

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Sistem Monitoring**

Sistem monitoring ialah proses mengawasi, meneliti, mengatur, dan meningkatkan setiap kegiatan yang dimaksudkan untuk diketahui oleh publik. Tujuannya adalah untuk memperoleh informasi yang akurat dan terkini mengenai kondisi yang dipantau sehingga tindakan dapat diambil dengan cepat jika terjadi perubahan atau ketidaknormalan. Secara umum, sistem monitoring dilakukan untuk mengumpulkan informasi atau pendapat untuk mendapatkan masukan untuk kebutuhan tertentu.

Pada sistem monitoring terdapat tiga kegiatan utama yang terdiri dari pengumpulan data monitoring, prosedur untuk menganalisa data pemantauan, dan prosedur untuk menyajikan hasil data. Monitoring manual memiliki beberapa kelemahan, seperti data hasil monitoring yang tidak akurat, memakan waktu lama karena harus mengambil sampel untuk diamati di laboratorium atau menggunakan peralatan sensor, dan membutuhkan proses yang lebih banyak dalam pengumpulan data hasil monitoring yang dibutuhkan (Ariyani, 2022).

#### **2.2. Internet of Things (IoT)**

Istilah *Internet of Things* (IoT) muncul dan diperkenalkan pada sebuah presentasi tahun 1999 oleh Kevin Ashton. Ide di balik *Internet of Things*, adalah untuk memperbanyak keuntungan memiliki akses internet. Keuntungan dari akses internet yang konstan yang memungkinkan sensor dan aktuator jaringan untuk

terhubung ke mesin, peralatan, dan objek fisik lainnya untuk mengumpulkan data serta mengawasi operasi mereka. Melalui informasi yang telah didapatkan mesin dapat bekerja sama serta mengambil tindakan dengan sendirinya (Saputra & Siswanto, 2020).

Setiap objek di *Internet of Things* (IoT) harus memiliki alamat Protokol *Internet* (IP). Sebuah identifikasi pada jaringan yang memungkinkan sebuah benda dikendalikan oleh benda lain pada jaringan yang sama yang dapat menerima perintah dari benda-benda berjaringan lainnya. Selain itu, alamat Protokol *Internet* (IP) memungkinkan benda-benda ini untuk terhubung ke jaringan *internet* (Wijaya, & Sukarni, 2019). *Internet* digunakan dalam proses pemantauan *Internet of Things* untuk memungkinkan pengamatan secara terus menerus dan real-time. Oleh karena itu, *Internet of Things* ialah teknologi terbaru yang dapat dimanfaatkan sebagai solusi pemantauan kondisi secara real-time. Monitoring yang sebelumnya dilakukan secara manual dapat menjadi monitoring digital dengan penggunaan *Internet of Things* (IoT) (Ariyani, 2022).

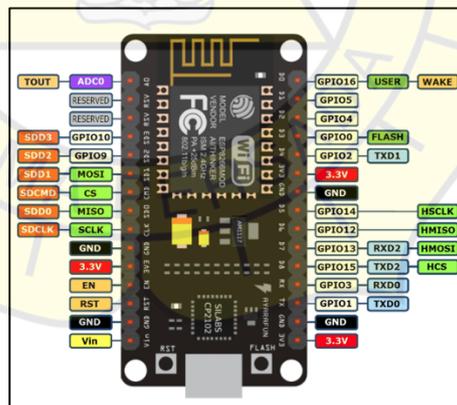
### **2.3. NodeMCU**

NodeMCU adalah alat pengontrol dengan ukuran kecil yang sering disebut sebagai mikrokontroler. NodeMCU bersifat *open source* berbasis *System on Chip* ESP8266. Meskipun NodeMCU dapat diprogram menggunakan Arduino IDE, NodeMCU didasarkan pada bahasa pemrograman Lua (Efendi, 2019). Karena kemampuan pemrosesan dan penyimpanan on-board ESP8266 yang kuat, ESP8266 dapat dengan mudah dikembangkan dan dimuat dengan cepat melalui integrasi

*General Purpose Input/Output* (GPIO) dengan sensor dan aplikasi perangkat khusus lainnya serta waktu *loading* yang diperlukan minimal (Hamrul & Mansyur, 2021). NodeMCU ESP8266 dapat dilihat pada Gambar 2.1 dan susunan kaki *board* NodeMCU ESP8266 dapat dilihat pada Gambar 2.2.



Gambar 2.1 NodeMCU ESP8266 (Efendi, 2019)



Gambar 2.2 Susunan Kaki Kaki NodeMCU ESP8266 (Efendi, 2019)

#### 2.4. Sensor pH

Sensor pH meter ialah sebuah jenis alat ukur yang berguna dalam pengukuran derajat keasaman atau kebasaan suatu cairan. Terdapat elektroda

khusus dalam alat tersebut yang berguna sebagai alat ukur pH semua bahan semi padat, elektroda (probe pengukur) yang sudah terhubung dengan sebuah alat elektronik yang mengukur dan menampilkan nilai pH (Nurwachid, 2023). Sensor pH dapat dilihat pada Gambar 2.3



Gambar 2.3 Sensor pH (Nurwachid, 2023)

## 2.5. Sensor *Turbidity*/Kekeruhan

Kekeruhan merupakan suatu kondisi dimana transparansi cairan menjadi berkurang karena adanya komponen lain. Cairan tampak keruh atau tidak jernih terjadi karena bahan kimia yang terlarut di dalamnya (Hamrul & Mansyur, 2021). Sesuai Peraturan Menteri Kesehatan RI Nomor 32 Tahun 2017, tingkat kekeruhan maksimum yang diperbolehkan adalah 25 NTU (*Nephelometric Turbidity Unit*). Sensor *turbidity* adalah sensor yang dimanfaatkan pada penentuan kekeruhan air dengan cara membandingkan cahaya yang akan dipantulkan dengan cahaya yang datang dan membaca karakteristik optik air yang disebabkan oleh cahaya. Dalam kasus sensor kekeruhan, tegangan keluaran sensor berubah seiring dengan

meningkatnya kekeruhan air. (Noor, et al., 2019). Sensor *turbidity* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Sensor *Turbidity*

## 2.6. Sensor Suhu DS18B20

Sensor DS18B20 adalah sensor suhu dengan output digital. Dengan akurasi 0,5 °C dalam kisaran suhu -10 °C hingga +85 °C, sensor ini memiliki tingkat presisi yang cukup tinggi (Noor, et al., 2019). Sensor ini hanya memerlukan sedikit kabel untuk instalasi karena memiliki antarmuka satu kabel. Khususnya, sensor ini dapat dihubungkan secara paralel ke satu input. Hal ini menunjukkan bahwa meskipun output sensor hanya terpasang pada satu Pin Arduino, Anda dapat menggunakan beberapa sensor DS18B20. Penggunaan sensor ini secara luas dapat dikaitkan dengan desainnya yang tahan air (Nurwachid, 2023).

Sensor DS18B20 beroperasi atas dasar bahwa tegangan listrik akan dihasilkan apabila probe mendeteksi perbedaan panas dalam suatu gradien. Temperatur yang meningkat menyebabkan ion bergerak lebih cepat, yang

menyebabkan peningkatan konduktivitas listrik juga akan meningkat (Ariyani, 2022). Gambar sensor suhu DS18B20 dapat dilihat pada Gambar 2.5.



Gambar 2.5 Sensor Suhu DS18B20 (Nurwachid, 2023)

## 2.7. Buzzer

Buzzer yakni sebuah perangkat elektronik yang dapat mengonversi getaran listrik menjadi getaran suara. Perangkat ini berfungsi dengan cara yang sama seperti pengeras suara. Buzzer sering digunakan untuk memberi tanda ketika suatu proses selesai atau ketika terjadi kesalahan alat pada suatu perangkat (Fani, et al., 2020). Ada dua jenis buzzer yaitu buzzer pasif dan buzzer aktif. Ketika buzzer aktif langsung dinyalakan, buzzer akan langsung berbunyi karena merupakan jenis buzzer yang mandiri. Sedangkan, buzzer pasif dapat digunakan dalam nada tinggi dan rendah dan memiliki volume rendah (Febrianti, et al., 2021). Buzzer dapat dilihat Gambar 2.6.



Gambar 2.6 Buzzer (Fani, et al., 2020)

## 2.8. ADS1115 Module

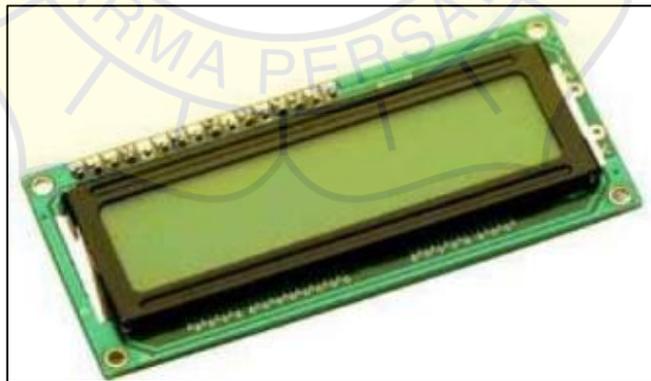
ADS1115 Module adalah modul konverter sinyal yang mengubah ADC (*Analog to Digital Converter*) dan sesuai untuk aplikasi berbasis mikroprosesor yang membutuhkan ADC resolusi tinggi dengan resolusi 16 bit (Setiowati, et al., 2022). Input analog yang masuk melewati rangkaian multiplexer dalam operasi ADS1115 untuk memilih data sebelum dikirim ke blok PGA (*Programmable Gain Amplifier*). Sinyal pertama-tama akan diperkuat di blok PGA sebelum dikirim ke sirkuit ADC untuk konversi. Data akan diterjemahkan dan kemudian ditransmisikan menggunakan komunikasi serial I2C, yaitu pin SCL dan SDA (Hasanah, et al., 2022). ADS1115 Module dapat dilihat pada Gambar 2.7.



Gambar 2.7 ADS1115 Module (Setiowati, et al., 2022).

## 2.9. Liquid Crystal Display (LCD)

*Liquid crystal display* (LCD) yakni salah satu jenis layar elektronik yang diproduksi dengan teknologi logika CMOS. LCD dapat digunakan dalam berbagai cara ketika membuat sistem yang menggunakan mikrokontroler. LCD dapat digunakan dalam aplikasi mikrokontroler untuk menampilkan menu, teks, atau nilai sensor (Simbar, et al., 2017). LCD dapat dilihat pada Gambar 2.8.



Gambar 2.8 *Liquid Crystal Display* (LCD) (Simbar, et al., 2017).

## 2.10. Arduino IDE

Dari segi bahasa, lingkungan yang terintegrasi untuk pengembangan dikenal sebagai *Integrated Development Environment* (IDE). Disebut sebagai lingkungan karena pemrograman Arduino dilakukan dengan menggunakan perangkat lunak ini untuk menjalankan operasi sintaks pemrograman tertanam. Bahasa pemrograman yang digunakan oleh Arduino IDE mirip dengan bahasa pemrograman C++ [6]. Tampilan Arduino IDE dapat dilihat pada Gambar 2.9.



Gambar 2.9 Arduino IDE

Sumber: <https://docs.arduino.cc/software/ide-v1/tutorials/Environment/>

## 2.11. Kualitas Air

Menurut Peraturan Menteri Kesehatan Republik Indonesia Nomor 32 Tahun 2017, kualitas air dapat diketahui melalui pengecekan Standar Baku Mutu Kesehatan Lingkungan untuk media Air Untuk Keperluan Higiene Sanitasi meliputi parameter fisik, kimia, dan biologi. Parameter wajib adalah parameter yang perlu

diperiksa secara teratur sesuai dengan peraturan perundangan dan pedoman tambahan, sedangkan parameter tambahan hanya perlu diperiksa jika kondisi geohidrologis menunjukkan adanya kemungkinan risiko pencemaran yang terkait dengannya. Untuk parameter kimia dapat dilihat tabel 2.1, untuk parameter biologi dapat dilihat pada tabel 2.2, dan untuk parameter fisik dapat dilihat pada tabel 2.3.

a. Parameter Kimia

Tabel 2.1. Parameter Kimia

No.	Parameter	Unit	Standar Baku Mutu (kadar maksimum)
<b>Wajib</b>			
1.	pH	mg/l	6,5 - 8,5
2.	Besi	mg/l	1
3.	Fluorida	mg/l	1,5
4.	Kesadahan (CaCO <sub>3</sub> )	mg/l	500
5.	Mangan	mg/l	0,5
6.	Nitrat, sebagai N	mg/l	10
7.	Nitrit, sebagai N	mg/l	1
8.	Sianida	mg/l	0,1
9.	Deterjen	mg/l	0,05
10.	Pestisida total	mg/l	0,1
<b>Tambahan</b>			
1.	Air raksa	mg/l	0,001
2.	Arsen	mg/l	0,05
3.	Kadmium	mg/l	0,005
4.	Kromium (valensi 6)	mg/l	0,05
5.	Selenium	mg/l	0,01
6.	Seng	mg/l	15
7.	Sulfat	mg/l	400
8.	Timbal	mg/l	0,05

b. Parameter Biologi

Tabel 2.2. Parameter Biologi

No.	Parameter Wajib	Unit	Standar Baku Mutu (kadar maksimum)
1.	Total coliform	CFU/100ml	50
2.	E. coli	CFU/100ml	0

c. Parameter Fisik

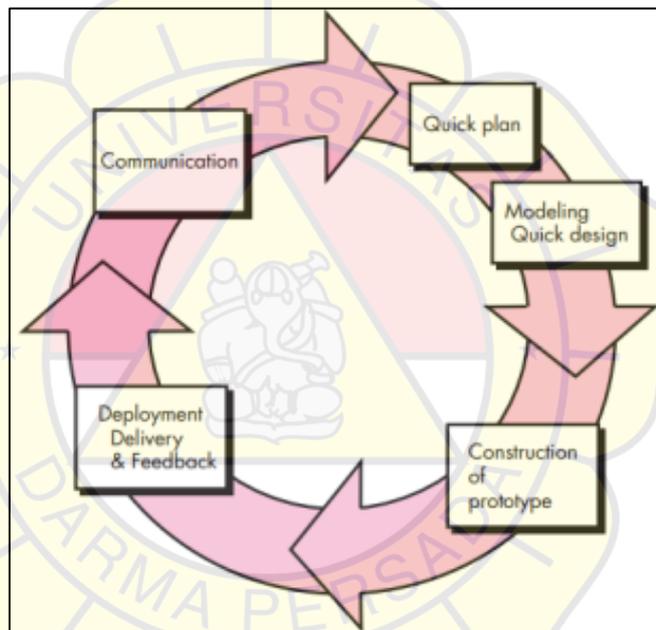
Tabel 2.3. Parameter Fisik

No.	Parameter Wajib	Unit	Standar Baku Mutu (kadar maksimum)
1.	Kekeruhan	NTU	25
2.	Warna	TCU	50
3.	Zat padat terlarut ( <i>Total Dissolved Solid</i> )	mg/l	1000
4.	Suhu	°C	suhu udara ± 3
5.	Rasa		tidak berasa
6.	Bau		tidak berbau

## 2.12. Metode Prototyping

Metode *Prototyping* adalah teknik yang digunakan oleh perancang perangkat lunak untuk menunjukkan bagaimana program atau komponen perangkat lunak akan berfungsi dalam pengaturan dunia nyata sebelum tahap pembangunan lingkungan dimulai. Metode ini berfungsi sebagai indikator gambaran yang akan dihasilkan di masa depan dan membuat perbedaan antara peran demonstrasi dan eksplorasi (Siswidiyanto, et al., 2020).

Kelebihan dari metode *prototyping* adalah mampu mengakomodasi (menyediakan sesuatu untuk memenuhi kebutuhan) perangkat lunak dengan spesifikasi kebutuhan yang tidak jelas, dapat dipraktikkan dalam model perangkat lunak apa pun, lebih banyak waktu yang dihemat selama pengembangan, dan pengguna secara aktif berpartisipasi dalam pengembangan sistem (Supandi & Sudir, 2019). Tahapan metode *prototyping* dapat dilihat pada Gambar 2.10.



Gambar 2.10 Tahapan Metode *Prototyping* (Ichwani, et al., 2021).

a. *Communication*

Tujuan dari tahap ini adalah mengidentifikasi berbagai kebutuhan aplikasi yang akan dirancang nantinya. Tahap ini melibatkan klien yang bersangkutan untuk memastikan bahwa proses desain menghasilkan hasil yang diinginkan (Ichwani, et al., 2021).

b. *Quick Plan*

Pada tahap ini perancang perangkat lunak akan dengan cepat merencanakan sesuai dengan spesifikasi kebutuhan pengguna berdasarkan data yang dikumpulkan pada tahap sebelumnya yang mendukung kebutuhan selama prosedur ini dengan membuat desain antarmuka yang diperlukan dan kebutuhan pendukung lainnya (Fani, et al., 2021).

c. *Modeling Quick Design*

Pada tahap ini, tim perancang sekarang akan mengembangkan model desain *Unified Modelling Language* (UML) atau pemodelan lain yang diperlukan dengan durasi desain yang efektif untuk menggambarkan kebutuhan pelanggan berdasarkan analisis sebelumnya (Ichwani, et al., 2021).

d. *Construction of Prototype*

Pada tahap ini, melalui data yang telah diperoleh pada tahap sebelumnya desainer akan memulai proses pengembangan perangkat lunak. Agar perancang bisa segera mendapatkan masukan dari klien mengenai perangkat lunak yang dihasilkan, pendekatan pengembangan ini lebih berkonsentrasi pada fitur-fitur utama perangkat lunak (Ichwani, et al., 2021).

e. *Deployment Delivery and Feedback*

Selama fase ini, prototipe akan diserahkan kepada klien untuk mendapatkan masukan. Masukan dari pelanggan akan digunakan untuk menginformasikan iterasi prototipe di masa depan yang akan lebih memenuhi kebutuhan klien (Ichwani, et al., 2021).

### **2.13. BlackBox Testing**

*BlackBox Testing* ialah suatu metode pengujian terhadap perangkat lunak dengan menguji fungsionalitas aplikasi tersebut, bukan operasi atau struktur internalnya. Biasanya tidak perlu memiliki pemahaman tertentu mengenai kode aplikasi atau struktur internal atau pengetahuan pemrograman. Meskipun biasanya bersifat fungsional, pengujian ini juga dapat bersifat non-fungsional (Rahayu & Sahlinal, 2017).

Kelebihan dari *BlackBox Testing* ialah proses uji perangkat dilaksanakan melalui sudut pandang pengguna, kemudian pengujian ini dapat digunakan untuk mengentahui serta mengungkap ambiguitas serta ketidakkonsistensian pada spesifikasi persyaratan. Penguji dan pemrogram saling bergantung dan penguji tidak perlu mengetahui bahasa pemrograman tertentu (Jaya, 2018).

### **2.14. Unified Modelling Language (UML)**

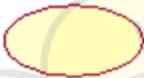
*Unified Modelling Language (UML)* ialah sebuah standar bahasa yang telah dimanfaatkan dalam industri dalam mendeskripsikan persyaratan, melakukan analisis dan desain, serta mendeskripsikan arsitektur dalam pemrograman berorientasi objek (Siswidiyanto, et al., 2020). Beberapa jenis UML beserta penjelasannya adalah sebagai berikut

#### **a. Use Case Diagram**

*Use case diagram* adalah pemodelan untuk perilaku (*behavior*) sistem informasi yang akan dibangun. Diagram ini menggambarkan interaksi yang

terjadi antara satu atau lebih subjek dengan sistem informasi yang baru dikembangkan (Siswidiyanto, et al., 2020).

Tabel 2.4. *Use Case Diagram*

No	Simbol	Keterangan
1		Aktor: Menunjukkan user yang akan menggunakan sistem baru.
2		<i>Use Case</i> : Menunjukkan proses yang terjadi pada sistem baru
3		<i>Association</i> : Menunjukkan hubungan antara aktor dengan <i>use case</i> atau antar <i>use case</i> .

b. *Activity Diagram*

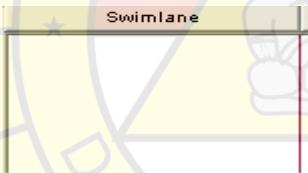
*Activity diagram* yakni sebuah representasi grafis dari alur kerja (*workflow*) sistem dari sebuah proses bisnis atau menu perangkat lunak (Siswidiyanto, et al., 2020).

c. *Sequence Diagram*

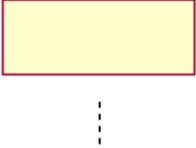
*Sequence diagram* menggunakan informasi tentang kehidupan objek dan pesan yang dikirim antar objek untuk menggambarkan bagaimana objek berperilaku dalam sebuah *use case*. Deskripsi *sequence diagram* dibuat dalam jumlah yang sama dengan definisi *use case*, yang memiliki prosedurnya sendiri dan *sequence diagram* mencakup interaksi jalur pesan di semua *use*

case yang didefinisikan. Semakin banyak *use case* yang didefinisikan, maka banyak pula *sequence diagram* yang perlu dibuat (Putra & Andriani, 2019).

Tabel 2.5. *Activity Diagram*

No	Simbol	Keterangan
1		Kondisi Awal: Menunjukkan awal dari suatu diagram aktivitas.
2		Kondisi Akhir: Menunjukkan akhir dari suatu diagram aktivitas.
3		Kondisi Transisi: Menunjukkan kondisi transisi dari suatu diagram aktivitas.
4		Swimlane: Menunjukkan aktor dari diagram aktivitas yang dibuat.
5		Aktivitas: Menunjukkan aktivitas-aktivitas yang terdapat pada diagram aktivitas.
6		Pengecekan Kondisi: Menunjukkan pengecekan terhadap suatu kondisi.

Tabel 2.6. *Sequence Diagram*

No	Simbol	Keterangan
1		<p>Objek:</p> <p>Menunjukkan objek yang terdapat di <i>sequence diagram</i>.</p>
2		<p>Pesan ke Objek Sendiri:</p> <p>Menunjukkan pesan yang diproses pada objek itu sendiri.</p>
3		<p>Pesan Objek:</p> <p>Menunjukkan pesan yang disampaikan ke objek lain dalam <i>sequence diagram</i>.</p>

