



**RANCANG BANGUN KENDALI GAS UNTUK MENCEGAH  
KEBAKARAN AKIBAT KEBOCORAN LPG MENGGUNAKAN  
*FUZZY LOGIC BERBASIS INTERNET OF THINGS***



INSTITUT BISNIS  
DAN INFORMATIKA

**stikom**  
SURABAYA

Oleh:

**Insan Nurseha Aziz**

**15410200047**

---

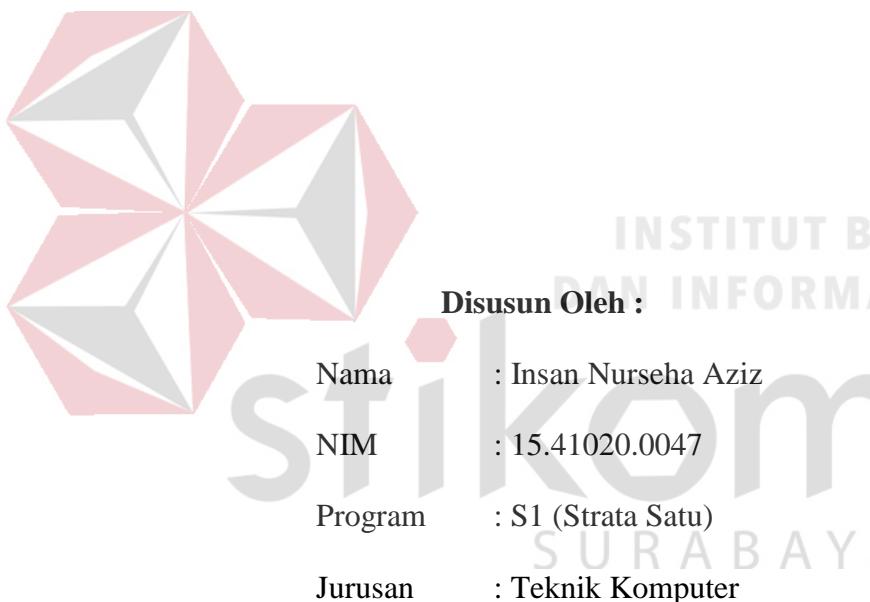
---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA  
2019**

**RANCANG BANGUN KENDALI GAS UNTUK MENCEGAH  
KEBAKARAN AKIBAT KEBOCORAN LPG MENGGUNAKAN  
*FUZZY LOGIC BERBASIS INTERNET OF THINGS***

**TUGAS AKHIR**

Diajukan sebagai salah satu syarat untuk menempuh menyelesaikan Program  
Sarjana Teknik



**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA  
2019**



*-Imam bin Al Qayim-*

## LEMBAR PERSEMBAHAN

Buku Tugas Akhir ini penulis peruntukan kepada:

Untuk kedua orang tua, Bapak Wasistoadi dan Ibu Suprapti Ningsih yang selalu memberikan do'a serta dukungan penuh kepada penulis.

Untuk keluarga penulis yang selalu memberikan masukan serta saran kepada penulis.



## TUGAS AKHIR

### RANCANG BANGUN KENDALI GAS UNTUK MENCEGAH KEBAKARAN AKIBAT KEBOCORAN LPG MENGGUNAKAN FUZZY *LOGIC BERBASIS INTERNET OF THINGS*

Dipersiapkan dan disusun oleh :

**Insan Nurseha Aziz**

**NIM: 15.41020.0047**

Telah diperiksa, diuji dan disetujui oleh Dewan Pengaji

Pada : Juli 2019



Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar Sarjana



Dekan Fakultas Teknologi dan Informatika

Institut Bisnis dan Informatika Stikom Surabaya

**SURAT PERNYATAAN**  
**PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, penulis:

Nama : Insan Nurseha Aziz  
NIM : 15.41020.0047  
Program Studi : S1 Teknik Komputer  
Fakultas : Fakultas Teknologi dan Informatika  
Jenis Karya : Tugas Akhir  
Judul Karya : **RANCANG BANGUN KENDALI GAS UNTUK  
MENCEGAH KEBAKARAN AKIBAT KEBOCORAN  
LPG MENGGUNAKAN FUZZY LOGIC BERBASIS  
INTERNET OF THINGS**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, penulis menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/ sebagian karya ilmiah penulis tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama penulis sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut di atas adalah karya asli penulis, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka penulis.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka penulis bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada penulis.

Demikian surat pernyataan ini penulis buat dengan sebenarnya.

Surabaya, Juli 2019



Insan Nurseha Aziz  
NIM: 15.41020.0047

## ABSTRAK

Angka konsumsi LPG untuk rumah tangga di Indonesia terus meningkat setiap tahunnya. Pemerintah Indonesia sendiri pernah mensosialisasikan program konversi minyak tanah ke gas LPG pada tahun 2007. Hal inilah yang mendasari maraknya penggunaan LPG untuk rumah tangga. Namun, ketergantungan terhadap LPG ini masih menimbulkan adanya masalah. LPG yang digunakan untuk kebutuhan rumah tangga sering kali mengalami kebocoran. Kebocoran gas tersebut sering mengakibatkan kebakaran dalam rumah, sehingga banyak masyarakat yang masih khawatir dalam penggunaan LPG.

Berdasarkan latar belakang inilah penulis membuat rancang bangun kendali gas untuk mencegah kebakaran akibat kebocoran LPG menggunakan *Fuzzy Logic* berbasis *Internet of Things*. Rancang bangun ini menggunakan *Wemos D1 Mini* karena merupakan modul *development board* yang berbasis *Wi-Fi*. Selain itu, rancang bangun ini dilengkapi dengan pemberian notifikasi kepada pemilik rumah dengan berbasis *Android* yaitu aplikasi *Blynk*, sehingga dapat dimonitoring dengan lebih mudah, serta mampu memberikan tindakan antisipasi untuk mencegah kebakaran yang akan dikontrol menggunakan *Fuzzy Logic*.

Sistem *Fuzzy Sugeno* dapat diaplikasikan untuk mengatur kecepatan kipas dengan baik dan sesuai dari sistem yang telah dirancang. Aplikasi *Blynk* ini dapat digunakan secara jarak jauh maupun jarak dekat. Sistem *Fuzzy* yang digunakan pada alat kebocoran gas ini membutuhkan kecepatan rata-rata sebesar 0,465 ADC/detik untuk kembali pada kondisi nilai ADC sebelum ada gas.

**Kata Kunci:** LPG, *Blynk*, *Fuzzy Logic*, *Monitoring*

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat yang telah diberikan - Nya, sehingga penulis dapat menyelesaikan Tugas Akhir ini. Penulisan Tugas Akhir ini adalah sebagai salah satu syarat dalam menyelesaikan studi penulis di Program Studi S1 Teknik Komputer Institut Bisnis dan Informatika Stikom Surabaya.

Dalam usaha menyelesaikan penulisan Tugas Akhir ini penulis banyak mendapat bantuan dari berbagai pihak baik moral maupun materi. Oleh karena itu penulis mengucapkan terima kasih dan penghargaan setinggi-tingginya kepada:

1. Orang tua dan saudara penulis tercinta, karena dengan segala dukungan serta motivasinya mulai dari awal penulis mengerjakan Tugas Akhir ini sampai selesai.
2. Bapak Harianto, S.Kom., M.Eng., selaku Dosen Pembimbing pertama dan Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku Dosen Pembimbing kedua yang telah meluangkan waktu dan pikiran serta dengan sabar membimbing penulis dari awal hingga Tugas Akhir ini dapat terselesaikan.
3. Pimpinan Institut Bisnis dan Informatika Stikom Surabaya yang telah banyak memberi motivasi serta teladan yang dapat membantu penulis selama menempuh pembelajaran hingga saat ini.
4. Bapak Dr. Jusak selaku Dekan Fakultas Teknologi dan Informatika (FTI) Institut Bisnis dan Informatika Stikom Surabaya telah membantu proses penyelesaian Tugas Akhir yang dibuat oleh penulis dengan baik.

5. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer telah membantu dalam proses penyelesaian Tugas Akhir ini dengan sebaik-baiknya.
6. Ibu Weny Indah Kusumawati, S.Kom, M.MT., selaku penguji yang telah membimbing penulis serta memberi masukan dalam menyusun buku Tugas Akhir dan membimbing selama menempuh perkuliahan di Institut Bisnis dan Informatika Stikom Surabaya.
7. Seluruh dosen pengajar Program Studi S1 Teknik Komputer yang telah mendidik, memberi motivasi kepada penulis selama masa kuliah di Institut Bisnis dan Informatika Stikom Surabaya.
8. Rekan-rekan angkatan 2015 maupun adik dan kakak angkatan jurusan S1 Teknik Komputer yang mendukung dan membantu penulis selama masa penyusunan buku Tugas Akhir ini.
9. Seluruh pihak yang tidak dapat penulis tuliskan satu yang membantu penulis secara langsung maupun tidak langsung dalam menyelesaikan Tugas Akhir ini.

Banyak hal dalam laporan Tugas Akhir ini yang masih perlu diperbaiki. Oleh karena itu penulis mengharapkan kritik dan saran yang dapat membangun dari semua pihak agar dapat menyempurnakan penulisan ini kedepanya. Akhir kata penulis mengucapkan banyak terima kasih yang sebesar-besarnya kepada para pembaca, semoga tulisan ini dapat bermanfaat dan memberi inspirasi lebih banyak untuk pembaca dan pembaca dapat berkarya melebihi karya penulis.

Surabaya, Juli 2019

Penulis

## DAFTAR ISI

Halaman

ABSTRAK .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL.....	xvii
DAFTAR LAMPIRAN.....	xviii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI .....	6
2.1 Wemos.....	6
2.1.1 Wemos D1 Mini.....	6
2.1.2 Chipset Pada Wemos .....	8
2.2 Sensor Figaro TGS2610 .....	8
2.3 Fuzzy Logic.....	9
2.3.1 Fuzzy Sugeno.....	10

2.4	<i>LCD M1632</i> .....	11
2.5	<i>Buzzer</i> .....	13
2.6	<i>Real Time Clock</i> .....	13
2.7	<i>Motor Driver</i> .....	15
2.8	Kipas DC .....	16
2.9	<i>Modul Relay</i> .....	17
2.10	<i>Solenoid Valve Gas</i> .....	19
2.11	Blynk .....	20
2.12	<i>Arduino Software IDE</i> .....	21
<b>BAB III METODOLOGI PENELITIAN.....</b>		<b>23</b>
3.1	Metode Penelitian.....	23
3.2	Diagram <i>Hardware</i> .....	23
3.3	Perancangan Mekanik .....	25
3.4	Perancangan Elektronika .....	27
	3.4.1 Rangkaian Sensor Figaro TGS2610 .....	29
	3.4.2 Rangkaian RTC.....	29
	3.4.3 Rangkaian LCD Dan <i>Buzzer</i> .....	31
	3.4.4 Rangkaian <i>Relay</i> Dan <i>Solenoid Valve Gas</i> .....	32
	3.4.5 Rangkaian Motor <i>Driver</i> Dan Kipas DC .....	33
3.5	Perancangan <i>Software</i> .....	34
	3.5.1 <i>Flowchart</i> Sistem Pendekripsi Kebocoran Gas LPG .....	34

3.5.2 Aplikasi <i>Monitoring</i> Dan Kontrol.....	36
3.5.3 Konfigurasi Arduino IDE .....	50
3.5.4 Program Koneksi Wemos D1 Mini Dengan <i>Wi-Fi</i> .....	54
<b>3.6 Desain Sistem <i>Fuzzy</i>.....</b>	<b>55</b>
3.6.1 <i>Input</i> Pada <i>Fuzzy Sugeno</i> .....	55
3.6.2 <i>Output</i> Pada <i>Fuzzy Sugeno</i> .....	58
3.6.3 <i>Flowchart</i> Sistem <i>Fuzzy Logic</i> .....	59
<b>BAB IV HASIL PENGUJIAN DAN PENGAMATAN .....</b>	<b>72</b>
4.1 Pengujian Wemos D1 Mini .....	72
4.2 Pengujian Sensor Figaro TGS2610 .....	74
4.3 Pengujian <i>Relay</i> Dan <i>Valve</i> .....	75
4.4 Pengujian <i>Real Time Clock</i> .....	78
4.5 Pengujian Notifikasi.....	79
4.6 Pengujian <i>Monitoring</i> Jarak Jauh Alat Dengan <i>Android</i> .....	81
4.7 Pengujian Sistem <i>Fuzzy Logic</i> .....	85
<b>BAB V PENUTUP.....</b>	<b>90</b>
5.1 Kesimpulan.....	90
5.2 Saran .....	90
<b>DAFTAR PUSTAKA .....</b>	<b>91</b>
<b>BIODATA PENULIS .....</b>	<b>110</b>

## DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Wemos D1 Mini .....	7
Gambar 2. 2 Sensor Figaro TGS2610 .....	9
Gambar 2. 3 Tampilan LCD 16x2.....	12
Gambar 2. 4 Buzzer.....	13
Gambar 2. 5 <i>Real Time Clock</i> .....	15
Gambar 2. 6 <i>Motor Driver L298N</i> .....	16
Gambar 2. 7 Kipas DC .....	17
Gambar 2. 8 Modul <i>Relay</i> .....	18
Gambar 2. 9 Diagram <i>Solenoid Valve Gas</i> 2 Saluran .....	19
Gambar 2. 10 <i>Solenoid Valve Gas</i> .....	19
Gambar 2. 11 Tampilan Aplikasi Blynk .....	20
Gambar 2. 12 Tampilan <i>Software Arduino IDE</i> .....	21
Gambar 3. 1 Blok Diagram Sistem Deteksi Kebocoran LPG .....	24
Gambar 3. 2 Desain Rancang Mekanik.....	25
Gambar 3. 3 Hasil Rancang Mekanik (A) Tampak Luar, (B) Tampak Dalam, Dan (C) Tampak Samping.....	26
Gambar 3. 4 Rangkaian Elektronika Rancang Bangun Pendekripsi Kebocoran LPG ..	28
Gambar 3. 5 Rangkaian Sensor Figaro TGS2610.....	29
Gambar 3. 6 Rangkaian RTC .....	30
Gambar 3. 7 Rangkaian LCD Dan Buzzer.....	31
Gambar 3. 8 Rangkaian Modul <i>Relay</i> Dan <i>Solenoid Valve Gas</i> .....	32
Gambar 3. 9 Rangkaian Motor <i>Driver</i> Dan Kipas DC .....	33
Gambar 3. 10 <i>Flowchart</i> Sistem Deteksi Kebocoran Gas .....	35

Gambar 3. 11 Tampilan <i>Login</i> Pada Aplikasi Blynk .....	37
Gambar 3. 12 Tampilan <i>New Project</i> .....	37
Gambar 3. 13 Tampilan <i>Creat New Project</i> .....	38
Gambar 3. 14 <i>Auth Token</i> Yang Dikirimkan Oleh Aplikasi .....	39
Gambar 3. 15 Tampilan Awal Dan <i>Widget Box</i> .....	40
Gambar 3. 16 Tampilan Aplikasi Blynk .....	40
Gambar 3. 17 <i>Setting Gauge</i> (A) ADC Gas Dan (B) Perubahan Gas .....	42
Gambar 3. 18 Program <i>Gauge</i> .....	42
Gambar 3. 19 <i>Setting Display Value</i> Untuk (A) PWM, (B) PPM Dan (C) RTC..	44
Gambar 3. 20 Program <i>Display Value</i> .....	44
Gambar 3. 21 Tampilan (A) <i>SuperChart Setting</i> Dan (B) <i>Setting Waktu</i> .....	45
Gambar 3. 22 Tampilan <i>Setting SuperChart</i> (A) ADC, (B) Perubahan ADC Dan (C) PWM .....	46
Gambar 3. 23 Program <i>SuperChart</i> .....	47
Gambar 3. 24 <i>Setting Button</i> (A) Pada <i>Valve</i> Dan (B) Pada <i>Buzzer</i> .....	48
Gambar 3. 25 Program <i>Button</i> .....	48
Gambar 3. 26 <i>Setting Eventor</i> (A) Tampilan Utama <i>Setting Eventor</i> ; (B) <i>Setting New Event</i> .....	49
Gambar 3. 27 <i>Setting Notification</i> (A) Untuk Pemberian Informasi Koneksi Internet Pada Wemos Dan (B) Untuk Pemberian Informasi Apabila Terjadi Kebocoran Gas .....	50
Gambar 3. 28 Instalasi <i>Library</i> ESP8266 Dengan <i>Arduino Board Manager</i> .....	51
Gambar 3. 29 <i>Install</i> Wemos <i>Board Manager</i> .....	51
Gambar 3. 30 Pemilihan <i>Board</i> Wemos D1 Mini.....	52
Gambar 3. 31 <i>Install Library</i> Blynk.....	53
Gambar 3. 32 <i>Manager Library</i> .....	53

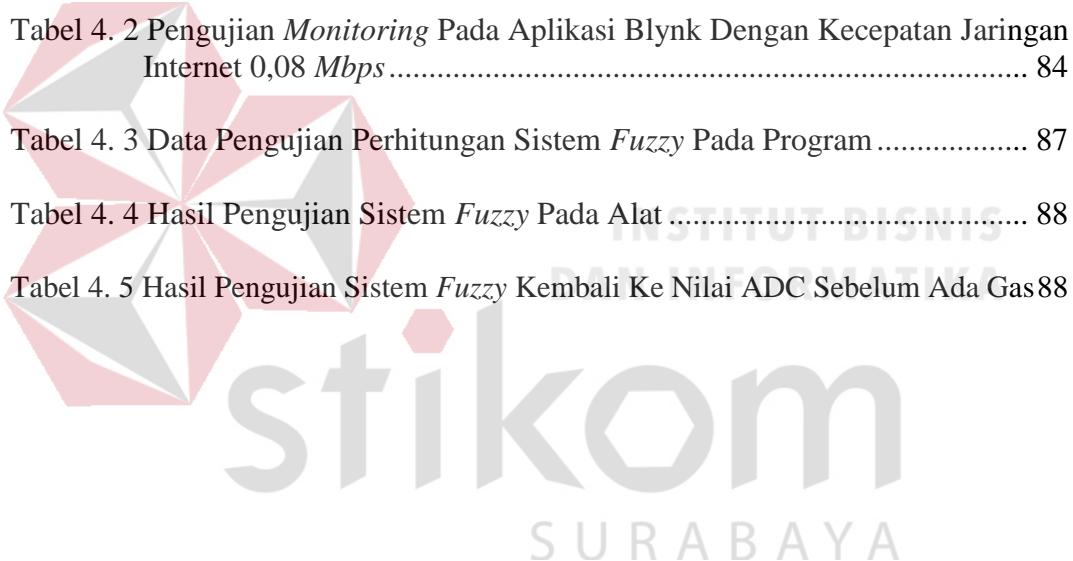
Gambar 3. 33 <i>Include Library Blynk</i> .....	54
Gambar 3. 34 Kode Koneksi Wemos D1 Mini Dengan Wi-Fi .....	55
Gambar 3. 35 Fungsi Keanggotaan Gas.....	56
Gambar 3. 36 Fungsi Keanggotaan Perubahan Gas (+).....	57
Gambar 3. 37 Fungsi Keanggotaan Perubahan Gas (-).....	57
Gambar 3. 38 Fungsi Keanggotaan Kipas.....	58
Gambar 3. 39 <i>Flowchart Fuzzifikasi</i> Gas Sedikit .....	60
Gambar 3. 40 <i>Flowchart Fuzzifikasi</i> Gas Sedang .....	62
Gambar 3. 41 <i>Flowchart Fuzzifikasi</i> Gas Banyak .....	63
Gambar 3. 42 <i>Flowchart Fuzzifikasi</i> Perubahan Gas Sedikit (+) .....	64
Gambar 3. 43 <i>Flowchart Fuzzifikasi</i> Perubahan Gas Sedang (+) .....	65
Gambar 3. 44 <i>Flowchart Fuzzifikasi</i> Perubahan Gas Banyak (+).....	66
Gambar 3. 45 <i>Flowchart Fuzzifikasi</i> Perubahan Gas Sedikit (-) .....	67
Gambar 3. 46 <i>Flowchart Fuzzifikasi</i> Perubahan Gas Sedang (-) .....	68
Gambar 3. 47 <i>Flowchart Fuzzifikasi</i> Perubahan Gas Banyak (-).....	69
Gambar 3. 48 <i>Flowchart</i> Mencari Nilai Minimum.....	70
Gambar 4. 1 (A) Proses <i>Upload</i> Dari Arduino IDE Menuju Wemos D1 Mini Dan (B) LED Pada Wemos Menyalा Saat Program Berjalan.....	74
Gambar 4. 2 Pembacaan Sensor Figaro TGS2610 Pada <i>Serial Monitor</i> .....	75
Gambar 4. 3 <i>Relay</i> Bernilai <i>High</i> .....	76
Gambar 4. 4 Kompor Menyalा Saat <i>Valve</i> Terbuka .....	77
Gambar 4. 5 <i>Relay</i> Bernilai <i>Low</i> .....	77
Gambar 4. 6 Kompor Tidak Menyalा Saat <i>Valve</i> Tertutup .....	78
Gambar 4. 7 (A) Proses Pembacaan RTC Pada <i>Serial Monitor</i> (B) <i>Real Time</i> Pada <i>Smartphone</i> .....	79

Gambar 4. 8 Tampilan LCD Pengujian Notifikasi.....	80
Gambar 4. 9 Tampilan Notifikasi Pada <i>Smartphone Android</i> .....	80
Gambar 4. 10 Tampilan Data Pada LCD .....	82
Gambar 4. 11 Tampilan Data Pada Aplikasi Blynk.....	82



## DAFTAR TABEL

	Halaman
Tabel 2. 1 Spesifikasi Wemos D1 Mini .....	8
Tabel 3. 1 Keanggotaan Gas .....	56
Tabel 3. 2 Keanggotaan Perubahan Gas(+) .....	56
Tabel 3. 3 Keanggotaan Perubahan Gas (-) .....	57
Tabel 3. 4 <i>Rule Base Fuzzy</i> .....	58
Tabel 4. 1 Pengujian <i>Monitoring</i> Pada Aplikasi Blynk Dengan Kecepatan Jaringan Internet 11,1 <i>Mbps</i> .....	83
Tabel 4. 2 Pengujian <i>Monitoring</i> Pada Aplikasi Blynk Dengan Kecepatan Jaringan Internet 0,08 <i>Mbps</i> .....	84
Tabel 4. 3 Data Pengujian Perhitungan Sistem <i>Fuzzy</i> Pada Program .....	87
Tabel 4. 4 Hasil Pengujian Sistem <i>Fuzzy</i> Pada Alat .....	88
Tabel 4. 5 Hasil Pengujian Sistem <i>Fuzzy</i> Kembali Ke Nilai ADC Sebelum Ada Gas	88



## **DAFTAR LAMPIRAN**

Halaman

LAMPIRAN 1 Hasil Uji *Monitoring* Dengan Kecepatan Internet 11,1 Mbps ..... 93

LAMPIRAN 2 Listing Program *Monitoring* Gas Dengan *Fuzzy Logic* ..... 96

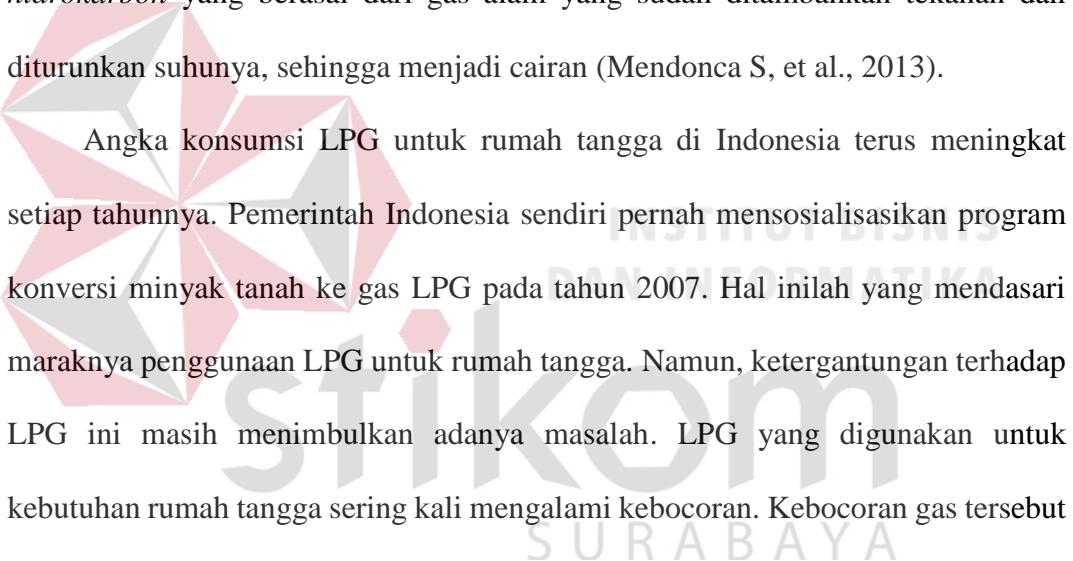


## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Energi memiliki peran yang sangat penting pada setiap aspek kehidupan. Hal ini yang menyebabkan ketergantungan terhadap energi di Indonesia sudah mencapai angka yang sangat tinggi. Salah satu energi yang menjadi aspek penting yaitu LPG (*Liquified Petroleum Gas*). LPG adalah campuran dari berbagai unsur *hidrokarbon* yang berasal dari gas alam yang sudah ditambahkan tekanan dan diturunkan suhunya, sehingga menjadi cairan (Mendonca S, et al., 2013).



Angka konsumsi LPG untuk rumah tangga di Indonesia terus meningkat setiap tahunnya. Pemerintah Indonesia sendiri pernah mensosialisasikan program konversi minyak tanah ke gas LPG pada tahun 2007. Hal inilah yang mendasari maraknya penggunaan LPG untuk rumah tangga. Namun, ketergantungan terhadap LPG ini masih menimbulkan adanya masalah. LPG yang digunakan untuk kebutuhan rumah tangga sering kali mengalami kebocoran. Kebocoran gas tersebut sering mengakibatkan kebakaran dalam rumah, sehingga banyak masyarakat yang masih khawatir dalam penggunaan LPG.

Menurut Badan Perlindungan Konsumen Nasional, kasus ledakan gas LPG terus meningkat dari tahun 2007 hingga 2010 dari 5 kasus hingga mencapai 33 kasus (Badan Perlindungan Konsumen Nasional, 2010). Ledakan gas tersebut diakibatkan oleh aksesoris diantaranya regulator, selang gas, karet maupun katup yang dibawah standar, serta pemasangan yang tidak sempurna (Mendonca S, et al., 2013).

Sampai saat ini terdapat beberapa penelitian yang mengarah pada permasalahan ini. Salah satu penelitian sebelumnya dilakukan oleh Bambang Eko Soemarsono, et al. pada tahun 2015. Penelitian tersebut membuat suatu rancangan pendekksi kebocoran gas LPG. Rancang bangun ini terdiri dari sensor pendekksi gas LPG HS-133 yang berfungsi mendekksi kadar gas pada udara, sinyal yang telah diubah oleh ADC kemudian diolah oleh *mikrokontroler* ATMega 8535 untuk mengaktifkan *buzzer* serta SMS (*Short Message Service*) (Soemarsono, et al., 2015). Selain itu, Desi Nurnaningsih juga melakukan hal serupa pada tahun 2018. Penelitian ini menggunakan sensor MQ-2, serta menggunakan sistem SMS *Gateway* dengan modul SIM 800L v.2 sebagai pengirim notifikasi dan *buzzer* sebagai alarm (Nurnaningsih, 2018).

Adapun penelitian lainnya dilakukan oleh Aditya Bram Wiratma pada tahun 2016. Penelitian ini membuat rancang bangun sistem deteksi kebocoran tabung gas Elpiji menggunakan teknologi *Wireless Sensor Network*. Sistem ini menggunakan *mikrokontroller* Arduino Uno sebagai pengolah data dari sensor MQ-6 yang berfungsi untuk mendekksi adanya kandungan gas metana dalam udara. Dari data sensor yang didapat akan dikirim ke *node* koordinator untuk ditampilkan dalam bentuk GUI dan menampilkan *warning* yang berada di dalamnya. Setiap sensor akan saling terhubung dan membentuk topologi *cluster-tree*. Sistem komunikasi pada *Zigbee* menggunakan modul *transceiver RF Xbee Series 2*. Sistem dibagi menjadi dua *cluster* untuk membagi beban *router node* (Wiratma, et al., 2016).

Berdasarkan latar belakang inilah penulis membuat rancang bangun kendali gas untuk mencegah kebakaran akibat kebocoran LPG menggunakan *Fuzzy Logic* berbasis *Internet of Things*. Rancang bangun ini menggunakan *Wemos D1 Mini*

karena merupakan modul *development board* yang berbasis *Wi-Fi*. Selain itu, rancang bangun ini dilengkapi dengan pemberian notifikasi kepada pemilik rumah dengan berbasis *Android*, sehingga dapat dimonitoring dengan lebih mudah. Selain pemberian notifikasi kepada pemilik, rancang bangun ini mampu memberikan tindakan antisipasi untuk mencegah kebakaran yang akan dikontrol menggunakan *Fuzzy logic*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka dirumuskan permasalahan:

1. Bagaimana cara *monitoring* dan notifikasi kebocoran LPG menggunakan *Fuzzy Logic* berbasis *Internet of Things*?
2. Bagaimana cara merancang kendali gas untuk mencegah kebakaran akibat kebocoran LPG menggunakan *Fuzzy Logic* berbasis *Internet of Things*?

## 1.3 Batasan Masalah

Untuk menghindari pembahasan yang lebih luas terkait dengan perancangan deteksi kebocoran gas berbasis *Android* ini, maka penelitian ini ditentukan pada ruang lingkup tertentu antara lain:

1. Pendekripsi kebocoran gas hanya dilakukan pada ruang lingkup tabung serta regulator.
2. Metode yang digunakan adalah logika *Fuzzy Sugeno*.
3. Sensor yang digunakan adalah TGS2610.

## 1.4 Tujuan

Berdasarkan rumusan masalah yang diuraikan diatas, maka tujuan penelitian ini adalah sebagai berikut:

1. Dapat merancang dan membangun sistem *monitoring* dan notifikasi kebocoran LPG menggunakan *Fuzzy Logic* berbasis *Internet of Things*.
2. Dapat merancang kendali gas untuk mencegah kebakaran akibat kebocoran LPG menggunakan *Fuzzy Logic* berbasis *Internet of Things*.

## 1.5 Manfaat

Tugas Akhir ini diharapkan mampu memberikan peringatan dini pada kasus kebocoran gas LPG serta mengurangi angka kasus ledakan gas LPG. Selain itu Tugas Akhir ini diharapkan mampu menjadi sumber pustaka untuk peneliti selanjutnya yang ingin mengembangkan alat pendekripsi kebocoran gas LPG dimasa depan.

## 1.6 Sistematika Penulisan

Pembahasan Tugas Akhir ini disusun menjadi 5 (lima) garis besar bab pembahasan, yaitu sebagai berikut:

### **BAB I PENDAHULUAN**

Bab ini menjelaskan tentang latar belakang, perumusan masalah, pembatasan masalah, tujuan, manfaat, serta sistematika penulisan pada Tugas Akhir.

## **BAB II LANDASAN TEORI**

Bab ini menjelaskan tentang teori dasar penunjang penelitian yang secara singkat sebagai acuan penelitian Tugas Akhir.

## **BAB III METODE PENELITIAN**

Bab ini membahas tentang tahapan dalam perancangan Tugas Akhir meliputi perancangan sistem, arsitektur mekanisme, arsitektur elektronik, dan arsitektur program.

## **BAB IV HASIL PENGUJIAN DAN PENGAMATAN**

Bab ini membahas tentang pengujian dari masing-masing komponen yang digunakan, meliputi pengujian sensor Figaro TGS2610, *real time clock*, pengujian aplikasi Blynk pada *Android*, sistem *Fuzzy Logic* dan pengujian sistem keseluruhan.

## **BAB V PENUTUP**

Bab ini berisi tentang kesimpulan dari penelitian yang telah dilaksanakan serta saran untuk pengembangan dalam penelitian berikutnya.

## BAB II

### LANDASAN TEORI

#### 2.1 Wemos

Wemos merupakan sebuah *board development system* berfitur *Wi-Fi* sehingga dapat memudahkan dalam pengembangan perangkat *Internet of Things* (IoT) (Prakoso, 2018). CPU pada Wemos dapat diprogram melalui serial *port* maupun melalui OTA serta transfer program secara *wireless*. Hal ini yang membuat Wemos dapat *running stand-alone* (Limantara, et al., 2017). Terdapat 2 tipe Wemos, yaitu Wemos D1 dan Wemos D1 Mini.

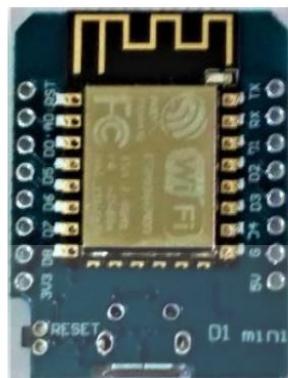
Wemos menjadi pilihan terbanyak daripada modul *Wi-Fi* lainnya. Hal ini disebabkan karena Wemos memiliki beberapa kelebihan. Berikut adalah kelebihan Wemos daripada modul *Wi-Fi* lainnya (Putri, 2017):

- a. Memiliki *Pinout* yang cocok digunakan dengan Arduino Uno.
- b. Memiliki *processor* utama 32 Bit dengan kecepatan 80 MHz sehingga *Frekuensi CPU*-nya cukup tinggi.
- c. Didukung oleh banyak bahasa pemrograman.
- d. Lebih terjangkau daripada modul *Wi-Fi* lainnya.
- e. Mampu untuk *running* tanpa Arduino serta mampu diprogram melalui *over the air* atau transfer program via *wireless*.

##### 2.1.1 Wemos D1 Mini

Wemos D1 Mini merupakan salah satu bentuk variasi dari *microcontroller* Wemos. Wemos D1 Mini memiliki ukuran yang relatif kecil daripada *board*

ESP8266 yang lain. Walaupun demikian, Wemos D1 Mini memiliki keunggulan yaitu dengan tersedianya sumber tegangan 5 Volt, sehingga Wemos D1 Mini dapat terhubung dengan berbagai modul elektronik lain yang membutuhkan tegangan 5 Volt (Putra, et al., 2016). Tampilan fisik Wemos D1 Mini ditunjukkan oleh Gambar 2.1.



Gambar 2. 1 Wemos D1 Mini

Wemos D1 Mini memiliki *Chipset* ESP8266 yang mendukung *stack TCP/IP*. Wemos D1 Mini juga memiliki *Chipset* CH340 yang mengubah USB serial menjadi serial antarmuka. Dalam modul Wemos ini juga terdapat *Pin* digital dan *Pin* analog. *Pin* digital atau I/O *port* modul dapat dikonfigurasi sebagai *input* maupun *output*, sedangkan *Pin* analognya memiliki resolusi sebesar 10 bit dengan nilai maksimal 3,2 Volt (Limantara, et al., 2017). Spesifikasi lengkap Wemos D1 Mini ditampilkan pada Tabel 2.1.

Tabel 2. 1 Spesifikasi Wemos D1 Mini

<i>Microcontroller</i>	<b>ESP8266EX</b>
<i>Operating Voltage</i>	3.3 V
<i>Digital I/O Pins</i>	11
<i>Analog Input Pins</i>	1 (Max <i>input</i> 3,2 V)
<i>Clock Speed</i>	80 MHz/160 MHz
<i>Flash</i>	4 Mbytes
<i>Length</i>	34,2 mm
<i>Width</i>	25,6 mm
<i>Weignt</i>	10g

### 2.1.2 *Chipset* Pada Wemos

Wemos memiliki 2 buah *Chipset* yang digunakan sebagai otak kerja antara lain:

a. *Chipset* ESP8266

*Chipset* ESP8266 merupakan salah satu *Chipset* Wemos yang memiliki fitur *Wi-Fi* serta mendukung *stack* TCP/IP dengan menggunakan *command* yang sederhana. Dengan *clock* 80 MHz, *Chipset* ESP8266 memiliki memori eksternal RAM sebesar 4 MB. *Chipset* ini juga mendukung format IEEE 802.11 b/g/n yang dapat menghindarkan gangguan bagi lainnya (Putri, 2017).

b. *Chipset* CH340

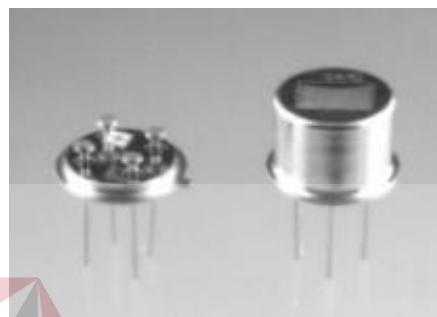
*Chipset* CH340 merupakan *Chipset* yang bertugas untuk mengubah USB serial menjadi serial antarmuka. Dalam mode serial antarmuka, CH340 akan mengirimkan sinyal penghubung. *Chipset* ini mengubah perangkat serial antarmuka umum untuk berhubungan dengan bus USB secara langsung (Putri, 2017).

## 2.2 Sensor Figaro TGS2610

Sensor Figaro TGS2610 merupakan jenis sensor semikonduktor yang memanfaatkan arus rendah. Sensor ini memiliki sensitivitas yang tinggi sehingga

sangat cocok digunakan untuk pendektsian kebocoran gas LPG. Elemen yang digunakan berupa semikonduktor dioksida timah ( $\text{SnO}_2$ ) yang sangat resisten terhadap udara bersih. Jika didapati adanya gas disekitar sensor, maka sensor akan memberikan perubahan resistensinya. Resistensi akan semakin rendah apabila konsentrasi gas LPG yang diterdetksi semakin tinggi (Putra, et al., 2016).

Tampilan fisik sensor Figaro TGS2610 ditunjukkan oleh Gambar 2.2.



Gambar 2. 2 Sensor Figaro TGS2610

Kelebihan sensor Figaro TGS2610 daripada sensor LPG lainnya adalah (Danur, et al., 2013):

- a. Menggunakan daya sebesar 280 mw.
- b. Mampu mendekksi gas antara 500-10.000 PPM.
- c. Tegangan kerja sebesar  $5,0 \pm 0,05$  Vdc.

### 2.3 *Fuzzy Logic*

Pada tahun 1965, Lotfi A. Zedeh mempublikasikan tulisannya yaitu *Fuzzy Sets*. Tulisan ini mendeskripsikan teori himpunan serta perluasannya tentang logika *Fuzzy*. Pada awalnya, penggunaan logika *Fuzzy* masih diragukan oleh para ilmuan walaupun sudah banyak aplikasi *Fuzzy* yang berhasil dioperasikan (Aryandhi & Talakua, 2013). *Fuzzy Logic* adalah salah satu komponen yang membentuk *Soft*

*Computing*. Dasar logika *Fuzzy* sendiri ialah teori himpunan *Fuzzy*. Pada teori ini, fungsi dari derajat keanggotaan sangat penting yaitu sebagai penentu keberadaan elemen dalam suatu himpunan (Yanto, 2017).

### 2.3.1 *Fuzzy Sugeno*

Metode *Sugeno* merupakan salah satu metode penalaran yang memiliki *output* (konsekuensi) sistem tidak berupa himpunan *Fuzzy* melainkan berupa konstanta atau persamaan linier. Metode *Sugeno* diperkenalkan pertama kali pada tahun 1985 oleh Takagi-Sugeno Kang (Kusumadewi, 2004). *Fuzzy Sugeno* menggunakan fungsi keanggotaan sederhana yang disebut *singleton*. *Singleton* sendiri merupakan fungsi keanggotaan yang memiliki derajat keanggotaan 1 dan 0. Nilai 1 diberikan pada suatu nilai *crisp* tunggal dan nilai 0 diberikan pada semua nilai *crisp* yang lain (Febrianto, et al., 2016).

Sistem *Fuzzy Sugeno* memperbaiki sistem *Fuzzy* murni yaitu dengan menambah suatu perhitungan matematika sederhana sebagai bagian THEN. System *Fuzzy* memiliki suatu nilai rata-rata tertimbang (*Weighted Average Values*) pada bagian aturan *Fuzzy* IF-THEN. Namun, sistem *Fuzzy Sugeno* juga memiliki keterbatasan, adanya perhitungan matematika menyebabkan sistem ini tidak dapat menghasilkan kerangka alami untuk direpresentasikan pengetahuan manusia. Selain itu, sistem ini tidak mampu memberikan kebebasan untuk menggunakan prinsip yang berbeda dalam logika *Fuzzy* (Kusumadewi, 2004).

Keunggulan metode *Sugeno* adalah sebagai berikut (Kusumadewi, 2004):

- a. Lebih efisien.
- b. Baik untuk teknik linear.

- c. Baik untuk teknik optimasi maupun adaptif.
- d. Kontinuitas permukaan output berjalan baik.
- e. Cocok untuk analisa matematik.

Sedangkan model-model *Fuzzy Sugeno* meliputi (Rofiq, 2013):

- a. Model *Fuzzy Sugeno* Orde-Nol

$$\text{IF } (x_1 \text{ is } A_1) \cdot (x_2 \text{ is } A_2) \cdot (x_3 \text{ is } A_3) \cdot \dots \cdot$$

$$(x_N \text{ is } A_N) \text{ THEN } z = k$$

Dengan  $A_i$  merupakan himpunan *Fuzzy* ke- $i$  sebagai *antecedent* dan  $k$  merupakan sebuah konstanta (tegas) sebagai konsekuensi.

- b. Model *Fuzzy Sugeno* Orde-Satu

$$\text{IF } (x_1 \text{ is } A_1) \cdot \dots \cdot (x_N \text{ is } A_N) \text{ THEN } z =$$

$$p_1 * x_1 + \dots + p_N * x_N + q$$

Dengan  $A_i$  merupakan himpunan *Fuzzy* ke- $i$  sebagai *antecedent* dan  $p_i$  merupakan sebuah konstanta (tegas) ke- $i$  dan  $q$  merupakan konstanta dalam konsekuensi.

## 2.4 LCD M1632

LCD M1632 adalah antarmuka antara pengguna dan sistem *microcontroler* dengan tampilan 16 x 2 karakter. LCD ini dikendalikan oleh *microcontroler* HD 44780 dengan DDRAM (*Display Data Access Memory*) sebagai memori dan CGROM (*Character Generator Read Only Memory*) sebagai tempat penyimpanan karakter yang telah permanen. Modul LCD ini memiliki 16 kaki untuk berkomunikasi dengan sistem *microcontroler* yang terdiri atas 3 kaki kontrol, 8 kaki data dan 5 kaki suplai tegangan (Soemarsono, et al., 2015).

Terdapat 2 jenis LCD M1632 yaitu LCD M1632 *refubrish* dan LCD M1632 *Hitachi*. Pada LCD M1632 *refubrish*, Pin nomor 1 diberikan untuk tegangan 5 Volt dan nomor 2 untuk tegangan 0 Volt, sedangkan pada LCD M1632 *Hitachi* berlaku sebaliknya (Soemarsono, et al., 2015). Bentuk fisik dari LCD M1632 ditunjukkan oleh Gambar 2.3.



Gambar 2. 3 Tampilan LCD 16x2

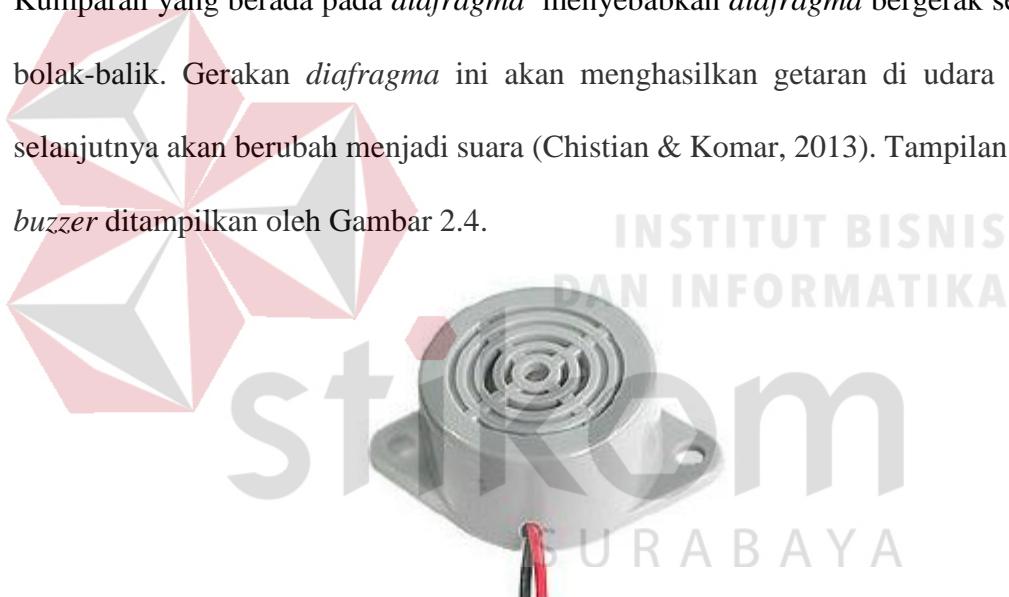
Spesifikasi LCD M1632 adalah sebagai berikut (Riyadi & Purnama, 2013):

- a. Memiliki tampilan karakter 16 x 2 dengan matrik 5 x 7 + kursor.
- b. Memiliki ROM pembangkit karakter 192 jenis.
- c. Memiliki RAM pembangkit karakter 8 jenis (diprogram pemakai).
- d. Memiliki RAM data tampilan 80 x 8 bit (*8 karakter*).
- e. *Duty ratio 1/16.*
- f. Mampu membaca RAM data tampilan dan RAM pembangkit karakter melalui unit *microprosesor*.
- g. Fungsi perintah yang sering digunakan yaitu penghapusan tampilan (*display clear*), posisi kursor awal (*cursor home*), tampilan karakter kedip (*display character blink*), penggeseran kursor (*cursor shift*) dan penggeseran tampilan (*display shift*).
- h. Rangkaian pembangkit detak.
- i. Rangkaian otomatis *reset* saat daya dinyalakan.
- j. Catu daya tunggal +5 Volt.

## 2.5 *Buzzer*

*Buzzer* berfungsi untuk mengubah getaran listrik menjadi getaran suara. Kemampuannya menghasilkan suara, menyebabkan *buzzer* sering dimanfaatkan sebagai alarm. Alarm yang dimaksud adalah penanda bahwa sebuah proses sudah selesai maupun terdapat hal yang salah dari proses tersebut (Soemarsono, et al., 2015).

*Buzzer* terdiri dari kumparan yang kemudian dialiri arus listrik. Kumparan tersebut akan tertarik sesuai arus listrik yang diberikan serta *polaritas* magnetnya. Kumparan yang berada pada *diafragma* menyebabkan *diafragma* bergerak secara bolak-balik. Gerakan *diafragma* ini akan menghasilkan getaran di udara yang selanjutnya akan berubah menjadi suara (Christian & Komar, 2013). Tampilan fisik *buzzer* ditampilkan oleh Gambar 2.4.



Gambar 2. 4 Buzzer

## 2.6 *Real Time Clock*

*Real Time Clock* (RTC) merupakan jenis waktu yang berdasarkan *real time* atau waktu yang sebenarnya. RTC membutuhkan dua parameter utama, yaitu parameter yang ditentukan pada saat memulai (*start*) dan pada saat berhenti (*stop*).

RTC berupa sirkuit yang memiliki catu daya terpisah, sehingga masih akan tetap berkerja walaupun catu daya utama sudah tidak bekerja (Alpares, 2016).

Salah satu modul RTC adalah DS3231, DS3231 adalah jenis RTC yang sudah terintegrasi dengan kompensasi suhu Kristal osilator (TCX0). TCX0 memiliki *clock* referensi stabil dan akurat. TCX0 dapat mengontrol akurasi RTC hingga sekitar 2 menit dalam satu tahun. DS3231 memiliki waktu serta kalender dengan dua waktu pengingat dalam satu hari. Waktu tersebut memberikan informasi mulai dari detik hingga jam serta hari hingga tahun yang sudah terdaftar pada register internal (Alpares, 2016).

Format waktu pada DS3231 bisa berupa format dalam 24 jam maupun 12 jam (AM/ PM). Sinkronisasi komunikasi *serial* I2C dengan kecepatan *clock* 400Khz dapat menyederhanakan proses tatap muka antara RTC dengan suatu *microprosesor*. RTC membutuhkan 2 saluran untuk komunikasi dengan *clock/RAM*. Kedua saluran tersebut adalah SCL (*serial clock*) dan SDA (*Serial I/O data*), serta dilengkapi dengan keluaran SQW/*Out*. *Output* berupa SQW/*Out* bisa diprogram untuk mengetahui perubahan data waktu pada RTC dan *Pin RST*. DS3231 hanya membutuhkan *power* yang sangat rendah serta mempertahankan data dan informasi waktu  $\pm 1$  microwatt.

Berikut merupakan karakteristik RTC tipe DS3231 yaitu:

- a. RTC dapat menghitung detik, menit, jam, tanggal, bulan, hari pada setiap minggu dan setiap tahun hingga tahun 2100.
- b. Serial I2C menggunakan *Pin* minimum untuk proses komunikasi RTC.
- c. Memiliki 2.0 –5.5 Volt *full operation*.
- d. Mempunyai 16 *Pin* SOICs.

- e. 3 simple wire interface ( I2C dan SQW/Out).
- f. Mampu memprogram square wave output.
- g. Mempunyai sensor temperatur dengan akurasi  $\pm 3^{\circ}\text{C}$ , Gambar 2.5 menunjukkan tampilan fisik RTC.



Gambar 2. 5 Real Time Clock

## 2.7 Motor Driver

*Motor Driver* yang berbasis H-Bridge adalah *driver* L298N. Pada tegangan 6V-46V, driver L298N mampu menangani beban sampai 4A. Terdapat dua rangkaian H-Bridge pada satu *chip*. *Driver* L298N mampu mengendalikan arah putaran serta kecepatan *motor DC* ataupun *Motor stepper*. Selain itu, *driver* ini juga mampu mengeluarkan *output* tegangan sebesar 50 Volt. IC L298 memiliki *transistor-transistor Logic (TTL)* dengan gerbang NAND sehingga mampu memudahkan penentuan arah putaran suatu *motor DC* dan *motor stepper*. *Driver* ini mampu mengendalikan 2 *motor DC* namun hanya mampu mengendalikan 1 *motor stepper* (Prabowo, 2017). Gambar fisik *driver* L298N ditampilkan pada Gambar 2.6.



Gambar 2. 6 *Motor Driver L298N*

## 2.8 Kipas DC

Kipas angin berfungsi sebagai pendingin udara, penyegar udara, ventilasi (*exhaust fan*), maupun pengering. Kipas angin juga ditemukan di mesin penyedot debu dan berbagai ornamen untuk dekorasi ruangan. Kipas angin memiliki 3 komponen utama, yaitu (Prabowo, 2017):

- a. Motor Pergerak

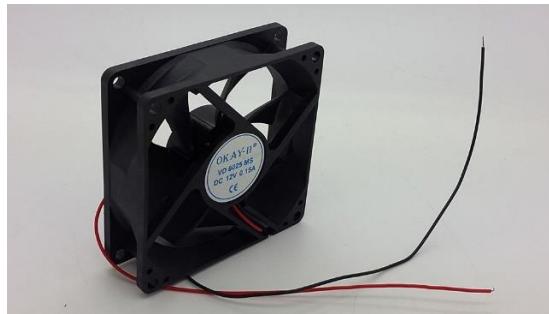
Motor kapasitor merupakan jenis motor pergerak yang sering digunakan. Motor ini memiliki kumparan utama dan kumparan bantu yang dirangkai secara seri.

- b. Bagian Kipas

Bagian ini adalah bagian yang berbentuk baling-baling. Bagan ini akan berputar pada poros tengah kipas. Bagian ini biasanya dilindungi oleh rumah kipas.

- c. Rumah Motor

Bagian ini merupakan tempat dudukan yang berfungsi untuk menampung semua komponen pada kipas angin. Rumah motor ini biasanya terbuat dari bahan ebonite, Gambaran fisik dari kipas akan ditampilkan pada Gambar 2.7.



Gambar 2. 7 Kipas DC

## 2.9 *Modul Relay*

*Modul relay* merupakan saklar elektrik yang dapat menghidupkan atau mematikan sirkuit dengan *voltage* atau arus tinggi. *Relay* disusun oleh inti besi yang dibungkus oleh kumparan kawat, kuk besi yang berfungsi menahan alur *flux magnetic*, angker (jangkar) besi. Angker, kuk dan kotak yang bergerak saling terhubung secara mekanik. Pegas menahan angker, sehingga akan ada celah udara udara ketika *relay* di putus.

Angker akan diaktifkan melalui medan elektromagnetik yang dihasilkan oleh arus listrik dari kumparan. Angker yang aktif akan bergerak bersamaan dengan kotak. Pergerakan kontak menyebabkan hubungan putus-sambung dengan kontak yang diam. Ketika *coil* sudah tidak dialiri arus, pegas akan membuat angker kembali seperti semula.

*Relay* melindungi antar masing-masing rangkaian. Pada masing-masing saluran pada *modul relay*, memiliki tiga koneksi bernama NC, COM, dan NO. Bagian NC dan NO *relay* berfungsi sebagai penghubung sumber listrik (kabel *fase*) dengan terminal SPO. *Normally Closed* (NC) digunakan sebagai kontak sehingga dapat menutup sambungan sumber ke SPO pada kondisi arus normal, sedangkan

ketika arus melebihi, kontak otomatis diputuskan (*open*). Kumparan (*coil*) *relay* dihubungkan menuju *Pin pengendali microcontroller* melalui *switch transistor*.

Pada *Relay*, terdapat istilah *pole* dan *throw*. *Pole* merupakan jumlah kontak pada sebuah *relay*, sedangkan *throw* merupakan kondisi dari kontak tersebut. *Relay* dapat digolongkan berdasar jumlah *pole* dan *thrownya*. Berikut penggolongan dari *relay* (Saleh & Haryanti, 2017):

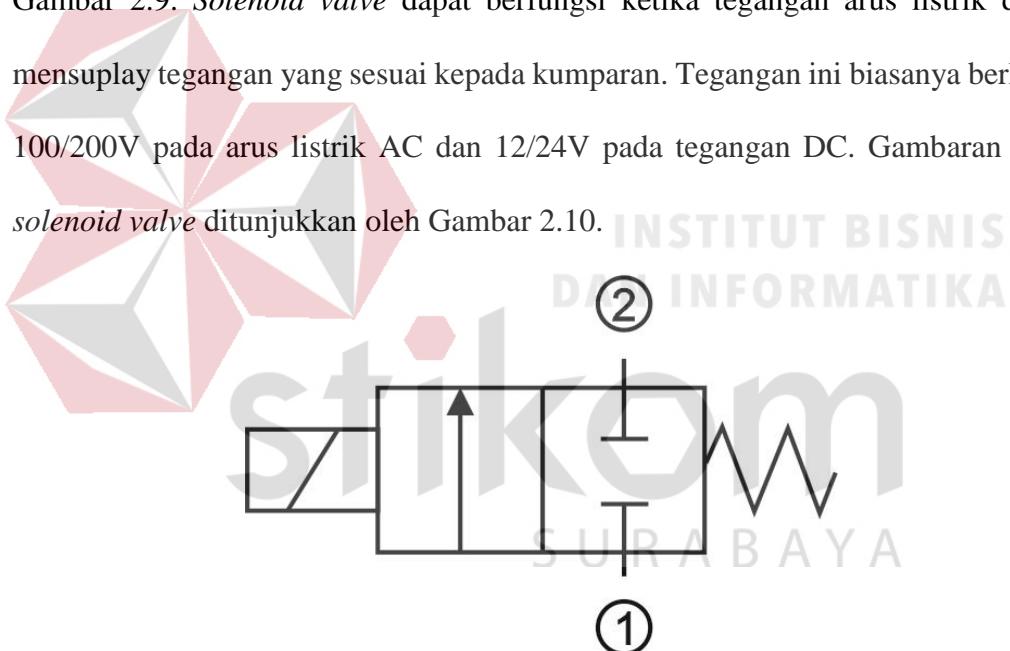
- a. SPST (*Single Pole Single Throw*): terdapat 4 Terminal, 2 Terminal untuk Saklar dan 2 Terminalnya lagi untuk *coil*.
- b. SPDT (*Single Pole Double Throw*): terdapat 5 Terminal, 3 Terminal untuk Saklar dan 2 Terminalnya lagi untuk *coil*.
- c. DPST (*Double Pole Single Throw*): terdapat 6 Terminal, 4 Terminal terdiri dari 2 Pasang Terminal Saklar dan 2 Terminal untuk *coil*. *Relay DPST* mampu menjadi 2 Saklar yang dikendalikan oleh 1 *coil*.
- d. DPDT (*Double Pole Double Throw*): terdapat 8 Terminal, 6 Terminal adalah 2 pasang *relay SPDT* dan 2 Terminal untuk *coil*. Rangkaian *relay* divisualisasikan seperti diperlihatkan pada Gambar 2.8.



Gambar 2. 8 Modul *Relay*

## 2.10 Solenoid Valve Gas

*Solenoid valve* merupakan elemen kontrol pada sistem *fluida*. *Solenoid valve* dikontrol oleh arus listrik AC ataupun DC oleh kumparan/*solenoida*. *Solenoid valve* dapat digunakan pada beberapa sistem. Salah satu contohnya pada sistem *pneumatik*, *solenoid valve* dapat digunakan sebagai pengendali saluran udara *cylinder*. *Solenoid valve* didesain khusus sesuai kegunaannya, sehingga *solenoid* ini memiliki berbagai *type*, yaitu *solenoid* dengan 2 saluran, 3 saluran, 4 saluran dan sebagainya. Gambaran desain *solenoid valve* dengan 2 saluran ditunjukkan oleh Gambar 2.9. *Solenoid valve* dapat berfungsi ketika tegangan arus listrik dapat mensupply tegangan yang sesuai kepada kumparan. Tegangan ini biasanya berkisar 100/200V pada arus listrik AC dan 12/24V pada tegangan DC. Gambaran fisik *solenoid valve* ditunjukkan oleh Gambar 2.10.



Gambar 2. 9 Diagram Solenoid Valve Gas 2 Saluran



Gambar 2. 10 Solenoid Valve Gas

## 2.11 Blynk

Blynk adalah *platform* baru yang digunakan untuk mengontrol Arduino, Raspberry Pi, dan lain sebagainya. Blynk sendiri bisa digunakan pada sistem iOS maupun *Android*. Blynk memungkinkan dengan cepat membangun *interface* untuk mengendalikan dan memantau proyek *hardware* dari iOS dan perangkat *Android* (Madjid, 2019). Blynk merupakan *dashboard* digital dengan antarmuka grafis pada sebuah *project*. Pada Blynk, terdapat fitur *Drag and Drop* yang digunakan untuk menambahkan komponen *Input/Output* walaupun pengguna tidak mampu membuat pemrograman *Android* maupun iOS.

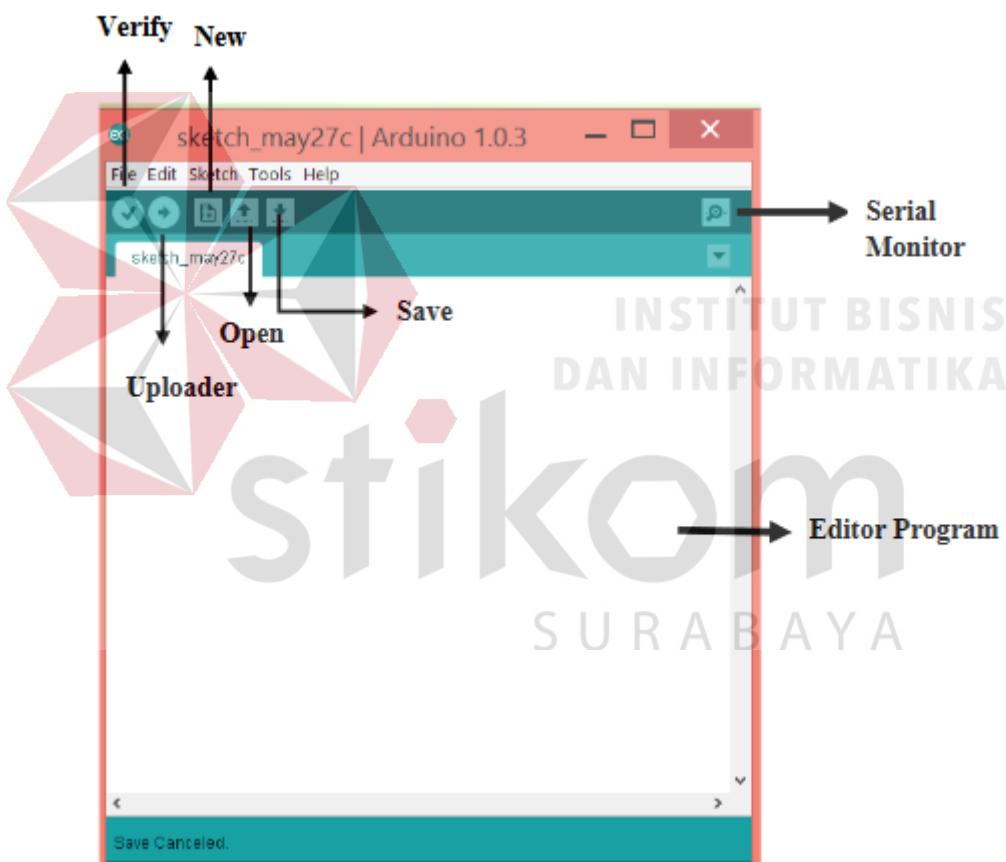
Blynk dapat membuat *server local*, yang memungkinkan untuk mengontrol setiap jaringan dalam rumah. Blynk dapat mendukung beberapa *microcontroller* seperti Arduino, *Partikel Core*, model *Raspberry Pi*, ESP8266, dan lain sebagainya. Namun, penggunaan aplikasi ini masih sangat bergantung pada *Wi-Fi* maupun *Ethernet* lainnya. Hal ini tentunya juga dapat diantisipasi dengan cara menghubungkan pada *port USB* komputer. Tampilan aplikasi Blynk ditunjukkan oleh Gambar 2.11.



Gambar 2. 11 Tampilan Aplikasi Blynk

## 2.12 Arduino Software IDE

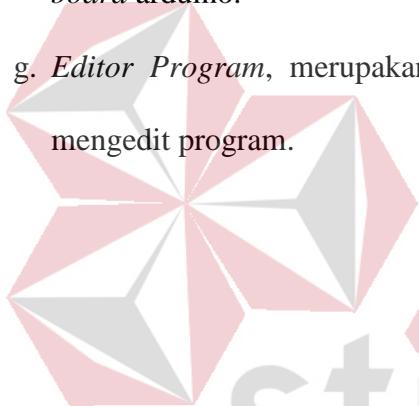
Arduino merupakan sistem elektronika yang berbasis *open-source*. Arduino terbagi menjadi perangkat *hardware* dan juga perangkat *software*. Perangkat *software* Arduino menggunakan *driver* dan IDE. IDE (*Integrated Development Environment*) adalah program komputer yang memungkinkan untuk merancang program pada *board* arduino (Aslamia, 2015). Tampilan *Software Arduino IDE* ditunjukkan oleh Gambar 2.12.



Gambar 2. 12 Tampilan *Software Arduino IDE*

Pada Gambar 2.12 terdapat beberapa fitur pada *Software Arduino IDE* yang dijelaskan sebagai berikut (Aslamia, 2015):

- a. *Verify*, digunakan untuk melihat kode *sketch* yang mengalami *error* sebelum mengupload ke *board arduino*.
- b. *New*, berfungsi untuk membuat *sketch* baru.
- c. *Uploader*, berguna untuk membuat kode biner dari komputer menuju memori pada *board arduino*.
- d. *Open*, untuk membuka *sketch* yang sudah tersimpan pada *sketchbook arduino*.
- e. *Save*, digunakan untuk menyimpan *sketch* pada *sketchbook arduino*.
- f. *Serial Monitor*, berguna untuk menampilkan data serial yang dikirimkan dari *board arduino*.
- g. *Editor Program*, merupakan *window* yang berguna untuk menulis maupun mengedit program.



INSTITUT BISNIS  
DAN INFORMATIKA  
**stikom**  
SURABAYA

## **BAB III**

### **METODOLOGI PENELITIAN**

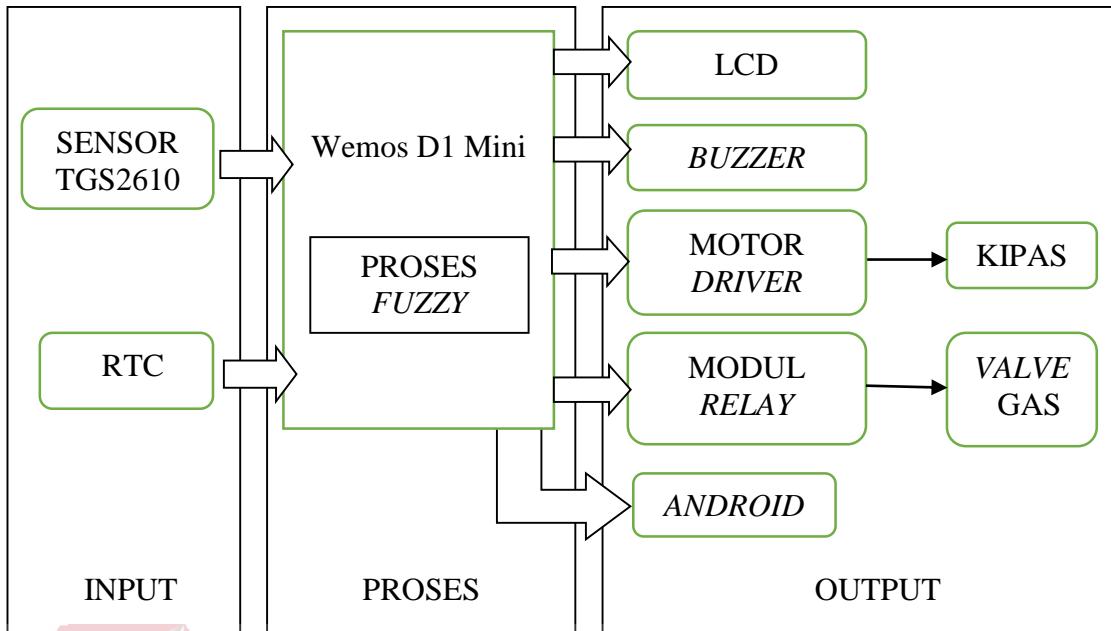
#### **3.1 Metode Penelitian**

Metode pada pembuatan rancang bangun ini diawali dengan instalasi aplikasi Blynk. Aplikasi ini dapat di-*download* pada *Play Store* pada *smartphone Android*. Setelah berhasil di-*instal*, *user* harus masuk ke aplikasi dengan cara *login* menggunakan *username* serta *password*. Setelah itu, *user* sudah dapat membuat projek baru yang terhubung dengan *microcontroller* Wemos D1 Mini .

Rancang bangun ini menggunakan Wemos D1 Mini yang telah terhubung dengan koneksi internet melalui modul *Wi-Fi* ESP8266, sedangkan sensor yang digunakan adalah TGS2610. Data yang dibaca oleh sensor, nantinya akan diolah pada *microcontroller* menggunakan *Fuzzy Logic*.

#### **3.2 Diagram Hardware**

Secara umum perancangan perangkat pada rancang bangun ini akan digambarkan pada Gambar 3.1.



Gambar 3. 1 Blok Diagram Sistem Deteksi Kebocoran LPG

Alur pada Gambar 3.1 dijelaskan sebagai berikut:

1. *Input*
  - a. TGS2610: Modul ini bertugas untuk mendeteksi kebocoran gas LPG.
  - b. RTC: Modul ini berfungsi untuk mengatur waktu secara *realtime*.
2. *Proses*
  - a. Wemos D1 Mini: *Microcontroller* berfungsi mengolah data yang dikirimkan oleh modul sensor.
  - b. *Fuzzy Logic*: Berfungsi memberikan *rule* pada proses pengolahan data yang akan mempengaruhi *output*.
3. *Output*
  - a. LCD: berfungsi sebagai penampil data kebocoran gas.
  - b. *Buzzer*: berfungsi sebagai alarm saat terjadi kebocoran gas.
  - c. *Motor Driver*: berfungsi untuk mengatur kecepatan Kipas DC.

- d. Kipas DC: bertugas untuk mengeluarkan gas dari ruangan.
- e. *Modul Relay*: berfungsi sebagai saklar elektrik untuk mengaktifkan *valve gas*.
- f. *Valve Gas*: berfungsi untuk menutup aliras gas pada selang jika terjadi kebocoran.
- g. *Android*: berfungsi untuk *monitoring* melalui aplikasi Blynk secara *real-time* oleh pemilik rumah.

### 3.3 Perancangan Mekanik

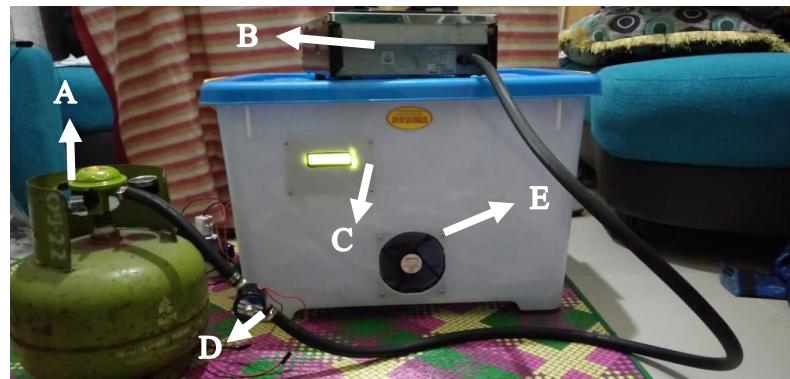
Rancangan mekanik pada rancang bagun ini adalah sebagai berikut:

1. Berikut merupakan desain mekanik dari rancang bangun pendekripsi kebocoran gas sebagaimana yang ditampilkan pada Gambar 3.2.

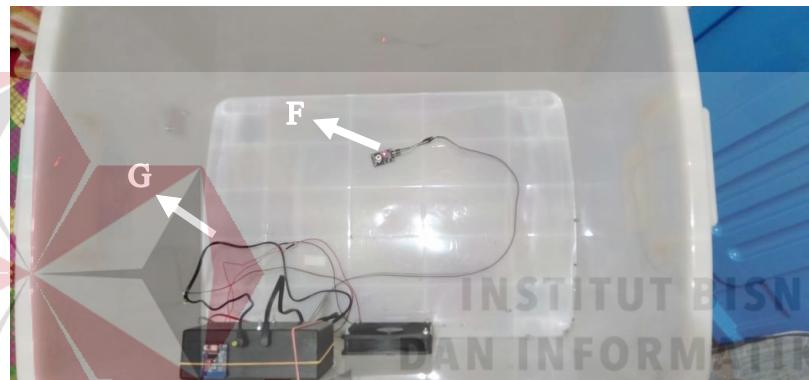


Gambar 3. 2 Desain Rancang Mekanik

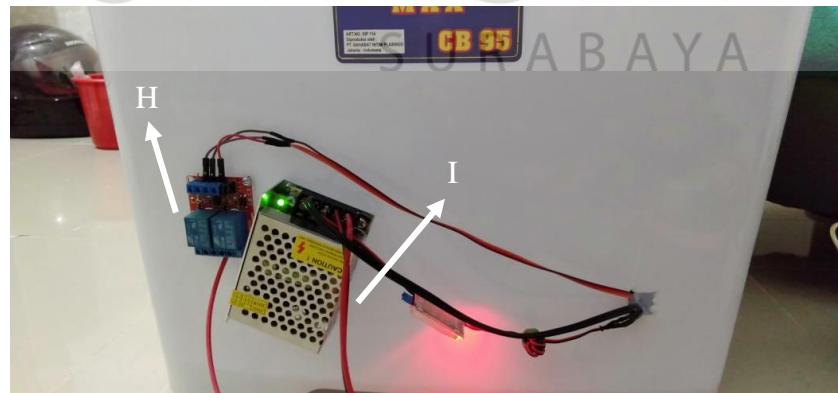
2. Berikut merupakan hasil mekanik dari rancang bangun pendekripsi kebocoran gas yang ditampilkan pada Gambar 3.3.



(A)



(B)



(C)

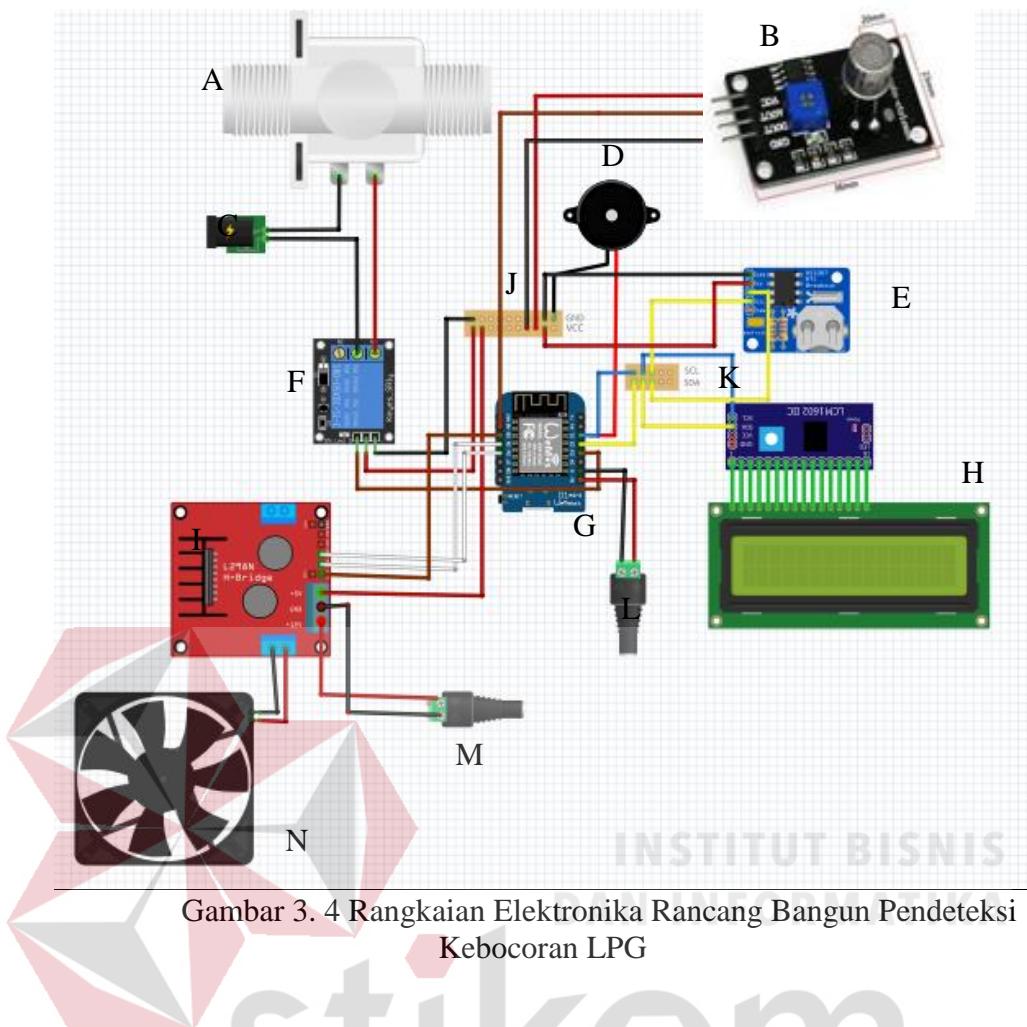
Gambar 3. 3 Hasil Rancang Mekanik (A) Tampak Luar, (B) Tampak Dalam, Dan (C) Tampak Samping

Keterangan:

- A. LPG merupakan sumber energi.
- B. Kompor untuk menyalakan api.
- C. LCD berfungsi sebagai penampil data kebocoran gas.
- D. *Valve Gas* berfungsi untuk menutup aliran gas pada selang jika terjadi kebocoran.
- E. Kipas DC bertugas untuk mengeluarkan gas dari ruangan.
- F. Sensor TGS2610 bertugas untuk mendeteksi kebocoran gas LPG.
- G. Wemos D1 Mini merupakan *Microcontroller* yang berfungsi mengolah data yang dikirimkan oleh modul sensor.
- H. *Relay* berfungsi sebagai saklar elektrik untuk mengaktifkan *valve gas*.
- I. *Power Supply* berfungsi sebagai sumber daya listrik.

### 3.4 Perancangan Elektronika

Rangkaian elektronika secara keseluruhan dari rancang bangun pendekripsi kebocoran LPG ini ditunjukkan oleh Gambar 3.4.

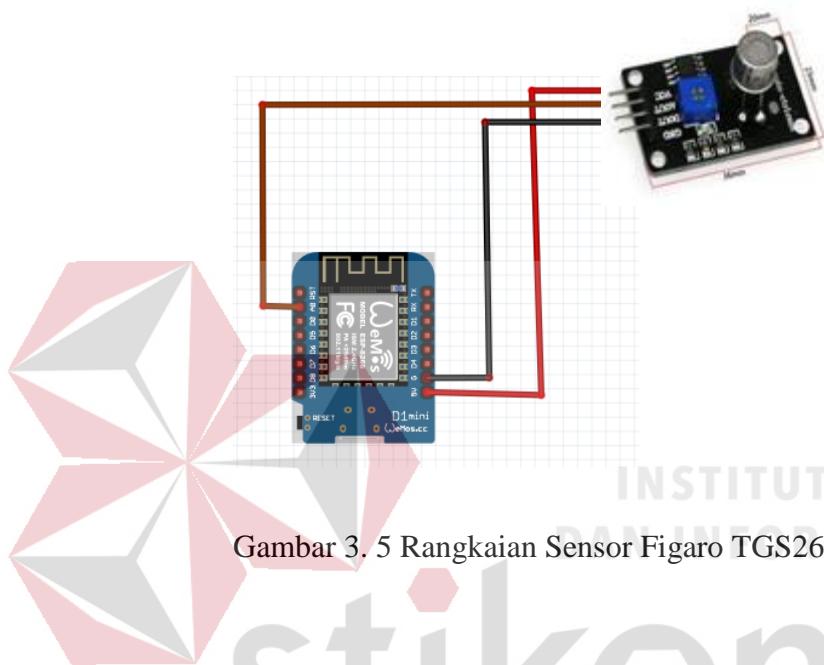


Keterangan :

- |                          |                       |
|--------------------------|-----------------------|
| A. Valve                 | H. LCD I2C            |
| B. Sensor Figaro TGS2610 | I. Motor Driver L298N |
| C. Tegangan AC 220 Volt  | J. Rangkaian VCC /GND |
| D. Buzzer                | K. Rangkaian SCL/SDA  |
| E. RTC                   | L. Adaptor 5 Volt     |
| F. Relay                 | M. Adaptor 12 Volt    |
| G. Wemos D1 Mini         | N. Kipas DC           |

### 3.4.1 Rangkaian Sensor Figaro TGS2610

Pada rangkaian rancang bangun ini, sensor Figaro TGS2610 akan dihubungkan dengan Wemos D1 Mini. Rangkaian sensor ini terdiri dari 4 *Pin* yaitu VCC, GND, A0 dan D0. Rangkaian sensor Figaro TGS2610 ditunjukkan oleh Gambar 3.5.



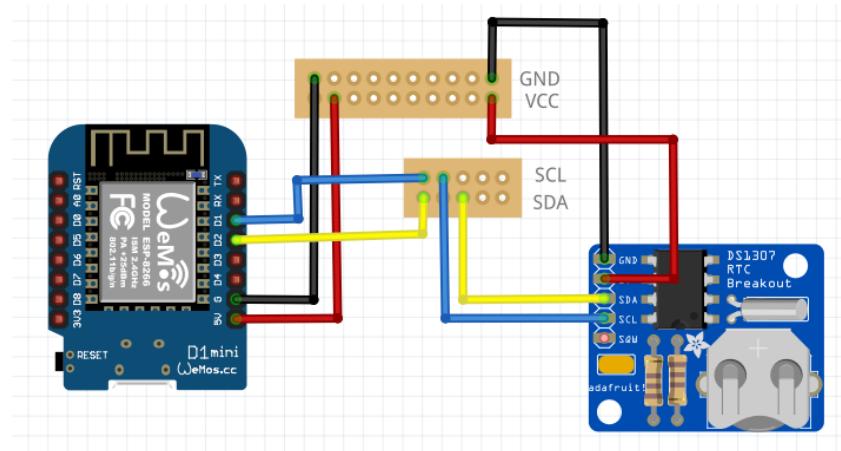
Gambar 3. 5 Rangkaian Sensor Figaro TGS2610

Keterangan:

- Pin* A0 dari Wemos terhubung dengan *Pin* AOUT pada sensor.
- Pin* VCC dari sensor terhubung dengan *Pin* 5 Volt Wemos D1 Mini.
- Pin* GND dari sensor terhubung dengan *Pin* GDN Wemos D1 Mini.

### 3.4.2 Rangkaian RTC

Pada rangkaian rancang bangun ini, RTC akan dihubungkan dengan Wemos D1 Mini. Rangkaian RTC ini terdiri dari 4 *Pin* yaitu SCL, SDA, VCC dan GND. Rangkaian modul RTC ditunjukkan oleh Gambar 3.6.



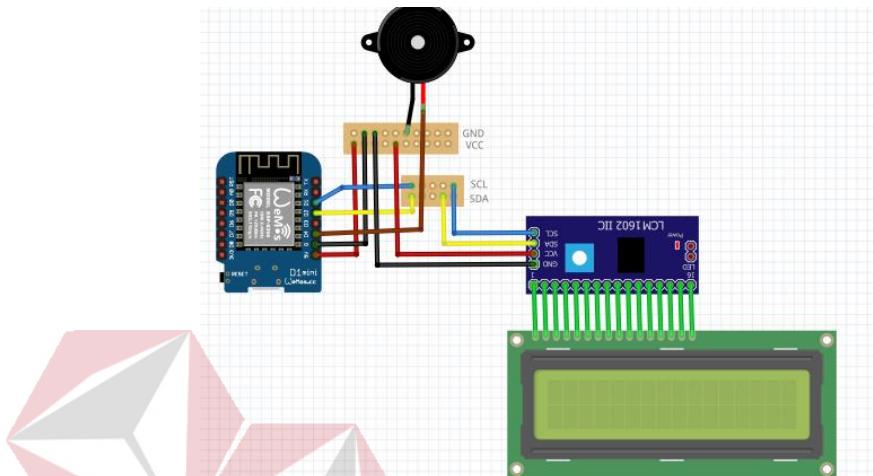
Gambar 3. 6 Rangkaian RTC

Keterangan:

- Pin SDA dari RTC terhubung dengan Pin SDA/D2 pada rangkaian SCL/SDA.*
- Pin SCL dari RTC terhubung dengan Pin SCL/D1 pada rangkaian SCL/SDA.*
- Pin SDA dari rangkaian SCL/SDA terhubung dengan Pin SDA dari Wemos D1 Mini.*
- Pin SCL dari rangkaian SCL/SDA terhubung dengan Pin SCL dari Wemos D1 Mini.*
- Pin GND dari RTC terhubung dengan GND pada rangkaian VCC/GND.*
- Pin VCC dari RTC terhubung dengan VCC pada rangkaian VCC/GND.*
- Pin VCC dari rangkaian VCC/GND terhubung dengan Pin 5V dari Wemos D1 Mini.*
- Pin GND dari rangkaian VCC/GND terhubung dengan Pin GND dari Wemos D1 Mini.*

### 3.4.3 Rangkaian LCD Dan Buzzer

Pada rangkaian rancang bangun ini, LCD I2C dan Buzzer akan dihubungkan dengan Wemos D1 Mini. Rangkaian LCD I2C ini terdiri dari 4 *Pin* yaitu SCL, SDA, VCC dan GND. Rangkaian LCD dan *buzzer* ditunjukkan oleh Gambar 3.7.



Gambar 3. 7 Rangkaian LCD Dan Buzzer

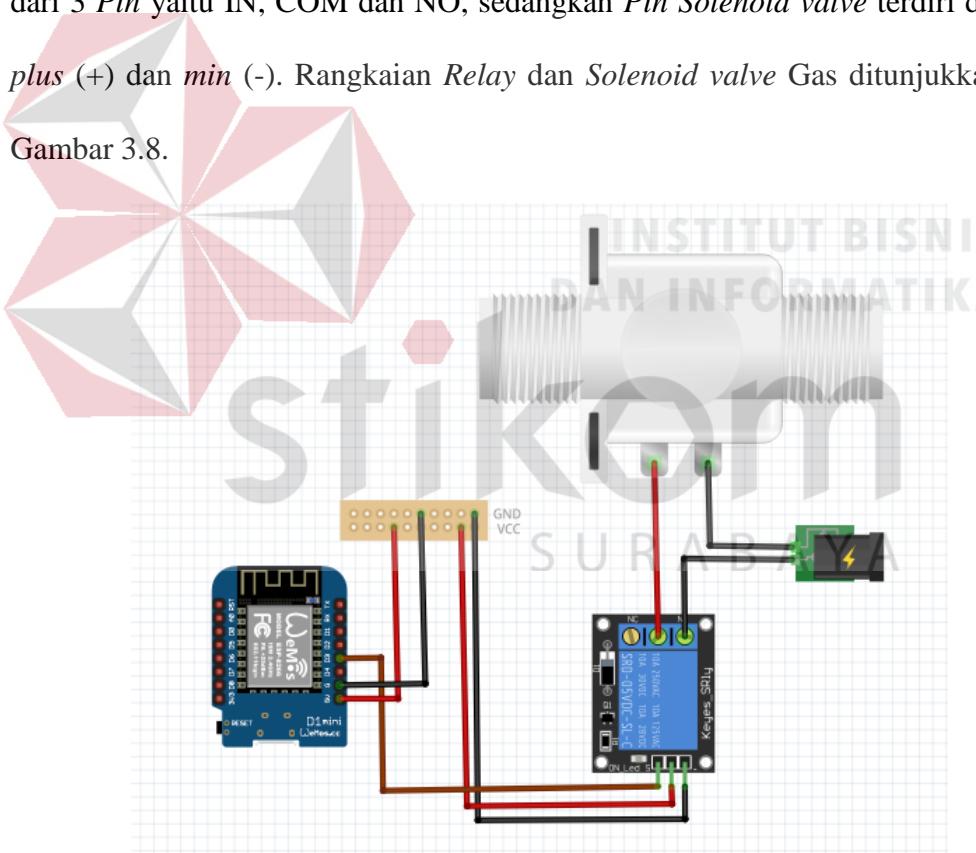
Keterangan:

- Pin* GND dari LCD I2C terhubung dengan GND pada rangkaian VCC/GND.
- Pin* VCC dari LCD I2C terhubung dengan VCC pada rangkaian VCC/GND.
- Pin* VCC dari rangkaian VCC/GND terhubung dengan *Pin* 5V dari Wemos D1 Mini.
- Pin* GND dari rangkaian VCC/GND terhubung dengan *Pin* GND dari Wemos D1 Mini.
- Pin* SCL dari LCD I2C terhubung dengan SCL pada rangkaian SCL/SDA.
- Pin* SDA dari LCD I2C terhubung dengan SDA pada rangkaian SCL/SDA.
- Pin* SCL dari rangkaian SCL/SDA terhubung dengan *Pin* SCL/D1 dari Wemos D1 Mini.

- h. Pin SDA dari rangkaian SCL/SDA terhubung dengan Pin SDA/D2 dari Wemos D1 Mini.
- i. Pin plus (+) Buzzer terhubung dengan Pin D4 Wemos D1 Mini.
- j. Pin min (-) Buzzer terhubung dengan Pin GND dari rangkaian VCC/GND.

#### 3.4.4 Rangkaian Relay Dan Solenoid Valve Gas

Pada rangkaian rancang bangun ini, *Relay* dan *Solenoid valve* akan dihubungkan dengan Wemos D1 Mini dan sumber tegangan 220 Volt. *Relay* terdiri dari 3 Pin yaitu IN, COM dan NO, sedangkan *Pin Solenoid valve* terdiri dari *Pin plus* (+) dan *min* (-). Rangkaian *Relay* dan *Solenoid valve* Gas ditunjukkan oleh Gambar 3.8.



Gambar 3. 8 Rangkaian Modul *Relay* Dan *Solenoid Valve* Gas

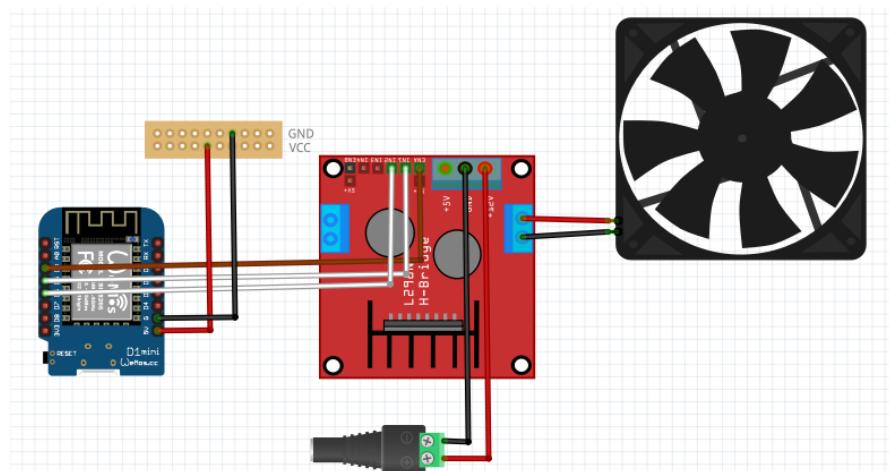
Keterangan:

- a. Pin IN dari *Relay* terhubung dengan Pin D3 pada Wemos D1 Mini.
- b. Pin COM dari *Relay* terhubung dengan Pin *min* (-) pada tegangan 220V.

- c. Pin NO dari *Relay* terhubung dengan Pin *plus* (+) pada tegangan 220V.
- d. Pin *plus* (+) dari *Solenoid Valve* terhubung dengan Pin NO pada *Relay*.
- e. Pin *min* (-) dari *Solenoid Valve* terhubung dengan Pin *plus* (+) pada tegangan 220V.
- f. Pin VCC dari *Relay* terhubung dengan VCC pada rangkaian VCC/GND.
- g. Pin GND dari *Relay* terhubung dengan GND pada rangkaian VCC/GND.
- h. Pin VCC dari rangkaian VCC/GND terhubung dengan Pin 5V dari Wemos D1 Mini.
- i. Pin GND dari rangkaian VCC/GND terhubung dengan Pin GND dari Wemos D1 Mini.

#### 3.4.5 Rangkaian Motor *Driver* Dan Kipas DC

Pada rangkaian rancang bangun ini, *Motor Driver* dan Kipas DC akan dihubungkan dengan Wemos D1 Mini dan adaptor 12 Volt. *Motor driver* terdiri dari 3 Pin yaitu IN A.1, IN A.2, EN, 12V, GND, *min* (-) dan *plus* (+), sedangkan Pin Kipas DC terdiri dari Pin *plus* (+) dan *min* (-). Rangkaian Motor *Driver* dan Kipas DC ditunjukkan oleh Gambar 3.9.



Gambar 3. 9 Rangkaian Motor *Driver* Dan Kipas DC

Keterangan:

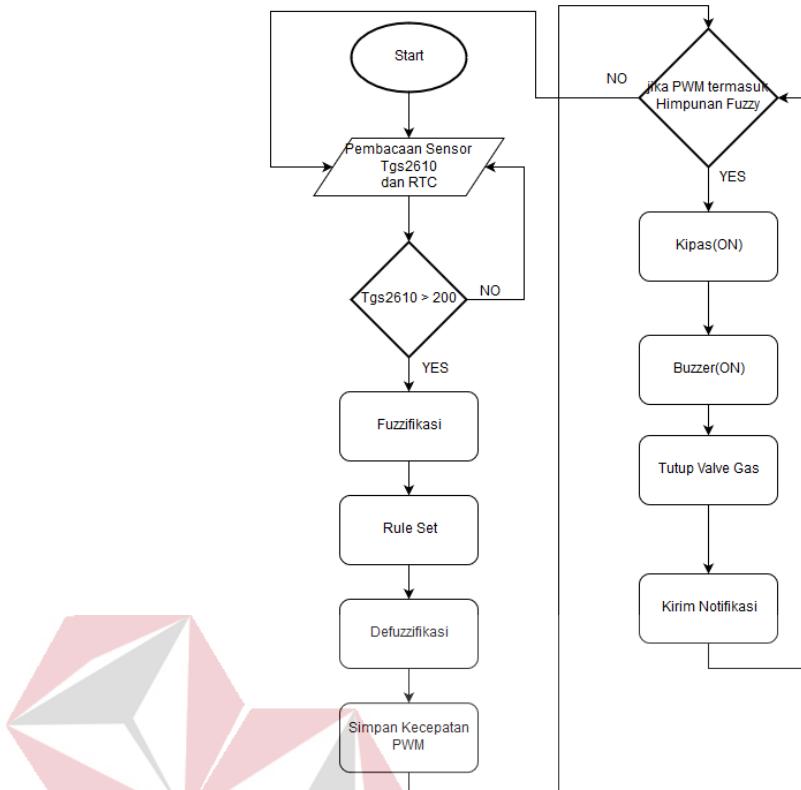
- a. *Pin GND* dari Wemos D1 Mini terhubung dengan *Pin GND* rangkaian VCC/GND.
- b. *Pin VCC* dari Wemos D1 Mini terhubung dengan *Pin VCC* rangkaian VCC/GND.
- c. *Pin IN A.1* dari *Motor Driver* terhubung dengan *Pin D5* dari Wemos D1 Mini.
- d. *Pin IN A.2* dari *Motor Driver* terhubung dengan *Pin D6* dari Wemos D1 Mini.
- e. *Pin EN* dari *motor Driver* terhubung dengan *Pin D0* dari Wemos D1 Mini.
- f. *Pin 12V* dari *Motor Driver* terhubung dengan *plus (+)* adaptor 12V.
- g. *Pin GND* dari *Motor Driver* terhubung dengan *min (-)* daptor 12V.
- h. *Pin Plus (+)* Motor A terhubung dengan *plus (+)* Kipas DC.
- i. *Pin Min (-)* Motor A terhubung dengan *min (-)* Kipas DC.

### 3.5 Perancangan Software

Pada hal ini menjelaskan terkait Perancangan *Software* dari rancang bangun ini adalah sebagai berikut:

#### 3.5.1 Flowchart Sistem Pendeksi Kebocoran Gas LPG

Pada diagram *software* terdapat *Flowchart* dimana *Flowchart* tersebut adalah alur program untuk sistem pendeksi kebocoran gas LPG. *Flowchart* ditunjukkan oleh Gambar 3.10.



Gambar 3. 10 Flowchart Sistem Deteksi Kebocoran Gas

Pada Gambar 3.10 ditunjukkan bahwa program dimulai dengan pembacaan data oleh sensor. Kebocoran gas akan dibaca oleh sensor Figaro TGS2610 dan RTC untuk menunjukkan waktu secara *real time*. Data yang telah dibaca oleh sensor akan dilakukan *Fuzzifikasi* berdasarkan *Rule Set* yang telah diatur. Setelah itu, mencari nilai rata rata *Defuzzifikasi* dan menghasilkan *output* kecepatan PWM. Jika tidak terjadi kebocoran, maka akan dilakukan pembacaan oleh sensor kembali. Jika terjadi kebocoran gas, maka kipas dan *buzzer* akan bekerja serta *valve gas* akan menutup. Kemudian notifikasi akan dikirim melalui aplikasi Blynk.

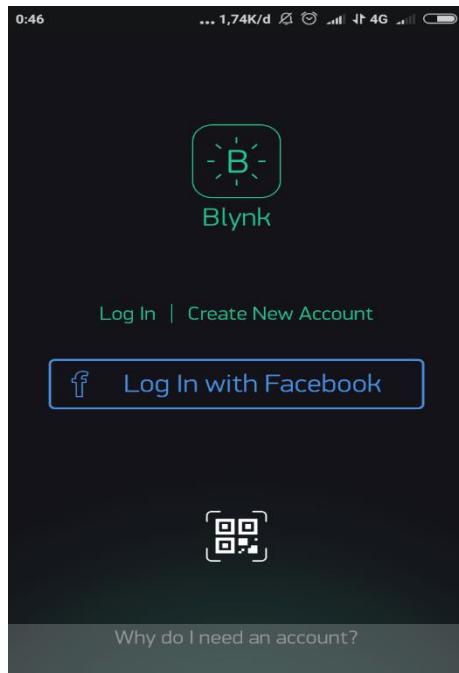
### 3.5.2 Aplikasi *Monitoring* Dan Kontrol

Aplikasi *monitoring* dan kontrol yang digunakan dalam rancangan bangun pengendali gas ini adalah Blynk. Aplikasi ini dapat diunduh pada *Play Store*. Blynk dapat membuat *interface* pada sebuah projek yang memungkinkan untuk mengirim maupun menerima data. Data dapat ditampilkan secara visual maupun grafik. Berikut merupakan komponen pada aplikasi Blynk:

1. *Controller*, berfungsi untuk mengirim data maupun perintah menuju *hardware*.
2. *Display*, berfungsi sebagai layar menampilkan data yang berasal dari *hardware*.
3. *Notification*, berfungsi sebagai pengirim pesan dan notifikasi.
4. *Interface*, berfungsi sebagai antarmuka dimana ditunjukkan berupa menu maupun *tab* pada aplikasi Blynk.

#### 1. Tampilan *Login*

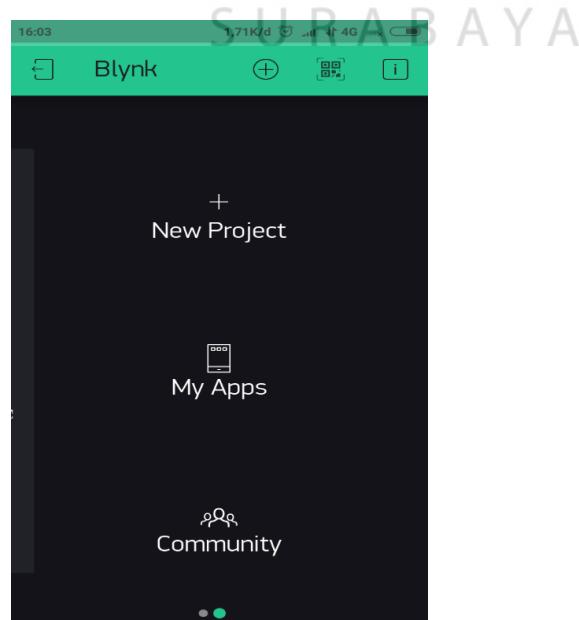
Tampilan pertama yang muncul ketika membuka aplikasi Blynk adalah tampilan yang memberi perintah untuk *login*. Pada tampilan ini, *user* yang belum mendaftarkan dirinya, bisa masuk aplikasi dengan cara membuat akun terlebih dahulu. Pembuatan akun dapat dilakukan dengan cara memilih ‘*Create New Account*’ yang terdapat pada tampilan *login*, sedangkan *user* yang sudah memiliki akun, dapat *login* dengan menggunakan *username* dan *password*. Tampilan *login* ditunjukkan oleh Gambar 3.11.



Gambar 3. 11 Tampilan Login Pada Aplikasi Blynk

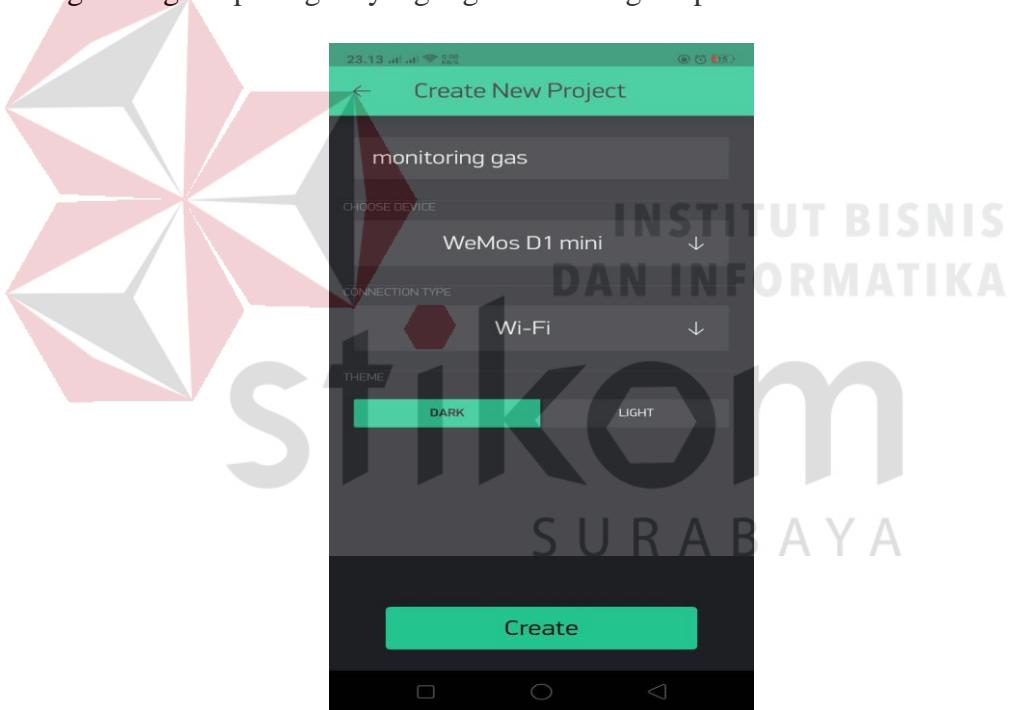
## 2. Pembuatan projek baru

User yang sudah berhasil *Login*, dapat membuat projek baru dengan cara memilih '+ New Project' pada Gambar 3.12. Setelah memilih '+ New Project', user akan ditampilkan seperti Gambar 3.13 untuk mengatur *project* yang akan dibuat.

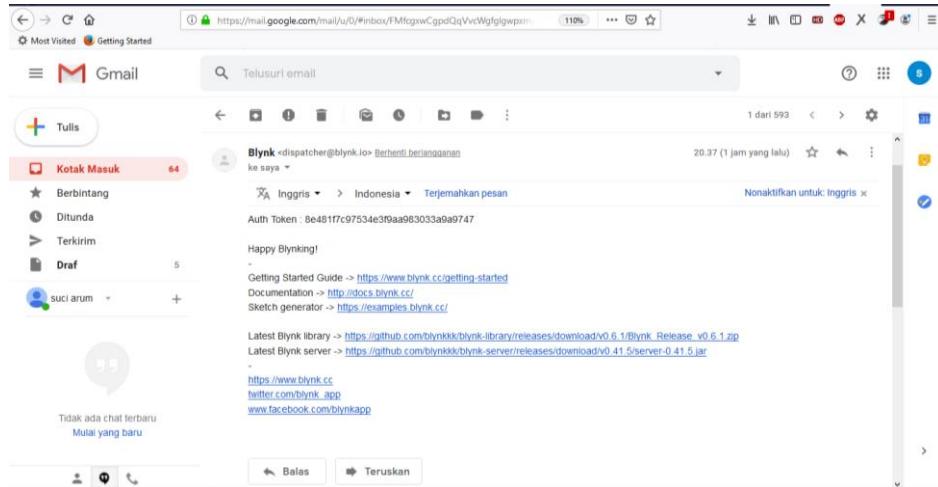


Gambar 3. 12 Tampilan New Project

Pada tampilan tersebut, terdapat 4 bagian pokok, yaitu *Project Name*, *Choose Device*, *Connector Type*, serta *Theme*. *Project Name* berfungsi untuk pemberian nama pada projek yang akan dilakukan. *Choose Device* berfungsi untuk memilih *device* yang akan digunakan. *Connector Type* adalah tempat untuk memilih koneksi yang akan digunakan, sedangkan *Theme* adalah tema tampilan yang akan digunakan. Apabila semua sudah diatur, lalu memilih ‘*Create*’ untuk membuat projek baru. Setelah itu, aplikasi akan mengirimkan *Auth Token* pada *Email user* seperti Gambar 3.14. *Auth Token* berisi kode yang berguna untuk menghubungkan perangkat yang digunakan dengan aplikasi.



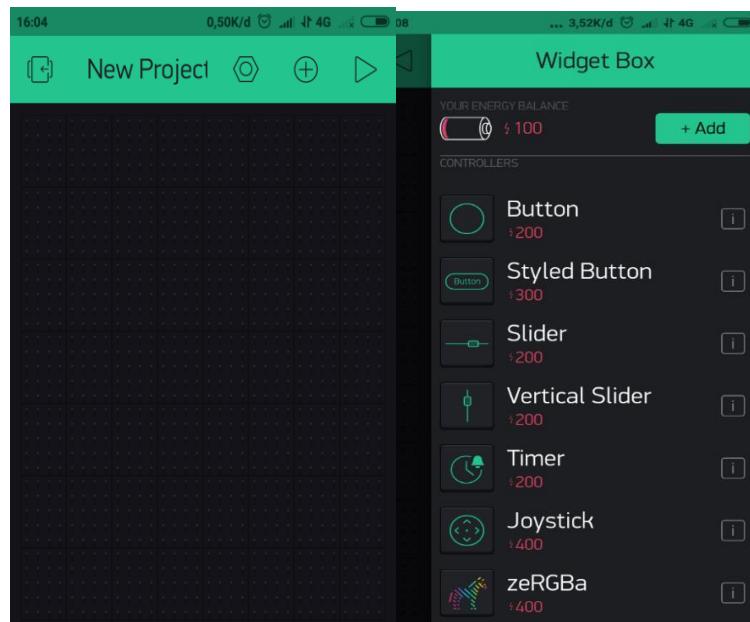
Gambar 3. 13 Tampilan *Creat New Project*



Gambar 3. 14 *Auth Token* Yang Dikirimkan Oleh Aplikasi

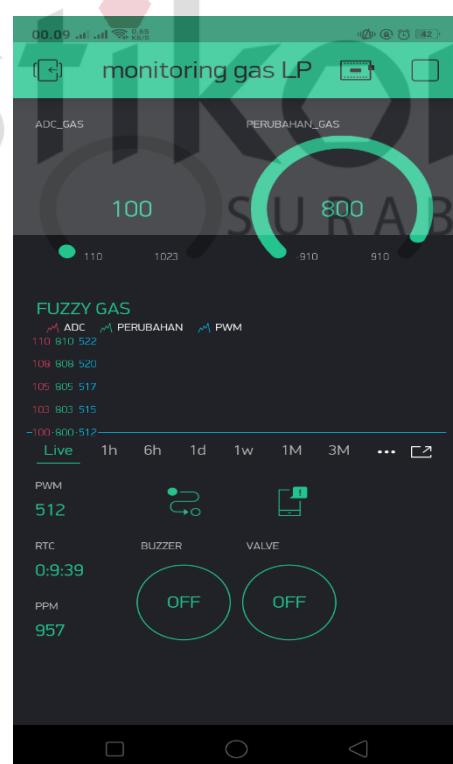
### 3. Pemilihan Widget Box

Pada gambar 3.15 ditunjukkan tampilan untuk menambah komponen *interface* yang digunakan dalam projek. Komponen tersebut tersedia berdasarkan *credit power* yang telah disediakan pada saat *login*. *Credit power* awal yang disediakan berjumlah 2000. *User* dapat memilih beberapa *widget* sesuai jumlah *credit power* yang tersisa. Ketika *credit power* tersebut telah habis digunakan seluruhnya, *user* dapat menambah dengan cara membeli pada menu *Add*.



Gambar 3. 15 Tampilan Awal Dan Widget Box

Setelah *widget* yang diperlukan dipilih, maka tampilan Blynk seperti Gambar 3.16. Pada tampilan tersebut tampak beberapa tambahan *widget* dan sudah diatur sebagaimana *port* yang akan digunakan.



Gambar 3. 16 Tampilan Aplikasi Blynk

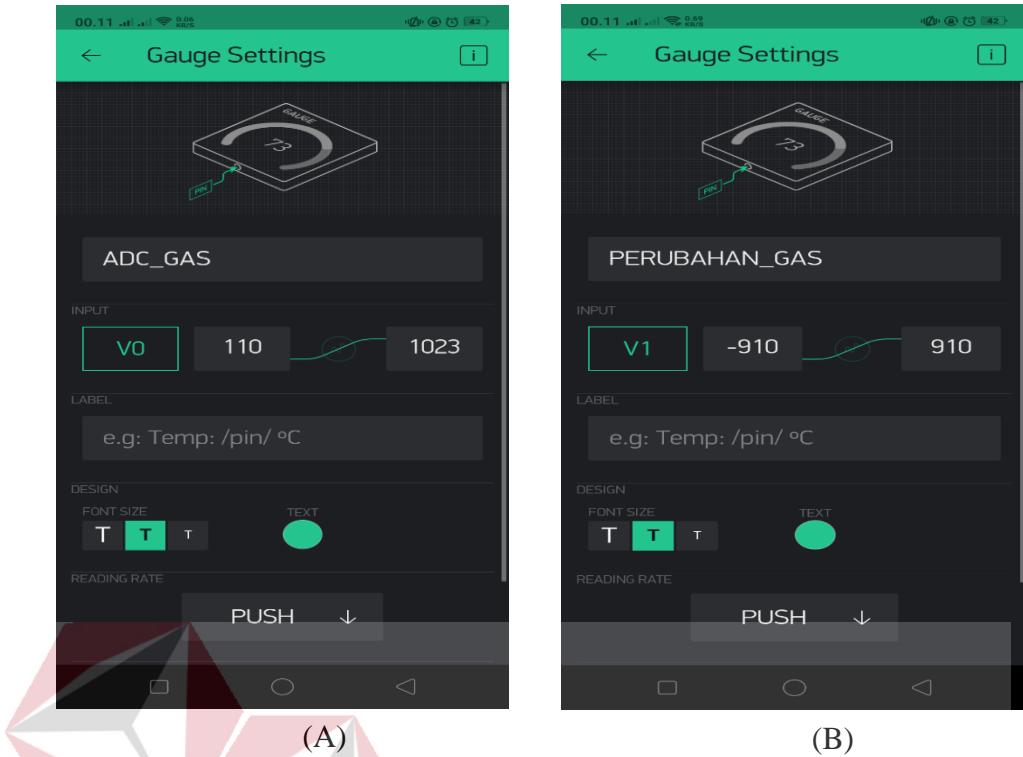
#### 4. Pengaturan *Widget Box*

*Widget* yang digunakan pada penyusunan rancang bangun ini adalah sebagai berikut:

##### 1. *Gauge*

Terdapat dua *Gauge* yang digunakan pada *monitoring* ini, yaitu *Gauge* ADC Gas dan Perubahan Gas. *Gauge* ADC Gas adalah *widget* yang berfungsi untuk menampilkan ADC gas yang dibaca dari sensor. *Input* diatur menjadi *Virtual 0* (V0). Nilai V0 dimulai dari 110 hingga 1023. PUSH digunakan agar tidak ada *delay* untuk pengiriman nilai pada V0.

Sedangkan *Gauge* Perubahan Gas berfungsi untuk menampilkan perubahan gas yang dibaca oleh sensor. *Input* diatur menjadi *Virtual 1* (V1). Nilai V1 dimulai dari -910 hingga 910. PUSH digunakan agar tidak *delay* untuk pengiriman nilai pada V1. Gambar 3.17 menunjukkan *setting* kedua *Gauge*, sedangkan program *Gauge* ditunjukkan oleh Gambar 3.18.



Gambar 3. 17 Setting Gauge (A) ADC Gas Dan (B) Perubahan Gas

```
#include <Wire.h>
#include "RTClib.h"
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

RTC_DS3231 rtc;

#define BLYNK_PRINT Serial

#define IN1 D5
#define IN2 D6
#define EN_A D0
#define buzz D4
#define valve D3

//cek email dan copy paste kan disini
char auth[] = "8e481f7c97534e3f9aa983033a9a9747";

//isikan nama wifi dan passwordnya
char ssid[] = "Kepo";
char pass[] = "dzulfikar";
```

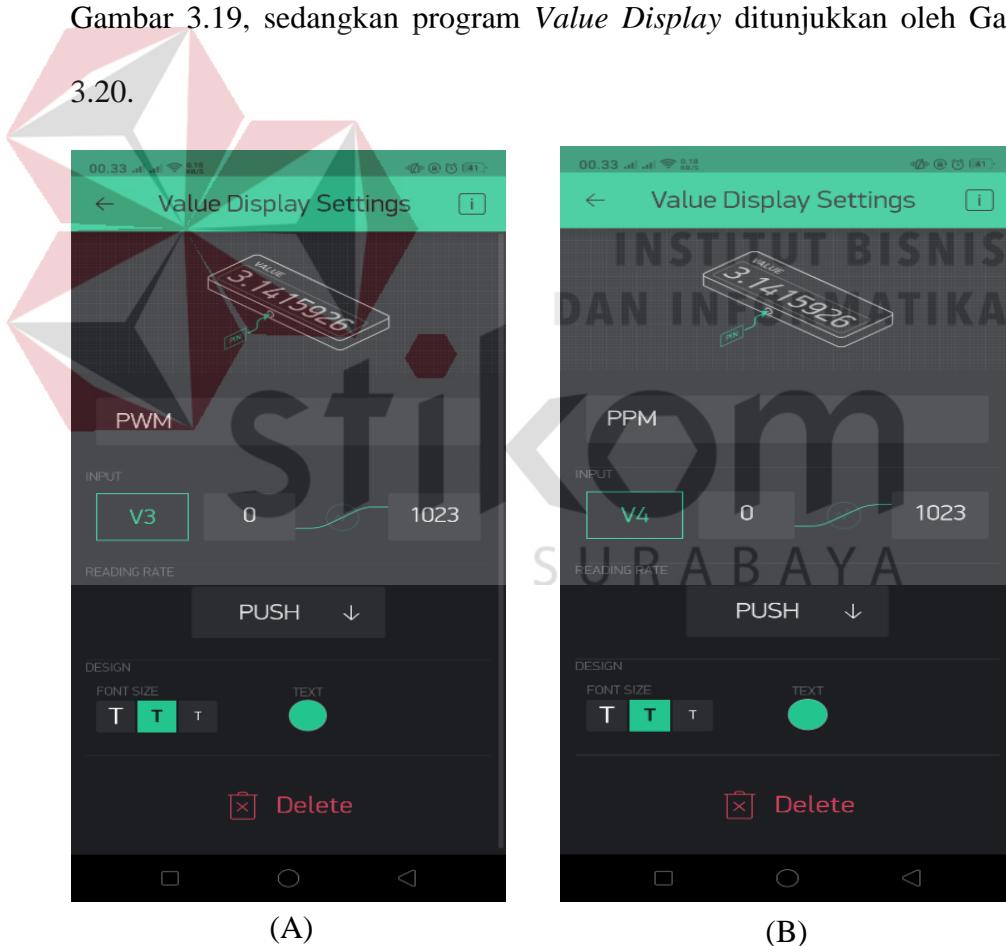
```
void setup () {
    Serial.begin(9600);
    Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 8442 );
    rtc.begin();
    lcd.begin();
    Wire.begin();
    pinMode(EN_A, OUTPUT);
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(buzz, OUTPUT);
    pinMode(valve, OUTPUT);
}

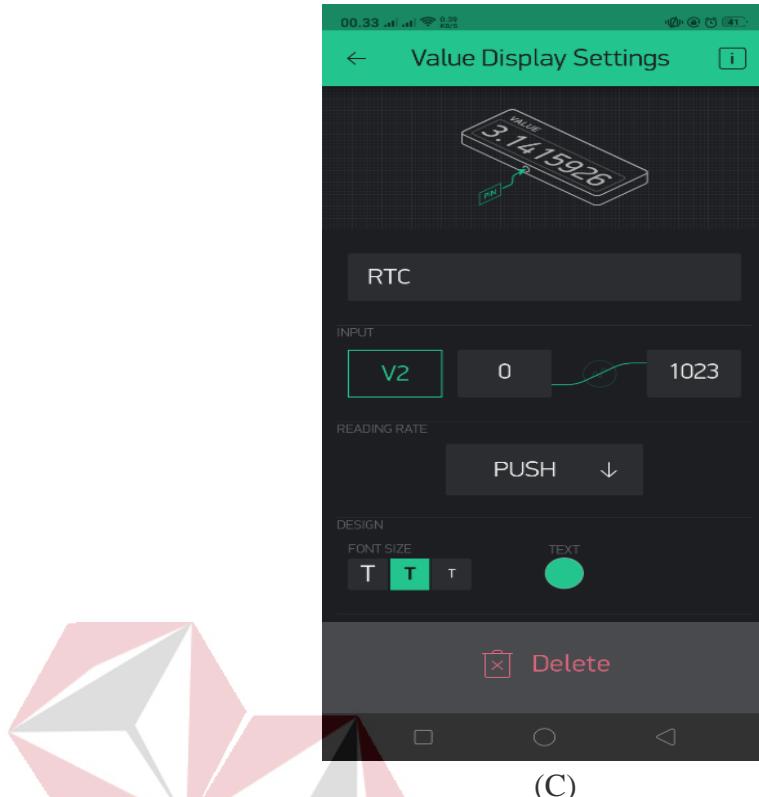
void loop () {
    // BLYNK
    Blynk.virtualWrite(V0, data_ADC);
    Blynk.virtualWrite(V1, PERUBAHAN);
    delay(1000);
}
```

Gambar 3. 18 Program Gauge

## 2. Value Display

*Value Display* merupakan *widget* yang berfungsi untuk menampilkan nilai kecepatan dari PWM, kadar gas pada PPM dan waktu dari RTC. Pada *Value Display* PWM, *input* diatur menjadi *Virtual 3* (V3). *Input Value Display* PPM adalah *Virtual 4* (V4), sedangkan pada *Value Display* RTC, *input* diatur sebagai *Vistual 5* (V5). Nilai dari masing-masing *Virtual* ini diatur mulai dari 0 hingga 1023. PUSH digunakan agar tidak ada *delay* dalam pengiriman nilai pada masing-masing *Virtual*. Tampilan *setting* *Value Display* ditampilkan oleh Gambar 3.19, sedangkan program *Value Display* ditunjukkan oleh Gambar 3.20.





Gambar 3. 19 Setting Display Value Untuk (A) PWM, (B) PPM Dan (C) RTC

```
#include <Wire.h>
#include "RTClib.h"
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

RTC_DS3231 rtc;

#define BLYNK_PRINT Serial

#define IN1 D5
#define IN2 D6
#define EN_A D0
#define buzz D4
#define valve D3

//cek email dan copy paste kan disini
char auth[] = "8e481f7c97534e3f9aa983033a9a9747";

//isikan nama wifi dan passwordnya
char ssid[] = "Kepo";
char pass[] = "dzulfikar";

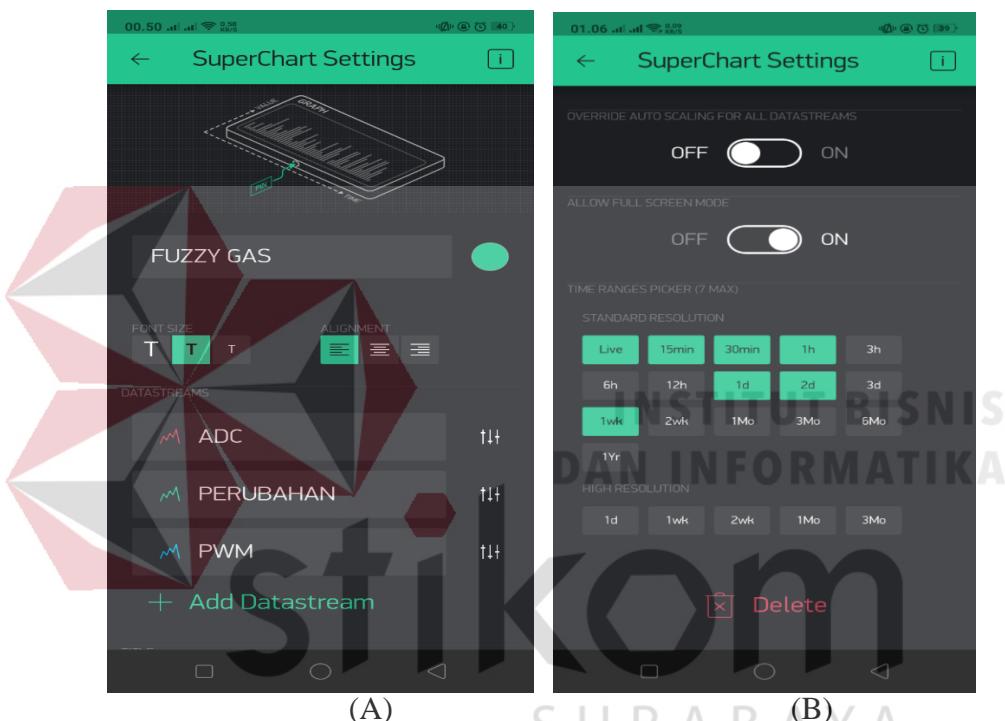
void setup () {
    Serial.begin(9600);
    Blynk.begin(auth, ssid, pass,"blynk-cloud.com", 8442 );
    rtc.begin();
    lcd.begin();
    Wire.begin();
    pinMode(EN_A, OUTPUT);
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(buzz, OUTPUT);
    pinMode(valve, OUTPUT);
}

void loop () {
    Blynk.virtualWrite(V2, waktublynk); //RTC
    Blynk.virtualWrite(V3, Hasil_defuzzy); //PWM
    Blynk.virtualWrite(V4, (int) (((10000-200)/1024.0)*data_ADC)); //PPM
    delay(1000);
}
```

Gambar 3. 20 Program Display Value

### 3. SuperChart

*SuperChart* merupakan *widget* yang digunakan untuk menampilkan *track record* sensor. Penambahan *Datastream* digunakan untuk menampilkan nilai V0, V1 dan V3 yang diambil menjadi grafik. *Setting* pada *SuperChart* ditampilkan oleh Gambar 3.21, dan Gambar 3.22, sedangkan program *SuperChart* ditampilkan oleh Gambar 3.23.



Gambar 3. 21 Tampilan (A) *SuperChart Setting* Dan (B) *Setting Waktu*

Pada Gambar 3.22 merupakan *setting datastream* yang ada di *SuperChart*. Tampilan dari pengaturan ini berbentuk grafik dan akan tersimpan selama alat berjalan untuk memudahkan pengambilan data.



Gambar 3. 22 Tampilan *Setting SuperChart* (A) ADC, (B) Perubahan ADC Dan (C) PWM

```

#include <Wire.h>
#include "RTClib.h"
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

RTC_DS3231 rtc;

#define BLYNK_PRINT Serial

#define IN1 D5
#define IN2 D6
#define EN_A D0
#define buzz D4
#define valve D3

//cek email dan copy paste kan disini
char auth[] = "8e481f7c97534e3f9aa983033a9a9747";

//isikan nama wifi dan passwordnya
char ssid[] = "Kepo";
char pass[] = "dzulfikar";

void setup () {
    Serial.begin(9600);
    Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 8442 );
    rtc.begin();
    lcd.begin();
    Wire.begin();
    pinMode(EN_A, OUTPUT);
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(buzz, OUTPUT);
    pinMode(valve, OUTPUT);
}

void loop () {
    Blynk.virtualWrite(V2, waktublynk); //RTC
    Blynk.virtualWrite(V3, Hasil_defuzzy); //PWM
    Blynk.virtualWrite(V4, (int) (((10000-200)/1024.0)*data_ADC)); //PPM
    delay(1000);
}

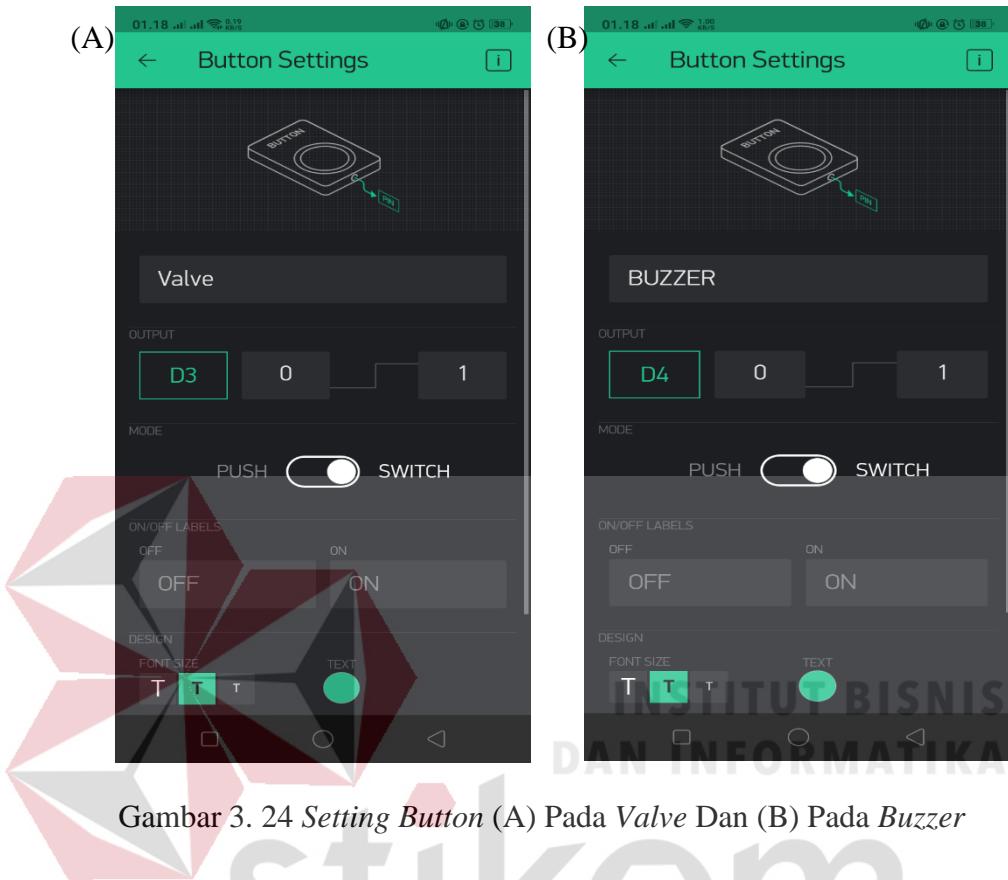
```

Gambar 3. 23 Program *SuperChart*

#### 4. Button

*Button* merupakan *widget* yang berfungsi untuk mengontrol *valve* dan *buzzer*. *Output* pada Blynk diatur pada D3 dan D4 karena *Pin* tersebut terhubung dengan *Pin* D3 dan D4 pada Wemos D1 Mini. Nilai 0 adalah kondisi OFF pada *valve* dan *buzzer*, sedangkan nilai 1 adalah kondisi saat keduanya

ON. Setting pada *Button* ditampilkan oleh Gambar 3.24, sedangkan program *Button* ditampilkan oleh Gambar 3.25.



Gambar 3. 24 Setting Button (A) Pada Valve Dan (B) Pada Buzzer

```
#include <Wire.h>
#include "RTClib.h"
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

RTC_DS3231 rtc;

#define BLYNK_PRINT Serial

#define IN1 D5
#define IN2 D6
#define EN_A D0
#define buzz D4
#define valve D3

//cek email dan copy paste kan disini
char auth[] = "8e481f7c97534e3f9aa983033a9a9747";

//isikan nama wifi dan passwordnya
char ssid[] = "Kepo";
char pass[] = "dzulfikar";
```

Gambar 3. 25 Program Button

## 5. Eventor

*Eventor* merupakan *wigdet* yang digunakan untuk menampilkan kalimat peringatan apabila terjadi kebocoran gas. Kebocoran gas ditandai dengan V0 yang lebih besar atau sama dengan 110. Pada *monitoring* ini, kalimat yang diberikan berupa “WARNING !! GAS ELPIJI BOCOR !!”. Kalimat tersebut diatur dengan memilih “+Add New Event” pada tampilan *setting. Setting* pada *Eventor* ditampilkan oleh Gambar 3.26.



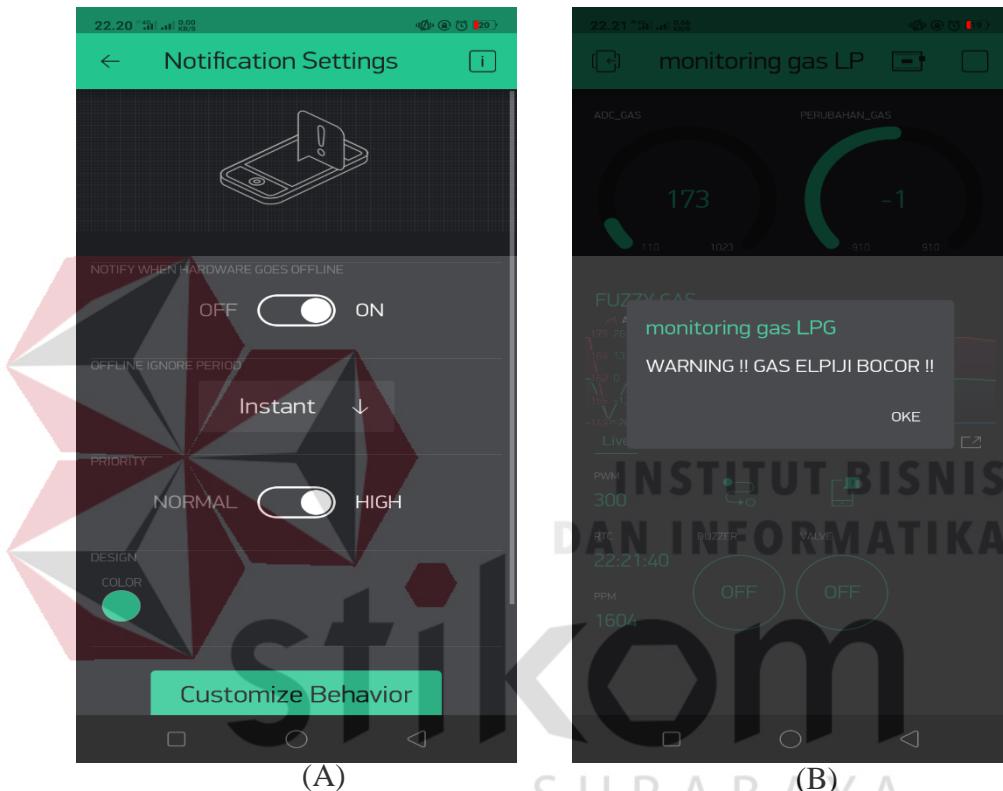
Gambar 3. 26 *Setting Eventor* (A) Tampilan Utama *Setting Eventor*; (B) *Setting New Event*

## 6. Notification

*Notification* merupakan *widget* yang digunakan untuk memberikan notifikasi. Jika *Notify When Hardware Goes Offline* diatur menjadi *On*, maka aplikasi akan memberikan notifikasi bahwa Wemos tidak terhubung ke

internet, sedangkan jika *Offline Ignore Period* diatur menjadi *Instant*, maka ketika Wemos tidak terhubung ke internet aplikasi akan memberikan notifikasi saat itu juga. *Setting* notifikasi ini ditampilkan seperti Gambar 3.27(a). Selain itu, *widget* ini akan memberikan notifikasi apabila terjadi kebocoran gas.

*Setting* pada *Notification* kebocoran gas ditampilkan pada Gambar 3.27(b).



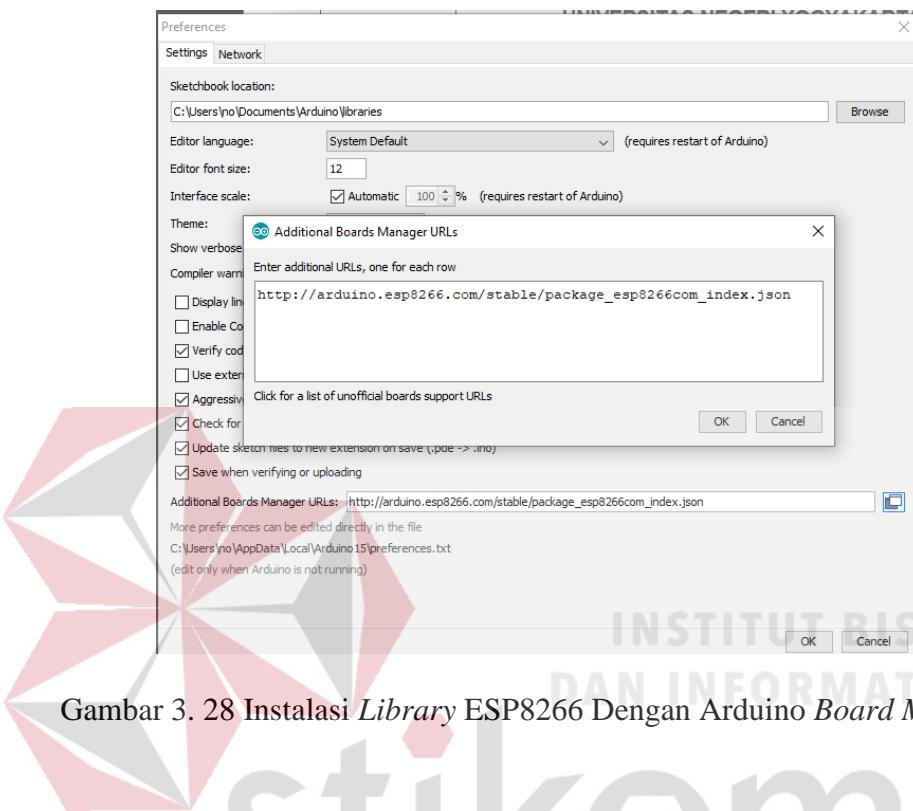
Gambar 3. 27 *Setting Notification* (A) Untuk Pemberian Informasi Koneksi Internet Pada Wemos Dan (B) Untuk Pemberian Informasi Apabila Terjadi Kebocoran Gas

### 3.5.3 Konfigurasi Arduino IDE

- Instalasi *Library* ESP8266 dengan Arduino *Board Manager*

Proses instalasi ESP8266 ditunjukkan seperti Gambar 3.28. Instalasi dilakukan dengan memilih menu *preferences* (**File > Preferences**) pada Arduino IDE. Selanjutnya masukan URL pada “Additional Board Manager URL’s” yang terletak pada bagian bawah *window*. Kemudian untuk menginstal ESP8266, masuk

ke menu **Board Manager (Tools > Boards > Boards Manager)** dan mengetik ESP8266 dikolom pencarian. Setelah itu mengklik *Install* seperti yang ditunjukkan oleh Gambar 3.29.



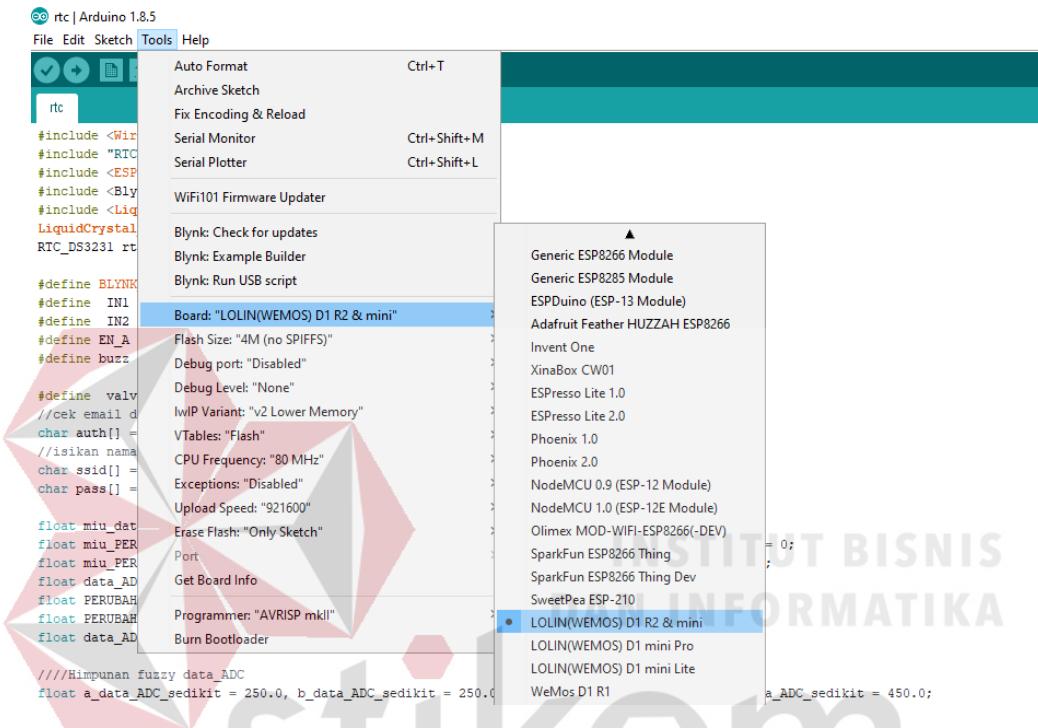
Gambar 3. 28 Instalasi Library ESP8266 Dengan Arduino Board Manager



Gambar 3. 29 Install Wemos Board Manager

## b. Memilih *Board* Wemos D1 Mini

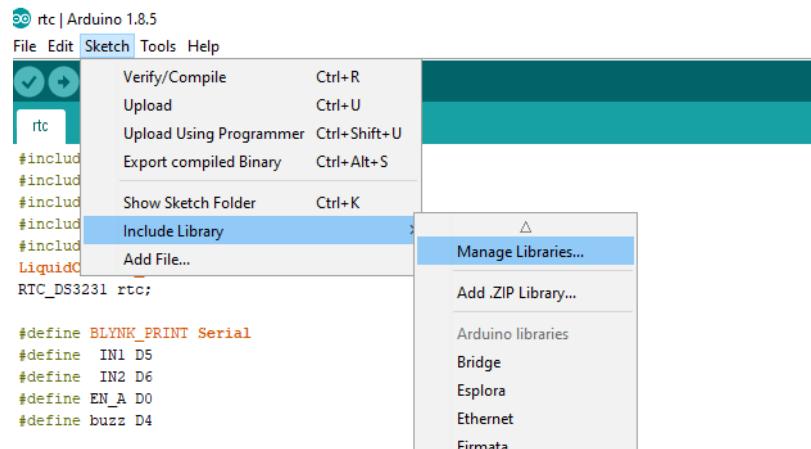
Setelah menambahkan *library board manager*, selanjutnya memilih *board* Wemos D1 Mini. Pemilihan *board* Wemos D1 Mini ditunjukkan oleh Gambar 3.30.



Gambar 3. 30 Pemilihan *Board* Wemos D1 Mini

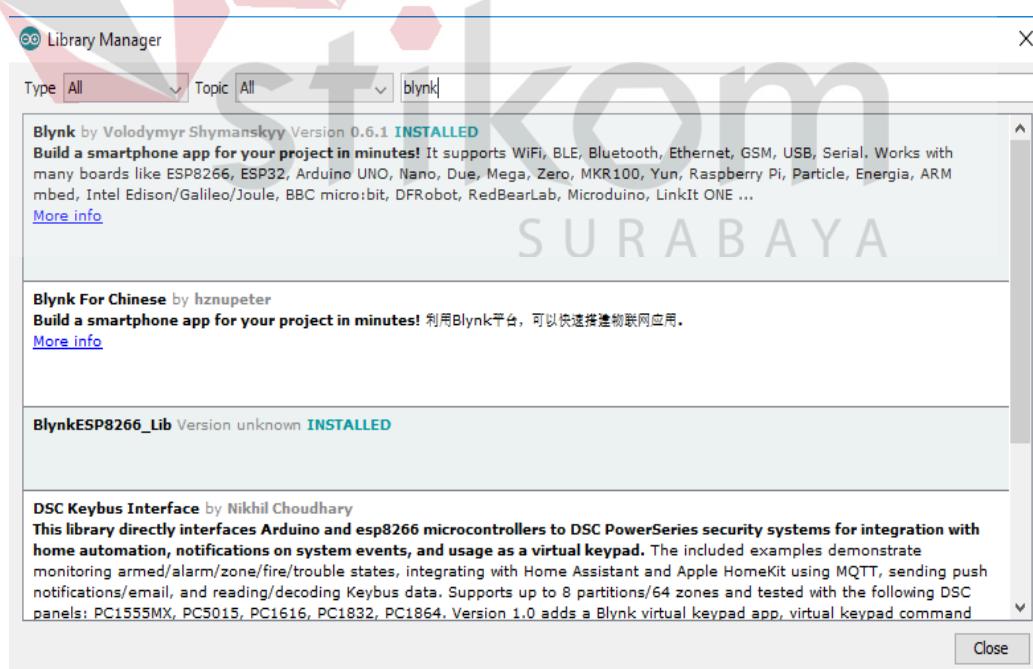
## c. Install *Library* Blynk

*Install Library* Blynk dilakukan dengan cara mengklik menu **Sketch → Include Library → Manage Libraries**. Kemudian mengetik Blynk pada kolom pencarian seperti yang ditunjukkan oleh Gambar 3.31.

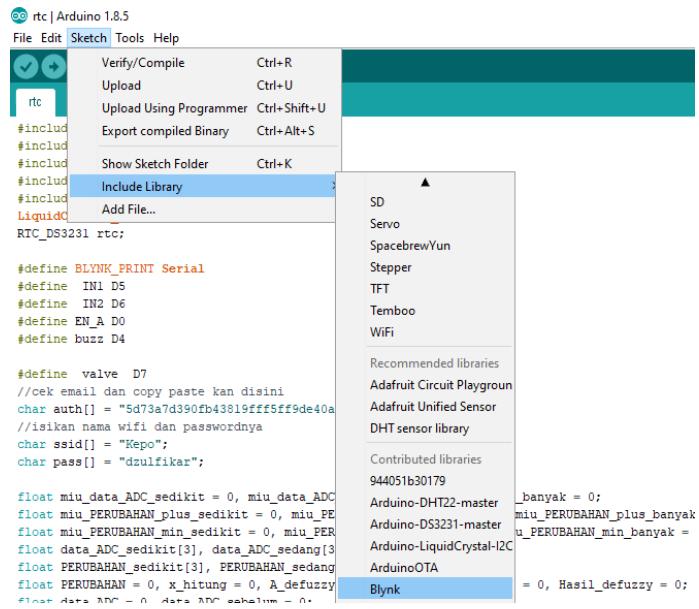


Gambar 3. 31 *Install Library Blynk*

Apabila proses instalasi telah selesai, maka Blynk sudah muncul pada *Include Library*. Cara menemukan menu *Library Blynk* yaitu dengan mengklik **Sketch** → **Include Library** → **Blynk** seperti yang ditampilkan oleh Gambar 3.32 dan Gambar 3.33.



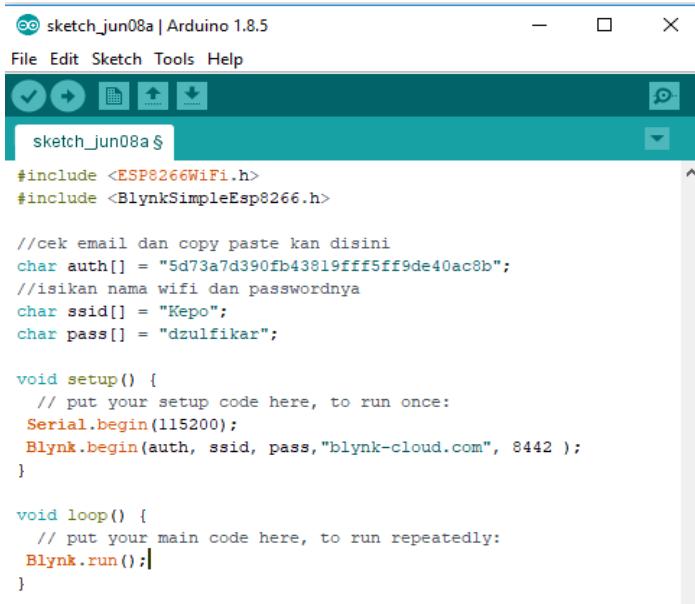
Gambar 3. 32 *Manager Library*



Gambar 3. 33 *Include Library* Blynk

### 3.5.4 Program Koneksi Wemos D1 Mini Dengan Wi-Fi

Kode yang digunakan sebagai koneksi Wemos D1 Mini dengan *Wi-Fi* ditunjukkan oleh Gambar 3.34. Pada kode tersebut, terdapat *library #include <ESP8266WiFi.h>* dan *#include <BlynkSimpleEsp8266.h>* yang berfungsi untuk menghubungkan perangkat dengan aplikasi Blynk dan juga sebagai penghubung *Wi-Fi* pada modul.



```

sketch_jun08a | Arduino 1.8.5
File Edit Sketch Tools Help
sketch_jun08a §
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

//cek email dan copy paste kan disini
char auth[] = "5d73a7d390fb43819fff5ff9de40ac8b";
//isikan nama wifi dan passwordnya
char ssid[] = "Kepo";
char pass[] = "dzulfikar";

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 8442 );
}

void loop() {
    // put your main code here, to run repeatedly:
    Blynk.run();
}

```

Gambar 3. 34 Kode Koneksi Wemos D1 Mini Dengan Wi-Fi

### 3.6 Desain Sistem Fuzzy

Pada hal ini menjelaskan terkait desain sistem Fuzzy ini adalah sebagai berikut:

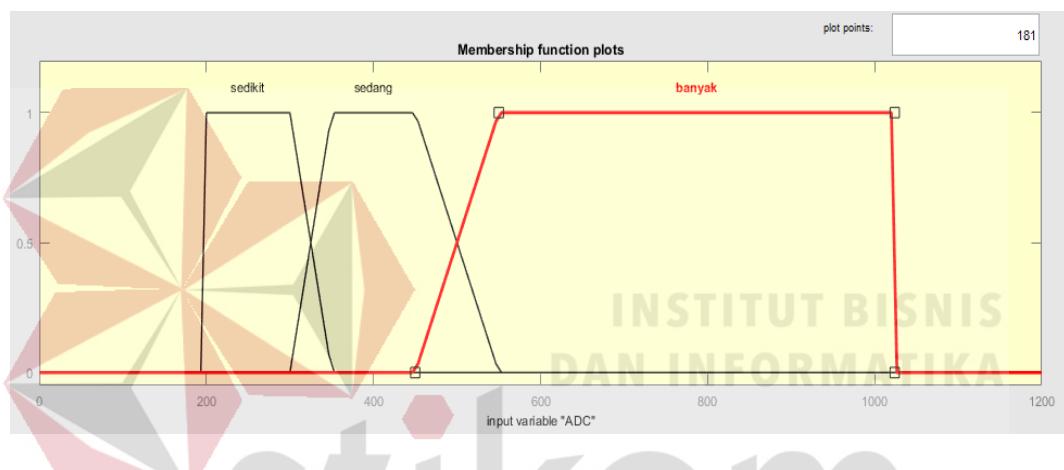
#### 3.6.1 Input Pada Fuzzy Sugeno

Dengan menggunakan metode *Sugeno Fuzzy* maka pada masing-masing sensor harus diset terlebih dahulu data-data yang digunakan sebagai parameter *input* pada *Fuzzifikasi*. Berikut adalah himpunan *Fuzzy* dari masing-masing *input* beserta dengan ilustrasi dari *Fuzzy* set-nya serta *rules* yang digunakan:

a. Himpunan *Fuzzy* Gas :

Tabel 3. 1 Keanggotaan Gas

Gas	MF	PARAMETER			
		A	B	C	D
Sedikit	Trapesium	200	200	300	350
Sedang	Trapesium	300	350	450	550
Banyak	Trapesium	450	550	1024	1024

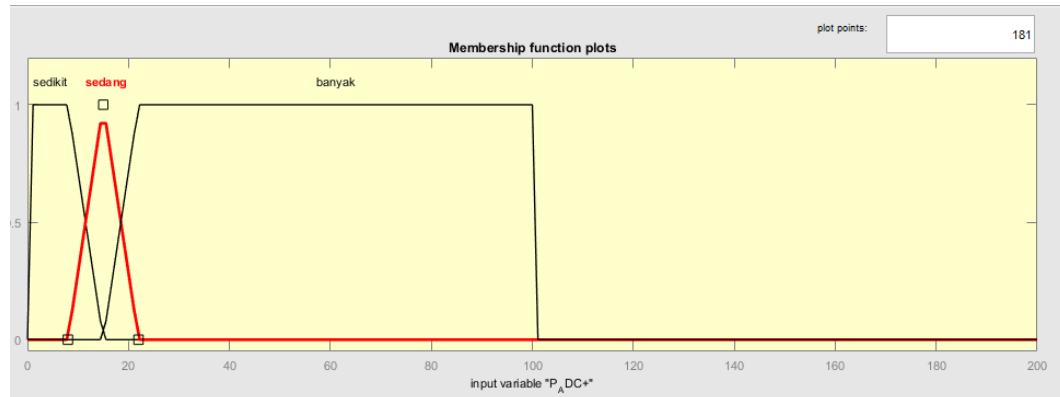


Gambar 3. 35 Fungsi Keanggotaan Gas

b. Himpunan *Fuzzy* Perubahan Gas (+):

Tabel 3. 2 Keanggotaan Perubahan Gas(+)

Gas	MF	PARAMETER			
		A	B	C	D
Sedikit	Trapesium	1	1	8	15
Sedang	Segitiga	8	15	22	-
Banyak	Trapesium	15	22	100	100

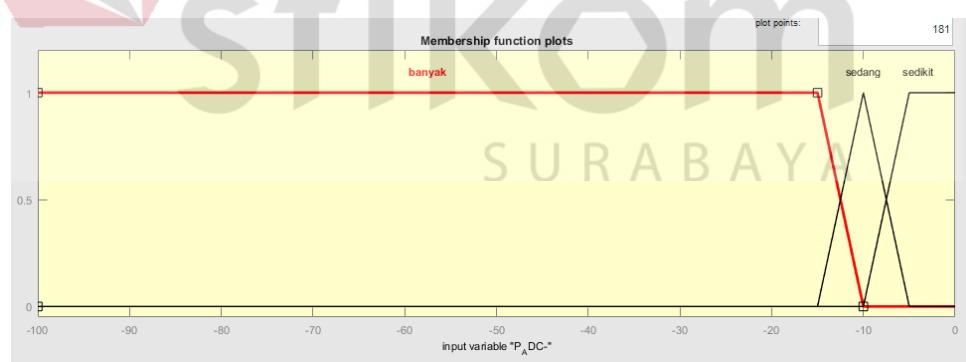


Gambar 3. 36 Fungsi Keanggotaan Perubahan Gas (+)

c. Himpunan *Fuzzy* Perubahan Gas (-) :

Tabel 3. 3 Keanggotaan Perubahan Gas (-)

Gas	MF	PARAMETER			
		A	B	C	D
Sedikit	Trapesium	0	0	-5	-10
Sedang	Segitiga	-5	-10	-15	-
Banyak	Trapesium	-10	-15	-100	-100



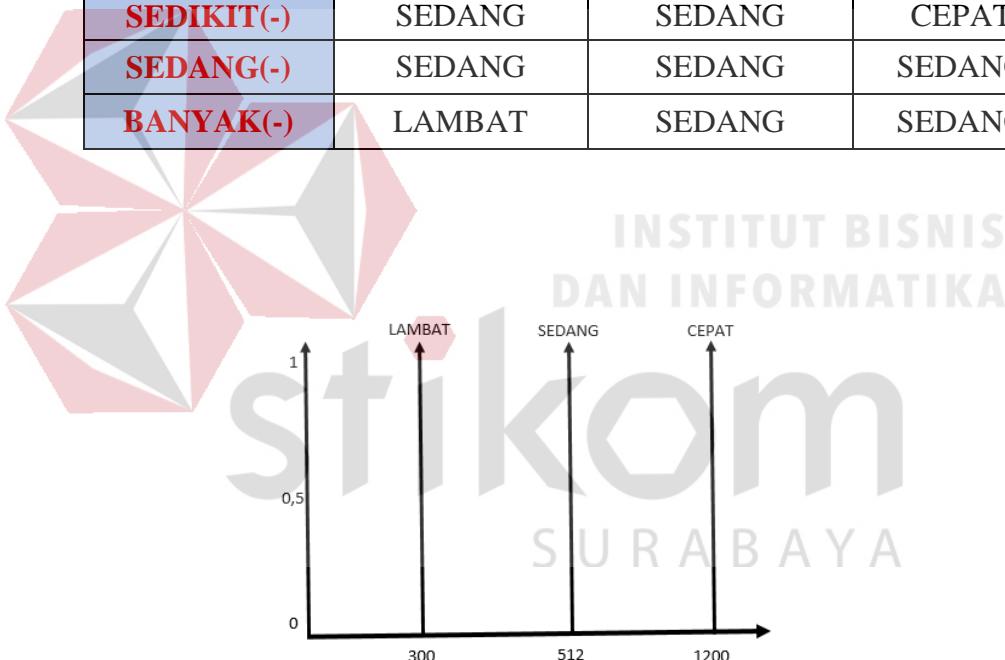
Gambar 3. 37 Fungsi Keanggotaan Perubahan Gas (-)

### 3.6.2 Output Pada Fuzzy Sugeno

*Output* pada sistem *Fuzzy* ini berupa kecepatan Kipas DC. Kecepatan Kipas DC diatur sesuai *rule base* pada Tabel 3.4, sedangkan fungsi keanggotaan Kipas DC digambarkan pada Gambar 3.38.

Tabel 3. 4 *Rule Base Fuzzy*

PERUBAHAN GAS / GAS	SEDIKIT	SEDANG	BANYAK
<b>SEDIKIT(+)</b>	LAMBAT	LAMBAT	CEPAT
<b>SEDANG(+)</b>	SEDANG	SEDANG	CEPAT
<b>BANYAK(+)</b>	SEDANG	CEPAT	CEPAT
<b>SEDIKIT(-)</b>	SEDANG	SEDANG	CEPAT
<b>SEDANG(-)</b>	SEDANG	SEDANG	SEDANG
<b>BANYAK(-)</b>	LAMBAT	SEDANG	SEDANG



Gambar 3. 38 Fungsi Keanggotaan Kipas

Pada Tabel 3.4 menampilkan *rule base* yang berisi aturan output dari sistem *Fuzzy*. *Rule base* ditentukan oleh ahli atau perancang sistem *Fuzzy* tersebut. *Rule base* dipengaruhi oleh himpunan gas dan himpunan perubahan gas. *Rule base* yang dihasilkan menentukan kecepatan kipas seperti yang ditampilkan oleh Gambar

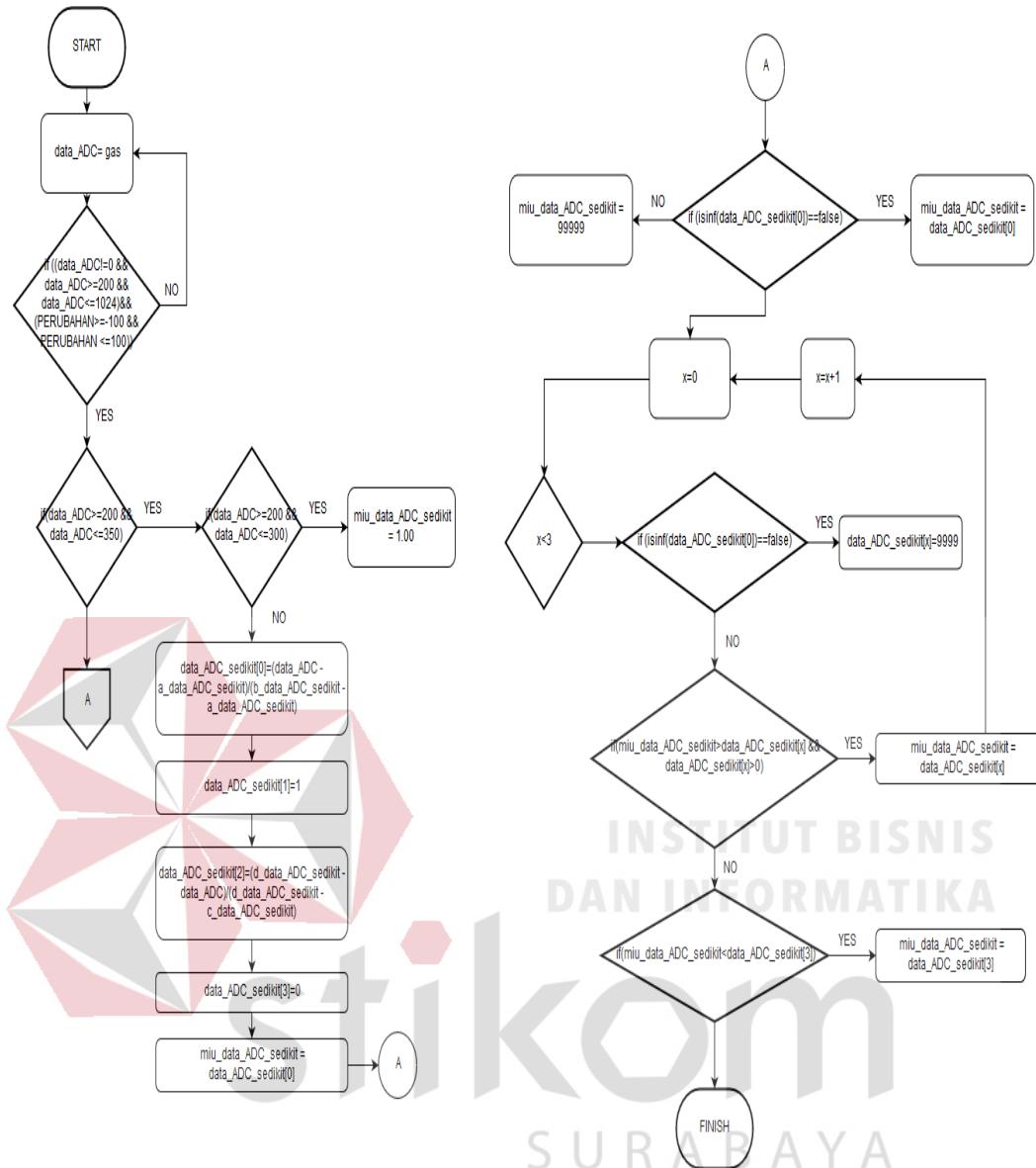
3.38. Saat *rule base* yang dihasilkan adalah “Lambat”, maka rentang kecepatan kipas adalah 0-300 PWM. Saat *rule base* yang dihasilkan “Sedang”, maka rentang kecepatan kipas sebesar 300-512 PWM, sedangkan saat *rule base* yang dihasilkan “Cepat”, maka kecepatan kipas sebesar 512-1200 PWM.

### **3.6.3 Flowchart Sistem Fuzzy Logic**

#### a. *Fuzzifikasi* Gas

*Fuzzifikasi* Gas dilakukan dengan mengolah data yang didapat dari pembacaan sensor TGS2610. *Fuzzifikasi* dilakukan sesuai *rule* yang sudah ditentukan. Himpunan Gas dikelompokan dalam 3 bagian, yaitu sedikit, sedang, dan banyak.





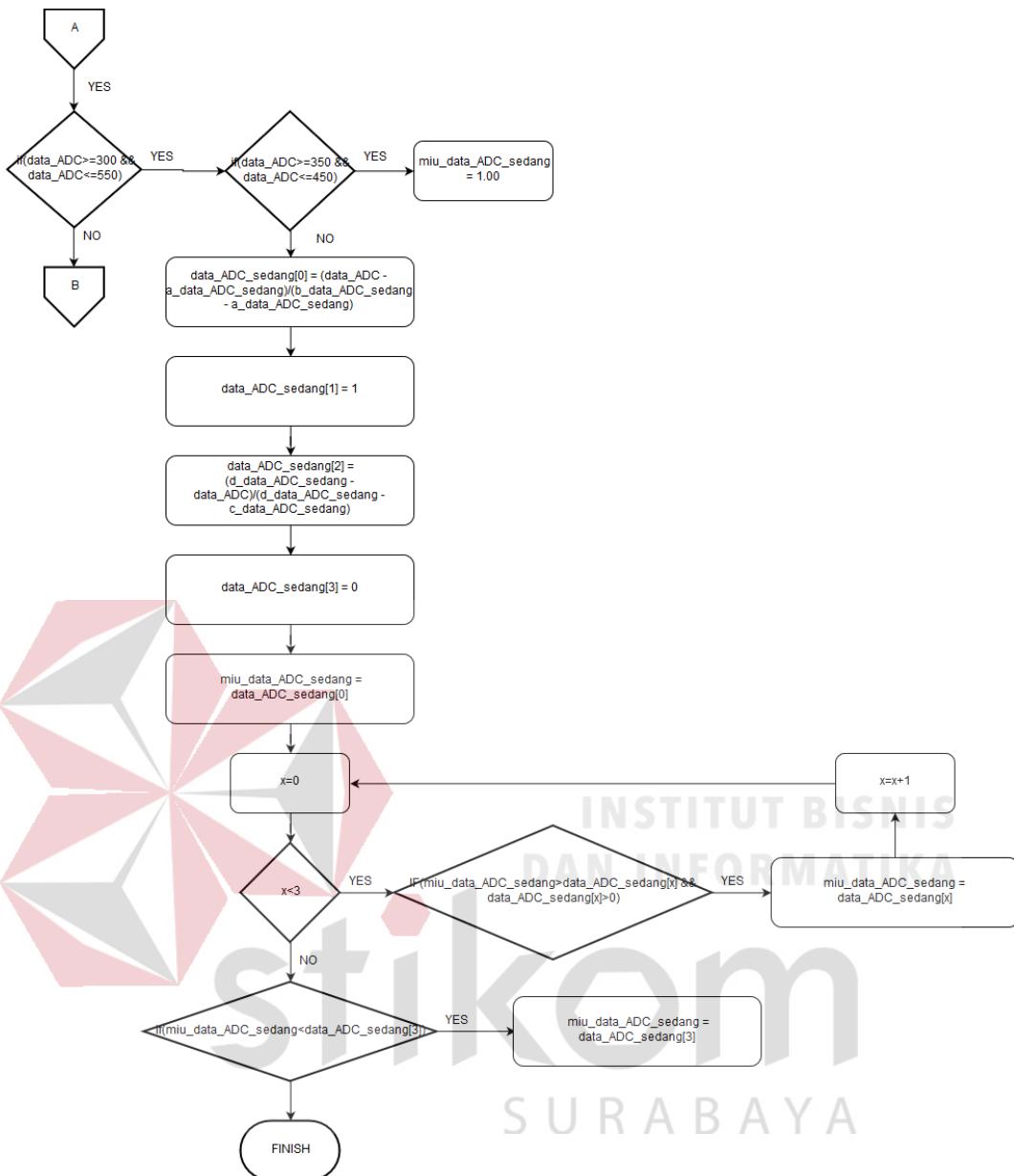
Gambar 3. 39 Flowchart Fuzzifikasi Gas Sedikit

Gambar 3.39 menunjukkan *Flowchart Fuzzifikasi* gas sedikit. Kondisi pertama jika ADC sensor lebih dari sama dengan 200 dan ADC sensor kurang dari sama dengan 350. Kondisi ini menunjukkan rentang nilai *Fuzzifikasi* gas dengan konsentrasi sedikit. Jika kondisi pertama terpenuhi, maka akan masuk pada kondisi kedua. Pada kondisi kedua, jika ADC sensor lebih besar sama dengan 200 dan lebih kecil sama dengan 300 maka gas akan dikategorikan rendah yaitu 1. Jika kondisi

kedua tidak terpenuhi, maka akan dihitung untuk rata-rata mencari derajat keanggotaannya dengan rumus trapesium.

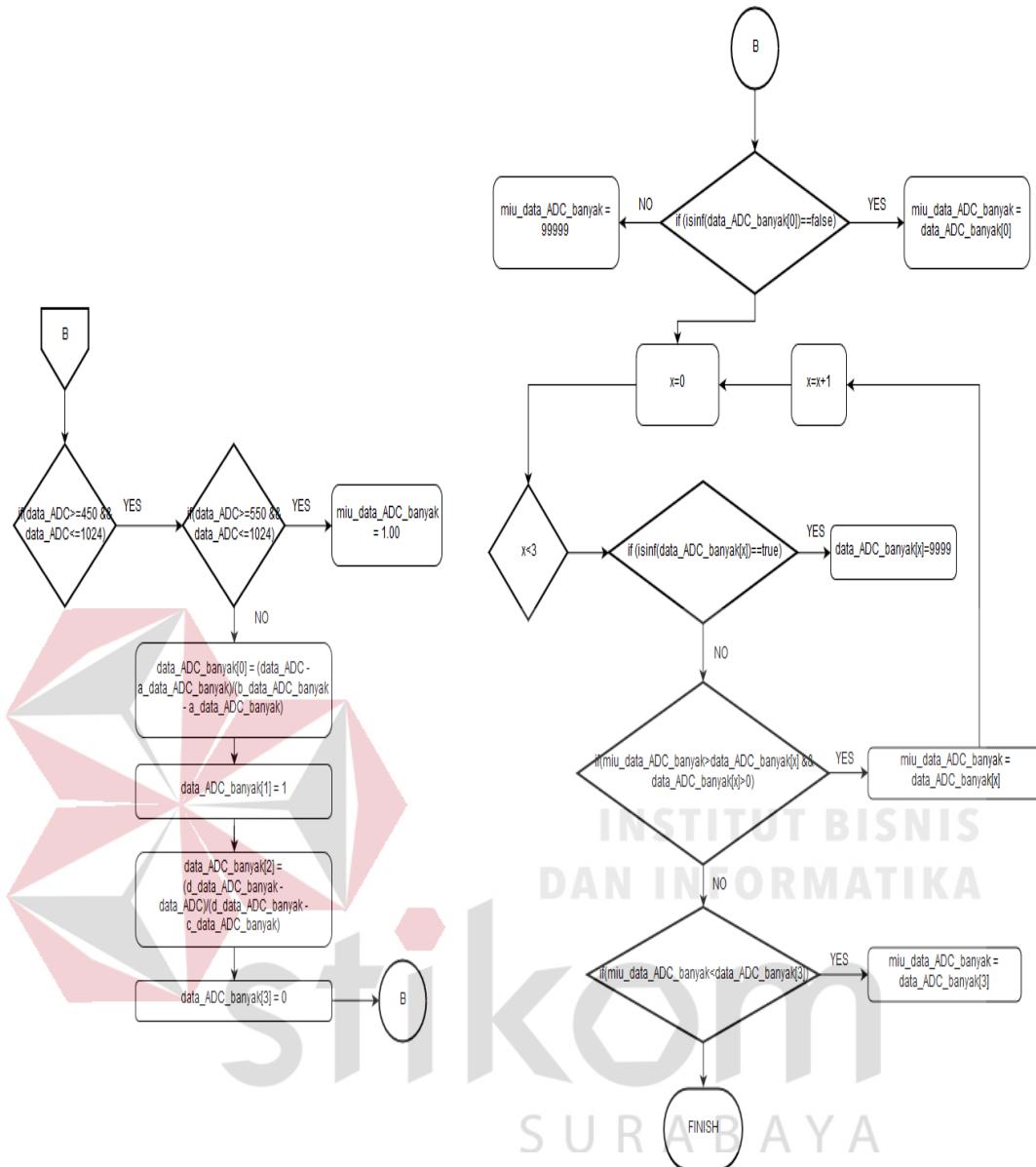
Gambar 3.40 menunjukkan *Flowchart Fuzzifikasi* gas sedang. Kondisi pertama jika ADC sensor lebih dari sama dengan 300 dan ADC sensor kurang dari sama dengan 550. Kondisi ini menunjukkan rentang nilai *Fuzzifikasi* gas dengan konsentrasi sedang. Jika kondisi pertama terpenuhi, maka akan masuk pada kondisi kedua. Pada kondisi kedua, jika ADC sensor lebih besar sama dengan 350 dan lebih kecil sama dengan 450 maka gas akan dikategorikan sedang yaitu 1. Jika kondisi kedua tidak terpenuhi, maka akan dihitung untuk mencari rata-rata derajat keanggotaannya dengan rumus trapesium.





Gambar 3. 40 Flowchart Fuzzifikasi Gas Sedang

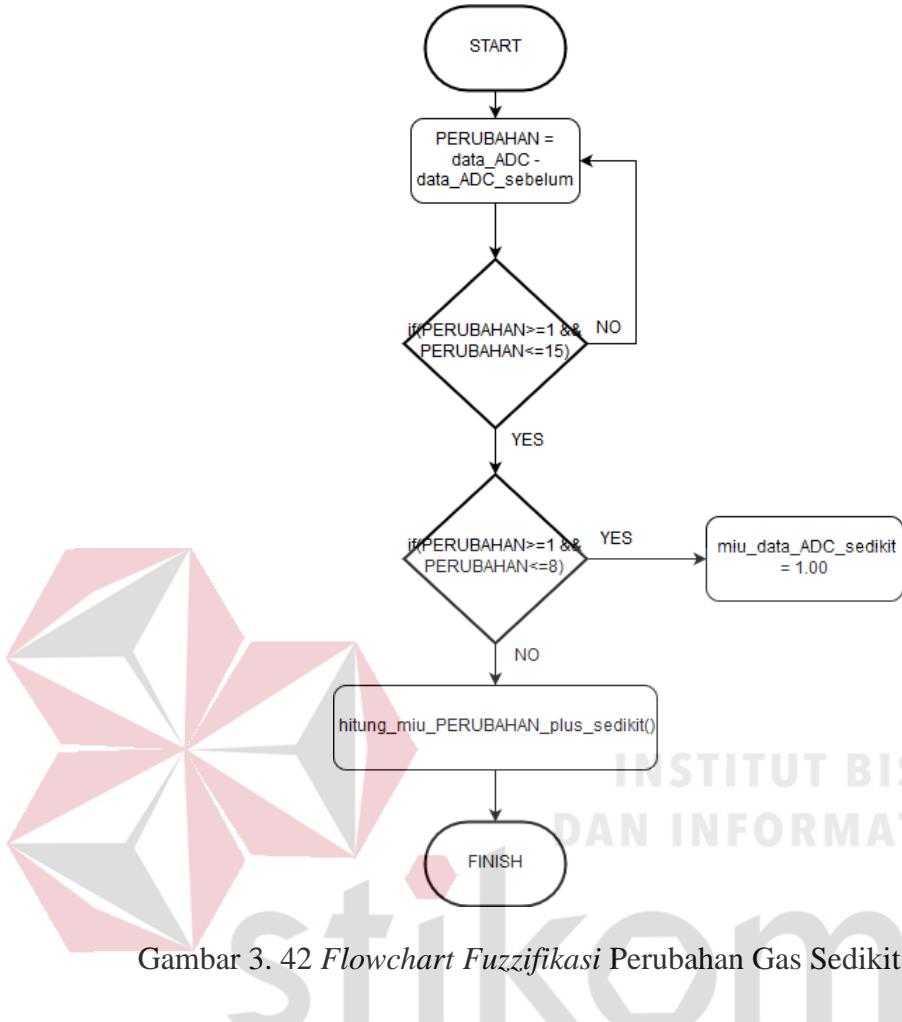
Gambar 3.41 menunjukkan Flowchart Fuzzifikasi gas banyak. Kondisi pertama jika ADC sensor lebih dari sama dengan 450 dan ADC sensor kurang dari sama dengan 1024.



Gambar 3. 41 Flowchart Fuzzifikasi Gas Banyak

Kondisi ini menunjukkan rentang nilai *Fuzzifikasi* gas dengan konsentrasi banyak. Jika kondisi pertama terpenuhi, maka akan masuk pada kondisi kedua. Pada kondisi kedua, jika ADC sensor lebih besar sama dengan 550 dan lebih kecil sama dengan 1024 maka gas akan dikategorikan banyak yaitu 1. Jika kondisi kedua tidak terpenuhi, maka akan dihitung untuk mencari rata-rata derajat keanggotaannya dengan rumus trapesium.

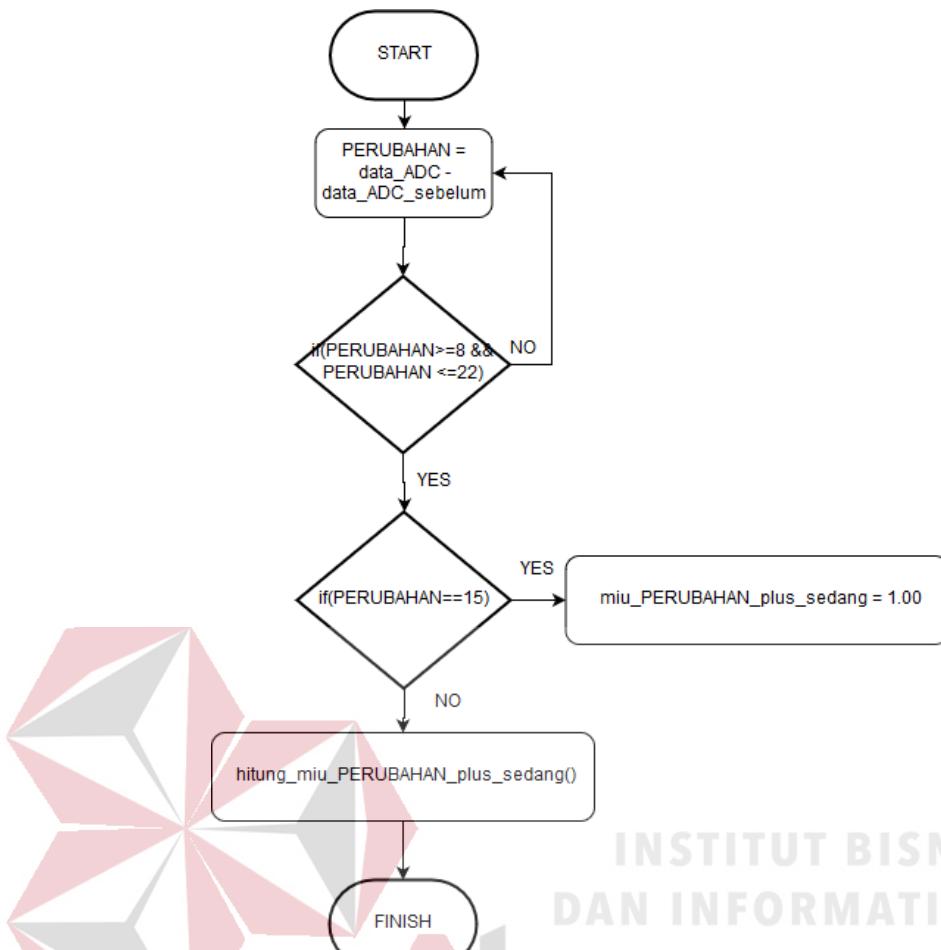
b. *Fuzzifikasi Perubahan Gas( + )*



Gambar 3.42 Flowchart *Fuzzifikasi* Perubahan Gas Sedikit (+)

Gambar 3.42 menunjukkan *Flowchart Fuzzifikasi* perubahan gas sedikit (+).

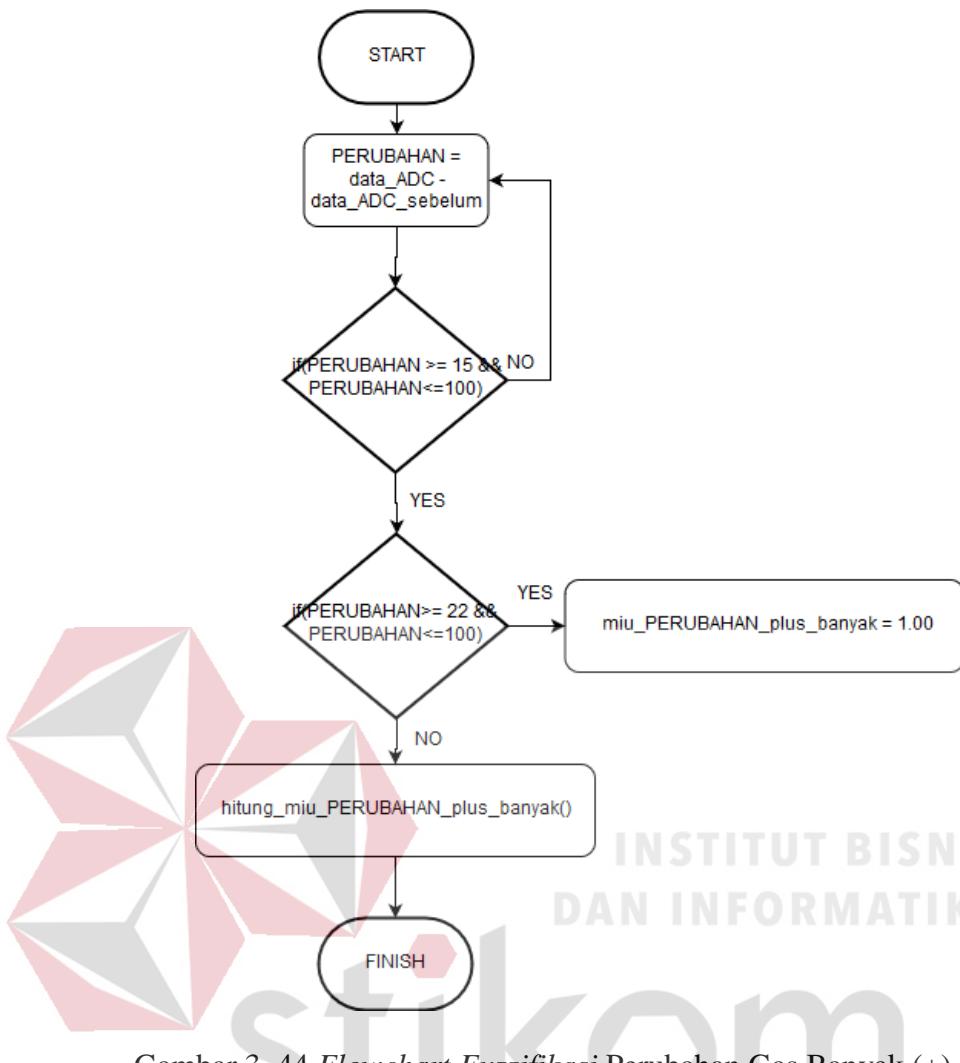
Kondisi pertama jika ADC sensor lebih dari sama dengan 1 dan ADC sensor kurang dari sama dengan 15. Kondisi ini menunjukkan rentang nilai *Fuzzifikasi* perubahan gas dengan konsentrasi sedikit (+). Jika kondisi pertama terpenuhi, maka akan masuk pada kondisi kedua. Pada kondisi kedua, jika ADC sensor lebih besar sama dengan 1 dan lebih kecil sama dengan 8 maka perubahan gas akan dikategorikan sedikit(+) yaitu 1. Jika kondisi kedua tidak terpenuhi, maka akan dihitung untuk mencari rata-rata derajat keanggotaannya dengan rumus trapesium.



Gambar 3.43 Flowchart Fuzzifikasi Perubahan Gas Sedang (+)

Gambar 3.43 menunjukkan Flowchart Fuzzifikasi perubahan gas sedang (+).

Kondisi pertama jika perubahan ADC sensor lebih dari sama dengan 8 dan perubahan ADC sensor kurang dari sama dengan 22. Kondisi ini menunjukkan rentang nilai *Fuzzifikasi* perubahan gas dengan konsentrasi sedang (+). Jika kondisi pertama terpenuhi, maka akan masuk pada kondisi kedua yaitu jika Perubahan ADC sama dengan 15 akan dikategorikan 1 dan jika kondisi kedua tidak terpenuhi maka dihitung untuk mencari rata-rata derajat keanggotaannya dengan rumus segitiga.



Gambar 3. 44 *Flowchart Fuzzifikasi* Perubahan Gas Banyak (+)

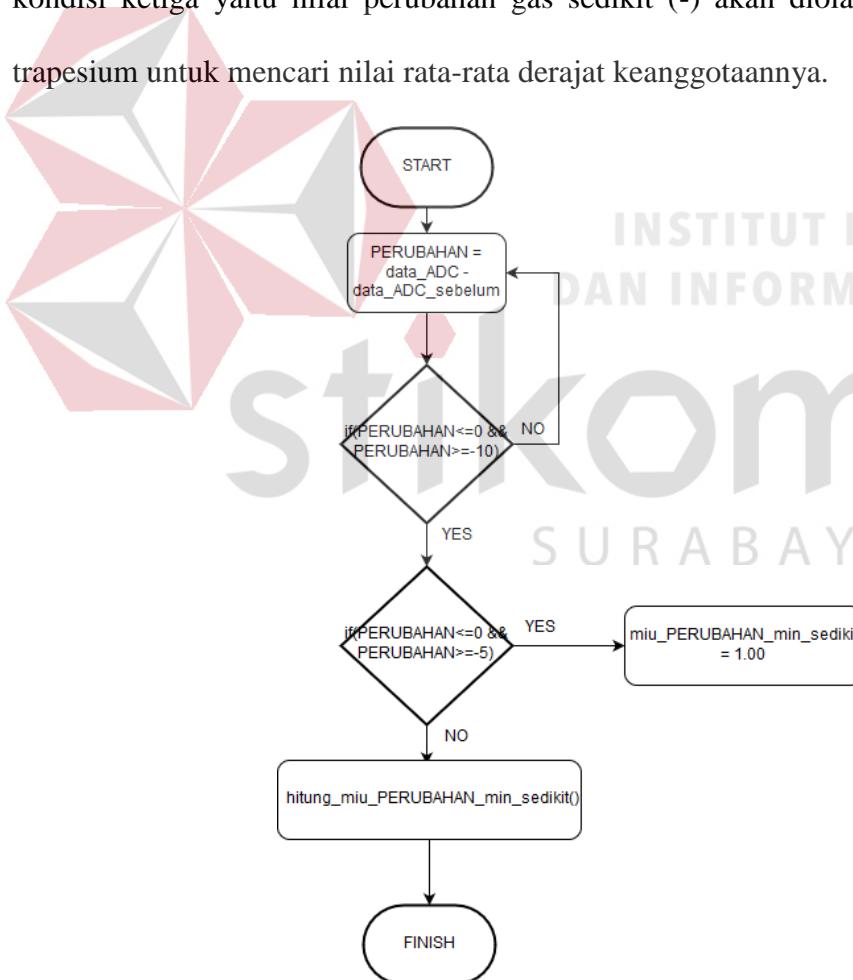
Gambar 3.44 menunjukkan *Flowchart Fuzzifikasi* perubahan gas banyak (+).

Kondisi pertama jika ADC sensor lebih dari sama dengan 15 dan ADC sensor kurang dari sama dengan 100. Kondisi ini menunjukkan rentang nilai *Fuzzifikasi* perubahan gas dengan konsentrasi sedikit (+). Jika kondisi pertama terpenuhi, maka akan masuk pada kondisi kedua. Pada kondisi kedua, jika ADC sensor lebih besar sama dengan 22 dan lebih kecil sama dengan 100 maka perubahan gas akan dikategorikan banyak(+) yaitu 1. Jika kondisi kedua tidak terpenuhi, maka akan dihitung untuk mencari rata-rata derajat keanggotaannya dengan rumus trapesium.

c. *Fuzzifikasi* Perubahan Gas (-)

Gambar 3.45 menunjukkan *Flowchart Fuzzifikasi* perubahan gas sedikit (-).

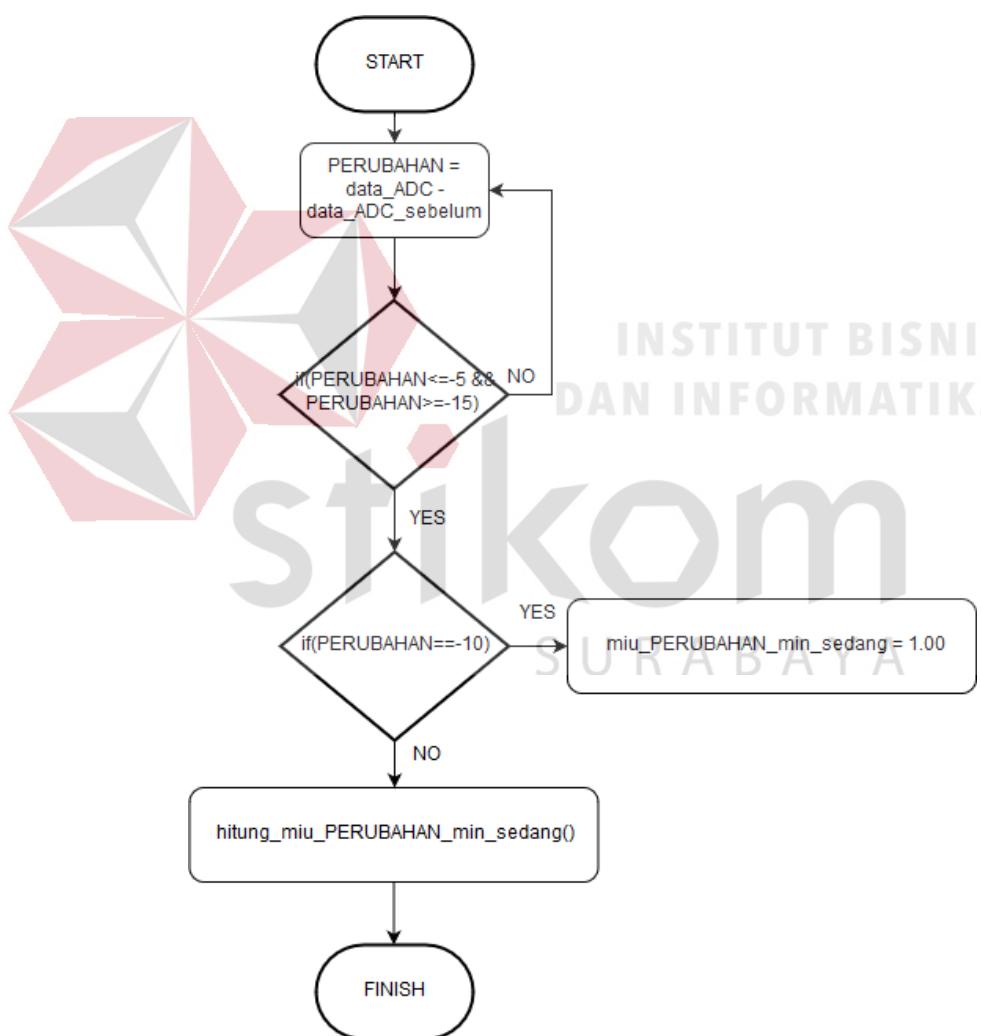
Kondisi pertama jika ADC sensor kurang dari sama dengan 0 dan ADC sensor lebih dari sama dengan -10. Kondisi ini menunjukkan rentang nilai *Fuzzifikasi* perubahan gas dengan konsentrasi sedikit (-). Jika kondisi pertama terpenuhi, maka akan masuk pada kondisi kedua. Pada kondisi kedua, jika ADC sensor kurang dari sama dengan 0 dan lebih dari sama dengan -5 maka perubahan gas akan dikategorikan sedikit (-) yaitu 1. Jika kondisi kedua tidak terpenuhi, maka akan tergolong pada kondisi ketiga yaitu nilai perubahan gas sedikit (-) akan diolah dengan rumus trapesium untuk mencari nilai rata-rata derajat keanggotaannya.



Gambar 3. 45 *Flowchart Fuzzifikasi* Perubahan Gas Sedikit (-)

Gambar 3.46 menunjukkan *Flowchart Fuzzifikasi* perubahan gas sedang (-).

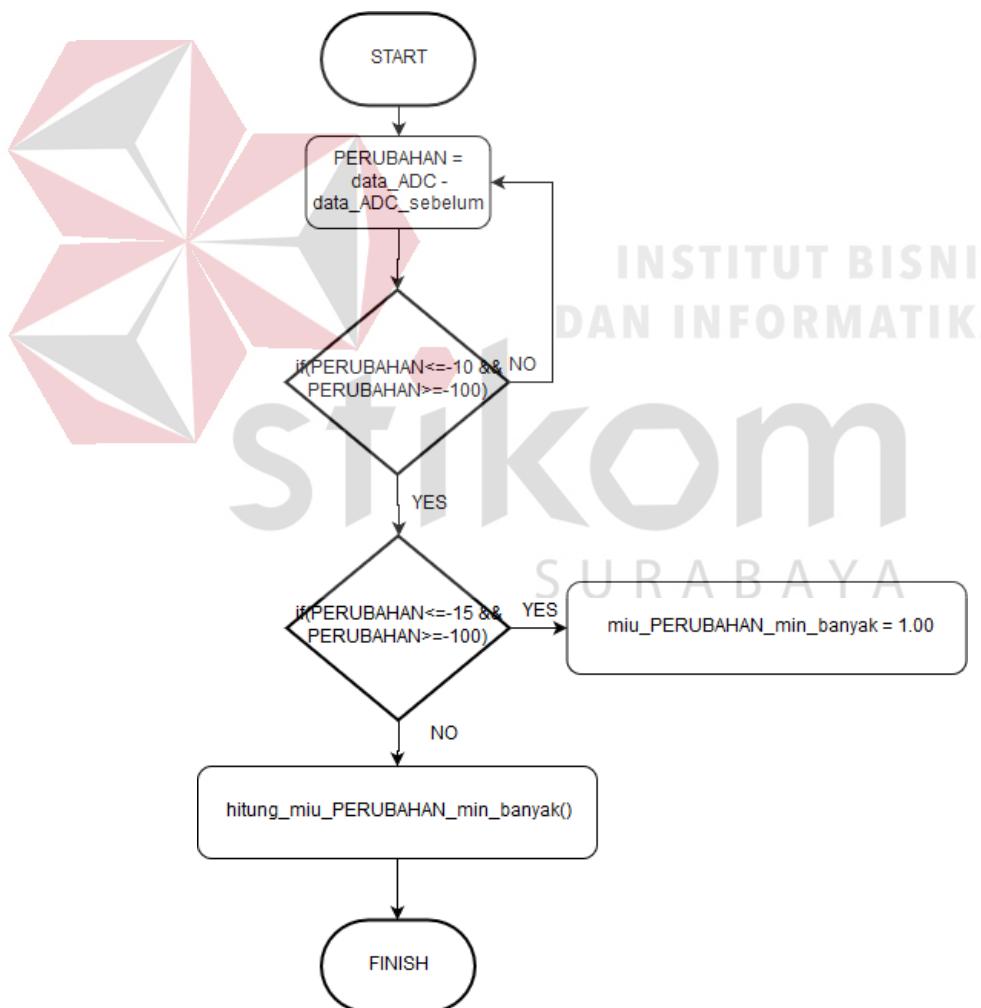
Kondisi pertama jika perubahan ADC sensor lebih dari sama dengan 5 dan perubahan ADC sensor kurang dari sama dengan -15. Kondisi ini menunjukkan rentang nilai *Fuzzifikasi* perubahan gas dengan konsentrasi sedang (+). Jika kondisi pertama terpenuhi, maka akan masuk pada kondisi kedua yaitu jika Perubahan ADC sama dengan -10 akan dikategorikan 1 dan jika kondisi kedua tidak terpenuhi maka dihitung untuk mencari rata-rata derajat keanggotaannya dengan rumus segitiga.



Gambar 3. 46 *Flowchart Fuzzifikasi* Perubahan Gas Sedang (-)

Gambar 3.47 menunjukkan *Flowchart Fuzzifikasi* perubahan gas banyak (-).

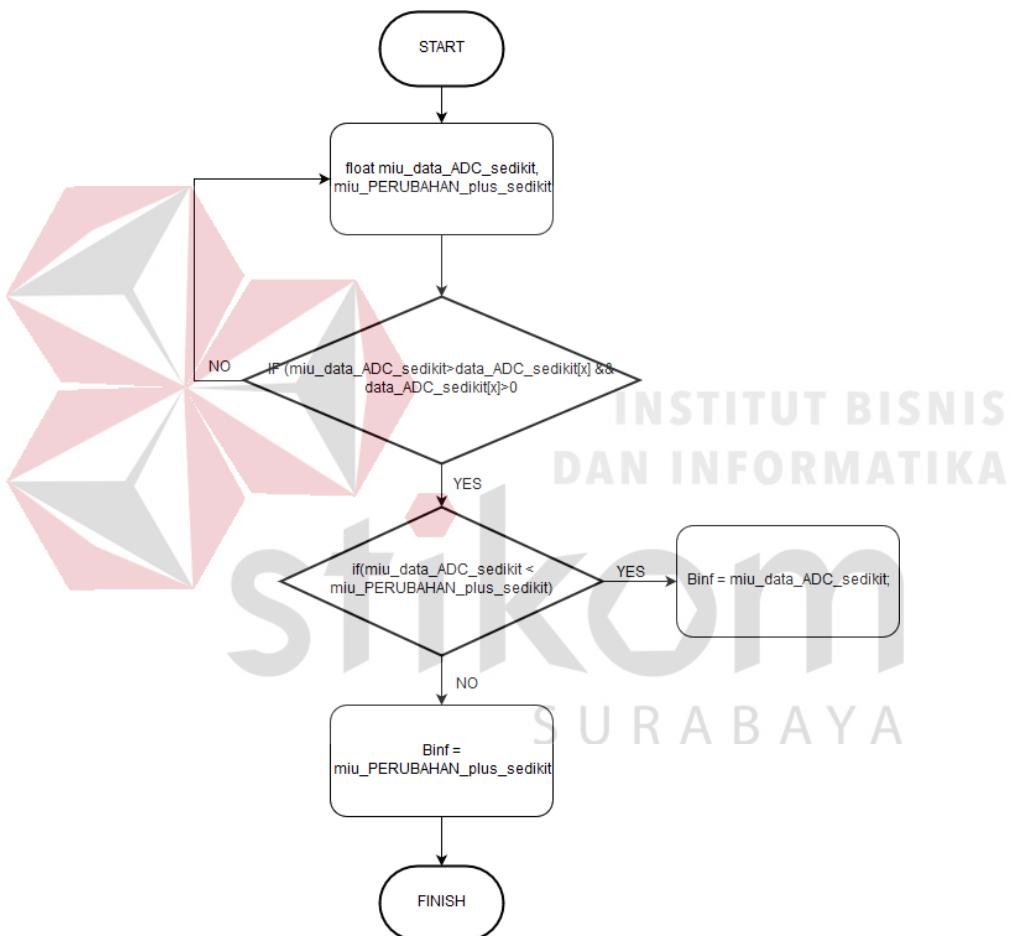
Kondisi pertama jika ADC sensor lebih dari sama dengan -10 dan ADC sensor kurang dari sama dengan -100. Kondisi ini menunjukkan rentang nilai *Fuzzifikasi* perubahan gas dengan konsentrasi sedikit (+). Jika kondisi pertama terpenuhi, maka akan masuk pada kondisi kedua. Pada kondisi kedua, jika ADC sensor lebih besar sama dengan -15 dan lebih kecil sama dengan -100 maka perubahan gas akan dikategorikan banyak(-) yaitu 1. Jika kondisi kedua tidak terpenuhi, maka akan dihitung untuk mencari rata-rata derajat keanggotaannya dengan rumus trapesium.



Gambar 3. 47 *Flowchart Fuzzifikasi* Perubahan Gas Banyak (-)

d. Evaluasi *Rules*

Nilai yang telah didapat dari *Fuzzifikasi* kemudian akan dimasukkan kedalam *rule* yang telah ditentukan sebelumnya untuk mendapatkan *output*. Evaluasi *rules* ini digunakan untuk mencari nilai minimum dari hasil proses *Fuzzifikasi*. Gambar 3.48 menunjukkan algoritma untuk mencari nilai minimum dari variabel *Fuzzifikasi*.



Gambar 3. 48 Flowchart Mencari Nilai Minimum

e. *Defuzzifikasi*

Metode yang digunakan pada proses *Defuzzifikasi* adalah *Weight-Average*.

*Defuzzifikasi* dilakukan untuk memperoleh nilai *real* dari PWM Kipas DC. Nilai dari evaluasi *rules*.

```
A_defuzzy=(Binf1*PWM_lambat)+(Binf2*PWM_lambat)+(Binf3*PWM_sedang)
+(Binf4*PWM_lambat)+(Binf5*PWM_lambat)+(Binf6*PWM_lambat)+(Binf7*P
WM_sedang)+(Binf8*PWM_sedang)+(Binf9*PWM_cepat)+(Binf10*PWM_sedang)
)+(Binf11*PWM_sedang)+(Binf12*PWM_lambat)+(Binf13*PWM_sedang)+(B
inf14*PWM_cepat)+(Binf15*PWM_cepat)+(Binf16*PWM_cepat)+(Binf17*PWM_s
edang)+(Binf18*PWM_sedang)
```

```
B_defuzzy=Binf1+Binf2+Binf3+Binf4+Binf5+Binf6+Binf7+Binf8+Binf9+Bi
nf10+Binf11+Binf12+Binf13+Binf14+Binf15+Binf16+Binf17+Binf18
```

```
Hasil_defuzzy = A_defuzzy/B_defuzzy
```

Pada variabel A\_defuzzy merupakan nilai minimum dari hasil proses evaluasi *rules* yang dikalikan dengan PWM Kipas DC yang telah ditetapkan sebelumnya, sedangkan pada variabel B\_defuzzy merupakan penjumlahan nilai minimal dari evaluasi *rules*. Nilai dari variabel Hasil\_defuzzy merupakan nilai PWM yang mengendalikan kecepatan Kipas DC.

## **BAB IV**

### **HASIL PENGUJIAN DAN PENGAMATAN**

Dalam bab ini penulis akan menguraikan dan menjelaskan hasil analisis pengujian dari hasil penelitian Tugas Akhir ini yang telah dilakukan, pengujian dilakukan dalam beberapa bagian yang disusun dalam urutan dari yang sederhana menuju sistem yang lengkap. Pengujian dilakukan meliputi pengujian perangkat lunak (*software*), aplikasi *smartphone Android* dan perangkat keras (*hardware*). Diharapkan hasil yang didapat adalah suatu sistem yang dapat menjalankan rancangan alat berjalan dengan baik, optimal, dan bermanfaat.

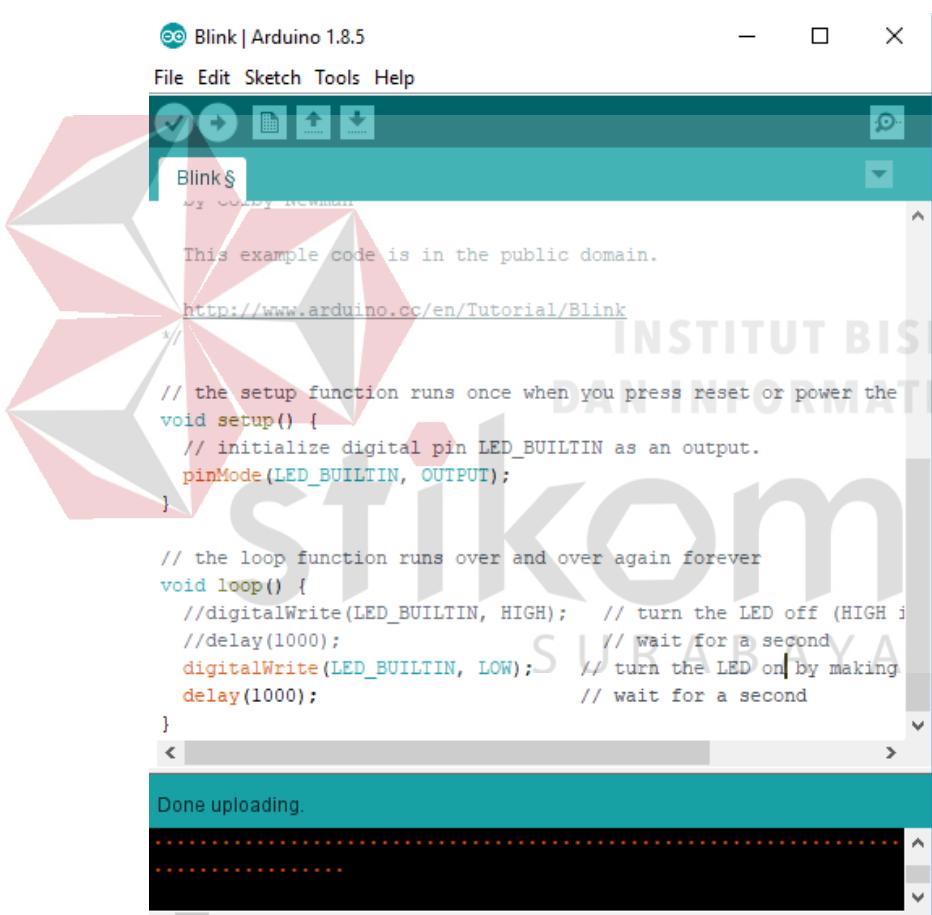
#### **4.1 Pengujian Wemos D1 Mini**

Wemos D1 Mini yang digunakan akan diuji dengan sebuah prosedur untuk memastikan tidak adanya kerusakan pada sistemnya. Pengujian ini juga berfungsi untuk memastikan Wemos D1 Mini yang digunakan mampu berjalan dengan baik. Alat yang digunakan pada pengujian Wemos D1 Mini adalah Laptop, Kabel *USB type B* dan Wemos D1 Mini. Berikut merupakan prosedur pengujian Wemos D1 Mini:

1. Menghubungkan Wemos D1 Mini dengan kabel *USB type B*.
2. Menjalankan program Arduino IDE pada laptop.
3. Menghubungkan kabel *USB* dengan laptop.
4. Membuka program yang *diupload* pada Wemos D1 Mini.
5. Mengatur *board*, *serial port* dan *programmer*.
6. *Mengupload* program dan tunggu hingga proses selesai.

7. Setelah proses *upload* selesai, dapat dievaluasi apakah program berhasil di-download atau tidak oleh Wemos D1 Mini.

Pengujian pada Wemos D1 Mini mengevaluasi ada atau tidaknya *comment error* pada saat proses *upload*. Proses *upload* ditampilkan oleh Gambar 4.1(a). Saat program berjalan, LED pada Wemos akan menyala yang ditampilkan oleh Gambar 4.1(b). *Comment error* dapat terjadi pada sambungan antara kabel *USB* dan Wemos D1 Mini.

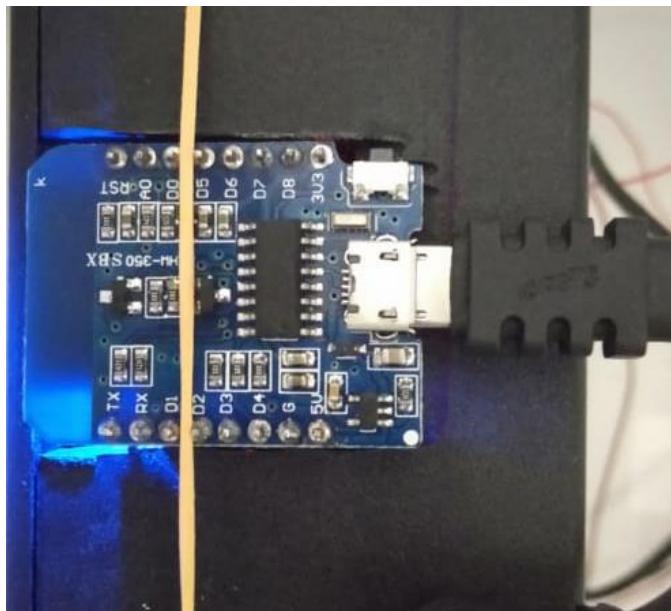


The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** Blink | Arduino 1.8.5
- Menu Bar:** File Edit Sketch Tools Help
- Toolbar:** Includes icons for upload, download, and other functions.
- Sketch Area:** Displays the "Blink" sketch code:

```
This example code is in the public domain.  
http://www.arduino.cc/en/Tutorial/Blink  
  
// the setup function runs once when you press reset or power the  
void setup() {  
    // initialize digital pin LED_BUILTIN as an output.  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
    //digitalWrite(LED_BUILTIN, HIGH); // turn the LED off (HIGH is  
    //delay(1000); // wait for a second  
    digitalWrite(LED_BUILTIN, LOW); // turn the LED on by making  
    delay(1000); // wait for a second  
}
```
- Status Bar:** Shows "Done uploading."

(A)



(B)

Gambar 4. 1 (A) Proses *Upload* Dari Arduino IDE Menuju Wemos D1 Mini Dan (B) LED Pada Wemos Menyalakan Saat Program Berjalan

#### 4.2 Pengujian Sensor Figaro TGS2610

Pengujian sensor Figaro TGS2610 bertujuan untuk memastikan sensor dapat berjalan dengan baik serta tidak terdapat kerusakan pada perangkatnya. Alat yang digunakan pada pengujian ini adalah Laptop, Wemos D1 Mini, sensor Figaro TGS2610, Kabel *USB type B*, serta kabel *jumper*. Langkah-langkah yang dilakukan pada pengujian sensor ini adalah :

1. Menghubungkan sensor Figaro TGS2610 ke Wemos D1 Mini dengan menggunakan kabel *jumper*.
2. Menjalankan Arduino IDE pada Laptop.
3. Menghubungkan kabel *USB type B* dengan Laptop.
4. Membuka program dan kemudian meng*upload* ke Wemos D1 Mini.
5. Mengatur *board*, *serial port* dan *programmer* yang digunakan.
6. Menyemprotkan gas ke dalam *box*.

7. Mengamati perubahan sensor pada *serial monitor*.

Pada proses pembacaan gas oleh sensor, nilai ADC gas akan berubah dari sebelumnya. Proses pengujian ini dinyatakan baik jika sensor mampu membaca perubahan nilai ADC gas saat gas disemprotkan ke dalam *box*. Proses pembacaan sensor Figaro TGS2610 pada *serial monitor* ditampilkan pada Gambar 4.2.

```

COM4
ADC_gas= 137.00
PERUBAHAN_gas= 1.00
18:19:10

ADC_gas= 145.00
PERUBAHAN_gas= 8.00
18:19:11

ADC_gas= 154.00
PERUBAHAN_gas= 9.00
18:19:12

ADC_gas= 171.00
PERUBAHAN_gas= 17.00
18:19:13

 Autoscroll

```

Gambar 4. 2 Pembacaan Sensor Figaro TGS2610 Pada *Serial Monitor*

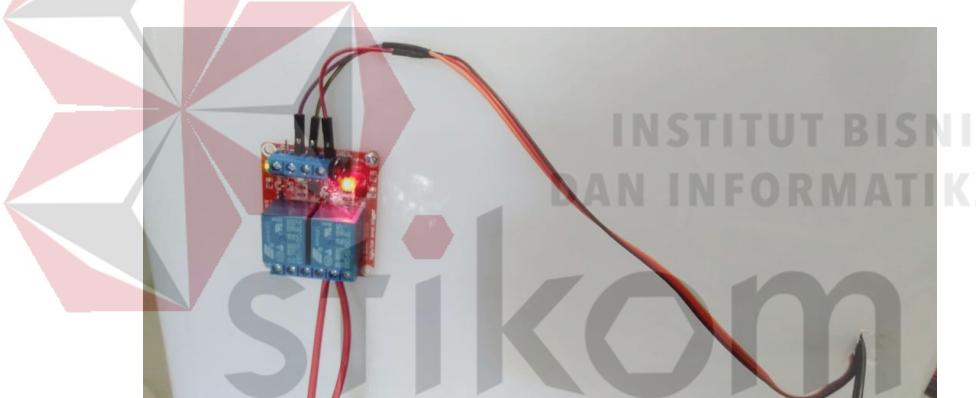
### 4.3 Pengujian *Relay* Dan *Valve*

Pengujian ini bertujuan untuk memastikan *relay* dan *valve* berjalan dengan baik dan tidak terdapat kerusakan. Alat yang digunakan dalam pengujian ini meliputi Laptop, Wemos D1 Mini, *Relay*, *Valve*, kabel *USB type B*, *adaptor 12 V* serta kabel *jumper*. Langkah-langkah yang dilakukan untuk menguji *relay* dan *valve* adalah sebagai berikut:

1. Membuka Arduino IDE pada Laptop.
2. Menghubungkan rangkaian pada Wemos D1 Mini – *Relay, adaptor 12 V – valve*, serta *valve – relay* dengan kabel *jumper* sesuai rangkaian masing-masing.
3. Mengatur SSID dan *password* pada Arduino IDE.
4. *Mengupload* program ke Wemos D1 Mini.
5. Membuka aplikasi Blynk dan melakukan pengujian *valve* pada aplikasinya.

Dari pengujian tersebut, diperoleh hasil proses pengujian *Relay* dan *valve*.

Apabila *Relay* bernilai *high* maka *Relay* akan terbuka sehingga dapat menyalaakan api dan jika *Relay* bernilai *low* maka *valve* akan tertutup maka api akan mati.



Gambar 4. 3 *Relay* Bernilai *High*

Pada Gambar 4.3, *Relay* bernilai *High* yang ditandai oleh lampu indikator yang menyala. Pada Gambar 4.4, kompor bisa dinyalakan karena kondisi *valve* terbuka maka aliran gas dapat mengalir ke kompor.



Gambar 4. 4 Kompor Menyala Saat *Valve* Terbuka



Gambar 4. 5 Relay Bernilai *Low*

Pada Gambar 4.5 *Relay* bernilai *Low* yang ditandai oleh lampu indikator yang menyala. Pada Gambar 4.6 kompor tidak bisa dinyalakan karena kondisi *valve* tertutup maka aliran gas tidak dapat mengalir ke kompor.



Gambar 4. 6 Kompor Tidak Menyalा Saat *Valve* Tertutup

#### 4.4 Pengujian *Real Time Clock*

Pengujian *Real Time Clock* (RTC) berfungsi untuk memastikan RTC berjalan dengan baik dan tidak terdapat kerusakan. Alat yang digunakan pada pengujian ini meliputi Laptop, Wemos D1 Mini, RTC, Kabel usb type B serta kabel *jumper*.

Langkah-langkah yang dilakukan adalah:

1. Menghubungkan Wemos D1 Mini dengan RTC menggunakan kabel *jumper*.
2. Membuka aplikasi Arduino IDE.
3. Menyambungkan laptop dengan kabel *USB type B*.
4. Membuka program yang digunakan serta mengupload ke Wemos D1 Mini.
5. Mengatur *board*, *serial port*, *programmer*.
6. Mengamati waktu pada *serial monitor* dan *smartphone*.

Dari hasil pengujian, didapati proses pembacaan RTC pada *serial monitor* yang ditunjukkan oleh Gambar 4.7(a), sedangkan Gambar 4.7(b) menunjukkan waktu pada *smartphone*. Karena tidak terdapat perbedaan waktu antara hasil pembacaan oleh RTC dan waktu pada *smartphone*, maka RTC berjalan dengan baik.



Gambar 4. 7 (A) Proses Pembacaan RTC Pada *Serial Monitor* (B) *Real Time* Pada *Smartphone*

#### 4.5 Pengujian Notifikasi

Pengujian Notifikasi berguna untuk memastikan pengiriman notifikasi oleh aplikasi Blynk berjalan dengan baik. Notifikasi akan muncul pada *smartphone Android*. Alat yang digunakan pada pengujian ini meliputi Laptop, Wemos D1 Mini, Aplikasi Blynk, Kabel *USB type B*, Kabel *Jumper*, sensor Figaro TGS2610, LCD dan RTC, sedangkan langkah-langkah yang dilakukan dalam pengujian ini adalah:

1. Membuka aplikasi Arduino IDE pada Laptop.
2. Menghubungkan Wemos D1 Mini dengan sensor Figaro TGS2610, RTC dan LCD menggunakan kabel *jumper* sesuai rangkaian masing-masing.
3. *Mengupload* program menuju Wemos D1 Mini.

4. Menginstall aplikasi Blynk pada *smartphone Android* dan *login* sesuai akun *user*.
5. Menyemprotkan gas ke dalam *box*.
6. Mengamati data pada aplikasi Blynk.
7. Menunggu notifikasi hingga muncul.

Hasil pengujian akan membuktikan proses notifikasi berjalan baik jika aplikasi Blynk mampu menampilkan notifikasi pada layar *smartphone Android*. Pada pengujian ini, LCD menampilkan nilai ADC sebesar 252 yang ditampilkan oleh Gambar 4.8. Karena melebihi batas yang sudah ditentukan, yaitu ADC=250, maka aplikasi Blynk akan menampilkan notifikasi pada *smartphone Android* yang ditampilkan oleh Gambar 4.9.



Gambar 4. 8 Tampilan LCD Pengujian Notifikasi



Gambar 4. 9 Tampilan Notifikasi Pada *Smartphone Android*

#### 4.6 Pengujian *Monitoring* Jarak Jauh Alat Dengan *Android*

Pengujian ini berfungsi untuk memastikan nilai kebocoran gas yang ditampilkan oleh aplikasi Blynk memiliki nilai yang sama dengan yang ditampilkan oleh LCD secara jarak jauh. Pengujian ini berfungsi untuk memastikan kerja sistem walaupun dalam jarak jauh. *Monitoring* jarak jauh ini berfungsi saat *user* sedang berada jauh dengan sistem, sehingga *user* tetap bisa melakukan *monitoring* terhadap kebocoran gas. Alat yang digunakan pada pengujian ini meliputi sensor Figaro TGS2610, Wemos D1 Mini, Laptop, Kabel *USB type B*, aplikasi Blynk *smartphone Android*, koneksi interner, LCD dan juga kabel *jumper*. Langkah-langkah yang digunakan adalah :

1. Menghubungkan Wemos D1 Mini dengan sensor figaro TGS2610, RTC, LCD menggunakan kabel *jumper* sesuai rangkaian masing masing.
2. Membuka aplikasi Arduino IDE pada Laptop.
3. Mengatur SSID dan *password*.
4. Mengupload program ke Wemos D1 Mini.
5. Login pada aplikasi Blynk.
6. Melakukan pengamatan data kebocoran gas pada aplikasi Blynk secara jarak jauh.

Pengujian jarak jauh untuk mengamati keakuratan data yang ditampilkan pada LCD dan aplikasi Blynk. Data pada LCD ditampilkan oleh Gambar 4.10, sedangkan data pada aplikasi Blynk ditampilkan oleh Gambar 4.11. Pengamatan yang dilakukan yaitu dengan mencocokkan data waktu dengan nilai kebocoran antara LCD dan juga aplikasi Blynk secara jarak jauh. Proses pengujian ini penulis

melakukan dengan *monitoring* lokasi *user* yang berada di Banyuwangi dan lokasi perangkat kebocoran gas berada di Surabaya.



Gambar 4. 10 Tampilan Data Pada LCD



Gambar 4. 11 Tampilan Data Pada Aplikasi Blynk

Proses *delay* maupun *monitoring* kebocoran gas antara alat dengan penerimaan data pada aplikasi Blynk hampir tidak memiliki *delay* sama sekali dengan kecepatan jaringan internet 11,1 Mbps. Hal tersebut dapat diamati pada hasil pengujian Tabel 4.1.

Tabel 4. 1 Pengujian *Monitoring* Pada Aplikasi Blynk Dengan Kecepatan Jaringan Internet 11,1 *Mbps*

Data Ke-	Waktu Data Tampil						<i>Delay</i> (s)	
	Alat			Blynk				
	Jam	Menit	Detik	Jam	Menit	Detik		
1	13	44	12	13	44	12	0	
2	13	44	13	13	44	13	0	
3	13	44	14	13	44	14	0	
4	13	44	15	13	44	15	0	
5	13	44	16	13	44	16	0	
6	13	44	17	13	44	17	0	
7	13	44	18	13	44	18	0	
8	13	44	19	13	44	19	0	
9	13	44	20	13	44	20	0	
10	13	44	21	13	44	21	0	
11	13	44	22	13	44	22	0	
12	13	44	23	13	44	23	0	
13	13	44	24	13	44	24	0	
14	13	44	25	13	44	25	0	
15	13	44	26	13	44	26	0	
16	13	44	27	13	44	27	0	
17	13	44	28	13	44	28	0	
18	13	44	29	13	44	29	0	
19	13	44	30	13	44	30	0	
20	13	44	31	13	44	31	0	
21	13	44	32	13	44	32	0	
22	13	44	33	13	44	33	0	
23	13	44	34	13	44	34	0	
24	13	44	35	13	44	35	0	
25	13	44	36	13	44	36	0	
26	13	44	37	13	44	37	0	
27	13	44	38	13	44	38	0	
28	13	44	39	13	44	39	0	
29	13	44	40	13	44	40	0	
30	13	44	41	13	44	41	0	
Rata-rata <i>Delay</i>							0	

Pada Tabel 4.2 dapat diketahui bahwa *monitoring* data dengan kecepatan jaringan internet 0,08 *Mbps* akan berpengaruh pada hasil *monitoring* kebocoran gas pada aplikasi Blynk.

Tabel 4. 2 Pengujian *Monitoring* Pada Aplikasi Blynk Dengan Kecepatan Jaringan Internet 0,08 Mbps

Data Ke-	Waktu Data Tampil						<i>Delay</i> (s)	
	Alat			Blynk				
	Jam	Menit	Detik	jam	Menit	detik		
1	21	1	56	21	1	56	0	
2	21	1	57	21	2	57	0	
3	21	1	58	21	2	59	1	
4	21	2	0	21	2	0	0	
5	21	2	1	21	2	3	2	
6	21	2	4	21	2	4	0	
7	21	2	5	21	2	5	0	
8	21	2	6	21	2	8	2	
9	21	2	7	21	2	9	2	
10	21	2	8	21	2	10	2	
11	21	2	9	21	2	13	4	
12	21	2	10	21	2	14	4	
13	21	2	12	21	2	15	3	
14	21	2	13	21	2	16	3	
15	21	2	14	21	2	17	3	
16	21	2	15	21	2	18	3	
17	21	2	16	21	2	19	3	
18	21	2	17	21	2	20	3	
19	21	2	18	21	2	21	3	
20	21	2	19	21	2	22	3	
21	21	2	20	21	2	23	3	
22	21	2	21	21	2	24	3	
23	21	2	22	21	2	25	3	
24	21	2	23	21	2	26	3	
25	21	2	24	21	2	27	3	
26	21	2	25	21	2	28	3	
27	21	2	26	21	2	29	3	
28	21	2	27	21	2	30	3	
29	21	2	28	21	2	31	3	
30	21	2	29	21	2	32	3	
Rata-rata Delay (s)						2,37		

Berdasarkan data pada Tabel 4.1 dan 4.2 dapat diketahui bahwa delay waktu yang ditunjukkan oleh alat dan aplikasi Blynk dipengaruhi oleh kecepatan jaringan. Saat kecepatan jaringan internet dibawah 0,08 Mbps, waktu yang ditunjukkan oleh alat dan aplikasi Blynk memiliki rata-rata *delay* sebesar 2,37 detik, sedangkan saat

kecepatan jaringan internet diatas 11,1 Mbps tidak terdapat delay, sehingga dilakukan uji *monitoring* lanjutan menggunakan kecepatan internet diatas 11,1 *Mbps* selama 3 hari yang ditampilkan pada Lampiran 1.

#### 4.7 Pengujian Sistem *Fuzzy Logic*

Pengujian Sistem *Fuzzy Logic* berfungsi untuk memastikan sistem bekerja sesuai *Fuzzifikasi*, evaluasi *rules* dan *Defuzzifikasi* yang telah ditetapkan. Alat yang digunakan pada pengujian ini adalah Wemos D1 Mini, *Motor Driver*, Kipas DC, kabel *USB type B*, kabel *jumper* dan laptop. Pengujian ini dilakukan dengan langkah sebagai berikut :

1. Menghubungkan Wemos D1 Mini dengan *Motor Driver* serta Kipas DC menggunakan kabel *jumper*.
2. Membuka aplikasi Arduino IDE.
3. Menyambungkan laptop dengan kabel *USB type B*.
4. Mengupload program ke Wemos D1 Mini.
5. Mengamati data pada *serial monitor*.

Pengujian ini dilakukan dengan mengamati data yang ditampilkan oleh *serial monitor*. Pada pengujian ini, jika diketahui ADC sebesar 260 dan perubahan ADC sebesar 10, maka :

- a. *Fuzzifikasi*

$$\begin{aligned}
 \mu_{ADC\text{ sedikit}} &= \text{Max} \left( \min \left( \frac{x-a}{b-a}, 1, \frac{d-x}{d-c} \right), 0 \right) \\
 &= \text{Max} \left( \min \left( \frac{260-200}{200-200}, 1, \frac{350-260}{350-300} \right), 0 \right) \\
 &= \text{Max} \left( \min \left( \frac{60}{0}, 1, \frac{90}{50} \right), 0 \right)
 \end{aligned}$$

$$= \text{Max}(1,0) = 1$$

$$\begin{aligned}\mu_{\text{perubahan\_ADC sedikit}} &= \text{Max} \left( \min \left( \frac{x-a}{b-a}, 1, \frac{d-x}{d-c} \right), 0 \right) \\ &= \text{Max} \left( \min \left( \frac{10-1}{1-1}, 1, \frac{15-10}{15-8} \right), 0 \right) \\ &= \text{Max} \left( \min \left( \frac{9}{0}, 1, \frac{5}{7} \right), 0 \right) \\ &= \text{Max}(0.7,0) = 0.71\end{aligned}$$

$$\begin{aligned}\mu_{\text{perubahan\_ADC sedang}} &= \text{Max} \left( \min \left( \frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right) \\ &= \text{Max} \left( \min \left( \frac{10-8}{15-8}, \frac{22-10}{22-15} \right), 0 \right) \\ &= \text{Max} \left( \min \left( \frac{2}{7}, \frac{12}{7} \right), 0 \right) \\ &= \text{Max}(0.29,0) = 0.29\end{aligned}$$

b. Evaluasi Rules

*Rule 1 = ADC sedikit AND Perubahan plus Sedikit THEN lambat*

$$\text{Rule 1} = \text{Min}(1,0.71) = 0.71$$

*Rule 2 = ADC sedikit AND Perubahan plus Sedang THEN sedang*

$$\text{Rule 2} = \text{Min}(1,0.29) = 0.29$$

c. Defuzzifikasi

$$\begin{aligned}\text{Defuzzifikasi} &= \frac{(Rule 1 \times PWM_{lambat}) + (Rule 2 \times PWM_{sedang})}{Rule 1 + Rule 2} \\ &= \frac{(0.71 \times 300) + (0.29 \times 512)}{1} \\ &= 361\end{aligned}$$

Tabel 4. 3 Data Pengujian Perhitungan Sistem *Fuzzy* Pada Program Dan Penghitungan Secara Manual

No	Input		PWM	
	ADC	Perubahan ADC	Sistem	Manual
1	260	10	361	361
2	440	-7	512	512
3	352	-11	512	512
4	500	12	807	807
5	361	19	905	904
6	355	16	610	608
7	435	-15	610	608
8	445	-17	512	512
9	445	-11	512	512
10	250	-20	300	300
11	600	9	1200	1200
12	550	-4	856	856
13	470	-30	512	512
14	472	30	1200	1200
15	473	40	1200	1200
16	300	-40	300	300
17	301	-40	304	304
18	302	-40	308	308
19	303	-39	312	312
20	304	-38	316	316
21	305	-37	321	321
22	304	-36	316	316
23	306	-35	325	325
24	307	-34	329	329
25	308	-33	333	333
26	309	-32	338	338
27	400	-31	512	512
28	401	-30	512	512
29	402	-29	512	512
30	403	-28	512	512

Pengamatan sistem *Fuzzy Logic* ini dilakukan sebanyak 30 kali yang ditampilkan pada Tabel 4.3. Berdasarkan Tabel 4.3 perhitungan sistem *Fuzzy* pada program dan perhitungan secara manual tersebut bisa diketahui bahwa sistem *Fuzzy* yang digunakan pada alat kebocoran gas memiliki tingkat kesesuaian yang tinggi. Terdapat beberapa nilai *Fuzzy* yang tidak sama karena perbedaan logika program

pada tipe data. Sistem *Fuzzy* ini menjadi faktor utama dalam menentukan tingkat kecepatan kipas. Perubahan kecepatan kipas berfungsi untuk mengeluarkan gas LPG yang bocor ke lingkungan luar.

Tabel 4. 4 Hasil Pengujian Sistem *Fuzzy* Pada Alat

Uji Ke-	Nilai PWM	Sistem <i>Fuzzy</i>					Berhasil	Gagal
		Lambat	Sedang	Cepat	Sesuai	Tidak Sesuai		
1	300	V			V		V	
2	512		V		V		V	
3	512		V		V		V	
4	600			V	V		V	
5	900			V	V		V	
6	1200			V	V		V	
7	380		V		V		V	
8	512		V		V		V	
9	670			V	V		V	
10	512		V		V		V	

Dari hasil pengujian sistem *Fuzzy* pada alat kebocoran gas yang dilakukan secara *realtime* dan mengamati kecepatan kipas dapat disimpulkan bahwa sistem *Fuzzy* juga bekerja secara akurat dan baik. Data hasil pengujian tersebut bisa diamati pada tabel 4.4.

Tabel 4. 5 Hasil Pengujian Sistem *Fuzzy* Kembali Ke Nilai ADC Sebelum Ada Gas

No	Input		Durasi Sistem Yang Berjalan (S)	Kecepatan ADC Berkurang Per-Detik
	Nilai ADC Sebelum Ada Gas	Nilai ADC Setelah Ada Gas		
1	186	398	195	1,087
2	198	354	207	0,754
3	264	437	201	0,861
4	261	449	227	0,828
5	196	356	355	0,451
6	209	260	225	0,227
7	206	290	252	0,333
8	199	340	281	0,502

No	Input		Durasi Sistem Yang Berjalan (S)	Kecepatan ADC Berkurang Per-Detik
	Nilai ADC Sebelum Ada Gas	Nilai ADC Setelah Ada Gas		
9	197	589	420	0,933
10	197	536	408	0,831
11	189	267	141	0,553
12	190	267	149	0,517
13	194	317	183	0,672
14	190	334	243	0,593
15	206	296	152	0,592
16	197	267	141	0,496
17	197	228	152	0,204
18	198	295	195	0,497
19	199	248	181	0,271
20	199	225	168	0,155
21	199	285	210	0,410
22	198	214	141	0,113
23	199	206	130	0,054
24	195	280	209	0,407
25	199	240	150	0,273
26	199	277	182	0,429
27	199	243	169	0,260
28	200	236	167	0,216
29	199	234	154	0,227
30	199	230	157	0,197
Rata-rata (ADC/s)				0,465

Adapun hasil pengujian alat kebocoran gas terkait nilai ADC sebelum ada gas dan setelah ada gas ditampilkan oleh Tabel 4.5. Pengujian ini menggunakan *box* berdimensi 60x43x38 cm. Sistem *Fuzzy* yang digunakan pada alat kebocoran gas ini menghasilkan penurunan kecepatan rata-rata sebesar 0,465 ADC/detik untuk kembali pada kondisi nilai ADC sebelum ada gas. Untuk mencari nilai ADC/s, Rumus yang digunakan adalah sebagai berikut:

$$ADC/s = \frac{ADC \text{ setelah ada gas} - ADC \text{ sebelum ada gas}}{\text{Durasi Sistem Berjalan}}$$

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan hasil perancangan serta pengujian sistem rancang bangun kebocoran gas ini, maka didapati kesimpulan sebagai berikut:

1. *Delay* waktu yang ditunjukkan oleh alat dan aplikasi Blynk dipengaruhi oleh kecepatan jaringan. Saat kecepatan jaringan internet dibawah 0,08 Mbps, waktu yang ditunjukkan oleh alat dan aplikasi Blynk memiliki rata-rata *delay* sebesar 2,37 detik dari 30 data uji, sedangkan saat kecepatan jaringan internet diatas 11,1 Mbps tidak terdapat *delay*.
2. Sistem *Fuzzy* yang digunakan pada alat kebocoran gas ini menghasilkan penurunan kecepatan rata-rata sebesar 0,465 ADC/detik untuk kembali pada kondisi nilai ADC sebelum ada gas.

#### 5.2 Saran

Pengembangan yang dapat dilakukan setelah pembuatan sistem rancang bangun ini adalah sebagai berikut:

1. Menggunakan kondisi internet yang stabil agar tidak terdapat *delay* pada saat pengiriman data selama proses sistem bekerja.
2. Peneliti berikutnya dapat mencoba untuk menggunakan beberapa metode lain yang lebih sesuai sehingga tidak ada data dari sistem yang terbaca ambigu karena berada pada titik perpotongan dari sistem.

## DAFTAR PUSTAKA

- Alpares, R., 2016. *Aplikasi Real Time Clock DS3231 sebagai Penjejak Matahari pada Solar Berbasis Arduino*, Palembang: Politeknik Negeri Sriwijaya.
- Aryandhi, Y. D. & Talakua, M. W., 2013. Penerapan Inferensi Fuzzy untuk Pengendali Suhu Ruangan Secara Otomatis pada Air Conditioner (AC). *Prosiding FMIPA Universitas Pattimura*, pp. 177-184.
- Aslamia, S., 2015. *Robot Pendekripsi Manusia sebagai Sistem Keamanan Ruangan Menggunakan Sensor PIR dengan Media Komunikasi Xbee Berbasis Arduino Leonardo (Sub Bahasan: Software)*, Palembang: Politeknik Negeri Sriwijaya.
- Badan Perlindungan Konsumen Nasional, 2010. *Data Insiden Kecelakaan Ledakan LPG s.d. Juni 2010*, Jakarta: Badan Perlindungan Konsumen Nasional.
- Chistian, J. & Komar, N., 2013. Prototipe Sistem Pendekripsi Kebocoran Gas LPG Menggunakan Sensor GAs MQ2, Board Arduino Duemilanove, Buzzer, dan Arduino GSM Shield pada PT. Alfa Retailindo (Carrefour Pasar Minggu). *TICOM Vol.2*, pp. 58-64.
- Danur, B. D., Satria, D. & Aisuwaryya, R., 2013. *Sistem Pendekripsi Kebocoran Gas LPG menggunakan Mikrokontroler*, Padang: Universitas Andalas Padang.
- Febrianto, N., Susanto, E. & Wibowo, A. S., 2016. Rancang Bangun Kontrol Suhu Air pada Prototipe Pemanas Air Menggunakan Logika Fuzzy. *e-Proceeding of Engineering*, Volume Vol.3, pp. 4253-4261.
- Kusumadewi, S., 2004. *Aplikasi Logika Fuzzy Untuk Pendukung Keputusan*. Yogyakarta: Graha Ilmu.
- Limantara, A. D., Candra, A. I. & Mudjanarko, S. W., 2017. Manajemen Data Lalu Lintas Kendaraan Berbasis Sistem Internet Cerdas Ujicoba Implementasi di Laboratorium Universitas Kadiri. *Seminar Nasional Sains dan Teknologi 2017 Fakultas Teknik Universitas Muhammadiyah Jakarta*, pp. 1-11.
- Madjid, A. R., 2019. Prototype *Monitoring Arus, dan Suhu pada Transformator Distribusi Berbasis Internet of Things*. *Jurnal Teknik Elektro*, Volume Vol.8 No.1, pp. 111-119.
- Mendonca S, M., Fernando GS, Mahfud B & Pramulyono, 2013. Sistem Pengaman Kebocoran Liquified Petroleum Gas (LPG) dan Pemadam Api pada Rumah Makan/Restoran. *Widya Teknika Vol.21*, pp. 19-26.

- Nurnaningsih, D., 2018. Pendekripsi Kebocoran Tabung LPG melalui SMS Gateway Menggunakan Sensor MQ-2 Berbasis Arduino Uno. *Jurnal Teknik Informatika Vol 11*, pp. 121-126.
- Prabowo, J. C., 2017. Prototype Kontrol Suhu dan Kelembaban pada Gudang Penyimpanan Kopi dengan Komunikasi WI-FI Berbasis Arduino Uno. *Program Studi Diploma III Teknik Elektronika Universitas Jember*, pp. 1-58.
- Prakoso, I. P., 2018. Voice Controlled Home Automation System Menggunakan Mikrokontroler Wemos. *Jurusan Teknik Informatika Fakultas Teknologi Industri UII*, pp. 1-85.
- Putra, O. A., Hersyah, M. H. & Firdaus, 2016. *Identifikasi Aroma Teh dengan E-Nose Menggunakan Metode Backpropagation*, Jakarta: Fakultas Teknik Universitas Muhammadiyah Jakarta.
- Putra, R. F. h., Lhaksmana, K. M. & Adytia, D., 2016. Aplikasi IoT untuk Rumah Pintar dengan Fitur Prediksi Cuaca. *e-Proceeding of Engineering*, Volume Vol.5, pp. 1746-1760.
- Putri, D. M., 2017. Mengenal Wemos D1 Mini dalam Dunia IoT. *ilmuti.org*, pp. 1-8.
- Riyadi, S. & Purnama, B. E., 2013. Sistem Pengendalian Keamanan Pintu Rumah Berbaasis SMS (Short Message Service) Menggunakan Mikrokontroler ATMEGA 8535. *IJNS*, pp. 7-11.
- Rofiq, M., 2013. Perancangan Manajemen Bandwidth Internet Menggunakan Metode Fuzzy Sugeno. *Jurnal Ilmiah Teknologi dan Informasi ASIA*, Volume Vol.7 No.1, pp. 1-15.
- Saleh, M. & Haryanti, M., 2017. Rancang Bangun Sistem Keamanan Rumah menggunakan Relay. *Jurnal Teknologi Elektro*, Volume Vol.8 No3, pp. 181-186.
- Soemarsono, B. E., Listriasri, E. & Kusuma, G. C., 2015. Alat Pendekripsi Dini Terhadap Kebocoran Gas LPG. *JURNAL TELE Volume 13 Nomor 1*, pp. 1-6.
- Wiratma, A. B., Munadi, R. & Mayasari, R., 2016. Implementasi dan Analisis Jaringan Sensor Nirkabel Sebagai Alat Pendekripsi Kebocoran Tabung Gas Elpiji Menggunakan Topologi Cluster-Tree dengan 7 Titik. *e-Proceeding of Engineering : Vol.3*, pp. 1779-1786.
- Yanto, G., 2017. Logika Fuzzy untuk Kendali Suhu Ruangan pada Air Conditioner (AC) di Ruang Dosen STMIK Indonesia Padang. *Jurnal Ilmu Fisika dan Teknologi Vol 1*, pp. 23-32.