

BAB IV

ANALISA PERANCANGAN

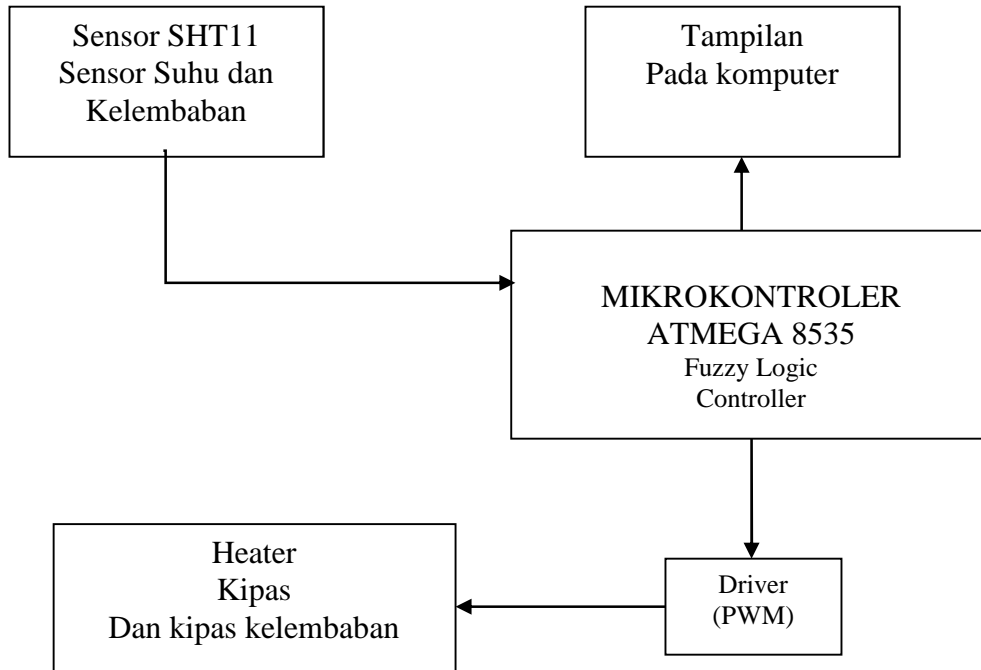
4.1 Analisa Perancangan

Secara umum prinsip kerja dari sistem pengontrolan suhu inkubator telur adalah sebagai berikut :

1. Dibagian inputan terdapat sensor SHT11 yang berguna untuk mendeteksi suhu sebagai pembaca suhu lingkungan. Disamping membaca suhu yang ada di dalam penetas telur, sensor SHT11 juga membaca kelembaban yang ada dalam penetas telur tersebut. Suhu dan kelembaban yang dibaca ini kemudian akan di bandingkan dengan set point yang telah diberikan dalam mikrokontroler, dimana suhu dan kelembaban yang dibaca atau diterima akan dikirimkan langsung menuju mikrokontroler.
2. Setelah data diterima oleh mikrokontroler, data akan diproses sesuai set point yang telah di tetapkan, dimana bila suhu berada pada 0 – 36 derajat *celcius* maka *heater* akan bekerja hingga titik 36 – 40 derajat *celcius*, sebalik nya bila suhu 40 – 90 derajat *celcius* maka kipas pendingin berputar. Pengaturan kelembaban dalam inkubator ini sama seperti pengaturan suhu, bila kelembaban berada pada 0 – 65 % RH maka kipas akan berputar meniup sebuah wadah yang berisi air hingga menciptakan gelombang sehingga menimbulkan uap air, bila kelembaban telah mencapai 65 – 70 maka kipas akan berhenti. Sebaliknya, bila kelembaban mencapai 70 – 90 % RH maka *heater* akan hidup dikarenakan bila kelembaban tinggi secara otomatis suhu akan turun.

Untuk mendapatkan sistem dengan prinsip kerja sebagaimana dijelaskan diatas, maka diperlukan proses perancangan yang meliputi perancangan rangkaian elektronika (rangkaian *driver*, sensor, dan rangkaian penghubung untuk mikrokontroller ATmega AVR)

4.2 Blok Diagram



Gambar 4.1 Block Diagram Proses Masukan dan Proses Keluaran.

Blok diagram pada gambar 4.1 menggambarkan sistem dari inkubator secara umum, seluruh sistem pada alat dikendalikan oleh mikrokontroler. SHT11 digunakan untuk mendeteksi suhu serta kelembaban ruangan inkubator. Keluaran dari rangkaian sensor tersebut langsung diproses oleh mikrokontroler untuk mengendalikan pergerakan *heater* serta pendingin.

Saat sensor membaca suhu ruangan, selanjutnya sensor akan mengirimkan data ke mikrokontroler. Setelah data diterima mikrokontroler, maka akan dilakukan proses fuzzifikasi dan defuzzifikasi yang mana akan menghasilkan nilai 0 hingga 1. Selanjutnya setelah data diolah dan dibandingkan dengan set point, maka mikrokontroler akan mengirimkan perintah kepada kipas, *heater* serta kipas kelembaban. Saat alat output bekerja, sensor akan terus membaca data lingkungan yang akan di teruskan dibandingkan dengan set point dan dilanjutkan dengan perintah kepada output.

4.3 Perancangan Mekanik

Perancangan mekanik dari inkubator ini meliputi bentuk desain, pemilihan alat, rangkaian elektronika, dan sebagainya.

4.3.1 Bentuk desain

Adapun perancangan ukuran dari inkubator ini

1. Dimensi inkubator : panjang = 30 cm ; lebar = 16 cm ; tinggi = 24 cm
2. Berat robot adalah $\pm 0,5$ kg
3. Material dinding inkubator adalah kaca akrilik,

Tidak ada spesifikasi yang mengharuskan bentuk dan ukuran dari box incubator, hanya saja dikarenakan ini merupakan simulasi pengontrolan suhu incubator telur, maka dibuat dalam bentuk yang mudah untuk dibawa.

4.3.2 Pemilihan alat

A. Sensor

Ada berbagai jenis sensor yang banyak beredar saat ini, antara lain LM35, SHT11 dan SHT75. Sensor-sensor ini memiliki fungsi yang hampir sama yaitu mendeteksi suhu, hanya saja sensor LM35 hanya mampu membaca suhu, sedangkan SHT11 dan SHT75 mampu membaca suhu dan kelembaban sekaligus serta memiliki kelebihan masing-masing. Sensor SHT11 merupakan sensor suhu serta kelembaban yang pertama diciptakan di dunia dan memiliki harga yang relatif murah di pasaran, sedangkan SHT75 memiliki daya baca sedikit lebih baik dari SHT11, karena SHT75 merupakan pengembangan dari SHT11 tetapi hal itu membuat harga dari SHT75 ini menjadi lebih mahal dari SHT11.

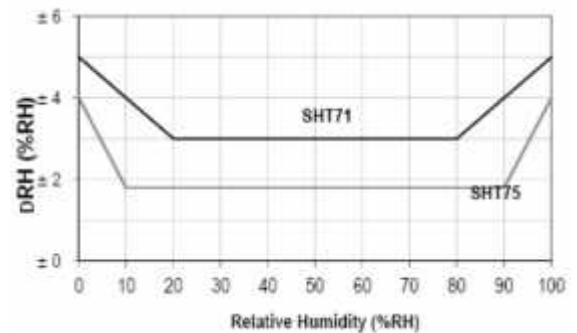
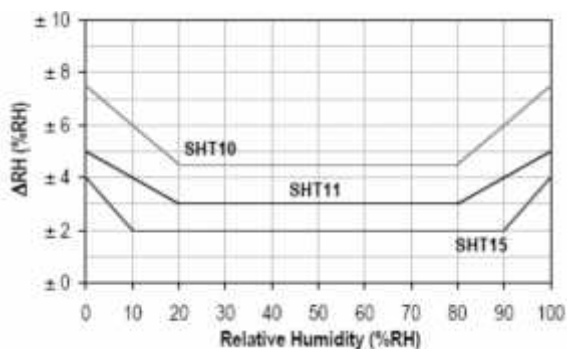
Berikut ini di tampilkan *data sheet performance* dari SHT11 dan SHT75 sebagai perbandingan :

Relative Humidity

Parameter	Condition	min	typ	max	Units
Resolution ¹		0.4	0.05	0.05	%RH
		8	12	12	bit
Accuracy ² SHT10	typical		±4.5		%RH
	maximal	see Figure 2			
Accuracy ² SHT11	typical		±3.0		%RH
	maximal	see Figure 2			
Accuracy ² SHT15	typical		±2.0		%RH
	maximal	see Figure 2			
Repeatability			±0.1		%RH
Hysteresis			±1		%RH
Non-linearity	linearized		<<1		%RH
Response time ³	τ (63%)		8		s
Operating Range		0		100	%RH
Long term drift ⁴	normal		< 0.5		%RH/yr

Relative Humidity

Parameter	Condition	min	typ	max	Units
Resolution ¹		0.4	0.05	0.05	%RH
		8	12	12	bit
Accuracy ² SHT71	typ		±3.0		%RH
	max	see Figure 2			
Accuracy ² SHT75	typ		±1.8		%RH
	max	see Figure 2			
Repeatability			±0.1		%RH
Hysteresis			±1		%RH
Nonlinearity	raw data		±3		%RH
	linearized		<<1		%RH
Response time ³	tau 63%		8		s
Operating Range		0		100	%RH
Long term drift ⁴	normal		< 0.5		%RH/yr



Gambar 4.2 Perbandingan RH dari Sensor SHT11 dan SHT75.

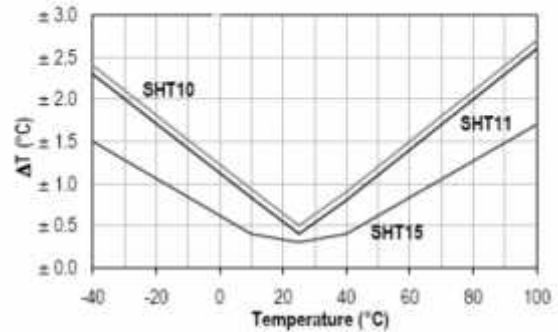
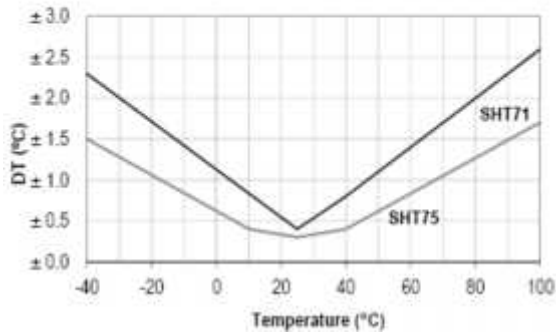
Dapat dilihat dari gambar 4.2, kemampuan akurasi pembacaan kelembaban SHT11 memiliki ± 3 %RH pada kondisi normal, sedangkan SHT75 memiliki keakurasian $\pm 1,8$ %RH dalam kondisi normal.

Temperature

Parameter	Condition	min	typ	max	Units
Resolution ¹		0.04	0.01	0.01	°C
		12	14	14	bit
Accuracy ² SHT71	typ		±0.4		°C
	max	see Figure 3			
Accuracy ² SHT75	typ		±0.3		°C
	max	see Figure 3			
Repeatability			±0.1		°C
Operating Range		-40		123.8	°C
		-40		254.9	°F
Response Time ³	tau 63%	5		30	s
Long term drift			< 0.04		°C/yr

Temperature

Parameter	Condition	min	typ	max	Units
Resolution ¹		0.04	0.01	0.01	°C
		12	14	14	bit
Accuracy ² SHT10	typical		±0.5		°C
	maximal	see Figure 3			
Accuracy ² SHT11	typical		±0.4		°C
	maximal	see Figure 3			
Accuracy ² SHT15	typical		±0.3		°C
	maximal	see Figure 3			
Repeatability			±0.1		°C
Operating Range		-40		123.8	°C
		-40		254.9	°F
Response Time ³	τ (63%)	5		30	s
Long term drift			< 0.04		°C/yr



Gambar 4.3 Perbandingan Suhu dari Sensor SHT11 dan SHT75.

Untuk pembacaan suhu berdasarkan gambar 4.3 dapat diketahui bahwa selisih pembacaan suhu antara SHT11 dan SHT75 hanyalah $\pm 0,1$ derajat celsius, dimana SHT11 memiliki selisih $\pm 0,4$ derajat celsius dalam suhu kamar serta SHT75 memiliki selisih $\pm 0,3$ derajat Celsius dalam suhu kamar. Berdasarkan perbandingan antara SHT11 dan SHT75 diatas, dimana memiliki fungsi yg sama hanya memiliki sedikit perbedaan kepekaan terhadap membaca, dipilihlah SHT11 dikarenakan dengan biaya lebih ringan serta bila digunakan untuk inkubator yang tidak

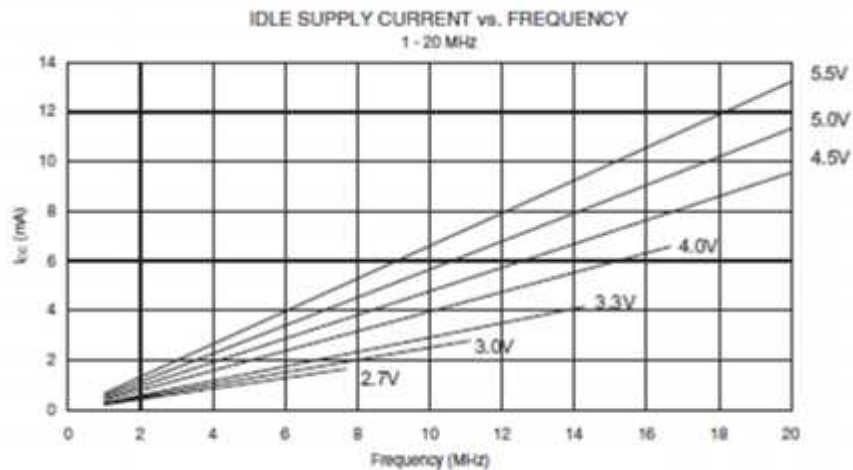
memperhitungkan derajat *celcius* yang lebih spesifik, SHT11 dan SHT75 hanya memiliki selisih pembacaan ± 0.3 derajat *celcius*.

B. Mikrokontroler

Mikrokontroler adalah CPU dalam sebuah sistem program, mengontrol semua aktifitas yang terhubung dengannya. Ada berbagai macam mikrokontroler yang disediakan, namun mikrokontroler yang umum digunakan untuk kegiatan instrument adalah mikrokontroler AVR, hal ini dapat ditinjau dari:

1. Kinerja Tinggi dan Daya Rendah

AVR (Alf (Egil Bogen) and Vegard (Wollan's Risc processor) mengeluarkan ATmega8 dengan fitur yang sangat baik. Dikatakan baik dikarenakan ATmega8 hemat energi (daya rendah), karena mampu beroperasi pada tegangan 2,7 sampai 5,5 Volt, dan hanya mengkonsumsi arus sebesar 3,6 mA.



Gambar 4.4 Kinerja AVR.

2. Arsitektur RISC dan Mikrokontroler AVR

Atmel AVR adalah jenis mikrokontroler yang paling sering dipakai dalam bidang elektronika dan instrumentasi. Mikrokontroler AVR ini memiliki arsitektur RISC delapan bit, di mana semua instruksi dikemas dalam kode 16-bit (*16 bits word*) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus *clock*. Register bekerja dengan 32x8 tujuan umum, mikrokontroler beroperasi sepenuhnya statik. Kecepatan mencapai 16 MIPS (*Million of Instructions Per Second*) ketika XTAL 16 MHz (Megahertz) dan terdapat 2 siklus pengganda (*multiplier*) di dalam Chip.

3. Fitur Perangkat Mikrokontroler AVR

Selain memiliki kinerja yang sangat baik, mikrokontroler ini pun memiliki fitur-fitur yang mampu menunjang kinerja menjadi lebih baik lagi seperti mempunyai dua buah *Timer/Counter* (pencacah) 8-bit dengan *Prescale* yang terpisah (berbeda-beda satu dengan yang lain), juga terdapat mode pembandingan (*Compare*), memiliki satu buah *Timer/Counter* (pencacah) 16-bit dengan *Prescale* yang terpisah, mode pembandingan dan mode *Capture*, mencacah (*Counter*) secara berkala (*real time*), mempunyai tiga saluran (*channel*) PWM (*Pulse Width Modulation*) dimana ini mampu mengatur daya keluaran berdasarkan kebutuhan, terdapat delapan saluran ADC (*Analog to Digital Converter*) sehingga tidak lagi memerlukan alat bantu sebagai pengkonversian data analog menjadi digital dan bisa mencapai 10-bit dengan akurasi yang tinggi, enam saluran ADC pada kemasan PDIP (*Plastic Dual Inline Package*), bisa mencapai 10-bit dengan akurasi tinggi, memiliki antarmuka serial dua kabel (*two-wire*) berorientasi *byte*, dapat diprogram dengan komunikasi Serial USART (*Universal Synchronous and Asynchronous serial Receiver and Transmitter*), memiliki antarmuka SPI (*Serial Peripheral Interface*) dengan tingkat *Master/Slave*, mempunyai *Watchdog* (monitoring) *Timer* dengan osilator yang terpisah

di dalam chip, serta memiliki *Analog Comparator* (pembanding analog) di dalam chip.

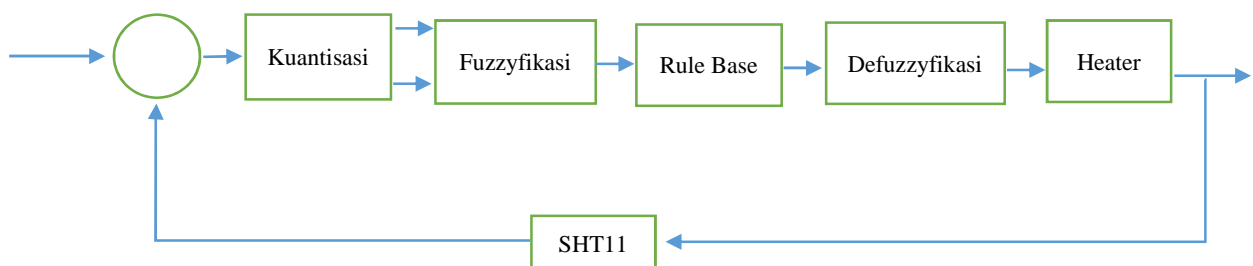
C. *Heater* (Elemen Pemanas)

Untuk menghasilkan ruang hangat maka diperlukan sebuah alat yang benar-benar mampu memberikan fungsi hangat yang sebenarnya. Beberapa penelitian menggunakan bohlam untuk menciptakan panas yang dibutuhkan oleh sebuah alat inkubator.

Berdasarkan fungsinya, bohlam bukanlah alat untuk menjadi sumber panas, melainkan sumber cahaya, oleh karena itu, penulis mencoba memilih alternatif lain untuk menghasilkan sumber panas tersebut dengan menggunakan *heater* (elemen pemanas) yang mana alat ini memang difungsikan sebagai alat untuk menciptakan panas, untuk itu penulis menggunakan elemen kompor listrik yang telah dipotong sekitar 20 cm, dengan hambatan sekitar 7 ohm yang diberikan tegangan sekitar 18 volt, diharapkan mampu memberikan panas yang cukup untuk inkubator telur.

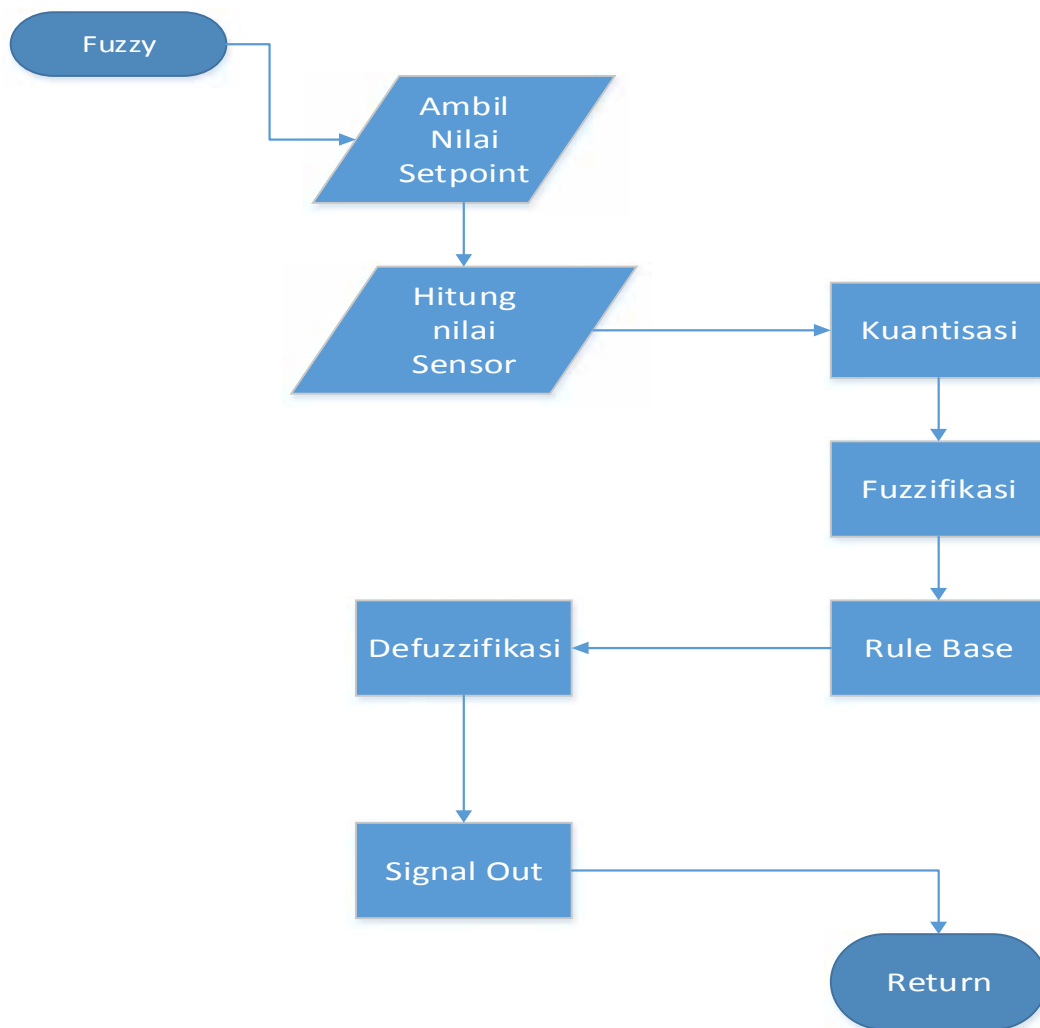
4.4 Analisa Sistem

Secara umum cara kerja sistem inkubator telur ini adalah menjaga suhu ruangan agar tetap stabil dari 36 – 40 derajat *celcius*, karena suhu tersebut adalah yang baik untuk menetas telur. Untuk menjaga suhu stabil maka diperlukanlah kontrol *fuzzy*. Berikut ini struktur pengendali logika *fuzzy* dalam sistem:



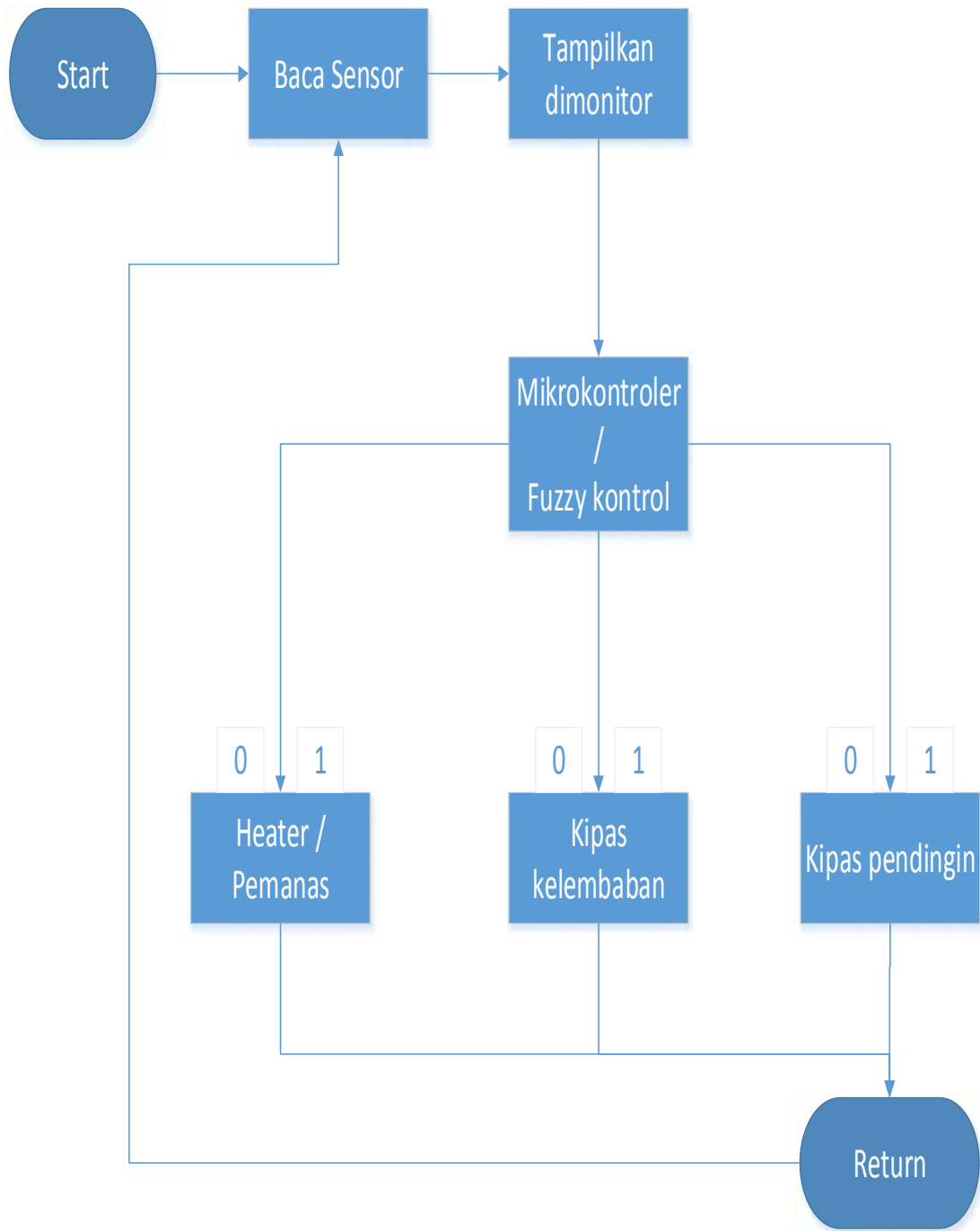
Gambar 4.5 Struktur Kontrol Logika *Fuzzy*.

Pada alat inkubator ini proses *control fuzzy* bekerja pada mikrokontroler, dimana setelah data di terima dari sensor suhu, nilai dari sensor suhu akan dibandingkan dengan nilai crisp input yang telah di berikan sebelumnya. Nilai tersebut akan difuzzifikasi dimana hasilnya akan dimasukkan kedalam *rule base*, dimana nilai *fuzzy* yang telah diperoleh akan diterjemahkan menjadi nilai-nilai logika *IF-THEN*, setelah melewati tahap *rule base* selanjutnya adalah defuzzifikasi dimana mengembalikan nilai logika menjadi nilai nyata yaitu 0 – 1. Berikut *flowchart* dari proses *fuzzy*:



Gambar 4.6 Flowchart *Fuzzy*.

Untuk keseluruhan flowchartnya dapat dilihat sebagai berikut:



Gambar 4.7 Flowchart Sistem.

a. Pengaturan *Heater*

Pengaturan *heater* dimana kondisi *range* suhu adalah 0 hingga 90 derajat celcius. Nilai crisp “Rendah” untuk *range* antara 0 hingga 36, “Pas” bila suhu antara 36 hingga 40 serta “Tinggi” pada suhu 40 hingga 90. Untuk kategori “Rendah” maka *heater* akan diberi nilai 1, sementara itu jika ketegori “Pas” dan “Tinggi” diberi nilai 0.

Tabel Nilai Crisp *Heater* berdasarkan nilai suhu.

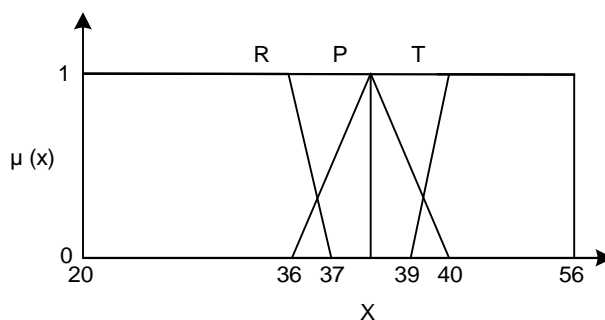
Tabel 4.1 Nilai Crisp Berdasarkan Suhu

Range Suhu	Kategori	Nilai Crisp
0 – 36	Rendah	1
36 – 40	Pas	0
40 – 90	Tinggi	0

b. Pengaturan Kipas

Pada pengaturan kipas saat kondisi suhu “Rendah”, kipas akan berputar. Perputaran kipas dipengaruhi oleh Pulse Width Modulation berdasarkan fungsi keanggotaan, dimana bila suhu semakin jauh dibawah kategori “Pas” maka kipas akan semakin kencang meniupkan panas dari sumber panas (*Heater*), agar panas cepat merata.

Untuk perhitungan fungsi keanggotaan adalah sebagai berikut:



Gambar 4.8 Kurva Suhu .

Keterangan :

R= Rendah

P= Pas

T= Tinggi

Contoh fungsi keanggotaan:

Fungsi keanggotaan untuk suhu “Rendah” pada variable suhu.

$$\mu[x]= \left\{ \begin{array}{ll} 1; & x \leq b \\ \frac{b-x}{b-a}; & a \leq x \leq b \\ 0; & x \geq b \end{array} \right\}$$

Bila suhu berada di bawah 36 derajat celcius, maka kipas akan berputar di beri nilai 1.

$$\mu_{\text{Rendah}}[36] = 1$$

Sedangkan untuk yang berada antara “Pas” dan “Rendah” yaitu antara 36 hingga 37 maka untuk mengukur derajat keanggotaan nya adalah sebagai berikut:

$$\begin{aligned} \mu_{\text{Rendah}}[36,6] &= (37-36,6)/(37-36) \\ &= 0,4/1 \\ &= 0,4 \end{aligned}$$

Jadi, untuk suhu 36.6 derajat celcius memiliki derajat keanggotaan 0,44, sehingga perputaran kipas menjadi 0,4 x 100% dari jumlah voltase penuh.

Bila suhu berada di atas 37 derajat Celsius maka bernilai 0.

$$\mu_{\text{Rendah}}[37] = 0$$

Tidak berbeda jauh dengan pengaturan kipas pada suhu “Rendah”, pada suhu “Tinggi” juga sama, hanya saja pada kondisi suhu “Rendah” representasi linearnya adalah menurun, sedangkan kan pada suhu “Tinggi” menggunakan representasi linear menaik.

Fungsi keanggotaan suhu “Tinggi”

Contoh fungsi keanggotaannya:

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

Bila suhu dibawah 39 derajat celcius, maka kipas akan diberi nilai 0.

Untuk suhu berada antara 39 – 40, maka:

$$\begin{aligned} \mu_{Tinggi}[39,7] &= (39,7-39)/(40-39) \\ &= 35/90 \\ &= 0,7 \end{aligned}$$

Sehingga, untuk keanggotaan pada suhu 39,7 derajat celcius adalah 0,7, dengan demikian untuk perputaran kipas menjadi 0,7 x 100% dari voltase penuh yang diberikan. Setelah suhu berada di atas 40 maka kipas akan bernilai 1 atau dalam PWM menjadi 100% dari voltase 12V.

Suhu berada diatas 40 derajat Celsius, kipas akan bernilai 1.

Tabel nilai perputaran kipas berdasarkan suhu:

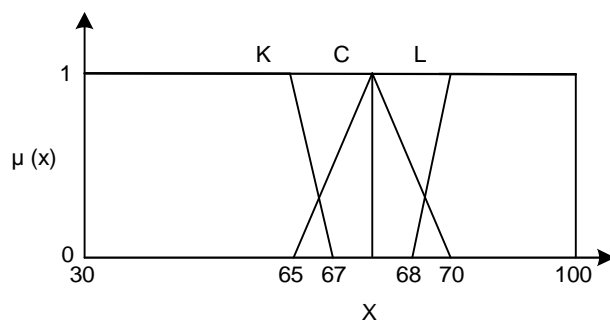
Tabel 4.2 Nilai Perputaran Kipas Berdasarkan Suhu

Range Suhu	Kategori	Nilai Crisp
0 – 36	Rendah	1
36 – 40	Pas	0
40 – 90	Tinggi	1

c. Pengaturan kipas kelembaban

Pada kelembaban terdapat 3 kategori yaitu “Kurang” berada antara 0 hingga 65 %RH, “Cukup” berada pada 60 hingga 75 %RH, serta “Berlebih” diatas 70 hingga 100%RH.

Penentuan keanggotaan perputaran kipas kelembaban hanya menggunakan representasi linear menurun, karena kipas kelembaban akan berputar bila kondisi kelembaban berada di bawah kategori “Cukup”..



Gambar 4.9 Kurva Kelembaban Kurang

Keterangan:

K= Kurang

C= Cukup

L= Lebih

Fungsi keanggotaan untuk kelembaban “Kurang” pada variable kelembaban.

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

Bila kelembaban dibawah 65 derajat Celsius, maka nilai untuk kipas kelembaban adalah 1.

Untuk kondisi kelembaban berada pada 65-67 %RH maka:

$$\begin{aligned} \mu_{\text{Kurang}}[66] &= (67-66)/(67-65) \\ &= 1/2 \\ &= 0.5 \end{aligned}$$

Jadi, untuk kelembaban “Kurang” pada pembacaan 63 %RH maka derajat keanggotaan nya adalah 0.4 sehingga nilai derajat keanggotaan ini akan di kali kan dengan 100% voltasi yang diberikan sebagai bentuk pengaturan PWM nya.

Kelembaban dengan kondisi diatas 67 %RH maka nilai dari kipas kelembaban adalah 0.

Tabel nilai perputaran kipas kelembaban berdasarkan kelembaban:

Tabel 4.3 Nilai Perputaran Kipas Berdasarkan Kelembaban

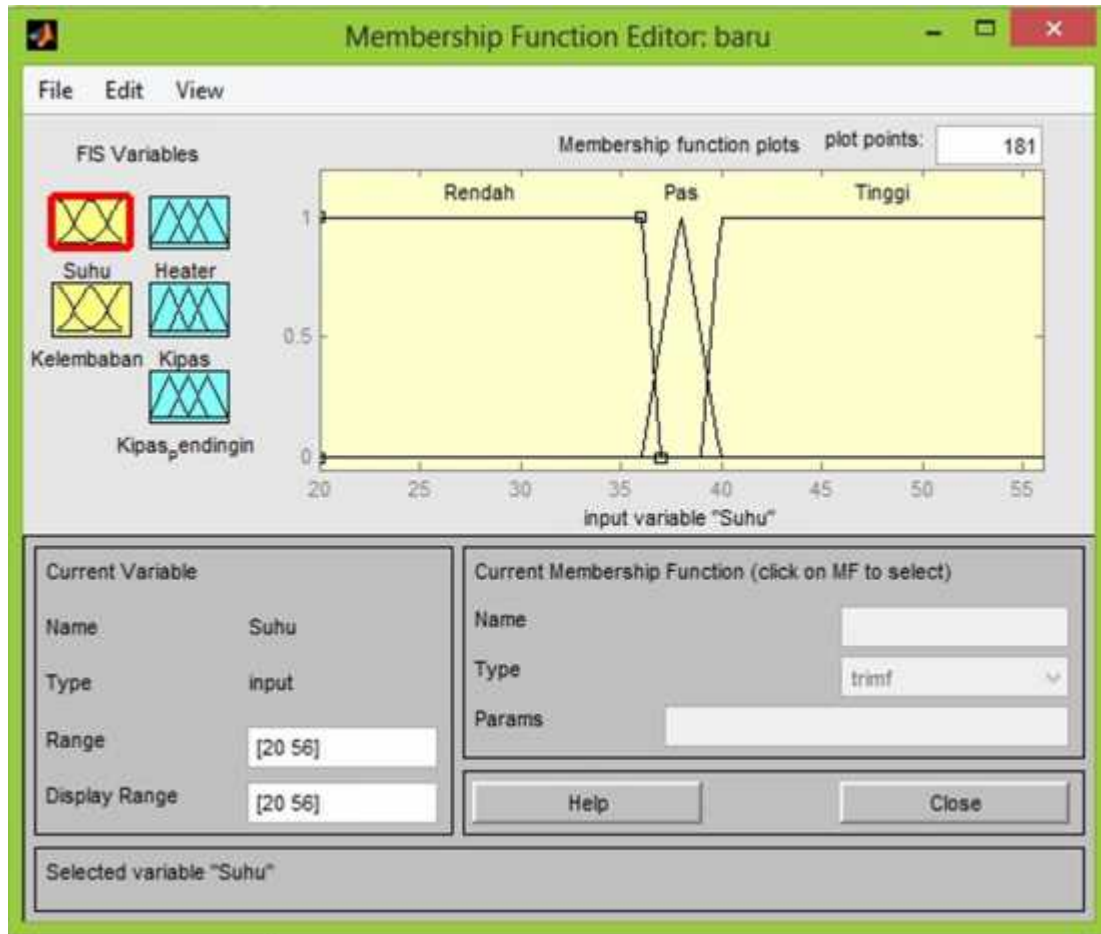
Range Kelembaban (%RH)	Kategori	Nilai Crisp
0 – 65	Cukup	1
65 - 70	Pas	0
70 - 100	Berlebih	0

4.5 Perancangan Sistem

Untuk mempermudah pembuatan sistem yang diinginkan, maka diperlukan perancangan menggunakan *software* Matlab dengan mensimulasikan kontrol *fuzzy* yang akan dibuat berdasarkan nilai – nilai input serta nilai – nilai output.

Berikut ini adalah desain memebership input, output serta *rule base fuzzy* dengan menggunakan *software* Matlab:

1. *Membership* input suhu



Gambar 4.10 Membership Suhu.

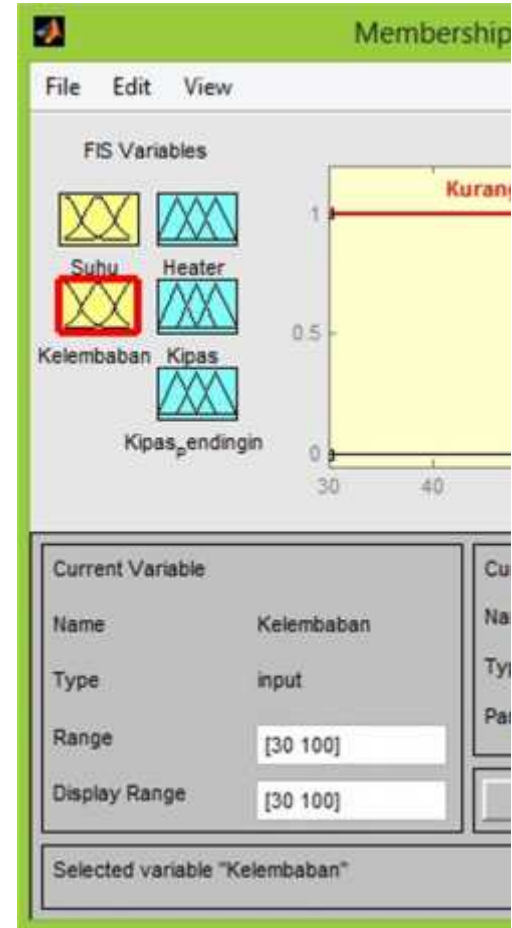
Berdasarkan penelitian, suhu yang pas untuk menetas telur adalah suhu yang berkisar antara 36 - 40 derajat *Celcius*, sehingga parameternya menjadi suhu 0 – 36 derajat adalah rendah, 36 – 40 derajat adalah pas dan 40 – 90 adalah tinggi.

Tabel 4.4 Parameter Suhu

No.	Parameter ($^{\circ}\text{C}$)	Member Fungsion
1.	0 - 36	Rendah
2.	36 - 40	Pas
3.	40 - 90	Tinggi

2. *Membership* input kelembaban

No.	Parameter (RH%)	Member Fungsion
-----	-----------------	-----------------



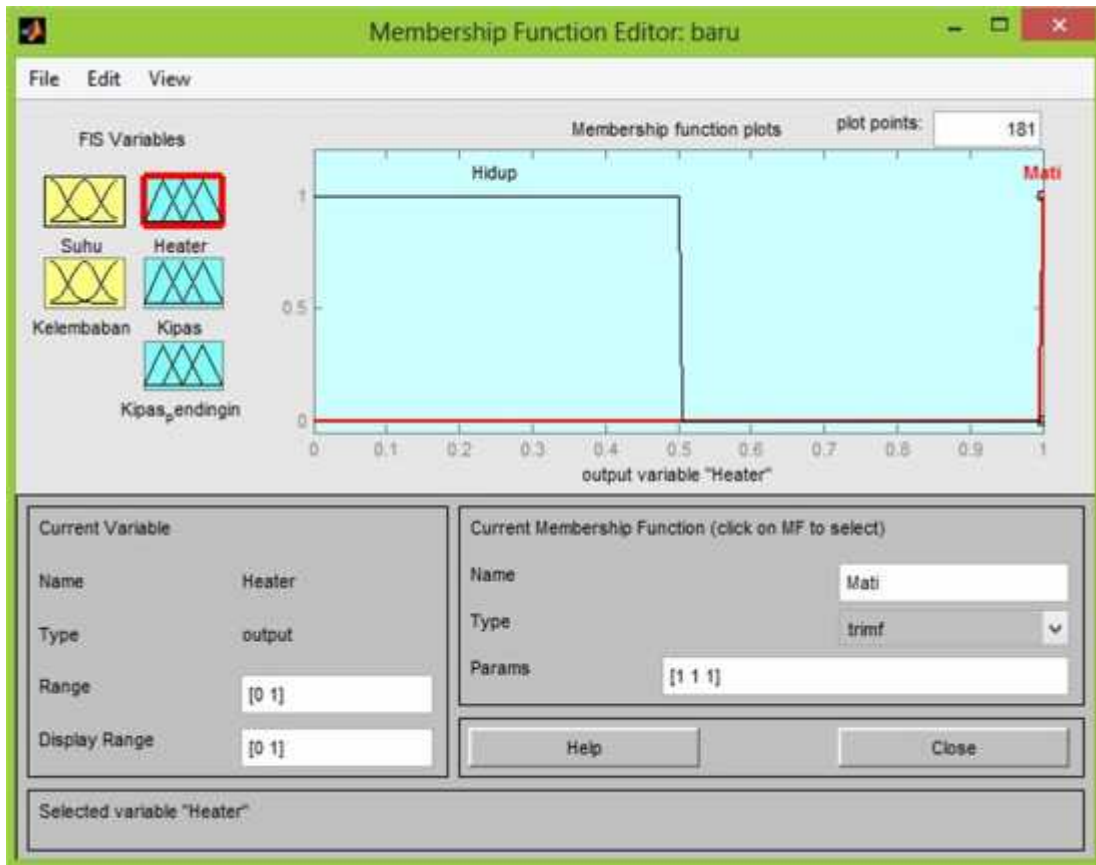
Gambar 4.11 *Membership* Kelembaban.

Tidak berbeda jauh dengan suhu, kelembaban yang merupakan faktor pendukung, tidak begitu diperhatikan namun untuk kelembaban yang diberikan pada inkubator telur adalah antara 65 hingga 70 % RH, dimana parameternya menjadi 30 – 65 % RH adalah kurang, 65 – 70 % RH adalah cukup dan 70 – 100 % RH adalah berlebih.

1.	0 - 65	Rendah
2.	65 - 70	Pas
3.	70 - 100	Tinggi

Tabel 4.5 Parameter Kelembaban

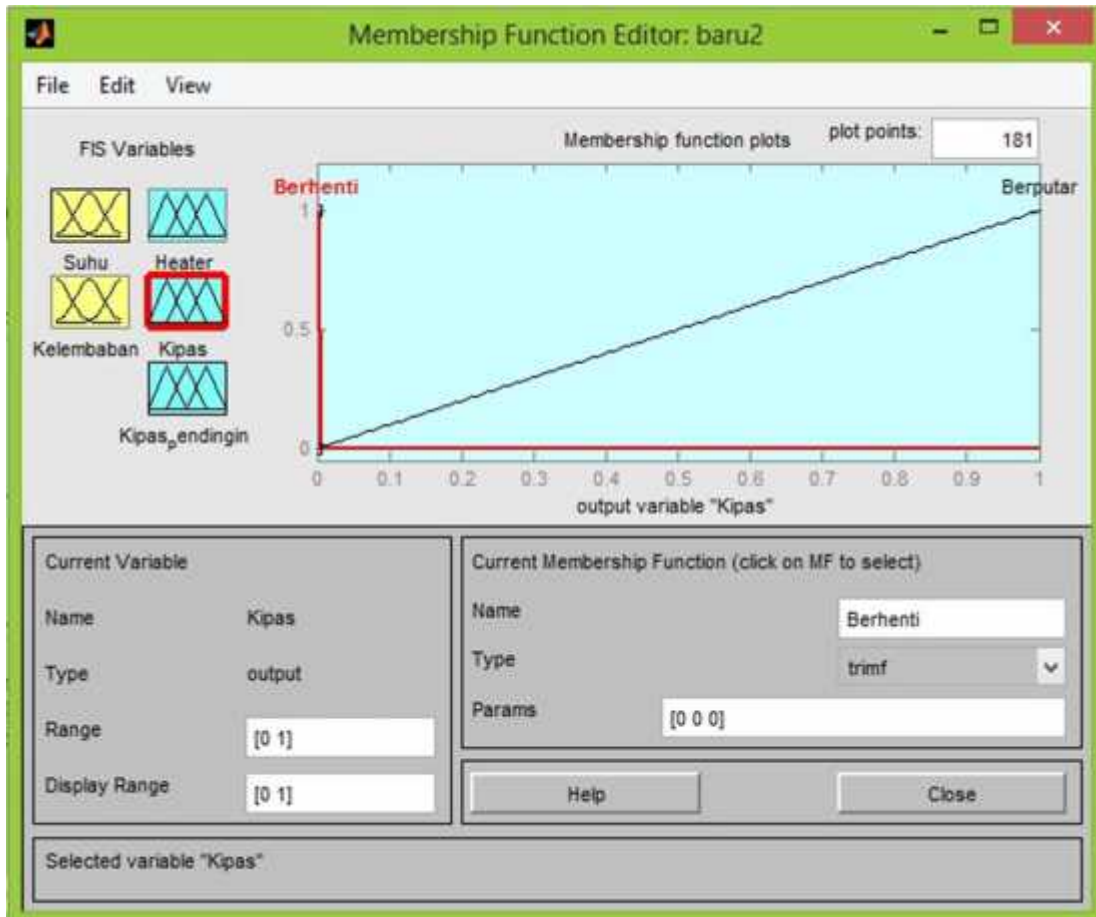
3. *Membership output heater*



Gambar 4.12 Membership Heater.

Untuk output hanya diberikan 2 parameter, yaitu hidup dan mati atau berputar dan berhenti. Nilai parameter pada output adalah 0 – 1, dimana pada *heater* untuk nilai output 1 maka hidup dan bila 0 maka mati.

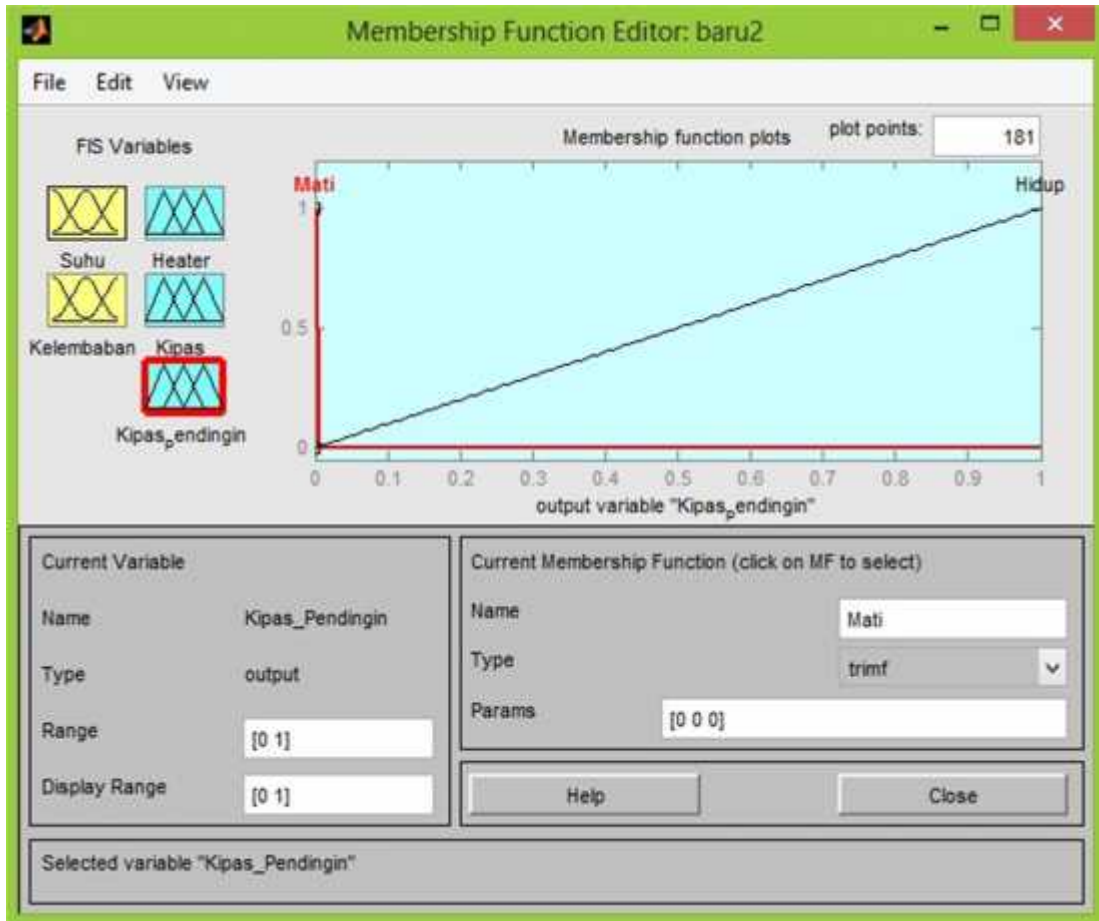
4. Membership output kipas



Gambar 4.13 Membership Kipas.

Untuk kipas juga diberikan dua buah *membership* fungsi, yaitu berputar dan berhenti. Fungsi dari kipas ini adalah membantu menaikkan dan menurunkan kelembaban dari ruangan tersebut, dimana akan diberikan sebuah wadah yang berisi air dan air tersebut akan ditiup oleh kipas sehingga air tersebut akan mempengaruhi tinggi rendahnya kelembaban.

5. Kipas Pendingin

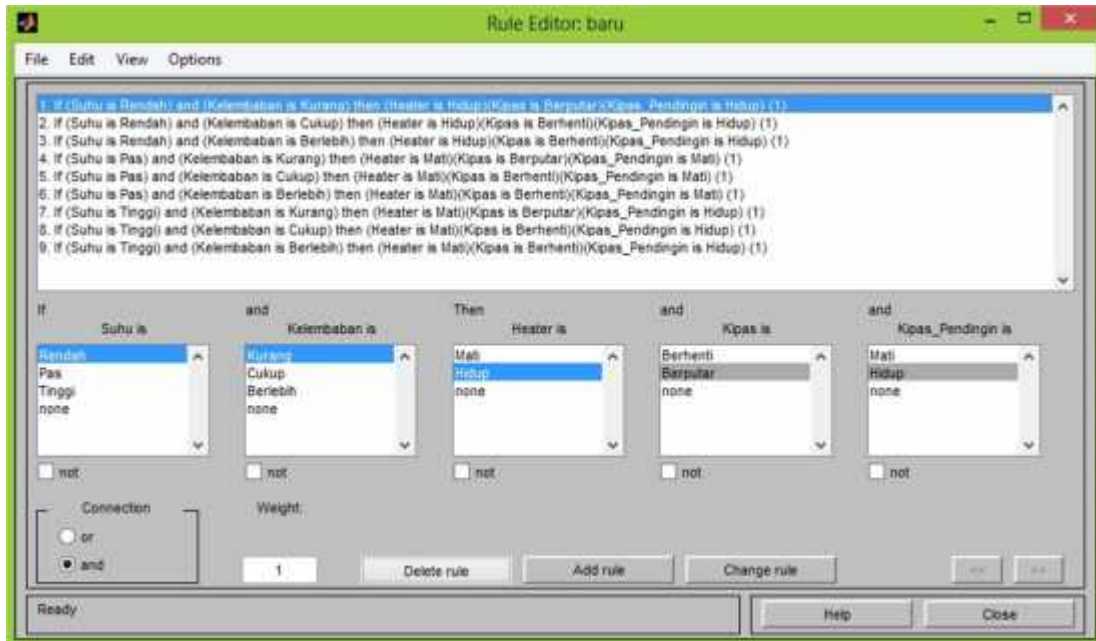


Gambar 4.14 Membership Kipas Pendingin.

Kipas pendingin berfungsi dimana kondisi bila ruangan inkubator melebihi suhu yang diinginkan, untuk parameternya sama seperti output yang lain yaitu 0 – 1.

6. Rule base

Pada *rule base* parameter – parameter yang ada akan dihubungkan sehingga menghasilkan sebuah bentuk logika sederhana. Berikut penyusunan *rule base* dalam *fuzzy control* menggunakan matlab:



Gambar 4.15 Rule Base 1.

Rule base ini nantinya akan diterjemahkan dalam bahasa C++ agar mikrokontroler mampu memahami perintah logika *fuzzy*. Berikut terjemahan *rule base* dalam bahasa C++.

```

if ((t_C < m_iTmin) && (t_C > 0))
{
    if ((rh_true < m_iRHmin) && (rh_true > 0))
    {
        HTR_Controller = 1;
        RH_Controller = (100 * (m_iRHmin -
rh_true) / (m_iRHmin - 0));
        fan_Controller = (100 * (m_iTmin -
t_C) / (m_iTmin - 0));
    }
    else
if ((rh_true >= m_iRHmin) && (rh_true <= m_iRHmax))
{
        HTR_Controller = 1;
        RH_Controller = 0;
        fan_Controller = (100 * (m_iTmin -
t_C) / (m_iTmin - 0));
    }
}

```

```

else if((rh_true>m_iRHmax)&&(rh_true<100))
{
    HTR_Controller=1;
    RH_Controller=0;
    fan_Controller=(100*(m_iTmin-
t_C)/(m_iTmin-0));
}
}
else if((t_C>=m_iTmin)&&(t_C<=m_iTmax))
{
    if((rh_true<m_iRHmin)&&(rh_true>0))
    {
        HTR_Controller=0;
        RH_Controller=(100*(m_iRHmin-
rh_true)/(m_iRHmin-0));
        fan_Controller=0;
    }
    else
if((rh_true>=m_iRHmin)&&(rh_true<=m_iRHmax))
    {
        HTR_Controller=0;
        RH_Controller=0;
        fan_Controller=0;
    }
    else if((rh_true>m_iRHmax)&&(rh_true<100))
    {
        HTR_Controller=0;
        RH_Controller=0;
        fan_Controller=0;
    }
}
else if((t_C>m_iTmax)&&(t_C<90))
{
    if((rh_true<m_iRHmin)&&(rh_true>0))
    {
        HTR_Controller=0;
        RH_Controller=(100*(m_iRHmin-
rh_true)/(m_iRHmin-0));
        fan_Controller=(100*(t_C-m_iTmax)/(90-
m_iTmax));
    }
    else
if((rh_true>=m_iRHmin)&&(rh_true<=m_iRHmax))

```

```

    {
        HTR_Controller=0;
        RH_Controller=0;
        fan_Controller=(100*(t_C-m_iTmax)/(90-
m_iTmax));
    }
else if((rh_true>m_iRHmax)&&(rh_true<100))
{
    HTR_Controller=0;
    RH_Controller=0;
    fan_Controller=0;
}
}

```

Setelah *rule base* dibentuk berdasarkan parameter – parameter yang ada, bila diimplementasikan kesistem maka akan mendapatkan gambaran seperti berikut:



Gambar 4.16 Rule Base 2.