



**RANCANG BANGUN PENGISIAN GALON OTOMATIS  
MENGUNAKAN SENSOR QR CODE GM-67, FLOWMETER YF-S201,  
DAN WEBSITE BERBASIS ESP 32**

**LAPORAN TUGAS AKHIR**

Diajukan Sebagai Syarat Menyelesaikan Pendidikan Pada  
Program Studi Sarjana Terapan Teknik Listrik Industri  
Departemen Teknologi Industri Sekolah Vokasi  
Universitas Diponegoro

Oleh :

Muhammad Rizky Akbar

40040621650039

**PROGRAM STUDI SARJANA TERAPAN TEKNIK LISTRIK INDUSTRI  
DEPARTEMEN TEKNOLOGI INDUSTRI  
SEKOLAH VOKASI  
UNIVERSITAS DIPONEGORO  
SEMARANG**

**2026**

**HALAMAN PERSETUJUAN**

**LAPORAN TUGAS AKHIR**


**RANCANG BANGUN PENGISIAN GALON OTOMATIS  
MENGUNAKAN SENSOR QR CODE GM-67, FLOWMETER YF-S201,  
DAN WEBSITE BERBASIS ESP 32**

Diajukan oleh:

Nama : Muhammad Rizky Akbar  
NIM : 40040621650039

TELAH DISETUJUI DAN DITERIMA DENGAN BAIK OLEH :

Dosen Pembimbing

  
Yuniarto S.T., M.T.  
NIP. 197106151998021001

Tanggal .....

Mengetahui,

Ketua Program Studi Sarjana Terapan Teknik Listrik Industri  
Departemen Teknik Industri Sekolah Vokasi  
Universitas Diponegoro

  
Arkhan Subari, ST.M.Kom  
NIP. 197710012001121002

Tanggal .....

**HALAMAN PENGESAHAN**

**LAPORAN TUGAS AKHIR**

**RANCANG BANGUN PENGISIAN GALON OTOMATIS  
MENGUNAKAN SENSOR QRCODE GM-67, FLOWMETER YF-S201,  
DAN WEBSITE BERBASIS ESP 32**

Diajukan oleh: Muhammad Rizky Akbar

Telah Dipertahankan di depan Dewan Penguji pada

Hari : Selasa  
Tanggal : 9 Juni 2026

Penguji 1



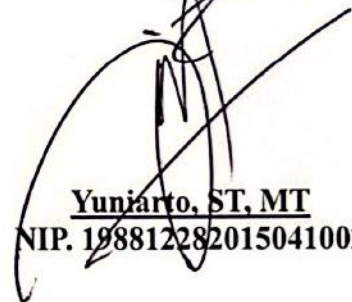
Dista Yoel Tadeus ST, MT  
NIP. 198908202019031012

Penguji II



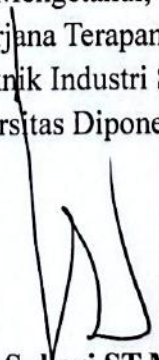
Fakhruddin Mangkusasmito, ST, MT  
NIP. 197106151998021001

Penguji III



Yuniarto, ST, MT  
NIP. 198812282015041002

Mengetahui,  
Ketua Program Studi Sarjana Terapan Teknik Listrik Industri  
Departemen Teknik Industri Sekolah Vokasi  
Universitas Diponegoro



Arkhan Subari, ST, M, Kom  
NIP. 197710012001121002

## **SURAT PERNYATAAN KEASLIAN**


Saya bertanda tangan dibawah ini:

Nama : Muhammad Rizky Akbar  
NIM : 40040621650039  
Program Studi : Sarjana Terapan Teknik Listrik Industri Departemen  
Teknologi Industri Sekolah Vokasi Universitas  
Diponegoro  
Judul Tugas Akhir : RANCANG BANGUN PENGISIAN GALON  
OTOMATIS MENGGUNAKAN SENSOR QRCODE  
GM-67, FLOWMETER YF-S201, DAN WEBSITE  
BERBASIS ESP 32

Dengan ini saya menyatakan bahwa judul tugas akhir ini belum pernah diajukan sebelumnya untuk mendapatkan gelar keahlian di sebuah perguruan tinggi. Sejauh pengetahuan saya, tidak ada karya atau pendapat yang pernah ditulis dan diterbitkan oleh orang lain kecuali yang saya rujuk secara tertulis dalam naskah ini dan tercantum dalam daftar pustaka.

Apabila dikemudian hari terbukti plagiat dalam tugas akhir ini, maka saya bersedia mendapat sanksi sesuai peraturan Mendiknas RI. No.17 Tahun 2010 dan Undang-Undang yang berlaku.

Semarang, 9 Juni 2026  
Yang membuat pernyataan

  
**Muhammad Rizky Akbar**  
**40040621650039**

## HALAMAN PERSEMBAHAN

Tugas akhir ini saya persembahkan untuk :

1. Kedua Orang Tua yang telah sabar kepada penulis serta tak luput untuk selalu memberikan do'a yang terbaik dan dukungan kepada penulis untuk selalu semangat dalam menjalani masa pendidikan hingga tuntas.
2. Bapak Arkhan Subari S.T.,M.Kom Selaku Ketua Program Studi Teknik Listrik Industri Sekolah Vokasi Universitas Diponegoro.
3. Bapak Ir. H. Saiful Manan S.T.,M.T. Selaku dosen pembimbing pertama yang telah membimbing, membantu penulis serta memberikan pengarahan dalam membuat Tugas Akhir
4. Bapak Yuniarto S.T., M.T. Selaku dosen pembimbing pengganti yang juga telah membantu penulis serta memberikan pengarahan dalam menyelesaikan Tugas Akhir ini.
5. Seluruh dosen dan tenaga pendidik Program Studi Teknik Listrik Industri Sekolah Vokasi Universitas Diponegoro.
6. Seluruh pihak yang telah banyak membantu dalam menyelesaikan Tugas Akhir ini yang tidak dapat penulis sebutkan satu persatu.
7. Serta, tidak lupa untuk yang paling berharga, diri saya sendiri. Terima kasih sudah bertahan sejauh ini, sudah tetap berdiri meski berkali-kali hampir jatuh, sudah memilih untuk melanjutkan di hari-hari ketika berhenti terasa jauh lebih mudah daripada terus melangkah. Ada malam-malam yang panjang, pikiran yang berat, langkah yang tersendat, dan hidup yang hampir berantakan di segala sisinya, namun di tengah semua itu, kamu tidak menyerah, kamu tetap hadir, kamu tetap berjuang, bahkan ketika tak ada yang melihat sekalipun. Dan kini, melalui selesainya tulisan ini, kita berhasil menuntaskan apa yang sudah kita mulai bersama, ini bukan sekadar sebuah karya yang selesai, ini adalah bukti nyata bahwa kamu mampu, bahwa kamu cukup, dan bahwa kamu layak untuk bangga pada dirimu sendiri.

## ABSTRAK

Proses pengisian air galon pada depot air minum umumnya masih dilakukan secara manual sehingga berpotensi menimbulkan ketidaktepatan volume pengisian, pemborosan air, serta keterbatasan dalam pencatatan dan pemantauan data penggunaan. Penelitian ini bertujuan untuk merancang dan membangun sistem pengisian galon otomatis berbasis Internet of Things (IoT) menggunakan mikrokontroler ESP32 sebagai pusat kendali yang mengintegrasikan sensor QR Code Scanner GM-67, sensor flowmeter YF-S201, dan sensor ultrasonik HC-SR04. Metode penelitian meliputi perancangan perangkat keras, perancangan perangkat lunak berbasis website, serta pengujian fungsional sistem secara menyeluruh. Proses kerja sistem diawali dengan pemindaian QR Code yang memuat token transaksi yang telah diterbitkan oleh sistem setelah pengguna memilih volume air dan melakukan konfirmasi pembelian melalui website, guna memverifikasi status token apakah masih aktif atau telah digunakan serta mengambil informasi volume pengisian yang telah ditentukan sebelumnya, dilanjutkan dengan validasi keberadaan galon menggunakan sensor ultrasonik pada rentang jarak 1 hingga 3 cm. Setelah galon terdeteksi pada posisi yang sesuai, pompa air DC diaktifkan melalui modul relay dan volume air yang mengalir dihitung berdasarkan akumulasi pulsa sensor flowmeter menggunakan faktor kalibrasi optimal  $K = 7,4$  hingga volume target tercapai, kemudian pompa dimatikan secara otomatis. Hasil pengujian menunjukkan bahwa sistem mampu melakukan pengisian galon secara otomatis pada tiga variasi volume yaitu 2.000 mL, 5.000 mL, dan 15.000 mL dengan persentase error terkecil. Sistem juga berhasil menerapkan mekanisme validasi token satu kali penggunaan (*one-time use*) sehingga token yang telah digunakan secara otomatis ditolak oleh sistem. Data hasil transaksi dikirimkan ke website monitoring secara *real-time* untuk keperluan pencatatan dan pengawasan operasional.

***Kata kunci: Pengisian Galon Otomatis, ESP32, QR Code, Flowmeter, Internet Of Things.***

## **ABSTRACT**

*The water gallon filling process at drinking water depots is generally still performed manually, which can lead to inaccurate filling volumes, water wastage, and the absence of systematic usage data recording. This research aims to design and implement an automatic water gallon filling system based on the Internet of Things (IoT) using an ESP32 microcontroller as the main controller, integrating a GM-67 QR Code Scanner, a YF-S201 water flowsensor, and an HC-SR04 ultrasonic sensor. The research methodology includes hardware design, website-based software development, and comprehensive functional system testing. The system operation begins with QR Code scanning, where the QR Code carries a transaction token previously issued by the system after the user selected the desired water volume and confirmed the purchase through the website, in order to verify whether the token is still active or has already been used and to retrieve the predetermined filling volume. This is followed by gallon position verification using the ultrasonic sensor within a detection range of 1 to 3 cm. Once the gallon is confirmed at the correct position, a DC water pump is activated through a relay module, and the volume of water flowing is calculated based on accumulated pulses from the flowsensor using an optimal calibration factor of  $K = 7.4$ , until the target volume is reached and the pump is automatically deactivated. Test results show that the system successfully performs automatic gallon filling at three volume targets of 2,000 mL, 5,000 mL, and 15,000 mL, with an error percentage of  $\pm 2.50\%$  for the 2,000 mL target and 0.00% for both the 5,000 mL and 15,000 mL targets using the optimal calibration factor. The system also successfully implements a one-time use token validation mechanism, ensuring that previously used tokens are automatically rejected. All transaction data is transmitted to a web-based monitoring platform in real-time for operational supervision and data logging.*

**Keywords:** *Automatic Gallon Filling, ESP32, QR Code, Flowmeter Sensor, Internet Of Things.*

## KATA PENGANTAR

Puji Syukur kehadirat Allah SWT yang telah memberikan Rahmat dan Hidayah-Nya, sehingga penulis dapat menyelesaikan sekaligus Menyusun laporan Tugas Akhir dengan judul “RANCANG BANGUN PENGISIAN GALON OTOMATIS MENGGUNAKAN SENSOR QR CODE GM-67, FLOWMETER YF-S201, DAN WEBSITE BERBASIS ESP 32” sebagai salah satu syarat bagi penulis dalam menyelesaikan program studi Sarjana Terapan di Teknik Listrik Industri Universitas Diponegoro.

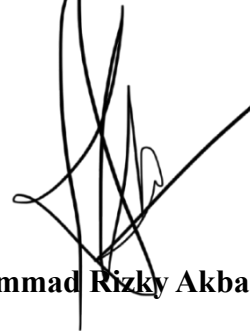
Laporan Tugas Akhir ini diharapkan dapat menambah kreativitas dan pengetahuan yang baik bagi penulis maupun pembaca laporan Tugas Akhir ini. Penulis mengucapkan terimakasih kepada semua pihak yang telah membantu dalam tersusunnya laporan ini dengan baik. Oleh karena itu penulis mengucapkan terimakasih kepada :

1. Tuhan Yang Maha Esa yang telah memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat selesai dengan baik.
2. Kedua Orang Tua yang telah memberi do'a dan dukungan kepada penulis untuk selalu semangat dalam menjalani masa pendidikan hingga tuntas.
3. Prof Dr. Ir. Budiyono, M.Si. selaku Dekan Sekolah Vokasi Universitas Diponegoro.
4. Bapak Arkhan Subari, S.T.,M.Kom. selaku Ketua Program Studi Teknik Listrik Industri Sekolah Vokasi Universitas Diponegoro.
5. Bapak Ir. H. Saiful Manan, M.T. Selaku dosen pembimbing pertama yang telah membimbing selama penulisan Tugas Akhir.
6. Bapak Yuniarto S.T., M.T. Selaku dosen pembimbing pengganti yang juga telah membantu penulis serta memberikan pengarahan dalam menyelesaikan Tugas Akhir ini.
7. Rekan-rekan Teknik Listrik Industri 2021 yang selalu berjuang bersama dari awal masa perkuliahan hingga lulus
8. Seluruh pihak yang telah banyak membantu dalam menyelesaikan Tugas Akhir ini yang tidak dapat penulis sebutkan satu persatu.

9. Serta, untuk diri saya sendiri yang tidak pernah benar-benar menyerah, meski berkali-kali hampir melakukannya, yang telah memilih untuk terus melangkah, bahkan di saat melangkah pun terasa berat, dan untuk diri saya sendiri yang telah bersabar, berjuang, dan tidak berhenti sebelum tulisan ini benar-benar selesai.

Penulis telah berupaya semaksimal mungkin dalam menyelesaikan Laporan Tugas Akhir ini, namun dengan segala kerendahan hati penulis mengakui bahwa masih banyak kekurangan yang hadir di dalamnya. Kritik dan saran yang membangun dari pembaca sangatlah berarti bagi penulis sebagai bahan penyempurnaan di masa mendatang. Semoga laporan ini dapat menjadi sesuatu yang bermanfaat dan memberi wawasan baru bagi siapa pun yang membacanya. Semoga pula Allah SWT memberkati setiap langkah dan usaha yang telah kita lakukan bersama, aamiin.

Semarang, 9 Juni 2026



**Muhammad Rizky Akbar**

## DAFTAR ISI

<b>ABSTRAK</b> .....	v
<i>ABSTRACT</i> .....	vi
<b>DAFTAR ISI</b> .....	ix
<b>DAFTAR GAMBAR</b> .....	xii
<b>DAFTAR TABEL</b> .....	xiv
<b>BAB I PENDAHULUAN</b> .....	1
<b>1.1 Latar Belakang</b> .....	1
<b>1.2 Perumusan Masalah</b> .....	3
<b>1.3 Tujuan Penelitian</b> .....	3
<b>1.4 Manfaat Penelitian</b> .....	3
<b>1.5 Batasan Masalah</b> .....	4
<b>1.6 Sistematika Penulisan Laporan</b> .....	4
<b>BAB II LANDASAN TEORI</b> .....	6
<b>2.1 Tinjauan Pustaka</b> .....	6
<b>2.2 Dasar Teori</b> .....	9
<b>2.2.1 Sistem Pengisian Galon</b> .....	9
<b>2.2.2 ESP32</b> .....	10
<b>2.2.3 QR Code Scanner GM-67</b> .....	13
<b>2.2.4 QR Code</b> .....	15
<b>2.2.5 Sensor Flowmeter YF-S201</b> .....	17
<b>2.2.6 Sensor Ultrasonik HC-SR04</b> .....	20
<b>2.2.7 Pompa Air DC</b> .....	22
<b>2.2.8 Modul Relay</b> .....	23
<b>2.2.9 Power Supply</b> .....	25
<b>2.2.10 LM2596</b> .....	26
<b>2.2.11 LCD I2C 20x4</b> .....	27
<b>2.2.12 Internet Of Things</b> .....	29

2.2.13 Logic Level Shifter .....	29
2.2.14 Arduino IDE.....	31
2.2.15 Visual Studio Code.....	33
2.2.16 Database .....	34
2.2.17 Alwaysdata.net .....	35
<b>BAB III PERANCANGAN TUGAS AKHIR .....</b>	<b>37</b>
3.1 Operasional Sistem.....	37
3.2 Identifikasi Sistem.....	38
3.3 Jenis dan Sumber Data .....	39
3.4 Prosedur Pengumpulan Data.....	39
3.5 Pengujian dan Analisa .....	40
3.6 Perancangan Alat .....	41
3.6.1 Desain 3D Alat.....	41
3.6.2 Blok Diagram .....	44
3.7 Cara Kerja Blok Diagram.....	44
3.8 Wiring Diagram .....	46
3.8.1 Rangkaian Catu Daya .....	47
3.8.2 Rangkaian Relay .....	48
3.8.3 Rangkaian Qr Code Scanner GM-67.....	50
3.8.4 Rangkaian Sensor Flowmeter .....	51
3.8.5 Rangkaian Ultrasonik.....	52
3.8.6 Rangkaian LCD20x4.....	53
3.8.7 Rangkaian Buzzer .....	55
3.9 Perancangan Software .....	56
3.9.1 Tampilan Website .....	56
3.10 Flowchart .....	58
<b>BAB IV PEMBUATAN ALAT.....</b>	<b>60</b>
4.1 Pembuatan Perangkat Keras.....	60
4.1.1 Alat dan Bahan.....	60

4.1.2 Pembuatan Papan PCB.....	61
4.1.3 Pembuatan Perangkat Mekanik.....	63
4.2 Pembuatan Perangkat Lunak.....	71
4.2.1 Pembuatan Program Pada Arduino IDE .....	71
4.2.2 Pembuatan Website .....	74
4.2.2.1 Program Pembentuk Website.....	75
4.2.3 Mekanisme Pembuatan Token dan QR Code .....	80
4.2.3.1 Pembuatan Token Unik .....	81
4.2.3.2 Penyimpanan Token Ke Database .....	82
4.2.3.3 Rendering QR Code pada Browser.....	83
4.2.3.4 Fitur Unduh QR Code .....	83
4.2.3.5 Alur Keseluruhan Proses Generate Token dan QR Code .....	85
4.2.4 Hosting dan Publikasi Website.....	86
4.2.5 Pembuatan Database .....	87
4.2.6 Tampilan Website .....	90
4.3 Alur Penggunaan Alat Alat Pengisian Galon Otomatis .....	93
<b>BAB V PENGUJIAN DAN PEMBAHASAN .....</b>	<b>95</b>
5.1 Pengukuran dan Pengujian .....	95
5.1.1 Tujuan dan Parameter Pengujian .....	95
5.1.2 Prosedur dan Lingkungan Pengujian .....	96
5.1.3 Pengujian Konsistensi Data Transaksi.....	97
5.1.4 Pengujian Validasi Token QR Code.....	101
5.1.5 Pengujian Akurasi Kalibrasi dan Volume Pengisian .....	108
5.1.6 Pengujian Konsistensi Pengisian dengan Faktor K 7.4.....	112
5.1.7 Pengujian Sensor Ultrasonik .....	114
<b>BAB VI PENUTUP .....</b>	<b>117</b>
6.1 Kesimpulan .....	117
6.2 Saran .....	118
<b>DAFTAR PUSTAKA.....</b>	<b>120</b>

**LAMPIRAN** ..... 122

## DAFTAR GAMBAR

Gambar 2. 1 Galon Air .....	9
Gambar 2. 2 ESP 32 DEVKIT V1 .....	10
Gambar 2. 3 Pinout ESP32 .....	12
Gambar 2. 4 QR CODE SCANNER GM-67 .....	13
Gambar 2. 5 QR CODE .....	15
Gambar 2. 6 Sensor Flowmeter YF-s201 .....	17
Gambar 2. 7 Sensor Ultrasonik Hc-Sr04.....	20
Gambar 2. 8 Pompa Air DC .....	22
Gambar 2. 9 Relay 4 Channel .....	23
Gambar 2. 10 Switch Mode Power Supply .....	25
Gambar 2. 11 LM2596 .....	26
Gambar 2. 12 LCD I2C 20X4 .....	27
Gambar 2. 13 Logic Level Shifter.....	29
Gambar 2. 14 Logo Arduino IDE.....	31
Gambar 2. 15 Logo Visual Studio Code .....	33
Gambar 2. 16 Logo Alwaysdata.net .....	35
Gambar 3. 1 Tampak Depan Kotak Pengisian .....	41
Gambar 3. 2 Tampak Samping Kotak Pengisian.....	41
Gambar 3. 3 Bagian Atas Pada kotak.....	42
Gambar 3. 4 Bagian Tengah Pada Kotak .....	42
Gambar 3. 5 Blok Diagram .....	44
Gambar 3. 6 Rangkaian Secara Keseluruhan .....	46
Gambar 3. 7 Wiring Diagram Secara Keseluruhan .....	46
Gambar 3. 8 Rangkaian Catu Daya.....	47
Gambar 3. 9 Rangkaian Relay .....	48
Gambar 3. 10 Rangkaian Sensor GM-67 .....	50
Gambar 3. 11 Rangkaian Sensor Flowmeter.....	51
Gambar 3. 12 Rangkaian Sensor Ultrasonik .....	52
Gambar 3. 13 Rangkaian LCD I2C 20x4.....	53
Gambar 3. 14 Rangkaian Buzzer .....	55
Gambar 3. 15 Tampilan Web Monitoring .....	56
Gambar 3. 16 Tampilan Web Pemilihan Ukuran Galon.....	57
Gambar 3. 17 Flowchart.....	58
Gambar 4. 1Desain PCB .....	61
Gambar 4. 2 Pencetakan PCB .....	62
Gambar 4. 3 Pembuatan Kotak Pengisian Galon .....	63
Gambar 4. 4 Pemberian Cat Pada Kotak Pengisian Galon .....	64

Gambar 4. 5 Pemasangan Komponen Sensor QR Code GM-67, LCD, Pilot Lamp,, dan Saklar.....	64
Gambar 4. 7 Perakitan Papan PCB .....	65
Gambar 4. 8 Pemasangan PCB dan PSU Pada Akrilik .....	66
Gambar 4. 9 Perakitan Pompa dan Sensor Flowmeter.....	66
Gambar 4. 10 Perakitan Komponen Kedalam Kotak Pengisian Galon .....	67
Gambar 4. 11 Pemasangan Sensor Ultrasonik pada Kotak Pengisian Galon.....	68
Gambar 4. 12 Kotak Pengisian Dengan Galon .....	69
Gambar 4. 13 Tampak Keseluruhan Komponen pada Kotak Pengisian Galon ....	70
Gambar 4. 14 Program pada Arduino IDE.....	71
Gambar 4. 15 Program db.php .....	76
Gambar 4. 16 Program Index.php .....	77
Gambar 4. 17 Program api.php .....	78
Gambar 4. 18 Program admin.php .....	79
Gambar 4. 19 Program Process.php .....	80
Gambar 4. 20 Kode Program Pembentukan Token Unik Menggunakan Fungsi MD5 pada PHP .....	81
Gambar 4. 21 Token QR Code yang dihasilkan.....	82
Gambar 4. 22 Kode Program Penyimpanan Data Transaksi ke Database .....	82
Gambar 4. 23 Kode Program Pembuatan QR Code Berdasarkan Token Menggunakan Library QRious.....	83
Gambar 4. 24 Kode Program Tombol Unduh QR Code .....	83
Gambar 4. 25 Kode Program Tombol Download QR Code pada Website .....	84
Gambar 4. 26 Kode Program Penamaan File QR Code Saat Proses Unduh .....	85
Gambar 4. 27 Hasil Unduhan Token dan QR Code .....	85
Gambar 4. 28 Panel Pembuatan Database pada Alwaysdata .....	88
Gambar 4. 29 Tampilan Antarmuka phpMyAdmin pada Database Sistem .....	89
Gambar 4. 30 Tampilan Data Tabel Tokens pada phpMyAdmin.....	89
Gambar 4. 31 Tampilan Website Pemilihan Galon .....	91
Gambar 4. 32 Tampilan Website Login Admin.....	91
Gambar 4. 33 Tampilan Website Monitoring .....	92
Gambar 4. 34 Flowchart Penggunaan Alat .....	93
Gambar 5. 1 Perubahan Status Semua Token Volume 2 Liter .....	100
Gambar 5. 2 Perubahan Status Semua Token Volume 5 Liter .....	101
Gambar 5. 3 Perubahan Status Semua Token Volume 15 Liter .....	101
Gambar 5. 4 Dokumentasi Pengujian Pengisian 2 Liter .....	112
Gambar 5. 5 Dokumentasi Pengujian Pengisian 5 Liter .....	113
Gambar 5. 6 Dokumentasi Pengujian Pengisian 10 Liter .....	114
Gambar 5. 7 Potongan Kode Program Mekanisme Timeout Sensor Ultrasonik..	116

## DAFTAR TABEL

Tabel 2. 1 Daftar Referensi .....	6
Tabel 2. 2 Spesifikasi Sensor QR GM-67 .....	14
Tabel 2. 3 Spesifikasi Sensor Flowmeter YF-S201.....	17
Tabel 2. 4 Spesifikasi Sensor Ultrasonik HC-SR04.....	21
Tabel 2. 5 Spesifikasi Pompa Air DC.....	23
Tabel 2. 6 Spesifikasi Modul Relay .....	24
Tabel 2. 7 Spesifikasi Power Supply 10A 12V .....	26
Tabel 2. 8 Spesifikasi Buck Converter LM2596 .....	27
Tabel 2. 9 Spesifikasi LCD I2C 20x4 .....	28
Tabel 2. 10 Spesifikasi Logic Level Shifter .....	30
Tabel 4. 1 Daftar Alat Yang Digunakan .....	60
Tabel 4. 2 Daftar Bahan Yang Digunakan.....	60
Tabel 5. 1 Pengujian Transaksi.....	97
Tabel 5. 2 Pengujian Validasi Token QR Volume 2L, 5L, dan 15L Sebelum Digunakan .....	102
Tabel 5. 3 Pengujian Validasi Token QR Volume 2L, 5L, dan 15L Setelah Digunakan .....	105
Tabel 5. 4 Pengujian Sensor Flowmeter.....	110
Tabel 5. 5 Tabel Pengujian Konsistensi Pengisian 2 Liter .....	112
Tabel 5. 6 Tabel Pengujian Konsistensi Pengisian 5 Liter .....	112
Tabel 5. 7 Tabel Pengujian Konsistensi Pengisian 15 Liter .....	113
Tabel 5. 8 Pengujian Sensor Ultrasonik .....	115

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Air minum merupakan salah satu kebutuhan utama masyarakat, dan penggunaannya terus meningkat seiring pertumbuhan penduduk. Depot air minum isi ulang menjadi solusi populer karena menyediakan air minum dengan harga terjangkau. Namun, proses pengisian air galon di banyak depot masih dilakukan secara manual, yaitu dengan membuka dan menutup keran atau saklar secara langsung. Metode ini tidak hanya memerlukan tenaga manusia secara terus-menerus tetapi juga menimbulkan berbagai masalah, seperti ketidakakuratan volume pengisian, potensi tumpahnya air, waktu pengisian yang tidak konsisten, serta kurangnya pencatatan penggunaan air yang akurat. Hal ini menunjukkan perlunya inovasi sistem pengisian air galon yang lebih otomatis, higienis, efektif, dan efisien [1].

Perkembangan teknologi IoT (Internet of Things) dan mikrokontroler modern seperti ESP32 memberikan peluang besar untuk mengatasi permasalahan tersebut. Dengan integrasi berbagai sensor, sistem otomatisasi dapat mengendalikan pengisian air secara presisi tanpa intervensi manusia. Salah satu aspek krusial dalam proses pengisian adalah penentuan volume air yang sesuai dengan kapasitas galon. Saat ini, terdapat beragam ukuran galon, seperti 2 liter, 5 liter, dan 15 liter. Pengisian manual sering kali menyebabkan ketidaksesuaian volume akibat variasi kecepatan aliran air atau human error. Oleh karena itu, diperlukan sistem yang mampu mengatur volume pengisian secara otomatis berdasarkan kapasitas galon [2].

Pemilihan variasi volume pengisian sebesar 2 liter, 5 liter, dan 15 liter dalam sistem ini didasarkan pada kebutuhan konsumsi air yang umum digunakan oleh masyarakat dalam kehidupan sehari-hari. Air merupakan kebutuhan pokok yang digunakan dalam berbagai aktivitas, baik untuk konsumsi langsung maupun keperluan rumah tangga. Oleh karena itu, sistem pengisian air yang dirancang perlu menyesuaikan dengan pola penggunaan air yang beragam, sehingga dapat

memberikan fleksibilitas bagi pengguna dalam memilih jumlah air sesuai kebutuhan.

Penggunaan QR Code sebagai sarana identifikasi transaksi menawarkan solusi yang praktis dan modern dalam sistem pengisian galon otomatis. Berbeda dengan sistem konvensional yang memerlukan input manual, QR Code pada sistem ini berfungsi sebagai token transaksi yang telah memuat informasi volume air yang sebelumnya dipilih dan dibayarkan oleh pelanggan melalui website. Artinya, sebelum datang ke alat pengisian, pelanggan terlebih dahulu mengakses website, memilih volume yang diinginkan, melakukan pembayaran, dan sistem secara otomatis menghasilkan QR Code yang berisi token transaksi tersebut.

Ketika QR Code dipindai menggunakan scanner yang terintegrasi dengan mikrokontroler ESP32, sistem akan mengirimkan token tersebut ke server untuk divalidasi. Apabila token dinyatakan valid, server akan mengembalikan informasi volume air yang telah dibayarkan, dan proses pengisian pun dimulai secara otomatis sesuai volume tersebut tanpa memerlukan input apapun dari pengguna di lokasi

Kebutuhan monitoring penggunaan air juga menjadi pertimbangan penting, terutama bagi pelaku usaha depot air minum. Dengan teknologi IoT, data seperti jumlah liter air yang digunakan per hari, jumlah pengisian berdasarkan kategori galon, dan lama waktu pengisian dapat dicatat dan dipantau secara real-time melalui aplikasi atau web. Data ini sangat bermanfaat untuk evaluasi operasional dan manajemen penggunaan air.

Berdasarkan tantangan dan peluang tersebut, penelitian ini mengusulkan perancangan dan implementasi sistem pengisian air galon otomatis berbasis QR Code, sensor flowmeter, dan IoT menggunakan ESP32. Sistem ini diharapkan dapat meningkatkan akurasi, efisiensi, dan kemudahan monitoring dalam pengisian air galon. Berdasarkan latar belakang tersebut, penulis melakukan penelitian dengan judul: “RANCANG BANGUN PENGISIAN GALON OTOMATIS MENGGUNAKAN SENSOR QRCODE GM-67, FLOWMETER YF-S201, DAN WEBSITE BERBASIS ESP 32”.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana merancang sistem monitoring berbasis website yang mampu mencatat dan menampilkan data transaksi pengisian air galon secara *real-time*
2. Bagaimana merancang mekanisme kontrol otomatis yang memastikan proses pengisian air hanya berlangsung ketika galon terdeteksi pada posisi yang benar dan berhenti tepat saat volume target tercapai?

## 1.3 Tujuan Penelitian

Adapun tujuan dari pelaksanaan Tugas Akhir ini adalah

1. Menghasilkan prototipe sistem pengisian air galon otomatis yang mampu memvalidasi token QR Code sebagai pemicu pengisian dan mengukur volume air yang diisikan secara akurat menggunakan sensor flowmeter YF-S201
2. Mengimplementasikan Sensor Ultrasonik yang mampu mendeteksi keberadaan galon air secara otomatis
3. Mengintegrasikan *Website* yang mampu mencatat pengisian air galon yang telah dilakukan secara *real-time*

## 1.4 Manfaat Penelitian

Penelitian dan pengembangan sistem ini diharapkan memberikan manfaat, antara lain:

1. Bagi Pelaku Usaha: Sistem ini dapat meningkatkan efisiensi dan akurasi dalam pengisian air galon, mengurangi ketergantungan pada tenaga manual, serta menyediakan data monitoring yang akurat untuk manajemen operasional.
2. Bagi Pelanggan: Proses pengisian menjadi lebih cepat, akurat, dan higienis karena tidak ada kontak langsung dengan operator.

3. Bagi Pengembangan Teknologi: Penelitian ini dapat menjadi referensi pengembangan sistem otomatis dan IoT dalam bidang pengelolaan air minum

### **1.5 Batasan Masalah**

Agar penelitian lebih terarah dan sesuai ruang lingkup jurusan Teknik Listrik Industri, maka batasan masalah dalam Tugas Akhir ini adalah:

1. Sistem hanya menangani tiga ukuran galon: 2L, 5L, dan 15L. Pengisian galon dilakukan secara otomatis setelah QR Code berhasil di-scan dan galon terdeteksi oleh sensor ultrasonic.
2. Pengukuran volume air dilakukan menggunakan sensor flowmeter dengan satuan liter untuk menentukan jumlah air yang keluar.
3. Sistem tidak membahas aspek keamanan air minum, kualitas air, atau proses filtrasi, melainkan hanya difokuskan pada proses pengisian dan pencatatan pada Website.

### **1.6 Sistematika Penulisan Laporan**

Sistematika penyusunan tugas akhir dibuat untuk memberikan beberapa gambaran umum tentang kegiatan yang dilakukan. Sistematika penyusunan tugas akhir ini adalah sebagai berikut:

#### **BAB I PENDAHULUAN**

Bab ini menjelaskan latar belakang perancangan tugas akhir, identifikasi dan perumusan masalah, batasan masalah, tujuan tugas akhir, manfaat tugas akhir, metode penyusunan tugas akhir, dan sistematika tugas akhir.

#### **BAB II LANDASAN TEORI**

Bab ini menjelaskan tentang tinjauan pustaka dan teori yang mendasari topik Rancang Bangun Pengisian Galon Otomatis Dengan Menggunakan Sensor Qrcode, Sensor Flowmeter, dan Website Berbasis Esp 32. Pembahasan dimulai penelitian terdahulu yang pernah dilakukan, dan penjelasan komponen-komponen yang digunakan dalam pembuatan alat

### **BAB III PERANCANGAN TUGAS AKHIR**

Bab ini menjelaskan tentang tahapan perancangan alat. Tahapan ini menjadi dua poin utama, yaitu perancangan hardware dan perancangan software. Perancangan hardware dilakukan dengan membuat perancangan diagram blok dan flowchart sistem.

### **BAB IV PEMBUATAN ALAT**

Dalam bab ini menjelaskan tentang tahapan pembuatan secara nyata dari keseluruhan sistem sesuai dengan perancangan yang telah dilaksanakan. Hardware alat dirangkai secara sempurna sesuai perancangan desain 3D yang telah dibuat. Cara penguploadan data program kedalam aplikasi Arduino IDE.

### **BAB V PENGUJIAN DAN ANALISA ALAT**

Bab ini menjelaskan tentang pengujian alat dengan tujuan untuk memastikan sistem yang dibuat dapat bekerja sesuai dengan perencanaan yang dibuat sebelumnya

### **BAB VI PENUTUP**

Dalam bab ini berisi tentang kesimpulan dan saran mengenai alat yang telah dibuat sehingga dapat digunakan untuk pengembangan selanjutnya.

## BAB II LANDASAN TEORI

### 2.1 Tinjauan Pustaka

Berikut tinjauan pustaka yang telah dilakukan oleh peneliti sebelumnya :

Tabel 2. 1 Daftar Referensi

No.	Peneliti	Judul	Topik
1.	Rashad, A. A., Girawan, B. A., Ariawan, R., Sodikin, J., & Hastuti, H. D. (2023)	Uji Fungsi dan Kalibrasi Sensor Water Flow YF- S201 Berbasis Arduino Uno pada Mesin Penjernih Air Sungai. <i>Jurnal Teknik Mesin, Elektro dan Ilmu Komputer</i> , 3(2), 1–9. DOI: 10.55606/teknik.v3i2.18 50	Penelitian ini melakukan pengujian fungsi dan kalibrasi empiris sensor flowmeter YF-S201 berbasis Arduino Uno pada mesin penjernih air. Hasil pengujian menunjukkan bahwa kalibrasi sensor secara langsung menggunakan wadah ukur menghasilkan error rata-rata sebesar 0,3%, jauh lebih akurat dibandingkan menggunakan nilai default pabrik. Penelitian ini relevan sebagai dasar metodologi kalibrasi sensor flowmeter YF-S201 yang digunakan pada sistem pengisian galon otomatis dalam penelitian ini[9].
2.	Munandar, A., Veronika, N.	Miniature Design of Liquid Filling Machine	Penelitian ini merancang miniatur mesin pengisi

No.	Peneliti	Judul	Topik
	D. M., Abdullah, D., & Sahputra, E. (2023)	Automatically Using ESP32 Based IoT (Internet of Things). <i>Jurnal Komputer, Informasi dan Teknologi (KOMITEK)</i> , 3(1), 69– 78. DOI: 10.53697/jkomitek.v3i1. 1185	cairan otomatis menggunakan mikrokontroler ESP32 berbasis IoT dengan kendali relay dan pompa air sebagai aktuator. Sistem mampu melakukan pengisian cairan secara otomatis berdasarkan volume yang ditentukan dan dapat dipantau melalui aplikasi Blynk. Penelitian ini relevan karena menggunakan komponen dan pendekatan yang identik dengan sistem pengisian galon otomatis dalam penelitian ini, yaitu ESP32, relay, dan pompa air [4].
3.	Nur, W. W., Ardian, A. D. S., & Muhammad, S. A. P. (2024)	Implementasi Sistem Keamanan IoT Berbasis QR Code pada Loker untuk Peningkatan Keamanan dan Aksesibilitas. <i>Jurnal Elektronika dan Otomasi Industri</i> , 11(3), 788–796	Penelitian ini mengimplementasikan QR Code sebagai sistem keamanan berbasis IoT menggunakan ESP32-CAM dan ESP32 untuk kendali akses loker secara otomatis. Sistem melakukan validasi QR Code secara real-time dan menampilkan status kondisi melalui aplikasi

No.	Peneliti	Judul	Topik
			<p>monitoring. Penelitian ini relevan sebagai referensi implementasi QR Code sebagai media identifikasi dan validasi berbasis IoT yang menjadi prinsip kerja utama sistem pengisian galon otomatis dalam penelitian ini [8].</p>
4	Muamaroh, N. & Christanto, F. W. (2024).	<p>Pengukur Penggunaan Air Otomatis Menggunakan Water Sensor Flowmeter YF-S201 dan NodeMCU ESP8266 Berbasis IoT. <i>JIKO (Jurnal Informatika dan Komputer)</i>, Universitas Teknologi Digital Indonesia</p>	<p>Penelitian ini merancang alat pengukur penggunaan air secara otomatis menggunakan sensor flowmeter YF-S201 berbasis IoT. Hasil kalibrasi sensor dibandingkan dengan gelas ukur menunjukkan error rata-rata sebesar 1,9%. Data penggunaan air dikirim ke server dan dapat dipantau secara real-time. Penelitian ini relevan karena menggunakan sensor dan metode pengukuran volume air yang sama dengan sistem yang dikembangkan dalam penelitian ini [6].</p>

## 2.2 Dasar Teori

### 2.2.1 Sistem Pengisian Galon



Gambar 2. 1 Galon Air

Sistem pengisian galon merupakan rangkaian proses yang bertujuan untuk memindahkan air dari sumber atau penampungan menuju wadah galon dengan volume tertentu. Pada sistem konvensional, proses pengisian dilakukan secara manual dengan membuka kran atau mengaktifkan pompa, kemudian menghentikan aliran air berdasarkan pengamatan visual. Metode ini memiliki kelemahan dalam hal akurasi volume, efisiensi waktu, serta pencatatan data penggunaan.

Pengisian galon otomatis merupakan pengembangan dari sistem konvensional yang memanfaatkan teknologi sensor dan mikrokontroler untuk mengendalikan proses secara mandiri. Sistem ini bekerja berdasarkan prinsip kontrol tertutup (closed-loop control), di mana sensor membaca kondisi aktual sistem, kemudian mikrokontroler memproses data tersebut untuk menghasilkan keputusan pengendalian aktuator. Dalam konteks penelitian ini, sensor flowmeter digunakan untuk mengukur debit dan volume air, sementara pompa air DC berfungsi sebagai aktuator pengalir air.

Proses pengisian dimulai dengan tahap identifikasi melalui QR Code yang dipindai menggunakan modul scanner. Data hasil pemindaian dikirim ke mikrokontroler untuk menentukan parameter volume yang harus diisi sesuai kapasitas galon, seperti 2 liter, 5 liter, atau 15 liter. Setelah galon terdeteksi berada pada posisi pengisian (melalui sensor ultrasonik), pompa air diaktifkan dan sensor

Flowmeter mulai menghitung jumlah air yang mengalir. Ketika volume yang dihitung telah mencapai nilai yang ditentukan, mikrokontroler secara otomatis mematikan pompa.

Integrasi sistem ini dengan website memungkinkan seluruh proses pengisian terdokumentasi secara digital. Data seperti jenis galon, volume pengisian, waktu transaksi, dan status token disimpan ke dalam database melalui koneksi internet. Dengan pendekatan ini, sistem pengisian galon tidak hanya berfungsi sebagai alat otomatisasi, tetapi juga sebagai sistem monitoring dan pencatatan berbasis IoT yang meningkatkan efisiensi, akurasi, dan keamanan operasional.

### 2.2.2 ESP32



Gambar 2. 2 ESP 32 DEVKIT V1

ESP32 merupakan mikrokontroler berbasis System on Chip (SoC) yang dikembangkan oleh Espressif Systems sebagai penerus dari ESP8266. Perangkat ini dirancang untuk mendukung pengembangan sistem berbasis Internet of Things (IoT) dengan menyediakan konektivitas WiFi dan Bluetooth yang terintegrasi dalam satu chip. Dengan teknologi fabrikasi 40 nm, ESP32 memiliki efisiensi daya yang baik serta performa radio yang stabil dan andal dalam berbagai aplikasi, baik untuk sistem berdaya rendah maupun sistem dengan kebutuhan pemrosesan yang lebih kompleks [7][8].

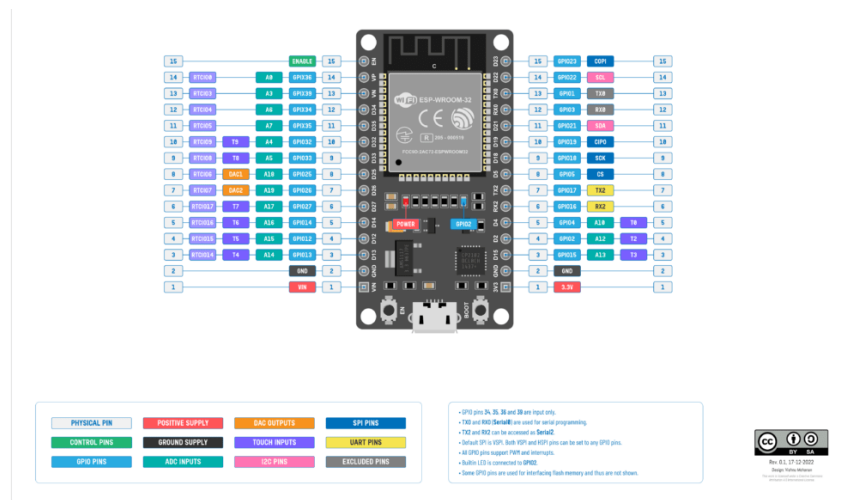
Secara arsitektur, ESP32 menggunakan prosesor Tensilica Xtensa LX6 32-bit yang tersedia dalam konfigurasi single-core maupun dual-core, dengan frekuensi kerja hingga 240 MHz. Performa tersebut memungkinkan ESP32 menjalankan berbagai proses secara simultan, termasuk pembacaan sensor, pengendalian aktuator, komunikasi jaringan, serta pengolahan data secara real-time. ESP32 juga

dilengkapi dengan memori internal sebesar 520 KiB SRAM serta memori flash eksternal yang umumnya berkapasitas 4 MB untuk penyimpanan program dan data [7][8].

Dalam hal antarmuka dan periferal, ESP32 menyediakan berbagai fitur yang mendukung fleksibilitas pengembangan sistem, seperti sejumlah pin digital input/output (GPIO), kanal analog-to-digital converter (ADC), digital-to-analog converter (DAC), serta dukungan komunikasi serial seperti UART, SPI, dan I2C. Fitur konektivitas nirkabelnya mendukung standar IEEE 802.11 b/g/n pada frekuensi 2.4 GHz serta Bluetooth versi 4.2 (BLE dan Classic), sehingga memungkinkan pertukaran data melalui jaringan WiFi maupun komunikasi langsung antar perangkat berbasis Bluetooth [7][8].

Salah satu papan pengembangan yang banyak digunakan adalah ESP32 DevKit V1, yaitu development board yang dirancang untuk mempermudah integrasi ESP32 ke dalam berbagai proyek elektronika. Board ini dilengkapi dengan regulator tegangan, antarmuka USB untuk pemrograman, serta tata letak pin yang memudahkan proses perancangan rangkaian. Dengan ukuran yang relatif ringkas, board ini dapat diintegrasikan ke dalam sistem yang lebih kompleks tanpa memerlukan ruang yang besar [7][8].

Dalam penelitian ini, ESP32 berperan sebagai pengendali utama sistem pengisian galon otomatis. Mikrokontroler ini bertugas membaca data dari QR Code Scanner, sensor flowmeter, dan sensor ultrasonik, kemudian memproses data tersebut untuk mengendalikan aktuator seperti pompa air dan modul relay. Selain itu, ESP32 juga mengirimkan data transaksi pengisian ke server melalui koneksi WiFi untuk keperluan monitoring berbasis website. Dengan kombinasi kemampuan pemrosesan yang tinggi, dukungan konektivitas nirkabel, serta jumlah periferal yang memadai, ESP32 menjadi platform yang tepat untuk implementasi sistem pengisian galon otomatis berbasis IoT.



Gambar 2. 3 Pinout ESP32

Berikut adalah penjelasan tentang konfigurasi pinout dari ESP32:

1. Pin GPIO: Papan ini menyediakan sejumlah pin Input/Output Serbaguna (GPIO) yang dapat digunakan untuk berbagai fungsi input/output digital. Pinpin ini juga mendukung fungsi seperti PWM, I2C, SPI, dan lainnya.
2. Input Analog: Beberapa pin pada ESP32 DevKit V1 mampu membaca sinyal analog, membuatnya cocok untuk dihubungkan dengan sensor analog.
3. Pin 3.3V dan GND: Digunakan untuk memberi daya pada komponen atau sensor eksternal.
4. V dan GND: Papan juga dapat menyediakan keluaran 5V, yang berguna untuk memberi daya pada modul eksternal yang memerlukan daya lebih besar.
5. VIN: Ini adalah pin tegangan input, yang dapat digunakan untuk memberi daya pada papan saat tidak menggunakan koneksi USB.
6. ID: Ini adalah pin pengaktif. Pin ini digunakan untuk mengatur ulang mikrokontroler.
7. TX/RX: Pin ini digunakan untuk komunikasi serial.
8. Antarmuka SPI: Papan memiliki pin untuk komunikasi SPI, memungkinkan transfer data cepat dengan periferal seperti layar atau memori flash.
9. Antarmuka I2C: ESP32 DevKit V1 mendukung komunikasi I2C, yang banyak digunakan untuk berinteraksi dengan sensor dan periferal lainnya.

### 2.2.3 QR Code Scanner GM-67



Gambar 2. 4 QR CODE SCANNER GM-67

QR Code Scanner GM67 merupakan modul pemindai kode batang satu dimensi (1D) dan dua dimensi (2D) yang bekerja menggunakan teknologi pencitraan (*image-based scanning*). Berbeda dengan pemindai laser konvensional, modul GM67 menggunakan sensor optik internal untuk menangkap citra kode, kemudian memprosesnya menggunakan algoritma decoding untuk mengekstraksi data yang tersimpan dalam barcode atau QR Code. Teknologi ini memungkinkan pembacaan kode dilakukan secara cepat, stabil, dan akurat pada berbagai kondisi pencahayaan.

Modul GM67 umumnya mendukung berbagai jenis kode 1D seperti EAN-13, Code 128, Code 39, serta kode 2D seperti QR Code, Data Matrix, dan PDF417. Proses kerja modul ini dimulai ketika sensor optik menangkap pola matriks kode, kemudian citra tersebut dikonversi menjadi sinyal digital. Selanjutnya, prosesor internal pada modul melakukan proses decoding untuk menerjemahkan pola tersebut menjadi data teks atau numerik yang kemudian dikirimkan melalui antarmuka komunikasi serial seperti UART atau TTL. Dengan dukungan komunikasi serial, modul ini dapat dengan mudah diintegrasikan dengan berbagai mikrokontroler.

GM67 bekerja pada tegangan operasi sekitar 5V DC dan memiliki jarak baca yang bergantung pada ukuran dan kualitas cetakan kode. Modul ini juga dirancang untuk memiliki waktu respons yang cepat sehingga cocok digunakan dalam sistem otomatisasi yang membutuhkan proses identifikasi real-time. Selain itu, beberapa

varian GM67 dilengkapi dengan fitur auto-sensing, yaitu kemampuan mendeteksi keberadaan kode secara otomatis tanpa perlu tombol pemicu (*trigger*).

Keunggulan QR Code Scanner GM67 terletak pada kemudahan integrasi, ukuran yang relatif kompak, serta kompatibilitas dengan berbagai platform sistem tertanam. Modul ini banyak digunakan dalam sistem identifikasi, sistem absensi, sistem pembayaran, sistem kontrol akses, serta berbagai aplikasi berbasis Internet of Things (IoT) yang memerlukan autentikasi berbasis QR Code. Dengan karakteristik tersebut, GM67 menjadi salah satu modul pemindai QR Code yang efisien dan andal untuk mendukung sistem identifikasi otomatis berbasis mikrokontroler [9][10].

Tabel 2. 2 Spesifikasi Sensor QR GM-67

No	Parameter	Spesifikasi
1	Tipe Sensor	CMOS Imaging
2	Resolusi Sensor	640 × 480 piksel
3	Jenis Barcode	1D dan 2D
4	Dukungan 2D	QR Code, Data Matrix, PDF417, Micro QR, Aztec, Micro PDF417
5	Dukungan 1D	EAN-13, EAN-8, UPC-A, UPC-E, Code 39, Code 128, Codabar, ITF-14, Interleaved 2 of 5, GS1 Databar
6	Tegangan Kerja	5V DC
7	Arus Operasi	±180 mA
8	Arus Standby (USB)	±25 mA
9	Arus Standby (TTL)	±10 mA
10	Antarmuka Komunikasi	UART (TTL) dan USB
11	Baud Rate Default	9600 bps

No	Parameter	Spesifikasi
12	Format Output Data	ASCII
13	Mode Trigger	Low level trigger (melalui pin TRIG)
14	Sumber Cahaya	White LED
15	Collimate Light	Red LED
16	Jarak Baca QR Code	30 mm – 220 mm
17	Jarak Baca EAN-13	60 mm – 220 mm
18	Jarak Baca Data Matrix	40 mm – 110 mm
19	Jarak Baca PDF417	30 mm – 130 mm
20	Akurasi Pembacaan	$\geq 5$ mil
21	Sudut Rotasi	360°
22	Elevation Angle	$\pm 55^\circ$
23	Deflection Angle	$\pm 55^\circ$

#### 2.2.4 QR Code



Gambar 2. 5 QR CODE

QR Code dirancang untuk memungkinkan proses pembacaan data yang cepat dan akurat dibandingkan dengan barcode satu dimensi konvensional. Berbeda dengan barcode linear yang hanya menyimpan data secara horizontal, QR Code menyimpan informasi dalam dua arah, yaitu horizontal dan vertikal, sehingga kapasitas penyimpanan datanya jauh lebih besar.

Secara struktur, QR Code terdiri atas modul-modul persegi kecil berwarna hitam dan putih yang tersusun dalam pola matriks. Di dalam QR Code terdapat beberapa bagian penting, seperti finder pattern yang berfungsi sebagai penanda orientasi, alignment pattern untuk menjaga akurasi pembacaan, serta area penyimpanan data yang memuat informasi yang telah dikodekan. Selain itu, QR Code dilengkapi dengan fitur error correction berbasis algoritma Reed-Solomon yang memungkinkan kode tetap dapat dibaca meskipun sebagian permukaan mengalami kerusakan atau gangguan.

Proses kerja QR Code melibatkan dua tahap utama, yaitu proses encoding dan decoding. Pada tahap encoding, data asli dikonversi menjadi pola biner yang kemudian disusun dalam bentuk matriks QR Code. Pada tahap decoding, perangkat pemindai menggunakan sensor optik atau kamera untuk menangkap citra QR Code, kemudian sistem melakukan analisis pola untuk mengekstraksi data yang tersimpan. Proses ini biasanya dilakukan secara otomatis oleh modul pemindai atau perangkat lunak pengolah citra.

Penggunaan QR Code telah berkembang luas dalam berbagai bidang, seperti sistem pembayaran digital, autentikasi pengguna, pelacakan produk, sistem tiket elektronik, serta sistem identifikasi berbasis teknologi informasi. Keunggulan QR Code terletak pada kemampuannya menyimpan data dalam kapasitas besar, kemudahan integrasi dengan sistem digital, serta proses pembacaan yang cepat dan akurat. Oleh karena itu, QR Code menjadi salah satu teknologi identifikasi yang efisien dan banyak diterapkan dalam sistem otomatisasi dan Internet of Things [9].

### 2.2.5 Sensor Flowmeter YF-S201



Gambar 2. 6 Sensor Flowmeter YF-s201

Sensor Flowmeter YF-S201 merupakan sensor aliran air berbasis efek Hall (Hall Effect) yang digunakan untuk mengukur debit dan volume fluida yang mengalir dalam suatu sistem perpipaan. Sensor ini bekerja dengan prinsip mekanis-elektronik, di mana aliran air yang melewati badan sensor akan memutar rotor internal yang dilengkapi dengan magnet permanen. Putaran rotor tersebut kemudian dideteksi oleh sensor Hall Effect yang menghasilkan pulsa listrik sesuai dengan kecepatan aliran fluida.

Struktur utama YF-S201 terdiri dari badan katup plastik, turbin atau rotor dengan magnet, serta sensor Hall Effect yang terpasang pada bagian atas modul. Ketika air mengalir melalui sensor, rotor akan berputar sebanding dengan kecepatan aliran air. Setiap satu putaran rotor menghasilkan sejumlah pulsa digital yang dapat dibaca oleh mikrokontroler melalui pin input digital. Jumlah pulsa yang dihasilkan dalam satu satuan waktu dapat dikonversi menjadi nilai debit (liter per menit), sedangkan akumulasi total pulsa dapat dihitung untuk mengetahui volume air yang telah mengalir (liter atau mililiter).

Tabel 2. 3 Spesifikasi Sensor Flowmeter YF-S201

Parameter	Spesifikasi YF-S201
Tegangan Kerja	5V – 24V DC
Arus Operasi	15 mA (pada 5V DC)
Diameter Dalam (Inner Diameter)	Ø 11 mm

Parameter	Spesifikasi YF-S201
Diameter Luar (Outer Diameter)	Ø 20 mm
Tekanan Air Maksimum	≤ 1,75 MPa
Rentang Debit Air	1 – 30 liter/menit
Suhu Operasi	≤ 80°C
Jenis Keluaran	Pulsa digital berbasis Hall Effect (gelombang kotak)
Level Logika Tinggi (High Level)	> 4,7 VDC (pada suplai 5V)
Level Logika Rendah (Low Level)	< 0,5 VDC (pada suplai 5V)
Faktor Kalibrasi	$F \text{ (Hz)} = 7,5 \times Q \text{ (L/menit)}$
Jumlah Pulsa per Liter	± 450 pulsa/liter
Tipe Output	NPN Open Collector
Dimensi Fisik	62 × 36 × 35 mm
Ukuran Pipa	1/2 inci

Hubungan antara frekuensi pulsa yang dihasilkan sensor dengan debit aliran air dinyatakan dalam persamaan dasar sebagai berikut:

$$F = K \times Q$$

dengan keterangan:

F = Frekuensi pulsa (Hz) atau jumlah pulsa per detik

K = Faktor kalibrasi sensor (pulsa per liter per menit)

Q = Debit aliran air (liter per menit)

Berdasarkan persamaan tersebut, untuk memperoleh nilai volume total air yang mengalir, dilakukan konversi dari frekuensi pulsa menjadi volume menggunakan persamaan turunan berikut:

$$\text{Pulsa Berhenti} = \frac{V \text{ Target}}{1000} \times (K \times 60)$$

dengan keterangan:

V = Volume air yang mengalir (liter)

K = Faktor kalibrasi sensor

60 = Faktor konversi satuan waktu dari menit ke detik

Faktor 60 pada penyebut persamaan tersebut merupakan faktor konversi satuan waktu yang wajib disertakan dalam perhitungan. Spesifikasi teknis sensor YF-S201 menyatakan hubungan antara frekuensi pulsa dan debit air dalam satuan liter per menit, bukan per detik maupun per jam. Sementara itu, mikrokontroler ESP32 menghitung pulsa menggunakan fungsi interrupt yang bekerja dalam domain waktu detik. Oleh karena itu, diperlukan konversi satuan dengan mengalikan nilai K dengan 60, yang merupakan jumlah detik dalam satu menit, agar satuan antara frekuensi pulsa yang dibaca mikrokontroler dan debit yang dinyatakan dalam spesifikasi sensor menjadi selaras.

Penggunaan satuan menit tanpa konversi akan menyebabkan hasil perhitungan volume menjadi 60 kali lebih kecil dari nilai sebenarnya, karena mikrokontroler menghitung pulsa per detik sedangkan faktor K merujuk pada pulsa per menit. Sebaliknya, penggunaan satuan jam juga tidak tepat karena akan memerlukan faktor konversi 3.600, yang tidak sesuai dengan cara kerja interrupt pada mikrokontroler yang beroperasi dalam resolusi waktu milidetik hingga detik. Dengan demikian, penggunaan faktor 60 merupakan satu-satunya konversi yang tepat untuk menyelaraskan satuan frekuensi pulsa mikrokontroler dengan satuan debit yang tertera pada spesifikasi sensor YF-S201.

Nilai K pada sensor YF-S201 secara default ditetapkan oleh pabrikan sebesar 7,5, yang berarti setiap 7,5 Hz frekuensi pulsa setara dengan debit 1 liter per menit dalam kondisi pengujian standar pabrikan. Namun demikian, nilai K dapat berbeda tergantung pada kondisi instalasi aktual seperti tekanan air, diameter pipa, posisi pemasangan sensor, serta karakteristik pompa yang digunakan. Oleh karena itu, kalibrasi empiris terhadap nilai K perlu dilakukan pada setiap instalasi baru untuk memperoleh akurasi pengukuran yang optimal.

Sensor YF-S201 umumnya bekerja pada tegangan 5V DC dengan keluaran sinyal digital berbentuk gelombang pulsa (square wave). Rentang pengukuran debit biasanya berkisar antara 1 hingga 30 liter per menit dengan tekanan kerja maksimum sekitar 1.75 MPa. Sensor ini memiliki keunggulan berupa struktur sederhana, harga ekonomis, serta kemudahan integrasi dengan mikrokontroler seperti Arduino dan ESP32 melalui interrupt digital.

Karena prinsip kerjanya berbasis putaran mekanis, akurasi sensor dapat dipengaruhi oleh faktor seperti tekanan air, suhu, posisi pemasangan, serta kualitas aliran (laminar atau turbulen). Oleh karena itu, pemasangan sensor harus memperhatikan arah aliran yang ditunjukkan oleh panah pada badan sensor serta menghindari adanya gelembung udara dalam sistem perpipaan. Dengan karakteristik tersebut, Flowmeter YF-S201 banyak digunakan dalam sistem pengukuran konsumsi air, sistem irigasi otomatis, dispenser air otomatis, serta berbagai aplikasi Internet of Things (IoT) yang memerlukan pemantauan debit dan volume fluida secara real-time[10].

#### 2.2.6 Sensor Ultrasonik HC-SR04



Gambar 2. 7 Sensor Ultrasonik Hc-Sr04

Sensor ultrasonik HC-SR04 merupakan perangkat elektronika yang berfungsi untuk mengukur jarak suatu objek dengan memanfaatkan gelombang ultrasonik. Komponen utama sensor ini terdiri dari sepasang transduser, yaitu transmitter yang bertugas memancarkan gelombang suara frekuensi tinggi, dan receiver yang berfungsi menangkap kembali pantulan gelombang tersebut setelah mengenai permukaan objek. Sensor ini bekerja pada tegangan operasional 5V DC dan memiliki empat pin terminal utama, yaitu VCC sebagai catu daya, GND sebagai

ground, Trigger sebagai pemicu transmisi, serta Echo sebagai penerima sinyal pantul [11].

Mekanisme kerja HC-SR04 didasarkan pada prinsip Time of Flight (ToF), yaitu penghitungan durasi waktu tempuh gelombang dari saat dipancarkan hingga diterima kembali. Proses dimulai ketika mikrokontroler mengirimkan pulsa logika HIGH selama 10  $\mu$  s pada pin Trigger. Sebagai respons, sensor akan memancarkan rentetan 8 pulsa ultrasonik pada frekuensi 40 kHz. Ketika gelombang tersebut mengenai objek dan memantul kembali ke arah sensor, pin Echo akan menghasilkan sinyal HIGH yang durasinya proporsional dengan waktu tempuh gelombang bolak-balik.

Tabel 2. 4 Spesifikasi Sensor Ultrasonik HC-SR04

Parameter	Spesifikasi HC-SR04
Tegangan Kerja	5 V DC
Arus Operasi	$\pm$ 15 mA
Frekuensi Gelombang	40 kHz
Jarak Minimum	2 cm
Jarak Maksimum	400 cm (4 meter)
Akurasi Pengukuran	$\pm$ 3 mm
Sudut Deteksi	$\pm$ 15°
Tipe Keluaran	Pulsa digital (durasi waktu)
Level Logika Trigger	TTL 5 V
Level Logika Echo	TTL 5 V
Lebar Pulsa Trigger	Minimal 10 $\mu$ s
Waktu Respons Maksimum	$\pm$ 60 ms
Jumlah Pin	4 pin (VCC, Trigger, Echo, GND)
Dimensi Modul	$\pm$ 45 mm $\times$ 20 mm $\times$ 15 mm

### 2.2.7 Pompa Air DC



Gambar 2. 8 Pompa Air DC

Pompa adalah alat mekanis yang digunakan untuk memindahkan cairan dari satu tempat ke tempat lain dengan cara meningkatkan tekanan cairan tersebut. Proses ini melibatkan konversi energi mekanik dari sumber penggerak (seperti motor) menjadi energi aliran fluida. Pompa bekerja dengan menciptakan perbedaan tekanan antara sisi masuk (suction) dan sisi keluar (discharge), sehingga memungkinkan cairan untuk mengalir.

Pompa DC merupakan aktuator yang berfungsi untuk mengalirkan air dari penampungan ke galon dengan mengubah energi listrik menjadi energi mekanik. Prinsip kerja pompa DC didasarkan pada motor listrik yang menghasilkan gerak rotasi untuk memindahkan fluida. Pengendalian pompa DC dilakukan melalui mikrokontroler dengan bantuan relay atau driver motor. Pada sistem pengisian galon otomatis, pompa DC diaktifkan dan dinonaktifkan secara otomatis berdasarkan hasil pengolahan data dari sensor aliran dan sensor ultrasonik.

Pompa beroperasi dengan memanfaatkan gaya sentrifugal atau prinsip positif displacement. Dalam pompa sentrifugal, energi mekanik dari motor diteruskan melalui poros ke impeller, yang berputar dan menciptakan gaya sentrifugal. Gaya ini menghisap cairan ke dalam pompa dan mendorongnya keluar dengan kecepatan yang lebih tinggi. Sebaliknya, pompa positif displacement memindahkan cairan dengan cara memaksa volume tertentu dari cairan melalui ruang pompa pada setiap siklus.

Pengendalian pompa DC dilakukan melalui mikrokontroler dengan bantuan modul relay atau driver motor. Dalam sistem pengisian galon otomatis, pompa DC diaktifkan dan dinonaktifkan secara otomatis berdasarkan hasil pengolahan data dari sensor aliran dan sensor ultrasonik. Pompa yang digunakan pada sistem ini memiliki spesifikasi 12V 22W, sehingga membutuhkan suplai tegangan 12 volt DC dengan daya maksimum 22 watt untuk menghasilkan debit air yang memadai.

Tabel 2. 5 Spesifikasi Pompa Air DC

Parameter	Spesifikasi
Tegangan Kerja	12 V DC
Daya Maksimum	22 Watt
Arus Maksimum	$\pm 1,8 - 2$ A
Jenis Pompa	Sentrifugal
Debit Maksimum	$\pm 800$ liter/jam
Head Maksimum	$\pm 5$ meter
Diameter Inlet/Outlet	1/2 inci
Material	Plastik ABS / Nylon
Suhu Operasi	$\leq 60^{\circ}\text{C}$

### 2.2.8 Modul Relay



Gambar 2. 9 Relay 4 Channel

Relay merupakan komponen elektromekanis yang berfungsi sebagai saklar listrik yang dikendalikan oleh sinyal arus rendah. Prinsip kerja relay didasarkan pada induksi elektromagnetik. Ketika arus listrik mengalir pada kumparan (coil), akan terbentuk medan magnet yang menarik tuas logam sehingga kontak saklar

berpindah posisi. Dengan mekanisme tersebut, relay memungkinkan mikrokontroler mengendalikan beban bertegangan dan berarus tinggi tanpa terhubung langsung secara listrik.

Modul relay 4 channel terdiri dari empat relay dalam satu papan rangkaian. Setiap channel dapat mengendalikan satu beban secara independen. Modul ini biasanya dilengkapi dengan transistor driver dan dioda flyback untuk melindungi rangkaian dari lonjakan tegangan akibat beban induktif. Beberapa modul juga memiliki optocoupler untuk meningkatkan isolasi antara rangkaian kontrol dan rangkaian beban. Modul ini bekerja dengan prinsip induksi magnetik yang menarik kontak saklar ketika koil mendapatkan tegangan dari ESP32. Hubungannya dengan sistem ini adalah sebagai unit antarmuka daya yang mengisolasi rangkaian kontroler 5V menuju rangkaian beban pompa 12V, sehingga meningkatkan faktor keamanan dan mencegah kerusakan pada unit mikrokontroler akibat arus balik dari motor pompa.

Penggunaan relay 4 channel sangat efisien pada sistem yang memiliki beberapa aktuator, seperti pompa, buzzer, dan indikator. Dengan konfigurasi Normally Open (NO), Normally Closed (NC), dan Common (COM), sistem dapat dikonfigurasi sesuai kebutuhan logika kendali [12].

Tabel 2. 6 Spesifikasi Modul Relay

Parameter	Spesifikasi
Tegangan Coil	5 V DC
Jumlah Channel	4 Channel
Arus Coil	$\pm 70$ mA per channel
Tegangan Maks AC	250 VAC / 10A
Tegangan Maks DC	30 VDC / 10A
Tipe Kontak	SPDT
Level Trigger	HIGH / LOW (tergantung modul)
Proteksi	Dioda flyback & transistor driver

### 2.2.9 Power Supply

Power Supply Unit (PSU) 12V 10A merupakan perangkat elektronik yang berfungsi mengubah sumber tegangan listrik AC (arus bolak-balik) dari jaringan listrik PLN menjadi tegangan DC (arus searah) sebesar 12 volt dengan arus maksimum 10 ampere. Prinsip kerjanya didasarkan pada proses penyearahan, penyaringan, dan pengaturan tegangan agar keluaran yang dihasilkan stabil dan sesuai dengan kebutuhan beban.

Pada tahap pertama, tegangan AC 220V yang masuk akan disearahkan oleh dioda penyearah (rectifier) sehingga berubah menjadi tegangan DC yang masih berdenyut (ripple). Kemudian, kapasitor filter digunakan untuk meratakan gelombang DC tersebut agar menjadi lebih halus. Setelah itu, rangkaian regulator tegangan seperti IC regulator atau sistem switching akan menjaga agar tegangan keluaran tetap konstan di 12V meskipun terjadi perubahan pada tegangan input atau beban keluaran.



Gambar 2. 10 Switch Mode Power Supply

PSU dengan kapasitas 10A berarti mampu menyuplai arus maksimum hingga 10 ampere pada tegangan 12V, yang setara dengan daya listrik sebesar 120 watt. Nilai ini menunjukkan kemampuan PSU untuk menghidupkan berbagai perangkat elektronik seperti mikrokontroler, motor DC, pompa, sensor, atau sistem kontrol otomatis yang membutuhkan suplai daya besar dan stabil. Dengan demikian, PSU 12V 10A menjadi komponen penting dalam sistem otomasi maupun rangkaian elektronik yang memerlukan suplai daya DC dengan performa tinggi dan keandalan yang baik.

Tabel 2. 7 Spesifikasi Power Supply 10A 12V

Parameter	Spesifikasi
Tegangan Input	220 VAC
Tegangan Output	12 V DC
Arus Maksimum	10 A
Daya Maksimum	120 Watt
Tipe	Switching Mode Power Supply
Frekuensi	50 – 60 Hz
Proteksi	Overload, Short Circuit, Over Voltage

### 2.2.10 LM2596



Gambar 2. 11 LM2596

LM2596 adalah modul regulator tegangan DC to DC tipe buck converter yang berfungsi untuk menurunkan (step-down) tegangan input yang lebih tinggi menjadi tegangan output yang lebih rendah dan stabil. Komponen utama dari modul ini adalah IC LM2596, yang mampu bekerja dengan efisiensi tinggi mencapai sekitar 80–92% karena menggunakan sistem switching regulator dibandingkan dengan linear regulator yang menghasilkan banyak panas. Modul ini memiliki potensiometer sebagai pengatur tegangan keluaran, serta dilengkapi dengan komponen pendukung seperti dioda Schottky, kapasitor, dan induktor untuk menjaga kestabilan dan mengurangi ripple pada tegangan output.

Prinsip kerjanya adalah dengan mengatur waktu buka-tutup transistor internal secara cepat pada frekuensi switching 150 kHz, sehingga tegangan output dapat

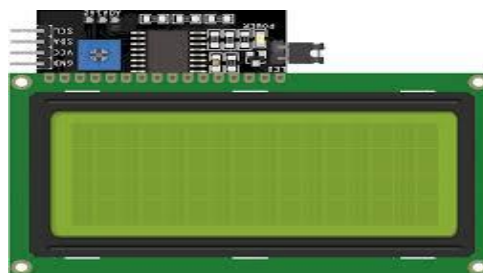
dijaga pada level tertentu sesuai dengan nilai yang diinginkan tanpa menghasilkan panas berlebih seperti pada regulator linier konvensional. Modul LM2596 yang digunakan pada sistem ini dilengkapi dengan display indikator tegangan tiga digit berbasis seven segment yang bekerja secara otomatis dan mandiri sebagai voltmeter bawaan modul. Display ini menampilkan nilai tegangan output secara real-time namun tidak terhubung dan tidak dikendalikan oleh mikrokontroler ESP32, melainkan hanya berfungsi sebagai indikator visual untuk memudahkan proses kalibrasi dan pemantauan tegangan output selama pengoperasian sistem.

Tabel 2. 8 Spesifikasi Buck Converter LM2596

Parameter	Spesifikasi
Tegangan Input	4 – 40 V DC
Tegangan Output	1,25 – 35 V DC
Arus Maksimum	3 A
Efisiensi	Hingga 92%
Frekuensi Switching	150 kHz
Tipe	Step-down (Buck Converter)

### 2.2.11 LCD I2C 20x4

LCD 20x4 merupakan modul penampil berbasis kristal cair yang mampu menampilkan 80 karakter teks dalam 4 baris. Dengan tambahan modul komunikasi I2C, perangkat ini hanya memerlukan dua jalur data (SDA dan SCL) untuk berkomunikasi dengan mikrokontroler. Pada alat ini, LCD berfungsi sebagai antarmuka visual utama bagi pengguna untuk melihat informasi real-time mengenai volume air yang sedang diisi, status sensor, serta instruksi operasional saat proses identifikasi galon berlangsung.



Gambar 2. 12 LCD I2C 20X4

LCD 20x4 berbasis antarmuka I2C merupakan modul tampilan yang banyak digunakan dalam sistem tertanam dan aplikasi Internet of Things karena kemampuannya menampilkan informasi dalam jumlah lebih besar dengan tetap mempertahankan kemudahan integrasi. Modul ini mampu menampilkan hingga 80 karakter, yaitu 20 karakter pada setiap baris dalam empat baris tampilan, sehingga sangat sesuai untuk aplikasi yang memerlukan penyajian data sistem secara lebih detail, seperti status proses, nilai pengukuran sensor, dan pesan notifikasi secara bersamaan. Dibandingkan dengan mode paralel konvensional, penggunaan antarmuka I2C pada LCD 20x4 hanya membutuhkan dua jalur komunikasi utama, yaitu SDA (Serial Data) dan SCL (Serial Clock), sehingga penggunaan pin mikrokontroler menjadi lebih efisien dan fleksibel, khususnya pada sistem dengan keterbatasan jumlah GPIO.

Tabel 2. 9 Spesifikasi LCD I2C 20x4

Parameter	Spesifikasi
Jenis Tampilan	LCD Karakter
Ukuran Tampilan	20 kolom × 4 baris (80 karakter)
Kontroler LCD	HD44780 atau kompatibel (SPLC780D)
Antarmuka Dasar	MPU 8-bit (mode paralel)
Antarmuka Komunikasi	I2C (SDA, SCL)
IC I2C Backpack	PCF8574 / PCF8574A
Tegangan Operasi	5 V DC
Tegangan Kontras (VOP)	± 4.2 V
Konsumsi Arus	± 20–30 mA (tanpa backlight)
Backlight	LED (dapat dikontrol ON/OFF)
Jenis Lampu Latar	LED putih (white side backlight)
Alamat I2C	Umumnya 0x27 atau 0x3F
Pengaturan Kontras	Potensiometer onboard
Duty Cycle LCD	1/16 Duty
Bias	1/5 Bias
Jumlah Pin (I2C)	4 pin (VCC, GND, SDA, SCL)

Parameter	Spesifikasi
Suhu Operasional	-20°C hingga 70°C
Suhu Penyimpanan	-30°C hingga 80°C
Kompatibilitas	Arduino, ESP32, ESP8266, Raspberry Pi

### 2.2.12 Internet Of Things

Konsep IoT adalah jaringan perangkat fisik yang terhubung ke internet. Perangkat ini meliputi sensor, aktuator, dan perangkat pintar yang terhubung melalui internet, yang memiliki kemampuan untuk berkomunikasi, mengumpulkan informasi, menganalisisnya, dan menggunakannya untuk menciptakan berbagai efisiensi, inovasi, dan pengalaman pengguna yang lebih baik. Dampak yang akan dibawa IoT bagi kehidupan dan pekerjaan manusia, jika sesuai dengan potensinya, akan melampaui pengaruh Internet sejauh ini. IoT memungkinkan pemantauan jarak jauh, pengelolaan proses industri, dan lebih banyak lagi otomatisasi yang digerakkan oleh TIK. Dalam hal ini, faktor pendorong revolusi ini adalah kemampuan perangkat untuk mengumpulkan data dalam jumlah besar dan menganalisisnya untuk membentuk wawasan baru di dunia buatan manusia dan alam. IoT bertindak sebagai enabler untuk otomatisasi proses, peningkatan efisiensi operasi, dan model bisnis baru yang digerakkan oleh data. Di sisi lain, teknologi ini akan menciptakan peluang besar bagi bidang-bidang inovatif, di antaranya adalah ekonomi dan bisnis digital. Namun, dalam penerapan IoT, perhatian harus diberikan pada masalah keamanan, privasi data, dan peraturan jika keberlanjutan dan keamanannya ingin dicapai.

### 2.2.13 Logic Level Shifter



Gambar 2. 13 Logic Level Shifter

*Logic level shifter* adalah komponen yang berfungsi untuk menyesuaikan perbedaan level tegangan logika antara dua perangkat elektronik yang tidak kompatibel secara langsung [14]. Logic Level Shifter (LLS) merupakan rangkaian elektronika yang digunakan untuk mengubah level logika tegangan antar perangkat yang memiliki tegangan kerja berbeda. Pada penelitian ini, Logic Level Shifter digunakan sebagai penghubung komunikasi serial antara modul QR Code Scanner GM67 yang bekerja pada level logika 5V dengan mikrokontroler ESP32 yang menggunakan level logika 3,3V.

ESP32 memiliki batas tegangan masukan maksimum sekitar 3,3V pada pin GPIO. Apabila sinyal keluaran GM67 sebesar 5V dihubungkan secara langsung ke pin ESP32, maka dapat menyebabkan pembacaan data yang tidak stabil bahkan berpotensi merusak mikrokontroler. Oleh karena itu diperlukan Logic Level Shifter untuk menurunkan level logika dari 5V menjadi 3,3V sehingga komunikasi data dapat berlangsung dengan aman.

Logic Level Shifter yang digunakan pada penelitian ini memiliki dua sisi tegangan, yaitu High Voltage (HV) dan Low Voltage (LV). Sisi HV dihubungkan ke perangkat yang menggunakan logika 5V, sedangkan sisi LV dihubungkan ke perangkat yang menggunakan logika 3,3V. Proses konversi level logika dilakukan secara otomatis menggunakan MOSFET BSS138 sehingga data serial dapat diteruskan tanpa mengubah isi informasi yang dikirimkan.[16].

Tabel 2. 10 Spesifikasi Logic Level Shifter

Parameter	Spesifikasi
Jenis Modul	Logic Level Converter / Level Shifter
Tipe Translasi	Bidirectional (dua arah)
Jumlah Channel	4 Channel independen
Metode Translasi	MOSFET-based level shifting
IC Utama	BSS138 MOSFET (umumnya)
Tegangan Sisi Rendah (LV)	1.8V – 3.3V
Tegangan Sisi Tinggi (HV)	3.3V – 5V
Tegangan Maksimum	Hingga 5V (HV side)

Parameter	Spesifikasi
Arus Maksimum per Channel	$\pm 50$ mA
Kecepatan Switching	Hingga 2 MHz (tergantung beban)
Tipe Sinyal	Digital (TTL/CMOS)
Arah Sinyal	Otomatis mendeteksi arah (auto-direction)
Pull-up Resistor	Terintegrasi (biasanya 10k $\Omega$ )
Jumlah Pin	10 Pin (HV, LV, GND, HV1–HV4, LV1–LV4)
Kompatibilitas Logika	1.8V, 2.5V, 3.3V, 5V
Dimensi Modul	$\pm 15$ mm $\times$ 13 mm
Aplikasi Umum	I2C, UART, SPI, GPIO digital
Konsumsi Daya	Sangat rendah (pasif, tergantung sinyal)

### 2.2.14 Arduino IDE



Gambar 2. 14 Logo Arduino IDE

Arduino IDE (Integrated Development Environment) merupakan perangkat lunak open-source yang dirancang sebagai sarana pemrograman untuk berbagai jenis mikrokontroler Arduino. Lingkungan pengembangan ini memiliki antarmuka yang sederhana serta ramah pengguna, sehingga memudahkan proses menulis, mengedit, menyusun, hingga mengunggah program (sketch) langsung ke papan mikrokontroler melalui koneksi USB.

Arduino IDE dikembangkan menggunakan bahasa pemrograman Java dengan dukungan pustaka C/C++ (wiring), yang berfungsi untuk menyederhanakan interaksi dengan perangkat keras. Melalui library tersebut, pengguna dapat dengan

mudah membaca data sensor, mengontrol aktuator seperti motor, relay, maupun LED, serta mengatur komunikasi serial antar perangkat.

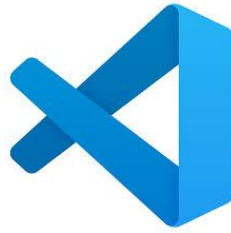
Beberapa fungsi utama Arduino IDE antara lain:

- Menyusun dan menuliskan kode program sesuai logika yang diinginkan.
- Mengompilasi kode menjadi instruksi biner yang dapat dieksekusi oleh mikrokontroler.
- Mengunggah program ke papan mikrokontroler seperti Arduino Uno, Mega, Nano, hingga ESP32.
- Mendukung penambahan board lain melalui fitur Board Manager, sehingga fleksibel untuk berbagai platform.

Struktur dasar pemrograman Arduino terdiri dari dua bagian utama, yaitu `setup()` dan `loop()`. Fungsi `setup()` dijalankan sekali saat papan pertama kali menyala atau setelah reset, umumnya digunakan untuk inisialisasi seperti mengatur pin input/output atau memulai komunikasi serial. Sementara itu, fungsi `loop()` berjalan berulang-ulang selama perangkat menyala, sehingga memungkinkan sistem untuk membaca sensor, memberikan output, serta menjalankan logika kontrol secara terus-menerus.

Selain struktur program, Arduino IDE mendukung penggunaan variabel dengan berbagai tipe data seperti `int`, `long`, `boolean`, `float`, maupun `char`, sesuai kebutuhan penyimpanan nilai. IDE ini juga dilengkapi berbagai fungsi bawaan, misalnya fungsi analog I/O (`analogRead()`, `analogWrite()`), fungsi waktu (`millis()`, `delay()`), serta fungsi matematika (`min()`, `max()`, `abs()`, `sqrt()`, `pow()`) yang memudahkan pengembangan aplikasi. Dengan kemudahan antarmuka, ketersediaan pustaka, dan fleksibilitas integrasi perangkat keras, Arduino IDE menjadi salah satu platform yang sangat populer dalam pengembangan sistem tertanam (embedded system) dan aplikasi IoT.

### 2.2.15 Visual Studio Code



Gambar 2. 15 Logo Visual Studio Code

Visual Studio Code (VS Code) merupakan perangkat lunak penyunting kode sumber (source code editor) yang dikembangkan oleh Microsoft dan bersifat open-source. Aplikasi ini dirancang untuk mendukung pengembangan berbagai bahasa pemrograman dengan performa ringan, antarmuka yang fleksibel, serta dukungan ekstensi (extensions) yang sangat luas. Visual Studio Code banyak digunakan dalam pengembangan aplikasi berbasis web, sistem embedded, serta proyek Internet of Things (IoT) karena kemampuannya mengintegrasikan berbagai tools dalam satu lingkungan kerja terpadu.

Secara arsitektural, Visual Studio Code berbasis pada teknologi Electron, yang memungkinkan aplikasi berjalan lintas platform seperti Windows, Linux, dan macOS. Editor ini mendukung fitur-fitur penting dalam proses pengembangan perangkat lunak, antara lain syntax highlighting, IntelliSense (auto-completion dan saran kode), debugging, version control (integrasi Git), terminal terintegrasi, serta manajemen proyek berbasis folder workspace. Fitur IntelliSense memungkinkan pengembang menulis kode secara lebih cepat dan minim kesalahan karena sistem memberikan rekomendasi struktur sintaks dan parameter fungsi secara otomatis.

Dalam konteks pengembangan sistem berbasis web dan IoT, Visual Studio Code sering digunakan untuk menulis dan mengelola file program seperti PHP, HTML, CSS, JavaScript, serta konfigurasi API yang berfungsi sebagai penghubung antara server dan perangkat keras. Selain itu, VS Code juga dapat dikonfigurasi untuk pengembangan mikrokontroler melalui ekstensi seperti PlatformIO atau integrasi dengan Arduino CLI. Kemampuan ini menjadikan Visual Studio Code sebagai lingkungan pengembangan terpadu yang mampu menangani sisi front-end, back-end, maupun embedded system dalam satu platform [17].

### 2.2.16 Database

Database merupakan komponen penting dalam sistem pengisian galon otomatis berbasis website karena berfungsi sebagai pusat penyimpanan, pengelolaan, dan validasi data transaksi. Pada sistem ini digunakan sistem manajemen basis data relasional berbasis MySQL/MariaDB yang diakses melalui bahasa pemrograman SQL (Structured Query Language). Database digunakan untuk menyimpan informasi token transaksi, volume air yang dibeli, status penggunaan, serta waktu transaksi.

Struktur basis data dirancang dengan pendekatan relasional untuk memastikan integritas dan konsistensi data. Tabel utama yang digunakan adalah tabel *tokens* yang berfungsi menyimpan seluruh transaksi pembelian air. Setiap baris data memiliki atribut seperti id sebagai primary key, token\_code sebagai identitas unik yang dihasilkan sistem, volume sebagai jumlah air dalam satuan mililiter, status untuk menandai apakah token masih aktif atau sudah digunakan, serta purchase\_date untuk mencatat waktu transaksi secara otomatis.

Dalam implementasinya, database mendukung operasi CRUD (Create, Read, Update, Delete). Proses *create* terjadi saat pengguna melakukan pembelian dan sistem menghasilkan token baru. Operasi *read* dilakukan saat ESP32 mengirimkan token ke API untuk divalidasi. Operasi *update* digunakan untuk mengubah status token dari “active” menjadi “used” setelah proses pengisian selesai. Sementara itu, operasi *delete* dapat dilakukan oleh administrator melalui dashboard untuk manajemen data.

Selain sebagai media penyimpanan transaksi, database juga berfungsi sebagai sumber data monitoring. Sistem administrator memanfaatkan query agregasi SQL seperti SUM() dan CASE WHEN untuk menghitung total volume pembelian dan total realisasi pengambilan air. Dengan demikian, database tidak hanya berperan sebagai penyimpan data, tetapi juga sebagai pendukung analisis dan pelaporan operasional sistem secara real-time.

### 2.2.17 Alwaysdata.net



Gambar 2. 16 Logo Alwaysdata.net

Alwaysdata.net merupakan penyedia layanan cloud hosting yang menerapkan model Platform as a Service (PaaS). Dalam konsep komputasi awan (cloud computing), PaaS adalah model layanan yang menyediakan lingkungan pengembangan dan eksekusi aplikasi tanpa pengguna harus mengelola infrastruktur fisik seperti server, sistem operasi, maupun konfigurasi jaringan secara langsung. Dengan pendekatan ini, pengembang cukup berfokus pada pengelolaan aplikasi dan basis data, sementara pengelolaan server dilakukan oleh penyedia layanan.

Secara arsitektural, layanan hosting berbasis PaaS menyediakan sumber daya komputasi yang berjalan pada infrastruktur virtualisasi di pusat data (data center). Infrastruktur ini umumnya mendukung bahasa pemrograman seperti PHP, Python, atau Node.js serta sistem manajemen basis data relasional seperti MySQL atau MariaDB. Alwaysdata mendukung eksekusi skrip PHP dan pengelolaan basis data MySQL/MariaDB melalui panel administrasi berbasis web, sehingga memudahkan proses konfigurasi domain, pengaturan database, serta manajemen file.

Salah satu aspek penting dalam layanan hosting modern adalah ketersediaan (availability) dan keandalan (reliability). Konsep uptime 24/7 mengacu pada kemampuan server untuk tetap aktif dan dapat diakses tanpa gangguan dalam jangka waktu yang panjang. Dalam sistem berbasis Internet of Things (IoT), ketersediaan server menjadi faktor krusial karena perangkat keras seperti ESP32 bergantung pada komunikasi jaringan secara real-time untuk proses validasi dan sinkronisasi data.

Dari sisi keamanan, Alwaysdata menyediakan dukungan SSL/TLS (Secure Socket Layer / Transport Layer Security) yang memungkinkan penggunaan

protokol HTTPS. HTTPS bekerja dengan mengenkripsi data yang dikirimkan antara klien dan server menggunakan mekanisme kriptografi kunci publik. Proses ini diawali dengan handshake protocol, di mana server mengirimkan sertifikat digital untuk diverifikasi oleh klien. Setelah proses verifikasi berhasil, komunikasi data dilakukan dalam bentuk terenkripsi. Mekanisme ini bertujuan untuk mencegah penyadapan dan serangan seperti Man-in-the-Middle (MITM) [18].

Selain itu, pengelolaan file pada layanan hosting umumnya dilakukan menggunakan protokol FTP (File Transfer Protocol) atau SFTP. FTP merupakan protokol standar yang memungkinkan proses transfer file dari komputer lokal ke server melalui jaringan TCP/IP. Dalam praktiknya, aplikasi klien seperti FileZilla digunakan untuk melakukan autentikasi dan pengunggahan file ke direktori publik server.

## **BAB III**

### **PERANCANGAN TUGAS AKHIR**

#### **3.1 Operasional Sistem**

Sistem pengisian galon otomatis dirancang untuk bekerja secara terintegrasi dengan memanfaatkan mikrokontroler ESP32 sebagai pusat kendali utama. Tujuan operasional sistem ini adalah untuk melakukan proses pengisian air galon secara otomatis, akurat, dan terdokumentasi, mulai dari identifikasi pengguna hingga pencatatan data pengisian ke dalam basis data berbasis web. Sistem ini menggabungkan perangkat input berupa QR Code Scanner GM67, sensor ultrasonik HC-SR04, dan sensor flowmeter YF-S201, serta perangkat output berupa pompa air DC, modul relay, dan antarmuka tampilan.

Proses operasional sistem diawali dengan pemindaian QR Code oleh QR Code Scanner GM-67. QR Code berfungsi sebagai media identifikasi transaksi yang memuat token unik yang telah diterbitkan oleh sistem setelah pengguna memilih volume air yang diinginkan dan melakukan konfirmasi pembelian melalui website. Data token hasil pemindaian dikirimkan ke ESP32 melalui komunikasi serial (UART), kemudian diteruskan ke server melalui koneksi internet untuk divalidasi terhadap database. Apabila token dinyatakan valid dan berstatus aktif, server akan mengembalikan informasi volume pengisian yang telah ditentukan sebelumnya oleh pengguna. Apabila token tidak terdaftar, sudah pernah digunakan, atau koneksi ke server gagal, maka proses pengisian tidak akan dilanjutkan.

Setelah QR Code berhasil diverifikasi, sistem melakukan pendeteksian keberadaan galon menggunakan sensor ultrasonik HC-SR04. Sensor ini memastikan bahwa galon berada pada posisi pengisian yang benar sebelum pompa air diaktifkan. Apabila galon terdeteksi, ESP32 akan mengaktifkan modul relay untuk menyalakan pompa air DC. Selama proses pengisian berlangsung, sensor flowmeter YF-S201 mengukur debit dan volume air yang mengalir menuju galon secara real-time. Ketika volume air telah mencapai nilai yang ditentukan sesuai

hasil identifikasi QR Code, ESP32 akan mematikan pompa secara otomatis dan mengakhiri proses pengisian.

Seluruh data proses pengisian, meliputi identitas QR Code, jenis galon (2 L, 5 L, atau 15 L), total volume air yang terpakai, serta waktu dan tanggal pengambilan, dikirimkan ke sistem web monitoring dan disimpan ke dalam database. Dengan mekanisme ini, sistem tidak hanya berfungsi sebagai alat pengisian otomatis, tetapi juga sebagai sistem pencatatan dan monitoring penggunaan air secara terstruktur.

### **3.2 Identifikasi Sistem**

Identifikasi sistem dilakukan untuk menentukan variabel, komponen, serta parameter penting yang terlibat dalam perancangan dan pengoperasian sistem pengisian galon otomatis. Variabel utama yang diamati dalam sistem ini meliputi volume air yang diisikan, debit aliran air, status keberadaan galon, identitas pengguna, serta waktu pengisian.

Komponen utama sistem terdiri dari ESP32 sebagai pengendali utama, QR Code Scanner GM67 sebagai perangkat identifikasi, sensor ultrasonik HC-SR04 sebagai pendeteksi galon, sensor flowmeter YF-S201 sebagai pengukur debit dan volume air, serta pompa air DC sebagai aktuator utama. Selain itu, sistem juga dilengkapi dengan modul relay sebagai pengendali beban dan sistem web monitoring yang terhubung dengan database sebagai media pencatatan data.

#### **A. Analisis Kebutuhan Sistem**

Kebutuhan utama sistem pengisian galon otomatis adalah adanya mekanisme pengisian air yang akurat, aman, dan dapat dipantau secara real-time. Sistem diharapkan mampu mengidentifikasi jenis galon secara otomatis melalui QR Code, mengukur volume air dengan presisi menggunakan sensor flowmeter, serta mencatat seluruh aktivitas pengisian ke dalam database. Selain itu, diperlukan antarmuka web yang dapat menampilkan data transaksi pengisian, statistik penggunaan air, dan riwayat pengambilan galon.

#### **B. Identifikasi Permasalahan**

Permasalahan yang diidentifikasi dalam proses pengisian galon manual antara lain:

1. Ketidaktepatan volume pengisian akibat pengendalian manual.
2. Tidak adanya sistem identifikasi galon atau pengguna secara otomatis.
3. Tidak tersedianya data penggunaan air dan waktu pengambilan galon.
4. Risiko pemborosan air akibat kesalahan operasional.

### **C. Tujuan Informasi**

Informasi yang dikumpulkan bertujuan untuk mendukung perancangan sistem otomatis yang mampu mengendalikan proses pengisian galon berdasarkan data sensor dan identifikasi QR Code. Informasi tersebut meliputi karakteristik aliran air, kapasitas galon, respon pompa terhadap kontrol relay, serta struktur data yang dibutuhkan untuk penyimpanan pada database.

### **3.3 Jenis dan Sumber Data**

Data yang digunakan dalam penelitian ini terdiri dari data primer dan data sekunder. Data primer diperoleh dari hasil pengujian langsung terhadap sistem pengisian galon otomatis, meliputi pembacaan debit air, volume pengisian, status deteksi galon, serta waktu proses pengisian. Data primer juga mencakup hasil pemindaian QR Code dan respons sistem terhadap berbagai kondisi operasional.

Data sekunder diperoleh dari literatur ilmiah, jurnal penelitian, tugas akhir, serta datasheet komponen yang digunakan, seperti ESP32, QR Code Scanner GM67, sensor flowmeter YF-S201, dan sensor ultrasonik HC-SR04. Data ini digunakan sebagai dasar teoritis dan teknis dalam perancangan sistem serta pengembangan logika kontrol dan web monitoring.

### **3.4 Prosedur Pengumpulan Data**

Pengumpulan data dilakukan secara sistematis selama proses perancangan, implementasi, dan pengujian sistem. Data dikumpulkan melalui pembacaan sensor yang terintegrasi dengan ESP32, kemudian dikirimkan ke sistem web monitoring dan disimpan ke dalam database. Parameter yang dicatat meliputi jenis galon yang terdeteksi (2 L, 5 L, atau 15 L), volume air yang terpakai, debit aliran, serta waktu dan tanggal pengisian.

Proses pengumpulan data juga mencakup observasi langsung terhadap kinerja sistem, pencatatan hasil pengujian, serta validasi data antara pembacaan sensor dan volume aktual. Seluruh data yang terkumpul digunakan sebagai bahan analisis untuk mengevaluasi kinerja sistem dan akurasi pengisian.

### **3.5 Pengujian dan Analisa**

Pengujian sistem dilakukan untuk memastikan bahwa seluruh komponen sistem pengisian galon otomatis dapat bekerja sesuai dengan fungsi dan logika yang telah dirancang. Pengujian meliputi pengujian QR Code Scanner GM67 dalam membaca dan memverifikasi QR Code, pengujian sensor ultrasonik HC-SR04 dalam mendeteksi keberadaan galon, serta pengujian sensor flowmeter YF-S201 dalam mengukur debit dan volume air.

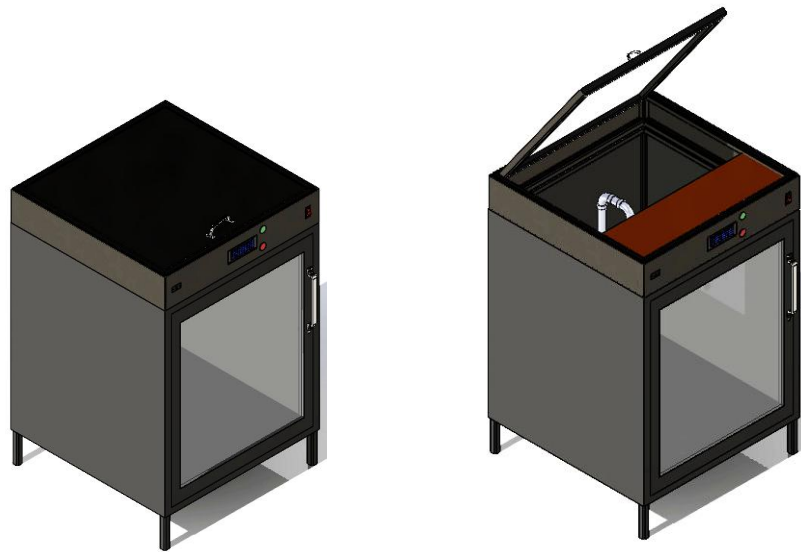
Pengujian juga dilakukan terhadap sistem database dan web monitoring untuk memastikan data pengisian tersimpan dengan benar, termasuk informasi jenis galon, total volume air, dan waktu pengambilan. Analisis dilakukan dengan membandingkan volume yang tercatat di sistem dengan volume aktual pengisian, serta mengamati respon sistem terhadap kondisi normal maupun kondisi kesalahan. Hasil pengujian digunakan untuk menilai tingkat keandalan, akurasi, dan efektivitas sistem dalam mendukung proses pengisian galon otomatis.

### 3.6 Perancangan Alat

#### 3.6.1 Desain 3D Alat



Gambar 3. 1 Tampak Depan Kotak Pengisian



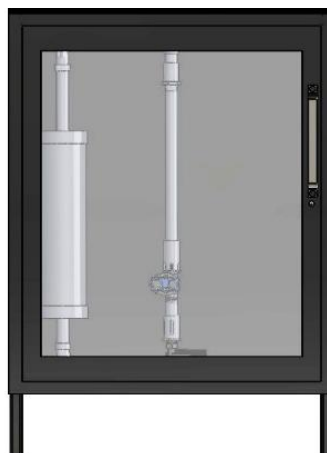
Gambar 3. 2 Tampak Samping Kotak Pengisian

Kotak pengisian galon otomatis pada sistem ini berfungsi sebagai enclosure mekanik yang menampung dan melindungi seluruh komponen sistem pengisian air secara terintegrasi, baik komponen elektronik maupun mekanik. Kotak dirancang dengan dimensi  $50 \times 50 \times 100$  cm dalam bentuk vertikal, dilengkapi rangka penyangga di bagian bawah untuk menopang beban galon serta memastikan kestabilan struktur selama proses pengisian berlangsung. Desain mekanik ini mempertimbangkan aspek kekuatan struktur, ergonomi pengguna, dan keamanan operasional, sehingga mampu meminimalkan risiko kebocoran air maupun kerusakan komponen internal.



Gambar 3. 3 Bagian Atas Pada kotak

Pada bagian atas kotak terdapat panel kontrol, yang berfungsi sebagai antarmuka pengguna. Bagian atas pada panel ini menampung Sensor Qr, layar LCD, lampu untuk indikator, dan saklar on off. Penempatan panel di bagian atas bertujuan agar informasi sistem mudah dibaca dan proses identifikasi galon melalui QR Code dapat dilakukan secara ergonomis. Seluruh komponen panel terhubung langsung dengan mikrokontroler ESP32 yang ditempatkan di dalam kotak sebagai pusat kendali sistem.

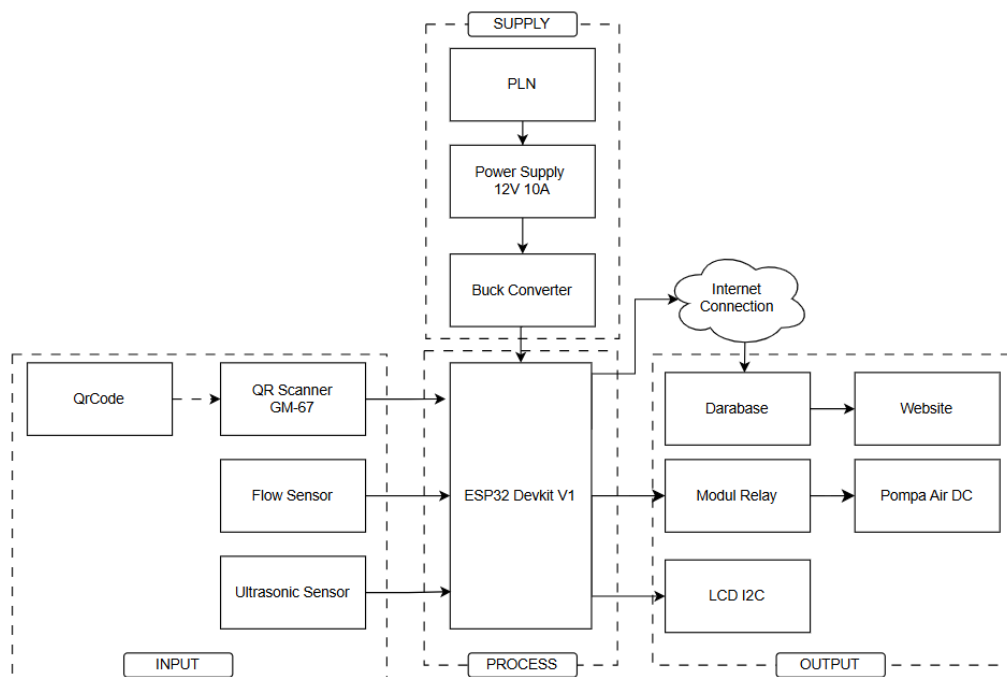


Gambar 3. 4 Bagian Tengah Pada Kotak

Bagian tengah kotak merupakan ruang pengisian galon, tempat galon diletakkan saat proses pengisian berlangsung. Di dalam ruang ini terdapat jalur pipa pengisian air yang terhubung ke pompa air DC. Pipa diarahkan secara vertikal ke bawah menuju mulut galon untuk memastikan aliran air masuk secara tepat, mengurangi percikan, serta meningkatkan akurasi pengisian volume. Pada area ini juga dipasang sensor ultrasonik HC-SR04 yang berfungsi mendeteksi keberadaan galon, sehingga sistem hanya akan mengaktifkan proses pengisian apabila galon terdeteksi berada pada posisi yang benar.

Pada sisi belakang kotak terdapat lubang masukan air berdiameter  $\frac{1}{2}$  inci, yang terhubung ke sumber air eksternal. Air dari saluran ini dialirkan menuju pompa air DC 12 V dengan daya 22 W, yang memiliki kapasitas aliran hingga 800 liter/jam. Pompa ini berfungsi sebagai aktuator utama yang mengalirkan air ke dalam galon dan dikendalikan oleh ESP32 melalui modul relay. Seluruh komponen elektronik seperti ESP32, modul relay, catu daya, dan buck converter ditempatkan di dalam ruang enclosure tertutup untuk melindungi dari paparan air, debu, dan gangguan lingkungan. Dengan rancangan mekanik tersebut, kotak pengisian galon mampu mendukung sistem pengisian galon otomatis yang aman, rapi, stabil, dan andal.

### 3.6.2 Blok Diagram



Gambar 3. 5 Blok Diagram

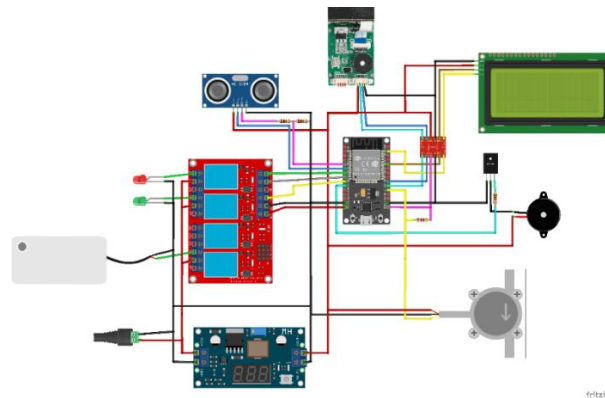
### 3.7 Cara Kerja Blok Diagram

1. Sumber Daya Listrik (PLN). Sistem memperoleh sumber daya listrik utama dari jaringan PLN. Tegangan AC dari PLN digunakan sebagai sumber daya awal untuk mengoperasikan seluruh sistem pengisian galon otomatis.
2. Power Supply 12V 10A. Tegangan AC dari PLN diubah menjadi tegangan DC sebesar 12 V menggunakan power supply 12V 10A. Power supply ini berfungsi menyediakan suplai daya DC yang stabil dan memiliki kapasitas arus yang cukup untuk menggerakkan beban utama seperti pompa air DC serta komponen elektronik lainnya.
3. Buck Converter. Tegangan keluaran 12 V dari power supply kemudian diturunkan menggunakan buck converter untuk menghasilkan tegangan DC yang sesuai dengan kebutuhan komponen logika, seperti ESP32, sensor, dan modul lainnya. Buck converter berperan menjaga kestabilan tegangan agar tidak merusak rangkaian elektronik.

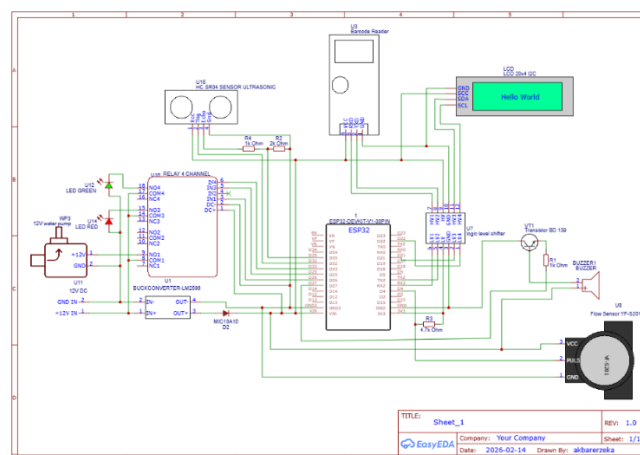
4. QR Code sebagai Media Identifikasi. QR Code berisi data identitas pengguna atau kategori galon. QR Code ini berfungsi sebagai input awal sistem yang menentukan parameter pengisian air, seperti volume yang akan diisikan.
5. QR Code Scanner GM67. QR Code Scanner GM67 membaca data yang terdapat pada QR Code menggunakan prinsip pencitraan optik. Data hasil pemindaian kemudian dikirimkan ke ESP32 Devkit V1 melalui komunikasi serial (UART) sebagai data input sistem.
6. Sensor Flowmeter. Sensor flowmeter berfungsi mengukur debit dan volume air yang mengalir menuju galon. Sensor ini menghasilkan pulsa yang jumlahnya sebanding dengan aliran air. Pulsa tersebut dikirim ke ESP32 untuk dihitung dan dikonversi menjadi nilai volume air.
7. Sensor Ultrasonik. Sensor ultrasonik digunakan untuk mendeteksi keberadaan galon pada area pengisian. Sensor ini mengirimkan data jarak ke ESP32 untuk memastikan bahwa galon berada pada posisi yang benar sebelum proses pengisian dimulai.
8. ESP32 Devkit V1 (Blok Proses). ESP32 Devkit V1 berperan sebagai pusat kendali sistem. Mikrokontroler ini menerima seluruh data dari QR Code Scanner GM67, sensor flow, dan sensor ultrasonik. Berdasarkan data tersebut, ESP32 memproses logika sistem untuk menentukan apakah pengisian dapat dilakukan, berapa volume air yang harus diisikan, serta kapan proses pengisian harus dihentikan. Selain itu, ESP32 juga mengelola komunikasi data dengan sistem monitoring.
9. Modul Relay. Modul relay berfungsi sebagai saklar elektronik yang dikendalikan oleh ESP32. Relay ini menghubungkan dan memutuskan aliran daya ke pompa air DC. Penggunaan relay memungkinkan ESP32 mengendalikan beban bertegangan dan berarus besar secara aman.
10. Pompa Air DC. Pompa air DC berfungsi sebagai aktuator utama yang mengalirkan air dari penampungan ke dalam galon. Pompa diaktifkan dan dinonaktifkan oleh ESP32 melalui modul relay berdasarkan hasil perhitungan volume air dari sensor flow.

11. LCD I2C. LCD I2C digunakan sebagai antarmuka tampilan lokal untuk menampilkan informasi sistem secara real-time, seperti status pembacaan QR Code, proses pengisian, volume air yang telah terisi, dan kondisi sistem.
12. Alur Kerja Sistem Secara Keseluruhan. Secara keseluruhan, sistem bekerja dengan alur: sumber daya → pemindaian QR Code → validasi galon → pengisian air → pengukuran volume → penghentian pengisian → penampilan dan pengiriman data. Seluruh proses dikendalikan secara terpusat oleh ESP32 Devkit V1.

### 3.8 Wiring Diagram

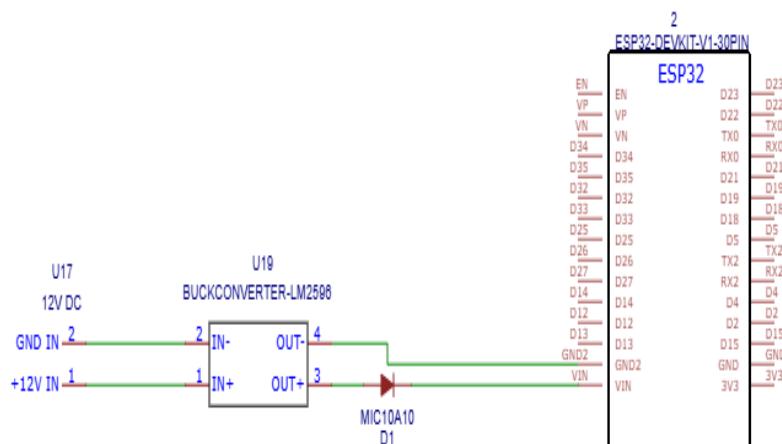


Gambar 3. 6 Rangkaian Secara Keseluruhan



Gambar 3. 7 Wiring Diagram Secara Keseluruhan

### 3.8.1 Rangkaian Catu Daya



Gambar 3. 8 Rangkaian Catu Daya

Rangkaian pada gambar merupakan sistem catu daya untuk mensuplai mikrokontroler ESP32 menggunakan sumber tegangan 12V DC. Sumber tegangan 12V diperoleh dari adaptor atau power supply eksternal yang terhubung melalui terminal input (U17). Tegangan ini memiliki dua jalur utama, yaitu jalur positif (+12V IN) dan jalur ground (GND IN).

Karena ESP32 bekerja pada tegangan 5V+ (melalui pin VIN) atau 3.3V internal, maka tegangan 12V perlu diturunkan terlebih dahulu menggunakan modul buck converter LM2596 (U19). Buck converter merupakan regulator switching tipe step-down yang mampu menurunkan tegangan dengan efisiensi tinggi. Pada rangkaian ini, pin IN+ dan IN- menerima masukan 12V DC dari sumber, kemudian modul LM2596 menurunkan tegangan tersebut menjadi 5V DC pada pin OUT+ dan OUT-.

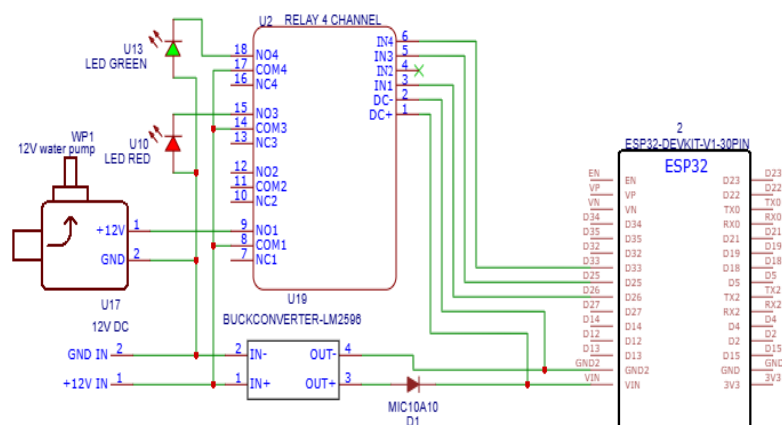
Setelah tegangan diturunkan menjadi 5V, jalur output positif dilewatkan melalui dioda MIC10A10 (D1). Dioda ini berfungsi sebagai proteksi polaritas terbalik dan pencegah arus balik (reverse current protection). Dengan adanya dioda ini, jika terjadi kesalahan polaritas atau lonjakan arus dari sisi beban, sistem catu daya tetap terlindungi.

Tegangan 5V yang telah melalui dioda kemudian masuk ke pin VIN pada ESP32 DevKit V1. Pin GND dari buck converter dihubungkan langsung ke pin GND ESP32 untuk memastikan referensi tegangan yang sama. Dari pin VIN inilah

regulator internal ESP32 akan menghasilkan tegangan 3.3V untuk kebutuhan logika mikrokontroler.

Dengan konfigurasi ini, sistem memperoleh suplai daya yang stabil, aman, dan sesuai dengan spesifikasi kerja ESP32.

### 3.8.2 Rangkaian Relay



Gambar 3. 9 Rangkaian Relay

Rangkaian pada gambar merupakan bagian sistem kendali aktuator yang mengatur proses pengoperasian pompa air menggunakan modul relay 4 channel yang dikendalikan oleh mikrokontroler ESP32. Sistem ini dirancang untuk memisahkan jalur daya tinggi (12V) dengan jalur logika kontrol (5V dan 3,3V) guna meningkatkan stabilitas dan keamanan rangkaian secara keseluruhan. Sumber tegangan utama berasal dari power supply 12V DC yang terhubung ke terminal input rangkaian. Tegangan 12V ini memiliki dua fungsi utama, yaitu sebagai sumber daya untuk pompa air DC dan sebagai sumber masukan bagi modul buck converter LM2596 yang berfungsi menurunkan tegangan menjadi 5V untuk kebutuhan sistem logika.

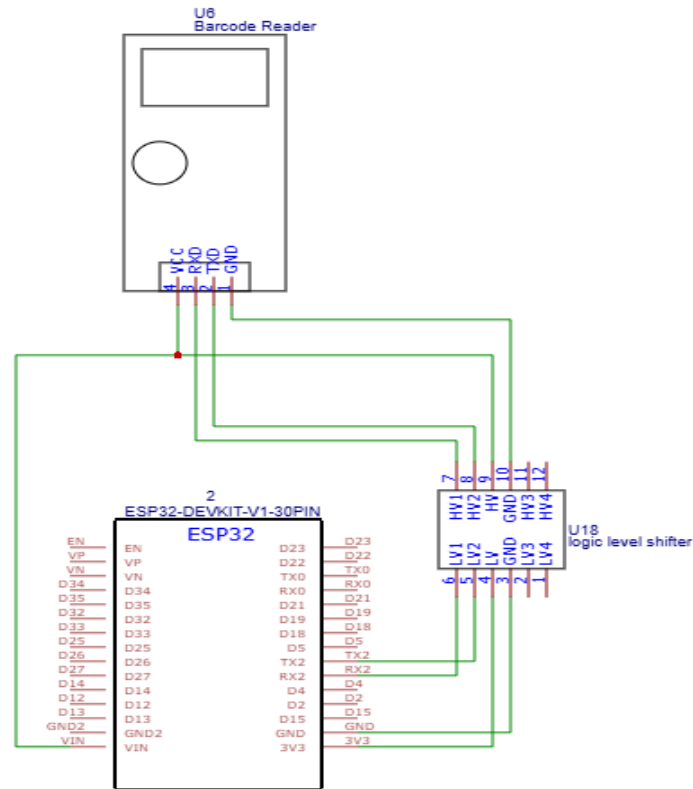
Modul relay 4 channel berperan sebagai saklar elektromekanis yang memungkinkan ESP32 mengendalikan beban bertegangan lebih tinggi menggunakan sinyal logika bertegangan rendah. Pin IN1 hingga IN4 pada modul relay terhubung langsung ke beberapa pin GPIO pada ESP32. Ketika ESP32 memberikan sinyal aktif pada salah satu pin input relay, maka kumparan

elektromagnet di dalam relay akan bekerja dan menghubungkan kontak COM (Common) dengan NO (Normally Open). Proses ini memungkinkan arus 12V mengalir menuju beban, dalam hal ini pompa air DC atau LED indikator. Sebaliknya, ketika sinyal kontrol dinonaktifkan, kontak relay kembali terbuka sehingga arus terputus dan beban berhenti bekerja.

Pompa air DC 12V pada rangkaian ini berfungsi sebagai aktuator utama yang mengalirkan air dari tangki menuju galon. Terminal positif pompa dihubungkan ke terminal NO relay, sedangkan terminal negatif pompa langsung terhubung ke ground 12V. Dengan konfigurasi ini, pompa hanya akan menerima tegangan ketika relay dalam kondisi aktif. Sistem ini memberikan kontrol penuh terhadap waktu dan durasi pengisian air yang dihitung berdasarkan pembacaan sensor flowmeter. Selain pompa, terdapat LED indikator merah dan hijau yang juga dikendalikan melalui relay sebagai penanda status sistem, misalnya kondisi siap, proses pengisian berlangsung, atau kesalahan sistem. Penggunaan LED sebagai indikator visual membantu pengguna maupun operator dalam memahami kondisi operasional alat secara langsung.

Sementara itu, tegangan 12V dari power supply juga masuk ke modul buck converter LM2596 untuk diturunkan menjadi 5V DC yang stabil. Modul buck converter bekerja menggunakan prinsip switching regulator sehingga memiliki efisiensi tinggi dan menghasilkan panas yang lebih rendah dibandingkan regulator linier. Tegangan 5V hasil konversi kemudian dilewatkan melalui dioda proteksi sebelum masuk ke pin VIN pada ESP32. Dioda ini berfungsi mencegah arus balik dan melindungi rangkaian dari kemungkinan kesalahan polaritas atau lonjakan arus dari sisi beban. Setelah masuk ke pin VIN, regulator internal ESP32 akan menurunkan tegangan tersebut menjadi 3,3V untuk keperluan logika mikrokontroler.

### 3.8.3 Rangkaian Qr Code Scanner GM-67



Gambar 3. 10 Rangkaian Sensor GM-67

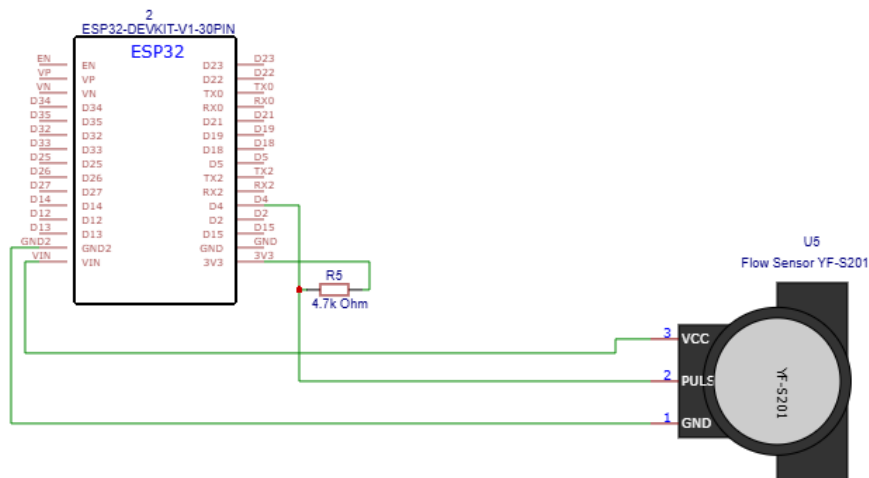
Rangkaian pada gambar mengintegrasikan modul QR Code Scanner (Barcode Reader GM67) dengan mikrokontroler ESP32 DevKit V1 menggunakan modul logic level shifter sebagai jembatan komunikasi. Implementasi logic level shifter (U18) menjadi krusial karena adanya perbedaan level tegangan logika antara kedua perangkat; modul GM67 beroperasi pada level logika 5V, sementara ESP32 bekerja pada level logika 3,3V. Tanpa penyesuaian level, sinyal 5V dari modul scanner dapat menyebabkan kerusakan permanen pada pin input (GPIO) mikrokontroler ESP32.

Modul GM67 menggunakan protokol komunikasi serial UART (Universal Asynchronous Receiver-Transmitter) melalui pin utama VCC, GND, TX, dan RX. Pin VCC dihubungkan ke sumber daya 5V, sementara pin GND berfungsi sebagai referensi ground bersama (common ground). Dalam proses transmisi data, sinyal dari pin TX GM67 yang membawa data hasil pemindaian diturunkan tegangannya oleh logic level shifter dari 5V menjadi 3,3V sebelum diterima oleh pin RX2 (GPIO16) pada ESP32. Sebaliknya, sinyal kontrol dari pin TX2 (GPIO17) ESP32

dinaikkan tegangannya dari 3,3V menjadi 5V untuk diteruskan ke pin RX modul GM67.

Logic level shifter yang digunakan bersifat bidirectional (dua arah), yang umumnya memanfaatkan konfigurasi transistor MOSFET sebagai sakelar cepat untuk melakukan konversi level tegangan tanpa menghambat laju baud rate komunikasi serial. Jalur LV (Low Voltage) pada modul ini dihubungkan ke penyedia daya 3,3V, sedangkan jalur HV (High Voltage) dihubungkan ke sumber daya 5V. Pada sisi perangkat lunak, ESP32 dikonfigurasi menggunakan port UART2 untuk berkomunikasi dengan GM67. Ketika sebuah kode QR berhasil dipindai, modul GM67 akan mengirimkan data string token yang kemudian ditangkap oleh ESP32 melalui jalur RX2. Data tersebut selanjutnya divalidasi oleh mikrokontroler melalui koneksi internet menuju database server.

### 3.8.4 Rangkaian Sensor Flowmeter



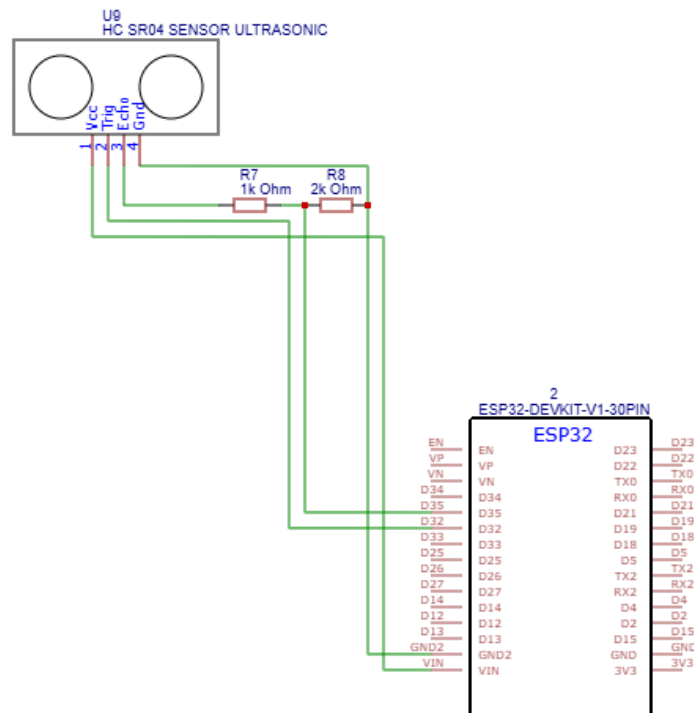
Gambar 3. 11 Rangkaian Sensor Flowmeter

Rangkaian pada gambar menunjukkan koneksi antara sensor flowmeter YF-S201 dengan mikrokontroler ESP32 DevKit V1. Sensor flowmeter YF-S201 merupakan sensor berbasis efek Hall yang digunakan untuk mengukur laju aliran air dengan cara menghasilkan pulsa digital setiap kali rotor internal berputar akibat aliran fluida. Jumlah pulsa yang dihasilkan sebanding dengan volume air yang mengalir melalui sensor, sehingga mikrokontroler dapat menghitung total volume berdasarkan jumlah pulsa yang diterima.

Sensor YF-S201 memiliki tiga pin utama, yaitu VCC, GND, dan PULSE (output sinyal). Pada rangkaian ini, pin VCC sensor dihubungkan ke sumber tegangan 5V, sedangkan pin GND dihubungkan ke ground sistem yang sama dengan ESP32 untuk memastikan referensi tegangan yang konsisten. Pin PULSE merupakan keluaran sinyal digital berbentuk gelombang kotak (square wave) yang akan berubah antara kondisi HIGH dan LOW sesuai dengan putaran rotor sensor.

Sinyal PULSE dari YF-S201 kemudian dihubungkan ke salah satu pin GPIO ESP32 yang dikonfigurasi sebagai input interrupt. Pada gambar terlihat adanya resistor 4,7 k $\Omega$  (R6) yang terhubung sebagai resistor pull-up. Resistor ini berfungsi untuk memastikan bahwa jalur sinyal tetap berada pada kondisi logika HIGH ketika tidak ada pulsa yang dihasilkan. Karena output sensor YF-S201 bersifat open-collector, maka tanpa resistor pull-up, sinyal tidak akan dapat mencapai level logika HIGH yang stabil. Dengan adanya resistor 4,7 k $\Omega$ , tegangan sinyal dapat ditarik ke level logika yang sesuai dengan sistem, sehingga pembacaan pulsa menjadi lebih akurat dan bebas noise.

### 3.8.5 Rangkaian Ultrasonik

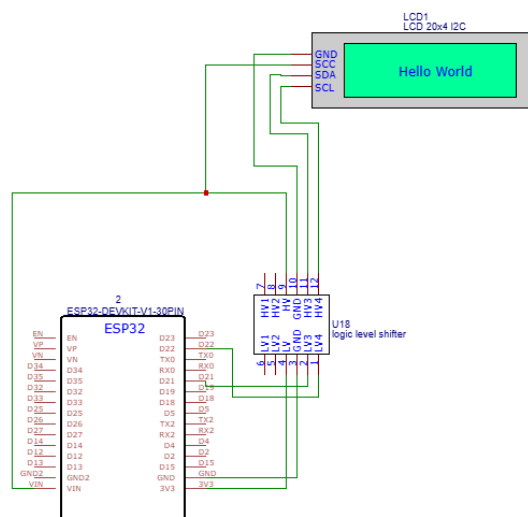


Gambar 3. 12 Rangkaian Sensor Ultrasonik

Sensor ultrasonik HC-SR04 dikonfigurasi sebagai perangkat input digital yang berfungsi untuk mendeteksi keberadaan galon dengan cara mengukur jarak antara sensor dan permukaan objek. Pada rangkaian ini, pin VCC sensor HC-SR04 dihubungkan ke sumber tegangan 5V untuk memenuhi kebutuhan tegangan operasional sensor. Pin GND sensor dihubungkan ke pin GND ESP32 guna menyamakan referensi ground sehingga proses pembacaan sinyal dapat berlangsung stabil dan akurat.

Pin Trigger pada sensor dihubungkan ke salah satu pin GPIO ESP32 yang dikonfigurasi sebagai output digital. Pin ini berfungsi untuk mengirimkan pulsa logika HIGH selama kurang lebih 10 mikrodetik sebagai perintah kepada sensor untuk memancarkan gelombang ultrasonik. Sementara itu, pin Echo dihubungkan ke pin GPIO ESP32 yang dikonfigurasi sebagai input digital. Karena sinyal keluaran Echo memiliki level tegangan 5V sedangkan ESP32 bekerja pada level logika 3,3V, maka pada rangkaian ditambahkan pembagi tegangan menggunakan resistor 1 k $\Omega$  dan 2 k $\Omega$  untuk menurunkan tegangan sinyal agar aman bagi pin input ESP32.

### 3.8.6 Rangkaian LCD20x4



Gambar 3. 13 Rangkaian LCD I2C 20x4

LCD 20x4 berbasis antarmuka I2C dikonfigurasi sebagai perangkat output digital yang berfungsi untuk menampilkan informasi sistem secara real-time,

seperti status pembacaan QR Code, proses pengisian, volume air, serta notifikasi kesalahan. Pada rangkaian ini, LCD berperan sebagai antarmuka visual utama antara sistem dan pengguna.

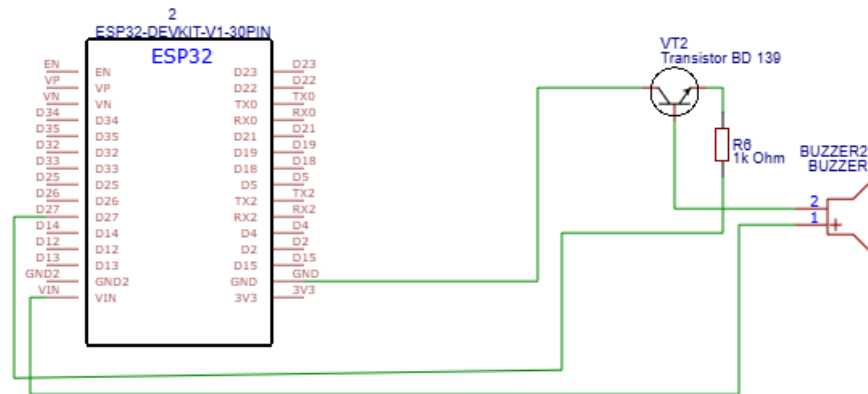
Pin VCC pada modul LCD dihubungkan ke sumber tegangan 5V untuk memenuhi kebutuhan catu daya modul, sedangkan pin GND dihubungkan ke ground sistem agar referensi tegangan antara perangkat tetap sama dan stabil. Komunikasi data dilakukan melalui dua jalur utama, yaitu SDA (Serial Data) dan SCL (Serial Clock) yang merupakan bagian dari protokol komunikasi I2C (Inter-Integrated Circuit).

Karena ESP32 bekerja pada level logika 3,3V sementara modul LCD I2C umumnya menggunakan level logika 5V, maka pada rangkaian ditambahkan modul logic level shifter sebagai pengaman dan penyesuai tegangan. Logic level shifter ini berfungsi menyesuaikan level tegangan dua arah (bidirectional level shifting), sehingga sinyal SDA dan SCL dari ESP32 yang berlevel 3,3V dapat diterjemahkan menjadi 5V untuk LCD, dan sebaliknya sinyal dari LCD tidak merusak pin ESP32.

Pada sisi ESP32, pin GPIO 21 dikonfigurasi sebagai jalur SDA dan GPIO 22 sebagai jalur SCL sesuai dengan konfigurasi standar I2C pada board ESP32 DevKit V1. Kedua pin ini terhubung ke sisi low-voltage (LV) pada modul logic level shifter. Sementara itu, sisi high-voltage (HV) pada level shifter terhubung ke pin SDA dan SCL LCD yang bekerja pada tegangan 5V.

Komunikasi I2C memungkinkan ESP32 mengirimkan data dalam bentuk byte secara serial untuk dikonversi menjadi karakter ASCII oleh IC kontroler LCD (umumnya kompatibel dengan HD44780 melalui backpack PCF8574). Dengan hanya menggunakan dua jalur komunikasi, sistem menjadi lebih efisien dalam penggunaan pin GPIO dibandingkan metode paralel.

### 3.8.7 Rangkaian Buzzer



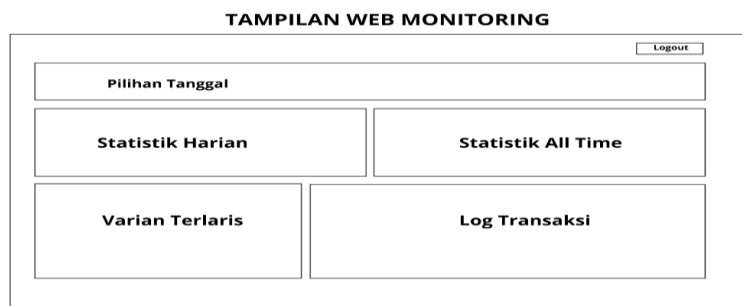
Gambar 3. 14 Rangkaian Buzzer

Buzzer pada rangkaian ini dikonfigurasi sebagai perangkat output yang berfungsi memberikan notifikasi suara pada kondisi tertentu, seperti token valid, proses pengisian dimulai, pengisian selesai, atau terjadi kesalahan sistem. Karena arus yang dibutuhkan buzzer relatif lebih besar dibandingkan kemampuan arus keluaran langsung dari pin GPIO ESP32, maka digunakan transistor BD139 sebagai penguat arus (current amplifier) sekaligus saklar elektronik.

Pada rangkaian, salah satu pin GPIO ESP32 dihubungkan ke basis transistor BD139 melalui resistor pembatas arus sebesar 1 k $\Omega$  (R6). Resistor ini berfungsi membatasi arus basis agar tidak melebihi batas aman transistor maupun pin GPIO ESP32. Ketika GPIO mengeluarkan logika HIGH (3,3V), arus kecil akan mengalir ke basis transistor sehingga transistor masuk ke kondisi saturasi (ON). Dalam kondisi ini, arus dapat mengalir dari kolektor ke emitor, sehingga buzzer mendapatkan suplai arus dan menghasilkan bunyi. Kaki kolektor transistor dihubungkan ke salah satu terminal buzzer, sedangkan terminal buzzer lainnya terhubung ke sumber tegangan positif. Emitor transistor dihubungkan ke ground sistem. Dengan konfigurasi ini, transistor bekerja sebagai saklar sisi bawah (low-side switch), yang menghubungkan buzzer ke ground saat aktif. Ketika GPIO berada pada logika LOW, tidak ada arus yang mengalir ke basis transistor, sehingga transistor berada pada kondisi cut-off (OFF). Dalam keadaan ini, jalur kolektor–emitor terbuka dan buzzer tidak mendapatkan jalur arus ke ground, sehingga buzzer tidak berbunyi.

### 3.9 Perancangan Software

#### 3.9.1 Tampilan Website



Gambar 3. 15 Tampilan Web Monitoring

Rancangan tampilan website pada sistem pengisian galon otomatis berfungsi sebagai antarmuka monitoring dan manajemen data berbasis web yang terintegrasi dengan mikrokontroler ESP32. Website ini dirancang untuk menampilkan informasi hasil proses pengisian galon secara terstruktur dan mudah dipahami oleh pengguna maupun operator. Seluruh data yang ditampilkan bersumber dari sistem IoT yang mengirimkan data pengisian secara real-time ke server.

Pada bagian utama tampilan web terdapat dashboard transaksi pengisian galon, yang menyajikan informasi detail mengenai aktivitas pengisian, seperti jenis atau ukuran galon yang diisi, jumlah air yang digunakan, serta waktu terjadinya transaksi. Informasi ini berfungsi sebagai catatan digital yang dapat digunakan untuk keperluan pencatatan penggunaan air, evaluasi sistem, maupun kebutuhan administrasi depot air minum. Tata letak dashboard dirancang agar informasi utama dapat diakses dengan cepat tanpa memerlukan navigasi yang kompleks.

Selain tampilan transaksi, website juga dilengkapi dengan ringkasan statistik pengisian, yang ditampilkan dalam bentuk jumlah pengisian berdasarkan ukuran galon serta grafik penggunaan air dalam satuan liter. Penyajian data dalam bentuk visual bertujuan untuk mempermudah analisis pola konsumsi air dan efisiensi sistem. Dengan adanya tampilan web ini, proses pengawasan dan pengelolaan data pengisian galon menjadi lebih sistematis, transparan, dan mendukung pengambilan keputusan berbasis data.

**TAMPILAN WEB UNTUK PEMILIHAN UKURAN GALON**

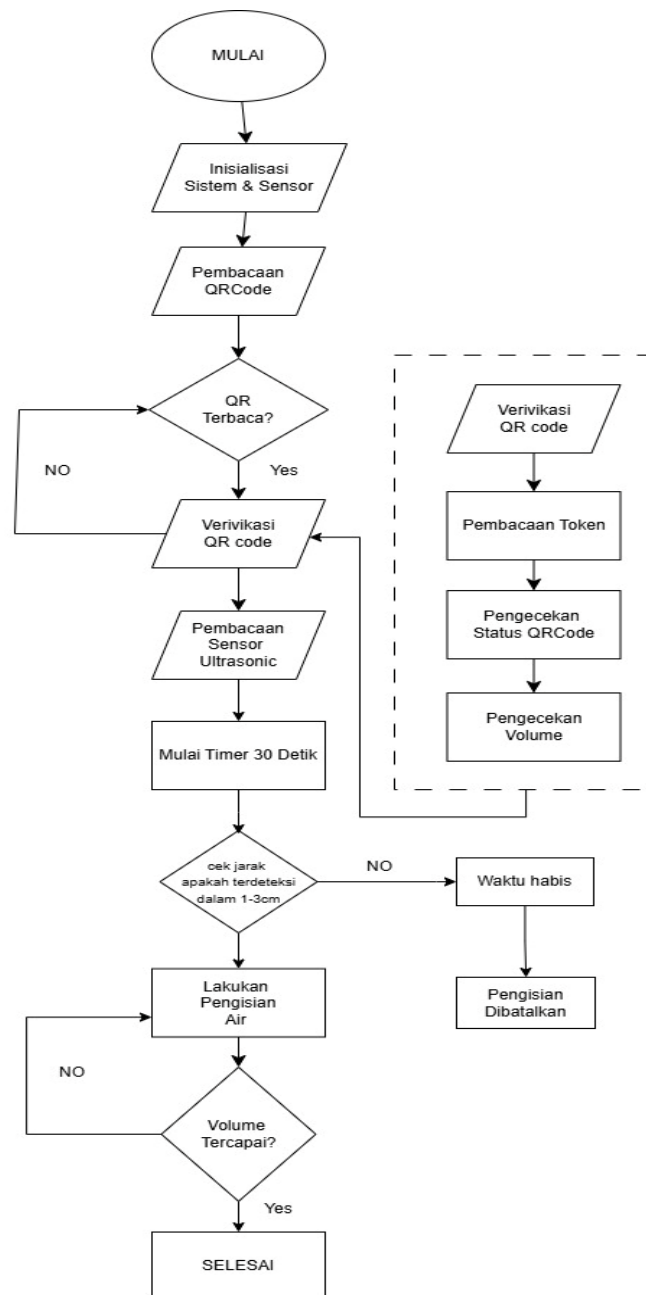
Gambar 3. 16 Tampilan Web Pemilihan Ukuran Galon

Desain rencana tampilan website pada gambar menunjukkan antarmuka halaman pemilihan pengisian galon yang dirancang sederhana dan fokus pada kemudahan interaksi pengguna. Pada bagian sisi kiri terdapat menu navigasi vertikal bertuliskan Beli Galon yang berfungsi sebagai penanda halaman aktif sekaligus akses utama ke fitur pemesanan. Penempatan menu di sisi kiri bertujuan menjaga konsistensi navigasi dan memudahkan pengguna berpindah antar halaman jika sistem dikembangkan lebih lanjut.

Pada bagian utama halaman ditampilkan tiga pilihan produk galon, yaitu Galon 2 liter, 5 liter, dan 15 liter. Masing-masing pilihan disajikan dalam bentuk panel terpisah untuk memperjelas perbedaan kapasitas. Di bawah setiap panel galon disediakan tombol Beli yang berfungsi sebagai pemicu proses transaksi. Ketika tombol ini ditekan, sistem akan mencatat pilihan pengguna dan mengirimkan data ke sistem backend sebagai dasar proses identifikasi dan pengisian galon otomatis.

Secara keseluruhan, desain tampilan website ini mengutamakan kejelasan informasi, kesederhanaan layout, dan efisiensi proses pemilihan galon. Struktur antarmuka yang ringkas meminimalkan potensi kesalahan pengguna saat memilih ukuran galon, sekaligus mendukung integrasi dengan sistem IoT pada alat pengisian galon otomatis. Dengan tampilan seperti ini, website tidak hanya berfungsi sebagai media transaksi, tetapi juga sebagai bagian dari sistem kontrol dan monitoring yang terhubung langsung dengan perangkat pengisian galon.

### 3.10 Flowchart



Gambar 3. 17 Flowchart

Flowchart sistem pengisian galon otomatis diawali dengan proses mulai, di mana sistem diaktifkan dan mikrokontroler ESP32 melakukan inisialisasi sistem dan sensor. Pada tahap ini, seluruh komponen seperti QR Code scanner GM67, sensor ultrasonik HC-SR04, sensor flowmeter, modul relay, serta modul komunikasi untuk web monitoring disiapkan agar siap digunakan. Inisialisasi ini

bertujuan memastikan bahwa seluruh perangkat berada dalam kondisi normal sebelum sistem mulai beroperasi.

Setelah proses inisialisasi selesai, sistem masuk ke tahap pembacaan QR Code. Pada tahap ini, QR Code scanner akan melakukan pemindaian terhadap QR Code yang diberikan oleh pengguna. Sistem kemudian melakukan pengecekan apakah QR Code berhasil terbaca atau tidak. Apabila QR Code tidak terbaca, sistem akan kembali ke proses pembacaan QR Code dan menunggu hingga QR Code berhasil dipindai dengan benar.

Apabila QR Code berhasil terbaca, sistem melanjutkan ke tahap verifikasi QR Code. Proses verifikasi ini mencakup pembacaan token yang terkandung dalam QR Code, pengecekan status QR Code, serta pengecekan parameter yang berkaitan dengan volume pengisian air. Jika hasil verifikasi menunjukkan bahwa QR Code tidak valid atau tidak memenuhi syarat, sistem akan menolak proses pengisian dan kembali ke kondisi awal untuk menunggu pemindaian QR Code berikutnya.

Setelah QR Code dinyatakan valid, sistem melanjutkan ke tahap pembacaan sensor ultrasonik. Sensor ultrasonik HC-SR04 digunakan untuk mendeteksi keberadaan galon pada area pengisian. Sistem kemudian mengevaluasi apakah galon terdeteksi berada pada posisi yang benar. Apabila galon tidak terdeteksi dalam jarak 1-3cm, sistem akan kembali melakukan pembacaan sensor ultrasonik hingga galon terdeteksi dengan benar, namun apabila galon tetap tidak terdeteksi dalam jarak 1-3cm maka sistem akan menyatakan kehabisan waktu atau *TIMEOUT*, dan proses pengisian galon dibatalkan.

Jika galon sudah posisi yang tepat, sistem melanjutkan ke tahap proses pengisian air. Pada tahap ini, ESP32 mengaktifkan modul relay sehingga pompa air DC menyala dan air mulai mengalir ke dalam galon. Selama proses pengisian berlangsung, sensor flowmeter secara terus-menerus mengukur debit dan volume air yang masuk ke dalam galon. Sistem kemudian melakukan pengecekan terhadap volume air yang telah terisi. Apabila volume air belum mencapai nilai yang telah ditentukan berdasarkan hasil verifikasi QR Code, maka proses pengisian akan terus berlangsung. Sebaliknya, apabila volume air telah tercapai, ESP32 akan mematikan modul relay sehingga pompa air berhenti bekerja.

**BAB IV**  
**PEMBUATAN ALAT**

**4.1 Pembuatan Perangkat Keras**

**4.1.1 Alat dan Bahan**

Tabel 4. 1 Daftar Alat Yang Digunakan

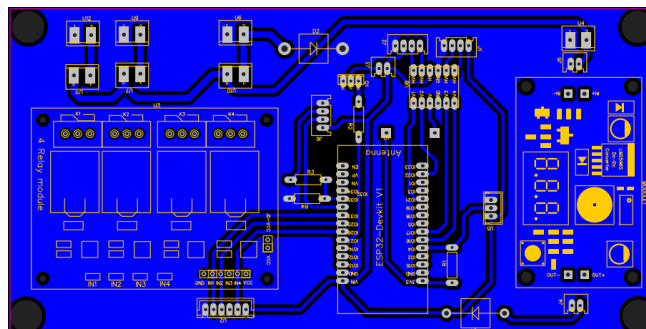
No	Nama Alat	Jumlah
1	Set Obeng	1 buah
2	Multimeter	1 buah
3	Tang Kupas	1 buah
4	Gerinda	1 buah
5	Bor	1
6	Alat Tulis	1
7	Meteran	1
8	Solder	1
9	Lem Tembak	Secukupnya
10	Lem Pipa	Secukupnya
11	Sealant	Secukupnya

Tabel 4. 2 Daftar Bahan Yang Digunakan

No.	Nama Bahan	Jumlah
1	ESP32 DEVKIT V1	1 buah
2	Sensor QR GM-67	1 buah
3	Sensor Ultrasonic HcSr-04	1 buah
4	Sensor Flowmeter Yf-S01	1 buah
5	Pompa Air DC 12V	1 buah
6	LCD I2C 20x4	1 buah
7	Power Supply Switching 10A	1 buah
8	Buck Converter LM2596	1 buah
9	PVC	Secukupnya

No.	Nama Bahan	Jumlah
10	Dioda Flyback	1 buah
11	Logic Level Shifter	1 buah
12	Buzzer	1 buah
13	Modul Relay	1 buah
14	Saklar	1 buah
15	Akrilik	Secukupnya
16	Kabel	Secukupnya
17	Pilot Lamp	2 buah
18	Spacer	Secukupnya

#### 4.1.2 Pembuatan Papan PCB



Gambar 4. 1Desain PCB

Pembuatan papan PCB pada sistem pengisian galon otomatis ini dilakukan untuk menghasilkan rangkaian yang lebih rapi, terstruktur, dan memiliki tingkat keandalan yang lebih tinggi dibandingkan dengan perakitan menggunakan kabel jumper. Pada tahap awal, dilakukan perancangan layout PCB menggunakan perangkat lunak desain rangkaian elektronik. Desain ini disesuaikan dengan wiring diagram yang telah dibuat sebelumnya, dengan mempertimbangkan posisi modul ESP32 sebagai pusat kendali, modul relay sebagai pengendali beban, serta modul buck converter sebagai pengatur tegangan sistem.

Pada Gambar 4.1 ditampilkan desain layout PCB yang telah selesai dirancang. Terlihat bahwa modul ESP32 ditempatkan di bagian tengah papan untuk memudahkan distribusi jalur sinyal ke seluruh komponen. Di sisi kiri papan terdapat area untuk modul relay 4 channel yang digunakan untuk mengendalikan

pompa dan indikator. Jalur suplai daya 12V menuju relay dirancang lebih tebal dibandingkan jalur sinyal sensor untuk mengakomodasi arus yang lebih besar serta mencegah terjadinya penurunan tegangan dan panas berlebih. Di sisi kanan papan terlihat tata letak modul buck converter yang berfungsi menurunkan tegangan 12V menjadi tegangan kerja yang sesuai bagi ESP32 dan komponen logika lainnya. Selain itu, pada bagian atas papan tersedia konektor untuk sensor flowmeter, sensor ultrasonik, serta QR Code scanner.



Gambar 4. 2 Pencetakan PCB

Setelah desain layout selesai, proses dilanjutkan dengan pencetakan pola jalur pada papan PCB berbahan dasar tembaga. Gambar 4.2 menunjukkan hasil pencetakan jalur PCB pada papan tembaga sebelum proses pengeboran dan pemasangan komponen dilakukan. Proses pencetakan ini bertujuan membentuk jalur konduktor sesuai dengan desain yang telah dibuat, sehingga setiap komponen dapat terhubung secara permanen dan stabil. Setelah jalur terbentuk, dilakukan pengeboran pada titik kaki komponen sesuai dengan layout yang dirancang, kemudian dilanjutkan dengan proses penyolderan komponen satu per satu.

Tahap akhir dari pembuatan PCB adalah pengujian jalur menggunakan multimeter untuk memastikan tidak terjadi hubungan singkat antar jalur serta memastikan kontinuitas koneksi sesuai desain. Pengujian juga dilakukan pada jalur suplai daya untuk memastikan tegangan yang diterima oleh ESP32 dan modul lainnya sesuai dengan spesifikasi kerja masing-masing komponen.

### 4.1.3 Pembuatan Perangkat Mekanik

Pembuatan perangkat mekanik merupakan tahapan penting dalam realisasi sistem pengisian galon otomatis, karena bagian ini berfungsi sebagai struktur fisik yang menopang seluruh komponen elektronik, sistem perpipaan, serta galon selama proses pengisian berlangsung. Perangkat mekanik tidak hanya berperan sebagai pelindung komponen internal, tetapi juga sebagai elemen pendukung stabilitas, keamanan operasional, dan kenyamanan pengguna. Oleh karena itu, perancangan dan pembuatan perangkat mekanik dilakukan dengan mempertimbangkan aspek kekuatan material, distribusi beban, perlindungan terhadap kelembaban, serta kemudahan perawatan dan pengembangan di masa mendatang.



Gambar 4. 3 Pembuatan Kotak Pengisian Galon

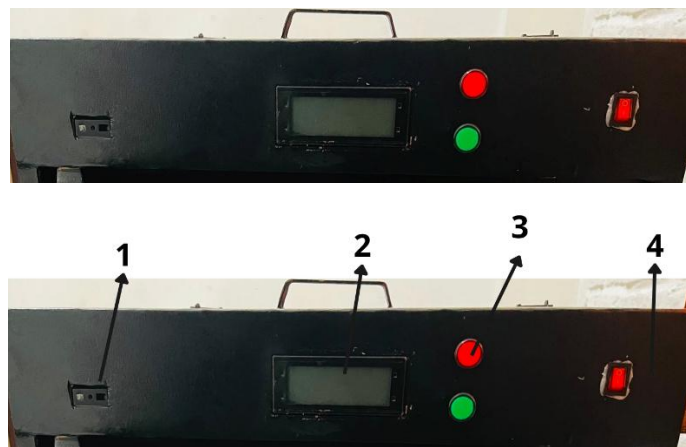
Proses pembuatan diawali dengan perakitan rangka utama kotak pengisian galon seperti yang ditunjukkan pada Gambar 4.3. Pada tahap ini dilakukan pemotongan dan penyambungan plat besi serta rangka besi hollow sesuai dengan dimensi yang telah dirancang sebelumnya. Struktur dibuat dalam bentuk vertikal dengan ruang bagian bawah sebagai area penempatan galon dan ruang bagian atas sebagai panel kontrol sistem. Pemilihan material besi bertujuan untuk memastikan kekuatan struktur dalam menopang beban galon yang terisi penuh air, yang dapat mencapai lebih dari 15 kilogram. Sambungan rangka dilakukan menggunakan teknik pengelasan untuk menghasilkan konstruksi yang kokoh dan tidak mudah

mengalami deformasi akibat beban statis maupun getaran pompa saat sistem bekerja.



Gambar 4. 4 Pemberian Cat Pada Kotak Pengisian Galon

Setelah rangka utama selesai dirakit, dilakukan proses finishing berupa pengamplasan permukaan dan pengecatan seperti yang terlihat pada Gambar 4.4. Proses pengecatan memiliki dua fungsi utama, yaitu memberikan perlindungan terhadap korosi akibat paparan udara lembab dan percikan air, serta meningkatkan nilai estetika alat agar tampak lebih profesional dan layak digunakan di lingkungan operasional depot air minum. Cat yang digunakan dipilih yang memiliki daya rekat dan ketahanan terhadap goresan, sehingga permukaan kotak tetap terjaga dalam jangka waktu penggunaan yang panjang.



Gambar 4. 5 Pemasangan Komponen Sensor QR Code GM-67, LCD, Pilot Lamp,, dan Saklar

Tahap berikutnya adalah pemasangan komponen pada panel kontrol bagian atas sebagaimana ditunjukkan pada Gambar 4.5. Panel ini dirancang sebagai antarmuka utama antara pengguna dan sistem. Komponen yang dipasang meliputi modul QR Code Scanner GM-67 (1), LCD(2) sebagai media tampilan informasi proses pengisian, pilot lamp(3) sebagai indikator status sistem (standby atau aktif), serta saklar utama(4) sebagai pengendali daya sistem. Penempatan komponen dilakukan secara ergonomis agar mudah dijangkau dan dibaca oleh pengguna. Selain itu, lubang pemasangan dibuat presisi agar komponen terpasang kuat dan tidak longgar saat alat dioperasikan.



Gambar 4. 6 Perakitan Papan PCB

Proses selanjutnya adalah perakitan dan integrasi papan PCB ke dalam sistem seperti yang terlihat pada Gambar 4.6. Pada tahap ini, PCB yang telah melalui proses pencetakan dan penyolderan dipasang padaudukan di dalam enclosure. Pengkabelan antar modul dilakukan dengan memperhatikan jalur distribusi daya dan jalur sinyal untuk menghindari interferensi listrik. Kabel daya dipisahkan dari kabel sinyal sensor guna menjaga kestabilan pembacaan data oleh mikrokontroler. Selain itu, setiap koneksi diperiksa kembali untuk memastikan tidak terjadi kesalahan polaritas atau sambungan yang kurang kuat.



Gambar 4. 7 Pemasangan PCB dan PSU Pada Akrilik

Pada Gambar 4.7 ditunjukkan pemasangan PCB dan Power Supply Unit (PSU) pada papan akrilik di bagian dalam kotak. Penggunaan papan akrilik sebagai dudukan bertujuan untuk memberikan isolasi listrik antara rangkaian dan badan logam kotak, sehingga mengurangi risiko hubungan singkat. PSU ditempatkan pada posisi yang memiliki ventilasi cukup untuk menjaga sirkulasi udara dan menghindari penumpukan panas. Tata letak komponen di dalam kotak dirancang agar memudahkan akses saat dilakukan perawatan atau penggantian komponen.

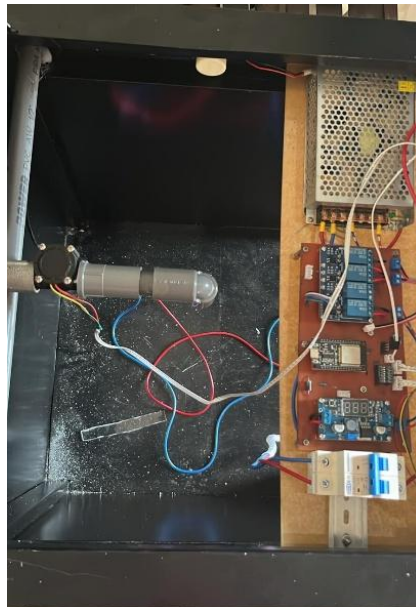


Gambar 4. 8 Perakitan Pompa dan Sensor Flowmeter

Tahap selanjutnya dalam pembuatan perangkat mekanik adalah proses perakitan pompa dan sensor aliran air seperti yang ditunjukkan pada Gambar 4.8. Pada tahap ini dilakukan penyambungan antara pompa air DC dan Sensor Flowmeter (flowmeter) menggunakan pipa PVC dan konektor yang sesuai dengan diameter jalur air. Perakitan dilakukan dengan memastikan arah aliran air sesuai dengan tanda panah yang terdapat pada body Sensor Flowmeter, sehingga pembacaan pulsa dapat dilakukan secara akurat. Sambungan pipa dibuat rapat dan diberi perekat atau sealant untuk mencegah kebocoran air saat sistem beroperasi. Posisi pompa dirancang sejajar dengan Sensor Flowmeter agar aliran air tetap stabil

dan tidak menimbulkan turbulensi berlebihan yang dapat memengaruhi akurasi pengukuran volume.

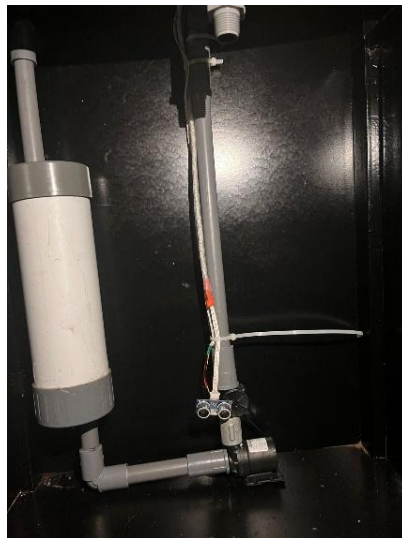
Sensor Flowmeter dipasang setelah pompa pada jalur keluaran air, sehingga seluruh air yang dipompa menuju galon akan melewati sensor terlebih dahulu. Konfigurasi ini bertujuan agar perhitungan volume air benar-benar merepresentasikan jumlah air yang masuk ke galon. Kabel keluaran sensor kemudian dihubungkan ke sistem kontrol melalui PCB untuk dibaca sebagai sinyal pulsa oleh ESP32. Proses perakitan ini dilakukan dengan memperhatikan kekuatan mekanis sambungan serta keamanan jalur kabel agar tidak terkena percikan air.



Gambar 4. 9 Perakitan Komponen Kedalam Kotak Pengisian Galon

Gambar 4.9 menampilkan proses perakitan komponen ke dalam kotak pengisian galon. Pada gambar tersebut terlihat sebuah papan akrilik yang telah dipasang seluruh komponen elektronik, meliputi mikrokontroler ESP32, modul relay, PSU, dan komponen pendukung lainnya, kemudian dimasukkan dan dipasang ke dalam kotak. Penggunaan papan akrilik sebagai media pemasangan komponen bertujuan untuk memudahkan proses instalasi sekaligus menjaga kerapian tata letak komponen agar tidak bergeser saat sistem beroperasi.

Di sisi seberang papan akrilik tersebut, terlihat jalur nozzle yang berfungsi sebagai saluran keluarnya air untuk proses pengisian galon. Jalur nozzle ini ditempatkan pada posisi yang telah diperhitungkan agar tepat berada di atas mulut galon ketika galon diletakkan pada posisi pengisian. Penempatan yang presisi ini menjadi faktor penting untuk memastikan air mengalir langsung ke dalam galon tanpa tumpah ke luar.



Gambar 4. 10 Pemasangan Sensor Ultrasonik pada Kotak Pengisian Galon

Instalasi sistem perpipaan dan sensor juga menjadi bagian penting dalam tahap pembuatan perangkat mekanik. Gambar 4.10 menunjukkan pemasangan sensor ultrasonik pada bagian dalam kotak pengisian galon. Sensor ini diposisikan sedemikian rupa agar mampu mendeteksi keberadaan galon dengan jarak yang sesuai dengan parameter dalam program. Penempatan sensor mempertimbangkan sudut deteksi dan jarak efektif agar pembacaan tidak terganggu oleh pantulan dari dinding kotak. Jalur pipa air dipasang secara vertikal dan terhubung langsung ke pompa serta nozzle pengisian untuk memastikan aliran air masuk secara tepat ke dalam galon. Penetapan jarak deteksi optimal sebesar 1 hingga 3 sentimeter pada sensor ultrasonik HC-SR04 didasarkan pada empat pertimbangan teknis yang saling berkaitan. Pertimbangan konstruksi mekanik alat. Posisi sensor ultrasonik dipasang tepat di atas lubang pengisian (nozzle). Galon yang diletakkan dengan benar akan memiliki mulut galon berada tepat di bawah nozzle. Jarak antara permukaan sensor dan mulut galon dalam kondisi ini terukur pada rentang 1 hingga 3 sentimeter. Jarak

ini cukup dekat untuk memastikan galon benar-benar terpasang, namun tidak terlalu rapat yang dapat menyebabkan gangguan pantulan gelombang ultrasonik.

Karakteristik teknis HC-SR04. Sensor HC-SR04 memiliki jarak minimum deteksi efektif sekitar 2 sentimeter. Di bawah jarak tersebut, gelombang ultrasonik yang dipancarkan dan dipantulkan dapat saling berinterferensi sehingga pembacaan menjadi tidak akurat. Oleh karena itu, batas bawah ditetapkan 1 sentimeter sebagai toleransi aman yang memperhitungkan kemungkinan variasi posisi pemasangan galon di lapangan. Jarak tersebut juga memperhatikan kebutuhan stabilitas sistem. Sistem dirancang agar galon harus terdeteksi pada jarak 1–3 cm secara stabil selama 2 detik berturut-turut sebelum pompa diaktifkan. Hal ini mencegah pompa aktif akibat benda lain yang secara tidak sengaja melintas di depan sensor. Jika jarak terdeteksi melebihi 3 sentimeter, sistem menganggap galon belum terpasang dengan benar dan tidak akan memulai proses pengisian.



Gambar 4. 11 Kotak Pengisian Dengan Galon

Hal ini juga menyesuaikan terhadap dimensi fisik platform dan galon referensi. Alas kotak alat dirancang dengan dimensi 50×50 cm, dan galon referensi yang digunakan sebagai patokan perancangan adalah galon 15 liter merek *Le Mineral*. Galon *Le Mineral* 15 liter memiliki diameter sekitar 27 hingga 30 cm dengan tinggi sekitar 30 hingga 50 cm. Dengan diameter tersebut, ketika galon diletakkan di tengah alas 50×50 cm, masih terdapat clearance sekitar 10 hingga 11,5 cm di setiap

sisi galon cukup untuk memastikan galon berdiri stabil tanpa miring, sekaligus tidak terlalu longgar sehingga posisi mulut galon tetap terjaga sejajar secara vertikal dengan nozzle. Posisi nozzle itu sendiri tidak dapat ditempatkan secara bebas, melainkan dibatasi oleh ketersediaan ruang di dalam kotak, khususnya area yang ditempati oleh komponen kontrol seperti modul ESP32, relay, dan rangkaian pendukung lainnya. Penempatan komponen-komponen tersebut menentukan seberapa tinggi dan di posisi mana nozzle dapat dipasang secara optimal tanpa mengganggu jalur kabel maupun akses perawatan.

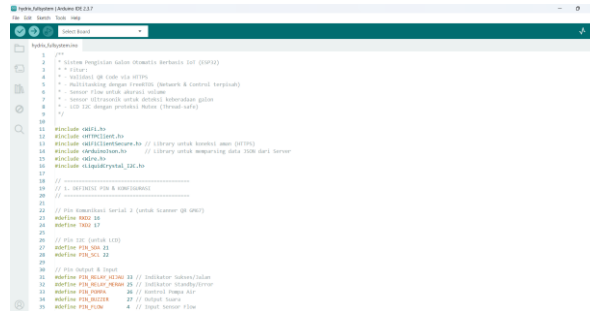


Gambar 4. 12 Tampak Keseluruhan Komponen pada Kotak Pengisian Galon

Tahap akhir perakitan perangkat mekanik dapat dilihat pada Gambar 4.11, yang menunjukkan tampak keseluruhan komponen di dalam kotak pengisian galon. Pada tahap ini seluruh sistem telah terintegrasi, mulai dari pompa air, modul relay, ESP32, PSU, sensor, hingga jalur perpipaan. Penataan kabel dilakukan menggunakan pengikat kabel untuk menjaga kerapian dan mencegah kabel bersentuhan dengan bagian bergerak atau jalur air. Penempatan komponen dilakukan dengan mempertimbangkan distribusi beban dan aliran panas agar sistem tetap stabil selama pengoperasian jangka panjang.

## 4.2 Pembuatan Perangkat Lunak

### 4.2.1 Pembuatan Program Pada Arduino IDE



Gambar 4. 13 Program pada Arduino IDE

Pembuatan program pada sistem pengisian galon otomatis ini dilakukan menggunakan perangkat lunak Arduino IDE sebagai lingkungan pengembangan untuk mikrokontroler ESP32. Arduino IDE dipilih karena memiliki kompatibilitas penuh dengan board ESP32, mendukung instalasi library tambahan melalui Library Manager, serta menyediakan fasilitas kompilasi dan upload program secara langsung melalui koneksi USB. Melalui platform ini, seluruh proses mulai dari pembacaan sensor, pengendalian aktuator, komunikasi dengan server, hingga manajemen multitasking dapat diintegrasikan dalam satu sistem program yang terstruktur.

Tahap awal pengembangan dilakukan dengan menginstal dukungan board ESP32 melalui fitur Board Manager pada Arduino IDE menggunakan URL repositori resmi Espressif. Setelah board terinstal, dilakukan instalasi library pendukung yang dibutuhkan oleh sistem, yaitu WiFi.h untuk manajemen koneksi jaringan nirkabel, HTTPClient.h untuk komunikasi HTTP dengan server API, WiFiClient.h sebagai lapisan koneksi jaringan dasar, ArduinoJson.h untuk parsing dan pengolahan data berformat JSON yang diterima dari server, Wire.h untuk protokol komunikasi I2C dengan modul LCD, LiquidCrystal\_I2C.h untuk pengendalian tampilan LCD 20x4, Preferences.h untuk penyimpanan data WiFi secara permanen pada memori flash internal ESP32, serta WebServer.h untuk menjalankan web server lokal pada ESP32 dalam mode konfigurasi WiFi.

Keseluruhan library tersebut menjadi fondasi integrasi antara perangkat keras dan sistem berbasis internet pada penelitian ini.

Program ESP32 dirancang secara modular dengan pembagian fungsi yang jelas agar sistem mampu menjalankan beberapa proses secara bersamaan menggunakan mekanisme multitasking FreeRTOS. Secara keseluruhan, program terdiri dari tujuh bagian utama yang masing-masing memiliki peran spesifik dalam mendukung operasional sistem. Berikut adalah penjelasan setiap bagian program secara terstruktur.

### **1. Konfigurasi Awal dan Definisi Pin**

Pada bagian paling awal program dilakukan pemanggilan seluruh library yang dibutuhkan, diikuti dengan pendefinisian seluruh pin input dan output yang digunakan oleh sistem. Pendefinisian pin menggunakan direktif `#define` agar kode lebih mudah dibaca dan perubahan nomor pin cukup dilakukan pada satu lokasi tanpa harus menelusuri seluruh baris program. Pin yang didefinisikan mencakup pin komunikasi serial UART2 untuk QR Code Scanner GM-67 (RXD2 = GPIO16, TXD2 = GPIO17), pin komunikasi I2C untuk LCD (SDA = GPIO21, SCL = GPIO22), pin kendali relay lampu indikator hijau (GPIO33) dan merah (GPIO25), pin kendali relay pompa air (GPIO26), pin buzzer (GPIO27), pin interrupt sensor flowmeter (GPIO4), serta pin trigger (GPIO32) dan echo (GPIO35) sensor ultrasonik.

Selain pendefinisian pin, bagian ini juga mendeklarasikan variabel dan parameter operasional sistem, yaitu `TARGET_JARAK` sebagai jarak ideal deteksi galon dalam satuan sentimeter, `TOLERANSI` sebagai batas simpangan jarak yang masih diterima, `WAKTU_STABIL` sebagai durasi minimum galon harus berada pada posisi yang benar sebelum pompa diaktifkan, serta `calibrationFactor` sebagai faktor kalibrasi sensor flowmeter dengan nilai optimal  $K = 7,4$  yang diperoleh dari hasil pengujian komparatif.

### **2. Fungsi Bantuan Sistem**

Program dilengkapi dengan tiga fungsi bantuan yang digunakan secara berulang oleh kedua task utama. Fungsi `lcdPrintRow()` bertugas menampilkan teks pada baris tertentu layar LCD dengan cara yang aman untuk lingkungan

multitasking. Fungsi ini secara otomatis menambahkan spasi di belakang teks yang panjangnya kurang dari 20 karakter agar karakter sisa dari tampilan sebelumnya tertimpa sepenuhnya, serta memotong teks yang melebihi 20 karakter. Akses ke LCD diproteksi menggunakan mutex `xSemaphoreTake()` untuk mencegah dua task menulis ke LCD pada waktu yang bersamaan yang dapat menyebabkan tampilan kacau.

Fungsi `beep()` bertugas menghasilkan suara notifikasi dari buzzer dengan jumlah dan durasi bunyi yang dapat dikonfigurasi melalui parameternya. Fungsi `getDistance()` bertugas mengukur jarak menggunakan sensor ultrasonik HC-SR04 dengan mengirimkan pulsa trigger selama 10 mikro detik kemudian mengukur durasi sinyal echo menggunakan fungsi `pulseIn()`. Jarak dalam satuan sentimeter dihitung menggunakan persamaan berikut.

$$\text{Jarak (cm)} = \text{durasi} \times 0,034 / 2$$

Apabila tidak ada objek yang terdeteksi dalam batas waktu 30 detik, fungsi mengembalikan nilai 999 sebagai penanda kondisi tidak ada objek.

### **3. Komunikasi HTTP dengan Server API**

Sistem berkomunikasi dengan server menggunakan protokol HTTP melalui objek `WiFiClient` dan `HTTPClient`. Terdapat dua fungsi utama dalam modul komunikasi ini.

Fungsi `checkDatabase()` berfungsi untuk memvalidasi token QR Code yang dipindai. Fungsi ini mengirimkan permintaan HTTP GET ke endpoint `api.php` dengan menyertakan token sebagai parameter URL. Respons dari server berupa data berformat JSON kemudian diurai menggunakan library `ArduinoJson`. Apabila server mengembalikan status "valid", fungsi ini mengembalikan nilai volume pengisian dalam satuan mililiter. Apabila token tidak valid atau tidak ditemukan, fungsi mengembalikan nilai 0 sebagai penanda kegagalan validasi.

Fungsi `updateTokenStatus()` berfungsi untuk memperbarui status token di database setelah proses pengisian selesai. Fungsi ini mengirimkan permintaan HTTP GET dengan parameter tambahan `action=finish` ke endpoint `api.php`, yang selanjutnya akan menjalankan perintah SQL UPDATE untuk mengubah status

token dari "active" menjadi "used" di database server. Mekanisme ini memastikan setiap token hanya dapat digunakan untuk satu kali proses pengisian.

#### **4. Pengaturan WiFi**

Sistem memungkinkan konfigurasi jaringan WiFi dilakukan tanpa perlu mengubah kode program dan mengupload ulang. Fitur ini bekerja dalam dua mode secara adaptif. Pada mode normal, ESP32 membaca data SSID dan password yang tersimpan di memori flash menggunakan library Preferences, kemudian mencoba terhubung ke jaringan WiFi tersebut. Apabila koneksi berhasil, sistem langsung berjalan dalam mode operasional.

Apabila koneksi gagal atau belum ada data WiFi yang tersimpan, sistem secara otomatis beralih ke mode konfigurasi. Pada mode ini, ESP32 mengaktifkan Access Point dengan nama WiFi\_SETUP dan menjalankan web server lokal menggunakan library WebServer pada port 80. Pengguna dapat menghubungkan perangkat smartphone atau laptop ke hotspot tersebut dan membuka halaman konfigurasi melalui browser pada alamat IP 192.168.4.1. Halaman tersebut menyediakan formulir untuk memasukkan SSID dan password WiFi yang baru. Setelah formulir dikirimkan, data disimpan ke memori flash menggunakan Preferences dan ESP32 melakukan restart otomatis untuk menerapkan konfigurasi yang baru.

##### **4.2.2 Pembuatan Website**

Pembuatan website pada sistem pengisian galon otomatis ini merupakan bagian penting dalam integrasi antara perangkat keras dan sistem berbasis Internet of Things (IoT). Website dikembangkan sebagai media transaksi digital yang berfungsi untuk menghasilkan token pembelian, melakukan validasi data secara daring, serta menyimpan dan mengelola seluruh riwayat transaksi dalam basis data. Keberadaan website memungkinkan sistem tidak hanya bekerja secara lokal, tetapi terhubung langsung dengan jaringan internet sehingga perangkat ESP32 dapat berkomunikasi dengan server secara real-time melalui protokol HTTPS.

Dalam proses pengembangannya, website dirancang menggunakan bahasa pemrograman PHP sebagai backend, HTML dan CSS sebagai struktur dan tampilan antarmuka, serta JavaScript untuk mendukung interaksi dinamis pada sisi

pengguna. Sistem ini menerapkan arsitektur client-server, di mana website bertindak sebagai server aplikasi dan perangkat ESP32 berperan sebagai client IoT yang mengakses layanan melalui Application Programming Interface (API). Basis data MySQL digunakan sebagai media penyimpanan utama untuk mencatat token, volume air, status transaksi, dan waktu pembelian.

Agar sistem lebih terstruktur dan mudah dalam pemeliharaan, pembuatan website dibagi ke dalam beberapa file utama yang masing-masing memiliki fungsi dan peran tersendiri namun tetap saling terintegrasi. Pembagian ini bertujuan untuk meningkatkan keteraturan kode program, mempermudah proses pengembangan lanjutan, serta menjaga keamanan dan efisiensi sistem secara keseluruhan.

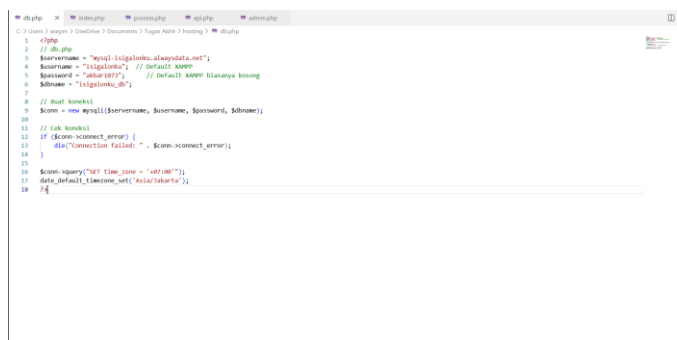
#### **4.2.2.1 Program Pembentuk Website**

Sistem website pada alat pengisian galon otomatis dibangun menggunakan empat file utama yang saling terintegrasi dan membentuk arsitektur aplikasi berbasis web dan Internet of Things (IoT). Website berperan sebagai server aplikasi yang menangani proses transaksi, pembentukan token, penyimpanan data, serta monitoring penggunaan sistem. Sementara itu, perangkat ESP32 berfungsi sebagai client IoT yang berkomunikasi dengan server untuk melakukan validasi token dan mengirimkan data hasil pengisian.

Keempat file tersebut terdiri dari file `index.php` sebagai antarmuka pengguna untuk pemilihan volume pengisian, `process.php` sebagai pemroses transaksi dan pembentuk token, `db.php` sebagai modul koneksi database, serta `admin.php` sebagai dashboard administrator untuk monitoring dan pengelolaan data transaksi. Seluruh file bekerja secara terintegrasi untuk mendukung proses pembelian token, pembentukan QR Code, validasi transaksi, serta pencatatan dan pelaporan data penggunaan sistem.

Berikut merupakan penjelasan dari masing-masing file pembentuk website yang digunakan pada sistem.

## 1. File db.php (Modul Koneksi Database)



```

1 // db.php
2 // db.php
3 $servername = "mysql-1sigaloka.alaydata.net";
4 $username = "1sigaloka" // default: xxxxx
5 $password = "1sigaloka"; // default: xxxxx klasanya kosong
6 $dbname = "1sigaloka_db";
7
8 // Buat koneksi
9 $conn = new mysqli($servername, $username, $password, $dbname);
10
11 // Cek koneksi
12 if ($conn->connect_error) {
13     die("Connection failed: " . $conn->connect_error);
14 }
15
16 $conn->query("set time_zone = '+07:00'");
17 $conn->query("set default_timezone = 'Asia/Jakarta'");
18 }

```

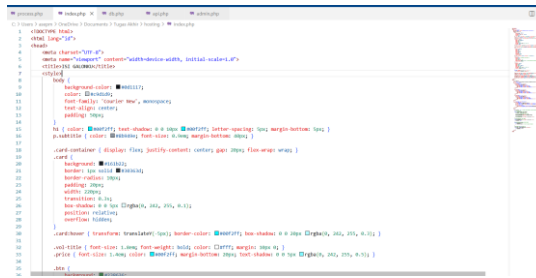
Gambar 4. 14 Program db.php

File db.php berfungsi sebagai modul koneksi database yang menjadi penghubung antara aplikasi berbasis PHP dengan sistem manajemen basis data MySQL. File ini menerapkan konsep koneksi terpusat sehingga konfigurasi koneksi hanya dituliskan satu kali dan dapat digunakan kembali oleh file-file lain yang memerlukan akses ke database, seperti process.php, admin.php, maupun modul validasi token. Pendekatan ini mempermudah pemeliharaan sistem dan mengurangi duplikasi kode program.

Di dalam file db.php tersimpan konfigurasi koneksi yang meliputi hostname, username, password, dan nama database. Koneksi database dibangun menggunakan ekstensi mysqli pada PHP sehingga memungkinkan proses penyimpanan, pengambilan, pembaruan, dan penghapusan data transaksi dilakukan secara terintegrasi. Sistem juga menerapkan mekanisme penanganan kesalahan koneksi dengan menghentikan proses apabila koneksi ke database tidak berhasil dilakukan, sehingga dapat mencegah terjadinya kesalahan berantai pada proses berikutnya.

Selain berfungsi sebagai modul koneksi database, file ini juga mengatur zona waktu sistem menggunakan fungsi `date_default_timezone_set('Asia/Jakarta')` serta pengaturan time zone pada server database. Pengaturan tersebut bertujuan untuk memastikan seluruh data transaksi, waktu pembelian token, dan riwayat penggunaan sistem tercatat sesuai dengan zona waktu Indonesia Barat (WIB). Dengan demikian, konsistensi data waktu antara aplikasi, database, dan perangkat IoT dapat terjaga dengan baik.

## 2. File index.php (Tampilan Website)



Gambar 4. 15 Program Index.php

File index.php merupakan halaman utama yang digunakan oleh pengguna untuk melakukan pembelian token pengisian air pada sistem pengisian galon otomatis. Halaman ini berfungsi sebagai antarmuka awal yang menyediakan pilihan volume pengisian, yaitu 2 liter, 5 liter, dan 15 liter. Setiap pilihan volume ditampilkan dalam bentuk kartu (card) sehingga memudahkan pengguna dalam melakukan transaksi. Tampilan halaman dikembangkan menggunakan HTML, CSS, dan JavaScript. Tata letak halaman memanfaatkan metode Flexbox untuk mengatur posisi elemen secara responsif sehingga tampilan tetap proporsional pada berbagai ukuran layar, baik perangkat desktop maupun perangkat bergerak. Selain itu, penggunaan CSS memberikan tampilan antarmuka yang lebih menarik dan mudah digunakan oleh pengguna. Untuk mengurangi kemungkinan kesalahan transaksi akibat salah memilih volume, sistem menerapkan mekanisme konfirmasi menggunakan modal dialog berbasis JavaScript. Setelah pengguna memilih volume dan menyetujui konfirmasi transaksi, data volume akan dikirimkan ke file process.php menggunakan metode HTTP POST melalui form yang telah disediakan.

File index.php berfungsi sebagai antarmuka utama yang menghubungkan pengguna dengan proses transaksi, mulai dari pemilihan volume hingga pengiriman data ke server untuk proses pembentukan token dan QR Code.

### 3. File api.php (Application Programming Interface / Endpoint IoT)

```

1 <?php
2 header('Content-Type: application/json');
3 include 'db.php';
4
5 if (isset($_GET['token'])) {
6     $token_input = $conn->real_escape_string($_GET['token']);
7     $action = isset($_GET['action']) ? $_GET['action'] : '';
8
9     if ($action == "finish") {
10        $sql = "UPDATE tokens SET status = 'used' WHERE token_code = '$token_input'";
11
12        if ($conn->query($sql) == TRUE) {
13            echo json_encode(["status" => "success"]);
14        } else {
15            echo json_encode(["status" => "error"]);
16        }
17    } else {
18        $sql = "SELECT volume FROM tokens WHERE token_code = '$token_input' AND status = 'active' LIMIT 1";
19        $result = $conn->query($sql);
20
21        if ($result && $result->num_rows > 0) {
22            $row = $result->fetch_assoc();
23            echo json_encode([
24                "status" => "valid",
25                "volume" => ($row['volume'])
26            ]);
27        } else {
28            echo json_encode(["status" => "invalid"]);
29        }
30    } else {
31        echo json_encode(["status" => "error", "message" => "No token provided"]);
32    }
33 }
34
35 $conn->close();

```

Gambar 4. 16 Program api.php

File api.php berfungsi sebagai Application Programming Interface (API) yang menjadi penghubung antara website, database, dan perangkat ESP32 pada sistem pengisian galon otomatis. File ini menerima permintaan (request) dari ESP32, melakukan proses validasi terhadap data yang tersimpan pada database, kemudian mengirimkan respons kembali ke mikrokontroler dalam format JSON. Format JSON dipilih karena memiliki struktur data yang ringan, mudah diproses, dan kompatibel dengan library ArduinoJson yang digunakan pada ESP32.

API menyediakan dua fungsi utama, yaitu validasi token dan penyelesaian transaksi. Pada proses validasi, ESP32 mengirimkan token hasil pembacaan QR Code menggunakan metode HTTP GET. Sistem kemudian melakukan pencarian data pada database dan memeriksa apakah token masih berstatus active. Apabila token ditemukan dan valid, API akan mengirimkan respons JSON yang berisi informasi status validasi serta volume air yang harus diisi oleh sistem.

Setelah proses pengisian selesai dilakukan, ESP32 mengirimkan permintaan kedua dengan parameter action=finish sebagai penanda bahwa transaksi telah selesai. API kemudian menjalankan proses pembaruan data pada database dengan mengubah status token dari active menjadi used. Mekanisme ini diterapkan untuk memastikan bahwa setiap token hanya dapat digunakan satu kali sehingga dapat meningkatkan keamanan sistem dan mencegah penggunaan ulang token yang tidak sah.

Dengan adanya file api.php, proses komunikasi antara website dan perangkat IoT dapat berjalan secara otomatis, terintegrasi, dan real-time sehingga seluruh transaksi pengisian galon dapat divalidasi serta dicatat dengan baik pada database.

#### 4. File admin.php (Dashboard Administrator)

```

1 <?php
2 session_start();
3 include('db.php');
4
5 // --- KEMERIAAN ADMIN ---
6 $admin_username = "akbar091";
7
8 // --- LOGIC LOGIN / LOGOUT ---
9 if (isset($_GET['action']) == "logout") {
10 session_destroy();
11 header("Location: admin.php");
12 exit();
13 }
14
15 if (isset($_POST['method']) == "login" && isset($_POST['password']) {
16 if (isset($_POST['password']) == $admin_password) {
17 } else {
18 $error_msg = "Password salah!";
19 }
20 }
21
22
23 // Jika belum login, tampilkan form login
24 if (isset($_SESSION['is_admin']) == 0) {
25
26 <div style="border: 1px solid #ccc; padding: 10px; margin: 10px auto; width: 300px;">
27 <div style="text-align: center;">
28 <h3 style="margin: 0;">Form Login
29 <div style="display: flex; justify-content: center; gap: 10px; margin-top: 10px;">
30 <input type="text" style="width: 60%; border: 1px solid #ccc; padding: 5px;"/>
31 <input type="password" style="width: 60%; border: 1px solid #ccc; padding: 5px;"/>
32 <input type="submit" value="Login" style="border: 1px solid #ccc; padding: 5px 15px; margin-left: 10px;"/>
33 </div>
34 <div style="margin-top: 10px; text-align: center;">
35 <small style="color: #666;">Jika belum login, tampilkan form login
36 </div>
37 </div>
38
39 </div>
40
41 <div style="display: flex; justify-content: space-between; align-items: center; padding: 10px 0;">
42 <div style="font-size: 0.9em; color: #666;">
43 <span>Dashboard Administrator
44 <span style="font-weight: bold; color: #007bff;">Admin
45 </div>
46 <div style="font-size: 0.8em; color: #666;">
47 <span>Logout
48 </div>
49 </div>
50
51 </div>
52
53 <div style="display: flex; justify-content: space-between; align-items: center; padding: 10px 0;">
54 <div style="font-size: 0.8em; color: #666;">
55 <span>Dashboard Administrator
56 <span style="font-weight: bold; color: #007bff;">Admin
57 </div>
58 <div style="font-size: 0.8em; color: #666;">
59 <span>Logout
60 </div>
61 </div>
62
63 </div>
64
65 </div>
66
67 </div>
68
69 </div>
70
71 </div>
72
73 </div>
74
75 </div>
76
77 </div>
78
79 </div>
80
81 </div>
82
83 </div>
84
85 </div>
86
87 </div>
88
89 </div>
90
91 </div>
92
93 </div>
94
95 </div>
96
97 </div>
98
99 </div>
100
101 </div>
102
103 </div>
104
105 </div>
106
107 </div>
108
109 </div>
110
111 </div>
112
113 </div>
114
115 </div>
116
117 </div>
118
119 </div>
120
121 </div>
122
123 </div>
124
125 </div>
126
127 </div>
128
129 </div>
130
131 </div>
132
133 </div>
134
135 </div>
136
137 </div>
138
139 </div>
140
141 </div>
142
143 </div>
144
145 </div>
146
147 </div>
148
149 </div>
150
151 </div>
152
153 </div>
154
155 </div>
156
157 </div>
158
159 </div>
160
161 </div>
162
163 </div>
164
165 </div>
166
167 </div>
168
169 </div>
170
171 </div>
172
173 </div>
174
175 </div>
176
177 </div>
178
179 </div>
180
181 </div>
182
183 </div>
184
185 </div>
186
187 </div>
188
189 </div>
190
191 </div>
192
193 </div>
194
195 </div>
196
197 </div>
198
199 </div>
200
201 </div>
202
203 </div>
204
205 </div>
206
207 </div>
208
209 </div>
210
211 </div>
212
213 </div>
214
215 </div>
216
217 </div>
218
219 </div>
220
221 </div>
222
223 </div>
224
225 </div>
226
227 </div>
228
229 </div>
230
231 </div>
232
233 </div>
234
235 </div>
236
237 </div>
238
239 </div>
240
241 </div>
242
243 </div>
244
245 </div>
246
247 </div>
248
249 </div>
250
251 </div>
252
253 </div>
254
255 </div>
256
257 </div>
258
259 </div>
260
261 </div>
262
263 </div>
264
265 </div>
266
267 </div>
268
269 </div>
270
271 </div>
272
273 </div>
274
275 </div>
276
277 </div>
278
279 </div>
280
281 </div>
282
283 </div>
284
285 </div>
286
287 </div>
288
289 </div>
290
291 </div>
292
293 </div>
294
295 </div>
296
297 </div>
298
299 </div>
300
301 </div>
302
303 </div>
304
305 </div>
306
307 </div>
308
309 </div>
310
311 </div>
312
313 </div>
314
315 </div>
316
317 </div>
318
319 </div>
320
321 </div>
322
323 </div>
324
325 </div>
326
327 </div>
328
329 </div>
330
331 </div>
332
333 </div>
334
335 </div>
336
337 </div>
338
339 </div>
340
341 </div>
342
343 </div>
344
345 </div>
346
347 </div>
348
349 </div>
350
351 </div>
352
353 </div>
354
355 </div>
356
357 </div>
358
359 </div>
360
361 </div>
362
363 </div>
364
365 </div>
366
367 </div>
368
369 </div>
370
371 </div>
372
373 </div>
374
375 </div>
376
377 </div>
378
379 </div>
380
381 </div>
382
383 </div>
384
385 </div>
386
387 </div>
388
389 </div>
390
391 </div>
392
393 </div>
394
395 </div>
396
397 </div>
398
399 </div>
400
401 </div>
402
403 </div>
404
405 </div>
406
407 </div>
408
409 </div>
410
411 </div>
412
413 </div>
414
415 </div>
416
417 </div>
418
419 </div>
420
421 </div>
422
423 </div>
424
425 </div>
426
427 </div>
428
429 </div>
430
431 </div>
432
433 </div>
434
435 </div>
436
437 </div>
438
439 </div>
440
441 </div>
442
443 </div>
444
445 </div>
446
447 </div>
448
449 </div>
450
451 </div>
452
453 </div>
454
455 </div>
456
457 </div>
458
459 </div>
460
461 </div>
462
463 </div>
464
465 </div>
466
467 </div>
468
469 </div>
470
471 </div>
472
473 </div>
474
475 </div>
476
477 </div>
478
479 </div>
480
481 </div>
482
483 </div>
484
485 </div>
486
487 </div>
488
489 </div>
490
491 </div>
492
493 </div>
494
495 </div>
496
497 </div>
498
499 </div>
500
501 </div>
502
503 </div>
504
505 </div>
506
507 </div>
508
509 </div>
510
511 </div>
512
513 </div>
514
515 </div>
516
517 </div>
518
519 </div>
520
521 </div>
522
523 </div>
524
525 </div>
526
527 </div>
528
529 </div>
530
531 </div>
532
533 </div>
534
535 </div>
536
537 </div>
538
539 </div>
540
541 </div>
542
543 </div>
544
545 </div>
546
547 </div>
548
549 </div>
550
551 </div>
552
553 </div>
554
555 </div>
556
557 </div>
558
559 </div>
560
561 </div>
562
563 </div>
564
565 </div>
566
567 </div>
568
569 </div>
569

```

Gambar 4. 17 Program admin.php

File admin.php merupakan halaman dashboard administrator yang berfungsi sebagai pusat monitoring, pelaporan, dan pengelolaan data transaksi pada sistem pengisian galon otomatis. Sistem keamanan dashboard diterapkan menggunakan manajemen sesi PHP melalui fungsi `session_start()`, sehingga hanya pengguna yang berhasil melakukan autentikasi yang dapat mengakses halaman administrator. Apabila sesi administrator tidak ditemukan, sistem akan mengarahkan pengguna ke halaman login untuk melakukan verifikasi password.

Dashboard administrator menyediakan berbagai informasi penting yang diperoleh dari basis data, meliputi statistik transaksi harian, statistik keseluruhan sistem, jumlah transaksi yang telah digunakan, volume air yang telah dibeli, serta volume air yang telah direalisasikan. Data tersebut ditampilkan secara real-time berdasarkan hasil pengolahan data transaksi yang tersimpan pada database MySQL.

Selain menampilkan data dalam bentuk tabel, dashboard juga dilengkapi dengan visualisasi grafik menggunakan pustaka Chart.js dan ChartDataLabels. Grafik digunakan untuk menampilkan distribusi volume pengisian yang paling banyak dipilih pengguna sehingga administrator dapat melakukan analisis terhadap pola penggunaan sistem.

Fitur manajemen data juga disediakan dalam dashboard, meliputi penghapusan data transaksi secara individual maupun penghapusan beberapa data sekaligus (bulk delete). Seluruh data transaksi ditampilkan dalam bentuk log yang memuat informasi waktu transaksi, kode token, volume pengisian, dan status penggunaan token. Dengan adanya dashboard administrator, proses monitoring, analisis, dan

pengelolaan data sistem dapat dilakukan secara lebih mudah, terstruktur, dan efisien.

## 5. File process.php (Pemroses Transaksi Backend)

```

1 <?php
2 include 'db.php';
3
4 if ($_SERVER["REQUEST_METHOD"] == "POST") {
5     $vol = $_POST['vol'];
6
7     $token = strtoupper(substr(md5($vol), 0, 8));
8
9     $sql = "INSERT INTO token (token_code, volume, status) VALUES ('$token', '$vol', 'active')";
10
11     if ($conn->query($sql) == TRUE) {
12         echo "Transaksi Berhasil";
13     } else {
14         echo "Gagal";
15     }
16 }
17
18 header("Location: index.php");
19 exit();
20 }
21
22 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-app.js"></script>
23 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-auth.js"></script>
24 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-database.js"></script>
25 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-qrcode.js"></script>
26 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-storage.js"></script>
27 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit.js"></script>
28 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-visualize.js"></script>
29 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-detection.js"></script>
30 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-ocr.js"></script>
31 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-text-recognition.js"></script>
32 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
33 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
34 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
35 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
36 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
37 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
38 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
39 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
40 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
41 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
42 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
43 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
44 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
45 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
46 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
47 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
48 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
49 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
50 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
51 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
52 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
53 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
54 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
55 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
56 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
57 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
58 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
59 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
60 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
61 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
62 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
63 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
64 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
65 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
66 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
67 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
68 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
69 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
70 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
71 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
72 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
73 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
74 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
75 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
76 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
77 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
78 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
79 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
80 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
81 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
82 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
83 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
84 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
85 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
86 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
87 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
88 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
89 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
90 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
91 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
92 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
93 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
94 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
95 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
96 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
97 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
98 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>
99 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-classification.js"></script>
100 <script src="https://www.googleapis.com/ajax/libs/firebase/4.2.0/firebase-ml-kit-image-labeling.js"></script>

```

Gambar 4. 18 Program Process.php

File process.php berfungsi sebagai pemroses transaksi pada sisi backend sistem. File ini menerima data volume pengisian yang dikirim dari halaman index.php melalui metode POST setelah pengguna memilih volume yang diinginkan. Selain itu, sistem melakukan validasi akses untuk memastikan bahwa halaman hanya dapat diakses melalui proses transaksi yang sah dan tidak dapat dijalankan secara langsung melalui URL.

Setelah data volume diterima, sistem akan menghasilkan token unik sebagai identitas transaksi, menyimpan data transaksi ke dalam database MySQL, serta menetapkan status awal token sebagai active. Informasi yang disimpan meliputi kode token, volume pengisian yang dipilih, status token, dan waktu transaksi. Token yang berhasil dibuat selanjutnya digunakan sebagai data utama dalam proses pembentukan QR Code yang akan dipindai oleh QR Code Scanner GM67 pada alat pengisian galon otomatis.

File process.php berperan sebagai penghubung antara antarmuka pemilihan volume, database, dan proses pembentukan QR Code sehingga seluruh transaksi dapat tercatat dan divalidasi secara terstruktur.

### 4.2.3 Mekanisme Pembuatan Token dan QR Code

Proses generate token dan QR Code merupakan inti dari mekanisme transaksi pada sistem pengisian galon otomatis ini. Proses ini sepenuhnya dijalankan oleh file process.php pada sisi server dan library JavaScript QRious pada sisi browser

pengguna. Secara keseluruhan, alur proses ini dapat dibagi menjadi empat tahap utama, yaitu pembangkitan token, penyimpanan ke database, rendering QR Code, serta penyediaan fitur unduh QR Code.

#### 4.2.3.1 Pembuatan Token Unik

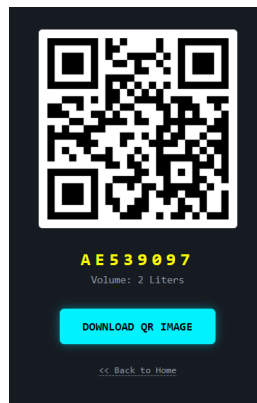
Token pada sistem ini dibuat menggunakan kombinasi fungsi bawaan PHP, yaitu `time()`, `md5()`, `substr()`, dan `strtoupper()`. Baris kode yang bertanggung jawab dalam proses ini adalah sebagai berikut.

```
$token = strtoupper(substr(md5(time()), 0, 8));
```

Gambar 4. 19 Kode Program Pembentukan Token Unik Menggunakan Fungsi MD5 pada PHP

Proses pembuatan token bekerja melalui empat tahap yang berurutan. Pada tahap pertama, fungsi `time()` mengambil nilai Unix timestamp pada saat transaksi flow terjadi, yaitu jumlah detik yang telah berlalu sejak 1 Januari 1970 pukul 00.00 UTC. Nilai ini bersifat unik karena selalu bertambah setiap detiknya, sehingga menjadi dasar keunikan setiap token yang dihasilkan. Pada tahap kedua, nilai timestamp tersebut diproses oleh fungsi `md5()` yang menghasilkan string hash sepanjang 32 karakter heksadesimal. Algoritma MD5 bekerja dengan mentransformasikan nilai input menjadi representasi hash yang tidak dapat diprediksi secara langsung. Pada tahap ketiga, fungsi `substr()` digunakan untuk mengambil 8 karakter pertama dari hasil hash 32 karakter tersebut. Pada tahap keempat, fungsi `strtoupper()` mengubah seluruh karakter huruf kecil menjadi huruf kapital agar token lebih mudah dibaca dan konsisten dalam format penyimpanannya.

Sebagai ilustrasi, apabila fungsi `time()` menghasilkan nilai 1716123456, maka `md5(1716123456)` akan menghasilkan string hash seperti a1b2c3d4e5f67890abcdef1234567890. Setelah dipotong 8 karakter pertama dan dikapitalisasi, token yang dihasilkan menjadi A1B2C3D4 sebagaimana dicontohkan pada Gambar 4.18.



Gambar 4. 20 Token QR Code yang dihasilkan

#### 4.2.3.2 Penyimpanan Token Ke Database

Setelah token berhasil dibangkitkan, sistem menyimpannya ke dalam tabel tokens pada database MySQL menggunakan perintah SQL INSERT sebagai berikut.

```
$sql = "INSERT INTO tokens (token_code, volume, status, purchase_date)
VALUES ('$token', '$vol', 'active', '$waktu_sekarang')";
```

Gambar 4. 21 Kode Program Penyimpanan Data Transaksi ke Database

Token disimpan bersama empat atribut pendukungnya secara bersamaan dalam satu operasi query. Kolom token\_code menyimpan token 8 karakter yang baru dibangkitkan sebagai identitas unik transaksi. Kolom volume menyimpan nilai volume dalam satuan mililiter sesuai pilihan pengguna, yaitu 2000, 5000, atau 15000. Kolom status diisi dengan nilai active yang menandakan bahwa token belum pernah digunakan dan masih dapat diproses oleh sistem. Kolom purchase\_date mencatat waktu transaksi secara otomatis dalam format Y-m-d H:i:s sesuai zona waktu WIB yang telah dikonfigurasi pada file db.php.

Penyimpanan ini menjadi titik awal siklus hidup token dalam sistem. Token dengan status active inilah yang selanjutnya akan divalidasi oleh mikrokontroler ESP32 melalui file api.php pada saat proses pengisian galon berlangsung.

### 4.2.3.3 Rendering QR Code pada Browser

Setelah token berhasil disimpan ke dalam database, file `process.php` mengirimkan respons dalam bentuk halaman HTML kepada browser pengguna. Di dalam respons tersebut, nilai token disisipkan secara dinamis oleh PHP sebagai parameter untuk library JavaScript QRious yang bertugas merender QR Code secara langsung di sisi browser tanpa memerlukan proses pembangkitan gambar di sisi server. Konfigurasi rendering QR Code dilakukan melalui potongan kode berikut.

```
var qr = new QRious({
  element: document.getElementById('qr'),
  value: '<?php echo $token; ?>',
  size: 200,
  background: 'white',
  foreground: 'black',
  level: 'H'
});
```

Gambar 4. 22 Kode Program Pembuatan QR Code Berdasarkan Token Menggunakan Library QRious

Library QRious bekerja dengan mengonversi string token menjadi representasi visual matriks QR Code menggunakan elemen `canvas` HTML5. Parameter `value` berisi token 8 karakter yang akan dikodekan ke dalam struktur matriks QR Code. Parameter `size` menentukan dimensi QR Code yang dihasilkan yaitu sebesar 200×200 piksel.

### 4.2.3.4 Fitur Unduh QR Code

Sistem menyediakan fitur unduh QR Code dalam format PNG melalui fungsi JavaScript `downloadQR()` yang dipanggil ketika pengguna menekan tombol `Download QR Image` pada halaman `process.php`. Fungsi ini dapat diaktifkan melalui elemen tombol berikut pada bagian HTML halaman.

```
<button onclick="downloadQR()" class="btn-download">
  Download QR Image
</button>
```

Gambar 4. 23 Kode Program Tombol Unduh QR Code

Atribut onclick pada elemen tombol tersebut menghubungkan aksi klik pengguna dengan fungsi downloadQR() secara langsung. Pendekatan ini dipilih karena proses unduhan yang dipicu dari dalam fungsi yang diinisiasi oleh interaksi pengguna akan diterima oleh browser sebagai aksi unduhan yang sah, sehingga tidak memerlukan konfirmasi tambahan maupun navigasi ke halaman lain.

```
function downloadQR() {
    var sourceCanvas = document.getElementById("qr");
    var borderSize = 20;

    var finalCanvas = document.createElement('canvas');
    var ctx = finalCanvas.getContext('2d');

    finalCanvas.width = sourceCanvas.width + (borderSize * 2);
    finalCanvas.height = sourceCanvas.height + (borderSize * 2);

    ctx.fillStyle = "#FFFFFF";
    ctx.fillRect(0, 0, finalCanvas.width, finalCanvas.height);

    ctx.drawImage(sourceCanvas, borderSize, borderSize);

    var image = finalCanvas.toDataURL("image/png");

    var link = document.createElement('a');
    link.download = 'HYDRIX-TOKEN-<?php echo $token; ?>.png';
    link.href = image;
    link.click();
}
```

Gambar 4. 24 Kode Program Tombol Download QR Code pada Website

Proses yang dijalankan oleh fungsi downloadQR() bekerja melalui beberapa tahapan yang berurutan. Pada tahap pertama, fungsi mengambil referensi elemen canvas yang berisi QR Code asli menggunakan perintah document.getElementById("qr"). Canvas ini berisi QR Code berukuran 200×200 piksel yang telah digenerate sebelumnya oleh library QRious. Pada tahap kedua, fungsi membuat sebuah canvas baru yang berukuran lebih besar, yaitu 240×240 piksel, yang merupakan hasil penjumlahan ukuran QR Code asli dengan dua kali nilai borderSize pada setiap dimensinya. Pada tahap ketiga, seluruh permukaan canvas baru diisi dengan warna putih menggunakan perintah ctx.fillRect() sebelum QR Code asli digambar di atasnya. Pada tahap keempat, QR Code asli digambar di atas canvas baru pada posisi koordinat (borderSize, borderSize) menggunakan perintah ctx.drawImage(), sehingga secara otomatis terbentuk area putih selebar 20 piksel pada keempat sisinya. Pada tahap kelima dan terakhir, canvas yang telah jadi

dikonversi menjadi URL data gambar dalam format PNG menggunakan perintah `finalCanvas.toDataURL("image/png")`, kemudian sebuah elemen tautan HTML dibuat secara dinamis dan diklik secara programatik untuk memicu proses unduhan pada browser pengguna.

Penambahan border putih tersebut bukan sekadar pertimbangan estetika, melainkan merupakan implementasi dari persyaratan teknis standar QR Code yang disebut *quiet zone*, yaitu area kosong di sekeliling matriks QR Code yang berfungsi membantu perangkat scanner membedakan batas QR Code dari latar belakangnya.

```
link.download = 'HYDRIX-TOKEN-<?php echo $token; ?>.png';
```

Gambar 4. 25 Kode Program Penamaan File QR Code Saat Proses Unduh

File hasil unduhan diberi nama secara dinamis mengikuti nilai token yang dihasilkan oleh sistem, sebagaimana ditunjukkan pada gambar 4.24. Dengan mekanisme penamaan dinamis tersebut, setiap file QR Code yang diunduh memiliki nama yang unik dan dapat langsung diidentifikasi tanpa risiko tertimpa oleh file dari transaksi lain. Sebagai contoh, apabila token yang dihasilkan adalah A1B2C3D4, maka file yang diunduh akan bernama HYDRIX-TOKEN-A1B2C3D4.png. Mekanisme ini juga memudahkan pengguna dalam mengelola dan mengarsipkan token QR Code yang dimiliki apabila melakukan beberapa kali pembelian, karena setiap file dapat langsung dikenali berdasarkan namanya tanpa perlu membuka isinya terlebih dahulu.



Gambar 4. 26 Hasil Unduhan Token dan QR Code

#### 4.2.3.5 Alur Keseluruhan Proses Generate Token dan QR Code

Secara keseluruhan, alur proses generate token dan QR Code dari sisi pengguna hingga tersimpan di database dan ditampilkan sebagai QR Code dapat dirangkum sebagai berikut. Pengguna memilih volume air pada halaman `index.php` kemudian mengkonfirmasi transaksi melalui modal konfirmasi. Setelah konfirmasi diberikan, data volume dikirimkan ke server melalui metode HTTP POST. File `process.php`

menerima data tersebut dan menjalankan proses pembangkitan token menggunakan `time()`, `md5()`, `substr()`, dan `strtoupper()`. Token yang dihasilkan kemudian disimpan ke dalam database dengan status `active` melalui perintah `INSERT`. Server mengirimkan respons HTML yang memuat nilai token kepada browser. Library `QRious` merender QR Code berdasarkan nilai token tersebut pada elemen `canvas` di browser. Halaman menampilkan QR Code beserta token dalam format teks kepada pengguna, yang selanjutnya dapat mengunduh QR Code tersebut sebagai file PNG untuk digunakan pada proses pengisian galon.

#### **4.2.4 Hosting dan Publikasi Website**

Sistem website dipublikasikan secara daring (online) menggunakan layanan cloud hosting. Pendekatan ini dipilih agar perangkat IoT ESP32 dapat berkomunikasi dengan server dari lokasi mana pun selama terhubung ke jaringan internet. Dengan sistem yang berjalan secara online, proses validasi token, pembaruan status transaksi, serta pencatatan data dapat dilakukan secara real-time tanpa terbatas oleh jaringan lokal.

Platform hosting yang digunakan adalah `Alwaysdata`, yang menerapkan arsitektur `Platform as a Service (PaaS)`. Model layanan ini memungkinkan pengelolaan aplikasi berbasis PHP dan basis data MySQL (`MariaDB`) tanpa memerlukan pengaturan infrastruktur server secara mandiri. Server yang disediakan aktif selama 24 jam penuh (24/7 uptime), sehingga sistem dapat diakses secara kontinu dan tidak bergantung pada perangkat pengembang. Selain itu, layanan ini menyediakan subdomain gratis, seperti `isigalonku.alwaysdata.net`, yang digunakan sebagai alamat utama akses sistem.

Salah satu aspek penting dari penggunaan `Alwaysdata` adalah dukungan sertifikat keamanan SSL/TLS otomatis yang memungkinkan penggunaan protokol HTTPS pada port 443. Fitur ini sangat krusial dalam integrasi dengan perangkat ESP32, karena komunikasi antara mikrokontroler dan server menggunakan library `WiFiClientSecure.h` yang memerlukan koneksi terenkripsi. Protokol HTTPS memungkinkan proses `secure handshake` antara klien (ESP32) dan server, sehingga meminimalkan risiko serangan seperti `Man-in-the-Middle (MITM)` saat pengiriman

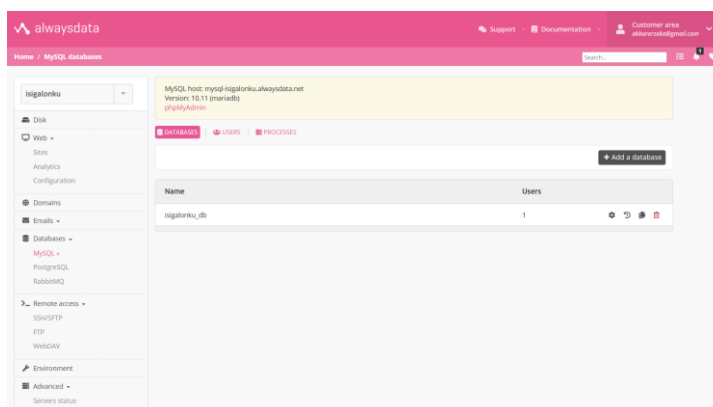
data token. Dengan demikian, keamanan data transaksi dan proses validasi sistem dapat terjamin.

Proses publikasi source code website dari komputer pengembang ke server Alwaysdata dilakukan menggunakan protokol File Transfer Protocol (FTP). Perangkat lunak klien yang digunakan adalah FileZilla, yang berfungsi sebagai media transfer file antara local site (komputer pengembang) dan remote site (direktori publik www/ pada server Alwaysdata). Proses koneksi dilakukan dengan memasukkan alamat host, username, dan password FTP yang diberikan oleh penyedia hosting. Setelah autentikasi berhasil, FileZilla membuka jalur komunikasi dan memungkinkan pengunggahan file berekstensi .php melalui metode drag-and-drop.

Melalui mekanisme ini, setiap perubahan kode yang diunggah ke server dapat langsung diterapkan secara real-time tanpa konfigurasi tambahan. Hal ini mendukung proses pengembangan dan pemeliharaan sistem secara efisien. Integrasi antara layanan hosting Alwaysdata, protokol HTTPS, serta mekanisme FTP menggunakan FileZilla memastikan bahwa sistem website dapat beroperasi secara stabil, aman, dan terhubung langsung dengan perangkat ESP32 dalam satu ekosistem IoT yang terintegrasi.

#### **4.2.5 Pembuatan Database**

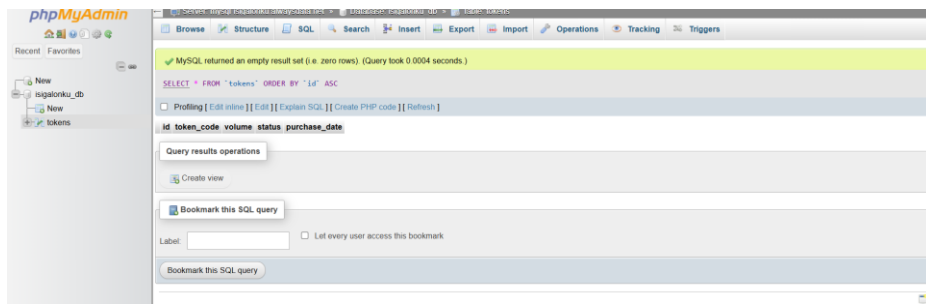
Pembuatan database pada sistem pengisian galon otomatis bertujuan untuk menyimpan, mengelola, dan memproses seluruh data transaksi yang terjadi pada sistem. Basis data berfungsi sebagai pusat penyimpanan informasi token, volume air yang dibeli, status penggunaan token, serta waktu transaksi. Dengan adanya database yang terstruktur, sistem dapat melakukan validasi token secara otomatis dan memastikan setiap token hanya dapat digunakan satu kali.



Gambar 4. 27 Panel Pembuatan Database pada Alwaysdata

Database yang digunakan pada sistem ini adalah MySQL (MariaDB) yang disediakan oleh layanan hosting Alwaysdata. Pemilihan database relasional ini didasarkan pada kemampuannya dalam mengelola data terstruktur menggunakan tabel, mendukung bahasa kueri SQL (Structured Query Language), serta kompatibel dengan bahasa pemrograman PHP yang digunakan pada sisi server. Seluruh proses manajemen database dilakukan melalui panel administrasi hosting atau menggunakan fitur phpMyAdmin yang disediakan oleh layanan hosting. Pada tahap awal, dilakukan pembuatan database melalui panel Alwaysdata sebagaimana ditunjukkan pada Gambar 4.26. Pada tahap ini dibuat sebuah database dengan nama sesuai kebutuhan sistem, yang kemudian akan dihubungkan dengan file konfigurasi koneksi (db.php) pada website.

Setelah database berhasil dibuat, tahap berikutnya adalah pembuatan tabel utama menggunakan phpMyAdmin. Antarmuka phpMyAdmin memungkinkan pengguna untuk membuat struktur tabel, menentukan tipe data setiap field, serta mengatur primary key dan atribut lainnya. Proses ini ditunjukkan pada Gambar 4.20, di mana database telah terhubung dan siap dikelola.



Gambar 4. 28 Tampilan Antarmuka phpMyAdmin pada Database Sistem

Struktur database dirancang sederhana namun efektif untuk mendukung kebutuhan sistem. Database terdiri dari satu tabel utama, yaitu tabel tokens, yang berfungsi untuk menyimpan data transaksi pembelian air. Tabel ini memiliki beberapa field utama, antara lain:

- id sebagai primary key dengan tipe data integer dan auto increment,
- token sebagai kode unik hasil generasi sistem,
- volume untuk menyimpan jumlah air dalam satuan mililiter,
- status untuk menunjukkan kondisi token (active atau used),
- created\_at untuk mencatat waktu pembuatan token.

Field id berfungsi sebagai identitas unik setiap baris data, sedangkan field token digunakan sebagai parameter utama dalam proses validasi oleh perangkat ESP32. Field status berperan penting dalam mencegah penyalahgunaan token, karena sistem hanya akan memproses token dengan status “active”. Setelah proses pengisian selesai, status akan diperbarui menjadi “used” melalui perintah SQL UPDATE yang dijalankan oleh file api.php.

Hasil penyimpanan data transaksi yang telah berjalan dapat dilihat pada Gambar 4.28, yang menampilkan isi tabel tokens di dalam phpMyAdmin. Pada gambar tersebut terlihat beberapa record data yang terdiri dari id, token, volume, status, serta waktu pembuatan.



Gambar 4. 29 Tampilan Data Tabel Tokens pada phpMyAdmin

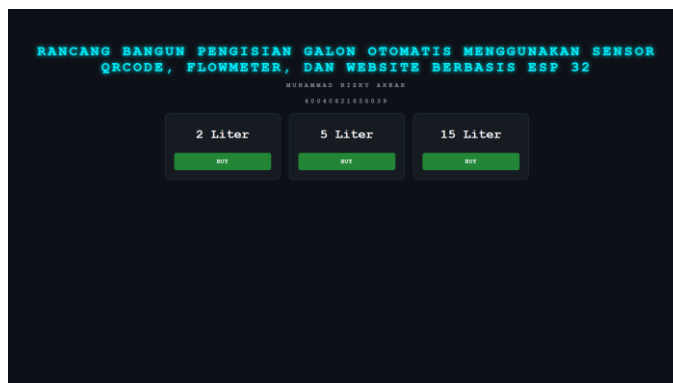
Proses pembuatan tabel dilakukan menggunakan perintah SQL CREATE TABLE melalui phpMyAdmin. Selanjutnya, sistem website secara otomatis akan menambahkan data baru ke dalam tabel menggunakan perintah INSERT INTO ketika terjadi pembelian token. Pada sisi perangkat ESP32, proses validasi dilakukan dengan perintah SELECT untuk memeriksa apakah token tersedia dan masih aktif. Setelah pengisian selesai, sistem menjalankan perintah UPDATE untuk mengubah status token. Selain itu, pada halaman admin tersedia fitur penghapusan data menggunakan perintah DELETE untuk manajemen riwayat transaksi.

#### **4.2.6 Tampilan Website**

Tampilan website pada sistem pengisian galon otomatis dirancang sebagai antarmuka utama yang menghubungkan pengguna dengan sistem berbasis Internet of Things (IoT). Website ini memiliki dua fungsi utama, yaitu sebagai media transaksi pembelian volume air oleh pelanggan serta sebagai pusat monitoring dan pengelolaan data bagi administrator. Perancangan tampilan dilakukan dengan memperhatikan aspek kemudahan penggunaan, kejelasan informasi, keamanan akses, serta integrasi langsung dengan sistem database dan perangkat ESP32.

Desain antarmuka dibangun menggunakan struktur HTML sebagai kerangka halaman, CSS untuk pengaturan tata letak dan tampilan visual, serta JavaScript untuk mendukung interaksi dinamis seperti konfirmasi transaksi dan penampilan QR Code. Tata letak halaman dirancang responsif agar dapat diakses melalui berbagai perangkat, baik smartphone maupun komputer, tanpa mengurangi kualitas tampilan dan fungsi sistem.

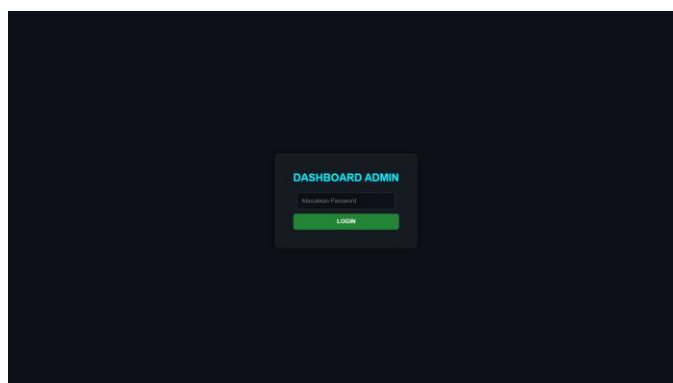
Pada halaman utama, pengguna disajikan pilihan volume air dalam bentuk kartu produk yang tersusun secara terpusat. Setiap kartu menampilkan informasi kapasitas dalam satuan liter beserta tombol aksi untuk melakukan pembelian. Desain ini bertujuan agar pengguna dapat dengan mudah memilih volume yang diinginkan tanpa kebingungan navigasi.



Gambar 4. 30 Tampilan Website Pemilihan Galon

Gambar 4.29 Tampilan Website Pemilihan Galon menunjukkan halaman utama yang menampilkan pilihan volume 2 Liter, 5 Liter, dan 15 Liter. Masing-masing pilihan dilengkapi tombol pembelian yang akan memulai proses transaksi. Setelah tombol ditekan, sistem akan menampilkan konfirmasi untuk mencegah kesalahan input. Jika pengguna menyetujui, data volume dikirim ke sistem backend untuk diproses dan menghasilkan token unik dalam bentuk QR Code. QR Code tersebut berfungsi sebagai identitas transaksi yang nantinya dipindai oleh alat pengisian galon otomatis.

Selain halaman pengguna, sistem juga menyediakan halaman khusus administrator yang dilindungi mekanisme autentikasi berbasis session PHP. Halaman ini hanya dapat diakses oleh pengguna yang memiliki kredensial yang valid.

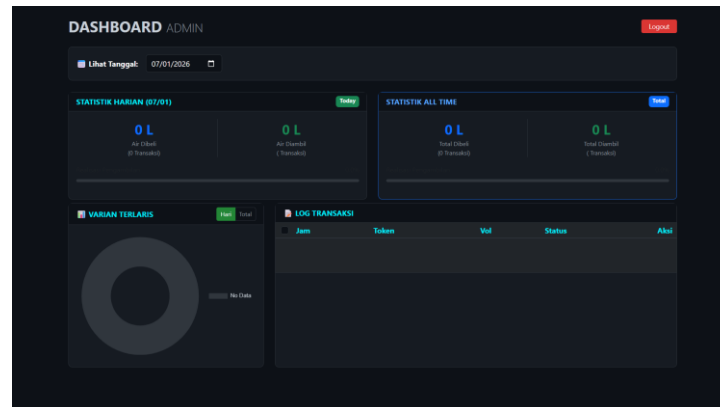


Gambar 4. 31 Tampilan Website Login Admin

Gambar 4.30 Tampilan Website Admin memperlihatkan halaman login administrator. Pada halaman ini tersedia kolom input kata sandi dan tombol login.

Sistem akan memverifikasi kata sandi yang dimasukkan sebelum memberikan akses ke dashboard monitoring. Mekanisme ini bertujuan untuk menjaga keamanan data transaksi dan mencegah akses oleh pihak yang tidak berwenang.

Setelah proses login berhasil, administrator akan diarahkan ke halaman dashboard monitoring yang menampilkan ringkasan data transaksi secara real-time.



Gambar 4. 32 Tampilan Website Monitoring

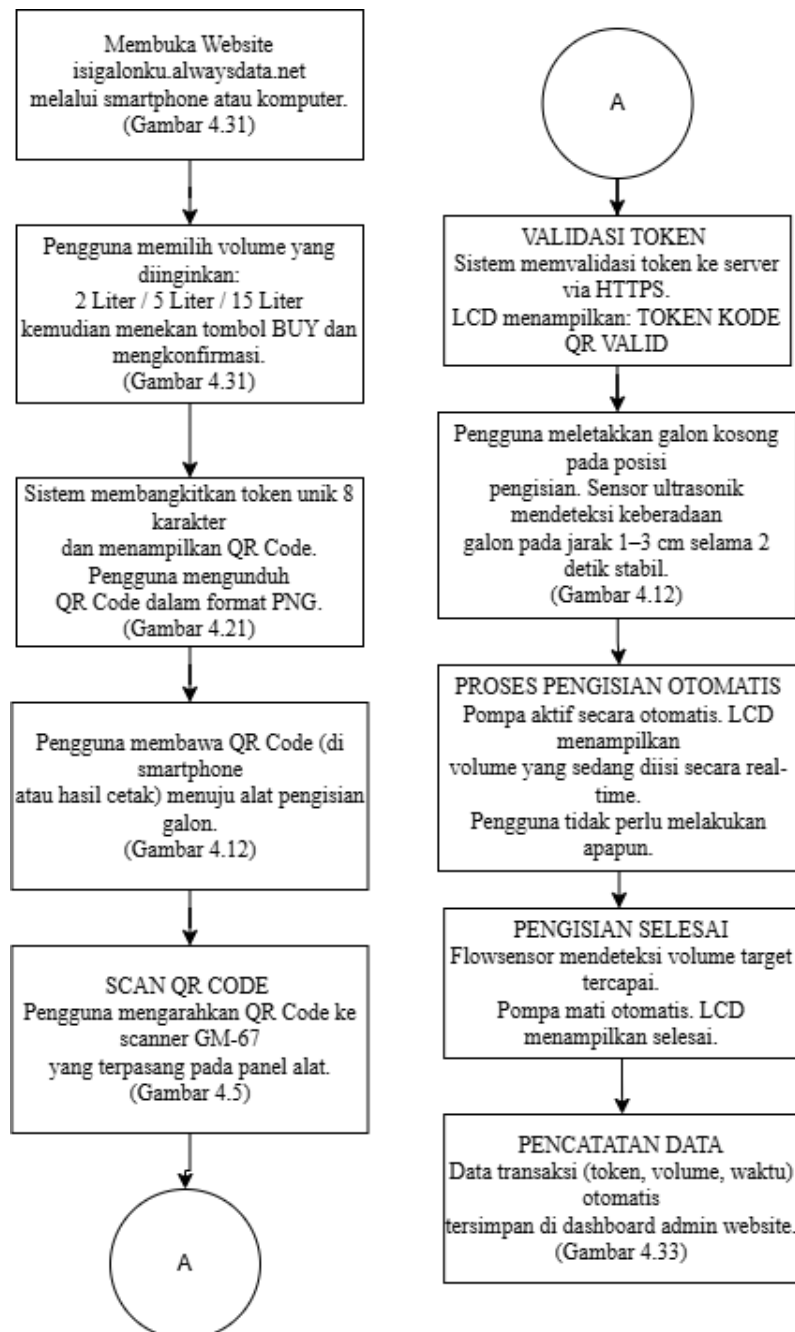
Gambar 4.31 Tampilan Website Monitoring menampilkan dashboard yang terdiri dari beberapa bagian utama, yaitu panel statistik, grafik visualisasi, dan tabel riwayat transaksi. Panel statistik menampilkan total volume air terjual, total volume air terealisasi, serta jumlah transaksi yang tercatat dalam database. Data tersebut diperoleh melalui proses query agregasi pada database MySQL.

Grafik visualisasi berbentuk diagram donat digunakan untuk menampilkan distribusi penjualan berdasarkan variasi volume. Visualisasi ini dibuat menggunakan pustaka Chart.js sehingga informasi dapat ditampilkan secara informatif dan mudah dipahami. Selain itu, tabel riwayat transaksi menyajikan detail data seperti ID transaksi, token, volume, status (active atau used), serta waktu transaksi. Data tersebut diperbarui secara otomatis setiap kali terjadi transaksi baru.

Integrasi antara website dan perangkat ESP32 dilakukan melalui sistem API berbasis protokol HTTPS. Ketika pengguna melakukan pembelian, data token disimpan dalam database dengan status “active”. Saat QR Code dipindai oleh alat, sistem akan melakukan validasi melalui API. Setelah proses pengisian selesai, status token diperbarui menjadi “used” dan perubahan tersebut langsung tercermin pada dashboard administrator. Dengan mekanisme ini, website berfungsi tidak

hanya sebagai media transaksi, tetapi juga sebagai sistem monitoring yang terhubung langsung dengan perangkat keras secara daring.

### 4.3 Alur Penggunaan Alat Alat Pengisian Galon Otomatis



Gambar 4. 33 Flowchart Penggunaan Alat

Alur penggunaan sistem dimulai ketika pengguna mengakses website pengisian galon melalui smartphone atau komputer. Pada halaman website,

pengguna memilih volume air yang diinginkan, yaitu 2 liter, 5 liter, atau 15 liter, kemudian menekan tombol pembelian (BUY) dan melakukan konfirmasi transaksi. Setelah proses berhasil, sistem secara otomatis menghasilkan token unik berupa QR Code dengan panjang 8 karakter yang hanya berlaku untuk satu kali penggunaan. QR Code tersebut ditampilkan pada website dan dapat diunduh oleh pengguna dalam format PNG sebagai media autentikasi pada alat pengisian.

Selanjutnya, pengguna menuju alat pengisian galon dan melakukan pemindaian QR Code menggunakan scanner GM-67 yang terpasang pada panel alat. Data QR Code yang terbaca kemudian dikirim ke server melalui komunikasi HTTPS untuk dilakukan validasi token. Pada tahap ini sistem memeriksa keaslian token, status penggunaan token, serta kesesuaian data volume yang telah dipilih sebelumnya. Jika token dinyatakan valid, LCD pada alat akan menampilkan informasi bahwa QR Code berhasil diverifikasi dan siap digunakan untuk proses pengisian.

Setelah validasi berhasil, pengguna meletakkan galon kosong pada area pengisian. Sensor ultrasonik yang terpasang pada alat akan mendeteksi keberadaan galon berdasarkan jarak yang telah ditentukan. Ketika galon terdeteksi dengan kondisi yang memenuhi syarat, sistem secara otomatis mengaktifkan pompa air tanpa memerlukan intervensi operator. Selama proses pengisian berlangsung, sensor water flow YF-S201 melakukan pengukuran debit dan volume air yang mengalir secara real-time. Data volume yang telah terisi terus dihitung oleh mikrokontroler dan ditampilkan pada LCD sehingga pengguna dapat memantau progres pengisian secara langsung.

Proses pengisian akan berhenti secara otomatis ketika volume air yang terukur oleh sensor telah mencapai nilai target sesuai pilihan pengguna pada saat pembelian token. LCD kemudian menampilkan informasi bahwa proses pengisian telah selesai dilakukan. Pada tahap akhir, seluruh data transaksi seperti token QR Code, volume air yang diambil, waktu transaksi, serta informasi penggunaan token dicatat secara otomatis dan disimpan ke database. Data tersebut selanjutnya dapat dipantau melalui dashboard admin website secara real-time untuk keperluan monitoring, pencatatan penggunaan, dan evaluasi kinerja sistem.

## **BAB V**

### **PENGUJIAN DAN PEMBAHASAN**

#### **5.1 Pengukuran dan Pengujian**

Untuk memastikan sistem pengisian galon otomatis berbasis Internet of Things (IoT) dapat bekerja sesuai dengan tujuan perancangan, dilakukan serangkaian pengukuran dan pengujian secara menyeluruh. Pengujian ini mencakup aspek fungsionalitas perangkat keras, akurasi pengukuran volume air, validasi token berbasis QR Code, konsistensi pencatatan data pada database, visualisasi monitoring pada dashboard administrator, serta kestabilan komunikasi antara ESP32 dan server hosting.

Selain pengujian pada kondisi operasional normal, sistem juga diuji dalam kondisi abnormal seperti token tidak valid, token telah digunakan, galon tidak terdeteksi, galon diangkat saat pengisian berlangsung, serta simulasi gangguan koneksi jaringan. Setiap pengujian dirancang untuk mengevaluasi performa sistem secara terintegrasi sekaligus mengidentifikasi potensi kekurangan yang dapat diperbaiki pada pengembangan selanjutnya.

##### **5.1.1 Tujuan dan Parameter Pengujian**

Pengujian dilakukan untuk memastikan bahwa seluruh komponen sistem dapat bekerja sesuai dengan logika yang telah dirancang pada Bab IV. Fokus utama pengujian adalah mengevaluasi kemampuan sistem dalam menjalankan proses transaksi air secara otomatis, aman, dan tercatat secara digital.

Adapun parameter yang diuji meliputi:

##### **1. Validasi Token QR Code**

Memastikan hanya token dengan status *active* yang dapat digunakan, sedangkan token dengan status *used* atau tidak terdaftar harus ditolak oleh sistem.

## 2. Akurasi Pengukuran Volume Air

Membandingkan hasil pembacaan sensor flowmeter dengan pengukuran manual menggunakan gelas ukur atau alat ukur volume sebagai pembanding.

## 3. Deteksi Keberadaan Galon

Memastikan sensor ultrasonik hanya mengizinkan proses pengisian saat galon berada pada posisi yang benar dan stabil.

## 4. Konsistensi Data Database

Memastikan data transaksi yang tersimpan pada database sesuai dengan volume aktual yang dikeluarkan sistem.

### 5.1.2 Prosedur dan Lingkungan Pengujian

Pengujian dilakukan pada lokasi yang memiliki koneksi internet stabil karena sistem menggunakan komunikasi HTTPS dengan server hosting Alwaysdata.

Peralatan yang digunakan dalam pengujian meliputi:

1. Alat pengisian galon otomatis berbasis ESP32
2. Smartphone atau laptop untuk mengakses website
3. QR Code hasil pembelian token
4. Gelas ukur sebagai alat pembanding volume
5. Jaringan WiFi dengan akses internet

Langkah pengujian dilakukan sebagai berikut:

1. Menyalakan alat dan memastikan koneksi WiFi berhasil.
2. Melakukan pembelian volume melalui website.
3. Memindai QR Code pada alat.
4. Meletakkan galon pada posisi yang telah ditentukan.
5. Mengamati proses pengisian hingga selesai.
6. Membandingkan volume hasil pengisian dengan pengukuran manual.
7. Memeriksa data transaksi pada database dan dashboard administrator.
8. Menguji kembali token yang telah digunakan untuk memastikan sistem menolak penggunaan ulang.


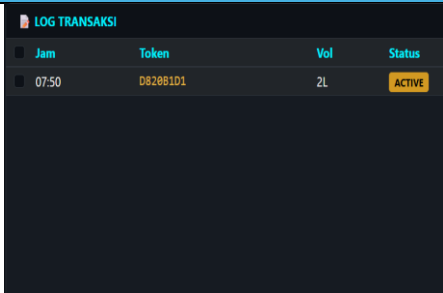

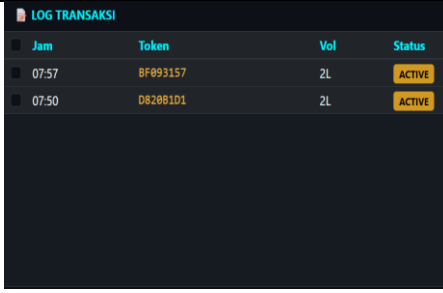

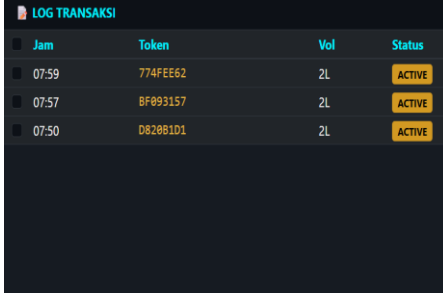
Simulasi gangguan jaringan dilakukan dengan memutus koneksi internet saat proses validasi token untuk mengamati respon sistem.

### 5.1.3 Pengujian Konsistensi Data Transaksi

Pengujian konsistensi data transaksi dilakukan untuk memastikan bahwa setiap transaksi pengisian galon tercatat secara konsisten pada database dan ditampilkan secara akurat pada dashboard administrator.

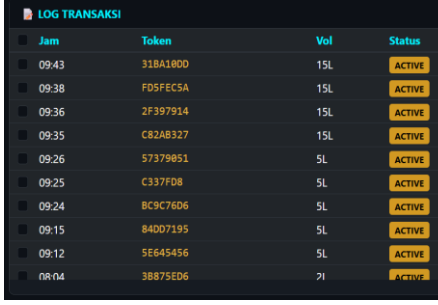

Pengujian ini bertujuan untuk mengecek kesesuaian data antara proses transaksi yang terjadi pada sistem dengan data yang tersimpan pada server. Parameter yang diamati dalam pengujian ini meliputi token QR Code yang digunakan, volume pengisian dalam satuan mililiter (mL), status transaksi apakah masih aktif atau sudah digunakan (active/used), serta waktu transaksi berlangsung.

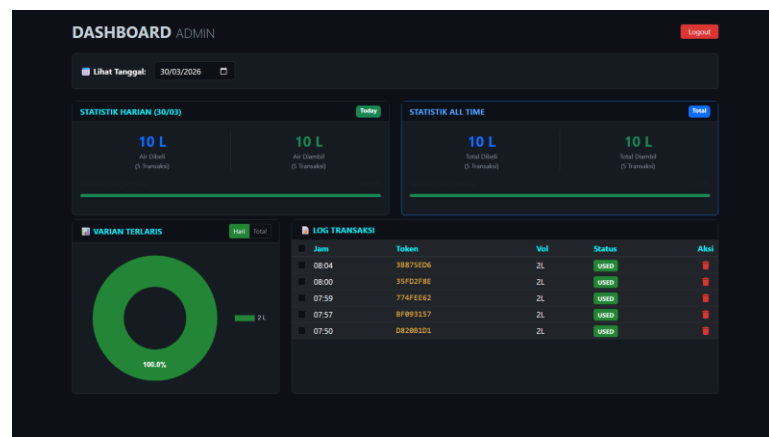
Tabel 5. 1 Pengujian Transaksi

No	Volume	Kode	QRCode	Tampilan Dashboard
1	2 Liter	D820B1D1		
2	2 Liter	BF093157		
3	2 Liter	774FEE62		

No	Volume	Kode	QRCode	Tampilan Dashboard																																				
4	2 Liter	35FD2F8E		<p><b>LOG TRANSAKSI</b></p> <table border="1"> <thead> <tr> <th>Jam</th> <th>Token</th> <th>Vol</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>08:00</td> <td>35FD2F8E</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>07:59</td> <td>774FEE62</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>07:57</td> <td>BF093157</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>07:50</td> <td>D828B1D1</td> <td>2L</td> <td>ACTIVE</td> </tr> </tbody> </table>	Jam	Token	Vol	Status	08:00	35FD2F8E	2L	ACTIVE	07:59	774FEE62	2L	ACTIVE	07:57	BF093157	2L	ACTIVE	07:50	D828B1D1	2L	ACTIVE																
Jam	Token	Vol	Status																																					
08:00	35FD2F8E	2L	ACTIVE																																					
07:59	774FEE62	2L	ACTIVE																																					
07:57	BF093157	2L	ACTIVE																																					
07:50	D828B1D1	2L	ACTIVE																																					
5	2 Liter	3B8753D6		<p><b>LOG TRANSAKSI</b></p> <table border="1"> <thead> <tr> <th>Jam</th> <th>Token</th> <th>Vol</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>08:04</td> <td>3B8753D6</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>08:00</td> <td>35FD2F8E</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>07:59</td> <td>774FEE62</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>07:57</td> <td>BF093157</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>07:50</td> <td>D828B1D1</td> <td>2L</td> <td>ACTIVE</td> </tr> </tbody> </table>	Jam	Token	Vol	Status	08:04	3B8753D6	2L	ACTIVE	08:00	35FD2F8E	2L	ACTIVE	07:59	774FEE62	2L	ACTIVE	07:57	BF093157	2L	ACTIVE	07:50	D828B1D1	2L	ACTIVE												
Jam	Token	Vol	Status																																					
08:04	3B8753D6	2L	ACTIVE																																					
08:00	35FD2F8E	2L	ACTIVE																																					
07:59	774FEE62	2L	ACTIVE																																					
07:57	BF093157	2L	ACTIVE																																					
07:50	D828B1D1	2L	ACTIVE																																					
6	5 Liter	5E645456		<p><b>LOG TRANSAKSI</b></p> <table border="1"> <thead> <tr> <th>Jam</th> <th>Token</th> <th>Vol</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>09:12</td> <td>5E645456</td> <td>5L</td> <td>ACTIVE</td> </tr> <tr> <td>08:04</td> <td>3B8753D6</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>08:00</td> <td>35FD2F8E</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>07:59</td> <td>774FEE62</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>07:57</td> <td>BF093157</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>07:50</td> <td>D828B1D1</td> <td>2L</td> <td>ACTIVE</td> </tr> </tbody> </table>	Jam	Token	Vol	Status	09:12	5E645456	5L	ACTIVE	08:04	3B8753D6	2L	ACTIVE	08:00	35FD2F8E	2L	ACTIVE	07:59	774FEE62	2L	ACTIVE	07:57	BF093157	2L	ACTIVE	07:50	D828B1D1	2L	ACTIVE								
Jam	Token	Vol	Status																																					
09:12	5E645456	5L	ACTIVE																																					
08:04	3B8753D6	2L	ACTIVE																																					
08:00	35FD2F8E	2L	ACTIVE																																					
07:59	774FEE62	2L	ACTIVE																																					
07:57	BF093157	2L	ACTIVE																																					
07:50	D828B1D1	2L	ACTIVE																																					
7	5 Liter	84DD7195		<p><b>LOG TRANSAKSI</b></p> <table border="1"> <thead> <tr> <th>Jam</th> <th>Token</th> <th>Vol</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>09:15</td> <td>84DD7195</td> <td>5L</td> <td>ACTIVE</td> </tr> <tr> <td>09:12</td> <td>5E645456</td> <td>5L</td> <td>ACTIVE</td> </tr> <tr> <td>08:04</td> <td>3B8753D6</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>08:00</td> <td>35FD2F8E</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>07:59</td> <td>774FEE62</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>07:57</td> <td>BF093157</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>07:50</td> <td>D828B1D1</td> <td>2L</td> <td>ACTIVE</td> </tr> </tbody> </table>	Jam	Token	Vol	Status	09:15	84DD7195	5L	ACTIVE	09:12	5E645456	5L	ACTIVE	08:04	3B8753D6	2L	ACTIVE	08:00	35FD2F8E	2L	ACTIVE	07:59	774FEE62	2L	ACTIVE	07:57	BF093157	2L	ACTIVE	07:50	D828B1D1	2L	ACTIVE				
Jam	Token	Vol	Status																																					
09:15	84DD7195	5L	ACTIVE																																					
09:12	5E645456	5L	ACTIVE																																					
08:04	3B8753D6	2L	ACTIVE																																					
08:00	35FD2F8E	2L	ACTIVE																																					
07:59	774FEE62	2L	ACTIVE																																					
07:57	BF093157	2L	ACTIVE																																					
07:50	D828B1D1	2L	ACTIVE																																					
8	5 Liter	BC9C76D6		<p><b>LOG TRANSAKSI</b></p> <table border="1"> <thead> <tr> <th>Jam</th> <th>Token</th> <th>Vol</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>09:24</td> <td>BC9C76D6</td> <td>5L</td> <td>ACTIVE</td> </tr> <tr> <td>09:15</td> <td>84DD7195</td> <td>5L</td> <td>ACTIVE</td> </tr> <tr> <td>09:12</td> <td>5E645456</td> <td>5L</td> <td>ACTIVE</td> </tr> <tr> <td>08:04</td> <td>3B8753D6</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>08:00</td> <td>35FD2F8E</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>07:59</td> <td>774FEE62</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>07:57</td> <td>BF093157</td> <td>2L</td> <td>ACTIVE</td> </tr> <tr> <td>07:50</td> <td>D828B1D1</td> <td>2L</td> <td>ACTIVE</td> </tr> </tbody> </table>	Jam	Token	Vol	Status	09:24	BC9C76D6	5L	ACTIVE	09:15	84DD7195	5L	ACTIVE	09:12	5E645456	5L	ACTIVE	08:04	3B8753D6	2L	ACTIVE	08:00	35FD2F8E	2L	ACTIVE	07:59	774FEE62	2L	ACTIVE	07:57	BF093157	2L	ACTIVE	07:50	D828B1D1	2L	ACTIVE
Jam	Token	Vol	Status																																					
09:24	BC9C76D6	5L	ACTIVE																																					
09:15	84DD7195	5L	ACTIVE																																					
09:12	5E645456	5L	ACTIVE																																					
08:04	3B8753D6	2L	ACTIVE																																					
08:00	35FD2F8E	2L	ACTIVE																																					
07:59	774FEE62	2L	ACTIVE																																					
07:57	BF093157	2L	ACTIVE																																					
07:50	D828B1D1	2L	ACTIVE																																					

No	Volume	Kode	QRCode	Tampilan Dashboard																																												
9	5 Liter	C337FD8		<p><b>LOG TRANSAKSI</b></p> <table border="1"> <thead> <tr> <th>Jam</th> <th>Token</th> <th>Vol</th> <th>Status</th> </tr> </thead> <tbody> <tr><td>09:25</td><td>C337FD8</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:24</td><td>BC9C76D6</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:15</td><td>84DD7195</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:12</td><td>5E645456</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>08:04</td><td>3B875ED6</td><td>2L</td><td>ACTIVE</td></tr> <tr><td>08:00</td><td>35FD2F8E</td><td>2L</td><td>ACTIVE</td></tr> <tr><td>07:59</td><td>774FEE62</td><td>2L</td><td>ACTIVE</td></tr> <tr><td>07:57</td><td>BF093157</td><td>2L</td><td>ACTIVE</td></tr> <tr><td>07:50</td><td>D82081D1</td><td>2L</td><td>ACTIVE</td></tr> </tbody> </table>	Jam	Token	Vol	Status	09:25	C337FD8	5L	ACTIVE	09:24	BC9C76D6	5L	ACTIVE	09:15	84DD7195	5L	ACTIVE	09:12	5E645456	5L	ACTIVE	08:04	3B875ED6	2L	ACTIVE	08:00	35FD2F8E	2L	ACTIVE	07:59	774FEE62	2L	ACTIVE	07:57	BF093157	2L	ACTIVE	07:50	D82081D1	2L	ACTIVE				
Jam	Token	Vol	Status																																													
09:25	C337FD8	5L	ACTIVE																																													
09:24	BC9C76D6	5L	ACTIVE																																													
09:15	84DD7195	5L	ACTIVE																																													
09:12	5E645456	5L	ACTIVE																																													
08:04	3B875ED6	2L	ACTIVE																																													
08:00	35FD2F8E	2L	ACTIVE																																													
07:59	774FEE62	2L	ACTIVE																																													
07:57	BF093157	2L	ACTIVE																																													
07:50	D82081D1	2L	ACTIVE																																													
10	5 Liter	57379051		<p><b>LOG TRANSAKSI</b></p> <table border="1"> <thead> <tr> <th>Jam</th> <th>Token</th> <th>Vol</th> <th>Status</th> </tr> </thead> <tbody> <tr><td>09:26</td><td>57379051</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:25</td><td>C337FD8</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:24</td><td>BC9C76D6</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:15</td><td>84DD7195</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:12</td><td>5E645456</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>08:04</td><td>3B875ED6</td><td>2L</td><td>ACTIVE</td></tr> <tr><td>08:00</td><td>35FD2F8E</td><td>2L</td><td>ACTIVE</td></tr> <tr><td>07:59</td><td>774FEE62</td><td>2L</td><td>ACTIVE</td></tr> <tr><td>07:57</td><td>BF093157</td><td>2L</td><td>ACTIVE</td></tr> <tr><td>07:50</td><td>D82081D1</td><td>2L</td><td>ACTIVE</td></tr> </tbody> </table>	Jam	Token	Vol	Status	09:26	57379051	5L	ACTIVE	09:25	C337FD8	5L	ACTIVE	09:24	BC9C76D6	5L	ACTIVE	09:15	84DD7195	5L	ACTIVE	09:12	5E645456	5L	ACTIVE	08:04	3B875ED6	2L	ACTIVE	08:00	35FD2F8E	2L	ACTIVE	07:59	774FEE62	2L	ACTIVE	07:57	BF093157	2L	ACTIVE	07:50	D82081D1	2L	ACTIVE
Jam	Token	Vol	Status																																													
09:26	57379051	5L	ACTIVE																																													
09:25	C337FD8	5L	ACTIVE																																													
09:24	BC9C76D6	5L	ACTIVE																																													
09:15	84DD7195	5L	ACTIVE																																													
09:12	5E645456	5L	ACTIVE																																													
08:04	3B875ED6	2L	ACTIVE																																													
08:00	35FD2F8E	2L	ACTIVE																																													
07:59	774FEE62	2L	ACTIVE																																													
07:57	BF093157	2L	ACTIVE																																													
07:50	D82081D1	2L	ACTIVE																																													
11	15 Liter	C82AB327		<p><b>LOG TRANSAKSI</b></p> <table border="1"> <thead> <tr> <th>Jam</th> <th>Token</th> <th>Vol</th> <th>Status</th> </tr> </thead> <tbody> <tr><td>09:35</td><td>C82AB327</td><td>15L</td><td>ACTIVE</td></tr> <tr><td>09:26</td><td>57379051</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:25</td><td>C337FD8</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:24</td><td>BC9C76D6</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:15</td><td>84DD7195</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:12</td><td>5E645456</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>08:04</td><td>3B875ED6</td><td>2L</td><td>ACTIVE</td></tr> <tr><td>08:00</td><td>35FD2F8E</td><td>2L</td><td>ACTIVE</td></tr> <tr><td>07:59</td><td>774FEE62</td><td>2L</td><td>ACTIVE</td></tr> <tr><td>07:57</td><td>BF093157</td><td>2L</td><td>ACTIVE</td></tr> </tbody> </table>	Jam	Token	Vol	Status	09:35	C82AB327	15L	ACTIVE	09:26	57379051	5L	ACTIVE	09:25	C337FD8	5L	ACTIVE	09:24	BC9C76D6	5L	ACTIVE	09:15	84DD7195	5L	ACTIVE	09:12	5E645456	5L	ACTIVE	08:04	3B875ED6	2L	ACTIVE	08:00	35FD2F8E	2L	ACTIVE	07:59	774FEE62	2L	ACTIVE	07:57	BF093157	2L	ACTIVE
Jam	Token	Vol	Status																																													
09:35	C82AB327	15L	ACTIVE																																													
09:26	57379051	5L	ACTIVE																																													
09:25	C337FD8	5L	ACTIVE																																													
09:24	BC9C76D6	5L	ACTIVE																																													
09:15	84DD7195	5L	ACTIVE																																													
09:12	5E645456	5L	ACTIVE																																													
08:04	3B875ED6	2L	ACTIVE																																													
08:00	35FD2F8E	2L	ACTIVE																																													
07:59	774FEE62	2L	ACTIVE																																													
07:57	BF093157	2L	ACTIVE																																													
12	15 Liter	2F397914		<p><b>LOG TRANSAKSI</b></p> <table border="1"> <thead> <tr> <th>Jam</th> <th>Token</th> <th>Vol</th> <th>Status</th> </tr> </thead> <tbody> <tr><td>09:36</td><td>2F397914</td><td>15L</td><td>ACTIVE</td></tr> <tr><td>09:35</td><td>C82AB327</td><td>15L</td><td>ACTIVE</td></tr> <tr><td>09:26</td><td>57379051</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:25</td><td>C337FD8</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:24</td><td>BC9C76D6</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:15</td><td>84DD7195</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:12</td><td>5E645456</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>08:04</td><td>3B875ED6</td><td>2L</td><td>ACTIVE</td></tr> <tr><td>08:00</td><td>35FD2F8E</td><td>2L</td><td>ACTIVE</td></tr> <tr><td>07:59</td><td>774FEE62</td><td>2L</td><td>ACTIVE</td></tr> </tbody> </table>	Jam	Token	Vol	Status	09:36	2F397914	15L	ACTIVE	09:35	C82AB327	15L	ACTIVE	09:26	57379051	5L	ACTIVE	09:25	C337FD8	5L	ACTIVE	09:24	BC9C76D6	5L	ACTIVE	09:15	84DD7195	5L	ACTIVE	09:12	5E645456	5L	ACTIVE	08:04	3B875ED6	2L	ACTIVE	08:00	35FD2F8E	2L	ACTIVE	07:59	774FEE62	2L	ACTIVE
Jam	Token	Vol	Status																																													
09:36	2F397914	15L	ACTIVE																																													
09:35	C82AB327	15L	ACTIVE																																													
09:26	57379051	5L	ACTIVE																																													
09:25	C337FD8	5L	ACTIVE																																													
09:24	BC9C76D6	5L	ACTIVE																																													
09:15	84DD7195	5L	ACTIVE																																													
09:12	5E645456	5L	ACTIVE																																													
08:04	3B875ED6	2L	ACTIVE																																													
08:00	35FD2F8E	2L	ACTIVE																																													
07:59	774FEE62	2L	ACTIVE																																													
13	15 Liter	FD5FEC5A		<p><b>LOG TRANSAKSI</b></p> <table border="1"> <thead> <tr> <th>Jam</th> <th>Token</th> <th>Vol</th> <th>Status</th> </tr> </thead> <tbody> <tr><td>09:38</td><td>FD5FEC5A</td><td>15L</td><td>ACTIVE</td></tr> <tr><td>09:36</td><td>2F397914</td><td>15L</td><td>ACTIVE</td></tr> <tr><td>09:35</td><td>C82AB327</td><td>15L</td><td>ACTIVE</td></tr> <tr><td>09:26</td><td>57379051</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:25</td><td>C337FD8</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:24</td><td>BC9C76D6</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:15</td><td>84DD7195</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>09:12</td><td>5E645456</td><td>5L</td><td>ACTIVE</td></tr> <tr><td>08:04</td><td>3B875ED6</td><td>2L</td><td>ACTIVE</td></tr> <tr><td>08:00</td><td>35FD2F8E</td><td>2L</td><td>ACTIVE</td></tr> </tbody> </table>	Jam	Token	Vol	Status	09:38	FD5FEC5A	15L	ACTIVE	09:36	2F397914	15L	ACTIVE	09:35	C82AB327	15L	ACTIVE	09:26	57379051	5L	ACTIVE	09:25	C337FD8	5L	ACTIVE	09:24	BC9C76D6	5L	ACTIVE	09:15	84DD7195	5L	ACTIVE	09:12	5E645456	5L	ACTIVE	08:04	3B875ED6	2L	ACTIVE	08:00	35FD2F8E	2L	ACTIVE
Jam	Token	Vol	Status																																													
09:38	FD5FEC5A	15L	ACTIVE																																													
09:36	2F397914	15L	ACTIVE																																													
09:35	C82AB327	15L	ACTIVE																																													
09:26	57379051	5L	ACTIVE																																													
09:25	C337FD8	5L	ACTIVE																																													
09:24	BC9C76D6	5L	ACTIVE																																													
09:15	84DD7195	5L	ACTIVE																																													
09:12	5E645456	5L	ACTIVE																																													
08:04	3B875ED6	2L	ACTIVE																																													
08:00	35FD2F8E	2L	ACTIVE																																													

No	Volume	Kode	QRCode	Tampilan Dashboard
14	15 Liter	31BA10DD		
15	15 Liter	A7E1B38E		



Gambar 5. 1 Perubahan Status Semua Token Volume 2 Liter



Gambar 5. 2 Perubahan Status Semua Token Volume 5 Liter



Gambar 5. 3 Perubahan Status Semua Token Volume 15 Liter





Pengujian menunjukkan bahwa setiap token yang berhasil digunakan berubah status dari *active* menjadi *used* secara otomatis yang tampil pada tampilan web setelah proses pengisian selesai.

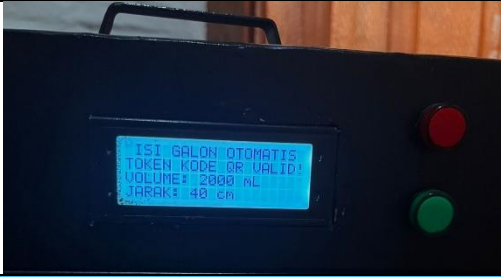
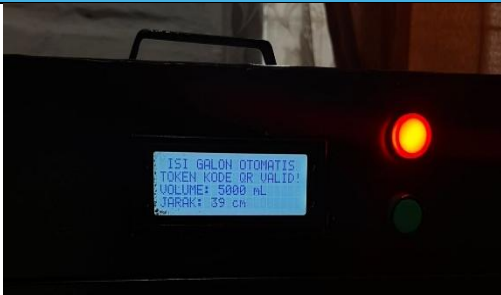




#### 5.1.4 Pengujian Validasi Token QR Code






Pengujian dilakukan untuk memastikan bahwa sistem mampu melakukan validasi token dengan benar sesuai dengan status yang tersimpan pada database. Dalam pengujian ini digunakan dua skenario utama, yaitu token yang masih valid dan aktif serta token yang sudah digunakan sebelumnya. Pada skenario pertama, token yang berstatus aktif diharapkan dapat diterima oleh sistem sehingga proses pengisian air dapat dimulai sesuai dengan volume yang telah ditentukan. Hal ini bertujuan untuk menguji apakah sistem mampu mengenali token yang sah dan

memberikan respon yang sesuai. Pada skenario kedua, token yang telah digunakan sebelumnya diuji kembali untuk memastikan bahwa sistem dapat menolak token tersebut dan tidak mengizinkan proses pengisian dilakukan ulang.



Tabel 5. 2 Pengujian Validasi Token QR Volume 2L, 5L, dan 15L Sebelum Digunakan






Pengujian Validasi Token QR Sebelum Digunakan	
Volume : 2L	
D820B1D1	
BF093157	
774FEE62	
35FD2F8E	




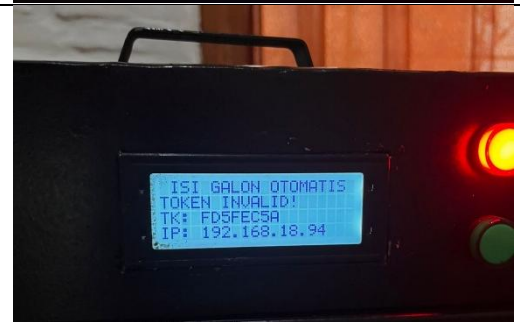

3B8753D6	
<b>Volume : 5 L</b>	
5E645456	
84DD7195	
BC9C76D6	
C337FD8	
57379051	

Volume :15 L	
C82AB327	
2F397914	
FD5FEC5A	
31BA10DD	
A7E1B38E	

Tabel 5. 3 Pengujian Validasi Token QR Volume 2L, 5L, dan 15L Setelah Digunakan

Pengujian Validasi Token QR Setelah Digunakan	
Volume : 2L	
D820B1D1	 The image shows a close-up of the LCD screen on a black automatic water dispenser. The screen displays the following text: "ISI GALON OTOMATIS", "TOKEN INVALID!", "TK: D820B1D1", and "IP: 192.168.18.94".
BF093157	 The image shows a close-up of the LCD screen on a black automatic water dispenser. The screen displays the following text: "ISI GALON OTOMATIS", "TOKEN INVALID!", "TK: BF093157", and "IP: 192.168.18.94". A red indicator light is visible on the right side of the machine.
774FEE62	 The image shows a close-up of the LCD screen on a black automatic water dispenser. The screen displays the following text: "ISI GALON OTOMATIS", "TOKEN INVALID!", "TK: 774FEE62", and "IP: 192.168.18.94".
35FD2F8E	 The image shows a close-up of the LCD screen on a black automatic water dispenser. The screen displays the following text: "ISI GALON OTOMATIS", "TOKEN INVALID!", "TK: 35FD2F8E", and "IP: 192.168.18.94". A red indicator light is visible on the right side of the machine.

3B8753D6	 <p>ISI GALON OTOMATIS TOKEN INVALID! TK: 3B8753D6 SISTEM SIAP!</p>
<b>Volume : 5 L</b>	
5E645456	 <p>ISI GALON OTOMATIS TOKEN INVALID! TK: 5E645456 IP: 192.168.18.94</p>
84DD7195	 <p>ISI GALON OTOMATIS TOKEN INVALID! TK: 84DD7195 IP: 192.168.18.94</p>
BC9C76D6	 <p>ISI GALON OTOMATIS TOKEN INVALID! TK: BC9C76D6 IP: 192.168.18.94</p>
C337FD8	 <p>ISI GALON OTOMATIS TOKEN INVALID! TK: C337FD8 IP: 192.168.18.94</p>

57379051	 <p>ISI GALON OTOMATIS TOKEN INVALID! TK: 57379051 IP: 192.168.18.94</p>
<b>Volume :15 L</b>	
C82AB327	 <p>ISI GALON OTOMATIS TOKEN INVALID! TK: C82AB327 IP: 192.168.18.94</p>
2F397914	 <p>ISI GALON OTOMATIS TOKEN INVALID! TK: 2F397914 IP: 192.168.18.94</p>
FD5FEC5A	 <p>ISI GALON OTOMATIS TOKEN INVALID! TK: FD5FEC5A IP: 192.168.18.94</p>
31BA10DD	 <p>ISI GALON OTOMATIS TOKEN INVALID! TK: 31BA10DD IP: 192.168.18.94</p>



Berdasarkan hasil pengujian validasi token yang telah dilakukan, sistem menunjukkan kemampuan dalam membaca dan memverifikasi status token yang tersimpan pada database secara real-time. Token yang berstatus *active* berhasil diproses oleh sistem, yang selanjutnya memicu alat untuk menjalankan proses pengisian air sesuai volume yang telah ditentukan pada token tersebut. Adapun token yang telah digunakan sebelumnya dan berstatus *used* secara otomatis ditolak oleh sistem tanpa memproses permintaan pengisian lebih lanjut. Hasil ini menunjukkan bahwa mekanisme validasi status token telah berjalan sesuai dengan perancangan sistem, yakni menerapkan prinsip *one-time use* pada setiap token yang diterbitkan. Mekanisme tersebut berfungsi sebagai lapisan pengamanan transaksi guna mencegah terjadinya penggunaan ulang token yang tidak sah, sehingga integritas setiap transaksi pengisian air dapat terjamin.

### 5.1.5 Pengujian Akurasi Kalibrasi dan Volume Pengisian

Pengujian akurasi kalibrasi bertujuan untuk menentukan nilai faktor kalibrasi (K) yang paling optimal pada sensor aliran YF-S201 sehingga volume cairan yang dikeluarkan oleh sistem sesuai dengan volume target yang telah ditetapkan.

Pengujian dilakukan pada beberapa variasi volume, yaitu 2.000 mL, 5.000 mL, dan 15.000 mL. Proses kalibrasi diperlukan karena nilai default faktor kalibrasi yang tercantum pada datasheet sensor tidak selalu memberikan hasil yang akurat pada setiap kondisi instalasi, oleh karena itu diperlukan pengujian secara empiris untuk memperoleh nilai K yang benar-benar sesuai dengan kondisi nyata sistem yang dirancang.

Sensor YF-S201 bekerja berdasarkan prinsip efek Hall, di mana putaran roda baling-baling (impeller) di dalam badan sensor menghasilkan sinyal pulsa listrik yang frekuensinya sebanding dengan laju aliran cairan yang melewatinya.

Mikrokontroler kemudian mencacah jumlah pulsa tersebut secara akumulatif untuk menghitung volume cairan yang telah mengalir. Berdasarkan datasheet resmi YF-S201, hubungan antara frekuensi pulsa dan laju aliran dinyatakan dalam persamaan berikut:

$$F = K \times Q$$

di mana F adalah frekuensi pulsa (Hz), K adalah faktor kalibrasi sensor, dan Q adalah laju aliran cairan (L/menit). Nilai default faktor kalibrasi dari pabrik adalah  $K = 7,5$ , yang berarti setiap 1 liter cairan secara teori menghasilkan 450 pulsa ( $7,5 \times 60$  detik). Nilai inilah yang menjadi titik awal dalam proses pengujian komparatif beberapa nilai K untuk menemukan faktor kalibrasi yang paling sesuai dengan kondisi instalasi sistem ini.

Agar sistem dapat menghentikan aliran tepat saat volume target tercapai, mikrokontroler perlu mengetahui berapa jumlah pulsa yang harus dicacah hingga volume target terpenuhi. Nilai ini disebut pulsa berhenti, yang dihitung berdasarkan konversi volume target dari satuan mililiter ke liter kemudian dikalikan dengan jumlah pulsa per liter sesuai faktor kalibrasi yang digunakan. Persamaan pulsa berhenti yang diterapkan dalam sistem adalah sebagai berikut:

$$\text{Pulsa Berhenti} = \frac{V \text{ Target}}{1000} \times (K \times 60)$$

di mana V Target adalah volume target pengisian dalam satuan mililiter, angka 1000 merupakan faktor konversi dari mililiter ke liter, K adalah faktor kalibrasi, dan angka 60 merupakan faktor konversi dari menit ke detik. Sebagai contoh, untuk  $K = 7,4$  dengan target volume 2.000 mL, nilai pulsa berhenti yang dihitung adalah  $(2.000 / 1000) \times (7,4 \times 60) = 2 \times 444 = 888$  pulsa. Artinya, sistem akan memerintahkan katup solenoida untuk menutup dan menghentikan aliran air secara otomatis setelah mikrokontroler berhasil mencacah sebanyak 888 pulsa dari sensor. Untuk mengevaluasi tingkat akurasi sistem pada setiap nilai K yang diuji, dilakukan perbandingan antara volume target dengan volume aktual yang diukur secara manual menggunakan wadah ukur terstandar. Selisih antara kedua nilai tersebut kemudian dinyatakan dalam bentuk persentase error guna memudahkan

perbandingan secara kuantitatif. Persentase error dihitung menggunakan persamaan berikut:

$$\% \text{Error} = \frac{(|V \text{ Aktual} - V \text{ Target}|)}{V \text{ Target}} \times 100\%$$

di mana  $V_{\text{aktual}}$  adalah volume aktual yang terukur pada wadah ukur (mL) dan  $V_{\text{target}}$  adalah volume target yang ditetapkan pada sistem (mL). Tanda nilai mutlak digunakan agar hasil error selalu bernilai positif terlepas dari arah selisihnya, baik kelebihan (*over*) maupun kekurangan (*under*).

Tabel 5. 4 Pengujian Sensor Flowmeter

Nilai K	(K x 60)	Target (ml)	Aktual (ml)	Selisih (ml)	Error (%)	Pulsa (Target / 1000) x (k x 60)
8.1	486	2.000	2.250	+250	12.50%	$(2000/1000) \times 486 = 972$
		5.000	5500	+500	10.00%	$(5000/1000) \times 486 = 2.430$
		15.000	16500	+1500	10.00%	$(15000/1000) \times 486 = 7.290$
7.7	462	2.000	2.150	+150	4.50%	$(2000/1000) \times 462 = 924$
		5.000	5.250	+250	7.50%	$(5000/1000) \times 462 = 2.310$
		15.000	15.650	+650	5.00%	$(15000/1000) \times 462 = 6.930$
7.5	450	2.000	2.050	+50	2.50%	$(2000/1000) \times 450 = 900$
		5.000	5.100	+100	2.00%	$(5000/1000) \times 450 = 2.250$
		15.000	15.200	+200	1.33%	$(15000/1000) \times 450 = 6.750$

Nilai K	(K x 60)	Target (ml)	Aktual (ml)	Selisih (ml)	Error (%)	Pulsa (Target / 1000) x (k x 60)
7.4	444	2.000	1950	-50	2.50%	$(2000/1000) \times 444 = 888$
		5.000	5.000	0	$\pm 0.00\%$	$(5000/1000) \times 444 = 2.220$
		15.000	15.000	0	$\pm 0.00\%$	$(15000/1000) \times 444 = 6.660$
7.3	438	2.000	1.900	-100	5.00%	$(2000/1000) \times 438 = 876$
		5.000	4.800	-200	4.00%	$(5000/1000) \times 438 = 2.190$
		15.000	14.800	-200	1.33%	$(15000/1000) \times 438 = 6.570$

Berdasarkan hasil pengujian yang disajikan pada Tabel 5.3, dilakukan pengujian terhadap lima nilai faktor kalibrasi yaitu K = 8,1, K = 7,7, K = 7,5, K = 7,4, dan K = 7,3 pada tiga variasi volume target untuk menentukan nilai K yang paling optimal. Pada pengujian dengan nilai K = 8,1 dan K = 7,7, sistem secara konsisten menghasilkan volume aktual yang melebihi target (over). Pada K = 8,1, error yang dihasilkan sangat besar yakni mencapai 12,50% pada target 2.000 mL dan 10,00% pada target 5.000 mL serta 15.000 mL. Pada K = 7,7, terjadi penurunan error yang cukup signifikan menjadi 4,50%, 7,50%, dan 5.00%.

Pengujian pada K = 7,5 yang merupakan nilai default pabrikan menunjukkan hasil yang lebih baik dengan error sebesar 2,50%, 2.00%, dan 1,33% untuk target 2.000 mL, 5.000 mL, dan 15.000 mL. Pada K = 7,4, sistem menghasilkan volume aktual yang identik dengan volume target 5.000 mL dan 15.000mL variasi pengujian dengan selisih 0 mL dan persentase error  $\pm 0,00\%$ , namun untuk volume 2.000mL terjadi error sebanyak 2.50%.

### 5.1.6 Pengujian Konsistensi Pengisian dengan Faktor K 7.4

Tabel 5. 5 Tabel Pengujian Konsistensi Pengisian 2 Liter

No	Volume Target (ml)	Volume Terukur	Selisih	Error(%)
1	2000	1950	-50	-2.50%
2	2000	1950	-50	-2.50%
3	2000	1950	-50	-2.50%
4	2000	1950	-50	-2.50%
5	2000	1950	-50	-2.50%
6	2000	1950	-50	-2.50%
7	2000	1950	-50	-2.50%
8	2000	1950	-50	-2.50%
9	2000	1950	-50	-2.50%
10	2000	1950	-50	-2.50%

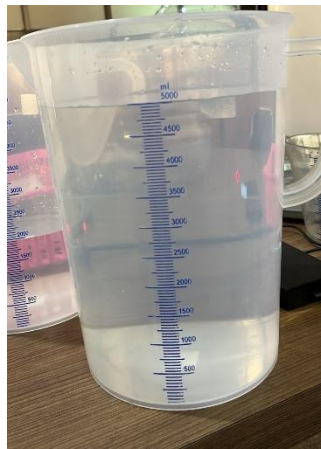


Gambar 5. 4 Dokumentasi Pengujian Pengisian 2 Liter

Tabel 5. 6 Tabel Pengujian Konsistensi Pengisian 5 Liter

No	Volume Target (ml)	Volume Terukur	Selisih	Error(%)
1	5000	5000	0	0,00%
2	5000	5000	0	0,00%
3	5000	5000	0	0,00%
4	5000	5000	0	0,00%

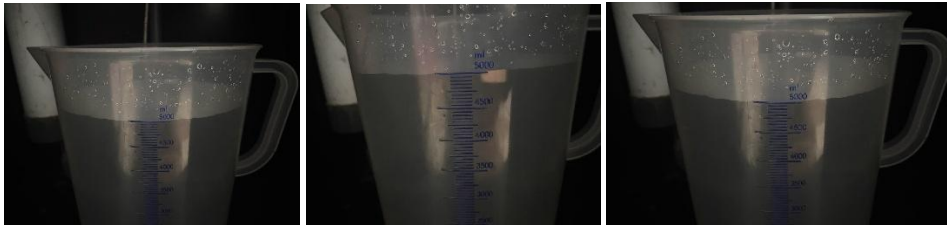
5	5000	5000	0	0,00%
6	5000	5000	0	0,00%
7	5000	5000	0	0,00%
8	5000	5000	0	0,00%
9	5000	5000	0	0,00%
10	5000	5000	0	0,00%



*Gambar 5. 5* Dokumentasi Pengujian Pengisian 5 Liter

Tabel 5. 7 Tabel Pengujian Konsistensi Pengisian 15 Liter

No	Volume Target (ml)	Volume Terukur	Selisih	Error(%)
1	15000	15000	0	0.00%
2	15000	15000	0	0.00%
3	15000	15000	0	0.00%
4	15000	15000	0	0.00%
5	15000	15000	0	0.00%
6	15000	15000	0	0.00%
7	15000	15000	0	0.00%
8	15000	15000	0	0.00%
9	15000	15000	0	0.00%
10	15000	15000	0	0.00%



Gambar 5. 6 Dokumentasi Pengujian Pengisian 10 Liter

Pengujian dilakukan menggunakan nilai faktor kalibrasi  $K = 7,4$  yang telah ditetapkan sebagai nilai optimal berdasarkan hasil kalibrasi sebelumnya. Pengujian ini dilaksanakan sebanyak 10 kali ulangan untuk masing-masing varian volume, yaitu 2.000 ml, 5.000 ml, dan 15.000 ml, dengan tujuan mengevaluasi konsistensi dan akurasi sistem secara menyeluruh.




Pada pengujian volume 2.000 ml, seluruh 10 kali pengukuran tetap menghasilkan volume terukur sebesar 1.950 ml dengan selisih tetap  $-50$  ml dan persentase error sebesar  $-2,50\%$ . Pada pengujian volume 5.000 ml, seluruh 10 kali pengukuran juga menghasilkan volume terukur yang tepat sebesar 5.000 ml dengan selisih 0 ml dan persentase error  $\pm 0,00\%$ . Hasil yang sama juga terjadi pada pengujian volume 15.000 ml, seluruh 10 kali pengukuran juga menghasilkan volume terukur yang tepat sebesar 15.000 ml dengan selisih 0 ml dan persentase error  $\pm 0,00\%$ . Konsistensi hasil pada volume besar ini membuktikan bahwa sistem mampu mempertahankan akurasi tinggi meskipun durasi pengisian lebih panjang dan jumlah pulsa yang dicacah jauh lebih besar.

Secara keseluruhan, hasil pengujian dengan  $K = 7,4$  menunjukkan bahwa sistem pengisian air galon otomatis ini memiliki tingkat akurasi dan konsistensi yang baik. Persentase error rata-rata untuk volume 5.000 ml dan 15.000 ml mencapai  $\pm 0,00\%$ , sementara volume 2.000 ml menghasilkan error sistematis sebesar  $-2,50\%$  yang masih dalam batas toleransi. Dengan demikian, nilai  $K = 7,4$  ditetapkan sebagai faktor kalibrasi final yang digunakan dalam sistem.

### 5.1.7 Pengujian Sensor Ultrasonik

Pengujian dilakukan untuk memastikan bahwa sistem hanya akan memulai proses pengisian air apabila galon berada pada posisi yang tepat sesuai dengan jarak yang telah ditentukan.

Tabel 5. 8 Pengujian Sensor Ultrasonik

Pengujian Ultrasonik	
Galon Tekdeteksi Namun Tidak Sesuai Jarak (>3cm)	
Galon Terdeteksi Sesuai Jarak (1-3cm)	
Galon Tekdeteksi Namun Tidak Sesuai Jarak (30 Detik)	

Berdasarkan hasil pengujian sensor ultrasonik yang ditunjukkan pada tabel 5.4 Pengujian sensor ultrasonik, sistem mampu membedakan kondisi keberadaan galon dengan cukup baik melalui tiga skenario utama. Pada kondisi pertama, galon terdeteksi oleh sensor namun berada pada posisi yang tidak sesuai dengan jarak yang telah ditentukan. Pada kondisi ini, sistem masih mengenali keberadaan objek, tetapi tidak mengizinkan proses pengisian untuk dimulai. Informasi jarak yang ditampilkan pada LCD menunjukkan bahwa sistem tetap melakukan pembacaan secara real-time, namun tetap menerapkan logika pembatasan agar pengisian hanya terjadi pada kondisi yang tepat. Hal ini menunjukkan bahwa sistem tidak hanya mendeteksi keberadaan objek, tetapi juga mempertimbangkan posisi yang akurat sebagai syarat pengisian.

Pada kondisi kedua, galon terdeteksi berada pada posisi yang sesuai dengan jarak yang telah ditentukan. Sistem menampilkan status “OK” pada LCD dan siap melanjutkan proses pengisian. Hal ini menunjukkan bahwa sensor ultrasonik mampu bekerja secara akurat dalam menentukan posisi galon dan menjadi parameter utama dalam pengambilan keputusan sistem.

Pada kondisi ketiga, sensor ultrasonik tetap melakukan pendeteksian terhadap galon, sistem akan menunggu keberadaan galon hingga 30 detik, jika sensor ultrasonik tidak mendeteksi keberadaan galon dalam waktu yang telah ditentukan oleh sistem, yakni 30 detik hingga waktu habis, maka sistem menampilkan informasi “TIMEOUT dan tidak ada galon” pada LCD dan tidak mengaktifkan proses pengisian. Hal ini menunjukkan bahwa sistem berhasil mencegah pengoperasian pompa ketika tidak terdapat objek di area pengisian, sehingga dapat menghindari pemborosan air dan meningkatkan keamanan sistem.

Ketiga kondisi pengujian tersebut membuktikan bahwa sensor ultrasonik HC-SR04 tidak hanya berfungsi sebagai pendeteksi keberadaan objek secara sederhana, melainkan juga sebagai mekanisme keamanan berlapis yang memastikan proses pengisian hanya terjadi dalam kondisi operasional yang benar-benar tepat. Sistem tidak akan mengaktifkan pompa hanya karena ada objek yang terdeteksi, melainkan mensyaratkan objek tersebut berada pada rentang jarak 1 hingga 3 sentimeter secara stabil selama minimal 2 detik berturut-turut sebelum perintah pengisian diberikan kepada relay.

```

if (millis() - waitStart > 30000) {
    timeout = true;
    break;
}

```

Gambar 5. 7 Potongan Kode Program Mekanisme Timeout Sensor Ultrasonik

Mekanisme waktu tunggu selama 30 detik yang diterapkan pada kondisi ketiga merupakan implementasi dari variabel waitStart pada program ESP32 yang mulai berjalan sejak token QR Code dinyatakan valid. Selama periode tersebut, sensor membaca jarak setiap 200 milidetik secara terus-menerus. Apabila dalam waktu 30 detik tidak ditemukan galon pada posisi yang sesuai, variabel timeout bernilai true dan sistem menghentikan proses.

## **BAB VI**

### **PENUTUP**

#### **6.1 Kesimpulan**

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Sistem pengisian galon otomatis menggunakan sensor QRCode GM-67, sensor flowmeter YF-S201, dan website berbasis ESP32 berhasil dirancang dan direalisasikan sesuai tujuan penelitian. Sistem mampu bekerja secara otomatis mulai dari proses identifikasi pengguna, pendeteksian galon, hingga proses pengisian air berdasarkan volume yang telah ditentukan.
2. Sensor QRCode GM-67 mampu membaca data QR Code dengan baik sebagai media identifikasi pelanggan dan penentuan kapasitas pengisian galon. QR Code yang terdaftar pada database dapat dikenali sistem secara cepat, sedangkan QR Code yang sudah digunakan atau tidak valid akan ditolak sehingga meningkatkan keamanan transaksi.
3. Sensor flowmeter YF-S201 dapat bekerja dengan baik dalam mengukur debit dan volume air yang mengalir selama proses pengisian. Berdasarkan hasil pengujian kalibrasi yang dilakukan secara komparatif terhadap lima nilai faktor kalibrasi ( $K = 8,1$ ,  $K = 7,7$ ,  $K = 7,5$ ,  $K = 7,4$ , dan  $K = 7,3$ ) pada tiga variasi volume target yaitu 2.000 mL, 5.000 mL, dan 15.000 mL, diperoleh nilai faktor kalibrasi optimal sebesar  $K = 7,4$  dengan nilai  $K \times 60 = 444$  pulsa. Nilai tersebut terbukti mampu menghasilkan volume aktual yang identik dengan volume target pada seluruh variasi pengujian dengan persentase error sebesar 0,00%, berbeda dengan nilai default pabrikan  $K = 7,5$  yang masih menghasilkan error hingga 2,50%, sehingga sistem dapat menghentikan aliran air secara otomatis dengan tepat saat volume target tercapai.
4. Pengujian sebanyak 10 kali ulangan untuk masing-masing varian volume dengan menggunakan nilai faktor kalibrasi  $K = 7,4$ , dapat disimpulkan

bahwa sistem pengisian air galon otomatis ini bekerja secara konsisten dan akurat. Pada volume 5.000 ml dan 15.000 ml, seluruh pengujian menghasilkan volume terukur yang tepat sesuai target dengan persentase error  $\pm 0,00\%$ , menunjukkan bahwa sistem mampu mempertahankan akurasi sempurna pada volume menengah hingga besar. Sementara itu, pada volume 2.000 ml terdapat deviasi sistematis sebesar  $-2,50\%$  dengan hasil terukur 1.950 ml, yang disebabkan oleh pengaruh inersia fluida dan waktu respons penutupan pompa pada volume kecil. Secara keseluruhan, nilai  $K = 7,4$  terbukti merupakan faktor kalibrasi yang paling optimal untuk sistem ini, menghasilkan tingkat akurasi yang tinggi dan konsistensi yang baik di seluruh varian volume yang diuji.

5. Website monitoring yang dirancang mampu menampilkan data transaksi pengisian galon, seperti kode token, volume pengisian, status transaksi, dan waktu penggunaan. Data yang tersimpan pada database dapat ditampilkan secara akurat sehingga memudahkan admin dalam melakukan pengawasan dan pencatatan operasional.
6. Sistem berhasil menerapkan mekanisme satu kali penggunaan token QR Code, dimana token dengan status aktif dapat diproses untuk pengisian, sedangkan token yang sudah digunakan akan otomatis ditolak. Fitur ini mampu mencegah penyalahgunaan dan meningkatkan efisiensi pelayanan.

## **6.2 Saran**

1. Website monitoring dapat dikembangkan menjadi aplikasi mobile berbasis Android/iOS agar admin dapat memantau data transaksi, penggunaan air, dan status alat dengan lebih mudah melalui smartphone.
2. Desain mekanik alat dapat disempurnakan agar lebih kokoh, tahan air, dan sesuai digunakan pada lingkungan depot air minum dalam jangka panjang.
3. Penelitian selanjutnya dapat mengembangkan sistem dengan kapasitas pengisian yang lebih beragam dan jumlah nozzle lebih dari satu, sehingga dapat melayani beberapa pelanggan secara bersamaan.
4. Sistem dapat dikembangkan dengan mengintegrasikan payment gateway seperti Midtrans, Xendit, atau layanan sejenis pada halaman website

pembelian token, sehingga pengguna dapat melakukan pembayaran secara digital menggunakan berbagai metode seperti transfer bank, dompet digital, maupun QRIS sebelum token QR diterbitkan. Integrasi ini akan menjadikan sistem lebih lengkap sebagai solusi depot air minum mandiri yang sepenuhnya cashless dan tidak memerlukan interaksi langsung dengan operator dalam proses transaksi.

### DAFTAR PUSTAKA

- [1] ARFANDI, Armin; SUPIT, Yonal. Prototipe Sistem Otomasi Pada Pengisian Depot Air Minum Isi Ulang Berbasis Arduino Uno. *Simtek: jurnal sistem informasi dan teknik komputer*, 2019, 4.1: 91-99.
- [2] VINCHI, REYNO DA. *Pemanfaatan Sistem IoT untuk Pemantauan Debit Air pada Pamsimas Dukuh Jetak*. 2025. PhD Thesis. Universitas Duta Bangsa Surakarta.
- [3] Ardiliansyah, A. R., Puspitasari, M. D., & Arifianto, T. (2021). Rancang bangun prototipe pompa otomatis dengan fitur monitoring berbasis IoT menggunakan sensor flowmeter dan ultrasonik. *Explore IT! Jurnal Keilmuan dan Aplikasi Teknik Informatika*, 13(2), 59–67.
- [4] Munandar, A., Veronika, N. D. M., Abdullah, D., & Sahputra, E. (2023). Miniature Design of Liquid Filling Machine Automatically Using ESP32 Based IoT (Internet of Things). *Jurnal Komputer, Informasi dan Teknologi (KOMITEK)*, 3(1), 69–78.
- Kurniawan, E., & Fadillah, M. (2020). Prototype Mesin Isi Ulang Air Minum Otomatis Berbasis Mikrokontroler. *Jurnal Rekayasa Elektronika*, 7(2), 14-21.
- [5] Bayu, R. B. S., Astutik, R. P., & Irawan, D. (2021). Rancang bangun smarthome berbasis QR code dengan mikrokontroller module ESP32. *JASEE Journal of Application and Science on Electrical Engineering*, 2(01), 47–60. <https://doi.org/10.31328/jasee.v2i01.60>
- [6] Muamaroh, N., & Christanto, F. W. (2024). Pengukur Penggunaan Air Otomatis Menggunakan Water Sensor Flowmeter YF-S201 dan NodeMCU ESP8266 Berbasis IoT. *JIKO (Jurnal Informatika dan Komputer)*, Universitas Teknologi Digital Indonesia.
- [7] Nadia, N., et al. (2024). Manufacture of Cooking Oil Content Drain Control Device in Oily Foods Using the ESP32-Based and Fuzzy Logic Method. *Transmisi: Jurnal Ilmiah Teknik Elektro*, Universitas Diponegoro, Vol. 26, No. 2.

- [8] NUR, W. W., ARDIAN, A. D. S., & MUHAMMAD, S. A. P. (2024). Implementasi Sistem Keamanan IoT Berbasis QR Code pada Loker untuk Peningkatan Keamanan dan Aksesibilitas. *JURNAL ELEKTRONIKA DAN OTOMASI INDUSTRI Учредители: Politeknik Negeri Malang*, 11(3), 788-796.
- [9] Rashad, A. A., Girawan, B. A., Ariawan, R., Sodikin, J., & Hastuti, H. D. (2023). Uji fungsi dan kalibrasi sensor water flow YF-S201 berbasis Arduino Uno pada mesin penjernih air sungai. *Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, 3(2), 1-9.
- [10] Ikhsan, D. Y. N., Rizkiyanto, M. D., Putra, A. R., & Fitrianti, F. A. (2026). Perancangan dan Implementasi Sistem Pengukuran Ketinggian dan Volume Air pada Tandon Berbasis Sensor Jarak dan IoT dengan Monitoring melalui Aplikasi Mobile. *Journal of Mechanical Engineering and Applied Technology*, 4(1), 7-14.
- [11] Wibowo, W. A., & Setyadjit, K. (2022, October). Rancang Bangun Sistem Kontrol Starter Berbasis IoT (Internet Of Things). In *Senakama: Prosiding Seminar Nasional Karya Ilmiah Mahasiswa* (Vol. 1, No. 1, pp. 551-558).
- [12] Texas Instruments. (2022). *LM2596 SIMPLE SWITCHER® power converter 150-kHz 3-A step-down voltage regulator* (Rev. 1).
- [13] SparkFun Electronics. (2024). Bi-directional logic level converter hookup guide..
- [14] Texas Instruments. (2024). TXS0108E 8-bit bi-directional, level-shifting, voltage translator (Rev. K)..
- [15] ESP Easy Documentation. (2023). Level converter — ESP Easy plugin documentation. Read the Docs..
- [16] Harmie, H. A., Putra, R. D. D., Josi, A., & Afriansyah, R. (2024). Perancangan aplikasi smart fence menggunakan Visual Studio Code berbasis IoT. *Jurnal Inovasi Teknologi Terapan*, 2(2).
- [17] Andrian, W., & Prasetya, D. K. (2022). Pengembangan manajemen keamanan informasi database dan aplikasi dengan optimasi keamanan website. *Jurnal Sistem Informasi dan Teknologi*, 2(2).

## LAMPIRAN

### Lampiran Program Arduino IDE

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <WiFiClient.h>
#include <ArduinoJson.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Preferences.h>
#include <WebServer.h>
// =====
// 1. DEFINISI PIN & KONFIGURASI
// =====
#define RXD2 16
#define TXD2 17
#define PIN_SDA 21
#define PIN_SCL 22
#define PIN_RELAY_HIJAU 33
#define PIN_RELAY_MERAH 25
#define PIN_POMPA 26
#define PIN_BUZZER 27
#define PIN_FLOW 4
#define PIN_TRIG 32
#define PIN_ECHO 35
String server_domain = "isigalonku.alwaysdata.net";
#define RELAY_ON LOW
#define RELAY_OFF HIGH

const int TARGET_JARAK = 2;
const int TOLERANSI = 1;
const int WAKTU_STABIL = 2000;
float calibrationFactor = 7.41;
volatile unsigned long pulseCount = 0;
volatile bool isProcessing = false;
volatile bool setupMode = false;
portMUX_TYPE mux = portMUX_INITIALIZER_UNLOCKED;
QueueHandle_t queueTransaction;
SemaphoreHandle_t mutexUI;
LiquidCrystal_I2C lcd(0x27, 20, 4);
Preferences preferences;
WebServer server(80);
struct Transaction {
    int volume_ml;
    String token;
};
// =====
// 2. INTERRUPT SERVICE ROUTINE (ISR)
// =====
void IRAM_ATTR pulseCounter() {
```

```

    portENTER_CRITICAL_ISR(&mux);
    pulseCount++;
    portEXIT_CRITICAL_ISR(&mux);
}

// =====
// 3. FUNGSI LCD — SEMUA AMAN DENGAN MUTEX
// =====

/**
 * [PERBAIKAN AKTIF] Reinisialisasi LCD + I2C Recovery yang aman.
 * Dipanggil secara strategis setelah Relay Pompa ON/OFF untuk
 * menanggulangi I2C crash akibat lonjakan EMI (Inductive Kickback).
 */
void lcdSafeInit() {
    if (xSemaphoreTake(mutexUI, pdMS_TO_TICKS(500))) {
        Wire.end();
        vTaskDelay(pdMS_TO_TICKS(50));
        Wire.begin(PIN_SDA, PIN_SCL);
        vTaskDelay(pdMS_TO_TICKS(50));

        lcd.init();
        lcd.backlight();
        xSemaphoreGive(mutexUI);
    }
}

void lcdPrintRow(int row, String text) {
    while (text.length() < 20) text += ' ';
    if (text.length() > 20) text = text.substring(0, 20);
    if (xSemaphoreTake(mutexUI, pdMS_TO_TICKS(200))) {
        lcd.setCursor(0, row);
        lcd.print(text);
        xSemaphoreGive(mutexUI);
    }
}

void lcdClear() {
    if (xSemaphoreTake(mutexUI, pdMS_TO_TICKS(200))) {
        lcd.clear();
        xSemaphoreGive(mutexUI);
    }
}

void lcdPrintAll(String r0, String r1, String r2, String r3) {
    auto pad = [](String s) -> String {
        while (s.length() < 20) s += ' ';
        return s.substring(0, 20);
    };
    if (xSemaphoreTake(mutexUI, pdMS_TO_TICKS(200))) {
        lcd.setCursor(0, 0); lcd.print(pad(r0));
        lcd.setCursor(0, 1); lcd.print(pad(r1));
        lcd.setCursor(0, 2); lcd.print(pad(r2));
        lcd.setCursor(0, 3); lcd.print(pad(r3));
        xSemaphoreGive(mutexUI);
    }
}

```

```

}
// =====
// 4. FUNGSI BANTUAN (HELPER)
// =====
void beep(int times, int duration_ms = 100) {
    for (int i = 0; i < times; i++) {
        digitalWrite(PIN_BUZZER, HIGH);
        vTaskDelay(pdMS_TO_TICKS(duration_ms));
        digitalWrite(PIN_BUZZER, LOW);
        vTaskDelay(pdMS_TO_TICKS(duration_ms));
    }
}

int getDistance() {
    digitalWrite(PIN_TRIG, LOW);
    delayMicroseconds(2);
    digitalWrite(PIN_TRIG, HIGH);
    delayMicroseconds(10);
    digitalWrite(PIN_TRIG, LOW);
    long duration = pulseIn(PIN_ECHO, HIGH, 30000);
    if (duration == 0) return 999;
    return (int)(duration * 0.034 / 2);
}
// =====
// 5. FUNGSI KOMUNIKASI API
// =====
void updateTokenStatus(String qrCode) {
    if (WiFi.status() == WL_CONNECTED) {
        WiFiClient client;
        HTTPClient http;
        String url = "http://" + server_domain + "/api.php?token=" + qrCode + "&action=finish";
        if (http.begin(client, url)) {
            http.setUserAgent("Mozilla/5.0 (Windows NT 10.0; Win64; x64)");
            http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
            http.GET();
            http.end();
        }
    }
}

int checkDatabase(String qrCode) {
    if (WiFi.status() == WL_CONNECTED) {
        WiFiClient client;
        HTTPClient http;
        String url = "http://" + server_domain + "/api.php?token=" + qrCode;
        if (http.begin(client, url)) {
            http.setUserAgent("Mozilla/5.0 (Windows NT 10.0; Win64; x64)");
            http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
            int httpCode = http.GET();
            if (httpCode > 0) {
                String payload = http.getString();
                DynamicJsonDocument doc(1024);
                DeserializationError error = deserializeJson(doc, payload);
                if (!error) {

```

```

        String status = doc["status"];
        if (status == "valid") {
            int vol = doc["volume"];
            http.end();
            return vol;
        }
    }
}
http.end();
}
}
return 0;
}
// =====
// 6. HALAMAN WEB SERVER (HTML LOKAL)
// =====
const char* htmlPage = R"rawliteral(
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>WiFi Setup</title>
  <style>
    body { background-color: #0d1117; color: #c9d1d9; font-family: sans-serif; text-align:
center; padding-top: 50px; }
    h2 { color: #00f2ff; }
    form { background: #161b22; display: inline-block; padding: 30px; border-radius: 10px;
border: 1px solid #30363d; }
    input { padding: 12px; margin: 10px 0; width: 250px; border-radius: 5px; border: 1px solid
#30363d; background: #0d1117; color: white; }
    input[type="submit"] { background: #238636; color: white; font-weight: bold; cursor:
pointer; border: none; }
    input[type="submit"]:hover { background: #2ea043; }
  </style>
</head>
<body>
  <h2>PENGATURAN WIFI</h2>
  <form action="/save" method="POST">
    <input type="text" name="ssid" placeholder="Nama WiFi (SSID)" required><br>
    <input type="password" name="pass" placeholder="Password WiFi"><br>
    <input type="submit" value="SIMPAN & RESTART">
  </form>
</body>
</html>
)rawliteral";
void handleRoot() {
  server.send(200, "text/html", htmlPage);
}

void handleSave() {
  String newSSID = server.arg("ssid");
  String newPass = server.arg("pass");

```

```

    preferences.begin("wifi_config", false);
    preferences.putString("ssid", newSSID);
    preferences.putString("pass", newPass);
    preferences.end();
    String htmlResp = "<!DOCTYPE html><html><body
style='background:#0d1117;color:#00f2ff;'
    "text-align:center;margin-top:100px;font-family:sans-serif;>"
    "<h2>Tersimpan!</h2><p>ESP32 sedang direstart. Silakan konek kembali ke
WiFi Anda.</p>"
    "</body></html>";
    server.send(200, "text/html", htmlResp);
    vTaskDelay(pdMS_TO_TICKS(2000));
    ESP.restart();
}
// =====
// 7. TASK 1: NETWORK & WEB SERVER (Core 0)
// =====
void taskNetwork(void *pvParameters) {
    preferences.begin("wifi_config", true);
    String savedSSID = preferences.getString("ssid", "");
    String savedPass = preferences.getString("pass", "");
    preferences.end();
    if (savedSSID != "") {
        lcdPrintRow(2, "WiFi: Connecting...");
        WiFi.begin(savedSSID.c_str(), savedPass.c_str());

        int attempts = 0;
        while (WiFi.status() != WL_CONNECTED && attempts < 20) {
            vTaskDelay(pdMS_TO_TICKS(500));
            attempts++;
        }
    }
    if (WiFi.status() != WL_CONNECTED) {
        setupMode = true;
        WiFi.mode(WIFI_AP);
        WiFi.softAP("WiFi_SETUP");
        lcdClear();
        lcdPrintAll(
            " KONEKSI WIFI GAGAL ",
            "Konek Hotspot HP ke:",
            " WiFi: WiFi_SETUP ",
            "Buka IP 192.168.4.1 "
        );
        server.on("/", handleRoot);
        server.on("/save", handleSave);
        server.begin();
        while (1) {
            server.handleClient();
            vTaskDelay(pdMS_TO_TICKS(10));
        }
    }
    lcdPrintRow(2, "WiFi: CONNECTED ");
    lcdPrintRow(3, "IP:" + WiFi.localIP().toString());
}

```

```

vTaskDelay(pdMS_TO_TICKS(2000));

String qrBuffer = "";
unsigned long lastScan = 0;
while (1) {
  while (Serial2.available()) {
    char inChar = (char)Serial2.read();
    if (isProcessing) {
      qrBuffer = "";
      continue;
    }
    if (inChar == '\r' || inChar == '\n') {
      if (qrBuffer.length() > 0) {
        qrBuffer.trim();
        String cleanToken = "";
        for (int i = 0; i < (int)qrBuffer.length(); i++) {
          char c = qrBuffer.charAt(i);
          if ((c >= '0' && c <= '9') || (c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z')) {
            if (c >= 'a' && c <= 'z') c -= 32;
            cleanToken += c;
          }
        }
        if (cleanToken.length() > 8) {
          cleanToken = cleanToken.substring(cleanToken.length() - 8);
        }
        if (millis() - lastScan > 3000) {
          lastScan = millis();
          lcdPrintRow(1, "VALIDASI TOKEN... ");
          lcdPrintRow(2, "TK: " + cleanToken);
          beep(1, 50);

          int vol = checkDatabase(cleanToken);
          if (vol > 0) {
            isProcessing = true;
            Transaction trans = {vol, cleanToken};
            xQueueSend(queueTransaction, &trans, 0);
          } else {
            lcdPrintRow(1, "TOKEN INVALID! ");
            beep(3, 100);
            vTaskDelay(pdMS_TO_TICKS(1500));
            lcdPrintRow(1, "MEMINDAI KODE QR... ");
            lcdPrintRow(2, " ");
          }
        }
        qrBuffer = "";
      } else {
        qrBuffer += inChar;
      }
    }
  }
  vTaskDelay(pdMS_TO_TICKS(10));
}
}

```

```

// =====
// 8. TASK 2: CONTROL & HARDWARE (Core 1)
// =====
void taskControl(void *pvParameters) {
    Transaction currentTrans;

    while (1) {
        if (setupMode) {
            vTaskDelay(pdMS_TO_TICKS(1000));
            continue;
        }
        digitalWrite(PIN_RELAY_MERAH, RELAY_ON);
        digitalWrite(PIN_RELAY_HIJAU, RELAY_OFF);
        digitalWrite(PIN_POMPA, RELAY_OFF);
        lcdPrintAll(
            " ISI GALON OTOMATIS ",
            "MEMINDAI KODE QR... ",
            "          ",
            "SISTEM SIAP!          "
        );
        isProcessing = false;
        if (xQueueReceive(queueTransaction, &currentTrans, portMAX_DELAY) == pdPASS) {
            lcdPrintRow(1, "TOKEN KODE QR VALID!");
            lcdPrintRow(2, "VOLUME: " + String(currentTrans.volume_ml) + " mL  ");
            lcdPrintRow(3, "MENDETEKSI GALON... ");
            beep(1, 200);
            // --- Fase 1: Deteksi keberadaan galon ---
            unsigned long stabilityStart = 0;
            bool timeout = false;
            unsigned long waitStart = millis();
            while (true) {
                int dist = getDistance();
                if (dist >= (TARGET_JARAK - TOLERANSI) && dist <= (TARGET_JARAK +
TOLERANSI)) {
                    if (stabilityStart == 0) stabilityStart = millis();
                    lcdPrintRow(3, "JARAK:" + String(dist) + "cm OK...  ");
                    if (millis() - stabilityStart > WAKTU_STABIL) break;
                } else {
                    stabilityStart = 0;
                    lcdPrintRow(3, "JARAK:" + String(dist) + " cm  ");
                }
            }
            digitalWrite(PIN_RELAY_MERAH, !digitalRead(PIN_RELAY_MERAH));
            if (millis() - waitStart > 30000) {
                timeout = true;
                break;
            }
        }
        vTaskDelay(pdMS_TO_TICKS(200));
    }
    if (timeout) {
        lcdPrintRow(2, "TIMEOUT!!!          ");
        lcdPrintRow(3, "TIDAK ADA GALON  ");
        beep(3, 200);
    }
}

```

```

vTaskDelay(pdMS_TO_TICKS(2000));
isProcessing = false;
continue;
}
// --- Fase 2: Mulai pengisian ---
digitalWrite(PIN_RELAY_MERAH, RELAY_OFF);
digitalWrite(PIN_RELAY_HIJAU, RELAY_ON);
portENTER_CRITICAL(&mux);
pulseCount = 0;
portEXIT_CRITICAL(&mux);
// NYALAKAN POMPA
digitalWrite(PIN_POMPA, RELAY_ON);

// [PERBAIKAN AKTIF]
// Berikan jeda waktu sebentar untuk membiarkan gelombang EMI (Noise) berlalu,
// lalu paksa reinisialisasi I2C dan LCD agar sembuh dari efek noise.
vTaskDelay(pdMS_TO_TICKS(250));
lcdSafeInit();
lcdClear();
lcdPrintAll(
  " ISI GALON OTOMATIS ",
  " MEMULAI PENGISIAN ",
  " JANGAN PINDAHKAN ",
  " "
);
beep(1, 500);
unsigned long lastUpdate = 0;
float totalVolumeML = 0;
bool galonCabut = false;
// --- Loop monitoring pengisian ---
while (totalVolumeML < currentTrans.volume_ml) {
  unsigned long currentPulses;
  portENTER_CRITICAL(&mux);
  currentPulses = pulseCount;
  portEXIT_CRITICAL(&mux);
  totalVolumeML = (currentPulses / (calibrationFactor * 60.0)) * 1000.0;
  if (millis() - lastUpdate > 200) {
    String statusMsg = "ISI:" + String((int)totalVolumeML)
      + "/" + String(currentTrans.volume_ml) + "mL";
    lcdPrintRow(3, statusMsg);
    lastUpdate = millis();
  }

  if (getDistance() > 20) {
    galonCabut = true;
    break;
  }
  vTaskDelay(pdMS_TO_TICKS(10));
}
// MATIKAN POMPA
digitalWrite(PIN_POMPA, RELAY_OFF);
// [PERBAIKAN AKTIF]
// Mematikan induktor juga menghasilkan flyback voltage (noise).

```

```

// Lakukan pemulihan I2C/LCD sekali lagi di sini.
vTaskDelay(pdMS_TO_TICKS(250));
lcdSafeInit();
if (galonCabut) {
  lcdPrintRow(2, " TIDAK ADA GALON! ");
  beep(5, 100);
  updateTokenStatus(currentTrans.token);
  vTaskDelay(pdMS_TO_TICKS(3000));
} else {
  String finalMsg = "ISI:" + String(currentTrans.volume_ml)
    + "/" + String(currentTrans.volume_ml) + "mL";
  lcdPrintRow(3, finalMsg);
  vTaskDelay(pdMS_TO_TICKS(1500));
  updateTokenStatus(currentTrans.token);
  lcdPrintAll(
    " ISI GALON OTOMATIS ",
    " PENGISIAN SELESAI ",
    " TERIMA KASIH ",
    " " );
  beep(2, 200);
  vTaskDelay(pdMS_TO_TICKS(3000));
}
isProcessing = false;
}
}
}
// =====
// 9. SETUP (INISIALISASI AWAL)
// =====
void setup() {
  Serial.begin(115200);
  Serial2.begin(9600, SERIAL_8N1, RXD2, TXD2);
  Wire.begin(PIN_SDA, PIN_SCL);
  lcd.init();
  lcd.backlight();
  pinMode(PIN_RELAY_HIJAU, OUTPUT);
  pinMode(PIN_RELAY_MERAH, OUTPUT);
  pinMode(PIN_POMPA, OUTPUT);
  pinMode(PIN_BUZZER, OUTPUT);
  pinMode(PIN_TRIG, OUTPUT);
  pinMode(PIN_ECHO, INPUT);
  pinMode(PIN_FLOW, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(PIN_FLOW), pulseCounter, FALLING);
  digitalWrite(PIN_RELAY_MERAH, RELAY_ON);
  digitalWrite(PIN_RELAY_HIJAU, RELAY_OFF);
  digitalWrite(PIN_POMPA, RELAY_OFF);
  digitalWrite(PIN_BUZZER, LOW);
  queueTransaction = xQueueCreate(1, sizeof(Transaction));
  mutexUI = xSemaphoreCreateMutex();
  lcdPrintRow(0, " ISI GALON OTOMATIS ");
  lcdPrintRow(1, "MUHAMMAD RIZKY AKBAR");
  lcdPrintRow(2, " 40040621650039 ");
}

```

```
lcdPrintRow(3, " YUNIARTO S.T., M.T.");  
delay(2500);  
xTaskCreatePinnedToCore(taskNetwork, "NetTask", 20000, NULL, 1, NULL, 0);  
xTaskCreatePinnedToCore(taskControl, "CtrlTask", 20000, NULL, 2, NULL, 1);  
}  
void loop() {  
  vTaskDelete(NULL);  
}
```

## Lampiran program Index.Php

```

<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>RANCANG BANGUN PENGISIAN GALON OTOMATIS MENGGUNAKAN
SENSOR QRCODE, FLOWMETER, DAN WEBSITE BERBASIS ESP 32</title>
  <style>
    body {
      background-color: #0d1117;
      color: #c9d1d9;
      font-family: 'Courier New', monospace;
      text-align: center;
      padding: 50px;
    }
    h1 { color: #00f2ff; text-shadow: 0 0 10px #00f2ff; letter-spacing: 5px; margin-bottom:
5px; }
    p.subtitle { color: #8b949e; font-size: 0.9em; font-weight: 1000; letter-spacing: 5px;
margin-bottom: 20px; }

    .card-container { display: flex; justify-content: center; gap: 20px; flex-wrap: wrap; }
    .card {
      background: #161b22;
      border: 1px solid #30363d;
      border-radius: 10px;
      padding: 20px;
      width: 220px;
      transition: 0.3s;
      box-shadow: 0 0 5px rgba(0, 242, 255, 0.1);
      position: relative;
      overflow: hidden;
    }
    .card:hover { transform: translateY(-5px); border-color: #00f2ff; box-shadow: 0 0 20px
rgba(0, 242, 255, 0.3); }

    .vol-title { font-size: 1.8em; font-weight: bold; color: #fff; margin: 10px 0; }

    /* KOMENTAR: Style harga disembunyikan */
    /* .price { font-size: 1.4em; color: #00f2ff; margin-bottom: 20px; text-shadow: 0 0 5px
rgba(0, 242, 255, 0.5); } */

    .btn {

```

```

background: #238636;
color: white;
border: none;
padding: 12px;
border-radius: 5px;
cursor: pointer;
font-weight: bold;
width: 100%;
font-family: inherit;
letter-spacing: 1px;
transition: 0.2s;
margin-top: 15px; /* Tambahan sedikit margin karena price dihilangkan */
}
.btn:hover { background: #2ea043; box-shadow: 0 0 10px #2ea043; }

.modal-overlay {
display: none;
position: fixed;
z-index: 999;
left: 0; top: 0;
width: 100%; height: 100%;
background-color: rgba(0,0,0,0.85);
backdrop-filter: blur(5px);
}

.modal-box {
background: #161b22;
margin: 10% auto;
padding: 30px;
border: 2px solid #00f2ff;
width: 350px;
border-radius: 10px;
box-shadow: 0 0 30px rgba(0, 242, 255, 0.4);
text-align: center;
animation: slideDown 0.3s ease-out;
}

@keyframes slideDown {
from { transform: translateY(-50px); opacity: 0; }
to { transform: translateY(0); opacity: 1; }
}

.modal-title { color: #fff; font-size: 1.5em; margin-bottom: 20px; border-bottom: 1px
solid #30363d; padding-bottom: 10px; }
.modal-info { font-size: 1.2em; margin: 10px 0; color: #c9d1d9; }

```

```

.highlight { color: #00f2ff; font-weight: bold; }

.modal-actions { display: flex; gap: 10px; margin-top: 30px; }
.btn-confirm { flex: 1; background: #00f2ff; color: #000; border: none; padding: 10px;
border-radius: 5px; font-weight: bold; cursor: pointer; }
.btn-confirm:hover { background: #fff; box-shadow: 0 0 15px #fff; }
.btn-cancel { flex: 1; background: transparent; color: #ff5555; border: 1px solid #ff5555;
padding: 10px; border-radius: 5px; font-weight: bold; cursor: pointer; }
.btn-cancel:hover { background: #ff5555; color: #fff; box-shadow: 0 0 15px #ff5555; }

</style>
</head>
<body>

<h1>RANCANG BANGUN PENGISIAN GALON OTOMATIS MENGGUNAKAN
SENSOR QRCODE, FLOWMETER, DAN WEBSITE BERBASIS ESP 32</h1>
<p class="subtitle">MUHAMMAD RIZKY AKBAR</p>
<p class="subtitle">40040621650039</p>

<div class="card-container">
  <div class="card">
    <div class="vol-title">2 Liter</div>
    <form action="process.php" method="POST">
      <input type="hidden" name="vol" value="2000">
      <button type="button" class="btn" onclick="askConfirm('2 Liter',
this)">BUY</button>
    </form>
  </div>

  <div class="card">
    <div class="vol-title">5 Liter</div>
    <form action="process.php" method="POST">
      <input type="hidden" name="vol" value="5000">
      <button type="button" class="btn" onclick="askConfirm('5 Liter',
this)">BUY</button>
    </form>
  </div>

  <div class="card">
    <div class="vol-title">15 Liter</div>
    <form action="process.php" method="POST">
      <input type="hidden" name="vol" value="15000">
      <button type="button" class="btn" onclick="askConfirm('15 Liter',
this)">BUY</button>
    </form>
  </div>

```

```

</div>
</div>

<div id="confirmModal" class="modal-overlay">
  <div class="modal-box">
    <div class="modal-title">APAKAH ANDA YAKIN?</div>
    <p class="modal-info">Volume: <span id="modal-vol"
class="highlight"></span></p>

    <br>
    <div class="modal-actions">
      <button onclick="closeModal()" class="btn-cancel">TIDAK</button>
      <button onclick="processTransaction()" class="btn-confirm">YA</button>
    </div>
  </div>
</div>

<script>
  let selectedForm = null;

  // KOMENTAR: Parameter 'price' dihapus dari fungsi
  // function askConfirm(volume, price, btnRef) {
  function askConfirm(volume, btnRef) {

    selectedForm = btnRef.closest('form');

    document.getElementById('modal-vol').innerText = volume;

    // KOMENTAR: Baris pengisian teks harga disembunyikan
    // document.getElementById('modal-price').innerText = price;

    document.getElementById('confirmModal').style.display = 'block';
  }

  function closeModal() {
    document.getElementById('confirmModal').style.display = 'none';
    selectedForm = null;
  }

  function processTransaction() {
    if (selectedForm) {
      selectedForm.submit();
    }
  }
}

```

```
    window.onclick = function(event) {  
      var modal = document.getElementById('confirmModal');  
      if (event.target == modal) {  
        closeModal();  
      }  
    }  
  </script>  
</body>  
</html>
```

## Lampiran Program Process.Php

```

<?php
include 'db.php';

if ($ SERVER["REQUEST METHOD"] == "POST") {
    $vol = $_POST['vol'];

    $token = strtoupper(substr(md5(time()), 0, 8));

    $waktu_sekarang = date('Y-m-d H:i:s');

    $sql = "INSERT INTO tokens (token code, volume, status, purchase date) VALUES
('$token', '$vol', 'active', '$waktu sekarang)";

    if ($conn->query($sql) !== TRUE) {
        echo "Error: " . $conn->error;
        exit();
    }
} else {
    header("Location: index.php");
    exit();
}
?>

<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Transaction Success</title>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/qrious/4.0.2/qrious.min.js"></script>
    <style>
        body { background-color: #0d1117; color: #c9d1d9; font-family: monospace; text-align:
center; padding-top: 50px; }
        .container {
            background: #161b22;
            display: inline-block;
            padding: 40px;
            border-radius: 10px;
            border: 1px solid #00f2ff;
            box-shadow: 0 0 20px rgba(0, 242, 255, 0.2);
            max-width: 400px;
        }
        h2 { color: #00f2ff; text-shadow: 0 0 5px #00f2ff; margin-top: 0;}

```

```

canvas { border: 10px solid #fff; margin: 20px 0; border-radius: 4px;}
.token-text { font-size: 1.5em; letter-spacing: 5px; color: #ffff00; margin-bottom: 5px;
font-weight: bold;}
.info-text { color: #8b949e; margin-bottom: 25px; font-size: 0.9em; }

.btn-download {
    background: #00f2ff;
    color: #000;
    border: none;
    padding: 12px 25px;
    border-radius: 5px;
    cursor: pointer;
    font-weight: bold;
    font-family: monospace;
    font-size: 1em;
    text-transform: uppercase;
    box-shadow: 0 0 10px rgba(0, 242, 255, 0.4);
    transition: 0.3s;
    margin-bottom: 20px;
    display: inline-block;
}
.btn-download:hover { background: #fff; box-shadow: 0 0 20px #fff; }

a.back-link { color: #8b949e; text-decoration: none; border-bottom: 1px dotted #8b949e;
font-size: 0.8em; }
a.back-link:hover { color: #fff; border-bottom: 1px solid #fff; }
</style>
</head>
<body>
<div class="container">
    <h2>Pemilihan Berhasil</h2>
    <p>Silahkan scan QR</p>

    <canvas id="qr"></canvas>

    <div class="token-text"><?php echo $token; ?></div>
    <div class="info-text">Volume: <?php echo $vol/1000; ?> Liters</div>

    <button onclick="downloadQR()" class="btn-download">
        Download QR Image
    </button>
    <br>

    <a href="index.php" class="back-link"><< Back to Home</a>
</div>

```

```

<script>
  var qr = new QRious({
    element: document.getElementById('qr'),
    value: '<?php echo $token; ?>',
    size: 200,
    background: 'white',
    foreground: 'black',
    level: 'H'
  });

  function downloadQR() {
    var sourceCanvas = document.getElementById("qr");
    var borderSize = 20;

    var finalCanvas = document.createElement('canvas');
    var ctx = finalCanvas.getContext('2d');

    finalCanvas.width = sourceCanvas.width + (borderSize * 2);
    finalCanvas.height = sourceCanvas.height + (borderSize * 2);

    ctx.fillStyle = "#FFFFFF";
    ctx.fillRect(0, 0, finalCanvas.width, finalCanvas.height);

    ctx.drawImage(sourceCanvas, borderSize, borderSize);

    var image = finalCanvas.toDataURL("image/png");
    var link = document.createElement('a');
    link.download = 'HYDRIX-TOKEN-<?php echo $token; ?>.png';
    link.href = image;
    link.click();
  }
</script>
</body>
</html>

```

### Lampiran Program Api.Php

```

<?php
header('Content-Type: application/json');
include 'db.php';

if (isset($_GET['token'])) {
  $token_input = $conn->real_escape_string($_GET['token']);
  $action = isset($_GET['action']) ? $_GET['action'] : "";

```

```

if ($action === 'finish') {
    $sql = "UPDATE tokens SET status = 'used' WHERE token_code = '$token_input'";

    if ($conn->query($sql) === TRUE) {
        echo json_encode(["status" => "success"]);
    } else {
        echo json_encode(["status" => "error"]);
    }
} else {
    $sql = "SELECT volume FROM tokens WHERE token_code = '$token_input' AND
status = 'active' LIMIT 1";
    $result = $conn->query($sql);

    if ($result && $result->num_rows > 0) {
        $row = $result->fetch_assoc();
        echo json_encode([
            "status" => "valid",
            "volume" => (int)$row['volume']
        ]);
    } else {
        echo json_encode(["status" => "invalid"]);
    }
} else {
    echo json_encode(["status" => "error", "message" => "No token provided"]);
}

$conn->close();
?>

```

### Lampiran Program db.Php

```

<?php
// db.php
$servername = "mysql-isigalonku.alwaysdata.net";
$username = "isigalonku"; // Default XAMPP
$password = "akbar1073"; // Default XAMPP biasanya kosong
$dbname = "isigalonku_db";

// Buat koneksi
$conn = new mysqli($servername, $username, $password, $dbname);

// Cek koneksi
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$conn->query("SET time zone = '+07:00'");
date_default_timezone_set('Asia/Jakarta');
?>

```

### Lampiran Program Admin.Php

```

<?php
session_start();
include 'db.php';

// --- KONFIGURASI ADMIN ---
$admin_password = "tli21";

// --- LOGIC LOGIN / LOGOUT ---
if (isset($_GET['action']) && $_GET['action'] == 'logout') {
    session_destroy();
    header("Location: admin.php");
    exit();
}

if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['password'])) {
    if ($_POST['password'] == $admin_password) {
        $_SESSION['is_admin'] = true;
    } else {
        $error_msg = "Password Salah!";
    }
}

// Jika belum login, tampilkan Form Login

```

```

if (!isset($_SESSION['is_admin'])) {
?>
<!DOCTYPE html>
<html lang="id">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Login Admin</title>
<style>
body { background: #0d1117; color: #c9d1d9; font-family: sans-serif; display: flex;
justify-content: center; align-items: center; height: 100vh; margin: 0; }
.login-box { background: #161b22; padding: 40px; border: 1px solid #30363d; border-
radius: 10px; text-align: center; box-shadow: 0 0 20px rgba(0,0,0,0.5); }
input { padding: 10px; border-radius: 5px; border: 1px solid #30363d; background:
#0d1117; color: white; margin-bottom: 10px; width: 200px; }
button { background: #238636; color: white; border: none; padding: 10px 20px;
border-radius: 5px; cursor: pointer; width: 100%; font-weight: bold;}
button:hover { background: #2ea043; }
h2 { color: #00f2ff; margin-top: 0; }
.error { color: #ff5555; margin-bottom: 10px; font-size: 0.9em; }
</style>
</head>
<body>
<div class="login-box">
<h2>DASHBOARD ADMIN</h2>
<?php if(isset($error msg)) echo "<div class='error'>$error msg</div>"; ?>
<form method="POST">
<input type="password" name="password" placeholder="Masukkan Password"
required><br>
<button type="submit">LOGIN</button>
</form>
</div>
</body>
</html>
<?php
exit();
}

// --- LOGIC DELETE DATA ---

if (isset($_GET['delete_id'])) {
$id = intval($_GET['delete_id']);
$conn->query("DELETE FROM tokens WHERE id=$id");
header("Location: admin.php?msg=deleted");
exit();
}

```

```

}

if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['bulk_delete'])) {
    if(!empty($_POST['ids'])) {
        $sids_to_delete = implode(",", $_POST['ids']);
        $conn->query("DELETE FROM tokens WHERE id IN ($sids_to_delete)");
        header("Location: admin.php?msg=bulk_deleted");
        exit();
    }
}

// --- LOGIC DASHBOARD & STATISTIK ---

$filter_date = isset($_GET['date']) ? $_GET['date'] : date('Y-m-d');

// Query Harian
$sql_daily = "SELECT SUM(volume) as bought_vol, COUNT(*) as bought_count,
SUM(CASE WHEN status = 'used' THEN volume ELSE 0 END) as used_vol, SUM(CASE
WHEN status = 'used' THEN 1 ELSE 0 END) as used_count FROM tokens WHERE
DATE(purchase_date) = '$filter_date'";
$res_daily = $conn->query($sql_daily)->fetch_assoc();
$d_bought_vol = $res_daily['bought_vol'] ? $res_daily['bought_vol'] / 1000 : 0;
$d_used_vol = $res_daily['used_vol'] ? $res_daily['used_vol'] / 1000 : 0;
$d_bought_cnt = $res_daily['bought_count'];
$d_used_cnt = $res_daily['used_count'];
$d_percent = ($d_bought_vol > 0) ? ($d_used_vol / $d_bought_vol) * 100 : 0;

// Query All Time
$sql_all = "SELECT SUM(volume) as bought_vol, COUNT(*) as bought_count, SUM(CASE
WHEN status = 'used' THEN volume ELSE 0 END) as used_vol, SUM(CASE WHEN status
= 'used' THEN 1 ELSE 0 END) as used_count FROM tokens";
$res_all = $conn->query($sql_all)->fetch_assoc();
$a_bought_vol = $res_all['bought_vol'] ? $res_all['bought_vol'] / 1000 : 0;
$a_used_vol = $res_all['used_vol'] ? $res_all['used_vol'] / 1000 : 0;
$a_bought_cnt = $res_all['bought_count'];
$a_used_cnt = $res_all['used_count'];
$a_percent = ($a_bought_vol > 0) ? ($a_used_vol / $a_bought_vol) * 100 : 0;

// Grafik Data
$sql_chart_daily = "SELECT volume, COUNT(*) as count FROM tokens WHERE
DATE(purchase_date) = '$filter_date' GROUP BY volume ORDER BY volume ASC";
$res_c_daily = $conn->query($sql_chart_daily);
$labels_daily = []; $data_daily = [];
// FIX 1: Cast (int) agar data dikirim sebagai angka, bukan string

```

```

while($row = $res_c_daily->fetch_assoc()) { $labels_daily[] = ($row['volume']/1000)." L";
$data_daily[] = (int)$row['count']; }

$sql_chart_all = "SELECT volume, COUNT(*) as count FROM tokens GROUP BY volume
ORDER BY volume ASC";
$res_c_all = $conn->query($sql_chart_all);
$labels_all = []; $data_all = [];
// FIX 1: Cast (int) disini juga
while($row = $res_c_all->fetch_assoc()) { $labels_all[] = ($row['volume']/1000)." L";
$data_all[] = (int)$row['count']; }

// Query Tabel
$sql_table = "SELECT * FROM tokens WHERE DATE(purchase_date) = '$filter_date'
ORDER BY id DESC";
$res_table = $conn->query($sql_table);
?>

<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Dashboard Admin</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
  <link href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@6.0.0/css/all.min.css"
rel="stylesheet">

  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/chartjs-plugin-datalabels@2.0.0"></script>

  <style>
    body { background-color: #0d1117; color: #c9d1d9; font-family: 'Segoe UI', Tahoma,
Geneva, Verdana, sans-serif; }
    .card { background-color: #161b22; border: 1px solid #30363d; color: #fff; margin-
bottom: 20px; border-radius: 8px;}
    .card-header { border-bottom: 1px solid #30363d; background: #0d1117; color: #00f2ff;
font-weight: bold;}
    .table { color: #c9d1d9; border-color: #30363d; }
    .table thead th { border-bottom: 2px solid #30363d; color: #00f2ff; vertical-align:
middle; }
    .table td { border-color: #30363d; vertical-align: middle; }
    h1, h4 { color: #00f2ff; }

    .form-control { background: #0d1117; border: 1px solid #30363d; color: white; }

```

```

::-webkit-calendar-picker-indicator { filter: invert(1); cursor: pointer; }

.btn-logout { background: #da3633; border: none; color: white; }
.status-badge { padding: 3px 8px; border-radius: 4px; font-size: 0.8em; font-weight:
bold;}
.st-active { background: #d29922; color: black; }
.st-used { background: #238636; color: white; }
.btn-toggle { border: 1px solid #30363d; color: #8b949e; background: #0d1117; font-
size: 0.8rem;}
.btn-toggle.active { background: #238636; color: white; border-color: #238636; }
.stat-value { font-size: 1.8em; font-weight: bold; color: white; }
.stat-label { font-size: 0.8em; color: #8b949e; }
.stat-divider { border-right: 1px solid #30363d; }
.progress { height: 6px; background-color: #30363d; margin-top: 10px;}

.form-check-input { background-color: #0d1117; border-color: #30363d; cursor: pointer;
}
.form-check-input:checked { background-color: #238636; border-color: #238636; }

.btn-trash { color: #da3633; background: transparent; border: none; transition: 0.2s; }
.btn-trash:hover { color: #ff5555; transform: scale(1.1); }
</style>
</head>
<body>

<div class="container py-4">
  <div class="d-flex justify-content-between align-items-center mb-4">
    <div><h2 class="fw-bold mb-0">DASHBOARD <span style="font-weight:300; font-
size:0.8em; color:#8b949e">ADMIN</span></h2></div>
    <a href="admin.php?action=logout" class="btn btn-logout btn-sm px-3">Logout</a>
  </div>

  <?php if(isset($_GET['msg'])): ?>
    <div class="alert alert-success alert-dismissible fade show bg-dark border-success text-
success" role="alert">
      <i class="fas fa-check-circle me-2"></i>
      <?php echo ($_GET['msg'] == 'deleted') ? 'Data berhasil dihapus!' : 'Beberapa data
berhasil dihapus sekaligus!'; ?>
      <button type="button" class="btn-close btn-close-white" data-bs-dismiss="alert" aria-
label="Close"></button>
    </div>
  <?php endif; ?>

  <div class="card p-3 mb-4">
    <form class="row g-3 align-items-center">

```

```

        <div class="col-auto"><label class="col-form-label fw-bold"><img alt="calendar icon" data-bbox="688 138 708 153"/> Lihat
Tanggal:</label></div>
        <div class="col-auto"><input type="date" name="date" class="form-control"
value="<?php echo $filter date; ?>" onchange="this.form.submit()"></div>
    </form>
</div>

<div class="row">
    <div class="col-md-6">
        <div class="card h-100">
            <div class="card-header d-flex justify-content-between">
                <span>STATISTIK HARIAN (<?php echo date('d/m', strtotime($filter date));
?>)</span>
                <span class="badge bg-success">Today</span>
            </div>
            <div class="card-body">
                <div class="row text-center">
                    <div class="col-6 stat-divider">
                        <div class="stat-value text-primary"><?php echo $d bought vol; ?>
L</div>
                        <div class="stat-label">Air Dibeli<br>(<?php echo $d_bought_cnt; ?>
Transaksi)</div>
                    </div>
                    <div class="col-6">
                        <div class="stat-value text-success"><?php echo $d_used_vol; ?> L</div>
                        <div class="stat-label">Air Diambil<br>(<?php echo $d_used cnt; ?>
Transaksi)</div>
                    </div>
                </div>
                <div class="mt-3">
                    <div class="d-flex justify-content-between small text-muted"><span>Realisasi
Pengambilan</span><span><?php echo number_format($d_percent, 1); ?>%</span></div>
                    <div class="progress"><div class="progress-bar bg-success"
role="progressbar" style="width: <?php echo $d_percent; ?>%"></div></div>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="col-md-6">
        <div class="card h-100" style="border-color: #1f6feb;">
            <div class="card-header d-flex justify-content-between" style="color: #58a6ff;">
                <span>STATISTIK ALL TIME</span>
                <span class="badge bg-primary">Total</span>
            </div>
            <div class="card-body">

```

```

        <div class="row text-center">
            <div class="col-6 stat-divider">
                <div class="stat-value text-primary"><?php echo $a_bought_vol; ?>
L</div>
                <div class="stat-label">Total Dibeli<br>( <?php echo $a_bought_cnt; ?>
Transaksi)</div>
            </div>
            <div class="col-6">
                <div class="stat-value text-success"><?php echo $a_used_vol; ?> L</div>
                <div class="stat-label">Total Diambil<br>( <?php echo $a_used_cnt; ?>
Transaksi)</div>
            </div>
        </div>
        <div class="mt-3">
            <div class="d-flex justify-content-between small text-muted"><span>Realisasi
Pengambilan</span><span><?php echo number_format($a_percent, 1); ?>%</span></div>
            <div class="progress"><div class="progress-bar bg-success"
role="progressbar" style="width: <?php echo $a_percent; ?>%"></div></div>
            </div>
        </div>
    </div>
</div>

<div class="row mt-2">
    <div class="col-md-4">
        <div class="card h-100">
            <div class="card-header d-flex justify-content-between align-items-center">
                <span><img alt="Bar chart icon" data-bbox="333 611 353 626" style="vertical-align: middle;"/> VARIAN TERLARIS</span>
                <div class="btn-group btn-group-sm">
                    <button type="button" class="btn btn-toggle active" id="btnChartDaily"
onclick="switchChart('daily')">Hari</button>
                    <button type="button" class="btn btn-toggle" id="btnChartAll"
onclick="switchChart('all')">Total</button>
                </div>
            </div>
            <div class="card-body d-flex justify-content-center align-items-center">
                <div style="height: 250px; width: 100%;"><canvas
id="variantChart"></canvas></div>
            </div>
        </div>
    </div>
</div>

<div class="col-md-8">
    <div class="card h-100">

```

```

<div class="card-header d-flex justify-content-between align-items-center">
  <span>  LOG TRANSAKSI</span>
  <button id="btnBulkDelete" onclick="submitBulkDelete()" class="btn btn-
danger btn-sm d-none">
    <i class="fas fa-trash-alt me-1"></i> Hapus Terpilih
  </button>
</div>
<div class="card-body p-0">
  <form id="formBulkDelete" method="POST" action="">
    <input type="hidden" name="bulk_delete" value="1">

    <div class="table-responsive" style="max-height: 350px; overflow-y: auto;">
      <table class="table table-hover table-dark mb-0 table-sm">
        <thead style="position: sticky; top: 0; background: #161b22; z-index:
5;">
          <tr>
            <th width="40" class="text-center"><input type="checkbox"
id="selectAll" class="form-check-input" onclick="toggleSelectAll()"></th>
            <th>Jam</th>
            <th>Token</th>
            <th>Vol</th>
            <th>Status</th>
            <th width="50" class="text-center">Aksi</th>
          </tr>
        </thead>
        <tbody>
          <?php if ($res_table->num_rows > 0): ?>
            <?php while($row = $res_table->fetch_assoc()): ?>
              <tr>
                <td class="text-center"><input type="checkbox" name="ids[]"
value="<?php echo $row['id']; ?>" class="form-check-input row-checkbox"
onclick="checkBulkButton()"></td>
                <td><?php echo date('H:i', strtotime($row['purchase_date']));
?></td>
                <td style="font-family: monospace; color: #e3b341"><?php
echo $row['token_code']; ?></td>
                <td><?php echo $row['volume'] / 1000; ?>L</td>
                <td>
                  <?php
                    $st = $row['status'];
                    $cls = ($st == 'active') ? 'st-active' : (($st == 'used') ? 'st-
used' : 'st-paid');
                    echo "<span class='status-badge $cls'" . strtoupper($st) .
" /></span>";
                  <?>

```

```

        </td>
        <td class="text-center">
            <a href="#" onclick="confirmDelete(<?php echo $row['id'];
?>)" class="btn-trash" title="Hapus"><i class="fas fa-trash"></i></a>
        </td>
    </tr>
<?php endwhile; ?>
<?php else: ?>
    <tr><td colspan="6" class="text-center text-muted py-4">Belum ada
data di tanggal ini.</td></tr>
<?php endif; ?>
</tbody>
</table>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>

```

```

<script>
    Chart.register(ChartDataLabels);

    const labelsDaily = <?php echo json_encode($labels_daily); ?>;
    const dataDaily = <?php echo json_encode($data_daily); ?>;
    const labelsAll = <?php echo json_encode($labels_all); ?>;
    const dataAll = <?php echo json_encode($data_all); ?>;

    const ctx = document.getElementById('variantChart');
    let myChart = new Chart(ctx, {
        type: 'doughnut',
        data: {
            labels: labelsDaily.length ? labelsDaily : ['No Data'],
            datasets: [{
                data: dataDaily.length ? dataDaily : [1],
                backgroundColor: dataDaily.length ? ['#238636', '#1f6feb', '#d29922', '#da3633'] :
[#30363d],
                borderWidth: 0
            }]
        },
        options: {
            responsive: true,
            maintainAspectRatio: false,
            plugins: {

```

```

    legend: { position: 'right', labels: { color: '#c9d1d9' } },
    datalabels: {
      color: '#ffffff',
      font: { weight: 'bold', size: 14 },
      formatter: (value, ctx) => {
        if(ctx.chart.data.labels[0] === 'No Data') return "";

        let sum = 0;
        let dataArr = ctx.chart.data.datasets[0].data;
        // FIX 2: Pastikan dikonversi ke Number agar tidak terjadi String
Concatenation
        dataArr.forEach(data => { sum += Number(data); });

        if(sum === 0) return "0%"; // Cegah pembagian nol

        let percentage = (value * 100 / sum).toFixed(1) + "%";
        return percentage;
      }
    }
  }
});

function switchChart(mode) {
  const btnDaily = document.getElementById('btnChartDaily');
  const btnAll = document.getElementById('btnChartAll');
  if (mode === 'daily') {
    btnDaily.classList.add('active'); btnAll.classList.remove('active');
    myChart.data.labels = labelsDaily.length ? labelsDaily : ['No Data'];
    myChart.data.datasets[0].data = dataDaily.length ? dataDaily : [1];
    myChart.data.datasets[0].backgroundColor = dataDaily.length ? ['#238636', '#1f6feb',
'#d29922', '#da3633'] : ['#30363d'];
  } else {
    btnDaily.classList.remove('active'); btnAll.classList.add('active');
    myChart.data.labels = labelsAll.length ? labelsAll : ['No Data'];
    myChart.data.datasets[0].data = dataAll.length ? dataAll : [1];
    myChart.data.datasets[0].backgroundColor = dataAll.length ? ['#238636', '#1f6feb',
'#d29922', '#da3633'] : ['#30363d'];
  }
  myChart.update();
}

// --- DELETE FUNCTIONS ---
function confirmDelete(id) {
  if(confirm("Apakah Anda yakin ingin menghapus data ini secara permanen?")) {

```

```

        window.location.href = "admin.php?delete_id=" + id;
    }
}

function toggleSelectAll() {
    const checkboxes = document.querySelectorAll('.row-checkbox');
    const master = document.getElementById('selectAll');
    checkboxes.forEach(cb => cb.checked = master.checked);
    checkBulkButton();
}

function checkBulkButton() {
    const checkboxes = document.querySelectorAll('.row-checkbox:checked');
    const btn = document.getElementById('btnBulkDelete');
    if (checkboxes.length > 0) {
        btn.classList.remove('d-none');
        btn.innerHTML = `<i class="fas fa-trash-alt me-1"></i> Hapus
($ {checkboxes.length}) Data`;
    } else {
        btn.classList.add('d-none');
    }
}

function submitBulkDelete() {
    if(confirm("PERINGATAN! Anda akan menghapus banyak data sekaligus. Tindakan ini
tidak bisa dibatalkan. Lanjutkan?")) {
        document.getElementById('formBulkDelete').submit();
    }
}
</script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>
</html>

```

# Lampiran Datasheet ESP 32 DEVKIT V1

## Chapter 1. ESP32-DevKitC

For details please refer to ESP Product Selector.

### 1.1.3 Functional Description

The following figure and the table below describe the key components, interfaces and controls of the ESP32-DevKitC V4 board.

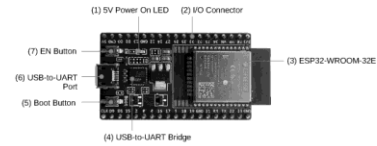


Fig. 1: ESP32-DevKitC V4 with ESP32-WROOM-32E module soldered

The key components of the board are described, starting from the 5V Power On LED, in a clockwise direction.

No.	Key Component	Description
1	5V Power On LED	Turns on when the USB or an external 5V power supply is connected to the board. For details see the schematics in <i>Internal Documents</i> .
2	I/O Connector	Most of the pins on the ESP module are broken out to the pin headers on the board. This can program ESP32 to enable multiple functions such as PWM, ADC, DAC, I2C, I2S, SPI, etc.
3	ESP32-WROOM-32E	A module with ESP32 at its core. For more information, see ESP32-WROOM-32E datasheet.
4	USB-to-UART Bridge	Single USB-to-UART bridge chip, providing transfer rates up to 3 Mbps.
5	Boot Button	Download button. Holding down <b>Boot</b> and then pressing <b>EN</b> initiates Firmware Download mode for downloading firmware through the serial port.
6	USB-to-UART Port	A Micro-USB port used for power supply to the board, as well as for communication between a computer and the ESP32-WROOM-32E module.
7	EN Button	Reset button.

### 1.1.4 Power Supply Options

There are three mutually exclusive ways to provide power to the board:

- Micro USB port, default power supply
- 5V and GND header pins
- 3V3 and GND header pins

**Warning:** The power supply must be provided using **one and only one** of the options above, otherwise the board and/or the power supply source can be damaged.

## Chapter 1. ESP32-DevKitC

### 1.1.5 Header Block

The two tables below provide the **Name** and **Function** of I/O header pins on both sides of the board, as shown in *ESP32-DevKitC V4 with ESP32-WROOM-32E module soldered*.

J2

No.	Name	Type <sup>1</sup>	Function
1	3V3	P	3.3 V power supply
2	EN	I	CHIP_PU_Reset
3	VP	I	GPIO06, ADC1_CH0, S_VN
4	VN	I	GPIO09, ADC1_CH1, S_VN
5	IO04	I	GPIO04, ADC1_CH0, VDET_1
6	IO05	I	GPIO05, ADC1_CH1, VDET_2
7	IO02	IO	GPIO02, ADC1_CH0, TOUCH_CH0, XTAL_32K_P, GPIO3,
8	IO03	IO	GPIO03, ADC1_CH1, TOUCH_CH1, XTAL_32K_N, ADC1_CH5, TOUCH_CH5,
9	IO05	IO	GPIO05, ADC2_CH0, DAC_1
10	IO06	IO	GPIO06, ADC2_CH0, DAC_2
11	IO07	IO	GPIO07, ADC2_CH1, TOUCH_CH7
12	IO04	IO	GPIO04, ADC2_CH0, TOUCH_CH0, MTMS
13	IO02	IO	GPIO02, ADC2_CH1, TOUCH_CH5, MTDS
14	GND	G	Ground
15	IO03	IO	GPIO03, ADC2_CH0, TOUCH_CH0, MTCK
16	D2	IO	GPIO9, D2 <sup>2</sup>
17	D3	IO	GPIO10, D3 <sup>2</sup>
18	CMD	IO	GPIO11, CMD <sup>2</sup>
19	5V	P	5 V power supply

<sup>1</sup> P: Power supply; I: Input; O: Output.  
<sup>2</sup> The pins D1, D2, D3, D4 and D5 can be used internally for communication between ESP32 and SPI flash memory. They are grouped on both sides near the USB connector. Avoid using these pins, as they divert access to the SPI flash memory/SPI RAM.

# Lampiran Datasheet LM2596

## TEXAS INSTRUMENTS LM2596

www.ti.com BVI129C - NOVEMBER 1999 - REVISED APRIL 2012  
**LM2596 SIMPLE SWITCHER® Power Converter 150 kHz 3A Step-Down Voltage Regulator**  
 Check for Samples: LM2596

- ### FEATURES
- 3.3V, 5V, 12V, and Adjustable Output Versions
  - Adjustable Version Output Voltage Range, 1.2V to 37V 84% Max Over-Line and Load Conditions
  - Available in TO-220 and TO-263 Packages
  - Ensured 3A Output Load Current
  - Input Voltage Range Up to 40V
  - Requires Only 4 External Components
  - Excellent Line and Load Regulation Specifications
  - 150 kHz Fixed Frequency Internal Oscillator
  - TTL Shutdown Capability
  - Low Power Standby Mode,  $I_q$  Typically 80  $\mu$ A
  - High Efficiency
  - Uses Readily Available Standard Inductors
  - Thermal Shutdown and Current Limit Protection

### DESCRIPTION

The LM2596 series of regulators are monolithic integrated circuits that provide all the active functions for a step-down (buck) switching regulator, capable of driving a 3A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3V, 5V, 12V, and an adjustable output version.

Requiring a minimum number of external components, these regulators are simple to use and include internal frequency compensation, and a fixed-frequency oscillator.

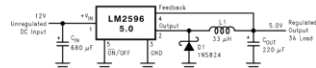
The LM2596 series operates at a switching frequency of 150 kHz, thus allowing smaller sized filter components than what would be needed with lower frequency switching regulators. Available in a standard 5-lead TO-220 package with several different lead bend options, and a 5-lead TO-263 surface mount package.

A standard series of inductors are available from several different manufacturers optimized for use with the LM2596 series. This feature greatly simplifies the design of switch-mode power supplies.

Other features include an ensured  $\pm 4\%$  tolerance on output voltage under specified input voltage and output load conditions, and  $\pm 15\%$  on the oscillator frequency. External shutdown is included, featuring typically 50  $\mu$ A standby current. Self protection features include a two stage frequency reducing current limit for the output switch and an over temperature shutdown for complete protection under fault conditions. <sup>(1)</sup>

(1) Patent Number 5,362,518.

### Typical Application (Fixed Output Voltage Versions)



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.  
 SIMPLE SWITCHER is a registered trademark of Texas Instruments.  
 All other trademarks are the property of their respective owners.  
 PRODUCTION DATA indicates a component that is suitable for production manufacturing. See the terms of the TI EULA for details.  
 Copyright © 1999-2013, Texas Instruments Incorporated

## LM2596 BVI129C - NOVEMBER 1999 - REVISED APRIL 2012 www.ti.com

### Connection Diagrams

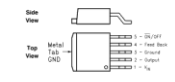
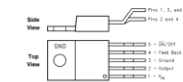


Figure 1. 5-Lead Bent and Staggered Leads, Through Hole TO-220 (T) Package See Package Number N000000  
 Figure 2. 5-Lead DOPAK/TO-263 (S) Package See Package Number KTT000000

These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOSFETs.

### Absolute Maximum Ratings <sup>(1)(2)</sup>

Maximum Output Voltage	40V
SW, OFF Pin Input Voltage	-0.3 to 5 V +20V
Feedback Pin Voltage	-0.3 to 5 V +25V
Output Voltage to Ground (Steady State)	-1V
Power Dissipation	Internally limited
Storage Temperature Range	-65°C to +150°C
ESD Susceptibility	
Human Body Model <sup>(3)</sup>	2 kV
Lead Temperature	
DOPAK/TO-263 Package	+215°C
Vapor Phase (90 sec.)	+215°C
Infrared (1 sec.)	+260°C
TO-220 Package (Soldering, 10 sec.)	+260°C
Maximum Junction Temperature	+150°C

(1) Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is intended to be functional, but do not ensure specific performance limits. For ensured specifications and test conditions, see the Electrical Characteristics.  
 (2) If Military/Aerospace specified devices are required, please contact the Texas Instruments Sales Office/Distributors for availability and specifications.  
 (3) The human body model is a 100-pF capacitor discharged through a 1.5-k resistor into each pin.

### Operating Conditions

Temperature Range	-40°C to +125°C
Supply Voltage	4.5V to 40V

# Lampiran Datasheet Sensor QR GM-67



## 1 Introduction of Module

### 1.1 Introduction

GM67 1D 2D barcode scanner, with high integration and compact size, can be easily integrated into various industrial equipment and application environments. It can not only read various 1D barcode easily, but also read 2D barcode at high speed. It has a very high scanning rate for linear bar codes. For barcode on paper and screen, can also be easily scanned.



### 1.2 Technical Specification

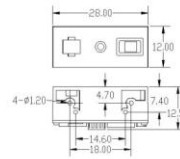
Scanning Performance	Scan Mode	640*480 CMOS		
	Lighting	White LED		
	Collimate	Red LED		
	Read Code Type	2D	QR Code, Data Matrix, PDF417, Aztec, Micro QR, Micro PDF417	
		1D	EAN,UPC,Code 39,Code 93,Code 128,UCC/EAN 128, Codabar, Interleaved 2 of 5, ITF-6,ITF-14,ISBN/JSSN,MS-Plessey,GS1 Database, Code 11,Industrial 25, Standard 25,Plessey, Matrix 2 of 5	
	Accuracy of reading	≥5mil		
	Depth of Field	EAN-13	60mm-220mm	(13mil)
		Code 39	40mm-90mm	(5mil 3D byte)
		QR Code	30mm-220mm	(10mil 1D byte)
		Data Matrix	40mm-220mm	(10mil 2D byte)
	PDF417	30mm-130mm	(6.67mil 7 byte)	
Contrast	≥25%			
Scanning angle	Intersection angle 360°, Elevation ± 55° Deflection angle ± 55°			
Viewing Angle	D:42° H:34° V:26°			
Mechanical/Interface	TTL-232 . USB (HID-KBW, Virtual serial port)			

1



Electrical Parameters	Dimension	27.5(W)*46.5(D)*14(H)mm
	Operating Voltage	DC 5 V
	Operating Current	180mA
Environmental Parameters	Standby Current	USB-25mA; TTL-10mA
	Operating Temperature	-20°C~+50°C
	Storage Temperature	-40°C~+70°C
	Operating Humidity	5%~95% (Non-Condensing)
Environmental Light	0~100000LUX	

### 1.3 Dimension (mm)

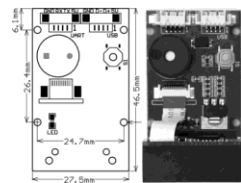


2

www.hgrow.com



### Circuit Board Diagram:



### 1.4 Scan Head Definition

The cable is 12PIN directly connected with FPC flexible line, with an interval of 0.5mm

Pin	Input/Output	Definition	Introductions
Pin 1	Power	VCC	Input-3.3V
Pin 2	Power	VCC	Input-3.3V
Pin 3	Ground	GND	-
Pin 4	Input	RX	Serial port receiver signal
Pin 5	Output	TX	Serial port send signal
Pin 6	Input	D-	USB - D-signal
Pin 7	Output	D+	USB - D-signal
Pin 8	Ground	GND	-
Pin 9	Output	BEEPER	sh output, low level when free
Pin10	Output	DLED	Decoding success light, low level when free
Pin11	-	NC	-
Pin12	Input	TRIG	Weak Pull-up, Low level trigger engine decoding

3

www.hgrow.com



### 1.5 Circuit Board Communication Interface

#### UART Communication interface definition:

Pin	Name	Definition	Description
1	5V	Power Input	Power input 5V
2	TX	Out	Data output, TTL logical level
3	RX	In	Data input, TTL logical level
4	GND	-	Signal ground

#### USB Communication interface definition:

Pin	Name	Definition	Description
1	5V	Power Input	Power input 5V
2	D+	Out	USB data output
3	D-	In	USB data input
4	GND	-	Signal ground

4

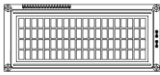
www.hgrow.com

## Lampiran Datasheet LCD I2C 20x4

**VISHAY** **LCD-020N004L** **Vishey**

www.vishay.com

### 20 x 4 Character LCD



**FEATURES**

- Type: Character
- Display format: 20 x 4 characters
- Built-in controller: ST 7066 (or equivalent)
- Display: 1.80"
- 8 x 8 dots includes cursor
- 8 x 8 green supply (also available for 3 x 3)
- LED can be driven by pin 1, pin 2, pin 15, pin 16 or A and K
- V<sub>CC</sub> options for a 3 V power supply
- Maximal compatibility: For definitions of compliance please see page 2 of the specification

**MECHANICAL DATA**

ITEM	STANDARD VALUE	UNIT
Module thickness	1.65 ± 0.1	mm
Display area	12.51 x 4.53	mm
Dot size	0.81 x 1.52	mm
Dot Pitch	0.81 x 1.52	mm
Mounting size	12.8 x 7.62	mm
Character size	4.81 x 9.23	mm

**ABSOLUTE MAXIMUM RATINGS**

ITEM	SYMBOL	STANDARD VALUE	UNIT
Power Supply	V <sub>CC</sub> to V <sub>EE</sub>	-0.3	V
Screen Voltage	V <sub>S</sub>	-0.3	V

**ELECTRICAL CHARACTERISTICS**

ITEM	SYMBOL	CONDITION	STANDARD VALUE			UNIT
			MIN.	TPP.	MAX.	
Input Voltage	V <sub>CC</sub>	V <sub>CC</sub> = +5 V	4.5	5.0	5.5	V
Supply Current	I <sub>CC</sub>	V <sub>CC</sub> = +5 V	2.2	3.0	3.5	mA
Recommended C Driving Voltage for Normal Temperature	V <sub>CC</sub> to V <sub>S</sub>	25 °C	5.0	5.1	5.2	V
		0 °C	4.8	4.8	5.0	V
		50 °C	4.1	4.1	4.7	V
LED Forward Voltage	V <sub>F</sub>	25 °C	3.9	4.2	4.5	V
		50 °C	3.1	3.8	4.3	V
LED Forward Current	I <sub>F</sub>	25 °C	-	8.0	16.0	mA
EL Power Supply Current	I <sub>EL</sub>	V <sub>CC</sub> = +5 V, I <sub>CC</sub> = 400 µA	-	-	8.0	mA

**OPTIONS**

PN	STN	STN	STN	STN	STN	None	LED	EL	COFL
Code	Code	Code	Code	Code	Code				
x	x	x	x	x	x		x	x	x

**DISPLAY CHARACTER ADDRESS CODE**

Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
DD RAM Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13
DD RAM Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13
DD RAM Address	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27
DD RAM Address	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D

Revision: 08-Oct-12

**VISHAY** **LCD-020N004L** **Vishey**

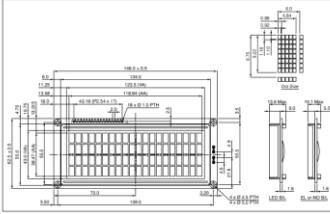
www.vishay.com

### 20 x 4 Character LCD

**INTERFACE PIN FUNCTION**

PIN NO.	SYMBOL	FUNCTION
1	V <sub>CC</sub>	Ground
2	V <sub>CC</sub>	+5V or +3V
3	V <sub>S</sub>	Control adjustment
4	RS	RS: register select signal
5	R/W	RS: register select signal
6	E	RS: enable signal
7	DB0	RS: data bus line
8	DB1	RS: data bus line
9	DB2	RS: data bus line
10	DB3	RS: data bus line
11	DB4	RS: data bus line
12	DB5	RS: data bus line
13	DB6	RS: data bus line
14	DB7	RS: data bus line
15	A	Power supply for LED (0.7 V)
16	K	Power supply for LED (0.7 V)
17	NC/No	NC or register voltage output
18	NC	NC connection

**DIMENSIONS in millimeters**

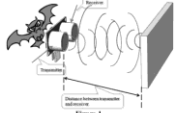


Revision: 08-Oct-12

## Lampiran Datasheet Sensor Ultrasonik HC-Sr04

**1. Introduction:**

Ultrasonic is an excellent way of figuring out what's in the immediate vicinity of your Arduino. The basics of using ultrasonic are like this: you shoot out a sound wave to hit a echo back, and if you have your timing right, you'll know if anything is out there and how far away it is. This is called echolocation and it's how bats and dolphins find objects in the dark and underwater, though they use lower frequencies than you can use with your Arduino. Figure 1 shows the working principle of ultrasonic ranging concept.



**Figure-1**

HC-SR04 Ultrasonic Sensor is a very affordable proximity/distance sensor that has been used mainly for object avoidance in various robotics projects. It has also been used in turret applications, water level sensing, and even in a parking sensor.


This module is the second generation of the popular HC-SR04 Low Cost Ultrasonic Sensor. Unlike the first generation HC-SR04 that can only operate between 4.5V-5V DC, this new version has wider input voltage range, allow it to work with controller operates on 3.3V. HC-SR04 Ultrasonic sensor provides a very low-cost and easy method of distance measurement. It measures distance using sound, an ultrasonic (well above human hearing) pulse (40KHz) is transmitted from the unit and distance-to-target is determined by measuring the time required for the echo return. This sensor offers excellent range accuracy and stable readings in an open-air package. An on-board 2.5mm pitch pin header allows the sensor to be plugged into a standard breadboard for easy prototyping.

**2. Module Specification**

Electrical Parameters	Value
Operating Voltage	3.3Vcc - 5Vcc
Quiescent Current	<2mA
Operating Current	50mA
Operating Frequency	40KHz
Operating Range & Accuracy	2cm - 400cm (1in - 150in) ± 3mm
Sensitivity	65µs/Nm
Sound Pressure	112dB
Effective Angle	15°
Connector	4-pin header with 2.54mm pitch
Dimension	45mm x 20mm x 15mm
Weight	5g

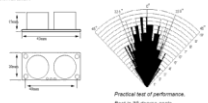
3 | www.handsontec.com

**4. Hardware Information**

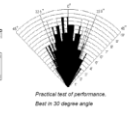


Pin	Function
VCC	Triggering Input Pin, 3.3V-5V
TRIG	Triggering Input Pin, 50µs-10µs Pulse
ECHO	TRIG Output Pin, Proportional to distance
GND	Ground Pin

**4.1 Mechanical Dimension**



**4.2 Timing Diagram**



The timing diagram, Figure-4 is shown below. You only need to supply a short 10µs pulse to "Trigger" input pin to start the ranging. The module will send out 8-cycles burst of ultrasonic at 40KHz and receive "Echo"

5 | www.handsontec.com

# Lampiran Datasheet Sensor Flowmeter YF-S201

Fujian Shunde Zhongjiang Energy Saving Electronics Co., Ltd. [www.shdshifi.com.cn](http://www.shdshifi.com.cn) Tel: 0757-22119057 26618322

Fujian Shunde Zhongjiang Energy Saving Electronics Co., Ltd. [www.shdshifi.com.cn](http://www.shdshifi.com.cn) Tel: 0757-22119057 26618322

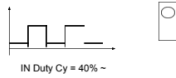
**Product Introduction:** The

water flow sensor is mainly composed of plastic valve body, water flow rotor assembly and Hall sensor. It is installed on the water inlet end of the water heater and is used to detect the water flow rate. When water passes through the water flow rotor assembly, the magnetic rotor rotates and the speed changes with the flow rate change. The Hall sensor outputs a corresponding pulse signal and feeds it back to the controller. The device determines the size of the water flow and regulates it.

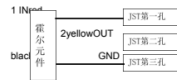
Cautions on use are strictly prohibited.

Do not throw or bump. The installation direction diagram is vertical installation, the inclination does not exceed 5 degrees. OUT medium temperature should not exceed 120°C.

TheThird,outputwaveform:



four lead:



**V. Technical**

parameters apply	to automatic gas water heater
1 the minimum Rated working voltage	DC 5V-24V
2 Maximum working current	15 mA (DC 5V)
3 Operating voltage range	DC 5 to 18 V
4 Load capacity	≤ 10 mA (DC 5V)
5 Operating temperature range	≤ 80 °C
6 Use humidity range of	35% ~ 90% RH (or frost)
7 allows pressure	or less 1.75Mpa pressure
8 storage temperature	-25 ~ + 80 °C
9 storage humidity	25% ~ 95% RH
1, Output pulse high level	> DC 4.5 V (input voltage DC 5 V)
2, Output pulse low level	< DC 0.5 V (input voltage DC 5 V)
3. Accuracy (flow rate-pulse output)	1 ~ 30 L / min±With±5%
4. Output pulse duty cycle	50 ± 10%
5. Output rise time	0.5μs
6. Output fall time	0.1μs
7 Flow rate-pulse characteristic	level test pulse frequency (Hz) = [7.5 Q] ± 3% (Level test) (Q is the flow rate L / min)
8. The impact-resistant	product is well packaged, and it drops freely from the height of 50cm in the X, Y, and Z directions to the concrete surface, and the accuracy changes within 5%.
9 insulation resistance	Hall sensor and copper wire body insulation resistance above 100MΩ (DC 500V)
10, heat resistance	in the environment of 80 ± 3 °C for 48 hours, returned to normal temperature for 1-2 hours without abnormalities, and parts without cracks, slack, expansion, deformation and other phenomena, the accuracy change within 10%.

Fujian Shunde Zhongjiang Energy Saving Electronics Co., Ltd. [www.shdshifi.com.cn](http://www.shdshifi.com.cn) Tel: 0757-22119057 26618322

11. Cold resistance	- Put it in the environment of -20 ± 3 °C for 48h, return to normal temperature for 1-2h without abnormality, and the parts are free from cracks, slackness, swelling, deformation and other phenomena, and the accuracy changes within 10%.
12. Moisture resistance	After leaving for 72 hours in an environment with 40 ± 2 °C and relative humidity 90% ~ 95% RH, the insulation resistance is above 1MΩ.
13.strength:	PullingA 10N pulling force is applied to the lead-out wire for 1 minute, no loosening or breaking, and no change in performance.
14. Durability	At normal temperature, 0.1 MPa water pressure is passed from the water inlet to turn on 1S and turn off 0.5S as a cycle. The test is 300,000 times without abnormality.