

Implementasi Aplikasi *Styins Home* pada *Smart Home Security* Menggunakan *Real-Time Database Firebase*

Agni Andhini¹, Ibrahim², Yuliarman Saragih³

Program Studi Teknik Elektro, Universitas Singaperbangsa Karawang¹

Program Studi Teknik Elektro, Universitas Singaperbangsa Karawang²

Program Studi Teknik Elektro, Universitas Singaperbangsa Karawang³

*agnianndhini16006@student.unsika.ac.id¹, ibra.lammada@gmail.com², yuliarman@gmail.com³

ABSTRACT

The IoT-based Styins Home (Security System Interface Smart Home) application system has been implemented. This study aims to build a control system and monitoring of security conditions of remote home devices in real-time so as to prevent losses due to negligence in maintaining home security conditions. Application features include camera streaming, smart light, smart fan and detection gas along with notification alerts. This application utilizes the Firebase real-time database as a database management service and Android Studio software. The research method in designing application systems uses the waterfall model. The data from the test results of the delay value according to the TIPHON standard include very good quality delay categories, namely 7.1 ms on 240 kbps camera streaming and moderate category delay quality, namely 425.08 ms on smart light, smart fan and gas detection on firebase 32 kbps while the result data testing the percentage value of the highest throughput percentage with medium network quality, namely 40.80% on 240 kbps camera streaming, while other throughput percentage values have poor network quality.

Keywords: *Delay, Real-Time Database, Styins Home, Throughput*

INTISARI

Telah dilakukan implementasi sistem aplikasi *Styins Home (Security System Interface Smart Home)* berbasis IoT. Penelitian ini bertujuan membangun sistem kontrol dan *monitoring* kondisi keamanan perangkat rumah jarak jauh secara *real-time* sehingga mencegah kerugian akibat kelalaian dalam penjagaan kondisi keamanan rumah. Fitur aplikasi meliputi *streaming* kamera, *smart light*, *smart fan* dan *detection gas* dilengkapi peringatan notifikasi. Aplikasi ini memanfaatkan *real-time database firebase* sebagai layanan pengelola *database* dan *software android studio*. Metode penelitian dalam perancangan sistem aplikasi menggunakan model *waterfall*. Data hasil pengujian nilai *delay* menurut standarisasi TIPHON termasuk kualitas *delay* kategori sangat bagus yaitu 7,1 ms pada *streaming* kamera 240 kbps dan kualitas *delay* kategori sedang yaitu 425,08 ms pada *smart light*, *smart fan* dan *detection gas* di *firebase* 32 kbps sedangkan data hasil pengujian nilai persentasi *throughput* tertinggi dengan kualitas jaringan sedang yaitu 40,80% pada *streaming* kamera 240 kbps sedangkan nilai persentasi *throughput* lainnya memiliki kualitas jaringan yang buruk.

Kata kunci: *Delay, Real-Time Database, Styins Home, Throughput*

I. PENDAHULUAN

Peralatan rumah pintar merupakan kebutuhan sekunder yang membuat aktivitas manusia menjadi lebih praktis dan juga efektif. Disamping itu seiring dengan kelalaian yang dilakukan oleh pemilik rumah dalam menjaga keamanan sehingga menyebabkan kerugian serta pemborosan dalam penggunaan peralatan listrik yang berlebihan dan tidak terkontrol.

Dengan adanya perkembangan ilmu teknologi dan pengetahuan, tercipta sebuah aplikasi rumah pintar (*smart home*) yaitu berfungsi untuk memantau ataupun mengontrol dari segi aspek keamanan rumah dan perangkat rumah dengan jarak jauh menggunakan *smartphone*.

Penelitian sebelumnya dilakukan oleh Atep Septian, dkk. (2019) dengan judul penelitian sebelumnya kontrol lampu secara *wireless* berbasis

Arduino Uno kontrol lampu menggunakan MYSQL sebagai database dan aplikasi berbasis web *platform blink* [1]. Mustaqim, Abi Sabila, Kurnianto, dkk (2020) dengan judul penelitian implementasi teknologi *internet of things* pada sistem pemantauan kebocoran gas LPG dan kebakaran menggunakan *database* pada *google firebase* [2]. kelebihan dari penelitian menggunakan *firebase* sebagai layanan *backend*, kekurangan dari 2 referensi ini belum adanya sistem keamanan dan pengontrol perangkat rumah untuk mengatasi masalah pemilik rumah secara menyeluruh seperti mengetahui kondisi keamanan sekitar rumah, dapat mengontrol beberapa lampu, meninggalkan ruangan terlalu lama dengan keadaan kipas menyala ataupun pemilik rumah lupa untuk melepaskan regulator gas sehingga dapat berpotensi kebakaran rumah memanfaatkan android studio sebagai *platform* membangun aplikasi [3].

Untuk mengatasi permasalahan berdasarkan kekurangan dari 2 referensi yang telah disebutkan diatas, dibuatlah sebuah rancangan sistem berupa sebuah aplikasi yang dapat mengetahui kondisi keamanan dan kontrol perangkat rumah setiap waktu dengan jarak yang jauh melalui *smartphone* menggunakan konsep *Internet of Things* (IoT) [4]. Layanan konsep IoT yang saya gunakan untuk pembuatan aplikasi ini adalah *firebase*. Produk aplikasi ini dinamakan aplikasi *styins home*. *Firebase* adalah sebuah layanan infrastruktur *backend-as-a-service* (BaaS) yang diakuisisi oleh google pada oktober 2014 silam [5]. *Firebase* pada aplikasi ini berfungsi menyatukan 3 data yang dikirimkan oleh mikrokontroler yang berbeda dan mengolahnya agar dapat memonitoring alat elektronik secara *real-time* dengan memanfaatkan *real-time database firebase* [6].

Real-time database firebase dapat mengimplementasi seperti kontrol lampu, kontrol *smartfan* dan memantau keadaan suhu dan kelembaban ruangan sehingga dapat mengatur secara otomatis kecepatan putaran *smart fan* dan implementasi *monitoring* kadar gas dalam ruangan dapur sehingga berfungsi untuk mengaktifkan *exhaust fan* ketika telah melewati pengaturan kadar >200 untuk asap dan >300 untuk gas. Sedangkan pada kamera mengakses data tidak melalui *firebase* tetapi melalui akses data berupa IP yang dimiliki oleh ESP32-Cam berfungsi menampilkan rekaman siaran langsung melalui aplikasi. Hal ini berfungsi untuk membuat efisien

dalam pemanfaatan waktu, keamanan dan kenyamanan pemilik rumah karena dapat memantau dan mengontrol keamanan dan perangkat rumah apabila sedang meninggalkan rumah dengan waktu yang lama menggunakan 1 aplikasi.

II. LANDASAN TEORI

A. Bandwidth

Bandwidth merupakan kapasitas luas maksimal cakupan suatu jaringan untuk melakukan proses pengiriman dan penerimaan data dari sebuah titik ke titik lain dalam jangka waktu tertentu yang dihitung dalam satuan waktu *bit per second* (bps) [7]. Besarnya alokasi *bandwidth* sangat berpengaruh untuk menentukan *bandwidth* mana yang lebih kuat seperti penggunaan antena *multiband* pada sebuah sistem *radio frequensi* (RF) dimaksudkan untuk meningkatkan efisiensi sistem tersebut secara ekonomi yang dapat memberikan efek peningkatan kapasitas dengan *coverage* yang lebih luas [8].

B. Real-time database

Real-time database merupakan sebuah layanan *database* NoSQL sehingga aplikasi ini dapat digunakan oleh banyak *client* dengan data diterima secara *real-time*. Penggunaan *fitur real-time database* ini membangun sebuah sistem bersifat *responsive* dengan didukung API *real-time database* yang dirancang mampu menjalankan operasi sistem dengan cepat [6],[9].

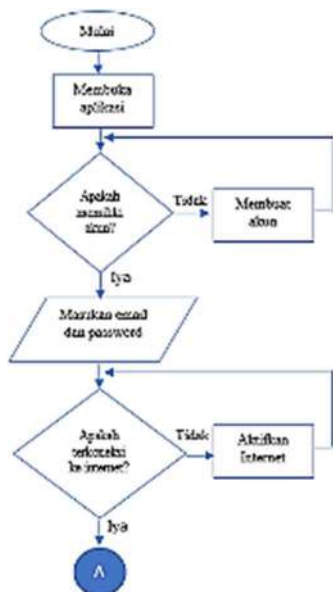
C. NodeMCU ESP8266

NodeMCU ESP8266 merupakan sebuah modul *wifi* perangkat tambahan mikrokontroler berfungsi terkoneksi TCP (*Transmission Control Protokol*) secara langsung [10]. Mikrokontroler adalah suatu komponen elektronika digital yang memiliki masukan, keluaran serta kendali, cara kerja mikrokontroler sebenarnya membaca dan menulis data [11].

III. METODE PENELITIAN

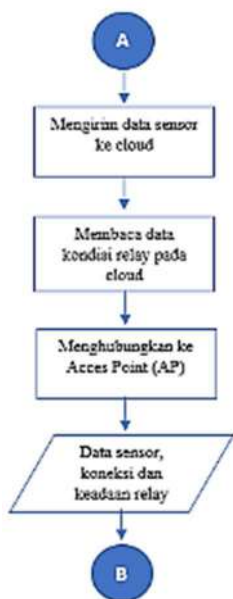
Penelitian perancangan sistem aplikasi *styins home* dilakukan beberapa tahapan, seperti: implementasi perancangan sistem *styins home*, perhitungan *delay* dan perhitungan *throughput*.

A. Perancangan Sistem Keseluruhan



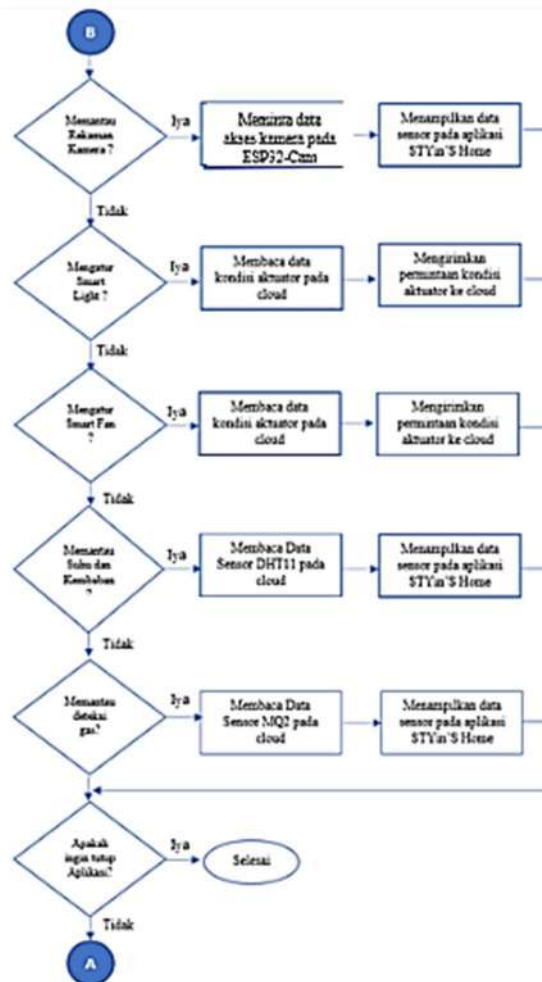
Gambar 1. Flowchart Login Aplikasi

Berdasarkan Gambar 1 dijelaskan bahwa pada saat memasuki aplikasi *styins home* akan dilakukan *authentication* akun dengan verifikasi email dan *password*. Jika pengguna belum memiliki akun disarankan untuk segera melakukan registrasi akun. Setelah memiliki akun, pengguna harus memastikan *smartphone* terkoneksi dengan jaringan internet yang berfungsi untuk menghubungkan perangkat lunak dengan perangkat keras.



Gambar 2. Flowchart Komunikasi Client dan Cloud

Berdasarkan Gambar 2, setelah aplikasi telah terkoneksi ke internet maka keempat NodeMCU *client* akan mengirimkan data sensor ke *cloud*. Pada sistem ini kami menggunakan *real-time database* sebagai tempat penyimpanan *database* di *firebase*.



Gambar 3. Flowchart Aplikasi Styin's Home dengan Komunikasi Cloud dan Nodemcu Client

Berdasarkan Gambar 3 masukkan berupa data sensor, koneksi dan keadaan *relay*, selanjutnya pada sistem *styins home* memiliki 4 fitur utama yaitu rekaman kamera, *smart light*, *smart fan* dan deteksi gas. Dimana fungsi *streaming* kamera pada ESP32-cam secara otomatis meminta akses data untuk memperoleh *output* menampilkan hasil *streaming* kamera sedangkan memantau *smart light*, suhu dan kelembaban pada *smart fan* dan deteksi gas kemudian aplikasi akan membaca data sensor yang ada pada

database cloud untuk memperoleh output menampilkan data sensor pada aplikasi.

B. Perhitungan Throughput

Perhitungan Throughput berfungsi untuk mengetahui nilai kecepatan jaringan pengiriman data yang aktual dengan jumlah paket data yang diterima oleh firebase dari nodeMCU client dan dari firebase ke aplikasi styins home sebagai tujuan info data dengan menggunakan durasi interval waktu selama < 2 menit. Biasanya throughput merupakan kemampuan sebenarnya suatu jaringan dalam melakukan pengiriman data [12] [13] [14]. Menghitung Throughput sebagai berikut:

$$Throughput = \frac{\text{Packet data yang diterima}}{\text{Lama pengamatan}}$$

Tabel 1. Nilai Standar Throughput (Sumber: TIPHON)

Kategori Throughput	Indeks	Throughput
Sangat Bagus	76% - 100%	4
Bagus	51% - 75%	3
Sedang	26% - 50%	2
Buruk	< 25%	1

C. Perhitungan Delay

Perhitungan Delay berfungsi untuk menentukan berapa lama waktu yang dibutuhkan oleh data NodeESP32-cam untuk sampai ke tempat tujuan data yaitu firebase dan IP 192.168.43.107 kemudian mengirimkan data ke aplikasi styins home menggunakan jaringan wifi yang dibatasi bandwidthnya [12] [13] [14]. Menghitung delay sebagai berikut:

$$\text{Rata-rata delay} = \frac{\text{Total Delay}}{\text{Packet Yang Dterima}}$$

Tabel 2. Nilai Standar Delay

Kategori Delay	Besar Delay	Indeks
Sangat Bagus	< 150 ms	4
Bagus	150 s/d 300 ms	3
Sedang	300 s/d 450 ms	2
Buruk	>450 ms	1

IV. HASIL PENELITIAN DAN PEMBAHASAN

Hasil implementasi aplikasi styins home dengan real-time database firebase. Membangun aplikasi styins home memanfaatkan firebase sebagai pengelola database dan software android studio sebagai platform untuk menjalankan aplikasi dalam smartphone.

A. Implementasi Tampilan Halaman Login



Gambar 4. Implementasi Tampilan Halaman Login



Gambar 5. Login Berhasil Membaca ID Authentication Di Firebase

Pada Gambar 4 bahwa halaman ini berfungsi untuk memastikan bahwa pengguna telah terdaftar pada sistem dan membaca kode ID di authentication firebase seperti pada Gambar 5. Pada tampilan halaman login berisikan email dan password sebagai sistem keamanan akun user.

B. Implementasi Tampilan Halaman Registrasi



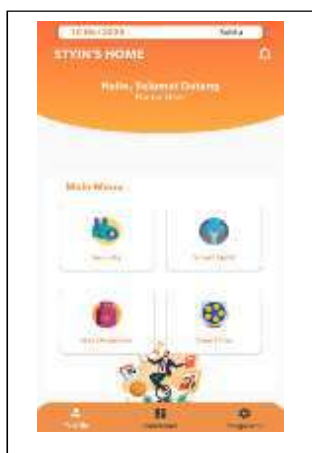
Gambar 6. Implementasi Halaman Registrasi



Gambar 7. Registrasi Berhasil di Authentication Firebase

Pada Gambar 6 merupakan sebuah halaman yang berfungsi untuk pengguna melakukan registrasi pendaftaran akun pengguna aplikasi *styins home* dan memberikan ID *authentication* sebagai pengenalan dalam memasuki aplikasi terlihat pada Gambar 7.

C. Implementasi Tampilan Halaman Dashboard



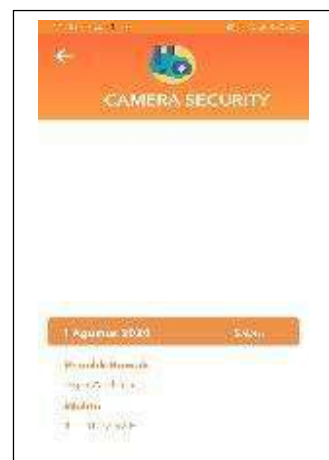
Gambar 8. Implementasi Halaman Dashboard



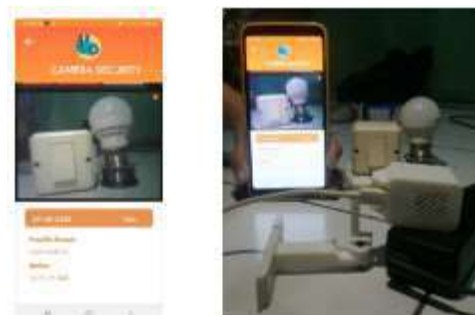
Gambar 9. Tampilan Real-time Database Firebase

Gambar 8 bahwa halaman ini akan ditampilkan oleh sistem jika *user* telah berhasil *login* kedalam sistem. Gambar 9 merupakan tampilan *real-time database firebase* yang akan membaca dan menerima pengiriman data dari NodeMCU. Halaman ini merupakan halaman utama yang menampilkan menu-menu yang dapat diakses oleh *user* berisikan menu *security* yang rekaman kamera, *smart light*, *smart fan* dan *gas detection*.

D. Implementasi Tampilan Halaman Menu Security



Gambar 10. Implementasi Halaman Menu Security



Gambar 11. Implementasi Menu security dan ESP32-Cam

Gambar 11 Menu *security* ini berfungsi melihat kondisi rumah secara siaran langsung dimana *user* dapat melihat melalui aplikasi *styins home*. Terlihat jelas pada Gambar 12 Menampilkan data *security* berupa hasil rekaman *streaming*.

E. Implementasi Tampilan Halaman Menu Smart Light



Gambar 12. Implementasi Halaman Menu *Smart Light*



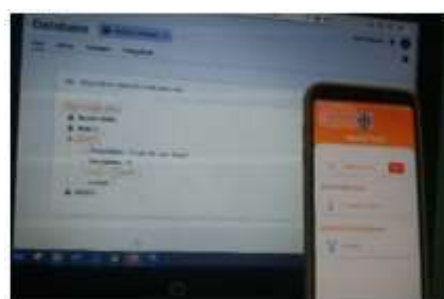
Gambar 13. Hasil Pengujian Fungsional *Smart Light* Menghidupkan Lampu 1

Berdasarkan Gambar 12 merupakan salah satu menu utama dalam aplikasi *styins home*. Pada halaman ini berisikan 4 *list* lampu yang berfungsi *user* dapat mengontrol lampu dengan cara menyalakan dan mematikan lampu dengan cara klik tombol pada lampu yang akan *user* atur dalam penggunaan seperti Gambar 13 dalam menghidupkan lampu depan.

F. Implementasi Tampilan Halaman Menu Smart Fan



Gambar 14. Implementasi Tampilan Halaman Menu *Smart Fan*



Gambar 15. Hasil Pengujian Fungsional *Smart Light* Saat Kecepatan Normal / Medium Berdasarkan Gambar 14 pada implementasi halaman menu *smart fan* menampilkan tombol menyalakan dan mematikan *smart fan* menampilkan tampilan kadar suhu dalam satuan celcius sedangkan tampilan kadar kelembaban menampilkan dalam satuan RH terlihat juga pada Gambar 15 menghasilkan suhu rentang >23 s/d 28 sehingga menghasilkan kecepatan normal/ medium.

G. Implementasi Tampilan Halaman Menu Detection Gas



Gambar 16. Implementasi Tampilan Halaman Menu *Detection Gas* Saat Keadaan Aman



Gambar 17. Hasil Pengujian *Fungsional Detection Gas* Saat Keadaan Aman



Gambar 18. Implementasi Tampilan Halaman Menu *Detection Gas* Saat Keadaan Bahaya



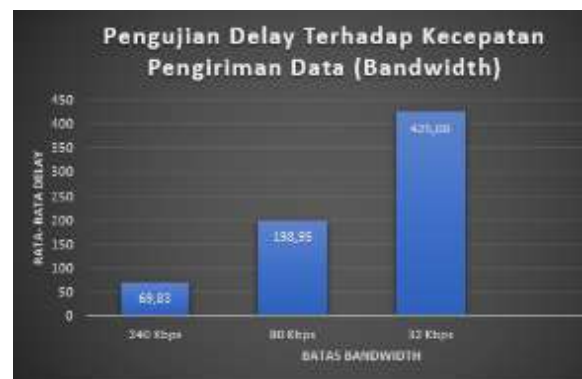
Gambar 19. Hasil Pengujian *Fungsional Detection Gas* Saat Keadaan Bahaya

Berdasarkan Gambar 18 dan Gambar 19 pada implementasi halaman menu *detection gas* menampilkan tampilan kadar asap yang terdeteksi sebelumnya pada NodeMCU dalam keadaan aman sedangkan berdasarkan Gambar 18 dan Gambar 19 menampilkan kadar gas asap dalam keadaan bahaya karena dalam pengaturan pada NodeMCU Ketika kadar gas >300 PPM dan kadar asap >200 PPM maka *buzzer* dan *exhaust fan* akan menyala serta diberikan notifikasi melalui aplikasi *styins home*.

Selain implementasi aplikasi *styins home* di *real-time database firebase*, terdapat pengujian *throughput* dan pengujian *delay* untuk mengetahui kualitas layanan aplikasi *styins home* dalam jaringan internet sesuai dengan fungsinya.

Tabel 3. Data *Delay Smart light, Smart Fan dan Detection Gas Di Firebase*

Pengujian ke	Waktu		Alokasi Bandwidth (kbps)	Total Delay (s)	Packet yang diterima	Delay (ms)
	Tanggal	Jam				
1	23/ 07/ 2020	14:50 s/d 14:52	240	104,142	1509	69,01
2	24/ 07/ 2020	10:55 s/d 10:57	240	104,791	1575	66,53
3	24/ 07/ 2020	18:44 s/d 18:46	240	112,912	1608	70,21
4	28/ 07/ 2020	16:16 s/d 16:18	240	112,456	1528	73,59
Rata – rata delay (ms)						69,83
5	24/07/ 2020	18:30 s/d 18:32	80	112,760	528	213,56
6	25/ 07/ 2020	02:39 s/d 02:41	80	108,747	639	170,18
7	25/ 07/ 2020	03:39 s/d 03:41	80	106,352	484	219,73
8	28/ 07/ 2020	22:08 s/d 22:10	80	113,872	592	192,35
Rata – rata delay (ms)						198,95
9	23/ 07/ 2020	23:41 s/d 23:43	32	111,126	234	474,89
10	24/ 07/ 2020	22:55 s/d 22:57	32	111,980	294	380,88
11	25/ 07/ 2020	04:49 s/d 04:51	32	118,045	299	394,79
12	28/ 07/ 2020	22:33 s/d 22:35	32	103,903	231	449,79
Rata – rata delay (ms)						425,08
Nilai delay tertinggi						425,08



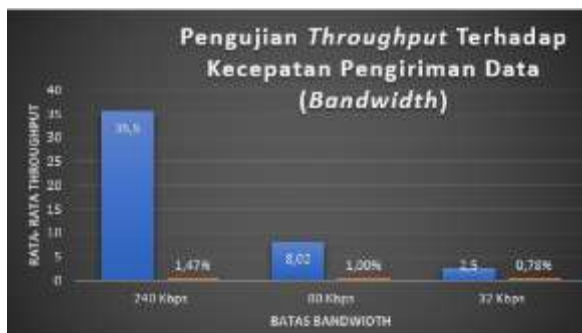
Gambar 20. Grafik Pengujian *Delay Data Smart light, Smart Fan dan Detection Gas*

Berdasarkan data dari Tabel 3 dan Gambar 20 disimpulkan bahwa semakin nilai *delay* mengalami penurunan maka semakin besar pula kecepatan pengiriman data jaringan. Kualitas jaringan pada nilai *delay* tertinggi untuk *smart light, smart fan dan detection gas* di *firebase* yaitu 425,08 ms memiliki kategori sedang dengan kecepatan pengiriman data 32 kbps dan nilai *delay* terendah yaitu 69,83 ms kategori

sangat bagus dengan kecepatan pengiriman data 240 kbps.

Tabel 4. Data Pengujian *Throughput* Dalam Penggunaan *Smart Light*, *Smart Fan* dan *Detection Gas* Di *Firestore*

Pengujian ke	Waktu		Alokasi Bandwidth (kbps)	Paket data yang diterima (bytes)	Lama Pengiriman (s)	Throughput (Kbps)	% Throughput
	Tanggal	Jam					
1	23/07/2020	14:50 s.d 14:52	240	552533	104,142	42,44	1,47
2	24/07/2020	10:55 s.d 10:57	240	440984	104,791	33,66	
3	24/07/2020	18:44 s.d 18:46	240	495277	112,912	35,09	
4	28/07/2020	16:16 s.d 16:18	240	433605	112,456	30,84	
Rata - rata throughput (kbps)						35,50	
5	24/07/2020	18:30 s.d 18:32	80	111875	112,760	7,937	1,00
6	25/07/2020	02:39 s.d 02:41	80	137059	108,747	10,092	
7	25/07/2020	03:39 s.d 03:41	80	98916	106,352	7,440	
8	28/07/2020	22:08 s.d 22:10	80	94571	113,872	6,844	
Rata - rata throughput (kbps)						8,025	
9	23/07/2020	23:41 s.d 23:43	32	23528	114,126	1,649	0,78
10	24/07/2020	22:55 s.d 22:57	32	41520	111,980	2,966	
11	25/07/2020	04:49 s.d 04:51	32	54056	118,045	3,663	
12	28/07/2020	22:33 s.d 22:35	32	22692	103,903	1,747	
Rata - rata throughput (kbps)						2,506	
Nilai throughput tertinggi						35,50	1,47
Nilai throughput terendah						2,506	0,78

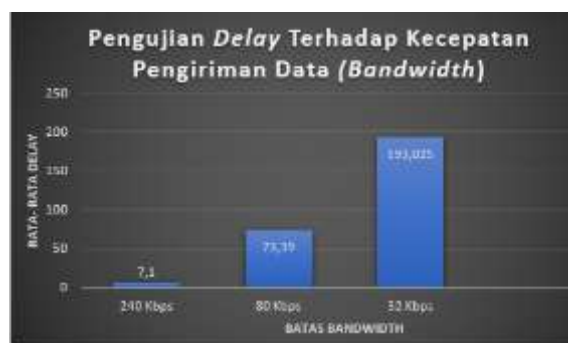


Gambar 21. Grafik Pengujian *Throughput* Data *Smart light*, *Smart Fan* dan *Detection Gas*

Berdasarkan data dari Tabel 4 dan Gambar 21 disimpulkan bahwa semakin nilai *throughput* mengalami kenaikan maka semakin besar pula kecepatan pengiriman data jaringan. Dimana kualitas jaringan pada nilai *throughput* tertinggi untuk *smart light*, *smart fan* dan *detection gas* di *firebase* yaitu 35,5 kbps dengan persen *throughput* 1,47 % memiliki kategori buruk menggunakan kecepatan pengiriman data 240 kbps dan nilai *throughput* terendah yaitu 0,78 kbps dengan persen *throughput* 2,5% kategori buruk menggunakan kecepatan pengiriman data 32 kbps.

Tabel 5. Data *Delay* Pada *Streaming* Kamera Di IP 192.168.43.107

Pengujian ke	Waktu		Alokasi Bandwidth (kbps)	Total Delay (s)	Packet yang diterima	Delay (ms)
	Tanggal	Jam				
1	03/08/2020	12:53 s.d 12:55	240	105,957	11748	9,01
2	03/08/2020	13:21 s.d 13:23	240	109,198	15999	6,82
3	03/08/2020	14:55 s.d 14:57	240	111,147	23219	4,78
4	03/08/2020	19:14 s.d 19:16	240	108,456	15101	7,18
Rata - rata delay (ms)						7,105
5	03/08/2020	13:15 s.d 13:17	80	119,472	1904	66,94
6	03/08/2020	15:10 s.d 15:12	80	117,531	1620	72,55
7	03/08/2020	16:32 s.d 16:34	80	116,352	1520	76,54
8	03/08/2020	20:08 s.d 20:10	80	116,172	1498	77,55
Rata - rata delay (ms)						73,39
9	03/08/2020	13:40 s.d 13:42	32	110,827	435	243,28
10	03/08/2020	14:25 s.d 14:27	32	113,709	588	193,38
11	03/08/2020	15:20 s.d 15:22	32	115,045	698	164,82
12	03/08/2020	21:02 s.d 21:04	32	115,003	674	170,62
Rata - rata delay (ms)						193,025
Nilai delay tertinggi						193,025
Nilai delay terendah						7,105

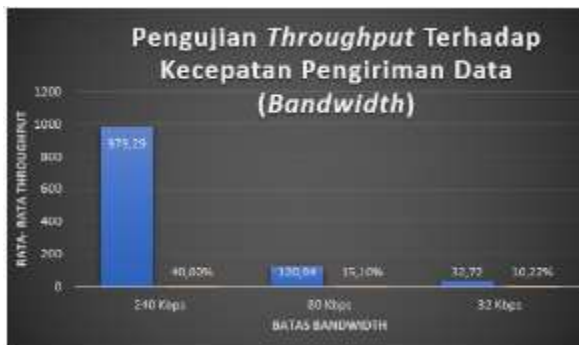


Gambar 22. Grafik Pengujian *Delay* Data *Security* Terhadap Kecepatan Pengiriman Data Jaringan

Berdasarkan data dari Tabel 5 dan Gambar 22 disimpulkan bahwa semakin nilai *delay* mengalami penurunan maka semakin besar pula kecepatan pengiriman data jaringan. Kualitas jaringan pada nilai *delay* tertinggi untuk untuk *streaming* kamera yaitu 193,02 ms kategori bagus dengan kecepatan pengiriman data 32 kbps dan nilai *delay* terendah yaitu 7,1 ms kategori sangat bagus dengan kecepatan pengiriman data 240 kbps.

Tabel 6. Data Pengujian *Throughput Streaming* Kamera Di IP 192.168.43.107

Pengujian ke	Waktu		Alokasi Bandwidth (kbps)	Paket data yang diterima (bytes)	Lama Pengiriman (s)	Throughput (Kbps)	% Throughput
	Tanggal	Jam					
1	03/08/2020	12:53 s/d 12:55	240	10176930	105.957	768,38	40,80
2	03/08/2020	13:21 s/d 13:23	240	13784345	109.198	1009,86	
3	03/08/2020	14:55 s/d 14:57	240	15121151	111.147	1088,37	
4	03/08/2020	19:14 s/d 19:16	240	13971252	108.456	1030,55	
Rata-rata throughput (kbps)						979,29	
5	03/08/2020	13:15 s/d 13:17	80	24115116	119.472	161,47	15,10
6	03/08/2020	15:10 s/d 15:12	80	1230805	117.531	83,76	
7	03/08/2020	16:32 s/d 16:34	80	1414375	117.997	95,89	
8	03/08/2020	20:08 s/d 20:10	80	2118971	119.174	142,24	
Rata-rata throughput (kbps)						120,84	
9	03/08/2020	13:40 s/d 13:42	32	346593	110.827	25,01	10,22
10	03/08/2020	14:25 s/d 14:27	32	455742	113.709	32,08	
11	03/08/2020	15:20 s/d 15:22	32	557681	115.043	38,78	
12	03/08/2020	21:02 s/d 21:04	32	504751	115.003	35,10	
Rata-rata throughput (kbps)						32,72	
Nilai throughput tertinggi (kbps)						451,72	40,80
Nilai throughput terendah (kbps)						69,82	10,22



Gambar 23. Grafik Pengujian *Throughput Data Streaming* Kamera Terhadap Kecepatan Pengiriman Data Jaringan

Berdasarkan data dari Tabel 6 dan Gambar 23 disimpulkan bahwa semakin nilai *throughput* mengalami kenaikan maka semakin besar pula kecepatan pengiriman data jaringan. Dimana kualitas jaringan pada nilai *throughput* tertinggi untuk *streaming* kamera yaitu 979,29 kbps dengan persen *throughput* 40,80% memiliki kategori sedang menggunakan kecepatan pengiriman data 240 kbps dan nilai *throughput* terendah yaitu 32,72 kbps dengan persen *throughput* 10,22% kategori buruk menggunakan kecepatan pengiriman data 32 kbps.

V. KESIMPULAN

Aplikasi *styins home* berhasil menampilkan halaman *login* dan registrasi, hasil *streaming* kamera, kondisi *smart light* dan dapat mengoperasikannya, data suhu dan kelembaban yang diletakkan di *smart fan* serta notifikasi kecepatan kipas dimana jika suhu >28 notifikasi kipas kecepatan *high*, >23 s/d 28 notifikasi suhu normal, >20 s/d < 23 notifikasi suhu *low* dan dapat menampilkan hasil data kadar gas dan asap beserta notifikasi bahaya jika kadar gas >300 PPM sedangkan kadar asap >200 PPM. Dengan hasil dari pengujian *delay* dimana semakin besar nilai *bandwidth* maka semakin nilai *delay* mengalami penurunan. Pada produk *styins home* memiliki nilai *delay* tertinggi pada *smart light*, *smart fan* dan *detection gas* di *firebase* yaitu 425,08 ms (sedang) dengan kecepatan pengiriman data 32 kbps dan nilai *delay* terendah pada *streaming* kamera yaitu 7,1 ms (sangat bagus) dengan kecepatan pengiriman data 240 kbps. Hasil pengujian *throughput* dimana semakin besar nilai *bandwidth* maka semakin besar pula nilai persentasi *throughput*. Dimana nilai persentasi *throughput* tertinggi untuk *smart light*, *smart fan* dan *detection gas* di *firebase* yaitu 1,47 % (buruk) menggunakan *bandwidth* 240 kbps sedangkan nilai persentasi *throughput* terendah pada *streaming* kamera yaitu 40,80% (sedang) menggunakan kecepatan pengiriman data 240 kbps.

REFERENSI

- [1] K. Lampu, S. Wireless, and B. Arduino, "Smart Campus STMIK Sumedang (Smart Classroom)."
- [2] A. S. Mustaqim, D. Kurnianto, F. T. Syifa, and C. Author, "Implementasi Teknologi *Internet of Things* Pada Sistem Pemantauan Kebocoran Gas LPG dan Kebakaran Menggunakan Database Pada *Google Firebase*," vol. 12, no. April, 2020.
- [3] A. Wijaya, Y. Saragih, and I. Lammada, "Yaiao Application Collecting Log File RF 4G Android-Based Replacing Data Cables to Laptop on Traditional RF Drive Test Models," pp. 271–275, 2020.
- [4] A. Kusumaningrum, A. Pujiastuti, and M. Zeny, "Pemanfaatan *Internet of Things* Pada Kendali Lampu," *Compiler*, vol. 6, no. 1, pp. 53–59, 2017, doi: 10.28989/compiler.v6i1.201.

-
- [5] Tamplin J. (2018, April 17). *Firestore Expands To Become A Unified App Platform*. Retrieved from <https://firebase.googleblog.com/2016/05/firebase-expands-to-become-unified-app-platform.html>
- [6] G. R. Paraya and R. Tanone, "Penerapan *Firestore Realtime Database* Pada *Prototype Aplikasi Pemesanan Makanan Berbasis Android*," *J. Tek. Inform. dan Sist. Inf.*, vol. 4, no. 3, pp. 397–406, 2018, doi: 10.28932/jutisi.v4i3.870.
- [7] I. P. Sari and S. Sukri, "Analisis Penerapan Metode Antrian *Hierarchical Token Bucket* untuk Management *Bandwidth* Jaringan Internet," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 2, no. 2, pp. 522–529, 2018, doi: 10.29207/resti.v2i2.458.
- [8] Y. Saragih, Ibrahim, and A. Elisabet, "Study of *Smart Antenna Wide Band Multi Beam by Algorithm Switch Beam*," *J. Sustain. Eng. Proc. Ser.*, vol. 1, no. 2, pp. 247–257, 2019, doi: 10.35793/joseps.v1i2.37.
- [9] A. Sharma, R. Kumar, and V. Mansotra, "Proposed *Stemming Algorithm for Hindi Information Retrieval*," *Int. J. Innov. Res. Comput. Commun. Eng. (An ISO Certif. Organ.)*, vol. 3297, no. 6, pp. 11449–11455, 2016, doi: 10.15680/IJIRCCE.2016.
- [10] Y. Mardiana and J. Sahputra, "Analisa Performansi Protokol TCP , UDP dan SCTP," *J. Media Infotama*, vol. 13, no. 2, pp. 73–84, 2017.
- [11] R. Hidayat, R. A. Darajatun, R. Setiawan, and V. P. Fahrani, "Alat pengukur kecepatan lari berbasis mikrokontroler 1," vol. 4, no. 1, pp. 167–173, 2019.
- [12] P. Sinyal, I. Berbasis, and I. R. U. G. Dqg, "1), 2), 3)."
- [13] A. Fauzi, "Analisis Kualitas Transmisi Data Pada *E-Learning Streaming Multimedia* Dengan *Quality Of Service (QoS)*," *Semin. Nas. Inov. Teknol.*, pp. 93–106, 2019.
- [14] R. Wulandari, "Analisis QoS (*Quality Of Service*) Pada Jaringan Internet (Studi Kasus : Upt Loka Uji Teknik Penambangan Jampang Kulon – LIPI)," *J. Tek. Inform. dan Sist. Inf.*, vol. 2, no. 2, pp. 162–172, 2016, doi: 10.28932/jutisi.v2i2.454.