

Research Article

Sistem Pendeteksi dan Pemadam Kebakaran Otomatis Berbasis *Internet of Things* (Iot) Dengan Notifikasi *Real-Time*Ery Safrianti^{1,*}, Arif Maulana¹, Linna Oktaviana Sari², R.A Rizka Qori Yuliani Putri¹

ABSTRAK: Bencana kebakaran tidak dapat diprediksi waktu dan tempatnya, sehingga dibutuhkan teknologi sistem peringatan dan penanggulangan dini. Penelitian ini bertujuan merancang sistem prototype berbasis *Internet of Things* (IoT) jika terjadi kebakaran dalam suatu ruangan. Terdapat beberapa fitur dalam perancangan sistem yaitu mikrokontroler (ESP32) yang terhubung dengan 3 yakni sensor api (*5-channel flame sensor*), suhu (DHT22) dan asap (MQ2). Data dari ketiga sensor di tampilkan *real-time* melalui *website server* lokal dalam kondisi normal atau tidak saat di monitoring. Saat kondisi tidak normal ketiga sensor terbaca oleh sistem maka akan terjadi pemutusan arus listrik dan pemadaman api secara otomatis. Sistem juga mengirimkan notifikasi berupa pesan dan panggilan kepada pemadam kebakaran melalui layanan *CallMeBot*, jika api tidak padam dalam waktu tertentu. Pengujian dilakukan untuk 4 objek yakni kertas, korek api gas, sumbu kompor dan rokok. Pengujian pada kertas sistem pemadaman aktif dan pesan telepon tidak aktif sedangkan pada sumbu kompor sistem keduanya aktif karena api tidak padam dalam waktu lebih dari 5 detik yang ditentukan. Hasil perhitungan QoS pada pengiriman data didapatkan nilai *packet loss* 0,1%, *delay request* dan *delay respond*, untuk nilai *webserver* yaitu 95,60758918 ms dan 95,06614461 ms. Pada telegram nilainya sebesar 89,96194943 ms dan 76,00959068 ms.

Kata kunci: Kebakaran, IoT, ESP32, DHT22, MQ-2

1. PENDAHULUAN

Kebakaran dikategorikan dalam bentuk salah satu bencana non-alam. Menurut Dewan Keselamatan dan Kesehatan Kerja Nasional (DK3N), kebakaran merupakan suatu peristiwa bencana yang berasal dari api yang tidak dikehendaki yang dapat menimbulkan kerugian, baik kerugian materi (berupa harta benda, bangunan fisik, deposit/asuransi, fasilitas sarana dan prasarana, dan lain-lain) maupun kerugian non materi (rasa takut, shock, ketakutan, dan lain-lain) hingga kehilangan nyawa atau cacat tubuh yang ditimbulkan akibat kebakaran tersebut.

Kebakaran melibatkan proses pembakaran yang tidak terkendali, terdapat 3 unsur yakni timbulnya api, panas, dan asap menyebar secara cepat hingga merusak lingkungan sekitarnya. Berdasarkan *National Fire Protection Association* (NFPA), kebakaran diklasifikasikan menjadi lima kelas berdasarkan jenis bahan yang terbakar dan metode pemadamannya, yakni Kelas A (bahan padat non-logam seperti kayu dan kertas, dipadamkan dengan air), Kelas B (cairan dan gas mudah terbakar seperti bensin dan alkohol, dipadamkan dengan busa atau karbon dioksida), Kelas C (peralatan listrik aktif, dipadamkan dengan bahan yang tidak menghantarkan listrik seperti karbon dioksida), Kelas D (logam yang mudah terbakar seperti magnesium dan natrium, dipadamkan dengan

OPEN ACCESS **Afiliasi:**

¹Program Studi Elektro, Universitas Riau, Pekanbaru, Indonesia

²Program Studi Informatika, Universitas Riau, Pekanbaru, Indonesia

Korespondensi:

Email: esafrianti@eng.unri.ac.id

Diterima Redaksi:

16-05-2026

Selesai Revisi:

26-05-2025

Diterbitkan Online:

01-06-2025

Cara Mengutip Artikel ini (IEEE):

E. Safrianti, A. Maulana, L. O. Sari, and R. R. Q. Y. Putri, "Sistem Pendeteksi dan Pemadam Kebakaran Otomatis Berbasis Internet of Things (Iot) Dengan Notifikasi Real-Time", *J. Electo*, vol. 2, no. 01, pp. 42–55, June. 2026.

This article is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.



serbuk kering khusus), dan Kelas K (minyak dan lemak goreng di dapur, dipadamkan dengan bahan kimia basah) [1].

Pada saat ini banyak hal dapat dilakukan melalui kegiatan remote atau jarak jauh dengan dukungan internet [2]. Oleh karena itu, IoT dapat dijadikan upaya dalam pengembangan sistem pendeteksi dan pemadaman kebakaran otomatis dengan notifikasi secara *real-time* untuk meminimalisir kejadian lebih awal.

Beberapa penelitian yang terkait dengan sistem pendeteksi serta pemadaman kebakaran telah dilakukan sebelumnya. Pada penelitian yang dilakukan oleh [3] merupakan sistem deteksi kebakaran berbasis IoT dengan perangkat arduino serta tiga sensor api, sensor suhu dan sensor asap dengan adanya sistem sms ataupun telepon sebagai notifikasi peringatan dan menggunakan suatu website sebagai monitoringnya. Pada penelitian lainnya yang dilakukan oleh [4] terdapat suatu prototipe sistem deteksi kebakaran berbasis IoT dengan menggunakan mikrokontroler NodeMCU, sensor asap, dan sensor api, serta didukung oleh aplikasi Android untuk pemantauan jarak jauh dan menggunakan pompa air kecil sebagai pemadaman api dengan sistem otomatis.

Pada penelitian ini, penulis menggunakan mikrokontroler ESP32 yang lebih canggih dan hemat daya dibandingkan dengan NodeMCU. Sistem ini dilengkapi dengan tiga jenis sensor, yaitu sensor api 5 channel, sensor MQ-2 untuk mendeteksi gas dan asap, serta sensor suhu DHT-22 untuk memantau suhu. Semua sensor ini berfungsi untuk memberikan data dan perintah secara real-time yang dapat dimonitoring melalui web server dengan mempunyai beberapa fitur. Fitur-fitur dalam web server ini meliputi tampilan data pembacaan sensor, notifikasi kebakaran, serta adanya sistem telepon pintar yang dapat secara otomatis menghubungi pemadam kebakaran dengan mengirimkan pesan ke lokasi jika terdeteksi kebakaran. Selain itu, terdapat juga mekanisme pemadaman api dan pemutus aliran listrik jika terjadi kebakaran ataupun korsleting listrik, sehingga dapat mencegah penyebaran kebakaran lebih lanjut.

Berdasarkan permasalahan yang ada sekaligus mengembangkan teknologi dan inovasi, maka dilakukan penelitian tentang sistem pendeteksi dan pemadaman kebakaran otomatis berbasis *internet of things* (iot) dengan notifikasi *real-time*. Dengan sistem ini, diharapkan risiko kejadian kebakaran dapat diminimalisir melalui deteksi dini dan respon cepat, sehingga kerugian baik materi maupun non-materi dapat dikurangi secara signifikan. Inovasi ini diharapkan mampu memberikan kontribusi nyata dalam meningkatkan keamanan dan keselamatan, serta mencegah bencana kebakaran sejak dini. Pada tabel 1 dapat dilihat beberapa penelitian terdahulu terkait topik penelitian ini.

Tabel 1 Perbandingan Penelitian Sebelumnya

Referensi	Perangkat yang digunakan	Kelebihan	Kekurangan
[5]	Sensor api, sensor asap MQ-135, Arduino Uno R3, modul GSM SIM900A, buzzer.	- Flame sensor efektif mendeteksi api dengan panjang gelombang 760 -1100nm. - Sensor MQ-135 mendeteksi gas berbahaya (NH₃, NO_x, alkohol, benzena, asap, CO₂). - Sistem dapat mengirim notifikasi SMS saat mendeteksi api/asap. - Sistem memiliki respon cepat dengan buzzer alarm local.	- Hanya mengandalkan SMS untuk notifikasi jarak jauh. - Tidak memiliki fitur pemantauan <i>real-time</i> melalui aplikasi atau web. - Tidak ada pemadaman otomatis.
[3]	Sensor api, sensor suhu DHT-11, sensor asap MQ-7, Arduino, modul GSM SIM800L V2, buzzer.	- Menggunakan tiga sensor utama untuk deteksi yang lebih akurat (api, suhu, asap). - Sistem tetap berfungsi meski salah satu sensor rusak. - Notifikasi melalui panggilan telepon dan buzzer. - Pemantauan real-time melalui website memungkinkan kontrol jarak jauh.	- Bergantung pada koneksi internet untuk pemantauan web. - Tidak memiliki aplikasi mobile khusus untuk notifikasi. - Tidak ada fitur pemadaman otomatis.

[6]	Arduino Uno, sensor api, DHT22, MQ-2, display OLED, buzzer, pompa air.	- Menggunakan tiga sensor utama untuk deteksi yang lebih akurat (api, suhu, asap). - Sistem tetap berfungsi meski salah satu sensor rusak. - Notifikasi melalui panggilan telepon dan buzzer. - Pemantauan real-time melalui website memungkinkan kontrol jarak jauh.	- Keterbatasan dalam jangkauan deteksi sensor api dan asap. - Tidak ada fitur notifikasi jarak jauh. - Tidak ada pemantauan real-time melalui aplikasi atau web.
[4]	NodeMCU ESP8266, sensor MQ-2, sensor api, buzzer, pompa air mini, aplikasi Android	- Integrasi IoT memungkinkan pemantauan dan kontrol real-time melalui aplikasi Android. - Notifikasi real-time dan pemadaman otomatis. - Menggunakan metode prototyping sehingga dapat disesuaikan dengan kebutuhan spesifik. - Sensor MQ-2 mendeteksi berbagai gas berbahaya.	- Bergantung pada koneksi internet yang stabil untuk fungsi IoT. - Delay notifikasi dapat mempengaruhi respons pengguna. - Potensi keterbatasan dalam integrasi sensor tambahan.
[7]	ESP32, Arduino, sensor suhu DHT11, sensor gas MQ-2, sensor api.	- Notifikasi melalui aplikasi Telegram memungkinkan peringatan cepat. - Pemantauan real-time melalui platform web meningkatkan kontrol jarak jauh - Integrasi beberapa sensor untuk deteksi kebakaran yang lebih komprehensif. - Metode R&D memastikan pengembangan sistem yang sistematis.	- Jarak deteksi sensor api dan gas terbatas, sehingga bisa kurang efektif dalam ruangan besar. - Bergantung pada aplikasi pihak ketiga untuk notifikasi. - Delay pada sensor dapat mempengaruhi kecepatan deteksi dan respons.
[8]	NodeMCU ESP32, sensor MQ-2, DHT22, sensor api ir, relay, buzzer, solenoid valve, Android app.	- Deteksi dini dan pemadaman otomatis meningkatkan keselamatan rumah. - Notifikasi real-time melalui aplikasi Android memudahkan pengguna memantau dari jarak jauh. - Integrasi Firebase untuk sinkronisasi data real-time. - Menggunakan Pusher Beams untuk notifikasi push yang cepat.	- Bergantung pada koneksi internet untuk fungsi IoT. - Delay notifikasi dapat mempengaruhi respons cepat pengguna. - Keterbatasan dalam deteksi jarak jauh untuk sensor api dan gas.

Dari Tabel 1 dapat diketahui bahwa penelitian sebelumnya masih memiliki kekurangan seperti keterbatasan jarak deteksi sensor, delay atau waktu pengiriman data yang lama, prinsip kerja yang masih belum kompleks dan user interface masih belum friendly bagi pengguna.

Dari kekurangan penelitian tersebut diusulkan penelitian ini dengan prinsip kerja yaitu pengguna menggunakan suatu *website* sebagai monitoring kondisi ruangan dengan koneksi dari internet WiFi maka alat akan bekerja secara *real-time*, sehingga pada tampilannya menjadi lebih ramah pengguna dan mudah dipahami. ESP32 akan mengumpulkan data yang telah didapatkan oleh ketiga sensor api, asap, dan suhu, kemudian mengirimkan data tersebut ke *web server* yang telah disinkronkan. Pada saat terjadi kebakaran sistem akan memadamkan api dan arus listrik secara otomatis.

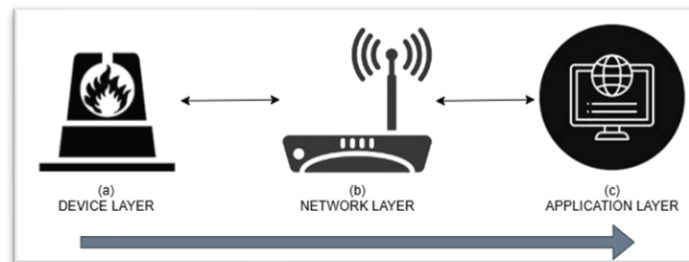
Keunggulan penelitian ini dibandingkan dengan penelitian sebelumnya adalah sebagai berikut:

1. Mudah digunakan dan lebih sederhana dibandingkan dengan penelitian sebelumnya.
2. Dapat memutus arus listrik jika terjadi korsleting dan memadamkan api otomatis, masih tetap berfungsi tanpa koneksi internet.

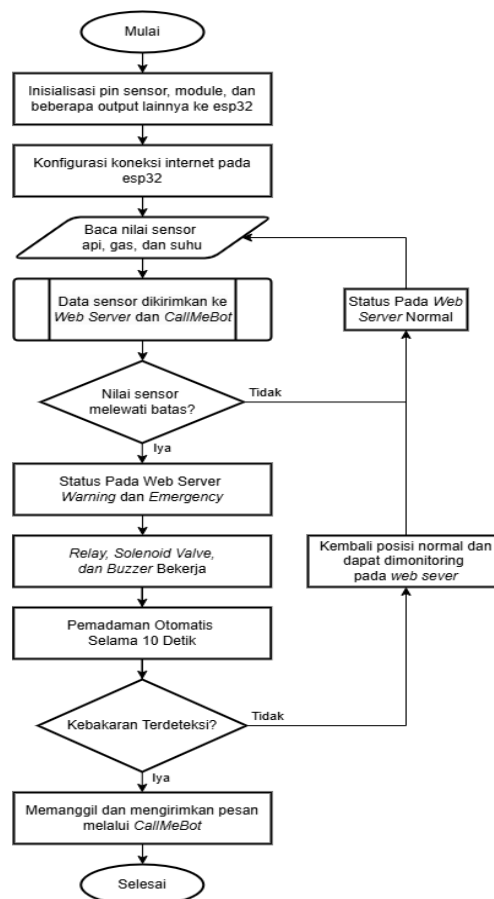
3. Dapat memanggil pemadam kebakaran secara otomatis serta notifikasi melalui pesan telepon yang memberitahukan informasi berdasarkan keinginan pengguna.
4. Dapat melakukan monitoring melalui *web server* selama ada koneksi internet yang terhubung pada ESP32.

2. METODE PENELITIAN

Perancangan arsitektur jaringan IoT pada pendeteksi kebakaran otomatis ini merupakan konsep interkoneksi jaringan antar beberapa perangkat untuk dapat saling terhubung dalam pengiriman data. Gambar 1 menunjukkan proses perancangan arsitektur jaringan untuk pendeteksi dan pemadam kebakaran otomatis. Sedangkan Proses kerja sistem berdasarkan arsitektur jaringan *internet of things* yang di rancang untuk pendeteksi dan pemadaman kebakaran otomatis ini dapat ditampilkan pada diagram alir yang ditunjukan pada Gambar 2.



Gambar 1. Arsitektur Jaringan Pendeteksi dan Pemadam Kebakaran



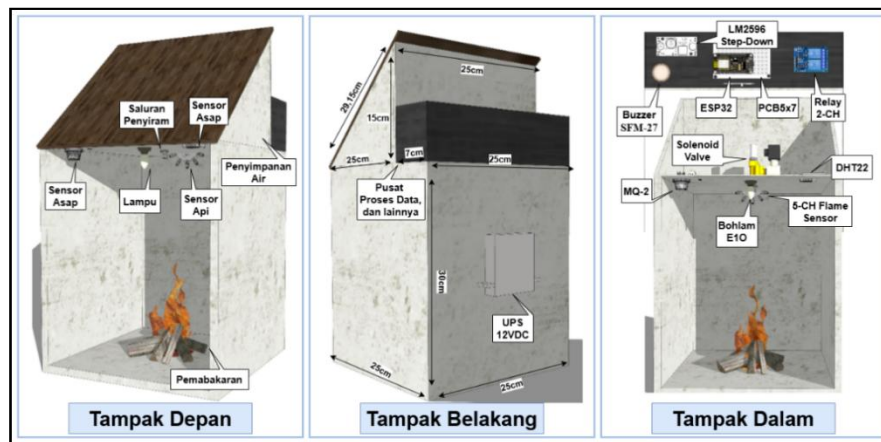
Gambar 2. Diagram Alir Alat Pendeteksi dan Pemadam Kebakaran

Arsitektur jaringan pendeteksi dan pemadaman kebakaran ini terdiri dari beberapa bagian yang saling terhubung antara lain:

1. *Device Layer*; berfungsi sebagai prototipe dari beberapa komponen penting dalam sistem pendeteksi dan pemadam kebakaran otomatis.
2. *Network Layer*; berfungsi sebagai koneksi internet menggunakan WiFi agar bisa mengirimkan data.
3. *Application Layer*; berfungsi sebagai layanan monitoring melalui *web server* serta pengiriman notifikasi seperti telepon dan pesan.

2.1 Perancangan *Hardware*

Perancangan ini mendefinisikan bentuk wujud desain arsitektur dari alat pendeteksi kebakaran yang akan dibuat dalam bentuk *box* yang terdapat komponen elektrik tersimpan didalamnya. Desain perancangan mekanik yang akan dibuat serta dikembangkan menggunakan model 3D ini terdapat gambar dari tiap-tiap posisi komponen yang diletakkan.

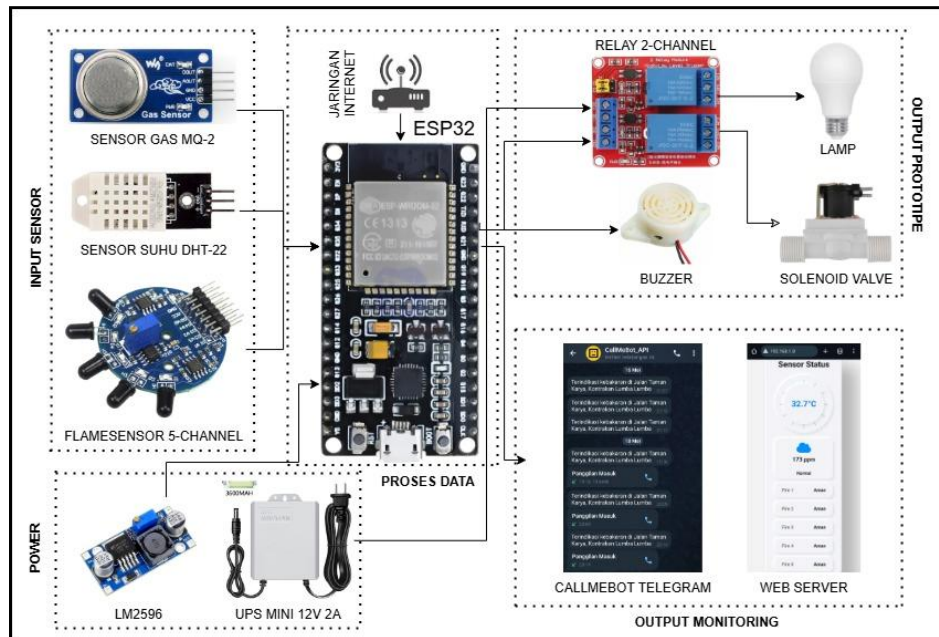


Gambar 3. Rancangan *Design Model* 3D

Gambar 3 menunjukkan hasil desain model 3D penelitian yang merupakan sebuah rumah dengan yang memiliki ruangan kecil yang berisikan beberapa komponen yang telah di sesuaikan berdasarkan nama-nama yang telah ditandai dengan keterangan pada gambar. Desain model 3D tersebut menunjukkan model untuk tampak depan dengan keterangan umum, tampak belakang keterangan ukuran dan beberapa keterangan umum dan tampak dalam dengan keterangan detail komponen yang ada.

Pada Gambar 4 dapat dilihat blok diagram dari beberapa komponen *input* dan *output* yang ada, berikut keterangan pin selain gnd dan vcc yang tersambung ke tiap bagian dari komponen tersebut pada ESP32 ke dalam pemrograman Arduino IDE [9];

- *5-Channel Flame Sensor* dengan pin analog 1 hingga 5 tersambung ke GPIO 32, 33, 25, 26, 27.
- *Gas Sensor MQ-2* dengan pin A0 tersambung ke GPIO 34.
- *Temperature Sensor DHT22* dengan pin out tersambung ke GPIO 12.
- *Buzzer* dengan pin positif ke GPIO 14.
- *Mini Lamp E10* tersambung dengan *relay* pada pin in 1 ke GPIO 2.
- *Solenoid Valve 12v dc* tersambung dengan *relay* pada pin in 2 ke GPIO 17.
- *LM2596* dengan menggabungkan vcc gnd nya pada kabel type micro untuk ke ESP32.
- *UPS Mini 12v 2a* disambung pada LM2596 dan *Relay COM Solenoid Valve*.



Gambar 4. Blok Diagram Pendeteksi dan Pemadam Kebakaran

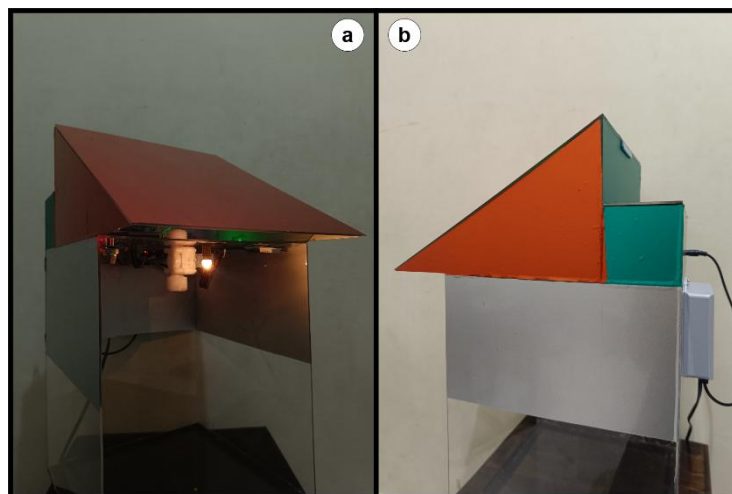
2.2 Perancangan Software

Pada perancangan perangkat lunak ini dapat meliputi dari pemrograman pada esp32 menggunakan software Arduino IDE dan pemrograman pada web server yang akan digunakan sebagai monitoring, beserta koneksi esp32 ke layanan CallMeBot Telegram. Sistem menggunakan arsitektur client-server dengan ESP32 sebagai server yang menyediakan web interface untuk monitoring *real-time*. Data sensor ditampilkan melalui dashboard web yang dapat diakses melalui jaringan lokal.

3. HASIL DAN PEMBAHASAN

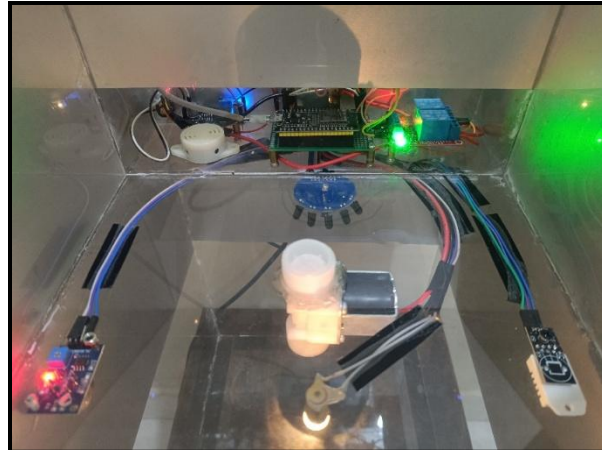
3.1 Hasil Perancangan Hardware

Hasil dari perancangan *hardware* terdiri dari prototipe rangkaian komponen yang berbentuk sebuah rumah satu ruangan. Berikut tampilan gambar secara keseluruhan dari prototipe yang telah dikerjakan.



Gambar 5. Prototipe Tampak (a) Depan dan (b) Belakang

Pada gambar 5 dapat dilihat bentuk prototipe berupa suatu ruangan dengan panjang 25cm, lebar 25cm dan tinggi 45cm. Untuk keterangan lebih jelas lainnya telah di tampilkan pada Gambar 3. Jika sistem ini diimplementasikan, dapat di letakkan seperti pada dapur dengan adanya gas kompor maupun dikamar atau ruang keluarga jika terjadi korsleting listrik yang berpotensi bisa terjadi kebakaran. Untuk posisi sensor bisa diletakkan disuatu sudut-sudut tertentu agar bisa membaca titik api asap dan suhu dengan stabil, kemudian letakkan UPS berada di dekat komponen yang dimana UPS ini berfungsi sebagai *power supply* atau daya tegangan yang bisa disimpan walau keadaan listrik telah mati, sehingga prototipe akan selalu tetap siaga. Terlihat juga ada dua bagian komponen yang bisa bekerja secara otomatis yaitu lampu sebagai indikator pemutus arus listrik dan juga ada *solenoid valve* sebagai penyiram air saat adanya api. Posisi untuk *solenoid valve* bisa berada di plafon atas rumah dengan menambahkan selang ataupun pipa air beserta *sprinkler* agar lebih baik. Kemudian posisi untuk idikator pemutus arus listrik yang menggunakan lampu, bisa diganti dengan saklar ataupun mcb yang digunakan pada ruangan tersebut.

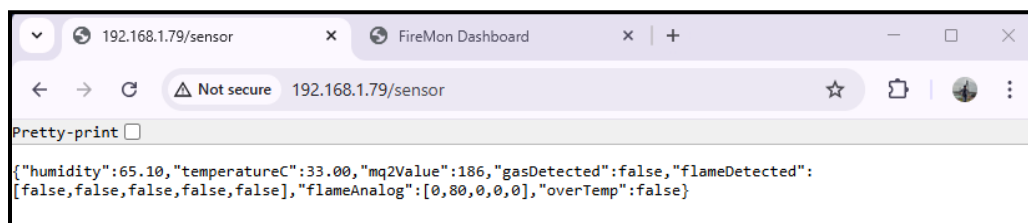


Gambar 6. Rangkaian Tampak Dalam

Pada Gambar 6 dapat dilihat tampilan prototipe dari bagian dalamnya, ada beberapa komponen bagian atap depan dan belakang. Pada bagian atap depan atas sebagai penampung air dan saluran *solenoid* sebagai penyiramnya kemudian bagian atap depan bawah terdiri dari lampu kecil, sensor api asap dan suhu. Pada bagian atap belakang atas sebagai tempat meletakkan komponen utama dari sistem ini yang terdiri dari adaptor 12v sebagai penghubung *solenoid*, mikrokontroler esp32, penurun tegangan lm2596, dan *relay*.

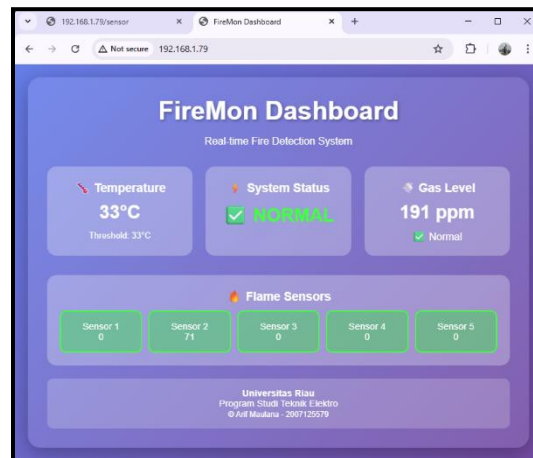
3.2 Hasil Perancangan Software

Hasil dari perancangan *software* terdiri dari tampilan *web server* serta layanan *callmebot* telegram. Berikut tampilan gambar dari kedua bagian *software* yang telah dikerjakan.

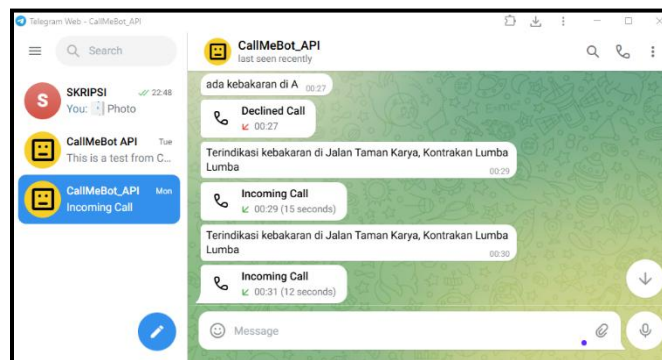


Gambar 7. Tampilan *Endpoint* dengan Format JSON

Pada Gambar 7 dapat diperhatikan bahwa tampilan *endpoint web server* dengan menambahkan */sensor* setelah IP Address hanya menampilkan mentahan dari data sensor berupa nilai analog ataupun digitalnya.

Gambar 8. Tampilan *Web Server Dashboard*

Pada Gambar 8 dapat diperhatikan bahwa tampilan *endpoint web server* hanya dengan menggunakan IP Address tanpa adanya /sensor akan menampilkan desain dari web server beserta data sensor atau disebut dengan *dashboard* yang berisikan informasi status peringatannya.

Gambar 9. Tampilan Telepon dan Pesan dari *CallMeBot* Telegram

Pada Gambar 9. dapat diperhatikan bahwa jika sensor terdeteksi melewati ambang batas yang telah ditentukan maka perintah sistem notifikasi telepon dan pesan akan diaktifkan kemudian akan mengirimkannya ke akun telegram pengguna bisa ke pemilik rumah ataupun langsung ke pemadam kebakaran.

3.3 Hasil Pengujian Sensor Kebakaran

Pengujian sensor dilakukan bersamaan dengan keluarannya yakni sistem pemadam api menggunakan *solenoid valve*, sistem peringatan dini berupa suara menggunakan *buzzer* dan sistem pemadaman listrik otomatis menggunakan *relay* yang tersambung dengan indikator lampu. Pengujian dilakukan menggunakan beberapa objek berupa kertas, korek api, sumbu kompor, dan rokok. Ambang batas atau dikenal dengan *threshold* pada tiap sensor yang akan digunakan yakni; sensor api (2000nm), sensor asap (800ppm), dan sensor suhu (34°C).

Tabel 1. Hasil Pengujian Sensor Kebakaran
Sensor

<i>Object</i>	<i>5-Ch Flame (nm)</i>	<i>MQ-2 (ppm)</i>	<i>DHT22 (°C)</i>	<i>Buzzer (On/Off)</i>	<i>Solenoid Valve (On/Off)</i>	<i>Lamp (On/Off)</i>
Kertas	1444	1273	34.1	On	On	Off
Korek Api	3206	1171	34.6	Off	Off	On

Sumbu Kemplor	4095	2361	42.2	On	On	Off
Rokok	91	1182	33	Off	Off	On

Dapat dilihat pada hasil pengujian yang dilakukan yang ditampilkan pada Tabel 2 ketika pengujian dilakukan menggunakan kertas terbakar saat api pada kertas sudah mulai mengecil hingga pembacaan apinya lebih besar dari korek api yang jarak dan besar apinya lebih dekat dengan sensornya. Pada asap keduanya (kertas dan korek api) hampir sama mendekati karena pada kertas mengandung senyawa bakar karbon dioksida (terbakar sempurna) sedangkan korek api mengandung gas yang bersenyawa hidrogen ataupun butana. Untuk suhu hampir sama karena korek api dilakukan dengan jarak dan besar api yang dekat dengan sensor sedangkan pada kertas api tidak terlalu besar dan jarak api tidak dekat dengan sensornya.

Pengujian menggunakan sumbu kompor dan rokok terdapat perbedaan yang besar karena rokok tidak menghasilkan api yang dapat dilihat secara jelas oleh sensor sedangkan sumbu kompor menghasilkan api yang lebih besar ketika dibakar karena ditambahkan bahan bakar minyak tanah hingga bisa dibaca dengan sangat jelas oleh sensor. Pada asap pula sumbu kompor menghasilkan senyawa karbon monoksida ataupun metana (terbakar tidak sempurna) karena terdapat asap hitam pekat tebal saat dibakar sedangkan rokok menghasilkan senyawa tar ataupun karbon monoksida terdapat asap yang tidak terlalu tebal (tipis).

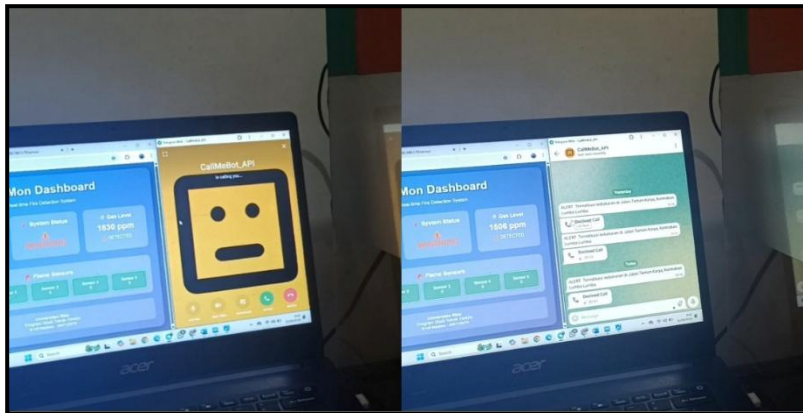
3.4 Hasil Pengujian Notifikasi Kebakaran

Pengujian notifikasi terdapat dua bagian yakni notifikasi secara *real-time* terhubung dengan *web server* berdasarkan dengan nilai sensor beserta status sistemnya dan satu lagi notifikasi pesan telepon otomatis menggunakan layanan *callmebot telegram* yang dibisa ditujukan ke pihak pemadam kebakaran ataupun pemilik rumah. Pengujian notifikasi ini dilakukan bersamaan dengan pengujian sensor sebelumnya, namun pengujian tersebut dilakukan selama 5 detik jika sistem masih mendeteksi api lebih dari waktunya maka sistem pesan telepon otomatis akan terkirim ke pihak pemadam kebakaran ataupun pemilik rumah.

Tabel 2. Hasil Pengujian Notifikasi Kebakaran

Objek	Sensor			Web Server System Status	CallMeBot Telegram	
	F	G	T		Pesan	Telepon
Kertas	1	1	1	Emergency	Tidak	Tidak
Korek Api	1	1	0	Warning	Tidak	Tidak
Sumbu Kemplor	1	0	1	Emergency	Ada	Ada
Rokok	0	1	0	Normal	Tidak	Tidak

Berdasarkan pada Tabel 3 hasil dari pengujian notifikasi yang telah dilakukan terdapat 4 macam objek digunakan dalam pengujian sensor dan notifikasinya. Pada pengujian pertama menggunakan kertas terdapat peringatan *emergency*, yang dimana api asap dan suhu terdeteksi semuanya namun api hanya terbakar selama 4 detik sehingga sistem pesan telepon otomatis tidak aktif. Pada pengujian kedua menggunakan korek api terdapat peringatan *warning*, yang dimana hanya api dan gas yang terdeteksi sehingga sistem tidak mendeteksi kebakaran dan pesan telepon otomatis tidak terkirim. Pada pengujian keempat juga saat menggunakan rokok hanya mendeteksi asap sehingga sistem tidak mendeteksi terjadinya kebakaran dan pesan telepon otomatis juga tidak dikirimkan. Ketiga pengujian ini dilakukan tidak terdapat pengiriman sistem pesan telepon otomatis hanya ada satu yang dilakukan pengujian hingga sistem pesan telepon otomatis dapat terkirim yakni saat percobaan ketiga yang menggunakan sumbu kompor dapat dilihat pada gambar berikut.



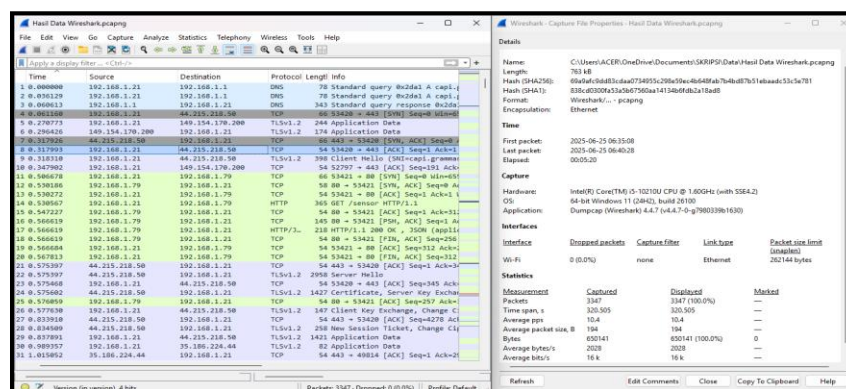
Gambar 10. Notifikasi Pesan Telepon Pengujian Sumbu Kompur

Pada pengujian ketiga menggunakan sumbu kompor yang ditambahkan bahan bakar (minyak tanah) agar api tahan lama pada sumbu, saat pendeteksi kebakaran terjadi peringatan *emergency* dan api tidak dapat padam lebih dari 5 detik sehingga sistem notifikasi pesan telepon terjadi seperti pada gambar 10 yang dimana layanan *callmebot* akan melakukan pengiriman pesan terlebih dahulu sebelum menelpon, saat berada dipanggilan telepon penyampaian pesan akan sama dengan penulisan pesan yang telah dikirimkan.

3.5 Hasil Pengukuran QoS Sistem Pendeteksi Kebakaran

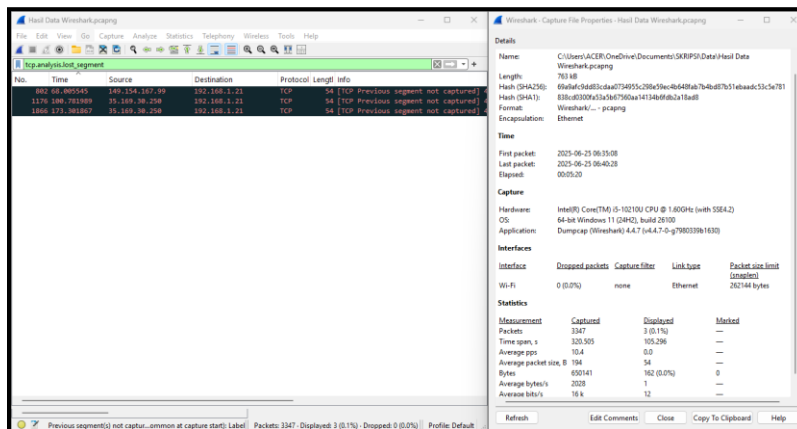
Dalam proses berlangsungnya sistem pendeteksi kebakaran ini terdapat beberapa parameter QoS yang ada seperti *throughput*, *delay*, dan *packet loss*. Maka pada sub ini akan dilakukan pengukurannya berdasarkan notifikasi real-time dari *web server* walaupun pengiriman data yang dilakukan oleh *web server* sangat cepat namun pasti ada sedikit *delay* ataupun paket yang tidak terkirim. Dan juga dilakukan pengukuran notifikasi dari pesan telepon yang ada pada layanan *callmebot telegram*. Pengukuran ini dilakukan menggunakan *software wireshark* [10].

Tahap awal yang dilakukan dengan memulai *capture* koneksi yang tersambung dengan perangkat IoT yakni wifi, lakukan *capture* selama proses pembacaan sensor hingga pesan telepon telegram aktif. Sebelum memulai lebih dalam cari tau dahulu IP dari *webserver* dan telegram yang akan dibaca pada *wireshark*. Pada web server udah didapatkan dari *code* awal program dengan IP 192.168.1.79 sedangkan pada telegram lakukan pengecekan ping telegram.org di *command prompt* atau CMD hingga mendapatkan IP 149.154.167.99, selanjutnya lakukan pengecekan secara berkala pada rekaman *capture* yang telah didapatkan.

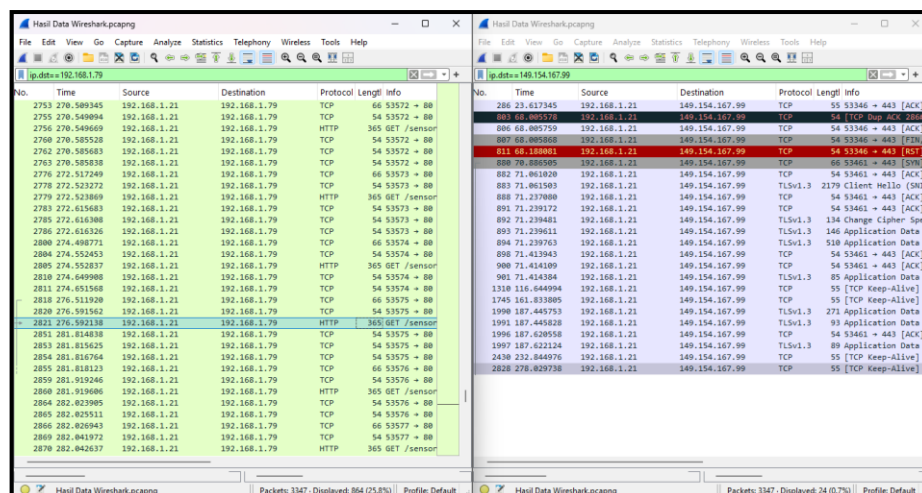


Gambar 11. Hasil Keseluruhan dari Capture Wireshark

Pada gambar 11 terdapat hasil *capture* dari keseluruhan pengujian sensor dalam *webserver* hingga aktifnya sistem pesan telepon otomatis layanan *callmebot*. Dapat dilihat paket yang dikirim dan diterima sebanyak 3347 *packets*, untuk *throughput* nya bisa dilakukan perhitungan manual dengan rumus ataupun pada *wireshark* telah ditampilkan memiliki *bandwidth* 16k bits/s.

Gambar 12 Hasil Keseluruhan *Capture Packet Loss*

Pada gambar 12 terdapat hasil *capture packet loss* dari keseluruhan pengujian dengan memasukkan perintah “tcp.analysis.lost_segment” pada filter yang di *Wireshark*. Walaupun *packet loss* dapat dilakukan perhitungan manual dengan rumus, namun pada tampilan detail *capture* bisa diperhatikan bahwa paket yang hilang hanya ada sekitar 0,1%. Dari hasil yang didapatkan menunjukkan hasil masih memenuhi untuk diterima VoIP (*Voice over IP*) dalam artian bisa melakukan pesan dan panggilan.

Gambar 13 Filter IP.DST *Webserver* dan *Telegram*

Pada gambar 13 dapat dilihat hasil *delay request* yang menggunakan filter `ip.dst==192.168.1.79` dan `ip.dst==149.154.167.99` dari *webserver* dan *telegram*. Sehingga pada bagian ini dilakukan perhitungan *delay request*nya berdasarkan *convert* ke file csv agar bisa dihitung dengan excel.

No.	Time Webserver	Time 1	Time 2	Delay	No.	Time Telegram	Time 1	Time 2	Delay
11	0.506678	0.5067	0.5303	0.024	286	71.61725	71.617	68.006	44.888
13	0.530272	0.5303	0.5306	3E-04	803	68.00558	68.006	68.006	0.0002
14	0.530567	0.5306	0.5667	0.036	806	68.00576	68.006	68.006	0.0001
19	0.566684	0.5667	0.5678	0.001	807	68.00587	68.006	68.188	0.1822
20	0.567813	0.5678	2.5051	1.941	811	68.18808	68.188	70.887	2.6984
78	2.509069	2.5091	2.5877	0.079	880	70.88651	70.887	71.061	0.1745
81	2.587731	2.5877	2.588	2E-04	882	71.06102	71.061	71.062	0.0005
82	2.587955	2.588	2.7	0.112	883	71.0615	71.062	71.237	0.1756
87	2.699971	2.7	2.7015	0.002	888	71.23708	71.237	71.239	0.0021
88	2.701527	2.7015	4.5133	1.812	891	71.23917	71.239	71.239	0.0003
102	4.513285	4.5133	4.5177	0.004	892	71.23948	71.239	71.24	0.0001
104	4.517708	4.5177	4.5183	6E-04	893	71.23961	71.24	71.24	0.0002
105	4.518329	4.5183	4.625	0.107	894	71.23976	71.24	71.414	0.1742
110	4.624995	4.625	4.6252	2E-04	898	71.41294	71.414	71.414	0.0002
111	4.625233	4.6252	6.5054	1.88	900	71.41411	71.414	71.414	0.0003
124	6.505391	6.5054	6.5606	0.055	901	71.41438	71.414	116.64	45.231
126	6.560602	6.5606	6.5612	6E-04	1310	116.645	116.64	161.83	45.189
127	6.561242	6.5612	6.6574	0.096	1745	161.8338	161.83	187.45	25.612
132	6.65743	6.6574	6.6584	0.001	1990	187.4458	187.45	187.45	0.0001
133	6.658435	6.6584	8.5043	1.846	1991	187.4458	187.45	187.62	0.1747
136	8.504341	8.5043	8.5099	0.006	1996	187.6206	187.62	187.62	0.0016

Gambar 14. Hasil Perhitungan Delay Request dengan Excel

Berdasarkan Gambar 14. proses perhitungan awal *delay* hanya mengurangi time 2 dengan time 1, untuk total delay ditambahkan keseleruhan bagian dan untuk nilai rata-rata total *delay* nya di bagi dengan jumlah *packet* sehingga menghasilkan nilai akhir dalam satuan *second*(detik), konversi nilai *delay* tersebut ke satuan ms dengan mengkalikan 1000. Hasil dari kedua ip.dst ataupun *delay request* nya adalah 95,60758918ms dan 89,96194943ms.

No.	Time	Source	Destination	Protocol	Length	Info
2801	274.592353	192.168.1.79	192.168.1.21	TCP	58	58 → 53574
2806	274.583884	192.168.1.79	192.168.1.21	TCP	54	58 → 53574
2807	274.649797	192.168.1.79	192.168.1.21	TCP	145	80 → 53574
2808	274.649797	192.168.1.79	192.168.1.21	HTTP/3.	222	HTTP/1.1 2
2809	274.649797	192.168.1.79	192.168.1.21	TCP	54	80 → 53574
2813	274.748034	192.168.1.79	192.168.1.21	TCP	54	80 → 53574
2819	276.591483	192.168.1.79	192.168.1.21	TCP	58	80 → 53575
2822	276.835628	192.168.1.79	192.168.1.21	TCP	54	80 → 53575
2849	281.814428	192.168.1.79	192.168.1.21	TCP	145	80 → 53575
2850	281.814776	192.168.1.79	192.168.1.21	HTTP/3.	218	HTTP/1.1 2
2852	281.815568	192.168.1.79	192.168.1.21	TCP	54	80 → 53575
2857	281.914874	192.168.1.79	192.168.1.21	TCP	54	80 → 53575
2858	281.915158	192.168.1.79	192.168.1.21	TCP	58	80 → 53576
2861	282.823759	192.168.1.79	192.168.1.21	TCP	145	80 → 53576
2862	282.823759	192.168.1.79	192.168.1.21	HTTP/3.	218	HTTP/1.1 2
2863	282.823759	192.168.1.79	192.168.1.21	TCP	54	80 → 53576
2867	282.841746	192.168.1.79	192.168.1.21	TCP	58	80 → 53576
2868	282.841746	192.168.1.79	192.168.1.21	TCP	58	80 → 53577
2871	282.885588	192.168.1.79	192.168.1.21	TCP	54	80 → 53577
2872	282.130283	192.168.1.79	192.168.1.21	TCP	145	80 → 53577
2873	282.130283	192.168.1.79	192.168.1.21	HTTP/3.	217	HTTP/1.1 2
2874	282.130283	192.168.1.79	192.168.1.21	TCP	54	80 → 53577
2877	282.130283	192.168.1.79	192.168.1.21	TCP	54	80 → 53577
2879	282.532257	192.168.1.79	192.168.1.21	TCP	58	80 → 53578
2882	282.555556	192.168.1.79	192.168.1.21	TCP	54	80 → 53578
2883	282.641592	192.168.1.79	192.168.1.21	TCP	145	80 → 53578
2884	282.641592	192.168.1.79	192.168.1.21	HTTP/3.	217	HTTP/1.1 2
2886	282.641838	192.168.1.79	192.168.1.21	TCP	54	80 → 53578
2889	282.695458	192.168.1.79	192.168.1.21	TCP	54	80 → 53578
2895	284.509887	192.168.1.79	192.168.1.21	TCP	58	80 → 53579
2898	284.570480	192.168.1.79	192.168.1.21	TCP	145	80 → 53579

Gambar 15. Filter IP.SRC Webserver dan Telegram

Pada gambar 15 dapat dilihat hasil *delay respond* yang menggunakan filter `ip.dst==192.168.1.79` dan `ip.dst==149.154.167.99` dari *webserver* dan *telegram*. Sehingga pada bagian ini dilakukan perhitungan *delay request*nya berdasarkan *convert* ke file csv agar bisa dihitung dengan *excel*.

[illegible]

Gambar 16. Hasil Perhitungan *Delay Respond* dengan Excel

Pada Gambar 16 perhitungan *delay respond* mempunyai cara menghitung yang sama dengan *delay request*, sehingga hasil dari kedua ip.src ataupun *delay respond* nya adalah 95,06614461ms dan 76,00959068ms.

Berdasarkan dari beberapa perhitungan sebelumnya menggunakan *wireshark*, bisa juga dilakukan perhitungan manual dalam notifikasi dari *webserver* dan pesan teleponnya. Dalam *webserver delay* seharusnya sangat kecil dan hanya terjadi lama pengiriman jika jaringan tidak stabil ataupun menurun, sedangkan notifikasi pesan telepon menggunakan layanan *callmebot* terdapat *delay* dalam beberapa detik. Hal tersebut dikarenakan ada nya faktor menunggu proses sistem pendeteksi kebakaran terindikasi lebih dari 5 detik, saat peringatan masih terjadi diatas 5 detik maka di saat itu akan dihitung proses pengirmanan data pesan dan teleponnya, bisa menggunakan rekaman video ataupun *stopwatch* agar bisa melihat waktu delaynya, normalnya dalam penelitian ini penulis menghitung proses delay dari *callmebot* sekitar 8 hingga 13 detik.

4. KESIMPULAN

Berdasarkan dari hasil penelitian dan pengujian yang telah dilakukan, dapat disimpulkan bahwa hasil prototipe ada dua perangkat yaitu; Pada perangkat keras sebagai sistem pendeteksi dan pemadam kebakaran otomatis terdapat ESP32, tiga sensor (api 5 chanel, dht22, dan mq-2), *relay*, *solenoid valve*, *buzzer*, lampu kecil, UPS, dan LM2596. Sedangkan pada perangkat lunak sebagai monitoring notifikasi *real-time* terdiri dari *web server* dan *callmebot telegram*. Hasil pengukuran tegangan dan nilai eror yang dilakukan pada beberapa komponen berkisar antara 1,6% hingga 16,3% sehingga selama proses perakitan rangkaian semua komponen berjalan dengan baik dengan mengikuti prosedur dari *datasheet*.

Hasil pengujian keberhasilan sensor dan notifikasi kebakaran berfungsi dengan baik pada sistem pendeteksi sensor kebakaran hingga pemadaman api dan listrik otomatisnya saat kertas terbakar. Saat api tidak padam lebih dari 5 detik sistem pesan telepon akan aktif seperti yang sudah diujikan menggunakan sumbu kompor. Hasil perhitungan QoS yang ada pada pengiriman data, untuk *packet loss* hanya 0,1% kemudian ada lagi dua bagian yakni *delay request* dan *delay respond*, untuk nilai *webserver* terdiri 95,60758918ms dan 95,06614461ms. Pada telegram nilainya sebesar 89,96194943ms dan 76,00959068ms.

❖ **DAFTAR PUSTAKA**

- [1] NFPA. (2021). *First Revision No. 36-NFPA 914-2021 [Detail]* (Issue 36). https://www.nfpa.org/api/files?path=%2Ffiles%2FAboutTheCodes%2F914%2F914_F2022_CUL_AAA_FD_FRStatements.pdf
- [2] Budiyantri, R. T. (2021). *Buku Ajar Internet of Things*. CV. Asta Karya Kreatifa Media. https://dokpak.undip.ac.id/10970/1/Buku_Ajar_Internet_of_Things.pdf
- [3] A. F. Amali. (2020). *Sistem Deteksi Kebakaran Berbasis Internet of Things (IoT) Dengan Perangkat Arduino, Politeknik Negeri Bali*.

- http://repository.pnb.ac.id/id/eprint/4361%0Ahttp://repository.pnb.ac.id/4361/1/RAMA_36304_181534404_2_artikel.pdf
- [4] Wijaya, A. C., Budiyo, U., Juliasari, N., & Amini, S. (2023). Aplikasi Android Untuk Pendeteksi Kebakaran Berbasis Internet of Things Menggunakan Mikrokontroler Nodemcu Esp8266. *Prosiding Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)*, 2(1), 466–473
- [5] Cahyadi, H., Mirza, D. Y., & Laila, E. (2022). Rancang Bangun Alat Pendeteksi Kebakaran Menggunakan Flame Sensor dan Sensor Asap Berbasis Arduino. *Jurnal Laporan Akhir Teknik ...*, 2(1), 60–69. <https://jurnal.polsri.ac.id/index.php/JLATK/article/view/6193%0Ahttps://jurnal.polsri.ac.id/index.php/JLATK/article/download/6193/2276>
- [6] Ferdiansyah, F., Suhradi Rahmat, R., Education Park, J., Ki Hajar Dewantara, J., & Cikarang, N. (2022). Alat Pendeteksi Kebakaran dan Pemadam Api Otomatis Menggunakan Kontrol Arduino. *Journal of Mechanical Engineering and Mechatronics*, 7(2), 77–89.
- [7] Ainun Najib, M., Pendeteksi Bencana Kebakaran Menggunakan ESP, S., Arduino Berbasis WEB, D., Sulartopo, S., Sasmoko, D., Danang, D., & Saufik Suasana, I. (2024). *Neptunus : Jurnal Ilmu Komputer Dan Teknologi Informasi*. 1. <https://doi.org/10.61132/neptunus.v2i1.62>
- [8] Bagus Saputra, D., & Hidayati, R. (2024). Sistem Deteksi Dini dan Pemadaman Kebakaran Otomatis di Rumah Berbasis IoT Menggunakan NodeMCU ESP32. *Jurnal Sistem Komputer Dan Informatika (JSON)* Hal: 581–, 589(3), 581–589. <https://doi.org/10.30865/json.v5i3.6974>
- [9] Espressif System. (2024). *ESP32 Series Datasheet* (Vol. 24, Issue 1). https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [10] Hasbi, M., & Saputra, N. R. (2021). Analisis Quality of Service (Qos) Jaringan Internet Kantor Pusat King Bukopin Dengan Menggunakan Wireshark. *Universitas Muhammadiyah Jakarta*, 12(1), 1–7. <https://jurnal.umj.ac.id/index.php/just-it/article/view/13596>