

## **BAB 2**

### **DASAR TEORI**

#### **2.1. KAJIAN PUSTAKA**

Penelitian [9] Penulis menyatakan bahwa sistem ini mampu melakukan penyiraman otomatis berdasarkan tingkat kelembaban tanah, mengirimkan notifikasi melalui Telegram jika suhu melebihi batas yang ditentukan, mengirimkan notifikasi berisi hasil pembacaan sensor setiap 3 jam memberikan pemberitahuan untuk pemupukan setiap 7 hari serta memberikan notifikasi setelah penyiraman, pengabutan, atau pemupukan dilakukan. Perangkat ini dilengkapi dengan modul WiFi memungkinkan pengguna untuk terhubung dengan internet dengan mudah. Fitur ini memudahkan pemantauan jarak jauh dari lokasi manapun.

Selanjutnya penelitian [10] menjelaskan hasil perancangan dengan menggunakan mikrokontroler NodeMCU 8266 serta sensor *soil moisture* untuk mendeteksi kelembaban tanah dan sensor DHT11 untuk mengukur suhu dan kelembaban udara. Dalam penelitian, alat *smart garden* sangat dibutuhkan oleh pemilik rumah untuk pemantauan jarak jauh secara *real-time* dan untuk menyiram tanaman secara otomatis. Penyiraman otomatis akan dilakukan ketika kelembaban tanah berada pada 0-50% dalam kondisi kering dan kelembaban udara mencapai 87°C.

Penelitian [11] dilakukan selama 30 hari dengan mengukur rata-rata kelembaban tanah pada berbagai waktu. Pada pukul 09.00, kelembaban tanah tercatat sebesar 52%, pada pukul 13.00 sebesar 50%, pada pukul 16.00 sebesar 50,35%, dan pada pukul 20.00 sebesar 48%. Hasil pengujian menunjukkan bahwa *delay* sensor kelembaban tanah adalah 3,48 ms. Pengujian jarak juga dilakukan di mana pada jarak 7 meter, *delay* notifikasi Blynk adalah 0,31 detik dan *delay* notifikasi Telegram adalah 3,45 detik. Pada jarak 10 meter, *delay* notifikasi Blynk adalah 0,36 detik dan *delay* notifikasi Telegram adalah 5,7 detik. Alat ini berfungsi dengan baik dalam operasionalnya. Kualitas layanan (QoS) yang ditawarkan oleh Blynk terbukti lebih baik dibandingkan dengan Telegram. Hal ini menunjukkan bahwa Blynk memberikan pengalaman yang lebih optimal dalam penggunaan alat.

Selanjutnya pada penelitian [12] memanfaatkan input dari sensor kelembaban tanah dengan mendeteksi Tingkat kelembaban dalam tanah. Alat ini menggunakan RTC dan sensor ultrasonik untuk jadwal penyiraman dan pemupukan tanaman. Melalui IoT dan aplikasi Blynk sistem dapat dipantau dan dikendalikan dari *smartphone*. Ketika tanah kering terdeteksi, NodeMCU ESP8266 mengaktifkan pompa melalui *Relay*. Pengujian berhasil dengan sensor kelembaban tanah memberikan input pada NodeMCU dan nilai sensor ditampilkan di layar LCD.

Penelitian [13] menciptakan alat monitoring dan penyiraman untuk petani menggunakan sensor DHT22 sebagai sensor kelembaban tanah, dan kamera OV2640. Alat ini dapat dioperasikan dari jarak jauh dengan mikrokontroler ESP8266 berbasis IoT. Tujuannya adalah memudahkan pemantauan suhu, kelembaban, dan kondisi tanaman secara *real-time*. Pada pengujian awal, saat tanah dalam keadaan kering dengan suhu udara 30°C dan kelembaban udara 67,7%, kamera telah diaktifkan dan mengirimkan gambar visual melalui telepon seluler dengan menggunakan alamat IP yang ditampilkan pada serial monitor di Arduino IDE. Pada uji coba, alat berhasil mengaktifkan katup saat tanah dalam keadaan kering, sementara kamera menampilkan gambaran kondisi tanaman.

Selanjutnya pada penelitian [14] menggunakan *prototype* untuk monitoring lahan pertanian dengan fokus pada suhu dan kelembaban tanah. Menggunakan sensor dan mikrokontroler, data ditampilkan melalui LCD dan terkoneksi ke aplikasi Telegram melalui *Bluetooth*. Simulasi menunjukkan kesesuaian data antara perangkat IoT dan aplikasi Telegram. Teknologi ini diharapkan dapat meningkatkan produksi pertanian menuju era industri 4.0. Selain itu, penelitian ini juga menentukan jarak maksimum komunikasi *Bluetooth* dalam berbagai lingkungan, yaitu 100 m dalam *free space* dan 20 m dalam urban area. Berdasarkan pemilihan nilai eksponen redaman untuk area *free space* dan area perkotaan, yaitu  $n=2$  dan  $n=3,5$ . *Bluetooth* dapat mencapai jarak maksimum komunikasi hingga 100 meter pada kondisi *free space* dengan daya terima -60 dB. Sementara itu, di area perkotaan jarak maksimum komunikasi *Bluetooth* adalah 20 meter dengan daya terima -70 dB, seperti yang ditunjukkan dalam persamaan dan grafik yang dihasilkan.

**Tabel 2. 1 Kajian Pustaka**

| Parameter Penelitian  | [9]  | [10]  | [11]                        | [12]  | [13]   | [14]   | Rencana Penelitian                              |
|-----------------------|--|---|-----------------------------|---|--|--|---|
| <i>Microprocessor</i> | ESP8266  | ESP8266   | ESP8266                     | ESP8266   | ESP8266                                      | STM32  | ESP8266   |
| Sensor                | Sensor <i>soil moisture</i><br>Sensor ultrasonik<br>Sensor DS18B20 | Sensor <i>soil moisture</i> ,<br>DHT11                  | Sensor <i>soil moisture</i> | Sensor <i>Capacitive soil moisture</i> ,<br>Sensor Ultrasonik | Sensor DHT22,<br>Sensor <i>soil moisture</i> | Sensor DHT11,<br>Sensor <i>soil moisture</i> | Sensor <i>soil moisture</i> ,<br>Sensor DS18B20 |
| Komunikasi            | MQTT,<br><i>Node red</i><br>dan<br>Telegram<br><i>bot</i>          | Blynk   | Telegram<br>dan<br>Blynk    | Blynk   | -  | Telegram dan<br><i>Bluetooth</i>             | MQTT dan<br><i>Website</i><br>TelkomIoT         |
| Objek penelitian      | Cabai  | <i>Smart garden</i><br>pada Central<br>Park<br>Cikarang | Bunga<br>mawar              | <i>Smart garden</i>   | <i>Smart garden</i>                          | <i>Smart farming</i>                         | Tanaman jahe                                    |

| Parameter Penelitian | [9]                                      | [10]  | [11]                        | [12]                              | [13]                      | [14]                                    | Rencana Penelitian   |
|----------------------|--|---|-----------------------------|-----------------------------------|---------------------------|---|--|
| Parameter pengujian  | Suhu, Kelembaban dan Penyiraman otomatis | Pencahayaann otomatis, suhu, Kelembaban dan Penyiraman otomatis | Kelembaban dan <i>Delay</i> | Penyiraman dan Pemupukan otomatis | Suhu dan Kelembaban Tanah | Suhu, Kelembaban dan <i>Link budget</i> | Suhu tanah, Kelembaban tanah, Qos ( <i>Delay</i> , Paket <i>length</i> dan Paket <i>loss</i> ) dan Penyiraman otomatis |

## **2.2. DASAR TEORI**

Dalam kerangka teoritis yang mendukung penelitian ini terdapat beberapa dasar yang relevan, seperti:

### **2.2.1. TANAMAN HORTIKULTURA BIOFARMAKA**

Tanaman hortikultura adalah jenis tanaman yang umumnya cocok untuk ditanam di pekarangan rumah atau kebun kecil termasuk sayuran, buah-buahan, tanaman hias, dan tanaman obat. Tanaman hortikultura memiliki banyak manfaat antara lain [15]:

1. Sayuran dan buah-buahan sebagai sumber makanan. Tanaman sayuran seperti tomat, selada, wortel, dan buah-buahan seperti stroberi, jeruk, dan apel memberikan sumber makanan yang segar, sehat, dan bergizi bagi manusia. Tanaman hortikultura mengandung banyak vitamin, mineral, dan serat yang penting untuk memelihara kesehatan tubuh.
2. Tanaman hias untuk mempercantik lingkungan, seperti bunga-bunga, tanaman merambat, dan tanaman *indoor* digunakan untuk memperindah lingkungan sekitar. Tanaman hias menciptakan nuansa yang menyegarkan, menarik serangga penyerbuk, serta meningkatkan kualitas udara dan estetika ruang.
3. Tanaman obat untuk pengobatan alami, seperti lidah buaya, jahe, dan kemangi memiliki khasiat penyembuhan dan dapat digunakan sebagai alternatif pengobatan alami untuk berbagai penyakit dan keluhan kesehatan. Tanaman obat mengandung senyawa-senyawa aktif yang bermanfaat bagi kesehatan tubuh dan dapat membantu mengurangi ketergantungan pada obat-obatan kimia.

Tanaman biofarmaka merupakan tanaman yang digunakan sebagai bahan obat-obatan, rempah-rempah dalam masakan, serta bahan dasar alami untuk kosmetik. Di Indonesia, beberapa jenis tanaman biofarmaka dimanfaatkan sebagai ramuan jamu yang memiliki beragam manfaat bagi kesehatan tubuh, seperti meningkatkan energi dan memperkuat sistem kekebalan tubuh untuk melawan penyakit. Terdapat 15 jenis tanaman biofarmaka yang diproduksi di Indonesia dapat dikelompokkan menjadi dua jenis utama, yaitu tanaman biofarmaka jenis rimpang

dan non-rimpang. Tanaman biofarmaka jenis rimpang mencakup jahe, laos atau lengkuas, kencur, kunyit, lempuyang, temulawak, temu ireng, temu kunci, dan *dringo*. Sedangkan tanaman biofarmaka jenis non-rimpang meliputi kapulaga, kejibeling, mengkudu, sambiloto, dan lidah buaya [16].

Tanaman jahe merupakan jenis tanaman rimpang memiliki ciri khas rasa pedas yang dominan yang berasal dari senyawa keton bernama *zingeron*. Rimpang jahe banyak dimanfaatkan sebagai rempah-rempah dalam berbagai masakan dan minuman penghangat badan.



**Gambar 2.1 Ilustrasi Tanaman Jahe [17]**

Gambar 2.1 menampilkan bagian-bagian dari tanaman jahe termasuk batang, daun, dan rimpang. Batang jahe berbentuk bulat dengan ukuran kecil dan berwarna hijau. Bagian bawah batangnya memiliki warna kemerahan dan tinggi tanaman biasanya antara 34,18 hingga 62,28 cm. Daun jahe tersusun secara teratur secara bergantian dengan warna hijau gelap di bagian atas dan hijau muda di bagian bawahnya. Rimpang jahe berwarna merah hingga jingga muda dengan ukuran sekitar 12,33 hingga 12,60 cm [17].

Tanaman jahe membutuhkan tanah yang subur, gembur, kaya akan bahan organik, dan memiliki *drainase* yang baik. Tanah rawa, tanah berat, serta tanah berpasir kasar atau berkerikil tidak cocok untuk pertumbuhan jahe. Tekstur tanah yang ideal adalah lempung berpasir, lempung berliat, liat berpasir, dan liat berdebu. pH tanah yang optimal untuk jahe adalah antara 6,8 hingga 7,0. Jahe dapat tumbuh pada ketinggian 0 hingga 1500 meter di atas permukaan laut (mdpl) namun, pertumbuhannya yang optimal terjadi antara 300 hingga 900 mdpl [18]. Suhu udara

juga mempengaruhi proses fisiologis tanaman. Tanaman jahe tumbuh optimal pada suhu 20-30°C. Peningkatan suhu selama fase vegetatif dapat menyebabkan penuaan dini pada tanaman, daun kehilangan warna hijau sehingga tidak mampu berfotosintesis, dan kondisi terburuk dapat menyebabkan kematian tanaman. Suhu yang berada di bawah optimal dalam jangka waktu lama memperlambat fase pertumbuhan dan proses metabolisme yang mengakibatkan siklus hidup lebih lama serta kerusakan hingga kematian tanaman.

Unsur iklim lainnya yaitu kelembaban udara juga berpengaruh pada proses fisiologis tanaman. Tanaman jahe membutuhkan kelembaban udara sekitar 60-90%. Kelembaban yang rendah meningkatkan laju transpirasi sehingga akar menyerap lebih banyak unsur esensial dan air dapat meningkatkan pertumbuhan tanaman. Namun, jika tidak diimbangi dengan ketersediaan air yang cukup tanaman dapat layu. Sebaliknya, kelembaban yang tinggi menurunkan laju transpirasi mengurangi kemampuan akar menyerap unsur esensial dan air yang dapat mengganggu pertumbuhan tanaman [7].

### 2.2.2. ARDUINO IDE

*Integrated Development Environment (IDE)* suatu wadah yang menyatukan berbagai alat yang dibutuhkan dalam pengembangan perangkat lunak. Perangkat lunak seperti Arduino diprogram untuk melakukan fungsi-fungsi khusus menggunakan *syntax* pemrograman. Arduino menggunakan bahasa pemrograman yang mirip dengan C. Arduino digunakan untuk mengembangkan dan menulis program untuk mikrokontroler Arduino [19]. Gambar 2.2 tampilan awal ketika aplikasi Arduino IDE dijalankan.



**Gambar 2. 2 Aplikasi Arduino IDE [20]**

Bahasa pemrograman Arduino disebut sebagai *sketch* telah disesuaikan agar lebih mudah dipahami oleh pemula dibandingkan dengan bahasa aslinya. Sebelum dijual ke pasaran IC mikrokontroler Arduino sudah diprogram dengan sebuah program yang disebut *bootloader*. *Bootloader* berfungsi sebagai perantara antara *compiler* Arduino dan mikrokontroler yang digunakan. Arduino IDE dikembangkan menggunakan bahasa pemrograman JAVA dan dilengkapi dengan perpustakaan C/C++ juga dikenal sebagai *wiring* untuk menyederhanakan operasi input dan *output*. Awalnya, Arduino IDE berasal dari perangkat lunak bernama *processing* yang kemudian dimodifikasi menjadi Arduino IDE khusus untuk pemrograman dengan Arduino. Program dibuat menggunakan Arduino *Software* (IDE) disebut sebagai *sketch*. *Sketch* ditulis dalam editor teks dan disimpan dalam file dengan ekstensi tertentu [20].

### 2.2.3. NODEMCU ESP8266

NodeMCU adalah sebuah *board* elektronik yang menggunakan *chip* ESP8266 sebagai basisnya. *Board* memiliki kemampuan untuk berfungsi sebagai mikrokontroler dan menyediakan koneksi internet melalui WiFi. NodeMCU dilengkapi dengan beberapa pin I/O yang memungkinkan pengembangan aplikasi monitoring dan kontrol dalam proyek IoT. Mikrokontroler NodeMCU ESP8266 dapat diprogram menggunakan *compiler* Arduino melalui Arduino IDE. Fisik NodeMCU ESP8266 dilengkapi dengan *port* USB (miniUSB) untuk mempermudah proses pemrograman. NodeMCU ESP8266 adalah turunan modul IoT berbasis ESP8266 tipe ESP-12 yang difokuskan pada kemampuan koneksi internet mirip dengan modul Arduino dalam hal fungsi dasar. Saat ini, terdapat tiga tipe versi dari modul NodeMCU [21].

#### Versi NodeMCU ESP8266



Gambar 2. 3 NodeMCU ESP8266 [22]

NodeMCU ESP8266 menjadi *platform IoT open source* yang didasarkan pada *system on chip* ESP8266. Pada Gambar 2.3 terdapat versi dari NodeMCU ESP8266 yang mengalami tiga kali peningkatan dengan versi terbarunya NodeMCU V3 (V1.0) yang memiliki kemampuan lebih baik dari versi sebelumnya.

**Tabel 2. 2 Spesifikasi NodeMCU ESP8266 [22]**

| Parameter           | Komponen           |
|---------------------|--------------------|
| Mikrokontroler      | ESP 8266           |
| Tegangan input      | 3.3 ~ 5V           |
| GPIO                | 17 Pin             |
| <i>Flash Memory</i> | 16 MB              |
| RAM                 | 32 KB + 80 KB      |
| Konsumsi Daya       | 10 uA ~ 170 mA     |
| Frekuensi           | 2,4 GHz – 22,5 Ghz |
| USB <i>Port</i>     | Micro USB          |
| WiFi                | IEEE 802.11b/g/n   |
| Kanal PWM           | 10 Kanal           |
| USB <i>Chip</i>     | CH340G             |
| <i>Clock Speed</i>  | 40/26/24 MHz       |

Tabel 2.2 NodeMCU ESP8266 memiliki spesifikasi 17 pin GPIO yang bisa digunakan untuk menghubungkan dengan komponen elektronika lainnya. Operasinya mendukung tegangan antara 3,3 volt hingga 5 volt dengan konsumsi daya yang bervariasi antara 10 uA hingga 170 mA. NodeMCU V1 memiliki kecepatan prosesor antara 80 hingga 160 megahertz, RAM 32 KB ditambah 80 KB, dan memori *flash* hingga 16 MB membuatnya lebih efisien dibandingkan versi sebelumnya [22].

#### **2.2.4. SENSOR SOIL MOISTURE**

Sensor *soil moisture* adalah sensor untuk mengukur tingkat kelembaban tanah. Biasanya digunakan dalam pertanian untuk memantau kebutuhan air tanaman. Sensor ini terdiri dari elektroda yang dimasukkan ke tanah dan menggunakan konduktivitas listrik untuk mengukur kelembaban. Data yang

diperoleh membantu dalam mengatur irigasi dan manajemen tata air untuk pertanian yang lebih efisien. Ketika tanah mengandung lebih banyak air konduktivitas listriknya lebih tinggi, dan sebaliknya. Data sensor dapat digunakan untuk mengawasi dan mengatur irigasi tanaman, mencegah kekeringan atau kelembaban berlebihan, serta mendukung manajemen air dan pertanian yang efisien [23].



**Gambar 2. 4 Sensor Soil moisture v1.2 [24]**

Gambar 2.4 menampilkan sensor dengan *probe* untuk mengalirkan arus melalui tanah dan membaca resistansinya untuk mengukur kelembaban. Kadar air yang tinggi membuat resistansi rendah, sedangkan tanah kering memiliki resistansi tinggi. Sensor ini memiliki tiga pin: *output* analog (biru), *ground* (hitam), dan *power* (merah) masing-masing dengan fungsinya sendiri. Sensor *soil moisture* membutuhkan tegangan input antara 3,3 V atau 5 V, dan menghasilkan keluaran tegangan antara 0 hingga 4,2 V dalam aplikasinya [24].

#### 2.2.5. SENSOR DS18B20

Sensor suhu DS18B20 adalah komponen yang penting dalam proyek-proyek yang membutuhkan pemantauan suhu berkat tingkat akurasi tinggi dan rentang suhu yang luas membuat sangat berguna untuk berbagai aplikasi.



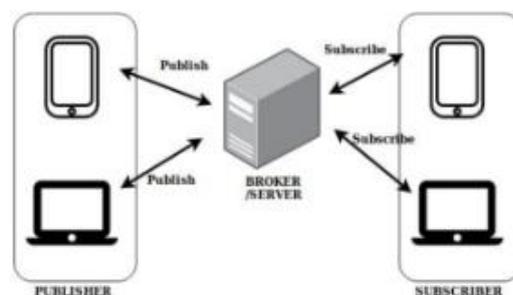
**Gambar 2. 5 Sensor DS18B20 [25]**

Pada Gambar 2.5 sensor DS18B20 memiliki tiga pin yang harus dihubungkan dengan benar untuk pengoperasiannya. Pin *Voltage Supply* (Vs) digunakan sebagai kaki tegangan sumber, dan tegangan kerjanya berkisar antara 3V hingga 5,5V. Biasanya, Vs diberi tegangan +5V sesuai dengan kebutuhan mikrokontroler. Selanjutnya, pin *ground* harus dihubungkan dengan *ground* rangkaian untuk menciptakan referensi tegangan yang stabil. Pin data input atau *output* digunakan untuk berkomunikasi antara sensor DS18B20 dan mikrokontroler. Informasi suhu yang telah dikonversi ke dalam format digital akan dikirim melalui pin ini [25]. Sensor DS18B20 memiliki beberapa fitur utama sebagai berikut [26]:

1. Menggunakan satu kabel untuk komunikasi (protokol *unique 1-wire*).
2. Memiliki ROM *onboard* yang menyimpan kode serial 64-bit.
3. Tidak membutuhkan tambahan komponen.
4. Mendukung rentang daya 3 hingga 5,5 volt.
5. Rentang pengukuran suhu dari 55°C hingga 125°C.
6. Akurasi  $\pm 0,5^\circ\text{C}$  pada rentang 10°C hingga +85°C.
7. Pilihan resolusi sensor dari 9 hingga 12 bit.
8. Konversi suhu ke format digital 12 bit dalam waktu maksimal 750 ms.

### 2.2.6. PROTOKOL MQTT

*Message Queuing Telemetry Transport* (MQTT) adalah protokol komunikasi *Machine-to-Machine* (M2M) yang ringan dan efisien. Konsep utamanya adalah *publish* dan *subscribe*, di mana informasi dipublikasikan oleh penerbit (*publisher*) dan disalurkan oleh *broker* ke penerima (*subscriber*) yang tertarik pada topik tersebut.



Gambar 2. 6 Arsitektur MQTT [27]

Pada Gambar 2.6 arsitektur MQTT melibatkan tiga komponen utama: *publisher*, *broker*, dan *subscriber*. Perangkat yang ingin terhubung ke suatu topik harus mendaftar sebagai *subscriber* pada topik tersebut di *broker*. Ketika informasi dipublikasikan oleh *publisher*, maka *broker* meneruskannya kepada semua *subscriber* pada topik tersebut. MQTT memungkinkan komunikasi efisien dan fleksibel antara perangkat dalam jaringan M2M mendukung implementasi sistem *scalable* dan andal untuk aplikasi IoT dan M2M [27].

Dalam pola pesan *publish-subscribe*, *broker* pesan diperlukan untuk mendistribusikan pesan sesuai dengan topik kepada klien. MQTT memiliki beberapa metode yang dapat dijelaskan sebagai berikut [28]:

1. *Connect*: menunggu koneksi yang stabil dengan server.
2. *Disconnect*: menunggu klien untuk menyelesaikan tugasnya sebelum memutuskan sambungan TCP/IP.
3. *Subscribe*: menunggu penyelesaian proses langganan atau pembatalan langganan.
4. *Unsubscribe*: meminta server untuk menghentikan langganan klien dari satu atau lebih topik.

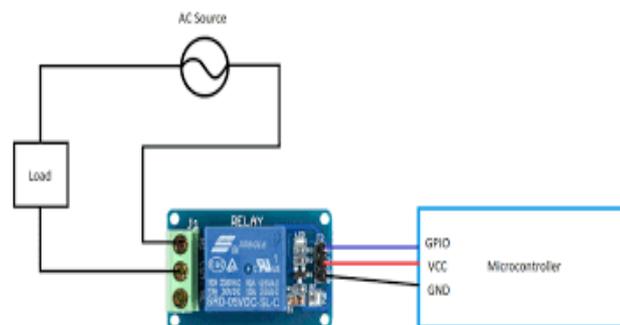
*Hypertext Transfer Protokol* (HTTP) adalah protokol IoT pertama yang diakui yang menggunakan arsitektur permintaan atau respons dalam model klien-server. Protokol ini terutama digunakan untuk menyebarkan server web, yang menyediakan data yang diminta oleh *browser*. Melalui HTTP data tersebut dikirimkan dalam format *HyperText Markup Language* (HTML). Komunikasi selalu dimulai oleh *browser* dengan mengajukan permintaan dokumen HTML ke server. Setelah itu, dokumen diproses dan permintaan tambahan dikirimkan untuk skrip, *Cascading Style Sheets* (CSS), serta berbagai elemen lainnya.

Keuntungan dari protokol MQTT adalah efisiensinya dalam penggunaan energi, terutama jika menggunakan *broker* berbasis *cloud* seperti yang digunakan dalam *prototype*. Dalam protokol MQTT model *device-to-gateway* diterapkan di mana *broker* berbasis *cloud* Adafruit IO berfungsi sebagai *gateway* berbasis *cloud*. Pada situasi lain diperlukan mikrokontroler tambahan sebagai *broker* untuk mengirimkan informasi [29]. MQTT menawarkan solusi yang lebih efisien daripada HTTP menjadikannya pilihan menarik ketika perlu mentransfer data

hampir secara *real-time* dengan pola komunikasi *publish-subscribe* dan kebutuhan akan jejak yang minimal. MQTT solusi efisien untuk mentransfer data hampir *real-time* dengan pola komunikasi *publish-subscribe* lebih baik daripada HTTP. Digunakan luas di IoT, M2M, dan proyek tertanam juga dalam aplikasi web serta seluler untuk pesan andal dan distribusi yang efisien. Dirancang sebagai protokol pesan ringan cocok untuk perangkat jarak jauh dengan kode minimal dan kebutuhan *bandwidth* rendah. MQTT telah merambah ke berbagai industri, termasuk otomotif, manufaktur, telekomunikasi, serta sektor minyak dan gas [30].

### 2.2.7. RELAY

*Relay* adalah komponen elektronik yang berfungsi sebagai saklar yang dapat dikendalikan oleh arus listrik. Digunakan untuk mengontrol arus atau tegangan yang besar, seperti peralatan listrik dengan rating 4A/AC 220V menggunakan arus atau tegangan yang lebih kecil misalnya 0,1A/12V DC. Dengan menggunakan *Relay* dapat mengalihkan atau mengontrol daya listrik yang besar dengan menggunakan daya yang lebih kecil sehingga memungkinkan penggunaan peralatan elektronik yang lebih efisien dan aman [31].



**Gambar 2. 7 Modul Relay Single Channel [22]**

Pada Gambar 2.7 *relay* memiliki empat komponen utama yaitu, elektromagnet (*coil*), *armature*, saklar, dan pegas. Elektromagnet (*coil*) yang menghasilkan medan magnet ketika dialiri arus listrik dan berfungsi untuk menggerakkan *armature*. *Armature* yaitu tuas yang bergerak ketika elektromagnet aktif untuk membuka atau menutup saklar. Saklar yang mengalirkan atau memutuskan arus listrik dalam rangkaian yang dikendalikan oleh gerakan *armature*. Pegas yang mengembalikan *armature* ke posisi semula ketika

elektromagnet tidak aktif, memastikan saklar kembali ke kondisi awal. Terdapat dua jenis kondisi kontak *relay*; *Normally Close* (NC) dan *Normally Open* (NO). Pada kondisi NC, kontak *relay* berada dalam posisi tertutup atau terhubung sebelum diaktifkan. Sedangkan pada kondisi NO, kontak *relay* berada dalam posisi terbuka atau tidak terhubung sebelum diaktifkan [22].

*Relay* bekerja dengan cara mengalihkan posisi kontaknya ketika diberi tegangan pada terminal kaki 1 dan terminal kaki *ground* pada kaki 2. Posisi kontak CO pada *Relay* akan beralih dari NC ke NO. *Relay* dapat dianggap sebagai saklar elektronik yang diaktifkan oleh arus listrik. Ketika arus mengalir melalui lilitan kawat solenoid menghasilkan gaya magnet untuk menarik tuas menutup kontak saklar. Saat arus dihentikan gaya magnet menghilang dan kontak saklar terbuka kembali. *Relay* digunakan untuk mengendalikan arus atau tegangan besar dengan menggunakan arus atau tegangan lebih kecil [32].

#### **2.2.8. POWER SUPPLY**

*Power supply* atau catu daya adalah perangkat yang menyediakan tegangan atau sumber daya listrik untuk peralatan elektronika. Fungsinya adalah mengubah tegangan listrik dari jaringan distribusi transmisi ke tingkat yang sesuai dengan kebutuhan perangkat yang digunakan. Proses ini mencakup beberapa tahap, seperti penyearahan arus AC ke DC penyaringan untuk menghilangkan riak pada tegangan, dan pengaturan tegangan untuk memastikan keluaran yang stabil. Dengan demikian, *power supply* memastikan perangkat elektronik menerima daya dengan spesifikasi yang tepat untuk beroperasi secara efisien dan aman. Hal ini menghasilkan perubahan daya listrik yang berimplikasi pada perubahan daya. Dalam proses konversi daya, terdapat empat jenis proses yaitu [33]:

1. AC ke DC: Mengubah arus bolak-balik (AC) menjadi arus searah (DC).
2. DC ke DC: Mengubah tegangan atau arus searah dari satu level ke level lain.
3. DC ke AC: Mengubah arus searah (DC) menjadi arus bolak balik (AC).
4. AC ke AC: Mengubah tegangan atau arus bolak-balik dari satu level ke level lain.

*Power supply* secara otomatis digunakan untuk menyediakan daya untuk perangkat. Sumber daya eksternal untuk Arduino Uno yang tidak menggunakan

koneksi USB dapat berasal dari adaptor AC ke DC atau baterai. Adaptor AC ke DC dapat dihubungkan langsung ke soket daya yang ada pada Arduino Uno, di mana adaptor tersebut mengubah arus AC dari jaringan listrik menjadi arus DC yang sesuai dengan kebutuhan Arduino. Jika menggunakan baterai sebagai sumber daya, kabel dari baterai harus dihubungkan ke pin GND dan Vin pada konektor daya Arduino. Pin GND berfungsi sebagai *ground* atau negatif, sedangkan pin Vin menerima tegangan positif dari baterai. Dengan kedua metode ini, Arduino dapat dijalankan tanpa bergantung pada sumber daya dari *port* USB. Arduino Uno dapat berfungsi pada rentang tegangan antara 6 hingga 20 volt. Jika tegangan yang disuplai ke Arduino Uno kurang dari 7 volt, pin 5V pada *board* Arduino akan menghasilkan tegangan yang kurang dari 5 volt. Hal ini dapat menyebabkan kinerja Arduino Uno menjadi tidak stabil karena kebutuhan tegangan yang tidak terpenuhi secara optimal. Tegangan yang stabil dan mencukupi sangat penting untuk memastikan operasi mikrokontroler dan komponen elektronik lainnya berjalan dengan baik. Hal ini juga diperlukan agar perangkat tersebut bekerja sesuai dengan spesifikasi yang direkomendasikan. Oleh karena itu, memastikan bahwa sumber daya yang digunakan memenuhi persyaratan tegangan minimal yang disarankan untuk Arduino Uno sangat penting. Tegangan yang memadai memastikan perangkat berfungsi dengan baik dan stabil. Dengan tegangan yang tepat, Arduino Uno dapat bekerja sesuai dengan spesifikasi yang direkomendasikan. Penggunaan sumber daya yang sesuai juga meningkatkan keandalan dan umur panjang perangkat. Pin-pin catu daya pada Arduino Uno meliputi [33]:

1. Vin: Pin digunakan untuk memasok tegangan ke Arduino Uno saat menggunakan sumber daya eksternal, seperti adaptor AC ke DC atau baterai. Tegangan juga dapat diberikan melalui pin ini jika sumber daya dialirkan melalui soket daya Arduino Uno.
2. 5V: Penyedia tegangan teregulasi sebesar 5 volt yang berasal dari regulator tegangan yang ada di Arduino Uno.
3. 3V3: Penyedia tegangan teregulasi sebesar 3,3 volt yang berasal dari regulator tegangan pada Arduino Uno.
4. GND: Pin ini merupakan *ground* yang digunakan sebagai referensi untuk tegangan lainnya.

### 2.2.9. AKURASI

Akurasi adalah seberapa dekat nilai prediksi dengan nilai aktual menggunakan metode perbandingan. Metode perbandingan kalibrasi melibatkan perbandingan antara pembacaan dari alat ukur yang akan dikalibrasi dengan pembacaan dari standar referensi yang diketahui dan akurat [34]. Presisi adalah ukuran yang menunjukkan seberapa konsisten hasil pengukuran ketika dilakukan berulang kali. Akurasi dapat ditentukan dengan satu kali pengukuran, sedangkan presisi membutuhkan pengukuran berulang untuk mengetahui seberapa dekat hasil-hasil pengukuran satu sama lain [35].

Berikut formula untuk menghitung akurasi ditunjukkan pada persamaan (2.3) sebagai berikut [36]:

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} \quad (2.1)$$

$$error (\%) = \left| \frac{x_{ref} - \bar{x}}{x_{ref}} \right| \times 100\% \quad (2.2)$$

$$Akurasi = 100\% - error (\%) \quad (2.3)$$

Keterangan:

$\bar{x}$  = Rata-rata dari sejumlah n pengukuran  $x_n$

$x_{ref}$  = Nilai referensi atau sebenarnya

$n$  = Jumlah nilai sensor

### 2.2.10. ANALOG to DIGITAL CONVERTER (ADC)

*Analog to Digital Converter* (ADC) adalah perangkat elektronik yang berfungsi untuk mengubah sinyal analog menjadi sinyal digital. ADC dapat berupa modul atau rangkaian elektronik, atau bahkan sebuah *chip* IC. ADC sering digunakan sebagai penghubung antara sensor-sensor yang umumnya menghasilkan sinyal analog dengan sistem komputer. Contohnya, sensor suhu, cahaya, kelembaban, dan lain sebagainya. Sistem mengukur sinyal analog tersebut menggunakan *converter* ADC memiliki dua karakteristik utama, yaitu kecepatan sampling dan resolusi.

Kecepatan sampling dari sebuah ADC menunjukkan frekuensi konversi sinyal analog ke sinyal digital dalam suatu interval waktu tertentu biasanya

dinyatakan dalam satuan sampel per detik. Resolusi ADC mengukur tingkat ketelitian dari hasil konversi sinyal tersebut. Dengan kata lain, resolusi lebih tinggi berarti hasil konversi akan lebih akurat dan mendetail. Kedua aspek sangat penting dalam memastikan bahwa sinyal digital yang dihasilkan adalah representasi dari sinyal analog aslinya. Kecepatan sampling yang tinggi memungkinkan penangkapan perubahan cepat dalam sinyal analog sementara resolusi tinggi memastikan bahwa setiap sampel memiliki tingkat presisi yang tinggi [37].

ADC digunakan untuk mengubah sinyal analog menjadi sinyal digital. Ada berbagai teknik untuk mengubah sinyal analog menjadi sinyal digital yang tercantum sebagai berikut [38]:

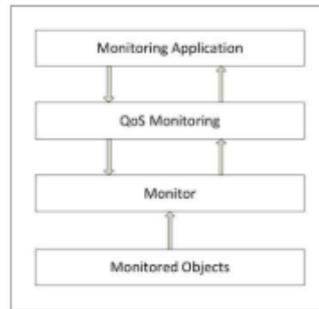
1. *Flash* ADC
2. ADC Perkiraan berurutan
3. ADC *ramp* bandingkan
4. *Wilkinson* ADC
5. ADC *Sigma-delta*
6. ADC *Pipeline*

Sinyal digital yang dihasilkan ADC berupa bilangan basis yang terdiri dari angka 0 dan 1. *Output* sinyal harus dapat mempresentasikan sinyal analog yang diterjemahkannya. Representasi akan semakin baik ketika ADC semakin sensitif terhadap perubahan nilai sinyal analog yang masuk. Misalnya, jika nilai 0-15 volt dapat diubah menjadi digital dengan skala 1 volt berarti rentang nilai digital yang diperoleh berupa 16 tahap dari 0 bertahap naik 1 volt hingga nilai 15 atau setara dengan 0000 sampai 1111. Jumlah tahapan ini dapat diperoleh dengan membuat rangkaian ADC 4 bit, karena jumlah bit ( $n$ ) merepresentasikan  $2^n$  nilai skala sehingga  $2^4 = 16$  skala [39].

### **2.2.11. QUALITY of SERVICE (QoS)**

*Quality of Service (QoS)* adalah metode untuk menilai kualitas jaringan dengan mendefinisikan karakteristik dan sifat layanan tertentu. QoS mengukur berbagai atribut kinerja yang telah ditetapkan dan terkait dengan layanan tersebut. Model pemantauan QoS pada Gambar 2.8 terdiri dari beberapa komponen: *monitoring application*, QoS monitoring, monitor, dan *monitored object*.

*Monitoring application* adalah antarmuka untuk administrator jaringan yang mengambil, menganalisis, dan mengirimkan informasi lalu lintas paket data kepada pengguna. QoS monitoring menyediakan mekanisme pemantauan dengan mengumpulkan nilai parameter QoS dari lalu lintas paket data. Monitor mengukur aliran paket data secara *real-time* dan melaporkan hasilnya kepada *monitoring application* [40].



**Gambar 2. 8 Model Monitoring QoS [40]**

Performa mengacu pada tingkat kecepatan dan keandalan dalam menyampaikan berbagai jenis beban data dalam komunikasi. Performa mencakup sejumlah parameter teknis yang dapat diukur, yaitu [41]:

1. *Throughput*

*Throughput* adalah kecepatan transfer data efektif yang diukur dalam bps. *Throughput* mengacu pada jumlah total paket yang berhasil diterima di tujuan selama interval waktu tertentu dibagi dengan durasi interval tersebut. Pada Tabel 2.3 menjadi parameter *throughput*.

**Tabel 2. 3 Parameter *Throughput***

| Kategori Digerasi | Nilai <i>Throughput</i> |
|-------------------|-------------------------|
| Sangat Bagus      | 100                     |
| Bagus             | 75                      |
| Sedang            | 50                      |
| Jelek             | <25                     |

2. Paket *Loss*

Parameter paket *loss* menggambarkan situasi dimana sejumlah paket data hilang akibat bentrokan (*collision*) atau kemacetan (*congestion*) dalam jaringan,

yang mempengaruhi semua aplikasi yang bergantung pada jaringan tersebut. Kehilangan paket dapat berdampak negatif pada efisiensi jaringan secara keseluruhan, karena paket yang hilang harus dikirim ulang yang pada gilirannya mengurangi kinerja jaringan meskipun *bandwidth* yang tersedia mungkin cukup. Biasanya, perangkat jaringan memiliki *buffer* untuk menyimpan data yang diterima sementara menunggu untuk diproses. Namun, jika kemacetan jaringan berlangsung terlalu lama *buffer* akan penuh menyebabkan data baru tidak dapat diterima dan akhirnya paket data tersebut akan hilang. Tabel 2.4 menjadi kategori parameter paket *loss* berdasarkan besar yang dihasilkan oleh paket *loss*.

**Tabel 2. 4 Parameter Paket Loss**

| Kategori Digeradasari | Besar Paket Loss |
|-----------------------|------------------|
| Sangat Bagus          | 0%               |
| Bagus                 | 3%               |
| Sedang                | 15%              |
| Jelek                 | 25%              |

### 3. *Delay (Latency)*

*Delay* atau *latency* adalah waktu yang diperlukan data untuk berpindah dari sumber ke tujuan. Faktor-faktor yang mempengaruhi *delay* termasuk jarak fisik, jenis media transmisi seperti kabel tembaga, serat optik, atau gelombang radio, kemacetan jaringan, dan durasi waktu pemrosesan data. Jarak yang lebih jauh media dengan karakteristik tertentu, kemacetan akibat lalu lintas data yang tinggi, serta proses yang tidak efisien dapat memperpanjang *delay*. Mengelola faktor-faktor ini penting untuk meningkatkan efisiensi dan kinerja komunikasi data. Tabel 2.5 merupakan kategori *delay* berdasarkan besaran *delay* yang diperoleh.

**Tabel 2. 5 Parameter Delay**

| Kategori Digeradasari | Besar Delay    |
|-----------------------|----------------|
| Sangat Bagus          | <150 ms        |
| Bagus                 | 150 s/d 300 ms |
| Sedang                | 300 s/d 450 ms |
| Jelek                 | >450 ms        |