

BAB IV

PROSES PEMBUATAN ALAT

4.1 Pembuatan Hardware

Pembuatan sistem *floodlight* COB LED 100 W dilakukan berdasarkan rancangan *hardware* dan *software* yang telah disusun pada bab sebelumnya. Tahap pembuatan ini bertujuan untuk merealisasikan rancangan menjadi alat yang dapat bekerja secara nyata, yaitu mampu mengatur kecerahan lampu menggunakan metode PWM, membaca kondisi kerja alat, menampilkan data pada LCD, serta mengirimkan data pengukuran ke Google Spreadsheet.

Secara umum, sistem terdiri dari bagian catu daya, bagian input, bagian proses, dan bagian output. Bagian catu daya berfungsi menyediakan sumber tegangan untuk seluruh komponen. Bagian input terdiri dari potensiometer, LDR, sensor suhu DS18B20, dan sensor INA226. Bagian proses dilakukan oleh ESP32 sebagai pusat kendali, sedangkan bagian output berupa lampu COB LED 100 W, LCD 16x2 I2C, kipas pendingin, dan pencatatan data pada Google Spreadsheet.

Pada tahap pembuatan, setiap rangkaian diperiksa terlebih dahulu sebelum digabungkan menjadi satu sistem. Pemeriksaan dilakukan untuk memastikan tegangan keluaran *power supply*, modul *step-down*, dan *boost converter* sesuai dengan kebutuhan rangkaian. Selain itu, jalur *ground* pada seluruh komponen dibuat menjadi satu atau *common ground* agar pembacaan sensor, komunikasi I2C, dan sinyal PWM dapat bekerja stabil.

4.1.1 Alat dan Bahan Pembuatan Hardware

Alat pembuatan *hardware* digunakan untuk membantu proses perakitan, penyambungan, pengukuran, dan pemeriksaan rangkaian. Alat-alat ini diperlukan agar setiap sambungan pada sistem dapat dibuat dengan rapi dan aman. Selain itu, alat ukur seperti multimeter digunakan untuk memastikan nilai tegangan pada setiap bagian rangkaian sudah sesuai sebelum seluruh komponen dihubungkan menjadi satu sistem.

Bahan dan komponen *hardware* merupakan bagian utama yang digunakan dalam pembuatan alat. Komponen dipilih sesuai fungsi yang dibutuhkan, seperti sumber daya, pusat kendali, sensor, pengatur kecerahan lampu, tampilan data, pendingin, dan bagian mekanik. Jumlah bahan dicantumkan agar kebutuhan komponen dalam pembuatan sistem lebih mudah diketahui. Selain komponen elektronik, digunakan juga bahan mekanik seperti box alat,udukan 3D print, pipa kaki-kaki, baut, spacer, spiral kabel, dan toples penutup LED. Bahan tersebut berfungsi untuk membantu pemasangan rangkaian, menjaga posisi komponen, melindungi bagian lampu, serta membuat alat lebih rapi dan stabil saat digunakan.

Tabel 4- 1 Daftar alat untuk pembuatan hardware

No.	Nama Alat	Fungsi
1	Multimeter digital	Digunakan untuk mengukur tegangan keluaran PSU, MP1584, <i>boost converter</i> , serta memeriksa jalur <i>ground</i> dan sambungan rangkaian.
2	Solder dan Timah Solder	Digunakan untuk menyambungkan kabel, resistor, dan komponen pendukung agar koneksi lebih kuat dan tidak mudah lepas.
3	Tang potong dan tang kupas kabel	Digunakan untuk memotong kabel sesuai kebutuhan dan mengupas isolasi kabel sebelum disambungkan ke komponen.
4	Obeng	Digunakan untuk mengencangkan terminal <i>power supply</i> , terminal converter, terminal kabel, serta memasang baut pada box.
5	Gergaji pipa kecil	Memotong pipa kecil yang digunakan sebagai kaki-kaki atau penopang alat agar ukurannya sesuai rancangan.
6	Cutter/gunting kecil	Digunakan untuk memotong isolasi, kabel kecil, atau bahan pendukung lain saat proses perakitan.
7	Printer 3D atau jasa 3D print	Membuatudukan, penyangga, atau bagian mekanik alat sesuai ukuran desain.

Tabel 4- 2 Bahan dan komponen hardware

No.	Bahan	Spesifikasi	Jumlah	Fungsi
1	COB LED	100 W	1 buah	Sumber cahaya utama sistem <i>floodlight</i> .
2	<i>Power supply</i> Luminox	12 V 10 A	1 buah	Mengubah AC PLN menjadi DC 12 V untuk suplai utama.
3	<i>Boost converter</i>	150 W	1 buah	Menaikkan tegangan dari PSU untuk kebutuhan lampu COB LED.
4	MP1584 <i>Step-down</i>	Output 5 V	1 buah	Menurunkan 12 V menjadi 5 V untuk ESP32 dan LCD.
5	ESP32	ESP-WROOM-32	1 buah	Pusat kendali sensor, PWM, LCD, Wi-Fi, dan pengiriman data.
6	MOSFET IRF540	N-channel	1 buah	Saklar elektronik pengatur arus lampu melalui PWM.
7	INA226	I2C, 0x44	1 buah	Membaca tegangan dan arus pada jalur lampu.
8	DS18B20	Sensor suhu digital	1 buah	Membaca suhu pada area lampu atau <i>heatsink</i> .
9	LDR	Sensor cahaya	1 buah	Membaca perubahan intensitas cahaya lampu.
10	Potensiometer	Input analog	1 buah	Mengatur nilai PWM secara manual.
11	LCD 16x2 I2C	Alamat 0x27	1 buah	Menampilkan suhu, LDR, PWM, tegangan, dan arus.
12	Kipas prosesor + <i>heatsink</i>	12 V	1 buah	Membantu pendinginan lampu dan <i>heatsink</i> .
13	Resistor	220 Ω	1 buah	Resistor pada jalur gate MOSFET.

No.	Bahan	Spesifikasi	Jumlah	Fungsi
14	Resistor	10 k Ω	2 buah	Pull-down gate MOSFET dan pembagi tegangan LDR.
15	Resistor	4,7 k Ω	1 buah	<i>Pull-up</i> pada jalur data DS18B20.
16	Kabel jumper	-	1 meter	Menghubungkan rangkaian sensor dan jalur daya.
17	Kabel hitam & merah	AWG 20	2 meter	Menghubungkan rangkaian ke LDR
18	Kabel hitam	AWG 22	2 meter	Menghubungkan kipas ke catu daya.
19	Kabel hitam & merah	AWG 16	4 meter	Menghubungkan rangkaian ke lampu COB.
20	Kabel hitam	NYAF 1,5	4 meter	Menghubungkan rangkaian ke <i>boost converter</i> & catu daya.
21	Terminal block	2 Pin	2 buah	Merapikan sambungan kabel catu daya dan beban.
22	JST XH 2,54 mm	4 Pin	2 buah	Konektor sambungan LCD I2C agar jalur VCC, GND, SDA, dan SCL lebih rapi serta mudah dilepas-pasang.
23	JST XH 2,54 mm	3 Pin	1 buah	Konektor sambungan sensor DS18B20.
24	JST XH 2,54 mm	2 Pin	1 buah	Konektor sambungan LDR
25	Saklar	ON/OFF mini 3 pin	1 buah	Untuk menghidupkan dan mematikan kipas pendingin sesuai kebutuhan penggunaan alat.

No.	Bahan	Spesifikasi	Jumlah	Fungsi
26	Baut	2 cm, 5 cm	10 buah	Untuk mengencangkan komponen, dudukan, spacer, dan bagian mekanik alat agar terpasang kuat.
27	Spacer	5 mm	4 buah	Sebagai dudukan PCB agar posisi rangkaian lebih stabil.
28	Spiral kabel	-	1 meter	Untuk merapikan jalur kabel agar susunan kabel tidak berantakan.
29	Pipa Conduit	18 mm	1 buah	Untuk kaki-kaki penyangga agar alat stabil dan memiliki ruang udara di bagian bawah.
30	3D Print	Sesuai desain	1 set	Sebagai tempat utama untuk meletakkan rangkaian, komponen elektronik, kabel, serta bagian pendukung sistem.
31	Toples kaca	Diameter 7,5 cm	1 buah	Sebagai penutup atau pelindung bagian lampu COB LED agar area LED lebih aman.
32	Project Box	15x9x6 cm	1 buah	Sebagai tempat penyimpanan atau pelindung komponen.
33	Isolasi dan heat shrink	Bahan pendukung	1 buah	Membantu penyambungan dan pelindungan kabel.

4.1.2 Langkah-Langkah Pembuatan Hardware

Pembuatan *hardware* dilakukan secara berurutan mulai dari persiapan komponen, pembuatan bagian mekanik, perakitan catu daya, pemasangan rangkaian kendali, pemasangan sensor, pemasangan lampu, hingga pemeriksaan

akhir sambungan. Urutan pembuatan *hardware* dibuat agar proses perakitan lebih mudah dikontrol dan kesalahan sambungan dapat diminimalkan.

Tahap awal pembuatan *hardware* adalah menyiapkan seluruh komponen yang akan digunakan. Komponen yang diperiksa meliputi PSU 12 V 10 A, *boost converter*, MP1584, ESP32, sensor, MOSFET, LCD, kipas, *heatsink*, kabel, resistor, terminal block,udukan 3D print, dan pipa kecil untuk kaki-kaki alat. Setiap komponen dipastikan dalam kondisi baik agar tidak menghambat proses perakitan.

A. Pembuatan 3D Print, Box, dan Kaki-Kaki Alat

Pembuatan bagian mekanik diawali dengan membuatudukan atau holder menggunakan 3D print. Dudukan hasil 3D print pada alat ini digunakan untuk menyangga beberapa komponen utama agar posisinya lebih stabil dan sesuai dengan tata letak yang telah dirancang. Pada bagian atas, dudukan 3D print digunakan sebagai penyangga lampu COB LED, kipas, *heatsink*, dan bagian *bus* atau jalur sambungan pendukung. Penempatan komponen tersebut pada bagian atas bertujuan agar lampu, pendingin, dan jalur sambungan dapat tersusun lebih rapi serta tidak mudah bergeser saat alat digunakan. Sementara itu, pada bagian bawah dudukan ditempatkan sensor LDR agar sensor dapat membaca perubahan intensitas cahaya dari lampu dengan lebih sesuai. Dengan adanya dudukan 3D print ini, pemasangan komponen menjadi lebih terarah, posisi antarbagian lebih stabil, dan bentuk keseluruhan alat terlihat lebih rapi.



Gambar 4- 1 Proses Pembuatan 3D Print

Setelah dudukan 3D print selesai dibuat, proses dilanjutkan dengan menyiapkan box alat sebagai badan utama sistem. Box digunakan sebagai tempat pemasangan rangkaian, komponen, kabel, serta bagian pendukung lainnya agar alat terlihat lebih rapi dan aman digunakan. Pada tahap ini, posisi komponen pada box ditentukan terlebih dahulu, terutama bagian saklar, LCD, jalur kabel, dan area pemasangan komponen utama. Penentuan posisi ini penting agar setiap komponen tidak saling mengganggu dan tetap mudah diakses ketika dilakukan pemeriksaan atau perbaikan.

Pada bagian panel box, dibuat lubang untuk pemasangan saklar ON/OFF kipas menggunakan proses laser cutting. Pembuatan lubang dengan laser cutting bertujuan agar ukuran lubang lebih presisi, sesuai dengan ukuran saklar, dan hasil potongannya lebih rapi. Dengan adanya lubang yang presisi, saklar dapat terpasang lebih kuat pada badan alat dan tidak mudah bergeser saat digunakan. Selain itu, posisi saklar dibuat pada bagian yang mudah dijangkau sehingga pengguna dapat menghidupkan dan mematikan kipas dengan lebih mudah. Selain saklar, lubang untuk pemasangan potensiometer dibuat menggunakan bor sesuai ukuran batang potensiometer agar komponen dapat terpasang dengan baik pada panel box.



Gambar 4- 2 Proses Laser pada box

Tahap berikutnya adalah menyiapkan pipa kecil yang digunakan sebagai kaki-kaki alat. Pipa diukur terlebih dahulu sesuai tinggi yang dibutuhkan, kemudian dipotong menggunakan gergaji pipa kecil agar setiap kaki memiliki ukuran yang seragam. Ukuran kaki-kaki dibuat sama agar posisi box tetap seimbang dan tidak miring saat diletakkan di permukaan meja. Setelah pipa selesai dipotong, bagian tepi pipa dirapikan agar tidak tajam dan lebih aman saat dipasang pada alat. Setelah seluruh bagian mekanik selesai disiapkan, dudukan hasil 3D print dan pipa kaki-kaki disatukan sesuai posisi yang telah ditentukan. Dudukan 3D print dipasang untuk menopang bagian tertentu dari alat, sedangkan pipa kaki-kaki dipasang pada bagian bawah dudukan 3D print sebagai penyangga.

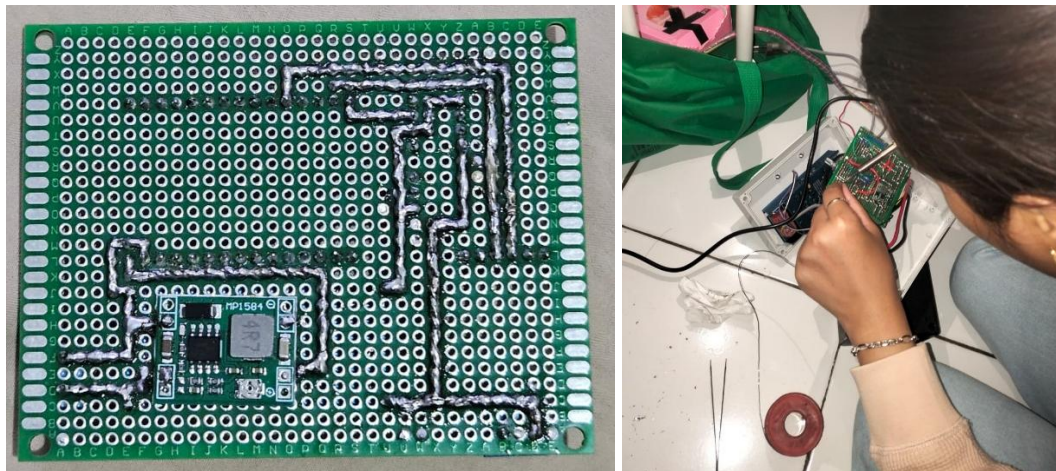


Gambar 4- 3 Dudukan hasil 3D dan kaki pipa

B. Pemasangan Komponen pada PCB

Pemasangan komponen pada PCB dilakukan setelah tata letak rangkaian dan jalur sambungan ditentukan. Pada tahap ini, komponen utama seperti ESP32, MP1584, INA226, MOSFET IRF540, terminal block, konektor JST, resistor, serta rangkaian pendukung lainnya ditempatkan terlebih dahulu pada posisi yang telah dirancang. Penempatan komponen dilakukan dengan memperhatikan fungsi masing-masing komponen agar jalur kabel lebih mudah disusun dan tidak saling bertumpuk.

Setelah posisi komponen sesuai, setiap kaki komponen dipasang pada lubang PCB atau titik sambungan yang telah disiapkan. Komponen kemudian disolder satu per satu agar terhubung kuat dengan jalur rangkaian. Proses penyolderan dilakukan secara hati-hati supaya tidak terjadi hubungan singkat antarjalur. Bagian kaki komponen yang terlalu panjang dipotong setelah proses solder selesai agar bagian bawah PCB terlihat lebih rapi dan tidak mengenai jalur lain.



Gambar 4- 4 Proses penyolderan PCB

Pada PCB, rangkaian kendali disusun agar ESP32 dapat terhubung dengan sensor dan modul pendukung. Potensiometer, LDR, DS18B20, INA226, LCD, dan MOSFET dihubungkan sesuai konfigurasi pin yang telah dirancang. Konektor JST digunakan pada beberapa sambungan agar kabel sensor dan modul lebih mudah dilepas-pasang saat perakitan atau pemeriksaan. Terminal block juga dipasang untuk memudahkan pembagian jalur tegangan dan *ground* pada rangkaian. Setelah seluruh komponen tersolder, dilakukan pemeriksaan visual pada setiap sambungan.

Pemeriksaan ini bertujuan untuk memastikan tidak ada solder yang menumpuk, jalur yang terhubung tidak sengaja, atau kabel yang kurang kuat.



Gambar 4- 5 PCB Tampilan PCB dan komponen setelah di solder

C. Pemasangan Power Supply (PSU)

Power supply atau PSU dipasang sebagai sumber utama tegangan DC pada sistem. PSU 12 V 10 A ditempatkan pada box yang sama dengan komponen lainnya, namun posisinya diletakkan pada bagian tutup box atau bagian atas. Penempatan ini dilakukan agar jalur masuk AC PLN dan jalur keluaran DC 12 V lebih mudah diatur. Selain itu, posisi PSU pada bagian atas juga memudahkan proses pemeriksaan sambungan daya ketika alat dirakit atau diuji.

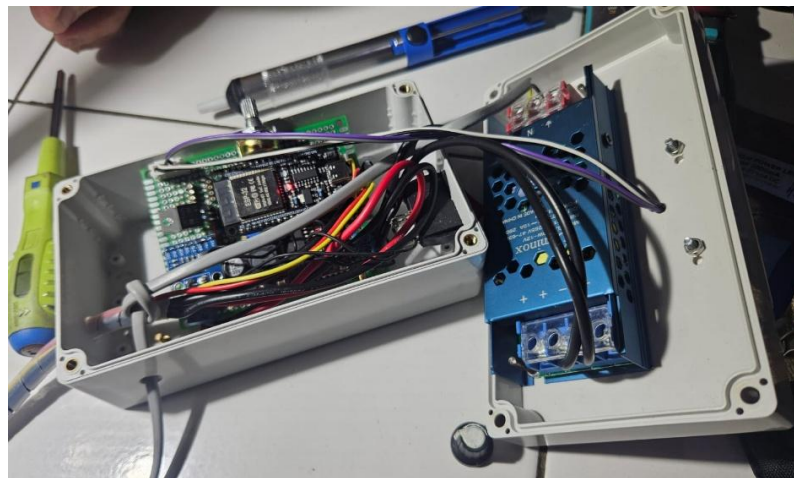
Terminal input PSU dihubungkan dengan sumber AC PLN melalui jalur L dan N. Sementara itu, terminal keluaran 12 V digunakan sebagai sumber tegangan untuk *boost converter*, MP1584 *step-down*, dan kipas pendingin. Jalur positif dan negatif dari PSU diberi pengenalan agar tidak tertukar saat proses penyambungan ke modul lain. Setelah itu, jalur keluaran PSU diarahkan menuju terminal block agar distribusi tegangan lebih tertata. MP1584 *step-down* dihubungkan ke keluaran PSU untuk menurunkan tegangan menjadi 5 V yang digunakan oleh ESP32 dan LCD. *Boost converter* juga memperoleh suplai dari PSU untuk menaikkan tegangan sesuai kebutuhan lampu COB LED 100 W. Dengan pemasangan PSU pada bagian

tutup atas box, jalur catu daya tetap mudah dijangkau, lebih rapi, dan tidak terlalu bercampur dengan jalur sensor maupun rangkaian kendali pada PCB.

D. Pemasangan Komponen pada Box Alat

Setelah PCB dan *power supply* selesai dipasang, tahap berikutnya adalah pemasangan komponen pada box alat. Pemasangan komponen pada box alat dilakukan setelah bagian mekanik dan tata letak komponen selesai ditentukan. Box alat digunakan sebagai tempat utama untuk meletakkan rangkaian kendali, komponen elektronik, kabel, serta bagian tampilan sistem. Pada bagian ini, komponen yang dipasang meliputi PCB rangkaian utama, ESP32, MP1584, INA226, MOSFET IRF540, terminal block, konektor JST, saklar ON/OFF kipas, LCD 16x2 I2C, serta jalur kabel penghubung antar bagian.

PCB yang telah berisi rangkaian utama dipasang di dalam box menggunakan spacer agar posisi PCB lebih stabil dan tidak langsung menempel pada permukaan box. Penggunaan spacer bertujuan untuk mengurangi risiko hubungan singkat pada bagian bawah PCB serta membuat rangkaian lebih mudah diperiksa. Komponen yang berada pada PCB sebelumnya telah disusun dan disolder sesuai konfigurasi pin yang telah dirancang, sehingga pada tahap pemasangan ke box, rangkaian hanya perlu diatur posisinya dan dihubungkan ke komponen lain melalui kabel serta konektor.



Gambar 4- 6 Tampilan PCB dan PSU

Pada bagian panel box, LCD 16x2 I2C dipasang pada bracket LCD yang telah disiapkan. Bagian panel untuk tempat LCD dibuat menyesuaikan ukuran bracket agar tampilan data dapat terlihat dari luar box. Selain LCD, saklar ON/OFF kipas juga dipasang pada lubang switch yang telah dibuat menggunakan proses *laser cutting*. Pembuatan lubang dengan *laser cutting* bertujuan agar ukuran lubang lebih presisi dan sesuai dengan ukuran saklar, sehingga saklar dapat terpasang kuat serta mudah dijangkau saat alat digunakan.

Setelah seluruh komponen pada box terpasang, kabel-kabel dihubungkan sesuai jalur rangkaian. Kabel dari PCB menuju LCD, saklar, catu daya, sensor, dan bagian lampu disusun agar tidak saling bertumpuk. Kabel yang berada di dalam box dirapikan menggunakan konektor JST, terminal block, dan pengikat kabel agar sambungan tidak mudah lepas. Sementara itu, kabel yang berada di bagian luar box dirapikan menggunakan spiral kabel agar tampilan alat lebih rapi, jalur kabel lebih mudah dikenali, dan kabel tidak mudah kusut saat alat dipindahkan atau digunakan.



Gambar 4- 7 Tampilan Bagian Luar Box Alat dengan LCD

E. Pemasangan Komponen pada Dudukan 3D Print

Pemasangan komponen pada dudukan 3D print dilakukan untuk menempatkan bagian lampu dan komponen pendukung yang posisinya terpisah dari box rangkaian

utama. Dudukan 3D print digunakan sebagai rangka atau penyangga bagian atas alat, sehingga komponen seperti lampu COB LED, *heatsink*, kipas pendingin, dan *boost converter* dapat dipasang lebih stabil. Dudukan ini dibuat menyesuaikan bentuk alat agar posisi setiap komponen sesuai dengan rancangan dan tidak mudah bergeser saat alat digunakan.

Pada bagian atas dudukan 3D print, lampu COB LED dipasang bersama *heatsink*. *Heatsink* digunakan untuk membantu menyerap dan membuang panas yang dihasilkan oleh lampu saat bekerja. Di dekat area *heatsink* dipasang kipas pendingin agar aliran udara dapat membantu proses pembuangan panas dengan lebih baik. Penempatan kipas pada bagian atas dudukan bertujuan agar udara dapat langsung diarahkan ke area lampu dan *heatsink*, sehingga suhu kerja lampu dapat lebih terkendali selama pengoperasian.



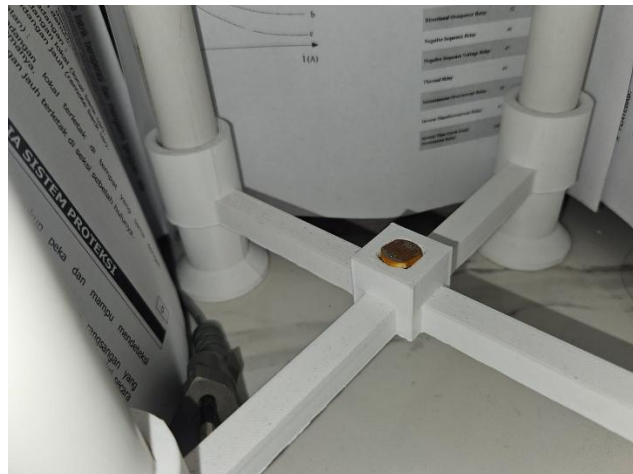
Gambar 4- 8 Tampilan kipas, sensor DS18B20 dan *heatsink* setelah pemasangan

Boost converter juga ditempatkan pada bagian atas dudukan 3D print. Penempatan *boost converter* pada area ini dilakukan agar modul lebih mudah dijangkau ketika dilakukan penyetelan tegangan keluaran. Selain itu, *boost converter* diposisikan terpisah dari rangkaian utama di dalam box karena modul ini berhubungan langsung dengan jalur daya lampu dan dapat menghasilkan panas saat bekerja. Dengan penempatan tersebut, jalur dari *boost converter* menuju lampu COB LED menjadi lebih mudah disusun dan diperiksa.



Gambar 4- 9 Tampilan boost converter setelah pemasangan

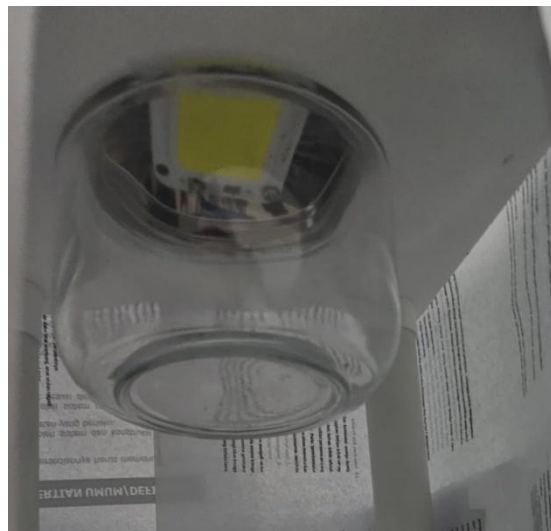
Pada bagian bawah dudukan 3D print, sensor LDR dipasang untuk membaca perubahan intensitas cahaya dari lampu COB LED. LDR ditempatkan pada posisi bawah agar sensor dapat menerima cahaya dari lampu sesuai arah pengujian. Penempatan LDR pada dudukan bawah juga membantu menjaga posisi sensor tetap stabil, sehingga pembacaan intensitas cahaya tidak mudah berubah akibat pergeseran posisi sensor.



Gambar 4- 10 Tampilan LDR setelah pemasangan

Setelah seluruh komponen terpasang pada dudukan 3D print, kabel dari bagian dudukan dihubungkan ke box utama. Kabel dari *boost converter*, kipas, lampu COB LED, dan LDR diarahkan menuju rangkaian utama sesuai jalur yang telah ditentukan. Setelah itu, toples penutup LED dipasang pada bagian bawah dudukan

yang sejajar dengan posisi lampu COB LED. Toples ini berfungsi sebagai pelindung bagian lampu agar tidak terbuka secara langsung, sekaligus tetap memungkinkan cahaya keluar karena bahan yang digunakan bersifat bening atau transparan.



Gambar 4- 11 PCB Tampilan toples dan lampu COB setelah terpasang

Bagian kabel luar dirapikan menggunakan spiral kabel agar sambungan antara box utama danudukan 3D print terlihat lebih rapi serta tidak mengganggu posisi alat. Dengan pemisahan antara box utama dan dudukan 3D print, alat menjadi lebih tertata karena rangkaian kendali berada di dalam box, sedangkan bagian lampu, pendingin, *boost converter*, LDR, dan toples penutup LED berada pada dudukan tersendiri.

4.2 Pembuatan Software

Pembuatan *software* dilakukan setelah proses pembuatan *hardware* selesai dilakukan. *Software* pada sistem *floodlight* COB LED 100 W dibuat untuk mengatur kerja ESP32 sebagai pusat kendali utama. Program yang dibuat berfungsi untuk membaca input potensiometer, membaca sensor LDR, DS18B20, dan INA226, mengatur sinyal PWM ke MOSFET, menampilkan data pada LCD 16x2 I2C, melakukan pengecekan status Wi-Fi, serta mengirimkan data hasil pengukuran ke Google Spreadsheet ketika koneksi Wi-Fi tersedia. Apabila Wi-Fi tidak

terhubung, alat tetap bekerja untuk mengatur kecerahan lampu dan menampilkan data pada LCD, sedangkan proses pengiriman data dilewati sementara.

Pembuatan program dilakukan menggunakan Visual Studio Code dengan ekstensi PlatformIO. PlatformIO digunakan karena dapat membantu proses penulisan program, pengaturan library, kompilasi, upload program ke ESP32, serta pemantauan data melalui Serial Monitor. Dengan menggunakan PlatformIO, proses pembuatan *software* dapat dilakukan secara lebih terstruktur karena file program, library, dan konfigurasi board berada dalam satu project yang sama.

4.2.1 Alat dan Bahan Pembuatan Software

Alat dan bahan *software* merupakan perangkat pendukung yang digunakan untuk menulis, mengatur, mengunggah, dan memantau program pada ESP32. Bagian ini tidak berhubungan langsung dengan perakitan fisik alat, tetapi berperan penting dalam proses pemrograman dan pengiriman data. Alat *software* digunakan untuk menjalankan proses coding, sedangkan bahan *software* digunakan sebagai file, library, dan akun pendukung agar program dapat bekerja sesuai rancangan.

Tabel 4- 3 Daftar alat untuk pembuatan *software*

No.	Alat Software	Fungsi
1	Laptop atau komputer	Menulis, menyimpan, dan mengunggah program ke ESP32 melalui Visual Studio Code dan PlatformIO.
2	Kabel USB	Menghubungkan ESP32 dengan laptop saat proses upload program dan pemantauan serial.
3	Visual Studio Code	Digunakan sebagai editor utama untuk menulis dan mengatur file program.
4	PlatformIO	Digunakan sebagai ekstensi pada Visual Studio Code untuk membuat project ESP32, mengatur board, library, kompilasi, upload, dan Serial Monitor.
5	Browser dan akun Google	Membuka Google Spreadsheet dan mengatur media pencatatan data hasil pengukuran.

Tabel 4- 4 Bahan dan file pendukung software

No.	Bahan/File Software	Fungsi
1	File program main.cpp	Berisi program utama ESP32 yang menjalankan pembacaan sensor, PWM, LCD, status Wi-Fi, dan pengiriman data ke Google Spreadsheet ketika Wi-Fi tersedia.
2	File platformio.ini	Berisi konfigurasi board ESP32, framework Arduino, serta library yang digunakan dalam project PlatformIO.
3	Library OneWire dan DallasTemperature	Digunakan untuk membaca data suhu dari sensor DS18B20.
4	Library LiquidCrystal_I2C	Digunakan untuk menampilkan data pada LCD 16x2 I2C.
5	Library INA226	Digunakan untuk membaca tegangan dan arus pada jalur lampu.
6	Library ESP_Google_Sheet_Client	Digunakan untuk menghubungkan ESP32 dengan Google Spreadsheet.
7	Google Spreadsheet	Digunakan sebagai media penyimpanan data hasil pengukuran secara online ketika koneksi Wi-Fi tersedia.
8	Kredensial layanan Google	Digunakan sebagai akses autentikasi ESP32 ke Google Spreadsheet.

4.2.2 Pembuatan Project pada Visual Studio Code

Tahap awal pembuatan *software* dilakukan dengan membuka Visual Studio Code dan membuat project baru menggunakan PlatformIO. Pada saat pembuatan project, board yang digunakan disesuaikan dengan modul ESP32 yang dipakai pada alat. Framework yang digunakan adalah Arduino karena program disusun menggunakan struktur `setup()` dan `loop()`. Setelah project dibuat, PlatformIO akan membentuk beberapa folder utama, salah satunya folder `src` yang digunakan untuk menyimpan file `main.cpp`.

File `main.cpp` digunakan sebagai tempat penulisan program utama. Pada file ini dituliskan seluruh bagian program, mulai dari pemanggilan library, pendefinisian pin, konfigurasi sensor, fungsi Wi-Fi, fungsi pengiriman data, hingga proses

pembacaan sensor di dalam loop. Dengan struktur tersebut, program dapat lebih mudah diperiksa karena setiap bagian fungsi disusun secara berurutan.

A. Penambahan Library Program

Setelah project dibuat, tahap berikutnya adalah menambahkan library yang dibutuhkan oleh program. Library digunakan agar ESP32 dapat berkomunikasi dengan sensor dan modul yang digunakan pada sistem. Pada program ini, library OneWire dan DallasTemperature digunakan untuk sensor DS18B20, LiquidCrystal_I2C digunakan untuk LCD, INA226 digunakan untuk pembacaan tegangan dan arus, Wi-Fi digunakan untuk koneksi internet, ESP_Google_Sheet_Client digunakan untuk pengiriman data ke Google Spreadsheet, dan time.h digunakan untuk memperoleh waktu berdasarkan NTP.

```
#include <Arduino.h>
#include <OneWire.h>
#include <WiFi.h>
#include <DallasTemperature.h>
#include <LiquidCrystal_I2C.h>
#include "INA226.h"
#include <ESP_Google_Sheet_Client.h>
#include <time.h>
```

B. Konfigurasi Pin dan Parameter Program

Konfigurasi pin dilakukan untuk menentukan hubungan antara program dengan rangkaian *hardware*. Pada sistem ini, sensor suhu DS18B20 menggunakan GPIO33, sensor LDR menggunakan GPIO32, potensiometer menggunakan GPIO34, dan MOSFET dikendalikan melalui GPIO23 sebagai keluaran PWM. Komunikasi I2C pada LCD dan INA226 menggunakan pin I2C ESP32, yaitu SDA pada GPIO21 dan SCL pada GPIO22. Program juga mengatur parameter PWM dengan frekuensi 5000 Hz, channel 0, dan resolusi 8 bit sehingga nilai PWM berada pada rentang 0 sampai 255.

Tabel 4- 5 Konfigurasi pin dan parameter program

No.	Bagian Program	Konfigurasi	Keterangan
1	DS18B20	GPIO33	Membaca suhu pada area lampu/ <i>heatsink</i> .
2	LDR	GPIO32	Membaca perubahan intensitas cahaya.
3	Potensiometer	GPIO34	Input analog untuk menentukan nilai PWM.
4	MOSFET IRF540	GPIO23	Output PWM untuk mengatur kecerahan lampu.
5	LCD I2C	0x27, 16x2	Menampilkan suhu, LDR, PWM, tegangan, dan arus.
6	INA226	0x44	Membaca tegangan dan arus jalur lampu.
7	PWM	5000 Hz, 8 bit	Mengatur duty cycle 0-255.

```
#define ONE_WIRE_BUS 33

int ldr = 32;
const int potPin      = 34;
const int ledPin      = 23;
const int pwmChannel  = 0;
const int pwmFreq     = 5000;
const int pwmResolution = 8;    // 8-bit (0-255)

LiquidCrystal_I2C lcd(0x27, 16, 2);
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

INA226 INA(0x44);
float InaCalibrate = 1.3;
```

C. Konfigurasi Wi-Fi, Waktu, dan Google Spreadsheet

Program dilengkapi koneksi Wi-Fi karena data hasil pengukuran dikirimkan secara online ke Google Spreadsheet.. Namun, koneksi Wi-Fi tidak dijadikan syarat utama agar sistem dapat bekerja. Apabila Wi-Fi tidak terhubung, sistem tetap menjalankan pembacaan sensor, pengaturan PWM, dan tampilan LCD, sementara proses reconnect Wi-Fi dilakukan secara berkala di *background*. Selain itu,

program juga menggunakan NTP dengan zona waktu WIB agar data yang dikirim ke Google Spreadsheet memiliki informasi waktu yang sesuai dengan waktu pengujian ketika koneksi internet tersedia.

Konfigurasi Google Spreadsheet dilakukan dengan menambahkan identitas project, email client, private key, dan spreadsheet ID. Bagian tersebut digunakan oleh library `ESP_Google_Sheet_Client` untuk melakukan autentikasi dan mengirimkan data ke spreadsheet.

```
const char* ssid      = "NAMA_WIFI";
const char* password  = "PASSWORD_WIFI";
int terhubung = LOW;

unsigned long wifiRetryTimer = 0;
const unsigned long wifiRetryInterval = 10000;

const char* ntpServer = "pool.ntp.org";
const long  gmtOffset_sec = 7 * 3600;    // WIB = UTC+7
const int  daylightOffset_sec = 0;
bool timeReady = false;

#define PROJECT_ID "ID_PROJECT_GOOGLE"
#define CLIENT_EMAIL "EMAIL_SERVICE_ACCOUNT"
const char PRIVATE_KEY[] PROGMEM = "PRIVATE_KEY_DISAMARKAN";
const char spreadsheetId[] = "ID_SPREADSHEET_DISAMARKAN";
```

D. Pembuatan Fungsi Pembacaan Potensiometer

Potensiometer digunakan sebagai input analog untuk mengatur tingkat kecerahan lampu. Nilai potensiometer dibaca oleh ESP32 melalui GPIO34. Agar pembacaan lebih stabil, program melakukan pembacaan beberapa kali, kemudian mengambil nilai rata-rata. Pada program ini, jumlah sampel yang digunakan adalah 10 kali pembacaan dengan jeda singkat 2 ms. Hasil rata-rata tersebut kemudian digunakan untuk menentukan nilai PWM.

```
const int numSamples = 10;

int readAveraged() {
    long sum = 0;
    for (int i = 0; i < numSamples; i++) {
        sum += analogRead(potPin);
    }
}
```

```

    delay(2);
  }
  return sum / numSamples;
}

```

E. Pembuatan Fungsi Koneksi Wi-Fi dan Sinkronisasi Waktu

Fungsi koneksi Wi-Fi dibuat untuk memeriksa status jaringan tanpa menghentikan kerja utama alat. Program menampilkan status Wi-Fi pada LCD menggunakan simbol O ketika terhubung dan X ketika tidak terhubung. Jika koneksi belum tersedia, ESP32 tetap melanjutkan pembacaan sensor, pengendalian PWM, dan tampilan LCD, sedangkan percobaan reconnect Wi-Fi dilakukan secara berkala di *background*. Dengan demikian, sistem tetap dapat bekerja meskipun Wi-Fi belum tersambung, dan pengiriman data hanya dilakukan ketika koneksi Wi-Fi tersedia.

```

bool wifi() {
  bool connected = (WiFi.status() == WL_CONNECTED);

  lcd.setCursor(15, 0);
  lcd.print(connected ? "O" : "X");
  digitalWrite(2, connected ? HIGH : LOW);

  if (connected) {
    if (!terhubung) {
      Serial.println("WiFi Connected!");
      terhubung = HIGH;
    }
    return true;
  }

  if (terhubung) {
    Serial.println("WiFi disconnected.");
  }

  terhubung = LOW;
  timeReady = false;

  if (millis() - wifiRetryTimer >= wifiRetryInterval) {
    wifiRetryTimer = millis();
    Serial.println("Mencoba konek WiFi di background...");
    WiFi.disconnect(false);
    WiFi.begin(ssid, password);
  }
}

```

```

    return false;
}

```

F. Integrasi Google Spreadsheet sebagai Media Penyimpanan Data

Google Spreadsheet digunakan sebagai media penyimpanan data hasil pengukuran secara online. Agar ESP32 dapat mengirim data ke *spreadsheet*, perlu dibuat alamat *spreadsheet*, project *Google Cloud*, serta akses *service account* yang berisi project ID, client email, dan *private key*. Tahapan pembuatannya dapat dilakukan sebagai berikut.

1. Membuka *Google Drive*, kemudian memilih menu New → *Google Sheets* untuk membuat file *spreadsheet* baru.
2. Memberi nama file lalu membuat kolom pada baris pertama spreadsheet sesuai data yang akan dikirim dari ESP32, yaitu Time, Suhu, LUX, PWM, Tegangan, dan Arus.
3. Mengambil *Spreadsheet ID* dari URL Google Spreadsheet. *Spreadsheet ID* merupakan kode panjang yang berada di antara bagian /d/ dan /edit pada alamat *spreadsheet*.
4. Membuka *Google Cloud Console*, kemudian membuat project baru sebagai tempat pengaturan akses Google Spreadsheet.
5. Mengaktifkan *Google Sheets API* melalui menu *APIs & Services* → *Library*, kemudian mencari *Google Sheets API* dan memilih *Enable*.
6. Membuat *service account* melalui menu *IAM & Admin* → *Service Accounts*. *Service account* ini digunakan sebagai akun akses agar ESP32 dapat mengirim data ke Google Spreadsheet.
7. Membuat *private key* pada *service account* dengan memilih menu *Keys* → *Add Key* → *Create New Key*, kemudian memilih format JSON.
8. Mengunduh file JSON yang berisi data akses, seperti project ID, client email, dan *private key*. Data tersebut digunakan untuk menghubungkan program ESP32 dengan Google Spreadsheet.
9. Membuka kembali Google Spreadsheet, kemudian membagikan file *spreadsheet* ke *client email* yang terdapat pada file JSON dengan akses *Editor*.

10. Memasukkan data akses berupa *spreadsheet* ID, project ID, *client* email, dan private key ke dalam program ESP32.
11. Menjalankan program pada ESP32 untuk menguji pengiriman data. Jika koneksi berhasil, data hasil pengukuran akan masuk secara otomatis ke Google Spreadsheet pada baris baru.

G. Pembuatan Fungsi Pengiriman Data ke Google Spreadsheet

Pengiriman data ke Google Spreadsheet dilakukan menggunakan fungsi `uploadToSheetOnlineOnly()`. Fungsi ini memeriksa status Wi-Fi terlebih dahulu sebelum data dikirim. Apabila Wi-Fi tidak terhubung, fungsi akan langsung membatalkan proses pengiriman data dan mengembalikan nilai `false`, sehingga program utama tetap berjalan tanpa menunggu koneksi. Jika Wi-Fi terhubung, program melakukan sinkronisasi waktu jika diperlukan, memeriksa kesiapan Google Sheet, lalu mengirimkan data ke spreadsheet.

```
bool uploadToSheetOnlineOnly(
    FirebaseJson &valueRange
) {
    if (WiFi.status() != WL_CONNECTED) {
        Serial.println("WiFi tidak tersambung. Data tidak
dikirim.");
        return false;
    }

    if (!timeReady) {
        syncTime(3000);
    }

    if (!GSheet.ready()) {
        Serial.println("Google Sheet belum ready. Upload
dilewati.");
        return false;
    }

    FirebaseJson response;
    bool success = GSheet.values.append(
        &response, spreadsheetId, "Sheet1!A1", &valueRange
    );

    return success;
}
```

H. Implementasi Fungsi Setup

Fungsi `setup()` digunakan untuk menjalankan proses awal saat ESP32 pertama kali dinyalakan. Pada bagian ini, Serial Monitor diaktifkan, sensor DS18B20 dijalankan, pin input dan output diatur, PWM dikonfigurasi, LCD diinisialisasi, Wi-Fi mulai dihubungkan tanpa menahan proses kerja alat, INA226 diperiksa, dan koneksi Google Spreadsheet disiapkan. Pada program terbaru, ESP32 menggunakan `WiFi.begin()` untuk mulai mencoba koneksi Wi-Fi, sehingga alat tetap dapat masuk ke proses utama walaupun Wi-Fi belum tersambung. Header Google Spreadsheet tidak dikirimkan secara otomatis pada setup agar alat tidak tertahan saat koneksi internet belum tersedia.

```
void setup() {
  Serial.begin(115200);
  sensors.begin();

  pinMode(ldr, INPUT);
  pinMode(potPin, INPUT);
  pinMode(2, OUTPUT);

  analogReadResolution(12);
  analogSetAttenuation(ADC_11db);
  ledcSetup(pwmChannel, pwmFreq, pwmResolution);
  ledcAttachPin(ledPin, pwmChannel);
  ledcWrite(pwmChannel, 0);

  lcd.init();
  lcd.backlight();

  WiFi.mode(WIFI_STA);
  WiFi.persistent(false);
  WiFi.setSleep(false);
  WiFi.setAutoReconnect(true);
  WiFi.begin(ssid, password);
  lcd.setCursor(15, 0);
  lcd.print("X");

  if (!INA.begin()) {
    lcd.setCursor(0, 0);
    lcd.print("INA Error");
  }

  INA.setMaxCurrentShunt(1, 0.002);
}
```

```
GSheet.begin(CLIENT_EMAIL, PROJECT_ID, PRIVATE_KEY);
}
```

I. Implementasi Fungsi Loop

Fungsi loop() merupakan bagian program yang berjalan secara berulang selama alat menyala. Pada bagian ini, ESP32 memeriksa status Wi-Fi, membaca suhu dari DS18B20, membaca nilai potensiometer, membaca LDR, membaca tegangan dan arus dari INA226, lalu menghitung nilai PWM untuk mengatur kecerahan lampu. Nilai potensiometer dipetakan dari rentang ADC 0 sampai 4095 menjadi nilai PWM 0 sampai 255. Meskipun Wi-Fi tidak terhubung, proses pembacaan sensor, pengendalian PWM, dan tampilan LCD tetap berjalan. Pengiriman data ke Google Spreadsheet hanya dilakukan ketika Wi-Fi tersedia.

```
void loop() {
  wifi();
  sensors.requestTemperatures();

  int potValue = readAveraged();
  int dutyCycle = map(potValue, 0, 4095, 0, 255);
  int cnd = analogRead(ldr);

  float suhu = sensors.getTempCByIndex(0);
  float busvoltage = INA.getBusVoltage() - InaCalibrate;
  float current_mA = INA.getCurrent_mA();

  ledcWrite(pwmChannel, dutyCycle);

  lcd.setCursor(0, 0);
  lcd.print("PWM : ");
  lcd.print(dutyCycle);
  lcd.print(" ");
  lcd.setCursor(15, 0);
  lcd.print(WiFi.status() == WL_CONNECTED ? "O" : "X");

  // Data tetap ditampilkan pada LCD.
  // Data dikirim ke Google Spreadsheet hanya saat Wi-Fi
  tersedia.
}
```

J. Pembuatan Tampilan LCD

Pada sistem ini, LCD 16x2 I2C digunakan sebagai media tampilan data secara langsung pada alat. Baris pertama LCD digunakan untuk menampilkan nilai PWM secara tetap karena PWM merupakan parameter utama dalam pengaturan kecerahan lampu COB LED. Pada bagian kanan baris pertama juga ditampilkan simbol status Wi-Fi, yaitu O ketika Wi-Fi terhubung dan X ketika Wi-Fi tidak terhubung. Baris kedua LCD digunakan untuk menampilkan data lain secara bergantian, yaitu suhu dari sensor DS18B20, nilai intensitas cahaya dari LDR, tegangan, dan arus dari INA226. Pergantian tampilan dilakukan setiap 3 detik menggunakan fungsi millis() agar tampilan dapat berubah secara otomatis tanpa menghentikan proses pembacaan sensor maupun proses reconnect Wi-Fi.

```

lcd.setCursor(0, 0);
lcd.print("PWM : ");
lcd.print(dutyCycle);
lcd.print(" ");
lcd.setCursor(15, 0);
lcd.print(WiFi.status() == WL_CONNECTED ? "O" : "X");

if ((millis() - timer) >= time_1) {
    timer = millis();
    page++;
    if (page > 3) page = 0;
}

switch (page) {
    case 0:
        lcd.setCursor(0, 1);
        lcd.print("Lux : ");
        lcd.print(cnd);
        lcd.print(" ");
        break;

    case 1:
        lcd.setCursor(0, 1);
        lcd.print("Suhu : ");
        lcd.print(suhuu);
        lcd.print(" C ");
        break;

    case 2:
        lcd.setCursor(0, 1);
        lcd.print("Volt : ");

```

```

        lcd.print(busvoltage);
        lcd.print(" V");
        lcd.print("      ");
        break;

    case 3:
        lcd.setCursor(0, 1);
        lcd.print("Arus : ");
        lcd.print(current_mA / 1000);
        lcd.print(" A");
        lcd.print("      ");
        break;
}

```

K. Pengiriman Data Berkala ke Google Spreadsheet

Pengiriman data ke Google Spreadsheet dilakukan secara berkala setiap 10 detik. Pengaturan waktu pengiriman dilakukan menggunakan variabel `duration` dengan nilai 10000 ms. Ketika waktu pengiriman terpenuhi, program mengambil waktu terbaru, membuat data dalam bentuk `FirestoreJson`, kemudian memeriksa status Wi-Fi sebelum mengirimkan data ke Google Spreadsheet. Data yang dikirim terdiri dari waktu, suhu, nilai LDR, nilai PWM, tegangan, dan arus. Apabila Wi-Fi tidak terhubung, pengiriman data dilewati dan alat tetap melanjutkan proses pembacaan sensor serta pengendalian lampu.

```

if ((millis() - stimer) > duration) {
    stimer = millis();
    String waktu = getTimeNow();

    FirestoreJson valueRange;
    valueRange.add("majorDimension", "ROWS");

    valueRange.set("values/[0]/[0]", waktu);
    valueRange.set("values/[0]/[1]", suhu);
    valueRange.set("values/[0]/[2]", cnd);
    valueRange.set("values/[0]/[3]", dutyCycle);
    valueRange.set("values/[0]/[4]", busvoltage / 10);
    valueRange.set("values/[0]/[5]", current_mA / 1000);

    bool success = uploadToSheetOnlineOnly(valueRange);

    if (success) {
        Serial.println("Upload selesai.");
    }
}

```

```

    } else {
        Serial.println("Offline / upload gagal, alat tetap
berjalan.");
    }
}

```

L. Upload Program dan Pemeriksaan Serial Monitor

Setelah program selesai ditulis, tahap berikutnya adalah melakukan kompilasi untuk memastikan tidak terdapat kesalahan penulisan kode. Jika proses kompilasi berhasil, program diunggah ke ESP32 melalui kabel USB menggunakan fitur upload pada PlatformIO. Setelah upload selesai, Serial Monitor dibuka untuk melihat respon program saat ESP32 bekerja.

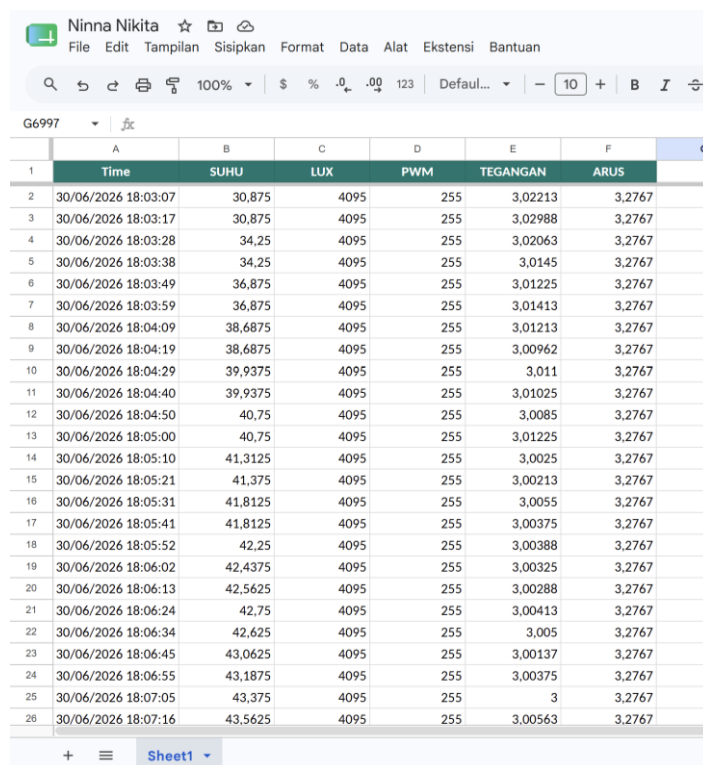
Pemeriksaan melalui Serial Monitor dilakukan untuk memastikan bahwa ESP32 dapat membaca nilai potensiometer, menghasilkan nilai PWM, membaca LDR, membaca suhu DS18B20, membaca tegangan dan arus dari INA226, menampilkan status Wi-Fi, serta mengirimkan data ke Google Spreadsheet ketika koneksi tersedia. Pada Serial Monitor, program menampilkan informasi nilai potensiometer, duty cycle, brightness, waktu, cahaya, suhu, tegangan, arus, dan status upload data. Jika muncul pesan upload selesai, berarti data berhasil dikirimkan ke Google Spreadsheet. Apabila Wi-Fi tidak terhubung, Serial Monitor menampilkan informasi bahwa data tidak dikirim, tetapi alat tetap berjalan.

4.2.3 Hasil Pembuatan Software

Berdasarkan pembuatan *software* yang telah dilakukan, program pada ESP32 sudah mencakup fungsi utama sistem *floodlight* COB LED 100 W. Program dapat membaca input potensiometer sebagai pengatur PWM, membaca sensor LDR sebagai parameter intensitas cahaya, membaca sensor DS18B20 sebagai parameter suhu, dan membaca INA226 sebagai parameter tegangan serta arus. Data hasil pembacaan tersebut kemudian ditampilkan pada LCD dan dikirimkan ke Google Spreadsheet ketika koneksi Wi-Fi tersedia.

Software juga dilengkapi dengan pemeriksaan koneksi Wi-Fi, sinkronisasi waktu, indikator status Wi-Fi pada LCD, serta sistem pengiriman data yang tidak menghentikan kerja alat ketika jaringan belum tersedia. Dengan adanya sistem

tersebut, ESP32 dapat mencoba menghubungkan kembali Wi-Fi secara berkala di *background*. Secara keseluruhan, program yang dibuat telah mendukung kerja *hardware* karena mampu mengatur kecerahan lampu, memantau parameter kerja, menampilkan data secara langsung, tetap beroperasi tanpa koneksi Wi-Fi, dan mencatat data hasil pengukuran secara online ketika Wi-Fi tersedia.



	A	B	C	D	E	F
	Time	SUHU	LUX	PWM	TEGANGAN	ARUS
2	30/06/2026 18:03:07	30,875	4095	255	3,02213	3,2767
3	30/06/2026 18:03:17	30,875	4095	255	3,02988	3,2767
4	30/06/2026 18:03:28	34,25	4095	255	3,02063	3,2767
5	30/06/2026 18:03:38	34,25	4095	255	3,0145	3,2767
6	30/06/2026 18:03:49	36,875	4095	255	3,01225	3,2767
7	30/06/2026 18:03:59	36,875	4095	255	3,01413	3,2767
8	30/06/2026 18:04:09	38,6875	4095	255	3,01213	3,2767
9	30/06/2026 18:04:19	38,6875	4095	255	3,00962	3,2767
10	30/06/2026 18:04:29	39,9375	4095	255	3,011	3,2767
11	30/06/2026 18:04:40	39,9375	4095	255	3,01025	3,2767
12	30/06/2026 18:04:50	40,75	4095	255	3,0085	3,2767
13	30/06/2026 18:05:00	40,75	4095	255	3,01225	3,2767
14	30/06/2026 18:05:10	41,3125	4095	255	3,0025	3,2767
15	30/06/2026 18:05:21	41,375	4095	255	3,00213	3,2767
16	30/06/2026 18:05:31	41,8125	4095	255	3,0055	3,2767
17	30/06/2026 18:05:41	41,8125	4095	255	3,00375	3,2767
18	30/06/2026 18:05:52	42,25	4095	255	3,00388	3,2767
19	30/06/2026 18:06:02	42,4375	4095	255	3,00325	3,2767
20	30/06/2026 18:06:13	42,5625	4095	255	3,00288	3,2767
21	30/06/2026 18:06:24	42,75	4095	255	3,00413	3,2767
22	30/06/2026 18:06:34	42,625	4095	255	3,005	3,2767
23	30/06/2026 18:06:45	43,0625	4095	255	3,00137	3,2767
24	30/06/2026 18:06:55	43,1875	4095	255	3,00375	3,2767
25	30/06/2026 18:07:05	43,375	4095	255	3	3,2767
26	30/06/2026 18:07:16	43,5625	4095	255	3,00563	3,2767

Gambar 4- 12 Tampilan data pada Google Spreadsheet

4.3 Hasil Perakitan Sistem Floodlight COB LED 100 W

Setelah proses pembuatan *hardware* dan *software* selesai dilakukan, sistem *floodlight* COB LED 100 W telah menjadi satu alat yang utuh dan siap untuk dilakukan pengujian. Hasil pembuatan alat ini terdiri dari bagian *hardware* berupa rangkaian kendali, rangkaian daya, sensor, lampu, kipas, LCD, serta bagian mekanik berupa box alat danudukan 3D *print*. Selain itu, bagian *software* juga telah dibuat dan diunggah ke ESP32 untuk mengatur kerja sistem secara keseluruhan.

Pada bagian *hardware*, seluruh komponen telah dipasang sesuai rancangan. Rangkaian utama ditempatkan pada box alat, sedangkan bagian lampu COB LED, *heatsink*, kipas, sensor LDR, dan *boost converter* dipasang padaudukan 3D *print*. LCD 16x2 I2C, potensiometer, dan saklar dipasang pada bagian panel box agar mudah dilihat dan dioperasikan oleh pengguna. Kabel-kabel penghubung dirapikan menggunakan spiral kabel dan konektor agar susunan alat terlihat lebih rapi serta mudah diperiksa saat pengujian.

Pada bagian *software*, program telah dibuat menggunakan Visual Studio Code dengan ekstensi PlatformIO. Program tersebut berfungsi untuk membaca nilai potensiometer sebagai pengatur PWM, membaca sensor suhu DS18B20, membaca sensor LDR, membaca tegangan dan arus melalui INA226, menampilkan data pada LCD, serta mengirimkan data hasil pengukuran ke Google Spreadsheet melalui koneksi Wi-Fi. Dengan demikian, sistem tidak hanya dapat mengatur kecerahan lampu, tetapi juga dapat melakukan pemantauan dan pencatatan data secara otomatis.



Gambar 4- 13 Tampilan akhir alat setelah perakitan