

## BAB III

### METODE PENELITIAN

#### 3.1 Metode Penelitian

Metode penelitian pada penelitian ini adalah mengatur suhu di dalam ruangan kandang *brooding* ayam sesuai keinginan *user*. Bisa dikatakan adalah sistem kontrol pada kandang *brooding* ayam yang efektif mengontrol titik suhu pada lampu pemanas berapa persen sesuai kebutuhan yang diperlukan dengan *setting point* yang di inputkan melalui *keypad*. untuk mengatur sirkulasi udara pada *fan* dalam penurunan suhu. Aktuator kipas dikontrol melalui arduino dengan pelan dan cepat putaran pada kipas untuk mengatur sirkulasi udara mengeluarkan udara panas dan menghembuskan udara membantu penurunan suhu. Dengan begitu ada 2 aktuator yang membantu mengontrol suhu. Input pada *keypad* yaitu berupa suhu yang ditentukan oleh *user* dengan *setting point* yang *flexible*. Dengan sensor LM35 sebagai inputan suhu yang diproses oleh arduino. Ketika *inputan* dari *keypad* berupa suhu serta keadaan awal suhu yang ditampilkan di LCD.

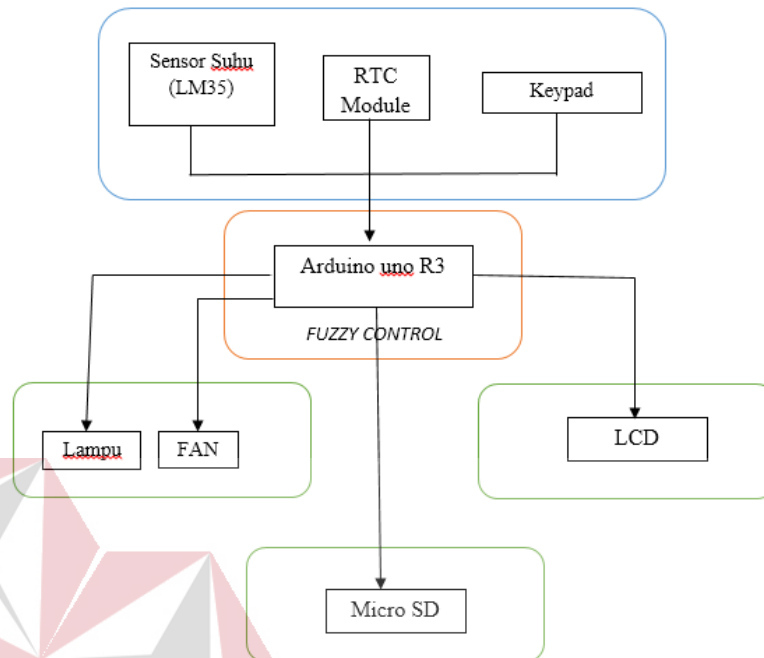
Dengan menerapkan *system RTC (Real Time Clock)* untuk menentukan waktu seperti jam, hari, dan tanggal sesuai kebutuhan sehingga bisa menentukan umur ayam secara efisien. Yang akan ditampilkan pada LCD dibagian bawah layar. dengan nilai PWM maksimal yang akan menghidupkan 2 kipas yang berada pada sisi-sisi kandang *brooding*. Jika nilai suhu pada ruangan tidak sesuai dengan *input* dari *user* maka arduino akan mengirim sinyal ke relay dan *dimmer* untuk mengontrol system pemanas pada lampu dan *fan* untuk membantu penurunan maupun menaikkan suhu.

Arduino berfungsi sebagai pengontrol yang mendapat masukan suhu sesuai dengan *setting point* yang diinputkan, suhu awal sudah dideteksi oleh sensor suhu. dan menerapkan *micro sd module* yang diolah di Arduino Uno sebagai tempat penyimpanan data seperti *database* : yang berisi nilai suhu pada kandang *brooding* ,perubahan suhu dari waktu, tanggal, jam dan hari. Kipas akan terus menyala walaupun suhu sudah tercapai untuk menjaga sirkulasi udara pada ruangan. Data suhu yang berupa data *digital* mempermudah untuk mendeteksi suhu sekaligus *monitoring*.

Jika pada *setting point input* suhu tidak sesuai, maka aktuator akan aktif dan menyesuaikan suhu yang dibutuhkan pada kondisi umur yang dibutuhkan anak ayam kampung, sistem ini terus menyesuaikan kebutuhan umur anak ayam menyesuaikan *input* dari suhu, kecuali sistem *input* diganti *user* untuk kebutuhan maka aktuator kipas, lampu pemanas karena *control* pada kandang *brooding* ini di *setting* dengan sistem yang menyesuaikan suhu yang dapat diturunkan dan dinaikkan sesuai nilai yang di harapkan oleh *user*.

### 3.2 Perancangan Sistem

Berikut gambar Diagram blok pada sistem kontrol suhu kandang *brooding*.



Gambar 3.1 Blok Diagram Sistem

Dari gambar Blok Diagram Sistem dijelaskan beberapa *input*, *process* dan *output* yang digunakan antara lain :

a. *Input*

1. Sensor LM35 : Digunakan untuk mendeteksi nilai suhu.
2. *Keypad* berguna untuk input suhu yang di inginkan oleh user.  
pada kandang *brooding* ayam.
3. *RTC module* : berfungsi untuk menampilkan data berupa waktu.

b. *Process*

4. Arduino UNO berfungsi mengolah hasil dari *input* dari sensor dan *keypad* yang nantinya akan diproses menggunakan *fuzzy control* kemudian hasilnya akan dikirim melalui *output*.

c. *Output*

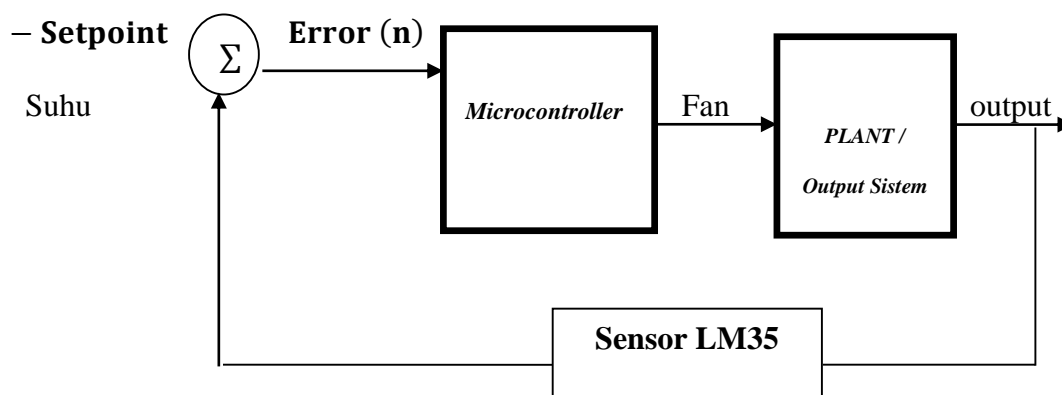
5. LCD : Digunakan untuk menampilkan suhu dan waktu jam tanggal dari RTC pada kandang *brooding*.
6. *Fan* : Untuk sirkulasi udara pada kandang *brooding* menggunakan kontrol *fuzzy*.
7. Lampu : digunakan sebagai aktuator pemancar pemanas pada kandang *brooding* agar suhu di ruangan tetap stabil hangat.
8. Micro SD *module* : sebagai keluaran penyimpanan data yang berisi perubahan suhu, waktu meliputi tanggal, jam, menit dan detik.

d. Penjelasan Blok Diagram

Pada penjelasan Blok Diagram ini, mula-mula sensor mendeteksi suhu pada sensor LM35 ketika *user input* nilai suhu pada *keypad* yang menjadi *setting point* pada sistem yang ditampilkan di *LCD*, dan lampu pemanas akan aktif baik kondisi redup maupun terang untuk menjaga kondisi dalam ruangan tetap hangat. Bila suhu dalam ruangan terlalu panas maka kondisi *fan* aktif secara penuh ataupun sedang dengan otomatis sesuai kondisi yang dibutuhkan pada ruangan. *outputkan* ketika *setpoint* terpenuhi 2 aktuator akan menyesuaikan nilai *fuzzy* secara otomatis dan suhu yang diinputkan akan di proses oleh arduino yang menjalankan *Motor Driver* untuk mengatur lama kipas yang aktif secara bersamaan sampai terpenuhinya *setpoint*, suhu yang diinputkan, maka kipas akan mengatur pelan, normal dan cepat berputar. Jika tidak, maka sensor akan menampilkan kondisi *setpoint* suhu pada tampilan *LCD*. Penggunaan *RTC module* menerapkan hasil waktu hari, jam dan tanggal secara *real time* sebagai penentuan perubahan suhu dari waktu ke waktu. hasil *output* penyimpanan data menggunakan Micro SD *module*.

### 3.3 Diagram Blok Pengendalian Sistem Kontrol pada Kandang *Brooding*

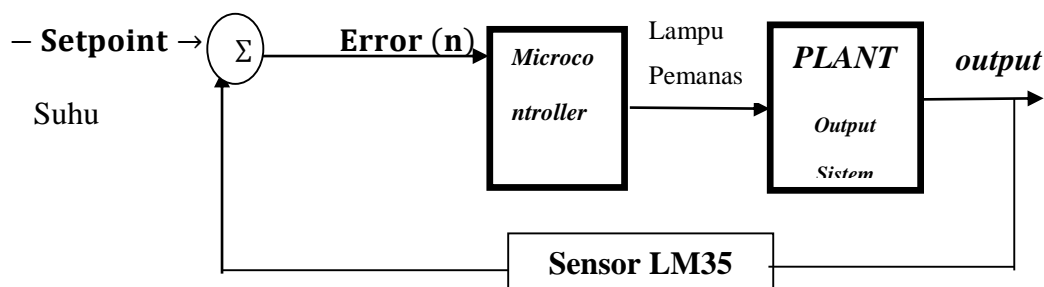
#### 3.3.1 Blok Pengendali Sistem *Fan*



Gambar 3.2 Digram Blok *Plant* dan Sistem Kendali *Fan*

Diagram sistem kendali *microcontroller* di atas terdiri dari input berupa *setting point* suhu dan *output* berupa *error* dan sedangkan *ouput* yang digunakan adalah kipas yang diproses pada sistem putaran pelan dan cepat *Control* secara bersamaan. *Setpoint* adalah *input* dari *user* berupa suhu, untuk mencapai perubahan *error* yang akan dideteksi menggunakan sensor LM35, aktuator akan aktif sesuai dengan *error* semakin banyak *error* pada *input* suhu maka semakin lama aktuator aktif menyesuaikan kebutuhan suhu.

#### 3.3.2 Blok Pengendali Lampu Pemanas



Gambar 3.3 Digram Blok *Plant* dan Sistem Kendali Lampu Pemanas

Gambar 3.3 adalah kontrol lampu pemanas pada sistem yang dideteksi sensor LM35. *Setpoint* dari *user* berupa suhu. Jika *error* pada *setting point* banyak atau sedikit maka *output* sistem pada lampu pemanas akan aktif mengatur redup dan terangnya cahaya untuk menyesuaikan *input* kebutuhan suhu pada ruangan.

### 3.4 Desain Kandang *Brooding* Ayam

*Design* dari pembuatan kandang anak ayam atau disebut juga kandang *brooding* (kandang pemanas) setiap dinding dari kandang dilapisi menggunakan triplek, bertujuan suhu *temperature* dari luar tidak langsung masuk ke dalam kandang, tirai menggunakan plastik bertujuan sebagai penutup kandang agar udara pada malam hari tidak langsung masuk kedalam kandang dari atap kandang menggunakan kawat ram untuk kandang *brooding* menggunakan pemanas berupa bohlam kekuningan atau lampu pijar, bukan lampu yang berwarna putih terdapat pula tempat pakan dan minum. Untuk tempat pakan menggunakan *baby chick feeder* ukuran kandang *brooding*.

- a. Ukuran kandang panjang 100 cm, lebar 53 cm dan tinggi 60 cm.
- b. Ukuran pintu tinggi 14 cm dan panjang 16 cm.



Gambar 3.4 Desain Kandang *Brooding*



Gambar 3.5 Tampak dari atas Kandang *Brooding*

Desain pada kandang *brooding* seperti Gambar 3.4 dan Gambar 3.5 masih dalam kandang tradisional. Dengan alat-alat komponen seperti lampu pemanas dan fan masih belum terpasang. Desain akan jadi sempurna apabila alat-alat komponen sudah terpasang dengan rapi dan berjalan sesuai yang di harapkan. Jika alat-alat komponen sudah terpasang maka akan jadi desain seperti Gambar 3.6 dan Gambar 3.7.



Gambar 3.6 Tampilan Sistem Kontrol Suhu Kandang *Brooding* Ayam





Gambar 3.7 Tampilan atas Kandang *Brooding* Ayam

### 3.5 Perancangan Perangkat Keras

Setelah desain penulis melakukan perancangan alat keras yang saling terhubung dengan Arduino dan terintegrasi. Pertama perancangan alat pada Arduino ke LCD menggunakan media *I2C* untuk menghemat *pin* yang ada pada arduino, kemudian perancangan *keypad* sebagai *input* suhu, setelah itu sensor LM35 serta perancangan *dimmer* disambung pada Motor DC sebagai penggerak ke aktuator lampu pemanas, motor *driver* dan *relay* sebagai penggerak putaran kecepatan *fan*, *RTC module* sebagai penanda waktu semua di disambung pada Arduino Uno.

#### 3.5.1 Perancangan *Liquid Crystal Display* (LCD)

Tampilan untuk monitoring suhu pada Gambar 3.11 serta *input* yang digunakan untuk sistem ini adalah *liquid* LCD dengan ukuran 16x4 dengan komunikasi *I2C*. Pada LCD ada dua *output* yaitu *pin* SDA dan *pin* SCL adalah *pin* *output* dari jenis komunikasi *I2C*. Pada sistem ini LCD digunakan sebagai tampilan



| Pin   | Symbol          | External connection  | Function  |
|-------|-----------------|----------------------|---|
| 1     | V <sub>SS</sub> | Power supply         | Signal ground for LCM (GND)   |
| 2     | V <sub>DD</sub> |                      | Power supply for logic (+5V) for LCM  |
| 3     | V <sub>0</sub>  |                      | Contrast adjust   |
| 4     | RS              | MPU                  | Register select signal  |
| 5     | R/W             | MPU                  | Read/write select signal  |
| 6     | E               | MPU                  | Operation (data read/write) enable signal   |
| 7-10  | DB0-DB3         | MPU                  | Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation. |
| 11-14 | DB4-DB7         | MPU                  | Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU   |
| 15    | LED+            | LED BKL power supply | Power supply for BKL "A" (5V)   |
| 16    | LED-            |                      | Power supply for BKL "K" (GND)  |





Gambar 3.9 Tampilan LCD 16x4

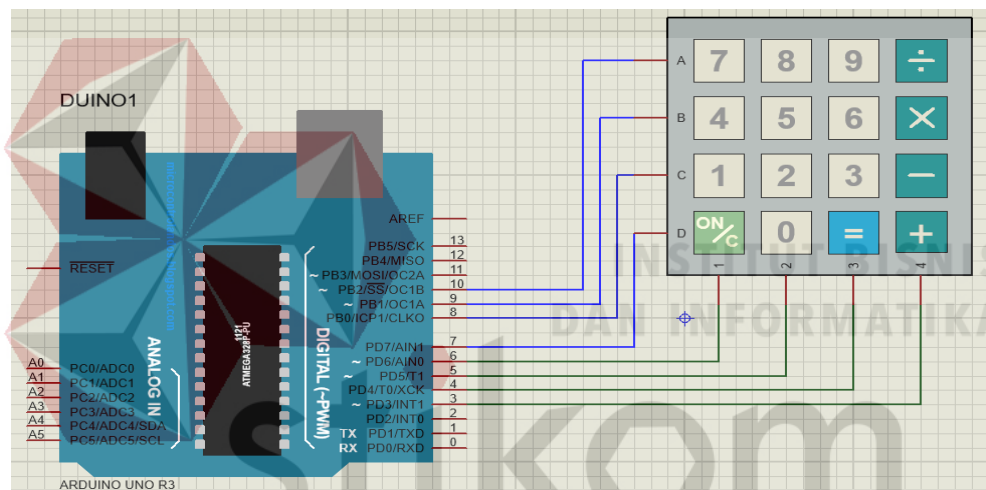
### 3.5.2 Perancangan Keypad

*Keypad* yang digunakan adalah *keypad* 4x4 yang mempunyai 8 *pin* dengan rangkaian kabel dengan 1-4 termasuk bagian dari baris sebelah kiri dan 5-8 termasuk pada bagian kolom *keypad* , 8 *pin* akan diproses di Arduino pada *pin digital* seperti pada Gambar 3.10 dibawah :



Gambar 3.10 Keypad 4x4

*Keypad* pada Gambar 3.10 digunakan untuk *input* suhu pada ruangan kandang *brooding* ayam *Input keypad* akan ditampilkan di LCD . *Input* angka di masukan berjumlah dua buah huruf bilangan yang dimasukan pada *setpoint*. *Input Pin* dari 1 sampai 4 dan 5 sampai 8 yang *real* yang ditempatkan di kotak elektro akan diperlihatkan secara ilustrasi melalui gambar *schematic* pada *input digital* pada Arduino dari *pin* 3, 4, 5, 6, 7, 8, 9, 10 menggunakan *proteus* untuk *schematic* pada Gambar 3.11.



Gambar 3.11 Keypad 4x4 terhubung ke Arduino



Gambar 3.12 Test Keypad 4x4 terhubung ke LCD

Pada Gambar 3.12 menunjukan pada layar LCD nilai masukan *inputan* pada *setpoint* sebagai acuan nilai *input* yang di butuhkan pada suhu di ruang kandang *brooding*.

### 3.5.3 Perancangan Kipas

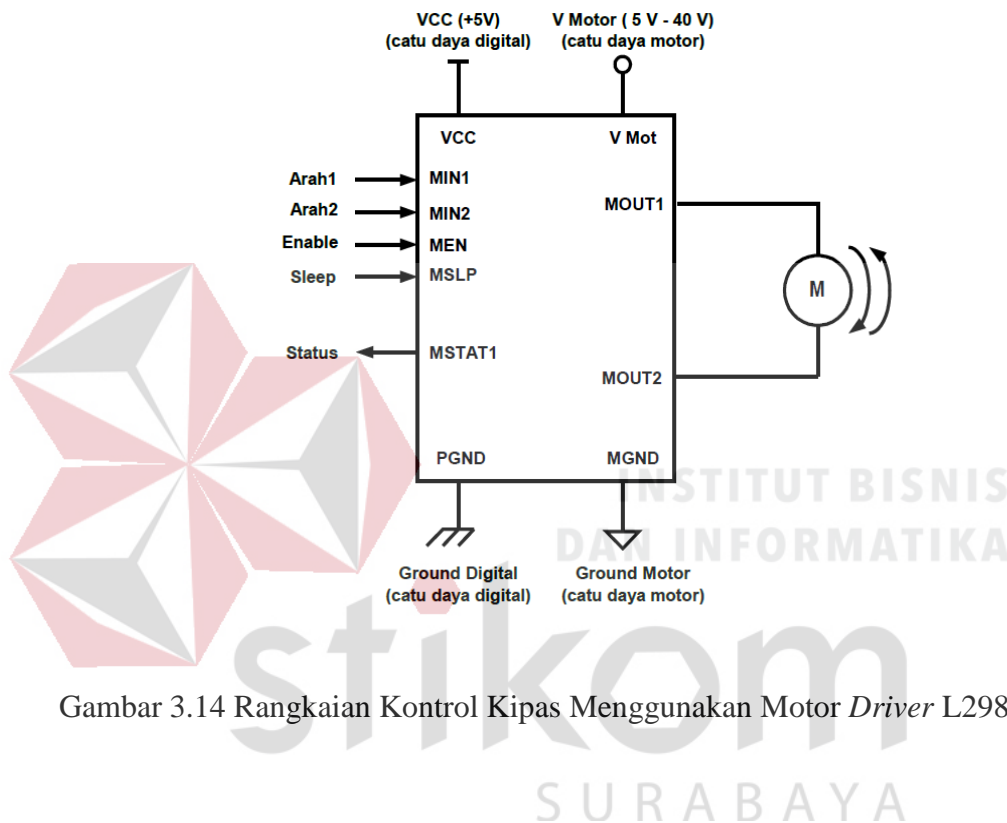
Aktuator kipas pada penelitian ini berpengaruh penting dalam membantu sirkulasi udara yang memnghembuskan udara panas keluar dan menghembuskan kedalam ruangan untuk mengganti udara yang panas agar udara lebih sejuk, kipas yang digunakan merupakan kipas DC 12 V, dalam kandang *brooding* terdapat 2 buah kipas yang terletak pada sisi kanan dan kiri pada kandang, semua kipas di hadapkan pada sisi yang sama bertujuan agar sirkulasi udara panas bisa di turunkan.



Gambar 3.13 *Fan Control*

Kontrol kipas pada penelitian ini tidak diatur kecepatannya karena dengan kecepatan maksimal kipas baru bisa membantu untuk sirkulasi udara dan penurunan suhu. Motor *driver* yang digunakan ssebagai pemicu kipas adalah motor *driver* L298N. Sistem kerja motor ini dimana ada *inputan* suhu dari LM35 dan *keypad*

untuk menuju pad suhu kipas akan berputar terus menjaga sirkulasi udara yang masuk dan keluar di kandang *brooding* dan ayam DOC didalam kandang pun dapat terkontrol suhunya, nilai kecepatan maksimal yang digunakan pada aktuator kipas. Rangkaian kontrol kipas pada *motor driver* LM L298N, lihat pada Gambar 3.14.

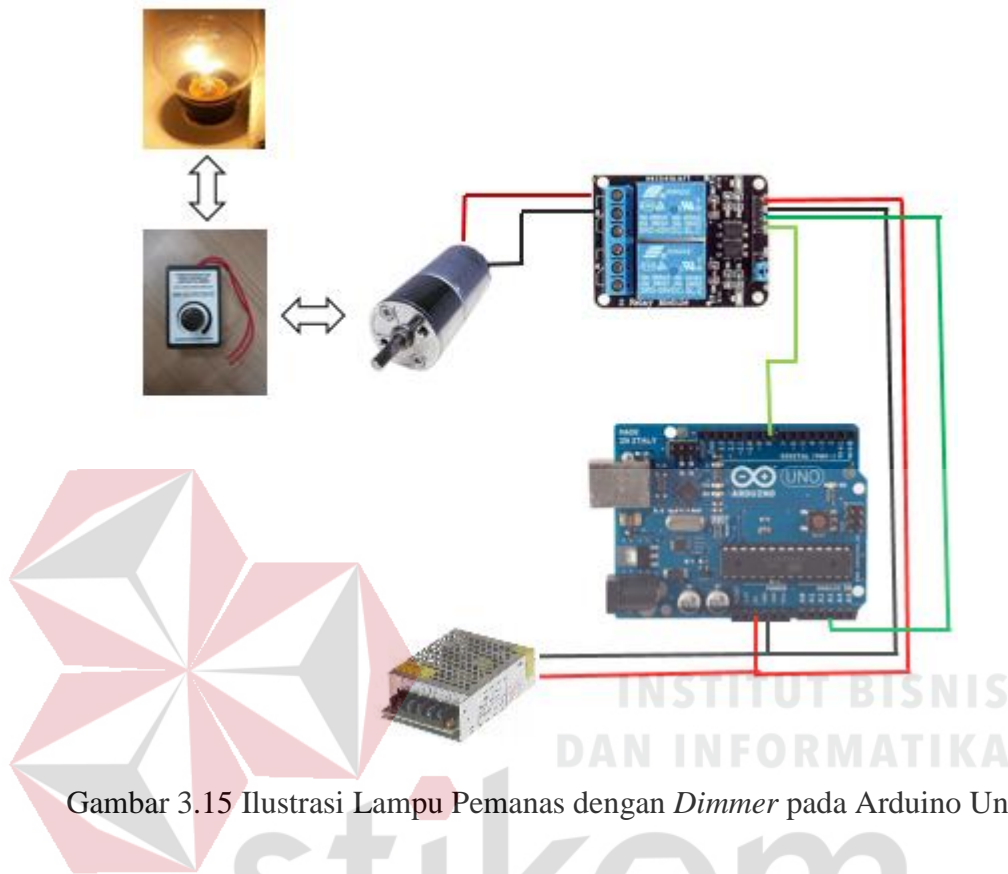


Gambar 3.14 Rangkaian Kontrol Kipas Menggunakan Motor Driver L298N

### 3.5.4 Perancangan Lampu Pemanas

Perancangan lampu pemanas di rancang melalui relay kemudian di kontrol melalui motor DC sebagai acuan penggerak *dimmer analog*. Ketika suhu di deteksi oleh sensor LM35 dan tidak sesuai dengan *setting point* dari input keypad ketika ada perbedaan *error* yang begitu besar maka arduino akan mengirim sinyal kepada relay untuk mengaktifkan coil pada relay sehingga motor DC 12V yang disambungkan pada putaran *dimmer* akan bergerak menyesuaikan kebutuhan suhu pada data tersebut diuji dan diketahui melalui pengujian data suhu pemanas secara

manual dari sehingga putaran *dimmer* dapat dikontrol secara otomatis menyesuaikan *setting point* yang diinputkan berikut rangkain Lampu Pemanas.

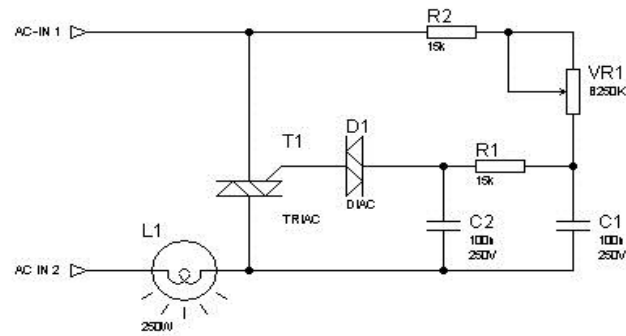
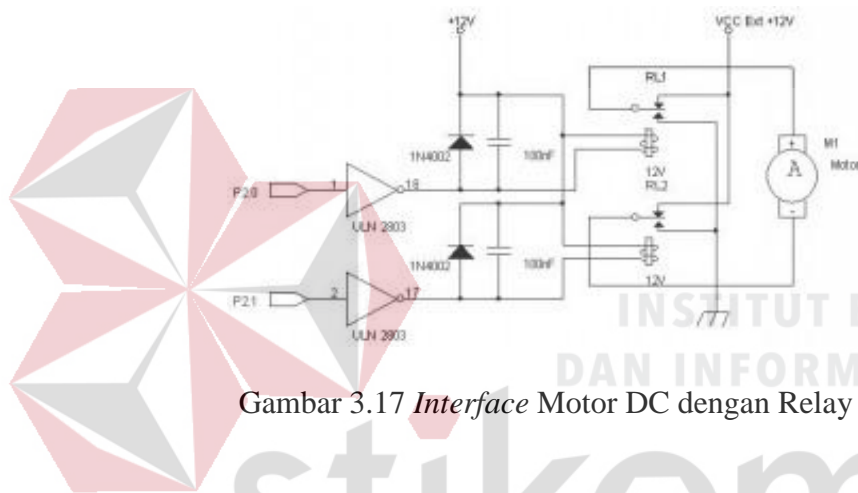


Gambar 3.15 Ilustrasi Lampu Pemanas dengan *Dimmer* pada Arduino Uno

### 3.5.5 Perancangan *Dimmer* dengan Motor DC

Pada perancangan *Dimmer* akan membahas tentang alur sistem kerja dan ulasan mengenai *dimmer*. *Dimmer* (peredup lampu) adalah suatu alat yang digunakan untuk mengatur cahaya bola lampu pijar dari padam, redup, terang, hingga sangat terang. Rangkaian ini dapat dipasang bola lampu pijar hingga daya 100Watt. Selain itu dengan menggunakan Potensiometer, kekuatan cahaya bias disesuaikan sesuai keinginan kita dengan cara memutar kekanan dan kekiri, dimana potensio ini dihubungkan dengan rangkaian yang terdiri dari beberapa komponen pendukung lainnya, contohnya resistor, kapasitor, IC 555, TRIAC, DIODA dan lain-lain.



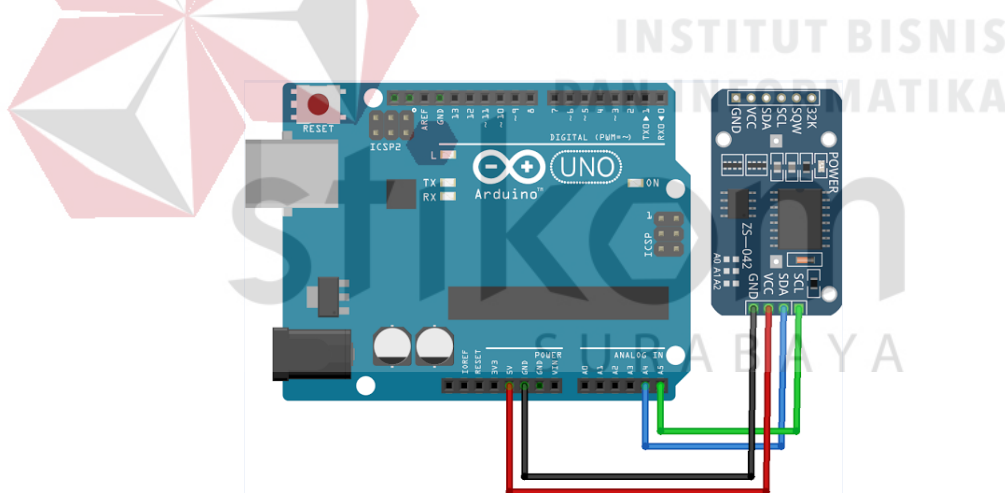
Gambar 3.16 Rangkaian *Dimmer*Gambar 3.17 *Interface* Motor DC dengan Relay

Rangkaian lampu *dimmer* pada Gambar 3.16 dibangun dengan TRIAC tipe Q 4004 LT yang dikontrol menggunakan DIAC dan potensiometer 250K $\Omega$ . Intensitas nyala lampu pada rangkaian lampu *dimmer* dengan TRIAC ini dikontrol dengan cara mengatur arus yang diberikan ke bola lampu melalui TRIAC. Secara teknis pengontrolan intensitas cahaya lampu dilakukan dengan mengatur tuas potensiometer. Arus *output* pada lampu dikendalikan oleh tegangan *gate* TRIAC melalui DIAC dari *output* pembagi tegangan potensiometer. kemudian kutub pada Motor DC di sambungkan ke potensiometer sebagai cara memutar *dimmer*, motor listrik memerlukan *supply* tegangan arus pada relay berfungsi

sebagai saklar untuk menggerakkan Motor DC pada dua arah seperti pada Gambar 3.17.

### 3.5.6 Perancangan RTC (*Real Time Clock*)

Modul RTC yang digunakan adalah menggunakan IC DS 1307 dimana *port* A4 adalah SDA dan *port* A5 adalah SCL. Baterai pada RTC berfungsi sebagai backup sumber daya untuk *module*, sehingga ketika modul RTC tidak mendapatkan sumber daya dari luar, maka IC DS1307 masih tetap dapat menghitung waktu. Dan sumber tegangan disambungkan port 5V dari arduino ke VCC dari RTC sedangkan *port ground* di sambungkan ke *ground* RTC. Pada perancangan RTC ini berfungsi sebagai penanda waktu ketika ada terjadi perubahan suhu.

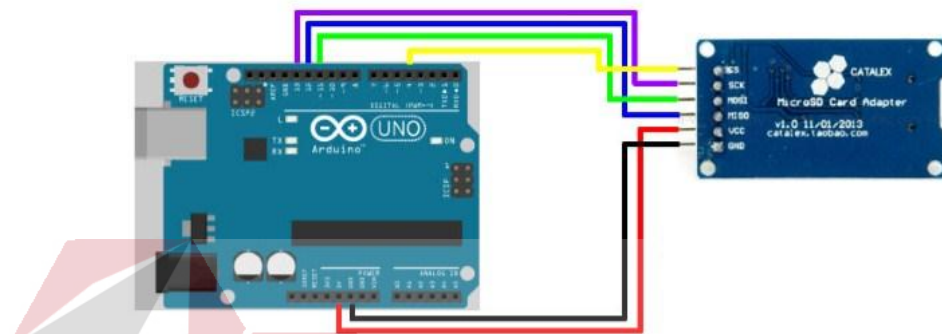


Gambar 3.18 Ilustrasi RTC *Module* pada Arduino Uno

### 3.5.7 Perancangan Micro SD *Module*

Micro SD *Module* merupakan solusi sederhana untuk tulis dan baca Micro SD menggunakan Arduino. *Pin Out* dari module Micro SD bisa langsung di

sambungkan ke Arduino. Salah satu contoh *project* yg sering memakai *module* ini adalah *Data Logger*. *Micro SD* atau *SD Card* komunikasi dengan Arduino menggunakan SPI, untuk *library*nya pakai "SD.h", secara *default library* micro SD sudah di sediakan oleh Arduino secara *default*.



Gambar 3.19 Ilustrasi Micro SD Module pada Arduino Uno

### 3.6 List Input dan Output

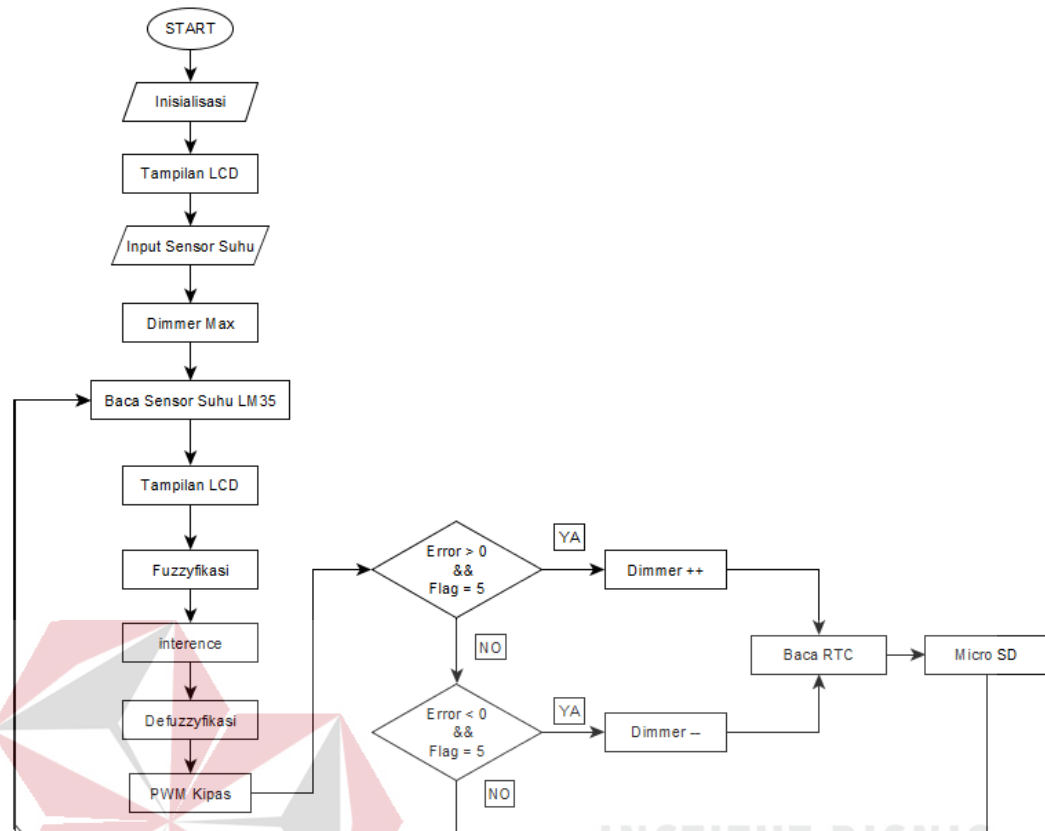
Tabel 3.2 List Input dan Output

| No | PIN    | Fungsi                    |
|----|--------|---------------------------|
| 1  | PIN 8  | Motor/Dimmer (Relay +)    |
| 2  | PIN A3 | Motor/Dimmer(Relay -)     |
| 3  | PIN 0  | Motor Driver L298N (IN A) |
| 4  | PIN 1  | Motor Driver L298N (IN B) |
| 5  | PIN 9  | Motor Driver L298N (EN A) |
| 6  | PIN A0 | Sensor LM35 (Data)        |
| 7  | PIN 12 | SD Card Module (MISO)     |
| 8  | PIN 13 | SD Card Module ( SCK)     |

| No | PIN    | Fungsi                |
|----|--------|-----------------------|
| 9  | PIN 11 | SD Card Module (CS)   |
| 10 | PIN 4  | SD Card Module (MOSI) |
| 11 | PIN A4 | RTC Module (SDA)      |
| 12 | PIN A5 | RTC Module (SCL)      |
| 13 | PD 2   | Keypad (Pin 1)        |
| 14 | PD 3   | Keypad (Pin 2)        |
| 15 | PD 4   | Keypad (Pin 3)        |
| 16 | PD 5   | Keypad (Pin 4)        |
| 17 | PD 6   | Keypad (Pin 5)        |
| 18 | PD 7   | Keypad (Pin 6)        |
| 19 | PD 8   | Keypad (Pin 7)        |
| 20 | PD 9   | Keypad (Pin 8)        |

### 3.7 Rancangan Alur Perangkat Lunak

Perancangan alur jalannya program dengan metode yang di implementasikan pada program untuk mendeteksi suhu dan untuk mencapai perubahan suhu sesuai dengan *setting point* yang ditentukan dan ketentuan aktuator yang aktif sesuai dengan rancangan serta lama aktif aktuator *fan chasing* dan lampu pemanas aktif dengan sistem yang maksimal PWM dan putaran potensiometer pada *dimmer* ketika mencapai *setting point* semua itu diatur oleh perangkat lunak yang akan di program. Rancangan perangkat lunak ini akan di tampilkan melalui *flowchart* pada Gambar 3.20.



Gambar 3.20 *Flowchart* Sistem Kontrol Suhu Kandang *Brooding* Ayam

### 3.8 Rancangan Desain Sistem *Fuzzy Logic*

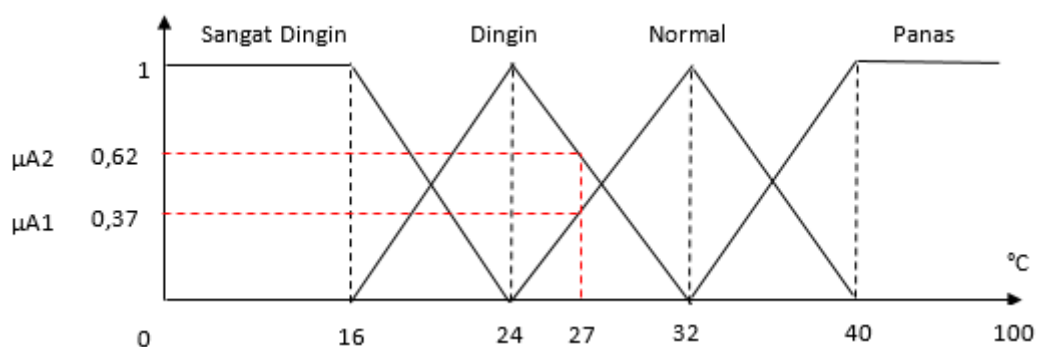
Pada rancangan ini penulis menggunakan sistem metode *fuzzy logic* atau sering disebut juga *Fuzzy Logic Controller* (FLC). merupakan sistem *fuzzy* yang diaplikasikan secara khusus dalam bidang kendali. *Input* dan *ouput* *fuzzy* akan diimplementasikan pada sistem kontrol suhu kandang *brooding* ayam sebagai berikut :

### 3.9 Fuzzyfikasi

Fuzzyfikasi merupakan tahap pertama dari proses inferensi *fuzzy*. Pada tahap ini data masukan diterima dan menentukan proses nilai fungsi keanggotaannya serta untuk mengubah suatu masukan dari bentuk tegas (*crisp*) menjadi *fuzzy* (variabel linguistik) (Jang *et al.* 1997). Fuzzifikasi memperoleh suatu nilai dan mengkombinasikannya dengan fungsi keanggotaan untuk menghasilkan nilai *fuzzy* (Sibigtroth 1992). yang biasanya disajikan dalam bentuk himpunan-himpunan *fuzzy* dengan suatu fungsi keanggotaannya masing-masing.

#### 3.9.1 Membership Function Suhu

Fungsi keanggotaan merupakan sebuah kurva yang menggambarkan pemetaan dari masukan ke derajat keanggotaan rentang antara 0 dan 1. Melalui fungsi keanggotaan yang telah di susun maka nilai-nilai masukan tersebut menjadi informasi *fuzzy* yang berguna nantinya untuk proses pengolahan secara *fuzzy*. Banyaknya jumlah fungsi keanggotaan dalam *fuzzy set* menentukan banyaknya aturan yang dibuat. Seperti pada Gambar 3.21 dan Gambar 3.24.



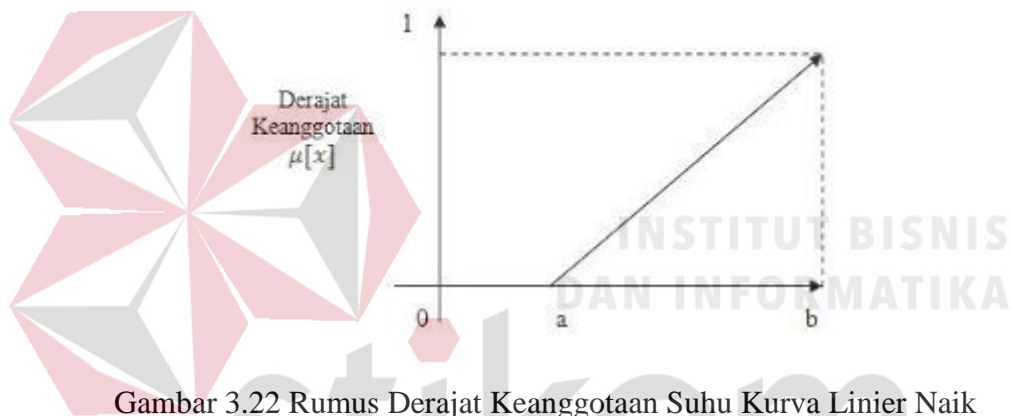
Gambar 3.21 Membership Function Suhu



Pada proses fuzzyfikasi fungsi keanggotaan suhu memiliki 4 nilai linguistik yaitu sangat dingin (SD), dingin (D), normal (N) dan panas (P) keluaran berupa suhu dan perubahan temperatur. Bentuk *membership function* yang digunakan yaitu kurva linier, trapesium dan segitiga.

Contoh perhitungan :

Untuk menentukan cepat dan lambatnya pergerakan *motor driver* dan putaran *dimmer* terlebih dahulu melakukan proses fuzzyfikasi. Nilai 27 didapat dari keluaran suhu yang berasal dari sensor LM35.

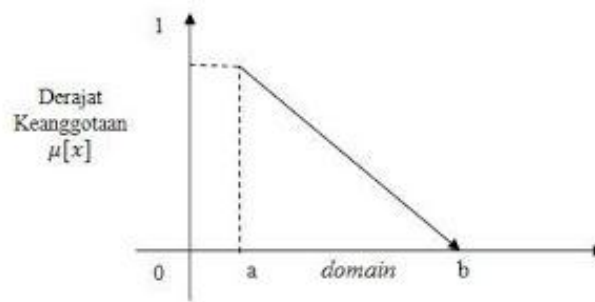


Gambar 3.22 Rumus Derajat Keanggotaan Suhu Kurva Linier Naik

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad \dots\dots\dots (3.1)$$

$$\mu_{A1} = \frac{x-a}{b-a} = \frac{27-24}{32-24} = \frac{3}{8} = 0,37 \quad \dots\dots\dots (3.2)$$



Gambar 3.23 Rumus Derajat Keanggotaan Suhu Kurva Linier Turun

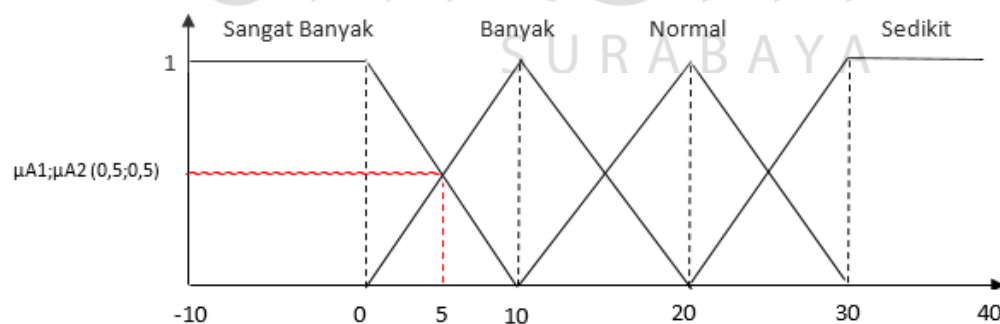
Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} \frac{b-x}{b-a}; & a \leq x \leq b \\ 0; & x \geq b \end{cases} \dots\dots\dots (3.3)$$

$$\mu_{A2} = \frac{b-x}{b-a} = \frac{32-27}{32-24} = \frac{5}{8} = 0,62 \dots\dots\dots (3.4)$$

Error = set point – pembacaan sensor

### 3.9.2 Membership Function Error



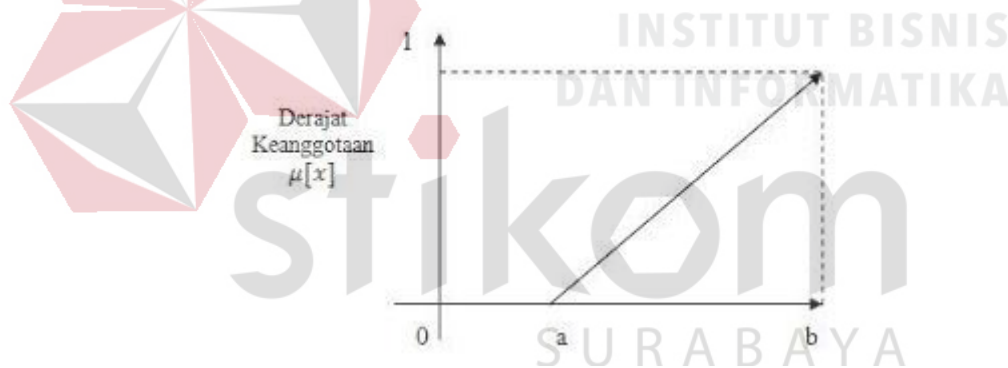
Gambar 3.24 Membership Function Error

Pada proses fuzzyfikasi fungsi keanggotaan memiliki 4 nilai linguistik yang dirancang yaitu Sangat Banyak (SB), Banyak (B), Normal (N) dan Sedikit (SD). error sedikit dan banyak ditentukan oleh hasil perolehan apabila hasil menunjukan

arah negatif maka *error* akan banyak dan apabila hasil ke positif maka *error* akan sedikit. semakin nilai *error* suhu menjauhi dari *setpoint* dalam keadaan suhu dalam ruangan turun maka proses *fuzzy* pada *relay* ke motor DC sebagai penggerak *dimmer* lampu pemanas akan terang dan *fan* kipas kecepatan pada PWM akan pelan, begitu pula apabila *input error* nilai pada suhu ruangan tinggi menjauhi *setpoint* maka *fan* kipas kecepatan PWM akan cepat dan lampu pemanas akan redup.

Contoh Perhitungan :

Nilai 7 didapat dari hasil pengurangan nilai *set point* (32) dengan keluaran suhu yang berasal dari sensor LM35 (27) dengan menggunakan rumus dibawah ini.

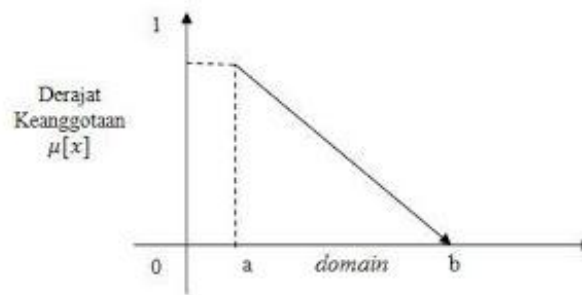


Gambar 3.25 Rumus Derajat Keanggotaan *Error* Kurva Linier Naik

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \\ \frac{x-a}{b-a}; & a \leq x \leq b \\ 1; & x \geq b \end{cases} \dots\dots\dots (3.5)$$

$$\mu_{A1} = \frac{x-a}{b-a} = \frac{5-0}{10-0} = \frac{5}{10} = 0,5 \dots\dots\dots (3.6)$$



Gambar 3.26 Rumus Derajat Keanggotaan *Error* Kurva Linier Turun

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} \frac{b-x}{b-a}; & a \leq x \leq b \\ 0; & x \geq b \end{cases} \dots\dots\dots (3.7)$$

$$\mu_{A2} = \frac{b-x}{b-a} = \frac{10-5}{10-0} = \frac{5}{10} = 0,5 \dots\dots\dots (3.8)$$

### 3.10 Inference (*Rule Set*)

*Rule set* adalah bentuk dari procedural knowledge. *Rules* menghubungkan informasi yang diberikan dengan beberapa tindakan. Tindakan ini dapat berupa pernyataan tegas dari informasi baru atau beberapa prosedur untuk dilakukan. Dalam hal ini, *rule* menggambarkan bagaimana cara memecahkan permasalahan. Struktur *rule* secara logika menghubungkan satu atau lebih *antecedents* (disebut juga *premis*) dalam bagian *IF*, dan satu atau lebih *consequents* (disebut juga kesimpulan) dalam bagian *THEN*. (Durkin, 1994).

Didalam *rule set* terdapat dua aturan masukan kombinasi (0 dan 1), kombinasi ini didapatkan dari himpunan *fuzzy* (keanggotaan suhu dan *error*), keanggotaan suhu terdiri dari 4 himpunan *fuzzy* (sangat dingin, dingin, normal dan sangat dingin). Sedangkan untuk keanggotaan *error* terdapat 4 himpunan *fuzzy* (

sangat banyak, banyak, normal, sedikit). keanggotaan suhu pada *rule set* di gunakan mendapatkan titik *setpoint* dari pembacaan sensor, sedangkan keanggotaan error digunakan mendapatkan nilai perpotongan dari pembacaan sensor dan nilai *setpoint* yang kemudian nilai perpotongan keduanya akan digunakan untuk proses defuzzyfikasi. Himpunan yang dihasilkan dari keanggotaan suhu dan *error* adalah 16 aturan yang mana 16 ini bertujuan untuk menghasilkan *output* berupa kecepatan putaran motor dan *dimmer*. Aturan *rule set* yang didapatkan dapat dilihat pada Tabel 3.3

Tabel 3.3 Tabel Aturan *Fuzzy*

| <b>Suhu</b><br><b>Error</b> | <b>SD</b> | <b>D</b> | <b>N</b> | <b>P</b> |
|-----------------------------|-----------|----------|----------|----------|
| <b>SB</b>                   | SP, NST   | SP, NST  | N, NT    | SC, NSR  |
| <b>B</b>                    | P, NST    | P, NT    | P, NN    | C, NSR   |
| <b>N</b>                    | P, NST    | SP, NST  | P, NT    | N, NN    |
| <b>S</b>                    | SP, NST   | N, NN    | N, NN    | P, NT    |

Tabel 3.4 Tabel Keterangan Aturan *Fuzzy*

| <b>Suhu</b>        | <b>Error</b>       | <b>Output Motor dan Kecepatan PWM</b> |
|--------------------|--------------------|---------------------------------------|
| SD = Sangat Dingin | SB = Sangat Banyak | SC = Sangat Cepat = 255               |
| D = Dingin         | B = Banyak         | C = Cepat = 192                       |
| N = Normal         | N = Normal         | N = Normal = 128                      |
| P = Panas          | S = Sedikit        | P = Pelan = 92                        |

|  |  |                               |
|--|--|-------------------------------|
|  |  | SP = Sangat Pelan = 64        |
|  |  | <b>Nilai Dimmer</b>           |
|  |  | NST = Nyala Sangat Terang = 5 |
|  |  | NT = Nyala Terang = 4         |
|  |  | NN = Nyala Normal = 3         |
|  |  | NR = Nyala Redup = 2          |
|  |  | NSR = Nyala Sangat Redup = 1  |

Dari Tabel 3.3 diatas dapat di artikan sebagai berikut :

*IF* sangat dingin *AND* sangat banyak *THEN* sangat pelan *AND* nyala sangat terang

*IF* sangat dingin *AND* banyak *THEN* pelan *AND* nyala sangat terang

*IF* sangat dingin *AND* normal *THEN* pelan *AND* nyala sangat terang

*IF* sangat dingin *AND* sedikit *THEN* sangat pelan *AND* nyala sangat terang

*IF* dingin *AND* sangat banyak *THEN* sangat pelan *AND* nyala sangat terang

*IF* dingin *AND* banyak *THEN* normal *AND* nyala normal

*IF* dingin *AND* normal *THEN* sangat pelan *AND* nyala sangat terang

*IF* dingin *AND* sedikit *THEN* normal *AND* nyala normal

*IF* normal *AND* sangat banyak *THEN* normal *AND* nyala terang

*IF* normal *AND* banyak *THEN* pelan *AND* nyala terang

*IF* normal *AND* normal *THEN* pelan *AND* nyala terang

*IF* normal *AND* sedikit *THEN* normal *AND* nyala normal

*IF* panas *AND* sangat banyak *THEN* sangat cepat *AND* nyala sangat redup

*IF* panas *AND* banyak *THEN* cepat *AND* nyala sangat redup

*IF* panas *AND* normal *THEN* normal *AND* nyala normal

*IF* panas *AND* sedikit *THEN* pelan *AND* nyala terang



Contoh perhitungan:

Tabel 3.5 Penerapan Nilai Fuzzyfikasi Terhadap *Rule Set*

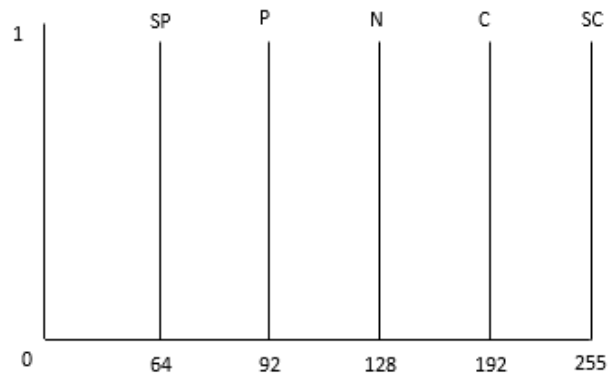
| <b>Suhu</b><br><b>Error</b> | <b>SD</b> | <b>D</b> | <b>N</b> | <b>P</b> |
|-----------------------------|-----------|----------|----------|----------|
| <b>SB</b>                   | 0         | 0,62     | 0,5      | 0        |
| <b>B</b>                    | 0         | 0,5      | 0,37     | 0        |
| <b>N</b>                    | 0         | 0        | 0        | 0        |
| <b>S</b>                    | 0         | 0        | 0        | 0        |

Dari penjelasan *rule set* Tabel 3.5 didapat dari hasil yang didapat dari perhitungan fuzzyfikasi mendapatkan nilai 0,5 yang merupakan titik potong dari hasil pengurangan *setpoint* dengan pembacaan sensor sedangkan 0,37 dan 0,62 merupakan titik potong pertama dan kedua dari hasil fuzzyfikasi dingin. *Rule set* mengambil data dari hasil himpunan fuzzyfikasi untuk mendapatkan nilai tegas sebagai penggerak motor pada *fan* dan penggerak *dimmer* pada lampu pemanas. Kemudian dilanjutkan ke proses defuzzyfikasi.

### 3.11 Defuzzyfikasi

Defuzzifikasi adalah suatu proses yang menggabungkan seluruh *fuzzy output* menjadi sebuah hasil spesifik yang dapat digunakan untuk masing-masing sistem *output* (Jang *et al.*1997). Defuzzyfikasi merupakan proses kebalikan dari fuzzyfikasi, merupakan proses kebalikan dari fuzzyfikasi, dimana nilai keanggotaan dari suatu gugus *fuzzy* dikonversi ke dalam suatu bilangan *real*.

Perancangan kandang *brooding* ini pada proses defuzzyfikasi menggunakan metode *weight-average*. Dan keluaran ini hasil defuzzyfikasi merupakan nilai *real* yang didapatkan dari hasil penjumlahan fuzzyfikasi di bagi hasil dari *rule set*.



Gambar 3.27 Fungsi Keanggotaan *Output* Defuzzyfikasi Motor *Fan*

Keterangan :

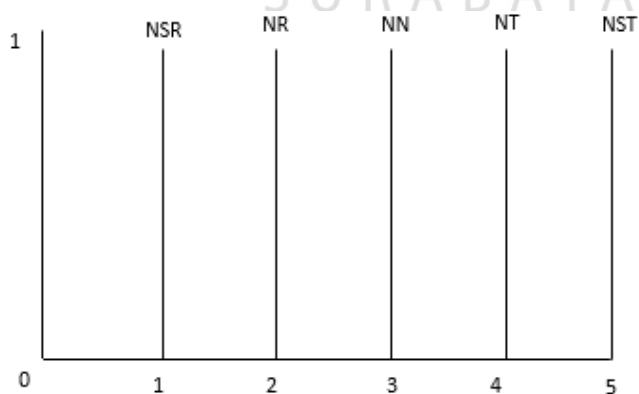
SP : Sangat Pelan

P : Pelan

N : Normal

C : Cepat

SC : Sangat Cepat



Gambar 3.28 Fungsi Keanggotaan *Output* Defuzzyfikasi *Dimmer* Lampu

Keterangan:

NSR : Nyala Sangat Redup

NR : Nyala Redup

NN : Nyala Normal

NT : Nyala Terang

NST : Nyala Sangat Terang

Nilai  $z$  yang dihasilkan defuzzyfikasi berfungsi menentukan suatu nilai *crisp output*. Suatu nilai *fuzzy output* yang berasal dari evaluasi *rule-set* diambil kemudian dimasukkan kedalam suatu fungsi keanggotaan keluaran (*membership function output*). Nilai-nilai tersebut dimasukkan ke dalam suatu rumus untuk mendapatkan hasil akhir yang disebut *crisp output*.

Realisasi proses pengambilan keputusan metode sugeno menjadi bentuk *crisp output* dapat menggunakan rumus sebagai berikut :

$$Crisp\_out = \frac{\sum_i FuzzyOutput_i \times (PosisiSingletonXaxis_i)}{\sum_i FuzzyOutput_i} \dots\dots\dots (3.9)$$

Contoh Perhitungan *output* motor *fan* : pada himpunan fuzzyfikasi sudah didapatkan nilai himpunan yaitu 0,62;0,37;0,5;0,5 yang dikalikan dengan ketentuan *rule-set*.

$$Crisp\_output = \frac{0,62 \times 64 + 0,37 \times 92 + 0,5 \times 128 + 0,5 \times 92}{0,62 + 0,37 + 0,5 + 0,5}$$

$$Crisp\_output = \frac{39,68+34,04+64+46}{1,99}$$

$$Crips\_ouput = \frac{183,72}{1,99} = 92 \dots\dots\dots (3.10)$$

Dengan rumus defuzzyfikasi diatas:

Nilai *crisp output* ini merupakan nilai akhir dari proses *ouput* defuzzyfikasi dimana nilai tersebut digunakan pergerakan motor pada kipas. Dari hasil nilai didapatkan 92 pada putaran motor PWM “Pelan”.

$$Crisp\_output = \frac{0,62 \times 5 + 0,37 \times 4 + 0,5 \times 3 + 0,5 \times 4}{0,62 + 0,37 + 0,5 + 0,5}$$

$$Crisp\_output = \frac{3,1 + 1,48 + 1,5 + 2}{1,99}$$

$$Crips\_ouput = \frac{8,08}{1,99} = 4 \dots\dots\dots (3.11)$$

Dengan rumus defuzzyfikasi diatas:

Nilai *crisp output* ini merupakan nilai akhir dari proses *ouput* defuzzyfikasi dimana nilai tersebut digunakan pergerakan *dimmer* sebagai pengontrol lampu pemanas. Dari hasil nilai didapatkan 4 pada putaran *dimmer* “Nyala Terang”.

