

## BAB 3

### METODE PENELITIAN

#### 3.1. ALAT DAN BAHAN

Dalam kajian ini, akan diuraikan bagaimana penggunaan alat dan bahan yang diperlukan untuk merancang prototipe sistem pemantau dan pengendali volume air pada tangki berbasis *Internet of Things*. Daftar alat dan bahan yang di gunakan dapat di lihat pada tabel 3.1.

**Tabel 3.1 Daftar Alat dan Bahan**

No.	Alat dan Bahan	Jumlah
1.	<i>Smartphone</i>	1
2.	Laptop	1
3.	Galon Air	1
4.	<i>Non-Contact Liquid Level Sensor XKC-Y25VXKC-Y25-V</i>	1
5.	<i>NodeMCU ESP8266</i>	1
6.	Pompa Air	1
7.	<i>Ubidots</i>	1
8.	Adapter	1

##### 3.1.1. *Smartphone*

*Smartphone* merupakan perangkat elektronik yang akan digunakan bertugas untuk menerima *input* data, melakukan pengolahan data, dan memberikan *output* informasi. Pada penelitian ini, *Smartphone* diaplikasikan untuk memproses semua informasi yang dikumpulkan oleh peneliti, serta untuk membuat aplikasi atau program. Selain itu, *Smartphone*, khususnya Redmi Note 11 Pro dengan spesifikasi memiliki ram 8 gb dan penyimpanan internal sebesar 128 gb yang akan digunakan sebagai media untuk mengakses dan mengambil hasil data dari sistem yang dikembangkan dalam penelitian ini.

##### 3.1.2. Laptop

Laptop, seperti *Smartphone*, merupakan perangkat elektronik yang digunakan memiliki fungsi untuk menerima *input* data, melakukan pengolahan data, dan memberikan *output* informasi. Pada penelitian ini,

laptop juga akan digunakan untuk mengelola data yang terkumpul, serta sebagai alat untuk membuat aplikasi atau program yang diperlukan dalam penelitian. Laptop yang digunakan adalah dari merek HP EliteBook 840 G3 dengan spesifikasi Processor: Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz (4 CPUs), ~2.5GHz dan Memory: 8192MB RAM.

### **3.1.3. Galon Air**

Galon air pada penelitian ini digunakan sebagai wadah atau penampungan air yang nantinya akan diteliti. Galon air yang digunakan adalah galon air bekas air mineral merk Aqua atau sejenisnya yang berukuran 16,5 Liter.

### **3.1.4. *Non-Contact Liquid Level Sensor* XKC-Y25-V**

*Non-Contact Liquid Level Sensor* XKC-Y25V adalah teknologi yang dapat memproses sinyal standar tinggi dengan memanfaatkan *chip* canggih dan kapasitas operasi dengan kecepatan tinggi pada saat pengecekan level cairan tanpa perlu kontak langsung. Jika sensor level cairan mendeteksi level cairan, itu akan mengeluarkan *output* TINGGI dan menyalakan LED. Jika tidak ada cairan yang terdeteksi, *outputnya* RENDAH dan mematikan LED. Pada penelitian ini, *Non-Contact Liquid Level Sensor* XKC-Y25V berfungsi untuk memantau atau menetapkan level cairan pada wadah. Dalam kajian ini wadah yang dimaksud adalah galon air.

### **3.1.5. *NodeMCU* ESP 8266**

Dalam penelitian ini, *NodeMCU* ESP 8266 berperan sebagai *platform* IoT dengan standar *open source*. *NodeMCU* ESP 8266 bertindak sebagai mikro pengendali yang berfungsi menjadi antarmuka *relay* dan penerimaan data dari sensor untuk dijadikan *input*. *NodeMCU* ESP 8266 bertindak menjadi pengelola utama dan juga menjadi sarana untuk mengirimkan hasil data ke website yang dituju. Proses transfer data dilakukan melalui koneksi *Wifi* yang terintegrasi di *NodeMCU* ESP 8266.

### **3.1.6. Pompa Air**

Pompa air pada penelitian ini berfungsi untuk memindahkan air ke wadah yang lain. Wadah yang dimaksud adalah ember kecil ke galon air yang sudah dijelaskan sebelumnya. Merk pompa air yang di gunakan adalah dari produk MaYun dengan spesifikasi pompa mampu menyedot debit air: 2-3 L/Menit, *Voltase*: 5 V, material: plastik dan komponen, dimensi: panjang pompa: 44 mm\r\nDiameter Pompa: 24 mm\r\nDiameter.

### **3.1.7. Ubidots**

*Ubidots* adalah platform cloud berguna untuk memonitor dan mengontrol perangkat secara jarak jauh melalui antarmuka pengguna yang ramah dan intuitif. Dengan menggunakan *Ubidots*, seseorang bisa tersambung dengan ponsel, komputer, dan sistem mikrokontroler untuk memantau data sensor, mengontrol perangkat, serta menganalisis data secara *real-time*. Pada penelitian ini *Ubidots* berfungsi untukmenautkan berbagai perangkat yang dibutuhkan pada penelitian ini,seperti yang telah dijabarkan sebelumnya.

### **3.1.8. Adapter**

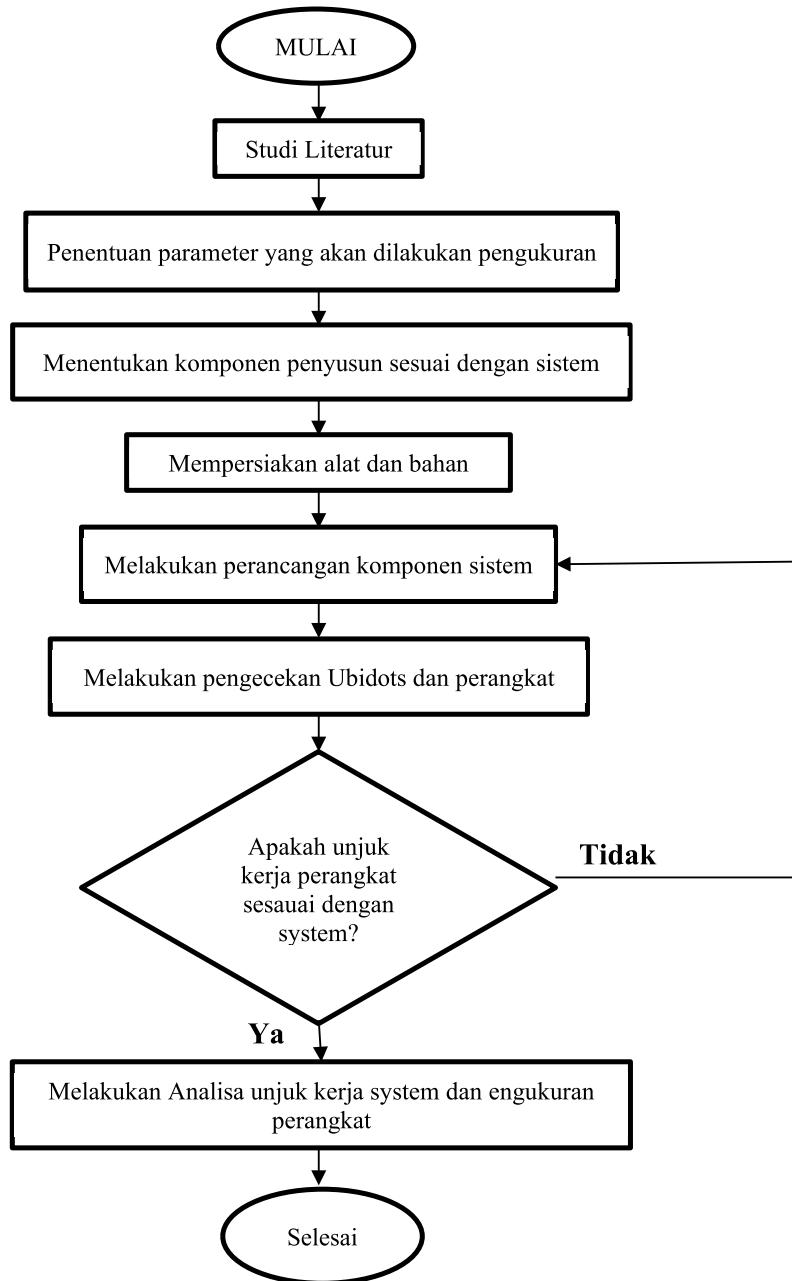
Adapter pada penelitian ini berfungsi Untuk mengonversi tingkat aliran listrik yang bertipe bolak-balik (AC) dengan nilai yang tinggi menjadi tegangan listrik tipe arus searah (DC), Anda dapat menggunakan alat yang disebut penyearah atau *diode bridge*. Penyearah akan melakukan perubahan terhadap arus bolak-balik ke searah dengan mengonversi siklus positif dan negatif dari tegangan AC menjadi arus searah. Ini sering digunakan dalam berbagai aplikasi, termasuk di dalam sirkuit catu daya dengan nilai yang rendah.

## **3.2. ALUR PENELITIAN**

### **3.2.1. Flowchart**

Penelitian ini disusun melalui serangkaian langkah, dimulai dari telah literatur, melalui tahap perancangan dan implementasi, hingga

mencapai puncaknya pada evaluasi kinerja sistem dan analisis data simulasi. Penelitian ini dilakukan menggunakan metode *experimental* yang dilakukan bersamaan dengan pencatatan hasil data. Penelitian ini melakukan pengujian fungsionalitas dari *Non-Contact Liquid Level Sensor* XKC-Y25V dari sistem pemantau dan pengendali volume air. *Flowchart* penelitian terdapat pada gambar 3.1.



**Gambar 3.1** *Flowchart* Penelitian

Berikut ini merupakan penjelasan untuk masing-masing langkah dalam kajian ini:

### 1. Studi Literatur

Hal yang pertama kali harus dilakukan yakni studi literatur. Studi literatur bertujuan guna memperdalam penelitian yang akan dilakukan. Studi literatur yang dimaksud adalah seperti referensi jurnal, buku, paper, dan juga skripsi. Pada tahap ini, peneliti akan menghimpun beragam teori yang relevan dengan subjek penelitian yang sedang diselidiki.

### 2. Penentuan Parameter yang Akan dilakukan Pengukuran

Setelah melakukan studi literatur, tahap berikutnya adalah menentukan parameter yang akan dilakukan pengukuran terhadap tingkat cairan yang merupakan parameter penting dalam berbagai aplikasi, dan pengukuran non-kontak seringkali lebih diinginkan untuk menghindari kontaminasi atau kerusakan pada cairan yang diukur. Salah satu metode non-kontak yang umum digunakan adalah menggunakan sensor ultrasonik. Sensor ini menggunakan gelombang ultrasonik untuk mengukur jarak antara sensor dan permukaan cairan, dan dari situ tingkat cairan dapat dihitung. Metode lain termasuk penggunaan sensor tekanan, sensor optik, atau sensor kapasitif. Pemilihan metode tergantung pada aplikasi, sifat cairan yang diukur, dan kebutuhan akurasi tingkat cairan menggunakan *Non-Contact Liquid Level Sensor XKC-Y25V*.

### 3. Menentukan Komponen Penyusun Sesuai dengan Sistem

Langkah selanjutnya adalah masuk ke tahap ketiga, yang melibatkan seleksi komponen sistem. Dalam sistem pemantau dan pengendali volume air, komponen penyusunnya mencakup perangkat pemantauan dan *Dashboard Realtime Ubidots*. Perangkat pemantauan dengan *NodeMCU ESP8266* sebagai mikrokontroler utama, *relay*, dan sensor level air *Non-Contact Liquid Level Sensor XKC-Y25V*. Sementara itu, *Dashboard Realtime Ubidots* berfungsi sebagai basis data yang terhubung dengan perangkat dan *Dashboard Ubidots*.

#### 4. Mempersiapkan Alat dan Bahan

Selanjutnya adalah menentukan alat serta bahan yang diperlukan untuk menunjang keberhasilan sistem pemantau dan pengendali volume air. Sistem ini menggunakan sensor *Non-Contact Liquid Level Sensor XKC-Y25V*, *NodeMCU ESP 8266*, *relay* dan pompa air.

#### 5. Melakukan Perancangan Komponen Sistem

Di tahapan kelima ini ialah membuat rancangan alat untuk pembuatan sistem. Pada tahap ini yang dilakukan adalah pembuatan *Dashboard* pada *Ubidots* dan mempersiapkan program untuk memprogram *NodeMCU ESP 8266* melalui *software* Arduino IDE agar bisa terintegrasi dengan *ubidots* melalui jaringan internet.

#### 6. Melakukan Pengecekan *Ubidots* dan Perangkat

Langkah berikutnya adalah tahap keenam, memastikan peran dari fitur-fitur pemantau dengan melakukan serangkaian pengujian.

Pengujian ini dapat mencakup:

##### A. Pengujian Fungsional :

Memastikan bahwa setiap fitur berfungsi seperti yang diharapkan sesuai dengan spesifikasi yang telah ditetapkan.

##### B. Pengujian Integrasi :

Memeriksa interaksi antara berbagai fitur dalam sistem untuk memastikan bahwa alat beroperasi dengan baik bersama.

##### C. Pengujian Kinerja :

Mengukur kinerja sistem dalam hal waktu respons, kecepatan, dan penggunaan sumber daya seperti memori dan CPU.

##### D. Pengujian Stres :

Menguji batas-batas sistem dengan memperoleh beban yang lebih besar dari yang diharapkan untuk melihat bagaimana sistem menanggapi.

E. Pengujian Kesalahan :

Mengidentifikasi dan memperbaiki kesalahan dalam sistem dengan melakukan pengujian terhadap skenario yang mungkin terjadi.

F. Pengujian Keamanan :

Memeriksa keamanan sistem untuk melindungi data dan mencegah akses yang tidak sah.

Dengan melakukan serangkaian pengujian ini, penulis dapat memastikan bahwa sistem pemantau berfungsi dengan baik dan dapat diandalkan dalam penggunaan sehari-hari. Pada tahap ini, dilakukan pengujian terhadap setiap fitur sebagai upaya pemastian sistem sebagaimana yang diharapkan dan sesuai dengan keperluan yang telah ditetapkan. Dimana dengan melakukan pemantauan dari *Ubidots Dashboard* padaperangkat laptop maupun *Handphone* ataupun *device* lain yang bisa mengakses internet. Pengecekan fitur tersebut sesuai dengan metode *Ubidots Dashboard*.

7. Melakukan Analisa Unjuk Kerja Sistem dan Pengukuran Perangkat

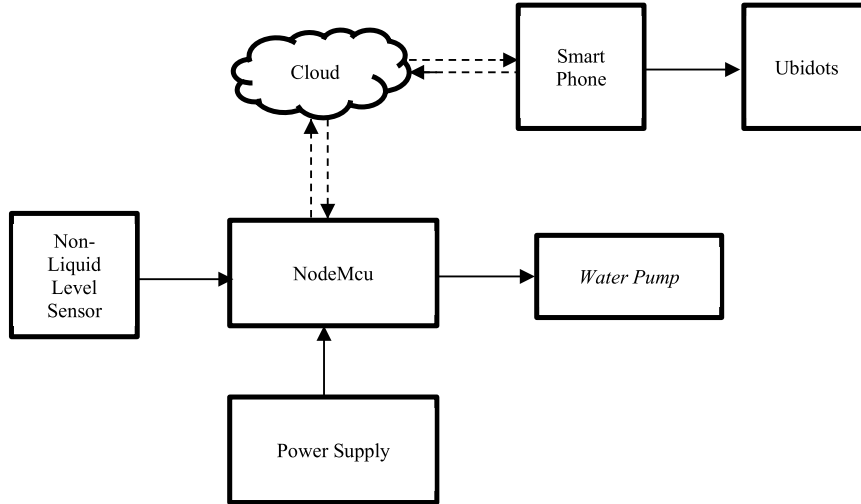
Pada tahapan ke tujuh ini, prototipe dan data yang ada diperiksa untuk memastikan fungsionalitasnya. Selanjutnya, dilakukan analisis untuk menilai apakah sistem yang dibuat telah mencapai standar ketetapan. Tahap ini memastikan bahwa prototipe berfungsi sesuai dengan yang diinginkan dan memenuhi kriteria yang telah ditetapkan sebelumnya.

8. Kesimpulan

Pada tahap kedelapan atau tahapan terakhir, akan dilakukan penarikan kesimpulan berdasarkan hasil data yang di dapatkan. Dalam kesimpulan ini, dijelaskan tentang keakuratan dan efektivitas sistem yang dikembangkan, serta *implikasi* dari temuan yang diperoleh terhadap tujuan penelitian prototipe sistem pemantau dan pengendali volume air pada tangki berbasis *Internet of Things* dan hasil evaluasi secara menyeluruh.

### 3.2.2. Blok Diagram Sistem

Blok diagram sistem menjelaskan tentang proses *input* dan *output* sistem yang di buat. Blok diagram sistem terdapat pada gambar 3.2 .



**Gambar 3.2 Diagram Sistem**

Gambar 3.2 menunjukkan blok diagram sistem yang di sajikan, *inputan* di mulai dari power supply yang memberikan daya kepada *NodeMCU* kemudian *NodeMCU* akan memproses *inputan* dari sensor *Non-Contact Liquid Level Sensor XKC-Y25V*. Ketika sensor *Non-Contact Liquid Level Sensor XKC-Y25V* yang di pasang di batas atas dan di batas bawah tidak mendeteksi adanya air maka sensor *Non-Contact Liquid Level Sensor XKC-Y25V* akan memberikan *inputan* ke *NodeMCU* untuk mengaktifkan *water pump*. *NodeMCU* juga meneruskan informasi ke *Smartphone* melalui cloud untuk menampilkan informasi kepada pengguna melalui *dashboard Ubidots*.

Untuk memastikan peran setiap fitur bekerja sesuai sistem pemantau, dapat dilakukan serangkaian pengujian. Pengujian ini dapat mencakup:

1. Pengujian Fungsional :

Memastikan bahwa setiap fitur berfungsi seperti yang diharapkan sesuai dengan spesifikasi yang telah ditetapkan.

2. Pengujian Integrasi :

Memeriksa interaksi antara berbagai fitur dalam sistem untuk memastikan



bahwa alat beroperasi dengan baik bersama.

3. Pengujian Kinerja :

Mengukur kinerja sistem dalam hal waktu respons, kecepatan, dan penggunaan sumber daya seperti memori dan CPU.

4. Pengujian Stres :

Menguji batas-batas sistem dengan memperoleh beban yang lebih besar dari yang diharapkan untuk melihat bagaimana sistem menanggapi.

5. Pengujian Kesalahan :

Mengidentifikasi dan memperbaiki kesalahan dalam sistem dengan melakukan pengujian terhadap skenario yang mungkin terjadi.

6. Pengujian Keamanan :

Memeriksa keamanan sistem untuk melindungi data dan mencegah akses yang tidak sah.

Dengan melakukan serangkaian pengujian ini, dapat memastikan bahwa sistem pemantau berfungsi dengan baik dan dapat diandalkan dalam penggunaan sehari-hari. Pada saat alat menyala karena adanya aliran listrik dan masuk pada mode inisialisasi atau yang disebut dengan persiapan sistem. Inisialisasi perangkat adalah perangkat yang sudah siap untuk melakukan sebuah persiapan sistem guna bisa digunakan oleh pengguna dengan ditandai dengan adanya tanda *Connect* pada *Ubidots*. Jika pada *Ubidots* sudah tertera tulisan *Connect* maka menandakan bahwa perangkat telah terhubung dengan sistem dan internet sehingga sudah siap untuk digunakan. Langkah yang dilakukan untuk pemantauan parameter adalah dimana apabila kondisi air kurang dari setengah/*middle* atau dengan kata lain *empty* atau kosong maka sensor *Non-Contact Liquid Level Sensor XKC-Y25V* secara otomatis akan mendeteksi untuk mengisi air dan pompa otomatis akan menyala untuk mengisi air. Selanjutnya apabila air lebih dari setengah/*middle* atau dalam kondisi *full*, maka sensor *Non-Contact Liquid Level Sensor XKC-Y25V* akan mendeteksi keberadaan air dalam kondisi *full* sehingga secara otomatis pompa akan mati dan berhenti mengisi air. Tahap terakhir setelah berhasil mendeteksi dua kondisi situasi diatas, maka kondisi selesai.

### 3.3. TAHAP PERANCANGAN ALAT

Perencanaan dan pembuatan alat terbagi dalam dua tahapan sensor, Proses pemrograman yang dilakukan tidak jauh berbeda dengan proses pemrograman pada umumnya, dimana dalam proses ini sebagai berikut:

#### 3.3.1. *NodeMCU* ESP8266

*NodeMCU* ESP8266 ialah sebuah *board* elektronik yang berbasis *chip* ESP8266 yang berfungsi menjadi mikrokontroler dan memiliki koneksi internet (WiFi), *NodeMCU* ESP8266 dilengkapi dengan pin I/O dan bisa digunakan untuk aplikasi monitoring dan pengendalian perangkat IoT. Dalam penelitian ini, *NodeMCU* ESP8266 digunakan sebagai antarmuka *relay* dan menerima data dari sensor-sensor sebagai *input*. Prototipe sistem pemantauan ketinggian level air sungai secara jarak jauh berbasis IoT dengan *NodeMCU* 1.0 membutuhkan proses uji coba secara menyeluruh terhadap perangkat hardware dan *software* dengan metode pengujian yaitu Black Box Testing, yang menganalisis detail desain tanpa memperhatikan struktur kontrol dengan program yang terstruktur. Proses uji ini bertujuan memahami fungsionalitas *software* secara internal dan memastikan operasi internal berdasarkan ketentuan yang telah ditetapkan.

#### 3.3.2. *Non-Contact Liquid Level Sensor* XKC-Y25-V

*Non-Contact Liquid Level Sensor* XKC-Y25V adalah sensor dengan teknologi yang memproses sinyal tingkat tinggi yang memanfaatkan *chip* canggih dan kapasitas operasi dengan kecepatan tinggi. Sensor *Non-Contact Liquid Level Sensor* XKC-Y25V ini dirancang untuk mendeteksi level cairan tanpa kontak fisik dengan media tersebut. Jika sensor *Non-Contact Liquid Level Sensor* XKC-Y25V level cairan mendeteksi level cairan, itu akan mengeluarkan *output* TINGGI dan menyalakan LED. Jika tidak ada cairan yang terdeteksi, *output*nya RENDAH dan mematikan LED.

Pada penelitian ini, sensor *Non-Contact Liquid Level Sensor* XKC-Y25V berfungsi untuk memantau atau menetapkan level cairan pada wadah yang digunakan. Kemudian, sensor *Non-Contact Liquid Level Sensor* XKC-Y25V akan mengirimkan *input* data ke *NodeMCU* ESP 8266.

### 3.4. PERANCANGAN SISTEM

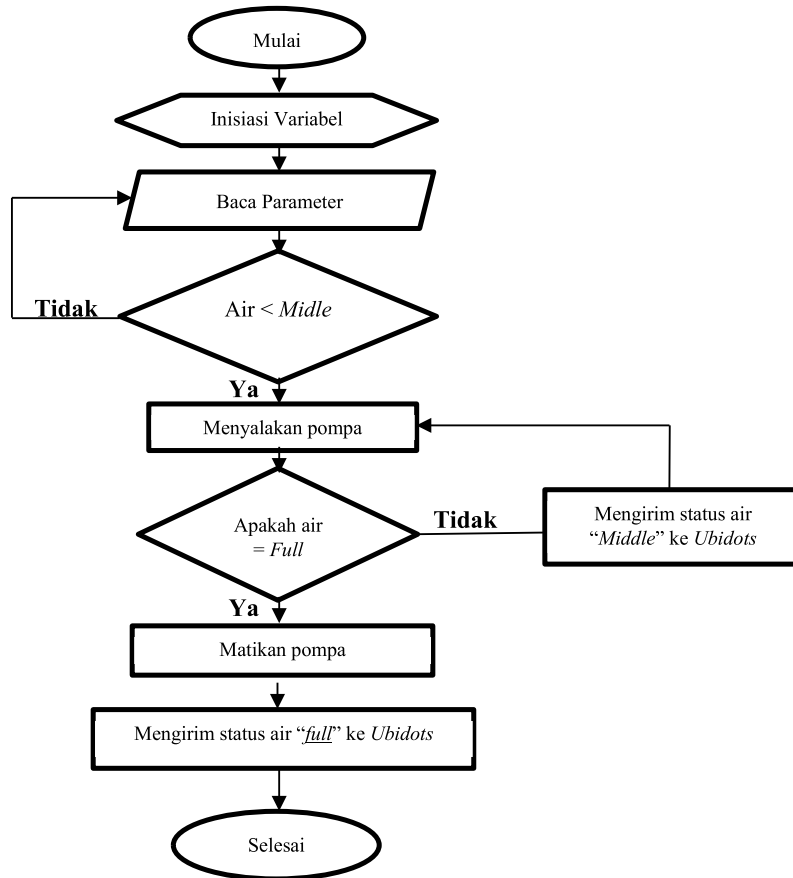
Sistem ini didesain sebagai berbasis IoT, yang memungkinkan pemantauan dan pengendalian dari jarak jauh melalui perangkat yang ada di telepon pintar. Komponen utama dari sistem ini mencakup sensor, mikrokontroler, aktuator, koneksi internet, dan telepon genggam.

#### 3.4.1. Perancangan Perangkat

Serangkaian pengujian tersebut adalah langkah penting dalam memastikan kinerja dan keandalan sistem pemantau. Dengan melakukan pengujian yang komprehensif seperti itu, dapat mengetahui kondisi sistem bisa beroperasi dalam berbagai kondisi dan sesuai kebutuhan pengguna. Perancangan sistem juga bertujuan untuk memeriksa secara menyeluruh perangkat dalam sistem, termasuk alat yang digunakan, melalui aplikasi *Ubidots Dashboard*.

#### 3.4.2. Pemasangan Perangkat

Pada gambar 3.3 di bawah ini terdapat *Flowchart* blok diagram alat.



Gambar 3.3 Blok Diagram Alat

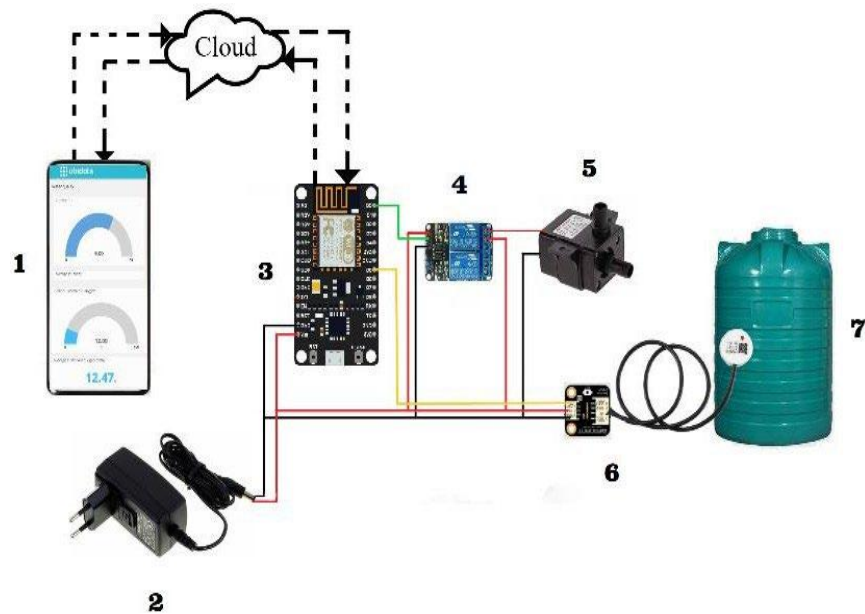
Pada gambar 3.3 menjelaskan gambaran cara kerja blok diagram alat yang disajikan, proses pertama alat setelah mulai adalah melakukan proses inisiasi variabel, kemudian setelah melakukan inisiasi variabel proses kedua adalah alat mulai membaca parameter. Alat yang bertugas untuk membaca parameter adalah sensor *Non-Contact Liquid Level Sensor XKC-Y25V* dimana sensor tersebut mendeteksi tentang keberadaan air yang berada di dalam wadah. Ketika sensor *Non-Contact Liquid Level Sensor XKC-Y25V* mulai mendeteksi adanya air maka kemudian sistem akan menyesuaikan dengan informasi yang di berikan oleh sensor. Bahwa informasi yang di berikan oleh sensor berupa *input* yang berupa sinyal digital.

Sensor *Non-Contact Liquid Level Sensor XKC-Y25V* memberikan *input* berupa sinyal digital yang bertugas memantau dan mengukur ketinggian air pada galon air yang digunakan selama proses pemantauan dan pengukuran. *Output* dari sensor *Non-Contact Liquid Level Sensor XKC-Y25V* tersebut sesuai data analog yang akan diteruskan ke *NodeMCU ESP8266*. Kemudian akan mentransmisikan data tersebut ke *Ubidots Dashboard*.

Terdapat tiga parameter pengukuran pada penelitian ini, yaitu *full*, *middle*, dan *empty*. Pada saat sensor *Non-Contact Liquid Level Sensor XKC-Y25V* melakukan pengukuran air dan menunjukkan kondisi air kurang dari *middle*, maka sensor akan memberikan sinyal ke *NodeMCU ESP8266* yang kemudian akan memberikan perintah untuk menyalakan pompa air secara otomatis guna mengisi wadah air tersebut sampai penuh. Selanjutnya apabila air lebih dari setengah/*middle* atau dalam kondisi *full*, maka sensor *Non-Contact Liquid Level Sensor XKC-Y25V* akan memberikan sinyal lanjutan dimana sinyal tersebut merupakan hasil dari sensor *Non-Contact Liquid Level Sensor XKC-Y25V* yang mendeteksi keberadaan air dalam kondisi *full* sehingga secara otomatis pompa akan mati.

Tahap terakhir setelah semua proses berhasil mendeteksi dua kondisi situasi diatas, maka kondisi selesai. *NodeMCU ESP8266* juga meneruskan informasi ke *Smartphone* melalui cloud untuk menampilkan informasi kepada pengguna melalui *dashboard Ubidots*.

Rangkaian diagram alat diperlihatkan pada gambar 3.4 sebagai berikut :



**Gambar 3.4 Rangkaian Diagram Alat**

Pada tabel 3.2 di bawah ini, menjelaskan tentang daftar rangkaian diagram alat yang terdapat pada gambar 3.4, yaitu:

**Tabel 3.2 Keterangan Gambar Rangkaian Diagram Alat**

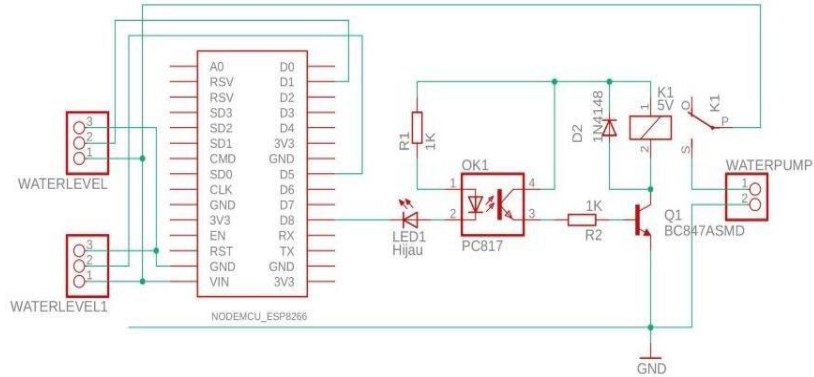
Gambar	Keterangan
1	<i>Dashboard Ubidots</i>
2	Power Suply
3	<i>NodeMCU 8266</i>
4	<i>Relay</i>
5	Pompa Air
6	Sensor
7	Tandon Air/Galon Air

Di bawah ini merupakan tabel 3.3 yang menjelaskan nama alat dan fungsi dari alat yang di gunakan.

**Tabel 3.3 Fungsi Perangkat Sensor**

Nama Alat	Fungsi
ESP8266	Mikrokontroler
<i>sensor Non-Contact Liquid Level Sensor XKC-Y25VXKC-Y25V</i>	Pengukur dan pemantau volume air

Rangkaian skematik alat terdapat pada gambar 3.5 di bawah ini:



**Gambar 3.5 Rangkaian Skematik Alat**

Pada gambar 3.5 menunjukkan rangkaian skematik alat dimana proses *input* berlangsung dari sensor *Non-Contact Liquid Level Sensor XKC-Y25V* yang akan di proses di *NodeMCU* ESP 8266 kemudian *NodeMCU* ESP 8266 akan memberikan intruksi kepada *relay* untuk menyalakan atau mematikan pompa air sesuai dengan kondisi *inputan* yang di berikan oleh sensor.

Tabel 3.4 di bawah ini berisikan tentang data koneksi pin *NodeMCU* ESP 8266 dengan sensor *Non-Contact Liquid Level Sensor XKC-Y25V*.

**Tabel 3.4 Koneksi Pin *NodeMCU* ESP 8266 dengan Sensor**

Pin	Komponen Pin	Fungsi
D1	Pin Data Sensor Water Level 1	Mengirimkan data level atas air sensor ke <i>NodeMCU</i>
D5	Pin Data Sensor Water Level 1	Mengirimkan data level bawah air sensor ke <i>NodeMCU</i>
D8	Pin Data <i>Relay</i>	Mengirimkan perintah On dan Off pada <i>relay</i>
VIN	VCC <i>Relay</i> , VCC Water Level 1, VCC Water Level 2	Sumber tegangan positif
GND	GND <i>Relay</i> , GND Water Level 1, GND Water Level 2	Sumber tegangan negatif

Tabel 3.4 diatas adalah tabel yang memberikan keterangan tentang koneksi pin *NodeMCU* ESP 8266 dengan sensor. Deskripsi yang diberikan menggambarkan proses fisik dalam merangkai komponen elektronik untuk

sistem yang disebutkan. Ini adalah langkah-langkah konkret dalam membangun rangkaian berdasarkan skematik yang telah dirancang sebelumnya.

Langkah-langkah tersebut termasuk merancang skematik, menyiapkan komponen, menghubungkan pin daya menggunakan *relay*, menghubungkan pin ground menggunakan adaptor konektor, menghubungkan pin daya adaptor dengan *relay*, dan menghubungkan pin *relay* menggunakan *NodeMCU*, termasuk koneksi antara pin VCC dan VIN *NodeMCU* serta pin GND dengan GND *NodeMCU*. Pin VCC dari *Non-Contact Liquid Level Sensor XKC-Y25V* dihubungkan dengan pin D1 dan D5 dari *NodeMCU*. Pin D8 digunakan sebagai pin data *relay* untuk mengirimkan perintah On dan Off pada *relay*.

Proses berikutnya adalah melakukan pembuatan *source code* pada *software* Arduino IDE dan mengunggahnya ke *NodeMCU* ESP 8266. Setelah itu, perangkat akan diuji menggunakan serangkaian tes sebagai berikut:

1. Tes Koneksi WiFi

Memastikan *NodeMCU* ESP 8266 terhubung ke jaringan WiFi dengan sukses.

2. Tes Sensor

Memeriksa apakah sensor *Non-Contact Liquid Level Sensor XKC-Y25V* mendeteksi level cairan dengan akurat.

3. Tes *Relay*

Mengaktifkan dan menonaktifkan *relay* untuk mengendalikan aliran listrik sesuai dengan tingkat cairan yang dideteksi oleh sensor *Non-Contact Liquid Level Sensor XKC-Y25V*.

4. Tes Komunikasi dengan *Ubidots Dashboard*

Memeriksa apakah data yang diperoleh dari sensor berhasil diunggah ke *Ubidots Dashboard*.

5. Tes Pengendalian Jarak Jauh

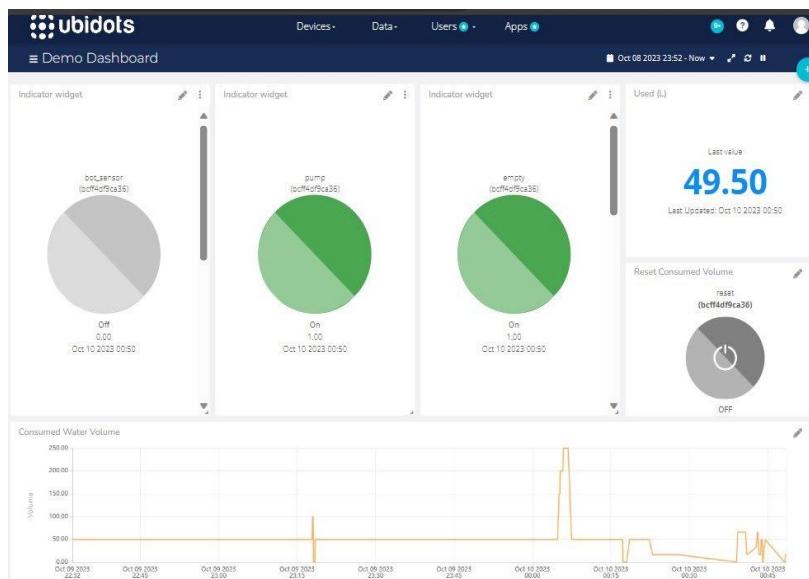
Tes pengendalian jarak jauh menggunakan *smartphone* atau laptop untuk mengamati sistem melalui jarak jauh dengan *Ubidots Dashboard*.

Pada tabel 3.5 membahas tentang perancangan aplikasi. Dimana di sini di jelaskan terkait skema pengujian dan hasil yang di dapatkan.

**Tabel 3.5 Perancangan Aplikasi**

Kategori	Skema Pengujian	Hasil
Aplikasi	Menampilkan data pengukuran dan pemantauan air	Menyajikan pengukuran air apakah kurang dari setengah atau lebih dari setengah
	Terhubung pada <i>realtime database</i>	Aplikasi terhubung dengan <i>Ubidots Dashboard</i>

Selanjutnya, pada gambar 3.6 di bawah ini disajikan gambar tampilan *Dashboard Ubidots* yang merupakan tampilan *Ubidots* pada laptop.

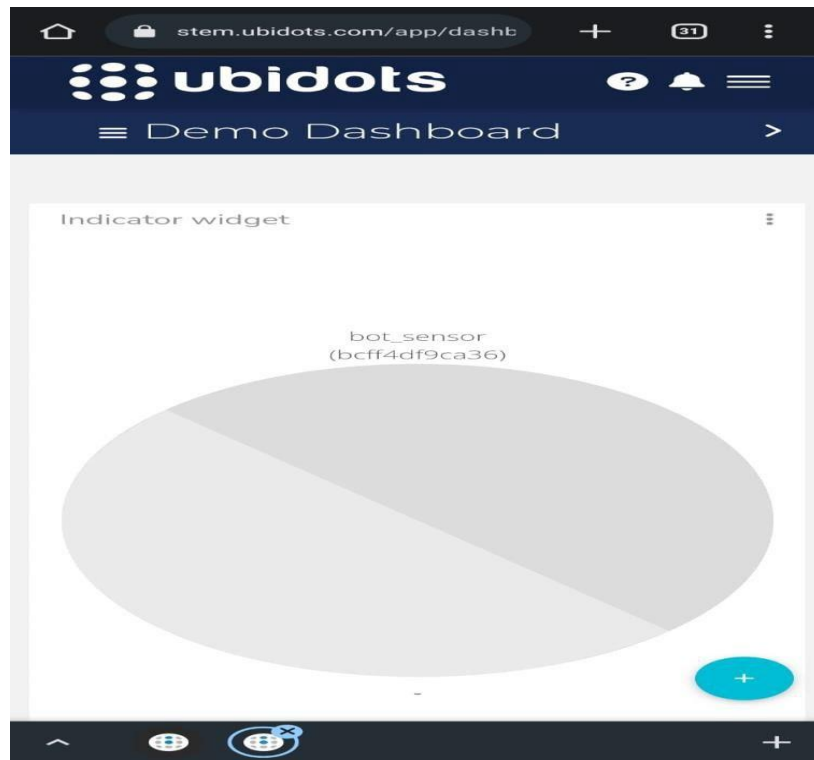


**Gambar 3.6 Tampilan *Dashboard Ubidots* pada Laptop**

Pada gambar 3.6 merupakan gambar tampilan *Dashboard Ubidots* yang di tampilkan melalui laptop. Dari tampilan tersebut terdapat informasi kondisi status sensor , kondisi status *water pump*, kondisi ketinggian air, dan informasi status volume air yang sudah di gunakan.



Pada gambar 3.7 di sajikan gambar tampilan *Dashboard Ubidots* melalui *handphone*.



**Gambar 3.7 Tampilan *Dashboard Ubidots* pada *Smartphone***

Pada gambar 3.7 merupakan gambar tampilan *Dashboard Ubidots* yang di tampilkan melalui *smartphone*. Tampilan yang di dapatkan sama dengan tampilan pada laptop pada gambar 3.6. Dari tampilan tersebut terdapat informasi kondisi status sensor , kondisi status *water pump*, kondisi ketinggian air, dan informasi status volume air yang sudah di gunakan.

### **3.4.3 Implementasi *Source code***

Setelah melakukan perakitan, selanjutnya peneliti melakukan pembuatan *source code* agar sistem dapat berfungsi sesuai dengan skenario perancangan sistem yang sudah dipaparkan pada subbab 3.4 tentang perancangan sistem. Pembuatan *source code* menggunakan Arduino IDE dengan versi 1.8.10 diawali dengan mengimport *library* yang diperlukan diantaranya *library* untuk mengakses fitur WiFi dan *library* untuk komunikasi mikrokontroler *NodeMCU* dengan platform *Ubidots* dan juga beberapa variable yang diperlukan.

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>

WiFiClient espClient;
PubSubClient client(espClient);

// Ubidots MQTT server and authentication
const char* mqttServer = "things.iot.Ubidots.com";
const int mqttPort = 1883;
const char* mqttUser = "khoerulanam";
const char* mqttPassword = "BBFF xYCdDLTL7m4qAMFVxSdwjdgTkvj9ly";
// Ubidots device and variable details
const char* deviceID = "bcff4df9ca36";

const char *ssid = "heyayo";
const char *pass = "12345678";

```

Pada bagian ini menunjukkan *source code* untuk *import library* dan beberapa variabel yang diperlukan diantaranya adalah *server*, *port*, *user*, token dan *device id* yang mana disesuaikan dengan akun *Ubidots* milik peneliti. Sedangkan untuk variabel “ssid” dan “pass” digunakan untuk menyimpan nama dan *password* yang digunakan untuk mengakses akses poin (WiFi).

Arduino IDE memiliki struktur fungsi *setup* dan *loop*. Untuk fungsi *setup* peneliti buat untuk menjalankan beberapa perintah awal pada saat alat pertama kali diaktifkan.

```

void setup() {
  Serial.begin(9600);
  delay(10);

  pinMode(top_sensor, INPUT);
  pinMode(bot_sensor, INPUT);
  pinMode(relay, OUTPUT);
  pinMode(D4, OUTPUT);
}

```

Pada *source code* fungsi *setup* berfungsi untuk perintah yang akan dijalankan. Fungsi ini diantaranya adalah perintah untuk set pin yang digunakan sebagai *input* dan *output*, perintah koneksi ke akses poin dan komunikasi dengan *Ubidots*.

Sedangkan fungsi *loop* peneliti buat untuk menjalankan perintah untuk membaca data sensor dan mengolah data untuk mengaktifkan *relay* serta untuk pengiriman data ke *Ubidots*.

```
void loop() {
  unsigned long now_time = millis();
  if (digitalRead(top_sensor) == HIGH && digitalRead(bot_sensor) == HIGH) {
    digitalWrite(relay, HIGH);
    full = 1;
    half = 0;
    empty = 0;
  } else if (digitalRead(top_sensor) == LOW && digitalRead(bot_sensor) ==
LOW) { digitalWrite(relay, LOW);
    full = 0;
    half = 0;
    empty = 1;
  } else if (digitalRead(top_sensor) == LOW && digitalRead(bot_sensor) ==
HIGH) {
    full = 0;
    half = 1;
    empty = 0;
  }
  if (digitalRead(relay) == LOW) {
    flag++;
    pump = 1;
  } else if (digitalRead(relay) == HIGH) {
    flag = 0;
    pump = 0;
  }
  if (flag == 1) {
    consumed += volume;}
}
```

*Source Code Fungsi Loop* merupakan program untuk menjalankan perintah membaca data sensor dan menjalankan *relay* sesuai kondisi sensor. Pada program tersebut juga terdapat data untuk menyimpan status dimana status ini akan dikirimkan ke *platform Ubidots*.

```
if (now_time - previous >= 5000) {
  // Subscribe to a topic of interest
  if (!client.connected()) {
    reconnect();
  }
  else {
    char topicToSubscribe[200];
    sprintf(topicToSubscribe, "%s", ""); // Cleans the content of the char
    sprintf(topicToSubscribe, "%s%s", "/v1.6/devices/", deviceID);
    sprintf(topicToSubscribe, "%s/%s/lv", topicToSubscribe, "reset");
    client.subscribe(topicToSubscribe);

    payload = "{\"top_sensor\": " + String (digitalRead(top_sensor)) + ", " +
    "\"bot_sensor\": " + String (digitalRead(bot_sensor)) + ", " + "\"full\": " + String
    (full) + ", " + "\"half\": " + String (half) + ", " + "\"empty\": " + String (empty) + ", "
    + "\"pump\": " + String (pump) + ", " + "\"consumed\": " + String (consumed) + "}";
    String publishTopic = String("/v1.6/devices/") + deviceID;
    Serial.println("Sending: " + payload);
    client.publish(publishTopic.c_str(), payload.c_str());
  }
  previous = now_time;
}
client.loop();
delay(500);
}
```

*Source code* pengiriman dan penerimaan data *ubidots* tersebut merupakan program untuk mengirimkan data status sensor maupun *relay* dan menerima data dari *Ubidots*. Semua variabel pada gambar sudah ditetapkan dari platform *Ubidots* sendiri agar perangkat atau mikrokontroler *NodeMCU* dapat terhubung dengan platform *Ubidots* dan dapat mengirimkan maupun menerima data. Fungsi *loop* ini akan berjalan berulang secara terus-menerus selama sistem diaktifkan.

```

void reconnect() {
  if (!client.connected()) {
    //Serial.println("Connecting to MQTT...");
    if (client.connect(deviceID, mqttPassword, "")) {
      digitalWrite(D4, LOW);
    } else {
      delay(500);
      digitalWrite(D4, HIGH);
      delay(500);
      digitalWrite(D4, LOW);
    }
  }
}
}

```

Selain fungsi *setup* dan *loop*, peneliti menambahkan fungsi tambahan “*reconnect*” yang berfungsi untuk menjalankan perintah mengulangi koneksi *NodeMCU* dengan *Ubidots*. Apabila *NodeMCU* tidak terhubung dengan platform *Ubidots* pada saat menjalankan perintah lain. Fungsi tersebut juga berisi perintah untuk mengedipkan lampu LED *NodeMCU* selama alat tidak terkoneksi dengan platform *Ubidots* dengan *interval* nyala dan mati masing-masing sebesar 500 *milisecond*.

### 3.5. SKENARIO PENGUJIAN SISTEM

Pembuatan skenario uji terdiri dari dua tahap: uji fungsionalitas dan evaluasi akurasi. Tujuan dari pengujian perangkat adalah untuk mengamati dan memverifikasi kinerja serta fungsionalitas prototipe. Proses pengujian melibatkan operasi perangkat dan penilaian kinerja setiap elemen, termasuk perangkat lunak yang terdiri dari program mikrokontroler untuk mengontrol alat dan *platform Internet of Things* untuk visualisasi data yang diterima.

Evaluasi pengujian dilakukan dengan membandingkan kinerja komponen dengan perintah yang di eksekusi. Selain itu, data yang diberikan oleh prototipe dibandingkan dengan data yang didapat oleh *platform Internet of Things* untuk mengevaluasi akurasi alat. Dalam metode skenario pengujian sistem penelitian ini menjelaskan parameter pengujian perangkat,

skenario pengujian perangkat mencakup dua aspek utama: pengujian fungsionalitas perangkat dan pengujian keseluruhan perangkat. Parameter pengujian perangkat dapat di lihat pada tabel 3.6.

**Tabel 3.6 Parameter Pengujian Perangkat**

No.	Quality	Skema Pengujian	Hasil
1.	Perangkat	Pengujian <i>Non-Contact Liquid Level Sensor XKC-Y25V</i>	Bertujuan untuk mengukur dan memantau volume air pada wadah yang disediakan
2.		Terhubung pada <i>realtime database</i>	Mikrokontroler <i>NodeMCU ESP8266</i> yang tersambung <i>Ubidots Dashboard</i>

Pada tabel 3.6 parameter pengujian perangkat akan mengevaluasi kelayakan dan tingkat keberhasilan prototipe sistem pemantau dan pengendali volume air.

### 3.5.1. Pengujian Perangkat

Metode skenario uji coba penelitian ini, terdapat dua aspek utama: pengujian fungsionalitas perangkat dan pengujian keseluruhan perangkat. Prototipe sistem kontrol dan pemantauan pada tangki air berbasis IoT menggunakan sensor untuk memantau volume air selama proses pengisian. Pengujian akan mengevaluasi kelayakan dan tingkat keberhasilan prototipe sistem pemantau dan pengendali volume air yang telah dibuat.

### 3.5.2. Pengujian Pengiriman Data

Berdasarkan pengujian pengiriman data, dapat dilakukan perhitungan akurasi dengan membandingkan *expected* status dengan *actual* status *relay* dan *water pump*. *Expected* status dapat diperoleh dengan menggunakan tabel 3.7 berikut :

**Tabel 3.7 Expected Output**

Sensor level		Expected Relay & Water pump
Atas	Bawah	
HIGH	HIGH	OFF
LOW	HIGH	OFF
LOW	LOW	ON

Dari tabel 3.7 *expected output* tersebut diperoleh data status *on/off* berdasarkan kondisi sensor level atas dan sensor level bawah, yang kemudian dibandingkan dengan status *actual*. Sehingga dapat diperoleh *confussion matrix* pada gambar 3.8 berikut :

		Actual	
		ON	OFF
Expected	ON	14	
	OFF		46

**Gambar 3.8 Confussion Matrix**

Untuk menghitung akurasi berdasarkan gambar 3.8 *confussion matrix* diperlukan 4 parameter yang didefinisikan:

TP (*True Positive*) = *Expected ON, Actual ON*

TN (*True Negative*) = *Expected OFF, Actual OFF*

FP (*False Positive*) = *Expected ON, Actual OFF*

FN (*False Negative*) = *Expected OFF, Actual ON*

Dengan parameter tersebut dapat diperoleh nilai akurasi dengan perhitungan berikut :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

$$Accuracy = \frac{14 + 46}{14 + 46 + 0 + 0} \times 100\%$$

$$Accuracy = \frac{60}{60} \times 100\% = 100\%$$

Dengan membandingkan status *expected* dengan *actual* dapat disimpulkan bahwa sistem mampu bekerja sesuai dengan rancangan dengan akurasi mencapai 100%.