartCoroutine(Transition(sceneName));
    }

    //fungsi quit game
    public void QuitGame()
    {
        Debug.Log("QUIT");
        Application.Quit();
    }

    IEnumerator Transition(string sceneName)
    {
        transitionAnim.SetTrigger("end");
        yield return new WaitForSeconds(1);
        SceneManager.LoadScene(sceneName);
    }
}
```

Gambar 32. Listing Program Menu

b. Listing program pada scene settings

1. Listing program suara

Listing program ini mengatur konfigurasi suara dari game.

```
//fungsi settings volume
public void SetVolumeLow (float volume)
{
    audioMixer.SetFloat("volume", -80);

    //Debug.Log("volume is " + -80);
}

public void SetVolumeMed (float volume)
{
    audioMixer.SetFloat("volume", -12);

    //Debug.Log("volume is " + -12);
}

public void SetVolumeHigh (float volume)
{
    audioMixer.SetFloat("volume", 0);

    //Debug.Log("volume is " + 0);
}
```

Gambar 33. Listing program suara

2. Listing program grafik

Listing program ini mengatur tingkat kualitas grafik dari game.

```
//fungsi settings graphics
public void SetQualityHigh (int qualityIndex)
{
    QualitySettings.SetQualityLevel(0);
}

public void SetQualityMed (int qualityIndex)
{
    QualitySettings.SetQualityLevel(1);
}

public void SetQualityLow (int qualityIndex)
{
    QualitySettings.SetQualityLevel(2);
}
```

Gambar 34. Listing program grafik

3. Listing program fullscreen

Listing program ini mengatur keadaan fulllscreen ataupun mode window dari game.

```
//fungsi bool fullscreen
public void SetFullscreen (bool isFullscreen)
{
    Screen.fullScreen = isFullscreen;

    //Debug.Log("Fullscreen is" + isFullscreen);
}
```

Gambar 35. Listing program fullscreen

4. Listing program resolusi

Listing program ini mengatur ukuran resolusi tampilan dari game.

```
//fungsi ubah resolusi
public void SetResolution(int resolutionIndex)
{
    //Resolution resoulution = resolutions[resolutionIndex];

    switch (resolutionIndex)
    {
        case 0 :
        Screen.SetResolution(800, 600, Screen.fullScreen);
        //Debug.Log("Resolution is 800 x 600");
        break;

        case 1 :
        Screen.SetResolution(1280, 768, Screen.fullScreen);
        //Debug.Log("Resolution is 1024 x 768");
        break;

        case 2 :
        Screen.SetResolution(1920, 1080, Screen.fullScreen);
        //Debug.Log("Resolution is 1920 x 1080");
        break;

    }
}
```

Gambar 36. Listing program resolusi

c. Listing program pada gameplay

Pada listing program gameplay ini mengatur *spawn random object*, pergerakan *player*, musuh, dan senjata.

1. Listing program *player*

a. Listing program pergerakan *player*

```
public float speed;

private Rigidbody2D rb;
private Animator anim;

private Vector2 moveAmount;
```

Gambar 37. Variabel player move

```
//player dash
private float activeMoveSpeed;
public float dashSpeed;
public float dashLength;
public float dashCooldown;
public float dashInvisible, dashCoolCounter;
private float dashCounter;

public GameObject dustWalk;
```

Gambar 38. Variabel dash

```
// Start is called before the first frame update
void Start()
{
    rb = GetComponent<Rigidbody2D>();
    anim = GetComponent<Animator>();
    sceneTransitions = FindObjectOfType<SceneTransition>();
    activeMoveSpeed = speed;

}
```

Gambar 39. Void start player

```
// Update is called once per frame
void Update()
{
    //movement player
    Vector2 moveInput = new Vector2(Input.GetAxisRaw("Horizontal"), Input.GetAxisRaw("Vertical"));
    //moveAmount = moveInput.normalized * speed;
    moveAmount = moveInput.normalized * activeMoveSpeed;

    //cek jika player gerak, animasi jalan
    if ( moveInput != Vector2.zero)
    {
        anim.SetBool("isRunning", true);
        Instantiate(dustWalk, transform.position, Quaternion.identity);
    } else
    {
        anim.SetBool("isRunning", false);
    }

    //kondisi player dash
    if(Input.GetKeyDown(KeyCode.Space))
    {
        if(dashCoolCounter < 0 && dashCounter <= 0)
        {
            activeMoveSpeed = dashSpeed;
            dashCounter = dashLength;
        }
    }

    if(dashCounter > 0)
    {
        dashCounter -= Time.deltaTime;
        if(dashCounter <= 0)
        {
            activeMoveSpeed = speed;
            dashCoolCounter = dashCooldown;
        }
    }

    if( dashCoolCounter > 0 )
    {
        dashCoolCounter -= Time.deltaTime;
    }

}
```

Gambar 40. Void update player

```
private void FixedUpdate()
{
    //menggerakan player
    rb.MovePosition(rb.position + moveAmount * Time.fixedDeltaTime);
}
```

Gambar 41. Void fixed update player

b. Listing program darah *player*

```
public int health;

public Image[] hearts;
public Sprite fullHearth;
public Sprite emptyHearth;

public Animator hurtAnim;
```

Gambar 42. Variabel darah player

```
//mengurangi darah player jika terkena damage
public void TakeDamage (int damageAmount)
{
    health -= damageAmount;

    //memanggil fungsi perubah gambar hati
    UpdateHealthUI(health);

    //manggil hurt panel
    hurtAnim.SetTrigger("sakit");

    //menghapus player dari scene jika mati
    if( health <= 0)
    {
        //manggil scene kalah
        sceneTransitions.LoadScene("Lose");

        Destroy(gameObject);
    }
}
```

Gambar 43. Void mengurangi darah player

```
//hearth berubah dari biru ke hitam
void UpdateHealthUI (int currentHealth)
{
    for (int i = 0; i < hearts.Length; i ++)
    {
        if(i < currentHealth)
        {
            hearts[i].sprite = fullHearth;
        } else
        {
            hearts[i].sprite = emptyHearth;
        }
    }
}
```

Gambar 44. Merubah sprites hearth player

```
//heal dengan pickup
public void Heal(int healAmount)
{
    if( health + healAmount > 5)
    {
        health = 5;
    } else
    {
        health += healAmount;
    }
    UpdateHealthUI(health);
}
```

Gambar 45. Menyembuhkan player dengan item

c. Listing program *pickup item*

```
//merubah senjata jika dapat pickup dari scene
public void ChanceWeapon (Weapon weaponToEquip)
{
    Destroy(GameObject.FindGameObjectWithTag("Weapon"));
    Instantiate(weaponToEquip, transform.position, transform.rotation, transform);
}
```

Gambar 46. Merubah senjata dari drop item

2. Listing program *spawn random object*

Listing program ini mengatur *spawn object* dengan posisi yang berubah ubah sesuai lokasi *GameObject* yang telah ada di *scene.*

```
public GameObject[] objects;

// Start is called before the first frame update
void Start()
{
    int rand = Random.Range(0, objects.Length);
    Instantiate(objects[rand], transform.position, Quaternion.identity);
}
```

Gambar 47. Spawn random objects

3. Listing program musuh

a. Listing program mengejar *player* dan mengurangi darah musuh.

```
// Start is called before the first frame update
public virtual void Start()
{
    player = GameObject.FindGameObjectWithTag("Player").transform;
}
```

Gambar 48. Program mengejar player

```
//darah berkurang terkena damage
public void TakeDamage (int damageAmount)
{
    health -= damageAmount;

    if( health <= 0)
    {

        //jatuhin random item kalo dead
        int randomNumber = Random.Range(0, 101);
        if( randomNumber < pickupChance)
        {
            GameObject randomPickup = pickups[Random.Range(0, pickups.Length)];
            Instantiate(randomPickup, transform.position, transform.rotation);
        }

        //jatuhin health item kalo dead
        int randomHealth = Random.Range(0, 101);
        if( randomHealth < healthPickupChance)
        {
            Instantiate(healthPickup, transform.position, transform.rotation);
        }


        //munculin partikel fx ketika enemy dead
        Instantiate(deathFX, transform.position, Quaternion.identity);

        /*munculin jejak blood ketika enemy dead
        Instantiate(blood, transform.position, Quaternion.identity);*/

        Destroy(gameObject);
    }

}
```

Gambar 49. Darah musuh berkurang jika kena serangan player

b. Listing program pergerakan dan serangan melee enemy

```
public float stopDistance;

private float attackTime;

public float attackSpeed;

// Update is called once per frame
void Update()
{
    if(player != null)
    {
        if(Vector2.Distance(transform.position, player.position) > stopDistance)
        {
            transform.position = Vector2.MoveTowards(transform.position, player.position, speed * Time.deltaTime);
        } else
        {
            if(Time.time >= attackTime)
            {
                //serang
                StartCoroutine(Attack());
                attackTime = Time.time + timeBetweenAttack;
            }
        }
    }
}
```

Gambar 50. Variabel dan void update melee enemy

```
IEnumerator Attack()
{
    player.GetComponent<Player>().TakeDamage(damage);

    Vector2 originalPosition = transform.position;
    Vector2 targetPosition = player.position;

    float percent = 0;
    while (percent <= 1)
    {
        percent += Time.deltaTime * attackSpeed;
        float formula = (-Mathf.Pow(percent, 2) + percent) * 4;
        transform.position = Vector2.Lerp(originalPosition, targetPosition, formula);
        yield return null;
    }
}
```

Gambar 51. Program melee enemy menyerang

c. Listing program range enemy

```
// Update is called once per frame
void Update()
{
    if(player != null)
    {
        if(Vector2.Distance(transform.position, player.position) > stopDistance)
        {
            transform.position = Vector2.MoveTowards(transform.position, player.position, speed * Time.deltaTime);
        }


        /*kabur dari player (spawn ketmpt random)
        if(Vector2.Distance(transform.position, player.position) < runAway)
        {
            transform.position -= player.position;
        } */

        if(Time.time >= attackTIme)
        {
            attackTIme = Time.time + timeBetweenAttack;
            anim.SetTrigger("attack");
        }
    }
}
```

Gambar 52. Program pergerakan range enemy

```
public void RangedAttack ()
{
    //arah weapon ke target
    Vector2 direction = player.position - shotPoint.position;
    float angle = Mathf.Atan2(direction.y, direction.x) * Mathf.Rad2Deg;
    Quaternion rotation = Quaternion.AngleAxis(angle - 90, Vector3.forward);
    shotPoint.rotation = rotation;

    Instantiate(enemyBullet, shotPoint.position, shotPoint.rotation);
}
```

Gambar 53. Fungsi menyerang range enemy

d. Listing program summoner dan minion

```
public float minX;
public float maxX;
public float minY;
public float maxY;

private Vector2 targetPosition;
private Animator anim;

//summon minion
public float timeBetweenSummons;
private float summonTime;

public Enemy enemyToSummon;

public float meleeAtackSpeed;
public float stopDistance;
private float timer;
```

Gambar 54. Variabel summoner dan minion

```
// Start is called before the first frame update
public override void Start()
{
    base.Start();
    float randomX = Random.Range(minX, maxX);
    float randomY = Random.Range(minY, maxY);
    targetPosition = new Vector2(randomX, randomY);
    anim = GetComponent<Animator>();
}
```

Gambar 55.Void start program summoner

```
// Update is called once per frame
void Update()
{
    if(player != null)
    {

        //gerak ke tempat random
        if(Vector2.Distance(transform.position, targetPosition) > .5f)
        {
            transform.position = Vector2.MoveTowards(transform.position, targetPosition, speed * Time.deltaTime);
            anim.SetBool("isRunning", true);
        } else
        {
            anim.SetBool("isRunning", false);

            if(Time.time >= summonTime)
            {
                summonTime = Time.time + timeBetweenSummons;
                anim.SetTrigger("summon");
            }
        }
```

Gambar 56. Bergerak ke tempat random

```
        if(Vector2.Distance(transform.position, player.position) < stopDistance)
        {
            if(Time.time > timer)
            {
                //serang
                StartCoroutine(Attack());
                timer = Time.time + timeBetweenAttack;
            }
        }
    }
}
```

Gambar 57. Menyerang player jika dalam jangkauan

```
IEnumerator Attack()
{
    player.GetComponent<Player>().TakeDamage(damage);

    Vector2 originalPosition = transform.position;
    Vector2 targetPosition = player.position;

    float percent = 0;
    while (percent <= 1)
    {
        percent += Time.deltaTime * meleeAtackSpeed;
        float formula = (-Mathf.Pow(percent, 2) + percent) * 4;
        transform.position = Vector2.Lerp(originalPosition, targetPosition, formula);
        yield return null;
    }
}
```

Gambar 58. Program menyerang player

```
public void summon()
{
    if(player != null)
    {
        Instantiate(enemyToSummon, transform.position, transform.rotation);
    }
}
```

Gambar 59. Memanggil minion

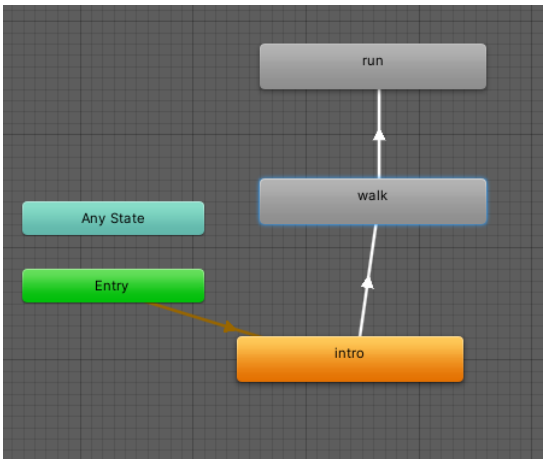
e. Listing program state machine boss

```
//ganti state animasi boss dari walk ke run
if(health <= halfHealth)
{
    anim.SetTrigger("stage2");
} else
{

    //spawn enemies dengan rentang waktu
    if(Time.time >= summonTime)
        {
            summonTime = Time.time + timeBetweenSummons;
            Enemy randomEnemy = enemies[Random.Range(0, enemies.Length)];
            Instantiate(randomEnemy, transform.position + new Vector3(spawnOffset, spawnOffset, 0), transform.rotation);
        }
}
```

Gambar 60. State machine boss

Boss menggunakan *state machine behaviour* yang dimana jika darah boss sisa setengah dari darah awal maka boss akan masuk ke stage ke dua yaitu berlari mengejar *player*.



Gambar 61. State Machine Behaviour Boss

Listing program stage patrol behaviour

```
private GameObject[] patrolPoints;
public float speed;
int randomPoint;
```

Gambar 62. Variabel patrol behaviour

```
// OnStateEnter is called when a transition starts and the state machine starts to evaluate this state
override public void OnStateEnter(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
{
    patrolPoints = GameObject.FindGameObjectsWithTag("PatrolPoints");
    randomPoint = Random.Range(0, patrolPoints.Length);
}
```

Gambar 63. Void OnStateEnter patrol behaviour

```
// OnStateUpdate is called on each Update frame between OnStateEnter and OnStateExit callbacks
override public void OnStateUpdate(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
{
    animator.transform.position = Vector2.MoveTowards(animator.transform.position,
    patrolPoints[randomPoint].transform.position, speed * Time.deltaTime);

    if ( Vector2.Distance(animator.transform.position, patrolPoints[randomPoint].transform.position) < 0.1f)
    {
        randomPoint = Random.Range(0, patrolPoints.Length);
    }
}
```

Gambar 64. Void OnStateUpdate patrol behaviour

Listing program stage chase behaviour

```
private GameObject player;
public float speed;
```

Gambar 65. Variabel chase behaviour

```
// OnStateEnter is called when a transition starts and the state machine starts to evaluate this state
override public void OnStateEnter(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
{
    player = GameObject.FindGameObjectWithTag("Player");
}
```

Gambar 66. Void OnStateEnter chase behaviour

```
// OnStateUpdate is called on each Update frame between OnStateEnter and OnStateExit callbacks
override public void OnStateUpdate(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
{
    if(player != null)
    {
        animator.transform.position = Vector2.MoveTowards(animator.transform.position,
        player.transform.position, speed * Time.deltaTime);
    }
}
```

Gambar 67. Void OnStateUpdate chase behaviour

4. listing program senjata

listing program ini mengatur *aim mouse* kearah *cursor*, mengatur kecepatan peluru pada setiap tembakan, dan berapa *damage* yang dihasilkan serta kecepatan peluru bergerak.

```
public GameObject projectile;
public Transform shotPoint;
public float timeBetweenShots;

private float shotTime;

Animator cameraAnim;
```

Gambar 68. Variabel senjata

```
// Start is called before the first frame update
void Start()
{
    cameraAnim = Camera.main.GetComponent<Animator>();
}
```

Gambar 69. Void start senjata

```
// Update is called once per frame
void Update()
{

    //arah weapon ke cursor
    Vector2 direction = Camera.main.ScreenToWorldPoint(Input.mousePosition) - transform.position;
    float angle = Mathf.Atan2(direction.y, direction.x) * Mathf.Rad2Deg;
    Quaternion rotation = Quaternion.AngleAxis(angle - 90, Vector3.forward);
    transform.rotation = rotation;

    //nembak pake klik kiri mouse
    if (Input.GetMouseButton(0))
    {
        if(Time.time >= shotTime)
        {
            Instantiate(projectile, shotPoint.position, transform.rotation);
            cameraAnim.SetTrigger("shake");
            shotTime = Time.time + timeBetweenShots;
        }
    }
}
```

Gambar 70. Void update senjata

Kemudian listing program projectile

```
public float speed;
public float lifeTime;

public GameObject explosion;

public int damage;

public GameObject soundObject;
```

Gambar 71. Variabel peluru

```
// Start is called before the first frame update
void Start()
{
    Invoke("DestroyProjectile", lifeTime);
    Instantiate(soundObject, transform.position, transform.rotation);
}
```

Gambar 72. Void start peluru

```
// Update is called once per frame
void Update()
{
    transform.Translate(Vector2.up * speed * Time.deltaTime);
}
```

Gambar 73. Void update peluru

```
void DestroyProjectile()
{
    Instantiate(explosion, transform.position, Quaternion.identity);
    Destroy(gameObject);
}
```

Gambar 74. Menimbulkan efek ledakan

```
void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.tag == "Enemy")
    {
        collision.GetComponent<Enemy>().TakeDamage(damage);
        DestroyProjectile();
    }

    if (collision.tag == "Boss")
    {
        collision.GetComponent<Boss>().TakeDamage(damage);
        DestroyProjectile();
    }
}
```

Gambar 75. Mendeteksi objek dengan tag enemy

dan boss

4.2.2 Pembahasan Hasil Respon Pengguna

Apabila input, proses, dan output aplikasi berjalan dengan baik sesuai dengan tujuan yang ingin dicapai, maka program dianggap berhasil. Oleh karena itu diperlukan suatu pengujian untuk mengetahui kelebihan dan kekurangan program. Agar dapat menganalisis data maka diperlukan pengujian melalui kuesioner dengan menggunakan skala likert.

Tabel 14. Skor Pilihan Jawaban

Jawaban	**Sangat tidak setuju**	**Tidak setuju**	**Setuju**	**Sangat Setuju**
Skor	1	2	3	4

Dari hasil kuesioner tentang Rancang bangun game bergenre roguelike adventure “Tales of Seleine” ini maka diperoleh hasil pengamatan sebagai berikut:

Tabel 15. Jawaban Responden

No	Pertanyaan/Pernyataan	Jawaban Responden											Jumlah
		1	2	3	4	5	6	7	8	9	10	11	
1	Apakah menurut anda tampilan antarmuka pada game ini menarik?	3	3	3	3	3	4	4	4	4	4	3	38
2	Apakah menurut anda petunjuk untuk memainkan game Tales of Seleine mudah dipahami?	3	4	3	4	4	4	4	4	4	4	3	41
3	Apakah menurut anda objek sprites dalam game seperti player, musuh dan pohon - pohon cukup menarik?	3	3	2	4	3	3	3	4	4	4	2	35
4	Apakah menurut anda musik dan efek suara pada game ini menarik?	3	4	3	4	3	4	4	3	4	4	4	40
5	Apakah menurut anda dari segi konfigurasi mudah dipahami?	3	3	3	4	3	3	4	4	3	4	3	37
6	Apakah menurut anda game ini menyenangkan dan juga sulit untuk ditaklukan?	4	4	3	4	3	3	3	4	3	4	3	38
Jumlah													229

Untuk mendapatkan hasilnya kita harus mengetahui interval (jarak) dan interpretasi persen agar mengetahui penilaian dengan metode mencari interval skor persen (I) :

Rumus Interval

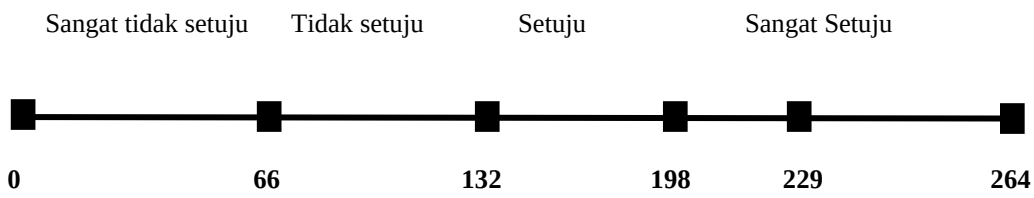
I = Jumlah Responden x Jumlah Kuesioner x Skor (likert)

Maka = 11 x 6 x 4 = 264

= 11 x 6 x 3 = 198

= 11 x 6 x 2 = 132

= 11 x 6 x 1 = 66



Maka penyelesaian dari kasus diatas adalah

= (Total Skor/Y) x 100%

= (229/264) x 100%

= 86,74%

Berdasarkan hasil responden dari keenam pertanyaan tersebut dapat disimpulkan bahwa aplikasi " Rancang Bangun Game Bergenre Roguelike Adventure Tales of Seleine" berada pada kategori yang sangat baik, dengan rata – rata persentase adalah 86,74%. Hasil kuesioner dari respon pengguna.



Gambar 76. Hasil Kuesioner

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan dari pemaparan dari bab – bab sebelumnya seperti analisis, hasil penelitian, pembahasan serta implementasi, maka penulis dapat menarik kesimpulan sebagai berikut:

1. Telah dibuat " Rancang Bangun Game Bergenre Roguelike Adventure Tales of Seleine"
2. Aplikasi ini dibuat menggunakan Unity dengan bahasa pemprograman C#.
3. Dari hasil pengujian, dapat diketahui aplikasi dapat dijalankan pada system dengan spesifikasi minimum Processor 2 GHz, memiliki 2 GB RAM, dengan Graphics 256 mb video memory, aplikasi dinyatakan optimal dan tidak masalah pada saat dijalankan.
4. Game Tales of Seleine ini menggunakan sistem control keyboard dan mouse.

5.2 Saran

Adapun saran dan harapan yang penulis berikan untuk penelitian selanjutnya adalah sebagai berikut:

a. Menambahkan algoritma *pathfinding* untuk musuh seperti A* atau Djikstra
b. Menambahkan sistem inventori
c. Menambahkan animasi untuk objek lainnya seperti animasi untuk pohon
d. Diharapkan pengembangan *game* mengarah ke *game mobile* berbasis *android* dan berwarna khas kedaerahan lokal kalimantan tengah.

DAFTAR PUSTAKA

Adams, E., 2013. *Fundamentals of Game Design.* 3 ed. s.l.:New Riders.

Cerny, V. & Dechterenko, F., 2015. Entertainment Computing - ICEC 2015. Volume 9353, pp. 261-271.

Dahria, M., 2008. Kecerdasan Buatan (Artificial Intelligence). *Jurnal Saintikom,* 5(2), p. 185.

Hendini, A., 2016. Pemodelan Uml Sistem Informasi Monitoring Penjualan Dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak). *JURNAL KHATULISTIWA INFORMATIKA,* 6(2), pp. 107-116.

Hermawan, H., 2019. Implementasi Algoritma A-Star Pada Permainan Komputer Roguelike Berbasis Unity. *Jurnal Algoritma, Logika dan Komputasi* , 2(1), pp. 111-120.

Irwandi, P., Erlansari, A. & Effendi, R., 2016. Perancangan Game First Person Shooter (Fps) "Boar Hunter" Berbasis Virtual Reality. *Jurnal Rekursif,* 4(1).

King, A., 2015. *Elemen Desain Kunci dari Roguelikes.* [Online] Available at: https://gamedevelopment.tutsplus.com/id/articles/the-key-design-elements-of-roguelikes--cms-23510
[Accessed 10 03 2020].

Kovess-Masfety, V. et al., 2016. Is time spent playing video games associated with mental health, cognitive and social skills in young children?. *Social Psychiatry and Psychiatric Epidemiology,* 51(3), pp. 349-357.

Kurniati, A., Nadia, Tanzil, F. & Purnomo, F., 2015. Game Development "Tales of Mamochi" With Role Playing Game Concept Based on Android. *Procedia Computer Science,* Volume 59, pp. 392-399.

Mongi, L. S., Lumenta, A. S. M. & Sambul, A. M., 2018. Rancang Bangun Game Adventure of Unsrat Menggunakan Game Engine Unity. *E-Journal Teknik Informatika,* 14(1).

Rachmatullah, A., 2012. *Mempelajari C#: Bahasa Pemrograman Modern.* s.l.:s.n.

Ridoi, M., 2018. *Cara Mudah Membuat Game Edukasi dengan Construct 2: Tutorial sederhana Construct 2.* s.l.:Maskha.

Roedavan, R., 2016. *Unity Tutorial Game Engine - Edisi Revisi.* Bandung: INFORMATIKA.

Roguebasin, 2008. *Berlin Interpretation.* [Online] Available at: http://www.roguebasin.com/index.php?title=Berlin_Interpretation [Accessed 10 03 2020].

Setiawan, I., 2006. Perancangan Software Embedded System Berbasis FSM. *Jurnal Teknik Elektro,* pp. 1-2.

Setiawan, M., Lumenta, A. S. & Tulenan, V., 2017. Aplikasi Pembelajaran Interaktif Berbasis Multimedia Untuk Sekolah Dasar (Studi Kasus : Sd Negeri I Bitung, Kelas Vi). *Jurnal Teknik Elektro dan Komputer,* 6(4), pp. 194-204.

Singkoh, R. T., Lumenta, A. S. & Tulenan, V., 2016. Perancangan Game FPS (First Person Shooter) Police Personal Training. *E-Journal Teknik Elektro dan Komputer,* 5(1).

SuperData, 2019. *2019 Year In Review: Digital Games and Interactive Media,* s.l.: s.n.

Tilak, G. & Lanjekar, K., 2019. Cognitive Benefits Of Playing Video Game. *Journal of Emerging Technologies and Innovative Research,* 6(5), p. 325.

Vince, 2018. *The Many Different Types of Video Games & Their Subgenres.* [Online]

Available at: https://www.idtech.com/blog/different-types-of-video-game-genres [Accessed 09 March 2020].

Wolf, M. J. ed., 2008. The Video Game Explosion: A History from PONG to Playstation and Beyond. In: s.l.:ABC-CLIO, p. 259.

Yee, N., 2015. *The Gamer Motivation Model in Handy Reference Chart and Slides.* [Online] Available at: https://quanticfoundry.com/2015/12/15/handy-reference/ [Accessed 5 March 2020].

Yee, N., 2016. *7 Things We Learned About Primary Gaming Motivations From Over 250,000 Gamers.* [Online] Available at: https://quanticfoundry.com/2016/12/15/primary-motivations/ [Accessed 5 March 2020].

DAFTAR LAMPIRAN

Lampiran 1. Surat Tugas Pembimbing

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp. 0536-3224593, 3225515 Fax. 0536-3225515 Palangkaraya
surel *(email)* : humas@stmikplk.ac.id – laman *(website)* : www.stmikplk.ac.id

SURAT TUGAS
No.37/STMIK-3.C.2/AU/I/2020

Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya menugaskan nama- nama tersebut di bawah ini :

1. Nama : Lili Rusdiana, M.Kom
   NIK : 198707282011007
   Sebagai Pembimbing I Dalam Pembuatan Program
2. Nama : Susi Hendartie, M.Kom
   NIK : 197803202008001
   Sebagai Pembimbing II Dalam Penulisan Tugas Akhir

Untuk membimbing Tugas Akhir mahasiswa : :

Nama	:	Rifa'i Al Amin
NIM	:	C1655201015
Program Studi	:	TEKNIK INFORMATIKA (55201)
Tanggal Daftar	:	27 Februari 2020
Judul Tugas Akhir	:	Rancang Bangun Game Bergenre Roguelike Adventure "Tales of Seleine".

Demikian surat ini dibuat agar dapat dipergunakan sebagaimana mestinya dan dilaksanakan dengan penuh tanggung jawab.

PalangkaRaya, 27 Februari 2020
Ketua Program Studi,

Hotmian Sitohang, M.Kom.
NIK. 198503282008002

Tembusan :
1. Pembimbing I dan II
2. Mahasiswa yang bersangkutan
3. Arsip

Lampiran 2. Surat Tugas Penguji Tugas Akhir

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangkaraya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

SURAT TUGAS PENGUJI TUGAS AKHIR

No. 14/STMIK-3.C.2/AK/I/2021

Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya menugaskan kepada nama- nama berikut :

1. Nama : Herkules, S.Kom., M.Cs.
   NIK : 198510042010106
   Sebagai Ketua
2. Nama : Rudini, M.Pd.
   NIK : 198709172015105
   Sebagai Sekretaris
3. Nama : Elia Zakaria, M.T.
   NIK : 199205262016104
   Sebagai Anggota
4. Nama : Lili Rusdiana, M.Kom.
   NIK : 198707282011007
   Sebagai Anggota
5. Nama : Susi Hendartie, M.Kom.
   NIK : 197803202008001
   Sebagai Anggota

Tim Penguji Tugas Akhir Mahasiswa :

Nama	:	Rifa'i Al Amin
NIM	:	C1655201015
Hari/ Tanggal Ujian	:	Sabtu, 30 Januari 2021
Waktu	:	09.00 WIB
Judul Tugas Akhir	:	Rancang Bangun Game Bergenre Roguelike Adventure "Tales of Seleine"

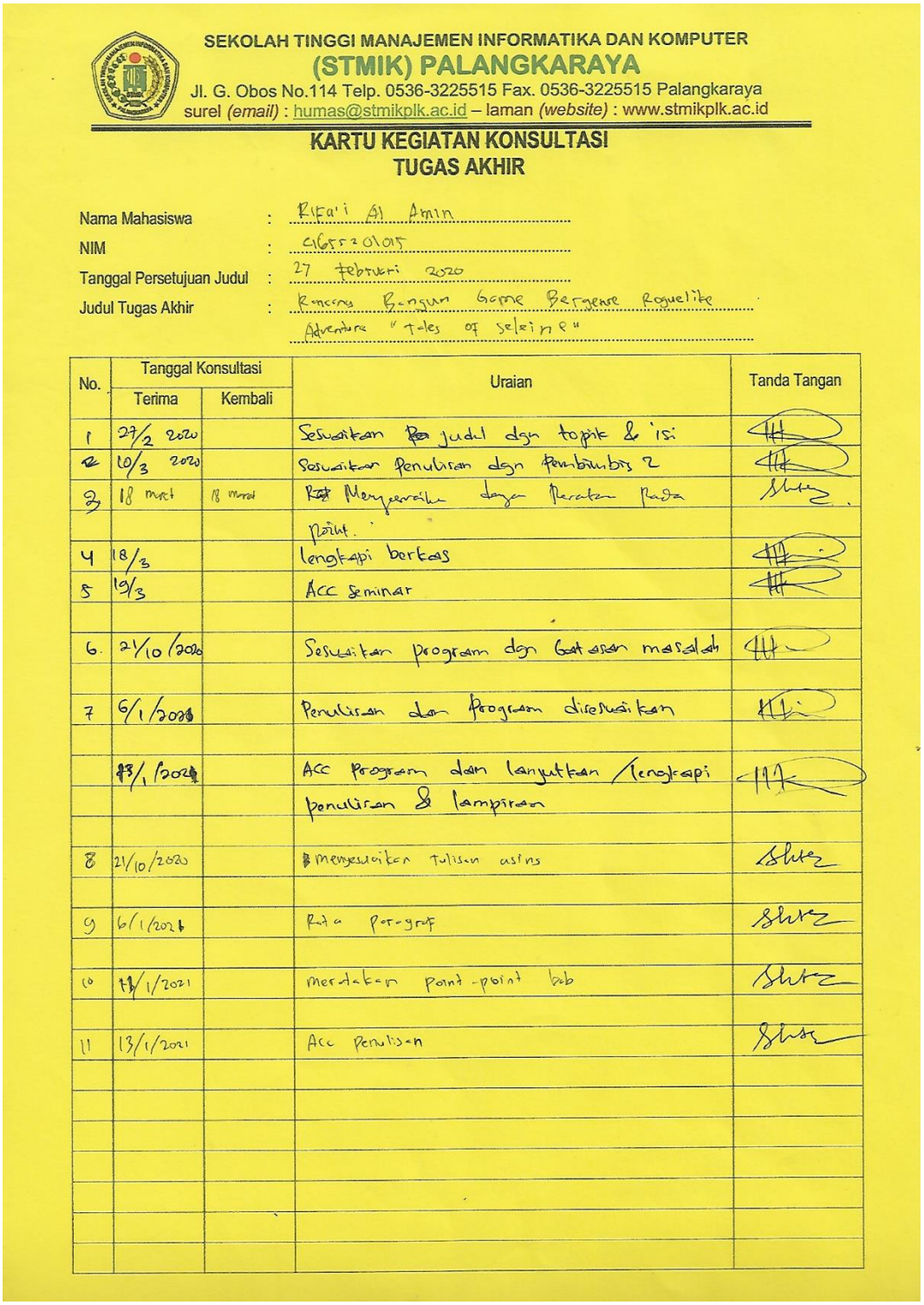
Demikian surat ini dibuat agar dapat dipergunakan sebagaimana mestinya dan dilaksanakan dengan penuh tanggung jawab.

Palangka Raya, 29 Januari 2021
Ketua Program Studi,

Hotmian Sitohang, M.Kom.
NIK. 198503282008002

Tembusan :
1. Dosen Penguji
2. Mahasiswa yang Bersangkutan
3. Arsip

Lampiran 3. Kartu Kegiatan Konsultasi Akhir



SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp. 0536-3225515 Fax. 0536-3225515 Palangkaraya
surel *(email)* : humas@stmikplk.ac.id – laman *(website)* : www.stmikplk.ac.id

KARTU KEGIATAN KONSULTASI
TUGAS AKHIR

Nama Mahasiswa : Rifa'i Al Amin
NIM : 61655201015
Tanggal Persetujuan Judul : 27 Februari 2020
Judul Tugas Akhir : Rancang Bangun Game Bergenre Roguelike Adventure "Tales of Selein"

No.	Tanggal Konsultasi		Uraian	Tanda Tangan
	Terima	Kembali		
1	27/2 2020		Sesuaikan judul dgn topik & isi	
2	10/3 2020		Sesuaikan penulisan dgn pembimbing 2	
3	18 maret	18 maret	Menyesuaikan dengan peraturan pada point.	
4	18/3		lengkapi berkas	
5	19/3		Acc seminar	
6	21/10/2020		Sesuaikan program dgn batasan masalah	
7	6/1/2021		Penulisan dan program disesuaikan	
	13/1/2021		Acc Program dan lanjutkan /lengkapi penulisan & lampiran	
8	21/10/2020		menyesuaikan tulisan asing	
9	6/1/2021		Rata perparagraf	
10	13/1/2021		merataken point-point bab	
11	13/1/2021		Acc penulisan	

Lampiran 4. Kartu Kegiatan Seminar Proposal Tugas Akhir

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp. 0536-3225515 Fax. 0536-3236933 Palangkaraya
Email : stmikplk@gmail.com -- website : www.stmikplk.ac.id

KARTU KEGIATAN SEMINAR PROPOSAL SKRIPSI

Nama Mahasiswa : RIFA'I AL AMIN
N I M : C1C552010015
Jurusan : ~~Sistem Informasi~~/ Teknik Informatika*)

No.	Hari/ Tanggal	Judul	Mahasiswa Penyaji	Nama Tim Dosen	Tanda Tangan
1	Rabu, 22-01-2020	Implementasi Virtual Reality Dalam Pengenalan Pelayanan kampus STMIK PalangkaRaya	Jurisdul Janeh	1. Sulistyowati, S.kom, Mcs 2. Hartcles, S.kom, M.cs 3. Catherina Elsina M.pd	
2	Rabu 22-01-2019	Penerapan Teknologi Augmented Reality Sebagai media promosi Perguruan	Ayu Wulan Ndari	1. Sulistyowati, S.kom, Mcs 2. Lili Rustiana, M.kom 3 Rudini, M.pd	
3	Selasa 1 February 2020	Aplikasi Penyaluran Darah Pada Kampus UPN Palangkaraya Berbasis Android	Wirarda Agnes Tanya	1. Elia Zarana, M.Kom 2. Lili Rustiana, M.kom 3. M. Praetno, S.kom, M.Si	
4.	Rabu 5 februari 2020	Implementasi Deteksi Tepi untuk menentukan citra Bentuk Daun	Rido Azwa Rizal	1. Anugra Andri Pamuji, S.kom, M.T. 2. Hotmian Sitohang, M.kom 3. Bagus Prakasa Nugraha, M.T	
5.	Rabu 5 Februari 2020	Klasifikasi Jenis Burung Berdasarkan Pengenalan suara Dengan jaringan Syaraf Tiruan Back Propagation	Andie Tarmudi	1. Lili Rustiana, M.kom 2. Hartules, S.kom, M.cs 3. Hotmian Sitohang, M.kom	

Palangka Raya, 22 Januari 2020
Mahasiswa ybs,

Keterangan :
- *) Coret yang tidak perlu
- Harap kartu jangan sampai hilang.

Lampiran 5. Berita Acara

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3225515 Fax. 0536-3225515 Palangkaraya
surel *(email)* : stmikplk@gmail.com – laman *(website)* : www.stmikplk.ac.id

BERITA ACARA
UJIAN TUGAS AKHIR

Periode (Bulan) : Januari Tahun 2021

1. Hari/Tanggal Ujian : Sabtu / 30 Januari 2021
2. Waktu (Jam) : 09.00 WIB sampai dengan 11.00 WIB
3. Nama Mahasiswa : Rifai Al Amin
4. Nomor Induk Mahasiswa : C1655201015
5. Program Studi : Teknik Informatika
6. Tahun Angkatan : 2016
7. Judul Tugas Akhir : Rancang Bangun Game Bergenre Roguelike Adventure "Tales Of Seleine"
8. Dosen Penguji :

	Nama	Nilai	Tanda Tangan
1.	Herkules, S.Kom., M.Cs.	=	(.....)
2.	Rudini, M.Pd.	=	(.....)
3.	Elia Zakaria, M.T.	=	(.....)
4.	Lili Rusdiana, M.Kom.	=	(.....)
5.	Susi Hendartie, M.Kom.	=	(.....)

9. Hasil Ujian : **LULUS** / ~~**TIDAK LULUS**~~ *) NILAI = 81,8
   Dengan Perbaikan/ ~~Tanpa Perbaikan~~ *)
10. Catatan Penting : 1. Lama Perbaikan : 29 hari
    2. Jika lebih dari 1 (satu) bulan dikenakan sanksi berupa denda sebesar Rp. 600.000,- (Enam ratus ribu rupiah) per bulan dari tanggal ujian
    3. Jika lebih dari 3 (tiga) bulan dari tanggal ujian maka hasil ujian dibatalkan dan wajib mengajukan judul dan pembimbing baru

Palangkaraya, 30 Januari 2021

Mengetahui :
Ketua Program Studi Teknik Informatika,

Hotmian Sitohang, M.Kom.
NIK. 198503282008002

Ketua Penguji,

Herkules, S.Kom., M.Cs.
NIK. 198510042010106

Tembusan:

1. Kabag AKMA
2. Arsip Prodi
3. Mahasiswa yang bersangkutan

Dibawa saat konsultasi perbaikan dengan dosen penguji
**) Coret yang tidak perlu*