

## BAB I PENDAHULUAN

### 1.1. Latar Belakang

Kenyamanan di dalam sebuah ruangan menjadi hal penting dalam menunjang aktivitas manusia. Dari berbagai hal yang ada di dalam ruangan, penerangan dan suhu merupakan parameter penting yang dalam menciptakan kenyamanan di dalam sebuah ruangan. Penggunaan teknologi seperti lampu penerangan dan pendingin ruangan sudah banyak digunakan dalam sebuah ruangan. Pada umumnya penggunaan teknologi ini masih dikendalikan secara manual. Hal ini dapat menyebabkan beberapa masalah seperti lampu dan pendingin ruangan yang dibiarkan menyala walaupun ruangan dalam keadaan kosong dan tingkat penerangan serta suhu di dalam ruangan sudah memenuhi tingkat kenyamanan. Untuk mengatasi masalah ini diperlukan sistem yang dapat mengontrol tingkat penerangan dan suhu secara otomatis. Sistem juga harus dapat mengetahui keberadaan manusia di dalam ruangan, sehingga manfaatnya dirasakan langsung oleh manusia dan tidak ada pemakaian energi yang sia-sia.

Dalam mengendalikan sistem penerangan dan pengendali suhu secara otomatis, dibutuhkan suatu kontroler dengan metode kontrol yang dapat memfungsikan sistem dengan baik. Oleh karena itu perlu dirancang sebuah kontroler yang dapat mengendalikan sistem tersebut secara otomatis dan memberikan performansi yang baik. Karena sistem mempunyai beberapa masukan dan keluaran maka digunakan metode kontrol logika fuzzy yang memiliki keunggulan dalam mengendalikan beberapa sistem secara bersamaan.

### 1.2. Rumusan Masalah

Berdasarkan masalah yang telah diuraikan pada latar belakang, maka pembahasan skripsi ini ditekankan pada:

- 1) Merancang dan membuat suatu sistem yang dapat mengenali keberadaan manusia di dalam ruangan.
- 2) Merancang dan membuat kontroler logika fuzzy untuk mengatur sistem penerangan dan suhu serta mengimplentasikannya pada sebuah miniatur ruangan.

### 1.3. Batasan Masalah

Pada skripsi ini dibatasi oleh hal-hal sebagai berikut:

- 1) Ruangannya berupa miniatur terbuat dari akrilik berukuran 40cm x 25cm x 30cm ( $p \times l \times t$ ).
- 2) Pengujian keberadaan manusia didalam miniatur diganti dengan tikus yang mempunyai suhu tubuh mendekati suhu manusia.
- 3) Pengaturan sistem berupa pengaturan suhu dengan menggunakan peltier dan kipas, dan pengaturan tingkat kuat penerangan (iluminasi) dengan menggunakan lampu LED.
- 4) *Set point* iluminasi nyaman adalah 200 lux, *set point* suhu nyaman adalah 25<sup>0</sup>C (Frick, 2008).
- 5) Pembahasan mekanik serta model matematis tidak dijabarkan, pembahasan ditekankan pada penggunaan kontrol logika fuzzy pada sistem.
- 6) Sensor keberadaan manusia menggunakan PIR (*passive infra red*), sensor intensitas penerangan menggunakan LDR (*light dependent resistor*) serta sensor suhu menggunakan sensor SHT11.

### 1.4. Tujuan

Tujuan dari penelitian ini adalah merancang sistem penerangan dan pengendali suhu ruangan secara otomatis berdasarkan keberadaan manusia dengan menggunakan kontrol logika fuzzy.

### 1.5. Sistematika Penulisan

Skripsi ini terdiri dari enam bagian dengan sistematika sebagai berikut:

#### **BAB I**

#### **Pendahuluan**

Membahas latar belakang, rumusan masalah, batasan masalah, tujuan dan sistematika pembahasan.

#### **BAB II**

#### **Tinjauan Pustaka**

Membahas teori-teori yang mendukung dalam perancangan dan pembuatan alat.

#### **BAB III**

#### **Metodologi Penulisan**

Membahas metode penelitian dan perencanaan alat.

**BAB IV Perancangan dan Pembuatan Alat**

Membahas perancangan kontroler menggunakan kontrol logika fuzzy dan menerapkannya ke dalam *software* di komputer, yang selanjutnya di tulis ke mikrokontroler sehingga sistem dapat bekerja dengan baik.

**BAB V Pengujian dan Analisis**

Membahas hasil pengujian sistem yang sudah dibuat, serta analisis hasil yang diperoleh.

**Bab VI Kesimpulan dan Saran**

Membahas kesimpulan perancangan ini dan saran-saran yang diperlukan untuk pengembangan selanjutnya.



## BAB II

### TINJAUAN PUSTAKA

Untuk memudahkan dalam memahami cara kerja rangkaian maupun dasar-dasar perencanaan dari sistem yang dibuat, maka perlu adanya penjelasan dan uraian mengenai teori penunjang yang digunakan dalam penulisan skripsi ini.

#### 2.1. Suhu dan Kenyamanan

Suhu suatu sistem adalah sifat yang menentukan apakah sistem itu setimbang termal dengan sistem lainnya atau tidak. Apabila dua sistem atau lebih berada dalam kesetimbangan termal, sistem sistem itu dikatakan mempunyai suhu yang sama (Sears, 1994 :357).

Ciri umum dari iklim Indonesia yang tropis lembab adalah temperatur udara yang relatif panas, intensitas radiasi matahari yang tinggi, dan kelembaban udara yang tinggi. Sedikit lebih spesifik mengenai iklim tropis lembab dapat dinyatakan sebagai berikut ;

- Temperatur udara maksimum rata rata adalah  $27^{\circ} - 32^{\circ}\text{C}$ , dan temperatur minimum rata rata adalah  $20^{\circ} - 23^{\circ}\text{C}$ ,
- Kelembaban udara rata rata adalah 75% - 82 % ,
- Curah hujan selama satu tahun antara 1000mm – 5000mm,
- Kondisi langit pada umumnya berawan dengan jumlah awan antara 60% - 90% ,
- Radiasi matahari global harian rata rata bulanan adalah sekitar  $400 \text{ W/m}^2$ ,
- Kecepatan angin rata rata adalah rendah, sekitar 2 – 4 m/detik.

(Soegijanto, 1998 : 8)

Kenyamanan termal lebih bersifat individual. Keadaan lingkungan tertentu bisa dirasakan berbeda oleh individu yang berbeda. Faktor faktor kenyamanan termal yang dapat mempengaruhi seseorang dibedakan menjadi dua, yaitu faktor dominan dari alam dan faktor pilihan manusia. Faktor faktor alam yang pasti mempengaruhi kenyamanan termal bagi manusia yaitu suhu udara, kelembaban udara, dan pergerakan udara.

Biasanya kelembaban udara menjadi penting saat suhu udara mendekati atau melampaui ambang batas atas daerah kenyamanan termal dan kelembaban

udara mencapai lebih dari 70% atau kurang dari 30%. Kelembaban udara yang tinggi mengakibatkan sulit terjadinya penguapan di permukaan kulit sehingga mekanisme pelepasan panas bisa terganggu. Bila kelembaban udara rendah, orang menderita efek keringnya udara seperti selaput lendir mengering, batuk rejan, radang mata dan sebagainya (Frick, 2004 :46-48).

Pengaruh kelembaban atas kenyamanan ruang tergantung pada keringat seseorang. Perasaan atas suhu dalam ruang dipengaruhi juga oleh kelembaban. Makin tinggi kelembaban, makin rendah suhu maksimal yang masih dirasakan nyaman, seperti ditunjukkan oleh Tabel 2.1.

**Table 2.1.** Pengaruh Kelembaban Terhadap Kenyamanan

Kelembaban	Suhu Nyaman
0 – 30%	22 – 30 <sup>0</sup> C
30 – 50%	22 – 29 <sup>0</sup> C
50 – 70 %	22 – 28 <sup>0</sup> C
70 – 100%	22 – 27 <sup>0</sup> C

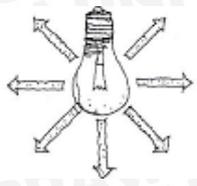
Sumber : Frick, 2004 : 75

Menurut penelitian yang dilakukan oleh Mom & Wiesebron sekitar tahun 1936-1940 di Bandung, tingkat kenyamanan termal untuk masyarakat asli Indonesia dikelompokkan menjadi beberapa yaitu *dingin tidak nyaman* dengan suhu dibawah 20<sup>0</sup>C, *sejuk nyaman* dengan suhu minimum 20<sup>0</sup>C dan kelembaban 50%, *nyaman* atau *optimal nyaman* dengan suhu 24<sup>0</sup> – 28<sup>0</sup>C dan kelembaban 70% , *hangat nyaman* dengan suhu maksimum 31<sup>0</sup>C dan kelembaban 60%, dan *panas tidak nyaman* dengan suhu diatas 31<sup>0</sup>C. (Soegijanto, 1998 :242).

## 2.2. Penerangan

Dalam menentukan sistem penerangan suatu ruangan perlu mempertimbangkan faktor penerangan terkait dengan ukuran dasar teknik penerangan. Ada empat macam ukuran dasar teknik penerangan, yaitu arus atau aliran cahaya ( $\phi$ ), kepadatan cahaya (I), tingkat kuat penerangan atau iluminasi (E), dan luminasi (L). Tabel 2.2 menunjukkan macam-macam ukuran dasar yang dipakai dalam teknik penerangan.

Tabel 2.2. Ukuran Dasar Teknik Penerangan

Gambar	Satuan	Simbol /Rumus	Definisi
	Lumen (lm)	$\phi$	Jumlah cahaya total yang dipancarkan oleh sebuah sumber cahaya 1 detik.
	Candela (cd)	$I = \phi / W$	Arus cahaya yang dipancarkan ke satu arah tertentu per satuan ruangnya.
	Lux (lx) = lumen/m <sup>2</sup>	$E = \phi / A$	Arus cahaya yang jatuh pada permukaan sebuah bidang per meter persegi.
	cd/m <sup>2</sup>	$L = I/A$	Kepadatan cahaya per meter persegi dari satu bidang permukaan yang dapat terlihat oleh mata

Sumber : Frick, 2004 : 26

Kriteria pokok dalam menerapkan iluminasi ialah;

- 1) Supaya tugas visual dapat terlaksana secara baik, yaitu cepat dan tepat (*visual performance*),
- 2) Tercapai kenyamanan dan suasana santai bagi mata (*visual comfort and pleasantness*),
- 3) Memperhitungkan faktor ekonomi,
- 4) Memperhitungkan faktor sosial.

Untuk memperoleh iluminasi yang baik yaitu yang memenuhi fungsi agar mata kita dapat melihat dengan jelas dan nyaman ada 5 kriteria yang harus dipenuhi, yaitu ;

- 1) Kuantitas atau jumlah cahaya pada permukaan tertentu (*lighting level*) atau tingkat kuat penerangan,
- 2) Distribusi kepadatan cahaya (*luminance distribution*),

- 3) Pembatasan agar cahaya tidak menyilaukan mata (*limitation of glare*),
- 4) Arah pencahayaan dan pembentuk bayangannya (*light directionality and shadows*),
- 5) Warna cahaya dan refleksi warnanya (*light color and colour rendering*).

Untuk mencapai kenyamanan visual suatu ruangan perlu diketahui taraf iluminasi yang dianjurkan, sebagaimana ditunjukkan oleh Tabel 2.3.

**Tabel 2.3.** Taraf Iluminasi Untuk Kenyamanan Visual

Uraian	Kuat Penerangan yang Dianjurkan (lux)
Minimum untuk koridor rumah tinggal, daerah pengamanan, visual pabrik, <i>night club</i> , lampu hias restoran.	20
Tempat tangga rumah tinggal.	30
Koridor perkantoran	50
Minimum penerangan umum rumah sakit	75
Penerangan dapur, kamar mandi rumah. Penerangan umum pada industri, pertokoan, restoran, lobi, lounge hotel, gedung pertemuan, dan tempat ibadah.	100
Ruang sidang perkantoran, ruang kelas, ruang pekerjaan biasa di pabrik, dapur restoran, pabrik, dan dapur hotel.	200
Ruang kerja perkantoran, ruang sekolah dengan ketepatan visual tinggi, ruang pusat perhatian gedung, ruang pemeriksaan di rumah sakit, penerangan laboratorium, dan ruang olahraga.	300
Ruang gambar perkantoran, area papan tulis sekolah, area pekerjaan halus / QC di pabrik, etalase toko, perumahan, pentas dan auditorium.	500
Daerah untuk pemeriksaan warna di pabrik.	750
Daerah untuk pekerjaan amat halus, tepat dan teliti di pabrik. Etalasi di pertokoan yang ramai	1000
Ruang operasi rumah sakit	10,000

Sumber : Frick, 2004 : 30

### 2.3. Sistem Kontrol

Dalam mempelajari teknik kontrol terdapat beberapa istilah yang dibutuhkan untuk menjelaskan sistem kontrol, yaitu *plant*, *process*, *system*, *disturbance*, dan *feedback control*.

*Plant* didefinisikan sebagai perangkat peralatan, mungkin hanya terdiri dari beberapa bagian mesin yang bekerja bersama sama yang digunakan untuk melakukan suatu operasi tertentu. Proses (*process*) menurut kamus Merriam-Webster, didefinisikan sebagai operasi atau perkembangan alamiah yang berlangsung secara kontinu yang diandai oleh suatu deretan perubahan kecil yang berurutan dengan cara relatif tetap dan menuju ke suatu hasil atau keadaan akhir tertentu. Jadi setiap operasi yang dikontrol disebut proses.

Sistem (*system*) didefinisikan sebagai kombinasi dari beberapa komponen yang bekerja bersama-sama dan melakukan suatu sasaran tertentu. Gangguan (*disturbance*) didefinisikan sebagai suatu sinyal yang cenderung mempunyai pengaruh yang merugikan pada harga keluaran sistem. Jika suatu gangguan dibangkitkan dalam sistem, disebut internal, sedangkan gangguan eksternal dibangkitkan di luar sistem dan merupakan suatu masukan. *Feedback Control* atau kontrol umpan balik mengacu pada suatu operasi, yang dengan gangguan, cenderung mengurangi perbedaan antara keluaran dari sistem dan suatu acuan masukan dan bahwa hal itu dilakukannya berdasarkan pada perbedaan ini. Disini hanya gangguan yang tidak diperkirakan atau gangguan yang dapat diperkirakan atau gangguan yang diketahui dapat selalu dikompensasi di dalam sistem tersebut.

Sistem kontrol umpan balik disebut juga sebagai sistem loop tertutup. Pada sistem kontrol loop tertutup, sinyal kesalahan yang bekerja, yaitu perbedaan antara sinyal masukan dan sinyal umpan balik (yang mungkin sinyal keluarannya sendiri atau fungsi dari sinyal keluaran dan turunannya), disajikan ke kontroler sedemikian rupa untuk mengurangi kesalahan dan membawa keluaran sistem ke nilai yang dikehendaki. Istilah kontrol loop tertutup selalu berarti penggunaan aksi kontrol umpan balik untuk mengurangi kesalahan sistem.

Istilah sistem kontrol loop terbuka dapat didefinisikan sebagai suatu sistem yang keluarannya tidak mempunyai pengaruh terhadap aksi kontrol disebut sistem kontrol loop terbuka. Dengan kata lain, sistem kontrol loop

terbuka keluarannya tidak dapat digunakan sebagai perbandingan umpan balik dengan masukan. Suatu contoh sederhana adalah mesin cuci. Perendaman, pencucian, dan pembilasan dalam mesin cuci dilakukan atas basis waktu. Mesin ini tidak mengukur sinyal keluaran yaitu tingkat kebersihan pakaian (Ogata, 2010 : 3-10).

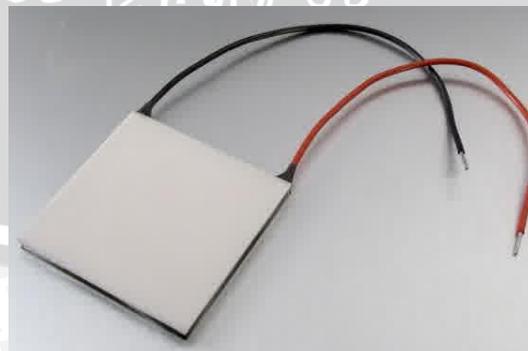
Pada skripsi ini digunakan sistem kontrol loop tertutup dengan *plant* berupa ruangan yang dikontrol iluminasi dan suhu.

### 2.3.1. Aktuator

Aktuator adalah suatu komponen, alat, atau peralatan (berupa mekanis, pneumatik, hidrolis, elektronik atau gabungan darinya) yang mampu mengolah data perintah (sinyal kontrol) menjadi sinyal aksi ke suatu *plant* (Ogata, 2010 : 22). Skripsi ini menggunakan peltier, kipas dc, dan lampu LED sebagai aktuator.

#### 2.3.1.1. Peltier

Sebuah peltier tersusun dari dua lempeng keramik dengan semikonduktor tipe-p dan tipe-n didalamnya. Ketika terjadi perbedaan suhu antara sisi satu dengan sisi lainnya, termoelektrik dapat menghasilkan tegangan DC. Fenomena ini disebut efek *seebeck*. Sebaliknya jika termoelektrik dialiri arus DC, maka akan dihasilkan panas pada satu sisi, dan dingin pada sisi yang lain. Fenomena ini disebut efek *peltier*. Bentuk dari peltier ditunjukkan pada Gambar 2.1.



**Gambar 2.1.** Bentuk Peltier

Sumber : Perancangan

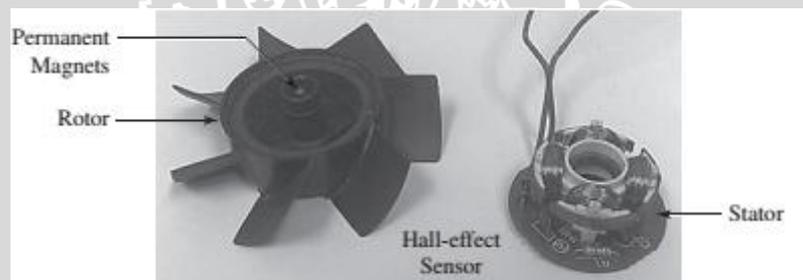
Sebuah peltier biasanya akan menghasilkan perbedaan suhu maksimal 70°C antara sisi panas dan dinginnya. Apabila peltier semakin panas maka akan semakin kurang efisiensinya. Jumlah panas yang dihasilkan sebanding dengan arus dan waktu. Untuk menghasilkan proses pendinginan pada peltier maka panas yang dihasilkan harus dibuang atau dikurangi, biasanya dengan menempelkan besi *heatsink* dan kipas di sisi panas peltier.

### 2.3.1.2. Kipas DC *Brushless*

Motor DC *brushless* (BLDC) dinamai demikian karena tidak memiliki sikat (*brush*). Karena tidak adanya sikat, motor BLDC menghasilkan listrik dan kebisingan (*noise*) yang rendah, serta perawatan yang mudah karena tidak menggunakan sikat yang membutuhkan pergantian dalam jangka waktu tertentu. Pada motor BLDC, rotor terbuat dari magnet permanen dan stator tersusun dari beberapa lilitan (*coil*). Karena rotor lebih ringan daripada rotor pada motor *brush*, motor *brushless* dapat beroperasi dengan kecepatan lebih tinggi dibandingkan motor *brush*. Karena ketiadaan *brush*, motor BLDC lebih efisien daripada motor *brush* dan lebih baik dalam disipasi panas karena kumparan yang berada diluar bagian motor. Motor *brushless* umumnya digunakan dalam aplikasi permesinan yang membutuhkan respon yang cepat, panas yang rendah, dan masa aktif yang panjang. Oleh karena itu, motor *brushless* banyak digunakan pada mesin dengan performa tinggi, robot, dan *disk drive* komputer

Rotor adalah bagian pada motor yang berputar. Stator adalah bagian motor yang diam atau statis. Stator berfungsi sebagai medan putar motor yang memberikan gaya elektromagnetik pada rotor sehingga dapat berputar. Semua motor *brushless* terdiri dari rotor yang terbuat dari magnet permanen dan dapat disusun dari dua sampai delapan kutub magnet utara (N) atau selatan (S), serta stator yang terdiri dari beberapa lilitan (koil), yang pada umumnya berjumlah 12 lilitan (Jouaneh, 2013 : 269).

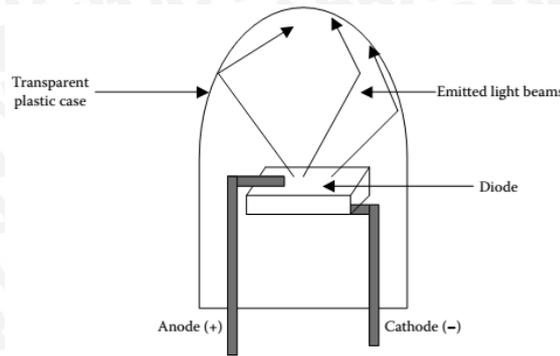
Kipas pendingin ,seperti yang digunakan pada komputer personal, merupakan motor DC *brushless* (BLDC), karena bebannya yang ringan dan menghasilkan listrik dan *noise* yang kecil. Kipas ini menggunakan motor BLDC dua fasa.Tata letak dari komponen kipas BLDC ditunjukkan oleh Gambar 2.2. Pada permukaan rotor terpasang magnet permanen, sedangkan pada stator terpasang kumparan dua fasa. Kipas ini menggunakan sebuah sensor *hall-effect* (sensor medan magnet) yang terpasang pada papan rangkaian stator. Ketika sumbu rotor sejajar dengan sensor, maka sensor akan mengirimkan dua siklus gelombang (50% *duty cycle*). Siklus ini diumpungkan menuju gerbang masukan transistor untuk mengontrol aliran arus pada dua kumparan. Saat dua gelombang ini lengkap, maka hanya satu kumparan yang akan aktif pada satu waktu. Kenaikan tegangan masukan pada motor akan meningkatkan kecepatan motor.



**Gambar 2.2.** Tata Letak Komponen Kipas DC *Brushless*  
Sumber : Jouaneh, 2013 : 271

### 2.3.1.3. Lampu LED

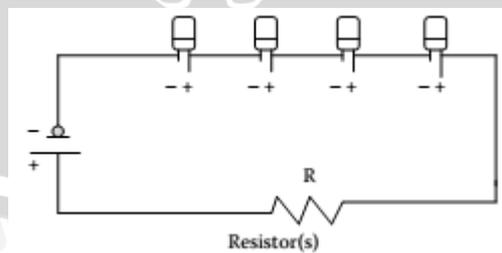
LED atau *light emitting diode* adalah semikonduktor *p-n junction* yang terdiri dari bagian positif (*p-side*) atau anoda, dan bagian negatif (*n-side*) atau katoda seperti ditunjukkan Gambar 2.3. Seperti halnya diode, LED akan aktif (menyala) jika ada arus yang mengalir dari anoda ke katoda, tapi tidak dengan sebaliknya.



**Gambar 2.3.** Bagian *Light Emitting Diode*  
 Sumber : Gilbert, 2009 : 9

Ketika LED dialiri arus, elektron akan melewati pembatas (*barrier*) dan menemui sebuah *hole*. Setelah itu elektron berubah menjadi energi yang rendah dan melepaskannya dalam bentuk foton. Foton inilah yang membawa radiasi elektromagnetik dari panjang gelombang. Cahaya dan warna yang dihasilkan dari panjang gelombang bergantung pada celah energi materi yang digunakan untuk membentuk *p-n junction*.

Pada sebuah rangkaian, LED dapat disusun baik secara seri maupun paralel seperti ditunjukkan Gambar 2.4 dan Gambar 2.5. Karena LED dirancang untuk bekerja pada tegangan antara 2 sampai 4 volt, maka kelebihan tegangan yang masuk mengakibatkan LED rusak terbakar. Untuk mencegah hal tersebut dalam sebuah rangkaian ditambahkan resistor dengan nilai hambatan tertentu. Resistor yang berfungsi membatasi arus masuk juga membatasi tegangan kerja pada LED.



**Gambar 2.4.** LED Pada Rangkaian Seri  
 Sumber : Gilbert, 2009, hlm.10

Jika beberapa LED dirangkai seri, maka arus yang mengalir pada masing masing LED adalah sama. Sedangkan pada rangkaian

paralel, nilai tegangan masing masing LED adalah sama. Untuk mencari nilai hambatan resistor pada sebuah rangkaian LED secara seri dapat menggunakan persamaan 2-1.

$$R = \frac{(V_s - N \times V_f)}{I} \dots\dots\dots (2-1)$$

dimana,

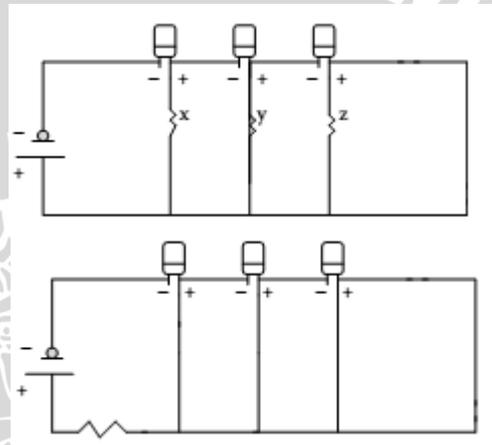
R = nilai hambatan ( $\Omega$ )

$V_s$  = tegangan sumber (V)

N = jumlah LED yang dirangkai seri

$V_f$  = tegangan kerja untuk masing masing LED (V)

I = Arus maksimum yang bekerja (A)



**Gambar 2.5.** LED Pada Rangkaian Paralel  
*Sumber : Gilbert, 2009 : 12*

Pada rangkaian LED secara paralel ada dua cara pemasangan resistor, yaitu satu resistor untuk semua LED, dan satu resistor untuk masing masing LED, seperti Gambar 2.5. Agar tidak ada kelebihan arus, pada pemasangan satu resistor bersama, LED yang terpasang harus memiliki  $V_f$  yang sama. Untuk nilai hambatan resistor bersama ini diperoleh dari nilai hambatan pengganti ( $R_T$ ) seperti ditunjukkan persamaan 2-2 dan 2-3.

$$\frac{1}{R_T} = \frac{1}{R_x} + \frac{1}{R_y} + \frac{1}{R_z} \dots\dots\dots (2-2)$$

dimana,



$$R_x = \frac{V_s - V_{fx}}{I} \dots \dots \dots (2-3)$$

(Gilbert,2009 : 16)

Kelebihan lampu LED dibandingkan lampu pijar (*incandescent*) dan lampu neon (*compact fluorescent light* atau CFL) terletak pada umur lampu, daya yang dikonsumsi, serta disipasi panas. Lampu LED memiliki umur lampu mencapai 60,000 jam, dibandingkan lampu pijar yang berumur 750 jam dan lampu CFL yang berumur 7,500 jam. Untuk menghasilkan cahaya 1050 lumen, dibutuhkan lampu pijar 75 watt, atau lampu CFL 20 watt. Sedangkan untuk cahaya yang sama, hanya dibutuhkan lampu LED 9 watt. Disipasi panas pada lampu LED tidak dipancarkan bersama cahaya sebagaimana lampu pijar dan CFL, melainkan terdapat pada bagian *chipset*.

### 2.3.2. Sensor

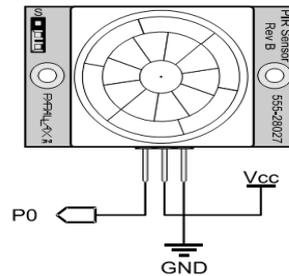
Sensor adalah suatu komponen yang mendeteksi keluaran atau informasi lainnya yang diperlukan dalam siste kontrol. Sedangkan tranduser adalah suatu komponen yang mampu merubah besaran-besaran non listrik (mekanis, kimia atau yang lainnya) menjadi besaran-besaran listrik atau sebaliknya (Ogata, 2010 : 22).

Sensor yang digunakan pada skripsi ini adalah PIR (*passive infrared receiver*) sebagai sensor keberadaan manusia, SHT11 sebagai sensor suhudaan LDR (*light dependent resistor*) sebagai sensor kuat cahaya.

#### 2.3.2.1. Passive Infrared Receiver (PIR)

PIR merupakan sebuah sensor berbasiskan infrared tapi tidak memancarkan apapun seperti IR LED. Sensor ini hanya merespon energi dari pancaran sinar inframerah pasif yang dimiliki oleh setiap benda yang terdeteksi. Sensor ini mampu menyaring panjang gelombang sinar inframerah pasif antara 8 sampai 14 mikrometer, sehingga panjang gelombang yang dihasilkan dari tubuh manusia yang berkisar antara 9 sampai 10 mikrometer dapat dideteksi oleh sensor. Untuk jarak jangkau dari sensor PIR sendiri bisa diatur sesuai kebutuhan, akan tetapi jarak maksimalnya hanya +/- 10 meter dan

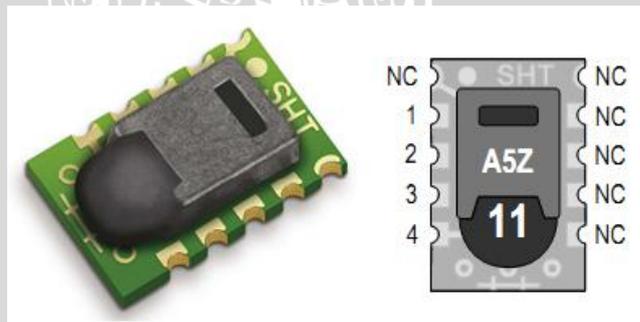
minimal +/- 30 cm. Skema rangkaian sensor PIR ditunjukkan oleh Gambar 2.6.



**Gambar 2.6.** Skema Rangkaian Sensor PIR  
Sumber : Parallax, 2012

### 2.3.2.2. SHT 11

Sensor SHT11 digunakan untuk mengukur suhu dan kelembaban relatif pada ruangan. Dapat mengukur kelembaban mulai dari 0 sampai 100% RH dengan akurasi  $\pm 3.0\%$  dan suhu mulai dari  $40^{\circ}\text{C}$  sampai  $123.8^{\circ}\text{C}$  dengan akurasi  $\pm 0.4^{\circ}\text{C}$ . Tegangan masukan sensor SHT11 yaitu 2.4V sampai 5.5V. Sensor SHT memiliki 4 pin utama seperti ditunjukkan pada Gambar 2.7. Pin 1 dihubungkan ke *ground*, pin 2 dihubungkan pada mikrokontroler sebagai pengirim dan penerima data, pin 3 dihubungkan pada mikrokontroler sebagai masukan *clock* (SCK) dan pin 4 dihubungkan pada sumber tegangan.

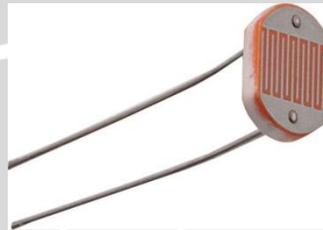


**Gambar 2.7.** Sensor SHT 11  
Sumber : Sensirion, 2001

### 2.3.2.3. Light Dependent Resistor (LDR)

Sensor LDR adalah salah satu jenis resistor yang dapat mengalami perubahan resistansinya apabila mengalami perubahan penerimaan cahaya. Besarnya nilai hambatan pada sensor bergantung pada besar kecilnya cahaya yang diterima. LDR terbuat dari *cadmium*

*sulfida* yaitu merupakan bahan semikonduktor yang resistansinya berubah-ubah menurut banyaknya cahaya (sinar) yang mengenainya. Resistansi LDR pada tempat yang gelap biasanya mencapai sekitar 10 M $\Omega$ , dan ditempat terang LDR mempunyai resistansi yang turun menjadi sekitar 150  $\Omega$ . Seperti halnya resistor konvensional, pemasangan LDR dalam suatu rangkaian sama persis seperti pemasangan resistor biasa. Bentuk dari sensor LDR ditunjukkan oleh Gambar 2.8.



**Gambar 2.8.** Sensor LDR  
Sumber : RS Component, 1997

### 2.3.3. Kontroler

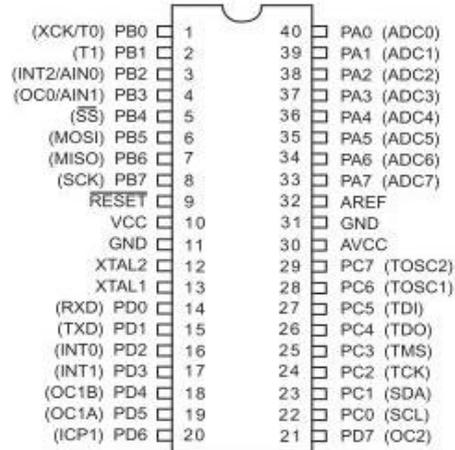
Variabel yang dikontrol adalah besaran atau keadaan yang diukur dan dikontrol. Variabel yang dimanipulasi adalah besaran atau keadaan yang diubah oleh kontroler untuk mempengaruhi nilai variabel yang dikontrol. Dalam keadaan normal, variabel yang dikontrol adalah keluaran dari sistem. Kontrol berarti mengukur nilai dari variabel sistem yang dikontrol dan menerapkan variabel yang dimanipulasi ke sistem untuk mengoreksi atau membatasi penyimpangan nilai yang diukur dari nilai yang dikehendaki (Ogata, 2010 : 3)

Skripsi ini menggunakan kontrol logika fuzzy (KLF) untuk mengontrol variabel kontrol berupa suhudaan iluminasi. Untuk menjalankan kontrol logika fuzzy digunakan komponen elektronik berupa mikrokontroler, IC L293, *optocoupler* dan *relay*.

#### 2.3.3.1. Mikrokontroler ATMEGA 16

Mikrokontroler ATmega16 memiliki kapasitas 16 kbyte EEPROM 512 byte, 1Kbyte SRAM, 32 pin I/O, CPU terdiri atas 32 *register* , memiliki 2 buah *timer/counter* 8 bit dan 1 buah *timer/counter* 16 bit, memiliki 4 channel PWM, 8 channel 10 bit ADC, dan memiliki

*watchdog timer* dengan *oscillator internal*. Masing-masing kaki dalam mikrokontroler ATMega16 mempunyai fungsi tersendiri. Mikrokontroler ATMega16 mempunyai 32 pin, susunan masing-masing pin ditunjukkan oleh Gambar 2.9.



**Gambar 2.9.** Konfigurasi Pin ATMEGA 16

Sumber : Atmel, 2007

Fungsi kaki-kaki ATMega16 adalah:

- Port A (Pin A0...7) merupakan saluran masukan/keluaran dua arah dan juga saluran masukan ADC (*analog to digital converter*).
- Port B (Pin B0...7), merupakan saluran masukan/keluaran dua arah dan juga mempunyai fungsi khusus, seperti dapat dilihat pada Tabel 2.3.

**Tabel 2.3.** Fungsi Khusus Port B

Port Pin	Fungsi
PB0	T0 ( <i>Timer/ counter 0 external counter input</i> ) XCK ( <i>USART External Clock Input/Output</i> )
PB1	T1 ( <i>Timer/ counter 1 external counter input</i> )
PB2	AIN0 ( <i>Analog comparator positive input</i> ) INT2 ( <i>External interrupt 2 input</i> )
PB3	AIN1 ( <i>Analog Comparator Negative Input</i> ) OC0 ( <i>Timer/Counter0 Output Compare Match Output</i> )
PB4	SS ( <i>SPI Slave Select input</i> )

PB5	MOSI ( <i>SPI bus Master Output/Slave Input</i> )
PB6	MISO ( <i>SPI bus Master Input/Slave Output</i> )
PB7	SCK ( <i>SPI bus Serial Clock</i> )

Sumber : Atmel, 2007

- Port C (Pin C0..6), merupakan saluran masukan/keluaran dua arah dan juga mempunyai fungsi khusus, seperti dapat dilihat pada Tabel 2.4.

**Tabel 2.4.** Fungsi Khusus Port C

Port Pin	Fungsi
PC0	SCL ( <i>2-wire serial bus clock line</i> )
PC1	SDA ( <i>2-wire serial bus data input/output line</i> )
PC2	TCK ( <i>JTAG Test clock</i> )
PC3	TMS ( <i>JTAG test mode select</i> )
PC4	TDO ( <i>JTAG test data out</i> )
PC5	TDI ( <i>JTAG test data in</i> )
PC6	TOSC1 ( <i>timer oscillator pin.1</i> )
PC7	TOSC2 ( <i>timer oscillator pin.2</i> )

Sumber : Atmel, 2007

- Port D (Pin D0..7), merupakan saluran masukan/keluaran dua arah dan juga mempunyai fungsi khusus, seperti dapat dilihat pada Tabel 2.5.

**Tabel 2.5.** Fungsi Khusus Port D

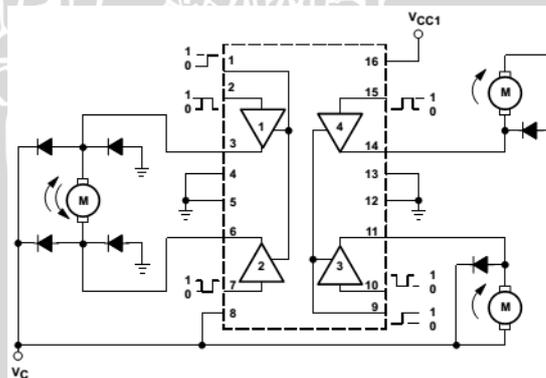
Port Pin	Fungsi
PC0	RXD ( <i>USART input pin</i> )
PC1	TXD ( <i>USART output pin</i> )
PC2	INT0 ( <i>external interrupt 0 input</i> )
PC3	INT1 ( <i>external interrupt 1 input</i> )
PC4	OC1B ( <i>timer/ counter1 output compare b match output</i> )
PC5	OC1A ( <i>timer/ counter2 output compare a match output</i> )
PC6	ICP ( <i>timer/counter1 input capture pin</i> )
PC7	OC2 ( <i>timer/counter2 output compare match output</i> )

Sumber : Atmel 2007

- Pin 10 VCC, merupakan saluran masukan untuk catu daya positif sebesar 5 volt DC.
- Pin 11 GND, merupakan *ground* dari seluruh rangkaian.
- Pin 12 XTAL2 dan pin 13 XTAL1 merupakan pin masukan *clock* eksternal
- Pin 30 AVCC, merupakan catu daya untuk perangkat ADC.
- Pin 31 GND, merupakan *ground* dari ADC.
- Pin 32 AREF, merupakan pin analog referensi untuk masukan ADC.

### 2.3.3.2. IC L293

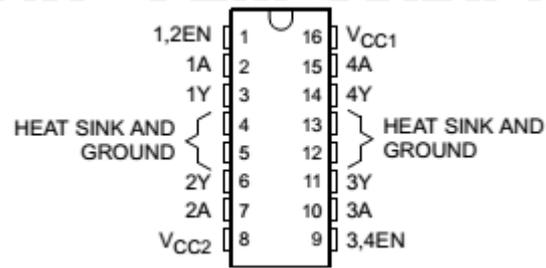
IC L293 adalah IC yang didalamnya terdapat 4 buah *half-H*, yaitu rangkaian yang digunakan untuk mengontrol beban dengan arus tinggi seperti motor. Arus maksimal yang masuk ke L293 mencapai 1 A pada tegangan 4.5 V sampai 36 V. Gambar 2.10 menunjukkan L293 digunakan untuk mengontrol motor. Pada bagian kiri, motor menggunakan 2 buah *half-H* sehingga dapat berputar searah maupun berlawanan arah jarum jam. Pada bagian kanan, motor menggunakan 1 *half-H* sehingga hanya dapat berputar satu arah.



**Gambar 2.10.** Contoh Penggunaan L293D Untuk Menggerakkan Motor

Sumber : Texas, 2002

Adapun fungsi pin pada L293D ditunjukkan oleh gambar 2.11 dan Tabel 2.6.



**Gambar 2.11.** Konfigurasi Pin L293D  
 Sumber : Texas, 2002

**Tabel 2.6.** Fungsi Kaki IC L293D

No.Kaki	Keterangan dan Fungsi
1	Enable untuk masukan 1 dan 2 (dapat digunakan PWM)
2	Masukan 1
3	Keluaran 1
4, 5	Ground
6	Keluaran 2
7	Masukan 2
8	Tegangan masukan (maksimal 7 V)
9	Enable untuk masukan 3 dan 4 (dapat digunakan PWM)
10	Masukan 3
11	Keluaran 3
12, 13	Ground
14	Keluaran 4
15	Masukan 4
16	Tegangan masukan untuk keluaran (maks. 36 V)

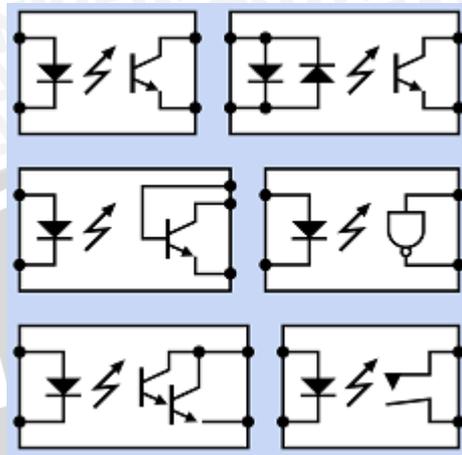
Sumber : Texas , 2002

### 2.3.3.3. Optocoupler

*Optocoupler* adalah komponen elektronika yang berfungsi sebagai penghubung berdasarkan cahaya optic. Sebuah optocoupler terdiri dari dua bagian utama, yaitu sebuah sumber cahaya optik (biasanya menggunakan LED) dan sebuah sensor cahaya (biasanya menggunakan *photo-transistor*), dimana keduanya dipasang dalam satu kemasan tanpa adanya hubungan konduktif rangkaian secara langsung.

Gambar 2.12 menunjukkan beberapa variasi komponen didalam *optocoupler*, yaitu *LED-phototransistor* (yang umum digunakan), 2

dioda, phototransistor dengan basis dihubungkan pada keluaran, *chip* logika NAND, *photo-darlington*, dan SSR (*solid state relay*).



**Gambar 2.12.** Variasi Komponen Dalam Optocoupler  
Sumber : Platt, 2015 : 35

Prinsip kerja optocoupler yaitu saat optocoupler dialiri arus, komponen LED akan memancarkan cahaya yang kemudian akan dideteksi oleh *photo-transistor* dan menyebabkan terjadinya hubungan (*switch on*) pada *photo-transistor*.

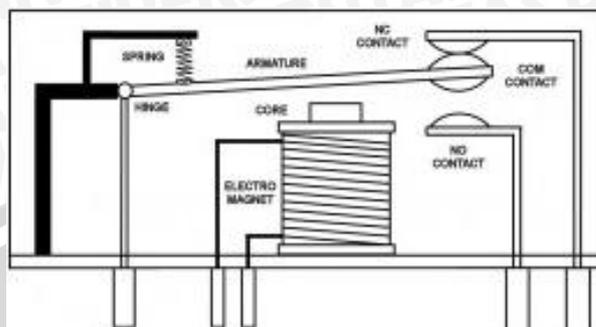
*Optocoupler* lebih sering digunakan sebagai komponen isolasi, yaitu memisahkan aliran listrik antara satu komponen dengan komponen yang lain. *Optocoupler* dapat melindungi komponen komponen sensitif, seperti mikrokontroler, dari lonjakan tegangan atau tegangan yang tidak sesuai dari bagian rangkaian yang lain (Platt, 2015 : 32-35).

#### 2.3.3.4. Relay

Sebuah *relay* memungkinkan sebuah sinyal atau pulsa listrik untuk menghidupkan atau mematikan (*switching*) aliran listrik secara terpisah. Relay pada umumnya menggunakan tegangan atau arus rendah untuk mengontrol tegangan atau arus yang lebih tinggi.

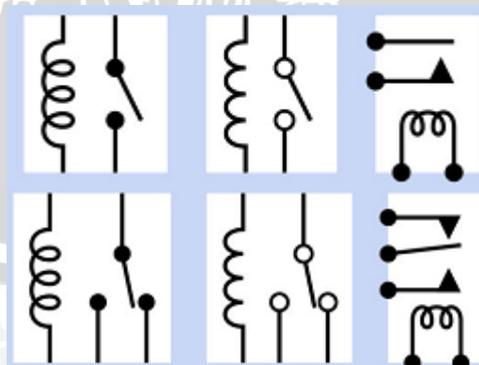
Sebuah *relay* memiliki sebuah kumparan, armatur (tuas logam yang dapat ditarik magnet) dan kontak, seperti ditunjukkan Gambar 2.13. Arus mengalir melalui kumparan yang berfungsi sebagai

elektromagnetik atau menghasilkan medan magnet. Medan magnet ini kemudian menarik armatur untuk menutup atau membuka kontak.



**Gambar 2.13.** Bagian Bagian Relay  
 Sumber : Platt, 2013 : 65

Konfigurasi relay menggunakan singkatan yang sama seperti saklar. SP, DP, 3P, atau 4P, menunjukkan jumlah *pole* atau bagian armatur yang menyentuh kontak (1,2,3,atau 4). ST dan DT menunjukkan pensaklaran *single-throw* (kondisi kontak terbuka atau *normally-open*) dan *double-throw* (kondisi kontak dapat terbuka atau tertutup atau *normally-closed*). Singkatan ini biasanya digabungkan, seperti 3PST atau SPDT. Singkatan lain yang juga digunakan adalah *form A* (menunjukkan kontak terbuka), *form B* (kontak tertutup), *form C* (*double-throw*). Penulisannya didahului dengan angka yang menunjukkan jumlah *pole*, seperti “2 form C” yang berarti relay DPDT. Dalam dunia elektronika, skema relay ditunjukkan oleh gambar 2.14.



**Gambar 2.14.** Simbol Skematik Relay SPST (atas) dan Relay SPDT (bawah)

Sumber : Platt, 2013 : 65

## 2.4. Logika Fuzzy

Logika fuzzy adalah metode untuk merumuskan kapasitas manusia dalam melakukan penalaran sesuatu yang tidak pasti (*imprecise reasoning*) atau penalaran perkiraan (*approximate reasoning*). Penalaran merupakan kemampuan manusia untuk berpikir dan memutuskan sesuatu ketidakpastian. Dalam logika fuzzy semua kebenaran adalah parsial atau perkiraan.

Ide dari logika fuzzy pertama kali ditemukan oleh Professor L.A.Zadeh pada 1965. Penemuan ini tidak terlalu terkenal sampai Dr.E.H.Mamdani menggunakan logika fuzzy untuk mengendalikan sebuah mesin uap otomatis. Dan sampai saat ini logika fuzzy sudah digunakan dalam banyak aspek kehidupan. Untuk menggunakan logika fuzzy dalam aplikasi nyata dibutuhkan tiga tahap, yaitu :

- Fuzzifikasi, yaitu mengubah data klasik atau data tegas (*crisp*) menjadi data fuzzy atau fungsi keanggotaan (*membership function* - MFs),
- Proses Inferensi Fuzzy, yaitu menggabungkan fungsi keanggotaan dengan aturan (*rule*) kontrol untuk memperoleh keluaran fuzzy,
- Defuzzifikasi, yaitu mengubah kembali data keluaran fuzzy menjadi data tegas (*crisp*).

(Yan, 1993 : 18)

### 2.4.1. Fuzzifikasi

Proses fuzzifikasi merupakan proses untuk mengubah variabel non fuzzy (variabel numerik) menjadi variabel fuzzy (variabel linguistik). Nilai masukan-masukan yang masih dalam bentuk variabel numerik yang telah dikuantisasi sebelum diolah oleh pengendali logika fuzzy harus diubah terlebih dahulu ke dalam variabel fuzzy. Melalui fungsi keanggotaan yang telah disusun, maka dari nilai-nilai masukan tersebut menjadi informasi fuzzy yang berguna nantinya untuk proses pengolahan secara fuzzy pula. Proses ini disebut fuzzifikasi. Proses fuzzifikasi diekspresikan sebagai berikut:

$$x = \text{fuzzifier}(x_0)$$

dimana,

$x_0$  = nilai *crisp* variabel masukan,

$x$  = himpunan fuzzy variabel yang terdefinisi,

*fuzzifier* = operator fuzzifikasi yang memetakan himpunan *crisp* ke himpunan fuzzy.

Pedoman memilih fungsi keanggotaan untuk proses fuzzifikasi yaitu ;

1. Himpunan fuzzy dengan distribusi simetris.
2. Gunakan himpunan fuzzy dengan jumlah ganjil, berkaitan erat dengan jumlah kaidah (*rules*).
3. Mengatur himpunan fuzzy agar saling menumpuk.
4. Menggunakan fungsi keanggotaan bentuk triangular atau trapezoidal untuk sistem yang membutuhkan variasi dinamik secara signifikan dalam periode waktu yang singkat. Menggunakan bentuk Gaussian atau sigmoidal untuk sistem yang membutuhkan akurasi kontrol yang tinggi.

(Yan, 1993 : 50)

#### 2.4.2. Basis Aturan dan Inferensi

Basis Aturan adalah bagian yang menggambarkan dinamika suatu sistem terhadap masukan yang dikarakteristikan oleh sekumpulan variabel-variabel linguistik dan berbasis pengetahuan seorang operator ahli. Pernyataan tersebut umumnya dinyatakan oleh suatu pernyataan bersyarat. Dalam pengendali berbasis fuzzy, aturan pengendalian fuzzy berbentuk aturan “ IF – THEN”. Untuk sebuah sistem MISO (*Multi Input Single Output*) basis aturan pengendalian fuzzy berbentuk seperti berikut ini.

Rule 1 IF  $X_1$  is  $A_{11}$  AND ... AND  $x_m$  is  $A_{1m}$  THEN  $Y$  is  $B_1$

Rule 2 IF  $X_1$  is  $A_{21}$  AND ... AND  $x_m$  is  $A_{2m}$  THEN  $Y$  is  $B_2$

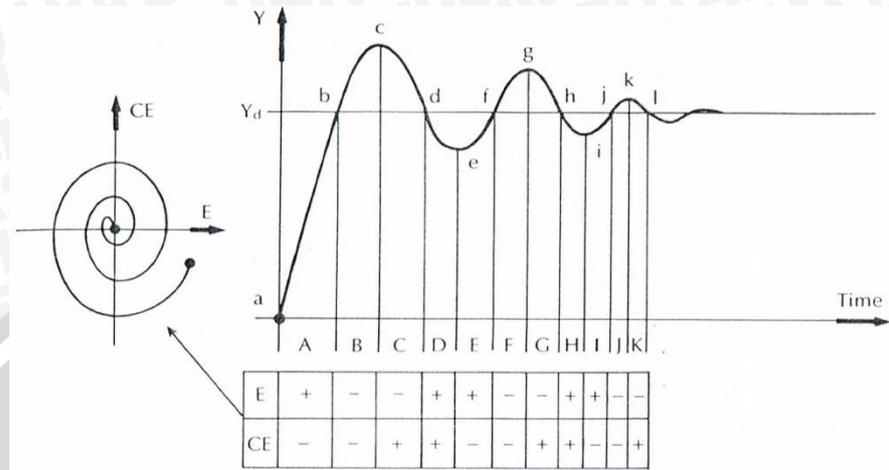
⋮  
⋮  
⋮

Rule n IF  $X_n$  is  $A_{n1}$  AND ... AND  $x_m$  is  $A_{nm}$  THEN  $Y$  is  $B_n$

Dengan  $X_j$  merupakan variabel masukan sistem,  $A_{ij}$  merupakan *fuzzy-set* untuk  $X_j$ ,  $Y$  merupakan variabel keluaran sistem,  $B_i$  merupakan *fuzzy-set* untuk  $Y$ , AND adalah operator fuzzy.

Basis Aturan disusun berdasarkan pengetahuan kualitatif sistem *loop* tertutup yang bertujuan untuk mengurangi lonjakan, waktu naik dan osilasi. Dalam menentukan *error* dan *delta error* (*change error*) dalam kondisi positif

atau negatif digunakan tanggapan undak (*step response*) sistem yang ditunjukkan oleh Gambar 2.15.



**Gambar 2.15.** Tanggapan Undak Sistem  
Sumber : Yan, 1993 : 59

Misalkan himpunan fuzzy untuk variabel masukan dan keluaran dinyatakan dengan istilah (N,Z,P) yang berarti negatif, zero, dan positif. Berdasarkan tanggapan undak sistem, maka aturan kendali fuzzy dapat diformulasikan dalam Tabel 2.7.

**Tabel 2.7.** Aturan Kendali Fuzzy dengan Variabel N, Z, P

No	E	CE	CI	Acuan	Fungsi
1	P	Z	P	titik a,e,i	mengurangi waktu naik
2	Z	N	N	titik b,f,i	mengurangi lonjakan
3	N	Z	N	titik c,g,k	mengurangi lonjakan
4	Z	P	P	titik d,h,l	mengurangi osilasi
5	Z	Z	Z	set point	mengerem sistem
6	P	N	P	rentang A,E	mengurangi waktu naik
7	N	N	N	rentang B,F,J	mengurangi lonjakan
8	N	P	N	rentang C,G	mengurangi lonjakan
9	P	P	P	rentang D, H	mengurangi osilasi
10	P	N	Z	rentang I	mengerem sistem
11	N	P	Z	rentang K	mengerem sistem

(Yan, 1993 : 59)

Dari Tabel 2.7 dapat dikembangkan dengan menggunakan tujuh variabel yaitu NB (*negative big*), NM (*negative medium*), NS (*negative small*), ZE (*zero error*), PS (*positive small*), PM (*positive medium*), dan PB (*positive big*) menjadi Tabel 2.8.

**Tabel 2.8.** Aturan Kendali Fuzzy dengan Variabel NB, NM, NS, ZE, PS, PM, PB

No	E	CE	CI	Acuan	Fungsi
1	PB	ZE	PB	titik a	mengurangi waktu naik
2	PM	ZE	PM	titik e	mengurangi waktu naik
3	PS	ZE	PS	titik i	mengurangi waktu naik
4	ZE	NB	NB	titik b	mengurangi lonjakan
5	ZE	NM	NM	titik f	mengurangi lonjakan
6	ZE	NS	NS	titik j	mengurangi lonjakan
7	NB	NB	NB	titik c	mengurangi lonjakan
8	NM	NM	NM	titik g	mengurangi lonjakan
9	NS	NS	NS	titik k	mengurangi lonjakan
10	ZE	PB	PB	titik d	mengurangi osilasi
11	ZE	PM	PM	titik h	mengurangi osilasi
12	ZE	PS	PS	titik i	mengurangi osilasi
13	ZE	ZE	ZE	set point	mengerem sistem
14	PB	PM	PM	rentang a	mengurangi waktu naik
15	PS	NM	NM	rentang a	mengurangi lonjakan
16	NB	NM	NM	rentang c	mengurangi lonjakan
17	NS	PM	PM	rentang c	mengurangi lonjakan
18	PS	ZE	ZE	rentang i	mengerem sistem
19	NS	ZE	ZE	rentang k	mengerem sistem

Sumber : Yan, 1993 : 60

Dasar untuk membentuk aturan kontrol fuzzy dalam proses pengendalian dapat diformulasikan sebagai berikut :

1. Jika keluaran berupa nilai yang diinginkan dan perubahan *error* adalah *zero*, maka keluaran kontrol logika fuzzy (KLF) dibuat konstan
2. kontrol bergantung pada tanda dan nilai dari *error* dan perubahan *error*. Jika kondisi *error* dapat dikoreksi dengan cepat dengan

aksi kendali, maka aksi kontrol harus membuat konstan. Sebaliknya keluaran KLF berubah berdasarkan pada *error* dan perubahan *error*.

(Yan, 1993 : 60)

Metode inferensi merupakan proses untuk mendapatkan keluaran dari suatu kondisi masukan dengan mengikuti aturan-aturan yang telah ditetapkan. Keputusan yang didapatkan pada proses ini masih dalam bentuk fuzzy yaitu derajat keanggotaan keluaran. Pada metode Max – Min aturan operasi minimum Mamdani digunakan untuk implikasi fuzzy. Persamaan aturan minimum adalah ;

$$\mu_{C'} = \bigcup_1^n \alpha_i \wedge \mu_{Ci} \tag{2-4}$$

dengan

$$\alpha_i = \mu_{Ai}(x_0) \wedge \mu_{Bi}(y_0)$$

Sebagai contoh , terdapat dua basis kaidah atur fuzzy, yaitu :

R<sub>1</sub> : Jika x adalah A<sub>1</sub> dan y adalah B<sub>1</sub> maka z adalah C<sub>1</sub>

R<sub>2</sub> : Jika x adalah A<sub>2</sub> dan y adalah B<sub>2</sub> maka z adalah C<sub>2</sub>

Pada metode penalaran MAX-MIN fungsi keanggotaan konsekuen dinyatakan dengan,

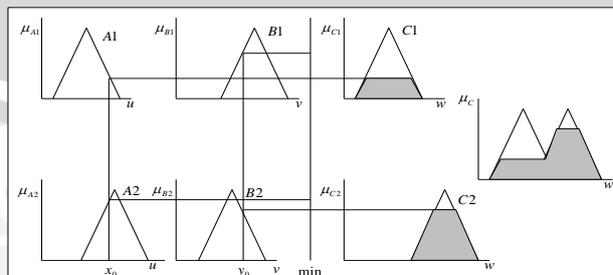
$$\mu_{C'}(W) = \mu_{C'1} \vee \mu_{C'2} = [\alpha_1 \wedge \mu_{C1}(w)] \vee [\alpha_2 \wedge \mu_{C2}(w)]$$

Dimana

$$\alpha_1 = \mu_{A1}(x_0) \wedge \mu_{B1}(y_0)$$

$$\alpha_2 = \mu_{A2}(x_0) \wedge \mu_{B2}(y_0)$$

Lebih jelas metode ini dideskripsikan dalam gambar 2.16.



**Gambar 2.16.** Inferensi Fuzzy Dengan Metode MAX-MIN

Sumber : Yan, 1994 : 55

### 2.4.3. Defuzzifikasi

Keputusan yang dihasilkan dari proses inferensi masih dalam bentuk fuzzy, yaitu derajat keanggotaan keluaran. Hasil ini harus diubah kembali menjadi variabel numerik non-fuzzy melalui proses defuzzifikasi. Metode defuzzifikasi yang umum digunakan yaitu metode COA (*center of area*). Metode COA didefinisikan dengan persamaan berikut :

$$v_o = \frac{\sum_{k=1}^m v_k \mu_k(v_k)}{\sum_{k=1}^m \mu_v(v_k)} \quad (2-5)$$

Dengan :

$v_o$  = nilai keluaran

$m$  = tingkat kuantisasi

$v_k$  = elemen ke- $k$

$\mu_v(v_k)$  = derajat keanggotaan elemen elemen pada fuzzy set  $v$

$V$  = semesta pembicaraan

(Kuswadi, 2000 : 37)

### 2.4.4. Struktur Dasar Kontrol Logika Fuzzy

Untuk merancang sistem kontrol otomatis yang mampu menterjemahkan pengetahuan dan aturan aturan fuzzy maka diperlukan teori logika fuzzy sebagai salah satu alternatif. Secara umum kontrol logika fuzzy memiliki kemampuan sebagai berikut ;

- 1) Beroperasi tanpa campur tangan manusia secara langsung, tetapi memiliki efektivitas yang sama dengan kontrol secara manual,
- 2) Mampu menangani sistem sistem yang kompleks, non-linier dan tidak stasioner,
- 3) Memenuhi spesifikasi operasional dan kriteria kinerja,
- 4) Strukturnya sederhana, kuat, dan beroperasi *real time*.

Struktur dasar sistem kontrol logika fuzzy ditunjukkan oleh Gambar 2.20.



**Gambar 2.20.** Struktur Dasar Kontrol Logika Fuzzy

*Sumber : Kuswadi,2000 : 38*

Basis pengetahuan dalam kontrol logika fuzzy adalah bagian yang berisi basis data dan basis aturan. Basis aturan berisi sejumlah aturan fuzzy yang memetakan nilai masukan fuzzy ke nilai keluaran fuzzy. Aturan ini sering dinyatakan dengan format IF-THEN. Basis data berisi fungsi keanggotaan dari himpunan fuzzy yang digunakan sebagai nilai bagi setiap variabel sistem.

Proses fuzzifikasi dari data masukan tegas (*crisp*) dilakukan dengan cara pemetaan nilai tegas variabel masukan ke semesta wacana kemudian mengkonversinya menjadi istilah linguistik yang sesuai dengan himpunan fuzzy yang telah didefinisikan.

Logika pengambilan keputusan atau inferensi adalah proses untuk mendapatkan aksi keluaran dari suatu kondisi masukan dengan mengikuti aturan-aturan (*rules*) yang telah ditetapkan. Keputusan yang dihasilkan dari proses ini masih dalam bentuk fuzzy yaitu berupa derajat keanggotaan keluaran.

Defuzzifikasi merupakan proses pemetaan himpunan fuzzy menjadi keluaran *crisp* yang digunakan untuk aksi kontrol. Metode defuzzifikasi yang digunakan pada penelitian adalah metode COA.

## BAB III

### METODOLOGI PENELITIAN

Kajian dalam skripsi ini merupakan penelitian yang bersifat aplikatif yaitu merancang suatu sistem pengendalian menggunakan kontroler fuzzy yang bertujuan agar dapat menampilkan performansi sistem sesuai dengan yang direncanakan.

Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat adalah sebagai berikut:

#### 3.1. Studi Literatur

Studi literatur mengacu pada prinsip kerja alat yang dirancang meliputi sistem pengendalian intensitas penerangan suhu menggunakan fuzzy, karakteristik dari komponen yang digunakan serta studi tentang pengendalinya yaitu menggunakan mikrokontroler ATmega16 sebagai pusat pengendali dengan kontroler fuzzy.

Adapun hal-hal yang berhubungan dengan hal tersebut adalah:

- a. Studi tentang proses pengendalian intensitas cahaya dan suhu
  - Karakteristik dari kipas DC dan peltier sebagai aktuatur pengatur suhu.
  - Karakteristik dari LED sebagai aktuatur pengatur intensitas penerangan.
- b. Studi tentang rangkaian sensor
  - Karakteristik spesifikasi sensor PIR, LDR, dan SHT11.
  - Penerapan sensor sesuai sistem.
- c. Kontroler fuzzy
  - Karakteristik umum kontroler fuzzy.
  - Teknik pemrograman *Basic Compiler AVR* pada ATmega 16 sebagai pusat pengontrol kendali fuzzy.

#### 3.2. Perancangan Sistem

Perancangan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya, hal ini dimaksudkan agar sistem tersebut dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan. Perancangan sistem yang dilakukan meliputi:

- Penentuan aktuator yang digunakan.
- Penentuan sensor yang digunakan.
- Penentuan rangkaian elektronik yang digunakan, meliputi :
  - a. Mikrokontroler.
  - b. *Driver* L293.
  - c. *Switching* rangkaian peltier.
- Penentuan spesifikasi sistem yang akan dibuat, meliputi:
  - a. Penentuan deskripsi kerja sistem secara keseluruhan.
  - b. Kontroler Logika Fuzzy dan rangkaian elektronik pendukung.
- Perancangan program pada *Basic Compiler* AVR ATmega 16 sebagai pusat pengontrol dengan kendali fuzzy.

### 3.3. Realisasi Pembuatan Sistem

Untuk merealisasikan alat atau sistem yang telah dirancang sebelumnya maka perlu diperhatikan beberapa tahapan dalam pembuatan alat tersebut, diantaranya yaitu :

- Pembuatan mekanik sistem meliputi ukuran miniatur, pemasangan serta pemilihan aktuator dan sensor.
- Pembuatan rangkaian elektronik meliputi rangkaian mikrokontroler, *driver* L293 dan *switching* rangkaian peltier.
- Pengisian program yang telah dirancang sebelumnya pada *Basic Compiler* AVR ATmega 16 dengan kontrol logika fuzzy.

### 3.4. Pengujian dan Analisis Data

Untuk memastikan sistem berjalan dengan baik maka perlu dilakukan pengujian sistem, meliputi pengujian perangkat keras (*hardware*) dan perangkat lunak (*software*) pada masing-masing blok sistem dan keseluruhan sistem.

### 3.5. Pengambilan Kesimpulan dan Saran

Kesimpulan diambil berdasarkan data yang diperoleh dari pengujian sistem secara keseluruhan. Sedangkan saran diberikan untuk pengembangan skripsi ini.

## BAB IV PERANCANGAN

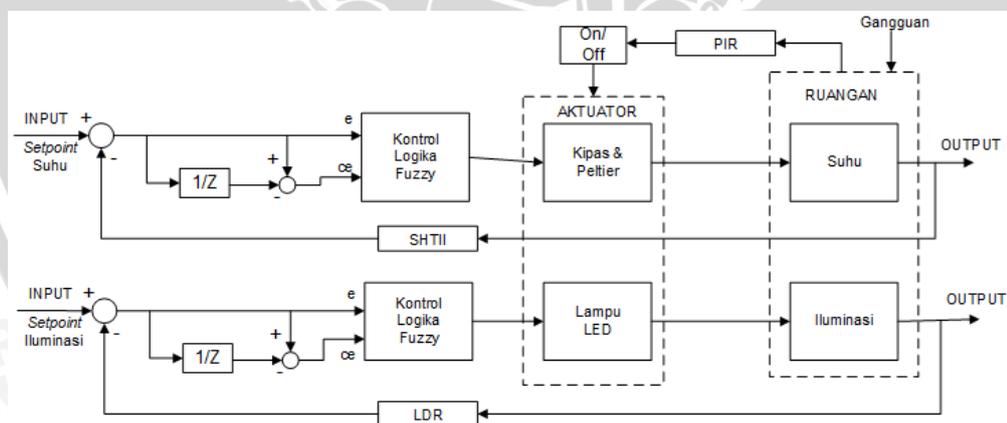
### 4.1. Perancangan Sistem

Perancangan alat ini dilakukan secara bertahap dalam bentuk blok sehingga akan memudahkan dalam analisis pada setiap bloknya maupun secara keseluruhan. Perancangan ini terdiri dari:

- Cara kerja alat.
- Spesifikasi alat.
- Perancangan perangkat keras (perancangan rangkaian mikrokontroler dan rangkaian *driver*)
- Perancangan perangkat lunak (Perancangan Algoritma Kontrol Logika Fuzzy pada *Basic Compiler AVR ATmega16*).

#### 4.1.1. Blok Diagram Sistem

Perancangan algoritma Kontrol Logika Fuzzy (KLF) pada mikrokontroler. Blok diagram sistem yang dirancang ditunjukkan dalam Gambar 4.1.



**Gambar 4.1.** Blok Diagram Sistem

Keterangan gambar 4.1 adalah sebagai berikut:

- *Input* berupa nilai *setpoint* iluminasi dan suhu diolah melalui kontrol logika fuzzy (KLF).
- KLF mengolah *error* dan *delta error* (*change error*) dari pembacaan sensor.

- Keluaran dari KLF adalah sinyal kontrol berupa nilai PWM untuk kipas dan lampu LED serta logika 1 (on) dan 0 (off) pada peltier.
- Untuk mengaktifkan aktuator (kipas, peltier, dan lampu LED) sensor PIR harus mendeteksi keberadaan manusia di dalam ruangan. Jika PIR aktif maka aktuator akan aktif (on) dan meneruskan sinyal aksi kepada *plant*. Jika PIR tidak aktif (ruangan kosong) maka aktuator tidak akan mengeluarkan sinyal aksi kepada *plant*.
- *Plant* berupa ruangan yang dikontrol iluminasi dan suhunya.
- *Output* dari *plant* akan dikembalikan (*feedback*) pada *input* melalui sensor intensitas pencahayaan LDR dan sensor suhu SHT11, hingga mencapai *setpoint*.
- Gangguan berupa penambahan jumlah tikus dan penambahan iluminasi dari luar.

#### 4.1.2. Prinsip Kerja Alat

Prinsip kerja alat adalah sebagai berikut:

- Sensor LDR mengukur kuat penerangan (iluminasi) dan sensor SHT11 mengukur suhu didalam miniatur ruangan.
- Sensor PIR mendeteksi keberadaan tikus (sebagai pengganti manusia) didalam miniatur ruangan.
- Jika sensor PIR aktif, maka sistem aktif. Jika PIR tidak aktif berapapun nilai suhu dan iluminasi, sistem kontrol tidak akan aktif.
- Jika suhu di dalam miniatur ruangan kurang dari *set point* maka peltier aktif sebagai pemanas. Jika suhu melebihi *set point* maka peltier aktif sebagai pendingin ruangan.
- Kecepatan kipas pada peltier menyesuaikan dengan kondisi suhu sesuai dengan *set point*.
- Jika intensitas penerangandi dalam miniatur ruangan kurang dari *set point* maka lampu LED aktif. Jika iluminasi di dalam miniatur ruangan melebihi *set point* maka lampu LED tidak aktif.

## 4.2. Perancangan Perangkat Keras (*hardware*)

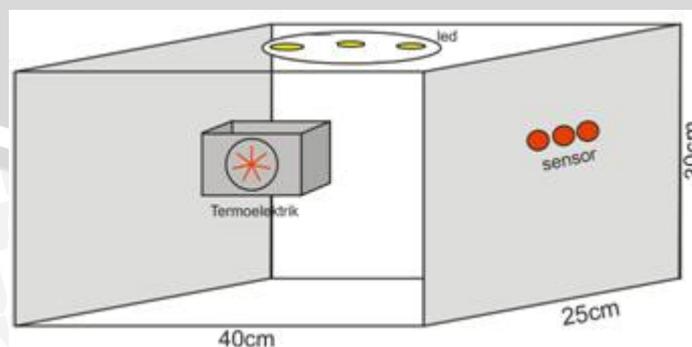
Berdasarkan blok diagram sistem yang telah disusun, perancangan perangkat keras meliputi perancangan mekanik, rangkaian sensor (SHT11, LDR, dan PIR), rangkaian *driver* (L293) dan rangkaian *switching* (*optocoupler-relay*), serta penggunaan modul AT Mega-16. Berikut ini adalah penjelasan masing-masing rangkaian penyusun keseluruhan alat.

### 4.2.1. Spesifikasi Alat

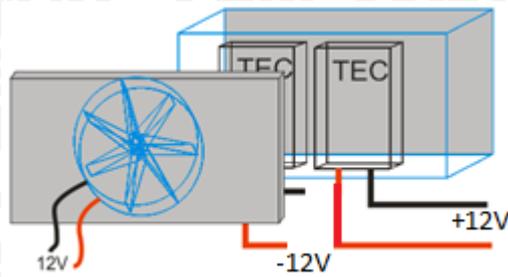
Spesifikasi miniatur ruangan adalah sebagai berikut :

- Miniatur ruangan terbuat dari bahan akrilik transparan 3mm.
- Ukuran miniatur adalah 40cm x 30cm x 25cm (p x l x t).
- Catu daya yang digunakan adalah +12V dan -12V DC untuk rangkaian *driver* dan *switching*, 5 VDC untuk rangkaian mikrokontroler dan sensor.
- 2 buah peltier masing-masing digunakan untuk pemanas dan pendingin. Keduanya dirangkaikan dengan 1 buah kipas dalam untuk mengalirkan panas atau dingin kedalam sistem dan 1 buah kipas luar untuk membuang panas yang dihasilkan ke luar sistem.
- Lampu LED 3 Watt.
- Sensor yang digunakan yaitu SHT11 untuk mengukur suhu dan LDR untuk mengukur intensitas cahaya, dan PIR untuk mendeteksi keberadaan tikus (sebagai ganti manusia) pada miniatur.

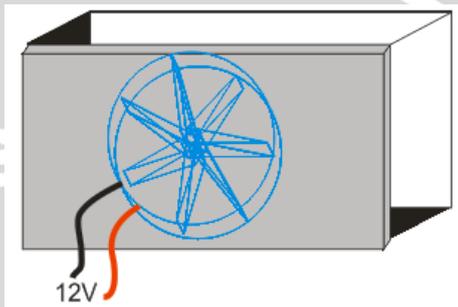
Skema miniatur ruangan dan peltier ditunjukkan oleh Gambar 4.2 dan Gambar 4.3 serta Gambar 4.4.



Gambar 4.2. Skema miniatur ruangan



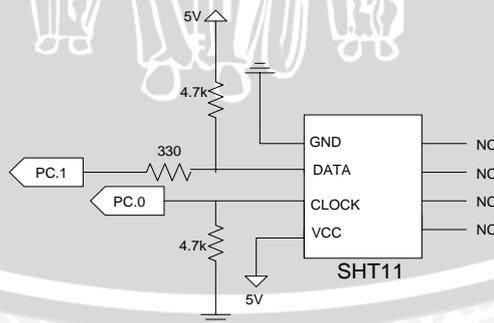
Gambar 4.3. Skema Pendingin/Pemanas Peltier Tampak Dalam



Gambar 4.4. Skema Pendingin/Pemanas Peltier Tampak Luar

#### 4.2.2. Sensor SHT 11

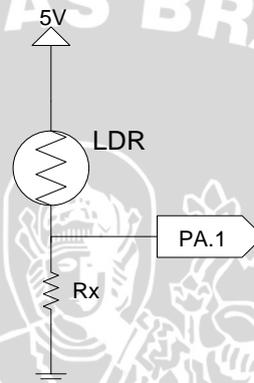
Sensor SHT 11 digunakan untuk mengukur suhu. Sensor ini membutuhkan catu tegangan 2.4 V sampai 5.5V. Keluaran dari sensor sudah berupa sinyal digital dan dikomunikasikan secara *2-wire* melalui pin C.0 dan pin C.1 pada mikrokontroler ATmega 16. Pin C.0 digunakan untuk mengaktifkan *clock* sedangkan pin C.1 digunakan untuk mengirim dan menerima data. Skema rangkaian sensor SHT11 dapat dilihat pada Gambar 4.5.



Gambar 4.5. Skema Rangkaian Sensor SHT11

#### 4.2.3. Sensor LDR

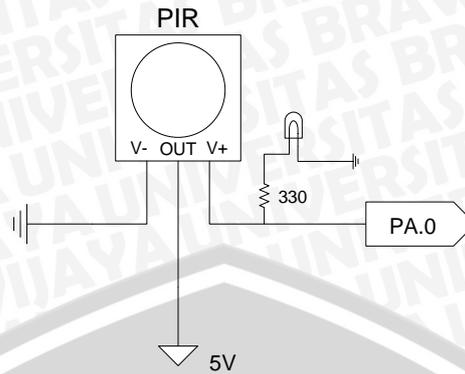
Sensor LDR digunakan untuk mengukur intensitas cahaya di dalam miniatur ruangan. Keluaran dari sensor ini berupa perubahan nilai resistansi. LDR dihubungkan dengan catu tegangan 5 V, sehingga setiap perubahan nilai resistansi juga memberi perubahan nilai tegangan. Nilai tegangan ini diproses pada mikrokontroler melalui pin ADC. Untuk mengetahui nilai intensitas cahaya (lux) maka hasil pembacaan rangkaian LDR dikalibrasikan dengan hasil pembacaan alat pengukur intensitas cahaya, yaitu luxmeter. Skema rangkaian sensor LDR dapat dilihat pada Gambar 4.6.



**Gambar 4.6.** Skema Rangkaian Sensor LDR

#### 4.2.4. Sensor PIR

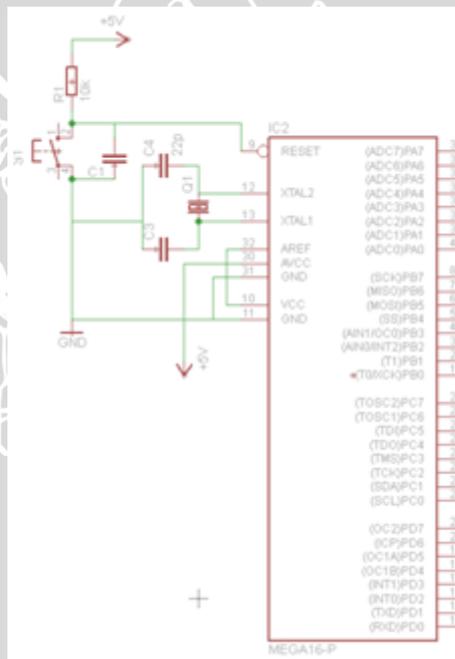
Sensor PIR digunakan untuk mendeteksi adanya gerakan tikus, sebagai pengganti manusia, di dalam miniatur. Sensor PIR menangkap pancaran sinar infra merah yang dihasilkan oleh tubuh yang bergerak. Sensor PIR diberi catu tegangan 5 V dan keluarannya dihubungkan ke mikrokontroler. Keluaran sensor berupa tegangan 5V saat mendeteksi adanya gerakan dan 0 V saat tidak ada gerakan. Skema rangkaian sensor PIR dapat dilihat pada Gambar 4.7.



Gambar 4.7. Skema rangkaian sensor PIR

#### 4.2.5. Modul Mikrokontroler AVR ATmega 16

Pada sistem menggunakan mikrokontroler ATmega16 sebagai pengolah dalam proses pengaturan. Konfigurasi kaki I/O dari mikrokontroler ATmega16 ditunjukkan oleh Gambar 4.8.



Gambar 4.8. Rangkaian Minimum Mikrokontroler

Mikrokontroler ATmega16 mempunyai 4 port, 32 jalur yang dapat diprogram menjadi masukan atau keluaran. Pin masukan dan keluaran mikrokontroler pada perancangan ini akan difungsikan sesuai Tabel 4.1.

Tabel 4.1. Fungsi Pin Mikrokontroler

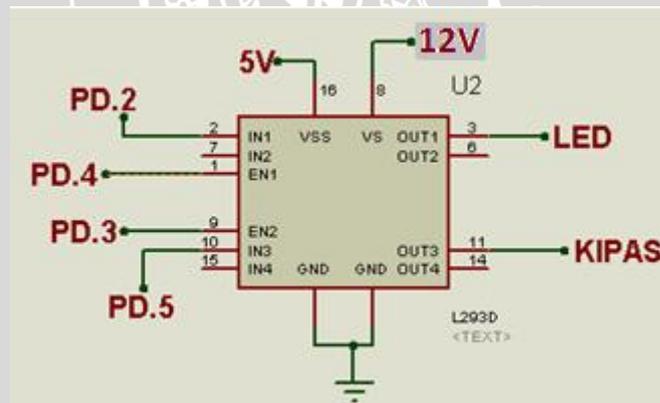
PIN	FUNGSI
A.0	Masukan dari sensor PIR

A.1	Masukan dari sensor LDR
C.0	Keluaran clock sensor SHT11
C.1	Masukan-keluaran data sensor SHT11
D.0	Keluaran Relay 1 (aktivasi mode pendingin)
D.1	Keluaran Relay 2 (aktivasi mode pemanas)
D.2	Keluaran aktivasi lampu LED (Masukan1 driver L293D)
D.3	Keluaran aktivasi kipas(Masukan2 driver L293D)
D.4	Keluaran PWM 1a
D.5	Keluaran PWM 1b

Sumber : Perancangan

#### 4.2.6. Driver L293

Driver L293 digunakan untuk mengatur nilai PWM pada keluaran berupa kipas dan lampu LED. Skema rangkaian dari driver L293D dapat dilihat pada Gambar 4.9.

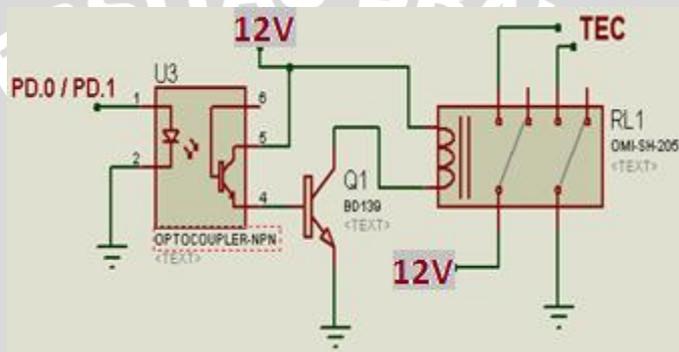


Gambar 4.9. Skema rangkaian driver L293D

Driver L293D memiliki 2 jalur PWM, 4 jalur masukan, dan 4 jalur keluaran. Driver diberi catu tegangan 12V dan hanya menggunakan 3 jalur masukan dan 3 jalur keluaran dengan 2 jalur PWM yang berbeda, yaitu PWM 1a dan PWM 1b. PWM 1a digunakan untuk mengatur nilai PWM pada rangkaian lampu led, sedangkan PWM 1b digunakan untuk mengatur nilai PWM pada kipas pendingin dan kipas pemanas.

#### 4.2.7. Rangkaian Switching

Rangkaian *switching* terdiri dari 3 komponen utama, yaitu *optocoupler*, transistor dan relay, seperti ditunjukkan Gambar 4.10. *Optocoupler* berfungsi sebagai penerima logika dari mikrokontroler yang kemudian diteruskan pada transistor untuk mengaktifkan relay. *Optocoupler* juga berfungsi untuk memisahkan antara tegangan tinggi (12V) yang digunakan peltier dan tegangan rendah (5V) yang digunakan mikrokontroler. Sedangkan relay berfungsi untuk proses *switching* peltier yang menggunakan arus tinggi.



Gambar 4.10. Skema rangkaian switching pemanas

### 4.3. Perancangan Sistem Kontrol Logika Fuzzy

#### 4.3.1. Variabel Masukan dan Keluaran

Sistem kendali logika fuzzy yang dikembangkan dalam penelitian ini yaitu kendali logika fuzzy untuk masukan *error* dan *deltaerror* untuk suhu dengan keluaran nilai PWM kipas, serta *error* dan *deltaerror* untuk iluminasi dengan keluaran nilai PWM lampu LED. *Error* merupakan selisih antara keluaran aktual dan *setpoint*. *Deltaerror* atau perubahan *error* merupakan selisih *error* pada waktu proses dengan *error* pada waktu proses sebelumnya. *Error* dan *deltaerror* ditunjukkan oleh persamaan 5-1 dan 5-2.

$$E(k) = T_{sp}(k) - T_r(k) \quad (5-1)$$

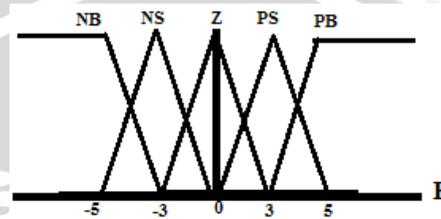
$$dE(k) = E(k) - E(k-1) \quad (5-2)$$

#### 4.3.2. Fungsi Keanggotaan Masukan

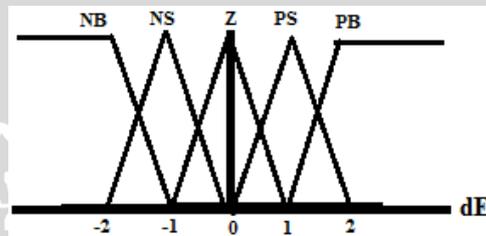
Fungsi keanggotaan masukan *error* dan *delta error* untuk suhu maupun iluminasi masing-masing terdiri atas lima label yaitu *negative big*

(NB), *negative small* (NS), *zero* (Z), *positive small* (PS), *positive big* (PB). Data masukan dari fungsi keanggotaan suhu didapat dari sensor SHT11 dan sedangkan untuk fungsi keanggotaan iluminasi didapat dari sensor LDR.

Dengan menggabungkan hasil dari penelitian Mom & Wiesebron, ditentukan nilai *setpoint* suhu adalah  $25^{\circ}\text{C}$  dan untuk nilai keanggotaan *error* dan *deltaerror* ditunjukkan oleh Gambar 4.11 dan 4.12.

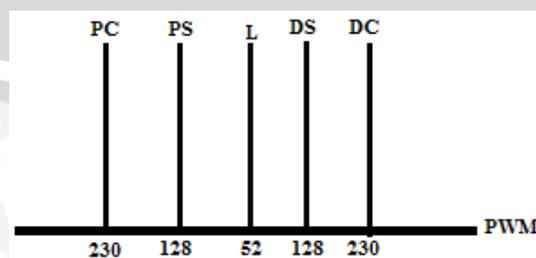


**Gambar 4.11.** Fungsi Keanggotaan Masukan *Error* Suhu



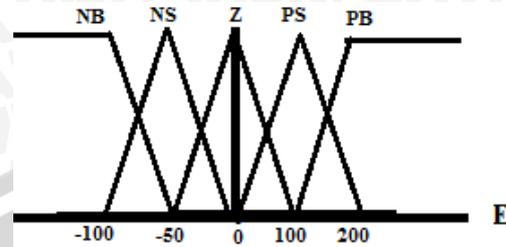
**Gambar 4.12.** Fungsi Keanggotaan Masukan *Deltaerror* suhu

Untuk nilai keanggotaan keluaran berupa nilai PWM kipas berdasarkan hasil pengujian PWM terhadap kecepatan putaran kipas. Bentuk fungsi keluaran menggunakan *singleton* untuk mempersingkat proses perhitungan. Fungsi keanggotaan keluaran PWM kipas memiliki 5 label yaitu *panas cepat* (PC), *panas sedang* (PS), *lambat* (L), *dingin sedang* (DS), dan *dingin cepat* (DC). Untuk label PS dan PC peltier aktif sebagai pemanas dan untuk label DS dan DC peltier aktif sebagai pendingin, sedangkan untuk label L peltier tidak aktif sama sekali. Gambar 4.13 menunjukkan fungsi keanggotaan keluaran PWM kipas.

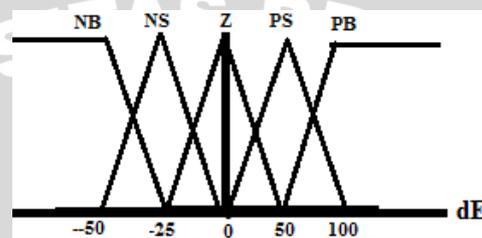


**Gambar 4.13.** Fungsi Keanggotaan Keluaran PWM Kipas

Untuk nilai iluminasi menggunakan Tabel 2.2 sebagai rujukan, dan ditentukan nilai *setpoint* 200 lux. Untuk nilai *error* dan *deltaerror* ditunjukkan oleh Gambar 4.14 dan 4.15.

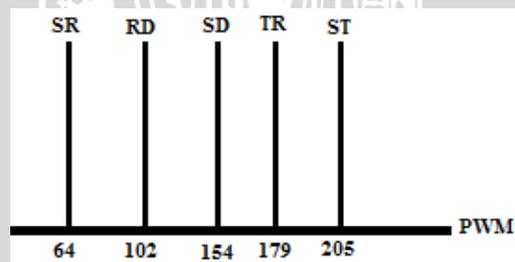


Gambar 4.14. Fungsi Keanggotaan Masukan *Error* Iluminasi



Gambar 4.15. Fungsi Keanggotaan Masukan *Deltaerror* Iluminasi

Untuk nilai keanggotaan keluaran iluminasi berupa nilai PWM LED berdasarkan hasil pengujian PWM terhadap iluminasi yang dihasilkan lampu LED. Bentuk fungsi keluaran menggunakan *singleton* untuk mempersempit proses perhitungan. Fungsi keanggotaan keluaran PWM kipas memiliki 5 label yaitu *sangat redup* (SR), *redup* (RD), *sedang* (SD), *terang* (TR), dan *sangat terang* (ST). Gambar 4.16 menunjukkan fungsi keanggotaan keluaran PWM lampu LED.



Gambar 4.16. Fungsi Keanggotaan Keluaran PWM LED

### 4.3.3. Perancangan Aturan Fuzzy

Perancangan aturan fuzzy didasarkan pada pengetahuan kualitatif sistem *loop* tertutup dan *delta error* dalam kondisi positif atau negatif digunakan tanggapan undak (*step response*). Dengan menggunakan aturan kontrol logika fuzzy terhadap tanggapan undak sistem yang disusun

Yan,1993, aturan yang dihasilkan memiliki 25 aturan untuk suhu dan 25 untuk iluminasi, diantaranya :

- Kondisi 1 untuk suhu yaitu saat *error* sistem lebih besar dari  $5^{\circ}\text{C}$  ( $E=PB$ ) dan respon sistem masih tahap awal ( $dE=Z$ ). Dalam kondisi ini peltier pendingin harus aktif dan kipas berputar cepat ( $DC$ ) agar suhu cepat turun menuju *setpoint*. Untuk iluminasi saat *error* lebih besar dari 200 ( $E=PB$ ) dan respon sistem masih tahap awal ( $dE=Z$ ) maka lampu harus mengeluarkan cahaya serendah mungkin ( $SR$ ) agar iluminasi turun menuju *setpoint*.
- Kondisi 2 untuk suhu yaitu saat *error* sistem sudah kurang  $5^{\circ}\text{C}$  ( $PS$ ) dan respon sistem naik ( $dE=NS$ ). Dalam kondisi ini peltier tidak aktif dan kipas berputar lambat ( $L$ ) agar respon sistem lebih landai. Untuk iluminasi pada kondisi ini lampu cukup mengeluarkan iluminasi sedang ( $SD$ ).
- Kondisi 3 untuk suhu yaitu saat *error* sistem kurang dari  $5^{\circ}\text{C}$  ( $PS$ ) dan respon sistem turun ( $dE=PS$ ). Dalam kondisi ini sistem sedang menuju *setpoint* dan *error* mengecil sehingga peltier pendingin masih aktif dan kipas cukup berputar sedang ( $DS$ ). Untuk iluminasi pada kondisi ini cahaya lampu harus sangat rendah ( $SR$ ).

Proses penurunan aturan dilanjutkan terus sampai kondisi ke-25 dan hasilnya dapat diringkas seperti pada Tabel 4.2.

**Tabel 4.2.** Aturan Fuzzy Masukan Error – Deltaerror Suhu Dengan Keluaran PWM Kipas

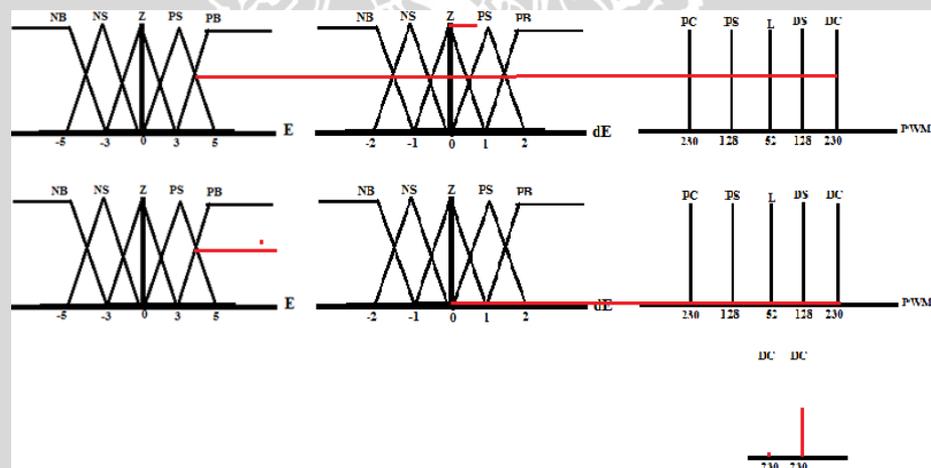
<b>dE</b> <b>E</b>	<b>NB</b>	<b>NS</b>	<b>Z</b>	<b>PS</b>	<b>PB</b>
<b>NB</b>	PC	PC	PC	PS	L
<b>NS</b>	PC	PS	PS	L	PS
<b>Z</b>	PC	PS	L	DS	DC
<b>PS</b>	DS	L	DS	DC	DC
<b>PB</b>	L	DS	DC	DC	DC

**Tabel 4.3.** Aturan Fuzzy Masukan Error – Deltaerror Iluminasi Dengan Keluaran PWM LED

$dE$	NB	NS	Z	PS	PB
NB	ST	ST	ST	TR	SD
NS	ST	ST	TR	SD	RD
Z	ST	TR	SD	RD	SR
PS	TR	SD	RD	SR	SR
PB	SD	RD	SR	SR	SR

#### 4.3.4. Metode Inferensi Max-Min

Untuk mengetahui proses metode ini pada sistem, dimisalkan visualisasi proses pengolahan masukansuhu SHT11 membaca nilai  $29^{\circ}\text{C}$  dengan  $error = 4$  dan  $deltaerror = 0$ , maka dengan metode Max-Min yang telah ditetapkan ditunjukkan dalam Gambar 4.17.



**Gambar 4.17.** Proses Inferensi Max-Min

#### 4.3.5. Defuzzifikasi

Defuzzifikasi adalah proses untuk mengubah keluaran *fuzzy* menjadi keluaran *crisp*. Hasil defuzzifikasi inilah yang digunakan untuk mengatur besarnya rasio kecepatan pada masing-masing motor DC. Metode defuzzifikasi yang digunakan adalah COA (*Center of Area*).

Berdasarkan contoh hasil proses inferensi pada Gambar 4.17, maka sesuai dengan Persamaan 2-9, didapatkan nilai keluaran kipas:

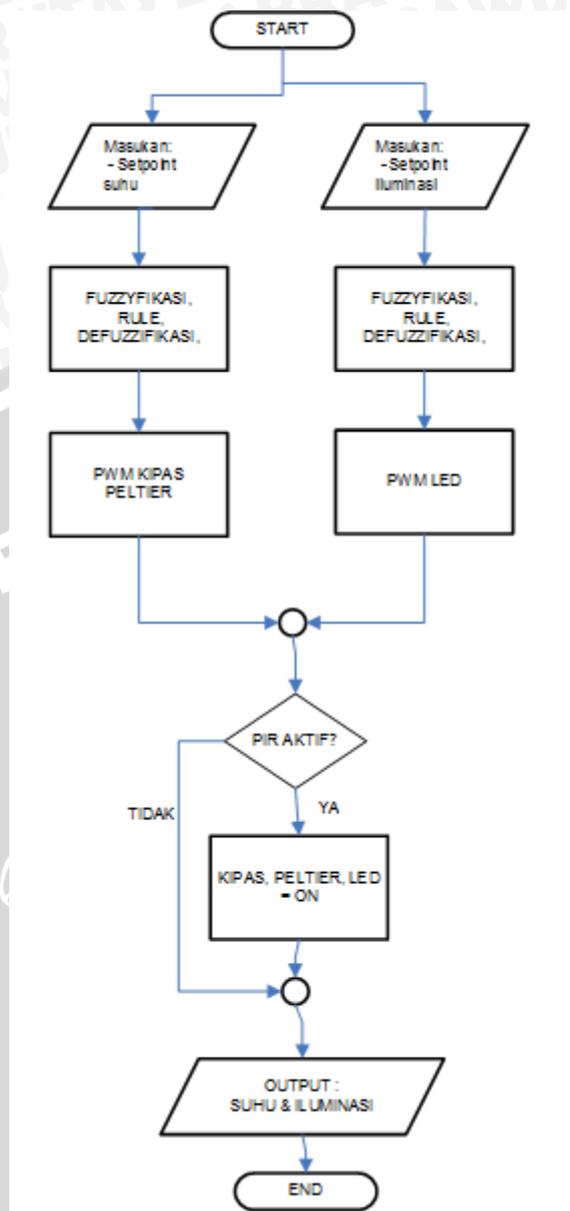
$$\begin{aligned}U &= \frac{\sum_{i=1}^n w_i u_i}{\sum_{i=1}^n w_i} \\ &= \frac{0.5 \times DC + 0 \times DC}{0.5 + 0} \\ &= \frac{0.5 \times (230) + 0 \times (230)}{0.5 + 0} \\ &= 230\end{aligned}$$

Jadi, pada saat sensor SHT membaca suhu  $29^{\circ}\text{C}$ , maka nilai PWM keluaran kipas adalah 230 dan peltier pendingin aktif.

#### 4.4. Perancangan Perangkat Lunak (*software*)

Diagram alir (*flowchart*) perancangan perangkat lunak ditunjukkan dalam Gambar 4.18.





Gambar 4.18. Diagram Alir Perangkat Lunak Sistem

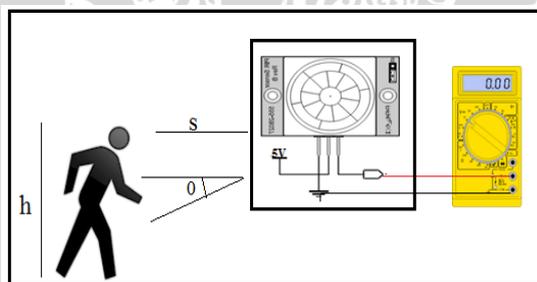
## BAB V PENGUJIAN DAN ANALISIS

Pengujian dan analisis dilakukan untuk mengetahui apakah sistem dapat bekerja sesuai perancangan yang telah dilakukan. Pengujian dilakukan pada tiap blok sistem kemudian secara keseluruhan. Adapun pengujian yang dilakukan adalah sebagai berikut :

3. Pengujian sensor yaitu sensor gerak PIR, sensor cahaya LDR, dan sensor suhu SHT11.
4. Pengujian aktuator yaitu kipas, lampu LED, dan termoelektrik.
5. Pengujian keseluruhan sistem.

### 5.1. Pengujian Sensor PIR

Pengujian sensor PIR (*Passive Infra Red*) bertujuan untuk melihat respon sensor terhadap pergerakan manusia sebagai objek. Prosedur pengujian dilakukan dengan mengukur tegangan keluaran dari sensor PIR seperti ditunjukkan Gambar 5.1.



**Gambar 5.1.** Diagram Blok Pengujian Sensor PIR

Pengujian dilakukan dengan meletakkan sensor PIR pada ketinggian 100 cm diatas permukaan tanah dengan ketinggian (h), jarak (s) dan sudut objek terhadap sensor berubah ubah. Hasil pengujian ditunjukkan oleh Tabel 5.1.

**Tabel 5.1.** Hasil Pengujian Sensor PIR

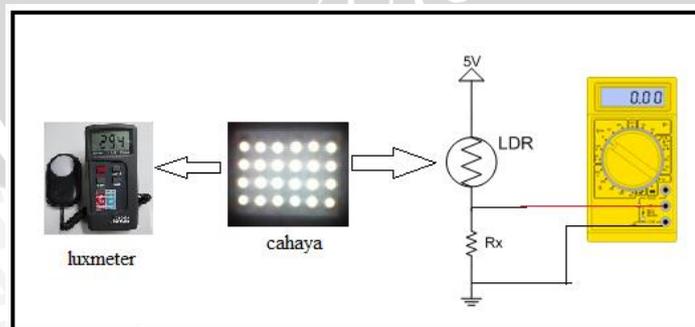
$h_{\text{objek}}$ (cm)	$s_{\text{objek}}$ (cm)	sudut objek ( $^{\circ}$ )	$V_{\text{out}}$ (V)
30	30	0	0
60	30	20	0
60	30	45	4,93
70	30	45	4,93

30	60	0	0
60	60	20	0
60	60	45	4,93
70	60	45	4,93
30	90	0	0
60	90	20	0
60	90	45	4,93
70	90	45	4,93
30	120	0	0
60	120	20	4,93
60	120	45	4,93
70	120	45	4,93
30	150	0	0
60	150	20	4,93
60	150	45	4,93
70	150	45	4,93

Dari Tabel 5.1 diperoleh kesimpulan bahwa objek dapat terdeteksi oleh sensor PIR pada jarak 30 cm sampai 150 cm dan jarak pemasangan sensor dengan tinggi objek tidak kurang dari 40 cm.

### 5.2. Pengujian Sensor LDR

Pengujian bertujuan untuk mengkalibrasi sensor LDR terhadap alat ukur intensitas cahaya (E) yaitu luxmeter. Prosedur pengujian dilakukan dengan mengukur intensitas cahaya lampu dalam ruang dengan menggunakan luxmeter KW06-22 (lux) dan rangkaian LDR (volt), seperti ditunjukkan Gambar 5.2. Tegangan masukan pada LDR ( $V_{in}$ ) sebesar 5 volt dan resistor referensi ( $R_{ref}$ ) 10 k $\Omega$ .



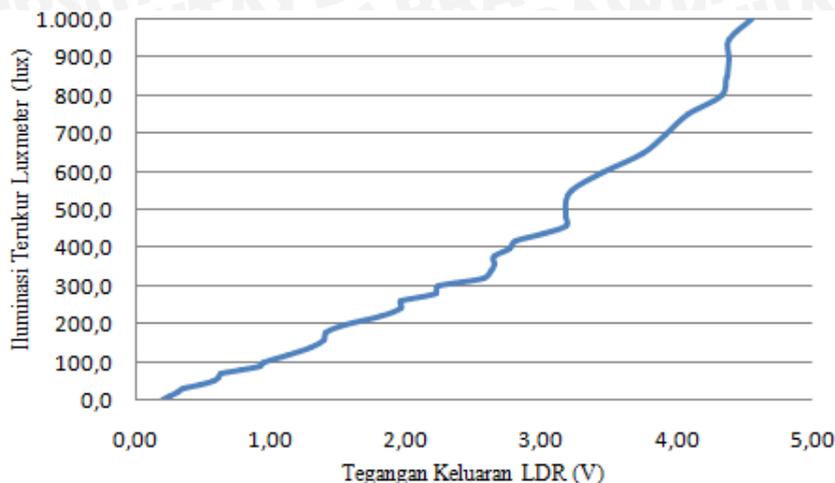
Gambar 5.2. Diagram Blok Pengujian LDR

Hasil pengujian tegangan keluaran LDR terhadap sumber cahaya dibandingkan dengan nilai intensitas cahaya yang diukur oleh luxmeter ditunjukkan oleh Tabel 5.2

**Tabel 5.2.** Hasil Pengujian LDR

E (lux)	Vout (V)	E (lux)	Vout (V)
0	0,20	300	2,24
10	0,25	320	2,57
20	0,31	340	2,63
30	0,35	360	2,65
40	0,47	380	2,65
50	0,57	400	2,76
60	0,61	420	2,80
70	0,63	440	3,01
80	0,78	460	3,18
90	0,92	480	3,18
100	0,94	500	3,18
120	1,12	550	3,22
140	1,29	600	3,46
160	1,39	650	3,75
180	1,41	700	3,92
200	1,55	750	4,08
220	1,82	800	4,33
240	1,96	850	4,37
250	1,96	900	4,39
260	1,96	950	4,39
280	2,22	1000	4,55

Dari Tabel 5.2 dapat ditampilkan dalam bentuk grafik pada Gambar 5.3.



**Gambar 5.3.** Grafik Tegangan Keluaran LDR terhadap Iluminasi yang Diukur Luxmeter

Untuk melakukan kalibrasi antara LDR dengan luxmeter, hasil pengujian pada Tabel 5.2 dimasukkan dalam persamaan regresi yaitu ;

$$y = a + bx \tag{5-1}$$

dimana,

$$a = \frac{(\sum Y)(\sum x^2) - (\sum x)(\sum xy)}{n \sum x^2 - (\sum x)^2} \tag{5-2}$$

$$b = \frac{n(\sum xy) - (\sum x)(\sum y)}{n \sum x^2 - (\sum x)^2} \tag{5-3}$$

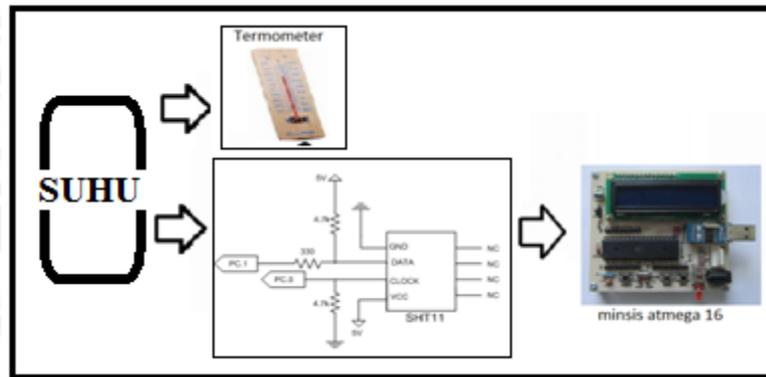
Melalui Persamaan 5-1, 5-2, dan 5-3 maka diperoleh persamaan kalibrasi LDR terhadap luxmeter yaitu :

$$Y=204,5x - 106,6$$

### 5.3. Pengujian Sensor Suhu SHT11

Pengujian bertujuan untuk mengetahui sensitivitas sensor SHT11 dengan membandingkan hasil pembacaan suhu sensor SHT11 terhadap termometer air raksa seperti ditunjukkan Gambar 5.3.





**Gambar 5.3.** Diagram Blok Pengujian Sensor SHT11

Hasil pengujian sensor SHT11 terhadap termometer ditunjukkan oleh Tabel 5.3.

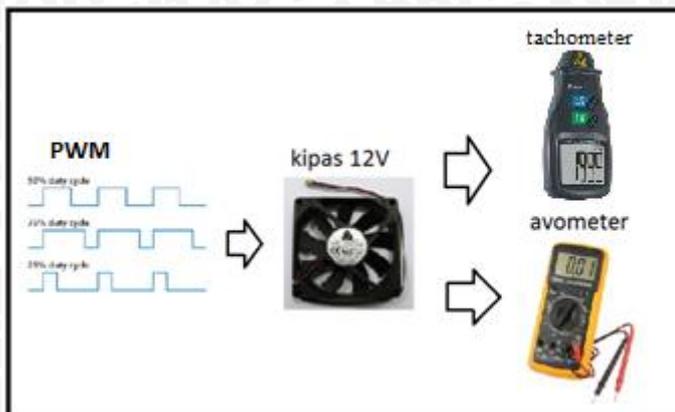
**Tabel 5.3.** Data Pengujian Sensor SHT11

No. Sampling	Suhu ( $^{\circ}\text{C}$ )		%Error
	Termometer ( $^{\circ}\text{C}$ )	SHT11 ( $^{\circ}\text{C}$ )	
1	24	24,12	0,5
2	25	24,72	1,13
3	26	25,96	0,15
4	27	27,39	1,42
5	28	28,38	1,34
6	29	28,97	0,1
7	30	30,12	0,4
8	31	31,24	0,77
9	32	32,15	0,47
10	33	33,06	0,18
% Error Rata-Rata			0,65

Data hasil pengujian Tabel 5.3 menunjukkan bahwa kesalahan (error) rata-rata pembacaan suhu pada SHT11 terhadap termometer air raksa adalah 0,65%. Nilai kesalahan ini dianggap wajar karena SHT11 memiliki akurasi suhu  $\pm 0,4^{\circ}\text{C}$ .

#### 5.4. Pengujian Kipas

Pengujian bertujuan untuk mengetahui kecepatan putaran yang dihasilkan oleh kipas dalam RPM dengan dengan nilai PWM yang bervariasi. Prosedur pengujian ditunjukkan oleh Gambar 5.4.



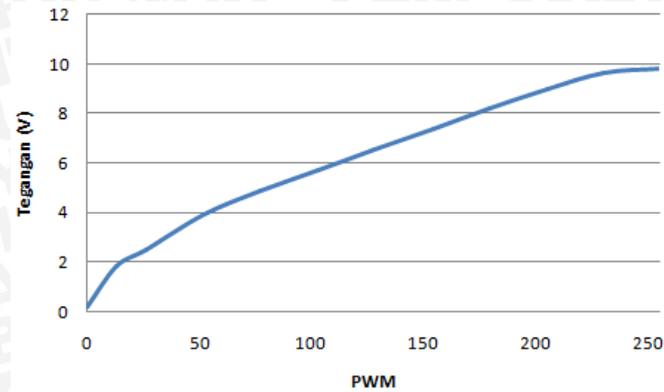
Gambar 5.4. Diagram Blok Pengujian Kipas

Hasil pengujian tegangan dan laju aliran udara yang dihasilkan kipas dengan variasi nilai PWM ditunjukkan oleh Tabel 5.4.

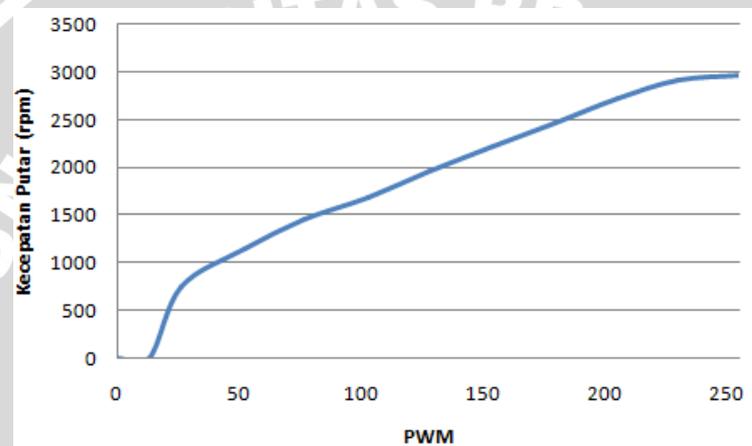
Tabel 5.4. Hasil Pengujian Kipas

PWM	$V_{in}$ (V)	RPM
255	9,79	2963
230	9.63	2914
205	9,01	2726
179	8,22	2466
154	7,4	2220
128	6,55	1964
103	5,69	1689
77	4,86	1455
52	3,9	1140
26	2,5	748
13	1,82	0
0	0	0

Data pengujian dari Tabel 5.4 menunjukkan bahwa kipas dapat berputar dengan PWM minimum 26 atau saat tegangan 2,5V. Kecepatan kipas dengan PWM 255 adalah 2963 rpm. Grafik hubungan antara nilai PWM dengan tegangan dan kecepatan putaran pada kipas ditunjukkan oleh Gambar 5.5 dan Gambar 5.6.



**Gambar 5.5.** Grafik Nilai Tegangan Input Kipas Terhadap Nilai PWM

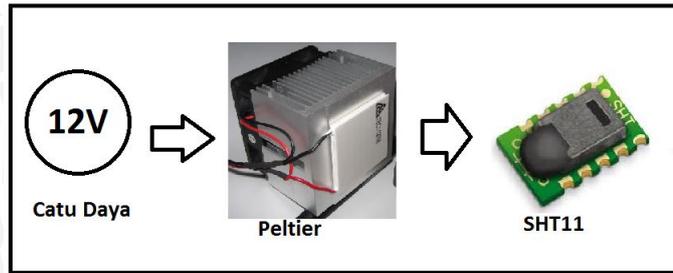


**Gambar 5.6.** Grafik Kecepatan Putaran Kipas Terhadap Nilai PWM

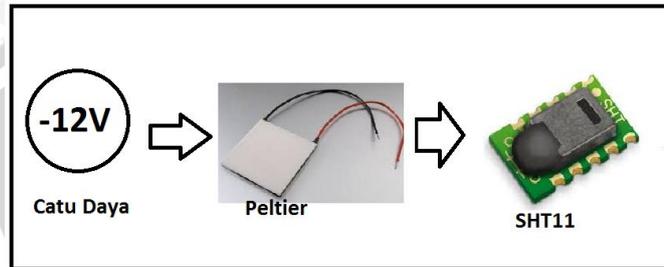
Dari grafik Gambar 5.5 dan 5.6 dapat dilihat bahwa nilai tegangan masuk input dan kecepatan putaran kipas berbanding lurus dengan nilai PWM yang diberikan.

### 5.5. Pengujian Peltier

Pengujian bertujuan untuk mengetahui respon suhu permukaan peltier terhadap waktu. Prosedur pengujian dilakukan dengan memberi catu tegangan 12V DC pada peltier kemudian suhu permukaan peltier diukur dengan SHT11. Pengujian terdiri dari 2 macam, yaitu pengujian sisi dingin dengan sisi panas dipasangi *heatsink* dengan catu daya 12V dan pengujian sisi panas dengan catu daya -12V. Prosedur pengujian ditunjukkan oleh Gambar 5.7 dan 5.8.



Gambar 5.7. Diagram Blok Pengujian Sisi Dingin Peltier



Gambar 5.8. Diagram Blok Pengujian Sisi Panas Peltier

Hasil pengujian suhu permukaan peltier (sisi dingin dan sisi panas) ditunjukkan oleh Tabel 5.5 dan 5.6.

Tabel 5.5. Hasil Pengujian Suhu Permukaan Peltier Sisi Dingin

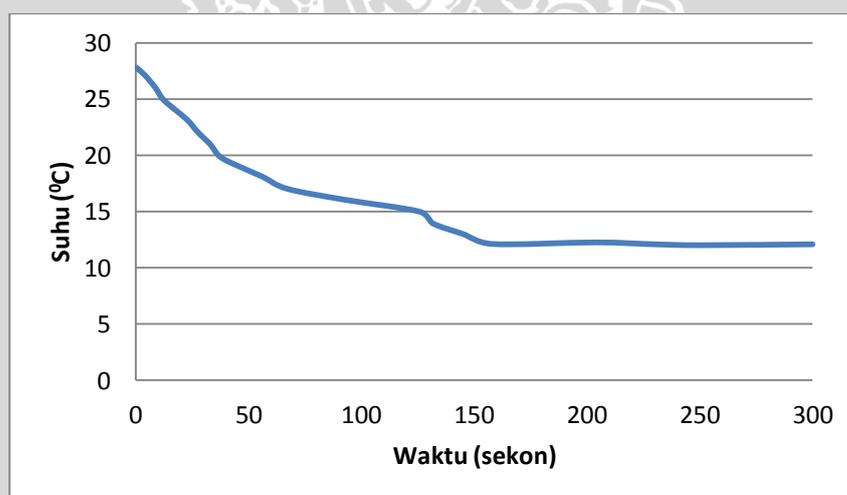
Suhu Permukaan Peltier (°C)	Waktu (detik)	Suhu Permukaan Peltier (°C)	Waktu (detik)
27,83	0	17,04	67
27,12	4	16,08	92
26,02	9	15,01	126
25,00	12	13,95	132
24,13	17	13,03	145
23,12	23	12,14	158
22,04	28	12,28	204
21,00	33	12,11	230
19,89	37	12,03	250
19,10	45	12,11	300
18,12	56	12,11	400

Tabel 5.6. Hasil Pengujian Suhu Permukaan Peltier Sisi Panas

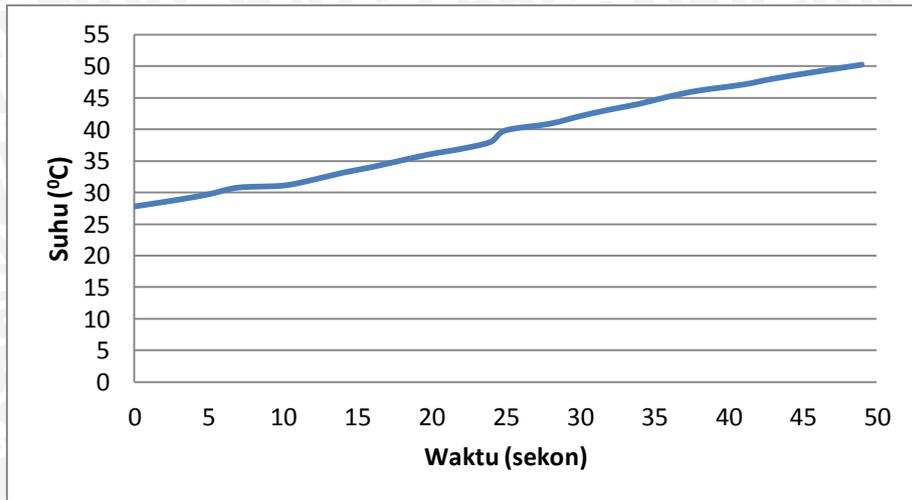
Suhu Permukaan Peltier (°C)	Waktu (detik)	Suhu Permukaan Peltier (°C)	Waktu (detik)
27,82	0	38,04	24
28,91	3	39,85	25
29,75	5	40,92	28
30,81	7	42,1	30

31,12	10	43,14	32
32,02	12	44,05	34
33,14	14	45,21	36
34,05	16	46,15	38
35,11	18	47,11	41
36,12	20	48,02	43
36,93	22	49,17	46
37,5	23	50,25	49

Data pengujian pada Tabel 5.5 menunjukkan bahwa suhu permukaan peltier yang dihasilkan sisi dingin peltier mencapai  $12,14^{\circ}\text{C}$  setelah 158 detik dan tetap *steady* setelah 300 detik. Pada Tabel 5.6 suhu permukaan sisi panas peltier terus naik secara linier dan dibatasi saat mencapai suhu  $50,25^{\circ}\text{C}$  setelah 49 detik untuk mencegah kerusakan pada peltier. Dari hasil pengujian diatas dapat diperoleh grafik hubungan antara suhu permukaan peltier terhadap waktu yang ditunjukkan gambar 5.9 dan 5.10.



**Gambar 5.9.** Grafik Suhu Permukaan Sisi Dingin Peltier Terhadap Waktu

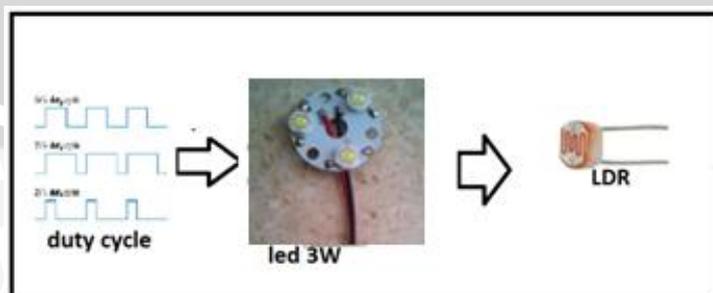


**Gambar 5.10.** Grafik Suhu Permukaan Sisi Panas Peltier Terhadap Waktu

Dari Gambar 5.9 dan Gambar 5.10 dapat dilihat bahwa suhu permukaan sisi dingin peltier *steady* pada suhu 12,14 °C sedangkan suhu permukaan sisi panas terus naik secara linier. Hal ini dikarenakan untuk menghasilkan suhu dingin, maka suhu pada sisi panas peltier harus dibuang agar suhu pada sisi dingin terus turun. Kemampuan alat untuk membuang suhu panas yang sudah mencapai titik jenuh tidak dapat mengimbangi suhu panas yang terus naik menyebabkan suhu dingin *steady* pada suhu 12,14 °C.

### 5.6. Pengujian Lampu LED

Pengujian ini bertujuan untuk mengetahui nilai iluminasi atau tingkat kuat penerangan lampu LED menggunakan LDR yang sudah dikalibrasi untuk menunjukkan nilai dalam lux. Prosedur pengujian ditunjukkan oleh Gambar 5.11. Jarak LDR terhadap lampu LED (sumber cahaya) adalah 30 cm dan diberi catu 12 volt DC.



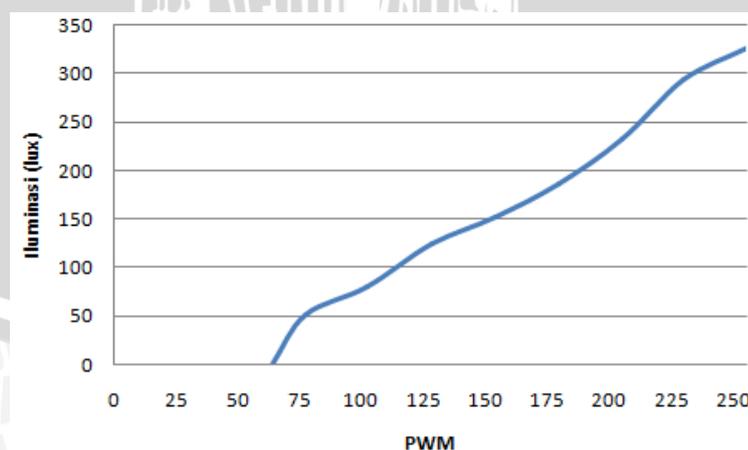
**Gambar 5.11.** Blok Diagram Pengujian Iluminasi Lampu LED 3W

Hasil pengujian iluminasi yang dihasilkan lampu LED 3W dengan alat ukur LDR yang sudah dikalibrasi ditunjukkan oleh Tabel 5.7.

**Tabel 5.7.** Hasil Pengujian Iluminasi Cahaya Lampu Led 3W

PWM	$V_{in}$ (v)	$E_{ldr}$ (lux)
255	11,16	327
230	10,41	294
205	9,43	233
179	8,57	187
154	7,42	152
128	6,8	124
102	6,62	80
77	5,67	52
64	4,52	0

Data Tabel 5.7 menunjukkan bahwa lampu led 3W dapat menyala jika nilai PWM lebih dari 128. Iluminasi yang dihasilkan lampu LED saat tegangan penuh (PWM 255) dengan jarak 30 cm terhadap alat ukur adalah 327 lux. Grafik nilai intensitas cahaya yang dihasilkan lampu LED 3W dengan variasi nilai ditunjukkan oleh Gambar 5.12.



**Gambar 5.12.** Grafik Nilai Iluminasi Cahaya LED 3W Terhadap PWM

Dari grafik pada Gambar 5.12 dapat dilihat bahwa nilai iluminasi berbanding lurus dengan nilai PWM, dan lampu LED 3W baru dapat menyala saat diberi nilai PWM 77 yaitu catu tegangan lebih dari 5,67V.

### 5.7. Pengujian Keseluruhan Sistem

Pengujian keseluruhan bertujuan untuk mengetahui kinerja keseluruhan sistem yang telah dirancang sebelumnya dengan menggunakan beberapa pengujian. Pengujian dilakukan saat ruangan berada dalam keadaan kosong dan dimasuki tikus, dengan suhu dan iluminasi di dalam ruangan bervariasi. Data hasil pengujian ditunjukkan oleh Tabel 5.8.

**Tabel 5.8.** Pengujian Keseluruhan

PIR	$^{\circ}\text{C}$	Lux	Peltier	PWM Kipas	PWM LED
0	29	64	0	0	0
1	29	64	Pendingin	230	128
1	21	64	Pemanas	128	128

VI

Dari Tabel 5.8 dapat dilihat bahwa sistem tidak berjalan saat sensor PIR tidak aktif (ruangan kosong) walaupun suhu dan iluminasi di dalam ruangan tidak nyaman. Hal ini sudah sesuai dengan prinsip kerja sistem, yaitu sistem aktif hanya jika PIR aktif (ruangan tidak kosong). Saat dimasukkan seekor tikus, PIR segera aktif dan sistem berjalan.

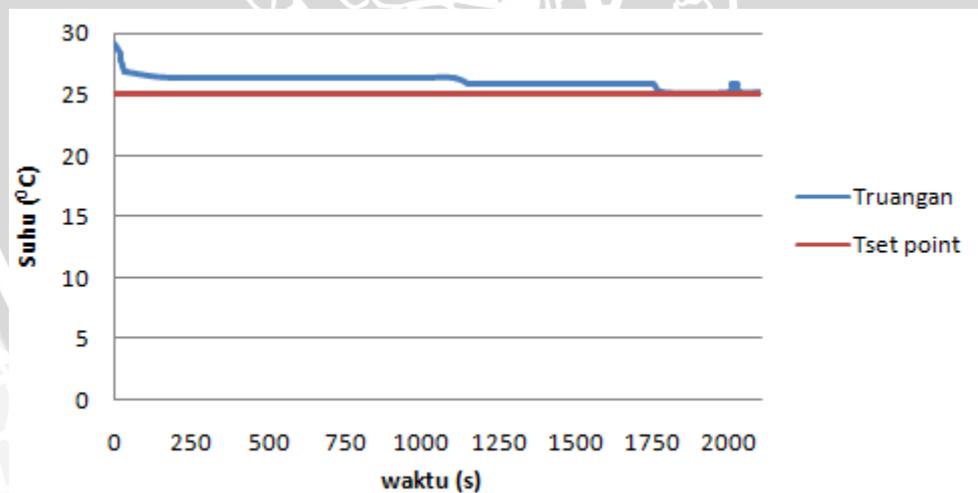
#### 5.7.1. Pengujian Respon Suhu

Pengujian bertujuan untuk melihat respon perubahan suhu terhadap waktu dengan kondisi awal yang sudah ditentukan. Saat suhu sudah mencapai *set point* dan *steady* diberikan gangguan berupa pembukaan pintu dan penambahan jumlah tikus Hasil pengujian saat suhu ruangan  $29^{\circ}\text{C}$  dan ditunjukkan oleh Tabel 5.9.

**Tabel 5.9.** Tabel Pengujian saat Suhu 29<sup>o</sup>C

Waktu (s)	Suhu (°C)	Waktu (s)	Suhu (°C)
0	29,23	1172	25,83
16	28,38	1174	25,83
20	27,82	1180	25,83
24	27,32	1182	25,83
30	26,83	1190	25,83
37	26,83	1218	25,83
195	26,37	1350	25,83
712	26,37	1450	25,83
800	26,37	1550	25,83
900	26,37	1650	25,83
1000	26,37	1750	25,83
1100	26,37	1772	25,23
1150	25,83	2000	25,23
1156	25,83	2005	25,83
1162	25,83	2020	25,83
1166	25,83	2030	25,23
1168	25,83	2100	25,23

Dari Tabel 5.9 dapat disajikan dalam bentuk grafik respon suhu ruangan terhadap waktu pada Gambar 5.13.

**Gambar 5.13.** Respon Suhu Ruangan Dengan Kondisi Awal 29<sup>o</sup>C

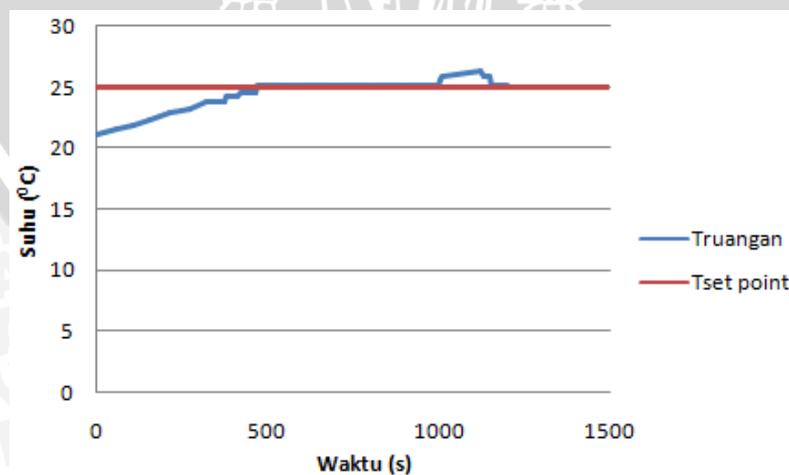
Dari Gambar 5.13 dapat dilihat bahwa respon penurunan suhu terjadi lebih cepat saat kondisi awal menuju suhu 26,37<sup>o</sup>C dengan waktu 195 detik. Suhu ruangan kemudian turun menjadi 25,83<sup>o</sup>C pada detik ke-1150. Suhu mencapai *set point* dan *steady* pada suhu 25,23<sup>o</sup>C pada detik ke-1772. Pada detik ke-2000

diberikan gangguan dengan membuka ruangan dan menambahkan seekor tikus. Gangguan membuat suhu ruangan naik menjadi  $25,83^{\circ}\text{C}$  dan kembali mencapai *set point* 30 detik kemudian. Hasil pengujian saat suhu ruangan  $21^{\circ}\text{C}$  ditunjukkan oleh Tabel 5.10.

**Tabel 5.10.** Tabel Pengujian Saat Suhu  $21^{\circ}\text{C}$

Waktu (s)	Suhu ( $^{\circ}\text{C}$ )	Waktu (s)	Suhu ( $^{\circ}\text{C}$ )
0	21,12	440	24,58
56	21,54	450	24,58
105	21,92	465	24,58
155	22,37	470	25,14
212	22,83	550	25,14
269	23,25	570	25,14
320	23,8	600	25,14
330	23,8	700	25,14
340	23,8	800	25,14
350	23,8	900	25,14
360	23,8	1000	25,14
370	23,8	1002	25,58
378	24,22	1010	25,88
390	24,22	1120	26,32
400	24,22	1130	25,88
410	24,22	1150	25,88
418	24,58	1155	25,14
430	24,58	1500	25,14

Dari Tabel 5.10 dapat disajikan dalam bentuk grafik respon suhu terhadap waktu pada Gambar 5.14.



**Gambar 5.14.** Respon Suhu Ruangan Dengan Kondisi Awal  $21^{\circ}\text{C}$

Dari Gambar 5.16 dapat dilihat bahwa respon kenaikan suhu terjadi lebih cepat dibandingkan dengan respon penurunan suhu pada Gambar 5.14. Waktu yang dibutuhkan untuk mencapai *set point* dari 21,12<sup>o</sup>C menjadi 25,14<sup>o</sup>C adalah 378 detik. Pada detik ke-1000 diberikan gangguan dengan membuka ruangan dan menambahkan seekor tikus. Gangguan membuat suhu ruangan naik menjadi 26,32<sup>o</sup>C dan kembali mencapai *set point* 155 detik kemudian.

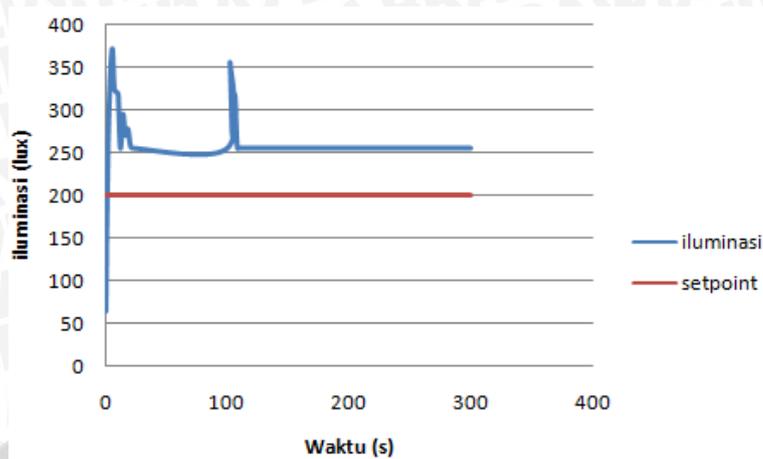
### 5.7.2. Pengujian Respon Iluminasi

Pengujian bertujuan untuk melihat respon perubahan iluminasi terhadap waktu dengan kondisi awal yang sudah ditentukan. Saat iluminasi sudah mencapai *set point* dan *steady* diberikan gangguan berupa penambahan cahaya dari luar ruangan. Hasil pengujian saat iluminasi 64 lux ditunjukkan oleh Tabel 5.11.

**Tabel 5.11.** Pengujian saat Iluminasi 64 lux

Waktu (s)	Iluminasi (lux)
0	64
2	269
5	372
7	325
10	319
12	256
14	296
16	271
18	279
20	260
22	256
100	256
102	357
104	271
106	319
108	256
200	256
300	256

Dari Tabel 5.11 dapat disajikan dalam bentuk grafik respon iluminasi terhadap waktu pada Gambar 5.15.



**Gambar 5.15.** Respon Iluminasi Ruang Dengan Kondisi Awal 64 Lux

Gambar 5.15 memperlihatkan bahwa nilai iluminasi mencapai nilai *set point* dan *steady* setelah 22 detik dengan nilai 256 lux. Pada detik ke 100 diberi gangguan berupa penambahan cahaya menjadi 300 lux. Iluminasi mulai *steady* kembali 8 detik kemudian.

### 5.7.3. Analisis

Dari hasil pengujian keseluruhan pada pengujian respon suhu dapat dilihat bahwa menurunkan suhu membutuhkan waktu yang lebih lama dibandingkan menaikkan suhu. Hal ini dikarenakan kemampuan peltier yang bekerja sebagai pendingin lebih lambat dibandingkan saat digunakan sebagai pemanas. Pada pengujian respon iluminasi dapat dilihat bahwa iluminasi *steady* pada nilai 256 lux. Waktu yang dibutuhkan untuk mencapai *steady* adalah 22 detik. Nilai *steady* yang cukup besar dari nilai *setpoint* dikarenakan kemampuan alat yang kurang mampu mengatasi iluminasi berlebih. Akan tetapi berdasarkan Tabel 2.2 tentang kenyamanan penerangan, nilai 256 masih termasuk kategori nyaman, sehingga tujuan dari penelitian ini sudah tercapai.

## BAB VI PENUTUP

### 6.1. Kesimpulan

Dari hasil perancangan dan pengujian yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut :

1. Sistem mulai bekerja setelah sensor PIR aktif atau terdeteksi keberadaan manusia. Sensor PIR harus diletakkan maksimum 40 centimeter dari tinggi manusia yang memasuki ruangan.
2. Dengan menggunakan kontrol logika fuzzy untuk mencapai *set point* 25<sup>0</sup>C suhu dari kondisi awal 29<sup>0</sup>C membutuhkan waktu 1772 detik sedangkan dari kondisi awal 21<sup>0</sup>C membutuhkan waktu 378 detik. Untuk mencapai *steady* iluminasi membutuhkan waktu 22 detik.

### 6.2. Saran

Beberapa hal yang direkomendasikan untuk pengembangan lebih lanjut adalah :

1. Penggunaan bahan prototipe yang lebih tahan dari pengaruh suhu dan iluminasi dari luar.
2. Penggunaan peltier lebih dari satu untuk mempercepat proses pendinginan.
3. Penambahan aktuator yang dapat mencegah terjadinya *glare* (silau), yaitu tingkat penerangan melebihi batas pada sistem penerangan di dalam ruangan.

## DAFTAR PUSTAKA

- Anonim. 2007. *Datasheet ATMEGA-16*. Atmel.
- Anonim. 2012. *Datasheet Passsive Infra Red*. Parallax.
- Anonim. 1997. *Datasheet Light Dependent Resistor*. Rs Component.
- Anonim. 2001. *Datasheet SHT1x*. Sensirion.
- Anonim. 2002. *Datasheet L293*. Texas Instrument
- Frick, Heinz.,dkk. 2008. *Ilmu Fisika Bangunan*. Yogyakarta : Kanisius.
- Gilbert, Held. 2009.*Light Emitting Diode Technology and Aplications*. London : CRC Press.
- Jouaneh, M. 2013. *Fundamentals of Mechantronics* . New York : Cengage Learning.
- Kuswadi, Son. 2000. *Kendali Cerdas*. Surabaya : EEPIS.
- Ogata, Katsuhiko. 2010. *Modern Control Engineering5th edition*. New Jersey : Prentice Hall.
- Platt, Charles. 2012.*Make: Encyclopedia of Electronic Components Vol.1*. New York : Maker Media.
- Sears, F., Zemansky. 1994. *Fisika untuk universitas 1 : Mekanika, Panas, Bunyi*. Bandung : Binacipta.
- Soegijanto.1994. *Bangunan di Indonesia dengan Iklim tropis Lembab Ditinjau dari Aspek Fisika Bangunan*. Bandung : Dirjen Dikti.
- Yan,J.,dkk. 1993. *Using Fuzzy Logic*. NewYork: Prentice-Hall.

UNIVERSITAS BRAWIJAYA

LAMPIRAN



**LAMPIRAN 1**  
**DOKUMENTASI ALAT**



**Gambar 9.** Foto Alat Tampak Atas



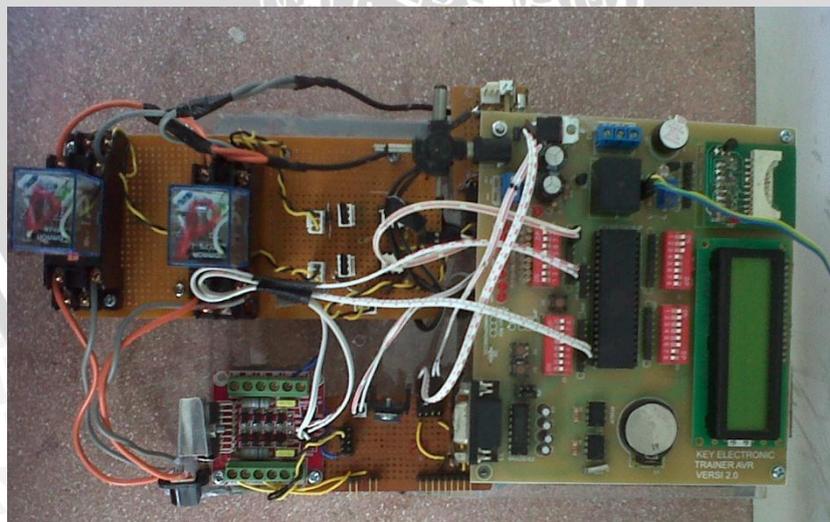
**Gambar 10.** Foto Alat Tampak Belakang



Gambar 11. Pengujian Kipas



Gambar 4. Pengujian Lampu LED



Gambar 5. Rangkaian Keseluruhan



**LAMPIRAN 2****LISTING PROGRAM MENGGUNAKAN BASIC COMPILER AVR**

```
$regfile = "m16def.dat"
```

```
$crystal = 11059200
```

```
$baud = 9600
```

```
Config Lcdpin = Pin , Db4 = Portc.5 , Db5 = Portc.4 , Db6 = Portc.3 , Db7 = Portc.2 , E = Portc.6 , Rs = Portc.7
```

```
Config Lcd = 16 * 2
```

```
Deflcdchar 1 , 28 , 20 , 28 , 32 , 32 , 32 , 32 , 32
```

```
Config Pind.0 = Output
```

```
Config Pind.1 = Output
```

```
Config Pind.2 = Output
```

```
Config Pind.3 = Output
```

```
Config Pina.1 = Input
```

```
Config Adc = Single , Prescaler = Auto , Reference = Avcc
```

```
Config Timer1 = Pwm , Prescale = 64 , Pwm = 8 , Compare A Pwm = Clear Down , Compare B Pwm = Clear Down
```

```
Enable Interrupts
```

```
Enable Timer1
```

```
'FUZZY suhu
```

```
Declare Function Fuze1(byref Evalue1 As Integer ) As Single
```

```
Declare Function Fuzce1(byref Devalue1 As Integer ) As Single
```

```
Declare Sub Inference1(byref Evalue1 As Integer , Byref Devalue1 As Integer )
```

```
Declare Function Minimum(byref Value1 As Single , Byref Value2 As Single) As Single
```

```
Declare Function Defuzz1()as Byte
```

```
'FUZZY lux
```

```
Declare Function Fuze2(byref Evalue2 As Integer ) As Single
```

```
Declare Function Fuzce2(byref Devalue2 As Integer ) As Single
```

```
Declare Sub Inference2(byref Evalue2 As Integer , Byref Devalue2 As Integer )
```

```
Declare Function Defuzz2()as Byte
```

```
'variable
```

```
Dim Lowval As Integer , Midval As Integer , Highval As Integer , Highmid As Integer , Lowmid As Integer
```

```
Dim Evalue1 As Integer , Devalue1 As Integer , Evalue_1 As Integer
```

```
Dim Evalue2 As Integer , Devalue2 As Integer , Evalue_2 As Integer
```

```
Dim Numinator As Single , Denominator As Single
```

```
Dim Numinator2 As Single , Denominator2 As Single
```

```
Dim Index As Byte , Kipas As Byte , Led As Byte
```

```
Dim Antecedent1(25) As Single , Antecedent2(25) As Single
```

```

Dim Valerror1 As Single , Valdeltal As Single , Temp1 As Single ,
Valconsequent1 As Byte , Tempstr1 As String * 3
Dim Valerror2 As Single , Valdeltal2 As Single , Temp2 As Single ,
Valconsequent2 As Byte , Tempstr2 As String * 3
Dim Sp1 As Word , Sp2 As Word
'PIR & LDR
  Dim Ldr_adc As Word , Ldr_convert As Single
'SHT11
Dim Dataword As Word
Dim Command As Byte
Dim Calc As Single
Dim Calc2 As Single
Dim Rhlinear As Single
Dim Rhlintemp As Single
Dim Tempc As Single
Dim Ctr As Byte
Dim X As Word
Dim Y As Single
Dim Z As Single
Dim Suhu As String * 10
Dim Kelembaban As String * 10
  Const C1 = -4
  Const C2 = 0.0405
  Const C3 = -0.0000028
  Const S = .01
  Const H = .00008
Sck Alias Portc.0 'sck output pada port C.0
Dataout Alias Portc.1 'ketika memanggil dataout portc.1
menjadi output
Datain Alias Pinc.1
Declare Sub Kirim()
Ddrc = &B11111111
'port c sebagai output
Config Pinc.0 = Output
Config Pinc.1 = Output
Wait 1
Set Dataout
For Ctr = 1 To 12
Set Sck
Waitus 2
Reset Sck
Waitus 2
Next Ctr 'ketika memanggil datain portc.1
menjadi input

Start Adc
Evalue1 = 0
Devalue1 = 0

```

```

Value_1 = 0
Value2 = 0
Devalue2 = 0
Value_2 = 0

Do

Ldr_adc = Getadc(0)
Ldr_convert = Ldr_adc / 1024
Ldr_convert = Ldr_convert * 5
Ldr_convert = Ldr_convert * 204.5
Ldr_convert = Ldr_convert - 106.5
If Ldr_convert <= 0 Then
Ldr_convert = 0
Else
Ldr_convert = Round(ldr_convert)
End If

```

```
Gosub Data_suhu
```

```
Cls
```

```
If Pina.1 > 0 Then
```

```
Locate 1, 1
```

```
Lcd "on"
```

```
Reset Pina.1
```

```
Else
```

```
Lcd "off"
```

```
End If
```

```
Locate 1, 5
```

```
Lcd Suhu ; "" ; Chr(1) ; " "
```

```
Lowerline
```

```
Lcd Ldr_convert ; "lx " ; Pwm1a ; " " ; Pwm1b ; " "
```

```
Sp1 = 25
```

```
Value1 = Sp1 - Tempc
```

```
Devalue1 = Value1 - Value_1
```

```
Value_1 = Value1
```

```
Sp2 = 200
```

```
Value2 = Sp2 - Ldr_convert
```

```
Devalue2 = Value2 - Value_2
```

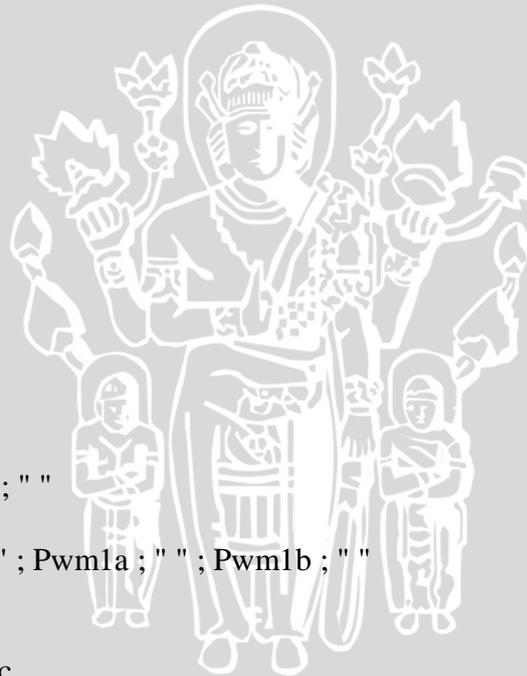
```
Value_2 = Value2
```

```
Numinator = 0
```

```
Denominator = 0
```

```
Numinator2 = 0
```

```
Denominator2 = 0
```



Inference1 Evalue1 , Devalue1  
 Inference2 Evalue2 , Devalue2

Kipas = Defuzz1()

Pwm1a = Kipas

Led = Defuzz2()

Pwm1b = Led

Loop

'SHT11

Data\_suhu:

Command = &B00000011

Call Kirim

Tempc = S \* Dataword

Tempc = Tempc - 40

Suhu = Fusing(tempc , "#.#")

Command = &B00000101

Call Kirim

Calc = C2 \* Dataword

Calc2 = Dataword \* Dataword

Calc2 = C3 \* Calc2

Calc = Calc + C1

Rhlinear = Calc + Calc2

Calc = H \* Dataword

Calc = Calc + S

Calc2 = Tempc - 25

Calc = Calc2 \* Calc

Rhlintemp = Calc + Rhlinear

Kelembaban = Fusing(rhlintemp , "#.#")

Return

//Panggil Kirim

Sub Kirim()

Local Datavalue As Word

Local Databyte As Byte

Set Sck

Reset Dataout

Reset Sck

Set Sck

Set Dataout

Reset Sck

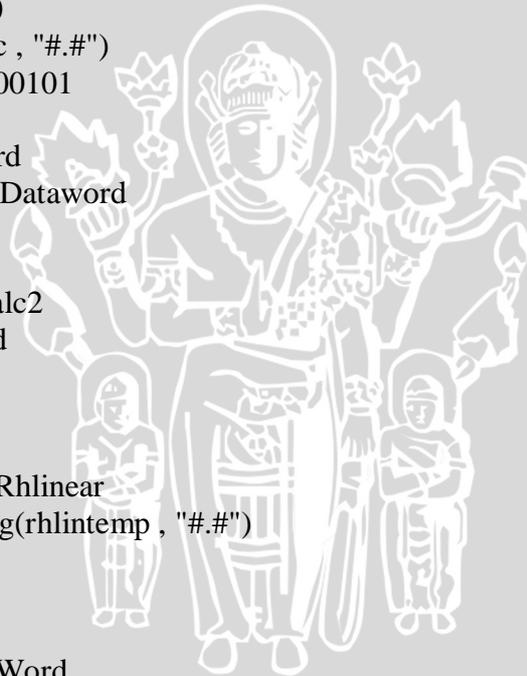
Shiftout Dataout , Sck , Command , 1

Ddrc = &B11111101

Config Pinc.1 = Input

Set Sck

Reset Sck



```

Waitus 10
Bitwait Pinc.1 , Reset
Shiftin Datin , Sck , Databyte , 1
Datavalue = Databyte
Ddrc = &B11111111
Config Pinc.1 = Output
Reset Dataout
Set Sck
Reset Sck
Ddrc = &B11111101
Config Pinc.1 = Input
Shiftin Datin , Sck , Databyte , 1
Shift Datavalue , Left , 8
Datavalue = Datavalue Or Databyte
Dataword = Datavalue
Ddrc = &B11111111
Config Pinc.1 = Output
Reset Dataout
Set Sck
Reset Sck
Ddrc = &B11111101
Config Pinc.1 = Input
Shiftin Datin , Sck , Databyte , 1
Ddrc = &B11111111
Config Pinc.1 = Output
Set Dataout
Set Sck
Reset Sck
End Sub

' fuzzy
Function Defuzz1()as Byte
For Index = 1 To 25
Valconsequent1 = Lookup(index , Consequent1)
Temp1 = Antecedent1(index) * Valconsequent1
Numerator = Numerator + Temp1
Denominator = Denominator + Antecedent1(index)
Next
Temp1 = Numerator / Denominator
Tempstr1 = Fusing(temp1 , "###.#")
Tempstr1 = Left(tempstr1 , 3)
Defuzz1 = Val(tempstr1)
End Function

Function Defuzz2()as Byte
For Index = 1 To 25
Valconsequent2 = Lookup(index , Consequent2)
Temp2 = Antecedent2(index) * Valconsequent2

```

```

Numinator2 = Numinator2 + Temp2
Denominator2 = Denominator2 + Antecedent2(index)
Next
Temp2 = Numinator2 / Denominator2
Tempstr2 = Fusing(temp2 , "###.#")
Tempstr2 = Left(tempstr2 , 3)
Defuzz2 = Val(tempstr2)
End Function

```

```

Function Minimum(byref Value1 As Single , Byref Value2 As Single) As Single
If Value1 <= Value2 Then
Minimum = Value1
Else
Minimum = Value2
End If
End Function

```

```

Sub Inference1(byref Evalue1 As Integer , Byref Devalue1 As Integer)
'kolom1
Index = 1
Index = Lookup(index , Uindex)
Valdelta1 = Fuzce1(devalue1)
Index = 1
Index = Lookup(index , Uindex)
Valerror1 = Fuze1(evalue1)
Antecedent1(1) = Minimum(valdelta1 , Valerror1)
Index = 2
Index = Lookup(index , Uindex)
Valerror1 = Fuze1(evalue1)
Antecedent1(2) = Minimum(valdelta1 , Valerror1)
Index = 3
Index = Lookup(index , Uindex)
Valerror1 = Fuze1(evalue1)
Antecedent1(3) = Minimum(valdelta1 , Valerror1)
Index = 4
Index = Lookup(index , Uindex)
Valerror1 = Fuze1(evalue1)
Antecedent1(4) = Minimum(valdelta1 , Valerror1)
Index = 5
Index = Lookup(index , Uindex)
Valerror1 = Fuze1(evalue1)
Antecedent1(5) = Minimum(valdelta1 , Valerror1)
'kolom2
Index = 2
Index = Lookup(index , Uindex)
Valdelta1 = Fuzce1(devalue1)
Index = 1
Index = Lookup(index , Uindex)

```

Valerror1 = Fuze1(evaluate1)  
 Antecedent1(6) = Minimum(valdelta1 , Valerror1)  
 Index = 2  
 Index = Lookup(index , Uindex)  
 Valerror1 = Fuze1(evaluate1)  
 Antecedent1(7) = Minimum(valdelta1 , Valerror1)  
 Index = 3  
 Index = Lookup(index , Uindex)  
 Valerror1 = Fuze1(evaluate1)  
 Antecedent1(8) = Minimum(valdelta1 , Valerror1)  
 Index = 4  
 Index = Lookup(index , Uindex)  
 Valerror1 = Fuze1(evaluate1)  
 Antecedent1(9) = Minimum(valdelta1 , Valerror1)  
 Index = 5  
 Index = Lookup(index , Uindex)  
 Valerror1 = Fuze1(evaluate1)  
 Antecedent1(10) = Minimum(valdelta1 , Valerror1)  
 'kolom3  
 Index = 3  
 Index = Lookup(index , Uindex)  
 Valdelta1 = Fuzce1(devalue1)  
 Index = 1  
 Index = Lookup(index , Uindex)  
 Valerror1 = Fuze1(evaluate1)  
 Antecedent1(11) = Minimum(valdelta1 , Valerror1)  
 Index = 2  
 Index = Lookup(index , Uindex)  
 Valerror1 = Fuze1(evaluate1)  
 Antecedent1(12) = Minimum(valdelta1 , Valerror1)  
 Index = 3  
 Index = Lookup(index , Uindex)  
 Valerror1 = Fuze1(evaluate1)  
 Antecedent1(13) = Minimum(valdelta1 , Valerror1)  
 Index = 4  
 Index = Lookup(index , Uindex)  
 Valerror1 = Fuze1(evaluate1)  
 Antecedent1(14) = Minimum(valdelta1 , Valerror1)  
 Index = 5  
 Index = Lookup(index , Uindex)  
 Valerror1 = Fuze1(evaluate1)  
 Antecedent1(15) = Minimum(valdelta1 , Valerror1)  
 'kolom4  
 Index = 4  
 Index = Lookup(index , Uindex)  
 Valdelta1 = Fuzce1(devalue1)  
 Index = 1  
 Index = Lookup(index , Uindex)

```

Valerror1 = Fuze1(evalue1)
Antecedent1(16) = Minimum(valdelta1 , Valerror1)
Index = 2
Index = Lookup(index , Uindex)
Valerror1 = Fuze1(evalue1)
Antecedent1(17) = Minimum(valdelta1 , Valerror1)
Index = 3
Index = Lookup(index , Uindex)
Valerror1 = Fuze1(evalue1)
Antecedent1(18) = Minimum(valdelta1 , Valerror1)
Index = 4
Index = Lookup(index , Uindex)
Valerror1 = Fuze1(evalue1)
Antecedent1(19) = Minimum(valdelta1 , Valerror1)
Index = 5
Index = Lookup(index , Uindex)
Valerror1 = Fuze1(evalue1)
Antecedent1(20) = Minimum(valdelta1 , Valerror1)
'kolom5
Index = 5
Index = Lookup(index , Uindex)
Valdelta1 = Fuzce1(devalue1)
Index = 1
Index = Lookup(index , Uindex)
Valerror1 = Fuze1(evalue1)
Antecedent1(21) = Minimum(valdelta1 , Valerror1)
Index = 2
Index = Lookup(index , Uindex)
Valerror1 = Fuze1(evalue1)
Antecedent1(22) = Minimum(valdelta1 , Valerror1)
Index = 3
Index = Lookup(index , Uindex)
Valerror1 = Fuze1(evalue1)
Antecedent1(23) = Minimum(valdelta1 , Valerror1)
Index = 4
Index = Lookup(index , Uindex)
Valerror1 = Fuze1(evalue1)
Antecedent1(24) = Minimum(valdelta1 , Valerror1)
Index = 5
Index = Lookup(index , Uindex)
Valerror1 = Fuze1(evalue1)
Antecedent1(25) = Minimum(valdelta1 , Valerror1)

End Sub
Sub Inference2(byref Evalue2 As Integer , Byref Devalue2 As Integer)
'kolom1
Index = 1
Index = Lookup(index , Uindex)

```

Valdelta2 = Fuzce2(devalue2)  
 Index = 1  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(1) = Minimum(valdelta2 , Valerror2)  
 Index = 2  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(2) = Minimum(valdelta2 , Valerror2)  
 Index = 3  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(3) = Minimum(valdelta2 , Valerror2)  
 Index = 4  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(4) = Minimum(valdelta2 , Valerror2)  
 Index = 5  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(5) = Minimum(valdelta2 , Valerror2)  
 'kolom2  
 Index = 2  
 Index = Lookup(index , Uindex)  
 Valdelta2 = Fuzce2(devalue2)  
 Index = 1  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(6) = Minimum(valdelta2 , Valerror2)  
 Index = 2  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(7) = Minimum(valdelta2 , Valerror2)  
 Index = 3  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(8) = Minimum(valdelta2 , Valerror2)  
 Index = 4  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(9) = Minimum(valdelta2 , Valerror2)  
 Index = 5  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(10) = Minimum(valdelta2 , Valerror2)  
 'baris3  
 Index = 3  
 Index = Lookup(index , Uindex)

Valdelta2 = Fuzce2(devalue2)  
 Index = 1  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(11) = Minimum(valdelta2 , Valerror2)  
 Index = 2  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(12) = Minimum(valdelta2 , Valerror2)  
 Index = 3  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(13) = Minimum(valdelta2 , Valerror2)  
 Index = 4  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(14) = Minimum(valdelta2 , Valerror2)  
 Index = 5  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(15) = Minimum(valdelta2 , Valerror2)  
 'baris4  
 Index = 4  
 Index = Lookup(index , Uindex)  
 Valdelta2 = Fuzce2(devalue2)  
 Index = 1  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(16) = Minimum(valdelta2 , Valerror2)  
 Index = 2  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(17) = Minimum(valdelta2 , Valerror2)  
 Index = 3  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(18) = Minimum(valdelta2 , Valerror2)  
 Index = 4  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(19) = Minimum(valdelta2 , Valerror2)  
 Index = 5  
 Index = Lookup(index , Uindex)  
 Valerror2 = Fuze2(evalue2)  
 Antecedent2(20) = Minimum(valdelta2 , Valerror2)  
 'kolom5  
 Index = 5  
 Index = Lookup(index , Uindex)

```

Valdelta2 = Fuzce2(devalue2)
Index = 1
Index = Lookup(index , Uindex)
Valerror2 = Fuze2(evalue2)
Antecedent2(21) = Minimum(valdelta2 , Valerror2)
Index = 2
Index = Lookup(index , Uindex)
Valerror2 = Fuze2(evalue2)
Antecedent2(22) = Minimum(valdelta2 , Valerror2)
Index = 3
Index = Lookup(index , Uindex)
Valerror2 = Fuze2(evalue2)
Antecedent2(23) = Minimum(valdelta2 , Valerror2)
Index = 4
Index = Lookup(index , Uindex)
Valerror2 = Fuze2(evalue2)
Antecedent2(24) = Minimum(valdelta2 , Valerror2)
Index = 5
Index = Lookup(index , Uindex)
Valerror2 = Fuze2(evalue2)
Antecedent2(25) = Minimum(valdelta2 , Valerror2)

```

End Sub

Function Fuze1(byref Evaluel As Integer)as Single

```

Lowval = Lookup(index , Terror1)

```

```

Incr Index

```

```

Midval = Lookup(index , Terror1)

```

```

Incr Index

```

```

Highval = Lookup(index , Terror1)

```

```

Highmid = Highval - Midval

```

```

Lowmid = Midval - Lowval

```

```

If Lowval = Midval And Evaluel <= Lowval Then

```

```

Fuze1 = 1

```

```

Else

```

```

If Highval = Midval And Evaluel >= Highval Then

```

```

Fuze1 = 1

```

```

Else

```

```

If Evaluel = Lowval Or Evaluel >= Highval Then

```

```

Fuze1 = 0

```

```

Else

```

```

If Evaluel = Midval Then

```

```

Fuze1 = 1

```

```

Elseif Evaluel > Midval Then

```

```

Fuze1 = Highval - Evaluel

```

```

Fuze1 = Fuze1 / Highmid

```

```

Else

```

```

Fuze1 = Evalue1 - Lowval
Fuze1 = Fuze1 / Lowmid
End If
End If
End If
End If
End Function

```

```

Function Fuzce1(byref Devalue1 As Integer)as Single

```

```

Lowval = Lookup(index , Derror1)
Incr Index
Midval = Lookup(index , Derror1)
Incr Index
Highval = Lookup(index , Derror1)
Highmid = Highval - Midval
Lowmid = Midval - Lowval

```

```

If Lowval = Midval And Devalue1 <= Lowval Then
Fuze1 = 1
Else
If Highval = Midval And Devalue1 >= Highval Then
Fuze1 = 1
Else
If Evalue1 = Lowval Or Devalue1 >= Highval Then
Fuze1 = 0
Else
If Evalue1 = Midval Then
Fuze1 = 1
Elseif Evalue1 > Midval Then
Fuze1 = Highval - Devalue1
Fuze1 = Fuze1 / Highmid
Else
Fuze1 = Devalue1 - Lowval
Fuze1 = Fuze1 / Lowmid
End If
End If
End If
End If
End Function

```

```

Function Fuze2(byref Evalue2 As Integer)as Single

```

```

Lowval = Lookup(index , Terror2)
Incr Index
Midval = Lookup(index , Terror2)
Incr Index
Highval = Lookup(index , Terror2)
Highmid = Highval - Midval
Lowmid = Midval - Lowval

```

```

If Lowval = Midval And Evalue2 <= Lowval Then
Fuze2 = 1
Else
If Highval = Midval And Evalue2 >= Highval Then
Fuze2 = 1
Else
If Evalue2 = Lowval Or Evalue2 >= Highval Then
Fuze2 = 0
Else
  If Evalue2 = Midval Then
    Fuze2 = 1
  ElseIf Evalue2 > Midval Then
    Fuze2 = Highval - Evalue2
    Fuze2 = Fuze2 / Highmid
  Else
    Fuze2 = Evalue2 - Lowval
    Fuze2 = Fuze2 / Lowmid
  End If
End If
End If
End If
End Function

```

```

Function Fuzce2(byref Devalue2 As Integer)as Single
Lowval = Lookup(index , Derror2)
Incr Index
Midval = Lookup(index , Derror2)
Incr Index
Highval = Lookup(index , Derror2)
Highmid = Highval - Midval
Lowmid = Midval - Lowval

If Lowval = Midval And Devalue2 <= Lowval Then
Fuze2 = 1
Else
If Highval = Midval And Devalue2 >= Highval Then
Fuze2 = 1
Else
If Evalue2 = Lowval Or Devalue2 >= Highval Then
Fuze2 = 0
Else
  If Evalue2 = Midval Then
    Fuze2 = 1
  ElseIf Evalue2 > Midval Then
    Fuze2 = Highval - Devalue2
    Fuze2 = Fuze2 / Highmid
  Else

```

```

Fuzce2 = Devalue2 - Lowval
Fuzce2 = Fuzce2 / Lowmid
End If
End If
End If
End If
End Function
    
```

Consequent1:

```

Data 0 , 230 , 230 , 230 , 128 , 0 , 230 , 230 , 128 , 0 , 128 , 230 , 128 , 0 , 128 ,
230 , 128 , 0 , 128 , 230 , 230 , 0 , 128 , 230 , 230 , 230
' /-----1-----/ /-----2-----/ /-----3-----/ /-----
-----4-----/ /-----5-----/
    
```

Terror1:

```

Data 0% , -5% , -5% , -3% , -5% , -3% , 0% , -3% , 0% , 3% , 0% , 3% , 5% , 3% ,
5% , 5%
    
```

Derror1:

```

Data 0% , -2% , -2% , -1% , -2% , -1% , 0% , 1% , 0% , 1% , 0% , 1% , 2% , 1% ,
2% , 2%
    
```

Consequent2:

```

Data 0 , 205 , 205 , 205 , 179 , 154 , 205 , 205 , 179 , 154 , 102 , 205 , 179 , 154 ,
102 , 64 , 179 , 154 , 102 , 64 , 64 , 154 , 102 , 64 , 64 , 64
' /-----1-----/ /-----2-----/ /-----3-----/ /-----
-----4-----/ /-----5-----/
    
```

Terror2:

```

Data 0% , -100% , -100% , -50% , -100% , -50% , 0 , -50% , 0% , 100% , 0% ,
100% , 200% , 100% , 200% , 200%
    
```

Derror2:

```

Data 0% , -50% , -50% , -25% , -50% , -25% , 0% , -25% , 0% , 100% , 0% ,
100% , 200% , 100% , 200% , 200%
    
```

Uindex:

```

Data 0 , 1 , 4 , 7 , 10 , 13
    
```