

## **BAB III**

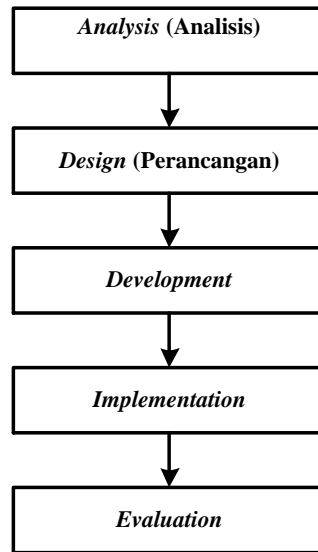
### **METODOLOGI PENELITIAN**

#### **3.1 Metode Penelitian**

Penelitian ini menggunakan pendekatan Research and Development (R&D). Research and Development (R&D) merupakan pendekatan sistematis yang digunakan untuk menghasilkan solusi praktis melalui proses yang tervalidasi [28]. R&D dalam penelitian ini bertujuan untuk merancang dan mengembangkan sebuah sistem *Smart Water Control System* berbasis *Internet of Things* (IoT) yang berfungsi mengendalikan pompa air secara otomatis menggunakan sensor pelampung serta menyediakan fitur kontrol jarak jauh melalui *Whatsapp*. Metode ini dipilih karena sesuai dengan tujuan utama penelitian, yaitu menghasilkan sebuah produk teknologi yang dapat diuji, dikembangkan, dan dievaluasi secara sistematis.

Pelaksanaan penelitian ini mengacu pada model pengembangan ADDIE, yang terdiri dari lima tahapan utama Analysis, Design, Development, Implementation, dan Evaluation. Model ADDIE dipilih karena memberikan struktur pengembangan yang terarah dan sistematis, mulai dari identifikasi kebutuhan hingga evaluasi efektivitas sistem.

Dalam konteks penelitian ini, model ADDIE digunakan untuk merancang dan membangun sistem *Smart Water Control* berbasis ESP32, sensor pelampung, *relay*, serta integrasi Twilio API secara terstruktur agar sistem dapat berfungsi optimal sesuai kebutuhan sekolah. Tahapan penelitian berdasarkan model ADDIE ditunjukkan pada gambar berikut.



**Gambar 3. 1 Alur Penelitian**

Adapun penjelasan tahapan pelaksanaan penelitian pada Gambar diatas adalah sebagai berikut.

1. *Analysis (Analisis)*

Tahap ini dilakukan dengan mengidentifikasi masalah pengelolaan air di SD Negeri 112305 Padang Halaban, khususnya proses pengisian tandon yang masih manual dan berpotensi menyebabkan pemborosan air, pompa terlambat dimatikan, dan kurang efisien bagi petugas sekolah. Analisis kebutuhan teknis dilakukan melalui observasi lapangan dan studi literatur tentang penggunaan *Internet of Things (IoT)*, *ESP32*, *Float Switch*, dan komunikasi *Whatsapp API*.

2. *Design (Perancangan)*

Tahap desain meliputi penyusunan rancangan perangkat keras dan perangkat lunak. Pada tahap ini disusun diagram blok sistem, skema rangkaian, *fLOWchart* kerja alat, perancangan logika deteksi level air, serta perancangan format perintah dan notifikasi *Whatsapp*. Komponen

utama yang dirancang meliputi ESP32, sensor pelampung (*Float Switch*), *relay*, pompa air, dan Twilio API.

### 3. *Development* (Pengembangan)

Tahap pengembangan melibatkan proses perakitan rangkaian elektronik dan pembuatan program menggunakan Arduino IDE. ESP32 diprogram untuk membaca kondisi air dari sensor pelampung, mengendalikan *relay* untuk pompa air, serta memproses perintah *Whatsapp* melalui Twilio API. Semua komponen diintegrasikan dan dipersiapkan untuk diuji.

### 4. *Implementation* (Implementasi)

Sistem yang telah dirancang dan dikembangkan kemudian dipasang pada lokasi penelitian, yaitu area tandon air di SD Negeri 112305 Padang Halaban. Pengujian dilakukan untuk memastikan bahwa sistem dapat menyalakan atau mematikan pompa secara otomatis, merespon perintah *Whatsapp*, dan mengirimkan notifikasi status air secara *real-time*.

### 5. *Evaluation* (Evaluasi)

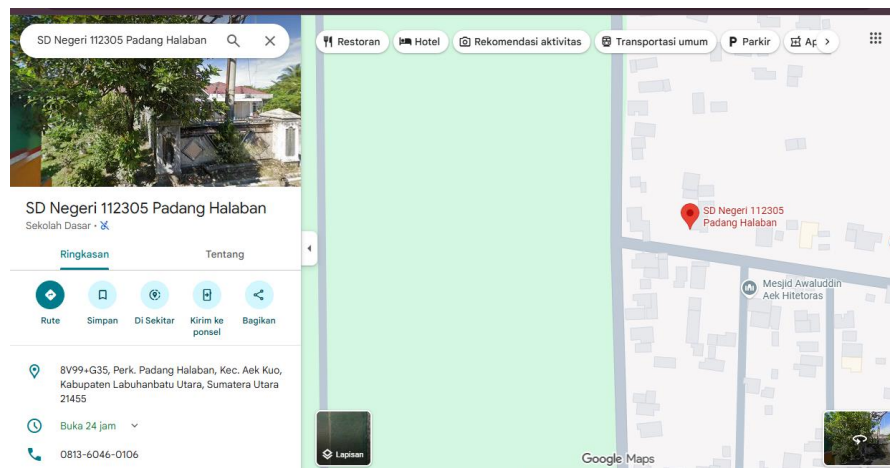
Tahap evaluasi dilakukan untuk menilai akurasi pembacaan sensor pelampung, keandalan *relay* dan pompa, kecepatan respons *Whatsapp*, serta efektivitas sistem dalam mencegah tandon meluap. Hasil evaluasi menjadi dasar untuk perbaikan sistem agar dapat bekerja lebih optimal.

## **3.2 Tempat dan Waktu Penelitian**

### **3.2.1 Tempat Penelitian**

Penelitian ini dilaksanakan di SD Negeri 112305 Padang Halaban, Kabupaten Labuhanbatu Utara, Provinsi Sumatera Utara. Lokasi ini dipilih karena

sekolah masih menggunakan sistem pengisian tandon secara manual sehingga sangat membutuhkan sistem otomatis berbasis *Internet of Things (IoT)* untuk meningkatkan efisiensi, mengurangi pemborosan air, serta membantu petugas sekolah dalam pengelolaan fasilitas air bersih.



**Gambar 3. 2 Tempat Penelitian**

Selain itu, lingkungan sekolah memiliki kondisi instalasi tandon dan pompa air yang memungkinkan penerapan sistem *Smart Water Control* berbasis ESP32, sensor pelampung, dan *Whatsapp* sebagai media kontrol jarak jauh.

### 3.2.2 Waktu Penelitian

Penelitian ini dilaksanakan dalam rentang waktu Desember 2025 hingga April 2026, yang mencakup beberapa tahap berikut:

**Tabel 3. 1 Tahapan Penelitian**

No	Tahap Penelitian	Bulan				
		November	Desember	Januari	Februari	Maret
1	Analisis kebutuhan					
2	Perancangan sistem					
3	Pengembangan perangkat keras dan perangkat					

	lunak					
4	Implementasi dan pemasangan alat					
5	Pengujian dan evaluasi					
6	Penyusunan laporan penelitian					

### 3.3 Alat dan Bahan Penelitian

Pada penelitian ini, alat dan bahan yang digunakan dibagi menjadi dua bagian, yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Seluruh alat dan bahan ini mendukung proses perancangan, pembuatan, serta pengujian sistem *Smart Water Control* berbasis *Internet of Things* (IoT).

#### 3.3.1 Alat dan Bahan Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam penelitian ini ditunjukkan pada tabel berikut:

**Tabel 3. 2 Daftar Alat dan Bahan Perangkat Keras**

No	Nama Alat dan Bahan	Fungsi
1	ESP32 <i>Development Board</i>	Mikrokontroler utama yang memproses input sensor dan mengendalikan <i>relay</i> serta komunikasi IoT.
2	Sensor Pelampung ( <i>Float Switch</i> )	Mendeteksi level air (penuh atau rendah) pada tandon untuk menjalankan otomatisasi pompa.
3	Modul <i>Relay</i> 1 Channel	Menghubungkan dan memutus arus listrik ke pompa air berdasarkan perintah ESP32.
4	Pompa Air Listrik	Mengisi tandon air sekolah sesuai perintah sistem.
5	Adaptor 5V & 12V	Menyediakan sumber daya untuk ESP32 dan pompa air.
6	Kabel Jumper & Kabel Power	Penghubung antar-komponen pada rangkaian.
7	<i>Breadboard</i> Atau <i>Project Board</i>	Media perakitan rangkaian elektronik.
8	Box Panel	Melindungi rangkaian agar aman digunakan di lingkungan sekolah.
9	Multimeter	Menguji tegangan, kontinuitas, dan memastikan

		instalasi komponen berjalan baik.
10	<i>Smartphone</i>	Digunakan untuk mengirim perintah dan menerima notifikasi melalui <i>Whatsapp</i> .

### 3.3.2 Alat dan Bahan Perangkat Lunak (Software)

Perangkat lunak yang digunakan dalam penelitian ini ditampilkan pada tabel berikut:

**Tabel 3. 3 Daftar Perangkat Lunak**

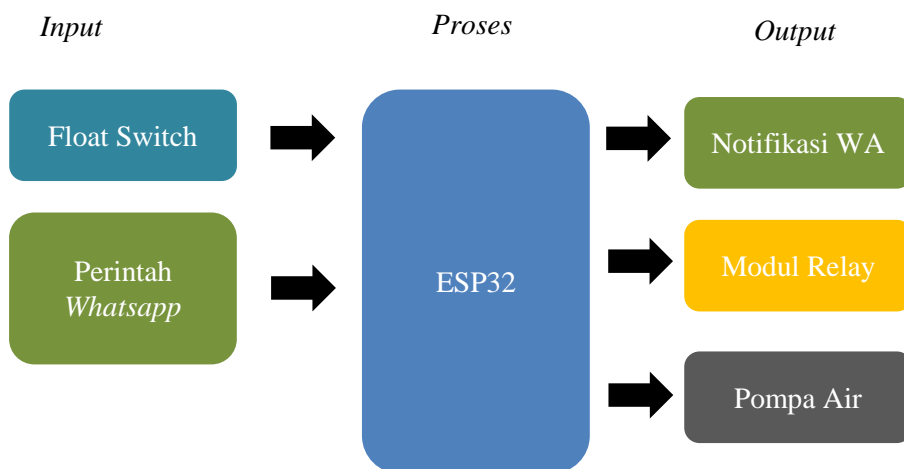
No	Software	Fungsi
1	Arduino IDE	Menulis, mengompilasi, dan mengunggah program ke ESP32.
2	Twilio <i>Whatsapp</i> API	Media pengiriman notifikasi dan kontrol sistem melalui <i>Whatsapp</i> .
3	Browser (Chrome)	Mengakses <i>dashboard</i> Twilio dan dokumentasi API.
4	Fritzing	Mendesain rangkaian elektronik dan memvisualisasikan wiring.
5	<i>Whatsapp</i>	Aplikasi utama kontrol dan pemantauan sistem.

### 3.4 Perancangan Sistem

Pada tahap ini dilakukan proses perancangan sistem sebagai dasar dalam membangun *Smart Water Control System* berbasis *Internet of Things* (IoT) yang mampu mengendalikan pompa air secara otomatis menggunakan sensor pelampung serta memberikan fitur kontrol jarak jauh melalui aplikasi *Whatsapp*. Perancangan sistem mencakup penyusunan blok diagram, alur kerja sistem, rancangan perangkat keras, rancangan perangkat lunak, hingga arsitektur komunikasi antara *Whatsapp*, Twilio API, dan ESP32. Setiap bagian dirancang secara terstruktur agar sistem mampu bekerja secara stabil, responsif, serta mudah diimplementasikan pada lingkungan sekolah. Perancangan ini juga menjadi acuan utama pada tahap pengembangan dan implementasi sistem.

### 3.4.1 Diagram Blok Sistem

Diagram blok sistem digunakan untuk menggambarkan hubungan kerja antar komponen utama yang membentuk *Smart Water Control System* berbasis *Internet of Things* (IoT). Sistem ini terdiri dari beberapa unit fungsional, yaitu sensor pelampung sebagai pendeteksi level air, ESP32 sebagai pusat pengendali, modul *relay* sebagai aktuator pemutus dan penghubung arus pompa, serta Twilio API yang menghubungkan sistem dengan *Whatsapp* untuk menerima perintah dan mengirimkan notifikasi. Melalui blok diagram ini, alur kerja sistem dapat dipahami secara menyeluruh mulai dari bagaimana data diterima, diproses, hingga menghasilkan aksi pada perangkat fisik. Penyajian blok diagram ini juga memudahkan proses pengembangan, pengujian, dan identifikasi bila terjadi gangguan pada sistem.



**Gambar 3. 3 Blok Diagram**

#### Deskripsi Blok Diagram Sistem

##### 1. *Input*

Sensor Pelampung (*Float Switch*) mengirimkan sinyal digital ke ESP32 mengenai kondisi air: penuh atau rendah. Perintah *Whatsapp* dari pengguna (ON/OFF/status) dikirim melalui *Whatsapp* → Twilio API →

server/device penerima → ESP32.

## 2. Proses

ESP32 menerima dan memproses data sensor pelampung untuk menentukan apakah pompa harus dinyalakan atau dimatikan secara otomatis. ESP32 menerima perintah manual dari *Whatsapp* melalui Twilio API dan mengeksekusi instruksi pengguna. ESP32 menjalankan logika kontrol:

- Jika air penuh maka sistem mematikan pompa.
- Jika air rendah maka sistem menghidupkan pompa (kecuali jika mode manual OFF).

ESP32 membentuk pesan status (air penuh, air rendah, pompa ON/OFF).

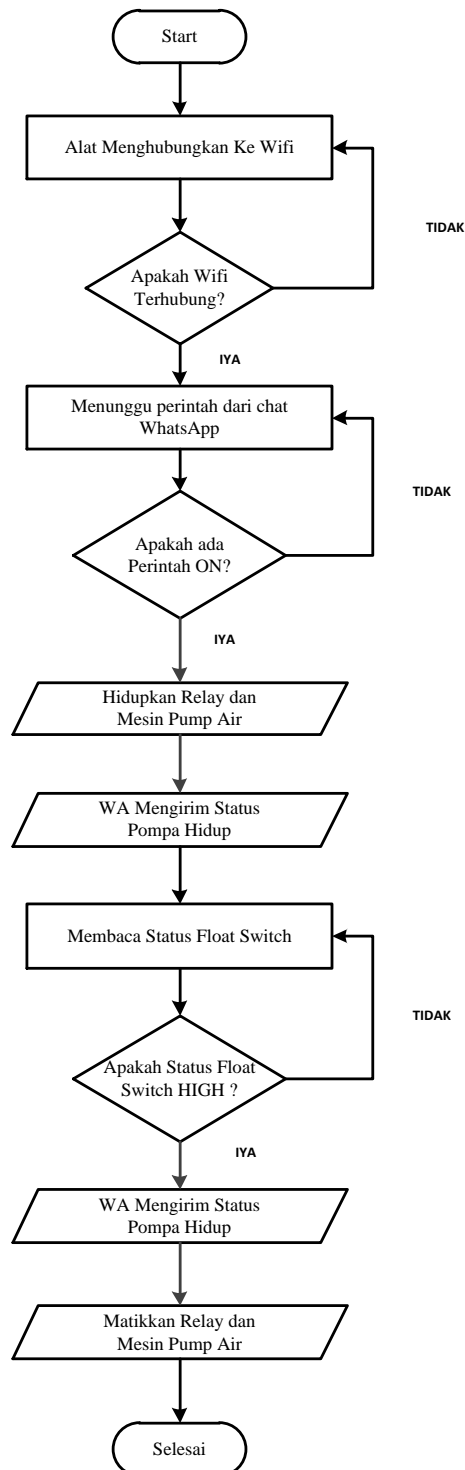
## 3. Output

Modul *Relay* mengaktifkan atau memutus arus ke pompa sesuai instruksi ESP32. *Whatsapp* mengirimkan notifikasi status melalui Twilio API ke pengguna (misalnya: “Air penuh, pompa dimatikan otomatis”). Pompa air bekerja untuk mengisi tandon sesuai keputusan sistem.

### 3.4.2 Diagram Alir (*FLOWchart*)

Diagram alir pada sistem ini digunakan untuk menjelaskan alur kerja *Smart Water Control System* berbasis IoT yang mengendalikan pompa air menggunakan satu sensor pelampung (*Float Switch*) dan kontrol jarak jauh melalui aplikasi *Whatsapp*. Karena sistem hanya menggunakan satu *Float switch* yang dipasang pada posisi penuh (*upper level*), maka logika otomatis yang digunakan adalah pemutusan pompa ketika air telah mencapai batas penuh. *FLOWchart* ini menggambarkan proses mulai dari sistem menyala, membaca

sensor, menerima perintah *Whatsapp*, hingga menjalankan logika otomatisasi.



**Gambar 3. 4 FLOWchat Sistem**

*FLOW*chart pada gambar menjelaskan alur kerja utama sistem Smart Water Control berbasis ESP32, di mana pompa air dapat dikendalikan melalui

perintah *Whatsapp* serta otomatis mati saat tandon penuh. Berikut penjelasan rinci setiap langkah:

1. *Start*

Proses dimulai ketika ESP32 dihidupkan. Sistem masuk ke tahap inisialisasi perangkat keras dan perangkat lunak.

2. Alat Menghubungkan ke WiFi

Pada tahap ini ESP32 mencoba menghubungkan diri ke jaringan WiFi sesuai SSID dan *password* yang telah ditentukan dalam program. Koneksi WiFi merupakan bagian penting karena komunikasi *Whatsapp* menggunakan layanan Twilio membutuhkan jaringan internet.

3. Apakah WiFi Terhubung?

Sistem melakukan pengecekan apakah koneksi WiFi berhasil:

- Jika berhasil, proses dilanjutkan ke langkah berikutnya.
- Jika gagal, ESP32 akan terus mencoba kembali hingga koneksi sukses. Tahap ini memastikan sistem siap menerima perintah *Whatsapp*.

4. Menunggu Perintah dari Chat *Whatsapp*

Setelah WiFi terhubung, ESP32 masuk ke mode *monitoring* perintah. Sistem menunggu pesan *Whatsapp* dari pengguna yang dikirim melalui Twilio API. Perintah ini menjadi kendali utama untuk mengaktifkan pompa air.

5. Apakah Ada Perintah ON?

ESP32 memeriksa apakah pesan *Whatsapp* berisi instruksi ON:

Jika ya, sistem melanjutkan proses untuk mengaktifkan pompa.

Jika tidak ada, sistem tetap berada pada mode menunggu perintah. Logika ini menjadikan *Whatsapp* sebagai kontrol manual yang memicu *start* pompa.

#### 6. Hidupkan *Relay* dan Mesin Pompa Air

Jika perintah ON diterima, ESP32 mengaktifkan *relay*. *Relay* menghubungkan sumber listrik ke mesin pompa sehingga pompa mulai bekerja. Pompa kemudian mengalirkan air ke tandon hingga mencapai batas yang ditentukan.

#### 7. *Whatsapp* Mengirim Status Pompa Hidup

Jika Perintah ON diterima maka ESP32 Memerintahkan Bot *Whatsapp* untuk mengirim Status "Pompa hidup.

#### 8. Membaca Status *Float Switch*

Setelah pompa menyala, ESP32 secara berkala membaca status *Float switch* yang dipasang pada posisi "air penuh". Sensor ini memberikan informasi apakah level air telah mencapai batas maksimum.

#### 9. Apakah Status *Float switch HIGH*?

Logika sensor diperiksa:

- Jika status *HIGH*, berarti pelampung naik dan tandon telah penuh.
- Jika status *LOW*, berarti pelampung belum naik dan tandon masih perlu diisi. Jika sensor belum *HIGH*, sistem kembali melakukan pembacaan sampai kondisi penuh tercapai.

#### 10. Matikan *Relay* dan Mesin Pompa Air

Ketika *Float switch* mendeteksi kondisi *HIGH*, ESP32 mematikan *relay* sehingga aliran listrik ke pompa terputus. Pompa berhenti bekerja secara

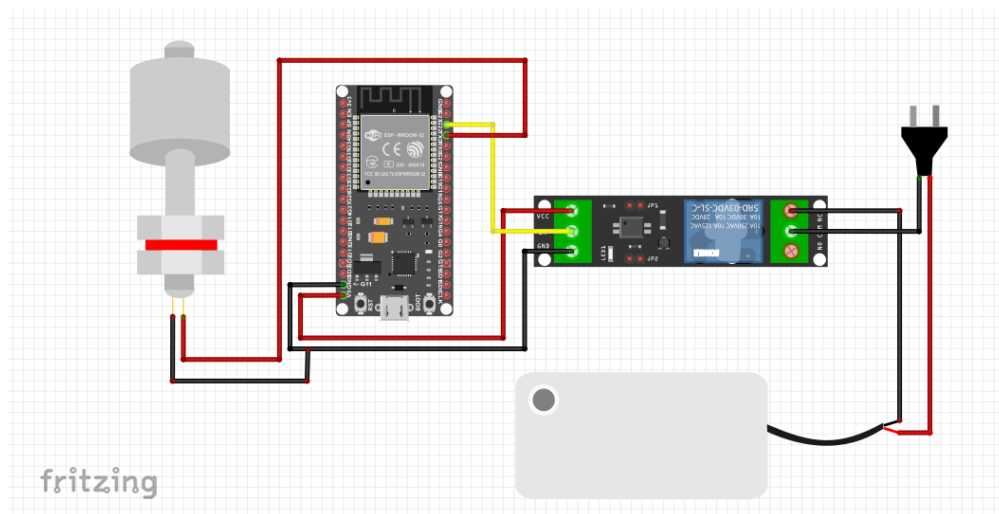
otomatis untuk mencegah tandon meluap.

#### 11. Selesai

Proses berakhir setelah pompa dimatikan. Pada implementasi sebenarnya, sistem tidak benar-benar berhenti, tetapi kembali ke loop pengawasan untuk menerima perintah *Whatsapp* berikutnya dan membaca status sensor secara berulang.

### 3.4.3 Perancangan Perangkat Keras

Perancangan perangkat keras bertujuan membangun rangkaian fisik yang menghubungkan ESP32 sebagai *controller*, sensor pelampung (*Float Switch*) sebagai *input* level penuh, modul *relay* sebagai aktuator untuk mengendalikan pompa, serta sumber daya yang sesuai. Rangkaian dibuat sederhana: ESP32 membaca status *Float Switch*, kemudian menggerakkan modul *relay* untuk memutus dan menghubungkan arus ke pompa. Diagram Fritzing menunjukkan tata letak modul yang digunakan sebagai *prototype* sebelum pemasangan akhir di box pelindung. Rangkain keseluruhan perangkat keras dapat dilihat pada gambar dibawah ini.



**Gambar 3. 5 Rangkaian Keseluruhan**

Berikut adalah tabel koneksi antar-komponen yang digunakan dalam rangkaian diatas.

**Tabel 3. 4 Wiring Keseluruhan**

No	Dari	Ke	Keterangan
1	ESP32 GND	<i>Relay</i> GND & Power supply (GND)	Semua GND harus common (ESP32 + <i>relay</i> Vcc GND)
2	ESP32 5V (VUSB) / 5V regulator	<i>Relay</i> VCC (JD-Vcc jika ada)	Jika modul <i>relay</i> membutuhkan 5V. Jika modul 3.3V-trigger, hubungkan VCC ke 3.3V ESP32.
3	ESP32 GPIO GPIO 21	<i>Relay</i> IN	Output digital untuk mengaktifkan <i>relay</i> ( <i>LOW/HIGH</i> sesuai modul). Contoh: gunakan GPIO21.
4	ESP32 GPIO GPIO 22	<i>Float switch</i> → (ke GPIO)	<i>Float switch</i> ke <i>input</i> GPIO (gunakan <i>INPUT_PULLUP</i> ). Satu sisi float ke GND, sisi lain ke GPIO.
5	<i>Relay</i> COM	Salah satu kabel pompa (fase)	<i>Relay</i> memutusdan menyambung jalur pompa; sisi lain pompa ke netral (atau sesuai <i>wiring</i> ).
6	Power supply mains L & N	Pompa & <i>relay</i> COM/NO (melalui <i>relay</i> )	Pastikan pemasangan aman dan memakai <i>fuse</i> .