

APLIKASI QUIZ DENGAN KONSEP GAMIFICATION BERBASIS WEB MENGGUNAKAN RUBY ON RAILS & REACT.JS

Fajar Juliansyah Eka Putra¹, Suharjanto Utomo², Ariawan D. Rachmanto³, Samsul Budiarto⁴
Program Studi Teknik Informatika Universitas Nurtanio Bandung
email: ¹juliansyahfajar19@gmail.com

ABSTRAK

Perkembangan *technology ICT* telah berdampak terhadap perkembangan industri games, yang membuat para pendidik semakin kreatif dalam mendesain proses pembelajarannya. Beberapa perancang games telah mempelajari lebih dalam tentang ilmu psikologi atau ilmu lain yang mempelajari motivasi dan perilaku manusia. Gamification adalah pendekatan pembelajaran menggunakan elemen-elemen di dalam game atau video game dengan tujuan memotivasi para mahasiswa dalam proses pembelajaran dan memaksimalkan perasaan enjoy dan engagement terhadap proses pembelajaran tersebut, selain itu, media ini dapat digunakan untuk menangkap hal-hal yang menarik minat mahasiswa dan menginspirasinya untuk terus melakukan pembelajaran. Foursquare, SimCity dan Kahoot, mewakili gamified sistem yang menggabungkan pengguna dengan geo-location.[1] Karena syarat gamifikasi, harus mengandung fitur Point, Badges, Levels, Leaderboards, Challenges, Rewards, OnBoarding, dan Engagement loops. Dalam proses pembelajaran menggunakan gamifikasi, memberikan alternative untuk membuat proses belajar lebih menarik, menyenangkan dan efektif. Walaupun menggunakan mekanika permainan, menerapkan gamifikasi tidak selalu harus membuat sebuah games, tetapi bagaimana membuat pembelajaran lebih menyenangkan, membangun engagement dengan tanpa disadari oleh para pembelajar.

Kata Kunci: Gamifikasi, Motivasi, Engagement.

1. PENDAHULUAN

1.1 LATAR BELAKANG

Indonesia Perkembangan teknologi ICT telah berdampak terhadap perkembangan industri games, yang membuat para pendidik semakin kreatif dalam mendesain proses pembelajarannya. beberapa perancang games telah mempelajari lebih dalam tentang ilmu psikologi atau ilmu lain yang mempelajari motivasi dan perilaku manusia. tanam.

Dalam penelitian ini, penulis mencoba untuk menganalisis dan mendiskusikan faktor penentu motivasi dalam efektivitas pembelajaran dengan menggunakan gamifikasi dari perspektif kognitif sosial. Isu-isu utama yang dibahas adalah pertama membahas mengapa para peneliti dan praktisi perlu mempertimbangkan masalah motivasi dalam proses pembelajaran yang menggunakan multimedia. Kedua, teori motivasi yang relevan dengan proses pembelajaran yang secara khusus dapat diterapkan pada pembelajaran multimedia, pengajaran dan pengembangan

bahan ajar. Menurut penelitian Lee dan Hammer (Lee & Hammer, 2011), games dapat memberikan 3 keuntungan psikologi, yaitu kognitif, emosional dan sosial, sehingga dapat meningkatkan motivasi pemain dalam mempelajari suatu games.

Berkaitan dengan metode pembelajaran, peserta didik memiliki kesempatan untuk menggunakan sumber daya multimedia sangat termotivasi untuk menggunakannya. Penelitian di bidang multimedia pembelajaran sejauh ini telah difokuskan pada efektivitas metode pembelajaran dan desainnya saja. Namun, berdasarkan literatur, ditemukan bahwa teknologi multimedia tidak menimbulkan motivasi dalam pembelajaran yang unggul. Mahasiswa pada umumnya tidak suka untuk melakukan pekerjaan sehari-hari pada waktu yang lama, tetapi rela menghabiskan banyak waktu untuk bermain game. Karena hal ini maka timbul ide untuk menggabungkan pembelajaran dan game sehingga dapat memotivasi siswa untuk melakukan pembelajaran tersebut. Membungkus

pembelajaran dalam sebuah game. Pendidikan dan pembelajaran merupakan salah satu bagian dari perubahan tersebut yang saat ini menjadi faktor penting dalam membentuk sebuah peradaban bangsa dalam melahirkan perubahan dan strategi baru dibidang ilmu pengetahuan dan teknologi [2].

Penggunaan Pembangunan E-leaning ini, tentunya diharapkan mampu menghasilkan suatu sistem belajar yang lebih menyenangkan. Berdasarkan latar belakang diatas, maka penulis bermaksud membuat penelitian yang berjudul “**Aplikasi Quiz Dengan Konsep Gamifikasi Berbasis WEB Menggunakan Ruby on Rails & React.js**”.

1.2 RUMUSAN MASALAH

Dari latar belakang masalah diatas, maka rumusan masalah yang didapat yaitu sebagai berikut:

1. Bagaimana cara membuat pertanyaan quiz yang sesuai dengan kebutuhan audiens?
2. Bagaimana cara belajar menggunakan teknik gamifikasi?

1.3 TUJUAN PENELITIAN

1. Dapat memberikan pertanyaan quiz yang sesuai dengan audiens.
2. Membangun motivasi belajar dengan berkompetisi sesuai kecepatan waktu dalam menjawab pertanyaan.

2. LANDASAN TEORI

2.1 GAMIFIKASI

Gamifikasi adalah penggunaan dari teknik desain permainan, permainan berpikir dan permainan mekanik untuk meningkatkan *non-game* konteks.[5] Biasanya gamifikasi berlaku untuk *non-game* aplikasi dan proses, untuk mendorong orang untuk mengadopsi mereka, atau untuk mempengaruhi bagaimana mereka digunakan. Gamifikasi bekerja dengan membuat teknologi yang lebih menarik, dengan mendorong pengguna untuk terlibat dalam perilaku yang diinginkan, dengan menunjukkan jalan untuk penguasaan dan otonomi, dengan membantu untuk memecahkan masalah dan tidak menjadi gangguan, dan dengan mengambil keuntungan dari kecenderungan psikologis manusia untuk terlibat dalam *game*. Teknik ini dapat mendorong orang untuk melakukan pekerjaan mereka yang biasanya

membosankan, seperti menyelesaikan survei, belanja, mengisi formulir pajak, atau membaca situs web. Data yang tersedia dari situs-situs *gamified*, aplikasi, dan proses perbaikan menunjukkan potensi di daerah seperti keterlibatan pengguna, ROI, kualitas data, ketepatan waktu, atau belajar.[6]

Nick Pelling pertama kali menggunakan istilah gamifikasi (*gamification*) di tahun 2002 pada presentasi dalam acara TED (*Technology, Entertainment, Design*). Gamification adalah pendekatan pembelajaran menggunakan elemen-elemen di dalam *game* atau *video game* dengan tujuan memotivasi para mahasiswa dalam proses pembelajaran dan memaksimalkan perasaan enjoy dan engagement terhadap proses pembelajaran tersebut, selain itu media ini dapat digunakan untuk menangkap hal-hal yang menarik minat mahasiswa dan menginspirasi untuk terus melakukan pembelajaran. Gamifikasi adalah menggunakan unsur mekanik game untuk memberikan solusi praktikal dengan cara membangun ketertarikan (engagement) kelompok tertentu. Secara lebih detil[7] mendefinisikan gamifikasi sebagai konsep yang menggunakan mekanika berbasis permainan, estetika dan permainan berfikir untuk mengikat orang-orang, tindakan memotivasi, mempromosikan pembelajaran dan menyelesaikan masalah. Glover menyimpulkan bahwa gamifikasi memberikan motivasi tambahan untuk menjamin para peserta didik (learners) mengikuti kegiatan pembelajaran secara lengkap[8]. Engagement dapat diartikan sebagai kesediaan untuk berpartisipasi, Frederick mendefinisikan student engagement sebagai tindakan metakonstruksi yang meliputi keterlibatan perilaku, emosi dan kognitif siswa dalam belajar. Seperti halnya game yang mengizinkan para pemainnya untuk melakukan restart atau bermain ulang, membuat kesalahan-kesalahan yang dapat diperbaiki sehingga membuat para pemain tidak takut mengalami kegagalan dan meningkatkan keterikatannya terhadap game tersebut. Gamifikasi bekerja dengan membuat teknologi yang lebih menarik (Takahashi, 2010), mendorong pengguna untuk terlibat dalam perilaku yang diinginkan (Stuart,2010), menunjukkan jalan untuk penguasaan dan otonomi, membantu untuk memecahkan masalah dan tidak menjadi gangguan, dan

mengambil keuntungan dari kecenderungan psikologis manusia untuk terlibat dalam game (Radoff, 2011). Menurut Zichermann gamification adalah proses cara berpikir games dan mekanika games untuk melibatkan pengguna dan memecahkan masalah. Definisi yang lebih umum (Deterding,2011) gamifikasi adalah penggunaan elemen desain yang membentuk sebuah games dalam konteks non-games.

2.2 RUBY

Ruby adalah bahasa pemrograman dinamis berbasis skrip yang berorientasi objek. Tujuan dari *ruby* adalah menggabungkan kelebihan dari semua bahasa-bahasa pemrograman skrip yang ada di dunia. Ruby ditulis dengan bahasa pemrograman C dengan kemampuan dasar seperti *Perl* dan *Python*. [7].

2.3 RUBY ON RAILS

Ruby on Rails, disebut juga *Rails*, adalah sebuah kerangka kerja aplikasi web sumber terbuka yang berjalan via bahasa pemrograman *Ruby*. Ia merupakan kerangka kerja *full-stack*: memungkinkan pembuatan laman dan aplikasi yang mengumpulkan informasi dari *server* web, berkomunikasi dengan atau memanggil sebuah kueri dari *server* web, dan mengolah templat dari sumber. Sebagai hasilnya, *Rails* menyajikan fitur sistem *routing* yang independen dari *server* web.

Ruby on Rails menegaskan penggunaan pola-pola dan prinsip-prinsip rekayasa perangkat lunak terkenal, seperti pola active record, konvensi diatas konfigurasi (CoC), *don't repeat yourself* (DRY), dan *model-view-controller* (MVC). [8]

2.4 REACT (WEB BROWSER)

React (juga dikenal sebagai *React.js* atau *ReactJS*) adalah *library JavaScript open-source, front end* untuk membangun antarmuka pengguna atau komponen *UI*. Itu dikelola oleh *Facebook* dan komunitas pengembang individu dan perusahaan. *React* dapat digunakan sebagai basis dalam pengembangan satu halaman atau aplikasi seluler. Bagaimanapun, *React* hanya berhubungan dengan rendering data ke *DOM*, dan membuat aplikasi *React* biasanya membutuhkan penggunaan perpustakaan tambahan untuk manajemen dan perutean

status. *Redux* dan *React Router* adalah contoh *library* tersebut. [9]

2.5 POSTGRESQL

PostgreSQL adalah sebuah sistem basis data yang disebarluaskan secara bebas menurut Perjanjian lisensi *BSD*. Peranti lunak ini merupakan salah satu basis data yang paling banyak digunakan saat ini, selain *MySQL* dan *Oracle*. *PostgreSQL* menyediakan fitur yang berguna untuk replikasi basis data. Fitur-fitur yang disediakan *PostgreSQL* antara lain *DB Mirror*, *PGPool*, *Slony*, *PGCluster*, dan lain-lain.

PostgreSQL adalah sistem database yang kuat untuk urusan relasi, open source. Memiliki lebih dari 15 tahun pengembangan aktif dan sudah terbukti segala rancangan arsitekturnya telah mendapat reputasi tentang “kuat”, “handal”, “integritas data”, dan “akurasi data” . [10]

2.6 APPLICATION PROGRAMMING INTERFACE (API)

Antarmuka pemrograman aplikasi (Inggris: *application programming interface disingkat API*) adalah penerjemah komunikasi antara klien dengan *server* untuk menyederhanakan implementasi dan perbaikan *software*.

Bisa diartikan juga sebagai sekumpulan perintah, fungsi, serta protokol yang dapat digunakan oleh *programmer* saat membangun perangkat lunak untuk sistem operasi tertentu. *API* memungkinkan *programmer* untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi.

2.7 DOCKER

Docker adalah platform perangkat lunak yang memungkinkan untuk membuat, menguji, dan menerapkan aplikasi dengan cepat. *Docker* mengemas perangkat lunak ke dalam unit standar yang disebut kontainer yang memiliki semua yang diperlukan perangkat lunak agar dapat berfungsi termasuk pustaka, alat sistem, kode, dan waktu proses. Dengan menggunakan *Docker*, dapat dengan cepat menerapkan dan menskalakan aplikasi ke lingkungan apa pun dan yakin bahwa kode akan berjalan [19].

Docker berfungsi dengan menyediakan cara standar untuk menjalankan kode. *Docker* adalah sistem operasi untuk kontainer. Mirip dengan cara mesin virtual memvirtualisasi (menghilangkan kebutuhan untuk secara langsung mengelola) perangkat keras *server*, kontainer memvirtualisasi sistem operasi *server*. *Docker* diinstal di setiap server dan memberikan perintah sederhana yang dapat digunakan untuk membuat, memulai, atau menghentikan kontainer.

2.8 GITLAB CI CD

GitLab CI/CD adalah alat yang dibangun ke dalam *GitLab* untuk pengembangan perangkat lunak melalui metodologi berkelanjutan yang terdiri dari *Continuous Integration* (CI), *Continuous Delivery* (CD), *Continuous Deployment* (CD) [20].

Continuous Integration bekerja dengan mendorong potongan kode kecil ke basis kode aplikasi yang di-hosting di repositori *Git*, dan untuk setiap dorongan, akan menjalankan pipeline skrip untuk membuat, menguji, dan memvalidasi perubahan kode sebelum menggabungkannya ke dalam cabang utama.

Continuous Delivery dan *Deployment* terdiri dari CI selangkah lebih maju, menerapkan aplikasi Anda ke produksi di setiap dorongan ke cabang default repositori.

Metodologi ini memungkinkan Anda untuk menemukan bug dan kesalahan di awal siklus pengembangan, memastikan bahwa semua kode yang diterapkan ke produksi sesuai dengan standar kode yang Anda tetapkan untuk aplikasi Anda.

3. ANALISA DAN PERANCANGAN

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi pemasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Aplikasi yang dibuat merupakan aplikasi aplikasi *quiz* dengan konsep *gamification* berbasis web menggunakan *ruby on rails & react.js* yaitu

platform pembelajaran berbasis *game*. Dalam aplikasi ini pengguna bisa membuat dan menjawab berbagai kuis pilihan ganda secara *realtime*. Aplikasi ini nantinya dapat di akses menggunakan browser.

3.1 ANALISA SISTEM

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi pemasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Aplikasi yang dibuat merupakan aplikasi *aplikasi quiz* dengan konsep *gamification* berbasis web menggunakan *ruby on rails & react.js* yaitu platform pembelajaran berbasis *game*. Dalam aplikasi ini pengguna bisa membuat dan menjawab berbagai kuis pilihan ganda secara *realtime*. Aplikasi ini nantinya dapat di akses menggunakan *browser*.

3.1.1 Analisis Metode

Dalam penelitian *quiz* ini menerapkan metode *Marczewski's Gamification Framework*.



Gambar 1. *Marczewski's Gamification Framework*

3.2 ANALISA KEBUTUHAN

Pada bab ini akan dibahas tentang analisis dan perancangan yang digunakan oleh penulis dalam membuat aplikasi.

3.2.1 Spesifikasi Kebutuhan Fungsional

Di dalam membuat aplikasi media pembelajaran ini dibutuhkan perancangan sehingga mempermudah proses desain pembangunan aplikasi, kebutuhan fungsional tersebut yaitu sebagai berikut :

Tabel 1. Tabel Fungsional

No	Kebutuhan Fungsional	Deskripsi
1.	<i>Login</i>	Admin dapat <i>login</i> kedalam aplikasi.
2.	<i>Dashboard Quiz</i>	Halaman utama <i>quiz</i> saat admin berhasil <i>login</i> . <i>Dashboard</i> isinya apa?
3.	<i>Create Quiz</i>	Admin dapat membuat <i>quiz</i> untuk nantinya di jawab <i>player</i> .
4.	<i>Delete Quiz</i>	Admin mampu menghapus <i>quiz</i> yang telah di buat.
5.	<i>Update Quiz</i>	Admin mampu mengubah <i>quiz</i> yang telah di buat.
6.	<i>Detail Quiz</i>	Admin mampu melihat detail dari <i>quiz</i> . Didalam detail <i>quiz</i> terdapat <i>create, update & delete question</i>
7.	<i>Create Question</i>	Admin mampu membuat <i>question</i> . 1 (Satu) <i>quiz</i> terdiri dari beberapa <i>question</i>
8.	<i>Session Play Quiz</i>	Mampu memulai sesi dari <i>quiz</i> yang di pilih.

3.2.2 Spesifikasi Kebutuhan Perangkat Keras

Berdasarkan studi literatur dan percobaan aplikasi terhadap proyek-proyek pengembang *Framework Ruby on rails* dan *React.js*, maka diperoleh spesifikasi minimum perangkat keras yang dibutuhkan oleh pihak pengembang untuk teknologi *Ruby on Rails* dan *React.js* sebagai berikut ini :

1. *Processor* Intel core 2 atau setara.
2. Sistem Operasi *Windows 7*
3. RAM minimal 2GB
4. Ruang sisa hardisk minimal 25 GB

3.2.3 Spesifikasi Software Development Tool

Perangkat lunak digunakan dalam sebuah sistem merupakan kumpulan perintah-perintah yang diberikan kepada perangkat keras agar saling berinteraksi untuk melakukan suatu tugas. Perangkat lunak yang dibutuhkan oleh pihak pengembang untuk membangun aplikasi adalah sebagai berikut :

1. Sistem operasi *Windows 10*
2. *Visual Studio Code*

3. *Adobe Photoshop*
4. *Firebase*
5. *WSL (Windows Subsystem for Linux)*
6. *Gitlab*
7. *Figma*

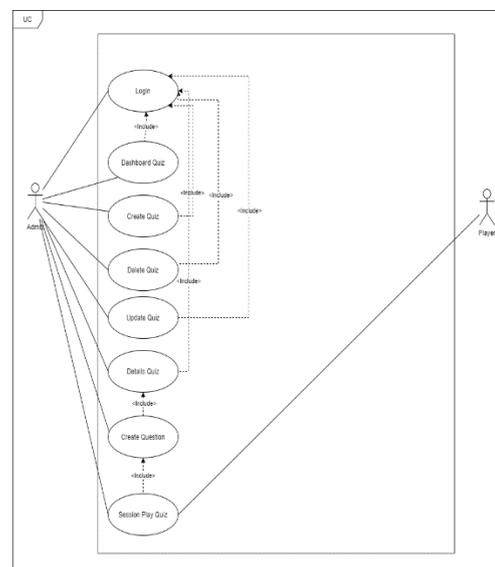
3.2.4 Spesifikasi Pengguna

Analisis pengguna dimaksud untuk mengetahui siapa saja pengguna yang terlibat *system*. Pengguna yaitu siswa dan *public* yang dapat mengerti dan memahami android sehingga dapat menggunakan aplikasi yang akan dibangun, karena aplikasi ini ditunjukkan agar siswa dapat tertarik untuk mengerjakan *quiz*.

3.3 PERANCANGAN SISTEM

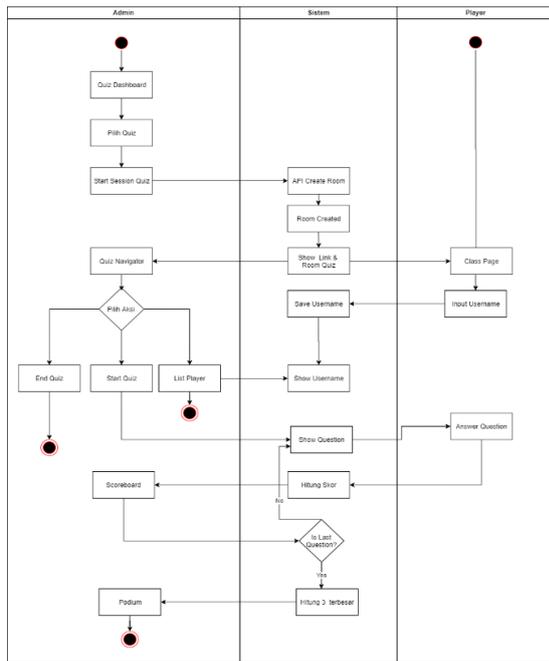
3.3.1 Use Case Diagram

Use Case Diagram ini digunakan untuk menggambarkan hubungan sejumlah *external actor* dengan *Use Case* yang terdapat dalam *system*. *Use Case Diagram* ini hanya menggambarkan keadaan lingkungan *system* yang dapat dilihat dari luar oleh *actor*. Pada *Use Case* ini terdapat dua *actor*, yaitu *user* dan *system*.



Gambar 2. *Use Case Diagram*

3.3.2 Diagram Activity



Gambar 3. Diagram Activity

4. IMPLEMENTASI DAN PENGUJIAN

4.1 IMPLEMENTASI PERANGKAT PEMBANGUNAN

Perangkat lunak yang digunakan pada sistem komputer yang digunakan untuk implementasi pada aplikasi adalah sebagai berikut :

a. Perangkat keras

Perangkat keras yang digunakan untuk mengimplementasikan program aplikasi yang dibangun adalah perangkat keras komputer PC atau *notebook* dengan spesifikasi yang disebutkan dibawah ini, yaitu :

1. *Processor* Intel i5-8250U.
2. *Memory* RAM 8 GB
3. *Hardisk* 1 TB
4. *VGA* MX150

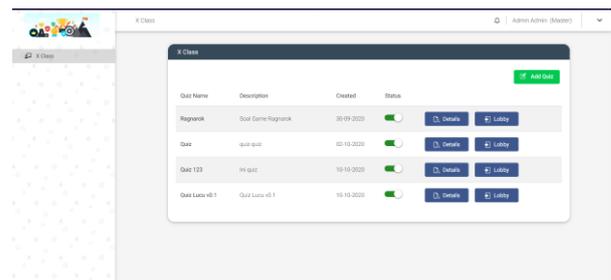
b. Perangkat lunak

Perangkat lunak yang digunakan untuk membangun aplikasi ini adalah sebagai berikut:

1. Sistem Operasi Windows 10.
2. Ruby.
3. Ruby on Rails.
4. React.js.
5. Firebase.
6. Postgresql.

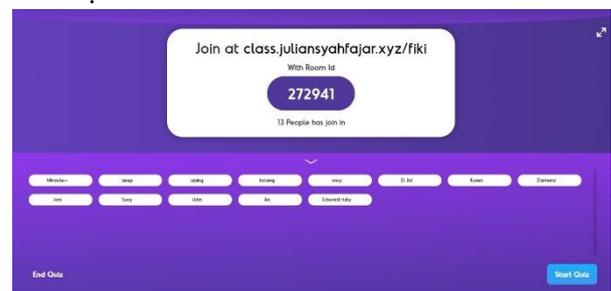
7. Visual Studio Code
8. WSL
9. NginX
10. Gitlab
11. Docker
12. Figma
13. Adobe Photoshop
14. Chrome

4.2 IMPLEMENTASI ANTAR MUKA



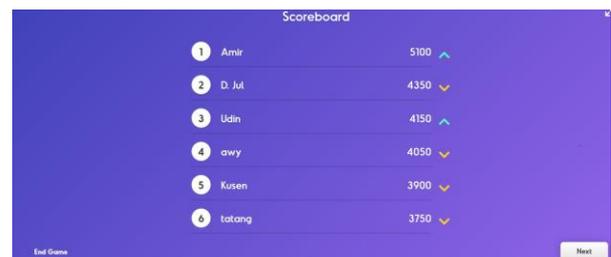
Gambar 4. Implementasi Antarmuka Dashboard Quiz

Pada halaman ini, user bisa membuat quiz mengubah status quiz, melihat detail quiz maupun ke lobby untuk memulai quiz.



Gambar 5. Implementasi Antarmuka Navigator

Semua player akan mengakses *link* dan *room* yang muncul di halaman navigator maka nama *user* akan otomatis muncul pada *list player*. Nama *user* yang muncul di *list player* itu adalah semua list player yang akan berpartisipasi pada quiz tersebut.



Gambar 6. Implementasi Antarmuka Scoreboard

Setelah player menjawab *question score* akan di hitung dan di urutkan secara *descending* atau dari urutan skor terbesar ke terkecil. Kemudian akan di tampilkan pada halaman *scoreboard*.



Gambar 7. Implementasi Antarmuka Podium

Jika question sudah habis, maka skor akhir akan di hitung dan sistem akan mencari dengan 3 skor terbesar, kemudian sistem akan menampilkan 3 user skor terbesar pada podium

4.3 PENGUJIAN SISTEM

Pengujian sistem dilakukan dengan tujuan untuk menemukan kesalahan atau kekurangan pada perangkat lunak yang telah dibangun. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak atau tidak. Dalam penelitian ini pengujian yang dilakukan terhadap sistem yaitu pengujian secara fungsional (alpha) dan beta. Metode yang digunakan dalam pengujian ini adalah pengujian blackbox yang berfokus pada fungsional dari sistem yang dibangun.

4.3.1 Pengujian Admin Panel

Pengujian Admin Panel merupakan pengujian fungsionalitas untuk Admin Panel yang telah diterapkan dalam aplikasi.

Tabel 2. Hasil Pengujian Admin Panel

No	Kasus/Diuji	Skenario Uji	Hasil yang diharapkan	Hasil Pengujian
1	Login	Mengisi Email dan Password	Verifikasi Email dan Password success	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil

2	Dashboard Quiz	Login Sukses	Mengarahkan user ke halaman Dashboard Quiz & menampilkan List Quiz	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
3	Create Quiz	Mengisi data Quiz	Data berhasil di simpan ke firebase	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
4	Delete Quiz	Mengubah status quiz menjadi draft kemudian klik tombol delete	Data berhasil di hapus dari firebase	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
5	Detail Quiz	Memilih Tombol Detail Quiz	Data quiz & data question berhasil di tampilkan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
6	Update Quiz	Memilih tombol Edit	Data berhasil di rubah dari firebase	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
7	Add Question	Memilih tombol Add Question	Data berhasil di simpan ke firebase	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
8	Lobby	Memilih tombol Lobby	Tombol Play Quiz muncul. History session quiz sebelumnya muncul (Jika Quiz sebelumnya pernah di mainkan)	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
9	Session Quiz	Memilih tombol Play Quiz	User di arahkan kehalaman navigator	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil

4.3.2 Pengujian Navigator

Pengujian Navigator dilakukan untuk mengetahui apakah Navigator yang di tunjukan mengalami kesalahan atau tidak.

Tabel 3. Hasil Pengujian Navigator

No	Kasus/Diuji	Skenario Uji	Hasil yang diharapkan	Hasil Pengujian
1	Show Link Class	Memilih tombol Play Quiz pada Admin Panel	Menampilkan link Class Page yang nantinya akan di akses oleh player	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil

2	Show Room	Memilih tombol <i>Play Quiz</i> pada Admin Panel	Menampilkan link <i>Room yang</i> nantinya akan di akses oleh player	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
3	List Player	Memilih tombol <i>List Player</i>	Menampilkan <i>List Player</i> yang akan berpartisipasi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
4	Scoreboard	<i>Question</i> yang muncul telah di jawab oleh <i>player</i>	Menampilkan skor dan mengurutkannya secara descending	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
5	Podium	Semua <i>Question</i> telah terjawab oleh user	Menampilkan halaman <i>Podium</i> dan menampilkan player dengan skor 3 terbesar	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil

4.3.3 Pengujian Class Page

Pengujian *Class Page* dilakukan untuk mengetahui apakah *Class Page* yang di tunjukan mengalami kesalahan atau tidak.

Tabel 4. Hasil Pengujian Class Page

No	Kasus/Diuji	Skenario Uji	Hasil yang diharapkan	Hasil Pengujian
1	<i>Input Room</i>	User mengakses link yang muncul pada navigator & menginput room	User berhasil join room & data di simpan pada <i>firebase</i>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
2	<i>Input Username</i>	Mengisi username	Data username berhasil di simpan pada <i>firebase</i> & data username muncul pada navigator	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
3	<i>Answer Question</i>	User bisa memilih jawaban	Jawaban dari user bisa di simpan ke <i>firebase</i> . Kemudian hasilnya tampil dihalaman <i>Scoreboard</i>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

5. KESIMPULAN DAN SARAN

5.1 KESIMPULAN

1. Pemanfaatan aplikasi *quiz* ini dapat digunakan sesuai dengan kebutuhan audiens.
2. Pemanfaatan dan Penggunaan *quiz* ini dapat digunakan untuk berkompetisi sesuai dengan kecepatan waktu audiens menjawab pertanyaan.

5.2 SARAN

Adapun saran yang dapat penulis berikan untuk pengembangan dari sistem yang telah dibuat antara lain:

1. Fitur-fitur yang ada pada aplikasi *quiz* lebih banyak menerapkan elemen - elemen gamification.
2. Aplikasi *quiz* dengan menerapkan metode gamification dibangun dalam platform mobile.

DAFTAR PUSTAKA

1. Jusuf, Heni, 2016. "Penggunaan Gamifikasi dalam Proses Pembelajaran". (JOINTECS) Jurnal TICOM Vol. 5
2. Farozi, Mohamad, 2016. "Rancang Bangun Website gamifikasi Sebagai Strategi Pembelajaran dan Evaluasi". (Seminar Nasional Teknologi Informasi dan Multimedia) ISSN : 2302-3805
3. Irwan Suprianto, Fajar Pradana, Fitra Abdurrachman Bachtiar, 2019. "Pengembangan Aplikasi E-Learning Dengan Menerapkan Metode Gamification".(Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer) e-ISSN: 2548-964X
4. Lee, J. & H. J., 2011. Gamification in education: What, how, why bother?. Academic Exchange Quarterly.
5. Badoystudio, <https://badoystudio.com/metode-waterfall/> (Diakses tanggal 5 Maret 2020 pukul 21:00 WIB)
6. Venturebeat, <https://venturebeat.com/2010/09/30/gamification-gets-its-own-conference/> (Diakses tanggal 10 September 2020 pukul 16:00 WIB)
7. Edition.Cnn, <http://edition.cnn.com/2010/TECH/web/0>

- 9/30/web.checkin.onetruefan/ (Diakses tanggal 10 Maret 2020 pukul 20:00 WIB)
8. Ruby, <https://www.ruby-lang.org/en/> (Diakses tanggal 10 Maret 2020 pukul 21:00 WIB)
 9. Ruby on Rails, <https://rubyonrails.org/> (Diakses tanggal 11 Maret 2020 pukul 12:00 WIB)
 10. React.js, <https://reactjs.org/> (Diakses tanggal 11 Maret 2020 pukul 16:00 WIB)
 11. Firebase, <https://firebase.google.com/> (Diakses tanggal 11 Maret 2020 pukul 16:00 WIB)
 12. Postgresql, <https://www.postgresql.org/> (Diakses tanggal 11 Maret 2020 pukul 16:00 WIB)