

**Oleh:**

**Yayang Rahmadina NRP. 4210161001**

**Dosen Pembimbing:**

**Mohamad Safrodin, B.Sc., MT.**

**NIP.** **197708262008121002**

**Fahim Nur Cahya Bagar, S.Kom, M.T.**

**NIP. 2000000200**

**SISTEM MANAJEMEN ASET UNTUK PENGEMBANGAN GAME BERBASIS CLOUD**

***ASSET MANAGEMENT SYSTEM FOR GAME DEVELOPMENT BASED ON CLOUD***

**PROYEK AKHIR**

**PROGRAM STUDI TEKNOLOGI GAME**

**DEPARTEMEN TEKNOLOGI MULTIMEDIA KREATIF**

**POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

**2020**

****

**PROYEK AKHIR**

**SISTEM MANAJEMEN ASET UNTUK**

**PENGEMBANGAN GAME BERBASIS CLOUD**

**Yayang Rahmadina**

**NRP. 4210161001**

**Dosen Pembimbing :**

**Mohamad Safrodin, B.Sc.,MT.**

**NIP. 197708262008121002**

**Fahim Nur Cahya Bagar, S.Kom., MT.**

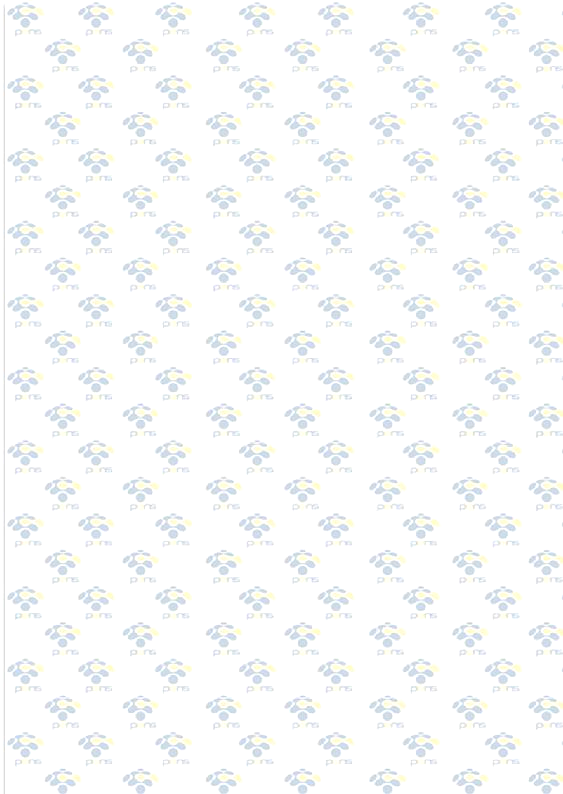
**NIP. 2000000200**

**PROGRAM STUDI D4 TEKNOLOGI GAME**

**DEPARTEMEN TEKNOLOGI MULTIMEDIA KREATIF**

**POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

**2020**

**SISTEM MANAJEMEN ASET UNTUK PENGEMBANGAN GAME BERBASIS CLOUD**

*Oleh :***Yayang Rahmadina  
NRP 4210161001**

Proyek Akhir ini Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Terapan Komputer (S.Tr.Kom)

Di

Politeknik Elektronika Negeri Surabaya

2020

Disetujui Oleh :

Dosen Pembimbing

**Mohamad Safrodin, B.Sc., MT.**NIP. 197708262008121002

**Fahim Nur Cahya Bagar, S.Kom., MT.**NIP. 2000000200

Tim Penguji:

**Halimatus Sa’dyah, S.Kom.,M.Kom.**NIP. 199007012015042001

**Rizky Yuniar Hakkun, S.Kom.,M.T.**NIP. 198106222008121003

**Kholid Fathoni, S.Kom., MT**NIP. 198012262008121003

Mengetahui:

**Ketua Program Studi D4 Teknologi Game Departemen Teknologi Multimedia Kreatif**

**Rizky Yuniar Hakkun, S.Kom.,M.T.**NIP. 198106222008121003

|  |  |  |
| --- | --- | --- |
| Mengetahui:  **Ketua Program Studi D4 Teknologi Game Departemen Teknologi Multimedia Kreatif**  **Mohamad Safrodin, B.Sc., MT.**  NIP. 197708262008121002 |  | |
|  |  |

# ABSTRAK

Game menjadi salah satu media yang banyak digemari oleh masyarakat baik dimainkan di perangkat pc, smartphone, maupun konsol. Oleh karena itu, banyak industri dan komunitas game gencar mengembangangkan produknya. Industri game dalam membangun produk game, menggunakan kode yang disediakan oleh programmer, namun sebagian besar apa yang dilihat dan dialami pemain tidak didefinisikan dalam kode, melainkan aset yang diproduksi. Dan tidak jarang dalam satu game yang di produksi dapat memiliki ratusan aset untuk membentuk satu produk game. Dengan meningkatnya jumlah aset dan orang-orang yang terlibat dalam proyek game, mengelola aset game secara manual akan mengambil porsi yang banyak dari proyek, baik waktu maupun uang. Untuk mengurangi hal tersebut, perlu untuk memanajemen asset dan menerapkan sistem untuk menyimpan dan mengelola semua aset game. Web service dipilih sebagai solusi tepat sebagai fasilitas penyedia layanan manajemen asset. Web service akan menyediakan fitur pengunggah asset kemudian arsitektur pada game akan bekerja mengelola asset dan menghasilkan data yang dapat diimplementasikan pada game engine apa saja. RESTful API adalah cara yang paling umum yang digunakan mengatasi pertukaran data antara server dan client. Selain itu, teknologi Cloud juga memiliki implikasi luar biasa untuk infrastruktur yang andal dan dapat diskalakan. Ketiga hal tersebut: manajemen aset, restful API, dan Cloud jika dikombinasikan akan menjadi sebuah alat pendukung berupa web service untuk pengembangan game yang mampu meningkatkan efektifitas pengembangan game dan mampu mengatasi asset yang sering diperbarui dan berjalan dalam game.

**Kata Kunci: Web Service, Manajemen Aset, Restful API, Cloud.**

**ABSTRACT**

Games become one of the media that is favored by the public whether played on a PC, smartphone, or console. Therefore, many industries and the game community are aggressively developing their products. The game industry in making game products, uses the code provided by the programmer, but most of what is seen and some players are not translated in the code, supplemented by the assets obtained. And not infrequently in one game in production can have large assets to make one game product. By taking the number of assets and people involved in the game, managing the game assets manually will take a large portion of the project, both time and money. To reduce this, it is necessary to manage assets and implement a system to store and manage all game assets. Web service was chosen as the right solution as an asset management service provider facility. Web services will provide an asset upload feature and then the architecture of the game will manage assets and produce data that can be applied to any game engine. The REST API is the most commonly used way to resolve data replacement between server and client. In addition, Cloud technology also has tremendous implications for reliable and scalable infrastructure. These three things: 2D game asset management, REST API, and the Cloud when combined will become a supporter of web services for game development that can improve the effectiveness of game development and be able to cope with assets that are often used and improved in games.

Keywords: *Web Service, 2D Game Asset Management, Restful API, Cloud.*

# KATA PENGANTAR

Dengan mengucap puji syukur kepada Allah, atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan proyek akhir ini dengan berjudul :

“Sistem Manajemen Aset Game 2D Untuk Pengembangan Game Berbasis Cloud”

Proyek Akhir ini merupakan kewajiban bagi setiap mahasiswa Politeknik Elektronika Negeri Surabaya dengan tujuan untuk memenuhi persyaratan untuk memperoleh kelulusan pada tingkat Diploma IV Program Studi Teknologi Game di Politeknik Elektronika Negeri Suarabaya.

Dengan selesainya buku laporan proyek akhir ini, penulis berharap semoga buku ini dapat membawa manfaat pembaca umumnya dan juga bagi penulis pada khususnya serta semua pihak yang berkepentingan. Penulis juga berharap agar proyek akhir ini dapat dikembangkan lebih lanjut sehingga dapat benar-benar digunakan sebaik-baiknya untuk mendukung perkembangan ilmu pengetahuan. Penulis menyadari bahwa penulis adalah manusia biasa yang tidak luput dari kesalahan dan kekurangan. Untuk itu, kritikan dan saran yang bersifat membangun kami harapkan untuk perbaikan selanjutnya.

Surabaya, 10 Januari 2020

Penulis

# UCAPAN TERIMA KASIH

Puji syukur yang tak terhingga saya sampaikan kepada Allah SWT Yang Maha Berkuasa Atas Segalanya, karena hanya dengan ridho, hidayah dan anugerah-Nya saya dapat menyelesaikan proyek akhir ini. Ucapan terima kasih saya sampaikan juga ke berbagai pihak yang turut membantu memperlancar penyelesaian proyek akhir ini. Dengan segala kerendahan hati, keikhlasan dan ketulusan, saya ingin menyampaikan rasa terima kasih dan penghargaan yang sebesar-besarnya kepada :

1. Allah SWT atas limpahan rahmat-Nya sehingga penulis dapat menyelesaikan Proyek Akhir ini dengan baik.
2. Rasulullah Muhammad SAW sebagai penyampai risalah dan sumber inspirasi bagi penulis untuk selalu memberikan yang terbaik.
3. Papa dan Ibu, serta keluarga yang telah memberikan semangat, kasih sayang, perhatian serta dukungan dan doa untuk penulis. Semoga Allah membalas beribu kebaikan dan mengasihi mereka sebagaimana mereka mengasihi penulis di waktu kecil.
4. Bapak Mohamad Safrodin, B.Sc., M.T. dan Bapak Fahim Nur Cahya Bagar, S.Kom., M.T. selaku Dosen Pembimbing yang telah memberikan bimbingan kepada penulis dalam menyelesaikan Proyek Akhir ini.
5. Bapak Rizky Yuniar Hakkun, S.T., M.T selaku Kepala Program Studi Teknologi Game.
6. Ibu Halimatus Sa’dyah, S.Kom., M.Kom. selaku Koordinator Proyek Akhir Teknologi Game.
7. Semua mentor yang telah membantu selama Proyek Akhir.
8. Teman-teman mahasiswa Teknologi Game angkatan 2016 Politeknik Elektronika Negeri Surabaya.
9. Dan semua pihak yang telah membantu baik materi maupun moral selama Proyek Akhir.

Segala ucapan terima kasih dari penulis tentunya belum cukup, Semoga Allah SWT membalas segala kebaikan-kebaikan yang diberikan tersebut.

# DAFTAR ISI

[ABSTRAK iv](#_Toc46378259)

[KATA PENGANTAR vi](#_Toc46378260)

[UCAPAN TERIMA KASIH vii](#_Toc46378261)

[DAFTAR ISI viii](#_Toc46378262)

[DAFTAR GAMBAR xi](#_Toc46378263)

[DAFTAR TABEL xiii](#_Toc46378264)

[BAB 1 PENDAHULUAN 1](#_Toc46378265)

[1.1 LATAR BELAKANG 1](#_Toc46378266)

[1.2 TUJUAN PROYEK AKHIR 2](#_Toc46378267)

[1.3 PERUMUSAN MASALAH 2](#_Toc46378268)

[1.4 BATASAN MASALAH 3](#_Toc46378269)

[1.5 METODOLOGI 3](#_Toc46378270)

[**1.5.1 ANALISA KEBUTUHAN 3**](#_Toc46378271)

[**1.5.2 DESAIN SISTEM 4**](#_Toc46378272)

[**1.5.3 IMPLEMENTASI SISTEM 4**](#_Toc46378273)

[**1.5.4 ANALISA DAN UJI COBA APLIKASI 4**](#_Toc46378274)

[**1.5.5 PENYUSUNAN LAPORAN 4**](#_Toc46378275)

[1.6 SISTEMATIKA PEMBAHASAN 5](#_Toc46378276)

[BAB II TINJAUAN PUSTAKA 6](#_Toc46378277)

[2.1 PENELITIAN TERKAIT 6](#_Toc46378278)

[**2.1.1 MANAJEMEN ASET DAN DATA PADA GAME 6**](#_Toc46378279)

[**2.1.2 PENGGUNAAN REST API PADA PENGEMBANGAN GAME 7**](#_Toc46378280)

[**2.1.3 APPCRAFTED: APLIKASI WEB SERVICE BERBASIS CLOUD UNTUK MANAJEMEN KONTEN GAME 7**](#_Toc46378281)

[2.2 TEORI PENUNJANG 7](#_Toc46378282)

[**2.2.1 WEB SERVICE 7**](#_Toc46378283)

[**2.2.2 REST API 8**](#_Toc46378284)

[**2.2.3 UNITY WEBREQUEST 8**](#_Toc46378285)

[BAB III PERANCANGAN SISTEM 9](#_Toc46378286)

[3.1 DESKRIPSI UMUM SISTEM 9](#_Toc46378287)

[3.2 PERANCANGAN ALUR APLIKASI 10](#_Toc46378288)

[**3.2.1 DESAIN DATABASE 12**](#_Toc46378289)

[**3.2.2 USERS 12**](#_Toc46378290)

[**3.2.3 PROJECTS 13**](#_Toc46378291)

[**3.2.4 PROJECT MEMBERS 13**](#_Toc46378292)

[**3.2.5 ASSETS 14**](#_Toc46378293)

[**3.2.6 TAGS 14**](#_Toc46378294)

[**3.2.7 ASSET TAGS 14**](#_Toc46378295)

[3.3 DESAIN FRONT-END 15](#_Toc46378296)

[**3.3.1 MOCKUP HALAMAN LOGIN DAN SIGN UP 15**](#_Toc46378297)

[**3.3.2 MOCKUP HALAMAN PROJECT 16**](#_Toc46378298)

[**3.3.3 MOCKUP HALAMAN ASSETS 17**](#_Toc46378299)

[**3.3.4 MOCKUP HALAMAN ASSET TERPILIH 18**](#_Toc46378300)

[**3.3.5 MOCKUP HALAMAN DOKUMENTASI MOCKUP HALAMAN ASSETS 19**](#_Toc46378301)

[3.4 PERSIAPAN PERANGKAT DAN SISTEM 19](#_Toc46378302)

[**3.4.1 INSTALASI KEBUTUHAN SERVER 20**](#_Toc46378303)

[BAB IV PENGUJIAN DAN ANALISA 23](#_Toc46378304)

[4.1 PERENCANAAN PENGUJIAN PERBANDINGAN DENGAN SISTEM MANAJEMEN ASSET LAIN 23](#_Toc46378305)

[4.2 PENGUJIAN FUNGSIONAL APLIKASI 24](#_Toc46378306)

[**4.2.1 PENGUJIAN REQUEST API MENGGUNAKAN TOOL POSTMAN 27**](#_Toc46378307)

[**4.2.2 PENGUJIAN FUNGSIONALITAS PADA DEVELOPER 30**](#_Toc46378308)

[BAB V PENUTUP 45](#_Toc46378321)

[5.1 KESIMPULAN 45](#_Toc46378322)

[5.2 SARAN 45](#_Toc46378323)

[DAFTAR PUSTAKA 47](#_Toc46378324)

# DAFTAR GAMBAR

[Gambar 3. 1 Use Case Diagram 10](#_Toc46376151)

[Gambar 3. 2 Desain Database 12](#_Toc46376152)

[Gambar 3. 3 Stuktur Tabel Users 12](#_Toc46376153)

[Gambar 3. 4 Stuktur Tabel Projects 13](#_Toc46376154)

[Gambar 3. 5 Struktur Tabel Project Members 13](#_Toc46376155)

[Gambar 3. 6 Strukter Tabel Assets 14](#_Toc46376156)

[Gambar 3. 7 Struktur Tabel Tags 14](#_Toc46376157)

[Gambar 3. 8 Struktur Tabel Asset Tags 14](#_Toc46376158)

[Gambar 3. 9 Mockup Halaman Login 16](#_Toc46376159)

[Gambar 3. 10 Mockup Halaman Dashboard Project 17](#_Toc46376160)

[Gambar 3. 11 Mockup Halaman Assets 18](#_Toc46376161)

[Gambar 3. 12 Mockup Halaman Asset Terpilih 18](#_Toc46376162)

[Gambar 3. 13 Mockup Halaman Dokumentasi 19](#_Toc46376163)

[Gambar 3. 14 Composer Install 20](#_Toc46376164)

[Gambar 3. 15 NPM Install Bootstrap 22](#_Toc46376165)

[Gambar 4. 1 Run Xampp Apache 27](#_Toc46376166)

[Gambar 4. 2 Halaman Login 28](#_Toc46376167)

[Gambar 4. 3 Halaman Asset Tags 28](#_Toc46376168)

[Gambar 4. 4 Pengujian API Get dengan Postman 29](#_Toc46376169)

[Gambar 4. 5 Testing Login Aplikasi 30](#_Toc46376170)

[Gambar 4. 6 Chart Hasil Testing Login Aplikasi 30](#_Toc46376171)

[Gambar 4. 7 Testing Halaman Selamat Datang 31](#_Toc46376172)

[Gambar 4. 8 Chart Hasil Testing Welcome Page 31](#_Toc46376173)

[Gambar 4. 9 Testing Halaman Add Projects 32](#_Toc46376174)

[Gambar 4. 10 Chart Hasil Testing Add Projects 32](#_Toc46376175)

[Gambar 4. 11 Testing Halaman Projects 33](#_Toc46376176)

[Gambar 4. 12 Chart Hasil Testing Halaman Projects 33](#_Toc46376177)

[Gambar 4. 13 Testing Halaman Upload Assets 34](#_Toc46376178)

[Gambar 4. 14 Chart Hasil Testing Upload Assets 34](#_Toc46376179)

[Gambar 4. 15 Testing Halaman Assets 35](#_Toc46376180)

[Gambar 4. 16 Chart Testing Halaman Assets 35](#_Toc46376181)

[Gambar 4. 17 Testing Halaman Edit Asset 36](#_Toc46376182)

[Gambar 4. 18 Chart Hasil Testing Edit Asset 37](#_Toc46376183)

[Gambar 4. 19 Testing Halaman Menambahkan Anggota Project 37](#_Toc46376184)

[Gambar 4. 20 Chart Hasil Testing Menambahkan Anggota Project 38](#_Toc46376185)

[Gambar 4. 21 Testing Menambahkan Tag 38](#_Toc46376186)

[Gambar 4. 22 Chart Hasil Testing Menambahkan Tag 39](#_Toc46376187)

[Gambar 4. 23 Testing Menambahkan Tag pada Asset 40](#_Toc46376188)

[Gambar 4. 24 Chart Hasil Testing Menambahkan Tag Pada Asset 40](#_Toc46376189)

[Gambar 4. 25 Testing Halaman Asset pada Tag Tertentu 41](#_Toc46376190)

[Gambar 4. 26 Chart Hasil Testing Halaman Asset pada Tag Tertentu 41](#_Toc46376191)

[Gambar 4. 27 Testing Halaman URL Image 42](#_Toc46376192)

[Gambar 4. 28 Testing Halaman API Get Asset 42](#_Toc46376193)

[Gambar 4. 29 Chart Hasil Testing URL Asset 43](#_Toc46376194)

[Gambar 4. 30 Testing Halaman Dokumentasi 43](#_Toc46376195)

[Gambar 4. 31 Chart Hasil Testing Halaman Dokumentasi 44](#_Toc46376196)

# DAFTAR TABEL

[Tabel 4. 1 Hasil Perbandingan Tools yang Sudah Ada 23](#_Toc46376197)

[Tabel 4. 2 Hardware Pengujian 24](#_Toc46376198)

[Tabel 4. 3 Spesifikasi Server 25](#_Toc46376199)

[Tabel 4. 4 Tabel Pengujian Fungsional 25](#_Toc46376200)

# BAB 1 PENDAHULUAN

## Latar Belakang

Pada era digital saat ini, game menjadi salah satu media yang banyak digemari oleh masyarakat baik dimainkan di perangkat pc, smartphone, maupun konsol. Industri dan komunitas game telah berkembang sangat baik. Terbukti pertumbuhan industri game terus melejit setiap tahun. Teranyar data survei Newzoo pada 2014, Industri game di Indonesia bernilai Rp 2,3 triliun. Setiap tahun sekitar 100 game baru dilahirkan oleh para pengembang game. Tentu ini merupakan kabar menggiurkan bagi para pelaku industri khususnya game.[1]

Untuk memudahkan para pengembang game dalam menggarap game nya menggunakan game engine. Namun tidak semua fitur untuk memproduksi game tersedia dalam satu game engine karena setiap game engine memiliki kompleksitasnya masing-masing. Maka dari itu pengembang game memerlukan alat/tools tambahan guna mendukung pengembangan game yang dikerjakannya.

Selain itu, game dibangun menggunakan kode yang disediakan oleh programmer, tetapi sebagian besar apa yang dilihat dan dialami pemain tidak didefinisikan dalam kode, melainkan aset yang diproduksi, seperti gambar, model 3D, dan suara, video, dan lain-lain. Dan tidak jarang dalam satu game yang di produksi dapat memiliki ratusan aset untuk membentuk satu produk game.

Namun untuk mencapai target pengembangan game terdapat beberapa kesulitan yang dihadapi oleh para pengembang gim, khususnya bagi pengembang gim independen (indie game developer) beberapa diantaranya mengalami sumber daya yang terbatas untuk menyediakan tools pendukung.[2]

Semakin berkembangnya fitur yang dapat diimplementasikan di dalam game dan banyak teknologi yang dapat membantu pembuatan game seperti teknologi REST API dan Cloud. REST API dapat dikostumisasikan dalam game engine dan digabungkan dengan pengaplikasian system tertentu untuk memudahkan manajemen dan optimasi asset pada produksi game. Dimana untuk menggunakan teknologi tersebut akan menambah waktu produksi cukup banyak. Untuk itu jika tersedia alat bantu yang lebih instan untuk menyelesaikan masalah tersebut produksi game akan menjadi lebih efektif.

Pada penelitan yang berjudul “Sistem Manajemen Aset untuk Pengembangan Game Berbasis Cloud” ini akan merujuk pada manajemen asset game yang dikelola pada web service yang berbasis Cloud. Aset diletakkan pada sistem Cloud dihubungkan menggunakan REST API, kemudian mengintegrasikan API ke dalam alur kerja kustom game dan melakukan beberapa teknik optimasi sehingga mendapatkan keuntungan dalam memudahkan developer untuk mengolah asset, mengoptimalkan jalannya game, serta menghemat waktu dan uang pada tahap pengembangan.

## Tujuan Proyek Akhir

Tujuan dari proyek akhir ini adalah untuk menciptakan sistem layanan antarmuka untuk mengelola asset game secara online sehingga menghasilkan alat pendukung yang lebih mudah digunakan oleh pengembang game juga menguntungkan dari segi optimasi pada game.

## Perumusan Masalah

Adapun permasalahan yang akan dibahas dalam proyek akhir ini adalah:

* + 1. Bagaimana merancang aplikasi yang dapat memudahkan para pengembang game mengelola asset game nya pada Cloud.
    2. Bagaimana memanfaatkan luaran dari aplikasi pada pengembangan game di *game engine*.

## Batasan Masalah

Agar tujuan utama tercapai dan pembahasan tidak meluas serta tidak menjadikan adanya penyimpangan permasalahan, maka penulis membuat batasan masalah yang akan dikaji sebagai berikut:

1. Sistem Manajemen Aset dikembangkan berupa aplikasi web service dan *tools plug-in* untuk game engine Unity.
2. Penelitian dan uji coba hanya dilakukan pada game 2D dan tipe data asset gambar.
3. Luaran dari aplikasi hanya dikembangkan dan di uji cobakan pada game engine Unity.

## Metodologi

Metodologi dalam pembuatan proyek akhir meliputi:

### Analisa Kebutuhan

Pada analisa kebutuhan dilakukan pembelajaran tentang apa yang dibutuhkan pada proyek akhir yang dikerjakan dan referensi mengenai teknologi yang akan digunakan pada progress proyek akhir. Beberapa kebutuhan diantaranya sebagai berikut :

1. Pembuatan backend pada sistem menggunakan Laravel PHP.
2. Bootstrap sebagai *framework* pembuatan frontend atau client.
3. Membuat plugin untuk game engine berupa *reusable script* yang di kemas menjadi UnityPackage.

### Desain Sistem

Setelah melakukan analisa kebutuhan, maka tahap selanjutnya adalah desain sistem dan aplikasi. Secara garis besar, desain sistem dan aplikasi tersebut meliputi:

1. Pembuatan alur sistem untuk pengguna.
2. Perancangan basis data untuk aplikasi.
3. Desain aplikasi web service menggunakan Adobe XD kemudian diimplementasikan menjadi frontend menggunakan Bootstrap.

### Implementasi Sistem

Hasil dari perancangan sistem akan diimplementasikan untuk dijadikan suatu aplikasi web service. Pembuatan sistem yang akan digunakan untuk menyelesaikan tugas akhir ini meliputi pembuatan frontend web, pembuatan backend berbasis Laravel PHP, serta pembuatan plugin untuk memproses data dari web pada game.

### Analisa dan Uji Coba Aplikasi

Pada tahap ini akan dilakukan ujicoba aplikasi dan analisa dari hasil. Pengujian sistem akan dilakukan setelah sistem telah selesai dibuat dengan melakukan demo sistem guna untuk memperbaiki kekurangan sistem yang sudah dibuat dan selanjutnya akan dibuat analisa.

### Penyusunan Laporan

Membuat dokumentasi dari semua tahapan proses mulai dari studi literatur hingga pengujian berupa laporan yang berisi tentang dasar teori, hasil proyek akhir serta hasil analisa.

## Sistematika Pembahasan

Sistematika pembahasan dalam penyusunan buku proyek akhir ini adalah sebagai berikut:

**BAB I: PENDAHULUAN**

Bab ini menguraikan secara singkat latar belakang, tujuan proyek akhir, perumusan masalah, batasan masalah, metodologi dan sistematika pembahasan.

**BAB II: TEORI PENUNJANG**

Bab ini menjelaskan mengenai teori penunjang yang berhubungan dengan penyelesaian proyek akhir ini, yang didapatkan dari berbagai referensi, seperti buku, paper, serta sumber – sumber lainnya yang berhubungan dengan pembuatan proyek akhir ini.

**BAB III: PERANCANGAN SISTEM**

Bab ini menjelaskan mengenai tahapan – tahapan dalam pembuatan proyek akhir ini. Mulai dari perancangan hingga tahap pembuatan aplikasi.

**BAB IV: PENGUJIAN DAN ANALISA**

Bab ini menjelaskan tentang hasil pengujian aplikasi yang telah di buat serta analisa-analisa mengenai hasil yang telah diperoleh.

**BAB V: PENUTUP**

Bab ini berisi kesimpulan dari uji coba perangkat lunak, dan saran untuk pengembangan, perbaikan serta penyempurnaan terhadap aplikasi game yang telah dibuat.

# BAB II TINJAUAN PUSTAKA

Dalam menyelesaikan rumusan permasalahan yang ada pada proyek akhir ini, dibutuhkan teori dasar yang dipergunakan sebagai acuan ilmu untuk melakukan pengukuran, pengolahan dan melakukan analisa data sehingga bisa didapatkan hasil yang sesuai dengan tujuan proyek akhir.

## Penelitian Terkait

Pada kepustakaan penelitian ini akan dibahas penelitian-penelitian yang pernah dilakukan sebelumnya. Berikut adalah beberapa penelitian yang pernah dilakukan :

### Manajemen Aset dan Data pada Game

Selain kode program yang berjalan pada game untuk menjalankan logika dan mekanika dalam game, sebagian besar apa yang dilihat dan dialami pemain tidak didefinisikan dalam kode, melainkan aset yang diproduksi. Dan tidak jarang dalam satu game yang di produksi dapat memiliki ratusan aset untuk membentuk satu produk game. Oleh karena itu, disinilah peran manajemen asset dan data game sangat krusial.

Penelitian tentang manajemen aset dan data mencakup masalah penting dalam pengembangan game. Hal tersebut akan membantu pengembang game mengurangi beban kerja para developer dalam proses yang cukup rumit, tidak hanya mengatur aset asli tetapi juga melacak perubahan dan versi. Sebagai contoh, pada “Where Is It” dari Chris Ronchi menjelaskan mengapa penting untuk mengikuti strategi penamaan (Naming Convention) yang konsisten dan dipertimbangkan dan menunjukkan cara membuatnya. Bab dasar namun bermanfaat ini meliputi area yang tidak melibatkan pemrograman, tetapi dapat membantu efisiensi dalam pengembangan game.[3]

### Penggunaan REST API pada Pengembangan Game

Penelitian ini membahas tentang analisis terhadap pengembangan game yang terhubung pada koneksi REST API. REST API merupakan pendekatan umum untuk membuat layanan web yang dapat diskalakan, seperti memberi pengguna *endpoint* untuk mengumpulkan dan membuat data menggunakan panggilan HTTP (GET, POST, PUT, DELETE). Lalu memberikan parsing data berupa JSON kepada game engine, dalam hal ini adalah Unity.[4]

### AppCrafted: Aplikasi Web Service berbasis Cloud untuk Manajemen Konten Game

Penelitian yang dilakukan Brandon Wu, 2014 “Appcrafted — a simple web service to manage content in your game” melakukan manajemen konten game yang dikemas berupa web service berbasis cloud yang bertujuan menjaga fitur dari sisi server pada pengembangan game, mengunggah asset dan mengembalikannya berupa plugin yang dapat diteruskan pada game engine, dan memperbarui asset secara dinamis yang terintegrasi oleh dashboard dan plugin Unity.[5]

## Teori Penunjang

### Web Service

Web Service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung *interaction* and interoperability antar sistem pada suatu jaringan. Web service digunakan sebagai suatu fasilitas yang menyediakan layanan (dalam bentuk informasi atau data) kepada system lain, sehingga dapat berinteraksi dengan sistem tersebut melalui layanan-layanan yang disediakan. Web service menyimpan data informasi dalam format JSON atau XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, dan bahasa pemrograman.

### Rest API

REST (REpresentational State Transfer) merupakan standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web. Umumnya menggunakan HTTP (Hypertext Transfer Protocol) sebagai protocol untuk komunikasi data. REST pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000.

Pada arsitektur REST, REST server menyediakan resources (sumber daya/data) dan REST client mengakses dan menampilkan resource tersebut untuk penggunaan selanjutnya. Setiap resource diidentifikasi oleh URIs (Universal Resource Identifiers) atau global ID. Resource tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML.[6]

### Unity WebRequest

UnityWebRequest adalah pengganti objek WWW yang sebelumnya digunakan Unity. Objek ini menyediakan sistem modular untuk menyusun permintaan HTTP dan menangani respons HTTP. Tujuan utama sistem UnityWebRequest adalah untuk mengizinkan game Unity berinteraksi dengan backend Web modern. Objek ini juga mendukung fitur permintaan tinggi seperti permintaan HTTP, streaming operasi POST / PUT dan kontrol penuh atas header dan kata kerja HTTP.[7]

# BAB III PERANCANGAN SISTEM

Dalam bab ini akan dibahas mengenai langkah-langkah perancangan sistem dalam pembuatan aplikasi tugas akhir beserta penjelasan di setiap tahapannya. Hal ini dilakukan dengan tujuan untuk mendapatkan hasil yang optimal dengan cara mempertimbangkan berbagai kebutuhan yang ada pada sistem dan berbagai permasalahan yang terjadi.

## Deskripsi Umum Sistem

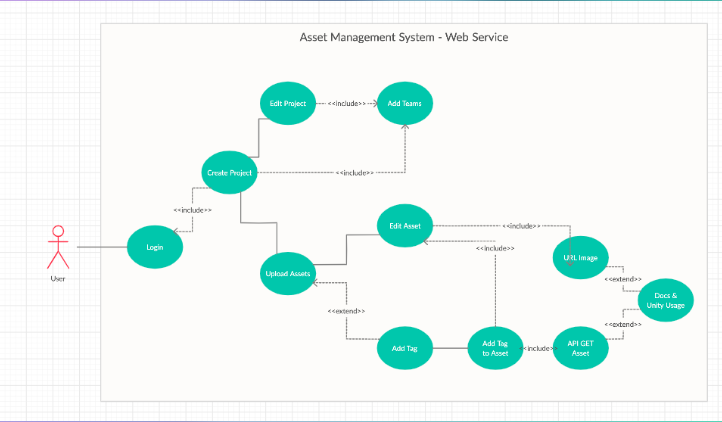
Sistem Manajemen Aset untuk Pengembangan Game berbasis Cloud ini adalah sebuah layanan berupa web service yang dapat membantu developer game menyimpan asset game nya secara online di cloud dan dapat diintegrasikan ke dalam game engine.

Pada sisi client yang merupakan tampilan muka website dikembangkan menggunakan Laravel Blade dan Framework Bootstrap 4 yang menjadi sarana pengguna dalam mengelola aset-asetnya melalui web service ini. Pada client, pengguna dapat membuat project, mengunggah dan melihat asset, membuat tag baru, dan mengumpulkan asset berasarkan koleksi tertentu yang kemudian data-data dari client akan diteruskan ke sisi server.

Pada sisi server, data-data yang sudah dimasukan oleh pengguna akan diolah dan disimpan ke database dan asset akan diteruskan ke system cloud. Data pengguna dan project diolah agar sistem ini dapat digunakan untuk kolaborasi antar pengguna. Data asset dan tag diolah supaya asset terkumpul sesuai tag yang dibutuhkan pengguna, serta luaran url asset dan endoint API GETasset yang dapat diteruskan ke game engine yang diintegrasikan menjadi sebuah plugin berupa *reusable script* UnityPackage.

## Perancangan Alur Aplikasi

Untuk memahami garis besar alur aplikasi pada proyek akhir ini, dapat dilihat pada *use case* diagram seperti pada gambar.



Gambar 3. 1 Use Case Diagram

Pada Gambar 3.1 terdapat beberapa unsur yang menunjukkan alur penggunaan aplikasinya diantaranya:

1. **Login**

Pada bagian ini merupakan implementasi sistem autentikasi yang sudah disediakan oleh Laravel yang mana table user pada database dapat otomatis terintegrasi dengan file migration Laravel sehingga dapat mempercepat dan memudahkan developer membuat fitur ini.

1. **Halaman Project**

Halaman project berisi project-project yang dimiliki oleh pengguna ketika pengguna berhasil login ke sistem web service. Pada halaman ini pengguna dapat membuat project baru dan melihat project yang sudah ada, selain itu pengguna dapat menambahkan pengguna lain untuk berkontribusi/ berkolaborasi dalam project.

1. **Halaman Asset**

Bagian ini berisi asset-aset yang ada pada project yang sebelumnya telah dibuat oleh pengguna. Pengguna dapat mengunggah asset, melihat asset dan melakukan modifikasi pada asset. Setiap asset tunggal yang diunggah akan menghasilkan url asset yang dapat digunakan pengguna untuk diteruskan pada pengembangan game selanjutnya.

1. **Mengorganisir Asset dengan Tag**

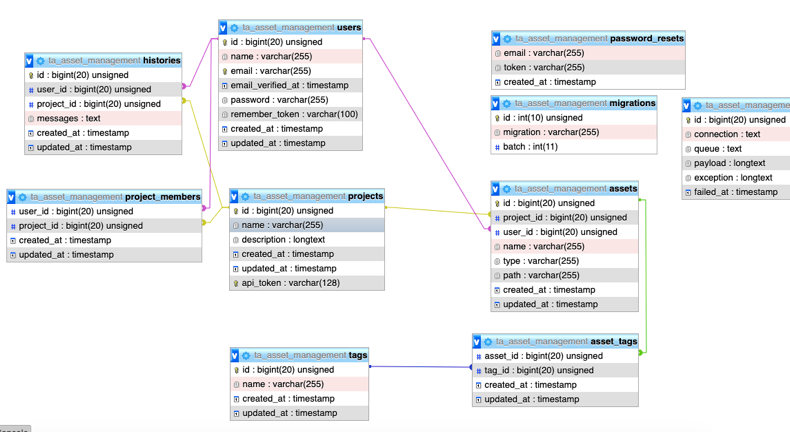
Pada bagian ini pengguna akan diarahkan untuk membuat tag yang kemungkinan dibutuhkan pengguna untuk mengorganisir asset-asset yang ada di dalam project. Pengguna dapat membuat tag kemudian menyematkan tag pada asset-asset tertentu pada saat pengguna mengunngah asset atau menambahkannya saat memodifikasi asset. Lalu pengguna dapat mengakses salah satu tag dan mendapatkan informasi dari kumpulan asset di dalam tag tersebut dan mendapatkan luaran endpoint API untuk diteruskan pada pengembangan game selanjutnya.

1. **Dokumentasi Penggunaan Pada Game Engine Unity**

Setelah fitur-fitur fungsional pada sistem web service ini berfungsi dan dapat digunakan oleh pengguna, dokumentasi sangat penting untuk meminimalkan ketidakfahaman pengguna menggunakan aplikasi ini dan memberikan langkah-langkah dasar kepada pengguna tentang penggunaan selanjutnya luaran yang didapat pada game engine, khususnya unity.

### Desain Database

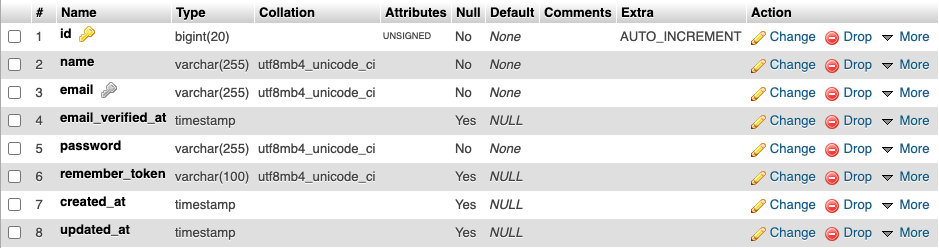
Tabel yang digunakan oleh penulis adalah users, projects, project\_members, assets, tags, asset\_tags. Juga terdapat 3 tabel tambahan yang merupakan bawaan dari Laravel, yaitu password\_resets, migration dan failed\_jobs.



Gambar 3. 2 Desain Database

Gambar 3.2 merupakan gambar rancangan desain database yang telah dibuat, terdapat beberapa table yang memiliki relasi antara satu dengan lainnya.

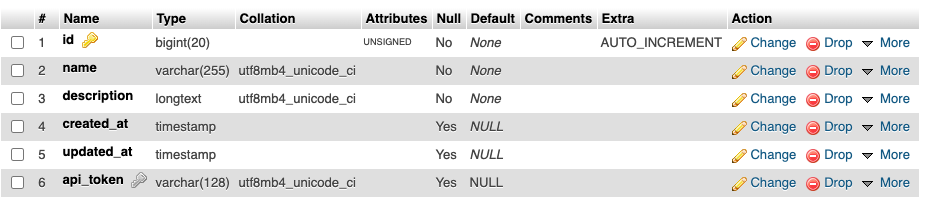
### Users



Gambar 3. 3 Stuktur Tabel Users

Gambar 3.3 merupakan struktur dari table users. Table users adalah table utama yang berguna menyimpan user yang telah login ke aplikasi.

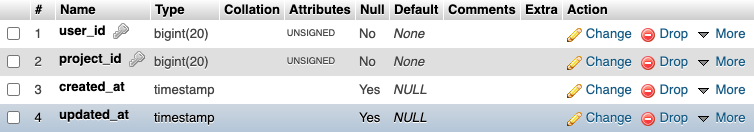
### Projects



Gambar 3. 4 Stuktur Tabel Projects

Gambar 3.4 merupakan struktur dari table projects. Tabel projects digunakan untuk menyimpan project-project yang dimiliki oleh user.

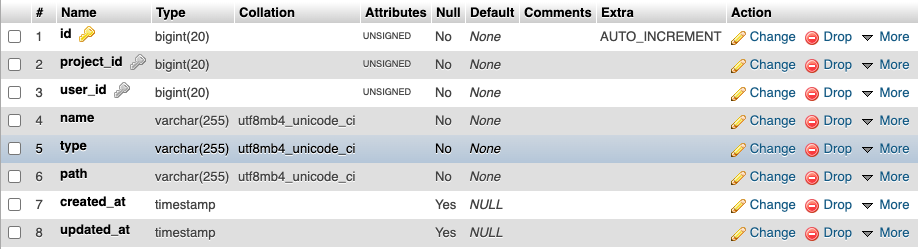
### Project Members



Gambar 3. 5 Struktur Tabel Project Members

Gambar 3.5 merupakan struktur dari table project members. Tabel project member mempunyai relasi antara table user dengan table projects. Tabel ini digunakan untuk menampung user-user yang merupakan anggota dari suatu project yang dibuat.

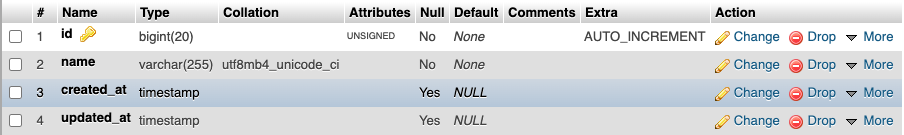
### Assets



Gambar 3. 6 Strukter Tabel Assets

Gambar 3.6 merupakan struktur dari table assets. Tabel assets adalah table yang digunakan untuk menyimpan asset-aset game pada suatu project tertentu yang telah dibuat oleh pengguna.

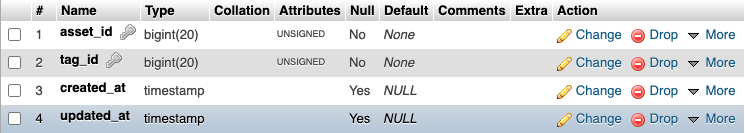
### Tags



Gambar 3. 7 Struktur Tabel Tags

Gambar 3.7 merupakan struktur dari table tags. Table tags berguna menampung tag yang sudah ditambahkan oleh pengguna dan akan digunakan pada asset-asset tertentu.

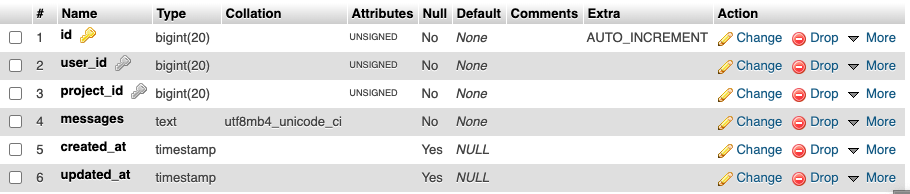
### Asset Tags



Gambar 3. 8 Struktur Tabel Asset Tags

Gambar 3.8 merupakan struktur dari table asset tags. Table asset tags adalah table yang menampung menampung asset-asset yang merupakan anggota dari tag yang sudah dibuat oleh pengguna. sebelumnya.

### Histories



Gambar 3. 9 Struktur Tabel Histories

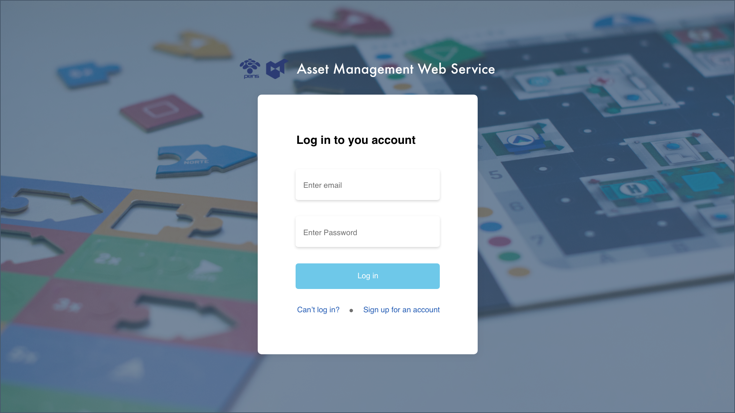
Gambar 3.9 merupakan struktur dari table histories. Table histories adalah table yang menampung aktifitas dari tiap project oleh pengguna-pengguna yang mempunyai project tersebut.

## Desain Front-End

Pada tampilan muka system web service diawali dengan halaman login. Jika login berhasil pengguna diarahkan ke halaman project dan seterusnya. Berikut penjelasan mengenai menu yang ada pada aplikasi sistem manajemen asset ini:

### Mockup Halaman Login dan Sign Up

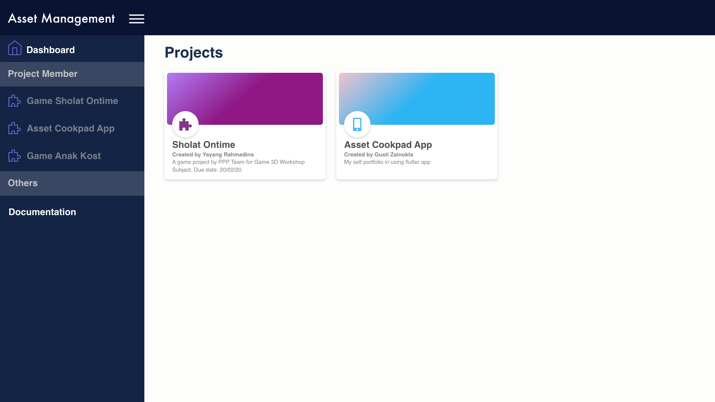
Tampilan login adalah tampilan awal pengguna berinteraksi dengan system web service ini. Sebelum pengguna dapat mengakses fitur aplikasi ini lebih lanjut pengguna wajib melakukan sign up terlebih dahulu dan melakukan login untuk dapat masuk ke aplikasi.



Gambar 3. 10 Mockup Halaman Login

### Mockup Halaman Project

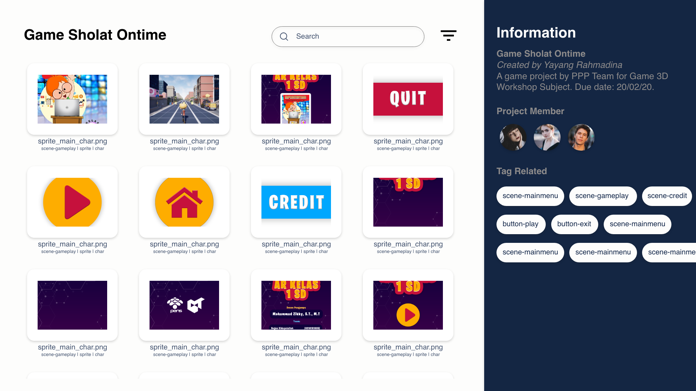
Pada halaman ini pengguna dapat melihat project-project yang sudah dibuat oleh pengguna. Pada halaman ini juga terdapat tombol add project yang berguna untuk menambah project baru. Project yang ditampilkan merupakan project yang telah dibuat oleh pengguna atau project yang telah ditambahkan oleh pengguna lain. Pengguna dapat memilih salah satu project untuk diarahkan ke halaman asset. Untuk melakukan modifikasi project, pengguna harus mengakses salah satu project.



Gambar 3. 11 Mockup Halaman Dashboard Project

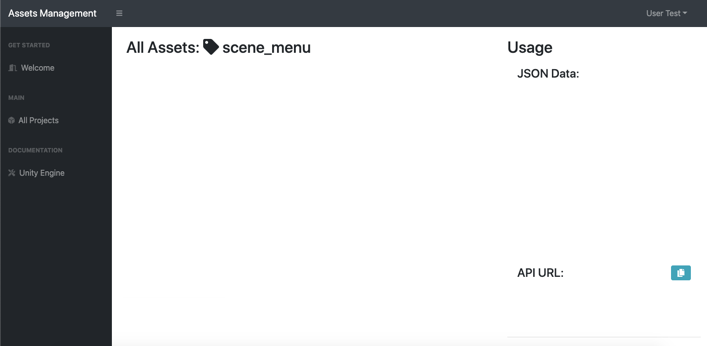
### Mockup Halaman Assets

Gambar diatas merupakan tampilan halaman assets pada aplikasi ini. Halaman ini terbagi menjadi 2 kolom utama yaitu kolom sebelah kanan yang terdiri dari judul project, tombol add assets dan kumpulan card asset. Juga kolom informasi yang menampilkan informasi terkait dari project dan asset. Serta komponen pendukung seperti tampilan pop-up untuk menambah asset, merubah informasi project, menambah anggota project, dan menambah tag baru. Juga terdapat informasi tag yang terhubung pada asset-asset yang ada di dalam project.



Gambar 3. 12 Mockup Halaman Assets

### Mockup Halaman Asset Terpilih

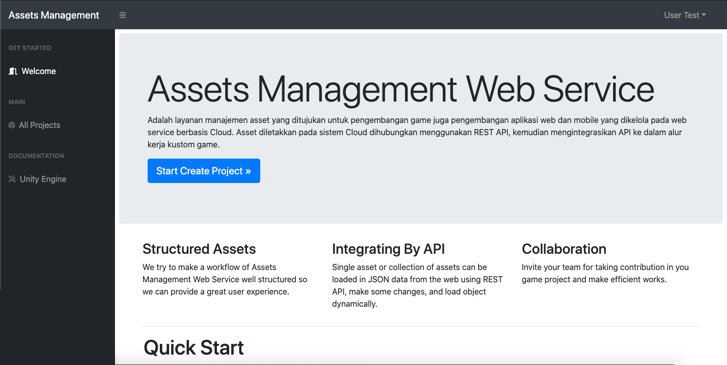


Gambar 3. 13 Mockup Halaman Asset Terpilih

Pada halaman sebelumnnya, setelah pengguna dapat melakukan 2 aksi untuk meninjau asset terpilih, asset tunggal maupun asset-asset yang terkumpul pada tag yang sama. Ketika pengguna ingin meninjau asset tunggal pengguna dapat memilih salah satu card asset pengguna dan akan diarahkan pada tampilan asset terpilih. Atau jika pengguna ingin meninjau asset-asset yang terkumpul pada suatu tag tertentu, pengguna dapat menekan salah satu tombol tag lalu diarahkan ke tampilan asset terpilih. Secara umum rancangan mockup kedua aksi tersebut sama.

Halaman asset terpilih ini terdiri 2 kolom utama yaitu kolom sebelah kanan yang terdiri dari judul asset dan *preview* dari asset. Juga kolom informasi yang menampilkan informasi terkait dari asset. Juga terdapat komponen pendukung seperti button edit asset yang menampilkan pop-up untuk merubah informasi asset dan terdapat button hapus asset.

### Mockup Halaman Dokumentasi Mockup Halaman Assets



Gambar 3. 14 Mockup Halaman Dokumentasi

Pada halaman dokumentasi ini berisi halaman statis yang memuat informasi tentang aplikasi dan petunjuk cara menggunakan aplikasi ini juga halaman cara menggunakan luaran aplikasi ini pada game engine.

## Persiapan Perangkat dan Sistem

Dalam pengerjaan proyek akhir ini diperlukan persiapan sistem terlebih dahulu. Perangkat yang digunakan adalah perangkat lunak yang berupa program sistem antara lain sebagai berikut :

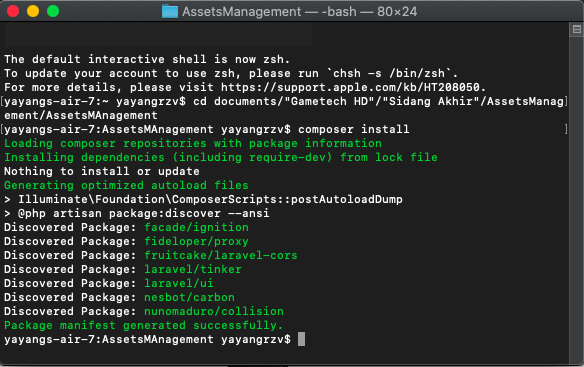
### Instalasi Kebutuhan Server

Pada server terdapat beberapa instalasi perangkat lunak yang harus terpasang dan terkonfigurasi terlebih dahulu untuk memastikan proses pengiriman data berjalan dengan lancar.

1. **Konfigurasi Laravel**

Instalasi Laravel pada komputer atau laptop dilakukan menggunakan composer. Namun, dalam instalasinya, komputer pada server juga harus terinstall XAMPP/PHP. Masuk ke dalam folder project Laravel kemudian jalankan command console yang mengarah ke project tadi, kemudian jalankan command:

composer install



Gambar 3. 15 Composer Install

Gambar diatas merupakan tampilan command console menjalankan perintah composer untuk menginstal dependency yang harus ada pada project Laravel. Dependency tersebut kemudian akan berada pada folder vendor.

Setelah itu buka file bernama .env kemudian setting database sesuai dengan konfigurasi MySQL yang terpasang pada XAMPP.

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=ta\_asset\_management

DB\_USERNAME=root

DB\_PASSWORD=

Jika konfigurasi pada file .env telah selesai maka selanjutnya buka kembali command console, kemudian jalankan *command*:

php artisan migrate

*Command* diatas bertujuan untuk membuat tabel dan kolom pada *database* sesuai dengan skema yang telah diprogram di Laravel secara otomatis.

Tabel yang ter-*generate* masih kosong, jika memerlukan data *dummy* dapat menjalankan command untuk melakukan *seeding* ke *database*:

php artisan db:seed

*Database* *seeding* adalah teknik memasukkan data ke dalam *database* secara otomatis dengan dataset yang telah diprogram pada Laravel. Untuk mengkonfigurasi *seeder* dapat diatur di dalam folder database/seeder.php

1. **Intalasi Kebutuhan Client**

Untuk membuat interface client, penulis menggunakan system templating yang ada pada Laravel yaitu Laravel Blade. Blade memiliki kode-kode yang lebih mudah untuk menghasilkan HTML, menulis peritah if else, looping, array, dan lain-lain. Untuk mempercantik dan mendukung komponen-komponen UI pada system web service ini penulis juga mengintall framework Bootstrap 4 . Proses instalasi Bootstrap dapat dilakukan beberapa cara salah satunya dengan menggunakan npm.

npm install bootstrap



Gambar 3. 16 NPM Install Bootstrap

Jalankan perintah diatas untuk melakukan instalasi bootstrap pada project Laravel.

# BAB IV PENGUJIAN DAN ANALISA

Pada bab ini akan dijelaskan mengenai pengujian terhadap beberapa bagian front-end aplikasi dan endpoint backend. Akan dilakukan analisa dari pengujian terhadap beberapa kondisi yang akan menghasilkan suatu kesimpulan. Berikut adalah penjelasan dari pengujian yang dilakukan.

## Perencanaan Pengujian Perbandingan dengan Sistem Manajemen Asset Lain

Berikut adalah tabel perbandingan antara tools yang sudah ada dengan tool yang menjadi proyek akhir ini:

Tabel 4. 1 Hasil Perbandingan Tools yang Sudah Ada

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nama Aplikasi | Fitur Pada Sistem Manajemen Asset | | | | |
|  | Organisasi Asset | API | Kolaborasi | Game Engine Usage/ Plugin | Akses |
| Google Drive |  |  |  |  | Free |
| Favro |  |  |  |  | In App Purchase |
| Cloudinary |  |  |  |  | Free |
| Assets Management |  |  |  |  | Free |

Keterangan: Hijau = Ada, Merah = Tidak ada.

Beberapa fitur yang dibandingkan dengan aplikasi lain adalah:

1. Organisasi Asset, kemampuan aplikasi tersebut untuk mengorganisir asset-asset.
2. API, merupakan fitur dalam aplikasi yang menandakan apakah aplikasi tersebut memliki API yang dapat diakses atau tidak.
3. Kolaborasi, menandakan apakah aplikasi tersebut dapat digunakan untuk bekerja secara kolaborasi atau tidak.
4. Game Engine Usage/ Plugin, fitur pada aplikasi tersebut yang mendukung penggunaan web service dengan pengembangan game development secara instan/ otomatis.
5. Akses yang disuguhkan oleh masing-masing web service, apakah berbayar atau penggunaan gratis.

Namun demikian, aplikasi Asset Management Web Service ini masih memiliki beberapa kekurangan diantaranya:

UI UX aplikasi yang perlu ditingkatkan untuk efektifitas penggunaan oleh pengguna.

Keterbatasan besar file upload, karena untuk tahap pengembangan ini masih terbatas 2.5 MB.

Penggunaan asset game masih terbatas pada tipe file image.

## Pengujian Fungsional Aplikasi

Tabel 4. 2 Hardware Pengujian

|  |  |
| --- | --- |
| Processor | Intel Core i7 |
| Memory | 8 GB |
| VGA | Intel HD Graphic 5000 |
| Sistem Operasi | macOS Catalina |

Tabel 4. 3 Spesifikasi Server

|  |  |
| --- | --- |
| Processor | 1 Core |
| VPS | Digitalocean |
| Memory | 1 GB |
| Storage | 25 GB |
| Sistem Operasi | Ubuntu 18.04.3 (LTS) x64 |

Pada tabel di atas, merupakan spesifikasi laptop yang digunakan pada pengujian awal yang bertujuan untuk memastikan apakah aplikasi sudah berjalan dengan tepat atau belum. Pengujian dilakukan dengan cara menjalankan semua fitur dan mencatat hasilnya. Adapun fiftur-fitur yang diuji dapat dilihat pada tabel 4.4:

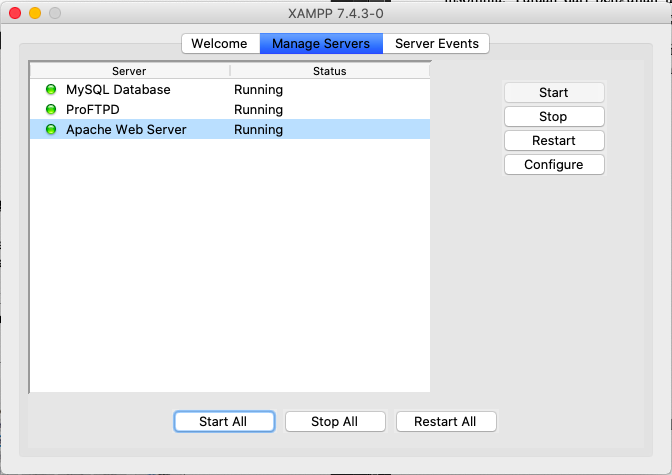
Tabel 4. 4 Tabel Pengujian Fungsional

|  |  |  |
| --- | --- | --- |
| Fitur | Keterangan | Definisi Keberhasilan |
| Login | Pendaftaran akun user | User dapat mendaftar dan dapat masuk ke aplikasi dan masuk ke dashoboard |
| Create Project | User membuat project | User berhasil membuat project dan muncul pada dashboard |
| Edit Project | User mengubah/memperbarui project | User berhasil mengubah attribut project sesuai yang di inputkan |
| Fitur | Keterangan | Definisi Keberhasilan |
| Upload Asset | User mengunggah asset | User berhasil menambahkan asset pada projectnya |
| Add Tag | User menambah tag | User berhasil menambah tag sesuai yang di inputkan |
| Edit Asset | User mengubah/memberbarui asset | User berhasil mengubah attribut asset sesuai dengan yang di inputkan |
| Add Team Member | User menambahkan user lain pada projectnya | User berhasil menambahkan user lain ke dalam projectnya |
| Add Tag to Asset | User menambahkan tag pada asset | User dapat menambahkan tag pada asset yang dipilihnya |
| Delete Asset | User menghapus project | User berhasil menghapus project |
| URL image dan API Asset | User dapat mengakses url dan endpoint API | User dapat terbantu dengan data dari url/ endpoint untuk kebutuhan pengembangan game nya di game engine |
| Dokumentasi | Informasi bagi user tentang penggunaan aplikasi dan penggunaan pada game engine | User paham tentang tujuan dan penggunaan web service serta cara menggunakan asset online pada game engine. |
| Log Aktivitas | Informasi bagi user tentang aktifitas-aktifitas yang berlangsung dalam project tersebut | User dapat terbantu oleh informasi log aktifitas ketika dalam suatu project tersebut terdapat perubahan-perubahan informasi atau asset. |

### Pengujian Request API menggunakan Tool Postman

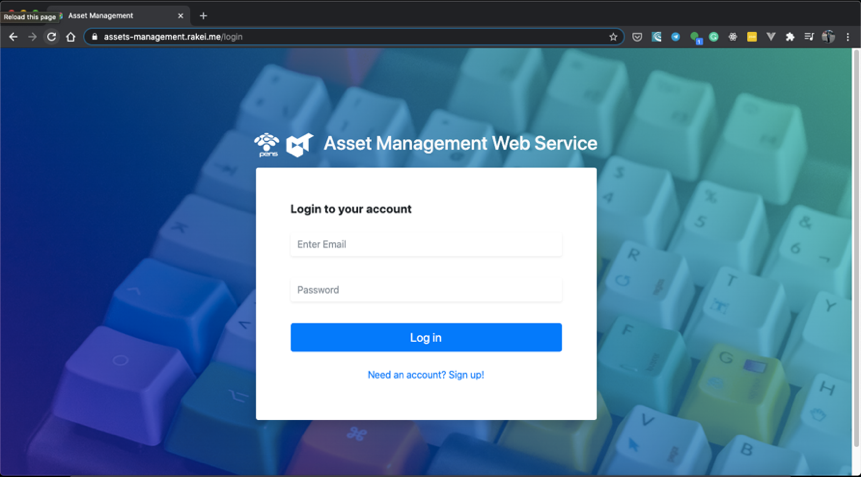
Pada pengujian ini dilakukan dengan cara menginputkan URL endpoint API pada postman. Tujuan dari pengujian ini adalah untuk memastikan system majamemen asset API berjalan dengan benar.

Sebelum memulai pengujian, pastikan XAMPP Apache dan MySQL sudah berjalan di server.



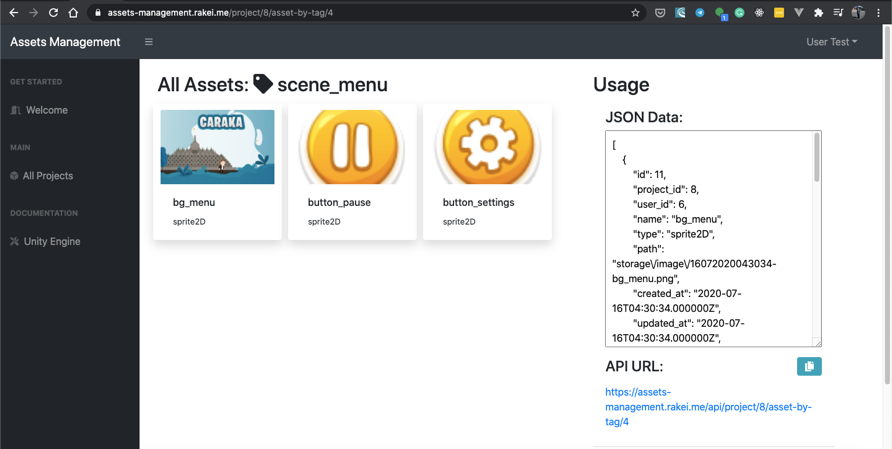
Gambar 4. 1 Run Xampp Apache

Setelah memastikan semua sudah terpasang, kemudian jalankan perintah *php artisan serve* pada command console kemudian pada local storage browser isikan dengan url <http://localhost:8000> atau <http://127.0.0.1:8000> dan lakukan login.



Gambar 4. 2 Halaman Login

Setelah berhasil login ke aplikasi, langkah selanjutnya lakukan beberapa aktifitas seperti membuat project, mengunggah asset, mengorganisir asset dengan tag, kemudian mengakses salah satu tag yang digunakan dalam asset.

****

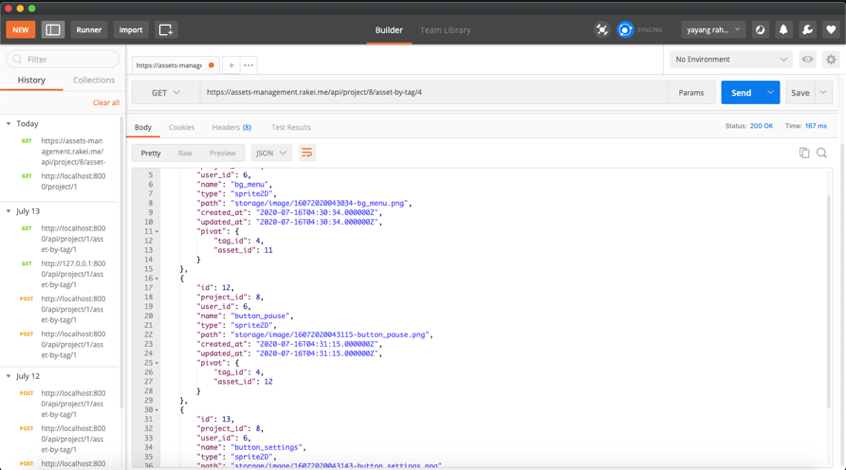
Gambar 4. 3 Halaman Asset Tags

Pengguna akan mendapatkan informasi endpoint API, salin endpoint tersebut kemudia buka aplikasi Postman. Pilih request method (GET) dan salin tautan tersebut pada kolom input type URL.

* + - * 1. **Get Asset By Tag**

API Endpoint: <http://127.0.0.1:8000/api/project/8/asset-by-tag/4>

Method : GET

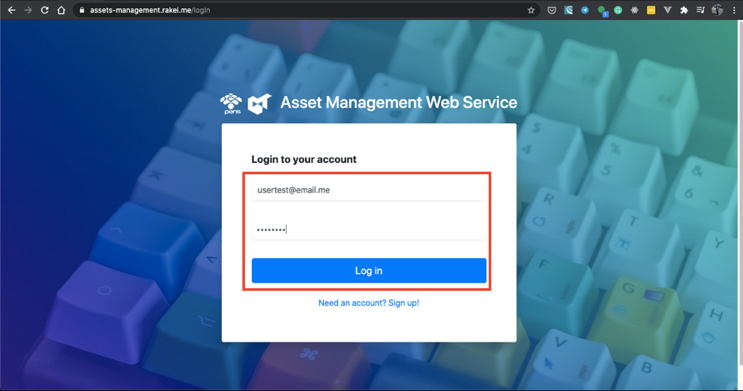


Gambar 4. 4 Pengujian API Get dengan Postman

API ini menampilkan data dari asset yang memiliki tag didalam project tertentu. Ketika request berhasil maka akan memperlihatkan hasil yang diperoleh dari endpoint tersebut dan mengembalikan code 200.

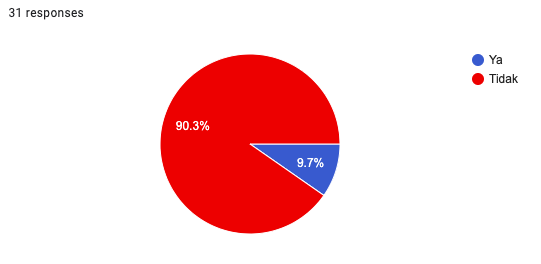
### Pengujian Fungsionalitas pada Developer

* 1. **Login dan sign up**



Gambar 4. 5 Testing Login Aplikasi

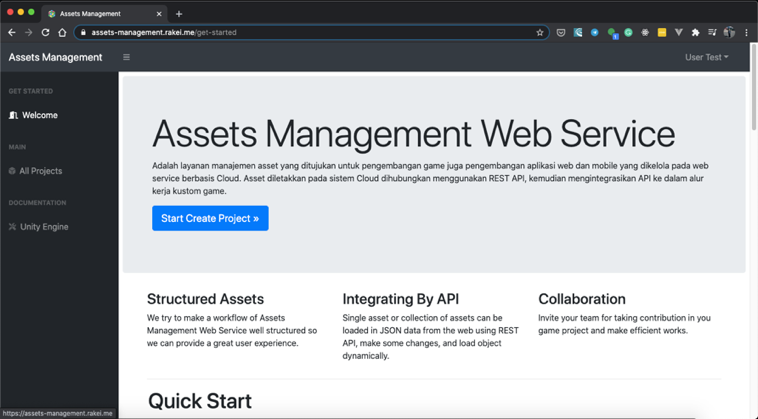
Pengguna mengisi field email dan password untuk masuk ke dalam aplikasi



Gambar 4. 6 Chart Hasil Testing Login Aplikasi

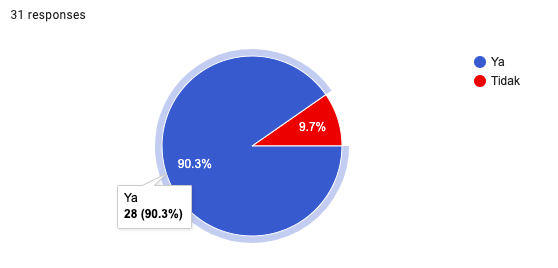
Hasil respon pengguna dari kuesioner yang dibuat, menyatakan seluruh pengguna tidak ada kendala Ketika masuk kedalam aplikasi.

* 1. Dokumentasi Welcome Page



Gambar 4. 7 Testing Halaman Selamat Datang

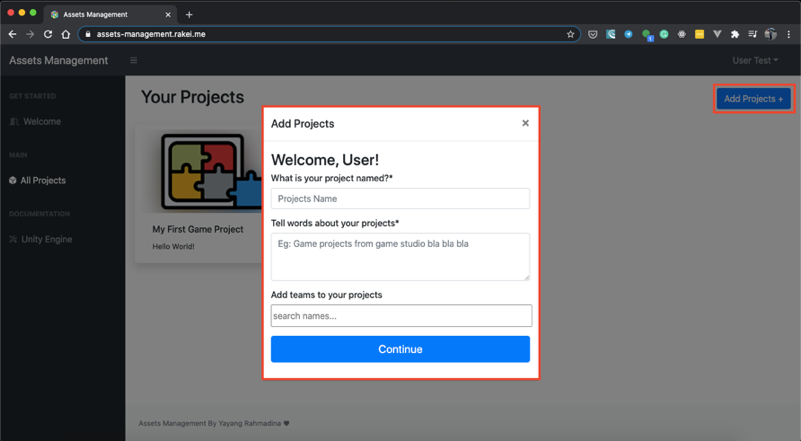
Pengguna diarahkan ke halaman selamat datang untuk mengenal tujuan dan penggunaan singkat aplikasi ini



Gambar 4. 8 Chart Hasil Testing Welcome Page

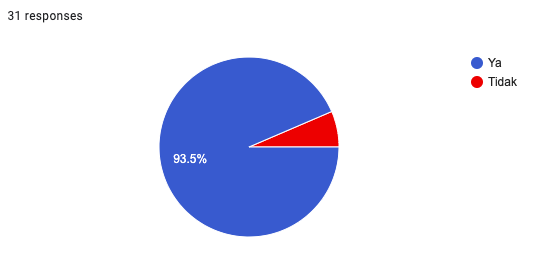
Hasil respon pengguna dari kuesioner yang dibuat, menyatakan 90.3% pengguna dapat memahami dan mengerti tujuan dan penggunaan singkat aplikasi ini dan 9.7% tidak dapat memahami isi halaman selamat datang.

* 1. Create Project



Gambar 4. 9 Testing Halaman Add Projects

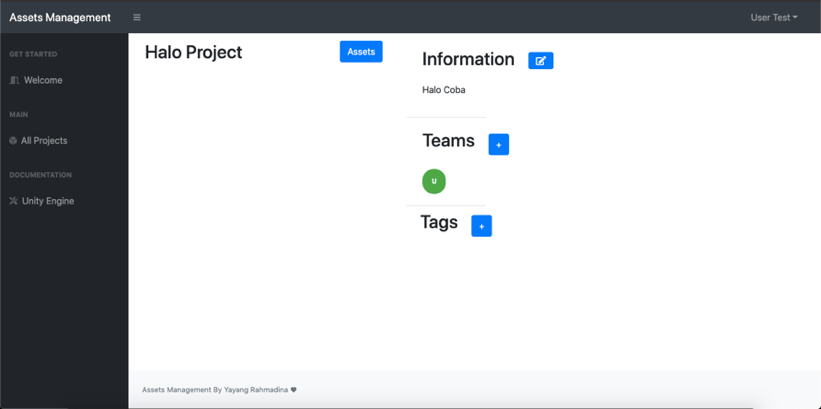
Pengguna menekan tombol ‘Add Project’ dan mengisikan atribut yang diperlukan, kemudian task tersebut menjadi sebuah card.



Gambar 4. 10 Chart Hasil Testing Add Projects

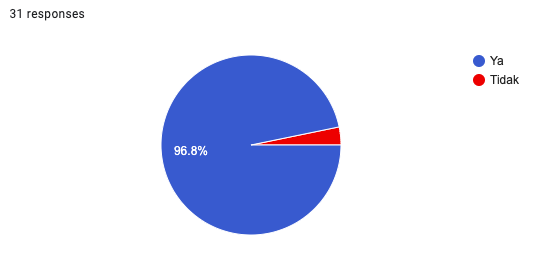
Hasil respon pengguna dari kuesioner yang dibuat, menyatakan 93.5% pengguna dapat membuat project dan project muncul menjadi card dan 6.5% pengguna tidak dapat membuat project baru.

* 1. Mengakses Halaman Project



Gambar 4. 11 Testing Halaman Projects

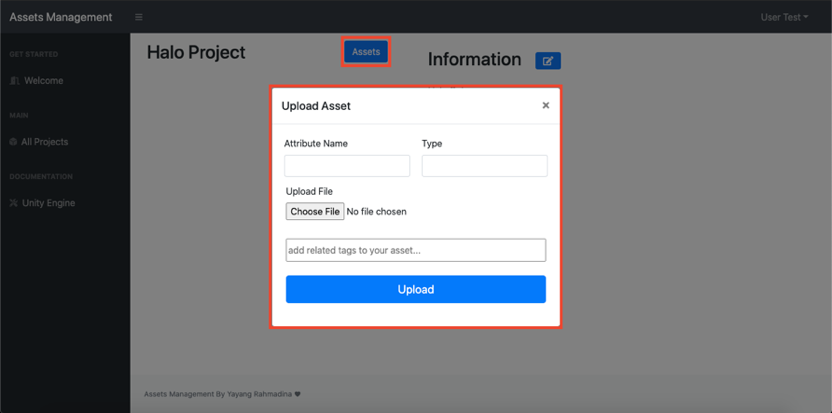
Ketika berada di halaman project pengguna menekan salah satu card project dan menuju halaman asset.



Gambar 4. 12 Chart Hasil Testing Halaman Projects

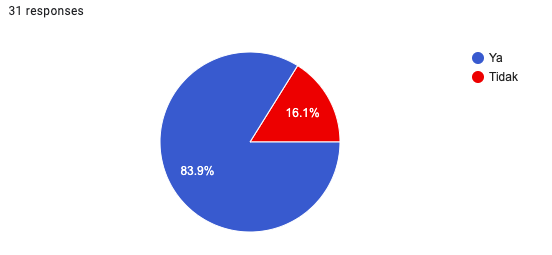
Hasil respon pengguna dari kuesioner yang dibuat, menyatakan 96.8% pengguna dapat mengakses card project menuju halaman asset dan 3.2% pengguna tidak mengakses card project.

* 1. Upload Asset



Gambar 4. 13 Testing Halaman Upload Assets

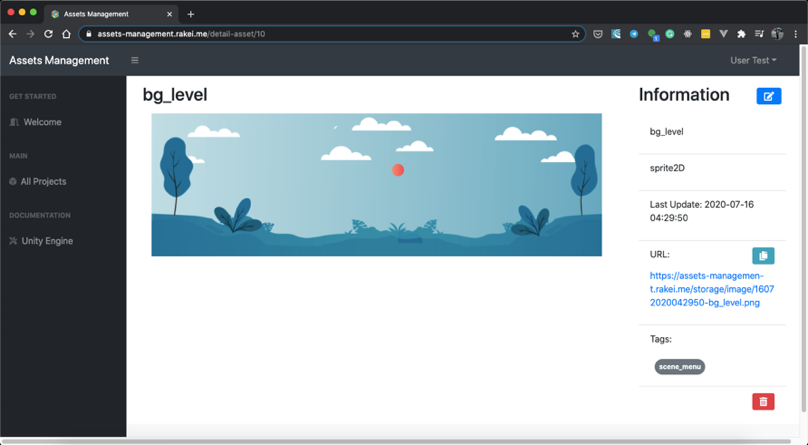
Ketika berada di halaman project yang sudah dipilih, pengguna akan menjumpai tombol Asset untuk mengunggah asset baru dan mengisi attribute yg diperlukan, kemudian asset yang sudah kita inputkan tersebut menjadi card yg tampil pada halaman ini.



Gambar 4. 14 Chart Hasil Testing Upload Assets

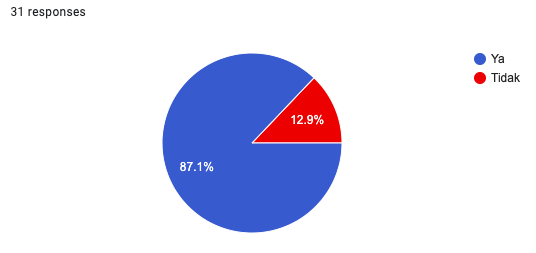
Hasil respon pengguna dari kuesioner yang dibuat, menyatakan 83.9% pengguna dapat mengunggah asset dan 16.1% pengguna tidak dapat mengunggah asset.

* 1. Mengakses Asset



Gambar 4. 15 Testing Halaman Assets

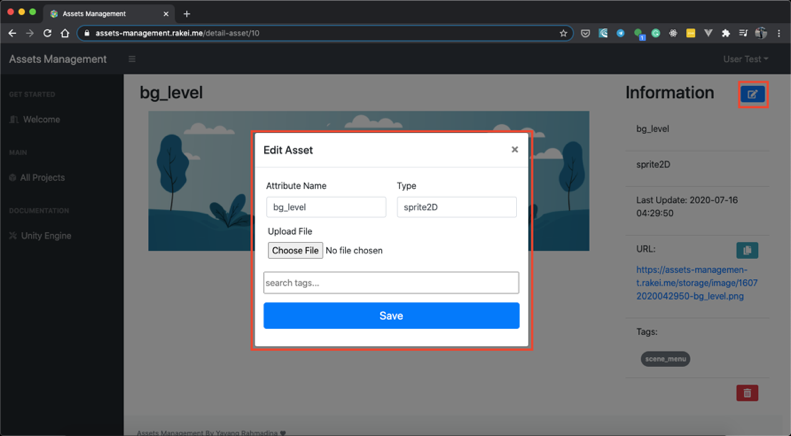
Asset-asset yang sudah di unggah akan mucul di halaman asset berupa card-card. Jika pengguna menekan salah satu card asset, maka pengguna akan diarahkan menuju halaman preview asset tersebut.



Gambar 4. 16 Chart Testing Halaman Assets

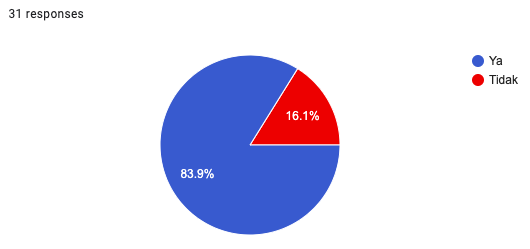
Hasil respon pengguna dari kuesioner yang dibuat, menyatakan 87.1% pengguna dapat mengakses asset dan 12.9% pengguna tidak dapat mengakses asset.

* 1. Edit Asset



Gambar 4. 17 Testing Halaman Edit Asset

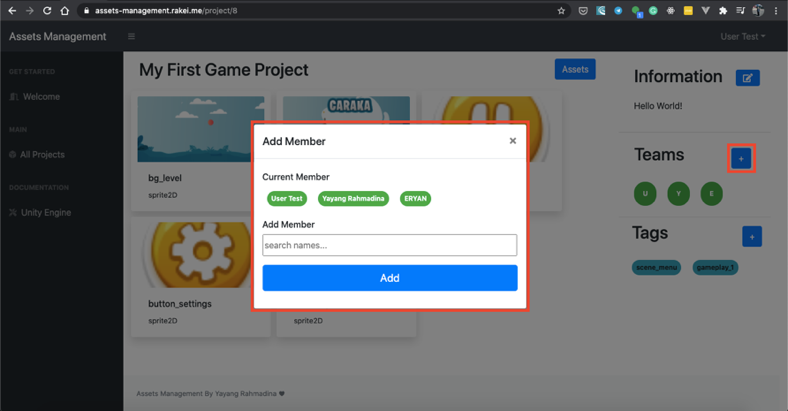
Ketika pengguna berada di halaman preview salah satu asset, maka pengguna akan melihat tombol edit asset jika ingin mengubah atau memperbarui isi attribute yang ada pada asset. Pengguna harus mengisikan atrribut-attribute yang ingin diubah.



Gambar 4. 18 Chart Hasil Testing Edit Asset

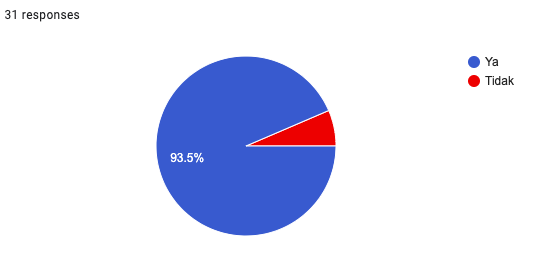
Hasil respon pengguna dari kuesioner yang dibuat, menyatakan 83.9% pengguna dapat mengubah/ memperbarui asset dan 16.1% pengguna tidak dapat mengubah/ memperbarui asset.

* 1. Menambahkan Anggota Project



Gambar 4. 19 Testing Halaman Menambahkan Anggota Project

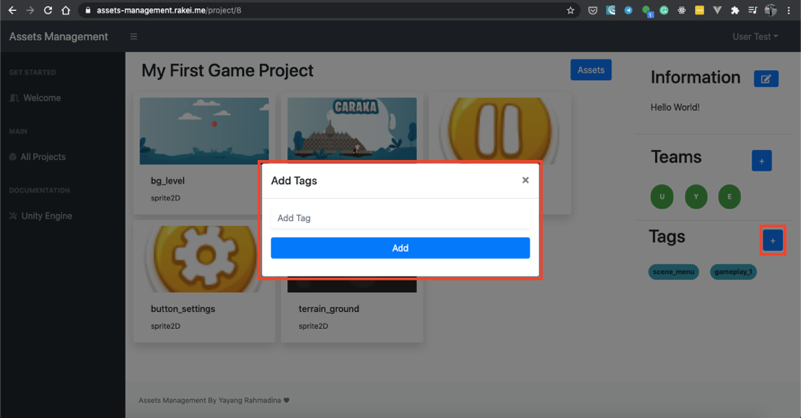
Pada saat pengguna membuat project baru, pengguna akan mendapatkan kolom “add teams”, begitu juga saat anda berada di halaman project pengguna, pada sidebar sebelah kanan ada fitur menambahkan anggota project.



Gambar 4. 20 Chart Hasil Testing Menambahkan Anggota Project

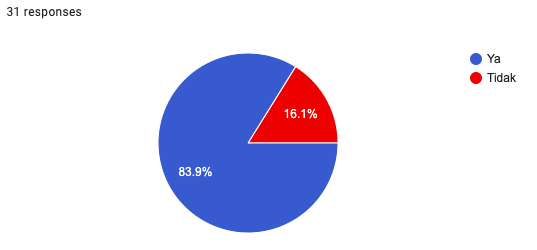
Hasil respon pengguna dari kuesioner yang dibuat, menyatakan 93.5% pengguna dapat menambah anggota project dan 6.5% pengguna tidak dapat menambah anggota project.

* 1. Menambahkan Tag



Gambar 4. 21 Testing Menambahkan Tag

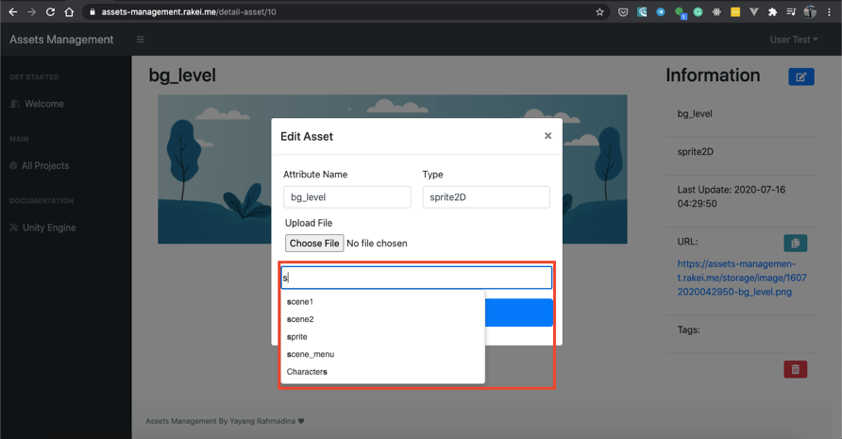
Pada saat pengguna membuat project baru, pengguna akan mendapatkan kolom “Add Tags”, begitu juga saat anda berada di halaman project pengguna, pada sidebar sebelah kanan ada fitur menambahkan tag yang akan digunakan pada asset.



Gambar 4. 22 Chart Hasil Testing Menambahkan Tag

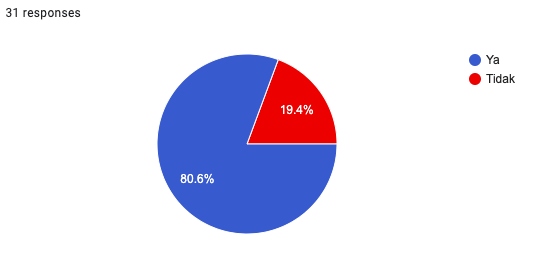
Hasil respon pengguna dari kuesioner yang dibuat, menyatakan 83.9% pengguna dapat menambah tag baru dan 16.1% pengguna tidak dapat menambah anggota tag baru.

* 1. Menambahkan Tag pada Asset



Gambar 4. 23 Testing Menambahkan Tag pada Asset

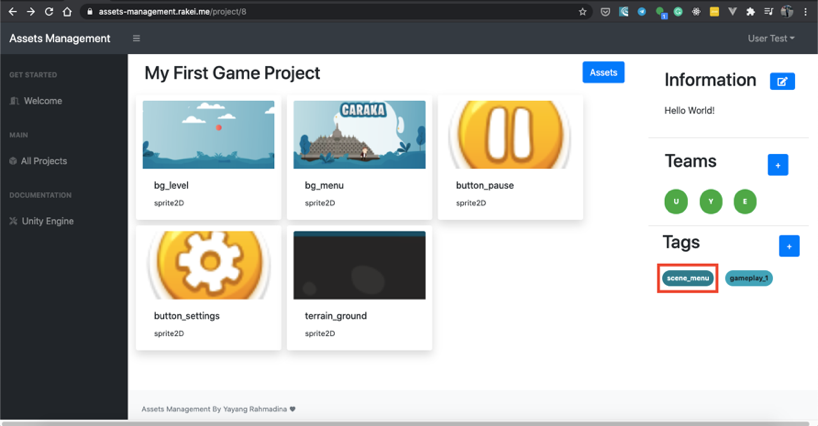
Pada saat pengguna membuat mengedit asset atau menambahkan asset baru, pengguna akan mendapatkan field untuk menambahkan attribute tag dari tag yang sudah ditambahkan sebelumnya.



Gambar 4. 24 Chart Hasil Testing Menambahkan Tag Pada Asset

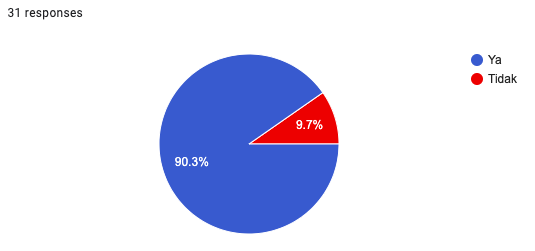
Hasil respon pengguna dari kuesioner yang dibuat, menyatakan 80.6% pengguna dapat menambah tag pada asset dan 19.4% pengguna tidak dapat menambahkan tag pada asset

* 1. Mengakses Asset pada Tag Tertentu



Gambar 4. 25 Testing Halaman Asset pada Tag Tertentu

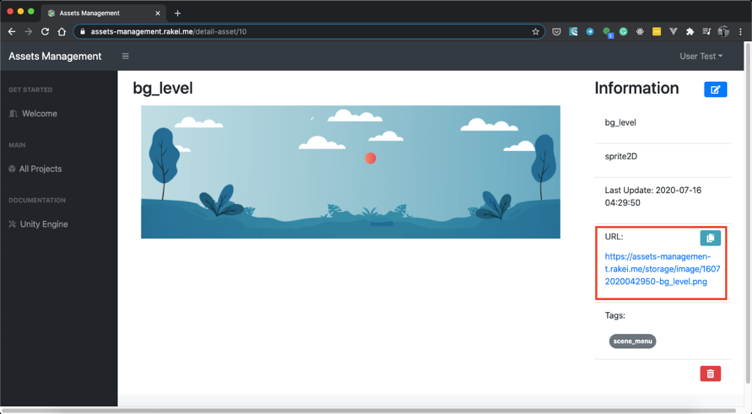
Pada halaman asset, pengguna akan menjumpai informasi dari tag-tag yang tersemat di asset-asset project tersebut. Jika salah satu tag ditekan, pengguna akan diarahkan menuju preview asset yang mengandung tag tersebut.



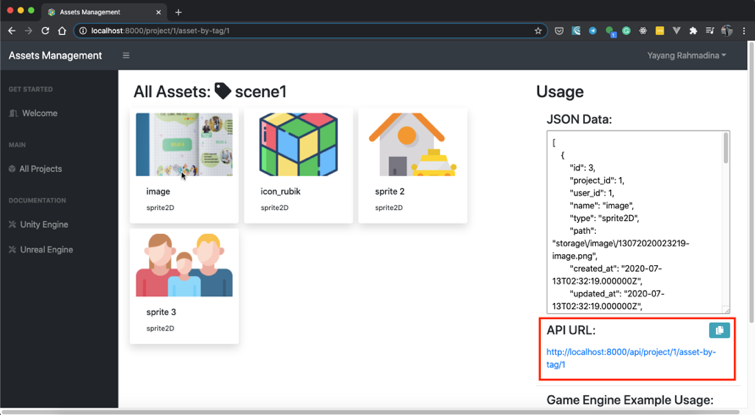
Gambar 4. 26 Chart Hasil Testing Halaman Asset pada Tag Tertentu

Hasil respon pengguna dari kuesioner yang dibuat, menyatakan 90.3% pengguna dapat mengakses tag tertentu dan 9.7% pengguna tidak dapat mengakses tag tertentu.

* 1. URL dan API

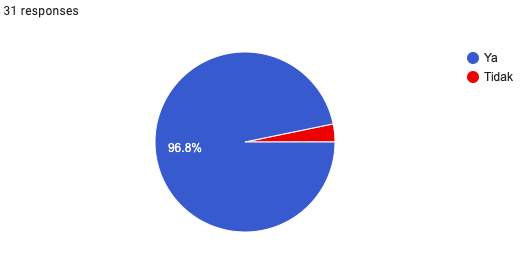


Gambar 4. 27 Testing Halaman URL Image



Gambar 4. 28 Testing Halaman API Get Asset

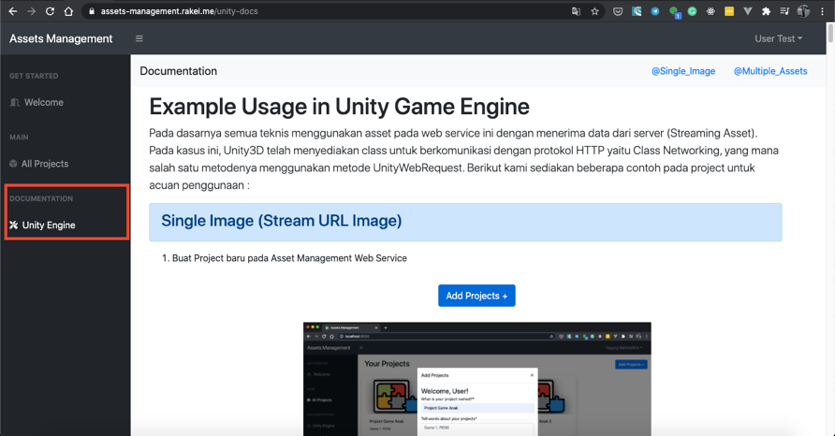
Pada halaman asset pengguna menjumpai URL gambar, begitu pula pada saat mengakses asset pada salah satu tag, pengguna disuguhkan API URL yang dapat anda gunakan untuk pengembangan game selanjutnya.



Gambar 4. 29 Chart Hasil Testing URL Asset

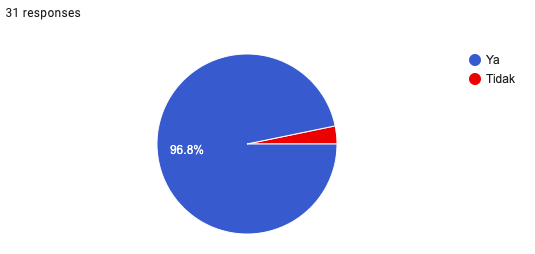
Hasil respon pengguna dari kuesioner yang dibuat, menyatakan 96.8% pengguna dapat menggunakan URL API dengan baik dan 3.2% pengguna tidak dapat dapat menggunakan URL API.

* 1. Dokumentasi Penggunaan Pada Game Engine



Gambar 4. 30 Testing Halaman Dokumentasi

Pada sidebar kiri, bagian “Documentation” tab “Unity Engine” pengguna dapat mengakses petunjuk dan contoh penggunaan fitur web service ini pada game engine Unity, pengguna juga dapat mengunduh unitypackage yang sudah penulis sediakan untuk memudahkan menggunakan fitur manajemen asset ini.



Gambar 4. 31 Chart Hasil Testing Halaman Dokumentasi

Hasil respon pengguna dari kuesioner yang dibuat, menyatakan 96.8% pengguna dapat menahami dokumentasi dengan baik dan 3.2% pengguna tidak dapat dapat memahami dokumentasi.

# BAB V PENUTUP

Pada bab ini dijelaskan mengenai kesimpulan dari pengujian langsung ke pengguna melalui survey.

## Kesimpulan

Berdasarkan pengujian yang dilakukan pada bab sebelumnya maka dapat disimpulkan bahwa:

1. Hasil pengujian request API menggunakan tool Postman mendapatkan hasil yang sesuai dan dapat digunakan pada pengembangan game di project Unity selanjutnya.
2. Hasil pengujian fungsionalitas pada developer/pengguna mendapatkan hasil yang menyatakan bahwa aplikasi yang sudah terpasang pada cloud server berjalan lancer dengan terdapat beberapa kendala di beberapa bagian. Namun kendala tersebut hanya Sebagian kecil dari seluruh pengguna yang melakukan uji coba sehingga tidak memiliki dampak yang signifikan bagi aplikasi.
   1. Saran

Dari hasil proyek akhir ini masih terdapat beberapa kekurangan dan dimungkinkan untuk pengembangan lebih lanjut. Saran dari pengembangan proyek akhir ini diantaranya adalah:

1. Memperbaiki usabilitas aplikasi dari segi UI UX supaya lebih efisien bagi developer/ pengguna.
2. Memperkaya dokumentasi pada banyak jenis asset untuk pengembangan game, karena dokumentasi cenderung pada satu jenis game asset yaitu image.
3. Menambahkan sekuritas pada system agar data-data pengguna lebih aman digunakan.
4. Membuat tools yang lebih terotomasi, efisien dan lebih mudah digunakan pada game engine.

*Halaman ini sengaja dikosongkan*

# DAFTAR PUSTAKA

[1] Asosiasi Game Indonesia. Developer Game Online Indonesia Paling Maju Di Asia Tenggara. Artikel, Http://www.agi.or.id/?p=960. 2015 (Diakses pada 10 Juli 2019)

[2] Hill-Whittall, Richard. *The Indie Game Developer Handbook.* New York And London: Focal Press, 2015. 52.

[3] Breinstein, P., Arnaud, R., Ardolino, A., Franco, S., Herubel A., McChutchan, J., Nedelcu, N., Nistchke, B., Olmstead, D., Robinet, F., Ronchi, C., Turkowski, R., Walter, R., Samour, G. *Game Development Tools Essentials.* US: Apress, 2014. 3-6.

[4] Denny. “Appcrafted — a simple web service to manage content in your game [PRE-RELEASE]”.Internet: <https://forum.unity.com/threads/appcrafted-a-simple-web-service-to-manage-content-in-your-game-pre-release.222649/#post-1492441>

. June 24, 2015. [17 Juli 2020].

[5] Wu, Brandon. “Using REST API with Unity Part 1”.Internet: <https://hub.packtpub.com/using-rest-api-unity-part-1-what-rest-and-basic-queries/>. Jan 15, 2014. [20 Desember 2019].

[6] JSON.ORG , Introducing JSON. Artikel, https://www.json.org/json-en.html (Diakses 10 Juli 2019)

[7] Unity Documentation, UnityWebRequest. Artikel, https://docs.unity3d.com/540/Documentation/Manual/UnityWebRequest.html (Diakses 22 Oktober 2019)