

BAB IV

PROSES PEMBUATAN ALAT

4.1 Pembuatan Perangkat Keras

Pembuatan perangkat keras merupakan tahapan awal dalam proses pembangunan sistem monitoring suhu dan tekanan transformator berbasis mikrokontroler ESP32. Pada tahap ini dilakukan proses perakitan seluruh komponen elektronik dan mekanik sehingga membentuk suatu sistem yang mampu melakukan pemantauan kondisi suhu dan tekanan tangki simulasi sistem secara *real-time*. Perangkat keras yang digunakan terdiri atas tangki simulasi sistem sebagai media pengujian, ESP32 sebagai pengendali utama, sensor termokopel tipe K yang dikuatkan dengan modul MAX6675 sebagai pembaca suhu, sensor tekanan sebagai pembaca tekanan internal tangki, *Relay* sebagai pengendali *Solenoid Valve*, modul *Dimmer* AC sebagai pengatur daya *Heater*, *Pressure Gauge* sebagai media penampil tekanan analog, serta LCD I2C 16x2 sebagai media penampil informasi. Seluruh komponen tersebut diintegrasikan menjadi satu sistem sehingga mampu menjalankan proses monitoring secara otomatis sesuai dengan ambang batas suhu dan tekanan yang telah ditentukan, serta mengirimkan data hasil pemantauan ke *website* Node Red dan Google *Spreadsheet* melalui jaringan Wi-Fi.

4.2 Alat dan Bahan

Alat dan bahan yang digunakan dalam pembuatan simulasi sistem monitoring suhu dan tekanan transformator terdiri dari komponen elektronik, komponen mekanik, serta alat pendukung perakitan. Komponen elektronik digunakan untuk membentuk rangkaian utama sistem, sedangkan komponen mekanik digunakan sebagai pelindung dan penempatan rangkaian maupun tangki simulasi sistem. Alat pendukung digunakan untuk proses penyolderan, pengecekan sambungan, serta pemasangan komponen pada box. Berikut adalah tabel alat dan bahan yang diperlukan.

Tabel 4.1 Daftar Alat yang Digunakan

No.	Nama Alat	Fungsi
1.	Solder	Menyolder komponen dan kabel pada PCB
2.	Timah Solder	Bahan penyambung antara kaki komponen dan jalur PCB
3.	Multimeter	Mengecek tegangan, kontinuitas, dan sambungan rangkaian
4.	Gunting	Memotong kabel atau kaki komponen
6.	Obeng	Memasang baut pada box dan terminal rangkaian
7.	Bor baterai	Membuat lubang pada desain box kontrol panel
8.	Gerinda	Memotong PCB
9.	Laptop	Membuat dan mengunggah program ke ESP32
10.	Kabel USB	Menghubungkan ESP32 dengan laptop saat pemrograman
11.	Lem Tembak	Menempelkan PCB pada box control panel.
12.	Termometer Digital	Memvalidasi data suhu

Selain alat, penelitian ini juga memerlukan berbagai bahan dan komponen elektronika sebagai penyusun utama sistem monitoring suhu dan tekanan transformator. Bahan yang digunakan dipilih berdasarkan fungsi, spesifikasi, serta kesesuaiannya dengan kebutuhan sistem agar seluruh komponen dapat bekerja secara terintegrasi. Daftar bahan yang digunakan dalam penelitian ini disajikan pada Tabel 4.2.

Tabel 4.2 Daftar Bahan yang Digunakan

No.	Nama Bahan	Jumlah
1.	ESP32	1 buah
2.	Sensor Termokopel Tipe K	1 buah
3.	Sensor Tekanan	1 buah
5.	<i>Dimmer AC</i>	1 buah
6.	<i>Relay 12 V</i>	2 buah

No.	Nama Bahan	Jumlah
7.	<i>Solenoid Valve</i>	1 buah
8.	Gauge Tekanan	1 buah
9.	<i>Heater</i> 220 VA	1 buah
10.	LCD	1 buah
11.	<i>Buzzer</i>	1 buah
12.	JST 3 Pin	3 buah
13.	JST 2 Pin	4 buah
13.	PCB	1 buah
14.	Kabel Jumper	10 buah
15.	Pin Header	2 buah
16.	Box Panel	1 buah
17.	Kabel Micro to Type C	1 buah
18.	Kabel AWG Merah	3 meter
19.	Kabel AWG Hitam	3 meter
20.	Modul MAX6675	1 buah
21.	Lem Stik	1 buah
22.	Push Button	2 buah
23.	Switch	2 buah
24.	Pipa <i>stainless</i> 1 inch 20 cm	1 buah
25.	Pipa <i>stainless</i> 1 inch 10 cm	1 buah
26.	Pipa <i>stainless</i> 42 mm 4 cm	1 buah
27.	Pipa <i>stainless</i> 26 cm x 16.5 cm	1 buah
28.	Kran nepel ½ inch	1 buah
29.	Kran nepel ¼ inch	1 buah
30.	Nepel drat ½ inch	3 buah
31.	Nepel drat ¼ inch	3 buah
32.	Nepel drat ukuran <i>Heater</i>	1 buah
33.	Selotip pipa	1 buah

4.3 Pembuatan Perangkat Elektrik

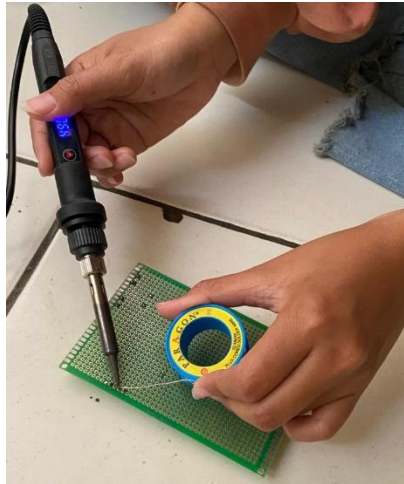
Pembuatan perangkat elektrik dilakukan dengan merangkai seluruh komponen elektronika sesuai dengan rancangan sistem yang telah dibuat. Komponen yang dirangkai meliputi mikrokontroler ESP32, sensor termokopel tipe K, modul MAX6675, sensor tekanan, *Relay*, modul *Dimmer* AC, LCD I2C 16x2, *Pressure Gauge*, serta elemen pemanas (*Heater*). Seluruh komponen dihubungkan sesuai dengan fungsi masing-masing sehingga membentuk sistem pengendalian dan *monitoring* pada alat pemantauan suhu dan tekanan transformator. Adapun langkah-langkah pembuatan perangkat elektrik adalah sebagai berikut:

1. Perancangan Skematik dan Tata Letak Komponen

Perancangan skematik ini bertujuan untuk menentukan jalur koneksi antar pin komponen, seperti pin SPI pada modul MAX6675, pin analog pada sensor tekanan, pin I2C pada LCD, serta pin *digital output* pada *Relay*, sebelum proses perakitan fisik dilakukan. Setelah skematik selesai dibuat, dilanjutkan dengan perancangan tata letak (*layout*) penempatan komponen pada PCB bolong (*perfboard*). Penempatan komponen disusun mempertimbangkan jarak antar komponen, kemudahan jalur kabel penghubung, serta efisiensi ruang pada box panel kontrol, sehingga proses penyolderan dan penyambungan kabel dapat dilakukan dengan lebih rapi dan risiko kesalahan sambungan dapat diminimalkan.

2. Proses Penyolderan PCB

Setelah tata letak komponen ditentukan, tahap selanjutnya adalah pemasangan dan penyolderan komponen pada PCB bolong. Kaki-kaki komponen dimasukkan ke dalam lubang PCB sesuai posisi yang telah dirancang, kemudian disolder pada bagian bawah PCB menggunakan solder dan timah untuk membentuk sambungan yang kuat dan menghantarkan listrik dengan baik. Proses penyolderan dilakukan secara bertahap guna memudahkan proses pengerjaan dan mengurangi risiko komponen saling bertabrakan.



Gambar 4. 1 Proses Penyolderan PCB

3. Pemasangan dan Penyambungan Kabel

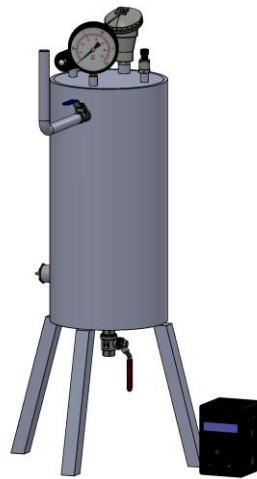
Setelah proses perakitan dan penyolderan komponen pada PCB selesai dilakukan, tahap selanjutnya adalah integrasi sistem pada sisi bagian kontrol. Pada tahap ini, PCB dihubungkan dengan berbagai komponen dan perangkat pendukung yang digunakan pada alat monitoring suhu dan tekanan transformator sesuai dengan rancangan rangkaian. Komponen yang diintegrasikan meliputi sensor termokopel tipe K, modul MAX6675, sensor tekanan, *Relay* sebagai bagian dari pengendali *Solenoid Valve*, modul *Dimmer AC* sebagai pengatur daya *Heater*, LCD I2C 16x2 sebagai media tampilan, serta sumber tegangan. LCD dihubungkan melalui komunikasi I2C menggunakan konektor yang sesuai. Seluruh hubungan antarsistem disesuaikan dengan rancangan kelistrikan yang telah dibuat.

Tahap terakhir dilakukan dengan mengaktifkan sistem untuk melakukan pemeriksaan awal terhadap fungsi setiap komponen. Pengujian dilakukan untuk memastikan perangkat dapat beroperasi sesuai dengan fungsi yang telah dirancang sebelum sistem digunakan pada proses monitoring suhu dan tekanan transformator. Setelah perangkat elektrik dinyatakan dapat beroperasi dengan baik, dilanjutkan proses pemrograman dan pengujian sistem secara keseluruhan.

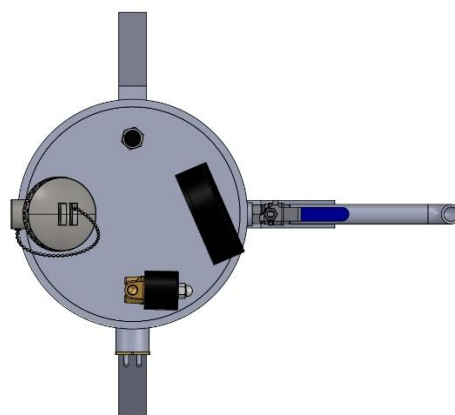
4.4 Pembuatan Perangkat Mekanik

1. Pembuatan Tangki Minyak

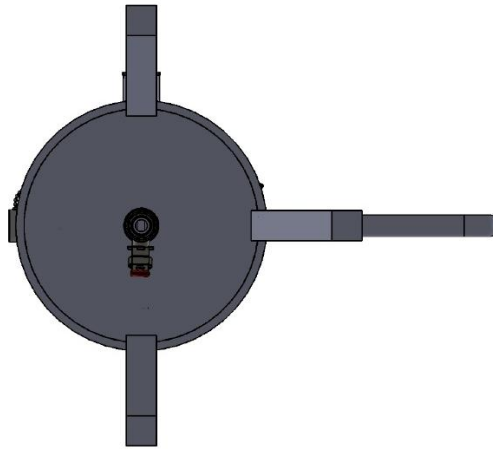
Pembuatan perangkat mekanik pada sistem monitoring suhu dan tekanan transformator dilakukan melalui beberapa tahapan, mulai dari perancangan desain tiga dimensi (3D), persiapan tangki simulasi sistem pembuatan dan pemasangan box panel kontrol, hingga proses perakitan seluruh komponen menjadi satu kesatuan sistem. Perancangan dilakukan terlebih dahulu untuk menentukan dimensi, posisi pemasangan, serta kebutuhan bukaan (*port*) pada bagian tangki dan box kontrol panel. Dengan adanya desain awal, proses fabrikasi dan perakitan dapat dilakukan berdasarkan ukuran dan posisi komponen yang telah ditentukan.



Gambar 4.2 Tangki Minyak dan Box Panel Kontrol

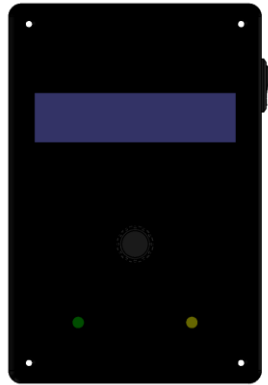


Gambar 4.3 Tangki Minyak Tampak Atas



Gambar 4.4 Tangki Minyak Tampak Bawah

Pada bagian atas tangki terdapat sensor termokopel tipe K, sensor tekanan, *Pressure Gauge*, serta *Solenoid Valve* yang berfungsi untuk memantau kondisi suhu dan tekanan internal tangki, sekaligus menjadi jalur pelepas tekanan berlebih secara otomatis apabila kondisi tekanan telah melampaui ambang batas yang ditentukan. Pada bagian samping tangki terdapat *port* pemasangan *Heater* yang berfungsi sebagai elemen pemanas untuk mensimulasikan kenaikan suhu di dalam tangki. Di bagian atas terdapat cerobong yang berfungsi sebagai *inlet*, yaitu jalur pemasukan cairan ke dalam tangki, sedangkan pada bagian bawah tangki terdapat *outlet* yang berfungsi sebagai jalur pengeluaran atau pengurasan cairan dari dalam tangki. Tangki ditopang oleh tiga kaki penyangga yang berfungsi menjaga tangki tetap berdiri stabil serta memberikan jarak antara bagian bawah tangki dengan permukaan tempat alat diletakkan, sekaligus memberikan ruang untuk penempatan box panel kontrol di sisi tangki.



Gambar 4.5 Box Panel Kontrol Tampak Depan



Gambar 4.6 Box Kontrol Panel Tampak Samping

Box panel kontrol berfungsi sebagai wadah komponen antarmuka pengguna sekaligus tempat pemasangan rangkaian kontrol sistem monitoring suhu dan tekanan transformator. Pada bagian depan *box* panel terdapat beberapa komponen utama. LCD I2C 16x2 ditempatkan pada bagian atas panel untuk menampilkan informasi suhu, tekanan, dan status kondisi sistem secara langsung kepada pengguna. Di bagian tengah panel terdapat potensiometer modul *Dimmer* AC yang berfungsi untuk mengatur besarnya daya yang disalurkan ke *Heater*, sehingga pengguna dapat mengendalikan laju kenaikan suhu pada tangki simulasi sistem secara manual.

Pada bagian bawah panel terdapat tiga tombol yang berfungsi sebagai kontrol manual sistem, yaitu saklar (*switch*) hijau yang berfungsi untuk memilih mode pengoperasian sistem antara mode manual dan mode otomatis, serta dua tombol tekan (*push button*) berwarna hitam dan kuning yang masing-masing berfungsi untuk mengaktifkan *Heater* dan *Solenoid Valve* secara manual ketika sistem berada dalam mode manual.

Selain itu, pada bagian samping box panel terdapat saklar utama (*main switch*) yang berfungsi sebagai sakelar *on/off* untuk menghidupkan atau mematikan seluruh catu daya sistem secara keseluruhan.

2. Pemeriksaan Akhir dan Hasil Pembuatan Perangkat Mekanik

Pemeriksaan akhir dilakukan setelah seluruh proses pemasangan dan modifikasi pada tangki simulasi sistem selesai. Pemeriksaan ini dilakukan untuk memastikan setiap bagian yang telah dibuat dan dipasang berada pada posisi yang sesuai rancangan. Bagian yang diperiksa meliputi box panel kontrol beserta LCD, potensiometer *Dimmer AC*, *switch* mode manual/otomatis, dan *push button*, *Heater* dan *solenoid* yang terdapat pada bagian depan panel, serta bagian tangki yang mengalami perubahan berupa pemasangan sensor, *Pressure Gauge*, dan *solenoid valve* dan *Heater*.

Hasil akhir pembuatan perangkat mekanik berupa satu unit simulasi sistem tangki minyak transformator yang telah disesuaikan dengan kebutuhan sistem monitoring suhu dan tekanan. Sistem kendali dan pemantauan berupa LCD, *switch*, dan *push button* dimanfaatkan sebagai antarmuka pengoperasian sistem, kemudian penempatan komponen diatas tangki dan pemanas disesuaikan dengan jalur kabel yang menuju panel kontrol. Elemen pemanas dipasang pada bagian samping tangki, sedangkan komponen lainnya berupa box panel kontrol, sensor termokopel tipe K, sensor tekanan, dan *Solenoid Valve* dipasang sesuai dengan rancangan.

Dengan selesainya tahap pemeriksaan tersebut, bagian mekanik alat telah terpasang dan terintegrasi dengan perangkat elektrik yang dibuat pada tahap sebelumnya. Kondisi fisik alat selanjutnya digunakan sebagai dasar untuk melanjutkan proses pembuatan perangkat lunak dan tahap pengujian sistem monitoring suhu dan tekanan transformator.

4.5 Pembuatan Perangkat Lunak

Setelah proses pembuatan dan perakitan perangkat keras selesai, tahap selanjutnya adalah pembuatan perangkat lunak (*software*) yang berfungsi sebagai pusat pengendalian sistem monitoring suhu dan tekanan tangki transformator. Pemrograman mikrokontroler dilakukan menggunakan *software* Arduino IDE dengan *board* ESP32 Dev Module. *Board* tersebut dipilih karena memiliki

konektivitas Wi-Fi serta jumlah GPIO yang memadai untuk menghubungkan sensor, aktuator, dan komponen pendukung lainnya. Program dirancang untuk mengatur proses pembacaan sensor termokopel tipe K yang terhubung melalui modul MAX6675 sebagai pengukur suhu, serta sensor tekanan sebagai pengukur tekanan internal tangki selama proses monitoring berlangsung.

Dalam pembuatan program digunakan beberapa *library* yang disesuaikan dengan komponen pada sistem, antara lain *WiFi.h* untuk menghubungkan ESP32 dengan jaringan internet, *library HTTP/WebServer* untuk komunikasi dan pengiriman data ke *website monitoring*, *SPI.h* untuk komunikasi antara ESP32 dengan modul MAX6675, serta *library* pendukung untuk pengiriman data ke *Google Spreadsheet*. Data dari sensor kemudian diproses oleh ESP32 untuk menentukan status kondisi tangki (aman, *warning*, atau bahaya) dan mengendalikan LED indikator, *buzzer*, *Relay*, serta *Solenoid Valve* yang digunakan sebagai respons otomatis terhadap kondisi *overpressure*, sementara modul *Dimmer AC* digunakan untuk mengatur daya *Heater* yang menghasilkan simulasi kenaikan suhu pada tangki.

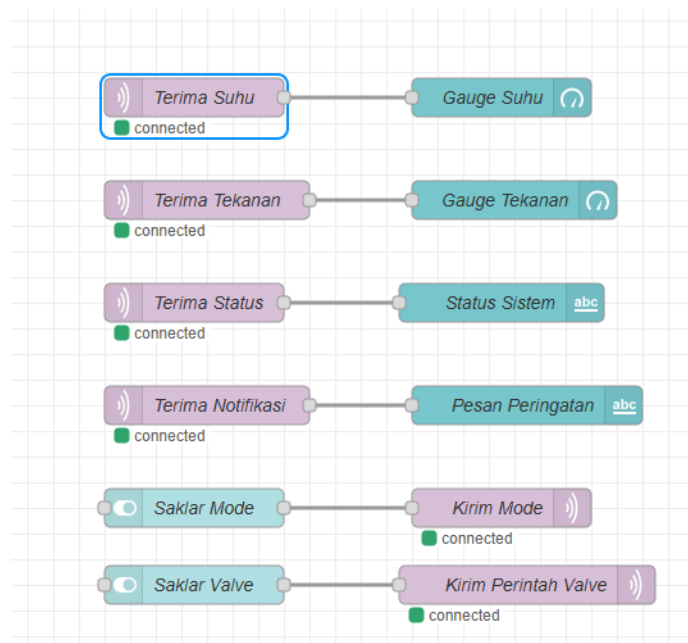
Ketika sensor tekanan dan termokopel mendeteksi suhu dan tekanan yang telah melewati ambang batas yang ditentukan, ESP32 menjalankan proses respons otomatis dengan mengaktifkan *buzzer* sebagai peringatan, serta mengaktifkan *Relay* untuk membuka *Solenoid Valve* apabila kondisi telah mencapai batas bahaya. Selama proses berlangsung, sensor termokopel tipe K dan sensor tekanan secara terus-menerus mengukur kondisi tangki dan memberikan data kepada ESP32 sebagai parameter pengendalian. Data suhu, tekanan, dan status kondisi sistem selanjutnya dikirimkan ke *website monitoring* dan *Google Spreadsheet* sehingga kondisi alat dapat dipantau secara *real-time*. Dengan demikian, program yang dibuat mengintegrasikan proses monitoring, pengendalian suhu dan tekanan, serta respons otomatis dalam satu sistem berbasis ESP32.

4.5.1 Pembuatan Template dan Konfigurasi Node Red

Node-RED digunakan sebagai perangkat lunak pengolah aliran data pada sistem monitoring suhu dan tekanan internal transformator daya. Node-RED berfungsi sebagai penghubung antara perangkat ESP32 dengan antarmuka website

melalui protokol MQTT. Pada tahap ini dilakukan pembuatan template flow serta konfigurasi node yang digunakan untuk menerima data monitoring, meneruskan informasi status sistem, menampilkan pesan peringatan, serta mengirimkan perintah kendali dari website menuju ESP32.

Template flow Node-RED dirancang berdasarkan kebutuhan sistem, yaitu memisahkan jalur data berdasarkan jenis informasi yang dikirimkan. Data hasil pembacaan suhu dan tekanan memiliki jalur masing-masing, sedangkan informasi status sistem dan notifikasi juga dipisahkan agar dapat ditampilkan pada komponen antarmuka yang sesuai. Selain jalur penerimaan data, dibuat pula jalur keluaran yang digunakan untuk mengirimkan perintah mode operasi dan perintah valve dari website menuju ESP32.



Gambar 4.7 Program Node Red

Berdasarkan Gambar 4.7 terlihat bahwa sistem memiliki empat jalur penerimaan data dan dua jalur pengiriman perintah. Jalur penerimaan terdiri atas *node* Terima Suhu, Terima Tekanan, Terima Status, dan Terima Notifikasi. Masing-masing *node* menerima data melalui MQTT dan meneruskannya ke bagian tampilan website. Sementara itu, jalur kendali terdiri atas *node* Saklar Mode dan Saklar Valve yang digunakan sebagai masukan dari pengguna untuk menghasilkan perintah menuju ESP32 melalui MQTT.

4.5.2 Konfigurasi MQTT pada Node-RED

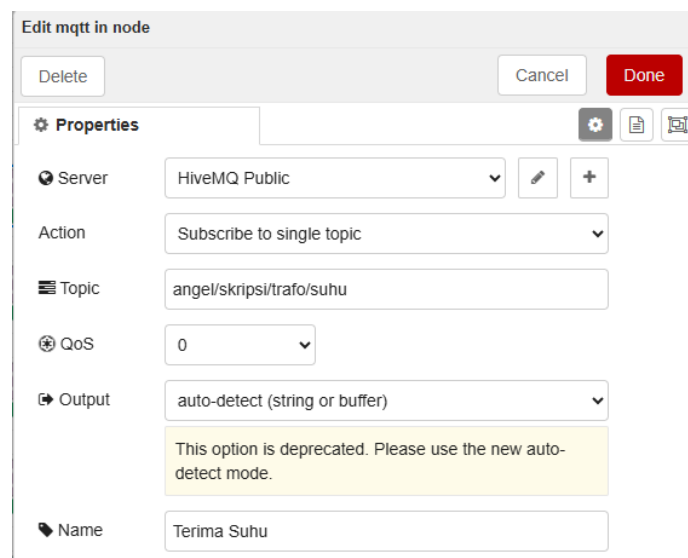
Konfigurasi MQTT dilakukan untuk menghubungkan Node-RED dengan *broker* yang digunakan sebagai media pertukaran data antara ESP32 dan website. Pada sistem ini digunakan HiveMQ Public sebagai MQTT broker. Konfigurasi dilakukan melalui *node* MQTT dengan menentukan server, *topic*, QoS, dan format keluaran data sesuai dengan kebutuhan sistem.

Tabel 4.3 Konfigurasi MQTT pada Node-RED

Parameter	Konfigurasi
Server	HiveMQ Public
Action	Subscribe to single topic
Topic	angel/skripsi/trafo/suhu
QoS	0
Output	Auto-detect
Name	Terima Suhu

Pada konfigurasi tersebut, Node-RED menggunakan mode Subscribe to single topic untuk menerima pesan dari satu *topic* MQTT. *Topic* angel/skripsi/trafo/suhu digunakan sebagai jalur komunikasi data suhu dari ESP32 menuju Node-RED. Nilai *QoS* 0 digunakan pada komunikasi MQTT, sedangkan format keluaran diatur menggunakan *Auto-detect* agar data yang diterima dapat dikenali sesuai format pesan yang dikirimkan. *Node* diberi nama Terima Suhu untuk mempermudah identifikasi fungsi pada *flow*.

Konfigurasi MQTT yang sama diterapkan pada *node* penerimaan parameter lainnya dengan menyesuaikan *topic* yang digunakan. Dengan konfigurasi tersebut, Node-RED dapat menerima data dari ESP32 melalui MQTT sesuai dengan parameter yang telah ditentukan pada sistem.



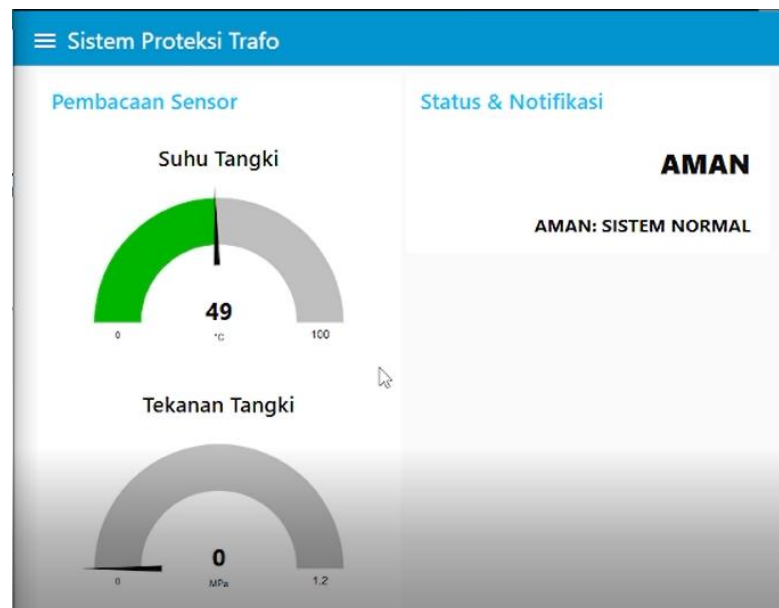
Gambar 4.8 Konfigurasi Node MQTT

Berdasarkan Gambar 4.8, konfigurasi *node* MQTT Terima Suhu menggunakan HiveMQ Public sebagai server MQTT dengan mode *Subscribe to single topic*. *Node* tersebut berlangganan pada *topic* *angel/skripsi/trafo/suhu* dengan nilai *QoS* 0. Data yang diterima kemudian diteruskan ke proses berikutnya pada *flow* Node-RED untuk ditampilkan sebagai parameter suhu pada sistem monitoring.

4.5.3 Perancangan Tampilan Monitoring

Setelah proses konfigurasi MQTT selesai, data yang diterima oleh Node-RED diteruskan ke komponen tampilan monitoring. Perancangan tampilan dilakukan untuk menyajikan informasi suhu, tekanan, status sistem, dan notifikasi secara terstruktur sehingga kondisi simulasi sistem dapat dipantau oleh pengguna. Parameter suhu dan tekanan ditampilkan dalam bentuk indikator nilai, sedangkan status sistem dan informasi peringatan ditampilkan sebagai informasi kondisi operasi.

Selain fungsi monitoring, tampilan juga dilengkapi dengan komponen kendali berupa saklar mode dan saklar valve. Komponen tersebut digunakan sebagai antarmuka pengguna untuk memberikan perintah kepada ESP32 melalui jalur MQTT. Dengan demikian, tampilan yang dirancang pada sistem tidak hanya digunakan untuk melihat kondisi transformator, tetapi juga menyediakan fungsi kendali sesuai dengan kebutuhan simulasi sistem.



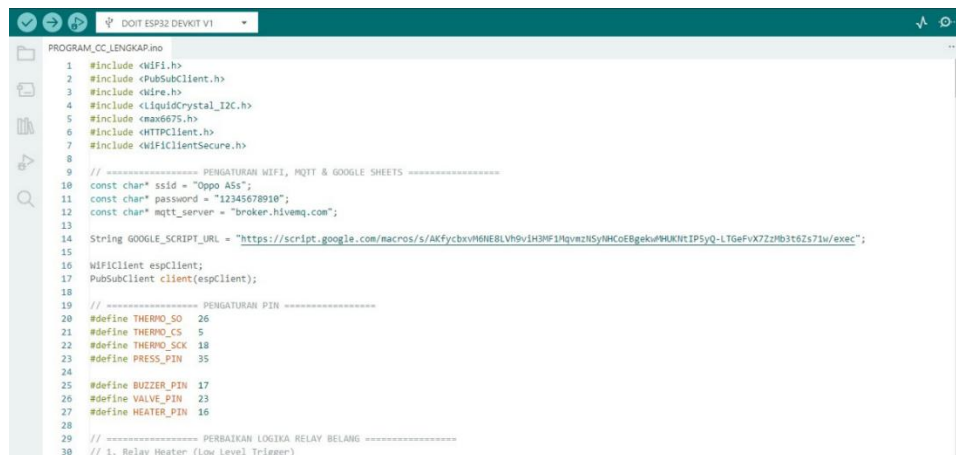
Gambar 4.9 Tampilan Dashboard Node Red

Berdasarkan Gambar 4.9, tampilan monitoring menyediakan informasi suhu dan tekanan internal tangki secara *real-time*. Informasi status sistem digunakan untuk menunjukkan kondisi simulasi sistem berdasarkan parameter yang terukur, sedangkan pesan peringatan digunakan sebagai indikator ketika sistem berada pada kondisi tertentu. Tampilan saklar mode dan valve digunakan sebagai antarmuka kendali untuk memberikan perintah kepada perangkat melalui sistem komunikasi MQTT.

4.5.4 Inisialisasi Sistem

Tahap awal pembuatan perangkat lunak pada sistem monitoring suhu dan tekanan tangki adalah inisialisasi sistem, yaitu proses mempersiapkan mikrokontroler ESP32 agar dapat mengenali, mengatur, dan menjalankan seluruh komponen sesuai dengan fungsi yang telah dirancang. Berdasarkan potongan program pada gambar tersebut, inisialisasi diawali dengan penyertaan beberapa *library* pendukung. *Library* WiFi.h digunakan untuk mengaktifkan konektivitas jaringan nirkabel, sedangkan PubSubClient.h digunakan untuk membangun komunikasi protokol MQTT. Selain itu, digunakan Wire.h dan LiquidCrystal_I2C.h untuk mengendalikan antarmuka layar LCD, max6675.h untuk membaca modul sensor *thermocouple type-k*, serta HTTPClient.h dan WiFiClientSecure.h yang bertugas menangani koneksi dan pengiriman data menuju

Google Spreadsheet.



Gambar 4.1 Inisialisasi Sistem

Pada bagian berikutnya, dilakukan konfigurasi parameter jaringan dan server. Sistem menggunakan variabel pendefinisian ssid ("Oppo A5s") dan password agar ESP32 dapat terhubung ke jaringan internet, serta menetapkan mqtt_server ("broker.hivemq.com") sebagai *broker* untuk bertukar data dengan *website* Node-RED. Sistem juga menginisialisasi tautan *Google Apps Script Web App* melalui variabel GOOGLE_SCRIPT_URL yang akan menjadi jalur pengiriman basis data menuju *Google Spreadsheet*. Melalui perintah WiFiClient dan PubSubClient, mikrokontroler dipersiapkan untuk menjalankan fungsi komunikasi tersebut secara bersamaan.

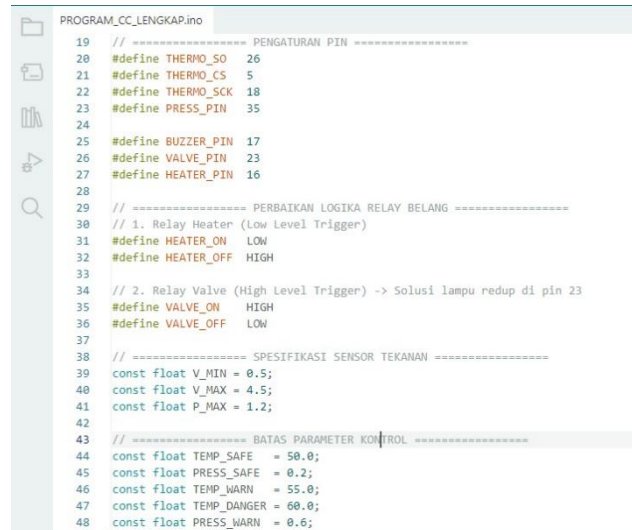
Setelah jalur komunikasi dikonfigurasi, dilakukan pendefinisian pin perangkat keras menggunakan perintah #define agar selaras dengan skema pengkabelan. Konfigurasi pin ini meliputi jalur komunikasi SPI modul MAX6675 yang ditetapkan pada pin SO (GPIO 26), CS (GPIO 5), dan SCK (GPIO 18). Selanjutnya, pin pembacaan sensor tekanan analog dialokasikan pada GPIO 35. Pendefinisian juga dilakukan pada sisi keluaran (*output*) aktuator, yang meliputi *buzzer* pada GPIO 17, *solenoid valve* pada GPIO 23, dan *Heater* pada GPIO 16.

Setelah seluruh deklarasi variabel dan konfigurasi awal ini selesai, ESP32 akan mengeksekusi fungsi setup() untuk mengaktifkan komunikasi *Serial*, menyambungkan perangkat ke Wi-Fi dan MQTT, serta menetapkan mode pin input/output. Pada tahap penyiapan tersebut, *Relay Heater* dan *valve* secara spesifik

diatur dalam kondisi mati (OFF) saat sistem pertama kali mendapat daya. Hal ini merupakan prosedur pengamanan wajib untuk mencegah pemanas atau katup langsung aktif bekerja sebelum mikrokontroler berhasil membaca status suhu dan tekanan tangki secara stabil.

4.5.4.1 Konfigurasi Sensor *Thermocouple Type-K* dan MAX6675

Salah satu komponen yang digunakan pada sistem yaitu *Thermocouple Type K* yang berfungsi sebagai parameter *input* untuk mengukur suhu internal pada tangki transformator daya. Sensor ini digunakan untuk mengetahui perubahan suhu di dalam tangki selama proses simulasi pemanasan, sehingga kondisi temperatur dapat dipantau dan dijadikan salah satu parameter utama dalam pengendalian *Heater* maupun *solenoid valve*. Pada sistem, *Thermocouple Type K* dihubungkan dengan modul MAX6675 yang berfungsi sebagai antarmuka pembacaan sensor dengan mikrokontroler ESP32. Untuk melakukan inisialisasi sensor, digunakan *library max6675.h* yang menyediakan fungsi untuk membaca data temperatur dari *Thermocouple Type K*. Komunikasi antara modul MAX6675 dan ESP32 dilakukan melalui tiga jalur komunikasi SPI, yaitu *Serial Clock (SCK)*, *Chip Select (CS)*, dan *Serial Data Out (SO)*. Konfigurasi ini menjadi bagian penting dalam proses pembacaan suhu karena data temperatur berwujud sinyal analog dari *thermocouple type-k* terlebih dahulu diproses dan dikonversi menjadi data digital 12-bit oleh MAX6675 sebelum diterima oleh ESP32. Penjelasan lebih rinci mengenai inisialisasi perangkat lunak suhu *thermocouple type-k* dan MAX6675 disampaikan pada bagian berikut.



```

PROGRAM_CC LENGKAP.ino
19 // ===== PENGATURAN PIN =====
20 #define THERMO_SO 26
21 #define THERMO_CS 5
22 #define THERMO_SCK 18
23 #define PRESS_PIN 35
24
25 #define BUZZER_PIN 17
26 #define VALVE_PIN 23
27 #define HEATER_PIN 16
28
29 // ===== PERBAIKAN LOGIKA RELAY BELANG =====
30 // 1. Relay Heater (Low Level Trigger)
31 #define HEATER_ON LOW
32 #define HEATER_OFF HIGH
33
34 // 2. Relay Valve (High Level Trigger) -> Solusi lampu redup di pin 23
35 #define VALVE_ON HIGH
36 #define VALVE_OFF LOW
37
38 // ===== SPESIFIKASI SENSOR TEKANAN =====
39 const float V_MIN = 0.5;
40 const float V_MAX = 4.5;
41 const float P_MAX = 1.2;
42
43 // ===== BATAS PARAMETER KONTROL =====
44 const float TEMP_SAFE = 50.0;
45 const float PRESS_SAFE = 0.2;
46 const float TEMP_WARN = 55.0;
47 const float TEMP_DANGER = 60.0;
48 const float PRESS_WARN = 0.6;

```

Gambar 4. 10 Konfigurasi Sensor *Thermocouple Type-K* dan MAX6675

Proses ini diawali dengan pendefinisian pin komunikasi yang digunakan oleh modul MAX6675, yaitu GPIO 26 sebagai *Serial Data Out* (THERMO_SO), GPIO 5 sebagai *Chip Select* (THERMO_CS), dan GPIO 18 sebagai *Serial Clock* (THERMO_SCK). Pin SO berfungsi sebagai jalur keluaran data dari MAX6675 menuju ESP32, sedangkan pin CS digunakan untuk memilih modul MAX6675 pada saat proses pembacaan data, dan pin SCK digunakan sebagai jalur *clock* sinkronisasi komunikasi.

Selanjutnya dilakukan pembuatan objek sensor dengan perintah MAX6675 `thermocouple(THERMO_SCK, THERMO_CS, THERMO_SO);`. Melalui inisialisasi tersebut, ESP32 dapat berkomunikasi dengan modul MAX6675 dan memperoleh nilai suhu aktual dari *Thermocouple Type K*. Data suhu yang diperoleh selanjutnya dapat digunakan untuk memantau temperatur minyak di dalam tangki prototipe selama sistem beroperasi, serta menjadi dasar pengendalian keselamatan (*interlock*) aktuator dan pengiriman data *monitoring* secara berkelanjutan menuju *website* Node-RED dan *Google Spreadsheet*.

4.5.4.2 Konfigurasi Wi-Fi dan Website



```

PROGRAM_CC LENGKAP.ino
65 void setup_wifi() {
66   delay(10);
67   WiFi.begin(ssid, password);
68   while (WiFi.status() != WL_CONNECTED) { delay(500); }
69 }
70
71 void callback(char* topic, byte* payload, unsigned int length) {
72   String pesan = "";
73   for (int i = 0; i < length; i++) { pesan += (char)payload[i]; }
74
75   if (String(topic) == "angel/skripsi/trafo/mode") {
76     if (pesan == "AUTO") isAutoMode = true;
77     else if (pesan == "MANUAL") isAutoMode = false;
78   }
79   if (String(topic) == "angel/skripsi/trafo/valve_cmd") {
80     if (pesan == "ON") manualValveState = true;
81     else if (pesan == "OFF") manualValveState = false;
82   }
83 }
84
85 void reconnect() {
86   while (!client.connected()) {
87     String clientId = "ESP32-Skripsi-";
88     clientId += String(random(0xffff), HEX);
89     if (client.connect(clientId.c_str())) {
90       client.subscribe("angel/skripsi/trafo/mode");
91       client.subscribe("angel/skripsi/trafo/valve_cmd");
92     } else {
93       vTaskDelay(5000 / portTICK_PERIOD_MS);
94     }
95   }
96 }

```

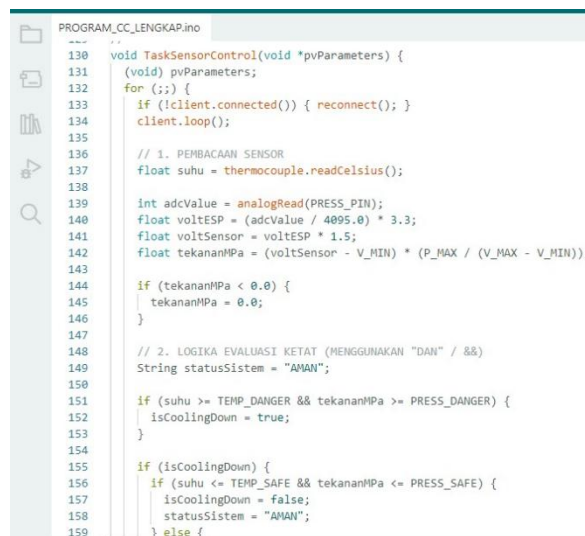
Gambar 4.11 Konfigurasi Wi-Fi dan Website

Untuk mengintegrasikan prototipe tangki transformator daya dengan ekosistem *Internet Of Things* (IoT), perangkat lunak menggunakan beberapa blok fungsi khusus untuk menangani konektivitas jaringan dan komunikasi data. Berdasarkan potongan program tersebut, fungsi `setup_wifi()` bertugas untuk menginisialisasi koneksi ESP32 ke jaringan internet nirkabel menggunakan parameter kredensial SSID dan kata sandi yang telah ditetapkan sebelumnya. Setelah perangkat berhasil terhubung ke jaringan Wi-Fi, sistem akan menjalankan fungsi `reconnect()` untuk membangun komunikasi dengan *broker* MQTT (HiveMQ). Fungsi ini tidak hanya bertugas menjaga koneksi agar tetap stabil dengan melakukan percobaan ulang otomatis setiap lima detik apabila terputus, tetapi juga mendaftarkan (*subscribe*) ESP32 pada topik perintah khusus, yaitu `angel/skripsi/trafo/mode` dan `angel/skripsi/trafo/valve_cmd`, sehingga perangkat siap menerima instruksi masuk dari *website* Node-RED.

Proses pemrosesan instruksi jarak jauh dari *website* tersebut ditangani secara langsung oleh fungsi `callback()`. Fungsi ini bertindak sebagai pendengar (*listener*) yang akan menangkap setiap pesan masuk dari *broker* MQTT, lalu mengonversi data biner (*payload*) tersebut menjadi format teks (*string*) agar dapat dipahami oleh program. Melalui struktur percabangan logika (*if-else*), sistem akan mengidentifikasi topik pesan yang diterima untuk mengeksekusi perintah yang

sesuai. Sebagai contoh, jika *website* mengirimkan pesan "AUTO" atau "MANUAL" pada topik mode, ESP32 akan langsung mengubah nilai variabel `isAutoMode` secara seketika. Demikian pula dengan kendali aktuator, di mana pengguna dapat membuka atau menutup *solenoid valve* secara manual dari jarak jauh melalui *dashboard* Node-RED yang terhubung ke topik `valve_cmd`.

4.5.4.3 Konfigurasi Sensor Tekanan



```

PROGRAM_CC_LENGKAP.ino
...
130 void TaskSensorControl(void *pvParameters) {
131     (void) pvParameters;
132     for (;;) {
133         if (!client.connected()) { reconnect(); }
134         client.loop();
135     }
136     // 1. PEMBACAAN SENSOR
137     float suhu = thermocouple.readCelsius();
138
139     int adcValue = analogRead(PRESS_PIN);
140     float voltESP = (adcValue / 4095.0) * 3.3;
141     float voltSensor = voltESP * 1.5;
142     float tekananMPa = (voltSensor - V_MIN) * (P_MAX / (V_MAX - V_MIN));
143
144     if (tekananMPa < 0.0) {
145         tekananMPa = 0.0;
146     }
147
148     // 2. LOGIKA EVALUASI KETAT (MENGUNAKAN "DAN" / &&)
149     String statusSistem = "AMAN";
150
151     if (suhu >= TEMP_DANGER && tekananMPa >= PRESS_DANGER) {
152         isCoolingDown = true;
153     }
154
155     if (isCoolingDown) {
156         if (suhu <= TEMP_SAFE && tekananMPa <= PRESS_SAFE) {
157             isCoolingDown = false;
158             statusSistem = "AMAN";
159         } else {

```

Gambar 4.12 Konfigurasi Sensor Tekanan

Pada blok fungsi `TaskSensorControl`, mikrokontroler melakukan proses akuisisi data dari lingkungan fisik secara berulang. Untuk pemantauan tekanan, proses diawali dengan mengeksekusi perintah `analogRead(PRESS_PIN)` yang akan menghasilkan nilai mentah *Analog-to-Digital Converter* (ADC) beresolusi 12-bit (direpresentasikan dengan rentang nilai 0 hingga 4095). Nilai ADC mentah ini kemudian dikonversi menjadi besaran tegangan listrik pada variabel `voltESP` menggunakan referensi tegangan maksimal pin masukan ESP32, yaitu 3,3 Volt. Mengingat keluaran asli sensor tekanan yang memiliki rentang tegangan lebih tinggi harus diturunkan terlebih dahulu menggunakan perangkat keras pembagi tegangan (*voltage divider*) agar aman bagi ESP32, program melakukan kompensasi dengan mengalikan tegangan tersebut menggunakan faktor pengali 1,5 (`voltESP * 1.5`). Hasil kali pada variabel `voltSensor` ini merepresentasikan nilai tegangan keluaran murni dari sensor tekanan.

Setelah tegangan keluaran aktual sensor diketahui, sistem menjalankan perhitungan matematis menggunakan metode interpolasi linear untuk mengonversi sinyal tegangan tersebut menjadi satuan besaran tekanan fisik (*Pound-force per Square Inch* atau Psi). Perhitungan dilakukan melalui persamaan matematis $(\text{voltSensor} - V_MIN) * (P_MAX / (V_MAX - V_MIN))$, yang mana batasan tegangan operasi sensor (V_MIN dan V_MAX) beserta rentang ukur tekanan maksimal (P_MAX) telah ditetapkan pada saat inisialisasi awal sistem. Selain itu, program menyertakan logika kalibrasi perangkat lunak tambahan berupa kondisi if ($\text{tekananPsi} < 0.0$). Fungsi ini bertindak sebagai filter pelindung yang akan memaksa (*clamp*) nilai tekanan menjadi 0.0 jika perhitungan matematis menghasilkan angka negatif akibat adanya derau sinyal (*noise*) saat tangki dalam keadaan kosong sepenuhnya, sehingga data pemantauan tetap rasional.

4.5.4.4 Konfigurasi Google Spreadsheet



```

248
249 // =====
250 // TASK 2: PENGIRIMAN GOOGLE SHEETS (Core 0) - UPDATE TIAP 15 DETIK
251 // =====
252 void TaskGoogleSheets(void *pvParameters) {
253     (void) pvParameters;
254     for (;;) {
255         vTaskDelay(15000 / portTICK_PERIOD_MS);
256
257         float tempSuhu = 0.0;
258         float tempTekanan = 0.0;
259         String tempStatus = "";
260
261         if (xSemaphoreTake(dataMutex, portMAX_DELAY)) {
262             tempSuhu = sharedSuhu;
263             tempTekanan = sharedTekanan;
264             tempStatus = sharedStatus;
265             xSemaphoreGive(dataMutex);
266         }
267
268         kirimKeGoogleSheets(tempSuhu, tempTekanan, tempStatus);
269     }
270 }
271

```

Gambar 4.13 Konfigurasi *Google Spreadsheet*

Untuk keperluan pencatatan riwayat data pemantauan (*data logging*) secara otomatis dan berkelanjutan, sistem memanfaatkan layanan *Google Spreadsheet* sebagai basis data (*database*) jarak jauh. Berdasarkan potongan program pada gambar tersebut, proses transmisi data ini ditangani secara khusus oleh sebuah *task* mandiri berbasis *FreeRTOS* bernama *TaskGoogleSheets*. Agar tidak membebani kinerja jaringan internet dan menghindari penolakan akses akibat pengiriman data yang terlalu cepat (*rate limit*) dari peladen keamanan Google, fungsi ini dikonfigurasi untuk memberikan jeda siklus pengiriman setiap 15 detik

menggunakan instruksi `vTaskDelay(15000 / portTICK_PERIOD_MS)`.

Sebelum proses pengiriman data dilakukan ke peladen, mikrokontroler harus mengambil nilai parameter aktual tangki terbaru. Karena arsitektur ESP32 memproses pembacaan sensor dan transmisi jaringan secara paralel (*multitasking*), proses akuisisi data ini dilindungi secara ketat menggunakan mekanisme sinkronisasi *Mutex* melalui blok perintah `xSemaphoreTake(dataMutex, portMAX_DELAY)`.

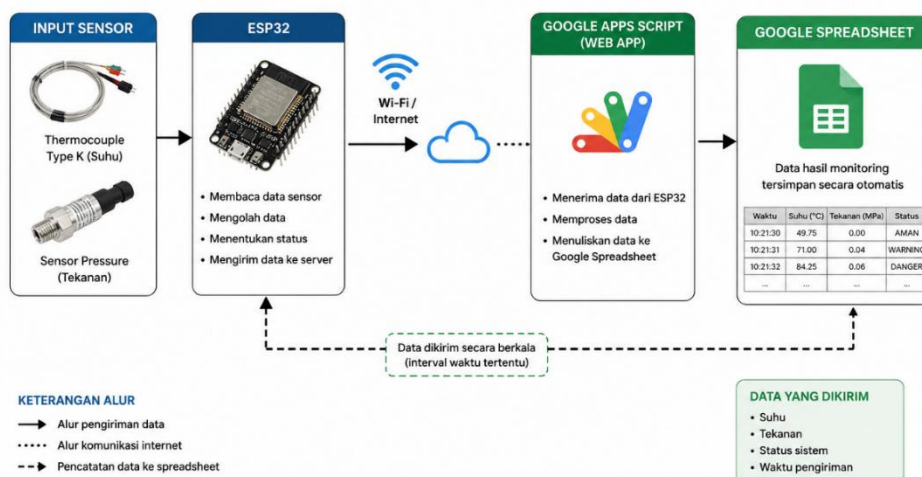
Mekanisme ini berfungsi untuk mengunci memori sementara, memastikan bahwa variabel global seperti suhu (`sharedSuhu`), tekanan (`sharedTekanan`), dan status tangki (`sharedStatus`) tidak berubah nilainya oleh *task* sensor saat sedang disalin. Perlindungan ini sangat krusial untuk mencegah terjadinya kerusakan atau tabrakan data (*data corruption*). Setelah data berhasil disalin dengan aman ke variabel lokal dan kunci memori dilepaskan kembali menggunakan `xSemaphoreGive`, sistem akan mengeksekusi fungsi `kirimKeGoogleSheets()` untuk mempublikasikan riwayat data tersebut ke Google *Spreadsheet*.

4.5.5 Perancangan Integrasi Google *Spreadsheet* dengan ESP32

Google *Spreadsheet* digunakan sebagai media penyimpanan data hasil monitoring suhu dan tekanan secara daring. Integrasi dengan ESP32 dirancang agar data hasil pembacaan sensor dapat dikirim secara otomatis melalui koneksi internet dan dicatat pada Google *Spreadsheet*. Data yang dikirim meliputi nilai suhu, tekanan, dan status kondisi sistem sesuai dengan data yang diperoleh dari ESP32.

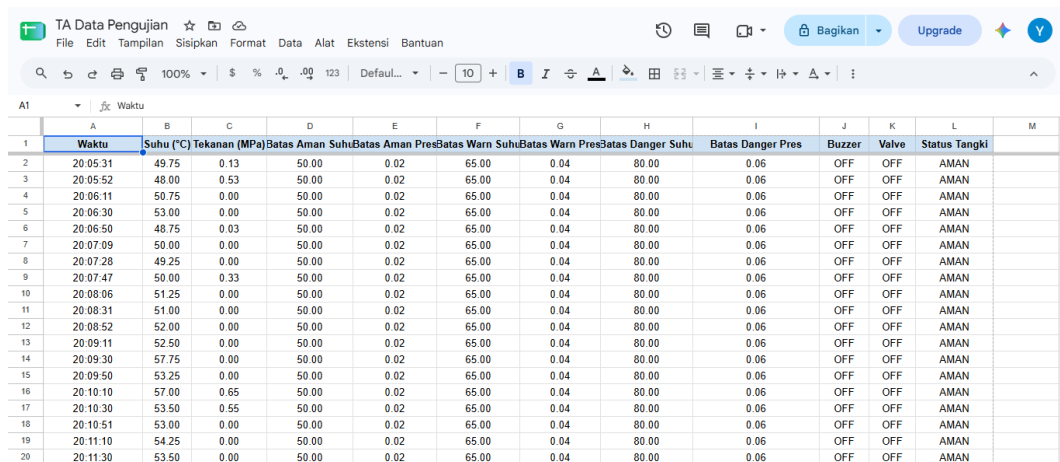
Proses integrasi dilakukan dengan memanfaatkan koneksi Wi-Fi pada ESP32 untuk mengirimkan data menuju Google *Spreadsheet* melalui layanan penghubung yang telah dikonfigurasi. Setelah data diterima, nilai tersebut dicatat pada baris berikutnya di Google *Spreadsheet* berdasarkan waktu pengiriman. Dengan rancangan tersebut, data hasil monitoring tidak hanya dapat ditampilkan pada sistem monitoring, tetapi juga tersimpan sebagai data historis yang dapat digunakan untuk proses analisis.

PERANCANGAN INTEGRASI ESP32 DENGAN **GOOGLE SPREADSHEET**



Gambar 4.14 Integrasi Google *Spreadsheet* dengan ESP32

Berdasarkan Gambar 4.15, data hasil pembacaan sensor suhu dan tekanan diproses oleh ESP32 kemudian dikirimkan melalui jaringan Wi-Fi menuju Google Apps Script yang berfungsi sebagai penghubung antara ESP32 dan Google *Spreadsheet*. Data yang dikirim mencakup nilai suhu, tekanan, status sistem, serta informasi pendukung lainnya sesuai kebutuhan monitoring. Google Apps Script menerima data yang dikirim oleh ESP32, kemudian meneruskannya ke Google *Spreadsheet* untuk dicatat pada baris data berikutnya. Dengan integrasi tersebut, proses pencatatan data monitoring dapat dilakukan secara otomatis dan tersusun berdasarkan waktu pengambilan data, sehingga data yang diperoleh dapat digunakan sebagai data historis untuk pemantauan dan analisis kondisi sistem.



	A	B	C	D	E	F	G	H	I	J	K	L	M
	Waktu	Suhu (°C)	Tekanan (MPa)	Batas Aman Suhu	Batas Aman Pres	Batas Warn Suhu	Batas Warn Pres	Batas Danger Suhu	Batas Danger Pres	Buzzer	Valve	Status Tangki	
2	20:05:31	49.75	0.13	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
3	20:05:52	48.00	0.53	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
4	20:06:11	50.75	0.00	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
5	20:06:30	53.00	0.00	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
6	20:06:50	48.75	0.03	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
7	20:07:09	50.00	0.00	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
8	20:07:28	49.25	0.00	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
9	20:07:47	50.00	0.33	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
10	20:08:06	51.25	0.00	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
11	20:08:31	51.00	0.00	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
12	20:08:52	52.00	0.00	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
13	20:09:11	52.50	0.00	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
14	20:09:30	57.75	0.00	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
15	20:09:50	53.25	0.00	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
16	20:10:10	57.00	0.65	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
17	20:10:30	53.50	0.55	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
18	20:10:51	53.00	0.00	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
19	20:11:10	54.25	0.00	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	
20	20:11:30	53.50	0.00	50.00	0.02	65.00	0.04	80.00	0.06	OFF	OFF	AMAN	

Gambar 4.15 Google Spreadsheet

Berdasarkan Gambar 4.16, data hasil monitoring tersimpan pada Google Spreadsheet secara terstruktur berdasarkan waktu pengambilan data. Data yang tersimpan meliputi nilai suhu, tekanan, batas kondisi suhu dan tekanan, serta kondisi *buzzer*, *valve*, dan status tangki. Penyimpanan tersebut memudahkan proses pemantauan dan penelusuran perubahan kondisi sistem berdasarkan data historis yang telah direkam.