

## RANCANG BANGUN SISTEM PENDETEKSI KEBAKARAN BERBASIS IOT

Andica Setya Perdana<sup>1</sup>, Lilik Anifah<sup>2</sup>, Rifqi Firmansyah<sup>3</sup>, Miftahur Rohman<sup>4</sup>

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya

email : [andicasetya.19023@mhs.unesa.ac.id](mailto:andicasetya.19023@mhs.unesa.ac.id), [lilikanifah@unesa.ac.id](mailto:lilikanifah@unesa.ac.id),[rifqifirmansyah@unesa.ac.id](mailto:rifqifirmansyah@unesa.ac.id), [miftahurrohman@unesa.ac.id](mailto:miftahurrohman@unesa.ac.id)**Abstract**

*The losses caused by fires themselves are often enormous, mainly because property owners are usually slow to respond. Therefore, the author wants to design an IoT-based fire detection device that users can access by opening a website. Relying on several sensors that read temperature (DHT22), fire (flame sensor), and carbon monoxide gas (MQ-7), then Arduino UNO and ESP32 as microcontrollers. ThingsBoard is used here as a database medium. A buzzer is also added to provide security and immediate warning when signs of fire are detected. In this study, the author chose the experimental research category, which is a method that aims to test the effect of one variable on another variable or to test the cause-and-effect relationship between one variable and another. The experimental research method has clear differences compared to other research methods, namely the control of research variables and the treatment of the experimental group (ASRIN, 2022). The device designed by the author works well. The Arduino UNO microcontroller can receive data from the sensor and send the data to the ESP32. The data is then sent to ThingsBoard and the buzzer is activated if the ESP32 detects a fire. The error found in the DHT22 when compared to the reference used, namely a thermometer, has a relatively small error value ranging from 0.34% to 4.89%. The buzzer delay time is 0.4 seconds for the fastest delay and 1.98 seconds for the longest time. For the dashboard delay time, it is 0.4 seconds for the fastest time and 2.0 seconds for the longest time.*

**Article History**

Submitted: 1 Februari 2026

Accepted: 4 Februari 2026

Published: 5 Februari 2026

**Key Words**

MQ-7 sensor, DHT22 sensor, ESP32, Arduino UNO, IoT-based fire detector.

**Abstrak**

Kerugian yang diakibatkan oleh kebakaran itu sendiri tidak jarang sangat besar, hal ini dikarenakan pemilik properti biasanya lambat dalam melakukan penanganan. Maka dari itu penulis ingin merancang sebuah alat pendeteksi kebakaran yang berbasis Iot, dan dapat diakses oleh pengguna dengan membuka suatu website. Dengan mengandalkan beberapa sensor yang membaca suhu (DHT22), api (*flame sensor*), dan gas karbon monoksida (MQ-7), kemudian Arduino UNO dan ESP32 sebagai mikrokontroler. ThingsBoard disini digunakan sebagai media database. *Buzzer* juga ditambahkan untuk memberikan keamanan, dan peringatan langsung apabila terdeteksi adanya indikasi kebakaran. Pada penelitian ini penulis memilih kategori penelitian eksperimen, yaitu metode yang bertujuan untuk menguji pengaruh suatu variabel terhadap variabel lain atau menguji bagaimana hubungan sebab akibat antara satu variabel dengan variabel lainnya. Metode penelitian eksperimen memiliki perbedaan yang jelas disbanding dengan metode penelitian lainnya, yaitu ada pengontrolan terhadap variabel penelitian dan adanya pemberian perlakuan terhadap kelompok eksperimen (ASRIN, 2022). Alat yang telah dirancang oleh penulis dapat bekerja dengan baik. Dimana mikrokontroler Arduino UNO dapat menerima data dari sensor dan mengirimkan data tersebut menuju ESP32. Kemudian data dikirim menuju ThingsBoard dan *buzzer* diaktifkan apabila terdapat indikasi adanya kebakaran oleh ESP32. error yang terdapat pada DHT22 jika dibandingkan dengan referensi yang digunakan yaitu *thermometer* memiliki nilai error yang relatif kecil berkisar antara 0,34% sampai 4,89%. Waktu delay *buzzer* adalah 0,4 detik untuk delay tercepat dan 1,98 detik untuk waktu yang paling lama. Untuk Waktu delay dashboard adalah 0,4 detik untuk waktu tercepat dan 2,0 detik untuk waktu yang terlama.

**Sejarah Artikel**

Submitted: 1 Februari 2026

Accepted: 4 Februari 2026

Published: 5 Februari 2026

**Kata Kunci**

Sensor MQ-7, Sensor DHT22, ESP32, Arduino UNO, pendeteksi kebakaran berbasis IoT.

## PENDAHULUAN

Kebakaran merupakan salah satu bencana yang sering terjadi dan menimbulkan kerugian besar, baik dari segi materi maupun keselamatan manusia. Kebakaran bisa disebabkan oleh korsleting listrik, kebocoran gas berbahaya seperti karbon monoksida, kelalaian manusia, atau suhu lingkungan yang meningkat secara drastis. Dalam banyak kasus, deteksi kebakaran yang terlambat menyebabkan kerusakan yang lebih parah karena tidak ada peringatan dini untuk mengambil tindakan segera.

Kemajuan teknologi dan mikrokontroler telah membuka peluang untuk merancang sistem pendeteksi kebakaran yang lebih efektif dan mudah digunakan oleh penggunanya. Sistem IoT memungkinkan perangkat sensor untuk membaca data lingkungan secara real-time dan mengirimkannya melalui jaringan internet ke platform monitoring atau aplikasi notifikasi. Dalam hal ini, mikrokontroler seperti ESP32 banyak digunakan sebagai pusat pengolahan data karena kemampuannya dalam konektivitas Wi-Fi serta integrasi sensor secara langsung, sehingga memudahkan pengembangan sistem deteksi kebakaran modern berbasis IoT. Penelitian sebelumnya menunjukkan bahwa sistem deteksi kebakaran berbasis IoT dengan mikrokontroler ESP32 mampu memberikan notifikasi secara real-time saat deteksi api atau gas berbahaya terjadi, sehingga meningkatkan respons terhadap potensi kebakaran (Muhammad Ravly, 2025).

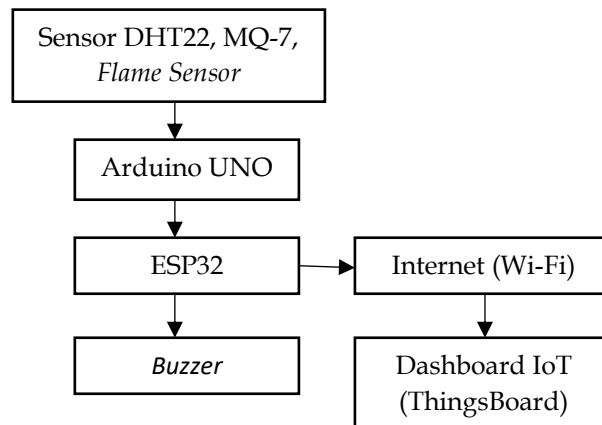
Selain sensor api (flame sensor), penggunaan sensor gas seperti MQ-7 atau MQ-2 sangat penting untuk mendeteksi asap atau kadar gas berbahaya di lingkungan. Sensor gas ini mampu mendeteksi gas karbon monoksida yang merupakan salah satu indikator awal terjadinya kebakaran. Penelitian lain juga menggunakan kombinasi sensor gas dan sensor suhu untuk meningkatkan kemampuan deteksi dini terhadap kebakaran dengan memanfaatkan platform IoT untuk pemantauan jarak jauh (Tika Hafzara Siregar, 2021).

Adapun sensor suhu dan kelembapan seperti DHT22 juga berperan penting dalam sistem pintas deteksi kebakaran karena kenaikan suhu yang signifikan sering kali mendahului terjadinya api yang nyata. Penggabungan berbagai sensor ini dalam satu sistem pendeteksi kebakaran mampu meningkatkan akurasi dan mengurangi jumlah false alarm sehingga lebih efektif dalam mendeteksi kondisi berbahaya sejak dini (Daffa Ebtra Satria, 2025).

Berdasarkan penjelasan di atas, maka dibutuhkan suatu alat pendeteksi kebakaran berbasis mikrokontroler dan IoT yang menggunakan sensor-sensor tersebut serta mampu memberi peringatan secara lokal maupun online. Alat yang dirancang dalam penelitian ini menggunakan mikrokontroler Arduino UNO sebagai pengolah sinyal sensor serta ESP32 sebagai modul komunikasi dan IoT untuk penyampaian data.

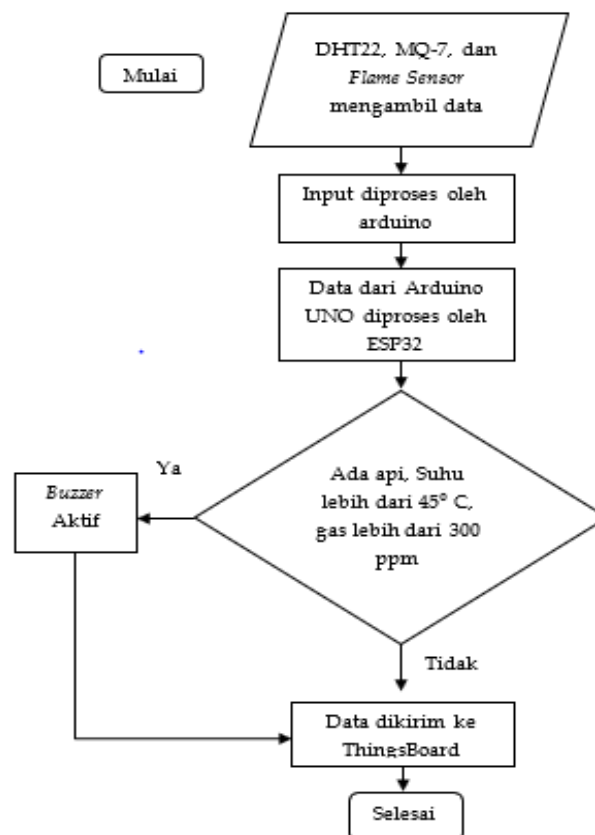
## METODE

Tujuan alat ini diciptakan adalah untuk menguji alat pendeteksi kebakaran yang penulis rancang dengan mengandalkan data yang didapat. Yang jadi perhatian dalam perancangan alat ini adalah tingkat kelembapan, suhu, potensi api, dan kandungan gas yang terkandung di udara. Variabel yang disebutkan di atas akan masuk melalui sensor, kemudian mikrokontroler akan menerima data dari output sensor dan mengunggah datanya ke internet. Data yang diunggah di dashboard IoT, serta buzzer sebagai alarm utama akan dinilai tingkat akurasi dan kecepatan responnya.



Gambar 1 Blok Diagram

Pada gambar 1 diatas, dapat dilihat sistem berjalan dengan menggunakan ketiga sensor tersebut sebagai pembaca kondisi di sekitarnya. Sensor api akan mendeteksi ada atau tidaknya api, sensor suhu DHT22 yang digunakan akan membaca perkiraan suhu disekitarnya, sensor gas MQ-7 akan membca tingkat kandungan gas khususnya gas CO pada udara, hasil baca sensor kemudian akan dikirim menuju Arduino Uno. EPS32 kemudian akan melakukan serial komunikasi untuk bertukar data dengan Arduino UNO, yang mana data tersebut akan dikirim menuju *database* ThingsBoard. Sebuah *buzzer* juga ditambahkan sebagai sistem peringatan utama apabila ditemukan beberapa indikasi seperti. Suhu yang terbaca oleh sensor DHT22 naik hingga mencapai 45 derajat *celcius*, kandungan gas yang terbaca oleh sensor gas MQ-7 mencapai 300ppm, dan ada atau tidaknya api yang terbaca oleh sensor api pada sistem.



Gambar 2 Flowchart Sistem

Pada gambar 2 diatas, dijelaskan bahwa proses awalnya dimulai dari sensor MQ-7, DHT22, dan *Flame sensor* menerima data dari lingkungan di sekitarnya. Kemudian data dari sensor diolah oleh Arduino UNO dan ESP32 akan mengirim datanya ke ThingsBoard. Apabila ditemukan suatu indikasi terjadinya kebakaran seperti suhu dan kelembapan yang naik drastis, terdeteksinya api, dan kandungan gas monoksida di dalam ruangan naik maka buzzer akan menyala. Sedangkan apabila tidak ditemukan salah satu indikasi yang telah disebutkan di atas. Maka *buzzer* akan berada di posisi *stand by*.

Pengujian akurasi suhu sistem dilakukan untuk mengevaluasi akurasi pengukuran sistem. Pengujian dilakukan sebanyak 30 kali percobaan dengan suhu yang berbeda—beda. Sensor DHT22 dan sensor suhu *thermometer* diletakkan di lokasi yang sama untuk memastikan kondisi pengukuran yang seragam. Nilai error rata-rata digunakan sebagai indikator tingkat ketelitian sistem. Akurasi sistem dihitung dengan membandingkan nilai suhu yang diukur oleh sistem dengan nilai dari referensi *thermometer* ruangan menggunakan persamaan persentase error, yaitu

$$Error(\%) = \frac{|D_{sensor} - D_{ref}|}{D_{ref}} \times 100\% \dots \dots \dots (1)$$

Dimana:

Error (%) = Kesalahan sensor dari sistem

$D_{sensor}$  = Data dari sensor sistem

$D_{ref}$  = Data dari pembanding

Pengujian waktu respon dilakukan untuk mengevaluasi kemampuan sistem dalam memberikan peringatan secara real-time. Waktu respon dihitung sebagai selisih antara waktu ketika indikasi kebakaran terdeteksi dan waktu ketika buzzer aktif, yang dinyatakan sebagai

$$T_{respon} = T_{buzzer} - T_{deteksi} \dots \dots \dots (2)$$

Dimana:

$T_{respon}$  : waktu respon sistem

$T_{buzzer}$  : waktu saat buzzer aktif

$T_{deteksi}$  : waktu saat terdeteksi

## HASIL DAN PEMBAHASAN

Untuk mengatur suhu disekitar sensor penulis menggunakan blower uap yang disesuaikan jaraknya hingga titik tertentu sehingga didapati hasil yang dikehendaki oleh penulis. Sensor suhu pada *thermometer* diletakkan tepat disebelah sensor suhu alat DHT22. Nilai yang keluar dari hasil baca sensor suhu DHT22 disesuaikan dengan daftar yang telah ditentukan. Kemudian membandingkan nilai yang terbaca oleh sensor DHT22 dengan nilai yang terbaca oleh *thermometer*. Untuk contoh kita mencari suatu nilai misalnya 42,0 derajat *celcius* untuk DHT22, dengan cara mengatur jarak antar sensor dengan corong blower uap. Apabila nilai yang terbaca oleh sensor suhu alat DHT22 sudah stabill di sekitar angka 42 derajat *celcius*, maka kita akan mengamati nilai yang terbaca oleh *thermometer*.





Gambar 3 Pengujian akurasi sensor suhu

Dari hasil pengujian ini, dapat diketahui besarnya kesalahan pengukuran (error) yang dilakukan oleh sistem dibandingkan dengan alat ukur referensi. Didapati bahwa persentase error terkecil dalam pengujian akurasi ini adalah 0%, dan persentase error terbesar pada pengujian akurasi sensor ini adalah 1,234%.

Tabel 1 Uji Akurasi Sensor

Parameter	Nilai
Rentang pengukuran (°C)	38,4-66,1
Error minimum (%)	0,34
Error maksimum (%)	4,89
Error rata-rata (%)	±2,834

Pengujian waktu respon *buzzer* juga dilakukan dengan mengamati waktu yang dibutuhkan oleh *buzzer* untuk aktif setelah salah satu indikasi kebakaran yang telah di atur pada sistem di *trigger*. Untuk pengujian terhadap ada atau tidaknya api didapati hasil waktu yang dibutuhkan *buzzer* hingga berbunyi adalah 0,52 detik untuk yang paling cepat, dan 1,98 detik untuk waktu yang paling lama. Untuk rata-rata lama waktu yang dibutuhkan untuk *buzzer* berbunyi adalah 1,338 detik.

Gambar 4 Pengujian respon *buzzer* terhadap api

Jurnal Ilmiah Sain dan Teknologi

Tabel 2 Uji Respon *Buzzer* Terhadap Api

Parameter	Nilai
Jumlah pengujian	30
Delay minimum (detik)	0,52
Delay maksimum (detik)	1,98
Delay rata-rata (detik)	$\pm 1,338$

Kemudian pengujian waktu respon *buzzer* terhadap perubahan kandungan gas dengan mengatur jumlah gas yang masuk hingga melewati ambang batas yang telah dilakukan. Hasil yang didapat adalah respon waktu tercepat yang dimiliki alat ini 0,54 detik, untuk respon terlama yang dimiliki alat ini adalah 1,95 detik, dan rata-rata waktu yang dibutuhkan adalah 1,22 detik.

Gambar 5 Pengujian Respon *buzzer* terhadap perubahan kandungan gasTabel 3 Hasil Uji Respon *Buzzer* Terhadap Gas

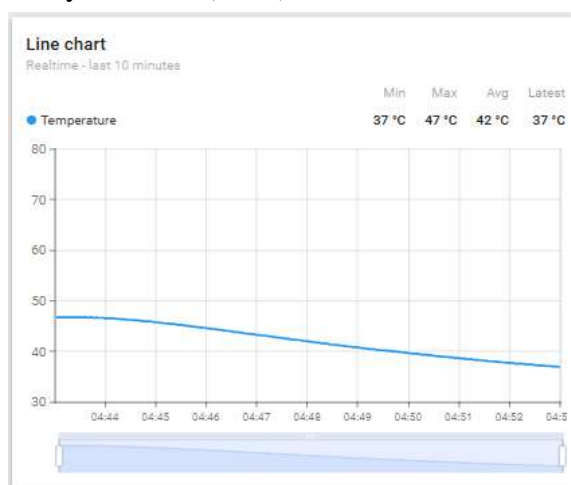
Parameter	Nilai
Jumlah pengujian	30
Delay minimum (detik)	0,54
Delay maksimum (detik)	1,95
Delay rata-rata (detik)	$\pm 1,22$

Pengujian sistem adalah pengujian sederhana untuk mengukur delay waktu yang dibutuhkan dashboard ThingsBoard untuk menampilkan nilai yang terdapat di serial monitor Arduino Ide. Dan memastikan *buzzer* bekerja sesuai dengan perintah atau *coding* di Arduino IDE. Dengan tujuan akhir adalah memastikan seluruh komponen baik alat maupun dashboard berfungsi dengan baik. Pengujian dilakukan dengan menggunakan *stopwatch* untuk mengukur waktu delay antara data yang telah ditampilkan di serial monitor Arduino IDE dengan dashboard yang telah dibuat di ThingsBoard. *Buzzer* juga akan diamati apakah sudah berperilaku sesuai dengan perintah yang telah diberikan di Arduino IDE. Untuk waktu yang dibutuhkan oleh sistem untuk mengupload nilai hasil baca sensor yang muncul di serial monitor hingga muncul di dashboard ThingsBoard. Dari 30 percobaan yang telah dilakukan didapatkan hasil pengujian sebagai berikut.

Untuk waktu delay pada bagian suhu di dashboard ThingsBoard adalah 0,4 detik untuk yang paling sedikit dan 1,4 detik untuk waktu delay terlama. Untuk waktu delay rata-ratanya adalah 0,883 detik. Dengan nilai rata-rata demikian maka bisa dibilang bahwa sistem sudah berjalan baik dari segi waktu delay. Untuk *Buzzer* didapati hasil dimana dia tidak bekerja apabila suhu tidak lebih dari 45 derajat *celcius*.

Tabel 4 Hasil Uji Waktu Delay DashBoard Bagian Suhu

Jumlah pengujian	30
Delay minimum (detik)	0,4
Delay maksimum (detik)	14
Delay rata-rata (detik)	$\pm 0,883$

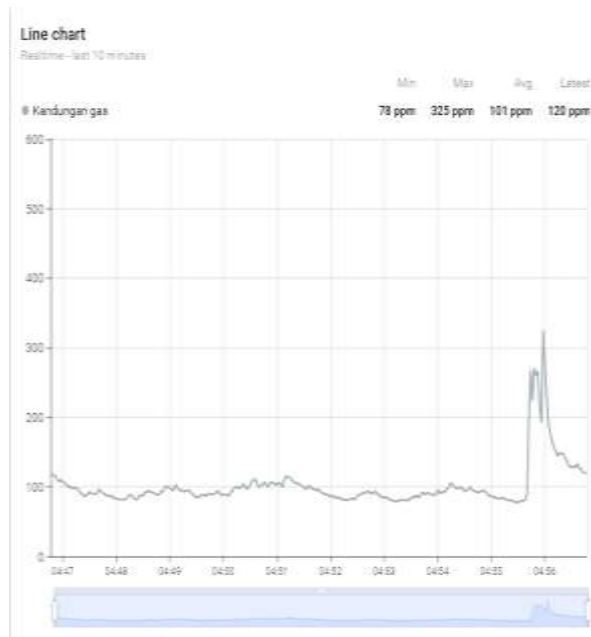


Gambar 6 Grafik Dashboard Thingsboard Bagian Suhu

Tes yang kedua disini adalah menguji keseluruhan sistem bagian kandungan gas. Disini dapat dilihat bahwa waktu delay terkecil adalah 0,6 detik dan waktu delay terlama adalah 2,0 detik. Dengan nilai rata-rata waktu delay 1,08 detik maka sistem dapat dikatakan baik pada waktu delay dashboard bagian gas. Dapat dilihat bahwa *buzzer* tidak bekerja apabila kandungan gas yang terbaca oleh sensor tidak mencapai 300 ppm.

Tabel 5 Hasil Uji Waktu Delay DashBoard Bagian Gas

Jumlah pengujian	30
Delay minimum (detik)	0,6
Delay maksimum (detik)	2,0
Delay rata-rata (detik)	$\pm 1,08$

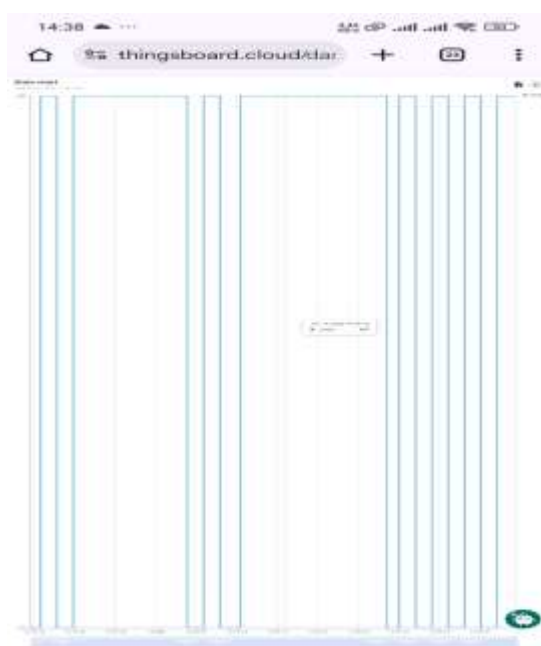


Gambar 7 Grafik Dashboard ThingsBoard Bagian Gas

Untuk tes yang terakhir adalah tes waktu delay dashboard bagian api, kita mendapati hasil dimana waktu delay terendah adalah 0,4 detik, dan waktu delay tertinggi adalah 1,2 detik. Dan untuk waktu delay rata-rata dashboard bagian ini adalah 0,796 detik.

Tabel 6 Hasil Uji Waktu Delay DashBoard Bagian Gas

Jumlah pengujian	30
Delay minimum (detik)	0,4
Delay maksimum (detik)	1,2
Delay rata-rata (detik)	$\pm 0,796$



Gambar 8 Grafik Dashboard ThingsBoard Bagian Api



Pada gambar 9 dibawah kita dapat melihat beberapa grafik dari empat variabel yang digunakan. Disini ThingsBoard menyediakan beberapa opsi pilihan seperti jangka waktu update, waktu data terakhir, dan masih banyak lagi. Namun disini penulis menggunakan waktu secepat mungkin untuk update pada dashboard, dan menggunakan waktu sepuluh menit terakhir untuk satu grafik.

Selain itu juga kita dapat melihat indikator lain seperti nilai minimal yang didapat sistem dalam waktu sepuluh menit terakhir, nilai maksimal yang didapat sistem dalam waktu sepuluh menit terakhir, rata-rata nilai yang didapat oleh sistem sejak sepuluh menit terakhir, dan nilai terkini yang didapat oleh sistem.

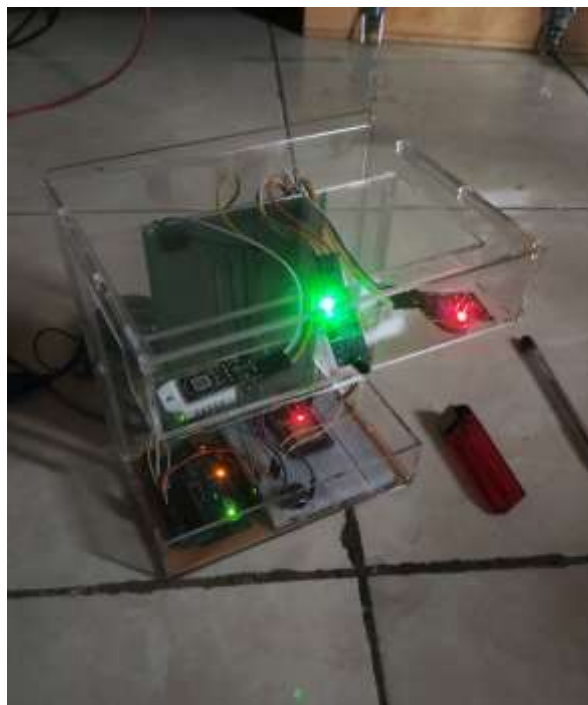
Pada bagian dashboard itu juga kita dapat melihat empat grafik yang disajikan. Yaitu grafik untuk data suhu yang masuk, grafik untuk data kandungan gas yang masuk, grafik untuk data kadar kelembapan yang masuk, kemudian yang terakhir adalah grafik data ada atau tidaknya api yang terdeteksi oleh sistem.



Gambar 9 *Widget Dashboard device* di ThingsBoard

Meskipun sistem menunjukkan kinerja yang efektif, penelitian ini masih memiliki beberapa keterbatasan

1. Penggunaan sensor DHT22 sebagai sensor suhu dalam sistem memang memiliki tingkat akurasi yang baik. Namun disisi lain sensor suhu DHT22 ini membutuhkan waktu yang relatif lama untuk merespon perubahan suhu yang berubah secara drastis.
2. Sistem peringatan yang digunakan hanyalah sebuah *buzzer*, dan tidak menggunakan kontrol lain seperti pompa air atau semacamnya. Hal ini tentunya adalah sebuah kekurangan apabila dibandingkan dengan penelitian lain. Ditambah juga untuk penelitian lain ada yang sudah mendukung fitur notifikasi langsung kepada pengguna lewat *smarthphone*.



Gambar 10 Alat pendeteksi kebakaran berbasis IoT

## PENUTUP

### Simpulan

Dari Penelitian diatas penulis dapat menarik kesimpulan bahwa alat pendeteksi kebakaran berbasis IoT yang telah dirancang memiliki hasil sebagai berikut.

1. Alat yang telah dirancang oleh penulis dapat bekerja dengan baik. Dimana mikrokontroler Arduino UNO dapat menerima data dari sensor dan mengirimkan data tersebut menuju ESP32. Kemudian data dikirim menuju ThingsBoard dan *buzzer* diaktifkan apabila terdapat indikasi adanya kebakaran oleh ESP32. Yang mana semua itu sudah sesuai dengan harapan penulis.
2. Dari uji akurasi yang telah dilakukan penulis terhadap sensor suhu yang di gunakan. Di dapati hasil dimana nilai error yang terdapat pada DHT22 jika dibandingkan dengan referensi yang digunakan yaitu *thermometer* memiliki nilai error yang relatif kecil berkisar antara 0,34% sampai 4,89%. Waktu delay *buzzer* terhadap api adalah 0,52 detik untuk delay tercepat dan 1,98 detik untuk waktu yang paling lama dengan rata-rata waktu delay 1,338 detik. Sedangkan untuk respon *buzzer* terhadap perubahan gas adalah 0,54 untuk waktu delay tercepat dan 1,95 detik untuk waktu delay. Untuk dashboard bagian api membutuhkan waktu upload 0,4 detik untuk waktu tercepat dan 1,2 detik untuk yang terlama, rata-rata waktu yang dibutuhkan adalah 0,796 detik. Untuk dashboard bagian gas didapati waktu upload yang dibutuhkan yaitu 0,6 detik untuk waktu tersingkat dan 2,0 detik untuk waktu terlama, dengan rata-rata waktu upload yang dibutuhkan 1,08 detik. Yang terakhir dashboard bagian suhu membutuhkan waktu upload 0,4 detik yang paling sedikit dan 1,4 detik untuk waktu upload yang terlama, dengan rata-rata waktu upload yang dibutuhkan 0,883 detik.

Kesimpulan akhir yang didapat adalah dengan nilai akurasi, waktu respon *buzzer*, waktu delay upload, dan visualisasi data yang terdapat pada sistem pendeteksi kebakaran berbasis IoT yang telah penulis rancang maka bisa dikatakan bahwa sistem ini telah berjalan normal. Kekurangan yang penulis cantumkan juga dapat menjadi referensi bagi penulis lain untuk membuat penelitian baru.

**Saran**

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat diberikan sebagai bahan pertimbangan untuk pengembangan sistem selanjutnya.

1. Penggunaan sensor suhu selain DHT22 dapat menjadi pertimbangan. Mengingat sensor DHT22 ini memiliki waktu baca dua detik sekali, dan respon pada perubahan drastis suhu dapat terbilang kurang responsif yaitu sekitar lima sampai sepuluh detik.
2. Menambahkan fitur kirim pesan atau notifikasi kepada gadget pengguna apabila ditemukan indikasi kebakaran. Dan Menggunakan kontrol lain seperti pompa air atau dinamo dapat menjadi opsi lain selain *buzzer*. Penggunaan variabel tersebut dapat membuat sistem lebih bagus karena selain berfungsi sebagai sistem monitoring dan peringatan, sistem juga dapat digunakan sebagai pemadam api.
3. Menggunakan pin output analog pada sensor api, agar mendapatkan hasil yang lebih akurat dan menghindari *false alarm*.
4. Pengujian akurasi sensor gas MQ-7 perlu dilakukan untuk sistem memiliki akurasi yang bagus dan dapat bekerja dengan optimal. perlu dilakukan apabila kita ingin sistem monitoring dapat digunakan dengan jangka waktu yang lama dan fitur alarm di ThingsBoard juga dapat digunakan.

**DAFTAR PUSTAKA**

- Agustini, S. (2025). Kajian Literatur: Perkembangan Edge Computing dalam Mendukung IoT. *Engineering and Technology International Journal*, 7(01), 36–43. <https://doi.org/10.55642/eatij.v7i01.969>
- ASRIN, A. (2022). Metode Penelitian Eksperimen. In *Maqasiduna: Journal of Education, Humanities, and Social Sciences* (Vol. 2, Issue 01). <https://doi.org/10.59174/mqs.v2i01.24>
- Austin, C., Mulyadi, M., & Octaviani, S. (2024). Implementasi IoT dengan ESP 32 Untuk Pemantauan Kondisi Suhu Secara Jarak Jauh Menggunakan MQTT Pada AWS. *Jurnal Elektro*, 15(2), 46–55. <https://doi.org/10.25170/jurnalelektro.v15i2.5141>
- Cahyadi, H. D., Mirza, 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>
- Daniswara, D. D., Damaliana, A. T., & Diyasa, I. G. S. M. (2024). Pengukuran Indeks Standar Pencemaran Udara Menggunakan Support Vector Machine. *Jurnal Penelitian*, 9(1), 62–69.
- Erfan Sriyono, P., & Malvin Valerian G. (2025). Sistem deteksi Kebakaran Berbasis IoT Dengan Mikrokontroler ESP32. (vol. 13, No. 2).
- Ervina Sari, T., & Siswandi Syahputra, T. (2019). Pemetaan Dan Monitoring Tingkat Kebisingan Berbasis Iot (Internet of Things) Di Institut Teknologi Sumatera. *Original Article Journal of Science and Applicative Technology*, 19(1), 15–25. <https://doi.org/10.35472/x0xx0000>
- Gultom, M. V., & Putro, I. S. (2025). Sistem Deteksi Kebakaran Berbasis Iot Dengan Mikrokontroler Esp32. *Jurnal Informatika Dan Teknik Elektro Terapan*, 13(2). <https://doi.org/10.23960/jitet.v13i2.6236>
- Hanafi, F. W., Ragasiwi, G., & Satria, D. E. (2025). *Prototype Sistem Deteksi dan Pemadaman Kebakaran Dini Indoor Berbasis ESP32 dengan Notifikasi Telegram*. 722–733.
- Herlambang, P., Kristiyono, A. E., & Eddi. (2025). *Rancang Bangun Detektor Karbon Monoksida ( CO ) dengan MQ-7 Berbasis Internet of Things Politeknik Pelayaran*

Surabaya , Indonesia di atas kapal , mencakup teknologi yang digunakan , tantangan yang dihadapi , manfaat yang. 3.

- Julianti Triswanti, S., & Basyit Mubarroq, R. (2025). Penerapan Iot Dalam Pendeteksi Gas (CO) Dan Kebakaran Dengan Notifikasi Aplikasi Telegram. (Vol. 14, No. 1).
- Kusuma, D. A., & Juliasari, N. (2024). Sistem Deteksi Kebocoran Gas Lpg Dan Kebakaran Menggunakan Mq-2 Dan Esp32 Berbasis Iot. *Jurnal Informatika Dan Sistem Informasi (JIFoSI)*, 5(3), 2722–130.
- Lim, T. K., Mohammad, A. Z., & Ismail, M. N. Bin. (2021). Development of arduino basic learning kit in robotics for beginner level learning. *Jurnal Pendidikan Teknologi Kejuruan*, 4(4), 118–122. <https://doi.org/10.24036/jptk.v4i4.20623>
- Lodan, M. W., Darkel, Y. B. M., Muda, H. B. N., & Labamaking, T. E. D. (2022). Desain Pendeteksi Kebakaran Menggunakan Sensor Suhu dan Sensor Api Berbasis IOT Dengan Metode Naive Bayes. *Digital Transformation Technology*, 4(1), 425–432. <https://doi.org/10.47709/digitech.v4i1.4121>
- Nuryadin, R. A., Yusuf, A. R., Reza, M., H, N. F. A., & A.K, P. S. D. (2024). Prototype Sistem Deteksi Kebakaran Menggunakan Sensor Mq-2 dan Flame Sensor Berbasis IoT. *Jurnal Riset Dan Aplikasi Mahasiswa Informatika (JRAMI)*, 5(4), 877–885. <https://doi.org/10.30998/jrami.v5i4.11149>
- Prathama, G. H., Andaresta, D., & Darmaastawan, K. (2021). Instalasi Framework IoT Berbasis Platform Thingsboard di Ubuntu Server. *TIERS Information Technology Journal*, 2(2), 1–9. <https://doi.org/10.38043/tiers.v2i2.3329>
- Puspasari, F., Satya, T. P., Oktawati, U. Y., Fahrurrozi, I., & Prisyanti, H. (2020). Analisis Akurasi Sistem Sensor DHT22 berbasis Arduino terhadap Thermohygrometer Standar. 1–6.
- Simanungkalit, J. T., & Rambe, B. M. (2025). Penerapan IOT Dalam Pendeteksi Gas (CO) Dan Kebakaran Dengan Notifikasi Aplikasi Telegram. *Jurnal Minfo Polgan*, 14(1), 691–701. <https://doi.org/10.33395/jmp.v14i1.14843>
- Siregar, H. . T., Permana Sutisna, S., Eka Pramono, G., & Malik Ibrahim, M. (2020). Hafzara Siregar, T., Permana Sutisna, S., Eka Pramono, G., & Malik Ibrahim, M. (2020). RANCANG BANGUN SISTEM PENDETEKSI KEBAKARAN BERBASIS IOT MENGGUNAKAN ARDUINO. In *Jurnal Ilmiah Teknik Mesin* (Vol. 7, Issue 2). <http://ejournal.uika-bogor.ac.id/index>. *Jurnal Ilmiah Teknik Mesin*, 7(2), 59–66. <http://ejournal.uika-bogor.ac.id/index.php/ame/index>
- Wahyudi, S. (2020). Buku Pintar Internet of Things dengan ESP32 dan Arduino. Jakarta: Elex Media Komputindo.