

PROPOSAL TUGAS AKHIR

SISTEM MONITORING DAN KONTROL TINGGI PERMUKAAN AIR BERBASIS INTERNET OF THING (IOT) MENGGUNAKAN ESP32 DAN BLNYK

Diajukan Untuk Memenuhi Satu Syarat Program Studi DIII Teknik Elektronika Untuk
Memperoleh Gelar Ahli Madia Fakultas Teknik Elektronika
Universitas Negeri Padang



Oleh :

Fajar ramadhan (21066010)

**PRODI DIII TEKNIK ELEKTRONIKA
JURUSAN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGRI PADANG
2024**

PERSETUJUAN PEMBIMBING

PROYEK AKHIR

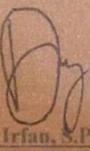
Judul : Sistem Monitoring Dan Kontrol Tinggi Permukaan Air Berbasis Internet Of Thing (IOT) Menggunakan ESP32 Dan Blynk
Nama : Fajar Ramadhan
NIM : 21066010
Program Studi : D3 Teknik Elektronika
Departemen : Teknik Elektronika
Fakultas : Teknik

Padang, 03 Juni 2025

Disetujui Oleh :
Pembimbing


NIP. 196312091988031004

Mengetahui
Kepala Departemen


Dr.Dedy Irfan, S.Pd., M.Kom
NIP. 197604082005011092

HALAMAN PENGESAHAN PROYEK AKHIR

Nama : Fajar Ramadhan
NIM : 21066010

Dinyatakan lulus setelah mempertahankan didepan Tim penguji Proyek Akhir
Program Studi Teknik Elektronika Departemen Teknik Elektronika Universitas
Negeri Padang dengan judul

**Sistem Monitoring Dan Kontrol Tinggi Permukaan Air Berbasis Internet Of
Thing (IOT) Menggunakan ESP32 Dan Blynk**

Padang, 03 Juni 2025

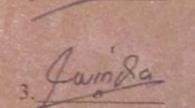
Tim penguji :

1. Dr. Edidas, M.T

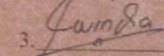
Tanda Tangan



2. Dr. Zulwisli, S.Pd, M.Eng.



3. Winda Agustiarmi,S.Pd, M.Pd.T



PERNYATAAN

Dengan ini saya menyatakan bahwa :

1. Karya tulis saya, tugas akhir berupa proyek akhir dengan judul Sistem Monitoring Dan Kontrol Tinggi Permukaan Air Berbasis Internet Of Thing (IOT) Menggunakan ESP32 Dan Blynk adalah asli karya saya sendiri;
2. Karya tulis ini murni gagasan, rumusan dan penilitian saya sendiri, dan bantuan dari pembimbing;
3. Pernyataan ini saya buat dengan sesungguhnya dan apabila terdapat penyimpangan di dalam pernyataan ini saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai norma dan ketentuan hukum yang berlaku.

Padang, 03 Juni 2025
Yang membuat pernyataan



Fajar Ramadhan
NIM :21066010

ABSTRAK

Fajar Ramadhan Sistem Monitoring Dan Kontrol Tinggi Permukaan Air Berbasis
(21066010) Internet Of Thing (IoT) Menggunakan ESP32 Dan Blynk

Air adalah sumber daya vital yang pemanfaatannya perlu dikontrol agar tidak menimbulkan pemborosan maupun kerusakan. Penelitian ini bertujuan merancang sistem monitoring dan kontrol tinggi permukaan air berbasis Internet of Things (IoT) menggunakan ESP32, sensor ultrasonik HC-SR04, dan platform Blynk. Sistem dirancang mampu mendeteksi tiga kondisi: aman, waspada, dan bahaya, berdasarkan jarak air terhadap sensor. Komponen seperti LED, buzzer, LCD I2C, dan servo motor digunakan untuk menampilkan status dan melakukan tindakan otomatis. Simulasi dilakukan di Wokwi dan hasilnya menunjukkan bahwa sistem bekerja efektif dan responsif. Sistem ini memberikan solusi praktis untuk mencegah luapan air, sekaligus mendukung efisiensi air di lingkungan rumah tangga dan penerapan teknologi smart home sederhana.

Kata Kunci : Internet of Thing, ESP32, sensor ultrasonik, Blynk, smart home, monitoring air.

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan penyusunan proposal tugas akhir ini dengan baik. Shalawat serta salam senantiasa tercurah kepada junjungan kita Nabi Muhammad SAW, yang telah membawa kita dari kegelapan menuju cahaya kebenaran.

Penyusunan proposal ini merupakan bagian dari rangkaian upaya untuk mengatasi permasalahan kelebihan air pada bak mandi melalui pengembangan sistem monitoring tinggi permukaan air berbasis Internet of Things (IoT). Kami menyadari bahwa proses penyusunan proposal ini tidak terlepas dari bimbingan, dukungan, serta motivasi dari berbagai pihak yang turut serta mendukung.

Pada kesempatan ini, penulis ingin menyampaikan penghargaan yang setinggi-tingginya kepada:

1. Kepada Allah, Ar - Rahman, Ar - Rahiim
2. Bapak Dr. Muhammad Anwar, S.Pd., M.T., selaku Dekan Fakultas Teknik Universitas Negeri Padang
3. Bapak Dr. Hendra Hidayat, S.Pd., M.Pd., selaku Kepala Departemen Teknik Elektronika Fakultas Teknik Universitas Negeri Padang
4. Bapak Dr. Yasdinul Huda, S.Pd., M.T., selaku Koordinator Prodi Teknik Elektronika Fakultas Teknik Universitas Negeri Padang.
5. Bapak Dr. Zulwisli, S.Pd., M.Eng., selaku Pembimbing yang telah banyak memberi masukan dan saran dalam penyelesaian Proyek Akhirini.
6. Alm Bapak Fujianto dan Ibu Erlinda Ningsih serta Abang Fahmi Prasetyawan yang sudah memberikan dukungan dan doa sehingga bisa menyelesaikan Proyek Akhir ini.

7. Tiwi Wulandari, yang selalu mendukung, menyemangati, dan mendoakan dengan sepenuh hati. Kehadirannya dan dukungannya menjadi sumber inspirasi dalam menyelesaikan proyek ini.

Tanpa bantuan dan dukungan dari berbagai pihak tersebut, penyusunan proposal ini tidak akan dapat terwujud dengan baik. Semoga upaya yang kami lakukan dapat memberikan manfaat serta kontribusi yang positif bagi pengembangan ilmu pengetahuan dan teknologi.

Akhir kata, kami mohon maaf atas segala kekurangan dan kesalahan yang terdapat dalam proposal ini. Semoga proposal ini dapat menjadi langkah awal yang baik dalam mewujudkan solusi yang efektif terhadap permasalahan kelebihan air pada bak mandi.

Padang, 2025

[Penulis]

DAFTAR ISI

	Halaman
HALAMAN PERSETUJUAN	i
HALAMAN PENGESAHAN.....	ii
PERNYATAAN.....	iii
KATAPENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	viii
DAFTAR TABEL	ix
DAFTAR LAMPIRAN.....	x
BAB I PENDAHULUAN.....	1
A. Latar Belakang	1
B. Indetifikasi Masalah	3
C. Rumusan Masalah.....	4
D. Batasan Masalah.....	4
E. Tujuan	5
F. Manfaat	5
BAB II HASIL DAN PEMBAHASAN.....	6
A. Analisis Kebutuhan Proyek.....	6
B. Desain Proyek Akhir	22
C. Deskripsi Hasil	31
BAB III SIMPULAN DAN REKOMENDASI.....	38
A. Simpulan.....	38
B. Rekomendasi	39
DAFTAR PUSTAKA.....	40
LAMPIRAN.....	41

DAFTAR GAMBAR

	Halaman
Gambar 1. isi dalam platform wokwi	7
Gambar 3. ESP32.....	12
Gambar 4. sensor ultrasonik HC-SR04.....	14
Gambar 5. DHT22	14
Gambar 6. motor servo	16
Gambar 7. Blynk.....	18
Gambar 8. LED (<i>Light Emitting Diode</i>)	19
Gambar 9. Resistor.....	20
Gambar 10. Breadboard.....	20
Gambar 11. Buzzer	21
Gambar 12. Lcd2004	22
Gambar 13. Diagram Blok	23
Gambar 14. flowchart sistem monitoring ketingian permukaan air pada simulasi wokwi.....	25
Gambar 15. Hasil desain sistem monitoring ketingian permukaan air pada simulasi wokwi.....	27
Gambar 16. Kode program pada simulasi wokwi.....	30
Gambar 17. Tampilan Web Dashboard Aplikasi Blynk	31
Gambar 18. Hasil simulasi pada wokwi	32
Gambar 19. Hasil Tampilan Blynk	35
Gambar 20. Tampilan Notifikasi Blynk.....	35

DAFTAR TABEL

Halaman

Tabel 1. Hasil Pengujian Alat Keseluruhan	36
---	----

DAFTAR LAMPIRAN

halaman

LAMPIRAN 1.....41

BAB I

PENDAHULUAN

A. Latar Belakang

Air merupakan sumber daya alam yang sangat vital bagi kelangsungan hidup semua makhluk di bumi. Keberadaannya tidak hanya penting untuk memenuhi kebutuhan dasar manusia seperti minum, mandi, dan mencuci, tetapi juga memiliki peran krusial dalam berbagai sektor seperti industri, pertanian, dan pembangkit listrik. Namun, pemanfaatan air yang tidak terkontrol dapat menimbulkan masalah, baik ketika jumlahnya kurang maupun berlebihan. Salah satu masalah yang sering terjadi dalam kehidupan sehari-hari adalah meluapnya air pada bak mandi akibat kelalaian dalam memantau ketinggian air. Hal ini tidak hanya menyebabkan pemborosan air, tetapi juga berpotensi menimbulkan kerusakan material dan infrastruktur di sekitar rumah.

Untuk mengatasi permasalahan ini, diperlukan sebuah sistem yang mampu memantau ketinggian air secara otomatis dan memberikan peringatan kepada pengguna ketika air mencapai batas tertentu. Selain itu, sistem ini juga harus mampu melakukan kontrol otomatis untuk mencegah meluapnya air. Teknologi *Internet of Things* (IoT) menawarkan solusi yang efektif untuk memenuhi kebutuhan tersebut. memanfaatkan IoT, sistem monitoring dan kontrol ketinggian air dapat dirancang untuk bekerja secara real-time, memberikan notifikasi, dan melakukan tindakan otomatis seperti menutup keran air ketika ketinggian air mencapai batas maksimum.

Salah satu platform yang dapat digunakan untuk mengembangkan sistem ini adalah Framework Blynk, yang memungkinkan pengguna untuk memantau dan

mengontrol perangkat IoT melalui antarmuka yang user-friendly. Dalam penelitian ini, penulis mengusulkan Sistem Monitoring dan Kontrol Tinggi Permukaan Air Berbasis IoT Menggunakan ESP32 dan Framework Blynk. ESP32 dipilih sebagai mikrokontroler utama karena kemampuannya dalam menghubungkan perangkat ke internet melalui WiFi, serta kemudahannya dalam mengintegrasikan sensor dan perangkat lain. Sensor ultrasonik digunakan untuk mengukur ketinggian air secara akurat, sementara Blynk berfungsi sebagai platform untuk menampilkan data secara real-time dan memberikan notifikasi ketika ketinggian air mencapai batas yang telah ditentukan.

Selain itu, sistem ini dilengkapi dengan servo motor yang berfungsi sebagai indikator penutup keran air secara otomatis ketika ketinggian air melebihi batas yang ditentukan. adanya fitur kontrol ini, sistem tidak hanya memberikan peringatan, tetapi juga mengambil tindakan otomatis untuk mencegah meluapnya air, sehingga mengurangi risiko kerusakan dan pemborosan air.

Penggunaan simulasi Wokwi sebagai platform pengujian awal memungkinkan untuk merancang dan menguji sistem sebelum diimplementasikan secara nyata. Simulasi ini membantu memastikan bahwa sistem dapat bekerja dengan baik dan mengurangi risiko kegagalan saat implementasi di lapangan. Dengan demikian, sistem ini tidak hanya membantu mengurangi pemborosan air, tetapi juga mencegah kerusakan akibat meluapnya air, serta meningkatkan efisiensi penggunaan air di rumah tangga.

Penelitian ini juga memiliki relevansi yang tinggi dengan upaya pelestarian sumber daya air. meminimalkan pemborosan air, sistem ini turut mendukung keberlanjutan lingkungan dan konservasi air. Selain itu, penerapan teknologi IoT dalam sistem monitoring dan kontrol ketinggian air membuka peluang untuk

pengembangan lebih lanjut, seperti integrasi dengan sistem smart home atau aplikasi lain yang memerlukan pemantauan dan kontrol ketinggian cairan.

Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi nyata dalam mengatasi permasalahan meluapnya air pada bak mandi, sekaligus memperkenalkan penggunaan teknologi IoT yang lebih canggih dan efisien dalam kehidupan sehari-hari. Melalui kombinasi ESP32, sensor ultrasonik, servo motor, dan Framework Blynk, sistem ini dirancang untuk menjadi solusi yang praktis, ekonomis, dan ramah lingkungan.

Sistem yang direncanakan dalam penelitian ini merupakan suatu solusi terintegrasi yang menggabungkan sensor ultrasonik HC-SR04, mikrokontroler ESP32, servo motor, dan platform Blynk. Sistem ini dirancang untuk memantau ketinggian air secara real-time, memberikan notifikasi kepada pengguna saat air mendekati batas maksimum, serta melakukan kontrol otomatis guna menutup aliran air. Pengujian awal dilakukan melalui simulasi Wokwi untuk memastikan fungsionalitas sistem sebelum implementasi di lapangan. Dengan pendekatan ini, sistem diharapkan mampu menjadi solusi praktis dan efisien dalam mengatasi permasalahan pemborosan air serta meningkatkan kenyamanan dan keamanan dalam penggunaan air di rumah tangga.

B. Indetifikasi Masalah

1. Pemanfaatan air yang tidak terkontrol masih menjadi permasalahan krusial dalam kehidupan sehari-hari, terutama di lingkungan rumah tangga
2. penerapan teknologi IoT dalam monitoring air masih jarang dimanfaatkan secara optimal untuk kebutuhan domestik.

3. Belum tersedianya sistem yang mampu memantau ketinggian air secara real-time, memberikan notifikasi, serta melakukan tindakan otomatis seperti menutup aliran air ketika mencapai batas tertentu, menjadi isu utama yang perlu dipecahkan.

C. Rumusan Masalah

Berdasarkan permasalahan yang terdapat pada latar belakang maka dibuat perumusan masalah yaitu **“Bagaimana Perancangan Sistem Monitoring Dan Kontrol Tinggi Permukaan Air Berbasis Internet Of Thing (IoT) Menggunakan esp32 dan Blynk”**.

D. Batasan Masalah

Untuk memastikan fokus dan keefektifan penelitian, beberapa batasan masalah ditetapkan sebagai berikut:

1. Simulasi dan pengujian sistem monitoring dan kontrol tinggi permukaan air akan dilakukan menggunakan platform Wokwi.
2. Penelitian ini akan menggunakan sensor ultrasonik untuk mengukur ketinggian permukaan air.dan tipe sensor ultrasonik adalah HC-SR04
3. Penambahan komponen pendukung lainnya seperti buzzer sebagai alarm pertanda penuhnya permukaan air dan led (hijau,kuning,dan merah) sebagai tanda level ketingkatan permukaan air, lcd untuk menampilkan status dari tinggi permukaan air, DHT22 sebagai pendekripsi suhu dan kelembapan udara dan servo sebagai untuk menutup keran air secara otomatis saat ketinggian air melebihi batas yang telah ditentukan.
4. Bahasa yang akan digunakan adalah bahasa Arduino,yang kompatibel dengan ESP32 dan mendukung integrasi dengan platform Blynk.

E. Tujuan

1. Tujuan tugas akhir ini adalah untuk merancang dan mengimplementasikan sistem monitoring dan kontrol ketinggian permukaan air berbasis Internet of Things (IoT) dengan menggunakan mikrokontroler ESP32 dan platform Blynk. Sistem ini dirancang untuk memantau ketinggian air secara real-time menggunakan sensor ultrasonik HC-SR04, memberikan notifikasi kepada pengguna melalui aplikasi Blynk

F. Manfaat

1. Dengan adanya sistem monitoring dan kontrol ketinggian air, pemborosan air dapat diminimalkan, sehingga mengurangi tagihan air dan meningkatkan efisiensi penggunaan air di rumah tangga.
2. Memberikan kemudahan bagi pengguna rumah tangga dalam memantau dan mengontrol ketinggian air tanpa harus selalu mengawasi secara manual.
3. Meningkatkan kesadaran pengguna akan pentingnya pengelolaan air yang efisien dan berkelanjutan.
4. Mengintegrasikan teknologi IoT dalam kehidupan sehari-hari, sehingga dapat menjadi bagian dari sistem rumah pintar (smart home) yang lebih canggih dan efisien.

BAB II

HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan Proyek

1. Wokwi

Wokwi adalah platform simulasi elektronik berbasis web yang dirancang untuk membantu para pengembang dan pelajar dalam merancang, menguji, dan memvalidasi proyek-proyek elektronik tanpa memerlukan perangkat keras fisik. Platform ini menawarkan berbagai fitur yang mendukung pengembangan prototipe secara virtual, sehingga memudahkan pengguna dalam melakukan eksperimen dan pengujian.

Fitur-fitur utama pada wokwi :

a. Simulasi real-time

Wokwi memungkinkan simulasi komponen elektronik seperti mikrokontroler, sensor, LED, dan lainnya dalam waktu nyata. Pengguna dapat melihat bagaimana rangkaian elektronik bekerja seolah-olah mereka menggunakan perangkat keras fisik.

b. Berbagai pilihan komponen

Platform ini menyediakan berbagai macam komponen yang dapat digunakan dalam simulasi, termasuk ESP32, Arduino, sensor ultrasonik, LED, buzzer, layar LCD dan motor servo, yang semuanya dapat diintegrasikan untuk membangun sistem monitoring dan kontrol.

c. Pembelajaran dan dokumentasi

Platform ini dilengkapi dengan dokumentasi yang lengkap dan tutorial interaktif yang membantu pengguna, terutama pemula, untuk memahami cara menggunakan berbagai fitur dan komponen yang tersedia.

Manfaat penggunaan wokwi dalam tugas akhir ini adalah

a. Prototipe tanpa perangkat fisik

Dengan menggunakan Wokwi, sistem monitoring dan kontrol tinggi permukaan air dapat dirancang dan diuji secara virtual tanpa memerlukan perangkat keras fisik. Hal ini menghemat biaya dan waktu dalam fase pengembangan awal.

b. Validasi desain

Simulasi pada Wokwi memungkinkan validasi desain dan identifikasi masalah potensial sebelum implementasi fisik, sehingga mengurangi risiko kegagalan dan kesalahan.

c. Efisiensi pengembangan

Penggunaan Wokwi mempercepat proses pengembangan dengan memungkinkan iterasi cepat dan pengujian berbagai konfigurasi dan kode program secara langsung di platform.



Gambar 1. isi dalam platform wokwi

2. Mikrokontroler

sebuah komputer kecil yang dikemas dalam bentuk chip berupa IC (Integrated Circuit) dan dirancang untuk melakukan tugas atau operasi tertentu seperti menerima sinyal input, mengolahnya, kemudian memberikan sinyal output sesuai dengan program yang telah diisikan ke mikrokontroler tersebut. Pada umumnya, sinyal input mikrokontroler berasal dari sensor yang merupakan informasi dari lingkungan sedangkan sinyal output ditujukan kepada aktuator yang dapat melakukan suatu tindakan ke lingkungan. Dengan demikian maka secara sederhana mikrokontroler dapat diasumsikan ibarat sebuah otak yang terdapat pada suatu perangkat dan memiliki kemampuan berinteraksi dengan lingkungan.

Pada dasarnya, pengendali mikro yang dalam bahasa Inggris disebut dengan Microcontroller ini terdiri dari satu atau lebih inti prosesor (CPU), memory (RAM dan ROM), serta perangkat INPUT dan OUTPUT (I/O) yang dapat diprogram. Walaupun mirip dengan komputer namun kecepatan pengolahan data pada mikrokontroler lebih rendah jika dibandingkan dengan komputer atau PC. Kecepatan pengolahan data mikrokontroler umumnya berkisar antara 1 – 16 MHz yang tentu lebih rendah dibandingkan komputer atau PC saat ini yang telah mencapai kecepatan hingga orde GHz. Begitu juga dengan kapasitas memory (RAM dan ROM) yang hanya berkisar pada orde Kbytes.

Meskipun kecepatan pengolahan data dan kapasitas memori yang dimiliki jauh lebih kecil jika dibandingkan dengan komputer atau PC, namun kemampuan mikrokontroler sudah cukup untuk dapat digunakan pada banyak aplikasi terutama karena bentuk ukuran mikrokontroler yang lebih sederhana.

Mikrokontroler sering digunakan pada aplikasi sistem dan perangkat yang tidak terlalu kompleks dan tidak memerlukan komputasi tinggi. Beberapa contoh mikrokontroler seperti mikrokontroler avr, mikrokontroler arduino, mikrokontroler atmega328 dan masih banyak yang lainnya.

Mikrokontroler memiliki berbagai jenis yang diklasifikasikan berdasarkan arsitektur, produsen, dan fitur-fitur yang dimilikinya. Beberapa jenis mikrokontroler yang umum digunakan dalam dunia elektronika dan Internet of Things (IoT) antara lain sebagai berikut:

a. Mikrokontroler AVR

AVR adalah jenis mikrokontroler 8-bit yang dikembangkan oleh Atmel (sekarang menjadi bagian dari Microchip Technology). AVR terkenal karena kemudahan pemrogramannya dan banyak digunakan dalam pendidikan serta proyek-proyek DIY. Salah satu varian paling populer dari keluarga AVR adalah ATmega328P, yang juga menjadi inti dari papan Arduino Uno.

b. . Mikrokontroler PIC

PIC (Peripheral Interface Controller) adalah mikrokontroler buatan Microchip Technology. Mikrokontroler ini juga sangat populer karena fleksibel, memiliki banyak varian dari 8-bit, 16-bit, hingga 32-bit, dan digunakan luas di industri dan pendidikan. PIC dikenal hemat daya dan cocok untuk aplikasi embedded system.

c. . Mikrokontroler ARM

ARM adalah arsitektur prosesor 32-bit (dan sekarang juga 64-bit) yang dilisensikan oleh banyak produsen seperti STMicroelectronics, NXP, dan Texas Instruments. Mikrokontroler berbasis ARM seperti STM32 banyak

digunakan pada aplikasi industri, medis, otomotif, dan IoT karena performa tinggi dan efisiensi daya.

d. Mikrokontroler 8051

Mikrokontroler ini merupakan salah satu yang paling tua dan legendaris dalam dunia embedded system. Diperkenalkan oleh Intel pada tahun 1980-an, 8051 masih digunakan di berbagai produk karena kestabilannya, meski sudah mulai tergantikan oleh jenis yang lebih modern. Mikrokontroler ini memiliki arsitektur 8-bit dan masih dipelajari dalam dunia pendidikan.

e. Mikrokontroler Arduino

Arduino sebenarnya bukan jenis mikrokontroler tersendiri, melainkan platform open-source yang menggunakan berbagai jenis mikrokontroler seperti AVR (ATmega328), ARM (Arduino Due), dan ESP (ESP32/ESP8266). Arduino dirancang agar mudah digunakan bagi pemula maupun profesional, dengan banyak dukungan library dan komunitas.

f. . Mikrokontroler ESP8266 dan ESP32

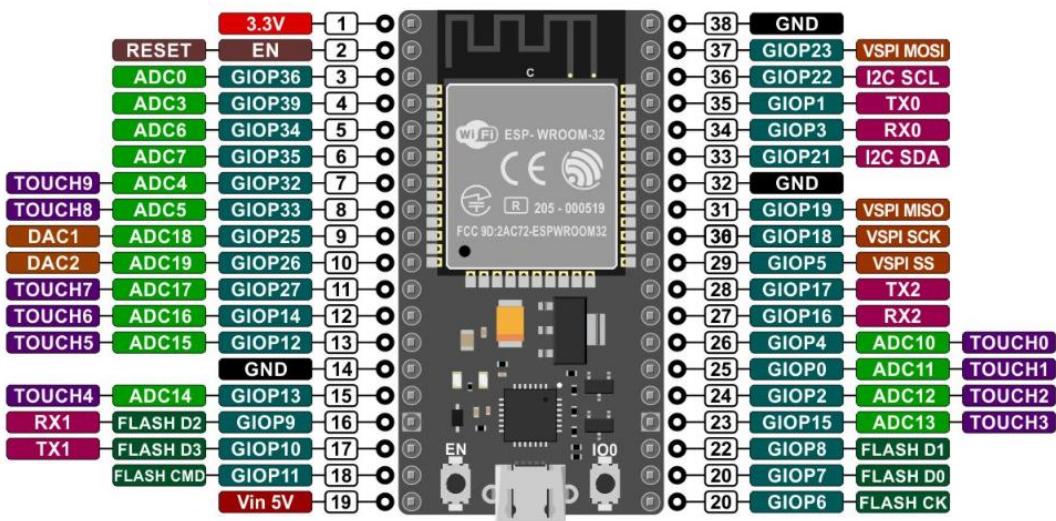
ESP8266 dan ESP32 adalah mikrokontroler buatan Espressif Systems yang populer dalam proyek-proyek IoT karena sudah dilengkapi dengan konektivitas Wi-Fi (dan juga Bluetooth untuk ESP32). ESP8266 adalah mikrokontroler 32-bit dengan harga murah dan kemampuan Wi-Fi. ESP32 adalah versi lebih canggih yang memiliki dual-core processor, Wi-Fi, Bluetooth, lebih banyak pin I/O, dan fitur tambahan seperti ADC, PWM, SPI, I2C, dan sensor internal.

3. ESP32

ESP32 adalah modul mikrokontroler terintegrasi yang memiliki fitur lengkap dan kinerja tinggi. Modul ini merupakan pengembangan dari ESP8266, yang merupakan modul WiFi populer. ESP32 memiliki dua prosesor komputasi, satu prosesor untuk mengelola jaringan WiFi dan Bluetooth, serta satu prosesor lainnya untuk menjalankan aplikasi. Dilengkapi dengan memori RAM yang cukup besar untuk menyimpan data, sebagai otak atau otak pusat sistem early warning banjir. NodeMCU ESP32 bertanggung jawab untuk mengendalikan operasi sistem, mengumpulkan data dari sensor, mengirim data ke server atau sistem pusat melalui koneksi internet, dan menghasilkan peringatan dini jika tinggi air mencapai tingkat kritis. Fitur yang berguna seperti TCP/IP, HTTP, dan FTP. Modul ini juga dilengkapi fitur pemrosesan sinyal analog, dukungan untuk sensor, dan dukungan untuk perangkat masukan/keluaran (I/O) digital. ESP32 juga memiliki dukungan untuk konektivitas Bluetooth. Dapat digunakan untuk mengendalikan perangkat yang terhubung dengan Bluetooth.

ESP32 sangat cocok untuk digunakan dalam proyek-proyek IoT (Internet of things (IoT)). Modul ini mampu menyambungkan perangkat ke jaringan Internet dengan mudah. ESP32 dapat digunakan dalam proyek-proyek yang membutuhkan pemrosesan sinyal analog dan perangkat I/O digital. Modul ini mudah digunakan dan tersedia dalam bentuk modul terpisah atau papan sirkuit terpadu (PCB) yang siap digunakan. ESP32 juga memiliki modul DSP untuk melakukan pemrosesan sinyal secara cepat dan efisien. Terdapat memori sebesar 520 KB untuk menyimpan program dan data yang diperlukan. Mudah digunakan dan tersedia library yang dapat digunakan

untuk memudahkan mengembangkan aplikasi. Modul ini dapat diprogram dengan bahasa pemrograman C atau C++. Secara keseluruhan, ESP32 adalah modul WiFi yang lengkap dan berkualitas untuk berbagai macam proyek IoT. Dengan fitur lengkap dan harga terjangkau, ESP32 pilihan tepat untuk kebutuhan komunikasi wireless. ESP32 memiliki banyak keunggulan seperti kemampuan multitasking yang luar biasa, konsumsi daya rendah, dan harga yang terjangkau. Dengan demikian, ESP32 merupakan pilihan tepat bagi yang ingin membuat proyek IoT dengan biaya yang terjangkau.



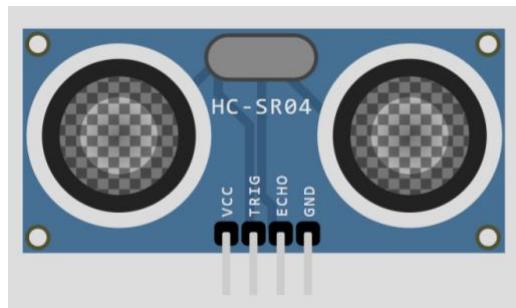
Gambar 3. ESP32

4. Sensor Ultrasonik

adalah sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu. Disebut sebagai sensor ultrasonik karena sensor ini menggunakan gelombang ultrasonik (bunyi ultrasonik). Digunakan untuk mengukur jarak antara sensor ke permukaan air.

Fungsi utamanya adalah untuk mentransmisikan sinyal ultrasonik ke permukaan air dan kemudian menerima pantulan bagi sinyal tersebut untuk kembali, sensor ini menghitung tinggi air dengan akurasi tertentu. Data ini penting untuk mendeteksi potensi tinggi permukaan air bak.

Gelombang ultrasonik adalah gelombang bunyi yang mempunyai frekuensi sangat tinggi yaitu 20.000 Hz. Bunyi ultrasonik tidak dapat di dengar oleh telinga manusia. Bunyi ultrasonik dapat didengar oleh anjing, kucing, kelelawar, dan lumba-lumba. Bunyi ultrasonik nisa merambat melalui zat padat, cair dan gas. Reflektivitas bunyi ultrasonik di permukaan zat padat hampir sama dengan reflektivitas bunyi ultrasonik di permukaan zat cair. Akan tetapi, gelombang bunyi ultrasonik akan diserap oleh tekstil dan busa. Pada sensor ultrasonik, gelombang ultrasonik dibangkitkan melalui sebuah alat yang disebut dengan piezoelektrik dengan frekuensi tertentu. Piezoelektrik ini akan menghasilkan gelombang ultrasonik (umumnya berfrekuensi 40kHz) ketika sebuah osilator diterapkan pada benda tersebut. Secara umum, alat ini akan menembakkan gelombang ultrasonik menuju suatu area atau suatu target. Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima



Gambar 4. sensor ultrasonik HC-SR04

5. DHT22

DHT22 adalah sensor tambahan yang digunakan untuk mengukur kelembaban dan suhu udara. Sensor ini dapat memberikan data tambahan yang berguna dalam sistem monitoring ketinggian air, terutama untuk memahami kondisi lingkungan sekitar yang dapat mempengaruhi ketinggian air. Namun, dalam konteks proyek ini, DHT22 hanya akan digunakan sebagai komponen tambahan untuk menyediakan data lingkungan yang mungkin relevan tetapi tidak krusial bagi fungsi utama sistem.



Gambar 5. DHT22

6. Motor servo

Motor servo merupakan perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol feedback loop tertutup (close loop), sehingga dapat memastikan dan menentukan posisi sudut dari poros output motor. Daya yang dimiliki motor servo bervariasi, mulai beberapa watt sampai ratusan watt.

Motor servo digunakan untuk berbagai keperluan seperti sistem pelacakan, peralatan mesin dan lain sebagainya. Motor servo dibagi menjadi dua, yaitu motor serco AC dan DC. Motor servo DC lebih cocok digunakan pada aplikasi yang lebih kecil, sedangkan motor servo AC cocok digunakan untuk berbagai mesin industri. Hal ini dikarenakan motor servo AC bisa menangani arus yang lebih tinggi atau beban berat. Motor servo AC dibagi menjadi dua tipe, yaitu 2 phase (untuk aplikasi berdaya rendah) dan 3 phase (untuk aplikasi berdaya tinggi).

Motor servo dibangun dengan presisi dan akurasi agar dapat memberikan pengguna kebebasan dalam mengurnya sehingga membuat motor servo sangat terkontrol. Motor servo dikendalikan dengan memberikan Pulse Wide Modulation / PWM melalui kabel kontrol. Durasi "denyut" (pulse) yang diberikan akan menentukan posisi sudut putaran dari poros motor servo. Poros motor servo akan bergerak dan bertahan di posisi yang telah diperintahkan ketika durasi "denyut"nya telah diberikan. Motor servo akan mencoba menahan atau melawan dengan besarnya kekuatan torsi yang dimilikinya apabila ada yang mencoba memutar atau mengubah posisi tersebut. Posisi motor servo tidak akan seterusnya diam saja karena sinyal "denyut"nya harus diulang setiap 20 ms (mili second) untuk menginstruksikan agar tetap pada posisinya.

Motor servo memiliki sistem kontrol loop tertutup yang digunakan untuk mengontrol gerakan dan posisi akhir dari poros motor servo. Secara sederhana, posisi poros output akan dibaca oleh sensor untuk mengetahui posisi poros sudah sesuai yang diinginkan atau belum. Jika belum, kontrol input akan mengirim sinyal kontrol untuk membuat posisi poros tersebut tepat

pada posisi yang diinginkan. Bukan kecepatan yang menentukan posisi dari poros servo melainkan durasi pulsa positif. Nilai pulsa netral yang bergantung pada motor servo menjaga poros motor servo di posisi tengah. Meningkatkan nilai pulsa akan membuat servo berputar searah jarum jam dan pulsa yang lebih pendek akan memutar poros berlawanan arah jarum jam. Pulsa kendali servo biasanya diulang setiap 20ms (tergantung dari motor servo).

Kita harus selalu memberi tahu servo ke mana harus bergerak (posisi). Ada beberapa jenis motor servo, yang kita pakai di sini adalah positional micro servo. Dibandingkan dengan motor DC, servo biasanya memiliki 3 kabel (power, ground, control). Kabel control pada servo berfungsi menarik arus untuk menggerakkan motor.



Gambar 6. motor servo

7. Blnyk

Blynk merupakan sebuah platform Internet of Things (IoT) yang dirancang untuk menyederhanakan proses pengembangan antarmuka pengguna (user interface) serta komunikasi data antara perangkat keras berbasis mikrokontroler dan aplikasi mobile. Platform ini berbasis cloud dan mengedepankan kemudahan integrasi, sehingga pengguna dari berbagai tingkat keahlian—baik pemula maupun profesional—dapat membangun sistem IoT secara efisien dan fleksibel.

Blynk menawarkan arsitektur komunikasi dua arah yang memungkinkan transfer data secara real-time antara perangkat dan pengguna. Arsitektur ini terdiri dari tiga komponen utama: perangkat keras (seperti ESP32, Arduino, NodeMCU), server Blynk yang menjadi pusat lalu lintas data, serta aplikasi mobile Blynk yang berperan sebagai antarmuka visual. Dalam proses komunikasi ini, sensor yang terhubung ke mikrokontroler mengirimkan data melalui jaringan internet ke server Blynk Cloud, yang kemudian ditampilkan dalam bentuk visual interaktif melalui widget pada aplikasi mobile. Sebaliknya, instruksi dari aplikasi dapat diteruskan kembali ke perangkat untuk mengendalikan aktuator seperti relay, LED, atau motor.

Salah satu fitur utama Blynk yang membedakannya dari platform IoT lainnya adalah penggunaan Virtual Pin. Virtual Pin ini bertindak sebagai kanal logis yang memungkinkan pemetaan data sensor atau perintah kontrol tanpa harus terikat pada pin fisik tertentu. Konsep ini memberikan keleluasaan yang besar dalam pengembangan sistem IoT, khususnya dalam proyek yang melibatkan banyak sensor atau logika pemrosesan. Dengan dukungan library Blynk yang kompatibel dengan berbagai jenis mikrokontroler, proses integrasi menjadi lebih efisien dan terstruktur.

Kemudahan lain yang ditawarkan Blynk adalah antarmuka pengguna yang intuitif dan dapat dikustomisasi sepenuhnya melalui sistem drag-and-drop widget. Widget seperti display, tombol, slider, dan grafik memungkinkan pengguna membangun dashboard monitoring yang interaktif tanpa perlu menulis kode antarmuka secara manual. Hal ini sangat menghemat waktu dalam pengembangan proyek, terutama pada tahap prototyping dan pengujian fungsionalitas awal.

Meskipun menawarkan banyak keunggulan, pemanfaatan Blynk tetap memiliki tantangan, terutama ketergantungannya terhadap koneksi internet yang stabil. Dalam konteks penggunaan di daerah dengan infrastruktur jaringan yang terbatas, performa sistem dapat terpengaruh secara signifikan. Selain itu, perubahan pada versi library maupun platform—terutama saat migrasi dari Blynk Legacy ke Blynk IoT terbaru—dapat menyebabkan ketidakcocokan dengan kode lama, sehingga pengguna perlu melakukan penyesuaian ulang.

Secara keseluruhan, Blynk menghadirkan solusi yang sangat kuat untuk mempercepat pengembangan sistem berbasis IoT. Fleksibilitas, kemudahan penggunaan, serta dukungan ekosistem yang luas menjadikan Blynk sebagai salah satu platform yang paling relevan dalam pembelajaran dan pengembangan teknologi berbasis Internet of Things, baik di tingkat akademik maupun industri.



Gambar 7. Blynk

8. LED (*Light Emitting Diode*)

Indikator visual yang digunakan untuk menampilkan tinggi air secara langsung kepada pengguna. LED bekerja berdasarkan prinsip elektroluminesensi, di mana cahaya dihasilkan sebagai respons terhadap arus listrik. Ketika arus listrik melewati persimpangan semikonduktor dioda, elektron dan lubang (holes) bertemu dan melepaskan energi dalam bentuk

foton (cahaya). Warna cahaya yang dihasilkan bergantung pada material semikonduktor yang digunakan.

Dalam konteks sistem monitoring dan kontrol ketinggian air berbasis IoT dengan simulasi Wokwi, LED dapat digunakan sebagai indikator visual untuk menunjukkan berbagai status ketinggian air. Berikut adalah beberapa cara penggunaan LED dalam proyek ini:

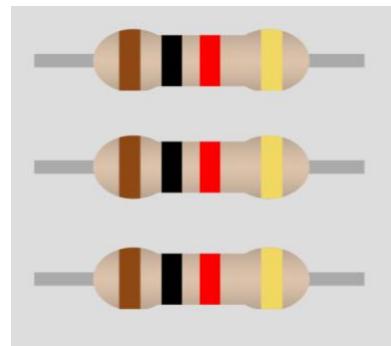
- a. LED Hijau: Menandakan bahwa ketinggian air berada dalam batas aman.
- b. LED Kuning: Menandakan bahwa ketinggian air mendekati batas maksimum yang ditentukan.
- c. LED Merah: Menandakan bahwa ketinggian air telah mencapai atau melampaui batas maksimum, memerlukan tindakan segera.



Gambar 8. LED (*Light Emitting Diode*)

9. Resistor

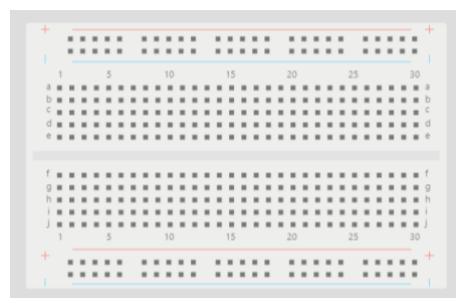
Resistor adalah komponen elektronika yang bersifat menghambat arus listrik. Resistor termasuk dalam komponen pasif karena komponen ini tidak membutuhkan arus listrik untuk bekerja. Resistor terbuat dari material atau bahankarbon dan keramik yangDigunakan untuk mengatur arus listrik dalam rangkaian dan melindungi komponen elektronik lainnya dari potensi overcurrent.



Gambar 9. Resistor

10. Breadboard

Breadboard adalah alat yang digunakan dalam pengembangan prototipe elektronik untuk merakit dan menghubungkan komponen-komponen sirkuit secara sementara. Ini memungkinkan perancang untuk dengan mudah menguji dan mengubah konfigurasi sirkuit tanpa harus melakukan soldering atau perakitan permanen.



Gambar 10. Breadboard

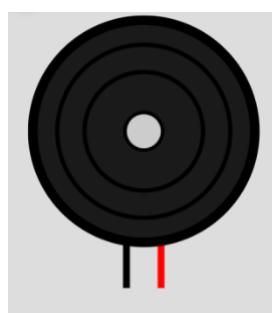
11. Buzzer

Buzzer adalah sebuah komponen elektronika yang dapat menghasilkan getaran suara berupa gelombang bunyi. Buzzer elektronika akan menghasilkan getaran suara ketika diberikan sejumlah tegangan listrik dengan taraf tertentu sesuai dengan spesifikasi bentuk dan ukuran buzzer elektronika itu sendiri. Digunakan untuk menghasilkan suara atau nada yang mengingatkan ketika tinggi air mencapai level bahaya. Buzzer dapat

menghasilkan bunyi peringatan atau alarm yang dapat di dengar dengan jelas oleh orang di sekitarnya.

Pada dasarnya, setiap buzzer elektronika memerlukan input berupa tegangan listrik yang kemudian diubah menjadi getaran suara atau gelombang bunyi yang memiliki frekuensi berkisar antara 1 - 5 KHz. Jenis buzzer elektronika yang sering digunakan dan ditemukan dalam rangkaian adalah buzzer yang berjenis Piezoelectric (Piezoelectric Buzzer). Hal itu karena Piezoelectric Buzzer memiliki berbagai kelebihan diantaranya yaitu lebih murah, relatif lebih ringan dan lebih mudah penggunaannya ketika diaplikasikan dalam rangkaian elektronika. Efek Piezoelektrik (Piezoelectric Effect) ditemukan pertama kali oleh dua orang ilmuwan Fisika pada tahun 1880 bernama Pierre Curie dan Jacques Curie yang berasal dari kebangsaan Perancis. Penemuan tersebut kemudian dikembangkan oleh sebuah perusahaan Jepang menjadi Piezoelectric Buzzer dan mulai populer digunakan pada tahun 1970-an.

Dalam rangkaian elektronika, piezoelectric buzzer dapat digunakan pada tegangan listrik sebesar 6 volt hingga 12 volt dan dengan tipikal arus sebesar 25 mA. Buzzer yang termasuk dalam keluarga Transduser ini sering disebut juga dengan Beeper.

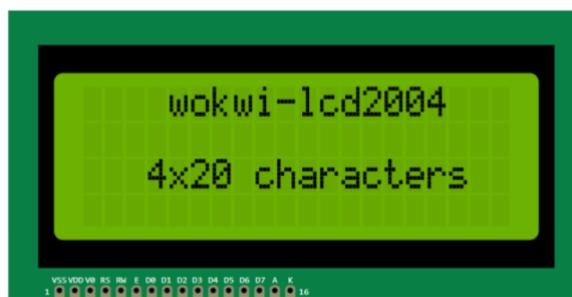


Gambar 11. Buzzer

12. LCD (*Liquid Crystal Display*)

LCD I2C adalah sebuah modul LCD (*Liquid Crystal Display*) yang telah dilengkapi dengan konverter I2C (InterIntegrated Circuit) yang memungkinkan Andamenghubungkannya ke mikrokontroler atau perangkat lainmelalui antarmuka I2C. Modul ini sangat populer dalam proyekproyek elektronika karena kemudahan penggunaannya.

I2C adalah sebuah protokol komunikasi yang memungkinkan perangkat-perangkat untuk berkomunikasi satu sama lain melalui jalur data dan clock yang sama. Penggunaan modul LCD I2C menghilangkan kebutuhan untuk menghubungkan banyak kabel antara mikrokontroler dan LCD, yang membuat instalasi dan penggunaan lebih sederhana. Digunakan sebagai antarmuka visual untuk menampilkan informasi sistem. Ini 27termasuk tampilan tinggi air, status sistem, pesan peringatan, dan informasi penting lainnya. Layar ini memberikan pemahaman visual tentang situasi air bak.

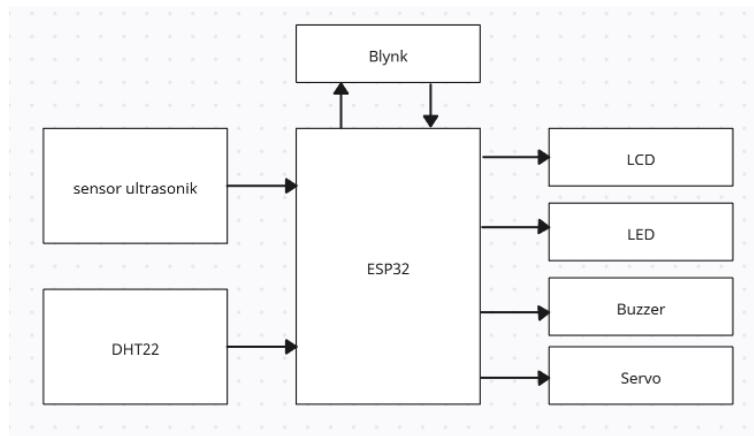


Gambar 12. Lcd2004

B. Desain Proyek Akhir

1. Blok Diagram

Diagram blok digunakan sebagai acuan pembuatan perangkat keras sepaya memudahkan merangkaian suatu rangkain dengan baik.



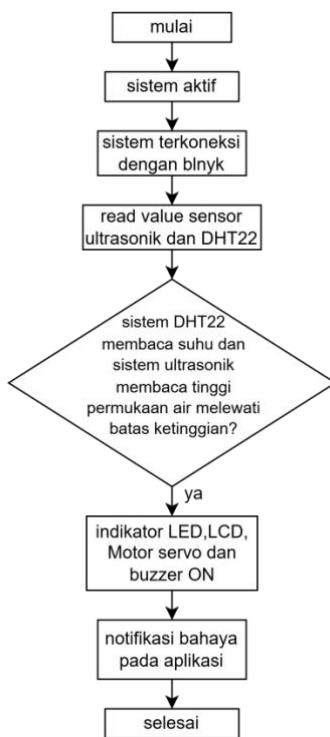
Gambar 13. Diagram Blok

Blog diagram diatas menunjukan **SISTEM MONITORING DAN KONTROL TINGGI PERMUKAAN AIR BERBASIS INTERNET OF THING (IOT) MENGGUNAKAN ESP32 DAN BLNYK** Pada diagram blok terdapat dua input yaitu sensor ultrasonic HC-SR04 dan sensor DHT22 yang nantinya akan mengirim dan menerima data yang diproses oleh mikrokontroler ESP32. ESP32 berfungsi sebagai pengontrol utama dari alat yang dibuat. Sensor DHT22 dan sensor HC-SR04 berfungsi sebagai pendekripsi kelembaban suhu dan jarak. Selanjutnya BUZZER sebagai alarm, LED sebagai indikator visual, LCD untuk menampilkan informasi dan motor servo berfungsi sebagai indikator penutup keran air secara otomatis ketika ketinggian air melebihi batas yang ditentukan.

Saat data sensor diperoleh, ESP32 akan mengirimkan informasi ke platform Blynk melalui koneksi WiFi. Di platform Blynk, data tersebut akan ditampilkan secara real-time pada aplikasi Blynk di perangkat mobile pengguna. Aplikasi Blynk juga memungkinkan pengguna untuk memantau kondisi ketinggian air dan menerima notifikasi bila ketinggian air telah mencapai batas tertentu.

2. Diagram Aliran Proses (Flowchart)

Flowchart merupakan gambar atau bagian yang memperlihatkan urutan langkah-langkah dari suatu program dan hubungan antar proses beserta pernyataan pada sistem yang akan dirancang. Flowchart sistem dapat digambarkan sebagai berikut:



Gambar 14. flowchart sistem monitoring ketinggian permukaan air pada simulasi

wokwi

Penjelasan flowchart pada gambar 14 yaitu proses kerja sistem monitoring tinggi permukaan air pada Kamar Mandi. Berikut penjelasan dari flowchart pada gambar diatas:

- Sensor ultrasonic mengukur jarak antara sensor ke permukaan air dengan memancarkan gelombang ultrasonik ke arah air, jarak yang dibaca oleh sensor akan dikelompokkan menjadi 3 kondisi, yaitu : aman, waspada, bahaya. Dan DHT22 mengukur suhu ruangan dikamar mandi
- Rentang jarak 30-20cm untuk kondisi aman, pada jarak 20- 10cm untuk kondisi waspada dan jarak -0cm untuk kondisi bahaya
- Apabila dalam kondisi “bahaya ”, notifikasi akan dikirimkan ke Blynk, LCD akan menampilkan jarak yang dibaca oleh sensor ditaris pertama dan di baris kedua

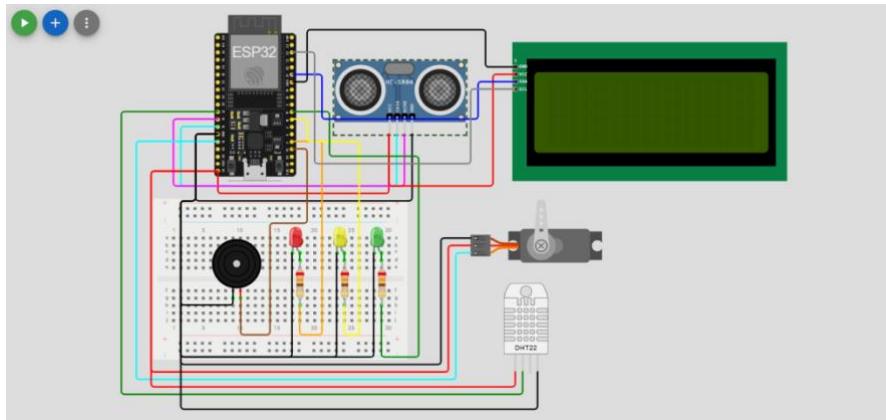
menampilkan tulisan kondisi “bahaya”, LED berwarna merah menyala, motor servo menutup, dan buzzer akan berbunyi.

d) Apabila dalam kondisi “waspada”, notifikasi akan dikirimkan ke Blynk, LCD akan menampilkan jarak yang dibaca oleh sensor ditaris pertama dan di baris kedua menampilkan tulisan kondisi “waspada”, LED berwarna kuning menyala dan motor servo setengah buka.

e) Apabila dalam kondisi “Aman”, notifikasi akan dikirimkan ke Blynk, LCD akan menampilkan jarak yang dibaca oleh sensor ditaris pertama dan ditaris kedua menampilkan tulisan kondisi “Aman”, LED berwarna hijau menyala dan motor servo membuka.

3. Hasil desain proyek akhir

Desain proyek akhir mengintegrasikan semua komponen utama pada satu papan ESP32, diuji melalui simulasi Wokwi untuk memastikan fungsi real-time dan kesiapan implementasi nyata. Sensor ultrasonik HC-SR04 untuk mengukur ketinggian air, sementara sensor DHT22 untuk memantau suhu dan kelembapan. Output audio-visual terdiri dari buzzer dan tiga buah LED (hijau , kuning , merah) yang menandakan status “aman”, “waspada”, dan “bahaya”. Informasi ketinggian dan status tampil di LCD I2C , dan servo motor yang secara otomatis menutup kran saat level air melewati batas kritis. Semua sinyal dialirkan melalui resistor yang sesuai untuk melindungi komponen. Loop utama pada ESP32 membaca data sensor, menentukan status, memperbarui LCD, mengubah warna LED, mengaktifkan buzzer, menggerakkan servo, dan mengirim notifikasi lewat Blynk—sehingga desain ini telah memenuhi tujuan sistem monitoring dan kontrol ketinggian air berbasis IoT secara andal dan praktis..



Gambar 15. Hasil desain sistem monitoring ketinggian permukaan air pada simulasi wokwi

Pastikan bahwa kode program yang digunakan dalam desain telah dikembangkan dengan baik dan sesuai dengan bahasa Arduino. Kode program harus mampu membaca data dari sensor, mengontrol perilaku sistem berdasarkan data tersebut, dan mengendalikan output seperti buzzer, LED, LCD dan motor servo

```

sketch.ino    diagram.json    libraries.txt    Library Manager
1  #define BLYNK_TEMPLATE_ID "TMPL6ZxZyehGJ"
2  #define BLYNK_TEMPLATE_NAME "esp32"
3  #define BLYNK_AUTH_TOKEN "jxtmFHUKZ5UbbKLflaPpkq7EVPAZe8R"
4
5  #define BLYNK_PRINT Serial
6
7  #include <WiFi.h>
8  #include <BlynkSimpleEsp32.h>
9  #include <Wire.h>
10 #include <LiquidCrystal_I2C.h>
11 #include <ESP32Servo.h>
12 #include <DHT.h>
13
14 char auth[] = BLYNK_AUTH_TOKEN;
15 char ssid[] = "Wokwi-GUEST";
16 char pass[] = "";
17
18 // Pin definitions
19 #define TRIGGER_PIN 12
20 #define ECHO_PIN 14
21 #define BUZZER_PIN 15
22 #define RED_LED_PIN 2
23 #define YELLOW_LED_PIN 16
24 #define GREEN_LED_PIN 17
25 #define SERVO_PIN 13
26 #define DHTPIN 27
27 #define DHTTYPE DHT22
28
29 // Initialize LCD 2004 (20x4)

```

```

sketch.ino    diagram.json    libraries.txt    Library Manager
30 LiquidCrystal_I2C lcd(0x27, 20, 4);
31 Servo myservo;
32 DHT dht(DHTPIN, DHTTYPE);
33 BlynkTimer timer;
34
35 int currentServoAngle = 0; // Variabel untuk menyimpan posisi servo saat ini
36
37 void setup() {
38   Serial.begin(115200);
39
40   // Initialize LCD 2004
41   lcd.init();
42   lcd.backlight();
43   lcd.setCursor(0, 0);
44   lcd.print("Proyek Tugas Akhir");
45   lcd.setCursor(0, 1);
46   lcd.print("Deteksi Permukaan Air");
47   lcd.setCursor(0, 2);
48   lcd.print("By: Fajar Ramadhan");
49   lcd.setCursor(0, 3);
50   lcd.print("nim : 21066010");
51   delay(2000);
52   lcd.clear();
53
54   // Initialize servo dengan rentang default (0-180)
55   ESP32PWM::allocateTimer(0);
56   ESP32PWM::allocateTimer(1);
57   ESP32PWM::allocateTimer(2);
58   ESP32PWM::allocateTimer(3);

```

```

sketch.ino    diagram.json    libraries.txt    Library Manager
37 void setup() {
38   myservo.setPeriodHertz(50); // Standard 50hz servo
39   myservo.attach(SERVO_PIN, 500, 2400); // Attach servo with min/max pulse width
40   myservo.write(0); // Set servo ke posisi awal
41   currentServoAngle = 0;
42
43   // Initialize DHT sensor
44   dht.begin();
45
46   // Initialize LEDs and buzzer
47   pinMode(BUZZER_PIN, OUTPUT);
48   pinMode(RED_LED_PIN, OUTPUT);
49   pinMode(YELLOW_LED_PIN, OUTPUT);
50   pinMode(GREEN_LED_PIN, OUTPUT);
51
52   // Initialize ultrasonic
53   pinMode(TRIGGER_PIN, OUTPUT);
54   pinMode(ECHO_PIN, INPUT);
55
56   // Turn off all LEDs and buzzer at startup
57   digitalWrite(BUZZER_PIN, LOW);
58   digitalWrite(RED_LED_PIN, LOW);
59   digitalWrite(YELLOW_LED_PIN, LOW);
60   digitalWrite(GREEN_LED_PIN, LOW);
61
62   // Connect to Blynk
63   Blynk.begin(auth, ssid, pass);
64 }
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86

```

WOKWI

```

sketch.ino  SAVE  SHARE  fajarrr
87 void moveServoSmoothly(int targetAngle) {
88     // Bergerak secara bertahap ke posisi target
89     while (currentServoAngle != targetAngle) {
90         if (currentServoAngle < targetAngle) {
91             currentServoAngle++;
92         } else {
93             currentServoAngle--;
94         }
95         myservo.write(currentServoAngle);
96         delay(15); // Delay untuk gerakan lebih halus
97     }
98 }
99
100 void loop() {
101     Blynk.run();
102
103     // Read DHT22 sensor
104     float suhu = dht.readTemperature();
105     float kelembapan = dht.readHumidity();
106
107     // Read ultrasonic sensor
108     long duration, distance;
109     digitalWrite(TRIGGER_PIN, LOW);
110     delayMicroseconds(2);
111     digitalWrite(TRIGGER_PIN, HIGH);
112     delayMicroseconds(10);
113     digitalWrite(TRIGGER_PIN, LOW);
114     duration = pulseIn(ECHO_PIN, HIGH);
115     distance = (duration / 2) / 29.1;

```

WOKWI

```

sketch.ino  SAVE  SHARE  fajarrr
100 void loop() {
101
102     // Send data to Blynk
103     Blynk.virtualWrite(V1, distance);
104     Blynk.virtualWrite(V2, suhu);
105     Blynk.virtualWrite(V3, kelembapan);
106
107     // Display data on LCD 2004
108     lcd.setCursor(0, 0);
109     lcd.print("Jarak: ");
110     lcd.print(distance);
111     lcd.print(" cm      ");
112
113     lcd.setCursor(0, 1);
114     lcd.print("Status: ");
115     if (distance <= 20) {
116         lcd.print("BAHAYA!      ");
117     } else if (distance > 20 && distance <= 50) {
118         lcd.print("WASPADA      ");
119     } else {
120         lcd.print("AMAN        ");
121     }
122
123     lcd.setCursor(0, 2);
124     lcd.print("Suhu: ");
125     lcd.print(suhu, 1);
126     lcd.print(" C      ");
127
128     lcd.setCursor(0, 3);
129     lcd.print("Kelembapan: ");
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172

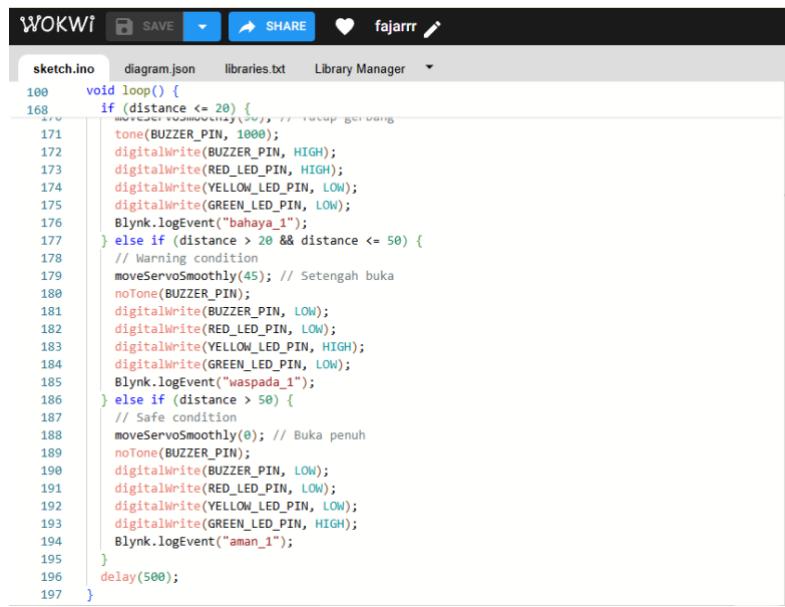
```

WOKWI

```

sketch.ino  SAVE  SHARE  fajarrr
100 void loop() {
101
102     lcd.print(kelembapan, 0);
103     lcd.print("%      ");
104
105     // Log data to serial monitor
106     Serial.print("Distance: ");
107     Serial.print(distance);
108     Serial.print(" cm | Status: ");
109     if (distance <= 20) {
110         Serial.print("BAHAYA");
111     } else if (distance > 20 && distance <= 50) {
112         Serial.print("WASPADA");
113     } else {
114         Serial.print("AMAN");
115     }
116     Serial.print(" | Suhu: ");
117     Serial.print(suhu);
118     Serial.print(" C | Kelembapan: ");
119     Serial.print(kelembapan);
120     Serial.print(" % | Servo: ");
121     Serial.print(currentServoAngle);
122     Serial.println("°");
123
124     // Check for flood condition and update LEDs, buzzer, and servo
125     if (distance <= 20) {
126         // Flood condition
127         moveServoSmoothly(90); // Tutup gerbang
128         tone(BUZZER_PIN, 1000);
129         digitalWrite(BUZZER_PIN, HIGH);
130     }
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172

```



```

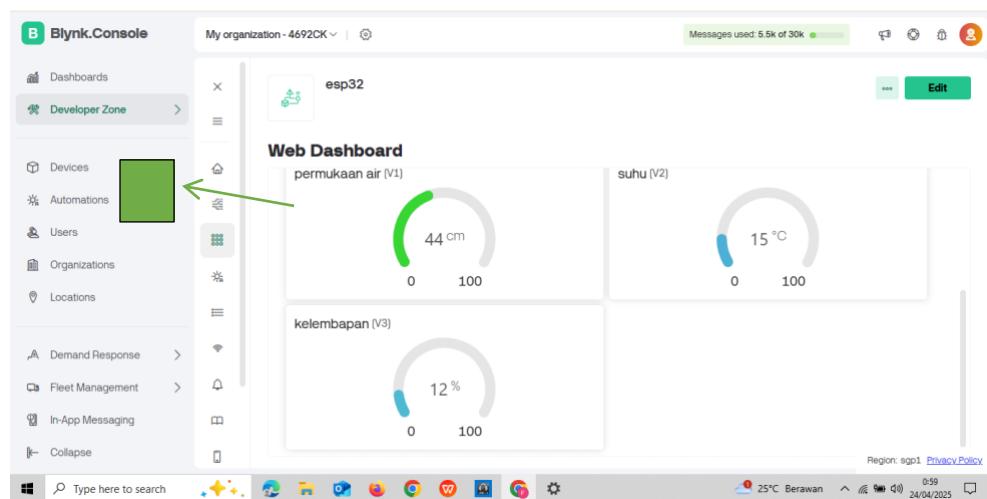
WOKWI
SAVE SHARE fajarr
sketch.ino diagram.json libraries.txt Library Manager
100 void loop() {
101   if (distance <= 20) {
102     moveServoSmoothly(180); // Banyak banting
103     tone(BUZZER_PIN, 1000);
104     digitalWrite(BUZZER_PIN, HIGH);
105     digitalWrite(RED_LED_PIN, HIGH);
106     digitalWrite(YELLOW_LED_PIN, LOW);
107     digitalWrite(GREEN_LED_PIN, LOW);
108     Blynk.logEvent("bahaya_1");
109   } else if (distance > 20 && distance <= 50) {
110     // Warning condition
111     moveServoSmoothly(45); // Setengah buka
112     noTone(BUZZER_PIN);
113     digitalWrite(BUZZER_PIN, LOW);
114     digitalWrite(RED_LED_PIN, LOW);
115     digitalWrite(YELLOW_LED_PIN, HIGH);
116     digitalWrite(GREEN_LED_PIN, LOW);
117     Blynk.logEvent("waspadा_1");
118   } else if (distance > 50) {
119     // Safe condition
120     moveServoSmoothly(0); // Buka penuh
121     noTone(BUZZER_PIN);
122     digitalWrite(BUZZER_PIN, LOW);
123     digitalWrite(RED_LED_PIN, LOW);
124     digitalWrite(YELLOW_LED_PIN, LOW);
125     digitalWrite(GREEN_LED_PIN, HIGH);
126     Blynk.logEvent("aman_1");
127   }
128   delay(500);
129 }

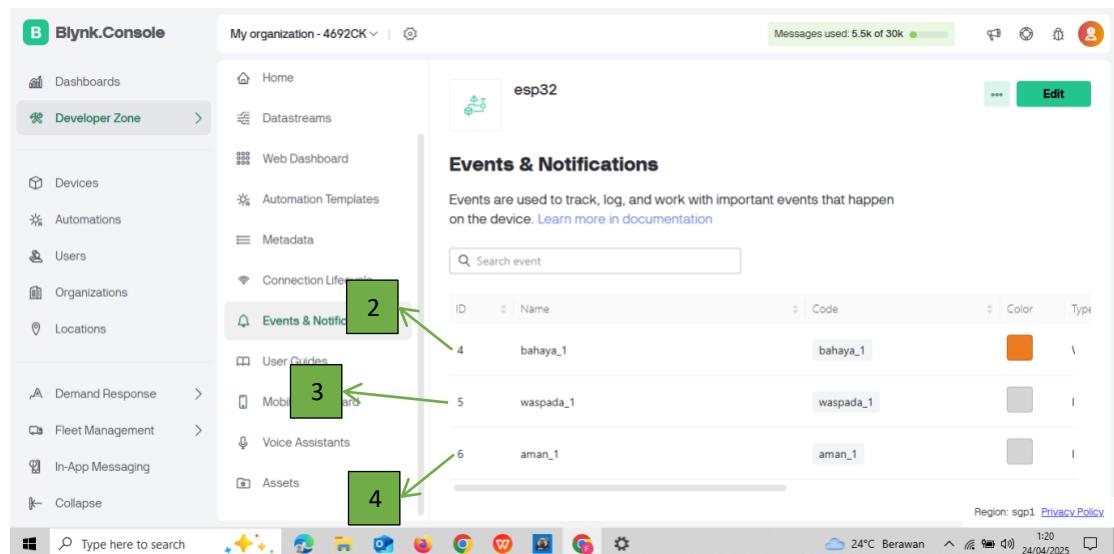
```

Gambar 16. Kode program pada simulasi wokwi

4. Perancangan Tampilan Web Dashboard Aplikasi Blynk

Blynk merupakan sebuah layanan aplikasi yang digunakan untuk mengontrol mikrokontroler dari jaringan internet. Aplikasi yang disediakan oleh blynk sendiri masih butuh disusun sesuai dengan kebutuhan. Berikut rancangan aplikasi Blynk untuk proyek akhir yang penulis buat





Gambar 17. Tampilan Web Dashboard Aplikasi Blynk

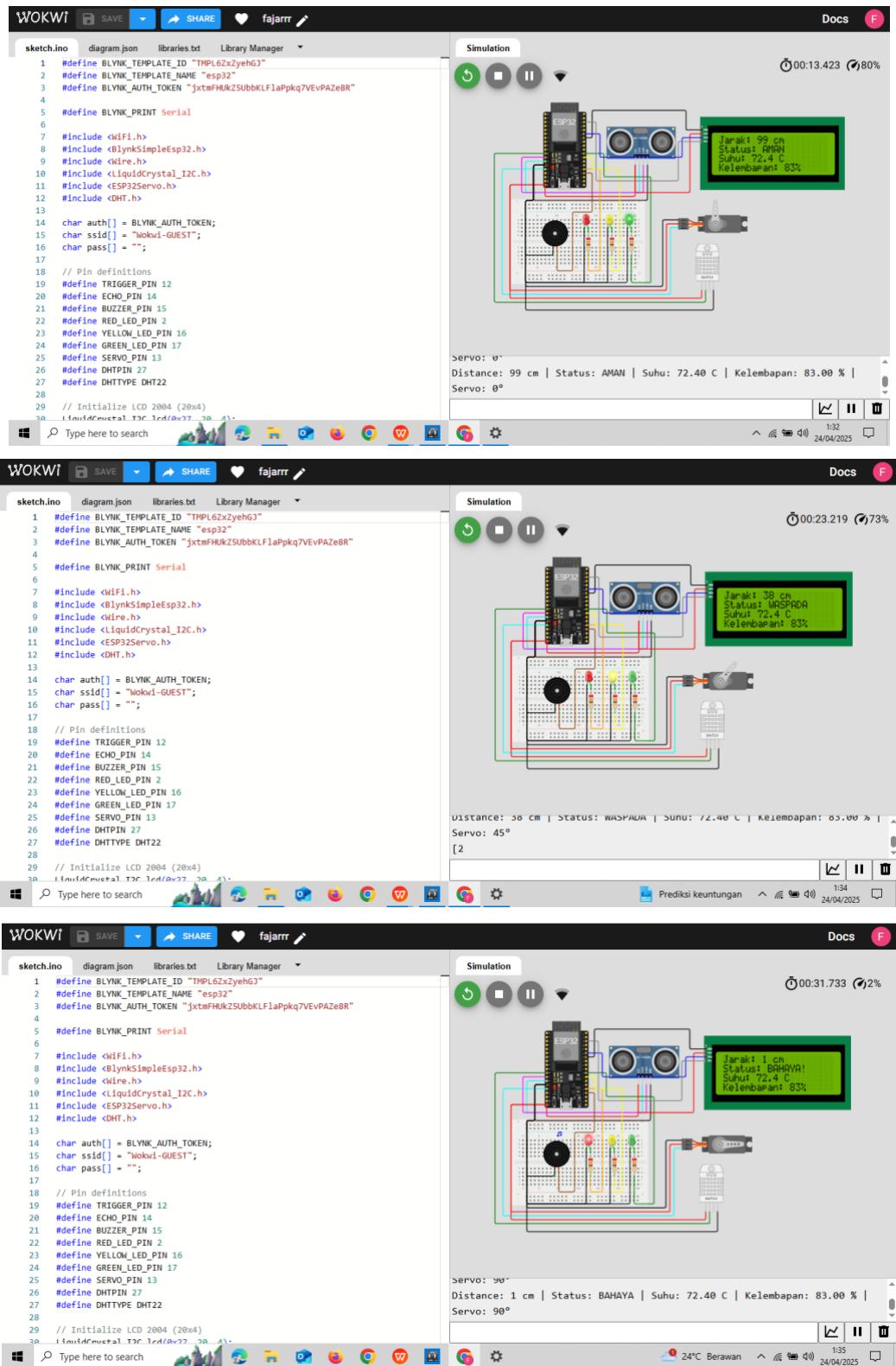
Gambar 17 adalah tampilan web dashboard blynk yang dirancang, berikut penjelasan pada gambar 14 :

1. Melihat pengukuran jarak sensor ke permukaan air dengan memancarkan gelombang ultrasonik ke arah air dengan pin V1 berbentuk Centimeter dan DHT22 yang mengukur suhu dan kelembapan dengan pin V2 dan V3
2. Memberikan notifikasi “Bahaya” bahwasannya air sudah naik.
3. Memberikan notifikasi “Waspada ” bahwasannya air mulai naik
4. Memberikan notifikasi “Aman” bahwsannya kondisi air masih aman.

C. Deskripsi Hasil

Hasil simulasi sistem monitoring dan kontrol tinggi permukaan air pada platform wokwi menunjukan bahwa sensor ultrasonik HC-SR04 dan DHT22 merespon cepat terhadap tinggi permukaan air,suhu dan kelembapan dengan. Akurasi yang tinggi

1. Hasil gambar simulasi pada wokwi



Gambar 18. Hasil simulasi pada wokwi

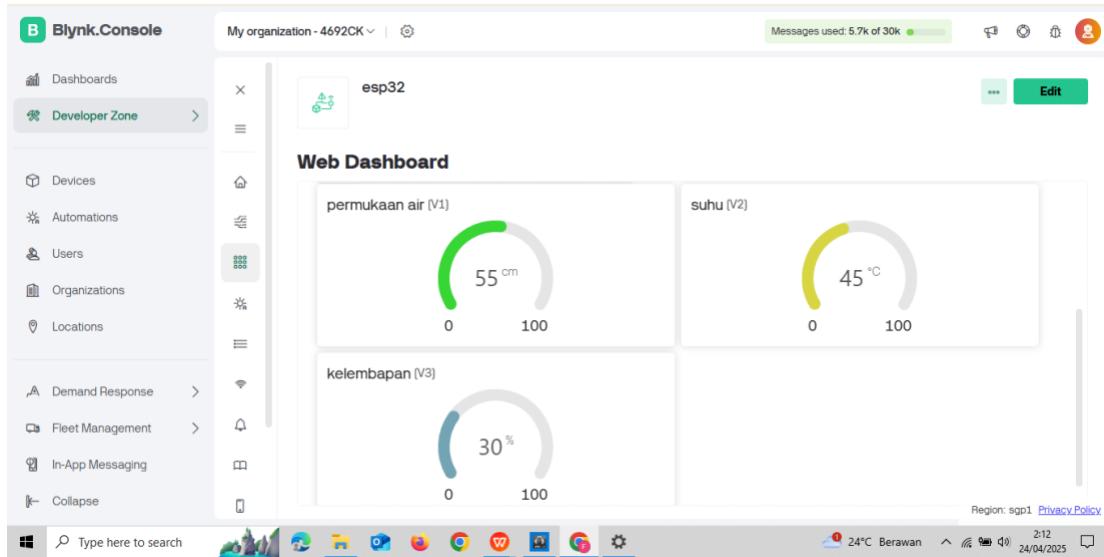
Pada hasil simulasi di platform Wokwi, sistem monitoring dan kontrol tinggi permukaan air yang dirancang menunjukkan kinerja yang baik, dengan respon yang cepat terhadap perubahan kondisi air. Sensor ultrasonik HC-SR04 berfungsi untuk mengukur jarak antara permukaan air dengan sensor, sementara sensor DHT22 digunakan untuk memantau suhu dan kelembapan ruangan. Semua data dari kedua sensor diproses oleh mikrokontroler ESP32 yang menjadi otak sistem ini.

Setelah data diproses, sistem akan menyesuaikan output berdasarkan kondisi ketinggian air. Jika ketinggian air berada pada jarak 30 hingga 20 cm, sistem menganggap kondisi aman. Dalam kondisi ini, LED hijau akan menyala, buzzer tidak aktif, dan servo motor membuka keran sepenuhnya. Jika ketinggian air mendekati batas, yaitu antara 20 hingga 10 cm, sistem masuk ke kondisi waspada. LED kuning menyala, servo motor hanya membuka sebagian, dan informasi kondisi ini ditampilkan di LCD serta dikirim ke aplikasi Blynk. Apabila ketinggian air sudah di bawah 10 cm, sistem mengenali kondisi bahaya. LED merah menyala, buzzer berbunyi sebagai alarm, servo motor menutup keran sepenuhnya, dan LCD menampilkan peringatan bahaya, yang juga langsung dikirimkan ke aplikasi Blynk.

Informasi dari sensor ditampilkan secara real-time di LCD I2C empat baris yang menampilkan suhu, kelembapan, jarak air, dan status sistem. Tampilan ini memberi kemudahan pengguna untuk membaca kondisi langsung di lokasi. Di sisi lain, aplikasi Blynk memudahkan pengguna untuk melakukan pemantauan jarak jauh dengan koneksi WiFi yang memungkinkan pengiriman notifikasi secara otomatis sesuai kondisi sistem.

Secara keseluruhan, simulasi membuktikan bahwa sistem ini mampu melakukan monitoring dan kontrol ketinggian air dengan akurat dan efektif. Semua komponen bekerja sesuai fungsi masing-masing, menciptakan sistem yang dapat digunakan untuk kebutuhan pemantauan air secara otomatis dan berbasis Internet of Things.

2. Tampilan Pada Bylnk



Gambar 19. Hasil Tampilan Blynk

Gambar 19 merupakan gambar sistem monitoring tinggi permukaan air yang telah dikontrol lewat Blynk

3. Tampilan Notifikasi Bylnk



Gambar 20. Tampilan Notifikasi Blynk

Gambar 20 merupakan gambar notifikasi sistem monitoring dan kontrol tinggi permukaan air yang telah dikontrol lewat Blynk. Proyek mencapai tujuan utama, yaitu menciptakan sistem monitoring dan kontrol tinggi permukaan air berbasis IoT yang dapat memberikan peringatan dini dan membantu mengurangi risiko kelebihan air. Sistem mencapai tingkat akurasi yang memadai dalam pengukuran tinggi air. Sistem berhasil memberikan peringatan dini secara efektif, memungkinkan respons cepat terhadap tinggi permukaan air.

Tabel 1. Hasil Pengujian Alat Keseluruhan

Level	Status	jarak	LED	Keterangan
1	Aman	30 <20cm	Hijau	LED hijau hidup dan motor servo masih terbuka dan muncul notifikasi aman
2	waspada	20 < 10cm	Kuning	LED kuning hidup dan motor servo tertutup setengah dan muncul notifikasi waspada
3	Bahaya	10 < 0cm	Merah	LED merah hidup, buzzer berbunyi dan motor servo tertutup rapat dan muncul notifikasi bahaya

D. Pembahasan Hasil

Pada simulasi yang dilakukan di platform Wokwi, sistem monitoring dan kontrol tinggi permukaan air yang dibangun dengan mikrokontroler ESP32 menunjukkan performa yang responsif dan stabil. Sensor ultrasonik HC-SR04 mampu mendeteksi perubahan ketinggian air secara cepat dan akurat. Sensor ini mengukur jarak permukaan air dengan mengirimkan gelombang ultrasonik dan mengukur waktu pantulannya, yang kemudian diproses oleh ESP32 untuk menentukan status kondisi air.

Selain itu, sensor DHT22 juga memberikan data suhu dan kelembapan secara real-time. Kedua sensor ini bekerja secara bersamaan dan saling melengkapi, sehingga sistem dapat memantau kondisi lingkungan dan ketinggian air dengan lebih menyeluruh.

ESP32 bertindak sebagai pusat kontrol yang tidak hanya memproses data sensor, tetapi juga mengaktifkan perangkat output seperti buzzer, LED, motor servo, dan LCD. LCD digunakan untuk menampilkan informasi secara lokal, sedangkan koneksi WiFi yang terhubung ke platform Blynk memungkinkan pemantauan jarak jauh melalui aplikasi mobile. Fitur ini sangat mendukung konsep Internet of Things (IoT) yang menjadi dasar rancangan sistem.

Meskipun simulasi menunjukkan sistem berjalan sesuai harapan, penting untuk mempertimbangkan tantangan saat implementasi di dunia nyata, seperti kestabilan koneksi WiFi, kondisi lingkungan yang bisa memengaruhi akurasi sensor, serta perawatan komponen seperti sensor ultrasonik, motor servo, dan modul ESP32 agar sistem tetap berfungsi optimal dalam jangka panjang.

Dengan kombinasi perangkat keras dan pemrograman yang tepat, sistem ini terbukti mampu memberikan solusi efektif dalam pemantauan dan pengendalian ketinggian air secara otomatis, serta dapat diandalkan sebagai sistem peringatan dini berbasis IoT.

BAB III

SIMPULAN DAN REKOMENDASI

A. Simpulan

Penelitian ini berhasil merancang dan menyimulasikan sistem monitoring dan kontrol tinggi permukaan air berbasis IoT menggunakan mikrokontroler ESP32, sensor ultrasonik HC-SR04, serta platform Blynk. Sistem ini mampu memantau ketinggian air secara real-time, memberikan notifikasi sesuai status air (aman, waspada, bahaya), dan mengaktifkan kontrol otomatis seperti buzzer, LED, serta penutupan keran menggunakan servo motor saat air melebihi batas.

Dari hasil simulasi di platform Wokwi, sistem menunjukkan kinerja yang efektif dan responsif terhadap perubahan ketinggian air. Dengan demikian, sistem ini tidak hanya menjawab rumusan masalah dan mencapai tujuan penelitian, tetapi juga menawarkan solusi praktis dalam pemanfaatan teknologi IoT untuk efisiensi air rumah tangga, serta mendukung penerapan smart home sederhana dan ramah lingkungan..

B. Rekomendasi

1. Pemeliharaan rutin

Meskipun sistem telah dirancang agar dapat beroperasi secara otomatis, pemeliharaan berkala terhadap komponen sangat disarankan. Sensor ultrasonik HC-SR04, sensor DHT22, motor servo, dan modul ESP32 rentan terhadap debu, kelembapan tinggi, atau gangguan eksternal lainnya yang bisa memengaruhi akurasi dan respons sistem. Oleh karena itu, perlu dilakukan pembersihan, kalibrasi, dan pengecekan koneksi secara berkala agar sistem tetap stabil dan andal dalam jangka panjang.

2. Pengujian di Lingkungan Nyata

Simulasi di Wokwi sangat membantu dalam tahap pengembangan awal, namun tetap perlu dilakukan pengujian langsung di lapangan. Hal ini bertujuan untuk mengetahui bagaimana sistem beradaptasi dengan kondisi lingkungan nyata, seperti perubahan suhu, kelembapan ekstrem, atau gangguan sinyal WiFi yang bisa memengaruhi performa sistem

3. Penguatan Proteksi Komponen

Dalam penerapan nyata, penggunaan casing tahan air atau pelindung fisik untuk sensor dan ESP32 sangat dianjurkan, terutama jika sistem ditempatkan di area lembap seperti kamar mandi atau bak penampungan terbuka. Hal ini untuk menghindari kerusakan akibat cipratan air atau korsleting...

DAFTAR PUSTAKA

- Prasetyo, R., Ujianto, N. T., Budiraharjo, E., & Hapsari, Y. (2023). Pengembangan Sistem Pendekripsi Hujan Berbasis Internet of Things (IoT) dengan Simulasi Wokwi. *Engineering: Jurnal Bidang Teknik*, 14(2), 93-99.
- Suparman, S., Suhartanto, E., & Shina, Y. I. (2022). Perancangan Alat Otomatisasi Sistem Monitoring Dan Kontrol Tinggi Permukaan Air Sungai Sebagai Peringatan Dini Terjadinya Banjir. *Jurnal Teknologi*, 15(1), 87-95
- Amin, A. (2018). Monitoring water level control berbasis arduino uno menggunakan lcd lm016L. *EEICT (Electric, Electronic, Instrumentation, Control, Telecommunication)*, 1(1).
- Purwanto, Y., & Handayani, T. (2011). Rancang Bangun Sistem Pengendalian Tinggi Permukaan Air Dan Suhu Cairan Berbasis PLC SCADA. *Jurnal Teknik Elektro*, 3(1), 6
- Fikri, R., Lapanporo, B. P., & Jumarang, M. I. (2015). Rancang Bangun Sistem Monitoring Ketinggian Permukaan Air Menggunakan Mikrokontroler ATMEGA328P Berbasis Web Service. *POSITRON*, 5(2)
- Sulthoni, A. (2017). Monitoring Tinggi Permukaan Air Pada Tandon Menggunakan Sensor Ultrasonik Sr04 Berbasis Arduino Uno Dengan Komunikasi Wireless Nrf24l01 (Doctoral dissertation, Universitas Brawijaya)
- Jafaruddin, J., Muhamimin, M., & Jamaluddin, J. (2016). Rancang bangun sistem pengontrolan dan monitoring level permukaan air secara jarak jauh. *Jurnal Litek: Jurnal Listrik Telekomunikasi Elektronika*, 13(2), 73-78.....

LAMPIRAN

LAMPIRAN 1

```

#define BLYNK_TEMPLATE_ID "TMPL6ZxZyehGJ"
#define BLYNK_TEMPLATE_NAME "esp32"
#define BLYNK_AUTH_TOKEN "jxtmFHUkZ5UbbKLFlaPpkq7VEvPAZe8R"

#define BLYNK_PRINT Serial

#include<WiFi.h>
#include<BlynkSimpleEsp32.h>
#include<Wire.h>
#include<LiquidCrystal_I2C.h>
#include<ESP32Servo.h>
#include<DHT.h>

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Wokwi-GUEST";
char pass[] = "";

// Pin definitions
#define TRIGGER_PIN 12
#define ECHO_PIN 14
#define BUZZER_PIN 15
#define RED_LED_PIN 2
#define YELLOW_LED_PIN 16
#define GREEN_LED_PIN 17
#define SERVO_PIN 13
#define DHTPIN 27
#define DHTTYPE DHT22

// Initialize LCD 2004 (20x4)
LiquidCrystal_I2C lcd(0x27, 20, 4);
Servo myservo;
DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;

int currentServoAngle = 0; // Variabel untuk menyimpan posisi servo saat ini

void setup() {
  Serial.begin(115200);

  // Initialize LCD 2004
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Proyek Tugas Akhir");
}

```

```

lcd.setCursor(0, 1);
lcd.print("Deteksi Permukaan Air");
lcd.setCursor(0, 2);
lcd.print("By: Fajar Ramadhan");
lcd.setCursor(0, 3);
lcd.print("nim : 21066010");
delay(2000);
lcd.clear();

// Initialize servo dengan rentang default (0-180)
ESP32PWM::allocateTimer(0);
ESP32PWM::allocateTimer(1);
ESP32PWM::allocateTimer(2);
ESP32PWM::allocateTimer(3);
myservo.setPeriodHertz(50);      // Standard 50hz servo
myservo.attach(SERVO_PIN, 500, 2400); // Attach servo with min/max
pulse width
myservo.write(0); // Set servo ke posisi awal
currentServoAngle = 0;

// Initialize DHT sensor
dht.begin();

// Initialize LEDs and buzzer
pinMode(BUZZER_PIN, OUTPUT);
pinMode(RED_LED_PIN, OUTPUT);
pinMode(YELLOW_LED_PIN, OUTPUT);
pinMode(GREEN_LED_PIN, OUTPUT);

// Initialize ultrasonic
pinMode(TRIGGER_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);

// Turn off all LEDs and buzzer at startup
digitalWrite(BUZZER_PIN, LOW);
digitalWrite(RED_LED_PIN, LOW);
digitalWrite(YELLOW_LED_PIN, LOW);
digitalWrite(GREEN_LED_PIN, LOW);

// Connect to Blynk
Blynk.begin(auth, ssid, pass);
}

void moveServoSmoothly(int targetAngle) {
// Bergerak secara bertahap ke posisi target
while (currentServoAngle != targetAngle) {
if (currentServoAngle < targetAngle) {
currentServoAngle++;
} else {
}
}
}

```

```
    currentServoAngle--;
}
myservo.write(currentServoAngle);
delay(15); // Delay untuk gerakan lebih halus
}
}

void loop() {
Blynk.run();

// Read DHT22 sensor
float suhu = dht.readTemperature();
float kelembapan = dht.readHumidity();

// Read ultrasonic sensor
long duration, distance;
digitalWrite(TRIGGER_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIGGER_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIGGER_PIN, LOW);
duration = pulseIn(ECHO_PIN, HIGH);
distance = (duration/2) / 29.1;

// Send data to Blynk
Blynk.virtualWrite(V1, distance);
Blynk.virtualWrite(V2, suhu);
Blynk.virtualWrite(V3, kelembapan);

// Display data on LCD 2004
lcd.setCursor(0, 0);
lcd.print("Jarak: ");
lcd.print(distance);
lcd.print(" cm      ");

lcd.setCursor(0, 1);
lcd.print("Status: ");
if (distance <= 20) {
    lcd.print("BAHAYA!      ");
} elseif (distance > 20 && distance <= 50) {
    lcd.print("WASPADA      ");
} else {
    lcd.print("AMAN      ");
}

lcd.setCursor(0, 2);
lcd.print("Suhu: ");
lcd.print(suhu, 1);
lcd.print(" C      ");
```

```
lcd.setCursor(0, 3);
lcd.print("Kelembapan: ");
lcd.print(kelembapan, 0);
lcd.print("%   ");

// Log data to serial monitor
Serial.print("Distance: ");
Serial.print(distance);
Serial.print(" cm | Status: ");
if (distance <= 20) {
    Serial.print("BAHAYA");
} elseif (distance >20&& distance <= 50) {
    Serial.print("WASPADA");
} else {
    Serial.print("AMAN");
}
Serial.print(" | Suhu: ");
Serial.print(suhu);
Serial.print(" C | Kelembapan: ");
Serial.print(kelembapan);
Serial.print(" % | Servo: ");
Serial.print(currentServoAngle);
Serial.println("°");

// Check for flood condition and update LEDs, buzzer, and servo
if (distance <= 20) {
    // Flood condition
    moveServoSmoothly(90); // Tutup gerbang
    tone(BUZZER_PIN, 1000);
    digitalWrite(BUZZER_PIN, HIGH);
    digitalWrite(RED_LED_PIN, HIGH);
    digitalWrite(YELLOW_LED_PIN, LOW);
    digitalWrite(GREEN_LED_PIN, LOW);
    Blynk.logEvent("bahaya_1");
} elseif (distance >20&& distance <= 50) {
    // Warning condition
    moveServoSmoothly(45); // Setengah buka
    noTone(BUZZER_PIN);
    digitalWrite(BUZZER_PIN, LOW);
    digitalWrite(RED_LED_PIN, LOW);
    digitalWrite(YELLOW_LED_PIN, HIGH);
    digitalWrite(GREEN_LED_PIN, LOW);
    Blynk.logEvent("waspada_1");
} elseif (distance >50) {
    // Safe condition
    moveServoSmoothly(0); // Buka penuh
    noTone(BUZZER_PIN);
    digitalWrite(BUZZER_PIN, LOW);
```

```
    digitalWrite(RED_LED_PIN, LOW);
    digitalWrite(YELLOW_LED_PIN, LOW);
    digitalWrite(GREEN_LED_PIN, HIGH);
    Blynk.logEvent("aman_1");
}
delay(500);
}
```