

## Rancang Bangun Alat Pendeteksi Uap *Vaping* dan Asap Rokok Berbasis *Internet of Things* Terintegrasi Aplikasi Android

Rifqi Fuadi Hasani<sup>1</sup>, Benny Nixon<sup>2</sup>, Alifa Humairah<sup>3</sup>, Augita Yasmin<sup>4</sup>

<sup>1,2,3,4</sup>Telekomunikasi, Teknik Elektro, Politeknik Negeri Jakarta, Jalan Prof. DR. G.A. Siwabessy, Kampus UI, Depok 16425, Indonesia

E-mail: rifqi.fuadihasani@elektro.pnj.ac.id

### Abstrak

Merokok adalah suatu tindakan yang memicu penyakit terutama pada saluran pernafasan karena kandungannya yang tidak ramah untuk tubuh. Bagi seorang perokok pasif juga sangatlah dirugikan karena asap/uap yang terhirup. Maka dari itu, alat pendeteksi asap rokok dirancang dengan memanfaatkan fungsi jaringan WiFi yang tersambung dengan ESP32 yang akan mengirimkan data deteksi dari sensor MQ-2 (pendeteksi kandungan asap rokok tembakau) dan sensor Adafruit (pendeteksi kandungan uap rokok elektrik) menuju firebase yang terintegrasi dengan aplikasi android. Hal ini akan memudahkan bagi firma perhotelan karena alat ini tidak hanya dapat mendeteksi asap rokok tembakau saja, namun uap rokok elektrik pun juga akan terdeteksi sehingga tamu yang melanggar selain menggunakan rokok juga akan dapat dikenai denda yang mana akan dipantau oleh pegawai hotel tersebut secara jarak jauh melalui aplikasi android. Sistem dari alat ini khusus untuk sekali terdeteksi saja yang mana jika terdeteksi, maka LED dari hijau berubah menjadi kuning (rokok elektrik) jika sudah menyentuh  $\geq 25$  ppm dan/atau merah (rokok tembakau) jika sudah menyentuh  $\geq 0,05$  ppm yang diikuti dengan suara buzzer dan kipas DC yang akan membuang asap/uap keluar ruangan. Ruangan netral yang teruji selama 10 menit menghasilkan maksimal asap yang terdeteksi sebanyak 0,03 ppm dan uap yang terdeteksi sebanyak 24 ppm. Pada percobaan sensor MQ-2 dengan kadar 1 semburan menghasilkan 0,42 ppm dan delay selama 2 detik untuk sensor mendeteksi, pada ruang yang bervolume 1050 m<sup>3</sup> kembali bersih dengan delay 11 detik dengan nilai 0,04 ppm. Pada percobaan sensor Adafruit dengan kadar 1 semburan menghasilkan 33 ppm, delay 2 detik untuk sensor mendeteksi, pada ruang yang bervolume 1050 m<sup>3</sup> kembali bersih dengan delay 8 detik dengan nilai 24 ppm.

*Kata Kunci: Android, Asap, Hotel, Internet of Things, MQ-2, Vaping*

### Abstract

Smoking is one of causes disease which especially on respiratory tract because the cigarette were content that is not good for body. Passive smoker is really made a loss for smoke/vapor that they inhaled. So therefore, smoke detector was created by using connection of WiFi that connected to ESP32 which sending data of detection from MQ-2 sensor (detect content's smoke of tobacco cigarette) and Adafruit sensor (detect content's vapor of e-cigarette) to firebase that integrated by android software. This would facilitated for hotel firm because the hardware not only detect smoke of cigarette, but also vapor of vape will detected so guest that disobey the rules who using except cigarette will be fined which will be monitored by hotel's employee remotely via android software. System of this hardware specifically only for once detected which if it detected, then green LED changed to yellow (e-cigarette) if the amount shows  $\geq 25$  ppm and/or red (tobacco cigarette) if the amount shows  $\geq 0.05$  ppm which followed by sound of buzzer and DC fan that will dispensed smoke/vapor out of room. Neutral room which was tested for 10 minutes produced a maximum of 0.03 ppm amount of detected smoke and 24 ppm amount of detected vapor. In the MQ-2 sensor experiment with a level of 1 burst it produces 0.42 ppm and delay of 2 seconds for the sensor to detect, in a room with a volume of 1050 m<sup>3</sup> it returns clean with a delay of 11 seconds with amount of 0.04 ppm. In the Adafruit sensor experiment with a concentration of 1 burst it produces 33 ppm, a delay of 2 seconds for the sensor to detect, in a room with a volume of 1050 m<sup>3</sup> it returns clean with a delay of 8 seconds with amount of 24 ppm.

*Keywords: Android, Internet of Things, Hotel, MQ-2, Smoke, Vaping*

### 1. Pendahuluan

Pada dunia perhotelan, manajemen hotel cukup ketat dengan aturan merokok maupun itu rokok tembakau atau rokok elektrik. Pada saat *check-in*, para tamu akan melihat papan aturan pengunjung hotel yang salah satunya berisi tidak diizinkan merokok di dalam kamar, termasuk balkon. Cara bekerja pegawai hotel saat pemeriksaan kamar-kamar yang memiliki bau khas rokok dinilai memakan waktu dan tidak efisien. Dari permasalahan yang kerap terjadi pada seluruh perhotelan, maka dibuatkan alat Rancang Bangun Sistem Pendeteksi Asap Rokok Tembakau dan Uap Rokok Elektrik di Kamar Hotel berbasis *Internet of Things* untuk mempermudah para pegawai melakukan pengendalian pemantauan serta menerapkan hidup sehat bagi para penghuni hotel.

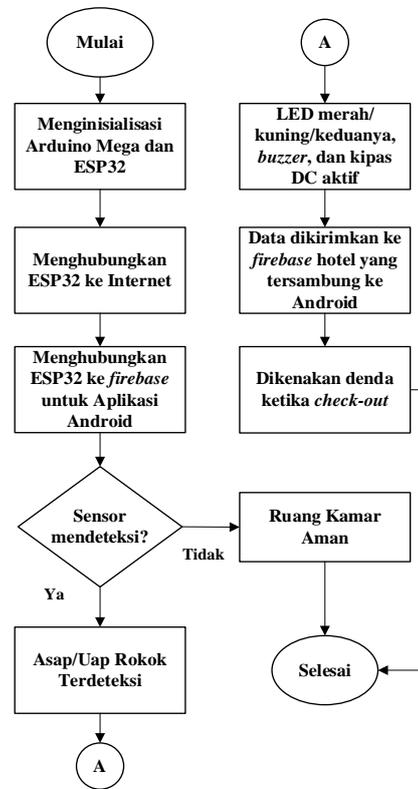
*Internet of Things* (IoT) adalah solusi komunikasi antar mesin yang dapat memberi visibilitas usaha secara *real-time*. Konektivitas 24 jam disertakan berbagai fitur otomasi menjadikan IoT sebagai cara efektif untuk memantau dan mengelola usaha dari jarak jauh [1].

### 2. Metode Penelitian

#### 2.1 Perancangan Sistem Alat

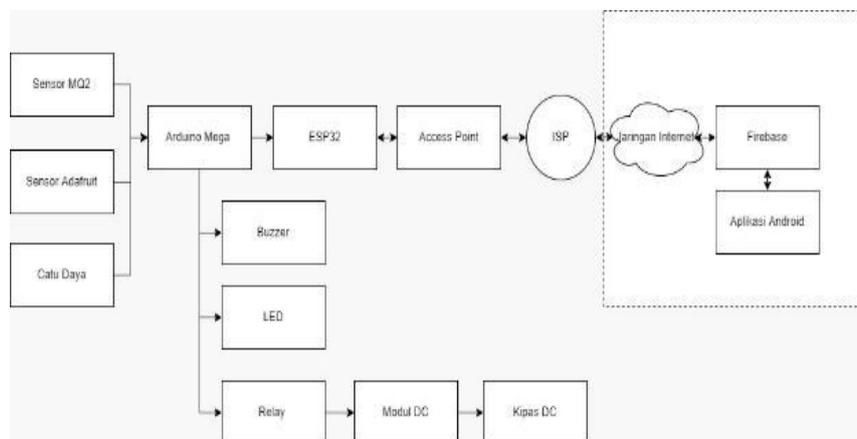
Cara kerja dari alat ini yaitu jika ruangan aman dari asap, maka ditandai dengan LED berwarna hijau. Ketika ada yang merokok, maka kandungan rokok tembakau atau rokok elektrik atau jika keduanya terdeteksi oleh sensor *MQ-2* [4] dan/atau sensor *Adafruit* yang menyebabkan LED kuning menyala sebagai tanda rokok elektrik terdeteksi oleh sensor *Adafruit* dan LED merah menyala sebagai tanda rokok tembakau terdeteksi oleh sensor *MQ-2* yang diikuti dengan *buzzer* yang berdering dan kipas DC yang aktif untuk mengeluarkan asap dalam

ruangan [4]. Hal ini ditunjukkan pada diagram alir pada Gambar 1.



Gambar 1. Diagram Alir Cara Kerja Sistem Alat

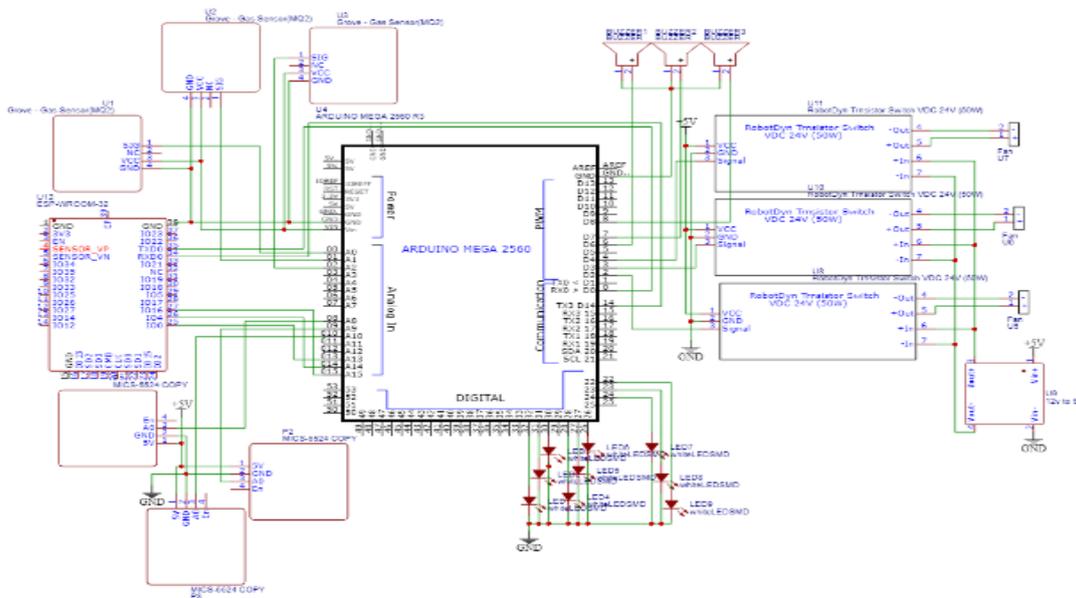
Pada realisasi perangkat catu daya untuk alat, maka dibuatkan rangkaian catu daya yang berfungsi sebagai tata letak penjaluran komponen-komponen yang akan digunakan di PCB [5] seperti yang ditunjukkan pada Gambar 4.



Gambar 2. Diagram Blok Sistem Alat

Setelah semua kaki komponen sudah di inialisasi langkah selanjutnya adalah menghubungkan *ESP32* ke internet melalui *access point*. Setelah terhubung ke internet, *ESP32* menyalurkan data ke *firebase* (*database* secara *real-time*) yang terhubung ke aplikasi android.

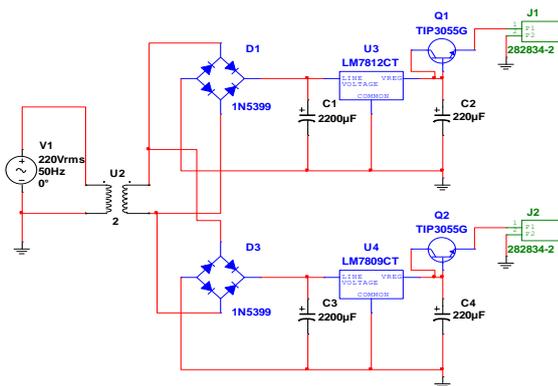
Lalu, *ESP32* akan menunggu hasil deteksi yang akan dikirimkan menuju *firebase* yang terhubung dengan aplikasi android. Sistem alat menuju aplikasi tersebut dapat dilihat dari diagram blok pada Gambar 2.



Gambar 3. Rangkaian Skematik Alat

Realisasi sistem pendeteksi asap rokok tembakau dan uap rokok elektrik adalah dengan menghubungkan semua komponen ke pusat mikrokontroler yaitu *Arduino Mega 2560*. Skematik perancangan sistem pendeteksi asap rokok tembakau dan uap rokok elektrik dapat dilihat pada Gambar 3

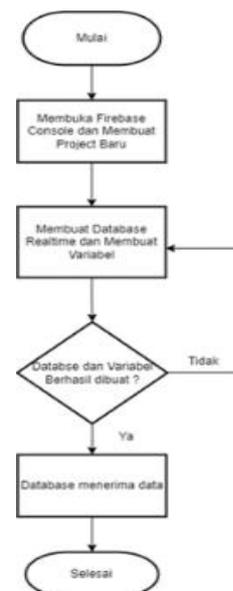
antara sistem mikrokontroler dengan aplikasi. *Database* yang digunakan dalam perancangan ini adalah *google firebase*. Gambar 5 merupakan diagram alir perancangan *real-time database firebase*.



Gambar 4. Rangkaian Catu Daya

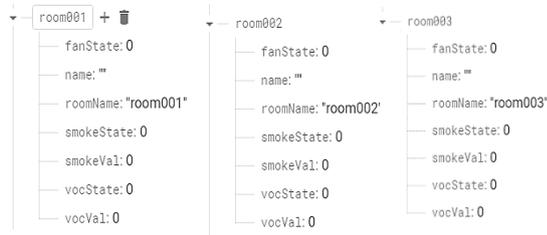
2.2. Perancangan Sistem Aplikasi

Pada perancangan sistem aplikasi hal yang harus dilakukan pertama kali adalah membuat *realtime database firebase*. Perancangan *realtime database firebase* diperlukan agar sistem dapat menyimpan hasil data nilai dari sensor dan sebagai jembatan penghubung



Gambar 5. Diagram Alir Perancangan *Real-time Database Firebase*

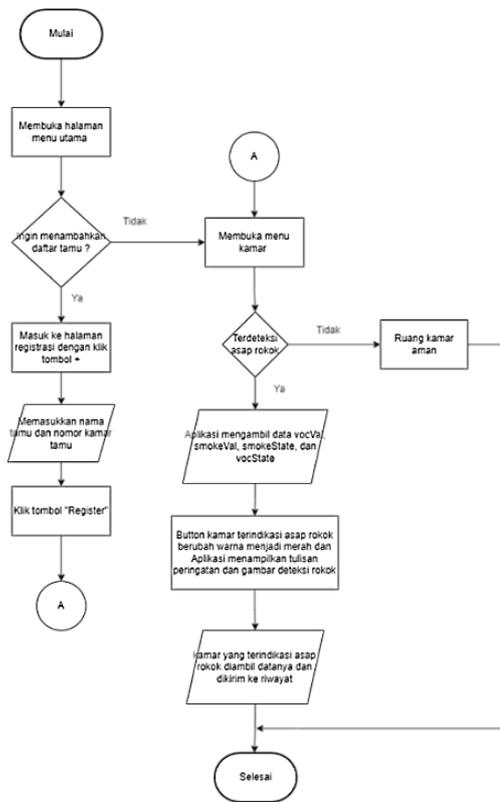
Untuk menampilkan data sensor maka dibuat beberapa variable dalam *database*. Dengan cara klik *realtime database* lalu penambahan variable dapat dilakukan dengan mengklik “+” dimana hasil yang didapat seperti Gambar 6 yaitu variable *database*



Gambar 6. Variabel Database

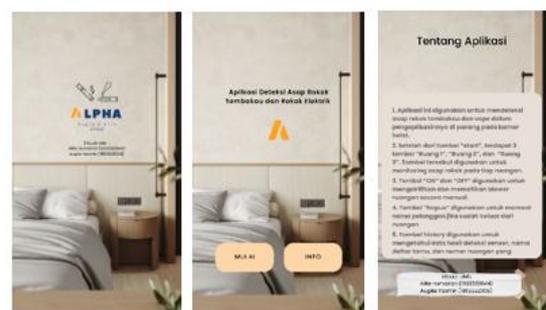
Variable *parent* yang digunakan meliputi “*room001*”, “*room002*”, dan “*room003*”. Ketiganya adalah *variable* yang merepresentasikan tiap-tiap ruang hotel. Variable tersebut menyimpan data/nilai pengukuran dari sensor MQ2 dan sensor adafruit berupa *fanState*, *name*, *roomName*, *smokeState*, *smokeVal*, *vocState*, dan *vocVal*. *Smoke state* akan menyimpan data berupa angka 0 dan 1 dari sensor MQ2, *SmokeVal* menyimpan data berupa nilai PPM dari sensor MQ2. Sedangkan *vocState* menyimpan data berupa angka 0 dan 1 dari sensor adafruit, dan *vocVal* menyimpan data berupa nilai PPM dari sensor adafruit.

Setelah dilakukan perancangan *database* selanjutnya dilakukan perancangan aplikasi. Gambar 7 menunjukkan diagram alir perancangan aplikasi “ALPHA”.



Gambar 7. Diagram Alir Perancangan Aplikasi "ALPHA"

Pada aplikasi terdapat 7 buah tampilan yaitu tampilan *splash screen*, tampilan awal (*welcome activity*), tampilan tentang aplikasi (*about activity*), tampilan menu utama (*room selection activity*), tampilan riwayat (*history*), tampilan daftar tamu (*register*), dan tampilan ruangan (*room activity*).



(a)



(b)

Gambar 8 (a) dan 8 (b). Tampilan Aplikasi ALPHA

### 3. Hasil dan Pembahasan

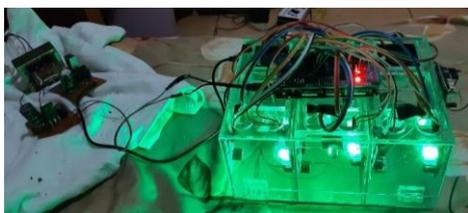
#### 3.1 Pengujian Catu Daya

Pengujian pertama yang dilakukan yaitu mengukur nilai dari tegangan serta arus masukan dan keluaran yang dihasilkan oleh catu daya yang telah dibuat dengan menggunakan alat ukur multimeter. Pengujian ini bertujuan untuk mengetahui catu daya yang telah dibuat dapat berfungsi dengan baik dan sesuai spesifikasi yang dibutuhkan untuk alat mikrokontroler. Adapun hasil pengujian catu daya terdapat pada Tabel 2 berikut.

Tabel 2. Hasil Uji Tegangan pada Input dan Output Catu Daya

Tegangan Input (VAC)	Spesifikasi (VAC)	Tegangan Output (VDC)	Spesifikasi (VDC)
17,9	18	11,9	12
8,8	9	8,9	9

Hasil uji dari catu daya didapatkan *input* dari catu daya 8,8 VAC dan 17,9 VAC menghasilkan *arduino mega 2560* yang terhubung dengan *jack power* catu daya dengan *output* 8,9 VDC dan modul DC terhubung dengan *jack power* catu daya dengan *output* 11,9 VDC yang terintegrasi keseluruhan sistem alat berjalan dengan sempurna yang ditunjukkan pada Gambar 9.



Gambar 9. Alat Terhubung dengan Catu Daya

#### 3.1 Pengujian Sensor dengan Rokok

Pengujian kedua yang dilakukan yaitu menguji sensor dengan bahan uji rokok tembakau dan rokok elektrik yang menggunakan alat ukur stopwatch dan dioperasikan dengan pemrograman Arduino IDE. Pengujian ini bertujuan untuk mengetahui respon sensor sudah bekerja dengan yang seharusnya dan nilai ppm yang sudah sesuai.

Adapun hasil pengujian sensor Adafruit terdapat pada Tabel 3 dan Tabel 4 berikut.

Tabel 3. Hasil Uji Sensor Adafruit

Kadar Upap (ppm)	Delay (s)	Indikator
24	-	Netral
33	2	1 Semburan
24	8	Aman
57	3	2 Semburan
23	10	Aman
101	3	3 Semburan
24	20	Aman
154	3	4 Semburan
24	29	Aman

Tabel 4. Hasil Uji Sensor MQ-2

Kadar Asap (ppm)	Delay (s)	Indikator
0,03	-	Netral
0,42	2	1 Semburan
0,04	8	Aman
5,9	3	2 Semburan
0,03	10	Aman
10,07	3	3 Semburan
0,04	20	Aman
13,24	3	4 Semburan
0,02	29	Aman

Hasil uji didapatkan bahwa semakin banyak semburan maka semakin tinggi ppm nya dan semakin lama waktu ruangan untuk kembali bersih.

#### 3.3 Pengujian Respon Jaringan

Pengujian ketiga yang dilakukan yaitu menguji respon jaringan terhadap data yang dikirimkan ke *firebase* dengan *stopwatch*, meteran, dan aplikasi *Network Cell*. Pengujian ini bertujuan untuk mengetahui jarak berpengaruh dengan kualitas data pada proses pengiriman data ke *firebase*. Adapun hasil pengujian konektivitas jaringan dari *ESP32* ke *firebase* ditunjukkan pada Tabel 5 dan Tabel 6 berikut.

Tabel 5. Hasil Uji Respon Jaringan

Jarak (m)	Delay (ms)	Ket.
1	63	Data terkirim
1,5	99	Data terkirim
3	206	Data terkirim
5	255	Data terkirim
8	479	Data terkirim
10	600	Data terkirim

Tabel 6. Hasil Uji Daya Pancar *Provider XL ke Firebase*

Jarak	RSSI	RSRP	RSRQ	Ket.
1 m	-28 dBm	-92 dBm	-10 dB	RSRP = Normal RSRQ = Baik RSSI = Sangat Baik
1,5 m	-35 dBm	-97 dBm	-12 dB	RSRP = Normal RSRQ = Normal RSSI = Sangat Baik
3 m	-50 dBm	-100 dBm	-14 dB	RSRP = Normal RSRQ = Normal RSSI = Sangat Baik
5 m	-64 dBm	-101 dBm	-14 dB	RSRP = Buruk RSRQ = Normal RSSI = Buruk
8 m	-74 dBm	-104 dBm	-15 dB	RSRP = Buruk RSRQ = Normal RSSI = Buruk
10 m	-80 dBm	-105 dBm	-16 dB	RSRP = Buruk RSRQ = Buruk RSSI = Sangat Buruk

*Firestore (Real Time Database)* merupakan *cloud database*. Data disimpan dalam format JSON dan disinkronkan secara *real-time* ke setiap klien yang terhubung [2]. Hasil uji respon jaringan terhadap data yang dikirimkan ke *firebase* menurut standar jaringan 4G LTE memiliki beberapa parameter ukur, yaitu :

1. RSRP (*Received Signal Reference Power*) merupakan parameter yang menyatakan tingkat

kekuatan sinyal yang diterima oleh user dalam satuan dBm. Nilainya bergantung jarak user dengan eNodeB. [3].

2. RSSI (*Received Signal Strength Indicator*) merupakan parameter yang menyatakan keseluruhan daya sinyal yang diterima oleh user dalam satuan dBm [3].
3. RSRQ (*Received Signal Reference Quality*) merupakan perbandingan antara RSRP dan RSSI [3].

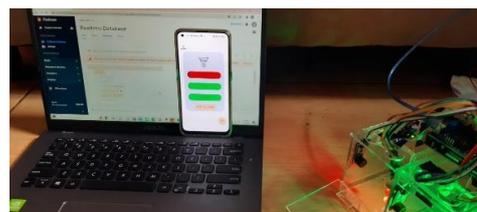
Maka, semakin jauh jarak antara *access point* dengan *ESP32* akan memengaruhi lambat dan kualitas yang buruk pada pengiriman data dari *ESP32* menuju *firebase*.

### 3.3 Pengujian Komunikasi Data dengan 2 Jaringan

Pengujian keempat yang dilakukan yaitu mengomunikasikan data dengan mengoneksikan dua jaringan yang berbeda pada alat dan aplikasi. Pengujian ini bertujuan untuk mengetahui mobilitas antara alat dan aplikasi sehingga dapat melakukan pemantauan alat jarak jauh melalui aplikasi yang terintegrasi dengan internet. Pengujian dilakukan seperti pada Gambar 10 dan hasil pada Gambar 11.



(a)



(b)

Gambar 10 (a) dan 10 (b). Pengujian *Internet of Things*

Hasil uji *Internet of Things* pada alat pendeteksi asap rokok tembakau dan uap rokok elektrik dibuktikan dengan sensor yang menangkap hasil deteksi lalu dikirimkan ke *firebase* yang menyebabkan aplikasi android mendapati tombol dari warna hijau menjadi merah dan dari aplikasi android dapat menekan tombol kipas yang dapat dihidupkan dan dimatikan kembali secara manual ke alat. Pada alat dikhususkan menggunakan hanya satu *access point* sesuai dengan yang sudah diprogramkan untuk *ESP32*, sedangkan

untuk aplikasi android dapat menggunakan *access point* apa saja dan dapat dioperasikan pada perangkat apa saja.

### 3.4 Pengujian QoS Terhadap Aplikasi

Pengujian QoS dilakukan pada ruangan 1, ruangan 2, dan ruangan 3 dengan menggunakan parameter *throughput*, *packet loss*, dan *delay*.

Tabel 7. Pengujian QoS

Ruang	Parameter	Hasil
1	Packets	722
	Time Span (s)	99,349
	Bytes	251568
2	Packets	684
	Time Span (s)	41,449
	Bytes	232545
3	Packets	1503
	Time Span (s)	67,099
	Bytes	730711

Dari Tabel 7, selanjutnya dilakukan perhitungan *throughput*, *packet loss*, dan *delay*. Dari data tersebut didapatkan besar nilai *throughput* dengan menggunakan persamaan

$$\text{Throughput} = \frac{\text{Packet received (kb)}}{\text{Time transmitted (s)}} \times 8 \text{ bit}$$

Besar *throughput* yang didapatkan pada ruangan 1 dengan paket yang diterima sebesar 251568 dan waktu yang didapatkan sebesar 99,348. Didapatkan *throughput* pada ruangan 1 sebesar :

$$\text{Throughput} = \frac{251568 \text{ bytes}}{99,348 \text{ s}} \times 8 = 20,256 \text{ KB/s}$$

Besar *throughput* yang didapatkan pada ruangan 2 dengan paket yang diterima sebesar 232545 dan waktu yang didapatkan sebesar 41,449. Didapatkan *throughput* pada ruangan 2 sebesar :

$$\text{Throughput} = \frac{232545 \text{ bytes}}{41,449 \text{ s}} \times 8 = 44,88 \text{ KB/s}$$

Besar *throughput* yang didapatkan pada ruangan 3 dengan paket yang diterima sebesar 730711 dan waktu yang didapatkan sebesar 67,099. Didapatkan *throughput* pada ruangan 3 sebesar :

$$\text{Throughput} = \frac{730711 \text{ bytes}}{67,099 \text{ s}} \times 8 = 87,12 \text{ KB/s}$$

Berdasarkan Tabel 7, selanjutnya dilakukan perhitungan untuk mendapatkan hasil nilai *packet loss*. Pada ruangan 1 didapatkan jumlah data yang dikirim sebanyak 722 dan paket yang diterima sebanyak 720.

$$\text{Packet Loss} =$$

$$= \frac{(\text{packet transmittes} - \text{packet received})}{\text{packet transmitter}} \times 100 \%$$

Besar *packet loss* yang dihasilkan pada ruangan 1 didapatkan :

$$\text{Packet Loss} = \frac{(722 - 720)}{722} \times 100 \% = 0,2776\%$$

Pada ruangan 2 didapatkan jumlah data yang dikirim sebanyak 684 dan data yang diterima sebanyak 682. Besar *packet loss* pada ruangan 2 didapatkan :

$$\text{Packet Loss} = \frac{(684 - 682)}{684} \times 100 \% = 0,292\%$$

Pada ruangan 3 didapatkan jumlah data yang dikirim sebanyak 1503 dan data yang diterima sebanyak 1494. Besar *packet loss* pada ruangan 3 didapatkan :

$$\text{Packet Loss} = \frac{(1503 - 1494)}{1503} \times 100 \% = 0,59\%$$

Berdasarkan Tabel 7, selanjutnya dilakukan pengukuran *delay* dengan menggunakan persamaan sebagai berikut.

$$\text{Delay} = \frac{\text{Total delay}}{\text{Jumlah total paket}}$$

Hasil *delay* yang didapat pada saat sistem terhubung dengan provider Telkomsel. Pada pengujian ruangan 1 total *delay* yang didapat sebesar 75,89873 dan jumlah total paket yang didapatkan sebesar 720. Besar *delay* yang didapatkan pada ruangan 1 sebesar :

$$\text{Delay} = \frac{75,89873}{720} = 0,105 \text{ s} = 105 \text{ ms}$$

Pada ruangan 2 total *delay* yang didapatkan sebesar 39,26892 dan jumlah total paket yang didapatkan sebesar 59. Besar *delay* yang didapatkan pada ruangan 2 sebesar :

$$\text{Delay} = \frac{39,26892}{682} = 0,059 \text{ s} = 59 \text{ ms}$$

Pada ruangan 3 total *delay* yang didapatkan sebesar 66,23584 dan jumlah total paket yang didapatkan sebesar 1494. Besar *delay* yang didapatkan pada ruangan 3 sebesar :

$$\text{Delay} = \frac{66,23584}{1494} = 0,044 \text{ s} = 44 \text{ ms}$$

## 4. Simpulan

Hasil uji dari catu daya didapatkan *input* dari catu daya 8,4 VAC dan 17,3 VAC menghasilkan *output* 8,9 VDC dan 11,9 VDC. Hasil uji sensor menunjukkan semakin banyak semburan maka semakin tinggi kadar nya dan semakin lama waktu ruangan kembali bersih. Hasil uji respon jaringan terhadap data yang dikirimkan ke *firebase* menunjukkan semakin jauh jarak antara *access point* dengan *ESP32* akan memengaruhi lambat dan

kualitas pengiriman data. Performansi jaringan internet terhadap aplikasi android didapatkan dari pengujian tiap ruangan secara berturut-turut adalah; nilai *throughput* sebesar 20,256 KB/s, 44,88 KB/s, dan 87 KB/s, nilai *packet loss* sebesar 0,2776 %, 0,292 %, dan 0,6 %, dan nilai *delay* sebesar 105 ms, 59 ms, dan 44 ms. Performansi jaringan ini dikategorikan sangat bagus yang ditetapkan oleh TIPHON.

### Daftar Acuan

- [1] A. Rinaldi, Artikel *Internet Of Things*, 2020, <https://smkn4tangsel.sch.id/read/7/artikel-internet-of-things>
- [2] I.F.Maulana, “Penerapan Firebase Realtime Database pada Aplikasi E-Tilang Smartphone berbasis Mobile Android”, *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 4, no. 5, pp. 854-863, Oct. 2020.
- [3] Anon., Pengertian RSSI RSRQ RSRP SINR Pada Sinyal LTE, 2012, <https://panduanmeta.com/2021/07/pengertian-rssi-sinr-pada-sinyal-lte.html>
- [4] F.Sibarani, “Alat Pendeteksi Kebocoran Gas Menggunakan Sensor MQ-2 Berbasis Arduino Uno dan Buzzer”, Skripsi USU, 2018.
- [5] E.P Sitohang, “Rancang Bangun Catu Daya DC Menggunakan Mikrokontroler ATmega 8535”, *Jurnal Teknik Elektro dan Komputer* Vol. 7 No.2, 2018.