

## Pemadam Api Otomatis dengan Kendali Suhu dan Kelembaban Ruangan Menggunakan Logika *Fuzzy*

Luthfi Aprilianto<sup>\*1</sup>, Tri Kuntoro Priyambodo<sup>2</sup>

<sup>1</sup>Prodi Elektronika dan Instrumentasi Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM

<sup>2</sup>Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM

e-mail: <sup>\*1</sup>[aprilianto.luthfi@gmail.com](mailto:aprilianto.luthfi@gmail.com), <sup>2</sup>[mastri.ugm@gmail.com](mailto:mastri.ugm@gmail.com)

### Abstrak

Pada penelitian ini dilakukan pengembangan inovasi rumah cerdas (*smart home*) sistem instrumentasi dan kendali kondisi ruangan dengan kendali logika fuzzy. Sistem ini mengendalikan suhu serta kelembaban ruangan dan mengatur kadar polusi udara dalam ruangan. Sistem ini digabungkan dengan sistem keamanan pemadam kebakaran otomatis yang dapat mendeteksi adanya sumber api dan melakukan pemadaman secara otomatis.

Sistem ini akan menggunakan nilai temperatur, polusi udara, dan paparan inframerah untuk menentukan adanya kebakaran. Selain itu, sistem ini menggunakan nilai temperatur, kelembaban, dan polusi udara untuk selanjutnya diproses di pemrosesan fuzzy untuk mengendalikan kipas DC sebagai pengatur kondisi ruangan.

Sistem ini memiliki response time untuk setiap penurunan suhu 1 °C adalah 6206.75 ms, setiap kenaikan kelembaban 1% RH adalah 2361.50 ms dan setiap penurunan gas CO 1 ppm adalah 1529.72 ms, sedangkan apabila tidak menggunakan kendali logika fuzzy maka response time untuk setiap penurunan suhu 1 oC adalah 7931.15 ms, setiap kenaikan kelembaban 1% RH adalah 4020.33 ms dan setiap penurunan gas CO 1 ppm adalah 1697.60 ms. Pemadaman api otomatis memiliki response time kurang lebih 30 detik sesaat setelah sumber api menyala. Selain itu, komunikasi data dapat mengirimkan dan menerima data tanpa adanya data loss dengan jarak kurang lebih 30 meter. Interfacing pada LabView dapat ditampilkan dengan baik dan data logging dapat dilakukan setiap satu detik.

**Kata kunci**—Pemadam Api Otomatis, XBee Transceiver, LabView, Kendali Fuzzy

### Abstract

In this research, the innovation development of smart home, instrumentation system and control room conditions with fuzzy logic control. This system controls the temperature and humidity of the room and set the levels of indoor air pollution. This system combined with automated fire safety system that can detect the source of the fire and fighters automatically.

This system will use the value of the temperature, air pollution, and exposure of infrared to determine the presence of fire. In addition, the system uses the value of the temperature, humidity, and air pollution to be further processed in the processing for the fuzzy control of DC fan as a regulator of room conditions.

This system has a response time for each 1 ° C drop in temperature is 6206.75 ms, every 1% RH increase in humidity is 2361.50 ms, and every 1 ppm decrease in CO gas is 1529.72 ms, whereas when not using the fuzzy logic control response time for each 1 ° C drop in temperature is 7931.15 ms, every 1% RH increase in moisture is 4020.33 ms and every 1 ppm decrease in CO gas is 1697.60 ms. Automatic fire suppression has a response time of approximately 30 seconds immediately after the source of flame. In addition, data communication can send and receive data without any loss of data at a distance of

*approximately 30 meters. Interfacing in LabView can be displayed properly and data logging can be done every single second.*

**Keywords**—Automatic Fire Extinguisher, XBee Transceiver, LabView, Fuzzy control

## 1. PENDAHULUAN

Berdasarkan penelitian dari *US Environmental Protection Agency* yang dikutip dari [www.epa.gov](http://www.epa.gov), kualitas udara dalam ruang yang buruk dapat mempengaruhi kesehatan dan nyaman dengan menyebabkan gangguan alergi dan asma, sakit kepala, lemas dan banyak gangguan kesehatan lainnya lagi. Beberapa penyebab kualitas udara dalam ruang yang buruk di sekolah maupun rumah adalah bau tidak sedap, dan pencemar lainnya, paparan kimiawi dan pemicu asma seperti debu, jamur dan gas buangan kendaraan bermotor. Timbulnya bau tidak sedap, debu, jamur dan gas buangan kendaraan bermotor, disebabkan oleh sirkulasi udara yang buruk serta suhu dan kelembaban dalam ruangan yang tinggi. Sehingga menyebabkan panas dan lembab yang dapat memancing pertumbuhan jamur di sekitar ruangan.

Panas dan lembab pada suatu ruangan dapat merusak perabotan rumah tangga dan alat elektronik. Hal tersebut selain dapat menyebabkan kerusakan pada alat elektronik, juga menyebabkan konsleting yang dapat memancing percikan api. Percikan api tersebut dapat menyebabkan kebakaran. Untuk mengantisipasi kebakaran, diperlukan Alat Pemadam Api Ringan (APAR). Namun seringkali, kepedulian terhadap pencegahan kebakaran sangat kecil sekali. Sehingga ketersediaan alat pemadam api (APAR) seperti *fireextinguisher* maupun *hydrant* sangat jarang dijumpai di sekitar rumah. Hal tersebut merupakan suatu kelalaian yang dapat menyebabkan sulitnya pemadaman api saat terjadinya kebakaran.

Oleh karena itu diperlukan suatu sistem yang dapat memantau serta mengendalikan kondisi udara dalam ruangan yang dapat mengatur suhu, kelembaban, dan membuang gas beracun seperti *carbon monoxide (CO)* serta dapat melakukan pemadaman api secara otomatis saat terjadi kebakaran. Sehingga kebakaran dapat ditangani secara cepat serta kondisi udara dan sirkulasi dalam ruangan dapat terjaga dengan baik dan ruangan pun terbebas dari bau tidak sedap, jamur, gas *carbon monoxide (CO)* dan terhindar dari kerusakan barang elektronik yang dapat menyebabkan konsleting dan kebakaran.

## 2. METODE PENELITIAN

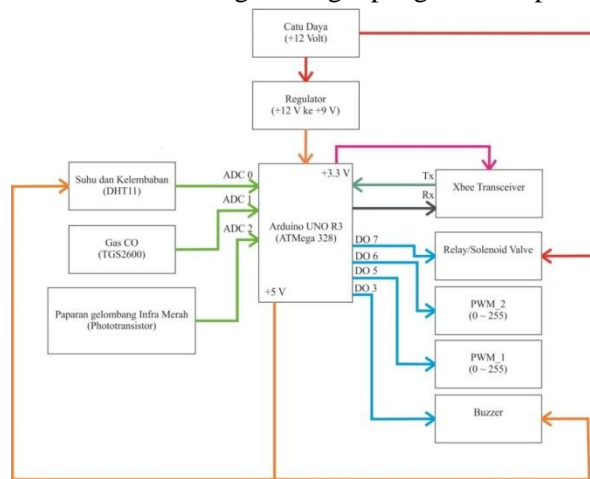
### 2.1 Analisis Sistem

Sistem ini merupakan pemadam api otomatis dengan kendali suhu dan kelembaban ruangan menggunakan logika *fuzzy*. Pemadam api otomatis menggunakan *direct control* berdasarkan parameter terukur sebagai indikator adanya sumber api. Kendali *fuzzy* digunakan untuk mengendalikan kipas DC yang diatur oleh N-Mosfet. Kendali *fuzzy* akan mengendalikan berdasarkan *fuzzy rules* yang ditentukan berdasarkan beberapa parameter sehingga kipas DC akan diatur kecepatan putarnya berdasarkan kondisi ruangan terukur.

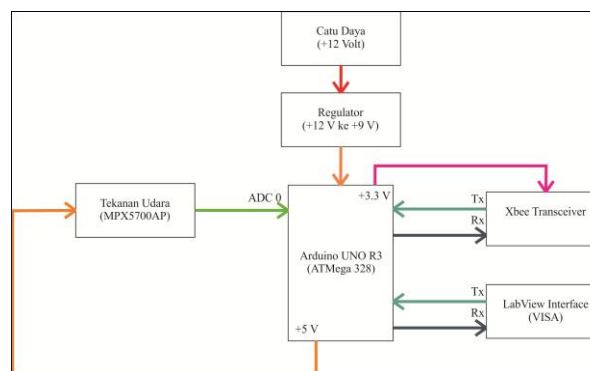
### 2.2. Perancangan Sistem

Rancangan perangkat keras dari sistem terdiri dari atas beberapa blok. Blok diagram *end devices* ditunjukkan pada Gambar 1 dan blok diagram *coordinator* ditunjukkan pada Gambar 2. Pada bagian *end devices* terdapat tiga sensor, yaitu DHT11, TGS2600, dan *phototransistor*. Sensor ini digunakan untuk mengukur suhu, kelembaban, gas CO, dan radiasi inframerah. Sebagai mikrokontroler digunakan arduino uno sebagai otak dari sistem yang berfungsi

mengendalikan semua kerja sistem. Mikrokontroler menerima data dari sensor yang kemudian mengolah, menyimpan, dan memberikan instruksi ke bagian keluaran. Pada bagian keluaran terdiri atas beberapa bagian yaitu modul XBee, *buzzer*, dan N-Mosfet *driver*. Modul XBee berfungsi sebagai pengirim dan penerima data dari *end devices* ke *coordinator* [1]. *Buzzer* berfungsi sebagai indikator ketika terdeteksi adanya sumber api atau saat gas CO mencapai nilai diatas 50 25 ppm. N-Mosfet *driver* berfungsi sebagai pengatur kecepatan putar kipas DC.



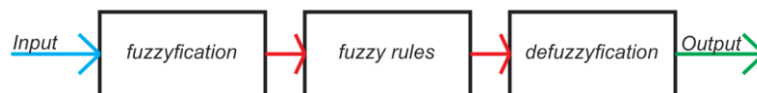
Gambar 1 Blok Diagram *End Devices*



Gambar 2 Blok Diagram *Coordinator*

2.2.1 Rancangan Sistem Pengendali Fuzzy

Pengendali *fuzzy* adalah pengendali utama yang digunakan pada sistem ini dengan menggunakan metode mamdani. Cara kerja kendali *fuzzy* ditunjukkan pada Gambar 3. Parameter masukan utama yang digunakan dalam pengendali *fuzzy* adalah *membership function* dengan tipe *trimf* (*tingular membership function*) yang menghimpun keanggotaan setiap parameter terukur yang akan diproses sesuai *fuzzy rules*[2].



Gambar 3 Blok Diagram Kendali Logika *Fuzzy*

Parameter yang akan digunakan dalam mengendalikan kecepatan kipas DC adalah suhu, kelembaban, dan gas CO. Nilai keanggotaan untuk *fuzzyfication* ditentukan oleh parameter pada Tabel 1 dan Tabel 2. Sedangkan nilai keanggotaan untuk *defuzzyfication* ditentukan oleh

parameter dari setiap keluaran dari *fuzzy rules*. Keluaran pada kendali *fuzzy* berupa pulsa PWM dengan lebar tertentu yang akan digunakan untuk mengatur kecepatan putar pada kipas DC.

Tabel 1 Keanggotaan Nilai Masukandan *Fuzzy Rules* Untuk PWM1

Temp/Humd	Lembab	Sedang	Cukup	Kering
Dingin	1	2	3	4
Sejuk	5	6	7	8
Hangat	9	10	11	12
Panas	13	14	15	16

Tabel 2 Keanggotaan Nilai Masukan dan *Fuzzy Rules* Untuk PWM2

CO/Humd	Kering	Cukup	Sedang	Lembab
Aman	A	B	C	D
Bahaya	E	F	G	H

Jangkauan nilai keanggotaan *membership function* dari masukan dan keluaran kendali *fuzzy* dijelaskan pada keterangan di bawah.

- Suhu : Dingin (10 – 20 °C)  
Sejuk (10 – 30 °C)  
Hangat (20 – 40 °C)  
Panas (30 – 40 °C)
- Kelembaban: Kering (20 – 40 % RH)  
Cukup (20 – 60 % RH)  
Sedang (40 – 80 % RH)  
Lembab (60 – 80 % RH)
- Gas CO : Aman (0 – 100 ppm)  
Bahaya (0 – 100 ppm)
- PWM1 : P1 (0 – 17) P9 (119 – 153)  
P2 (0 – 34) P10 (136 – 170)  
P3 (17 – 51) P11 (153 – 187)  
P4 (34 – 68) P12 (170 – 204)  
P5 (51 – 85) P13 (187 – 221)  
P6 (68 – 102) P14 (204 – 238)  
P7 (85 – 119) P15 (221 – 255)  
P8 (102 – 136) P16 (238 – 255)
- PWM2 : PA (-25 – 45) PE (115 – 185)  
PB (10 – 80) PF (150 – 220)  
PC (45 – 115) PG (185 – 255)  
PD (80 – 150) PH (220 – 255)

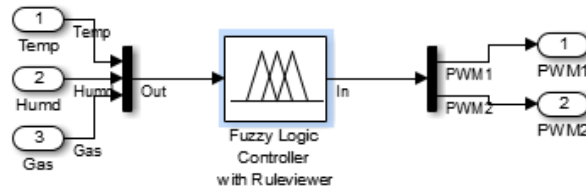
Mikrokontroler akan mengolah sinyal masukan dari sensor. Selanjutnya mikrokontroler akan menentukan keanggotaan dari setiap masukan sensor, lalu ditentukan bobot nilai dari setiap masukan dalam proses *fuzzyfication* sehingga nilai *crisp* dapat diubah menjadi nilai *grey*. Lalu nilai *grey* diproses sesuai *fuzzy rules* sehingga dihasilkan nilai keluaran. Nilai keluaran tersebut lalu diproses dari nilai *grey* menjadi nilai *crisp* dalam proses *defuzzyfication*.

Penentuan nilai keanggotaan disesuaikan dengan variabel bebas yang mengacu kepada kondisi dalam merasakan suhu maupun kelembaban ruangan. Penentuan *fuzzy rules* sebagai

*world model* dalam kendali *fuzzy* berdasarkan banyaknya pengalihan jumlah keanggotaan masukan. Sehingga didapatkan untuk *fuzzy rules* PWM1 terdapat 16 aturan dan untuk *fuzzy rules* PWM2 terdapat 8 aturan.

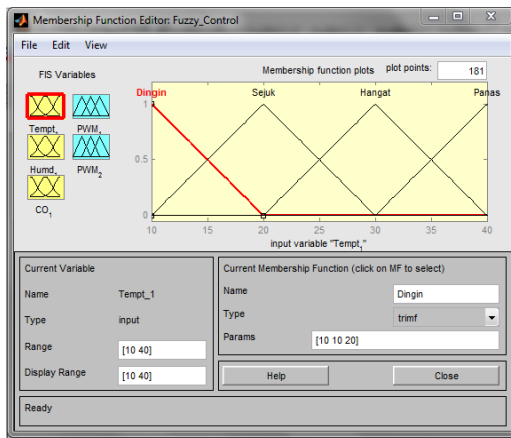
Setelah *fuzzy* dibuat pada *fuzzy inference system*, kemudian *fuzzy* diimplementasikan pada *Simulink* untuk simulasi dan *generate code* untuk mendapatkan program *fuzzy* untuk ditanamkan pada mikrokontroler.

Simulasi pada *Simulink* dilakukan dengan cara memasukan *fuzzy* pada *subsystem* pada *Fuzzy Logic Controller with Ruleviewer*. Setelah simulasi, maka dapat dilakukan *generate code* untuk mendapatkan program dalam bahasa C. Simulasi *Simulink Environment* pada MatLab ditunjukkan pada Gambar 4.

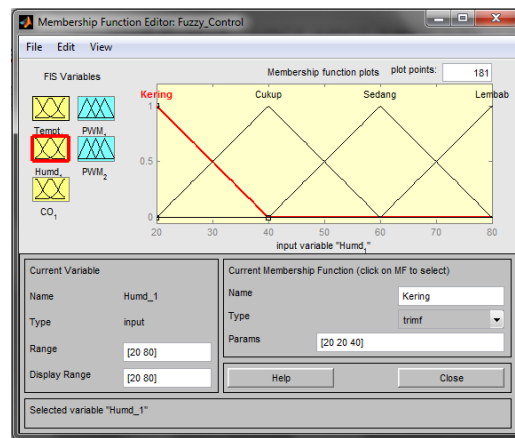


Gambar 4 Simulasi *Fuzzy* Pada *Simulink*

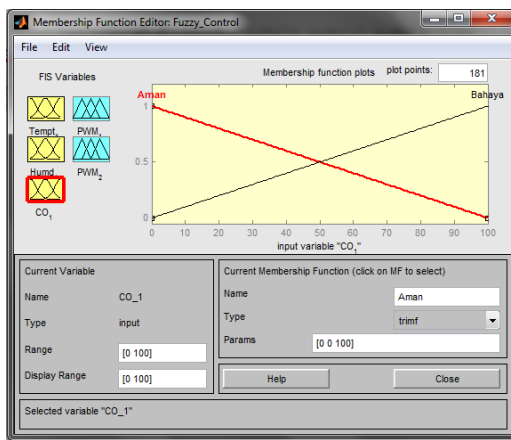
Kendali *fuzzy* dibuat pada *Fuzzy Inference System* yang terdapat pada MatLab. Implementasi kendali *fuzzy* pada *fuzzy inference system* dapat dilihat pada Gambar 5 sampai Gambar 9.



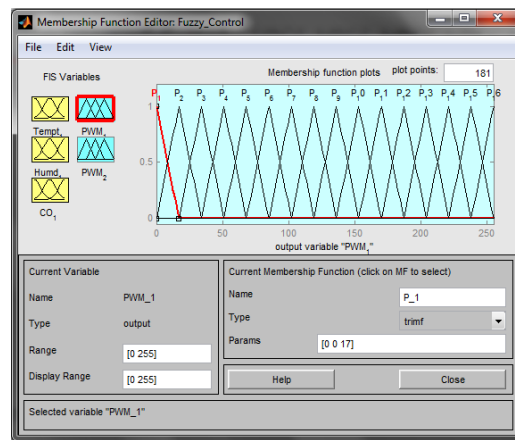
Gambar 5 Keanggotaan Suhu



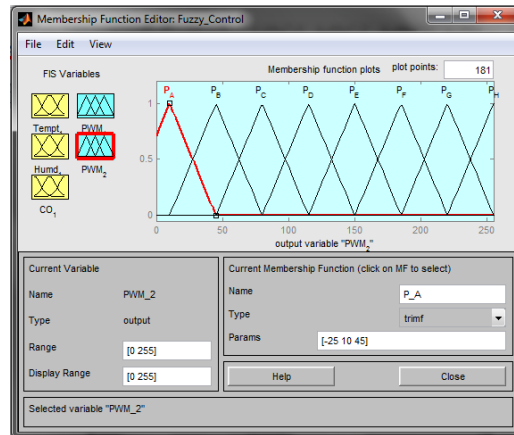
Gambar 6 Keanggotaan Kelembaban



Gambar 7 Keanggotaan Gas CO



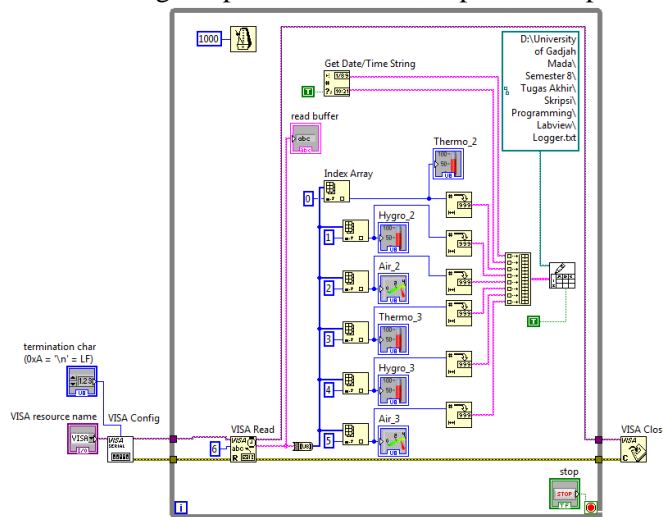
Gambar 8 Keanggotaan PWM1



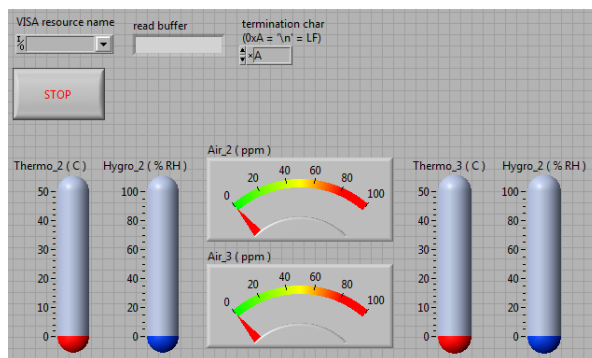
Gambar 9 Keanggotaan PWM2

2.2.2 Rancangan Sistem Antarmuka LabView

Antarmuka LabView berfungsi sebagai penampil data yang dikirim oleh coordinator. Komunikasi antara mikrokontroler dan LabView melalui jembatan VISA [3]. Blok diagram antarmuka dan data logging pada LabView dapat dilihat pada Gambar 10 di bawah. Hasil implementasi dari blok diagram pada Gambar 10, dapat dilihat pada Gambar 11.



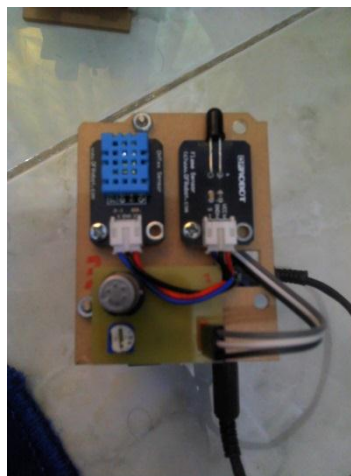
Gambar 10 Blok Diagram Antarmuka dan Data Logging



Gambar 11 Antarmuka LabView

### 2.3 Implementasi Perangkat Keras Sistem

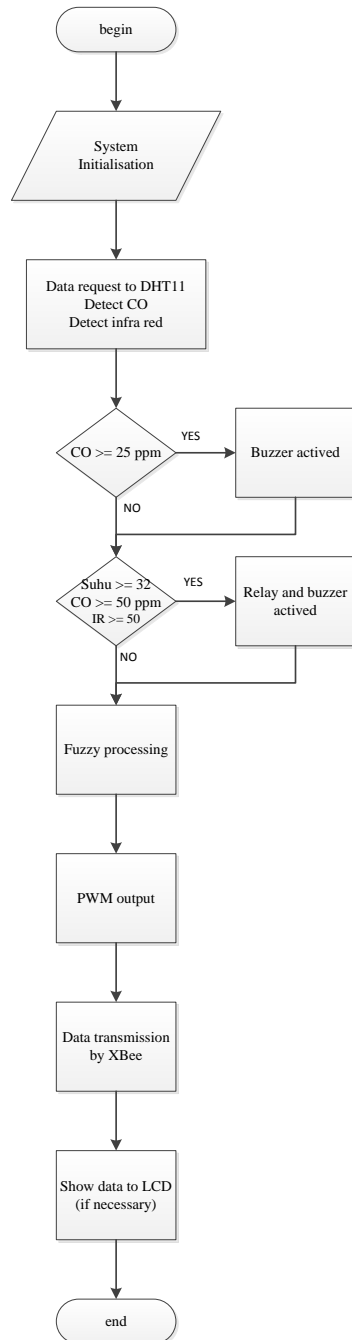
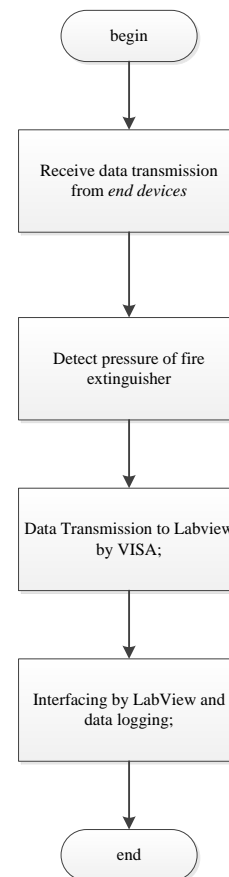
Rangkaian mikrokontroler Arduino UNO ATmega328 merupakan sistem utama yang mengendalikan seluruh aktivitas perangkat elektronik lainnya. Pada Arduino Uno ATmega328 tersebut dipasang dengan modul XBee. Modul tersebut merupakan modul komunikasi *series 1*. Setelah rangkaian Arduino Uno ATmega328 dan modul XBee terpasang, diberikan *shield* untuk mempermudah pemasangan antara Arduino UNO ATmega328 dan modul XBee dengan sensor DHT11, TGS2600, *phototransistor*, dan *buzzer*. Rangkaian mikrokontroler Arduino UNO ATmega328 dan modul XBee sudah tersedia dalam bentuk modul siap pakai sehingga pengguna hanya memerlukan *shield* tambahan yang terhubung dengan bagian *input/output* sistem. Terdapat 3 sensor yang terhubung dengan pin analog 0, analog 1 dan analog 2 pada Arduino UNO ATmega328. Untuk pin keluaran PWM digunakan pin 5 dan 6 yang selanjutnya dihubungkan dengan N-Mosfet *driver*, dan pin 3 terhubung dengan alarm. Perintah pengiriman data digunakan pin *library* XBee yang dikembangkan oleh *google developer*. Rangkaian mikrokontroler Arduino UNO ATmega328 dan XBee beserta *shield* ditunjukkan pada Gambar 12.



Gambar 12 Rangkaian Arduino UNO ATmega328, XBee, dan *Shield*

### 2.4 Implementasi Perangkat Lunak Sistem

Program dibuat dengan menggunakan bahasa pemrograman *C* dengan menggunakan *Arduino IDE (Integrated Development Environment)*. Dalam *Arduino IDE* ini juga telah terdapat fungsi dan *library* untuk menjalankan *feature-feature* khusus yang dimiliki mikrokontroler Aduino UNO ATmega328. *Arduino IDE* tersebut akan mengkompil bahasa *C* yang ditulis kedalam bahasa mesin yang dimengerti oleh mikrokontroler Arduino Uno ATmega328. Program pada Arduino UNO ATmega328 terdiri dari 4 bagian, yaitu program yang menangani proses pembacaan sensor untuk mendeteksi suhu, kelembaban, gas CO dan radiasi inframerah, program pengiriman data yang menggunakan protokol *application programming interface (API)*, program untuk mengaktifkan *buzzer* dan *relay/solenoid valve*, dan program utama yang menangani sistem kecepatan kipas DC menggunakan pengendali *fuzzy*.

Gambar 13 Diagram Alir *End Devices*Gambar 14 Diagram Alir *Coordinator*

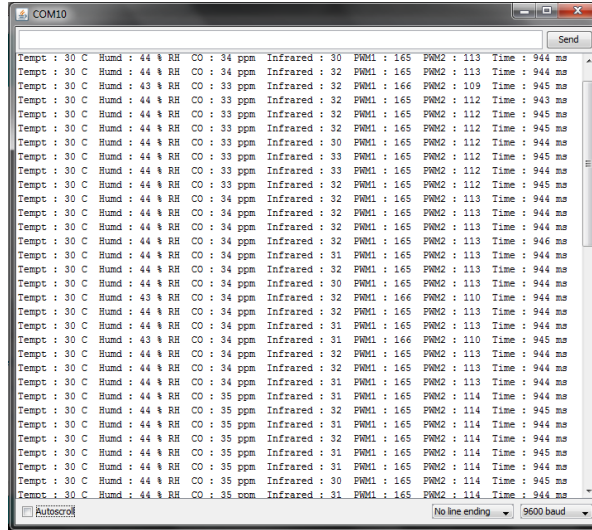
Gambar 13 merupakan diagram alir dari sistem yang terdapat pada *end devices* dimana pada diagram alir sistem terdapat kendali *fuzzy* untuk mengendalikan kecepatan kipas DC, komunikasi untuk mengirimkan data ke *coordinator* dan kendali *solenoid valve* untuk mengaktifkan pemadam api saat sumber api terdeteksi. Sedangkan Gambar 14 merupakan diagram alir sistem yang terdapat pada *coordinator* dimana pada diagram alir sistem terdapat komunikasi untuk menerima data dari *end devices* dan mengirimkan data tersebut ke LabView untuk ditampilkan pada antarmuka dan disimpan dalam *data logging*.



### 3. HASIL DAN PEMBAHASAN

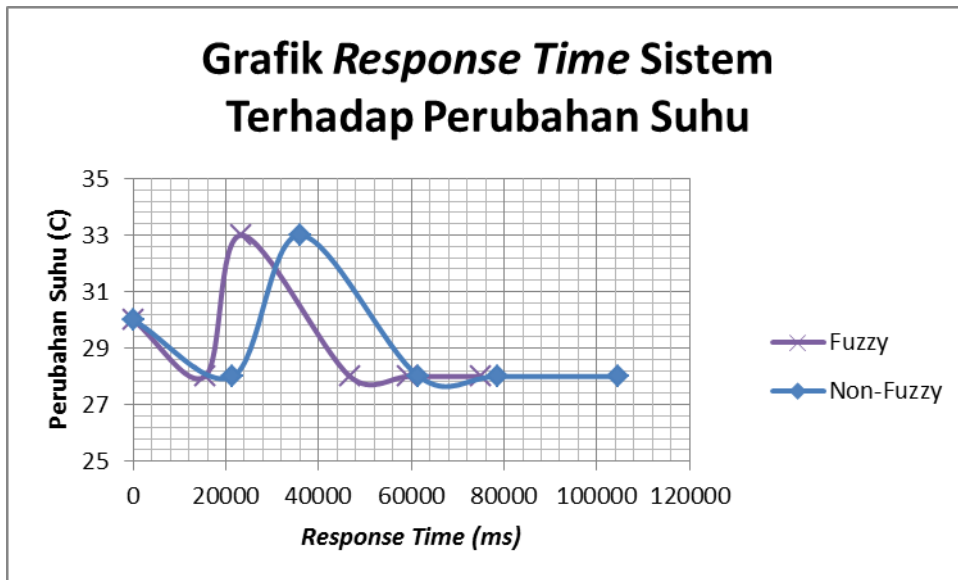
#### 3.1 Hasil Penelitian

Gambar 15 merupakan wujud implementasi dari eksekusi dari sistem yang terdapat pada *end devices*. Setiap sekali eksekusi, dibutuhkan waktu 945 ms dari awal program, hingga data ditampilkan pada *serial*.

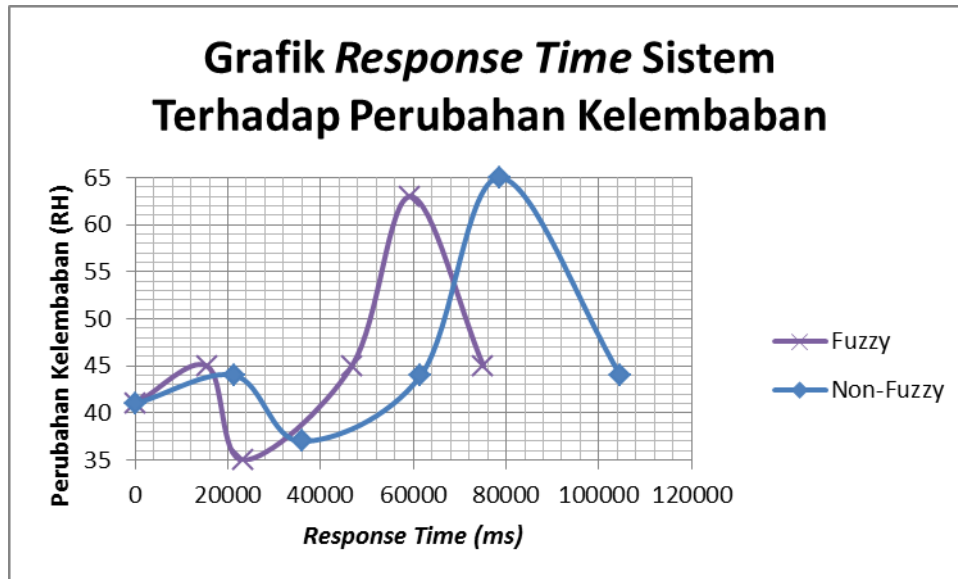


Gambar 15 Eksekusi Program Pada *End Devices*

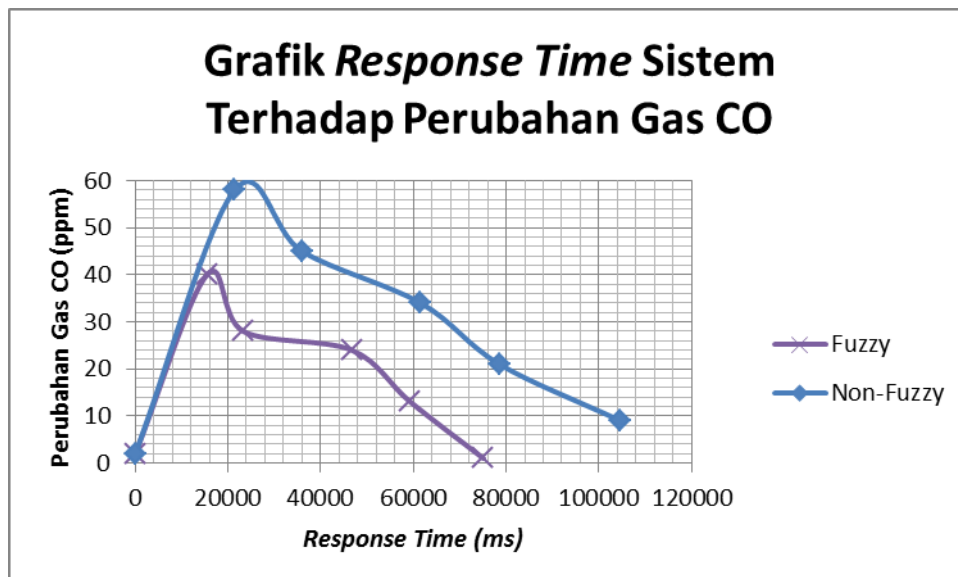
Pada sistem dilakukan pengujian kecepatan kipas DC dengan kendali *fuzzy*. Pengujian dilakukan dengan melihat *response time* dari sistem dengan menguji gangguan pada sistem. Gangguan yang diuji pada pengujian ini adalah gangguan terhadap suhu, kelembaban, dan gas CO.



Gambar 16 Grafik *Response Time* Sistem Terhadap Suhu



Gambar 17 Grafik *Response Time* Sistem Terhadap Kelembaban



Gambar 18 Grafik *Response Time* Sistem Terhadap gas CO

### 3.2 Pembahasan

Pengujian dilakukan dengan menggunakan *fuzzy* dan tanpa menggunakan *fuzzy*. Pada simulasi MatLab, akan dilakukan diberikan masukan nilai suhu, kelembaban, dan gas CO. Sedangkan pada Arduino, akan menampilkan lima data, yaitu suhu, kelembaban, gas CO, keluaran PWM pertama, dan keluaran PWM kedua. Gambar 16 merupakan percobaan logika *fuzzy* dan *non-fuzzy* yang diberi gangguan terhadap perubahan suhu.

Gambar 17 menunjukkan *response time* sistem terhadap perubahan kelembaban. Dapat dilihat bahwa respon sistem lebih cepat apabila menggunakan *fuzzy* dikarenakan sistem kendali *fuzzy* akan terus bekerja sedangkan sistem kendali tanpa *fuzzy* hanya bekerja sesuai *set point* yang ditentukan.

Gambar 18 menunjukkan *response time* sistem terhadap perubahan gas CO. Gangguan terhadap masukan gas CO mencapai 50 ppm sehingga penurunan gas CO di dalam model maket sangat lama. Sehingga variabel bebas untuk gangguan gas CO, hanya dapat diambil beberapa

data, yaitu saat sistem diberi gangguan gas CO tertinggi, hingga perlahan mencapai titik normal kembali lagi, yaitu antara 2 sampai 3 ppm.

Dari ketiga pengujian, didapat *response time* untuk setiap penurunan suhu 1 °C adalah 6206.75 ms, setiap kenaikan kelembaban 1% RH adalah 2361.50 ms dan setiap penurunan gas CO 1 ppm adalah 1529.72 ms, sedangkan apabila tidak menggunakan kendali logika *fuzzy* maka *response time* untuk setiap penurunan suhu 1 °C adalah 7931.15 ms, setiap kenaikan kelembaban 1% RH adalah 4020.33 ms dan setiap penurunan gas CO 1 ppm adalah 1697.60 ms.

Pengujian pemadaman api otomatis memiliki *response time* kurang lebih 30 detik sesaat setelah sumber api menyala. Selain itu, komunikasi data dapat mengirimkan dan menerima data tanpa adanya *data loss* dengan jarak kurang lebih 30 meter. *Interfacing* pada LabView dapat ditampilkan dengan baik dan *data logging* dapat dilakukan setiap satu detik.

#### 4. KESIMPULAN

1. Secara keseluruhan, sistem dapat melakukan pengukuran terhadap lima parameter terukur yaitu, suhu, kelembaban, gas CO, radiasi inframerah, dan tekana udara.
2. Sensor DHT11 dapat mendeteksi dengan sensitifitas suhu sebesar 1 °C dan sensitifitas kelembaban 1 % RH serta jangkauan 0 - 60 °C dan 20 – 90 % RH.
3. Sensor gas TGS2600 dapat mendeteksi dengan sensitifitas satu ppm CO dengan jangkauan 0 – 250 ppm.
4. Sensor *phototransistor* dapat mendeteksi sumber api hingga enam meter dengan besar api menggunakan kertas koran yang dibakar dengan *probe angle* 30° dari pusat *probe*.
5. Sensor tekanan gas MPX5700AP mampu mendeteksi tekanan tabung pemadam api hingga 650 kPa dengan sensitifitas 0.1 kPa.
6. Pemadaman api otomatis berhasil dilaksanakan dengan respon kurang lebih 30 detik saat sumber api ada dengan penentuan parameter terukur, yaitu suhu  $\geq 32$  °C, gas  $\geq$  CO 50 ppm dan pembacaan ADC *phototransistor*  $\geq 50$ .
7. Sistem dapat memproses logika *fuzzy* asas mamdani sesuai simulasi pada MatLab.
8. Sistem ini memiliki *response time* untuk setiap penurunan suhu 1 °C adalah 6206.75 ms, setiap kenaikan kelembaban 1% RH adalah 2361.50 ms dan setiap penurunan gas CO 1 ppm adalah 1529.72 ms, sedangkan apabila tidak menggunakan kendali logika *fuzzy* maka *response time* untuk setiap penurunan suhu 1 oC adalah 7931.15 ms, setiap kenaikan kelembaban 1% RH adalah 4020.33 ms dan setiap penurunan gas CO 1 ppm adalah 1697.60 ms.
9. Sistem dapat mengirim dan menerima data antara *end devices* dengan *coordinator* dengan jarak  $\leq 30$  meter dalam keadaan *LOS (Line of Sight)*.
10. Sistem dapat mengirimkan data ke LabView dan menampilkan data dalam bentuk *interface* serta dapat menyimpan data pada *harddisk* dalam bentuk *spreadsheet*.

#### 5. SARAN

1. Sistem disarankan menggunakan sensor yang memiliki sensitifitas tinggi dan *response time* yang cepat.
2. Sistem disarankan menggunakan pendingin ruangan yang lebih hemat daya dan efisien.
3. Sistem dapat dikembangkan dengan menggunakan web sehingga dapat diakses secara *online*.

## DAFTAR PUSTAKA

- [1] Ardiyanto, L, 2012,*Implementasi Jaringan Sensor Nirkabel Berbasis Xbee: Studi Kasus Pemantau Suhu dan Kelembaban*, Tugas Akhir S1, Jurusan Ilmu Komputer dan Elektronika, Fakultas MIPA, Universitas Gadjah Mada, Yogyakarta
- [2] Wiguna, E. A., 2011,*Prototipe Rumah Jamur Tiram Otomatis Dengan Pengendali Suhu dan Kelembaban Menggunakan Kontrol Logika Fuzzy*, Tugas Akhir S1, Jurusan Ilmu Komputer dan Elektronika, Fakultas MIPA, Universitas Gadjah Mada, Yogyakarta
- [3] Artanto, D, 2012,*Interaksi Arduino dan LabVIEW*, PT. Gramedia, Jakarta