

BAB 2

TINJAUAN PUSTAKA

Pada bab ini membahas berbagai konsep dasar dan teori-teori yang berkaitan dalam pembangunan sistem *monitoring greenhouse* berbasis IoT menggunakan LoRa dan MQTT. Konsep dan teori tersebut diantaranya adalah *Greenhouse*, Mikrokontroler, NodeMCU ESP88, Sensor DHT-11, Sensor LDR, *Internet of Things* (IoT), LoRA, MQTT dan *Unified Modeling Language* (UML).

1.1 Greenhouse

Greenhouse merupakan sebuah bangunan terbuat dari kaca atau plastik yang berkerangka atau dibentuk menggelembung, diselubungi bahan bening atau tembus cahaya yang dapat meneruskan cahaya secara optimum untuk produksi dan melindungi tanaman dari kondisi iklim yang merugikan bagi pertumbuhan tanaman [1]. *Greenhouse* digunakan untuk pengembangbiakan tumbuhan baik untuk tujuan riset maupun intensifikasi pertanian. Intensitas cahaya, suhu dan kelembaban udara merupakan faktor pendukung bagi pertumbuhan tanaman. Suhu dan kelembaban di dalam ruangan *greenhouse* lebih tinggi dibanding di luar *greenhouse* karena udara yang masuk terhalang oleh dinding dan atap *greenhouse*. Intensitas cahaya yang berlebihan dapat menyebabkan suhu ruangan *greenhouse* meningkat yang dapat menyebabkan tanaman menjadi layu. Suhu dan kelembaban udara di dalam *greenhouse* sulit untuk dipantau dengan kasat mata oleh manusia secara langsung dikarenakan *greenhouse* memerlukan perlakuan khusus untuk menjaga suhu dan kelembabannya. Suhu ideal pada *greenhouse* adalah 25 sampai 27 derajat celcius dan untuk kelembaban udara adalah 50% sampai 70%.

1.2 Mikrokontroler

Mikrokontroler pada suatu rangkaian elektronik berfungsi sebagai pengendali yang mengatur jalannya proses kerja dari rangkaian elektronik. Di dalam sebuah IC

mikrokontroler terdapat CPU, memori, timer, saluran komunikasi serial dan paralel, port *input/output*, dan ADC. Terdapat beberapa contoh penggunaan mikrokontroler pada sistem elektronik modern seperti pada mesin mobil, *keyboard*, komputer, televisi, robot, sistem otomasi, sistem keamanan, dan lain-lain [6].

1.3 NodeMCU ESP8266

NodeMCU merupakan sebuah open source platform IoT dan pengembangan kit yang menggunakan bahasa pemrograman Lua untuk membantu dalam membuat *prototype* produk IoT atau bisa dengan memakai *sketch* dengan Arduino IDE. Pengembangan kit ini didasarkan pada modul ESP8266, yang mengintegrasikan GPIO, PWM (*Pulse Width Modulation*), IIC, 1-Wire dan ADC (*Analog to Digital Converter*) semua dalam satu *board*. NodeMCU berukuran panjang 4.83cm, lebar 2.54cm, dan berat 7 gram. *Board* ini sudah dilengkapi dengan fitur WiFi dan *firmware*-nya yang bersifat *opensource* [6].

Tabel 0.1 Spesifikasi NodeMCU ESP8266

Mikrokontroler/Chip	Tensilica 32 bit/ ESP8266-12E
Tegangan Operasi	3.3 ~ 5 V
Tegangan Masukan	7 – 12 V
GPIO	13 pin
Kanal PWM	10 kanal
10 bit ADC Pin	1 pin
Flash Memory	4 mb
WiFi	IEEE 802.11 b/g/n
USB Port	Micro USB
USB Driver	CP212

1.4 Sensor DHT-11

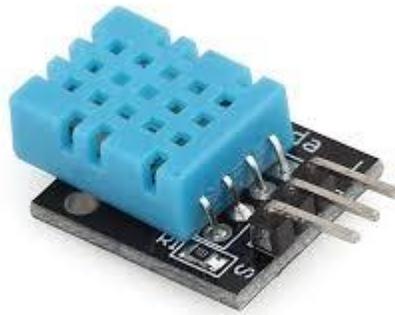
Sensor DHT-11 adalah salah satu sensor yang banyak digunakan pada proyek berbasis Arduino. Sensor ini dapat membaca parameter suhu ruangan dan kelembaban

udara. Sensor ini berbentuk kecil dan ringkas serta harganya yang terjangkau. Sensor DHT-11 merupakan serangkaian komponen sensor dan IC kontroler yang dikemas dalam satu paket. Sensor ini ada yang memiliki 4 pin ada pula yang 3 pin. Tapi tidak menjadi masalah karena dalam penerapannya tidak ada perbedaan.

Di dalam bodi sensor yang berwarna biru atau putih terdapat sebuah resistor dengan tipe NTC (*Negative Temperature Coefficient*). Resistor jenis ini memiliki karakteristik di mana nilai resistansinya berbanding terbalik dengan kenaikan suhu. Artinya, semakin tinggi suhu ruangan maka nilai resistansi NTC akan semakin menurun, begitu pun sebaliknya.

Selain itu di dalamnya terdapat sensor kelembaban dengan karakteristik resistif terhadap perubahan kadar air di udara. Data dari kedua sensor ini diolah di dalam IC kontroler. IC kontroler ini akan mengeluarkan *output* data dalam bentuk *single wire bi-directional*.

Sensor ini mampu mengukur kelembaban relatif antara 20% - 90% RH dengan akurasi 5% dan mengukur suhu dalam kisaran 0 - 50 derajat Celsius dengan akurasi 2 derajat [7].



Gambar 0.1 Sensor DHT-11

1.5 Sensor LDR

Light Dependent Resistor (LDR) ialah jenis resistor yang berubah hambatannya yang dipengaruhi oleh cahaya. Nilai hambatan pada sensor LDR tergantung pada besar kecilnya cahaya yang diterima oleh sensor LDR itu sendiri. Jika cahayanya gelap maka nilai tahanannya akan semakin besar, begitu juga jika cahayanya terang maka nilai tahanannya akan semakin kecil. LDR adalah jenis resistor yang biasa digunakan sebagai detektor cahaya atau pengukur besaran konversi cahaya. LDR terdiri dari sebuah cakram semikonduktor yang mempunyai dua buah elektroda pada permukaannya. Resistansi LDR berubah seiring dengan perubahan intensitas cahaya yang mengenainya. Dalam keadaan gelap resistansi LDR sekitar $10\text{ M}\Omega$ dan dalam keadaan terang sebesar $1\text{ K}\Omega$ atau kurang. LDR terbuat dari bahan semikonduktor seperti senyawa kimia cadmium sulfide. Dengan bahan ini energi dari cahaya yang jatuh menyebabkan lebih banyak muatan yang dilepas atau arus listrik meningkat, artinya resistansi bahan telah mengalami penurunan. Seperti halnya resistor konvensional, pemasangan LDR dalam suatu rangkaian sama persis seperti pemasangan resistor biasa [8].



Gambar 0.2 Sensor LDR

1.6 Internet of Things

Internet of Things (IoT) merupakan sebuah konsep dimana objek tertentu punya kemampuan mentransfer data lewat jaringan tanpa memerlukan adanya interaksi dari manusia ke manusia ataupun dari manusia ke perangkat komputer. IoT telah berkembang dari konvergensi teknologi nirkabel, *microelectromechanical systems*

(MEMS), dan internet. “A Things” pada IoT dapat didefinisikan sebagai subjek misalkan sebuah mobil yang telah dilengkapi sensor untuk memperingatkan pengemudi ketika tekanan ban rendah. Sejauh ini IoT paling erat hubungannya dengan komunikasi *machine-to-machine* (M2M) di bidang manufaktur dan listrik, perminyakan, dan gas. Produk dibangun dengan kemampuan komunikasi M2M yang sering disebut dengan sistem cerdas atau *smart* [9].

1.7 LoRa

LoRa (*Long Range*) adalah suatu format modulasi yang unik dan mengagumkan yang dibuat oleh *Semtech*. modulasi yang dihasilkan menggunakan modulasi FM. Ini pada pemrosesan menghasilkan nilai frekuensi yang stabil. metode transmisi juga bisa menggunakan PSK (*Phase Shift Keying*), FSK (*Frequency Shift Keying*) dan lainnya. Nilai frekuensi pada LoRa bermacam-macam sesuai daerahnya, jika di Asia frekuensi yang digunakan yaitu 433 MHZ, di Eropa nilai frekuensi yang digunakan yaitu 868 MHZ, di Amerika Utara frekuensi yang digunakan yaitu 915 MHZ, dan di Indonesia frekuensi yang digunakan yaitu 920 – 923 MHz [2].

1.8 MQTT

Protokol *Message Queuing Telemetry Transport* atau yang biasa disingkat dengan MQTT adalah protokol yang berjalan pada stack TCP/IP. Jenis protokol ini memiliki model *publish/subscribe*, yang dirancang untuk perangkat yang memiliki konsumsi daya rendah dan menjalankan *bandwidth* terbatas. Model *publish/subscribe* sendiri adalah konsep dimana semua perangkat yang telah melakukan *subscribe* akan menerima data tertentu dari *publisher*. Berikut adalah konsep sederhana dari model *publish/subscribe*. MQTT merupakan protokol komunikasi yang sangat sederhana dan ringan dengan metode *machine to machine* (M2M). Perangkat ini pertama kali dikembangkan tahun 1999 untuk melakukan pemantauan terhadap pipa minyak dalam skala lokasi yang jauh. Dengan konsep kerja yang ringan, protokol ini sangat cocok digunakan untuk mengirimkan data sensor dalam pengembangan Internet of Things [3].

1.9 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan visual yang digunakan dalam menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak dalam sebuah sistem perangkat lunak yang akan dibangun [10]. UML digunakan dalam untuk memahami sebuah rancangan, konfigurasi dan memelihara kontrol dalam sistem informasi. UML menangkap informasi tentang struktur statis dan perilaku dinamis untuk suatu sistem. Suatu sistem yang dimodelkan sebagai kumpulan objek diskrit yang berinteraksi untuk melakukan pekerjaan yang pada akhirnya menguntungkan pengguna. UML bukan merupakan sebuah bahasa pemrograman namun merupakan alat yang menyediakan sebuah generator kode dari UML ke berbagai bahasa pemrograman. Pada dasarnya, setiap orang memiliki pendapat masing-masing mengenai UML. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancang bangun perangkat lunak efektif.

1.9.1 Use Case Diagram

Use Case merupakan salah satu bagian dalam UML yang menggambarkan *external view* dari sistem yang akan kita buat modelnya. Pooley mengatakan bahwa 23 model *use case* dapat dijabarkan dalam diagram *use case*, tetapi perlu diingat, diagram tidak identik dengan model karena model lebih luas dari diagram. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai serta dapat digunakan untuk membentuk perilaku sistem yang akan dibuat.

1.9.2 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal*

processing). Oleh karena itu *activity diagram* tidak menggambarkan *behavior internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari *level* atas secara umum. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada *business modeling* untuk memperlihatkan urutan aktivitas proses bisnis. 24 Struktur diagram ini mirip *flowchart* atau *Data Flow Diagram* pada perancangan terstruktur. Sangat bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. *Activity diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram*.

1.9.3 Class Diagram

Pemodelan berbasis kelas pada dasarnya memperlihatkan objek-objek yang akan dimanipulasi oleh sistem/perangkat lunak, memperlihatkan operasi-operasi (juga dinamakan metode-metode atau layanan-layanan) yang akan diterapkan pada objek-objek untuk menghasilkan imbas tertentu pada manipulasi objek, memperlihatkan relasi-relasi antar objek (yang beberapa di antaranya bersifat 25 bersifat hierarkis), serta memperlihatkan kolaborasi-kolaborasi yang terjadi di antara kelas-kelas yang didefinisikan. Elemen-elemen model berbasis kelas mencakup di dalamnya elemen-elemen kelas-kelas dan objek-objek, atribut-atribut, operasi-operasi, model tanggung jawab kelas (*Class Responsibility Collaborator (CRC)*), diagram-diagram kolaborasi, dan paket-paket (*packages*).

1.9.4 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara

internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline vertikal*. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya.