

BAB II

LANDASAN TEORI

2.1 IoT (*Internet of Things*)

IoT (*Internet of Thing*) dapat didefinisikan kemampuan berbagai device yang bisa saling terhubung dan saling bertukar data melalui jaringan internet. IoT merupakan sebuah teknologi yang memungkinkan adanya sebuah pengendalian, komunikasi, kerjasama dengan berbagai perangkat keras, data melalui jaringan internet. Sehingga bisa dikatakan bahwa *Internet of Things* (IoT) adalah ketika kita menyambungkan sesuatu (things) yang tidak dioperasikan oleh manusia, ke internet (Hardyanto, 2017).

Namun IOT bukan hanya terkait dengan pengendalia perangkat melalui jarak jauh, tapi juga bagaimana berbagi data, memvirtualisasikan segala hal nyata ke dalam bentuk internet, dan lain-lain. Internet menjadi sebuah penghubung antara sesama mesin secara otomatis. Selain itu juga adanya user yang bertugas sebagai pengatur dan pengawas bekerjanya alat tersebut secara langsung. Manfaatnya menggunakan teknologi IoT yaitu pekerjaan yang dilakukan oleh manusia menjadi lebih cepat, muda dan efisien.

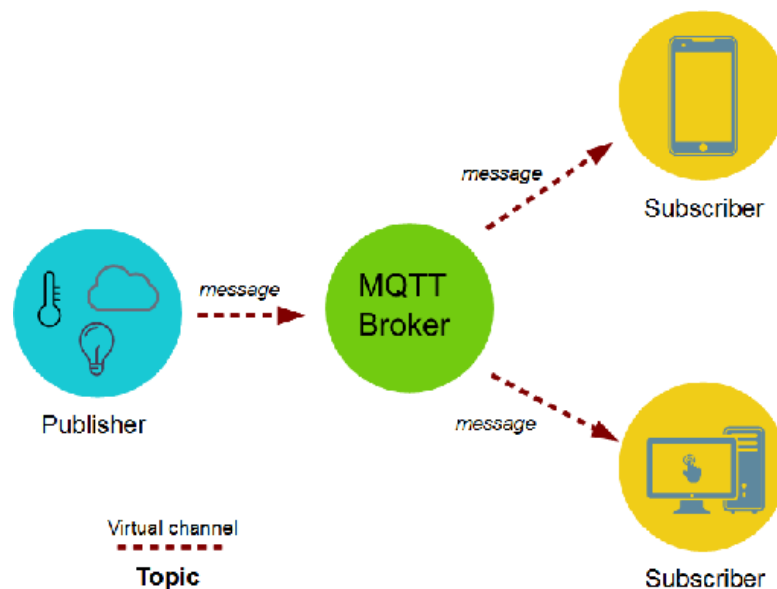
2.2 MQTT

Message Queue Telemetry Transport (MQTT) adalah sebuah protokol komunikasi data *machine to machine* (M2M) yang berada pada layer aplikasi, MQTT bersifat *lightweight message* artinya MQTT berkomunikasi dengan

mengirimkan data pesan yang memiliki header berukuran kecil yaitu hanya sebesar 2 bytes untuk setiap jenis data, sehingga dapat bekerja di dalam lingkungan yang terbatas sumber dayanya seperti kecilnya bandwidth dan terbatasnya sumber daya listrik, selain itu protokol ini juga menjamin terkirimnya semua pesan walaupun koneksi terputus sementara, protokol MQTT menggunakan metode publish/subscribe untuk metode komunikasinya. Publish/subscribe sendiri adalah sebuah pola pertukaran pesan di dalam komunikasi jaringan dimana pengirim data disebut publisher dan penerima data disebut dengan subscriber, metode publish/subscribe memiliki beberapa kelebihan salah satunya yaitu *loose coupling* atau *decouple* dimana berarti antara publisher dan subscriber tidak saling mengetahui keberadaannya, terdapat 3 buah *decoupling* yaitu *time decoupling*, *space decoupling* dan *synchronization decoupling*, *time decoupling* adalah sebuah kondisi dimana publisher dan subscriber tidak harus saling aktif pada waktu yang sama, *space decoupling* adalah dimana publisher dan subscriber aktif di waktu yang sama akan tetapi antara publisher dan subscriber tidak saling mengetahui keberadaan dan identitas satu sama lain, dan yang terakhir adalah *synchronization decoupling* kondisi dimana pengaturan event baik itu penerimaan atau pengiriman pesan di sebuah node hingga tidak saling mengganggu satu sama lain (Rochman et al., 2017).

Pengiriman data pada MQTT didasari oleh topik, topik ini nantinya yang akan menentukan pesan dari publisher harus dikirim pada subscriber yang mana, topik ini dapat bersifat hirarki, MQTT topic memiliki tipe data string dan untuk perbedaan hirarki atau level dari topik digunakan tanda baca “/” . MQTT memiliki

3 level *quality of service* (QOS) dalam pengiriman pesannya yaitu 0,1,2. Untuk lebih jelas tentang prinsip kerja dari MQTT dapat dilihat prinsip dari sensor ultrasonik pada gambar 2.1.



Gambar 2.1 Ilustrasi kerja dari protokol MQTT

2.3 Bahasa Pemrograman C

Menurut Wirdasari (Vol.8:2010) Akar dari bahasa C adalah dari bahasa BCPL yang dikembangkan oleh Martin Richards pada tahun 1967. Bahasa C adalah bahasa yang standar, artinya suatu program ditulis dengan versi bahasa C tertentu akan dapat dikompilasi dengan versi bahasa C yang lain dengan sedikit modifikasi.

Beberapa alasan mengapa bahasa C banyak digunakan, diantaranya adalah sebagai berikut :

1. Bahasa C tersedia hampir di semua jenis komputer.

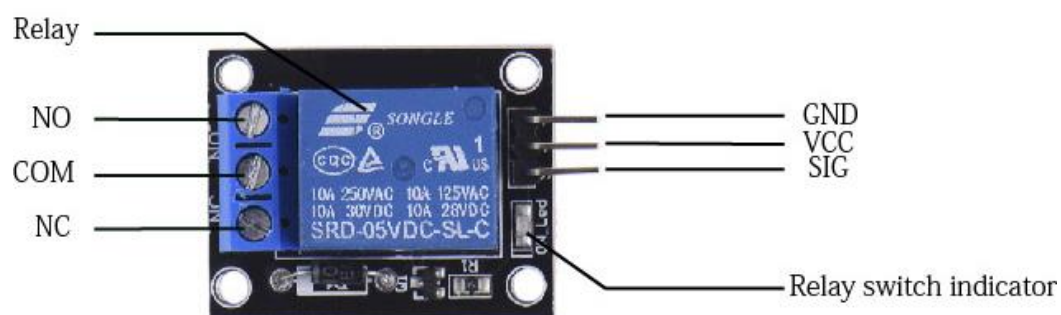
2. Kode bahasa C sifatnya adalah portable Aplikasi yang ditulis dengan bahasa C untuk suatu komputer tertentu dapat digunakan di komputer lain hanya dengan sedikit modifikasi.
3. Bahasa C hanya menyediakan sedikit kata-kata kunci.
4. Proses executable program bahasa C lebih cepat.
5. Dukungan pustaka yang banyak Keandalan bahasa C dicapai dengan adanya fungsi-fungsi pustaka.
6. Bahasa C adalah bahasa yang terstruktur. Bahasa C mempunyai struktur yang baik sehingga mudah untuk dipahami. C mempunyai fungsi-fungsi sebagai program bagiannya.
7. Selain bahasa tingkat tinggi, C juga dianggap sebagai bahasa tingkat menengah. Bahasa C mampu menggabungkan kemampuan bahasa tingkat tinggi dengan bahasa tingkat rendah.
8. Bahasa C adalah compiler Karena C sifatnya adalah compiler, maka akan menghasilkan executable program yang banyak dibutuhkan oleh program-program komersial.

2.4 Modul Relay

Relay adalah komponen elektronik berupa saklar elektronik yang digerakkan oleh arus listrik. Gambar relay ditunjukkan pada gambar 2.2. Secara prinsip, *relay* merupakan tuas saklar dengan lilitan kawat pada batang besi (*solenoid*) di dekatnya ditunjukkan pada gambar 2.3, ketika solenoid dialiri arus listrik, tuas akan tertarik karena adanya gaya magnet yang terjadi pada solenoid sehingga kontak saklar akan menutup. Pada saat arus dihentikan, gaya magnet akan hilang, tuas akan kembali

keposisi semula dan kontak saklar kembali terbuka. *Relay* biasanya digunakan untuk menggerakkan arus/tegangan yang besar (misalnya peralatan listrik 4 A/AC 220V) dengan memakai arus/tegangan yang kecil (misalnya 0.1 A/12 volt DC).

Relay adalah komponen listrik yang bekerja berdasarkan prinsip induksi medan elektromagnetis. Jika sebuah penghantar dialiri oleh arus listrik, maka disekitar penghantar tersebut timbul medan magnet. Medan magnet yang dihasilkan oleh arus listrik tersebut selanjutnya diinduksikan ke logam ferromagnetis. Penemu *relay* pertama kali adalah Joseph Henry pada tahun 1835 (Elangskrafti,2013).



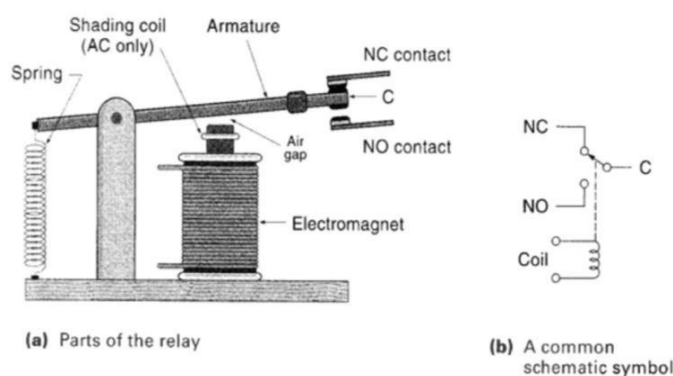
Gambar 2.2 Modul Relay

Prinsip kerja sama dengan kontraktor magnet yaitu sama-sama berdasarkan kemagnetan yang dihasilkan oleh kumparan *coil*, jika kumparan *coil* tersebut diberi sumber listrik. Berdasarkan sumber listrik yang masuk maka *relay* dibagi menjadi 2 macam yaitu *relay* DC dan *relay* AC, besar tegangan DC yang masuk pada *coil relay* bervariasi sesuai dengan ukuran yang tertera pada body *relay* tersebut diantaranya *relay* dengan tegangan 5 Volt, 12 Volt, 24 Volt, 48 Volt, sedangkan untuk tegangan AC sebesar 220 Volt.

Relay terdiri dari *coil* dan *contact*, *coil* adalah gulungan kawat yang mendapat arus listrik, sedangkan *contact* adalah sejenis saklar yang pergerakannya

tergantung dari ada tidaknya arus listrik di *coil*. *Contact* ada 2 jenis : *Normally Open* (kondisi awal sebelum diaktifkan *open*), dan *Normally Closed* (kondisi awal sebelum diaktifkan *close*).

Secara sederhana berikut ini prinsip kerja dari *relay* : ketika *coil* mendapat listrik (*energized*), akan timbul gaya elektromagnet yang akan menarik *armature* yang berpegas, dan *contact* akan menutup.



Gambar 2.3 Prinsip Kerja Modul Relay

Adapun spesifikasi dari module relay 2 channel, sebagai berikut :

- Menggunakan tegangan rendah, 5V, sehingga dapat langsung dihubungkan pada sistem mikrokontroler.
- Tipe relay adalah SPDT (Single Pole Double Throw): 1 COMMON, 1 NC (Normally Close), dan 1 NO (Normally Open).
- Memiliki daya tahan sampai dengan 10A.
- Pin pengendali dapat dihubungkan dengan port mikrokontroler mana saja, sehingga membuat pemrogram dapat leluasa menentukan pin mikrokontroler yang digunakan sebagai pengendali.
- Dilengkapi rangkaian penggerak (*driver*) relay dengan level tegangan TTL sehingga dapat langsung dikendalikan oleh mikrokontroler.

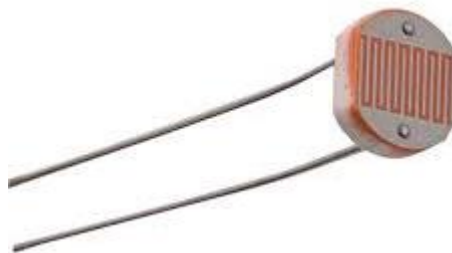
- Driver bertipe “*active high*” atau kumparan relay akan aktif saat pin pengendali diberi logika “1”.
- Driver dilengkapi rangkaian peredam GGL induksi sehingga tidak akan membuat reset sistem mikrokontroler.

Connection:

1. VCC connect to 5 V
2. GND connect to GND
3. 1N1-1N2 relay control interface connected MCU’s IO port.

2.5 Sensor LDR

LDR (*Light Dependent Resistor*) yang ditunjukkan pada gambar 2.4 merupakan salah satu komponen resistor yang nilai resistansinya akan berubah-ubah sesuai dengan intensitas cahaya yang mengenai sensor ini. LDR juga dapat digunakan sebagai sensor cahaya. Perlu diketahui bahwa nilai resistansi dari sensor ini sangat bergantung pada intensitas cahaya. Semakin banyak cahaya yang mengenainya, maka akan semakin menurun nilai resistansinya. Sebaliknya jika semakin sedikit cahaya yang mengenai sensor (gelap), maka nilai hambatannya akan menjadi semakin besar sehingga arus listrik yang mengalir akan terhambat.



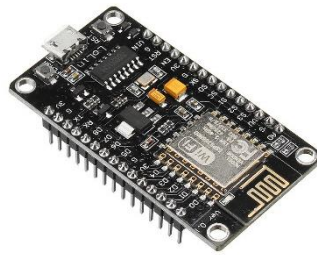
Gambar 2.4 Sensor LDR

Prinsip kerja LDR sangat sederhana tak jauh berbeda dengan variable resistor pada umumnya. LDR dipasang pada berbagai macam rangkaian elektronika dan dapat memutus dan menyambungkan aliran listrik berdasarkan cahaya. Semakin banyak cahaya yang mengenai LDR maka nilai resistansinya akan menurun, dan sebaliknya semakin sedikit cahaya yang mengenai LDR maka nilai hambatannya akan semakin membesar.

2.6 NodeMCU V3

NodeMCU pada dasarnya adalah pengembangan dari ESP 8266 dengan *firmware* berbasis e-Lua. NodeMcu ditunjukkan pada gambar 2.5 .Pada NodeMcu dilengkapi dengan *micro usb port* yang berfungsi untuk pemrograman maupun *power supply*. Selain itu juga pada NodeMCU di lengkapi dengan tombol push button yaitu tombol reset dan flash. NodeMCU menggunakan bahasa pemrograman Lua yang merupakan *package* dari esp8266. Bahasa Lua memiliki logika dan susunan pemrograman yang sama dengan c hanya berbeda syntax. Jika menggunakan bahasa Lua maka dapat menggunakan tool Lua loader maupun Lua uploder. Selain dengan bahasa Lua NodeMCU juga support dengan *software* Arduino IDE dengan melakukan sedikit perubahan board manager pada Arduino IDE. Sebelum digunakan Board ini harus di Flash terlebih dahulu agar support terhadap tool yang akan digunakan. Jika menggunakan Arduino IDE menggunakan *firmware* yang cocok yaitu *firmware* keluaran dari AiThinker yang support AT Command. Untuk penggunaan tool loader *Firmware* yang di gunakan adalah

firmware NodeMCU. NodeMCU adalah arduino yang telah di lengkapi dengan wifi untuk menghubungkan ke IoT. Untuk spesifikasi NodeMCU V3 dapat dilihat pada tabel 2.1



Gambar 2.5 NodeMCU V3

Tabel 2.1 Spesifikasi NodeMCU V3

<i>Microcontroller</i>	ESP8266
<i>Ukuran Board</i>	57 mm x 30 mm
<i>Input Voltage (recommended)</i>	3.3 ~ 5V
<i>GPIO</i>	13 PIN
<i>Kanal PWM</i>	10 Kanal
<i>10 bit ADC Pin</i>	1 Pin
<i>Flash Memory</i>	4 MB
<i>Clock Speed</i>	40/26/24 MHz
<i>WiFi</i>	IEEE 802.11 b/g/n
<i>Frekuensi</i>	2.4 GHz – 22.5 Ghz
<i>USB Port</i>	Micro USB
<i>Card Reader</i>	Tidak Ada
<i>USB to Serial Converter</i>	CH340G

2.7 Rangkaian dimmer

Rangkaian dimmer lampu pijar ini berfungsi untuk mengatur tingkat intensitas cahaya penerangan lampu pijar. Rangkaian ini bisa diatur mulai dari yang redup hingga ke remang-remang sampai ke nyala lampu yang terang. Dan juga bisa membuat rangkaian dimmer pengatur nyala lampu dengan pola sederhana.

Di dalam rangkaian dimmer ini, terdapat 3 komponen penting guna mengatur kerja dimmer ini. Komponen TRIAC berfungsi untuk mengatur besaran tegangan AC yang masuk ke perangkat lampu ini. Sementara komponen DIAC dan VR berfungsi untuk mengatur bias TRIAC guna menentukan titik on dan off pada komponen TRIAC ini. Daya output rangkaian dimmer ini dapat digunakan untuk mengendalikan intensitas cahaya lampu pijar dengan daya 60 Watt.

Pada rangkaian ini potensiometer berfungsi sebagai sensor mekanis pengatur besar kecilnya lampu atau transduser pasif yang perlu mendapatkan daya dari luar. Rangkaian dimmer ini hanya cocok untuk di pakai untuk lampu pijar saja. Jika digunakan untuk lampu neon atau TL, dan juga lampu hemat energi, rangkaian ini tidak bisa bekerja sempurna. Bahkan rangkaian dimmer akan mengalami kerusakan pada rangkaian dimmer tersebut. Rangkaian dimmer lampu pijar dapat digunakan untuk jaringan listrik PLN 220VAC.

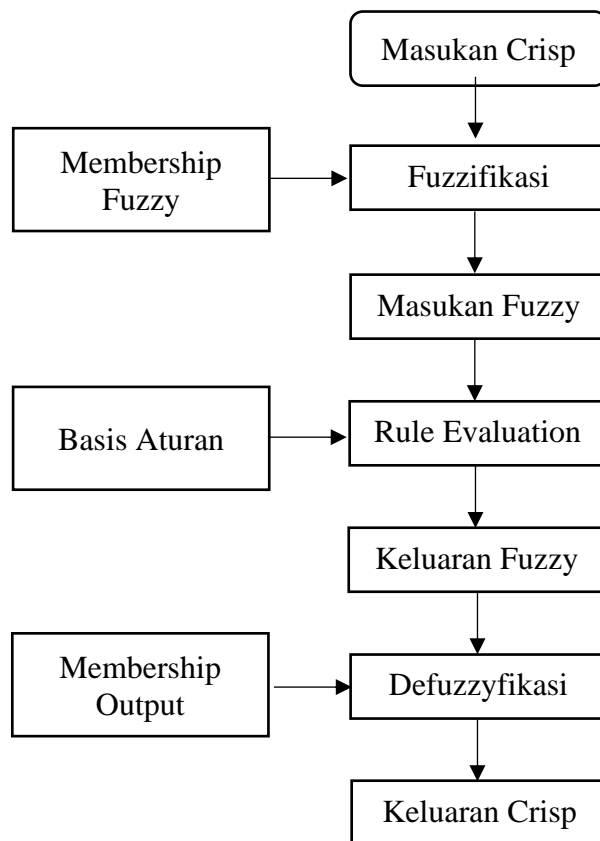
2.8 Fuzzy Logic Systems

Logika *fuzzy* merupakan salah satu pembentuk soft computing. Logika *fuzzy* pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965. Dasar logika *fuzzy* adalah teori himpunan *fuzzy*. Pada teori himpunan *fuzzy*, peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting. Nilai keanggotaan atau derajat keanggotaan atau membership function menjadi ciri utama dari penalaran dengan logika *fuzzy* tersebut. (Irfan et al., 2018)

Ada beberapa definisi logika *fuzzy*, diantaranya :

1. Logika *fuzzy* adalah logika yang digunakan untuk menjelaskan keambiguan, logika himpunan yang menyelesaikan keambiguan. (Vrusias, 2008).

2. Logika *fuzzy* menyediakan suatu cara untuk merubah pernyataan linguistik menjadi suatu numerik. (Synaptic, 2006).



Gambar 2.6. Flowchart Tahapan – tahapan dasar logika Fuzzy

Dari Gambar 2.6 dapat kita lihat bahwa secara garis besar ada 3 tahapan dalam logika Fuzzy yaitu fuzzifikasi, rule evaluation dan defuzzifikasi.

1. Fuzzifikasi Fuzzifikasi merupakan sutau tahapan untuk merubah nilai input yang berupa variabel crisp menjadi bentuk Fuzzy dengan menentukan nilai derajat keanggotaan terlebih dahulu. Sehingga kemudian input dapat dikelompokkan pada himpunan Fuzzy yang tepat agar masukan controller Fuzzy bisa dipetakan agar sesuai dengan himpunan Fuzzy (Triyani, 2018).

2. Rule Evaluation Basis aturan merupakan aturan Fuzzy yang digunakan untuk pengendalian sistem. Aturan itu sendiri dibuat berdasarkan logika manusia yang berkaitan dengan jalan pikiran serta penelitian yang dilakukan pembuatnya. Aturan ini dibuat untuk membandingkan antara masukan dari fuzzifikasi dengan aturan yang sudah dibuat sesuai intuisi. Sehingga output yang diinginkan akan sesuai dengan rule base yang telah dibuat (Triyani, 2018).
3. Defuzzifikasi merupakan proses lanjutan setelah pengambilan keputusan berdasarkan basis aturan (rule evaluation) yang sudah ditetapkan. Pada penelitian ini digunakan output non-singleton dengan metode COA (Centre of Area) sebagai keluarannya (Triyani, 2018).

Logika *fuzzy* memiliki derajat keanggotaan dalam rentang 0 hingga 1. Berbeda dengan logika digital yang hanya memiliki dua nilai 1 atau 0. Logika *fuzzy* digunakan untuk menerjemahkan suatu besaran yang diekspresikan menggunakan bahasa (linguistic), misalkan besaran kecepatan laju kendaraan yang diekspresikan dengan pelan, agak cepat, cepat, dan sangat cepat. Dan logika *fuzzy* menunjukkan sejauh mana suatu nilai itu benar dan sejauh mana suatu nilai itu salah. Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang input kedalam suatu ruang output. *Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama. (Irfan et al., 2018).

Metode Mamdani adalah salah satu teknik inferensi *fuzzy* yang juga

disebut dengan Metode *Max-Min*. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Pada metode ini, terdapat 4 tahap untuk mendapatkan *output*, yaitu : *Fuzzification*, *Rule Evaluation*, *Rule Aggregation*, *Defuzzification* .

2.8.1 Fuzzification (Pembentukan himpunan fuzzy)

Fuzzification adalah langkah pertama dari metode Mamdani yang bertugas mengambil nilai *input* berupa nilai *crisps*, dan menentukan derajat dari *input* sehingga *input* dapat dikelompokkan pada himpunan *fuzzy* yang tepat. Tahap pertama ini, nilai *input* yang berupa nilai *crisp* akan dikonversikan menjadi nilai *fuzzy*, sehingga dapat dikelompokkan pada himpunan *fuzzy* tertentu (Simorangkir & Nur, 2013).

2.8.2 Rule Evaluation (Aplikasi fungsi implikasi)

Langkah kedua adalah mengambil nilai *input* yang telah di-fuzzifikasikan dan mengaplikasikannya ke dalam *antecedents* pada aturan-aturan *fuzzy* lalu diimplikasikan. Fungsi implikasi yang digunakan adalah *Min*. Seperti pada Persamaan 2.1 :

$$\mu_{A \cap B}(x) = \min(\mu_A[x], \mu_B[x]) \quad (2.1)$$

2.8.3 Rule Aggregation (Komposisi Aturan)

Aggregasi aturan adalah proses dari penggabungan nilai keluaran dari semua aturan. Pada tahap ini, terdapat 3 metode yang digunakan dalam melakukan inferensi sistem *fuzzy*, yaitu *Max*, *Additive* dan Probabilistik OR (*probor*).

2.8.4 Metode Max (Maximum)

Solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah *fuzzy*, dan mengaplikasikannya ke *output* dengan menggunakan operator OR

(*union*). Secara umum dapat dituliskan seperti pada Persamaan 2.2 :

$$\mu_{sf}[x_i] = \mathbf{max}(\mu_{sf}[x_i], \mu_{kf}[x_i]) \quad (2.2)$$

Dengan :

$\mu_{sf}[x_i]$ = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;

$\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen *fuzzy* aturan ke-i; Misalkan ada 3 aturan (proposisi) sebagai berikut :

[R1] IF Biaya Produksi RENDAH And Permintaan NAIK THEN Produksi Barang BERTAMBAH;

[R2] IF Biaya Produksi STANDAR THEN Produksi Barang NORMAL;

[R3] IF Biaya Produksi TINGGI And Permintaan TURUN THEN Produksi Barang BERKURANG;

2.8.5 Metode Additive (Sum)

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara melakukan *bounded-sum* terhadap semua *output* daerah *fuzzy*. Secara umum dapat dituliskan seperti pada Persamaan 2.3 :

$$\mu_{sf}[x_i] = \mathbf{min}(1, \mu_{sf}[x_i] + \mu_{kf}[x_i]) \quad (2.3)$$

Dengan :

$\mu_{sf}[x_i]$ = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;

$\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen *fuzzy* aturan ke-i;

2.8.6 Metode Probabilistik OR (Probor)

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara melakukan *product* terhadap semua *output* daerah *fuzzy*. Secara umum dapat dituliskan seperti pada Persamaan 2.4 :

$$\mu_{sf}[x_i] = (\mu_{sf}[x_i] + \mu_{kf}[x_i]) - (\mu_{sf}[x_i]) * (\mu_{kf}[x_i]) \quad (2.4)$$

Dengan :

$\mu_{sf}[x_i]$ = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;

$\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen *fuzzy* aturan ke-i;

2.8.7 Defuzzification (Penegasan)

Langkah terakhir dari proses inferensi *fuzzy* adalah untuk mengkonversi versi nilai *fuzzy* hasil dari agregasi aturan ke dalam sebuah bilangan *crisp*. Input dari proses defuzzifikasi adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *fuzzy* dalam *range* tertentu (Simorangkir & Nur, 2013). Terdapat beberapa metode defuzzifikasi pada komposisi aturan Mamdani, yaitu:

- Metode Centroid

Solusi *crisp* diperoleh dengan cara mengambil titik pusat (z^*) daerah *fuzzy*. Secara umum dapat dituliskan seperti pada persamaan 2.5 :

$$Z^* = \frac{\int_z z\mu(z)dz}{\int_z \mu(z)dz} \quad \text{untuk variabel kontinu} \quad (2.5)$$

$$Z^* = \frac{\sum_{j=1}^n z_j \mu(z_j)}{\sum_{j=1}^n \mu(z_j)} \quad \text{untuk variabel diskret}$$

- Metode Biseksi

Solusi crisp diperoleh dengan cara mengambil nilai pada domain fuzzy yang memiliki nilai keanggotaan setengah dari jumlah total nilai keanggotaan pada daerah fuzzy. Secara umum dapat dituliskan seperti pada persamaan 2.6 :

$$z_p \text{ sedemikian hingga } \int_{R_1} \mu(z) dz = \int_p \mu(z) dz \quad (2.6)$$

- Metode Mean of Maximum (MOM)

Pada metode ini, solusi crisp diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.

- Metode Largest of Maximum (LOM)

Pada metode ini, solusi crisp diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

- Metode Smallest of Maximum (SOM)

Terakhir, pada metode ini, solusi crisps diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

2.9 Tingkat Pencahayaan Minimum yang Direkomendasi

Tingkat pencahayaan minimum dan renderasi warna yang direkomendasikan sesuai Standar Nasional Indonesia (SNI) untuk berbagai fungsi ruangan ditunjukkan pada tabel 2.2.

Tabel 2.2 Tingkat Pencahayaan Minimum Rumah Tinggal sesuai Standar Nasional Indonesia (SNI)

Fungsi Ruangan	Tingkat Pencahayaan (Lux)	Kelompok Renderasi Warna
Teras	60	1 atau 2
Ruang Tamu	120 – 250	1 atau 2
Ruang Makan	120 – 250	1 atau 2
Ruang Kerja	120 – 250	1
Kamar Tidur	120 – 250	1 atau 2
Kamar Mandi	250	1 atau 2
Dapur	250	1 atau 2
Garasi	60	3 atau 4

2.10 Penelitian Sebelumnya

Penelitian tentang sistem kendali pencahayaan ruang kerja telah banyak dilakukan oleh peneliti terdahulu. Penelitian yang dilakukan menghasilkan berbagai hasil yang berbeda. Penelitian terdahulu yang menjadi acuan peneliti yang sesuai dengan penelitian saat ini antara lain :

1. (Putro & Kambey, 2016)

Penelitian ini bertujuan untuk mengatur pencahayaan ruangan berbasis Android pada rumah pintar. Kinerja pencahayaan ruangan dalam penelitian ini diukur dengan menggunakan Luxmeter. Hasil dari penelitian ini meliputi sistem perangkat keras yang terdiri atas perangkat sistem pengendali berbasis arduino nano yang secara langsung mengendalikan pencahayaan ruangan dengan menggunakan sensor cahaya dan mekanik dimmer untuk meredupkan dan menerangkan bola lampu. Sedangkan hasil dari sistem perangkat lunak terdiri atas perancangan sistem pemrograman cerdas yang dilakukan pada sistem arduino nano dan pemrograman

mobile dengan hasil tampilan aplikasi smartphone berbasis android ini terdiri atas 9 screen, yang didalamnya terdapat menu utama, menu pilihan pencahayaan, menu otomatis, dan menu manual. Untuk Tingkat akurasi tertinggi dalam proses kalibrasi pembacaan sensor pada sistem pengaturan pencahayaan ruangan terdapat pada ruang tamu dengan nilai kesalahan 0,4% dan ketelitian 99,6%. Sedangkan tingkat akurasi terendah terdapat pada toilet yaitu dengan nilai kesalahan sebesar 29,4% dan ketelitian 70,6%. Adapun Perbedaan dengan penelitian terdahulu yaitu :

- a. Menggunakan Metode *fuzzy logic* dalam pengambilan keputusan.
- b. Menggunakan NodeMCU sebagai *processor* inti pada alat yang dibuat
- c. Menggunakan Dimmer Robotdyn sebagai pengatur cahaya lampu.

Sedangkan untuk Persamaan dengan penelitian terdahulu yaitu menggunakan sensor LDR sebagai *Input* pada alat.