

# BAB I

## PENDAHULUAN

### A. Latar Belakang

Perkembangan ilmu teknologi hari demi hari semakin menunjukkan perkembangan teori-teori baru diberbagai bidang, teknologi *game* adalah salah satunya. Pada sekitar tahun 2000-an permainan yang dikenal dengan sebutan *game PlayStation* (PS) menjamur di daerah perkotaan hingga sebagian pedesaan di kawasan Indonesia, *game* ini hanya bisa dimainkan oleh satu atau dua orang pemain saja pada sebuah *console*. Pesawat televisi diperlukan sebagai media *output* atau tampilan. Beraneka macam *genre game* dapat dimainkan di perangkat *PlayStation* diantaranya yaitu *game* ber-genre RPG dengan tampilan 2D. Tetapi sayangnya kebanyakan *game – game* yang beredar dan terkenal dewasa ini hanyalah menonjolkan manfaat dan tujuan sebagai media untuk hiburan semata. Dan kebanyakan anak – anak, remaja bahkan dewasa menjadikan *game* sebagai pelepas penat, sebagai hiburan ditengah kesibukan

Kebanyakan dari *Gamers* menyukai *game – game adventure/ petualangan*, strategi dan yang paling digemari oleh anak – anak kecil dan remaja ialah *game* sepak bola sejenis *Winning Eleven*.

(*WE*) ataupun *PES*. Selain sebagai sarana hiburan, para *Gamers* biasanya juga menjadikan permainan untuk tujuan komersial. Dari sekian banyak *game*, disini mengangkat *game* Sejarah dengan mengangkat salah satu tokoh

Pahlawan yang berasal dari Sulawesi Selatan yaitu Sultan Hasanuddin dengan *genre game rpg*.

Sultan Hasanuddin merupakan pahlawan yang mengusir penjajah yang dikenal gigih, kegigihannya tersebut membuat pahlawan nasional satu ini mendapatkan julukan Ayam Jantan Dari Timur. Banyak hal yang telah diperjuangkan oleh Sutan Hasanuddin dalam mempertahankan tanah kelahirannya salah satunya yaitu saat melawan belanda dengan perlawanannya.

*Game* ini merupakan *game* yang menggambarkan perjuangan Sultan Hasanuddin dan pasukannya dalam menjaga wilayahnya dan untuk itu diperlukan 2 mode dalam *game* ini yaitu mode alur cerita maupun *survival*. Alur cerita dalam mode ini adalah bermain dengan alur cerita sejarah Sultan Hasanuddin dan untuk *Survival* adalah mode random denngan tujuan agar *game* lebih menarik dan seimbangan. Tujuan *game* ini yang utama ialah sebagai memperkenalkan tokoh pahlawan yang bagaimana memperjuangkan tanah kelahiran dan sebagai sarana untuk menjadikan alternatif untuk mengurangi kecenderungan pada *game* asing.

Berdasarkan hal diatas, perlu adanya *game* yang menggambarkan perjuangan pahlawan dalam menjaga dan mempertahankan tanah kelahiran untuk mengingat perjuangannya dengan memanfaatkan *game* yang sebaik – baiknya, bukan hanya sebagai media hiburan semata. Maka dari itu, berdasarkan hal diatas, peneliti membuat judul penelitian **“GAME 2 DIMENSI PERJUANGAN SULTAN HASANUDDIN MENGGUNAKAN ROLE PLAYING GAME BERBASIS DEKSTOP”**.

## **B. Rumusan Masalah**

Berdasarkan latar belakang diatas, maka dapat diidentifikasi beberapa permasalahan dalam membuat *Game 2 Dimensi Perjuangan Sultan Hasanuddin Menggunakan Role Playing Game Berbasis Dekstop* adalah sebagai berikut:

1. Bagaimana membuat *Game 2 Dimensi Perjuangan Sultan Hasanuddin Menggunakan Role Playing Game Berbasis Dekstop*?
2. Bagaimana memperkenalkan sejarah perjuangan Sultan Hasanuddin melalui permainan *game*?

## **C. Tujuan Penelitian**

Tujuan penelitian ini adalah untuk mengetahui implementasi dalam *Game 2 Dimensi Perjuangan Sultan Hasanuddin Menggunakan Role Playing Game Berbasis Dekstop* .

## **D. Batasan Masalah**

Agar penelitian ini terarah maka penulis memberikan batasan masalah sehingga permasalahan tidak menyimpang dari tujuan yang ditetapkan. Adapun batasan masalah tersebut adalah sebagai berikut :

1. *Game* ini bersifat *single player* dan *offline*.
2. *Game* ini dibuat dengan menggunakan *Game maker Studio*.
3. *Game* dengan tampilan 2D.
4. Mode alur cerita di ambil dari buku sejarah Sultan Hasanuddin.
5. *Game* ini hanya dapat dimainkan di desktop.

### **E. Manfaat Penelitian**

Manfaat penelitian ini adalah untuk memberikan alternatif hiburan disertai dengan pengetahuan tentang sejarah dan menjadikan penelitian ini untuk melakukan pengembangan *game* atau pemula yang ingin membuat *game*.

### **F. Sistematika Penulisan**

Secara garis besar sistematika penulisan yang dapat disajikan sebagai berikut:

#### **1. BAB I PENDAHULUAN**

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan dan manfaat, serta sistematika penulisan.

#### **2. BAB II TINJAUAN PUSTAKA**

Bab ini berisi tentang teori-teori yang menyangkut penelitian ini.

#### **3. BAB III METODE PENELITIAN**

Bab ini diuraikan mengenai jenis penelitian, lokasi dan waktu penelitian, dan tahap-tahap metode penelitian

#### **4. BAB IV HASIL DAN PEMBAHASAN**

Bab ini membahas tentang detail Pembuatan *Game 2 Dimensi Sejarah Perjuangan Sultan Hasanuddin Menggunakan Role Playing Game Berbasis Desktop*.

#### **5. BAB V PENUTUP**

Bab ini membahas tentang Kesimpulan dan Saran.

## BAB II

### TINJAUAN PUSTAKA

#### A. Kajian Hasil Penelitian Terdahulu

1. Penelitian dalam sebuah jurnal yang dilakukan oleh Risca Amelia (2020) dari Universitas Teknokrat Indonesia “*Game Edukasi Dan Cerita Interaktif Sejarah Kerajaan Di Sumatra Menggunakan Algoritma Fuzzy Sugeno Untuk Mengatur Perilaku NPC*”. Tujuan penelitian ini adalah memperkenalkan sejarah Kerajaan Di Sumatra berbasis interaktif. Interaktif merupakan hal yang saling berhubungan, mempengaruhi, serta adanya aksi dan reaksi sehingga dapat meningkatkan pemahaman konsep pada siswa. *Game* ini ditujukan kepada siswa dan siswi baik dari tingkat Sekolah Dasar sampai Sekolah Menengah Atas yang sedang mempelajari tentang sejarah kerajaan terutama di Pulau Sumatera.
2. Penelitian dalam sebuah jurnal yang dilakukan oleh (Tri, & Rio, 2021) dari Universitas Kuningan “*Pembuatan Game Edukasi Sejarah Kerajaan Sriwijaya Menggunakan RPG Maker MV*”. *Game* ini dibuat menggunakan *RPG Maker MV* yaitu sebuah jenis *game* yang biasanya memakai dasar cerita dan diubah menjadi sebuah permainan. Didalam *game* yang akan dibuat player berperan sebagai seorang tokoh utama, dimana tokoh utama tersebut sebagai ksatria kerajaan sriwijaya. Tujuan dari pemilihan sejarah diambil sebagai tema *game* adalah untuk mendorong penikmat *game* dalam hal ini mudah untuk lebih tau mengenai sejarah lampau dari indonesia terutama di sumatera.

3. Penelitian dalam sebuah jurnal yang dilakukan oleh (Lidya, Jhon, & Gerlan, 2021) dari Universitas Citra Bangsa “Perancangan *Game* Edukasi Sejarah Kemerdekaan Indonesia Menggunakan Aplikasi *Construct 2* Berbasis Android”. *Game* ini adalah media pembelajaran dibuat oleh peneliti yaitu pembelajaran *Game* Edukasi Sejarah Kemerdekaan Indonesia dengan Aplikasi *Construct 2* dengan *Game* ini terdapat permainan penyusunan *puzzle* tokoh pahlawan, *quiz*, dan juga menampilkan materi Sejarah Kemerdekaan Indonesia dan materi tokoh Sejarah yang memuat audio berisi narasi materi juga teks materi tersebut.

### **B. Game**

*Game* adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius atau dengan tujuan *refreshing*. Suatu cara belajar yang digunakan dalam menganalisa interaksi antara sejumlah pemain maupun perorangan yang menunjukkan strategi-strategi yang rasional.

Permainan terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri atau pun untuk meminimalkan kemenangan lawan. Peraturan-peraturan menentukan kemungkinan tindakan untuk setiap pemain, sejumlah keterangan diterima setiap pemain sebagai kemajuan bermain, dan sejumlah kemenangan atau kekalahan dalam berbagai situasi. (Febriyanto Pratama Putra, 2012).

## 1. Jenis – Jenis Game

Menurut Sulistyو (2010), ada beberapa jenis platform di dunia *game* yang selalu dipilih oleh pengguna *games* , yaitu:

- a. *Arcade games*, yaitu yang sering disebut ding-dong di Indonesia, biasanya berada di daerah / tempat khusus dan memiliki *box* atau mesin yang memang khusus di design untuk jenis *video games* tertentu dan tidak jarang bahkan memiliki fitur yang dapat membuat pemainnya lebih merasa masuk dan menikmati, seperti pistol, kursi khusus, sensor gerakan, sensor injakkan dan stir mobil (beserta transmisinya tentunya).
- b. *PC Games*, yaitu *video game* yang dimainkan menggunakan *Personal Computers*.
- c. *Console games*, yaitu *video games* yang dimainkan menggunakan *console* tertentu, seperti *Playstation 2*, *Playstation 3*, *XBOX 360*, dan *Nintendo Wii*.
- d. *Handheld games*, yaitu yang dimainkan di *console* khusus *video game* yang dapat dibawa kemana-mana, contoh *Nintendo DS* dan *Sony PSP*.
- e. *Mobile games*, yaitu yang dapat dimainkan atau khusus untuk *mobile phone* atau *PDA*.

## 2. Genre Game

*Game* terbagi atas beberapa genre, diantaranya yaitu:

- a. *Action Shooting* (tembak–menembak): permainan pada genre ini menunjukkan aksi yang cukup memiliki konten kekerasan tinggi, dimana terdapat aksi tembak menembak, memukul, bisa juga tusuk -

tusukan, tergantung cerita dan tokoh di dalamnya. Pada permainan jenis ini, pemain memerlukan kecepatan dalam reflex serta kordinasi yang baik dalam memainkannya.

Contoh : PB (*Point Blank*), CS (*Counter Strike*) dan *Crysis*.

- b. *Fighting* (pertarungan). Ada yang mengelompokan permainan genre *fighting* di bagian Aksi, namun penulis berpendapat berbeda, permainan ini memang memerlukan kecepatan refleks dan koordinasi mata dan tangan, tetapi inti dari permainan ini adalah penguasaan pada jurus atau *special action* (hafal caranya dan lancar mengeksekusinya), pengenalan karakter.
- c. *Adventure* (Petualangan). Permainan genre ini merupakan permainan yang melakukan penjelajahan seperti memanjat, menelusuri hutan, meloncati tebing yang terpisah jurang, berayun dari pohon ke pohon lainnya, bergulat melawan tanaman atau pun hewan liar demi mencari *clue* atau petunjuk menuju rintangan berikutnya. Adapun yang berpetualang diantara jalan - jalan perkotaan sekedar mencari tongkat kayu ataupun sabuk untuk membuat alat untuk misi berikutnya, itulah beberapa dari banyak hal yang karakter pemain harus lakukan dan lalui dalam permainan jenis ini.

Contoh : *Kings Quest*, dan *Space Quest*.

- d. *Strategy* (strategi). Video game strategi biasanya memberikan pemain atas kendali tidak hanya satu orang tapi minimal sekelompok orang dengan berbagai jenis tipe kemampuan, sampai kendaraan, bahkan

hingga pembangunan berbagai bangunan, pabrik dan pusat pelatihan tempur, tergantung dari tema ceritanya. Kebanyakan *game* strategi adalah *game* perang.

Contoh : Warcraft, *Red Alert*.

- e. *Simulation* (Simulasi). Permainan jenis ini seringkali menggambarkan kehidupan dunia nyata dan memperhatikan dengan detail berbagai faktor. Dari mencari makan hingga pekerjaan, membangun tempat tinggal hingga kota, mengatur pajak penghasilan dan dana kota. Permainan *genre* ini selayaknya hidup dari awal lahir yang tidak memiliki apa-apa hingga menjadi konglomerat penguasa bisnis dan lain sebagainya. Ada juga seperti melakukan eksperimen percobaan antara gen A terhadap gen lainya hingga mendapatkan hasil kloning yang unik. Pada permainan jenis ini membuat pemain harus berpikir dalam mendirikan, membangun dan mengatasi masalah dengan menggunakan dana yang terbatas.
- f. *Puzzle* (teka-teki). Permainan jenis ini sesuai dengan namanya mengenai pemecahan teka-teki, baik itu menyusun balok, menyamakan warna, menyamakan bentuk, memecahkan perhitungan matematika, menggeser, menarik dan mendorong kotak ke tempat yang seharusnya. Sering pula permainan jenis ini merupakan unsur dalam permainan *genre* petualangan maupun edukasi.

Contoh: Tetris, *Bubble Party*.

- g. *Sport game* (Olahraga). *Game* ini merupakan adaptasi dari kehidupan nyata, pemain game jenis sport membutuhkan kelincahan dan juga strategi dalam memainkannya. *Game* ini berupa kompetisi antara dua pemain atau lebih, dimana pemain dapat melakukan secara individual atau tim. Contoh *game* tipe ini antara lain, PES (*pro evolution soccer*), Mario Kart, tenis.
- h. *RPG (Role Playing Game)*. Permainan ini sesuai dengan terjemahannya, bermain peran, memiliki penekanan pada tokoh/peran perwakilan pemain di dalam *game*, yang biasanya adalah tokoh utamanya, dimana seiring kita memainkannya, karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan pemain dalam berbagai parameter yang biasanya ditentukan dengan naiknya level, baik dari status kepintaran, kecepatan dan kekuatan karakter, senjata yang semakin sakti, ataupun jumlah teman maupun makhluk peliharaan.

Contoh: *Final Fantasy, Dungeon Hunter, Ragnarok*

- i. *Education* (edukasi). *Game* edukasi merupakan paket software yang menciptakan kemampuan pada lingkungan *game* yang diberikan sebagai alat bantu untuk memotivasi atau membantu siswa untuk melalui prosedur *game* secara teliti untuk mengembangkan kemampuannya. *Developer* yang membuatnya, harus memperhitungkan berbagai hal agar *game* ini benar- benar dapat mendidik, menambah pengetahuan dan meningkatkan keterampilan

yang memainkannya. Target segmentasi pemain harus pula disesuaikan dengan tingkat kesulitan dan *design* visual ataupun animasinya.

### 3. Unsur-Unsur Game

*Game* terdiri dari berbagai unsur dasar yang dipadukan sehingga menciptakan suatu pengalaman yang menarik, adapun komponen-komponen *game* antara lain:

#### a. Fitur

Fitur merupakan suatu hal yang menggambarkan *game* kedalam bentuk bentuk yang dapat dilihat maupun dirasakan.

#### b. *Gameplay*

*Gameplay* merupakan cara kerja suatu *game*, dimana fitur-fitur yang ada akan membentuk suatu *gameplay*.

#### c. *Interface*

*Interface* merupakan semua bentuk tampilan yang ada pada sebuah *game*. *Interface* yang baik akan membuat pemainnya betah dan tidak bosan dalam memainkan *game*.

#### d. Aturan (Rules)

*Rules* merupakan sekumpulan aturan yang ada dalam memainkan sebuah *game*.

#### e. *Desain Level*

Desain *level* merupakan tingkat kesulitan yang menggambarkan jalan cerita pada sebuah *game*.

### **C. Sejarah Perjuangan Sultan Hasanuddin**

Selanjutnya kerajaan Makassar mencapai puncak kebesarannya pada masa pemerintahan Sultan Hasanudin 1653-1669 M. Pada masa pemerintahannya, Makassar berhasil memperluas wilayah kekuasaannya yaitu dengan menguasai daerah-daerah yang subur, serta daerah-daerah yang dapat menunjang keperluan perdagangan Makassar, perluasan daerah Makassar tersebut sampai ke Nusa Tenggara Barat.

Sultan Hasanuddin dalam Syair Perang Mengkasar adalah tokoh pahlawan Kerajaan Gowa yang dengan gigih melawan kolonialisme Belanda sebagai pihak yang akan menguasai Makassar. Dalam syair ini, Sultan Hasanuddin digambarkan sebagai tokoh pemberani, pandai dalam strategi perang, dan bijaksana. Sultan Hasanuddin ditampilkan sebagai pahlawan yang memimpin kerajaan dengan penuh tanggung jawab dan adil. Oleh sebab itu, tokoh ini dipuja rakyatnya. Gambaran Sultan Hasanuddin dimulai dari pembukaan Syair Perang Mengkasar. Pada bagian awal dipanjatkan puji-pujian untuk Sultan Gowa. Dalam Syair Perang Mengkasar bait 14–19.

Dalam melakukan perlawanan Sultan Hasanuddin terdapat faktor-faktor yang mempengaruhi adanya peristiwa itu terjadi yaitu dalam faktor ekonomi dari dalam maupun dari luar, faktor dari dalam faktor dari dalam, yaitu tumbuhnya Kerajaan Gowa-Tallo menjadi kerajaan yang menghimpun, dan melindungi

wilayah orang Makassar di sepanjang pesisir selatan jazirah Selatan Sulawesi. Pertumbuhan itu didorong pula oleh adanya ancaman dari arah daratan Sulawesi Selatan dengan berkembangnya beberapa kerajaan Bugis di pedalaman, seperti Soppeng, Wajo, dan Bone yang mulai menanam pengaruhnya di daerah pantai, atau pesisir sebelah utara, dan sepanjang Teluk Bone. Persaingan untuk memperebutkan pengaruh antara kerajaan Bugis dan Makassar itulah yang menyebabkan makin ditingkatkannya usaha oleh Kerajaan untuk mengadakan tempat konsolidasi kekuatan, dengan membangun benteng pertahanan, dan pangkalan militer untuk menyerang dan memperluas daerah pengaruh Kerajaan mereka dan adapun faktor dari luar yaitu kedatangan bangsa Eropa ke Nusantara untuk berniaga. Ditemukannya jalan menuju pulau rempah-rempah, dan timbulnya persaingan antara bangsa yang melakukan perniagaan itu. Karena keinginan mereka untuk memperoleh keuntungan yang sebanyak-banyaknya, maka mereka pun memerlukan pangkalan niaga yang dapat dijadikan tempat, dan mata rantai dalam perjalanan pulang ke negeri asal mereka. Terjadilah penaklukan oleh Bangsa Barat itu, mulai di pantai India, Malaka pulau Jawa, Maluku, dan sebagainya. Karena ditaklukkannya Malaka oleh Portugis, maka perjalanan niaga bangsa Portugis ke pulau rempah-rempah lebih lancar tanpa gangguan Kerajaan Nusantara seperti Aceh di bagian barat, dan niagawan-niagawan Keling, dan juga Makassar yang memiliki potensi dalam hal perniagaan pun tak luput dari minat bangsa Asing, sehingga menjadikan Kerajaan ini ramai dan banyak dikenal di Dunia.

Bukan hanya faktor ekonomi yang memicu adanya perlawanan Sultan Hasauddin, tetapi faktor politik sangat berpengaruh terhadap adanya sejarah perjuangan Sultan Hasanuddin Pada bulan Maret 1602, perseorangan dagang di negeri Belanda bergabung membentuk perserikata Maskapai Hindia Timur VOC *Vereenigde Oost- Indische Compagnie*. Salah satu alasan dari pembentukan organisasi ini adalah, untuk memperkuat posisi Belanda dalam persaingan dagang melawan Spanyol, Inggris, Portugis. Alasan lainnya dalam pembentukan VOC adalah, untuk melakukan monopoli atas hasil perkebunan pada masa itu, contoh adalah beras dari Mataram, rempah-rempah, dan hasil-hasil perkebunan lainnya.

Direktur-direktornya memutuskan untuk menempatkan pemimpin tertinggi di Asia bukan di negara aslinya yaitu Belanda. dan Raad van Indië, Dewan Hindia. Keduanya membentuk suatu pemerintahan yang dinamakan *Hoge Regering* (Pemerintahan Agung), lembaga ini menjadi pusat otoritas VOC di Asia, dan berkedudukan di Batavia. Anggota Dewan Hindia memiliki tugas membantu Gubernur Jenderal pada semua masalah seperti manajemen umum perdagangan, perang, pemerintahan, serta penegakkan keadilan di semua masalah publik dan kriminal.

Ketika Sultan Hasunuddin Memimpin Kerajaan Gowa dia tetap menjalankan, dan melanjutkan kebijaksanaan yang diambil serta ditempuh oleh almarhum Sultan Alaudin dan almarhum ayahnya Sultan Malikussaid. Kerajaan Gowa tetap tidak mau mengakui hak monopoli perdagangan VOC, yang hendak dipaksakannya di Indonesia bagian Timur. Oleh karena itu hubungan antara Kerajaan Gowa Dan, VOC makin tegang, dan memburuk. Kemudian VOC

memang selalu berusaha untuk menyingkirkan, dan menghancurkan Kerajaan Gowa, karena VOC menganggap Kerajaan Gowa adalah musuh yang sangat berbahaya, dan penghalang yang besar bagi VOC untuk melaksanakan monopoli perdagangannya, terutama di Indonesia bagian timur. Ancaman dari pihak VOC yang makin merajalela di Indonesia bagian timur yang memaksa rakyat, dan Kerajaan Gowa untuk memilih seseorang pemimpin yang kuat, berwatak dan berwibawa.

Pada bulan april 1655 armada Gowa yang dipimpin langsung Sultan Hasanuddin menyerang orang-orang Belanda yang telah menduduki kerajaan Buton. Dalam menghadapi armada Gowa, Belanda menghasut Sultan Buton agar mempertahankan negerinya sendiri dan Belanda akan membantu sedapat mungkin. Akan tetapi karena serangan armada Gowa sangat hebat, maka pimpinan pasukan Belanda tidak dapat bertahan lagi. Setelah menghancurkan perlawanan Belanda di Buton, maka Sultan Hasanuddin bersama armada Gowa meninggalkan perairan Buton dan kembali ke Gowa.

Ketika Laksamana de Vlamingh tiba di Buton, dan yang didapat hanyalah puing-puing pertahanan Belanda yang ditemukan di Buton. Setelah melihat, dan mengalami kenyataan bahwa perang dengan Kerajaan Gowa telah menelan banyak korban, dan biaya yang sangat banyak serta merugikan perdagangan Belanda akibat peperangan itu. Belanda berusaha keras mengadakan hubungan perdamaian dengan Kerajaan Gowa. Maka pada tanggal 23 oktober 1655, para pembesar VOC di Batavia memutuskan untuk berusaha mengadakan perjanjian perdamaian dengan Kerajaan Gowa. Belanda menunjuk Willem Van Der Beeck yang dibantu seorang

bangsa Armenia yang bernama Khoja Sulaeman sebagai utusan VOC ke Somba Opu. Mereka menghadap Sultan Hasanuddin, dan menyampaikan pesan Jendral Maetsyuiker.

Pihak Kerajaan Gowa meminta waktu untuk memikirkan tuntutan itu, hingga pada akhirnya tanggal 2 Februari 1656, Sultan Hasanuddin menolak tuntutan itu yang didukung Karaeng Karunrung, Karaeng Galesong, dan Karaeng Bontomarannu. Tuntutan itu dianggap merugikan Kerajaan Gowa. Mereka juga tahu Belanda pasti mengingkari perjanjian tersebut, Bahkan mereka mengumumkan perang terbuka sehingga untuk periode selanjutnya akan terus-menerus terjadi bentrokan.

Pada tanggal 10 Februari 1659 Dewan Hindia mengadakan perundingan bagaimana hendaknya sikap VOC terhadap Kerajaan Gowa. VOC menghendaki agar Kerajaan Gowa jangan mencampuri rakyat di daerah yang dikuasainya, seperti Sultan Mandarsyah dari Ternate. Akan tetapi apa yang dikehendaki VOC tidak tercapai, sebab Kerajaan Gowa ingin melindungi rakyat dari Belanda. Pada tanggal 27 April 1659 diadakan perundingan antara VOC yang diwakili William Bastingh, dan Kerajaan Gowa yang diwakili Karaeng Popo atas nama raja Gowa Sultan Hasanuddin, pertemuan ini membahas tentang larangan Kerajaan Gowa berdagang di pulau, dan pelabuhan Maluku yang dikuasai VOC. Namun tidak disetujui Kerajaan Gowa, pihak Gowa menganggap hal itu bertentangan dengan ajaran Tuhan, sebab Tuhan menciptakan bumi agar semua orang dapat hidup, dan menikmati manfaatnya.

#### D. Role Playing Game

Singkatan dari *Role Playing Game*, di mana permainannya dilakukan seolah-olah menjadi peran dalam sebuah cerita dan sejarahnya. Awalnya, RPG merupakan *game* yang berbasis teks, tetapi seiring perkembangan teknologi permainan ini menggunakan grafik dan visual yang memanjakan mata.

Biasanya, *game* RPG menyediakan sejumlah karakter yang bisa digunakan sebagai diri sendiri tetapi dilakukan di dalam *game*. Saat ini, *game* RPG sudah dimodifikasi dan dikombinasikan sehingga menjadi mudah dilakukan dimanapun. Salah satu modifikasinya adalah cross-platform, yang berarti bisa digunakan dalam perangkat apapun.

Permainan dengan genre ini, umumnya menyediakan cerita atau sejarah fiksi yang harus diselesaikan melalui tugas-tugas atau misi. Dengan menyelesaikan misi atau tugas tersebut, para pemain mendapatkan hadiah atau membuka wilayah yang terkunci sebelumnya.

#### E. Game Maker Studio

Menurut [www.yoyogames.com](http://www.yoyogames.com), *Game maker Studio* adalah sebuah aplikasi *game* editor yang dibuat pertama kali oleh Mark Overmars yang kemudian dilanjutkan pengembangannya oleh Yoyo Games. Aplikasi ini digunakan untuk membuat *game* 2D 3D atau *multiplayer*.

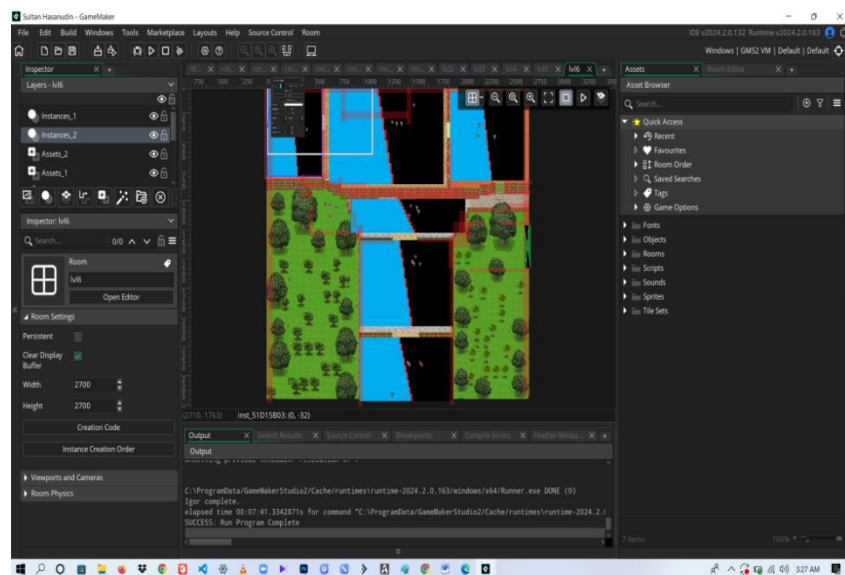
##### 1. *Game Maker Language*(Bahasa Pemrograman)

*Game Maker Language* (GML) adalah *primer* diartikan bahasa *scripting* yang digunakan dalam *GameMaker*, yang biasanya secara signifikan lebih lambat

daripada yang dihimpun bahasa seperti C++ atau *Delphi*. Hal ini digunakan untuk lebih meningkatkan dan mengendalikan desain permainan melalui lebih konvensional pemrograman, sebagai lawan untuk *drag and drop* sistem.

## 2. Keuntungan atau kelebihan menggunakan *game maker*

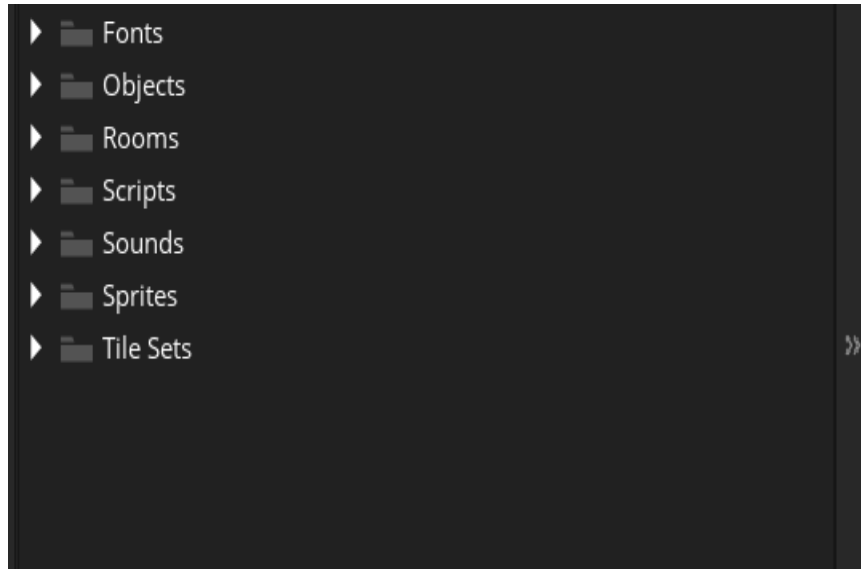
- a. Sangat mudah digunakan karena untuk membuat sebuah permainan cukup menggunakan *drag-and-drop* sederhana, dan tanpa menggunakan *coding* yang banyak.
- b. Mudah dipelajari karena banyak terdapat tutorialnya
- c. *Free* atau gratis jadi tidak perlu mengeluarkan biaya Memulai menjalankan *Game Maker* pada saat akan membuat sebuah *game*, maka langkah pertama yang dilakukan adalah menjalankan *game maker* yang sudah diinstal sebelumnya. Langkahnya adalah klik *start* → menu program → *game maker*. Ketika *game maker* dipilih maka akan muncul tampilan seperti Gambar 2.3 berikut ini:



**Gambar 2.1** Tampilan awal *Game Maker Studio*

### 3. Bagian – Bagian Game Maker Studio

Didalam *game maker studio* terdapat beberapa *resource* yang dapat digunakan untuk membuat sebuah *game*.



**Gambar 2.2** Bagian – bagian dari *game maker studio*

- a. *Font* digunakan untuk menambahkan *font* yang akan di gunakan dalam *game*  
*font* biasanya digunakan dengan fungsi *text drawing*.
- b. Obyek merupakan hal terpenting dalam *Game*. Obyek merupakan benda hidup yang bertindak dalam permainan. Sebuah obyek diberikan pergerakan melalui *event* yang harus direaksi oleh *game*, dan bagaimana seharusnya bereaksi. Sebuah obyek biasanya diwakili oleh sebuah *sprite*. Kejadian yang dapat dilakuakn oleh obyek misalnya kejadian input (*keyboard*, *joystick*, *mouse*), peristiwa tabrakan (tabrakan dengan objek lain), alarm kejadian (*timer*) dan beberapa jenis kejadian lainnya. Ketika 13 sebuah kejadian terjadi, sejumlah tindakan akan diambil. Tindakan ini dipilih dari daftar tindakan. Tindakan bisa merupakan gerakan, penciptaan objek lain,

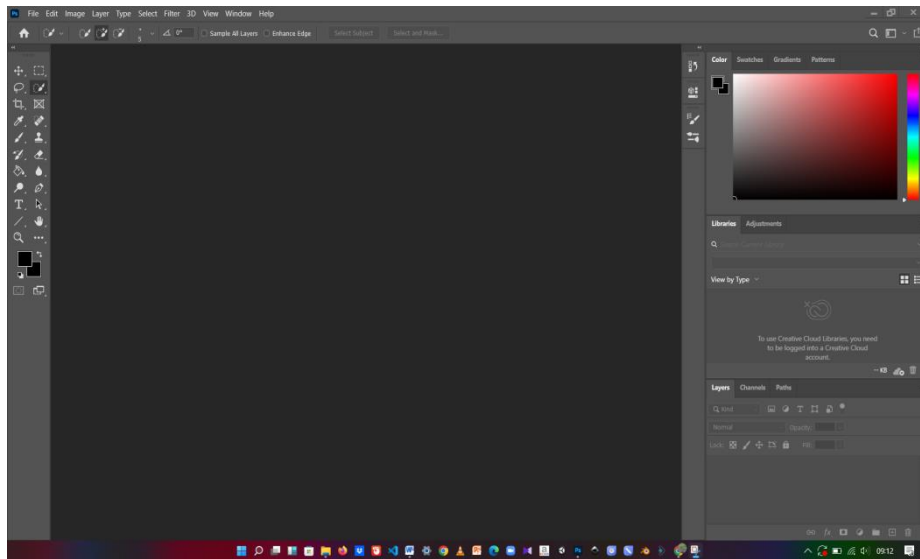
memainkan suara, mengubah *sprite*, dll. Ini merupakan benda yang "menyusun" permainan

- c. *Room* adalah ruangan dimana *game* berjalan dan juga sebagai tempat untuk suatu obyek untuk di letakan atau tempat suatu obyek untuk bergerak. Dan dapat membuat banyak *room* sehingga terdapat rintangan yang berbeda pada setiap *game* pada masing – masing roomnya. *Room* dapat dihubungkan dengan *room* lain sehingga suatu obyek dapat berpindah – pindah.
- d. *Script* dalam *game maker* terdapat dua cara membuat program, yaitu dengan cara *drag-and-drop*, dan cara membuat script program. *Script* merupakan koleksi dari fungsi-fungsi statement program yang dapat dijalankan dari berbagai obyek, atau berbagai *script* kode.
- e. *Sound*, Dengan *sound* dapat menyertakan suara latar atau efek suara sehingga *game* terasa lebih hidup. Dalam *game maker* sudah terdapat beberapa efek usara yang telah di sediakan, contoh penggunaanya adalah: Klik Menu *Resources > Create Sound > Load Sound*.
- f. *Sprite* adalah gambar dari semua objek dalam *game*. *Sprite* bisa terdiri dari satu gambar saja atau lebih dari satu gambar sehingga terlihat seperti gerakan animasi.

## **F. Adobe Photoshop**

*Adobe Photoshop* Penulis menggunakan *adobe photoshop* untuk mendesain sebuah gambar yang digunakan untuk membantu penelitian ini. Menurut Munir “*Adobe Photoshop* merupakan perangkat lunak aplikasi untuk mendesain gambar, atau disebut *photo design and production tools*”. Menurut Kuryanti “*adobe*

*photoshop* perangkat lunak editor citra buatan *adobe system* yang dikhususkan untuk pengeditan gambar dan pembuatan efek” Dapat beberapa pendapat ahli diatas dapat disimpulkan bahwa *adobe photoshop* suatu aplikasi perangkat lunak yang digunakan untuk mengedit suatu gambar, yang biasa digunakan oleh *photografer* digital dan tentunya untuk mendapatkan karakter pada *photoshop* ini.



**Gambar 2.3** Tampilan Awal Adobe Photoshop

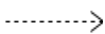
### **G. UML (Unified Modeling Language)**

Menurut Nugroho, UML (*Unified Modeling Language*) adalah Metodologi kolaborasi antara metode-metode Booch, OMT (*Object Modeling Technique*), serta OOSE (*Object Oriented Software Engineering*) dan beberapa metoda lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa “pemrograman berorientasi objek” (OOP).

Dalam UML sendiri terdapat beberapa diagram yaitu:

a. *Use Case Diagram*

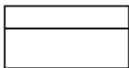
**Tabel 2. 1** Simbol-Simbol *Use Case Diagram*

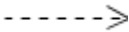
| NO | GAMBAR                                                                              | NAMA                  | KETERANGAN                                                                                                                                                                                 |
|----|-------------------------------------------------------------------------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  |    | <i>Actor</i>          | Mengspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .                                                                                       |
| 2  |  | <i>Dependency</i>     | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ). |
| 3  |  | <i>Generalization</i> | Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).                                        |
| 4  |  | <i>Include</i>        | Mengspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .                                                                                                                  |
| 5  |  | <i>Extend</i>         | Mengspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.                                                           |
| 6  |  | <i>Association</i>    | Apa yang menghubungkan antara objek satu dengan objek lainnya.                                                                                                                             |
| 7  |  | <i>System</i>         | Menspesifikasikan paket yang menampilkan sistem secara terbatas.                                                                                                                           |

| NO | GAMBAR                                                                            | NAMA                 | KETERANGAN                                                                                                                                         |
|----|-----------------------------------------------------------------------------------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| 8  |  | <i>Use Case</i>      | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.                               |
| 9  |  | <i>Collaboration</i> | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemen nya (sinergi). |
| 10 |  | <i>Note</i>          | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.                                                     |

b. *Class Diagram*

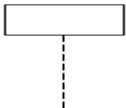
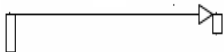
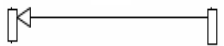
**Tabel 2. 2** Simbol-Simbol *Class Diagram*

| NO | GAMBAR                                                                              | NAMA                    | KETERANGAN                                                                                                                                          |
|----|-------------------------------------------------------------------------------------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  |  | <i>Generalization</i>   | Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ). |
| 2  |  | <i>Nary Association</i> | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.                                                                                         |
| 3  |  | <i>Class</i>            | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.                                                                             |
| 4  |  | <i>Collaboration</i>    | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.                                |

| NO | GAMBAR                                                                            | NAMA               | KETERANGAN                                                                                                                                                         |
|----|-----------------------------------------------------------------------------------|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5  |  | <i>Realization</i> | Operasi yang benar-benar dilakukan oleh suatu objek.                                                                                                               |
| 6  |  | <i>Dependency</i>  | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |

c. Sequence Diagram

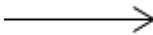
**Tabel 2. 3** Sequence Diagram

| NO | GAMBAR                                                                              | NAMA            | KETERANGAN                                                                                              |
|----|-------------------------------------------------------------------------------------|-----------------|---------------------------------------------------------------------------------------------------------|
| 1  |  | <i>LifeLine</i> | Objek <i>entity</i> , antarmuka yang saling berinteraksi.                                               |
| 2  |  | <i>Message</i>  | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi. |
| 3  |  | <i>Message</i>  | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi. |

d. State Chart Diagram

**Tabel 2. 4** State chart Diagram

| NO | GAMBAR                                                                              | NAMA         | KETERANGAN                                                                              |
|----|-------------------------------------------------------------------------------------|--------------|-----------------------------------------------------------------------------------------|
| 1  |  | <i>State</i> | Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek. |

| NO | GAMBAR                                                                            | NAMA                        | KETERANGAN                                                                                                 |
|----|-----------------------------------------------------------------------------------|-----------------------------|------------------------------------------------------------------------------------------------------------|
| 2  |  | <i>Initial Pseudo State</i> | Bagaimana objek dibentuk atau diawali.                                                                     |
| 3  |  | <i>Final State</i>          | Bagaimana objek dibentuk dan dihancurkan.                                                                  |
| 4  |  | <i>Transition</i>           | Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya. |
| 5  |  | <i>Association</i>          | Apa yang menghubungkan antara objek satu dengan objek lainnya.                                             |
| 6  |  | <i>Node</i>                 | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.             |

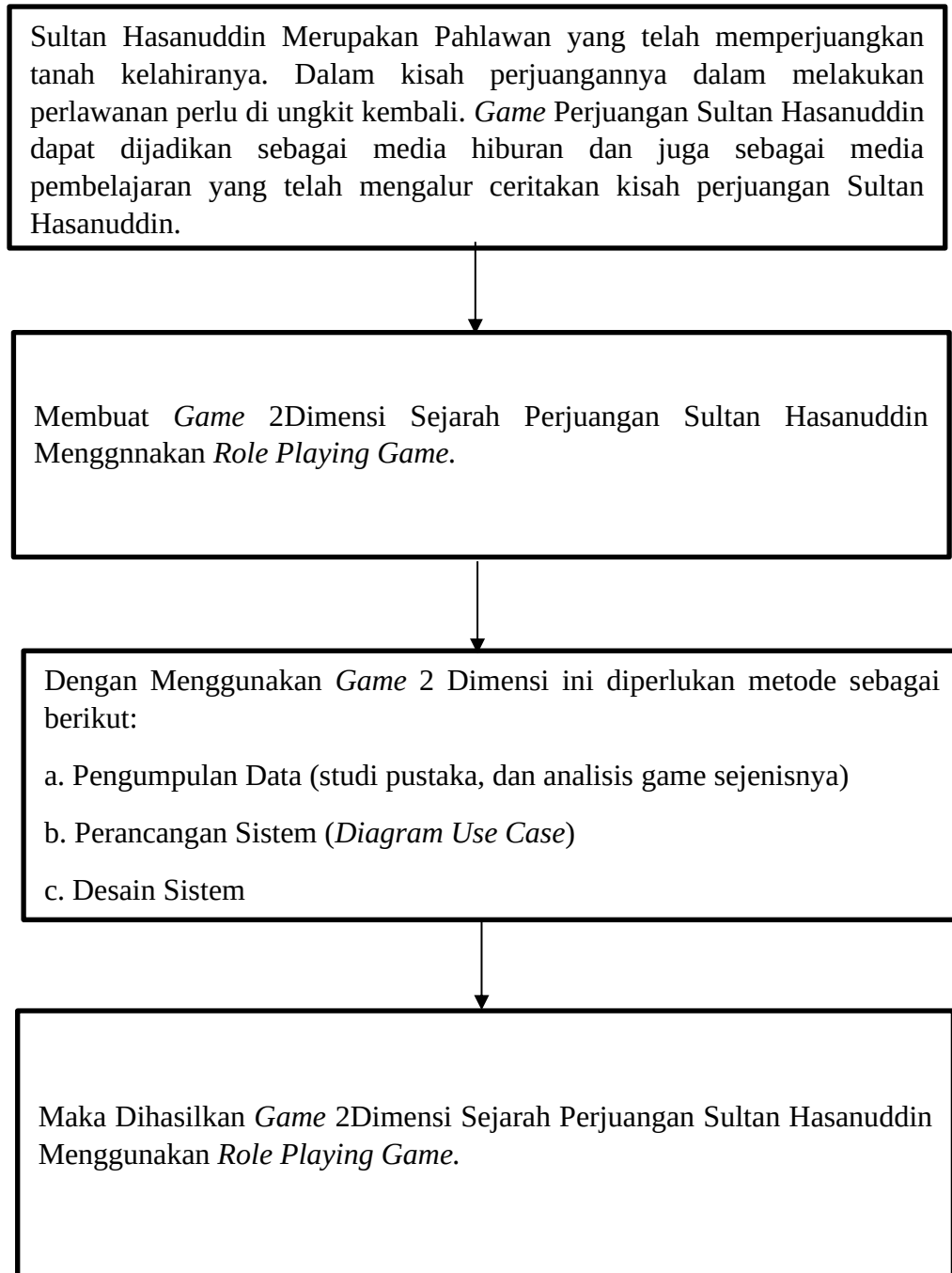
e. *Activity Diagram*

**Tabel 2. 4** Simbol-Simbol *Activity Diagram*

| NO | GAMBAR                                                                              | NAMA                       | KETERANGAN                                                                                 |
|----|-------------------------------------------------------------------------------------|----------------------------|--------------------------------------------------------------------------------------------|
| 1  |  | <i>Activity</i>            | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain. |
| 2  |  | <i>Action</i>              | State dari sistem yang mencerminkan eksekusi dari suatu aksi.                              |
| 3  |  | <i>Initial Node</i>        | Bagaimana objek dibentuk atau diawali.                                                     |
| 4  |  | <i>Activity Final Node</i> | Bagaimana objek dibentuk dan dihancurkan.                                                  |
| 5  |  | <i>Fork Node</i>           | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.                      |

## H. Kerangka Pikir

Adapun Kerangka Pikir dari penelitian ini sebagai berikut:



## **BAB III**

### **METODE PENELITIAN**

#### **A. Jenis Penelitian**

Penulis melakukan penelitian studi literatur dengan mempelajari buku-buku dan referensi dari website atau situs-situs dan jurnal yang berhubungan dengan penelitian yang dilakukan.

#### **B. Waktu Penelitian**

Waktu yang digunakan dalam pelaksanaan penelitian ini adalah  $\pm$  5 bulan yaitu pada bulan September 2023 hingga Maret 2024.

#### **C. Metode Pengumpulan Data**

##### **1. Studi Pustaka**

Mencari teori dari informasi yang berhubungan dengan topik yang akan dibuat. Pencarian teori dan informasi akan dicari melalui buku, internet, dan hasil penelitian lainnya.

##### **2. Analisis *Game* Sejenis**

Mencari Informasi tentang aplikasi atau game dengan topik dan membandingkannya.

#### **D. Alat Dan Bahan Penelitian**

##### **1. Perangkat Keras (*Hardware*)**

Perangkat keras (*Hardware*) yang digunakan untuk membangun aplikasi ini adalah Laptop Asus dengan spesifikasi sebagai berikut:

- Laptop : Aspire A515-45
- Processor : AMD Ryzen 5 5500U with Radeon Graphics 2.10 GHz

- RAM : 8GB
- SSD : 512GB

## 2. Perangkat Lunak (Software)

Perangkat lunak (*Software*) yang digunakan untuk membangun aplikasi ini adalah sebagai berikut:

- *Windows 11*
- *Game Maker Studio*
- *Adobe Photoshop*
- *Google Chrome*

## E. Objek Penelitian

Objek penelitian yang dilakukan adalah pada pembuatan game ini untuk semua umur yang dan sudah bisa memainkan game pada dekstop yang berlatar belakang Sejarah Perjuangan Sultan Hasanuddin.

## F. Tahapan Penelitian

Tahapan yang dimaksud dalam penelitian ini ada beberapa tahapan yaitu persiapan penelitian, pengumpulan data, analisis data, pembuatan, pengujian dan implementasi. Adapun uraian dari tahapan tersebut adalah sebagai berikut:

### 1. Persiapan penelitian

Pada tahapan ini peneliti melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel-artikel tentang topic penelitian serta software yang digunakan selama penelitian.

## 2. Pengumpulan data

Pada tahapan ini peneliti melakukan apa yang disebut dengan kajian pustaka, yaitu mempelajari buku-buku referensi dan hasil penelitian sejenis sebelumnya yang pernah dilakukan oleh orang lain. Tujuannya adalah untuk mendapatkan landasan teori mengenai masalah yang akan diteliti. Teori merupakan pijakan bagi peneliti untuk memahami persoalan yang diteliti dengan benar dan sesuai dengan kerangka berpikir ilmiah.

## 3. Analisis

Pada tahap analisis, peneliti melakukan analisa terhadap sistem yang di terapkan sekarang berdasarkan kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

## 4. Pembuatan

Peneliti kemudian membuat game yang ingin dibuat berdasarkan alternatif pemecahan masalah yaitu “Game 2dimensi Sejarah Perjuangan Sultan Hasanuddin Menggunakan Role Playing Game Berbasis Dekstop.

## 5. Pengujian

Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

## 6. Implementasi

Setelah pada pembuatan tidak terdapat kekurangan maka aplikasi siap untuk di gunakan oleh user.

### **G. Metode Pengujian**

Metode pengujian yang penulis gunakan yaitu:

#### 1. *White box*

*White box testing* adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan *white box testing* merupakan petunjuk untuk mendapatkan program yang benar secara 100%.

#### 2. *Black box*

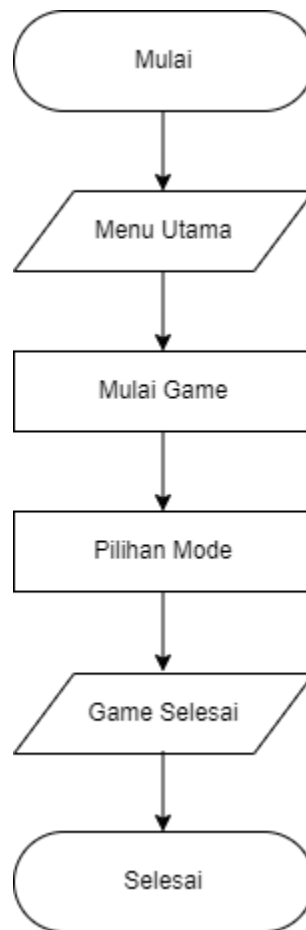
*Black box testing* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tahu ada apa dibalik bungkus hitamnya. Sama seperti pengujian *black box*, mengevaluasi hanya dari tampilan luarnya (interface-nya), 30 fungsionalitasnya tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detailnya (hanya mengetahui *input* dan *output*).

## H. Desain Sistem

Pembuatan *Game* RPG disajikan menggunakan *Unified Modelling Language* (UML). *Flowchart* dan *Use Case* diagram yang digunakan untuk menggambarkan keadaan lingkungan yang ada didalam sistem.

### 1. Sistem yang diusulkan

Adapun sistem yang diusulkan menggunakan *flowchart* saat ini yaitu:



**Gambar 3.1** Flowchart sistem yang diusulkan

- a. Pengguna Akan memilih menu utama yaitu mulai *game*, lanjutkan, petunjuk dan keluar.
- b. Pengguna akan memilih mode pada mulai *game* yaitu mode alur cerita dan mode *survival*.
- c. Pengguna menyelesaikan *game* setelah melewati mode alur cerita dan mode *survival*.

## **BAB IV**

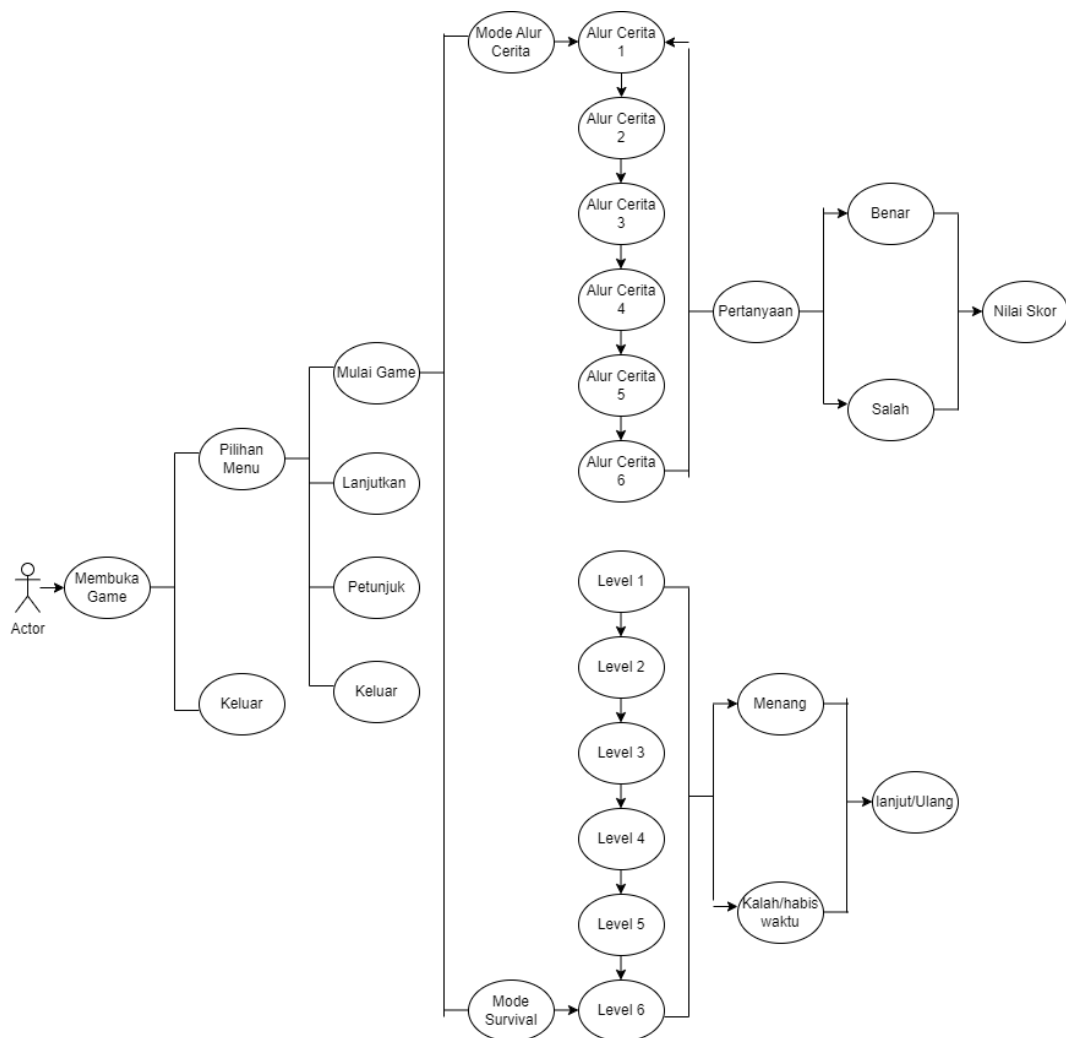
### **HASIL DAN PEMBAHASAN**

#### **A. Analisis Aliran Data Dengan UML**

Proses pengembangan selanjutnya adalah proses desain pengembangan perangkat lunak. Desain sistem perangkat lunak yang dibuat nantinya akan dijadikan sebagai acuan pengembang dalam penulisan kode. Desain sistem perangkat lunak ini harus sesuai karakteristik perangkat lunak yang akan dibuat serta mampu mempermudah dan memperjelas pengembang dalam proses pembuatan perangkat lunak. Penelitian ini menggunakan desain sistem model *Unified Model Language* (UML) dikarenakan model ini paling sesuai digunakan Untuk mengembangkan sistem berinteraksi.

##### **1. *Use Case Diagram***

*Use case* menggambarkan bagaimana seorang pengguna berinteraksi dengan sistem dengan cara menentukan langkah - langkah yang diperlukan untuk mencapai suatu tujuan tertentu. Variasi-variasi dalam urutan langkah yang terjadi antara aktor dan sistem merupakan sebuah skenario. Diagram *use case* memberikan suatu gambaran besar tentang fungsionalitas yang diharapkan dari sebuah sistem dilihat dari sudut pandang pengguna.



**Gambar 4. 1** Use Case Diagram Sistem yang diusulkan

**Tabel 4.1** Deskripsi *Use Case*

| <b>Nama <i>Use Case</i></b> | <b>Deskripsi <i>Use Case</i></b>                                                                             |
|-----------------------------|--------------------------------------------------------------------------------------------------------------|
| Pilihan Menu                | Merupakan tampilan untuk memilih menu <i>game</i> antara mulai <i>game</i> , lanjutkan, Petunjuk, dan Keluar |
| Mulai Game                  | Merupakan awal proses memainkan permainan yang terdapat 2 mode yaitu alur cerita dan <i>survival</i>         |
| Lanjutan                    | Merupakan Lanjutan permainan yang telah disimpan sebelumnya                                                  |
| Petunjuk                    | Merupakan Petunjuk sebelum bermain dan informasi mengenai aturan permainan                                   |

Definisi *use case* pada game Sultan Hasanuddin adalah sebagai berikut:

a. Definisi Aktor

**Tabel 4.2** Deskripsi *Use Case*

| <b>NO</b> | <b>Aktor</b> | <b>Deskripsi</b>                                                                                                                                           |
|-----------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1         | Pengguna     | Pengguna merupakan aktor dari perangkat lunak. Pengguna dapat membuka <i>Game</i> , membuka Mulai <i>Game</i> , membuka petunjuk, dan keluar dari aplikasi |

b. Defenisi Use Case

**Tabel 4.3** Defenisi Usecase

| NO | UseCase                       | Deksripsi                                                                                                                                                    |
|----|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Membuka Aplikasi/Game         | Proses Awal ketika pengguna membuka aplikasi atau <i>game</i> akan ditampilkan dan kemudian akan muncul alur cerita sejarah dan ditampilkan halaman Menu     |
| 2  | Membuka Play                  | Merupakan menu untuk memulai permainan, ketika pengguna memilih <i>Play</i> maka akan menampilkan Alur cerita terlebih dahulu                                |
| 3  | Membuka Lanjutan              | Merupakan Data <i>Game</i> yang telah disimpan sebelumnya dalam permainan                                                                                    |
| 4  | Membuka Petunjuk              | Merupakan menu yang menampilkan petunjuk dan informasi ke pengguna terkait penggunaan <i>game</i>                                                            |
| 5  | Memainkan Game                | Proses ketika pengguna memainkan <i>game</i> Perjuangan Sultan Hasanuddin berdasarkan Alur cerita, diakhir permainan sistem pertanyaan dan skor              |
| 6  | Mode Alur Cerita dan survival | Merupakan 2 mode berdeda yang dimana mode alur cerita memilih unsur cerita dan untuk mode survival merupakan mode random untuk menghasilkan output pada game |
| 7  | Skor                          | Merupakan hasil dari menjawab pertanyaan                                                                                                                     |

Penjabaran pada tabel diatas akan didefenisikan lebih lanjut pada tabel skenario *usecase* berikut:

Nama *use-case*: Membuka Aplikasi/ *Game*

**Tabel 4.4** Skenario Membuka Aplikasi/*Game*

| Aksi Aktor                       | Reaksi Sistem                                              |
|----------------------------------|------------------------------------------------------------|
| Skenario Normal                  |                                                            |
| 1.Pengguna Aplikasi/ <i>Game</i> |                                                            |
|                                  | a. Menjalankan splash screen<br>b. Menuju ke menu aplikasi |

Nama *use-case*: Membuka Lanjutan

**Tabel 4.5** Skenario Membuka Lanjutan

| Aksi Aktor                  | Reaksi Sistem                                                   |
|-----------------------------|-----------------------------------------------------------------|
| Skenario Normal             |                                                                 |
| 1.Pengguna Membuka Lanjutan |                                                                 |
|                             | a. Menuju Ke halaman <i>game</i> yang telah disimpan sebelumnya |

Nama *use-case*: Membuka Petunjuk

**Tabel 4.6** Skenario Membuka Petunjuk

| Aksi Aktor                                         | Reaksi Sistem                                                                    |
|----------------------------------------------------|----------------------------------------------------------------------------------|
| Skenario Normal                                    |                                                                                  |
| 1.Pengguna Membuka petunjuk                        |                                                                                  |
|                                                    | a. menampilkan petunjuk dan informasi ke pengguna terkait penggunaan <i>game</i> |
| Skenario Alternatif                                |                                                                                  |
| 1.Pengguna Membuka Petunjuk saat mulai <i>game</i> | a. menampilkan petunjuk dan informasi ke pengguna terkait penggunaan <i>game</i> |

Nama *use-case*: Membuka Memainkan Game

**Tabel 4.7** Skenario Memainkan Game

| Aksi Aktor                                                                           | Reaksi Sistem                                                                                                                                     |
|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Skenario Normal                                                                      |                                                                                                                                                   |
| 1.Pengguna Memainkan <i>game</i>                                                     |                                                                                                                                                   |
|                                                                                      | a Proses ketika pengguna memainkan <i>game</i> Perjuangan Sultan Hasanuddin berdasarkan Alur cerita, diakhir permainan sistem pertanyaan dan skor |
| Skenario Alternatif                                                                  |                                                                                                                                                   |
| 1.Pengguna Menekan <i>Space</i> untuk melanjutkan saat intro alur cerita berlangsung | a. Menuju Halaman Pertempuran                                                                                                                     |

Nama *use-case*: Membuka Mode Alur cerita/Survival

**Tabel 4.8** Skenario Membuka Petunjuk

| Aksi Aktor                                     | Reaksi Sistem                                                         |
|------------------------------------------------|-----------------------------------------------------------------------|
| Skenario Normal                                |                                                                       |
| 1.Mode Alur Cerita/ Survival                   |                                                                       |
|                                                | a. Menampilkan dan Memulai Game dengan mode pilihan                   |
| Skenario Alternatif                            |                                                                       |
| 1.Pengguna Membuka Mode Alur Cerita / Survival | a. Memainkan Game Mode alur cerita<br>b. Memainkan Game Mode Survival |

Nama *use-case*: Menampilkan Skor

**Tabel 4.9** Skenario Menampilkan skor

| Aksi Aktor          | Reaksi Sistem                                                                         |
|---------------------|---------------------------------------------------------------------------------------|
| Skenario Normal     |                                                                                       |
| 1. Menampilkan Skor |                                                                                       |
|                     | a. Terlebih dahulu menjawab pertanyaan<br>b. Merupakan hasil dari menjawab pertanyaan |

Nama *use-case*: Keluar Aplikasi

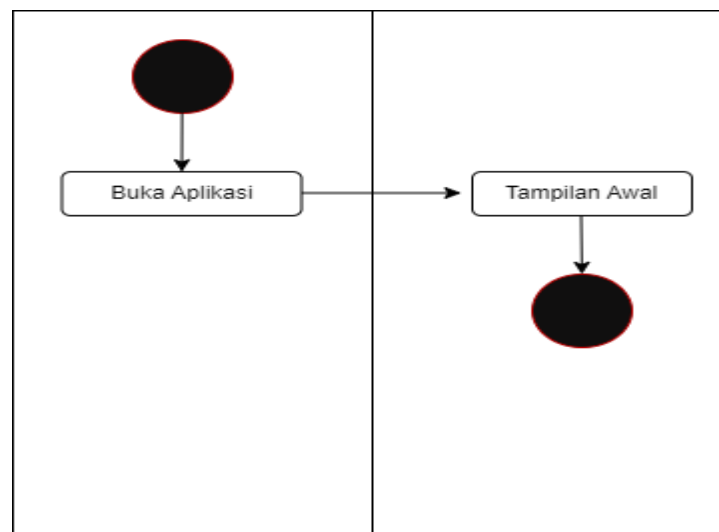
**Tabel 4.10** Skenario Keluar Aplikasi

| Aksi Aktor      | Reaksi Sistem                         |
|-----------------|---------------------------------------|
| Skenario Normal |                                       |
| 1. Keluar       |                                       |
|                 | a. Proses keluar dan menutup aplikasi |

## 2. Activity Diagram

Activity Diagram ini menjelaskan tentang aktivitas-aktivitas yang terjadi dalam sebuah aliran proses pada Game 2 Dimensi Sejarah Perjuangan Sultan Hasanuddin Menggunakan *Role Playing Game* Berbasis Dekstop.

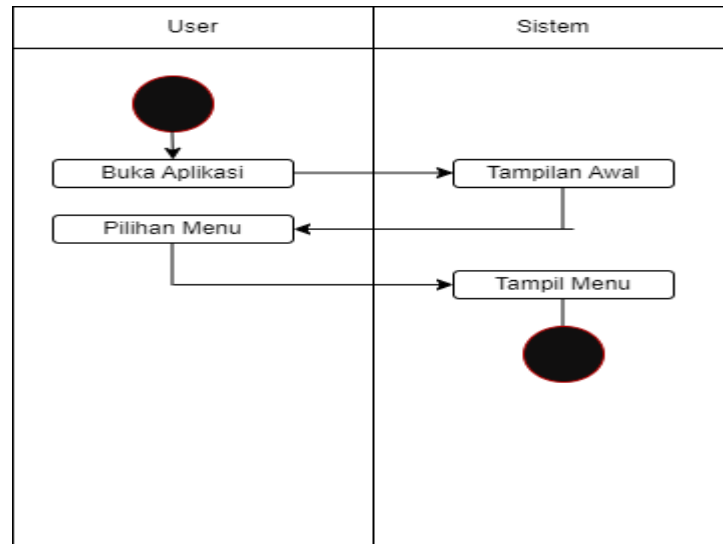
### a. Activity diagram Tampilan Awal



**Gambar 4.2** Activity diagram Tampilan Awal

Pada gambar 4.2 menjelaskan pada saat user membuka aplikasi. Pertama yang dilakukan user adalah membuka aplikasi kemudian sistem akan menampilkan tampilan awal.

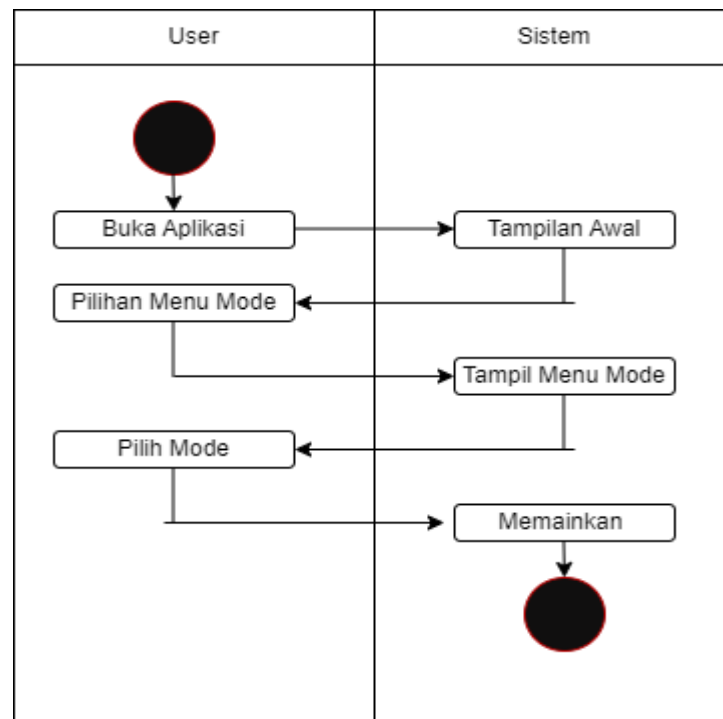
b. *Activity diagram* Tampilan Pilihan Menu



**Gambar 4.3** *Activity diagram* Tampilan Pilihan Menu

Pada gambar 4.3 menjelaskan bagaimana proses memilih menu. Langkah yang dilakukan pertama yaitu user membuka aplikasi, selanjutnya *system* menampilkan tampilan awal. Lalu user memilih tombol mulai lalu memilih menu.

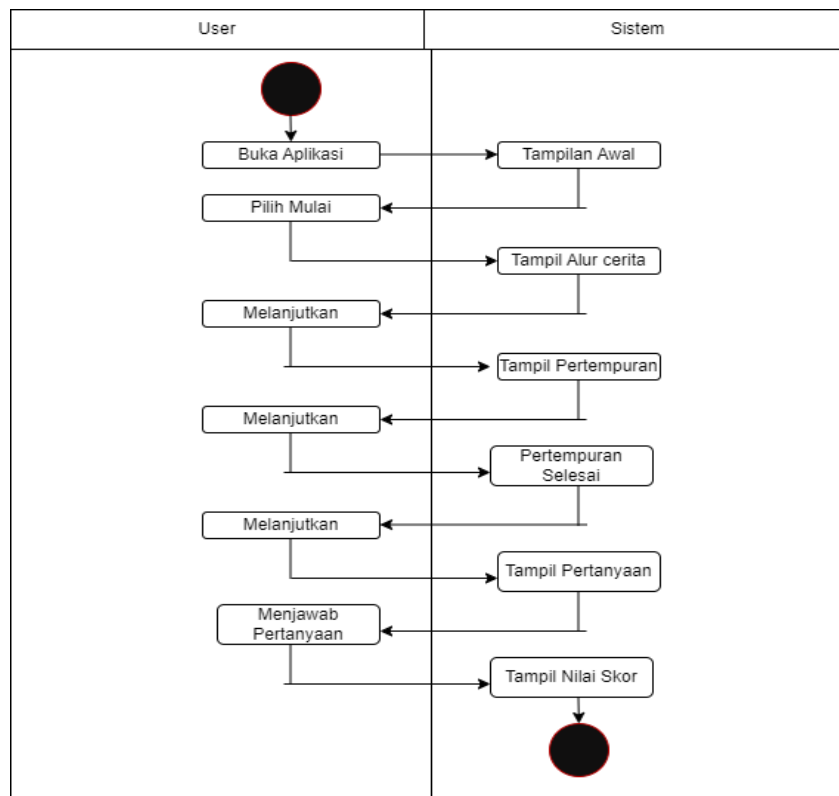
c. Activity diagram Tampilan Menu Mode



**Gambar 4.4** Activity diagram Tampilan Pilihan Menu Mode

Pada gambar 4.4 menjelaskan bagaimana proses memilih menu mode. Langkah yang dilakukan pertama yaitu user membuka aplikasi, selanjutnya system menampilkan tampilan awal. Lalu user memilih tombol menu mode dan memilih menu mode.

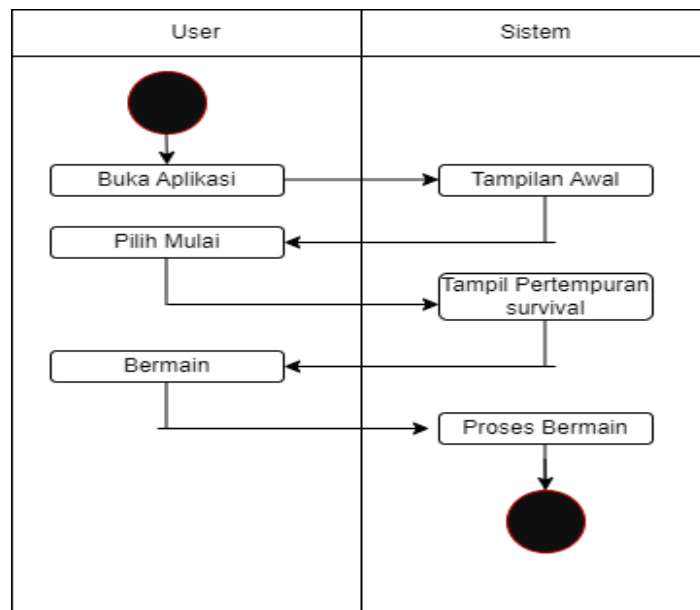
d. *Activity Diagram* Tampilan Proses Alur Cerita



**Gambar 4.5** *Activity Diagram* Tampilan Proses Alur Cerita

Pada gambar 4.5 menjelaskan bagaimana proses bermain dalam mode alur cerita dengan melakukan alur cerita dan pertempuran diakhiri dengan pertanyaan dari alur cerita sebelumnya.

c. *Activity Diagram Tampilan Proses Survival*



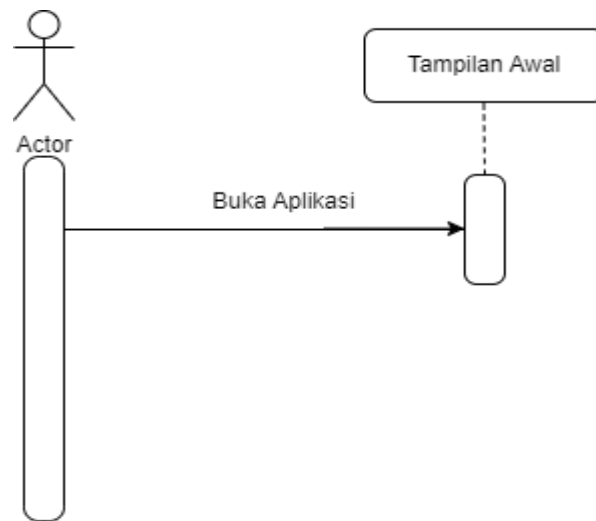
**Gambar 4.6** *Activity Diagram Tampilan Proses Survival*

Pada gambar 4.6 menjelaskan bagaimana proses bermain dalam mode *Survival* dengan melewati beberapa *level* dalam mode *Survival*.

### 3. *Sequence Diagram*

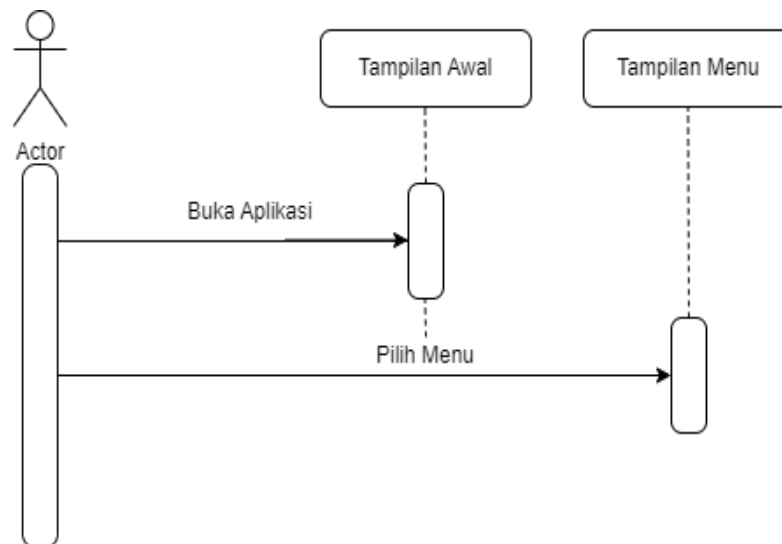
*Sequence Diagram* merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan *message* (pesan) apa yang dikirim dan kapan pelaksanaannya

a. *Sequence* diagram Tampilan awal



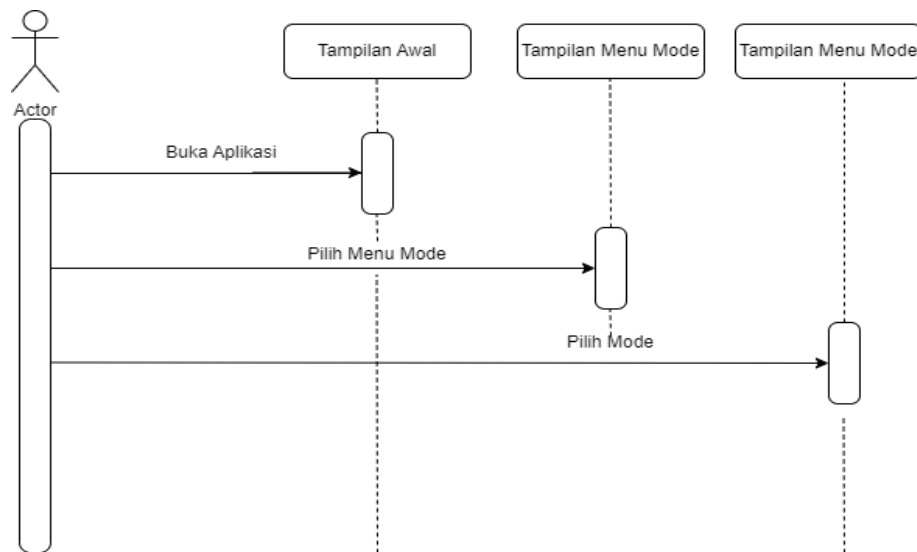
**Gambar 4.7** *Sequence* diagram Tampilan awal

b. *Sequence* diagram Pilihan Menu



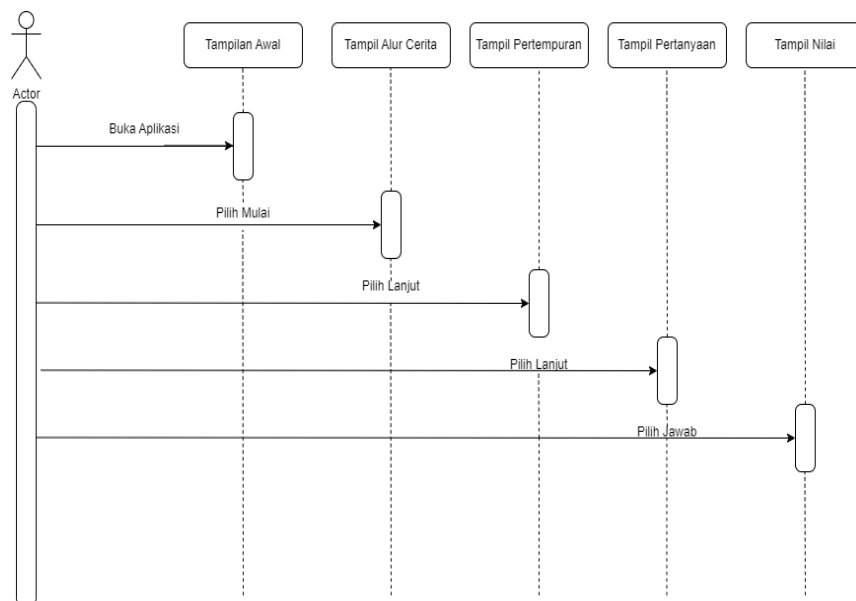
**Gambar 4.8** *Sequence* diagram Piliha Menu

c. *Sequence diagram* Pilihan Menu Mode



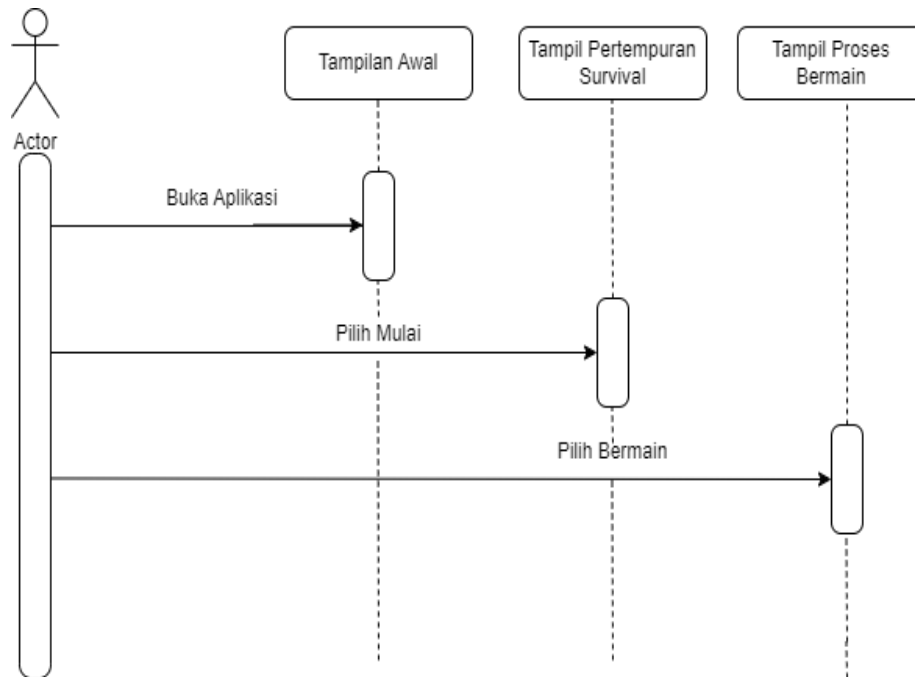
**Gambar 4.9** *Sequence diagram* Pilihan Menu Mode

d. *Sequence Diagram* Tampilan Proses Alur Cerita



**Gambar 4.10** *Sequence diagram* Proses Alur Cerita

e. *Sequence Diagram* Tampilan Proses Survival

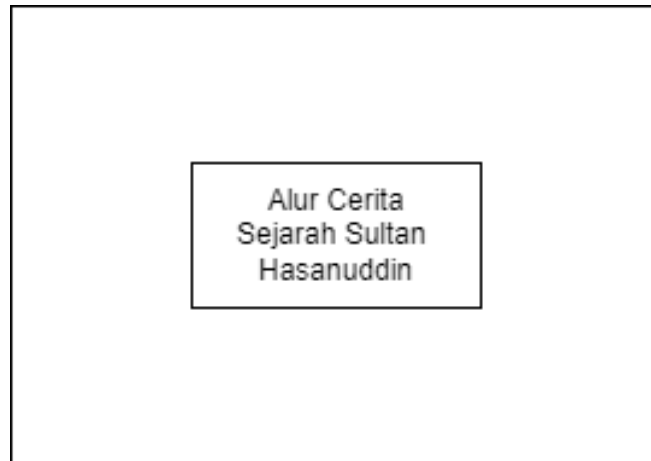


**Gambar 4.11** *Sequence Diagram* Tampilan Proses Survival

## B. Rancangan Sistem

### 1. Rancangan Tampilan Intro

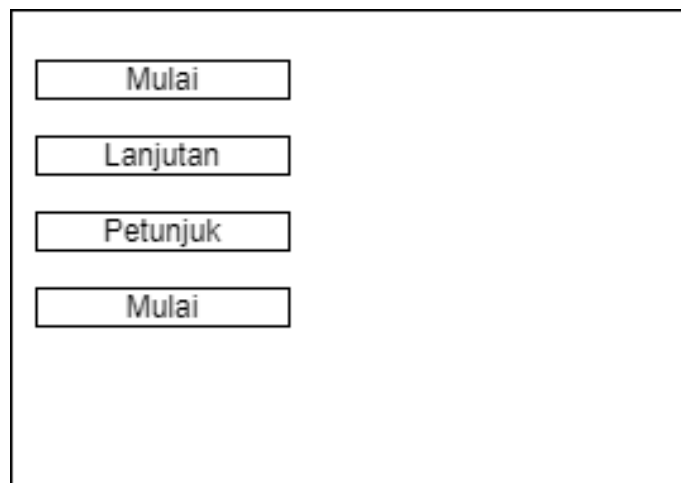
Merupakan tampilan intro sejarah Sultan Hasanuddin



**Gambar 4.12** Rancangan Tampilan Intro

### 2. Rancangan Tampilan Menu

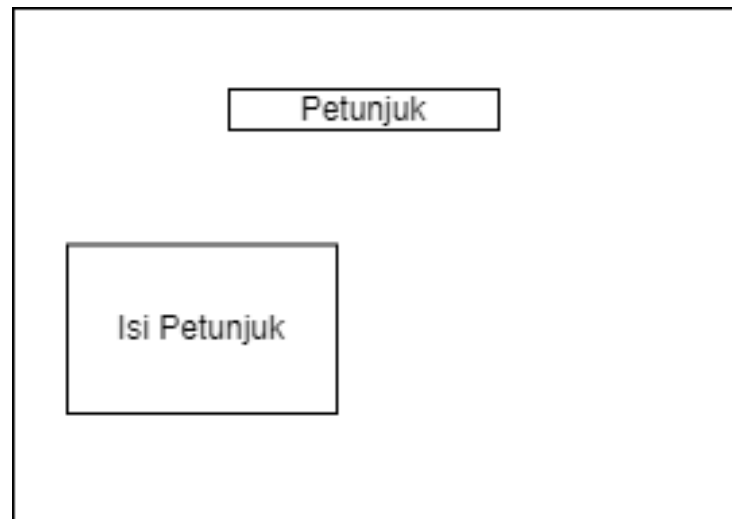
Merupakan tampilan beranda pada game yang terdiri dari mulai, lanjutkan, petunjuk dan keluar



**Gambar 4.13** Rancangan Tampilan Menu

### 3. Rancangan Tampilan Petunjuk

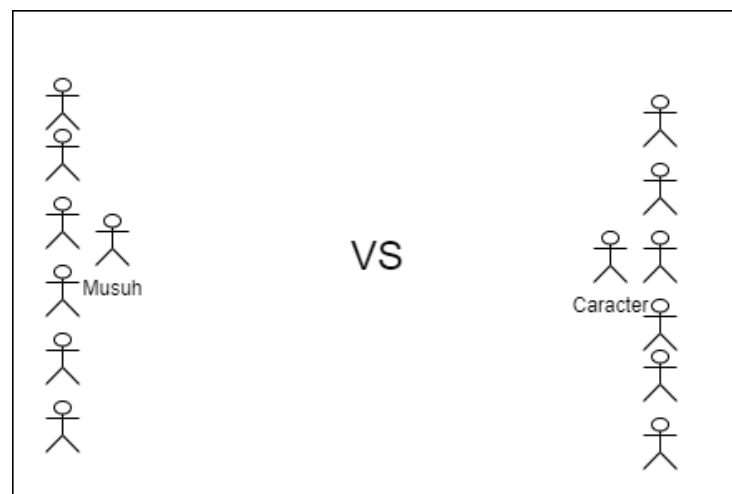
Merupakan tampilan Petunjuk pada game untuk mendapatkan informasi penggunaan game



**Gambar 4.14** Rancangan Tampilan Petunjuk

### 4. Rancangan Tampilan Lanjutan

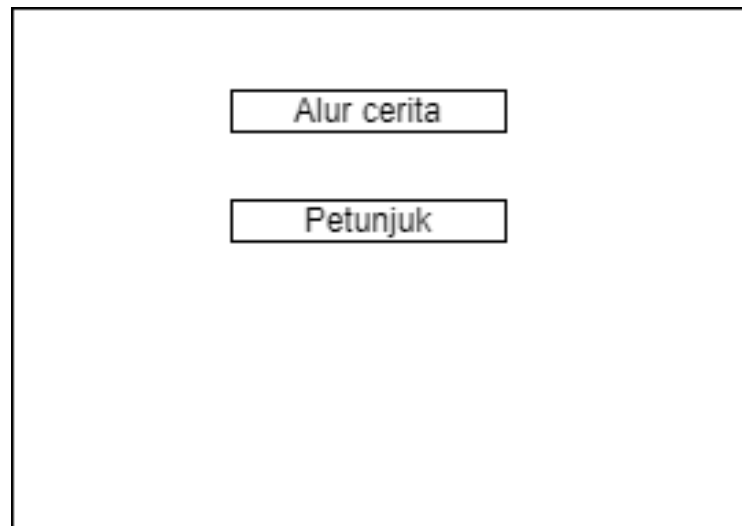
Merupakan tampilan lanjutan menuju pada halaman pertempuran yang telah disimpan sebelumnya



**Gambar 4.15** Rancangan Tampilan Lanjutan

### 5. Rancangan Tampilan Mulai

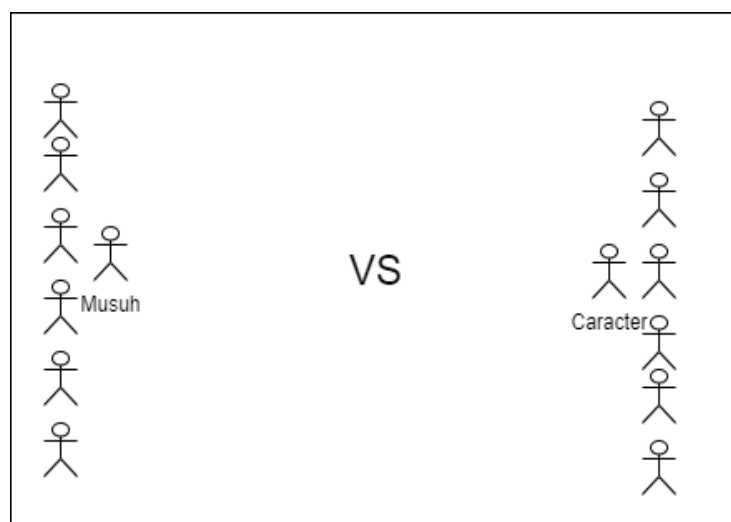
Merupakan tampilan mulai menuju pada halaman alur cerita yang terdiri 6 alur cerita dan dilanjutkan ke pertempuran



**Gambar 4.16** Rancangan Tampilan Mulai

### 6. Rancangan Tampilan Pertempuran

Merupakan tampilan pertempuran yang terdiri atas 6 alur cerita pertempuran



**Gambar 4.17** Rancangan Tampilan Pertempuran

## 7. Rancangan Tampilan *Puase*

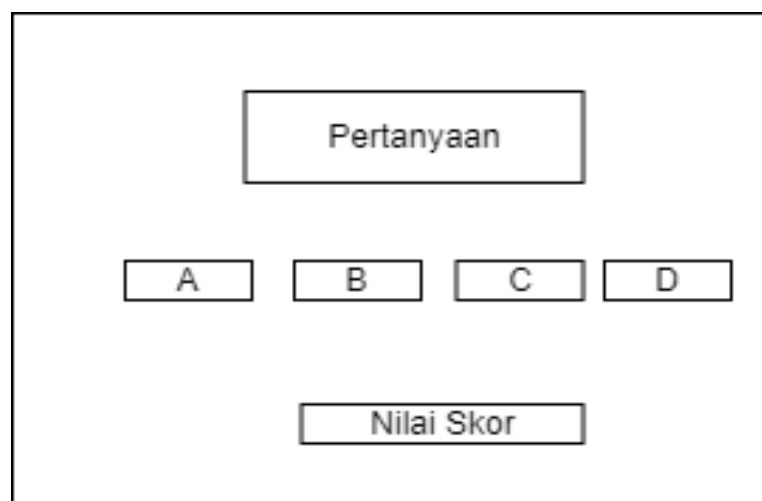
Merupakan tampilan *puase* untuk memberentikan permainan



**Gambar 4.18** Rancangan Tampilan *Pause*

## 8. Rancangan Tampilan Pertanyaan

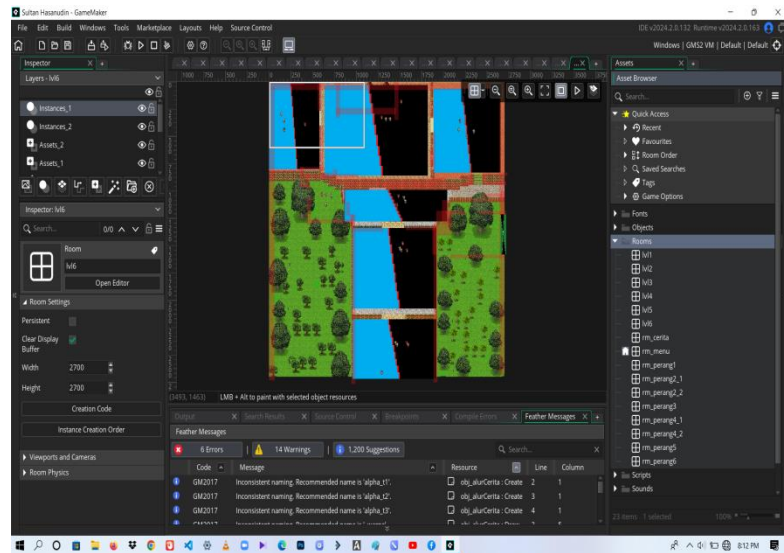
Merupakan tampilan Pertanyaan yang disediakan setelah melewati alur cerita dan pertempuran untuk memberentikan permainan



**Gambar 4.19** Rancangan Tampilan Pertanyaan

## C. Desain Sistem

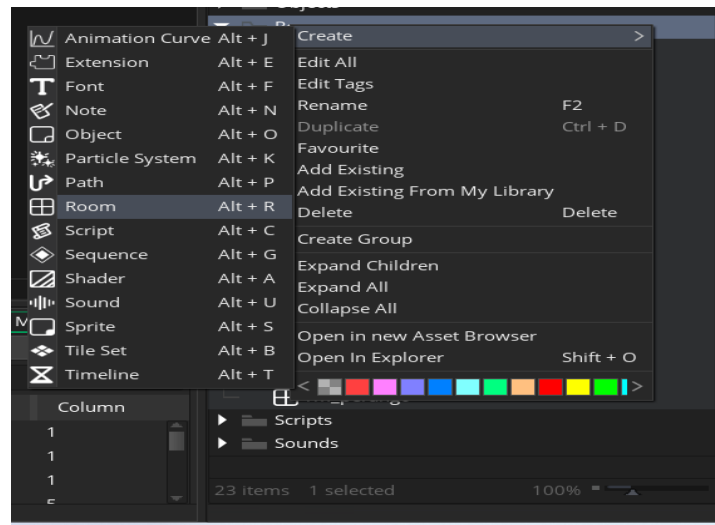
### 1. Tampilan Room



**Gambar 4.20** Tampilan Room

Pada Tampilan *room* merupakan gambaran map atau ruangan yang dipakai pada pada program seperti halnya perencanaan map. Adapun tahap yang harus dilakukan dalam pembuatan map atau *room* sebagai berikut:

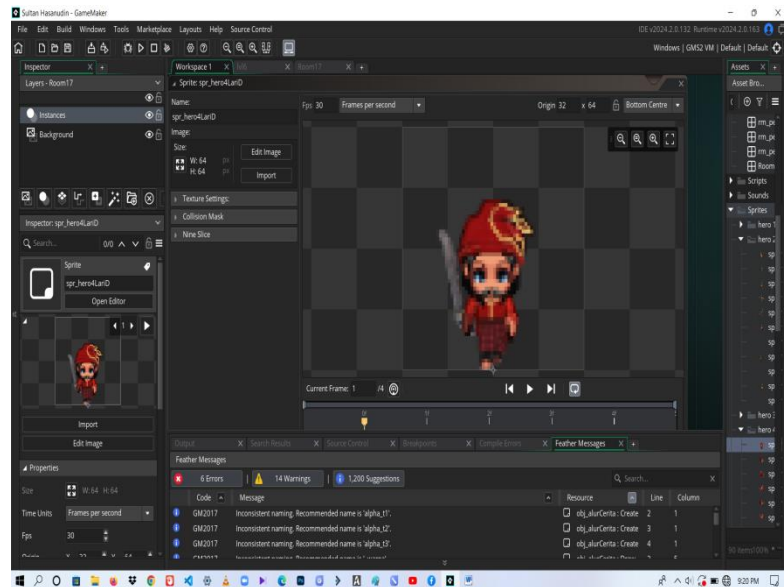
a. *Create Room*



**Gambar 4.21** Tampilan *Create Room*

- 1) Pertama yaitu tempatkan kursor pada bagian *Asset Room* yaitu *create*
- 2) Lalu pilih bagian *room*
- 3) Maka *room* selesai dibuat dan siap untuk di isi dengan desain yang memerlukan *spirites* dan *Tile Sets*.

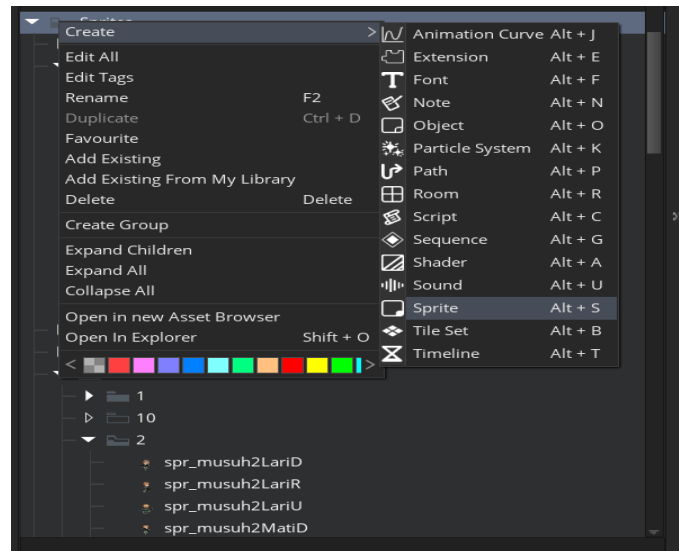
## 2. Tampilan *Sprites*



**Gambar 4.22** Tampilan *Sprites*

Pada Tampilan *Sprites* merupakan gambaran desain karakter atau bagian pendukung map yang dipakai pada pada program seperti halnya perencanaan karakter dan map. Adapun Tahap yang harus dilakukan dalam pembuatan map atau *room* sebagai berikut:

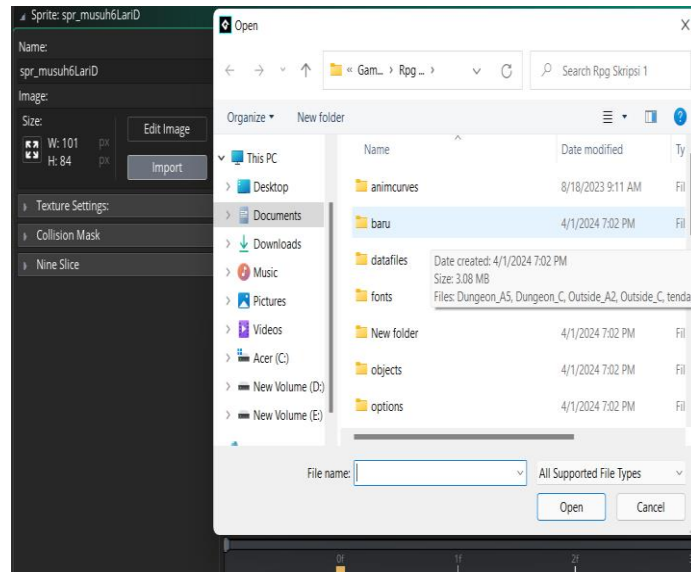
b. *Create Sprites*



**Gambar 4.23** Tampilan *Create Sprite*

- 1) Pertama yaitu tempatkan kursor pada bagian *Asset Sprites* yaitu *create*
- 2) Lalu pilih bagian *sprites*
- 3) Maka *sprites* selesai dibuat dan siap untuk di isi dengan desain *sprites* yang telah disediakan.

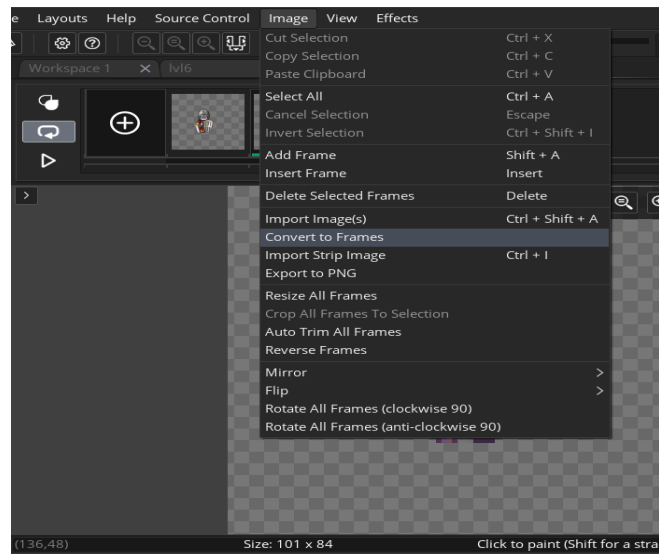
### c. Import Sprites



**Gambar 4.24** Tampilan *Import Sprites*

- 1) Pertama yaitu klik bagian *sprites* yang telah dibuat sebelumnya
- 2) Lalu pilih bagian *import*
- 3) Silahkan pilih bagian *sprites* yang telah disediakan seperti karakter dalam bentuk *png* maupun animasi.

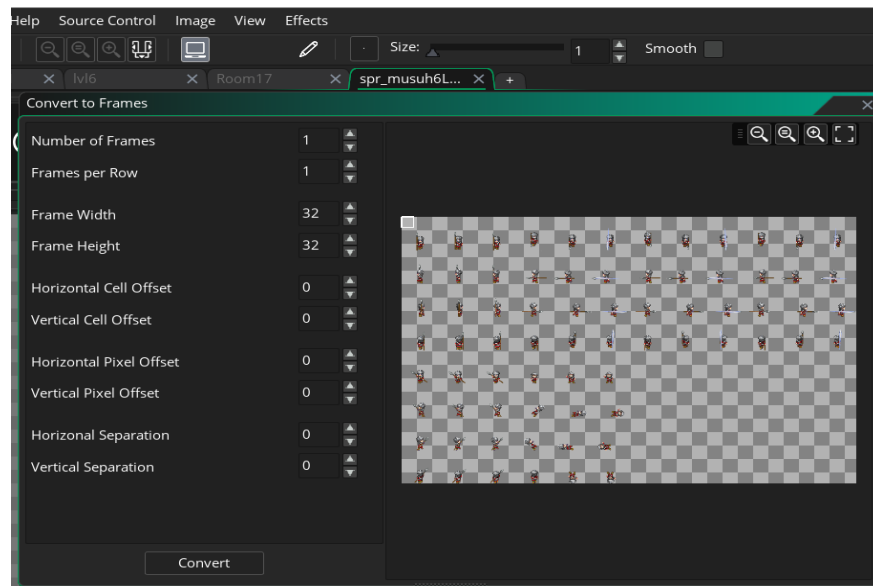
d. *Edit Image*



**Gambar 4.25** Tampilan *Edit Image*

- 1) Setelah klik bagian *Edit Image* pada sprites, Silahkan klik bagian *image*
- 2) Lalu pilih bagian *Convert to Frames*
- 3) Selanjutnya akan di arahkan ke pemilihan gerakan

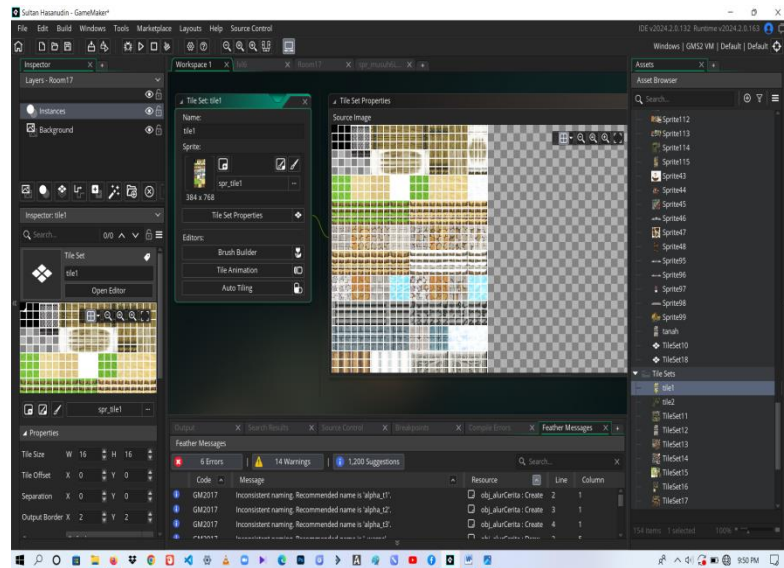
e. *Convert Frame*



**Gambar 4.26** Tampilan *Convert Frame*

- 1) Menentukan Pemilihan Gerakan dengan jumlah *frame* yang tersedia pada *Frame width* dan *Frame Height*
- 2) Silahkan Klik *Convert* ketika sudah sesuai dengan pemilihan gerakan
- 3) *Sprites* telah selesai dibuat.

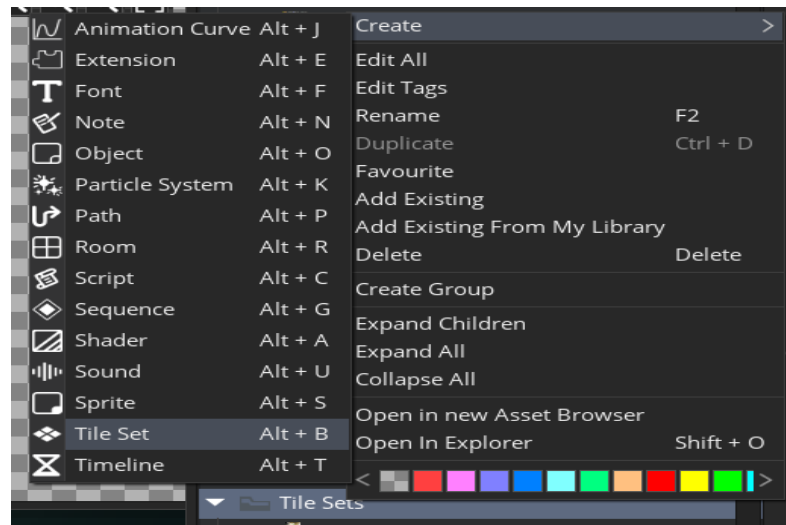
### 3. Tampilan *Tile Sets*



**Gambar 4.27** Tampilan *Tile Sets*

Pada Tampilan *Tile Sets* merupakan gambaran desain bagian pendukung map atau karpet pada map yang dipakai pada pada program seperti halnya perencanaan map. Adapun Tahap yang harus dilakukan dalam pembuatan map atau *room* sebagai berikut:

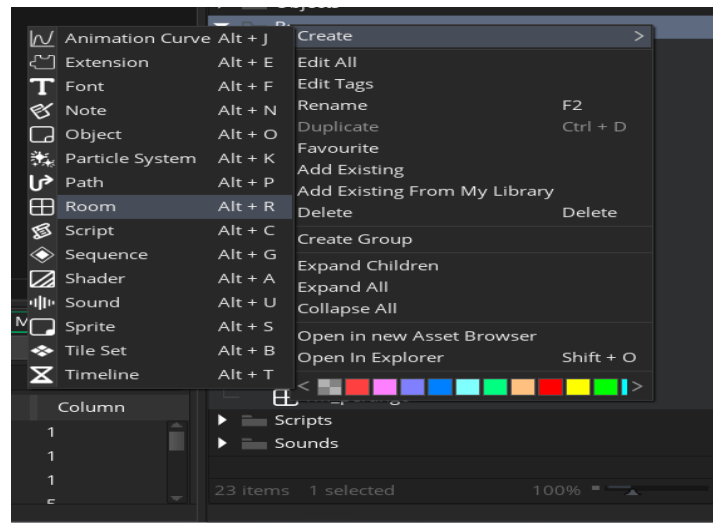
a. *Create Tile Set*



**Gambar 4.28** Tampilan *Create Tile Set*

- 1) Pertama yaitu tempatkan kursor pada bagian *Asset Tile Sets* yaitu *create*
- 2) Lalu pilih bagian *tile Set*
- 3) Maka *Tile set* selesai dibuat dan siap untuk mengimport *spirite* yang telah dibuat.

b. *Select Sprites*

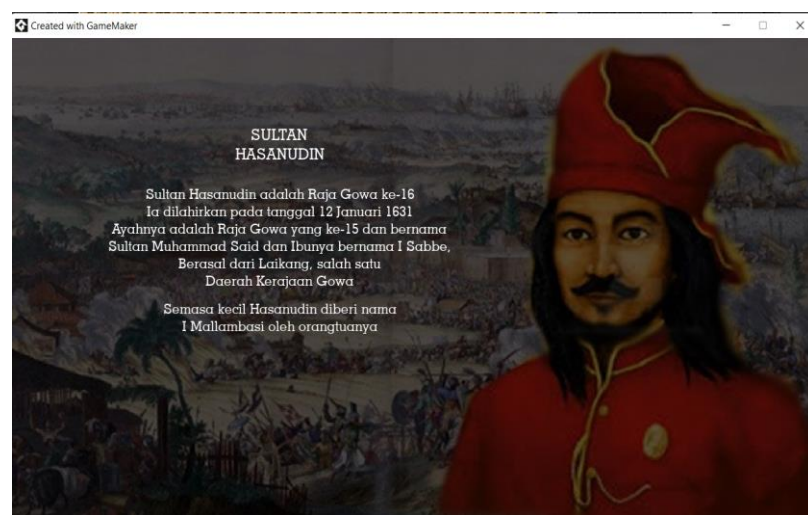


**Gambar 4.29** Tampilan *Select Sprites*

- 1) Mengganti Nama *Tile Set* terlebih dahulu agar lebih mudah digunakan
- 2) Memilih bagian *Select Sprites* untuk mengkonfirmasi *sprites* yang telah dibuat
- 3) Maka akan diarahkan untuk pemilihan *sprites* yang telah dibuat sebelumnya.

#### 4. Tampilan Intro

Pada Tampilan Intro terdapat cerita singkat mengenai sejarah tentang Sultan Hasanuddin dengan latar *background* yaitu Sultan Hasanuddin dilengkapi dengan mengklik *space* pada *keyboard* untuk melanjutkan ke menu selanjutnya.

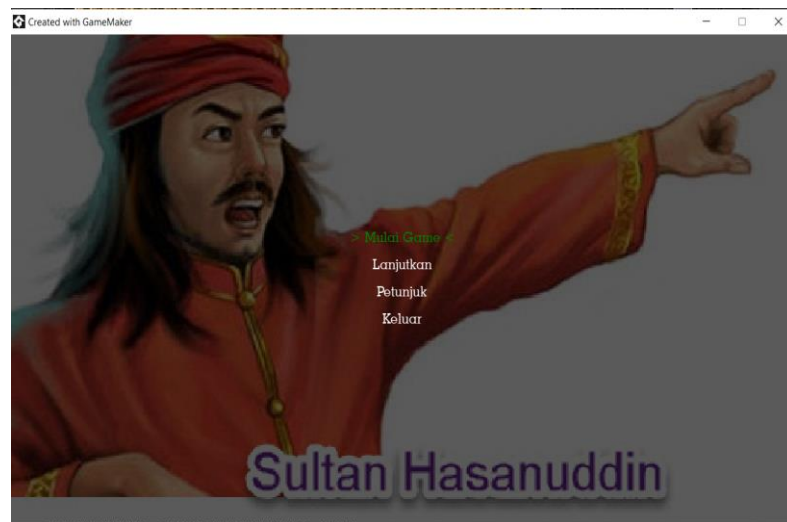


**Gambar 4.30** Tampilan Intro

Pada Gambar 4.30 diatas dirancancang menggunakan aplikasi *Game Maker Studio* dengan bahasa programan yang diciptakan oleh *Game maker* itu sendiri yaitu *Game Maker Language* dan *font* yang digunakan adalah *GeoSlab703 Md BT*.

## 5. Tampilan Menu

Pada Tampilan Menu utama terdapat button Mulai *game*, Lanjutkan, Petunjuk, dan keluar. Pengguna dapat memilih button Mulai *Game* jika ingin langsung memaikannya.



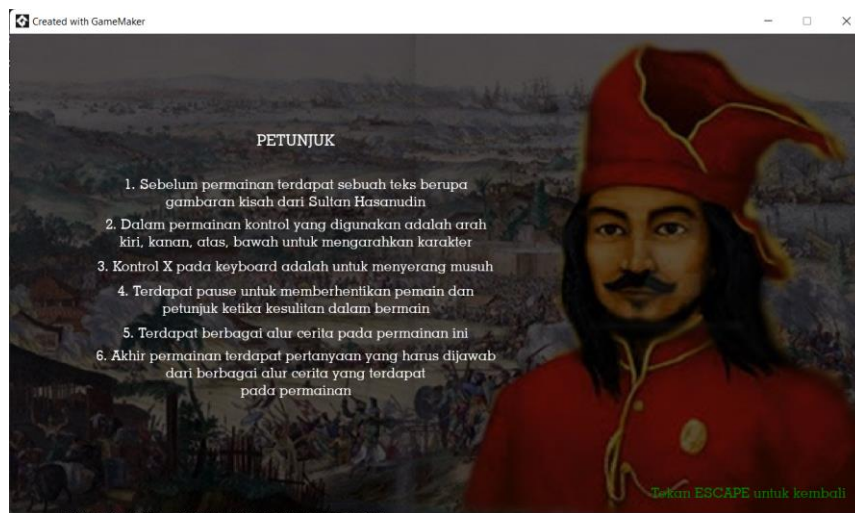
**Gambar 4.31** tampilan menu

Pada Gambar 4.31 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa *Pemrograman Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan menu sebagai berikut:

- 1) Pertama yaitu menentukan Font yang ingin digunakan adalah *GeoSlab703 Md BT*
- 2) Dibuat dengan rooms yang terdapat pada *asset* aplikasi (rm\_menu)
- 3) Agar jalannya program pada menu ini diatur pada *asset object* pada aplikasi dengan mengatur perintah yang sesuai agar menu jalan (Kontrol obj\_menu) adalah bagian pada program ini

## 6. Tampilan Petunjuk

Pada tampilan petunjuk yaitu menampilkan beberapa cara penggunaan tentang aturan *game* atau cara memainkan game ini dan menekan *escape* untuk kembali ke menu sebelumnya.



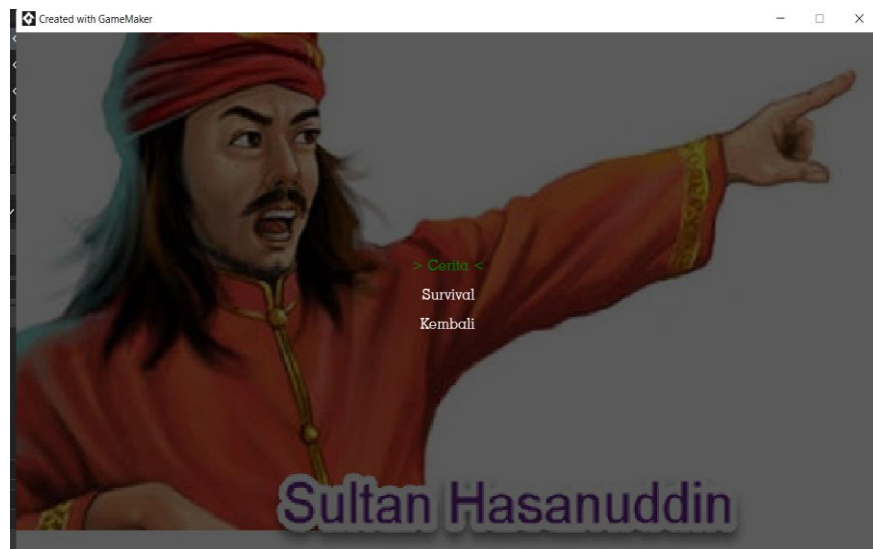
**Gambar 4.32** Tampilan Petunjuk

Pada Gambar 4.32 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa *Pemrograman Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan petunjuk sebagai berikut:

- a. Menentukan Font yang digunakan adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada *asset* aplikasi(*rm\_menu*)
- c. Untuk menjalankan program tampilan petunjuk dengan mengatur *object* pada *asset* (Kontrol *obj\_menu*) dengan *events* Sistem dan Perintah dan Penampilan menu pada program ini.

#### 4. Tampilan Menu Mulai *Game*

Pada Tampilan Mulai *Game* terdapat dua pilihan yaitu *button mode* Alur cerita dan *Survival* dengan gaya yang berbeda yaitu alur cerita dan alur *random*



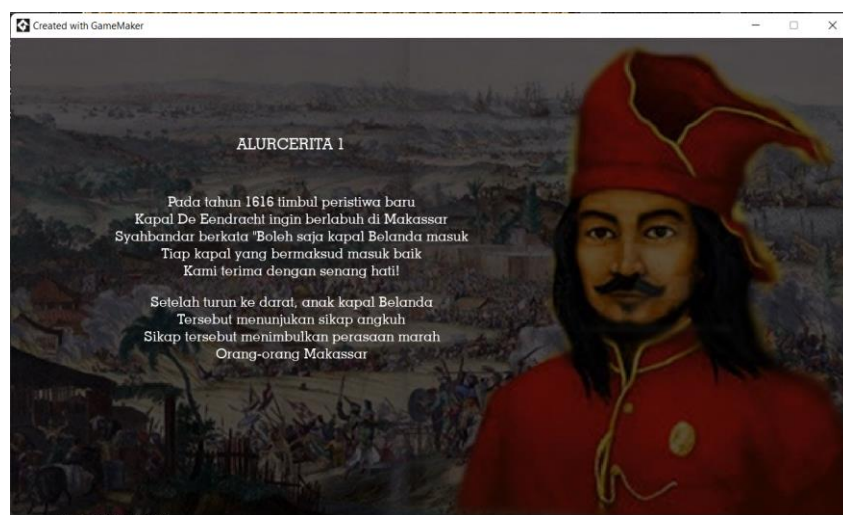
**Gambar 4.33** Tampilan menu mulai game

Pada Gambar 4.33 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan menu mulai *game* sebagai berikut:

- a. Menentukan Font yang akan digunakan adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada *asset* aplikasi (*rm\_menu*)
- c. Agar program tampilan menu mulai *game* dapat di jalan yaitu dengan mengatur *object* pada *asset* (Kontrol *obj\_menu*) dengan *events* Sistem dan Perintah dan Penampilan menu pada program ini.

#### 4. Tampilan Alur Cerita 1

Pada Tampilan Alur Cerita 1 terdapat alur cerita mengenai peristiwa yang terjadi sejarah tentang Sultan Hasanuddin dengan latar *background* yaitu Sultan Hasanuddin dilengkapi dengan mengklik *space* pada *keyboard* untuk melanjutkan ke halaman pertempuran



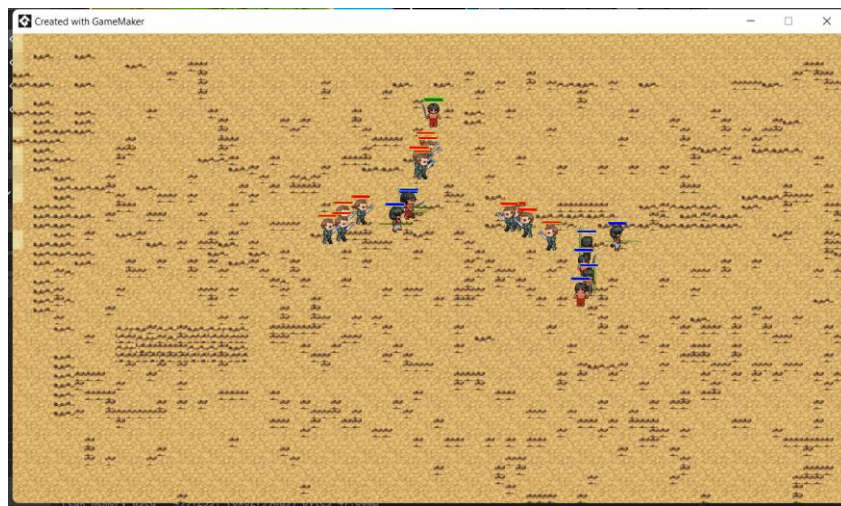
**Gambar 4.34** Tampilan Alur Cerita 1

Pada Gambar 4.34 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan alur cerita 1 sebagai berikut:

- a. Menentukan Font yang akan digunakan adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada *asset* aplikasi (*rm\_cerita*)
- c. Untuk menjalankan tampilan menu mulai *game* kita dapat mengatur *object* pada *asset* (Kontrol *obj\_alur cerita*) dengan *events* tampilan.

## 5. Tampilan Alur Pertempuran 1

Pada Tampilan Alur Pertempuran 1 yaitu menampilkan pertempuran dari objek yang telah diprograman sesuai dengan perintah yang diinginkan dengan latar yang sesuai dengan peristiwa alur cerita sebelumnya.



**Gambar 4.35** Tampilan Pertempuran 1

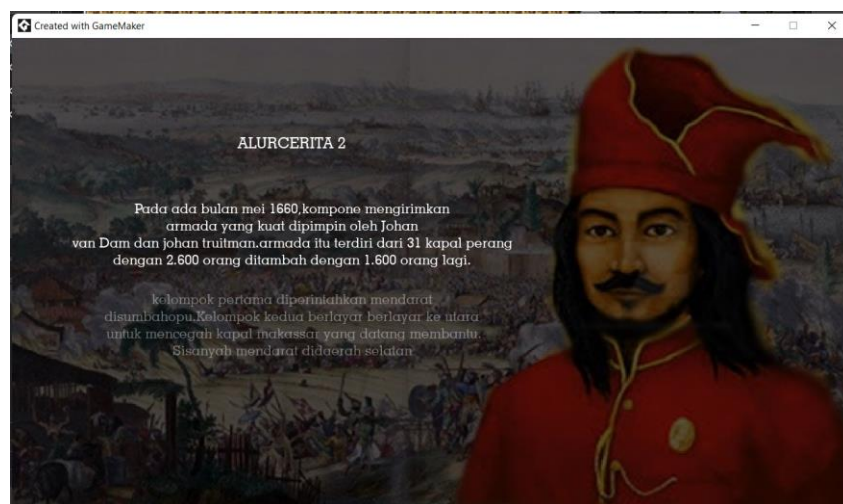
Pada Gambar 4.35 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan pertempuran 1 sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada asset aplikasi (rm\_perang 1)
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Tittle Sets* sebagai latar atau biasa dikenal keramik pada asset

- d. Untuk menjalankan program pertempuran 1 dapat mengatur *object* pada *asset* (Pengikut, Musuh dan Sultan) dengan mengatur kontrolnya sebagai berikut:
- Pengikut *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
  - Musuh *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
  - Untuk Menjalan Keseluruhan diatur pada Kontrol *Asset* (obj\_global, obj\_global\_ai, obj\_kontrol\_musuh, obj\_kontrol pengikut, obj\_mask\_serangan).

## 6. Tampilan Alur Cerita 2

Pada Tampilan Alur Cerita 2 terdapat alur cerita mengenai peristiwa yang terjadi sejarah tentang Sultan Hasanuddin dengan latar *background* yaitu Sultan Hasanuddin dilengkapi dengan mengklik *space* pada *keyboard* untuk melanjutkan ke halaman pertempuran



**Gambar 4.36** Tampilan Alur Cerita 2

Pada Gambar 4.36 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan alur cerita 2 sebagai berikut:

- a. Menentukan Font yang digunakan adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada *asset* aplikasi (*rm\_cerita2*)
- c. Untuk menjalankan tampilan menu mulai *game* kita dapat mengatur *object* pada *asset* (Kontrol *obj\_alur cerita*) dengan *events* tampilan.

## 7. Tampilan Alur Pertempuran 2

Pada Tampilan Alur Pertempuran 2 yaitu menampilkan pertempuran dari objek yang telah diprograman sesuai dengan perintah yang diinginkan dengan latar yang sesuai dengan peristiwa alur cerita sebelumnya.



**Gambar 4.37** Tampilan Pertempuran 2

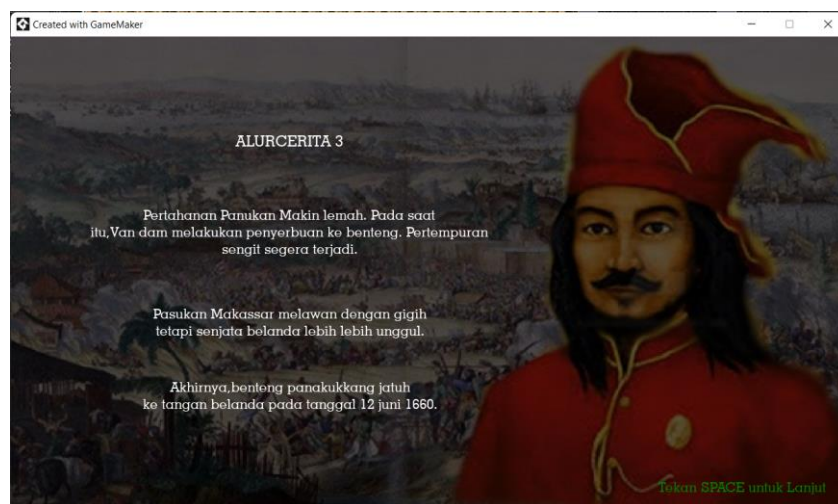
Pada Gambar 4.37 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan pertempuran 2 sebagai berikut:

- a. Menentukan *Font* yang akan digunakan yang terdapat pada asset adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada asset aplikasi (rm\_perang 2)
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Tittle Sets* sebagai latar atau biasa dikenal keramik pada *asset*

- d. Untuk menjalankan program pertempuran 1 dapat mengatur *object* pada *asset* (Pengikut, Musuh dan Sultan) dengan mengatur kontrolnya sebagai berikut:
- Pengikut *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
  - Musuh *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
  - Untuk Menjalan Keseluruhan diatur pada Kontrol *Asset* (obj\_global, obj\_global\_ai, obj\_kontrol\_musuh, obj\_kontrol pengikut, obj\_mask\_serangan).

## 8. Tampilan Alur Cerita 3

Pada Tampilan Alur Cerita 3 terdapat alur cerita mengenai peristiwa yang terjadi sejarah tentang Sultan Hasanuddin dengan latar *background* yaitu Sultan Hasanuddin dilengkapi dengan mengklik *space* pada *keyboard* untuk melanjutkan ke halaman pertempuran



**Gambar 4.38** Tampilan Alur Cerita 3

Pada Gambar 4.38 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan alur cerita 3 sebagai berikut:

- a. Menentukan Font yang akan digunakan adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada *asset* aplikasi(*rm\_cerita3*)
- c. Untuk menjalankan tampilan menu mulai *game* kita dapat mengatur *object* pada *asset* (Kontrol *obj\_alur cerita*) dengan *events* tampilan.

### 9. Tampilan Alur Pertempuran 3

Pada Tampilan Alur Pertempuran 3 yaitu menampilkan pertempuran dari objek yang telah diprograman sesuai dengan perintah yang diinginkan dengan latar yang sesuai dengan peristiwa alur cerita sebelumnya.



**Gambar 4.39** Tampilan Pertempuran 3

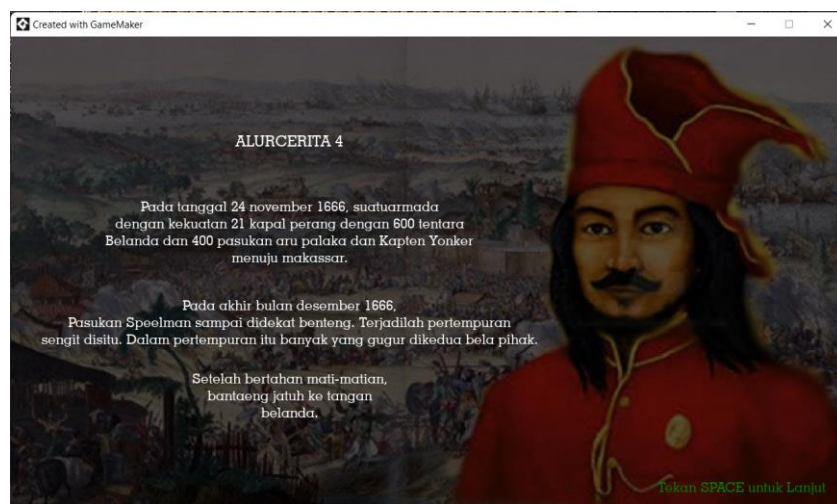
Pada Gambar 4.39 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan pertempuran 3 sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *GeoSlab703 Md BT*
- b. Dibuat dengan rooms yang terdapat pada asset aplikasi (rm\_perang 3)
- c. Menambahkan Asset *Sprite* yang telah disediakan sebagai karakter dan *Tittle Sets* sebagai latar atau biasa dikenal keramik pada asset

- d. Untuk menjalankan program pertempuran 3 dapat mengatur *object* pada *asset* (Pengikut, Musuh dan sultan) dengan mengatur kontrolnya sebagai berikut:
- Pengikut *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
  - Musuh *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
  - Untuk Menjalan Keseluruhan diatur pada Kontrol *Asset* (obj\_global, obj\_global\_ai, obj\_kontrol\_musuh, obj\_kontrol pengikut, obj\_mask\_serangan).

## 10. Tampilan Alur Pertempuran 3

Pada Tampilan Alur Cerita 3 terdapat alur cerita mengenai peristiwa yang terjadi sejarah tentang Sultan Hasanuddin dengan latar *background* yaitu Sultan Hasanuddin dilengkapi dengan mengklik *space* pada *keyboard* untuk melanjutkan ke halaman pertempuran



**Gambar 4.40** Tampilan Alur Cerita 4

Pada Gambar 4.40 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan alur cerita 4 sebagai berikut:

- a. Menentukan Font yang akan digunakan adalah *GeoSlab703 Md BT*
- b. Dibuat dengan rooms yang terdapat pada *asset* aplikasi (*rm\_cerita4*)
- c. Untuk menjalankan tampilan menu mulai *game* kita dapat mengatur *object* pada *asset* (Kontrol *obj\_alur cerita*) dengan *events* tampilan.

## 11. Tampilan Alur Pertempuran 4

Pada Tampilan Alur Pertempuran 4 yaitu menampilkan pertempuran dari objek yang telah diprograman sesuai dengan perintah yang diinginkan dengan latar yang sesuai dengan peristiwa alur cerita sebelumnya.



**Gambar 4.41** Tampilan Pertempuran 4

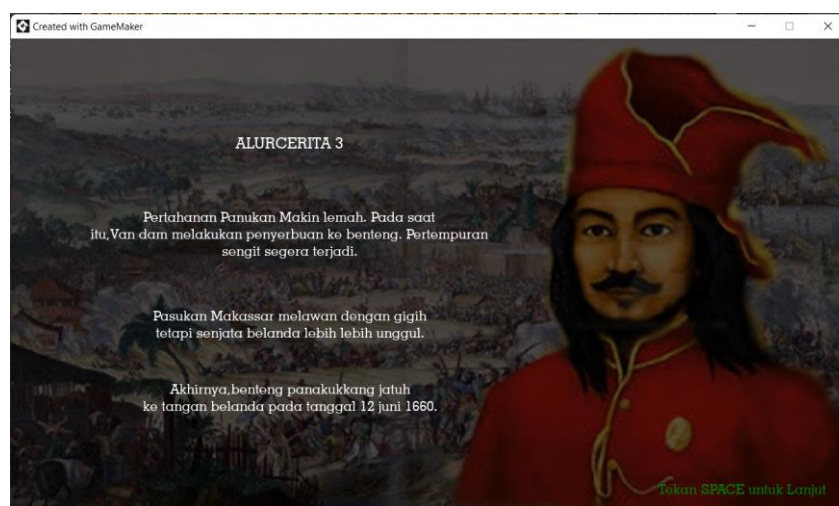
Pada Gambar 4.41 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan *Bahasa Pemrograman Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan pertempuran 4 sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada asset aplikasi (rm\_perang 4)
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Tittle Sets* sebagai latar atau biasa dikenal keramik pada asset

- d. Untuk menjalankan program pertempuran 5 dapat mengatur *object* pada *asset* (Pengikut, Musuh dan sultan) dengan mengatur kontrolnya sebagai berikut:
- Pengikut *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
  - Musuh *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
  - Untuk Menjalan Keseluruhan diatur pada Kontrol *Asset* (obj\_global, obj\_global\_ai, obj\_kontrol\_musuh, obj\_kontrol pengikut, obj\_mask\_serangan).

## 12. Tampilan Alur Cerita 5

Pada Tampilan Alur Cerita 5 terdapat alur cerita mengenai peristiwa yang terjadi sejarah tentang Sultan Hasanuddin dengan latar *background* yaitu Sultan Hasanuddin dilengkapi dengan mengklik *space* pada *keyboard* untuk melanjutkan ke halaman pertempuran



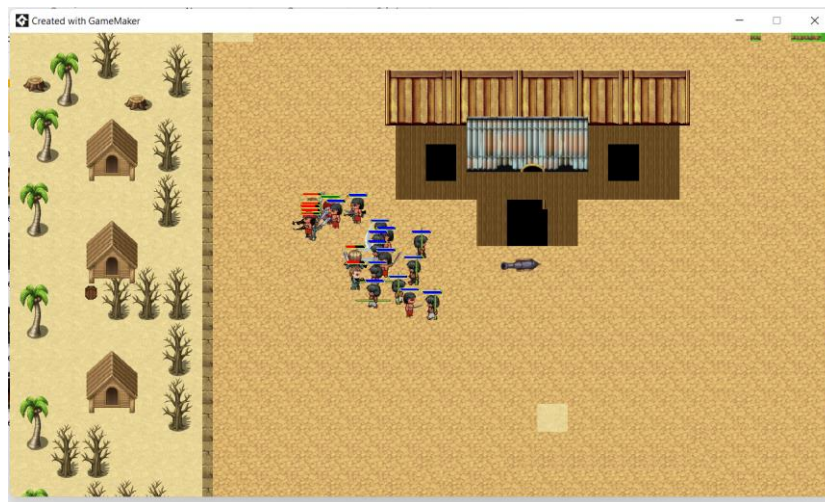
**Gambar 4.42** Tampilan Alur Cerita 5

Pada Gambar 4.42 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan alur cerita 5 sebagai berikut:

- Menentukan Font yang akan digunakan adalah *GeoSlab703 Md BT*
- Dibuat dengan *rooms* yang terdapat pada asset aplikasi(rm\_cerita5)
- Untuk menjalankan tampilan menu mulai *game* kita dapat mengatur *object* pada *asset* (Kontrol obj\_alur cerita) dengan *events* tampilan.

### 13. Tampilan Alur Pertempuran 5

Pada Tampilan Alur Pertempuran 5 yaitu menampilkan pertempuran dari objek yang telah diprograman sesuai dengan perintah yang diinginkan dengan latar yang sesuai dengan peristiwa alur cerita sebelumnya.



**Gambar 4.43** Tampilan Pertempuran 5

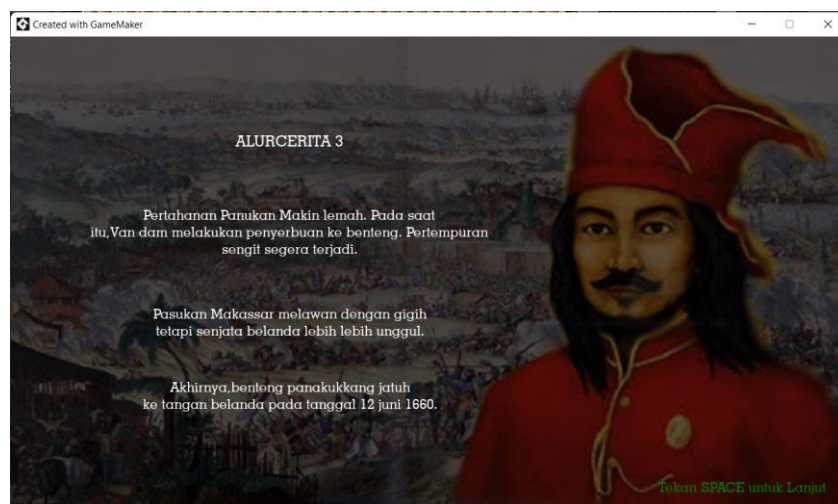
Pada Gambar 4.43 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan pertempuran 5 sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada *asset* aplikasi (rm\_perang 5)
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Tittle Sets* sebagai latar atau biasa dikenal keramik pada *asset*

- d. Untuk menjalankan program pertempuran 5 dapat mengatur *object* pada *asset* (Pengikut, Musuh dan sultan) dengan mengatur kontrolnya sebagai berikut:
- Pengikut *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
  - Musuh *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
  - Untuk Menjalan Keseluruhan diatur pada Kontrol *Asset* (obj\_global, obj\_global\_ai, obj\_kontrol\_musuh, obj\_kontrol pengikut, obj\_mask\_serangan).

#### 14. Tampilan Alur Cerita 6

Pada Tampilan Alur Cerita 6 terdapat alur cerita mengenai peristiwa yang terjadi sejarah tentang Sultan Hasanuddin dengan latar *background* yaitu Sultan Hasanuddin dilengkapi dengan mengklik *space* pada *keyboard* untuk melanjutkan ke halaman pertempur



**Gambar 4.44** Tampilan Alur Cerita 6

Pada Gambar 4.44 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan alur cerita 6 sebagai berikut:

- a. Menentukan Font yang akan digunakan adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada *asset* aplikasi(*rm\_cerita6*)
- c. Untuk menjalankan tampilan menu mulai *game* kita dapat mengatur *object* pada *asset* (Kontrol *obj\_alur cerita*) dengan *events* tampilan.

## 15. Tampilan Alur Pertempuran 6

Pada Tampilan Alur Pertempuran 6 yaitu menampilkan pertempuran dari objek yang telah diprograman sesuai dengan perintah yang diinginkan dengan latar yang sesuai dengan peristiwa alur cerita sebelumnya.



**Gambar 4.45** Tampilan Pertempuran 6

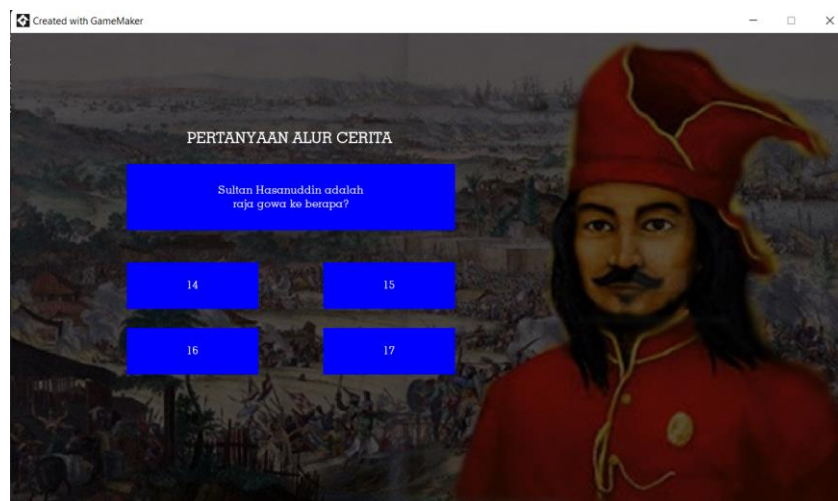
Pada Gambar 4.45 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan pertempuran 6 sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada asset aplikasi (rm\_perang 6)
- c. Menambahkan Asset *Sprite* yang telah disediakan sebagai karakter dan *Tittle Sets* sebagai latar atau biasa dikenal keramik pada asset

- d. Untuk menjalankan program pertempuran 6 dapat mengatur *object* pada *asset* (Pengikut, Musuh dan sultan) dengan mengatur kontrolnya sebagai berikut:
- Pengikut *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
  - Musuh *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
  - Untuk Menjalan Keseluruhan diatur pada Kontrol *Asset* (obj\_global, obj\_global\_ai, obj\_kontrol\_musuh, obj\_kontrol pengikut, obj\_mask\_serangan).

## 16. Tampilan Pertanyaan

Pada Tampilan Pertanyaan yaitu berupa pertanyaan seputar alur cerita yang telah berlalu yang dijadikan sebagai pertanyaan untuk mendapatkan nilai skor



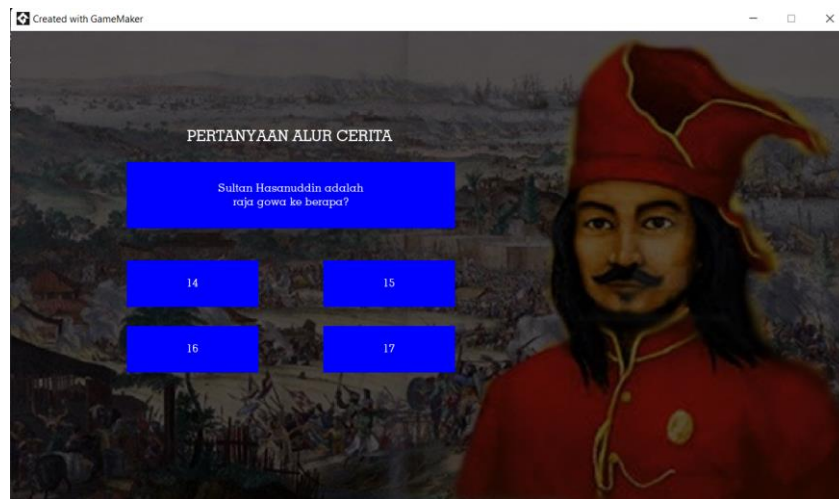
**Gambar 4.46** Tampilan Pertanyaan

Pada Gambar 4.46 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan pertanyaan sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada *asset* aplikasi(*rm\_cerita*)
- c. Untuk menjalankan tampilan Pertanyaan alur cerita kita dapat mengatur *object* pada *asset* (Kontrol *obj\_alur cerita*) dengan events pertanyaan dan jawab pertanyaan.

## 17. Tampilan Skor

Pada Tampilan Skor yaitu berupa Hasil rekap dari hasil jawaban yang telah dijawab pada pertanyaan alur



**Gambar 4.47** Tampilan Skor

Pada Gambar 4.47 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan skor sebagai berikut:

- a. Menentukan Font yang akan digunakan adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada *asset* aplikasi (*rm\_cerita*)
- c. Untuk menjalankan tampilan Pertanyaan alur cerita kita dapat mengatur *object* pada *asset* (Kontrol *obj\_alur cerita*) dengan *events Variabel*.

## 18. Tampilan *Survival Level 1*

Pada Tampilan *Survival Level 1* yaitu menampilkan pertempuran dari objek yang telah diprograman sesuai dengan perintah yang diinginkan dengan latar yang sesuai dengan peristiwa alur cerita sebelumnya.



**Gambar 4.48** Tampilan *Survival Level 1*

Pada Gambar 4.48 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan *survival level 1* sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada *asset* aplikasi(lvl1)
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Tittle Sets* sebagai latar atau biasa dikenal keramik pada *asset*

- d. Untuk menjalankan program *Survival Level 1* dapat mengatur *object* pada *asset* (Musuh dan sultan) dengan mengatur kontrolnya sebagai berikut:
- Sultan *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
  - Musuh *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
  - Untuk Menjalan Keseluruhan diatur pada Kontrol *Asset* (obj\_global, obj\_global\_ai, obj\_kontrol\_musuh, obj\_kontrol pengikut, obj\_mask\_serangan).
  - Untuk Mengatur Waktu diatur pada object Kontrol yaitu obj\_global dengan event atur ulang variabel.
  - Menggunakan 1 obj ruang (obj ruang adalah bagian batas pergerakan musuh yang diatur pada *object* yaitu obj\_ruang).

### 19. Tampilan *Survival Level 2*

Pada Tampilan *Survival Level 2* yaitu menampilkan pertempuran dari objek yang telah diprograman sesuai dengan perintah yang diinginkan dengan latar yang sesuai dengan peristiwa alur cerita sebelumnya.



**Gambar 4.49** Tampilan *Survival Level 2*

Pada Gambar 4.49 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan *survival level 2* sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada *asset* aplikasi(lvl2)
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Tittle Sets* sebagai latar atau biasa dikenal keramik pada *asset*

d. Untuk menjalankan program *Survival Level 2* dapat mengatur *object* pada *asset* (Musuh dan sultan) dengan mengatur kontrolnya sebagai berikut:

- Sultan *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
- Musuh *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
- Untuk Menjalan Keseluruhan diatur pada Kontrol Asset (obj\_global, obj\_global\_ai, obj\_kontrol\_musuh, obj\_kontrol pengikut, obj\_mask\_serangan).
- Untuk Mengatur Waktu diatur pada object Kontrol yaitu obj\_global dengan *event* atur ulang variabel.
- Menggunakan 2 obj ruang (obj ruang adalah bagian batas pergerakan musuh yang diatur pada object yaitu obj\_ruang).

## 20. Tampilan *Survival Level 3*

Pada Tampilan *Survival Level 1* yaitu menampilkan pertempuran dari objek yang telah diprograman sesuai dengan perintah yang diinginkan dengan latar yang sesuai dengan peristiwa alur cerita sebelumnya.



**Gambar 4.50** Tampilan *Survival Level 3*

Pada Gambar 4.50 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan *survival level3* sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada asset aplikasi(lvl3)
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Tittle Sets* sebagai latar atau biasa dikenal keramik pada asset

d. Untuk menjalankan program *Survival Level 3* dapat mengatur *object* pada *asset* (Musuh dan sultan) dengan mengatur kontrolnya sebagai berikut:

- Sultan *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
- Musuh *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
- Untuk Menjalan Keseluruhan diatur pada Kontrol *Asset* (obj\_global, obj\_global\_ai, obj\_kontrol\_musuh, obj\_kontrol pengikut, obj\_mask\_serangan).
- Untuk Mengatur Waktu diatur pada *object* Kontrol yaitu obj\_global dengan event atur ulang variabel.
- Menggunakan 3 obj ruang (obj ruang adalah bagian batas pergerakan musuh yang diatur pada *object* yaitu obj\_ruang).

## 21. Tampilan *Survival Level 4*

Pada Tampilan *Survival Level 4* yaitu menampilkan pertempuran dari objek yang telah diprograman sesuai dengan perintah yang diinginkan dengan latar yang sesuai dengan peristiwa alur cerita sebelumnya.



**Gambar 4.51** Tampilan *Survival Level 4*

Pada Gambar 4.51 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan *survival level 4* sebagai berikut:

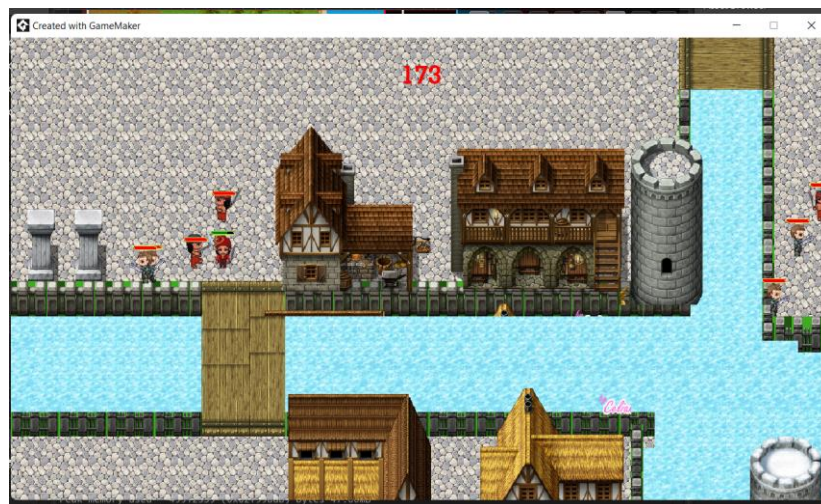
- a. Menentukan Font yang akan digunakan pada asset adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada asset aplikasi(lvl4)
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Tittle Sets* sebagai latar atau biasa dikenal keramik pada asset

d. Untuk menjalankan program *Survival Level 4* dapat mengatur *object* pada *asset* (Musuh dan sultan) dengan mengatur kontrolnya sebagai berikut:

- Sultan *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
- Musuh *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
- Untuk Menjalan Keseluruhan diatur pada Kontrol *Asset* (obj\_global, obj\_global\_ai, obj\_kontrol\_musuh, obj\_kontrol pengikut, obj\_mask\_serangan).
- Untuk Mengatur Waktu diatur pada object Kontrol yaitu obj\_global dengan event atur ulang variabel.
- Menggunakan 4 obj ruang (obj ruang adalah bagian batas pergerakan musuh yang diatur pada object yaitu obj\_ruang).

## 22. Tampilan *Survival Level 5*

Pada Tampilan *Survival Level 5* yaitu menampilkan pertempuran dari objek yang telah diprograman sesuai dengan perintah yang diinginkan dengan latar yang sesuai dengan peristiwa alur cerita sebelumnya.



**Gambar 4.52** Tampilan *Survival Level 5*

Pada Gambar 4.52 diatas dirancang dengan menggunakan aplikasi Game Maker Studio dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan *survival level 5* sebagai berikut:

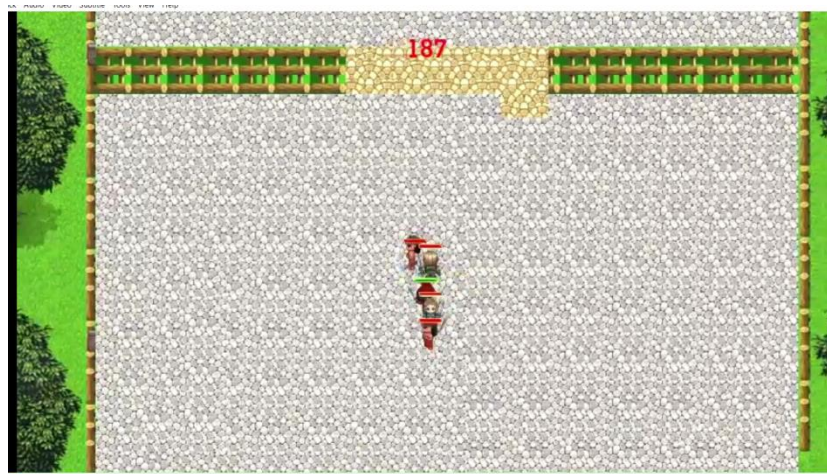
- a. Menentukan Font yang akan digunakan pada asset adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada *asset* aplikasi(lvl5)
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Tittle Sets* sebagai latar atau biasa dikenal keramik pada *asset*

d. Untuk menjalankan program *Survival Level 5* dapat mengatur *object* pada *asset* (Musuh dan sultan) dengan mengatur kontrolnya sebagai berikut:

- Sultan *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
- Musuh *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
- Untuk Menjalan Keseluruhan diatur pada Kontrol *Asset* (obj\_global, obj\_global\_ai, obj\_kontrol\_musuh, obj\_kontrol pengikut, obj\_mask\_serangan).
- Untuk Mengatur Waktu diatur pada *object* Kontrol yaitu obj\_global dengan *event* atur ulang variabel.
- Menggunakan 5 obj ruang (obj ruang adalah bagian batas pergerakan musuh yang diatur pada object yaitu obj\_ruang).

### 23. Tampilan *Survival Level 6*

Pada Tampilan *Survival Level 6* yaitu menampilkan pertempuran dari objek yang telah diprograman sesuai dengan perintah yang diinginkan dengan latar yang sesuai dengan peristiwa alur cerita sebelumnya.



**Gambar 4.53** Tampilan *Survival Level 6*

Pada Gambar 4.53 diatas dirancang dengan menggunakan aplikasi *Game Maker Studio* dengan Bahasa Pemrograman *Game Maker Language* dengan beberapa tindakan sehingga terciptanya tampilan *survival level 6* sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *GeoSlab703 Md BT*
- b. Dibuat dengan *rooms* yang terdapat pada asset aplikasi(lvl6)
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Tittle Sets* sebagai latar atau biasa dikenal keramik pada asset

d. Untuk menjalankan program *Survival Level 6* dapat mengatur *object* pada *asset* (Musuh dan sultan) dengan mengatur kontrolnya sebagai berikut:

- Sultan *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
- Musuh *Events* (aturanimasi, sistem pergerakan, animasi, darah, menghapus, persiapan, dan mengembalikan)
- Untuk Menjalan Keseluruhan diatur pada Kontrol *Asset* (obj\_global, obj\_global\_ai, obj\_kontrol\_musuh, obj\_kontrol pengikut, obj\_mask\_serangan).
- Untuk Mengatur Waktu diatur pada *object* Kontrol yaitu obj\_global dengan *event* atur ulang variabel.
- Menggunakan 6 obj ruang (obj ruang adalah bagian batas pergerakan musuh yang diatur pada object yaitu obj\_ruang).

## D. Implementasi

Implementasi *system* merupakan tahap penerapan dari suatu teknologi yang di desain untuk siap di operasikan. Tahap ini merupakan terjemahan perancangan dari bab hasil analisis sebelumnya dalam suatu Bahasa pemrograman. Bahasa pemrograman yang digunakan dalam membangun sebuah *Game* 2 dimensi sejarah perjuangan sultan Hasanuddin menggunakan *Role Playing Game* berbasis dekstop menggunakan Bahasa pemrograman *Game Maker Language*.

### 1. Kebutuhan Perangkat Keras

Spesifikasi minimum perangkat keras sebagai berikut:

**Tabel 4. 11** *Kebutuhan Perangkat Keras*

| Jenis            | Spesifikasi                                        |
|------------------|----------------------------------------------------|
| <i>Laptop/PC</i> | <i>ACER ASPIRE</i>                                 |
| <i>Processor</i> | AMD Ryzen 5 5500U with Radeon Graphics<br>2.10 GHz |
| Memory           | 8.00 GB                                            |

## 2. Kebutuhan Perangkat Lunak.

Spesifikasi minimum perangkat lunak sebagai berikut:

**Tabel 4. 12 Kebutuhan Perangkat Lunak**

| <b>Jenis</b>          | <b>Spesifikasi</b>                   |
|-----------------------|--------------------------------------|
| <i>Sistem Operasi</i> | <i>Windows 11</i>                    |
| <i>Aplikasi</i>       | <i>Game Maker Studio</i>             |
| <i>Text Editor</i>    | <i>Visual Studio Code Game Maker</i> |

## E. Pengujian Sistem

### 1. *BlackBox*

Pengujian *system* dilakukan dengan cara pengujian *BlackBox*. *Blackox* adalah metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi tanpa mengintip ke dalam struktur atau cara kerja internalnya. Metode pengujian ini dapat diterapkan secara *virtual* ke setiap tingkat pengujian perangkat lunak : unit, integrasi, *system*, dan penerimaan.

**Tabel 4. 13** Hasil *BlackBox test*

| Nama Pengujian                | Tujuan                                                                 | Skenario                                                                          | Hasil Yang diharapkan                                                              | Hasil |
|-------------------------------|------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------|
| Membuka Aplikasi/ <i>Game</i> | Mengetahui aplikasi atau <i>game</i> dapat berjalan tanpa <i>error</i> | Mencoba Membuka aplikasi/ <i>Game</i>                                             | Aplikasi/ <i>Game</i> Dapat berjalan dengan baik, Menampilkan <i>Splash Screen</i> | ✓     |
| Membuka <i>Play</i>           | Agar pemain dapat memulai permainan                                    | Membuka <i>Play</i> , menekantombol <i>space</i> untuk melanjutkan ke pertempuran | Menampilkan Menu Alur cerita dan Survival                                          | ✓     |
| Mode Alur Cerita              | Pemain Dapat bermain dalam mode alur cerita                            | Memilih mode alur Cerita                                                          | Menampilkan Permainan dalam mode alur cerita                                       | ✓     |
| Mode Survival                 | Pemain Dapat bermain dalam mode alur cerita                            | Memilih Mode Survival                                                             | Menampilkan permainan dalam mode Survival                                          | ✓     |
| Membuka Lanjutan              | Agar Pemain dapat melanjutkan permainan yang telah disimpan sebelumnya | Mengklik tombol lanjutan otomatis akan menuju dalam permainan yang telah disimpan | Dapat menjalankan permainan yang telah disimpan                                    | ✓     |
| Membuka Petunjuk              | Agar Pemain dapat melihat atau mengetahui petunjuk permainan           | Membuka tombol untuk menampilkan halaman petunjuk permainan                       | Menampilkan Halaman Petunjuk permainan                                             | ✓     |
| Pause/Jeda                    | Pemain dapat memberhentikan sementara permainan dan dapat              | Menekan tombol <i>esc</i> agar fitur pause berjalan                               | Memberhentikan permainan sementara dengan kata lain pause/jeda                     | ✓     |

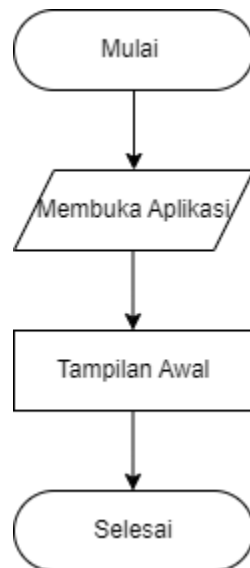
| Nama Pengujian | Tujuan                                                    | Skenario                                                                    | Hasil yang diharapkan                                               | Hasil |
|----------------|-----------------------------------------------------------|-----------------------------------------------------------------------------|---------------------------------------------------------------------|-------|
| Simpan         | Agar pemain dapat menyimpan data teraktif dalam permainan | Menekan tombol esc dan memilih fitur simpan dengan cara mengklik            | Menyimpan data game terakhir dimainkan                              | ✓     |
| Skor           | Pemain dapat menjawab pertanyaan dan mendapatkan nilai    | Menjawab pertanyaan setelah pertempuran telah selesai dan mendapatkan nilai | Menampilkan pertanyaan dan nilai atau skor dari pertanyaan tersebut | ✓     |
| Keluar         | Agar pemain dapat keluar aplikasi dengan benar            | Menekan tombol keluar perangkat                                             | Aplikasi dapat menutup dengan sempurna                              | ✓     |

## 2. *WhiteBox*

Pengujian ini dilakukan dengan cara pengujian *WhiteBox*. *WhiteBox* adalah metode pengujian perangkat lunak yang menguji struktur internal atau cara kerja aplikasi yang bertentangan dengan fungsinya. Dalam pengujian kotak putih, perspektif internal sistem digunakan untuk merancang kasus uji.

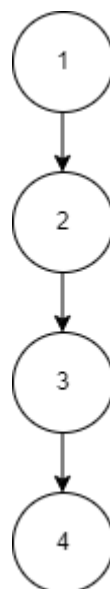
a. *White Box* Tampilan Membuka Aplikasi

1) *Flowchart*



**Gambar 4.54** *flowchart* Membuka Aplikasi

2) *Flowgraph*



**Gambar 4.55** *flowgraph* Membuka Aplikasi

Berdasarkan Gambar 4. 55 di atas dapat dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity*  $V(G)$  pada egde dan node Pada rumus :  $V(G) = E - N + 2$

$$E (\text{edge}) = 3$$

$$N (\text{node}) = 4$$

$$P (\text{Predikat node}) = 0$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 3 - 4 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki Region = 1

(3) *Independent path* pada flowgraph tersebut yakni:

$$\text{Path 1} = 1 - 2 - 3 - 4$$

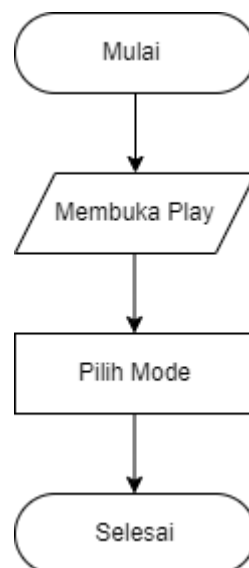
(4) Grafik matriks halaman membuka aplikasi

**Tabel 4. 14** Grafik Matriks Halaman Tampilan Awal

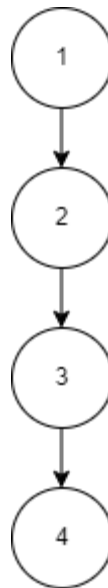
|           | 1 | 2 | 3 | 4 | E-1   |
|-----------|---|---|---|---|-------|
| 1         |   | 1 |   |   |       |
| 2         |   |   | 1 |   |       |
| 3         |   |   |   | 1 |       |
| 4         |   |   |   |   | 0     |
| SUM (E+1) |   |   |   |   | 0+1=1 |

b. *White Box* Tampilan Membuka Play

1) *Flowchart*



**Gambar 4.56** *flowchart* Tampilan Membuka Play

2) *Flowgraph***Gambar 4.57** *flowgraph* Tampilan Membuka *Play*

Berdasarkan Gambar 4. 57 di atas dapat dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity*  $V(G)$  pada *edge* dan *node* Pada rumus :  $V(G) = E - N + 2$

$$E (\text{edge}) = 3$$

$$N (\text{node}) = 4$$

$$P (\text{Predikat node}) = 0$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 3 - 4 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4

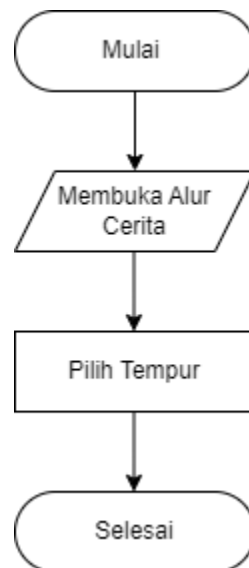
(4) Grafik matriks halaman Membuka *Play*

**Tabel 4. 15** Grafik matriks Membuka *Play*

|           | 1 | 2 | 3 | 4 | E-1   |
|-----------|---|---|---|---|-------|
| 1         |   | 1 |   |   |       |
| 2         |   |   | 1 |   |       |
| 3         |   |   |   | 1 |       |
| 4         |   |   |   |   | 0     |
| SUM (E+1) |   |   |   |   | 0+1=1 |

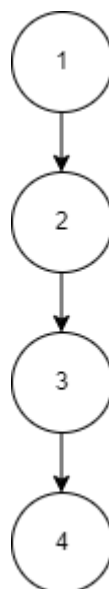
c. *White Box* Tampilan mode alur cerita

1) *Flowchart*



**Gambar 4.58** *flowchart* mode alur cerita

2) *Flowgraph*



**Gambar 4.59** *flowgraph* mode alur cerita

Berdasarkan Gambar 4. 59 di atas dapat dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity*  $V(G)$  pada *edge* dan *node* Pada

rumus :  $V(G) = E - N + 2$

$E$  (*edge*) = 3

$N$  (*node*) = 4

$P$  (Predikat *node*) = 0

Penyelesaian :

$V(G) = E - N + 2$

$= 3 - 4 + 2$

$= 1$

Predikat ( $P$ ) =  $P + 1$

$= 0 + 1$

$= 1$

(2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4

(4) Grafik matriks Mode Alur Cerita

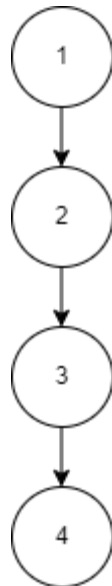
**Tabel 4. 16** Grafik Matriks Mode Alur Cerita

|           | 1 | 2 | 3 | 4 | E-1   |
|-----------|---|---|---|---|-------|
| 1         |   | 1 |   |   |       |
| 2         |   |   | 1 |   |       |
| 3         |   |   |   | 1 |       |
| 4         |   |   |   |   | 0     |
| SUM (E+1) |   |   |   |   | 0+1=1 |

d. *White Box* Tampilan mode *survival*

1) *Flowchart*

**Gambar 4.60** *flowchart mode alur survival*

2) *flowgraph***Gambar 4.61** *flowgraph mode Survival*

Berdasarkan Gambar 4.61 di atas dapat dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity*  $V(G)$  pada *egde* dan *node* Pada

rumus :  $V(G) = E - N + 2$

$E$  (*edge*) = 3

$N$  (*node*) = 4

$P$  (Predikat node) = 0

Penyelesaian :

$V(G) = E - N + 2$

$= 3 - 4 + 2$

$= 1$

Predikat (P) =  $P + 1$

$= 0 + 1$

=1

(2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4

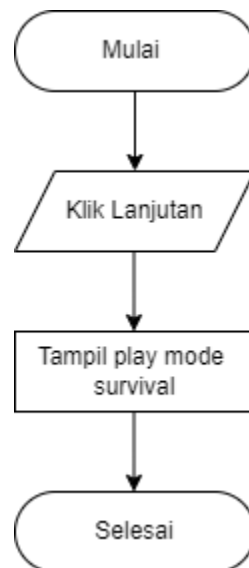
(4) Grafik matriks Mode *Survival*

**Tabel 4. 17** Grafik Matriks Mode *Survival*

|           | 1 | 2 | 3 | 4 | E-1   |
|-----------|---|---|---|---|-------|
| 1         |   | 1 |   |   |       |
| 2         |   |   | 1 |   |       |
| 3         |   |   |   | 1 |       |
| 4         |   |   |   |   | 0     |
| SUM (E+1) |   |   |   |   | 0+1=1 |

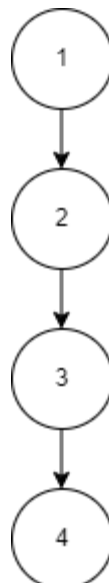
e. *White Box* Tampilan Lanjutan

1) *Flowchart*



**Gambar 4.62** *flowchart* tampilan Lanjutan

2) *Flowgraph*



**Gambar 4.63** *flowgraph* lanjutan

Berdasarkan Gambar 4. 63 di atas dapat dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity*  $V(G)$  pada *edge* dan *node* Pada rumus :  $V(G) = E - N + 2$

$$E (\text{edge}) = 3$$

$$N (\text{node}) = 4$$

$$P (\text{Predikat node}) = 0$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 3 - 4 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

(3) *Independent path* pada *flowgraph* tersebut yakni:

$$\text{Path 1} = 1 - 2 - 3 - 4$$

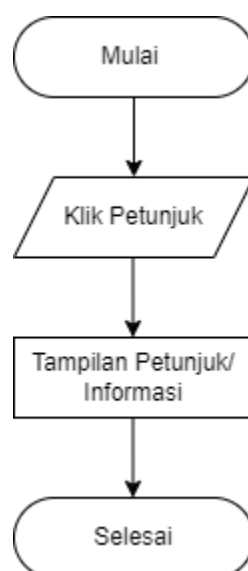
(4) Grafik matriks lanjutan

**Tabel 4. 18** Grafik Matriks Lanjutan

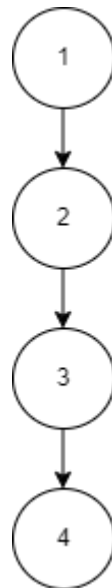
|           | 1 | 2 | 3 | 4 | E-1   |
|-----------|---|---|---|---|-------|
| 1         |   | 1 |   |   |       |
| 2         |   |   | 1 |   |       |
| 3         |   |   |   | 1 |       |
| 4         |   |   |   |   | 0     |
| SUM (E+1) |   |   |   |   | 0+1=1 |

f. *White Box* Petunjuk

1) *Flowchart*

**Gambar 4.64** *flowchart* Petunjuk

## 2) Flowgraph



**Gambar 4.65** *flowgraph* petunjuk

Berdasarkan Gambar 4. 65 di atas dapat dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity*  $V(G)$  pada *edge* dan *node* Pada

rumus :  $V(G) = E - N + 2$

$E$  (*edge*) = 3

$N$  (*node*) = 4

$P$  (Predikat node) = 0

Penyelesaian :

$V(G) = E - N + 2$

$= 3 - 4 + 2$

$= 1$

Predikat (P) = P + 1

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

(3) *Independent path* pada *flowgraph* tersebut yakni:

$$\text{Path 1} = 1 - 2 - 3 - 4$$

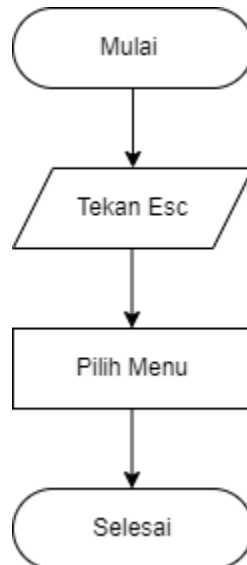
(4) Grafik matriks Petunjuk

**Tabel 4. 19** Grafik Matriks Petunjuk

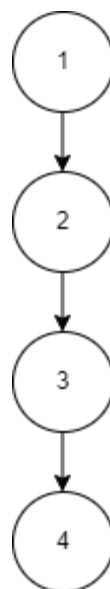
|           | 1 | 2 | 3 | 4 | E-1   |
|-----------|---|---|---|---|-------|
| 1         |   | 1 |   |   |       |
| 2         |   |   | 1 |   |       |
| 3         |   |   |   | 1 |       |
| 4         |   |   |   |   | 0     |
| SUM (E+1) |   |   |   |   | 0+1=1 |

## g. White Box Pause

## 1) Flowchart

**Gambar 4.66** flowchart Pause

## 2) Flowgraph

**Gambar 4.67** flowgraph pause

Berdasarkan Gambar 4. 67 di atas dapat dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity*  $V(G)$  pada *egde* dan *node* Pada

rumus :  $V(G) = E - N + 2$

$E$  (*edge*) = 3

$N$  (*node*) = 4

$P$  (Predikat node) = 0

Penyelesaian :

$V(G) = E - N + 2$

$= 3 - 4 + 2$

$= 1$

Predikat ( $P$ ) =  $P + 1$

$= 0 + 1$

$= 1$

(2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki Region = 1

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4

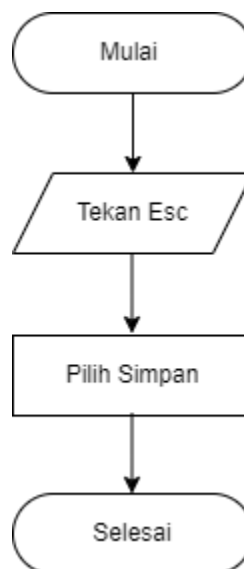
(4) Grafik matriks *Pause*

**Tabel 4. 20** Grafik Matriks *Pause*

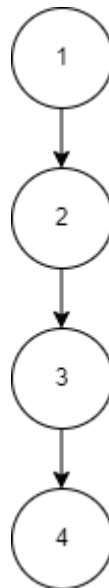
|           | 1 | 2 | 3 | 4 | E-1   |
|-----------|---|---|---|---|-------|
| 1         |   | 1 |   |   |       |
| 2         |   |   | 1 |   |       |
| 3         |   |   |   | 1 |       |
| 4         |   |   |   |   | 0     |
| SUM (E+1) |   |   |   |   | 0+1=1 |

h. *White Box* Simpan

1) *Flowchart*

**Gambar 4.68** *flowchart* Simpan

## 2) Flowgraph



**Gambar 4.69** *flowgraph* Simpan

Berdasarkan Gambar 4. 69 di atas dapat dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity*  $V(G)$  pada *egde* dan *node* Pada rumus :  $V(G) = E - N + 2$

$$E \text{ (edge)} = 3$$

$$N \text{ (node)} = 4$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 3 - 4 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

=1

(2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki Region = 1

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 4

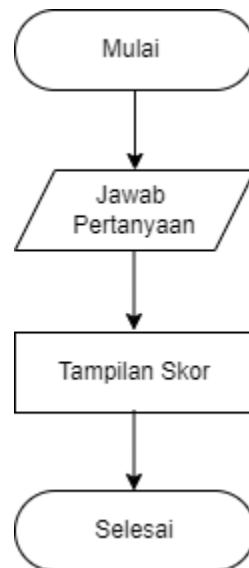
(4) Grafik matriks Simpan

**Tabel 4. 21** Grafik Matriks Simpan

|           | 1 | 2 | 3 | 4 | E-1   |
|-----------|---|---|---|---|-------|
| 1         |   | 1 |   |   |       |
| 2         |   |   | 1 |   |       |
| 3         |   |   |   | 1 |       |
| 4         |   |   |   |   | 0     |
| SUM (E+1) |   |   |   |   | 0+1=1 |

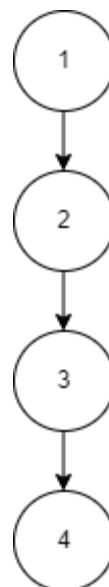
i. *White Box* Skor

1) *Flowchart*



**Gambar 4.70** *flowchart* Skor

2) *Flowgraph*



**Gambar 4.71** *flowgraph* Skor

Berdasarkan Gambar 4. 71 di atas dapat dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity*  $V(G)$  pada *edge* dan *node* Pada rumus :  $V(G) = E - N + 2$

$$E (\text{edge}) = 3$$

$$N (\text{node}) = 4$$

$$P (\text{Predikat node}) = 0$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 3 - 4 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki Region = 1

(3) *Independent path* pada *flowgraph* tersebut yakni:

$$\text{Path 1} = 1 - 2 - 3 - 4$$

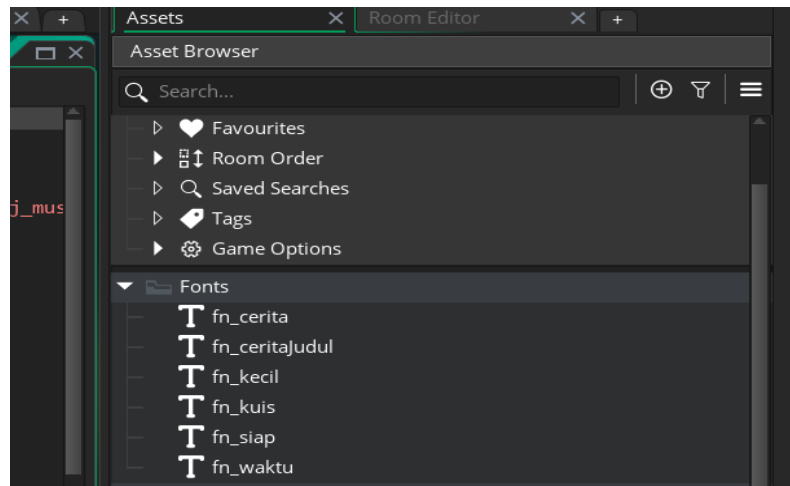
(4) Grafik matriks Skor

**Tabel 4. 22** Grafik Matriks Skor

|           | 1 | 2 | 3 | 4 | E-1   |
|-----------|---|---|---|---|-------|
| 1         |   | 1 |   |   |       |
| 2         |   |   | 1 |   |       |
| 3         |   |   |   | 1 |       |
| 4         |   |   |   |   | 0     |
| SUM (E+1) |   |   |   |   | 0+1=1 |

## F. Tahapan Game Maker Studio

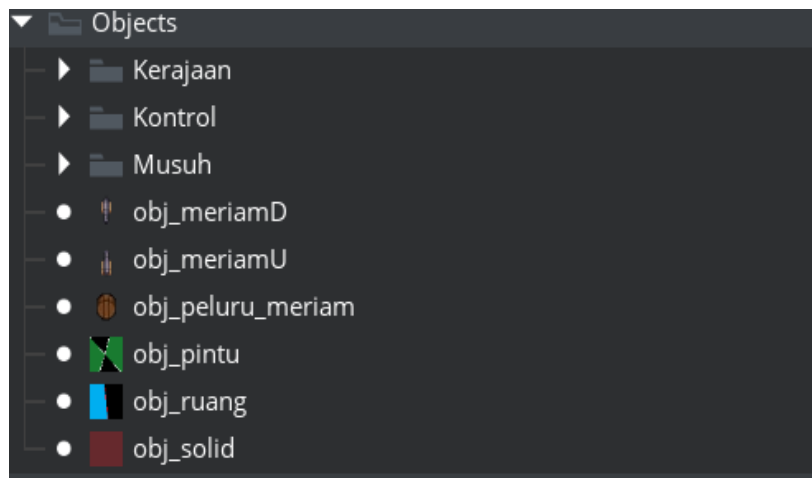
### 1. *Fonts game maker studi*



**Gambar4.72** *fonts game maker studio*

Fonts berfungsi untuk menentukan text atau tulisan yang akan dipakai disetiap room yang dibuat.

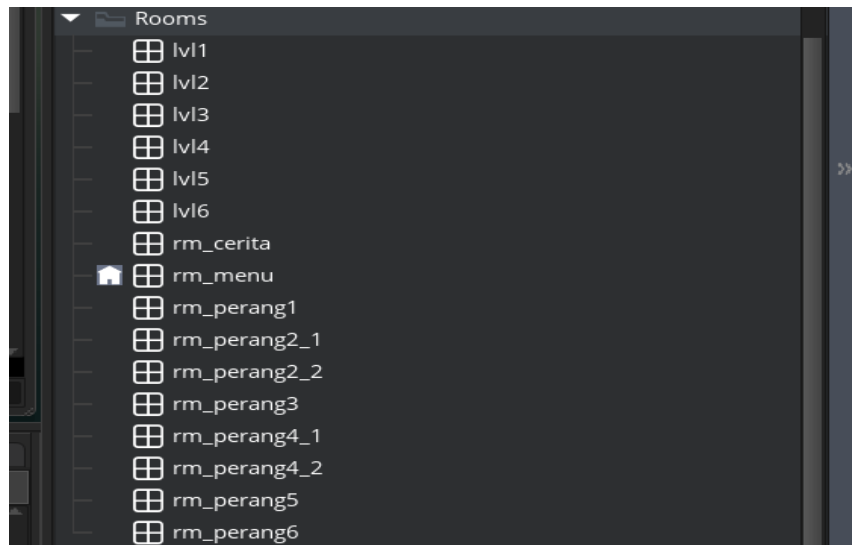
## 2. *Object Game Maker Studio*



**Gambar 4.73** *Object game maker studio*

Fungsi dari *object game maker studio* ini adalah untuk menambahkan beberapa bagian- bagian tertentu seperti pada gambar diatas dan mengontrol batasan – batasan pergerakan pada *object*.

### 3. *Rooms game maker studio*



**Gambar 4.74** *Rooms game maker studio*

*Rooms* pada *game maker studio* berfungsi untuk menentukan implementasi bagian – bagian yang telah di buat sebelumnya seperti map, karakter dan beberapa fitur lainnya.

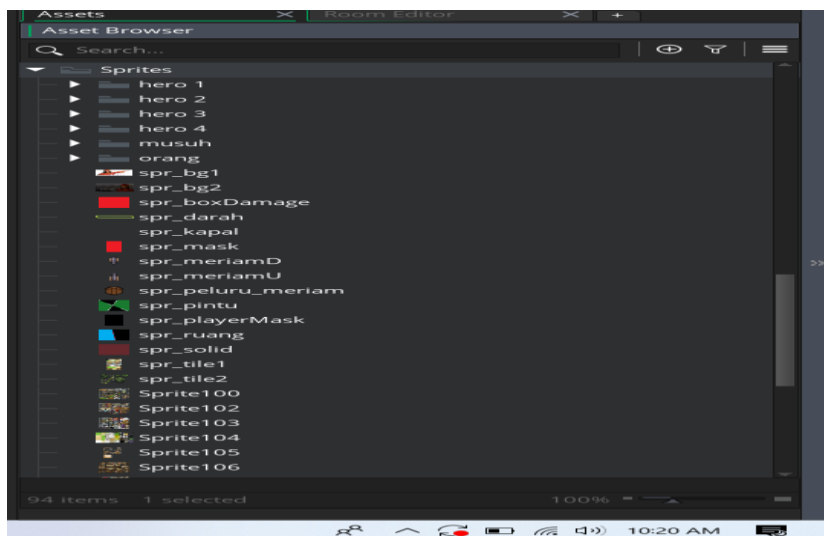
#### 4. Script Game Maker Studio



**Gambar 4.75** Script game maker studio

Script Game Maker Studio Berfungsi untuk menentukan jalannya suatu perintah agar dapat berjalan dengan lancar dan sesuai kemauan menggunakan bahasa pemrograman yang dipakai

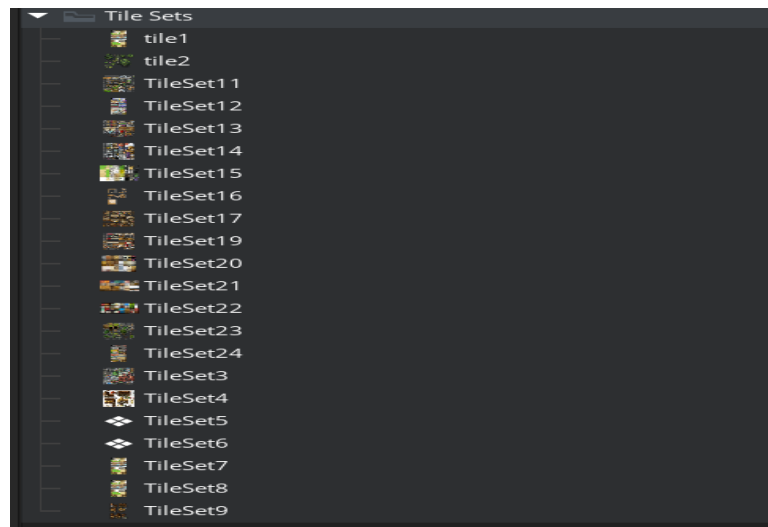
#### 5. Sprites Gamer Maker Studio



**Gambar 4.76** Sprites game maker studio

Sprites pada game maker studio merupakan bagian terpenting dikarenakan bagian ini untuk menentukan suatu objek untuk dijalankan seperti karakter, benda dan bahan pendukung lainnya.

## 6. Tile Sets Gamer Maker studio



**Gambar 4.77** *Tile Sets game maker studio*

*Tile Sets* ini tidak kalah penting dengan *sprite* karena bagian yang menentukan bentuk suatu *room* atau map dengan fungsi untuk mendesain.

## **BAB V**

### **PENUTUP**

#### **A. Kesimpulan**

Kesimpulan dari hasil penelitian dan pembahasan dari pembuatan Game 2 Dimensi Sejarah Perjuangan Sultan Hasanuddin menggunakan Role Playing Game Berbasis Dekstop antara lain:

1. Pembuatan *game* 2 dimensi sejarah perjuangan sultan hasanuddin menggunakan *engine role playing game* sebagai *software* utama.
2. Dengan adanya *game* ini diharapkan membantu dalam mengenal sejarah Indonesia yaitu Sejarah Perjuangan Sultan Hasanuddin melalui *Game*.

#### **B. Saran**

Pada penelitian ini penulis menyadari bahwa masih ada beberapa kekurangan yang sangat diperlukan perbaikan dan pengembangan di penelitian selanjutnya. Oleh karena itu, penulis memiliki beberapa saran untuk pengembangan selanjutnya, sebagai berikut:

1. Penelitian selanjutnya perlu menambah jumlah level pada mode *survival* agar lebih menarik.
2. *Game* yang penulis buat masih sederhana, terutama penampilan grafis dan animasi, sehingga untuk pengembangan berikutnya dapat diperbaharui tampilan lebih menarik dan lebih enteraktif

## DAFTAR PUSTAKA

- Adi Nugroho. 2009. *Rekayasa Perangkat Lunak Menggunakan UML Dan Java* Yogyakarta : Andi
- Basuki, Sulistyo. *Metode Penelitian*. (Jakarta: Penaku, 2010).
- Enci' Amin, *Syair Perang Mengkasar* (Makassar: Innawa, 2008).
- Ford, Jerry (2009). *Getting Started with Game Maker*.
- Hasnawati, & Asriadi. (2022). *JURNAL SINTAKS LOGIKA APLIKASI GAME MEMASAK KUE TRADISIONAL BUGIS MENGGUNAKAN CONSTRUCT 2 BERBASIS ANDROID* Informasi Artikel (Vol. 2, Issue 2).  
<https://jurnal.umpar.ac.id/index.php/sylog> 55
- Lidya dan Jhon.(2021).*Perancangan Game Edukasi Sejarah Kemerdekaan Indonesia Menggunakan Aplikasi Construct 2 Berbasis Android*.Universitas Citra Bangsa.
- Maswadi Rauf, *Konsensus dan Konflik Politik* (Jakarta: Dirjen Pendidikan Tinggi Depdiknas, 2000).
- Muh. Amir, dkk, *Pahlawan Nasional Dari Sulawesi Selatan* (Makassar:Dinas Kebudayaan dan Pariwisata Propinsi Sulawesi Selatan, 2008).
- Munir, (2013). *MULTIMEDIA dan Konsep Aplikasi Dalam Pendidikan*. Bandung: Penerbit Alfabeta
- (Noname). Introduction Tutorials Game Maker Language. Diakses 16 April 2013.  
[http:// www.gamemakerlanguage.herokuapp.com](http://www.gamemakerlanguage.herokuapp.com)
- (Noname). (2013). Key Lime Pie, OS Android Terbaru. Diakses 10 Mei 2013.  
<http://m.neraca.co.id/harian/article/26532/Key.Lime.Pie.OS.Android>.  
Terbaru
- Putra ,Febriyanto Pratama. 2012. “ *Pembuatan Game Animasi 3D Role Playing Game Untuk Pendidikan Budaya Dengan Unity3D dan Bahasa Pemrograman C#*.”.Program Studi Teknik Informatika. Universitas Muhammadiyah Surakarta.Surakarta
- Ramdhany, T., Bas, I., Pahrilah, D., & Krisdiawan, R. A. (2021). *PEMBUATAN GAME EDUKASI SEJARAH KERAJAAN SRIWIJAYA MENGGUNAKAN RPG MAKER MV*. 15. <https://journal.uniku.ac.id/index.php/ilkom>
- Risca Amelia. (2020) .*Game Edukasi Dan Cerita Interaktif Sejarah Kerajaan Di Sumatra Menggunakan Algoritma Fuzzy Sugeno Untuk Mengatur Perilaku NPC*”Program Studi Informatika,Universitas Teknokrat Indonesia

Rekaharta, P. (2008). ROLE PLAY GAME SAYEMBARA KUMALA DEWI Oleh. In *Agustus* (Vol. 1, Issue 2).

Tri dan Rio. (2021). *Pembuatan Game Edukasi Sejarah Kerajaan Sriwijaya Menggunakan RPG Maker MV*”.Program Studi informatika,Universitas Kuningan