

BAB IV

PEMBUATAN ALAT

Dalam proses perancangan dan implementasi sistem monitoring kualitas air (pH, TDS) dan debit air rumah tangga berbasis IoT dan cloud, diperlukan suatu metode penelitian yang sistematis untuk memastikan setiap tahapan pengembangan dapat dilaksanakan secara terstruktur dan menghasilkan sistem yang berfungsi sesuai dengan tujuan penelitian. Bab ini menjelaskan tahapan perancangan dan pembuatan sistem yang meliputi proses perencanaan, perancangan, implementasi, serta pengujian untuk memastikan seluruh bagian sistem dapat bekerja secara optimal dan terintegrasi.

Setiap tahapan dalam pengembangan sistem memiliki keterkaitan yang erat sehingga diperlukan perencanaan yang matang agar perangkat yang dihasilkan mampu memenuhi kebutuhan pemantauan kualitas dan debit air rumah tangga secara real-time. Secara umum, proses pembuatan sistem pada penelitian ini dibagi menjadi dua tahapan utama, yaitu sebagai berikut.

4.1. Pembuatan Perangkat Keras

Proses perancangan perangkat keras diawali dengan studi literatur yang dilakukan melalui pengumpulan berbagai referensi dari sumber ilmiah yang relevan, seperti buku, artikel ilmiah, jurnal, serta penelitian terdahulu yang berkaitan dengan sistem monitoring kualitas air dan debit air rumah tangga berbasis IoT dan cloud. Tahap ini bertujuan untuk memperoleh pemahaman yang komprehensif mengenai konsep, metode, dan teknologi yang dapat diterapkan dalam penelitian. Berdasarkan hasil studi literatur, dilakukan analisis kebutuhan sistem untuk menentukan spesifikasi perangkat keras yang diperlukan agar mampu mendukung fungsi dan kinerja sistem secara optimal. Tahap analisis ini mencakup identifikasi kebutuhan perangkat, karakteristik operasional, serta kesesuaian antar komponen yang akan digunakan dalam sistem. Selanjutnya, dilakukan proses perancangan rangkaian perangkat keras dengan mempertimbangkan integrasi dan konektivitas antar komponen agar dapat bekerja secara terkoordinasi. Perancangan

dilakukan secara sistematis untuk memastikan setiap bagian sistem dapat menjalankan fungsinya sesuai dengan kebutuhan yang telah ditetapkan.

4.1.1. Alat dan Bahan

Identifikasi kebutuhan merupakan tahapan awal untuk memastikan bahwa perangkat keras dan perangkat lunak yang digunakan telah sesuai dengan kebutuhan fungsional serta spesifikasi sistem yang dirancang. Tabel 4.1 menunjukkan daftar kebutuhan perangkat keras dan perangkat lunak yang digunakan sebagai acuan dalam proses perancangan, implementasi, dan pengujian sistem. Tabel Melalui identifikasi kebutuhan yang sistematis, proses pengembangan dapat dilakukan secara lebih efektif, terstruktur, dan mampu meminimalkan kesalahan dalam pemilihan komponen maupun pengalokasian sumber daya selama penelitian berlangsung.

Tabel 4- 1 Bahan Pembuatan Tugas Akhir

NO	KOMPONEN	SPESIFIKASI	JUMLAH
1	Mikrokontroler	Wroom	1
2	Sensor pH	PH-4502C	1
3	Sensor TDS	V1.0	1
4	Sensor Water Flow	YF-S201C	1
5	Sensor Suhu	DS18B20	1
6	RTC	DS3231	1
7	ADS	1115	1
8	Power Supply 5V 2A	M-510	1
9	Kapasitor	470 μ F	2
10	Resistor	4,7k Ω	1
11	Terminal Blok	Pitch 5.08 mm	1
12	Kabel	JST 3 Pin	3

NO	KOMPONEN	SPESIFIKASI	JUMLAH
13	Kabel 2 x 0,75	NYMHY	2 Meter
14	PCB	Single Layer	1
15	Switch	ON\OFF	1
16	Steker	Broco	1
17	Seal Tape	M-Seal	1
18	Spacer	1,5 Cm	7
19	Pipa ½ inch	Rucika	1 Meter
20	Drag Dalam ½ inch	Rucika	2
21	Box Hitam	X5	1

Selain bahan pada Tabel 4.1, berikut adalah daftar peralatan yang digunakan dalam proses pembuatan alat Tugas Akhir ini (Tabel 4.2).

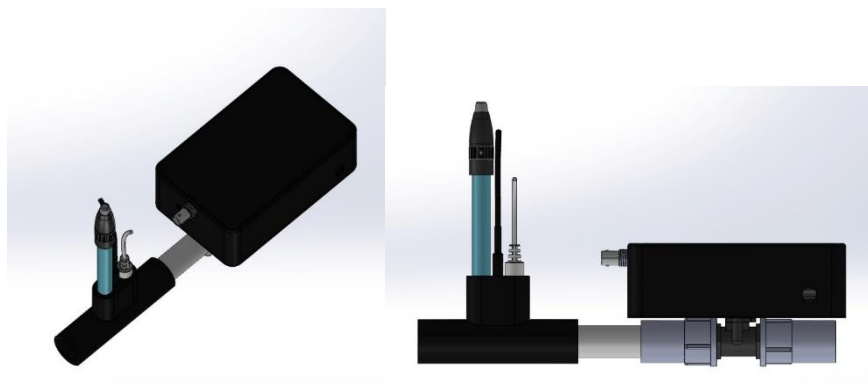
Tabel 4- 2 Alat yang digunakan

NO	KOMPONEN	JUMLAH
1	Multimeter	1
2	Solder	1
3	Penyedot Timah	1
4	Tang Potong	1
5	Obeng (+)	1
6	Obeng (-)	1
7	Gerinda	2
8	Pemotong Pipa	1
9	Bor	1

NO	KOMPONEN	JUMLAH
10	Mata Bor	1

4.1.2. Perancangan Mekanik

Setelah tahap perancangan dan pembuatan perangkat keras elektronik selesai dilakukan, tahapan selanjutnya adalah perancangan mekanik. Tahap ini bertujuan untuk membangun struktur fisik yang berfungsi sebagai wadah sekaligus pelindung bagi seluruh komponen elektronik, sehingga alat dapat beroperasi secara aman dan andal ketika bersentuhan langsung dengan lingkungan instalasi air rumah tangga. Perancangan mekanik pada penelitian ini mencakup tiga aspek utama, yaitu perancangan casing (box) sebagai pelindung komponen elektronik, perancangan jalur perpipaian sebagai media pengujian sensor secara langsung pada aliran air, serta integrasi antara struktur mekanik dengan rangkaian elektronik yang telah dibuat sebelumnya. Desain mekanik casing hasil perancangan ditunjukkan pada Gambar 4.1.

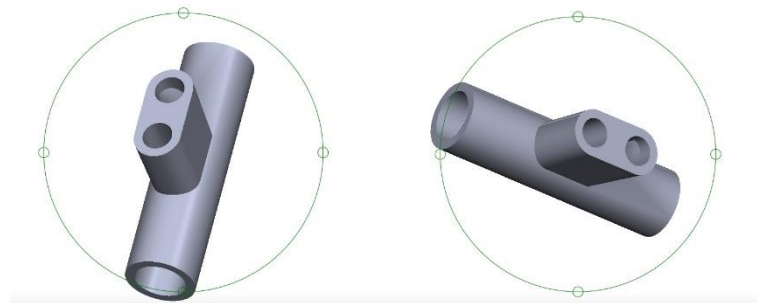


Gambar 4. 1 Perancangan Mekanik

Pada bagian casing, digunakan box hitam X5 yang berukuran 14,8 cm x 9,8 cm x 5 cm sebagai rumah utama bagi ESP32, PCB, modul ADS1115, RTC DS3231, serta rangkaian catu daya. Pemilihan box tertutup ini dimaksudkan untuk melindungi komponen elektronik dari risiko kontak langsung dengan air, debu, maupun gangguan fisik lainnya, mengingat alat akan dipasang berdekatan dengan instalasi perpipaan rumah tangga yang berpotensi lembap. Di dalam box, komponen PCB dipasang menggunakan spacer setinggi 1,5 cm yang berfungsi memberi jarak antara papan rangkaian dengan permukaan dasar box, sehingga mencegah

terjadinya kontak langsung yang dapat menyebabkan korsleting serta memudahkan sirkulasi udara di sekitar komponen. Pada sisi luar box, dipasang saklar on/off dan steker sebagai jalur masuk sumber daya listrik dari power supply, yang memungkinkan pengguna menyalakan dan mematikan sistem tanpa harus membuka casing.

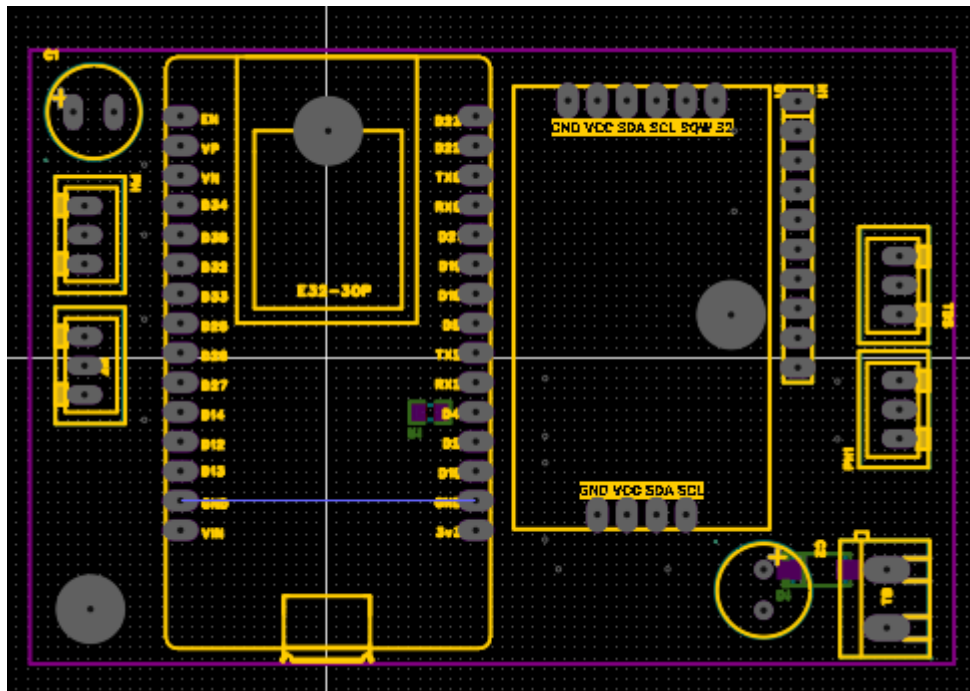
Pada bagian instalasi perpipaan, digunakan pipa berdiameter $\frac{1}{2}$ inch yang berfungsi sebagai jalur aliran air tempat sensor flow YF-S201 dipasang untuk mengukur debit air yang mengalir. Sambungan antara pipa dengan sensor menggunakan drat dalam $\frac{1}{2}$ inch, yang dipasang pada kedua ujung sensor flow agar dapat tersambung dengan pipa instalasi secara rapat. Untuk mencegah kebocoran pada setiap sambungan ulir, digunakan seal tape yang dililitkan pada bagian drat sebelum sambungan dikencangkan, sehingga aliran air tidak merembes keluar dari titik sambungan.



Gambar 4. 2 Gambar 3d bagian T Sensor

Desain mekanik bagian T sensor hasil perancangan ditunjukkan pada Gambar 4.2. Proses pembuatan struktur mekanik dilakukan menggunakan beberapa peralatan pendukung, yaitu bor dan mata bor untuk membuat lubang pemasangan pada box sesuai dengan ukuran komponen dan kabel yang akan dipasang, gerinda untuk memotong maupun merapikan bagian box yang memerlukan penyesuaian ukuran, serta pemotong pipa untuk memotong pipa sesuai dengan panjang jalur instalasi yang dibutuhkan. Penggunaan alat-alat tersebut dilakukan secara hati-hati agar hasil pemasangan memiliki presisi yang baik, sehingga tidak mengganggu kinerja maupun estetika alat secara keseluruhan.

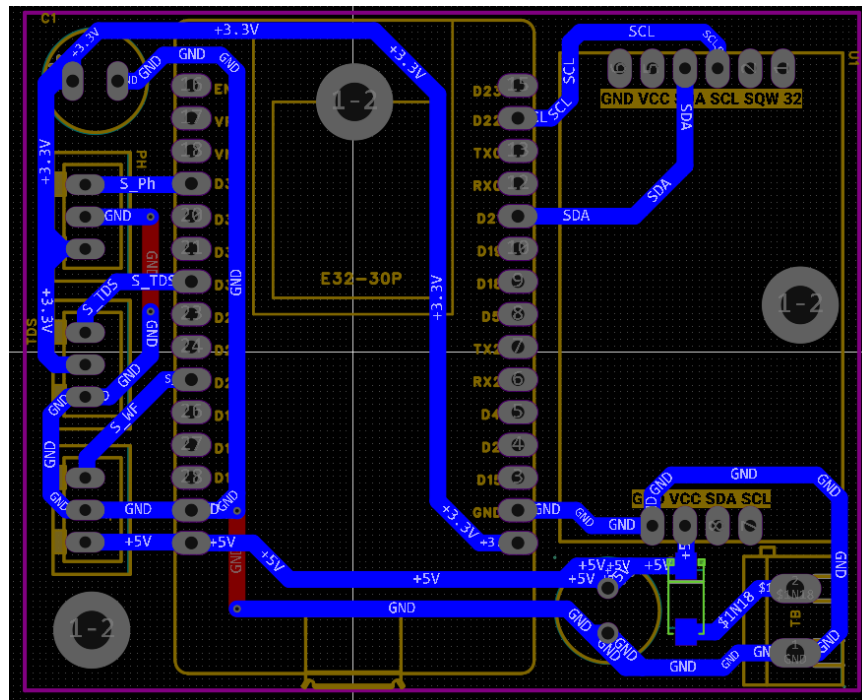
4.1.3. Perancangan Alat



Gambar 4. 3 Design PCB

Hasil perancangan PCB untuk sistem ini ditunjukkan pada Gambar 4.3. Setelah proses perakitan perangkat keras selesai dilakukan, tahapan berikutnya adalah perancangan alat secara menyeluruh yang meliputi integrasi seluruh komponen elektronik ke dalam suatu sistem yang memiliki fungsi dan struktur yang sesuai dengan tujuan penelitian. Tahap perancangan ini bertujuan untuk memastikan bahwa perangkat yang telah dikembangkan mampu beroperasi secara optimal sesuai dengan kondisi penggunaannya, serta memenuhi aspek fungsional, kemudahan pengoperasian, dan keandalan sistem secara keseluruhan. Pada tahap ini, penempatan setiap komponen dirancang dengan mempertimbangkan efisiensi penggunaan ruang, kemudahan proses perakitan, aksesibilitas saat pemeliharaan, serta kestabilan kinerja perangkat. Tata letak komponen tersebut selanjutnya dijadikan sebagai acuan dalam proses perancangan dan pencetakan Printed Circuit Board (PCB) agar koneksi antar komponen dapat tersusun secara sistematis dan sesuai dengan desain yang telah ditetapkan.

4.1.4. Perancangan PCB



Gambar 4. 4 Design Line PCB

Jalur PCB yang telah dirancang ditunjukkan pada Gambar 4.4. Setelah proses pembuatan perangkat keras dan perancangan alat selesai dilakukan, tahapan selanjutnya adalah melakukan perancangan Printed Circuit Board (PCB) sebagai media integrasi seluruh komponen elektronik yang digunakan dalam sistem monitoring kualitas air dan debit air rumah tangga. Perancangan PCB bertujuan untuk menyusun hubungan antar komponen secara lebih terstruktur, sehingga menghasilkan sistem yang lebih ringkas, stabil, dan efisien dibandingkan dengan penggunaan media perakitan sementara, seperti breadboard. Perancangan PCB juga dilakukan untuk memastikan bahwa setiap jalur koneksi listrik antar komponen tersusun sesuai dengan kebutuhan sistem, sehingga dapat meminimalkan terjadinya kesalahan koneksi serta meningkatkan keandalan perangkat dalam jangka panjang.

4.2. Pembuatan Perangkat Lunak

Pada tahap ini dilakukan pengembangan perangkat lunak yang melibatkan perancangan dan penulisan kode program menggunakan Arduino IDE sebagai lingkungan pemrograman. Pengembangan ini merupakan implementasi algoritma sistem pada ESP32 DevKit V1 agar sistem dapat berjalan sesuai dengan

perencanaan. Perangkat lunak yang dibuat mampu mengendalikan perangkat keras secara optimal serta menjalankan fungsi pengukuran, pengolahan data, dan komunikasi sesuai kebutuhan aplikasi. Secara keseluruhan, kode program yang dikembangkan terdiri dari beberapa bagian utama, yaitu inisialisasi perangkat keras, konfigurasi sensor, pengolahan data, pengendalian tampilan, serta komunikasi dengan platform cloud.

4.2.1. Konfigurasi Sensor Ph

Sensor pH pada sistem ini difungsikan untuk mengukur tingkat keasaman atau kebasaan air secara kontinu. Sensor bekerja berdasarkan prinsip elektrokimia, di mana elektroda sensor menghasilkan beda potensial (tegangan) yang berbanding linear terhadap nilai pH larutan yang diukur. Karena sinyal tegangan yang dihasilkan sensor bersifat analog dan rentan terhadap noise, pembacaan tidak dilakukan secara langsung melalui ADC internal ESP32, melainkan melalui modul ADC eksternal ADS1115 yang terhubung pada channel AIN1. ADS1115 memiliki resolusi 16-bit dengan kemampuan pengukuran tegangan hingga $\pm 4.096V$ pada pengaturan gain GAIN_ONE, sehingga menghasilkan presisi pembacaan yang jauh lebih tinggi dibandingkan ADC internal ESP32 yang hanya memiliki resolusi 12-bit.

Hasil pembacaan tegangan analog dari ADS1115 kemudian dikonversi menjadi nilai pH menggunakan persamaan linear hasil kalibrasi, yaitu:

$$pH = m \times V + b + correction$$

di mana V adalah tegangan terukur dalam volt, m adalah gradien (slope) sebesar -6.627, b adalah intercept sebesar 24.156, dan *correction* merupakan faktor koreksi tambahan yang dapat disesuaikan apabila terjadi pergeseran (drift) pembacaan di lapangan. Nilai m dan b diperoleh dari proses kalibrasi menggunakan larutan buffer pH standar (pH 4.01 dan pH 7.01), di mana pasangan data tegangan dan nilai pH yang terukur pada masing-masing larutan digunakan untuk membentuk persamaan garis linear melalui metode regresi. Implementasi fungsi pembacaan pH pada sistem ini ditunjukkan pada kode yang ditampilkan pada Gambar 4.5.

```
//PH SENSOR
float getPH(){
    float voltage = ads.computeVolts(ads.readADC_SingleEnded(1));
    float pH = (m * voltage + b) + correction;
    return pH;
}
```

Gambar 4. 5 Konfigurasi Sensor pH

Untuk menjaga kualitas air dalam rentang aman, sistem menetapkan ambang batas nilai pH antara 6.5 hingga 8.5 sesuai standar baku mutu air bersih. Apabila nilai pH yang terukur melampaui batas maksimum (>8.5) atau berada di bawah batas minimum (<6.5), sistem secara otomatis mengirimkan notifikasi peringatan kepada pengguna melalui platform Blynk menggunakan fungsi `Blynk.logEvent()`. Selain itu, nilai pH yang diperoleh pada setiap siklus pengukuran juga dikirimkan secara *real-time* ke aplikasi Blynk melalui virtual pin V2, serta disimpan secara historis ke Google Sheets untuk keperluan pemantauan dan analisis tren kualitas air dari waktu ke waktu.

4.2.2. Konfigurasi Sensor TDS

Sensor TDS (Total Dissolved Solids) digunakan untuk mengukur jumlah total zat padat terlarut dalam air, yang dinyatakan dalam satuan ppm (parts per million). Parameter ini merepresentasikan kandungan mineral, garam, dan senyawa organik maupun anorganik lainnya yang terlarut dalam air, sehingga menjadi salah satu indikator penting dalam menentukan kualitas dan kemurnian air. Sensor TDS bekerja berdasarkan prinsip konduktivitas listrik, di mana semakin tinggi konsentrasi zat terlarut dalam air, semakin besar pula kemampuan air tersebut dalam menghantarkan arus listrik. Sama seperti sensor pH, sinyal analog dari sensor TDS dibaca melalui modul ADS1115 pada channel AIN0 untuk mendapatkan hasil pembacaan yang stabil dan bebas noise.

Dalam proses pembacaan, sistem mengambil 10 sampel tegangan secara berulang dengan jeda 10 ms antar sampel, kemudian merata-ratakannya untuk meminimalkan fluktuasi sinyal akibat noise elektris. Teknik ini dikenal sebagai oversampling, dan implementasinya pada sistem ini ditunjukkan pada Gambar 4.6.

```

//TDS SENSOR
float getPPMTDS(){
    float voltage = 0.0;
    for (int i = 0; i < TDS_SAMPLE; i++) {
        voltage += ads.computeVolts(ads.readADC_SingleEnded(0));
        delay(10);
    }
    voltage /= TDS_SAMPLE;
    float compensationCoefficient = 1.0 + 0.02 * (tdsTemperature - 25.0);
    float compensationVoltage = voltage / compensationCoefficient;
    float tdsValue =
        (133.42 * compensationVoltage * compensationVoltage * compensationVoltage
        - 255.86 * compensationVoltage * compensationVoltage
        + 857.39 * compensationVoltage)
        * 0.5;
    return tdsValue;
}

```

Gambar 4. 6 Konfigurasi Sensor TDS

Karena nilai konduktivitas larutan sangat dipengaruhi oleh suhu, sistem menerapkan kompensasi suhu menggunakan data suhu aktual dari sensor DS18B20 dengan suhu referensi standar 25°C dan koefisien kompensasi 2% per derajat Celsius. Tegangan hasil kompensasi selanjutnya dikonversi menjadi nilai TDS dalam satuan ppm menggunakan persamaan polinomial orde tiga yang umum digunakan pada modul sensor TDS berbasis konduktivitas:

$$TDS (ppm) = (133.42V^3 - 255.86V^2 + 857.39V) \times 0.5$$

Sistem menetapkan ambang batas maksimum nilai TDS sebesar 1000 ppm mengacu pada standar baku mutu air bersih. Apabila nilai TDS yang terukur melampaui batas tersebut, sistem akan mengirimkan notifikasi peringatan secara otomatis kepada pengguna melalui Blynk. Nilai TDS juga dikirimkan ke virtual pin V1 pada Blynk untuk pemantauan *real-time*, serta disimpan ke Google Sheets sebagai rekaman data historis.

4.2.3. Konfigurasi Sensor Flow

Sensor flow rate pada sistem ini digunakan untuk mengukur debit aliran air yang melewati pipa dalam satuan liter per menit (L/min). Sensor yang digunakan merupakan flow meter berbasis efek Hall, yang menghasilkan sinyal pulsa digital setiap kali baling-baling di dalamnya berputar akibat dorongan aliran air. Jumlah pulsa yang dihasilkan per satuan waktu berbanding lurus dengan kecepatan aliran air, sehingga dengan menghitung frekuensi pulsa dapat diperoleh nilai debit aliran

secara akurat. Sensor terhubung ke pin digital 27 pada ESP32 dengan konfigurasi input pull-up. Berbeda dengan sensor pH dan TDS yang bersifat analog, pembacaan sensor flow rate dilakukan menggunakan mekanisme interrupt hardware pada ESP32. Setiap transisi sinyal dari HIGH ke LOW (FALLING edge) yang dihasilkan sensor akan memicu Interrupt Service Routine (ISR) bernama pulseCounter() secara otomatis, tanpa perlu menunggu proses loop() utama selesai berjalan. Hal ini penting untuk memastikan tidak ada pulsa yang terlewat meskipun mikrokontroler sedang menjalankan tugas lain secara bersamaan. Fungsi ISR dideklarasikan dengan atribut IRAM_ATTR agar dieksekusi dari Internal RAM sehingga prosesnya lebih cepat dan tidak terganggu oleh akses flash memory. Potongan program konfigurasi pembacaan sensor flow pada ESP32 ditunjukkan pada Gambar 4.7.

```
//FLOW SENSOR
void setupFlow(){
    pinMode(FLOW_PIN, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(FLOW_PIN), pulseCounter, FALLING);
}
void IRAM_ATTR pulseCounter(){
    pulseCount++;
}
```

Gambar 4. 7 Konfigurasi Sensor Flow

Penghitungan nilai debit aliran aktual dilakukan pada setiap siklus polling dalam fungsi loop(), menggunakan rumus berikut:

$$\text{Flow Rate (L/min)} = \frac{\left(\frac{1000}{t_{\text{polling}}}\right) \times \text{pulsa}}{\text{faktor kalibrasi}}$$

di mana t_{polling} adalah durasi aktual antar siklus polling dalam milidetik, sehingga perhitungan tetap akurat meskipun terjadi sedikit penyimpangan waktu dari interval yang direncanakan. Faktor kalibrasi yang digunakan adalah 4.5, sesuai dengan spesifikasi sensor flow meter yang digunakan. Sistem menetapkan ambang batas maksimum debit aliran sebesar 20 L/min, dan apabila nilai tersebut terlampaui, notifikasi peringatan akan dikirimkan ke pengguna melalui Blynk. Data

flow rate juga dikirimkan ke virtual pin V3 pada Blynk serta disimpan ke Google Sheets pada setiap siklus pengukuran.

4.2.4. Konfigurasi Sensor DS18B20

Sensor suhu yang digunakan pada sistem ini adalah DS18B20, yaitu sensor suhu digital yang berkomunikasi menggunakan protokol 1-Wire buatan Dallas Semiconductor. Protokol 1-Wire memungkinkan sensor untuk mengirimkan data suhu dalam format digital langsung ke mikrokontroler hanya melalui satu pin data, sehingga meminimalkan jumlah pin yang digunakan serta terbebas dari masalah noise yang umum terjadi pada sensor suhu analog. Sensor terhubung ke pin 4 pada ESP32, dengan dua library pendukung yang digunakan, yaitu OneWire untuk mengelola komunikasi protokol 1-Wire pada tingkat rendah (low-level), dan DallasTemperature untuk menyediakan fungsi-fungsi pembacaan suhu pada tingkat tinggi (high-level) yang lebih mudah digunakan.

Proses pembacaan suhu dilakukan melalui dua tahap. Pertama, sistem mengirimkan perintah konversi (`requestTemperatures()`) kepada sensor untuk memulai proses pengukuran suhu secara internal. Kemudian, hasil pengukuran diambil menggunakan fungsi `getTempCByIndex(0)` yang mengembalikan nilai suhu dalam satuan derajat Celsius dari sensor pertama yang terdeteksi pada bus 1-Wire. Penggunaan index memungkinkan sistem untuk dikembangkan dengan menambahkan beberapa sensor DS18B20 sekaligus pada satu bus yang sama di masa mendatang, tanpa memerlukan perubahan wiring yang signifikan. Mengingat peran suhu yang sangat penting sebagai parameter kompensasi dalam perhitungan TDS, sistem dilengkapi dengan mekanisme penanganan kesalahan (error handling) yang cukup ketat. Nilai -127.0°C merupakan nilai default yang dikembalikan library DallasTemperature ketika sensor gagal terdeteksi atau terjadi gangguan komunikasi, misalnya akibat kabel yang tidak terpasang dengan baik. Apabila nilai ini terdeteksi, sistem akan melakukan percobaan ulang pembacaan hingga 5 kali dengan jeda 200 ms, sebelum akhirnya memutuskan untuk melewati (skip) seluruh siklus polling pada iterasi yang ditampilkan pada potongan program konfigurasi pembacaan sensor DS18B20 pada Gambar 4.8.

```
//DS18B20
void setupDS18B20(){
    Serial.println("Preparing DS18B20 Sensor");
    sensors.begin();
}
float getTemperature(){
    sensors.requestTemperatures();
    return sensors.getTempCByIndex(0);
}
```

Gambar 4. 8 Konfigurasi Sensor DS18B20

Mekanisme ini memastikan bahwa tidak ada data yang tidak valid ikut tersimpan ke Google Sheets maupun dikirimkan ke Blynk, sehingga integritas data monitoring tetap terjaga. Nilai suhu yang valid selanjutnya digunakan untuk tiga keperluan sekaligus: ditampilkan secara real-time ke virtual pin V0 pada Blynk, disimpan ke Google Sheets sebagai data historis, serta digunakan sebagai nilai `tdsTemperature` pada proses kompensasi suhu dalam perhitungan sensor TDS pada siklus polling yang sama.

4.2.5. Konfigurasi Spreadsheet

Google Sheets digunakan pada sistem ini sebagai media penyimpanan data historis hasil pengukuran kualitas air secara cloud-based. Setiap data yang berhasil diukur pada satu siklus polling, mencakup waktu pengukuran, suhu, TDS, pH, dan flow rate, akan secara otomatis dikirimkan dan disimpan ke dalam sebuah spreadsheet Google Sheets melalui mekanisme Google Sheets API. Pendekatan ini dipilih karena Google Sheets menyediakan penyimpanan data yang mudah diakses, dapat divisualisasikan secara langsung, serta memungkinkan analisis data lebih lanjut tanpa memerlukan perangkat lunak tambahan.

Integrasi dengan Google Sheets pada sistem ini menggunakan library `ESP_Google_Sheet_Client`, yang memanfaatkan mekanisme autentikasi Google Service Account berbasis OAuth 2.0. Autentikasi ini memerlukan tiga parameter identitas yang didefinisikan pada kode program, yaitu `PROJECT_ID` sebagai identitas proyek Google Cloud, `CLIENT_EMAIL` sebagai alamat email akun

layanan (service account) yang telah diberikan izin akses ke spreadsheet, serta PRIVATE_KEY sebagai kunci privat RSA yang digunakan untuk menandatangani token autentikasi secara kriptografis. Ketiga parameter tersebut diperoleh dari Google Cloud Console saat membuat service account dan mengaktifkan Google Sheets API pada proyek yang bersangkutan. Proses inisialisasi koneksi ke Google Sheets dilakukan pada fungsi GsheetSetup() ditunjukkan pada potongan program konfigurasi pada Gambar 4.9.

```
void GsheetSetup(){
  GSheet.printf("ESP Google Sheet Client v%s\n\n", ESP_GOOGLE_SHEET_CLIENT_VERSION);
  GSheet.setTokenCallback(tokenStatusCallback);
  GSheet.setPrereshSeconds(10 * 60);
  GSheet.begin(CLIENT_EMAIL, PROJECT_ID, PRIVATE_KEY);
  delay(1000);
}
```

Gambar 4. 9 Konfigurasi Google Spreadsheet

Fungsi setPrereshSeconds(10*60) mengatur sistem untuk memperbarui token autentikasi secara otomatis setiap 10 menit sebelum token tersebut kedaluwarsa, sehingga koneksi ke Google Sheets dapat terjaga secara kontinu tanpa perlu intervensi manual. Sementara itu, fungsi setTokenCallback(tokenStatusCallback) mendaftarkan sebuah fungsi callback yang akan dipanggil setiap kali terjadi perubahan status token, baik berhasil maupun gagal, sehingga proses autentikasi dapat dipantau melalui Serial Monitor untuk keperluan debugging. Proses pengiriman data ke spreadsheet dilakukan melalui fungsi sendToGoogleSheets() yang dipanggil pada setiap akhir siklus polling. Sebelum data dikirimkan, sistem terlebih dahulu membaca waktu aktual dari modul RTC DS3231 untuk digunakan sebagai timestamp data, sehingga setiap baris data yang tersimpan memiliki keterangan waktu yang akurat. Data kemudian dikemas dalam format FirebaseJson dengan struktur kolom (COLUMNS) yang mencakup timestamp, suhu, TDS, pH, dan flow rate, lalu dikirimkan ke spreadsheet menggunakan metode append pada rentang Sheet1!A2. Lanjutan konfigurasi integrasi dengan Google Spreadsheet ditunjukkan pada Gambar 4.10.

```

valueRange.add("majorDimension", "COLUMNS");
valueRange.set("values/[0]/[0]", timeStr);
valueRange.set("values/[1]/[0]", ds18Temperature);
valueRange.set("values/[2]/[0]", tdsPPMValue);
valueRange.set("values/[3]/[0]", PHValue);
valueRange.set("values/[4]/[0]", flowRate);
bool success = GSheet.values.append(&response /* returned response */, spreadsheetId /* spreadsheet Id to append */, "Sheet1!A2" /* range to append
if (success){
    response.toString(Serial, true);
    valueRange.clear();
    Serial.println("AFTER APPEND");
}

```

Gambar 4. 10 Konfigurasi Google Spreadsheet

Penggunaan metode append memastikan bahwa setiap data baru ditambahkan sebagai baris baru di bawah data sebelumnya secara otomatis, tanpa menimpa data yang sudah ada. Setelah data berhasil dikirimkan, objek valueRange dikosongkan kembali (valueRange.clear()) untuk mempersiapkan pengiriman data pada siklus polling berikutnya. Apabila pengiriman gagal, sistem akan mencetak pesan kesalahan melalui Serial Monitor melalui fungsi GSheet.errorReason() untuk memudahkan proses identifikasi dan debugging masalah koneksi.

4.2.6. Konfigurasi Blynk

```

//BLYNK
void setupBlynk(){
    Blynk.config(BLYNK_AUTH_TOKEN);
    Serial.print("Connect Blynk");
    while(!Blynk.connect(1000)){
        Serial.print(".");
    }
    Serial.println();
    Serial.println("Blynk Connected");
}

void sendToBlynk(){
    Blynk.virtualWrite(V0, ds18Temperature);
    Blynk.virtualWrite(V1, tdsPPMValue);
    Blynk.virtualWrite(V2, PHValue);
    Blynk.virtualWrite(V3, flowRate);
}

void checkThreshold(){
    if (tdsPPMValue > TDS_MAX) Blynk.logEvent("tds_tinggi", String("TDS tinggi: ") + tdsPPMValue + " ppm");
    if (PHValue > PH_MAX) Blynk.logEvent("ph_tinggi", String("pH tinggi: ") + PHValue);
    else if (PHValue > PH_MIN) Blynk.logEvent("ph_rendah", String("pH rendah: ") + PHValue);
    if (flowRate > FLOW_MAX) Blynk.logEvent("flow_tinggi", String("Flow Rate tinggi: ") + flowRate + " L/min");
}

```

Gambar 4. 11 Konfigurasi Blynk

Tahapan konfigurasi Blynk pada sistem ini ditunjukkan pada Gambar 4.11. Aplikasi Blynk digunakan sebagai platform pemantauan data secara real-time berbasis Internet of Things (IoT). Melalui aplikasi Blynk yang dapat diakses dari perangkat smartphone, pengguna dapat memantau seluruh parameter kualitas air secara langsung, menerima notifikasi peringatan otomatis apabila terdapat parameter yang melampaui ambang batas yang telah ditetapkan, serta memantau kondisi sistem dari mana saja selama terhubung dengan internet. Integrasi Blynk pada sistem ini menggunakan library BlynkSimpleEsp32 yang dirancang khusus

untuk platform ESP32. Proses inisialisasi koneksi Blynk dilakukan pada fungsi `setupBlynk()` yang dipanggil saat sistem pertama kali dinyalakan. Gambar 4.12 menunjukkan sistem akan terus mencoba terhubung ke server Blynk dengan interval 1 detik hingga koneksi berhasil terbentuk, memastikan sistem tidak melanjutkan proses inisialisasi sebelum koneksi Blynk benar-benar tersedia.

```
//BLYNK
void setupBlynk(){
  Blynk.config(BLYNK_AUTH_TOKEN);
  Serial.print("Connect Blynk");
  while(!Blynk.connect(1000)){
    Serial.print(".");
  }
  Serial.println();
  Serial.println("Blynk Connected");
}
```

Gambar 4. 12 Konfigurasi Blynk

Pengiriman data sensor ke aplikasi Blynk dilakukan melalui mekanisme Virtual Pin, yang ditunjukkan pada Gambar 4.13, yaitu pin virtual yang dipetakan ke masing-masing parameter pengukuran pada dashboard Blynk. Setiap siklus polling, fungsi `sendToBlynk()` dipanggil untuk mengirimkan seluruh data sensor secara bersamaan:

```
void sendToBlynk(){
  Blynk.virtualWrite(V0, ds18Temperature);
  Blynk.virtualWrite(V1, tdsPPMValue);
  Blynk.virtualWrite(V2, PHValue);
  Blynk.virtualWrite(V3, flowRate);
}
```

Gambar 4. 13 Konfigurasi Blynk

Pemetaan virtual pin yang digunakan pada sistem ini adalah: V0 untuk data suhu, V1 untuk data TDS, V2 untuk data pH, dan V3 untuk data flow rate. Setiap virtual pin ini terhubung ke widget tampilan yang bersesuaian pada dashboard aplikasi Blynk, sehingga pengguna dapat memantau perubahan nilai setiap parameter secara visual dan real-time. Selain pengiriman data, Blynk juga difungsikan sebagai sistem notifikasi peringatan otomatis melalui fungsi `checkThreshold()`. Apabila nilai suatu parameter melampaui ambang batas yang telah ditetapkan, sistem akan mengirimkan *event* notifikasi ke aplikasi Blynk menggunakan fungsi `Blynk.logEvent()` disertai pesan keterangan nilai yang memicu peringatan tersebut. Tabel 4-3 merangkum seluruh kondisi peringatan yang dikonfigurasi pada sistem:

Tabel 4- 3 Kondisi Peringatan pada Sensor pH

Parameter	Kondisi Peringatan	Event ID	Pesan Notifikasi
TDS	> 1000 ppm	tds_tinggi	TDS tinggi: [nilai] ppm
pH	> 8.5	ph_tinggi	pH tinggi: [nilai]
pH	< 6.5	ph_rendah	pH rendah: [nilai]
Flow Rate	> 20 L/min	flow_tinggi	Flow Rate tinggi: [nilai] L/min

Agar koneksi Blynk tetap aktif dan responsif selama sistem berjalan, fungsi Blynk.run() dipanggil secara kontinu di awal setiap iterasi fungsi loop(). Fungsi ini bertugas menangani semua proses komunikasi latar belakang (background communication) antara ESP32 dan server Blynk, termasuk menjaga koneksi tetap hidup (keep-alive), memproses data masuk dari server, serta mengirimkan antrian data yang belum terkirim, sehingga sistem pemantauan dapat berjalan secara stabil dan berkesinambungan.

4.2.7. Kalibrasi Sensor pH

```

#define SENSOR_PIN A0

// ===== ISI DENGAN HASIL KALIBRASI ANDA =====
const float VOLTAGE_PH7 = 2.535; // contoh nilai, GANTI sesuai hasil Anda
const float VOLTAGE_PH4 = 3.025; // contoh nilai, GANTI sesuai hasil Anda
// =====

const int JUMLAH_SAMPLING = 20;
const int JEDA_SAMPLING = 30;

// Dihitung otomatis dari 2 titik kalibrasi di atas
float slope, intercept;

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

  // Hitung persamaan garis: (pH7, V_ph7) dan (pH4, V_ph4)
  slope = (7.0 - 4.0) / (VOLTAGE_PH7 - VOLTAGE_PH4);
  intercept = 7.0 - slope * VOLTAGE_PH7;

  Serial.println("=== Sensor PH-4582C Siap Digunakan ===");
  Serial.print("Slope    = "); Serial.println(slope, 4);
  Serial.print("Intercept = "); Serial.println(intercept, 4);
  Serial.println();
}

void loop() {
  long totalADC = 0;

  for (int i = 0; i < JUMLAH_SAMPLING; i++) {
    totalADC += analogRead(SENSOR_PIN);
    delay(JEDA_SAMPLING);
  }

  float rataADC = (float)totalADC / JUMLAH_SAMPLING;
  float voltage = rataADC * (5.0 / 1024.0);
  float pH = slope * voltage + intercept;

  Serial.print("Voltage: ");
  Serial.print(voltage, 3);
  Serial.print(" V | pH: ");
  Serial.println(pH, 2);

  delay(1000);
}

```

Gambar 4. 14 Kalibrasi Sensor pH

Proses kalibrasi sensor pH yang ditunjukkan pada Gambar 4.14, kode program ini didasarkan pada metode kalibrasi dua titik (two-point calibration), yaitu pendekatan yang umum digunakan dalam pengukuran pH karena hubungan antara tegangan keluaran sensor dengan nilai pH bersifat linear pada rentang tertentu. Pada bagian awal program, dua variabel konstanta yaitu VOLTAGE_PH7 dan VOLTAGE_PH4 didefinisikan untuk menyimpan nilai tegangan hasil pembacaan

sensor ketika elektroda dicelupkan ke dalam larutan buffer standar dengan pH 7,0 dan pH 4,0. Kedua larutan buffer ini dipilih sebagai titik acuan karena mewakili kondisi netral dan asam, sehingga memungkinkan penentuan garis linear yang menghubungkan nilai tegangan dengan nilai pH secara akurat.

Berdasarkan kedua titik acuan tersebut, program kemudian menghitung dua parameter utama, yaitu slope (kemiringan garis) dan intercept (titik potong garis terhadap sumbu). Nilai slope diperoleh dari selisih nilai pH kedua buffer ($7,0 - 4,0$) dibagi dengan selisih tegangan pada kedua kondisi tersebut ($VOLTAGE_PH7 - VOLTAGE_PH4$), yang secara matematis merepresentasikan besarnya perubahan nilai pH untuk setiap satuan perubahan tegangan. Sementara itu, nilai intercept dihitung dengan mengurangkan hasil kali slope dan $VOLTAGE_PH7$ dari nilai pH 7,0, sehingga diperoleh persamaan garis lurus dalam bentuk $pH = slope \times tegangan + intercept$. Persamaan inilah yang nantinya digunakan sebagai dasar konversi nilai tegangan analog yang terbaca oleh sensor menjadi nilai pH yang sesungguhnya pada tahap pembacaan data (loop program).

Selain itu, terdapat pula dua konstanta tambahan, yaitu `JUMLAH_SAMPLING` dan `JEDA_SAMPLING`, yang berfungsi untuk meningkatkan akurasi dan stabilitas pembacaan sensor. Konstanta ini menentukan jumlah pengambilan data (sampling) yang dilakukan secara berulang serta jeda waktu antar pengambilan data, sehingga nilai tegangan yang digunakan dalam perhitungan merupakan hasil rata-rata dari beberapa kali pembacaan, bukan hanya satu kali pengukuran instan yang rentan terhadap noise atau fluktuasi sinyal. Dengan demikian, keseluruhan proses kalibrasi pada bagian `setup()` ini bertujuan untuk membangun model matematis (persamaan garis) yang menjadi dasar konversi sinyal tegangan analog dari sensor PH-4502C menjadi nilai pH yang valid dan dapat diandalkan pada tahap pengukuran selanjutnya.