

BAB II

LANDASAN TEORI

2.1. Tinjauan Terhadap Penelitian Yang Terkait

Heriyanto Antony dan Sean Coonery Sumarta (2017), telah melakukan penelitian dengan judul "Sistem Otomatisasi Rumah Berbasis Logika *Fuzzy* Menggunakan *Arduino UNO R3* Untuk Mengoptimalkan Penggunaan Listrik". Suatu sistem yang mampu mengendalikan peralatan listrik secara optimal dengan menggunakan *Arduino Uno R3* dan beberapa sensor dengan menggunakan metode *fuzzy tsukamoto*. Logika *fuzzy tsukamoto* berperan mengontrol suhu AC yang akan digunakan dalam pengontrolan sehingga suhu ruangan tidak terlalu panas maupun terlalu dingin. Sehingga penggunaan peralatan listrik menggunakan sistem otomatisasi rumah dapat bekerja secara optimal.

Nesi Syafitri. N (2016), melakukan penelitian yang berjudul "Simulasi Sistem Untuk Pengontrolan Lampu Dan *Air Conditioner* Dengan Menggunakan Logika *Fuzzy*". Suatu sistem yang mampu mengontrol penggunaan lampu dan AC tersebut secara otomatis untuk mendapatkan efisiensi penggunaan listrik. Sistem simulasi pengontrolan ini dirancang dengan prinsip kendali logika *fuzzy* dengan menggunakan sistem inferensi *fuzzy* adalah metode *tsukamoto*. Parameter yang digunakan untuk mengatur pemakaian lampu berdasarkan intensitas cahaya yang ada. Sementara parameter untuk mengatur temperatur AC berdasarkan banyak orang dan temperatur udara. Nilai tegas output yang dihasilkan berupa temperatur AC dan jumlah lampu yang bisa dihidupkan.

Sistem pengontrolan dengan logika *fuzzy* lebih efektif dibandingkan dengan cara konvensional, hal ini dikarenakan sistem pengontrolan dengan logika *fuzzy* dapat menyesuaikan dengan banyak orang, temperatur udara dan intensitas cahaya yang terjadi di sebuah ruangan.

Bobby Y. Prawira, Vecky C. Poekoel dan Feisy D. Kambey (2018), telah melakukan penelitian yang berjudul "Efisiensi Pencahayaan Ruangan Perkuliahan dengan Logika *Fuzzy*". Sistem cerdas yang menggunakan logika *fuzzy* ini menggunakan *microcontroller Arduino Nano* sebagai pengendali. Untuk menangkap sinar matahari digunakan sensor LDR. Sedangkan, untuk membuat redup dan terangnya lampu digunakan dimmer yang digerakkan menggunakan motor servo sehingga kita dapat menjalankan fungsi dari sistem tersebut. Hasil dari penelitian ini menunjukkan bahwa sistem pencahayaan ruang perkuliahan yang menggunakan pengendali lebih efisien daripada tanpa menggunakan pengendali.

Kemudian Galoeh Otomo dan Wildian (2015), telah melakukan penelitian dengan judul "Sistem Kontrol Penyalaan Lampu Ruang Berdasarkan Pendeteksian Ada Tidaknya Orang Di Dalam Ruangan". Sistem kontrol lampu akan menyala selama ada orang didalam ruangan, dan lampu akan mati ketika orang meninggalkan ruangan. Keberadaan orang akan di deteksi oleh sensor *passive infrared (PIR)*. Jarak waktu respon dari sensor PIR KC7783R telah dicobakan, dimana sensor hanya dapat mendeteksi objek selama 5,37 detik, namun dapat diatasi dengan menggunakan program yang ditanamkan ke dalam mikrokontroler

AT89S51. Jarak maksimum yang dapat di deteksi sensor PIR adalah 4,3 meter pada sudut 0° (lurus dari depan sensor), dan 2 meter pada sudut 30° (kekiri dan kekanan). Sensor membutuhkan waktu pemanasan selama 25,52 detik. Relay digunakankan untuk menghubungkan antara arus dc dan arus ac.

2.2. Efisiensi Energi Listrik

Menurut Hidayanto (2012) pada hakekatnya telah diketahui bahwa efisiensi energi merupakan bagian dari konservasi energi. Dalam kebijakan energi nasional disebutkan bahwa konservasi energi merupakan upaya yang sistematis terencana dan terpadu guna melestarikan sumber daya energi dalam negeri serta meningkatkan efisiensi pemanfaatannya. Di Indonesia, upaya konservasi energi ini sangat penting mengingat besarnya kesenjangan antara sisi permintaan dan sisi penyediaan.

Menurut Suprayogi (2014) audit energi dianjurkan untuk dilaksanakan pada bangunan seperti gedung perkantoran, sekolah, hotel, apartemen, pusat perbelanjaan dan rumah sakit. Melalui audit energi diharapkan dapat mengetahui besarnya intensitas konsumsi energi (IKE), mencegah pemborosan energi tanpa mengurangi kenyamanan gedung, dapat diketahui profil energi dan mencari upaya yang diperlukan untuk meningkatkan efisiensi penggunaan energi.

Salah satu contoh upaya dalam melakukan efisiensi energi listrik misalnya pada bangunan ruang kelas adalah dengan membuat suatu sistem pengaturan pencahayaan dan pendinginan pada ruang kelas supaya bisa

menghemat konsumsi energi listrik dengan cara mengontrol tingkat pencahayaan dan pendinginan ruangan secara otomatis menyesuaikan dengan kondisi cahaya dan jumlah orang yang ada dengan target tingkat penerangan dan pendinginan ruangan yang telah ditentukan.

2.3. Fuzzy Logic

Pada tahun 1965 seorang guru besar di *University of California* yang bernama Prof. Lotfi A. Zadeh merupakan pencetus sekaligus memasarkan ide tentang cara mekanisme pengolahan atau manajemen ketidakpastian yang kemudian dikenal sebagai logika *fuzzy*. Dalam penyajiannya variabel-variabel yang akan digunakan harus menggambarkan ke-*fuzzy*-an tetapi di lain pihak persamaan-persamaan yang dihasilkan dari variabel-variabel itu haruslah cukup sederhana sehingga komputasinya menjadi agak cukup mudah. Karena itu Profesor Lotfi A Zadeh kemudian memperoleh ide untuk menyajikannya dengan menentukan “derajat keanggotaan” (*membership function*) dari masing-masing variabelnya.

Terdapat lima langkah dalam melakukan penalaran secara umum pada logika fuzzy, yaitu :

- a) Memasukan input *fuzzy*
- b) Mengaplikasikan operator
- c) Mengaplikasikan metode implikasi
- d) Komposisi semua *output*
- e) Defuzzifikasi

2.3.1. Fungsi Keanggotaan

Fungsi keanggotaan (*membership fuction*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering juga disebut derajat keanggotaan) yang memiliki interval antara nol sampai dengan satu (Frans Susilo, 2003). Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaannya adalah dengan melalui pendekatan fungsi.

Terdapat beberapa fungsi keanggotaan yaitu :

A. Linear

Pada representasi linear ini pemetaan input ke dalam derajat keanggotaannya digambarkan sebagai suatu garis lurus. untuk mendekati suatu konsep yang kurang jelas. Ada 2 keadaan himpunan *fuzzy* yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol (0) kemudian bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi seperti yang dapat dinyatakan dalam persamaan ;

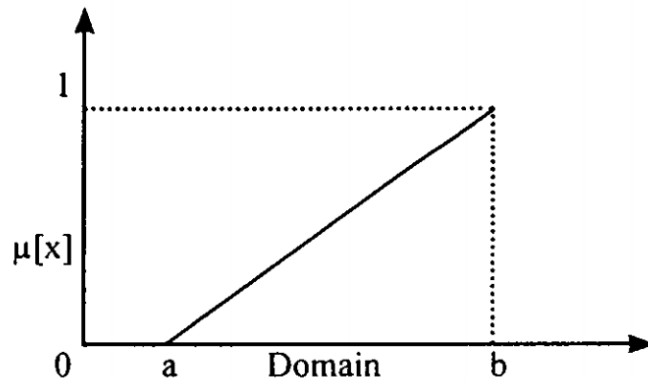
$$\mu(x) = \begin{cases} 0; & x \leq a \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ 1; & x \geq b \end{cases}$$

Keterangan :

a = Nilai domain yang mempunyai derajat keanggotaan nol

b = Nilai domain yang mempunyai derajat keanggotaan Satu

x = Nilai input yang akan di ubah kedalam bilangan *fuzzy*



Gambar 2.1 Representasi Linear Naik

B. Kurva Segitiga

Kurva segitiga ini pada dasarnya merupakan gabungan antara dua garis linear. Bentuk fungsi keanggotaan segitiga dapat dinyatakan dalam persamaan ;

$$\mu(x) = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ \frac{(b-x)}{(c-b)}; & b \leq x \leq c \end{cases}$$

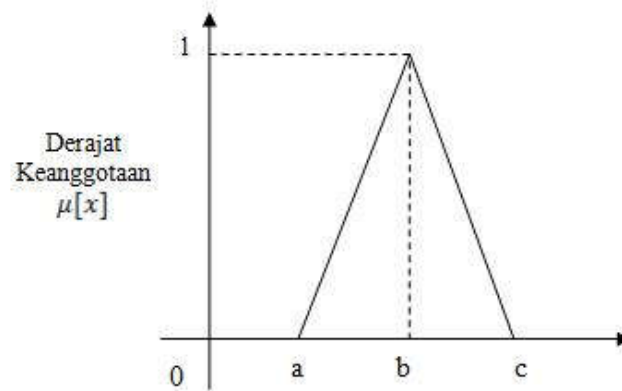
Keterangan :

a = Nilai domain terkecil yang mempunyai derajat keanggotaan nol

b = Nilai domain yang mempunyai derajat keanggotaan Satu

c = Nilai domain terbesar yang mempunyai derajat keanggotaan nol

x = Nilai input yang akan di ubah kedalam bilangan *fuzzy*



Gambar 2.2 Representasi Kurva Segitiga

C. Kurva Trapesium

Kurva trapesium ini pada dasarnya sama dengan kurva segitiga hanya saja pada kurva trapesium nilai keanggotaan 1 lebih banyak atau ada di beberapa titik. Berikut merupakan fungsi keanggotaan trapesium yang dapat dilihat pada persamaan ;

$$\mu(x) = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ 1; & b \leq x \leq c \\ \frac{(d-x)}{(d-c)}; & x \geq d \end{cases}$$

Keterangan :

a = Nilai domain terkecil yang mempunyai derajat keanggotaan nol

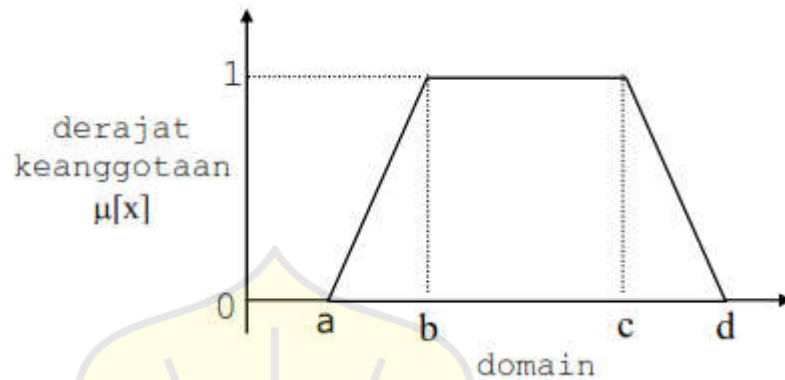
b = Nilai domain terkecil yang mempunyai derajat keanggotaan

Satu

c = Nilai domain terbesar yang mempunyai derajat keanggotaan nol

d = Nilai domain terbesar yang mempunyai derajat keanggotaan satu

x = Nilai input yang akan di ubah kedalam bilangan *fuzzy*



Gambar 2.3 Representasi Kurva Trapesium

D. Kurva S

Kurva penyusutan atau pertumbuhan merupakan kurva -s atau signoid yang berhubungan dengan kenaikan dan penurunan permukaan secara tidak linear. Kurva-S untuk kenaikan akan bergerak dari sisi paling kiri (nilai keanggotaan = 0) ke sisi paling kanan (nilai keanggotaan = 1). Fungsi keanggotaannya akan tertumpu pada 50% nilai keanggotaannya yang sering disebut dengan titik infleksi. Kurva-S untuk penurunan akan bergerak dari sisi paling kanan (nilai keanggotaan = 1) ke sisi paling kiri (nilai keanggotaan = 0).

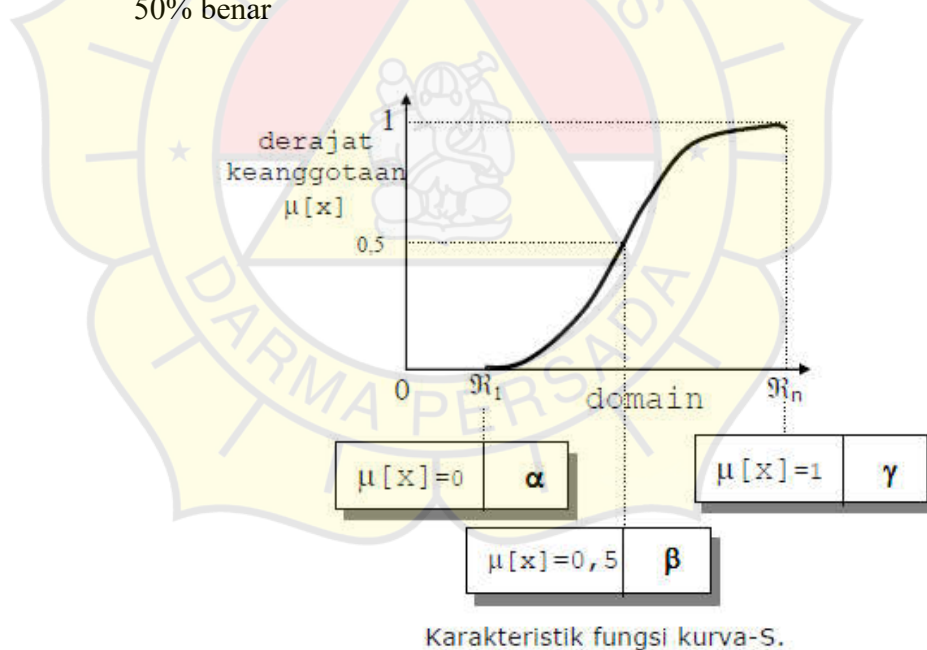
Kurva-S didefinisikan dengan menggunakan 3 parameter, yaitu: nilai keanggotaan nol (α), nilai keanggotaan lengkap (γ), dan titik infleksi atau crossover (β) yaitu titik yang memiliki domain 50%

benar. Berikut merupakan fungsi keanggotaan signoid yang dapat dilihat pada persamaan ;

$$S(x; \alpha, \beta, \gamma) = \begin{cases} 0; & x \leq \alpha \\ 2 \frac{(x - \alpha)}{(\gamma - \alpha)^2}; & \alpha \leq x \leq \beta \\ 1 - 2 \left(\frac{(\gamma - x)}{(\gamma - \alpha)} \right)^2; & \beta \leq x \leq \gamma \\ 1; & x \geq \gamma \end{cases}$$

Kurva-S didefinisikan dengan menggunakan tiga parameter, yaitu:

- 1) Nilai keanggotaan nol (α)
- 2) Nilai keanggotaan lengkap (γ)
- 3) Titik infleksi atau crossover (β) yaitu titik yang memiliki domain 50% benar



Gambar 2.4 Representasi Kurva S

Sistem inferensi *fuzzy* atau yang lebih dikenal dengan *Fuzzy Inference System (FIS)* merupakan penarikan sebuah kesimpulan dari kumpulan kaidah *fuzzy*. Jadi pada sistem inferensi *fuzzy*

melakukan penalaran menggunakan masukan *fuzzy* (*Crisp Value*) dan *fuzzy rules* yang telah di tentukan sehingga menghasilkan sebuah keluaran *fuzzy* (*Crisp Value*). Terdapat dua metode pada sistem inferensi *fuzzy* yaitu : Metode Mamdani dan Metode Sugeno. Pembahasan teori disini hanya akan membahas metode Sugeno.

2.3.2. Fuzzy Sugeno

Metode Fuzzy Metode inferensi Sugeno diperkenalkan oleh Tomohiro Takagi dan Michio Sugeno pada tahun 1985. Metode ini mirip dengan metode Mamdani. Hanya saja, *output* aturan *fuzzy* pada metode Sugeno tidaklah berbentuk himpunan *fuzzy*, melainkan konstanta atau persamaan linear. Selain itu, defuzzifikasi pada metode ini biasanya menggunakan metode rata-rata berbobot.

Output aturan *fuzzy* yang berupa konstanta atau persamaan linear membuat metode Takagi-Sugeno menjadi metode yang mudah dikomputasi, efisien, dan mudah digabungkan dengan persamaan linear.

Sistem *fuzzy* Sugeno memperbaiki kelemahan yang dimiliki oleh sistem *fuzzy* murni untuk menambah suatu perhitungan matematika sederhana sebagai bagian THEN. Pada perubahan ini, sistem *fuzzy* memiliki suatu nilai rata-rata tertimbang (*Weighted Average Values*) di dalam bagian aturan *fuzzy* IF-THEN.

Ada 2 model *fuzzy* metode Sugeno yaitu sebagai berikut:

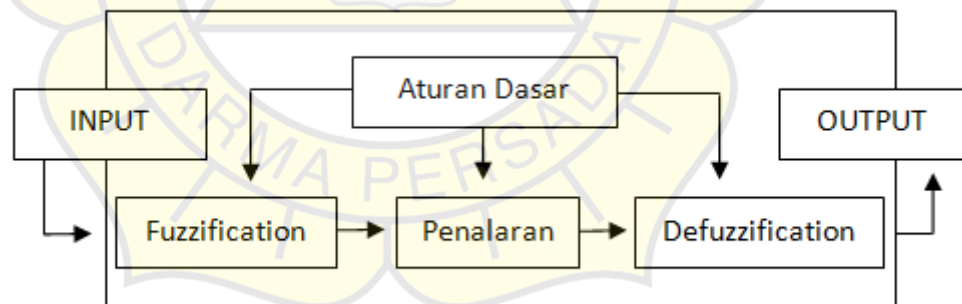
a. Model Fuzzy Sugeno Orde-Nol

Secara umum bentuk model Fuzzy Sugeno Orde Nol adalah: IF (x_1 is A_1) \circ (x_2 is A_2) \circ (x_3 is A_3) \circ ... \circ (x_N is A_N) THEN $z = k$ dengan A_i adalah himpunan *fuzzy* ke- i sebagai antesenden, dan k adalah suatu konstanta sebagai konsekuen.

b. Model Fuzzy Sugeno Orde-Satu

Secara umum bentuk model Fuzzy Sugeno Orde-Satu adalah: IF (x_1 is A_1) \circ (x_2 is A_2) \circ (x_3 is A_3) \circ ... \circ (x_N is A_N) THEN $z = p_1 * x_1 + \dots + p_N * x_N + q$ dengan A_i adalah himpunan *fuzzy* ke- i sebagai antesenden, dan p_i adalah suatu konstanta ke- i dan q juga merupakan konstanta dalam konsekuen.

Berikut ini adalah tahapan – tahapan metode *fuzzy* sugeno :



Gambar 2.5 Tahapan Metode Fuzzy Sugeno

1) Pembentukan himpunan Fuzzy

Pada tahapan ini variabel input dari sistem *fuzzy* ditransfer ke dalam himpunan *fuzzy* untuk dapat digunakan dalam perhitungan nilai kebenaran dari premis pada setiap aturan dalam

basis pengetahuan. Dengan demikian tahap ini mengambil nilai-nilai tegas dan menentukan derajat di mana nilai-nilai tersebut menjadi anggota dari setiap himpunan *fuzzy* yang sesuai.

2) Penalaran / Implikasi

Tiap-tiap aturan (proposisi) pada basis pengetahuan *fuzzy* akan berhubungan dengan suatu relasi *fuzzy*. Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah sebagai berikut: IF x is A THEN y is B dengan x dan y adalah skalar, dan A dan B adalah himpunan *fuzzy*. Proposisi yang mengikuti IF disebut sebagai antesenden sedangkan proposisi yang mengikuti THEN disebut konsekuen. Proposisi ini dapat diperluas dengan menggunakan operator *fuzzy* seperti, IF $(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_N \text{ is } A_N)$ THEN y is B dengan \circ adalah operator (misal: OR atau AND).

Secara umum fungsi implikasi yang dapat digunakan yaitu sebagai berikut:

- Min (minimum) Fungsi ini akan memotong *output* himpunan *fuzzy*.
- Dot (*product*) Fungsi ini akan menskala *output* himpunan *fuzzy*.

Pada metode Sugeno ini, fungsi implikasi yang digunakan adalah fungsi min.

3) Aturan / Rule

Tahap ketiga dari prosedur metode *fuzzy* sugeno adalah komposisi aturan. Jika sistem terdapat beberapa aturan, maka inferensi akan didapat dengan menggabungkan antar aturan. Untuk memperoleh inferensi *fuzzy* dapat dilakukan dengan tiga metode, yaitu sebagai berikut :

a. Metode Max (maximum)

Pada metode max, solusi himpunan *fuzzy* dapat diperoleh dengan melakukan modifikasi daerah *fuzzy*, dan menggunakan operator OR untuk mengaplikasikan ke *output*. Setelah dievaluasi semua proporsi maka *output* akan memiliki nilai himpunan *fuzzy* yang merepresentasikan dari tiap-tiap proporsi. Secara umum dapat ditulis sebagai berikut :

$$\mu_{sf} [X_i] = \max(\mu_{sf} [x_i] - \mu_{sf} [x_i])$$

Keterangan :

$\mu_{sf} (X_i)$ = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i

$\mu_{sf} (X_i)$ = nilai keanggotaan konsekuen *fuzzy* aturan ke-i

b. Metode Additive (sum)

Pada metode ini, solusi untuk memperoleh himpunan *fuzzy* didapat dengan melakukan *bounded sum* terhadap semua daerah *output fuzzy*. Dapat dituliskan secara umum sebagai berikut :

$$\mu_{sf} [X_i] = \min (1. \mu_{sf} [x_i] + \mu_{sf} [x_i])$$

Keterangan :

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

$\mu_{sf} [X_i]$ = nilai keanggotaan konsekuen *fuzzy* aturan ke-i

c. Metode Probabilistik OR (probor)

Pada metode ini, untuk memperoleh solusi himpunan *fuzzy* dilakukan dot pada semua daerah *output fuzzy*. Dapat dituliskan secara umum sebagai berikut :

$$\mu_{sf} [X_i] = (\mu_{sf} [x_i] + \mu_{sf} [x_i]) - (\mu_{sf} [x_i] - \mu_{sf}[x_i]))$$

Keterangan :

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

$\mu_{sf} [X_i]$ = nialai keanggotaan konsekuen *fuzzy* aturan ke-i

4) Defuzzifikasi

Input dari proses defuzzifikasi adalah suatu himpunan *fuzzy* yang diperoleh dari suatu komposisi aturan-aturan *fuzzy*, sedangkan output yang dihasilkan merupakan suatu bilangan pada himpunan *fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *fuzzy* dalam range tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai output. Dalam metode sugeno, defuzzifikasi dilakukan dengan cara mencari rata-rata terbobot (*weighted average*) ;

$$WA = \frac{\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3 + \dots + \alpha_n z_n}{\alpha_1 + \alpha_2 + \alpha_3 + \dots + \alpha_n}$$

Keterangan:

WA= Nilai rata-rata

αn = nilai predikat aturan ke-n

zn = indeks nilai output (konstanta) ke-n.

2.4. Arduino Uno

Menurut (Feri Djuandi, 2011) *Arduino* merupakan sebuah *board* minimum sistem mikrokontroler yang bersifat *open source*. Didalam rangkaian *board arduino* terdapat mikrokontroler AVR seri ATmega 328 yang merupakan produk dari Atmel.

Arduino memiliki kelebihan tersendiri dibanding *board* mikrokontroler yang lain selain bersifat *open source*, *arduino* juga mempunyai bahasa pemrogramannya sendiri yang berupa bahasa C. Selain itu dalam *board arduino* sendiri sudah terdapat *loader* yang berupa USB sehingga memudahkan kita ketika kita memprogram mikrokontroler didalam *arduino*. Sedangkan pada kebanyakan *board* mikrokontroler yang lain yang masih membutuhkan rangkaian *loader* terpisah untuk memasukkan program ketika kita memprogram mikrokontroler. Port USB tersebut selain untuk *loader* ketika memprogram, bisa juga difungsikan sebagai port komunikasi serial.

2.4.1. Software Arduino

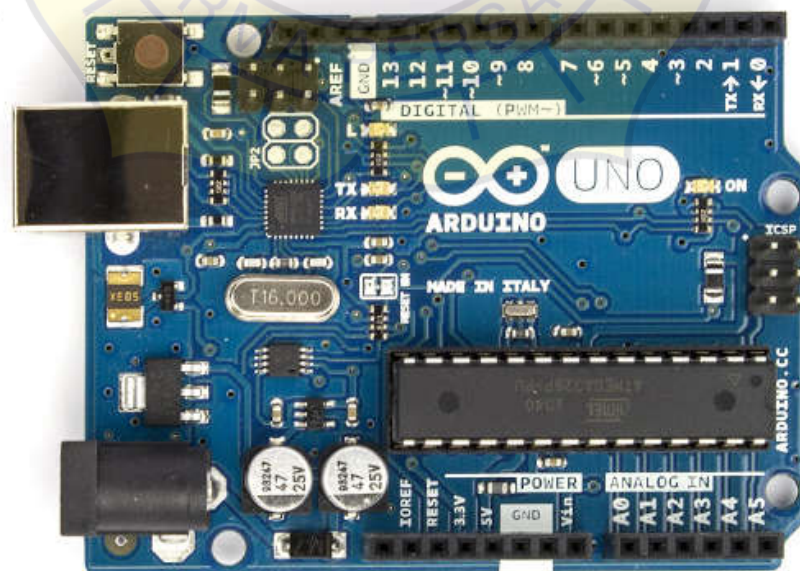
Arduino Uno dapat diprogram dengan perangkat lunak *arduino*. Pada ATmega328 di *arduino* terdapat *bootloader* yang memungkinkan Anda untuk meng-*upload* kode baru untuk itu tanpa menggunakan *programmer hardware* eksternal.

IDE Arduino adalah *software* yang sangat canggih ditulis dengan menggunakan *Java*. IDE Arduino terdiri dari :

1. Editor Program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *Processing*.
2. Compiler, sebuah modul yang mengubah kode program (bahasa *processing*) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *processing*. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya compiler diperlukan dalam hal ini.
3. Uploader, sebuah modul yang memuat kode biner dari komputer ke dalam memory didalam papan *arduino*.

2.4.2. Bagian - Bagian Papan Arduino

Dengan mengambil contoh sebuah papan Arduino tipe USB, bagian-bagiannya dapat dijelaskan sebagai berikut :



Gambar 2.6 Papan Arduino Uno

a. 14 pin input/output digital (0-13)

Berfungsi sebagai *input* atau *output*, dapat diatur oleh program. Khusus untuk 6 buah pin 3, 5, 6, 9, 10 dan 11, dapat juga berfungsi sebagai pin analog *output* dimana tegangan *output*-nya dapat diatur. Nilai sebuah pin *output* analog dapat diprogram antara 0-255, dimana hal itu mewakili nilai tegangan 0-5V.

b. USB

USB berfungsi untuk :

1. Memuat program dari komputer ke dalam papan
2. Komunikasi serial antara papan dan komputer
3. Memberi daya listrik kepada papan

c. Sambungan SV1

Sambungan atau jumper untuk memilih sumber daya papan, apakah dari sumber eksternal atau menggunakan USB. Sambungan ini tidak diperlukan lagi pada papan Arduino versi terakhir karena pemilihan sumber daya eksternal atau USB dilakukan secara otomatis.

d. Q1 – Kristal (quartz crystal oscillator)

Jika mikrokontroler dianggap sebagai sebuah otak, maka kristal adalah jantung-nya karena komponen ini menghasilkan detak-detak yang dikirim mikrokontroler agar melakukan sebuah operasi untuk setiap detaknya. Kristal ini dipilih yang berdetak 16 juta kali per detik (16MHz).

e. Tombol Reset

Untuk me-reset papan sehingga program akan mulai lagi dari awal. Perhatikan bahwa tombol reset ini bukan untuk menghapus program atau mengosongkan mikrokontroler.

f. In-Circuit Serial Programming (ICSP)

Port ICSP memungkinkan pengguna untuk memprogram mikrokontroler secara langsung, tanpa melalui bootloader. Umumnya pengguna Arduino tidak melakukan ini sehingga ICSP tidak terlalu dipakai walaupun disediakan.

g. IC 1-Mikrokontroler Atmega

Komponen utama dari papan Arduino, di dalamnya terdapat CPU, ROM dan RAM.

h. X1-Sumber Daya Eksternal

Jika hendak disuplai dengan sumber daya eksternal, papan Arduino dapat diberikan tegangan DC antara 9-12V.

i. 6 pin input analog (0-5)

Pin ini sangat berguna untuk membaca tegangan yang dihasilkan oleh sensor analog, seperti sensor suhu. Program dapat membaca nilai sebuah pin input antara 0-1023, dimana hal itu mewakili nilai tegangan 0-5V.

2.5. Sensor

Sensor merupakan interface kehidupan nyata dengan elektronik seperti komputer. Sensor merupakan sebuah perangkat yang dapat

mengubah fenomena fisis menjadi sinyal elektrik. Dalam penelitian yang akan dilakukan sensor digunakan sebagai input untuk memberi respon terhadap output dari arduino uno sehingga dapat mengontrol konsumsi energi listrik. Sensor yang digunakan dalam penelitian ini adalah sensor suhu, sensor ultrasonik, dan sensor cahaya.

2.5.1. Sensor Suhu DHT-21

Sensor DHT21 adalah sensor suhu dan kelembaban udara. Sensor ini memiliki output digital yang dengan rentang 8 bit. Dalam penggunaannya di dalam mikrokontroller, sensor ini telah diberi fungsi tersendiri pada Arduino IDE untuk memudahkan penggunaannya.

Sensor DHT21 ini memiliki jangkauan pengukuran suhu antara -40°- 80°C dan jangkauan kelembaban udara 0-100% RH. Sensor ini memiliki elemen resistif dan perangkat pengukur suhu NTC. Dengan ukuran yang kecil modul sensor ini mudah untuk diimplementasikan menggunakan mikrokontroller, sensor ini juga memiliki kualitas yang baik, memiliki respon yang cepat, kemampuan anti-gangguan dan biaya yang sangat murah.



Gambar 2.7 Sensor Suhu dan Kelembapan DHT-21

2.5.2. Sensor Ultrasonik HC-SR-04

Menurut Sukarjadi,dkk (2017) Sensor ultrasonik HC-SR-04 adalah sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu. Disebut sebagai sensor ultrasonik karena sensor ini menggunakan gelombang ultrasonik (bunyi ultrasonik).

Secara umum, alat ini akan menembakkan gelombang ultrasonik menuju suatu area atau suatu target. Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh sensor,

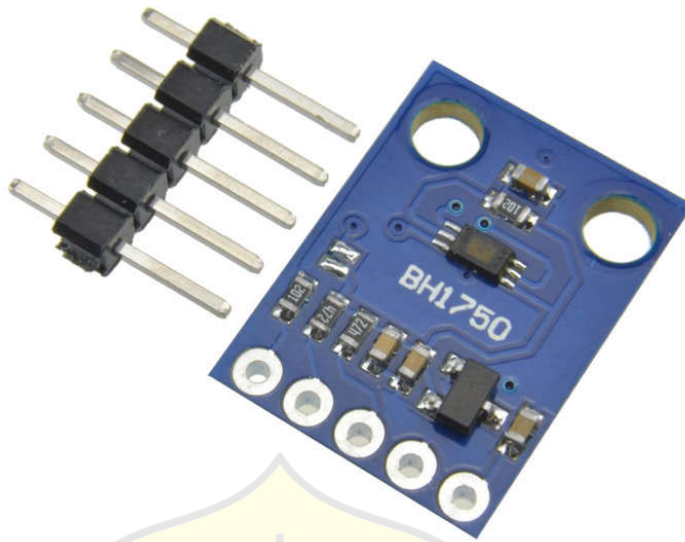
kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima.



Gambar 2.8 Sensor Ultrasonik HC-SR-04

2.5.3. Sensor Cahaya BH1750FVI

Modul sensor intensitas cahaya BH1750 adalah sensor cahaya digital yang memiliki keluaran sinyal digital, sehingga tidak memerlukan perhitungan yang rumit. Sensor BH1750 ini lebih akurat dan lebih mudah digunakan jika dibandingkan dengan sensor lain seperti foto diode dan LDR yang memiliki keluaran sinyal analog dan perlu melakukan perhitungan untuk mendapatkan data intensitas. Sensor cahaya digital BH1750 ini dapat melakukan pengukuran dengan keluaran lux (lx) tanpa perlu melakukan perhitungan terlebih dahulu. Data output dengan sensor ini langsung output di satuan Lux (Lx). Ketika benda-benda yang menyala di honogen mendapatkan 1 lx fluks bercahaya dalam satu meter persegi, intensitas cahaya mereka 1 LX.



Gambar 2.9 Sensor Cahaya BH1750FVI

2.6. Website

Web Site adalah kumpulan dari banyak halaman, biasanya dalam format *HTML (Hypertext Markup Language)*, yang berisi teks, grafis, dan elemen multimedia seperti *flash*, audio, ataupun video. Halaman utama dari sebuah *site* biasanya disebut dengan *Home Page*, berisi tautan menuju ke dokumen lain di site tersebut dengan menggunakan *Hyperlinks*. Semua halaman *web* disimpan pada sebuah *web server*. (Richard Wagner, 2011).

2.6.1. HTML

Sebuah halaman web ditulis dalam *Hypertext Markup Language (HTML)*, sebuah bahasa berbasis tak yang digunakan untuk menyajikan informasi. *HTML* terdiri dari dua tipe data :

- a. *Content*: teks dan grafis yang akan ditampilkan pada halaman *web*.
- b. *Instructions*: elemen format yang terdefinisi, atau *tags*, untuk menentukan bagaimana teks dan grafis akan ditampilkan dan

disusun pada sebuah halaman. Intruksi ini tidak akan tampak saat halaman *web* ditampilkan pada *web browser*. (Richard Wagner, 2011).

2.6.2. Cascading Style Sheet (CSS)

Cascading style sheet (CSS) adalah sebuah bahasa untuk mendeskripsikan tampilan dari sebuah dokumen yang ditulis dalam bahasa *markup* seperti *HTML* misalnya. Dengan CSS maka akan dapat mengatur warna tulisan, bentuk huruf, jarak antar paragraf, ukuran kolom, gambar latar atau warna latar yang akan digunakan, dan berbagai variasi efek visual. Salah satu keuntungan utama dari CSS adalah satu CSS yang sama dapat digunakan lebih dari satu halaman, yang artinya keseluruhan *style* dari satu *website* dapat diatur tanpa harus mengganti di tiap halamannya. CSS biasanya digunakan untuk mengatur tampilan visual dari sebuah halaman *web*, dan dikombinasikan dengan *HTML* atau *XHTML* (digunakan untuk mendeskripsikan konten) dan *JavaScript* (yang digunakan untuk menambahkan kesan interaktif pada sebuah halaman). (Ian Pouncey dan Richard York, 2011).

2.6.3. PHP

Menurut (Budi Raharjo, 2015) dalam buku “Belajar Otodidak Framework Codeigneter Teknik Pemrograman web dengan PHP dan Framework Codeigneter 3” PHP (singkatan dari "*PHP: Hypertext Preprocessor*") adalah bahasa pemrograman umum yang dibuat oleh Rasmus Lerdorf dan dirilis pertama kali pada tahun 1995. Meskipun sebenarnya PHP dapat digunakan untuk membuat aplikasi berjenis

desktop, *Command-Line Interface* (CLI), dan aplikasi *mobile* (berbasis android), tapi pada umumnya PHP lebih banyak digunakan untuk mengembangkan aplikasi berbasis web. Untuk memulai menggunakan PHP, kita perlu membuat file dengan ekstensi .php. file dapat dibuat menggunakan aplikasi Text Editor apa saja, bisa Notepad, Notepad++, dan lain-lain.

2.6.4. Javascript

JavaScript adalah sebuah bahasa pemrograman yang dapat menambahkan animasi, interaksi, dan efek visual dinamis ke *HTML*. *JavaScript* dapat membuat halaman *web* menjadi lebih berarti dengan menambahkan umpan balik yang cepat. Sebagai contoh, *JavaScript* dapat memberikan pesan kesalahan dengan segera saat pengguna melakukan *submit* pada sebuah *web form* namun ada salah informasi yang tidak lengkap. (David Sawyer McFarland, 2011).

2.6.5. JQuery

jQuery adalah librari atau kumpulan kode JavaScript siap pakai. Keunggulan menggunakan jQuery dibandingkan dengan JavaScript standar, yaitu menyederhanakan kode JavaScript dengan cara memanggil fungsi-fungsi yang disediakan oleh jQuery. JavaScript sendiri merupakan bahasa Scripting yang bekerja disisi Client/Browser sehingga website bisa lebih interaktif. (Aloysius Sigit W., 2011).

2.7. Database

Menurut (R.H. Sianipar, 2015) dalam buku “Membangun Web PHP dan MySQL untuk pemula dan programmer” database merupakan kumpulan informasi yang besar. Dengan database, dapat dengan mudah merekam kemudian mengakses sejumlah informasi untuk berbagai kepentingan.

MySQL merupakan database yang paling populer digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengelola datanya. Database *MySQL*, merupakan database yang menjajikan sebagai alternatif pilihan database yang digunakan untuk sistem database personal atau organisasi kita. *Oracle* sebagai database besar telah membuat kit (modul) untuk memudahkan proses migrasi dari *MySQL* ke dalam *Oracle*, hal ini dapat menunjukkan bahwa *Oracle* telah memperhitungkan database *MySQL* sebagai database alternatif masa depan. Demikian juga dengan database *MySQL*, menunjukkan makin banyaknya perusahaan besar menggunakannya (Betha Sidik, 2005).

2.8. UML

Menurut (Rosa A.S. dan M. Shalahuddin, 2015) dalam buku “Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek”. Unified Modelling Language (UML) merupakan bahasa visual untuk permodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan permodelan. Jadi penggunaan UML tidak terbatas pada metodologi

tertentu, meskipun pada kenyataanya UML paling banyak digunakan pada metodologi berorientasi objek.

2.8.1. Use Case




Menurut (Rosa A.S. dan M. Shalahuddin, 2015) dalam buku “Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek”. Diagram *use case* merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut actor dan *use case*.

1. Actor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

Tabel 2.1 Simbol - Simbol Diagram *Use Case* (Rosa A.S dan M. Shalahuddin, 2015)

NO	SIMBOL	KETERANAGAN
1	<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase dengan nama <i>use case</i> .
2	<p>Aktor</p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dengan kata benda di awal frase nama aktor.
3	<p>Asosiasi</p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> memiliki interaksi dengan aktor.
4	<p>Ekstensi</p> <p><<extend>></p>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat


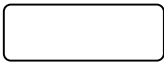


	----->	berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu, mirip dengan prinsip inheritance objek.
--	--------	--


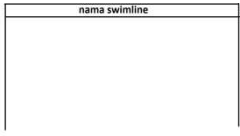
2.8.2. Activity Diagram

Menurut (Rosa A.S. dan M. Shalahuddin, 2015) dalam buku “Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek”. Diagram aktifitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah system atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas yang dapat dilakukan aktor, jadi di aktivitas yang dapat dilakukan oleh system.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Tabel 2.2 Simbol - Simbol Diagram *Activity* (A.S Rosa dan Salahuddin M, 2015)


NO	SIMBOL	KETERANGAN
1	Status Awal 	Status awal aktivitas system, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan/ decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.





5	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.8.3. Sequence Diagram

Menurut (Rosa A.S. dan M. Shalahuddin, 2015) dalam buku “Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek”. Diagram sekuen menggambarkan kelakuan objek pada *use case* dan *message* yang dikirimkan dan di terima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang memiliki kelas yang diintansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat scenario yang ada pada *use case*.

Tabel 2.3 Simbol - Simbol *Sequence Diagram* (A.S Rosa dan Salahuddin M, 2015)

Notasi	Nama	Deskripsi
	Object	Menyatakan objek yang berinteraksi pesan.

	<p>Actor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang, biasanya dinyatakan menggunakan benda di awal frase nama aktor.</p>
	<p>Lifeline</p>	<p>Menyatakan kehidupan suatu objek.</p>
	<p>Activation / Waktu aktif</p>	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.</p>
	<p>Message</p>	<p>Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>

2.8.4. Deployment Diagram

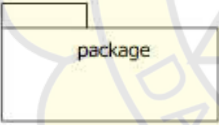
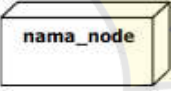
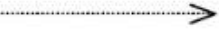

Menurut (Rosa A.S. dan M. Shalahuddin, 2015) dalam buku “Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek”.

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. *Deployment* diagram juga dapat digunakan untuk memodelkan hal-hal berikut :

1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan device, node, dan hardware.
2. Sistem *client/server*
3. Sistem terdistribusi murni
4. Rekayasa ulang aplikasi

Berikut adalah simbol-simbol yang ada pada *deployment* diagram:

Tabel 2.4 Simbol - Simbol *Deployment Diagram* (A.S Rosa dan Salahuddin M, 2015)

Simbol	Deskripsi
Package 	package merupakan sebuah bungkusan dari satu atau lebih <i>node</i>
Node 	biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen
Kebergantungan / dependency 	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai
Link 	relasi antar <i>node</i>