

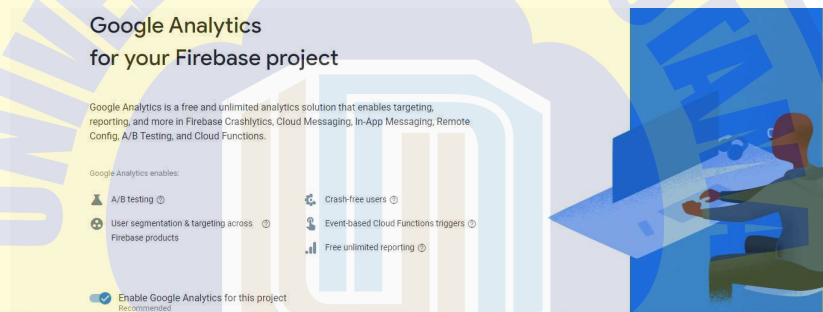
BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Firebase

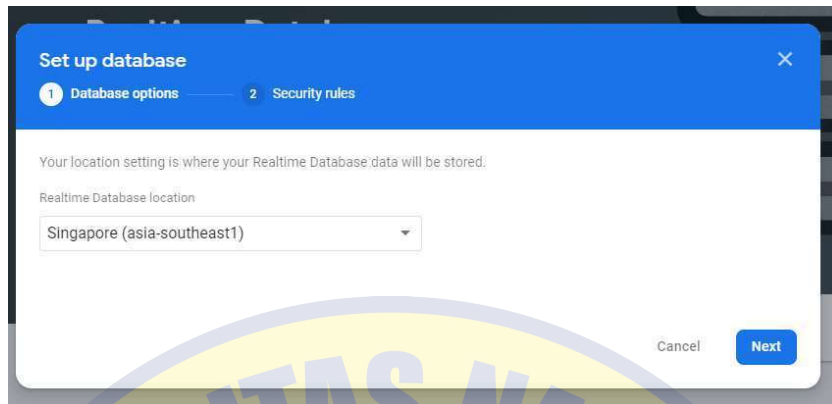
4.1.1 Perancangan dan Konfigurasi Database

Pada tahap perancangan dan konfigurasi Realtime Database Firebase akan menyesuaikan dengan kebutuhan akses data dari *IoT (Internet of Things)* dan Unity.



Gambar 4.1. Setting Google Analytics

Konfigurasi pertama adalah aktifkan Google Analytics Project pada Project Database Firebase untuk mempermudah pemakaian akses data, konfigurasi Google Analytics dapat dicek pada Gambar 4.1



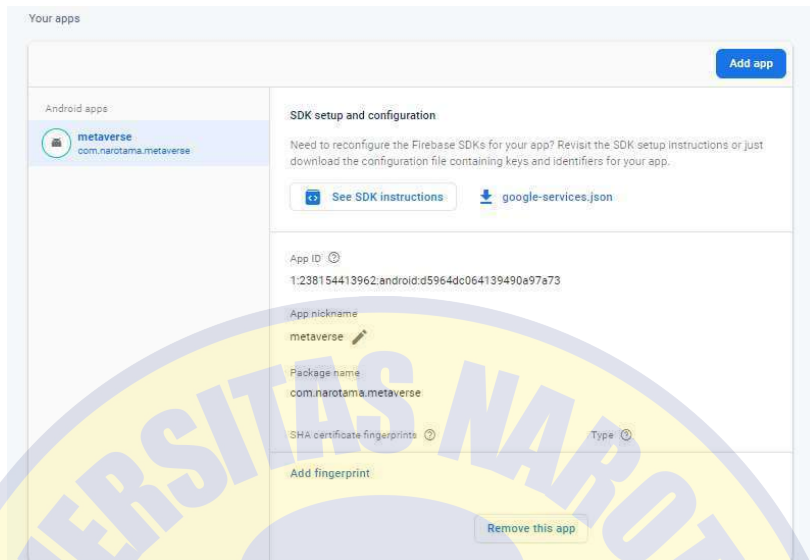
Gambar 4.2. Setting Lokasi Realtime Database

Setting lokasi database juga perlu dilakukan untuk waktu kasus secara real berdasarkan waktu Indonesia. Sesuai dengan Gambar 4.2 lokasi waktu Indonesia dapat mengikuti waktu Singapura.



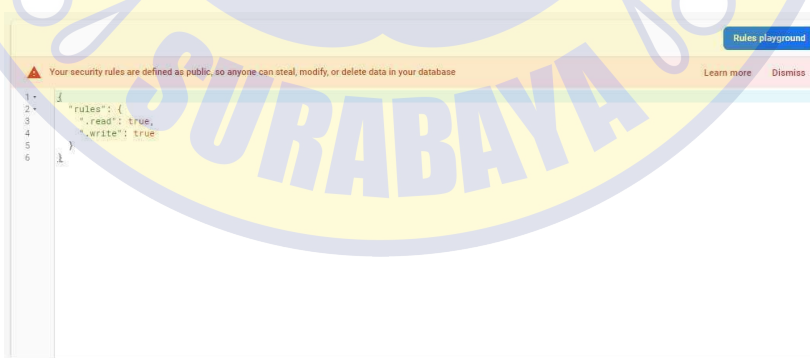
Gambar 4. 3. Settings Security Rules

Diawal Security Rules diatur dengan Locked mode sesuai dengan Gambar 4.3 untuk mempermudah mengubah rules read dan write menjadi true



Gambar 4. 4. Setting aplikasi yang akan menggunakan Firebase

Pada Gambar 4.4 Menjelaskan aplikasi yang dapat melakukan akses ke Firebase adalah aplikasi Android dengan nama aplikasi metaverse dan Package Name adalah com.narotama.metaverse dengan memasukkan SDK dari google-services.json.



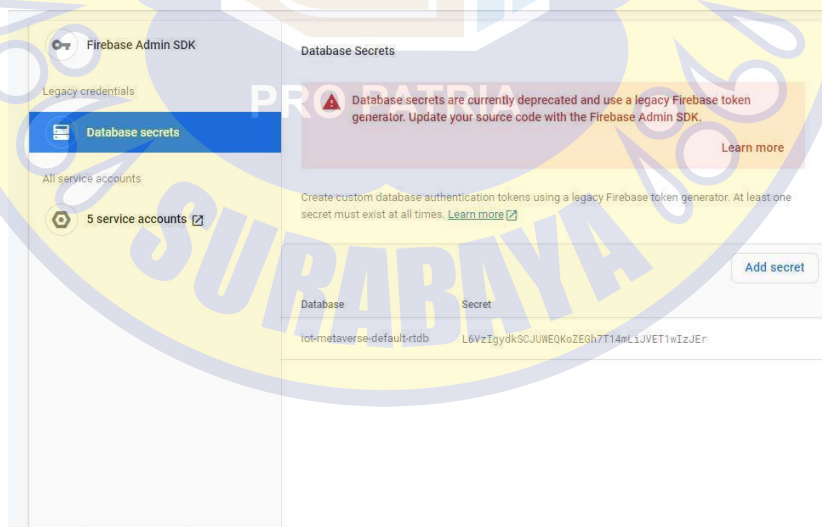
Gambar 4. 5. Setting Rules akses Realtime Database Firebase

Pada Gambar 4.5 Adalah perizinan akses dari aplikasi yang akan menggunakan Realtime Database Firebase. Penjelasan seperti pada tabel 4.1.

Tabel 4.1. Penjelasan Rules Realtime Database Firebase

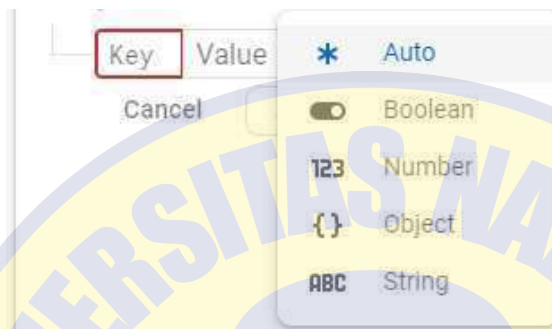
Rules	kondisi	Penjelasan
.read	true	Diizinkan untuk akses membaca data untuk semua aplikasi yang sudah diberi akses sebelumnya.
.write	True	Diizinkan untuk akses membaca membuat dan mengubah data untuk semua aplikasi yang sudah diberi akses sebelumnya.

Pada Gambar 4.6 adalah Database Secrets Realtime Database Firebase yang akan digunakan untuk token akses Realtime Database di Unity maupun *IoT (Internet of Things)*



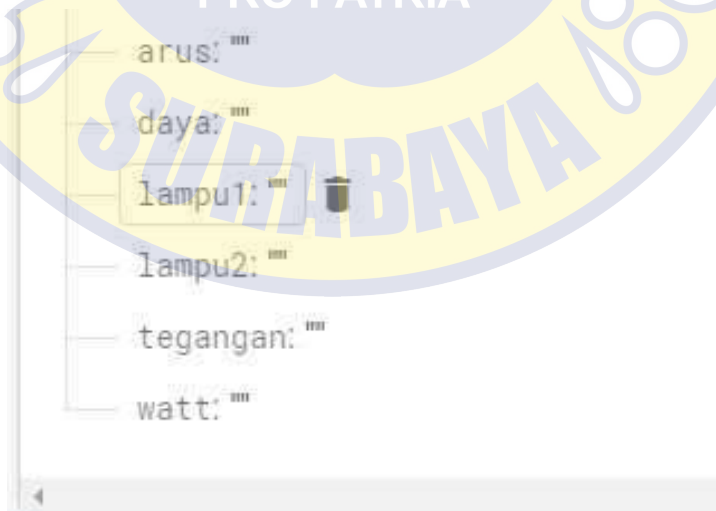
Gambar 4. 6. Database Secrets Realtime Database Firebase

Setelah Semua Konfigurasi selesai, perancangan database sudah bisa dilakukan. Perancangan database bisa dilakukan dengan membuat satu persatu data sesuai dengan kebutuhan *IoT (Internet of Things)* dan Unity sebelumnya.

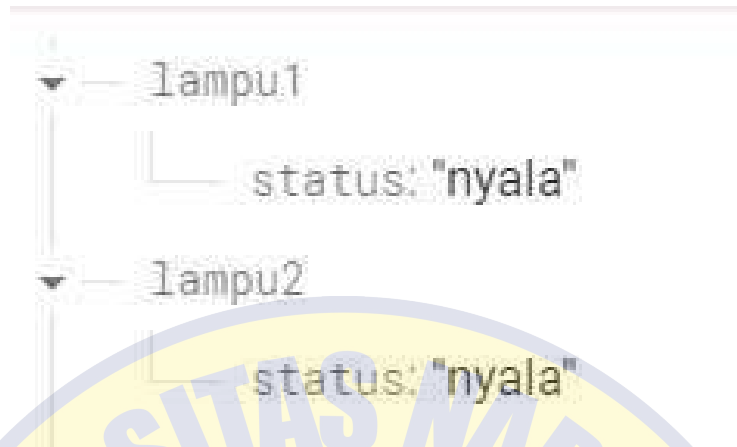


Gambar 4. 7. Pembuatan child Realtime Database Firebase

Pada Gambar 4.7 adalah bagian dari child yang dimana child berisikan Key sebagai nama data, Value sebagai data yang dimasukkan, dan Data Type merupakan tipe data yang akan disimpan.



Gambar 4. 8. Struktur data pada Realtime Database Firebase

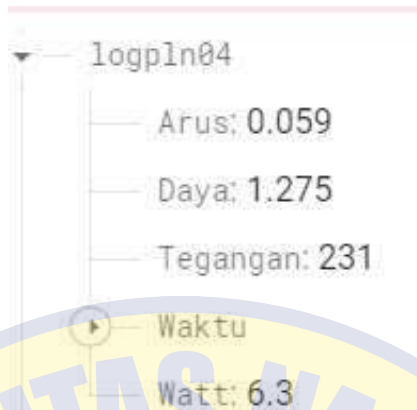


Gambar 4. 9. Struktur data pada lampu

Pada Gambar 4.8. adalah struktur data pada Realtime Database Firebase yang telah dibuat sesuai dengan kebutuhan *IoT (Internet of Things)* dan Unity pada sub bab 3.5.1. Kemudian di Gambar 4.9 merupakan child lampu1 dan lampu2 isinya adalah status yang dimana status berisi sebuah data bertipe string.

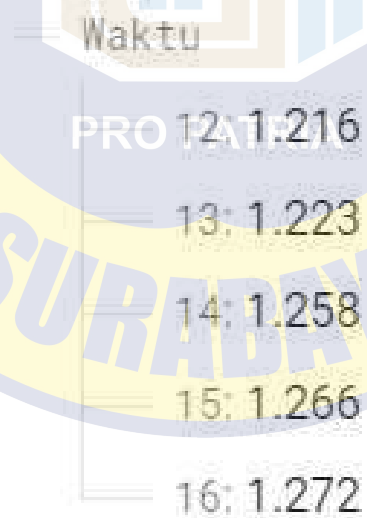


Gambar 4.10. Hasil pembuatan log harian selama 7 hari



Gambar 4.11. Data pada sebuah log harian

Pada Gambar 4.10 merupakan contoh data log harian selama 7 hari, yang dimana isi masing-masing log seperti pada Gambar 4.11 yaitu data Arus, Daya, Tegangan, Waktu, Watt. Pada data waktu di setiap log berisi data pemakaian Kwh pada setiap jam di hari itu. Contoh isi data Waktu bisa dilihat pada Gambar 4.11.



Gambar 4. 12. Data waktu log harian

Kemudian dari semua data log yang telah dibuat setiap hari, akan dihitung total daya atau Kwh perbulan dari pemakaian setiap harinya. Contoh seperti pada Gambar 4.13 adalah contoh total Kwh bulan Februari.



— Total Kwh Bulan Feb: 4.44

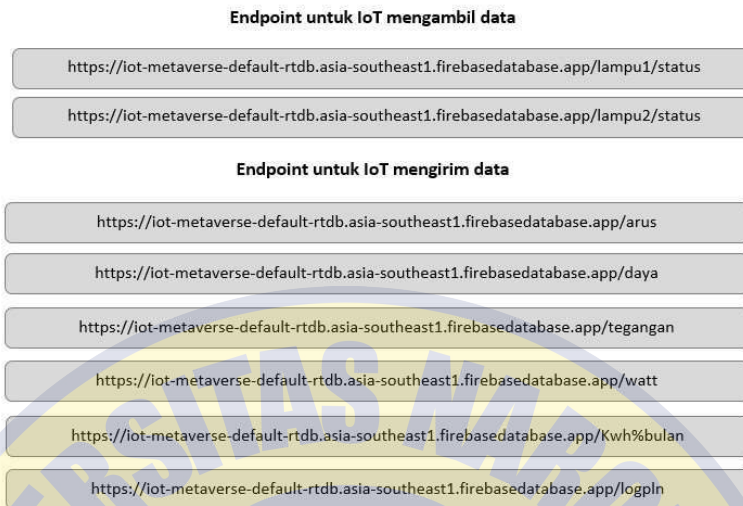
Gambar 4. 13.Total Kwh bulan Februari

4.1.2 Pembuatan End Point Realtime Database Firebase

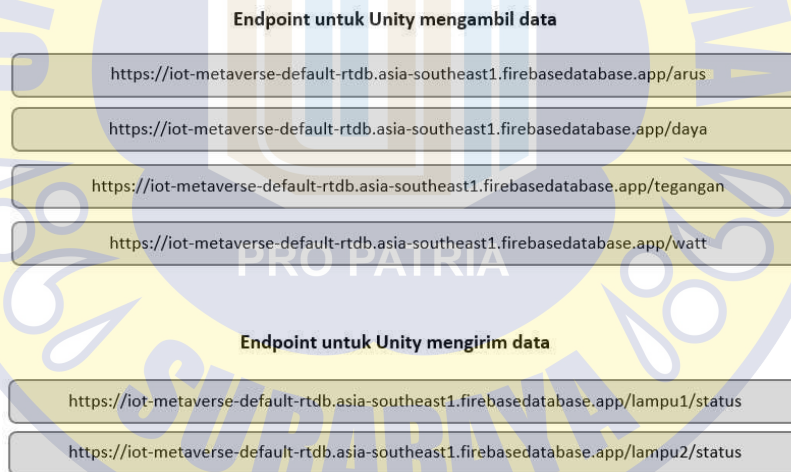
Pembubuatan Endpoint Realtime Database Firebase akan mengikuti struktur data, satu endpoint berupa link dengan lanjutannya adalah database yang akan di akses. Pada gambar 4.14 adalah struktur Endpoint untuk akses data pada Realtime Database Firebase.



Gambar 4. 14. Struktur Endpoint Realtime Database Firebase



Gambar 4. 15. Enpoint untuk *IoT (Internet of Things)*



Gambar 4. 16. Endpoint untuk Unity

Endpoint Realtime Database Firebase berbentuk seperti link pada umumnya, namun fungsi Endpoint dari Realtime Database Firebase adalah mengakses data dari Realtime Database Firebase tersebut dan hanya bisa diakses oleh aplikasi yang diberikan akses. Pada Gambar 4.15 terdapat semua Endpoint

yang diperlukan oleh *IoT (Internet of Things)* untuk akses mengambil data lampu1 dan lampu2. Kemudian untuk *IoT (Internet of Things)* mengirim data arus, daya, tegangan, dan watt. Pada Gambar 4.16 merupakan Endpoint untuk akses Unity mengambil data arus, daya, tegangan, dan watt. Kemudian untuk Unity mengirim data dapat menggunakan Endpoint lampu1 dan lampu2.

4.2 Pengujian Firebase

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. Objek yang diuji pada penelitian ini adalah 2 lampu dan 2 adaptor dengan spesifikasi yang ada pada Realisasi Yang Diharapkan, Hasil Pengujian, dan Kesimpulan. Hasil pengujian *Black Box* dapat dilihat di Tabel 4.2.

Tabel 4. 2. 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

<p>IoT mengirim data beban listrik lampu dengan spesifikasi :</p> <p>Daya : 4 W Tegangan : 240 V Arus :0.016 A</p>	<p>Unity dapat melakukan akses mengambil data melalui Endpoint:</p> <p>1. arus : 0.027 2. daya : 0.634 3. tegangan: 196.7 4. watt : 2.4</p>	Berhasil
<p>IoT mengirim data beban listrik 2 lampu bersamaan dengan spesifikasi :</p> <p>Lampu 1 Daya : 4 W Tegangan : 250 V Arus :0.016 A</p> <p>Lampu 2 Daya : 5 W Tegangan : 250 V Arus :0.01 A</p>	<p>Unity dapat melakukan akses mengambil data gabungan 2 lampu melalui Endpoint:</p> <p>1. arus : 0.06 2. daya : 0.1259 3. tegangan: 231 4. watt : 7.3</p>	Berhasil
<p>IoT mengirim data beban listrik 2 adaptor secara bersamaan dengan spesifikasi</p> <p>Adaptor1 Daya : 8 W Tegangan : 240 V Arus : 0.5 A</p> <p>Adaptor2 Daya : 27 W Tegangan : 240 V Arus :0.7 A</p>	<p>Unity dapat melakukan akses mengambil data gabungan 2 adaptor melalui Endpoint:</p> <p>1. arus : 0.174 2. daya : 0.772 3. tegangan: 224 4. watt : 33.8</p>	Berhasil
<p>IoT menghitung total Kwh selama 4 jam dengan adaptor 27 Watt.</p>	<p>Firestore menerima total Kwh harian secara dinamis ± 0.1 Kwh.</p>	Berhasil
<p>IoT menghitung total Kwh selama 4 jam dengan adaptor 8 Watt.</p>	<p>Firestore menerima total Kwh harian secara dinamis ± 0.03 Kwh.</p>	Berhasil
<p>IoT menghitung total Kwh selama 3 jam dengan adaptor 8 Watt dan 27 Watt.</p>	<p>Firestore menerima total Kwh harian secara dinamis ± 0.1 Kwh.</p>	Berhasil

IoT menghitung total Kwh selama 3 jam dengan lampu 5 Watt dan 4 Watt.	Firestore menerima total Kwh harian secara dinamis ± 0.02 Kwh.	Berhasil
IoT mengirim data total Kwh bulan februari mulai tanggal 1 sampai 3	Firestore menerima total Kwh perbulan dinamis ± 4.44 Kwh.	Berhasil
IoT mengirim data total Kwh bulan februari mulai tanggal 1 sampai 4	Firestore menerima total Kwh perbulan dinamis ± 5.68 Kwh.	Berhasil
IoT mengirim data total Kwh bulan februari mulai tanggal 1 sampai 7	Firestore menerima total Kwh perbulan dinamis ± 10.31 Kwh	Berhasil
IoT mengirim data Kwh perjam di satu hari selama 2 jam, disertai berbagai beban listrik.	Firestore menerima data perjam : 1. 12 = 1.216 Kwh 2. 13 = 1.223 Kwh	Berhasil
IoT mengirim data Kwh perjam di satu hari selama 3 jam, disertai berbagai beban listrik.	Firestore 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.	Firestore 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 3 hari dari tanggal 1.	Firestore menerima data : 1. logpln1 2. logpln2 3. logpln3	Berhasil
IoT mengirim Log harian dengan beban listrik sebanyak 4 hari dari tanggal 1.	Firestore menerima data : 1. logpln1 2. logpln2 3. logpln3 4. logpln4	Berhasil
IoT mengirim Log harian dengan beban listrik sebanyak 7 hari dari tanggal 1.	Firestore menerima data : 1. logpln1 2. logpln2 3. logpln3 4. logpln4 5. logpln5 6. logpln6 7. logpln7	Berhasil