JIKB
Jurnal Ilmu Komputer dan Bisnis
STMIK DHARMAPALA RIAU

Jurnal Ilmu Komputer dan Bisnis (JIKB)
Desember-2023, Vol. XIV, No.2a, hal.72-82
ISSN(P): 2087-3921; ISSN(E): 2598-9715

Perancangan *Realtime Database Firebase* untuk *IoT* dan *Unity* Menggunakan Metode *SDLC*

[1]Dhewan Agum Mahendra, [2]Slamet Winardi
[1,2]Universitas Narotama

Alamat Surat
Email: dhewanagumm@gmail.com, slamet.winardi@narotama.ac.id

Article History:
Diajukan: 25 Oktober 2023**; Direvisi:** 15 November 2023**; Accepted:** 25 November 2023

ABSTRAK

IoT (*Internet of Thing*) merupakan teknologi yang dapat digunakan untuk melakukan kontrol jarak jauh dan *Unity* adalah *game engine* yang sekarang sering dijumpai dalam beberapa aplikasi *mobile* maupun *desktop*. Dalam beberapa pengembangan IoT (*Internet of Thing*) dan *Unity* membutuhkan basis data untuk penyimpanan data, data yang disimpan juga beragam seperti waktu, kondisi, dan sebagainya. Penggunaan *Realtime Database Firebase* pada IoT (*Internet of Thing*) dan *Unity* dengan data yang sama agar bisa saling bertukar data dapat menjadi uji coba pengembangan teknologi di era sekarang. Metode yang digunakan adalah analis permasalahan dan kebutuhan sistem, perancangan *database*, implementasi. Hasil yang diharapkan dari perancangan *database* ini adalah membuatkan basis data yang dapat digunakan untuk saling akses oleh IoT (*Internet of Thing*) dan *Unity*, serta mengetahui bagaimana pengaruh penggunaan *Realtime Database Firebase* pada IoT (*Internet of Thing*) dan *Unity* sekaligus.

Kata kunci: ***Database, Firebase, IoT, Unity*, SDLC**

ABSTRACT

IoT (Internet of Thing) is a technology that can be used to perform remote control and Unity is a game engine that is now often found in several mobile and desktop applications. In some IoT (Internet of Thing) developments and Unity requires a database for data storage, the data stored also varies, such as time, condition, and so on. The use of Realtime Database Firebase in IoT (Internet of Thing) and Unity with the same data so that data can be exchanged can be a test of technology development in the current era. The method used is the analysis of problems and system requirements, database design, implementation. The expected result of designing this database is to create a database that can be used for mutual access by IoT (Internet of Thing) and Unity, and to know the effect of using Realtime Database Firebase on IoT (Internet of Thing) and Unity simultaneously.

Keywords*: *Database, Firebase, IoT, Unity, SDLC

1. PENDAHULUAN

Penggunaan komputer di era sekarang dapat mendominasi beberapa kegiatan yang dapat dilakukan manusia seperti mengontrol peralatan elektronik secara jarak jauh tanpa melakukan interaksi langsung, IoT (*Internet of Things*) memungkinkan para pengguna untuk mengoptimalkan pengelolaan perangkat elektronik atau peralatann listrik menggunakan internet(Adani & Salsabil, n.d.). *Unity* adalah aplikasi atau perangkat lunak untuk mengembangkan game multi-pengguna yang mudah digunakan. *Unity* sangat bagus dan dikemas dengan fitur yang dibuat untuk para profesional(Pangestu et al., 2021).

https://doi.org/10.47927/jikb.v14i2a.525

Dalam Pengembangan beberapa project IoT (*Internet of Things*) dan *Unity* sering kali membutuhkan sebuah basis data untuk menyimpan berbagai macam data seperti waktu, kondisi, dan data-data lain yang diperlukan untuk project IoT (*Internet of Things*) maupun *Unity*. *Firebase* merupakan suatu layanan dari Google berupa *Cloud Storage* yang memiliki berbagai macam fitur berfungsi mempermudah para pengembang (*developer*) untuk mengembangkan aplikasi yang dikerjakan. *Firebase* adalah Baas (*Backend as a Service*) merupakan solusi yang ditawarkan oleh Google untuk memudahkan dan mempercepat pekerjaan *developer*(Pangestu et al., 2021). Dari penjelasan Firebase bisa disimpulkan jika IoT (*Internet of Things*) dan *Unity* dapat menggunakan Firebase sebagai *Cloud Database* untuk penyimpanan data. Salah satu fitur Firebase yang bisa dipakai untuk penyimpanan data secara cloud adalah *Realtime Database Firebase*. Penggunaan *Realtime Database Firebase* pada IoT (*Internet of Thing*) dan *Unity* dengan data yang sama agar bisa saling bertukar data dapat menjadi uji coba pengembangan teknologi di era sekarang.

Dari pembahasan diatas tepatnya pada uji coba Penggunaan *Realtime Database Firebase* pada IoT (*Internet of Thing*) dan *Unity* dengan data yang sama , maka dapat dibuat rumusan permasalahan sebagai berikut :

1. Bagaimana Perancangan dan konfigurasi *Database Firebase* untuk IoT dan *Unity* menggunakan *Realtime Database Firebase*?
2. Bagaimana cara membuat *Endpoint Realtime Database Firebase* untuk akses *database* dari IoT dan *Unity*?
3. Bagaimana pengaruh dari penggunaan *Firebase* pada IoT dan *Unity* yang sudah saling tersambung?

Tujuan dari penelitian ini adalah agar dapat merancang *Realtime Database Firebase* yang dapat digunakan IoT (*Internet of Things*) dan *Unity* dengan Enpoint untuk akses masing-masing data, sekaligus mengetahui pengaruh penggunaan *Realtime Database Firebase* pada IoT (*Internet of Things*) dan *Unity*. Dari lingkup penelitian ini memiliki beberapa batasan mengenai apa yang dibahas. Bahwa Penelitian ini hanya membuat Endpoint, merancang dan mengkonfigurasi *Realtime Database Firebase* sebagai media pertukaran data antar IoT (*Internet of Things*) dan *Unity*.

2. METODE

2.1 Studi Literatur

Studi literatur adalah sekumpulan kegiatan terkait dengan metode pengumpulan data perpustakaan, membaca dan Menyimpan dan mengelola bahan penelitian(Judithia, 2019). Pada tahap studi litertur penelitian ini akan melakukan eksplorasi dan meneliti teori-teori yang berkaitan dengan penelitian sebelumnya atau penelitian terdahulu yang sesuai dan berhubungan dengan penelitian yang sedang dilakukan sekarang. Untuk kegiatan yang dilakukan pada tahap studi literatur ini adalah megumpulkan berbagai macam sumber-sumber terpercacaya seperti jurnal, website resmi, dan juga skripsi yang terkait dengan topik penelitian ini.

2.2 *SDLC Waterfall*

Pada penelitian ini akan membutuhkan sebuah kerangka pengerjaan yang sistematis agar pembuatan *database* ini dapat diselesaikan denngan hasil terbaik dan tepat waktu. Metode SDLC *(System Development Life Cycle)* jenis *Waterfall* merupakan jenis SDLC *(System Development Life Cycle)* yang memiliki pendektatan alur hidup perangkat lunak secara urut atau bisa disebut sekuensial atau terurut dengan dimulai dari tahap analis, desain, implementasi, dan pengujian(Santoso, 2021). Pada penelitian kali ini akan menggunakan tahap analis permasalahan dan kebutuhan sistem, Desain sistem data, Implementasi, dan Pengujian.

a. Analis

   Proses pada pengumpulan keperluan dan kebutuhan yang akan dilakukan untuk menentukan spesifikasi dari rekayasa perangkat lunak agar dapat diengerti pada saat perancangan sistem.

b. Desain
Desain *Software* adalah proses yang memiliki fokus pada desain seluruh bagian perancangan sebuah sistem.
c. Implementasi
Implementasi dilakukan dengan translasi desain kedalam *software*. Hasil dari Implementasi adalah sistem yang sesuai dengan desain dan tahapannya.
d. Pengujian
Pengujian akan berfokus pada *software* dari kecocokan sistem dan fungsional, serta memastikan semua bagian telaj melalui pengujian.

3. HASIL DAN PEMBAHASAN

3.1 Hasil

Dari hasil yang didapat dari analisa menggunakan metode Black Box penelitian ini, *Realtime Database Firebase* diharapkan bisa berjalan sesuai dengan yang diharapkan. Hasil dari perancangan yang telah dibuat berdasarkan analisa dan desain sistem adalah sebagai berikut:

a. Proses pembuatan dan konfigurasi *Realtime Database Firebase* dapat dilakukan.
b. *Endpoint* dapat dibuat dan dapat diakses oleh *Unity* dan IoT (*Internet of Things*).
c. IoT (*Internet of Things*) dapat mengirim dan mengambil data dari *Realtime Database Firebase*.
d. *Unity* dapat mengirim dan mengambil data dari *Realtime Database Firebase*.
e. *Firebase* dapat menerima data log harian dan log per jam dari IoT (*Internet of Things*).

3.2 Pembahasan

Database yang menggunakan *Realtime Database* ini dirancang sesuai dengan kebutuhan IoT (*Internet of Things*) dan *Unity*. Pada uji coba ini memiliki tujuan untuk menambah inovasi pemakaian *database* pada IoT (*Internet of Things*) dan *Unity* secara bersamaan untuk pertukaran data. Pada bagian pembahasan ini akan membahas juga yang terkait pada *Realtime Database Firebase* saat sedang diakses oleh IoT (*Internet of Things*) dan *Unity*. Berikut tahapan yang diperlukan, yaitu Analis permasalahan, Analis kebutuhan sistem, Desain *database*, Perancangan *Realtime Database Firebase*, dan Pengujian.

3.3 Perancangan *Database*

3.3.1 Analis permasalahan

Pada tahap ini akan melakukan analisa tentang kekurangan data apa yang dapat digunakan untuk bahan uji coba dan dapat diselesaikan *Realtime Database Firebase* nantinya.

3.3.2 Analis kebutuhan sistem

Pada tahap Analis kebutuhan sistem akan memahammi kebutuhan sistem apa yang diperlukan dengan mengacu pada kekurangan di tahap Analis permasalahan. Kebutuhan yang ditentukan adalah kebutuhan *Software* dan *Hardware*.

3.3.3 Desain *database*

Pada tahap ini akan melakukan Desain awal data apa saja yang akan dibuat mengikuti dari analis kebutuhan sistem pada *software*. Data yang sudah ditentukan akan dinamai, ditenkukan tipe data, dan struktur datanya.

3.3.4 Perancangan *Realtime Database Firebase*

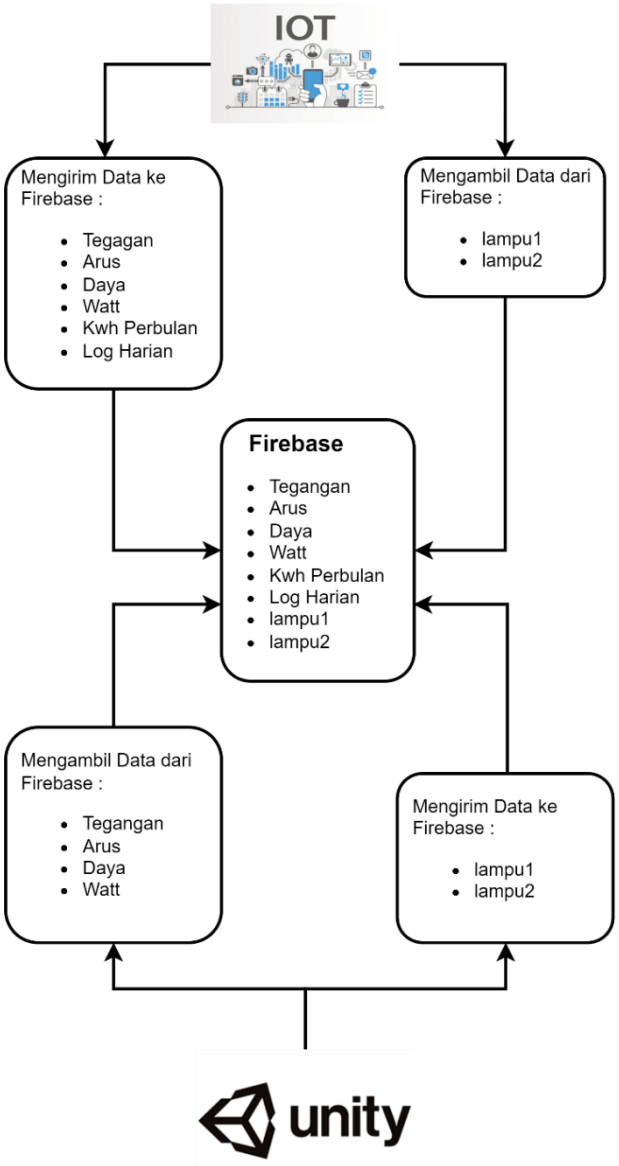
Pada tahap perancangan *Realtime Database Firebase* akan melakukan realisasi dari Desain *database* sebelumnya. Konfigurasi *Realtime Database Firebase* agar bisa diakses oleh IoT (*Internet of Things*) dan *Unity* juga akan dilakukan pada tahap ini, sekaligus membuat Endpoint untuk akses masing masing data.

3.3.5 Pengujian

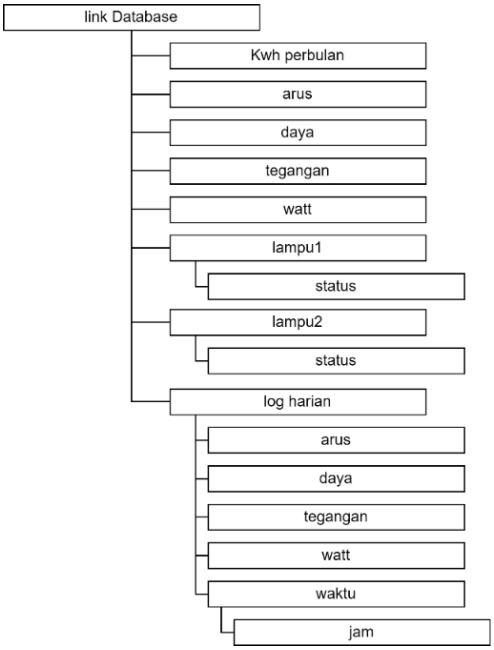
Pada tahap pengujian dilakukan dengan menggunakan metode *Black Box* dengan mencoba akses mengirim dan mengambil data dari *Realtime Database Firebase* yang akan dilakukan oleh IoT (*Internet of Things*) dan *Unity*.

3.4 Desain Sistem

Rancangan awal sistem yang akan digunakan untuk membuat *database* adalah Diagram Alir data IoT (*Internet of Things*), Diagram Alir data *Unity*, Struktur Data *Realtime Database Firebase*, Tabel Tipe Data.



Gambar 1. Diagram Alir Data



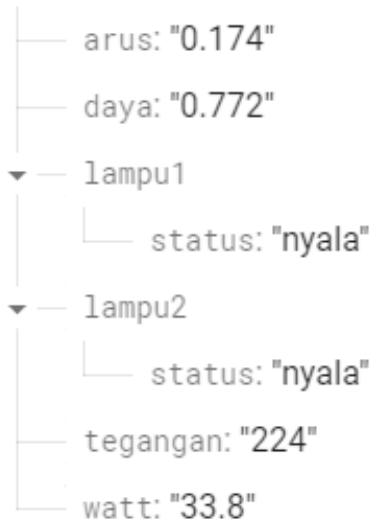
Gambar 2. Struktur Data *Realtime Database Firebase*

Tabel 1. Tabel Tipe Data *Realtime Database Firebase*

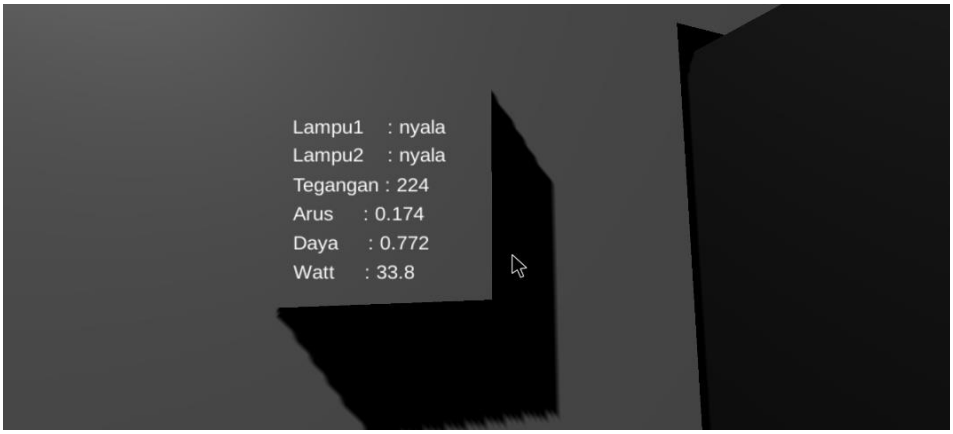
Child	Tipe Data
Kwh perbulan	*String*
arus	*String*
daya	*String*
tegangan	*String*
watt	*String*
lampu1	status:*String*
lampu2	status: *String*
Log hari	Arus: *String* daya: *String* tegangan: *String* watt: *String* waktu: *String*

3.5 Implementasi

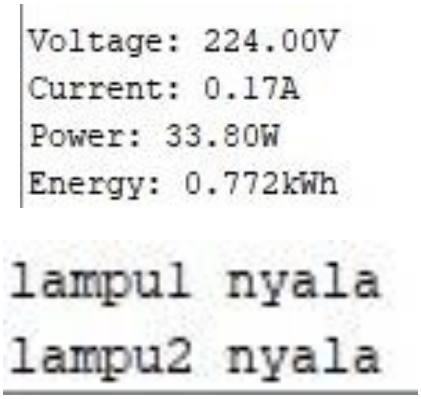
Implementasi yang akan dilakukan adalah dengan membuat penyimpanan data dengan menyesuaikan desain Struktur data *Realtime Database Firebase* dengan Tipe Data yang ada pada sub bab sebelumnya. Dan berikut adalah hasil dari perancangan *Realtime Database Firebase* beserta hasilnya setelah diuji dengan IoT (*Internet of Things*) dan *Unity* yang mengirim dan mengambil data.



Gambar 3. Data *Realtime Database Firebase*

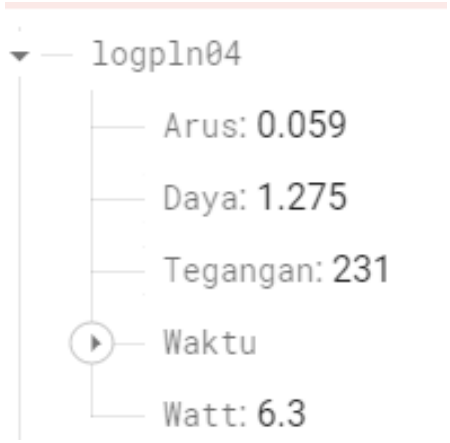


Gambar 4. *Unity* Mengakses *Realtime Database Firebase*



Gambar 5. IoT Mengakses *Realtime Database Firebase*

Gambar 6. Log Harian yang Dikirim Dari Iot



Gambar 7. Isi Log Harian Di *Firebase*

Gambar 8. Isi Data Waktu Pada Log Harian

Pada pengujian kali ini menggunakan metode *black box* yang dimana data kegiatan pengujian akan dijadikan sebuah tabel agar mempermudah pendataan kegiatan,kolom yang akan dipakai ada 4 kolom, mulai dari Aktivitas Pengujian, Realisasi Yang Diharapkan, Hasil Pengujian, dan Kesimpulan. Hasil pengujian *Black Box* bisa dilihat pada Tabel.

Tabel 2. Hasil Pengujian *Black Box*

Aktivitas Pengujian	Hasil Pengujian	Kesimpulan
Unity mengirim data lampu1 nyala dan lampu2 nyala.	IoT dapat melakukan akses mengambil data melalui Endpoint: 1. lampu1: nyala 2. lampu2: nyala	Berhasil
Unity mengirim data lampu1 nyala dan lampu2 mati.	IoT dapat melakukan akses mengambil data melalui Endpoint: 1. lampu1: nyala 2. lampu2: mati	Berhasil
Unity mengirim data lampu1 mati dan lampu2 nyala.	IoT dapat melakukan akses mengambil data melalui Endpoint: 3. lampu1: mati 4. lampu2: nyala	Berhasil
Unity mengirim data lampu1 mati dan lampu2 mati.	IoT dapat melakukan akses mengambil data melalui Endpoint: 1. lampu1: mati 2. lampu2: mati	Berhasil
IoT mengirim data beban listrik lampu dengan spesifikasi : Daya : 4 W Tegangan : 240 V Arus :0.016 A	*Unity* dapat melakukan akses mengambil data melalui Endpoint: 1. arus: 0.027 2. daya: 0.634 3. tegangan: 196.7 4. watt: 2.4	Berhasil

Aktivitas Pengujian	Hasil Pengujian	Kesimpulan
IoT mengirim data beban listrik 2 lampu bersamaan dengan spesifikasi: **Lampu 1** Daya: 4 W Tegangan: 250 V Arus:0.016 A **Lampu 2** Daya: 5 W Tegangan: 250 V Arus:0.01 A	*Unity* dapat melakukan akses mengambil data gabungan 2 lampu melalui Endpoint: 1. arus: 0.06 2. daya: 0.1259 3. tegangan: 231 4. watt: 7.3	Berhasil
IoT mengirim data beban listrik 2 adaptor secara bersamaan dengan spesifikasi **Adaptor1** Daya: 8 W Tegangan: 240 V Arus: 0.5 A **Adaptor2** Daya: 27 W Tegangan: 240 V Arus:0.7 A	*Unity* dapat melakukan akses mengambil data gabungan 2 adaptor melalui Endpoint: 1. arus: 0.174 2. daya: 0.772 3. tegangan: 224 4. watt: 33.8	Berhasil
IoT menghitung total Kwh selama 4 jam dengan adaptor 27 Watt.	Firebase menerima total Kwh harian secara dinamis ± 0.1 Kwh.	Berhasil
IoT menghitung total Kwh selama 4 jam dengan adaptor 8 Watt.	Firebase menerima total Kwh harian secara dinamis ± 0.03 Kwh.	Berhasil
IoT menghitung total Kwh selama 3 jam dengan adaptor 8Watt dan 27 Watt.	Firebase menerima total Kwh harian secara dinamis ± 0.1 Kwh.	Berhasil
IoT menghitung total Kwh selama 3 jam dengan lampu 5Watt dan 4 Watt.	Firebase menerima total Kwh harian secara dinamis ± 0.02 Kwh.	Berhasil
IoT mengirim data total Kwh bulan februari mulai tanggal 1 sampai 3	Firebase menerima total Kwh perbulan dinamis ± 4.44 Kwh.	Berhasil
IoT mengirim data total Kwh bulan februari mulai tanggal 1 sampai 4	Firebase menerima total Kwh perbulan dinamis ± 5.68 Kwh.	Berhasil

Aktivitas Pengujian	Hasil Pengujian	Kesimpulan
IoT mengirim data total Kwh bulan februari mulai tanggal 1 sampai 7	Firebase menerima total Kwh perbulan dinamis ± 10.31 Kwh	Berhasil
IoT mengirim data Kwh perjam di satu hari selama 3 jam, disertai berbagai beban listrik.	Firebase menerima data perjam: 1. 12 = 1.216 Kwh 2. 13 = 1.223 Kwh 3. 14 = 1.258 Kwh	Berhasil
IoT mengirim data Kwh perjam di satu hari selama 5 jam, disertai berbagai beban listrik.	Firebase menerima data perjam: 1. 12 = 1.216 Kwh 2. 13 = 1.223 Kwh 3. 14 = 1.258 Kwh 4. 15 = 1.266 Kwh 5. 16 = 1.272 Kwh	Berhasil
IoT mengirim Log harian dengan beban listrik sebanyak 4 hari dari tanggal 1.	Firebase menerima data: 1. logpln1 2. logpln2 3. logpln3 4. logpln4	Berhasil
IoT mengirim Log harian dengan beban listrik sebanyak 7 hari dari tanggal 1.	Firebase menerima data: 1. logpln1 2. logpln2 3. logpln3 4. logpln4 5. logpln5 6. logpln6 7. logpln7	Berhasil

4. SIMPULAN DAN SARAN

Dengan meninjau dan melihat hasil dari penelitian ini dari bab sebelumnya dapat diambil beberapa kesimpulan seperti berikut:

1. *IoT (Internet of Things)* dan *Unity* bisa saling bertukar data melalui *Realtime Database Firebase* versi free, namun di beberapa kesempatan ada satu atau dua data yang delay untuk diakses.
2. Metode SDLC bisa digunakan untuk metode pembuatan *Realtime Database Firebase*, meninjau dari hasil penelitian ini, metode *SDLC (System Development Life Cycle)* jenis waterfall mempunyai kelebihan yaitu mempermudah proses pengerjaan, namun memiliki kekurangan yaitu pada bagian optimasi sesudah pengerjaan tidak masuk dalam metode tersebut.
3. Semua Endpoint *Realtime Database Firebase* yang sudah dalam kondisi terpasang pada *Unity* dan Arduino IDE dapat digunakan untuk mengakses data *Realtime Database Firebase* oleh *IoT (Internet of Things)* dan *Unity*, data yang diakses adalah tegangan, arus, daya, watt, lampu1, lampu2, dan log harian beserta jamnya.

Database yang dibuat sudah dapat berjalan dengan normal sesuai dengan tujuan dari penelitian ini. Dengan hasil penelitian ini, ada saran untuk pengembangan pada sistem *database* ini kedepannya meliputi sebagai berikut:

1. Menggunakan Firebase selain versi free untuk mengetahui kelebihan lainnya.
2. Menggunakan data yang lebih kompleks strukturnya.
3. Lakukan Optimasi pada *Realtime Database Firebase*.

4. Menambah sistem *IoT (Internet of Things)* maupun *Unity* untuk akses *Realtime Database Firebase*.

5. DAFTAR PUSTAKA

Adani, F., & Salsabil, S. (n.d.). *Internet Of Things: Sejarah Teknologi Dan Penerapannya*.

Judithia, D. (2019). Proses Adaptasi Ikatan Mahasiswa Fakfak Di Kota Bandung. *Journal of Chemical Information and Modeling*, *53*(9), 54–69. Retrieved from https://elibrary.unikom.ac.id/id/eprint/1558/

Pangestu, R. A., Purboyo, T. W., Siswo, A., Ansori, R., Telkom, U., & Kelereng, P. B. (2021). *Permainan Tradisional Balap Kelereng Berbasis Virtual Reality Menggunakan Algoritma Complementary Filter Traditional Marbles Racing Game Based on Virtual Reality Using*. *8*(5), 6583–6600.

Santoso, A. (2021). *Software Development Life Cycle (SDLC)*. Sekolah Tinggi Informasi NITT.