Game Real-Time Strategy Conquest War Berbasis Web Menggunakan Finite-State Machine

Jimmy Christian Tanian, Alexander Setiawan, Lily Puspa Dewi
Program Studi Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
.Jln. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
E-Mail: jimmychrtanian22@gmail.com, alexander@petra.ac.id, lily@petra.ac.id

ABSTRAK

Dengan semakin berkembangnya teknologi saat ini, game merupakan salah satu kegiatan yang digemari banyak orang. Mulai dari kalangan anak-anak maupun orang dewasa sekalipun tidak luput dari kegiatan bermain game salah satunya yaitu game strategy.

Dari sekian banyak game *real-time strategy* yang ada, masih banyak game *real-time strategy* saat ini yang terlalu kompleks dan sulit untuk dipahami oleh pemain. Maka, pada skripsi ini akan dibuat game real-time strategy yang mudah untuk dipahami dan dimainkan namun tidak menghilangkan unsur *strategy* pada game.

Agar game ini dapat menjadi semakin menarik, *artificial intelligence* (AI) *finite state machine* (FSM) ditambahkan pada game ini untuk mengatur pergerakan lawan agar dapat menambah dinamis pada game. Pada aplikasi game ini terdapat fitur-fitur yaitu *save/load* untuk menyimpan dan memuat data dari game yang sedang dimainkan dan fitur pengaturan suara untuk mengatur suara. Hasil dari 5 responden yang memberikan penilaian mendapatkan : 76% game kontrol, 68% grafik game, 80% suara game, 80% gameplay dan 80% kejelasan tutorial.

Kata Kunci: Kecerdasan buatan *Finite state Machine*, Game *Real-time Strategy*.

ABSTRACT

With the development of technology today, games are one of the activities that are popular for many people. Starting from children and even adults are also playing videogames. strategy game is the one of them.

Of the many real-time strategy games available, there are still many real-time strategy games today that are too complex and difficult for players to understand. In this thesis, a real-time strategy game will be made that is easy to understand and play but does not eliminate the strategy element in the game.

In order for this game to be even more interesting, Artificial Intelligence (AI) finite state machine (FSM) was added to this game to regulate the opponent's movements in order to make the game lively. In this game, there are features i.e. save/load to save and load data and sound settings to adjust the sound. The result from *5 respondents that give rating : 76% game control, 68% game graphic, 80% game sound , 80% gameplay dan 80% tutorial clearance.*

Keywords: *Artificial Intelligence Finite state Machine, Real-time Strategy Game.*

1. PENDAHULUAN

Dengan semakin berkembangnya teknologi saat ini, game merupakan salah satu kegiatan yang digemari banyak orang. Mulai dari kalangan anak-anak maupun orang dewasa sekalipun tidak luput dari kegiatan bermain game salah satunya yaitu game *strategy*. Game *Strategy* terbagi menjadi dua yaitu *Turn-Based Strategy* dan *Real-Time Strategy*.

Game *Turn-Based Strategy* merupakan game *strategy* yang dimana pemain menunggu giliran ketika bermain seperti catur dan game *board* modern lainnya. Salah satu contoh Game *Turn-Based Strategy* yang terkenal yaitu Civilization V. Game *Real-Time Strategy* dapat dimainkan tanpa menunggu giliran pemain lain. Hal ini yang membedakan Game *Turn-Based Strategy* dan *Real-Time Strategy*.

Game *Real-Time Strategy* sendiri memiliki ciri khas berupa permainan perang yang terdiri atas pembangunan kekuatan/negara, pengumpulan sumber daya, serta pembangunan dan pengaturan pasukan-pasukan tempur. Pada Game Real-Time Strategy biasanya memerlukan pemain untuk mengatur strategi untuk dapat mencapai tujuan tertentu seperti menguasai semua daerah lawan yang ada. Beberapa contoh game *Real-time Strategy* yang terkenal yaitu Warcraft III, Starcraft, dan Age of Empire.

Dari sekian banyak game *strategy* yang ada, masih banyak Game *Real-Time Strategy* saat ini yang terlalu kompleks dan sulit untuk dipahami oleh pemain. Maka, pada skripsi ini akan dibuat game Real-Time Strategy yang mudah untuk dipahami dan dimainkan namun tidak menghilangkan unsur *strategy* pada game. Agar game ini dapat menjadi semakin menarik, *Artificial Intelligence* (AI) *Finite State Machine* (FSM) ditambahkan pada game ini untuk mengatur pergerakan lawan agar dapat menambah dinamis pada game.

2. TINJAUAN PUSTAKA

2.1 Artificial Intelligence Finite State machine

Finite State Machine (FSM) atau *Finite-State Automaton* (FSA, plural: automata), *finite Automaton*, atau singkatnya *State Machine* adalah perhitungan model matematis.FSM terdiri dari serangkaian *state* yang menentukan pengambilan keputusan. Setiap *state* dapat berpindah ke *state* lainnya jika memenuhi kondisi yang telah ditentukan sebelumnya. FSM hanya dapat berada di satu state pada waktu tertentu. [5]

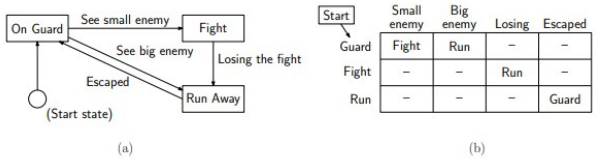
Misalnya, karakter dapat berperilaku lebih agresif ketika hp masih *full* dan kurang agresif saat karakter terluka. Sebagai contoh lain, seorang desainer mungkin ingin memiliki transisi karakter dalam berbagai *state* (patroli, mengejar, berkelahi, mundur) secara

berurutan atau ketika dipicu oleh *event* di game. Di setiap *state*, perilaku karakter mungkin sangat berbeda.

FSM dapat dimodelkan sebagai sebuah *directed graph*, di mana setiap *node* grafik sesuai dengan keadaan, dan setiap tepi diarahkan sesuai dengan suatu peristiwa, yang memicu sebuah perubahan status dan secara opsional beberapa tindakan terkait. Tindakan yang terkait mungkin termasuk hal-hal seperti memulai animasi, memainkan suara, atau mengubah kondisi permainan saat ini. Contoh, sebuah program bot prajurit NPC yang berperilaku sebagai berikut :

- Jika tidak melihat musuh, berjaga-jaga (*state On guard*).
- Saat berjaga-jaga (*state On guard*), jika melihat musuh kecil, maka bertarung(ke state fight), tetapi jika terlalu besar, maka lari (ke *state flee*).
- Jika sedang bertarung (*state fight*), jika tahu bahwa kalah dalam pertarungan, maka lari (ke *state flee*).
- Saat melarikan diri (*state flee*), jika telah melarikan diri ke titik di mana tidak ada ancaman, maka kembali berjaga (*state On guard*). [16]

Perilaku diatas dapat disandikan dalam bentuk FSM pada Gambar 1(a) :



Gambar 1. Sebuah state machine untuk bot warrior [16]

2.2 Game Real-Time Strategy

Game adalah sebuah sistem dengan konflik buatan antar pemainnya dengan disertai aturan-aturan, untuk mencapai tujuan yang diinginkan. (Salen & Zimmerman, 2003) Salah satu tujuannya adalah untuk mengalahkan pemain lain dalam game tersebut. Game memiliki beberapa komponen yang harus dipenuhi, antara lain: alat, peraturan, tujuan, dan pemain. [1]

Untuk klasifikasinya, game dibagi menjadi 3 yaitu :

1. *Sports* (sepakbola, basket, tenis, dll.)

Apa yang dimaksud dengan olahraga? Secara umum, pengertian olahraga adalah suatu aktivitas yang melibatkan pengerahan tenaga fisik dan pikiran yang dilakukan untuk melatih tubuh manusia, baik secara jasmani maupun secara rohani.

Olahraga bertujuan untuk meningkatkan derajat kesehatan seseorang, memperbaiki dan meningkatakan kinerja otak, meningkatkan kebugaran jasmani dan rohani, meningkatkan rasa percaya diri, dan mengurangi stres akibat rutinitadan beban pekerjaan sehari-hari. [18]

2. *Tabletop Games* (ular tangga, black jack, halma, dll.)

Pada dasarnya *tabletop* game adalah istilah yang lebih umum yang memiliki banyak padanan khusus. Setiap padanan khusus dari tabletop game bisa memiliki bentuk yang berbeda, namun maknanya tetaplah sama yaitu permainan yang dimainkan di atas meja.

Board games, Card games, Dice games, Paper and pencil games, Tabletop role-playing games, Tile-based games termasuk dalam padanan khusus dari *tabletop* game. Yang membedakan antara tiap padanan biasanya terdapat pada komponen yang digunakan dalam *tabletop* game. [22][7]

3. Video Game

Video game merupakan bagian dari game. Definisi dari video game adalah sebuah bentuk permainan elektronik berupa teks maupun gambar, yang melibatkan interaksi antara perangkat lunak permainan, orang yang memainkannya, dan dijembatani oleh perangkat keras pengolah permainan tersebut. Perangkat lunak game akan memberikan output berupa gambar atau teks yang ditampilkan melalui media (televisi, komputer, telepon seluler, dll.), kemudian pemain memberikan input berupa perintah yang disalurkan melalui perangkat keras permainan tersebut untuk ditampilkan kembali pada media. [8]

Game memiliki berbagai jenis genre yang umum digunakan, beberapa genre pada game beserta subgenrenya adalah:

- *Action* - Game yang menampilkan tantangan fisik, termasuk kerjasama mata-tangan dan waktu reaksi. Subgenre : *Platform game, Shooter game , Fighting game, Survival game.*
- *Adventure* - Game di mana pemain memainkan peran protagonis dalam cerita interaktif dengan memasukkan elemen eksplorasi dan pemecahan teka-teki dan tidak selalu mengenai tantangan fisik Subgenre : *Text adventures, Graphic adventures, Visual novel, Real-time 3D adventure.*
- *Puzzle* - Game dengan tujuan mencari tahu solusi dengan memecahkan teka-teki, menavigasi, serta memanipulasi dan mengkonfigurasi ulang objek. Subgenre : Logical, Physics game, Coding, Hidden object game, Tile-matching.
- *Role-playing* - Game di mana pemainnya memainkan peran karakter dalam latar berupa fiksi. Subgenre : *Role-playing, Action RPG, MMORPG, JRPG.*
- *Simulation* - Game yang umumnya dirancang untuk mensimulasikan aspek-aspek realitas nyata atau fiksi. Subgenre : *Construction & management simulation, Life simulation, Vehicle simulation*.
- *Strategy* - Game yang berfokus pada pemikiran dan perencanaan yang matang untuk mencapai kemenangan. Subgenre : *MOBA, Real-time strategy (RTS), Turn-based strategy (TBS), Tower defense. [6]*

Game *Real-Time Strategy* (RTS) adalah salah satu subgenre dari video game strategi karena jenis permainan ini melibatkan pengaturan perang tingkat strategi misalnya Pasukan, Peperangan, dan Diplomasi. Meskipun militer merupakan aspek dominan dalam RTS, RTS juga melibatkan aspek lain seperti ekonomi, pembangunan, dan diplomasi suatu Negara. [12]

Game RTS muncul secara bertahap dari eksperimen yang panjang untuk menggabungkan reaksi dan kecepatan dari game *action* dengan pertimbangan matang dan kedalaman dari game *strategy*. Sebelum pertengahan 1990-an, game *strategy* masih merupakan "*turn-based*", sementara game *action* menggunakan "*real-time*", Mereka adalah dua hal yang saling bertentangan dari desain game, yang sebelumnya berakar pada tradisi yang rumit yang kemudian menyatu menjadi sederhana. Hal ini yang membedakan game "*real-time*" dan "*turn-based*" *strategy*. [15]

Sebuah game dapat dikategorikan sebagai RTS jika memiliki beberapa elemen *gameplay* di bawah ini:

- Dapat memilih dan mengendalikan banyak unit dengan cara "*dragging and clicking*",

- Dapat menggerakkan unit untuk menguasai area penting seperti *resource*,
- Dapat menyerang dan menghancurkan unit musuh,
- Dapat membuat berbagai macam bangunan, seperti untuk sistem pertahanan ataupun untuk membuat pasukan unit
- Dapat terus menerus membuat dan mengontrol banyak unit sepanjang permainan berlangsung,
- Dapat mengumpulkan *resource* untuk dipakai mendirikan bangunan atau membeli unit,
- Dapat memberi perintah secara real time kepada unit yang dikontrol tanpa menunggu *timer* atau menunggu giliran dari *player* lain. [10]

Karenanya, ada beberapa genre game yang memiliki kesamaan konsep dan gameplay dengan template RTS namun tidak dianggap sebagai game RTS, seperti *City-Building, Construction and Management Simulation, dan Real-Time Tactics*. [9]

3. DESAIN GAME

Game *Conquest War* merupakan game *Real-Time Strategy* 2D tentang perebutan wilayah antara 2-4 pemain (1 pemain melawan 1-3 bot yang menggunakan *Artificial Intelligence* berbasis *finite-state machine* untuk mengatur pengambilan keputusan/strategi yang memiliki *state* untuk menyerang (dalam jangka waktu tertentu apabila jumlah pasukan lebih banyak dari pasukan lawan), bertahan (melindungi bangunan lain yang diserang oleh lawan), *full* (penuh), dan *idle* (diam)).

Pada aplikasi game ini terdapat fitur-fitur yaitu *save/load* untuk menyimpan dan memuat data dari game yang sedang dimainkan dan fitur pengaturan suara untuk mengatur suara.

Kondisi menang pada game apabila pemain menguasai seluruh daerah lawan (tidak termasuk bangunan netral). Kondisi kalah pada game apabila lawan menguasai seluruh daerah pemain. Sehingga pemain dan para lawan harus memperebutkan bangunan satu sama lain untuk menang.

Setiap *faction* yang bermain diawali dengan setidaknya 1 bangunan. Pemain dapat menyerang bangunan *faction* lain (*faction* lawan / *faction* netral) untuk meng-ekspansi daerah untuk mempercepat pertumbuhan para pasukan.

Jumlah pasukan bertambah 1 setiap detik untuk setiap bangunan, Bangunan *faction* netral tidak bisa membuat pasukan bahkan pada saat bangunan belum full sehingga lebih mudah untuk diserang. jumlah maksimal pasukan dalam bangunan adalah 20. Jika jumlah pasukan didalam sebuah bangunan melebihi batas maksimal maka bangunan akan berhenti membuat pasukan.

Pemain dapat memindahkan pasukan dari bangunan yang diplih ke bangunan lainnya dengan cara :

- Pemain memilih setidaknya 1 bangunan milik timnya dengan menekan tombol kiri mouse ke bangunan yang akan dipilih.
- Pemain dapat membatalkan pilihannya pada bangunan yang telah dipilih dengan menekan tombol kiri mouse ke bangunan yang telah dipilih.
- Pemain dapat mereset pilihan bangunan dengan menekan tombol kiri mouse pada daerah selain bangunan tim dan UI.
- Pemain dapat mereset pilihan bangunan dengan menekan tombol kiri mouse pada daerah selain bangunan tim dan UI. Jika pasukan menuju ke bangunan *faction* sendiri maka, jumlah pasukan dalam bangunan tujuan akan bertambah ketika pasukan yang dikirim sampai ke bangunan tujuan. Bila salah satu bangunan pemain diserang, kirim pasukan ke bangunan yang membutuhkan pasukan tambahan agar tidak direbut oleh lawan dengan mudah.

Jika pasukan menuju ke bangunan *faction* lawan maka, jumlah pasukan dan jumlah pasukan lawan dalam bangunan akan saling mengurangi ketika pasukan sampai ke bangunan tujuan. Apabila jumlah pasukan lawan pada bangunan tujuan < 0 maka, bagunan tersebut akan berpindah tangan ke tim pasukan yang terakhir kali menyerang bangunan tersebut.

4. HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai pengujian terhadap game yang telah dibuat. Pengujian yang dilakukan untuk game agar dapat memastikan sistem AI pada game dan kemampuan program untuk menge-*save/load* data dari *in-game* berjalan dengan baik sesuai dengan yang telah dijelaskan pada bab sebelumnya. Pengujian akan dilakukan pada PC/Laptop pada web Chrome versi terbaru.

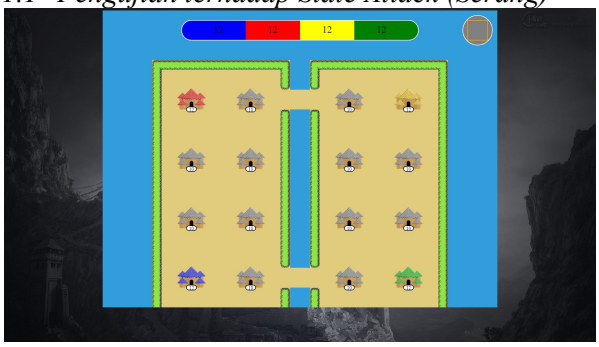
4.1 Pengujian Sistem AI

Game akan diuji kemampuannya terhadap Sistem *state* AI dilakukan untuk memastikan apakah *state* AI dapat berjalan dengan baik.

Tipe pengujian yang akan dilakukan antara lain:

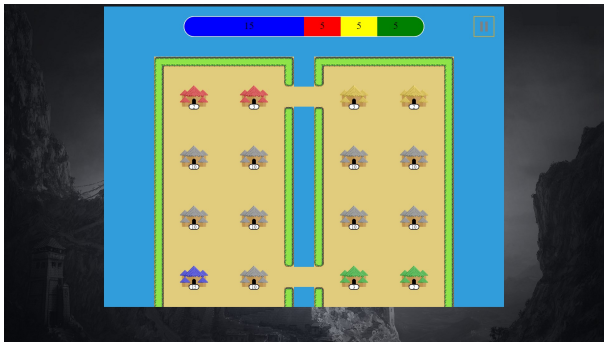
- Pengujian terhadap *State Attack* (Serang)
- Pengujian terhadap *State Defend* (Bertahan)
- Pengujian terhadap *State Full* (Penuh)

4.1.1 Pengujian terhadap State Attack (Serang)



Gambar 2. Pengujian sebelum State Attack

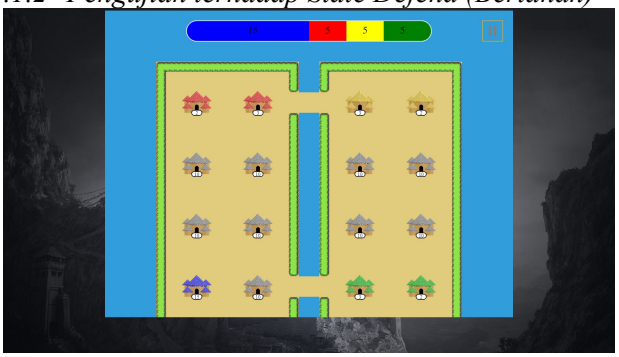
Gambar 2 menunjukkan kondisi sebelum menyerang bangunan lawannya dengan pasukan dari 1 atau lebih bangunan.



Gambar 3. Pengujian setelah State Attack

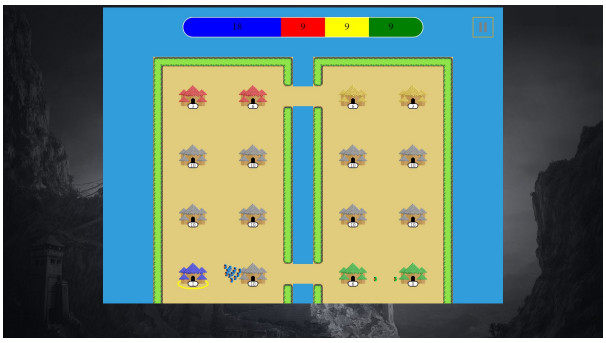
Gambar 3 menunjukkan kondisi setelah AI/Bot melakukan penyeragan dengan mengecek apakah *timer* sudah lewat dan apakah lawan yang bertahan lebih lemah.

4.1.2 Pengujian terhadap State Defend (Bertahan)



Gambar 4. Pengujian sebelum State Defend

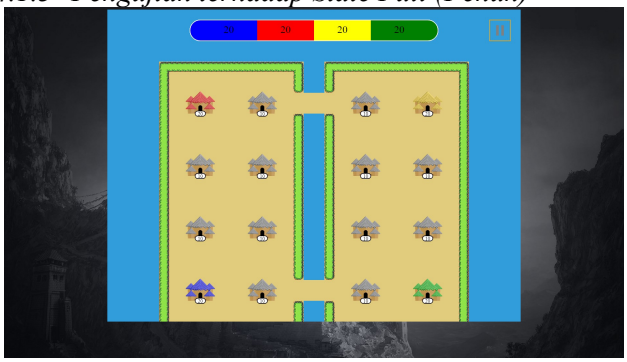
Gambar 4 menunjukkan kondisi sebelum bertahan dengan memanggil pasukan bantuan.



Gambar 5. Pengujian setelah State Defend

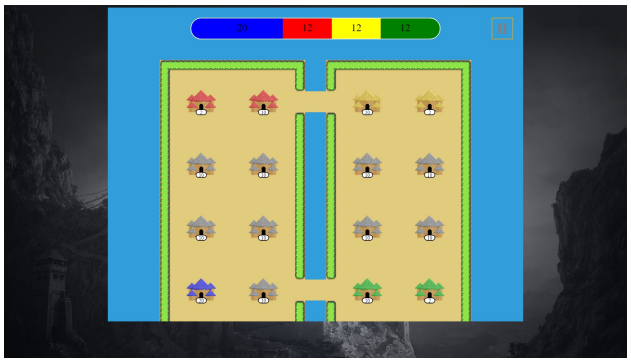
Gambar 5 menunjukkan kondisi setelah AI/Bot melakukan pertahanan dengan memanggil pasukan bantuan dengan mengecek apakah lawan melakukan *deploy/move*, apakah lawan sedang menyerang salah satu bangunan milik Bot tersebut dan apakah penyerang lebih lemah dari dirinya.

4.1.3 Pengujian terhadap State Full (Penuh)



Gambar 6. Pengujian sebelum State Full

Gambar 6 menunjukkan kondisi sebelum memindahkan pasukannya ke tempat yang lebih memerlukan bantuan pasukan.



Gambar 7. Pengujian setelah State Full

Gambar 7 menunjukkan kondisi setelah AI/Bot melakukan pemindahan pasukan ke bangunan lain miliknya dengan mengecek apakah bangunan tersebut *full*.

4.2 Pengujian Sistem Save/Load

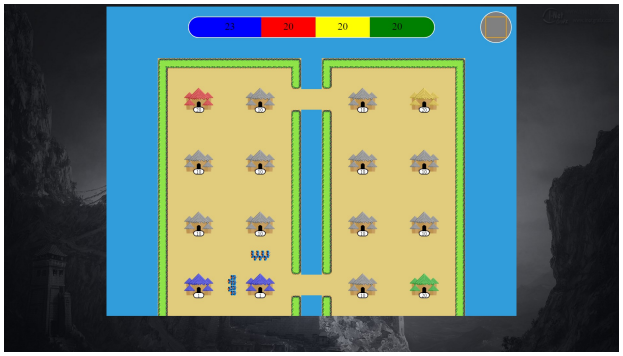
Game akan diuji kemampuannya dalam menge-*save* dan menge-*load* kondisi *In-game* terakhir.

Tipe pengujian yang akan dilakukan antara lain:

- Pengujian terhadap Cara Pembuatan *Save*
- Pengujian terhadap Cara Pembuatan *Load*

4.2.1 Pengujian Terhadap Cara Pembuatan Save

Subbab ini digunakan untuk menguji pembuatan sistem *save* yang digunakan oleh game.



Gambar 8. Interface sebelum *save* game

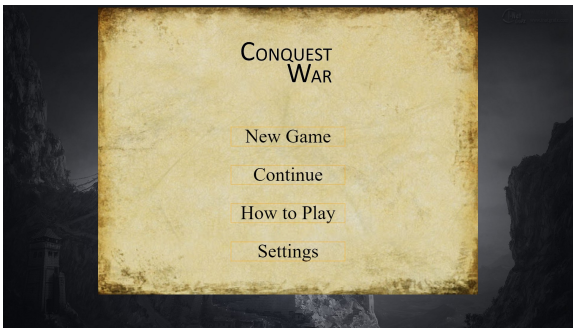
Gambar 8 diambil pada saat game berada pada *Interface In-game* setelah memlih *level* pada *select level* Gambar menujukan posisi *building, road,* dan *troop*.

Gambar 9 Interface setelah *save* game

Gambar 9 diambil sebelum menutup aplikasi dengan menekan tombol *pause* kemudian menekan tombol *exit* pada *interface pause*.

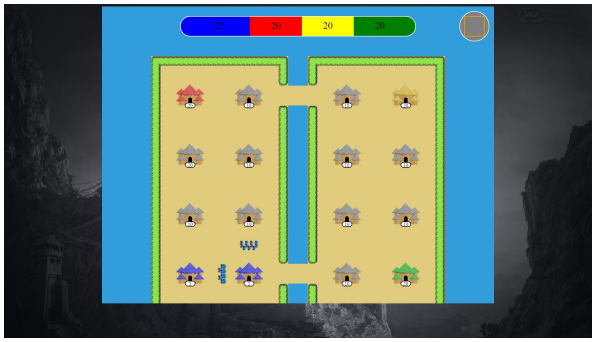
4.2.2 Pengujian Terhadap Cara Pembuatan Load

Subbab ini digunakan untuk menguji pembuatan sistem *load* yang digunakan oleh game.



Gambar 10. Interface sebelum *load* game

Gambar 10 diambil pada saat game berada *pada Interface main menu*. sebelum memlih *load* game pada *interface main menu*.



Gambar 11. Interface setelah *load* game

Gambar 11 diambil setelah game di-*load* dengan menekan tombol *continue* pada *interface main menu*.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian yang dilakukan, dapat disimpulkan beberapa hal sebagai berikut :

- Berdasarkan hasil penilaian kuisioner, dapat disimpulkan bahwa Game *Conquest War* ini cukup memuaskan dan telah memenuhi ekspetasi dari responden yang menadakan game ini cukup baik bagi responden baik dari sisi grafik, *sound*, *gameplay* dan kontrol. Dapat disimpulkan juga bahwa adanya *tutorial* pada game cukup membantu responden untuk memaikan Game *Conquest war*.
- Dapat disimpulkan bahwa penggunaan *Finite-State Machine* pada Game *Conquest War* cukup mampu untuk mengontrol AI/Bot pada game.

5.2 Saran

Beberapa hal yang dapat dijadikan saran untuk pengembangan game ini kedepannya sebagai berikut :

- Penambahan fitur seperti *upgrade*, *skill*, jenis bangunan, monster yang dapat ditaklukan, *mode* 2v2, tema game beragam agar membuat game semakin menarik.
- Penambahan fitur *multiplayer* agar dapat dimainkan oleh lebih banyak pamain.

6. REFERENSI

[1] Adams, E. 2013. Fundamentals of Game Design (3rd ed.). San Francisco: New Riders.

[2] Ariata 2020, Apa Itu HTML? Fungsi dan Cara Kerja HTML. Retrieved from https://www.hostinger.co.id/tutorial/apa-itu-html/#Jadi8230-Apa-Itu-HTML

[3] Baumgarten, R., Colton, S., Morris, M. 2009. Combining AI Methods for Learning Bots in Real -Time Strategy Game. London: Imperial college London.

[4] Belwariar, R. 2016. A* Search Algorithm. Retrieved from https://www.geeksforgeeks.org/a-search-algorithm/

[5] Bourg, David M., 2009. AI for Games Developers, California: O'Reilly Media.

[6] Christiantho, D. 2017. INI 8 Jenis Genre Game dan Sub Genre-nya. Retrieved from https://www.inigame.id/8-jenis-genre-game-dan-sub-genre-nya/

[7] Con, A. n.d.. What is Tabletop?. Retrieved November 14, 2020 from https://www.ameliacon.com/special-events/tabletop-gaming/what-is-tabletop/

[8] Djaouti D. et al., 2008. A Gameplay Definition through Videogame Classification. Internasional Journal of Computer Games Technology.

[9] Geryk, B. 2006. A History of Real-Time Strategy Games. USA: GameSpot.

[10] Hidayatullah, M. 2018. Ada Apa dengan Genre Real-Time Strategy?. Retrieved from https://esportsnesia.com/fokus/ada-apa-dengan-real-time-strategy/

[11] Kapalomen 2017. Pengertian dan Sejarah Html - Html 1. Retrieved from https://www.kapalomen.com/2017/09/pengertian-dan-sejarah-html.html

[12] Krisna, K. 2016. 10 Game RTS Real Time Strategy Terbaik. Retrieved from http://silkmask.blogspot.com/2016/10/10-game-rts-real-time-strategy-terbaik.html

[13] Kurniawan, R.2015. Pengertian Dan Sejarah JavaScript. Retrieved from https://ryankurniawan201.wordpress.com/2015/10/02/pengertian-dan-sejarah-javascript/

[14] Millington, I., Funge, J. 2009. Artificial Intelligence for Games Second Edition. USA:Morgan Kaufmann Publisher

[15] Moss, R. 2017. Build, gather, brawl, repeat: The history of real-time strategy games. Retrieved from https://arstechnica.com/gaming/2017/09/build-gather-brawl-repeat-the-history-of-real-time-strategy-games/

[16] Mount, D., Eastman, R. 2018. CMSC 425: Lecture 21 Artificial Intelligence for Games: Decision Making. Retrieved November 21, 2019 from https://www.cs.umd.edu/class/spring2018/cmsc425/Lects/lect21-ai-dec-making.pdf

[17] Nainggolan, R.F. 2017. Perancangan Dan Pembuatan Aplikasi Game Adventure Berbasis 2D Menggunakan Adobe Flash CS6 ActionScript 3.0. Retrieved from

http://repositori.usu.ac.id/bitstream/handle/123456789/3665/142606010.pdf

[18] Pengertian Olahraga: Tujuan, Manfaat, dan Jenis-Jenis Olahraga n.d.. Retrieved November 14, 2020 from https://www.maxmanroe.com/vid/umum/pengertian-olahraga.html

[19] Pengertian, Fungsi, Tujuan Dan Sejarah CSS. 2017. Retrieved from https://naimlogdasar.blogspot.com/2017/05/pengertian-fungsi-tujuan-dan-sejarah-css

[20] Salen, K, Zimmerman, E. 2003. Rules of Play : Game Design Fundamentals. Cambridge, MA: MIT Press.

[21] Santiago, E. 2017. A* Search Algorithm. Retrieved from http://amazingstudiotechnology.com/a-star-algorithm/

[22] Vagansza 2019. Jangan Bingung! Ini Bedanya Board Game, Card Game, dan Tabletop Game. Retrieved from https://boardgame.id/bedanya-board-game-card-game-tabletop-game/#:~:text=Pada%20 dasarnya%20tabletop%20game%20adalah,yang%20dimainkan%20di%20atas%20meja

[23] Zakaria n.d., Pengertian HTML dan Fungsinya yang Perlu Anda Ketahui Sebagai Pengguna Internet. Retrieved November 13, 2020 from https://www.nesabamedia.com/pengertian-dan-fungsi-html/