

BAB IV

PEMBUATAN ALAT

4.1 Pembuatan Perangkat Keras

Pembuatan perangkat keras merupakan tahap realisasi dari rancangan yang telah disusun pada BAB III. Tahap ini mencakup pembuatan konstruksi mekanik, pemasangan media penyimpan panas, perakitan sistem pemanas, pemasangan sensor, serta penyusunan rangkaian monitoring. Proses pembuatan mengacu pada dimensi dan spesifikasi komponen yang telah ditetapkan agar prototipe *sand battery* dapat bekerja sesuai tujuan penelitian.

Prototipe menggunakan konstruksi dua tabung yang terdiri atas tabung dalam dan tabung luar. Tabung dalam berfungsi menampung 57 kg pasir silika berukuran 30 sampai 50 mesh, dua elemen pemanas, *coil* tembaga, dan dua thermocouple tipe K. Tabung luar berfungsi melindungi tabung dalam dan menjadi tempat pemasangan rockwool serta aluminium foil sebagai isolasi termal.

Sistem menggunakan dua panel surya 545 Wp yang terhubung secara seri sebagai sumber energi listrik. Kedua panel menyalurkan energi secara langsung menuju dua elemen pemanas 42 V dan 500 W yang juga terhubung secara seri. Elemen pemanas mengubah energi listrik menjadi energi panas, kemudian pasir silika menyerap dan menyimpan panas tersebut.

Sistem monitoring menggunakan sensor arus ACS712, sensor tegangan DC, dua modul MAX6675, ESP32 DevKit V1, Arduino Nano, LCD 20 × 4, dan keypad 4 × 4. Panel surya sekitar 10 Wp berfungsi sebagai panel referensi untuk mengestimasi iradiasi matahari. PSU 5 V 2 A mencatu rangkaian monitoring, sedangkan ESP32 mengirimkan data hasil pengukuran menuju *spreadsheet*.

4.1.1 Alat dan Bahan

Proses pembuatan prototipe *sand battery* memerlukan berbagai alat dan bahan yang saling mendukung. Peralatan digunakan dalam proses pengukuran, pemotongan, pembentukan, penyambungan, perakitan, serta pemeriksaan komponen. Bahan yang digunakan mencakup komponen mekanik, termal,

kelistrikan, elektronik, dan sistem monitoring yang membentuk satu kesatuan prototipe.

Pemilihan setiap alat dan bahan mengacu pada hasil perancangan yang telah dijelaskan pada BAB III. Komponen mekanik membentuk struktur utama prototipe yang terdiri atas tabung dalam, tabung luar, penutup tabung, dan *coil* tembaga. Pasir silika berukuran 30 sampai 50 mesh dengan massa 57 kg berfungsi sebagai media penyimpan energi termal, sedangkan rockwool dan aluminium foil berfungsi sebagai material isolasi.

Dua panel surya monokristalin berkapasitas masing-masing 545 Wp menyediakan energi listrik bagi dua elemen pemanas 42 V dan 500 W. Sistem monitoring menggunakan sensor arus ACS712, sensor tegangan DC, dua thermocouple tipe K, dua modul MAX6675, ESP32 DevKit V1, Arduino Nano, LCD 20×4 , dan keypad 4×4 . Panel surya berkapasitas sekitar 10 Wp berfungsi sebagai panel referensi untuk mengestimasi iradiasi matahari. PSU 5 V 2 A menyediakan catu daya bagi rangkaian monitoring, sedangkan *spreadsheet* menyimpan data hasil pengukuran.

Daftar alat yang digunakan selama proses pembuatan dan perakitan prototipe disajikan pada Tabel 4.1. Daftar bahan dan komponen yang membentuk prototipe *sand battery* disajikan pada Tabel 4.2.

Tabel 4 - 1 Daftar Bahan

No.	Nama Alat	Fungsi
1	Mesin las	Menyambungkan bagian konstruksi logam
2	Mesin gerinda	Memotong dan merapikan permukaan pelat
3	Mesin bor	Membuat lubang pemasangan komponen
4	Pemotong pipa	Memotong pipa tembaga sesuai ukuran
5	Tang pembentuk pipa	Membantu pembentukan pipa tembaga
6	Solder	Menyambungkan komponen dan kabel elektronik
7	Multimeter	Mengukur tegangan, arus, resistansi, dan kontinuitas

No.	Nama Alat	Fungsi
8	Jangka sorong	Mengukur dimensi komponen secara presisi
9	Meteran	Mengukur dimensi tabung dan konstruksi
10	Timbangan	Mengukur massa pasir silika
11	Obeng	Memasang dan melepas baut atau terminal
12	Tang kombinasi	Memotong, menjepit, dan membentuk kabel
13	Komputer atau laptop	Memprogram mikrokontroler dan memantau data
14	Blower	Mengalirkan udara melalui <i>coil</i> tembaga

Tabel 4 - 2 Bahan dan Komponen Prototipe

No.	Nama Bahan atau Komponen	Spesifikasi dan Fungsi
1	Tabung dalam	Diameter 32 cm dan tinggi 46 cm
2	Tabung luar	Diameter 50 cm dan tinggi 40 cm
3	Pasir silika	Massa 57 kg dan ukuran 30 sampai 50 mesh
4	Rockwool	Ketebalan 5 cm sebagai isolasi termal
5	Aluminium foil	Pelindung isolasi dan pengurang radiasi panas
6	Pipa tembaga	Material pembentuk <i>coil</i>
7	Elemen pemanas	2 unit, masing-masing 42 V dan 500 W
8	Panel surya utama	2 unit, masing-masing 545 Wp
9	Panel surya referensi	Sekitar 10 Wp untuk estimasi iradiasi
10	Thermocouple tipe K	2 unit untuk mengukur temperatur pasir
11	Modul MAX6675	2 unit untuk membaca thermocouple
12	Sensor arus ACS712	Mengukur arus pada rangkaian heater
13	Sensor tegangan DC	Mengukur tegangan pada rangkaian heater
14	ESP32 DevKit V1	Mengolah dan mengirimkan data monitoring

No.	Nama Bahan atau Komponen	Spesifikasi dan Fungsi
15	Arduino Nano	Mengelola LCD dan keypad
16	LCD	LCD 20×4 sebagai perangkat tampilan
17	Keypad	Keypad 4×4 sebagai perangkat masukan
18	PSU	5 V 2 A untuk catu daya sistem monitoring
19	Kabel listrik	Menghubungkan panel, heater, dan sensor
20	Kabel thermocouple	Menghubungkan thermocouple dengan MAX6675
21	Baut dan mur	Mengikat bagian konstruksi dan penutup tabung
22	Bahan penyekat	Menutup celah dan mengurangi kebocoran panas
23	<i>Spreadsheet</i>	Mencatat dan menyimpan data hasil pengukuran

4.2 Hardware

4.2.1 Pembuatan Mekanik

Pembuatan mekanikal merupakan tahap penerapan rancangan konstruksi prototipe *sand battery* yang telah disusun pada BAB III. Tahap ini bertujuan membentuk struktur yang mampu menampung pasir silika, elemen pemanas, *coil* tembaga, sensor temperatur, dan material isolasi secara aman dan stabil.

Konstruksi mekanikal terdiri atas tabung dalam, tabung luar, penutup tabung, *coil* tembaga, elemen pemanas, thermocouple tipe K, rockwool, dan aluminium foil. Tabung dalam berfungsi sebagai wadah utama penyimpanan 57 kg pasir silika berukuran 30 sampai 50 mesh. Tabung tersebut memiliki diameter 32 cm dan tinggi 46 cm.

Tabung luar memiliki diameter 50 cm dan tinggi 40 cm. Komponen ini berfungsi melindungi tabung dalam dan material isolasi. Ruang antara kedua tabung menampung rockwool setebal 5 cm dan aluminium foil. Kedua material tersebut membantu mengurangi perpindahan panas dari tabung dalam menuju lingkungan.

Bagian dalam tabung memuat dua elemen pemanas dengan spesifikasi masing-masing sebesar 42 V dan 500 W. Kedua elemen pemanas ditempatkan secara vertikal melalui penutup tabung. Sistem juga menggunakan *coil* tembaga sebagai jalur aliran udara dan penukar panas antara pasir silika dengan udara dari blower.

Rancangan awal *coil* menggunakan pipa tembaga berukuran 1/2 inci dengan diameter keseluruhan 12 cm dan tinggi 48 cm. Hasil evaluasi menunjukkan bahwa rancangan tersebut menimbulkan penurunan tekanan udara blower yang cukup besar. Oleh karena itu, rancangan akhir menggunakan pipa tembaga 3/4 inci pada bagian bawah dan pipa 1/4 inci pada bagian lainnya. *Coil* akhir memiliki diameter keseluruhan 25,1 cm dan tinggi 36 cm.

Dua thermocouple tipe K dipasang secara vertikal melalui penutup tabung. Kedua sensor menempati posisi horizontal yang berbeda untuk mengukur temperatur pasir pada dua lokasi. Penempatan sensor tidak bersentuhan langsung dengan heater, *coil* tembaga, atau dinding tabung agar hasil pengukuran mewakili temperatur pasir silika.

Penutup tabung menopang dua elemen pemanas, dua thermocouple, serta saluran masuk dan keluar *coil* tembaga. Baut pada bagian tepi menghubungkan penutup dengan badan tabung dan memudahkan proses pembukaan saat pemeriksaan atau pemeliharaan. Spesifikasi pengerjaan mekanikal dapat dilihat pada Tabel 4 – 3.

Tabel 4 - 3 Spesifikasi Pengerjaan Mekanikal

Bagian	Spesifikasi
Bentuk konstruksi	Silinder dengan dua lapisan tabung
Tabung dalam	Diameter 32 cm dan tinggi 46 cm
Tabung luar	Diameter 50 cm dan tinggi 40 cm
Media penyimpan	Pasir silika
Massa pasir	57 kg
Ukuran pasir	30 sampai 50 mesh
Material isolasi	Rockwool 5 cm dan aluminium foil

Bagian	Spesifikasi
Jumlah heater	2 unit
Spesifikasi heater	42 V dan 500 W per unit
Orientasi heater	Vertikal
Material penukar panas	Pipa tembaga
Diameter <i>coil</i> akhir	25,1 cm
Tinggi <i>coil</i> akhir	36 cm
Pipa bagian bawah <i>coil</i>	3/4 inci
Pipa bagian lainnya	1/4 inci
Sensor temperatur	2 thermocouple tipe K
Jalur aliran udara	Masuk dan keluar melalui bagian atas
Sumber aliran udara	Blower

Hasil pengerjaan mekanikal membentuk satu konstruksi yang mengintegrasikan media penyimpanan panas, elemen pemanas, penukar panas, sensor temperatur, dan isolasi termal. Susunan tersebut menyediakan ruang yang memadai bagi pasir silika dan mendukung proses penyimpanan serta pengambilan energi termal.

4.2.2 Fabrikasi Komponen Mekanikal oleh Vendor

Vendor melakukan fabrikasi komponen mekanikal berdasarkan gambar kerja yang telah penulis susun pada tahap perancangan. Proses fabrikasi menghasilkan tabung dalam, tabung luar, penutup tabung, dan *coil* tembaga sebagai komponen utama prototipe.

Tabung dalam memiliki diameter 32 cm dan tinggi 46 cm. Tabung tersebut berfungsi sebagai wadah pasir silika, elemen pemanas, *coil* tembaga, dan thermocouple. Tabung luar memiliki diameter 50 cm dan tinggi 40 cm. Tabung luar berfungsi sebagai pelindung mekanis dan rumah bagi material isolasi termal.

Vendor juga membuat penutup tabung yang memuat lubang pemasangan dua elemen pemanas, dua thermocouple, serta saluran masuk dan keluar udara. Posisi lubang mengikuti tata letak komponen yang telah penulis tentukan pada gambar kerja.

Selain konstruksi tabung, vendor membentuk dan menyambungkan *coil* tembaga berdasarkan rancangan akhir. *Coil* memiliki diameter keseluruhan 25,1 cm dan tinggi 36 cm. Struktur tersebut menggunakan pipa tembaga berukuran 3/4 inci pada bagian bawah dan pipa berukuran 1/4 inci pada bagian lainnya.

Tabel 4 - 4 Spesifikasi Tabung

No.	Komponen	Spesifikasi utama
1	Tabung dalam	Diameter 32 cm dan tinggi 46 cm
2	Tabung luar	Diameter 50 cm dan tinggi 40 cm
3	Penutup tabung	Memuat lubang heater, thermocouple, dan saluran udara
4	<i>Coil</i> tembaga	Diameter 25,1 cm dan tinggi 36 cm
5	Pipa bagian bawah <i>coil</i>	Ukuran nominal 3/4 inci
6	Pipa bagian lainnya	Ukuran nominal 1/4 inci

4.2.3 Pembuatan Heat collector coil

Heat Collector yang dibuat dan dicoba pada alat ini disesuaikan dengan kebutuhan target suhu dan perhitungan perpindahan panas dari baterai pasir ke pengering. Terdapat dua heater coil yang dibuat dengan versi kedua sebagai versi final yang dipakai pada alat ini. Perbedaan performa dari versi 1 dan versi 2 cukup signifikan jika membandingkan aliran udara yang pada sisi output coil dengan aliran udara dari blower sebesar 10.6m/s seperti terlihat pada Gambar 4. 1.



Gambar 4. 1 Anemometer

4.2.3.1 Pembuatan Coil Heat Collector versi 1

Rancangan awal dari Heat Collector mempertimbangkan luas area coil dengan harapan udara dapat menangkap panas yang lebih banyak. Coil ini menggunakan pipa tembaga berukuran $\frac{1}{2}$ inci dengan total panjang pipa sepanjang 15 meter lalu di gulung berbentuk kumparan coil dengan diameter 12 Cm dan tinggi coil 48 Cm seperti diperlihatkan pada Gambar 4. 2.



Gambar 4. 2 Tutup Baterai

Karena dimensi coil yang terlalu tinggi. Coil tidak terkubur di dalam pasir silika sepenuhnya dan ini menyebabkan tembaga tidak teroksidasi secara merata. Pada coil versi pertama terjadi penurunan tekanan udara yang signifikan dari aliran udara keluaran blower sebesar 10.6m/s menjadi 5.1m/s seperti diperlihatkan pada Gambar 4. 3 karena jalur yang ditempuh udara terlalu panjang dan pipa yang terlalu sempit untuk mempertahankan tekanan udara.



Gambar 4. 3 Anemometer

Suhu maksimal yang dapat dikeluarkan oleh heat collector pada versi ini adalah 40 Derajat dengan pengukuran menggunakan thermocouple yang diletakan pada bagian keluaran udara coil. Karena penurunan tekanan udara pada coil menyebabkan aliran udara menjadi pelan sehingga udara panas dari baterai tidak mampu untuk menaikkan suhu pada pengering. Coil versi pertama hanya dapat menaikkan suhu pengering dari suhu ruangan menjadi suhu maksimal 33.7 Derajat Celcius seperti diperlihatkan pada Gambar 4. 4.



Gambar 4. 4 LCD DHT21_OVEN

4.2.3.2 Pembuatan Coil Heat Collector versi 2

Pembuatan Coil Heat Collector versi 2 bertujuan untuk menaikkan kemampuan perpindahan panas dari baterai pasir menuju pengering. Pada versi ini terjadi perubahan berupa jalur udara yang diperpendek, susunan coil dirubah menjadi paralel untuk mengurangi tahanan udara tetapi tetap mempertahankan luas permukaan coil, membuat struktur coil menjadi jalur pipa utama menggunakan pipa tembaga ukuran $\frac{3}{4}$ inci dan jalur pipa pendukung dengan menggunakan pipa tembaga ukuran $\frac{1}{2}$ inci dengan bentuk fisik diperlihatkan pada Gambar 4. 5.



Gambar 4. 5 Coil Ver 2

Susunan coil tidak dibuat sejajar seperti versi sebelumnya dengan tujuan mempermudah proses fabrikasi karena Cross fitting untuk pipa tembaga susah ditemukan jika dibanding dengan tee fitting reducer yang memungkinkan untuk menghubungkan pipa pendukung dengan ukuran lebih kecil ke pipa utama supaya dapat menekan ongkos produksi dengan mengurangi penggunaan pipa tembaga ukuran $\frac{3}{4}$ inci tanpa mempengaruhi kemampuan coil untuk menyerap panas pada baterai. Memperbesar ukuran pipa, mengurangi panjang dan total lekukan pada coil menghasilkan penyaluran udara yang lebih baik dan tetap mempertahankan suhu yang mampu diserap oleh heat collector. Suhu pada output coil tetap berada di angka 70 derajat celsius tetapi aliran udara menjadi lebih lancar saat diukur menggunakan anemometer menjadi 8.6m/s seperti diperlihatkan pada Gambar 4.6.



Gambar 4. 6 Anemometer

Peningkatan aliran udara sangat mempengaruhi kemampuan penghantaran panas dari baterai pasir menuju pengering yang sebelumnya tidak bisa melebihi 33.7 derajat menjadi 50.2 derajat.

4.2.4 Pembuatan Pemanas pada Pengering

Pemanas pada pengering memiliki fungsi untuk meradiasikan panas yang telah diambil dari baterai pasir. Pada perancangan awal, pemanas ini memiliki

desain dan fungsi kerja yang mirip dengan radiator yaitu melepaskan panas melalui perpindahan secara radiasi dengan tujuan menaikkan udara di dalam enklosur pengering. Desain kedua menghilangkan bagian pipa yang berfungsi sebagai radiator dan langsung melepaskan udara panas dari baterai ke ruang pengering. Kedua metode ini memiliki kelebihan dan kekurangan masing-masing. Pada pembuatan pemanas versi pertama bisa disebut fully close loop karena udara yang bersirkulasi tidak berganti dan versi kedua disebut partially colse loop karena udara yang dilepas ke oven di sirkulasi lagi ke coil heat collector.

4.2.4.1 Fully Close Loop

Pada pemanas Fully Close Loop udara yang dialirkan pada pipa akan terus bersirkulasi dalam jalur yang sama sehingga sisa udara panas yang sudah bersirkulasi di oven akan dibawa kembali ke coil memungkinkan konservasi panas pada baterai karena udara yang masuk ke Coil Heat Collector memiliki suhu yang lebih tinggi jika dibandingkan dengan suhu ruangan. Saluran pipa dibuat berliku untuk meningkatkan luas permukaan radiator sehingga radiasi panas di dalam pengering dapat berlangsung dengan efisien. Bentuk coil pada sistem close loop setelah dipasang pada oven dapat dilihat pada Gambar 4. 7.



Gambar 4. 7 Piping pengering

4.2.4.2 Partially Close Loop

Pada pemanas Partially Close Loop udara dari Coil Heat Collector dilepaskan di dalam oven. Udara panas dilepaskan pada sisi bawah oven dengan tujuan untuk mendorong udara dingin ke atas dan udara dingin di hisap oleh blower untuk di panaskan lagi melalui Coil Heat Collector. Untuk pelepasan udara panas ke dalam pengering maka jalur pipa tembaga terputus di dalam pengering. Karena sistem ini masih menghisap udara panas dari Coil Heat Collector tetapi udara meninggalkan saluran pipa tembaga maka ada udara dari ruang bebas akan terhisap bersamaan dengan udara panas maka suhu udara yang di sirkulasi kembali ke Heat Collector lebih rendah dibandingkan dengan suhu udara pada Fully Closed Loop. Kelebihan pemanas ini adalah kenaikan suhu yang lebih cepat karena udara dingin langsung digantikan oleh udara panas tetapi panas pada baterai pasir akan lebih cepat turun. Bentuk coil pada sistem semi close loop setelah dipasang pada oven dapat dilihat pada Gambar 4. 8.



Gambar 4. 8 piping Semi Close Loop

4.2.5 Pembuatan saluran pipa tembaga untuk Udara

Sistem perpindahan energi termal dari baterai pasir (sand battery) menuju ruang pengering dirancang menggunakan instalasi perpipaan siklus tertutup (*closed-loop circulation*). Jalur transmisi aliran udara panas ini dibangun

menggunakan material pipa tembaga berdiameter 1/2 inci yang memiliki durabilitas sangat baik terhadap paparan suhu tinggi. Pada setiap area sudut atau perubahan arah lintasan, diaplikasikan sambungan siku (elbow joint) berbahan tembaga untuk memastikan kelancaran dinamika aliran udara (airflow) tanpa menimbulkan hambatan atau turbulensi yang berarti. Guna menjamin sistem benar-benar kedap udara, setiap titik sambungan pipa disegel secara rapat menggunakan selotip aluminium (aluminium foil tape). Penggunaan selotip aluminium ini sangat penting untuk mencegah terjadinya kebocoran udara panas pada celah-celah sambungan, sehingga tekanan udara dan volume kalor yang disalurkan tetap terjaga secara konstan di sepanjang jalur perpipaan. Pemasangan aluminium tape pada pipa tembaga dapat dilihat pada Gambar 4. 9.



Gambar 4. 9 Aluminium tape

Untuk mempertahankan efisiensi perpindahan panas selama proses distribusi, bagian luar lintasan pipa tembaga yang menjembatani baterai pasir dan oven dilapisi dengan material isolator berupa selongsong rockwool (pipe insulation sleeve). Pengaplikasian insulasi termal ini berfungsi untuk menekan rugi-rugi panas (heat loss) yang diakibatkan oleh disipasi termal ke lingkungan sekitar. Dengan adanya selongsong rockwool ini, suhu udara panas yang mengalir di dalam lintasan pipa tidak akan terpengaruh oleh fluktuasi suhu udara luar ruangan (ambient temperature), sehingga panas hasil ekstraksi dapat diantarkan secara utuh menuju ruang pengering. Pemasangan rockwool selongsong dapat dilihat pada Gambar 4. 10.



Gambar 4. 10 Rockwool Selongsong

Secara operasional, sirkulasi perpindahan energi pada instalasi ini bekerja secara terus-menerus dalam satu lintasan berkesinambungan yang digerakkan oleh aktuator blower. Proses dimulai ketika blower meniupkan udara secara paksa ke dalam lintasan pipa menuju baterai pasir. Di dalam ruang penyimpan panas tersebut, udara mengalir melewati coil heat collector (gulungan pipa tembaga) yang tertanam pada media pasir silika bersuhu tinggi untuk menyerap energi termal, sehingga suhu udara meningkat drastis. Udara panas ini kemudian mengalir melewati jalur pipa berinsulasi menuju ruang oven. Di dalam ruang pengering, pelapis isolator ditiadakan sehingga susunan pipa tembaga bertindak sebagai radiator yang meradiasikan energi panas secara merata ke seluruh bahan yang dikeringkan. Setelah melepaskan kalor di dalam oven, udara yang kini bersuhu lebih rendah akan ditarik kembali melalui jalur perpipaan menuju ke blower untuk kembali ditiupkan ke dalam baterai pasir, membentuk siklus aliran panas yang stabil, berkelanjutan, dan hemat energi.

4.2.6 Pemeriksaan Hasil Fabrikasi

Pemeriksaan hasil fabrikasi bertujuan memastikan kesesuaian komponen dengan gambar kerja. Penulis menggunakan alat ukur untuk memeriksa diameter dan tinggi tabung, ukuran penutup, dimensi *coil*, serta posisi lubang pemasangan.

Pemeriksaan visual mencakup kondisi permukaan pelat, sambungan, bentuk tabung, dan kondisi pipa tembaga. Penulis memastikan bagian tabung tidak memiliki deformasi yang dapat menghambat pemasangan. Penulis juga memeriksa sambungan *coil* untuk mengurangi risiko kebocoran udara

4.2.7 Pemasangan *Coil* Tembaga

Penulis memasang *coil* tembaga di dalam tabung dalam sesuai dengan posisi yang telah ditentukan pada tahap perancangan. *Coil* berfungsi sebagai jalur aliran udara dan media pemindahan energi termal dari pasir silika menuju udara.

Rancangan akhir *coil* memiliki diameter 25,1 cm dan tinggi 36 cm. Diameter tersebut menghasilkan jarak radial sekitar 3,45 cm dari dinding tabung dalam. Ruang tersebut memungkinkan pasir silika mengelilingi permukaan pipa tembaga. Penulis mengarahkan kedua pipa vertikal menuju bagian atas tabung. Kedua pipa tersebut berfungsi sebagai jalur masuk dan keluar udara. Penulis memastikan bagian spiral tidak menyentuh langsung dinding tabung agar panas tidak mengalir secara langsung menuju konstruksi logam.

Setelah memasang *coil*, penulis memeriksa kestabilan posisi dan kondisi sambungan pipa. Pemeriksaan kebocoran membantu memastikan udara dari blower dapat mengalir melalui seluruh jalur *coil*.

4.2.8 Pemasangan Thermocouple Tipe K

Penulis memasang dua thermocouple tipe K melalui lubang pada penutup tabung. Kedua sensor memiliki orientasi vertikal dan kedalaman yang relatif sama, tetapi menempati posisi horizontal yang berbeda.

Penulis menjaga ujung sensor agar tidak bersentuhan langsung dengan heater, *coil*, dasar tabung, atau dinding tabung. Susunan tersebut membantu sensor mengukur temperatur pasir silika pada masing-masing titik pengukuran.

Bagian konektor thermocouple berada di atas penutup tabung. Posisi tersebut memudahkan penyambungan sensor dengan dua modul MAX6675 dan memungkinkan proses pemeriksaan tanpa mengeluarkan seluruh media pasir;

4.2.9 Hasil Akhir Perakitan Mekanikal

Hasil perakitan mekanikal membentuk prototipe *sand battery* dengan konstruksi dua lapis. Tabung dalam menampung pasir silika, dua elemen pemanas, *coil* tembaga, dan dua thermocouple. Tabung luar melindungi rockwool serta aluminium foil yang mengelilingi tabung dalam.

Penulis menempatkan terminal heater, konektor thermocouple, dan saluran udara pada bagian atas konstruksi. Susunan tersebut memudahkan proses penyambungan, pemeriksaan, dan pemeliharaan.

Perakitan mekanikal mengacu pada hasil fabrikasi vendor dan rancangan yang telah penulis tetapkan. Penulis melakukan pemeriksaan serta penyesuaian selama proses perakitan agar setiap komponen berada pada posisi yang sesuai. Spesifikasi akhir alat setelah perakitan dapat dilihat pada Tabel 4 – 5.

Tabel 4 - 5 Spesifikasi Akhir

Bagian	Hasil perakitan
Tabung dalam	Diameter 32 cm dan tinggi 46 cm
Tabung luar	Diameter 50 cm dan tinggi 40 cm
Media penyimpan	Pasir silika 57 kg
Ukuran pasir	30 sampai 50 mesh
Isolasi	Rockwool 5 cm dan aluminium foil
Elemen pemanas	2 unit, 42 V dan 500 W
Orientasi heater	Vertikal
Coil tembaga	Diameter 25,1 cm dan tinggi 36 cm
Pipa bagian bawah	Nominal 3/4 inci
Pipa bagian lainnya	Nominal 1/4 inci
Thermocouple	2 unit tipe K
Jalur udara	Masuk dan keluar melalui bagian atas
Pelaksana fabrikasi	Vendor
Pelaksana perakitan	Penulis

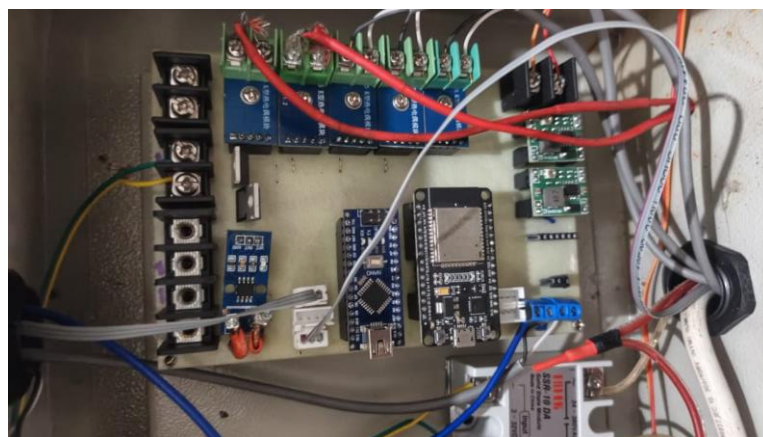
4.3 Perancangan Elektrikal

Perancangan Elektrikal merupakan bagian pembuatan alat yang mengembangkan sistem kelistrikan dari alat. Dimulai dari pemilihan komponen, sensor serta mikrokontroller. Penentuan perangkat dan komponen pendukung, wiring diagram, pendesainan pcb, pembuatan pcb dan perakitan pcb dan jalur kelistrikan alat, jalur kelistrikan panel surya dan peletakan panel surya.

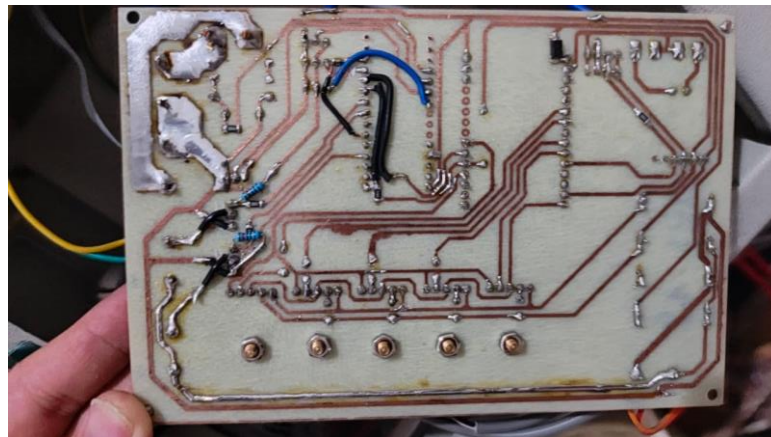
4.3.1 Pembuatan PCB

Pembuatan PCB dilakukan menggunakan pcb polos yang di cetak sesuai dengan desain pcb yang sudah dibuat. Proses pencetakan jalur PCB dimulai dengan penyablonan jalur PCB supaya proses pengikisan tembaga pada PCB (etching) menggunakan kimia Ferric Chlorida dapat menghasilkan jalur sesuai dengan desain yang sudah dibuat. Setelah PCB di etching, PCB di lubangi pada setiap pin komponen menggunakan bor tangan dengan mata bor 0,8 mm untuk resistor dan dioda, 1 mm untuk pin header, terminal block kecil dan IRF540, 3 mm untuk terminal block KF-45 dan baut spacer.

Penempatan komponen pada PCB disesuaikan dengan desain yang sudah dibuat. Komponen diletakan sesuai dengan jalur PCB dan lubang pada PCB. Komponen disolder pada PCB untuk memastikan jalur PCB dan pin komponen terhubung dengan baik dan memiliki resistansi yang baik. Setelah komponen disolder ke PCB harus dilakukan pengecekan kontinuitas untuk memastikan jalur kelistrikan terhubung dengan baik dan terisolasi terhadap jalur lainnya. Setelah PCB jadi maka pcb diletakkan di dalam box panel utama dan PCB juga dihubungkan ke panel surya, heater, dan komponen external lainnya. Tampak depan dan belakang PCB dapat dilihat pada Gambar 4. 11 dan Gambar 4. 12.



Gambar 4. 11 PCB tampak depan



Gambar 4. 12 PCB tampak belakang

4.3.2 Koneksi dari PCB ke Komponen External

Untuk mengintegrasikan unit mikrokontroler dengan berbagai komponen eksternal, sistem ini menggunakan papan sirkuit cetak (PCB) utama yang dirancang secara khusus dengan beberapa variasi antarmuka konektor. Pemilihan jenis konektor ini disesuaikan dengan kapasitas beban arus, tingkat tegangan, serta protokol komunikasi data pada masing-masing periferal. Pada jalur distribusi daya listrik dan komponen berarus tinggi, antarmuka PCB dihubungkan menggunakan konektor *terminal block* seri KF-45. Penggunaan *terminal block* berukuran besar ini diaplikasikan secara spesifik pada jalur sambungan menuju dua panel surya utama, elemen pemanas (*heater*), serta unit catu daya (*power supply*). Pemilihan seri KF-45 bertujuan untuk mengamankan ikatan kabel berpenampang besar, menjamin keandalan kontak mekanis, serta meminimalisasi resistansi dan risiko pemanasan (*overheating*) pada titik sambungan akibat aliran arus tinggi. Sementara itu, untuk jalur kelistrikan sekunder dengan beban arus yang jauh lebih rendah, seperti sambungan menuju panel surya referensi 10 W dan beban tiruan (*dummy load*), perancangan menggunakan *terminal block* baut dengan ukuran yang lebih kecil guna mengoptimalkan tata letak ruang pada PCB.

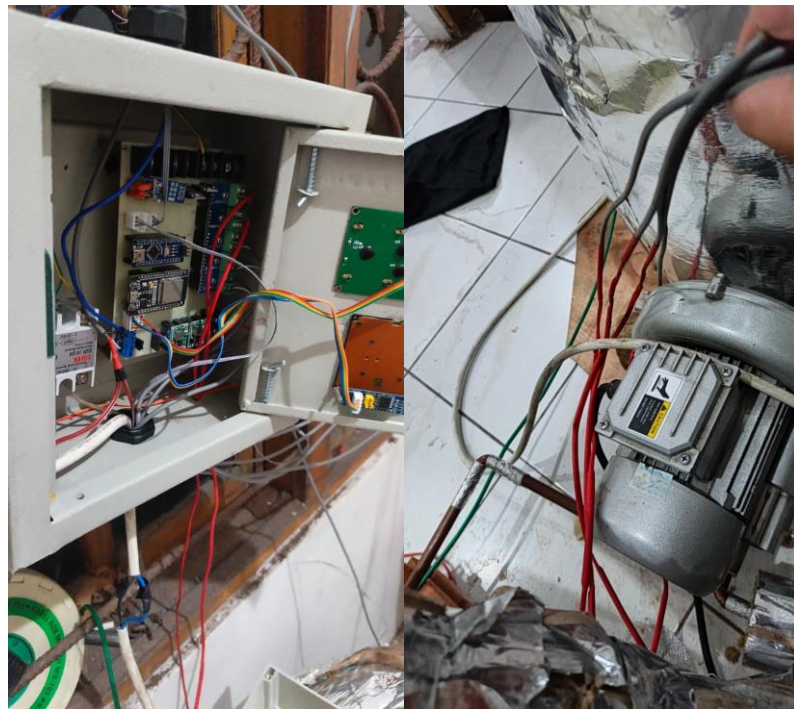
Pada subsistem antarmuka pengguna (UI) dan instrumentasi sensor digital, perancangan sirkuit mengadopsi penggunaan konektor soket bertipe JST (*Japan Solderless Terminal*). Konektor JST 4-pin dialokasikan secara khusus untuk menghubungkan PCB utama dengan modul layar LCD 20x4 dan matriks *keypad*

4x4. Kedua perangkat antarmuka tersebut beroperasi menggunakan protokol komunikasi I2C, sehingga keempat pin pada soket JST dimanfaatkan untuk jalur suplai tegangan (VCC), *ground* (GND), jalur data serial (SDA), dan *clock serial* (SCL). Selain itu, konektor JST 3-pin digunakan untuk mengintegrasikan mikrokontroler dengan sensor suhu dan kelembaban DHT21. Pengaplikasian soket JST pada perangkat digital ini menawarkan keunggulan berupa kemudahan bongkar pasang (*plug-and-play*) yang memudahkan proses perawatan (*maintenance*) maupun penggantian sensor, serta dilengkapi dengan mekanisme pengunci bawaan yang mencegah kabel terlepas dari PCB akibat getaran saat sistem beroperasi.

Di sisi lain, untuk kebutuhan pemantauan suhu tinggi pada baterai pasir dan ruang oven, kabel ujung probe sensor termokopel tipe K tidak disolder atau dihubungkan langsung ke antarmuka konektor JST pada PCB. Kabel termokopel tersebut dihubungkan melalui *terminal block* baut bawaan berukuran kecil yang telah terintegrasi secara pabrikan pada setiap modul pembaca MAX6675. Penggunaan baut pengunci pada *terminal block* MAX6675 ini sangat krusial untuk memastikan polaritas kabel logam bimetal termokopel terjepit dengan kencang dan sempurna. Sambungan fisik yang solid pada titik ini sangat menentukan stabilitas tegangan *millivolt* yang dihasilkan oleh efek termoelektrik, sehingga mikrokontroler ESP32 dapat menerima dan memproses data pengukuran temperatur ekstrem dengan tingkat akurasi yang tinggi tanpa adanya gangguan fluktuasi sinyal akibat koneksi kabel yang longgar. Bentuk box panel dan sambungan kabel ke alat dapat dilihat pada Gambar 4. 13 dan Gambar 4. 14.



Gambar 4. 13 Box Panel



Gambar 4. 14 Sambungan Kabel

4.3.3 Penempatan dan jalur kelistrikan panel surya

Dalam implementasi sistem konversi energi, penempatan fisik unit panel surya diposisikan di atas atap bangunan guna memastikan paparan iradiasi sinar matahari dapat ditangkap secara maksimal sepanjang hari tanpa terhalang oleh bayangan (*shading effect*). Untuk menjaga stabilitas dan orientasi kemiringan panel, sistem ini ditopang menggunakan struktur penyangga (*mounting*) yang dirakit dari material profil aluminium *hollow* berdimensi 4x4 cm. Pemilihan material aluminium *hollow* ini didasarkan pada rasio kekuatan terhadap bobot mekanis yang sangat baik, serta ketahanannya terhadap proses korosi akibat paparan cuaca luar ruangan (*outdoor*). Rangka penopang ini dirancang agar mampu menahan beban statis dari dimensi panel surya utama secara kokoh serta memastikan infrastruktur tidak bergeser meskipun terkena terpaan angin.

Untuk mentransmisikan daya listrik arus searah (DC) dari sumber pembangkit surya menuju unit kontrol pada *Printed Circuit Board* (PCB), sistem penyaluran kelistrikan (*wiring*) dirancang dengan membedakan spesifikasi kabel berdasarkan kapasitas beban arus. Pada jalur kelistrikan dari dua unit panel surya

utama menuju *terminal block* PCB dan elemen pemanas (*heater*), transmisi daya dihubungkan menggunakan kabel berjenis NYA dengan luas penampang 2,5 mm². Kabel berinti tembaga tunggal (pejal) ini diaplikasikan karena memiliki kemampuan hantar arus (KHA) yang mumpuni serta tingkat penurunan tegangan (*voltage drop*) yang sangat rendah untuk menyalurkan arus berkapasitas tinggi secara konstan dan aman.

Sementara itu, pendekatan berbeda diterapkan pada jalur kelistrikan panel surya sekunder berkapasitas 10 W, yang fungsinya dikhususkan sebagai sensor referensi pembacaan intensitas radiasi matahari (*solar irradiance*). Jalur kelistrikan dari panel kecil ini menuju PCB dihubungkan menggunakan kabel fleksibel berjenis NYAF dengan luas penampang 0,5 mm². Penggunaan kabel berinti serabut (*stranded wire*) berukuran kecil ini dipilih karena beban arus yang mengalir dari panel 10 W menuju sensor INA219 sangatlah rendah. Selain efisien secara kapasitas, kabel NYAF 0,5 mm² juga memberikan fleksibilitas mekanis yang tinggi, sehingga memudahkan proses instalasi kelistrikan dan perapian kabel (*cable management*) saat dihubungkan ke konektor terminal baut berukuran kecil pada PCB utama. Penempatan panel surya di atas atap dapat dilihat pada Gambar 4. 15.

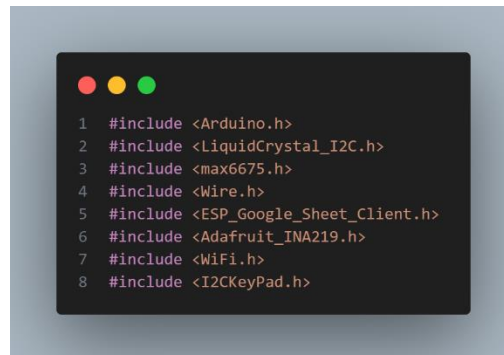


Gambar 4. 15 Panel Surya

4.4 Software

4.4.1 inisiasi dan Deklarasi

4.4.1.1 Library



Gambar 4. 16 Deklarasi Library ESP32

Deklarasi library yang digunakan pada keseluruhan program ESP32 dengan menggunakan format kode `#include <Nama_Library>`. Library yang digunakan pada ESP32 adalah `<Arduino.h>` sebagai library untuk framework arduino pada platform.io, `<LiquidCrystal_I2C.h>` sebagai library LCD 20x4 i2c, `<max6675.h>` sebagai library pembacaan thermocouple, `<Wire.h>` sebagai library untuk komunikasi serial pada mikrokontroller dengan framework arduino, `<ESP_Google_Sheet_Client.h>` sebagai library untuk upload data menggunakan token google ke spreadsheet, `<Adafruit_INA219.h>` sebagai library INA219 untuk membaca radiasi matahari melalui panel 10W, `<WiFi.h>` sebagai library untuk menjalankan wifi ESP32, `<I2CkeyPad.h>` sebagai library untuk keypad i2c. Deklarasi library pada ESP32 dapat dilihat pada Gambar 4. 16.



Gambar 4. 17 Deklarasi Library Arduino Nano

Deklarasi Library yang digunakan pada keseluruhan program Arduino Nano dengan menggunakan format `#include <Nama_Library>`. Library yang digunakan pada Arduino Nano adalah `<Arduino.h>` sebagai library untuk framework arduino pada platform.io, `<ACS712.h>` sebagai library untuk pembacaan arus oleh sensor ACS712, `<DHT.h>` sebagai library untuk membaca suhu dan kelembaban pada oven dan ruang sekitar oven dan baterai, `<SoftwareSerial.h>` sebagai library untuk komunikasi uart tanpa menggunakan pin default uart Arduino Nano Deklarasi library pada Arduino Nano dapat dilihat pada Gambar 4. 17.

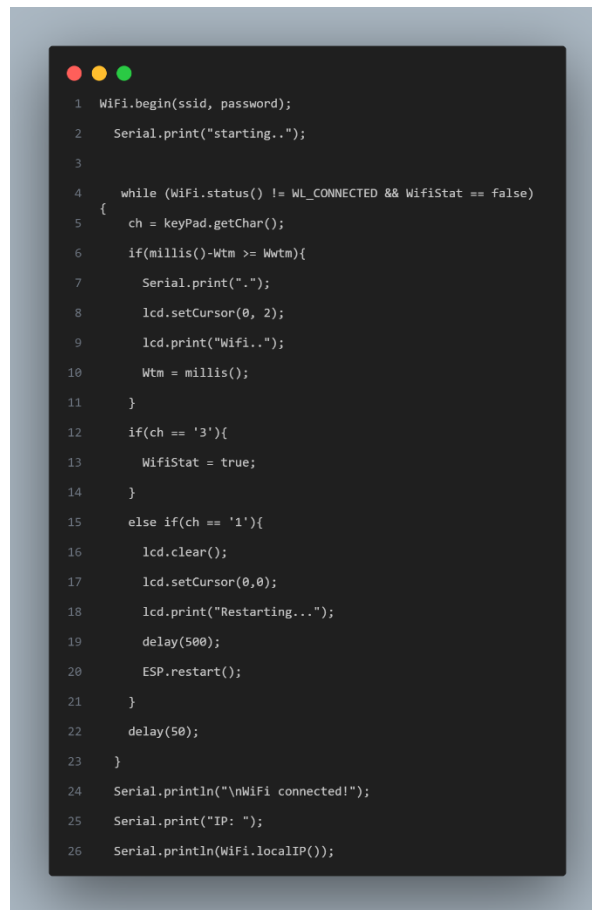
4.4.1.2 Wifi

A screenshot of a code editor window with a dark background and light-colored text. The code is written in C++ and is for configuring WiFi on an ESP32. It includes comments and variable declarations. The code is as follows:

```
1 //Wifi
2 const char* ssid    = "SSID";
3 const char* password = "PASSWORD";
4 int rssi = 0;
5 char wipistat = 'X';
6 bool WifiStat = false;
```

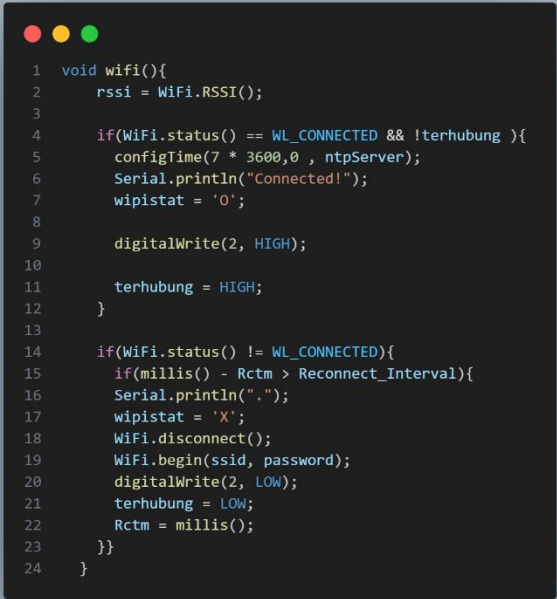
Gambar 4. 18 Deklarasi Wifi ESP32

Deklarasi wifi dimana SSID dan Password wifi dicantumkan supaya ESP32 dapat terhubung ke wifi yang disediakan seperti pada Gambar 4. 18.



Gambar 4. 19 Eksekusi Wifi ESP32

Eksekusi wifi pada void loop saat ESP32 pertama kali di hidupkan, sistem menunggu sampai wifi terhubung untuk melanjutkan ke proses selanjutnya. Percobaan hubung wifi bisa di lewati dengan menekan tombol '0' pada keypad. Program untuk eksekusi wifi ESP32 dapat dilihat pada Gambar 4. 19.



```

1  void wifi(){
2      rssi = WiFi.RSSI();
3
4      if(WiFi.status() == WL_CONNECTED && !terhubung ){
5          configTime(7 * 3600,0 , ntpServer);
6          Serial.println("Connected!");
7          wipistat = '0';
8
9          digitalWrite(2, HIGH);
10
11         terhubung = HIGH;
12     }
13
14     if(WiFi.status() != WL_CONNECTED){
15         if(millis() - Rctm > Reconnect_Interval){
16             Serial.println(".");
17             wipistat = 'X';
18             WiFi.disconnect();
19             WiFi.begin(ssid, password);
20             digitalWrite(2, LOW);
21             terhubung = LOW;
22             Rctm = millis();
23         }}
24     }

```

Gambar 4. 20 Pemantauan Kondisi Wifi

Void wifi() memiliki fungsi untuk memantau kondisi wifi disaat sistem sedang berjalan, void ini akan terus mencoba terhubung ke wifi apabila koneksi wifi ESP32 terputus dan akan menunjukkan kondisi wifi (terhubung atau terputus) pada LCD. Program untuk pemantauan kondisi wifi dapat dilihat pada Gambar 4. 20.

4.4.1.3 Deklarasi

```

1 #define R02 35
2 #define T02 32
3 //MAMA7%
4 #define thermo00 34
5 #define thermoC01 39
6 #define thermoC02 38
7 #define thermoC03 37
8 #define thermoC04 36
9 #define thermoC05 35
10 #define thermoC06 33
11 M00675 thermocouple1(thermoC01, thermoC01, thermo00);
12 M00675 thermocouple2(thermoC01, thermoC02, thermo00);
13 M00675 thermocouple3(thermoC01, thermoC03, thermo00);
14 M00675 thermocouple4(thermoC01, thermoC04, thermo00);
15 M00675 thermocouple5(thermoC01, thermoC05, thermo00);
16 float C1 = 0;
17 float C2 = 0;
18 float C3 = 0;
19 float C4 = 0;
20 float C5 = 0;
21 float Avg_Bat = 0;
22 float Avg_Ov = 0;
23 float Delta_Bat = 0;
24 float DeltaOJBat = 0;
25 float Delta_Ov = 0;
26 float DeltaOJ_Ov = 0;
27 bool FirstDataBat = true;
28 bool FirstDataOv = true;
29
30 //I2C
31 #include <Wire.h>;
32 char ch = 'W';
33 char ch = 'W';
34 bool KeyIn = false;
35 unsigned long lastPressTime = 0;
36 const unsigned long debounceDelay = 50;
37 char keypad[10] = "1234567890";
38
39 /*
40  * 0 1 2 3
41  * C 0 1 2
42  * R 0 1 2 3
43  * A 0 1 2 3
44  */
45
46 LiquidCrystal_I2C lcd(0x27, 20, 4);
47
48 #define INIT_I2C 0x27;
49 float Solar_Radiation = 0;
50
51 //GSM
52 float voltage = 0;
53 float current = 0;
54 float Kiloampere_Ov_Oven = 0;
55 float Sohu_Ov_Oven = 0;
56 float Kiloampere_Ov_Battery = 0;
57 float Sohu_Ov_Battery = 0;
58
59 //INA219
60 float BusVoltage = 0;
61 float Current_mA = 0;
62 float DayLine = 0;
63
64 //Ipouch time
65 const char* httpServer = "pool.ntp.org";
66 String waktuStr = "";
67 String getFormattedTime() {
68   struct tm timeinfo;
69   if (!getLocalTime(&timeinfo)) {
70     return "time failed";
71   }
72   char buf[25];
73   sprintf(buf, "%d %d %d %d %d %d", timeinfo.tm_Mon, timeinfo.tm_May, timeinfo.tm_Day, timeinfo.tm_Hour, timeinfo.tm_Min, timeinfo.tm_Sec);
74   return String(buf);
75 }
76
77 //LCD Display
78 enum Layer{Home, Changing, Sohu_Battery, Sohu_Oven, DHF21_OVEN, DHF22_BATTERY, Panel_Itama, Radiasi, Oven_Run, Oven_Stop, Oven_Running, Oven_Stopping, Restart, LCD_Restart, zzz};
79 Layer layer = Home;
80 Layer lastLayer = Home;
81 unsigned long lastSwitch = 0;
82 const unsigned long lcdInterval = 3000;
83 int lcdPage = 0;
84 const int totalLCDPages = 6;
85 int lcdReset = -1;
86
87 //Run Oven
88 bool OvenMenu = false;
89 bool OvenState = false;
90 bool OvenRunning = false;
91 unsigned long OverUp = 0;
92 const unsigned long OverUpInv = 2000;

```

Gambar 4. 21 Deklarasi Library dan Data Lain ESP32

```

1  #define Vin A0
2  #define ACS_Pin A1
3  #define DHTPIN1 A2
4  #define DHTPIN2 A3
5  #define DHTTYPE DHT21
6  #define BDC 5
7  #define BAC 6
8  #define RX 8
9  #define TX 7
10
11
12  ACS712 ACS(ACS_Pin, 5.0, 1023, 100);
13
14  SoftwareSerial Serial2(RX, TX);
15
16  DHT dht(DHTPIN1, DHTTYPE);
17  DHT dht1(DHTPIN2, DHTTYPE);
18  //UART
19  float C1 = 0;
20  float C2 = 0;
21  float C3 = 0;
22
23  //Read Sens
24  int WriteInterval = 500;
25  unsigned long Wrtm = 0;
26  int ReadInterval = 200;
27  unsigned long Rdtm = 0;
28
29  //Read avg Voltage
30  int Voltage_Interval = 2;
31  unsigned long Vltm = 0;
32  long voltTotal = 0;
33  int voltSampleCount = 0;
34  const int VoltSamples = 100; //
35
36  //Read avg Current
37  int Current_Interval = 1;
38  unsigned long Cltm = 0;
39  long CurrentTotal = 0;
40  int CurrentSampleCount = 0;
41  const int CurrentSamples = 200; //
42
43  float mA = 0;
44  float h = 0;
45  float t = 0;
46  float h1 = 0;
47  float t1 = 0;
48  float avgAdc = 0;
49  double Vp = 0;
50  double Vreal = 0;
51
52  //Kontrol
53  float SetPoint = 45;
54  float Hysteresis = 2;
55  int ControlInterval = 500;
56  unsigned long Cttm = 0;
57  bool BlowerAC = false;
58  float HighHyst;
59  float LowHyst;
60  int OvenSerial = 0;
61  bool OvenState = 0;
62
63  //BLOWER
64  unsigned long Idop = 60000; // 1 menit = 60000
65  float TimerIdop = 3600000; //1 jam = 3600000
66  unsigned long LagiAktip = 0;
67  unsigned long LagiAktipJugo = 0;
68  bool BoolBombong = true;
69  bool Pakum = false;
70

```

Gambar 4. 22 Deklarasi Library dan Data Lain Arduino Nano

Deklarasi library, tipe data dan lain-lain untuk menjalankan program pada ESP32 dan Arduino Nano dapat dilihat pada Gambar 4. 21 dan Gambar 4. 22.

4.4.1.4 Spreadsheet

```

1 //Google SpreadSheet
2 #define PROJECT_ID "****"
3 #define CLIENT_EMAIL "****"
4 const char PRIVATE_KEY[] PROGMEM = "****";
5 const char spreadsheetId_Battery[] = "****";
6 const char spreadsheetId_Oven[] = "****";
7
8 bool SheetStat = false;
9 int interval = 300000;
10 unsigned long tmnw = 0;
11 int ReadInterval = 200;
12 unsigned long Rdtm = 0;
13 int WriteInterval = 500;
14 unsigned long Wrtm = 0;
15 int Reconnect_Interval = 500;
16 unsigned long Rctm = 0;
17 //600000 = 10 menit

```

Gambar 4. 23 Deklarasi Spreadsheet

Deklarasi program spreadsheet yang menentukan spreadsheet id dan project id untuk mengupload data ke spreadsheet dan interval upload data ke spreadsheet. seperti diperlihatkan pada Gambar 4. 23.

```

1 void setup_Battery()
2 {
3   pinMode(LED_BUILTIN, OUTPUT);
4   Serial.begin(115200);
5
6   //Inisialisasi variabel
7   Serial.print("Inisialisasi variabel...");
8   Serial.print("\n");
9
10  //Inisialisasi variabel
11  digitalWrite(LED_BUILTIN, HIGH); // on 10 detik
12  digitalWrite(LED_BUILTIN, LOW); // on 10 detik
13  digitalWrite(LED_BUILTIN, HIGH); // on 10 detik
14  digitalWrite(LED_BUILTIN, LOW); // on 10 detik
15  digitalWrite(LED_BUILTIN, HIGH); // on 10 detik
16  digitalWrite(LED_BUILTIN, LOW); // on 10 detik
17  digitalWrite(LED_BUILTIN, HIGH); // on 10 detik
18  digitalWrite(LED_BUILTIN, LOW); // on 10 detik
19  digitalWrite(LED_BUILTIN, HIGH); // on 10 detik
20  digitalWrite(LED_BUILTIN, LOW); // on 10 detik
21  digitalWrite(LED_BUILTIN, HIGH); // on 10 detik
22  digitalWrite(LED_BUILTIN, LOW); // on 10 detik
23  digitalWrite(LED_BUILTIN, HIGH); // on 10 detik
24  digitalWrite(LED_BUILTIN, LOW); // on 10 detik
25
26  Serial.println("Done - Inisialisasi variabel...");
27
28  if (true) {
29    Serial.println("Inisialisasi variabel...");
30    digitalWrite(LED_BUILTIN, LOW);
31    digitalWrite(LED_BUILTIN, HIGH);
32  }
33  Serial.println("Done - Inisialisasi variabel...");
34  Serial.println("Done - Inisialisasi variabel...");
35
36  //
37
38  //
39
40  //
41
42  //
43
44  //
45
46  //
47
48  //
49
50  //
51
52  //
53
54  //
55
56  //
57
58  //
59
60  //
61
62  //
63
64  //
65
66  //
67
68  //
69
70  //
71
72  //
73
74  //
75
76  //
77
78  //
79
80  //
81
82  //
83
84  //
85
86  //
87
88  //
89
90  //
91
92  //
93
94  //
95
96  //
97
98  //
99
100 //

```

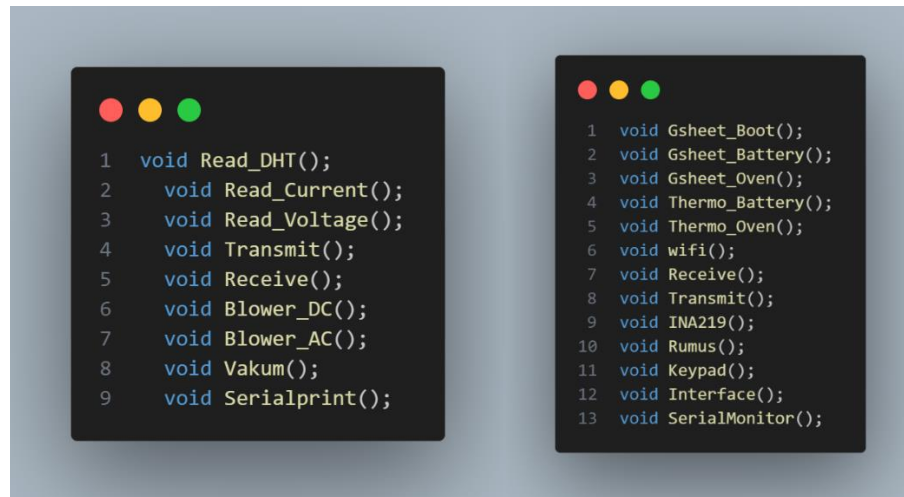
[illegible]

Gambar 4. 24 Program pengiriman data menuju spreadsheet

Kode untuk mengupload data ke masing-masing spread sheet, terdapat Gsheet_boot untuk mengirim data ke semua spreadsheet saat Sistem booting up, Gsheet_Battery untuk mengirim logging data dari sensor dan perhitungan yang berkaitan dengan pengisian energi ke baterai pasir dan Gsheet_Oven untuk mengirim login data dari sensor yang berkaitan dengan discharging baterai ke

pengering. Program keseluruhan pengiriman data menuju spreadsheet dapat dilihat pada Gambar 4. 24.

4.4.1.5 Prototype



Gambar 4. 25 Deklarasi fungsi software

Merupakan bagian yang mendeklarasikan fungsi yang digunakan di dalam software. Deklarasi fungsi software dapat dilihat pada Gambar 4. 25.

4.4.1.6 Interface

```

1 void Interface(){
2     char Nch = keyPad.getChar();
3
4     if(Nch != 0ch) {
5         if(Nch != 'N' && Nch != 'F') { ch = Nch; }
6         else { Keyin = false; ch = 'N'; }
7         Och = Nch;
8     } else {
9         ch = 'N';
10    }
11
12    if(ch != 'N') {
13        Serial.print("Key Pressed : ");
14        Serial.println(ch);
15    }
16
17    if(layar != zzz){
18        static unsigned long Memek = 0;
19        if(millis()-Memek>=500){
20            Memek = millis();
21            Serial.print("Layar : ");
22            Serial.println(layar);
23            Serial.print("Bool : ");
24            Serial.print(OvenMenu);
25            Serial.print("/");
26            Serial.print(OvenState);
27            Serial.print("/");
28            Serial.println(OvenRunning);
29        }
30    }

```

```

1 if(ch == '3' && !OvenMenu) { layar = Home; }
2 else if(ch == '3' && OvenMenu) { layar = LastLayar; }
3 else if(ch == '+') { layar = Charging; }
4 else if(ch == '0') { layar = Sunu_Battery; }
5 else if(ch == '8') { layar = Sunu_Oven; }
6 else if(ch == '0') { layar = DHT22_OVH; }
7 else if(ch == 'C') { layar = DHT22_BATTERY; }
8 else if(ch == '9') { layar = Panel_Utama; }
9 else if(ch == '8') { layar = Radiasi; }
10 else if(ch == '7' && !OvenMenu && !OvenState) { LastLayar = layar; layar = Oven_Run; }
11 else if(ch == '7' && !OvenMenu && OvenState) { LastLayar = layar; layar = Oven_Stop; }
12 else if(ch == '1') { layar = Restart; }
13 else if(ch == 'A' && layar != zzz) { LastLayar = layar; layar = zzz; }
14 else if(ch == 'A' && layar == zzz) { layar = LastLayar; }
15 else if(ch == '2') { LastLayar = layar; layar = LCD_Restart; }
16
17 if(layar == Oven_Run || layar == Oven_Stop) { OvenMenu = true; }
18 else { OvenMenu = false; }
19
20 if(OvenRunning && !OvenMenu && ch == '4') { layar = Oven_Running; OvenPup = millis(); }
21 if(!OvenRunning && !OvenMenu && ch == '4') { layar = Oven_Stopping; OvenPup = millis(); }
22
23 if(layar == Oven_Running) { OvenState = true; }
24 else if(layar == Oven_Stopping) { OvenState = false; }
25
26 if((layar == Oven_Running || layar == Oven_Stopping) && millis() - OvenPup >= OvenPupInv) {
27     layar = LastLayar;
28 }
29

```

```

1 static unsigned long lastLcdDraw = 0;
2 if(millis() - lastLcdDraw >= 250 || layar != LcdReset) {
3     lastLcdDraw = millis();
4
5     if(layar != LcdReset) { lcd.clear(); }
6     if(layar == zzz) { lcd.noBacklight(); }
7     else if(layar != zzz) { lcd.backlight(); }
8

```

```

1  switch(layer) {
2      case Home:
3          lcd.setCursor(0,0); lcd.print("Home");
4          lcd.setCursor(19,0); lcd.print(wipistat);
5          lcd.setCursor(0, 1); lcd.print("Pencet Apa kek ");
6          break;

```

```

1  case Changing:
2      #pragma region changing
3      if (millis() - lastlcdSwitch >= lcdInterval || layer != lcdReset) {
4          lastlcdSwitch = millis();
5          lcd.clear();
6
7          switch (lcdPage) {
8              case 0:
9              lcd.setCursor(0,0); lcd.print("Battery"); lcd.setCursor(19,0); lcd.print(wipistat);
10             lcd.setCursor(0, 1); lcd.print("Suhu 1 : "); lcd.print(C1); lcd.print(" C");
11             lcd.setCursor(0, 2); lcd.print("Suhu 2 : "); lcd.print(C2); lcd.print(" C");
12             break;
13             case 1:
14             lcd.setCursor(0,0); lcd.print("Oven"); lcd.setCursor(19,0); lcd.print(wipistat);
15             if(OvenState){
16                 lcd.setCursor(0, 1); lcd.print("Suhu 1 : "); lcd.print(C3); lcd.print(" C");
17                 lcd.setCursor(0, 2); lcd.print("Suhu 2 : "); lcd.print(C4); lcd.print(" C");
18                 lcd.setCursor(0, 3); lcd.print("Suhu 3 : "); lcd.print(C5); lcd.print(" C");
19             }else{lcd.setCursor(0, 1); lcd.print("Oven isn't Running!");}
20             break;
21             case 2:
22             lcd.setCursor(0, 0); lcd.print("DHT21_OVEN "); lcd.setCursor(19,0); lcd.print(wipistat);
23             if(OvenState){
24                 lcd.setCursor(0, 1); lcd.print("Suhu DHT : "); lcd.print(Suhu_DHT_Oven);
25                 lcd.setCursor(0, 2); lcd.print("Humid DHT : "); lcd.print(Kelembapan_DHT_Oven);
26             }else{lcd.setCursor(0, 1); lcd.print("Oven isn't Running!");}
27             break;
28             case 3:
29             lcd.setCursor(0, 0); lcd.print("DHT21_BATTERY "); lcd.setCursor(19,0); lcd.print(wipistat);
30             lcd.setCursor(0, 1); lcd.print("Suhu DHT : "); lcd.print(Suhu_DHT_Battery);
31             lcd.setCursor(0, 2); lcd.print("Humid DHT : "); lcd.print(Kelembapan_DHT_Battery);
32             break;
33             case 4:
34             lcd.setCursor(0, 0); lcd.print("Panel Utama"); lcd.setCursor(19,0); lcd.print(wipistat);
35             lcd.setCursor(0, 1); lcd.print("Volt : "); lcd.print(voltage); lcd.print(" V ");
36             lcd.setCursor(0, 2); lcd.print("Current : "); lcd.print(current); lcd.print(" A ");
37             break;
38             case 5:
39             lcd.setCursor(0, 0); lcd.print("Radiasi"); lcd.setCursor(19,0); lcd.print(wipistat);
40             lcd.setCursor(0,1); lcd.print("Daya : "); lcd.print(DayaIna); lcd.print(" W ");
41             lcd.setCursor(0,2); lcd.print("BusVoltage"); lcd.print("V"); lcd.print(Current_ma); lcd.print("A");
42             lcd.setCursor(0,3); lcd.print("P/A : "); lcd.print(Solar_Radiation); lcd.print(" W/m^2");
43             break;
44         }
45         lcdPage++;
46         if (lcdPage >= totalcdPages) { lcdPage = 0; }
47     }
48     #pragma endregion
49     break;

```

```

1  case Suhu_Battery:
2      lcd.setCursor(0,0); lcd.print("Battery"); lcd.setCursor(19,0); lcd.print(wipistat);
3      lcd.setCursor(0, 1); lcd.print("Suhu 1: "); lcd.print(C1); lcd.print(" C");
4      lcd.setCursor(0, 2); lcd.print("Suhu 2: "); lcd.print(C2); lcd.print(" C");
5      break;
6
7  case Suhu_Oven:
8      lcd.setCursor(0,0); lcd.print("Oven"); lcd.setCursor(19,0); lcd.print(wipistat);
9      if(OvenState){
10         lcd.setCursor(0, 1); lcd.print("Suhu 1: "); lcd.print(C3); lcd.print(" C");
11         lcd.setCursor(0, 2); lcd.print("Suhu 2: "); lcd.print(C4); lcd.print(" C");
12         lcd.setCursor(0, 3); lcd.print("Suhu 3: "); lcd.print(C5); lcd.print(" C");
13     }else{lcd.setCursor(0, 1); lcd.print("Oven Isn't Running!");}
14     break;
15
16 case DHT21_OVEN:
17     lcd.setCursor(0, 0); lcd.print("DHT21_OVEN "); lcd.setCursor(19,0); lcd.print(wipistat);
18     if(OvenState){
19         lcd.setCursor(0, 1); lcd.print("Suhu DHT : "); lcd.print(Suhu_DHT_Oven);
20         lcd.setCursor(0, 2); lcd.print("Humid DHT : "); lcd.print(Kelambapan_DHT_Oven);
21     }else{lcd.setCursor(0, 1); lcd.print("Oven Isn't Running!");}
22     break;
23
24 case DHT21_BATTERY:
25     lcd.setCursor(0, 0); lcd.print("DHT21_BATTERY "); lcd.setCursor(19,0); lcd.print(wipistat);
26     lcd.setCursor(0, 1); lcd.print("Suhu DHT : "); lcd.print(Suhu_DHT_Battery);
27     lcd.setCursor(0, 2); lcd.print("Humid DHT : "); lcd.print(Kelambapan_DHT_Battery);
28     break;
29
30 case Panel_Utama:
31     lcd.setCursor(0, 0); lcd.print("Panel Utama"); lcd.setCursor(19,0); lcd.print(wipistat);
32     lcd.setCursor(0, 1); lcd.print("Volt : "); lcd.print(voltage); lcd.print(" V ");
33     lcd.setCursor(0, 2); lcd.print("Current : "); lcd.print(current); lcd.print(" A ");
34     break;
35
36 case Radiasi:
37     lcd.setCursor(0, 0); lcd.print("Radiasi"); lcd.setCursor(19,0); lcd.print(wipistat);
38     lcd.setCursor(0,1); lcd.print("Daya : "); lcd.print(Dayaina); lcd.print(" W ");
39     lcd.setCursor(0,2); lcd.print("BusVoltage"); lcd.print("V"); lcd.print(Current_mA); lcd.print("A ");
40     lcd.setCursor(0,3); lcd.print("W/A : "); lcd.print(Solar_radiation); lcd.print(" W/m^2");
41     break;
42
43 case Oven_Run:
44     lcd.setCursor(0,0); lcd.print("Run Oven?");
45     lcd.setCursor(0,1); lcd.print("C to Run");
46     lcd.setCursor(0,2); lcd.print("0 Back");
47     OvenRunning = true;
48     break;
49
50 case Oven_Stop:
51     lcd.setCursor(0,0); lcd.print("Stop Oven?");
52     lcd.setCursor(0,1); lcd.print("C to Stop");
53     lcd.setCursor(0,2); lcd.print("0 Back");
54     OvenRunning = false;
55     break;
56
57 case Oven_Running:
58     lcd.setCursor(0,0); lcd.print("Oven is Running!");
59     break;
60
61 case Oven_Stopping:
62     lcd.setCursor(0,0); lcd.print("Oven Stop Running!");
63     break;
64
65 case Restart:
66     lcd.setCursor(0,0); lcd.print("Restarting..");
67     delay(1000);
68     ESP.restart();
69     break;
70
71 case LCD_Restart:
72     lcd.init();
73     lcd.backlight();
74     lcd.clear();
75     layer = LastLayer;
76     break;
77
78 case zzz:
79     break;
80 }
81 lcdReset = layer;
82 }
83 }

```

Gambar 4. 26 Program Interface

Bagian dari program yang menampilkan interface kepada user melalui LCD 20x4 dan Keypad matrix 4x4 sebagai perangkat interaktif. Terdapat menu Home sebagai menu utama yang bisa diakses dengan menekan tombol '0' pada keypad, tekan tombol '1' untuk mengakses menu Suhu_Battery, tekan tombol '2' untuk mengakses menu Suhu_Oven, tekan tombol '3' untuk mengakses menu DHT21_OVEN, tekan tombol '4' untuk mengakses menu DHT21_BATTERY, tekan tombol '5' untuk mengakses menu Panel_Utama, tekan tombol '6' untuk mengakses menu Radiasi, tekan tombol 'A' untuk mengakses menu Changing, tekan tombol 'B' untuk mengakses menu Run atau stop pengering, tekan tombol '*' untuk masuk mode sleep interface, tekan tombol '#' untuk reset LCD, tekan tombol 'D' untuk restart ESP32. Program untuk interface pada LCD dapat dilihat pada Gambar 4. 26. Penjelasan dari masing-masing menu antara lain

4.4.1.6.1 Home

Menu ini merupakan menu utama yang ditampilkan saat sistem pertamakali booting. Pada saat sedang berada di menu lain user dapat kembali ke menu Home dengan menekan tombol '0' pada keypad.

4.4.1.6.2 Changing

Menu ini akan menampilkan semua menu pembacaan sensor secara bergantian setiap 1 detik dengan urutan Suhu Baterai – Suhu Pengering – DHT21 Pengering – DHT21 Ruangan – Pembacaan arus, tegangan dan daya Panel utama – Radiasi matahari. Menu ini dapat diakses dengan menekan tombol 'A' pada keypad.

4.4.1.6.3 Suhu_Battery



Gambar 4. 27 Tampilan LCD pada Menu Suhu Battery

Menu ini menampilkan hasil pembacaan suhu pada baterai yang di ukur oleh 2 thermocouple. Menu ini dapat diakses dengan menekan tombol '1' pada keypad. Gambar tampilan menu suhu battery dapat dilihat pada Gambar 4. 27.

4.4.1.6.4 Suhu_Oven



Gambar 4. 28 Tampilan LCD pada Menu Suhu Oven

Menu ini menampilkan hasil pembacaan suhu pada pengering yang di ukur oleh 3 thermocouple. Menu ini dapat diakses dengan menekan tombol '2' pada keypad. Gambar tampilan menu suhu oven dapat dilihat pada Gambar 4. 28.

4.4.1.6.5 DHT21_OVEN



Gambar 4. 29 Tampilan LCD pada Menu DHT21 OVEN

Menu ini menampilkan hasil pembacaan suhu pada pengering yang di ukur oleh sensor DHT21 di dalam pengering. Menu ini dapat diakses dengan menekan tombol '3' pada keypad. Gambar tampilan menu DHT21 oven dapat dilihat pada Gambar 4. 29.

4.4.1.6.6 DHT21_BATTERY



Gambar 4. 30 Tampilan LCD pada Menu DHT21 BATTERY

Menu ini menampilkan hasil pembacaan suhu pada ruangan yang di ukur oleh sensor DHT21. Menu ini dapat diakses dengan menekan tombol '4' pada keypad. Gambar tampilan menu DHT21 Battery dapat dilihat pada Gambar 4. 30.

4.4.1.6.7 Panel_Utama



Gambar 4. 31 Tampilan LCD pada Menu Panel Utama

Menu ini menampilkan hasil pembacaan dari Arus, Tegangan dan perhitungan daya yang di ukur oleh sensor tegangan dan ACS712 yang terhubung dengan panel utama. Menu ini dapat diakses dengan menekan tombol '5' pada keypad. Gambar tampilan menu panel utama dapat dilihat pada Gambar 4. 31.

4.4.1.6.8 Radiasi



Gambar 4. 32 Tampilan LCD pada Menu Radiasi

Menu ini menampilkan hasil pembacaan daya dan perhitungan radiasi yang di ukur oleh sensor INA219 yang terhubung dengan panel 10w. Menu ini dapat diakses dengan menekan tombol '6' pada keypad. Gambar tampilan menu radiasi dapat dilihat pada Gambar 4. 32.

4.4.1.6.9 OvenRun & OvenStop



Gambar 4. 33 Tampilan LCD pada Menu OvenRun

Menu ini adalah menu untuk menghidupkan pengering. Menu ini dapat diakses dengan menekan tombol 'B' pada keypad pada saat pengering tidak beroperasi. Gambar tampilan menu Oven Run dapat dilihat pada Gambar 4. 33.



Gambar 4. 34 Tampilan LCD pada Menu OvenStop

Menu ini adalah menu untuk mematikan pengering. Menu ini dapat diakses dengan menekan tombol 'B' pada keypad pada saat pengering sedang beroperasi. Gambar tampilan menu Oven Stop dapat dilihat pada Gambar 4. 34.

4.4.1.6.10 Restart

Menu ini adalah menu untuk merestart ESP32. Menu ini dapat diakses dengan menekan tombol ‘D’ pada keypad.

4.4.1.6.11 LCD_Restart

Menu ini adalah menu untuk mereset koneksi LCD ke ESP32 apabila terjadi error komunikasi. Menu ini dapat diakses dengan menekan tombol ‘#’ pada keypad.

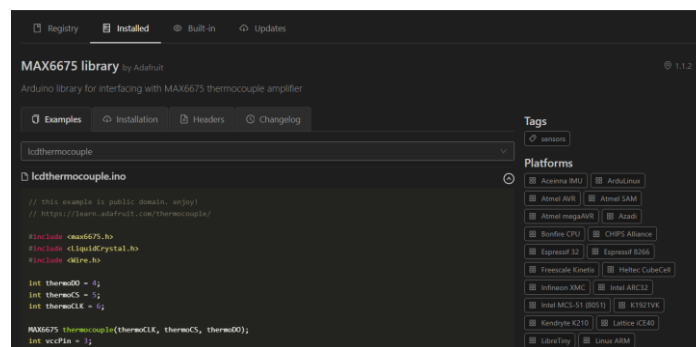
4.4.1.6.12 LCD_Sleep

Menu ini adalah menu untuk mode sleep LCD. Menu ini dapat diakses dengan menekan tombol ‘*’ pada keypad.

4.4.2 Pembacaan Sensor

4.4.2.1 Thermocouple & Max6675

Pembacaan suhu pada thermocouple yang di baca oleh MAX6675 menggunakan MAX6675 library oleh Adafruit untuk mempermudah proses pembuatan program. Library ini dipanggil dengan menuliskan #Include <max6675.h> pada program. Library untuk MAX6675 dapat dilihat pada Gambar 4. 35.



Gambar 4. 35 Library MAX6675

Karena MAX6675 berkomunikasi menggunakan protokol SPI maka dibutuhkan pin Digital Out, pin Clock dan pin Chip Select yang di program sedemikian rupa menyesuaikan dengan ketersediaan pin pada ESP32. Deklarasi data untuk pembacaan dan perhitungan yang berkaitan dengan suhu dari thermocouple juga di tulis di bagian ini. Program pembacaan hasil data thermocouple dapat dilihat pada Gambar 4. 36.

```

1 //MAX6675
2 #define thermoD0 34
3 #define thermoCS1 19
4 #define thermoCS2 18
5 #define thermoCS3 17
6 #define thermoCS4 16
7 #define thermoCS5 33
8 #define thermoCLK 23
9 MAX6675 thermocouple1(thermoCLK, thermoCS1, thermoD0);
10 MAX6675 thermocouple2(thermoCLK, thermoCS2, thermoD0);
11 MAX6675 thermocouple3(thermoCLK, thermoCS3, thermoD0);
12 MAX6675 thermocouple4(thermoCLK, thermoCS4, thermoD0);
13 MAX6675 thermocouple5(thermoCLK, thermoCS5, thermoD0);
14 float C1 = 0;
15 float C2 = 0;
16 float C3 = 0;
17 float C4 = 0;
18 float C5 = 0;
19 float Avg_Bat = 0;
20 float Avg_Ov = 0;
21 float Delta_Bat = 0;
22 float DeltaOldBat = 0;
23 float Delta_Ov = 0;
24 float DeltaOldOv = 0;
25 bool FirstDataBat = true;
26 bool FirstDataOv = true;

```

Gambar 4. 36 Kode Pembacaan Hasil Data Sensor

Pembacaan sensor secara real time dilakukan pada bagian void loop. Pembacaan dilakukan dengan memanggil masing-masing MAX6675 menggunakan kode dari library dan ESP32 akan membaca hasil data dari sensor sesuai dengan urutan dari program. Perhitungan rata-rata suhu dari thermocouple pada oven dan thermocouple pada baterai dilakukan setelah data dari thermocouple terkumpul. Program untuk perhitungan rata-rata suhu thermocouple dapat dilihat pada Gambar 4. 37.

```

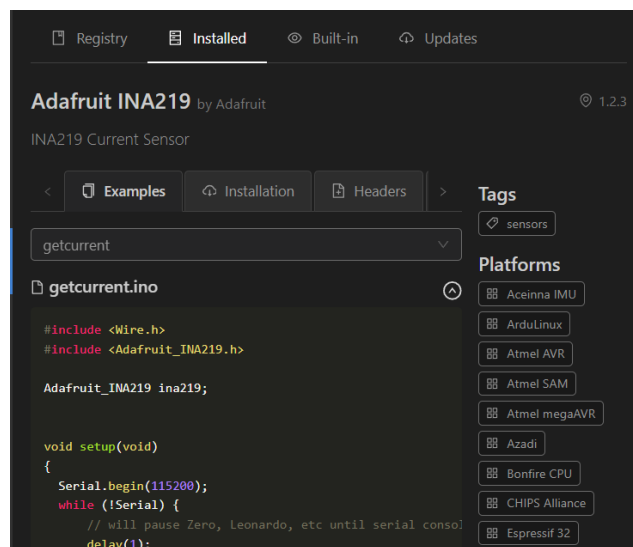
1 void Thermo_Battery(){
2     C1 = thermocouple5.readCelsius();
3     C2 = thermocouple4.readCelsius();
4     Avg_Bat = ((C1+C2)/2);
5 }
6 void Thermo_Oven(){
7     C3 = thermocouple3.readCelsius();
8     C4 = thermocouple2.readCelsius();
9     C5 = thermocouple1.readCelsius();
10    Avg_Ov = ((C3+C4+C5)/3);
11 }

```

Gambar 4. 37 Kode Perhitungan Rata-Rata Suhu Thermocouple

4.4.2.2 INA219

Pembuatan program untuk membaca Tegangan dan Arus pada INA219 menggunakan library Adafruit INA219 untuk mempermudah proses pemrograman. Library yang digunakan untuk INA219 dapat dilihat pada Gambar 4. 38.



Gambar 4. 38 Library INA219

Pendeklarasian data yang akan digunakan dalam proses pembacaan dan perhitungan tegangan, arus dan daya dari sensor INA219 dengan kode program yang dapat dilihat pada Gambar 4. 39.



Gambar 4. 39 deklarasi data sensor

Selanjutnya library perlu memanggil dan memastikan bahwa sensor terhubung dan terdeteksi oleh mikrokontroler pada void setup untuk memastikan

pembacaan data berhasil dengan kode program yang dapat dilihat pada Gambar 4. 40.



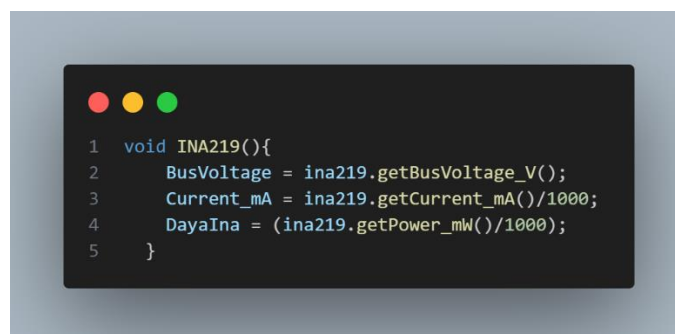
```

1  if (! ina219.begin()) {
2      Serial.println("Failed to find INA219 chip");
3      lcd.clear();
4      lcd.setCursor(0,0);
5      lcd.print("Failed to find ");
6      lcd.setCursor(0,1);
7      lcd.print("INA219 chip");
8      while (1) { delay(10); }
9  }
10

```

Gambar 4. 40 Kode Pengecekan Sensor

Pembacaan sensor ini dilakukan dengan memanggil perintah baca sensor dari library dengan kode program yang dapat dilihat pada Gambar 4. 41.



```

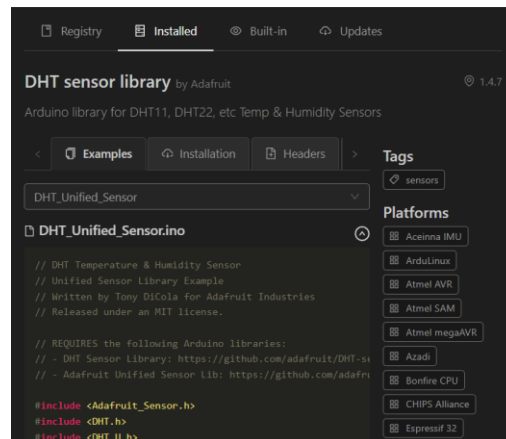
1  void INA219(){
2      BusVoltage = ina219.getBusVoltage_V();
3      Current_mA = ina219.getCurrent_mA()/1000;
4      DayaIna = (ina219.getPower_mW()/1000);
5  }

```

Gambar 4. 41 Kode Pembacaan Hasil Data Sensor INA219

4.4.2.3 DHT21

Pembuatan program untuk membaca suhu dan kelembaban menggunakan library DHT sensor library oleh Adafruit untuk mempermudah proses pembuatan program. Library yang digunakan untuk DHT21 dapat dilihat pada Gambar 4. 42.



Gambar 4. 42 LibraryDHT21

Library perlu mengetahui pin yang digunakan, tipe dan nama yang digunakan untuk memanggil masing-masing sensor DHT21. Kode untuk define pin untuk sensor DHT21 dapat dilihat pada Gambar 4. 43.



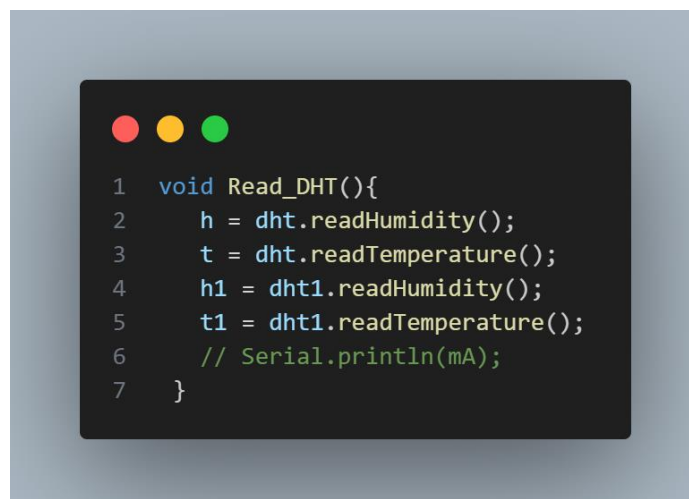
Gambar 4. 43 Define Pin DHT21

Library juga memerlukan inisiasi sensor pada void loop untuk menjalankan proses pembacaan sensor. Program inisiasi sensor DHT21 dapat dilihat pada Gambar 4. 44.



Gambar 4. 44 Inisiasi sensor

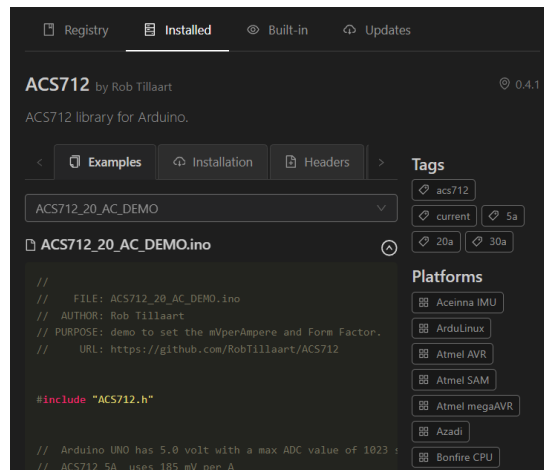
Pembacaan sensor dilakukan dengan menggunakan perintah baca sensor yang disediakan oleh library. Program pembacaan sensor ACS712 dapat dilihat pada Gambar 4. 45.



Gambar 4. 45 Kode Pembacaan Sensor DHT21

4.4.2.4 ACS712

Pembuatan program untuk membaca arus oleh sensor AC712 menggunakan library ACS712 oleh Rob Tillaart untuk mempermudah proses pembuatan program. Library yang digunakan untuk ACS712 dapat dilihat pada Gambar 4. 46.



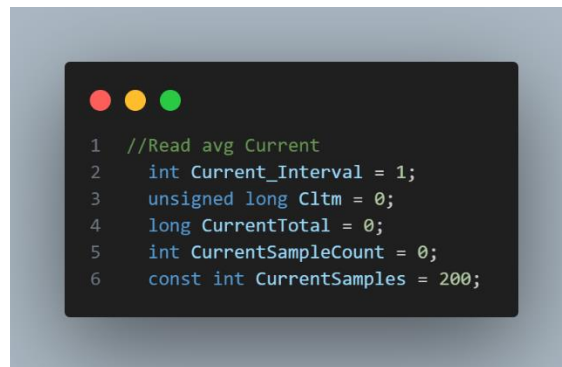
Gambar 4. 46 Library ACS712

Library perlu mengetahui Pin yang digunakan untuk membaca sinyal analog dari sensor, tegangan Vcc pada sensor, resolusi ADC mikrokontroller dan rating arus dari sensor tersebut. Kode untuk define pin untuk sensor ACS712 dapat dilihat pada Gambar 4. 47.



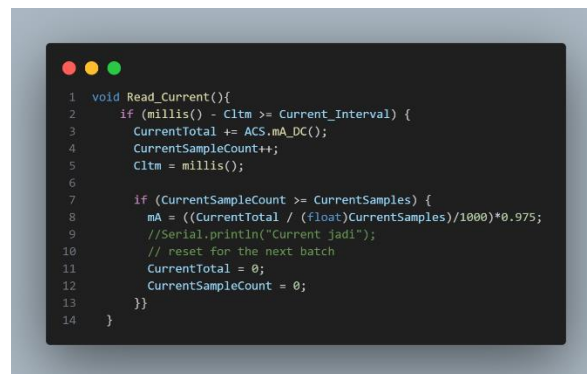
Gambar 4. 47 Define Pin ACS712

Karena pembacaan langsung dari ADC menghasilkan data yang selalu berfluktuasi karena sifat sinyal analog rentan terhadap gangguan (noise). Data pembacaan dibaca sampai dengan 200 sampel lalu dirata-ratakan untuk mendapatkan pembacaan yang stabil. Program pembacaan sensor ACS712 dapat dilihat pada Gambar 4. 48.



Gambar 4. 48 Kode Pembacaan Sensor ACS712

Pembacaan pada sensor dilakukan dengan menggunakan perintah baca sensor dari library dengan program yang dapat dilihat pada Gambar 4. 49.



Gambar 4. 49 Perintah Baca Sensor

4.4.2.5 Voltage Sensor

Pembacaan sensor tegangan dilakukan dengan membaca tegangan pada keluaran voltage divider dengan range 0V-5V yang mewakili tegangan tinggi 0V-105V. Pin yang digunakan untuk pembacaan adalah A0. Define pin untuk pembacaan tegangan dapat dilihat pada Gambar 4. 50.



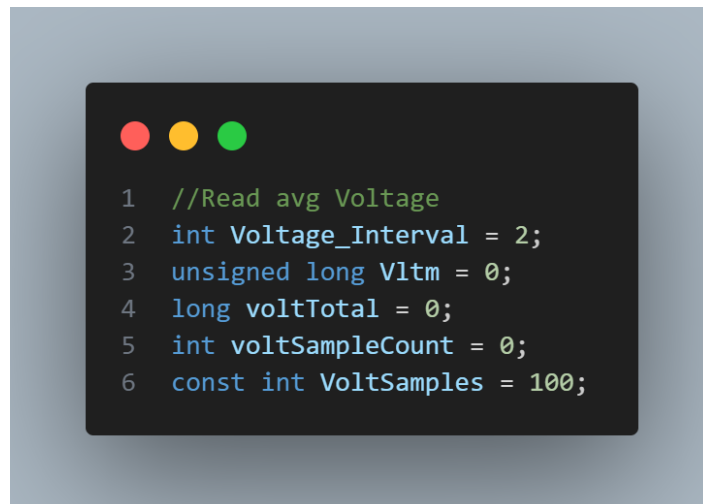
Gambar 4. 50 Define Pin A0

Lalu pin di inisiasi pada void loop supaya pin menjadi mode input dan dapat membaca sinyal analog dari sensor dengan kode program yang dapat dilihat pada Gambar 4. 51.



Gambar 4. 51 Inisiasi Pin pada Void Loop

Karena pembacaan langsung dari ADC menghasilkan data yang selalu berfluktuasi karena sifat sinyal analog rentan terhadap gangguan (noise). Data pembacaan dibaca sampai dengan 200 sampel lalu dirata-ratakan untuk mendapatkan pembacaan yang stabil. Program untuk pembacaan tegangan dapat dilihat pada Gambar 4. 52.



```

1  //Read avg Voltage
2  int Voltage_Interval = 2;
3  unsigned long Vltm = 0;
4  long voltTotal = 0;
5  int voltSampleCount = 0;
6  const int VoltSamples = 100;

```

Gambar 4. 52 Kode Pembacaan Sensor

Pembacaan sensor dilakukan dengan memanggil perintah Analog Read untuk mendapatkan nilai dari sinyal analog pada sensor tegangan. Kemudian hasil dari rata-rata sinyal analog di proses dengan rumus

Untuk mendapatkan tegangan pada pin. Setelah itu dilakukan konversi dari tegangan pada pin menjadi nilai tegangan real dengan menggunakan rasio dari rangkaian pembagi tegangan. Kode program untuk konversi tegangan dapat dilihat pada Gambar 4. 53.



```

1  void Read_Voltage(){
2      if (millis() - Vltm >= Voltage_Interval) {
3          voltTotal += analogRead(Vin);
4          voltSampleCount++;
5          Vltm = millis();
6
7          if (voltSampleCount >= VoltSamples) {
8              avgAdc = voltTotal / (float)VoltSamples;
9              Vp = (avgAdc * 4.98) / 1024.0;
10             Vreal = Vp / 0.0484;
11
12             // reset for the next batch
13             voltTotal = 0;
14             voltSampleCount = 0;
15         }
16     }
17 }

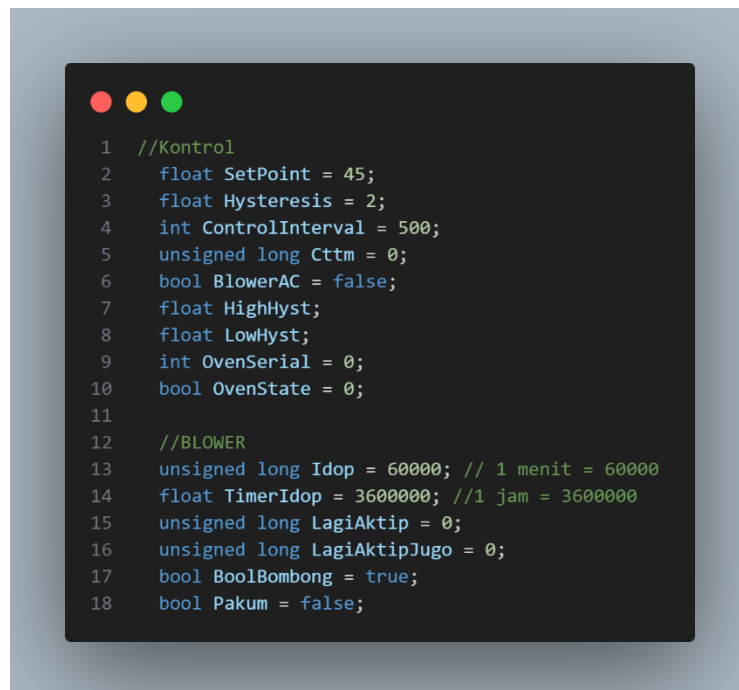
```

Gambar 4. 53 Kode konversi tegangan pada pin ke tegangan real

4.4.3 Aktuator

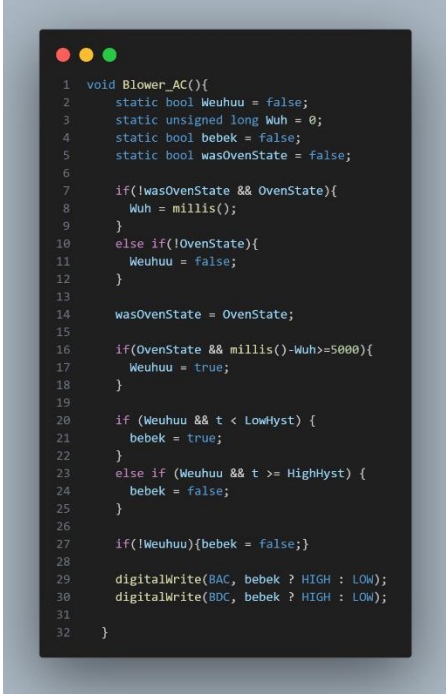
4.4.3.1 Blower AC

Blower di kontrol oleh Arduino berdasarkan pembacaan suhu thermocouple pada ESP32. Blower dikontrol menggunakan kontrol On/Off dengan histerisis ± 2 derajat dari suhu target 45 derajat celcius. Program untuk kontrol blower AC dapat dilihat pada Gambar 4. 54.



Gambar 4. 54 Kontrol Blower AC

Pada bagian deklarasi variabel untuk kontrol blower. Setpoint di set menjadi 45 derajat sesuai dengan target suhu pada pengering, histerisis, interval pengecekan kontrol, variabel state dan pembanding histerisis juga di deklarasikan pada bagian ini.



```

1 void Blower_AC(){
2     static bool Weuhuu = false;
3     static unsigned long Wuh = 0;
4     static bool bebek = false;
5     static bool wasOvenState = false;
6
7     if(!wasOvenState && OvenState){
8         Wuh = millis();
9     }
10    else if(!OvenState){
11        Weuhuu = false;
12    }
13
14    wasOvenState = OvenState;
15
16    if(OvenState && millis()-Wuh>=5000){
17        Weuhuu = true;
18    }
19
20    if (Weuhuu && t < LowHyst) {
21        bebek = true;
22    }
23    else if (Weuhuu && t >= HighHyst) {
24        bebek = false;
25    }
26
27    if(!Weuhuu){bebek = false;}
28
29    digitalWrite(BAC, bebek ? HIGH : LOW);
30    digitalWrite(BDC, bebek ? HIGH : LOW);
31
32 }

```

Gambar 4. 55 Pemantauan Blower

Pada bagian void Blower_AC merupakan bagian yang terus memantau perbandingan antara suhu pengering dan suhu set point untuk memberikan output pada blower. Blower akan di hidupkan apabila suhu pada pengering kurang dari suhu set point dikurang nilai histerisis dan blower akan dimatikan apabila suhu pengering sama dengan nilai set point ditambah nilai histerisis. Program untuk pemantauan kondisi blower dapat dilihat pada Gambar 4. 55.

4.4.4 Komunikasi Mikrokontroller

Komunikasi antara ESP32 dan Arudino Uno bertujuan untuk saling mengirim data sehingga masing-masing mikrokontroller dapat memproses data yang di terima dan memberikan keluaran berdasarkan data tersebut. Komunikasi dilakukan dengan menggunakan protokol komunikasi UART (Universal Asynchronous Receiver-Transmitter).

4.4.4.1 Program Arduino Nano

Arduino Nano menerima data berupa data suhu thermocouple pengering (3 thermocouple) dan kondisi pengering berdasarkan input dari pengguna. Apabila pengguna menjalankan perintah untuk running pengering maka Arduino Nano akan menerima sinyal OvenState = true dan menjalankan algoritma pengontrolan suhu

berdasarkan data thermocouple yang diterima dari ESP32. Arduino Nano juga mengirimkan data pembacaan Tegangan panel, Arus panel, suhu dan kelembaban dari kedua sensor DHT21. Pengiriman data dilakukan setiap 500milidetik. Dan penerimaan data dilakukan setiap saat ada data yang dikirimkan. Setiap Arduino Nano menerima data, LED_BUILTIN akan berkedip menandakan penerimaan data telah berhasil. Kode program untuk pengiriman data oleh Arduino Nano dapat dilihat pada Gambar 4. 56.

```

1 void Transmit(){
2   // Serial.print("Vp/Vreal/I : ");
3   // Serial.print(Vp);
4   // Serial.print("/");
5   Serial2.print(Vreal);
6   Serial2.print("/");
7   Serial2.print(mA);
8   Serial2.print("/");
9   Serial2.print(h);
10  Serial2.print("/");
11  Serial2.print(t);
12  Serial2.print("/");
13  Serial2.print(h1);
14  Serial2.print("/");
15  Serial2.println(t1);
16 }
17 void Receive(){
18   if (Serial2.available()) {
19     digitalWrite(LED_BUILTIN, HIGH);
20     Serial.println("Mashooookk");
21     String data = Serial2.readStringUntil('\n');
22     data.trim();
23
24     int c1 = data.indexOf('/');
25     int c2 = data.indexOf('/', c1 + 1);
26     int c3 = data.indexOf('/', c2 + 1);
27
28     if (c1 > 0 && c2 > c1 && c3 > c2) {
29       Serial.println("Jadeeee");
30       C1 = data.substring(0, c1).toFloat();
31       C2 = data.substring(c1 + 1, c2).toFloat();
32       C3 = data.substring(c2 + 1, c3).toFloat();
33       OvenSerial = data.substring(c3 + 1).toInt();
34       // Serial.print(C1);
35       // Serial.print("/");
36       // Serial.print(C2);
37       // Serial.print("/");
38       // Serial.print(C3);
39     }
40   }
41   else{digitalWrite(LED_BUILTIN, LOW);}
42   if(OvenSerial == 0){OvenState = false;}
43   else{OvenState = true;}
44 }

```

Gambar 4. 56 Kode Pengiriman Data Arduino Nano

4.4.4.2 Program ESP32

Program komunikasi UART pada ESP32 bekerja selayaknya program pada Arduino Nano. ESP32 menerima data dari Arduino Nano berupa Nilai pembacaan Tegangan panel, Arus panel, suhu dan kelembaban dari 2 sensor DHT21. Data yang diterima dari Arduino akan di kirim ke Spreadsheet sebagai logging data yang di upload setiap 5 menit. ESP32 juga mengirim data ke Arduino

Nano berupa pembacaan suhu 3 thermocouple pada pengering dan kondisi OvenState berdasarkan perintah dari interface yang diberikan oleh pengguna. LED_BUILTIN pada ESP32 akan berkedip setiap kali data dari Arduino Nano berhasil di terima. Kode program untuk penerimaan data oleh ESP32 dapat dilihat pada Gambar 4. 57



```

1 void Receive(){
2   if (Serial2.available()) {
3     digitalWrite(LED_BUILTIN, HIGH);
4     String data = Serial2.readStringUntil('\n');
5     data.trim();
6
7     int c1 = data.indexOf('/');
8     int c2 = data.indexOf('/', c1 + 1);
9     int c3 = data.indexOf('/', c2 + 1);
10    int c4 = data.indexOf('/', c3 + 1);
11    int c5 = data.indexOf('/', c4 + 1);
12
13    if (c1 > 0 && c2 > c1 && c3 > c2 && c4 > c3 && c5 > c4) {
14      voltage      = data.substring(0, c1).toFloat();
15      current      = data.substring(c1 + 1, c2).toFloat();
16      Kelembapan_DHT_Oven = data.substring(c2 + 1, c3).toFloat();
17      Suhu_DHT_Oven    = data.substring(c3 + 1, c4).toFloat();
18      Kelembapan_DHT_Battery = data.substring(c4 + 1, c5).toFloat();
19      Suhu_DHT_Battery   = data.substring(c5 + 1).toFloat();
20    } else {
21      Serial.print("Data format tidak valid: " );
22      Serial.println(data);
23    }
24    else{digitalWrite(LED_BUILTIN, LOW);}
25  void Transmit(){
26    //Transmit
27    Serial2.print(C3);
28    Serial2.print("/");
29    Serial2.print(C4);
30    Serial2.print("/");
31    Serial2.print(C5);
32    Serial2.print("/");
33    Serial2.println(OvenState);
34  }

```

Gambar 4. 57 Kode Penerimaan Data ESP32

4.4.5 Perhitungan

4.4.5.1 Delta Suhu

Delta suhu merupakan pertambahan suhu setiap waktu pembacaan pada logging data. Besaran delta suhu didapatkan dari rumus

Pada program perhitungan delta suhu, setiap data diupload ESP32 akan menyimpan data tersebut untuk perhitungan pada data selanjutnya dan membagi hasil pengurangan data sekarang dengan data sebelumnya lalu membagi dengan

besaran waktu interval upload yang sudah dikonversi dari milidetik ke menit. Program untuk perhitungan delta suhu dapat dilihat pada Gambar 4. 58.



Gambar 4. 58 Kode Perhitungan Delta Suhu

4.4.5.2 Radiasi Matahari

Perhitungan Radiasi Matahari didapatkan dari data mentah hasil pembacaan arus, tegangan dan daya dari panel 10W yang dibaca oleh sensor INA219. Konversi dari daya panel menjadi Radiasi Matahari menggunakan rumus

Daya Panel = pembacaan daya oleh INA219

Luas panel = 0.061425

Efisiensi panel= 17.46

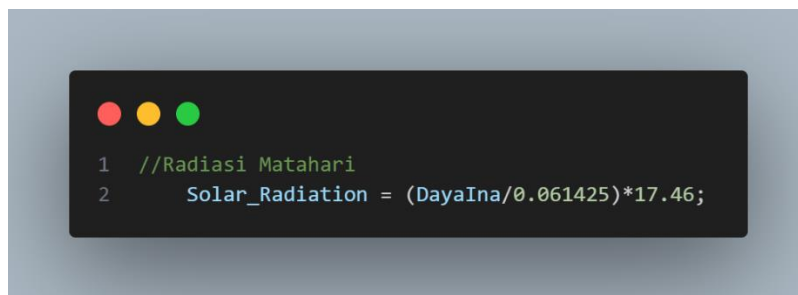
Luas panel didapatkan dari perkalian luas solar cell pada panel surya yaitu 0,19m dan 0,32m didapatkan 0,0614m².

Gambar pengukuran perhitungan luas panel dapat dilihat pada Gambar 4. 59.



Gambar 4. 59 Pengukuran Luas Panel

Lalu efisiensi panel didapatkan dari datasheet panel. Namun menggunakan besaran efisiensi dari datasheet masih belum menghasilkan pembacaan yang akurat. Untuk mendapatkan nilai efisiensi yang real diperlukan nilai radiasi matahari pada saat pembacaan daya panel dilakukan, perhitungan menggunakan rumus pada program perhitungan menjadi. dapat dilihat pada Gambar 4. 60.

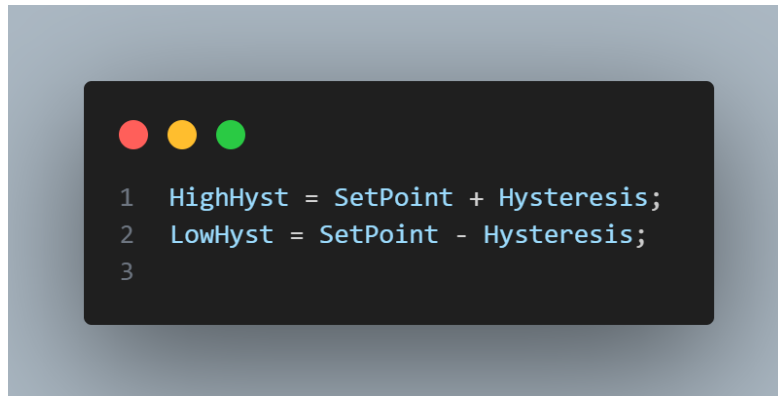


Gambar 4. 60 Kode Perhitungan Radiasi Sinar Matahari

4.4.5.3 Histerisis

Perhitungan histerisis diperlukan untuk mendapatkan nilai batas hidup dan batas mati blower. Histerisis terbagi menjadi 2 yaitu HighHyst dengan rumus

Setpoint + Histerisis dan LowHyst dengan rumus Setpoint – Histerisis. Program Perhitungan histeris dapat dilihat pada Gambar 4. 61.



Gambar 4. 61 Kode Perhitungan Histerisis

4.4.6 Data Logging

Upload data logging ke spreadsheet terbagi menjadi data pada baterai pasir dan data pada pengering. Spreadsheet dibedakan menjadi 2 untuk memudahkan proses analisa data. Spreadsheet baterai pasir akan selalu mengupload data setiap 5 menit apabila sistem sudah di hidupkan dan terhubung dengan wifi sedangkan spreadsheet pengering hanya akan melakukan proses upload apabila perintah untuk menjalankan pengering diberikan oleh user. Proses data loggin juga mengupload header data ke masing-masing spreadsheet setiap kali ESP32 di hidupkan atau di restart.

4.4.6.1 SpreadSheet Boot

Spreadsheet Boot akan mengupload header yang menampilkan data apa saja yang di upload ke masing-masing spreadsheet. Pada spreadsheet baterai header data bertuliskan Jam, Suhu 1, Suhu 2, Suhu rerata, Delta T, Tegangan, Arus, Daya, Radiasi, V INA, A INA, P INA, Suhu DHT, Kelembaban.

Jam	Suhu 1	Suhu 2	Suhu rerata	Delta T	Tegangan	Arus	Daya	Radiasi	V INA	A INA	P INA	Suhu DHT	Kelembaban
03 Sep 2026, 17:58	19.25	27.5	23.375	0	0.02	-24.2	-0.484	0	0	0	0	29.1	71.3

Pada Spreadsheet pengering header data berisikan Jam, Suhu 1, Suhu 2, Suhu 3, Suhu rerata, Delta T, Suhu DHT, Kelembaban.

Jam	Suhu 1	Suhu 2	Suhu 3	Suhu Rerata	Delta T	Suhu DHT	Kelembaban
03 Sep 2026, 17:58	26.5	26.25	26.75	26.5	0	28.5	95.4

Header diupload setiap booting berujuan untuk memberikan batas pada spreadsheet dan juga menjadi indikasi jika terjadi gangguan pada sistem yang

[illegible]

Gambar 4. 62 Kode SpreadSheet Boot

Pada upload data pengering. Proses upload hanya dilakukan saat pengering sedang beroperasi dengan tujuan menghemat token spreadsheet. Data akan di upload setiap 5 menit apabila pengering beroperasi dan ESP32 terhubung ke internet melalui wifi. Data monitoring pengering yang diupload ke spreadsheet adalah waktu realtime yang didapatkan dari epoch time, pembacaan suhu

thermocouple pada pengering, rata-rata suhu dari 3 thermocouple, delta suhu dari rata-rata, pembacaan suhu dan kelembapan dari sensor DHT21. Kode program spreadsheet pengering dapat dilihat pada gambar 4. 63.

```

1 void Gsheet_Oven() {
2   waktuStr = getFormattime();
3   Serial.println("Uploading ke Oven");
4
5
6   FirebaseJson response;
7   // Serial.println("update spreadsheet values...");
8   Serial.println("-----");
9   FirebaseJson valueRange;
10  valueRange.set("majorDimension", "ROWS");
11  valueRange.set("values[0][0]", waktuStr); // row 1/ column 1
12  valueRange.set("values[0][1]", C1); // row 2/ column 1
13  valueRange.set("values[0][2]", C2);
14  valueRange.set("values[0][3]", C3);
15  valueRange.set("values[0][4]", Avg_Ov);
16  valueRange.set("values[0][5]", Delta_Ov);
17  valueRange.set("values[0][6]", Suhu_Oven);
18  valueRange.set("values[0][7]", Kelembapan_Oven);
19  //valueRange.set("values[0][8]", "Day");
20
21  bool success = gsheet.values.append(response /* returned response */, spreadsheetID_Oven /* spreadsheet id to update */, "sheet1!A:" /* range to update */, valueRange /* data to update */);
22
23  if (success) {
24    Serial.println("Upload Oven berhasil");
25    response.toString(Serial, true);
26  } else {
27    Serial.println("Upload Oven gagal");
28    Serial.println(gsheet.errorReason());
29  }
30 }

```

Gambar 4. 63 Kode SpreadSheet Pengering

4.4.6.3 SpreadSheet Baterai Pasir

Pada upload data baterai pasir. Proses upload dilakukan setiap 5 menit seama sistem beroperasi dan terhubung dengan wifi. Data monitoring baterai pasir yang diupload ke spreadsheet adalah waktu realtime yang didapatkan dari epoch time, pembacaan suhu thermocouple pada baterai, rata-rata suhu dari 2 thermocouple, delta suhu dari rata-rata, Tegangan, Arus, Daya dari panel utama, radiasi matahari, Tegangan, Arus, Daya dari panel 10W, pembacaan suhu dan kelembapan dari sensor DHT21. Kode program spreadsheet baterai pasir dapat dilihat pada gambar 4. 64.

```

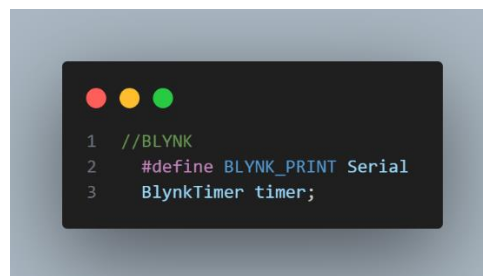
1 void Gsheet_Baterai() {
2   waktuStr = getFormattime();
3   Serial.println("Uploading ke Baterai");
4
5
6   FirebaseJson response;
7   // Serial.println("update spreadsheet values...");
8   Serial.println("-----");
9   FirebaseJson valueRange;
10  valueRange.set("majorDimension", "ROWS");
11  valueRange.set("values[0][0]", waktuStr); // row 1/ column 1
12  valueRange.set("values[0][1]", C1); // row 2/ column 1
13  valueRange.set("values[0][2]", C2);
14  valueRange.set("values[0][3]", Avg_Bat);
15  valueRange.set("values[0][4]", Volt_Bat);
16  valueRange.set("values[0][5]", Volt_Amp);
17  valueRange.set("values[0][6]", Volt_Amp_Current);
18  valueRange.set("values[0][7]", Suhu_Baterai);
19  valueRange.set("values[0][8]", Suhu_Panel);
20  valueRange.set("values[0][9]", Suhu_Panel_10W);
21  valueRange.set("values[0][10]", Suhu_Pem_Baterai);
22  valueRange.set("values[0][11]", Kelembapan_Bat_Baterai);
23
24  bool success = gsheet.values.append(response /* returned response */, spreadsheetID_Baterai /* spreadsheet id to update */, "sheet1!A:" /* range to update */, valueRange /* data to update */);
25
26  if (success) {
27    Serial.println("Upload Battery berhasil");
28    response.toString(Serial, true);
29  } else {
30    Serial.println("Upload Battery gagal");
31    Serial.println(gsheet.errorReason());
32  }
33 }

```

Gambar 4. 64 Kode SpreadSheet Baterai Pasir

4.4.7 Dashboard

Perancangan perangkat lunak antarmuka pengguna dikembangkan menggunakan platform Blynk IoT untuk memfasilitasi pemantauan parameter operasional alat pengering secara *real-time*. Sistem ini mengandalkan mikrokontroler ESP32 sebagai unit akuisisi data utama. Agar pengiriman data dalam jumlah masif ini tidak mengganggu atau menunda rutinitas pembacaan sensor, perancangan perangkat lunak menggunakan metode penanda waktu (*timer interval*) terstruktur sebagai pengganti fungsi penundaan konvensional. Langkah awal perancangan dimulai dengan mendefinisikan pustaka dan objek yang dibutuhkan, seperti yang ditunjukkan pada Gambar 4. 65. Baris program diawali dengan instruksi `#define BLYNK_PRINT Serial` untuk mengaktifkan fitur pemantauan status koneksi melalui *Serial Monitor*. Selanjutnya, objek penanda waktu dideklarasikan dengan sintaks `BlynkTimer timer;`. Penggunaan objek *timer* bawaan Blynk ini bersifat krusial untuk menjalankan fungsi *multitasking*, sehingga instruksi pengiriman data dapat dieksekusi tanpa menghentikan proses eksekusi kode utama di dalam sistem.



Gambar 4. 65 Potongan Program Deklarasi Objek Blynk

Sebelum mikrokontroler dapat mengirimkan data ke platform *dashboard*, perangkat harus melalui tahapan autentikasi dan sinkronisasi dengan *server* pusat Blynk seperti yang terlihat pada Gambar 4. 66. Konfigurasi koneksi diletakkan di dalam fungsi inisialisasi jaringan dengan menerapkan kondisi keamanan `if(WiFi.status() == WL_CONNECTED)`. Kondisi ini memastikan bahwa perangkat hanya akan mencoba terhubung ke Blynk jika ESP32 telah sukses terkoneksi dengan jaringan WiFi. Apabila jaringan tersedia, instruksi

Blynk.config(BLYNK_AUTH_TOKEN) akan mengatur token keamanan unik perangkat, dilanjutkan dengan perintah Blynk.connect() untuk membangun jembatan komunikasi ke *server*. Pada akhir blok logika tersebut, instruksi timer.setInterval(1000L, myTimerEvent) diberikan untuk memerintahkan mikrokontroler mengeksekusi sub-rutin pengiriman data secara otomatis setiap 1000 milidetik atau 1 detik sekali.



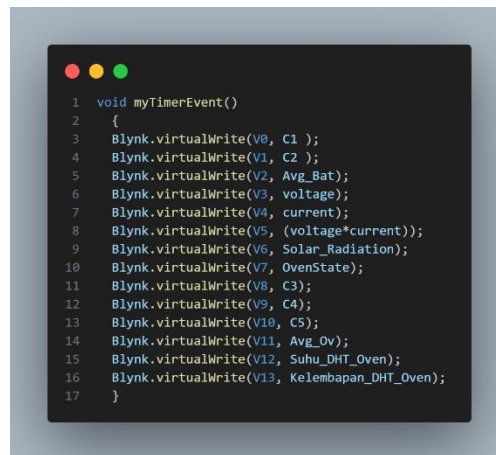
```

1 //blynk
2 if(WiFi.status() == WL_CONNECTED){
3   Blynk.config(BLYNK_AUTH_TOKEN);
4   Blynk.connect();
5   timer.setInterval(1000L, myTimerEvent);
6 }

```

Gambar 4. 66 Potongan Program Inisialisasi Koneksi Blynk

Proses akuisisi data dan pengiriman ke antarmuka pengguna dieksekusi secara terpusat di dalam fungsi penanda waktu yang memuat seluruh protokol transmisi, seperti yang ditunjukkan pada Gambar 4. 67. Mikrokontroler memanfaatkan perintah Blynk.virtualWrite() untuk memetakan data hasil komputasi sensor ke alamat *Virtual Pin* secara paralel. Pengalokasian pin ini mencakup data tegangan, arus, dan kalkulasi daya pada pin V2 hingga V5, data radiasi surya, hingga parameter suhu berlapis pada V8 hingga V10 beserta kelembapan ruang pengering. Pembagian pin yang independen ini secara efektif mencegah terjadinya tumpang tindih paket data saat transmisi jaringan berlangsung. Program untuk pengiriman data ke virtual dapat dilihat pada Gambar 4. 67.



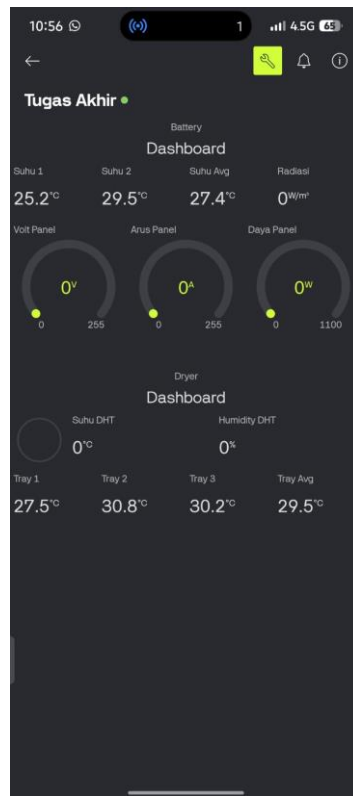
```

1 void myTimerEvent()
2 {
3   Blynk.virtualWrite(V0, C1 );
4   Blynk.virtualWrite(V1, C2 );
5   Blynk.virtualWrite(V2, Avg_Bat);
6   Blynk.virtualWrite(V3, voltage);
7   Blynk.virtualWrite(V4, current);
8   Blynk.virtualWrite(V5, (voltage*current));
9   Blynk.virtualWrite(V6, Solar_Radiation);
10  Blynk.virtualWrite(V7, OvenState);
11  Blynk.virtualWrite(V8, C3);
12  Blynk.virtualWrite(V9, C4);
13  Blynk.virtualWrite(V10, C5);
14  Blynk.virtualWrite(V11, Avg_Ov);
15  Blynk.virtualWrite(V12, Suhu_DHT_Oven);
16  Blynk.virtualWrite(V13, Kelembapan_DHT_Oven);
17 }

```

Gambar 4. 67 Potongan Program Pengiriman Data ke Virtual Pin

Seluruh aliran data yang dipancarkan secara berkesinambungan melalui *Virtual Pin* kemudian ditangkap oleh aplikasi *mobile* Blynk dan divisualisasikan dalam bentuk antarmuka grafis seperti pada Gambar 4. 68. Implementasi antarmuka pada *dashboard* ini disusun secara vertikal pada satu halaman utama (*scrollable interface*) untuk mempermudah navigasi pemantauan tanpa membebani kinerja perangkat. Antarmuka tersebut secara visual terbagi menjadi dua sektor pemantauan yang komprehensif, yaitu sektor hulu dengan label "Battery Dashboard" dan sektor hilir dengan label "Dryer Dashboard". Pada bagian atas, metrik kelistrikan panel surya ditampilkan menggunakan indikator melingkar (*widget gauge*). Sementara itu, pada bagian hilir, parameter persebaran kalor di dalam kabinet oven, seperti Tray 1 hingga Tray 3, suhu rata-rata, dan indikator sensor DHT, direpresentasikan secara presisi melalui indikator nilai numerik (*labeled value*).



Gambar 4. 68 Hasil Implementasi Antarmuka Dashboard Blynk IoT