

PENDETEKSI KEBOCORAN GAS BERBASIS IOT (INTERNET OF THINGS)

TUGAS AKHIR

Disusun oleh:
Muhammad Tio
3311501065

Disusun untuk memenuhi salah satu syarat kelulusan Program Diploma III



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
BATAM
2018

HALAMAN PENGESAHAN
PENDETEKSI KEBOCORAN GAS BERBASIS IOT
(INTERNET OF THINGS)

Disusun oleh:
Muhammad Tio
3311501065

Telah diuji dan dipertahankan di depan Tim Penguji
dalam Sidang Tugas Akhir
pada tanggal 21 Agustus 2018
dan dinyatakan **LULUS**.

Batam, 3 September 2018

Disetujui dan disahkan oleh:

Pembimbing,

Metta Santiputri, S.T, M.Sc
NIP. 197707202012122004

Tim Penguji,

Ketua,

Anggota,

Yeni Rokhayati, S.Si., M.Sc.
NIP. 198602192014042001

Rina Yulius, S.Pd.M.Eng.
NIK. 118199

HALAMAN PERNYATAAN

Dengan ini, saya:

NIM : Muhammad Tio

Nama : 3311501065

adalah mahasiswa Program Studi Teknik Informatika Politeknik Negeri Batam menyatakan bahwa Tugas Akhir dengan judul:

PENDETEKSI KEBOCORAN GAS BERBASIS IOT (INTERNET OF THINGS)

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Negeri Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 3 September 2018

Muhammad Tio
3311501065

ABSTRAK

PENDETEKSI KEBOCORAN GAS BERBASIS IOT (INTERNET OF THINGS)

Gas elpiji merupakan salah satu kebutuhan rumah tangga dalam kehidupan sehari-hari. Setelah keluarnya Keputusan Menteri ESDM No: 1971/26/MEM/2007 tanggal 22 Mei 2007 yang berisi konversi penggunaan minyak bumi (minyak tanah) menjadi gas elpiji sebagai upaya untuk beralih dari keterbatasan sumber daya alam dari energi fosil ke sumber daya alam yang lebih melimpah yaitu gas elpiji. Gas elpiji memiliki kekurangan yaitu mudah terbakar jika terkena percikan api. Sudah banyak terjadi kecelakaan akibat kebocoran dari penggunaan gas elpiji. Pemicunya juga bermacam-macam mulai dari pemasangan yang kurang tepat, sampai penggunaan tabung gas yang sudah tidak layak dipakai.

Penelitian ini memberikan solusi akan permasalahan tersebut dengan memanfaatkan sensor gas yang terhubung ke NodeMCU. Perangkat akan selalu melakukan monitoring area. Sensor gas akan memberikan data ke dalam sistem NodeMCU, kemudian dari hasil tersebut akan ditampilkan sebuah peringatan kepada pengguna melalui perangkat smartphone (Android). Dari perangkat tersebut pengguna bisa mencegah kecelakaan yang terjadi akibat kebocoran gas yang terjadi dan bisa melakukan penanganan yang tepat agar kecelakaan tidak terjadi.

Kata kunci: *NodeMCU, Android, monitoring, Gas Elpiji.*

ABSTRACT

IOT BASED GAS LEAK DETECTION DEVICE

LPG gas is one of the household things in daily life. The decree of the Ministry of Energy and Mineral Resource number 1971/26/MEM/2007 dated May 22, 2007 contains the requirement of the conversion from the petroleum (kerosene) into LPG as an effort to shift from the limited natural resources from fossil energy to the more abundant natural resources. However, LPG gas is highly flammable. There have been many accidents due to leakage of LPG gas. The trigger varies from the incorrect installation to the use of defective gas cylinders.

This paper aims to provide a solution to this problem by building a device that utilizing sensors connected to NodeMCU. The device performs area monitoring continuously. The gas sensor provides data to NodeMCU, and then the results are displayed as a warning to the user via an Android-based smartphone device. Using this device, users will be able to prevent accidents that occur due to gas leaks so that accidents can be avoided.

Keywords: NodeMCU, Android, monitoring, LPG Gas

KATA PENGANTAR

Puji syukur penulis ucapkan, karena berkat rahmat dan hidayah Allah SWT yang telah memberikan kelancaran dan kemudahan sehingga penulis dapat menyelesaikan Tugas Akhir 1 dengan judul "Pendeteksi Kebocoran Gas Berbasis IoT (Internet of Things) sebagai syarat untuk memenuhi kelulusan mata kuliah Workshop III.

Dalam menyelesaikan Tugas Akhir 1 penulis banyak mendapatkan bimbingan dan dukungan dari berbagai pihak, sehingga penulis mengucapkan terima kasih yang sebesar-besarnya kepada pihak yang telah membantu sehingga Tugas Akhir 1 dapat terselesaikan khususnya kepada :

1. Kedua orang tua yang selalu mendoakan dan memberi dukungan moral maupun material;
2. Ibu Metta Santiputri, S.T, M.Sc sebagai dosen pembimbing yang telah membimbing, memberikan masukan dan dukungan dalam menyelesaikan Tugas Akhir 1.
3. Teman-teman seperjuangan yang telah memberikan semangat kepada penulis untuk menyelesaikan Tugas Akhir 1.

Penulis menyadari bahwa dalam penulisan Tugas Akhir 1 masih banyak kekurangan. Oleh sebab itu segala kritik dan saran, penulis harapkan demi kemajuan dan perbaikan dimasa mendatang. Semoga Tugas Akhir 1 ini dapat bermanfaat bagi pembaca maupun penulis, amin.

Batam, 3 September 2018

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN.....	Error! Bookmark not defined.
HALAMAN PERNYATAAN	iii
ABSTRAK	iv
KATA PENGANTAR	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	x
PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah	2
1.4. Tujuan.....	2
1.5. Manfaat.....	2
1.6. Sistematika Penulisan	3
LANDASAN TEORI.....	4
2.1. Tinjauan Pustaka.....	4
2.2. Dasar Teori	5
ANALISIS DAN PERANCANGAN.....	9
3.1. Deskripsi Umum Sistem.....	9
3.2. Kebutuhan Fungsional.....	10
3.3. Kebutuhan Non Fungsional	10
3.4. Perancangan Sistem.....	11
3.4.1. Perancangan Perangkat Keras	11
3.4.2. Perancangan Perangkat Lunak	16
HASIL DAN PEMBAHASAN.....	26
4.1. Hasil Implementasi	26
4.1.1 Implementasi Perangkat Keras.....	26
4.1.2 Implementasi Perangkat Lunak.....	28

4.2. Hasil Pengujian.....	42
4.2.1. Deskripsi Pengujian	42
4.2.2. Hasil Pengujian	43
KESIMPULAN DAN SARAN.....	47
5.1. Kesimpulan	47
5.2. Saran	48

DAFTAR GAMBAR

Gambar 1. Sensor Gas MQ2	6
Gambar 2. Sensor Api	7
Gambar 3. Grafik hasil Sensor Gas MQ2	8
Gambar 4. Deskripsi Umum Sistem.....	9
Gambar 5. Blok Diagram Perangkat Pendeteksi Kebocoran Gas	11
Gambar 6. Use Case Diagram.....	16
Gambar 7. Sequence Diagram Melihat notifikasi	17
Gambar 8. Sequence Diagram Melihat nilai kandungan gas dan nyala api.....	18
Gambar 9. Sequence Diagram Mengelola data penerima notifikasi.....	19
Gambar 10. Sequence Diagram Mengelola data pengaturan	20
Gambar 11. Class Diagram	21
Gambar 12. Perancangan Antarmuka (Tambah List Penerima & Update List Penerima)	22
Gambar 13. Perancangan Antarmuka (List Penerima Notifikasi)	23
Gambar 14. Perancangan Antarmuka (Pengaturan).....	24
Gambar 15. Perancangan Antarmuka (Alarm).....	25
Gambar 16. Perangkat Pendeteksi Kebocoran Gas (Luar).....	26
Gambar 17. Perangkat Pendeteksi Kebocoran Gas (Dalam)	26
Gambar 18. Halaman Pengaturan	28
Gambar 19. Halaman List Penerima	30
Gambar 20. Gambar Halaman Tambah List Penerima	32
Gambar 21. Gambar Halaman Update List Penerima.....	34
Gambar 22. Notifikasi pada bar Android.....	36
Gambar 23. Notifikasi berupa Alarm pada Android	38
Gambar 24. Notifikasi SMS	40

DAFTAR TABEL

Tabel 1. Perbandingan Aplikasi Sejenis.....	4
Tabel 2. Spesifikasi NodeMCU V2	12
Tabel 3. Spesifikasi MQ2 Gas Sensor.....	13
Tabel 4. Spesifikasi Sensor Api	13
Tabel 5. Tabel Keputusan	14
Tabel 6. Uji Pendaftaran pada Perangkat Keras.....	43
Tabel 7. Uji Sensor.....	44
Tabel 8. Uji Pendaftaran pada Perangkat Lunak.....	45

BAB I

PENDAHULUAN

1.1.Latar Belakang

Gas elpiji merupakan salah satu kebutuhan rumah tangga dalam kehidupan sehari-hari. Gas elpiji digunakan sebagai alternatif dari minyak bumi (minyak tanah) yang saat ini semakin sedikit jumlahnya. Dengan peralihan tersebut timbul beberapa masalah yang diakibatkan dari kebocoran gas yang dapat mengakibatkan kecelakaan. Untuk mencegah hal tersebut terjadi dibutuhkan sebuah perangkat yang dapat menangani kebocoran gas.

Dalam perangkat yang telah dibuat sebelumnya hanya mendeteksi kebocoran gas melalui sensor gas. Metode ini masih memiliki beberapa kekurangan, salah satu contohnya adalah jika terjadi kasus kebocoran gas yang menyebabkan penghuni rumah mengalami pingsan, maka fungsi dari perangkat tersebut menjadi tidak berguna sehingga perlu adanya bantuan dari pihak lain untuk mencegah kasus tersebut terjadi. Oleh karena itu dibutuhkan perangkat tambahan seperti sensor pendeteksi manusia dan sensor pendeteksi api sehingga pencegahan terjadinya kecelakaan dapat menjadi lebih efisien, cepat dan tepat.

Teknologi internet dapat membuat komunikasi antar perangkat menjadi lebih cepat dan dengan memanfaatkan konektivitas secara berkala membuat perangkat yang digunakan menjadi solusi yang sesuai dari permasalahan yang ada.

Sehingga diusulkan teknologi yang tepat untuk permasalahan tersebut dengan membangun sebuah alat “**Pendeteksi Kebocoran Gas Berbasis IoT (Internet of Things)**” yang akan memantau kandungan gas yang mudah terbakar di udara, keberadaan manusia, dan adanya api dalam rumah secara berkala. Diharapkan kedepannya kecelakaan yang terjadi di masyarakat menjadi berkurang dan tidak menyebabkan kerugian yang besar.

1.2.Rumusan Masalah

Rumusan masalah yang akan diambil untuk membangun aplikasi adalah:

1. Bagaimana cara mendeteksi kebocoran gas dengan NodeMCU yang terhubung ke perangkat Android?
2. Bagaimana cara mengetahui tingkat kandungan gas pada udara?

1.3.Batasan Masalah

1. Menggunakan NodeMCU sebagai micro-controller.
2. Menggunakan bahasa pemrograman Arduino dibantu dengan aplikasi berbasis android yang dibuat menggunakan Android Studio dengan menggunakan komunikasi berbasis TCP/IP
3. Parameter yang digunakan adalah kadar metana, butana dan LPG dalam udara.

1.4.Tujuan

Tujuan dalam tugas akhir ini adalah:

1. Melakukan perancangan sistem deteksi kebocoran gas agar ketika terjadi kebocoran gas, pengguna dapat melakukan langkah pencegahan yang diketahui dari monitoring aplikasi.
2. Menghasilkan aplikasi Android untuk dihubungkan ke perangkat pendeteksi kebocoran gas.

1.5.Manfaat

Manfaat dari penulisan tugas akhir ini adalah:

1. Mencegah terjadinya kecelakaan karena kebocoran gas elpiji yang sering terjadi di masyarakat.
2. Sebagai sarana penelitian yang berkesinambungan.

1.6.Sistematika Penulisan

Sistematika penulisan disajikan untuk menajikan gambaran mengenai permasalahan yang akan dibahas dalam penelitian ini. Gambaran tentang isi dari penulisan ini yang terdiri dari:

BAB I PENDAHULUAN

Berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

Berisi penelitian serupa yang pernah dilakukan sebelumnya. Selain itu pada bab ini juga menjelaskan tentang teori-teori yang berhubungan dengan penelitian ini seperti

BAB III ANALISIS DAN PERANCANGAN

Menjelaskan analisis deskripsi umum sistem, perancangan sistem, objek sampel, alat dan bahan.

BAB II LANDASAN TEORI

2.1. Tinjauan Pustaka

Dalam membangun aplikasi, mengacu kepada aplikasi sejenis serta membandingkan dengan aplikasi yang akan dibuat.

Tabel 2.1. Perbandingan Aplikasi Sejenis

No	Fitur	Rancang Bangun Data Logging Berbasis Web Server Pada Robot Balon Udara Untuk Deteksi Kebocoran Pipa Gas (Bagus, A.S, 2016)	Implementation of Smart Home Automation System through E-mail using Raspberry Pi and Sensors (Shruti G.S, 2016)	Alat Monitoring Gas/Asap Pada Ruangan Berbasis Relay Raspberry (Muddassir, 2017)	Sistem yang akan dibangun (Pendeteksi kebocoran gas berbasis IoT)
1	Fungsi monitoring kandungan gas di udara	Ada, dengan menggunakan robot balon untuk mendeteksi kebocoran gas	Ada	Ada	Ada
2	Fungsi mendeteksi nyala api	Tidak ada	Tidak ada	Ada	Ada
4	Fungsi menampilkan keadaan gas di udara dalam bentuk grafik	Ada	Tidak ada	Tidak ada	Tidak ada
5	Jenis Peringatan	Mengirimkan notifikasi beserta grafik melalui aplikasi berbasis web	Mengirimkan notifikasi melalui email	Menampilkan peringatan berupa suara dari sirine.	Mengirimkan notifikasi ke perangkat Android.

2.2.Dasar Teori

2.2.1 IoT (Internet of Things)

Menurut Casagras (*Coordinator and support action for global RFID-related activities and standadisation*) mendefinisikan Internet of Things sebagai sebuah infrastruktur jaringan global, yang menghubungkan benda-benda fisik dan virtual melalui eksploitasi data capture dan kemampuan komunikasi. IoT merupakan sebuah cara untuk mengkomunikasikan data yang berasal dari sebuah perangkat ke perangkat yang lain melalui media internet.

Salah satu contoh implementasi IoT adalah dengan mengkomunikasikan data dari NodeMCU ke aplikasi pada perangkat Android. Data tersebut berasal dari pengukuran kandungan gas dalam udara menggunakan sensor gas yang terhubung ke NodeMCU. Pengiriman data ini dilakukan melalui media internet sehingga kedua perangkat harus selalu terhubung ke internet agar komunikasi antar perangkat dapat berjalan dengan baik.

2.2.2 NodeMCU

NodeMCU adalah sebuah platform IoT yang bersifat opensource. Terdiri dari perangkat keras berupa *System on Chip* ESP8266 dari ESP8366 buatan Espressif System, juga *firmware* yang digunakan, yang menggunakan Bahasa pemrograman *scripting* Lua. NodeMCU dapat dianalogikan sebagai *micro-controller* Arduino yang digabung dengan modul ESP8266. NodeMCU memiliki kelebihan dibandingkan dengan *micro-controller* lainnya yaitu:

- NodeMCU memiliki kapabilitas akses terhadap WiFi, contohnya yaitu kemampuan untuk terhubung dengan WiFi yang ada dalam jangkauan serta perangkat ini juga dapat membuat sebuah *access-point* yang dapat digunakan oleh perangkat lain.

- Menggunakan komunikasi USB to serial. Untuk melakukan pemrograman pada perangkat ini dapat dilakukan dengan menghubungkan perangkat pada komputer dengan menggunakan kabel USB yang biasanya digunakan pada *smartphone*, yaitu kabel USB dengan tipe Micro-B.
- Ukurannya yang kecil namun memiliki fitur-fitur yang lengkap sebagai sebuah *micro-controller* membuat NodeMCU sangat cocok digunakan untuk pengembangan perangkat IoT (*Internet of Things*).

2.2.3 Sensor Gas MQ2

Sensor gas adalah sebuah modul yang digunakan pada micro-controller untuk mendeteksi gas pada udara. Ada beberapa tipe sensor gas dan gas yang dapat diidentifikasi oleh modul tersebut juga berbeda-beda. Untuk mendeteksi gas-gas rumah tangga digunakan sensor gas tipe M2 untuk mendeteksi Metana, Butana, LPG dan Asap Rokok.

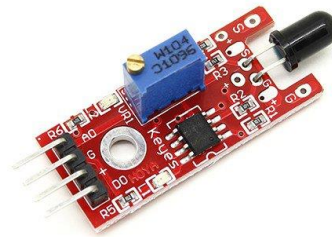


Gambar 2.1. Sensor Gas MQ2

Cara kerja dari sensor gas ini adalah dengan menggunakan rangkaian elektronik yang sensitif terhadap kandungan gas diudara, ketika konsentrasi gas diudara semakin tinggi maka hasil tegangan listrik yang dihasilkan juga semakin tinggi.

2.2.4 Sensor Api

Menurut Yayan (2016), Sensor api adalah sensor yang dapat mendeteksi keberadaan api dan mengubahnya menjadi besaran analog representasinya. Sensor api ini berbeda dengan sensor panas, sensor panas menggunakan parameter temperatur sedangkan sensor api yang diukur adalah nyala apinya.



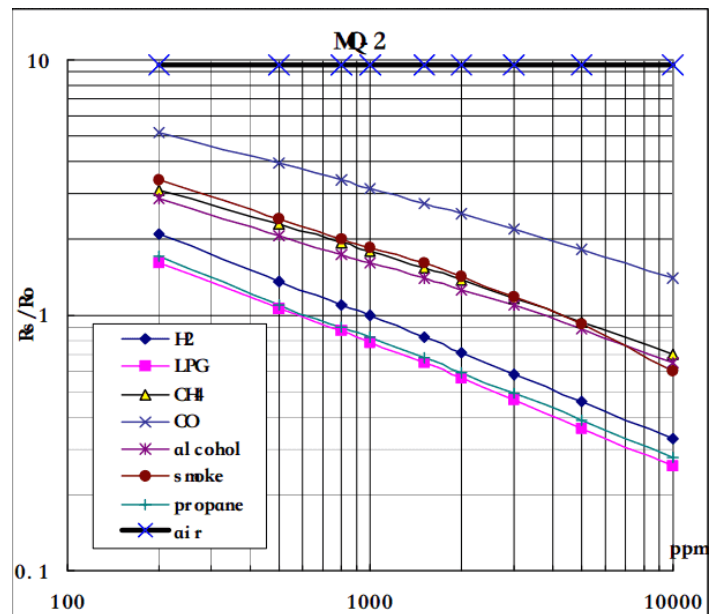
Gambar 2.2. Sensor Api

2.2.5 Android

Menurut Nazruddin (2014), Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Sistem operasi ini bersifat *open-source* yang artinya pengembang dapat membuat dan mengembangkan aplikasi mereka sendiri untuk digunakan pada perangkat yang menggunakan sistem operasi Android.

Android merupakan sistem operasi yang sangat terkenal dan mudah diterima oleh pasar dan cepat berkembang. Hal ini dikarenakan android menggunakan Bahasa pemrograman Java dan bersifat *open-source*. Dengan kemudahan untuk mengembangkan aplikasi pada sistem operasi ini, banyak para pengembang aplikasi membangun aplikasi Android.

2.2.6 Data yang dihasilkan



Gambar 2.3. Grafik hasil Sensor Gas MQ2

(sumber: http://wiki.seeed.cc/Grove-Gas_Sensor-MQ2/)

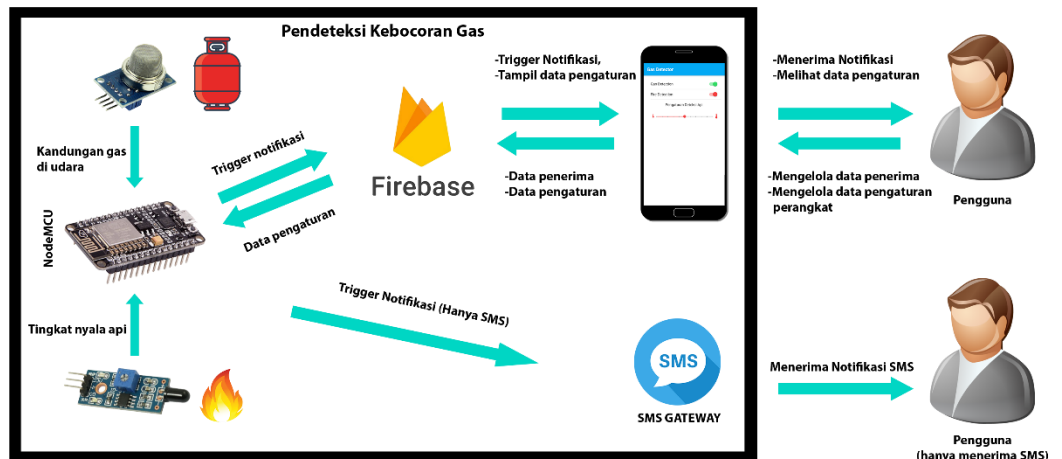
Data yang dihasilkan dari perangkat pendeteksi kebocoran gas adalah berupa nilai kandungan gas diudara yang menggunakan satuan *ppm* (*parts per million*). Gambar 2.3 adalah contoh hasil dari pengukuran kandungan gas di udara dengan menggunakan sensor gas MQ2 dimana pengukuran tersebut dilakukan pada kondisi udara yang normal.

BAB III ANALISIS DAN PERANCANGAN

3.1. Deskripsi Umum Sistem

Pada aplikasi ini hanya terdapat satu *user* atau pengguna yang dapat menggunakan perangkat ini dan satu pengguna yang hanya menerima notifikasi melalui SMS. Pada Gambar 3.1 perangkat pendeteksi kebocoran gas akan mengelola kandungan gas di udara dan nilai nyala api sebelum ditampilkan kepada pengguna.

Alur kerja dari sistem ini yaitu sensor gas dan sensor api yang terpasang pada perangkat NodeMCU akan terus bekerja untuk memeriksa kandungan gas di udara dan nyala api yang ada. Ketika kandungan gas di udara melebihi batas normal ataupun terdeteksi nilai nyala api, maka perangkat akan mengirimkan notifikasi kepada pengguna.



Gambar 3.1. Deskripsi Umum Sistem

3.2.Kebutuhan Fungsional

Berikut Kebutuhan Fungsional yang harus dimiliki oleh sistem:

F001. Mengukur kandungan gas di udara

F002. Mendeteksi api di lingkungan sekitar

F003. Mengirim Notifikasi (Perangkat Keras)

F004. Menerima Notifikasi (Perangkat Lunak)

F005. Pengguna mengelola data penerima notifikasi (Perangkat Lunak)

3.3.Kebutuhan Non Fungsional

Berikut Kebutuhan Non Fungsional sistem:

NF001. Antarmuka sistem mudah dipahami

NF002. Sistem menggunakan Bahasa Inggris dan Bahasa Indonesia

3.4. Perancangan Sistem

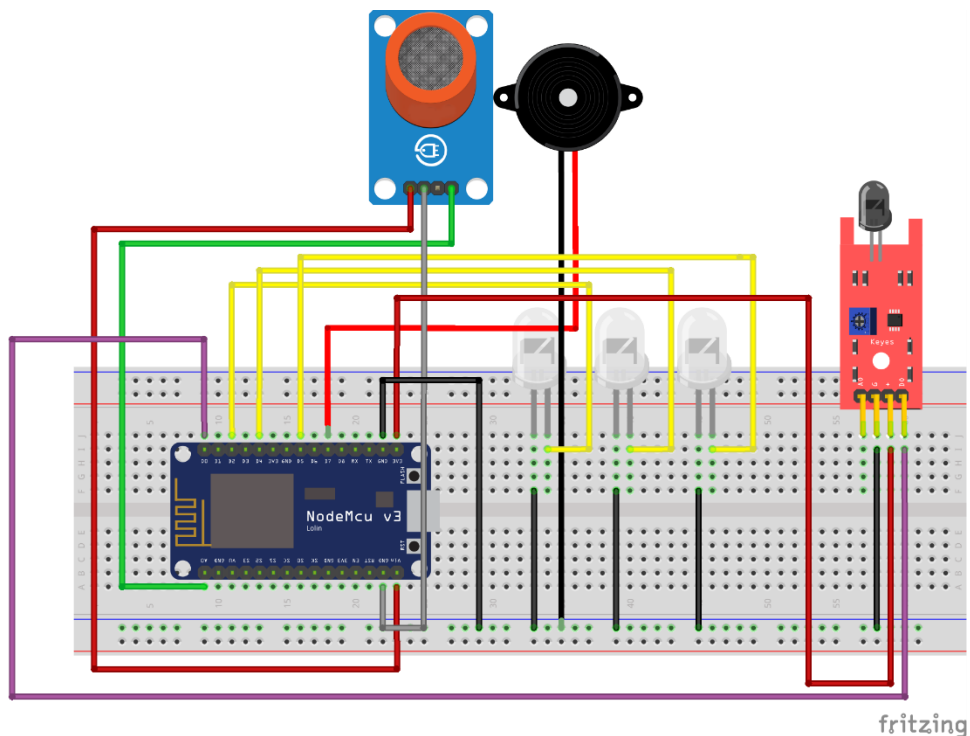
Sistem yang dibangun akan terdiri dari dua bagian, yaitu perangkat keras dan perangkat lunak.

3.4.1. Perancangan Perangkat Keras

Pada perancangan perangkat keras, sistem yang akan dibangun adalah untuk mendeteksi kandungan gas pada udara dengan menggunakan sensor gas dan mendeteksi nilai nyala api dengan menggunakan sensor api yang terhubung dengan NodeMCU.

3.4.1.1. Blok Diagram

NodeMCU akan terhubung dengan sensor api dan sensor gas. Kedua sensor ini akan mengirimkan data yang diperoleh ke perangkat NodeMCU.



Gambar 3.2. Blok Diagram Perangkat Pendeteksi Kebocoran Gas

3.4.1.2. Spesifikasi Perangkat Keras

Beberapa alat yang digunakan untuk membangun perangkat pendeteksi kebocoran gas diantaranya:

1. NodeMCU

Tabel 3.1. Spesifikasi NodeMCU V2

<i>System on Chip (SOC)</i>	NodeMCU DEVKIT 1.0 ESP8266, Wi-Fi SoC, ESP-12 module
<i>Center Processing Unit</i>	ESP8266 (LX106)
<i>Operating System</i>	XTOS
<i>Memory</i>	128 Kbytes
<i>Storage</i>	4 MB
<i>Onboard Network</i>	802.11b/g/n wireless standards
<i>Low-level Peripherals</i>	10× GPIO Pin
<i>Transfer rate</i>	110 – 460800 bps
<i>Operating temperature</i>	-40 °C ~ + 125 °C
<i>Power consumption</i>	On Keep sending data ~ 70 mA (200mA MAX) Standby: < 200 mA
Suplai Daya	4.5V ~ 9V (10V Max), USB-Powered, providing USB debugging interface
Ukuran	47 mm × 31 mm
Berat	7 gram

2. MQ 2 Gas Sensor

Tabel 3.2. Spesifikasi MQ2 Gas Sensor

Item	Parameter	Min	Typical	Max	Unit
VCC	<i>Working Voltage</i>	4.9	5	5.1	V
PH	<i>Heating consumption</i>	0.5	-	800	mW
RL	<i>Load resistance</i>		<i>adjustable</i>		
RH	<i>Heater resistance</i>	-	33	-	Ω
Rs	<i>Sensing Resistance</i>	3	-	30	k Ω

3. Sensor Api

Tabel 3.3. Spesifikasi Sensor Api

Item	Min	Typical	Max	Unit
<i>Voltage</i>	4.75	5.0	5.30	VDC
<i>Current</i>	/	20	/	mA
<i>Range of Spectral Bandwidth</i>	760	940	1100	nm
<i>Detection range</i>	0	~	1	m
<i>Response Time</i>	15			μ S
<i>Operating Temperature</i>	-25	~	85	$^{\circ}$ C

3.4.1.3. Tabel Keputusan

Tabel 3.4. Tabel Keputusan

Kondisi	TC1	TC2	TC3	TC4
Gas Sensor aktif	1	1	0	0
Sensor api aktif	0	1	1	0
Aksi				
Mengirimkan notifikasi ke pengguna	1	0	0	0
Mengaktifkan alarm pada perangkat android	0	1	1	0
Mengaktifkan alarm pada perangkat NodeMCU	0	1	0	0

NodeMCU terhubung dengan 2 sensor yaitu sensor gas dan sensor api. Oleh karena itu terdapat beberapa kondisi yang akan terjadi dan dengan kondisi tersebut sistem akan melakukan aksi sesuai dengan kondisi yang berlaku. Pada bagian kondisi nilai 1 berarti sensor mendeteksi kandungan bahan yang melebihi batas dan nilai 0 berarti kandungan bahan pada lingkungan sekitar masih dalam batas normal. Pada bagian aksi nilai 1 berarti aksi tersebut akan dilakukan oleh sistem dan nilai 0 berarti aksi tidak akan dilakukan.

Notifikasi yang akan dihasilkan dari sistem Pendeteksi Kebocoran Gas terbagi menjadi 3 macam yaitu:

1. Notifikasi Bar pada perangkat Android

Notifikasi ini hanya berupa pesan pada perangkat Android dan tanpa adanya getaran ataupun suara. Pesan ini muncul seperti pesan notifikasi biasa yang sering muncul pada perangkat Android.

2. Notifikasi berupa Alarm

Notifikasi ini muncul seperti notifikasi alarm yang biasa ada pada perangkat Android. Notifikasi ini akan menghasilkan getaran serta suara.

3. Notifikasi pada perangkat Pendeteksi Kebocoran Gas

Notifikasi ini hanya berupa suara *beep*. Notifikasi ini hanya dapat diterima ketika pengguna dekat dengan perangkat Pendeteksi Kebocoran Gas.

4. Notifikasi berupa SMS

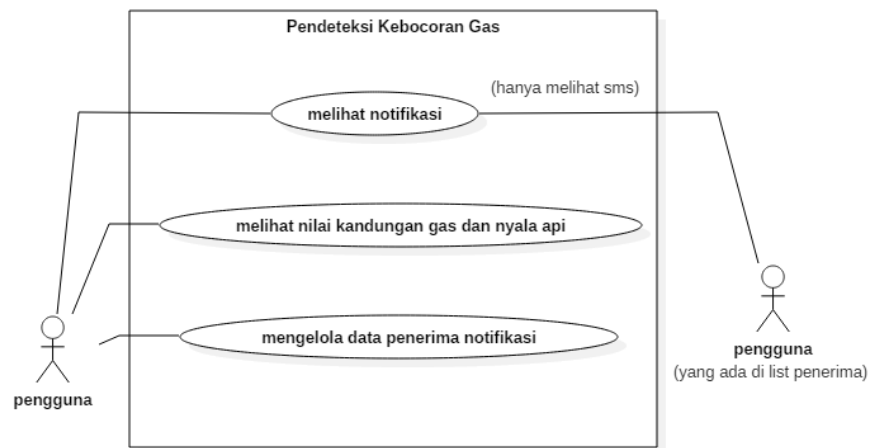
Notifikasi ini dikirimkan kepada penerima yang telah dimasukkan ke dalam list penerima yang ada pada perangkat Android.

3.4.2. Perancangan Perangkat Lunak

Perangkat lunak yang akan dibangun adalah perangkat lunak yang akan dipasang pada smartphone dengan sistem operasi Android. Aplikasi ini akan digunakan mengatur data penerima notifikasi dan juga sebagai penerima notifikasi ketika pada perangkat keras mendeteksi kandungan bahan yang melebihi nilai batas normal.

3.4.2.1. Use Case Diagram

Aplikasi pada perangkat Android memiliki 3 use case yaitu melihat notifikasi, melihat nilai kandungan gas dan mengelola data penerima notifikasi. Pengguna dapat mengatur penerima notifikasi melalui Android, kemudian data tersebut akan disimpan ke dalam database. Ketika perangkat keras mendeteksi kandungan gas yang berlebihan ataupun mendeteksi nyala api, maka penerima yang telah disimpan datanya di dalam database akan menerima notifikasi.

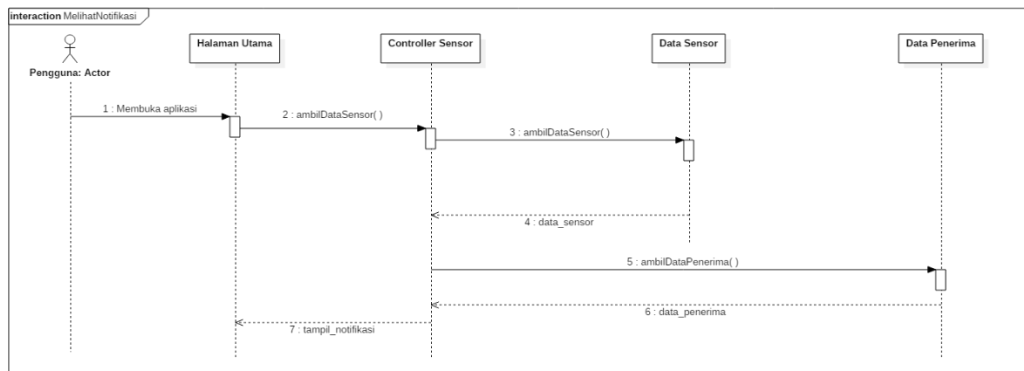


Gambar 3.3. Use Case Diagram

3.4.2.2. Sequence Diagram

Berdasarkan use case diagram, maka terdapat 3 sequence diagram pada Perangkat Pendeteksi Kebocoran Gas.

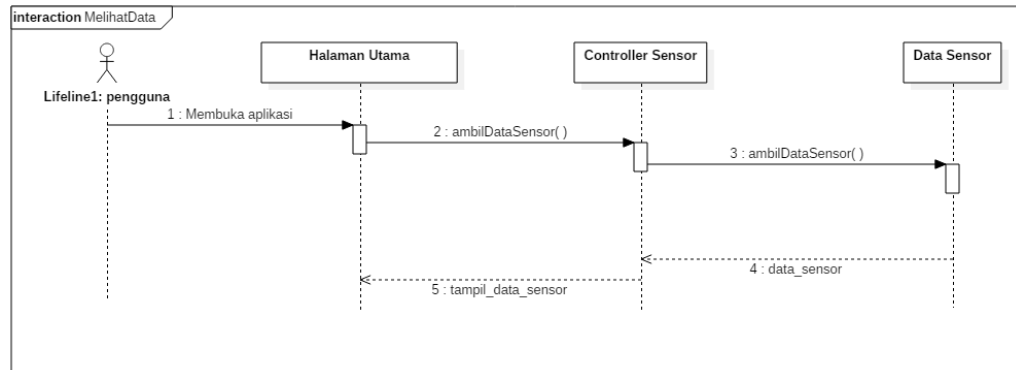
1) Sequence Diagram Melihat Notifikasi



Gambar 3.4. Sequence Diagram Melihat notifikasi

Sensor gas akan mengambil nilai kandungan gas di udara kemudian dikirimkan ke perangkat NodeMCU. Selanjutnya NodeMCU akan memproses data tersebut apakah kandungan gas berada di batas normal. Jika kandungan melebihi batas normal maka NodeMCU akan mengirimkan data notifikasi ke perangkat Android. Selanjutnya perangkat Android akan menampilkan notifikasi tersebut kepada pengguna.

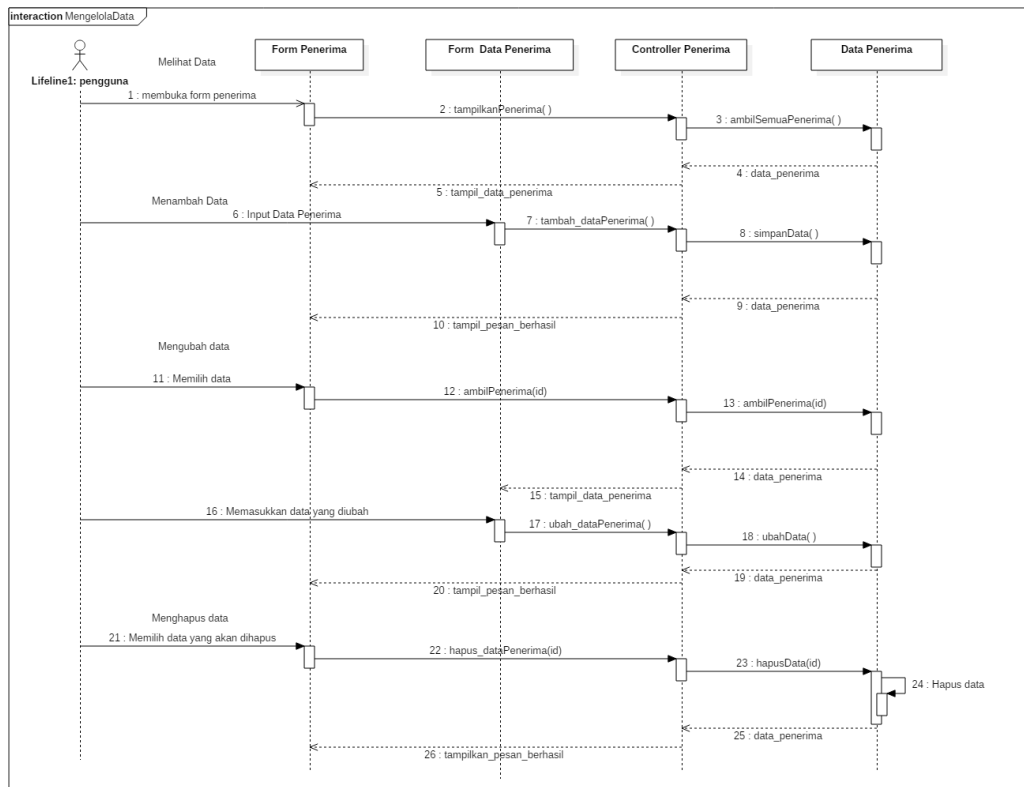
2) Sequence Diagram Melihat Nilai Kandungan Gas dan Nyala Api



Gambar 3.5. Sequence Diagram Melihat nilai kandungan gas dan nyala api

Pada gambar 3.5 perangkat NodeMCU akan menerima data dari sensor gas dan sensor api. Kemudian perangkat NodeMCU akan mengirimkan data tersebut ke perangkat Android. Aplikasi pada perangkat Android akan menampilkan nilai kandungan gas dan nyala api ke pengguna.

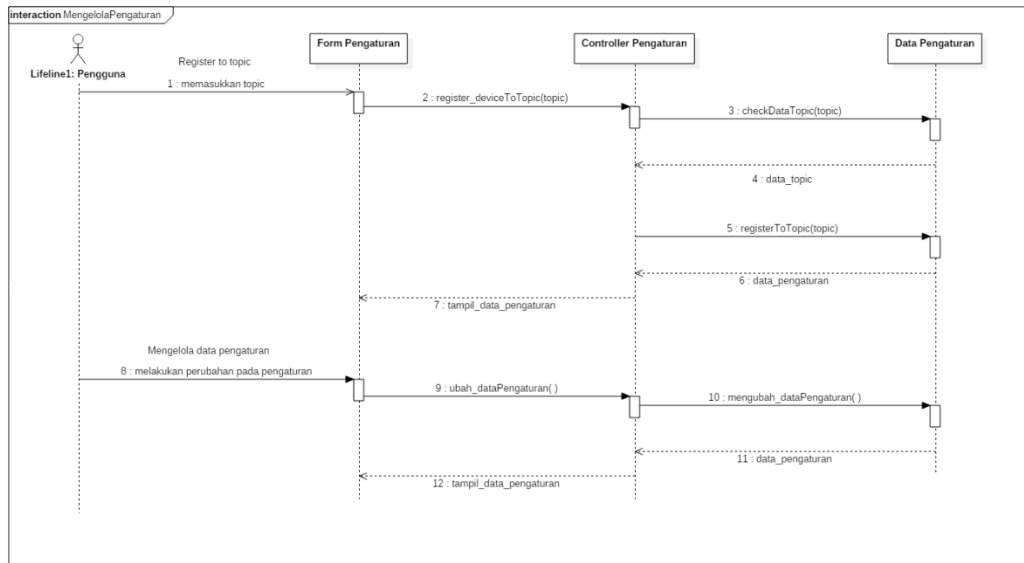
3) Sequence Diagram Mengelola Data Penerima Notifikasi



Gambar 3.6. Sequence Diagram Mengelola data penerima notifikasi

Gambar 3.6 menjelaskan tentang mengelola data penerima notifikasi pada perangkat pendeteksi kebocoran gas. Dari data ini perangkat pendeteksi kebocoran gas akan mengirimkan notifikasi ketika kandungan gas dan nyala api melebihi batas normal.

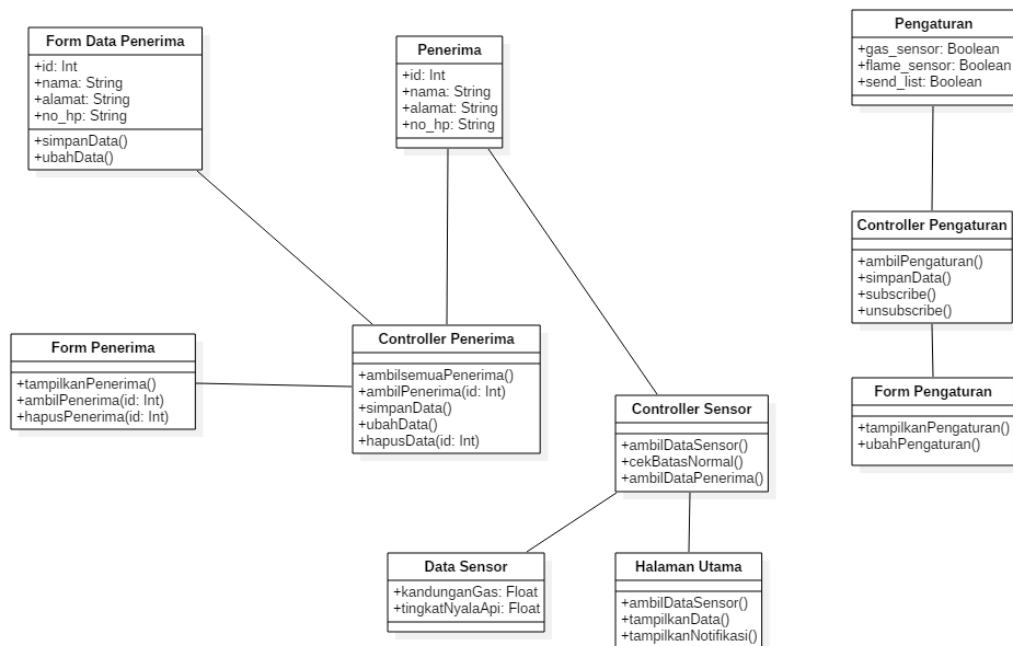
4) Sequence Diagram Mengelola Data Pengaturan



Gambar 3.7. Sequence Diagram Mengelola data pengaturan

Gambar 3.7 menjelaskan tentang mengelola data pengaturan yang akan digunakan oleh perangkat pendeteksi kebocoran gas. Pengguna akan melakukan pendaftaran *topic* yang akan digunakan melalui perangkat Android. Jika *topic* ada pada data pengaturan maka pada perangkat Android akan tampil pengaturan yang sudah diatur sebelumnya. Jika pengguna memasukkan *topic* yang salah maka akan muncul pesan pada perangkat Android.

3.4.2.3. Class Diagram



Gambar 3.8. Class Diagram

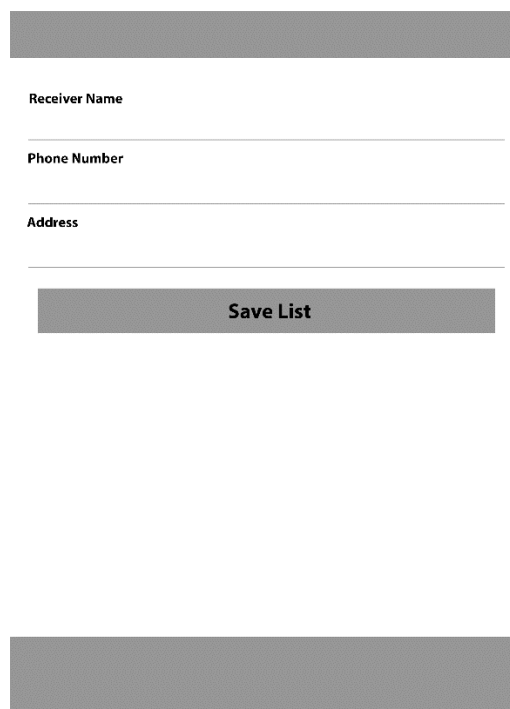
Diagram diatas merupakan class diagram untuk perangkat pendeteksi kebocoran gas. Pada Class Diagram terdapat 3 objek yaitu Penerima, Pengaturan dan Data Sensor. Pada aplikasi, Controller sensor akan mengambil data kemudian menampilkannya pada halaman utama. Pada pengambilan data Controller sensor juga melakukan pengecekan data, ketika data melebihi batas normal maka controller akan mengirimkan perintah untuk menampilkan notifikasi.

Data penerima dapat dikelola melalui halaman Form Penerima dan Form Data Penerima. Controller ListPenerima digunakan untuk memberikan perintah untuk melakukan perubahan data ke dalam data Penerima.

Data pengaturan dapat dikelola melalui halaman Form Pengaturan, ketika pertama kali menjalankan perangkat pendeteksi kebocoran gas, perangkat tersebut akan langsung menyimpan pengaturan data. Pada perangkat Android hanya dapat mengubah data yang telah ada sebelumnya.

3.4.2.4. Perancangan Antarmuka

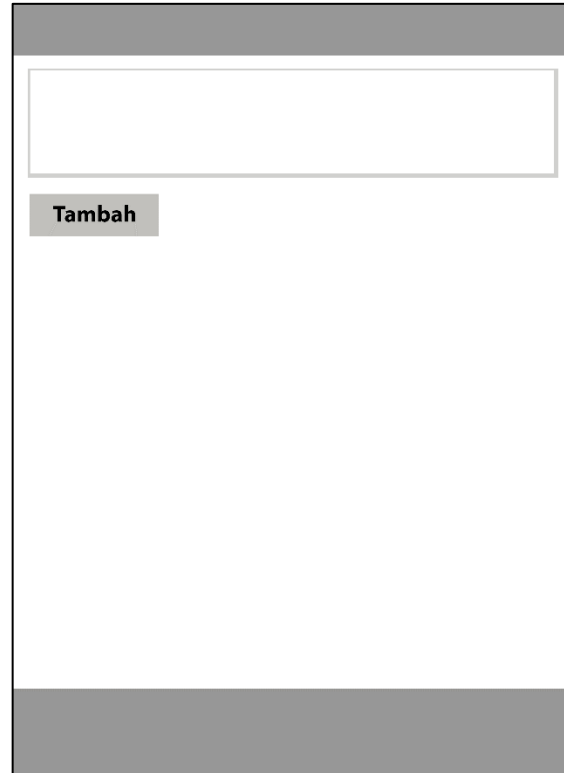
Pada Gambar 3.9 dan Gambar 3.10 terdapat perancangan antarmuka untuk mengelola list penerima notifikasi. List ini berfungsi untuk menyimpan data penerima notifikasi ketika sensor pada NodeMCU mendeteksi nilai kandungan gas yang melebihi batas. Data-data ini akan tersimpan ke dalam sebuah file JSON yang ada pada perangkat pendeteksi kebocoran gas yang dibangun.



The image shows a web form with a grey header bar at the top. Below the header, there are three input fields, each with a label and a horizontal line for text entry. The labels are "Receiver Name", "Phone Number", and "Address". At the bottom of the form, there is a grey button with the text "Save List".

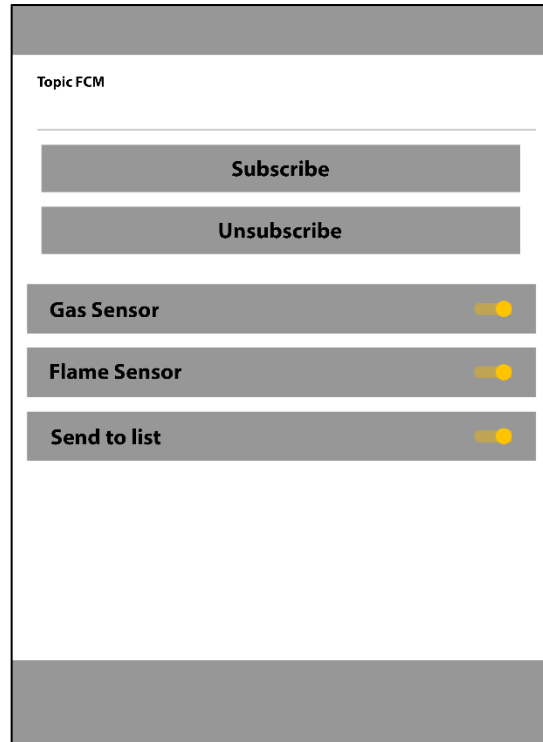
Gambar 3.9. Perancangan Antarmuka (Tambah List Penerima & Update List Penerima)

Gambar 3.10 merupakan halaman yang akan menampilkan semua penerima yang telah didaftarkan oleh pengguna.



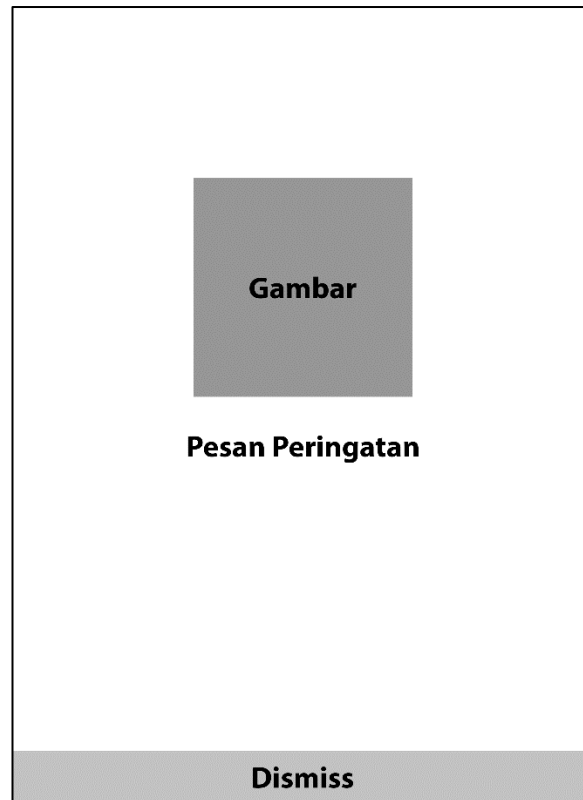
Gambar 3.10. Perancangan Antarmuka (List Penerima Notifikasi)

Pada Gambar 3.11 terdapat tampilan pengaturan aplikasi. Pengguna dapat mengatur perangkat pendeteksi kebocoran gas melalui aplikasi pada perangkat Android.



Gambar 3.11. Perancangan Antarmuka (Pengaturan)

Pada Gambar 3.12 terdapat tampilan yang akan muncul ketika perangkat mendeteksi adanya kebocoran gas ataupun terdapat api yang menyala.



Gambar 3.12. Perancangan Antarmuka (Alarm)

BAB IV HASIL DAN PEMBAHASAN

4.1. Hasil Implementasi

4.1.1 Implementasi Perangkat Keras



Gambar 4.1. Perangkat Pendeteksi Kebocoran Gas (Luar)



Gambar 4.2. Perangkat Pendeteksi Kebocoran Gas (Dalam)

Perangkat ini ketika pertama kali dijalankan akan menampilkan LED berwarna kuning. Hal ini menunjukkan perangkat harus di atur terlebih dahulu. Untuk mengaturnya dapat dengan menyambungkan perangkat Android pada *WiFi* yang telah dibuat pada perangkat tersebut (Default SSID: GasDetector).

Setelah tersambung pada perangkat kita harus mengakses halaman website dengan menggunakan IP 192.168.4.1 untuk mendaftarkan pengaturan. Pengaturan yang diperlukan adalah mendaftarkan perangkat pada WiFi yang menyediakan akses internet serta mendaftarkan *topic* yang akan digunakan.

Kode Program

```

void detect_gas(){
  gas_value = analogRead(gas_sensor);
  Serial.println("gas sensor value");
  Serial.println(gas_value, DEC);
  if (gas_value > 700){
    digitalWrite(led_warning, HIGH);
    digitalWrite(led_work, LOW);
    if (gasstate == 0){
      Serial.println("Gas Detected!!! Help Me!!!");
      doitGasOnly("Gas", "Gas has been detected",
topic_firebase);
      list = Firebase.getBool(ls + "/send_list");
      if(list == true){
        sendsms();
      }
      gasstate = 1;
    }
    else
    {}
  }
  else{
    digitalWrite(led_work, HIGH);
    digitalWrite(led_warning, LOW);
    gasstate = 0;
  }
}

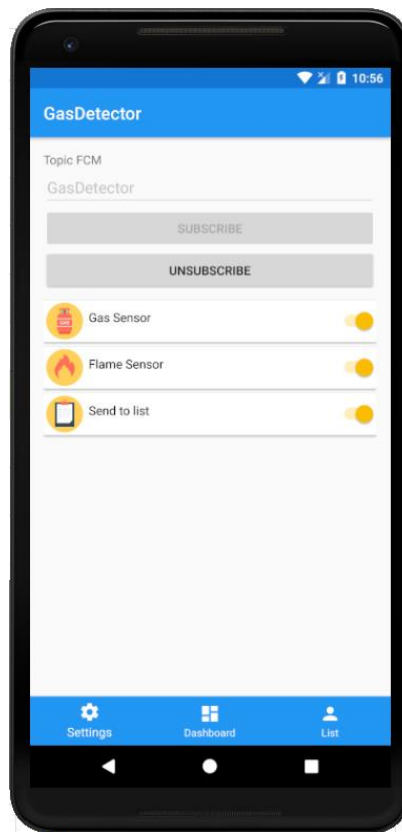
void detect_flame(){
  Flame = digitalRead(fire_sensor);
  if (Flame == HIGH) {
    Serial.println("No fire detected");
    digitalWrite(led_detect, LOW);
    digitalWrite(led_work, HIGH);
  }
  else{
    Serial.println("Fire detected");
    digitalWrite(led_detect, HIGH);
    digitalWrite(led_work, LOW);
    doitTOPIC("Title", "Fire has has been detected",
topic_firebase);
    if(gas_value > 700){
      tone(buzzer, 1000, 200);
    }
  }
}

```

4.1.2 Implementasi Perangkat Lunak

4.1.2.1. Halaman Pengaturan

Pada halaman ini pengguna terlebih dahulu harus mendaftarkan *topic* yang sama yang telah didaftarkan pada perangkat Pendeteksi Kebocoran Gas sebelumnya. Pengguna hanya dapat mengubah pengaturan seperti menyalakan sensor gas atau menonaktifkan sensor tersebut.



Gambar 4.3. Halaman Pengaturan

Kode Program

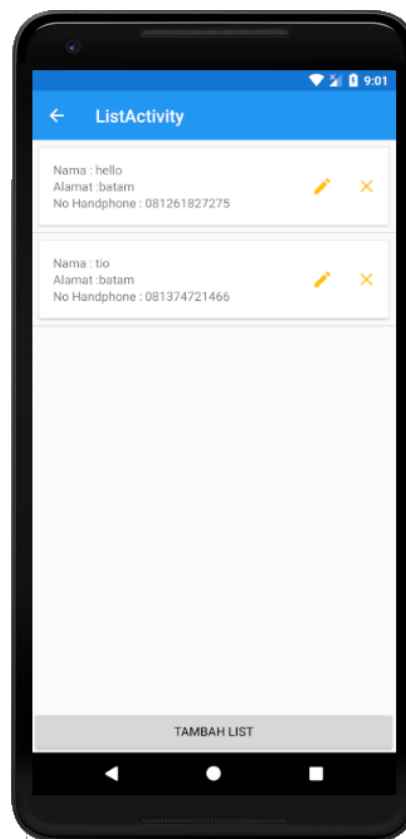
```

public void unsubscribe()
{
    FirebaseMessaging.getInstance().unsubscribeFromTopic(topic_app);
    Toast.makeText(SettingActivity.this, "Unsubscribed from
    "+topic_app+" topic",Toast.LENGTH_LONG).show();
    checkdatauser.child("Topic").removeValue();
    checkdatauser.setValue(0);
    getData();
}
public void Subscribe(String topic)
{
    final String topictocheck = topic;
    DatabaseReference ref =
    FirebaseDatabase.getInstance().getReference().child("topic").child(topic);
    ref.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot)
        {
            if(dataSnapshot.exists())
            {
                FirebaseMessaging.getInstance().subscribeToTopic(topictocheck);
                Toast.makeText(SettingActivity.this, "Subscribed
                to "+topictocheck+" topic",Toast.LENGTH_LONG).show();
                String data_topic =
                editTextTopic.getText().toString();
                checkdatauser.child("Topic").setValue(data_topic);
                getData();
            }
            else
            {
                Toast.makeText(SettingActivity.this, "Cannot
                subscribed to "+topictocheck+" because topic didn't exist",
                Toast.LENGTH_LONG).show();
                getData();
            }
        }
        @Override
        public void onCancelled(@NonNull DatabaseError
        databaseError) {
        }
    });
}

```

4.1.2.2. Halaman List Penerima

Pada halaman list penerima kita dapat melihat daftar penerima yang akan menerima notifikasi saat perangkat keras mendeteksi gas.



Gambar 4.4. Halaman List Penerima

Kode Program

```

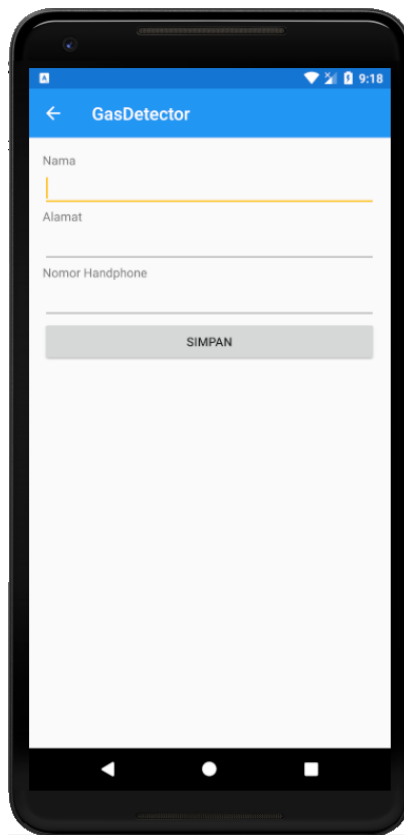
private void loading()
{
    class GetJSON extends AsyncTask<Void, Void, String >{
        ProgressDialog loading;
        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            swipeRefreshLayout.setRefreshing(true);
            loading =
ProgressDialog.show(ListActivity.this, "Mengambil Data", "Mohon
Tunggu...", false, false);
        }

        @Override
        protected void onPostExecute(String s) {
            super.onPostExecute(s);
            if(s.equals("done"))
            {
                swipeRefreshLayout.setRefreshing(false);
                loading.dismiss();
            }
            JSON_STRING = s;
        }
        @Override
        protected String doInBackground(Void... params) {
            showPenerima();
            return "done";
        }
    }
    GetJSON gj = new GetJSON();
    gj.execute();
}

```

4.1.2.3. Halaman Tambah List Penerima

Pada halaman tambah list penerima user akan memasukkan nama, alamat dan nomor handphone dengan benar. Selanjutnya jika data yang dimasukkan benar maka data akan tersimpan dan sistem akan menampilkan halaman list penerima



Gambar 4.5. Gambar Halaman Tambah List Penerima

Kode Program

```

private void setData()
{
    data =
    FirebaseDatabase.getInstance().getReference().child("penerima").
    child(topic);
    data.addListenerForSingleValueEvent(new ValueEventListener()
    {
        @Override
        public void onDataChange(@NonNull DataSnapshot
dataSnapshot)
        {
            String alamat = mAlamat.getText().toString();
            String nama = mName.getText().toString();
            String nohp = mNohp.getText().toString();

            data.child(String.valueOf(arraysize)).child("alamat").setValue(a
alamat);

            data.child(String.valueOf(arraysize)).child("nama").setValue(nam
a);

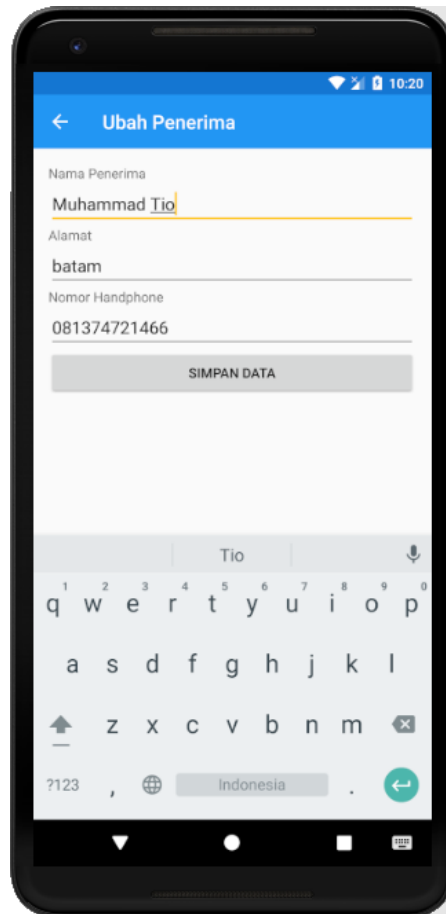
            data.child(String.valueOf(arraysize)).child("nohp").setValue(noh
p);

                onBackPressed();
            }
            @Override
            public void onCancelled(@NonNull DatabaseError
databaseError) {}
        });
    }
}

```

4.1.2.4. Halaman Update List Penerima

Pada halaman ini pengguna dapat mengubah data penerima. Jika pengguna mengisi data dengan benar dan menekan tombol simpan maka sistem akan mengubah data sesuai dengan apa yang dimasukkan oleh pengguna.



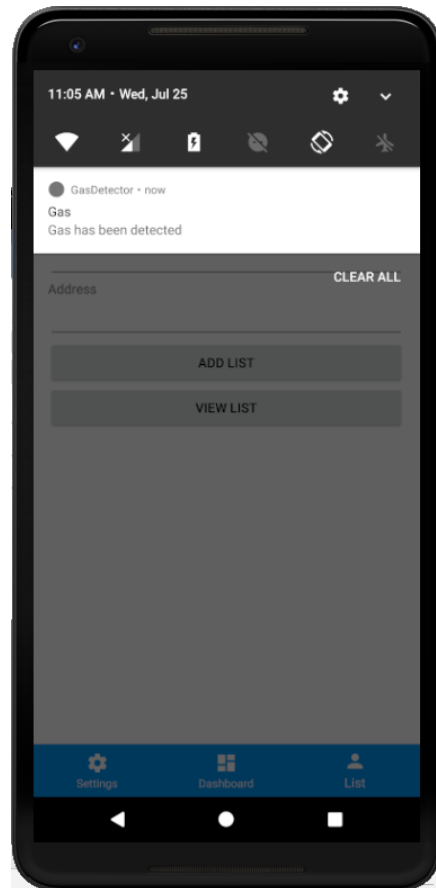
Gambar 4.6. Gambar Halaman Update List Penerima

Kode Program

```
private void setData()
{
    data =
    FirebaseDatabase.getInstance().getReference().child("penerima").
    child(topic);
    data.addListenerForSingleValueEvent(new ValueEventListener()
    {
        @Override
        public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
            if(dataSnapshot.exists())
            {
                String alamat = mAlamat.getText().toString();
                String nama = mName.getText().toString();
                String nohp = mNohp.getText().toString();
                data.child(id).child("alamat").setValue(alamat);
                data.child(id).child("nama").setValue(nama);
                data.child(id).child("nohp").setValue(nohp);
                onBackPressed();
            }
            showList();
        }
        @Override
        public void onCancelled(@NonNull DatabaseError
databaseError) {
        }
    });
}
```

4.1.2.5. Notifikasi

Terdapat 3 (Tiga) notifikasi yang akan muncul pada perangkat Android. Notifikasi yang pertama ditampilkan pada Gambar 4.7 yaitu menampilkan notifikasi hanya pada bar pada Android.



Gambar 4.7. Notifikasi pada bar Android

Kode Program

```

    private void showNotifications(String title, String msg) {
        Intent i = new Intent(this, MainActivity.class);

        PendingIntent pendingIntent =
PendingIntent.getActivity(this, REQUEST_CODE,
            i, PendingIntent.FLAG_UPDATE_CURRENT);

        Notification notification = new
NotificationCompat.Builder(this)
            .setContentText(msg)
            .setContentTitle(title)
            .setContentIntent(pendingIntent)
            .setSmallIcon(R.mipmap.ic_launcher_round)
            .build();

        NotificationManager manager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
        manager.notify(NOTIFICATION_ID, notification);
    }

    private void sendMyNotification(String message) {

        //On click of notification it redirect to this Activity
        Intent intent = new Intent(this, MainActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        PendingIntent pendingIntent =
PendingIntent.getActivity(this, 0, intent,
PendingIntent.FLAG_ONE_SHOT);

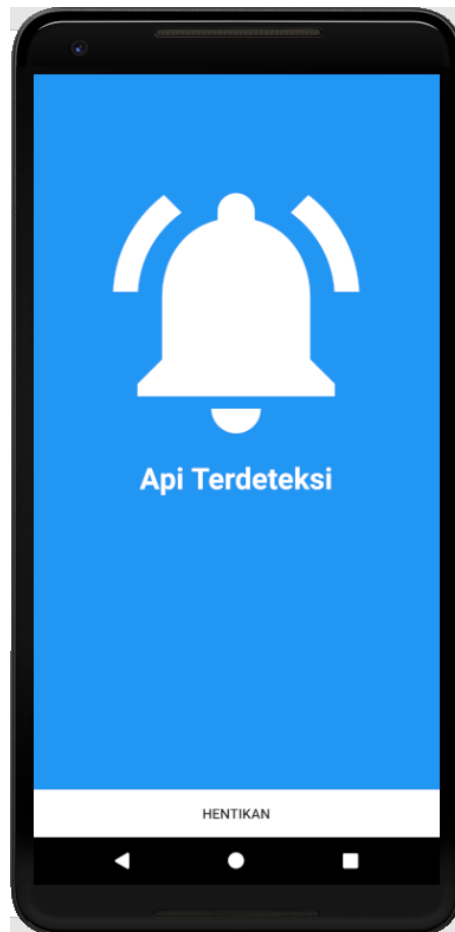
        NotificationCompat.Builder notificationBuilder = new
NotificationCompat.Builder(this)
            .setSmallIcon(R.mipmap.app_icon)
            .setContentTitle("Gas Detector")
            .setContentText(message)
            .setAutoCancel(true)
            .setContentIntent(pendingIntent);

        NotificationManager notificationManager =
(NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

        notificationManager.notify(0,
notificationBuilder.build());
    }
}

```

Notifikasi yang kedua ditampilkan seperti pada Gambar 4.8 yaitu berupa notifikasi alarm. Pada notifikasi ini perangkat Android akan menghasilkan getaran serta menghasilkan suara layaknya alarm.



Gambar 4.8. Notifikasi berupa Alarm pada Android

Kode Program

```

public class AlarmActivity extends AppCompatActivity {

    private Button mDismiss;

    MediaPlayer mMediaPlayer;
    Vibrator mVibrator;
    long[] pattern = { 0, 200, 0 };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_alarm);
        mDismiss = (Button) findViewById(R.id.dismiss);

        mDismiss.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(AlarmActivity.this,
MainActivity.class);
                startActivity(intent);
                finish();
                return;
            }
        });
        mVibrator = (Vibrator)
getSystemService(Context.VIBRATOR_SERVICE);
    }

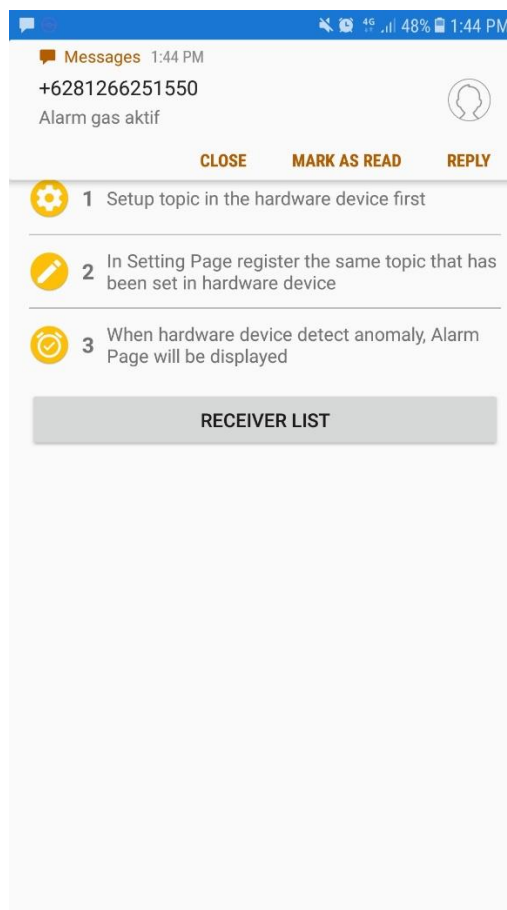
    @Override
    protected void onStart() {
        super.onStart();
        mVibrator.vibrate(pattern, 0);
        mMediaPlayer = new MediaPlayer();
        mMediaPlayer = MediaPlayer.create(AlarmActivity.this,
R.raw.alarm);

        mMediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
        mMediaPlayer.setLooping(true);
        mMediaPlayer.start();
    }

    @Override
    protected void onStop() {
        super.onStop();
        mMediaPlayer.stop();
        mVibrator.cancel();
    }
}

```

Notifikasi yang ke-3 (ketiga) adalah notifikasi berupa SMS yang dikirimkan ke penerima yang telah didaftarkan pada list di perangkat Android. Notifikasi ini dikirimkan melalui *sms gateway* yang dibangun sendiri. Ketika perangkat mendeteksi gas, perangkat akan mengakses web serta mengirimkan data dengan menggunakan metode POST. Selanjutnya web tersebut akan menjalankan file python yang berisi perintah untuk mengirimkan SMS melalui Gammu.



Gambar 4.9. Notifikasi SMS

Kode Program (Perangkat Pendeteksi Kebocoran Gas)

```

void sendsms(){
    sizedata = Firebase.getInt(sd);
    Serial.print("Jumlah data : ");
    Serial.println(sizedata);
    if(sizedata > 0){
        for(int i = 1; i <= sizedata; i++){
            Serial.println(Firebase.getString(pn + "/" + i + "/nohp"
));
            dosms(Firebase.getString(pn + "/" + i + "/nohp"));
            delay(3000);
        }
    }
}

```

Kode Program (Web dengan Bahasa PHP)

```

<?php
if(!empty($_POST["pass"])    &&    !empty($_POST["text"])    &&
!empty($_POST["nohp"])){
    $pass = $_POST["pass"];
    $text = $_POST["text"];
    $nohp = $_POST["nohp"];
    exec('sudo python /home/pi/sms.py "'.$pass.'" "'.$text.'"
'.$nohp, $output);
    print_r($output);
}

```

Kode Program (Python untuk menjalankan perintah Gammu)

```

#!/usr/bin/env python
import os
import sys
if(sys.argv[1] == "trips0907"):
    os.system("echo '" + sys.argv[2] + "' | sudo gammu sendsms
TEXT " + sys.argv[3]);
else:
    print("Password salah");

```

4.2. Hasil Pengujian

4.2.1. Deskripsi Pengujian

Pengujian dilakukan untuk menguji setiap fungsional dan proses dengan kemungkinan kesalahan yang terjadi untuk setiap fungsional.

Deskripsi pengujian dalam sistem ini adalah sebagai berikut:

1. Melakukan proses pendaftaran pada perangkat keras (NodeMCU)
 - a. Menyambungkan perangkat keras pada jaringan internet (WiFi).
 - b. Mendaftarkan *topic* yang akan digunakan.
 - c. Deteksi Gas
 - d. Deteksi Api
2. Melakukan proses pendaftaran pada perangkat lunak (Android)
 - a. Mendaftarkan perangkat ke *topic* yang sama sesuai dengan perangkat keras.
 - b. Menambahkan data penerima
 - c. Melihat notifikasi

4.2.2. Hasil Pengujian

Tabel 4.1. Uji Pendaftaran pada Perangkat Keras

Nama Proses	Skenario Pengujian	Data Uji	Target	Validasi
Daftar ke jaringan WiFi dan mendaftarkan <i>Topic</i>	User memasukkan SSID dan Password yang benar sesuai dengan jaringan WiFi	SSID : <i>(sesuai dengan wifi)</i> Password : <i>(sesuai dengan wifi)</i> <i>Topic : (sesuai dengan keinginan pengguna)</i>	Perangkat keras bekerja dengan menampilkan tanda berupa lampu LED berwarna hijau	Sukses
	User memasukkan SSID dan Password yang salah	SSID : <i>(SSID yang salah ataupun tidak ada dalam jangkauan)</i> Password : <i>(password yang tidak sesuai dengan SSID yang ada)</i>	Perangkat akan melakukan <i>reset</i> dan user harus memasukkan kembali data yang benar agar perangkat bekerja	Sukses

Tabel 4.2. Uji Sensor

Nama Proses	Skenario Pengujian	Data Uji	Target	Validasi
Deteksi Gas	Sensor gas dalam keadaan normal	Data sensor < 700	Lampu LED pada perangkat berwarna hijau, tidak muncul notifikasi	Sukses
	Lingkungan sekitar mengandung gas LPG	Data sensor > 700	Lampu LED berwarna merah, muncul notifikasi pada perangkat android dan pada perangkat menghasilkan bunyi.	Sukses
Deteksi api	Tidak ada api yang menyala	-	Lampu LED pada perangkat berwarna hijau dan tidak muncul notifikasi	Sukses
	Api menyala	-	Lampu LED pada perangkat berwarna merah dan muncul notifikasi berupa alarm pada perangkat Android	Sukses

Tabel 4.3. Uji Pendaftaran pada Perangkat Lunak

Nama Proses	Skenario Pengujian	Data Uji	Target	Validasi
Proses <i>Subscribe</i> ke <i>topic</i>	<i>Topic</i> yang digunakan sesuai dengan perangkat keras.	<i>Topic: (sesuai dengan perangkat keras)</i>	Sistem berhasil mendaftarkan perangkat lunak agar bisa berkomunikasi dengan perangkat keras.	Sukses
	<i>Topic</i> yang digunakan tidak ada pada perangkat manapun	<i>Topic: (topic yang dimasukkan tidak terdaftar pada perangkat manapun)</i>	Muncul pesan kesalahan, karena <i>topic</i> tidak ada	Sukses
Daftar penerima	User mengisi data nama, alamat dan nomor handphone dengan benar	nama: Muhammad Tio alamat: Botania Garden 3 nohp: 081374721466	Sistem berhasil menyimpan data dan menampilkan halaman list penerima	Sukses
	User tidak mengisi semua data	nama: - alamat: Batam nohp: -	Muncul pesan kesalahan, user tetap pada halaman tambah list penerima	Sukses

Nama Proses	Skenario Pengujian	Data Uji	Target	Validasi
Melihat notifikasi	User berhasil <i>Subscribe</i> ke <i>topic</i> dan pada perangkat keras sensor mendeteksi adanya gas ataupun api	-	Muncul notifikasi sesuai dengan apa yang dideteksi oleh perangkat keras	Sukses
	User belum melakukan <i>Subscribe</i> ke <i>topic</i> dan pada perangkat keras sensor mendeteksi adanya gas ataupun api	-	Tidak muncul notifikasi apapun	Sukses

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil implementasi pada Sistem Pendeteksi Kebocoran Gas Berbasis IoT didapatkan hasil berupa

1. Perangkat Pendeteksi Kebocoran Gas yang dibangun menggunakan NodeMCU dan perangkat Android dapat memberikan informasi akan keadaan berbahaya seperti kebocoran gas dapat secara cepat diketahui oleh pemilik rumah dengan memanfaatkan perangkat yang dibangun.
2. Firebase digunakan untuk mendukung komunikasi antara perangkat IoT dengan perangkat Android menjadi lebih mudah.
3. Menampilkan notifikasi pada perangkat Android adalah pilihan yang tepat agar pengguna ataupun pemilik rumah dapat langsung mengetahui jika ada kebocoran gas dibandingkan dengan perangkat yang telah dibangun sebelumnya.

5.2. Saran

Untuk pengembangan lebih lanjut maka penulis memberikan saran agar Sistem Pendeteksi Kebocoran Gas Berbasis IoT menjadi lebih baik.

- Untuk pengembangan agar sistem menjadi lebih baik, perlu ditambahkan perangkat tambahan agar ketika terdeteksi gas ataupun api, perangkat tersebut dapat langsung mengatasi ataupun mengurangi kandungan gas tersebut.

DAFTAR PUSTAKA

- Bagus A.S., 2016, Rancang Bangun Data Logging Berbasis Web Server Pada Robot Balon Udara Untuk Deteksi Kebocoran Pipa Gas, Institut Teknologi Sepuluh November.
- Muddassir, 2017, Alat Monitoring Gas/Asap Pada Ruangan Berbasis Relay Raspberry, Universitas Negeri Makasar.
- Nazruddin S.H., 2014, Pemrograman Aplikasi Smartphone dan Tablet PC berbasis Android, Penerbit Informatika, Bandung.
- Shruti G.S., 2016, Implementation of Smart Home Automation System through E-mail using Raspberry Pi and Sensors, International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering Vol 4, Issue 3, March 2016.
- Yayan C., 2016, Prototype Early Warning Kebakaran Dengan Microcontroller Menggunakan Algoritma Fuzzy Logic dan Sistem Monitoring Berbasis Website, Universitas Komputer Indonesia.

BIODATA PENULIS



Muhammad Tio, lahir di Batam pada tanggal 9 Juli 1997. Anak pertama dari tiga bersaudara dari Koswara dan Sumiati. Penulis menyelesaikan Pendidikan Sekolah Dasar di SD Negeri 003 Bintan pada tahun 2009. Pada tahun itu juga penulis melanjutkan Pendidikan di SMP Negeri 009 Bintan dan tamat pada tahun 2012. Kemudian, penulis melanjutkan Sekolah Menengah Atas di SMA Negeri 05 Bintan dan tamat pada tahun 2015. Setelah lulus Sekolah Menengah Atas, penulis melanjutkan pendidikan ke perguruan tinggi negeri di Politeknik Negeri Batam jurusan D3 Teknik Informatika dan sekarang berada di semester 6. Selama kuliah di Politeknik Negeri Batam penulis pernah aktif mengikuti organisasi seperti Batam Linux User Group. Selain itu penulis pernah mengikuti beberapa kegiatan seperti Seminar Nasional Repackaging OS sebagai Ketua Pelaksana dan juga mengikuti kegiatan BYOC. Penulis memiliki sertifikasi keahlian diantaranya Sertifikasi MTA (*Microsoft Technology Associate*) Software Development Fundamental.