

Sistem Deteksi dan Penanggulangan Kebakaran Dini Berbasis *Internet of Things* (IoT)

Nicola Gianina Suryadi^{a)}, Eko Satria^{b)}, Mitra Djamal^{c)}

Laboratorium Elektronika dan Instrumentasi,
Kelompok Keahlian Fisika Teoritik Energi Tinggi dan Instrumentasi,
Program Studi Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Institut Teknologi Bandung,
Jl. Ganesha no. 10 Bandung, Indonesia, 40132

^{a)} nicolagianina.s@gmail.com

^{b)} ekosatria004@gmail.com

^{c)} mitra@fi.itb.ac.id

Abstrak

Kebakaran rumah merupakan hal yang sering terjadi, terutama di daerah pemukiman warga dimana api dapat dengan cepat menyebar ke rumah-rumah lainnya. Kebakaran dapat menyebabkan korban jiwa maupun kerugian materiil, oleh karena itu alangkah baiknya jika kebakaran dapat dideteksi dini dan ditangani secara cepat. Sistem prevensi kebakaran rumah sederhana berbasis Internet of Things (IoT) dibuat dengan modul sensor suhu dan kelembaban DHT11, modul sensor gas/asap MQ2, dan modul flame detector. Ketika dideteksi ada asap, suhu yang tinggi, dan api yang menandakan terbentuknya kebakaran, maka sistem pemadam sederhana akan menyala dan memberitahu pengguna lewat aplikasi Android. Namun jika setelah beberapa saat masih terdeteksi asap, suhu tinggi, dan api maka sistem akan mengirim pesan kepada pemadam kebakaran untuk segera memadamkan sumber api di rumah. Sistem kebakaran dini ini diprogram dengan Espruino Web IDE menggunakan board Wemos D1 mini dengan ESP8266, sedangkan aplikasi Android untuk monitoring dibuat dengan menggunakan MIT App Inventor. Dari percobaan yang telah dilakukan, dapat dilihat bahwa prototipe berjalan dengan semestinya untuk data dummy.

Kata kunci: Android, aplikasi monitoring, IoT, kebakaran rumah, Wemos

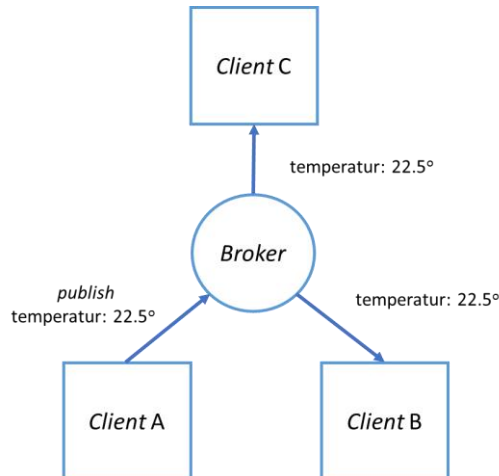
PENDAHULUAN

Kebakaran rumah merupakan hal yang sering terjadi, terutama di daerah pemukiman warga yang mana api dapat cepat menyebar. Dengan adanya teknologi yang berkembang dengan pesat dalam segala aspek kehidupan, baik adanya jika kebakaran dapat dideteksi dan ditanggulangi secepatnya. Salah satu teknologi yang sedang berkembang pesat saat ini adalah teknologi *Internet of Things* (IoT), terutama dalam era industri 4.0 yang merupakan sebutan untuk industri dengan adanya otomatisasi semakin meningkat. Terdapat beberapa riset yang telah dilakukan mengenai prevensi dan penanggulangan kebakaran rumah berbasis IoT. Salah satu riset tersebut adalah pengembangan sistem prevensi kebakaran untuk rumah dengan menggunakan WSN (*Wireless Sensor Network*), dan GSM (*Global System for Mobile Communications*) untuk menghindari adanya *false alarm*. WSN merupakan jaringan sensor yang terhubung secara nirkabel dan merupakan salah satu bagian dari IoT. Pada riset tersebut, terdapat beberapa sensor kebakaran yang dipasang pada beberapa lokasi di rumah dalam satu jaringan, sehingga ketika dideteksi kebakaran pada salah satu atau beberapa lokasi, sistem akan memperingati pengguna dan menunggu konfirmasi user melalui GSM. Sistem kemudian akan menghubungi pemadam kebakaran jika pengguna sudah mengkonfirmasi bahwa terjadi kebakaran [1]. Perbedaan dengan penelitian ini adalah pada riset tersebut tidak terdapat sistem pemadam kebakaran otomatis yang berusaha memadamkan api jika dideteksi ada kebakaran. Selain itu, pada penelitian ini, sistem hanya terdiri dari satu set sensor yang diletakkan dalam satu model ruangan karena masih berupa prototipe.

Terdapat dua protokol komunikasi yang umum digunakan yaitu HTTP (*HyperText Transfer Protocol*) dan MQTT (*Message Queuing Telemetry Transport*). HTTP merupakan protokol *request/response* dimana

pengguna mengirim *request* ke server, dan server memberikan respon. Pada HTTP, selalu klien yang melakukan *request* [2]. HTTP menggunakan kode untuk mengidentifikasi apakah *request* berhasil atau tidak. Secara umum kode yang berawal dengan angka 2 menandakan bahwa *request* berhasil, 4 menandakan bahwa terdapat kesalahan pada sisi klien, 5 menandakan bahwa terdapat kesalahan pada sisi server.

MQTT adalah protokol pertukaran data yang didesain untuk komunikasi M2M (*Machine to Machine*) yang ringan. Model MQTT adalah model *server-client*, dimana setiap sensor adalah client yang terhubung ke server yang disebut broker melalui TCP (*Transmission Control Protocol*). MQTT merupakan protokol *publish/subscribe*. MQTT berbasis pesan dan setiap pesan dipublikasikan sebagai topik yang dapat diikuti oleh satu atau lebih klien. Setiap klien yang mengikuti suatu topik akan menerima setiap pesan yang dikirimkan ke topik tersebut. Model ini memungkinkan klien MQTT untuk melakukan komunikasi *one to one*, *one to many*, atau *many to one* [3]. Jadi pada protokol MQTT tidak selalu klien yang melakukan *request*.



Gambar 1. Diagram MQTT [3]

Agar pengolahan data mudah, data dikirim dalam format JSON (*JavaScript Object Notation*). Dalam JSON, data terdiri dari pasangan *key* dan *value*. Jika ingin mengakses suatu *value*, maka cukup merujuk ke *key*-nya. Cara penulisan data adalah sebagai berikut: *key* diapit dengan tanda petik dua, kemudian diikuti dengan tanda titik dua dan nilainya (*value*). Setiap data dipisah dengan menggunakan koma, *Value* pada JSON harus berupa string, angka, objek, array, boolean, atau *null* [4].

Pertukaran data antara *device* ataupun *broker* dapat diatur dengan NodeRed. NodeRed adalah alat pemrograman yang dapat dibuka lewat *browser* untuk menyatukan *device*, API (*Application Program Interface*), dan layanan *online* dengan cara yang mudah yaitu dengan menggunakan *nodes* yang dapat dihubungkan satu dengan yang lainnya [5]. Sedangkan untuk pembuatan aplikasi Android untuk monitoring sederhana digunakan MIT App Inventor. MIT App Inventor adalah sebuah *platform* yang dikembangkan oleh staff *Computer Science and Artificial Intelligence* (CSAIL) MIT (Massachusetts Institute of Technology) untuk membuat aplikasi Android dengan cara yang mudah dan sederhana, yaitu dengan menggunakan blok-blok program [6].

ALAT DAN BAHAN

Dari sisi *hardware* dibutuhkan 1 buah *board* Wemos D1 mini, 1 buah sensor suhu dan kelembaban DHT11, 1 buah modul sensor *flame detector*, 1 buah modul pendeteksi asap MQ2, 1 buah relay 2 *channel*, serta 1 buah pompa air kecil untuk demonstrasi sistem pemadam kebakaran otomatis. Sedangkan dari sisi *software* yang meliputi pembuatan aplikasi dan pemrograman *board*, digunakan MIT App Inventor untuk membuat aplikasi Android untuk monitoring sederhana, Espruino Web IDE untuk memrogram Wemos, *broker* publik hiveMQ untuk menyimpan data di server, dan NodeRed untuk mengambil data dari *broker* publik dan mengolahnya menjadi *webpage* untuk protokol HTTP. Selain itu, dibutuhkan juga 2 potong PCB (*Printed Circuit Board*), beberapa kabel *jumper*, 1 buah lilin sebagai sumber api, beberapa korek sebagai sumber asap, 1 kotak kotak seng sebagai model ruangan, dan 1 buah catu daya sebagai sumber daya untuk pompa, *relay*, dan modul pendeteksi asap MQ2.

PEMBUATAN PROTOTIPE

Kalibrasi Sensor Suhu dan Kelembaban DHT11

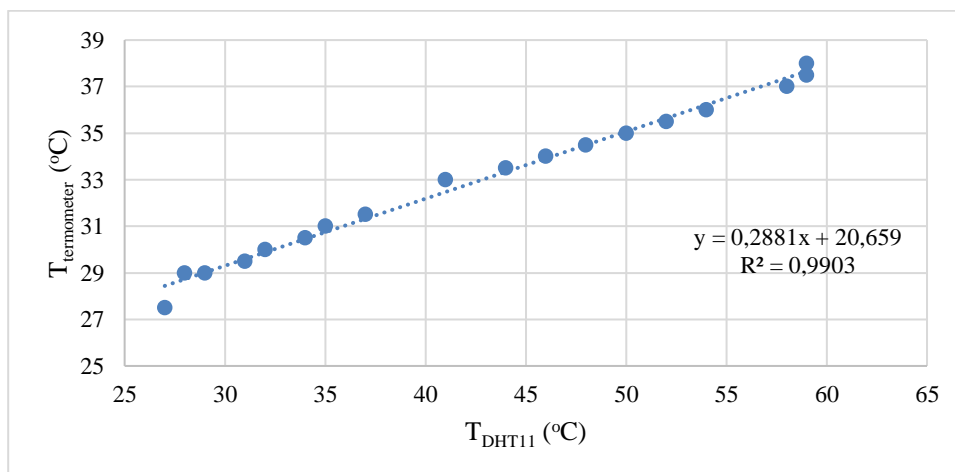
Sebelum digunakan, sensor suhu dan kelembaban (DHT11) dikalibrasi terlebih dahulu. Kalibrasi yang dilakukan adalah kalibrasi suhu dengan tahapan seperti yang dapat dilihat pada diagram alir pada Gambar 2, karena data kelembaban tidak digunakan dalam penentuan keputusan program. Untuk mengkalibrasi suhu DHT11 digunakan termometer raksa yang dimasukkan ke dalam model ruangan. Api kemudian dinyalakan dan data suhu dari termometer dan DHT11 dicatat setiap 10 detik selama 3 menit, sehingga diperoleh data seperti Tabel 1 dibawah.

Tabel 1. Data temperatur DHT11 dan termometer

No.	t (s)	T _{DHT11} (°C)	T _{Termometer} (°C)
1	10	27	27,5
2	20	28	29
3	30	29	29
4	40	31	29,5
5	50	32	30
6	60	34	30,5
7	70	35	31
8	80	37	31,5
9	90	41	33
10	100	44	33,5
11	110	46	34
12	120	48	34,5
13	130	50	35
14	140	52	35,5
15	150	54	36
16	160	58	37
17	170	59	37,5
18	180	59	38

Berdasarkan tabel tersebut diperoleh relasi antara suhu hasil bacaan termometer dan suhu hasil bacaan DHT11 seperti yang dapat dilihat dari Grafik 1. Persamaan hubungan antara suhu bacaan termometer dan DHT11 dapat dilihat pada persamaan (1) dibawah, dengan y adalah suhu akhir dan x adalah suhu bacaan DHT11. (1)

$$y = 0,2881x + 20,659$$



Grafik 1. Hubungan suhu bacaan termometer dengan DHT11

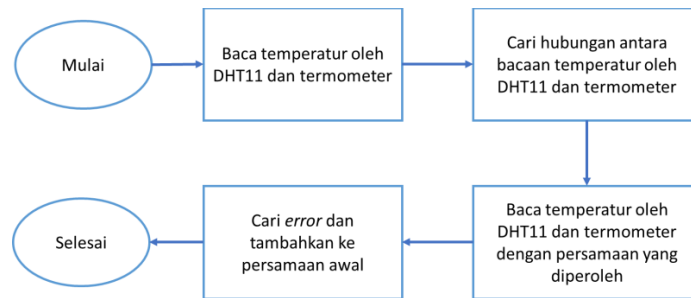
Setelah itu, dilakukan pengambilan data lagi dengan DHT11 yang sudah dikoreksi berdasarkan persamaan (1) setiap 10 detik selama 2 menit untuk mencari *error*, sehingga diperoleh data seperti pada Tabel 2.

Tabel 2. Data temperatur DHT11 yang sudah dikoreksi dan termometer

No.	t (s)	T _{DHT11} (°C)	T _{Termometer} (°C)	Δ (°C)
1	10	28,44	27	-1,44
2	20	28,44	27	-1,44
4	30	28,72	28	-0,72
5	40	29,01	29	-0,01
6	50	29,59	29,5	-0,09
7	60	30,17	29,5	-0,67
8	70	30,45	30	-0,45
9	80	31,03	31	-0,03
10	90	31,89	31,5	-0,39
11	100	32,47	32	-0,47
12	110	33,05	32,5	-0,55
13	120	33,62	33	-0,62

Nilai Δ kemudian dirata-ratakan sehingga diperoleh rata-rata perbedaan suhu antara DHT11 yang sudah dikoreksi dengan termometer raksa, yaitu -0,573. Sehingga temperatur koreksi untuk DHT11 adalah:

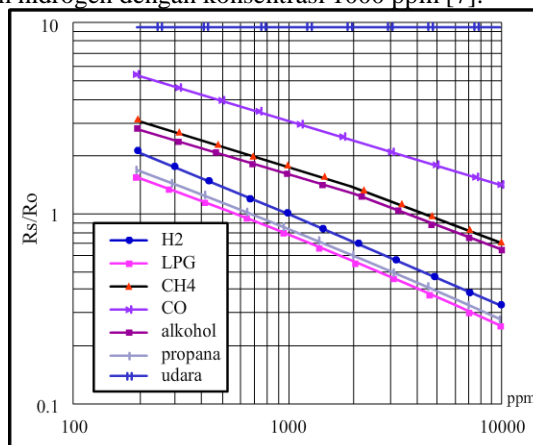
$$y = 0,2881x + 20,659 - 0,573 \tag{2}$$



Gambar 2. Diagram alir kalibrasi DHT11

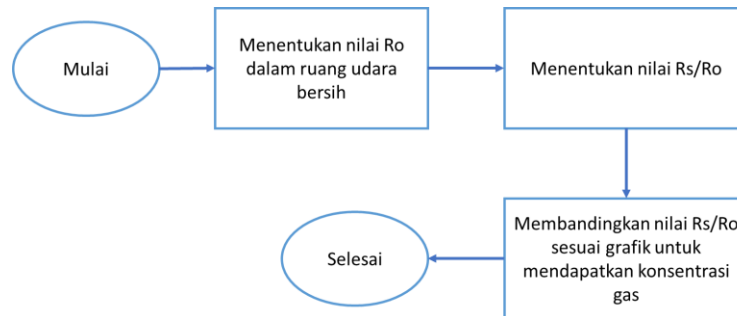
Kalibrasi Sensor Gas (Asap) MQ2

Sensor gas MQ2 tidak dikalibrasi terhadap alat karena tidak tersedianya alat, sehingga penentuan konsentrasi asap dilakukan dengan melihat relasi dari R_s/R_0 terhadap konsentrasi gas asap seperti yang dapat dilihat grafik dalam *datasheet* pada Gambar 3. R_s adalah resistansi sensor dalam gas tertentu, sedangkan R_0 adalah resistansi sensor dalam hidrogen dengan konsentrasi 1000 ppm [7].



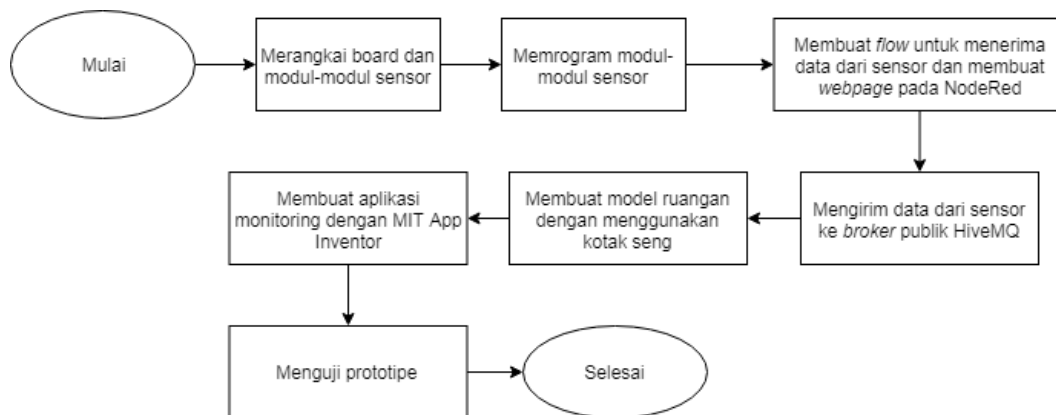
Gambar 3. Grafik R_s/R_0 terhadap konsentrasi gas dalam ppm [7]

Proses kalibrasi dapat dilihat pada Gambar 4. Tujuan dari kalibrasi adalah menemukan nilai R_o yang diturunkan dari R_s . Fungsi utama dari kode kalibrasi adalah menghitung R_o dengan membaca kondisi gas ruangan ketika modul sensor diletakkan dalam ruangan dengan udara bersih. Setelah nilai R_o didapat, konsentrasi dari gas dapat diketahui dari rasio R_s/R_o [8].



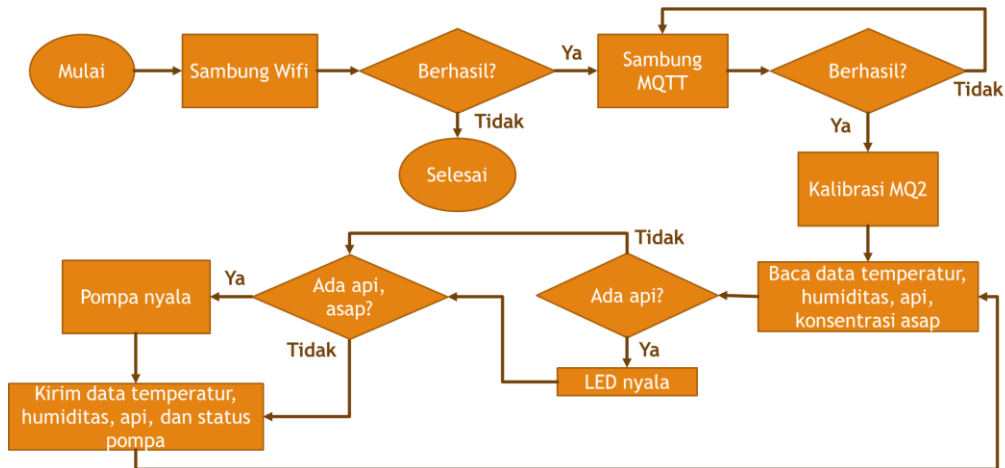
Gambar 4. Diagram alir kalibrasi MQ2

Pembuatan Prototipe



Gambar 5. Diagram alir kerja pembuatan prototipe

Langkah-langkah pembuatan prototipe sistem deteksi dan penanggulangan kebakaran dapat dilihat pada diagram alir pada Gambar 5. Pembuatan prototipe dimulai dari pengujian ketiga sensor yang akan digunakan, serta kalibrasi DHT11 dan MQ2. Setelah sensor diuji satu-persatu dan sensor DHT11 dan MQ2 dikalibrasi, sensor-sensor tersebut dirangkai dalam satu rangkaian. Setelah itu dilakukan pengujian pengiriman data dengan protokol MQTT ke *broker* publik hiveMQ, baru kemudian dibuat sebuah *flow* di NodeRed untuk menerima data dari sensor dan membuat sebuah *webpage* untuk protokol HTTP. Diagram alir program yang membaca sensor dan mengatur pengiriman data dari mikrokontroler ke broker dapat dilihat pada Gambar 6 di bawah sedangkan diagram alir program pada NodeRed dapat dilihat pada Gambar 7.

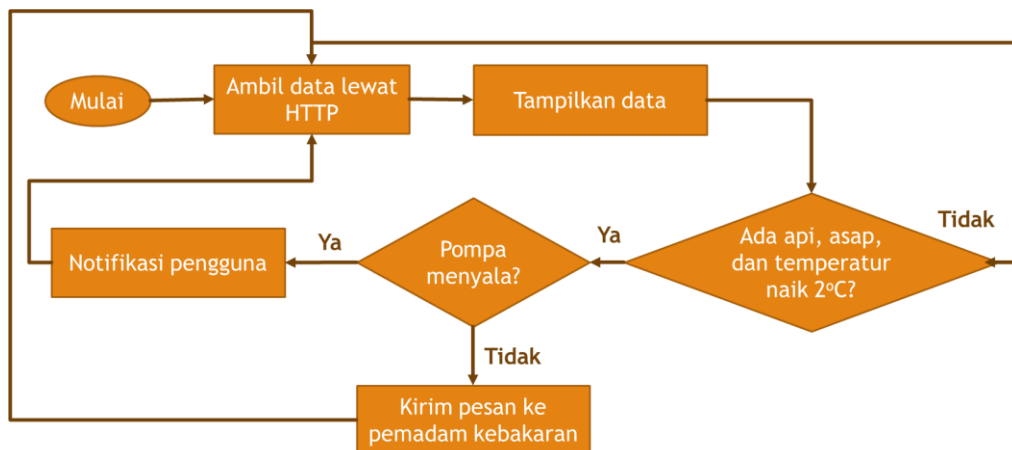


Gambar 6. Diagram alir program pada mikrokontroller



Gambar 7. Diagram alir pada NodeRed

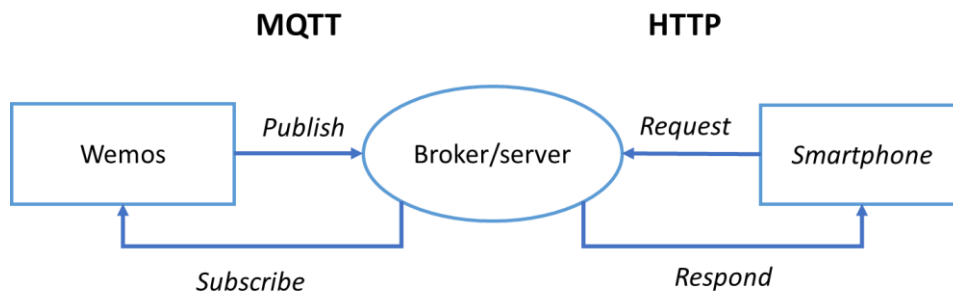
Model ruangan dari kotak seng kemudian dibuat dan rangkaian sensor ditempatkan dalam kotak tersebut. Kemudian dibuat sebuah aplikasi sederhana bernama “FireWatch” dengan menggunakan MIT App Inventor, setelah itu barulah prototipe dapat diuji. Diagram alir untuk program aplikasi dapat dilihat pada Gambar 8, sedangkan aliran sistem prototipe secara keseluruhan dapat dilihat pada Gambar 9. Pada Gambar 8 dapat dilihat bahwa saat aplikasi dinyalakan, aplikasi mengambil data dari HTTP *webpage* yang dibuat oleh NodeRed, kemudian menampilkan data yang didapat pada layar. Jika ada asap, api, dan kenaikan temperatur sebesar 2°C, dan pompa air menyala, maka aplikasi akan menotifikasi pengguna bahwa terjadi kebakaran dan sistem pemadam otomatis menyala. Namun jika pompa tidak menyala, aplikasi akan mengirim pesan ke pemadam kebakaran. Jika tidak dideteksi ada api, asap, dan kenaikan temperatur, maka aplikasi akan terus mengambil data seperti biasanya.



Gambar 8. Diagram alir program aplikasi “FireWatch”

Keseluruhan sistem penanggulangan kebakaran dini dapat dilihat pada Gambar 9 di bawah. Tiga komponen utama dalam sistem ini adalah Wemos yang merupakan mikrokontroller, *broker/server*, dan *smartphone*.

Wemos berperan dalam mengambil data dan mempublikasikan data tersebut ke *broker*, *broker/server* berperan untuk membuat sebuah *webpage* HTTP agar data bisa diambil oleh *smartphone* melalui protokol HTTP, dan *smartphone* berperan untuk menampilkan data yang diambil dari *broker/server* ke pengguna, dan melakukan panggilan ke pemadam kebakaran apabila dideteksi kebakaran dan sistem pemadam otomatis gagal untuk memadamkannya.

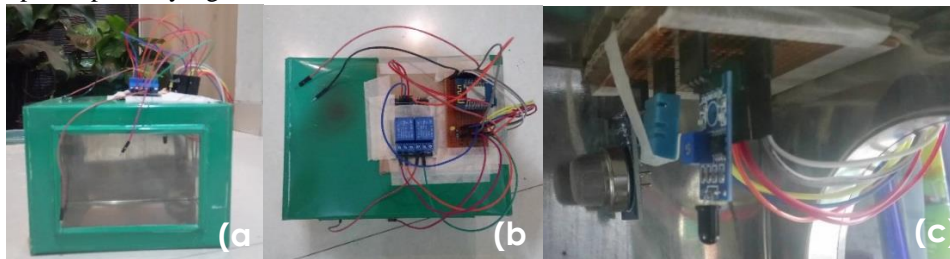


Gambar 9. Flow sistem secara keseluruhan

HASIL PERCOBAAN

Rupa Prototipe

Berikut prototipe alat yang sudah dibuat:



Gambar10. Prototipe alat tampak depan (a), atas (b), dan dalam dengan ketiga modul sensor (c)

Aplikasi Monitoring

Aplikasi yang dinamai “FireWatch” ini dibuat dengan menggunakan MIT App Inventor. “FireWatch” merupakan aplikasi Android yang berfungsi untuk mengambil data sensor dari webpage yang dibuat oleh NodeRed. Untuk memulai pengambilan data tombol “Get Data” harus diklik terlebih dahulu. Ketika aplikasi mendeteksi adanya kemungkinan kebakaran, maka aplikasi akan memperingati pengguna dengan menggetarkan gawai dan menampilkan teks informasi bahwa ada kemungkinan kebakaran yang terjadi. Ketika sistem pemadam otomatis tidak berhasil memadamkan api, maka aplikasi akan memperingati pengguna dan mengirim pesan berupa informasi kebakaran di rumah pengguna ke pemadam kebakaran. Alamat rumah pengguna diisi manual oleh pengguna pada bagian “Home Address”, sedangkan kontak pemadam kebakaran pada bagian “Phone Number” secara *default* adalah kontak pemadam kebakaran pusat, namun dapat diganti oleh pengguna menjadi kontak pemadam kebakaran cabang jika diinginkan. Tombol reset pada kedua bagian tersebut berfungsi untuk mengosongkan textbox pada bagian “Home Address”, dan mengubah nomor telepon menjadi nomor telepon default pada bagian “Phone Number”. Pada percobaan ini, kontak di-set secara *default* adalah nomor telepon pribadi.



Gambar 11. Tampilan aplikasi “FireWatch” menunjukkan deskripsi aplikasi (a), *textbox* untuk mengisi alamat rumah dan nomor telepon (a, b), tombol untuk menginisiasi pengambilan data (b), dan tampilan data (b)

Percobaan dengan Menggunakan Data *Dummy*

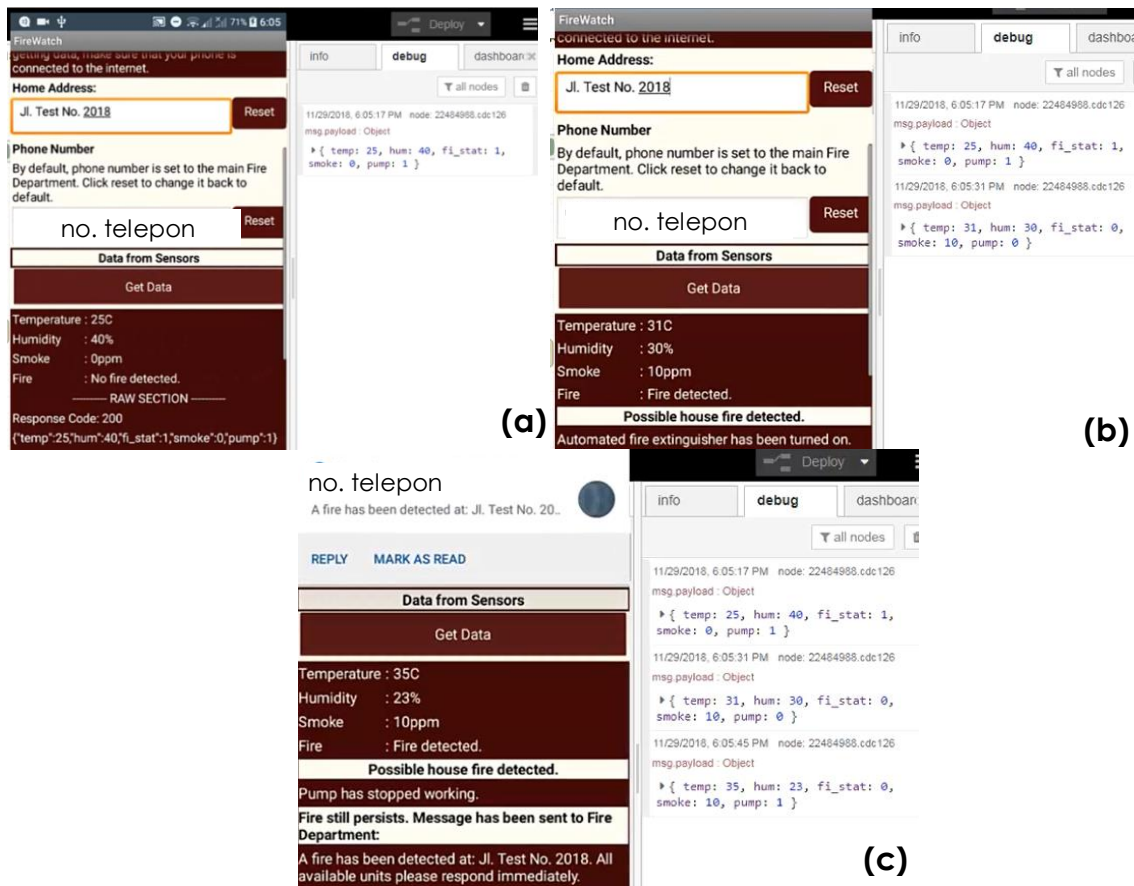
Percobaan ini dilakukan untuk menguji apakah aplikasi Android yang dibuat sudah berjalan sesuai dengan yang diharapkan. Oleh karena itu, pada percobaan ini tidak digunakan data dari sensor melainkan data *dummy*. Data *dummy* yang dikirim adalah data temperatur (“temp”), data humiditas (“hum”), data konsentrasi asap (“smoke”), data status api (“fi_stat”), dan data status pompa (“pump”) seperti yang dapat dilihat pada Gambar 12. Terdapat 3 variasi data yang dapat dikirim, variasi pertama merepresentasikan keadaan normal (tidak kebakaran), variasi kedua merepresentasikan keadaan ketika kebakaran, dan variasi ketiga merepresentasikan keadaan ketika masih terjadi kebakaran ketika sistem pemadam otomatis (pompa air) sudah mati.

```

1 {
2   "temp": 25,
3   "hum": 40,
4   "fi_stat": 1,
5   "smoke": 0,
6   "pump": 1 (1)
7 }
1 {
2   "temp": 31,
3   "hum": 30,
4   "fi_stat": 0,
5   "smoke": 10,
6   "pump": 0 (2)
7 }
1 {
2   "temp": 35,
3   "hum": 23,
4   "fi_stat": 0,
5   "smoke": 10,
6   "pump": 0 (3)
7 }
    
```

Gambar 12. Tiga variasi data yang dikirim (variasi 1, 2, dan 3).

Ketika variasi pertama dikirim, aplikasi hanya menampilkan data sensor seperti biasa. Ketika variasi kedua dikirim, gawai akan bergetar dan muncul status bahwa terdeteksi kebakaran. Ketika variasi ketiga dikirim, aplikasi akan mengirim pesan bahwa ada kebakaran di alamat yang di-input pengguna ke pemadam kebakaran. Pengiriman ketiga variasi data tersebut dapat dilihat pada Gambar 13.

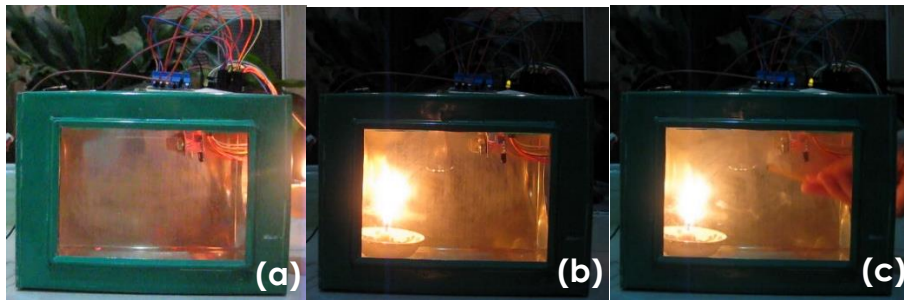


Gambar 13. Pengiriman variasi data pertama (a), kedua (b) dan ketiga (c) beserta dengan keterangan data di NodeRed.

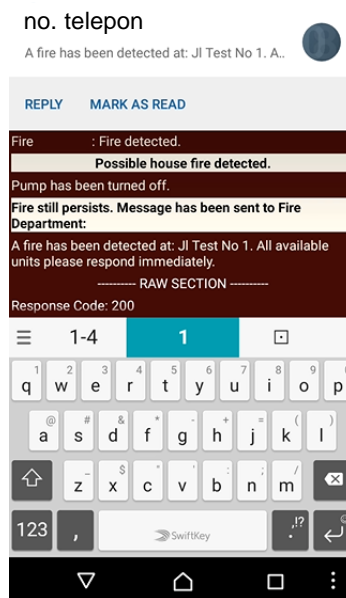
Percobaan dengan Menggunakan Data *Realtime*

Pada percobaan ini digunakan modul-modul sensor serta pompa sebagai pemadam otomatis, namun pompa tidak diletakkan dalam model ruangan untuk menghindari rusaknya modul karena terkena cipratan air. Pertama-tama kode di-*upload* terlebih dahulu dari Espruino Web IDE ke Wemos, kemudian catu daya dinyalakan sebagai sumber daya modul MQ2, relay, dan pompa. Tunggu beberapa saat hingga Wemos berhasil terhubung ke internet dan dapat berkomunikasi dengan *broker*. Setelah itu, jalankan aplikasi “FireWatch” dan tekan tombol “Get Data” untuk mulai mengambil data dari sensor.

Untuk simulasi kebakaran digunakan lilin dan korek api seperti yang dapat dilihat pada Gambar 14. Lilin pertama-tama dinyalakan dan dimasukkan kedalam kotak seng. Setelah dimasukkan, lampu LED yang terdapat pada bagian atas kotak akan menyala, menandakan bahwa terdeteksi api. Kemudian untuk asap digunakan korek api yang dinyalakan dan dipadamkan lalu didekatkan dengan sensor asap. Ketika terdapat konsentrasi asap yang melebihi normal (dalam kasus ini di *set* sebagai 0 ppm), dan api, maka pompa akan menyala. Pompa diatur agar hanya akan menyala selama 3 detik saja, namun pada percobaan ini pompa mati nyala akibat deteksi asap yang kurang baik karena sumber asap tidak menghasilkan asap secara konstan. Jika pompa dalam keadaan mati namun masih dideteksi ada asap dan api, aplikasi akan mengirimkan pesan kepada pemadam kebakaran seperti yang dapat dilihat pada Gambar 15.



Gambar 14. LED (bagian atas kotak seng) menyala ketika dideteksi adanya api (a,b), dan simulasi asap dengan menggunakan korek api (c)



Gambar 15. Pesan terkirim dan diterima

KESIMPULAN

Berdasarkan percobaan yang telah dilakukan dapat disimpulkan bahwa aplikasi dari sistem deteksi dan penganggulangan kebakaran ini berjalan dengan semestinya. Hal ini dapat dilihat dari respon yang sesuai ketika diberi tiga variasi data yang merepresentasikan keadaan normal (tidak ada kebakaran), kebakaran, dan kebakaran namun pompa air sudah mati. Sedangkan pada simulasi dengan menggunakan data *real*, terdapat kendala pada cara produksi asap yang menyebabkan pompa terkadang tidak berjalan sebagaimana mestinya dan memberi data yang keliru pada aplikasi. Untuk mengatasi hal ini, dapat digunakan sumber asap yang dapat menghasilkan asap secara konsisten seperti misalnya kertas yang dibakar, ataupun obat nyamuk bakar.

UCAPAN TERIMAKASIH

Penulis mengucapkan terimakasih kepada Prof. Mitra Djamil selaku dosen mata kuliah Kapita Selekta Instrumentasi dan Sdr. Eko Satria selaku asisten mata kuliah terkait yang telah memberikan saran dan bimbingan dalam pengerjaan prototipe sistem deteksi kebakaran ini. Ucapan terimakasih juga penulis sampaikan kepada Dr. Maman Budiman sebagai dosen lab IoT Fisika ITB yang telah membantu penulis menemukan solusi dalam menggunakan protokol HTTP untuk NodeRed. Penulis juga mengucapkan terima kasih kepada staf Laboratorium Elektronika Fisika ITB yang telah mengizinkan peminjaman tempat dan alat-alat yang dibutuhkan untuk mengerjakan prototipe ini, dan juga kepada teman-teman kelas Kapita Selekta Instrumentasi yang telah membantu dan juga menyemangati penulis dalam pengerjaan prototipe ini. Terakhir, penulis juga mengucapkan terimakasih kepada kedua orang tua penulis yang juga telah menyemangati penulis hingga akhirnya penulis bisa menyelesaikan prototipe ini.

REFERENSI

1. Saeed, Faisal, dkk., *IoT Based Intelligent Modeling of Smart Home Environment for Fire Prevention and Safety*, Journal of Sensors and Actuator Networks, 2018, 7, 11, doi:10.3390/jsan7010011
2. Vangie Beal. *HTTP – HyperText Transfer Protocol*.
<https://www.webopedia.com/TERM/H/HTTP.html>
3. Jaffey, Toby. 2014. MQTT and CoAP, IoT Protocols.
https://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php
4. Anonim. *JSON – Introduction*.
https://www.w3schools.com/js/js_json_intro.asp
5. Anonim. *Node-RED*.
<https://nodered.org/>
6. Anonim. *Connectivity Components – App Inventor for Android*.
<http://ai2.appinventor.mit.edu/reference/components/connectivity.html#Web>
7. Anonim. Datasheet: *MQ-2 Semiconductor Sensor for Combustible Gas*.
8. Kim. 2014. *MQ-2 Smoke/LPG/CO Gas Sensor Module*.
<https://sandboxelectronics.com/?p=165>