

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di zaman modern ini, orang-orang menghabiskan sebagian besar waktu mereka berada di dalam ruangan seperti di rumah, kantor, sekolah, ataupun di dalam kendaraan. Oleh karena itu, kualitas udara di dalam ruangan (*Indoor Air Quality*) merupakan masalah yang perlu mendapat perhatian karena akan berpengaruh terhadap kesehatan manusia terutama anak-anak, orang lanjut usia, dan orang-orang yang memiliki sistem pernafasan yang sensitif. Dampak penyakit yang terjadi karena buruknya kualitas udara seperti penyakit paru-paru (*bronchitis*), gangguan kesehatan pada hemoglobin (metalprotein pengangkut oksigen yang mengandung besi dalam sel darah merah) yang dapat terjadi jika menghirup zat kimia pada udara seperti karbon monoksida, penyakit ISPA (infeksi saluran pernapasan akut) termasuk diantaranya asma, bronkitis, dan gangguan pernapasan lainnya [1].

Bahan polutan di dalam rumah, tempat kerja, maupun dalam gedung yang merupakan tempat-tempat umum, kadarnya berbeda dengan bahan polutan di luar ruangan. Peningkatan kadar bahan polutan di dalam ruangan selain dapat berasal dari penetrasi polutan dari luar ruangan, dapat pula berasal dari sumber polutan di dalam ruangan, seperti asap rokok, asap yang berasal dari dapur dan kebocoran gas LPG atau pemakaian obat anti nyamuk. Sumber lain polutan di dalam ruangan adalah perlengkapan kerja seperti pakaian, sepatu, ataupun perlengkapan lainnya yang dibawa masuk ke dalam dari luar ruangan.

Dari latar belakang di atas, maka dibuatlah sistem pengendali kualitas udara dalam ruang dengan mengimplementasikan logika fuzzy untuk mengatur kecepatan kipas. Alasan penggunaan logika fuzzy adalah untuk penggunaan energi listrik yang lebih efisien dan tidak lagi menggunakan prinsip *on-off* seperti peralatan elektronik kebanyakan. Alasan lain mengapa digunakannya logika

fuzzy pada penelitian ini yaitu logika fuzzy sangat fleksibel dan memiliki toleransi terhadap data-data yang tidak tepat atau samar.

1.2 Rumusan Masalah

Dari uraian di atas dapat dirumuskan suatu permasalahan dari perencanaan ini adalah sebagai berikut:

1. Bagaimana cara merancang dan membuat alat pengendali kualitas udara dalam ruangan agar terhindar dari bahaya polutan?
2. Bagaimana mengimplementasikan algoritma cerdas pada sistem pengendali kualitas udara dalam ruangan berbasis Arduino Uno?
3. Membuat suatu rancangan sistem yang mampu mendeteksi tingkat kualitas udara berdasarkan dua indikator (dua sensor).

1.3 Batasan Masalah

Untuk membatasi pembahasan masalah yang terlalu luas, maka diberi batasan masalah pada penelitian ini, yaitu:

1. Indikator polutan pada kualitas udara dalam penelitian ini difokuskan pada polutan gas, asap, dan debu.
2. Perancangan dan pembuatan alat ini berbasis mikrokontroler Arduino Uno dan sensor yang digunakan adalah sensor MQ-2 untuk mendeteksi polutan gas kimia dan asap, dan sensor GP2Y1010AU0F untuk mendeteksi debu.
3. Implementasi logika fuzzy untuk mengatur kecepatan putar kipas DC pada alat ini.
4. Pengujian sistem dilakukan pada simulasi miniatur ruangan tertutup.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Tujuan dari penelitian ini antara lain :

1. Membangun sistem pengendali kualitas udara pada suatu ruang tertutup dengan dua sensor sebagai input dengan mikrokontroler Arduino Uno, dan kipas DC sebagai aktuator .
2. Merancang sistem kendali logika fuzzy untuk mengatur kecepatan putar kipas DC berdasarkan data input sensor.

1.4.2 Manfaat

Manfaat dari penelitian ini adalah sebagai salah satu contoh pengaplikasian dari Arduino sebagai mikrokontroler dan sistem pengendali kualitas udara dapat dijadikan dasar rancangan dasar untuk diimplementasikan dalam sistem nyata.

1.5 Metodologi Penelitian

Metodologi penelitian yang digunakan untuk mencapai keberhasilan tugas akhir ini berupa:

1. Studi Literatur, yaitu dengan mencari dan mengumpulkan kajian-kajian dan literatur-literatur yang berkaitan dengan penelitian ini, berupa artikel, buku referensi, jurnal penelitian terkait, dan sumber-sumber lainnya yang berhubungan.
2. Perancangan sistem, merancang sistem berdasarkan tujuan penelitian.
3. Pembuatan sistem, mengimplimentasikan sistem yang telah dirancang.
4. Pengujian sistem, menguji sistem yang telah dibuat.
5. Analisis hasil, menganalisis hasil yang telah diperoleh dari pengujian sistem yang telah dirancang.
6. Kesimpulan, memberikan kesimpulan berdasarkan hasil dan analisis yang telah dilakukan.

1.6 Sistematika Penulisan

Penulisan yang digunakan dalam laporan skripsi ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang permasalahan, perumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan laporan skripsi.

BAB II TINJAUAN PUSTAKA

Pada bab ini berisi tentang pengertian dan teori-teori penunjang yang dibutuhkan dalam pembuatan skripsi.

BAB III METODA PENELITIAN

Bab ini berisi tentang penjelasan mengenai bagaimana langkah proses penelitian dilakukan. Pada bagian ini juga menjelaskan bagaimana alur penelitian dilakukan yang dapat dibuat dalam bentuk flowchart atau diagram blok.

BAB IV DATA DAN ANALISIS

Bab ini berisi hasil pengujian dan analisis data hasil pengujian secara keseluruhan.

BAB V PENUTUP

Bab ini berisi kesimpulan yang dapat diambil berdasarkan analisis data hasil pengujian, serta saran-saran yang bersifat membangun untuk perbaikan lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Kualitas Udara dalam Ruangan

Menurut *National Health Medical Research Council* mendefinisikan udara dalam ruangan adalah udara yang berada dalam suatu ruang gedung yang ditempati oleh sekelompok orang yang memiliki tingkat kesehatan yang berbeda-beda selama minimal satu jam. Ruang gedung yang dimaksud dalam pengertian ini meliputi sekolah, restoran, rumah, gedung untuk umum, hotel, rumah sakit, dan perkantoran, tidak termasuk tempat kerja atau tempat-tempat yang mengacu pada standart kesehatan kerja [2].

Faktor terbesar penyebab buruknya kualitas udara dalam ruangan berdasarkan jurnal yang berjudul “Kualitas Udara dalam Ruang” tulisan Rina Prihatmanti (2016) adalah ketidak lancarannya sirkulasi udara dan adanya zat polutan. Polutan sendiri dibagi menjadi tiga jenis besar, yaitu:

1. Polutan Kimia

Polutan kimia merupakan polutan yang bersumber pada baha kimia, seperti karbon dioksida, karbon monoksida, dan lain-lain. Polutan kimia ini banyak yang berwujud gas dan sangat rignan, umumnya disebut dengan golongan *Volatile Organic Compound* (VOC). VOC ini merupakan zat kimia yang berbahaya dan mempunyai efek buruk jangka pendek dan jangka panjang.

2. Polutan Fisik

Polutan fisik ini berupa debu dan asap, termasuk asap kendaraan bermotor. Secara umum terdapat 2 jenis ukuran polutan fisik, PM10 dan PM2,5. *Particulate Matter* (PM)10 dan 2,5 menunjukkan ukuran diameter partikel dalam satuan mikron. Makin kecil ukuran partikelnya, makin dapat jauh masuk ke dalam paru-paru bila terhirup.

3. Polutan Biologis

Polutan Biologis ini berupa virus, bakteri maupun jamur. Jamur adalah masalah yang paling umum ditemui di dalam maupun luar ruangan. Masalah serius akan timbul bila jamur tumbuh di dalam ruangan yang lembab dan ventilasi udaranya kurang [3].

2.2 Arduino Uno

Arduino Uno adalah sebuah board yang menggunakan ATmega328. Arduino Uno memiliki 14 pin digital (6 pin dapat digunakan sebagai output PWM), 6 input analog, sebuah 16 MHz osilator kristal, sebuah koneksi USB, sebuah konektor sumber tegangan, sebuah header ICSP, dan sebuah tombol reset. Arduino Uno memuat segala hal yang dibutuhkan untuk mendukung sebuah mikrokontroler. Hanya dengan menghubungkannya ke sebuah komputer melalui USB atau memberikan tegangan DC dari baterai atau adaptor AC ke DC sudah dapat membuatnya bekerja. Arduino Uno menggunakan ATmega328 yang diprogram sebagai USB to serial converter untuk komunikasi serial ke komputer melalui port USB [4]. Board dari Arduino Uno dapat dilihat pada Gambar 2.1.



Gambar 2.1 Arduino Uno (Sumber : Datasheet, 2018)

Tabel 2.1 Spesifikasi Arduino Uno

Mikrokonktroler	Atmega328
Operasi Tegangan	5 Volt
Input tegangan	7 – 12 V (Rekomendasi)
Pin I/O Digital	14
Pin Analog	6
Arus DC tiap Pin I/O	20 mA
Arus DC ketika 3.3 mA	50 mA
Memori Flash	32 KB
SRAM	2 KB
EEPROM	1 KB
Kecepatan Clock	16 MHz

Tabel 2.1 diatas merupakan tabel spesifikasi dari Arduino Uno. Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber listrik dipilih secara otomatis. Catu daya eksternal (non-USB) dapat berasal dari AC – DC adaptor atau baterai. Arduino Uno dapat beroperasi pada pasokan daya dari 6 – 20 volt. Jika diberikan tegangan kurang dari 7 volt, pin 5V dapat menyuplai kurang dari 5 volt dan *board* mungkin tidak stabil. Jika menggunakan lebih dari 12 volt, regulator tegangan bisa panas dan merusak *board*. Sehingga rentang yang dianjurkan adalah 7 – 12 volt.

2.2.1 Input dan Output

Masing-masing dari 14 pin digital pada Arduino Uno dapat digunakan sebagai input atau output, menggunakan fungsi “pinMode(), digitalWrite(), dan digitalRead()” pada program. Mereka beroperasi di 5 volt, setiap pin dapat

memberikan atau menerima maksimum 40 mA. Selain itu, beberapa pin memiliki fungsi khusus :

- Serial pin 0 (Rx) dan pin 1 (Tx) : Pin 0 dan pin 1 digunakan untuk menerima (Rx) dan mengirimkan (Tx) data TTL serial.
- External Interupsi pada pin 2 dan pin 3 : Pin 2 dan pin 3 dapat dikonfigurasi untuk memicu interupsi pada nilai yang rendah, nilai yang naik atau turun, atau perubahan nilai.
- PWM pada pin 3, 5, 6, 9, dan 11 : Pin PWM ini menyediakan 8-bit output PMW (Pulse Width Module) dengan menggunakan fungsi “`analogWrite()`” pada program Arduino.
- SPI pada pin 10 (SS), pin 11 (MOSI), pin 12 (MISO), dan pin 13 (SCK) : Pin SPI digunakan untuk komunikasi SPI (Serial Peripheral Interface).
- LED pada pin 13 : Terdapat *built-in* lampu LED yang terhubung ke pin digital 13. Ketika pin diberikan nilai “HIGH”, maka lampu LED akan menyala, dan ketika pin 13 diberikan nilai “LOW”, maka lampu LED akan mati.

Selain 14 pin digital, terdapat 6 pin analog yang masing-masing menyediakan 10-bit resolusi, yaitu 1024 nilai berbeda. Pin analog ini mengukur dari 0 sampai 5 volt, namun batas nilai tersebut dapat ditingkatkan dengan menggunakan pin AREF dengan fungsi “`analogReference()`” pada program [4].

2.3 Sensor MQ-2

Sensor gas (MQ-2) ini digunakan untuk mendeteksi konsentrasi gas yang mudah terbakar di udara serta asap dan menghasilkan keluaran sebagai tegangan analog. Sensor gas asap (MQ-2) dapat diatur sensitivitasnya secara langsung. Sensor ini biasa digunakan untuk mendeteksi kebocoran gas baik pada skala rumah tangga maupun di industri. Gas yang dapat dideteksi oleh sensor MQ-2 ini diantaranya adalah LPG, *i-butane*, *propane*, *methane*, *alcohol*, *Hydrogen*, dan asap. Sensor gas MQ-2 dapat dilihat pada Gambar 2.2 dibawah ini.



Gambar 2.2 Sensor MQ-2 (Sumber : Datasheet)

Sensor asap MQ-2 ini pada dasarnya terdiri dari tabung alumunium yang dikelilingi oleh silikon dan di pusatnya ada elektroda yang terbuat dari aurum di mana ada elemen pemanasnya. Ketika terjadi proses pemanasan, kumparan akan dipanaskan sehingga SnO₂ keramik menjadi semikonduktor atau sebagai penghantar sehingga melepaskan elektron dan ketika asap dideteksi oleh sensor dan mencapai aurum elektroda, maka output sensor MQ-2 akan menghasilkan tegangan analog. Sensor MQ-2 ini memiliki 6 buah masukan yang terdiri dari tiga buah power supply (Vcc) sebesar +5 volt untuk mengaktifkan *heater* dan sensor, Vss (*Ground*), dan pin keluaran dari sensor tersebut.

Sensor MQ-2 dapat mengukur konsentrasi gas mudah terbakar dari 200 sampai 10.000 ppm. Dapat beroperasi pada suhu dari -20°C sampai 50°C dan mengkonsumsi arus kurang dari 150 mA pada 5V. Sensor MQ-2 terdapat 2 masukan tegangan yakni V_h dan V_c. V_h digunakan untuk tegangan pemanas (*heater*) internal dan V_c merupakan tegangan sumber serta memiliki keluaran yang menghasilkan tegangan berupa tegangan analog [5]. Berikut konfigurasi dari sensor MQ-2 :

1. Pin 1 merupakan *heater* internal yang terhubung dengan ground.
2. Pin 2 merupakan tegangan sumber (V_c) dimana V_c < 24 VDC.
3. Pin 3 (V_h) digunakan untuk tegangan pada pemanas (*heater* internal) dimana V_h = 5 VDC
4. Pin 4 merupakan output yang akan menghasilkan tegangan analog.

Spesifikasi sensor MQ-2 adalah sebagai berikut :

1. Catu daya pemanas : 5V AC/DC
2. Catu daya rangkaian : 5VDC
3. Range pengukuran : 200 – 5000 ppm untuk LPG, propane
300 – 5000 ppm untuk butane
300 – 5000 ppm untuk Hidrogen
5000 – 20000 ppm untuk methane
200 – 10.000 ppm untuk udara
4. *Output* : tegangan analog (perubahan tegangan)

4.4 Sensor GP2Y1010AU0F

Debu merupakan partikel yang memiliki ukuran diameter kurang dari 10 μm atau yang sering disebut *particulate matter* (PM10). Sensor yang dapat digunakan untuk mengukur debu adalah sensor GP2Y1010AU0F. Sensor GP2Y1010AU0F adalah sensor yang memanfaatkan hamburan cahaya atau disebut dengan sistem penginderaan optic. Sensor ini dilengkapi dengan LED dan fotodiode yang diatur secara diagonal. Sensor ini sangat efektif dalam mendeteksi partikel yang sangat halus seperti debu atau asap rokok, dan umumnya digunakan dalam sistem pembersih udara.

Pada bagian tengah sensor GP2Y1010AU0F terdapat lubang yang tembus dari bagian depan hingga ke bagian belakang. Lubang tersebut berdiameter 8 mm \pm 0.15 mm dengan kedalaman 18 mm. Pada bagian sisi lain dari lubang tersebut, terdapat sepasang sensor yang dilengkapi dengan lensa kolimator. Sensor tersebut terdiri atas sebuah *light emitting diode* sebagai sumber cahaya (*light source/transmitter*) dan sebuah photodiode sebagai penerima hamburan cahaya yang dipantulkan oleh debu pada tingkat intensitas tertentu. Cahaya yang diterima oleh photodiode kemudian diubah ke dalam bentuk sinyal listrik berupa nilai tegangan, dimana nilai tegangan ini bergantung pada seberapa besar intensitas

cahaya yang diterima oleh photodiode [6]. Bentuk fisik sensor GP2Y1010AU0F dapat dilihat pada Gambar 2.3 dibawah ini.

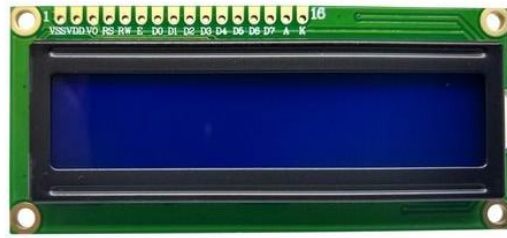


Gambar 2.3 Sensor GP2Y1010AU0F (Sumber: openimpulse.com)

2.5 LCD

LCD adalah salah satu komponen elektronika yang berguna untuk menampilkan suatu data, baik karakter, huruf, maupu grafik. Tampilan LCD sudah tersedia dalam bentuk modul, yaitu tampilan LCD beserta rangkaian pendukungnya termasuk ROM dan pelengkap lainnya. LCD mempunyai pin data, kontrol catu daya, dan pengatur kontras tampilan. LCD dapat bekerja dengan tegangan sebesar 5 volt yang didapat dari keluaran mikrokontroler, untuk itu biasanya LCD dihubungkan dengan mikrokontroler. LCD adalah modul penampil yang banyak digunakan karena tampilannya yang menarik. LCD yang paling banyak digunakan saat ini ialah LCD M16632 *refurbish*, karena harganya cukup murah. LCD M16632 merupakan modul LCD dengan tampilan 2x16 (2 baris x 16 kolom) dengan konsumsi daya didesain khusus untuk mengendalikan LCD.

LCD dibuat dari lapisan campuran organik antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan *seven segment* dan lapisan elektroda pada kaca belakang. Ketika elektroda diaktifkan dengan medan listrik (tegangan), molekul organik yang panjang dan silindris menyesuaikan diri dengan elektroda dari *segment* yang diaktifkan terlihat menjadi gelap dan membentuk karakter data yang ingin ditampilkan [7]. Bentuk LCD dapat dilihat pada Gambar 2.4 dibawah ini.



Gambar 2.4 LCD (Sumber: components101.com)

Pada LCD memiliki mikrokontroler sendiri yang digunakan untuk mengolah sinyal berupa bit data yang kemudian diolah untuk menampilkan data informasi baik berupa huruf, angka, kalimat maupun karakter. Selain itu juga terdapat memori didalamnya seperti :

- *CGROM (Character Generator Read Only Memory)* yaitu memori yang digunakan untuk menggambarkan pola sebuah karakter dimana pola tersebut merupakan karakter dasar yang sudah ditentukan secara permanen oleh pabrikan pembuat LCD tersebut sehingga pengguna tinggal mengambilnya sesuai alamat memorinya dan tidak dapat merubah karakter dasar yang ada dalam CGROM.
- *CGRAM (Character Generator Random Access Memory)* yaitu memori yang digunakan untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter itu dapat diubah-ubah sesuai dengan keinginan.
- *DDRAM (Display Data Random Access Memory)* yaitu memori tempat karakter yang akan ditampilkan.

Tidak hanya memori, namun LCD memiliki beberapa register dan pin-pin sebagai inputan. Berikut adalah register dan pin yang terdapat pada LCD.

- Register perintah merupakan sebuah register yang berisi instruksi-instruksi dari mikrokontroler ke panel LCD pada saat proses penulisan data.
- Register data merupakan register yang digunakan untuk menuliskan atau membaca data dari atau ke DDRAM. Penulisan data pada register

aka menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur sebelumnya.

Pin-pin yang terdapat pada LCD :

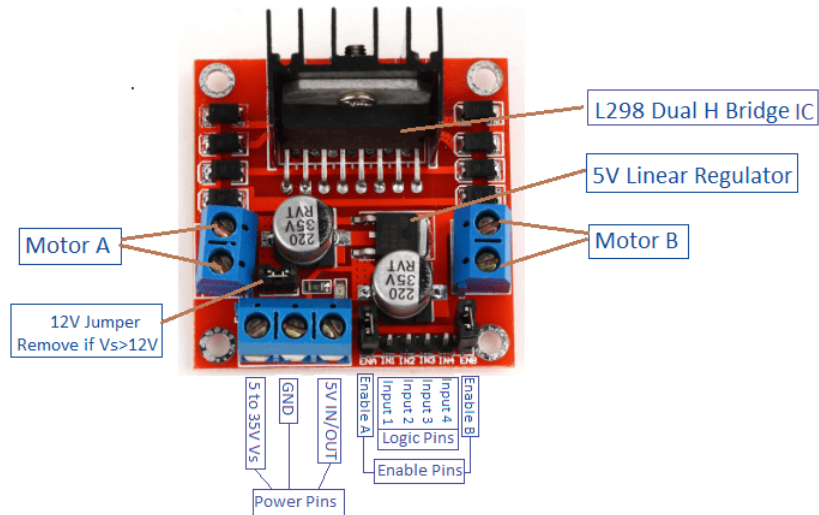
- Pin data yaitu sebuah pin yang digunakan sebagai jalur untuk mengirimkan data baik berupa huruf, angka, karakter, atau grafik untuk ditampilkan pada LCD dengan lebar data sebesar 8 bit.
- Pin RS (*Register Select*) berfungsi sebagai indikator atau yang menentukan jenis data yang masuk, apakah data atau perintah. Logika *low* menunjukkan yang masuk adalah perintah, sedangkan logika *high* menunjukkan data.
- Pin R/W (*Read / Write*) berfungsi sebagai instruksi pada modul jika *low* tulis data, sedangkan *high* untuk membaca data.
- Pin E (*Enable*) digunakan untuk memegang data baik masuk atau keluar.
- Pin VLCD berfungsi mengatur kecerahan tampilan (kontras) dimana pin ini dihubungkan dengan trimpot 5 atau 10 Kohm, jika tidak digunakan maka dihubungkan ke *ground* dan sumber tegangan yang dibutuhkan oleh LCD yaitu 5 volt DC (*Direct Current*).

2.7 Driver Motor L298N

Driver motor L298N merupakan module driver motor DC yang paling banyak digunakan atau dipakai di dunia elektronika yang difungsikan untuk mengontrol kecepatan serta arah perputaran motor DC.

IC L298 merupakan sebuah IC tipe H-bridge yang mampu mengedalikan beban-beban induktif seperti relay, solenoid, motor DC dan motor stepper. Pada IC L298 terdiri dari transistor-transistor logik (TTL) dengan gerbang NAND yang berfungsi untuk memudahkan dalam menentukan arah putaran suatu motor DC maupun motor stepper [8].

Untuk dipasaran sudah terdapat modul driver motor menggunakan IC L298 ini, sehingga lebih praktis dalam penggunaannya karena pin I/O nya sudah tersusun dengan rapi dan mudah digunakan. Kelebihan akan modul driver motor L298N ini yaitu dalam hal kepresisian dalam mengontrol motor sehingga motor lebih mudah untuk dikontrol. Adapun gambar pinout beserta keterangannya dapat diperhatikan pada Gambar 2.5 dibawah ini.



Gambar 2.5 Driver Motor L298 (Sumber: electronicshobbysts.com)

Keterangan :

- Enable A : Berfungsi untuk mengaktifkan bagian output motor A
- Enable B : Berfungsi untuk mengaktifkan bagian output motor B
- Jumper 12 V : Berfungsi sebagai mode pemilihan sumber tegangan 12V
- Logic Pin : Berfungsi sebagai kendali perputaran dan kecepatan motor yang dihubungkan ke mikrokontroler

Adapun untuk spesifikasi dari driver motor L298N dapat dijabarkan sebagai berikut :

- Menggunakan IC L298N (Double H-bridge Drive Chip)
- Tegangan minimal untuk masukan power antara 5 V – 35 V
- Tegangan operasional : 5 V

- Arus untuk masukan antara 0 – 36 mA
- Arus maksimal untuk keluaran per output A maupun output B yaitu 2 A
- Daya maksimal : 25 W
- Dimensi modul : 43 x 43 x 26 mm

2.6 Kipas DC 12V

Kipas DC merupakan sebuah kipas arus searah yang dipasang dengan tujuan untuk membersihkan udara ruangan dari polutan seperti asap, debu, dan polutan kimia, disamping itu kipas DC juga digunakan untuk menjaga kelembaban udara, dimana kipas DC akan membuat udara bergerak dan mengurangi tingkat kelembaban udara dalam ruangan tidak sampai melebihi dari *setting point* yang diinginkan. Bentuk fisik kipas DC dapat dilihat pada Gambar 2.6 dibawah ini.



Gambar 2.6 Kipas DC 12 volt

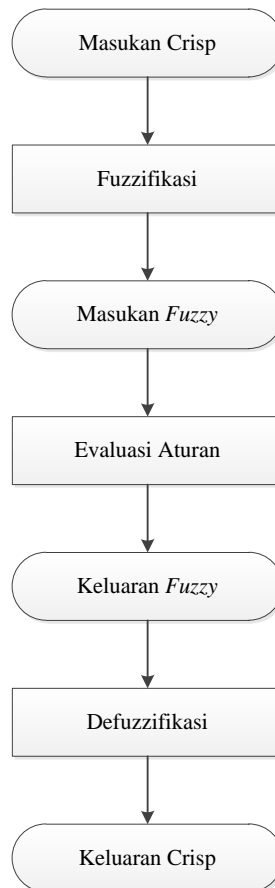
2.7 Logika Fuzzy

Fuzzy secara bahasa diartikan sebagai kabur atau samar yang artinya, suatu nilai dapat bernilai benar atau salah secara bersamaan. Dalam *fuzzy* dikenal derajat keanggotaan yang memiliki rentang nilai 0 (nol) hingga 1 (satu) yang menunjukkan sejauh mana suatu nilai benar dan sejauh mana suatu nilai itu salah. Logika fuzzy merupakan suatu logika yang memiliki nilai kekaburan atau kesamaran antara benar atau salah dan merupakan suatu cara yang tepat untuk memetakan ruang input ke dalam suatu ruang output dan mempunyai nilai kontinu.

Secara umum dalam sistem logika *fuzzy* terdapat empat buah elemen dasar, yaitu :

1. Basis kaidah (*rule base*), yang berisi aturan-aturan secara *linguistic* yang bersumber dari para pakar.
2. Proses fuzzifikasi, yang mengubah besaran tegas (*crisp*) ke besaran *fuzzy*.
3. Satu mekanisme pengambilan keputusan atau inferensi, yang memperagakan bagaimana para pakar mengambil suatu keputusan dengan menerapkan pengetahuan (*knowledge*).
4. Proses defuzzifikasi, yang mengubah besaran *fuzzy* dari mekanisme inferense, menjadi besaran tegas (*crisp*).

Untuk struktur dasar atau flowchart pengendalian *fuzzy* dapat dilihat pada Gambar 2.7 dibawah ini.



Gambar 2.7 Flowchart Logika Fuzzy

2.7.1 Fungsi Keanggotaan (*Member Function*)

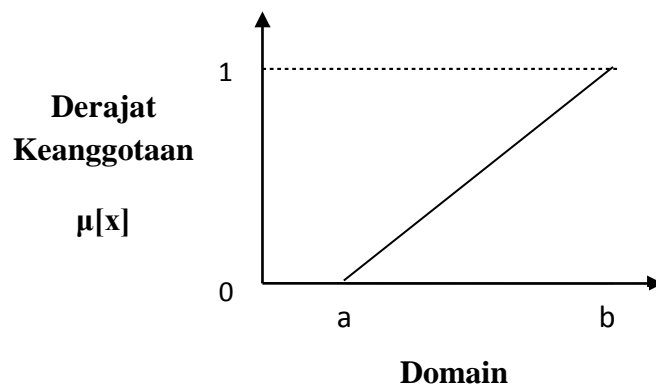
Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai *membership* adalah dengan pendekatan fungsi. Ada beberapa fungsi yang dapat digunakan, yaitu :

1. Representasi Linier Naik

Representasi linier naik digunakan untuk menghitung pemetaan kenaikan dari 0 (nol) ke 1 (satu), berikut fungsi keanggotaan linier naik :

$$\mu[x] = \begin{cases} 0; & x \leq a \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad (2.1)$$

Gambar 2.8 menunjukkan representasi linier naik dari variabel a yang bernilai 0 (nol) menuju variabel b yang bernilai 1 (satu).



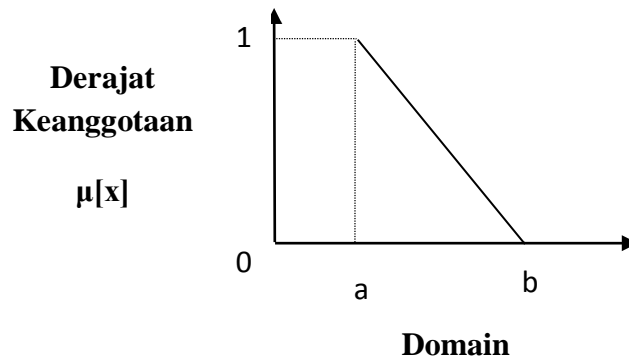
Gambar 2.8 Representasi Linier Naik

2. Representasi Linier Turun

Representasi linier turun digunakan untuk menghitung pemetaan penurunan dari 1 (satu) ke 0 (nol). Berikut rumus fungsi keanggotaan linier turun :

$$\mu[x] = \begin{cases} \frac{b-x}{b-a}; & a \leq x \leq b \\ 0; & x \geq b \end{cases} \quad (2.2)$$

Gambar 2.9 menunjukkan representasi linier turun dari variabel a yang bernilai 1 (satu) menuju variabel b yang bernilai 0 (nol).



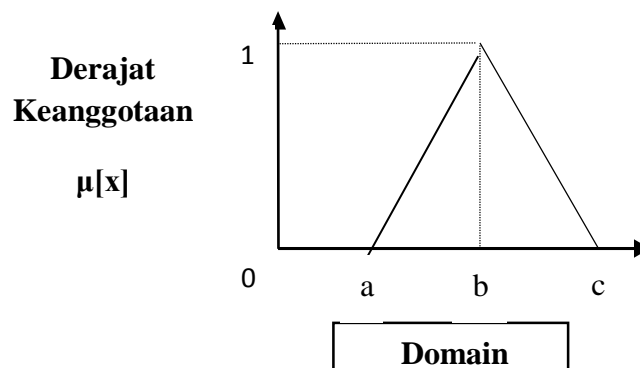
Gambar 2.9 Representasi linier turun

3. Representasi Kurva Segitiga

Representasi kurva segitiga adalah gabungan dari dua garis linier (naik dan turun). Berikut rumus fungsi keanggotaan kurva segitiga :

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ \frac{c-x}{c-b}; & b \leq x \leq c \end{cases} \quad (2.3)$$

Gambar 2.10 menunjukkan representasi kurva segitiga dari dua garis linier naik dan turun.



Gambar 2.10 Representasi Kurva Segitiga

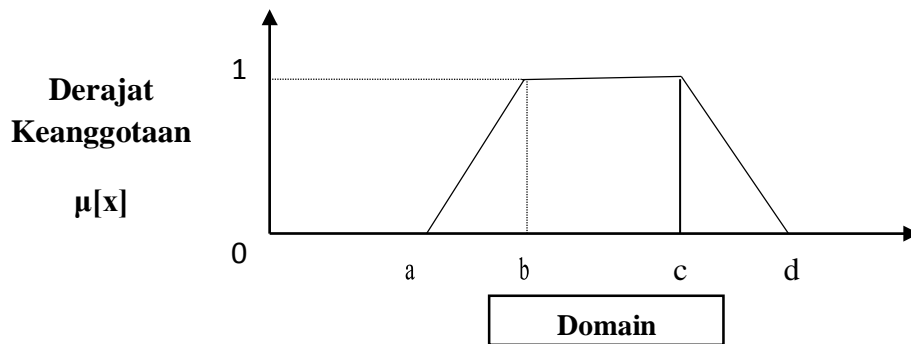
4. Representasi Kurva Trapesium

Representasi kurva trapesium adalah gabungan dari dua garis linier (naik dan turun) dan terdapat beberapa titik yang memiliki nilai keanggotaan 1 (satu).

Berikut rumus fungsi keanggotaan kurva trapesium :

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ 1; & b \leq x \leq c \\ \frac{d-x}{d-c}; & c \leq x \leq d \end{cases} \quad (2.4)$$

Gambar 2.11 menunjukkan representasi kurva trapesium dari 2 garis linier naik dan turun serta ada beberapa titik yang memiliki nilai keanggotaan 1.



Gambar 2.11 Representasi Kurva Trapesium

5. Representasi Kurva Bentuk Bahu

Representasi kurva bentuk bahu adalah daerah yang terletak di tengah-tengah suatu variabel yang dipresetasikan dalam bentuk segitiga, pada sisi kanan dan kiri akan naik dan turun. Himpunan *fuzzy* bahu, bukan segitiga, digunakan untuk mengakhiri variabel suatu daerah *fuzzy*, bahu kiri bergerak dari benar ke salah (dari nilai satu ke nol), sedangkan bahu kanan bergerak dari salah ke benar (dari nilai nol ke satu).

2.7.2 Fuzzifikasi

Fuzzifikasi yaitu suatu proses untuk mengubah suatu masukan dari bentuk tegas (*crisp*) menjadi *fuzzy* (variabel linguistik) yang biasanya disaikan dalam bentuk himpunan-himpunan *fuzzy* dengan suatu fungsi keanggotaannya masing-masing.

2.7.3 Basis Aturan (*Rule Base*)

Basis aturan berisi aturan-aturan *fuzzy* yang digunakan untuk pengendalian sistem. Aturan-aturan ini dibuat berdasarkan logika dan nalar manusia, serta berkaitan erat dengan jalan pikiran dan pengalaman pribadi yang membuatnya. Bisa dikatakan bahwa aturan ini bersifat subektif, tergantung dari ketajaman yang membuat. Aturan yang telah ditetapkan, digunakan untuk menghubungkan antara variabel masukan dan variabel keluaran.

Basis Aturan ini berbentuk “JIKA-MAKA” (IF-THEN), sebagai contoh adalah :

Aturan 1 : JIKA x adalah A1 DAN y adalah B1 MAKA z adalah C1

Aturan 2 : JIKA x adalah A2 DAN y adalah B1 MAKA z adalah C2

Aturan i : JIKA x adalah Ai DAN y adalah Bi MAKA z adalah Ci

2.7.4 Defuzzifikasi

Defuzzifikasi dapat didefinisikan sebagai proses perubahan besaran *fuzzy* yang disajikan dalam bentuk himpunan-himpunan *fuzzy* keluaran dengan fungsi keanggotaannya untuk mendapatkan kembali bentuk tegasnya (*crisp*). Hal ini diperlukan sebab dalam aplikasi dibutuhkan nilai tegas (*crisp*).

Ada beberapa metode defuzzifikasi yang bisa dipakai pada komposisi aturan mamdani, antara lain:

1. Metode Centroid

Metode centroid ini juga dikenal sebagai metode COA (*Centre of Area*) atau metode *centre of Gravity*. Pada metode ini, nilai tegas

keluarannya diperoleh berdasarkan titik berat kurva hasil proses pengambilan keputusan.

2. Metode Bisektor

Pada metode ini nilai tegasnya diperoleh dengan cara pengambilan nilai pada domain *fuzzy* yang memiliki keanggotaan setengah jumlah total nilai keanggotaan pada daerah *fuzzy*.

3. Metode MOM (Mean of Maximum)

Pada metode ini nilai tegas keluarannya diperoleh berdasarkan rata-rata semua aksi kontrol *fuzzy* yang mempunyai fungsi keanggotaan maksimum.

4. Metode LOM (Largest of Maximum)

Pada metode ini, nilai tegas keluarannya diperoleh berdasarkan tingkat keanggotaan terbesar.

5. Metode SOM (Smallest of Maximum)

Pada metode ini, nilai tegas keluarannya diperoleh berdasarkan tingkat keanggotaan terkecil.

6. Metode Weighted Average

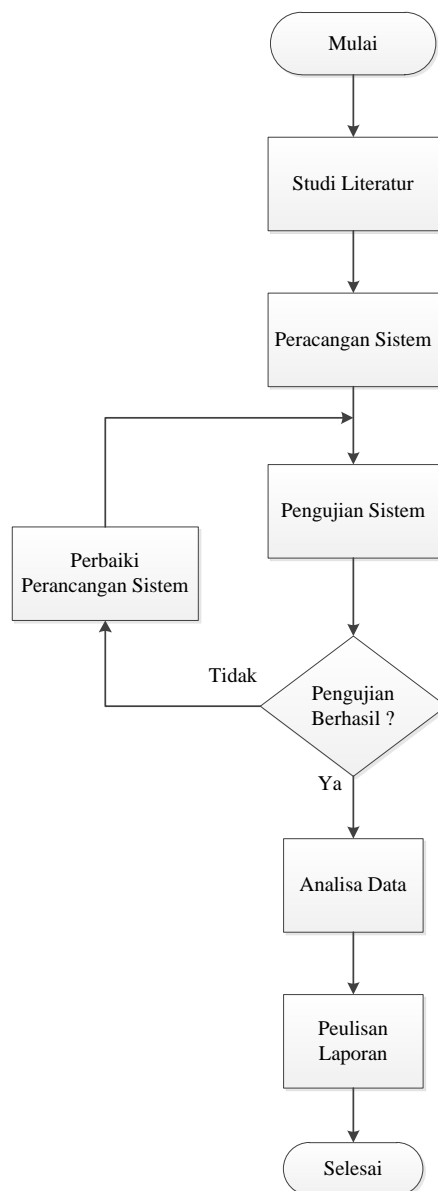
Pada metode ini, nilai tegas *output* diperoleh dengan mengambil nilai rata-rata dengan menggunakan pembobotan berupa derajat keanggotaan.

BAB III

METODE PENELITIAN

3.1 Metode Penelitian

Dalam merancang sistem pengendali kualitas udara dalam ruangan ini memiliki beberapa tahapan dalam pelaksanaannya dan berikut tahapan perancangannya akan digambarkan dalam bentuk flowchart pada Gambar 3.1.

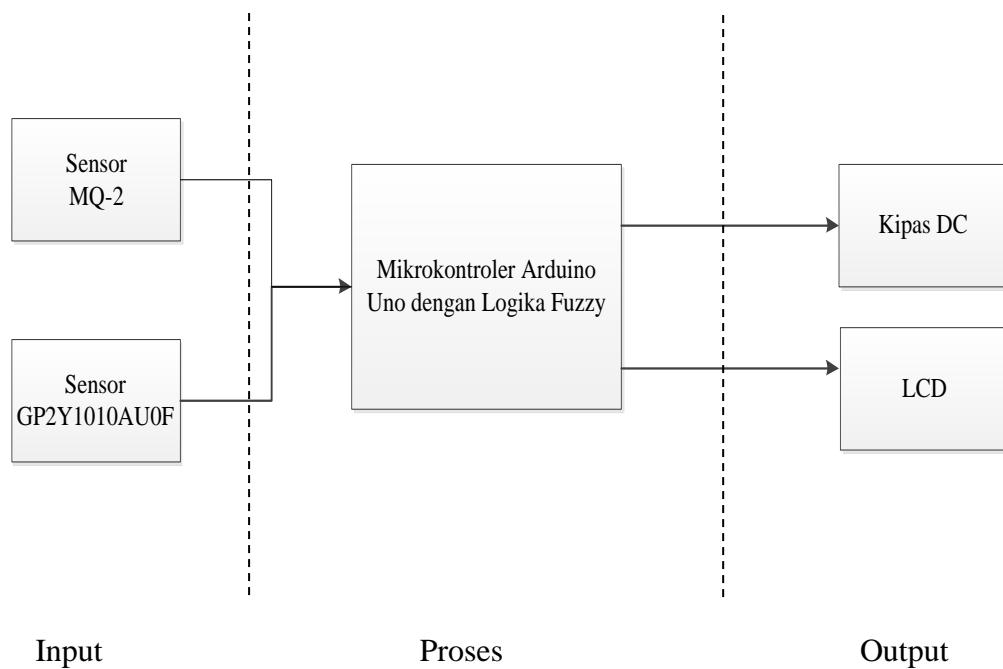


Gambar 3.1 Flowchart Rancangan Penelitian

Gambar 3.1 merupakan flowchart dari racangan penelitian yang dilakukan mulai dari studi literatur, perancangan sistem, pengujian sistem, kemudian menganalisa data dan diakhiri dengan penulisan laporan.

3.2 Perancangan Alat

Rancangan dari sistem pengendali kualitas udara dalam ruangan berbasis Arduino ini dibagi menjadi tiga, yaitu input, proses, dan output yang dapat dilihat pada Gambar 3.2.

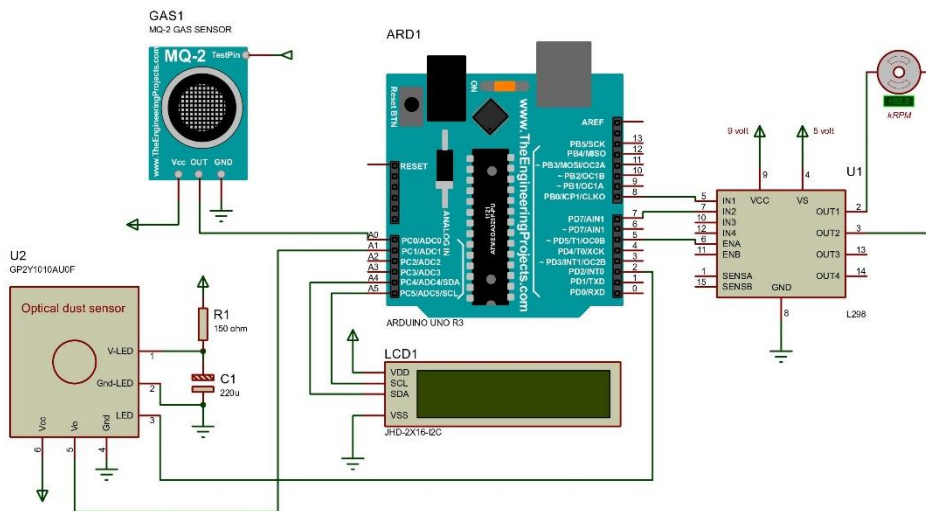


Gambar 3.2 Rancangan Alat

Rancangan alat pada penelitian ini dibagi menjadi tiga bagian, yaitu bagian *input* (masukan), proses, dan *output* (keluaran). Bagian input atau masukan merupakan bagian dari sistem yang melakukan kontak langsung dengan objek penelitian, yang mana dalam penelitian ini yang menjadi objek penelitian adalah kualitas udara pada ruangan dan inputnya adalah dua jenis sensor yaitu, sensor MQ-2 untuk mendeteksi polutan kimia dan asap, dan sensor GP2Y1010AU0F untuk mendeteksi debu.

Bagian proses merupakan bagian dari alat yang berupa mikrokontroler yang menerima data dari bagian input dan memprosesnya untuk menentukan kerja dari aktuator pada bagian output. Mikrokontroler yang digunakan pada sistem ini berupa Arduino Uno. Pada penelitian ini akan menggunakan sistem kedali logika fuzzy yang akan dimasukkan ke dalam program Arduino, agar dapat mengatur kecepatan putar dari kipas DC.

Bagian output atau keluaran merupakan bagian yang dikendalikan oleh mikrokontroler dan biasa disebut dengan aktuator. Pada penelitian ini aktuator yang digunakan berupa kipas DC 12 volt yang fungsinya untuk membuang udara kotor yang ada di dalam ruangan dan membuat sirkulasi udara sehingga udara di dalam ruangan akan bersih dan kelembaban terjaga. Skematik racangan sistem dapat dilihat pada Gambar 3.3 dibawah ini.



Gambar 3.3 Skematik Rangkaian Sistem

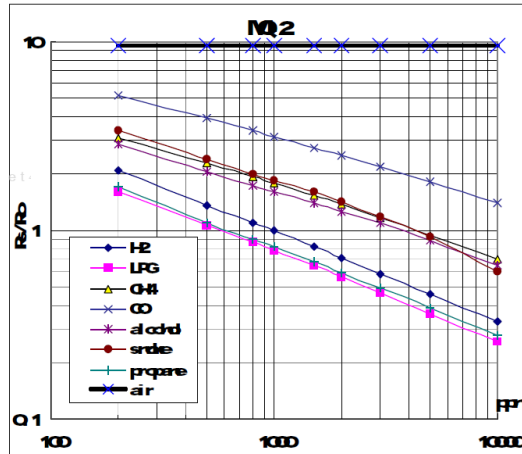
3.2.1 Perancangan Bagian Input

Bagian input pada penelitian ini menggunakan dua sensor yaitu sensor gas MQ-2 dan sensor debu. Karena output dari sensor berupa tegangan analog, sehingga diperlukan untuk dilakukan kalibrasi dan mengkonversi dari tegangan analog menjadi nilai satuan (ppm dan mg/m^3). kalibrasi dilakukan dengan rumus

persamaan yang dibuat berdasarkan datasheet sensor yang digunakan dan referensi dari jurnal yang menjadi acuan.

1. Kalibrasi Sensor Gas MQ-2

Kalibrasi Sensor Gas ini mengacu pada grafik pada datasheet berikut :



Gambar 3.4 Grafik Karakteristik Sensor MQ-2 (Sumber: Datasheet, 2018)

Dari grafik di atas, dapat diketahui rentang pengukuran sensor yaitu 200 sampai 10.000 ppm, sehingga untuk mengkalibrasi sensor menggunakan persamaan berikut :

$$\begin{aligned} \text{Rentang Pengukuran} &= | 300 - 10.000 | = 9700 \\ \text{ADC 10-bit} &= 1024 \end{aligned}$$

Persamaan :

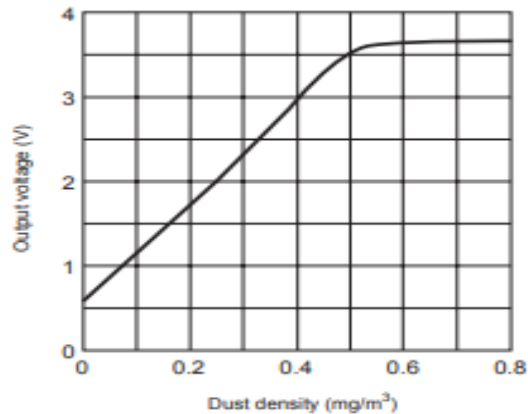
$$\text{kadarAsap} = \left(\text{outputSensor} \times \frac{\text{Rentang Pengukuran}}{\text{ADC 10bit}} \right) + 300 \quad (3.1)$$

Maka :

$$\text{kadarAsap} = \left(\text{outputSensor} \times \frac{9700}{1023} \right) + 300 \quad (3.2)$$

2. Kalibrasi Sensor Debu GP2Y1010AU0F

Kalibrasi sensor debu mengacu pada grafik berikut :



Gambar 3.5 Grafik Sensor Debu GP2Y1010AU0F (Sumber: Datasheet)

Dari grafik di atas, dapat dilihat kenaikan tegangan pada grafik dimulai dari 0,6 volt, artinya jika tegangan output sensor di bawah 0,6 volt berarti sensor tidak membaca adanya debu, maka hasil pengukuran perlu kita kurangi 0,6. Kemudian setelah 0,6 volt, grafik cenderung naik sebanding dengan bertambahnya intensitas debu yang terbaca, dimana naiknya garis pada grafik merupakan garis lurus. Dengan persamaan garis lurus (persamaan garis linear) diperlukan konstanta agar suatu koordinat dapat dicari berdasarkan nilai koordinat lain, yang dalam hal ini adalah agar kepadatan debu dapat dicari berdasarkan tegangan *output* sensor. Persamaan untuk mencari konstantanya adalah sebagai berikut :

$$\text{Rentang pembacaan debu} = |0 - 0,5| = 0,5$$

$$\text{Rentang pembacaan output voltage} = |0,6 - 3,5| = 2,9$$

Persamaan untuk konstanta :

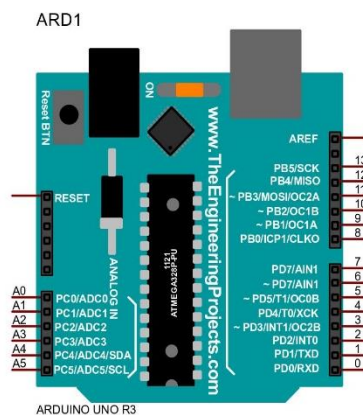
$$\text{Konstanta} = \frac{0,5}{2,9} = 0,17 \quad (3.3)$$

Setelah mendapat nilai konstanta, maka perhitungan untuk mencari nilai kepadatan debu adalah sebagai berikut :

$$\text{dustDensity} = \left(\frac{\text{outputSensor} \times 5}{1023} \right) \times 0.17 \quad (3.4)$$

3.2.2 Perancangan Bagian Pengendali

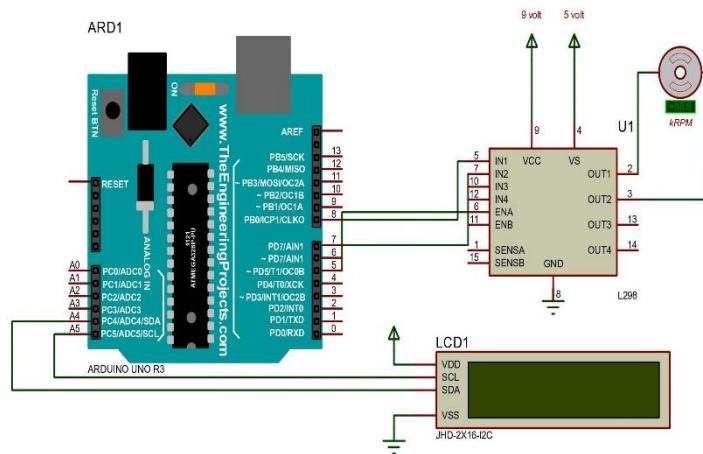
Bagian pemroses pada penelitian ini adalah mikrokontroler Arduino Uno. Digunakannya Arduino Uno ini karena mudah dipelajari karena menggunakan bahasa pemrograman bahasa C yang cenderung sederhana. Rangkaian pengendali ini akan membaca input asap dan debu pada pin A0 dan pin A1 dan outputnya untuk menyalakan kipas pada pin 7 dan pin 6 untuk keluaran berupa PWM (*Pulse Width Module*) untuk mengatur kecepatan putar kipas. Sedangkan untuk menampilkan hasil pengukuran polutan yang terbaca di LCD menggunakan pin A4 dan A5. Rancangan bagian pengendali dapat dilihat pada Gambar 3.6 dibawah ini.



Gambar 3.6 Rancangan Bagian Pengendali

3.2.3 Perancangan Bagian Output

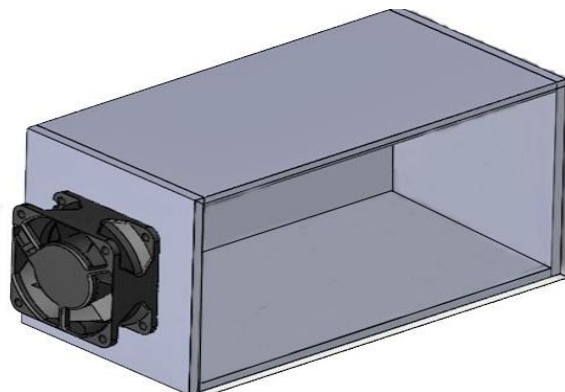
Bagian Output yang digunakan pada penelitian ini yaitu LCD sebagai penampil hasil pengukuran sensor dan kipas DC sebagai *exhaust* yang akan mengeluarkan polutan udara dari dalam miniatur ruang tertutup. Skematik rancangan bagian output dapat dilihat pada Gambar 3.7 dibawah ini.



Gambar 3.7 Rancangan Bagian Output

3.2.4 Desain Miniatur Ruang

Desain mekanik pada penelitian ini menggunakan sebuah kotak dengan ukuran 50 x 25 x 21 cm , yang digunakan sebagai model miniatur ruangan untuk melakukan pengujian pada sistem yang dibangun. Seperti pada Gambar 3.8 di bawah ini.

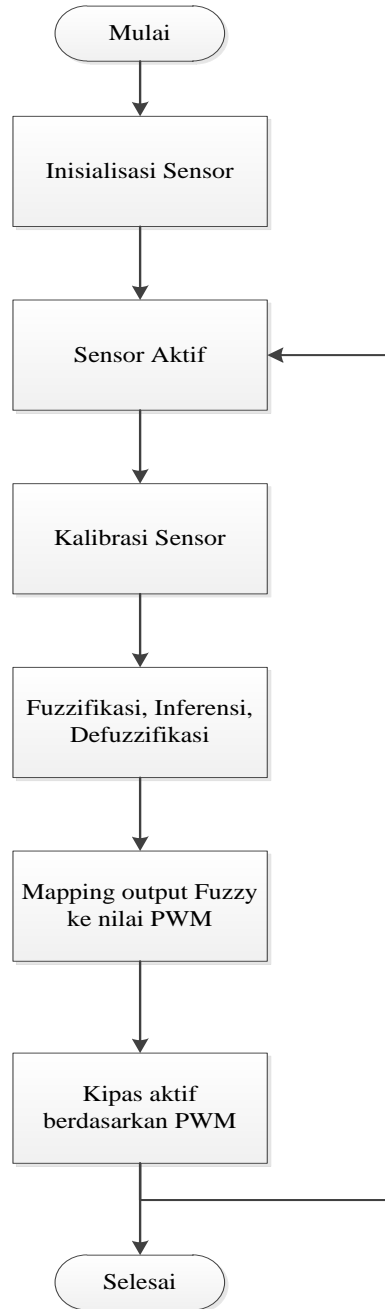


Gambar 3.8 Model Miniatur Ruang

Pada pembuatan miniatur ruang ini digunakan akrilik sebagai *case* dari sistem yang dibuat. Alasan digunakannya akrilik sendiri karena jenis bahan yang ringan namun kuat untuk digunakan sebagai miniatur ruangan yang dapat menampung komponen-komponen yang digunakan, dan juga karena akrilik yang

digunakan berwarna bening transparan sehingga dapat melihat keadaan di dalam dan dapat dengan mudah melihat respon dari sistem saat dilakukan pengujian.

3.3 Perancangan Program

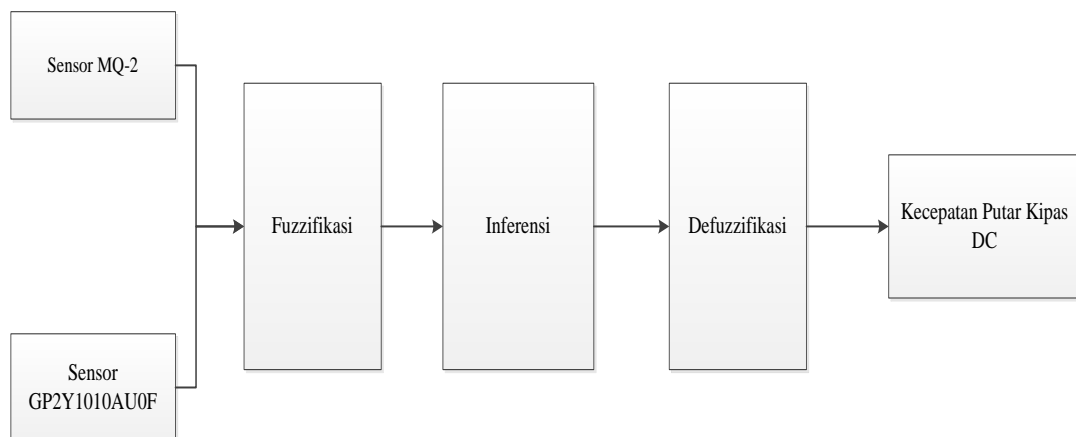


Gambar 3.9 Flowchart Program

Gambar 3.9 di atas merupakan gambaran besar dari program yang akan dibuat untuk dimasukkan ke dalam Arduino. Dari flowchart di atas, ketika Arduino menyala maka seluruh sistem akan dimulai dengan aktifnya sensor yang akan membaca kondisi udara, kemudian keluaran dari sensor yang masih berupa tegangan analog akan dikalibrasi dan dirubah ke dalam satuan konsentrasi ppm untuk sensor gas dan satuan mg/m^3 untuk sensor debu. Kemudian nilai keluaran dari sensor yang telah dikalibrasi akan diolah datanya menggunakan logika *fuzzy* dengan keluaran berupa nilai PWM (0 – 255). Nilai PWM yang didapat akan menentukan kecepatan putar dari kipas DC.

3.4 Perancangan Logika Fuzzy

Garis besar perancangan logika fuzzy ditunjukkan pada Gambar 3.3 berikut:

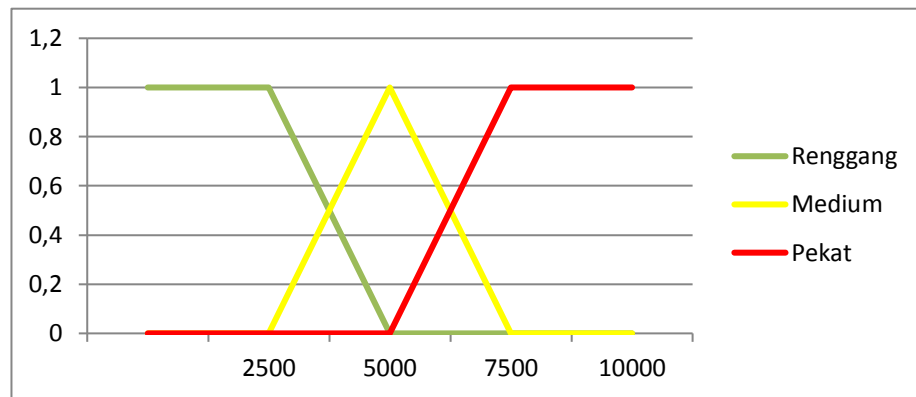


Gambar 3.10 Diagram Blok Logika Fuzzy

3.4.1 Fuzzifikasi

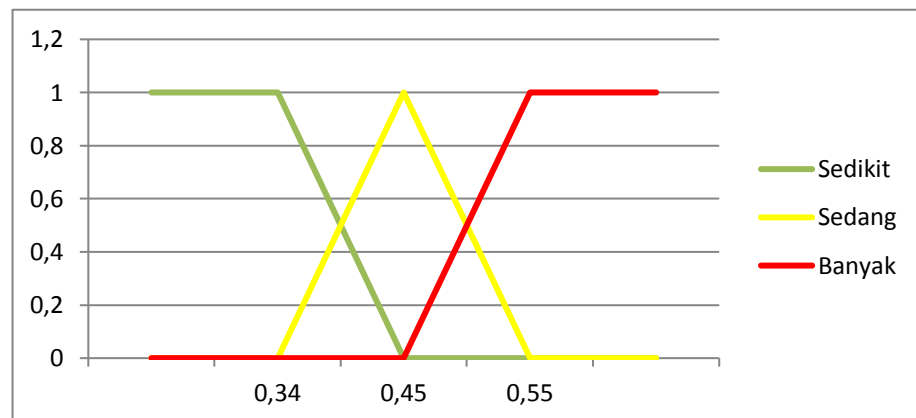
Pada proses fuzzifikasi, masukan tegas (*crisp*) yang berupa angka atau nilai akan dirubah menjadi variabel linguistik (*fuzzy*) atau bahasa manusia sehingga lebih mudah dipahami. Dalam penelitian ini, yang digunakan sebagai input adalah sensor gas MQ-2 dan sensor debu GP2Y1010AU0F. Sedangkan output berupa PWM yang digunakan untuk mengontrol kecepatan motor kipas.

- Fungsi keanggotaan dari *input* sensor gas



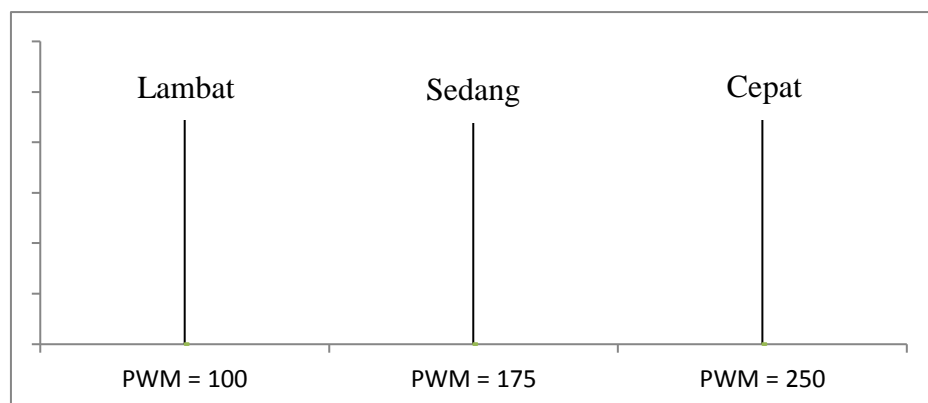
Gambar 3.11 Grafik Fungsi Keanggotaan dari Input Sensor Gas

- Fungsi keanggotaan dari *input* sensor debu



Gambar 3.12 Grafik Fungsi Keanggotaan dari Input Sensor Debu

- Fungsi keanggotaan dari *output* kecepatan kipas



Gambar 3.13 Grafik Fungsi Keanggotaan dari Output Kecepatan Kipas

3.4.2 Inferensi

Inferensi adalah proses penggabungan banyak aturan (*rule*) berdasarkan data yang tersedia. Hasil aturan dasar dari logika fuzzy dapat dilihat pada Tabel 3.1 dibawah ini.

Tabel 3.1 Rule Base

Asap	Debu		
	Sedikit	Sedang	Banyak
Renggang	Lambat	Lambat	Sedang
Medium	Lambat	Sedang	Cepat
Pekat	Sedang	Cepat	Cepat

Berdasar pada **Tabel 3.1** di atas, kemudian dibuat 9 aturan *fuzzy* yang berupa pernyataan kondisi dalam bentuk “JIKA-MAKA (IF-THEN)”, seperti di bawah ini :

1. Jika Debu Sedikit dan Asap Renggang, maka Kipas Lambat
2. Jika Debu Sedikit dan Asap Medium, maka Kipas Lambat
3. Jika Debu Sedikit dan Asap Pekat, maka Kipas Sedang
4. Jika Debu Sedang dan Asap Renggang, maka Kipas Lambat
5. Jika Debu Sedang dan Asap Medium, maka Kipas Sedang
6. Jika Debu Sedang dan Asap Pekat, maka Kipas Cepat
7. Jika Debu Banyak dan Asap Renggang, maka Kipas Sedang
8. Jika Debu Banyak dan Asap Medium, maka Kipas Cepat
9. Jika Debu Banyak dan Asap Pekat, maka Kipas Cepat

3.4.3 Defuzzifikasi

Proses defuzzifikasi merupakan proses terakhir dalam sistem *Fuzzy*. Proses defuzzifikasi merupakan sebuah proses perubahan data *input* yang telah dimasukkan dalam himpunan-himpunan *fuzzy* untuk mendapatkan kembali bentuk nilai tegasnya (*crisp*). Pada *fuzzy* ini, metode defuzzifikasi yang digunakan adalah *weight average*. Dimana nilai tegas *output* diperoleh dengan mengambil nilai rata-

rata dengan menggunakan pembobotan berupa derajat keanggotaan. Dengan menggunakan persamaan berikut :

Metode *Weighted Average* :

$$y^* = \frac{\sum_{i=1}^n ai \times ki}{\sum_{i=1}^n ai} \quad (3.5)$$

Dengan :

y^* = Nilai Defuzzifikasi (nilai crisp)

ai = Derajat keanggotaan (α – Hasil) ke-i

ki = Nilai crisp (pada penelitian ini berupa output PWM)

BAB IV

HASIL DAN ANALISA

Pengujian sistem dilakukan untuk mengetahui apakah sistem yang dibangun telah bekerja seperti yang direncanakan. Jika belum, maka perlu dicari dimana letak kesalahan atau kelemahannya untuk selanjutnya dilakukan perbaikan. Jika telah tercapai maka yang diperlukan selanjutnya adalah pengembangan untuk mendapat sistem yang lebih handal.

Analisis sistem dilakukan untuk mencari kelebihan dan kekurangan sistem yang telah dibuat. Dengan menganalisis sistem yang dibuat maka akan didapat kesimpulan mengenai tingkat efektifitas dan efisiensi sistem yang kita bangun.

4.1 Pengujian Sensor Gas MQ-2

Pengujian dilakukan untuk mengetahui sensor yang digunakan dapat bekerja dengan baik. Namun sebelum itu perlu dilakukan kalibrasi untuk merubah hasil keluaran sensor menjadi nilai PPM (*Part per Million*). Kalibrasi dilakukan dengan perhitungan matematis seperti dibawah ini.

Perhitungan Matematis :

$$\text{Sample outputSensor} = 512$$

$$\text{kadarAsap} = \left(\text{outputSensor} \times \frac{9700}{1023} \right) + 300$$

$$\text{kadarAsap} = \left(512 \times \frac{9700}{1023} \right) + 300$$

$$\text{kadarAsap} = 5154.74$$

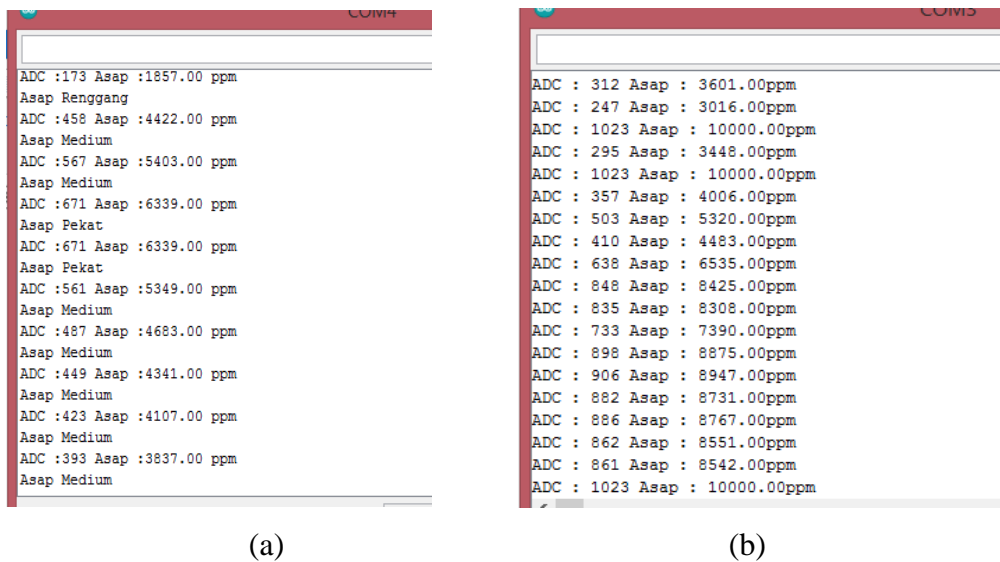
```
void loop(){  
  
  outMeasured = analogRead(pinAsap);  
  kadarAsap   =(outMeasured * (9700 / 1023)) + 300;
```

Gambar 4.1 Program Kalibrasi Sensor Gas MQ-2

Tabel 4.1 Hasil Pengujian Kalibrasi Sensor MQ-2

No.	ADC	Volt	PPM
1	1	0.004 volt	309.48
2	2	0.009 volt	318.96
3	4	0.01 volt	337.92
4	8	0.03 volt	375.85
5	16	0.07 volt	451.71
6	32	0.15 volt	603.42
7	64	0.31 volt	906.84
8	128	0.62 volt	1513.68
9	256	1.25 volt	2727.37
10	512	2.50 volt	5154.74
11	768	3.75 volt	7582.11
12	896	4.37 volt	8795.79
13	960	4.69 volt	9402.63
14	992	4.84 volt	9706.06
15	1008	4.92 volt	9857.77
16	1016	4.96 volt	9933.62
17	1020	4.98 volt	9971.55
18	1023	5 volt	10000

Setelah kalibrasi, pengujian sensor MQ-2 dilakukan dengan memasukkan asap ke dalam miniatur ruangan. Hasil pengujian sensor gas MQ-2 dapat dilihat pada Gambar 4.2.



Gambar 4.2 (a) Hasil Pengujian Sensor MQ-2 (b) Pengukuran Maksimal Sensor

Pengukuran maksimal sensor gas MQ-2 dapat mencapai 10.000 ppm dengan pengujian menggunakan gas LPG yang disorotkan langsung ke arah sensor. Dengan begitu hasil pengujian untuk pengukuran maksimal sensor sesuai dengan hasil perhitungan kalibrasi sensor.

4.2 Pengujian Sensor Debu

Pengujian dilakukan untuk mengetahui sensor yang digunakan dapat bekerja dengan baik. Namun sebelum itu perlu dilakukan kalibrasi untuk merubah hasil keluaran sensor menjadi nilai dengan satuan mg/m^3 . Kalibrasi dilakukan dengan perhitungan matematis seperti di bawah ini.

Perhitungan matematis :

$$\text{Sample outputSensor} = 512$$

$$\text{dustDensity} = \left(\frac{\text{outputSensor} \times 5}{1023} \right) \times 0.17$$

$$\text{dustDensity} = \left(\frac{512 \times 5}{1023} \right) \times 0.17$$

$$\text{dustDensity} = 0.425$$

```
void loop() {
  digitalWrite(ledPower,LOW); // power on the LED
  delayMicroseconds(samplingTime);

  voMeasured = analogRead(measurePin); // read the dust value
  digitalWrite(ledPower,HIGH); // turn the LED off
  delayMicroseconds(sleepTime);

  calcVoltage = ( ( voMeasured * 5 ) / 1023);
  dustDensity = calcVoltage * 0.17;
}
```

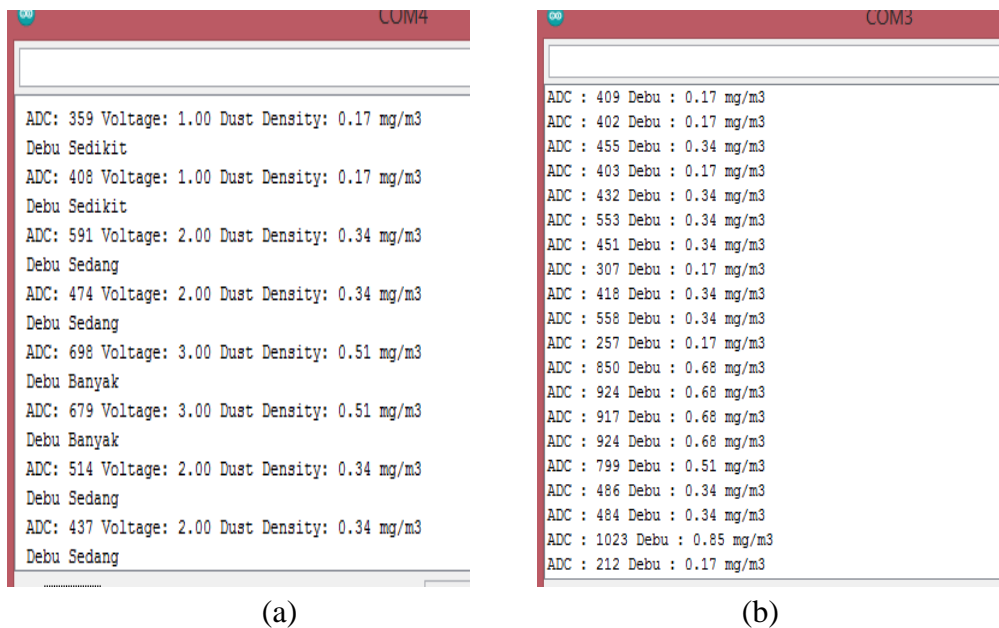
Gambar 4.3 Program Kalibrasi Sensor Debu

Tabel 4.2 Hasil Pengujian Kalibrasi Sensor Debu

No.	ADC	Volt	mg/m^3
1	1	0.005 volt	0.00083
2	2	0.009 volt	0.0016
3	4	0.02 volt	0.003
4	8	0.04 volt	0.006
5	16	0.07 volt	0.011

6	32	0.15 volt	0.025
7	64	0.31 volt	0.053
8	128	0.62 volt	0.105
9	256	1.25 volt	0.212
10	512	2.50 volt	0.425
11	768	3.75 volt	0.637
12	896	4.37 volt	0.742
13	960	4.69 volt	0.797
14	992	4.84 volt	0.822
15	1008	4.92 volt	0.836
16	1016	4.96 volt	0.843
17	1020	4.98 volt	0.846
18	1023	5 volt	0.85

Pengujian sensor ini dilakukan dengan debu dan asap untuk menguji apakah dengan bahan tersebut bisa terdeteksi oleh sensor debu ini atau tidak terdeteksi dan membandingkan hasil pengukuran sensor dengan penghitungan kalibrasi sehingga dapat diketahui sensor telah bekerja dengan baik. Pada pengukuran maksimal sensor debu dapat mendeteksi hingga 0,85 mg/m³ dengan begitu hasil perhitungan sesuai dengan hasil pengujian. Hasil pengujian sensor debu dapat dilihat pada Gambar 4.4 dibawah ini.



Gambar 4.4 (a) Hasil Pengujian Sensor Debu (b) Pengukuran Maksimal Sensor

Perbandingan hasil dan perhitungan kalibrasi :

Hasil pengukuran sensor : - ADC = 437
 - Desitas Debu = 0,34

Perhitungan kalibrasi :

ADC = 437

$$\text{dustDensity} = \left(\frac{\text{outputSensor} \times 5}{1023} \right) \times 0.17$$

$$\text{dustDensity} = \left(\frac{437 \times 5}{1023} \right) \times 0.17$$

$$\text{dustDensity} = 0,36$$

Faktor Kesalahan :

$$\text{Error} = \frac{0.36 - 0.34}{0.36} \times 100\% = 5.5\%$$

Tabel 4.3 Hasil Pengujian Sensor Debu

ADC	Hasil Pengukuran		Error
	Sensor	Matematis	
474	0.34	0.39	12.80%
679	0.51	0.56	8.92%
437	0.34	0.36	5.55%

4.3 Perhitungan Logika Fuzzy

4.3.1 Fuzzifikasi

Proses fuzzifikasi ini merubah nilai keluaran sensor yang telah terkalibrasi menjadi variabel linguistik dengan menghimpun keanggotaan setiap parameter terukur yang akan diproses sesuai aturan fuzzy.

Fungsi Keanggotaan :

Sensor Gas MQ-2 :
 - Renggang (0 – 2500 ppm)
 - Medium (2501 – 7500 ppm)
 - Pekat (7500 – 10000 ppm)

Sensor Debu : - Sedikit (0 – 0,35 mg/m³)
 - Sedang (0,36 – 0,55 mg/m³)
 - Banyak (0,56 – 0,61 mg/m³)

Kipas DC : - Lambat (100 pwm)
 - Sedang (175 pwm)
 - Cepat (250 pwm)

4.3.2 Inferensi

Inferensi adalah proses penggabungan banyak aturan (*rule*) berdasarkan data yang tersedia.

Tabel 4.4 Rule Base

Asap	Debu		
	Sedikit	Sedang	Banyak
Renggang	Lambat	Lambat	Sedang
Medium	Lambat	Sedang	Cepat
Pekat	Sedang	Cepat	Cepat

Berdasar pada **Tabel 4.4** di atas, kemudian dibuat 9 aturan *fuzzy* yang berupa pernyataan kondisi dalam bentuk “JIKA-MAKA (IF-THEN)”, seperti **Tabel 4.5** bawah ini.

Tabel 4.5 Aturan Fuzzy

No.		Kondisi Debu		Kondisi Asap		Kondisi Kipas
R1	JIKA	Sedikit	DAN	Renggang	MAKA	Lambat
R2	JIKA	Sedikit	DAN	Medium	MAKA	Lambat
R3	JIKA	Sedikit	DAN	Pekat	MAKA	Sedang
R4	JIKA	Sedang	DAN	Renggang	MAKA	Lambat
R5	JIKA	Sedang	DAN	Medium	MAKA	Sedang
R6	JIKA	Sedang	DAN	Pekat	MAKA	Cepat
R7	JIKA	Banyak	DAN	Renggang	MAKA	Sedang
R8	JIKA	Banyak	DAN	Medium	MAKA	Cepat
R9	JIKA	Banyak	DAN	Pekat	MAKA	Cepat

Setelah mendapat aturan-aturan fuzzy, kemudian mencari nilai derajat keanggotaan dari aturan (*rule*) tersebut untuk digunakan pada proses defuzzifikasi. Untuk mencari nilai derajat keanggotaan, berikut persamaan dan perhitungannya.

$$\alpha - \text{Hasil} = \min(\mu_{\text{Kondisi_Debu}}[x] \cap \mu_{\text{Kondisi_Asap}}[x])$$

$$\mu_{\text{Debu_Sedikit}}[x] = \begin{cases} 1; & x \leq 0.34 \\ \frac{0.45-x}{0.45-0.34}; & 0.34 \leq x \leq 0.45 \\ 0; & x \geq 0.45 \end{cases}$$

$$\mu_{\text{Asap_Renggang}}[x] = \begin{cases} 1; & x \leq 2500 \\ \frac{5000-x}{5000-2500}; & 2500 \leq x \leq 5000 \\ 0; & x \geq 5000 \end{cases}$$

$$\mu_{\text{Debu_Sedang}}[x] = \begin{cases} 0; & x \leq 0.34 \text{ atau } x \geq 0.55 \\ \frac{x-0.34}{0.45-0.34}; & 0.34 \leq x \leq 0.45 \\ \frac{0.55-x}{0.55-0.45}; & 0.45 \leq x \leq 0.55 \end{cases}$$

$$\mu_{\text{Asap_Medium}}[x] = \begin{cases} 0; & x \leq 2500 \text{ atau } x \geq 7500 \\ \frac{x-2500}{5000-2500}; & 2500 \leq x \leq 5000 \\ \frac{7500-x}{7500-5000}; & 5000 \leq x \leq 7500 \end{cases}$$

$$\mu_{\text{Debu_Banyak}}[x] = \begin{cases} 0; & x \leq 0.45 \\ \frac{x-0.45}{0.55-0.45}; & 0.45 \leq x \leq 0.55 \\ 1; & x \geq 0.55 \end{cases}$$

$$\mu_{\text{Asap_Pekat}}[x] = \begin{cases} 0; & x \leq 5000 \\ \frac{x-5000}{7500-5000}; & 5000 \leq x \leq 7500 \\ 1; & x \geq 7500 \end{cases}$$

[R1] : Jika Debu Sedikit dan Asap Renggang, maka Kipas Lambat.

$$\alpha - \text{Hasil} = \min (\mu_{\text{Debu_Sedikit}}[x] \cap \mu_{\text{Asap_Renggang}}[x])$$

Sample :

$\mu_{\text{Debu_Sedikit}}[0,3]$ dan $\mu_{\text{Asap_Renggang}}[1850]$

$$\begin{aligned}\alpha - \text{Hasil} &= \min (\mu_{\text{Debu_Sedikit}}[0,3] \cap \mu_{\text{Asap_Renggang}}[1850]) \\ &= \min (1 ; 1) \\ &= 1\end{aligned}$$

[R2] : Jika Debu Sedikit dan Asap Medium, maka Kipas Lambat.

$$\alpha - \text{Hasil} = \min (\mu_{\text{Debu_Sedikit}}[x] \cap \mu_{\text{Asap_Medium}}[x])$$

Sample :

$\mu_{\text{Debu_Sedikit}}[0,31]$ dan $\mu_{\text{Asap_Medium}}[4548]$

$$\begin{aligned}\alpha - \text{Hasil} &= \min (\mu_{\text{Debu_Sedikit}}[0,31] \cap \mu_{\text{Asap_Medium}}[4548]) \\ &= \min (1 ; 0,82) \\ &= 0,82\end{aligned}$$

[R3] : Jika Debu Sedikit dan Asap Pekat, maka Kipas Sedang.

$$\alpha - \text{Hasil} = \min (\mu_{\text{Debu_Sedikit}}[x] \cap \mu_{\text{Asap_Pekat}}[x])$$

Sample :

$\mu_{\text{Debu_Sedikit}}[0,3]$ dan $\mu_{\text{Asap_Pekat}}[7500]$

$$\begin{aligned}\alpha - \text{Hasil} &= \min (\mu_{\text{Debu_Sedikit}}[0,3] \cap \mu_{\text{Asap_Pekat}}[7500]) \\ &= \min (1 ; 1) \\ &= 1\end{aligned}$$

[R4] : Jika Debu Sedang dan Asap Renggang, maka Kipas Lambat.

$$\alpha - \text{Hasil} = \min (\mu_{\text{Debu_Sedang}}[x] \cap \mu_{\text{Asap_Renggang}}[x])$$

Sample :

$\mu_{\text{Debu_Sedang}}[0,48]$ dan $\mu_{\text{Asap_Renggang}}[1200]$

$$\begin{aligned} \alpha - \text{Hasil} &= \min (\mu_{\text{Debu_Sedang}}[0,48] \cap \mu_{\text{Asap_Renggang}}[1200]) \\ &= \min (0,7 ; 1) \\ &= 0,7 \end{aligned}$$

[R5] : Jika Debu Sedang dan Asap Medium, maka Kipas Sedang.

$$\alpha - \text{Hasil} = \min (\mu_{\text{Debu_Sedang}}[x] \cap \mu_{\text{Asap_Medium}}[x])$$

Sample :

$\mu_{\text{Debu_Sedang}}[0,5]$ dan $\mu_{\text{Asap_Medium}}[5000]$

$$\begin{aligned} \alpha - \text{Hasil} &= \min (\mu_{\text{Debu_Sedang}}[0,5] \cap \mu_{\text{Asap_Medium}}[5000]) \\ &= \min (0,5 ; 1) \\ &= 0,5 \end{aligned}$$

[R6] : Jika Debu Sedang dan Asap Pekat, maka Kipas Cepat.

$$\alpha - \text{Hasil} = \min (\mu_{\text{Debu_Sedang}}[x] \cap \mu_{\text{Asap_Pekat}}[x])$$

Sample :

$\mu_{\text{Debu_Sedang}}[0,51]$ dan $\mu_{\text{Asap_Pekat}}[7781]$

$$\begin{aligned} \alpha - \text{Hasil} &= \min (\mu_{\text{Debu_Sedang}}[0,51] \cap \mu_{\text{Asap_Pekat}}[7781]) \\ &= \min (0,4 ; 1) \\ &= 0,4 \end{aligned}$$

[R7] : Jika Debu Banyak dan Asap Renggang, maka Kipas Sedang.

$$\alpha - \text{Hasil} = \min (\mu_{\text{Debu_Banyak}}[x] \cap \mu_{\text{Asap_Renggang}}[x])$$

Sample :

$\mu_{\text{Debu_Banyak}}[0,6]$ dan $\mu_{\text{Asap_Renggang}}[1060]$

$$\alpha - \text{Hasil} = \min (\mu_{\text{Debu_Banyak}}[0,6] \cap \mu_{\text{Asap_Renggang}}[1056])$$

$$= \min (1 ; 1)$$

$$= 1$$

[R8] : Jika Debu Banyak dan Asap Medium, maka Kipas Cepat.

$$\alpha - \text{Hasil} = \min (\mu_{\text{Debu_Banyak}}[x] \cap \mu_{\text{Asap_Medium}}[x])$$

Sample :

$\mu_{\text{Debu_Banyak}}[0,6]$ dan $\mu_{\text{Asap_Medium}}[5576]$

$$\alpha - \text{Hasil} = \min (\mu_{\text{Debu_Banyak}}[0,6] \cap \mu_{\text{Asap_Medium}}[5576])$$

$$= \min (1 ; 0,76)$$

$$= 0,76$$

[R9] : Jika Debu Banyak dan Asap Pekat, maka Kipas Cepat.

$$\alpha - \text{Hasil} = \min (\mu_{\text{Debu_Banyak}}[x] \cap \mu_{\text{Asap_Pekat}}[x])$$

Sample :

$\mu_{\text{Debu_Banyak}}[0,65]$ dan $\mu_{\text{Asap_Pekat}}[7000]$

$$\alpha - \text{Hasil} = \min (\mu_{\text{Debu_Banyak}}[0,65] \cap \mu_{\text{Asap_Pekat}}[7000])$$

$$= \min (1 ; 0,8)$$

$$= 0.8$$

4.3.3 Defuzzifikasi

Input dari proses defuzzifikasi adalah suatu logika fuzzy yang diperoleh dari komposisi aturan fuzzy, sedangkan *output* yang dihasilkan merupakan nilai tegas yang pada penelitian ini adalah nilai PWM (*Pulse Width Modulation*) untuk mengatur kecepatan kipas dc.

Metode *Weighted Average* :

$$y^* = \frac{\sum_{i=1}^n a_i \times k_i}{\sum_{i=1}^n a_i}$$

Dengan :

y^* = Nilai Defuzzifikasi (nilai crisp)

a_i = Derajat keanggotaan (α – Hasil) ke-i

k_i = Nilai crisp (pada penelitian ini berupa output PWM)

- Defuzzifikasi R1

$$y^* = \frac{(a_1 * PWM 1) + (a_2 * PWM 1) + (a_3 * PWM 2) + (a_4 * PWM 1) + (a_5 * PWM 2) + \dots + (a_9 * PWM 3)}{a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9}$$

$$y^* = \frac{(1*100)+(0*100)+(0*175)+(0*100)+(0*175)+(0*250)+(0*175)+(0*250)+(0*250)}{1+0+0+0+0+0+0+0+0}$$

$$y^* = \frac{(1*100)}{1} = 100$$

- Defuzzifikasi R2

$$y^* = \frac{(a_1 * PWM 1) + (a_2 * PWM 1) + (a_3 * PWM 2) + (a_4 * PWM 1) + (a_5 * PWM 2) + \dots + (a_9 * PWM 3)}{a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9}$$

$$y^* = \frac{(0*100)+(0.82*100)+(0*175)+(0*100)+(0*175)+(0*250)+(0*175)+(0*250)+(0*250)}{0+0.82+0+0+0+0+0+0+0}$$

$$y^* = \frac{(0.82*100)}{0.82} = 100$$

- Defuzzifikasi R3

$$y^* = \frac{(a_1 * PWM 1) + (a_2 * PWM 1) + (a_3 * PWM 2) + (a_4 * PWM 1) + (a_5 * PWM 2) + \dots + (a_9 * PWM 3)}{a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9}$$

$$y^* = \frac{(0*100)+(0*100)+(1*175)+(0*100)+(0*175)+(0*250)+(0*175)+(0*250)+(0*250)}{0+0+1+0+0+0+0+0+0}$$

$$y^* = \frac{(1*175)}{1} = 175$$

- Defuzzifikasi R4

$$y^* = \frac{(a_1 * PWM 1) + (a_2 * PWM 1) + (a_3 * PWM 2) + (a_4 * PWM 1) + (a_5 * PWM 2) + \dots + (a_9 * PWM 3)}{a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9}$$

$$y^* = \frac{(0*100)+(0*100)+(0*175)+(0.7*100)+(0*175)+(0*250)+(0*175)+(0*250)+(0*250)}{0+0+0+0.7+0+0+0+0+0}$$

$$y^* = \frac{(1*175)}{1} = 175$$

- Defuzzifikasi R5

$$y^* = \frac{(a_1 * PWM 1) + (a_2 * PWM 1) + (a_3 * PWM 2) + (a_4 * PWM 1) + (a_5 * PWM 2) + \dots + (a_9 * PWM 3)}{a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9}$$

$$y^* = \frac{(0*100)+(0*100)+(0*175)+(0*100)+(0.5*175)+(0*250)+(0*175)+(0*250)+(0*250)}{0+0+0+0+0.5+0+0+0+0}$$

$$y^* = \frac{(0.5*175)}{0.5} = 175$$

- Defuzzifikasi R6

$$y^* = \frac{(a_1 * PWM 1) + (a_2 * PWM 1) + (a_3 * PWM 2) + (a_4 * PWM 1) + (a_5 * PWM 2) + \dots + (a_9 * PWM 3)}{a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9}$$

$$y^* = \frac{(0*100)+(0*100)+(0*175)+(0*100)+(0*175)+(0.4*250)+(0*175)+(0*250)+(0*250)}{0+0+0+0+0+0.4+0+0+0}$$

$$y^* = \frac{(0.4*250)}{0.4} = 250$$

- Defuzzifikasi R7

$$y^* = \frac{(a_1 * PWM 1) + (a_2 * PWM 1) + (a_3 * PWM 2) + (a_4 * PWM 1) + (a_5 * PWM 2) + \dots + (a_9 * PWM 3)}{a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9}$$

$$y^* = \frac{(0*100)+(0*100)+(0*175)+(0*100)+(0*175)+(0*250)+(1*175)+(0*250)+(0*250)}{0+0+0+0+0+0+1+0+0}$$

$$y^* = \frac{(1*175)}{1} = 175$$

- Defuzzifikasi R8

$$y^* = \frac{(a_1 * PWM 1) + (a_2 * PWM 1) + (a_3 * PWM 2) + (a_4 * PWM 1) + (a_5 * PWM 2) + \dots + (a_9 * PWM 3)}{a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9}$$

$$y^* = \frac{(0*100)+(0*100)+(0*175)+(0*100)+(0*175)+(0*250)+(0*175)+(0.76*250)+(0*250)}{0+0+0+0+0+0+0+0.76+0}$$

$$y^* = \frac{(0.76*250)}{0.76} = 250$$

- Defuzzifikasi R9

$$y^* = \frac{(a_1 * PWM 1) + (a_2 * PWM 1) + (a_3 * PWM 2) + (a_4 * PWM 1) + (a_5 * PWM 2) + \dots + (a_9 * PWM 3)}{a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9}$$

$$y^* = \frac{(0*100)+(0*100)+(0*175)+(0*100)+(0*175)+(0*250)+(0*175)+(0*250)+(0.8*250)}{0+0+0+0+0+0+0+0+0.8}$$

$$y^* = \frac{(0.8*250)}{0.8} = 250$$

4.4 Pengujian Keseluruhan Sistem

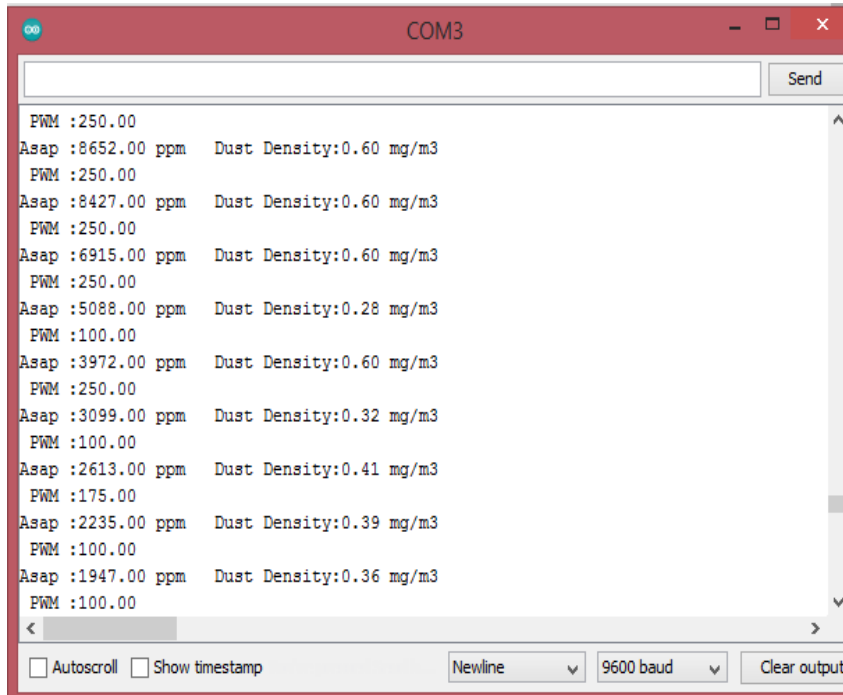
Pada sistem dilakukan pengujian kendali kipas DC dengan logika fuzzy. Pengujian dilakukan dengan membandingkan alat simulasi yang dibuat dengan mengimplementasikan logika fuzzy dan alat simulasi yang tanpa logika fuzzy atau dengan sistem *on-off*. Kemudian membandingkan waktu tanggapan (*response*

time) dari sistem dengan menguji gangguan pada sistem. Gangguan yang diuji pada pengujian ini adalah berupa polutan udara seperti asap.

Tabel 4.6 Hasil Pengujian Sistem

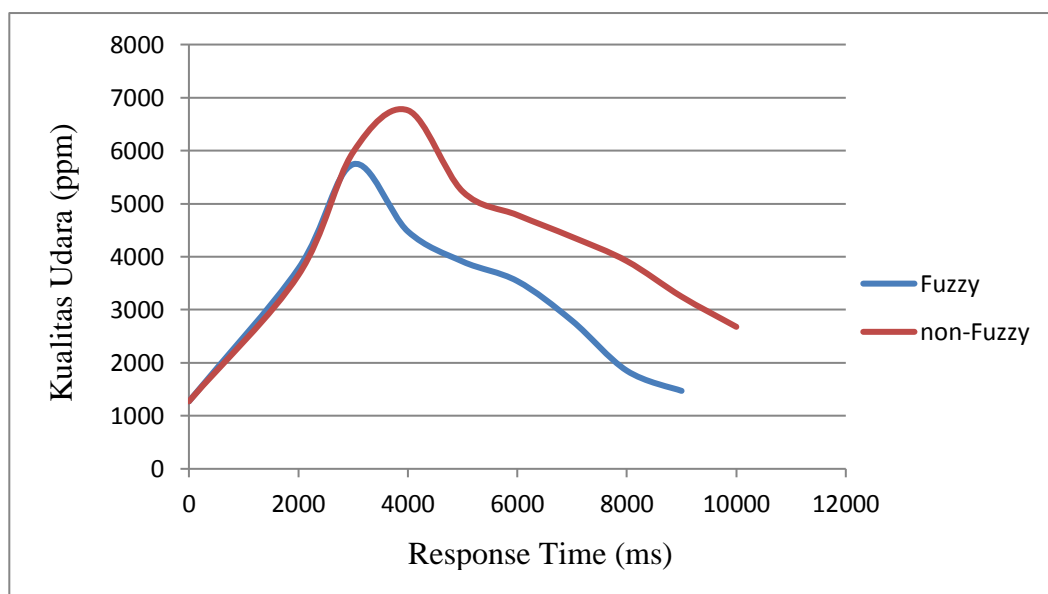
No.	Input / Nilai Sensor		Output / Aktuator	
	Gas (ppm)	Debu (mg/m ³)	non-Logika Fuzzy	Logika Fuzzy
1	822	0.3	0	100
2	4548	0.31	0	100
3	7500	0.3	0	175
4	1200	0.48	0	100
5	4818	0.5	255	175
6	7788	0.51	255	250
7	1056	0.6	255	175
8	5576	0.6	255	250
9	8454	0.6	255	250

Pada Tabel 4.6 diatas didapat hasil pengujian dimana pengendali kualitas udara dengan mengimplementasikan logika fuzzy dibandingkan dengan pengendali kualitas udara tanpa logika fuzzy (sistem *on-off*). Pada sistem tanpa logika fuzzy, *set-point* diatur pada nilai 6500 ppm untuk sensor gas dan 0,5 mg/m³ untuk sensor debu, sehingga saat polutan terdeteksi berada di nilai *set-point* atau telah melebihi batas nilai tersebut maka kipas dc akan menyala 255 pwm (kecepatan maksimal) dan setelah polutan yang terdeteksi berada dibawah nilai *set-point* kipas dc akan mati (0 pwm). Hal ini membuat sistem tanpa logika fuzzy tidak membersihkan polutan secara maksimal karena kipas dc telah berhenti berputar sebelum udara benar-benar bersih. Sedangkan sistem dengan logika fuzzy memiliki tiga tingkat kecepatan dari 100 pwm (lambat), 175 pwm (sedang), dan 250 pwm (cepat). Dimana sejak awal sistem dinyalakan kipas telah berputar 100 pwm dan kecepatan dapat bertambah seiring dengan bertambahnya polusi yang terdeteksi, sehingga mencegah pengumpulan polutan di dalam ruangan dan memaksimalkan saat pembersihan plusi udara tersebut. Sistem logika fuzzy ini bertujuan tidak hanya untuk menjaga kebersihan udara tetapi juga untuk menjaga sirkulasi udara tetap berjalan sehingga mengurangi tingkat kelembaban udara dalam ruangan.



Gambar 4.5 Tampilan Serial Monitor Hasil Pengujian

Pengujian selanjutnya adalah membandingkan waktu tanggapan (*response time*) dari sistem dengan memberi gangguan pada sistem. Sample gangguan yang diuji pada pengujian ini adalah berupa polutan udara seperti asap. Hasil pengujian ini dapat dilihat pada Gambar 4.6 dibawah ini.



Gambar 4.6 Grafik *Response Time* Sistem Terhadap Kualitas Udara

Hasil pengujian dapat dilihat pada Gambar 4.6 diatas menunjukkan respon sistem terhadap perubahan kualitas udara setelah diberi polutan berupa asap. Dapat dilihat bahwa respon sistem lebih cepat apabila menggunakan logika fuzzy dikarenakan sistem dengan implementasi logika fuzzy akan terus bekerja sedangkan pada sistem kendali tanpa fuzzy (sistem *on-off*), kipas dc (aktuator) hanya bekerja sesuai dengan *set-point* yang telah ditentukan.

BAB V

KESIMPULAN

5.1 Kesimpulan

Dari hasil pengamatan, pengujian, dan analisa pada hasil perancangan yang dibuat, dapat diperoleh beberapa kesimpulan sebagai berikut.

1. Nilai PPM pada sensor MQ-2 bersifat tidak tetap bergantung pada ketepatan saat melakukan kalibrasi sensor, dimana saat melakukan kalibrasi membutuhkan nilai PPM terendah atau udara cukup bersih pada ruangan
2. Sensor GP2Y1010AU0F dapat mendeteksi debu dan asap pada ruangan karena prinsip kerjanya yang menggunakan cahaya inframerah yang dicerminkan, kekurangan dari sensor ini adanya debu yang menempel di dalam sehingga sensor tetap membaca adanya debu / partikel.
3. Rancangan alat pengendali kualitas udara dalam ruangan dengan implementasi logika fuzzy memiliki performa yang lebih baik dalam hal menjaga kualitas udara dan dalam kecepatan membersihkan polutan yang terdeteksi dibandingkan dengan rancangan tanpa logika fuzzy seperti *on-off* sistem yang bekerja berdasarkan nilai *set-point*.

5.2 Future Work

Saran yang dapat diberikan untuk meningkatkan kinerja sistem kendali kualitas udara ini diantaranya adalah :

1. Kalibrasi sensor dilakukan tidak hanya secara matematis tetapi juga dengan dibandingkan dengan alat pengukur kualitas udara yang telah teruji, sehingga sistem bisa mendapat hasil pengukuran yang lebih baik dan akurat.
2. Menambahka sensor suhu dan kelembaban agar sistem daat mejaga kualitas udara lebih baik lagi.

DAFTAR PUSTAKA

- [1] Faisal, Haruyuki Dewi & Susanto, Agus Dwi, 2017, “Peran Masker/Respirator dalam Pencegahan Dampak Kesehatan Paru Akibat Polusi Udara”. *Jurnal Respirasi*, vol. 3, no. 1, hh. 18 – 25.
- [2] Brown, N. J, 2019, “Indoor Air Quality (Electronic version)”, *Workplace Health and Safety Program*, Ithaca, NY: Cornell University,
- [3] Prihatmanti, Rina, 2016, “Kualitas Udara dalam Ruang”, *Rumahku*, Universitas Ciputra. No.118. halaman 48.
- [4] _____, “Arduino Uno Overview”, *Arduino Datasheet*.
- [5] _____, “Technical Data MQ-2 Gas Sensor”, *Hanwei Electronics Datasheet*.
- [6] _____, “GP2Y1010AU0F Compact Optical Dust Sensor”, *Sharp Datasheet*.
- [7] _____, “Spesification of LCD Module”, *Xiamen Amotec Display Co*.
- [8] _____, “L298N Motor Driver Module”, *Handson Technology*.
- [9] Fernanda, Dio Andhika, 2017, “Kipas Otomatis Menggunakan Fuzzy Logic”, Politeknik Negeri Batam.

LAMPIRAN

LAMPIRAN A

Datasheet Sensor MQ-2

MQ-2 Semiconductor Sensor for Combustible Gas

Sensitive material of MQ-2 gas sensor is SnO_2 , which with lower conductivity in clean air. When the target combustible gas exist, The sensor's conductivity is more higher along with the gas concentration rising. Please use simple electrocircuit, Convert change of conductivity to correspond output signal of gas concentration.

MQ-2 gas sensor has high sensility to LPG, Propane and Hydrogen, also could be used to Methane and other combustble steam, It is with low cost and suitable for different application.

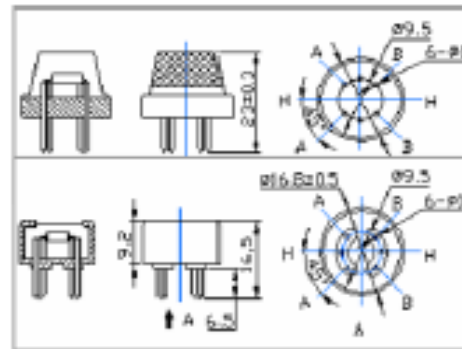
Character

- * Good sensitivity to Combustible gas in wide range
- * High sensitivity to LPG, Propane and Hydrogen
- * Long life and low cost
- * Simple drive circuit

Application

- * Domestic gas leakage detector
- * Industrial Combustible gas detector
- * Portable gas detector

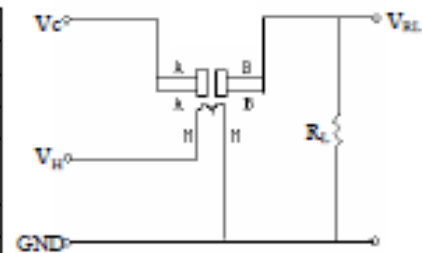
Configuration



Technical Data

Model No.		MQ-2	
Sensor Type		Semiconductor	
Standard Encapsulation		Bakelite (Black Bakelite)	
Detection Gas		Combustible gas and smoke	
Concentration		300-10000ppm (Combustible gas)	
Circuit	Loop Voltage	V_L	$\leq 24V$ DC
	Heater Voltage	V_H	$5.0V \pm 0.2V$ AC or DC
	Load Resistance	R_L	Adjustable
Character	Heater Resistance	R_H	$31\Omega \pm 3\Omega$ (Room Temp.)
	Heater consumption	P_H	$\leq 90mW$
	Sensing Resistance	R_s	$2K\Omega - 20K\Omega$ (in 2000ppm C.B.)
	Sensitivity	S	$R_s(\text{in air})/R_s(1000\text{ppm Isobutane}) \geq 5$
	Slope	α	$\leq 0.5(R_{3000\text{ppm}}/R_{300\text{ppm}} \text{CH}_4)$
Condition	Temp. Humidity	$20^\circ\text{C} \pm 2^\circ\text{C}$, $65\% \pm 5\% \text{RH}$	
	Standard test circuit	$V_L: 5.0V \pm 0.1V$ $V_H: 5.0V \pm 0.1V$	
	Preheat time	Over 48 hours	

Basic test loop



The above is basic test circuit of the sensor. The sensor need to be put 2 voltage, heater voltage (V_H) and test voltage (V_C). V_H used to supply certified working temperature to the sensor, while V_C used to detect voltage (V_{RL}) on load resistance (R_L) whom is in series with sensor. The sensor has light polarity, V_C need DC power. V_C and V_H could use same power circuit with precondition to assure performance of sensor. In order to make the sensor with better performance, suitable R_L value is needed:
Power of Sensitivity body (P_s):
 $P_s = V_C^2 \times R_s / (R_s + R_L)^2$

Resistance of sensor(R_s): $R_s = (V_o/V_{RL} - 1) \times R_L$

Sensitivity Characteristics

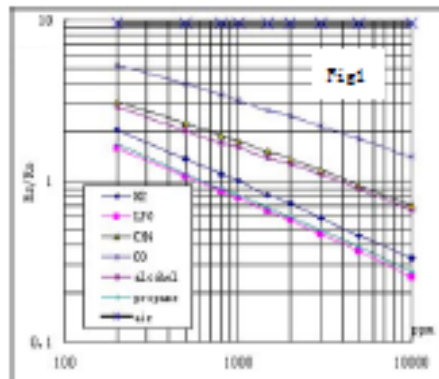


Fig.1 shows the typical sensitivity characteristics of the MQ-2, ordinate means resistance ratio of the sensor (R_s/R_0), abscissa is concentration of gases. R_s means resistance in different gases, R_0 means resistance of sensor in 1000ppm Hydrogen. All test are under standard test conditions.

Influence of Temperature/Humidity

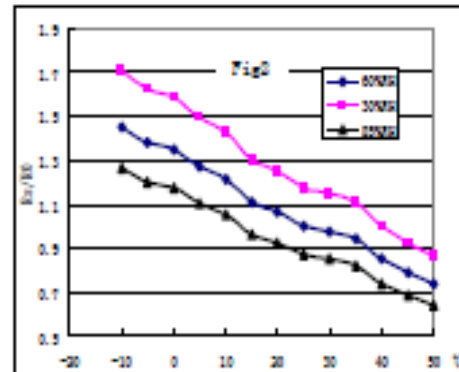
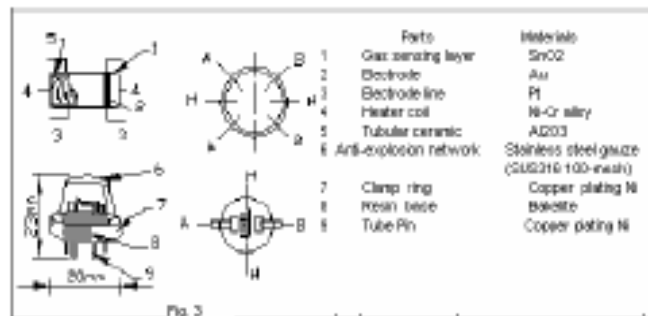


Fig.2 shows the typical temperature and humidity characteristics. Ordinate means resistance ratio of the sensor (R_s/R_0), R_s means resistance of sensor in 1000ppm Butane under different tem. and humidity. R_0 means resistance of the sensor in environment of 1000ppm Methane, 20°C/65%RH

Structure and configuration



Structure and configuration of MQ-2 gas sensor is shown as Fig. 3, sensor composed by micro AL₂O₃ ceramic tube, Tin Dioxide (SnO₂) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-2 have 6 pin, 4 of them are used to fetch signals, and other 2 are used for providing heating current.

LAMPIRAN B

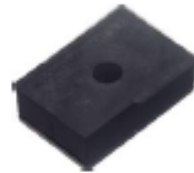
Datasheet Sensor Debu GP2Y1010AU0F

SHARP

GP2Y1010AU0F

GP2Y1010AU0F

Compact Optical Dust Sensor



■ Description

GP2Y1010AU0F is a dust sensor by optical sensing system.

An infrared emitting diode (IRED) and a phototransistor are diagonally arranged into this device.

It detects the reflected light of dust in air.

Especially, it is effective to detect very fine particles like cigarette smoke.

In addition, it can distinguish smoke from house dust by pulse pattern of output voltage.

■ Features

1. Compact, thin package (46.0 × 30.0 × 17.6 mm)
2. Low consumption current (Icc: MAX. 20 mA)
3. The presence of dust can be detected by the photometry of only one pulse
4. Enable to distinguish smoke from house dust
5. Lead-free and RoHS directive compliant

■ Compliance

1. Compliant with RoHS directive (2002/95/EC)

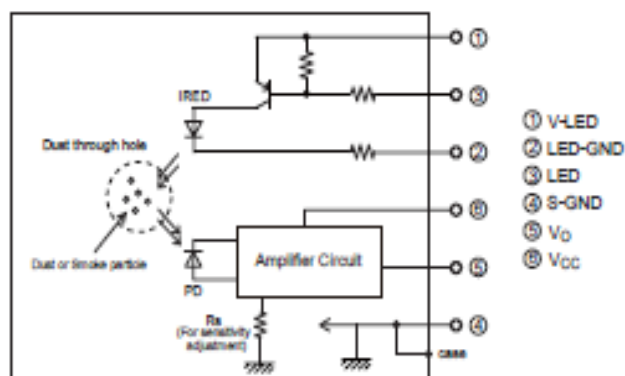
■ Applications

1. Detecting of dust in the air.
2. Example: Air purifier, Air conditioner, Air monitor

Notice: The content of data sheet is subject to change without prior notice.

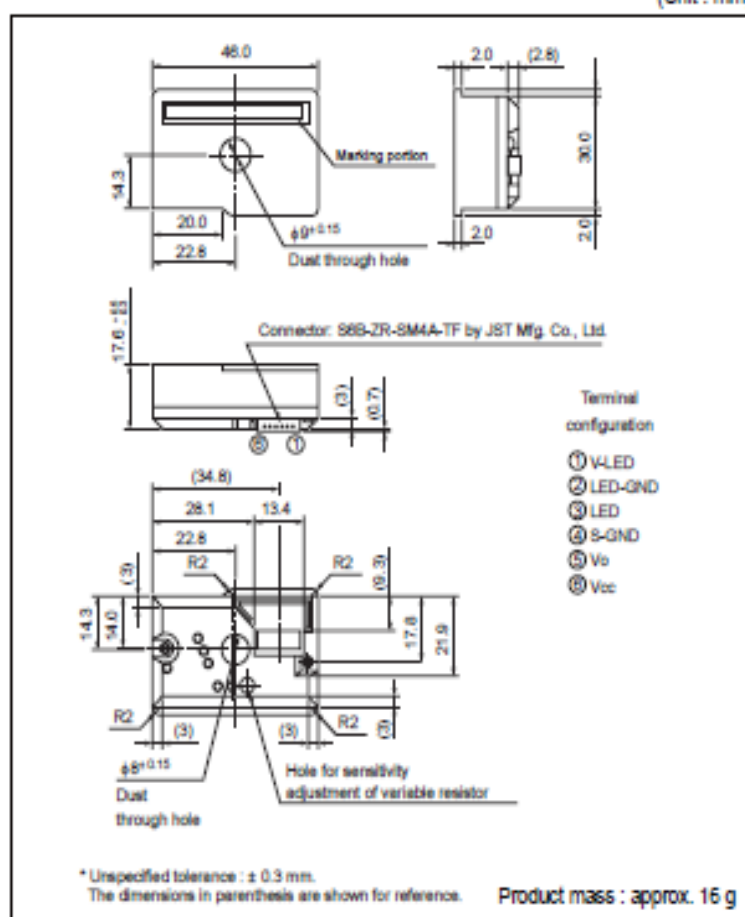
In the absence of confirmation by device specification sheets, SHARP takes no responsibility for any defects that may occur in equipment using any SHARP devices shown in catalogs, data books, etc. Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device.

■ Internal schematic



■ Outline Dimensions

(Unit : mm)



■ Absolute Maximum Ratings

(T_a=25°C)

Parameter	Symbol	Rating	Unit
Supply voltage	V _{CC}	-0.3 to +7	V
*1 Input terminal voltage	V _{LED}	-0.3 to V _{CC}	V
Operating temperature	T _{OP}	-10 to +65	°C
Soldering temperature	T _{sol}	-20 to +80	°C

*1 Open drain drive input

■ Electro-optical Characteristics

(T_a=25°C, V_{CC}=5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Sensitivity	K	*1 *2 *3	0.35	0.5	0.65	V/(0.1mg/m ³)
Output voltage at no dust	V _{OC}	*2 *3	0	0.9	1.5	V
Output voltage range	V _{OH}	*2 *3 R _L =4.7kΩ	3.4	-	-	V
LED terminal current	I _{LED}	*2 LED terminal voltage = 0	-	10	20	mA
Consumption current	I _{CC}	*2 R _L =∞	-	11	20	mA

*1 Sensitivity is specified by the amount of output voltage change when dust density changes by 0.1 mg/m³.
And the dust density for detection is a value of the density of cigarette (MILD SEVEN®) smoke measured by the digital dust monitor (D-SL2: manufactured by SHIBATA SCIENTIFIC TECHNOLOGY LTD.).

*2 Input condition is shown in Fig. 1

*3 Output sampling timing is shown in Fig. 2

■ Recommended input condition for LED input terminal

Parameter	Symbol	Value	Unit
Pulse Cycle	T	10 ± 1	ms
Pulse Width	P _W	0.32 ± 0.02	ms
Operating Supply voltage	V _{CC}	5 ± 0.5	V

Fig. 1 Input Condition for LED Input Terminal

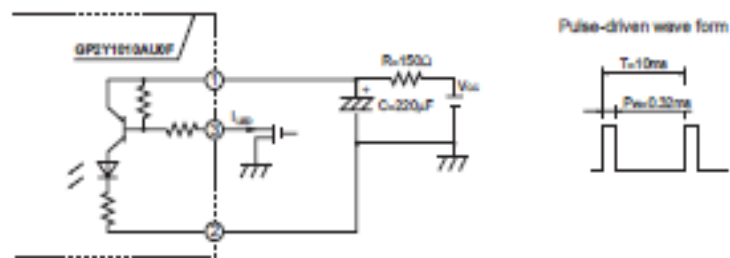


Fig. 2 Sampling Timing of Output Pulse

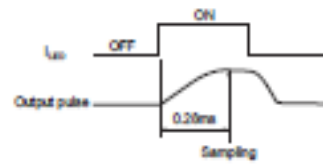
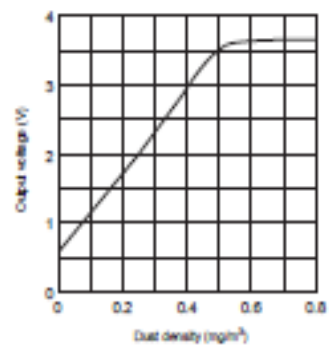


Fig. 3 Output Voltage vs. Dust Density



Remarks : Please be aware that all data in the graph are just for reference and are not for guarantee.

LAMPIRAN C

Program Alat dengan Logika Fuzzy

```
File Edit Sketch Tools Help
Program_akhir
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

int pinAsap      = A0;

int pinDebu      = A1;
int ledPower     = 2;
int samplingTime = 280;
int sleepTime    = 100;

int enablePin    = 6;
int in1Pin       = 7;
int in2Pin       = 8;

float outMeasured = 0;
float calcTegangan = 0;
float kadarAsap   = 0;

float voMeasured  = 0;
float calcVoltage = 0;
float dustDensity = 0;

//Rule Base
float asap [3];
float debu [3];
float rule [3][3];
float rule00, rule01, rule02;
float rule10, rule11, rule12;
float rule20, rule21, rule22;

float defuz, pwm;
float result;
void setup() {

  pinMode(pinAsap, INPUT);
  pinMode(ledPower, OUTPUT);
  pinMode(in1Pin, OUTPUT);
  pinMode(in2Pin, OUTPUT);
  pinMode(enablePin, OUTPUT);

  Serial.begin(9600);
  lcd.begin();
}

void loop() {
  //asap
  outMeasured = analogRead(pinAsap);
  kadarAsap   = (outMeasured * (9700 / 1023)) + 300;

  //debu
  digitalWrite(ledPower, LOW);
  delayMicroseconds(samplingTime);

  voMeasured = analogRead(pinDebu);
  digitalWrite(ledPower, HIGH);
  delayMicroseconds(sleepTime);

  calcVoltage = ( ( voMeasured * 5 ) / 1023);
  dustDensity = calcVoltage * 0.17;
```

```
FuzzyAsap();
FuzzyDebu();
Defuzzy();

Serial.print("Asap :");
Serial.print(kadarAsap);
Serial.print(" ppm");
Serial.print(" ");
Serial.print("Dust Density:");
Serial.print(dustDensity);
Serial.println(" Kg/m3");
Serial.print("PWM :");
Serial.print(pwm);
Serial.print(" ");
Serial.print("\n");
delay(500);

lcd.setCursor(0, 0);
lcd.print("A:");
lcd.print(kadarAsap);
lcd.print(" D:");
lcd.print(dustDensity);
lcd.setCursor(0, 1);
lcd.print("PWM :");
lcd.print(pwm);

analogWrite(enablePin, pwm);

digitalWrite(in1Pin, HIGH);
digitalWrite(in2Pin, LOW);
}

void FuzzyAsap() {
  // kadar asap renggang
  if (kadarAsap <= 2425)
  {
    asap [0] = 1;
  }
  else if (kadarAsap > 2425 && kadarAsap <= 4850)
  {
    asap [0] = (4850 - kadarAsap) / (4850 - 2425);
  }
  else
  {
    asap [0] = 0;
  }

  // kadar asap medium
  if (kadarAsap <= 2425)
  {
    asap [1] = 0;
  }
  else if (kadarAsap > 2425 && kadarAsap <= 4850)
  {
    asap [1] = (kadarAsap - 2425) / (4850 - 2425);
  }
}
```

```

    asap [1] = 0;
}
else if (kadarAsap > 2425 && kadarAsap <= 4850)
{
    asap [1] = (kadarAsap - 2425) / (4850 - 2425);
}
else if (kadarAsap > 4850 && kadarAsap <= 7275)
{
    asap [1] = (7275 - kadarAsap) / (7275 - 4850);
}
else
{
    asap [1] = 0;
}

// kadar asap pekat
if (kadarAsap <= 4850)
{
    asap [2] = 0;
}
else if (kadarAsap > 4850 && kadarAsap <= 7275)
{
    asap [2] = (kadarAsap - 4850) / (7275 - 4850);
}
else
{
    asap [2] = 1;
}
}

void FuzzyDebu() {
// debu sedikit
if (dustDensity <= 0.21 )
{
    debu [0] = 1;
}
else if (dustDensity > 0.21 && dustDensity <= 0.43)
{
    debu [0] = (0.43 - dustDensity) / (0.43 - 0.21);
}
else
{
    debu [0] = 0;
}

// debu sedang
if (dustDensity <= 0.21)
{
    debu [1] = 0;
}
else if (dustDensity > 0.21 && dustDensity <= 0.43)
{
    debu [1] = (dustDensity - 0.21) / (0.43 - 0.21);
}
else if (dustDensity > 0.43 && dustDensity <= 0.64)
{
    debu [1] = (0.64 - dustDensity) / (0.64 - 0.43);
}
else

```

```

{
    debu [1] = 0;
}

// debu banyak
if (dustDensity <= 0.43)
{
    debu [2] = 0;
}
else if (dustDensity > 0.43 && dustDensity <= 0.64)
{
    debu [2] = (dustDensity - 0.43) / (0.64 - 0.43);
}
else
{
    debu [2] = 1;
}
}

void RuleEva() {
int i, j;
for ( i = 0; i <= 2; i = i + 1)
{
    for ( j = 0; j <= 2; j = j + 1)
    {
        result = min( asap[i], debu[j]);
        rule [i][j] = result;
    }
}

rule00 = rule [0][0];
rule01 = rule [0][1];
rule02 = rule [0][2];

rule10 = rule [1][0];
rule11 = rule [1][1];
rule12 = rule [1][2];

rule20 = rule [2][0];
rule21 = rule [2][1];
rule22 = rule [2][2];
}

void Defuzzy() {
//weighted average
float lambat = 100;
float sedang = 175;
float cepat = 250;

RuleEva();
pwm = (rule00 * lambat) + (rule01 * lambat) + (rule02 * sedang) + (rule10 * lambat)

defuz = 0;
int i, j;
for ( i = 0; i <= 2; i = i + 1)
{
    for ( j = 0; j <= 2; j = j + 1)
    {
        defuz = defuz + rule [i][j];
    }
}

```

LAMPIRAN D

Program Alat tanpa Logika Fuzzy

```
File Edit Sketch Tools Help
Program_non_Fuzzy
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

int pinAsap      = A0;

int pinDebu      = A1;
int ledPower     = 2;
int samplingTime = 280;
int sleepTime    = 100;

int enablePin    = 6;
int in1Pin       = 7;
int in2Pin       = 8;

float outMeasured = 0;
float calcTegangan = 0;
float kadarAsap   = 0;

float voMeasured  = 0;
float calcVoltage = 0;
float dustDensity = 0;

void setup() {

  pinMode(pinAsap, INPUT);
  pinMode(ledPower, OUTPUT);
  pinMode(in1Pin, OUTPUT);
  pinMode(in2Pin, OUTPUT);
  Serial.begin(9600);
  lcd.begin();
}

void loop() {
  //asap
  outMeasured = analogRead(pinAsap);
  kadarAsap   = (outMeasured * (9700 / 1023)) + 300;

  //debu
  digitalWrite(ledPower, LOW);
  delayMicroseconds(samplingTime);

  voMeasured = analogRead(pinDebu);
  digitalWrite(ledPower, HIGH);
  delayMicroseconds(sleepTime);

  calcVoltage = ( ( voMeasured * 5 ) / 1023);
  dustDensity = calcVoltage * 0.17;

  if (kadarAsap >= 6500 && dustDensity >= 0.51)
  {
    digitalWrite(in1Pin, HIGH);
    digitalWrite(in2Pin, LOW);
    digitalWrite(enablePin, LOW);

    Serial.print ("Asap :");
    Serial.print (kadarAsap);
    Serial.print ("Debu :");
  }
}
```

```
Serial.print (dustDensity);
Serial.print ("Udara Kotor");

lcd.setCursor(0, 0);
lcd.print("A:");
lcd.print(kadarAsap);
lcd.print(" D:");
lcd.print(dustDensity);
lcd.setCursor(0, 1);
lcd.print("Udara Kotor");
}
else
{
  digitalWrite(in1Pin, LOW);
  digitalWrite(in2Pin, LOW);
  digitalWrite(enablePin, LOW);

  Serial.print ("Asap :");
  Serial.print (kadarAsap);
  Serial.print ("Debu :");
  Serial.print (dustDensity);
  Serial.print ("Udara Bersih");

  lcd.setCursor(0, 0);
  lcd.print("A:");
  lcd.print(kadarAsap);
  lcd.print(" D:");
  lcd.print(dustDensity);
  lcd.setCursor(0, 1);
  lcd.print("Udara Bersih");
}
}
```

