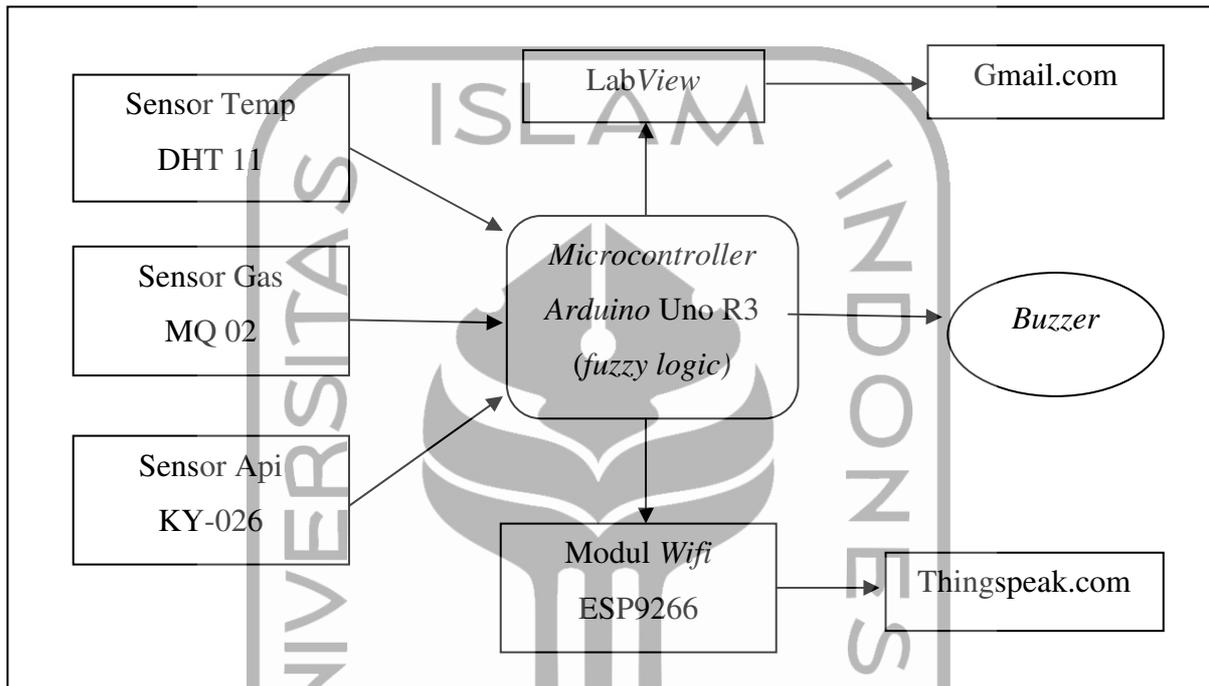


BAB 3

METODOLOGI

3.1 Perancangan pendeteksi kebakaran otomatis secara umum

Pada bagian ini berisi mengenai perancangan secara umum dari pendeteksi kebakaran otomatis dan berikut adalah blok diagram mengenai perangkat keras pada pendeteksi kebakaran otomatis yang sedang dirancang :



Gambar 3.1 Perancangan *prototype* pendeteksi kebakaran otomatis secara umum

Pada Gambar 3.1 ditunjukkan bahwa pada sistem pendeteksi kebakaran otomatis digunakannya beberapa komponen guna membuat pendeteksi kebakaran otomatis pada penelitian ini dan penjelasan masing-masing komponen perangkat keras tersebut adalah sebagai berikut :

1. Sensor *temperature* DHT 11

Sensor DHT11 ini merupakan salah satu jenis sensor *temperature* ruangan yang biasa digunakan. *Output* sensor DHT11 ini berupa nilai *temperature real time* pada suatu ruangan dan nilai *output* dari sesor ini akan menjadi *input* bagi sistem pendeteksi kebakaran otomatis.

2. Sensor gas MQ-02

Sensor MQ-02 ini merupakan salah satu jenis sensor gas yang biasa digunakan dalam membuat sebuah sistem. Sensor gas MQ-02 ini dapat mengukur konsentrasi dari berbagai jenis gas yang ada pada suatu ruangan, namun pada penelitian ini sensor gas MQ-02 ini akan lebih berfokus pada gas CO saja. *Output* dari sensor gas MQ-02 ini

berupa nilai konsentrasi gas pada suatu ruangan dan nilai *output* dari sensor ini akan menjadi *input* bagi sistem pendeteksi kebakaran otomatis.

3. Sensor api KY-026

Sensor api KY-026 ini merupakan sensor api yang menggunakan infra merah guna mendeteksi sebuah api. *Output* dari sensor api KY-026 ini berupa nilai *analog* yang berikisar antara 0 sampai dengan 1024 dan nilai *analog* inilah yang akan menjadi *input* bagi sistem pendeteksi kebakaran otomatis.

4. *Microcontroller Arduino R3*

Arduino uno ini berisikan semua program-program yang mana diantaranya dari mulai program sensor maupun program *fuzzy logic* itu sendiri. Program yang dituliskan ke dalam *microcontroller Arduino R3* ini berguna untuk membuat sistem pendeteksi kebakaran otomatis beroperasi sesuai yang diinginkan.

5. *Buzzer*

Buzzer pada penelitian ini digunakan sebagai output yang mana *Buzzer* akan mengeluarkan suara apabila dideteksinya kebakaran pada tempat yang sedang diamati.

6. Modul *Wifi* ESP8266

Modul *wifi* ESP8266 adalah sebuah modul *wifi* yang memiliki fungsi sebagai perangkat tambahan pada mikrokontroler seperti *Arduino* agar mikrokontroler dapat membuat koneksi TCP/IP dan juga dapat tersambung langsung dengan *wifi*.

7. Thingspeak.com

Data dari *Arduino* akan dikirimkan ke *web server* Thingspeak sehingga data yang berupa nilai *temperature*, gas, dan kondisi api dapat dimonitor dimana saja

8. Gmail

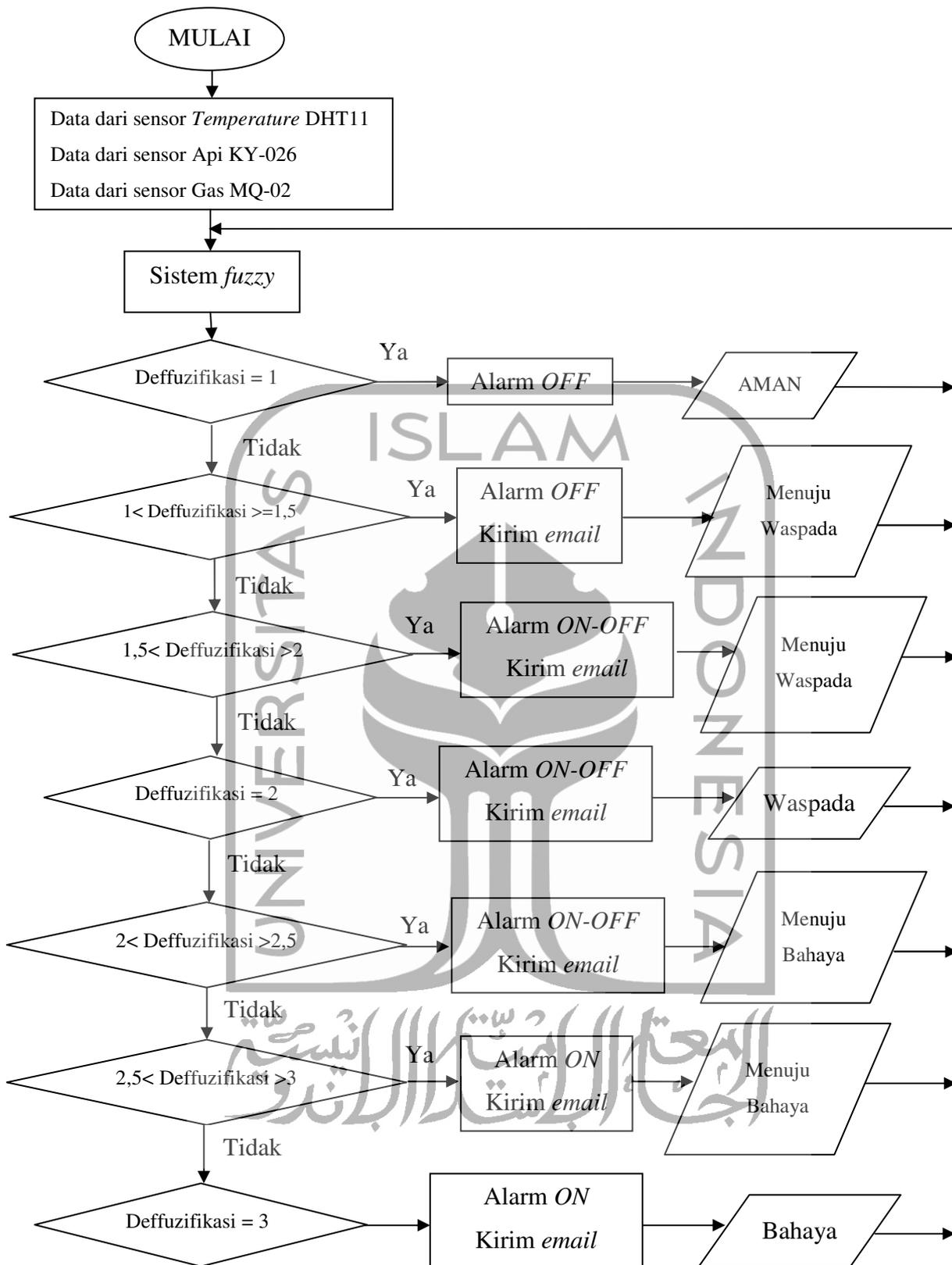
Jika sistem mendeteksi sebuah kebakaran maka sistem akan mengirimkan *e-mail* ke pemilik hal ini ditujukan agar pemilik tahu kondisi rumahnya.

9. LabVIEW

Perangkat lunak ini digunakan untuk memprogram pengiriman *e-mail* ke pemilik.

3.2 Perancangan *software* pendeteksi kebakaran otomatis

Pada bagian ini akan dijelaskan bagaimana perangkat lunak dirancang agar pendeteksi kebakaran otomatis dapat berjalan sesuai dengan fungsinya itu sendiri. Pada penelitian ini program-program dibuat menggunakan perangkat lunak *Arduino IDE* serta *LabVIEW*. Perancangan perangkat lunak pendeteksi kebakaran otomatis dapat dilihat pada diagram alir berikut ini



Gambar 3.2 Diagram alir perancangan *software* untuk pendeteksi kebakaran otomatis

Pada Gambar 3.2 ditunjukkan alur pembuatan program guna membuat sistem pendeteksi kebakaran otomatis ini dapat bekerja sesuai dengan apa yang diinginkan dan penjelasan diagram alir tersebut adalah sebagai berikut:

1. Pengambilan data dari sensor-sensor

Sistem pendeteksi kebakaran yang sedang dirancang akan beroperasi secara otomatis, sehingga masukan yang dibutuhkan oleh sistem didapatkan dari sensor-sensor yang digunakan dan sensor-sensor ini yaitu sensor *temperature* DHT11, sensor gas MQ-02, dan sensor api KY-026. Setiap sensor memiliki keluaran berupa nilai yang mana nilai keluaran dari masing-masing sensor inilah yang menjadi masukan ke sistem.

2. Sistem *fuzzy*

Pada tahapan ini, nilai keluaran dari masing-masing sensor akan diolah agar nilai-nilai keluaran sensor ini dapat menjadi sebuah informasi yang dapat mendeteksi adanya sebuah kebakaran. Sistem *fuzzy* ini sendiri memiliki beberapa bagian di dalamnya, yaitu diantaranya *fuzzifikasi*, *inferensi fuzzy*, serta *defuzzifikasi*. Bagian-bagian ini pada sistem ini diimplementasikan dalam bentuk program yang dibuat di *Arduino IDE*. Pada tahapan ini juga, sistem *fuzzy* akan mengolah 3 nilai keluaran dari 3 buah sensor sehingga nantinya akan menjadi satu buah keluaran saja dan keluaran inilah nantinya yang akan menentukan sebuah kondisi ruangan dalam keadaan kebakaran atau tidak.

3. Menyalakan *alarm* dan mengirimkan *e-mail*

Pada tahapan ini, kondisi kebakaran di suatu ruang dideteksi oleh sistem pendeteksi kebakaran secara otomatis. Saat sistem mendeteksi adanya sebuah kebakaran pada suatu ruangan maka sistem akan langsung menghidupkan *alarm* dan mengirimkan *e-mail* pemberitahuan kondisi di ruangan tersebut. *E-mail* akan dikirimkan ke alamat *e-mail* pemilik rumah dengan maksud apabila pemilik rumah tidak ada di rumah pada saat terjadi kebakaran maka *e-mail* pemberitahuan ini akan dikirimkan ke *user*.

4. Menampilkan level kebakaran

Selain mendeteksi terjadinya sebuah kebakaran, sistem pendeteksi kebakaran otomatis ini juga akan melihat seberapa besar kebakaran yang sedang terjadi. Pada sistem ini terdapat 3 buah level dari kondisi sebuah kebakaran, yaitu diantaranya AMAN yang mana level dibuat untuk menginformasikan ke pemilik rumah bahwasannya sedang tidak terjadi kebakaran pada rumah tersebut, dan level selanjutnya adalah WASPADA pada level ini sistem mengidentifikasi adanya kebakaran pada suatu ruangan, namun kebakaran ini masih dapat ditangani oleh pemilik rumah itu sendiri, oleh karena ini dengan adanya pemberitahuan ini pemilik dapat langsung bertindak tangani secara cepat dan sigap, dan level yang terakhir adalah BAHAYA, pada level ini kebakaran

teridentifikasi oleh sistem sudah besar, sehingga pemilik rumah harus segera menghubungi pihak pemadam kebakaran guna membantu memadamkan kebakaran yang ada pada rumahnya. Setiap level ini dapat ditentukan melalui hasil dari *defuzzifikasi* pada tahapan sistem *fuzzy*.

3.3 Kalibrasi Sensor *Temperature* DHT11

Pada pengkalibrasian sensor *temperature* DHT11 digunakan sebuah *library* dari *Arduino* yaitu DHT.h yang mana *library* ini akan mengubah nilai keluar sensor yang berupa nilai dari 0 sampai dengan 1024 menjadi nilai *temperature* dengan satuan °C.

3.4 Kalibrasi Sensor Gas MQ-02

Sensor Gas MQ-02 dapat mengukur berbagai macam jenis gas. Pada penelitian ini jenis gas yang akan diukur adalah gas CO yang mana gas ini akan dihasilkan disaat adanya ketidaksempurnaan dalam proses pembakaran. Guna mengkalibrasi sensor gas MQ-02 ini agar yang terukur adalah nilai gas CO dan mengubah nilai analog menjadi nilai ppm maka digunakan persamaan berikut ini :

$$\begin{aligned}
 Y &= mX + c \\
 Y - y_1 &= m(X - x_1) + 0 \\
 X &= \frac{(Y - y_1)}{m} + x_1 \\
 X &= \frac{\left(\frac{R_s}{R_o} - y_1\right)}{m} + x_1 \quad (3.1) \\
 \log X &= \frac{\left(\log\left(\frac{R_s}{R_o}\right) - y_1\right)}{m} + x_1 \\
 X &= 10^{\left(\frac{\log\left(\frac{R_s}{R_o}\right) - y_1}{m} + x_1\right)}
 \end{aligned}$$

Keterangan :

- X = Nilai PPM di grafik.
- Y = Nilai R_s/R_o .
- R_s = Resistansi sensor pada konsentrasi gas tertentu.
- R_o = Resistansi sensor pada udara bersih.
- m = Gradien konsentrasi gas tertentu di grafik.
- x_1 = nilai titik awal gas CO terhadap sumbu ppm.
- y_1 = nilai titik awal gas CO terhadap sumbu R_s/R_o .

3.5 Kalibrasi sensor api KY-026

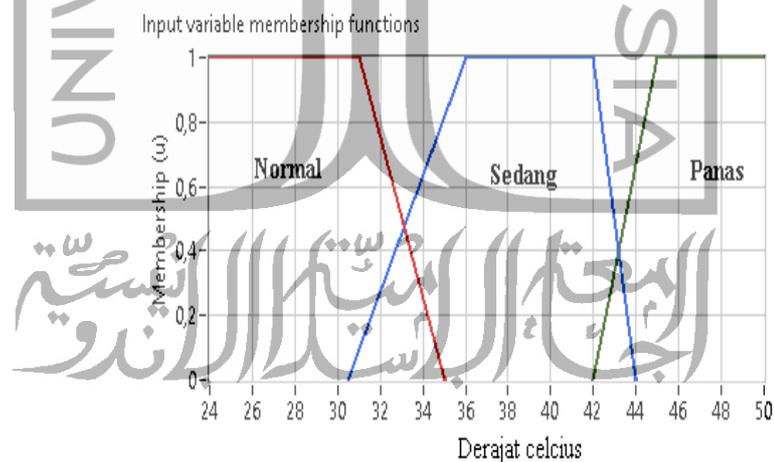
Pada sensor deteksi api KY-026 tidak dilakukannya pengkalibrasian dikarenakan sensor ini hanya mengukur satu jenis objek saja yaitu inframerah sehingga nilai keluaran sensor yang digunakan untuk sistem ini masih sama yaitu berupa nilai dari 0 sampai 1024.

3.6 Perancangan sistem *fuzzy*

Pada perancangan sistem *fuzzy* sendiri terbagi dari beberapa tahap, dan beberapa tahap tersebut yaitu diantaranya :

1. Membuat himpunan *fuzzy*

Membuat himpunan *fuzzy* merupakan langkah pertama yang mana pada tahapan ini ditentukannya jumlah variabel *linguistik* yang akan digunakan serta pembagian data pada masing-masing variabel *linguistik*. Pada sistem pendeteksi kebakaran otomatis menggunakan 3 buah variabel *input* yang mana variabel *input* ini berupa nilai keluaran dari sensor *temperature*, sensor gas, dan sensor api dan juga setiap variabel *input* masing-masing memiliki 3 buah variabel *linguistik*. Sehingga untuk perancangan *input* sistem *fuzzy* pada pendeteksi kebakaran otomatis adalah sebagai berikut :



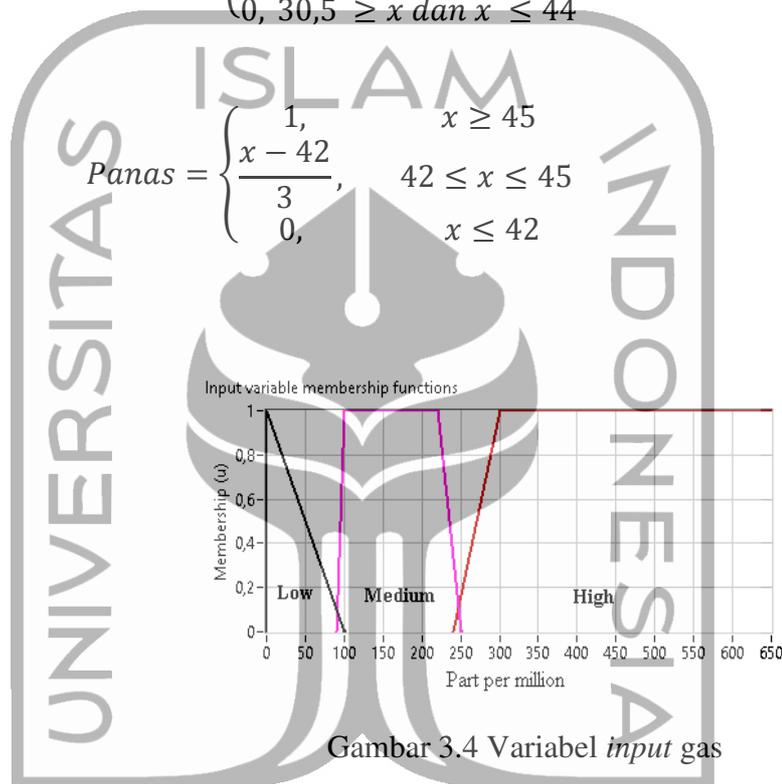
Gambar 3.3 Variabel *input temperature*

Pada Gambar 3.3 dibuat 3 buah variabel *linguistik*, yang mana diantaranya adalah Normal (0 °C sampai 35 °C), Sedang (30,5°C sampai 44 °C) dan panas (42 °C sampai seterusnya). Sehingga pada Gambar 3.3 jika dijabarkan akan membentuk persamaan sebagai berikut :

$$Normal = \begin{cases} 1, & x \leq 31 \\ \frac{35-x}{4}, & 31 \leq x \leq 35 \\ 0, & x \geq 35 \end{cases} \quad (3.2)$$

$$Sedang = \begin{cases} 1, & 36 \leq x \leq 42 \\ \frac{44-x}{2}, & 31 \leq x \leq 35 \\ \frac{x-30,5}{5,5}, & 31 \leq x \leq 35 \\ 0, & 30,5 \geq x \text{ dan } x \leq 44 \end{cases} \quad (3.3)$$

$$Panas = \begin{cases} 1, & x \geq 45 \\ \frac{x-42}{3}, & 42 \leq x \leq 45 \\ 0, & x \leq 42 \end{cases} \quad (3.4)$$



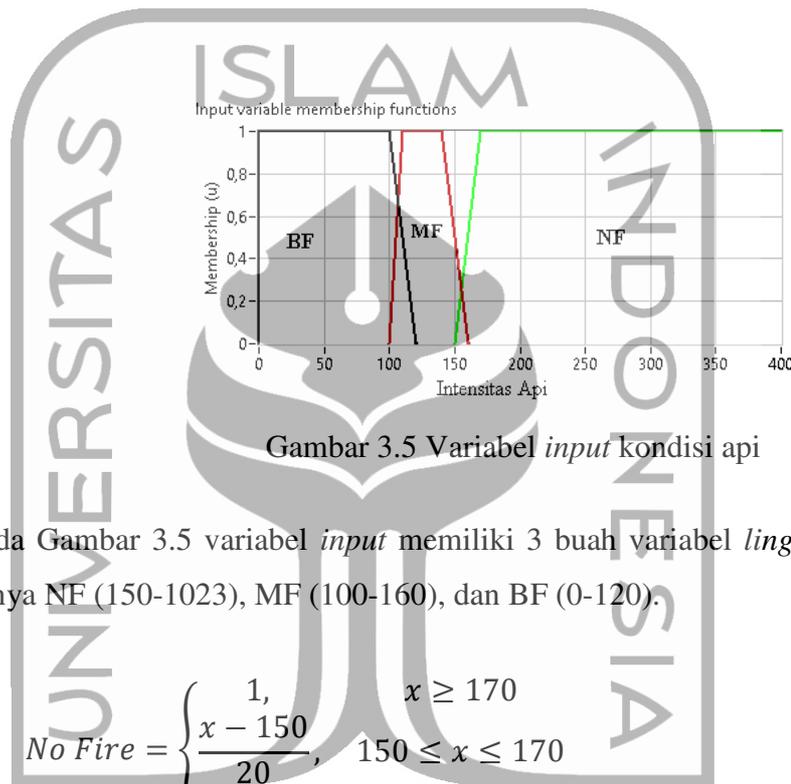
Gambar 3.4 Variabel *input* gas

Pada Gambar 3.4 variabel *input* gas juga memiliki 3 buah variabel *linguistik* yaitu diantaranya Low (0 ppm – 100 ppm), Medium (90 ppm – 250 ppm), dan High (240 ppm – 700 ppm).

$$Low = \begin{cases} 1, & x \leq 0 \\ \frac{100-x}{100}, & 0 \leq x \leq 100 \\ 0, & x \geq 100 \end{cases} \quad (3.5)$$

$$Medium = \begin{cases} 1, & 100 \leq x \leq 220 \\ \frac{250 - x}{30}, & 220 \leq x \leq 250 \\ \frac{x - 90}{10}, & 90 \leq x \leq 100 \\ 0, & x \geq 250 \text{ dan } x \leq 90 \end{cases} \quad (3.6)$$

$$High = \begin{cases} 1, & x \geq 300 \\ \frac{x - 240}{60}, & 240 \leq x \leq 300 \\ 0, & x \leq 240 \end{cases} \quad (3.7)$$



Gambar 3.5 Variabel *input* kondisi api

Pada Gambar 3.5 variabel *input* memiliki 3 buah variabel *linguistik* juga yaitu diantaranya NF (150-1023), MF (100-160), dan BF (0-120).

$$No\ Fire = \begin{cases} 1, & x \geq 170 \\ \frac{x - 150}{20}, & 150 \leq x \leq 170 \\ 0, & x \leq 150 \end{cases} \quad (3.8)$$

$$Medium\ Fire = \begin{cases} 1, & 110 \leq x \leq 140 \\ \frac{160 - x}{20}, & 140 \leq x \leq 160 \\ \frac{x - 100}{10}, & 100 \leq x \leq 110 \\ 0, & x \geq 160 \text{ dan } x \leq 100 \end{cases} \quad (3.9)$$

$$Big\ Fire = \begin{cases} 1, & x \leq 100 \\ \frac{120 - x}{20}, & 100 \leq x \leq 120 \\ 0, & x \geq 120 \end{cases} \quad (3.10)$$

Selain merancang variabel input *fuzzy*, pada tahap ini variabel *output* juga dilakukan perancangan dan perancangan variabel *output* pada sistem ini adalah sebagai berikut :



Gambar 3.6 Variabel *output* sistem *fuzzy*

Pada Gambar 3.6 dapat dilihat bahwa variabel *output* dari sistem *fuzzy* dirancang dengan memiliki 3 buah variabel *linguistik* dan ketiga variabel *linguistik* tersebut adalah Aman (1 – 2), Waspada (1,5 – 2,5), dan Bahaya (2 – 3). Dari variabel *output* inilah level pada suatu kebakaran dapat diprediksi oleh sistem pendeteksi kebakaran otomatis.

2. Membuat *rules* sistem *fuzzy*

Dikarenakan pada sistem pendeteksi kebakaran otomatis ini memiliki 3 variabel *input* dan masing-masing variabel *input* memiliki 3 buah variabel *linguistik* maka pada sistem ini *rules* yang harus dibuat sebanyak 27 *rules* yang mana *rule-rule* tersebut adalah sebagai berikut :

Tabel 3.1 *Rules* pada sistem *fuzzy* pendeteksi kebakaran otomatis

RULE	INPUT SISTEM FUZZY			OUTPUT
	TEMP	GAS	API	LEVEL
[R1]	Normal	Low	NF	AMAN
[R2]	Normal	Low	MF	WASPADA
[R3]	Normal	Low	BF	BAHAYA
[R4]	Normal	Medium	NF	WASPADA
[R5]	Normal	Medium	MF	WASPADA
[R6]	Normal	Medium	BF	BAHAYA
[R7]	Normal	High	NF	BAHAYA
[R8]	Normal	High	MF	BAHAYA
[R9]	Normal	High	BF	BAHAYA
[R10]	Sedang	Low	NF	WASPADA
[R11]	Sedang	Low	MF	WASPADA

RULE	INPUT SISTEM FUZZY			OUTPUT SISTEM FUZZY
	TEMP	GAS	API	LEVEL
[R12]	Sedang	Low	BF	BAHAYA
[R13]	Sedang	Medium	NF	WASPADA
[R14]	Sedang	Medium	MF	WASPADA
[R15]	Sedang	Medium	BF	BAHAYA
[R16]	Sedang	High	NF	BAHAYA
[R17]	Sedang	High	MF	BAHAYA
[R18]	Sedang	High	BF	BAHAYA
[R19]	Panas	Low	NF	BAHAYA
[R20]	Panas	Low	MF	BAHAYA
[R21]	Panas	Low	BF	BAHAYA
[R22]	Panas	Medium	NF	BAHAYA
[R23]	Panas	Medium	MF	BAHAYA
[R24]	Panas	Medium	BF	BAHAYA
[R25]	Panas	High	NF	BAHAYA
[R26]	Panas	High	MF	BAHAYA
[R27]	Panas	High	BF	BAHAYA

3. Membuat defuzzifikasi

Metode defuzzifikasi terdiri dari berbagai jenis metode dan pada perancangan sistem pendeteksi kebakaran otomatis ini metode defuzzifikasi yang digunakan adalah *Center of Maximum* (COM) yang mana metode ini memiliki persamaan sebagai berikut:

$$x_{final} = \frac{(x_1\mu_1 + x_2\mu_2 + \dots + x_n\mu_n)}{(\mu_1 + \mu_2 + \dots + \mu_n)} \quad (3.11)$$

Keterangan :

x_{final} = Hasil akhir dari defuzzifikasi

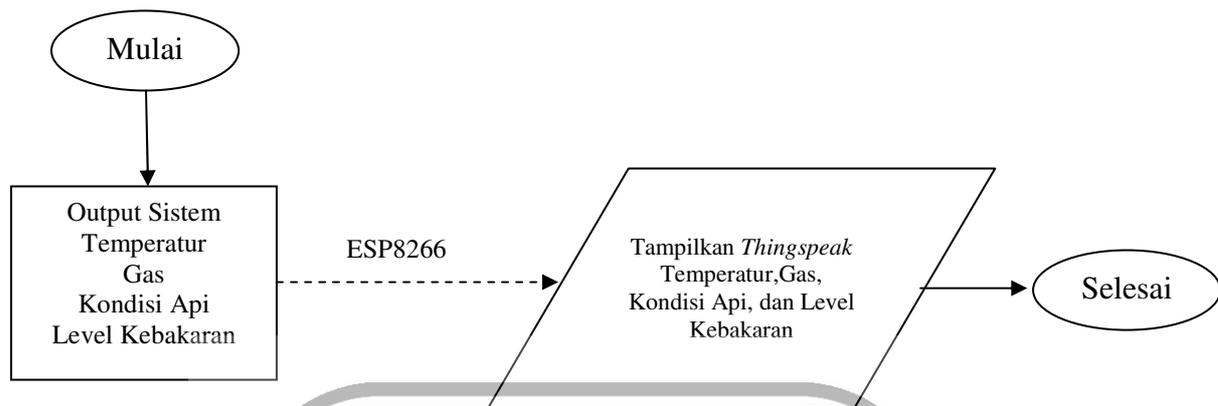
x_n = Nilai level kebakaran

μ_n = Nilai derajat keanggotaan dari fungsi keanggotaan n

3.7 Perancangan Internet of Things

Sistem pendeteksi kebakaran otomatis ini didesain berbasis dengan Internet of Things yang mana hasil dari sistem pendeteksi kebakaran otomatis akan dikirimkan ke web server dan dalam

penelitian ini web server yang digunakan adalah thingspeak. Ada pun perancangan IoT dari sistem pendeteksi kebakaran otomatis ini adalah sebagai berikut :



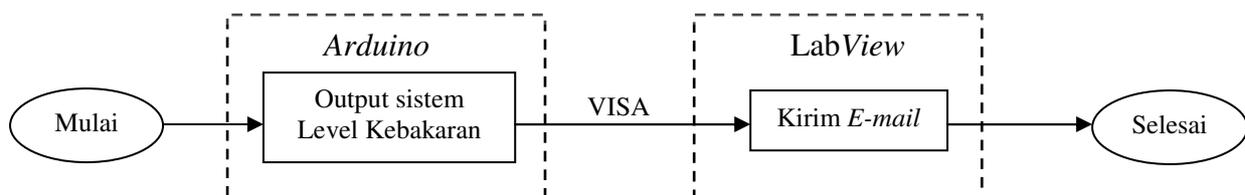
Gambar 3.7 Perancangan *thingspeak*

Gambar 3.7 menunjukkan perancangan IoT pada sistem pendeteksi kebakaran otomatis ini. Pada gambar memiliki beberapa tahapan dengan penjelasan sebagai berikut :

1. Output sistem ini terdiri dari 4 buah nilai yaitu nilai *temperature*, nilai kadar gas CO, nilai kondisi api, dan nilai level kebakaran. Keempat nilai dari output sistem ini yang akan dikirimkan ke *Thingspeak* selaku *web server* yang digunakan dengan pengiriman menggunakan *wireless network* berupa sinyal *Wifi* dengan menggunakan ESP8266.
2. Pada *Thingspeak* nantinya nilai keempat output sistem tersebut akan ditampilkan sehingga masing-masing nilai dapat dimonitoring oleh *user* melalui komputer maupun melalui *smartphone*.

3.8 Perancangan notifikasi *E-mail*

Sistem yang dirancang ini selain menggunakan *buzzer* sebagai indikator adanya sebuah kebakaran, sistem juga akan mengirimkan sebuah *e-mail* ke alamat *e-mail* milik *user* yang sudah ditentukan sebelumnya. Perancangan notifikasi *e-mail* ini memiliki tahapan-tahapan sebagai berikut :



Gambar 3.8 Perancangan notifikasi *e-mail*

Gambar 3.8 menunjukkan tahapan-tahapan pada perancangan notifikasi pengiriman *e-mail* guna sebagai indikator sistem tentang kondisi di suatu ruangan. Penjelasan tahap-tahap tersebut adalah sebagai berikut :

1. Output sistem yang digunakan untuk pengiriman *e-mail* hanya nilai dari level kebakaran yang mana nilai ini nantinya akan dikirimkan ke *LabView* dengan menggunakan komunikasi serial yaitu *VISA*.
2. Nilai level kebakaran yang sudah dikirimkan ke *LabView* kemudian akan diproses sehingga pesan yang akan dikirimkan ke *user* nantinya akan sesuai pada gambar perancangan software.

