

Pembaruan Data Permainan Berbasis Downloadable Content pada Permainan Action Tower Defense : Saga Legacy

Achmad Ridlo Nuur Abdillah¹, Supeno Mardi Susiki Nugroho²

¹PT. MOLCA TEKNOLOGI NUSANTARA

²Dept. Teknik Komputer Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia

Email: arna.ridlo@molca.id, mardi@its.ac.id

Abstrak

Dalam pengembangan sebuah game, banyak hal yang harus diperhatikan untuk membuat pengalaman bermain tetap menarik. Mulai dari jalan cerita sampai mekanisme gameplay. Salah satu cara paling efektif agar pemain tetap tertarik memainkan game yang kita buat adalah melakukan update secara berkala. Hal ini dapat memberikan pengalaman baru secara terus menerus kepada pemain. Namun, terkadang untuk merilis update, pengguna diharuskan untuk mengunduh ulang keseluruhan assets pada game tersebut. Hal ini dirasa kurang efektif apabila pengembang hanya menginginkan untuk mengubah data berukuran kilobyte saja. Oleh karena itu, perlu adanya sistem yang dapat membantu pengembang merilis update secara cepat. Dengan Real-Time Database Management System pengembang dapat merilis update tanpa mengharuskan pemain untuk mengunduh ulang pada client pada distributor bahkan saat pemain sedang memainkan game tersebut. Memungkinkan pemain untuk memilih konten mana yang ingin disimpan.

Kata kunci: Downloadable Content, Realtime Database, Penyimpanan Awan

Diterima Redaksi: 1-11-2023 Selesai Revisi: 16-11-2023 Diterbitkan Online: 15-12-2023

DOI: DOI: <https://doi.org/10.59378/jcenim.v1i3.38>

I. PENDAHULUAN

Sebagai pengembang permainan, untuk mempertahankan pengalaman bermain agar tetap menarik bagi pengguna, diperlukan pembaruan permainan secara berkala sehingga permainan yang dimainkan tidak menimbulkan kejenuhan [1, 3]. Namun, dari sisi pengguna, pembaruan konten permainan sering kali menjadi tidak praktis apabila mengharuskan pemasangan ulang seluruh berkas permainan untuk mengimplementasikan pembaruan dari pengembang [4, 7]. Kondisi tersebut dinilai kurang efektif dan efisien, terutama apabila pembaruan yang dilakukan hanya bersifat pemeliharaan dengan ukuran data yang relatif kecil, misalnya dalam skala kilobyte [4].

Permasalahan lain yang muncul dalam pengelolaan konten permainan adalah penerapan sistem *Downloadable Content* (DLC), di mana pengguna diwajibkan untuk memasang keseluruhan konten DLC sebelum dapat mengakses atau memainkannya [5]. Hal ini menjadi semakin bermasalah pada platform permainan mobile, karena DLC sering kali digabungkan dalam satu paket dengan aplikasi permainan utama meskipun tidak seluruh konten tersebut dibutuhkan oleh pengguna [6, 9]. Oleh karena itu, pengembang permainan memerlukan suatu sistem yang mampu menangani manajemen data permainan secara efisien dan fleksibel kepada pengguna [1, 8].

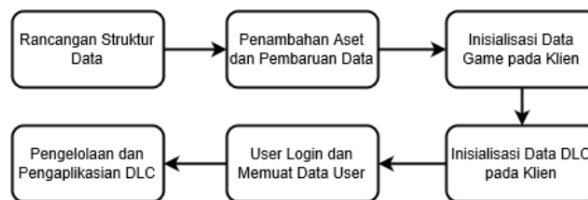
Salah satu pendekatan yang dapat diterapkan untuk mengatasi permasalahan tersebut adalah metode *seamless update* dalam manajemen data permainan [7, 10]. Dengan sistem manajemen data yang berjalan secara dinamis, pengguna tidak perlu mengunduh ulang berkas permainan secara keseluruhan setiap kali terjadi pembaruan [3, 4]. Penerapan *seamless update* memerlukan dukungan basis data yang beroperasi secara *real-time* untuk memastikan sinkronisasi data berjalan dengan baik [2, 8]. Nilai data (*value data*) dalam permainan disimpan pada infrastruktur komputasi awan di sisi server, sehingga permainan dapat melakukan pembaruan secara otomatis ketika terjadi perubahan data. Dengan demikian, nilai data pada permainan dapat bersifat dinamis dan selalu selaras dengan pembaruan yang dilakukan oleh pengembang [1, 6, 10].

II. DESAIN SISTEM

A. Desain Sistem

Bagian ini menjelaskan perancangan sistem *seamless update* yang diterapkan pada permainan aksi *tower-defense* Saga Legacy. Sistem yang dirancang terbagi menjadi dua komponen utama, yaitu *game data update* yang bertugas memperbarui data karakter dan skill, serta *game assets update* yang menangani pembaruan aset permainan seperti musik, sprite karakter, dan elemen visual lainnya. Seluruh data karakter dan skill dalam permainan mengacu pada Firebase *real-time database*, di mana setiap perubahan data pada basis data akan diteruskan secara langsung ke seluruh klien yang sedang aktif.

Pada permainan Saga Legacy, diterapkan mekanisme *listener* untuk mendeteksi adanya perubahan data pada basis data. Ketika terjadi perubahan, sistem secara otomatis mengambil data terbaru, memperbarui data yang tersimpan pada permainan, dan mengaplikasikannya ke dalam *gameplay*. Gambaran umum dari sistem basis data waktu nyata yang digunakan dalam permainan ini ditunjukkan pada Gambar 1.



Gambar 1: Gambaran umum desain sistem

B. Rancangan Struktur Data

Semua data pada sistem database waktu nyata Firebase disimpan sebagai objek JSON. Ketika data ditambahkan ke JSON tree, data tersebut menjadi sebuah node pada struktur JSON yang ada dengan kunci terkait. Database waktu nyata Firebase membolehkan nesting data hingga 32 tingkat kedalaman, namun ketika pengguna mengambil data pada sebuah lokasi di database, maka semua child pada data tersebut juga ikut terambil. Karena itu perlu, dirancang struktur data dengan jumlah nesting seminimal mungkin.

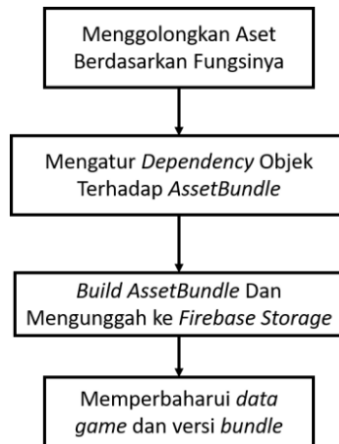
Data-data yang disimpan dalam database adalah sebagai berikut:

1. Data DLC (Downloadable Content)
2. Data Hero
3. Data Troop
4. Data Skill
5. Data Stage
6. Data Pengguna

C. Penambahan Aset dan Pembaruan Data

Tahap selanjutnya dalam sistem pembaruan waktu nyata adalah proses penambahan aset dan pembaruan data pada basis data berbasis komputasi awan. Untuk melakukan pembaruan aset, Unity3D menyediakan mekanisme *AssetBundle* yang digunakan untuk menyimpan berbagai aset permainan, seperti musik, gambar, dan elemen visual lainnya. *AssetBundle* dibangun menggunakan UnityEditor dan aset yang tersimpan di dalamnya dapat digunakan pada versi rilis permainan sebagai sarana untuk memperbarui aset tanpa harus merilis ulang keseluruhan aplikasi permainan, selama tidak diperlukan perubahan pada sistem inti permainan.

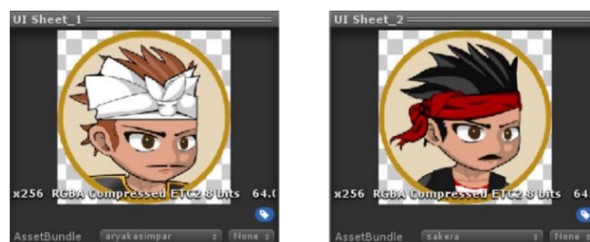
Sementara itu, data masing-masing karakter pada sisi klien mengacu pada sistem basis data waktu nyata, sehingga data permainan dapat tetap terbaru selama klien memiliki koneksi internet. Dengan pendekatan ini, pembaruan data dan aset dapat dilakukan secara dinamis dan terintegrasi. Tahapan dalam pembuatan dan integrasi *AssetBundle* dengan sistem basis data ditunjukkan pada Gambar 2.



Gambar 2: Alur integrasi Unity3D dengan Firebase

Menggolongkan Aset Berdasarkan Fungsinya

Dalam pengembangan game, aset-aset yang digunakan digolongkan berdasarkan fungsi dari aset tersebut didalam game. AssetBundle berfungsi sebagai arsip dari masing-masing kategori dari aset dalam game tersebut. Pada game Saga Legacy, terdapat banyak karakter mulai dari pahlawan hingga musuh, dan setiap karakter memiliki aset yang berbeda. Aset-aset tersebut nantinya digolongkan pada AssetBundle yang berbeda. Semua aset seperti animasi, sprite, maupun senjata digolongkan berdasarkan karakter masing-masing. Aset-aset ini nantinya akan dimuat dari AssetBundle saat aset tersebut dibutuhkan. Misal ikon dari salah satu karakter bernama Arya Kasimpar, akan dimasukkan kedalam AssetBundle *aryakasimpar*, sedangkan ikon karakter lain bernama sakera, akan dimasukkan kedalam AssetBundle *sakera*. Penggolongan ini nantinya akan memudahkan pengembang dalam mengatur dependency objek seperti antarmuka saat memilih hero yang membutuhkan ikon-ikon tersebut, sehingga dapat diakses dengan memuat AssetBundle sesuai dengan nama dari karakter yang dibutuhkan. Contoh dari penggolongan ini dapat dilihat pada gambar 3.6 dibawah ini.

Gambar 3: Sprite pada AssetBundle *aryakasimpar* dan *sakera*

Mengatur Dependency Objek Terhadap AssetBundle

Karena aset yang digunakan dalam game merujuk pada AssetBundle, maka perlu adanya data yang mencatat nama AssetBundle dan nama file yang dibutuhkan oleh masing-masing GameObject agar aset pada AssetBundle dapat dimuat. Informasi mengenai nama AssetBundle dan nama aset dalam aset bundle disimpan pada 3 struktur yaitu *HeroData*, *TroopData*, dan *SkillData*. Berikut potongan dari data tersebut pada database dapat dilihat pada gambar 4 dan 5.

Dapat dilihat pada struktur data pada gambar 4 dan 5 terdapat variabel dengan nama “iconBundle” dan “iconName”. Dua variabel tersebut merupakan informasi mengenai dimana ikon dari objek tersebut berada. “iconBundle” menyimpan nama dari AssetBundle sedangkan “iconName” menyimpan nama aset didalam AssetBundle dengan nama yang sudah disebutkan. Data tersebut disimpan dalam real-time database dengan format json yang biasa digunakan saat transmit data.



Gambar 4: Nilai data hero pada firebase database



Gambar 5: Nilai data skill dan troop pada firebase database

Build AssetBundle dan Mengunggah ke Firebase Storage

Setelah aset-aset digolongkan berdasarkan fungsinya, AssetBundle di-build menggunakan AssetBundle BuildPipeline yang mengarsipkan aset-aset tersebut menjadi file ter-serialisasi. AssetBundle memiliki 2 struktur sedikit berbeda tergantung apakah AssetBundle tersebut merupakan normal bundle atau scene bundle. Normal bundle merupakan AssetBundle yang berisi aset-aset seperti musik, gambar, dan lain-lain. Sedangkan scene bundle berisi scene-scene yang dapat dimuat secara asinkron saat game berjalan. Menyimpan scene kedalam AssetBundle membuat aset-aset yang digunakan pada scene tersebut tidak ikut kedalam versi build sehingga ukuran file instalasi menjadi lebih kecil. AssetBundle memiliki AssetBundleManifest yang menyimpan data aset dan dependency-nya terhadap aset-aset yang lain. AssetBundle yang telah diunggah ke Firebase Storage dapat dilihat pada gambar 6.

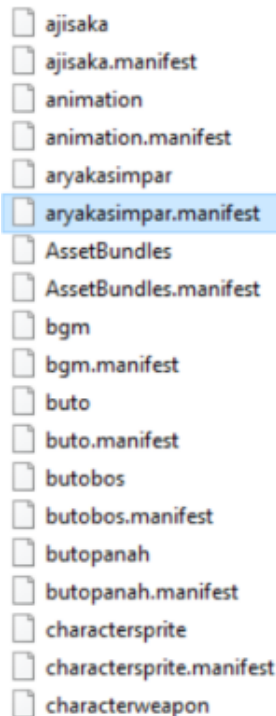
AssetBundle yang sudah di-build kemudian diunggah ke Firebase Storage berdasarkan versi yang sudah ditentukan pengembang.

Setiap folder pada gambar 7 merepresentasikan versi dari AssetBundle yang dapat diakses oleh klien untuk mengunduh aset-aset yang akan digunakan didalam permainan. Versi mana yang harus diunduh oleh klien bergantung pada nilai "version" pada struktur DLC dalam database waktu nyata Firebase, dapat dilihat pada gambar 8.

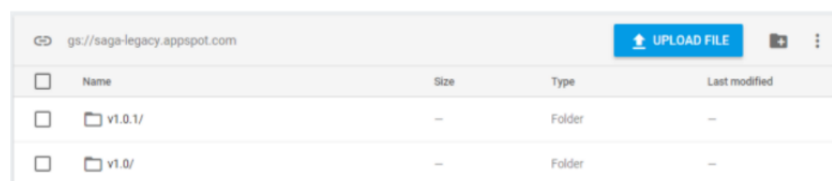
Setiap file yang ada didalam Firebase Storage memiliki storageReference yang dapat digunakan untuk mengambil link dari file yang bersangkutan. AssetBundle pada Firebase Storage dapat dilihat pada gambar 9.

Memperbaharui Data Game dan Versi Bundle

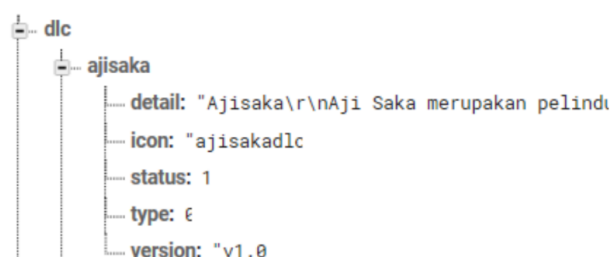
Setelah AssetBundle ter-unggah kedalam Firebase Storage. Klien tidak akan langsung mengunduh AssetBundle baru tersebut. Segala perubahan terhadap dependency juga harus dirubah melalui realtime database. Pengembang perlu memperbaharui nilai dari "assetbundleversion" pada Firebase Real-time Database agar perbedaan nilai versi dapat ditangkap oleh Listener pada klien. Kemudian klien akan melakukan pengecekan ulang terhadap aset-aset yang tersimpan pada penyimpanan lokal. Verifikasi aset akan dijelaskan pada subbab berikutnya.



Gambar 6: Hasil dari build AssetBundle pada Firebase Storage



Gambar 7: Folder versi AssetBundle pada Firebase Storage



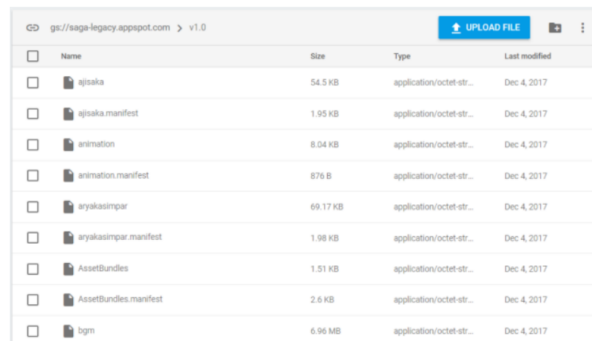
Gambar 8: Versi assetbundle yang sedang aktif digunakan sebagai referensi pada database dan klien

D. Inisialisasi Data Game pada Klien

Pada subbab sebelumnya, telah dijelaskan langkah-langkah dalam pengembangan sistem pengelolaan data game waktu nyata. Pada subbab ini akan dijelaskan proses integrasi klien dengan database Firebase sehingga data permainan menjadi responsive terhadap pembaruan. Berikut merupakan alur dari proses inisialisasi data permainan, diilustrasikan pada gambar 10.

Integrasi Unity3D dengan Firebase

Sebelum sistem seamless update dapat berfungsi, pertama yang harus diatur terlebih dahulu ialah menyiapkan segala kebutuhan dari software dan plugin yang digunakan dalam pembuatan aplikasi, terutama pengintegrasian IDE Unity3D dengan platform pengembangan Firebase. Software dan plugin ini digunakan untuk menyimpan serta menghubungkan data pada game Saga Legacy di penyimpanan perangkat



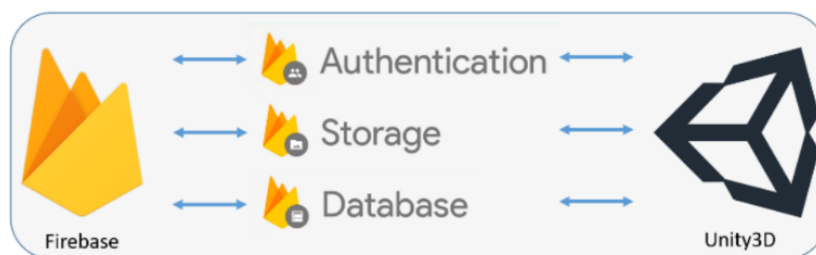
Name	Size	Type	Last modified
ajisaka	54.5 KB	application/octet-str...	Dec 4, 2017
ajisaka.manifest	1.95 KB	application/octet-str...	Dec 4, 2017
animation	8.04 KB	application/octet-str...	Dec 4, 2017
animation.manifest	876 B	application/octet-str...	Dec 4, 2017
aryakasimpar	69.17 KB	application/octet-str...	Dec 4, 2017
aryakasimpar.manifest	1.98 KB	application/octet-str...	Dec 4, 2017
AssetBundles	1.51 KB	application/octet-str...	Dec 4, 2017
AssetBundles.manifest	2.6 KB	application/octet-str...	Dec 4, 2017
bgm	6.96 MB	application/octet-str...	Dec 4, 2017

Gambar 9: AssetBundle pada Firebase Storage



Gambar 10: Alur inisialisasi data permainan pada klien

lokal dengan cloud database sehingga setiap perubahan data yang terjadi pada cloud database dapat diteruskan ke setiap perangkat dalam waktu nyata. Pada proyek ini, saya memakai tiga SDK Firebase yang dapat diilustrasikan pada gambar 11.



Gambar 11: Integrasi Unity3D dengan Firebase menggunakan tiga firebase sdk

Saat game pertama kali dibuka, terdapat `ConnectionHandler` yang mengatur segala koneksi client dengan firebase. Diawali dengan menginisiasi `FirebaseApp` dan mengatur url database untuk real-time database. Dilanjutkan dengan menginisiasi `Firebase Storage` dan memulai listener yang menangani perubahan data. Data tersebut akan tersinkronisasi dengan klien sehingga apabila sewaktu-waktu terjadi perubahan, data pada klien dapat langsung diperbarui. Proses ini dapat dilihat pada gambar 12.

Inisialisasi Listener Data Permainan pada Klien

Inisiasi Listener dilakukan diawal ketika game baru saja dijalankan. Pemain akan dihadapkan pada scene verifikasi data, yang mencocokkan data pada penyimpanan lokal dengan database. Ada 4 listener data permainan yang diinisialisasi pada tahap ini, sesuai dengan rancangan struktur data pada subbab 3.2. Struktur data tersebut yaitu data hero, data troop, data skill, dan data stage. Nilai data dari keempat struktur ini akan senantiasa tersinkronisasi dengan database waktu nyata Firebase.



Gambar 12: Proses integrase data permainan dengan Firebase

Mengambil Data Permainan dari Firebase Database

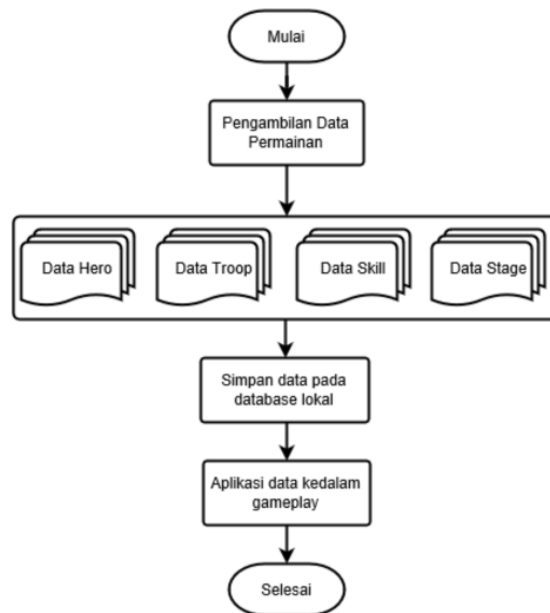
Setelah inisialisasi listener, sistem akan mengambil nilai dari keempat tersebut untuk memastikan data yang dipakai pada klien merupakan data permainan yang terbaru sebelum permainan dimulai. Proses ini dapat dilihat pada gambar 13.

Mengaplikasikan Data Pada Database Lokal

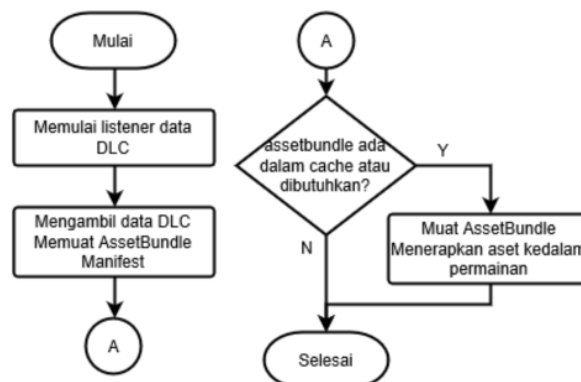
Berdasarkan struktur dari data. Data yang telah diambil dari database Firebase mendapat perlakuan yang berbeda. Untuk data hero, troop, dan skill perubahan akan langsung diteruskan kedalam gameplay. Tidak peduli meskipun pemain sedang didalam stage, data akan langsung dirubah dan mempengaruhi gameplay. Sedangkan untuk data stage. Perubahan baru terjadi hanya pada awal stage.

E. Inisialisasi Data DLC pada Klien

Setelah data permainan terinisialisasi sistem akan melakukan pengecekan aset pada penyimpanan lokal. Dimulai dengan menginisialisasi listener data DLC yang menyimpan informasi aset yang dibutuhkan untuk menjalankan permainan. Data ini digunakan untuk menentukan apakah suatu assetbundle harus diunduh atau tidak berdasarkan status dari DLC. Alur dari proses verifikasi aset ini dapat dilihat pada gambar 14.



Gambar 13: Alur pengambilan dan pengaplikasian data permainan



Gambar 14: Alur inisialisasi data DLC pada klien

Memuat AssetBundle dari Cache

Apabila nilai hash dari AssetBundle yang tersimpan sudah sama dengan nilai pada database. Sistem akan mencoba untuk memuat aset dari cache pada penyimpanan lokal. Proses pengambilan data AssetBundle dari cache dapat dilihat pada gambar 15.

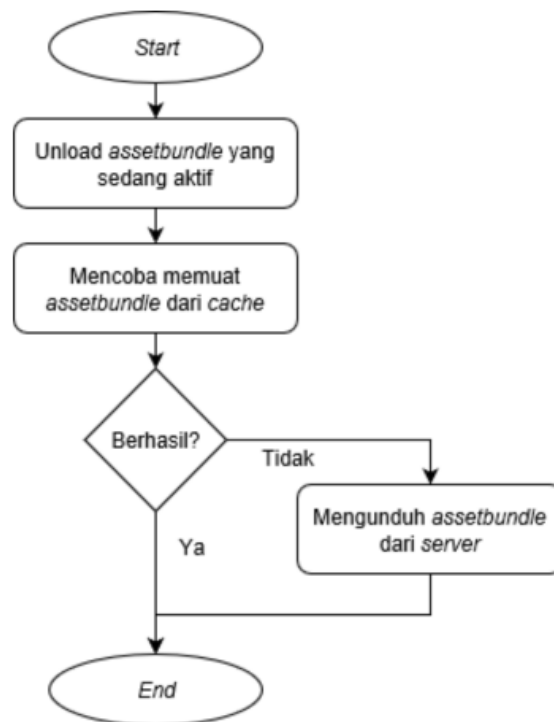
Mengunduh AssetBundle dari Server

Apabila sistem tidak berhasil memuat assetbundle dari penyimpanan lokal maka menggunakan informasi url, hash, dan versi dari bundle, sistem akan mengunduh assetbundle pembaharuan dari server, kemudian menyimpannya pada penyimpanan lokal. Proses pengunduhan assetbundle baru dapat dilihat pada gambar 16.

F. User Login dan Memuat Data Pengguna

Setelah semua data DLC terinisialisasi, pengguna akan dihadapkan pada panel login. Pada panel ini, pengguna dapat memilih untuk melakukan login menggunakan surel apabila pengguna telah memiliki akun sebelumnya. Berikut adalah sketsa dari panel login, ditunjukkan pada gambar 17.

1. Input surel pengguna



Gambar 15: Proses muat AssetBundle dari cache

2. Input password akun
3. Tombol login
4. Tombol registrasi akun

Apabila pengguna belum memiliki akun, pengguna dapat melakukan registrasi menggunakan surel dengan menekan tombol registrasi. Berikut merupakan ilustrasi dari panel registrasi, dapat dilihat pada gambar 18.

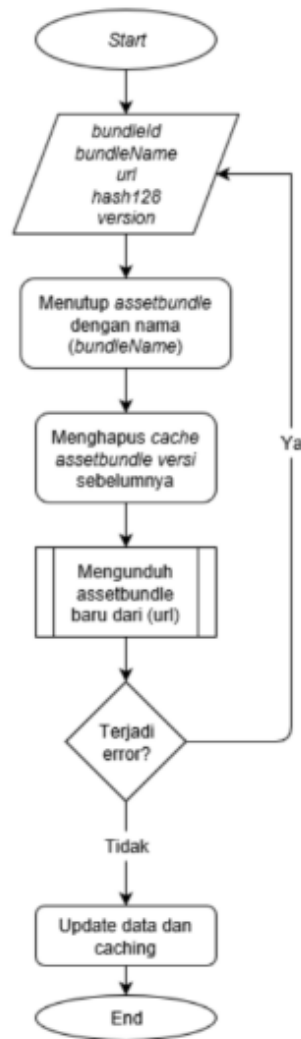
1. Input surel pengguna
2. Input password akun
3. Tombol registrasi menggunakan surel
4. Tombol batal registrasi

Setelah pengguna berhasil masuk menggunakan surel mereka, sistem akan mengambil data pengguna dari akun terkait dan membaca daftar DLC yang pernah diunduh. Masing-masing DLC pada daftar DLC akan diunduh kedalam penyimpanan lokal perangkat.

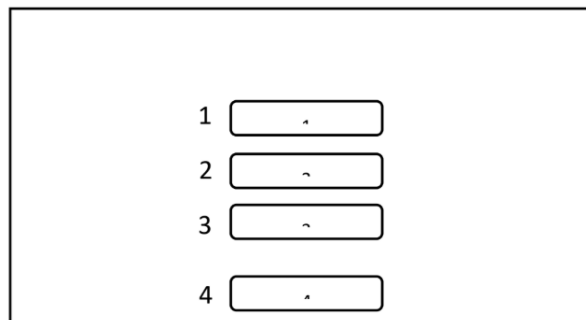
G. textitPengelolaan dan Pengaplikasian DLC

Didalam permainan terdapat DLC Manager yang dapat digunakan oleh pemain untuk mengelola DLC yang tersedia. Panel DLC Manager ini memungkinkan pemain untuk mengunduh dan menghapus setiap DLC yang tersedia pada Saga Legacy. Pembaruan aset melalui DLC ini dilakukan dengan seamless update, yaitu tanpa loading screen ataupun mengharuskan pemain untuk membuka ulang Saga Legacy. Sehingga pemain dapat mengelola DLC tanpa mengganggu jalannya permainan. Desain dari DLC Manager dapat dilihat pada gambar 19 berikut.

1. Ikon dari DLC
2. Deskripsi mengenai DLC
3. Status dari DLC apakah tersedia atau sudah diunduh



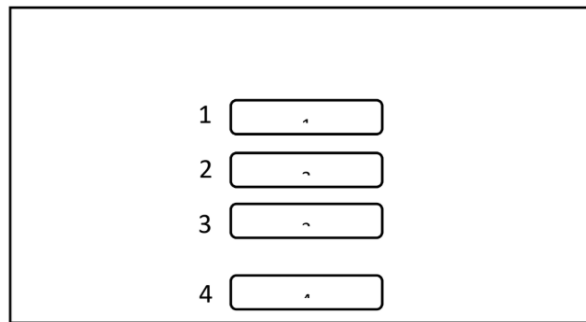
Gambar 16: Proses unduh assetbundle dari server



Gambar 17: Sketsa panel login menggunakan surel

4. Tombol unduh atau menghapus DLC

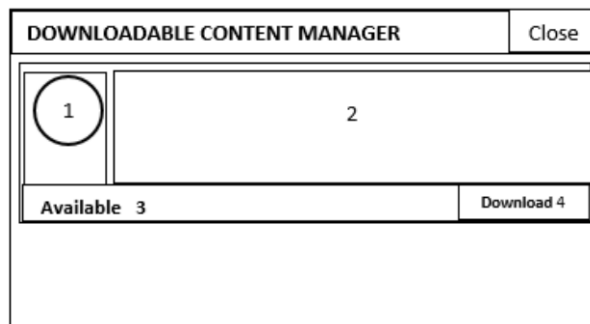
Penunjuk status DLC juga digunakan untuk menampilkan progress unduh ketika pemain melakukan pengunduhan DLC. Setelah proses pengunduhan selesai, nama DLC akan ditambahkan pada data pengguna sebagai referensi apabila pengguna melakukan login pada perangkat lain, sehingga data tetap tersinkronisasi antar perangkat.



A registration panel with four numbered input fields:

- 1
- 2
- 3
- 4

Gambar 18: Panel registrasi menggunakan surel



A sketch of a DLC Manager interface. It features a title bar with 'DOWNLOADABLE CONTENT MANAGER' and a 'Close' button. Below the title bar is a large rectangular area divided into two sections: a left section labeled '1' and a right section labeled '2'. At the bottom, there are two buttons: 'Available 3' and 'Download 4'.

Gambar 19: Sketsa DLC Manager

III. PENGUJIAN

A. Implementasi Pembaruan Data dan Penambahan DLC

Ada beberapa langkah yang harus dilakukan untuk menambah konten baru pada permainan Saga Legacy. Mulai dari menggolongkan aset hingga penambahan data dlc baru pada database waktu nyata Firebase. Pada pengujian ini, akan dilakukan penambahan karakter dengan mengunduh DLC yang tersedia. Adapun alur penambahan dan pembaruan data pada pengujian ini adalah sebagai berikut.

Sebelum membangun assetbundle, pengembang harus mengelompokkan masing-masing aset dalam permainan berdasarkan fungsinya. Dari hasil pengelompokan tersebut, berikut adalah daftar nama dan ukuran dari masing-masing assetbundle yang telah dibangun ditampilkan pada tabel 1.

Hampir semua assetbundle pada tabel diatas berstatus “Necessary” yang artinya sistem akan langsung mengunduh aset karena assetbundle tersebut dibutuhkan untuk menjalankan permainan. Hanya dua assetbundle yang berstatus “Downloadable” yaitu Sakera dan Arya Kasimpar yang dapat diunduh apabila diinginkan oleh pemain saat didalam permainan.

Selanjutnya pengembang mengunggah assetbundle yang sudah dibangun ke Firebase Storage dalam folder sesuai dengan versi bundle yang dikehendaki. Nama assetbundle akan digunakan sebagai referensi dari database storage, sehingga meskipun assetbundle pada suatu folder diperbaharui dengan link yang berbeda. Pengguna tetap dapat mengakses assetbundle terkait menggunakan referensi nama tersebut. Berikut adalah assetbundle yang telah dibangun dan diunggah ke Firebase Storage dapat dilihat pada gambar 20.

Setelah aset-aset yang dibutuhkan pada permainan diunggah, pengembang lalu menambahkan data DLC pada Firebase Database agar assetbundle dapat diakses oleh klien. Data DLC memiliki key yang sama dengan nama assetbundle pada Firebase Storage sehingga dapat direferensi untuk mengambil tautan unduh dari DLC terkait. Selain itu, terdapat parameter “version” yang merupakan versi dari assetbundle pada DLC. Version merupakan referensi folder tempat assetbundle berada. Data DLC pada Firebase Database dapat dilihat pada gambar 21.

Tabel 1: Nama dan ukuran dari hasil bangun assetbundle

No.	Nama AssetBundle	Ukuran
1	Ajisaka	106 kb
2	Animation	9 kb
3	Aryakasimpar	71 kb
4	Bgm	7.125 kb
5	Buto	39 kb
6	Butoboss	38 kb
7	Butopanah	45 kb
8	Charactersprite	39 kb
9	Characterweapon	4 kb
10	Fonts	1.596 kb
11	Kenededess	36 kb
12	Prajurit	40 kb
13	Prajuritdukun	46 kb
14	Prajuritpanah	18 kb
15	Prefabs	3.545 kb
16	Rhoma	31 kb
17	Sakera	85 kb
18	Sfx	19 kb
19	Sprite	2.975 kb
20	Textassets	2 kb
21	Weaponanimation	26 kb
22	Zscene	7.650 kb

☐	Name	Size
☐	 ajisaka	78.53 KB
☐	 ajisaka.manifest	2.33 KB
☐	 animation	8.04 KB
☐	 animation.manifest	876 B
☐	 AssetBundles	1.51 KB
☐	 AssetBundles.manifest	2.47 KB
☐	 bgm	6.96 MB
☐	 bgm.manifest	536 B
☐	 buto	38.76 KB
☐	 buto.manifest	1.82 KB
☐	 butobos	37.98 KB
☐	 butobos.manifest	1.81 KB
☐	 butopanah	24.39 KB

Gambar 20: AssetBundle pada Firebase Storage

B. *Pengelolaan DLC (Downloadable Content) pada Klien*

Pemain dapat mengelola DLC yang tersedia dalam permainan dengan membuka DLC Manager. DLC Manager memberikan kemampuan pada pemain untuk memilih mengunduh dan menghapus konten yang ada pada permainan Saga Legacy tanpa mengganggu jalannya permainan. Pada panel DLC Manager terdapat daftar DLC yang tersedia serta penjelasan mengenai DLC terkait.



Gambar 21: Data DLC pada Firebase Database



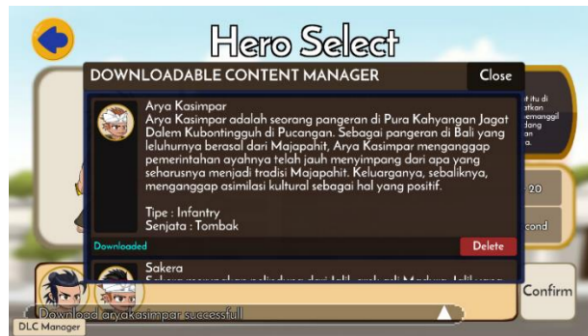
Gambar 22: Screenshot tampilan pilih karakter sebelum mengunduh DLC

Pada gambar 22, merupakan tampilan pilih hero yang menampilkan hanya satu karakter saja pada bar daftar hero, yaitu Aji Saka. Apabila pengguna ingin menambahkan karakter yang dapat dimainkan, pengguna dapat menekan tombol “DLC Manager” pada bagian pojok kiri bawah layar. Kemudian proses memuat “DLC Manager” akan dijalankan seperti yang terlihat pada gambar 23.



Gambar 23: Screenshot pengguna memuat DLC manager

Pada panel DLC Manager terdapat daftar DLC yang tersedia dalam permainan Saga Legacy, salah satunya adalah karakter Arya Kasimpar yang merupakan salah satu hero yang dapat dimainkan. Ketika player menekan tombol download, maka proses pengunduhan assetbundle akan dilakukan tanpa mengganggu permainan.



Gambar 24: Screenshot pengguna telah mengunduh DLC Arya Kasimpar

Setelah proses unduh selesai, aset akan langsung dimuat dan ditampilkan pada tampilan pilih hero. Pengguna kemudian dapat memilih hero tersebut dan dimainkan pada stage seperti pada gambar 25.



Gambar 25: : Screenshot Arya Kasimpar setelah pengunduhan DLC

Dengan sistem ini pengguna dapat memilih konten mana yang ingin disimpan dalam perangkat mereka namun tetap mengetahui apabila pengembang melakukan pembaruan atau penambahan konten.

C. Hasil Pengujian Performa Sistem

Pengujian terhadap performa sistem dibagi menjadi dua bagian. Bagian pertama ialah pengujian sistem terhadap perbedaan jaringan. Tujuan dari pengujian bagian pertama ialah untuk menguji waktu unduh berdasarkan ukuran berkas dan kecepatan jaringan. Pengujian bagian kedua ialah pengujian sistem terhadap perbandingan waktu pembaruan menggunakan DLC Manager dan dengan tanpa menggunakan DLC Manager. Tujuan dari pengujian bagian kedua ialah untuk melihat perbedaan sistem yang menggunakan DLC Manager dan tidak menggunakan DLC Manager terhadap waktu. Pengujian ini dilakukan dengan spesifikasi perangkat sebagai berikut :

1. Operator jaringan : Telkomsel
2. Lokasi : Surabaya, Indonesia

Sedangkan untuk spesifikasi handphone yang digunakan ialah :

1. Merk Handphone : Xiao-Mi A1
2. OS Android : 8.0.0 (Oreo)
3. Chipset :Qualcomm MSM8953 Snapdragon 625
4. CPU : Octa-core 2.0 GHz Cortex-A53
5. GPU : Adreno 506

Pengujian Sistem terhadap Perbedaan Jaringan

Pengujian ini dilakukan untuk mengetahui kecepatan respon sistem dan kecepatan unduh pada 3 jaringan yang berbeda yaitu 2G, 3G, dan 4G.

Pengujian pada jaringan 4G mendapatkan hasil yang ditunjukkan pada tabel 2. Dari data pada tabel dapat dilihat ada ketidakstabilan kecepatan seperti saat mengunduh assetbundle “characterweapon” yang memiliki ukuran hanya 4 kilobyte, membutuhkan waktu 1,38 detik, tidak berbeda jauh dengan waktu mengunduh prefab yaitu 1,65 detik meskipun ukuran file-nya yang jauh lebih besar yaitu 3.545 kilobyte. Selain itu waktu tercepat yang didapat adalah 0.76 detik dengan ukuran bundle 106 kilobyte, meskipun terdapat assetbundle dengan ukuran yang jauh lebih kecil, hal ini bisa terjadi dikarenakan terdapat delay saat permintaan unduhan ataupun faktor lain. Kecepatan maksimal pada jaringan 4G ialah 2873,0 kilobyte tiap detik saat mengunduh berkas AssetBundle Bgm. Sedangkan kecepatan minimum pada jaringan 4G ialah 2,0 kilobyte tiap detik saat mengunduh AssetBundle Textassets. Rata-rata kecepatan pada koneksi 4G yaitu 529,6 kilobyte tiap detik.

Tabel 2: Waktu unduh konten berdasarkan kecepatan dan ukuran berkas pada jaringan 4G

No	Nama AssetBundle	Ukuran AssetBundle	Waktu Mengunduh	Kecepatan (kb/s)
1	Ajisaka	106 kb	0,76 detik	139 kb/s
2	Animation	9 kb	0,10 detik	90 kb/s
3	Aryakasimpar	71 kb	1,66 detik	42,8 kb/s
4	Bgm	7.125 kb	2,45 detik	2.908,2 kb/s
5	Buto	39 kb	1,45 detik	26,9 kb/s
6	Butoboss	38 kb	1,38 detik	27,5 kb/s
7	Butopannah	45 kb	0,82 detik	54,9 kb/s
8	Charactersprite	39 kb	1,36 detik	28,6 kb/s
9	Characterweapon	4 kb	0,28 detik	14,3 kb/s
10	Fonts	1.596 kb	1,02 detik	1.564,7 kb/s
11	Kenededess	36 kb	1,02 detik	35,3 kb/s
12	Prajurit	40 kb	0,78 detik	51,3 kb/s
13	Prajuritdukun	46 kb	0,67 detik	68,7 kb/s
14	Prajuritpanah	18 kb	0,46 detik	39,1 kb/s
15	Prefabs	3.545 kb	1,65 detik	2.148,5 kb/s
16	Rhoma	31 kb	1,14 detik	27,2 kb/s
17	Sakera	85 kb	1,15 detik	73,9 kb/s
18	Sfx	19 kb	0,62 detik	30,6 kb/s
19	Sprite	2.975 kb	1,24 detik	2.398,4 kb/s
20	Textassets	2 kb	1,01 detik	2 kb/s
21	Weaponanimation	26 kb	1,49 detik	17,4 kb/s
22	Zscene	7.650 kb	2,9 detik	2.637,9 kb/s

Dari data pada tabel diatas dapat dilihat ada ketidakstabilan kecepatan seperti saat mengunduh assetbundle “characterweapon” yang memiliki ukuran hanya 4kb, membutuhkan waktu 1,38 detik, tidak berbeda jauh dengan waktu mengunduh prefab yaitu 1,65 detik meskipun ukuran filenya yang jauh lebih besar yaitu 3.545 kb. Selain itu waktu tercepat yang didapat adalah 0.76 detik dengan ukuran bundle 106 kb, meskipun terdapat assetbundle dengan ukuran yang jauh lebih kecil, hal ini bisa terjadi dikarenakan terdapat delay saat permintaan unduhan ataupun faktor lain. Rata-rata kecepatan pada koneksi 4G yaitu 529,6 kb/s. Tabel 3 menunjukkan data pengujian pada jaringan 3G.

Kecepatan unduh pada jaringan 3G mengalami penurunan daripada 4G meskipun tidak terlalu signifikan, rata-rata kecepatan turun dari 529,6 kilobyte tiap detik, menjadi 412,5 kilobyte tiap detik. Assetbundle dengan ukuran kecil cenderung tidak efektif karena terdapat jeda antara inisialisasi unduhan hingga proses unduh itu sendiri dan memperlambat waktu memuat aset. Terutama pada jaringan dengan kecepatan unduh yang tinggi. Data pengujian pada jaringan 2G dapat dilihat pada tabel 4.

Saat mengunduh pada jaringan 2G, kecepatan unduh turun drastis jika dibandingkan dengan pengujian sebelumnya dengan kecepatan maksimal 24,5 kilobyte tiap detik dan paling lambat dengan kecepatan 1,1 kilobyte tiap detik. Kecepatan rata-rata unduh pada jaringan ini adalah 12,4 kilobyte tiap detik. Meski demikian nilai kecepatan unduh yang didapat dari masing-masing assetbundle terbilang seragam.

Tabel 3: Waktu unduh konten berdasarkan kecepatan dan ukuran berkas pada jaringan 3G.

No	Nama AssetBundle	Ukuran AssetBundle	Waktu Mengunduh	Kecepatan (kb/s)
1	Ajusaka	106	1,41	75,2
2	Animation	9	1,03	8,7
3	Aryakasimpar	71	1,63	43,6
4	Bgm	7.125	3,19	2233,5
5	Buto	39	1,40	27,9
6	Butobos	38	1,44	26,4
7	Butopanah	25	0,82	30,5
8	Charactersprite	39	1,01	38,6
9	Characterweapon	4	0,97	4,1
10	Fonts	1.596	1,40	1140,0
11	Kendedes	36	1,39	25,9
12	Prajurit	44	1,01	43,6
13	Prajuritdukun	46	1,40	32,9
14	Prajuritpanah	50	1,41	35,5
15	Prefabs	3.545	2,02	1755,0
16	Rhoma	31	1,38	22,5
17	Sakera	81	1,44	56,3
18	Sfx	19	1,05	18,1
19	Sprite	2.975	2,24	1328,1
20	Textassets	2	1,44	1,4
21	Weaponanimation	7	0,82	8,5
22	Zscene	7.650	3,61	2119,1

Tabel 4: Waktu unduh konten berdasarkan kecepatan dan ukuran berkas pada jaringan 2G.

No	Nama AssetBundle	Ukuran AssetBundle	Waktu Mengunduh	Kecepatan (kb/s)
1	Ajisaka	106 kb	6,28 detik	16,9 kb/s
2	Animation	9 kb	2,15 detik	4,2 kb/s
3	Aryakasimpar	71 kb	6,10 detik	11,6 kb/s
4	Bgm	7.125 kb	326,52 detik	21,8 kb/s
5	Buto	39 kb	4,82 detik	8,1 kb/s
6	Butobos	38 kb	3,54 detik	10,7 kb/s
7	Butopanah	25 kb	2,47 detik	10,1 kb/s
8	Charactersprite	39 kb	2,45 detik	15,9 kb/s
9	Characterweapon	4 kb	1,60 detik	2,5 kb/s
10	Fonts	1.596 kb	75,09 detik	21,3 kb/s
11	Kendedes	36 kb	3,37 detik	10,7 kb/s
12	Prajurit	44 kb	4,26 detik	10,3 kb/s
13	Prajuritdukun	46 kb	3,28 detik	14,0 kb/s
14	Prajuritpanah	50 kb	3,00 detik	16,7 kb/s
15	Prefabs	3.545 kb	242,73 detik	14,6 kb/s
16	Rhoma	31 kb	3,54 detik	8,8 kb/s
17	Sakera	81 kb	5,43 detik	14,9 kb/s
18	Sfx	19 kb	2,14 detik	8,9 kb/s
19	Sprite	2.975 kb	121,42 detik	24,5 kb/s
20	Textassets	2 kb	1,85 detik	1,1 kb/s
21	Weaponanimation	7 kb	2,18 detik	3,2 kb/s
22	Zscene	7.650 kb	358,78 detik	21,3 kb/s

Perbandingan Waktu Pembaruan Menggunakan DLC Manager dengan Tidak Menggunakan DLC Manager

Seiring dengan penambahan konten baru, ukuran file instalasi akan semakin besar, membuat waktu unduh file instalasi semakin lama. Ukuran file instalasi awal sebelum ditambahkan DLC adalah 47,387 kb. Pengujian ini dilakukan untuk mengetahui tingkat efisiensi waktu unduh menggunakan DLC Manager

dengan cara membandingkan waktu unduh tersebut dengan mengunduh file instalasi ulang. Hal tersebut dijelaskan pada persamaan 4.1.

$$x = \frac{\text{Waktu unduh DLC Manager}}{\text{Waktu unduh tidak menggunakan DLC}} \times 100\% \quad (1)$$

Dimana x merupakan presentase waktu unduh dengan menggunakan DLC Manager jika dibandingkan dengan waktu unduh keseluruhan file instalasi. Hasil dari pengujian ini akan didapatkan nilai tingkat efisiensi waktu unduh menggunakan DLC Manager dengan cara menghitung waktu unduh menggunakan DLC Manager dibandingkan dengan waktu unduh tidak menggunakan DLC Manager.

Pengambilan data dalam pengujian ini diambil dari waktu unduh dalam pengunduhan satu demi satu AssetBundle yang terdapat pada tabel 4.1. Hasil dari pengujian waktu unduh menggunakan DLC Manager dengan tidak menggunakan DLC Manager dijelaskan oleh tabel 5.

Tabel 5: Tabel 5 Waktu unduh konten berdasarkan kecepatan dan ukuran berkas pada jaringan Wi-Fi.

No	AssetBundle	Waktu unduh DLC	Waktu unduh tidak DLC	% waktu
1	Ajisaka	0,76 detik	27,01 detik	2,81 %
2	Animation	1,0 detik	28,01 detik	3,55 %
3	Aryakasimpar	1,66 detik	29,67 detik	5,59 %
4	Bgm	2,48 detik	31,15 detik	7,96 %
5	Buto	1,45 detik	32,55 detik	4,45 %
6	Butobos	1,44 detik	33,99 detik	4,24 %
7	Butopanah	0,82 detik	34,81 detik	2,35 %
8	Charactersprite	0,82 detik	35,63 detik	2,30 %
9	Characterweapon	1,38 detik	37,01 detik	3,73 %
10	Fonts	1,02 detik	38,03 detik	2,68 %
11	Kendedes	0,82 detik	38,85 detik	2,11 %
12	Prajurit	0,78 detik	39,63 detik	1,97 %
13	Prajuritdukun	0,82 detik	40,45 detik	2,03 %
14	Prajuritpanah	1,46 detik	41,91 detik	3,48 %
15	Prefabs	1,65 detik	43,56 detik	3,79 %
16	Rhoma	1,42 detik	44,98 detik	3,15 %
17	Sakera	1,44 detik	46,42 detik	3,10 %
18	Sfx	0,78 detik	47,2 detik	1,65 %
19	Sprite	1,67 detik	48,87 detik	3,42 %
20	Textassets	1,01 detik	49,88 detik	1,97 %
21	Weaponanimation	1,49 detik	51,37 detik	2,90 %
22	Zscene	2,9 detik	54,27 detik	5,34 %

Pada pengujian tersebut menunjukkan bahwa rata-rata waktu unduh dengan menggunakan DLC ialah 1,32 detik. Sedangkan rata-rata waktu unduh dengan tidak menggunakan DLC ialah 39,78 detik. Sehingga didapatkan persentase waktu unduh dengan menggunakan DLC Manager jika dibandingkan dengan waktu unduh keseluruhan file instalasi menggunakan persamaan 4.1 ialah 3.31%. Dari pengujian ini dapat ditarik kesimpulan bahwa dengan menggunakan DLC Manager, maka hanya memerlukan waktu 3.31% dari waktu pengunduhan dan instalasi file ulang. Angka 3.31% tersebut menunjukkan bahwa dengan menggunakan DLC Manager, maka akan meningkatkan efisiensi waktu 30 kali lebih cepat user dalam menerima pembaharuan konten dari pengembang.

IV. PENUTUP

A. Kesimpulan

Dengan adanya sistem pengelolaan data permainan waktu nyata pada Saga Legacy ini, data permainan menjadi dinamis dan responsif terhadap penambahan dan pembaruan data, sehingga memudahkan pengembang dan pemain dalam melakukan pembaruan. Pengujian performa sistem dibagi menjadi dua bagian pengujian, yaitu pengujian sistem terhadap perbedaan jaringan dan pengujian sistem terhadap perbandingan waktu pembaruan menggunakan DLC Manager dan dengan tanpa menggunakan DLC Manager. Kesimpulan pada pengujian pertama ialah sistem akan berjalan lebih cepat pada

jaringan 4G dengan kecepatan rata-rata 529,6 kilobyte per detik, tetapi sistem akan berjalan lambat apabila user menggunakan jaringan 2G dengan kecepatan rata-rata 12,37 kilobyte per detik. Lalu pada pengujian kedua mendapatkan kesimpulan bahwa pembaruan data menggunakan DLC dengan seamless update hanya membutuhkan 3.31% waktu yang dibutuhkan dibandingkan dengan pembaruan yang dilakukan dengan cara mengunduh ulang keseluruhan file instalasi yang akan terus bertambah besar seiring dengan terus bertambahnya konten dalam permainan. Sistem ini memungkinkan pemain untuk memilih konten mana yang ingin disimpan pada perangkat dan konten mana yang ingin dihapus tanpa harus menghentikan permainan terlebih dahulu, sehingga meminimalisir gangguan yang terjadi saat pengembang merilis konten baru. Dari rancangan hingga terimplementasinya sistem pengolahan data waktu nyata pada permainan Saga Legacy ini dapat memudahkan pengembang menambahkan konten baru pada permainan mereka, tanpa mengharuskan pemain untuk menghentikan permainan terlebih dahulu.

B. Saran

Penelitian ini masih memiliki beberapa kekurangan, salah satunya adalah belum bisa menambahkan script kedalam AssetBundle. Sehingga ketika pengembang ingin melakukan perubahan pada sistem permainan, user harus mengunduh ulang paket instalasi yang baru. Permasalahan tersebut dapat dijadikan acuan untuk penelitian selanjutnya. Sehingga keseluruhan sistem menjadi seamless atau mengharuskan pemain menghentikan permainan mereka terlebih dahulu.

Pustaka

- [1] A. A. Laghari *et al.*, "Quality of Experience in Cloud Gaming Architectures," *Future Generation Computer Systems*, 2019.
- [2] Y. Chen *et al.*, "Real-Time Data Synchronization for Online Games Using Cloud Computing," in *Proceedings of IEEE Conference*, 2019.
- [3] K. Alha *et al.*, "LiveOps: Managing Content Updates in Mobile Games," *International Journal of Computer Games Technology*, 2020.
- [4] M. Claypool and K. Claypool, "Patch Distribution and Update Efficiency in Online Games," *ACM Digital Library*, 2020.
- [5] D. B. Nieborg, "Downloadable Content and Game-as-a-Service Models," *Entertainment Computing*, 2020.
- [6] W. Li *et al.*, "Cloud-Based Asset Streaming for Online and Mobile Games," *Future Generation Computer Systems*, 2021.
- [7] J. Kim *et al.*, "Live Update System Design for Online Games," in *Proceedings of IEEE Conference*, 2021.
- [8] C. Pahl *et al.*, "Backend-as-a-Service Architectures for Online Applications," *Journal of Cloud Computing*, 2022.
- [9] T. Zhang *et al.*, "Optimizing Data Management in Mobile Online Games," *Mobile Information Systems*, 2022.
- [10] S. Park *et al.*, "Seamless Update Framework for Live Service Games," *IEEE Access*, 2023.