

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Sumber daya alam yang bermanfaat bagi kehidupan manusia sangatlah banyak tersedia di bumi ini. Baik itu sumber daya alam yang dapat diperbaharui maupun sumber daya alam yang tidak diperbaharui. Ketergantungan masyarakat terhadap alam seperti minyak bumi tidak bisa dihindarkan untuk keperluan sehari-hari mulai dari memasak, hingga bahan bakar minyak untuk kendaraan transportasi dan penggunaan lainnya yang terus mengalami peningkatan, sehingga menyebabkan bahan bakar fosil ini menjadi terbatas. Perkembangan ilmu pengetahuan dan teknologi yang terjadi dengan cepat dewasa ini berhasil menemukan energi alternatif lain yang bisa menggantikan minyak bumi dalam berbagai kebutuhan, energi alternatif tersebut dinamakan *Liquefied Petroleum Gas* (LPG). Gas LPG merupakan salah satu hasil dari sumber daya alam yang tidak dapat diperbaharui. Peranan gas LPG pada saat ini sangatlah penting bagi kehidupan manusia. Konsumen LPG bervariasi, mulai dari rumah tangga, kalangan komersial (restoran, hotel, dll) hingga industri.

Semenjak pemerintah melakukan konversi minyak tanah ke kompor gas, banyak sekali kejadian meledaknya tabung gas, sering terjadi kebocoran tabung gas yang berbahaya bagi pengguna maupun masyarakat sekitar. Berita kebakaran pun sering terdengar sebagai akibat tabung gas LPG meledak. Penyebab meledaknya tabung gas ini karena kebocoran pada selang, tabung atau pada regulatornya yang tidak terpasang dengan baik. Pada saat terjadi kebocoran akan tercium gas yang menyengat, gas inilah yang nantinya akan meledak apabila ada sulutan atau percikan api, atau adanya nyala rokok. Pada intinya ledakan dapat dihindarkan apabila adanya pencegahan dini, saat gas keluar atau pada saat kebocoran gas terjadi.

Pencegahan dini atau solusinya untuk masalah di atas adalah mengendalikan kadar gas tersebut dengan membuat sistem sirkulasi udara dalam ruangan yang mengalami kebocoran gas menggunakan kipas. ketika mengalami kebocoran gas, sensor MQ-6 akan mendeteksi kadar gas yang bocor didalam ruang tersebut yang ditampilkan di display LCD (*Liquid Crystal Display*). Kecepatan kipas tergantung dengan kadar gas LPG dalam ruang yang dibaca oleh sensor MQ-6. Jadi, semakin besar kadar gas LPG yang bocor maka kecepatan

kipas akan meningkat. Sistem ini mensirkulasi udara agar ruang yang mengalami kebocoran LPG tetap dalam kondisi normal.

### 1.2 Rumusan Masalah

Mengacu pada permasalahan yang telah diuraikan pada latar belakang, maka rumusan masalah dapat ditekankan pada *point* berikut:

1. Bagaimana merancang algoritma kontroler proporsional dengan mikrokontroler sebagai sistem kendali otomatis pada ruang yang mengalami kebocoran LPG sesuai dengan *setpoint*?
2. Bagaimana pengaplikasian algoritma kontroler proporsional di miniatur pengendalian kadar gas?

### 1.3 Batasan Masalah

1. Mikrokontroler yang digunakan adalah arduino mega 2560 R3.
2. Kontroler yang digunakan adalah kontroler proporsional.
3. Sensor yang digunakan adalah sensor gas MQ-6.
4. Menggunakan 1 buah kipas DC 12V.
5. Kinerja *driver* dan elektronika tidak di bahas secara mendalam.
6. Metode yang digunakan adalah *hand tuning*.
7. Sistem dibuat dalam suatu modul berupa miniatur ruang dapur tertutup dengan ukuran panjang 35 cm, lebar 30 cm, dan tinggi 25 cm.
8. Pembahasan ditekankan pada proses pengendalian kadar gas terhadap kebocoran LPG dalam ruang sesuai dengan *setpoint*.
9. Menggunakan LCD 16x2 dalam sistem untuk menampilkan kadar gas dalam ruang.

### 1.4 Tujuan

Terwujudnya suatu sistem pengendalian kadar gas agar kadar gas dalam ruang yang mengalami kebocoran LPG sesuai dengan *setpoint* yang di inginkan sehingga walaupun ada kebocoran tetap aman.

## 1.5 Sistematika Pembahasan

Skripsi ini terdiri dari lima bab dengan sistematika pembahasan sebagai berikut:

### **BAB I       Pendahuluan**

Membahas latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika pembahasan.

### **BAB II       Tinjauan Pustaka**

Membahas teori-teori yang mendukung dalam perancangan dan pembuatan alat.

### **BAB III      Metodologi Penelitian**

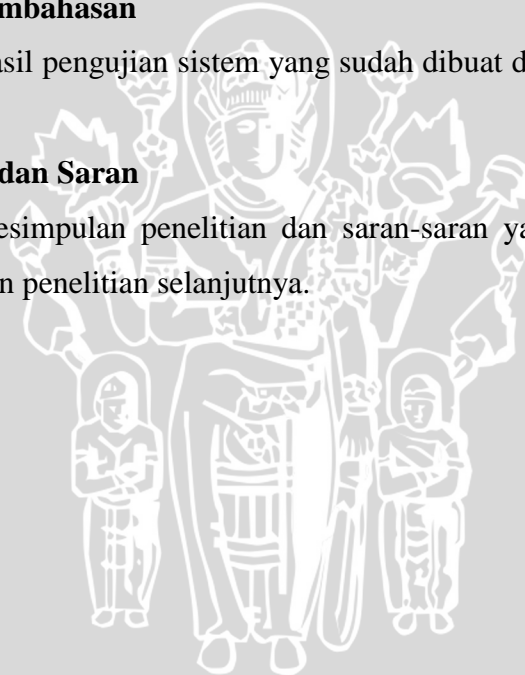
Membahas perancangan alat yang merupakan spesifikasi, perencanaan blok diagram, prinsip kerja dan pembuatan alat, serta memuat hasil pengujian terhadap alat yang telah dibuat.

### **BAB IV      Hasil dan Pembahasan**

Membahas hasil pengujian sistem yang sudah dibuat dan analisis hasil yang diperoleh.

### **BAB V       Kesimpulan dan Saran**

Membahas kesimpulan penelitian dan saran-saran yang diperlukan untuk pengembangan penelitian selanjutnya.





## BAB II

### TINJAUAN PUSTAKA

Untuk memudahkan dalam memahami cara kerja rangkaian maupun dasar-dasar perencanaan dari sistem yang di buat, maka perlu adanya penjelasan dan uraian mengenai teori penunjang yang digunakan dalam penulisan skripsi ini.

Teori penunjang yang akan di bahas dalam bab ini adalah:

1. Arduino mega 2560 R3
2. Kontroler *Proportional* (P).
3. Metode Kontroler *hand tuning*.
4. Sensor Pendeteksi gas MQ-6.
5. *Brushless* motor DC 12V.
6. *Driver* motor DC L298N *Dual H-Bridge*

#### 2.1 Mikrokontroler Arduino Mega 2560 R3

Arduino mega 2560 adalah papan mikrokontroler Atmega 2560 memiliki 54 digital pin *input* atau *output* (di mana 15 pin dapat digunakan sebagai *output* PWM atau *Pulse Width Modulation*), 16 *analog input*, 4 UART (*Universal Asynchronous Receiver/Transmitter*), osilator kristal 16 MHz, 8 koneksi USB, jack listrik, *header ICSP (In-Circuit Serial Programing)*, dan tombol reset. Semuanya diperlukan untuk mendukung mikrokontroler, hanya menghubungkannya ke komputer dengan kabel USB atau *power* dengan adaptor AC (*Alternating Current*) ke DC (*Direct Current*) atau baterai.



Gambar 2.1 Arduino Mega 2560 R3

Sumber: <http://arduino.cc/en/Main/ArduinoBoardMega2560>

Arduino Mega 2560 (Gambar 2.1) berbeda dari semua papan sebelumnya dalam hal itu tidak menggunakan FTDI *chip driver* USB-to-serial. Sebaliknya, fitur Atmega 16U2 (Atmega 8U2 dalam *board* revisi 1 dan revisi 2) diprogram sebagai *converter* USB-to-serial. Revisi 2 dari Arduino Mega 2560 memiliki resistor menarik garis 8U2 HWB ke *ground*, sehingga lebih mudah untuk dimasukkan ke dalam mode DFU (*Device Firmware Update*). Revisi 3 dari arduino mega 2560 memiliki fitur-fitur baru sebagai berikut:

- a. 1 pinout tambah SDA (*Serial Data*) dan SCL (*Serial Clock*) pin yang dekat dengan pin AREF (*ADC Reference*) dan dua pin baru lainnya ditempatkan dekat dengan pin reset.
- b. Sirkuit reset lebih kuat.
- c. Atmega 16U2 menggantikan 8U2.

## 2.2 Kontroler

Sistem pengontrolan yang di rancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengontrolan adalah harus stabil. Di samping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relative, yakni tolak ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika dikena gangguan (Ogata K.,1997). Selain itu analisis juga di lakukan bagaimana kecepatan sistem dalam merespon *input* dan meredam adanya lonjakan (*over shoot*).

Suatu sistem dikatakan stabil, jika di beri gangguan maka sistem tersebut akan kembali ke dalam *steady state* yaitu *output* berada dalam keadaan tetap seperti tidak ada gangguan. Sistem di katakan tidak stabil, jika *output* berosilasi terus menerus ketika dikenai suatu gangguan. Karena suatu sistem pengendalian biasanya melibatkan penyimpanan energi maka *output* sistem ketika diberi suatu *input*, tidak dapat mengikuti *input* secara serentak, tapi menunjukkan respon *transien* berupa suatu osilasi teredam sebelum mencapai *steady state*. Dalam sistem pengontrolan terdapat 2 loop.

1. Pengontrolan dengan loop terbuka

Sistem kontrol loop terbuka adalah sistem kontrol yang keluarannya tidak berpengaruh pada aksi pengontrolannya. Jadi pada sistem kontrol loop terbuka, keluaran tidak diukur atau diumpun balikan untuk dibandingkan dengan masukan.

2. Pengontrolan dengan loop tertutup

Sistem kontrol loop tertutup adalah sistem kontrol yang keluarannya mempunyai pengaruh langsung pada aksi pengontrolannya. Di sebut juga sistem yang menggunakan umpan balik untuk memperkecil kesalahan sistem.

### 2.2.1 Kontroler Proporsional

Kontroler proporsional memiliki keluaran yang sebanding dengan sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Untuk kontroler dengan aksi kontrol proporsional, hubungan antara keluaran kontroler  $m(t)$  dan sinyal kesalahan penggerak  $e(t)$  adalah:

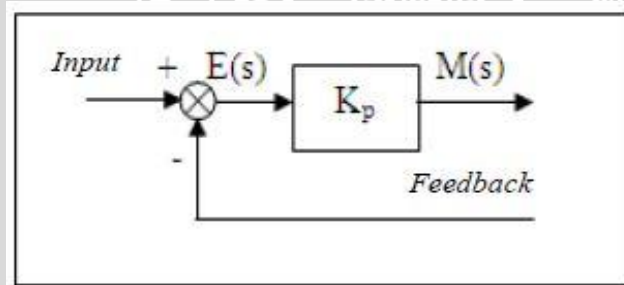
$$m(t) = K_p \cdot e(t) \dots \dots \dots (1-1)$$

atau, dalam besaran transformasi Laplace,

$$\frac{M(s)}{E(s)} = K_p \dots \dots \dots (1-2)$$

Di mana  $K_p$  adalah kepekaan proporsional atau penguatan.

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggeraknya, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata K.,1997). Diagram blok kontroler proporsional ditunjukkan dalam Gambar 2.2.



**Gambar 2.2** Diagram Blok Kontroler Proporsional

Sumber: Ogata K., 1997

### 2.3 Brushless Motor DC

Motor DC adalah jenis motor listrik yang bekerja menggunakan sumber tegangan DC. Motor DC atau motor arus searah sebagaimana namanya, menggunakan arus langsung dan tidak langsung/*direct-unidirectional*. Motor DC digunakan pada penggunaan khusus di mana diperlukan penyalan torque yang tinggi atau percepatan yang tetap untuk kisaran kecepatan yang luas. Pada sistem ini aktuator yang di gunakan adalah *brushless* motor DC. Fungsi dari motor DC ini adalah untuk mengeluarkan gas yang ada di dalam ruang keluar

ruangan tersebut. BLDC motor adalah suatu jenis motor-sinkron. Artinya medan magnet yang dihasilkan oleh stator dan medan magnet yang dihasilkan oleh Rotor berputar di frekwensi yang sama. BLDC motor tidak mengalami Slip, tidak seperti yang terjadi pada motor induksi biasa. Untuk gambar BLDC akan ditunjukkan dalam gambar 2.3.



**Gambar 2.3** Brushless Motor 12V DC.

Sumber: <http://id.aliexpress.com/cheap/cheap-cooling-fan-motor-12v.html>

#### 2.4 Sensor Gas MQ-6

MQ-6 Sensor gas yang digunakan untuk mendeteksi LPG, Iso-butane, Propane dengan sensitivitas yang tinggi. Sensor gas MQ-6 (Gambar 2.4) ini mempunyai sensitivitas yang kecil terhadap zat alcohol dan asap rokok. Sensor gas MQ-6 merupakan sensor yang mempunyai respon cepat terhadap LPG (*Liquid Petroleum Gas*), stabil dan tahan lama, serta dapat digunakan dalam rangkaian drive yang sederhana. Sensor Gas MQ 6 biasa digunakan didalam perlengkapan mendeteksi kebocoran gas dalam kegiatan rumah tangga dan industri, yang cocok untuk mendeteksi LPG, iso-butane, propane, LNG, serta menghindari gangguan dari pendeteksian zat Alkohol, asap masakan, dan rokok untuk mengurangi kesalahan pendeteksian.



**Gambar 2.4** Sensor MQ-6

Sumber: <http://baskarapunya.blogspot.sg/2013/07/mq-6-lpgiso-butane-propane-sensor.html>

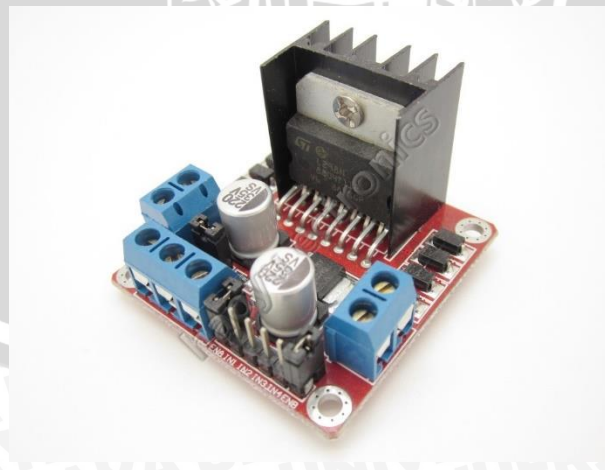


## 2.5 *Driver Motor DC L298N*

Motor DC tidak dapat dikontrol secara langsung oleh mikrokontroler, karena kebutuhan daya listrik yang cukup besar sedangkan daya keluaran pada mikrokontroler sangat kecil. Untuk dapat melakukan pengontrolan motor DC dibutuhkan suatu *driver* yang mampu memperbesar arus dan tegangan sesuai dengan kebutuhan motor DC. Ada beberapa jenis *driver* motor yang dapat digunakan untuk mengontrol kecepatan motor, *Driver* motor L298N (Gambar 2.5) merupakan *driver* motor yang paling populer digunakan untuk mengontrol kecepatan dan arah pergerakan motor terutama pada robot *line foller / line tracer*.

Kelebihan dari *driver* motor L298N ini adalah cukup presisi dalam mengontrol motor. Selain itu, kelebihan *driver* motor L298N adalah mudah untuk dikontrol. Untuk mengontrol *driver* L298N ini dibutuhkan 6 buah pin mikrokontroler. Dua buah untuk pin *Enable* (satu buah untuk motor pertama dan satu buah yang lain untuk motor kedua. Karena *driver* L298N ini dapat mengontrol dua buah motor DC). 4 buah untuk mengatur kecepatan motor motor tersebut. Rangkaian *driver* motor DC dengan IC L298 dapat digunakan untuk mengendalikan motor DC dengan arus maksimum hingga 4A. IC L298 adalah *driver* motor DC *H-Bridge* dengan 2 unit *driver* didalam 1 chip IC.

Skematik rangkaian *driver* motor L298N harus ditambahkan beberapa komponen lagi agar dapat bekerja. Yang pertama berupa rangkaian regulator yang berada dibagian atas skematik. dan yang kedua adalah rangkaian pendukung *driver* motor yang berupa beberapa dioda.

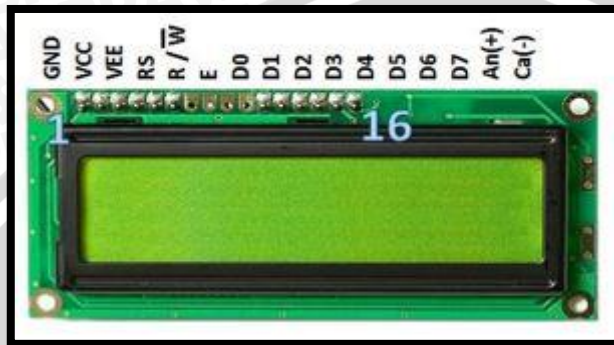


**Gambar 2.5** L298 Driver *H-Bridge Drive PWM*

Sumber : [http://www.haoyuelectronics.com/Attachment/L298N\\_module/2.jpg](http://www.haoyuelectronics.com/Attachment/L298N_module/2.jpg)

## 2.6 LCD (*Liquid Cristal Display*)

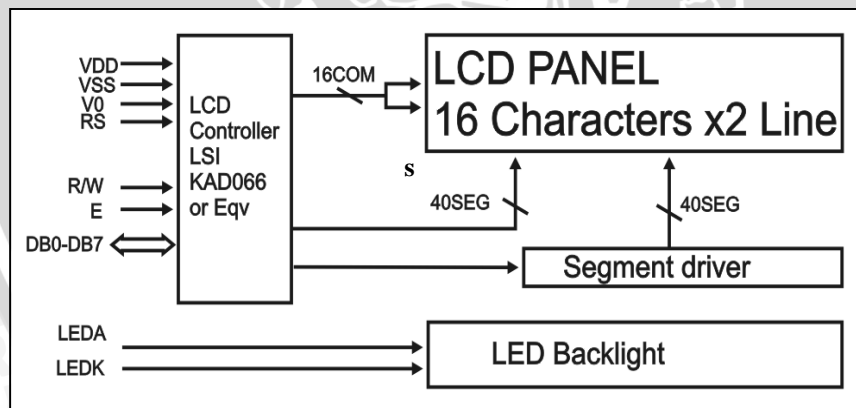
LCD (*Liquid Cristal Display*) adalah salah satu jenis display elektronik yang di buat dengan teknologi CMOS *logic* yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekitarnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*. LCD (Gambar 2.6) berfungsi sebagai penampil data baik dalam bentuk karakter, huruf ataupun grafik.



**Gambar 2.6** LCD (*liquid Cristal Display*) 16x2

Sumber: <http://elektronika-dasar.web.id/lcd-liquid-cristal-display/>

Masukan modul LCD ini berupa bus data dan 3 sinyal kontrol yaitu RS, R/W dan E. Pengendali *dot* matriks LCD dilakukan secara internal oleh kontroler yang sudah terpasang di dalam modul LCD. Diagram blok LCD tipe QC1602A ditunjukkan dalam Gambar 2.7.



**Gambar 2.7.** Blok Diagram *liquid crystal display* (LCD)

Sumber: *datasheet* QC1602A

LCD tipe QC1602A memiliki 16 pin koneksi antarmuka dimana setiap pin memiliki fungsi tertentu. Fungsi masing-masing pin ditunjukkan dalam Tabel 2.1.

pin	Nama pin	Level	Fungsi
1	VSS	0V	Power ground
2	VDD	+5V	Power supply for logic
3	V0	-	Contrast adjust
4	RS	High/Low	High: data Low: commmand
5	R/W	High/Low	High: read Low: write
6	E	High. High □ Low	Enable signal
7	DB0	High/Low	Data bus bit 0
8	DB1	High/Low	Data bus bit 1
9	DB2	High/Low	Data bus bit 2
10	DB3	High/Low	Data bus bit 3
11	DB4	High/Low	Data bus bit 4
12	DB5	High/Low	Data bus bit 5
13	DB6	High/Low	Data bus bit 6
14	DB7	High/Low	Data bus bit 7
15	LEDA	+5V	Power supply for LED Backlight
16	LEDK	0V	Power supply for LED Backlight

**Tabel 2.1** LCD (liquid Cristal Display) 16x2

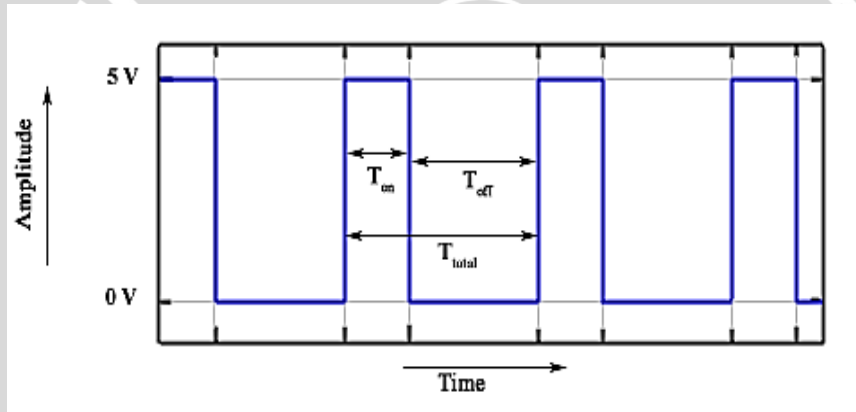
Sumber : *datasheet* QC160

## 2.7 Metode Hand Tuning

Kontroler P dapat di tuning dalam beberapa cara, antara lain *ziegler-nichols tuning*, *loop shaping*, metode analitis, optimasi, *pole placement*, *autotuning* dan *hand tuning* (Smith,1997; Astrom & Hagglund,2004). Dengan menggunakan metode *hand tuning*, nilai parameter P perlu diubah-ubah secara *trial* agar respon yang diperoleh sesuai dengan yang diinginkan.

## 2.8 Pulse Width Modulation

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. Persamaan untuk perhitungan *duty cycle* ditunjukkan pada Persamaan 2.5 dengan  $T_{on}$  adalah periode logika tinggi, dan  $T$  adalah periode keseluruhan. Sinyal PWM secara umum dapat dilihat dalam Gambar 2.8.



Gambar 2.8 Sinyal PWM secara umum

Sumber: [www.8051projects.net](http://www.8051projects.net)

$$\text{Duty Cycle} = \frac{T_{on}}{T_{total}} \times 100\% \quad (2.5)$$

Sedangkan frekuensinya dapat ditentukan dengan Persamaan 2.6

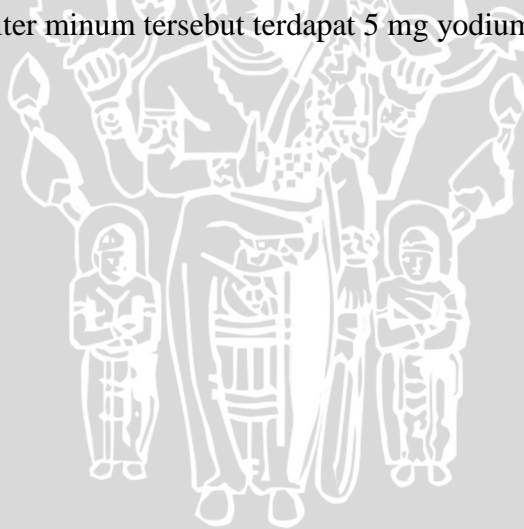
$$f_{OCn} = \frac{f_{clk} / 0}{N \cdot 256} \quad (2.6)$$

Timer atau counter yang digunakan pada PWM ini yaitu timer atau counter 0 (8 bit) dengan metode *fast PWM* dan *prescaler factor* (N) yaitu 256.

## 2.9 Satuan ppm (*Part Per Million*)

PPM atau “Part per Million” jika dibahasa Indonesiakan akan menjadi “Bagian per Sejuta Bagian” adalah satuan konsentrasi yang sering dipergunakan dalam di cabang Kimia Analisa. ‘ppm’ adalah ‘part per million’ bisa dalam volume (ppm volume) atau massa/berat (ppm mass/weight), yaitu satuan kadar atau konsentrasi. Kalau tidak disebut apa-apa (hanya ‘ppm’), umumnya diartikan sebagai ppm volume. Satuan ini sering digunakan untuk menunjukkan kandungan suatu senyawa dalam suatu larutan misalnya kandungan garam dalam air laut, kandungan polutan dalam sungai, atau biasanya kandungan yodium dalam garam juga dinyatakan dalam ppm.

Seperti halnya namanya yaitu ppm, maka konsentrasinya merupakan perbandingan antara berapa bagian senyawa dalam satu juta bagian suatu sistem. Sama halnya dengan “prosentase” yang menunjukkan bagian per seratus. Contoh lainya bisa seperti, Kandungan gas metane dalam bak valve 2000 ppm artinya dalam setiap Kg udara dalam bak valve terdapat 2000 ppm gas metane, bisa juga kandungan karbon dalam baja adalah 5 ppm artinya dalam 1 Kg baja terdapat 5 mg karbon, atau air minum mengandung yodium sebesar 15 ppm, bisa diartikan bahwa setiap liter minum tersebut terdapat 5 mg yodium.





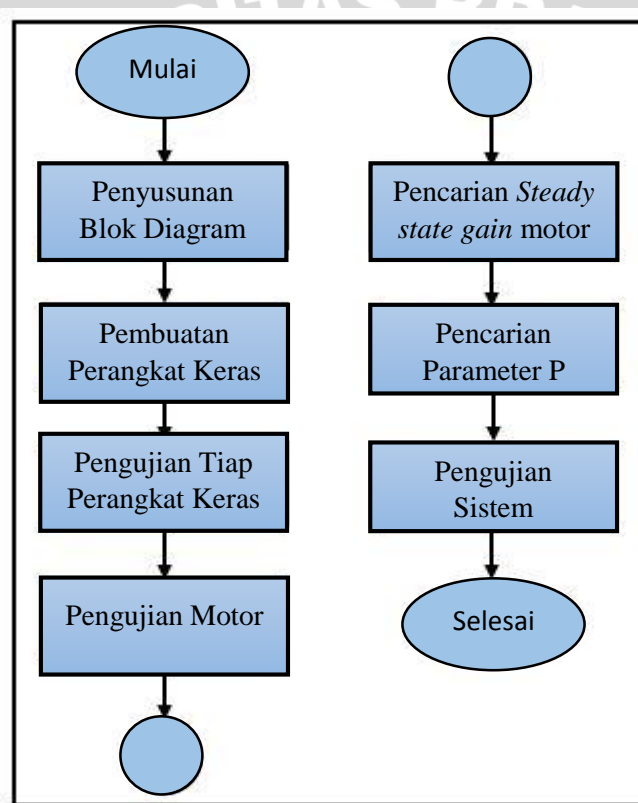
### BAB III

## METODOLOGI PENELITIAN

Dalam skripsi kali ini kajian yang digunakan adalah kajian pengendalian kadar gas pada miniatur ruang dapur sebagai *plant*-nya guna mengetahui respon sistem menggunakan kontroler P.

### 3.1 Kerangka Penelitian

Berikut kerangka penelitian pada skripsi ini terdapat dalam Gambar 3.1.



Gambar 3.1 Kerangka Penelitian

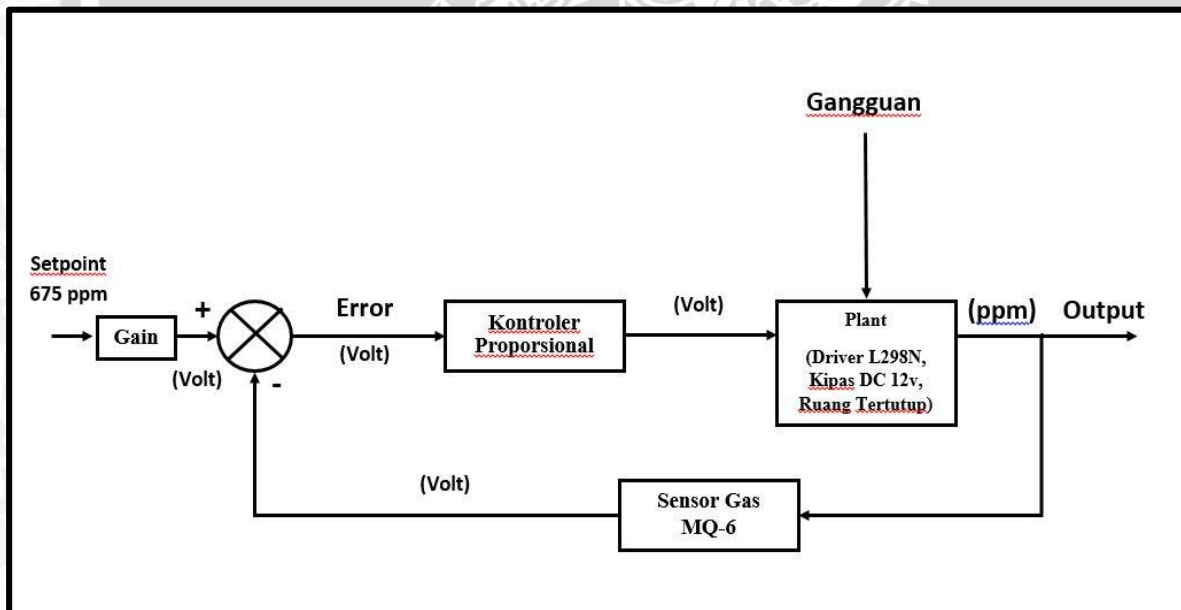
Adapun penjelasan dari *flowchart* sebagai berikut :

1. Penyusunan blok diagram sistem bertujuan untuk memahami struktur sistem pada setiap bloknnya sehingga akan mempermudah dalam analisis secara keseluruhan.
2. Pembuatan perangkat keras bertujuan mengetahui kebutuhan perangkat keras sistem yang dibutuhkan serta mengetahui cara kerja sistem dari sisi perangkat keras.

3. Pengujian setiap perangkat keras bertujuan untuk mengetahui karakteristik dan fungsi alihnya.
4. Pengujian motor bertujuan mengetahui hubungan tegangan masukan dengan putaran motor.
5. Pencarian *Steady state gain* motor yang di mana untuk mengetahui kemampuan kipas mengeluarkan gas LPG yang bocor.
6. Mencari parameter Kontroler P menggunakan metode *Hand Tuning* agar sistem dapat bekerja dengan baik yaitu keluaran diharapkan sama dengan masukan yang diinginkan.
7. Pengujian sistem bertujuan untuk mengetahui apakah sistem dapat bekerja dengan baik atau tidak.

### 3.2 Penyusunan Blok Diagram Sistem

Pada perencanaan alat diperlukan perencanaan blok diagram sistem yang dapat menjelaskan sistem secara garis besar dan diharapkan alat dapat bekerja sesuai rencana, berikut blok diagram sistem (Gambar 3.2).



Gambar 3.2 Diagram Blok Sistem

*Setpoint* pada blok diagram berupa nilai kadar gas dalam plant yang diatur oleh user. Nilai *setpoint* sebesar 675 ppm akan disimpan pada mikrokontroler. Selain *setpoint* terdapat juga nilai *error*, nilai *error* ini merupakan deviasi antara pembacaan aktual dari sensor gas MQ-6 dan nilai *setpoint*. Kedua nilai yang didapatkan tersebut akan dikalkulasi dan dioleh oleh mikrokontroler. Nilai yang telah dikalkulasi tadi akan berfungsi sebagai manipulated



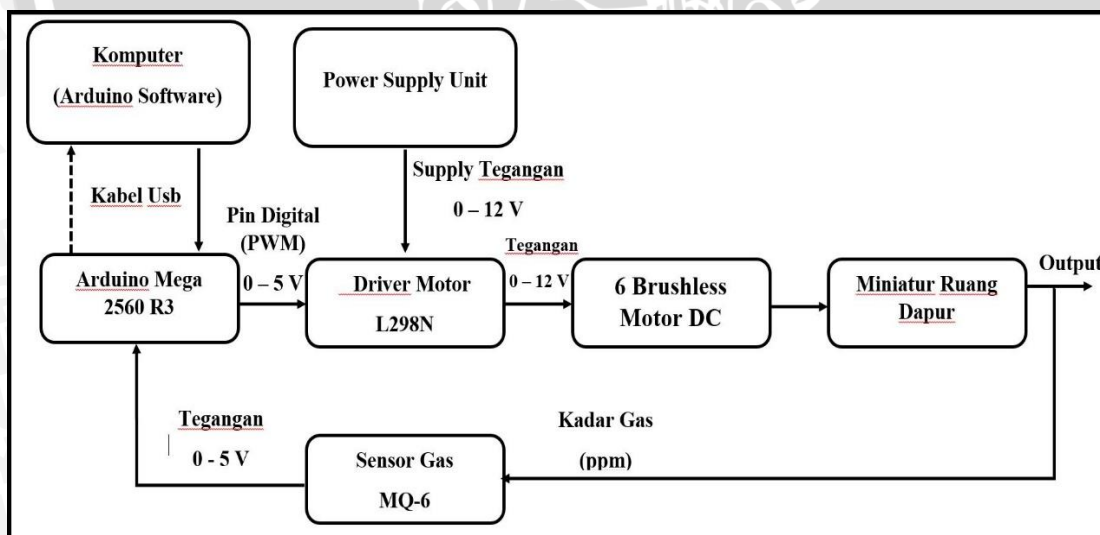
variable sinyal kontrol yang digunakan sebagai input *driver* motor. *Driver* motor ini akan menggerakkan motor DC tanpa sikat (*brushless* motor DC) sesuai dengan sinyal kontrol keluaran mikrokontroler. Kondisi kecepatan sekarang berdasarkan pembacaan kadar gas dalam plant oleh MQ-6 dan perbedaan pembacaan kadar gas akan menjadi masukan *error* lagi bagi kontroler.

### 3.3 Pembuatan Perangkat Keras

Pembuatan perangkat keras merupakan langkah awal sebelum terbentuknya suatu sistem, hal ini bertujuan supaya *output* kecepatan motor sesuai dengan *input* yang diinginkan dan dapat bekerja dengan baik sesuai deskripsi awal yang telah direncanakan.

Pembuatan perangkat keras meliputi:

1. Skema pembuatan perangkat keras (Gambar 3.3)
2. Penentuan modul elektronik yang digunakan, meliputi :
  - *Power supply* Unit
  - Mikrokontroler Arduino Mega 2560 R3
  - *Driver Motor* L298N
  - Brushless Motor Dc
  - Sensor Gas MQ-6



Gambar 3.3 Skema Pembuatan Perangkat Keras

### 3.4 Spesifikasi Alat

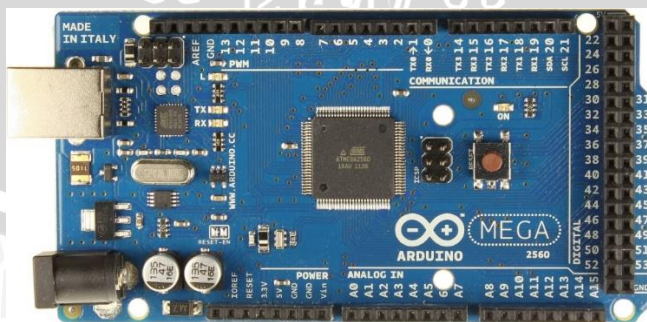
Adapun spesifikasi alat dari skema pembuatan perangkat keras diatas adalah sebagai berikut:

1. Komputer atau PC (*Personal Computer*) yang sudah terinstall *software* arduino dan MATLAB 2012a.
2. *Power supply Unit* (PSU) yang ditunjukkan dalam Gambar 3.4 berfungsi sebagai sumber tegangan atau catu daya pada *driver* motor di mana PSU yang digunakan ini mempunyai tegangan keluaran 0-12V.



**Gambar 3.4** Power supply Unit

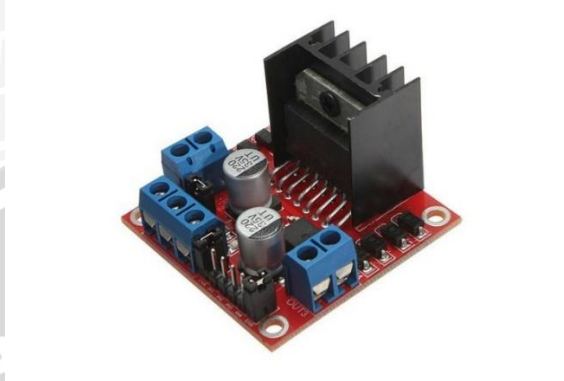
3. Perangkat kontroler yang digunakan adalah arduino mega 2560 R3. Mikrokontroler Arduino Mega 2560 dalam Gambar 3.5 berfungsi untuk menyimpan semua algoritma kontroler sistem, pengendalian *driver* motor, menerima data yang dikirimkan oleh sensor gas MQ-6 serta mengolah datanya. Mikrokontroler menggunakan catu tegangan 5 V yang didapat dari laptop atau PC



**Gambar 3.5** Arduino Mega 2560 R3

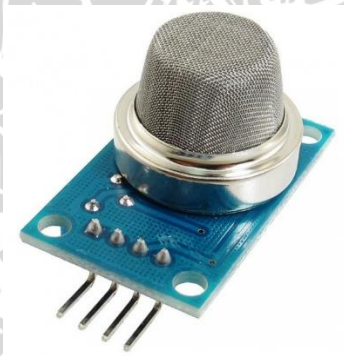
4. *Driver* motor yang digunakan L298N *Dual H-Bridge* yang ditunjukkan dalam Gambar 3.6. *Driver* Motor L298N ini berfungsi untuk mengalikan tegangan antara tegangan masukan dari mikrokontroler Arduino dengan sebuah nilai

gain atau penguatan tertentu sehingga tegangan keluaran dari *driver* motor digunakan untuk mencatu motor. *Driver* motor ini dicatu dengan tegangan yang didapat dari *power supply unit* (PSU) dengan tegangan maksimal 12v dan arus maksimalnya sebesar 3.2 A.



**Gambar 3.6 Driver Motor Dual H-Bridge L298N**

5. Sensor Gas yang digunakan adalah sensor gas MQ-6 (Gambar 3.7). Sensor gas MQ-6 membutuhkan daya sebesar 5 V untuk menjalankannya yang di mana diperoleh dari mikrokontroler. Keluaran dari MQ-6 dikirimkan ke mikrokontroler untuk diproses selanjutnya oleh arduino mega.



**Gambar 3.7 Sensor Gas MQ-6**

6. Brushless Motor DC (Gambar 3.8) yang digunakan memiliki tegangan maksimal sebesar 12 V, dan memiliki arus maksimal sebesar 0.15 A. Motor DC yang digunakan dalam percobaan ada sebanyak 1 buah sehingga arus maksimal dari seluruh motor sebesar 0.15 A. *Brushless* Motor DC ini yang berfungsi sebagai aktuator untuk mengeluarkan gas dari miniatur ruang dapur.



**Gambar 3.8** *Brushless Motor DC 12V*

7. Plant berupa miniatur ruang dapur yang digunakan terbuat dari bahan *acrylic* dengan ukuran panjang 35 cm, lebar 30 cm dan tinggi 25 cm.

### 3.5 Prinsip Kerja Alat

Cara kerja alat ini yaitu sebagai berikut:

1. *Power supply unit* (PSU) akan dicatu tegangan AC 220 VAC.
2. *Power supply unit* (PSU) akan mengkonversi tegangan AC 220 VAC menjadi 12 VDC sehingga dapat dipakai untuk suplai rangkaian *driver* motor.
3. Catu daya Arduino Mega 2560 sebesar 5 VDC didapatkan dengan menghubungkan Arduino dengan laptop atau PC menggunakan kabel USB.
4. Keluaran dari sensor gas MQ-6 akan berupa nilai ADC dari 0 sampai dengan 1023 yang di mana sebagai masukan analog pada pin A0 di Arduino Mega R3 yang kemudian akan dikonversi menjadi dalam bentuk ppm (*part per million*) dan tegangan.
5. Sinyal kontrol dari arduino mega masuk ke *driver* motor L298N (*Dual H-Bridge*). *Driver* motor berfungsi sebagai penguatan sinyal yang dihasilkan oleh mikrokontroler Arduino Mega 0 – 5 V menjadi 0 – 12 V.
6. Mencari parameter kontroler P sebagai kontroler pengendalian kadar gas.
7. Mengimplemantasikan hasil desain perancangan sistem.

### 3.6 Pin Arduino Mega

Arduino Mega 2560 adalah papan pengembangan mikrokontroler yang berbasis Arduino dengan menggunakan chip ATmega2560. *Board* ini memiliki pin I/O yang cukup banyak, sejumlah 54 buah digital I/O pin (15 pin diantaranya adalah PWM), 16 pin analog *input*, 4 pin UART (*serial port hardware*). Arduino Mega 2560 dilengkapi dengan sebuah oscillator 16 Mhz, sebuah port USB, *power jack* DC, *ICSP header*, dan tombol reset. Fungsi setiap pin ditunjukkan dalam Tabel 3.1.

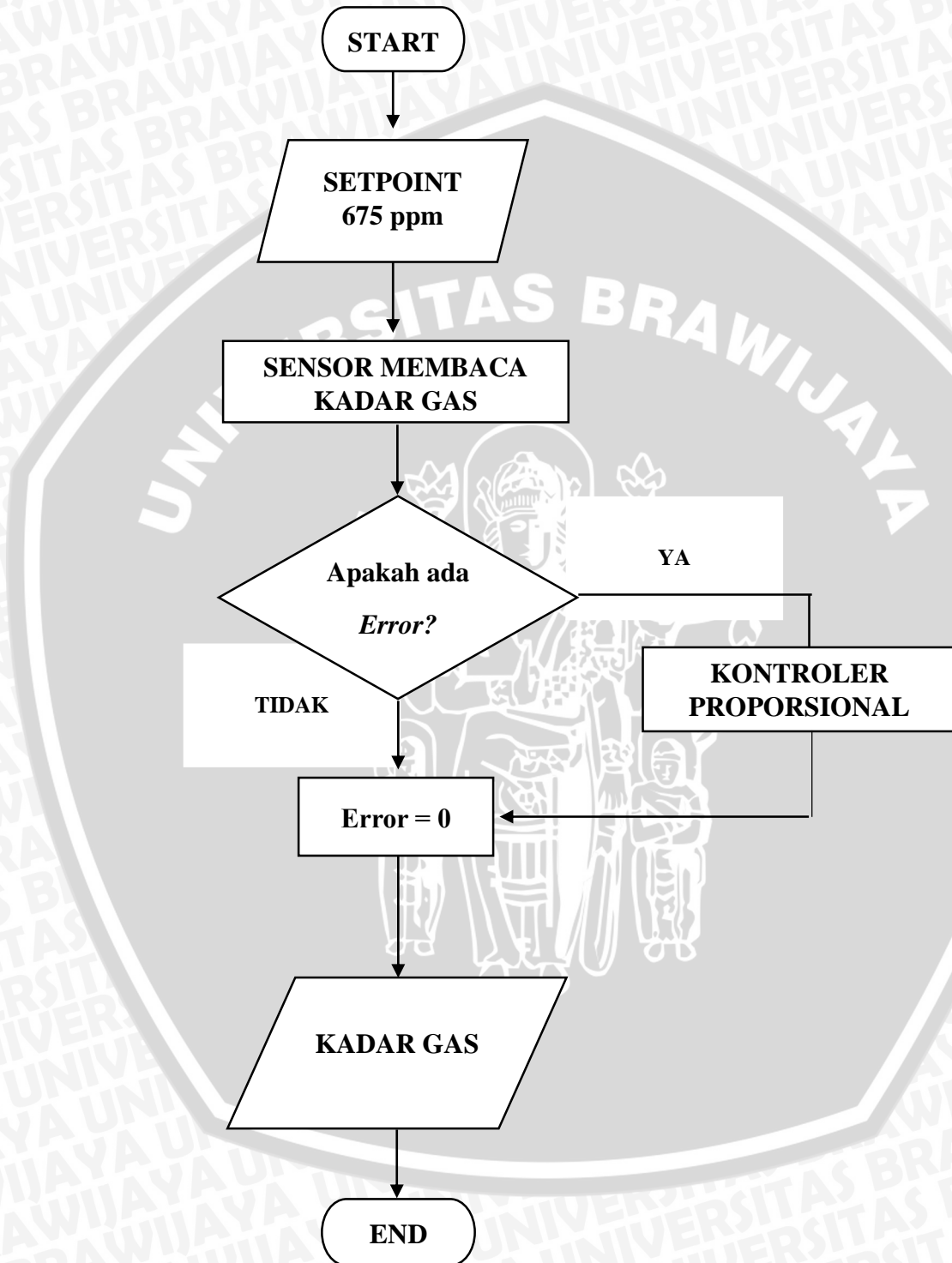
NO	Pin	Fungsi
1	5 Volt	Catu daya LCD, sensor MQ-6 dan <i>driver</i> motor L298N <i>Dual H-Bridge</i>
2	SCL	Pin <i>communication</i> untuk LCD I2C
3	SDA	Pin <i>communication</i> untuk LCD I2C
4	GND	Sebagai ground untuk LCD, MQ-6 dan <i>driver</i> motor L298N <i>Dual H-Bridge</i>
5	A0	Analog pin untuk sensor MQ-6
6	PWM 13,12,11,10	PWM pin untuk <i>driver</i> motor 1
7	PWM 9,8,7,6	PWM pin untuk <i>driver</i> motor 2
8	PWM 5,4,3,2	PWM pin untuk <i>driver</i> motor 3
9	Digital Pin 22,23,24,25	Digital pin untuk <i>driver</i> motor 1
10	Digital Pin 26,27,28,29	Digital pin untuk <i>driver</i> motor 2
11	Digital Pin 30,31,32,33	Digital pin untuk <i>driver</i> motor 3

Tabel 3.1 Fungsi Pin Arduino Mega

### 3.7 Flowchart Perancangan Perangkat Lunak

Flowchart sistem/diagram alir sistem secara keseluruhan dapat dilihat pada Gambar

3.9.



Gambar 3.9 Flowchart sistem.

### 3.8 Pengujian *Driver Motor L298N Dual H-Bridge* dengan beban *Brushless Motor DC 12 V*

#### a. Tujuan

Mengetahui kinerja dan respon rangkaian *driver* motor L298N dengan membandingkan output tegangan efektif *driver* terhadap masukan sinyal PWM yang diberikan oleh Arduino mega 2560 dengan dibebani *brushless Motor DC 12 V*.

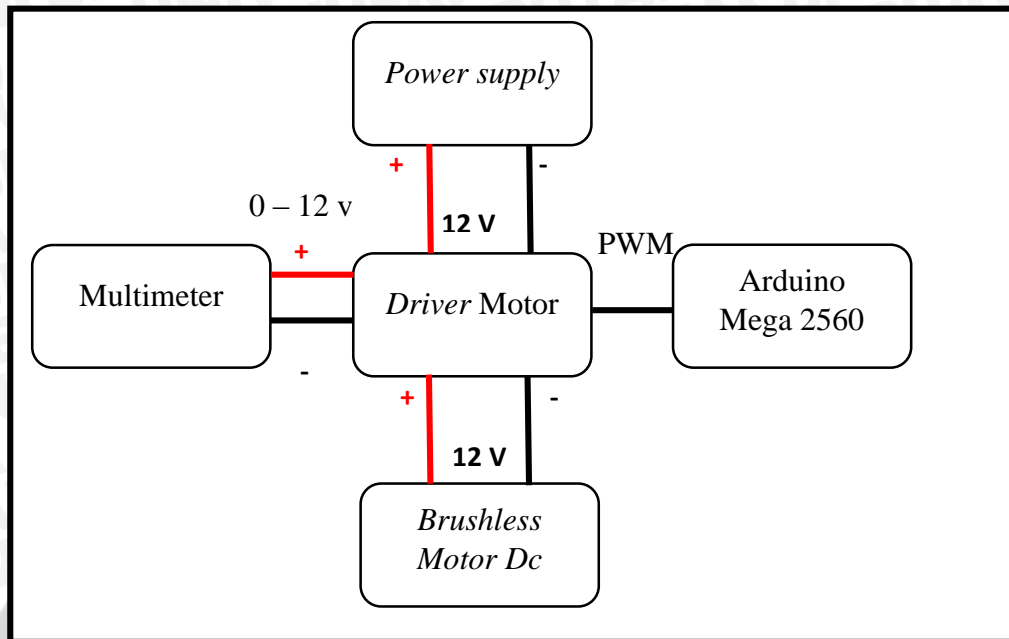
#### b. Peralatan yang digunakan

1. *Power supply*.
2. *Driver L298N*.
3. Arduino Mega 2560 R3.
4. Multimeter.
5. Laptop.
6. Kabel penghubung (*Jumper*).
7. *Brushless Motor DC*.

#### c. Prosedur pengujian

1. Menuliskan listing program pada software arduino untuk mengubah *duty cycle*.
2. Hubungkan *output* tegangan *Power supply* sebagai input tegangan pada *driver* motor L298N *Dual H-Bridge* seperti dalam Gambar 3.10.
3. Hubungkan *input* tegangan *driver* motor L298N *Dual H-Bridge* dengan pin keluaran PWM di Arduino Mega R3 seperti dalam Gambar 3.10.
4. Hubungkan *output* tegangan *driver* motor L298N *Dual H-Bridge* dengan multimeter seperti dalam Gambar 3.10.
5. Hubungkan *output* tegangan *driver* motor L298N *Dual H-Bridge* dengan *Brushless motor DC*.
6. Atur *duty cycle* sinyal PWM pada Arduino Mega 2560 R3 dengan nilai 0%-100%.
7. Amati dan catat hasil pembacaan multimeter di setiap kenaikan 5% *duty cycle*.

Perancangan sistem untuk pengujian *Driver L298N Dual H-Bridge* dengan beban *Brushless Motor DC* ditunjukkan dalam gambar 3.10.



**Gambar 3.10** Perancangan *Driver L298N Dual H-Bridge* dengan Beban *Brushless Motor DC*.

#### d. Hasil Pengujian

Data pengujian *Driver Motor L298N Dual H-Bridge* dengan beban ditunjukkan dalam Tabel 3.2.

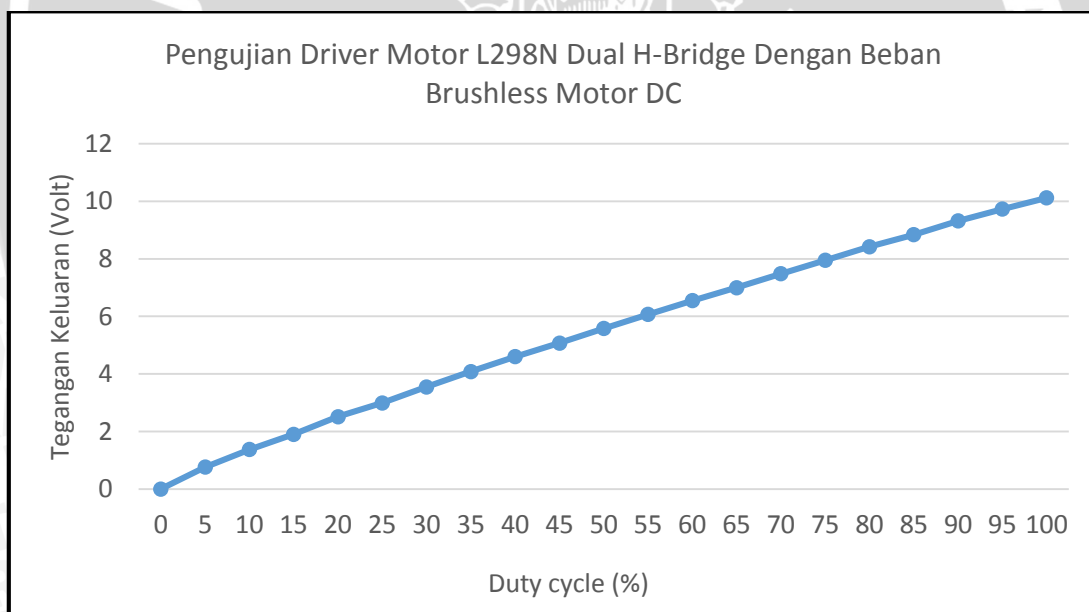
no	Duty cycle (%)	PWM	Vout (Volt)
1	0	0	0
2	5	12.75	0.76
3	10	25.5	1.37
4	15	38.25	1.9
5	20	61	2.51
6	25	63.75	2.99
7	30	76.5	3.55
8	35	89.25	4.08
9	40	102	4.6
10	45	114.75	5.07
11	50	127.5	5.58
12	55	140.25	6.07
Lanjutan			



Lanjutan			
13	60	153	6.55
14	65	165.75	7
15	70	178.5	7.48
16	75	191.25	7.95
17	80	204	8.42
18	85	216.75	8.84
19	90	229.5	9.32
20	95	242.25	9.73
21	100	255	10.12

**Tabel 3.2** Hasil pengujian *driver* motor L298N Dual H-Bridge dengan Menggunakan Beban Brushless Motor DC 12V pada Tegangan Keluaran (Volt) terhadap *Duty cycle* (%).

Berdasarkan Tabel 3.2 akan didapatkan kurva antara *output* tegangan *driver* (Volt) terhadap masukan *duty cycle* (%) sinyal PWM, seperti yang ditunjukkan dalam Gambar 3.11.



**Gambar 3.11** Grafik Perubahan Tegangan *Output Driver* L298N Dual H-Bridge terhadap *Input Duty cycle* dengan Beban Brushless Motor DC.

Sesuai dengan hasil grafik dalam Gambar 3.11 dapat disimpulkan bahwa semakin besar nilai *duty cycle* maka semakin besar pula nilai tegangan keluaran dari *driver*, tetapi tegangan keluaran *driver* dipengaruhi oleh beban motor DC sehingga nilai tegangan keluarannya tidak sebesar nilai tegangan input *driver* yaitu mencapai 12 V. Kesimpulan dari pengujian diatas didapat *gain* atau fungsi alih *driver* motor sebesar:

$$m = \frac{V_{out_{13}} - V_{out_9}}{VDuty_{cycle_{13}} - VDuty_{cycle_9}} = \frac{6.55 - 4.6}{60 - 40} = 0.0975$$

### 3.9 Pengujian Driver Motor L298N Dual H-Bridge

#### a. Tujuan

Mengetahui kinerja dan respon rangkaian *driver* motor L298N Dual H-Bridge dengan membandingkan output tegangan efektif *driver* terhadap masukan sinyal PWM yang diberikan oleh Arduino mega 2560.

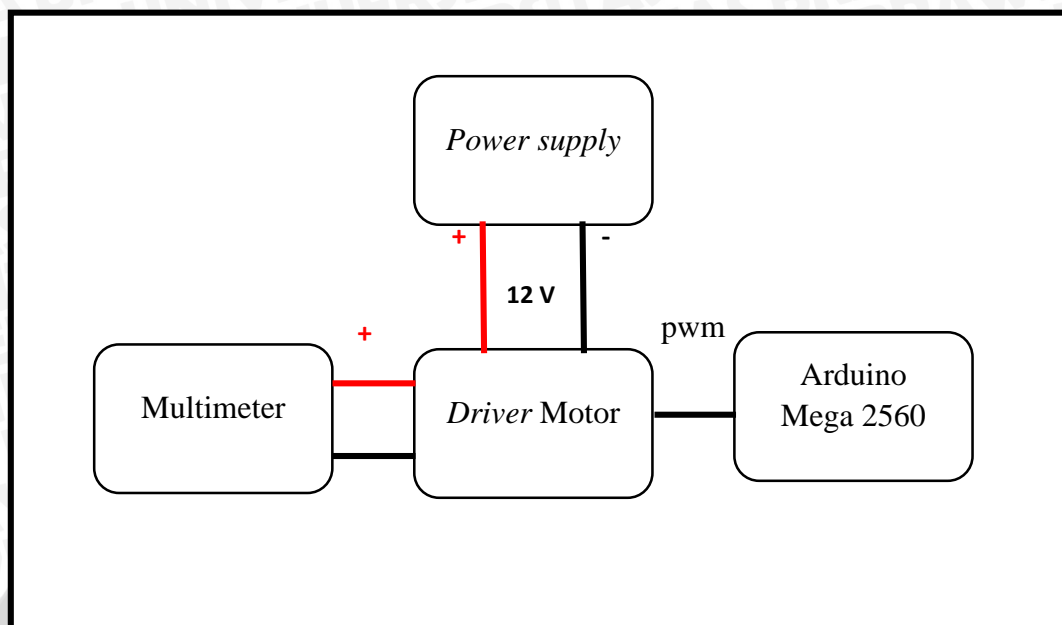
#### b. Peralatan yang digunakan

1. Power supply Unit.
2. Driver L298N Dual H-Bridge.
3. Arduino Mega 2560 R3.
4. Multimeter.
5. Laptop.
6. Kabel penghubung (*Jumper*).

#### c. Prosedur Pengujian

1. Menuliskan listing program pada software arduino untuk mengubah *duty cycle*.
2. Hubungkan *output* tegangan Power supply sebagai input tegangan pada *driver* motor L298N Dual H-Bridge seperti dalam Gambar 3.12.
3. Hubungkan *input* tegangan *driver* motor L298N Dual H-Bridge dengan pin keluaran PWM di arduino mega R3 seperti pada Gambar 3.12.
4. Hubungkan *output* tegangan *driver* motor L298N Dual H-Bridge dengan multimeter seperti pada Gambar 3.12.
5. Atur *duty cycle* sinyal PWM pada Arduino Mega 2560 R3 dengan nilai 0%-100%.
6. Amati dan catat hasil pembacaan multimeter di setiap kenaikan 5% *duty cycle*.

Perancangan sistem untuk pengujian *Driver L298N Dual H-Bridge* ditunjukkan dalam gambar 3.12.



Gambar 3.12 Perancangan *Driver L298N Dual H-Bridge*.

#### d. Hasil Pengujian

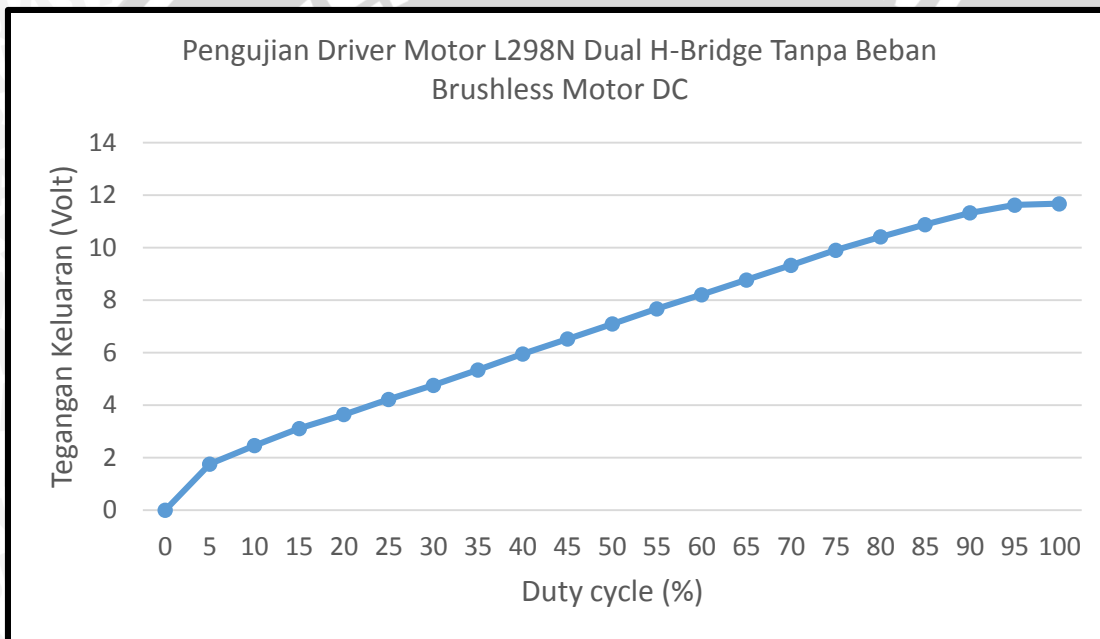
Data pengujian *Driver Motor L298N Dual H-Bridge* tanpa beban ditunjukkan dalam Tabel 3.3

No	Duty cycle (%)	PWM	Vout (Volt)
1	0	0	0
2	5	12.75	1.76
3	10	25.5	2.46
4	15	38.25	3.11
5	20	61	3.64
6	25	63.75	4.22
7	30	76.5	4.76
8	35	89.25	5.34
9	40	102	5.95
10	45	114.75	6.52
11	50	127.5	7.09
12	55	140.25	7.67
13	60	153	8.21
14	65	165.75	8.77
Lanjutan			

Lanjutan			
15	70	178.5	9.33
16	75	191.25	9.9
17	80	204	10.41
18	85	216.75	10.88
19	90	229.5	11.32
20	95	242.25	11.62
21	100	255	11.67

**Tabel 3.3** Hasil Pengujian *Driver* Motor L298N Dual H-Bridge

Berdasarkan Tabel 3.3 akan didapatkan kurva antara *output* tegangan *driver* (Volt) terhadap masukan *duty cycle* (%) sinyal PWM, seperti yang ditunjukkan dalam Gambar 3.13.



**Gambar 3.13** Grafik Perubahan Tegangan *Output Driver* L298N Dual H-Bridge terhadap *Input Duty cycle*.

Sesuai dengan hasil grafik dalam Gambar 3.13 dapat disimpulkan bahwa semakin besar nilai *duty cycle* maka semakin besar pula nilai tegangan keluaran dari *driver*. tegangan keluaran *driver* tidak dipengaruhi oleh beban karena pengujian ini dilakukan tanpa beban sehingga nilai tegangan keluaran nya mendekati nilai tegangan *input driver* yaitu mencapai 12 V. Kesimpulan dari pengujian diatas didapat *gain* atau fungsi alih *driver* motor sebesar:

$$m = \frac{V_{out13} - V_{out9}}{VDutycycle_{13} - VDutycycle_9} = \frac{8.21 - 5.95}{60 - 40} = 0.113$$

### 3.10 Pengujian Sensor Gas

#### a. Tujuan

Dapat mengetahui Karakteristik, kinerja serta *output* sensor Gas dengan membandingkan nilai ADC yang telah dibaca oleh sensor yang telah dikonversi ke dalam bentuk ppm dan *output* sebesar 0-5 V yang diberikan oleh Arduino.

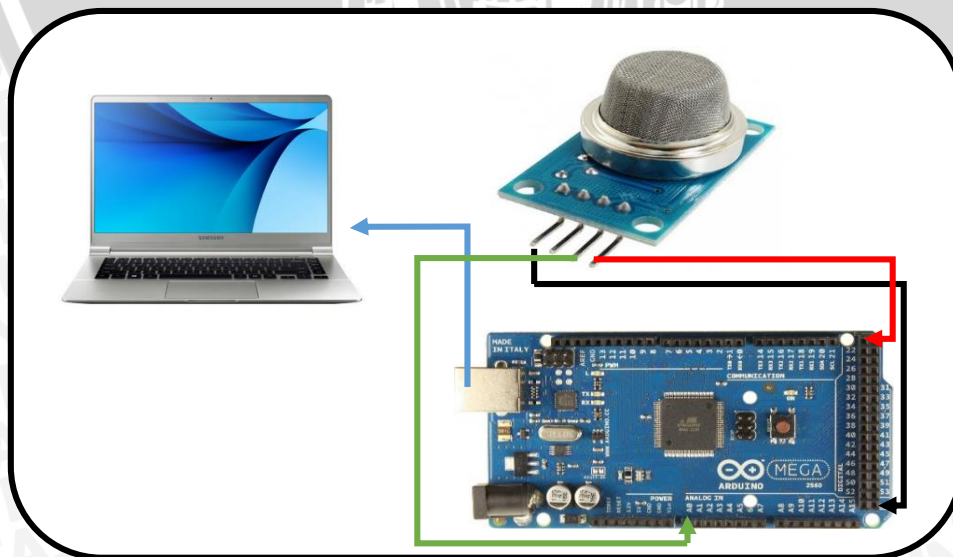
#### b. Peralatan yang digunakan

1. *Power supply* Unit.
2. Arduino Mega 2560 R3.
3. Sensor Gas MQ-6.
4. Laptop.
5. *Project Board*.
6. Kabel Penghubung (*Jumper*).

#### c. Prosedur Pengujian

1. Membuat *listing program* untuk sensor gas di *software* arduino.
2. Hubungkan AO,  $V_{cc}$ , dan GND sensor MQ-6 ke arduino mega seperti dalam Gambar 3.14.
3. Letakan sensor di dalam plant yang sudah ditentukan.
4. Hubungkan arduino dengan PC seperti dalam Gambar 3.14.
5. Letakkan korek gas tepat di sensor MQ-6
6. Amati dan catat hasil pembacaan sensor gas yang di mana gas dibocorkan tadi selama 30 detik.

Perancangan sistem untuk pengujian sensor gas MQ-6 ditunjukkan dalam gambar 3.14.



Gambar 3.14 Perancangan Sensor Gas MQ-6.

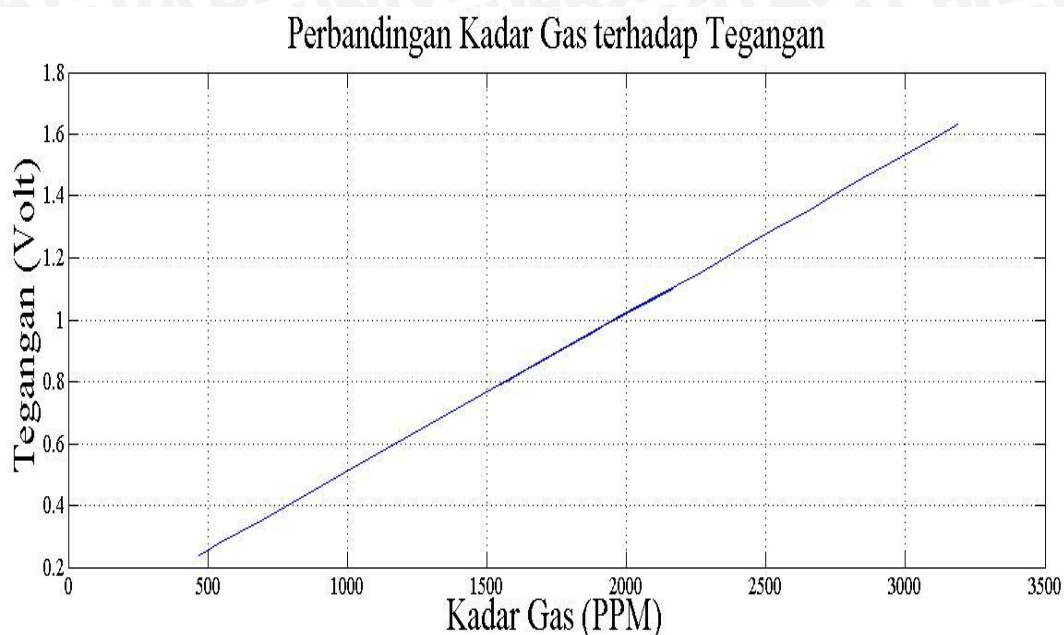
#### d. Hasil Pengujian

Data hasil pengujian sensor gas MQ-6 ditunjukkan dalam Tabel 3.4.

No	Waktu	ADC	Kadar Gas (ppm)	Tegangan (Volt)
1	1	49	469.17	0.24
2	2	49	469.17	0.24
3	3	51	488.32	0.25
4	4	57	545.77	0.28
5	5	72	689.4	0.35
6	6	104	995.8	0.51
7	7	137	1311.78	0.67
8	8	162	1551.15	0.79
9	9	163	1560.72	0.8
10	10	164	1570.3	0.8
11	11	166	1589.45	0.81
12	12	167	1599.03	0.82
13	13	172	1646.9	0.84
14	14	176	1685.2	0.86
15	15	184	1761.8	0.9
16	16	186	1780.95	0.91
17	17	195	1867.13	0.95
18	18	202	1933.105	0.99
19	19	206	1972.45	1.01
20	20	212	2029.9	1.04
21	21	221	2116.07	1.08
22	22	226	2163.95	1.1
23	23	236	2259.7	1.15
24	24	252	2412.9	1.23
25	25	266	2546.95	1.3
26	26	277	2652.27	1.35
27	27	288	2757.6	1.41
28	28	298	2853.35	1.46
29	29	306	2929.95	1.5
30	30	325	3111.87	1.59

Tabel 3.4 Hasil Pengujian Sensor Gas MQ-6

Berdasarkan Tabel 3.4 didapatkan grafik *output* tegangan pada arduino mega 2560 R3 sebesar 0-5V terhadap pembacaan kadar gas pada sensor sebesar 200-10000 ppm, seperti yang ditunjukkan dalam Gambar 3.15.



**Gambar 3.15** Grafik Tegangan 0-5V Arduino Mega 2560 Terhadap Sensor Gas MQ-6.

Sesuai dengan hasil grafik dalam Gambar 3.15 dapat disimpulkan bahwa semakin besar kadar gas yang dibaca oleh sensor MQ-6 maka semakin besar pula tegangan arduino mega 2560. Kesimpulan dari pengujian diatas didapatkan *gain* atau fungsi alih sensor sebesar:

$$m = \frac{V_{out_{30}} - V_{out_8}}{Kadargas_{30} - Kadargas_8} = \frac{1.59 - 0.79}{3111.87 - 1551.15} = 0.00051258$$

### 3.11 Pengujian Sistem Untuk Penentuan *Steady State Gain* Kipas

#### a. Tujuan

Pengujian dilakukan untuk mengetahui bagaimana penentuan *steady state gain* sistem pengendalian kadar gas dapat bekerja dengan baik sesuai dengan *setpoint* yang diharapkan. *steady state gain* ini menggambarkan hubungan antara kadar gas pembacaan dengan tegangan masukan kipas. Pengujian ini juga berguna untuk menentukan tegangan yang diperlukan untuk mengeluarkan kadar gas dalam *plant*.

#### b. Peralatan yang digunakan

1. Miniatur ruang dapur dengan p x l x t sebesar 35cm x 30 cm x 25 cm.
2. *Power supply Unit* (PSU).
3. 3 buah *driver motor L298N Dual H-Bridge*.
4. 1 kipas DC 12V 2600 rpm.

5. Winn gas *portable*
6. Sensor gas MQ-6.
7. Program dan *software* arduino.
8. LCD 16x2 *blue* dengan modul I2C.
9. Arduino mega 2560 R3.

#### c. Prosedur Pengujian

1. Hubungkan *power supply* unit ke stop kontak AC.
2. Hubungkan 3 buah *driver* motor L298N ke *power supply* unit
3. Hubungkan kipas DC ke *driver*.
4. Hubungkan pin  $v_{cc}$  dan GND arduino ke *driver* motor.
5. Hubungkan juga pin  $v_{cc}$ ,GND,SCL,SDA dari LCD dengan modul I2C ke arduino.
6. Hubungkan sensor MQ-6 ke pin  $v_{cc}$ ,GND, dan AO yang ada pada arduino mega 2560.
7. Membuat program arduino buat tiap pengujian.
8. Mengunduh program sesuai dengan tipe pengujiannya.
9. Catat,amati, dan analisis respon dari sistem setelah mengunduh program sesuai dengan tipe pengujiannya.

#### d. Hasil Pengujian

Data hasil pengujian kipas dc ditunjukkan dalam Tabel 3.5 berikut.

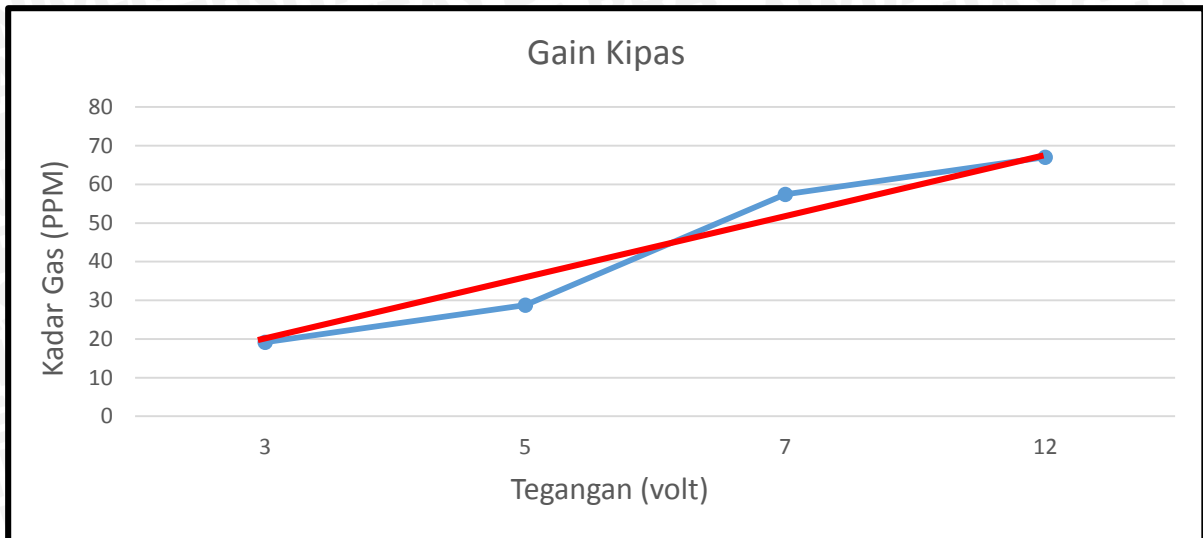
Waktu	Tegangan Kipas							
	3V		5V		7V		12V	
1	3388.47	0	3053.35	0	3187.4	0	3235.27	0
2	3388.47	0	3053.35	0	3158.67	28.73	3206.55	28.72
3	3369.32	19.15	3024.62	28.73	3129.95	57.45	3168.25	67.02
4	3331.02	57.45	2976.75	76.6	3110.8	76.6	3043.77	191.5
5	3302.3	86.17	2967.17	86.18	3101.22	86.18	3273.57	-38.3
6	3283.15	105.32	2967.17	86.18	3101.22	86.18	3120.37	114.9
7	3264	124.47	2976.75	76.6	3091.65	95.75	3024.62	210.65
8	3244.85	143.62	3034.2	19.15	3082.07	105.33	2928.87	306.4
9	3225.7	162.77	3082.07	-28.72	3101.22	86.18	2861.85	373.42
10	3225.7	162.77	3053.35	0	3110.8	76.6	2823.55	411.72
11	3216.12	172.35	3053.35	0	3129.95	57.45	2813.97	421.3
12	3206.55	181.92	3091.65	-38.3	3168.25	19.15	2833.12	402.15
13	3187.4	201.07	3101.22	-47.87	3216.12	-28.72	2861.85	373.42
14	3168.25	220.22	3110.8	-57.45	3273.57	-86.17	2890.57	344.7
15	3177.82	210.65	3158.67	-105.32	3283.15	-95.75	2919.3	315.97
Lanjutan								



Lanjutan								
16	3158.67	229.8	3158.67	-105.32	3302.3	-114.9	2976.75	258.52
17	3187.4	201.07	3168.25	-114.9	3292.72	-105.32	3005.47	229.8
18	3158.67	229.8	3139.52	-86.17	3302.3	-114.9	3024.62	210.65
19	3254.42	134.05	3091.65	-38.3	3110.8	76.6	2909.72	325.55
20	3264	124.47	2995.9	57.45	2392.67	794.73	2708.65	526.62
21	3311.87	76.6	2976.75	76.6	1492.63	1694.77	1674.55	1560.72
22	3311.87	76.6	2727.8	325.55	831.95	2355.45	774.5	2460.77
23	3206.55	181.92	2507.57	545.78	688.32	2499.08	678.75	2556.52
24	3139.52	248.95	1722.42	1330.93	688.32	2499.08	669.17	2566.1
25	2938.45	450.02	1061.75	1991.6	678.75	2508.65	659.6	2575.67
26	2823.55	564.92	697.9	2355.45	669.17	2518.23	659.6	2575.67
27	2641.62	746.85	688.32	2365.03	669.17	2518.23	659.6	2575.67
28	2383.1	1005.37	678.75	2374.6	669.17	2518.23	659.6	2575.67
29	1684.13	1704.34	669.17	2384.18	669.17	2518.23	650.03	2585.24
30	889.4	2499.07	669.17	2384.18	659.6	2527.8	650.03	2585.24
31	707.47	2681	669.17	2384.18	659.6	2527.8	650.03	2585.24
32	688.32	2700.15	659.6	2393.75	669.17	2518.23	650.03	2585.24
33	697.9	2690.57	659.6	2393.75	659.6	2527.8	659.6	2575.67
34	688.32	2700.15	669.17	2384.18	659.6	2527.8	650.03	2585.24
35	678.75	2709.72	659.6	2393.75	659.6	2527.8	650.03	2585.24
36	678.75	2709.72	659.6	2393.75	669.17	2518.23	650.03	2585.24
37	678.75	2709.72	659.6	2393.75	669.17	2518.23	650.03	2585.24
38	678.75	2709.72	650.03	2403.32	669.17	2518.23	650.03	2585.24
39	678.75	2709.72	659.6	2393.75	659.6	2527.8	650.03	2585.24
40	678.75	2709.72	659.6	2393.75	669.17	2518.23	650.03	2585.24
41	669.17	2719.3	650.03	2403.32	669.17	2518.23	650.03	2585.24
42	669.17	2719.3	659.6	2393.75	659.6	2527.8	650.03	2585.24
43	669.17	2719.3	650.03	2403.32	669.17	2518.23	650.03	2585.24
44	659.6	2728.87	650.03	2403.32	659.6	2527.8	650.03	2585.24
45	659.6	2728.87	650.03	2403.32	669.17	2518.23	659.6	2575.67
46	659.6	2728.87	659.6	2393.75	669.17	2518.23	650.03	2585.24
47	659.6	2728.87	650.03	2403.32	669.17	2518.23	650.03	2585.24
48	659.6	2728.87	650.03	2403.32	669.17	2518.23	659.6	2575.67
49	659.6	2728.87	659.6	2393.75	669.17	2518.23	650.03	2585.24
50	669.17	2719.3	650.03	2403.32	669.17	2518.23	650.03	2585.24
51	659.6	2728.87	650.03	2403.32	669.17	2518.23	650.03	2585.24
52	659.6	2728.87	650.03	2403.32	669.17	2518.23	659.6	2575.67

**Tabel 3.5** Hasil Pengujian Kipas DC.

Berdasarkan Tabel 3.5 didapatkan kurva jumlah kadar gas yang dapat dikeluarkan oleh kipas terhadap tegangan yang diberikan pada kipas, seperti yang ditunjukkan dalam gambar 3.16.

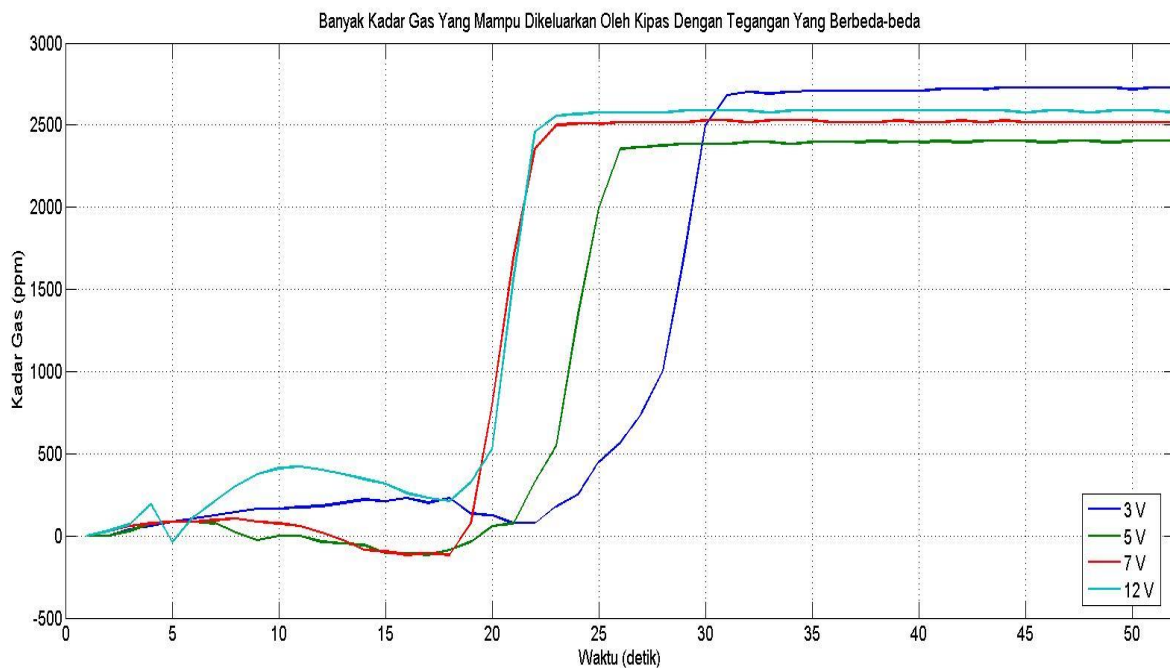


**Gambar 3.16** Steady State Gain Kipas.

Sesuai dengan hasil grafik dalam Gambar 3.16 dapat disimpulkan bahwa banyaknya kadar gas yang dapat dikeluarkan oleh kipas berbeda-beda tergantung dari tegangan yang diberikan. Kesimpulan dari pengujian didapatkan *gain* kipas, sebagai berikut:

$$m = \frac{\text{Kadar gas}_{12} - \text{Kadar gas}_3}{V_{out_{12}} - V_{out_3}} = \frac{67.02 - 19.15}{12 - 3} = 5.318889$$

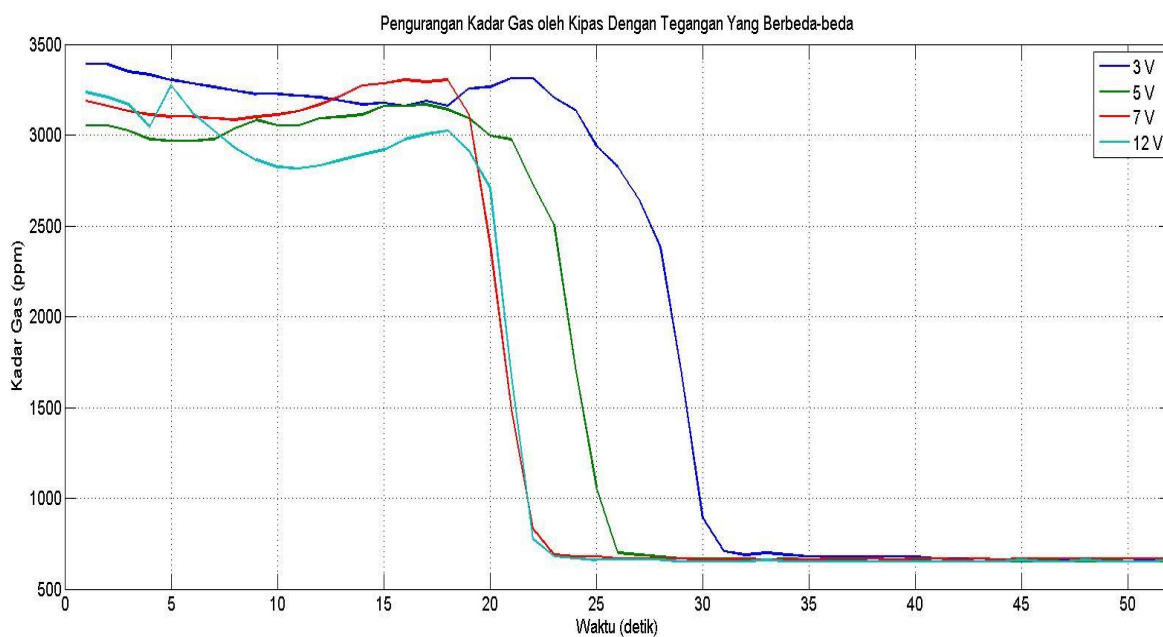
Berdasarkan Tabel 3.5 didapatkan grafik banyaknya kadar gas yang dapat dikeluarkan oleh kipas terhadap waktu dengan memberikan tegangan kipas yang berbeda, seperti yang ditunjukkan dalam gambar 3.17.



**Gambar 3.17** Grafik Jumlah Gas Yang Dapat Dikeluarkan Oleh Kipas Dengan Tegangan Yang Berbeda-beda.

Sesuai dengan hasil grafik dalam Gambar 3.17 dapat disimpulkan bahwa pengurangan kadar gas dapat mencapai 2500 ppm, akan tetapi waktu yang diperlukan untuk mencapai pengurangan kadar gas sebesar 2500 ppm berbeda-beda tergantung tegangan yang diberikan pada *driver* motor. Untuk *driver* motor dengan tegangan 3 V memerlukan waktu selama 35 detik. *Driver* motor dengan tegangan 5 V memerlukan waktu selama 28 detik. *Driver* motor dengan tegangan 7 V memerlukan waktu selama 25 detik. Sedangkan *driver* motor dengan tegangan 12 V memerlukan waktu selama 23 detik. Gambar grafik didapatkan dari Tabel 3.5.

Berdasarkan Tabel 3.5 didapatkan grafik penurunan kadar gas terhadap waktu dengan memberikan tegangan kipas yang berbeda, seperti yang ditunjukkan dalam gambar 3.18.



**Gambar 3.18** Grafik Penurunan Kadar Gas Hingga Mencapai Keadaan Normal Dengan Tegangan Yang Berbeda-beda.

Sesuai dengan hasil grafik dalam Gambar 3.18 dapat disimpulkan bahwa penurunan kadar gas memerlukan waktu yang berbeda-beda untuk mencapai keadaan normal atau 675 ppm tergantung dengan tegangan yang diberikan pada *driver* motor. Untuk *driver* motor dengan tegangan 3 V memerlukan waktu selama 35 untuk mencapai keadaan normal. *Driver* motor dengan tegangan 5 V memerlukan waktu selama 28 detik untuk mencapai keadaan normal. *Driver* motor dengan tegangan 7 V memerlukan waktu selama 25 detik untuk

mencapai keadaan normal. Sedangkan *driver* motor dengan tegangan maksimal atau 12 V memerlukan waktu 23 detik untuk mencapai keadaan normal. Gambar grafik didapatkan dari Tabel 3.5.

### 3.12 Perancangan Kontroler P dengan Metode *Hand Tuning*

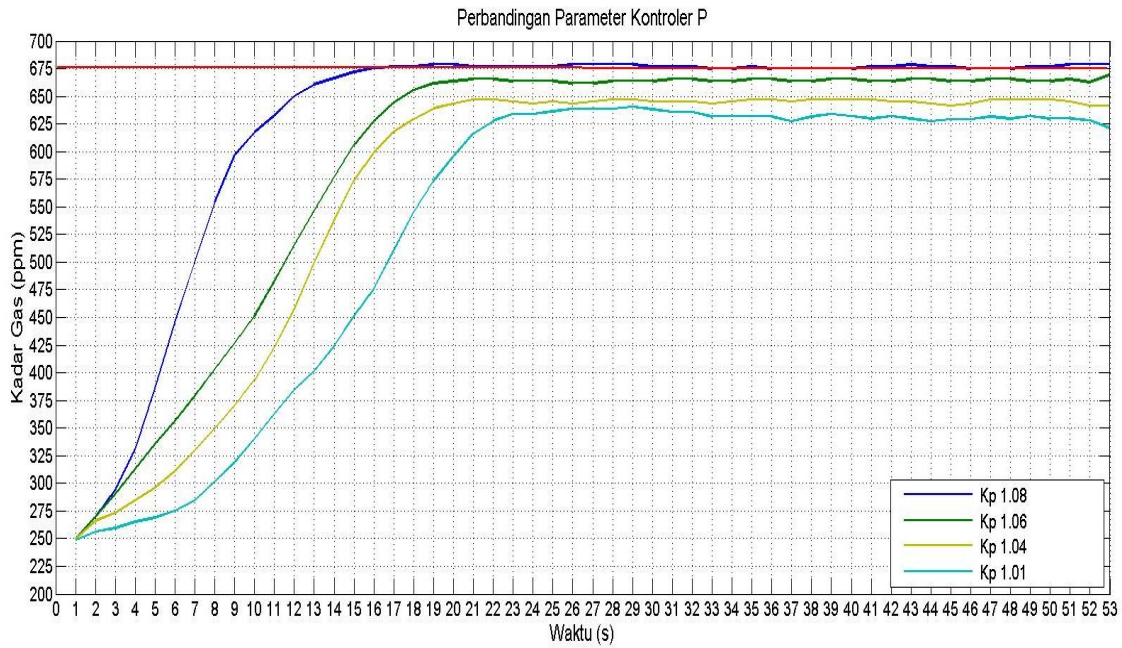
Kontroler P dapat di-*tuning* dalam beberapa cara, antara lain *Ziegler-Nichols tuning*, *loop shaping*, metode analitis, optimasi, *pole placement*, *auto tuning* dan *hand tuning* (Smith,1997; Astrom & Hagglund,2004). Perancangan kontroler P diperlukan pada sistem ketika sistem diberi gangguan agar sistem bisa mencapai keadaan normalnya atau *steady*. Pada Perancangan ini, metode yang digunakan untuk menentukan besar  $K_p$  adalah metode *Hand Tuning*. Penggunaan metode ini dilakukan karena terdapat kendala untuk menggunakan metode lain yang disebutkan di atas yaitu tidak adanya model matematis, sehingga membuat cara analitis sulit untuk dilakukan dan karena mendapatkan respon yang tidak sesuai dengan yang diinginkan ketika menggunakan metode yang lain. Prosedur untuk melakukan *tuning* adalah sebagai berikut:

1. Atur dengan memberi nilai  $K_p$  yang berbeda-beda hingga sistem dapat mencapai nilai *setpoint*.

Pengujian Ke-	$K_p$	Mencapai Setpoint
1.	1.01	Tidak
2.	1.04	Tidak
3.	1.06	Tidak
4.	1.08	Ya

**Tabel 3.6** Pengujian nilai  $K_p$  pada sistem.

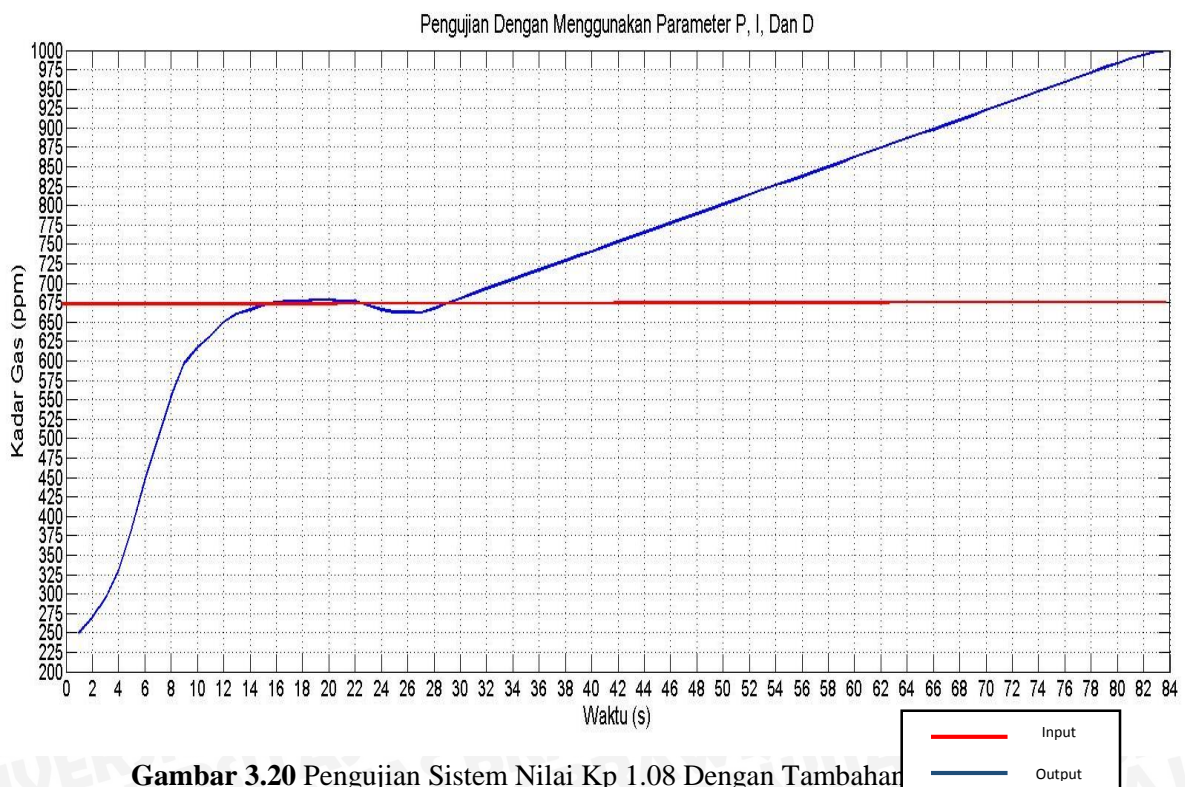
Berdasarkan Tabel 3.6 dapat disimpulkan bahwa sistem yang menggunakan nilai  $K_p$  1.08 dapat mencapai *setpoint*.



**Gambar 3.19** Pengujian Sistem Dengan Nilai Kp Yang Berbeda-beda.

Sesuai dengan hasil grafik dalam Gambar 3.19 dapat disimpulkan bahwa sistem yang menggunakan nilai Kp 1.08 dapat mencapai *setpoint* dibanding dengan nilai Kp yang lainnya dan respon sistem yang didapatkan jauh lebih baik jika menggunakan Kp 1.08.

2. Beri nilai Ki = 10 dan Kd = 0.0001 pada sistem dengan Kp = 1.08 untuk mengetahui respon sistem jika diberi penambahan kontroler integral dan derivatif.



**Gambar 3.20** Pengujian Sistem Nilai Kp 1.08 Dengan Tambahan

Sesuai dengan hasil grafik dalam Gambar 3.20 dapat disimpulkan bahwa sistem yang menggunakan nilai  $K_p = 1.08$ ,  $K_i = 10$ , dan  $K_d = 0.0001$  tidak dapat mencapai *setpoint* karena untuk sistem pengendalian gas dalam ruang tertutup mengalami kenaikan secara drastis dari nilai *setpoint* yang ditentukan yaitu 675 ppm hingga mencapai nilai 1000 ppm. itu membuktikan bahwa dengan menambahkan kontroler integral dan derivatif pada sistem membuat sistem menjadi *error* padahal sistem tidak mengalami kebocoran gas tapi *output* sistem mengalami kenaikan. Oleh karena itu dalam pengendalian kadar gas dalam ruang hanya menggunakan kontroler proporsional dalam penelitian.

UNIVERSITAS BRAWIJAYA



## BAB IV

### HASIL DAN PEMBAHASAN

Dalam hasil dan pembahasan dilakukan pengujian sistem secara keseluruhan, hal ini untuk mengetahui apakah alat yang telah dibuat sesuai dengan rencana awal dan berfungsi dengan sebagaimana mestinya. Pengujian pada sistem secara keseluruhan dibagi menjadi dua bagian yaitu, Pengujian sistem dengan kontroler tanpa gangguan dan pengujian sistem dengan kontroler dengan gangguan.

Dalam pengujian keseluruhan sistem ini terdapat beberapa aspek yang harus diperhatikan, antara lain:

#### a Tujuan

Pengujian dilakukan untuk mengetahui bagaimana respon keseluruhan sistem pengendalian kadar gas sesuai dengan *setpoint* yang diharapkan.

#### b Peralatan yang digunakan

1. Miniatur ruang dapur dengan p x l x t sebesar 35cm x 30 cm x 25 cm.
2. *Power supply Unit* (PSU).
3. 3 buah *driver motor L298N Dual H-Bridge*.
4. Kipas DC 12V 2600 rpm.
5. Winn gas *portable*
6. Sensor gas MQ-6.
7. Program dan *software* arduino.
8. LCD 16x2 *blue* dengan modul I2C.
9. Arduino mega 2560 R3.

#### c Prosedur Pengujian

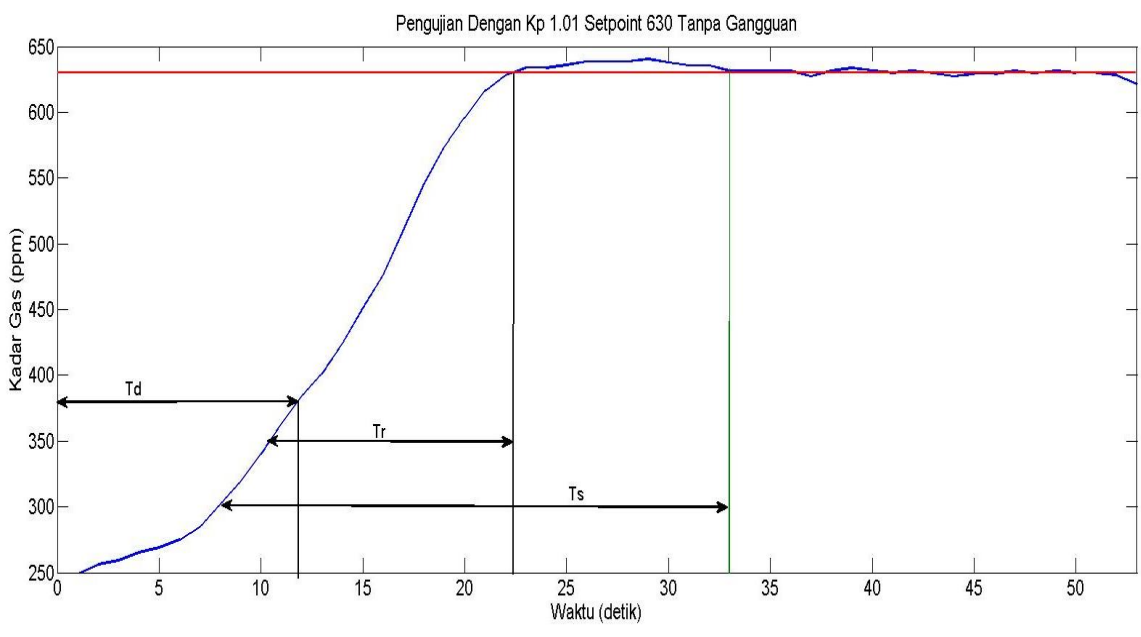
1. Hubungkan *power supply* unit ke stop kontak AC.
2. Hubungkan 3 buah *driver motor L298N* ke *power supply* unit
3. Hubungkan kipas DC, 2 kipas pertama ke *driver*.
4. Hubungkan pin  $v_{cc}$  dan GND arduino ke *driver motor*.
5. Hubungkan juga pin  $v_{cc}$ ,GND,SCL,SDA dari LCD dengan modul I2C ke arduino.
6. Hubungkan sensor MQ-6 ke pin  $v_{cc}$ ,GND, dan AO yang ada pada arduino mega 2560.

7. Membuat program arduino buat tiap pengujian.
8. Mengunduh program sesuai dengan tipe pengujiannya.
9. Catat, amati, dan analisis respon dari sistem seteleh mengunduh program sesuai dengan tipe pengujiannya.

#### d Hasil Pengujian

##### 1. Pengujian sistem dengan kontroler tanpa gangguan

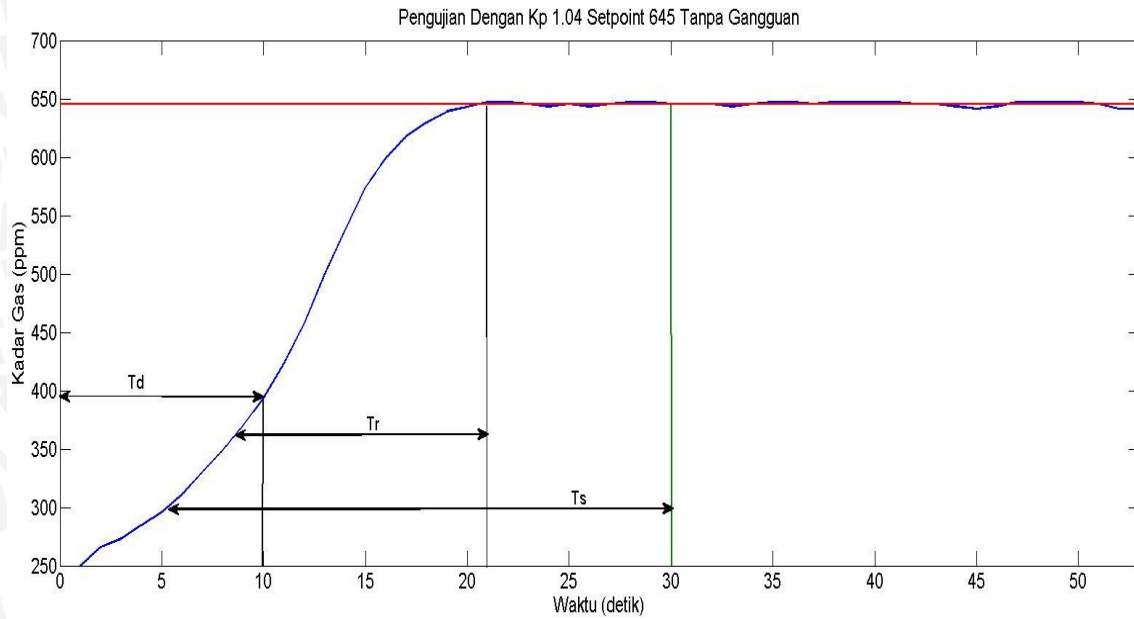
Hasil dari pengujian nilai  $K_p$  1.01 dengan *setpoint* 630 tanpa gangguan dapat dilihat dalam Gambar 4.1.



**Gambar 4.1** Grafik Respon Sistem *Setpoint* 630 Dengan Kontroler Tanpa Gangguan.

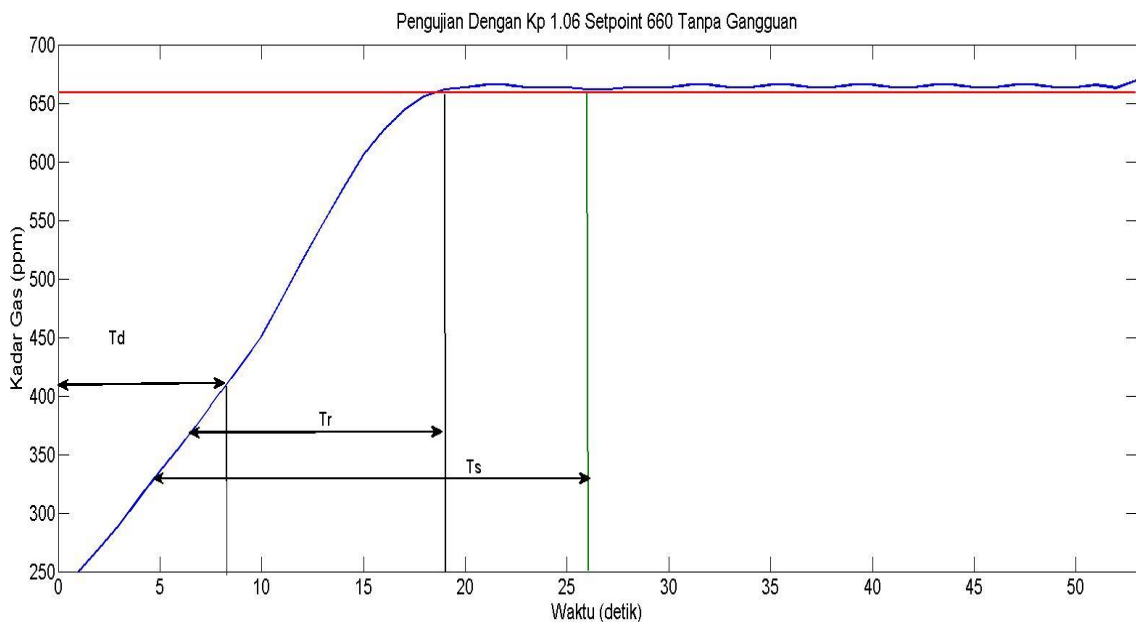
Sesuai dengan hasil grafik dalam Gambar 4.1 dapat disimpulkan bahwa waktu yang diperlukan sistem untuk mencapai keadaan mantap ( $t_s$ ) selama 33 detik, waktu naik ( $t_r$ ) selama 22.6 detik, waktu tunda ( $t_d$ ) selama 11.8 detik, *error steady state* 0.139 %. Hasil pengujian sistem  $K_p$  1.04 dengan *setpoint* 645 tanpa gangguan dapat dilihat dalam Gambar 4.2.





**Gambar 4.2** Grafik Respon Sistem *Setpoint* 645 Dengan Kontroler Tanpa Gangguan.

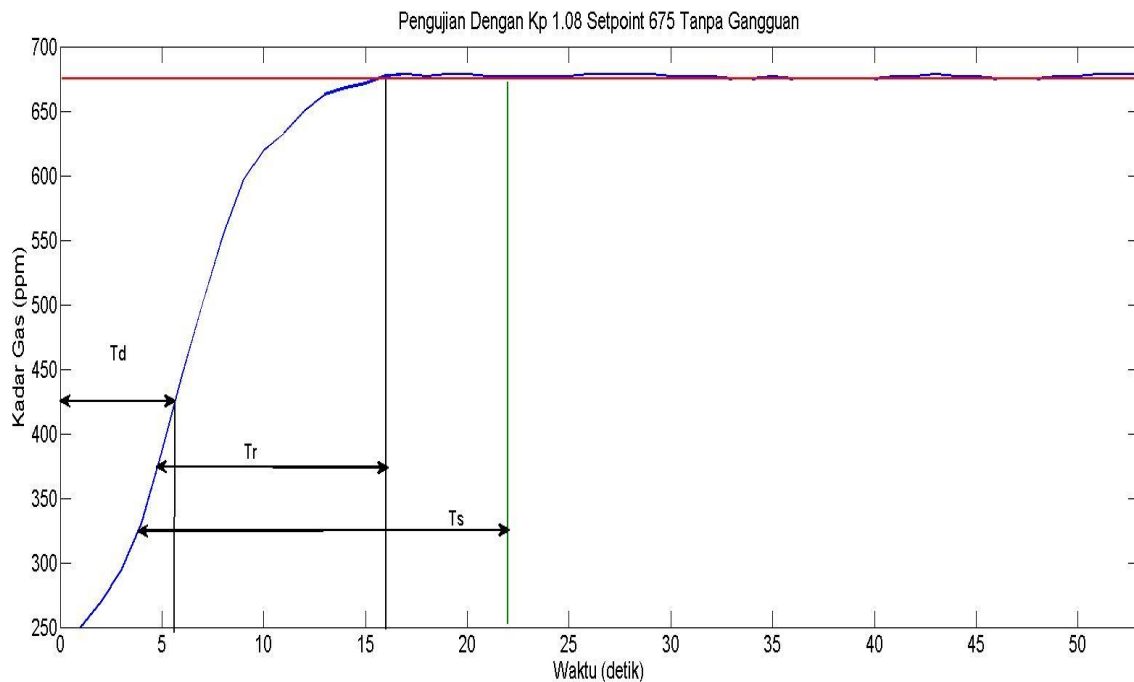
Sesuai dengan hasil grafik dalam Gambar 4.2 dapat disimpulkan bahwa waktu yang diperlukan sistem untuk mencapai keadaan mantap ( $t_s$ ) selama 30 detik, waktu naik ( $t_r$ ) selama 20.3 detik, waktu tunda ( $t_d$ ) selama 10 detik, *error steady state* 2.170 %. Hasil pengujian sistem Kp 1.06 dengan *setpoint* 660 tanpa gangguan dapat dilihat dalam Gambar 4.3.



**Gambar 4.3** Grafik Respon Sistem *Setpoint* 660 Dengan Kontroler Tanpa Gangguan.

Sesuai dengan hasil grafik dalam Gambar 4.3 dapat disimpulkan bahwa waktu yang diperlukan sistem untuk mencapai keadaan mantap ( $t_s$ ) selama 26 detik, waktu naik ( $t_r$ )

selama 19 detik, waktu tunda ( $t_d$ ) selama 8 detik, *error steady state* 1.389 %. Hasil pengujian sistem Kp 1.08 dengan *setpoint* 675 tanpa gangguan dapat dilihat dalam Gambar 4.4.



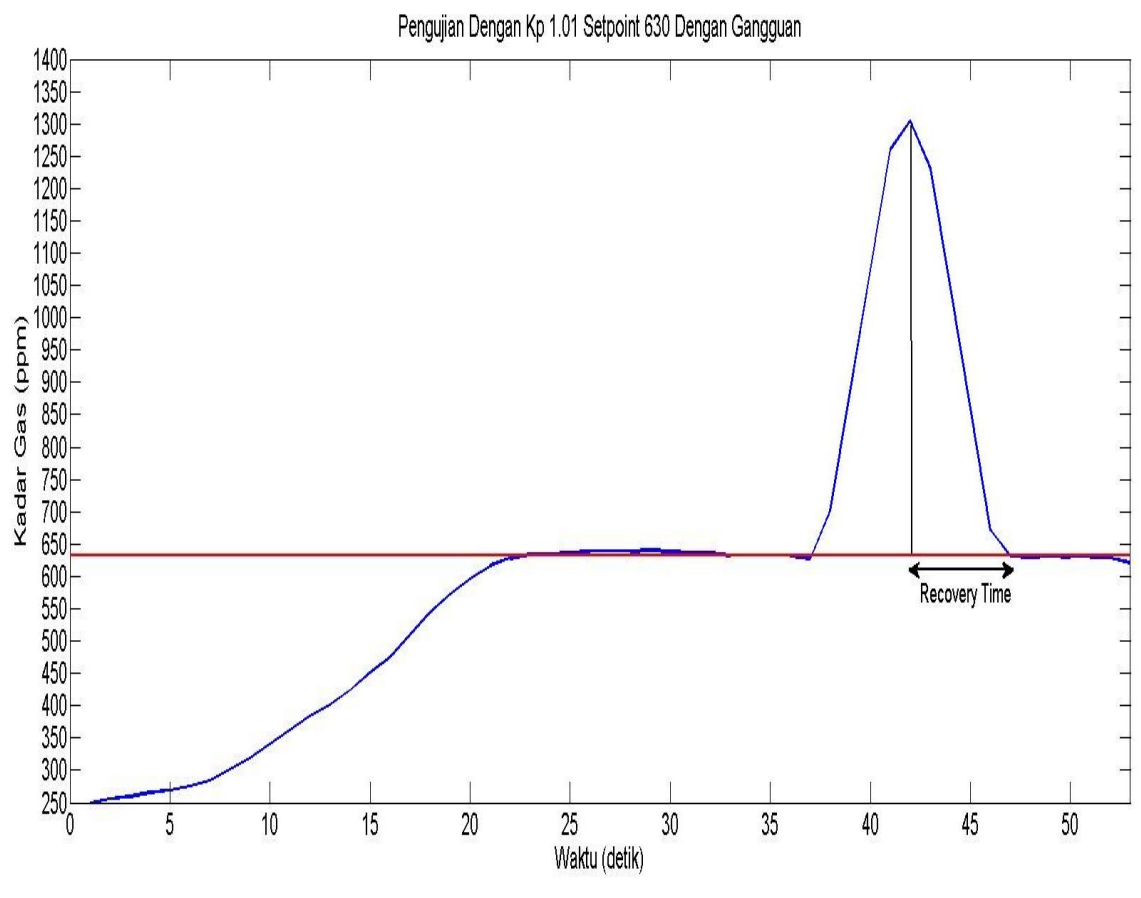
**Gambar 4.4** Grafik Respon Sistem *Setpoint* 675 Dengan Kontroler Tanpa Gangguan.

Sesuai dengan hasil grafik dalam Gambar 4.4 dapat disimpulkan bahwa waktu yang diperlukan sistem untuk mencapai keadaan mantap ( $t_s$ ) selama 22 detik, waktu naik ( $t_r$ ) selama 16 detik, waktu tunda ( $t_d$ ) selama 5.8 detik, *error steady state* 0.88 %.

Kesimpulan dari 4 pengujian dengan kontroler tanpa gangguan dengan nilai Kp yang berbeda-beda akhirnya didapatkan bahwa nilai respon sistem yang bagus ada pada nilai Kp 1.08 karena dapat mencapai nilai *setpoint* yang diinginkan dengan waktu *steady*-nya paling cepat yaitu 22 detik.

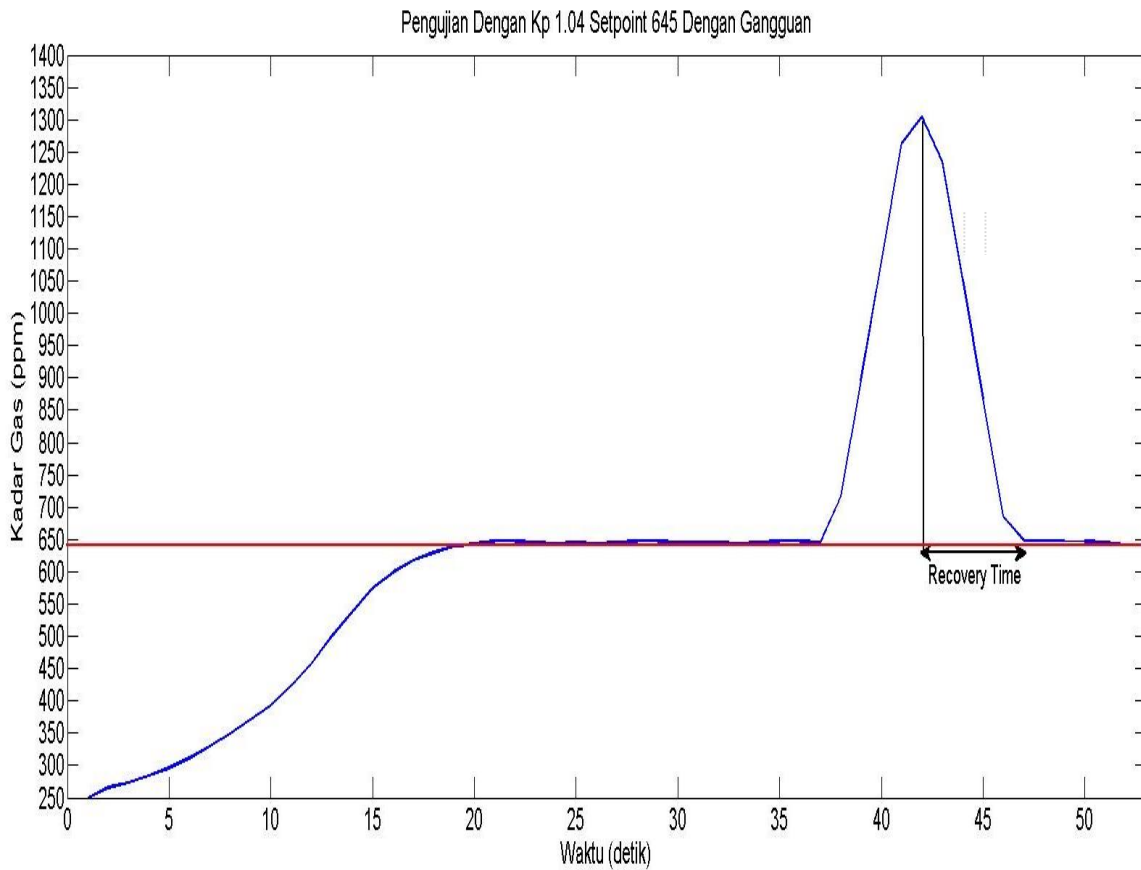
## 2. Pengujian sistem dengan kontroler tanpa gangguan

Hasil dari pengujian nilai  $K_p$  1.01 dengan *setpoint* 630 dengan gangguan dapat dilihat pada Gambar 4.5.



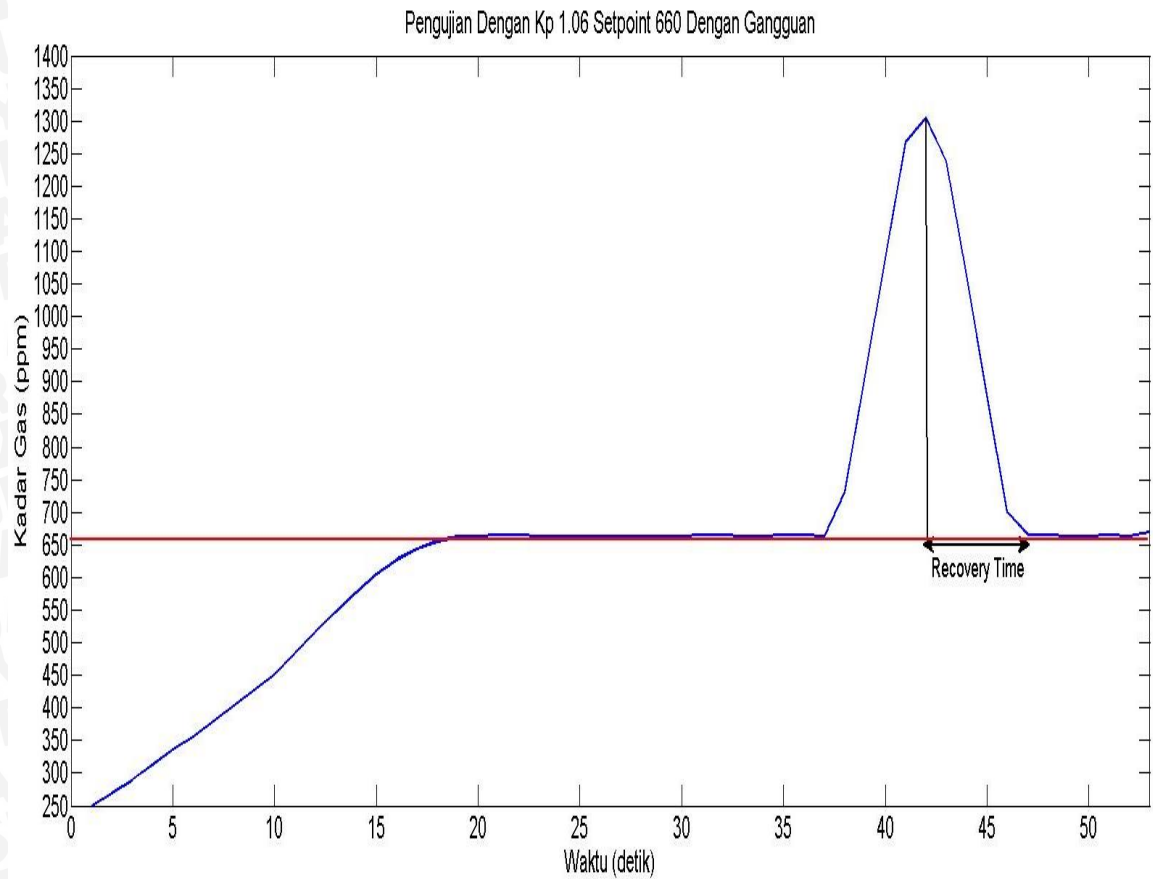
**Gambar 4.5** Grafik Respon Sistem *Setpoint* 630 Dengan Kontroler Dengan Gangguan.

Sesuai dengan hasil grafik dalam Gambar 4.5 dapat disimpulkan bahwa sistem dengan nilai  $K_p$  1.01 mengalami perubahan kondisi setelah diberi gangguan dengan membocorkan gas. Sistem akan kembali ke keadaan mantap ketika gas dalam ruang sudah sesuai dengan *setpoint*. *Recovery time* pada sistem adalah 5 detik. Hasil pengujian sistem  $K_p$  1.04 dengan *setpoint* 645 dengan gangguan dapat dilihat dalam Gambar 4.6.



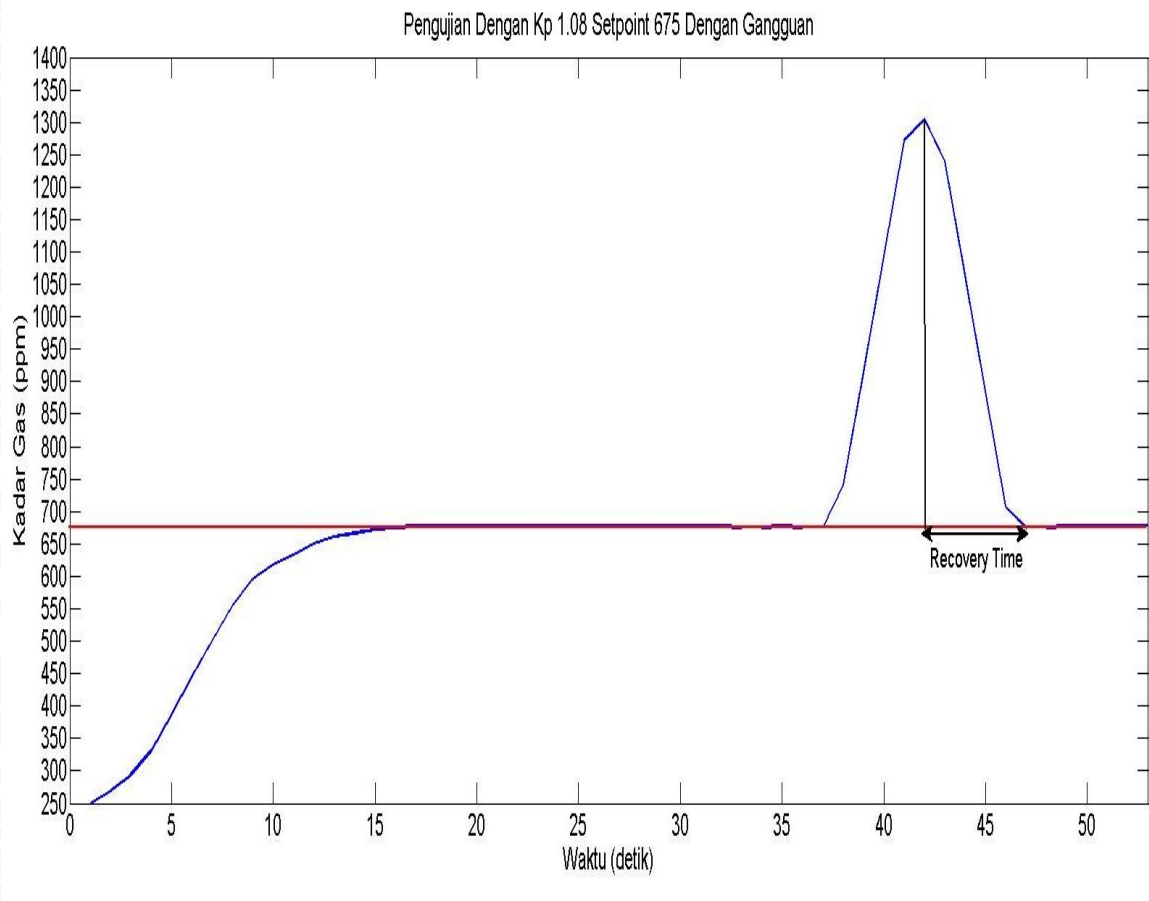
**Gambar 4.6** Grafik Respon Sistem *Setpoint* 645 Dengan Kontroler Dengan Gangguan.

Sesuai dengan hasil grafik dalam Gambar 4.6 dapat disimpulkan bahwa sistem dengan nilai Kp 1.04 mengalami perubahan kondisi setelah diberi gangguan dengan membocorkan gas. Sistem akan kembali ke keadaan mantap ketika gas dalam ruang sudah sesuai dengan *setpoint*. *Recovery time* pada sistem adalah 5 detik. Hasil pengujian sistem Kp 1.06 dengan *setpoint* 660 tanpa gangguan dapat dilihat dalam Gambar 4.7.



**Gambar 4.7** Grafik Respon Sistem *Setpoint* 660 Dengan Kontroler Dengan Gangguan.

Sesuai dengan hasil grafik dalam Gambar 4.7 dapat disimpulkan bahwa dengan nilai Kp 1.06 mengalami perubahan kondisi setelah diberi gangguan dengan membocorkan gas. Sistem akan kembali ke keadaan mantap ketika gas dalam ruang sudah sesuai dengan *setpoint*. *Recovery time* pada sistem adalah 5 detik. Hasil pengujian sistem Kp 1.08 dengan *setpoint* 675 tanpa gangguan dapat dilihat dalam Gambar 4.8.



**Gambar 4.8** Grafik Respon Sistem *Setpoint* 675 Dengan Kontroler Dengan Gangguan.

Sesuai dengan hasil grafik dalam Gambar 4.8 dapat disimpulkan bahwa sistem dengan nilai  $K_p$  1.08 mengalami perubahan kondisi setelah diberi gangguan dengan membocorkan gas. Sistem akan kembali ke keadaan mantap ketika gas dalam ruang sudah sesuai dengan *setpoint*. *Recovery time* pada sistem adalah 5 detik.

Kesimpulan dari 4 pengujian dengan kontroler dengan gangguan dengan nilai  $K_p$  yang berbeda-beda akhirnya didapatkan bahwa respon sistem yang mengalami gangguan akan kembali *steady* dalam waktu 5 detik. *Recovery time* dari  $K_p$  1.01, 1.04, 1.06, dan 1.08 yang diuji sama yaitu 5 detik. Hal ini dikarenakan nilai pembacaan sensor gas di dalam ruang yang mengalami kebocoran gas masih dalam *range* yang sama sehingga nilai *recovery time*-nya sama.

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Dari hasil perancangan dan pengujian alat yang telah dilakukan, maka didapatkan:

1. Sistem pengendalian kadar gas dengan menggunakan metode *hand tuning* untuk menentukan nilai parameter kontroler P, didapatkan parameter P terbaik yaitu  $K_p = 1.08$  dengan nilai *setpoint*-nya 675 ppm.
2. Alat ini telah mencapai hasil sesuai dengan yang diharapkan, di mana kadar dapat dikendalikan sesuai dengan *setpoint*. Waktu yang diperlukan sistem untuk mencapai keadaan mantap adalah keadaan mantap ( $t_s$ ) selama 22 detik dengan *error steady state* 0.88 %.

#### 5.2 Saran

Dalam perancangan dan pembuatan alat ini tentunya tidak terlepas dari kesalahan maupun kekurangan. Diharapkan untuk kedepannya diperbaiki kinerja alat dan pengembangan lebih lanjut agar dapat mencapai sistem yang optimal, seperti:

1. Dengan menambahkan jumlah sensor di setiap sisi agar pembacaan sensor dapat dibandingkan dengan pembacaan sensor lainnya yang di tempatkan di posisi yang berbeda.
2. Menambahkan perancangan motor servo pada proses membuka-menutup plant ketika mengalami kebocoran gas, karena pada penelitian ini ketika mencapai nilai pembacaan maksimal kadar gas dalam plant proses membuka-menutup untuk mengeluarkan gas masih dilakukan secara manual.
3. Penggunaan sensor gas selain MQ-6 agar dapat dibandingkan sensor mana yang memiliki sensitivitas terbaik dalam pembacaan gas baik dari penempatan sensor maupun *range* pembacaan sensor.
4. Pengujian sensitivitas sensor hanya bisa mendapatkan nilai 400 ppm hingga 3500 ppm masih jauh dari range sensor yang sebenarnya. diharapkan agar dalam penelitian selanjutnya dapat melakukan pengujian sensitivitas sensor mendekati sensitivitas sensor yang sebenarnya yaitu 200 ppm hingga 10000 ppm.





## DAFTAR PUSTAKA

- Ogata, K. 1985. *Teknik Kontrol Automatik (Sistem Pengaturan) Jilid 1*. Jakarta: Erlangga.
- Ogata, Katsuhiko. 1997. *Modern Control Engineering*. Prentice Hall. New Jersey.
- Philip, C. L. & Harbor, R. D. 1996. *Feedback Control System*. Diterjemahkan oleh Widodo.R.J. Prentice Hall. New Jersey.
- Resa, Wibowo. 2011. ALAT PENDETEKSI DINI KEBOCORAN GAS ELPIJI MENGGUNAKAN SENSOR GAS MQ-6. Skripsi Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Yogyakarta, Yogyakarta.
- Supriadi,S. 2010. brusshles motor DC. <http://solihinsupriadielectric.blogspot.sg/2010/02/brusshles-motor-dc.html> (diakses 28 April 2016).
- Akbar, T. H. 2010. Pendeteksi Kebocoran Tabung Gas Dengan Menggunakan Sensor Gas Figarro TGS 2610 Berbasis Mikrokontroler AT89S52. Jurnal Jurusan Sistem Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Gunadarma, Depok.
- Susana, S., Nataliana, D. & Atiah, U. 2015. Sistem Monitoring Pendeteksi Kebocoran LPG berbasis Mikrokontroler ATmega16 menggunakan RF APC220. Jurnal ELKOMIKA, Jurusan Teknik Elektro Institut Teknologi Nasional (ITENAS) Bandung.
- Erlansyah, D. & Widyanto. 2014. ALAT DETEKSI KEBOCORAN TABUNG GAS ELPIJI BERBASIS MIKROKONTROLER. Jurnal Dosen Universitas Bina Darma, Palembang.
- Herdianto. 2012. Pengetahuan Dasar Tentang Gas. <http://herdiantomedan.blogspot.sg/2012/12/pengetahuan-dasar-tentang-gas.html> (diakses 10 Agustus 2016)



UNIVERSITAS BRAWIJAYA



# LAMPIRAN I

## FOTO ALAT



Gambar tampak depan



Gambar tampak belakang



Gambar tampak kiri





Gambar tampak kanan



Gambar tampak atas



## LAMPIRAN II



## Listing Program

```
float sensor;  
float terbaca;  
float tegangan;  
  
#include <Wire.h>  
#include <LiquidCrystal_I2C.h>  
#include <PID_v1.h>  
  
double Setpoint, Input, Output; // these are just variables for storing values  
  
// Tuning parameters  
  
float Kp=1.08; //Initial Proportional Gain  
float Ki=0; //Initial Integral Gain  
float Kd=0; //Initial Differential Gain  
  
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
```



```
// Specify the links and initial tuning parameters
```

```
PID myPID(&Input, &Output, &Setpoint,Kp,Ki,Kd, DIRECT);
```

```
// Timestamp
```

```
unsigned long serialTime; //this will help us know when to talk with processing
```

```
const int sampleRate = 1; // Variable that determines how fast our PID loop runs
```

```
const long serialPing = 500; //This determines how often we ping our loop
```

```
unsigned long now = 0; //This variable is used to keep track of time
```

```
unsigned long lastMessage = 0; //This keeps track of when our loop last spoke to serial
```

```
int Error;
```

```
float Errorpwm1;
```

```
float k=5.318888889; //gain
```

```
float pwm1;
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  lcd.begin(16,2);
```

```
  //PIN DRIVER SATU MOTOR SATU
```

```
  pinMode(13,OUTPUT); //MIN1 VCC motor1 PWM, driver pin 1
```

```
  pinMode(12,OUTPUT); //MIN2 GRD motor1 PWM, driver pin 2
```

```
  pinMode(22,OUTPUT); //VCC LN1
```

```
  pinMode(23,OUTPUT); //GRD LN2
```

```
  // PIN DRIVER SATU MOTOR DUA
```

```
  pinMode(11,OUTPUT); //MEN, driver pin 3
```



```
pinMode(10,OUTPUT); //MSLP, driver pin 4

pinMode(24,OUTPUT); //VCC LN1

pinMode(25,OUTPUT); //GRD LN2

//PIN DRIVER DUA MOTOR SATU

pinMode(9,OUTPUT); //MIN1 VCC motor1 PWM, driver pin 3

pinMode(8,OUTPUT); //MIN2 GRD motor1 PWM, driver pin 4

pinMode(26,OUTPUT); //VCC LN1

pinMode(27,OUTPUT); //GRD LN2

// PIN DRIVER DUA MOTOR DUA

pinMode(7,OUTPUT); //MEN, driver pin 5

pinMode(6,OUTPUT); //MSLP, driver pin 6

pinMode(28,OUTPUT); //VCC LN1

pinMode(29,OUTPUT); //GRD LN2

//PIN DRIVER TIGA MOTOR SATU

pinMode(5,OUTPUT); //MIN1 VCC motor1 PWM, driver pin 7

pinMode(4,OUTPUT); //MIN2 GRD motor1 PWM, driver pin 8

pinMode(30,OUTPUT); //VCC LN1

pinMode(31,OUTPUT); //GRD LN2

// PIN DRIVER TIGA MOTOR DUA

pinMode(3,OUTPUT); //MEN, driver pin 9

pinMode(2,OUTPUT); //MSLP, driver pin 10

pinMode(32,OUTPUT); //VCC LN1

pinMode(33,OUTPUT); //GRD LN2
```

```

Input = sensor; // read the sensor adc

Setpoint = 675; // Read the SetPoint value from the potentiometer

myPID.SetMode(AUTOMATIC); // Turn on the PID loop as automatic control
myPID.SetOutputLimits(0, 1000);
myPID.SetSampleTime(sampleRate); // Sets the sample rate

lastMessage = millis(); // timestamp
}

void loop() {

// PIN DRIVER SATU MOTOR SATU
digitalWrite(12, LOW);
digitalWrite(22, HIGH);
digitalWrite(23, LOW);

// PIN DRIVER SATU MOTOR DUA
digitalWrite(10, LOW);
digitalWrite(24, HIGH);
digitalWrite(25, LOW);

// PIN DRIVER DUA MOTOR SATU
digitalWrite(8, LOW);
digitalWrite(26, HIGH);
digitalWrite(27, LOW);

// PIN DRIVER DUA MOTOR DUA
digitalWrite(6, LOW);
digitalWrite(28, HIGH);
digitalWrite(29, LOW);

```



```
// PIN DRIVER TIGA MOTOR SATU
```

```
digitalWrite(4, LOW);
```

```
digitalWrite(30, HIGH);
```

```
digitalWrite(31, LOW);
```

```
// PIN DRIVER TIGA MOTOR DUA
```

```
digitalWrite(2, LOW);
```

```
digitalWrite(32, HIGH);
```

```
digitalWrite(33, LOW);
```

```
sensor=analogRead(A0);
```

```
terbaca=((sensor*9.575)+200);
```

```
teganganarduino=(sensor*0.00488759);
```

```
Input = sensor;
```

```
Setpoint = 675; //nilai setpoint yang diinginkan
```

```
myPID.Compute(); // Run the PID loop
```

```
pwm1= Output/k;
```

```
analogWrite(13,pwm1);
```

```
analogWrite(11,0);
```

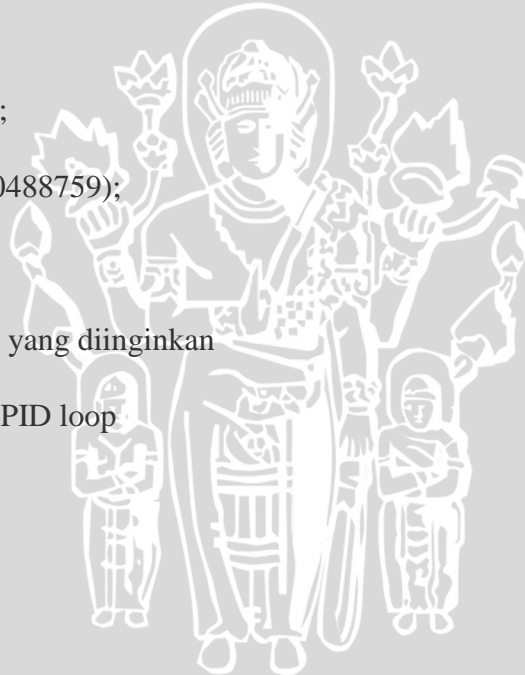
```
analogWrite(9,0);
```

```
analogWrite(7,0);
```

```
analogWrite(5,0);
```

```
analogWrite(3,0);
```

```
delay (1000);
```



```
Serial.print(sensor);  
Serial.print("\t");  
Serial.print(terbaca);  
Serial.print("\t");  
Serial.print(teganganarduino);  
Serial.print("\t");  
Serial.print(pwm1);  
Serial.print("\t");  
Serial.println(Output);  
  
now = millis(); // Keep track of time  
  
lcd.clear();  
lcd.setCursor(2,0);  
lcd.print("Nilai sensor:");  
  
lcd.setCursor(4,1);  
lcd.print(terbaca);  
lcd.print(" ");  
lcd.print("ppm");  
}
```







## LAMPIRAN III

# Datasheet

