



PERANCANGAN DAN IMPLEMENTASI SISTEM PENDETEKSI DAN PERINGATAN KEBAKARAN BERBASIS IOT MENGUNAKAN NODEMCU ESP8266 DAN SENSOR API

Gabriel Gracia Salindeho¹, Theophilus Wellem^{2,*}

Program Studi Teknik Informatika, Fakultas Teknologi Informasi

Jl. O. Notohamidjojo no 1-10, Blotongan, Salatiga

Email: 672018069@student.uksw.edu, theophilus.wellem@uksw.edu*

Riwayat artikel:

Submitted: 30-05-2023

Revised: 12-07-2023

Published: 24-10-2023

Abstrak – Sistem pengendalian dan pemantauan yang menggunakan *Internet-of-Things* (IoT) dapat ditemukan dalam kehidupan sehari-hari, misalnya pada aplikasi *smart home*, *smart transportation*, dan *smart city*. Pemantauan terhadap kondisi lingkungan berbasis IoT baik dalam ruangan maupun luar ruangan umumnya dilakukan untuk mengurangi resiko terjadinya kejadian yang tidak diinginkan. Salah satu contoh kejadian yang tidak diinginkan adalah kebakaran. Kebakaran dapat terjadi di rumah, gedung perkantoran, maupun sawah atau hutan. Untuk membantu memberikan peringatan dini akan timbulnya api yang dapat menyebabkan kebakaran, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pendeteksi api dan peringatan kebakaran berbasis IoT menggunakan NodeMCU ESP8266 dan sensor api. Sensor yang digunakan merupakan sensor inframerah yang akan mendeteksi radiasi cahaya inframerah yang berasal dari api. Hasil pengujian menunjukkan bahwa sistem berhasil mendeteksi keberadaan api hingga berjarak 50 cm dari sensor api dan dapat dipantau melalui *thinger.io*. Selain itu, notifikasi dari *thinger.io* melalui email dan Telegram kepada pengguna juga berhasil dilakukan.

Kata Kunci – kebakaran, NodeMCU ESP8266, sensor api, IoT, *thinger.io*.

Abstract – *Control and monitoring systems that use the Internet-of-Things (IoT) can be found in everyday life, for example, in smart home applications, smart transportation, and smart cities. IoT-based monitoring of environmental conditions both indoors and outdoors is generally carried out to reduce the risk of unwanted events. One example of an unwanted event is a fire. Fires can occur in houses, office buildings, rice fields, or forests. In order to help provide early warning of a flame that can cause a fire, this study aims to design and implement an IoT-based fire detection and warning system using the NodeMCU ESP8266 and a flame sensor. The sensor used is an infrared sensor that will detect infrared light radiation coming from a fire. The test results show that the system successfully detects the presence of fire up to 50 cm from the fire sensor and can be monitored via thinger.io. Additionally, notifications from thinger.io via e-mail and Telegram to users were successfully executed.*

Keywords – *fire, NodeMCU ESP8266, flame sensor, IoT, thinger.io.*

I. PENDAHULUAN

Kebakaran merupakan salah satu kejadian yang berpotensi terjadi pada berbagai tempat, baik pada perkantoran maupun pada perumahan masyarakat. Kebakaran merupakan kejadian berkobarnya api yang tidak dikehendaki dan dapat menimbulkan kerugian materil [1]. Kebakaran dapat disebabkan oleh faktor alam misalnya, sambaran petir yang mengenai rumah atau pohon, gempa bumi, kekeringan yang panjang, dan gunung meletus. Selain itu, kebakaran dapat juga disebabkan oleh faktor kelalaian manusia. Contoh faktor kelalaian manusia misalnya, instalasi listrik yang tidak layak, instalasi pipa gas yang tidak sesuai, lupa mematikan kompor, pembakaran sampah sembarangan, serta merokok di tempat yang tidak semestinya [2][3]. Oleh karena berbahayanya kebakaran, maka dikembangkan peralatan untuk memberi peringatan (alarm), mencegah, maupun menangani kebakaran, misalnya, *fire hydrant* lengkap dengan kotaknya, yang umum terpasang pada gedung-gedung perkantoran dan sekolah.

Dengan berkembangnya teknologi *Internet-of-Things* (IoT) yang kemudian melahirkan konsep *smart city*, *smart building*, *smart home*, dan lain sebagainya, sistem untuk mendeteksi api dan memberikan peringatan dini terhadap bahaya kebakaran juga dapat diimplementasikan menggunakan *platform* perangkat keras yang ditujukan untuk aplikasi IoT (misalnya, Arduino, NodeMCU ESP8266, dan ESP32) dan perangkat lunak yang menyediakan *cloud IoT platform* (misalnya, ThingSpeak, blynk.io, dan thingsboard.io) [4]–[6]. Pencegahan dan deteksi dini terhadap potensi kebakaran merupakan hal yang sangat penting karena kebakaran dapat menimbulkan kerugian yang besar. Oleh karena itu, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pendeteksi dan peringatan kebakaran berbasis IoT. Sistem yang diimplementasikan pada penelitian ini menggunakan NodeMCU ESP8266 dan sensor api (*flame sensor*) sebagai komponen utamanya. Untuk pelaporan data dan notifikasi kepada pengguna, digunakan *thinger.io* [7] yang akan mengirimkan email dan pesan melalui Telegram. Sistem ini diharapkan dapat bermanfaat untuk peringatan dini keberadaan api dan bahaya kebakaran sehingga meminimalkan kerugian yang ditimbulkan.

Secara umum, penelitian-penelitian mengenai sistem pendeteksi api atau sistem peringatan kebakaran berbasis IoT menggunakan berbagai *platform* perangkat keras maupun perangkat lunak seperti yang telah disebutkan sebelumnya. NodeMCU ESP8266 menjadi *platform* yang paling sering digunakan karena mempunyai modul WiFi yang terintegrasi. Perbedaan dari berbagai sistem deteksi kebakaran yang ada terletak pada jumlah fitur dan sensor, serta *platform cloud IoT* yang digunakan. Perbedaan penelitian ini dengan beberapa penelitian sebelumnya adalah penggunaan *platform thinger.io*, sedangkan umumnya *platform* atau aplikasi yang digunakan oleh penelitian-penelitian sebelumnya adalah Blynk [8].

II. KAJIAN PUSTAKA

A. Penelitian sebelumnya

Penelitian oleh Napu dkk. [1] misalnya, menggunakan NodeMCU ESP8266, sensor api, sensor suhu DHT11, dan sensor gas MQ2, serta aplikasi Blynk untuk memantau sistem. Penelitian oleh Putra dan Pramudita [2], menggunakan NodeMCU ESP8266,

sensor api, *buzzer*, dan *relay*. Untuk notifikasi, sistem tersebut menggunakan Pushsafer [9] yang merupakan layanan *push notification* ke *mobile phone* dan *browser*. Hartono, dkk. mengembangkan sistem pendeteksi kebakaran berbasis website pada penelitian [3]. Peralatan yang digunakan antara lain NodeMCU ESP8266, sensor api, sensor suhu DHT11, *buzzer*, LED, dan *relay*.

Beberapa penelitian akademik lainnya juga telah mengembangkan sistem berbasis IoT yang dapat digunakan untuk mendeteksi api dan membantu mengurangi resiko bahaya kebakaran [6][10]–[12]. Penelitian oleh Danang, dkk. [6] mengembangkan sistem keamanan rumah yang mempunyai fitur pendeteksi kebakaran. Penelitian tersebut menggunakan NodeMCU ESP8266, sensor api, sensor *passive infrared* (PIR), kamera ESP32 CAM, dan peralatan pendukung lainnya. Notifikasi dilakukan melalui aplikasi Blynk dan Telegram. Sistem pemantauan dan alarm kebakaran rumah pada penelitian [11] juga menggunakan NodeMCU ESP8266, sensor suhu DHT11, *relay*, dan *smartphone* Android. Sensor api tidak digunakan pada penelitian tersebut, sehingga monitoring didasarkan pada data suhu. Jika suhu lebih besar dari 42 °C, maka *relay* akan menyalakan pompa air. Pada penelitian [12], Manfaluthy, dkk. merancang prototipe sistem pendeteksi awal kebakaran rumah menggunakan NodeMCU ESP8266, sensor api, sensor suhu DHT22, sensor MQ5 (LPG), sensor MQ9 (CO), *buzzer*, dan *relay*. Sistem terhubung dengan internet dan dapat dimonitor dari aplikasi berbasis web. Samudera dan Sugiharto [13] mengembangkan sistem peringatan kebocoran gas dan penanganan kebakaran berbasis IoT. Sistem tersebut menggunakan NodeMCU ESP8266, sensor api, sensor MQ2 (asap, LPG), motor servo, *load cell*, dan *buzzer*. Sebuah aplikasi web dikembangkan untuk memonitor sistem tersebut sehingga data konsentrasi gas LPG dapat selalu terpantau.

Penelitian oleh Tarigan [14] mengembangkan sistem pendeteksi kebakaran menggunakan NodeMCU ESP8266, sensor api, sensor MQ2 (sebagai sensor asap), sensor suhu DHT11, dan pompa. Sistem tersebut dikontrol melalui aplikasi Blynk. Pada penelitian [15], Yasir dan Az-Zahra mengembangkan prototipe sistem untuk mendeteksi kebakaran menggunakan NodeMCU ESP8266 dan logika fuzzy. Komponen dan sensor yang digunakan oleh sistem tersebut adalah LED, LCD, *buzzer*, sensor api, sensor suhu, dan sensor MQ2. Logika fuzzy diterapkan untuk mengontrol *buzzer*, LED, dan LCD. Aplikasi Blynk digunakan untuk memantau dan mengontrol sistem. Sistem rumah cerdas dikembangkan oleh Agung, dkk. pada penelitian [16]. Sistem ini menggunakan NodeMCU ESP8266, sensor suhu DHT11, *buzzer*, *relay*, dan aplikasi Telegram untuk mengimplementasikan konsep rumah cerdas di mana lampu (yang disimulasikan dengan LED) dapat dikontrol melalui Telegram. Berdasarkan kajian yang dilakukan terhadap beberapa penelitian sebelumnya, sistem pendeteksi api dan peringatan kebakaran yang diimplementasikan pada penelitian ini menggunakan peralatan (perangkat keras) yang mirip dengan penelitian-penelitian sebelumnya. Perbedaannya terletak pada peralatan, jumlah dan jenis sensor, serta *cloud IoT platform* yang digunakan, yaitu *thinger.io*.

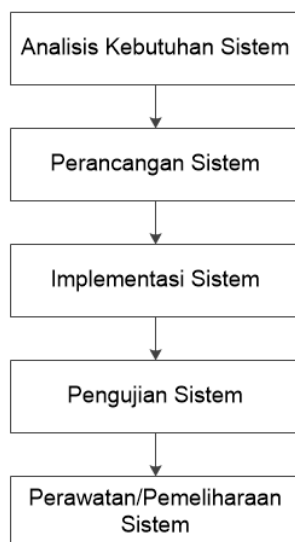
B. Platform *thinger.io*

Thinger.io [7] merupakan sebuah *cloud IoT platform* yang menyediakan server (*backend*) dan perangkat lunak (*web-based application* yang berperan sebagai *frontend*) untuk pengembangan aplikasi IoT. Keunggulan dari *thinger.io* adalah kompatibel dengan berbagai jenis perangkat dan mendukung koneksi/komunikasi dua arah dengan perangkat

seperti Arduino, Raspberry, maupun perangkat yang menggunakan MQTT (*Message Queuing Telemetry Transport*). Selain itu, *thinger.io* juga dapat menyimpan data yang diterima dari peralatan IoT maupun menampilkannya secara *real-time*. Bagian *frontend* atau aplikasi berbasis web yang disediakan oleh *thinger.io* juga dapat dimodifikasi atau ditambahkan fitur-fitur lainnya agar sesuai dengan kebutuhan dari pengembang aplikasi IoT [17].

III. METODE PENELITIAN

Metode yang digunakan dalam perancangan dan implementasi sistem pendeteksi dan peringatan kebakaran berbasis IoT pada penelitian ini adalah *System Development Life Cycle* (SDLC), yang terdiri dari lima tahap yaitu, analisis kebutuhan sistem (*analysis*), perancangan sistem (*design*), implementasi sistem (*implementation*), pengujian sistem (*testing/verification*), dan perawatan/pemeliharaan sistem (*maintenance*). Tahap-tahap ini ditunjukkan pada Gambar 1.



Gambar 1. Tahap pada *System Development Life Cycle* (SDLC)

Tahap-tahap tersebut diuraikan sebagai berikut:

1. Analisis Kebutuhan Sistem: Analisis kebutuhan dilakukan untuk menentukan perangkat keras dan perangkat lunak yang dibutuhkan oleh sistem yang akan dirancang. Berdasarkan analisis yang dilakukan, komponen perangkat keras yang dibutuhkan antara lain, NodeMCU ESP8266, modul sensor api, *buzzer*, LED, kabel USB, kabel jumper, dan *breadboard*. Kebutuhan perangkat lunak terdiri dari Arduino IDE, Fritzing, *thinger.io*, dan Telegram. Penjelasan singkat mengenai komponen-komponen perangkat keras dan perangkat lunak tersebut adalah sebagai berikut:
 - a. NodeMCU ESP8266 merupakan *board* mikrokontroler yang menggunakan ESP8266 *system-on-chip* (SoC). SoC ini terdiri dari *microcontroller unit* (MCU), modul WiFi, dan komponen-komponen perangkat keras pendukung lainnya. *Board* NodeMCU ESP8266 mempunyai 11 *digital I/O* pin dan satu *analog I/O* pin yang dapat digunakan untuk menghubungkan *board* tersebut dengan berbagai

- macam perangkat keras lain seperti sensor, *buzzer*, motor DC, dan LED. NodeMCU ESP8266 berperan sebagai pengendali utama pada sistem yang dirancang. Modul sensor api, LED, *buzzer*, dan komponen perangkat keras pendukung lainnya dihubungkan ke *board* NodeMCU ESP8266.
- b. Sensor api yang digunakan pada sistem yang dirancang adalah YG1006 *infrared* (IR) *phototransistor*. Sensor ini dapat mendeteksi api atau sumber cahaya dengan panjang gelombang antara 760 nm hingga 1100 nm. Sensor ini tersedia dalam bentuk modul dan pada modul sensor tersebut juga terdapat potensiometer untuk mengatur tingkat sensitivitas pendeteksian cahaya (radiasi inframerah yang berasal dari api). Saat mendeteksi api, ukuran dan jarak api sangat mempengaruhi kemampuan sensor untuk mendeteksi cahaya. Modul sensor ini memiliki 3 pin yaitu, DO (*digital output*), GND, dan VCC. Pada sistem yang dirancang dan diimplementasikan, modul sensor ini digunakan sebagai pendeteksi cahaya (ada atau tidaknya radiasi inframerah dari api).
 - c. Komponen perangkat keras lainnya: *light emitting diode* (LED) berfungsi sebagai indikator bahwa NodeMCU ESP8266 telah terkoneksi ke internet melalui WiFi. *Buzzer* berfungsi sebagai alarm yang akan berbunyi ketika sensor mendeteksi keberadaan api. *Breadboard* digunakan sebagai tempat untuk merangkai komponen-komponen perangkat keras. Kabel jumper digunakan sebagai penghubung komponen-komponen pada *breadboard*, dan kabel USB digunakan sebagai penghubung antara perangkat lunak Arduino IDE dengan NodeMCU ESP8266 untuk melakukan *upload* program ke mikrokontroler.
 - d. Arduino IDE merupakan perangkat lunak yang digunakan untuk menulis program, kompilasi program, dan melakukan *upload* program untuk mikrokontroler pada *board* NodeMCU ESP8266.
 - e. Perangkat lunak lainnya: Fritzing digunakan untuk menggambarkan rangkaian perangkat keras dan Thingier.io yang digunakan untuk memantau kondisi sistem, mengirimkan email dan notifikasi melalui Telegram kepada pengguna untuk memberikan informasi mengenai sistem (misalnya, kondisi aman atau ada api yang terdeteksi).
2. Perancangan Sistem: Perancangan perangkat keras (berupa skematik rangkaian perangkat keras) dan perangkat lunak yang dibutuhkan oleh sistem dilakukan pada tahap ini. Diagram alir (*flowchart*) yang kemudian diimplementasikan dalam bentuk program untuk mikrokontroler juga direalisasikan pada tahap ini.
 3. Implementasi Sistem: Implementasi perangkat keras dan perangkat lunak sistem dilakukan pada tahap ini. Komponen-komponen perangkat keras dirangkai dan dihubungkan berdasarkan skematik rangkaian perangkat keras yang dihasilkan pada tahap Perancangan Sistem. Penulisan program untuk di-*upload* ke NodeMCU ESP8266 dilakukan menggunakan Arduino IDE.
 4. Pengujian Sistem: Pengujian sistem dilakukan dengan cara menjalankan sistem dan memeriksa apakah respon dari sistem sudah sesuai dengan tujuan perancangan sistem. Perbaikan akan dilakukan secara berulang hingga sistem dapat berjalan sesuai tujuannya.

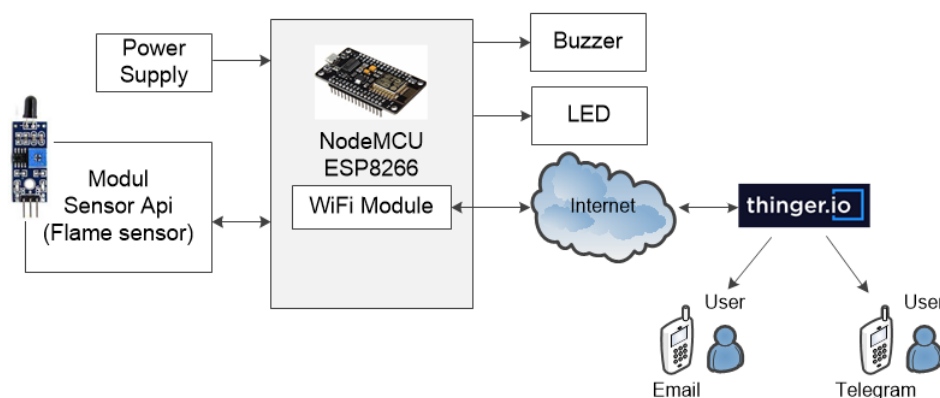
5. Perawatan/Pemeliharaan Sistem: Tahap ini tidak dilakukan pada penelitian ini karena sistem yang dibuat masih berupa prototipe untuk tujuan penelitian dan bukan untuk penggunaan sehari-hari.

IV. HASIL DAN PEMBAHASAN

A. Perancangan dan Implementasi Sistem

Diagram blok sistem yang dirancang pada penelitian ini ditunjukkan pada Gambar 2. Sistem ini menggunakan *board* NodeMCU ESP8266 dan modul sensor api sebagai komponen utamanya. NodeMCU ESP8266 harus terhubung ke internet agar sistem dapat mengirimkan data dari sensor ke thinger.io. Jika sistem mendeteksi keberadaan api, maka akan dikirim notifikasi dari thinger.io kepada pengguna melalui email dan Telegram yang terdaftar. Pin DO pada modul sensor api akan menghasilkan *output* (“1” atau “0”) yang mengindikasikan ada atau tidaknya api yang terdeteksi. *Output* ini kemudian diproses oleh NodeMCU untuk mengaktifkan *buzzer* (alarm) jika api terdeteksi dan mengirimkan data sensor ke thinger.io. Tabel 1 menunjukkan komponen perangkat keras yang digunakan untuk melakukan implementasi sistem pada penelitian ini.

Untuk menggunakan thinger.io, sistem harus didaftarkan ke thinger.io (untuk mendapatkan *device id* dan *device credential*) agar sistem dapat dipantau dan mengirimkan data ke thinger.io. Program yang dibuat juga harus memanfaatkan *library* thinger.io (*ThingierWifi.h*, *ThingierESP8266.h*, dan sebagainya) dan menggunakan fungsi-fungsi yang disediakan thinger.io untuk mengirimkan data.



Gambar 2. Diagram blok sistem

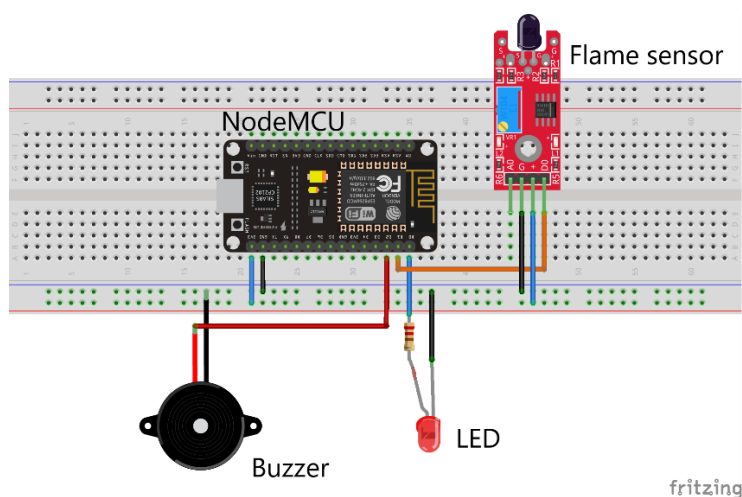
Tabel 1. Komponen perangkat keras yang digunakan

No	Nama	Jumlah
1	NodeMCU ESP8266	1
2	Modul sensor api	1
3	LED	1
4	<i>Buzzer</i>	1
5	<i>Breadboard</i>	1
6	Kabel jumper	9
7	Kabel USB	1

Skematik rangkaian perangkat keras (dibuat menggunakan perangkat lunak Fritzing) yang merupakan hasil dari tahap Perancangan Sistem ditunjukkan pada Gambar 3. Tabel 2 menunjukkan koneksi pin dari komponen perangkat keras yang digunakan dengan pin pada NodeMCU ESP8266.

Tabel 2. Koneksi pin komponen dengan *board* NodeMCU ESP8266

Komponen	Pin	Pin pada NodeMCU
Modul sensor api	DO (<i>output</i>)	D1 (GPIO5)
	GND	GND
	VCC	3V3
LED	Anode (+)	D0 (GPIO16)
	Cathode (-)	GND
<i>Buzzer</i>	Supply (+)	D2 (GPIO4)
	GND (-)	GND



Gambar 3. Skematik rangkaian perangkat keras sistem

Implementasi perangkat lunak dilakukan menggunakan pemrograman bahasa C serta menggunakan pustaka (*library*) yang disediakan dalam bentuk file *header* (.h). File header yang digunakan adalah ThingyESP8266.h dan ESP8266WiFi.h yang masing-masing memuat fungsi untuk thingy.io dan koneksi WiFi. Gambar 4 menunjukkan contoh kode program yang diimplementasikan pada fungsi `setup()`. Fungsi ini akan dieksekusi hanya satu kali saat sistem diaktifkan atau saat sistem direset. Gambar 5 menunjukkan *flowchart* yang mendeskripsikan bagaimana sistem yang dirancang pada penelitian ini bekerja. Saat sistem diaktifkan (*power ON*), NodeMCU akan melakukan inisialisasi perangkat keras dan mencoba koneksi ke internet melalui WiFi. Jika koneksi berhasil, maka LED akan diaktifkan dan sistem siap bekerja untuk mendeteksi ada atau tidaknya api. Data dari sensor dikirimkan ke thingy.io dan kondisi sistem dapat dipantau dari thingy.io. Jika sensor mendeteksi adanya api, maka sistem akan mengaktifkan *buzzer* dan mengirimkan data ke thingy.io dan pengguna akan menerima notifikasi melalui email dan Telegram.

```

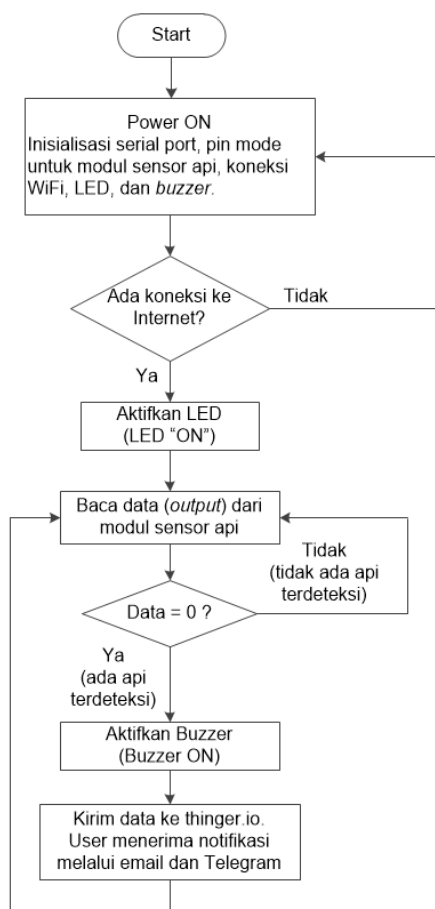
void setup() {
  Serial.begin(9600);
  pinMode(FLAME_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(LED_PIN, OUTPUT);

  //koneksi ke wifi
  WiFi.begin(ssid, ssid_password);
  //cek koneksi wifi
  while(WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    digitalWrite(LED_PIN, LOW);
  }

  //apabila terkoneksi
  digitalWrite(LED_PIN, HIGH);
  //hubungkan nodemcu ke thinger io
  thing.add_wifi(ssid, ssid_password);

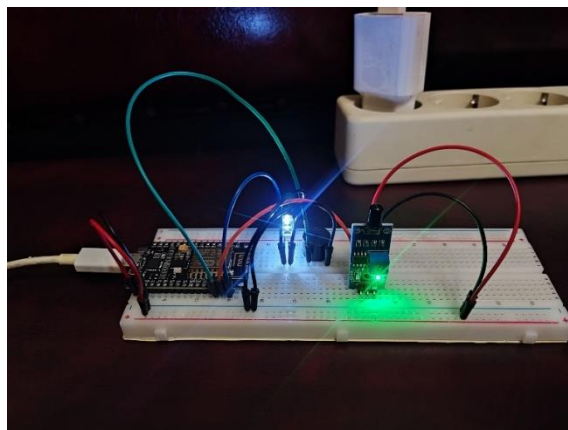
  //data yang akan dikirim ke
  thinger.io
  thing["data"] >> [] (pson & out)
  {
    out["kondisi"] = kondisi;
  };
}
    
```

Gambar 4. Contoh kode program untuk fungsi setup ()



Gambar 5. Flowchart program untuk sistem yang dirancang

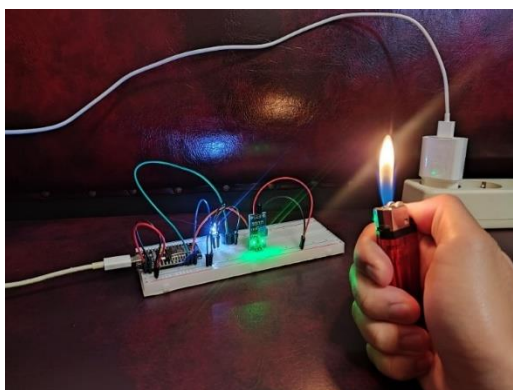
Hasil implementasi dari sistem pada penelitian ini ditunjukkan pada Gambar 6. LED berwarna biru menandakan sistem telah terhubung ke internet melalui WiFi dan satu LED berwarna hijau menyala pada modul sensor menandakan bahwa sensor telah aktif dan siap untuk mendeteksi adanya api.



Gambar 6. Hasil implementasi perangkat keras

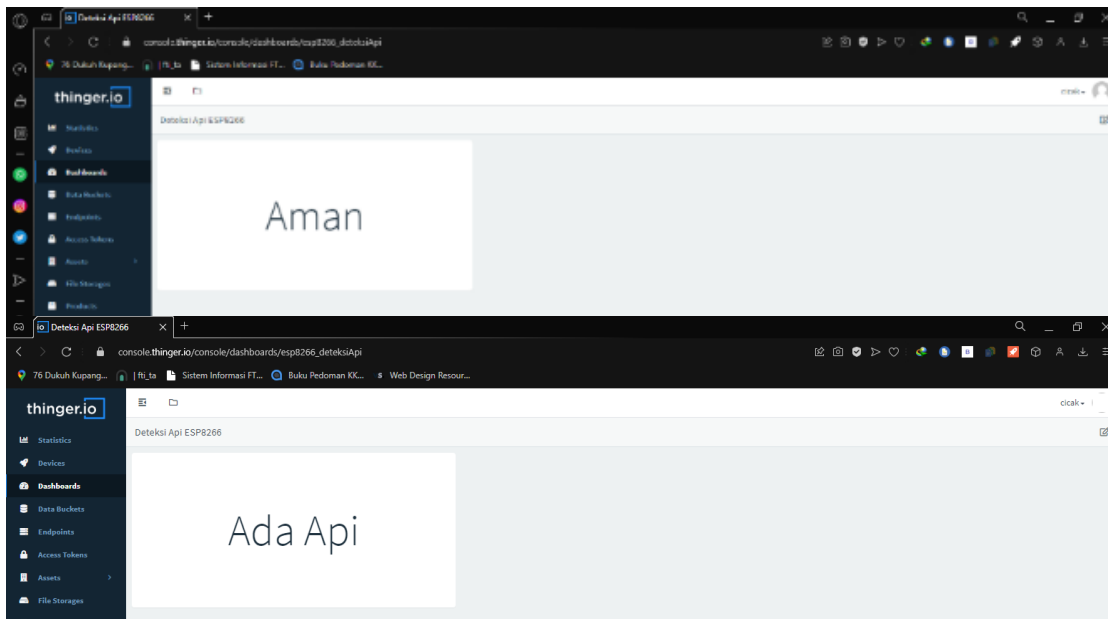
B. Pengujian Sistem

Pengujian awal terhadap sistem dilakukan untuk memastikan bahwa LED menyala saat sistem terhubung dengan internet melalui WiFi dan modul sensor dapat berfungsi dengan baik untuk mendeteksi adanya api. Pengujian ini ditunjukkan pada Gambar 7 di mana LED berwarna biru menyala (sistem terhubung ke internet) dan indikator adanya api (dua LED berwarna hijau menyala pada modul sensor) aktif.

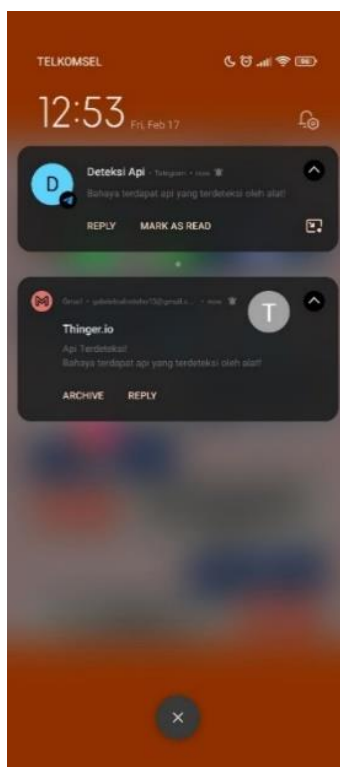


Gambar 7. Uji coba fungsi sensor

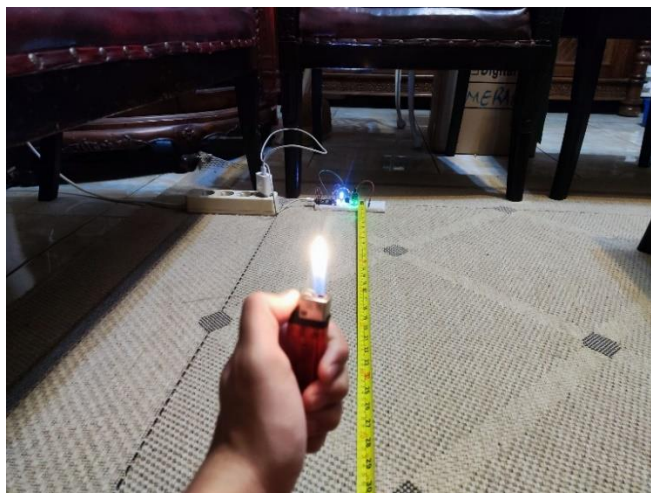
Selanjutnya, dilakukan pengujian apakah data dari sensor ketika dapat diterima oleh *thinger.io*. Pengujian ini ditunjukkan pada Gambar 8 di mana terlihat bahwa kondisi aman dan tidak terdeteksi adanya api. Gambar 9 menunjukkan tampilan pada *smartphone* ketika terdapat api yang dideteksi oleh sensor. Data dari sensor akan diterima oleh *thinger.io* yang kemudian mengirimkan notifikasi kepada pengguna melalui email dan Telegram. Pengujian berikutnya adalah pengujian jarak efektif sistem saat mendeteksi api yang ditunjukkan pada Gambar 10. Pengujian jarak sumber api ke sensor dapat dilakukan dengan jarak terjauh 100 cm.



Gambar 8. Hasil pemantauan sistem pada thinger.io saat aman dan api terdeteksi



Gambar 9: Notifikasi melalui Telegram dan email bahwa ada api yang terdeteksi



Gambar 10. Uji coba jarak efektif sensor

Hasil pengujian jarak efektif sistem saat mendeteksi api ditunjukkan pada Tabel 3. Berdasarkan pengujian yang dilakukan, deteksi api yang berasal dari korek gas akan optimal hingga jarak 50 cm dari sensor, sedangkan di atas jarak tersebut sensor kesulitan untuk mendeteksi api. Besarnya api serta jarak dari sensor berpengaruh terhadap kemampuan sensor mendeteksi api. Pada jarak 60 cm dan seterusnya, sensor tidak dapat mendeteksi api sehingga sistem ini tidak akan mengeluarkan bunyi alarm. Hal ini dapat ditingkatkan dengan menggunakan sensor api yang lebih akurat dibandingkan sensor yang digunakan pada penelitian ini.

Tabel 3. Hasil pengujian jarak

No	Jarak	Aksi	Status
1	0 cm – 10 cm	LED sensor ON dan buzzer ON	Api terdeteksi
2	20 cm – 30 cm	LED sensor ON dan buzzer ON	Api terdeteksi
3	40 cm – 50 cm	LED sensor ON dan buzzer ON	Api terdeteksi
4	60 cm – 70 cm	LED sensor berkedip dan buzzer OFF	Api tidak terdeteksi
5	80 cm – 90 cm	LED sensor berkedip dan buzzer OFF	Api tidak terdeteksi
6	\geq 100 cm	LED sensor berkedip dan buzzer OFF	Api tidak terdeteksi

V. SIMPULAN DAN SARAN

Penelitian ini membahas perancangan dan implementasi sistem pendeteksi dan peringatan kebakaran berbasis IoT menggunakan NodeMCU ESP8266 dan sensor api. Thingier.io digunakan untuk menampilkan hasil pemantauan sensor dan memberikan notifikasi melalui Telegram dan email kepada pengguna ketika api terdeteksi. Berdasarkan hasil pengujian yang dilakukan, sistem dapat bekerja sesuai dengan tujuan yang ditetapkan, yaitu memberikan peringatan adanya api yang terdeteksi oleh sensor. *Buzzer* yang digunakan sebagai alarm akan berbunyi ketika sensor mendeteksi api. Selain itu, hasil pemantauan (kondisi aman atau api terdeteksi) oleh sistem dapat dilihat melalui thingier.io.

Pengujian jarak efektif sensor terhadap sumber api menunjukkan bahwa sensor kesulitan untuk mendeteksi api pada jarak lebih besar dari 50 cm. Untuk pengembangan lebih lanjut dapat digunakan sensor api yang lebih baik dan sensor-sensor lain sebagai

pendukung, misalnya sensor CO dan CO₂, serta dapat dikembangkan aplikasi, misalnya aplikasi berbasis Android, untuk memantau dan mengendalikan sistem.

DAFTAR PUSTAKA

- [1] A. Napu, O. Kembuan, and K. Santa, "Sistem Peringatan dan Penanganan Dini Kebakaran berbasis Internet of Things (IoT)," *JOINTER-Journal of Informatics Engineering*, vol. 03, no. 01. pp. 10–16, 2022.
- [2] M. D. D. Putra and R. Pramudita, "Sistem Deteksi Api berbasis Internet of Things pada Rumah," *Media J. Inform.*, vol. 13, no. 2, pp. 64–69, 2021, doi: 10.35194/mji.vvii.1835.
- [3] A. Hartono, Siswanto, and A. Widjaja, "Prototype Pendeteksi Kebakaran menggunakan Sensor Flame, Sensor DHT11 dan Mikrokontroler NodeMCU ESP8266 berbasis Website," *Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI) Jakarta-Indonesia*. pp. 734–741, 2022.
- [4] A. Sihombing, D. Setiawan, and M. A. Sembiring, "Implementasi IoT (Internet of Things) pada Sistem Fire and Gas Detection dengan Platform Blynk," *J. CyberTech*, vol. 1, no. 12, pp. 1–12, 2018.
- [5] U. A. Saputro and A. Tuslam, "Sistem Deteksi Kebakaran berbasis Internet of Things dengan Pesan Peringatan menggunakan NodeMCU ESP8266 dan Platform ThingSpeak," *J. InfoMedia*, vol. 7, no. 1, pp. 24–30, 2022.
- [6] D. Danang, E. Fredyan, and I. S. Suasana, "Prototype Alat Keamanan Rumah Internet of Things (IoT) berbasis NodeMCU ESP8266 dengan ESP32 CAM dan Kombinasi Sensor menggunakan Telegram," *UNITECH*, vol. 1, pp. 1–9, 2022.
- [7] "Thinger.io - Open Source IoT Platform," *Thinger.io*. Accessed: May 29, 2023. [Online]. Available: <https://thinger.io/>.
- [8] "Blynk: A low-code IoT software platform for businesses and developers." <https://blynk.io/> (accessed May 29, 2023).
- [9] "Pushsafer - send and receive of Push Notifications to iOS, Android and Windows 10 Devices." <https://www.pushsafer.com/> (accessed May 29, 2023).
- [10] I. Aulia and M. Munasir, "Rancang Bangun Alat Deteksi Kebocoran Gas LPG serta Penanggulangan Kebakaran menggunakan Sensor MQ2 dan Sensor Api berbasis IoT," *J. Fis. Unand*, vol. 11, no. 3, pp. 306–312, 2022, doi: 10.25077/jfu.11.3.306-312.2022.
- [11] M. Imamuddin and Zulwisli, "Sistem Alarm dan Monitoring Kebakaran Rumah berbasis NodeMCU dengan Komunikasi Android," *Voteknika*, vol. 7, no. 2, pp. 40–45, 2019.
- [12] M. Manfaluthy, A. Pangestu, and I. Nurjaman, "Prototipe Sistem Pendeteksi Kebakaran berbasis ESP8266 dan IFTTT," *TELKA*, vol. 8, no. 1, pp. 60–73, 2022.
- [13] D. Samudera and A. Sugiharto, "Sistem Peringatan dan Penanganan Kebocoran Gas Flammable dan Kebakaran berbasis Internet of Things (IoT)," *J. TeknoSAINS Seri Tek. Elektro*, vol. 1, no. 1, pp. 1–13, 2018.

- [14] P. Tarigan, “Rancang Bangun Pendeteksi Kebakaran menggunakan NodeMCU ESP8266,” *CTIS*, vol. 5, no. 2, pp. 15–24, 2021.
- [15] G. A. Yasir and F. F. Az-Zahra, “Prototipe Pendeteksi Kebakaran menggunakan NodeMCU ESP8266 dengan Implementasi Logika Fuzzy,” *J. Elektron. dan Komput.*, vol. 15, no. 2, pp. 280–288, 2022, [Online]. Available: <http://journal.stekom.ac.id/index.php/elkom>.
- [16] P. Agung, A. Z. Iftikhor, D. Damayanti, M. Bakri, and M. Alfarizi, “Sistem Rumah Cerdas berbasis IoT dengan Mikrokontroler NodeMCU dan Aplikasi Telegram,” *JTIKOM*, vol. 1, no. 1, pp. 8–14, 2020.
- [17] “Thingier.io Open Source IoT Platform Documentation.” Thingier.io, Accessed: Jul. 12, 2023. [Online]. Available: <https://github.com/thingier-io/Docs>.