

BAB 2 DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian [5] membahas tentang penanganan kualitas air pada kolam ikan koi berbasis IoT menggunakan informasi berupa nilai suhu, pH dan salinitas. Terdapat sistem kontrol secara otomatis untuk nilai pH dan nilai suhu. Data pembacaan sensor dari setiap sensor diakses melalui telegram. Penelitian ini belum menggunakan pengukuran kualitas layanan internet untuk mengetahui apakah kualitas layanan internet dari *hardware* menuju telegram sudah baik atau belum.

Penelitian [6] mengenai sistem kontrol dan *monitoring* kualitas air pada kolam ikan koi berbasis IoT menggunakan informasi berupa nilai suhu dan pH. Penelitian ini hanya difokuskan pada sistem kontrol suhu pada kolam ikan koi untuk nilai dari pH air belum terdapat kontrol melainkan hanya *monitoring*. Dalam penelitian ini mengukur kualitas layanan internet dengan menggunakan analisis jarak untuk mengetahui perbedaan dari pengukuran setiap jarak, parameter yang di ukur yaitu *delay* dan *throughput* namun belum menggunakan protokol.

Penelitian selanjutnya [7] penelitian ini memiliki tujuan untuk mengetahui kualitas air tambak ikan koi dengan parameter suhu dan pH. Menggunakan mikrokontroler ESP32 untuk tempat pengirim dan penerima data dari sensor pH dan suhu air. Menggunakan sistem kontrol untuk menguras dan mengisi air di tambak. Tidak terdapat sistem kontrol untuk nilai suhu dan pH hanya *monitoring* yang dilakukan di thingspeak serta tidak melakukan pengukuran kualitas layanan internet untuk pengiriman data.

Penelitian selanjutnya [2] melakukan sistem kontrol dan *monitoring* untuk nilai suhu, pH dan kadar *ammonia* dengan menggunakan protokol MQTT untuk pengukuran kualitas layanan internet. Dalam penelitian ini mengukur kualitas layanan internet dengan parameter *delay* dan *throughput*. Penelitian menggunakan sistem pendingin air (*cooler*) namun hasil yang didapatkan belum optimal.

Penelitian selanjutnya [8] membahas mengenai pemantauan suhu dan ketinggian air pada akuarium ikan hias berbasis IoT. Terdapat sistem kontrol dan

monitoring untuk sensor suhu dan ultrasonik, pemantauan data yang didapatkan terdapat pada bot telegram. Menggunakan ESP8266 untuk mikrokontroler, WiFi yang digunakan untuk *network* pengiriman data. Namun dalam penelitian ini belum mengukur kualitas layanan internet untuk pengiriman data.

Sedangkan penelitian [4] berbeda dengan penelitian sebelumnya, di mana jurnal ini memberikan informasi tentang pengembangan budi daya ikan air tawar. Pengembangan budi daya ikan air tawar ini menggunakan garam ikan sebagai inovasi baru dalam budi daya ikan air tawar. Menggunakan metode SWOT untuk mendapatkan strategi yang tepat sasaran. Namun dalam penelitian ini belum terdapat sistem kontrol dan *monitoring* untuk nilai kuliatas air serta belum terdapat pengukuran kualitas layanan internet.

Tabel 2.1 merupakan tabel penjelasan tentang kajian pustaka sebelumnya. Berisikan tentang tahun penelitian, nama pengarang, judul penelitian dan sensor digunakan.

Tabel 2. 1 Kajian Penelitian Sebelumnya

No	Year	Author	Title	Sensor Used
1.	2020	Riyan Kharisma dan Suryadhi Thaha	Rancang Bangun Alat <i>Monitoring</i> dan Penanganan Kualitas Air Pada Akuarium Ikan Hias Berdasarkan <i>Internet of Things (IoT)</i>	Sensor pH, sensor suhu, dan sensor TDS
2.	2020	Ahmad Hafizh Giffari Akbar, Ahmad Tri Hanuranto, dan UkeUsman Kurniawan	Sistem Pintar Berbasis <i>Internet of Things (IoT)</i> Untuk Kolam Ikan Koi <i>Internet of Things (IoT)</i> <i>Based Smart System For Koi Fish Pond</i>	Sensor suhu dan sensor pH
3.	2021	Sofiana Yuli Damayanti, Teguh	Sistem <i>Monitoring</i> Kualitas Air Tambak Ikan Koi (<i>Cyprinus</i>	Sensor suhu dan sensor pH

No	Year	Author	Title	Sensor Used
		Andriyanto, dan Aidina Ristywan	<i>Carpio</i>) Berbasis Teknologi <i>Internet of Things</i> (IOT)	
4.	2021	Riezky Fakhriza, Basuski Rahmat, dan Sri Astuti	Perancangan dan Implementasi Alat <i>Monitoring</i> dan <i>Controlling</i> Kualitas Air Pada Kolam Ikan Koi	Sensor suhu, sensor <i>ammonia</i> , dan sensor pH
5.	2022	Piter Wijaya, dan Theophilus Wellem	Perancangan dan Implementasi Sistem Pemantauan Suhu dan Ketinggian Air pada Akuarium Ikan Hias berbasis IoT	Sensor DS18B20 dan sensor ultrasonik
6.	2023	Heru Winarno, Reza Wiliyanto, Vini Nurhayuni, Efa Isnawati	Pemberdayaan Masyarakat Melalui Pengembangan Budidaya Ikan Air Tawar di Sekitar Aliran Sungai Yang Rentan Terjadi Banjir	-

2.2 DASAR TEORI

2.2.1 Ikan Koi (*Cyprinus Carpio*)

Koi merupakan ikan hias asal Jepang, kini telah menjadi salah satu ikan yang banyak dibudidayakan di Indonesia. Ikan ini sangat diminati karena keindahan dan ragam warnanya yang menarik. Selain itu, dalam sebagian masyarakat Indonesia, ikan koi juga dipercaya membawa keberuntungan [9]. Suhu yang ideal untuk pertumbuhan ikan koi adalah antara 25 hingga 30°C. Jika suhu melebihi batas tersebut, dapat melemahkan daya tahan ikan. Selain suhu, tingkat pH di kolam juga

berpengaruh pada pertumbuhan ikan koi, dan pH yang ideal adalah antara 7 hingga 8 [3].

Ciri-ciri ikan koi yang berada dalam kondisi sehat dapat dilihat dari gerakan mereka yang lincah dan aktif saat merespons makanan. Di sisi lain, ikan koi yang tidak dalam keadaan sehat akan menunjukkan gerakan yang lambat dan kurang responsif terhadap pakan [10].

Manfaat pemberian garam ikan pada kolam ikan koi adalah mencegah adanya pertumbuhan bakteri pada kolam ikan koi yang bisa datang kapan saja. Pemberian garam berpengaruh pada ikan koi yang terserang bakteri, semakin tinggi kadar garam dalam air semakin sedikit parasit yang menyerang ikan [11].

Pemberian kadar garam pada kolam ikan menyesuaikan dengan berapa liter air yang terdapat pada kolam. Kolam dengan ukuran 200 x 50 x 100 cm membutuhkan kadar garam 1 hingga 2 ppm, jika nilai kadar garam kurang dari batas tersebut maka akan berdampak pada pertumbuhan bakteri yang terdapat pada kolam ikan koi.

2.2.2 *Internet of Things*

IoT adalah inovasi teknologi terkini yang memungkinkan akses internet untuk mengenali objek dengan cerdas terkait perilakunya. Secara umum, IoT mencerminkan konsep dari perangkat jaringan yang dapat mengumpulkan data dari seluruh dunia, dan kemudian membagikan data tersebut melalui internet untuk pengolahan dan penggunaan berbagai tujuan. Dengan IoT, objek dapat terhubung melalui koneksi internet, memungkinkan mereka berbagi informasi dan menjalankan proses otomatis [12].

IoT memiliki kemampuan untuk mengumpulkan data dari seluruh dunia dan menyebarkannya melalui internet untuk diolah dan dimanfaatkan dalam berbagai keperluan. Konsep IoT membuka peluang bagi banyak objek untuk terhubung dalam jaringan internet secara luas dan ekonomis, termasuk dalam berbagai bentuk seperti objek fisik dan sistem kontrol. Salah satu keunggulan IoT adalah kemampuannya untuk mengumpulkan dan mengirim data melalui jaringan tanpa perlu campur tangan manusia [12].

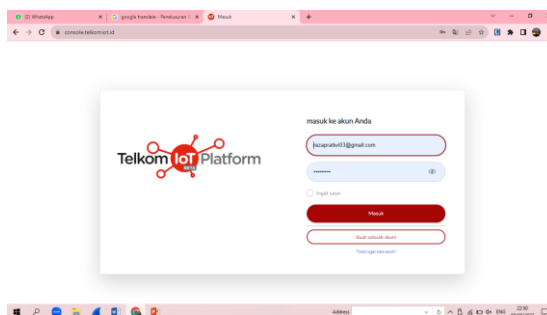


Gambar 2. 1 Arsitektur *Internet of Things* [12]

Gambar 2.1 merupakan arsitektur IoT yang terdiri dari 4 aspek, aspek yang pertama yaitu *device* terdiri perangkat IoT, seperti sensor, perangkat pintar, dan perangkat yang dapat terkoneksi dengan internet. Perangkat ini bisa berkomunikasi dengan *network* dan layanan protokol komunikasi. Kemudian aspek kedua yaitu *network* terdiri jaringan perangkat dan teknologi yang memungkinkan alat untuk berkomunikasi dengan internet. Beberapa protokol jaringan yang digunakan dalam IoT termasuk *WiFi*, *Bluetooth*, *ZigBee*, dan *LoRaWAN*. Kemudian aspek ketiga yaitu platform terdiri dari perangkat lunak dan layanan yang berfungsi mengelola data yang telah dikumpulkan oleh perangkat IoT dan menyediakan layanan. Kemudian aspek ke empat yaitu *application* terdiri dari layanan yang difungsikan untuk memproses serta menganalisis data pada perangkat IoT sehingga memungkinkan pengguna untuk memantau, mengontrol, dan mengotomatisasi perangkat IoT.

2.2.3 Telkom IoT

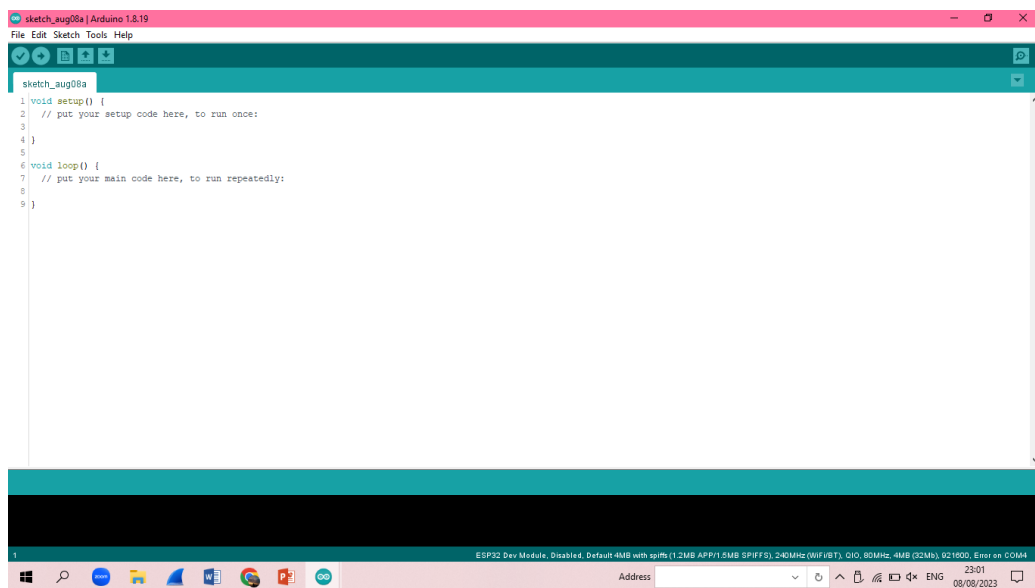
Gambar 2.2 merupakan salah satu platform IoT yang dinaungi oleh PT Telkom Indonesia. Fungsi dari platform ini untuk lokasi dokumentasi data.



Gambar 2. 2 Platform Telkom IoT

2.2.4 Arduino IDE

Gambar 2.3 merupakan tampilan awal pada *software* arduino. Dalam penelitian ini, digunakan perangkat lunak bernama *Arduino integrated development environment (IDE)*. Suatu pemrograman yang digunakan untuk menulis perintah atau sumber kode, melakukan pemeriksaan kesalahan dalam skrip kode, dan menyimpan program yang dibuat dalam satu aplikasi. Bahasa pemrograman yang digunakan untuk membuat program Arduino Uno adalah turunan dari bahasa C/C++ [13].

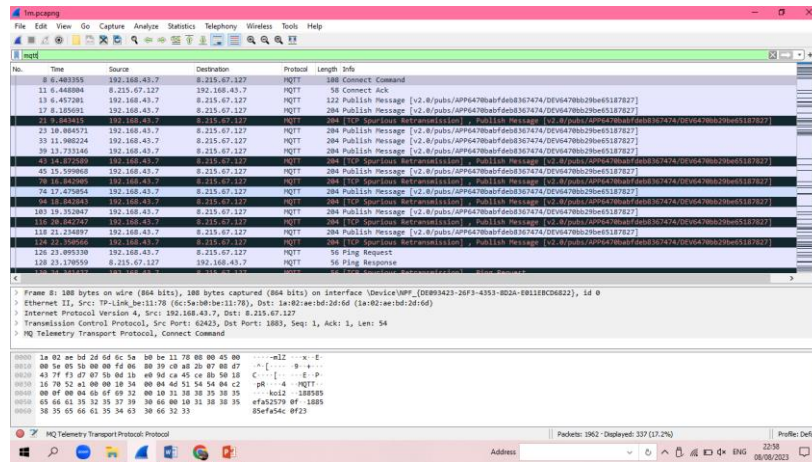


Gambar 2. 3 Tampilan Awal Arduino

2.2.5 Software Wireshark

Gambar 2.4 merupakan tampilan di *software wireshark*. Merupakan sebuah alat yang digunakan untuk melakukan analisis lebih lanjut terhadap paket data dalam jaringan, yang juga disebut sebagai jaringan penganalisis paket. Fungsinya adalah menangkap setiap paket data yang melewati jaringan dan menampilkan detail informasi dari setiap paket tersebut. Dengan *Wireshark*, berbagai macam paket informasi dapat direkam dan dianalisis dengan lebih mudah. Alat ini berperan sebagai *network packet analyzer* yang berfungsi untuk *memonitor* dan menampilkan seluruh informasi yang lengkap dari paket-paket dalam jaringan. Sebagai alat untuk memeriksa, memantau, dan menganalisis paket-paket yang

berada dalam jaringan, *Wireshark* memudahkan proses analisis jaringan dengan efektif [14].



Gambar 2. 4 Software Wireshark

2.2.6 Wireless Local Area Network

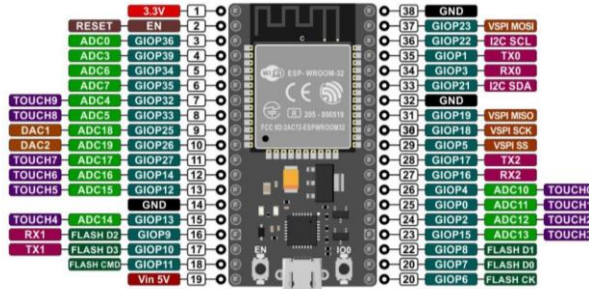
WLAN adalah jaringan lokal nirkabel. WLAN menggunakan standar 802.11 yang disediakan oleh IEEE. Teknologi jaringan WLAN telah mengalami perkembangan hingga lima generasi. Berikut adalah daftar kecepatan data untuk setiap kode IEEE 802.11:

- a. IEEE 802.11b memiliki kecepatan 11 Mbps
- b. IEEE 802.11g memiliki kecepatan 54 Mbps
- c. IEEE 802.11a memiliki kecepatan 54 Mbps
- d. IEEE 802.11n kecepataannya lebih dari 100 Mbps hingga 500 Mbps
- e. IEEE 802.11b memiliki kecepatan 1300 Mbps
- f. IEEE 802.11b / g / a / n / ac menunjukkan frekuensi
- g. IEEE 802.11ax memiliki kecepatan data 1,4 Gbps [15]

2.2.7 NodeMCU ESP32

Gambar 2.5 merupakan pin dari mikrokontroler NodeMcu ESP32. Mikrokontroler ESP32 merupakan pengembangan dari mikrokontroler ESP8266. Secara spesifikasi ESP32 sangat lengkap terdapat inti CPU serta *WiFi* yang lebih cepat, GPIO yang lebih banyak dan kemampuan *bluetooth low energy*. ESP32 menggunakan mikroprosesor *tensilica xtensa LX6 dual-core* atau *single-core*

dengan *clock rate* hingga 240 MHz [16]. Tabel 2.2 merupakan spesifikasi dari ESP32 dengan spesifikasi daya, CPU, arsitektur, *flash memory*, SRAM, *Bluetooth*, *WiFi*, dan UART



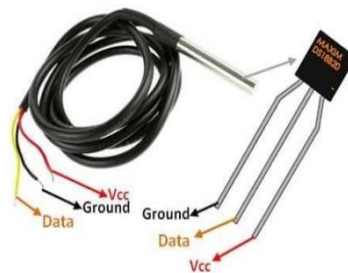
Gambar 2. 5 Arsitektur ESP32 [17]

Tabel 2. 2 Spesifikasi ESP32

Spesifikasi	ESP32
Daya	3.3V
CPU	<i>Xtensa dual core</i> LX6 – 160 MHz
Arsitektur	32 bit
<i>Flash Memory</i>	16 MB
SRAM	512 kB
GPIO PIN (DAC/ADC)	36 (18/2)
<i>Bluetooth</i>	Ada
<i>WiFi</i>	Ada
SPI/I2C/UART	4/2/2

2.2.8 Sensor suhu

Sensor suhu DS18B20 adalah sebuah sensor suhu dengan keluaran digital. Rentang pengukurannya mulai dari -55°C hingga +125°C. Sensor suhu DS18B20 ini dilengkapi dengan pelindung air (*waterproof*), sehingga cocok digunakan dalam pengukuran suhu di dalam air [18].



Gambar 2. 6 Sensor DS18B20

Gambar 2.6 terlihat bahwa sensor DS18B20 memiliki 3 pin yang berbeda, yaitu *ground* (GND) (berwarna hitam), Data (berwarna kuning), dan VCC (berwarna merah). Nama pin menunjukkan koneksi yang tepat, di mana VCC terhubung ke sumber tegangan (+5V), *ground* terhubung ke *ground*, dan pin data terhubung ke pin digital pada ESP32 [18].

Spesifikasi dari sensor suhu antara lain adalah:

- a. Bekerja pada tegangan 3V hingga 5V
- b. Range *temperature* berkisar -55°C to $+125^{\circ}\text{C}$
- c. *Waterproof* atau tahan air
- d. *Output* digital

2.2.9 Relay

Relay merupakan salah satu komponen yang umum digunakan dalam proyek-proyek elektronika. Fungsinya adalah untuk mengontrol pembukaan dan penutupan aliran listrik dalam rangkaian [19]. Menggunakan *type relay* SRD-5V.



Gambar 2. 7 Relay [19]

Gambar 2.7 merupakan relay dengan *normally close* (NC) merupakan pin yang menghubungkan salah satu kabel bila menginginkan kondisi arus listrik tersambung. *Common* (CO) merupakan pin yang dihubungkan dengan salah satu dari dua kabel yang terpakai. *Normally open* (NO) merupakan pin yang menghubungkan salah satu kabel bila menginginkan kondisi arus listrik terputus.

2.2.10 Pemanas Air

Gambar 2.8 merupakan pemanas air yaitu perangkat yang berperan dalam memanaskan air dengan menggunakan energi sebagai sumber panas. Fungsinya adalah untuk menjaga suhu air agar tetap stabil, terutama dalam konteks perawatan ikan. Perubahan suhu yang signifikan dan suhu yang ekstrim dapat berdampak besar terhadap kesehatan ikan koi [20].



Gambar 2. 8 Heater [20]

2.2.11 Motor DC

Motor DC yang digunakan pada penelitian ini yaitu kipas DC dan pompa air berikut penjelasannya:

2.2.11.1 Kipas DC

Gambar 2.9 merupakan kipas DC, perangkat yang berfungsi untuk mengubah udara menjadi angin dan digunakan sebagai pendingin. Dalam penelitian ini, kipas digunakan untuk mengalirkan udara dari luar kolam ke dalam kolam, sehingga aliran udara dan sirkulasi dapat menjaga suhu di dalam kolam tetap stabil. Selain itu, kipas juga berperan sebagai pendingin untuk menjaga suhu air kolam. Suhu yang tidak normal akibat kurangnya sirkulasi udara di dalam kolam dapat menyebabkan stres pada ikan dan bahkan menyebabkan kematian.



Gambar 2. 9 Kipas DC

2.2.11.2 Pompa Air

Gambar 2.10 merupakan pompa air adalah jenis pompa yang menggunakan motor DC dan tegangan DC sebagai sumber tenaganya. Pompa air ini menggunakan tegangan 12V dengan kapasitas 3.5L per menit.



Gambar 2. 10 Pompa Air

2.2.12 Sensor Ph

Gambar 2.11 merupakan sensor pH dengan modul pH *diy more* pH-502C merupakan salah satu jenis modul pH yang menghasilkan analog. Menurut spesifikasinya, modul ini beroperasi pada tegangan $5\pm 0,2V$ (AC DC) dan mampu mendeteksi rentang pH dari 0 hingga 14, serta rentang suhu dari 0 hingga $80^{\circ}C$. Modul ini memiliki waktu respon sekitar 5 detik dan waktu penyelesaian sekitar 60 detik dengan daya sebesar 0,5W [18].



Gambar 2. 11 Sensor pH [18]

Spesifikasi dari sensor pH antara lain adalah:

- Bekerja pada tegangan 3 volt
- Menggunakan modul pH *DIY More* pH-502C
- Waterproof* atau tahan air
- Output* analog

2.2.13 Sensor Salinitas

Gambar 2.12 merupakan sensor salinitas yaitu perangkat yang digunakan untuk mengukur tingkat salinitas atau kandungan zat terlarut dalam air, khususnya kadar garam. Sensor ini terdiri dari dua elektroda utama yang dapat membaca nilai salinitas dan menghasilkan data karakterisasi terkait kandungan garam dalam air [21].



Gambar 2. 12 Sensor Salinitas

Spesifikasi dari sensor salinitas antara lain adalah:

- a. Bekerja pada tegangan 5 *volt*
- b. *Waterproof* atau tahan air
- c. *Output* analog
- d. Sensor berada di dalam pipa PVC 1½ inch
- e. Dimensi: panjang pipa 18cm x ½ inch

2.2.14 *Message Queuing Telemetry Transport Protocol*

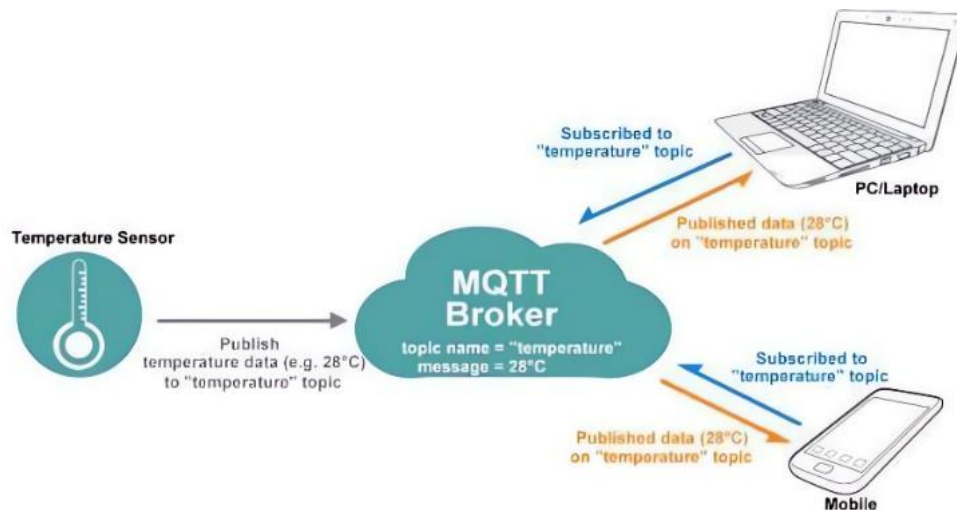
Message Queuing Telemetry Transport (MQTT) adalah sebuah protokol publikasi langganan yang ringan yang dirancang khusus untuk komunikasi dengan perangkat berdaya rendah atau pesan sederhana. MQTT merupakan salah satu protokol yang dioptimalkan untuk digunakan dalam jaringan dengan *bandwidth* rendah dan konsumsi energi yang efisien. Protokol ini awalnya dikembangkan oleh IBM dan telah digunakan dalam berbagai aplikasi IoT.

Gambar 2.13 sistem komunikasi terdiri dari tiga komponen utama, yaitu *publisher*, *broker*, dan *subscriber*. *Publisher*, atau disebut juga klien MQTT,

merupakan perangkat yang terhubung dengan sensor dan mikrokontroler, bertugas mengirimkan informasi dengan subjek atau kelompok data yang berbeda. Sementara itu, *subscriber*, juga merupakan klien MQTT, berperan sebagai penerima informasi yang dipilih sesuai dengan topik yang diminati.

Peran penting lainnya adalah *broker*, yang bertindak sebagai *server* MQTT, bertugas menghubungkan proses transfer data sebagai perantara. *Broker* mengelola dan menyusun semua informasi yang dikirim oleh penerbit dan meneruskannya ke pelanggan yang sesuai dengan topik pesan yang dipublikasikan.

Dalam lingkungan IoT, data dari sensor dan aktuator berkomunikasi melalui mediator. Protokol *publish-subscribe* berbasis pesan yang ringan ini memerlukan *broker* pesan, di mana *broker* bertanggung jawab untuk mendistribusikan pesan ke klien yang tertarik berdasarkan topik pesan yang telah dipublikasikan [2].



Gambar 2. 13 Arsitektur Protokol *MQTT* [2]

2.2.15 Regresi Linear

Persamaan regresi linier sederhana adalah sebuah model matematika yang menggambarkan keterkaitan antara variabel *independen* (X) dan variabel *dependen* atau *respons* (Y). Persamaan regresi linier sederhana direpresentasikan secara matematis sebagai berikut:

$$Y=a+bX$$

2.1

Di mana Y merupakan garis regresi atau variable *respons*, a merupakan konstanta perpotongan dengan sumbu *vertical*, b merupakan konstanta regresi (*slope*) dan X merupakan variabel bebas [22].

Untuk mencari nilai a dan b pada persamaan maka diperlukan persamaan berikut:

$$a = \frac{\Sigma Y \Sigma X^2 - \Sigma X \Sigma XY}{n (\Sigma X^2) - (\Sigma X)^2} \quad 2.2$$

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

Dalam konteks ini, X merujuk pada variabel *independen*, sementara Y merujuk pada variabel *dependen*, dan n adalah jumlah data yang terdapat dalam analisis regresi linier. Terdapat istilah "R square" yang merupakan koefisien determinasi yang memberikan gambaran sejauh mana data *dependen* dapat dijelaskan oleh data *independen*. Nilai R square berada dalam rentang 0 hingga 1, di mana semakin mendekati angka satu menandakan tingkat kecocokan yang lebih baik [23].

2.2.16 Quality of Service (QoS)

Quality of Service (QoS) adalah kualitas layanan untuk mengukur seberapa baik sebuah jaringan. Parameter analisis jaringan QoS adalah *delay*, *throughput*, *jitter*, dan *packet loss* [2].

2.2.16.1 Delay

Delay adalah waktu yang dibutuhkan data mulai dari saat data ditransmisikan hingga data sampai di tujuan. Adapun [14] rumus untuk menghitung *delay* yaitu:

$$\text{Rata-Rata Delay} = \frac{\text{Total Delay}}{\text{Jumlah paket yang diterima}} \quad 2.3$$

Untuk menghitung *delay* dengan menghitung total *delay* yang didapatkan dari total paket yang diterima. Dengan perhitungan ini didapatkan hasil perhitungan rata-rata *delay*. Dari rumus di atas, nilai *delay* yang dihasilkan selanjutnya akan

dibandingkan dengan tabel indeks *delay* standar TIPHON. Tabel 2.3 merupakan indeks kategori *delay* dengan kategori sangat bagus, bagus, sedang dan jelek.

Tabel 2. 3 Indeks Kategori *Delay* [14]

Kategori	Besar <i>Delay</i>	Indeks
Sangat Bagus	< 150 ms	4
Bagus	150 – 300 ms	3
Sedang	300 – 450 ms	2
Jelek	> 450 ms	1

2.2.16.2 *Jitter*

Jitter menunjukkan variasi *delay* dalam pengiriman data pada suatu jaringan. Nilai *jitter* dapat dipengaruhi oleh antrian trafik. Semakin tinggi nilai *jitter*, berbanding terbalik dengan penurunan nilai QoS. Untuk mencapai QoS yang baik, nilai *jitter* harus dijaga dengan menjaganya sekecil mungkin [14]. Adapun rumus perhitungan untuk menentukan *jitter* yaitu :

$$\text{Jitter} = \frac{\text{Total Variasi Delay}}{\text{Total paket Delay} - 1} \quad 2.4$$

Dari rumus di atas, nilai *jitter* yang dihasilkan selanjutnya akan dibandingkan dengan tabel indeks *jitter* standar TIPHON. Tabel 2.4 merupakan indeks kategori *jitter* dengan kategori sangat bagus, bagus, sedang dan jelek:

Tabel 2. 4 Indeks Kategori *Jitter* [14]

Kategori	Besar <i>Jitter</i>	Indeks
Sangat Bagus	0 ms	4
Bagus	0 – 75 ms	3
Sedang	75 – 125 ms	2
Jelek	125– 225 ms	1

2.2.16.3 *Throughput*

Throughput merupakan jumlah data yang dapat dikirim atau diterima dalam satu satuan waktu. *Throughput* biasanya tergantung pada *bandwidth* dalam kondisi

dunia nyata, tetapi *throughput* lebih dinamis tergantung pada lalu lintas yang terjadi saat *bandwidth* tetap [14]. Berikut merupakan perhitungan yang digunakan untuk menentukan nilai *throughput*, yaitu:

$$Throughput = \left(\frac{\text{Jumlah data yang dikirim}}{\text{Waktu pengiriman data}} \right) \quad 2.5$$

Dari rumus di atas, nilai *throughput* yang dihasilkan selanjutnya akan dibandingkan dengan tabel indeks *throughput* standar TIPHON. Tabel 2.5 merupakan indeks kategori *throughput* dengan kategori sangat bagus, bagus, sedang dan jelek.

Tabel 2. 5 Indeks Kategori *Throughput* [24]

Kategori <i>Throughput</i>	<i>Troughput</i>	Indeks
Sangat Bagus	>2,1 Mb	4
Bagus	1200 kbps-2,1 Mbps	3
Sedang	338-700 kbps	2
Jelek	0-338 kbps	1

2.2.16.4 Packet Loss

Packet loss menggambarkan kegagalan transmisi total paket dalam mencapai tujuannya. Kegagalan tersebut dapat disebabkan oleh beberapa kemungkinan, diantaranya adalah terjadi *collision* dan *congestion* dalam jaringan. terjadinya *overload* pada trafik di dalam jaringan dan *error* yang terjadi pada media fisik [14]. Adapun rumus perhitungan untuk mendapatkan hasil presentase nilai *packet loss* yaitu:

$$Packet Loss = \left(\frac{\text{data yang dikirim-paket data yang diterima}}{\text{paket data yang dikirim}} \right) \times 100 \% \quad 2.6$$

Dari rumus di atas, nilai *packet loss* yang dihasilkan selanjutnya akan dibandingkan dengan tabel indeks *packet loss* standar TIPHON. Tabel 2.6 merupakan indeks kategori *packet loss* dengan kategori sangat bagus, bagus, sedang dan jelek

Tabel 2. 6 Indeks Kategori *Packet Loss* [14]

Kategori	<i>Packet Loss</i>	Indeks
Sangat Bagus	0%	4
Bagus	3%	3
Sedang	15%	2
Jelek	25%	1

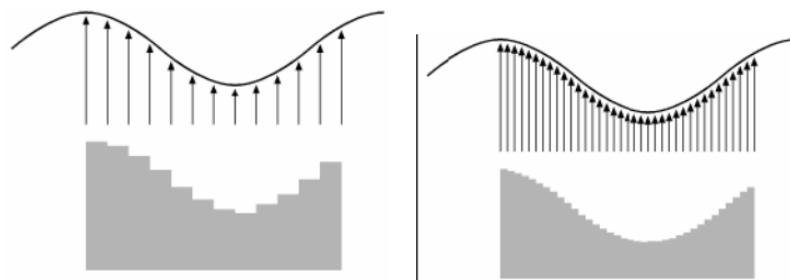
2.2.17 *Analog to Digital Converter (ADC)*

ADC merupakan suatu perangkat konverter yang mengubah sinyal *input* analog menjadi kode digital. Penggunaan ADC sangat luas di berbagai bidang, seperti dalam sistem kontrol proses industri, komunikasi digital, dan perangkat pengukuran dan pengujian. ADC biasanya berperan sebagai perantara antara sensor-sensor yang mayoritas menghasilkan sinyal analog [25].

ADC (*analog to digital converter*) memiliki dua prinsip karakteristik, yaitu kecepatan sampling dan resolusi. Kecepatan sampling pada ADC menunjukkan seberapa sering sinyal analog diubah menjadi bentuk sinyal digital dalam *interval* waktu tertentu. Biasanya, kecepatan pengambilan sampel dinyatakan dalam satuan sampel per sekon (SPS) [25]. Contoh sampling *voltage* dan nilai ADC dari sensor sebagai berikut:

$$Voltage = \frac{\text{nilai ADC dari sensor} * \text{nilai tegangan referensi}}{\text{nilai kapasitas ADC}} \quad 2.7$$

$$\text{Nilai ADC} = \frac{\text{voltage terbaca} * \text{nilai max kapasitas ADC}}{\text{tegangan referensi}} \quad 2.8$$



Gambar 2. 14 ADC dengan kecepatan sampling [25]

Gambar 2.14 merupakan kecepatan sampling dengan kecepatan rendah dan sampling kecepatan tinggi. ADC 12 bit memiliki 12 bit *output* data digital, ini berarti sinyal *input* dapat dinyatakan dalam 4095 ($2^n - 1$) nilai diskrit [25].

2.2.18 Akurasi

Akurasi mencerminkan hubungan yang ideal antara masukan dan keluaran yang konstan sepanjang waktu. Penghitungan akurasi digunakan untuk mengevaluasi sejauh mana sensor dalam sistem yang dirancang menghasilkan hasil yang akurat. Persamaan untuk menghitung nilai akurasi sebagai berikut [26]:

$$Akurasi\ relatif = 1 - \left| \frac{X_n - Y_n}{Y_n} \right| \quad 2.9$$

$$(\%)Akurasi = 100\% - Error\ \% \quad 2.10$$

Dalam konteks ini, A merujuk pada ukuran seberapa akurat suatu metode atau proses dalam memberikan hasil yang tepat. X_n mewakili angka atau data yang telah dihasilkan melalui pengukuran atau evaluasi suatu sistem. Di sisi lain, Y_n mewakili standar atau nilai yang diharapkan sebagai acuan perbandingan yang ideal. Lebih jauh lagi, penggunaan A, X_n , dan Y_n memungkinkan kita untuk menganalisis sejauh mana hasil pengukuran X_n mendekati nilai yang diharapkan Y_n . Jika A mendekati 100%, maka hasil pengukuran X_n secara akurat mencerminkan nilai yang diharapkan Y_n . Namun, jika A lebih rendah, hal ini menunjukkan adanya selisih atau kesalahan antara hasil yang diperoleh (X_n) dan harapan yang diinginkan (Y_n). Dengan demikian, konsep ini memungkinkan untuk mengukur kualitas, keandalan, dan efektivitas suatu proses atau sistem melalui perbandingan antara hasil pengukuran aktual (X_n) dan nilai yang diharapkan (Y_n), yang diwakili oleh tingkat akurasi A.

2.2.19 Measurement Error

Measurement error atau kesalahan pengukuran adalah perbedaan antara nilai sebenarnya dan nilai yang diinginkan. Galat atau *error* merupakan salah satu parameter yang digunakan dalam pengujian sensor dengan tujuan untuk mengevaluasi kinerja data hasil pengukuran sensor. Berikut adalah rumus perhitungan nilai *error* [26] :

$$\text{Absolute Error, } e = |X_n - Y_n| \quad \mathbf{2.11}$$

$$(\%) \text{Error} = \left| \frac{X_n - Y_n}{Y_n} \right| \times 100\% \quad \mathbf{2.12}$$

$$(\%) \text{Rata - rata error} = \left| \frac{d_1 + d_2 + d_3 + \dots + d_x}{\text{Banyak data}} \right| \times 100\% \quad \mathbf{2.13}$$