

## **BAB 2**

### **DASAR TEORI**

#### **2.1 KAJIAN PUSTAKA**

Beberapa penelitian terdahulu telah mengkaji metode non-invasif untuk pengukuran gula darah, saturasi oksigen, dan detak jantung sebagai bagian dari upaya meningkatkan pemantauan kesehatan personal. Penelitian-penelitian ini mengeksplorasi kemungkinan implementasi teknologi sensor dan IoT untuk mengukur parameter-parameter kesehatan ini secara akurat tanpa memerlukan tindakan *invasive*.

Pada penelitian sebelumnya [6] yang membahas rancangan alat pendeteksi gula darah serta kolesterol menggunakan metode *reflectance* secara *non-invasive* dengan cara meletakkan jari telunjuk di atas *infrared* dan *photodiode*. Metode penelitian yang dipakai adalah secara *non-invasive* yang memanfaatkan *infrared* sebagai *transmitter* dan *photodiode* sebagai *receiver* yang ada pada sensor MAX30100, selanjutnya data dari sensor diolah langsung oleh mikrokontroler Arduino (ATMega2560) dan hasilnya langsung ditampilkan pada layar. Berdasarkan perhitungan pada hasil penelitian tersebut, keakuratan alat mencapai 98,86%.

Pada penelitian sebelumnya [7] membahas sistem deteksi reflektif saturasi oksigen berbasis MAX30100. Metode *reflectance* digunakan untuk menentukan saturasi oksigen dalam darah, dimana saturasi oksigen (SpO<sub>2</sub>) dapat didefinisikan sebagai rasio tingkat oksigenasi hemoglobin (HbO<sub>2</sub>) di atas kadar total hemoglobin (Hb). Berdasarkan penelitian tersebut dapat disimpulkan bahwa koefisien serapan Hb pada daerah spektrum merah (600nm-700nm) sedangkan koefisien serapan Hb pada daerah spektrum inframerah (810nm-1000nm). Sehingga pada penelitian tersebut menggunakan spektrum *infrared* 660nm. Dari hasil pengujian sistem didapat kesalahan relatif antara nilai terukur dan nilai standar sebesar 3%. Dapat dibuktikan bahwa sistem deteksi saturasi oksigen darah dengan metode *reflectance* menggunakan MAX30100 memiliki kelayakan.

Pada penelitian [10] mengeksplorasi rancang bangun alat pengukur kadar gula darah *non-invasive* dengan menggunakan Arduino Uno. Tujuan utamanya

adalah memberikan kemudahan bagi penderita diabetes dalam memantau kadar gula darah tanpa pengukuran *invasive*. Metode *non-invasive* memanfaatkan penyerapan gelombang pada rentang 750-2500 nm. Perancangan hardware melibatkan rangkaian *power supply*, sensor *infrared*, *photodiode*, *push button*, dan LCD. Data dikumpulkan dengan survei di Poliklinik Universitas Negeri Padang. Alat ini membaca daya yang diterima oleh *photodiode* dari sensor *infrared*, dan hasilnya ditampilkan di LCD setelah diolah menggunakan Arduino Uno. Dengan demikian, alat ini diharapkan memberikan kontribusi dalam pemantauan dan manajemen kadar gula darah bagi penderita diabetes. Berdasarkan pengujian yang dilakukan ditemukan bahwa rata-rata kesalahan sebesar 2,14%, dengan tingkat keakuratan pembacaan mencapai sekitar 97,86%.

Pada penelitian [11] membahas implementasi *fuzzy inference system* untuk pengambilan keputusan dalam diagnosa penyakit DM. Dalam penelitian tersebut, telah diterapkan sistem pendukung keputusan dengan pendekatan *fuzzy inference system* menggunakan metode Tsukamoto. Hasil pengujian menunjukkan bahwa sistem tersebut terbukti mampu mendiagnosa penyakit diabetes melitus dengan tingkat validitas sebesar 96% setelah dilakukan pengujian menggunakan *software* Matlab. Dengan demikian, sistem pendukung keputusan yang dihasilkan dapat digunakan sebagai alat bantu untuk mendiagnosa penyakit DM.

Pada penelitian [12] menyajikan metode untuk mengembangkan teknik *non-invasive* guna memprediksi kadar gula darah berdasarkan sensor cahaya inframerah. Prototipe yang dikembangkan menggunakan sensor jari berbasis LED 940 nm dan *photodiode* untuk mengumpulkan sinyal *photoplethysmography* yang bervariasi sesuai perubahan konsentrasi glukosa. Pada penelitian ini juga digunakan rangkaian untuk pra-proses sinyal, termasuk penguat dan sirkuit filter analog. Arduino UNO dipakai untuk konversi analog-digital. Filter digital *Butterworth* digunakan untuk menghilangkan tren sinyal sensor, kemudian mendeteksi puncak data pembacaan sensor untuk menentukan hubungan antara nilai pembacaan sensor dan gula darah, serta menggunakannya sebagai parameter input untuk membangun model kalibrasi berdasarkan regresi linier. Hasil pengujian menunjukkan bahwa *Root Mean Squares Error* (RMSE) prediksi berada dalam

rentang 8,264 mg/dL hingga 13,166 mg/dL, dengan rata-rata RMSE sekitar 10,44 mg/dL dan koefisien korelasi 0,839.

Pada penelitian [13] mengembangkan alat pengukuran SpO<sub>2</sub> dan BPM. Dalam pembuatan alat pengukuran SpO<sub>2</sub> dan BPM, digunakan sensor MAX30102 dengan metode *photoplethysmography* reflektif dan modul WiFi ESP8266 agar dapat dipantau melalui platform Blynk. Pengukuran dilakukan saat subjek dalam keadaan relaksasi tanpa aktivitas fisik berat. Beberapa faktor yang memengaruhi nilai SpO<sub>2</sub> dan BPM pada alat ini adalah gerakan, aktivitas fisik, intensitas pencahayaan lingkungan, dan noise dari perangkat elektronik lainnya. Alat yang dibuat memiliki keakuratan dengan selisih 0,2% untuk BPM dan 0,4% untuk SpO<sub>2</sub>.

Berdasarkan penelitian-penelitian yang telah dilakukan sebelumnya, maka penelitian ini akan mengkombinasikan beberapa penelitian tersebut diantaranya pengukuran detak jantung, saturasi oksigen, dan gula darah secara *non-invasive* menggunakan sensor MAX30102, serta sistem pengambilan keputusan diagnosa atau rekomendasi kesehatan menggunakan metode *fuzzy* Tsukamoto untuk menghasilkan “Sistem Rekomendasi Kondisi Kesehatan Secara *Non-Invasive* Dengan *Fuzzy* Tsukamoto Berbasis *Internet of Things*”.

## **2.2 DASAR TEORI**

Pada bagian ini, membahas dasar teori yang menjadi landasan konseptual untuk pemahaman lebih lanjut terhadap topik penelitian. Dasar teori ini mencakup konsep-konsep kunci, teori-teori terkait, dan hasil penelitian terbaru yang memiliki relevansi dengan penelitian ini.

### **2.2.1 Gula Darah**

Gula darah merupakan monosakarida yang berfungsi sebagai sumber utama energi dalam tubuh. Karbohidrat kompleks dipecah menjadi glukosa dan monosakarida lainnya melalui proses pencernaan. Transportasi glukosa ke dalam sel melibatkan protein transpor khusus dan membutuhkan energi, serta melibatkan penyerapan ion natrium. Konsentrasi glukosa dalam sirkulasi darah diatur oleh hormon seperti insulin, kortisol, dan glukagon, yang mengatur penyerapan glukosa ke dalam sel dan berbagai proses metabolisme seperti glikolisis, glukoneogenesis,

dan glikogenolisis [14]. Glukosa dalam darah dikendalikan oleh beberapa mekanisme homeostatik yang dalam keadaan sehat dapat menjaga kadar gula dalam rentang 70 sampai 110mg/dL ketika perut kosong, disebut juga dengan gula darah puasa. Setelah pencernaan makanan yang mengandung glukosa, umumnya kadar glukosa akan meningkat, namun tidak melebihi 200mg/dL atau disebut sebagai gula darah sewaktu. Umumnya kadar gula darah sewaktu, normalnya pada angka dibawah 145mg/dL, kategori agak tinggi sampai 180mg/dL, melebihi itu dianggap tinggi [15]. Secara fluktuatif, kadar gula dalam darah bergantung pada asupan makan. Pada dua jam setelah makan, gula darah akan mencapai kadar paling tinggi, namun tidak akan mencapai 180mg per 100cc darah (180mg/dL). Nilai tersebut merupakan ambang batas maksimal yang dapat ditampung oleh ginjal manusia. Jika kadar gula dalam darah lebih tinggi dari nilai tersebut, ginjal tidak dapat menahan gula lagi dan gula akan keluar bersama urin, menyebabkan kondisi yang dikenal sebagai kencing manis atau diabetes melitus. Kondisi ini dapat menyebabkan berbagai komplikasi kesehatan yang serius jika tidak segera ditangani dengan baik [4].

**Tabel 2. 1 Kriteria Kadar Gula Darah [4]**

Kategori	Klasifikasi		
	Baik/Normal	Agak Tinggi	Tinggi
Gula darah puasa	80-109 mg/dL	110-125 mg/dL	≥ 126 mg/dL
Gula darah sewaktu	80-144 mg/dL	145-179 mg/dL	≥ 180 mg/dL

Pada Tabel 2.1 kandungan gula darah puasa dikatakan baik jika berada pada rentang 80-109mg/dL, dikatakan agak tinggi jika berada pada rentang 110-125mg/dL, dan dikatakan tinggi jika melebihi 125mg/dL. Sedangkan kandungan gula darah 2 jam (*post prandial*) dikatakan baik jika berada pada rentang 80-144mg/dL, dikatakan agak tinggi jika berada pada rentang 145-179mg/dL, dan tinggi jika lebih dari sama dengan 180mg/dL [4]. Namun selain itu, apabila kadar gula darah berada di bawah 70mg/dL atau disebut sebagai hipoglikemia, juga cukup berbahaya. Kondisi hipoglikemia dapat meningkatkan kondisi aritmia (kondisi detak jantung yang tidak teratur) pada penderita DM yang tentunya akan sangat membahayakan kesehatan [5].

### **2.2.2 Detak Jantung**

Jantung merupakan organ penting yang terdapat di dalam tubuh manusia. Denyut jantung mengalami fluktuasi yang disesuaikan dengan kondisi tubuh, sehingga tidak dapat dikendalikan secara langsung oleh manusia [16]. Denyut jantung merujuk pada jumlah detak jantung dalam satu periode waktu tertentu. Denyut jantung diukur dengan menggunakan satuan *Beats Per Minute* (BPM), yang merupakan jumlah detak jantung dalam satu menit. Laju detak jantung orang dewasa pada keadaan *relax*(santai) yang normal berkisar antara 60-100BPM. Kelainan detak jantung terjadi ketika detak jantung kurang dari 60BPM yang dikenal sebagai bradikardia(lambat). Selain itu kelainan juga dapat terjadi jika laju detak jantung lebih dari 100BPM yang dikenal sebagai takikardia(cepat) [17].

### **2.2.3 Saturasi Oksigen**

Saturasi oksigen ( $SpO_2$ ) merupakan konsentrasi oksigen dalam darah, yang merupakan parameter fisiologis penting dari sistem pernapasan dan peredaran darah. Kekurangan kadar oksigen dalam darah (hipoksia) dapat menyebabkan gejala seperti ketidakseimbangan perhatian, hilang ingatan, pusing, kecemasan, dan lain-lain. Hipoksia pada miokard sangat sensitif terhadap hipoksia akut, yang dapat menyebabkan fibrilasi ventrikel dan henti jantung. Hipoksia yang parah atau berkepanjangan dapat menyebabkan kegagalan miokard, penurunan tekanan darah, dan kegagalan sirkulasi darah. Yang lebih buruk, hipoksia dapat secara langsung merusak korteks cerebral, menyebabkan degenerasi dan nekrosis jaringan otak.  $SpO_2$  dimodelkan sebagai rasio kadar hemoglobin teroksigenasi ( $HbO_2$ ) dengan total hemoglobin (Hb) [7]. Saturasi oksigen normal adalah 95-100% dalam kedokteran, oksigen saturasi sering digunakan untuk mengukur presentase oksigen yang diikat oleh hemoglobin di dalam aliran darah [18].

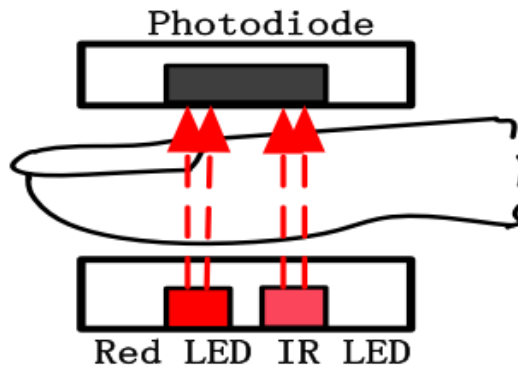
### **2.2.4 Photoplethysmography**

*Photoplethysmography* (PPG) merupakan metode pengukuran yang digunakan untuk mengukur perubahan volume darah pada pembuluh darah yang dekat dengan lapisan kulit manusia berbasis optik. Metode PPG memanfaatkan perubahan gelombang cahaya untuk mendeteksi perubahan volume darah dengan

menggunakan LED dan *photodiode*. Metode PPG umumnya digunakan untuk mengukur jumlah kadar oksigen dalam darah dan jumlah denyut jantung. Pengukuran ini mempunyai dua metode yaitu *transmittance* dan *reflectance* [19].

a. Metode *Transmittance*

Metode *transmittance* dilakukan dengan cara meletakkan jari di antara LED dengan *photodiode*.

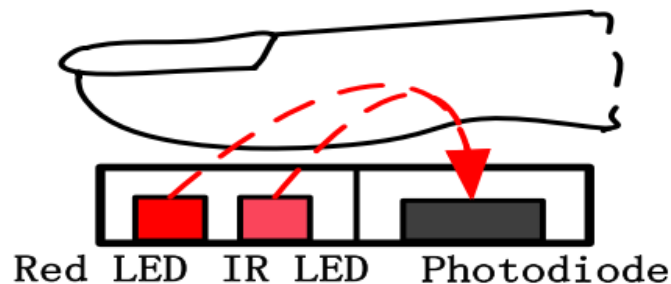


**Gambar 2. 1 Metode *Transmittance*** [7]

Pada Gambar 2.1 metode ini, sumber cahaya akan melewati pembuluh darah untuk mengukur perubahan volume darah yang kemudian cahaya diterima oleh *photodiode*. Kelemahan metode ini terbatas pada area yang dapat diukur, hanya dapat digunakan pada bagian jari tangan, jari kaki, hidung, pipi, lidah, dan daun telinga [6].

b. Metode *Reflectance*

Metode *reflectance* dilakukan dengan cara meletakkan ibu jari diatas LED dan *photodiode* yang berada pada posisi sejajar.

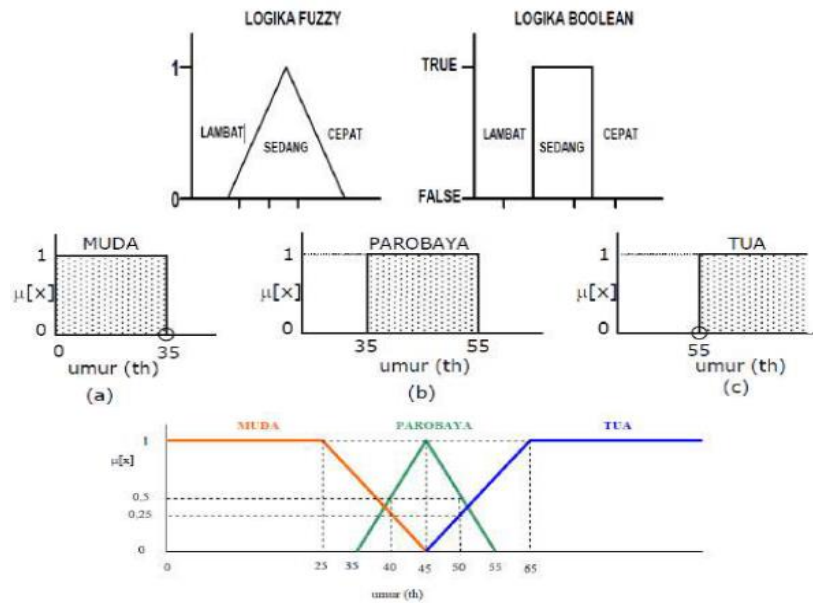


**Gambar 2. 2 Metode *Reflectance*** [7]

Pada Gambar 2.2 sumber cahaya akan melalui pembuluh darah dan dipantulkan dari jaringan tulang dan pembuluh darah menuju *photodiode*. Metode ini dapat diaplikasikan pada semua bagian kulit tubuh yang terdapat pembuluh darah [6].

### 2.2.5 Fuzzy Logic

*Fuzzy logic* atau logika *fuzzy* merupakan metodologi sistem kontrol yang digunakan untuk mengatasi berbagai masalah, dan cocok untuk diimplementasikan dalam sistem-sistem seperti sistem sederhana, sistem terbenam (*embedded system*), sistem kecil, jaringan PC, *multi-channel* atau *workstation* berbasis akuisisi data, serta sistem kontrol. Metodologi ini dapat diterapkan baik dalam perangkat keras, perangkat lunak, maupun kombinasi keduanya [8].



**Gambar 2. 3 Ilustrasi Logika Fuzzy [8]**

Pada gambar 2.3 merupakan ilustrasi logika *fuzzy*. Dalam logika klasik, segala hal diungkapkan secara biner, seperti "Benar atau Salah", "Ya atau Tidak", dan sebagainya. Oleh karena itu, setiap hal hanya dapat memiliki nilai keanggotaan 0 atau 1. Namun, dalam logika *fuzzy*, nilai keanggotaan dapat berada di antara 0 dan 1. Hal ini berarti suatu keadaan dapat memiliki dua nilai yang berbeda secara bersamaan, seperti "Ya dan Tidak", "Benar dan Salah", namun nilai-nilai tersebut memiliki bobot keanggotaan yang berbeda [8].

#### 2.2.5.1 Cara Menuliskan Himpunan Fuzzy

Suatu himpunan *fuzzy*  $A$  dalam semesta pembicaraan dinyatakan dengan fungsi keanggotaan (*membership function*)  $\mu_A$ , yang harganya berada dalam interval  $[0,1]$  [8]. Secara matematika hal ini dinyatakan dengan:

$$\mu_A = U \rightarrow [0,1] \quad (2.1)$$

### 2.2.5.2 Operasi Dasar *Fuzzy Set*

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan *fuzzy*. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama *fire strength* atau  $\alpha$ -predikat. Ada 3 operator dasar yang diciptakan, diantaranya operator AND, operator OR, dan operator NOT [8].

#### a. Operator AND

Operator ini berhubungan dengan operasi interseksi pada himpunan.  $\alpha$ -predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan [8].

$$\mu A \cap B = \min(\mu A[x], \mu B[y]) \quad (2.2)$$

#### b. Operator OR

Operator ini berhubungan dengan operasi union pada himpunan.  $\alpha$ -predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan [8].

$$\mu A \cup B = \max(\mu A[x], \mu B[y]) \quad (2.3)$$

#### c. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan.  $\alpha$ -predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangi nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1 [8].

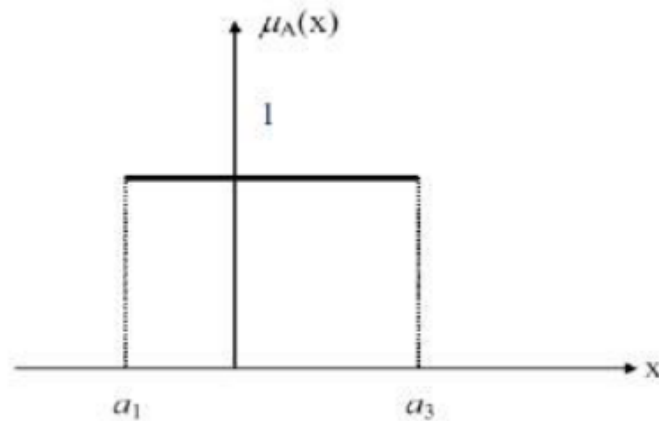
$$\mu A = 1 - \mu A[x] \quad (2.4)$$

### 2.2.5.3 Interval *Fuzzy*

Selama batas dari interval adalah ambigu/kabur, intervalnya juga merupakan sebuah himpunan *fuzzy*. Suatu fungsi keanggotaan (*membership function*) dalam bentuk interval dinyatakan [8] :

$$\mu A[x] = \begin{cases} 0, & x < a1 \\ 1, & a1 \leq x \leq a3 \\ 0, & x \geq a3 \end{cases} \quad (2.5)$$





**Gambar 2. 4 Interval Fuzzy [12]**

Pada gambar 2.4 merupakan grafik yang dihasilkan dari persamaan 2.5. Dimana jika  $x$  kurang dari  $a_1$  atau lebih dari  $a_3$ ,  $y$  akan bernilai 0, sedangkan apabila nilai  $x$  berada diantara  $a_1$  dan  $a_1$ ,  $y$  akan bernilai 1.

#### 2.2.5.4 Fungsi Implikasi Fuzzy

Tiap-tiap aturan (proposisi) pada basis pengetahuan *fuzzy* akan berhubungan dengan suatu relasi *fuzzy*. Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah:

$$\text{IF } x \text{ is } A \text{ THEN } y \text{ is } B$$

Dimana  $x$  dan  $y$  adalah skalar, dan  $A$  dan  $B$  adalah himpunan *fuzzy*. Proposisi yang mengikuti IF disebut sebagai anteseden, sedangkan proposisi yang mengikuti THEN disebut sebagai konsekuen [8].

#### 2.2.6 Fungsi Keanggotaan

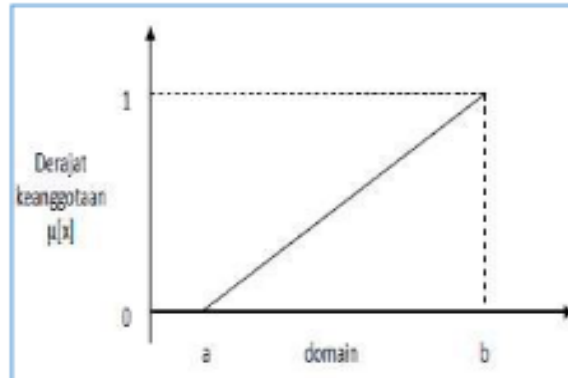
Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya. Sering juga disebut dengan derajat keanggotaan yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi [8]. Pada penelitian ini akan digunakan 3 fungsi keanggotaan, yaitu kurva linier, kurva segitiga, dan kurva bahu.

##### a. Kurva Linier

Pada kurva linear, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Ada 2 keadaan himpunan *fuzzy* yang linear yaitu linier naik dan linier turun [8].

Fungsi Keanggotaan Linier Naik :

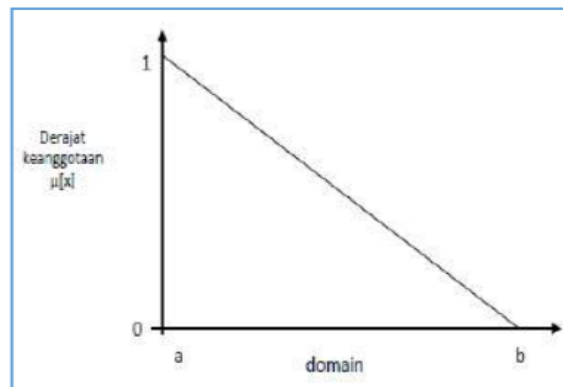
$$\mu_A[x] = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & x \geq b \end{cases} \quad (2.6)$$



**Gambar 2. 5 Kurva Linier Naik [8]**

Fungsi Keanggotaan Linier Turun :

$$\mu_A[x] = \begin{cases} 0, & x \geq b \\ \frac{b-x}{b-a}, & a < x < b \\ 1, & x \leq a \end{cases} \quad (2.7)$$



**Gambar 2. 6 Kurva Linier Turun [8]**

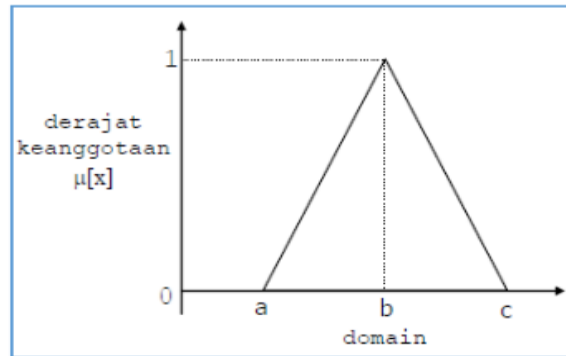
b. Kurva Segitiga

Kurva segitiga, seperti namanya, merupakan kurva yang membentuk bidang segitiga. Kurva ini terbentuk dari gabungan dua kurva linear, yaitu kurva linear turun dan kurva linear naik. Pada bagian awal, kurva segitiga memiliki kemiringan yang positif dan cenderung naik secara linear. Kemudian, pada titik tertentu, kurva segitiga mencapai puncaknya dan berubah arah menjadi menurun secara linear. Dalam hal ini, garis lurus yang naik dan garis

lurus yang turun membentuk sudut tajam, menciptakan bentuk segitiga pada kurva tersebut [8].

Fungsi Keanggotaan :

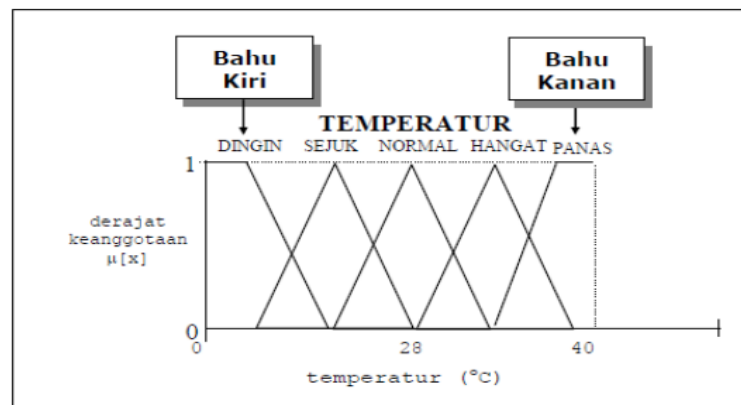
$$\mu A[x] = \begin{cases} 0, & x \leq a \cup x \geq c \\ (x - a)/(b - a), & a < x < b \\ 1, & x = b \\ (c - x)/(c - b), & b < x < c \end{cases} \quad (2.8)$$



**Gambar 2. 7 Kurva Segitiga [8]**

c. Kurva Bahu

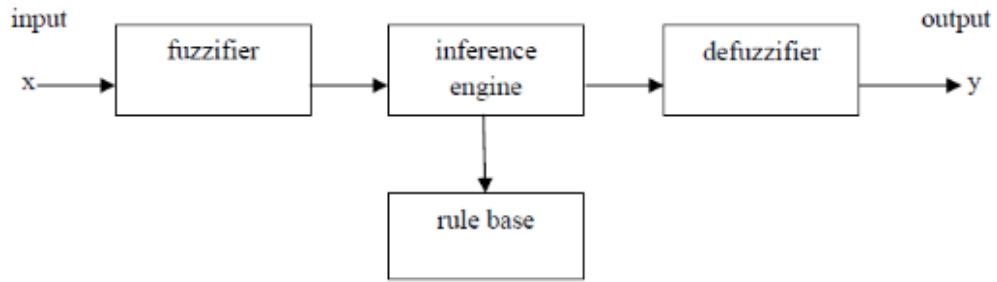
Kurva bahu merupakan daerah yang memiliki sisi kanan dan kiri yang mengalami kenaikan dan penurunan (misalnya: dari DINGIN menjadi SEJUK menjadi HANGAT dan menjadi PANAS). Namun, terkadang salah satu sisi dari variabel tersebut tetap stabil. Sebagai contoh, setelah mencapai kondisi PANAS, kenaikan suhu akan tetap berada pada kondisi PANAS [8].



**Gambar 2. 8 Kurva Bahu [8]**

**2.2.7 Fuzzy Inference System (FIS)**

*Fuzzy Inference System (FIS)* merupakan sebuah framework komputasi populer berdasarkan pada konsep teori himpunan *fuzzy*, aturan *if - then fuzzy*, dan penalaran *fuzzy*.



**Gambar 2. 9 Model Umum FIS [8]**

Pada Gambar 2.9 merupakan model umum FIS. Sistem inferensi FIS pada dasarnya mendefinisikan pemetaan nonlinear dari vektor data input menjadi *scalar output*. Proses pemetaan pada Gambar 2.9 melibatkan input /output fungsi keanggotaan, operator-operator *fuzzy*, aturan *fuzzy if - then*, agregasi dari himpunan output dan *defuzzification* [8].

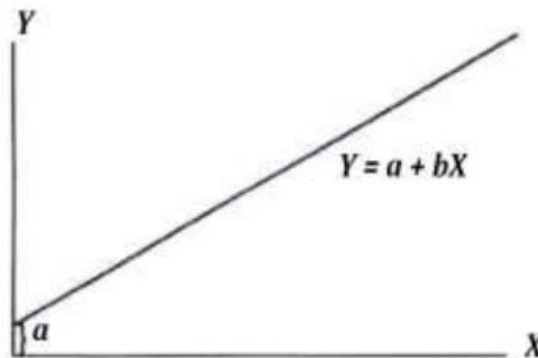
### 2.2.8 Metode Tsukamoto

Pada logika *fuzzy* metode Tsukamoto, setiap akibat pada aturan yang berbentuk *IF-THEN* perlu diwakili oleh himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Hasil inferensi dari setiap aturan diberikan secara tegas (*crisp*) berdasarkan  $\alpha$ -predikat (*fire strength*). Hasil akhir diperoleh dengan mengambil rata-rata terbobot. Kelebihan metode ini terletak pada kemampuannya dalam menangani ketidakpastian dan ambiguitas dalam pengambilan keputusan, serta kemampuannya dalam memodelkan aturan-aturan yang kompleks secara linguistik [20]. Logika *fuzzy* dapat diterapkan dalam berbagai bidang, seperti sistem diagnosa penyakit (dalam bidang kedokteran), riset operasi (dalam bidang ekonomi), pemodelan sistem pemasaran, prediksi gempa bumi, kendali kualitas air, klasifikasi, dan pencocokan pola (dalam bidang teknik), dan banyak lagi [21].

### 2.2.9 Regresi Linier Sederhana

Regresi linear sederhana merupakan teknik statistik yang digunakan untuk menganalisis hubungan antara dua variabel. Teknik ini disebut "sederhana" karena hanya melibatkan satu variabel independen dan satu variabel dependen. Variabel independen juga dikenal sebagai variabel prediktor, sedangkan variabel dependen adalah variabel yang diprediksi atau dijelaskan. Regresi linear sederhana umumnya digunakan dalam berbagai bidang seperti ekonomi, keuangan, pemasaran, dan ilmu

lingkungan, untuk beberapa contoh. Ini adalah alat yang berguna untuk memprediksi hasil masa depan berdasarkan data masa lalu [22].



**Gambar 2. 10 Bentuk Grafik Regresi Linier [22]**

$$Y = a + bX \quad (2.9)$$

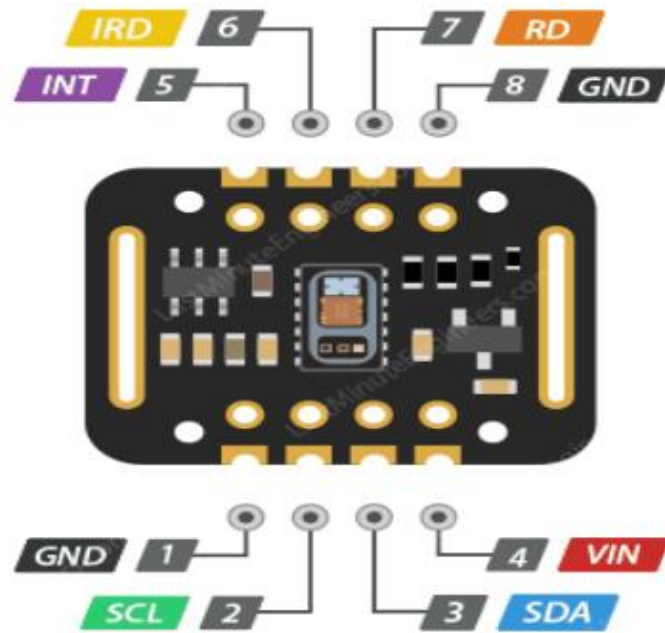
$$a = \frac{(\sum Y) (\sum X^2) - (\sum X) (\sum XY)}{n(\sum X^2) - (\sum X)^2} \quad (2.10)$$

$$b = \frac{n(\sum XY) - (\sum X) (\sum Y)}{n(\sum X^2) - (\sum X)^2} \quad (2.11)$$

Pada Gambar 2.10 tujuan utama dari regresi linear sederhana adalah untuk menemukan hubungan linear antara variabel independen dan variabel dependen. Hubungan ini direpresentasikan oleh garis lurus yang paling cocok dengan data. Persamaan dari garis ini dikenal sebagai persamaan regresi, yang dapat digunakan untuk memprediksi nilai variabel dependen berdasarkan nilai variabel independen. Persamaan regresi memiliki bentuk  $Y = a + bX$ , di mana  $Y$  adalah variabel dependen,  $X$  adalah variabel independen,  $a$  adalah intercept (nilai  $Y$  ketika  $X = 0$ ), dan  $b$  adalah slope (perubahan  $Y$  untuk setiap perubahan satu unit dalam  $X$ ). Slope dari garis ini menunjukkan kekuatan dan arah hubungan antara kedua variabel [23].

### 2.2.10 Pulse Sensor

*Pulse* sensor merupakan modul sensor yang sangat sensitif dan akurat yang mendeteksi cahaya yang ditransmisikan melalui pembuluh darah di ujung jari untuk mengukur saturasi oksigen dan detak jantung per menit. Untuk berkomunikasi dengan mikrokontroler, sensor ini berkomunikasi dengan menggunakan protokol komunikasi I2C yang mana hanya memerlukan dua jalur yaitu Serial Data (SDA) dan Serial *Clock* (SCL) [24].



**Gambar 2. 11 Pulse Sensor** [13]

Pada Gambar 2.11 sensor ini terdapat *Light Emitting Diode* (LED) internal yang mana LED *red* dengan spektrum 660nm dan LED *infrared* dengan spektrum 880nm, selain itu juga terdapat *photodetector*, elemen optik, dan internal *Analog to Digital Converter* (ADC) dengan resolusi maksimal 18bit. Dengan resolusi ADC yang tinggi sebesar 18bit dengan maksimal pembacaan ADC 262144, dengan demikian pembacaan dapat lebih akurat [25].

### 2.2.11 Mikrokontroler

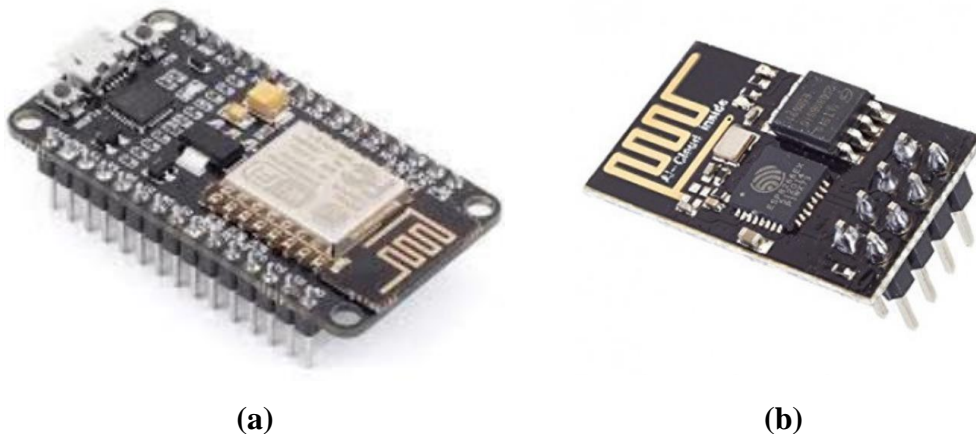
Mikrokontroler merupakan sebuah alat komputasi mini yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya berisi mikroprosesor, memori, dan perlengkapan input-*output*. ESP8266 merupakan sebuah mikrokontroler atau *embedded chip* yang dirancang untuk komunikasi nirkabel berbasis WiFi. *Chip* ini dilengkapi dengan *output* serial TTL dan GPIO. ESP8266 dapat digunakan secara mandiri (*standalone*) sebagai mikrokontroler atau dapat diintegrasikan dengan pengendali lain seperti mikrokontroler lainnya. *Chip* ESP8266 memiliki kemampuan jaringan yang lengkap dan dapat berfungsi sebagai *client* maupun sebagai *Access Point*. *Firmware* yang tersedia untuk ESP8266 sangat beragam, dan chip ini juga dapat diprogram sesuai dengan kebutuhan khusus, misalnya untuk komunikasi dengan situs *web* yang menggunakan protokol HTTPS [26].

Mikrokontroler ESP8266 sendiri menggunakan *System on Chip* (SoC) dengan tipe ESP8266EX yang memiliki spesifikasi seperti pada Tabel 2.2.

**Tabel 2. 2 Spesifikasi ESP8266 [27]**

<b>Mikrokontroler</b>	ESP8266
<b><i>Clock Speed</i></b>	80~160 MHz
<b>Tegangan Input</b>	3.3~3.6V
<b>GPIO</b>	17 Pin
<b><i>Flash Memory</i></b>	4 MB
<b>RAM</b>	32KB+80KB
<b>Konsumsi Daya</b>	10uA~170mA
<b>USB Port</b>	Micro USB
<b>WiFi</b>	2.4 GHz – 22.5 Ghz
<b>Kanal PWM</b>	10 Kanal

Pada Tabel 2.2 merupakan spesifikasi umum ESP8266. *Chip* ESP8266 dibekali oleh prosesor 32-bit Tensilica's seri L106 Diamond. ESP8266 dapat berfungsi sebagai klien yang terhubung ke router WiFi, yang membutuhkan pengaturan nama dan kata sandi dari access point. Selain itu, ESP8266 juga dapat digunakan sebagai *Access Point*, yang memungkinkannya menerima koneksi WiFi [26].



**Gambar 2. 12 (a) NodeMCU, (b) ESP-01 [28]**

Gambar 2.12 merupakan varian modul ESP8266 yang beredar di pasaran, dimana perbedaan dari modul tersebut terdapat pada *form factor* dan *chip* serial yang digunakan, tergantung dari produsen yang membuat, namun pada dasarnya sama sama menggunakan *chip* ESP8266EX.

### 2.2.12 Display OLED

SSD1306 merupakan *display Organic Light-Emitting Diode* (OLED) yang terdiri dari satu unit. *Display* OLED SSD1306 ini memiliki ukuran kecil, sekitar 1", namun mudah dibaca karena memiliki kontras tinggi pada layar OLED.



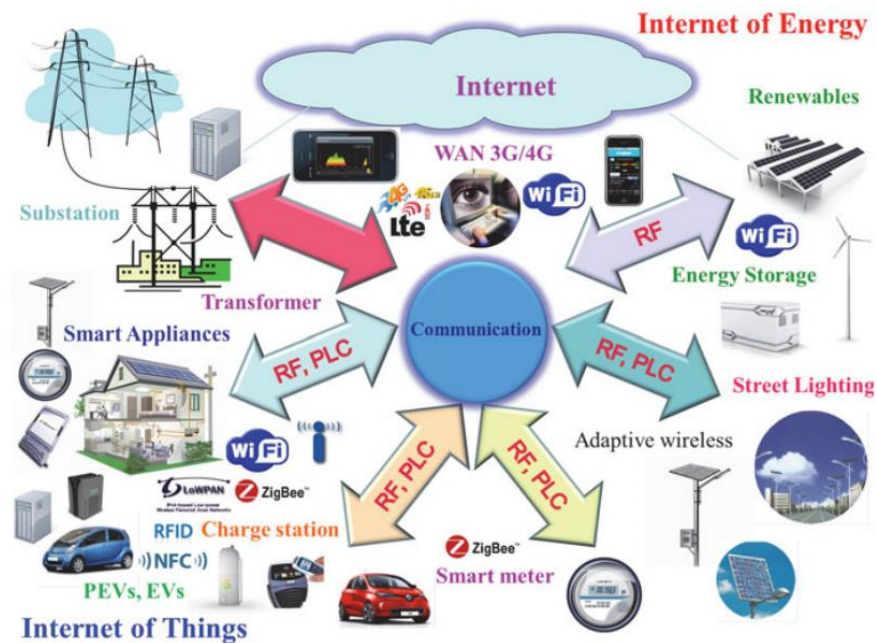
**Gambar 2. 13** *Display* OLED SSD1306 [6]

Pada gambar 2.13 merupakan *Display* yang terdiri dari 128x64 piksel OLED putih individu, yang dapat dihidupkan atau dimatikan oleh *chip controller*. IC ini dirancang untuk panel OLED dengan jenis katoda umum. Karena OLED ini menghasilkan cahaya sendiri, tidak memerlukan backlight, sehingga konsumsi daya OLED ini sangat efisien. OLED ini juga dilengkapi dengan kontrol kecerahan yang dapat diatur hingga 256 langkah. Data atau perintah dikirim dari *Microcontroller Unit* (MCU) melalui antarmuka paralel yang kompatibel dengan seri 6800/8000, antarmuka I2C, atau antarmuka *Serial Peripheral Interface* (SPI). OLED ini sangat cocok digunakan dalam berbagai aplikasi *portabel* seperti *sub-display* ponsel, pemutar MP3, kalkulator, dan lain-lain [6].

### 2.2.13 Internet of Things

*Internet of Things* (IoT) merupakan istilah yang mengacu pada keberadaan objek-objek yang dapat diidentifikasi dan diwakili secara virtual di dalam struktur Internet. Tujuan utama dari konsep IoT adalah untuk memungkinkan semua hal terhubung secara terus-menerus, di mana pun dan kapan pun, menggunakan berbagai jalur atau jaringan serta layanan yang tersedia [29].



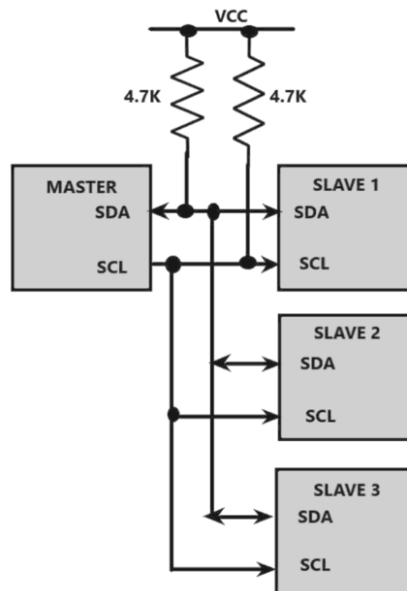


**Gambar 2. 14 Implementasi IoT dalam Kehidupan [29]**

Pada Gambar 2.14 menunjukkan bahwa IoT memberikan kontribusi di berbagai bidang. Melalui implementasi IoT, manajemen dan pemantauan distribusi listrik dapat menjadi lebih efisien. IoT memungkinkan kendaraan terhubung untuk meningkatkan keamanan, efisiensi, dan kenyamanan. IoT memungkinkan perangkat rumah tangga cerdas yang dapat dihubungkan ke jaringan dan diatur secara otomatis. Dengan menerapkan IoT pada bidang-bidang ini, kita dapat mencapai tingkat integrasi yang lebih tinggi, efisiensi operasional, dan memberikan solusi yang lebih adaptif terhadap kebutuhan.

### 2.2.13.1 Komunikasi I2C

*Inter-Integrated Circuit* atau disebut I2C merupakan protocol komunikasi *hardware* yang ditemukan oleh Phillips pada tahun 1982. Protokol I2C memungkinkan komunikasi antara *multiple slaves* ke *single master* maupun *multiple slaves* ke *multiple master*. Protokol I2C memungkinkan untuk banyak perangkat dapat berkomunikasi melalui satu jalur/*bus* yang sama. Yang menjadi pembeda untuk masing-masing perangkat tersebut ialah alamatnya yang mana dijadikan sebagai identitas untuk mengidentifikasi dari masing-masing komponen maupun perangkat yang terhubung walaupun terhubung secara paralel.

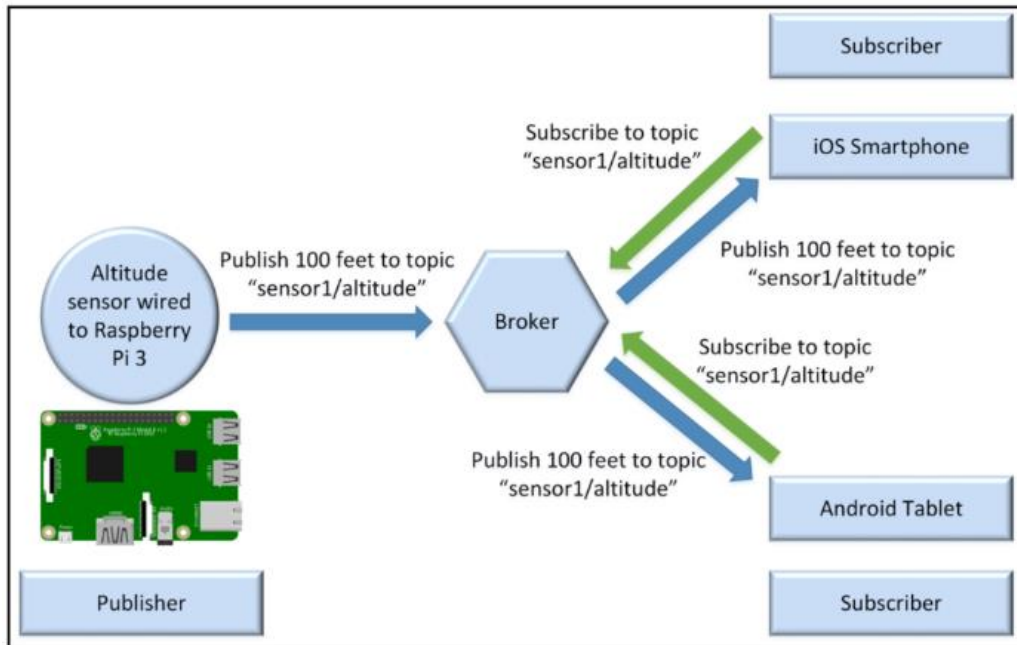


**Gambar 2. 15 Komunikasi I2C dengan *Multiple Slave* [30]**

Pada Gambar 2.15 protokol komunikasi I2C menggunakan dua sinyal, yaitu serial data (SDA) dan serial *clock* (SCL). Jalur serial data (SDA) digunakan oleh *slave* maupun *master* untuk mengirim dan menerima data, sedangkan untuk jalur serial *clock* (SCL) digunakan oleh *master* untuk *clock*. Implementasi komunikasi dengan protokol I2C dapat dilihat pada Gambar 2.19 dimana terdapat *single master* dengan *multiple slave* [30].

### 2.2.13.2 Protokol MQTT

*Message Queuing Telemetry Transport* (MQTT) merupakan sebuah protokol komunikasi IoT yang ringan. Protokol ini bekerja dengan mekanisme *broker* serta *publish-subscribe* yang berjalan diatas protokol *Transmission Control Protocol/Internet Protocol* (TCP/IP). Protokol MQTT dinilai lebih ringan dibandingkan dengan protokol HTTP karena mendukung komunikasi berkelanjutan (*always connected*). Pola *publish-subscribe* membutuhkan suatu *broker*, yang juga dikenal sebagai server. Semua *client* menjalin koneksi dengan *broker*. *Client* yang mengirim pesan melalui *broker* dikenal sebagai *publisher*. *Broker* menyaring pesan-pesan masuk dan mendistribusikannya kepada *client* yang tertarik dengan jenis pesan yang diterima. *Client* yang mendaftar ke *broker* sebagai yang tertarik pada jenis pesan tertentu dikenal sebagai *subscriber*. Oleh karena itu, baik *publisher* maupun *subscriber* saling terkoneksi dengan *broker* [31].



**Gambar 2. 16 Ilustrasi Protokol MQTT [31]**

Pada Gambar 2.16 *publisher* berkomunikasi dengan broker, yang kemudian meneruskan pesan-pesan yang relevan kepada *subscriber* yang tertarik untuk menerimanya. Model ini memfasilitasi komunikasi yang efisien dan memisahkan *publisher* dari *subscriber*, memungkinkan sistem yang dapat diperluas dan fleksibel.

Pada protokol MQTT, kaitannya dengan *Quality of Service* (QoS) mengacu pada level layanan mekanisme pengiriman layanan. Pada protokol MQTT terdapat 3 level QoS yang meliputi :

1. *At most once* (QoS 0)

Pada jenis QoS ini, *publisher* hanya akan melakukan pengiriman data tanpa memerlukan *acknowledgment* dari *broker* bahwa pesan sudah diterima. Jenis layanan ini biasa disebut dengan “*fire and forget*” yang mana pesan hanya akan dikirim sekali walaupun *broker* tidak menerima pesan tersebut.

2. *At least once* (QoS 1)

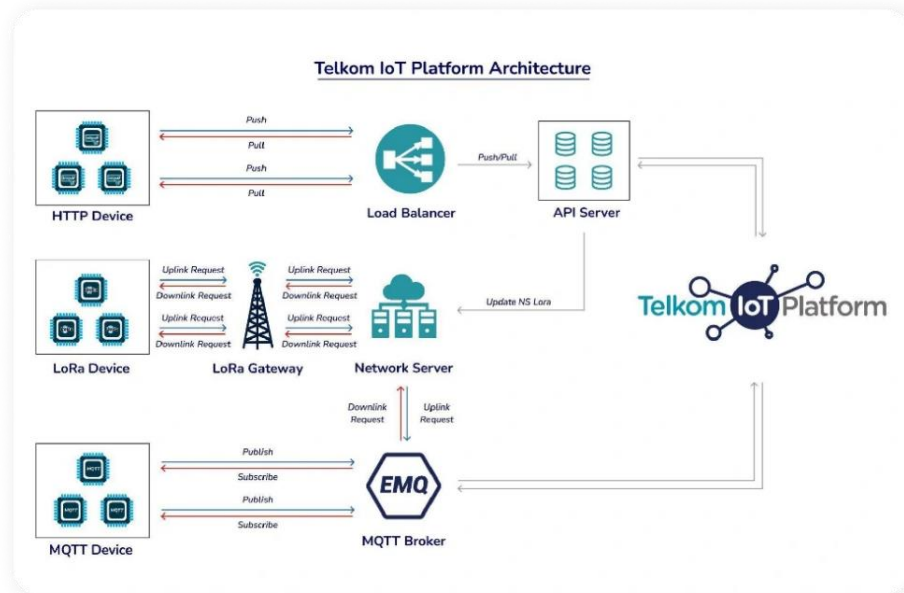
Pada level layanan ini, ketika *publisher* mengirimkan pesan ke *broker*, *publisher* akan menunggu *acknowledgment* dari *broker* apakah pesan sudah diterima. Jika dalam jangka waktu yang ditentukan *broker* tidak mengirimkan *acknowledgment* bahwa pesan diterima, maka *publisher* akan mengirimkan ulang pesan tersebut.

### 3. *Exactly once* (QoS 2)

QoS ini memiliki kelebihan dimana dapat memastikan bahwa data benar-benar berhasil dikirimkan, tanpa mengalami duplikasi terhadap data yang sama. Pada layanan jenis ini baik *publisher* maupun *broker* saling mengkonfirmasi bahwa pesan benar-benar berhasil dikirimkan [31].

#### 2.2.14 Telkom IoT

Telkom IoT merupakan platform *Internet of Things* (IoT) yang dikembangkan oleh Telkom, Indonesia *Telecommunication & Digital Research Institute* (ITDRI), menyediakan arsitektur lengkap untuk memfasilitasi pengembangan dan implementasi solusi IoT bagi berbagai industri di Indonesia. Platform ini terdiri dari lapisan perangkat, lapisan platform yang menyediakan layanan inti seperti manajemen perangkat, pengumpulan data, analitik, dan integrasi, serta lapisan aplikasi yang memungkinkan pengembangan berbagai aplikasi IoT. Telkom IoT mendukung berbagai protokol konektivitas diantaranya HTTP, MQTT, dan *Long Range Wide Area Network* (LoRaWAN) [32].

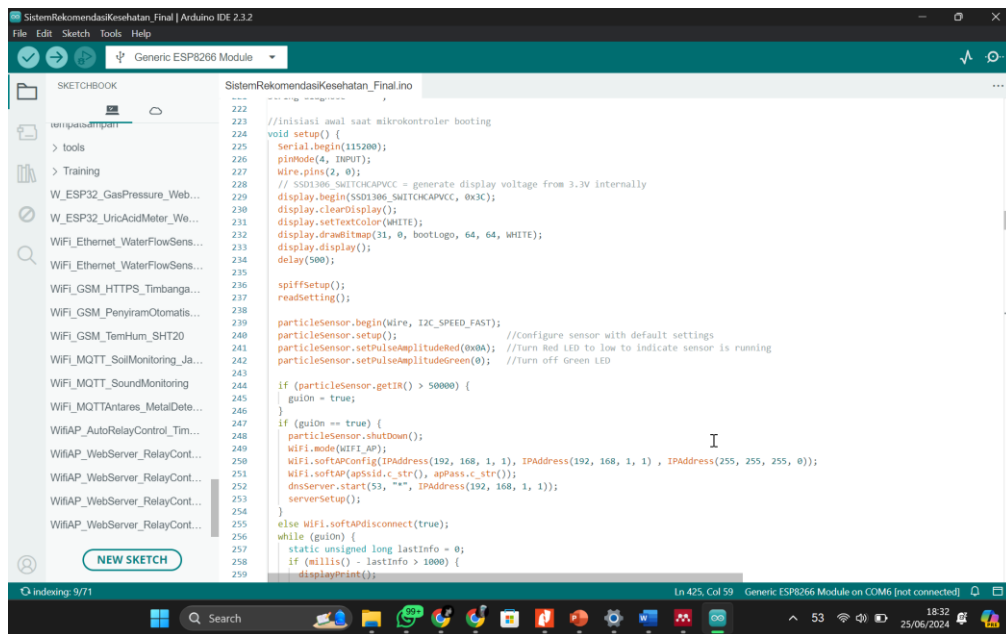


**Gambar 2. 17 Arsitektur LayananTelkom IoT [32]**

Pada Gambar 2.17 merupakan arsitektur layanan yang disediakan oleh Telkom IoT. Platform ini menyediakan tiga layanan konektivitas untuk perangkat HTTP, MQTT, serta LoRaWAN, yang dapat terintegrasi dengan platform Telkom IoT.

### 2.2.15 Arduino IDE

Arduino IDE merupakan *Integrated Development Environment* (IDE), seperti yang tertera pada Gambar 2.18, yang pada dasarnya merupakan perangkat lunak yang digunakan untuk melakukan pengembangan. Disebut sebagai "lingkungan" karena melalui perangkat lunak ini, Arduino atau mikrokontroler lainnya dapat diprogram untuk menjalankan fungsi-fungsi yang telah ditanamkan melalui sintaks pemrograman [33].



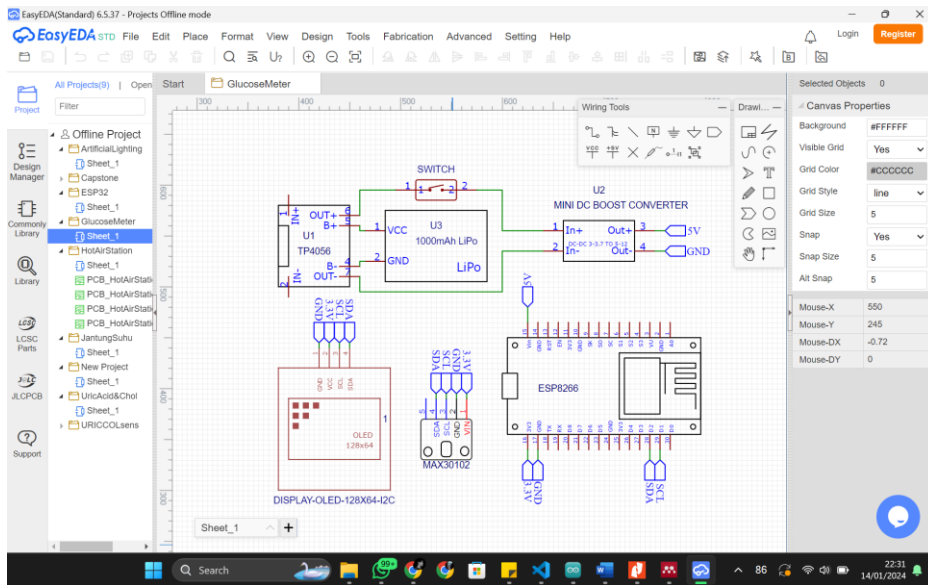
**Gambar 2. 18 Tampilan Arduino IDE**

Pada Gambar 2.18 ialah tampilan dari *software* Arduino IDE. Pada perangkat lunak Arduino IDE, terdapat banyak fitur yang dapat memudahkan proses pengembangan program untuk mikrokontroler. Arduino menggunakan bahasa pemrograman khusus yang mirip dengan bahasa C. ESP8266 memiliki *bootloader* yang sama dengan Arduino yang bertindak sebagai *compiler*, sehingga kompatibel untuk membuat dan mengunggah program ke NodeMCU [33].

### 2.2.16 EasyEDA

EasyEDA, *Electronic Design Automation* (EDA) merupakan perangkat lunak desain yang digunakan untuk mengembangkan rangkaian elektronika. *Software* ini menawarkan antarmuka yang sederhana dan mudah digunakan serta menyediakan berbagai pustaka komponen yang lengkap. Selain mampu membuat skema rangkaian, EasyEDA juga memiliki fitur desain 3D yang memungkinkan

pengguna untuk melihat tampilan visual rangkaian dalam bentuk yang sebenarnya [33].



**Gambar 2. 19 Tampilan EasyEDA**

Pada Gambar 2.19 merupakan tampilan dari *software* EastEDA. Pada perangkat lunak EastEDA ada banyak fitur yang mendukung kaitannya dengan mendesain rangkaian elektronik.

### 2.2.17 Baterai *Lithium Ion*

Baterai *lithium ion* merupakan jenis baterai yang dapat digunakan secara berulang apabila kapasitasnya habis pakai, maka pengguna dapat melakukan pengisian ulang daya. Baterai *litium ion* merupakan sistem elektrokimia yang dinamis dan mengalami perubahan seiring waktu, menunjukkan perilaku nonlinier dan mekanisme internal yang kompleks. Seiring bertambahnya jumlah siklus pengisian-pengurasan, kinerja dan masa pakai baterai *litium ion* juga secara perlahan mengalami penurunan [34].

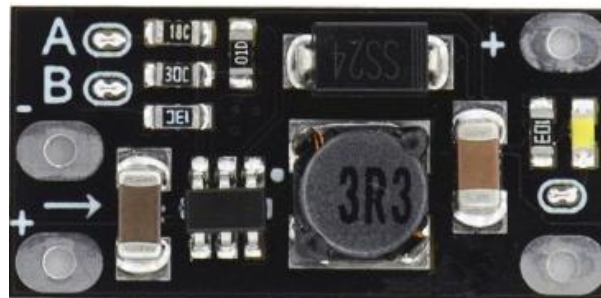


**Gambar 2. 20 Baterai *Lithium Single Cell* [35]**

Pada Gambar 2.20 merupakan contoh dari baterai *lithium single cell*. Tegangan nominal pada baterai *lithium* didesain pada tegangan 3.7V. Tegangan *cut-off discharging* merupakan parameter penting yang terkait dengan masa pakai baterai *litium ion*. Biasanya produsen baterai memberikan *discharge cut-off* pada rentang 2.5-2.75V namun tidak akan lebih rendah dari 2.4V karena walaupun baterai masih menyimpan daya namun hal tersebut dapat merusak baterai, bahkan ada produsen yang memberikan *cut-off* pada 3.0V untuk faktor kesehatan baterai. Sedangkan untuk *charging cut-off* berada pada tegangan 4.2V. Kemudian tegangan pengisian yang disarankan sebesar 1.2 kali dari desain tegangan atau sebesar 4.4V [34].

### 2.2.18 Mini DC Boost Converter

*Mini DC Boost Converter* merupakan sebuah perangkat kecil yang berfungsi sebagai konverter tegangan DC dengan metode "*Boost*". Perangkat ini dapat meningkatkan tegangan masukan menjadi tegangan keluaran yang lebih tinggi [36].



**Gambar 2. 21 Mini DC Boost Converter** [36]

Pada Gambar 2.21 merupakan modul *boost converter* yang digunakan. Pada modul ini, tegangan keluaran dapat diatur pada nilai 5V, 8V, 9V, atau 12V. Pengaturan tegangan keluaran dilakukan dengan mengubah konfigurasi sambungan pada dua pad yang terdapat di bagian depan PCB, yang diberi label A dan B. Rentang tegangan masukan yang diperbolehkan adalah antara 2,5V hingga nilai tegangan keluaran dikurangi 0,5V [36].

### 2.2.19 Akurasi Sensor

Pengertian akurasi sensor sendiri, mengacu pada seberapa mendekati hasil pengukuran sensor jika dibandingkan dengan alat ukur pembanding yang dijadikan



acuan. Dalam hal semakin tinggi nilai akurasi artinya nilai pembacaan sensor semakin mendekati dengan hasil pembacaan alat pembanding [37]. Sebelum menghitung akurasi sensor diperlukan perhitungan nilai *error* terlebih dahulu. Berikut ini merupakan rumus untuk mencari nilai *error* dan akurasi sensor [38].

$$Error = \left| \frac{(\text{pembacaan sensor} - \text{alat acuan})}{\text{alat acuan}} \right| \times 100\% \quad (2.12)$$

$$Akurasi = 100\% - Error \quad (2.13)$$

Nilai *error* didapatkan dari hasil mutlak perbandingan antara nilai pembacaan sensor dengan nilai acuan. Presentase akurasi didapatkan dari hasil pengurangan 100% dikurangi presentase *error*.

### 2.2.20 Presisi Sensor

Presisi sendiri mengacu pada seberapa konsisten hasil pengukuran sensor yang dilakukan secara berulang pada satu kondisi yang sama [37]. Untuk mencari nilai presisi dari sensor diperlukan perhitungan untuk Standar Deviasi (SD) dan Relatif Standar Deviasi (RSD) dari hasil pengukuran sensor. Berikut ini merupakan rumus untuk menghitung nilai SD, RSD, dan presisi sensor [38].

$$SD = \sqrt{\frac{\sum_{i=1}^n (x - \bar{x})^2}{n - 1}} \quad (2.14)$$

$$RSD = \frac{SD}{\bar{x}} \times 100\% \quad (2.15)$$

$$Presisi = 100\% - RSD \quad (2.16)$$

Dimana  $x$  merupakan nilai pembacaan sensor dan  $\bar{x}$  adalah rerata nilai pembacaan sample dari  $i=1$  sampai  $i$  ke  $n$ . Presentase presisi didapatkan dari 100% dikurang nilai persentase RSD.

### 2.2.21 Packet Length

*Packet length* merupakan salah satu parameter QoS yang terkait dengan efisiensi dan performa pengiriman data melalui jaringan. *Packet length* mengacu pada ukuran total *byte* dari data yang dikirimkan dalam satu paket, termasuk *header*, *payload*, dan *footer* protokol. Semakin kecil *packet length*, maka transmisi data akan semakin efisien. Hal ini karena *overhead* protokol terbagi pada jumlah data yang lebih sedikit, sehingga dapat meningkatkan *throughput* dan mengurangi



*latency*. Pada protokol MQTT, kemampuan maksimum pengiriman data dalam satu paket adalah 268,435,456 *byte*. Semakin kecil nilai *packet length*, maka sistem tersebut akan semakin efisien dalam mengirimkan data. Optimalisasi *packet length* bertujuan untuk mencapai keseimbangan antara efisiensi dan reliabilitas transmisi. *Packet length* yang terlalu besar dapat mengurangi efisiensi, serta dapat meningkatkan kemungkinan *error*. Pemilihan *packet length* optimal mempertimbangkan berbagai faktor, seperti jenis protokol, karakteristik jaringan, dan kebutuhan [39].