

Rancang Bangun Sistem
Monitoring dan Kontrol Tandon Air Berbasis IoT dengan Antarmuka
Web Server



Oleh :
Achmad yusuf fattah(01)

Guru Pembimbing :
Fajar Setiawan S.Pd

XI SISTEM INFORMASI JARINGAN DAN APLIKASI
SMK NEGERI KABUH
TAHUN PELAJARAN 2024/2025

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan laporan proyek ini dengan judul "Rancang Bangun Sistem Monitoring dan Kontrol Tandon Air Berbasis IoT dengan Antarmuka Web Server". Laporan ini disusun sebagai bagian dari upaya untuk mengembangkan solusi teknologi yang inovatif dan bermanfaat dalam kehidupan sehari-hari.

Proyek ini lahir dari kesadaran akan pentingnya pengelolaan sumber daya air yang efisien dan cerdas di era modern. Dengan memanfaatkan teknologi Internet of Things (IoT) dan antarmuka web server berbasis framework CodeIgniter 4, kami berupaya menciptakan sistem tandon air otomatis yang dapat dimonitor dan dikendalikan secara real-time. Harapan kami, sistem ini dapat memberikan kemudahan, meningkatkan efisiensi, serta mendukung gaya hidup yang lebih berkelanjutan.

Proses penyusunan laporan ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu, kami mengucapkan terima kasih kepada guru pembimbing yang telah memberikan arahan dan masukan berharga, teman-teman yang turut berkontribusi dalam diskusi dan pengujian, serta keluarga yang senantiasa memberikan motivasi dan doa. Kami juga menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat kami harapkan demi perbaikan di masa mendatang.

Jombang, 1 Maret 2025

Achmad yusuf fattah

DAFTAR ISI

KATA PENGANTAR	2
DAFTAR ISI	3
BAB I	5
PENDAHULUAN	5
1.1. Latar Belakang	5
1.2. Rumusan Masalah	5
1.3. Ruang Lingkup Kegiatan	5
1.4. Tujuan Kegiatan	6
1.5. Manfaat Kegiatan	6
1.6. Sistematika Penulisan	6
BAB II	7
TINJAUAN PUSTAKA	7
2.1. Internet of Things (IoT)	7
2.2. Mikrokontroler	7
2.3. IDE Arduino	11
2.5. Sensor Ultrasonik HC-SR04	12
2.5. Buzzer	13
2.6. Power Supply MB-102	13
2.7. Breadboard MB-102	14
2.8. LCD (Liquid Crystal Display)	15
2.9. I2C (Inter Integrated Circuit)	15
2.10. Pompa Air 5v	16
2.11. Relay SSR	17
3.1. Waktu dan Tempat Penelitian	18
3.2. Alat dan Bahan	18
3.2.1. Alat	18
3.2.2. Bahan	18
3.3. Prosedur Kerja	19
3.3.1. Studi literatur	19
3.3.2. Identifikasi masalah	19
3.4. Flowcart	20
3.5. Perancangan web ci4	20
A. Langkah-langkah Menambahkan Fungsi Monitor Pompa	20
B. Langkah-langkah Menambahkan Fungsi Manual	27

C. Langkah-Langkah Menambahkan Fungsi Mode (Auto dan Manual)	32
BAB IV	36
HASIL DAN PEMBAHASAN.....	36
4.1. Hasil	36
4.2. Hasil Kegiatan	36
4.3. Hasil Pengujian	38
BAB III	46
PENUTUP.....	46
DAFTAR PUSTAKA.....	48



BAB I

PENDAHULUAN

1.1. Latar Belakang

Seiring berkembangnya teknologi dan mobilitas penduduk Indonesia yang tinggi, perangkat – perangkat microcomputer telah banyak digunakan dalam berbagai aspek kehidupan sehari-hari. Perangkat microcomputer tersebut dapat berupa smart phone ataupun microcontroller. Pengguna smartphone juga semakin meningkat dikarenakan dengan mudahnya akses peruntukan data dan dapat di akses dan terbuka untuk menyediakan layanan pertukaran data ataupun pengendalian jarak jauh.

Rumah dapat berfungsi sebagai tempat untuk menikmati kehidupan yang nyaman, tempat untuk beristirahat, tempat untuk berkumpulnya keluarga dan tempat untuk menunjukkan tingkat sosial dalam masyarakat. Tetapi terkadang fungsi rumah tersebut tidak dapat terus dirasakan nyaman mungkin. Hal ini terjadi karena adanya kekhawatiran pada pemilik rumah yang belum bisa memantau rumahnya secara menyeluruh. Kekhawatiran ini juga terjadi ketika pemilik rumah yang bepergian keluar kota dalam waktu yang cukup lama, sehingga pemilik rumah memiliki kendali dalam pengawasan rumah. Oleh karena itu saya ingin merancang sebuah alat kendali Smart home. Dengan adanya rancangan alat kendali smart home ini dapat di kontrol dengan jarak yang jauh sehingga pemilik rumah tersebut tidak merasa khawatir lagi.

Smart home ini mulai populer pada tahun 2000 dengan di terapkannya 2 perangkat lokal sederhana, jaringan lokal, dan perangkat sederhana lainnya. Smart home merupakan konsep yang menjanjikan, dengan menawarkan beberapa keuntungan seperti menghadirkan sebuah kenyamanan, meningkatkan keselamatan dan keamanan, serta dapat menghemat penggunaan energi. Terdapat beberapa faktor yang perlu diperhatikan sebelum merancang sistem smart home. Perangkat harus dapat diakses dengan mudah, mudah diperluas sehingga dapat dengan mudah menambahkan perangkat baru, dan harus dapat dengan mudah di kendalikan.

1.2. Rumusan Masalah

- a. Bagaimana cara merancang sistem otomatis yang dapat memantau level air pada tandon secara real-time menggunakan teknologi Internet of Things (IoT)?
- b. Bagaimana cara mengintegrasikan sensor dan aktuator dalam sistem tandon air agar dapat dikendalikan melalui antarmuka web server?
- c. Bagaimana cara mengimplementasikan framework CodeIgniter 4 sebagai platform web server untuk mendukung monitoring dan kontrol tandon air?
- d. Bagaimana mengontrol perangkat yang terhubung di WebServer?

1.3. Ruang Lingkup Kegiatan

Proyek ini akan membahas alat monitoring dan kontrol tandon air berbasis teknologi Internet of Things (IoT) yang terhubung ke web server dengan antarmuka berbasis CodeIgniter 4, dengan ruang lingkup sebagai berikut:

- a. Perangkat yang digunakan untuk melakukan pengontrolan adalah PC atau smartphone yang mendukung akses internet melalui browser.

- b. Software aplikasi yang digunakan adalah framework CodeIgniter 4 sebagai basis pengembangan web server untuk monitoring dan kontrol sistem.
- c. Jenis sensor yang digunakan meliputi sensor ultrasonik untuk mengukur tingkat air, dan sensor tambahan buzzer jika diperlukan untuk mendukung fungsi sistem.

1.4. Tujuan Kegiatan

Adapun tujuan dari proyek ini yang ingin dicapai yaitu sebagai berikut:

- a. Mengembangkan teknologi pengelolaan tandon air secara otomatis dengan memanfaatkan teknologi Internet of Things (IoT) dan antarmuka web server berbasis CodeIgniter 4.
- b. Mengetahui prinsip cara kerja sistem monitoring dan kontrol tandon air berbasis IoT yang terintegrasi dengan web server untuk mendukung efisiensi pengelolaan air.

1.5. Manfaat Kegiatan

Adapun manfaat dari proyek ini yang ingin dicapai yaitu sebagai berikut:

- a. Menambah pengetahuan serta wawasan dalam bidang teknologi Internet of Things (IoT) dan pengembangan web server, khususnya menggunakan framework CodeIgniter 4.
- b. Membantu pengguna untuk memantau dan mengendalikan sistem tandon air secara praktis dan efisien melalui perangkat yang terhubung ke internet.

1.6. Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini meliputi latar belakang masalah, rumusan masalah, ruang lingkup kegiatan, tujuan kegiatan, manfaat kegiatan, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini meliputi teori-teori yang berkaitan tentang rancang bangun sistem monitoring dan kontrol tandon air berbasis IoT serta pengembangan web server menggunakan CodeIgniter 4 yang mendukung proses perancangan.

BAB III METODE PERANCANGAN

Bab ini meliputi cara pembuatan sistem monitoring dan kontrol tandon air berbasis IoT yang terintegrasi dengan antarmuka web server.

BAB IV HASIL DAN PEMBAHASAN

Bab ini meliputi hasil pengembangan sistem, pengolahan data, serta analisis hasil penelitian.

BAB V KESIMPULAN DAN SARAN

Bab ini meliputi kesimpulan dari hasil yang diperoleh serta saran-saran yang diharapkan dapat bermanfaat untuk pengembangan dan pemanfaatan sistem di masa mendatang.

BAB II

TINJAUAN PUSTAKA

2.1. Internet of Things (IoT)

Internet of Things adalah suatu konsep dimana objek tertentu punya kemampuan untuk mentransfer data lewat jaringan tanpa memerlukan adanya interaksi dari manusia ke manusia ataupun dari manusia ke perangkat komputer.

Apa saja contohnya? Contohnya adalah untuk pengolahan bahan pangan, elektronik, dan berbagai mesin atau teknologi lainnya yang semuanya tersambung ke jaringan lokal maupun global lewat sensor yang tertanam dan selalu menyala aktif.

Jadi, sederhananya istilah Internet of Things ini mengacu pada mesin atau alat yang bisa diidentifikasi sebagai representasi virtual dalam strukturnya yang berbasis Internet.

Cara Kerja Internet of Things itu seperti apa? Sebenarnya IoT bekerja dengan memanfaatkan suatu argumentasi pemrograman, dimana tiap-tiap perintah argumen tersebut bisa menghasilkan suatu interaksi antar mesin yang telah terhubung secara otomatis tanpa campur tangan manusia dan tanpa terbatas jarak berapapun jauhnya.

Jadi, Internet di sini menjadi penghubung antara kedua interaksi mesin tersebut. Lalu di mana campur tangan manusia? Manusia dalam IoT tugasnya hanyalah menjadi pengatur dan pengawas dari mesin-mesin yang bekerja secara langsung tersebut.

Adapun tantangan terbesar yang bisa menjadi hambatan dalam mengkonfigurasi IoT adalah bagaimana menyusun jaringan komunikasinya sendiri.

2.2. Mikrokontroler

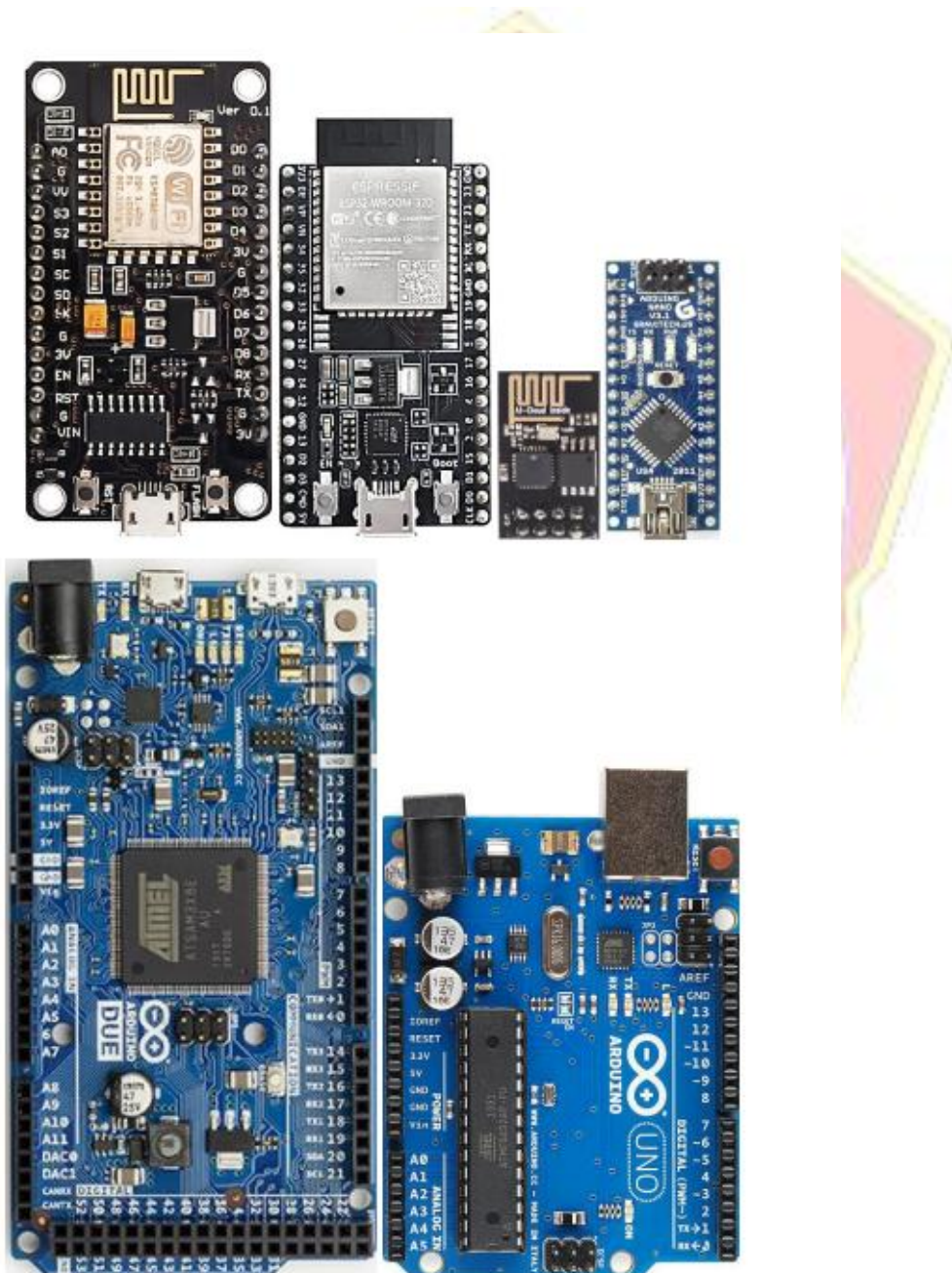
Mikrokontroler adalah suatu jenis integrated circuit (IC) yang mengintegrasikan unit pemrosesan pusat, memori, dan perangkat masukan/keluaran ke dalam satu chip kecil. Fungsi utama mikrokontroler adalah untuk mengontrol operasi dari suatu sistem atau perangkat elektronik. Mikrokontroler banyak digunakan dalam berbagai aplikasi, mulai dari perangkat rumah tangga hingga sistem kontrol industri.

Beberapa karakteristik umum dari mikrokontroler meliputi:

1. Unit Pemrosesan Pusat (CPU): Mikrokontroler memiliki CPU yang dapat melakukan berbagai operasi aritmetika dan logika. Ini adalah otak dari mikrokontroler yang mengatur eksekusi program.
2. Memori: Terdapat dua jenis memori utama pada mikrokontroler, yaitu program memory (tempat penyimpanan program yang akan dieksekusi) dan data memory (tempat penyimpanan variabel dan data sementara selama eksekusi program).
3. Perangkat Masukan/Keluaran (I/O): Mikrokontroler memiliki pin-pin I/O yang memungkinkan komunikasi dengan dunia luar, seperti sensor dan aktuator. Ini memungkinkan mikrokontroler untuk menerima informasi dari lingkungannya dan mengontrol perangkat eksternal.

4. Sistem Pengaturan Clock: Sebagian besar mikrokontroler memiliki sumber clock internal atau eksternal untuk mengatur kecepatan operasionalnya.
5. Peripheral Devices: Beberapa mikrokontroler memiliki perangkat tambahan seperti timer, counter, USART (Universal Synchronous/Asynchronous Receiver/Transmitter), PWM (Pulse Width Modulation), dan lainnya, yang memperluas kemampuan mikrokontroler untuk berbagai aplikasi.

Kelebihan penggunaan mikrokontroler termasuk ukuran yang kecil, konsumsi daya yang rendah, dan fleksibilitas yang tinggi dalam pengembangan berbagai sistem elektronik. Mikrokontroler banyak digunakan dalam perangkat seperti mikrooven, mesin cuci, kamera digital, sistem kontrol otomotif, dan sebagainya.

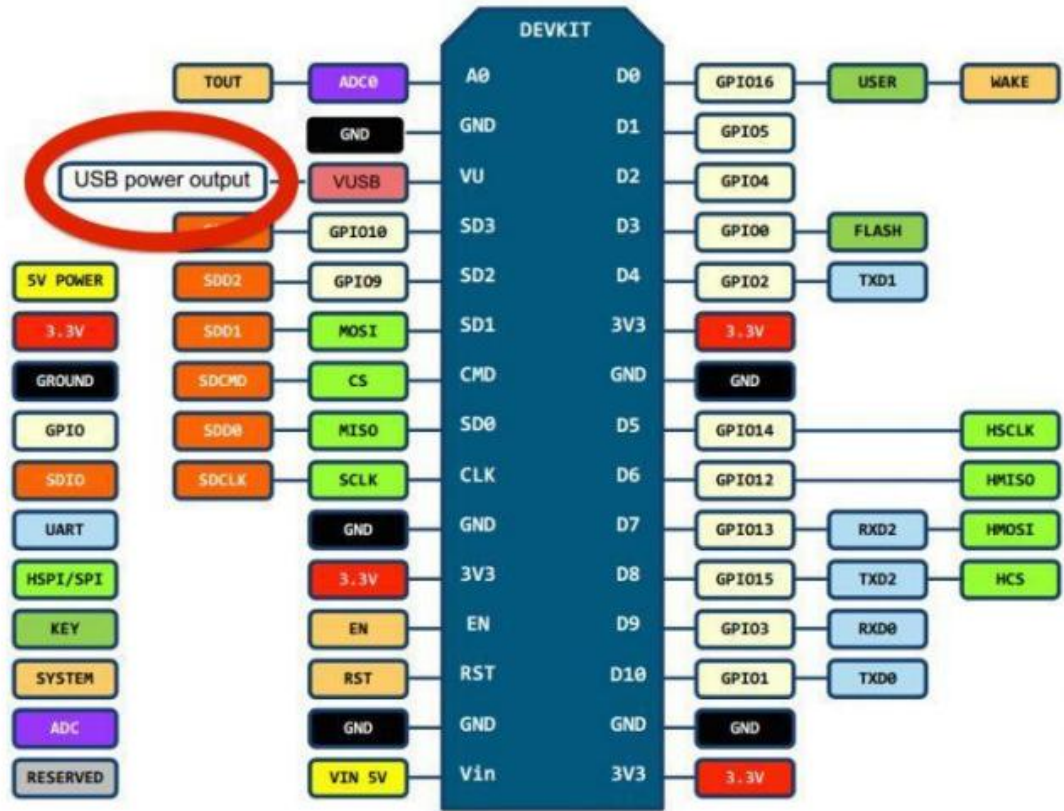


Ada banyak jenis mikrokontroler yang berbeda di pasaran, dan perbedaan antara mereka terletak pada arsitektur, fitur, kinerja, dan aplikasi yang ditujukan. Beberapa merek dan keluarga mikrokontroler yang populer termasuk:

1. Espressif Systems; Sebuah perusahaan yang terkenal dengan pengembangan mikrokontroler dan modul Wi-Fi yang populer dalam proyek Internet of Things (IoT), dengan fitur builtin Wi-Fi 802.11 b/g/n., UART
 - ESP8266 ; CPU Tensilica L106, clock hingga 160 MHz,
 - ESP32 ; penerus ESP8266, dua core prosesor Xtensa 32-bit LX6, Bluetooth Low Energy (BLE), SPI, I2C, clock hingga 240 MHz
2. Arduino; adalah platform pengembangan perangkat keras open-source yang mencakup papan mikrokontroler dan lingkungan pengembangan perangkat lunak (IDE). Platform ini dirancang untuk memudahkan orang yang tidak memiliki latar belakang teknik elektronik dan pemrograman yang mendalam untuk memulai dan mengembangkan proyek elektronik. Beberapa jenis board Arduino; Uno, Nano, Micro, Mega, Due, Leonardo.
3. Atmel AVR: Mikrokontroler AVR dikenal karena kepopulerannya dalam dunia hobi dan DIY. Mereka digunakan dalam Arduino, sebuah platform pengembangan elektronik open-source yang sangat populer.
4. ARM Cortex-M: Keluarga mikrokontroler ARM Cortex-M memiliki arsitektur yang lebih kompleks dan biasanya digunakan dalam aplikasi yang memerlukan kinerja tinggi. Banyak produsen mikrokontroler, seperti STMicroelectronics, NXP, dan Texas Instruments, memproduksi mikrokontroler berbasis arsitektur ARM Cortex-M.
5. Microchip PIC (Peripheral Interface Controller): PIC adalah keluarga mikrokontroler yang banyak digunakan dalam berbagai aplikasi. Mereka dikenal dengan arsitektur RISC (Reduced Instruction Set Computing) mereka dan dapat ditemukan dalam berbagai varian dengan berbagai fitur.
6. Raspberry Pi (bukan mikrokontroler, tapi single-board computer): Raspberry Pi adalah mini komputer, berjalan dengan Operating System Linux, papan tunggal yang memiliki kemampuan komputasi yang lebih besar dibandingkan dengan mikrokontroler.

Perbedaan antar mikrokontroler dapat melibatkan hal-hal seperti arsitektur, kecepatan clock, kapasitas memori, jumlah pin I/O, dan kemampuan perangkat keras tambahan. Pemilihan mikrokontroler yang tepat tergantung pada persyaratan spesifik dari proyek atau aplikasi yang sedang dikembangkan. Beberapa mikrokontroler dirancang untuk aplikasi tertentu, seperti kendali motor, sistem sensor, atau komunikasi nirkabel, sementara yang lainnya dirancang untuk tugas umum.

Pada kelas ini kita akan gunakan mikrokontoler ESP8266 dengan operasi kerja tegangan JEDEC (tegangan 3.3VDC), Tidak seperti mikrokontroler AVR dan sebagian besar board Arduino yang memiliki tegangan TTL 5 volt. Meskipun begitu, node mcu masih bisa terhubung dengan 5V namun melalui port micro USB atau pin Vin yang disediakan oleh board-nya. Namun karena semua pin pada ESP8266 tidak toleran terhadap masukan 5V. Maka sebaiknya tidak langsung mencatunya dengan tegangan TTL jika tidak ingin merusak board anda. Anda bisa menggunakan Level Logic Converter untuk mengubah tegangan ke nilai aman 3.3v. ESP8266 memiliki pin Input Output sbb.:

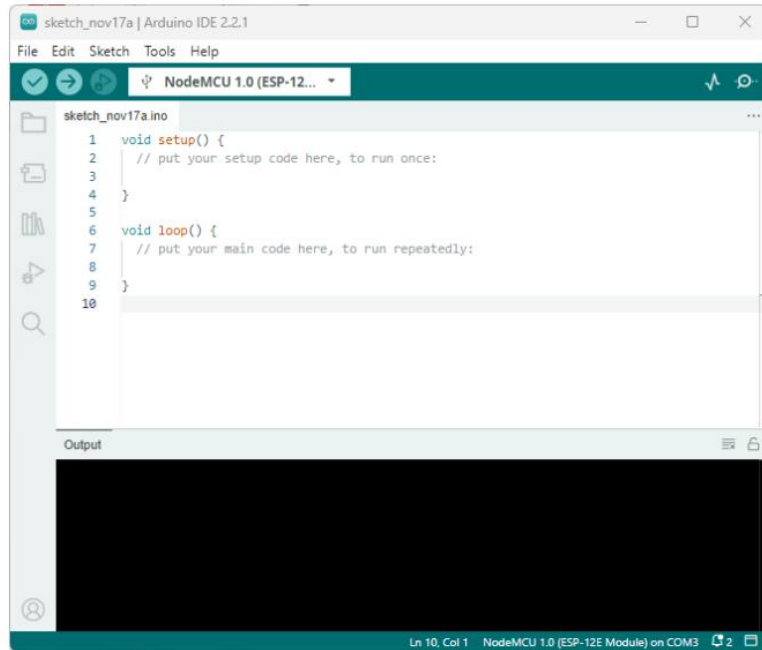


Pin	Name	Desired level during power on	Before first software download	Necessary Function	ANT Switch	Extendable function	Extendable GPIO/I2C
8	XPD_DCDC		Output high	Deep-Sleep Wakeup	Yes [0]		GPIO16 (optional internal pull-down)
9	MTMS		Input, Internal pull-up resistor			HSPICLK	GPIO14 (optional internal pull-up)
10	MTDI		Input, Internal pull-up resistor			HSPIQ	GPIO12 (optional internal pull-up)
12	MTCK		Input, Internal pull-up resistor			HSPID	GPIO13 (optional internal pull-up)
13	MTDO	Low	Input, Internal pull-up resistor			HSPICS	GPIO15 (optional internal pull-up)
14	GPIO2	High	Output, Many toggles	U1TXD			GPIO2 (optional internal pull-up)
15	GPIO0	High (uart download: Low)	Output, Many toggles		Yes [0]	SPICS2	GPIO0 (optional internal pull-up)
16	GPIO4		Input, Internal pull-up resistor				GPIO4 (optional internal pull-up)
18	SD_DATA_2		Input, Hi-Z	SPIHD	Yes	HSPIHD	GPIO9 (optional internal pull-up)
19	SD_DATA_3		Input, Hi-Z	SPIWP	Yes[0]	HSPIWP	GPIO10 (optional internal pull-up)
20	SD_CMD		Input, Hi-Z	SPICS0			No
21	SD_CLK		Input, Hi-Z	SPICLK			No
22	SD_DATA_0		Input, Hi-Z	SPIQ			No
23	SD_DATA_1		Input, Hi-Z	SPID	Yes		No
24	GPIO5		Input, Internal pull-up resistor				GPIO5 (optional internal pull-up)
25	U0RXD		Input, Internal pull-up resistor	U0RXD	Yes		GPIO3 (optional internal pull-up)
26	U0TXD	High	Output, Many toggles	U0TXD	Yes	SPICS1	GPIO1 (optional internal pull-up)

2.3. IDE Arduino

IDE itu merupakan kependekan dari Integrated Development Environment, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dinamakan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (Sketch) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama Bootlader yang berfungsi sebagai penengah antara compiler Arduino dengan mikrokontroler.

IDE Arduino memiliki tampilan program yang berfungsi sebagai Program Editor, Compiler, Uploader, Monitor dan Plotter. Compiler ini mengubah file program *.ino menjadi *.asm *.hex dan lainnya, hex file yang terbentuk akan kita tanamkan pada Arduino Board.



2.5. Sensor Ultrasonik HC-SR04

Sensor ultrasonik HC-SR04 adalah alat pengukur jarak non-kontak yang menggunakan gelombang suara ultrasonik (frekuensi tinggi di atas 20 kHz, khususnya 40 kHz untuk HC-SR04) untuk mendeteksi jarak antara sensor dan objek di depannya. Prinsip kerjanya mirip sonar atau radar, yaitu mengirimkan gelombang suara, menunggu pantulannya (echo), lalu menghitung jarak berdasarkan waktu tempuh gelombang tersebut. Sensor ini banyak digunakan dalam otomasi, robotika, dan IoT karena presisi yang cukup baik, kemudahan integrasi dengan mikrokontroler, dan kemampuannya bekerja di berbagai kondisi pencahayaan.



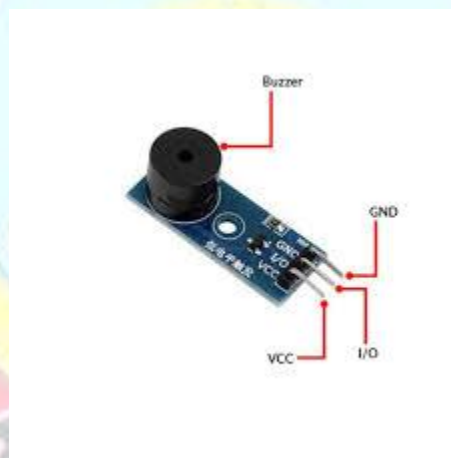
Sensor HC-SR04 memiliki empat pin utama:

1. **VCC**: Pin daya, biasanya terhubung ke sumber tegangan 5V.
2. **GND**: Pin ground untuk menyelesaikan rangkaian listrik.
3. **Trig (Trigger)**: Pin input yang digunakan untuk memulai pengukuran dengan mengirim sinyal pulsa.
4. **Echo**: Pin output yang memberikan sinyal waktu pantulan gelombang ultrasonik.

2.5. Buzzer

Modul buzzer adalah komponen elektronik yang digunakan untuk menghasilkan suara atau bunyi sederhana, seperti beep atau nada tertentu. Biasanya, buzzer digunakan dalam proyek elektronik, seperti pada Arduino atau Esp8266, untuk memberikan notifikasi suara, alarm, atau indikator tertentu. Ada dua jenis buzzer utama: buzzer aktif (active buzzer) dan buzzer pasif (passive buzzer).

- Buzzer Aktif: Sudah memiliki osilator internal, sehingga hanya perlu diberikan tegangan (misalnya 5V atau 3.3V) untuk menghasilkan suara dengan frekuensi tetap.
- Buzzer Pasif: Tidak memiliki osilator internal, sehingga perlu sinyal input dengan frekuensi tertentu (misalnya dari PWM pada mikrokontroler) untuk menghasilkan suara.



Modul Buzzer dengan 3 Pin (umum pada buzzer pasif atau modul tertentu):

- **Pin VCC**: Tegangan positif (biasanya 3.3V atau 5V).
- **Pin GND**: Ground.
- **Pin SIG (Signal)**: Pin sinyal, digunakan untuk mengirimkan sinyal frekuensi dari mikrokontroler (misalnya PWM) untuk mengatur nada pada buzzer pasif.

2.6. Power Supply MB-102

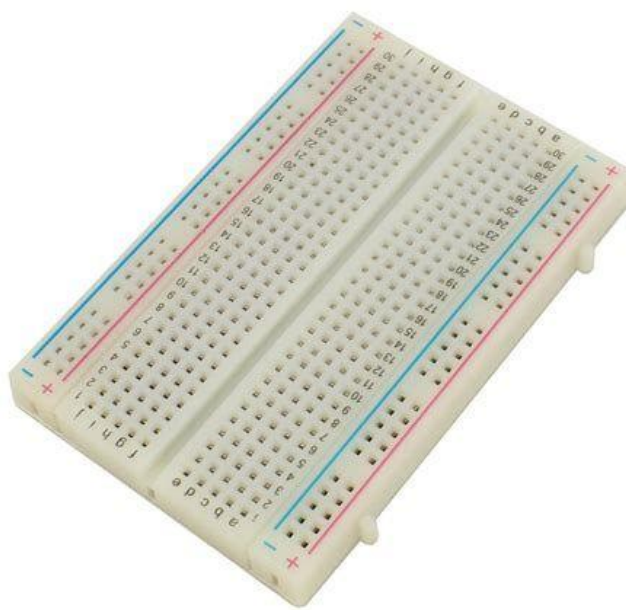
Power supply MB-102 adalah modul yang menyediakan tegangan keluaran 3.3V dan 5V yang dapat dipilih, dengan input tegangan dari sumber eksternal sebesar 6.5V hingga 12V (melalui jack DC) atau melalui konektor USB (5V). Modul ini dilengkapi dengan regulator tegangan untuk

memastikan keluaran stabil, sakelar on/off, dan indikator LED untuk menunjukkan status daya. Fungsi utamanya adalah menyediakan daya yang konsisten ke breadboard tanpa perlu menyusun rangkaian catu daya tambahan, sehingga mempermudah prototyping.



2.7. Breadboard MB-102

Breadboard MB-102 adalah papan dengan banyak lubang koneksi yang diatur dalam pola tertentu, memungkinkan pengguna untuk memasukkan kaki komponen elektronik (seperti resistor, kapasitor, IC, atau kabel jumper) untuk membuat rangkaian. Di dalam breadboard, terdapat klip logam yang menghubungkan lubang-lubang tertentu secara internal, sehingga memudahkan pembuatan koneksi listrik tanpa alat khusus.



2.8. LCD (Liquid Crystal Display)

LCD (Liquid Crystal Display) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama. Adapun fitur yang disajikan dalam LCD ini adalah:

- 1) Terdiri dari 16 karakter dan 2 baris.
- 2) Mempunyai 192 karakter tersimpan.
- 3) Terdapat karakter generator terprogram.
- 4) Dapat dialamati dengan mode 4-bit dan 8-bit.
- 5) Dilengkapi dengan back light.

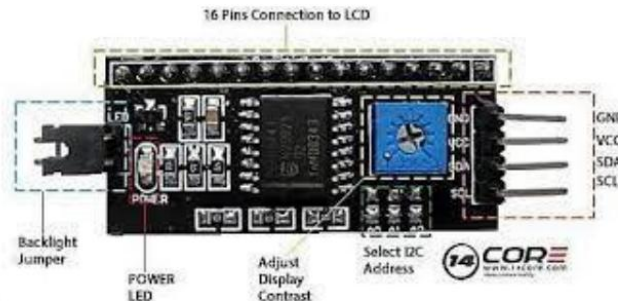
Proses inisialisasi pin arduino yang terhubung ke pin LCD RS, Enable, D4, D5, D6, dan D7, dilakukan dalam baris LiquidCrystal (2, 3, 4, 5, 6, 7), dimana lcd merupakan variable yang dipanggil setiap kali intruksi terkait LCD akan digunakan.



2.9. I2C (Inter Integrated Circuit)

Inter Integrated Circuit atau sering disebut I2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didisain khusus untuk mengirim maupun menerima data. Sistem I2C terdiri dari saluran SCL (Serial Clock) dan SDA (Serial Data) yang membawa informasi data antara I2C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai Master dan Slave. Master adalah piranti yang memulai transfer data pada I2C Bus dengan membentuk sinyal Start, 17 mengakhiri transfer data dengan membentuk sinyal Stop, dan membangkitkan sinyal clock. Slave adalah piranti yang dialamati master. Sinyal Start merupakan sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “1” menjadi “0” pada saat SCL “1”. Sinyal Stop merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “0” menjadi “1” pada saat SCL “1”. Sinyal dasar yang lain dalam I2C Bus adalah sinyal acknowledge yang disimbolkan dengan ACK. Setelah transfer data oleh master berhasil diterima slave, slave akan menjawabnya dengan mengirim sinyal acknowledge, yaitu dengan membuat SDA menjadi “0” selama siklus clock ke 9. Ini menunjukkan bahwa Slave telah menerima 8 bit data dari Master. Dalam melakukan transfer data pada I2C Bus, kita harus mengikuti tata cara yang telah ditetapkan yaitu:

- 1) Transfer data hanya dapat dilakukan ketika Bus tidak dalam keadaan sibuk.
- 2) Selama proses transfer data, keadaan data pada SDA harus stabil selama SCL dalam keadaan tinggi. Keadaan perubahan “1” atau “0” pada SDA hanya dapat dilakukan selama SCL dalam keadaan rendah. Jika terjadi perubahan keadaan SDA pada saat SCL dalam keadaan tinggi, maka perubahan itu dianggap sebagai sinyal Start atau sinyal Stop



2.10. Pompa Air 5v

Pompa air 5V umumnya merupakan pompa submersi (submersible) atau pompa mikro yang bekerja dengan mekanisme sederhana, seperti tipe sentrifugal atau diafragma. Pompa ini menggerakkan air dengan cara menyedot melalui saluran masuk (inlet) dan mengeluarkannya melalui saluran keluar (outlet). Ada dua jenis utama:

- **Pompa Brushless:** Menggunakan motor tanpa sikat (brushless DC motor) yang lebih tahan lama, efisien, dan minim kebisingan.
- **Pompa Standar:** Menggunakan motor sikat (brushed motor) yang lebih murah tetapi kurang awet dibandingkan tipe brushless.

Pompa ini sering dilengkapi dengan kabel sederhana (biasanya merah untuk positif dan hitam untuk ground) yang mudah dihubungkan ke sumber daya 5V.

Pin out pompa air 5V biasanya sangat sederhana:

- **Kabel Merah (+):** Dihubungkan ke tegangan positif 5V.
- **Kabel Hitam (-):** Dihubungkan ke ground (GND).

Spesifikasi teknisnya bervariasi tergantung model, tetapi berikut adalah contoh umum

- **Tegangan Operasi:** 5V DC (beberapa model bisa bekerja pada 3-6V).
- **Konsumsi Arus:** Biasanya 100-500 mA, tergantung daya pompa.
- **Laju Alir (Flow Rate):** 50-300 liter per jam (L/h), tergantung ukuran dan tipe pompa.
- **Tinggi Angkat Maksimum (Head):** 0,5-1,5 meter (jarak vertikal maksimum yang bisa dicapai air).
- **Daya:** 0,5-3 watt.
- **Dimensi:** Kecil dan ringkas, biasanya sekitar 2-5 cm panjangnya.

- **Material:** Plastik tahan lama atau kombinasi dengan komponen logam anti-karat (misalnya poros keramik pada tipe brushless).



2.11. Relay SSR

Solid State Relay (SSR) adalah jenis relay yang menggunakan komponen elektronik semikonduktor (seperti transistor, thyristor, atau triac) untuk melakukan fungsi switching, berbeda dengan relay mekanis yang menggunakan kontak fisik. SSR tidak memiliki bagian yang bergerak, sehingga lebih tahan lama, cepat, dan andal dalam aplikasi tertentu.



SSR umumnya memiliki 4 pin utama, meskipun jumlahnya bisa bervariasi tergantung jenis dan modelnya. Berikut adalah penjelasan pin-out SSR:

Input Control (Sisi Kontrol):

Pin 1 (Input +): Terminal positif untuk sinyal kontrol (biasanya tegangan DC rendah, seperti 3V, 5V, atau 12V).

Pin 2 (Input -): Terminal negatif atau ground untuk sinyal kontrol.

Output Load (Sisi Beban):

Pin 3 (Output +): Terminal positif untuk beban (output).

Pin 4 (Output -): Terminal negatif untuk beban (output).

BAB III

METODE PENELITIAN

3.1. Waktu dan Tempat Penelitian

Perancangan, pembuatan serta pengujian alat kendali tandon air otomatis ini dimulai pada tanggal 1 maret 2025 sampai dengan 7 april 2025, yang bertempat di Labsija smk negeri kabuh dan rumah penulis.

3.2. Alat dan Bahan

Dalam perancangan ini akan digunakan berbagai alat dan bahan. Adapun alat dan bahan yang akan digunakan dapat dilihat pada tabel 3.1 dan tabel 3.2.

3.2.1. Alat

Tabel 3.1 Daftar alat

NO	ALAT	JUMLAH
1	Laptop	1 buah
2	Kabel usb to micro	1 buah
3	Gunting	1 buah
4	Solasi kertas	1 buah
5	Obeng (+) dan (-)	1 buah
6	Breadboard	1 buah
7	Smartphone	1 buah
8	Bor listrik mini	1 buah
9	Kabel jumper	1 buah
10	Penggaris 30cm	1 buah
11	Kabel data Usb to Usb	1 buah
12	Solasi isolator	1 buah
13	Kabel kecil 50cm	1 buah

3.2.2. Bahan

Tabel 3.2 Daftar bahan

NO	BAHAN	JUMLAH
1	Nodemcu Lolin v3 esp8266	1 buah
2	Sensor ultrasonik	1 buah

3	Lcd 12c	1 buah
4	Relay SSR	1 buah
5	Buzzer	1 buah
6	Led 5mm	2 buah
7	Resistor 220 OHM	2 buah
8	Holder baterai AA 4x	1 buah
9	Pompa air 5v	1 buah
10	Baterai AA 4x	4 buah
11	Box projek	1 buah
12	Power supply mb-102	1 buah
13	Selang kecil 1m	1 buah
14	Toples 20cm	2 buah

3.3. Prosedur Kerja

Untuk memperoleh suatu alat yang baik dari efisiensi yang baik, maka dibutuhkan langkah-langkah sebagai berikut:

3.3.1. Studi literatur

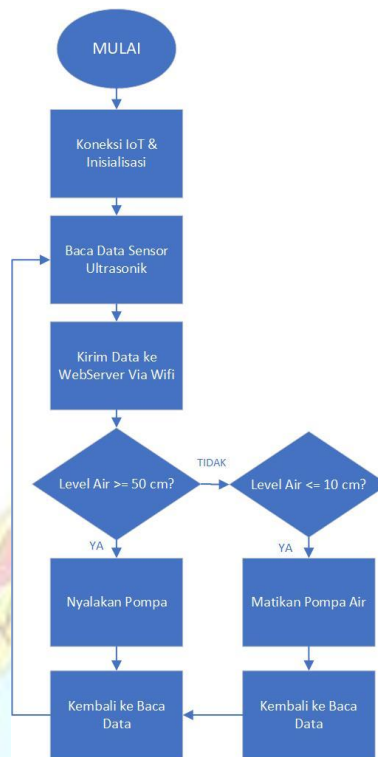
Dalam perancangan alat ini, langkah awal yang dilakukan adalah mencari sebanyak-banyaknya data serta informasi melalui media cetak maupun elektronik, dimana informasi tersebut harus relevan dengan alat yang akan dibuat. Referensi yang diperlukan dalam penulisan laporan ini yaitu: Nodemcu v3, Sensor ultrasonik, Lcd 12c, Relay SSR, Buzzer, Led, Pompa air, Powersupply mb-102, Breadboard.

3.3.2. Identifikasi masalah

Hal yang paling penting dalam perancangan ini adalah identifikasi masalah. Adapun masalah yang mungkin akan dihadapi dalam proses perancangan dan pembuatan alat ini yaitu:

- Menguji keseluruhan alat
- Membuat program pada Arduino
- Melakukan uji fungsional

3.4. Flowcart



3.5. Perancangan web ci4

Di bagian ini, kita akan membuat aplikasi web sederhana menggunakan CodeIgniter 4 (CI4) untuk memantau level air dengan nama "Water Level Monitoring". Aplikasi ini akan menampilkan data level air, grafik, dan memperbarui data secara otomatis. Berikut langkah-langkahnya:

A. Langkah-langkah Menambahkan Fungsi Monitor Pompa

1. Menentukan Rute (Routes)

Rute adalah alamat URL yang digunakan untuk mengakses halaman atau fungsi di aplikasi. File rute ada di **app/Config/Routes.php**. Ini adalah rute yang kita gunakan:

- **/pompa**: Halaman utama untuk melihat data level air.
- **/cekpompa**: Alamat untuk mengambil data level air terbaru.
- **/pompa/grafikpompa**: Alamat untuk mengambil data level air dalam bentuk grafik.
- **/oleh-data/(angka)**: Alamat untuk memperbarui data level air dari luar.

```
//fungsi monitor air

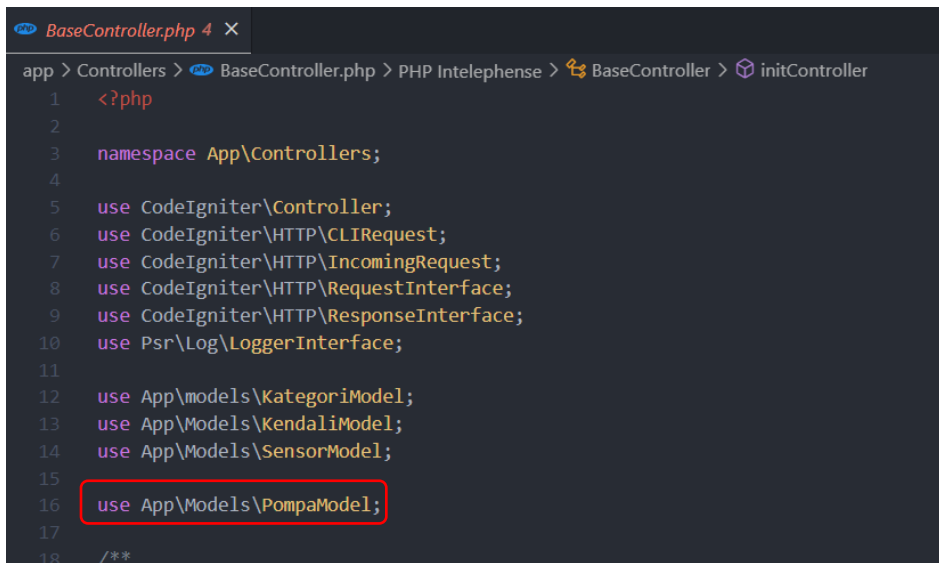
$route->get(from: 'pompa', to: 'PompaController::index'); // Halaman utama monitoring
$route->get(from: 'cekpompa', to: 'PompaController::cekpompa'); // API untuk mendapatkan status pompa
$route->get(from: 'pompa/grafikpompa', to: 'PompaController::getPompa'); // API untuk memperbarui status pompa
$route->get(from: 'oleh-data/(segment)', to: 'PompaController::update/$1'); // Endpoint untuk menerima data eksternal
```


2. Menambahkan Fungsi pada BaseController

Untuk mempermudah penggunaan model **PompaModel** di semua controller, kita bisa menambahkan kode di **BaseController.php**. File ini ada di **app/Controllers/BaseController.php**. Dengan ini, kita tidak perlu membuat ulang objek **PompaModel** di setiap controller.

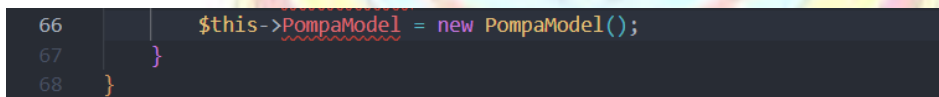
Langkahnya:

- Tambahkan baris untuk memanggil **PompaModel**.
- Buat properti agar bisa digunakan di semua fungsi turunan (controller lain).



```
app > Controllers > BaseController.php > PHP Intelephense > BaseController > initController
1  <?php
2
3  namespace App\Controllers;
4
5  use CodeIgniter\Controller;
6  use CodeIgniter\HTTP\CLIRequest;
7  use CodeIgniter\HTTP\IncomingRequest;
8  use CodeIgniter\HTTP\RequestInterface;
9  use CodeIgniter\HTTP\ResponseInterface;
10 use Psr\Log\LoggerInterface;
11
12 use App\Models\KategoriModel;
13 use App\Models\KendaliModel;
14 use App\Models\SensorModel;
15
16 use App\Models\PompaModel;
17
18 /**
```

Pada bagian bawah juga di tambahkan objek **PompaModel**

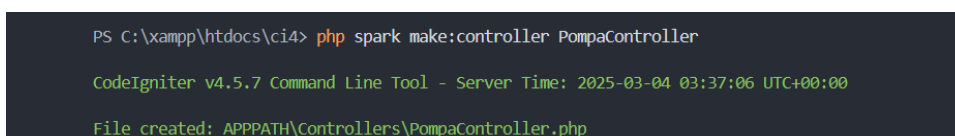


```
66     $this->PompaModel = new PompaModel();
67 }
68 }
```

3. Membuat Controller

Controller adalah bagian yang mengatur logika aplikasi. Kita buat file **PompaController.php** di **app/Controllers**. Fungsinya:

- **index**: Menampilkan halaman utama dengan data level air.
- **cekpompa**: Mengambil data terbaru untuk ditampilkan secara langsung.
- **getPompa**: Mengambil data untuk grafik dalam format JSON.
- **update**: Memperbarui data level air di database.



```
PS C:\xampp\htdocs\ci4> php spark make:controller PompaController

CodeIgniter v4.5.7 Command Line Tool - Server Time: 2025-03-04 03:37:06 UTC+00:00

File created: APPPATH\Controllers\PompaController.php
```

```

PompaController.php 9+ X
app > Controllers > PompaController.php > PHP > PompaController
1  <?php
2
3  namespace App\Controllers;
4
5  use App\Controllers\BaseController;
6  use CodeIgniter\HTTP\ResponseInterface;
7  use App\Models\PompaModel;
8
9  0 references | 0 implementations
10 class PompaController extends BaseController
11 {
12     0 references | 0 overrides
13     public function __construct() {
14         $this->db = \Config\Database::connect();
15         $this->builder = $this->db->table(tableName: 'pompa');
16     }
17
18     0 references | 0 overrides
19     public function index(): string
20     {
21         $data = [
22             'title' => 'Pompa',
23             'dataPompa' => $this->PompaModel->where('id',1)->findAll()
24         ];
25
26         return view(name: 'pompa/index', data: $data);
27     }
28
29     0 references | 0 overrides
30     public function cekpompa(): string
31     {
32         $data = [
33             'dataPompa' => $this->PompaModel->where('id',1)->findAll()
34         ];
35
36         return view(name: 'pompa/cekpompa', data: $data);
37     }
38
39     0 references | 0 overrides
40     public function getPompa(): void
41     {
42         $db = \Config\Database::connect();
43         $data = [];
44         $query = $db->query(sql: "SELECT * FROM pompa WHERE id = 1");
45
46         $row = $query->getRowArray();
47         $data[] = $row; //tampilkan seluruh data array, jika menampilkan perkolom $data[] = $row['suhu'];
48
49         echo json_encode(value: $data);
50     }
51
52     0 references | 0 overrides
53     public function update($pompa): RedirectResponse
54     {
55         $request = \Config\Services::request();
56         $path = $request->getPath();
57
58         $segments = explode(separator: '/', string: $path);
59
60         $segment1 = $segments[1];
61         // $segment2 = $segments[2];
62
63         $updateData = array(
64             'pompa' => $segment1,
65             // 'kelembaban' => $segment2,
66         );
67
68         $this->builder->where(key: 'id', value: 1);
69         $this->builder->update(set: $updateData);
70         return redirect()->to(uri: base_url(relativePath: 'pompa'));
71     }
72 }

```

4. Membuat Tabel di Database (Migration)

Kita buat tabel pompa di database untuk menyimpan data level air. File migrasi ada di **app/Database/Migrations**. Tabel ini punya dua kolom:

- id: Nomor unik untuk setiap data.
- pompa: Nilai level air (misalnya "4 CM").

ketik prompt dibawah ini untuk membuat file migrasi

```
PS C:\xampp\htdocs\ci4> php spark make:migration Pompa

CodeIgniter v4.5.7 Command Line Tool - Server Time: 2025-03-04 03:47:38 UTC+00:00

File created: APPPATH\Database\Migrations\2025-03-04-034738_Pompa.php
```

Ketik prompt dibawah ini supaya tabel migrasi tampil di database

```
PS C:\xampp\htdocs\ci4> php spark migrate

CodeIgniter v4.5.7 Command Line Tool - Server Time: 2025-03-04 04:11:28 UTC+00:00

Running all new migrations...
Running: (App) 2025-03-04-034738_App\Database\Migrations\Pompa
Migrations complete.
```

```
2025-03-04-034738_Pompa.php 1 X
app > Database > Migrations > 2025-03-04-034738_Pompa.php > PHP Intelephense > Pompa > down

1  <?php
2
3  namespace App\Database\Migrations;
4
5  use CodeIgniter\Database\Migration;
6
7  class Pompa extends Migration
8  {
9      public function up(): void
10     {
11         $this->forge->addField(fields: [
12             'id' => [
13                 'type'          => 'INT',
14                 'constraint'    => 5,
15                 'auto_increment' => true,
16                 'unsigned'      => false,
17             ],
18             'pompa' => [
19                 'type'          => 'VARCHAR',
20                 'constraint'    => '6',
21             ],
22         ]);
23         $this->forge->addKey(key: 'id', primary: true);
24         $this->forge->createTable(table: 'pompa');
25     }
26
27     public function down(): void
28     {
29         $this->forge->dropTable(tableName: 'pompa');
30     }
31 }
32
```

5. Membuat Model

Model digunakan untuk mengambil atau menyimpan data ke database. File model ada di **app/Models/PompaModel.php**.

ketik prompt dibawah ini untuk membuat file migrasi

```
PS C:\xampp\htdocs\ci4> php spark make:model PompaModel

CodeIgniter v4.5.7 Command Line Tool - Server Time: 2025-03-04 03:39:33 UTC+00:00

File created: APPPATH\Models\PompaModel.php
```



```
PompaModel.php X
app > Models > PompaModel.php > PHP Intelephense > PompaModel
4
5 use CodeIgniter\Model;
6
7 class PompaModel extends Model
8 {
9     0 references
10    protected $table          = 'pompa';
11    0 references
12    protected $primaryKey     = 'id';
13    0 references
14    protected $useAutoIncrement = true;
15    0 references
16    protected $returnType      = 'object';
17    0 references
18    protected $useSoftDeletes  = false;
19    0 references
20    protected $protectFields   = true;
21    0 references
22    protected $allowedFields   = ['pompa'];
23
24    0 references | 0 overrides
25    public function getPompa(): array
26    {
27        return $this->findAll();
28    }
29 }
```

6. Membuat Tampilan Halaman Utama (Index)

Halaman utama (**index.php**) ada di **app/Views/pompa**. Halaman ini menampilkan:

- Data level air secara langsung (diperbarui tiap detik).
- Grafik batang yang menunjukkan level air (diperbarui tiap 3 detik).


```

index.php X
app > Views > pompa > index.php > div#layoutSidenav_content
1 <?=> $this->extend('admin/layout/template'); ?>
2
3 <?=> $this->section('content'); ?>
4
5 <div id="layoutSidenav_content">
6     <main>
7         <div class="container-fluid px-4">
8             <h1 class="mt-4">Water Level Monitoring</h1>
9             <ol class="breadcrumb mb-4">
10                 <li class="breadcrumb-item"><a href="<?=> base_url(relativePath: 'dashboard'); ?>">Dashboard</a></li>
11                 <li class="breadcrumb-item active">Static Navigations</li>
12             </ol>
13             <div class="row">
14                 <div class="col-lg-6 offset-lg-3">
15                     <div class="card" style="background-color: #C108CB;">
16                         <div class="card-body" style="text-align: center;">
17                             <h5 class="card-title"><i class="bi-thermometer-sun text-white"></i> Water Level monitor</h5>
18                             <button type="button" class="btn btn-lg mb-2 text-white" style="background-color: #6A9C89; border-color: #6A9C89;">
19                                 Level: Normal &nbsp;<span id="cekpompa" class="badge bg-white" style="color: #6A9C89;">4 &nbsp; </span>
20                             </button>
21                         </div>
22                     </div>
23                 </div>
24             </div>
25
26             <div class="row mt-3">
27                 <div class="col-lg-6 offset-lg-3">
28                     <div class="card" style="background-color: #FFF5E4;">
29                         <div class="card-body">
30                             <h5 class="card-title">Water level ghrapic</h5>
31                             <canvas id="myChart"></canvas>
32                             <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
33
34                             <script>
35                                 let dataPompa;
36                                 setInterval(tampilGrafik, 3000);

```

```

37
38 function tampilGrafik() {
39     $.ajax({
40         type: 'GET',
41         url: '<?=> base_url(relativePath: 'pompa/grafikpompa'); ?>',
42         data: {
43             functionName: 'getPompa'
44         },
45         success: function(response) {
46             //console.log(response)
47             let grafikPompa = JSON.parse(response)
48             let xpompa = collect(grafikPompa).map(function(item) {
49                 return item.pompa
50             }).all()
51             console.log(xpompa)
52             dataPompa = xpompa;
53         }
54     });
55 }
56
57 var categories = ["Pompa"];
58 var initialData = [dataPompa];
59 var updatedDataSet;
60 var ctx = document.getElementById("myChart");
61 var barChart = new Chart(ctx, {
62     type: "bar",
63     data: {
64         labels: categories,
65         datasets: [
66             {
67                 label: dataPompa + ' CM',
68                 data: initialData

```

```

69             data: initialData
70         }
71     ],
72     options: {
73         scales: {
74             y: {
75                 beginAtZero: true
76             }
77         }
78     }
79 });
80
81 function updateBarGraph(chart, data) {
82     console.log(chart.data);
83     chart.data.datasets.pop();
84     chart.data.datasets.push({
85         label: dataPompa + ' CM',
86         data: data,
87         backgroundColor: [
88             'rgba(0, 121, 73, 0.28)'
89         ],
90         borderColor: [
91             'rgba(0, 121, 73, 0.28)',
92         ],
93         borderWidth: 1
94     });
95     chart.update();
96 }
97
98 // update each 1 sec

```

```

97         setInterval(function() {
98             updatedDataSet = [dataPompa];
99             console.log(updatedDataSet);
100             updateBarGraph(barChart, updatedDataSet);
101         }, 3000);
102     }
103 }
104 </div>
105 </div>
106 </div>
107 </main>
108
109 <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
110 <!-- Load otomatis / realtime -->
111 <script type="text/javascript">
112     $(document).ready(function() {
113         setInterval(function() {
114             $("#cekpompa").load("<?>= base_url(relativePath: 'cekpompa'); ?>");
115         }, 1000);
116     });
117 </script>
118
119 <?> $this->endSection('content'); ?>
120

```

7. Membuat Tampilan Data Terbaru (cekpompa.php)

File ini ada di **app/Views/pompa/cekpompa.php**. Fungsinya menampilkan data level air terbaru.

```

cekpompa.php X
app > Views > pompa > cekpompa.php > ...
1 <?php
2 foreach ($dataPompa as $ds) :
3     echo $ds->pompa . '&nbsp;CM';
4 endforeach;
5
6 ?>

```

8. Membuat tampilan slide navbar

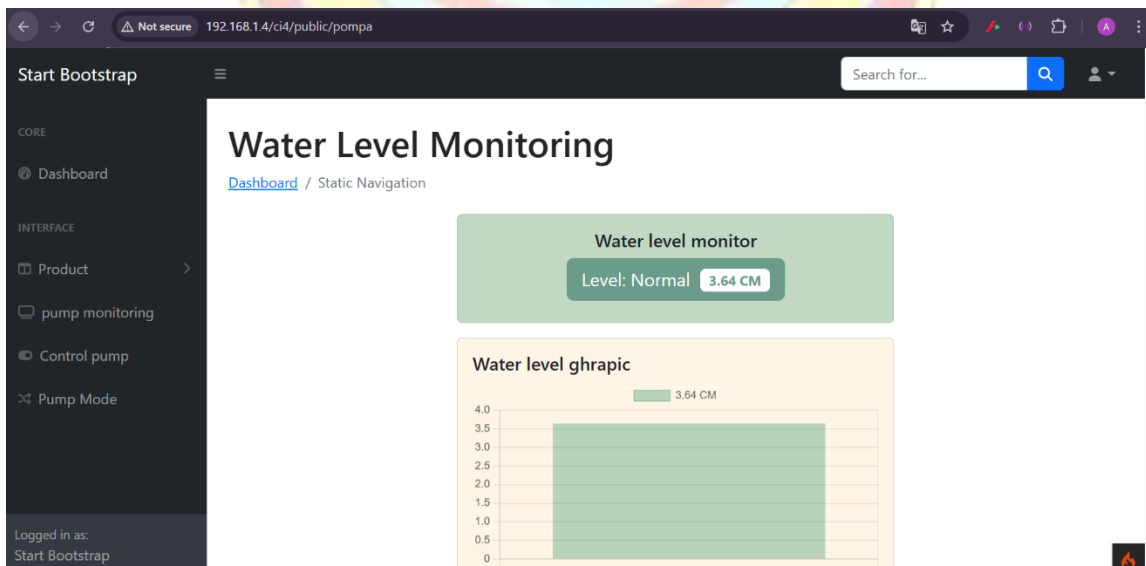
Tambahkan fungsi slide navbar pada **views/admin/layout/menu.php**

```

33 <a class="nav-link" href="<?>= base_url(relativePath: 'pompa'); ?>">
34     <div class="sb-nav-link-icon"><i class="fa-solid fa-tv"></i></div>
35     pump monitoring
36 </a>

```

9. Tampilan pada web



B. Langkah-langkah Menambahkan Fungsi Manual

Berikut adalah langkah-langkah yang dilakukan untuk menambahkan fungsi manual pada aplikasi:

1. Membuat Routes

Mendefinisikan Routes:

- Routes didefinisikan untuk mengarahkan permintaan ke metode yang sesuai di ManualController.
- Route manual akan menampilkan halaman manual relay.
- Route value-relay/(:num) akan menangani permintaan AJAX untuk mengubah status relay.

```
//fungsi relay pompa
$route->get(from: 'manual',to: 'ManualController::index');
$route->get(from: 'value-relay/(:num)',to: 'ManualController::ganti/$1');
```

2. Membuat Controller

Membuat Controller ManualController:

- Controller ini berisi dua metode utama: index() dan ganti().
- Metode index() digunakan untuk menampilkan halaman manual relay.
- Metode ganti() digunakan untuk mengubah status relay (ON/OFF) melalui permintaan AJAX.

```
ManualController.php 9+ X
app > Controllers > ManualController.php > PHP Intelephense > ManualController > ganti
1  <?php
2
3  namespace App\Controllers;
4
5  use App\Controllers\BaseController;
6  use CodeIgniter\HTTP\ResponseInterface;
7
8  class ManualController extends BaseController
9  {
10     public function __construct() {
11         $this->db = \Config\Database::connect();
12         $this->builder = $this->db->table('manualrelay');
13     }
14
15     public function index(): string
16     {
17         $data = [
18             'title' => "Kendali Relay + NodeMCU v3.0"
19         ];
20         return view('manual/manual',data: $data);
21     }
22
23     public function ganti(): void
24     {
25         if ($this->request->isAJAX()) {
26             $id = $this->request->getVar('id');
27             $db = \Config\Database::connect();
28             $query = $db->query("SELECT * FROM manualrelay WHERE id = $id");
29             $manual = $query->getRow()->manual;
30
31             if ($manual == 0) {
```

```

32         $data = [
33             'manual' => '1',
34         ];
35         $this->builder->where(key: 'id', value: $id);
36         $this->builder->update(set: $data);
37         $result = [
38             'success' => 'Relay OFF'
39         ];
40     } else {
41         $data = [
42             'manual' => '0',
43         ];
44         $this->builder->where(key: 'id', value: $id);
45         $this->builder->update(set: $data);
46         $result = [
47             'success' => 'Relay ON'
48         ];
49     }
50
51     echo json_encode(value: $result);
52 } else {
53     exit('404 Not Found!');
54 }
55 }
56 }

```

3. Membuat Tabel Database

Membuat Migrasi:

- File migrasi dibuat untuk mendefinisikan struktur tabel manualrelay. Tabel ini memiliki dua kolom, yaitu id (primary key, auto increment) dan manual (tipe VARCHAR dengan panjang 1).
- Migrasi dijalankan untuk membuat tabel manualrelay di database.

```

2025-03-10-040543_Manual.php 1 X
app > Database > Migrations > 2025-03-10-040543_Manual.php > PHP Intelephense > Manual > down
1  <?php
2
3  namespace App\Database\Migrations;
4
5  use CodeIgniter\Database\Migration;
6
7  0 references | 0 implementations
8  class Manual extends Migration
9  {
10     0 references | 0 overrides
11     public function up(): void
12     {
13         $this->forge->addField(fields: [
14             'id' => [
15                 'type' => 'INT',
16                 'constraint' => 5,
17                 'auto_increment' => true,
18                 'unsigned' => false,
19             ],
20             'manual' => [
21                 'type' => 'VARCHAR',
22                 'constraint' => '1',
23             ],
24         ]);
25         $this->forge->addKey(key: 'id', primary: true);
26         $this->forge->createTable(table: 'manualrelay');
27     }
28
29     0 references | 0 overrides
30     public function down(): void
31     {
32         $this->forge->dropTable(tableName: 'manualrelay');
33     }
34 }

```


- Jalankan migrasi dengan perintah di terminal: `php spark migrate`.

```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\xampp\htdocs\ci4> php spark migrate

CodeIgniter v4.5.7 Command Line Tool - Server Time: 2025-03-10 04:11:55 UTC+00:00

Running all new migrations...
Running: (App) 2025-03-10-040543_App\Database\Migrations\Manual
Migrations complete.
PS C:\xampp\htdocs\ci4>

```

4. Membuat Model

Membuat Model ManualModel:

- Model ini digunakan untuk berinteraksi dengan tabel `manualrelay` di database.
- Model ini mendefinisikan tabel yang digunakan, primary key, dan field yang diizinkan untuk diisi.

```

ManualModel.php X
app > Models > ManualModel.php > PHP Intelephense > ManualModel > $allowedFields

1  <?php
2
3  namespace App\Models;
4
5  use CodeIgniter\Model;
6
7  2 references | 0 implementations
8  class ManualModel extends Model
9  {
10     0 references
11     | protected $table          = 'manualrelay';
12     0 references
13     | protected $primaryKey     = 'id';
14     0 references
15     | protected $useAutoIncrement = true;
16     0 references
17     | protected $returnType      = 'object';
18     0 references
19     | protected $useSoftDeletes   = false;
20     0 references
21     | protected $protectFields   = true;
22     0 references
23     | protected $allowedFields    = ['manual'];
24 }

```

5. Membuat View

Membuat View `manual.php`:

- View ini menampilkan status relay dan tombol untuk mengubah status relay.
- Tombol akan berubah warna dan teks berdasarkan status relay (ON/OFF).

- [illegible]

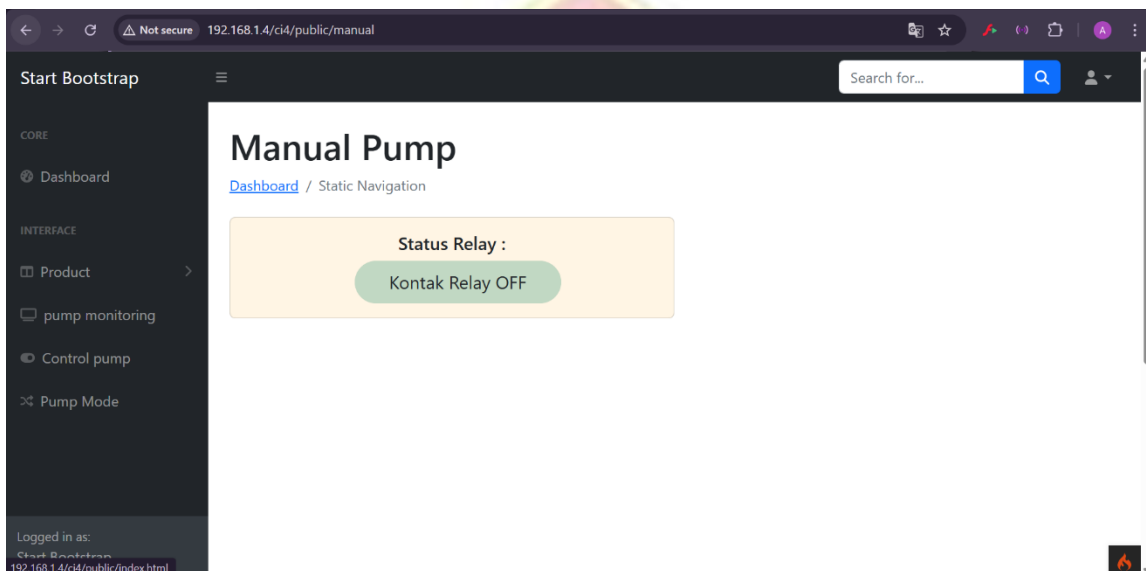
- ## 6. Membuat File untuk Membaca Status Relay

- File ini digunakan untuk membaca status relay dari database dan menampilkannya.
- File ini dapat diakses secara publik untuk mendapatkan status relay.

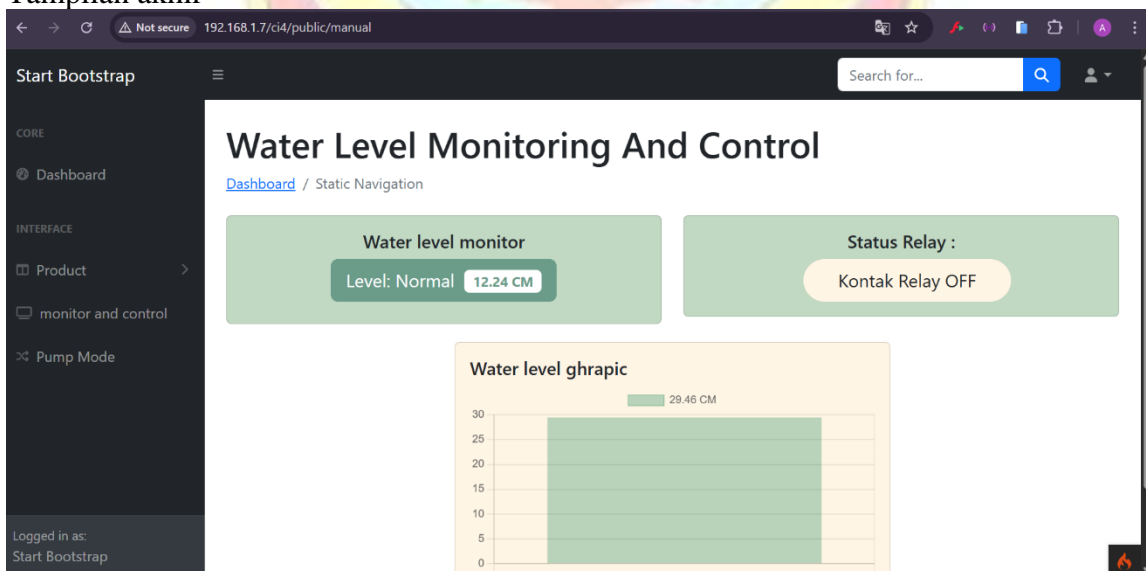
```
public > bacamanual.php > ...
1 <?php
2 $db = mysqli_connect(hostname: 'localhost', username: 'root', password: '', database: 'latihan');
3 $sql = mysqli_query($db, query: "SELECT * FROM manualrelay WHERE id=1");
4 $data = mysqli_fetch_array($sql);
5 $manual = $data['manual'];
6
7 echo $manual;
```

Simpan file. Pastikan database latihan sudah dibuat dan sesuai dengan migrasi.

7. Tampilan pada web



Tampilan akhir



C. Langkah-Langkah Menambahkan Fungsi Mode (Auto dan Manual)

1. Modifikasi Database

Membuat Tabel Mode

Tabel mode dibuat untuk menyimpan status mode yang aktif (Auto atau Manual). Berikut adalah struktur tabelnya:



```
2025-03-21-061316_Mode.php 1 X
app > Database > Migrations > 2025-03-21-061316_Mode.php > ...
1  <?php
2
3  namespace App\Database\Migrations;
4
5  use CodeIgniter\Database\Migration;
6
7  class Mode extends Migration
8  {
9      public function up(): void
10     {
11         $this->forge->addField(fields: [
12             'id' => [
13                 'type'          => 'INT',
14                 'constraint'    => 5,
15                 'auto_increment' => true,
16                 'unsigned'      => false,
17             ],
18             'mode' => [
19                 'type'          => 'VARCHAR',
20                 'constraint'    => '10', // Auto atau Manual
21             ],
22         ]);
23         $this->forge->addKey(key: 'id', primary: true);
24         $this->forge->createTable(table: 'mode');
25     }
26
27     public function down(): void
28     {
29         $this->forge->dropTable(tableName: 'mode');
30     }
31 }
```

2. Membuat Model untuk Tabel Mode

Model **ModeModel** dibuat untuk mengelola data mode. Berikut adalah kode untuk **ModeModel.php**:

```
ModeModel.php X
app > Models > ModeModel.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use CodeIgniter\Model;
6
7  class ModeModel extends Model
8  {
9      protected $table          = 'mode';
10     protected $primaryKey     = 'id';
11     protected $useAutoIncrement = true;
12     protected $returnType     = 'object';
13     protected $useSoftDeletes = false;
14     protected $protectFields  = true;
15     protected $allowedFields  = ['mode'];
16
17     public function getMode(): Builder
18     {
19         return $this->find(id: 1); // Ambil data mode dengan ID 1
20     }
21
22     public function updateMode($mode): bool
23     {
24         return $this->update(id: 1, row: ['mode' => $mode]); // Update mode dengan ID 1
25     }
26 }
```

3. Membuat Controller untuk Mengelola Mode

Controller **ModeController** dibuat untuk menangani perubahan mode. Berikut adalah kode untuk **ModeController.php**:

```
ModeController.php X
app > Controllers C:\xampp\htdocs\ci4\app\Controllers\ModeController.php (preview)
1  <?php
2
3  namespace App\Controllers;
4
5  use App\Controllers\BaseController;
6  use App\Models\ModeModel;
7
8  class ModeController extends BaseController
9  {
10     protected $modeModel;
11
12     public function __construct()
13     {
14         $this->modeModel = new ModeModel();
15     }
16 }
```



```

0 references | 0 overrides
17 public function index(): string
18 {
19     $data = [
20         'title' => 'Mode Pengaturan',
21         'mode' => $this->modeModel->getMode(),
22     ];
23     return view(name: 'mode/index', data: $data);
24 }
25
0 references | 0 overrides
26 public function switchMode(): void
27 {
28     if ($this->request->isAJAX()) {
29         $mode = $this->request->getVar(index: 'mode'); // Ambil mode dari request
30         $this->modeModel->updateMode(mode: $mode); // Update mode
31
32         echo json_encode(value: ['success' => 'Mode berhasil diubah menjadi ' . $mode]);
33     } else {
34         exit('404 Not Found');
35     }
36 }
37 }

```

4. Menambahkan Routes untuk Mode

Routes ditambahkan untuk mengakses fungsi mode. Berikut adalah kode yang ditambahkan ke **routes.php**:

```

40
41 //fungsi mode auto atau manual
42 $routes->get(from: 'mode', to: 'ModeController::index'); // Halaman mode
43 $routes->post(from: 'switch-mode', to: 'ModeController::switchMode'); // API untuk mengubah mode

```

5. Membuat View untuk Mode

View **mode/index.php** dibuat untuk menampilkan dan mengubah mode. Berikut adalah kode untuk view tersebut:

```

index.php x
app > Views > mode > index.php > ...
1 <?= $this->extend('admin/layout/template'); ?>
2 <?= $this->section('content'); ?>
3 <div id="layoutSidenav_content">
4     <main>
5         <div class="container-fluid px-4">
6             <h1 class="mt-4">Pump Mode</h1>
7             <ol class="breadcrumb mb-4">
8                 <li class="breadcrumb-item"><a href="<?= base_url(relativePath: 'dashboard'); ?>">Dashboard</a></li>
9                 <li class="breadcrumb-item active">Static Navigation</li>
10            </ol>
11            <!-- <div class="row">
12                <div class="col-lg-6"> -->
13                    <div class="card mb-4" style="background-color: #C1D8C3;">
14                        <div class="card-body" style="text-align: center;">
15                            <h5 class="card-title"><i class="bi-thermometer-sun text-white"></i> Pilih Mode</h5>
16                            <p class="card-text">
17                                Mode saat ini: <strong><?= $mode->mode; ?></strong>
18                            </p>
19                            <button id="switchMode" class="btn btn-lg mb-2 text white" style="background-color: #6A9C88; border-color:
20                                Ubah ke <?= ($mode->mode == 'Auto') ? 'Manual' : 'Auto'; ?>
21                            </button>
22                        </div>
23                    </div>
24                </div>
25            </div>
26        </div>
27    </main>
28 </div>

```

```

30 <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
31 <script>
32     $(document).ready(function() {
33         $('#switchMode').click(function() {
34             const newMode = ($(this).text().trim() === 'Ubah ke Manual') ? 'Manual' : 'Auto';
35
36             $.ajax({
37                 type: 'POST',
38                 url: '<?= base_url(relativePath: "switch-mode"); ?>',
39                 data: {
40                     mode: newMode
41                 },
42                 dataType: 'json',
43                 success: function(response) {
44                     if (response.success) {
45                         alert(response.success);
46                         location.reload(); // Reload halaman setelah mode diubah
47                     }
48                 }
49             });
50         });
51     });
52 </script>
53
54 <?= $this->endSection('content'); ?>

```

6. Membuat Endpoint bacamode.php di Server

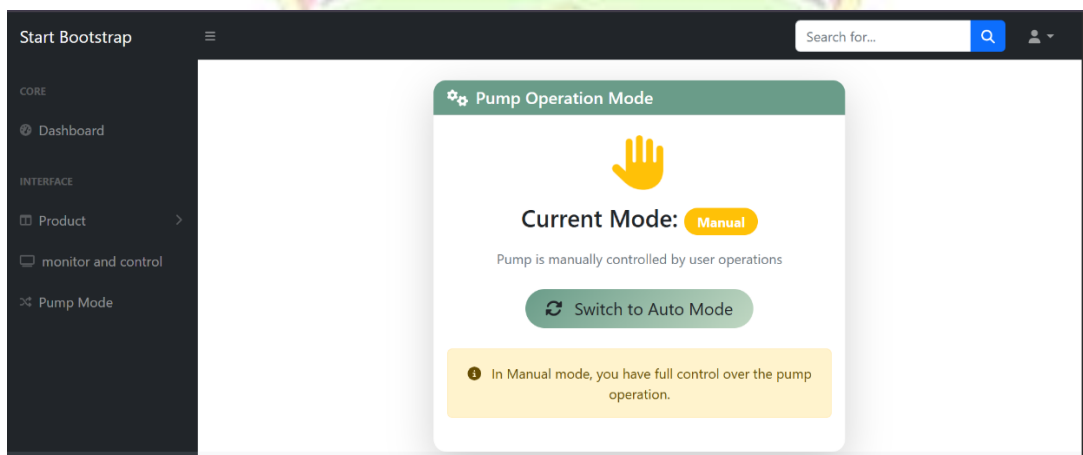
File **bacamode.php** dibuat untuk mengembalikan status mode. Berikut adalah kodenya:

```

public > bacamode.php > ...
1 <?php
2 $db = mysqli_connect(hostname: 'localhost', username: 'root', password: '', database: 'latihan');
3 $sql = mysqli_query(mysql: $db, query: "SELECT * FROM mode WHERE id=1");
4 $data = mysqli_fetch_array(result: $sql);
5 $mode = $data['mode'];
6
7 echo $mode;
8 ?>

```

7. Tampilan pada web

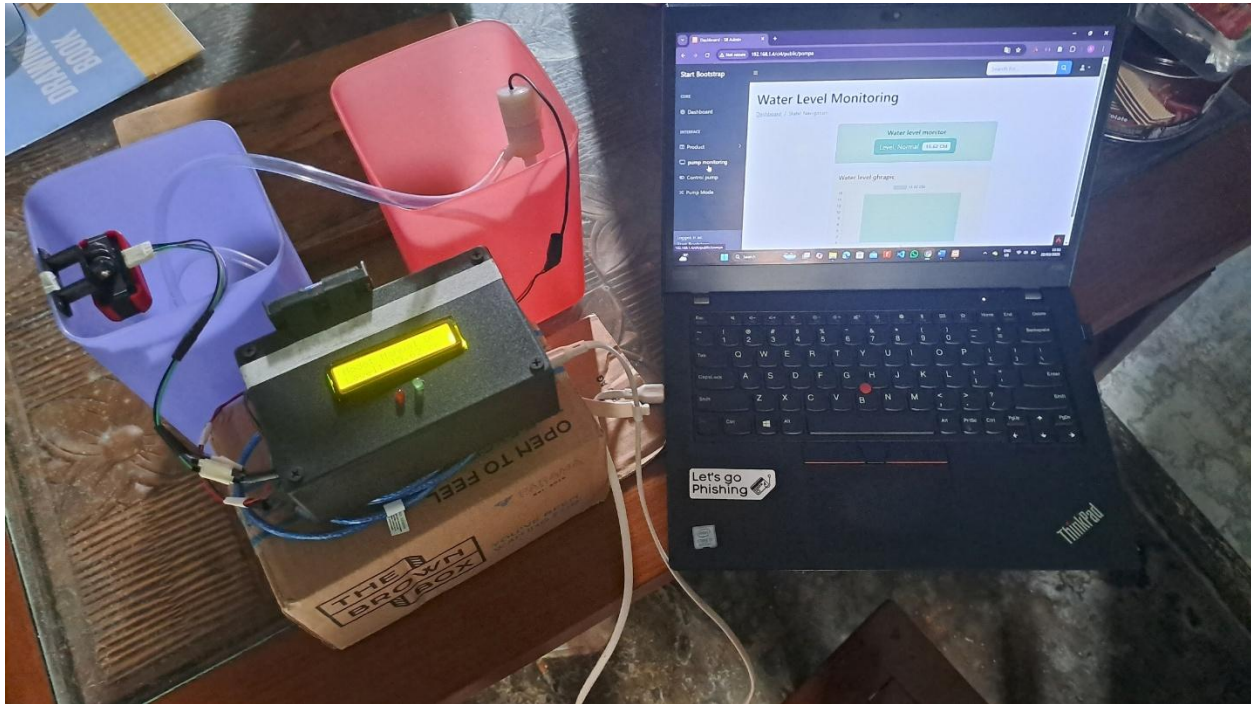


BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil

Hasil perancangan dari kendali smart home berbasis IoT menggunakan Laptop dapat dilihat pada gambar di bawah ini:

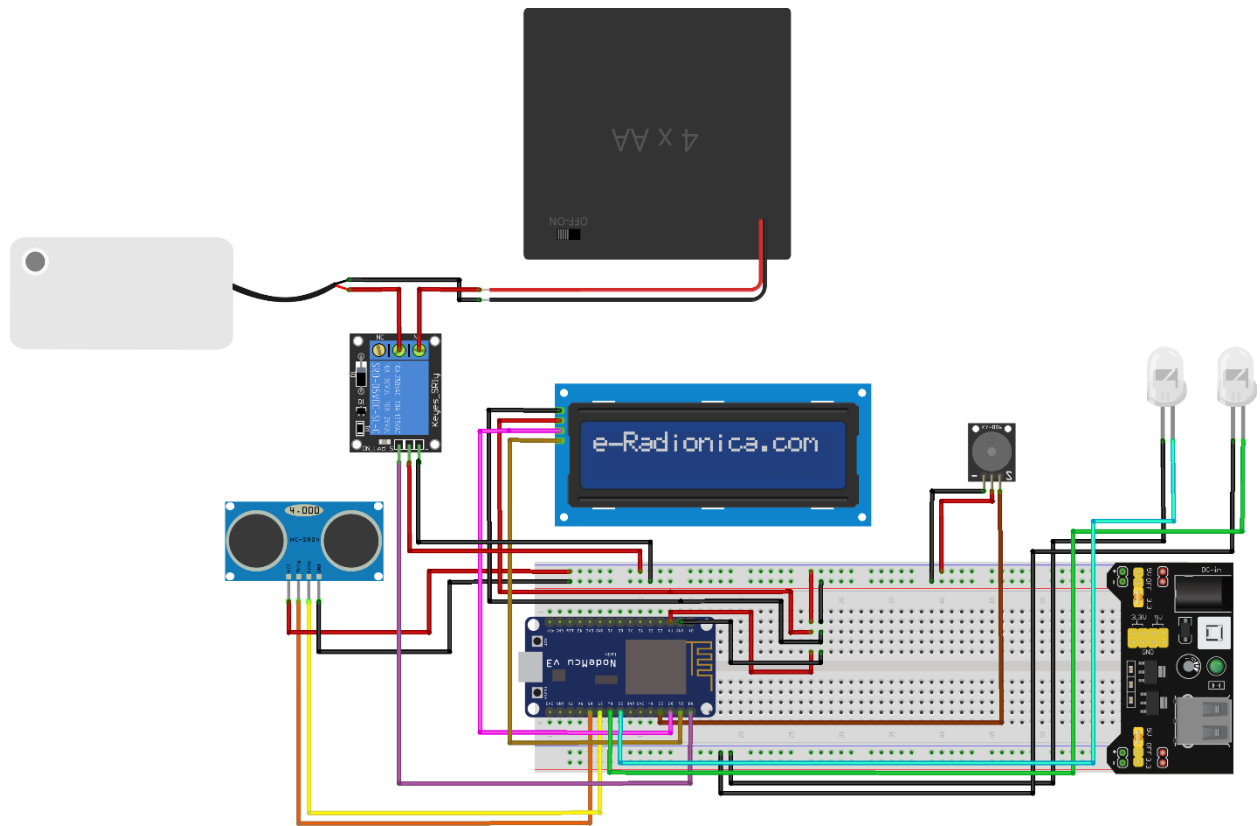


4.2. Hasil Kegiatan

Dari kegiatan yang dilakukan, dapat diperoleh hasil dari perangkat lunak.

4.2.1. Perangkat Lunak

Rangkaian perangkat lunak dapat dilihat pada gambar dibawah ini:



fritzing

Keterangan:

- ✓ trig = D8
 - ✓ echo = D7
 - ✓ relay = D0
 - ✓ sda = D2
 - ✓ scl = D1
 - ✓ buzzer = D3
-
- lcd12c :
 - vcc = vu (esp)
 - gnd = gnd (esp)
 - sda = D2 (esp)
 - scl = D1 (esp)
 - ultrasonic :
 - vcc = + (psu)
 - gnd = - (psu)
 - trig = D8 (esp)
 - echo = D7 (esp)

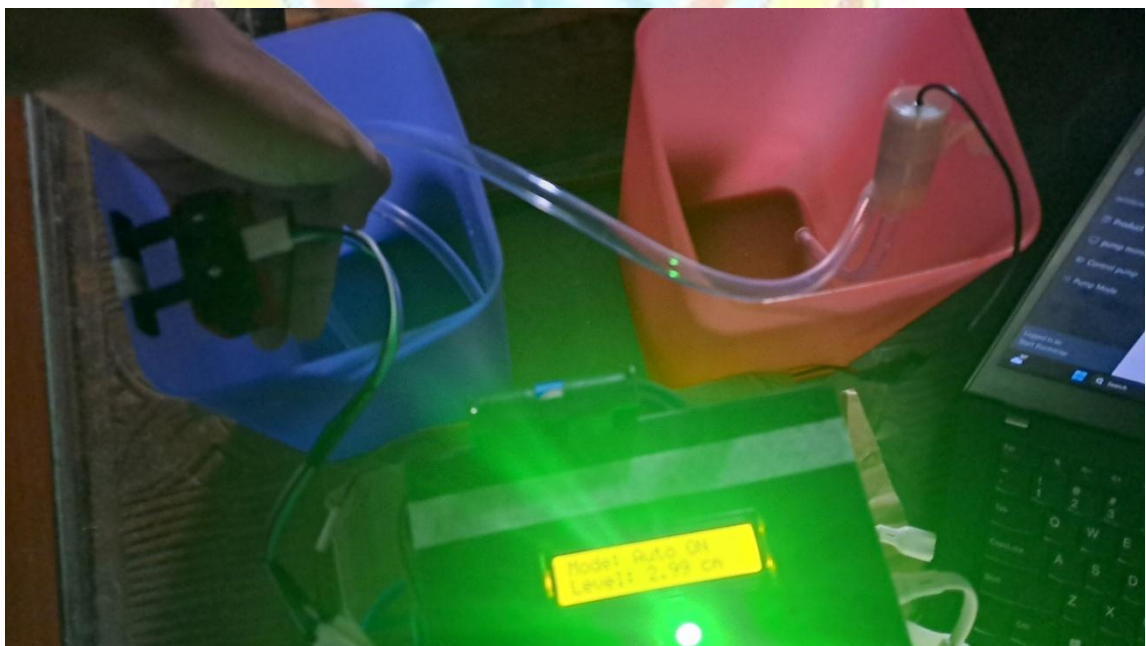
- relay :
 - DO+ = + (psu)
 - DO- = - (psu)
 - CH1 = D0 (esp)
- buzzer :
 - vcc = + (psu)
 - gnd = - (psu)
 - 1/0 = D3 (esp)
- led1 :
 - (led) = - (psu)
 - + (led) = D5 (esp)
- led2 :
 - (led2) = - (psu)
 - + (led2) = D6 (esp)
- vu (esp) = + (psu)
- gnd (esp) = - (psu)

Adapun keterangan gambar diatas, Sketsa rangkaian monitor dan kendali untuk sensor pada komponen. Gambar diatas menggunakan aplikasi Fritzing.

4.3. Hasil Pengujian

Adapun hasil pengujian yang telah dilakukan pada alat guna mengecek apakah semua sudah berjalan seperti yang diharapkan, sehingga dapat digunakan.

4.3.1. Hasil rangkaian



4.3.2. Program sistem

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Network credentials
const char* ssid = "IIL";
const char* password = "yusuffattah";
const char* host = "192.168.1.4"; // IP server XAMPP (adjust if needed)

// Pin definitions
#define TRIGGER_PIN D8 // Ultrasonic Trigger Pin
#define ECHO_PIN D7 // Ultrasonic Echo Pin
#define BUZZER_PIN D3 // Buzzer Pin
#define RELAY_PIN D0 // Relay Pin
#define PIN_LED D6 // LED 1 Pin
#define PIN_LED2 D5 // LED 2 Pin

// LCD I2C setup (adjust 0x27 if needed, check with I2C Scanner)
LiquidCrystal_I2C lcd(0x27, 16, 2); // 16 columns, 2 rows

// Function prototypes
float readDistance();

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

    // Initialize pins
    pinMode(TRIGGER_PIN, OUTPUT);
    pinMode(ECHO_PIN, INPUT);
    pinMode(BUZZER_PIN, OUTPUT);
    pinMode(RELAY_PIN, OUTPUT);
    pinMode(PIN_LED, OUTPUT);
    pinMode(PIN_LED2, OUTPUT);

    // Initialize LCD
    lcd.begin();
    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("Water Tank");

    // Connect to WiFi
    WiFi.hostname("NodeMCU");
```

```

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
    lcd.setCursor(0, 1);
    lcd.print("Connecting...");
}

Serial.println("\nWiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
lcd.setCursor(0, 1);
lcd.print("WiFi Connected ");
delay(1000);
}

void loop() {
    WiFiClient client;
    HTTPClient http;

    // Read distance
    float distance = readDistance();

    // Check mode and manual control from server
    if (WiFi.status() == WL_CONNECTED) {
        // Get current mode from server
        String LinkMode = "http://" + String(host) +
"/ci4/public/bacamode.php"; // Endpoint untuk membaca mode
        http.begin(client, LinkMode);
        http.GET();
        String mode = http.getString();
        http.end();

        // Get manual control status from server
        String LinkRelay = "http://" + String(host) +
"/ci4/public/bacamanual.php";
        http.begin(client, LinkRelay);
        http.GET();
        String statusrelay = http.getString();
        http.end();

        Serial.println("Mode: " + mode);
        Serial.println("Relay Status: " + statusrelay);
    }
}

```

```

// Mode control logic
if (mode == "Manual") { // Manual Mode
    if (statusrelay == "0") { // Manual ON
        digitalWrite(RELAY_PIN, LOW); // Pump ON
        lcd.setCursor(0, 0);
        lcd.print("Mode: Manual ON ");
    }
    else if (statusrelay == "1") { // Manual OFF
        digitalWrite(RELAY_PIN, HIGH); // Pump OFF
        lcd.setCursor(0, 0);
        lcd.print("Mode: Manual OFF ");
    }
}
else if (mode == "Auto") { // Auto Mode
    lcd.setCursor(0, 0);
    lcd.print("Mode: Auto ON ");

    // Automatic control based on distance
    if (distance >= 15) {
        digitalWrite(RELAY_PIN, LOW); // Turn ON pump
        digitalWrite(PIN_LED2, HIGH);
        digitalWrite(PIN_LED, LOW);
        // Buzzer sequence
        tone(BUZZER_PIN, 1000); // 1000 Hz
        delay(500);
        noTone(BUZZER_PIN);
        delay(100);
        tone(BUZZER_PIN, 0); // Silent (or 293 Hz for D4)
        delay(500);
        noTone(BUZZER_PIN);
        delay(100);
        tone(BUZZER_PIN, 1000); // 1000 Hz
        delay(500);
        noTone(BUZZER_PIN);
        delay(1000);
    }
    else if (distance <= 4) {
        digitalWrite(RELAY_PIN, HIGH); // Turn OFF pump
        digitalWrite(PIN_LED2, LOW);
        digitalWrite(PIN_LED, HIGH);
    }
}

// Send distance data to server

```

```

    String Link = "http://" + String(host) + "/ci4/public/oleh-data/" +
String(distance);
    if (client.connect(host, 80)) {
        http.begin(client, Link);
        http.GET();
        http.end();
    } else {
        Serial.println("Connection to server failed");
    }
}

// Display distance on LCD
lcd.setCursor(0, 1);
lcd.print("Level: ");
lcd.print(distance);
lcd.print(" cm");

// Display on Serial
Serial.print("Distance: ");
Serial.print(distance);
Serial.println(" cm");

delay(1000); // Main loop delay
}

// Function to read distance
float readDistance() {
    digitalWrite(TRIGGER_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIGGER_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIGGER_PIN, LOW);

    long duration = pulseIn(ECHO_PIN, HIGH);
    float dist = duration * 0.034 / 2;

    return dist;
}

```

Program ini dirancang untuk mengukur jarak menggunakan sensor ultrasonik dan memberikan peringatan berupa nyala LED dan buzzer jika jarak yang terdeteksi berada di bawah atau di atas nilai tertentu. Data jarak dan mode juga ditampilkan pada LCD dan serial monitor.

1. Inisialisasi Library dan Variabel

- Library:
 - ESP8266WiFi.h: Untuk menghubungkan NodeMCU ke jaringan WiFi.
 - ESP8266HTTPClient.h: Untuk mengirim dan menerima data HTTP ke/dari server.
 - Wire.h: Untuk komunikasi I2C.
 - LiquidCrystal_I2C.h: Untuk mengontrol layar LCD dengan antarmuka I2C.
- Deklarasi Objek LCD:
 - LCD diinisialisasi dengan alamat I2C `'0x27'` dan ukuran layar 16 kolom x 2 baris.
- Kredensial Jaringan:
 - ssid: Nama jaringan WiFi ("IIL").
 - password: Kata sandi WiFi ("yusuffattah").
 - host: Alamat IP server (192.168.1.4) yang digunakan untuk komunikasi HTTP.
- Definisi Pin:
 - TRIGGER_PIN (D8): Pin untuk mengirimkan pulsa ultrasonik.
 - ECHO_PIN (D7): Pin untuk menerima pulsa pantulan ultrasonik.
 - BUZZER_PIN (D3): Pin untuk buzzer sebagai alarm.
 - RELAY_PIN (D0): Pin untuk mengontrol relay (pompa air).
 - PIN_LED (D6): Pin untuk LED 1 sebagai indikator.
 - PIN_LED2 (D5): Pin untuk LED 2 sebagai indikator.
- Variabel:
 - distance: Jarak yang diukur oleh sensor ultrasonik dalam satuan cm.

2. Fungsi setup()

Fungsi ini mengatur konfigurasi awal perangkat:

- Serial Monitor:
 - Inisialisasi komunikasi serial dengan baud rate 9600 untuk debugging.
- Mode Pin:
 - `'TRIGGER_PIN'` diatur sebagai OUTPUT untuk mengirimkan sinyal ultrasonik.
 - `'ECHO_PIN'` diatur sebagai INPUT untuk menerima sinyal pantulan ultrasonik.
 - `'BUZZER_PIN'`, `'RELAY_PIN'`, `'PIN_LED'`, dan `'PIN_LED2'` diatur sebagai OUTPUT untuk kontrol perangkat.
- Inisialisasi LCD:
 - `'lcd.begin()'`: Memulai komunikasi dengan LCD.
 - `'lcd.backlight()'`: Menyalakan lampu latar LCD.

- Menampilkan teks "Water Tank" di baris pertama LCD.
- Koneksi WiFi:
 - Menghubungkan NodeMCU ke jaringan WiFi menggunakan SSID dan password yang telah ditentukan.
 - Selama proses koneksi, menampilkan teks "Connecting..." di baris kedua LCD.
 - Setelah terhubung, menampilkan "WiFi Connected" di LCD dan alamat IP di Serial Monitor.

3. Fungsi loop()

Fungsi ini berjalan berulang kali untuk membaca data sensor, mengontrol perangkat, dan berkomunikasi dengan server.

a. Baca Jarak:

Memanggil fungsi `readDistance()` untuk mengukur jarak menggunakan sensor ultrasonik. Hasil jarak disimpan dalam variabel `distance`.

b. Komunikasi dengan Server:

1. Baca Mode Operasi:

- Mengirim permintaan HTTP ke server untuk membaca mode operasi (Manual/Auto).
- Mode disimpan dalam variabel `mode`.

2. Baca Status Relay (Manual Control):

- Mengirim permintaan HTTP ke server untuk membaca status relay (ON/OFF).
- Status disimpan dalam variabel `statusrelay`.

3. Logika Kontrol:

- Mode Manual:
 - Jika `statusrelay == "0"`, relay diaktifkan (pompa ON) dan menampilkan "Mode: Manual ON" di LCD.
 - Jika `statusrelay == "1"`, relay dinonaktifkan (pompa OFF) dan menampilkan "Mode: Manual OFF" di LCD.
- Mode Auto:
 - Menampilkan "Mode: Auto ON" di LCD.
 - Jika jarak (`distance`) ≥ 15 cm:
 - Relay diaktifkan (pompa ON).
 - LED 2 menyala, LED 1 mati.
 - Buzzer berbunyi dengan pola tertentu sebagai alarm.
 - Jika jarak (`distance`) ≤ 4 cm:
 - Relay dinonaktifkan (pompa OFF).
 - LED 1 menyala, LED 2 mati.

4. Kirim Data Jarak ke Server:

- Mengirim data jarak ke server menggunakan permintaan HTTP GET.

- c. Tampilkan Data di LCD dan Serial Monitor:
 - Menampilkan jarak (`distance`) di baris kedua LCD dengan format "Level: [nilai] cm".
 - Menampilkan jarak di Serial Monitor untuk debugging.
- d. Delay:
 - Delay 1000 ms (1 detik) sebelum mengulang siklus berikutnya.

4. Fungsi `readDistance()`

Fungsi ini mengukur jarak menggunakan sensor ultrasonik:

1. Kirim Pulsa Ultrasonik:

- `digitalWrite(TRIGGER_PIN, LOW)` untuk memastikan pulsa bersih.
- `digitalWrite(TRIGGER_PIN, HIGH)` untuk mengirimkan pulsa selama 10 mikrodetik.
- `digitalWrite(TRIGGER_PIN, LOW)` untuk menghentikan pulsa.

2. Baca Durasi:

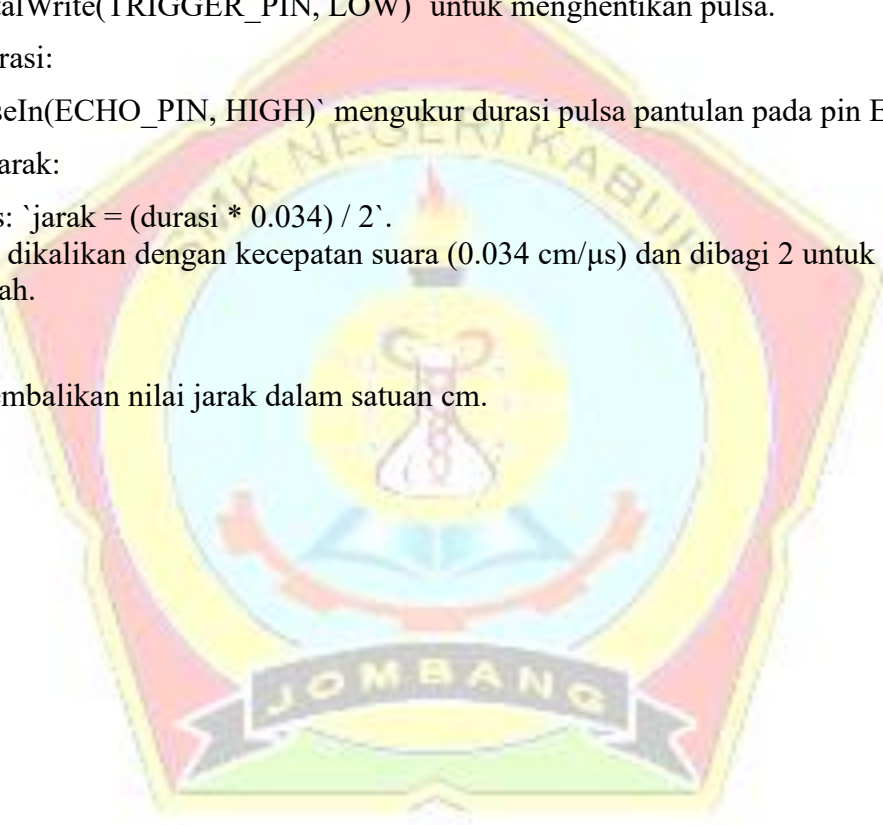
- `pulseIn(ECHO_PIN, HIGH)` mengukur durasi pulsa pantulan pada pin Echo.

3. Hitung Jarak:

- Rumus: `jarak = (durasi * 0.034) / 2`.
- Durasi dikalikan dengan kecepatan suara (0.034 cm/ μ s) dan dibagi 2 untuk waktu tempuh satu arah.

4. Return:

- Mengembalikan nilai jarak dalam satuan cm.



BAB III

PENUTUP

A. Kesimpulan

Kesimpulan dari proyek sistem kendali water level pada tandon air ini adalah bahwa sistem berhasil mengotomatisasi pengendalian ketinggian air dengan menggunakan sensor ultrasonik sebagai alat pengukur utama, yang dipadukan dengan LCD untuk menampilkan informasi jarak, serta LED dan buzzer sebagai indikator peringatan. Sistem ini mampu mendeteksi ketinggian air secara akurat dan memberikan respons otomatis dengan menyalakan atau mematikan pompa berdasarkan jarak yang terukur, sehingga mencegah tandon meluap atau kehabisan air. Proyek ini menunjukkan bahwa teknologi berbasis mikrokontroler seperti ESP8266 dapat digunakan secara efisien untuk menciptakan solusi otomatisasi sederhana namun efektif. Selain itu, implementasi perangkat keras yang ekonomis dan perangkat lunak yang fleksibel memungkinkan sistem ini untuk diadaptasi ke berbagai skenario lain, seperti pengelolaan tangki air rumah tangga atau industri kecil. Dengan memanfaatkan sistem ini, efisiensi air dan energi dapat ditingkatkan, serta risiko kerusakan akibat kelalaian dapat diminimalkan.

B. Saran

untuk pengembangan lebih lanjut dari proyek sistem kendali water level pada tandon air ini adalah sebagai berikut.

1. peningkatan Akurasi Sensor

Penggunaan sensor ultrasonik dapat dipengaruhi oleh faktor-faktor seperti suhu dan kelembapan, yang dapat memengaruhi akurasi pengukuran jarak. Oleh karena itu, penggunaan sensor lain seperti sensor tekanan atau sensor kapasitif dapat dipertimbangkan untuk mendapatkan hasil yang lebih akurat.

2. Integrasi dengan IoT

Mengintegrasikan sistem dengan teknologi IoT (Internet of Things) dapat memungkinkan pemantauan jarak air dan status sistem secara real-time melalui aplikasi smartphone atau web. Hal ini akan memudahkan pengguna untuk mengontrol dan memantau tandon air dari jarak jauh, meningkatkan kenyamanan dan efisiensi.

3. Peningkatan Daya Tahan dan Keandalan Sistem

Menambahkan proteksi terhadap kegagalan sistem, seperti deteksi jika sensor atau pompa gagal berfungsi, serta memberi peringatan kepada pengguna melalui buzzer atau notifikasi IoT, dapat meningkatkan keandalan dan keamanan sistem.

4. Penggunaan Energi Terbarukan

Untuk meningkatkan keberlanjutan proyek ini, sistem ini bisa dikombinasikan dengan sumber daya energi terbarukan, seperti panel surya, untuk mengurangi ketergantungan pada sumber daya listrik konvensional.

5. Peningkatan Antarmuka Pengguna (UI)

Untuk mempermudah interaksi dengan sistem, tampilan LCD dapat diperbaharui dengan menampilkan lebih banyak informasi, seperti status pompa, level air secara real-time, atau indikator kegagalan sensor.



DAFTAR PUSTAKA

tiara rannu zefanya sarira/september2021

https://repository.poliupg.ac.id/id/eprint/2794/1/TUGAS%20AKHIR_ANDI%20BAU%20KHAE%20MUHARDIKA%20LOLOGAU_TIARA%20RANNU%20ZEFANYA%20SARIRA%20%20%20wtr.pdf

DADAN NURDIN BAGENDA, ST., MT./2024

Internet Of Things (IoT) Device base on Esp8266 (NodeMCU), DNB 2024

Farid Andesta Pratama 23063007, Lutfi Jenang HS 23063034/19 Desember 2024

<https://www.scribd.com/document/829555102/Laporan-proyek-akhir-kelompok-3-23063007-23063034>

Deepseek/maret 2025

<https://chat.deepseek.com/>

Grok 3 ai/maret 2025

<https://grok.com/?referrer=website>

