

## BAB II DASAR TEORI

### 2.1 KAJIAN PUSTAKA

Pada penelitian [2] tentang sistem *monitoring* suhu dan kelembapan pada optimasi pembuatan tempe, penelitian tersebut membuat *prototype* dalam bentuk inkubator, menggunakan Esp8266 yang terhubung dengan sensor DHT22. Untuk kendali suhu dan kelembapan menggunakan 2 buah *fan* dan 1 buah penghangat ruangan [2].

Pada penelitian [3] tentang pengontrolan suhu dan kelembapan pada proses fermentasi tempe berbasis *internet of things*, membuat sebuah alat untuk *memonitoring* dan memproses fermentasi tempe menggunakan *internet*. Rentang suhu yang dibutuhkan antara 30°C –35°C, dengan kurun waktu 16 jam sampai tempe matang. Menggunakan sensor DHT22 dan *mikrokontroller NodeMcu*. Jika suhu lebih rendah dari nilai yang ditentukan maka lampu akan *on*, dan jika suhu lebih tinggi maka lampu akan *off*. Hasil pembacaan dari sistem ini dikirimkan ke aplikasi Blynk [3].

Pada penelitian [4] tentang sistem kendali dan *monitoring* dengan aplikasi Blynk dan *mikrokontroller* Wemos D1 R2 berbasis *IoT* pada pembuatan tempe, parameter yang diuji berupa stabilitas, respon sistem, keakuratan pengendalian, dan rerata waktu pengendalian. Komponen dari sistem berupa Wemos D1 R2, servo, kipas, lampu, dan *relay*. Ukuran inkubator 40 cm x 40 cm x 40 cm. Hasil pengujian stabilitas pada masing-masing alat menunjukkan alat tersebut dapat mempertahankan *setting point* yang telah ditentukan dan hasilnya stabil. Berdasarkan hasil uji respon sistem terhadap pengendalian aktuator diperoleh rata-rata 3,8 detik. Pengujian keakuratan pengendalian pada alat kendali dan *monitoring* menunjukkan nilai rata-rata akurasi 99,99%. Rerata waktu pengendalian diperoleh hasil yaitu 3 detik pada masing-masing alat. Pada proses kendali dan *monitoring* pada fermentasi tempe di ruang inkubator butuh waktu 33 jam untuk hasil yang sempurna, dibandingkan dengan fermentasi konvensional yang membutuhkan 45 jam untuk hasil yang sempurna [4].

Pada penelitian [5] tentang prototipe sistem *IoT* pada pengaturan suhu dan kelembapan pada fermentasi tempe. Penelitian ini menggunakan *mikrokontroller* WeMos D1 R2, sensor suhu DHT22 dan RTC DS3231. Sistem akan mengontrol kipas DC dan pemanas, lalu data dikirimkan ke aplikasi Blynk. Hasil dari penilitain menunjukkan suhu dan kelembapan stabil pada rentang 30°C -37°C dan 60-70 RH [5].

Pada penelitian [6] tentang sistem kendali suhu dan kelembapan pada proses fermentasi tempe dengan metode PID. Penelitian ini menggunakan *mikrokontroller* Atmega 328 dengan sensor DHT22. Suhu yang ditetapkan antara 25°C – 35°C. dan udara pada kelembapan 30%-80%. Fermentasi tempe akan optimal pada suhu stabil 28°C dan udara lempap 60% sehingga dibutuhkan waktu selama 20 jam hingga tempe matang [6].

Pada penelitian [7] tentang sistem kendali suhu dan kelembapan pada proses fermentasi tempe pada tahun 2020, penelitian ini berfokus untuk mempercepat proses fermentasi tempe, tempe mentah menjadi matang dalam rentang waktu 18 jam. Suhu normal yang diatur antara 30°C -37°C. kriteria keberhasilan dari penelitian ini diambil dari parameter warna, bau, dan rasa pada tempe. Dengan menggunakan *microcontroller* Arduino Mega2560, sensor suhu DHT11 berhasil menjaga suhu normal sehingga proses fermentasi tempe selesai dalam kurun waktu 16 jam [7].

Pada penelitian [8] tentang sistem *monitoring* suhu dan kelembapan pada *incubator* fermentasi tempe menggunakan *THINGER.IO*, penelitian ini berfokus pada optimasi dalam *incubator* fermentasi tempe sehingga proses fermentasi selesai dalam kurun waktu 22 jam. Suhu normal yang diatur dalam *incubator* antara 34°C-35°C dan kelembapan antara 65%-70%RH. Sistem ini menggunakan NodeMCU ESP8266 sebagai pengirim data ke *platform thinger.io*. sensor suhu yang digunakan adalah sensor DHT22, dan komponen pengendali berupa *heater*, dan *mist maker*. Dengan diperoleh persentase rata-rata *error* sebesar 1,2% untuk pembacaan suhu dan 3,2% untuk pembacaan kelembapan [8].

Pada penelitian [9] rancang bangun alat pengendali suhu pada fermentasi tempe berbasis *mikrokontroller*, berfokus pada optimasi suhu dalam proses fermentasi tempe dengan logika *fuzzy*. Suhu normal yang diatur supaya sistem

bekerja antara 30°C-37°C. dengan menggunakan sensor suhu DHT22, dan *mikrokontroller* Arduino UNO, proses otomatisasi sistem dibantu dengan komponenn lampu, sehingga apabila suhu lebih rendah dari *set point*, maka lampu akan menyala, dan apabila suhu lebih tinggi dari *set point*, maka lampu akan padam. Lama waktu dari proses fermentasi menggunakan alat otomatisasi membutuhkan 22 jam [9].

## 2.2 DASAR TEORI

Pada penelitian ini bertujuan untuk merancang sistem monitoring suhu dan kelembapan pada ruang fermentasi tempe dage. Terdapat beberapa teori yang digunakan sebagai acuan dalam perancangan sistem ini.

### 2.2.1 *Tempe Dage*

Tempe dage adalah produk olahan makanan yang populer di daerah Banyumas, Jawa Tengah. Olahan makanan ini merupakan hasil fermentasi dari ampas kelapa (bungkil) atau kacang menggunakan *Rhizopus sp* [10]. Jenis dari tempe dage ini tidak diproduksi di daerah lain di Indonesia, sehingga menjadi ciri khas tersendiri bagi Masyarakat Banyumas [11].



**Gambar 2.1 Olahan Makanan Tempe Dage [11]**

Gambar 2.1 menunjukkan tempe dage yang telah matang, proses pengolahan tempe dage dari bahan dasar bungkil memerlukan 2-3 hari hingga matang. Bungkil yang digunakan biasanya menggunakan dua jenis bungkil yang memiliki kualitas yang berbeda, salah satunya berkualitas baik dan lainnya kurang

baik. Ini dilakukan untuk menekan modal produksi dengan tetap menjamin tempe dage yang berkualitas baik,

Bungkil direndam dengan air selama satu malam penuh, kemudian pada pagi di esok hari air ditiriskan dan bungkil diperas untuk mengurangi kadar air, selanjutnya bungkil dimasak selama satu jam hingga tanak dan dituang ke tampah lebar untuk didinginkan. Proses ini kemudian diberi bakteri *Rhizpous Sp* yang biasa Masyarakat sebut dengan ragi atau laru untuk proses fermentasi. Proses selanjutnya bungkil diatur ketebalannya sedemikian rupa sehingga sesuai dengan kebutuhan pasar dan dibiarkan selama 2-3 hari sampai matang.

### **2.2.2 Fermentasi**

Fermentasi adalah proses yang namanya berasal dari kata Latin "*fervere*," yang berarti mendidih. Istilah ini digunakan karena cairan yang mengalami fermentasi tampak mendidih dan menghasilkan gelembung gas karbondioksida. Peneliti awal seperti Gay Lussac menemukan bahwa fermentasi melibatkan penguraian gula menjadi alkohol dan karbondioksida. Penemuan ini memberikan dasar bagi pemahaman kita tentang bagaimana bahan organik dapat diubah melalui proses fermentasi.

Penelitian lebih lanjut oleh Louis Pasteur mengungkapkan bahwa *mikroorganisme*, seperti ragi dan bakteri, memainkan peran kunci dalam fermentasi. Pasteur menunjukkan bahwa *mikroorganisme* ini menggunakan enzim untuk mengubah gula dan bahan organik lainnya menjadi produk yang berbeda, seperti alkohol, asam laktat, dan gas karbondioksida. Temuan ini penting karena menunjukkan bahwa fermentasi adalah proses biokimia yang dikendalikan oleh *mikroorganisme* dan enzim.

Dalam beberapa dekade terakhir, istilah fermentasi telah berkembang untuk mencakup berbagai jenis pemecahan bahan organik oleh *mikroorganisme*. Tidak hanya gula yang dapat dipecah, tetapi juga karbohidrat kompleks, protein, dan lemak. Proses ini digunakan dalam berbagai industri, termasuk pembuatan makanan dan minuman seperti roti, keju, yoghurt, bir, dan anggur. Selain itu, fermentasi juga digunakan dalam produksi bahan kimia dan obat-obatan, menunjukkan betapa pentingnya pemahaman kita tentang proses ini bagi banyak aspek kehidupan modern.

Fermentasi tidak hanya penting dalam konteks produksi makanan dan minuman, tetapi juga dalam bidang kesehatan dan lingkungan. Proses fermentasi dapat membantu dalam pembuatan probiotik, yang bermanfaat bagi kesehatan pencernaan manusia. Selain itu, fermentasi juga dapat digunakan dalam pengolahan limbah organik, mengubah bahan yang tidak berguna menjadi produk yang lebih bernilai dan ramah lingkungan.

Dengan demikian, fermentasi adalah proses kompleks yang melibatkan penguraian bahan organik oleh *mikroorganisme* melalui aktivitas enzimatik. Proses ini memiliki aplikasi luas dalam industri, kesehatan, dan lingkungan, membuatnya menjadi salah satu proses biokimia yang paling penting dan serbaguna [12].

### **2.2.3 Internet of Things (IoT)**

*Internet of Things*, atau *IoT*, adalah teknologi mutakhir yang dirancang untuk memperluas manfaat dari konektivitas internet yang selalu aktif. Tujuan utama *IoT* adalah menghubungkan berbagai objek di sekitar kita agar kegiatan sehari-hari menjadi lebih mudah dan efisien, sehingga sangat membantu dalam berbagai aspek pekerjaan manusia.

Pentingnya *IoT* semakin jelas dengan meningkatnya penerapan teknologi ini dalam berbagai aspek kehidupan modern. Contoh penerapannya dapat ditemukan di rumah pintar, kota pintar, hingga industri dan sektor kesehatan. Rumah pintar, misalnya, menggunakan perangkat *IoT* untuk mengontrol pencahayaan, keamanan, dan peralatan rumah tangga melalui smartphone atau perangkat lain yang terhubung ke internet. Di kota pintar, *IoT* membantu dalam manajemen lalu lintas, pengelolaan limbah, dan layanan publik lainnya, meningkatkan kualitas hidup warga.

Salah satu teknologi kunci dalam *IoT* adalah RFID (*Radio Frequency Identification*), yang digunakan untuk metode identifikasi dan komunikasi. RFID memungkinkan objek untuk dikenali dan dilacak menggunakan gelombang radio. Namun, *IoT* tidak terbatas pada RFID saja; teknologi ini juga mencakup sensor lain, teknologi nirkabel, dan kode *QR* (*Quick Response*). Sensor memungkinkan perangkat untuk mengumpulkan data lingkungan, seperti suhu, kelembapan, atau gerakan, dan mengirimkan informasi tersebut ke sistem pusat untuk analisis lebih lanjut. Teknologi nirkabel memastikan bahwa perangkat *IoT* dapat berkomunikasi

tanpa memerlukan kabel fisik, sementara kode QR memungkinkan akses cepat ke informasi melalui pemindaian.

Dengan kombinasi berbagai teknologi ini, *IoT* menciptakan jaringan perangkat yang saling terhubung, memungkinkan pengumpulan dan pertukaran data secara real-time. Hal ini memungkinkan otomatisasi dan pengelolaan yang lebih efisien, memberikan manfaat besar dalam hal produktivitas, kenyamanan, dan penghematan energi. *IoT* juga membuka peluang baru dalam berbagai sektor, termasuk transportasi, pertanian, dan perawatan kesehatan, di mana data yang akurat dan *real-time* dapat meningkatkan efektivitas dan efisiensi operasional.

Secara keseluruhan, *IoT* adalah teknologi yang semakin penting dalam dunia modern, memberikan banyak manfaat melalui peningkatan konektivitas dan otomatisasi, serta membuka jalan bagi inovasi dan peningkatan kualitas hidup di berbagai bidang.

Istilah “*Internet of Things*” terdiri dari dua bagian kata utama yaitu *Internet* yang menghubungkan dan mengatur sebuah konektivitas dan *Things* yang memiliki arti objek atau sebuah perangkat. Sederhananya, kamu memiliki “*Things*” yang dapat saling terhubung untuk mengumpulkan data dan mengirimkannya ke Internet. Data ini juga dapat diakses oleh “*Things*” lainnya juga. dimana sebuah “*Things*” tertentu memiliki kemampuan untuk mengirimkan data melalui jaringan dimanapun pengguna berada dan tanpa adanya interaksi dari manusia ke manusia ataupun dari manusia ke perangkat *computer* [13].

#### **2.2.4 Wemos D1 R1**

Wemos D1 R1 merupakan papan pengembangan yang menggunakan chip ESP2866, sebuah modul yang dirancang untuk memberikan kemampuan konektivitas WiFi pada perangkat elektronik. Dengan menggunakan ESP2866, mikrokontroler pada Wemos D1 R1 dapat dengan mudah terhubung ke *internet* melalui jaringan WiFi. Ini memungkinkan berbagai aplikasi *Internet of Things (IoT)* seperti pengendalian jarak jauh, pemantauan, dan pengumpulan data melalui koneksi internet tanpa memerlukan kabel fisik. Wemos D1 R1 mendukung berbagai protokol komunikasi dan memiliki antarmuka yang kompatibel dengan berbagai sensor dan perangkat eksternal, sehingga sangat fleksibel untuk berbagai proyek elektronik dan pengembangan perangkat pintar [14].



**Gambar 2.2 Wemos DIR1 [14]**

Gambar 2.2 menampilkan perangkat Wemos D1 R1, Wemos D1 R1 adalah papan yang menggunakan modul Wi-Fi ESP8266EX dan didesain mirip dengan Arduino Uno. Papan ini memiliki beberapa kelebihan, antara lain sifatnya yang *open source* dan kompatibilitasnya dengan Arduino. Wemos D1 R1 dapat diprogram menggunakan Arduino IDE, dan pinout-nya juga kompatibel dengan Arduino Uno. Selain itu, papan ini dapat berfungsi secara mandiri tanpa perlu mikrokontroler tambahan. Wemos D1 R1 dilengkapi dengan prosesor 32-bit yang memiliki kecepatan 80 MHz dan mendukung *High Level Language*, sehingga ideal untuk berbagai proyek pengembangan yang memerlukan konektivitas Wi-Fi dan kemampuan pemrosesan yang baik [15]. Adapun spesifikasi dari *board Wemos DIR1* terlihat pada Tabel 2.1

**Tabel 2.1 Wemos DIR1 Spesifikasi [15]**

<i>Microcontroller</i>	ESP8266 Tensilica 32-bit
<i>Serial to USB converter</i>	CH340G
<i>Operating voltage</i>	3,3V
<i>Input voltage</i>	7-12V
<i>Digital I/O pins</i>	11
<i>PWM I/O Pins (Shared with Digital I/O)</i>	10
<i>Analog Input Pins</i>	1 (10-bit)
<i>DC Current per I/O Pin</i>	12mA (Max)
<i>Hardware Serial Ports</i>	1

<i>Microcontroller</i>	ESP8266 Tensilica 32-bit
<i>Flash Memory</i>	4 Mbytes
<i>Instruction RAM</i>	64 Kbytes
Data RAM	96 Kbytes
<i>Clock speed</i>	80MHz
<i>USB conector style</i>	<i>Micro-B female</i>
<i>Board dimension</i>	69 x 53mm
<i>datasheet</i>	ESP8266EX

### 2.2.5 Antares

Antares merupakan platform *IoT* milik Telkom dengan berbagai fitur seperti *device management* hingga *data storage*. Antares mengunggulkan *zero infrastructure* management sehingga para pengembang aplikasi (*developer*) tidak perlu mengurus *server* karena sudah disediakan API dengan tingkat keamanan yang cukup baik karena sudah berstandar *global one M2M*. Penggunaan Antares bisa dilakukan dengan beberapa mikrokontroler seperti Arduino dan Raspberry Pi. Antares juga terhubung dengan *connectivity provider* yaitu GSM, LTE, LoRa, dan lain sebagainya. Tampilan logo pada antares seperti pada gambar 2.3 [16].

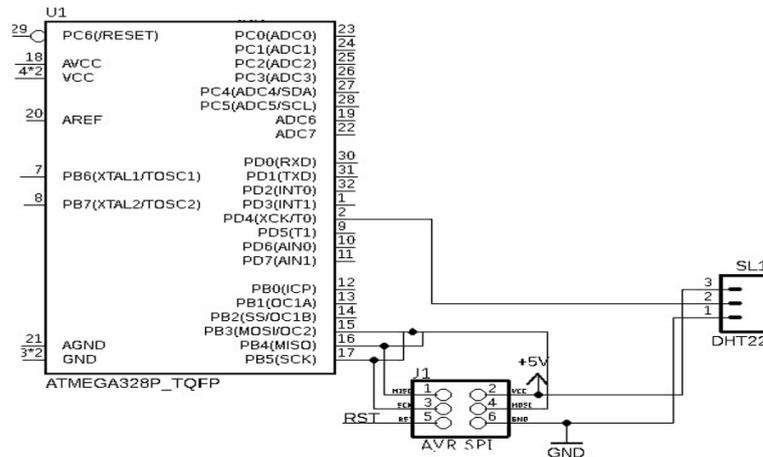


**Gambar 2.3 Logo Antares [17]**

### 2.2.6 DHT22

Sensor DHT22 adalah komponen perangkat untuk mengukur besaran suhu dan kelembapan. Sensor ini menghasilkan sinyal digital yang memerlukan kalibrasi agar berfungsi dengan baik. Dengan memanfaatkan teknik akuisisi sinyal digital yang khusus dan teknologi modern untuk pembacaan besaran nilai suhu serta kelembapan, sensor DHT22 memberikan keandalan dengan tinggi dan kestabilan

dengan jangka panjang. Setiap elemen atau *part* penunjang DHT22 dikalibrasi dengan ketat di laboratorium untuk menjamin akurasi dalam pengukurann suhu dan kelembapan [18].



**Gambar 2.4** Sirkuit DHT22 [19]

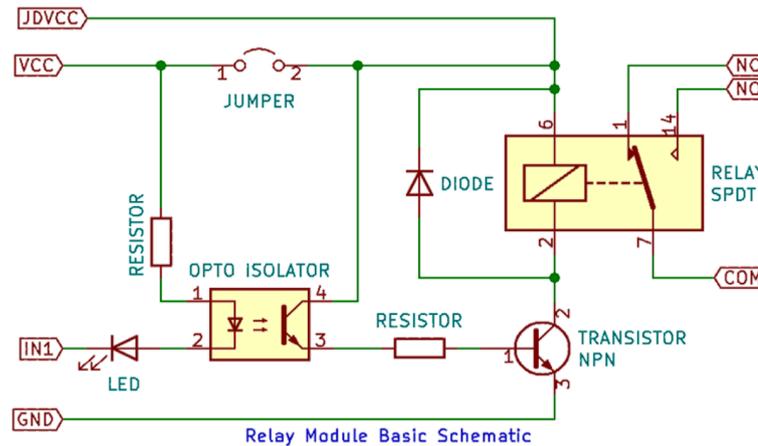
Gambar 2.4 menampilkan *circuit* dari sensor DHT22 sebagai sensor yang dapat membaca nilai suhu dan kelembapan. Spesifikasi dari sensor pembaca suhu dan kelembapan DHT22 ditampilkan pada Tabel 2.2

**Tabel 2.2** Spesifikasi Sensor DHT22 [19]

Model	DHT22
<i>Power Supply</i>	3,3-6V DC
<i>Ouput Signal</i>	<i>Digital signal via single-bus</i>
<i>Sensing Element</i>	<i>Polymer capacitor</i>
<i>Operating Range</i>	<i>Humidity 0-100% RH</i> <i>Temperature -40~80<sup>0</sup>C</i>
<i>Accuracy</i>	<i>Humidity +-2% RH(MAX+-5% RH)</i> <i>Temperature &lt;+-0,5<sup>0</sup>C</i>
<i>Sensitivity</i>	<i>Humidity 0,1% RH</i> <i>Temperature 0,1<sup>0</sup>C</i>
<i>Sensing Period</i>	<i>Average 2s</i>

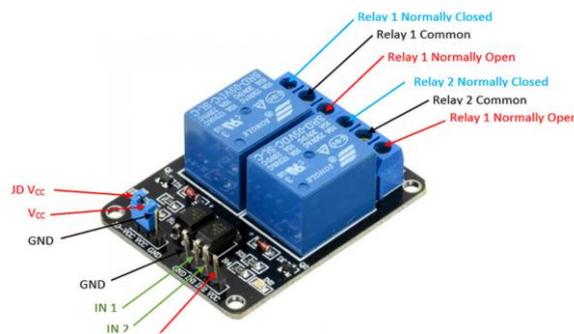
### 2.2.7 Relay 2 Channel

Secara prinsip kerjanya *relay* tergolong ke dalam jenis saklar (*switch*), di mana *relay* memiliki fungsi sebagai penghubung dan pemutus dari arus listrik yang kendalikan oleh sebuah pemacu atau sinyal picu dari perangkat elektronik lain yang terhubung dengannya. [20].



**Gambar 2.5 Diagram Sirkuit Relay 2 channel [20]**

Gambar 2.5 menampilkan *circuit* dari modul *relay 2 channel*. *Relay 2 channel* digunakan sebagai saklar otomatis pada sistem, sehingga dapat digunakan dalam otomatisasi dalam sistem yang akan dibuat khususnya *project IoT*.



**Gambar 2.6 Relay 2 channel [20]**

Gambar 2.6 menampilkan bentuk perangkat dari komponen *relay 2 channel*. Terdapat pin input data 1 dan 2, daya, dan *ground*. *Relay 2 channel* dapat diatur kondisi awalnya dengan logika 0 dan 1. Pada Tabel 2.3 menjelaskan tentang *pinout relay 2 channel*.

**Tabel 2.3 Pinout Relay 2 Channel [20]**

No	Nama Pin	Deskripsi
1	Vcc	Input sumber daya

No	Nama Pin	Deskripsi
2	GND	<i>Ground</i>
3	IN1	<i>Input aktivasi relay pertama</i>
4	IN2	<i>Input aktivasi relay kedua</i>

Tabel 2.4 menjelaskan tentang spesifikasi *relay 2 channel*. *Relay 2 channel* menggunakan tegangan operasi sebesar 5V, sehingga dapat langsung dihubungkan ke sistem mikrokontroler. Tipe relay ini adalah SPDT (Single Pole Double Throw), yang memiliki 1 COMMON, 1 NC (Normally Close), dan 1 NO (Normally Open). Relay ini dapat menangani beban hingga 10A.

**Tabel 2.4 Spesifikasi Relay 2 Channel [20]**

Model	<i>Relay 2 Channel</i>
Sumber tegangan	3,75-6V
Arus	5mA
Arus saat aktif	~70mA(tunggal),~140mA(keduanya)
Tegangan maksimum	250VAC, 30VDC
<i>Relay arus maksimum</i>	10A

### 2.2.8 Lampu Bohlam

Lampu bohlam atau lampu pijar merupakan sumber cahaya sekaligus sumber panas. Lampu bohlam bekerja dengan cara mengalirkan arus Listrik melalui filamen, sehingga filamen akan memanas dan Menghasilkan Cahaya. Pada Gambar 2.7 merupakan tampilan dari lampu yang menampilkan Lampu bohlam, memiliki kaca yang menyelubungi filamen supaya terhindar dari kontak langsung dengan udara yang dapat memicu oksidasi [21].



**Gambar 2.7 Lampu Bohlam [21]**

### 2.2.9 *Blower*

*Blower* atau kipas angin disini merupakan komponen untuk pengendali suhu ruangan pada sistem, komponen ini banyak diaplikasikan pada *internet of things*, terutama dalam sistem untuk pengendali suhu pada suatu ruangan. Gambar 2.8 menampilkan komponen kipas angin, kipas angin akan menerima informasi dari sensor suhu, jika suhu dalam ruangan yang terbaca tergolong panas, maka kipas angin akan menyala untuk mengalirkan udara panas supaya suhu dalam ruangan menjadi normal seperti yang diinginkan [22].



**Gambar 2.8 Kipas Angin/Blower [22]**

### 2.2.10 *Akurasi Sensor*

Akurasi adalah konsep penting dalam berbagai bidang, termasuk teknologi dan ilmu pengetahuan. Secara umum, akurasi merujuk pada tingkat kebenaran atau ketepatan suatu informasi atau data. Artikel ini akan membahas secara rinci pengertian akurasi, pentingnya akurasi, serta cara-cara untuk mengukur dan meningkatkan akurasi dalam berbagai bidang.

Pengukuran akurasi dapat dilakukan dengan berbagai metode, tergantung pada bidang atau konteksnya. Di dunia teknologi, akurasi sering kali dinilai melalui metode statistik, seperti tingkat kebenaran atau tingkat kesalahan. Sebagai contoh, dalam pengenalan wajah, akurasi dapat dihitung berdasarkan persentase wajah yang dikenali dengan tepat. Sementara itu, dalam penelitian ilmiah, akurasi diukur

menggunakan metode validasi seperti uji coba ulang atau perbandingan dengan data yang telah ada. Perhitungan akurasi menggunakan parameter *absolut error* pada Persamaan(2.1), sehingga didapat persentase *error* dari Persamaan (2.2) dan persentase akurasi dari Persamaan (2.3) [23].

$$\mathbf{Absolut\ error} = \left| \frac{\mathbf{sensor-alat\ pembanding}}{\mathbf{alat\ pembanding}} \right| \quad (2.1)$$

$$\%error = \left| \frac{\mathbf{sensor-alat\ pembanding}}{\mathbf{alat\ pembanding}} \right| \times 100 \quad (2.2)$$

$$\%akurasi = \left| 1 - \frac{\mathbf{sensor-alat\ pembanding}}{\mathbf{alat\ pembanding}} \right| \times 10 \quad (2.3)$$

### 2.2.11 Quality of Service (QoS)

Perhitungan *Quality of Service* digunakan untuk mengukur perhitungan pada suatu kelompok data yang digunakan. *QoS* digunakan untuk menentukan kualitas jaringan dalam pengiriman data sistem. Parameter yang akan diuji dari *QoS* berupa *Throughput*, *Packet loss*, *Delay*, *Jitter*, kemudian dari hasil pengukuran ini akan di standarisasikan dengan standar ITU [24].

#### 1. *Throughput*

*Throughput* adalah ukuran kecepatan transfer data yang sebenarnya, dinyatakan dalam bit per detik (bps). Ini menunjukkan jumlah total paket yang berhasil diterima di tujuan dalam jangka waktu tertentu, dibagi dengan lamanya jangka waktu tersebut [25]. Untuk mengukur nilai *throughput* dapat memasukkan rumus Persamaan (2.4) [26].

$$\mathbf{Throughput} = \frac{\mathbf{Jumlah\ data\ diterima}}{\mathbf{Lama\ pengamatan}} \quad (2.4)$$

#### 2. *Packet Loss*

*Packet Loss* adalah parameter yang menggambarkan jumlah total paket yang hilang dalam jaringan. Hal ini bisa terjadi karena adanya tabrakan (*collision*) atau kemacetan (*congestion*) pada jaringan, dan berdampak pada semua aplikasi karena pengulangan transmisi akan berdampak dalam pengaruh efisiensi jaringan secara keseluruhan, meskipun ada cukup *bandwidth* yang tersedia untuk aplikasi-aplikasi tersebut. Jika kemacetan berlangsung lama, *buffer* akan penuh dan data baru tidak akan diterima [25]. Kategori dalam pengukuran *Packet loss* menggunakan standarisasi ITU pada Tabel 2.5. Untuk mengukur nilai *packet loss* dapat menggunakan rumus Persamaan (2.5) [26].

**Tabel 2.5 Kategori Packet Loss Standar ITU [26]**

Kategori	Packet Loss (%)	Indeks
Sangat Bagus	0%	4
Bagus	3%	3
Sedang	15%	2
Jelek	25%	1

Kategori dalam pengukuran *Packet loss* menggunakan standarisasi ITU pada Tabel 2.5. Dalam mengukur nilai *packet loss* dapat menggunakan rumus persamaan *packet loss* pada Persamaan (2.5)

$$Packet\ loss = \frac{paket\ data\ dikirim - paket\ data\ diterima}{paket\ data\ dikirim} \times 100\% \quad (2.5)$$

### 3. Delay

*Delay* merupakan lama waktu yang dibutuhkan dari pengiriman data untuk menempuh jarak dari pengirim ke penerima. *Delay* dapat terpengaruh oleh berbagai kondisi, seperti jauhnya jarak pengiriman data, media transmisi yang digunakan, media pengirim yang digunakan, dan jenis media penerima yang digunakan [26]. Pengukuran *delay* dalam standarisasi ITU Menghasilkan beberapa kategori pada Tabel 2.6. [26].

**Tabel 2.6 Kategori Delay Standar ITU [26]**

Kategori	Besar delay (m/s)	Indeks
Sangat Bagus	<150m/s	4
Bagus	150m/s-300m/s	3
Sedang	300m/s-450m/s	2
Jelek	>450m/s	1

Untuk mengukur nilai dan kualitas dari parameter *QoS* berupa *delay* pada pengiriman paket data dalam sistem monitoring yang dibuat menggunakan Persamaan (2.6).

$$Rata - rata\ delay = \frac{Total\ delay}{Total\ paket\ diterima} \quad (2.6)$$

### 4. Jitter

*Jitter* adalah perubahan atau variasi dalam waktu kedatangan paket. *Jitter* juga dapat diartikan sebagai gangguan pada komunikasi digital atau analog akibat fluktuasi sinyal. Gangguan ini bisa menurunkan nilai dari kualitas komunikasi yang

terjadi antara pengirim dan penerima, gangguan pada pengiriman data dapat terjadi dalam berbagai keadaan, dari banyak kemungkinan terjadinya gangguan terjadi seperti suara yang terputus-putus dalam panggilan *VoIP* atau penurunan kualitas video saat streaming. Penyebab *jitter* sering kali adalah kemacetan jaringan, manajemen lalu lintas yang kurang efektif, atau masalah pada perangkat keras jaringan. Untuk mengurangi *jitter*, dapat digunakan teknik seperti buffering dan pengaturan prioritas *QoS (Quality of Service)* [26]. Standarisasi menurut ITU terbagi menjadi beberapa kategori pada Tabel 2.7. Untuk mengukur nilai *jitter* dapat menggunakan rumus (2.7) [26].

**Tabel 2.7 Kategori *Jitter* Standar ITU [26]**

Kategori	Besar <i>jitter</i> (m/s)	Indeks
Sangat Bagus	0m/s	4
Bagus	0m/s-75m/s	3
Sedang	75m/s-125m/s	2
Jelek	125m/s-225m/s	1

Total variasi delay diperoleh dari penjumlahan selisih delay antara paket satu dengan paket lainnya. Sehingga dapat diperoleh nilai *jitter* dan total variasi *delay* dengan Persamaan (2.7) untuk pengukuran *jitter* dan Persamaan (2.8) untuk total variasi *delay*.

$$\mathbf{Jitter} = \frac{\mathbf{Total\ variasi\ delay}}{\mathbf{Total\ paket\ diterima}} \quad (2.7)$$

$$\mathbf{Total\ variasi\ delay} = \mathbf{delay} - (\mathbf{rata} - \mathbf{rata\ delay}) \quad (2.8)$$

### 2.2.12 *Wireshark*

*Wireshark* adalah alat analisis paket yang tersedia secara gratis dan bersifat open-source. Fungsinya mencakup pemecahan masalah jaringan, analisis, pengembangan perangkat lunak, serta pengembangan protokol komunikasi. Selain itu, *Wireshark* juga digunakan dalam konteks pendidikan. Sebagai salah satu dari banyak aplikasi *Network Analyzer* yang populer, *Wireshark* sangat berguna bagi administrator jaringan untuk memantau kinerja jaringan dan mengatur arus data. Dengan kemampuannya untuk menangkap berbagai jenis paket data di jaringan, *Wireshark* memfasilitasi analisis yang mendalam terhadap informasi yang dikirimkan melalui berbagai format protokol [27].