

SISTEM MONITORING DATA SENSOR DAN KONTROL GEDUNG DARI BAHAYA KEBAKARAN BERBASIS IOT

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun oleh:

Eko Firlianto NIM: 0031807

Dini Anjelina NIM: 0031806

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2021**

LEMBAR PENGESAHAN

JUDUL PROYEK AKHIR
SISTEM MONITORING DATA SENSOR DAN KONTROL GEDUNG DARI
BAHAYA KEBAKARAN BERBASIS IOT

Oleh:

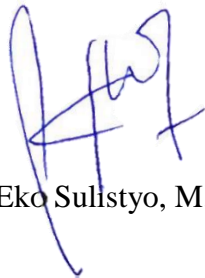
Eko Firlianto/0031807

Dini Anjelina/0031806

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1



Eko Sulistyono, M.T.

Pembimbing 2



Aan Febriansyah, M.T.

Penguji 1



Irwan, M.Sc., Ph.D.

Penguji 2



Muhammad Iqbal Nugraha, S.ST, M.Eng.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1: Eko Firlianto

NIM: 0031807

Nama Mahasiswa 2: Dini Anjelina

NIM: 0031806

Dengan Judul: SISTEM MONITORING DATA SENSOR DAN KONTROL
GEDUNG DARI BAHAYA KEBAKARAN BERBASIS IOT

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata di kemudian hari melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 30 Mei 2021

Nama Mahasiswa

Tanda Tangan

1. Eko Firlianto



2. Dini Anjelina



ABSTRAK

Kebakaran merupakan sebuah bencana yang ditandai dengan api menyala secara tidak terkendali dan dapat disebabkan oleh berbagai faktor diantaranya kebocoran tabung gas, hubungan pendek arus listrik, putung rokok, petasan, dan lain-lain. Dampak dari peristiwa kebakaran ini dapat berupa kerusakan bangunan, terganggunya kegiatan usaha, rusaknya fasilitas umum bahkan sampai menimbulkan korban jiwa. Umumnya bencana kebakaran baru dapat terdeteksi ketika api sudah terlanjur membesar dan keluar asap yang banyak dari gedung atau bangunan lainnya. Oleh karena itu, dibutuhkan sebuah alat pendeteksi kebakaran yang dapat memantau kondisi gedung melalui smartphone dan diikuti tindakan penanganan kebakaran. Monitoring bertujuan untuk menampilkan data dari sensor api, sensor asap dan sensor suhu, sedangkan tindakan kontrol berupa aktivasi alarm, exhaust, membuka pintu dan jendela serta LED sebagai indikator aktivasi air pemadaman yang bertujuan untuk meminimalkan dampak buruk dari peristiwa kebakaran tersebut. Metode pelaksanaan yang digunakan yaitu perancangan dan pembuatan konstruksi miniatur rumah, perancangan dan pembuatan sistem kontrol menggunakan Arduino Mega 2560, perancangan dan pembuatan monitoring dan kontrol IoT serta pengujian alat. Dari hasil pengujian menggunakan sensor api, sensor asap (rentang kadar asap 0-241 ppm) dan sensor suhu (rentang suhu 29°C-53°C) diketahui bahwa ketika terdeteksi api kemudian kadar asap >110 ppm dan suhu ruangan >38°C maka sistem kontrol akan secara otomatis mengaktifkan alarm, exhaust, LED, dan motor servo serta mengirimkan notifikasi kebakaran ke smartphone sehingga peristiwa kebakaran dapat dideteksi dan ditangani secara dini.

Kata kunci: kebakaran, sensor api, sensor asap, sensor suhu

ABSTRACT

Fire is a disaster characterized by fire burning uncontrollably and can be caused by various factors including gas cylinder leakage, electrical short circuit, cigarette butts, firecrackers, and others. The impact of this fire event can be in the form of damage to buildings, disruption of business activities, damage to public facilities and even fatalities. Generally, a fire disaster can only be detected when the fire has already grown and a lot of smoke comes out of the building or other buildings. Therefore, a fire detection device is needed that can monitor the condition of the building via a smartphone and follow the fire handling actions. Monitoring aims to display data from fire sensors, smoke sensors and temperature sensors, while control measures in the form of alarm activation, exhaust, opening doors and windows and LEDs as indicators of extinguishing water activation aim to minimize the adverse effects of the fire event. The implementation method used is the design and manufacture of miniature house construction, design and manufacture of control systems using Arduino Mega 2560, design and manufacture of IoT monitoring and control as well as tool testing. From the test results using a fire sensor, smoke sensor (smoke content range 0-241 ppm) and temperature sensor (temperature range 29°C-53°C) it is known that when a fire is detected, the smoke content is >110 ppm and the room temperature is >38°C. then the control system will automatically activate the alarm, exhaust, LED, and servo motor and send fire notifications to smartphones so that fire events can be detected and handled early.

Keywords: fire, fire sensor, smoke sensor, temperature sensor

KATA PENGANTAR

Bismillahirrahmanirrahim

Puji dan syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa berkat rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan makalah proyek akhir ini dengan baik dan sesuai dengan waktu yang telah ditentukan.

Makalah ini dibuat untuk memenuhi salah satu syarat wajib kelulusan Diploma III di Politeknik Manufaktur Negeri Bangka Belitung. Dengan adanya makalah ini diharapkan para pembaca dapat mengetahui gambaran proyek akhir yang dibuat oleh penulis. Makalah proyek akhir ini dibuat dengan melakukan pengembangan dari jurnal-jurnal penelitian yang sudah pernah dilakukan sebelumnya. Penulis mencoba menerapkan ilmu pengetahuan yang didapat selama 3 tahun menempuh pendidikan di Politeknik Manufaktur Negeri Bangka Belitung dalam pembuatan proyek akhir ini. Selain itu, penulis juga mendapatkan informasi berupa data-data pendukung dari makalah-makalah proyek akhir mahasiswa Politeknik Manufaktur Negeri Bangka Belitung tahun-tahun sebelumnya.

Selama menyusun makalah proyek akhir ini penulis mendapatkan banyak bantuan dan bimbingan dari berbagai pihak sehingga proses penulisan makalah ini dapat diselesaikan dengan baik. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Orangtua dan keluarga tercinta yang selalu memberikan dukungan baik secara moral maupun materil sehingga penulis dapat memberikan hasil yang terbaik.
2. Bapak I Made Andik Setiawan, M.Eng., Ph.D. selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Eko Sulistyono, M.T. selaku Wakil Direktur III Politeknik Manufaktur Negeri Bangka Belitung sekaligus dosen pembimbing 1 dalam proyek akhir ini.
4. Bapak Aan Febriansyah, M.T. selaku dosen pembimbing 2 dalam proyek akhir ini.

5. Bapak Muhammad Iqbal Nugraha, M.Eng. selaku Ketua Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
6. Bapak Ocsirendi, M.T. selaku Ketua Program Studi DIII Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung.
7. Seluruh tenaga pendidik dan kependidikan Politeknik Manufaktur Negeri Bangka Belitung.
8. Rekan-rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung yang telah membantu dalam penyelesaian proyek akhir ini.
9. Seluruh pihak yang ikut terlibat baik secara langsung maupun tidak langsung dalam penyelesaian proyek akhir ini yang tidak bisa disebutkan satu persatu.

Penulis menyadari bahwa dalam penulisan laporan ini masih terdapat banyak kekurangan karena terbatasnya pengetahuan dan kemampuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari semua pihak demi perbaikan dan pengembangan penulisan makalah ini di masa yang akan datang. Penulis berharap makalah ini dapat bermanfaat bagi pihak yang berkepentingan pada khususnya dan bagi perkembangan ilmu pengetahuan dan teknologi pada umumnya.

Sungailiat, 30 Mei 2021

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK.....	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN.....	xviii
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan	3
BAB II.....	5
LANDASAN TEORI.....	5
2.1 Pengertian Kebakaran	5
2.1.1 Ambang Batas Kebakaran.....	5
2.2 Studi Literatur Terkait.....	6
2.3 Sensor.....	9
2.3.1 <i>Flame Sensor</i>	10
2.3.2 Sensor MQ-9.....	12
2.3.3 Sensor DHT-11	13
2.4 Arduino Mega 2560	15
2.5 Servo Motor SG-90.....	17
2.5.1 Prinsip Kerja Servo Motor SG-90.....	17
2.6 DF Mini Player	18

2.7	LED (<i>Light Emitting Diode</i>)	20
2.7.1	Prinsip Kerja LED	20
2.8	<i>Internet of Thing</i> (IoT)	21
2.9	NodeMCU ESP8266	22
2.10	ARDUINO IDE	23
2.11	<i>Firebase</i>	24
2.12	MIT App Inventor	24
BAB III		26
METODE PELAKSANAAN		26
3.1	Pengumpulan dan Pengolahan Data	29
3.2	Rancangan <i>Hardware</i> dan <i>Software</i>	29
3.2.1	Rancangan <i>Hardware</i>	29
3.2.2	Rancangan <i>Software</i>	30
3.3	Pembuatan <i>Hardware</i> dan <i>Software</i>	31
3.3.1	Pembuatan <i>Hardware</i>	31
3.3.2	Pembuatan <i>Software</i>	32
3.4	Pengujian <i>Hardware</i> dan <i>Software</i>	32
3.4.1	Pengujian <i>Hardware</i>	33
3.4.2	Pengujian <i>Software</i>	33
3.4.3	Pengujian Keseluruhan Sistem	33
3.5	Analisis Data	34
3.6	Pembuatan Makalah Proyek Akhir	34
BAB IV		35
PEMBAHASAN		35
4.1	Deskripsi Alat	35
4.2	Perancangan dan Pembuatan <i>Hardware</i> Sistem <i>Monitoring</i> Data Sensor dan Kontrol Gedung	36
4.2.1	Perancangan <i>Hardware</i> secara Mekanik	36
4.2.2	Pembuatan <i>Hardware</i> secara Mekanik	38
4.2.3	Perancangan dan Pembuatan <i>Hardware</i> Elektrik	40

4.3 Pengujian Hardware Elektrik Sistem Monitoring Data Sensor dan Kontrol Gedung	42
4.3.1 Pengujian <i>Flame Sensor</i>	42
4.3.2 Pengujian Sensor MQ-9	48
4.3.3 Pengujian Sensor DHT-11	51
4.4 Perancangan dan Pembuatan <i>Software</i>	56
4.4.1 Perancangan dan Pembuatan <i>Interface</i> pada <i>Smartphone</i>	57
4.4.2 Perancangan dan Pembuatan <i>Database</i>	63
4.4.3 Komunikasi Serial pada <i>Software</i> MIT App Inventor	65
4.4.4 Koneksi MIT App Inventor dengan <i>Database</i>	65
4.5 Pengujian Alat.....	67
4.5.1 Pengujian <i>Prototype</i> Pendeteksi Kebakaran	70
4.5.2 Pengujian <i>Prototype</i> Pendeteksi Kebakaran Berbasis IoT.....	95
BAB V	113
PENUTUP.....	113
5.1 Kesimpulan	113
5.2 Saran	113
DAFTAR PUSTAKA	115

DAFTAR TABEL

Tabel 2. 1 Tabel Pengujian Sensor DHT-11	10
Tabel 2. 2 Spesifikasi Flame Sensor	11
Tabel 2. 3 Spesifikasi dari Sensor MQ-9	13
Tabel 2. 4 Spesifikasi dari Sensor DHT-11	14
Tabel 2. 5 Tabel Spesifikasi Arduino Mega 2560	16
Tabel 2. 6 Konfigurasi Warna Kabel Motor Servo	17
Tabel 2. 7 Karakteristik Servo Motor SG-90	18
Tabel 2. 8 Penjelasan Pin DF Mini Player	19
Tabel 2. 9 Tabel Spesifikasi NodeMCU ESP8266	22
Tabel 4. 1 Hasil Pengujian Flame Sensor terhadap Api	47
Tabel 4. 2 Hasil Pengujian Sensor MQ-9 terhadap Asap	51
Tabel 4. 3 Hasil Pengujian Sensor DHT-11 terhadap Suhu Panas	55
Tabel 4. 4 Hasil Pengujian Terdeteksi Api Pada Ruang 1	73
Tabel 4. 5 Hasil Pengujian Terdeteksi Api Pada Ruang 2	73
Tabel 4. 6 Hasil Pengujian Terdeteksi Asap pada Ruang 1	75
Tabel 4. 7 Hasil Pengujian Terdeteksi Asap pada Ruang 2	76
Tabel 4. 8 Pengujian Terdeteksi Suhu pada Ruang 1	78
Tabel 4. 9 Hasil Pengujian Terdeteksi Suhu pada Ruang 2	78
Tabel 4. 10 Pengujian Terdeteksi Api dan Asap pada Ruang 1	81
Tabel 4. 11 Hasil Pengujian Terdeteksi Api dan Asap pada Ruang 2	82
Tabel 4. 12 Pengujian Terdeteksi Api dan Suhu pada Ruang 1	85
Tabel 4. 13 Hasil Pengujian Terdeteksi Api dan Suhu pada Ruang 2	86
Tabel 4. 14 Hasil Pengujian Terdeteksi Asap dan Suhu pada Ruang 1	89
Tabel 4. 15 Pengujian Terdeteksi Asap dan Suhu pada Ruang 2	89
Tabel 4. 16 Pengujian ke-1 Ruang 1	93
Tabel 4. 17 Pengujian ke-2 Ruang 1	93
Tabel 4. 18 Pengujian ke-1 Ruang 2	94
Tabel 4. 19 Pengujian ke-2 Ruang 2	94

DAFTAR GAMBAR

Gambar 2. 1 Pengujian Kebakaran dengan Kolam Api	6
Gambar 2. 2 Pengujian Asap dengan CO Meter dan Sensor MQ-9	9
Gambar 2. 3 Flame Sensor	11
Gambar 2. 4 Sensor MQ-9	12
Gambar 2. 5 Sensor DHT-11	14
Gambar 2. 6 Arduino Mega 2560	16
Gambar 2. 7 Motor Servo SG-90	17
Gambar 2. 8 Pulse Wide Modulation (PWM) Motor Servo	18
Gambar 2. 9 DF Mini Player	19
Gambar 2. 10 Light Emitting Diode (LED)	20
Gambar 2. 11 Prinsip Kerja LED	21
Gambar 2. 12 Diagram kerja Internet of Things (IoT)	21
Gambar 2. 13 NodeMCU ESP8266	22
Gambar 2. 14 Arduino IDE	23
Gambar 2. 15 Tampilan Awal Firebase	24
Gambar 2. 16 Tampilan pada MIT App Inventor	25
Gambar 3. 1 Flowchart Tahapan Pelaksanaan Proyek Akhir	28
Gambar 3. 2 Rancangan Hardware Elektrik	30
Gambar 4. 1 Blok Diagram Prinsip Kerja Sistem Pendeteksi Kebakaran	35
Gambar 4. 2 Rancangan Konstruksi Prototype Pendeteksi Kebakaran	36
Gambar 4. 3 Tampak Depan Rancangan Prototype Pendeteksi Kebakaran	37
Gambar 4. 4 Tampak Samping Rancangan Prototype Pendeteksi Kebakaran	37
Gambar 4. 5 Tampak Atas Rancangan Prototype Pendeteksi Kebakaran	37
Gambar 4. 6 Tampak Depan Konstruksi Pendeteksi Kebakaran	38
Gambar 4. 7 Tampak Samping Konstruksi Pendeteksi Kebakaran	39
Gambar 4. 8 Tampak Atas Konstruksi Pendeteksi Kebakaran	39
Gambar 4. 9 Tampak Belakang Konstruksi Pendeteksi Kebakaran	39
Gambar 4. 10 Skema Pengkabelan Sistem Kontrol	40

Gambar 4. 11 Rangkaian Sistem Kontrol Pendeteksi Kebakaran.....	41
Gambar 4. 12 Skema Pengkabelan Flame Sensor dengan Arduino Mega 2560...	42
Gambar 4. 13 Rangkaian Flame Sensor dengan Arduino Mega 2560.....	43
Gambar 4. 14 Diagram Blok Pengujian Flame Sensor	44
Gambar 4. 15 Diagram Pengujian Flame Sensor terhadap Api pada Sudut 0° dan 60°	44
Gambar 4. 16 Pengujian Flame Sensor pada Jarak 4 cm dan Sudut 0°	44
Gambar 4. 17 Hasil Pengujian Flame Sensor pada Jarak 4 cm dan Sudut 0°	45
Gambar 4. 18 Pengujian Flame Sensor pada Jarak 12 cm dan Sudut 60°	45
Gambar 4. 19 Hasil Pengujian Flame Sensor pada Jarak 12 cm dan Sudut 60° ...	45
Gambar 4. 20 Pengujian Flame Sensor pada Jarak 16 cm dan Sudut 70°	46
Gambar 4. 21 Hasil Pengujian Flame Sensor pada Jarak 16 cm dan Sudut 70° ...	46
Gambar 4. 22 Skema Pengkabelan Sensor MQ-9 dengan Arduino Mega 2560...	48
Gambar 4. 23 Rangkaian Sensor MQ-9 dengan Arduino Mega 2560.....	49
Gambar 4. 24 Blok Diagram Pengujian Sensor MQ-9	50
Gambar 4. 25 Diagram Pengujian Sensor MQ-9 terhadap Asap	50
Gambar 4. 26 Pengujian Sensor MQ-9 terhadap Asap selama 1 Menit	50
Gambar 4. 27 Hasil Pengujian Sensor MQ-9 terhadap Asap selama 1 Menit.....	51
Gambar 4. 28 Skema Pengkabelan Sensor DHT-11 dengan Arduino Mega 2560 52	
Gambar 4. 29 Rangkaian Sensor DHT-11 dengan Arduino Mega 2560	52
Gambar 4. 30 Diagram Blok Pengujian Sensor DHT-11.....	53
Gambar 4. 31 Diagram Pengukuran Sensor DHT-11 terhadap Suhu Panas.....	54
Gambar 4. 32 Pengujian Sensor DHT-11 pada Jarak 10 cm terhadap Lilin.....	54
Gambar 4. 33 Hasil Pengujian Sensor DHT-11 pada Jarak 10 cm terhadap Lilin	54
Gambar 4. 34 Pengujian Sensor DHT-11 pada Jarak 15 cm terhadap Lilin.....	55
Gambar 4. 35 Hasil Pengujian Sensor DHT-11 pada Jarak 15 cm terhadap Lilin	55
Gambar 4. 36 Desain Menu Home pada Software MIT App Inventor.....	57
Gambar 4. 37 Tampilan Menu Home pada Smartphone	58
Gambar 4. 38 Desain Menu Login pada Software MIT App Inventor	59
Gambar 4. 39 Tampilan Menu Login pada Smartphone.....	60
Gambar 4. 40 Desain Menu Sign Up pada Software MIT App Inventor	60

Gambar 4. 41 Tampilan Menu Sign Up pada Smartphone	61
Gambar 4.42 Desain Tampilan Menu Monitoring dan Kontrol pada MIT App Inventor	62
Gambar 4. 43 Tampilan Menu Monitoring dan Kontrol pada Smartphone.....	63
Gambar 4.44 Tampilan Data yang disimpan pada Firebase	64
Gambar 4. 45 Tampilan Menu Designer pada MIT App Inventor	66
Gambar 4. 46 Program Pemanggilan Database ke Aplikasi MIT App Inventor ..	66
Gambar 4. 47 Program Penyimpanan Data dari MIT App Inventor ke Firebase .	67
Gambar 4. 48 Penempatan Komponen pada Ruang 1	70
Gambar 4. 49 Penempatan Komponen pada Ruang 2	71
Gambar 4. 50 Pengujian Terdeteksi Api pada Ruang 1	72
Gambar 4. 51 Pengujian Terdeteksi Api pada Ruang 2.....	72
Gambar 4. 52 Tampilan LCD Ketika Terdeteksi Api pada Ruang 1	72
Gambar 4. 53 Tampilan LCD Ketika Terdeteksi Api pada Ruang 2.....	73
Gambar 4. 54 Pengujian Terdeteksi Asap pada Ruang 1.....	74
Gambar 4. 55 Pengujian Terdeteksi Asap pada Ruang 2.....	74
Gambar 4. 56 Tampilan LCD Pengujian Asap pada Ruang 1	75
Gambar 4. 57 Tampilan LCD Pengujian Asap pada Ruang 2	75
Gambar 4. 58 Pengujian Terdeteksi Suhu pada Ruang 1.....	77
Gambar 4. 59 Pengujian Terdeteksi Suhu pada Ruang 2.....	77
Gambar 4. 60 Tampilan LCD Terdeteksi Suhu pada Ruang 1	77
Gambar 4. 61 Tampilan LCD Terdeteksi Suhu pada Ruang 2	78
Gambar 4. 62 Pengujian Terdeteksi Api dan Asap pada Ruang 1	79
Gambar 4. 63 Pengujian Terdeteksi Api dan Asap pada Ruang 2.....	80
Gambar 4. 64 Tampilan LCD Terdeteksi Api dan Asap pada Ruang 1.....	80
Gambar 4. 65 Tampilan LCD Terdeteksi Api dan Asap pada Ruang 2.....	80
Gambar 4. 66 Pengujian Terdeteksi Api dan Suhu pada Ruang 1	83
Gambar 4. 67 Pengujian Terdeteksi Api dan Suhu Ruang 2	84
Gambar 4. 68 Tampilan LCD Terdeteksi Api dan Suhu pada Ruang 1.....	84
Gambar 4. 69 Tampilan LCD Terdeteksi Api dan Suhu pada Ruang 2.....	84
Gambar 4. 70 Pengujian Terdeteksi Asap dan Suhu pada Ruang 1.....	87

Gambar 4. 71 Pengujian Terdeteksi Asap dan Suhu pada Ruang 2.....	88
Gambar 4. 72 Tampilan LCD Terdeteksi Asap dan Suhu pada Ruang 1	88
Gambar 4. 73 Tampilan LCD Terdeteksi Asap dan Suhu pada Ruang 2	88
Gambar 4. 74 Pengujian ke-1 Api, Asap dan Suhu pada Ruang 1	90
Gambar 4. 75 Pengujian ke-2 Api, Asap dan Suhu pada Ruang 1	91
Gambar 4. 76 Pengujian ke-1 Api, Asap dan Suhu pada Ruang 2	91
Gambar 4. 77 Pengujian ke-2 Api, Asap dan Suhu pada Ruang 2	91
Gambar 4. 78 Tampilan LCD Pengujian ke-1 Ruang 1	92
Gambar 4. 79 Tampilan LCD Pengujian ke-1 Ruang 2.....	92
Gambar 4. 80 Tampilan LCD Pengujian ke-2 Ruang 1	92
Gambar 4. 81 Tampilan LCD Pengujian ke-2 Ruang 2.....	92
Gambar 4. 82 Tampilan 1 Monitoring Data Sensor Ruang 1: (a) pada Serial Monitor Arduino IDE; dan (b) pada Smartphone	97
Gambar 4. 83 Tampilan 2 Monitoring Data Sensor Ruang 1: (a) pada Serial Monitor Arduino IDE; dan (b) pada Smartphone	97
Gambar 4. 84 Tampilan 3 Monitoring Data Sensor Ruang 1: (a) pada Serial Monitor Arduino IDE; dan (b) pada Smartphone	98
Gambar 4. 85 Tampilan Ketika Alarm Ruang 1 ON: (a) pada Smartphone; (b) pada Serial Monitor Arduino IDE; dan (c) Modul DF Mini Player ON.....	99
Gambar 4. 86 Tampilan Ketika Jendela Ruang 1 ON: (a) pada Smartphone; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi Jendela Terbuka	99
Gambar 4. 87 Tampilan Ketika Pintu Ruang 1 ON: (a) pada Smartphone; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi Pintu Terbuka	100
Gambar 4. 88 Tampilan Ketika Exhaust Ruang 1 ON: (a) pada Smartphone; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi Exhaust Aktif	100
Gambar 4. 89 Tampilan Ketika LED Ruang 1 ON: (a) pada Smartphone; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi LED Aktif.....	101
Gambar 4. 90 Tampilan Ketika Semua Action Ruang 1 ON: (a) pada Smartphone; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi Semua Action Aktif.....	101
Gambar 4. 91 (a) Notifikasi Terdeteksi Api pada Smartphone; (b) Tampilan Data Sensor pada Room 1	102

Gambar 4. 92 (a) Notifikasi Terdeteksi Asap >100 ppm pada Smartphone; (b) Tampilan Data Sensor pada Room 1	103
Gambar 4. 93 (a) Notifikasi Terdeteksi Suhu >35°C pada Smartphone; (b) Tampilan Data Sensor pada Room 1	103
Gambar 4. 94 (a) Notifikasi Terdeteksi Api dan Asap >110 ppm dan Suhu >38°C pada Smartphone; (b) Tampilan Data Sensor pada Room 1	104
Gambar 4. 95 Tampilan 1 Monitoring Data Sensor Ruang 2: (a) pada Serial Monitor Arduino IDE; dan (b) pada Smartphone	105
Gambar 4. 96 Tampilan 2 Monitoring Data Sensor Ruang 2: (a) pada Serial Monitor Arduino IDE; dan (b) pada Smartphone	105
Gambar 4. 97 Tampilan 3 Monitoring Data Sensor Ruang 2: (a) pada Serial Monitor Arduino IDE; dan (b) pada Smartphone	106
Gambar 4. 98 Tampilan Ketika Alarm Ruang 1 ON: (a) pada Smartphone; (b) pada Serial Monitor Arduino IDE; dan (c) Modul DF Mini Player ON.....	107
Gambar 4. 99 Tampilan Ketika Jendela Ruang 1 ON: (a) pada Smartphone; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi Jendela Terbuka	107
Gambar 4. 100 Tampilan Ketika Pintu Ruang 1 ON: (a) pada Smartphone; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi Pintu Terbuka.....	108
Gambar 4. 101 Tampilan Ketika Exhaust Ruang 1 ON: (a) pada Smartphone; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi Exhaust Aktif	108
Gambar 4. 102 Tampilan Ketika LED Ruang 1 ON: (a) pada Smartphone; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi LED Aktif	109
Gambar 4. 103 Tampilan Semua Action Ruang 1 ON: (a) pada Smartphone; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi Semua Action Aktif	109
Gambar 4. 104 (a) Notifikasi Terdeteksi Api pada Smartphone; (b) Tampilan Data Sensor pada Room 2	110
Gambar 4. 105 (a) Notifikasi Terdeteksi Asap >100 ppm pada Smartphone; (b) Tampilan Data Sensor pada Room 2	111
Gambar 4. 106 (a) Notifikasi Terdeteksi Suhu >35°C pada Smartphone; (b) Tampilan Data Sensor pada Room 2	111

Gambar 4. 107 (a) Notifikasi Terdeteksi Api dan Asap >110 ppm dan Suhu >38°C pada Smartphone; (b) Tampilan Data Sensor pada Room 2..... 112

DAFTAR LAMPIRAN

LAMPIRAN 1: RIWAYAT HIDUP PERORANGAN

LAMPIRAN 2: PROGRAM ARDUINO

LAMPIRAN 3: PROGRAM NODEMCU ESP8266

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kebakaran merupakan sebuah bencana yang ditandai dengan adanya api atau pembakaran yang tidak terkendali hingga muncul asap dan gas sehingga membahayakan kehidupan manusia dan lingkungan di sekitarnya [1]. Peristiwa kebakaran bisa terjadi di mana saja mulai dari perumahan, kantor, jalan raya, pusat perbelanjaan, hutan, laut, jalan raya, kebun maupun persawahan [2]. Kebakaran dapat menimbulkan kerugian yang sangat besar baik bagi pihak yang mengalami musibah maupun orang lain yang berada di sekitarnya. Kerugian ini dapat berupa kerusakan bangunan, terganggunya kegiatan usaha, rusaknya fasilitas umum bahkan sampai menimbulkan korban jiwa [3]. Data kebakaran di Indonesia yang dihimpun oleh Badan Nasional Penanggulangan Bencana (BNPB) dalam kurun waktu 7 tahun sejak tahun 2010 hingga 2017 telah terjadi bencana kebakaran sebanyak 1.212 kejadian [4].

Peristiwa kebakaran dapat terjadi kapan saja dan banyak sekali yang menjadi faktor penyebab dari kebakaran tersebut diantaranya korsleting listrik, penggunaan daya yang berlebihan karena lupa mematikan lampu, lupa mematikan kompor, kebocoran tabung gas, putung rokok, petasan, dan lain-lain [5]. Tidak ada satupun yang bisa memprediksi kapan akan terjadi kebakaran karena biasanya kebakaran baru dapat diketahui ketika api sudah terlanjur membesar dan keluar asap yang banyak dari gedung atau bangunan lainnya. Saat kondisi kebakaran sudah semakin parah maka akan sangat sulit untuk melakukan pemadaman kebakaran dan penyelamatan korban apalagi jika lokasi kebakaran sulit dijangkau tim pemadam kebakaran [6]. Oleh karena itu, diperlukan sebuah sistem pendeteksi terhadap indikasi kebakaran untuk mengurangi resiko dan bahaya yang disebabkan oleh kebakaran tersebut apalagi hal ini menjadi standar pembangunan gedung-gedung [7].

Pendeteksian terhadap kasus kebakaran ini sudah pernah dilakukan dalam beberapa penelitian sebelumnya yaitu penelitian [8] yang membuat sebuah sistem pendeteksi kebakaran pada rumah dengan cara mengirimkan notifikasi kepada pengguna melalui *email* berupa lokasi terjadinya kebakaran menggunakan *Google Maps*. Adapun penelitian [9] juga membuat sistem pendeteksi kebakaran dengan cara memberikan peringatan berupa alarm dan mengirimkan pesan kepada pengguna dengan mencantumkan dari mana pesan tersebut dikirim dan waktu diterimanya pesan tersebut. Sedangkan penelitian [10] membuat sebuah *prototype* pendeteksi kebakaran gedung menggunakan algoritma *fuzzy logic* yang dapat memantau kondisi di suatu ruangan berdasarkan parameter yang ada.

Berdasarkan beberapa penelitian yang telah dilakukan di atas maka dalam proyek akhir ini akan dibuat sebuah alat pendeteksi kebakaran dengan tambahan pembahasan berupa tindakan mengaktifkan alarm dan *exhaust* serta membuka pintu dan jendela ketika terdeteksi kebakaran. Dan yang terakhir yaitu tindakan mengaktifkan LED sebagai indikator pemadaman api saat terjadi kebakaran. Alat pendeteksi yang digunakan terdiri dari tiga buah sensor yaitu sensor api, sensor asap dan sensor suhu. Data berupa perubahan kadar asap, api dan suhu pada gedung yang dibaca oleh ketiga sensor tersebut akan ditampilkan secara *real time* ke *smartphone* sehingga keadaan gedung dapat selalu dipantau. Saat ini kemajuan teknologi komunikasi telah melahirkan sebuah konsep *Internet of Things* (IoT) yang memberikan kemudahan bagi manusia dalam mentransfer data dengan menggunakan internet sebagai medianya. Melalui konsep ini, proses yang berlangsung dalam sebuah sistem dapat menjadi lebih efektif dan efisien. Berkaitan dengan implementasi dari teknologi tersebut, maka pada proyek akhir ini akan dibuat “Sistem *Monitoring* Data Sensor dan Kontrol Gedung dari Bahaya Kebakaran Berbasis *Internet of Things* (IoT)”. Dengan adanya alat ini diharapkan pendeteksian terhadap bahaya kebakaran dapat dilakukan secara cepat sebagai bentuk penanganan awal saat terjadi kebakaran di dalam gedung.

1.2 Rumusan Masalah

Permasalahan dalam peristiwa kebakaran adalah bencana kebakaran baru dapat terdeteksi ketika api sudah terlanjur membesar dan keluar asap yang banyak dari dalam gedung atau bangunan lainnya sehingga dapat dibuat rumusan masalah sebagai berikut.

1. Bagaimana cara merancang dan membuat konstruksi *prototype* pendeteksi kebakaran?
2. Bagaimana cara membuat sistem kontrol menggunakan Arduino Mega 2560 untuk mengontrol alarm, *exhaust*, motor servo dan LED sebagai indikator pemadaman api saat terjadi kebakaran gedung?
3. Bagaimana cara membuat desain *interface* aplikasi pada *smartphone* menggunakan MIT App Inventor?
4. Bagaimana cara mengirimkan data secara nirkabel untuk menampilkan data sensor dan mengontrol alarm, *exhaust*, motor servo dan LED melalui *smartphone* serta mengirimkan notifikasi kebakaran ke *smartphone*?

1.3 Batasan Masalah

Untuk memperjelas arah pembahasan dari proyek akhir ini, maka diberikan batasan masalah sebagai berikut.

1. Sistem pendeteksi kebakaran dibuat dalam bentuk miniatur dengan ukuran dinding-dinding miniatur yaitu panjang 90 cm, lebar 45 cm dan tinggi 30 cm sedangkan bagian atap miniatur berukuran panjang 102 cm, lebar 45 cm dan tinggi 12 cm untuk keperluan simulasi.
2. Tidak melakukan pemadaman api secara *real* melainkan hanya menggunakan LED sebagai indikator bahwa sprinkler aktif untuk menyemprotkan air ketika terdeteksi kebakaran.

1.4 Tujuan

Adapun tujuan yang ingin dicapai dalam pembuatan proyek akhir ini adalah sebagai berikut.

1. Merancang dan membuat konstruksi *prototype* pendeteksi kebakaran.

2. Membuat sistem kontrol menggunakan Arduino Mega 2560 untuk mengontrol alarm, *exhaust*, motor servo dan LED sebagai indikator pemadaman api saat terjadi kebakaran gedung.
3. Membuat desain *interface* aplikasi pada *smartphone* menggunakan MIT App Inventor.
4. Mengirimkan data secara nirkabel untuk menampilkan data sensor dan mengontrol alarm, *exhaust*, motor servo dan LED melalui *smartphone* serta mengirimkan notifikasi kebakaran ke *smartphone*.

BAB II

LANDASAN TEORI

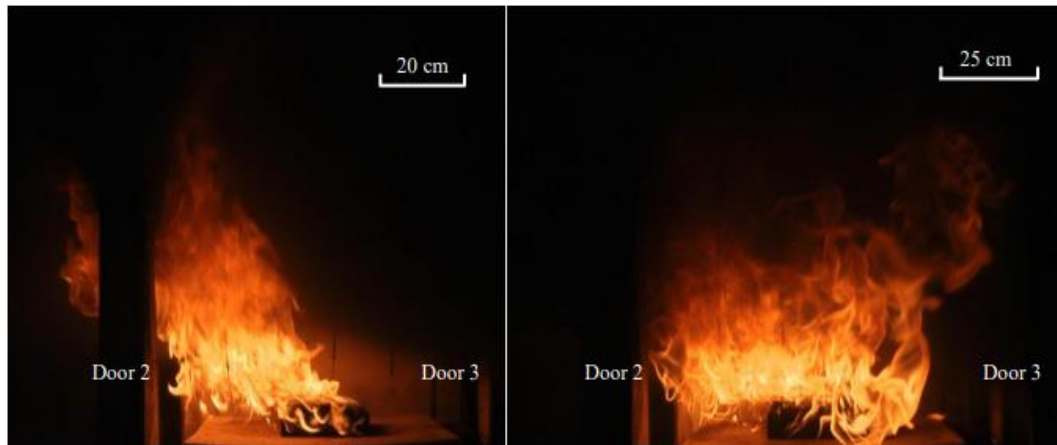
2.1 Pengertian Kebakaran

Kebakaran merupakan suatu reaksi oksidasi yang berlangsung dengan cepat karena adanya bahan bakar yang disertai dengan timbulnya percakan api atau nyala api secara tidak terkendali. Api merupakan suatu reaksi kimia yang terbentuk dari tiga unsur atau yang lebih dikenal dengan segitiga api yaitu bahan bakar, oksigen dan panas yang menimbulkan efek panas dan cahaya. Terjadinya peristiwa kebakaran ini dibutuhkan suatu komponen lain yaitu rantai reaksi kimia. Rantai reaksi kimia adalah peristiwa ketiga unsur tersebut saling bereaksi secara kimiawi sehingga menghasilkan nyala api atau kebakaran [11].

2.1.1 Ambang Batas Kebakaran

Suatu kondisi dikatakan terjadi kebakaran apabila sudah mencapai nilai ambang batas kebakaran. Dikatakan bahwa suhu api dapat berkisar mulai dari 400°F-9000°F (200°C-4980°C). Suhu api yang dihasilkan dari peristiwa kebakaran dapat bervariasi tergantung dari sumber bahan bakar dan kandungan oksigen [12].

Pada penelitian [13] dibuat sebuah model bangunan dengan skala 1:3 untuk menguji distribusi suhu dan radiasi ruang api berdasarkan pengaruh dari tingkatan lantai. Suhu ruangan pada model bangunan berada dalam kondisi normal antara 30°C-32°C. Kemudian di dalam ruangan tersebut diletakkan sumber api dengan ukuran kolam api 20 cm² dapat dilihat pada Gambar 2.1. Kolam api dibuat menggunakan papan baja dengan ketebalan 2 mm. Pengujian dilakukan pada ruangan berukuran 0,8 m x 0,8 m x 1,2 m dan didapatkan suhu ruangan pada saat terjadi kebakaran mencapai 100°C hingga naik menjadi 600°C. Ketika diberikan sumber api dengan ukuran kolam api 25 cm² didapatkan nyala api memenuhi seluruh ruangan dan sebagian nyala api menyebar ke luar seperti terlihat pada Gambar 2.1 sehingga diperoleh suhu ruangan yang sangat tinggi lebih dari 640°C yang menunjukkan terjadi kebakaran *flashover* [14].



Gambar 2. 1 Pengujian Kebakaran dengan Kolam Api [14]

Selain suhu api, ambang batas kebakaran juga ditentukan berdasarkan kadar asap yang dihasilkan. Pembakaran Konfigurasi Prancis dan AS menyatakan bahwa peristiwa kebakaran menghasilkan gas emisi beracun seperti gas CO dengan kadar 28.000 ppm dan 38.200 ppm yang akan terdeteksi antara 6 menit dan 8 menit pada ruangan berukuran 3,66 m x 2,44 m x 2,44 m [15].

2.2 Studi Literatur Terkait

Penelitian mengenai *prototype* alat pendeteksi kebakaran di dalam gedung ini sudah banyak dilakukan oleh para peneliti dari berbagai perguruan tinggi di Indonesia, terutama untuk keperluan tugas akhir. *Prototype* alat pendeteksi kebakaran ini bertujuan untuk mendeteksi terjadinya kebakaran di dalam gedung secara dini untuk mengatasi bahaya dan dampak buruk yang diakibatkan dari peristiwa kebakaran. Pada penelitian sebelumnya, sistem pendeteksi kebakaran ini sudah dibahas dalam berbagai pokok permasalahan. Ada beberapa persamaan dan perbedaan dari semua penelitian yang telah dilakukan, baik dari segi tujuan penelitian, pokok pembahasan, mikrokontroler yang digunakan, jenis sensor yang digunakan, hingga komponen penyusun lainnya.

Beberapa penelitian yang sudah pernah dilakukan berkaitan dengan sistem pendeteksi kebakaran menggunakan Arduino yang dapat dijadikan referensi dalam pembuatan proyek akhir diantaranya penelitian [8] yaitu sistem pendeteksi kebakaran rumah menggunakan buzzer sebagai peringatan kebakaran dan

pengiriman notifikasi ke *smartphone* melalui aplikasi Blynk dengan menggunakan mikrokontroler Arduino Uno sebagai pengontrol sensor api KY-026, sensor asap MQ-2 dan sensor suhu DHT11. Sistem ini dapat melakukan *monitoring* dan menanggulangi kebakaran dengan menggunakan pompa air sebagai media pemadam api. Dari hasil penelitian diketahui bahwa sistem ini berhasil mendeteksi adanya api, asap dan suhu serta pompa air mampu bekerja dengan baik saat sensor mendeteksi terjadi kebakaran dan buzzer berbunyi sebagai peringatan kebakaran.

Selanjutnya penelitian [9] yaitu sistem alarm dan *monitoring* kebakaran rumah yang terdiri dari NodeMCU dan sensor suhu yang terhubung dengan internet. Dari hasil penelitian diketahui pada suhu diatas 37°C maka alarm peringatan bahaya kebakaran akan aktif dan dikirimkan notifikasi ke android bahwa telah terjadi kebakaran. Pada suhu di atas 42°C maka modul relay akan bekerja dan menghidupkan pompa air untuk memadamkan api. Dengan sistem IoT yang mempercepat pengiriman data maka informasi kebakaran dapat diketahui dengan cepat oleh pengguna. Sistem ini membutuhkan waktu lebih cepat ≤ 5 menit jika dibandingkan dengan sistem sebelumnya yang membutuhkan waktu sekitar 30 menit untuk menginformasikan kebakaran kepada pihak-pihak terkait sehingga rumah sudah terlanjur habis terbakar.

Penelitian [16] yaitu prototipe pendeteksi kebakaran menggunakan konsep *Internet-of-Things*. Pada penelitian ini dirancang alat pendeteksi kebakaran menggunakan *flame sensor* (sensor api), sensor gas MQ-2 dan sensor suhu DHT11. Platform IoT yang digunakan pada penelitian ini adalah aplikasi Blynk dan modul komunikasi yang digunakan adalah ESP8266. Dari hasil pengujian diketahui bahwa sensor bekerja dengan baik dan data yang dideteksi oleh sensor dapat dikirim ke telepon seluler melalui aplikasi Blynk. Dari segi spesifikasi teknis, sensor MQ-2 dapat mendeteksi gas lebih baik jika sumber gas semakin dekat. Adapun sensor api mempunyai kemampuan deteksi yang baik terhadap api sampai jarak maksimum 20 cm, sedangkan sensor DHT11 mampu mendeteksi dua indikator sekaligus yaitu suhu dan kelembaban. Dengan data ini bisa membantu penulis dalam merancang pendeteksi kebakaran bahwa sensor api dapat bekerja dengan baik pada jarak di bawah 20 cm.

Selanjutnya penelitian [10] yaitu prototipe pendeteksi kebakaran gedung menggunakan Algoritma *Fuzzy Logic* berbasis *Internet of Things* (IoT). Pada penelitian ini dibuat prototipe pendeteksi kebakaran menggunakan Arduino mega 2560 sebagai pengontrol dengan sensor DHT11, MQ-2, *flame sensor* dan buzzer sebagai alarm. Algoritma *fuzzy logic* digunakan untuk memantau kondisi di suatu ruangan apakah berbahaya atau tidak yang nantinya akan mengaktifkan buzzer. Dari hasil pengujian diketahui bahwa alat terhubung dengan *database thingspeak* dan pembacaan *database* mampu bekerja dengan baik. Pada sensor DHT11 terjadi *error* sebesar 1.18% untuk suhu dan 2.04% untuk kelembaban. Sedangkan untuk *flame sensor*, jarak api dapat memengaruhi panjang gelombang yang ditangkap. Pada pengujian ini juga diketahui bahwa semakin besar masukan sensor maka hasil keluaran nilai *fuzzy* juga akan semakin tinggi dan berbahaya.

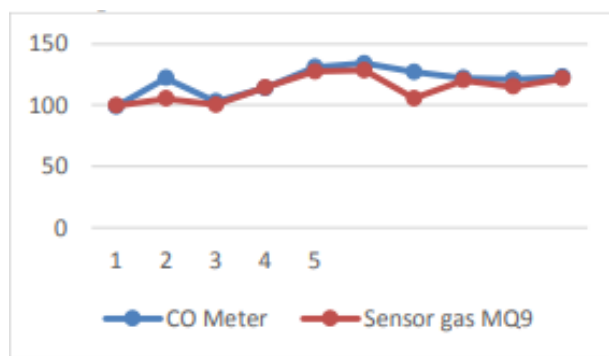
Kemudian penelitian [17] yaitu rancang bangun sistem deteksi kebakaran pada rumah berbasis IoT. Penelitian ini menggunakan aplikasi Blynk untuk memantau kondisi rumah dan memberikan notifikasi melalui *e-mail* sebagai peringatan terindikasi dan terjadi kebakaran. Dari hasil penelitian dapat diketahui bahwa ketika *flame sensor* (sensor api), sensor MQ-2 (sensor asap) dan sensor DHT11 (sensor suhu) mendeteksi adanya api, asap dan suhu maka buzzer akan berbunyi dan sistem akan mengirimkan notifikasi pada aplikasi Blynk yang selanjutnya akan dikirimkan kepada pengguna melalui *e-mail*. Informasi yang dikirimkan berupa peringatan terjadi kebakaran dan lokasi terjadinya kebakaran menggunakan *Google Maps*.

Berdasarkan beberapa penelitian yang telah dilakukan sebelumnya, maka pada proyek akhir ini akan dibuat sebuah *prototype* sistem pendeteksi kebakaran menggunakan tiga buah sensor yaitu *flame sensor* sebagai pendeteksi api, MQ-9 sebagai pendeteksi asap, dan DHT-11 sebagai pendeteksi suhu. Pada sistem pendeteksi kebakaran ini juga dilakukan beberapa pengembangan yaitu dengan menambahkan tindakan-tindakan penanganan awal untuk mengurangi bahaya yang ditimbulkan dari peristiwa kebakaran yaitu mengaktifkan alarm sebagai peringatan bahaya kebakaran. Ada juga tindakan mengaktifkan *exhaust* serta membuka pintu dan jendela untuk mengeluarkan asap dari dalam gedung saat terjadi peristiwa

kebakaran. Dan yang terakhir yaitu mengaktifkan LED sebagai indikator tindakan pemadaman api saat terjadi kebakaran. Sistem pendeteksi kebakaran ini akan bekerja sesuai dengan pemrograman yang telah dibuat di Arduino.

2.3 Sensor

Untuk mendeteksi terjadinya kebakaran digunakan tiga buah sensor yaitu sensor suhu, sensor api dan sensor asap [1] yang memperlihatkan bahwa pemakaian banyak sensor dapat menambah kemampuan pendeteksian dini terhadap kebakaran lebih baik. Penelitian [18] telah melakukan pengujian pengambilan data sensor api (*flame sensor*) yang dapat membaca panjang gelombang berkisar antara 760 nm-1100 nm. Cahaya infra merah yang dipancarkan oleh api mempunyai panjang gelombang sekitar 700 nm-1 mm sehingga sensor ini dapat digunakan sebagai pendeteksi kebakaran. Adapun pada penelitian [19] dilakukan pengujian untuk membandingkan pembacaan sensor asap (MQ-9) dengan CO meter untuk mengetahui tingkat kepekaan terhadap kepekatan CO. Diketahui pada gambar 2.4 terdapat hasil 10 kali pengujian sebagai berikut.



Gambar 2. 2 Pengujian Asap dengan CO Meter dan Sensor MQ-9 [19]

Dari data tersebut diketahui bahwa nilai yang diperoleh CO meter memiliki rentang 99 ppm-134 ppm sedangkan nilai sensor MQ-9 memiliki rentang 99,67 ppm-128,4 ppm. Perbedaan kadar asap ini terjadi dikarenakan pada saat pengujian, asap yang timbul memiliki kuantitas yang berbeda-beda. Dari pengujian, diperoleh rata-rata *error* sebesar 4,85% sehingga dapat disimpulkan bahwa sensor asap MQ-9 mampu bekerja dengan baik [19].

Pada penelitian [16] dilakukan pengujian terhadap sensor DHT-11 yang bertujuan untuk mengetahui kemampuan sensor dalam mendeteksi suhu ruang dan kelembabannya. Berdasarkan spesifikasinya, sensor DHT-11 mempunyai rentang pengukuran suhu antara 0-50°C, kelembaban 20%-95%, resolusi suhu dan kelembaban masing-masing 8 bit, serta tegangan kerja 3,3 atau 5 VDC. Pada proses pengujian, sebuah lilin dinyalakan untuk mendapatkan suhu panas kemudian diuji menggunakan sensor DHT-11 pada jarak yang bervariasi. Nilai suhu dan kelembaban yang diperoleh akan ditampilkan dalam sebuah LCD. Dari hasil pengujian menunjukkan bahwa sensor ini mampu mendeteksi suhu dan kelembaban dengan baik sehingga dapat digunakan sebagai pendeteksi kebakaran. Adapun hasil pengujian suhu dan kelembaban ruang dapat dilihat pada Tabel 2.1 sebagai berikut [16].

Tabel 2. 1 Tabel Pengujian Sensor DHT-11 [16]

Jarak (cm)	Kriteria Pengujian	Sumber Api	LCD Menampilkan Suhu dan Kelembaban	Suhu (°C)	Kelembaban (RH)
1	Deteksi panas dari api	Lilin	√	29	33
2	dan LCD		√	28	35
3	menampilkan hasil		√	27	36
4	pengukuran suhu dan		√	25	38
5	kelembaban		√	25	41

2.3.1 Flame Sensor

Sensor api (*flame sensor*) merupakan sensor yang dapat mendeteksi nyala api dengan ketelitian tinggi dalam rentang panjang gelombang 760-1100 nm. Sensor *photodiode* yang terdapat pada sensor ini berfungsi untuk mendeteksi adanya nyala api di sekitar sensor tersebut. Jarak *flame sensor* tidak boleh terlalu dekat dengan objek yang dideteksi untuk menghindari kerusakan sensor. Sensor ini memiliki 3 buah pin yaitu pin Vcc untuk dihubungkan ke sumber tegangan, pin AO

digunakan sebagai keluaran sinyal analog (*analog output*) dan pin GND untuk dihubungkan ke *ground* [20]. Untuk bentuk fisik dari *flame sensor* ini dapat dilihat pada Gambar 2.3 berikut.



Gambar 2. 3 *Flame Sensor* [20]

2.3.1.1 Prinsip Kerja *Flame Sensor*

Flame sensor dapat bekerja dengan baik untuk mendeteksi nyala api melalui keberadaan spektrum cahaya infra red maupun ultraviolet menggunakan metode optik. Perlu diketahui bahwa *flame sensor* ini digunakan untuk mendeteksi keberadaan api bukan panas. Hasil pendeteksian berupa nyala api akan diteruskan ke *microprocessor* pada unit *flame sensor* untuk membedakan spektrum cahaya yang terdapat pada nyala api tersebut dengan sistem *delay* selama 2-3 detik sehingga dapat mendeteksi api secara dini. *Flame sensor* dirancang khusus menggunakan transduser berupa infrared (IR) sebagai *sensing sensor* yang dapat menyerap cahaya pada gelombang tertentu untuk membedakan spektrum cahaya dari nyala api dengan spektrum cahaya lainnya, seperti cahaya lampu, kilatan petir, *welding arc*, *metal grinding*, dan lain-lain sehingga tidak terjadi kesalahan dalam melakukan pendeteksian api [21]. Spesifikasi dari *flame sensor* ini dapat dilihat pada Tabel 2.2 berikut

Tabel 2. 2 Spesifikasi *Flame Sensor* [21]

No.	Spesifikasi	Nilai
1.	Tegangan Kerja	3.3-5 V
2.	Format keluaran	Digital (<i>High</i> atau <i>Low</i>)
3.	Rentang deteksi panjang gelombang	760-1100 nm
4.	Sudut deteksi	60°, sangat sensitif terhadap spektrum api

No.	Spesifikasi	Nilai
5.	Jarak pendeteksian sensor	20 (4.8 V)-100 cm (1 V)
6.	<i>Responsive time</i>	15 μ s
7.	<i>Size</i>	22 x 30 mm

2.3.2 Sensor MQ-9

Sensor asap MQ-9 merupakan sensor yang memiliki sensitivitas tinggi pada karbon monoksida (CO), metana dan LPG. Sensor MQ-9 memiliki 4 buah pin, yaitu pin Vcc untuk dihubungkan ke sumber tegangan (5V), pin GND untuk dihubungkan ke *ground*, pin DO sebagai *digital output* dan pin AO sebagai *analog output*. Sensor ini memerlukan 2 buah tegangan yaitu tegangan *heater* VH dan tegangan uji VC. VH berfungsi untuk mengaktifkan sensor yang bekerja terhadap suhu, sementara VC berfungsi untuk mendeteksi tegangan (VRL) pada RL yaitu resistansi beban yang dipasang seri dengan sensor. Sensor ini memiliki polaritas cahaya sehingga VC memerlukan daya DC. VC dan VH dapat menggunakan rangkaian daya yang sama agar kinerja sensor lebih baik [22]. Untuk bentuk fisik dari sensor MQ-9 ini dapat dilihat pada Gambar 2.4 berikut.



Gambar 2. 4 Sensor MQ-9 [22]

2.3.2.1 Prinsip Kerja Sensor MQ-9

Sensor MQ-9 memiliki pemanas internal yang akan mulai panas jika diberikan sumber tegangan sebesar 5 V. Resistansi internal sensor akan berubah seiring dengan perubahan densitas gas yang dideteksi. Nilai inilah yang diolah oleh Arduino dan kemudian akan menghasilkan *output* berupa nilai kadar gas yang dideteksi dalam satuan ppm. Untuk mendapatkan data nilai yang benar dan akurat

maka dibutuhkan waktu pemanasan awal dengan cara menghubungkan sensor dengan catu daya selama beberapa waktu hingga siap digunakan [23]. Adapun spesifikasi dari sensor MQ-9 ini dapat dilihat pada tabel 2.3 berikut.

Tabel 2. 3 Spesifikasi dari Sensor MQ-9 [23]

No.	Spesifikasi	Keterangan
1.	Jenis Sensor	Merupakan sensor semikonduktor
2.	<i>Detection gas</i>	CO, CH ₄ , LPG
3.	<i>Concentration gas</i>	CO 10 ppm-1000 ppm dan gas yang mudah terbakar 100 ppm-1000 ppm
4.	<i>Circuit Voltage (Vc)</i>	5 V
5.	<i>Heating Voltage (Vh) High</i>	5 V
6.	<i>Heating Voltage (Vh) Low</i>	1.4 V
7.	<i>Heating Time Th (High)</i>	60s
8.	<i>Heating Time Th (Low)</i>	90s
9.	<i>Sensing resistance</i>	2K-20K pada gas CO 100 ppm
10.	<i>Heater resistance (RL)</i>	33 ohm
11.	<i>Heater consumption</i>	<340mW
12.	<i>Slope</i>	≥5
13.	<i>Preheat time</i>	≥48 jam

2.3.3 Sensor DHT-11

Sensor DHT-11 merupakan sensor yang berfungsi untuk mendeteksi suhu dan kelembaban. Sensor ini memiliki 3 buah pin yaitu pin Vcc (+) untuk dihubungkan ke sumber tegangan, pin Dout merupakan pin *output* berupa sinyal digital yang bisa dihubungkan ke mikrokontroler untuk diolah dan pin GND (-) untuk dihubungkan ke *ground*. Kelebihan dari sensor DHT-11 adalah dari segi

kualitas pembacaan data suhu dan kelembaban lebih cepat dan data yang terbaca tidak mudah terinterferensi. Sensor DHT-11 memiliki fitur kalibrasi nilai pembacaan suhu dan kelembaban secara otomatis yang cukup akurat [17]. Adapun bentuk fisik dari sensor DHT-11 dapat dilihat pada Gambar 2.5 berikut.



Gambar 2. 5 Sensor DHT-11 [17]

2.3.3.1 Prinsip Kerja dari Sensor DHT-11

Sensor DHT-11 mengukur suhu menggunakan *thermistor* dengan tipe NTC (*Negative Temperature Coefficient*) yang terpasang pada permukaan sensor. *Thermistor* NTC merupakan sebuah resistor variable dengan nilai resistansi yang berbanding terbalik dengan kenaikan suhu sehingga ketika suhu semakin tinggi maka nilai resistansi dari *thermistor* akan semakin kecil. Sensor DHT-11 akan mengeluarkan *output* berupa nilai analog berdasarkan nilai resistansi dari *thermistor*. Nilai *output* analog ini kemudian akan dibaca dan dikonversi oleh arduino menjadi nilai suhu (dalam bentuk °C) dan kelembaban (dalam bentuk %) [24]. Spesifikasi dari sensor DHT-11 dapat dilihat pada Tabel 2.4 berikut [25].

Tabel 2. 4 Spesifikasi dari Sensor DHT-11 [25]

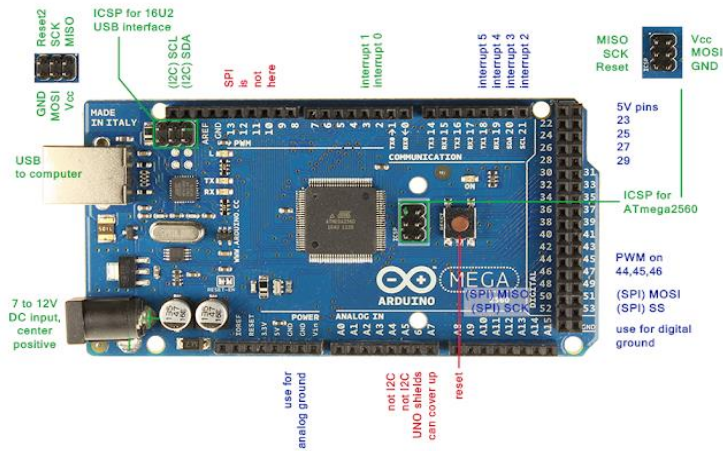
No.	Spesifikasi	Keterangan
1.	Tegangan kerja	3.3-5 V
2.	Arus maksimum	2.5 mA
3.	Range pengukuran kelembaban	20-80%
4.	Akurasi pengukuran kelembaban	5%
5.	Range pengukuran suhu	0-50°C

No.	Spesifikasi	Keterangan
6.	Akurasi pengukuran suhu	2°C
7.	Kecepatan pengambilan data	1 Hz per detik
8.	Ukuran sensor	15.5 mm x 12 mm x 5.5 mm
9.	Jumlah pin sensor	4 buah dengan jarak 0.1”

2.4 Arduino Mega 2560

Mikrokontroler adalah sebuah chip yang berfungsi sebagai pengontrol pada rangkaian elektronika dan terdapat program berisi perintah untuk melakukan sesuatu di dalamnya. Mikrokontroler umumnya terdiri dari *processor*, memori, input dan output serta unit pendukung seperti ADC (*Analog to Converter*) yang berfungsi untuk mengubah data analog menjadi data digital sehingga bisa dibaca oleh mikrokontroler. Dengan demikian mikrokontroler adalah sebuah alat pengontrol rangkaian elektronika yang di dalamnya terdapat program berisi perintah untuk melakukan sesuatu sesuai keinginan serta mempunyai *input* dan *output* [26]. Jenis mikrokontroler yang paling banyak digunakan adalah Arduino. Oleh karena itu dalam pembuatan *prototype* sistem pendeteksi kebakaran ini jenis Arduino yang digunakan adalah Arduino Mega 2560.

Arduino Mega 2560 merupakan sebuah *board* Arduino yang menggunakan IC Mikrokontroler ATmega 2560. Arduino ini memiliki pin I/O yang cukup banyak yaitu sekitar 54 *digital Input/Output*. Adapun 15 buah pin digunakan sebagai *output* PWM, 16 buah *analog input* dan 4 UART (port serial perangkat keras). Selain itu, Arduino Mega 2560 juga dilengkapi dengan kristal 16 Mhz. Penggunaan Arduino Mega 2560 ini cukup mudah yaitu dengan menghubungkan *power* dari USB ke PC/laptop atau menggunakan adaptor 7-12 V DC [27].



Gambar 2. 6 Arduino Mega 2560 [27]

Tabel 2. 5 Tabel Spesifikasi Arduino Mega 2560 [27]

No	Spesifikasi	Nilai
1.	Mikrokontroler	ATmega2560
2.	Tegangan Operasional	5 V
3.	Tegangan <i>Input</i> (rekomendasi)	7-12 V
4.	Tegangan <i>Input</i> (limit)	6-20 V
5.	Pin Digital I/O	54 (of which 15 provide PWM output)
6.	Pin <i>Analog Input</i>	16
7.	Arus DC per Pin I/O	20 mA
8.	Arus DC untuk Pin 3.3 V	50 mA
9.	Memori <i>Flash</i>	256 KB of which 8 KB used by <i>bootloader</i>
10.	SRAM	8 KB
11.	EEPROM	4 KB
12.	<i>Clock Speed</i>	16 MHz
13.	LED_BUILTIN	13
14.	Panjang	101.52 mm
15.	Lebar	53.3 mm
16.	Berat	37 g

2.5 Servo Motor SG-90

Motor servo merupakan sebuah motor DC yang mempunyai prinsip kerja dengan sistem *closed loop* dimana posisi rotornya akan dikembalikan ke rangkaian kontrol dari motor servo. Di dalam sebuah motor servo terdapat motor DC, rangkaian *gear*, potensiometer dan rangkaian kontrol. Batas sudut putaran dari motor servo diatur oleh potensiometer sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim dari kabel motor servo. Adapun bentuk fisik dari servo motor SG-90 dapat dilihat pada Gambar 2.7 berikut.



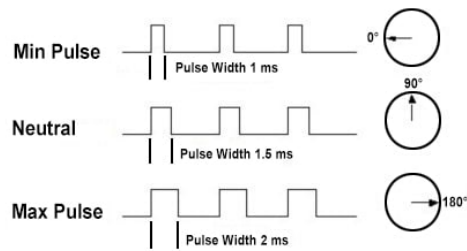
Gambar 2. 7 Motor Servo SG-90 [28]

Tabel 2. 6 Konfigurasi Warna Kabel Motor Servo [28]

No.	Warna Kabel	Deskripsi
1.	Coklat	Kabel <i>Ground</i> yang dihubungkan ke sistem <i>Ground</i>
2.	Merah	Kabel <i>Power</i> yang dihubungkan ke sumber tegangan 5V.
3.	<i>Orange</i>	Kabel yang berfungsi untuk mengirimkan sinyal PWM ke motor servo.

2.5.1 Prinsip Kerja Servo Motor SG-90

Sudut putaran motor servo dapat diatur dengan memberikan sinyal PWM (*Pulse Wide Modulation*). Lebar pulsa sinyal ini akan menentukan posisi sudut putaran yang dihasilkan oleh motor servo. Apabila ketika lebar pulsa dalam waktu 1,5 ms maka sudut putaran motor servo sebesar 90°. Apabila lebar pulsa <1,5 ms maka motor servo akan berputar ke kiri yaitu ke posisi sudut 0°. Sedangkan ketika lebar pulsa >1,5 ms maka motor servo akan berputar ke kanan yaitu ke posisi sudut 180°. Adapun sinyal PWM dapat dilihat pada Gambar 2.8 berikut [29].



Gambar 2. 8 *Pulse Wide Modulation (PWM) Motor Servo* [29]

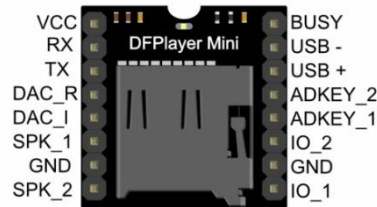
Tabel 2. 7 Karakteristik Servo Motor SG-90 [29]

No.	Karakteristik	Keterangan
1.	Dimensi	22.6 mm x 21.8 mm x 11.4 mm
2.	Berat (hanya motor)	9 gram
3.	Kecepatan	0.12 S/60°
4.	<i>Pulse Width</i>	500-2400 μ s
5.	<i>PWM Period</i>	20 ms (50 Hz)
6.	Tegangan kerja	4.8-6 V
7.	Arus	<500 mA
8.	<i>Temperature range</i>	-30 sampai 60°C
9.	Panjang kabel	150 mm
10.	<i>Stall Torque</i>	1.98 Kg/Cm
11.	<i>Gear type</i>	Plastic
12.	<i>Limit angle</i>	180° (\pm 10°)
13.	<i>Neutral position</i>	1500 μ s

2.6 DF Mini Player

Modul DF Mini Player merupakan modul suara atau *music player* dalam format mp3 dilengkapi dengan koneksi serial terhadap mp3 dan perangkat keras MWV. Modul ini dapat terhubung dengan kartu SD dan didukung dengan sistem FAT16 dan FAT32. DF Mini Player dapat memainkan musik dan dihubungkan ke *speaker* maupun *headset*. Modul ini berukuran 2 cm x 2 cm x 1,2 cm yang dapat

langsung terhubung dengan sumber baterai dan dapat dikombinasikan dengan Arduino atau lainnya dengan koneksi melalui pin RX/TX [30]. Adapun bentuk fisik dari DF Mini Player ini dapat dilihat pada Gambar 2.9 berikut.



Gambar 2. 9 DF Mini Player [30]

Tabel 2. 8 Penjelasan Pin DF Mini Player [30]

Pin	Deskripsi	Keterangan
Vcc	<i>Input Voltage</i>	DC 3.2~5.0 V; Type DC 4.2 V
RX	<i>UART serial input</i>	
TX	<i>UART serial output</i>	
DAC_R	<i>Audio output right channel</i>	<i>Drive earphone and amplifier</i>
DAC_L	<i>Audio output left channel</i>	<i>Drive earphone and amplifier</i>
SPK2	<i>Speaker-</i>	<i>Drive speaker less than 3 W</i>
GND	<i>Ground</i>	<i>Power GND</i>
SPK1	<i>Speaker+</i>	<i>Drive speaker less than 3 W</i>
IO1	<i>Trigger port 1</i>	<i>Short press to play previous (long press to decrease volume)</i>
GND	<i>Ground</i>	<i>Power GND</i>
IO2	<i>Trigger port 2</i>	<i>Short press to play next (long press to increase volume)</i>
ADKEY1	<i>AD Port 1</i>	<i>Trigger play first segment</i>
ADKEY2	<i>AD Port 2</i>	<i>Trigger play fifth segment</i>
USB+	<i>USB+ DP</i>	<i>USB Port</i>
USB-	<i>USB- DM</i>	<i>USB Port</i>
BUSY	<i>Playing Status</i>	<i>Low means playing/High means no playing</i>

2.7 LED (*Light Emitting Diode*)

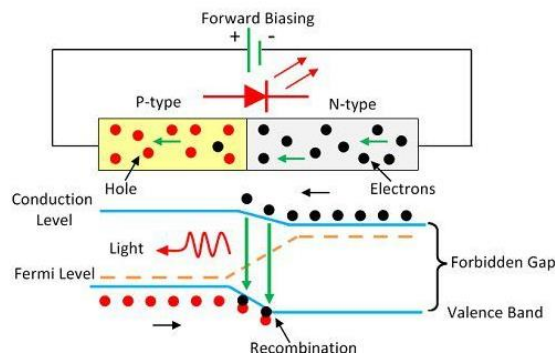
LED (*Light Emitting Diode*) merupakan komponen elektronika yang dapat memancarkan cahaya ketika diberikan arus listrik dengan arah maju (*forward bias*). LED terbuat dari bahan semikonduktor dan termasuk ke dalam golongan dioda. Warna yang dipancarkan oleh LED tergantung dari jenis bahan semikonduktor yang digunakan. Selain itu, LED juga dapat memancarkan sinar inframerah yang tidak dapat ditangkap oleh mata seperti *remote control* TV ataupun perangkat elektronik lainnya. Adapun bentuk fisik dari LED dapat dilihat pada Gambar 2.10 berikut.



Gambar 2. 10 *Light Emitting Diode* (LED) [31]

2.7.1 Prinsip Kerja LED

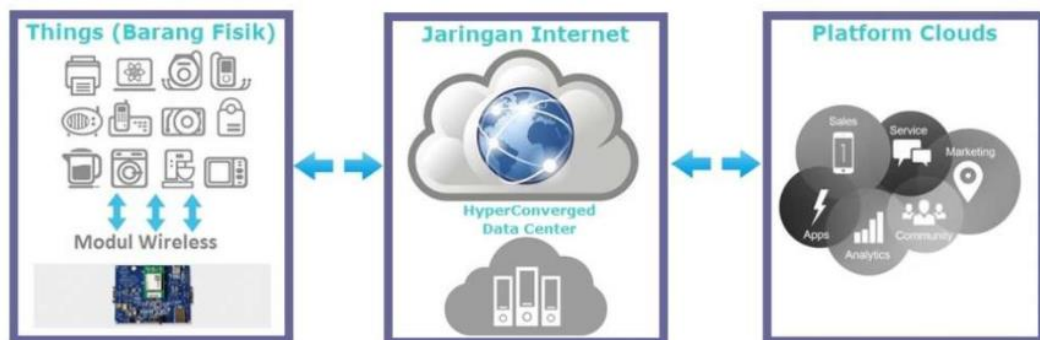
Ketika LED dalam kondisi bias maju maka *electron* yang berada di sisi N diode melintasi jalur PN dan masuk ke daerah P bergabung dengan *hole* yang berada di daerah p dan kemudian memancarkan cahaya. Seperti yang kita ketahui bahwa pita konduksi yang memiliki banyak *electron* merupakan daerah N. Sedangkan pita kelambu merupakan daerah P yang memiliki *hole* di dalamnya. Jadi, ketika *electron* berpindah dari pita konduksi tinggi ke pita valensi atau bergabung dengan *hole* maka akan melepaskan energi dalam bentuk *foton* berupa cahaya. Proses ini dikenal sebagai *electroluminescence* [31].



Gambar 2. 11 Prinsip Kerja LED [31]

2.8 *Internet of Thing (IoT)*

Internet of Things (IoT) merupakan sebuah sistem komunikasi yang memiliki kemampuan mentransmisikan data melalui jaringan internet untuk koneksi dalam sebuah sistem seperti sensor, aktuator atau pengendali [32]. Cara kerja *Internet of Things* yaitu dengan memanfaatkan pemrograman yang berisi perintah untuk menghasilkan sebuah interaksi sistem secara otomatis tanpa campur tangan manusia dan dalam jangkauan sistem kerja yang luas.



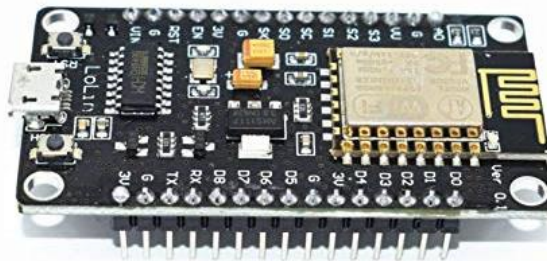
Gambar 2. 12 Diagram kerja *Internet of Things (IoT)* [33]

Konsep cara kerja *Internet of Things* mengacu pada tiga elemen utama, yaitu barang fisik yang sudah dilengkapi dengan modul IoT, seperti *handphone*, kamera, printer, dan lain-lain. Kemudian perangkat yang bisa menghubungkan ke jaringan internet seperti modem dan *Router Wireless Speedy*. Terakhir adalah *Cloud Data Center* yang berfungsi sebagai tempat menyimpan aplikasi dan *database*. Pada

database inilah program yang berisi perintah untuk melakukan sesuatu disimpan dan suatu waktu dapat dipanggil ketika program ingin dijalankan [33].

2.9 NodeMCU ESP8266

NodeMCU merupakan *platform IoT* bersifat *open source firmware* untuk mengendalikan sebuah aplikasi melalui jaringan *wifi* atau rangkaian aplikasi lain. NodeMCU sebagai *board* dari ESP8266 saat dijadikan suatu *host* aplikasi maka akan bekerja langsung dari aplikasi tersebut. Dalam ESP8266 ini telah terintegrasi dengan *cache* yang dapat meningkatkan kinerja dari sistem aplikasi tersebut. NodeMCU mempunyai spesifikasi berupa *board* berbasis ESP8266 dengan *Serial Wifi Single on Chips (SoC)* dengan *onboard USB to TTL* serta IEEE 802.11b/g/n sebagai standar untuk *wireless*. NodeMCU juga dilengkapi dengan *micro USB port* yang berfungsi untuk pemrograman maupun *power supply* serta tombol reset dan *flash* [34]. Untuk bentuk fisik dari NodeMCU ESP8266 dapat dilihat pada Gambar 2.13 berikut.



Gambar 2. 13 NodeMCU ESP8266 [34]

Tabel 2. 9 Tabel Spesifikasi NodeMCU ESP8266 [34]

No.	Spesifikasi	Keterangan
1.	Mikrokontroler	ESP8266
2.	Tegangan <i>input</i> yang disarankan	3.3V - 12V
3.	GPIO	13 Pin
4.	Kanal PWM	10 Kanal
5.	10 bit ADC Pin	1 Pin
6.	Memori <i>Flash</i>	4 Mb
7.	<i>Wifi</i>	IEEE 802.11 b/g/n

No.	Spesifikasi	Keterangan
8.	Frekuensi	2.4 GHz – 22.5 GHz
9.	<i>Clock Speed</i>	40/26/24 MHz
10.	USB to Serial Converter	CH340G
11.	Panjang	57 mm
12.	Lebar	30 mm
13.	USB <i>Port</i>	Micro USB

2.10 ARDUINO IDE

IDE (*Integrated Development Environment*) *software* yang digunakan Arduino untuk melakukan pemrograman sehingga dapat menjalankan perintah-perintah yang diprogram di dalamnya. Bahasa pemrograman Arduino (*sketch*) merupakan bahasa pemrograman yang ditulis sendiri oleh aplikasi Arduino IDE, mirip dengan pemrograman bahasa C dan sudah dilakukan perubahan dari bahasa aslinya sehingga memudahkan pemula dalam melakukan pemrograman. Program ini ditulis dalam sebuah *editor text* dan disimpan dalam file dengan format *.ino*. Tampilan Arduino IDE dapat dilihat pada Gambar 2.14 berikut.



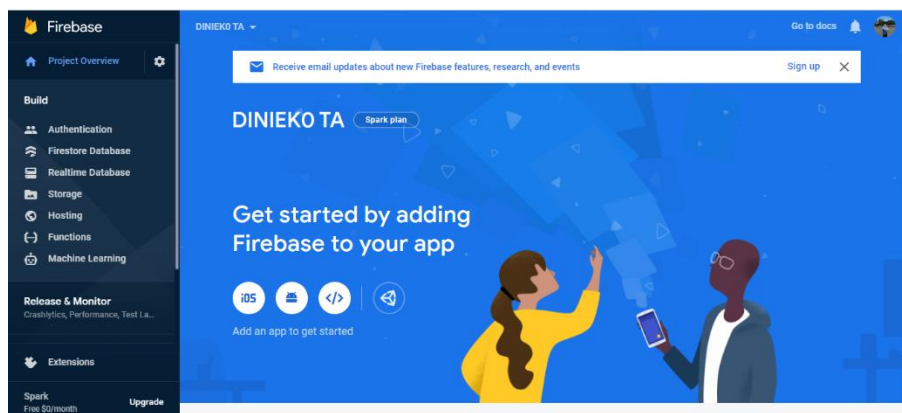
Gambar 2. 14 Arduino IDE [35]

Pada IC mikrokontroler Arduino telah ditanam suatu program yaitu *Bootlader* yang berfungsi sebagai perantara *compiler* Arduino dengan mikrokontroler. Pada program IDE terdapat kotak hitam pada bagian bawah yang berfungsi untuk menampilkan status seperti pesan *error compile* dan *upload* program kemudian dibagian pojok kanan atas terdapat serial monitor yang

berfungsi untuk menampilkan hasil eksekusi dari program yang dijalankan oleh Arduino IDE [35].

2.11 *Firebase*

Firebase merupakan salah satu *platform* berbasis *Backend as a Service* (BaaS) yang disediakan oleh *Google* untuk menyimpan data sehingga dapat digunakan untuk pengembangan aplikasi *mobile*. Tampilan *firebase* dapat dilihat pada Gambar 2.15 berikut.



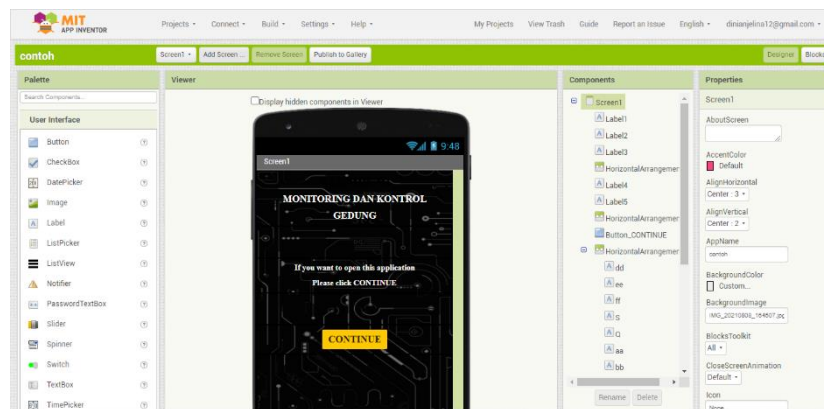
Gambar 2. 15 Tampilan Awal *Firebase*

Adapun fitur yang disediakan oleh *firebase* ini antara lain *Firebase Analytics*, *Firebase Authentication*, *Firebase Cloud Messaging and Notification*, *Firebase Crash Reporting*, *Firebase Remote Config* dan *Firebase Real Time Database*. Fitur yang memiliki peran cukup penting adalah *Firebase Real Time Database* dan *Firebase Remote Config*. *Firebase Real Time Database* adalah *database* yang digunakan untuk menyimpan data dan disinkronkan dalam *cloud* secara *real time* dan tetap tersedia meskipun aplikasi dalam keadaan *offline*. Adapun *Firebase Remote Config* adalah fitur yang memungkinkan pengguna untuk mengubah atau mengganti beberapa konfigurasi iOS atau Android tanpa harus memberikan *update* melalui App Store atau Play Store [36].

2.12 *MIT App Inventor*

MIT App Inventor merupakan aplikasi yang disediakan oleh *Google* dan bersifat *web open source* dibawah pengawasan *Massachusetts Institute of*

Technology (MIT). MIT adalah sebuah *platform* untuk membuat aplikasi android dalam bentuk pemrograman *visual* pada *smartphone*. App Inventor menggunakan *interface* secara gratis yang memungkinkan pengguna untuk mengubah logika dalam bentuk objek *visual* dan dapat dijalankan dalam perangkat *smartphone*. MIT App Inventor ini memudahkan dalam pembuatan aplikasi sederhana sehingga tidak perlu mempelajari bahasa pemrograman terlalu banyak. Desain aplikasi android dapat dibuat dengan berbagai macam *layout* dan komponen yang tersedia pada MIT App Inventor. Tampilan pada MIT App Inventor dapat dilihat pada Gambar 2.16 berikut.



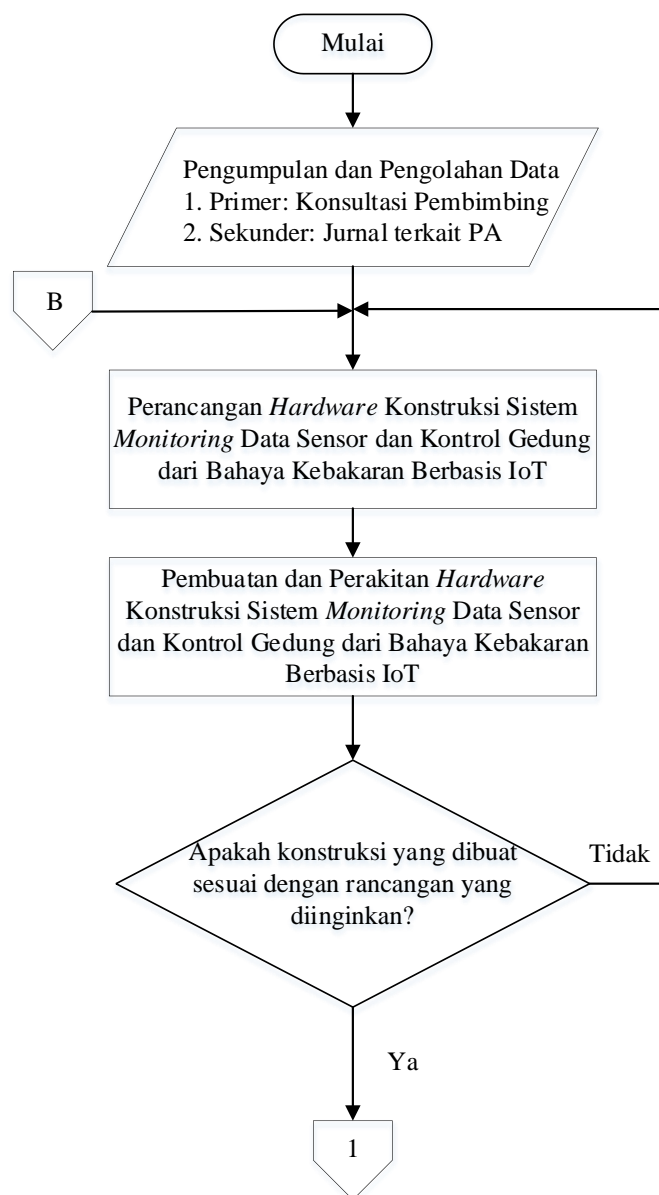
Gambar 2.16 Tampilan pada MIT App Inventor

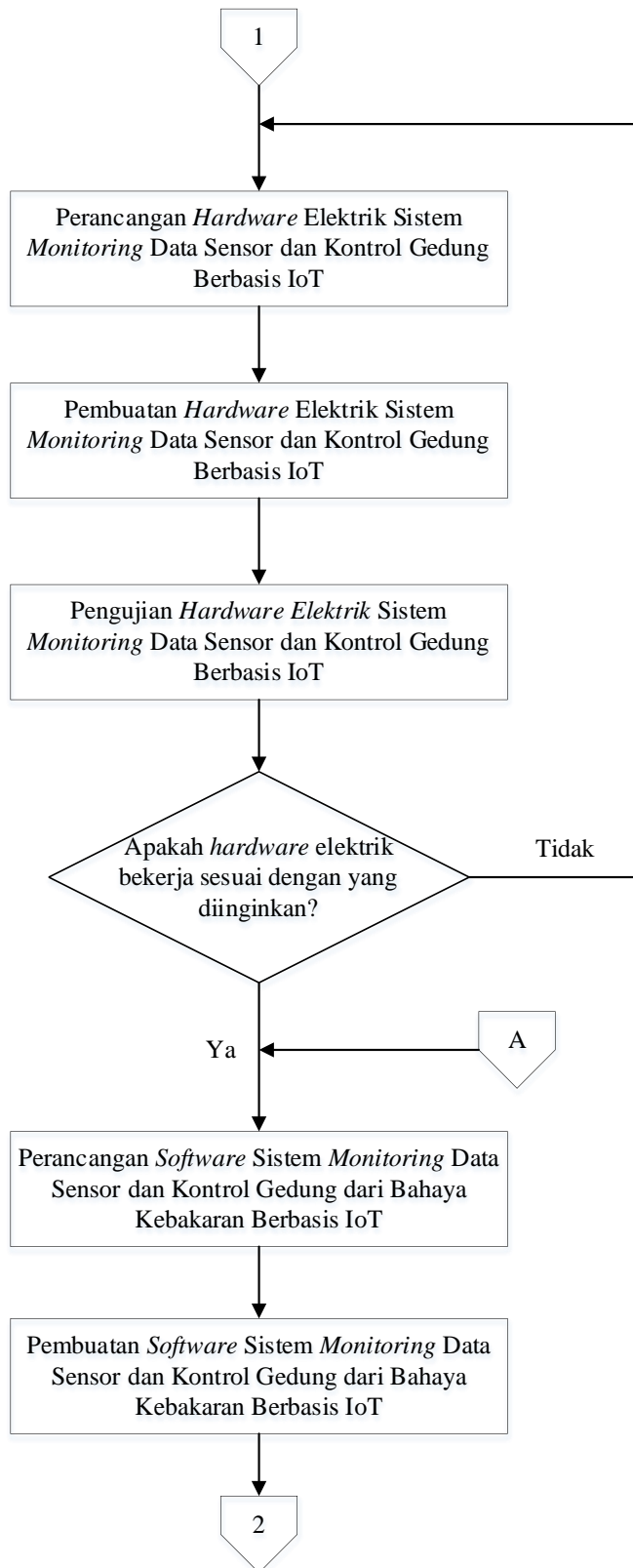
MIT App Inventor mempunyai dua jendela utama yaitu *Designer* yang berfungsi untuk membuat tampilan *visual* pada aplikasi dan *Blocks* yang berfungsi untuk membuat program untuk menjalankan perintah tertentu. Pada bagian *Designer*, pengguna dapat membuat tampilan *visual* aplikasi dengan menggunakan komponen-komponen yang tersedia. Setelah tampilan *visual* aplikasi selesai dibuat maka pada bagian *Blocks*, pengguna bisa membuat program untuk menjalankan fungsi yang ditampilkan pada tampilan *visual* aplikasi yang sudah dibuat. Pengguna dapat menguji aplikasi yang telah dibuat menggunakan *smartphone* secara langsung atau melalui *emulator* dari *smartphone* yang berjalan di komputer atau laptop. MIT App Inventor *server* juga menyimpan semua proyek yang telah dibuat dan memungkinkan pengguna untuk masuk kembali ke akun mereka untuk melanjutkan proyek yang telah dibuat [37].

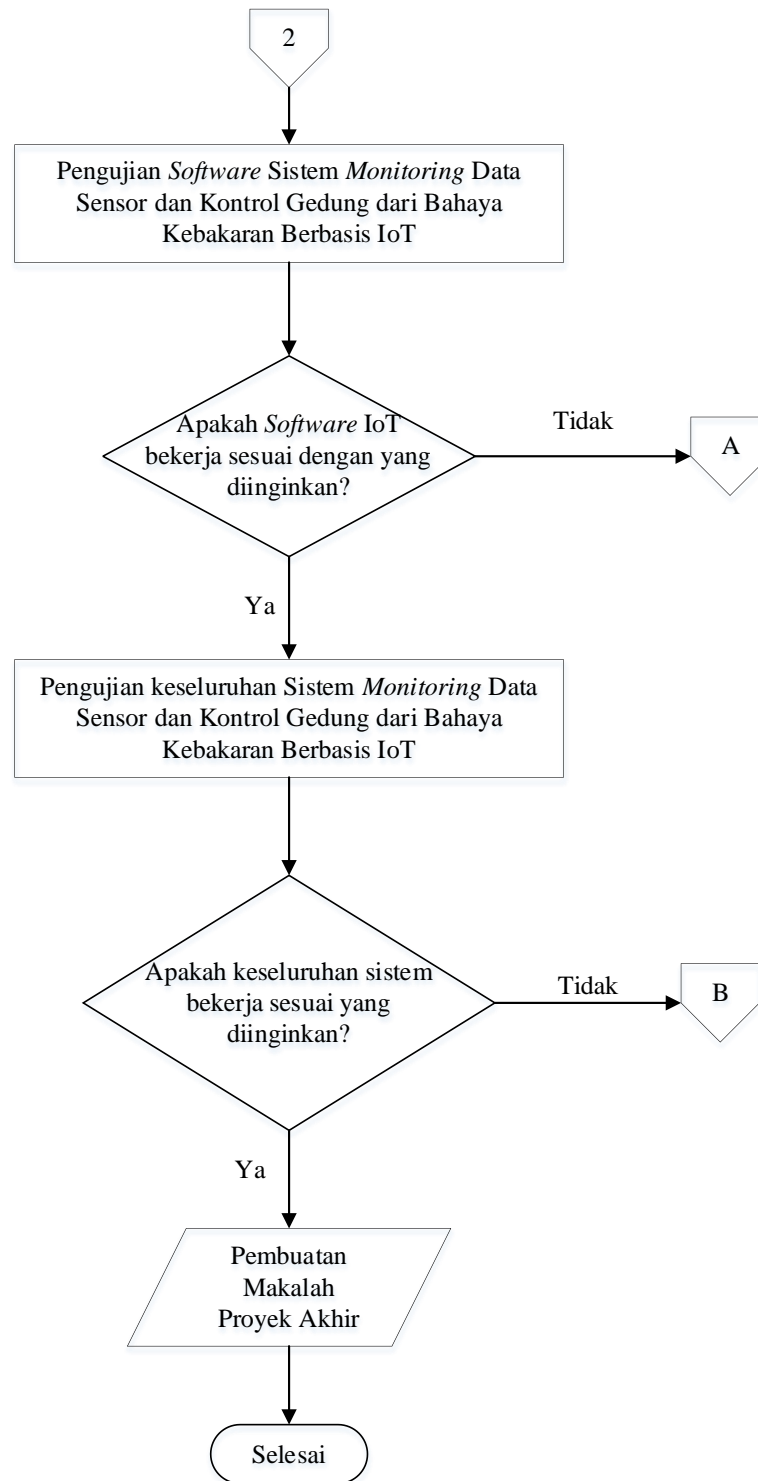
BAB III

METODE PELAKSANAAN

Dalam melaksanakan proyek akhir ini, ada beberapa tahapan yang dilakukan untuk mempermudah proses pembuatan proyek akhir. Tahapan-tahapan pelaksanaan proyek akhir ini dapat digambarkan melalui *flowchart* sebagai berikut.







Gambar 3. 1 *Flowchart* Tahapan Pelaksanaan Proyek Akhir

3.1 Pengumpulan dan Pengolahan Data

Pada tahap ini dilakukan pengumpulan data berupa studi literatur yang berkaitan dengan judul proyek akhir. Pengumpulan studi literatur ini bertujuan untuk mendapatkan informasi berupa data-data yang dapat dijadikan sumber referensi dalam pengerjaan proyek akhir maupun penulisan makalah proyek akhir. Melalui studi literatur ini juga dapat diketahui gambaran tentang sistem pendeteksi kebakaran yang sudah pernah dilakukan pada penelitian sebelumnya yang akan dikembangkan dalam proyek akhir ini.

Ada 2 metode pengumpulan data yang dilakukan pada tahap ini yaitu pengumpulan data primer (langsung) dan sekunder (tidak langsung). Pengumpulan data primer yaitu melakukan konsultasi dengan dosen pembimbing berkaitan dengan pengerjaan proyek akhir. Sedangkan pengumpulan data sekunder yaitu berupa pengumpulan jurnal-jurnal yang berhubungan secara langsung dengan proyek akhir.

Tahap selanjutnya yaitu mengolah data yang sudah didapatkan dari berbagai sumber tersebut dan dirumuskan menjadi sebuah ide baru dalam pembuatan proyek akhir ini.

3.2 Rancangan *Hardware* dan *Software*

Tahap perancangan *hardware* bertujuan untuk menentukan bentuk fisik dari miniatur pendeteksi kebakaran dan penempatan sensor pada miniatur tersebut. Selain itu, pada tahap ini juga dirancang sistem kontrol untuk mengendalikan sistem pendeteksi kebakaran. Sedangkan perancangan *software* bertujuan untuk merancang sistem *monitoring* dan kontrol IoT, *interface* aplikasi android, serta pengiriman notifikasi kebakaran ke *smartphone*.

3.2.1 Rancangan *Hardware*

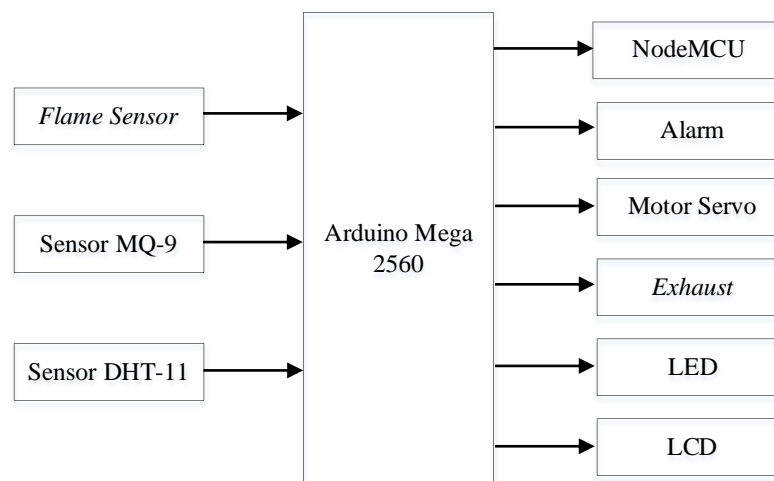
Rancangan *hardware* pada *prototype* pendeteksi kebakaran ini terdiri dari dua jenis yaitu *hardware* mekanik dan *hardware* elektrik. Pada tahap ini dibuat rancangan bentuk fisik (konstruksi) dari miniatur *prototype* pendeteksi kebakaran

dengan menggunakan aplikasi *sketch up*. Pada tahap ini juga ditentukan alat dan bahan yang digunakan serta ukuran dari *prototype* yang akan dibuat.

Adapun rancangan *hardware* secara mekanik yang akan dibuat adalah sebagai berikut.

1. Pembuatan dinding-dinding miniatur menggunakan bahan akrilik yang terdiri dari 2 ruangan dengan masing-masing ruangan memiliki ukuran panjang 45 cm, lebar 45 cm, dan tinggi 30 cm. Setiap ruangan memiliki 1 buah pintu utama, 1 buah pintu darurat, dan 2 buah jendela.
2. Pemasangan *flame sensor*, sensor MQ-9, sensor DHT-11, mikrokontroler (Arduino Mega 2560), NodeMCU ESP8266, DF Mini Player, *speaker*, *exhaust*, motor servo, dan LED pada miniatur.

Sementara pada tahap perancangan *hardware* elektrik dibuat rancangan skema rangkaian kontrol menggunakan Arduino Mega 2560 yang digunakan untuk mengontrol *prototype* sistem pendeteksi kebakaran sebagai upaya penanganan bahaya kebakaran.



Gambar 3. 2 Rancangan *Hardware* Elektrik

3.2.2 Rancangan *Software*

Pada tahap perancangan *software* ini dibuat rancangan berupa:

1. Pemrograman Arduino untuk menampilkan perubahan data dari *flame sensor*, sensor MQ-9 dan sensor DHT-11 serta mengontrol DF Mini Player, *speaker*, *exhaust*, motor servo, dan LED.

2. Membuat rancangan *interface* aplikasi android pada *smartphone* untuk menampilkan perubahan data dari sensor api (*flame sensor*), sensor asap (sensor MQ-9) dan sensor suhu (sensor DHT-11) dan mengontrol DF Mini Player, *speaker*, *exhaust*, motor servo, dan LED secara manual melalui *smartphone*.
3. Pemrograman untuk pengiriman notifikasi ke *smartphone* sebagai peringatan bahaya kebakaran menggunakan MIT App Inventor.

3.3 Pembuatan *Hardware* dan *Software*

Tahap pembuatan *hardware* dan *software* merupakan tahap merealisasikan rancangan alat yang sudah dibuat sebelumnya.

3.3.1 Pembuatan *Hardware*

Pada tahap pembuatan *hardware* secara mekanis dilakukan dengan pembuatan bentuk fisik dari *prototype* sesuai dengan ukuran yang sudah ditentukan. Alat dan bahan yang digunakan dalam pembuatan *prototype* juga disesuaikan dengan rancangan yang sudah ditentukan sebelumnya.

Adapun tahapan dalam pembuatan *hardware* secara mekanis adalah sebagai berikut.

1. Melubangi akrilik bening untuk membentuk pintu dan jendela dengan menggunakan gerinda listrik.
2. Membuat dinding-dinding miniatur dengan cara merekatkan akrilik bening yang sudah dibuat pintu dan jendela pada proses sebelumnya membentuk persegi panjang yang terdiri dari 2 ruangan dengan masing-masing ruangan memiliki ukuran panjang 45 cm, lebar 45 cm, dan tinggi 25 cm. Setiap ruangan memiliki 1 buah pintu utama, 1 buah pintu darurat, dan 2 buah jendela.
3. Memasang *flame sensor*, sensor MQ-9, sensor DHT-11, mikrokontroler (Arduino Mega), NodeMCU ESP8266, DF Mini Player, *speaker*, *exhaust*, motor servo, dan LED pada miniatur.

3.3.2 Pembuatan *Software*

Tahap pembuatan *software* juga dilakukan sesuai dengan rancangan yang sudah ditentukan sebelumnya. Tahapan pembuatan *software* ini meliputi *software* untuk mikrokontroler dan *software* untuk IoT.

Tahapan pembuatan *software* untuk mikrokontroler adalah sebagai berikut.

1. Pemrograman Arduino Mega 2560 untuk menampilkan perubahan data sensor dari sensor api (*flame sensor*), sensor asap (sensor MQ-9) dan sensor suhu (sensor DHT-11).
2. Pemrograman Arduino Mega 2560 untuk mengaktifkan DF Mini Player, *speaker*, *exhaust*, motor servo dan LED sebagai tindakan penanganan kebakaran.

Tahapan pembuatan *software* untuk IoT adalah sebagai berikut.

1. Pembuatan desain *interface* untuk aplikasi android pada *smartphone* menggunakan *platform* MIT App Inventor.
2. Pemrograman untuk menampilkan perubahan data dari sensor api (*flame sensor*), sensor asap (sensor MQ-9) dan sensor suhu (sensor DHT-11) dan mengontrol DF Mini Player, *speaker*, *exhaust*, motor servo, dan LED secara manual menggunakan MIT App Inventor.
3. Pemrograman untuk pengiriman notifikasi ke *smartphone* sebagai peringatan bahaya kebakaran menggunakan MIT App Inventor.

3.4 Pengujian *Hardware* dan *Software*

Setelah tahap pembuatan *hardware* dan *software*, tahap selanjutnya yaitu melakukan pengujian terhadap *hardware* dan *software* dalam sistem *monitoring* data sensor dan kontrol gedung dari bahaya kebakaran. Tujuan dari tahap pengujian ini adalah untuk mengetahui apakah sistem *monitoring* data sensor dan kontrol gedung dari bahaya kebakaran berbasis IoT dapat bekerja sesuai dengan target yang diinginkan.

3.4.1 Pengujian *Hardware*

1. Pengujian Arduino Mega 2560 dengan *flame sensor* untuk mengetahui kemampuan deteksi *flame sensor* terhadap api.
2. Pengujian Arduino Mega 2560 dengan sensor MQ-9 untuk mengetahui kemampuan deteksi sensor MQ-9 terhadap asap.
3. Pengujian Arduino Mega 2560 dengan sensor DHT-11 untuk mengetahui kemampuan deteksi sensor DHT-11 terhadap suhu.
4. Pengujian Arduino Mega 2560 dengan *flame sensor*, sensor MQ-9, sensor DHT-11, DF Mini Player, *speaker*, *exhaust*, motor servo, dan LED ketika ada api, asap, atau suhu atau ketiganya yang bertujuan untuk mengetahui apakah sistem dapat berfungsi dengan baik.

3.4.2 Pengujian *Software*

1. Pengujian *interface* aplikasi android untuk *monitoring* data sensor melalui *smartphone*.
2. Pengujian untuk mengaktifkan DF Mini Player, *speaker*, *exhaust*, motor servo, dan LED secara manual melalui *smartphone*.
3. Pengujian pengiriman notifikasi sebagai peringatan bahaya kebakaran ke *smartphone*.

3.4.3 Pengujian Keseluruhan Sistem

Pada tahap ini dilakukan pengujian secara keseluruhan terhadap sistem *monitoring* data sensor dan kontrol gedung dari bahaya kebakaran berbasis IoT untuk mengetahui apakah keseluruhan sistem dapat berfungsi dengan baik. Pengujian ini dilakukan dengan cara membuat suatu kondisi di dalam ruangan ketika hanya ada api atau asap atau suhu apakah sistem akan bekerja. Kemudian dibuat suatu kondisi ketika hanya ada api dan asap, atau api dan suhu, atau asap dan suhu apakah sistem akan bekerja. Dan yang terakhir ketika ada api, asap dan suhu yang mencapai ambang batas yang ditentukan apakah sistem pendeteksian kebakaran akan bekerja. Setelah itu

barulah dilakukan pengujian sistem *monitoring* data sensor dan kontrol gedung dari bahaya kebakaran berbasis IoT.

3.5 Analisis Data

Tahap selanjutnya yaitu melakukan analisis data dari hasil pengujian sistem yang telah dilakukan baik dari segi konstruksi, *hardware*, dan *software*. Jika masih terdapat kekurangan maka pada tahap inilah dicari solusi untuk melakukan perbaikan pada kekurangan tersebut sehingga didapatkan hasil yang optimal.

3.6 Pembuatan Makalah Proyek Akhir

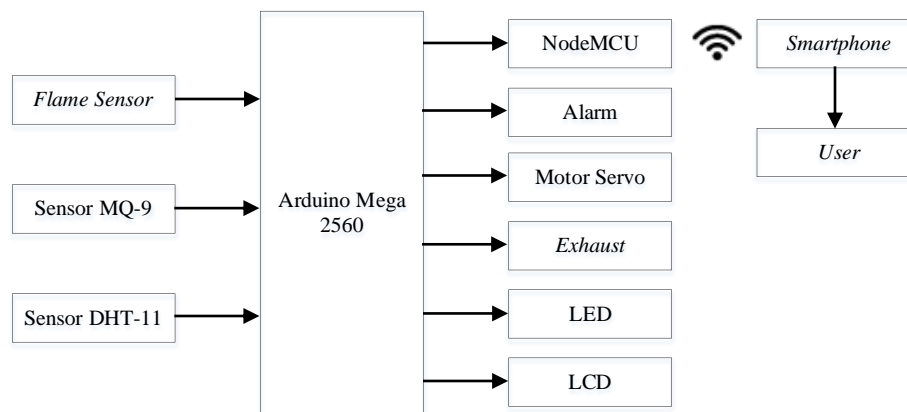
Tahap ini merupakan tahap terakhir dalam proses pembuatan proyek akhir. Pada makalah proyek akhir ini harus dicantumkan segala sesuatu yang berhubungan dengan pembuatan proyek akhir mulai dari pendahuluan, landasan teori, metode pelaksanaan, pembahasan, serta kesimpulan dan saran.

BAB IV PEMBAHASAN

Pada bab 4 ini akan dibahas mengenai proses dalam pembuatan proyek akhir mulai dari proses perancangan dan pembuatan konstruksi, perancangan dan pembuatan sistem kontrol, perancangan dan pembuatan *interface* pada *smartphone*, perancangan dan pembuatan sistem *monitoring* dan kontrol IoT serta pengujian alat sebagai berikut.

4.1 Deskripsi Alat

Alat pendeteksi kebakaran ini dirancang agar dapat menampilkan perubahan data dari sensor api, sensor asap dan sensor suhu secara *real time* melalui *smartphone*. Adapun blok diagram prinsip kerja alat dapat dilihat pada Gambar 4.1 berikut.



Gambar 4. 1 Blok Diagram Prinsip Kerja Sistem Pendeteksi Kebakaran

Sistem pengontrolan alat ini menggunakan Arduino Mega 2560 untuk mengolah data *input* berupa api, asap dan suhu kemudian dikirimkan ke NodeMCU ESP8266 sehingga diperoleh *output* berupa pembacaan data dari sensor api, sensor asap dan sensor suhu pada *smartphone*. Apabila terdeteksi adanya api, kadar asap >110 ppm dan suhu ruangan >38°C maka sistem secara otomatis akan mengaktifkan alarm, *exhaust*, motor servo untuk membuka pintu dan jendela serta LED sebagai

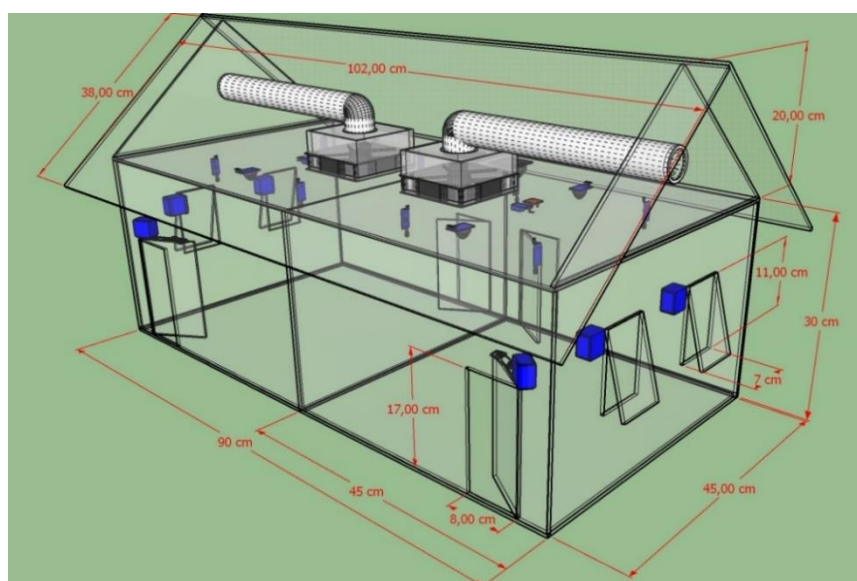
indikator pemadaman api dan akan dikirimkan notifikasi kebakaran ke *smartphone* sehingga pengguna dapat mengetahui informasi kebakaran dengan cepat. Adapun pengontrolan secara manual juga dapat dilakukan melalui *smartphone* ketika pengguna merasa khawatir akan terjadi kebakaran sehingga tindakan penanganan dapat dilakukan secara dini.

4.2 Perancangan dan Pembuatan *Hardware* Sistem *Monitoring* Data Sensor dan Kontrol Gedung

Perancangan dan pembuatan *hardware* ini meliputi dua bagian yaitu bagian mekanik dan elektrik. Berikut adalah tahap-tahap perancangan dan pembuatan *hardware*.

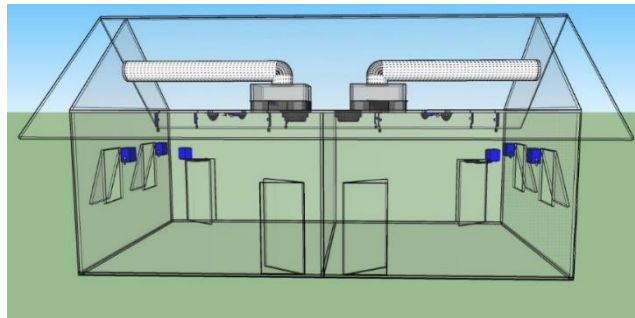
4.2.1 Perancangan *Hardware* secara Mekanik

Pada tahap perancangan *hardware* secara mekanik dibuat rancangan bentuk fisik (konstruksi) dari *prototype* pendeteksi kebakaran menggunakan aplikasi *sketch up*. Miniatur ini dibuat menggunakan bahan akrilik bening berbentuk persegi panjang yang terdiri dari dua ruangan dengan ukuran masing-masing ruangan adalah panjang 45 cm, lebar 45 cm dan tinggi 30 cm. Berikut adalah tampilan rancangan konstruksi *prototype* pendeteksi kebakaran.

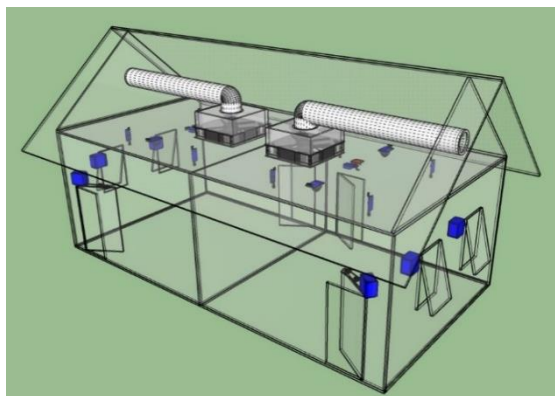


Gambar 4. 2 Rancangan Konstruksi *Prototype* Pendeteksi Kebakaran

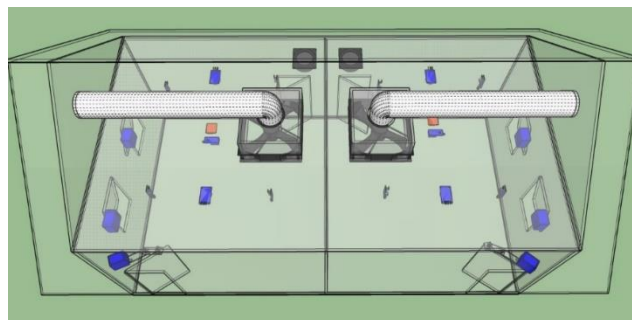
Setiap ruangan miniatur memiliki satu buah pintu utama, satu buah pintu darurat dan dua buah jendela. Kemudian pada tahap ini juga dibuat rancangan penempatan posisi *flame sensor*, sensor MQ-9, sensor DHT-11, Arduino Mega2560, NodeMCU ESP8266, alarm, *exhaust*, motor servo, dan LED pada masing-masing ruangan miniatur. Berikut ini adalah tampak depan, samping dan atas dari *prototype* pendeteksi kebakaran.



Gambar 4. 3 Tampak Depan Rancangan *Prototype* Pendeteksi Kebakaran



Gambar 4. 4 Tampak Samping Rancangan *Prototype* Pendeteksi Kebakaran

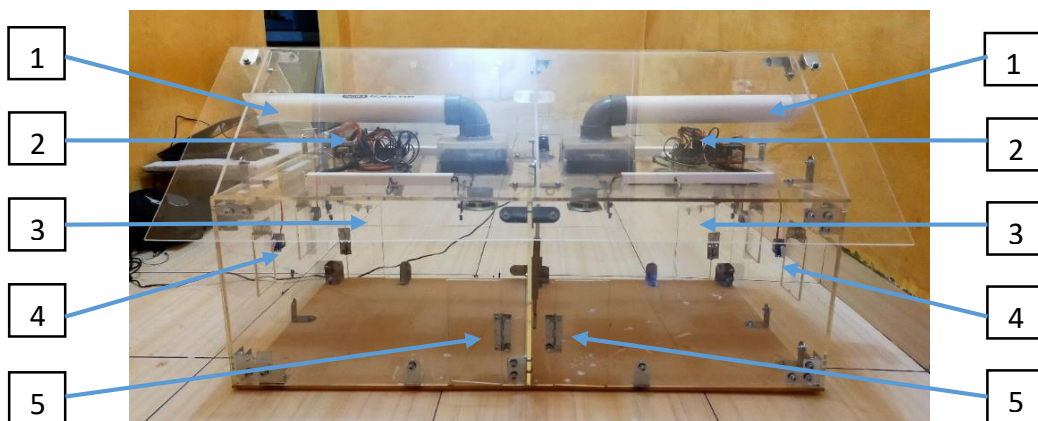


Gambar 4. 5 Tampak Atas Rancangan *Prototype* Pendeteksi Kebakaran

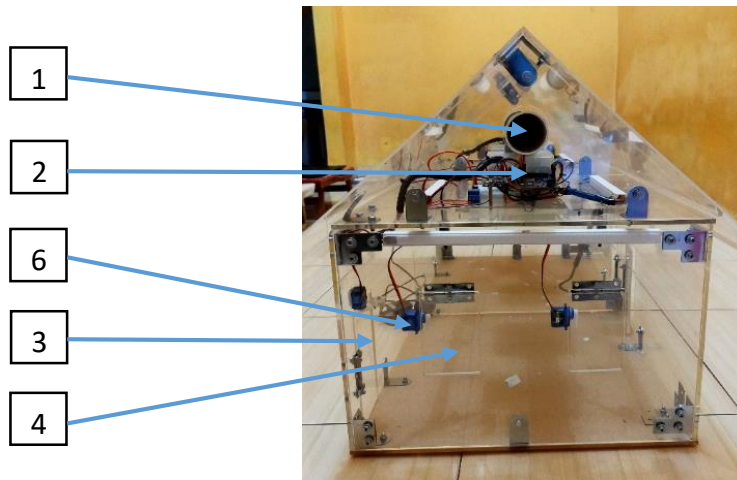
4.2.2 Pembuatan *Hardware* secara Mekanik

Kerangka konstruksi miniatur dibuat dengan cara merekatkan akrilik bening yang sudah dipotong sesuai ukuran membentuk persegi panjang. Dinding-dinding miniatur ini direkatkan menggunakan baut sehingga penyambungan setiap dinding lebih kokoh. Miniatur ini terdiri dari 2 buah ruangan dengan masing-masing ruangan memiliki ukuran panjang 45 cm, lebar 45 cm, dan tinggi 30 cm. Setiap ruangan memiliki satu buah pintu utama dan satu buah pintu darurat dengan ukuran tinggi pintu 17 cm dan lebar pintu 8 cm serta dua buah jendela dengan ukuran tinggi jendela 11 cm dan lebar jendela 7 cm.

Kemudian pada setiap ruangan miniatur dipasang 4 buah *flame sensor*, 2 buah sensor MQ-9, 1 buah sensor DHT-11, 1 buah Arduino Mega 2560, 1 buah NodeMCU ESP8266, 1 buah DF Mini Player, 1 buah *speaker*, 1 buah *exhaust*, 1 buah Modul Relay dan 1 buah LED pada bagian atas miniatur. Adapun motor servo yang digunakan sebanyak 3 buah yaitu 1 buah motor servo dipasang pada pintu darurat dan 1 buah pada masing-masing jendela yang berguna untuk membuka pintu darurat dan jendela. Berikut adalah hasil akhir dari pembuatan konstruksi *prototype* pendeteksi kebakaran.



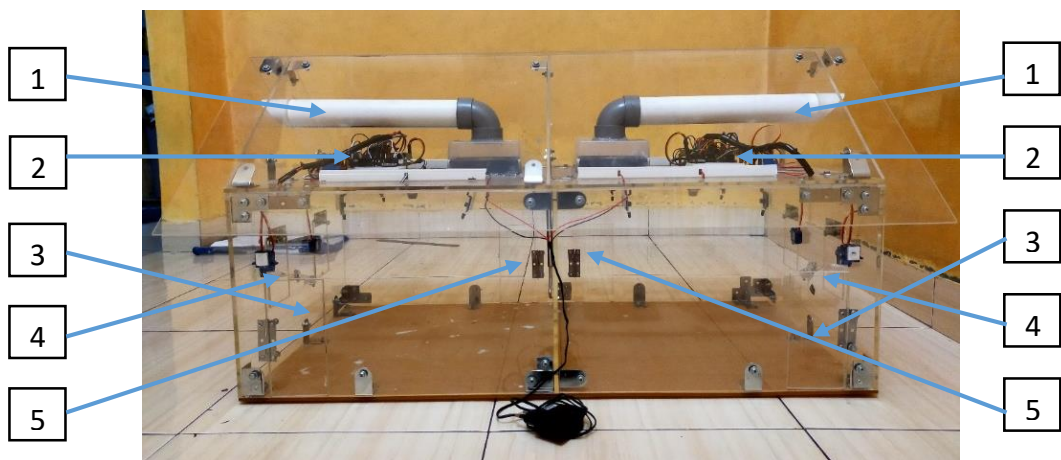
Gambar 4. 6 Tampak Depan Konstruksi Pendeteksi Kebakaran



Gambar 4. 7 Tampak Samping Konstruksi Pendeteksi Kebakaran



Gambar 4. 8 Tampak Atas Konstruksi Pendeteksi Kebakaran



Gambar 4. 9 Tampak Belakang Konstruksi Pendeteksi Kebakaran

Keterangan:

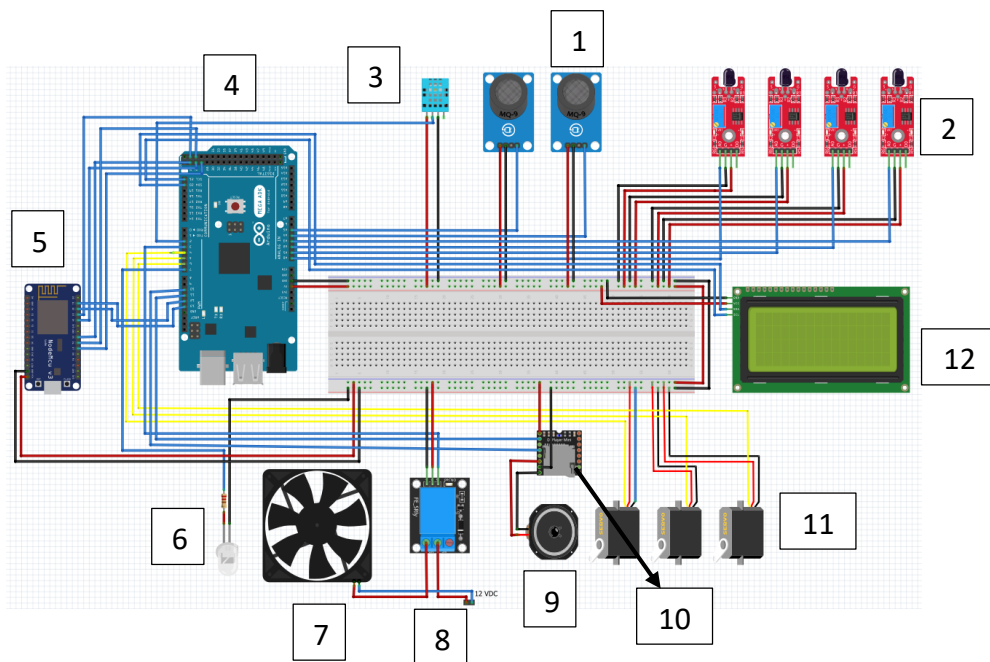
- | | | |
|-----------------------|------------------|----------------|
| 1. Pipa cerobong asap | 3. Pintu darurat | 5. Pintu utama |
| 2. Rangkaian kontrol | 4. Jendela | 6. Motor servo |

4.2.3 Perancangan dan Pembuatan *Hardware* Elektrik

Tahap ini merupakan tahap membuat rancangan sistem kontrol yang berfungsi untuk mengendalikan *actuator* pada sistem pendeteksi kebakaran. Pada rancangan ini terdapat tiga bagian penting yaitu data *input* dari sensor api, sensor asap dan sensor suhu. Data dari ketiga sensor ini kemudian akan diproses oleh Arduino Mega 2560 untuk mengontrol alarm, *exhaust*, motor servo, dan LED.

Adapun proses perancangan dan pembuatan sistem kontrol meliputi tahap-tahap sebagai berikut.

1. Membuat rancangan skema pengkabelan atau *wiring* untuk rangkaian kontrol menggunakan aplikasi Fritzing. Pembuatan rancangan ini bertujuan untuk menghindari kesalahan saat melakukan pemasangan *wiring* yang sesungguhnya. Adapun rancangan skema pengkabelan dari komponen yang digunakan dapat dilihat pada Gambar 4.10 berikut.

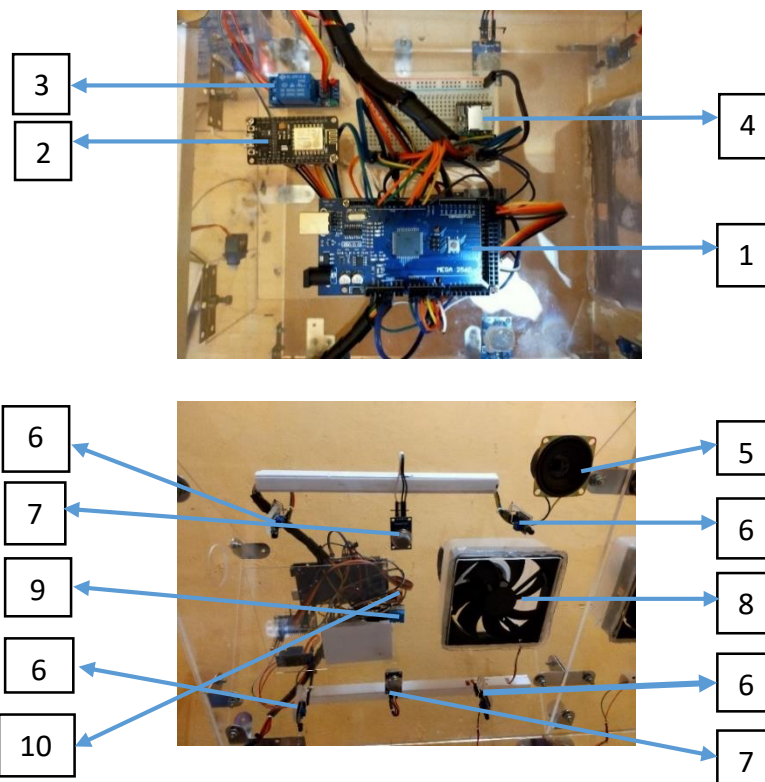


Gambar 4. 10 Skema Pengkabelan Sistem Kontrol

Keterangan:

- | | | | |
|------------------------|----------------------|-------------------|--------------------|
| 1. Sensor MQ-9 | 4. Arduino Mega 2560 | 7. <i>Exhaust</i> | 10. DF Mini Player |
| 2. <i>Flame Sensor</i> | 5. NodeMCU | 8. Relay | 11. Motor Servo |
| 3. Sensor DHT-11 | 6. LED | 9. <i>Speaker</i> | 12. LCD |

2. Memasang *wiring* pada komponen dengan cara menghubungkan *flame sensor*, sensor MQ-9, sensor DHT-11, DF Mini Player, *speaker*, *exhaust*, motor servo dan LED ke pin-pin Arduino Mega 2560. Berikut adalah hasil akhir dari pembuatan sistem kontrol yang sudah dipasang pada miniatur.



Gambar 4. 11 Rangkaian Sistem Kontrol Pendeteksi Kebakaran

Keterangan:

- | | |
|----------------------|------------------------|
| 1. Arduino Mega 2560 | 6. <i>Flame Sensor</i> |
| 2. NodeMCU ESP8266 | 7. Sensor MQ-9 |
| 3. Relay | 8. <i>Exhaust</i> |
| 4. DF Mini Player | 9. Sensor DHT-11 |
| 5. <i>Speaker</i> | 10. LED |

4.3 Pengujian Hardware Elektrik Sistem Monitoring Data Sensor dan Kontrol Gedung

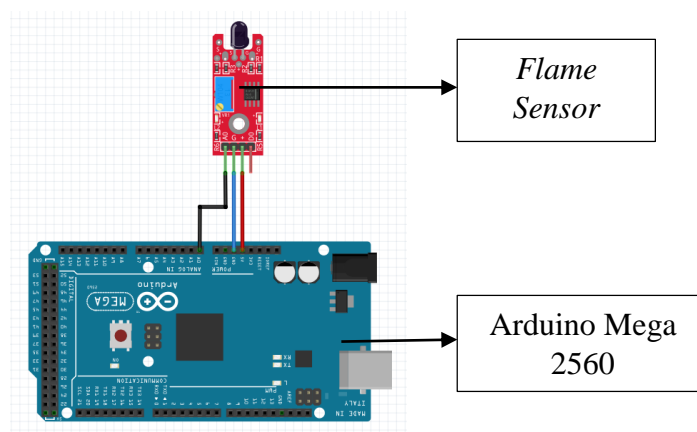
Pengujian hardware elektrik ini bertujuan untuk mengetahui kemampuan dan kesesuaian fungsi dari komponen-komponen yang digunakan dalam pembuatan proyek akhir ini.

4.3.1 Pengujian *Flame Sensor*

Tujuan dari pengujian ini adalah untuk mengetahui kemampuan deteksi dari *flame sensor* terhadap api. Setelah mengetahui batas kemampuan deteksi dari sensor ini maka akan lebih mudah dalam menentukan penempatan posisi dan jumlah *flame sensor* yang diperlukan dalam satu ruangan pada miniatur agar *flame sensor* dapat berfungsi secara maksimal dalam melakukan pendeteksian api. Adapun pengujian *flame sensor* ini meliputi tahap-tahap sebagai berikut:

4.3.1.1 Perancangan dan Pembuatan *Flame Sensor*

Pada tahap ini dilakukan perancangan dan pembuatan skema rangkaian dengan cara menghubungkan pin-pin pada *flame sensor* ke Arduino Mega 2560. Berikut adalah skema rangkaian dari *flame sensor* dengan Arduino Mega 2560 pada aplikasi Fritzing.

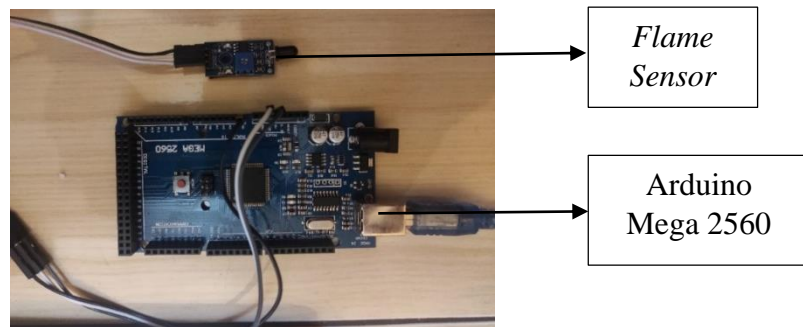


Gambar 4. 12 Skema Pengkabelan *Flame Sensor* dengan Arduino Mega 2560

Dilihat dari Gambar 4.12, penjelasan dari pin yang digunakan *flame sensor* ke Arduino mega 2560 adalah sebagai berikut.

- Pin Vcc *flame sensor* ke pin 5 V Arduino Mega 2560
- Pin A0 *flame sensor* ke pin A0 Arduino Mega 2560
- Pin Gnd *flame sensor* ke pin Gnd Arduino Mega 2560

Berikut adalah hasil akhir dari rangkaian *flame sensor* dengan Arduino Mega 2560.



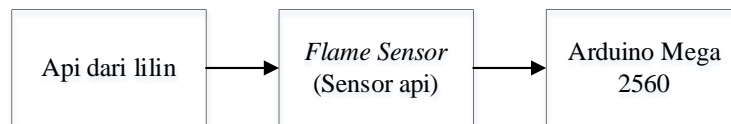
Gambar 4. 13 Rangkaian *Flame Sensor* dengan Arduino Mega 2560

Setelah dibuat rangkaian *flame sensor* dengan Arduino Mega 2560 maka selanjutnya membuat program pada Arduino IDE untuk menguji *flame sensor* yaitu sebagai berikut.

```
int sensorPin = A0; //pin yg di gunakan
int sensorValue = 0;
void setup() {
  Serial.begin(9600);
  pinMode( sensorPin, INPUT);
}
void loop() {
  Serial.println("pembacaan sensor");
  sensorValue = digitalRead(sensorPin);
  Serial.print("Sensor Api: ");
  Serial.println(sensorValue);
  delay(1000);
}
```

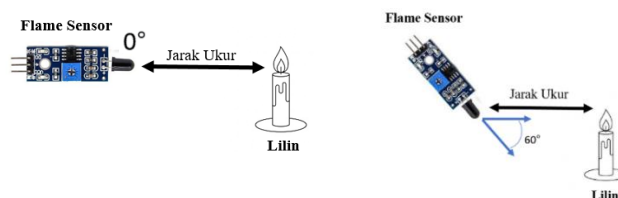
4.3.1.2 Prosedur Pengujian *Flame Sensor*

Pengujian ini menggunakan *input* berupa api dari satu buah lilin dan dilakukan pada ruangan tertutup dengan cara memindahkan sensor pada jarak dan sudut tertentu terhadap lilin. Kemudian hasil pengujian dari sensor api ini dicatat dalam Tabel 4.1. Perlu diketahui prinsip kerja *flame sensor* adalah mendeteksi api berdasarkan gelombang cahaya dari nyala api bukan panas sehingga pembacaan data sensor berupa logika 1 ketika tidak mendeteksi api dan logika 0 ketika mendeteksi api. Berikut adalah diagram blok pengujian *flame sensor* terhadap api pada jarak dan sudut tertentu.



Gambar 4. 14 Diagram Blok Pengujian *Flame Sensor*

Diagram pengujian *flame sensor* terhadap api dapat dilihat pada Gambar 4.15 berikut.



Gambar 4. 15 Diagram Pengujian *Flame Sensor* terhadap Api pada Sudut 0° dan 60°

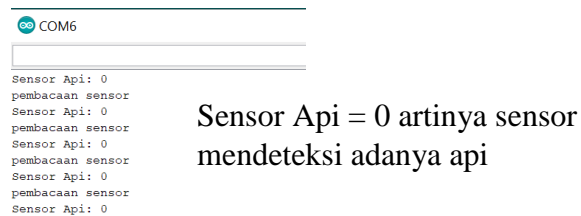
4.3.1.3 Hasil Pengujian *Flame Sensor*

- Pengujian *flame sensor* pada jarak 4 cm dengan sudut 0° terhadap api



Gambar 4. 16 Pengujian *Flame Sensor* pada Jarak 4 cm dan Sudut 0°

Berdasarkan gambar pengujian *flame sensor* pada jarak 4 cm dengan sudut deteksi 0° terhadap api menunjukkan bahwa *flame sensor* dalam kondisi ON atau mampu mendeteksi api yang ditandai dengan LED menyala berwarna hijau. Berikut adalah tampilan hasil pengujian pada serial monitor.

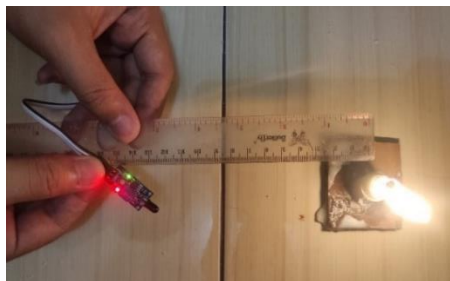


Gambar 4. 17 Hasil Pengujian *Flame Sensor* pada Jarak 4 cm dan Sudut 0°

Kesimpulan:

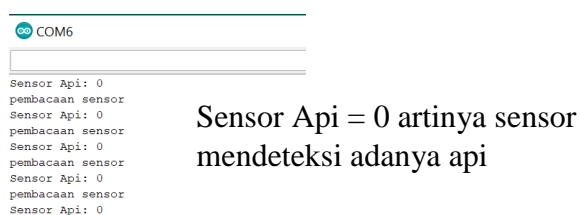
Dari hasil pengujian yang ditampilkan pada serial monitor diketahui bahwa *flame sensor* menampilkan *output* logika 0 yang artinya sensor ini mampu mendeteksi api pada jarak 4 cm dengan sudut deteksi sebesar 0°.

- Pengujian *flame sensor* pada jarak 12 cm dengan sudut 60° terhadap api



Gambar 4. 18 Pengujian *Flame Sensor* pada Jarak 12 cm dan Sudut 60°

Berdasarkan gambar pengujian *flame sensor* pada jarak 12 cm dengan sudut deteksi 60° terhadap api menunjukkan bahwa *flame sensor* dalam kondisi ON atau mampu mendeteksi api yang ditandai dengan LED menyala berwarna hijau. Berikut adalah tampilan hasil pengujian pada serial monitor.



Gambar 4. 19 Hasil Pengujian *Flame Sensor* pada Jarak 12 cm dan Sudut 60°

Kesimpulan:

Dari hasil pengujian yang ditampilkan pada serial monitor diketahui bahwa *flame sensor* menampilkan *output* logika 0 yang artinya sensor ini mampu mendeteksi api pada jarak 12 cm dengan sudut deteksi sebesar 60°.

- Pengujian *flame sensor* pada jarak 16 cm dengan sudut 70° terhadap api



Gambar 4. 20 Pengujian *Flame Sensor* pada Jarak 16 cm dan Sudut 70°

Berdasarkan gambar pengujian *flame sensor* pada jarak 16 cm dengan sudut deteksi 70° terhadap api menunjukkan bahwa *flame sensor* dalam kondisi OFF atau tidak mampu mendeteksi api yang ditandai dengan LED menyala berwarna merah. Berikut adalah tampilan hasil pengujian pada serial monitor.

```
COM6
pembacaan sensor
Sensor api= 1
pembacaan sensor
Sensor api= 1
pembacaan sensor
Sensor api= 1
pembacaan sensor
Sensor api= 1
pembacaan sensor
Sensor api= 1
```

Sensor Api = 1 artinya sensor tidak mendeteksi adanya api

Gambar 4. 21 Hasil Pengujian *Flame Sensor* pada Jarak 16 cm dan Sudut 70°

Kesimpulan:

Dari hasil pengujian yang ditampilkan pada serial monitor diketahui bahwa *flame sensor* menampilkan *output* logika 1 yang artinya sensor ini tidak dapat mendeteksi api pada jarak 16 cm dengan sudut deteksi sebesar 70°.

Berikut adalah tabel hasil pengujian *flame sensor* terhadap api pada beberapa jarak dan sudut tertentu.

Tabel 4. 1 Hasil Pengujian *Flame Sensor* terhadap Api

<i>Input</i>	Jarak sensor terhadap api	Sudut deteksi sensor terhadap api	<i>Output</i>
			Pembacaan Api pada <i>Serial Monitor</i>
Api dari lilin	4 cm	0°	0
		30°	0
		60°	0
		70°	1
		90°	1
Api dari lilin	8 cm	0°	0
		30°	0
		60°	0
		70°	1
Api dari lilin	12 cm	0°	0
		30°	0
		60°	0
		70°	1
Api dari lilin	16 cm	0°	0
		30°	0
		60°	0
		70°	1
Api dari lilin	20 cm	0°	0
		30°	1
	30 cm	0°	0
		30°	1
	35 cm	0°	1

Keterangan:

1. Logika 0 menyatakan bahwa sensor api mendeteksi adanya api sedangkan logika 1 menyatakan bahwa sensor api tidak mendeteksi adanya api.

Kesimpulan:

Dari tabel hasil pengujian dapat disimpulkan bahwa *flame sensor* ini dapat berfungsi dengan baik sehingga dapat digunakan dalam pembuatan proyek akhir sebagai pendeteksi api. Akan tetapi sensor ini tidak bisa digunakan untuk sistem

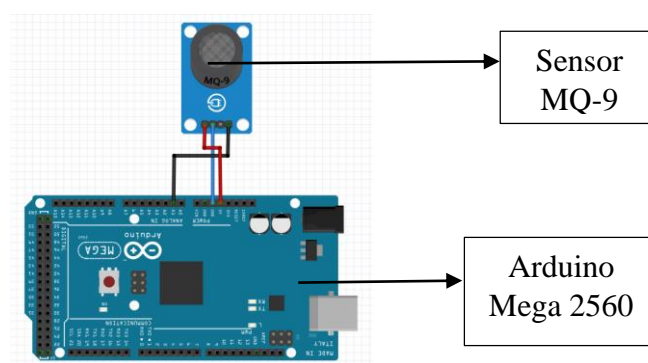
pendeteksi kebakaran pada kondisi ruangan yang sebenarnya dikarenakan sensor ini memiliki kemampuan deteksi sensor pada jarak yang terbatas yaitu hanya dapat mendeteksi api sampai jarak maksimum 16 cm dengan rentang sudut 0 – 60°. Sensor ini masih bisa mendeteksi api sampai jarak maksimum 30 cm namun hanya dengan sudut deteksi sebesar 0°.

4.3.2 Pengujian Sensor MQ-9

Tujuan dari pengujian ini adalah untuk mengetahui kemampuan dari sensor MQ-9 dalam mendeteksi kadar asap. Setelah mengetahui kemampuan deteksi dari sensor MQ-9 ini maka akan lebih mudah dalam menentukan penempatan posisi dan jumlah sensor MQ-9 yang dibutuhkan dalam satu ruangan miniatur sehingga sensor MQ-9 dapat berfungsi secara optimal dalam melakukan pendeteksian asap. Adapun pengujian sensor MQ-9 meliputi tahap-tahap sebagai berikut.

4.3.2.1 Perancangan dan Pembuatan Sensor MQ-9

Pada tahap ini dilakukan perancangan dan pembuatan skema rangkaian dengan cara menghubungkan pin-pin dari sensor MQ-9 ke Arduino Mega 2560. Berikut adalah skema rangkaian dari sensor MQ-9 dengan Arduino Mega 2560 pada aplikasi Fritzing.

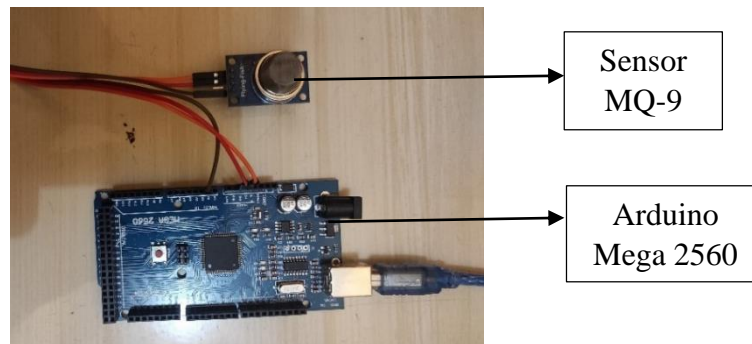


Gambar 4. 22 Skema Pengkabelan Sensor MQ-9 dengan Arduino Mega 2560

Dilihat dari Gambar 4.22, penjelasan dari pin yang digunakan sensor MQ-9 ke Arduino mega 2560 adalah sebagai berikut.

- Pin Vcc sensor MQ-9 ke pin 5 V Arduino Mega 2560
- Pin A0 sensor MQ-9 ke pin A1 Arduino Mega 2560
- Pin Gnd sensor MQ-9 ke pin Gnd Arduino Mega 2560

Berikut adalah hasil akhir dari rangkaian sensor MQ-9 dengan Arduino Mega 2560.



Gambar 4. 23 Rangkaian Sensor MQ-9 dengan Arduino Mega 2560

Setelah dibuat rangkaian sensor MQ-9 dengan Arduino Mega 2560 maka selanjutnya membuat program pada Arduino IDE untuk menguji sensor MQ-9 yaitu sebagai berikut.

```
#include<SoftwareSerial.h>

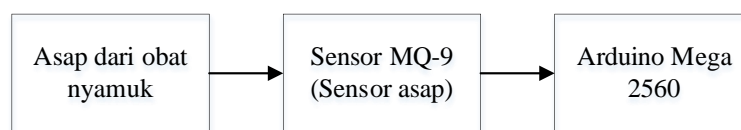
int asapPin = A1;
int asapValue = 0;

void setup()
{
  Serial.begin(9600);
}

void loop() {
  asapValue = analogRead(asapPin);
  Serial.print("Asap = ");
  Serial.println(asapValue-105);
  delay (1000);
}
```

4.3.2.2 Prosedur Pengujian Sensor MQ-9

Pengujian ini menggunakan *input* berupa asap dari obat nyamuk dan dilakukan pada ruangan tertutup. Pada pengujian ini digunakan dua buah obat nyamuk dan diletakkan di bawah sensor asap kemudian dibiarkan selama 1 menit untuk mengetahui kadar asap yang dideteksi. Selanjutnya hasil pengujian kadar asap selama 1 menit ini dicatat dalam Tabel 4.2. Berikut adalah blok diagram pengujian sensor MQ-9 terhadap sumber asap (obat nyamuk).



Gambar 4. 24 Blok Diagram Pengujian Sensor MQ-9

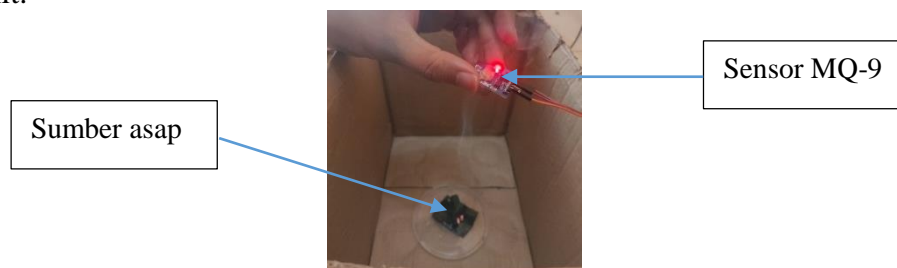
Diagram pengujian sensor MQ-9 terhadap asap obat nyamuk dapat dilihat pada Gambar 4.25 berikut.



Gambar 4. 25 Diagram Pengujian Sensor MQ-9 terhadap Asap

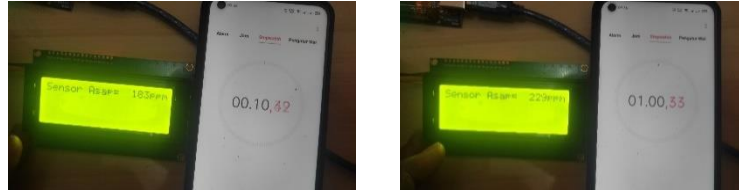
4.3.2.3 Hasil Pengujian Sensor MQ-9

Berikut adalah gambar pengujian Sensor MQ-9 terhadap sumber asap dalam waktu 1 menit.



Gambar 4. 26 Pengujian Sensor MQ-9 terhadap Asap selama 1 Menit

Pengujian sensor MQ-9 terhadap asap obat nyamuk selama 1 menit diperoleh hasil pada LCD sebagai berikut.



Gambar 4. 27 Hasil Pengujian Sensor MQ-9 terhadap Asap selama 1 Menit

Tabel 4. 2 Hasil Pengujian Sensor MQ-9 terhadap Asap

<i>Input</i>	<i>Output</i>	
	Pembacaan kadar asap pada Serial Monitor	
	Waktu (s)	Kadar asap (ppm)
Asap dari obat nyamuk	10	183
	20	221
	30	234
	40	265
	50	197
	60	229

Kesimpulan:

Dari hasil pengujian diketahui bahwa kadar asap yang dideteksi oleh sensor MQ-9 berubah-ubah seiring dengan perubahan waktu. Hal ini disebabkan kuantitas asap yang dihasilkan oleh obat nyamuk berbeda-beda yang menyebabkan nilai dari kadar asap bisa saja meningkat atau menurun dalam satu satuan waktu. Setiap perubahan nilai kadar asap per satuan waktu bisa dideteksi oleh sensor MQ-9 sehingga dapat disimpulkan bahwa sensor ini dapat berfungsi dengan baik dan dapat digunakan dalam pembuatan proyek akhir sebagai pendeteksi asap.

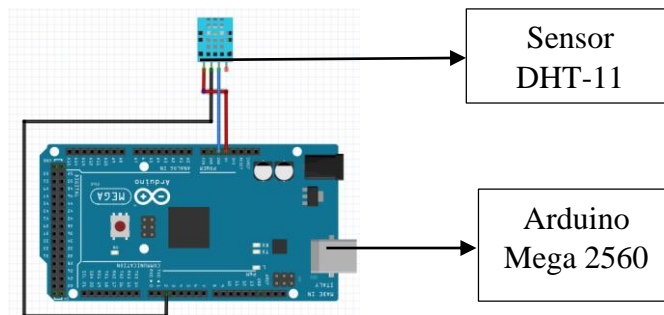
4.3.3 Pengujian Sensor DHT-11

Tujuan dari pengujian ini adalah untuk mengetahui kemampuan dan keakurasian deteksi dari sensor DHT-11 terhadap suhu. Setelah mengetahui batas kemampuan deteksi dari sensor ini maka akan lebih mudah dalam menentukan penempatan posisi dan jumlah sensor DHT-11 yang diperlukan dalam satu ruangan

pada miniatur agar sensor ini dapat berfungsi secara maksimal dalam melakukan pendeteksian suhu. Adapun pengujian sensor MQ-9 ini meliputi tahap-tahap sebagai berikut:

4.3.3.1 Perancangan dan Pembuatan Sensor DHT-11

Pada tahap ini dilakukan perancangan dan pembuatan skema rangkaian dengan cara menghubungkan pin-pin pada sensor DHT-11 ke Arduino Mega 2560. Berikut adalah skema rangkaian dari sensor DHT-11 dengan Arduino Mega2560 pada aplikasi Fritzing.

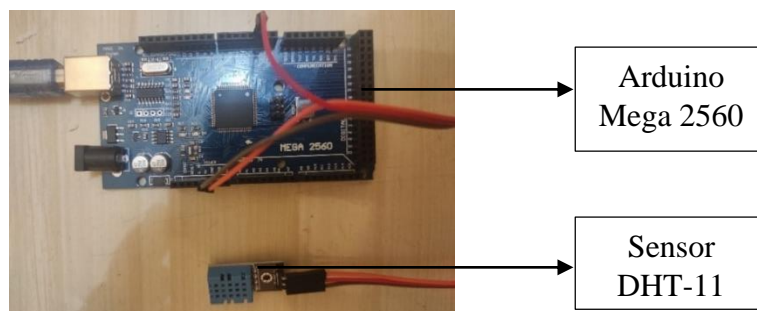


Gambar 4. 28 Skema Pengkabelan Sensor DHT-11 dengan Arduino Mega 2560

Dilihat dari Gambar 4.28 penjelasan dari pin yang digunakan sensor DHT-11 ke Arduino mega 2560 adalah sebagai berikut.

- Pin (+) sensor DHT-11 ke pin 5 V Arduino Mega 2560
- Pin Out sensor DHT-11 ke pin 2 Digital Arduino Mega 2560
- Pin (-) sensor DHT-11 ke pin Gnd Arduino Mega 2560

Berikut adalah hasil akhir dari rangkaian sensor DHT-11 dengan Arduino Mega 2560.



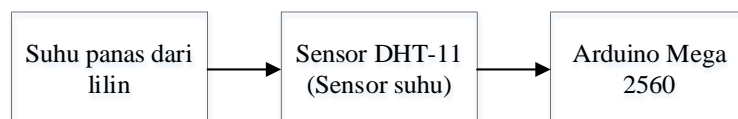
Gambar 4. 29 Rangkaian Sensor DHT-11 dengan Arduino Mega 2560

Setelah dibuat rangkaian sensor DHT-11 dengan Arduino Mega 2560 maka selanjutnya membuat program pada Arduino IDE untuk menguji sensor DHT-11 yaitu sebagai berikut.

```
#include <DHT.h>
DHT dht(2, DHT11);
void setup() {
  Serial.begin(9600);
  dht.begin();
}
void loop() {
  float suhu = dht.readTemperature();
  Serial.print(" ");
  Serial.print("suhu: ");
  Serial.println(suhu);
  delay(1000);
}
```

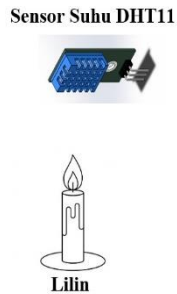
4.3.3.2 Prosedur Pengujian Sensor DHT-11

Pengujian ini menggunakan *input* berupa suhu panas dari lilin dan dilakukan pada ruangan tertutup. Pada pengujian ini digunakan satu buah lilin dan diletakkan di bawah sensor suhu dan termometer kemudian data hasil pengujian dicatat pada Tabel 4.3. Selanjutnya hasil pengukuran suhu oleh sensor DHT-11 dibandingkan dengan hasil pengukuran suhu menggunakan termometer untuk mendapatkan nilai keakurasian deteksi dari sensor DHT-11. Berikut adalah blok diagram pengujian sensor DHT-11 terhadap sumber suhu panas dari lilin.



Gambar 4. 30 Diagram Blok Pengujian Sensor DHT-11

Diagram pengujian sensor DHT-11 terhadap sumber suhu panas dari lilin dapat dilihat pada Gambar 4.31 berikut.



Gambar 4. 31 Diagram Pengukuran Sensor DHT-11 terhadap Suhu Panas

4.3.3.3 Hasil Pengujian Sensor DHT-11

- Pengujian sensor DHT-11 dan termometer pada jarak 10 cm terhadap lilin



Gambar 4. 32 Pengujian Sensor DHT-11 pada Jarak 10 cm terhadap Lilin

Pengujian sensor DHT-11 pada jarak 10 cm terhadap sumber panas (lilin) diperoleh hasil pada serial monitor sebagai berikut.

```
COM6
suhu: 34.00
suhu: 34.00
suhu: 35.00
suhu: 35.00
suhu: 36.00
suhu: 36.00
suhu: 37.00
suhu: 37.00
suhu: 37.00
suhu: 37.00
suhu: 37.00
suhu: 37.00
suhu: 37.00
suhu: 37.00
suhu: 37.00
suhu: 38.00
```

Gambar 4. 33 Hasil Pengujian Sensor DHT-11 pada Jarak 10 cm terhadap Lilin

- Pengujian sensor DHT-11 dan termometer pada jarak 15 cm terhadap lilin



Gambar 4. 34 Pengujian Sensor DHT-11 pada Jarak 15 cm terhadap Lilin

Pengujian sensor DHT-11 pada jarak 15 cm terhadap sumber panas (lilin) diperoleh hasil pada serial monitor sebagai berikut.

```

COM6
suhu: 33.00
suhu: 33.00
suhu: 34.00
suhu: 34.00
suhu: 34.00
suhu: 34.00
suhu: 34.00
suhu: 34.00
suhu: 34.00
suhu: 34.00
suhu: 34.00
suhu: 34.00
suhu: 34.00
suhu: 34.00
suhu: 35.00
suhu: 35.00
suhu: 35.00

```

Gambar 4. 35 Hasil Pengujian Sensor DHT-11 pada Jarak 15 cm terhadap Lilin

Berikut adalah tabel hasil pengujian sensor DHT-11 dan termometer terhadap suhu panas pada beberapa posisi.

Tabel 4. 3 Hasil Pengujian Sensor DHT-11 terhadap Suhu Panas

<i>Input</i>	Jarak sensor terhadap lilin (cm)	Suhu terukur oleh sensor DHT-11 (°C)	Suhu terukur oleh termometer (°C)	% Error	Rata-rata error (%)
Suhu panas dari lilin	10	38	38,7	1,81	1,21
	15	35	35,5	1,41	
	20	34	34,6	1,73	
	25	34	34,4	1,16	
	30	34	34,1	0,29	
	35	34	34,3	0,87	

Keterangan:

1. Suhu normal ruangan saat pengujian adalah 32°C.

Rumus *persentase error*, dimana:

$$\text{Persentase error} = \left| \frac{\text{Alat Ukur} - \text{Sensor}}{\text{Alat Ukur}} \right| \times 100\% \dots\dots\dots(4.3)$$

Alat Ukur = Hasil pengukuran pada alat ukur suhu

Sensor = Hasil pengukuran pada sensor DHT-11

Berikut ini perhitungan secara teori:

Perhitungan *persentase error* pada tabel 4.3 pengujian sensor DHT-11 dengan alat ukur suhu.

- *Persentase error* pengukuran ke-1 = $\left| \frac{38,7-38}{38,7} \right| \times 100\% = 1,81\%$
- *Persentase error* pengukuran ke-4 = $\left| \frac{34,4-34}{34,4} \right| \times 100\% = 1,16\%$
- *Persentase error* pengukuran ke-5 = $\left| \frac{34,1-34}{34,1} \right| \times 100\% = 0,29\%$

Kesimpulan:

Dari tabel hasil pengujian sensor DHT-11 dan termometer terhadap pengukuran suhu, diketahui bahwa *persentase rata-rata error* dari hasil pengukuran sebesar 1,21%. Berdasarkan nilai rata-rata *error* tersebut menunjukkan bahwa sensor DHT-11 mempunyai kemampuan deteksi terhadap suhu yang cukup baik sehingga dapat digunakan untuk mendeteksi suhu pada keperluan proyek akhir ini.

4.4 Perancangan dan Pembuatan *Software*

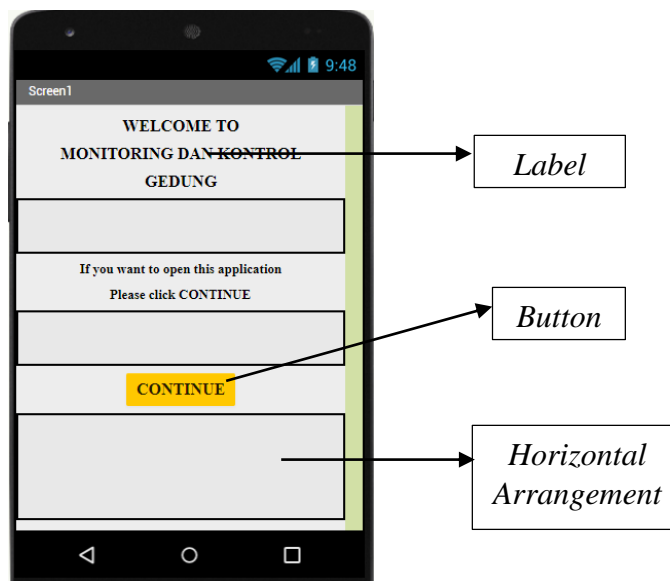
Pada perancangan dan pembuatan *software* akan dibuat rancangan *interface* aplikasi pada *smartphone*, perancangan dan pembuatan *database*, komunikasi serial pada *software* MIT App Inventor, dan koneksi MIT App Inventor dengan *database*.

4.4.1 Perancangan dan Pembuatan *Interface* pada *Smartphone*

Perancangan dan pembuatan *interface* ini bertujuan untuk memantau perubahan data sensor dan mengontrol alarm, *exhaust*, pintu dan jendela serta LED sebagai indikator pemadaman api oleh pengguna melalui *smartphone*. *Interface* menggunakan aplikasi MIT App Inventor ini dirancang dalam 5 *layout* yaitu pertama, *layout* awal yang merupakan tampilan pertama saat aplikasi dibuka. Pada *layout* ini terdapat *button* CONTINUE untuk melanjutkan ke *layout* berikutnya. Kedua, yaitu *layout* untuk *login* ke menu utama. Sebelum melakukan *login*, pengguna harus mendaftarkan *username* dan *password* terlebih dahulu pada *layout* *sign up*. Setelah berhasil melakukan *sign up* maka selanjutnya pengguna bisa melakukan *login* untuk masuk ke *layout* utama yaitu menu *monitoring* dan kontrol yang dapat memantau data sensor dan mengontrol *actuator* dari dua ruangan. Berikut ini *interface* yang dibuat menggunakan MIT App Inventor:

1. *Home*

Menu *home* merupakan tampilan awal ketika aplikasi baru dibuka. Jika ingin melanjutkan ke halaman berikutnya maka pengguna dapat menekan tombol CONTINUE pada menu tersebut. Adapun tampilan menu *home* pada *smartphone* yang dibuat menggunakan *software* MIT App Inventor dapat dilihat pada Gambar 4.55 sebagai berikut.



Gambar 4. 36 Desain Menu *Home* pada *Software* MIT App Inventor

Berikut keterangan dari *interface* menu *home* pada MIT App Inventor:

- *Horizontal Arrangement*, digunakan untuk menyusun komponen pada desain aplikasi secara horizontal. Namun pada kasus ini, komponen *horizontal arrangement* digunakan sebagai spasi penulisan *label* pada layar aplikasi sehingga antara *label* satu dengan *label* lain terdapat jarak penulisan tertentu sehingga tampilan menjadi lebih rapi.
- *Label*, digunakan untuk menampilkan teks. Komponen ini dapat digunakan untuk menuliskan apa saja yang ingin kita tampilkan pada desain aplikasi.
- *Button*, digunakan untuk mendeteksi ketukan ketika pengguna menekan tombol dan menjalankan perintah tertentu. Pada menu *home* ini, ketika *button* CONTINUE ditekan maka akan dilakukan perintah untuk melanjutkan ke halaman berikutnya.

Adapun tampilan layout *home* yang muncul pada layar *smartphone* ketika aplikasi MIT App Inventor dibuka adalah sebagai berikut.

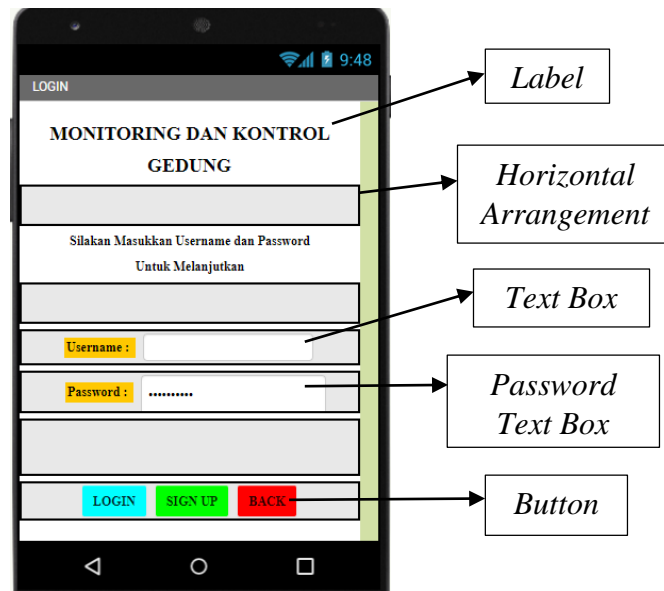


Gambar 4. 37 Tampilan Menu *Home* pada *Smartphone*

2. Menu *Login*

Menu *login* merupakan menu yang digunakan untuk masuk ke menu utama yaitu *monitoring* dan kontrol. Pada menu *login*, pengguna harus memasukkan

username dan *password* untuk mendapatkan hak akses ke menu utama. Akan tetapi, jika sebelumnya pengguna belum pernah melakukan *sign up* maka *login* tidak bisa dilakukan.



Gambar 4. 38 Desain Menu *Login* pada *Software MIT App Inventor*

Berikut keterangan dari *interface* menu *Login* pada MIT App Inventor:

- *Horizontal Arrangement*, digunakan untuk menyusun komponen pada desain aplikasi secara horizontal. Dan pada kasus ini, komponen *horizontal arrangement* juga digunakan sebagai spasi penulisan *label* pada layar aplikasi sehingga antara *label* satu dengan *label* lain terdapat jarak penulisan tertentu sehingga tampilan menjadi lebih rapi.
- *Label*, digunakan untuk menampilkan teks. Komponen ini dapat digunakan untuk menuliskan apa saja yang ingin kita tampilkan pada desain aplikasi.
- *Text Box*, digunakan untuk memasukkan username dari pengguna.
- *Password Text Box*, digunakan untuk menyembunyikan *password* yang dimasukkan pengguna agar terjamin kerahasiaannya.
- *Button*, digunakan untuk mendeteksi ketukan ketika pengguna menekan tombol dan menjalankan perintah tertentu. Pada menu ini terdapat tiga buah *button* yaitu *button* LOGIN untuk melanjutkan ke *layout monitoring* dan kontrol, *button* SIGN UP digunakan untuk masuk ke halaman *sign up* jika

belum mendaftarkan *username* dan *password*, serta *button* BACK untuk kembali ke menu *home*.

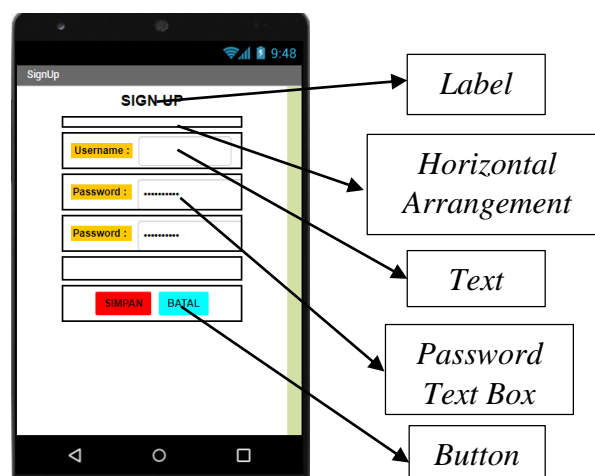
Adapun tampilan *layout* Login yang muncul pada layar *smartphone* adalah sebagai berikut.



Gambar 4. 39 Tampilan Menu *Login* pada *Smartphone*

3. Menu *Sign Up*

Menu *Sign Up* merupakan menu yang berfungsi untuk mendaftarkan *username* dan *password*. Setelah *Sign Up* berhasil dilakukan maka pengguna sudah bisa melakukan *login* menggunakan *username* dan *password* yang sudah didaftarkan.



Gambar 4. 40 Desain Menu *Sign Up* pada *Software* MIT App Inventor

Berikut keterangan dari *interface* menu *sign up* pada MIT App Inventor:

- *Horizontal Arrangement*, digunakan untuk menyusun komponen pada desain aplikasi secara horizontal. Dan pada kasus ini, komponen *horizontal arrangement* juga digunakan sebagai spasi penulisan *label* pada layar aplikasi sehingga antara *label* satu dengan *label* lain terdapat jarak penulisan tertentu sehingga tampilan menjadi lebih rapi.
- *Label*, digunakan untuk menampilkan teks. Komponen ini dapat digunakan untuk menuliskan apa saja yang ingin kita tampilkan pada desain aplikasi.
- *Text Box*, digunakan untuk memasukkan username dari pengguna.
- *Password Text Box*, digunakan untuk menyembunyikan *password* yang dimasukkan pengguna agar terjamin kerahasiaannya.
- *Button*, digunakan untuk mendeteksi ketukan ketika pengguna menekan tombol dan menjalankan perintah tertentu. Pada menu ini terdapat dua buah *button* yaitu *button* SIMPAN untuk menyimpan data *username* dan *password* yang sudah didaftarkan ke *firebase* dan *button* BATAL yang digunakan untuk membatalkan *username* dan *password* yang didaftarkan dan kembali ke menu *login*.

Adapun tampilan menu *sign up* yang muncul pada layar *smartphone* adalah sebagai berikut.

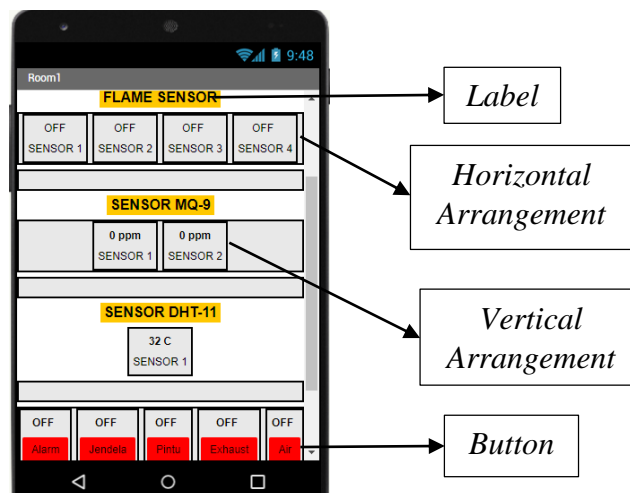


The image shows a mobile application interface for a sign-up screen. At the top, there is a status bar with the time '17:46' and various icons. Below the status bar is a title bar with the text 'SignUp'. The main content area is titled 'SIGN UP' and contains three input fields: 'Username', 'Password', and 'Password'. Below the input fields are two buttons: 'SIMPAN' (red) and 'BATAL' (cyan).

Gambar 4. 41 Tampilan Menu *Sign Up* pada *Smartphone*

4. Menu Utama

Menu utama berisi tampilan untuk *monitoring* dan kontrol. Pada menu *monitoring* terdapat informasi berupa data-data dari sensor api, sensor asap dan sensor suhu sehingga pengguna bisa memantau perubahan data dari ketiga sensor tersebut. Selain *monitoring*, pada menu ini juga terdapat beberapa tombol yang dapat digunakan untuk mengontrol tindakan penanganan kebakaran secara manual oleh *operator*. Pada menu utama terdiri dari 2 *room* yaitu *Room1* untuk menampilkan data sensor dan mengontrol *actuator* pada ruangan 1 dan *Room2* untuk menampilkan data sensor dan mengontrol *actuator* pada ruangan 2. Berikut tampilan Menu *Monitoring* dan Kontrol pada MIT App Inventor:



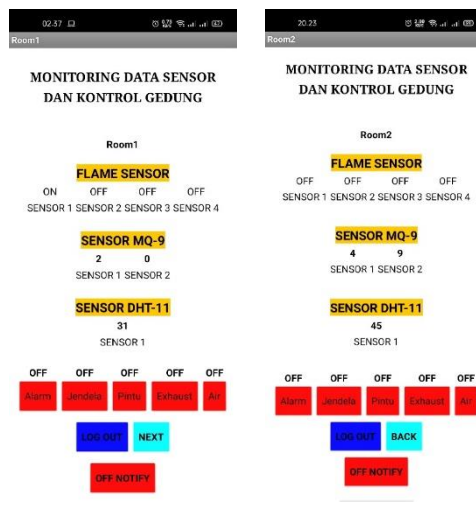
Gambar 4.42 Desain Tampilan Menu *Monitoring* dan Kontrol pada MIT App Inventor

Berikut keterangan dari *interface* menu utama pada MIT App Inventor:

- *Horizontal Arrangement*, digunakan untuk menyusun komponen pada desain aplikasi secara horizontal. Dan pada kasus ini, komponen *horizontal arrangement* juga digunakan sebagai spasi penulisan *label* pada layar aplikasi sehingga antara *label* satu dengan *label* lain terdapat jarak penulisan tertentu sehingga tampilan menjadi lebih rapi.
- *Vertical Arrangement*, digunakan untuk menyusun komponen pada desain aplikasi secara vertikal.

- *Label*, digunakan untuk menampilkan teks. Komponen ini dapat digunakan untuk menuliskan apa saja yang ingin kita tampilkan pada desain aplikasi.
- *Button*, digunakan untuk mendeteksi ketukan ketika pengguna menekan tombol dan menjalankan perintah tertentu. Pada *Room1* terdapat tiga buah *button* yaitu *button* LOGOUT yang digunakan untuk keluar dari aplikasi dan kembali ke menu *home*, *button* NEXT yang digunakan untuk melanjutkan ke *Room2* serta *button* STOP NOTIFY untuk menghentikan notifikasi pada *smartphone*. Adapun pada *Room2* ini juga terdapat tiga buah *button* yaitu *button* LOGOUT yang digunakan untuk keluar dari aplikasi dan kembali ke menu *home*, *button* BACK yang digunakan untuk kembali ke *Room1* serta *button* STOP NOTIFY untuk menghentikan notifikasi ke *smartphone*.

Adapun tampilan menu utama yang muncul pada layar *smartphone* adalah sebagai berikut.



Gambar 4. 43 Tampilan Menu *Monitoring* dan *Kontrol* pada *Smartphone*

4.4.2 Perancangan dan Pembuatan *Database*

Database adalah kumpulan data yang saling berhubungan, dapat disimpan dan diakses untuk berbagai kepentingan. Pada proyek akhir ini *platform* yang digunakan untuk membuat *database* adalah *firebase*. *Firestore* merupakan sebuah wadah yang berguna untuk menampung *database* atau basis data. Berikut adalah *database* yang dibuat dan disimpan pada *firebase*:

https://diniek0-ta-default-rtdb.firebaseio.com/

Your security rules are defined as public, so anyone can steal, modify, or delete data in your database [Learn more](#) [Dismiss](#)

```

diniek0-ta-default-rtdb
├── Api1: "1"
├── Api2: "1"
├── Api3: "1"
├── Api4: "1"
├── Api5: "1"
├── Api6: "1"
├── Api7: "1"
├── Api8: "1"
├── Asap1: "6"
├── Asap2: "1"
├── Asap3: "-5"
├── Asap4: "5"
├── K: "0"
├── Q: "0"
├── R: "0"
├── S: "0"
├── Suhu1: "32"
├── Suhu2: "32"
├── X: "0"
├── aa: "0"
├── al123: "\"a10011\""
├── alarmhp1: "0"
├── alarmhp2: "0"
├── bb: "0"
├── cc: "0"
├── dd: "0"
├── dinian: "\"123123\""
├── dinianjeli: "\"123567\""
├── dinianjelina: "\"121000\""
├── dinijiett: "\"123123\""
├── ee: "0"
├── ekofirlianto: "\"110500\""
├── fanhp1: "0"
├── fanhp2: "0"
├── ff: "0"
├── jendelahp1: "0"
├── jendelahp2: "0"
├── ledhp1: "0"
├── ledhp2: "0"
├── notif_off: "0"
├── notif_off2: "0"
├── pintuhp1: "0"
└── pintuhp2: "0"

```

Inisialisasi untuk menampilkan data sensor dari Arduino ke aplikasi MIT App Inventor

Inisialisasi untuk menampilkan data ketika syarat atau kondisi pada program Arduino terpenuhi

Inisialisasi untuk menampilkan data ketika syarat atau kondisi pada program Arduino terpenuhi

Data yang diinput dari MIT App Inventor disimpan di *firebase*

Inisialisasi untuk mentransfer data dari MIT App Inventor ke Arduino

Gambar 4.44 Tampilan Data yang disimpan pada *Firestore*

4.4.3 Komunikasi Serial pada *Software* MIT App Inventor

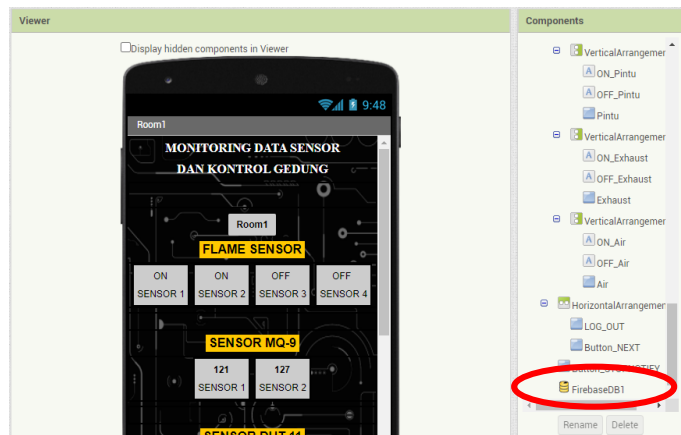
Interface merupakan tampilan visual sebuah aplikasi yang bertujuan untuk menghubungkan sistem dengan pengguna atau *user*. *Interface* yang dibuat pada aplikasi MIT App Inventor ini bertujuan agar pengguna dapat menampilkan perubahan data sensor dan mengontrol *actuator* secara manual melalui *smartphone*. Agar *interface* dapat berfungsi sebagaimana mestinya maka perlu dibuat komunikasi serial antara Arduino dengan aplikasi MIT App Inventor dengan cara membuat pemrograman terlebih dahulu untuk mentransfer data dari Arduino ke NodeMCU. Kemudian data dari NodeMCU ini akan diolah dan dikirim ke *firebase*. Selanjutnya data yang disimpan di *firebase* akan dipanggil melalui program yang telah dibuat pada aplikasi MIT App Inventor sehingga program yang ada di Arduino dapat terkoneksi dengan aplikasi MIT App Inventor. Program untuk mentransfer data dari Arduino ke NodeMCU dapat dilihat pada Lampiran 2 dan program untuk mentransfer data dari NodeMCU ke *firebase* dapat dilihat pada Lampiran 3.

4.4.4 Koneksi MIT App Inventor dengan *Database*

Data yang dibuat pada *firebase* akan ditampilkan di dalam aplikasi MIT App Inventor. Untuk mengolah *database* agar terhubung dengan MIT App Inventor maka perlu dibuat program berupa pemanggilan data yang tersimpan di dalam *firebase* ke aplikasi MIT App Inventor. Kemudian juga dilakukan input data dari sensor yang ada di dalam masing-masing ruangan *prototype* ke dalam *firebase*.

Cara membuat koneksi dari MIT App Inventor ke *firebase* adalah sebagai berikut.

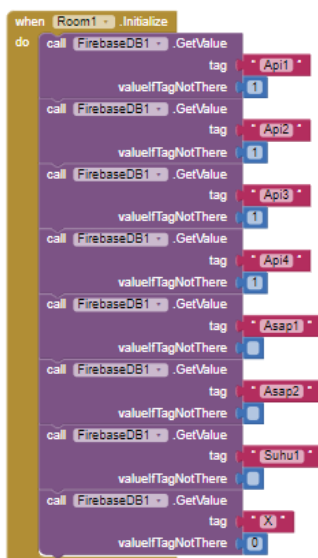
1. Membuat *database* pada *firebase* terlebih dahulu seperti yang ditampilkan pada Gambar 4.45.
2. Membuat pemrograman pada aplikasi MIT App Inventor untuk koneksi antara *database* dengan MIT App Inventor. Berikut salah satu contoh pemrograman pemanggilan data dari *firebase* ke MIT App Inventor.



Gambar 4. 45 Tampilan Menu *Designer* pada MIT App Inventor

Untuk menghubungkan aplikasi MIT App Inventor dengan *firebase* maka pada menu *designer* yang berfungsi untuk membuat tampilan visual aplikasi pada *smartphone* perlu dimasukkan komponen FirebaseDB seperti yang dapat dilihat pada Gambar 4.64 di atas. Komponen ini dapat ditemukan pada menu Experimental di sebelah kiri tampilan menu *designer* pada MIT App Inventor.

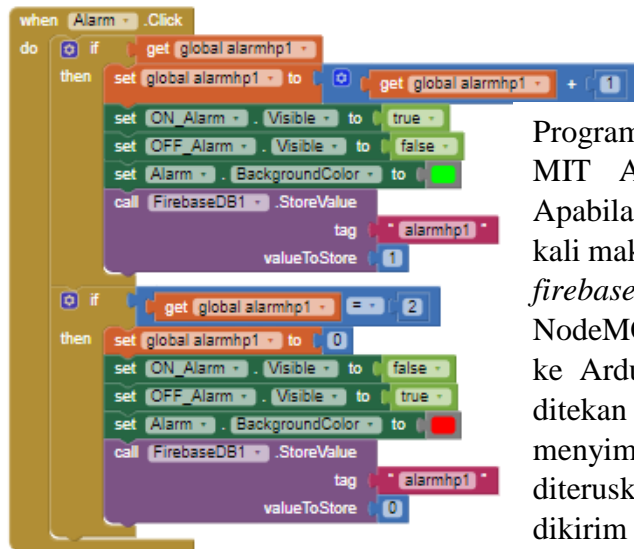
Setelah membuat tampilan visual aplikasi MIT App Inventor pada menu *designer* maka selanjutnya yaitu membuat pemrograman untuk menghubungkan MIT App Inventor ke *firebase* pada menu *blocks* sebagai berikut.



Program untuk menampilkan data dari Arduino ke aplikasi MIT App Inventor. Pada program ini data sensor yang dikirim dari Arduino akan ditampilkan ke aplikasi MIT App Inventor secara *real time*.

Gambar 4. 46 Program Pemanggilan *Database* ke Aplikasi MIT App Inventor

Selain pemanggilan data dari *firebase* ke MIT App Inventor, dapat juga dibuat pemrograman untuk menyimpan data dari MIT App Inventor ke *firebase*. Salah satu contoh pemrograman untuk penyimpanan data dari MIT App Inventor ke *firebase* adalah sebagai berikut.



Program untuk mentransfer data dari MIT App Inventor ke Arduino. Apabila tombol Alarm ditekan satu kali maka akan menyimpan nilai 1 ke *firebase* dan kemudian diteruskan ke NodeMCU dan selanjutnya dikirim ke Arduino. Apabila tombol alarm ditekan kedua kali maka akan menyimpan nilai 0 ke *firebase* dan diteruskan ke NodeMCU kemudian dikirim ke Arduino.

Gambar 4. 47 Program Penyimpanan Data dari MIT App Inventor ke *Firestore*

4.5 Pengujian Alat

Pengujian alat secara keseluruhan dilakukan untuk mengetahui apakah sistem *monitoring* data sensor dan kontrol gedung dari bahaya kebakaran telah berfungsi dengan baik. Selain itu, dilakukan juga pengujian terhadap sistem IoT untuk mengetahui koneksi antara *prototype* pendeteksi kebakaran dengan *smartphone*.

Sebelum melakukan pengujian alat pendeteksi kebakaran, perlu ditentukan terlebih dahulu nilai ambang batas kebakaran untuk mendefinisikan suatu kondisi dikatakan terjadi kebakaran atau tidak. Dalam menentukan nilai ambang batas kebakaran untuk pengujian alat proyek akhir ini, dilakukan perbandingan antara nilai ambang batas kebakaran dalam suatu ukuran ruangan atau gedung yang sesungguhnya dengan ukuran *prototype* pendeteksi kebakaran yang dibuat untuk mendapatkan nilai ambang batas kebakaran yang sesuai.

▪ **Nilai ambang batas suhu saat terjadi kebakaran**

Pada penelitian yang dilakukan oleh [13] dibuat sebuah model bangunan dengan skala 1:3 untuk menguji distribusi suhu dan radiasi ruang api. Diketahui bahwa suhu ruangan pada model bangunan berada dalam kondisi normal antara 30-32°C. Pengujian dilakukan pada ruangan berukuran 0,8 m x 0,8 m x 1,2 m dan didapatkan suhu ruangan pada saat terjadi kebakaran bisa mencapai 100°C hingga naik menjadi 600°C [13]. Berikut adalah perhitungan untuk mendapatkan nilai ambang batas suhu saat terjadi kebakaran.

Diketahui:

- Suhu ruangan normal = 32°C
- Ukuran ruangan sebenarnya = 0,8 m × 0,8 m × 1,2 m
= 80 cm × 80 cm × 120 cm
= 768.000 cm³
- Ambang batas suhu sebenarnya = 100°C
- Ukuran prototype yang dibuat = 45 cm × 45 cm × 30 cm
= 60.750 cm³
- Ambang batas suhu prototype = x

Maka,

$$\frac{\text{Ukuran prototype}}{\text{Ukuran ruangan sebenarnya}} = \frac{\text{Ambang batas suhu prototype}}{\text{Ambang batas suhu sebenarnya}}$$

$$\frac{60.750 \text{ cm}^3}{768.000 \text{ cm}^3} = \frac{x - 32^\circ\text{C}}{100^\circ\text{C} - 32^\circ\text{C}}$$

$$x - 32^\circ\text{C} = \frac{60.750 \times 68^\circ\text{C}}{768.000}$$

$$x - 32^\circ\text{C} = \frac{4.131.000 \text{ }^\circ\text{C}}{768.000}$$

$$x - 32^\circ\text{C} = 5,38^\circ\text{C}$$

$$x = 5,38^\circ\text{C} + 32^\circ\text{C}$$

$$x = 37,38^\circ\text{C}$$

Jadi, nilai ambang batas suhu saat terjadi kebakaran adalah 37,38°C.

▪ **Nilai ambang batas asap saat terjadi kebakaran**

Selain suhu api, ambang batas kebakaran juga ditentukan berdasarkan kadar asap yang dihasilkan. Pembakaran Konfigurasi Prancis dan AS menyatakan bahwa peristiwa kebakaran menghasilkan gas emisi beracun seperti gas CO dengan kadar 28.000 ppm dan 38.200 ppm yang akan terdeteksi antara 6 menit dan 8 menit pada ruangan berukuran 3,66 m x 2,44 m x 2,44 m [15].

Diketahui:

- Ukuran ruangan sebenarnya = $3,66 \text{ m} \times 2,44 \text{ m} \times 2,44 \text{ m}$
 $= 366 \text{ cm} \times 244 \text{ cm} \times 244 \text{ cm}$
 $= 21.790.176 \text{ cm}^3$
- Ambang batas asap sebenarnya = 28.000 ppm dan 38.200 ppm
- Ukuran prototype yang dibuat = $45 \text{ cm} \times 45 \text{ cm} \times 30 \text{ cm}$
 $= 60.750 \text{ cm}^3$
- Ambang batas suhu prototype = y

Maka,

1). Jika Ambang batas asap sebenarnya = 28.000 ppm

$$\frac{\text{Ukuran prototype}}{\text{Ukuran ruangan sebenarnya}} = \frac{\text{Ambang batas asap prototype}}{\text{Ambang batas asap sebenarnya}}$$

$$\frac{60.750 \text{ cm}^3}{21.790.176 \text{ cm}^3} = \frac{y}{28.000 \text{ ppm}}$$

$$y = \frac{60.750 \times 28.000 \text{ ppm}}{21.790.176}$$

$$y = \frac{1.701.000.000 \text{ ppm}}{21.790.176}$$

$$y = 78 \text{ ppm}$$

2). Jika Ambang batas asap sebenarnya = 38.200 ppm

$$\frac{\text{Ukuran prototype}}{\text{Ukuran ruangan sebenarnya}} = \frac{\text{Ambang batas asap prototype}}{\text{Ambang batas asap sebenarnya}}$$

$$\frac{60.750 \text{ cm}^3}{21.790.176 \text{ cm}^3} = \frac{y}{38.200 \text{ ppm}}$$

$$y = \frac{60.750 \times 38.200 \text{ ppm}}{21.790.176}$$

$$y = \frac{2.320.650.000 \text{ ppm}}{21.790.176}$$

$$y = 106,45 \text{ ppm}$$

Jadi, nilai ambang batas asap saat terjadi kebakaran yang digunakan pada pengujian *prototype* ini adalah 106,45 ppm.

4.5.1 Pengujian *Prototype* Pendeteksi Kebakaran

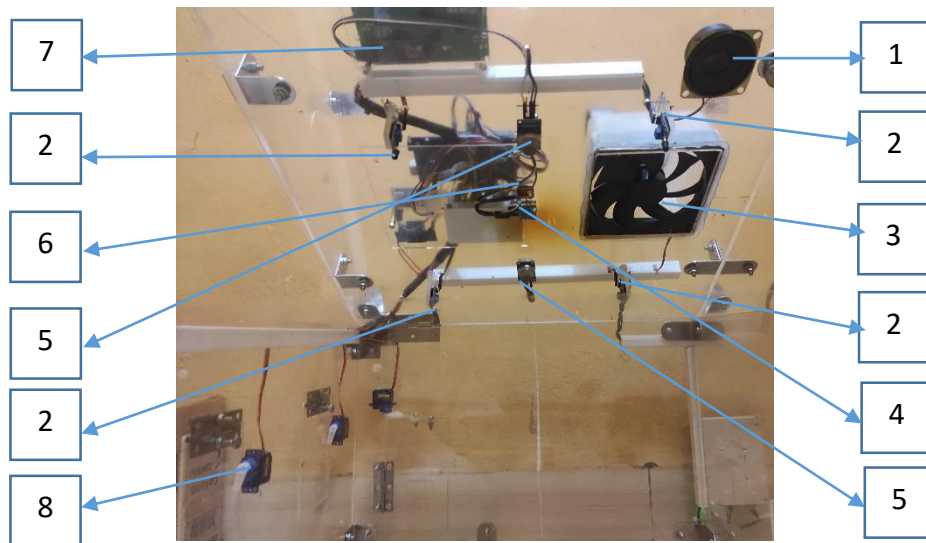
Pengujian ini bertujuan untuk menganalisa apakah sistem *monitoring* dari data sensor api, sensor suhu dan sensor asap serta sistem kontrol untuk mengaktifkan alarm, *exhaust*, motor servo dan LED sudah berfungsi sesuai dengan yang diharapkan. Pengujian ini dilakukan pada ruang 1 dan ruang 2 *prototype* pendeteksi kebakaran dengan penempatan komponen-komponen untuk keperluan pengujian alat sebagai berikut.



Gambar 4. 48 Penempatan Komponen pada Ruang 1

Keterangan Ruang 1:

- | | | | |
|------------------------|-------------------|----------------|------------------|
| 1. Alarm | 3. <i>Exhaust</i> | 5. Motor servo | 7. LCD |
| 2. <i>Flame sensor</i> | 4. Sensor MQ-9 | 6. LED | 8. Sensor DHT-11 |



Gambar 4. 49 Penempatan Komponen pada Ruang 2

Keterangan Ruang 2:

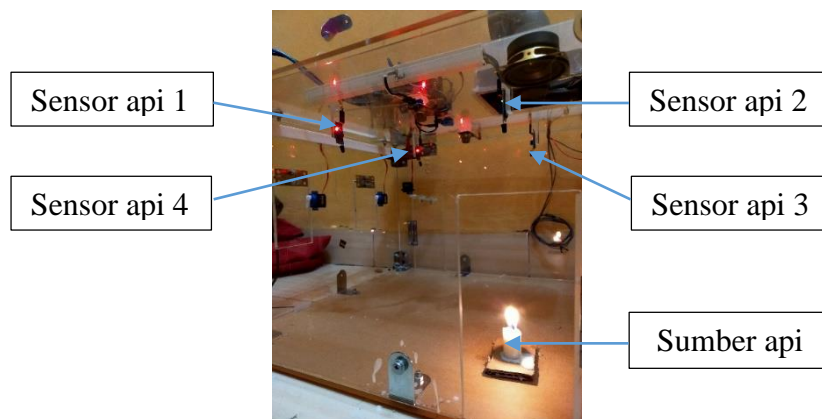
- | | | | |
|------------------------|-------------------|----------------|----------------|
| 1. Alarm | 3. <i>Exhaust</i> | 5. Sensor MQ-9 | 7. LED |
| 2. <i>Flame sensor</i> | 4. Sensor DHT-11 | 6. LCD | 8. Motor servo |

Pengujian ini dilakukan dengan cara menguji sistem pendeteksi kebakaran ketika hanya terdeteksi api, asap, suhu, api dan asap, api dan suhu, asap dan suhu serta ketiganya yaitu api, asap dan suhu. Pada pengujian ini terdapat nilai ambang batas kebakaran yaitu ketika terdeteksi adanya api dengan kadar asap >110 ppm dan suhu >35°C. Ketika nilai ambang batas ini terpenuhi maka sistem akan secara otomatis mengaktifkan alarm, *exhaust*, motor servo untuk membuka pintu dan jendela serta LED sebagai indikator pemadaman api. Oleh karena mempunyai prosedur pengujian yang sama maka data hasil pengujian dari ruang 1 dan ruang 2 akan ditampilkan sebagai berikut.

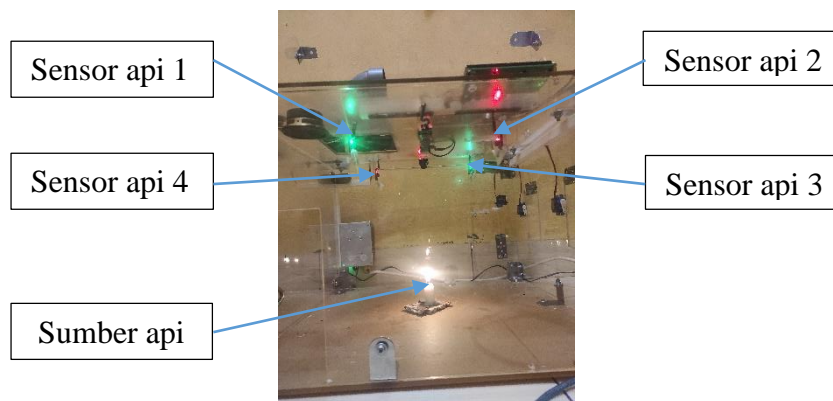
4.5.1.1 Pengujian Kontrol Kebakaran dengan *Input Api*

Pengujian ini bertujuan untuk menganalisa kontrol tindakan penanganan kebakaran jika diberikan *input* berupa api. Prosedur pengujian dilakukan dengan memberikan *input* api pada masing-masing ruangan dengan simulasi api menggunakan lilin. Sensor api yang digunakan pada setiap ruangan sebanyak 4

buah. Oleh karena itu, pada pengujian ini lilin digeser ke berbagai posisi yang berbeda untuk mendapatkan variasi data pembacaan api oleh sensor-sensor api yang lain. Kemudian dilakukan pengamatan terhadap pembacaan data sensor pada LCD dan *output* alarm, *exhaust*, pintu, jendela dan LED menjadi aktif atau tidak. Hasil pengujian ini kemudian dicatat pada Tabel 4.4 untuk ruang 1 dan Tabel 4.5 untuk ruang 2.



Gambar 4. 50 Pengujian Terdeteksi Api pada Ruang 1



Gambar 4. 51 Pengujian Terdeteksi Api pada Ruang 2

Hasil pembacaan pada LCD adalah sebagai berikut.



Gambar 4. 52 Tampilan LCD Ketika Terdeteksi Api pada Ruang 1



Gambar 4. 53 Tampilan LCD Ketika Terdeteksi Api pada Ruang 2

Pada tampilan LCD tersebut diketahui bahwa ketika terdeteksi api maka akan ditampilkan nilai 0 dan ketika tidak terdeteksi api akan ditampilkan nilai 1.

Tabel 4. 4 Hasil Pengujian Terdeteksi Api Pada Ruang 1

Input	Pembacaan data sensor api ke- pada LCD				Output Kondisi Penanganan Kebakaran				
	1	2	3	4	Alarm	Exhaust	Pintu	Jendela	LED
Api dari lilin	0	1	1	1	OFF	OFF	CLOSE	CLOSE	OFF
	1	0	1	1	OFF	OFF	CLOSE	CLOSE	OFF
	0	0	1	1	OFF	OFF	CLOSE	CLOSE	OFF
	1	1	0	0	OFF	OFF	CLOSE	CLOSE	OFF
	1	0	0	0	OFF	OFF	CLOSE	CLOSE	OFF
	0	0	1	0	OFF	OFF	CLOSE	CLOSE	OFF
	0	0	0	0	OFF	OFF	CLOSE	CLOSE	OFF

Tabel 4. 5 Hasil Pengujian Terdeteksi Api Pada Ruang 2

Input	Pembacaan data sensor api ke- pada LCD				Output Kondisi Penanganan Kebakaran				
	1	2	3	4	Alarm	Exhaust	Pintu	Jendela	LED
Api dari lilin	0	1	1	1	OFF	OFF	CLOSE	CLOSE	OFF
	1	0	1	1	OFF	OFF	CLOSE	CLOSE	OFF
	0	0	1	1	OFF	OFF	CLOSE	CLOSE	OFF
	1	1	0	0	OFF	OFF	CLOSE	CLOSE	OFF
	1	0	0	0	OFF	OFF	CLOSE	CLOSE	OFF
	0	0	0	1	OFF	OFF	CLOSE	CLOSE	OFF
	0	0	0	0	OFF	OFF	CLOSE	CLOSE	OFF

Keterangan:

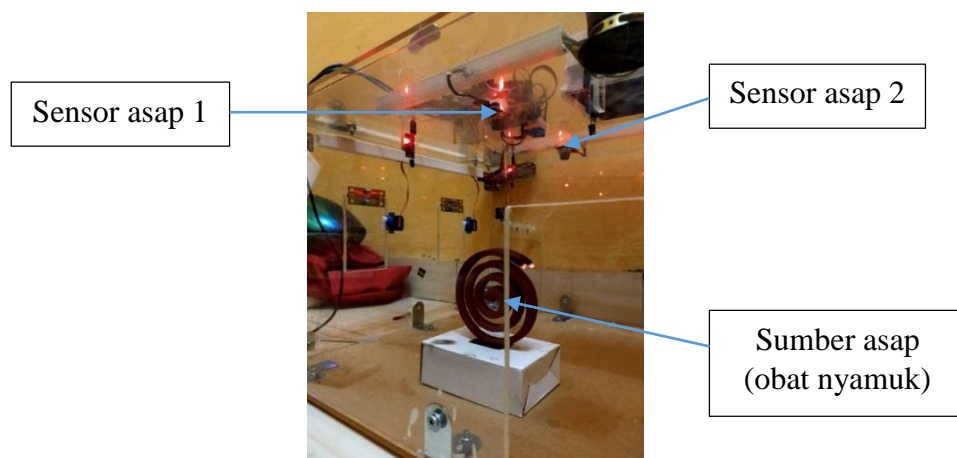
1. Dalam pengujian ini, logika 0 menyatakan sensor api mendeteksi adanya api sedangkan logika 1 menyatakan sensor api tidak mendeteksi adanya api.

Kesimpulan:

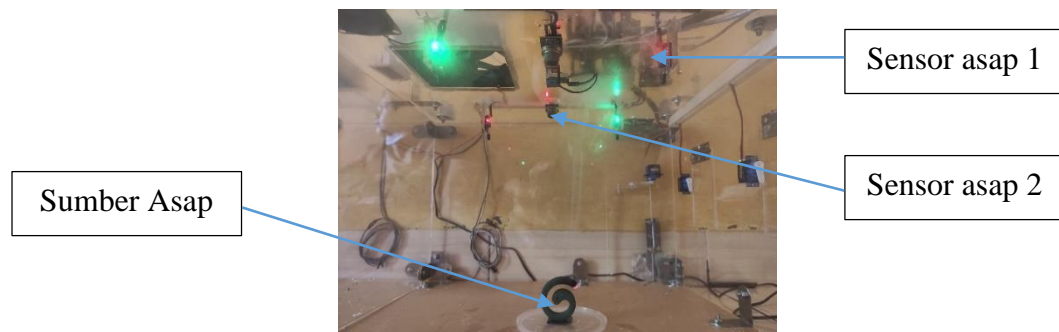
Dari tabel hasil pengujian dapat disimpulkan bahwa ketika hanya terdeteksi api pada ruangan maka kondisi tersebut dikatakan tidak kebakaran sehingga semua tindakan penanganan kebakarana berupa alarm, *exhaust*, motor servo untuk membuka pintu dan jendela serta LED tidak aktif.

4.5.1.2 Pengujian Kontrol Kebakaran dengan *Input* Asap

Pengujian ini bertujuan untuk menganalisa kontrol tindakan penanganan kebakaran jika diberikan *input* berupa asap. Prosedur pengujian dilakukan dengan memberikan *input* asap pada masing-masing ruangan dengan simulasi asap menggunakan obat nyamuk yang diuji selama ± 1 menit. Kemudian dilakukan pengamatan terhadap pembacaan sensor pada LCD dan *output* alarm, *exhaust*, pintu, jendela dan LED menjadi aktif atau tidak. Hasil pengujian ini kemudian dicatat pada Tabel 4.6 untuk ruang 1 dan Tabel 4.7 untuk ruang 2.

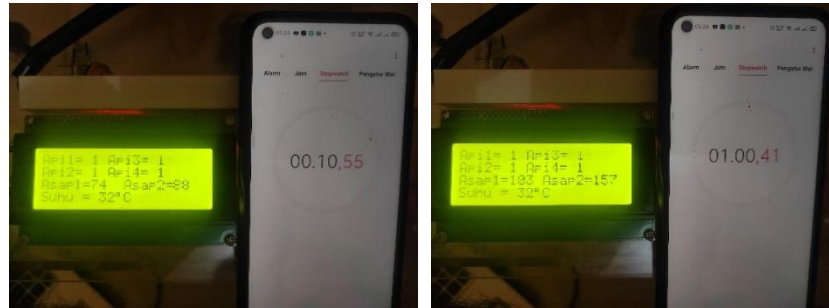


Gambar 4. 54 Pengujian Terdeteksi Asap pada Ruang 1

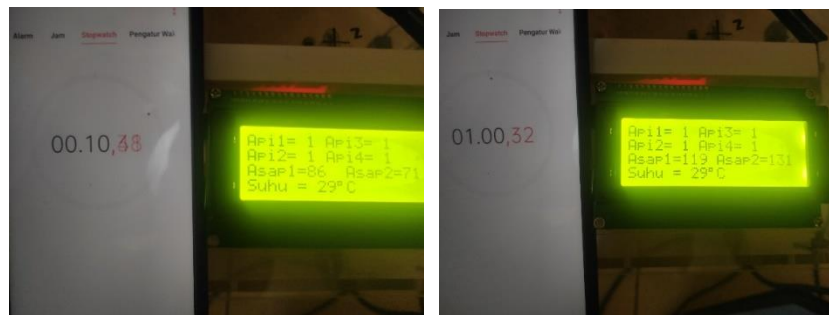


Gambar 4. 55 Pengujian Terdeteksi Asap pada Ruang 2

Hasil pembacaan pada LCD adalah sebagai berikut.



Gambar 4. 56 Tampilan LCD Pengujian Asap pada Ruang 1



Gambar 4. 57 Tampilan LCD Pengujian Asap pada Ruang 2

Tabel 4. 6 Hasil Pengujian Terdeteksi Asap pada Ruang 1

<i>Input</i>	Pembacaan data sensor asap ke-pada LCD (ppm)		<i>Output</i> kontrol saat terdeteksi asap					
	Waktu (s)	1	2	Alarm	<i>Exhaust</i>	Pintu	Jendela	LED
Asap dari obat nyamuk	10	86	71	OFF	OFF	CLOSE	CLOSE	OFF
	20	93	93	OFF	OFF	CLOSE	CLOSE	OFF
	30	102	103	OFF	OFF	CLOSE	CLOSE	OFF
	40	109	116	OFF	OFF	CLOSE	CLOSE	OFF
	50	114	125	OFF	OFF	CLOSE	CLOSE	OFF
	60	119	131	OFF	OFF	CLOSE	CLOSE	OFF

Tabel 4. 7 Hasil Pengujian Terdeteksi Asap pada Ruang 2

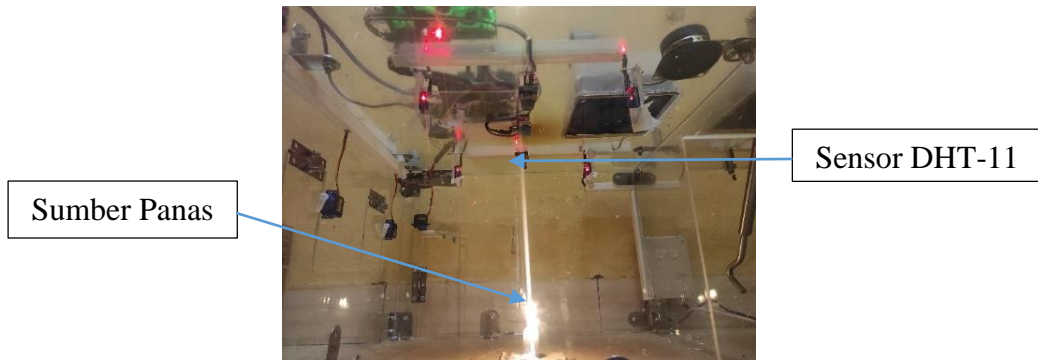
<i>Input</i>	Pembacaan data sensor asap ke-pada LCD (ppm)			<i>Output</i> kontrol saat terdeteksi asap				
	Waktu (s)			Alarm	<i>Exhaust</i>	Pintu	Jendela	LED
		1	2					
Asap dari obat nyamuk	10	86	71	OFF	OFF	CLOSE	CLOSE	OFF
	20	93	93	OFF	OFF	CLOSE	CLOSE	OFF
	30	102	103	OFF	OFF	CLOSE	CLOSE	OFF
	40	109	116	OFF	OFF	CLOSE	CLOSE	OFF
	50	114	125	OFF	OFF	CLOSE	CLOSE	OFF
	60	119	131	OFF	OFF	CLOSE	CLOSE	OFF

Kesimpulan:

Dari tabel hasil pengujian diketahui bahwa nilai kadar asap yang dideteksi oleh sensor MQ-9 berubah-ubah dan mengalami peningkatan seiring dengan bertambahnya waktu. Hal ini disebabkan kuantitas asap yang dihasilkan oleh obat nyamuk tidak selalu tetap dan bergerak mengikuti udara yang berada di sekitarnya sehingga nilai dari kadar asap bisa saja meningkat atau menurun dalam satu satuan waktu. Dalam kasus ini, dikarenakan pengujian dilakukan pada ruangan tertutup maka kadar asap semakin lama semakin meningkat. Dari hasil pengujian didapatkan jika hanya terdapat asap saja maka tidak terjadi kebakaran dan *output* kontrol kebakaran yaitu alarm, *exhaust*, motor servo untuk membuka pintu dan jendela serta LED tidak aktif.

4.5.1.3 Pengujian Kontrol Kebakaran dengan *Input* Suhu

Pengujian ini bertujuan untuk menganalisa kontrol tindakan penanganan kebakaran jika diberikan *input* berupa suhu. Prosedur pengujian dilakukan dengan memberikan *input* suhu pada masing-masing ruangan dengan simulasi suhu menggunakan lilin untuk mendapatkan suhu panas yang diuji selama ± 1 menit. Kemudian dilakukan pengamatan terhadap pembacaan sensor pada LCD dan *output* alarm, *exhaust*, pintu, jendela dan LED menjadi aktif atau tidak. Hasil pengujian ini kemudian dicatat pada Tabel 4.8 untuk ruang 1 dan Tabel 4.9 untuk ruang 2.

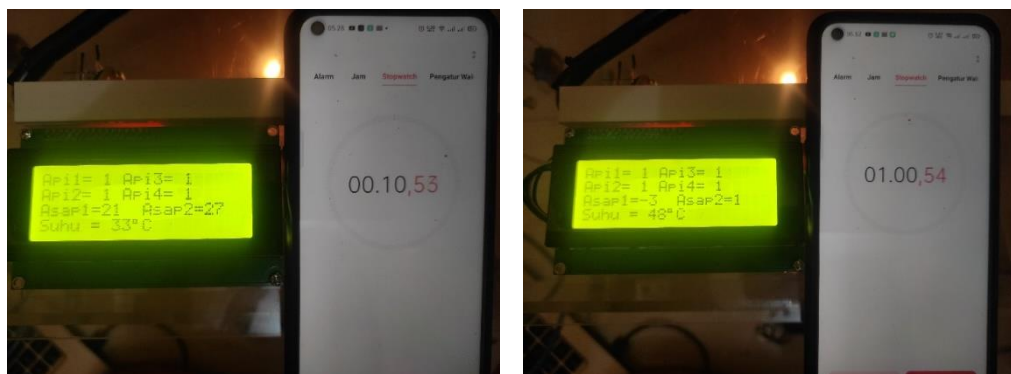


Gambar 4. 58 Pengujian Terdeteksi Suhu pada Ruang 1

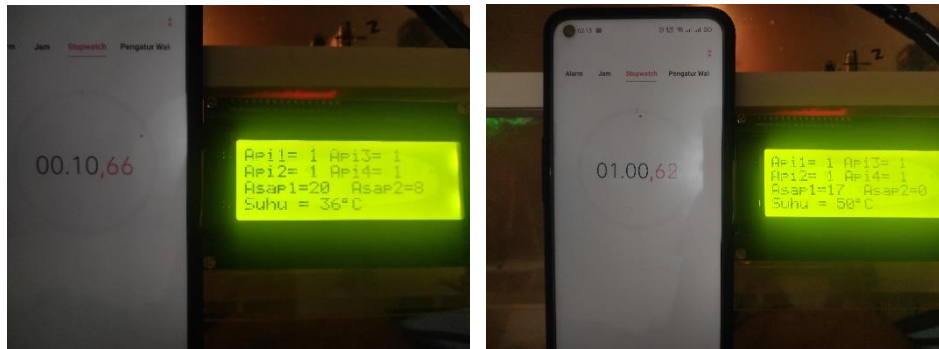


Gambar 4. 59 Pengujian Terdeteksi Suhu pada Ruang 2

Hasil pembacaan pada LCD adalah sebagai berikut.



Gambar 4. 60 Tampilan LCD Terdeteksi Suhu pada Ruang 1



Gambar 4. 61 Tampilan LCD Terdeteksi Suhu pada Ruang 2

Tabel 4. 8 Pengujian Terdeteksi Suhu pada Ruang 1

<i>Input</i>	Pembacaan data sensor		<i>Output</i> Kondisi Tindakan Penanganan Kebakaran				
	Waktu (s)	suhu pada LCD (°C)	Alarm	<i>Exhaust</i>	Pintu	Jendela	LED
Suhu panas dari lilin	10	33	OFF	OFF	CLOSE	CLOSE	OFF
	20	36	OFF	OFF	CLOSE	CLOSE	OFF
	30	38	OFF	OFF	CLOSE	CLOSE	OFF
	40	39	OFF	OFF	CLOSE	CLOSE	OFF
	50	43	OFF	OFF	CLOSE	CLOSE	OFF
	60	48	OFF	OFF	CLOSE	CLOSE	OFF

Tabel 4. 9 Hasil Pengujian Terdeteksi Suhu pada Ruang 2

<i>Input</i>	Pembacaan data sensor		<i>Output</i> Kondisi Tindakan Penanganan Kebakaran				
	Waktu (s)	suhu pada LCD (°C)	Alarm	<i>Exhaust</i>	Pintu	Jendela	LED
Suhu panas dari lilin	10	36	OFF	OFF	CLOSE	CLOSE	OFF
	20	42	OFF	OFF	CLOSE	CLOSE	OFF
	30	47	OFF	OFF	CLOSE	CLOSE	OFF
	40	49	OFF	OFF	CLOSE	CLOSE	OFF
	50	50	OFF	OFF	CLOSE	CLOSE	OFF
	60	50	OFF	OFF	CLOSE	CLOSE	OFF

Keterangan:

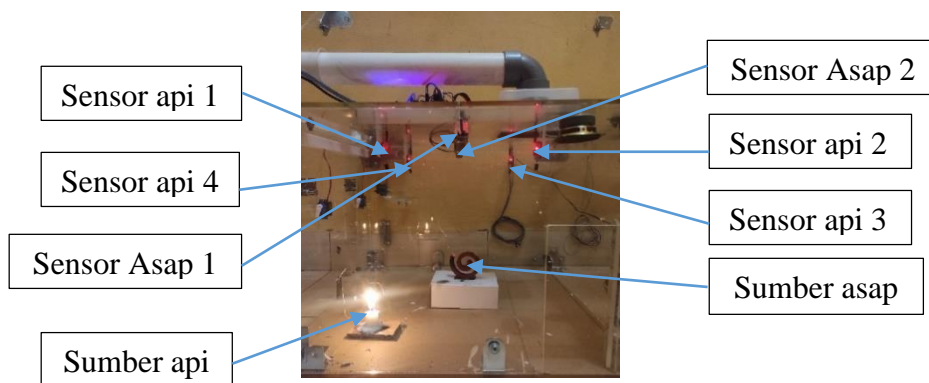
1. Suhu awal ruangan pada pengujian ini adalah 32°C.

Kesimpulan:

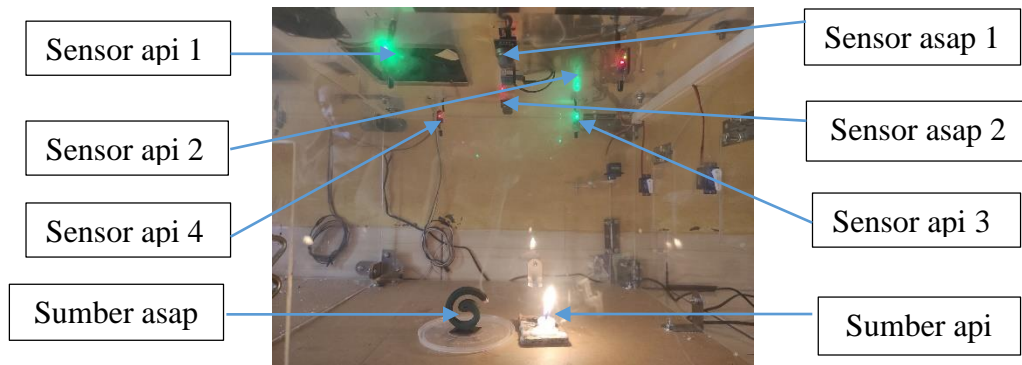
Dari tabel hasil pengujian diketahui bahwa nilai suhu ruangan mengalami peningkatan seiring dengan bertambahnya waktu. Hal ini disebabkan terjadi pemanasan suhu ruangan dari api lilin selama 1 menit sehingga suhu ruangan menjadi meningkat. Kemudian dari *output* yang dihasilkan dapat disimpulkan bahwa ketika hanya terdeteksi suhu maka kondisi tersebut dikatakan tidak terjadi kebakaran sehingga semua tindakan penanganan kebakaran berupa alarm, *exhaust*, motor servo untuk mengaktifkan pintu dan jendela serta LED tidak aktif.

4.5.1.4 Pengujian Kontrol Kebakaran dengan *Input* Api dan Asap

Pengujian ini bertujuan untuk menganalisa kontrol tindakan penanganan kebakaran jika diberikan *input* berupa api dan asap. Prosedur pengujian dilakukan dengan memberikan *input* api dan asap pada masing-masing ruangan dengan simulasi api menggunakan lilin diletakkan sembarang di bawah salah satu sensor api dan asap dari obat nyamuk diletakkan di tengah ruang miniatur yang diuji selama ± 1 menit dan setiap 10 detik lilin digeser ke posisi yang berbeda untuk mendapatkan variasi data dari pendeteksian sensor api. Kemudian dilakukan pengamatan terhadap pembacaan sensor pada LCD dan *output* alarm, *exhaust*, pintu, jendela dan LED menjadi aktif atau tidak. Hasil pengujian ini kemudian dicatat pada Tabel 4.10 untuk ruang 1 dan Tabel 4.11 untuk ruang 2.



Gambar 4. 62 Pengujian Terdeteksi Api dan Asap pada Ruang 1

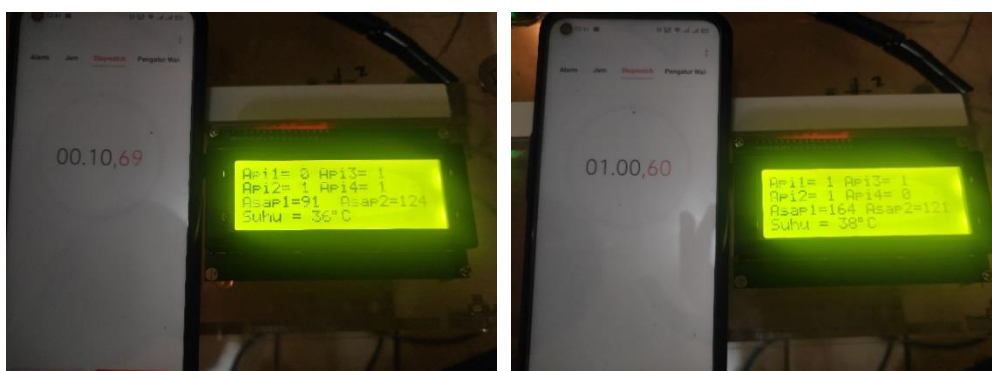


Gambar 4. 63 Pengujian Terdeteksi Api dan Asap pada Ruang 2

Hasil pembacaan pada LCD adalah sebagai berikut.



Gambar 4. 64 Tampilan LCD Terdeteksi Api dan Asap pada Ruang 1



Gambar 4. 65 Tampilan LCD Terdeteksi Api dan Asap pada Ruang 2

Tabel 4. 10 Pengujian Terdeteksi Api dan Asap pada Ruang 1

<i>Input</i>	Waktu (s)	Pembacaan data sensor api ke-pada LCD				Pembacaan data sensor asap ke- pada LCD (ppm)		<i>Output</i> Kondisi Tindakan Penanganan Kebakaran				
		1	2	3	4	1	2	Alarm	<i>Exhaust</i>	Pintu	Jendela	LED
								OFF	OFF	CLOSE	CLOSE	OFF
Api dari lilin dan asap dari obat nyamuk	10	0	1	1	1	126	223	OFF	OFF	CLOSE	CLOSE	OFF
	20	0	0	1	1	146	235	OFF	OFF	CLOSE	CLOSE	OFF
	30	1	0	0	1	147	227	OFF	OFF	CLOSE	CLOSE	OFF
	40	1	1	0	0	167	233	OFF	OFF	CLOSE	CLOSE	OFF
	50	1	1	1	0	151	241	OFF	OFF	CLOSE	CLOSE	OFF
	60	1	1	0	1	167	222	OFF	OFF	CLOSE	CLOSE	OFF

Keterangan:

1. Dalam pengujian ini, logika 0 menyatakan sensor api mendeteksi api dan logika 1 menyatakan bahwa sensor api tidak mendeteksi api.

Tabel 4. 11 Hasil Pengujian Terdeteksi Api dan Asap pada Ruang 2

<i>Input</i>	Waktu (s)	Pembacaan pada LCD data sensor api ke-				Pembacaan data sensor asap ke- pada LCD (ppm)		<i>Output</i> Kondisi Tindakan Penanganan Kebakaran				
		1	2	3	4	1	2	Alarm	<i>Exhaust</i>	Pintu	Jendela	LED
Api dari lilin dan asap dari obat nyamuk	10	0	1	1	1	91	124	OFF	OFF	CLOSE	CLOSE	OFF
	20	0	0	1	1	117	122	OFF	OFF	CLOSE	CLOSE	OFF
	30	1	0	0	1	135	137	OFF	OFF	CLOSE	CLOSE	OFF
	40	1	1	0	1	157	118	OFF	OFF	CLOSE	CLOSE	OFF
	50	1	1	0	0	161	132	OFF	OFF	CLOSE	CLOSE	OFF
	60	1	1	1	0	164	121	OFF	OFF	CLOSE	CLOSE	OFF

Keterangan:

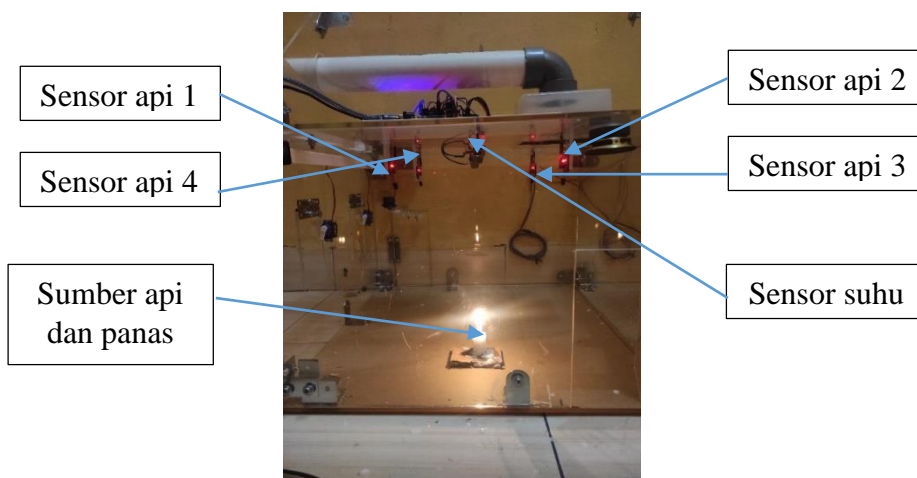
1. Dalam pengujian ini, logika 0 menyatakan sensor api mendeteksi api dan logika 1 menyatakan bahwa sensor api tidak mendeteksi api.

Kesimpulan:

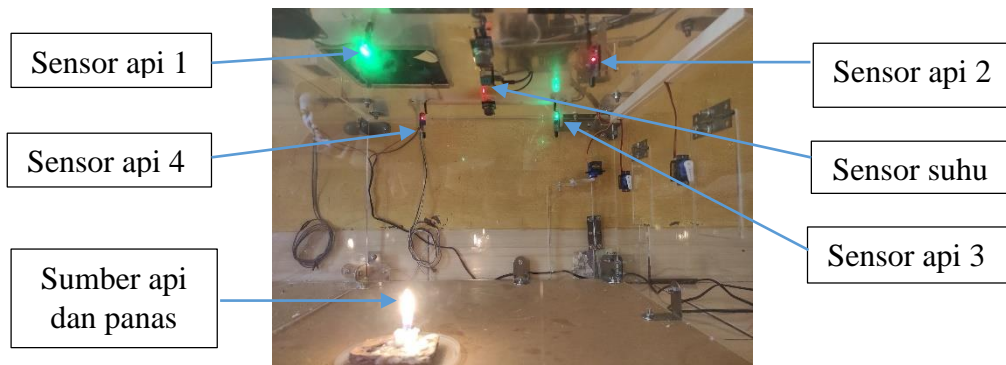
Dari tabel hasil pengujian diketahui bahwa nilai kadar asap yang dideteksi oleh sensor MQ-9 berubah-ubah dan tidak konsisten seiring dengan bertambahnya waktu. Hal ini disebabkan kuantitas asap yang dihasilkan oleh obat nyamuk tidak selalu tetap dan bergerak mengikuti udara yang berada di sekitarnya sehingga nilai dari kadar asap bisa saja meningkat atau menurun dalam satu satuan waktu. Kemudian dari *output* yang dihasilkan dapat disimpulkan bahwa ketika hanya terdeteksi api dan asap hingga melebihi nilai ambang batas yaitu >110 ppm maka kondisi tersebut dikatakan tidak terjadi kebakaran sehingga alarm, *exhaust*, motor servo untuk membuka pintu dan jendela serta LED tidak aktif.

4.5.1.5 Pengujian Kontrol Kebakaran dengan *Input* Api dan Suhu

Pengujian ini bertujuan untuk menganalisa kontrol tindakan penanganan kebakaran jika diberikan *input* berupa api dan suhu. Prosedur pengujian dilakukan dengan memberikan *input* api dan suhu pada masing-masing ruangan dengan simulasi api dan suhu panas dari diletakkan di tengah ruang miniatur yang diuji selama ± 1 menit. Kemudian dilakukan pengamatan terhadap pembacaan sensor pada LCD dan *output* alarm, *exhaust*, pintu, jendela dan LED menjadi aktif atau tidak. Hasil pengujian ini kemudian dicatat pada Tabel 4.12 untuk ruang 1 dan Tabel 4.13 untuk ruang 2.

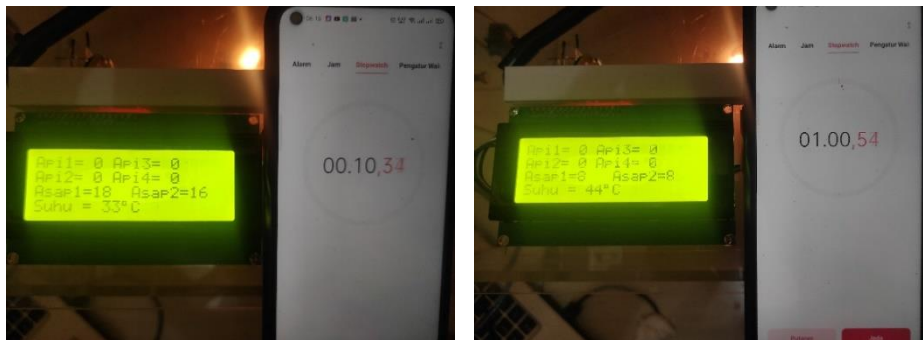


Gambar 4. 66 Pengujian Terdeteksi Api dan Suhu pada Ruang 1

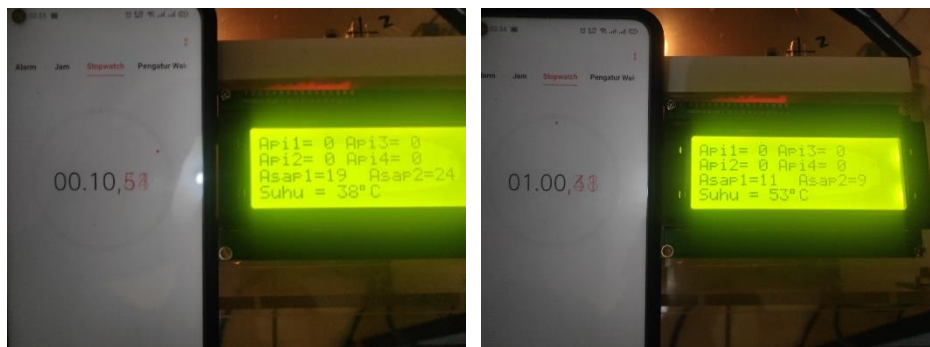


Gambar 4. 67 Pengujian Terdeteksi Api dan Suhu Ruang 2

Hasil pembacaan pada LCD adalah sebagai berikut.



Gambar 4. 68 Tampilan LCD Terdeteksi Api dan Suhu pada Ruang 1



Gambar 4. 69 Tampilan LCD Terdeteksi Api dan Suhu pada Ruang 2

Tabel 4. 12 Pengujian Terdeteksi Api dan Suhu pada Ruang 1

<i>Input</i>	Waktu (s)	Pembacaan data sensor api ke- pada LCD				Pembacaan data sensor suhu pada LCD (°C)	<i>Output</i> kondisi tindakan penanganan kebakaran				
		1	2	3	4		Alarm	<i>Exhaust</i>	Pintu	Jendela	LED
Api dan suhu panas dari lilin	10	0	0	0	0	33	OFF	OFF	CLOSE	CLOSE	OFF
	20	0	0	0	0	36	OFF	OFF	CLOSE	CLOSE	OFF
	30	0	0	0	0	38	OFF	OFF	CLOSE	CLOSE	OFF
	40	0	0	0	0	39	OFF	OFF	CLOSE	CLOSE	OFF
	50	0	0	0	0	40	OFF	OFF	CLOSE	CLOSE	OFF
	60	0	0	0	0	44	OFF	OFF	CLOSE	CLOSE	OFF

Keterangan:

Suhu awal ruangan pada pengujian ini adalah 31°C

Tabel 4. 13 Hasil Pengujian Terdeteksi Api dan Suhu pada Ruang 2

<i>Input</i>	Waktu (s)	Pembacaan data sensor api ke- pada LCD				Pembacaan data sensor suhu pada LCD (°C)	<i>Output</i> kondisi tindakan penanganan kebakaran				
		1	2	3	4		Alarm	<i>Exhaust</i>	Pintu	Jendela	LED
Api dan suhu panas dari lilin	10	0	0	0	0	38	OFF	OFF	CLOSE	CLOSE	OFF
	20	0	0	0	0	43	OFF	OFF	CLOSE	CLOSE	OFF
	30	0	0	0	0	47	OFF	OFF	CLOSE	CLOSE	OFF
	40	0	0	0	0	49	OFF	OFF	CLOSE	CLOSE	OFF
	50	0	0	0	0	51	OFF	OFF	CLOSE	CLOSE	OFF
	60	0	0	0	0	53	OFF	OFF	CLOSE	CLOSE	OFF

Keterangan:

1. Suhu awal ruangan pada pengujian ini adalah 30°C.

Kesimpulan:

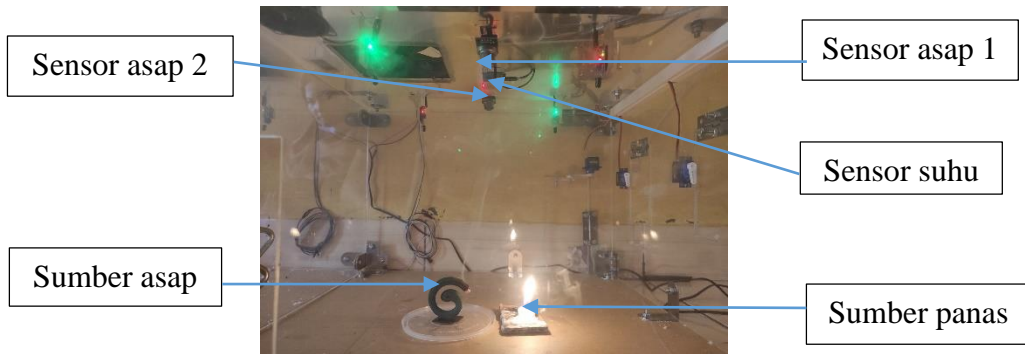
Dari tabel hasil pengujian diketahui bahwa semua sensor api mendeteksi adanya api atau berlogika 0. Hal ini disebabkan pada saat pengujian lilin diletakkan di tengah ruangan miniatur sehingga api bisa dideteksi oleh sensor api. Kemudian nilai suhu ruangan mengalami peningkatan seiring dengan bertambahnya waktu. Hal ini disebabkan terjadi pemanasan suhu ruangan dari api lilin selama ± 1 menit sehingga suhu ruangan menjadi meningkat. Kemudian dari *output* yang dihasilkan dapat disimpulkan bahwa ketika hanya terdeteksi api dan suhu hingga mencapai nilai ambang batas yaitu $>38^{\circ}\text{C}$ maka kondisi tersebut dikatakan tidak terjadi kebakaran sehingga semua tindakan penanganan kebakaran berupa alarm, *exhaust*, motor servo untuk mengaktifkan pintu dan jendela tidak aktif.

4.5.1.6 Pengujian Kontrol Kebakaran dengan *Input* Asap dan Suhu

Pengujian ini bertujuan untuk menganalisa kontrol tindakan penanganan kebakaran jika diberikan *input* berupa asap dan suhu. Prosedur pengujian dilakukan dengan memberikan *input* asap dan suhu pada masing-masing ruangan dengan simulasi asap dari obat nyamuk dan suhu panas dari lilin diletakkan di tengah ruang miniatur yang diuji selama ± 1 menit. Kemudian dilakukan pengamatan terhadap pembacaan sensor pada LCD dan *output* alarm, *exhaust*, pintu, jendela dan LED menjadi aktif atau tidak. Hasil pengujian ini kemudian dicatat pada Tabel 4.14 untuk ruang 1 dan Tabel 4.15 untuk ruang 2.

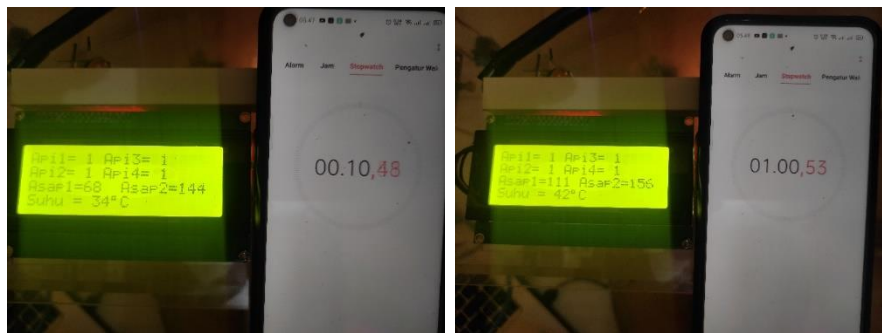


Gambar 4. 70 Pengujian Terdeteksi Asap dan Suhu pada Ruang 1

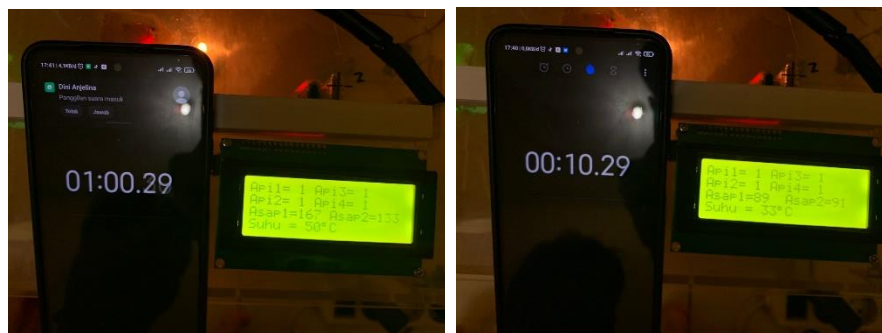


Gambar 4. 71 Pengujian Terdeteksi Asap dan Suhu pada Ruang 2

Hasil pembacaan pada LCD adalah sebagai berikut.



Gambar 4. 72 Tampilan LCD Terdeteksi Asap dan Suhu pada Ruang 1



Gambar 4. 73 Tampilan LCD Terdeteksi Asap dan Suhu pada Ruang 2

Tabel 4. 14 Hasil Pengujian Terdeteksi Asap dan Suhu pada Ruang 1

<i>Input</i>	Waktu (s)	Pembacaan data sensor asap ke- pada LCD (ppm)		Pembacaan data sensor pada LCD (°C)	<i>Output</i> kondisi tindakan penanganan kebakaran				
		1	2		Alarm	<i>Exhaust</i>	Pintu	Jendela	LED
		Asap dari obat nyamuk, suhu panas dari lilin	10		68	144	34	OFF	OFF
	20	93	153	36	OFF	OFF	CLOSE	CLOSE	OFF
	20	96	160	38	OFF	OFF	CLOSE	CLOSE	OFF
	40	101	155	40	OFF	OFF	CLOSE	CLOSE	OFF
	50	104	157	41	OFF	OFF	CLOSE	CLOSE	OFF
	60	111	156	42	OFF	OFF	CLOSE	CLOSE	OFF

Tabel 4. 15 Pengujian Terdeteksi Asap dan Suhu pada Ruang 2

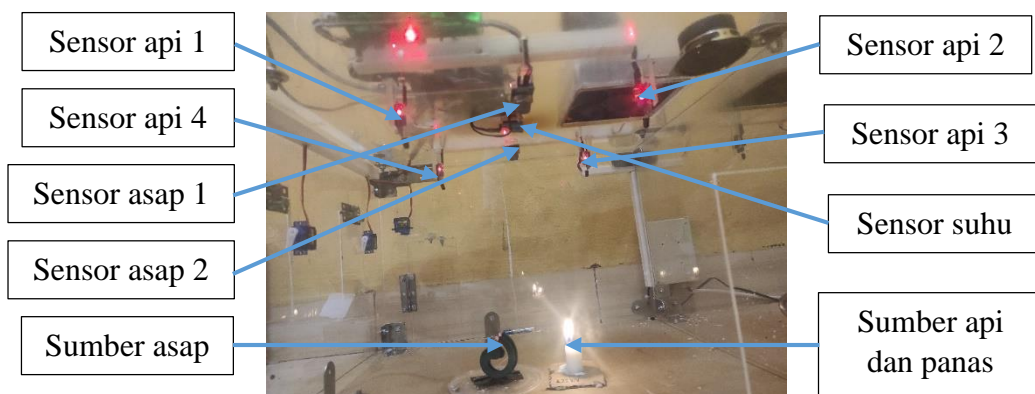
<i>Input</i>	Waktu (s)	Pembacaan data sensor asap ke- pada LCD (ppm)		Pembacaan data sensor pada LCD (°C)	<i>Output</i> kondisi tindakan penanganan kebakaran				
		1	2		Alarm	<i>Exhaust</i>	Pintu	Jendela	LED
		Asap dari obat nyamuk, suhu panas dari lilin	10		89	91	33	OFF	OFF
	20	128	107	37	OFF	OFF	CLOSE	CLOSE	OFF
	30	142	131	40	OFF	OFF	CLOSE	CLOSE	OFF
	40	154	123	44	OFF	OFF	CLOSE	CLOSE	OFF
	50	166	133	48	OFF	OFF	CLOSE	CLOSE	OFF
	60	167	133	50	OFF	OFF	CLOSE	CLOSE	OFF

Kesimpulan:

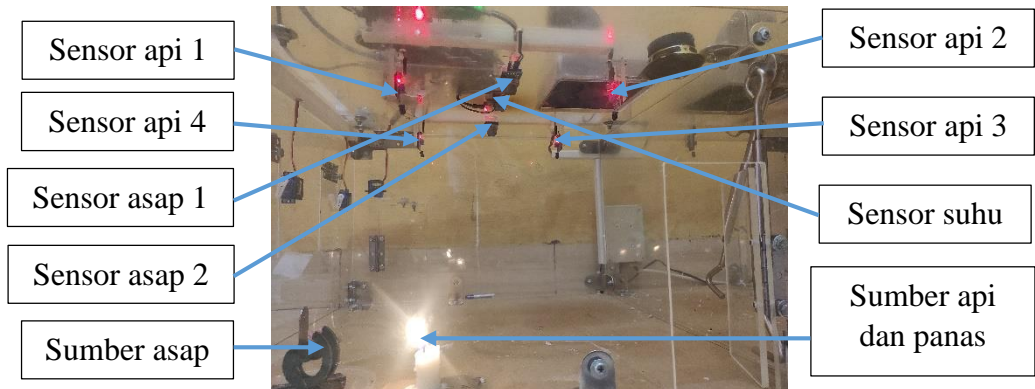
Dari tabel hasil pengujian dapat disimpulkan bahwa ketika terdeteksi kadar asap >110 ppm dan suhu >38°C pada ruangan sementara tidak terdeteksi adanya api maka kondisi tersebut tidak dikatakan kebakaran sehingga semua tindakan penanganan kebakaran yaitu alarm, *exhaust*, motor servo untuk membuka pintu dan jendela serta LED tidak aktif.

4.5.1.7 Pengujian Kontrol Kebakaran dengan *Input* Api, Asap dan Suhu

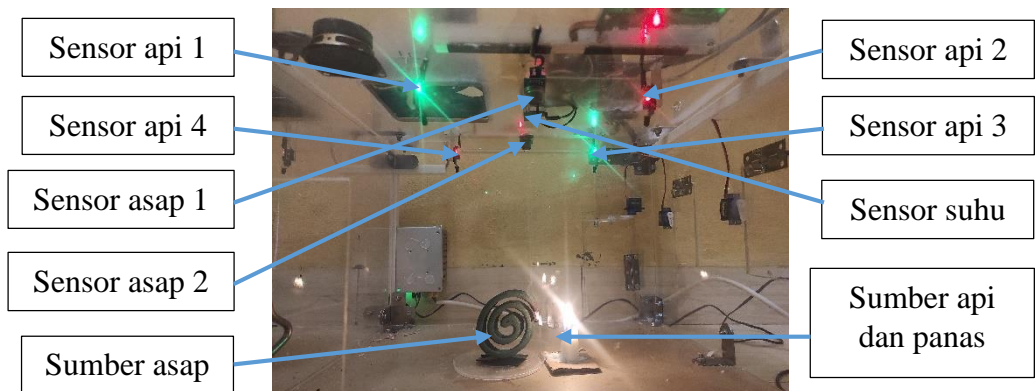
Pengujian ini bertujuan untuk menganalisa kontrol tindakan penanganan kebakaran jika diberikan *input* berupa api, asap dan suhu. Prosedur pengujian dilakukan dengan memberikan *input* api, asap dan suhu pada masing-masing ruangan dengan simulasi api dan suhu panas dari lilin serta asap dari obat nyamuk. Pengujian ini dilakukan sebanyak 2 kali yaitu dengan cara menempatkan lilin dan obat nyamuk di tengah ruangan miniatur dan kedua dengan cara menempatkan lilin pada sudut-sudut ruangan sementara obat nyamuk tetap berada di tengah ruangan. Kemudian lilin dipindahkan ke beberapa posisi dan hitung berapa lama waktu yang dibutuhkan ketiga input mencapai ambang batas kebakaran pada kedua pengujian tersebut menggunakan stopwatch. Kemudian dilakukan pengamatan terhadap pembacaan sensor pada LCD dan *output* alarm, *exhaust*, pintu, jendela dan LED menjadi aktif atau tidak. Hasil pengujian ini kemudian dicatat pada Tabel 4.16 untuk ruang 1 dan Tabel 4.17 untuk ruang 2.



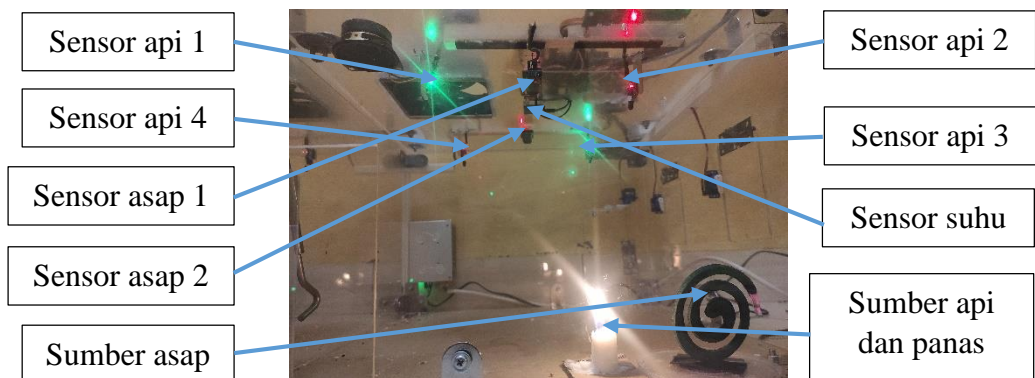
Gambar 4. 74 Pengujian ke-1 Api, Asap dan Suhu pada Ruang 1



Gambar 4. 75 Pengujian ke-2 Api, Asap dan Suhu pada Ruang 1

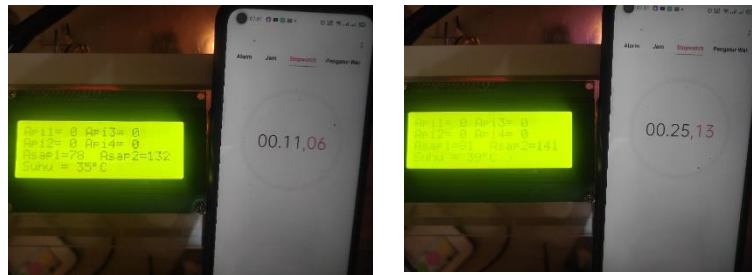


Gambar 4. 76 Pengujian ke-1 Api, Asap dan Suhu pada Ruang 2

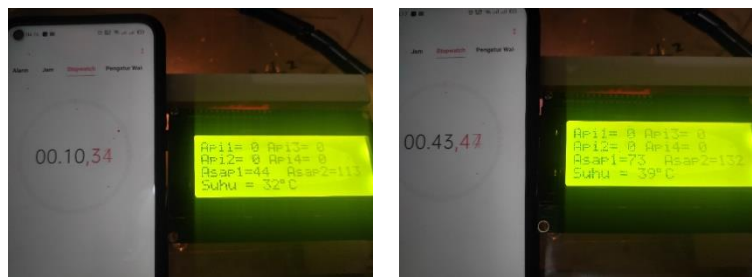


Gambar 4. 77 Pengujian ke-2 Api, Asap dan Suhu pada Ruang 2

Hasil pembacaan pada LCD untuk pengujian 1 yaitu lilin dan obat nyamuk diletakkan ditengah ruangan adalah sebagai berikut.

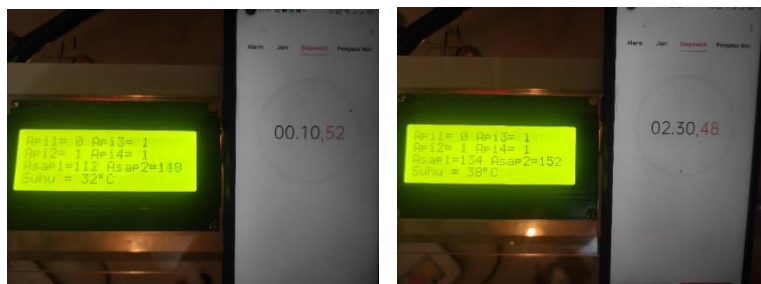


Gambar 4. 78 Tampilan LCD Pengujian ke-1 Ruang 1

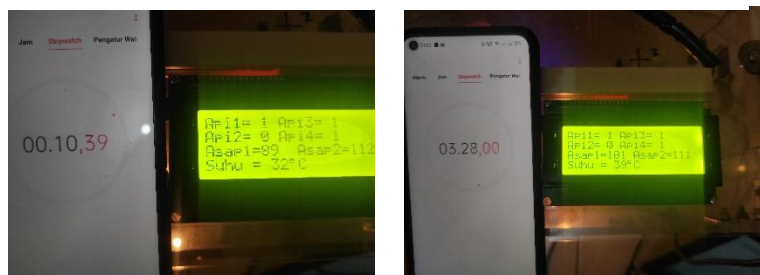


Gambar 4. 79 Tampilan LCD Pengujian ke-1 Ruang 2

Hasil pembacaan pada LCD untuk pengujian 2 yaitu lilin diletakkan di sudut-sudut ruangan dan obat nyamuk diletakkan ditengah ruangan adalah sebagai berikut.



Gambar 4. 80 Tampilan LCD Pengujian ke-2 Ruang 1



Gambar 4. 81 Tampilan LCD Pengujian ke-2 Ruang 2

Tabel 4. 16 Pengujian ke-1 Ruang 1

<i>Input</i>	Waktu (s)	Pembacaan data sensor api ke- pada LCD				Pembacaan data sensor asap ke- pada LCD (ppm)		Pembacaan data sensor suhu pada LCD (°C)	<i>Output</i> kondisi tindakan penanganan kebakaran				
		1	2	3	4	1	2		Alarm	Exhaust	Pintu	Jendela	LED
Api dari lilin, asap dari obat	11	0	0	0	0	78	132	29	OFF	OFF	CLOSE	CLOSE	OFF
	20	0	0	0	0	77	133	38	OFF	OFF	CLOSE	CLOSE	OFF
nyamuk dan suhu panas dari lilin	25	0	0	0	0	81	141	39	ON	ON	OPEN	OPEN	ON
	95	0	0	0	0	99	110	37	OFF	OFF	CLOSE	CLOSE	OFF
	97	1	1	1	1	88	97	37	OFF	OFF	CLOSE	CLOSE	OFF

Tabel 4. 17 Pengujian ke-2 Ruang 1

<i>Input</i>	Waktu (s)	Pembacaan data sensor api ke- pada LCD				Pembacaan data sensor asap ke- pada LCD (ppm)		Pembacaan data sensor suhu pada LCD (°C)	<i>Output</i> kondisi tindakan penanganan kebakaran				
		1	2	3	4	1	2		Alarm	Exhaust	Pintu	Jendela	LED
Api dari lilin, asap dari obat	10	0	1	1	1	112	140	32	OFF	OFF	CLOSE	CLOSE	OFF
	50	0	1	1	1	135	160	34	OFF	OFF	CLOSE	CLOSE	OFF
nyamuk dan suhu panas dari lilin	60	0	1	1	1	134	159	34	OFF	OFF	CLOSE	CLOSE	OFF
	70	0	1	1	1	131	161	35	OFF	OFF	CLOSE	CLOSE	OFF
	110	0	0	1	1	134	160	36	OFF	OFF	CLOSE	CLOSE	OFF
	150	0	1	1	1	96	105	38	OFF	OFF	CLOSE	CLOSE	OFF
	170	0	0	1	1	132	151	39	ON	ON	OPEN	OPEN	ON
	246	1	1	1	1	103	108	36	OFF	OFF	CLOSE	CLOSE	OFF

Tabel 4. 18 Pengujian ke-1 Ruang 2

<i>Input</i>	Waktu (s)	Pembacaan data sensor api ke- pada LCD				Pembacaan data sensor asap ke- pada LCD (ppm)		Pembacaan data sensor suhu pada LCD (°C)	<i>Output</i> kondisi tindakan penanganan kebakaran				
		1	2	3	4	1	2		Alarm	Exhaust	Pintu	Jendela	LED
Api dari lilin, asap dari obat nyamuk dan suhu panas dari lilin	0	1	1	1	1	6	3	29	OFF	OFF	CLOSE	CLOSE	OFF
	10	0	0	0	0	44	113	32	OFF	OFF	CLOSE	CLOSE	OFF
	20	0	0	0	0	66	133	34	OFF	OFF	CLOSE	CLOSE	OFF
	30	0	0	0	0	68	136	37	OFF	OFF	CLOSE	CLOSE	OFF
	40	0	0	0	0	72	135	38	OFF	OFF	CLOSE	CLOSE	OFF
	43	0	0	0	0	73	132	39	ON	ON	OPEN	OPEN	ON
	122	0	0	0	0	28	25	37	OFF	OFF	CLOSE	CLOSE	OFF

Tabel 4. 19 Pengujian ke-2 Ruang 2

<i>Input</i>	Waktu (s)	Pembacaan data sensor api ke- pada LCD				Pembacaan data sensor asap ke- pada LCD (ppm)		Pembacaan data sensor suhu pada LCD (°C)	<i>Output</i> kondisi tindakan penanganan kebakaran				
		1	2	3	4	1	2		Alarm	Exhaust	Pintu	Jendela	LED
Api dari lilin, asap dari obat nyamuk dan suhu panas dari lilin	10	1	0	1	1	89	112	32	OFF	OFF	CLOSE	CLOSE	OFF
	20	1	0	1	1	95	114	33	OFF	OFF	CLOSE	CLOSE	OFF
	50	1	0	1	1	96	114	34	OFF	OFF	CLOSE	CLOSE	OFF
	100	1	0	1	1	92	113	36	OFF	OFF	CLOSE	CLOSE	OFF
	120	1	0	1	1	100	117	36	OFF	OFF	CLOSE	CLOSE	OFF
	200	1	0	1	1	96	105	38	OFF	OFF	CLOSE	CLOSE	OFF
	211	0	0	0	0	100	169	39	ON	ON	OPEN	OPEN	ON
	254	1	1	1	1	69	60	37	OFF	OFF	CLOSE	CLOSE	OFF

Keterangan:

1. Jumlah sensor api yang digunakan dalam pengujian: sensor api sebanyak 4 buah, sensor asap sebanyak 2 buah dan sensor suhu sebanyak 1 buah.
2. Sensor api berlogika *low* atau 0 ketika mendeteksi adanya api dan berlogika *high* atau 1 ketika tidak mendeteksi adanya api.
3. Nilai ambang batas kebakaran: terdeteksi adanya api, asap >110 ppm dan suhu >38°C.

Kesimpulan:

Dari tabel hasil pengujian dapat disimpulkan bahwa ketika *input* data sensor yaitu api, suhu dan asap tidak terpenuhi maka kondisi tersebut tidak dikatakan kebakaran sehingga semua tindakan penanganan kebakaran tidak aktif. Namun apabila semua syarat *input* data sensor terpenuhi dan mencapai nilai ambang batas yang ditentukan yaitu ada api asap >110 ppm dan suhu >38°C maka kondisi tersebut dikatakan kebakaran sehingga semua tindakan penanganan kebakaran berupa alarm, *exhaust*, pintu, jendela, dan LED akan aktif.

4.5.2 Pengujian *Prototype* Pendeteksi Kebakaran Berbasis IoT

Pengujian ini bertujuan untuk menganalisa koneksi antara Arduino Mega 2560 dengan *smartphone* apakah sistem *monitoring* data sensor dan kontrol gedung dari bahaya kebakaran berbasis IoT sudah berfungsi dengan baik.

4.5.2.1 Hasil pengujian Alat Berbasis IoT pada Ruang 1

Pengujian alat berbasis IoT yang akan dilakukan terdiri dari *monitoring* perubahan data sensor melalui *smartphone*, kontrol tindakan penanganan kebakaran melalui *smartphone*, dan pengiriman notifikasi ke *smartphone* pada ruang 1.

1. *Monitoring* perubahan data sensor melalui *smartphone*

Pengujian terhadap *monitoring* data sensor melalui *smartphone* dilakukan dengan cara membandingkan data sensor yang tampil pada serial monitor Arduino IDE dengan data sensor yang tampil pada *smartphone*.

pembacaan sensor
Api 1: 0
Api 2: 1
Api 3: 1
Api 4: 1
Asap 1: 2
Asap 2: 0
Suhu 1: 31
Semua action OFF

MONITORING DATA SENSOR DAN KONTROL GEDUNG

Room1

FLAME SENSOR

ON OFF OFF OFF
SENSOR 1 SENSOR 2 SENSOR 3 SENSOR 4

SENSOR MQ-9

2 0
SENSOR 1 SENSOR 2

SENSOR DHT-11

31
SENSOR 1

OFF OFF OFF OFF OFF
Alarm Jendela Pintu Exhaust Air

LOG OFF NEXT

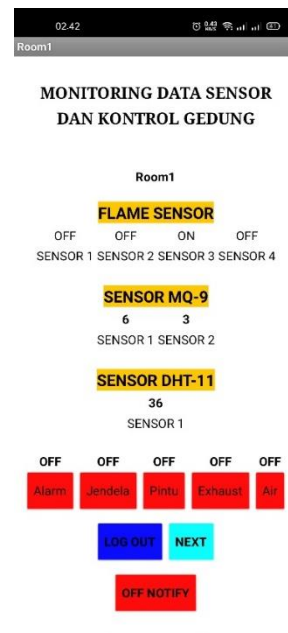
OFF NOTIFY

(a)

(b)

Gambar 4. 82 Tampilan 1 *Monitoring Data Sensor Ruang 1*: (a) pada Serial Monitor Arduino IDE; dan (b) pada *Smartphone*

```
COM6
pembacaan sensor
Api 1: 1
Api 2: 1
Api 3: 0
Api 4: 1
Asap 1: 6
Asap 2: 3
Suhu 1: 36
Semua action OFF
```



(a)

(b)

Gambar 4. 83 Tampilan 2 *Monitoring Data Sensor Ruang 1*: (a) pada Serial Monitor Arduino IDE; dan (b) pada *Smartphone*

```
COM6
pembacaan sensor
Api 1: 1
Api 2: 1
Api 3: 1
Api 4: 0
Asap 1: 34
Asap 2: 104
```



(a)

(b)

Gambar 4. 84 Tampilan 3 *Monitoring Data Sensor Ruang 1*: (a) pada Serial Monitor Arduino IDE; dan (b) pada *Smartphone*

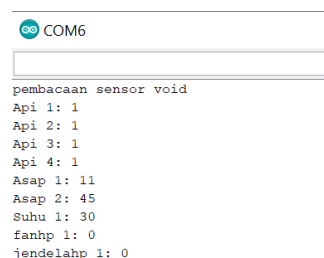
Kesimpulan:

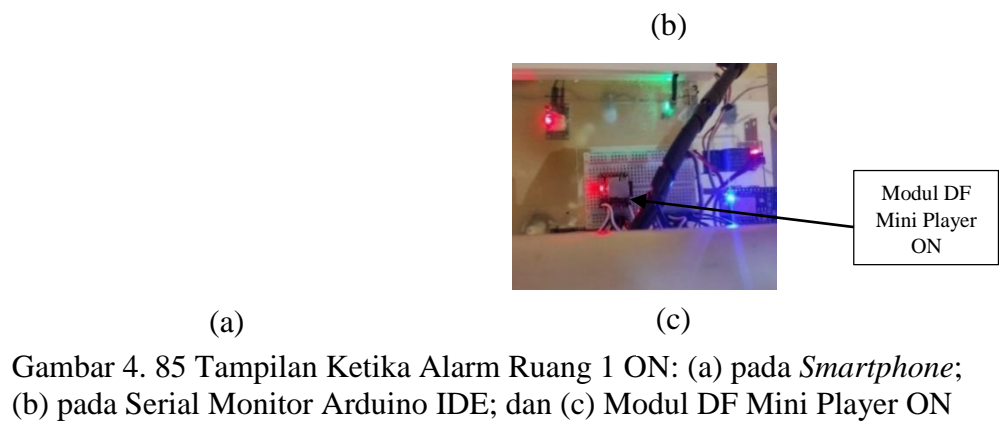
Dari 3 gambar di atas dapat disimpulkan bahwa *monitoring* data sensor api, sensor asap dan sensor suhu yang tampil pada serial monitor Arduino IDE sama dengan tampilan pada *smartphone*. Hal ini menunjukkan bahwa koneksi antara Arduino dengan *smartphone* pada ruang 1 sudah berfungsi dengan baik sehingga pengguna dapat memantau perubahan data sensor dari jarak jauh melalui *smartphone*.

2. Kontrol tindakan penanganan kebakaran melalui *smartphone*

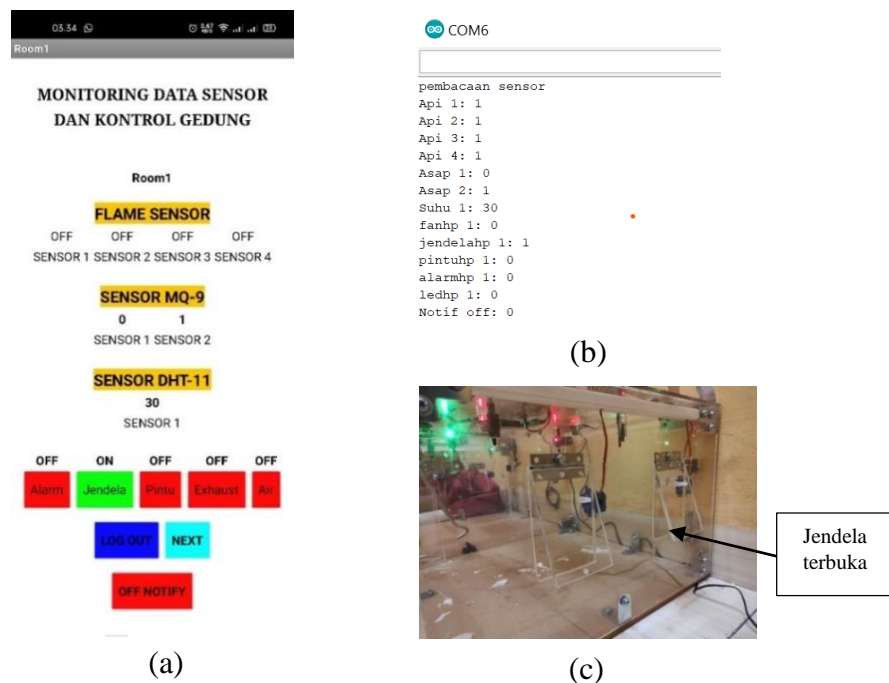
Pengujian terhadap kontrol tindakan penanganan kebakaran berupa tindakan mengaktifkan alarm, *exhaust*, pintu, jendela dan LED dilakukan dengan cara menekan tombol pada *smartphone*.

- Ketika tombol alarm pada *smartphone* ditekan

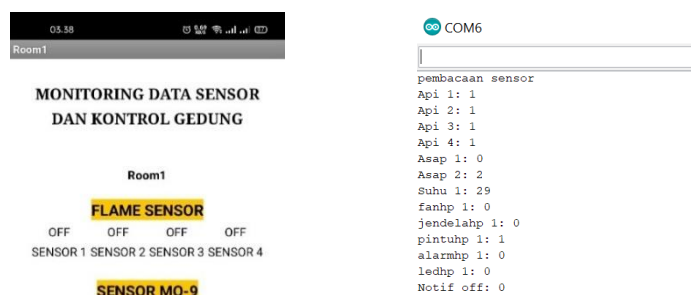


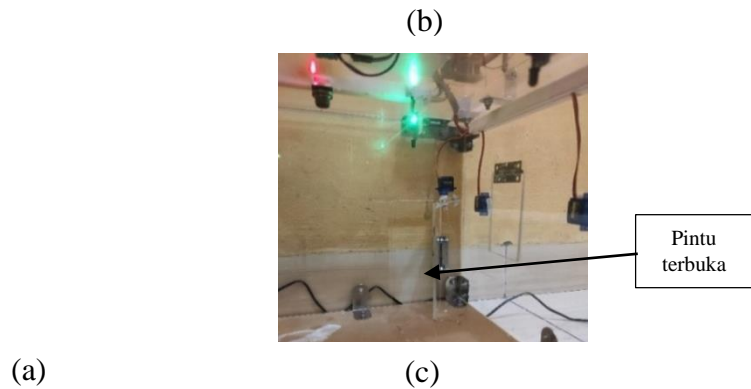


- Ketika tombol jendela pada *smartphone* ditekan



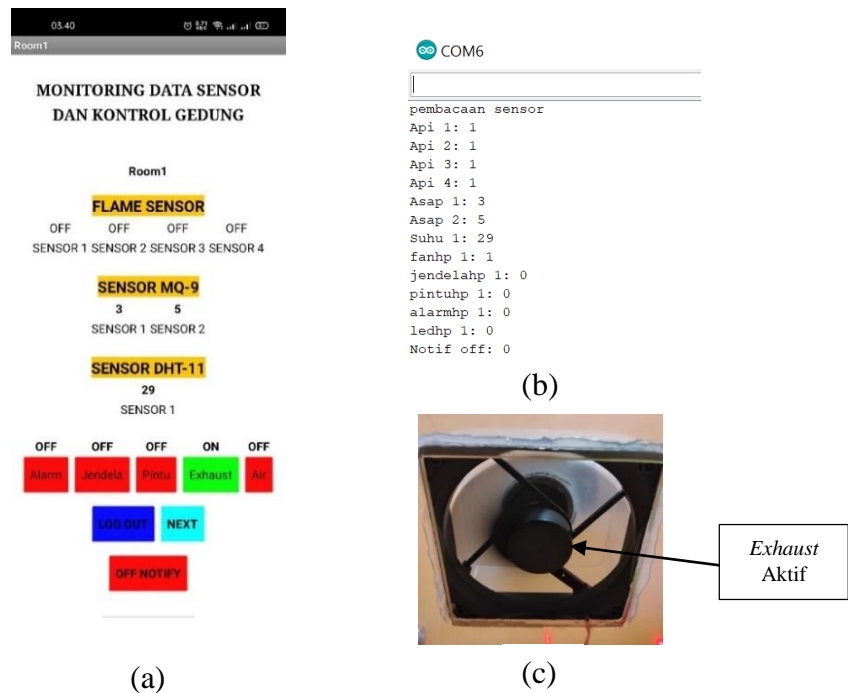
- Ketika tombol pintu pada *smartphone* ditekan





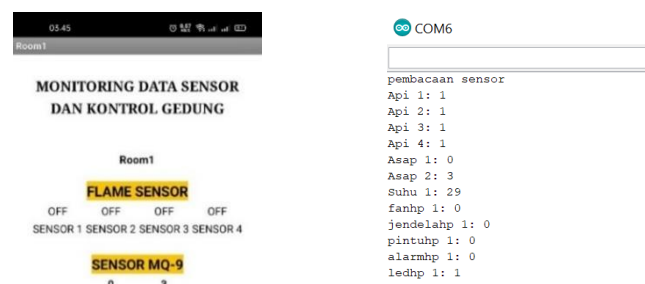
Gambar 4. 87 Tampilan Ketika Pintu Ruang 1 ON: (a) pada *Smartphone*; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi Pintu Terbuka

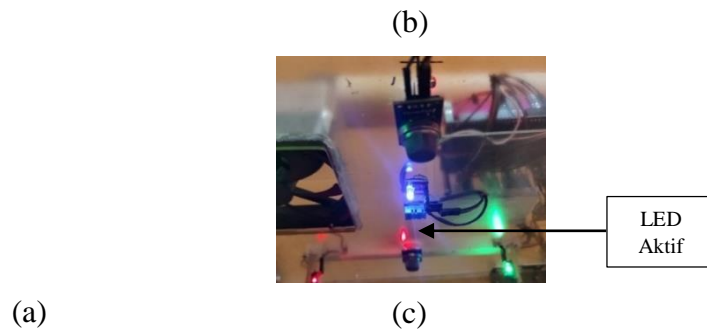
- Ketika tombol *exhaust* pada *smartphone* ditekan



Gambar 4. 88 Tampilan Ketika *Exhaust* Ruang 1 ON: (a) pada *Smartphone*; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi *Exhaust* Aktif

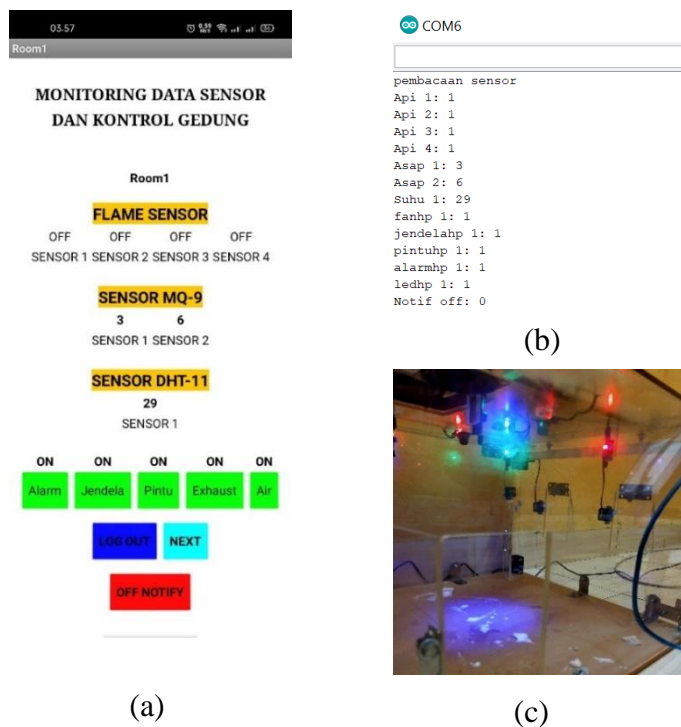
- Ketika tombol air pada *smartphone* ditekan





Gambar 4. 89 Tampilan Ketika LED Ruang 1 ON: (a) pada *Smartphone*; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi LED Aktif

- Ketika semua tombol tindakan penanganan kebakaran pada *smartphone* ditekan



Gambar 4. 90 Tampilan Ketika Semua *Action* Ruang 1 ON: (a) pada *Smartphone*; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi Semua *Action* Aktif

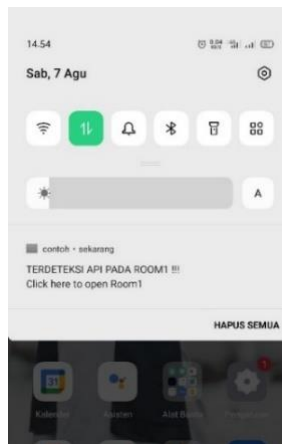
Kes.....

Dari 6 gambar di atas dapat disimpulkan bahwa ketika tombol *action* pada *smartphone* ditekan maka *action* pada ruang 1 akan aktif dan melakukan eksekusi. Kemudian pada tombol *action* yang ditekan akan berubah kondisi dari OFF menjadi ON serta warna tombol juga berubah dari merah menjadi hijau yang menandakan bahwa *action* dalam kondisi aktif. Hal ini menunjukkan bahwa koneksi antara Arduino IDE dengan *smartphone* sudah berfungsi dengan baik sehingga pengguna dapat mengontrol tindakan penanganan kebakaran berupa mengaktifkan alarm, *exhaust*, pintu, jendela dan LED dari jarak jauh melalui *smartphone*.

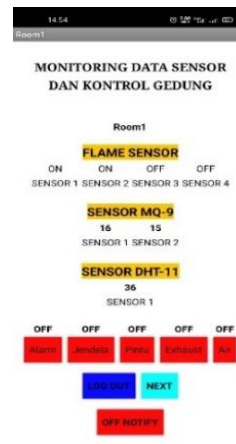
3. Pengiriman Notifikasi ke *Smartphone*

Pengujian terhadap pengiriman notifikasi ke *smartphone* dilakukan dengan cara menguji jika hanya ada api, atau asap, atau suhu atau api dan asap dan suhu di dalam ruang 1 *prototype* apakah ada notifikasi yang dikirimkan ke *smartphone*.

- Ketika terdeteksi adanya api



(a)

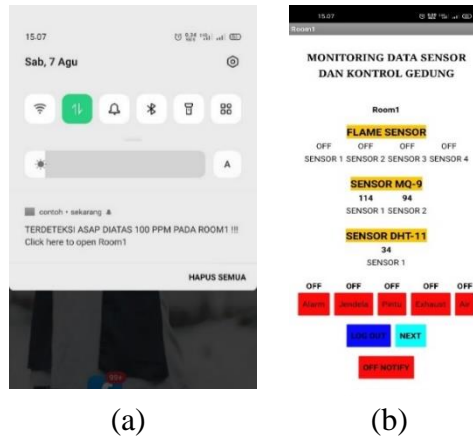


(b)

Gambar 4. 91 (a) Notifikasi Terdeteksi Api pada *Smartphone*; (b) Tampilan Data Sensor pada *Room 1*

Dari hasil pengujian di atas ada notifikasi yang dikirimkan ke *smartphone* bahwa terdeteksi adanya api pada *Room 1*. Ketika notifikasi diklik maka akan masuk ke tampilan *Room 1* pada aplikasi *smartphone* seperti yang ditunjukkan Gambar 4.91 (b). Pada tampilan tersebut diketahui bahwa Sensor 1 dan Sensor 2 dari *flame sensor* mendeteksi adanya api pada *Room 1*.

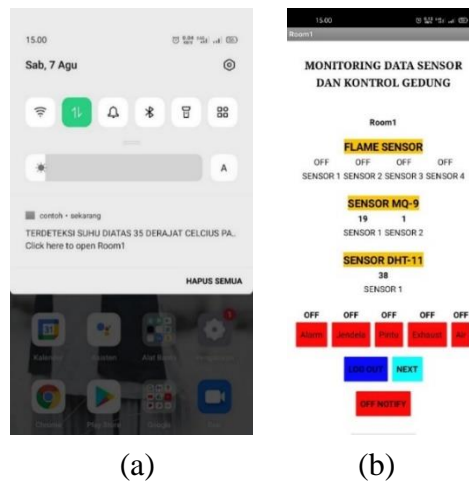
- Ketika terdeteksi adanya asap >100 ppm



Gambar 4. 92 (a) Notifikasi Terdeteksi Asap >100 ppm pada *Smartphone*; (b) Tampilan Data Sensor pada *Room 1*

Dari hasil pengujian di atas ada notifikasi yang dikirimkan ke *smartphone* bahwa terdeteksi adanya asap >100 ppm pada *Room 1*. Ketika notifikasi diklik maka akan masuk ke tampilan *Room 1* pada aplikasi *smartphone* seperti yang ditunjukkan Gambar 4.92 (b). Pada tampilan tersebut diketahui bahwa Sensor 1 dari sensor MQ-9 mendeteksi adanya asap >100 ppm pada *Room 1*.

- Ketika terdeteksi adanya suhu >35°C

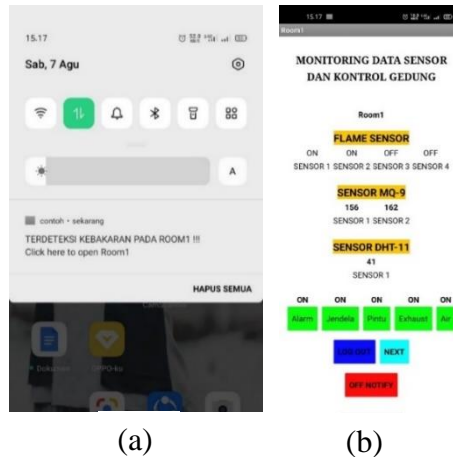


Gambar 4. 93 (a) Notifikasi Terdeteksi Suhu >35°C pada *Smartphone*; (b) Tampilan Data Sensor pada *Room 1*

Dari hasil pengujian di atas ada notifikasi yang dikirimkan ke *smartphone* bahwa terdeteksi adanya suhu >35°C pada *Room 1*. Ketika notifikasi diklik maka akan masuk ke tampilan *Room 1* pada aplikasi *smartphone* seperti yang ditunjukkan

Gambar 4.93 (b). Pada tampilan tersebut diketahui bahwa Sensor 1 dari sensor DHT-11 mendeteksi adanya suhu $>35^{\circ}\text{C}$ pada *Room 1*.

- Ketika terdeteksi adanya api, asap >110 ppm dan suhu $>38^{\circ}\text{C}$



Gambar 4. 94 (a) Notifikasi Terdeteksi Api dan Asap >110 ppm dan Suhu $>38^{\circ}\text{C}$ pada *Smartphone*; (b) Tampilan Data Sensor pada *Room 1*

Dari hasil pengujian di atas ada notifikasi yang dikirimkan ke *smartphone* bahwa terdeteksi kebakaran pada *Room 1*. Ketika notifikasi diklik maka akan masuk ke tampilan *Room 1* pada aplikasi *smartphone* seperti yang ditunjukkan Gambar 4.94 (b). Pada tampilan tersebut diketahui bahwa sensor 1 dan sensor 2 dari *flame sensor* mendeteksi adanya api, kedua sensor MQ-9 mendeteksi kadar asap >110 ppm, dan sensor DHT-11 mendeteksi peningkatan suhu $>38^{\circ}\text{C}$ sehingga semua nilai ambang batas kebakaran terpenuhi pada *Room 1*.

Kesimpulan:

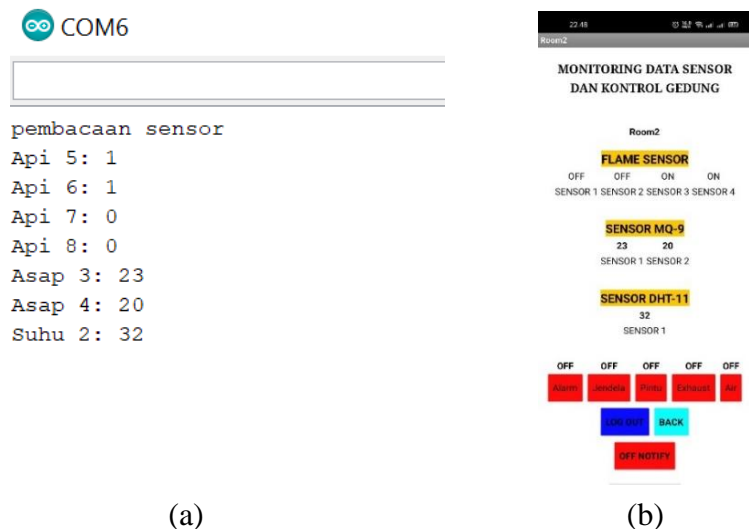
Dari 4 kondisi tersebut dapat disimpulkan bahwa notifikasi akan dikirimkan ke *smartphone* ketika terdeteksi hanya ada api, atau kadar asap >100 ppm, atau suhu $>35^{\circ}\text{C}$, atau ada api dan asap >110 ppm dan suhu $>38^{\circ}\text{C}$ secara bersamaan. Notifikasi ini akan terus dikirimkan selama keempat kondisi masih terpenuhi. Hal ini menunjukkan bahwa koneksi antara Arduino dengan *smartphone* sudah berfungsi dengan baik sehingga pengguna dapat mengetahui telah terdeteksi kebakaran melalui notifikasi yang dikirim ke *smartphone*.

4.5.2.2 Hasil Pengujian Alat Berbasis IoT pada Ruang 2

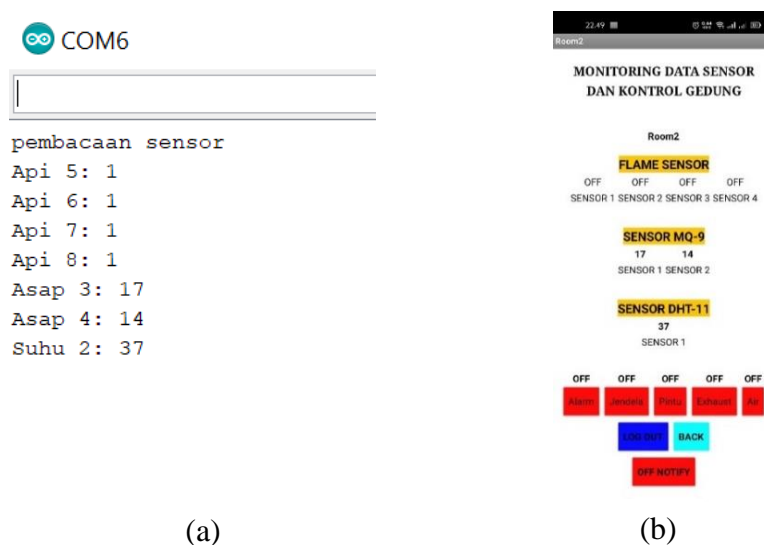
Pengujian alat berbasis IoT yang akan dilakukan terdiri dari *monitoring* perubahan data sensor melalui *smartphone*, kontrol tindakan penanganan kebakaran melalui *smartphone*, dan pengiriman notifikasi ke *smartphone* pada ruang 2.

1. *Monitoring* perubahan data sensor melalui *smartphone*

Pengujian terhadap *monitoring* data sensor melalui *smartphone* dilakukan dengan cara membandingkan data sensor yang tampil pada serial monitor Arduino IDE dengan data sensor yang tampil pada *smartphone*.



Gambar 4. 95 Tampilan 1 *Monitoring* Data Sensor Ruang 2: (a) pada Serial Monitor Arduino IDE; dan (b) pada *Smartphone*



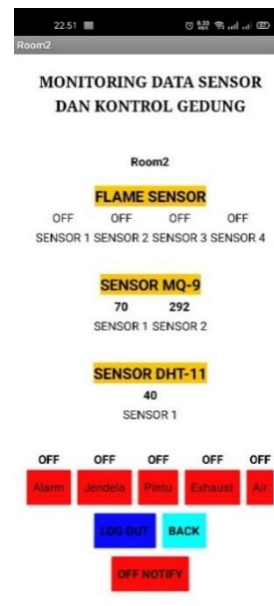
Gambar 4. 96 Tampilan 2 *Monitoring* Data Sensor Ruang 2: (a) pada Serial Monitor Arduino IDE; dan (b) pada *Smartphone*

```

COM6
pembacaan sensor
Api 5: 1
Api 6: 1
Api 7: 1
Api 8: 1
Asap 3: 70
Asap 4: 292
Suhu 2: 40

```

(a)



(b)

Gambar 4. 97 Tampilan 3 *Monitoring* Data Sensor Ruang 2: (a) pada Serial Monitor Arduino IDE; dan (b) pada *Smartphone*

Kesimpulan:

Dari 3 gambar di atas dapat disimpulkan bahwa *monitoring* data sensor api, sensor asap dan sensor suhu yang tampil pada serial monitor Arduino IDE sama dengan tampilan pada *smartphone*. Hal ini menunjukkan bahwa koneksi antara Arduino dengan *smartphone* pada ruang 2 sudah berfungsi dengan baik sehingga pengguna dapat memantau perubahan data sensor dari jarak jauh melalui *smartphone*.

2. Kontrol tindakan penanganan kebakaran melalui *smartphone*

Pengujian terhadap kontrol tindakan penanganan kebakaran berupa tindakan mengaktifkan alarm, *exhaust*, pintu, jendela dan LED dilakukan dengan cara menekan tombol pada *smartphone*.

- Ketika tombol alarm pada *smartphone* ditekan

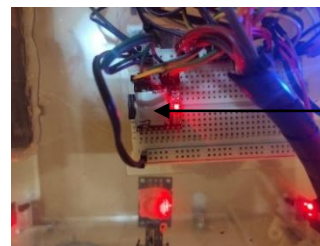


(a)

```

COM6
pembacaan sensor void
Api 5: 1
Api 6: 1
Api 7: 1
Api 8: 1
Asap 3: 10
Asap 4: 8
Suhu 2: 32
fanhp 2: 0
jendelahrp 2: 0
pintuhrp 2: 0
alarmhp 2: 1
ledhp 2: 0
  
```

(b)



(c)

Gambar 4. 98 Tampilan Ketika Alarm Ruang 2 ON: (a) pada *Smartphone*; (b) pada Serial Monitor Arduino IDE; dan (c) Modul DF Mini Player ON

- Ketika tombol jendela pada *smartphone* ditekan

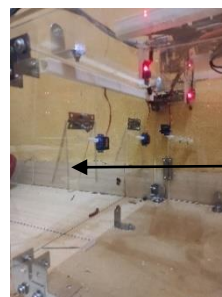


(a)

```

COM6
pembacaan sensor
Api 5: 1
Api 6: 1
Api 7: 1
Api 8: 1
Asap 3: 7
Asap 4: 2
Suhu 2: 32
fanhp 2: 0
jendelahrp 2: 1
pintuhrp 2: 0
alarmhp 2: 0
ledhp 2: 0
  
```

(b)



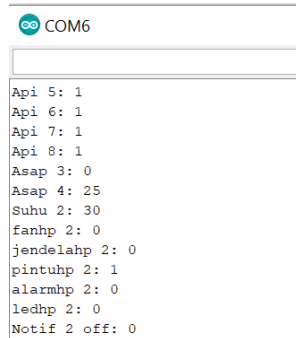
(c)

Gambar 4. 99 Tampilan Ketika Jendela Ruang 2 ON: (a) pada *Smartphone*; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi Jendela Terbuka

- Ketika tombol *pintu* pada *smartphone* ditekan



(a)



(b)



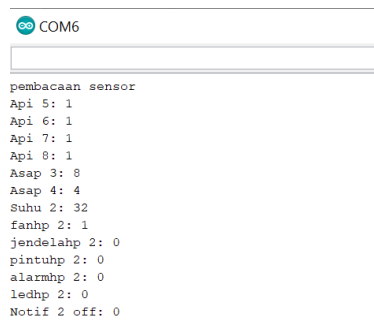
(c)

Gambar 4. 100 Tampilan Ketika Pintu Ruang 2 ON: (a) pada *Smartphone*; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi Pintu Terbuka

- Ketika tombol *exhaust* pada *smartphone* ditekan



(a)



(b)



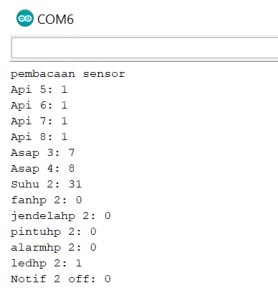
(c)

Gambar 4. 101 Tampilan Ketika *Exhaust* Ruang 2 ON: (a) pada *Smartphone*; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi *Exhaust* Aktif

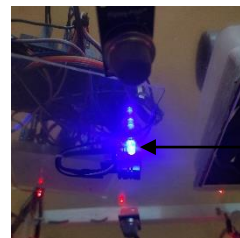
- Ketika tombol air pada *smartphone* ditekan



(a)



(b)

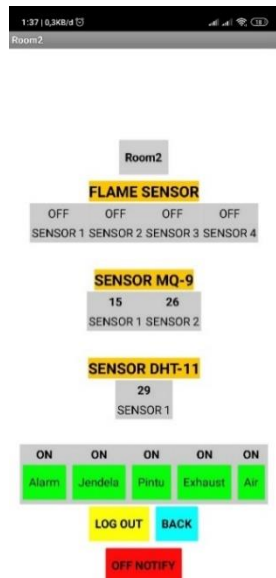


LED Aktif

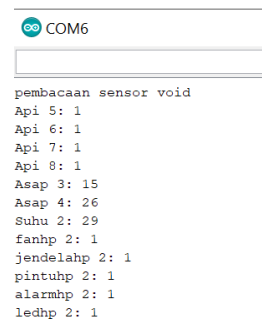
(c)

Gambar 4. 102 Tampilan Ketika LED Ruang 2 ON: (a) pada *Smartphone*; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi LED Aktif

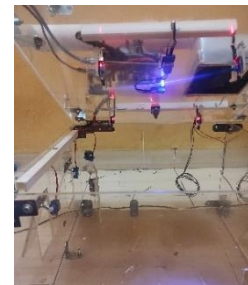
- Ketika semua tombol tindakan penanganan kebakaran pada *smartphone* ditekan



(a)



(b)



(c)

Gambar 4. 103 Tampilan Semua *Action* Ruang 2 ON: (a) pada *Smartphone*; (b) pada Serial Monitor Arduino IDE; dan (c) Kondisi Semua *Action* Aktif

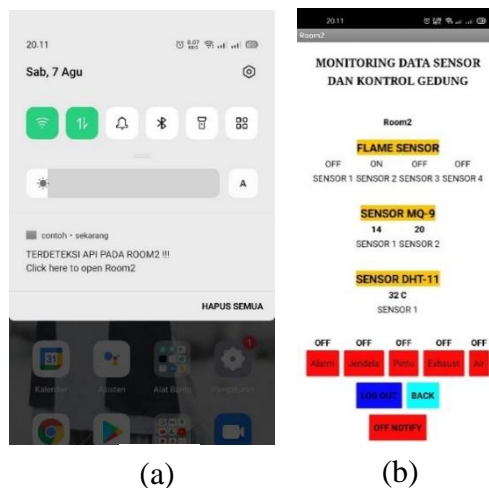
Kesimpulan:

Dari 6 gambar di atas dapat disimpulkan bahwa ketika tombol *action* pada *smartphone* ditekan maka *action* pada ruang 2 akan aktif dan melakukan eksekusi. Kemudian pada tombol *action* yang ditekan akan berubah kondisi dari OFF menjadi ON serta warna tombol juga berubah dari merah menjadi hijau yang menandakan bahwa *action* dalam kondisi aktif. Hal ini menunjukkan bahwa koneksi antara Arduino IDE dengan *smartphone* sudah berfungsi dengan baik sehingga pengguna dapat mengontrol tindakan penanganan kebakaran berupa mengaktifkan alarm, *exhaust*, pintu, jendela dan LED dari jarak jauh melalui *smartphone*.

3. Pengiriman notifikasi ke *smartphone*

Pengujian terhadap pengiriman notifikasi ke *smartphone* dilakukan dengan cara menguji jika hanya ada api, atau asap, atau suhu atau api dan asap dan suhu di dalam ruang 2 *prototype* apakah ada notifikasi yang dikirimkan ke *smartphone*.

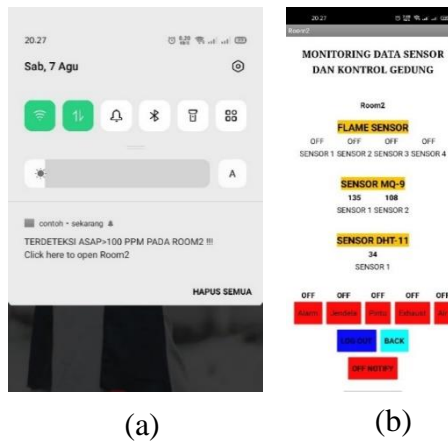
- Ketika terdeteksi adanya api



Gambar 4. 104 (a) Notifikasi Terdeteksi Api pada *Smartphone*; (b) Tampilan Data Sensor pada *Room 2*

Dari hasil pengujian di atas ada notifikasi yang dikirimkan ke *smartphone* bahwa terdeteksi adanya api pada *Room 2*. Ketika notifikasi diklik maka akan masuk ke tampilan *Room 2* pada aplikasi *smartphone* seperti yang ditunjukkan Gambar 4.104 (b). Pada tampilan tersebut diketahui bahwa sensor 2 dari *flame sensor* mendeteksi adanya api pada *Room 2*.

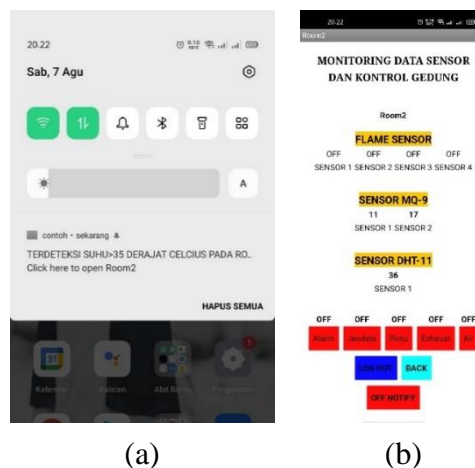
- Ketika terdeteksi adanya asap >100 ppm



Gambar 4. 105 (a) Notifikasi Terdeteksi Asap >100 ppm pada *Smartphone*; (b) Tampilan Data Sensor pada *Room 2*

Dari hasil pengujian di atas ada notifikasi yang dikirimkan ke *smartphone* bahwa terdeteksi adanya asap >100 ppm pada *Room 2*. Ketika notifikasi diklik maka akan masuk ke tampilan *Room 2* pada aplikasi *smartphone* seperti yang ditunjukkan Gambar 4.105 (b). Pada tampilan tersebut diketahui bahwa sensor 1 dan sensor 2 dari sensor MQ-9 mendeteksi adanya asap >100 ppm pada *Room 2*.

- Ketika terdeteksi adanya suhu >35°C

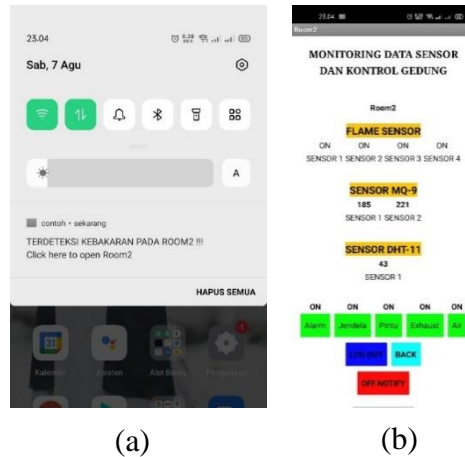


Gambar 4. 106 (a) Notifikasi Terdeteksi Suhu >35°C pada *Smartphone*; (b) Tampilan Data Sensor pada *Room 2*

Dari hasil pengujian di atas ada notifikasi yang dikirimkan ke *smartphone* bahwa terdeteksi adanya suhu >35°C pada *Room 2*. Ketika notifikasi diklik maka akan masuk ke tampilan *Room 2* pada aplikasi *smartphone* seperti yang ditunjukkan

Gambar 4.106 (b). Pada tampilan tersebut diketahui bahwa Sensor 1 dari sensor DHT-11 mendeteksi adanya suhu $>35^{\circ}\text{C}$ pada *Room 2*.

- Ketika terdeteksi adanya api, asap >110 ppm dan suhu $>38^{\circ}\text{C}$



Gambar 4. 107 (a) Notifikasi Terdeteksi Api dan Asap >110 ppm dan Suhu $>38^{\circ}\text{C}$ pada *Smartphone*; (b) Tampilan Data Sensor pada *Room 2*

Dari hasil pengujian di atas ada notifikasi yang dikirimkan ke *smartphone* bahwa terdeteksi kebakaran pada *Room 2*. Ketika notifikasi diklik maka akan masuk ke tampilan *Room 2* pada aplikasi *smartphone* seperti yang ditunjukkan Gambar 4.107 (b). Pada tampilan tersebut diketahui bahwa sensor 1, sensor 2, sensor 3 dan sensor 4 dari *flame sensor* mendeteksi adanya api, sensor 1 dan sensor 2 dari sensor MQ-9 mendeteksi kadar asap >110 ppm, dan sensor 1 dari sensor DHT-11 mendeteksi suhu $>38^{\circ}\text{C}$ sehingga semua nilai ambang batas kebakaran terpenuhi pada *Room 2*.

Kesimpulan:

Dari 4 kondisi tersebut dapat disimpulkan bahwa notifikasi akan dikirimkan ke *smartphone* ketika terdeteksi hanya ada api, atau asap >100 ppm, atau suhu $>35^{\circ}\text{C}$, atau api dan asap >110 ppm dan suhu $>38^{\circ}\text{C}$ secara bersamaan. Notifikasi ini akan terus dikirimkan selama keempat kondisi masih terpenuhi. Hal ini menunjukkan bahwa koneksi antara Arduino dengan *smartphone* sudah berfungsi dengan baik sehingga pengguna dapat mengetahui telah terdeteksi kebakaran melalui notifikasi yang dikirim ke *smartphone*.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil pengujian yang dilakukan menggunakan sensor api, sensor asap (rentang kadar asap 0-241 ppm) dan sensor suhu (rentang suhu 29°C-53°C) diperoleh kesimpulan sebagai berikut.

1. Apabila sensor api mendeteksi api dari lilin atau sensor asap mendeteksi kadar asap >100 ppm atau sensor suhu mendeteksi suhu >35°C maka akan dikirimkan notifikasi berupa informasi adanya api, kadar asap dan nilai suhu yang terbaca oleh sensor ke *smartphone*.
2. Ketika sensor api mendeteksi api dari lilin dan sensor asap mendeteksi kadar asap >110 ppm dan sensor suhu mendeteksi suhu >38°C maka sistem kontrol secara otomatis akan mengaktifkan tindakan penanganan kebakaran dan mengirimkan notifikasi kebakaran ke *smartphone*.
3. Perubahan data dari sensor api, sensor asap dan sensor suhu dapat di-*monitoring* dari jarak jauh melalui *smartphone* (aplikasi MIT App Inventor) selama NodeMCU ESP8266 terhubung dengan jaringan.
4. Sistem kontrol berupa tindakan mengaktifkan alarm, *exhaust*, motor servo untuk membuka pintu dan jendela, serta LED dapat dilakukan secara manual oleh pengguna melalui *smartphone* dari jarak jauh selama NodeMCU ESP8266 masih terhubung dengan jaringan.
5. Pengiriman data dari Arduino Mega 2560 ke *smartphone* dan sebaliknya mengalami rata-rata *delay* selama 4 detik karena harus melalui NodeMCU ESP8266 dan *firebase* agar Arduino Mega 2560 dan *smartphone* dapat terhubung.

5.2 Saran

Proyek akhir ini masih terdapat beberapa kekurangan yang dapat menjadi saran untuk pengembangan proyek akhir ke depannya, yaitu:

1. Saat ini alat proyek akhir menggunakan aplikasi MIT App Inventor untuk *monitoring* data sensor dan kontrol tindakan penanganan kebakaran serta pengiriman notifikasi saat terdeteksi kebakaran. Namun, aplikasi ini hanya bisa mengirimkan notifikasi dengan syarat aplikasi harus tetap berjalan di latar belakang *smartphone*. Untuk memaksimalkan fungsi alat ke depannya dapat menggunakan aplikasi lain seperti Android Studio.
2. Pengiriman data dari Arduino Mega 2560 ke *smartphone* dan sebaliknya mengalami *delay* karena program pada Arduino yang terlalu banyak. Untuk pengiriman data yang lebih baik maka perlu dilakukan perbaikan pada program untuk mengurangi *delay* tersebut.

DAFTAR PUSTAKA

- [1] R. A. Sowah, K. Apeadu, F. Gatsi, K. O. Ampadu and B. S. Mensah, "Hardware Module Design and Software Implementation of Multisensor Fire Detection and Notification System Using Fuzzy Logic and Convolutional Neural Networks (CNNs)," *Journal of Engineering (United Kingdom)*, pp. 1-16, 2020.
- [2] L. Yuwanto, "Mengenal Bencana Kebakaran," Fakultas Psikologi Universitas Surabaya, 26 Agustus 2019. [Online]. Available: https://ubaya.ac.id/2018/content/articles_detail/281/MengenalBencanaKebakaran.html. [Accessed 17 Maret 2021].
- [3] Anizar, *Teknik Keselamatan dan Kesehatan Kerja di Industri*, Yogyakarta: Graha Ilmu, 2012.
- [4] BNPB, "Data Bencana Indonesia," 2017. [Online]. Available: <http://dibi.bnpb.go.id>. [Accessed 16 Maret 2021].
- [5] D. H. Saputra, N. Nabilah, H. I. Islam, G. M. Pradipta, S. S. Atsaurri, A. Kurniawan and A. A. I. Heriyanto Syafutra, "Pembuatan Model Pendeteksi Api Berbasis Arduino Uno dengan Keluaran SMS Gateway," *Prosiding Seminar Nasional Fisika (E-Journal) SNF2016*, vol. V, pp. 103-108, 2016.
- [6] I. D. S. M. G. Rika Sri Rizki, "Sistem Deteksi Kebakaran pada Gedung Berbasis Programmable Logic Controller (PLC)," *Jurnal Online Teknik Elektro*, vol. 2, no. 3, pp. 99-104, 2017.
- [7] C. Grant, A. Hamins and N. B. e. al, "Research Roadmap for Smart Fire Fighting," 2015.
- [8] Saifullana and J. W. Simatupang, "Sistem Pendeteksi Kebakaran Rumah Terintegrasi Smartphone dan Aplikasi Online," *Journal of Electrical and Electronics*, vol. 6, no. 2, pp. 91-98, 2018.

- [9] M. Imamuddin and Zulwisli, "Sistem Alarm dan Monitoring Kebakaran Rumah Berbasis NodeMCU dengan Komunikasi Android," *Jurnal Vokasional Teknik Elektronika dan Informatika*, vol. 7, no. 2, 2019.
- [10] C. G. Simbolon, M. Ir. Ahmad Tri Hanuranto and S. M. Atik Novianti, "Desain dan Implementasi Prototipe Pendeteksi Dini Kebakaran Gedung Menggunakan Algoritma Fuzzy Logic Berbasis Internet of Things (IoT)," *e-Proceeding of Engineering*, vol. 7, no. 2, pp. 2355-9365, 2020.
- [11] ILO, Manajemen Resiko Kebakaran, Jakarta: International Labour Organization 2018, 2018.
- [12] Chase, "What is the Temperature of Fire?," 2021. [Online]. Available: <https://firefighterinsider.com/temperature-of-fire/>. [Accessed 23 Juli 2021].
- [13] W. Shi, J. Ji, J. Sun, S. Lo, L. Li and X. Yuan, "Experimental Study on the Characteristics of Temperature Field of Fire Room under Stack Effect in a Select High-rise Building Model," *Fire Safety Science*, vol. 11, pp. 419-431, 2014.
- [14] T. Waterman, "Room Flashover-criteria and Synthesis," *Fire Technology*, vol. 4, pp. 25-31, 1968.
- [15] M. S. Blais, K. Carpenter and K. Fernandez, "Comparative Room Burn Study of Furnished Rooms from the United Kingdom, France and the United States," *Fire Technology*, vol. 56, pp. 489-514, 2020.
- [16] Kusnandar, N. K. H. Dharmi and D. A. Pratika, "Rancang Bangun Prototipe Pendeteksi Kebakaran Menggunakan Konsep Internet-of Things," *Jurnal Teknik: Media Pengembangan Ilmu dan Aplikasi Teknik*, vol. 18, no. 01, pp. 17-26, 2019.
- [17] B. Panjaitan and R. R. Mulyadi, "Rancang Bangun Sistem Deteksi Kebakaran pada Rumah Berbasis IoT," *Jurnal Ilmiah Fakultas Teknik LIMIT'S*, vol. 16, no. 20, 2020.

- [18] A. R. Putra, T. Novianti and T. Haryanti, "Prototipe Sistem Pendeteksi Kebakaran dan Kontroling Lampu Berbasis Internet of Things," *Jurnal Ilmiah Computing Insight*, vol. 2, no. 2, pp. 60-68, 2020.
- [19] D. Yendri, Wildian and A. Tiffany, "Perancangan Sistem Pendeteksi Kebakaran Rumah Penduduk pada Daerah Perkotaan Berbasis Mikrokontroler," *Seminar Nasional Sains dan Teknologi*, pp. 1-2, 2017.
- [20] G. Cerdas, "Flame Sensor," 2014. [Online]. Available: <http://www.geraicerdas.com/sensor/light/flame-sensor-v2-detail>. [Accessed 21 Maret 2021].
- [21] D. P. Listrik, "Pengertian dan Prinsip Kerja Flame Detector," 8 Februari 2018. [Online]. Available: <https://www.duniapembangkitlistrik.com/2018/02/pengertian-dan-prinsip-kerja-flame.html>. [Accessed 13 Juni 2021].
- [22] H. Electronics, "Gas Sensor. MQ-9 Datasheet," 2014. [Online]. Available: <https://datasheetspdf.com/pdf/904644/HANWEIELETRONICS/MQ-9/1>. [Accessed 30 April 2021].
- [23] "How to Calibrate and Use MQ9 Gas Sensor," 10 April 2019. [Online]. Available: <https://create.arduino.cc/projecthub/electropeak/how-to-calibrate-use-mq9-gas-sensor-w-arduino-e93cb1>. [Accessed 31 Mei 2021].
- [24] Admin_AlfStudio, "Sensor Suhu dan Kelembaban DHT11 dan DHT22," 4 Agustus 2020. [Online]. Available: <https://www.teknikelektro.com/2020/08/sensor-suhu-dan-kelembaban.html>. [Accessed 21 Maret 2021].
- [25] M. H. A. Khairi, "Cara Mengukur Suhu dan Kelembaban dengan DHT11 dan Arduino," 2021. [Online]. Available: <https://www.mahirelektro.com/2020/02/tutorial-menggunakan-sensor-DHT11-pada-Arduino.html>. [Accessed 12 Juni 2021].

- [26] J. Ohoiwutun, "Analisis dan Perancangan Smart Dump Menggunakan Arduino Mega 2560 Rev3 dan GSM SIM900," *Jurnal Electro Luceat*, vol. 4, no. 1, p. 3, 2018.
- [27] L. Elektronika, "ARDUINO MEGA 2560 MIKROKONTROLER ATmega2560," 27 Februari 2017. [Online]. Available: <http://www.labelektronika.com/2017/02/arduino-mega-2560-mikrokontroler.html>. [Accessed 20 Mei 2021].
- [28] Components101, "Servo Motor SG-90 Basics, Pinout, Wire Description, Datasheet," 18 September 2017. [Online]. Available: <https://components101.com/motors/servo-motor-basics-pinout-datasheet>. [Accessed 30 April 2021].
- [29] Sinaupedia, "Pengertian, Fungsi dan Cara Kerja Motor Servo," 18 Januari 2020. [Online]. Available: <https://sinaupedia.com/pengertian-motor-servo/>. [Accessed 11 Mei 2021].
- [30] "MP3 Player Menggunakan DFPlayer Mini dan Arduino," 13 Juni 2019. [Online]. Available: <https://www.nn-digital.com/blog/2019/06/13/mp3-player-menggunakan-dfplayer-mini-dan-arduino/>. [Accessed 8 April 2021].
- [31] C. GLOBE, "What is Light Emitting Diode (LED)," 7 Januari 2019. [Online]. Available: <https://circuitglobe.com/light-emitting-diode-led.html>. [Accessed 21 Juni 2021].
- [32] J. A. Stankovic, "Research Directions for the Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3-9, 2014.
- [33] Myspsolution, 25 Maret 2019. [Online]. Available: <http://www.myspsolution.com/news-events/cara-kerja-konsep-internet-of-things/>. [Accessed 31 Juli 2021].
- [34] I. Ghivena, D. R. Munadi and Sussi, "Rancang Bangun Mikrokontroler Monitoring Penggunaan Air PDAM Berbasis IoT," *e-Proceeding of Engineering*, vol. 7, no. 1, p. 991, 2020.

- [35] Sinauarduino, "Mengenal Arduino Software (IDE)," 2016. [Online]. Available: <https://www.sinauarduino.com/artikel/mengenal-arduino-software-ide/>. [Accessed 23 Juni 2021].
- [36] D. Intern, "Apa itu Firebase? Pengertian, Jenis-Jenis, dan Fungsi Firebase," 2020 November 2020. [Online]. Available: <https://www.dicoding.com/blog/apa-itu-firebase-pengertian-jenis-jenis-dan-fungsi-kegunaannya/>. [Accessed 31 April 2021].
- [37] K. S. Salamah, T. M. Kadarina and Z. Iklima, "Pengenalan MIT Inventor untuk Siswa/i di Wilayah Kembangan Utara," *Jurnal Abdi Masyarakat (JAM)*, vol. 5, no. 2, pp. 5-9, 2020.

LAMPIRAN 1

(Riwayat Hidup Perorangan)

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Eko Firlianto
Tempat, Tanggal Lahir : Rambat, 11 Mei 2000
Alamat Rumah : Jl. Dermaga Desa Rambat
No. Hp : 082266032625
Email : eko.firlianto@gmail.com
Jenis Kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

1. SD Negeri 9 Simpang Teritip Lulus 2012
2. SMP Negeri 2 Simpang Teritip Lulus 2015
3. SMK Negeri 1 Muntok Lulus 2018
4. Politeknik Manufaktur Negeri Bangka Belitung 2018—sekarang

3. Pengalaman Kerja

Praktik Kerja Lapangan di PT Bumi Sawit Sukses Pratama

Sungailiat, 22 Agustus 2021

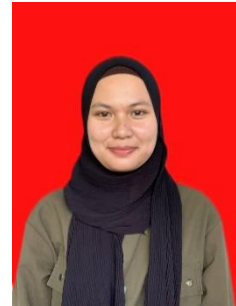


Eko Firlianto

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Dini Anjelina
Tempat, Tanggal Lahir : Belinyu, 12 Oktober 2000
Alamat Rumah : Jl. Yos. Sudarso PMD, Belinyu
No. Hp : 081272247620
Email : dinianjelina12@gmail.com
Jenis Kelamin : Perempuan
Agama : Islam



2. Riwayat Pendidikan

1. SD Negeri 1 Belinyu Lulus 2012
2. SMP Negeri 1 Belinyu Lulus 2015
3. SMA Negeri 1 Belinyu Lulus 2018
4. Politeknik Manufaktur Negeri Bangka Belitung 2018—sekarang

3. Pengalaman Kerja

Praktik Kerja Lapangan di Divisi Pengadaan PT Timah Tbk

Sungailiat, 22 Agustus 2021



Dini Anjelina

LAMPIRAN 2

Program Arduino Mega Ruang 1 :

```
#include <Servo.h>
#include<DFPlayer_Mini_Mp3.h>
#include<SoftwareSerial.h>
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 20, 4);

int apiPin_5 = A0; //pin yg di gunakan
int apiPin_6 = A1;
int apiPin_7 = A2;
int apiPin_8 = A3;
int asapPin_3 = A4;
int asapPin_4 = A5;

int led_pin_2 = 7;

#define DHTPIN 2
DHT dht(DHTPIN, DHT11);
int suhu2;
int apiValue_5 = 0;
int apiValue_6 = 0;
int apiValue_7 = 0;
int apiValue_8 = 0;
int asapValue_3 = 0;
int asapValue_4 = 0;
int pin_servo_4 = 4;
int pin_servo_5 = 5;
int pin_servo_6 = 6;
int fan = 3;
int pos_2 = 90;
Servo myservo_4;
Servo myservo_5;
Servo myservo_6;
SoftwareSerial mySerial(10, 11);
SoftwareSerial Arduino_SoftSerial(12, 13);
int y = 0;
int z = 0;
int u = 0;
int v = 0;
int w = 0;
int r = 0;
int s = 0;
int k = 0;
int dd = 0;
int ee = 0;
int ff = 0;

int fanhp2 = 22;
int fanhp2Value = 0;

int jendelahp2 = 23;
```

```

int jendelahp2Value = 0;

int pintuhp2 = 24;
int pintuhp2Value = 0;

int alarmhp2 = 25;
int alarmhp2Value = 0;

int ledhp2 = 26;
int ledhp2Value = 0;

int notif_off2 = 27;
int notif_offValue2 = 0;
////////////////////////////////////
void setup()
{
  Serial.begin(57600);
  Serial1.begin(57600);
  dht.begin();
  lcd.begin();
  lcd.backlight();

  pinMode( pin_servo_4, OUTPUT);
  myservo_4.attach(pin_servo_4);
  myservo_4.write(90);

  pinMode( pin_servo_5, OUTPUT);
  myservo_5.attach(pin_servo_5);
  myservo_5.write(90);

  pinMode( pin_servo_6, OUTPUT);
  myservo_6.attach(pin_servo_6);
  myservo_6.write(90);

  pinMode( apiPin_5, INPUT);
  pinMode( apiPin_6, INPUT);
  pinMode( apiPin_7, INPUT);
  pinMode( apiPin_8, INPUT);

  pinMode( fan, OUTPUT);
  digitalWrite (fan,1);

  pinMode( led_pin_2, OUTPUT);
  digitalWrite (led_pin_2,0);

  mySerial.begin (9600);
  mp3_set_serial (mySerial);
  delay(1);
  mp3_set_volume (15);

  pinMode( fanhp2, INPUT);
  pinMode( jendelahp2, INPUT);
  pinMode( pintuhp2, INPUT);
  pinMode( alarmhp2, INPUT);
  pinMode( ledhp2, INPUT);
  pinMode( notif_off2, INPUT);

```

```

}

////////////////////////////////////

void loop()
{
  Serial.println("pembacaan sensor");

  fanhp2Value = digitalRead(fanhp2);
  jendelahp2Value = digitalRead(jendelahp2);
  pintuhp2Value = digitalRead(pintuhp2);
  alarmhp2Value = digitalRead(alarmhp2);
  ledhp2Value = digitalRead(ledhp2);
  notif_offValue2 = digitalRead(notif_off2);

  suhu2 = dht.readTemperature();
  apiValue_5 = digitalRead(apiPin_5);
  apiValue_6 = digitalRead(apiPin_6);
  apiValue_7 = digitalRead(apiPin_7);
  apiValue_8 = digitalRead(apiPin_8);
  asapValue_3 = analogRead(asapPin_3);
  asapValue_4 = analogRead(asapPin_4);
  asapValue_3 = asapValue_3 - 150;
  asapValue_4 = asapValue_4 - 120;

  program_lcd();

  Serial.print("Api 5: ");
  Serial.println(apiValue_5);
  Serial.print("Api 6: ");
  Serial.println(apiValue_6);
  Serial.print("Api 7: ");
  Serial.println(apiValue_7);
  Serial.print("Api 8: ");
  Serial.println(apiValue_8);
  Serial.print("Asap 3: ");
  Serial.println(asapValue_3);
  Serial.print("Asap 4: ");
  Serial.println(asapValue_4);
  Serial.print("Suhu 2: ");
  Serial.println(suhu2);

  // Serial.print("R: ");
  // Serial.println(r);
  // Serial.print("Y: ");
  // Serial.println(y);
  // Serial.print("Z: ");
  // Serial.println(z);
  // Serial.print("U: ");
  // Serial.println(u);
  // Serial.print("V: ");
  // Serial.println(v);
  // Serial.print("W: ");
  // Serial.println(w);
  // Serial.print("S: ");

```



```

// Serial.println(s);
// Serial.print("K: ");
// Serial.println(k);
// Serial.print("DD: ");
// Serial.println(dd);
// Serial.print("EE: ");
// Serial.println(ee);
// Serial.print("FF: ");
// Serial.println(ff);
// Serial.print("fanhp 2: ");
// Serial.println(fanhp2Value);
// Serial.print("jendelahp 2: ");
// Serial.println(jendelahp2Value);
// Serial.print("pintuhp 2: ");
// Serial.println(pintuhp2Value);
// Serial.print("alarmhp 2: ");
// Serial.println(alarmhp2Value);
// Serial.print("ledhp 2: ");
// Serial.println(ledhp2Value);
// Serial.print("Notif 2 off: ");
// Serial.println(notif_offValue2);
Serial.println("=====");
// delay(100);
if (fanhp2Value == 1 && y == 0)
{
    digitalWrite(fan,0);
    y=1;
}
if(fanhp2Value == 0 && y == 1)
{
    digitalWrite(fan,1);
    y=0;
}
////////////////////////////////////
if (jendelahp2Value == 1 && z == 0)
{
    digitalWrite(pin_servo_4,HIGH);
    digitalWrite(pin_servo_5,HIGH);
    for (pos_2 = 90; pos_2 >= 0; pos_2 -= 1)
    {
        myservo_4.write(pos_2);
        delay(15);
        myservo_5.write(pos_2);
        delay(15);
    }
    z=1;
}
if(jendelahp2Value == 0 && z == 1)
{
    digitalWrite(pin_servo_4,HIGH);
    digitalWrite(pin_servo_5,HIGH);
    for (pos_2 = 0; pos_2 <= 90; pos_2 += 1)
    {
        myservo_4.write(pos_2);
        delay(15);
        myservo_5.write(pos_2);
    }
}

```

```

        delay(15);
    }
    z=0;
}
////////////////////////////////////
if (pintuhp2Value == 1 && u == 0)
{
    digitalWrite(pin_servo_6,HIGH);
    for (pos_2 = 90; pos_2 >= 0; pos_2 -= 1)
    {
        myservo_6.write(180);
        delay(7);
    }
    u=1;
}
if(pintuhp2Value == 0 && u == 1)
{
    digitalWrite(pin_servo_6,HIGH);
    for (pos_2 = 0; pos_2 <= 90; pos_2 += 1)
    {
        myservo_6.write(pos_2);
        delay(15);
    }
    u=0;
}
////////////////////////////////////
if (alarmhp2Value == 1 && v == 0)
{
    mp3_play();
    while(alarmhp2Value == 1 && v == 0)
    {
        programhp();
        alarmhp2Value = digitalRead(alarmhp2);
        mp3_play();Serial.println("musik2");
        baca_sensor();
        send_to_nodemcu();
        program_lcd();
        delay(1000);
    }
    v=1;
}
if(alarmhp2Value == 0 && v == 1)
{
    mp3_stop();
    v=0;
}
////////////////////////////////////
if (ledhp2Value == 1 && w == 0)
{
    digitalWrite (led_pin_2,1);
    w=1;
}
if(ledhp2Value == 0 && w == 1)
{
    digitalWrite (led_pin_2,0);
    w=0;
}

```

```

    }
    //////////////////////////////////////
    if(notif_offValue2 == 1 && k == 0)
    {
        s=0;
        k=1;
        Serial.println("notif 2 OFF");
    }
    if(notif_offValue2 == 0 && k == 1)
    {
        k=0;
        Serial.println("notif 2 normal");
    }
    //////////////////////////////////////
    if((apiValue_5 == 0 || apiValue_6 == 0 || apiValue_7 == 0 ||
    apiValue_8 == 0) && dd == 0)
    {
        dd=1;
    }
    if((apiValue_5 == 1 && apiValue_6 == 1 && apiValue_7 == 1 &&
    apiValue_8 == 1) && dd == 1)
    {
        dd=0;
    }
    //////////////////////////////////////
    if((asapValue_3 > 100 || asapValue_4 > 100) && ee == 0)
    {
        ee =1;
    }
    if((asapValue_3 < 100 && asapValue_4 < 100) && ee == 1)
    {
        ee =0;
    }
    //////////////////////////////////////
    if((suhu2 > 35) && ff ==0)
    {
        ff=1;
    }

    if(suhu2 <= 35 && ff ==1)
    {
        ff=0;
    }

    //////////////////////////////////////

    Serial1.print(apiValue_5);    Serial1.print("A");
    Serial1.print(apiValue_6);    Serial1.print("B");
    Serial1.print(apiValue_7);    Serial1.print("C");
    Serial1.print(apiValue_8);    Serial1.print("D");
    Serial1.print(asapValue_3);    Serial1.print("E");
    Serial1.print(asapValue_4);    Serial1.print("F");
    Serial1.print(suhu2);          Serial1.print("G");
    Serial1.print(r);              Serial1.print("H");
    Serial1.print(s);              Serial1.print("I");
    Serial1.print(dd);            Serial1.print("J");

```

```
Serial1.print(ee);          Serial1.print("K");
Serial1.print(ff);         Serial1.print("L");
Serial1.print("\n");
```

```
////////////////////////////////////
```

```
if ((apiValue_5 == 0 || apiValue_6 == 0 || apiValue_7 == 0 ||
apiValue_8 == 0) && (asapValue_3 > 110 || asapValue_4 > 110) &&
suhu2 > 38 && r==0)
```

```
{
  Serial.println("semua action ON");
  digitalWrite(fan,0);
  digitalWrite (led_pin_2,1);
  r=1;
  s=1;
  digitalWrite(pin_servo_4,HIGH);
  digitalWrite(pin_servo_5,HIGH);
  digitalWrite(pin_servo_6,HIGH);
  mp3_play();//Serial.println("musik");

  for (pos_2 = 90; pos_2 >= 0; pos_2 -= 1)
  {
    myservo_4.write(pos_2);
    delay(15);
    myservo_5.write(pos_2);
    delay(15);
    myservo_6.write(180);
    delay(15);
  }
```

```
while(apiValue_5 == 0 || apiValue_6 == 0 || apiValue_7 == 0
|| apiValue_8 == 0)
```

```
{
  mp3_play();Serial.println("musik1");
  baca_sensor();
  send_to_nodemcu();
  program_lcd();
  delay(1000);

  if(notif_offValue2 == 1 && k == 0)
  {
    s=0;
    k=1;
    Serial.println("notif 2 OFF");
  }
  if(notif_offValue2 == 0 && k == 1)
  {
    k=0;
    Serial.println("notif 2 normal");
  }
}
```

```
while(asapValue_3 > 230 || asapValue_4 > 230)
```

```
{
  mp3_play();Serial.println("musik2");
  baca_sensor();
```

```

    send_to_nodemcu();
    program_lcd();
    delay(1000);

    if(notif_offValue2 == 1 && k == 0)
    {
        s=0;
        k=1;
        Serial.println("notif 2 OFF");
    }

    if(notif_offValue2 == 0 && k == 1)
    {
        k=0;
        Serial.println("notif 2 normal");
    }
}

while(suhu2 > 38)
{
    mp3_play();Serial.println("musik3");
    baca_sensor();
    send_to_nodemcu();
    program_lcd();
    delay(1000);

    if(notif_offValue2 == 1 && k == 0)
    {
        s=0;
        k=1;
        Serial.println("notif 2 OFF");
    }

    if(notif_offValue2 == 0 && k == 1)
    {
        k=0;
        Serial.println("notif 2 normal");
    }
}
}

```

////////////////////////////////////

```

if ((apiValue_5 == 1 || apiValue_6 == 1 || apiValue_7 == 1 ||
apiValue_8 == 1) && (asapValue_3 < 110 || asapValue_4 < 110)
&& suhu2 < 38 && r==1)
{
    Serial.println("semua action OFF");
    digitalWrite(fan,1);
    digitalWrite (led_pin_2,0);

    mp3_stop();

    digitalWrite(pin_servo_4,HIGH);
    digitalWrite(pin_servo_5,HIGH);
    digitalWrite(pin_servo_6,HIGH);
    for (pos_2 = 0; pos_2 <= 90; pos_2 += 1)

```



```

// Serial.println(u);
// Serial.print("V: ");
// Serial.println(v);
// Serial.print("W: ");
// Serial.println(w);
// Serial.print("S: ");
// Serial.println(s);
// Serial.print("K: ");
// Serial.println(k);
// Serial.print("DD: ");
// Serial.println(dd);
// Serial.print("EE: ");
// Serial.println(ee);
// Serial.print("FF: ");
// Serial.println(ff);
// Serial.print("fanhp 2: ");
// Serial.println(fanhp2Value);
// Serial.print("jendelahp 2: ");
// Serial.println(jendelahp2Value);
// Serial.print("pintu hp 2: ");
// Serial.println(pintu hp2Value);
// Serial.print("alarmhp 2: ");
// Serial.println(alarmhp2Value);
// Serial.print("ledhp 2: ");
// Serial.println(ledhp2Value);
// Serial.print("Notif 2 off: ");
// Serial.println(notif_offValue2);
}
void send_to_nodemcu()
{
Serial1.print(apiValue_5);    Serial1.print("A");
Serial1.print(apiValue_6);    Serial1.print("B");
Serial1.print(apiValue_7);    Serial1.print("C");
Serial1.print(apiValue_8);    Serial1.print("D");
Serial1.print(asapValue_3);    Serial1.print("E");
Serial1.print(asapValue_4);    Serial1.print("F");
Serial1.print(suhu2);        Serial1.print("G");
Serial1.print(r);            Serial1.print("H");
Serial1.print(s);            Serial1.print("I");
Serial1.print(dd);          Serial1.print("J");
Serial1.print(ee);          Serial1.print("K");
Serial1.print(ff);          Serial1.print("L");
Serial1.print("\n");
}

void programhp()
{
    if (fanhp2Value == 1 && y == 0)
    {
        digitalWrite(fan,0);
        y=1;
    }
    if(fanhp2Value == 0 && y == 1)
    {
        digitalWrite(fan,1);
    }
}

```

```

        y=0;
    }
    ///////////////////////////////////////////////////////////////////
    if (jendelahp2Value == 1 && z == 0)
    {
        digitalWrite(pin_servo_4,HIGH);
        digitalWrite(pin_servo_5,HIGH);
        for (pos_2 = 90; pos_2 >= 0; pos_2 -= 1)
        {
            myservo_4.write(pos_2);
            delay(15);
            myservo_5.write(pos_2);
            delay(15);
        }
        z=1;
    }
    if(jendelahp2Value == 0 && z == 1)
    {
        digitalWrite(pin_servo_4,HIGH);
        digitalWrite(pin_servo_5,HIGH);
        for (pos_2 = 0; pos_2 <= 90; pos_2 += 1)
        {
            myservo_4.write(pos_2);
            delay(15);
            myservo_5.write(pos_2);
            delay(15);
        }
        z=0;
    }
    ///////////////////////////////////////////////////////////////////
    if (pintuhp2Value == 1 && u == 0)
    {
        digitalWrite(pin_servo_6,HIGH);
        for (pos_2 = 90; pos_2 >= 0; pos_2 -= 1)
        {
            myservo_6.write(180);
            delay(7);
        }
        u=1;
    }
    if(pintuhp2Value == 0 && u == 1)
    {
        digitalWrite(pin_servo_6,HIGH);
        for (pos_2 = 0; pos_2 <= 90; pos_2 += 1)
        {
            myservo_6.write(pos_2);
            delay(15);
        }
        u=0;
    }
    ///////////////////////////////////////////////////////////////////
    if (ledhp2Value == 1 && w == 0)
    {
        digitalWrite (led_pin_2,1);
        w=1;
    }
}

```



```

    if(ledhp2Value == 0 && w == 1)
    {
        digitalWrite (led_pin_2,0);
        w=0;
    }
    //////////////////////////////////////
    if(notif_offValue2 == 1 && k == 0)
    {
        s=0;
        k=1;
        Serial.println("notif 2 OFF");
    }
    if(notif_offValue2 == 0 && k == 1)
    {
        k=0;
        Serial.println("notif 2 normal");
    }

    //////////////////////////////////////
    if((apiValue_5 == 0 || apiValue_6 == 0 || apiValue_7 == 0 ||
    apiValue_8 == 0) && dd == 0)
    {
        dd=1;
    }

    if((apiValue_5 == 1 && apiValue_6 == 1 && apiValue_7 == 1 &&
    apiValue_8 == 1) && dd == 1)
    {
        dd=0;
    }
    //////////////////////////////////////

    if((asapValue_3 > 100 || asapValue_4 > 100) && ee == 0)
    {
        ee =1;
    }
    if((asapValue_3 < 100 && asapValue_4 < 100) && ee == 1)
    {
        ee =0;
    }
    //////////////////////////////////////
    if((suhu2 > 35) && ff ==0)
    {
        ff=1;
    }
    if(suhu2 <= 35 && ff ==1)
    {
        ff=0;
    }
}

void program_lcd()
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Api1=");
}

```

```

lcd.setCursor(6,0);
lcd.print(apiValue_5);

lcd.setCursor(8,0);
lcd.print("Api3=");
lcd.setCursor(14,0);
lcd.print(apiValue_7);

lcd.setCursor(0,1);
lcd.print("Api2=");
lcd.setCursor(6,1);
lcd.print(apiValue_6);

lcd.setCursor(8,1);
lcd.print("Api4=");
lcd.setCursor(14,1);
lcd.print(apiValue_8);
////////////////////////////////////
lcd.setCursor(0,2);
lcd.print("Asap1=");
lcd.setCursor(6,2);
lcd.print(asapValue_3);

lcd.setCursor(10,2);
lcd.print("Asap2=");
lcd.setCursor(16,2);
lcd.print(asapValue_4);
////////////////////////////////////
lcd.setCursor(0,3);
lcd.print("Suhu =");
lcd.setCursor(7,3);
lcd.print(suhu2);
lcd.setCursor(9,3);
lcd.print((char)223);
lcd.setCursor(10,3);
lcd.print("C");
}

```

Sedangkan program Arduino Mega 2560 untuk ruang 2 dapat dilihat pada link berikut:

https://drive.google.com/file/d/1L701AIY_YMWbwBzhrebEqKc6vYuPAPNS/view?usp=sharing

LAMPIRAN 3

Program NodeMCU Ruang 1 :

```
#include<SoftwareSerial.h>
#include <ArduinoJson.h>
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>

#define FIREBASE_HOST "diniek0-ta-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "P7YzY6dehIhq3Qa35f7g7EYSWEH0YytXvOc3aIRl"
#define WIFI_SSID "Cintaku"
#define WIFI_PASSWORD "123456789"

int jendelahp2 = D3;
int pintuhp2 = D4;
int fanhp2 = D5;
int alarmhp2 =D6;
int ledhp2 =D7;
int notif_off2 = D8;
SoftwareSerial NodeMcu_SoftSerial(D1, D2);

char h;
String dataIn;
int8_t indexOfA, indexOfB, indexOfC, indexOfD, indexOfE, indexOfF,
indexOfG, indexOfH, indexOfI, indexOfJ, indexOfK, indexOfL;

String apiValue_5, apiValue_6, apiValue_7, apiValue_8,
asapValue_3, asapValue_4, suhu2, r, s, dd, ee, ff;

String i;
String j;
String k;
String l;
String m;
String n;

void setup() {

  Serial.begin(57600);
  NodeMcu_SoftSerial.begin(57600);while (!Serial) continue;

  pinMode(D3,OUTPUT);
  pinMode(D4,OUTPUT);
  pinMode(D5,OUTPUT);
  pinMode(D6,OUTPUT);
  pinMode(D7,OUTPUT);
  pinMode(D8,OUTPUT);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
```

```

    Serial.print("WiFi connected! ");
    // Serial.println(WiFi.localIP());
    Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}
void loop() {

    i=Firebase.getString("fanhp2");
    //Serial.println (i);
    if (i=="1"){
        //Serial.println("fan 2 ON");
        digitalWrite(fanhp2,HIGH); //Device1 is ON

    }
    else{
        //Serial.println("fan 2 OFF");
        digitalWrite(fanhp2,LOW); //Device1 if OFF
    }
    ///////////////////////////////////////////////////////////////////
    j=Firebase.getString("jendelahp2");
    //Serial.println (j);
    if (j=="1"){
        //Serial.println("jendela 2 ON");
        digitalWrite(jendelahp2,HIGH); //Device1 is ON
    }
    else{
        //Serial.println("jendela 2 OFF");
        digitalWrite(jendelahp2,LOW); //Device1 if OFF
    }
    ///////////////////////////////////////////////////////////////////
    k=Firebase.getString("pintuhp2");
    //Serial.println (k);
    if (k=="1"){
        //Serial.println("Pintu 2 ON");
        digitalWrite(pintuhp2,HIGH); //Device1 is ON
    }
    else{
        //Serial.println("Pintu 2 OFF");
        digitalWrite(pintuhp2,LOW); //Device1 if OFF
    }
    ///////////////////////////////////////////////////////////////////
    l=Firebase.getString("alarmhp2");
    //Serial.println (l);
    if (l=="1"){
        //Serial.println("alarm 2 ON");
        digitalWrite(alarmhp2,HIGH); //Device1 is ON
    }
    else{
        //Serial.println("alarm 2 OFF");
        digitalWrite(alarmhp2,LOW); //Device1 if OFF
    }
    ///////////////////////////////////////////////////////////////////
    m=Firebase.getString("ledhp2");
    //Serial.println (m);
    if (m=="1"){
        //Serial.println("LED 2 ON");
        digitalWrite(ledhp2,HIGH); //Device1 is ON
    }
}

```

```

    }
    else{
        //Serial.println("LED 2 OFF");
        digitalWrite(ledhp2,LOW);//Device1 if OFF
    }
    //////////////////////////////////////
    n=Firebase.getString("notif_off2");
    //Serial.println (n);
    if (n=="1"){
        //Serial.println("Notif off2");
        digitalWrite(notif_off2,HIGH); //Device1 is ON
    }
    else{
        //Serial.println("Notif2 netral");
        digitalWrite(notif_off2,LOW);
    }
    //////////////////////////////////////

    while(NodeMcu_SoftSerial.available ()>0)
    {
        h = NodeMcu_SoftSerial.read();

        if(h=='\n') {break;}
        else      {dataIn+=h;}
    }

    if(h=='\n')
    {
        Parse_the_Data();

        Serial.println("Api5 = " + apiValue_5);
        Serial.println("Api6 = " + apiValue_6);
        Serial.println("Api7 = " + apiValue_7);
        Serial.println("Api8 = " + apiValue_8);
        Serial.println("Asap3 = " + asapValue_3);
        Serial.println("Asap4 = " + asapValue_4);
        Serial.println("Suhu2 = " + suhu2);
        //      Serial.println("R = " + r);
        //      Serial.println("S = " + s);
        //      Serial.println("dd = " + dd);
        //      Serial.println("ee = " + ee);
        //      Serial.println("ff = " + ff);

        Serial.println("=====");
        h=0;
        dataIn="";

        Firebase.setString("Api5", String(apiValue_5));
        Firebase.setString("Api6", String(apiValue_6));
        Firebase.setString("Api7", String(apiValue_7));
        Firebase.setString("Api8", String(apiValue_8));
        Firebase.setString("Asap3", String(asapValue_3));
        Firebase.setString("Asap4", String(asapValue_4));
        Firebase.setString("Suhu2", String(suhu2));
        Firebase.setString("R", String(r));
    }

```

```

        Firebase.setString("S", String(s));
        Firebase.setString("dd", String(dd));
        Firebase.setString("ee", String(ee));
        Firebase.setString("ff", String(ff));
    }

}
////////////////////////////////////
void Parse_the_Data()
{
    indexOfA = dataIn.indexOf("A");
    indexOfB = dataIn.indexOf("B");
    indexOfC = dataIn.indexOf("C");
    indexOfD = dataIn.indexOf("D");
    indexOfE = dataIn.indexOf("E");
    indexOfF = dataIn.indexOf("F");
    indexOfG = dataIn.indexOf("G");
    indexOfH = dataIn.indexOf("H");
    indexOfI = dataIn.indexOf("I");
    indexOfJ = dataIn.indexOf("J");
    indexOfK = dataIn.indexOf("K");
    indexOfL = dataIn.indexOf("L");

    if(indexOfA != -1 && indexOfB != -1 && indexOfC != -1 &&
    indexOfD != -1 && indexOfE != -1 && indexOfF != -1 && indexOfG
    != -1 && indexOfH != -1 && indexOfI != -1 && indexOfJ != -1 &&
    indexOfK != -1 && indexOfL != -1)
    {
        apiValue_5 = dataIn.substring (0, indexOfA);
        apiValue_6 = dataIn.substring (indexOfA+1, indexOfB);
        apiValue_7 = dataIn.substring (indexOfB+1, indexOfC);
        apiValue_8 = dataIn.substring (indexOfC+1, indexOfD);
        asapValue_3 = dataIn.substring (indexOfD+1, indexOfE);
        asapValue_4 = dataIn.substring (indexOfE+1, indexOfF);
        suhu2 = dataIn.substring (indexOfF+1, indexOfG);
        r = dataIn.substring (indexOfG+1, indexOfH);
        s = dataIn.substring (indexOfH+1, indexOfI);
        dd = dataIn.substring (indexOfI+1, indexOfJ);
        ee = dataIn.substring (indexOfJ+1, indexOfK);
        ff = dataIn.substring (indexOfK+1, indexOfL);
    }
}

```

Sedangkan program NodeMCU untuk ruang 2 bisa dilihat pada link berikut:

https://drive.google.com/file/d/1L701AIY_YMWbwBzhrebEqKc6vYuPAPNS/view?usp=sharing