

BAB I

PENDAHULUAN

A. Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi kian pesat, hal tersebut dapat dirasakan di sekeliling kita secara langsung maupun tidak langsung. Perkembangan tersebut tengah berdampak pada segala aspek kehidupan manusia. Salah satu jenis perkembangan ilmu pengetahuan dan teknologi adalah perkembangan pada dunia game. Game merupakan salah satu pemanfaatan teknologi yang sangat banyak diminati oleh kalangan tua, muda dan anak yang pada umumnya untuk memperoleh kesenangan dan mengisi waktu kejenuhan semata, namun dengan kemajuan teknologi saat ini berbagai macam game telah tercipta tidak hanya untuk tujuan kesenangan, namun dapat dimanfaatkan untuk media pembelajaran untuk peningkatan mutu pendidikan.

Mengetik merupakan kegiatan mencatat yang dilakukan dengan bantuan papan ketik. Mengetik bisa dilakukan dengan mesin tik atau dengan keyboard pada komputer. Mengetik pada komputer merupakan pengoperasian paling lumrah yang dilakukan oleh para penggunanya. Tindakan tersebut diwujudkan dalam proses menekan pada keyboard yang bertujuan untuk memasukkan angka atau huruf sehingga tampil di layar monitor. Jika mengetik dilakukan tanpa disertai dengan pengetahuan serta keterampilan, bisa jadi ketikan yang dihasilkan tidak memuaskan bahkan cenderung akan terjadi salah ketik. Keterampilan mengetik sepuluh jari

sangat membantu dalam melakukan pekerjaan yang berhubungan dengan komputer, misalnya penulis, programmer atau sekretaris.

Konsep gim yang digunakan penelitian ini adalah “*Typing*” dimana musuh yang datang dapat dikalahkan dengan mengetikkan kalimat yang muncul. *Genre* gim jenis ini dipilih karena memiliki kontribusi cukup besar untuk membantu seseorang untuk terbiasa mengetik tanpa harus benar-benar belajar secara serius. Platform gim ini kelak dapat dimainkan di PC (*Personal Computer*).

Rata-rata kecepatan mengetik orang dewasa (yang tergolong pemula) adalah 30 WPM, sementara kecepatan rata-rata mengetik untuk mereka yang cukup sering menggunakan kibor (intermediate) adalah 40 WPM. Masalahnya, kecepatan minimum mengetik untuk bekerja secara efisien setidaknya harus mencapai 40. Terlebih untuk tingkat profesional, syarat minimumnya tentu lebih tinggi lagi.

Berdasarkan hal diatas, perlu adanya media untuk melatih kecepatan mengetik yang nantinya akan memudahkan user untuk terbiasa mengetik dengan cepat dan akan berpengaruh terhadap dunia pekerjaan. Sehingga berdasarkan hal diatas, peneliti membuat judul penelitian “APLIKASI GAME MENGETIK CEPAT SEBAGAI SARANA MELATIH KECEPATAN MENGETIK BERBASIS DESKTOP”.

B. Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat dirumuskan permasalahannya yaitu bagaimana membuat Aplikasi Game Mengetik Cepat Sebagai Sarana Melatih Kecepatan Mengetik Berbasis Desktop.

C. Tujuan Penelitian

Tujuan yang akan dicapai dalam penelitian ini adalah mengembangkan Aplikasi Game Mengetik Cepat Berbasis Desktop ini dapat membantu melatih kecepatan mengetik.

D. Batasan Masalah

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

1. Aplikasi ini hanya dapat dimainkan secara *single player* dan offline.
2. Aplikasi ini dibuat menggunakan gambar 2 dimensi.
3. Aplikasi ini tidak dapat menyimpan score.
4. Aplikasi ini hanya bisa dimainkan oleh remaja keatas.
5. Aplikasi ini hanya dapat dimainkan di desktop

E. Manfaat Penelitian

Manfaat penelitian dari tugas akhir ini adalah untuk memberikan alternatif hiburan yang dapat menghilangkan kejenuhan serta melatih kecepatan tangan mengetik dalam bentuk *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 menguraikan tentang teori-teori yang menyangkut penelitian ini.

3. BAB III METODE PENELITIAN

Bab ini diuraikan mengenai jenis penelitian, lokasi dan waktu penelitian, parameter pengujian, tahap-tahap metode penelitian, dan bagan alir penelitian.

4. BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas tentang detail Aplikasi Game Mengetk Cepat Berbasis Desktop.

5. BAB V PENUTUP

Bab ini membahas tentang Kesimpulan dan Saran.

BAB II

TINJAUAN PUSTAKA

A. Kajian Hasil Penelitian Terdahulu

1. Game Edukasi Berbasis *Role Playing Game* Dengan Metode *Finite State Machine*

Peneliti (Abdullah, Reza, & Erliana) dalam penelitian yang berjudul “Game Edukasi Berbasis Role Playing Game Dengan Metode *Finite State Machine*” Metode penelitian yang diterapkan menggunakan metode *Finite State Machine* pada game yang berbasis *role playing game*, dengan menggunakan *software delphi* dan menggunakan metode pengembangan multimedia (*Mutimedia Methodology*). Hasil penelitian menunjukkan bahwa game ini dapat dijadikan media pembelajaran bagi anak dan dapat belajar sambil bermain sehingga dapat menambah ilmu pengetahuan di bidang mengetik.

2. Pengembangan Gim Edukasi Mengetik menggunakan Metode Touch-Typing untuk Meningkatkan Kecepatan Mengetik

Peneliti (Pamengas, Afirianto, & Wardhono S, 2020) dalam penelitiannya yang berjudul “Pengembangan Gim Edukasi Mengetik menggunakan Metode Touch-Typing untuk Meningkatkan Kecepatan Mengetik” menggunakan metode pembelajaran latihan (*drill*)

3. Aplikasi Pembelajaran Mengetik Cepat Berbasis Game Edukasi Dengan Linear Congruent Method (LCM)

Peneliti (Setiyadi, 2016) dalam penelitiannya yang berjudul “Aplikasi Pembelajaran Mengetik Cepat Berbasis Game Edukasi Dengan Linear Congruent Method (LCM)” Metode yang digunakan adalah Linear Congruent Method (LCM).

B. Game

Game menurut (Yudhanto, 2010) adalah permainan yang menggunakan media elektronik, merupakan sebuah hiburan berbentuk multimedia yang dibuat semenarik mungkin agar pemain bisa mendapatkan sesuatu sehingga adanya kepuasan batin.

Game mempunyai potensi yang sangat besar dalam membangun motivasi pada proses pembelajaran. Berbeda dengan pada penerapan metode konvensional, untuk menciptakan motivasi belajar sebesar motivasi dalam *game*, dibutuhkan seorang guru/instruktur yang berkompeten dalam pengelolaan proses pembelajaran.

a. Elemen Game

Di dalam sebuah game terdapat beberapa elemen-elemen yang akan membentuk sebuah game itu sendiri, antara lain :

1) Aturan-aturan

Sebuah game mengambil tempat atau setting di dunia buatan yang diatur oleh aturan-aturan. Aturan inilah yang akan menentukan aksi dan gerakan pemain dalam sebuah game.

2) Pengaturan

Sebuah game yang dibuat akan mengambil tempat atau setting di dunia tertentu. Contohnya pada game petualangan yang bersetting pada sebuah hutan atau kota dengan batasan-batasan tertentu.

3) Perspective (sudut pandang).

Perspective disini yaitu menjelaskan bagaimana pemain melihat gameworld dari suatu game pada layar side view atau melihat dari sisi samping.

4) Peran

Peran disini merupakan tokoh yang dimainkan oleh pemain dalam suatu game. Dengan adanya peran ini maka pemain akan lebih mudah untuk memahami tujuan apa yang sebenarnya ingin dicapai dan aturan apa yang dimainkan pada game tersebut.

5) Struktur

Hubungan antara mode dan aturan menentukan kapan dan mengapa game berubah secara bersama-sama membentuk struktur permainan.

6) Realisme

Game menggambarkan sebuah dunia, bahkan mungkin dunia khayal. Sebuah game yang menerapkan akal sehat dan logika pada aturan permainannya dapat dikatakan sebagai game yang realistis.

7) Cerita

Game komputer merupakan perpaduan antara media pasif, naratif seperti televisi dan film, dan media aktif, non-aktif seperti permainan poker dan domino. Beberapa game komputer seperti tetris tidaklah mempunyai cerita. Lain halnya seperti game Final Fantasy. Game tersebut mempunyai alur cerita yang linear atau hanya satu jalan cerita dan bersifat non-interaktif. Namun ada juga yang bersifat interaktif dimana cerita akan berbeda atau bercabang tergantung pada pilihan atau tindakan yang diambil pemain. Sehingga ending yang didapat akan berbeda-beda yang biasa disebut dengan multiple ending.

b. Jenis Game

Menurut (Henry, 2010) format sebuah *game* bisa murni sebuah *genre* atau bisa merupakan campuran (*hybrid*) dari beberapa *genre* lain. Game yang penulis buat juga merupakan campuran dari beberapa *genre*. Jenis-jenis *game* yang ada

Jenis Game :

1. *Maze Game*

Jenis *game* ini biasanya menggunakan *maze* sebagai setting atau latar game. Jenis *game maze* ini termasuk jenis *game* yang paling awal muncul. Contoh *game* ini adalah *game Pacman* dan *Digger*.

2. *Board Game*

Game jenis ini sama dengan *game board* tradisional seperti monopoli. Hanya saja permainan tradisional ini dimainkan melalui komputer.

3. *Card Game*

Jenis *game* kartu juga tidak jauh berbeda dari *game* tradisional aslinya. Namun, tampilannya lebih bervariasi dari versi tradisional. *Game* ini juga termasuk *game* yang awal muncul. Contoh *game* ini adalah *Solitaire* dan *Hearts*.

4. *Battle Card Game*

Contoh *game* ini yang populer yaitu *Battle Card Pokemon*. *Game* ini jarang ditemukan di Indonesia. Film kartun yang bercerita tentang permainan *battle card* ini pernah ditayangkan di stasiun televisi Indonesia.

5. *Quiz Game*

Game jenis ini merupakan *game* dengan bentuk kuis. Contoh *Quiz Game* yang pernah beredar yaitu *game* kuis *Who Wants to Be Millionaire*.

6. *Puzzle Game*

Jenis *game* ini memberi tantangan dengan cara menjatuhkan atau melenyapkan sesuatu dari sisi atas ke bawah atau dari kiri ke kanan. Contoh *game* ini adalah Tetris.

7. *Side Scroller Game*

Saat pertama kali muncul *game* ini berbentuk 2D. Sekarang sudah banyak yang dibuat dengan efek 3D. Pada *game* jenis ini pemain diharuskan bergerak searah di alur yang disediakan. Dia diharuskan untuk berjalan, melompat, merunduk serta menghindari rintangan-

rintangan. Contoh game ini yang populer yaitu Mario Bros dan Prince of Persia.

8. *Fighting Game*

Jenis *game* ini sesuai dengan namanya berisi tentang pertarungan. Contoh game ini yaitu *Street Fighter*, *Samurai Showdown*, *Virtual Fighter* dan Kungfu.

9. *Racing Game*

Racing game adalah *game* tentang balapan. Contoh *game* ini yaitu *Need for Speed Underground* dan *Toca RaceDriver*.

10. *Turn-Based Strategy Game*

Pemain dalam *game* ini melakukan gerakan setelah pemain lain melakukan gerakan jadi saling bergantian. Contoh *game* yang terkenal adalah *Empire* dan *Civilization*.

11. *Real-Time Strategy Game*

Game ini seperti *game Turn- Based Strategy* (RTS), namun pada *game* ini pemain tidak perlu menunggu pemain lain. Pemain tercepatlah yang akan menang. Contoh game ini yaitu *Warcraft*.

12. *SIM*

Game genre ini merupakan bentuk permainan simulasi. Di sini pemain membangun sebuah area, kota, negara atau koloni. Contoh *game* ini yaitu *Ship Simulator*, *Train Simulator*, dan *Crane Simulator*.

13. *First Person Shooter*

Disebut *first person shooter* karena pandangan pemain adalah pandangan orang pertama. Banyak baku tembak dan *game* ini mengutamakan kecepatan gerakan. Contoh *game* ini yaitu *game Counterstrike* dan *Doom*.

14. *Role Playing Game*

Jenis *game* ini pemainnya memainkan sebuah tokoh atau karakter. Biasanya ada alur cerita yang harus dijalankan. Contoh *game* ini adalah *Legacy*

c. Teori Pembuatan *Game*

Urutan pengembangan *game* tercakup dalam teori pembuatan *game* yang dibagi menjadi tiga bagian, antara lain :

- 1) Tentukan tipe permainan apa yang ingin dibuat Penentuan ini sebagai dasar sebelum mulai bekerja sampai mendapatkan ide yang bagus untuk dibuat program permainan.
- 2) Definisikan model permainan dan tujuannya. Ide yang didapat itu dituangkan dalam bentuk model permainan yang dibuat. Pada tahap ini sebaiknya model permainannya dituliskan secara jelas sehingga jika hendak menambah cerita, karakter lain, atau suatu aksi baru, maka bisa dilihat atau dicek dari yang sudah dituliskan sehingga permainannya tetap konsisten dan tidak membingungkan.

C. Pengertian Mengetik

Menurut Marimin, dkk (2012:1) menyatakan bahwa “mengetik adalah pekerjaan yang mendasar terdapat pada semua bidang, baik itu dalam organisasi swasta, organisasi pemerintah atau organisasi kepartaian maupun organisasi yang lain”. Kompetensi mengetik yang diterapkan di SMK pada umumnya menggunakan mesin ketik manual sebagai media belajar.

Mengetik merupakan sebuah teknik mengenai penggunaan komputer yang harus dipelajari dan dilatih. Jika mengetik dilakukan tanpa disertai dengan pengetahuan serta keterampilan, maka boleh jadi ketikan yang dihasilkan tidak akan memuaskan dan berpotensi timbul kesalahan.

Jika ditelusuri secara seksama, asal mula kata mengetik sebenarnya adalah *onomatope* atau tiruan bunyi dari bunyi *tuts* atau tombol yang ditekan. Bunyi tik yang dihasilkan adalah suara dari *tuts* pada mesin yang memiliki *tuts* atau tombol yang bisa ditekan. Jadi, secara *onomatope* mengetik itu adalah mengetuk atau menekan dengan ujung jari yang menghasilkan bunyi tik-tik.

a. Metode Pengetikan

Mengetik juga memiliki beberapa metode yang bisa digunakan agar mempermudah proses pengetikan. Metode-metode ini baru bisa dilakukan jika kemampuan dalam mengetik sudah terbilang mahir. Berikut ini adalah beberapa metode pengetikan yang biasa digunakan :

1) Mengetik Sistem Buta

Mengetik dengan sistem buta adalah metode pengetikan yang biasa digunakan ketika akan menyalin naskah ke dalam komputer atau mesin tik. Ketika menyalin naskah tersebut, pandangan mata hanya tertuju pada naskah, sama sekali tidak melihat ke arah layar komputer atau *tuts*. Metode ini bias dikatakan hanya mengandalkan perasaan.

2) Mengetik Sistem Berirama

Suara tik yang ditimbulkan ketika menekan *tuts* menjadi hal pokok dalam sistem ini. Bunyi yang dihasilkan oleh tekanan-tekanan *tuts* harus memiliki ketukan yang sama sehingga suara hentakan yang dihasilkan menghasilkan irama. Kedua metode pengetikan itu hanya dapat dilakukan jika metode mengetik sepuluh jari sudah mahir dilakukan.

3) Mengetik Sepuluh Jari

Pada metode ini, sepuluh jari sudah memiliki peranan dalam menekan *tuts* masing-masing. Metode seperti ini harus dilatih agar jari dapat menekan *tuts* tanpa ragu-ragu. Gerakan jari harus dilakukan dengan gerakan yang teratur sehingga *tuts* akan terlihat seperti tertekan secara otomatis. Berhasil atau tidaknya metode ini sangat mengandalkan perasaan. Perasaan dapat diasah dengan latihan yang rutin.

b. Pengertian Mengetik Sepuluh Jari

Dalam bahasa Inggris, mengetik sepuluh Jari disebut juga dengan istilah *Touch Typing*, artinya mengetik tanpa menggunakan indra penglihatan untuk mencari *tuts keyboard* yang diinginkan. Secara spesifik, pengetik atau disebut juga dengan *typist* mengetahui lokasi *tuts keyboard* lewat memori otot (*muscle memory*).

Memori otot dikenal sebagai *motor learning*, artinya suatu bentuk prosedur memori yang melibatkan konsolidasi tugas motorik untuk masuk ke dalam memori melalui pengulangan gerakan otot. Ketika dilakukan gerakan berulang dari waktu ke waktu, memori otot jangka panjang diberi tugas tertentu sehingga akhirnya memungkinkan untuk melakukan suatu tugas tersebut tanpa upaya sadar. Proses ini mengurangi kebutuhan untuk perhatian dan menciptakan efisiensi maksimum dalam sistem motor dan memori. Contoh memori otot yang ditemukan dalam aktivitas sehari-hari banyak yang menjadi otomatis dan meningkat secara praktik, seperti makan, mengetik pada keyboard, naik sepeda, memainkan alat musik, bermain *video game*, atau bahkan memecahkan teka-teki kubus.

Pengetik sepuluh jari menyimpan delapan jarinya di sepanjang *tuts horizontal* bagian tengah *keyboard (the home row)* dan menggunakan keduapuluh jari tersebut untuk meraih *tuts* lainnya yang berada di barisan keyboard bagian atas dan bawah. Dua jari lagi, yaitu jari jempol, digunakan untuk menekan tombol spasi. Akan tetapi, sebagian besar pengetik menggunakan jempol tangan kanan untuk menekan tombol spasi. Orang yang mengetik dengan sepuluh jari biasa disebut

copy typist atau *audio typist*. Terdapat dua keahlian yang harus dimiliki *copy typist*. Yang pertama adalah mengetik dengan cepat dan yang kedua adalah mengetik dengan akurat. Kecepatan mengetik sepuluh biasanya diukur dalam dua skala, yaitu CPM (*Characters Per Minute*) dan WPM (*Words Per Minute*). CPM adalah jumlah karakter benar yang diketik dalam satu menit sedangkan WPM ialah jumlah kata, standarnya lima huruf, yang diketik dalam satu menit. WPM bisa didapat dari CPM dibagi lima

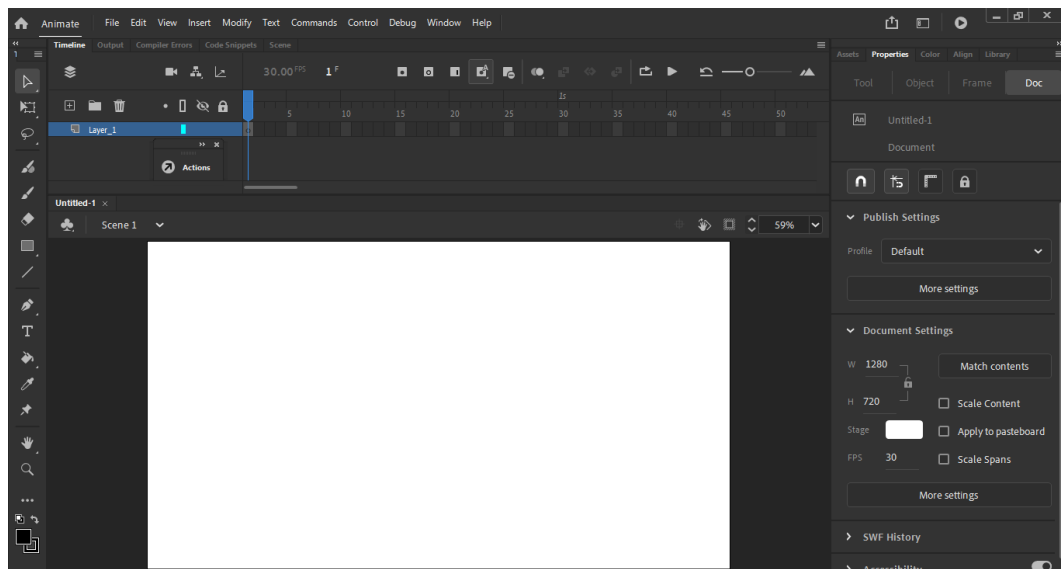
Di dalam pengetikan, ada aspek lain yang berhubungan dengan jari, yaitu waktu rata-rata respon. Waktu rata-rata jari merespon diukur dalam satuan *millisecond* (ms) atau seper-ribu detik. Waktu tersebut dihitung dari awal pengetikan atau karakter yang diketik sampai karakter (yang bukan salah ketik) setelahnya yang diketik. Untuk waktu rata-ratanya didapat dari jumlah waktu suatu karakter yang benar diketik dibagi jumlah karakter ketikan itu.

D. Adobe Animate

Software yang digunakan adalah Adobe Animate yang merupakan versi paling baru dari Adobe Flash. Adobe Animate adalah aplikasi desain grafis yang sering digunakan oleh desainer-desainer untuk membuat karya profesional, khususnya dalam bidang animasi (Hanafri, 2015). Adobe animate merupakan pengembangan dari adobe flash profesional, macromedia flash, dan futures plash animator adalah 17 program multimedia authoring dan animasi komputer yang dikembangkan oleh adobe systems. Program grafis dan animasi yang keberadaannya ditujukan bagi pecinta desain dan animasi untuk berkreasi membuat animasi (Ahmadi, 2018:30).

Adobe Animate dapat beroperasi pada Microsoft Windows dan Mac Os, serta produk yang dihasilkan oleh Adobe Animate dapat beroperasi pada system Windows, Mac, Xbox 360, iPad, iPhone, dan Android.

a. Tampilan dan Fitur Adobe Animate



Gambar 2. 1 Tampilan Workspace Adobe Animate

b. Fitur Adobe Animate

(MyEduSolve, 2024) Adobe Animate memiliki sejumlah fitur dasar yang membedakannya dari program Adobe lainnya. Beberapa fitur tersebut diantaranya:

1. *Assets Panel*

Fitur panel ini memiliki fungsi yang sangat bermanfaat dalam mengelola, mengatur, dan menemukan dokumen dari proyek-proyek yang pernah kamu kerjakan sebelumnya. Tak hanya dokumen lama, kamu juga bisa memulai proyek yang baru dengan *Assets Panel* ini. Fitur ini memungkinkan kamu untuk menyusun semua aset yang

digunakan dalam animasi, seperti gambar, suara, video, dan simbol, serta dengan mudah menambahkan aset baru ke dalam dokumen yang kamu kerjakan.

2. *Advanced Rigging*

Fitur ini berfungsi untuk menghubungkan bagian-bagian tubuh karakter animasi yang telah kamu buat dengan cepat dan efektif. *Advanced rigging* membantu kamu dalam mengendalikan gerakan animasi secara lebih efisien, seperti melakukan gerakan realistis tanpa harus membuat setiap *frame* secara manual.

3. *Bone Tool Enhancements*

Fitur selanjutnya adalah *Bone Tool Enhancements*. Fitur ini dapat membantu kamu dalam mengatur pergerakan objek menggunakan *tab Object* di panel properti. *Bone Tool Enhancements* juga berperan untuk membagi dan menggabungkan rentang *armature* saat karakter yang dibuat memiliki jumlah node yang sama.

4. *Quick Publish*

Sesuai dengan namanya, fitur ini memberikanmu kemudahan untuk mempublikasi animasi yang telah kamu buat ke dalam format video, animasi GIF atau kanvas HTML5. Kamu bisa mengakses fitur ini melalui menu **File > Export**.

5. *Timeline dan Symbol Enhancements*

Fitur ini menyediakan berbagai alat yang memungkinkan kamu untuk menyesuaikan dan mengelola *timeline* secara lebih efisien. Pada *Timeline & Symbol Enhancements*, tersedia beberapa pilihan *tools* yang memiliki spesifikasi fungsi berbeda-beda. Beberapa *tools* tersebut adalah *Customize Timeline Tools*, *Break Apart Symbol to Layers*, *Copy paste Tween Settings*, *Reverse play and loop support for Graphic Symbols*. Dengan adanya fitur ini, kamu dapat lebih mudah dan fleksibel dalam membuat dan mengedit animasi di Adobe Animate.

6. Adobe Animate Camera

Adobe Animate juga dilengkapi dengan fitur kamera. Kamu dapat menciptakan efek kamera realistis dalam animasi yang kamu buat, seperti memperbesar atau memperkecil objek dan menerapkan efek warna atau filter. Fungsi kamera pada Adobe Animate adalah mengatur stimulasi gerakan kamera yang dapat meningkatkan visual animasi. Fitur kamera ini tersedia untuk semua jenis dokumen di Animate, termasuk HTML Canvas, WebGL, dan Actionscript.

7. *Transformation Anchor*

Transformasi anchor adalah fitur yang digunakan untuk mengubah skala, rotasi, atau kemiringan objek yang akan kamu pilih yang dapat disesuaikan dengan keinginanmu. Fitur ini akan

mempengaruhi cara objek berubah ketika kamu mengubah ukurannya atau memutarinya.

c. Kelebihan Adobe Animate

1. Terintegrasi penuh dengan *Creative Cloud*.
2. Mendukung untuk beragam format, termasuk HTML5, WebGL, SVG animasi, Sprite Sheets, Flash Player (SWF), dan Adobe AIR.
3. Mendukung berbagai platform, seperti *desktop*, *tablet*, dan juga *mobile*.
4. Adanya dukungan ActionScript.
5. Mendukung untuk ditampilkan ke berbagai media mulai dari DVD, CD-ROM, Web, hingga televisi.
6. Mudah digunakan oleh pemula dan profesional, serta memiliki fitur lengkap.

d. Kekurangan Adobe Animate

1. Terbatasnya *template* siap pakai yang disediakan.
2. Tidak terlalu mendukung untuk pembuatan animasi 3D, lebih cocok untuk animasi 2D.
3. Adanya beberapa menu yang tidak *user-friendly*.
4. Adobe Animate memiliki harga yang cukup tinggi dibandingkan dengan produk sejenis

e. Action Script

Action Script adalah bahasa pemrograman berorientasi objek yang awalnya dikembangkan oleh Macromedia Inc. (kemudian diakuisisi oleh Adobe Systems).

Ini adalah turunan dari HyperTalk, bahasa *scripting* untuk HyperCard. Ini sekarang adalah dialek ECMAScript (yang berarti *superset* dari sintaks dan semantik bahasa lebih dikenal sebagai JavaScript), meskipun awalnya muncul sebagai saudara kandung, keduanya dipengaruhi oleh HyperTalk. *Action Script* digunakan terutama untuk pengembangan situs web dan perangkat lunak yang menargetkan *platform* Adobe *Flash Player*, yang digunakan pada halaman Web dalam bentuk *file* SWF yang disematkan. *Action Script* 3 juga digunakan dengan sistem Adobe AIR untuk pengembangan aplikasi desktop dan seluler. Bahasa itu sendiri adalah open- source yang spesifikasinya ditawarkan secara gratis dan kedua kompilernya open source (sebagai bagian dari Apache Flex) dan mesin virtual open source (Mozilla Tamarin) tersedia. *ActionScript* juga digunakan dengan *Scaleform* GfX untuk pengembangan antarmuka pengguna 3D *video game* dan HUDs

E. UML (Unified Modeling Language)

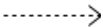
Menurut (Ansori, 2020) UML atau “Unified Modelling Language” adalah suatu metode permodelan secara visual yang berfungsi sebagai sarana perancangan sistem berorientasi objek.

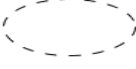
Definisi UML adalah sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan, dan juga pendokumentasian sistem aplikasi. Saat ini UML menjadi bahasa standar dalam penulisan blue print software (arsitektur).

Dalam UML sendiri terdapat beberapa diagram yaitu :

a. Use Case Diagram

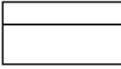
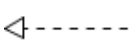
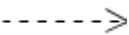
Tabel 2. 1 Simbol-Simbol Use Case Diagram

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.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

NO	GAMBAR	NAMA	KETERANGAN
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
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

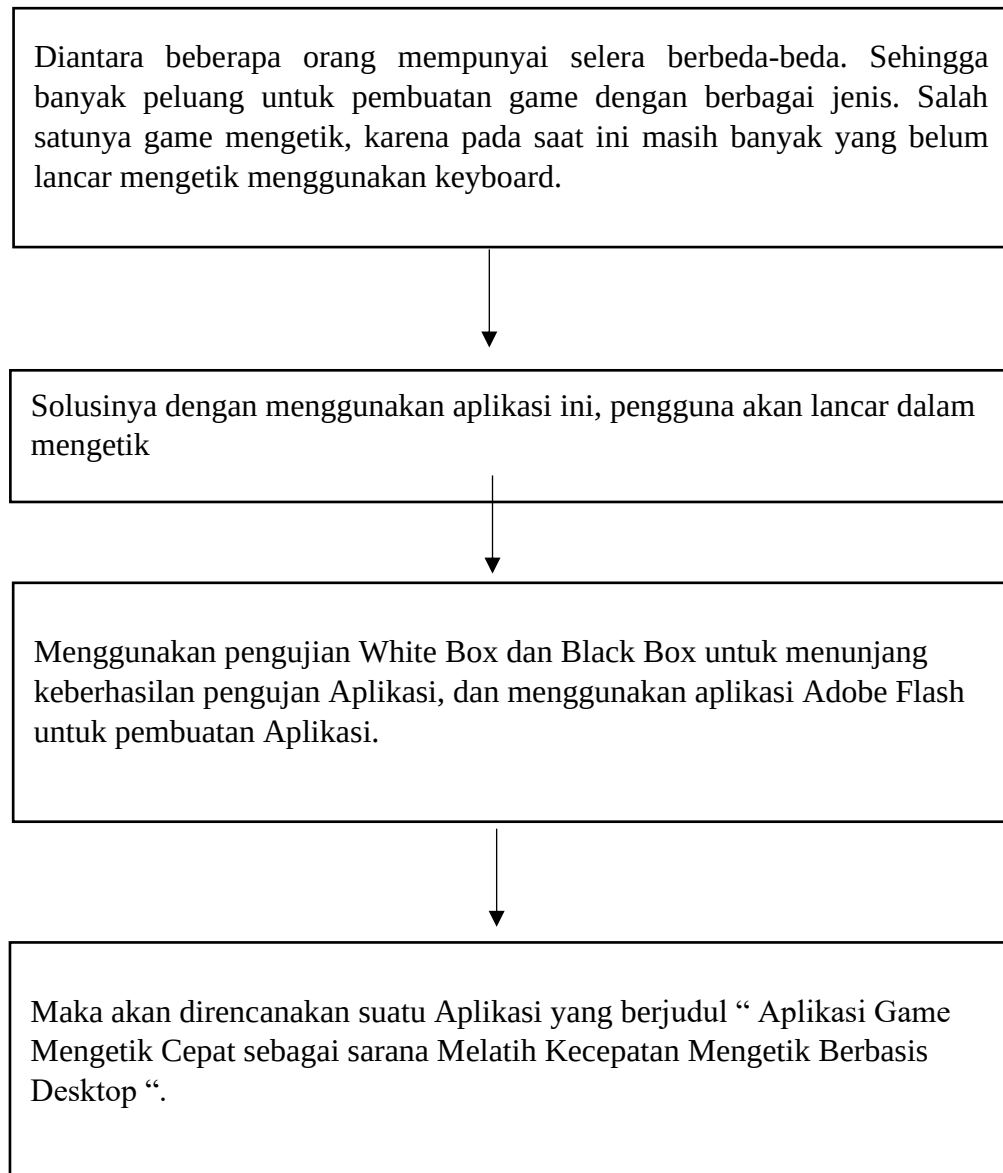
NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang 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. 5** 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

F. Kerangka Pikir

Untuk lebih memahami alur penelitian di atas, di uraikan ke dalam kerangka berpikir yang akan disajikan dalam bentuk diagram berikut ini :



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis Penelitian yang digunakan dalam melakukan penelitian ini adalah Penelitian kualitatif dimana dalam pembuatan proposal ini menggambarkan fakta – fakta dan informasi secara sistematis, faktual dan akurat.

B. Waktu Penelitian

Waktu yang digunakan dalam pelaksanaan penelitian ini adalah $\pm$ 6 bulan yaitu pada bulan Desember 2023 hingga Mei 2024.

C. Teknik Pengumpulan Data

Pengumpulan data dilakukan untuk mengumpulkanseluruh informasi yang terkait dan mendukung pelaksanaan penelitian tugas akhir ini. Karena itu seorang peneliti harus terampil dalam mengumpulkan data agar mendapat data yang valid.

1. Secara tidak langsung (studi literatur)

Metode tidak langsung ini maksudnya ialah mengumpulkan data-data maupun informasi yang terkait atau artikel yang berasal dari media internet.

2. Secara langsung

Metode secara langsung yaitu mengumpulkan data-data atau informasi yang terkait dengan perancangan program aplikasi.

D. Alat dan Bahan Penelitian

Alat dan Bahan penelitian yang digunakan selama proses penelitian antara lain:

a. Perangkat Keras (*Hardware*)

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

- Laptop : Asus A456U
- Processor : *Intel® Core™ i5-7200U*
- RAM : 8GB DDR4
- SSD : 120 GB
- HDD : 1 TB

b. Perangkat Lunak (*Software*)

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

- Windows 10
- Adobe Animate

E. Tahapan Penelitian

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

a. 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.

b. 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.

c. 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.

d. Perancangan

Peneliti kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah.

e. Pengujian

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

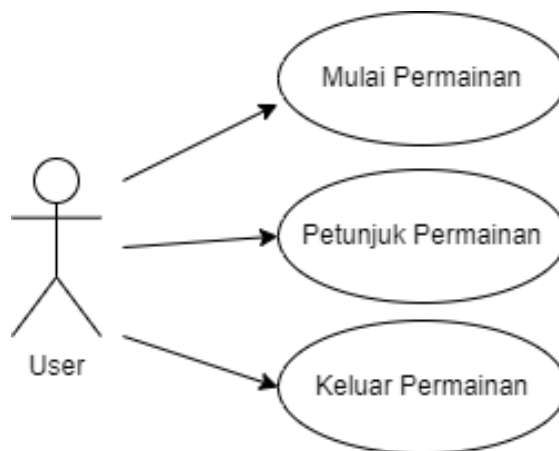
f. Implementasi

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

F. Desain Sistem

1. Sistem yang berjalan

Adapun sistem yang berjalan saat ini yaitu:

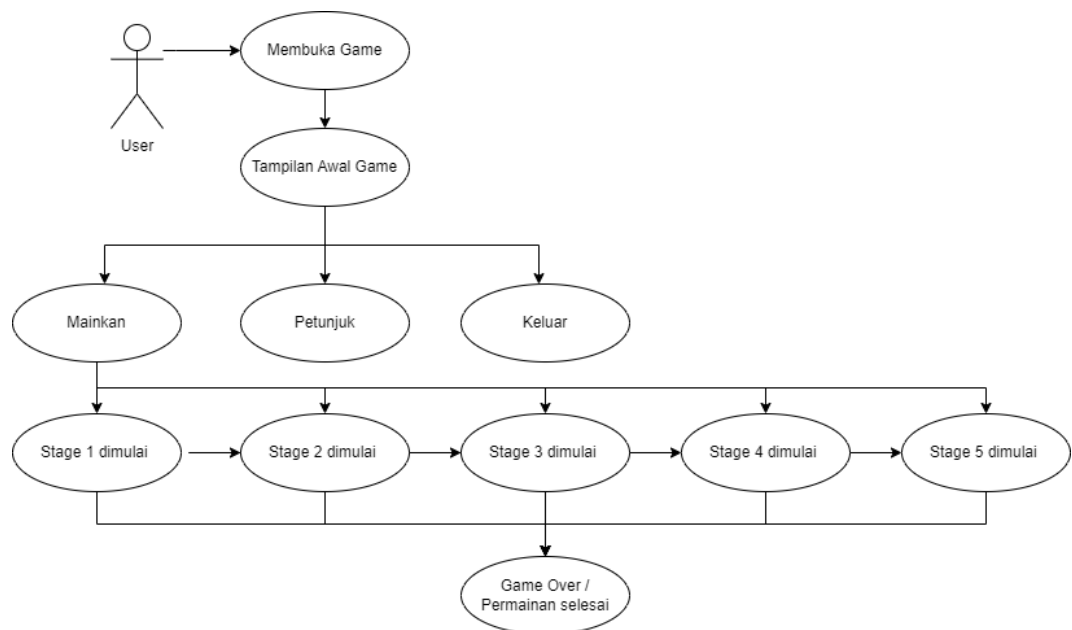


Gambar 3. 1 Sistem yang berjalan

Sistem yang berjalan saat ini adalah user dapat memilih memulai permainan, petunjuk permainan, atau keluar permainan.

2. Sistem yang diusulkan

Adapun sistem yang diusulkan saat ini yaitu:

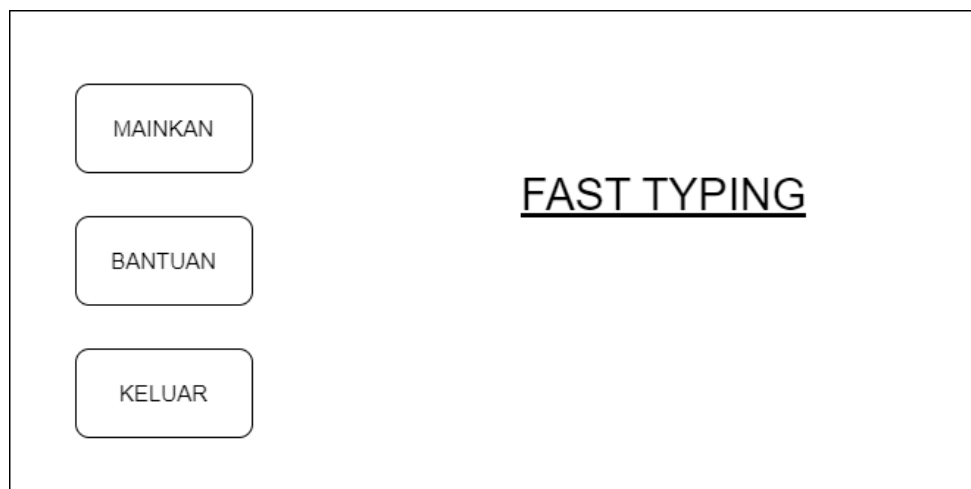


Gambar 3. 2 Sistem yang diusulkan

Sistem yang diusulkan saat ini adalah user membuka game dan muncul tampilan awal dan ada 3 pilihan yaitu mainkan, petunjuk, dan keluar, setelah itu mainkan dan akan muncul stage 1 dan seterusnya jika kalah akan game over.

3. Sistem Interface

a. Tampilan Awal Game

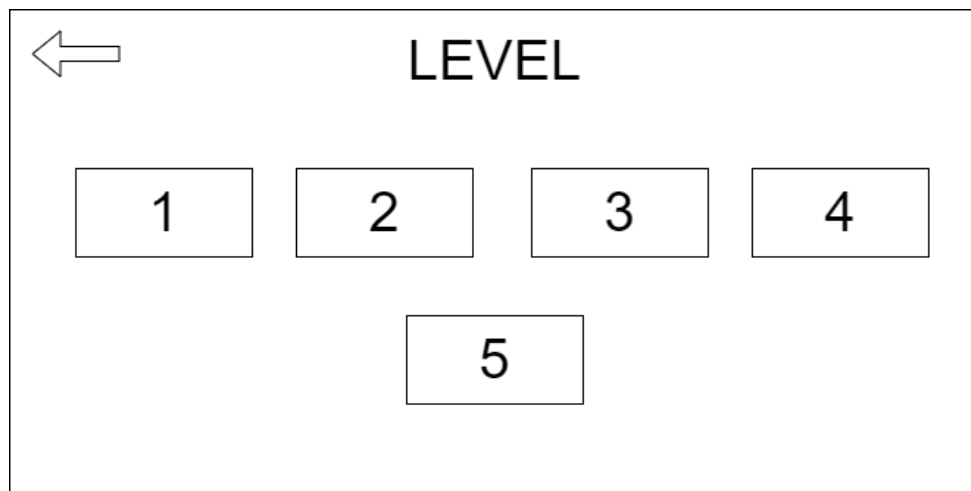


Gambar 3. 3 Tampilan Awal

Pada gambar diatas menampilkan Judul *Game* yaitu “*Fast Typing*” dan ada 3 tombol yaitu:

1. Mainkan = Tombol untuk memulai permainan.
2. Bantuan = Tombol untuk menampilkan informasi tentang cara bermain.
3. Keluar = Tombol untuk keluar dari permainan.

b. Tampilan *Level*



Gambar 3. 4 Tampilan *Level*

Pada gambar diatas menampilkan jumlah level yang tersedia yang berisikan 5 *level*.

c. Tampilan *Level 1*

Gambar 3. 5 Tampilan *Level 1*

Gambar diatas menampilkan *gameplay* pada *level 1*, di *level 1* waktu yang tersedia hanya 1 menit dan kata yang akan diketik oleh pengguna berada di bawah musuh yang terdiri dari 3 huruf saja. Pengguna berperan sebagai Anggota Militer dan Musuh berperan sebagai *zombie*. Musuh akan berjalan menghampiri pengguna, jika musuh telah berhasil sampai di pengguna dan menggigit nya maka game berakhir (*GameOver*). Kecepatan jalan pada musuh akan selalu meningkat disetiap levelnya .Jika pengguna berhasil menyelesaikan dengan waktu yang ditentukan maka bisa lanjut ke *level* berikutnya.

d. Tampilan *Level 2*

Gambar 3. 6 Tampilan *Level 2*

Gambar diatas menampilkan *gameplay* pada *level 2*, di *level 2* waktu yang tersedia hanya 55 detik dan kata yang akan diketik oleh pengguna berada di bawah musuh yang terdiri dari 4 huruf. Pengguna berperan sebagai Anggota Militer dan Musuh berperan sebagai *zombie*. Musuh akan berjalan menghampiri pengguna, jika musuh telah berhasil sampai di pengguna dan menggigit nya maka game berakhir (*GameOver*). Kecepatan jalan pada musuh akan selalu meningkat disetiap levelnya .Jika pengguna berhasil menyelesaikan dengan waktu yang ditentukan maka bisa lanjut ke *level* berikutnya.

e. Tampilan *Level 3*

Gambar 3. 7 Tampilan *Level 3*

Gambar diatas menampilkan *gameplay* pada *level 3*, di *level 3* waktu yang tersedia hanya 50 detik dan kata yang akan diketik oleh pengguna berada di bawah musuh yang terdiri dari 5 huruf. Pengguna berperan sebagai Anggota Militer dan Musuh berperan sebagai *zombie*. Musuh akan berjalan menghampiri pengguna, jika musuh telah berhasil sampai di pengguna dan menggigit nya maka game berakhir (*GameOver*). Kecepatan jalan pada musuh akan selalu meningkat disetiap levelnya .Jika pengguna berhasil menyelesaikan dengan waktu yang ditentukan maka bisa lanjut ke *level* berikutnya.

f. Tampilan *Level 4*

Gambar 3. 8 Tampilan *Level 4*

Gambar diatas menampilkan *gameplay* pada *level 4*, di *level 4* waktu yang tersedia hanya 45 detik dan kata yang akan diketik oleh pengguna berada di bawah musuh yang terdiri dari 6 huruf. Pengguna berperan sebagai Anggota Militer dan Musuh berperan sebagai *zombie*. Musuh akan berjalan menghampiri pengguna, jika musuh telah berhasil sampai di pengguna dan menggigit nya maka game berakhir (*GameOver*). Kecepatan jalan pada musuh akan selalu meningkat disetiap levelnya. Jika pengguna berhasil menyelesaikan dengan waktu yang ditentukan maka bisa lanjut ke *level* berikutnya.

g. Tampilan *Level 5***Gambar 3. 9** Tampilan *Level 5*

Gambar diatas menampilkan *gameplay* pada *level 5*, di *level 5* waktu yang tersedia hanya 40 detik dan kata yang akan diketik oleh pengguna berada di bawah musuh yang terdiri dari 6 huruf. Pengguna berperan sebagai Anggota Militer dan Musuh berperan sebagai *zombie*. Musuh akan berjalan menghampiri pengguna, jika musuh telah berhasil sampai di pengguna dan menggigit nya maka game berakhir (*GameOver*). Kecepatan jalan pada musuh akan selalu meningkat disetiap levelnya. Jika pengguna berhasil menyelesaikan dengan waktu yang ditentukan maka bisa lanjut ke *level* berikutnya.

h. Tampilan Pause



Gambar 3. 10 Tampilan Menu *Pause*

Pada Gambar diatas, Terdapat 3 tombol yaitu:

1. Lanjutkan : Untuk melanjutkan *game* yang sedang berjalan.
2. Ulangi : Untuk mengulangi *game* yang sedang berjalan.
3. Keluar : Untuk keluar dari *game* yang sedang berjalan.

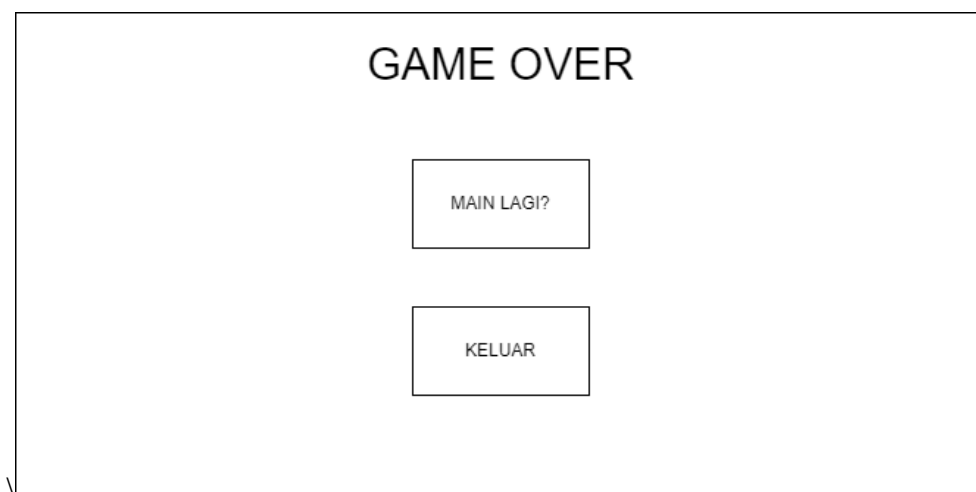
i. Tampilan *Level* Selanjutnya



Gambar 3. 11 Tampilan *Level* Selanjutnya

Pada Gambar diatas, terdapat tombol untuk melanjutkan ke level selanjutnya.

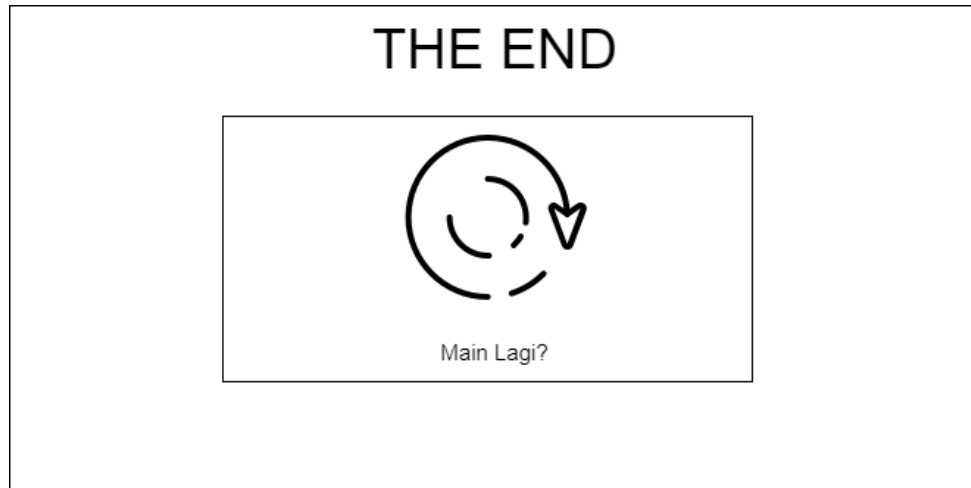
j. Tampilan *Game Over*



Gambar 3. 12 Tampilan *GameOver*

Pada Gambar diatas, terdapat tombol Main Lagi untuk mengulangi *game* dari awal, dan tombol keluar untuk keluar dari *game*

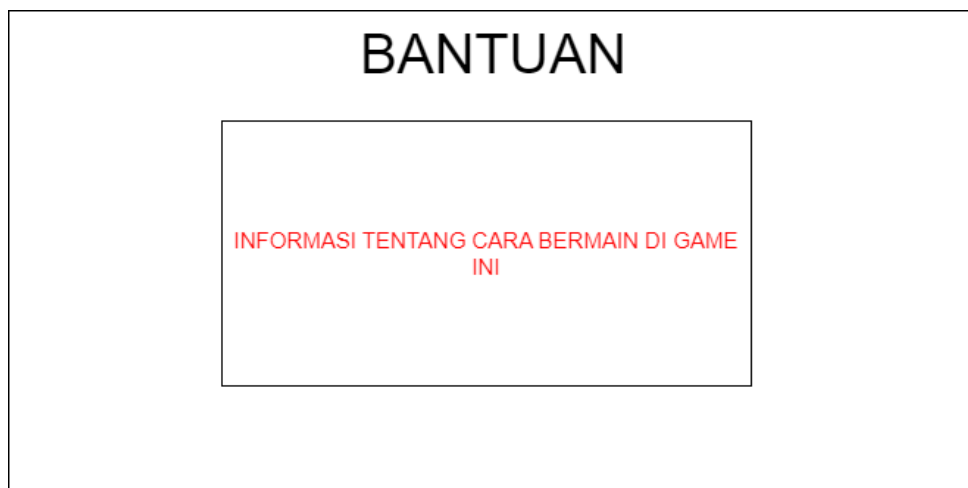
k. Tampilan *The End*



Gambar 3. 13 Tampilan *The End*

Pada Gambar diatas, dijelaskan bahwa *game* nya telah selesai, terdapat tombol main lagi untuk mengulangi *game* dari awal.

l. Tampilan Bantuan



Gambar 3. 14 Tampilan Bantuan

Pada Gambar diatas, di menu bantuan berisikan tentang cara bermain di game ini.

m. Tampilan Keluar



Gambar 3. 15 Tampilan Keluar

Pada Gambar diatas, di menu keluar ada tombol ya dan tidak.

BAB IV

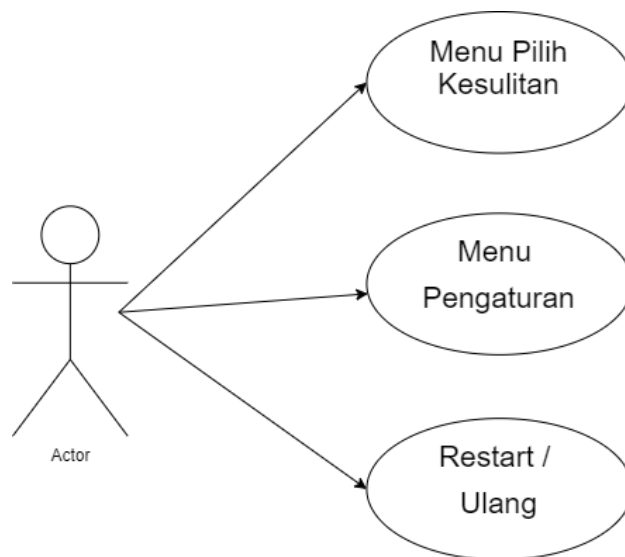
HASIL PENELITIAN

A. Analisis Aliran Data dengan UML

Analisis aliran data bertujuan mengetahui alur proses Aplikasi *Game Mengetik* ini. Dalam *analisis system* ini, penulis menggunakan pengembangan *orientasi objek* sehingga menggunakan *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

1. Use Case Diagram

Use Case Diagram berfungsi untuk menjalankan manfaat *system* jika dilihat menurut pandangan orang yang berada di luar *system*.



Gambar 4. 1 *Use Case Diagram*

Tabel 4. 1 Deskripsi *Use Case*

Nama <i>Use Case</i>	Deskripsi <i>Use Case</i>
Pilih Kesulitan	Merupakan Menu yang menampilkan 2 tingkat kesulitan yaitu Mudah dan Sulit.
Pengaturan	Merupakan Pengaturan untuk mengatur suara di dalam <i>game</i>
Restart / Ulang	Merupakan Menu yang akan muncul ketika <i>user</i> menang ataupun kalah.

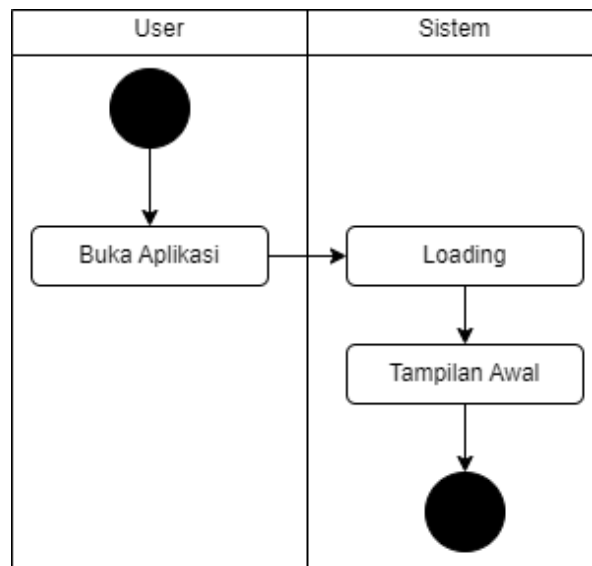
Tabel 4. 2 Definisi *Use Case*

Nama <i>Use Case</i>	Definisi <i>Use Case</i>
Membuka Aplikasi/ <i>Game</i>	Proses Awal ketika pengguna membuka aplikasi atau game akan ditampilkan halaman <i>Menu</i>
Membuka Mulai	Merupakan menu untuk memulai permainan, ketika pengguna memilih Mulai maka akan menampilkan 2 pilihan tingkat kesulitan
Membuka Pengaturan	Merupakan menu yang menampilkan fitur untuk <i>on/off</i> suara yang berjalan
Mode Mudah dan Sulit	Merupakan dua tingkat kesulitan yang terdapat didalam <i>game</i>

2. Activity Diagram User

Activity Diagram ini menjelaskan tentang aktivitas-aktivitas yang terjadi dalam sebuah aliran proses pengenalan Aplikasi.

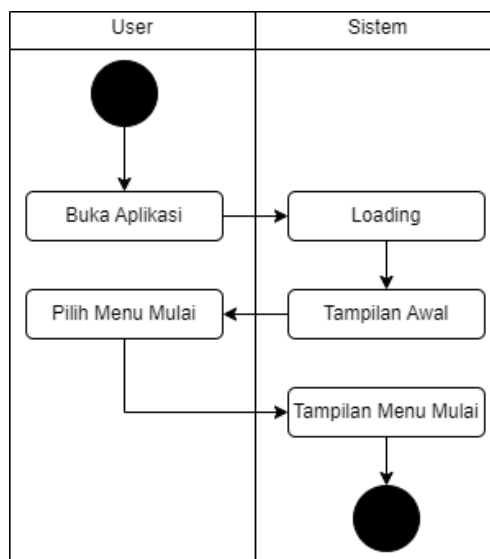
a. *Activity Diagram Tampilan Awal*



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 *loading* lalu menampilkan tampilan Awal.

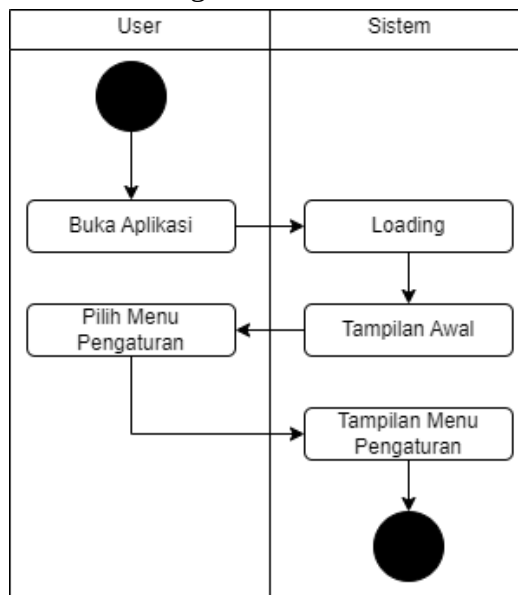
b. *Activity Diagram Menu Mulai*



Gambar 4. 3 *Activity Diagram Tampilan Menu Mulai*

Pada gambar 4.3 menjelaskan bagaimana proses memilih *Menu Mulai*. Langkah pertama yang dilakukan pertama yaitu *user* membuka aplikasi, selanjutnya sistem menampilkan tampilan *loading* kemudian menampilkan tampilan Awal. Lalu *user* memilih tombol Mulai dan sistem akan menampilkan tampilan *Menu Mulai*.

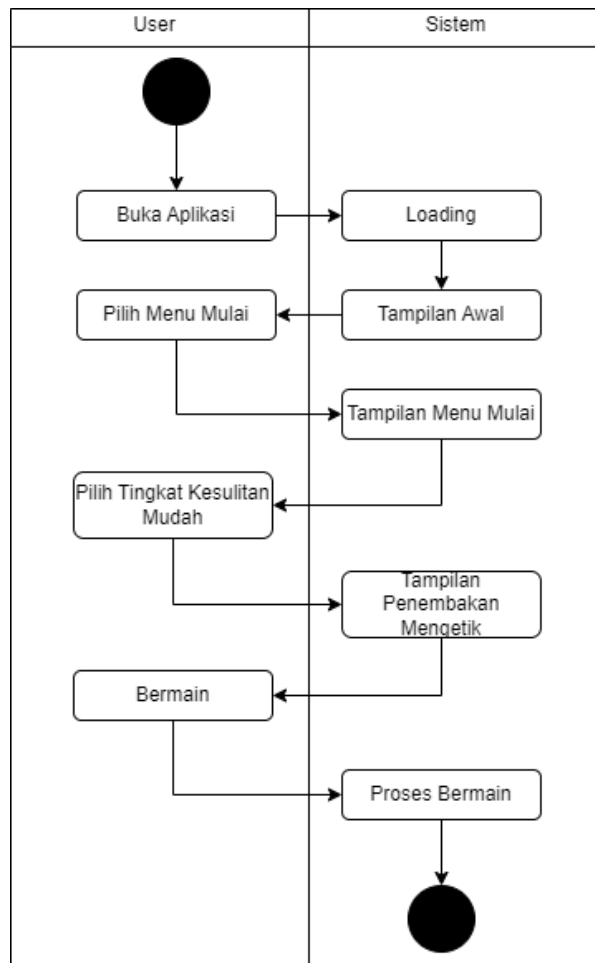
c. *Activity Diagram Menu Pengaturan*



Gambar 4. 4 *Activity Diagram Tampilan Menu Pengaturan*

Pada gambar 4.4 menjelaskan bagaimana proses memilih *Menu Pengaturan*. Langkah pertama yang dilakukan yaitu *user* membuka aplikasi, selanjutnya sistem menampilkan tampilan *loading* kemudian menampilkan tampilan Awal. Lalu *user* memilih tombol Pengaturan dan sistem akan menampilkan tampilan *Menu Pengaturan*.

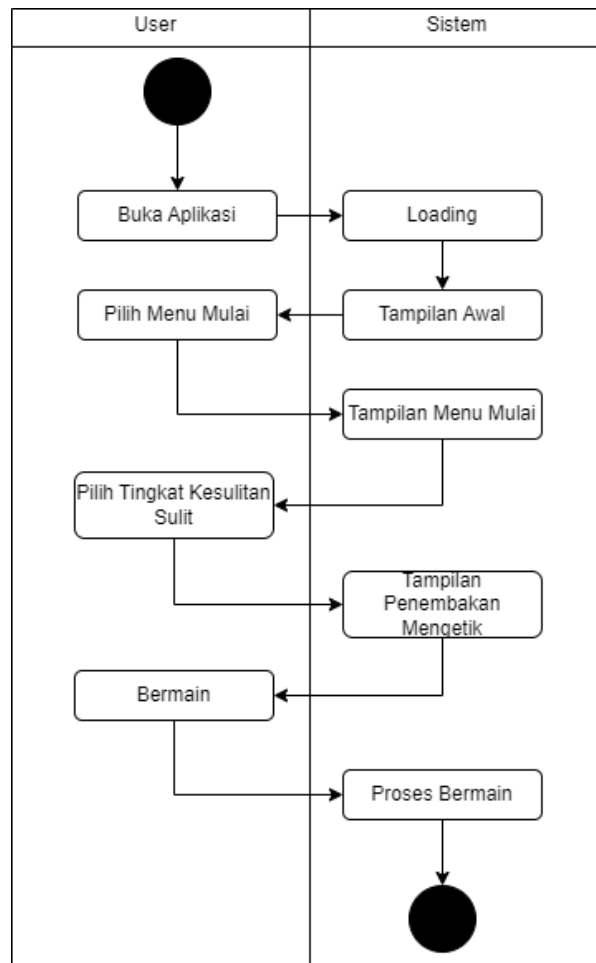
d. *Activity Diagram Tampilan Menu Mudah*



Gambar 4. 5 *Activity Diagram Tampilan Menu Mudah*

Pada gambar 4.5 menjelaskan bagaimana proses memilih *Menu Mudah*. Langkah pertama yang dilakukan yaitu *user* membuka aplikasi, selanjutnya sistem menampilkan tampilan *loading* kemudian menampilkan tampilan Awal. Lalu *user* memilih tombol Mulai dan sistem akan menampilkan tampilan *Menu Mulai*. Selanjutnya *user* akan memilih tingkat kesulitan Mudah dan sistem akan menampilkan Tampilan Penembakan Mengetik, kemudian *user* akan bermain.

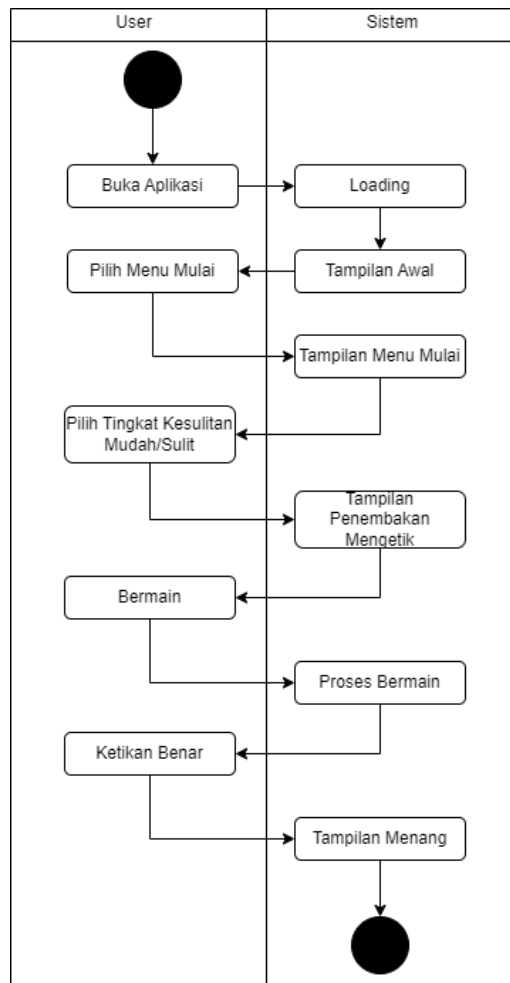
e. *Activity Diagram Tampilan Menu Sulit*



Gambar 4. 6 *Activity Diagram Tampilan Menu Sulit*

Pada gambar 4.6 menjelaskan bagaimana proses memilih *Menu Sulit*. Langkah pertama yang dilakukan yaitu *user* membuka aplikasi, selanjutnya sistem menampilkan tampilan *loading* kemudian menampilkan tampilan Awal. Lalu *user* memilih tombol Mulai dan sistem akan menampilkan tampilan *Menu Mulai*. Selanjutnya *user* akan memilih tingkat kesulitan Sulit dan sistem akan menampilkan Tampilan Penembakan Mengetik, kemudian *user* akan bermain.

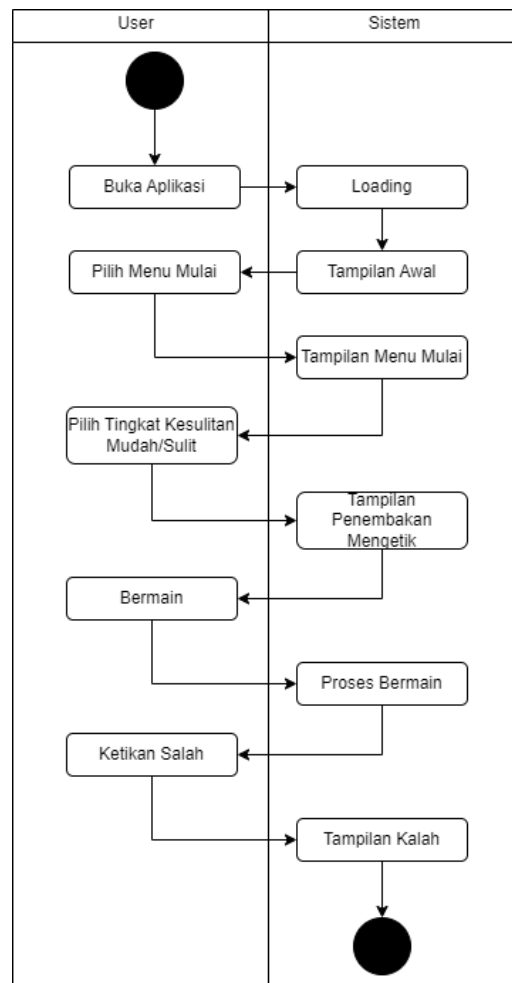
f. *Activity Diagram* Tampilan Menu Menang



Gambar 4. 7 *Activity Diagram* Tampilan Menu Menang

Pada gambar 4.7 menjelaskan bagaimana proses munculnya tampilan Menang. Langkah pertama yang dilakukan yaitu *user* membuka aplikasi, selanjutnya sistem menampilkan tampilan *loading* kemudian menampilkan tampilan Awal. Lalu *user* memilih tombol Mulai dan sistem akan menampilkan tampilan *Menu* Mulai. Selanjutnya *user* akan memilih tingkat kesulitan Sulit dan sistem akan menampilkan Tampilan Penembakan Mengetik, kemudian *user* akan bermain. Setelah itu jika *user* berhasil mengetik dengan benar maka sistem akan menampilkan tampilan Menang.

g. Activity Diagram Tampilan Menu Kalah



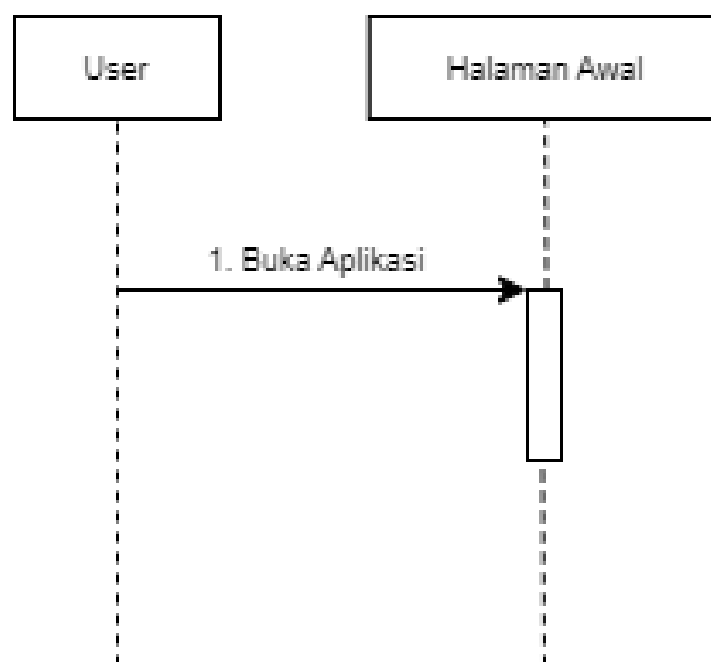
Gambar 4. 8 Activity Diagram Tampilan Menu Kalah

Pada gambar 4.8 menjelaskan bagaimana proses munculnya tampilan Kalah. Langkah pertama yang dilakukan yaitu *user* membuka aplikasi, selanjutnya sistem menampilkan tampilan *loading* kemudian menampilkan tampilan Awal. Lalu *user* memilih tombol Mulai dan sistem akan menampilkan tampilan *Menu* Mulai. Selanjutnya *user* akan memilih tingkat kesulitan Sulit dan sistem akan menampilkan Tampilan Penembakan Mengetik, kemudian *user* akan bermain. Setelah itu jika *user* tidak dapat mengetik dengan benar maka sistem akan menampilkan tampilan Kalah.

3. *Sequence Diagram User*

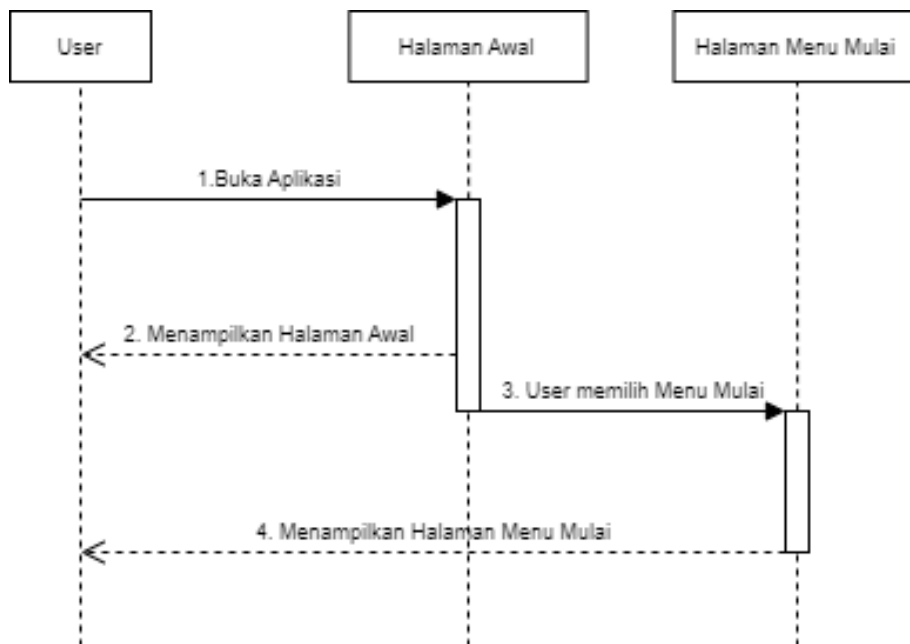
Sequence Diagram merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan *message* (pesan) apa yang dikirim dan kapan pelaksanaannya.

a. *Sequence Diagram Tampilan Awal*



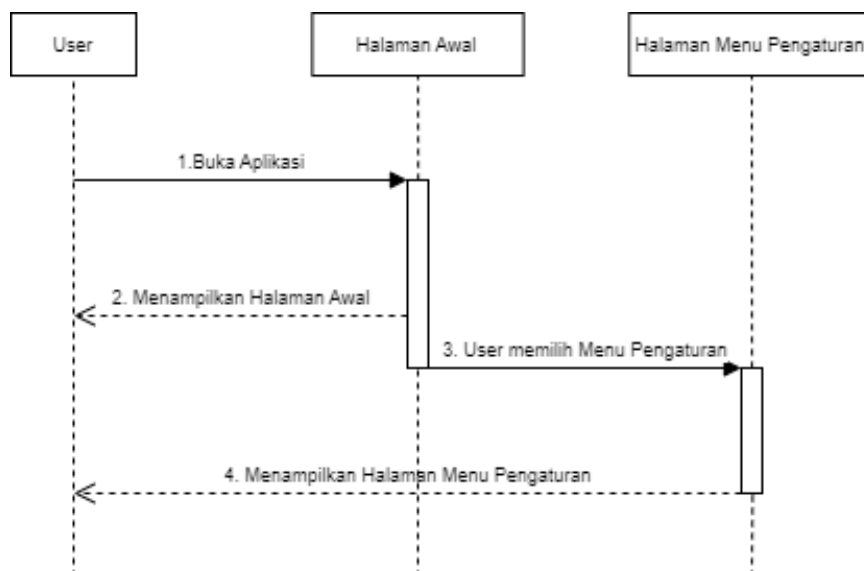
Gambar 4. 9 *Sequence Diagram Tampilan Awal*

b. *Sequence Diagram* Tampilan Mulai



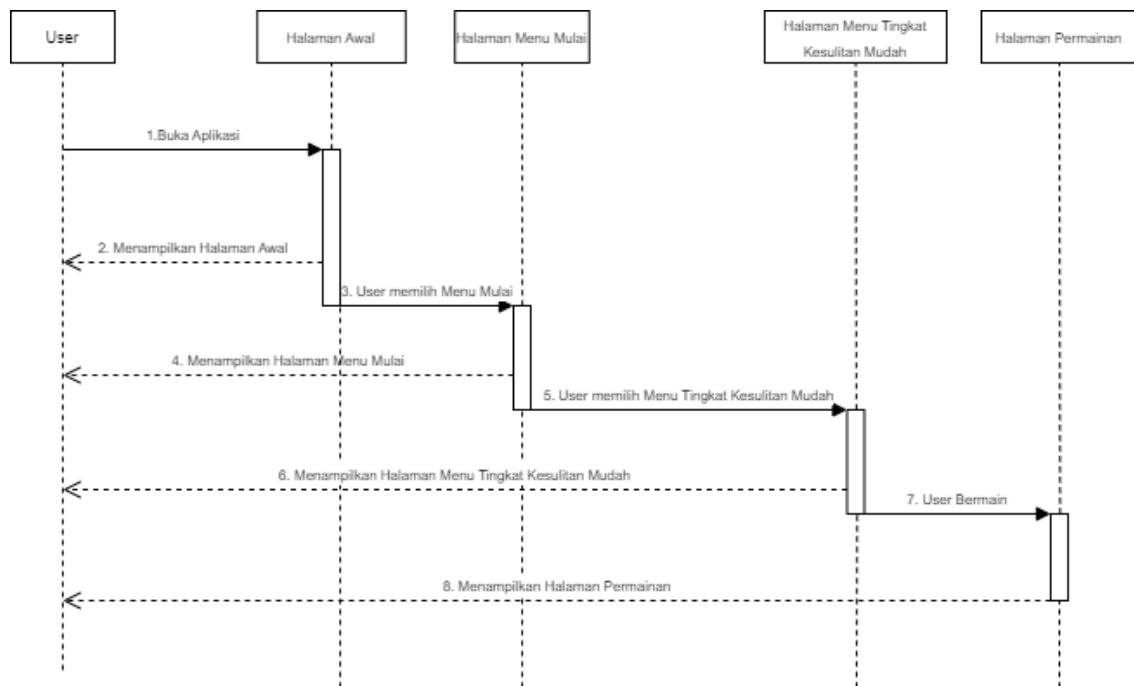
Gambar 4. 10 *Sequence Diagram* Tampilan Mulai

c. *Sequence Diagram* Tampilan Menu Pengaturan



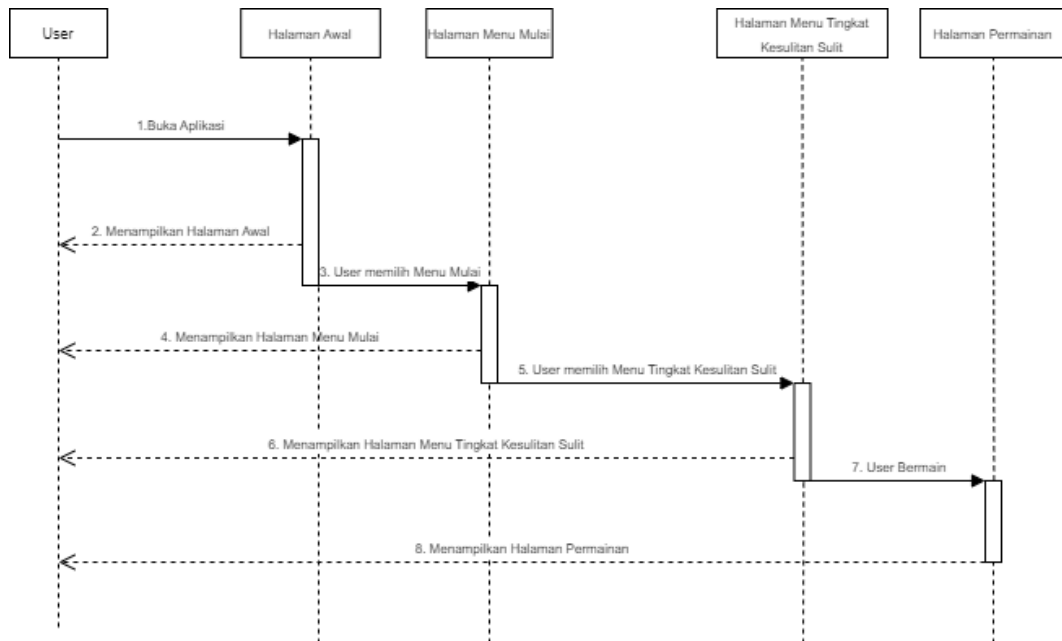
Gambar 4. 11 *Sequence Diagram* Tampilan Pengaturan

d. *Sequence Diagram Tampilan Menu Mudah*



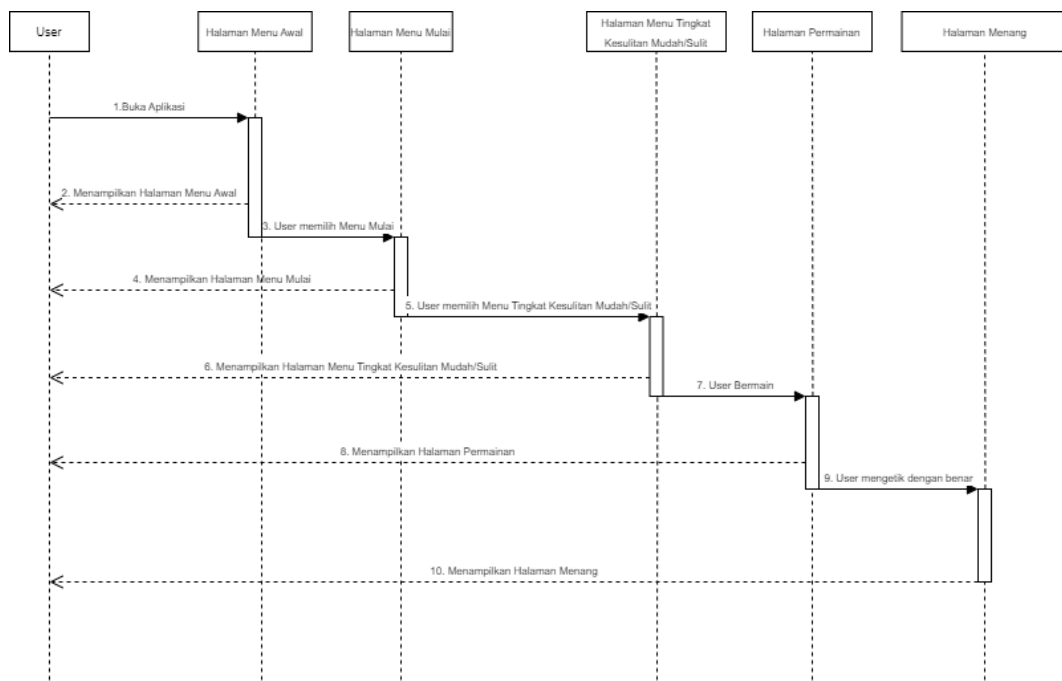
Gambar 4. 12 *Sequence Diagram Tampilan Menu Mudah*

e. *Sequence Diagram Tampilan Menu Sulit*



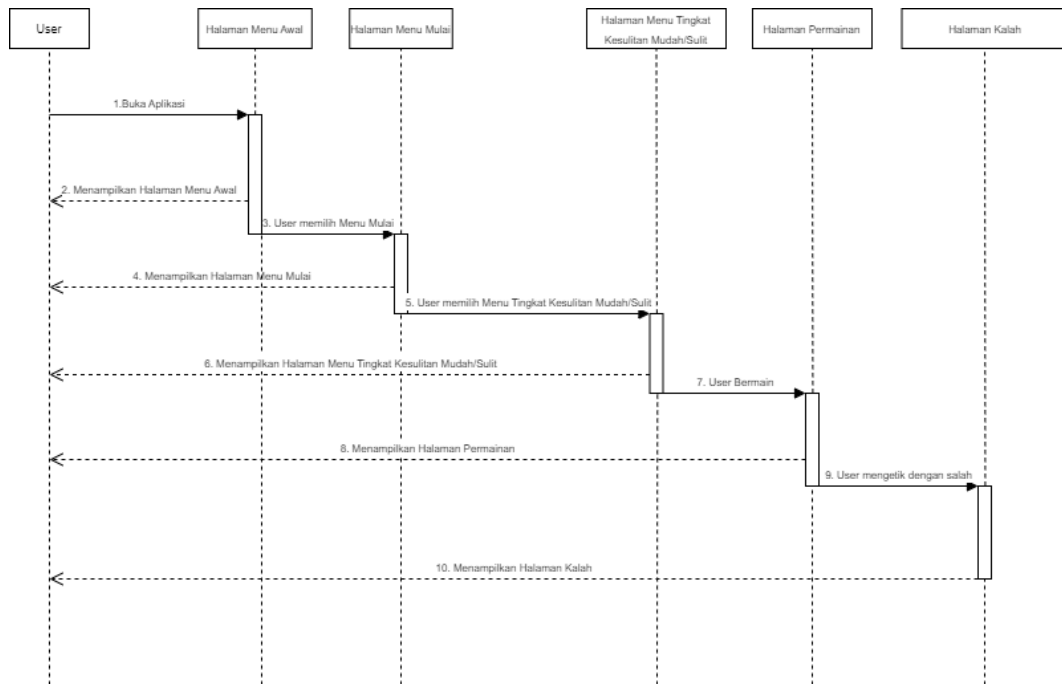
Gambar 4. 13 *Sequence Diagram Tampilan Menu Sulit*

f. *Sequence Diagram Tampilan Menu Menang*



Gambar 4. 14 *Sequence Diagram Tampilan Menu Menang*

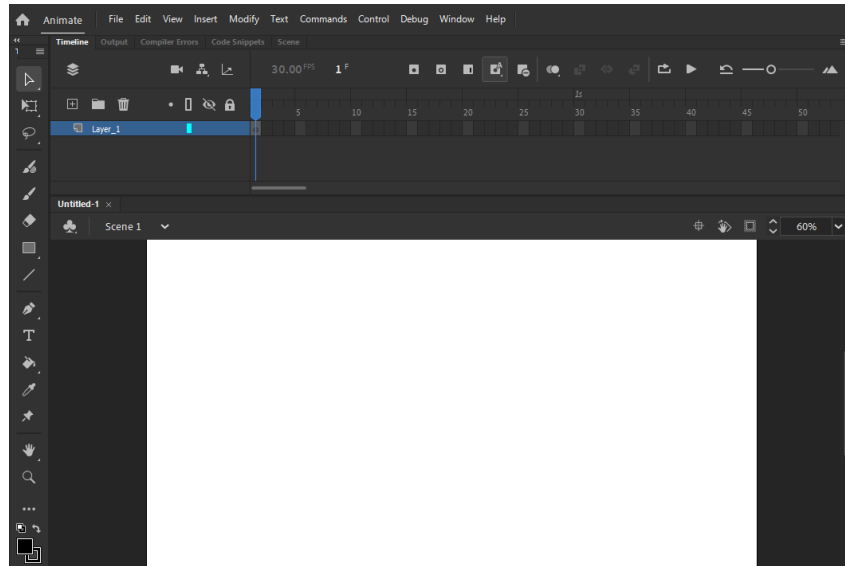
g. *Sequence Diagram Tampilan Menu Kalah*



Gambar 4. 15 *Sequence Diagram Tampilan Menu Kalah*

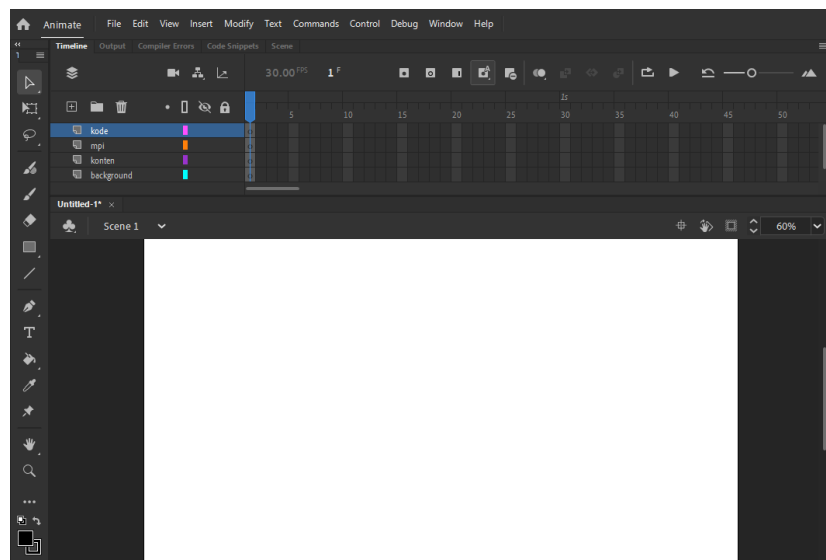
B. Pembuatan Aplikasi

1. Buka Aplikasi Adobe Animate yang sudah ter *install* di laptop anda, seperti gambar di bawah ini :



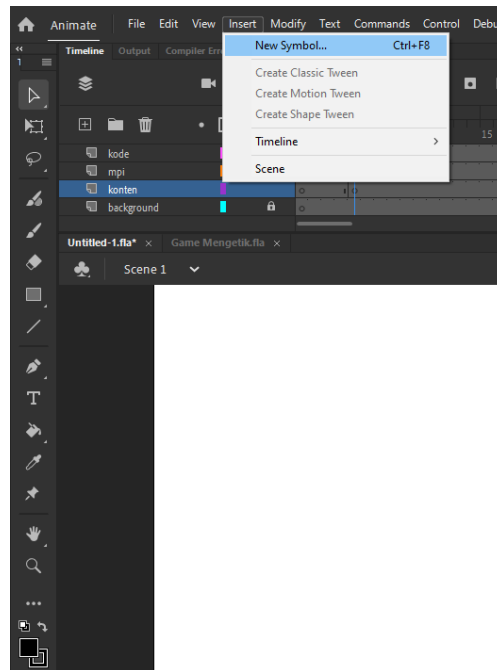
Gambar 4. 16 Tampilan Awal Adobe Animate

2. Lalu buatlah 4 layer yang bernama : *background*, konten, MPI, Kode.



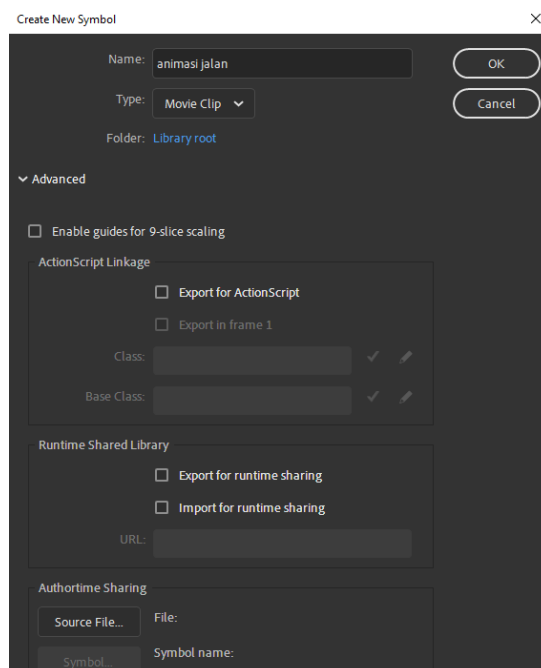
Gambar 4. 17 Membuat *Layer*

3. Selanjutnya buat *symbol* dengan cara klik *insert -> New Symbol*.



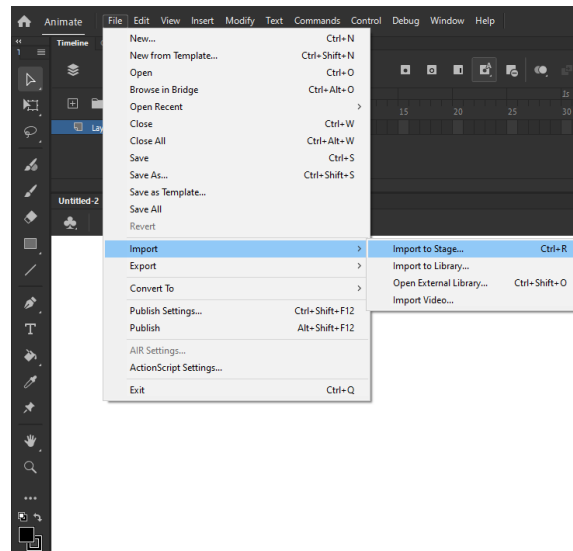
Gambar 4. 18 Membuat *Symbol*

4. Dan akan muncul tampilan seperti dibawah, Di kolom *name* kita isi nama *symbol* yang ingin dibuat dan di kolom *type* yang dipilih *Movie Clip*.



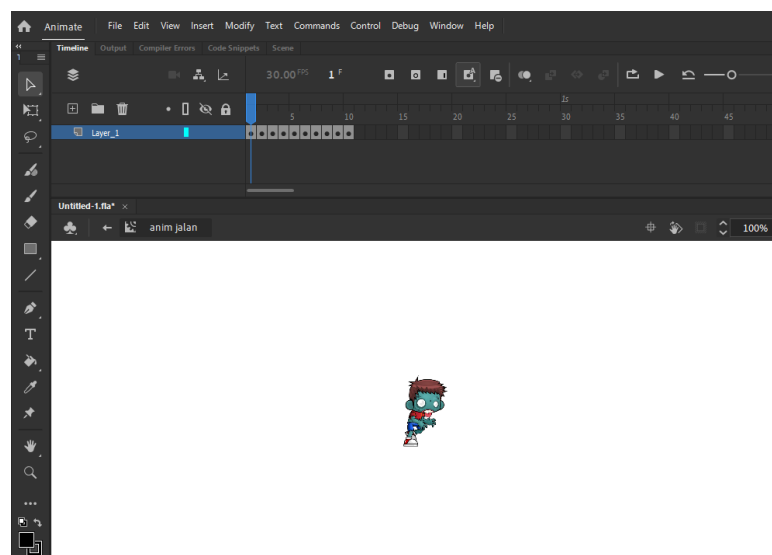
Gambar 4. 19 *Settingan* untuk *Symbol* *Zombie*

5. Setelah itu *import sprite* yang sudah disediakan dengan cara klik *file -> Import -> Import to Stage*.



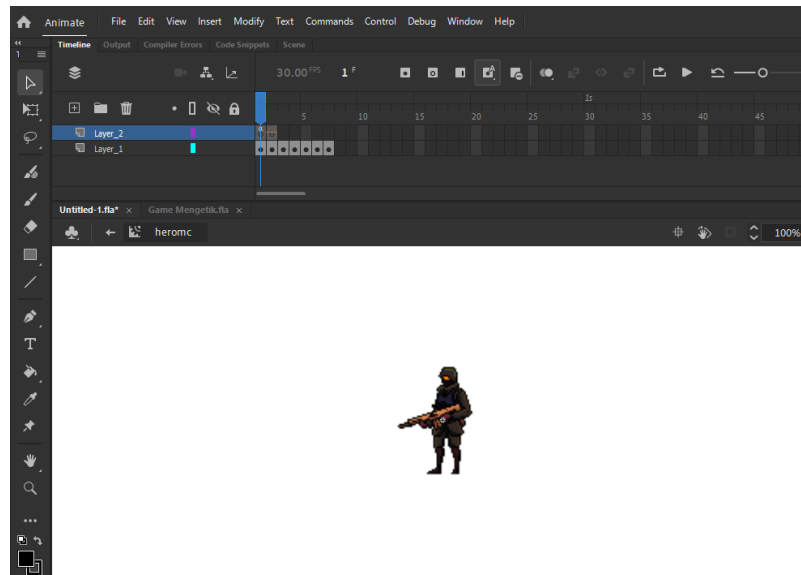
Gambar 4. 20 Proses *Import Sprite*

6. Setelah *sprite* di *import* maka jadilah animasi berjalan untuk *zombie*



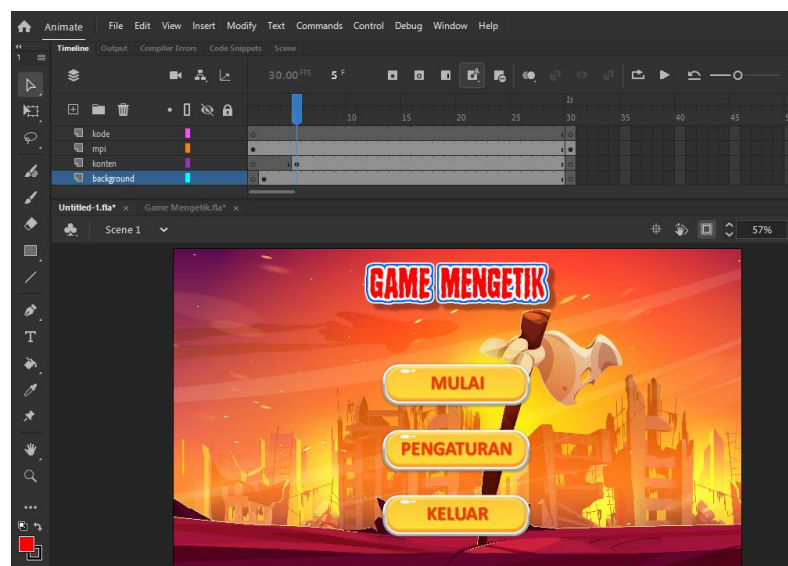
Gambar 4. 21 Membuat Animasi Berjalan *Zombie*

7. Setelah itu kita membuat animasi menembak untuk karakter *Hero* kita dengan cara seperti di *point* 3 – 6 menggunakan *sprite* yang sudah disiapkan.



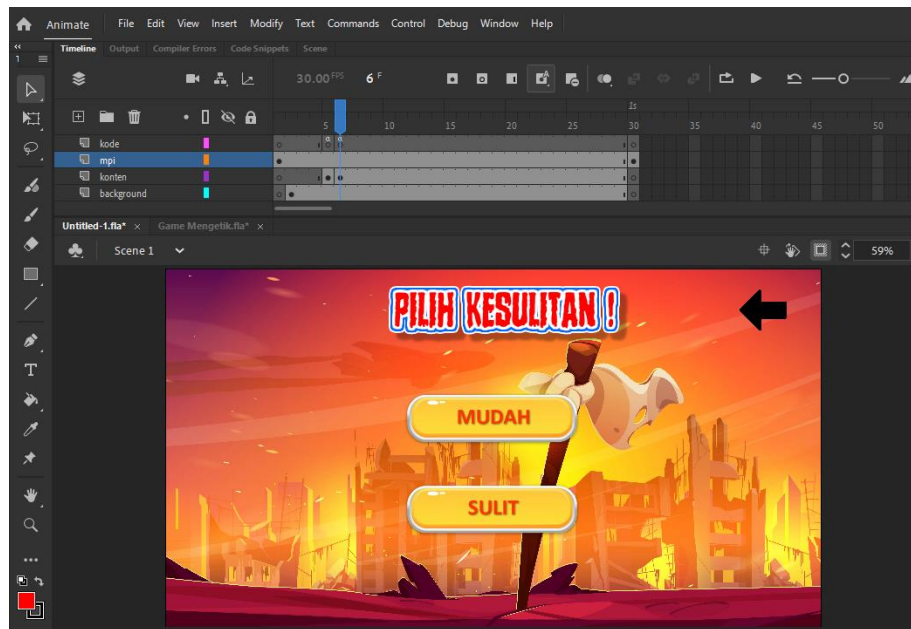
Gambar 4. 22 Membuat Animasi Menembak *Hero*

8. Kemudian masukkan *background* di *layer background* dan kita buat 3 *button* di *layer* konten seperti gambar di bawah ini :



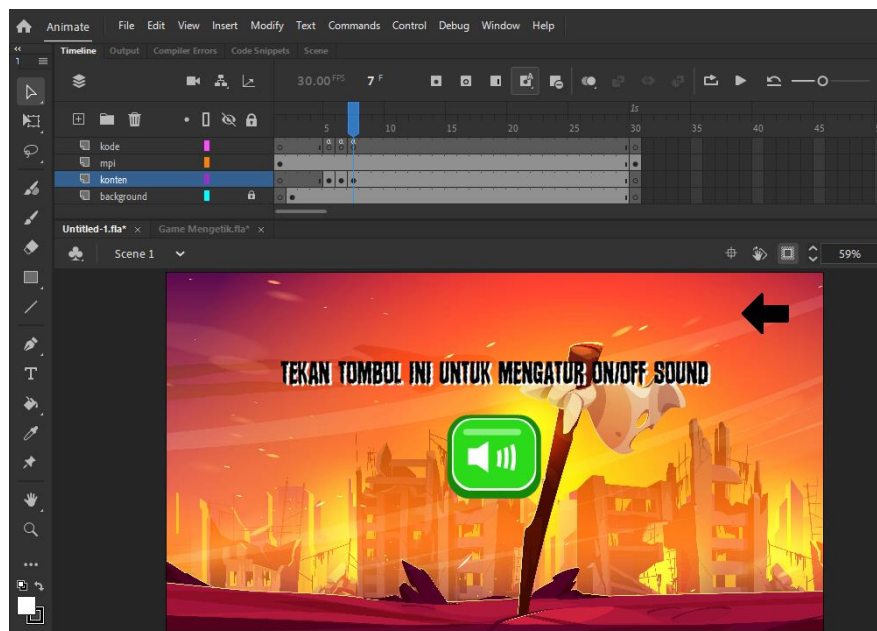
Gambar 4. 23 Membuat Tampilan Awal

9. Setelah itu kita membuat *Menu Tingkat Kesulitan*, kita buat 2 *button* di *layer* konten seperti gambar di bawah ini :



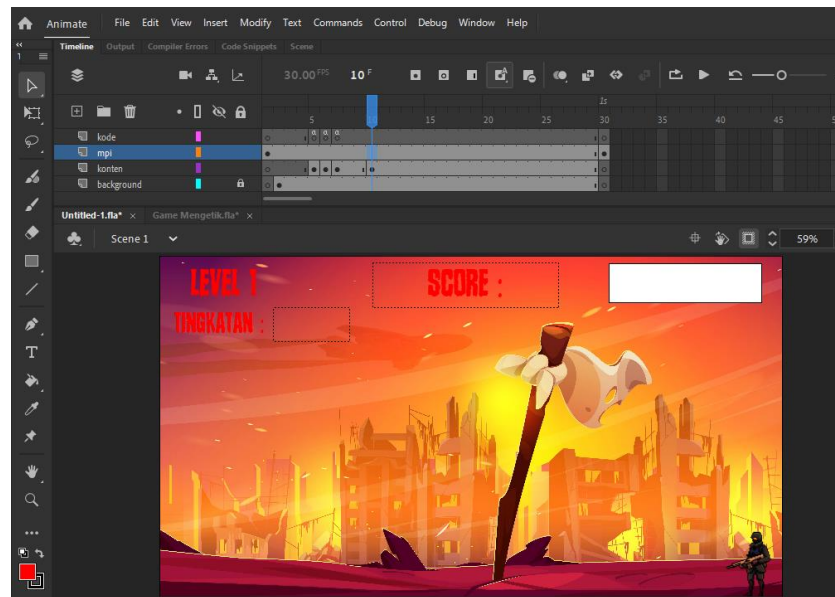
Gambar 4. 24 Membuat Tampilan Tingkat Kesulitan

10. Selanjutnya kita membuat *Menu Pengaturan* untuk mengutar *on/off* suara yang terdapat di dalam *game* ini



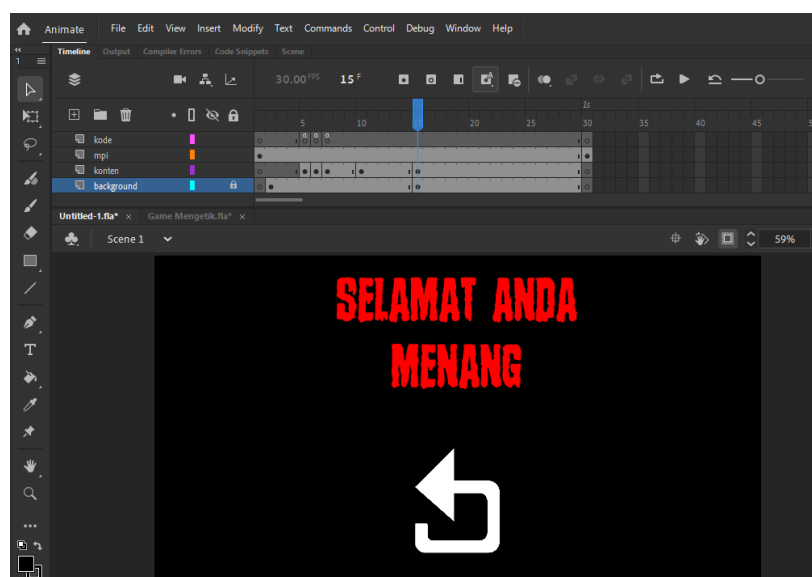
Gambar 4. 25 Membuat Tampilan Pengaturan

11. Setelah itu kita membuat *level 1, 2, 3, 4, 5* dengan kecepatan *zombie* yang meningkat di setiap *level*, tetapi disini saya akan membuat *level 1* saja karena *level* yang lain itu sama saja yang berbeda hanyalah dibagian *background* dan *script* nya.



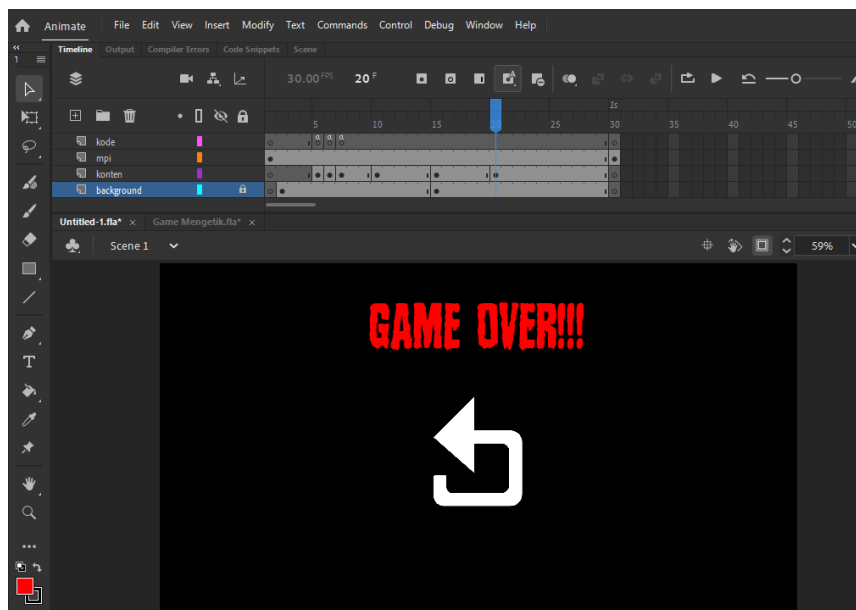
Gambar 4. 26 Membuat Tampilan *Level*

12. Kemudian kita membuat *Menu* menang, ketika pemain berhasil menyelesaikan tantangan level 1 sampai 5 tampilan menang akan muncul.



Gambar 4. 27 Membuat Tampilan Menang

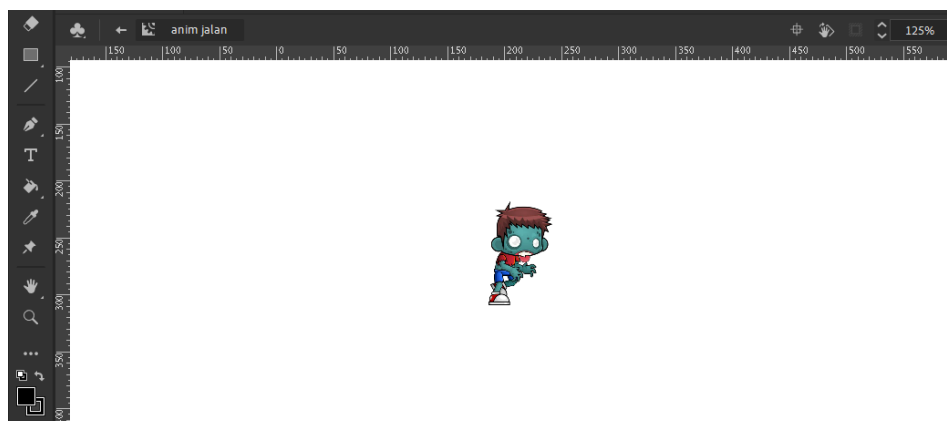
13. kita membuat *Menu* kalah, ketika pemain tidak berhasil menyelesaikan tantangan level 1 sampai 5 tampilan kalah akan muncul.



Gambar 4. 28 Membuat Tampilan Kalah

C. Deskripsi Aplikasi

1. Tampilan *Sprites*



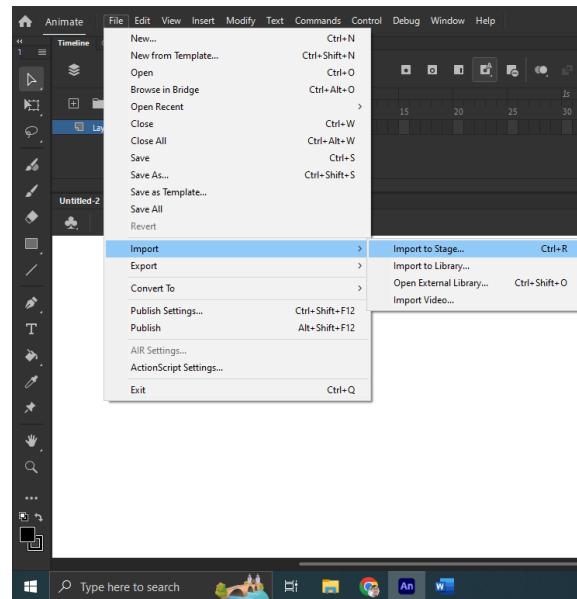
Gambar 4. 29 Tampilan *Sprite*

Pada Tampilan *Sprite* merupakan gambaran desain karakter atau bagian

pendukung yang dipakai pada pada program seperti halnya perencanaan karakter.

Adapun cara untuk meng *import Sprite* yang sudah disiapkan sebagai berikut :

klik File, kemudian pilih *import* setelah itu klik *Import to Stage*, dan silahkan pilih sprite yang sudah disiapkan.



Gambar 4. 30 Tampilan *Import Sprite*

2. Tampilan Awal Aplikasi



Gambar 4. 31 Tampilan Awal Aplikasi

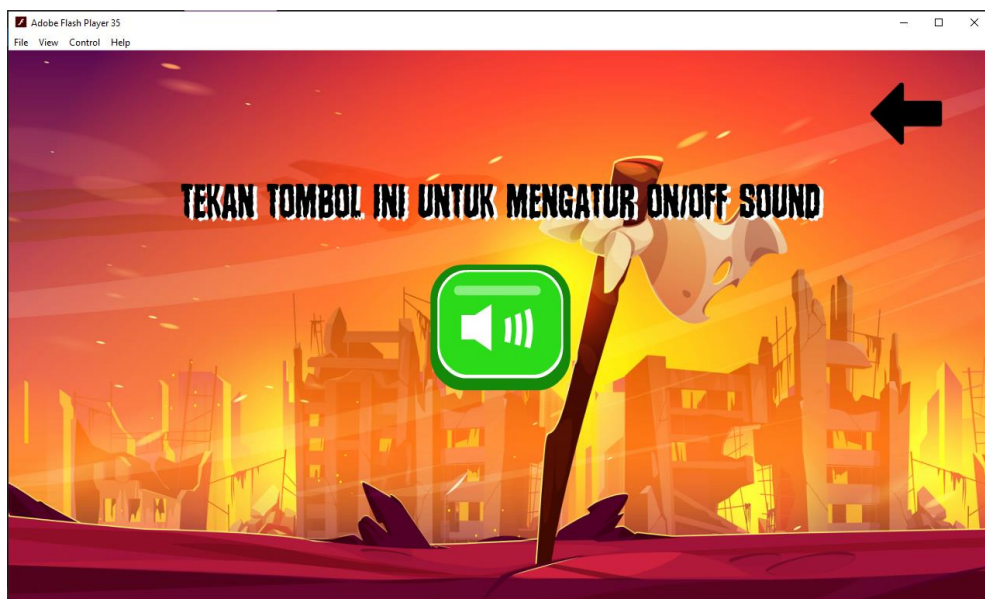
Pada Tampilan Menu utama terdapat button Mulai, Pengaturan, dan Keluar. Pengguna dapat memilih button Mulai Game jika ingin langsung memainkannya.

Pada Gambar 4.19 diatas dirancang dengan menggunakan aplikasi Adobe Animate dengan Bahasa Pemrograman *ActionScript 3* dengan beberapa tindakan sehingga terciptanya tampilan menu sebagai berikut:

- a. Pertama yaitu menentukan Font yang ingin digunakan adalah Calibri.
- b. Dibuat dengan *asset MPI Component* yang sudah disiapkan.
- c. Untuk menjalankan program tampilan awal dengan mengatur object pada asset dengan events Sistem dan Perintah dan Penampilan menu pada program ini.

3. Tampilan *Menu* Pengaturan

Pada tampilan pengaturan yaitu menampilkan fitur untuk mengatur *on/off* suara yang ada didalam *game* dan menekan tombol panah berwarna hitam untuk kembali ke menu awal.

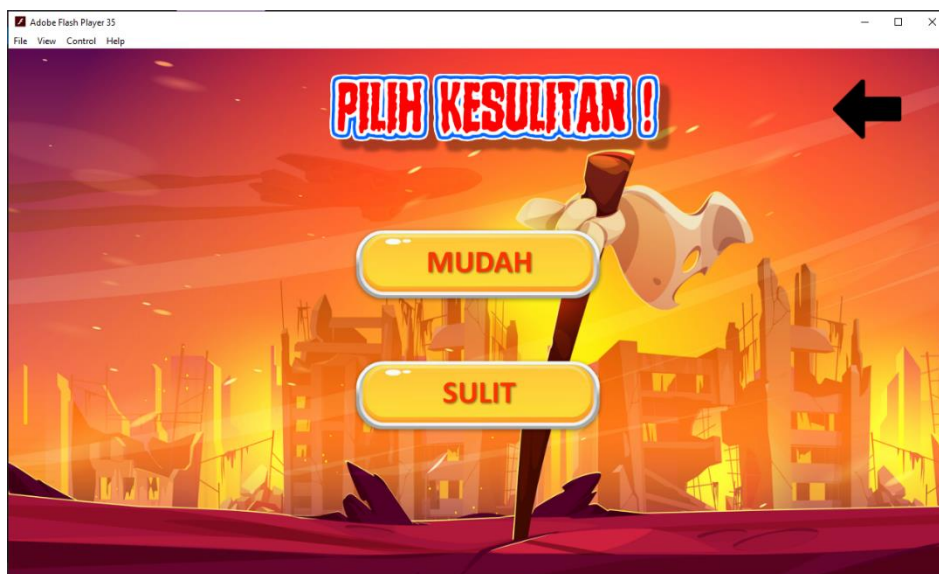


Gambar 4. 32 Tampilan *Menu* Pengaturan

Pada Gambar 4.20 diatas dirancang dengan menggunakan aplikasi Adobe Animate dengan Bahasa Pemrograman *ActionScript* 3 dengan beberapa tindakan sehingga terciptanya tampilan menu sebagai berikut:

- a. Pertama yaitu menentukan Font yang ingin digunakan adalah *Jesus Franco*.
 - b. Dibuat dengan *asset MPI Component* yang sudah disiapkan.
 - c. Untuk menjalankan program tampilan awal dengan mengatur object pada asset dengan events Sistem dan Perintah dan Penampilan menu pada program ini.
4. Tampilan Menu Tingkat Kesulitan

Pada Tampilan Tingkat Kesulitan terdapat dua pilihan yaitu button mode Mudah dan Sulit.



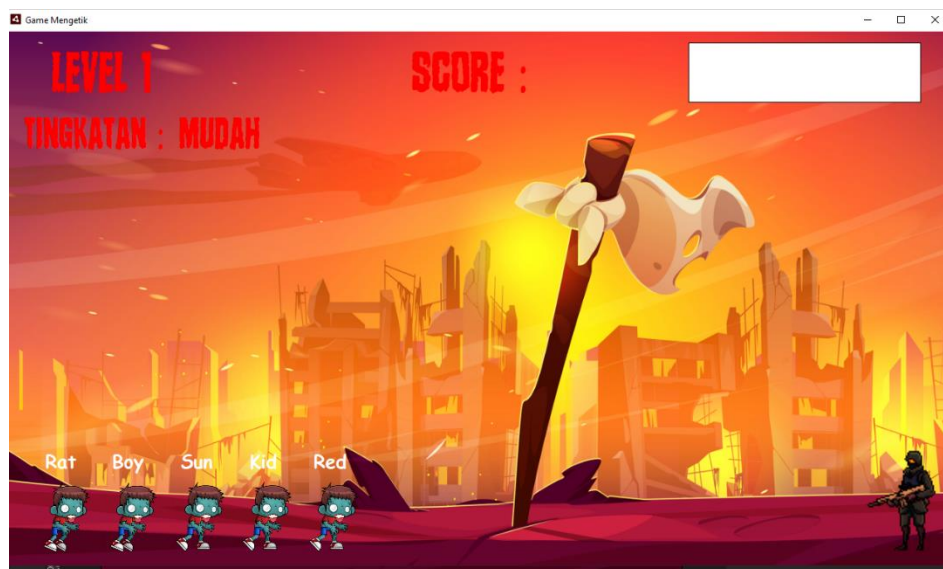
Gambar 4. 33 Tampilan *Menu* Tingkat Kesulitan

Pada Gambar 4.21 diatas dirancang dengan menggunakan aplikasi Adobe Animate dengan Bahasa Pemrograman *ActionScript* 3 dengan beberapa tindakan sehingga terciptanya tampilan menu sebagai berikut:

- a. Pertama yaitu menentukan Font yang ingin digunakan adalah Calibri.
- b. Dibuat dengan *asset MPI Component* yang sudah disiapkan.
- c. Untuk menjalankan program tampilan awal dengan mengatur object pada asset dengan events Sistem dan Perintah dan Penampilan menu pada program ini.

5. Tampilan *Level 1 Mudah*

Pada Tampilan *Level 1 mudah* yaitu menampilkan permainan dari objek yang telah diprogram sesuai dengan perintah yang diinginkan.



Gambar 4. 34 Tampilan *Level 1 Mudah*

Pada Gambar 4.22 diatas dirancang dengan menggunakan aplikasi Adobe Animate dengan Bahasa Pemrograman *ActionScript 3* dengan beberapa tindakan sehingga terciptanya tampilan menu sebagai berikut:

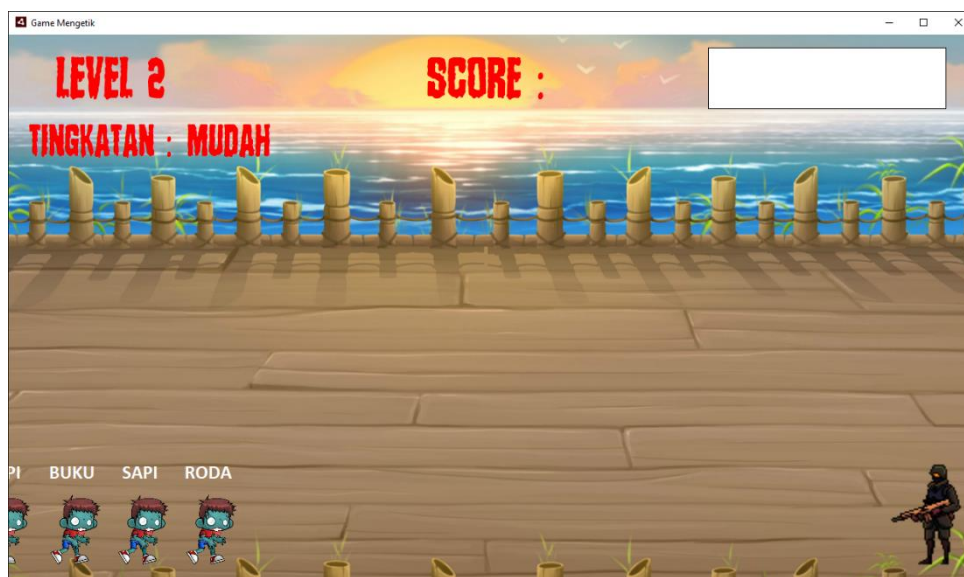
- a. Menentukan Font yang akan digunakan pada asset adalah *Jesus Franco*
- b. Dibuat dengan *asset MPI Component* yang sudah disiapkan.
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Background* sebagai latar dalam permainan.

d. Untuk menjalankan program *level 1* mudah dapat mengatur object pada asset (Zombie dan Penembak) dengan mengatur kontrolnya sebagai berikut:

1. *Zombie Events* (*fn_Delay*, *gerakZombie*, *hapuszombi*, *delayZombie*).
2. *Penembak Events* (*heroMC*).
3. Untuk Menjalan Keseluruhan diatur pada Kontrol Asset (*bankWords*, *zombieSpeed*, *typeTXT*, *kesulitankecepatan*).

4. Tampilan *Level 2* Mudah

Pada Tampilan *Level 2* mudah yaitu menampilkan permainan dari objek yang telah diprogram sesuai dengan perintah yang diinginkan



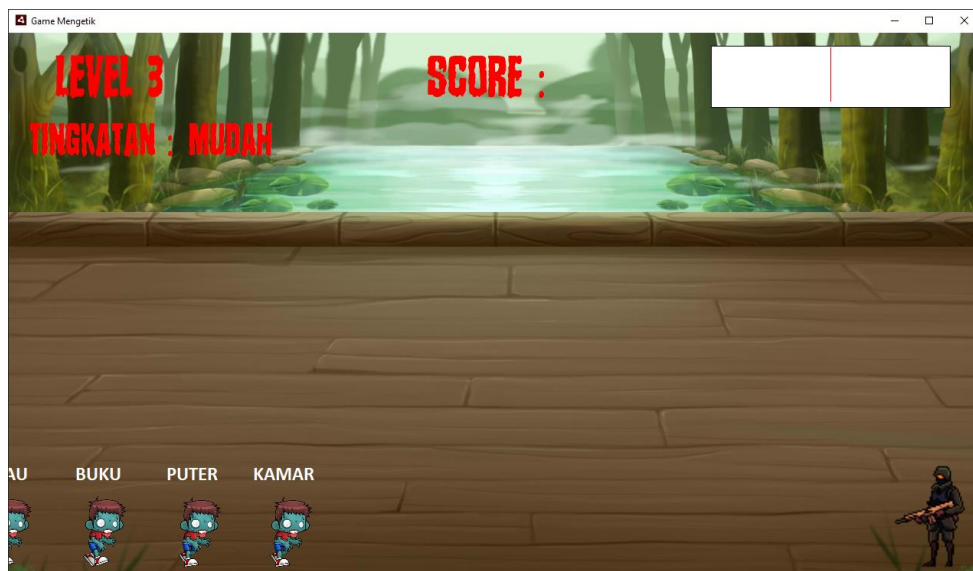
Gambar 4. 35 Tampilan *Level 2* Mudah

Pada Gambar 4.23 diatas dirancang dengan menggunakan aplikasi Adobe Animate dengan Bahasa Pemrograman *ActionScript 3* dengan beberapa tindakan sehingga terciptanya tampilan menu sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *Jesus Franco*

- b. Dibuat dengan *asset MPI Component* yang sudah disiapkan.
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Background* sebagai latar dalam permainan.
- d. Untuk menjalankan program *level 2* mudah dapat mengatur object pada asset (*Zombie* dan *Penembak*) dengan mengatur kontrolnya sebagai berikut:
 - 5. *Zombie Events* (*fn_Delay*, *gerakZombie*, *hapuszombi*, *delayZombie*).
 - 6. *Penembak Events* (*heroMC*).
 - 7. Untuk Menjalan Keseluruhan diatur pada Kontrol Asset (*bankWords*, *zombieSpeed*, *typeTXT*, *kesulitankecepatan*).
- 8. Tampilan Level 3 Mudah

Pada Tampilan *Level 3* mudah yaitu menampilkan permainan dari objek yang telah diprogram sesuai dengan perintah yang diinginkan.



Gambar 4. 36 Tampilan Level 3 Mudah

Pada Gambar 4.24 diatas dirancang dengan menggunakan aplikasi Adobe

Animate dengan Bahasa Pemrograman *ActionScript 3* dengan beberapa tindakan sehingga terciptanya tampilan menu sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *Jesus Franco*
- b. Dibuat dengan asset *MPI Component* yang sudah disiapkan.
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Background* sebagai latar dalam permainan.
- d. Untuk menjalankan program *level 3* mudah dapat mengatur object pada asset (*Zombie* dan *Penembak*) dengan mengatur kontrolnya sebagai berikut:

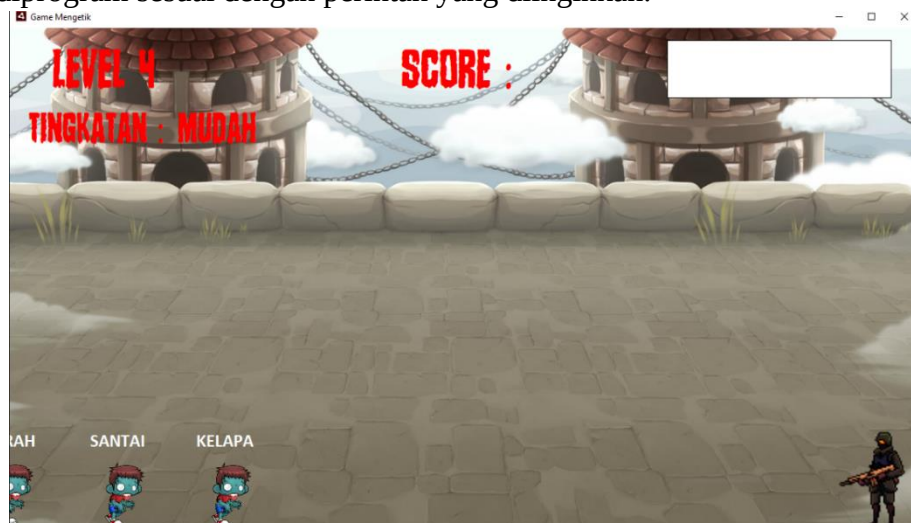
9. *Zombie Events* (*fn_Delay*, *gerakZombie*, *hapuszombi*, *delayZombie*).

10. *Penembak Events* (*heroMC*).

11. Untuk Menjalan Keseluruhan diatur pada Kontrol Asset (*bankWords*, *zombieSpeed*, *typeTXT*, *kesulitankecepatan*).

12. Tampilan *Level 4* Mudah

Pada Tampilan *Level 4* mudah yaitu menampilkan permainan dari objek yang telah diprogram sesuai dengan perintah yang diinginkan.



Gambar 4. 37 Tampilan *Level 4* Mudah

Pada Gambar 4.25 diatas dirancang dengan menggunakan aplikasi Adobe Animate dengan Bahasa Pemrograman *ActionScript 3* dengan beberapa tindakan sehingga terciptanya tampilan menu sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *Jesus Franco*
- b. Dibuat dengan *asset MPI Component* yang sudah disiapkan.
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Background* sebagai latar dalam permainan.
- d. Untuk menjalankan program *level 4* mudah dapat mengatur object pada asset (*Zombie* dan *Penembak*) dengan mengatur kontrolnya sebagai berikut:

1. *Zombie Events* (*fn_Delay*, *gerakZombie*, *hapuszombi*, *delayZombie*).
2. *Penembak Events* (*heroMC*).
3. Untuk Menjalan Keseluruhan diatur pada Kontrol Asset (*bankWords*, *zombieSpeed*, *typeTXT*, *kesulitankecepatan*).

13. Tampilan *Level 5* Mudah

Pada Tampilan *Level 5* mudah yaitu menampilkan permainan dari objek yang telah diprogram sesuai dengan perintah yang diinginkan.



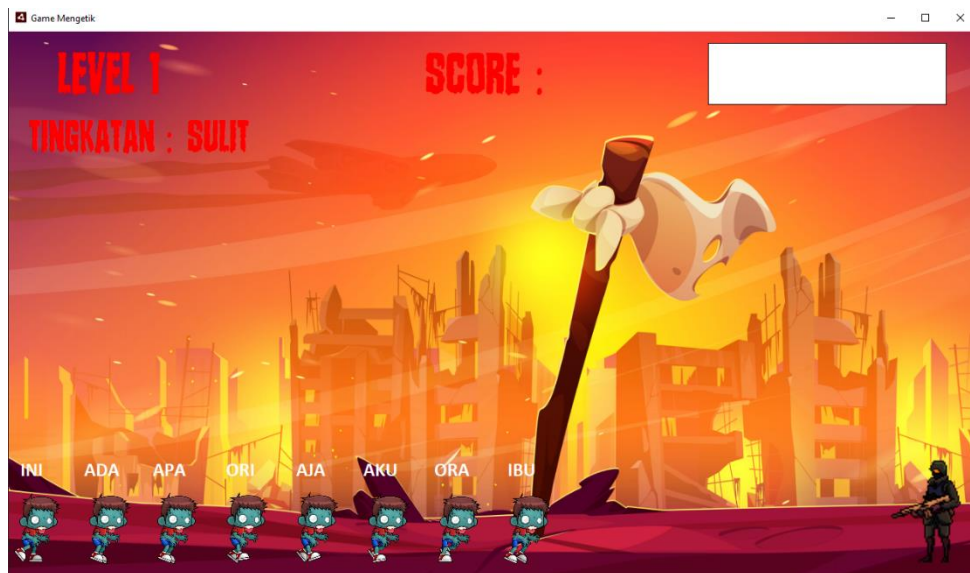
Gambar 4. 38 Tampilan *Level 5* Mudah

Pada Gambar 4.26 diatas dirancang dengan menggunakan aplikasi Adobe Animate dengan Bahasa Pemrograman *ActionScript 3* dengan beberapa tindakan sehingga terciptanya tampilan menu sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *Jesus Franco*
- b. Dibuat dengan asset *MPI Component* yang sudah disiapkan.
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Background* sebagai latar dalam permainan.
- d. Untuk menjalankan program *level 5* mudah dapat mengatur object pada asset (*Zombie* dan *Penembak*) dengan mengatur kontrolnya sebagai berikut:
 1. *Zombie Events* (*fn_Delay*, *gerakZombie*, *hapuszombi*, *delayZombie*).
 2. *Penembak Events* (*heroMC*).
 3. Untuk Menjalan Keseluruhan diatur pada Kontrol Asset (*bankWords*, *zombieSpeed*, *typeTXT*, *kesulitankecepatan*).

14. Tampilan *Level 1* Sulit

Pada Tampilan *Level 1* sulit yaitu menampilkan permainan dari objek yang telah diprogram sesuai dengan perintah yang diinginkan.



Gambar 4. 39 Tampilan *Level 1* Sulit

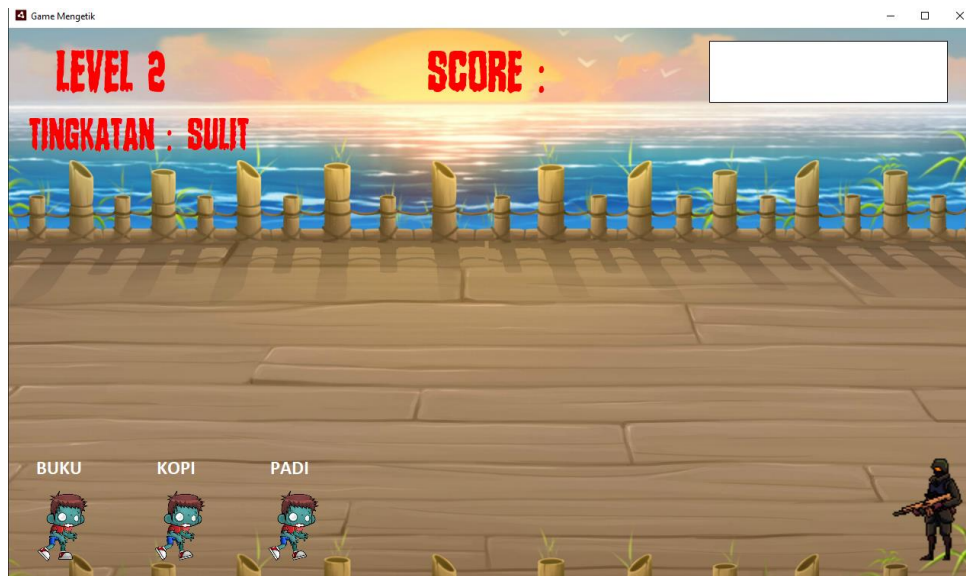
Pada Gambar 4.27 diatas dirancang dengan menggunakan aplikasi Adobe Animate dengan Bahasa Pemrograman *ActionScript 3* dengan beberapa tindakan sehingga terciptanya tampilan menu sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *Jesus Franco*
- b. Dibuat dengan *asset MPI Component* yang sudah disiapkan.
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Background* sebagai latar dalam permainan.
- d. Untuk menjalankan program *level 1* sulit dapat mengatur object pada asset (*Zombie* dan *Penembak*) dengan mengatur kontrolnya sebagai berikut:
 1. *Zombie Events* (*fn_Delay*, *gerakZombie*, *hapuszombi*, *delayZombie*).
 2. *Penembak Events* (*heroMC*).

3. Untuk Menjalan Keseluruhan diatur pada Kontrol Asset (*bankWords*, *zombieSpeed*, *typeTXT*, kesulitankecepatan).

15. Tampilan *Level 2* Sulit

Pada Tampilan *Level 2* sulit yaitu menampilkan permainan dari objek yang telah diprogram sesuai dengan perintah yang diinginkan.



Gambar 4. 40 Tampilan *Level 2* Sulit

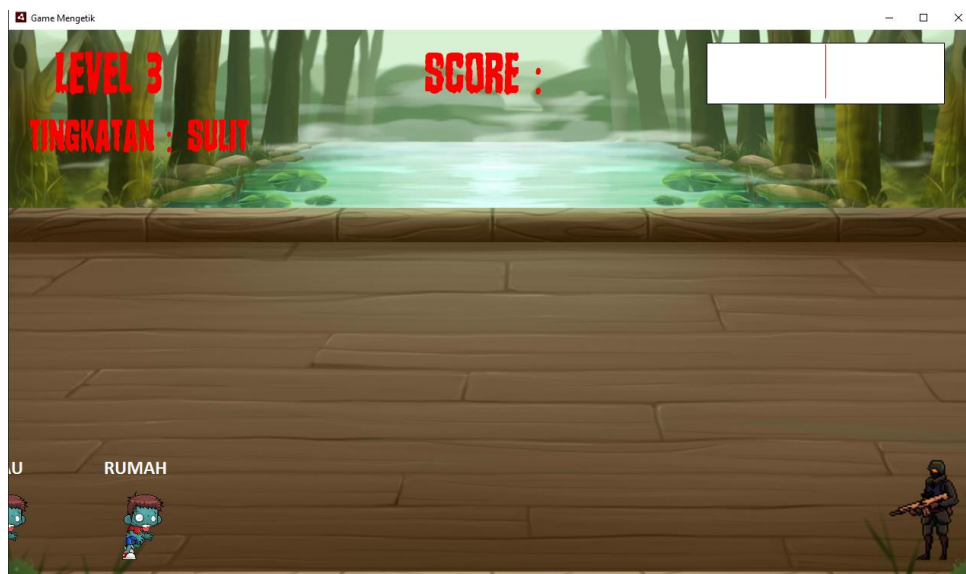
Pada Gambar 4.28 diatas dirancang dengan menggunakan aplikasi Adobe Animate dengan Bahasa Pemrograman *ActionScript 3* dengan beberapa tindakan sehingga terciptanya tampilan menu sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *Jesus Franco*
- b. Dibuat dengan asset *MPI Component* yang sudah disiapkan.
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Background* sebagai latar dalam permainan.
- d. Untuk menjalankan program *level 2* sulit dapat mengatur object pada asset (*Zombie* dan *Penembak*) dengan mengatur kontrolnya sebagai berikut:

1. *Zombie Events* (*fn_Delay*, *gerakZombie*, *hapuszombi*, *delayZombie*).
2. *Penembak Events* (*heroMC*).
3. Untuk Menjalan Keseluruhan diatur pada Kontrol Asset (*bankWords*, *zombieSpeed*, *typeTXT*, *kesulitankecepatan*).

16. Tampilan *Level 3* Sulit

Pada Tampilan *Level 3* sulit yaitu menampilkan permainan dari objek yang telah diprogram sesuai dengan perintah yang diinginkan .



Gambar 4. 41 Tampilan *Level 3* Sulit

Pada Gambar 4.29 diatas dirancang dengan menggunakan aplikasi Adobe Animate dengan Bahasa Pemrograman *ActionScript 3* dengan beberapa tindakan sehingga terciptanya tampilan menu sebagai berikut:

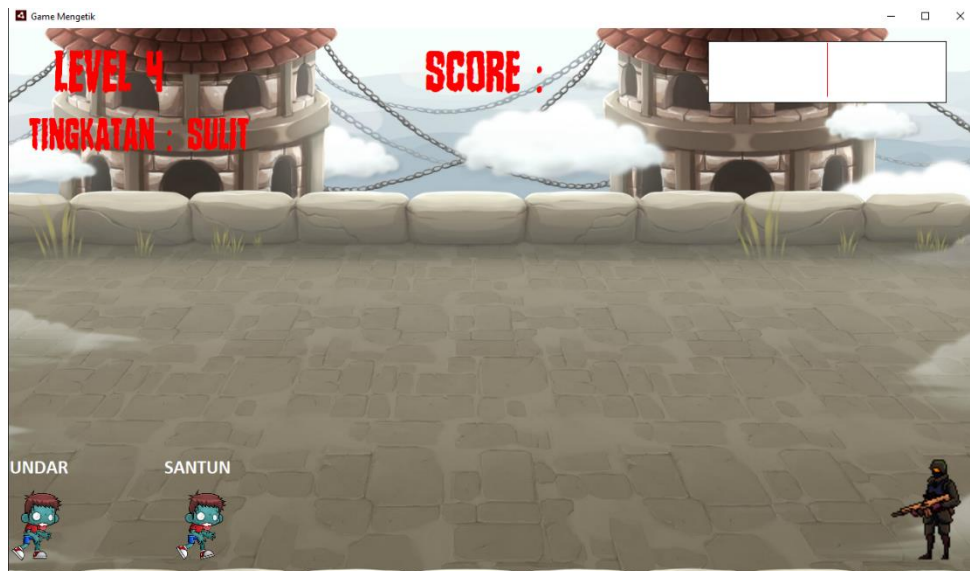
- a. Menentukan Font yang akan digunakan pada asset adalah *Jesus Franco*
- b. Dibuat dengan *asset MPI Component* yang sudah disiapkan.
- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *Background* sebagai latar dalam permainan.
- d. Untuk menjalankan program *level 3* sulit dapat mengatur object pada asset

(Zombie dan Penembak) dengan mengatur kontrolnya sebagai berikut:

1. *Zombie Events* (*fn_Delay*, *gerakZombie*, *hapuszombi*, *delayZombie*).
2. *Penembak Events* (*heroMC*).
3. Untuk Menjalan Keseluruhan diatur pada Kontrol Asset (*bankWords*, *zombieSpeed*, *typeTXT*, *kesulitankecepatan*).

17. Tampilan *Level 4 Sulit*

Pada Tampilan *Level 4* sulit yaitu menampilkan permainan dari objek yang telah diprogram sesuai dengan perintah yang diinginkan.



Gambar 4. 42 Tampilan *Level 4* Sulit

Pada Gambar 4.30 diatas dirancang dengan menggunakan aplikasi Adobe Animate dengan Bahasa Pemrograman *ActionScript 3* dengan beberapa tindakan sehingga terciptanya tampilan menu sebagai berikut:

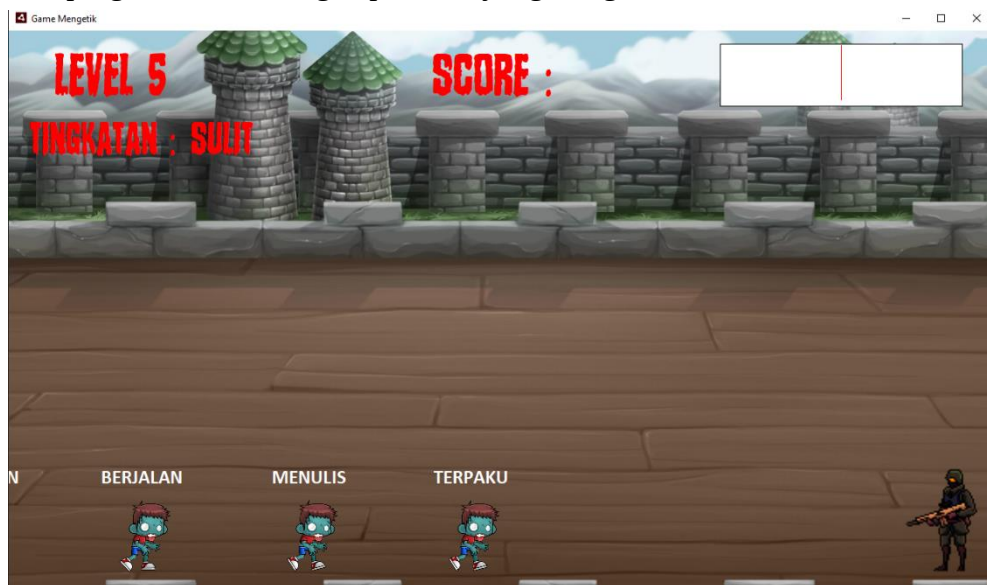
- a. Menentukan Font yang akan digunakan pada asset adalah *Jesus Franco*
- b. Dibuat dengan asset *MPI Component* yang sudah disiapkan.
- c. Menambahkan Asset *Sprite* yang telah disediakan sebagai karakter dan *Background* sebagai latar dalam permainan.

d. Untuk menjalankan program *level 4* sulit dapat mengatur object pada asset (*Zombie dan Penembak*) dengan mengatur kontrolnya sebagai berikut:

4. *Zombie Events* (*fn_Delay*, *gerakZombie*, *hapuszombi*, *delayZombie*).
5. *Penembak Events* (*heroMC*).
6. Untuk Menjalan Keseluruhan diatur pada Kontrol Asset (*bankWords*, *zombieSpeed*, *typeTXT*, *kesulitankecepatan*).

18. Tampilan *Level 5* Sulit

Pada Tampilan *Level 5* sulit yaitu menampilkan permainan dari objek yang telah diprogram sesuai dengan perintah yang diinginkan.



Gambar 4. 43 Tampilan *Level 5* Sulit

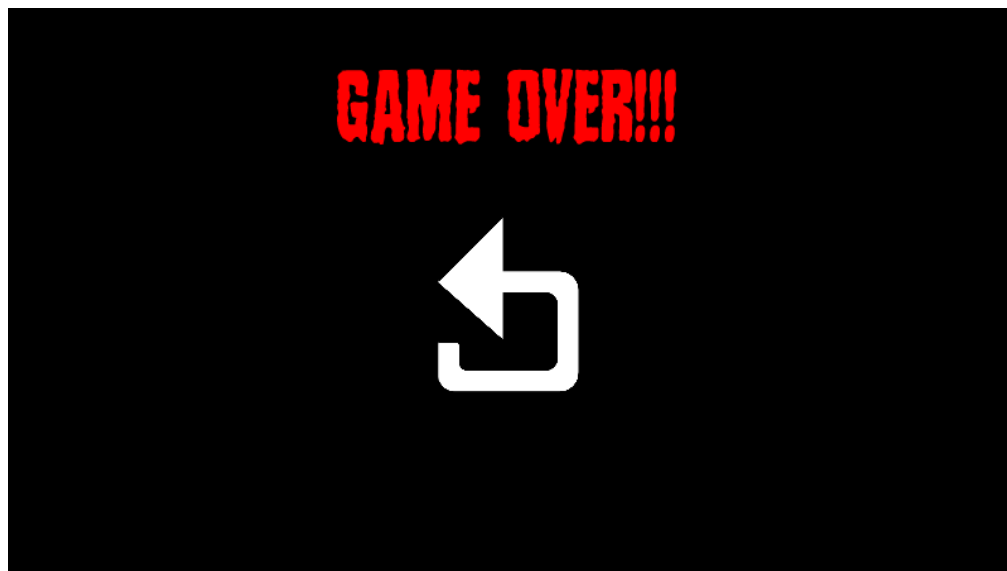
Pada Gambar 4.31 diatas dirancang dengan menggunakan aplikasi Adobe Animate dengan Bahasa Pemrograman *ActionScript 3* dengan beberapa tindakan sehingga terciptanya tampilan menu sebagai berikut:

- a. Menentukan Font yang akan digunakan pada asset adalah *Jesus Franco*
- b. Dibuat dengan asset *MPI Component* yang sudah disiapkan.

- c. Menambahkan *Asset Sprite* yang telah disediakan sebagai karakter dan *tabe;Background* sebagai latar dalam permainan.
- d. Untuk menjalankan program *level 5* sulit dapat mengatur object pada asset (*Zombie* dan *Penembak*) dengan mengatur kontrolnya sebagai berikut:
 - 1. *Zombie Events* (*fn_Delay*, *gerakZombie*, *hapuszombi*, *delayZombie*).
 - 2. *Penembak Events* (*heroMC*).
 - 3. Untuk Menjalan Keseluruhan diatur pada Kontrol Asset (*bankWords*, *zombieSpeed*, *typeTXT*, *kesulitankecepatan*).

19. Tampilan *Menu Kalah*

Gambar di atas merupakan Tampilan *Menu Kalah* dimana tampilan ini akan muncul ketika pengguna tidak berhasil menyelesaikan *level 1* sampai *level 5*. Dan Tombol Kembali berfungsi untuk kembali ke Menu Tampilan Awal.



Gambar 4. 44 Tampilan *Menu Kalah*

Pada Gambar 4.32 diatas dirancang dengan menggunakan aplikasi Adobe Animate dengan Bahasa Pemrograman *ActionScript 3* dengan beberapa tindakan sehingga terciptanya tampilan menu sebagai berikut:

- a. Pertama yaitu menentukan Font yang ingin digunakan adalah Calibri.
- b. Dibuat dengan *asset MPI Component* yang sudah disiapkan.
- c. Untuk menjalankan program tampilan awal dengan mengatur object pada asset dengan events Sistem dan Perintah dan Penampilan menu pada program ini.

20. Tampilan *Menu Menang*

Gambar di atas merupakan Tampilan *Menu Menang* dimana tampilan ini akan muncul ketika pengguna tidak berhasil menyelesaikan *level 1* sampai *level 5*. Dan Tombol Kembali berfungsi untuk kembali ke Menu Tampilan Awal.



Gambar 4. 45 Tampilan *Menu Menang*

Pada Gambar 4.33 diatas dirancang dengan menggunakan aplikasi Adobe Animate dengan Bahasa Pemrograman *ActionScript 3* dengan beberapa tindakan sehingga terciptanya tampilan menu sebagai berikut:

- d. Pertama yaitu menentukan Font yang ingin digunakan adalah Calibri.
- e. Dibuat dengan *asset MPI Component* yang sudah disiapkan.

- f. Untuk menjalankan program tampilan awal dengan mengatur object pada asset dengan events Sistem dan Perintah dan Penampilan menu pada program ini.

D. Implementasi

Implementasi sistem merupakan tahap penerapan dari suatu teknologi yang di desain siap untuk dioperasikan. Tahap ini merupakan terjemahan perancangan dari bab hasil analisis sebelumnya dalam suatu bahasa pemrograman.

Bahasa pemrograman yang digunakan untuk membuat APLIKASI GAME MENGETIK CEPAT SEBAGAI SARANA MELATIH KECEPATAN MENGETIK BERBASIS *DESKTOP* ini adalah *ActionScript*.

a. Kebutuhan Perangkat Keras

Spesifikasi minimum perangkat keras sebagai berikut :

Tabel 4. 3 Kebutuhan Perangkat Keras

Jenis	Spesifikasi
<i>Laptop / komputer</i>	<i>Asus</i>
<i>Processor</i>	<i>Intel® Core™ i5-7200U</i> <i>Processor (2.5GHz, 3M Cache)</i> <i>up to 3.10 GHz</i>
<i>Memory</i>	<i>8 GB RAM</i>

b. Kebutuhan perangkat lunak

Spesifikasi minimum perangkat lunak sebagai berikut:

Tabel 4. 4 Kebutuhan Perangkat Lunak

Sistem Operasi	Spesifikasi
<i>Windows</i>	<i>Windows 10, x64-bit</i>
<i>Adobe Animate</i>	<i>Adobe Animate 2023 Version 23.0.2</i>

E. Pengujian Sistem

1. Pengujian *Black Box*

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. 5 Pengujian Menu Utama

Test Faktor	Hasil	Keterangan
Membuka Aplikasi/ <i>Game</i>	√	Berhasil menampilkan halaman utama

Screenshot



Tabel 4. 6 Pengujian Menu Tingkat Kesulitan

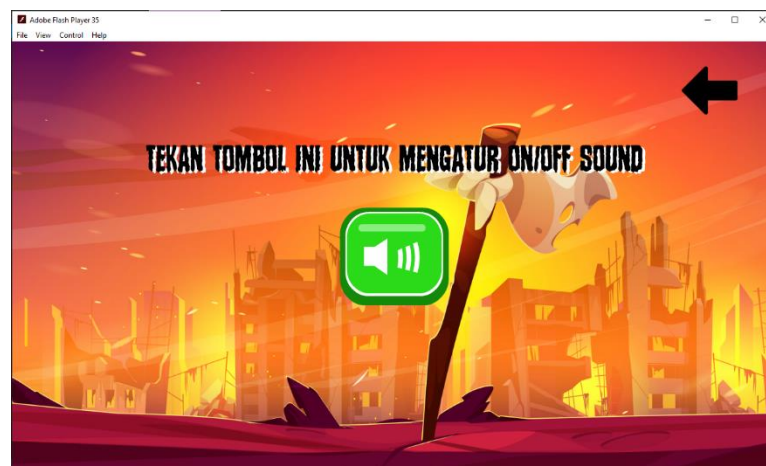
Test Faktor	Hasil	Keterangan
Tampilan Menu Tingkat Kesulitan	√	Berhasil menampilkan halaman tingkat kesulitan

Screenshot

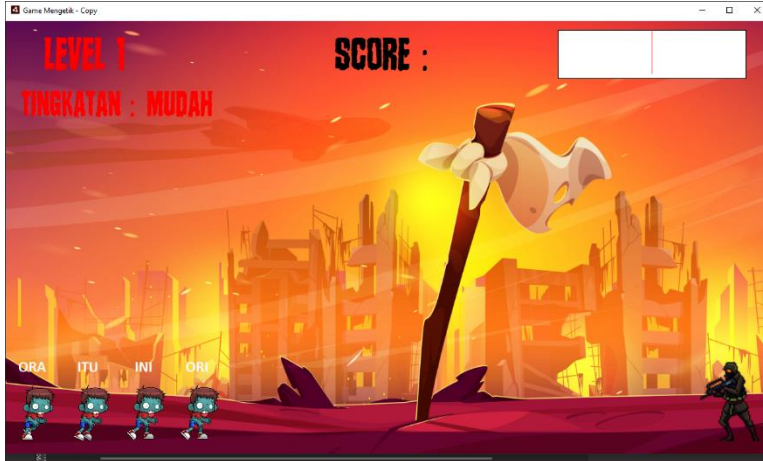
**Tabel 4. 7** Pengujian Menu Pengaturan

Test Faktor	Hasil	Keterangan
Tampilan Menu Pengaturan	√	Berhasil menampilkan halaman Pengaturan

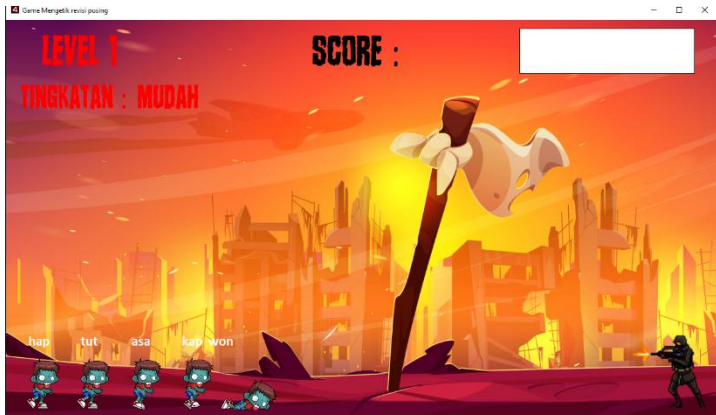
Screenshot



Tabel 4. 8 Pengujian Level 1

Test Faktor	Hasil	Keterangan
Tampilan Level 1	√	Berhasil menampilkan Level 1
Screenshot		
		

Tabel 4. 9 Pengujian *Gameplay*

Test Faktor	Hasil	Keterangan
Tampilan <i>Gameplay</i>	√	Berhasil menampilkan ketika <i>user</i> berhasil mengetik yang benar maka <i>zombie</i> akan dikalahkan.
Screenshot		
		

Tabel 4. 10 Pengujian *Game Over*

Test Faktor	Hasil	Keterangan
Tampilan Menu <i>Game Over</i>	√	Berhasil menampilkan menu <i>Game Over</i> yang akan muncul jika <i>user</i> gagal menyelesaikan level, dan terdapat tombol untuk <i>restart</i>
Screenshot 		

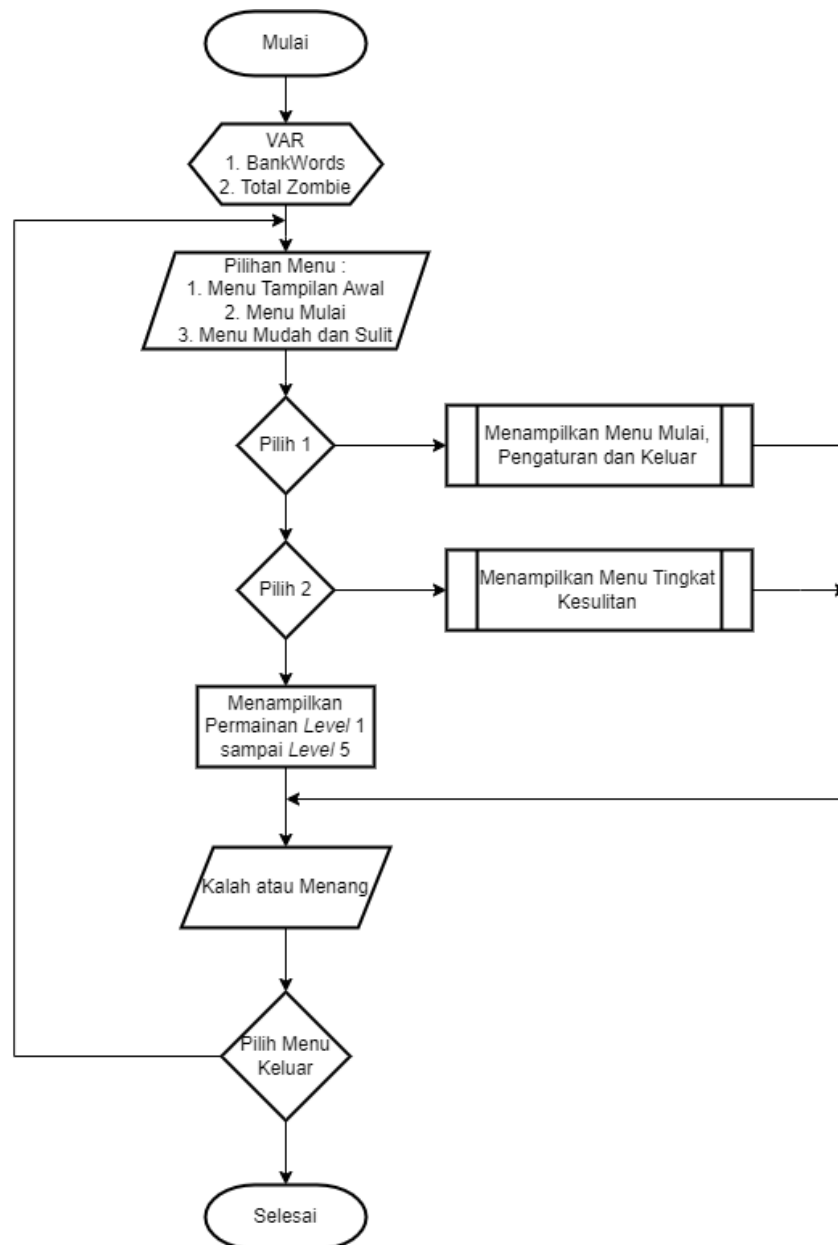
Tabel 4. 11 Pengujian Menang

Test Faktor	Hasil	Keterangan
Tampilan Menu Menang	√	Berhasil menampilkan menu Menang yang akan muncul jika <i>user</i> berhasil menyelesaikan level, dan terdapat tombol untuk <i>restart</i>

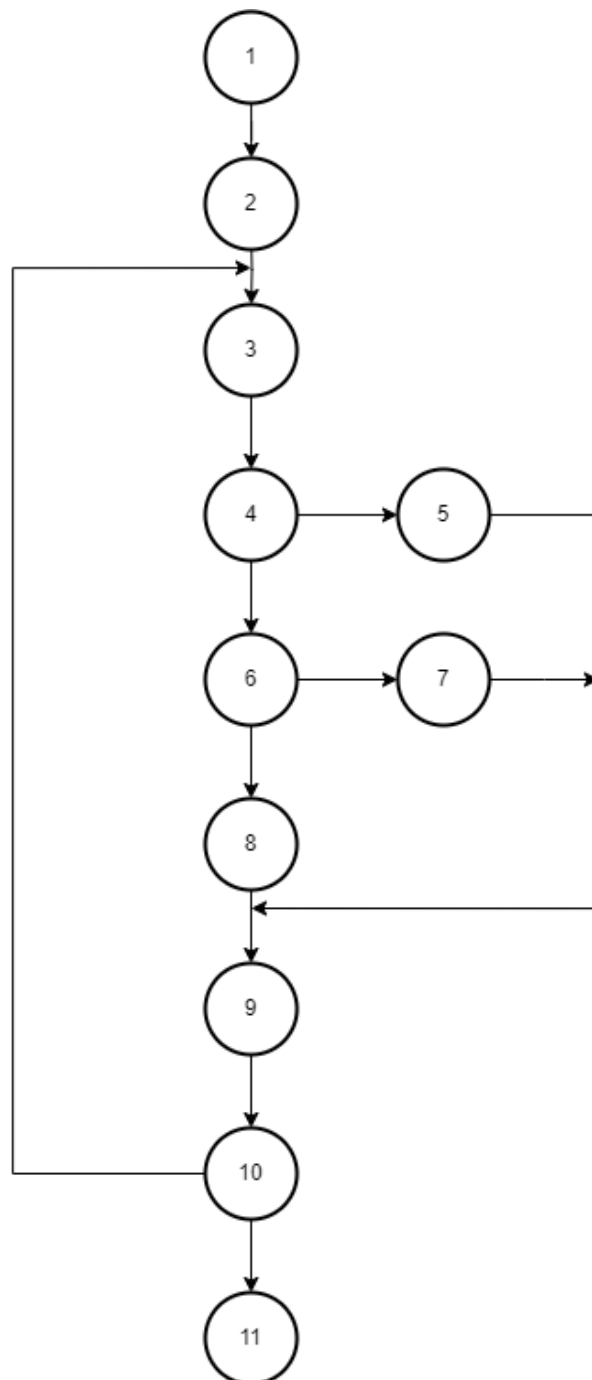
Screenshot



2. Pengujian *White Box* dan *Flowgraph* Aplikasi



Gambar 4. 46 *Flowchart* Aplikasi



Gambar 4. 47 Flowgraph Aplikasi

Menghitung Cyclomatic Complexity $V(G)$ dari Edge dan Node :

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

$E(\text{edge}) = 13$

$$N \text{ (node)} = 11$$

$$\begin{aligned} \text{Penyelesaian : } V(G) &= E - N + 2 \\ &= 13 - 11 + 2 \\ &= 4 \end{aligned}$$

1. Berdasarkan perhitungan *Cylomatic Complexity* dari *flowgraph* diatas memiliki

$$Region = 1$$

2. *Independent Path* pada *Flowgraph* diatas adalah :

$$Path\ 1 = 1 - 2 - 3 - 4 - 5 - 9 - 10 - 11$$

$$Path\ 2 = 1 - 2 - 3 - 4 - 6 - 7 - 9 - 10 - 11$$

$$Path\ 3 = 1 - 2 - 3 - 4 - 6 - 8 - 9 - 10 - 11$$

$$Path\ 4 = 1 - 2 - 3 - 4 - 6 - 8 - 9 - 10 - 3$$

Tabel 4. 6 Grafik *Matriks Aplikasi*

	1	2	3	4	5	6	7	8	9	10	11	E - 1
1		1										1 - 1 = 0
2			1									1 - 1 = 0
3				1								1 - 1 = 0
4					1	1						2 - 1 = 1
5									1			1 - 1 = 0
6							1	1				2 - 1 = 1
7									1			1 - 1 = 0
8									1			1 - 1 = 0
9										1		1 - 1 = 0
10			1								1	2 - 1 = 1
11												0
SUM (E+1)												3 + 1 = 4

BAB V

PENUTUP

A. Kesimpulan

Setelah dilakukan pengujian berdasarkan hasil penelitian, analisis, perancangan sampai dengan tahap pembuatan hingga selesai menurut sistem kerja dan analisis dari Game Arcade untuk mendukung mengetik cepat ini, maka dapat ditarik beberapa kesimpulan antara lain :

1. Melalui Game Arcade untuk mendukung mengetik cepat dapat memberikan alternatif hiburan serta untuk mendukung pembelajaran mengetik cepat.
2. Game yang menarik serta animasi pergerakan yang dapat memotivasi pemain untuk kembali mencoba menyelesaikan permainan ketangkasan game ini.

B. Saran

Berdasarkan pada pengujian yang telah dilakukan terhadap Game Arcade ini, masih terdapat kekurangan dan keterbatasan, maka disarankan :

1. Penambahan durasi level dan tingkat kesulitan serta identifikasi level yang harus ditambahkan agar mempermudah pengguna untuk mengetahui tahap level.
2. Lebih banyak variasi *game* dan fungsi resolusi agar dapat lebih nyaman dimainkan

DAFTAR PUSTAKA

- Abdullah, D., Reza, B., & Erliana, I. C. (n.d.). Game Edukasi Berbasis Role Playing Game Dengan Metode Finite State Machine.
- Adilla, A. M., & Windriyani, P. (2022). Pengembangan Typing Game Sebagai Sarana Melatih Meningkatkan Kecepatan Mengetik Berbasis Desktop. *Jurnal Mahasiswa Institut Teknologi dan Bisnis Kalbis*.
- Ansori. (2020). Retrieved from Pengertian UML (Unified Modeling Language) : Jenis, Tujuan, Notasi, dan Contohnya:
<https://www.ansoriweb.com/2020/03/pengertian-uml.html>
- Hanafri, M. I. (2015). Game Edukasi Tebak Gambar Bahasa Jawa Menggunakan Adobe Flash CS6 Berbasis Android. *Jurnal Sisfotek Global*.
- Henry, S. (2010). *Cerdas dengan Game: Panduan Praktis Bagi Orangtua dalam Mendampingi Anak Bermain Game*. Jakarta: PT. Gramedia Pustaka Utama.
- Kurniawati, W., & Wibawa, R. (2018). Efektivitas Media Game Mengetik Interaktif Berbasis Adobe Flash CS5 Terhadap Motivasi Belajar Siswa. *Jurnal Teknologi Pendidikan*.
- MyEduSolve. (2024, Mei 5). *MyEduSolve*. Retrieved from
<https://myedusolve.com/id/blog/pengertian-adobe-animate-kekurangan-dan-kelebihannya-yang-wajib-diketahui>
- Pamengas, W. M., Afirianto, T., & Wardhono S, W. (2020). Pengembangan Gim Edukasi Mengetik Menggunakan Metode Touch-Typing untuk Meningkatkan Kecepatan Mengetik. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*.
- Pranowo, G. (2011). *Kreasi Animasi Interaktif dengan ActionScript 3.0 pada Flash CS5*. Yogyakarta: ANDI.
- Radion, K. (2012). *Easy Game Programming Using Flash and ActionScript3.0*. Yogyakarta: ANDI.
- Setiyadi, D. (2016). Aplikasi Pembelajaran Mengetik Cepat Berbasis Game Edukasi Dengan Linear Congruent Method(LCM). *Bina Insani ICT Journal*.
- Yudhanto, P. A. (2010). Perancangan Promosi Produk Edu-Games Melalui Event. *Laporan Tugas Akhir*.