

# Sistem Pendeteksi Kebakaran dengan Arduino pada Gudang

Gideon Ekacipta Utama  
Program Studi Informatika  
Fakultas Teknologi Industri  
Universitas Kristen Petra  
Jl. Siwalankerto 121-131  
Surabaya 60236  
Telp. (031)-2983455  
gideon11699@gmail.com

Agustinus Noertjahyana  
Program Studi Informatika  
Fakultas Teknologi Industri  
Universitas Kristen Petra  
Jl. Siwalankerto 121-131  
Surabaya 60236  
Telp. (031)-2983455  
agust@petra.ac.id

Resmana Lim  
Program Studi Teknik Elektro  
Fakultas Teknologi Industri  
Universitas Kristen Petra  
Jl. Siwalankerto 121-131  
Surabaya 60236  
Telp. (031)-2983455  
resmana@petra.ac.id

## ABSTRAK

Kebakaran memang tidak diinginkan tetapi terkadang kebakaran tidak bisa dihindari. Gudang milik perorangan yang ukurannya kecil hingga menengah biasanya kurang mendapat pengawasan yang cukup. Oleh karena itu diperlukan sebuah sistem pintar yang dapat memberi informasi kepada pemilik mengenai keadaan gudang secara *real time* agar pemilik dapat menerima notifikasi secara cepat jika kebakaran terjadi.

*Internet of Things* sebagai platform dimana setiap alat menjadi semakin pintar, setiap proses jadi memiliki kepandaian, dan komunikasi setiap hari semakin informatif. *Internet of Things* melalui perkembangannya telah memberi kita banyak teknologi baru yang memungkinkan komunikasi alat dengan alat, serta alat dengan manusia. Dari teknologi ini memungkinkan kita menciptakan rangkaian alat yang berfungsi untuk mengawasi gudang untuk kita dan memberi kita notifikasi apabila terjadi kebakaran.

Dari pengujian yang dilakukan oleh penulis dapat diketahui bahwa tingkat kepekatan asap yang menyebabkan kebakaran ada di sekitar 5000PPM setelah dibandingkan dengan produk serupa yang ada di pasaran. Dari pengujian juga didapati bahwa rata-rata delay notifikasi adalah 3.8 detik dimana menurut standard ITU-TG.1010 untuk *delay (end to end <10s)* sehingga 3.8 adalah *delay* yang baik dan memenuhi standard tersebut.

**Kata Kunci:** Arduino, Wemos, Alarm Kebakaran

## ABSTRACT

*Fire is undesirable but fire is unavoidable. Individually owned warehouses that are small to medium in size usually do not receive sufficient supervision. Therefore we need a smart system that can provide information to the owner about the state of the warehouse in real time so that the owner can receive notifications quickly in the event of a fire.*

*Internet of Things as a platform where every tool is getting smarter, every process is smart, and communication is getting more informative every day. The Internet of Things through its development has provided us with many new technologies that enable tool-to-tool communication, as well as tool-to-human communication. From this technology allows us to create a suite of tools that function to monitor the warehouse for us and give us notifications in case of fire.*

*From the tests conducted by the author, it can be seen that the level of smoke density that causes fires is around 5000PPM after being compared with similar products on the market. From the test it was also found that the average notification delay is 3.8 seconds which according to the ITU-TG.1010 standard is for delay (end to end <10s) so 3.8 is a good delay and meets these standards.*

**Keywords:** Arduino, Wemos, Fire Alarm.

## 1. PENDAHULUAN

Tidak ada dari kita yang menginginkan kebakaran. Namun terkadang kebakaran adalah sesuatu yang tidak bisa dihindari. Mulai dari arus pendek hingga udara yang terlalu kering seringkali mengakibatkan kebakaran yang tidak terduga. Gudang, sebagai tempat penyimpanan seringkali mendapat sedikit pengawasan. Terlebih gudang barang yang dianggap tidak terlalu berharga seperti kardus atau karet seringkali tidak diawasi setiap saat. Gudang yang tidak diawasi ini bisa saja terbakar karena akibat-akibat diatas dan seringkali terlambat mendapat pengamanan dan menimbulkan kerugian material dan korban jiwa.

*Internet of Things* sebagai platform dimana setiap alat menjadi semakin pintar, setiap proses jadi memiliki kepandaian, dan komunikasi setiap hari semakin informatif. Saat *Internet of Things* masih terus berubah, efek yang ditimbulkan sudah menjadi langkah yang besar dalam memberi solusi bagi berbagai skenario. [4] *Internet of Things* melalui perkembangannya telah memberi kita banyak teknologi baru yang memungkinkan komunikasi alat dengan alat, serta alat dengan manusia. Komunikasi ini dapat terjadi karena adanya *sensor, actuator, microcontroller*, dll. Dari teknologi ini memungkinkan kita menciptakan rangkaian alat yang berfungsi untuk mengawasi gudang untuk kita dan memberi kita notifikasi apabila terjadi kebakaran.

Arduino merupakan *single-board microcontroller opensource* yang memiliki komunitas yang luas, serta memiliki *software* mereka sendiri. Memiliki banyak kelebihan mulai dari harganya yang murah, hingga banyaknya *sensor* yang kompatibel membuat Arduino menjadi pilihan banyak orang dibanding pesaingnya.

Tujuan dari skripsi ini adalah untuk pembuatan suatu sistem yang dapat mendeteksi kebakaran dan memberi notifikasi kepada pemilik dengan segera agar mendapat respons yang cepat, sehingga dapat mengurangi dampak dari kebakaran yang terjadi, baik dari segi lingkungan maupun material.

## 2. LANDASAN TEORI

### 2.1 Tinjauan Pustaka

#### 2.1.1 Wemos D1

Wemos D1 merupakan *module development board* yang berbasis Wi-Fi dari keluarga ESP8266 dimana dapat diprogram menggunakan *software IDE* Arduino. Meskipun bentuk board ini dirancang menyerupai Arduino Uno, namun dari sisi spesifikasi sebenarnya jauh lebih unggul Wemos D1. Salah satunya dikarenakan pada Wemos D1 sudah terimplementasi modul Wi-Fi sehingga memudahkan pengguna dalam memasukkan *code* kedalam modul tersebut.

#### 2.1.2 MQ2 Gas Sensor

*Sensor* yang ditujukan untuk mendeteksi dan merespons gas. MQ2 bekerja dengan cara mendeteksi gas melalui *Metal Oxide Semiconductor* (MOS) yang disebut juga dengan *Chemiresistors* yang dapat mendeteksi perubahan material ketika ada gas yang mengenai material ini. Menggunakan perbedaan voltase yang diterima maka konsentrasi gas dapat dideteksi. MQ2 Gas Sensor dapat mendeteksi LPG, Asap, propana, alkohol, hidrogen, metana, dan karbon monoksida.

MQ2 *smoke sensor* memiliki 4 buah pin yang terdiri dari pin VCC untuk *power*, GND untuk *ground*, pin analog untuk *output* analog, dan pin digital untuk *output* digital. Pada sistem kali ini penulis menggunakan pin analog yang dihubungkan pada analog 0 (AO) pada wemos untuk melihat konsentrasi gas sehingga konsentrasi gas dapat diubah menjadi angka yang akan ditampilkan pada aplikasi.

#### 2.1.3 HC-SR501 PIR Sensor

*Passive Infrared* (PIR) *sensor* bekerja dengan cara mendeteksi pergerakan objek yang memancarkan inframerah, seperti tubuh manusia. Oleh karena itu PIR *sensor* banyak digunakan untuk mendeteksi pergerakan pada suatu tempat atau ruangan. HC-SR501 adalah salah satu jenis PIR *Sensor* yang ada di pasaran. Memiliki harga yang murah dan *sensor* yang efisien, serta memiliki modul yang dapat diatur dalam mendeteksi pergerakan, juga memiliki ukuran yang kecil sehingga cocok digunakan dalam sistem ini.

HC-SR501 PIR *Sensor* memiliki 3 buah pin berupa VCC untuk *power*, GND untuk *ground*, dan pin *output* yang memberi *output* digital. Ada juga dua buah potensiometer yang digunakan untuk mengatur waktu dan jarak. Potensiometer waktu digunakan untuk mengatur berapa lama *sensor* memberi sinyal setelah mendeteksi panas, waktu dapat diatur pada 3-300 detik, sedangkan jarak mengatur jarak kemampuan *sensor* bekerja, jarak dapat diatur 3-7 meter. Selain itu ada juga switch untuk mengatur apakah *sensor* bekerja secara *High* atau *Low*. *High* berarti *sensor* bekerja terus-menerus selama ada *sensor* panas yang ditangkap, sedangkan *Low* berarti *sensor* menyala selama beberapa saat seperti yang diatur pada potensiometer waktu.

#### 2.1.4 DHT 11 Temperature Sensor

*Sensor* DHT11 adalah salah satu jenis *sensor* yang banyak digunakan pada proyek berbasis Arduino untuk membaca suhu (*temperature*) ruangan dan kelembapan udara (*humidity*). *Sensor* ini kecil dan ringkas, serta harganya yang terjangkau. Cara kerja *sensor* ini adalah dengan resistor di dalamnya memiliki nilai resistansi terbalik dengan suhu. Jadi, semakin tinggi suhu ruangan maka semakin kecil nilai resistansinya, demikian sebaliknya. Di dalam *sensor* ini juga terdapat *sensor* kelembapan yang resistif

kepada perubahan kadar air di udara. Kedua data dari *sensor* ini diolah dalam IC *controller* yang akan mengeluarkan *output* dalam bentuk single wire bidirectional.

DHT11 yang digunakan oleh penulis sudah menggunakan modul sehingga hanya ada 3 pin yang tersisa. Pin yang ada merupakan VCC untuk *power*, GND untuk *ground* dan data sebagai *output*. Pin data dihubungkan pada input digital wemos untuk mengetahui nilai suhu dan kelembapan dari besar *output* yang diberikan.

#### 2.1.5 Firebase

*Cloud Firestore* adalah *database* yang fleksibel dan skalabel untuk pengembangan seluler, web, dan server di *Firestore* dan *Google Cloud Platform*. Seperti *Firestore Realtime Database*, *Cloud Firestore* membuat data Anda tetap terhubung di aplikasi klien melalui listener *real time* dan menawarkan dukungan secara *offline* untuk seluler dan web. Dengan begitu, Anda dapat membuat aplikasi yang responsif dan mampu bekerja tanpa harus bergantung pada latensi jaringan atau koneksi Internet. *Cloud Firestore* juga menawarkan integrasi yang lancar dengan produk *Firestore* dan *Google Cloud Platform* lainnya, termasuk *Cloud Functions*. [2]

#### 2.1.6 Android

*Android* adalah sistem operasi berbasis *Linux* yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Sebelum dibeli oleh *Google*, *Android* dikembangkan dan dimiliki oleh *Android, Inc.*, tetapi masih dibantu dana oleh *google* dalam pengembangannya. Sistem operasi ini dirilis resmi tahun 2007. Bersamaan dengan *Open Handset Alliance*, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler.

#### 2.1.7 Kotlin

*Kotlin* adalah bahasa pengembangan dari *Java* yang sejak 2016 menjadi bahasa utama untuk *Android studio*. Di antara perkembangan perangkat lunak *Java* yang penting dalam beberapa tahun terakhir adalah munculnya bahasa alternatif untuk JVM seperti *Groovy*, *Jython*, dan *Java* — dan, baru-baru ini, munculnya *Scala*. Bahasa baru yang diketik secara statis, *Kotlin*, dinamai menurut nama sebuah pulau Rusia di lepas pantai St. Petersburg, tempat tinggal *Andrey Breslav* dari *Kotlin* dan tim *Kotlin*, baru-baru ini mendapat perhatian. Sebuah gagasan dari perusahaan pengembang perangkat lunak Ceko yang sangat dipuji, *JetBrains*, pembuat *Java IDE IntelliJ IDEA*, *Kotlin* dinobatkan sebagai Bahasa Bulan Ini dalam edisi Januari 2012 *Jurnal Dr. Dobbs*. [3].

## 2.2 Tinjauan Studi

### 2.2.1 Rancang Bangun Sistem Pendeteksi Kebakaran Berbasis IoT dan SMS Gateway Menggunakan Arduino [5]

- Pada penelitian ini membahas tentang penggunaan sistem *alarm* kebakaran dengan arduino pada hutan di dataran tinggi.

- Menggunakan metode *Internet of Things*, peneliti membuat sistem dengan Arduino yang dihubungkan dengan *sensor* asap, api, dan suhu lalu menggunakan SIM900 sebagai modul untuk koneksi ke internet. Pengiriman data melalui *Internet of Things* terbukti lebih cepat sehingga informasi cepat tersampaikan dan situasi dapat ditangani dengan segera.

- Prinsip kerja alat pada keadaan awal diaktifkan, *sensor* suhu menampilkan data suhu, jika suhu normal, warna hijau menyala.

Jika *sensor* asap mendeteksi adanya asap, indikator kuning menyala. Jika suhu diatas 45 C, indikator merah menyala. Jika mendeteksi adanya api, maka system akan melakukan panggilan ke pengawas dan petugas yang berjaga.

### 2.2.2 Desain Dan Implementasi Purwarupa Pendeteksi Dini Kebakaran Gedung Menggunakan Aplikasi Mobile Berbasis Android dan Internet of Things (IoT) [1]

- Penelitian yang dilakukan meliputi pembuatan alat dan aplikasi database *ThingSpeak* mendapat rata-rata *delay end to end* sebesar 7.7 detik yang sesuai dengan standard ITU-TG.1010 untuk *delay (end to end <10s)* yang dikategorikan baik.

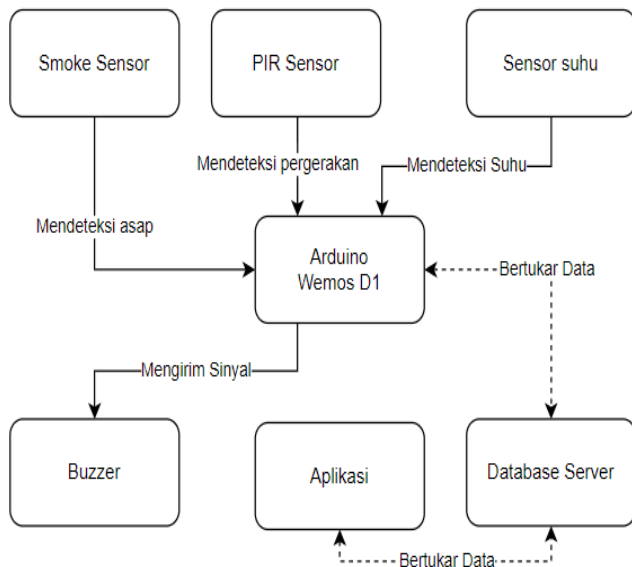
- Penelitian terfokus pada kecepatan aplikasi dalam menerima informasi dari *sensor*, serta memperingatkan pemilik maupun petugas pemadam kebakaran mengenai kebakaran yang terjadi pada gedung.

- Hasil penelitian didapat bahwa aplikasi yang dijalankan melalui database *ThingSpeak* mendapat rata-rata *delay end to end* sebesar 7.7 detik yang sesuai dengan standard ITU-TG.1010 untuk *delay (end to end <10s)* yang dikategorikan baik.

## 3. DESAIN SISTEM

Desain sistem yang ada menggunakan Arduino dan Aplikasi. Pada Arduino desain rangkaian terbagi menjadi lima bagian yaitu desain perangkat *sensor* DHT11, desain perangkat, *sensor* asap MQ2, desain perangkat *Passive Infrared sensor*. Pada aplikasi akan digunakan oleh pengguna untuk melakukan pengecekan alat dan *sensor* serta mendapat notifikasi apabila ada kejadian terdeteksi.

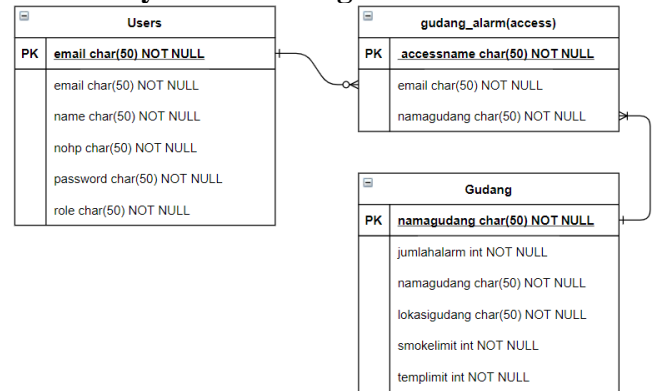
### 3.1 Arsitektur Sistem



Gambar 1. Desain Arsitektur Sistem

Pada gambar 1 desain arsitektur ini terdapat mikrokontroler yang terhubung dengan *buzzer*, *Passive Infrared sensor*, *sensor* suhu, dan *smoke sensor*. Keseluruhan aktivitas mikrokontroler akan diatur dari aplikasi melalui *android*. Dengan adanya *Wi-Fi* yang ada pada *wemos D1* maka seluruh info *sensor* dapat dikirim ke *database* secara *online*.

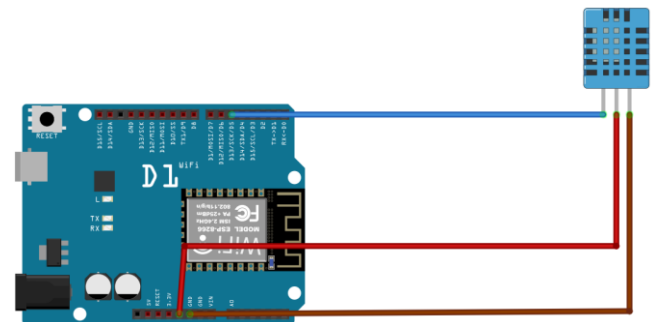
### 3.2 Entity Relation Diagram



Gambar 2. Entity Relation Diagram

Pada desain *Entity Relationship Diagram* yang telah dibuat seperti Gambar 2 terdapat tabel *user*, *gudang* dan *access*. Tabel *user* dan tabel *access* mempunyai relasi yaitu *one to many*, tetapi bisa saja ada *user* yang tidak memiliki *access*. Sedangkan pada tabel *Gudang* dan tabel *access* juga memiliki hubungan atau relasi yang sama, yaitu *one to many*.

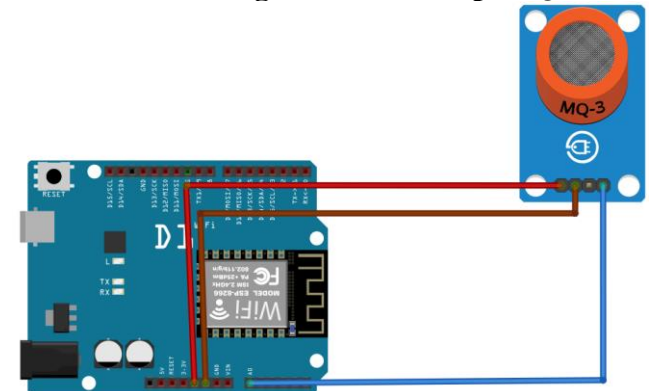
### 3.3 Desain Perangkat Sensor DHT11



Gambar 3. Desain Rangkaian Sensor DHT11

Pada Gambar 3 merupakan desain rangkaian *sensor* suhu dan kelembaban. Pin D5 input *digital* pada *Wemos D1* akan menerima data dari modul *sensor* DHT11 pin data *output* digital. Pin 5V dan Gnd pada *Wemos D1* merupakan sumber daya listrik yang akan diberikan pada modul *sensor* DHT11 melalui pin *Vcc* dan *Gnd*.

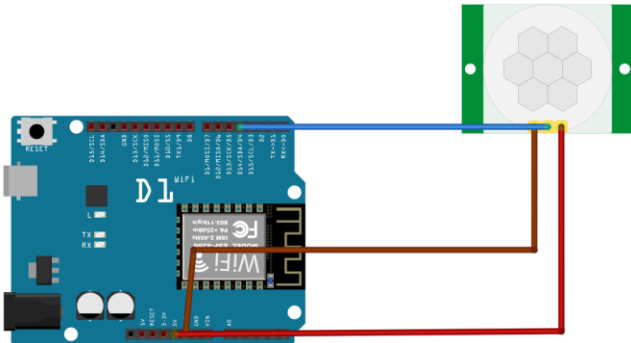
### 3.4 Desain Perangkat Sensor Asap MQ2



Gambar 4. Desain Rangkaian Sensor asap MQ2

Pada Gambar 4 merupakan desain rangkaian *sensor* asap MQ2. Pin A0 input analog pada Wemos D1 akan menerima data dari modul *sensor* MQ2 pin data *output* analog. Pin 5V dan Gnd pada Wemos D1 merupakan sumber daya listrik yang akan diberikan pada modul asap MQ2 melalui pin Vcc dan Gnd.

### 3.5 Desain Perangkat PIR Sensor



Gambar 5. Desain Rangkaian Passive Infrared Sensor

Pada Gambar 5 merupakan desain rangkaian *Passive Infrared* (PIR) *sensor*. Pin D4 *input* digital pada Wemos D1 akan menerima data dari modul PIR *sensor* pin data *output* digital. Pin 5V dan Gnd pada Wemos D1 merupakan sumber daya listrik yang akan diberikan pada modul PIR *sensor* melalui pin Vcc dan Gnd.

## 4. PENGUJIAN SISTEM

### 4.1 Pengujian Banyak Sensor

Program dibuat untuk banyak gudang, setiap gudang dapat memiliki banyak alat. Oleh karena itu pada program dibutuhkan media untuk mendaftarkan alat dan menambahkan akses. Bagian terakhir dari program loop pada coding untuk Wemos adalah mengirim data ke database sesuai dengan path yang diinginkan, “gudangX” dimana X menandai nomor gudang dan “wemosY” dimana Y menandai nomor alat. X dan Y dapat diganti sesuai kebutuhan. Lebih detailnya dapat dilihat pada Segmen Program 1.

#### Segmen Program 1. Memasukkan Sensor Baru

```
//FIREBASE
Firebase.setString("/gudang/gudangX/wemosY/DHT11/Humidity", fireHumid); //setup path to send Humidity readings
Firebase.setString("/gudang/gudangX/wemosY/DHT11/Temperature", fireTemp); //setup path to send Temperature readings
Firebase.setString("/gudang/gudangX/wemosY/MQ2/Smoke", fireSmoke); //setup path to send Smoke readings

Firebase.setString("/gudang/gudangX/wemosY/PIR/Sensor", firePIR); //setup path to send PIR readings

if (Firebase.failed())
{
  Serial.print("pushing /logs failed:");
  Serial.println(Firebase.error());
  return;
}
```

### 4.2 Pengujian Sensor

Sistem diuji dengan membuat ruangan tertutup menggunakan kardus lalu menempatkan alat di bagian atas kardus. Semua komponen dirangkai dan direkatkan kepada bagian atas kardus. Suhu di dalam tempat pengujian naik perlahan, tetapi kepekatannya asap mulai naik dengan cepat. Selain itu pada saat *sensor* asap menunjukkan 5000PPM *sensor* asap pembanding menyala yang menunjukkan standard kepekatannya asap standard akan bunyi saat kepekatannya asap mencapai 5000PPM.

### 4.3 Pengujian Pada Gudang

Pengujian pada gudang dilakukan pada gudang rumah penulis dengan panjang gudang 5 meter dan lebar gudang 3 meter, tinggi 3.5 meter. Sebuah alarm dengan sebuah *sensor* suhu, sebuah *sensor* asap, dan sebuah *sensor* PIR dipasang pada tembok gudang dengan tinggi 2 meter dari lantai. Percobaan dilakukan dengan memasang wadah berisi api di lantai. Dan menunggu respons alarm dan notifikasi pada aplikasi.

Pada saat percobaan penulis memasang limit asap menjadi 250 sehingga respons dapat lebih cepat terlihat, dan alarm bekerja dengan baik. Notifikasi juga berjalan dengan baik dan keluar sesuai dengan *delay* telah dicoba sekitar 3 detik.

### 4.4 Pengujian Notifikasi

Pengujian notifikasi pada aplikasi untuk mengecek berapa lama *delay* dari sistem dihitung dengan melihat berapa lama notifikasi muncul setelah *sensor* mendeteksi suhu diatas rata-rata. Untuk percobaan kali ini penulis mengeset rata-rata suhu maksimal menjadi 32 °C. Percobaan diukur dengan membandingkan kenaikan suhu pada *serial monitor* dan munculnya notifikasi pada *avd* seperti pada tabel 1. Penulis melakukan 5 kali percobaan dan didapat hasil sebagai berikut:

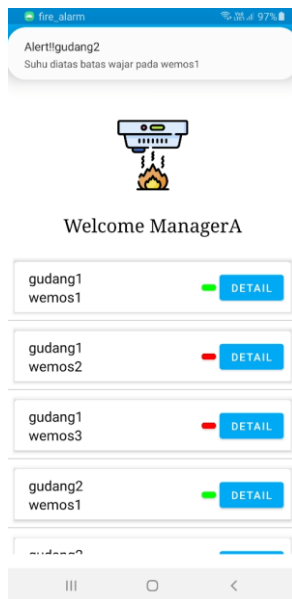
Tabel 1. Hasil Pengujian Notifikasi

Percobaan ke	Sensor mencapai 32 °C pada detik ke	Notifikasi keluar di AVD pada detik ke	Delay notifikasi (detik)
1	5	9	4
2	10	16	6
3	9	13	4
4	9	12	3
5	11	13	2

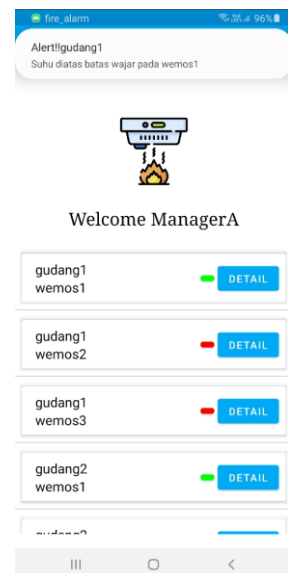
Dari pengujian diatas didapati bahwa rata-*delay* notifikasi adalah 3.8 detik dimana menurut standar ITU-TG.1010 untuk *delay* (end to end <10s) sehingga 3.8 adalah *delay* yang baik dan memenuhi standard tersebut. Selain itu pada percobaan sebelumnya yang dituliskan pada tinjauan studi memiliki hasil *delay* 7.7 detik yang menunjukkan bahwa pengujian kali ini menghasilkan sistem yang lebih baik.

### 4.5 Pengujian Banyak Gudang

Pengujian pada banyak gudang dapat dilihat pada gambar 6 dan gambar 7 dimana dapat dilihat ada notifikasi individual pada setiap alarm. Contoh alarm yang aktif adalah Wemos 1 pada Gudang 2 dan Wemos 1 pada Gudang 1.



Gambar 6. Notifikasi pada Gudang 2



Gambar 7. Notifikasi pada Gudang 1

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Dari hasil pengujian sistem yang telah dilakukan, dapat diambil beberapa kesimpulan yaitu:

- Kepekatan asap yang berpotensi menyebabkan kebakaran adalah 5000PPM sesuai dengan kepekatan saat *sensor* asap pembeding berbunyi.
- Dari pengujian diatas didapati bahwa rata-rata delay notifikasi adalah 3.8 detik dimana menurut standard ITU-TG.1010 untuk delay (end to end <10s) sehingga 3.8 adalah delay yang baik dan memenuhi standard tersebut.

### 5.2 Saran

Saran yang dapat diberikan untuk menyempurnakan dan mengembangkan aplikasi ini lebih lanjut antara lain:

- Perbaiki aplikasi, dan penambahan fitur seperti map gudang dan map ruangan tempat *sensor* berada mungkin bisa ditambahkan ke depannya
- Grafik naik turunnya suhu dan kepekatan asap mungkin dapat ditambahkan kedalam detail alarm.

## 6. DAFTAR REFERENSI

- [1] Ayu, A. S., Ir. Ahmad Tri Hanuranto, M., & Atik Novianti, S. M. 2020. 2.2.2 Desain Dan Implementasi Purwarupa Pendeteksi Dini Kebakaran Gedung Menggunakan Aplikasi Mobile Berbasis Android dan Internet of Things (IoT). *e-Proceeding of Engineering* : Vol.7, 3540.
- [2] Google. 2020, 12 3. Cloud Firestore. Retrieved from Firebase: <https://firebase.google.com/docs/firestore>
- [3] Heiss, J. J. 2013. The Advent of Kotlin: A Conversation with JetBrains' Andrey Breslav. Retrieved from Oracle: <https://www.oracle.com/technical-resources/articles/java/breslav.html>
- [4] Ray, P. P. 2018. A survey on Internet of Things architectures. *Journal of King Saud University - Computer and Information Sciences, Volume 30, Issue 3*, 291-319.
- [5] Sasmoko, D., & Mahendra, A. 2017. 2.2.1 Rancang Bangun Sistem Pendeteksi Kebakaran Berbasis IoT dan SMS Gateway Menggunakan Arduino. *SIMETRIS*, Vol 8 No 2