

BAB IV

PEMBUATAN ALAT

4.1 Pembuatan Alat

Pembuatan alat "RANCANG BANGUN SISTEM LAMPU UV MAJEMUK DENGAN METODE SATU TABUNG DAN SISTEM PEMBERSIH SLEVE OTOMATIS PADA ALAT STERILISASI AIR" dirancang dengan dua bagian utama, yaitu,

1. Pembuatan Perangkat Keras (*Hardware*)

Tahapan pembuatan perangkat keras mencakup komponen mekanik dan instalasi komponen elektrik, yang disesuaikan dengan rancangan sistem yang telah ditetapkan sebelumnya.

2. Pembuatan Perangkat Lunak (*Software*)

Tahapan pembuatan perangkat lunak mencakup pembuatan sistem pembersihan, menu konfigurasi dan datalogging pada alat, yang disesuaikan dengan kebutuhan informasi pada alat.

Dalam pembuatan alat perangkat keras dan perangkat lunak, dibutuhkan berbagai alat dan bahan untuk mendukung proses pengerjaan.

Tabel 4- 1Alat yang digunakan

NO	Alat	Jumlah
1	Tang Kombinasi	1
2	Tang Potong	1
3	Tang Kupas	1
4	Bor Tangan	1
5	Bor Duduk	1
6	Obeng Set	1
7	Gerinda	1
8	Mesin 3D print	1
9	Mesin CNC Laser	1

NO	Alat	Jumlah
10	Solder	1
11	Kunci L 1 set	1

Tabel 4- 2 Bahan yang digunakan

NO	Bahan	Jumlah
1	Filament 3D Print	3
2	Grommet $\frac{3}{4}$ Inch	8
3	Mur M8	20
4	As Ulir M8 (1 meter)	4
5	Akrilik (50x50 cm)	2
6	Pipa Akrilik	4
7	Baut Nylon M3	12
8	<i>Lead Screw</i> T8	1 x 50cm
9	Tnut M8	1
10	Pillow Block Bearing KF08L	1
11	<i>Coupling</i> 8 to 12	1
12	<i>Coupling</i> 8 to 5	1
13	Snap ring 12s	1
14	Snap ring 8s	1
15	Mecahnichal Seal	1
16	Kaki kaki karet	8
17	Mur M4	8
18	Baut M4 (15mm)	8
19	Motor Stepper Nema17	1
20	Box Durados 220 x 150 x 120 mm	1
21	PSU SMPS 24V/5A	1
22	TB6600 <i>Driver</i> Stepper Motor	1
23	LCD 20x4 I2C	1
24	Rotary Encoder EC11	1

NO	Bahan	Jumlah
25	Rocker Switch On-Off	1
26	Kabel NYMHY 0.75x2	2 meter
27	Insert Nut M3	4
28	PCB FR4 20x10cm	1
29	Socket JST XH2.54 – 4pin	2
30	Socket JST XH2.54 – 5pin	1
31	ESP32 DOIT 30pin	1
32	Buck Converter MP1584 – 5V	1
33	Buzzer Aktif 3.3V	1
34	Modul RTC DS3231	1
35	Capacitor Elektrolit 470uF/25V	2
36	Dioda SS34	1
37	Kabel Stepper Nema17	1
38	Socket plugable 5.08 – 2pin	1

4.2 Pembuatan Perangkat Keras (*Hardware*)

Pembuatan perangkat keras membahas mengenai langkah-langkah dan proses pembuatan perangkat keras pada alat yang meliputi 3 bagian pembuatan yaitu, pembuatan box kontroler utama, penutup tabung reaktor dan mekanisme pembersihan Sleeve lampu UV.

4.2.1 Pembuatan PCB Kontroler

Pada pembuatan PCB kontroller dibagi menjadi 7 bagian yaitu, perancangan komponen, pembuatan *schematic* dan layout PCB pada aplikasi EasyEDA, konversi file gerber menjadi CAM file, pembuatan jalur PCB dengan CNC laser, proses *etching* dan pelubangan PCB, penyolderan komponen, dan pengetesan PCB.

1) Perancangan komponen

Tahap perancangan komponen dilakukan dengan menentukan kebutuhan sistem kontrol yang akan digunakan pada alat. Beberapa komponen utama yang digunakan meliputi modul ESP32 DOIT 30P sebagai mikrokontroler

utama, modul step down MP1584 sebagai penurun tegangan, RTC DS3231 sebagai pewaktu sistem, buzzer 3.3V sebagai indikator suara, konektor JST XH 2.54 untuk antarmuka rotary encoder, LCD, dan motor stepper, serta socket Plugable EDG 2P 5.08 sebagai terminal catu daya 24V. Selain itu digunakan dua buah kapasitor elektrolit 470 μ F digunakan sebagai kapasitor filter pada jalur catu daya 3.3V dan 5V untuk membantu menjaga kestabilan tegangan serta meredam ripple pada sistem. Penentuan komponen dilakukan dengan mempertimbangkan kebutuhan sistem, kemudahan perawatan, serta efisiensi penempatan pada PCB.

LCSC Electronics [EasyEDA](#) Search, at least 2 characters [Request New Part](#) [Refresh](#) [Add](#) [Batch Save As](#) [Baki](#)

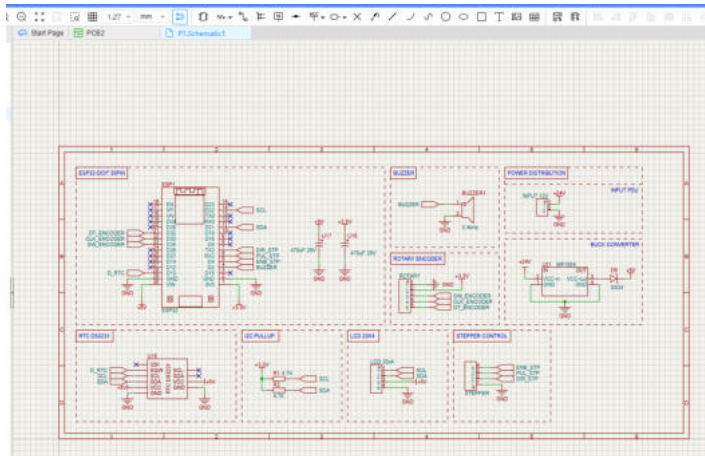
N.	Device	Footprint	Supplier Part	Manufacturer
1	Drawing-Symbol_A4			
2	ESP32	ESP32-30PIN		
3	STEPCDOWN 5V	STEPCDOWN5V		
4	Netport-IN			
5	Ground-GND			
6	Power-5V			
7	Power-VCC			
8	SS34_C53058727	SMA_L4.3-W2.5-LS5.0-RD	C53058727	ku Jing (晶晶)
9	2EDG-5 08-2AWH	CONN-TH_2P-P5 08_2EDG-5 08-2AWH	C22387921	DLL (德利)
10	Res_0603	R0603		
11	Res_0603	R0603		
12	XY-XH2.54-4A11	CONN-TH_4P-P2.50-V-M	C19271370	XYECONN (芯源)
13	SH-XH2.54-5P22	CONN-TH_5P-P2.50_BX-XH2.54-5P22-R	C54431936	SHOU HAN (首韩)
14	470uF 25V	CAP-TH_B08.0-P3.50-D0.6-FD	C22383480	
15	Buzzer_1	BUZ-TH_BD12.5-P5.00-D0.8-RD	C9900210808	
16	TMB12A12	BUZ-TH_BD12.0-P7.60-D0.6-FD	C781861	YUEXIN (悦鑫)
17	Mini560	MINI560	mini560	
18	ESP32	ESP32-30PIN		
19	Mini560	MINI560	mini560	
20	RTC DS3231	RTC DS3231		
21	0805W8F6801T5E	R0805	C17772	UNI-ROYAL (源声)
22	CAP_0402	C0402	C1525	
23	U1PW1E471MPD6TD	CAP-TH_B08.0-P3.50-D0.6-FD	C1588647	Nichicon (尼吉康)
24	YPRH1209-330M	IND-SMD_L12.0-W12.0_CDRH129L125...	C718306	YJYCOIN (益奇源)
25	Short-Symbol			
26	SS34	SMA_L4.4-W2.8-LS5.4-RD	C142458	
27	S3JHE3-TP	DO-214AB_L6.9-W5.9-L57.9-RD	C19991155	MCC (美微利)
28	LM2595S-5.0RG	TO-263-5_L10.2-W8.9-P1.70-TL-LS15.2	C2977092	HGC (深圳汉芯)
29	CAP_0805	C0805		
30	MP1584 step down Module	MP1584 STEP DOWN MODULE - PCB		

Gambar 4. 1 Pemilihan komponen pada aplikasi EasyEda

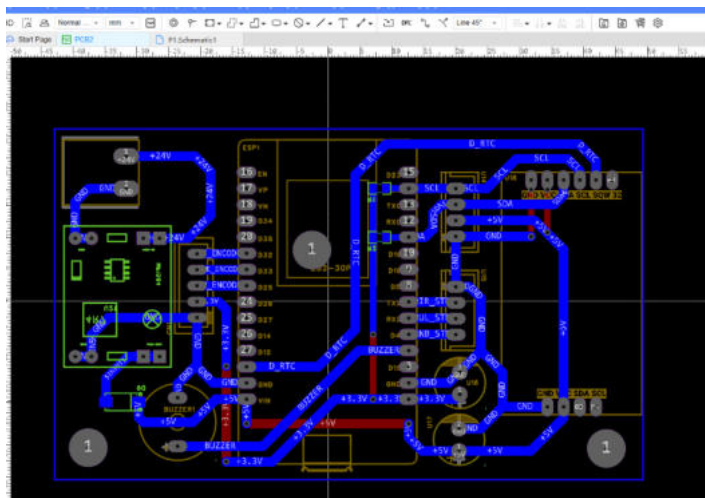
2) Pembuatan *schematic* dan layout PCB pada aplikasi EasyEDA

Setelah komponen ditentukan, tahap berikutnya adalah pembuatan *schematic* dan layout PCB menggunakan aplikasi EasyEDA. Pada tahap *schematic* dilakukan perancangan hubungan antar komponen sesuai dengan kebutuhan sistem kontrol. Selanjutnya dilakukan pembuatan layout PCB dengan memperhatikan ukuran jalur, posisi komponen, serta kemudahan

proses penyolderan dan perakitan. Penempatan konektor dan modul disusun sedemikian rupa agar mempermudah integrasi dengan komponen eksternal seperti LCD, rotary encoder, motor stepper, dan sumber daya.



Gambar 4. 2 Schematic komponen pada sistem

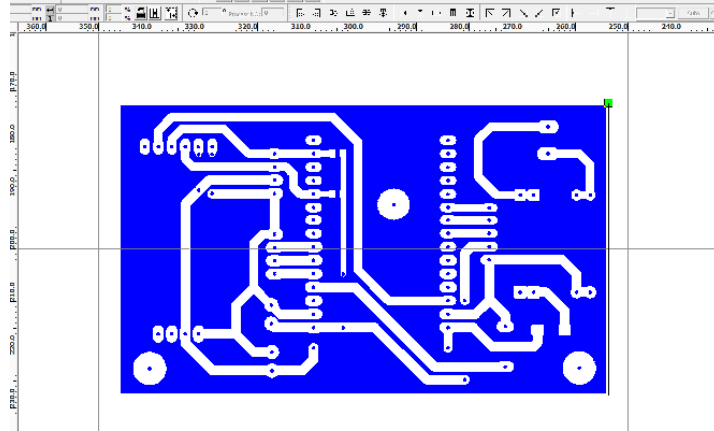


Gambar 4. 3 Routing jalur pada PCB kontrol

3) Konversi file gerber menjadi CAM file

Setelah desain PCB selesai, file Gerber hasil ekspor dari EasyEDA dikonversi menjadi file CAM menggunakan aplikasi FlatCAM. Proses ini bertujuan untuk menghasilkan jalur *toolpath* yang dapat dibaca oleh mesin CNC laser. Pengaturan parameter seperti kecepatan gerak, kedalaman jalur,

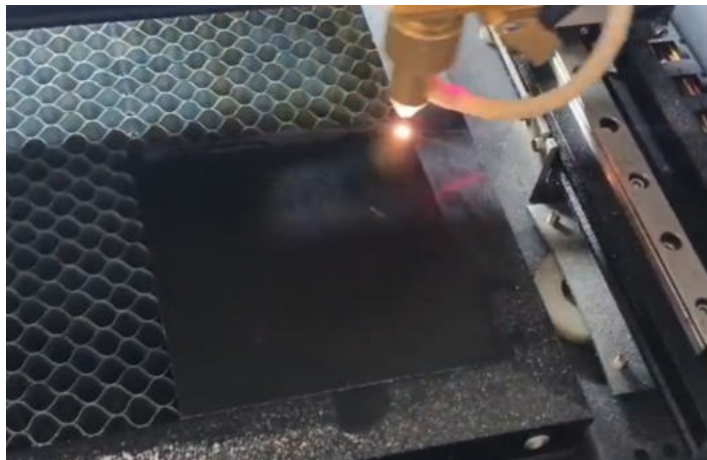
dan area *scanning* dilakukan agar hasil pembentukan jalur PCB sesuai dengan desain yang telah dibuat.



Gambar 4. 4 Pembuatan CAM jalur PCB dengan software RdWorks

4) Pembuatan jalur PCB dengan CNC laser

Tahap pembuatan jalur PCB dilakukan menggunakan mesin CNC laser dengan bantuan aplikasi RDWorks sebagai pengendali proses laser. Sebelum proses *scanning* laser dilakukan, permukaan PCB terlebih dahulu dilapisi menggunakan pilox hitam agar area tertentu dapat terlindungi saat proses *etching*. Selanjutnya mesin laser melakukan *scanning* sesuai pola jalur PCB yang telah dibuat sehingga membentuk pola rangkaian dengan tingkat presisi yang baik.



Gambar 4. 5 Proses pembuatan jalur dengan CNC laser

5) Proses *etching* dan pelubangan PCB

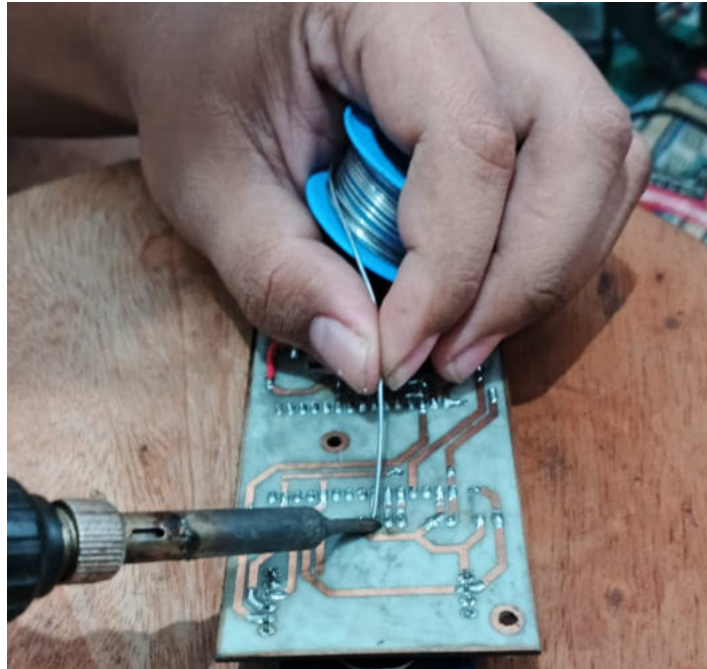
Setelah proses *scanning* laser selesai, PCB masuk ke tahap *etching* untuk menghilangkan lapisan tembaga yang tidak diperlukan sehingga terbentuk jalur rangkaian sesuai desain. Setelah jalur PCB terbentuk, dilakukan proses pelubangan pada titik kaki komponen menggunakan bor duduk (*bench drill*) agar komponen dapat dipasang dengan presisi sesuai ukuran lubang yang dibutuhkan.



Gambar 4. 6 Proses *etching* PCB

6) Penyolderan komponen

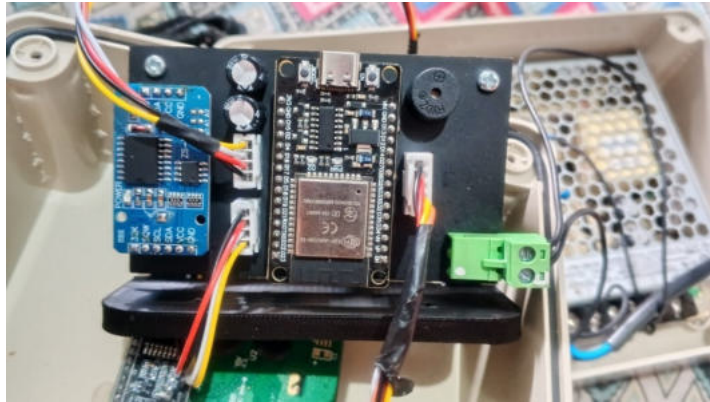
Tahap berikutnya adalah pemasangan dan penyolderan seluruh komponen pada PCB. Proses penyolderan dilakukan secara bertahap mulai dari komponen berukuran kecil hingga modul utama seperti ESP32 dan konektor. Penyolderan dilakukan dengan memperhatikan kualitas sambungan timah agar tidak terjadi *short circuit* maupun sambungan yang kurang sempurna sehingga rangkaian dapat bekerja secara optimal.



Gambar 4. 7 Proses penyolderan komponen pada PCB

7) Pengetesan PCB.

Tahap akhir adalah pengujian PCB yang telah dirakit. Pengujian dilakukan melalui beberapa tahapan seperti pengecekan kontinuitas jalur untuk memastikan tidak terdapat hubungan terputus atau *short circuit*, pengujian tegangan keluaran dari modul MP1584, serta pengujian fungsi antarmuka seperti LCD, rotary encoder, RTC DS3231, buzzer, dan motor stepper. Selain itu dilakukan pengujian komunikasi dan kinerja sistem kontrol berbasis ESP32 untuk memastikan seluruh rangkaian dapat bekerja sesuai perancangan.



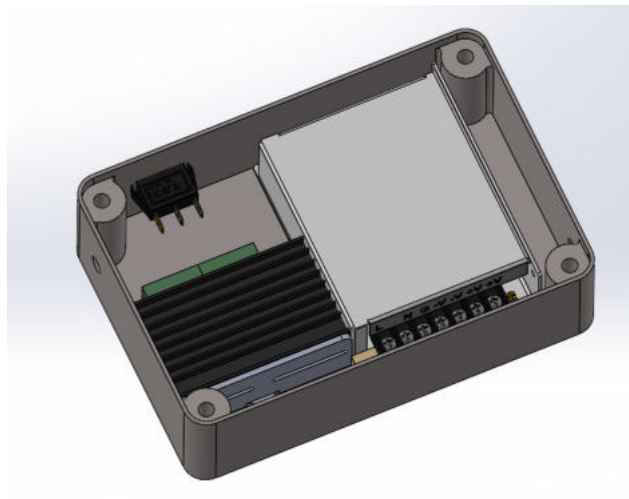
Gambar 4. 8 Proses pengecekan koneksi socket pada PCB

4.2.2 Pembuatan Kontroler Utama

Pada pembuatan kontroler utama dibagi menjadi 4 bagian yaitu, penempatan komponen via software 3D, pembuatan *mounting bracket* komponen, melubangi box kontroler, dan perakitan komponen ke dalam box. Pada tahap pengujian bertujuan untuk memastikan bahwa alat bekerja secara normal tanpa adanya kendala.

1) Penempatan komponen via software 3D

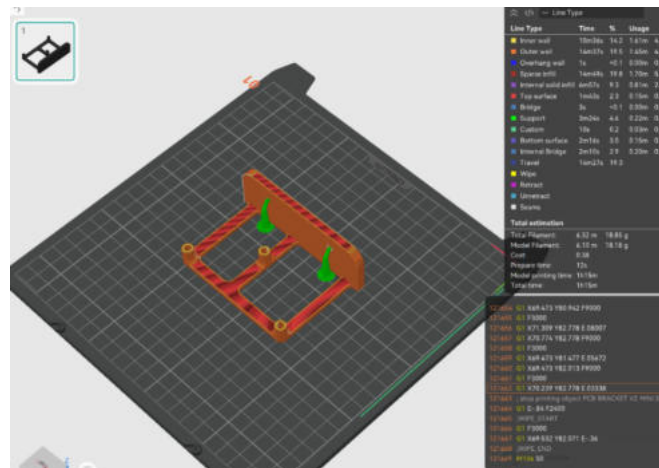
Proses penempatan komponen dengan bantuan software 3D Solidworks bertujuan untuk memperoleh visualisasi dan validasi posisi komponen sehingga mengurangi kesalahan dalam eksekusi alat.



Gambar 4. 9 Visual penempatan komponen dengan software 3D Solidoworks

2) Pembuatan *mounting bracket* komponen

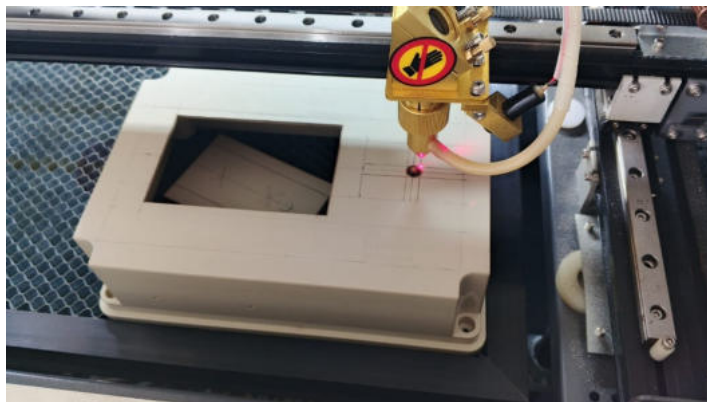
Setelah melakukan visualisasi penempatan komponen maka akan tampak beberapa komponen yang memerlukan *mounting bracket* agar dapat ditata di dalam box. LCD, PCB kontroler dan *driver* motor stepper memerlukan *mounting bracket* agar dapat ditempatkan didalam box, maka mouting dibuat sedemikian rupa,



Gambar 4. 10 Proses slicing desain mounting PCB pada software OrcaSlicer

3) Melubangi box kontroler

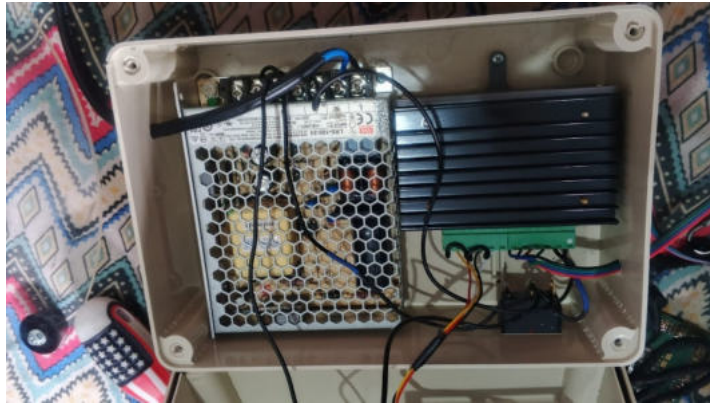
Selanjutnya eksekusi box kontroler dengan melubangi beberapa bagian untuk penempatan mouting dan output kabel power serta stepper. Pada pembuatan lubang pada box dibantu dengan mesin CNC laser guna didapatkan hasil yang presisi.



Gambar 4. 11 Proses melubangi box dengan mesin CNC laser

4) Perakitan komponen

Setelah semua komponen siap mulai dari *mounting bracket*, PCB kontroler dan pelubangan box maka dilakukan perakitan ke dalam box sesuai dengan visualisasi dengan software 3D yang telah dilakukan sebelumnya



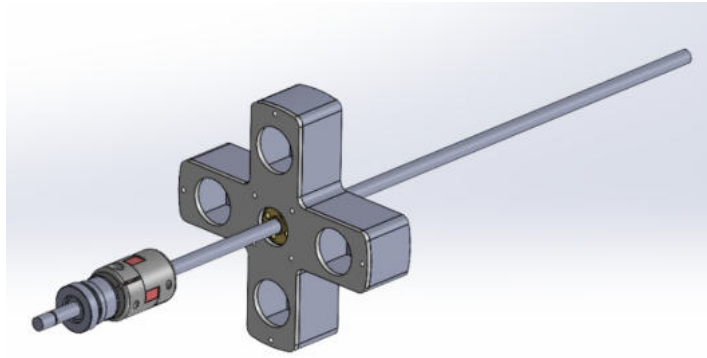
Gambar 4. 12 Proses perakitan komponen pada box bagian bawah

4.2.3 Pembuatan Pembersihan Sleeve Lampu UV

Pada pembuatan pembersihan Sleeve dibagi menjadi 4 bagian yaitu, perancangan 3D model berdasarkan ukuran asli, mencetak wadah spon pembersih, memotong spon dan akrilik pembersih, dan perakitan komponen.

1) Perancangan 3D model berdasarkan ukuran asli

Tahap perancangan dilakukan dengan membuat model 3D sistem pembersih menggunakan SolidWorks berdasarkan ukuran asli tabung, mekanisme *lead screw*, serta komponen pendukung lainnya. Perancangan ini bertujuan untuk memperoleh visualisasi sistem pembersih sehingga mekanisme gerak dapat bekerja dengan baik menyerupai sistem *wiper*. Selain itu, desain 3D digunakan untuk memastikan posisi komponen sesuai dan mengurangi kesalahan pada tahap fabrikasi.



Gambar 4. 13 Visualisasi desain pembersih

2) Mencetak wadah spon pembersih

Setelah desain selesai dibuat, dilakukan proses pencetakan wadah spons pembersih menggunakan printer 3D dengan material PLA. Wadah spons dirancang sebagaiudukan spons yang terhubung dengan mekanisme *lead screw* sehingga dapat bergerak maju dan mundur mengikuti sistem pembersihan tabung. Proses pencetakan dilakukan berdasarkan hasil desain 3D yang telah dibuat sebelumnya.



Gambar 4. 14 Proses pemotongan busa pembersih

3) Memotong spon dan akrilik pembersih

Tahap berikutnya adalah proses pemotongan spons dan akrilik pembersih menggunakan mesin CNC laser. Desain dari SolidWorks terlebih dahulu dikonversi ke format DXF kemudian diproses menggunakan aplikasi RDWorks untuk mengatur pola pemotongan. Penggunaan CNC laser

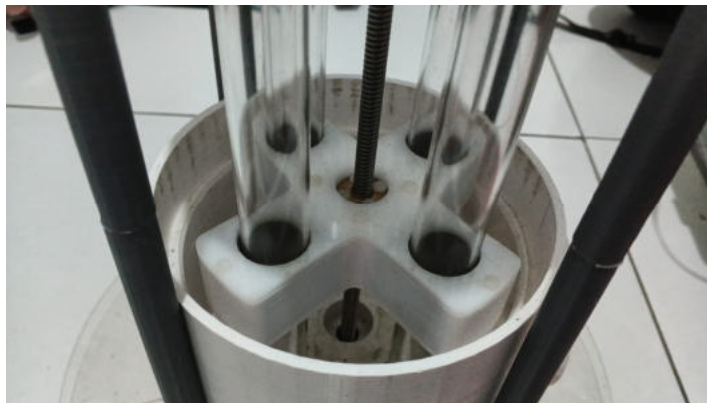
bertujuan memperoleh hasil potongan yang presisi sehingga spons dan akrilik dapat dipasang dengan baik pada mekanisme pembersih.



Gambar 4. 15 Hasil pemotongan akrilik dengan CNC laser

4) Perakitan komponen

Setelah seluruh komponen selesai dibuat, dilakukan proses perakitan meliputi pemasangan wadah spons, spons pembersih, akrilik, *lead screw*, serta komponen mekanik lainnya. Perakitan dilakukan sesuai hasil rancangan 3D agar mekanisme gerak sistem pembersih dapat bekerja dengan baik dan stabil saat proses pembersihan berlangsung.



Gambar 4. 16 Bentuk pembersih setelah di rakit

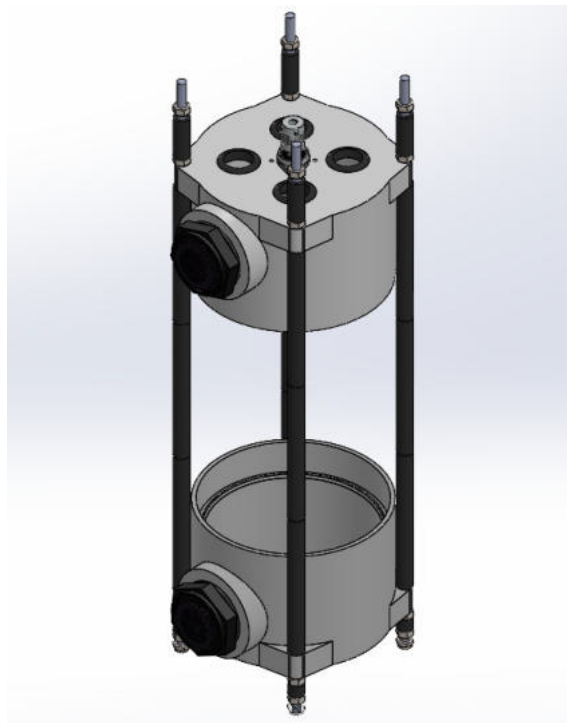
4.2.4 Pembuatan Tabung Reaktor

Pada pembuatan tabung reaktor dibagi menjadi 5 bagian yaitu, perancangan 3D model berdasarkan ukuran asli, mencetak cap dan *spacer* tabung, memotong akrilik

untuk *mounting* stepper dan kaki kaki, memotong ass drat penyangga, dan perakitan komponen.

1) Perancangan 3D model berdasarkan ukuran asli

Tahap perancangan dilakukan dengan membuat model 3D tabung reaktor menggunakan SolidWorks berdasarkan ukuran asli komponen dan dimensi tabung yang digunakan. Proses ini bertujuan untuk memperoleh visualisasi bentuk keseluruhan alat serta memastikan setiap komponen dapat terpasang dengan presisi. Selain itu, perancangan 3D juga membantu meminimalkan kesalahan ukuran pada tahap fabrikasi dan perakitan.



Gambar 4. 17 Tampak visual 3D desain tabung dan penyangga

2) Mencetak cap dan *spacer* tabung

Setelah desain 3D selesai dibuat, dilakukan proses pencetakan cap dan *spacer* tabung menggunakan printer 3D dengan material PLA. Sebelum proses pencetakan dilakukan, file model terlebih dahulu diproses menggunakan aplikasi OrcaSlicer untuk menentukan parameter pencetakan seperti ketebalan layer, *infill*, dan support. Hasil pencetakan digunakan

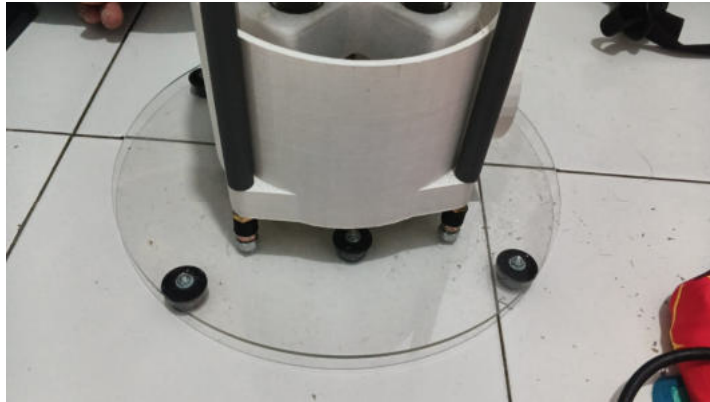
sebagai penutup tabung sekaligus *spacer* untuk menjaga posisi dan kestabilan komponen di dalam sistem reaktor.



Gambar 4. 18 Proses mencetak spacer dengan mesin 3D Print

3) Memotong akrilik untuk *mounting* stepper dan kaki kaki

Tahap berikutnya adalah pemotongan akrilik untuk pembuatan *mounting* motor stepper dan kaki-kaki penyangga alat. Desain hasil dari SolidWorks dikonversi ke format DXF kemudian diproses menggunakan aplikasi RDWorks untuk mengatur proses pemotongan dengan mesin CNC laser. Penggunaan CNC laser bertujuan memperoleh hasil pemotongan yang presisi sesuai desain yang telah dibuat.



Gambar 4. 19 Hasil pemotongan akrilik dengan mesin CNC laser

4) Memotong as drat penyangga

Pada tahap ini dilakukan pemotongan as drat sesuai ukuran yang telah ditentukan pada desain 3D. Proses pemotongan dilakukan menggunakan gerinda tangan agar diperoleh panjang as drat yang sesuai kebutuhan konstruksi alat. As drat digunakan sebagai struktur penyangga utama untuk menghubungkan bagian tabung, *mounting*, dan kaki-kaki alat sehingga konstruksi reaktor menjadi lebih kokoh dan stabil.



Gambar 4. 20 Hasil pemotongan As drat

5) Perakitan komponen

Setelah seluruh komponen mekanik selesai dibuat, dilakukan proses perakitan meliputi pemasangan cap tabung, *spacer*, *mounting* motor stepper, kaki-kaki penyangga, mekanisme pembersih Sleeve, serta as drat. Perakitan dilakukan sesuai hasil visualisasi dan rancangan 3D yang telah

dibuat sebelumnya agar seluruh komponen dapat terpasang dengan baik dan sesuai posisi yang direncanakan.



Gambar 4. 21 Tampak alat setelah perakitan

4.3 Pembuatan Perangkat Lunak (*Software*)

Perangkat lunak pada alat pembersih sleeve lampu UV sterilizer dirancang menggunakan bahasa pemrograman C++ pada platform Arduino IDE dengan mikrokontroler ESP32 sebagai pusat kendali sistem. Program dikembangkan secara modular untuk mempermudah proses pengembangan, pemeliharaan, dan penambahan fitur pada sistem. Secara umum, perangkat lunak bertugas mengatur seluruh proses operasi alat yang meliputi antarmuka pengguna, kalibrasi

mekanisme pembersih, pengaturan interval pembersihan, pengendalian motor stepper, pencatatan data, serta komunikasi dengan layanan penyimpanan data berbasis *cloud*. Pembagian sistem perangkat lunak dibagi menjadi beberapa bagian anatara lain,

1. Struktur program utama
2. Sistem antarmuka menu
3. Mekanisme kalibrasi alat
4. Pembersihan Sleeve lampu UV
5. Pengiriman Data ke Spreadsheet

4.3.1 Struktur Program Utama

Pada struktur program utama mengacu pada *framework* Arduino yang dibagi menjadi dua bagian utama, yaitu fungsi *setup()* dan fungsi *loop()*. Fungsi *setup()* dijalankan satu kali ketika sistem pertama kali dinyalakan dan berfungsi untuk melakukan inisialisasi seluruh perangkat keras maupun perangkat lunak yang digunakan. Sementara itu, fungsi *loop()* merupakan bagian utama program yang dijalankan secara terus-menerus selama alat beroperasi. Fungsi ini bertanggung jawab dalam menjalankan algoritma sistem, pembacaan masukan pengguna, proses pembersihan otomatis, serta pengelolaan data monitoring.

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Gambar 4. 22 kerangka dasar program Arduino.

Sebelum memasuki kedua fungsi tersebut, program diawali dengan deklarasi library, konfigurasi pin masukan dan keluaran, deklarasi variabel global,

struktur data, serta pembuatan objek yang digunakan oleh seluruh sistem. Pada bagian ini juga dilakukan pendefinisian parameter yang berkaitan dengan RTC, komunikasi WiFi, sistem penyimpanan data, antarmuka LCD, rotary encoder, dan mekanisme penggerak motor stepper. Pendefinisian tersebut bertujuan untuk mempermudah pengelolaan program serta meningkatkan keterbacaan kode program.

```
//-----INCLUDE LIBRARY SECTION-----//
#include < Arduino.h >
#include < LiquidCrystal_I2C.h >
#include < RTCLib.h >
#include "AiEsp32RotaryEncoder.h"
#include "stepper.h"
#include "indicator.h"
#include "esp_task_wdt.h"
#include "time.h"
#include < WiFi.h >

#include < FS.h >
#include < LittleFS.h >
#include < ESP_Google_Sheet_Client.h >
#include < GS_SDHelper.h >
```

Gambar 4. 23 Deklarasi library

```
//ROTARY ENCODER PIN
#define SW_ENCODER      25
#define DT_ENCODER      32
#define CLK_ENCODER      33
#define ENCODER_STEP      5
```

Gambar 4. 24 Deklarasi pin rotary encoder

```
//RTC
char daysOfTheWeek[7][12] = {"MINGGU", "SENIN",
"SELASA", "RABU", "KAMIS", "JUMAT", "SABTU"};
```

Gambar 4. 25 Deklarasi hari pada RTC

```
#define PROJECT_ID "monitoring-tugas-akhir-492206"
#define CLIENT_EMAIL "ta-datalogging@monitoring-tugas-akhir-492206.iam.gserviceaccount.com"
const char PRIVATE_KEY[] PROGMEM = "-----BEGIN PRIVATE KEY-----
\nMIEvgIBADANBgkqhkiG9w0BAQEFAASCBKggggSkAgEAAoIBAQRdG80+8UoSLuYc\nznxF/uxAX2B8340eKzSVH0ZWIrQZxj0JmPTS6/UaPAYutLjDxRiXPZUoBJXRDMj2\nnYMHe2ZrotqFkZtAE/jhPR7rnc39wUyuA2K49WelvDz2edzZ5sB+4dZDA28oI27RF\n\nvMQwAkTfSGZIE5UgOVfX8qJm4Rvo/ifM0b08y87aPzZ9sCDHK17MC7aVjInAEKYA\n\nnChvPLeAf9Lwyr1+oxSrgW4ziQTZtHRxWOXKDS+SSGhs0BYd1AEjyMHca2GFSMqgp\n\nn41LjIyuLPs21eiUftAXp42btHsGdJDQMUBYhhEH+DHI5Fyklki4Q/ZfRQLbylKow\n\nn4ZHtrq75AgMBAAECggEAEfqB9R6XTRaxtYLMjQv6RD3xmfsEYfm52VbSlnmsPiBW\n\n\nnaz/Ne70G5E0HcquSnt3upc9+yC2kQKASehd8xoxAzV+nSI9C2cQ7DXerHPYhwow0\n\n\nnUn0/d0+Yi0v7ZqfafERYAzVyWV01aPysH30c9Pq8G4iQJx41lDOZD8QtPngjyJWp\n\n\nnb8G3zbg2s+lKoxC4DSB/E4+AULiGi7XcJ7N0Ddx/jUrQIVraFz9ubXbM7QD6KpCL\n\n\nnTECLMNDKoCDLG5vdGsn5QgDqRUcKhMvsfGq10seUA3ugCIxhoztRL2+xfU0p7W1M\n\n\nnNrg59oV2YBc7fQWKnAcfR1AxnQQ/3aDaTTAnkVv4EQKBgQD77q2qX8YNYbwwqjox\n\n\nnJboBYX1pKnsKj55FQsNmWfWWuNIK6W9I9yzqQ+Bbg08YX6m/sFdqA60bkSYN8VvL\n\n\nn53wE7THQrP5wMZF7wr1F2wixH1Ij8se+V5EeeTxI9dksyFjhmVM1rVsaIZhdtwME\n\n\n\nnaNVfmA01Y8wXCzOo5PoSQVIpRQKBgQDu55XNLw6wuhU0fuMBMXrSTaGz0uyBIqet\n\n\n\nn170x246+1NJJacor3sHvExijWfhXu1TWnS0h5JLpcbOKh20qFXkGiVG6QXI5ZegX\n\n\n\nn3j1rfm9Mp2nV+Hswp99jUthHEbcfULAQMk2owJt1C3NtzArV+/AtqjDnPnOBwV9F\n\n\n\nnm0K1HJNYJQKBgQCodQpu+CeIXyflKo5Kur1LmxKutZcckrfz/BaIY/LR27enUDv3\n\n\n\nngD/qTYTB6s71b1IkORfababc3guiGxeUmq7U5H4A2QwxnRN4u9YDHdkMBghKvVZO\n\n\n\nnsmL5C3wCFVionbbA8zMum0J1kL0/PNscTM8QYq3CW37mkbjzYFjOyBrW0QKBgGMz\n\n\n\nnXHunt5Tgn999dKtRLWvX+nYd9kCk8DV6aMAKKA DLhEohmGZhB+xK5Kcnu7xJzeKT\n\n\n\nnu3EE5Y/zu5ahI1xDsE5rhLkvZcu5qgcHLD/Pb03IEmVb2ViN+m9WviG81Gp3abQ\n\n\n\n\n/n/wk8eQobGCIWYMV76L49djL/Np1Be1r+3lMv+2rVAoGBAMInOUjeyxpN6u/EjesE\n\n\n\n\nnriuQLV00z1IUICvxn6TavVPIWwCEJAYUUU9phPjbWQJs5vD30F4WRAMa5iesJFxu\n\n\n\n\nnCbnX51mij+uDATHC3E85UBTC6h2w8DJC1aOFQjERW8Hpx0gQ/ua+mDq0h+J40OpE\n\n\n\n\nn5wENggRvUZqQBFWKSSTMU92T\n\n\n\n\nn-----END PRIVATE KEY-----\n";
```

```
const char spreadsheetId[] = "196fPXT2wCThiAbBL25cXKxXfoJ-  
tBKkoQMy26suv11s";
```

Gambar 4. 27 Deklarasi parameter google spreadsheet

```
//NTP  
const char * ntpServer = "pool.ntp.org";  
const char * ntpServer2 = "id.pool.ntp.org";  
const long gmtOffset_sec = 7 * 3600;  
const int daylightOffset_sec = 0;  
unsigned long epochTime;
```

Gambar 4. 28 Deklarasi parameter NTP

```
//ENUM  
typedef enum{  
    NOT_SET = 0,  
    ON_TOP,  
    GOTO_BOTTOM,  
    GOTO_TOP,  
    ON_BOTTOM  
}stepperPosition;  
  
typedef enum{  
    HOME_1 = 0,  
    HOME_2,  
    HOME_3,  
    HOME_4,  
    TOTAL_HOME_MENU  
}homePageMenu;  
  
typedef enum{  
    BACK = 0,  
    CALIBRATION,  
    SETUP_INTERVAL,  
    SETUP_PULSE,  
    LAMP_CHECK,
```

```

    RESET_DATA,
    TOTAL_SUBMENU_SELECTION
}subMenuSelection;

```

Gambar 4. 29 Deklarasi enum posisi stepper, menu homepage, dan submenu

```

//STRUCTURE
struct sistemSetup{
    int8_t stepPos;
    int32_t totalStep;    long pulseCalibration;
    int32_t lastStepReading;
    int cleanInterval;
    int totalClean;
};

struct logTimeSistem{
    DateTime logTimeBoot[99];
    int32_t totalLog;
    bool logFull;
};

struct sendToSpreadsheet{
    DateTime time;
    int8_t stepPos;
    int totalClean;
};

struct pendingDataSpreadsheet{
    bool pending;
    sendToSpreadsheet data;
};

const char * systemFile = "/config.bin";
const char * loggerFile = "/logger.bin";
const char * pendingFile = "/pendingData.bin";

```

Gambar 4. 30 Deklarasi struct dan nama filesystem

```

//-----OBJECT DECLARATION-----//
LiquidCrystal_I2C lcd(0x27,20,4);
AiEsp32RotaryEncoder rotaryEncoder
                        = AiEsp32RotaryEncoder(CLK_ENCODER,
DT_ENCODER,
SW_ENCODER,
-1,
ENCODER_STEP);
sistemSetup DataSet;
FirebaseJson valueRange;
FirebaseJson response;
logTimeSistem DataLog;
sendToSpreadsheet GsData;
pendingDataSpreadsheet GsPend[50];
RTC_DS3231 rtc;
stepper St;

```

Gambar 4. 31 Deklarasi objek class dan library

Pada fungsi *setup()*, sistem melakukan serangkaian proses inisialisasi yang meliputi konfigurasi komunikasi serial, inisialisasi RTC DS3231, rotary encoder, motor stepper, LCD, koneksi WiFi, sinkronisasi waktu melalui NTP, inisialisasi Google Spreadsheet, pembacaan data konfigurasi dari LittleFS, serta proses penyesuaian posisi mekanisme pembersih berdasarkan data terakhir yang tersimpan. Setelah seluruh proses inisialisasi selesai dilakukan, sistem menampilkan informasi awal pada LCD dan masuk ke mode operasi normal.

```

void setup() {
    delay(1000);
    Serial.begin(115200);
    indicatorSetup();
    rtcBegin();
    rotaryBegin();
    St.begin();
    timeOff = St.setRPM(100);
    lcdBegin();

```

```

welcome();
setupWiFi();
configTime(gmtOffset_sec,
           daylightOffset_sec,ntpServer2);
waitNtpTime();
rtcAdjustTime();
GsheetSetup();
loadDataOnSetup();
adjustCleanerOnSetup();
showTime();
St.enableStepper(ON_STEPPER);
}

```

Gambar 4. 32 Listing program inialisasi sistem pada fungsi setup()

Pada fungsi *loop()*, program dibagi menjadi dua mode operasi utama yaitu menu informasi dan menu pengaturan parameter. Menu informasi digunakan untuk menampilkan data operasional alat seperti waktu sistem, interval pembersihan, jumlah pembersihan, dan posisi mekanisme cleaner. Sementara itu, menu pengaturan parameter digunakan untuk melakukan konfigurasi alat seperti pengaturan interval pembersihan, kalibrasi mekanisme pembersih, pengaturan parameter motor stepper, serta reset sistem. Perpindahan antar menu dilakukan menggunakan rotary encoder sebagai media interaksi pengguna.

```

void loop() {
    int lastMenuPos;
    int mainMenu = 0;
    unsigned long lastSerialPrintTime = millis();
    DateTime runTime = rtc.now();
    DateTime cleanTime = runTime + TimeSpan(0,0,DataSet.cleanInterval,0);
    rotaryEncoder.setBoundaries(0,TOTAL_HOME_MENU - 2,false);
    lastMenuPos = mainMenu;
    while (1){
//MENU PEMBERSIHAN DAN EKSEKUSI PEMBERSIHAN
    }
    int lastSubMenu;

```

```

    int subMainMenu = 0;
    int16_t SubMainLastClick = 0;
    rotaryEncoder.setBoundaries(0,TOTAL_SUBMENU_SELECTION - 1,false);
    rotaryEncoder.setEncoderValue(0);
    lcd.createChar(0,panah);
    lastSubMenu = subMainMenu;
    while (1){
//MENU PENGATURAN PARAMETER
    }
}

```

Gambar 4. 33 Ringkasan program menu dan eksekusi pembersihan alat pada fungsi loop()

Selain mengelola antarmuka pengguna, fungsi loop() juga melakukan pemeriksaan waktu pembersihan secara berkala. Ketika waktu pembersihan yang telah ditentukan tercapai, sistem akan menjalankan mekanisme pembersihan otomatis dengan menggerakkan cleaner menggunakan motor stepper. Setelah proses pembersihan selesai, sistem memperbarui data operasional, menyimpan data ke media penyimpanan internal, dan melakukan pengiriman data monitoring ke Google Spreadsheet apabila koneksi internet tersedia. Apabila koneksi internet terputus, data akan disimpan sementara menggunakan mekanisme *pending* data dan dikirim kembali ketika koneksi berhasil dipulihkan.

4.3.2 Sistem Antarmuka Menu

Sistem antarmuka menu dirancang sebagai media interaksi antara pengguna dan alat menggunakan LCD I2C 20×4 serta rotary encoder. Antarmuka ini memungkinkan pengguna untuk melakukan pemantauan kondisi sistem, mengubah parameter operasi, serta menjalankan fungsi-fungsi tertentu tanpa memerlukan perangkat tambahan. Seluruh proses navigasi dilakukan menggunakan rotary encoder, dimana putaran encoder digunakan untuk berpindah menu sedangkan tombol encoder digunakan untuk memilih menu atau mengonfirmasi suatu pengaturan.

//ROTARY ENCODER

```
void IRAM_ATTR readEncoderISR(){
    rotaryEncoder.readEncoder_ISR();
}
```

Gambar 4. 34 Program pembacaan encoder dengan interrupt

Navigasi pada menu informasi dilakukan dengan memutar rotary encoder sehingga pengguna dapat berpindah antar halaman informasi yang tersedia. Program menggunakan sistem pembatas (*boundaries*) pada rotary encoder sehingga perpindahan halaman hanya terjadi pada rentang menu yang telah ditentukan. Setiap perubahan posisi rotary encoder akan menyebabkan sistem memperbarui tampilan LCD sesuai halaman yang sedang dipilih. Dengan metode ini, informasi dapat ditampilkan secara terstruktur dan mudah dipahami oleh pengguna.

```
rotaryEncoder.setBoundaries(0, TOTAL_HOME_MENU - 2, false);
```

Gambar 4. 35 Program setting pembatas encoder

```
mainMenu = rotaryEncoder.readEncoder();
menuDisplayV2(mainMenu, cleanTime);
```

Gambar 4. 36 Program pergantian tampilan menu utama

Apabila tombol rotary encoder ditekan pada menu informasi, sistem akan berpindah menuju menu pengaturan parameter. Pada menu ini tersedia beberapa fungsi yang dapat digunakan pengguna untuk melakukan konfigurasi alat, yaitu menu kalibrasi mekanisme cleaner, pengaturan interval pembersihan, pengaturan nilai pulse motor stepper, serta reset sistem. Setiap pilihan menu ditampilkan menggunakan indikator panah pada LCD sehingga pengguna dapat mengetahui posisi menu yang sedang dipilih.

```
if (rotaryEncoder.isEncoderButtonClicked()) {
    Serial.println("JUMP TO SUB MENU");
    lcd.clear();
    delay(300);
    break;
}
```

Gambar 4. 37 Program untuk masuk ke menu parameter

Ketika pengguna memilih salah satu menu pengaturan, program akan menjalankan fungsi yang sesuai dengan parameter yang dipilih. Sebagai contoh, pada menu pengaturan interval pembersihan, rotary encoder digunakan untuk menentukan interval waktu pembersihan dalam satuan menit. Pada menu pengaturan pulse stepper, rotary encoder digunakan untuk menentukan nilai pulse yang digunakan sebagai parameter gerakan motor stepper. Setelah parameter dipilih dan tombol encoder ditekan, nilai tersebut akan disimpan ke dalam memori LittleFS sehingga tetap tersimpan meskipun sistem dimatikan.

```

subMainMenu = rotaryEncoder.readEncoder();
if(subMainMenu != lastSubPageshow) lcd.clear();
//ISI SUB MENU &CURSOR
if (subMainMenu == BACK || subMainMenu == LAMP_CHECK){
    lcd.setCursor(0,0);
    lcd.write(0);
}else if(subMainMenu == CALIBRATION || subMainMenu == RESET_DATA){
    lcd.setCursor(0,1);
    lcd.write(0);
}else if (subMainMenu == SETUP_INTERVAL){
    lcd.setCursor(0,2);
    lcd.write(0);
}else if (subMainMenu == SETUP_PULSE){
    lcd.setCursor(0,3);
    lcd.write(0);
}
if (subMainMenu < LAMP_CHECK){
    lcd.setCursor(1,0);
    lcd.print("KEMBALI");
    lcd.setCursor(1,1);
    lcd.print("KALIBRASI STEP");
    lcd.setCursor(1,2);
    lcd.print("SET INTERVAL");
    lcd.setCursor(1,3);
    lcd.print("SET PULSE");
}

```

```

}else if(subMainMenu >= LAMP_CHECK){
    lcd.setCursor(1,0);
    lcd.print("CEK LAMPU");
    lcd.setCursor(1,1);
    lcd.print("RESET SISTEM");
}

```

Gambar 4. 38 Program tampilan menu pada menu setting parameter

Selain berfungsi sebagai media konfigurasi, sistem antarmuka juga dirancang untuk memberikan informasi status proses yang sedang berlangsung. Saat proses pembersihan berlangsung, pengiriman data dilakukan, atau ketika sistem melakukan sinkronisasi waktu dan koneksi jaringan, LCD akan menampilkan pesan status yang sesuai. Dengan demikian pengguna dapat mengetahui kondisi operasi alat secara langsung tanpa perlu melakukan pemantauan melalui komputer atau perangkat lainnya.

```

if(DataSet.stepPos == ON_TOP){
    lcd.setCursor(0,0);
    lcd.print("=== START CLEANING ===");
    lcd.setCursor(0,1);
    lcd.print("< GOTO BOTTOM >");
}else if (DataSet.stepPos == ON_BOTTOM){
    lcd.setCursor(0,0);
    lcd.print("=== START CLEANING ===");
    lcd.setCursor(0,1);
    lcd.print("< GOTO TOP >");
}

```

Gambar 4. 39 Program tampilan ketika melakukan pembersihan

```

lcd.clear();
lcd.setCursor(0,0);
lcd.print("Waiting for GSheet");
Serial.println("Waiting for GSheet to be ready...");
if (ready) break;
}

```

```

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Sending Data....");

```

Gambar 4. 40 Program tampilan ketika data dikirimkan ke spreadsheet

4.3.3 Mekanisme Kalibrasi Alat

Mekanisme kalibrasi alat digunakan untuk menentukan panjang lintasan aktual mekanisme pembersih yang bergerak menggunakan sistem *lead screw*. Proses kalibrasi diperlukan karena panjang lintasan yang ditempuh cleaner harus diketahui secara tepat agar proses pembersihan dapat berlangsung dari ujung atas hingga ujung bawah sleeve lampu UV secara optimal. Nilai hasil kalibrasi selanjutnya digunakan sebagai acuan pada setiap proses pembersihan otomatis yang dijalankan sistem.

Proses kalibrasi diawali dengan pengguna memilih menu kalibrasi pada antarmuka sistem. Setelah proses dimulai, motor stepper akan menggerakkan cleaner menuju posisi atas. Ketika cleaner telah mencapai batas atas yang diinginkan, pengguna dapat menghentikan pergerakan melalui tombol rotary encoder. Selanjutnya sistem menyediakan fitur penyesuaian posisi secara manual menggunakan putaran rotary encoder untuk memperoleh posisi awal yang lebih presisi.

```

void kalibrasiStep(){
    //Inisialisasi default parameter
    while (1){//START KALIBRASI
        //Konfirmasi prose kalibrasi
        }
        while (1){//SLIDER KEATAS
            //Pembersih bergerak ke posisi atas
            }
            while (1){//ADJUST POSISI SLIDER
                //Mengatur posisi atas pembersih secara manual dengan rotary
            }
            while (1){

```

```

//Pembersih bergerak ke posisi bawah dan menghitung step
}
while (1){//ADJUST POSISI SLIDER
//Mengatur posisi bawa pembersih dan kalkulasi step total
}
}

```

Gambar 4. 41 Rangkuman program fungsi kalibrasi step

Setelah posisi atas ditentukan, cleaner digerakkan menuju posisi bawah sambil dilakukan perhitungan jumlah langkah motor stepper yang ditempuh selama perpindahan tersebut. Nilai jumlah langkah yang diperoleh akan digunakan sebagai panjang lintasan sementara mekanisme pembersih. Ketika cleaner mencapai posisi bawah, pengguna kembali dapat melakukan penyesuaian posisi akhir menggunakan rotary encoder sehingga titik akhir lintasan sesuai dengan kondisi mekanik aktual alat.

```

while (1){//SLIDER KEATAS
  St.oneStep(PULSE_STEP,timeOff);
  if (rotaryEncoder.isEncoderButtonClicked()) {
    Serial.println("ADJUST TOP POS");
    St.enableStepper(OFF_STEPPER);
    lcd.clear();
    delay(200);
    break;
  }
}

```

Gambar 4. 42 Program pembersih bergerak ke posisi atas

```

lcd.setCursor(0,0);
lcd.print(" SLIDER GOTO BOTTOM");
lcd.setCursor(0,1);
lcd.print(" < PRESS TO STOP >");
while (1){
  St.moveOneDistace(DEFAULT_PULSE_10MM,CW_ROTATE,PULSE_STEP,timeOff);
  stepCount ++;
}

```

```

if (rotaryEncoder.isEncoderButtonClicked()) {
    Serial.println("ADJUST BOT POS");
    St.enableStepper(OFF_STEPPER);
    DataSet.totalStep = stepCount;
    DataSet.stepPos = ON_BOTTOM;
    saveData(systemFile, &DataSet, sizeof(DataSet));
    lcd.clear();
    break;
}
}

```

Gambar 4. 43 Program pembersih bergerak ke bawah dan perhitungan step pembersih

Setelah proses penyesuaian selesai, jumlah langkah hasil kalibrasi disimpan pada variabel totalStep dan dituliskan ke memori LittleFS. Selain menyimpan jumlah langkah lintasan, sistem juga menyimpan informasi posisi cleaner dan parameter kalibrasi lainnya yang diperlukan untuk proses operasi berikutnya. Dengan penyimpanan tersebut, proses kalibrasi tidak perlu dilakukan kembali setiap kali alat dinyalakan.

```

while (1){//ADJUST POSISI SLIDER
    sprintf(buff2," < %02d > ",stepCount);
    int currentEncoderPos = rotaryEncoder.readEncoder();

    if (currentEncoderPos != lastAdjustBotPos) {
        if (currentEncoderPos > lastAdjustBotPos) {
            St.moveOneDistace(DEFAULT_PULSE_10MM,CCW_ROTATE,PULSE_STEP,timeOff);
            stepCount --;
            lcd.clear();
        } else {
            St.moveOneDistace(DEFAULT_PULSE_10MM,CW_ROTATE,PULSE_STEP,timeOff);
            stepCount ++;
            lcd.clear();
        }
    }
    if (stepCount < 0) stepCount = 0;
}

```

```

    lastAdjustBotPos = currentEncoderPos;
}
int currentDigits = digitCount(stepCount);
if (currentDigits != lastDigits) {
    lcd.clear();
    lastDigits = currentDigits;
}
lcd.setCursor(0,0);
lcd.print(" ROTATE TO ADJUST");
lcd.setCursor(7,1);
lcd.print(buff2);
lcd.setCursor(0,2);
lcd.print(" PRESS TO SAVE");
lcd.setCursor(0,3);
lcd.print("    < PRESSSS >");
if (rotaryEncoder.isEncoderButtonClicked()) {
    Serial.println("START CALCULATE STEP");
    rotaryEncoder.correctionOffset = 4;
    St.enableStepper(OFF_STEPPER);
    DataSet.totalStep = stepCount;
    saveData(systemFile,&DataSet,sizeof(DataSet));
    lcd.clear();
    delay(200);
    break;
}
}

```

Gambar 4. 44 Program pengaturan posisi slider posisi bawah dan menyimpan nilai total step

4.3.4 Pembersihan Sleeve Lampu UV

Mekanisme pembersihan sleeve lampu UV merupakan fungsi utama dari sistem yang dirancang untuk mengurangi penumpukan kotoran pada permukaan sleeve secara otomatis. Proses pembersihan dilakukan menggunakan mekanisme

cleaner yang digerakkan oleh motor stepper melalui sistem *lead screw*. Dengan adanya mekanisme ini, proses pembersihan dapat dilakukan secara berkala tanpa memerlukan intervensi operator sehingga diharapkan mampu menjaga efektivitas penyinaran lampu UV selama proses sterilisasi berlangsung.

Proses pembersihan dijalankan berdasarkan interval waktu yang telah ditentukan oleh pengguna melalui menu pengaturan. Sistem memanfaatkan RTC DS3231 sebagai sumber waktu utama untuk menghitung waktu pembersihan berikutnya. Pada program utama, waktu saat ini dibandingkan dengan waktu pembersihan yang telah dijadwalkan. Ketika waktu pembersihan telah tercapai dan interval pembersihan telah diatur, sistem akan memulai proses pembersihan secara otomatis.

```
DateTime cleanTime = runTime + TimeSpan(0,0,DataSet.cleanInterval,0);
```

Gambar 4. 45 Program kalkulasi waktu pembersihan berdasarkan parameter interval

```
if ((runTime.unixtime() >= cleanTime.unixtime()) && DataSet.cleanInterval != 0)
```

Gambar 4. 46 Program kondisi untuk eksekusi proses pembersihan

Sebelum proses pembersihan dimulai, sistem terlebih dahulu memeriksa posisi terakhir cleaner yang tersimpan pada variabel *stepPos*. Informasi posisi ini digunakan untuk menentukan arah pergerakan motor stepper. Apabila cleaner berada pada posisi atas (*ON_TOP*), maka motor stepper akan menggerakkan cleaner menuju posisi bawah. Sebaliknya, apabila cleaner berada pada posisi bawah (*ON_BOTTOM*), maka cleaner akan digerakkan menuju posisi atas. Dengan metode ini, proses pembersihan dilakukan secara bergantian dari satu ujung lintasan menuju ujung lainnya sehingga seluruh permukaan sleeve dapat dibersihkan secara merata.

```
if(DataSet.stepPos == ON_TOP){
    lcd.setCursor(0,0);
    lcd.print("=== START CLEANING ===");
    lcd.setCursor(0,1);
    lcd.print("< GOTO BOTTOM >");
```

```

} else if (DataSet.stepPos == ON_BOTTOM){
    lcd.setCursor(0,0);
    lcd.print(" === START CLEANING === ");
    lcd.setCursor(0,1);
    lcd.print(" <  GOTO TOP  > ");
}

```

Gambar 4. 47 Program cek posisi pembersih

Selama proses perpindahan berlangsung, motor stepper bergerak berdasarkan jumlah langkah yang diperoleh dari hasil kalibrasi dan disimpan pada variabel *totalStep*. Setiap langkah yang ditempuh akan dicatat pada variabel *lastStepReading* dan disimpan ke dalam memori LittleFS. Mekanisme ini berfungsi sebagai sistem *failsafe* apabila terjadi gangguan daya saat proses pembersihan sedang berlangsung. Dengan adanya pencatatan posisi tersebut, sistem dapat mengetahui posisi terakhir cleaner dan melakukan pemulihan posisi ketika alat dinyalakan kembali.

```

void goToClean(int posCleaner){
    if (posCleaner == ON_BOTTOM){
        DataSet.stepPos = GOTO_TOP;
        saveData(systemFile,&DataSet,sizeof(DataSet));
        St.enableStepper(ON_STEPPER);
        for (int32_t i = 0; i < DataSet.totalStep; i++){
            St.moveOneDistace(DEFAULT_PULSE_10MM,CCW_ROTATE,PULSE_STEP,timeOff);
            DataSet.lastStepReading++;
            saveData(systemFile,&DataSet,sizeof(DataSet));
            delayMicroseconds(5);
        }
        DataSet.stepPos = ON_TOP;
        St.enableStepper(OFF_STEPPER);
        DataSet.lastStepReading = 0;
        saveData(systemFile,&DataSet,sizeof(DataSet));
    } else if (posCleaner == ON_TOP){
        DataSet.stepPos = GOTO_BOTTOM;
        saveData(systemFile,&DataSet,sizeof(DataSet));
    }
}

```

```

St.enableStepper(ON_STEPPER);
for (int32_t i = 0; i < DataSet.totalStep; i++){
    St.moveOneDistace(DEFAULT_PULSE_10MM,CW_ROTATE,PULSE_STEP,timeOff);
    DataSet.lastStepReading++;
    saveData(systemFile,&DataSet,sizeof(DataSet));
    delayMicroseconds(5);
}
DataSet.stepPos = ON_BOTTOM;
St.enableStepper(OFF_STEPPER);
DataSet.lastStepReading = 0;
saveData(systemFile,&DataSet,sizeof(DataSet));
delayMicroseconds(10);
}
}

```

Gambar 4. 48 Program eksekusi proses pembersihan dan algoritma failsafe jika daya terputus

Setelah cleaner mencapai posisi tujuan, sistem akan memperbarui status posisi cleaner menjadi ON_TOP atau ON_BOTTOM sesuai dengan arah gerakan yang telah dilakukan. Selanjutnya motor stepper dinonaktifkan untuk mengurangi konsumsi daya dan mencegah pemanasan yang tidak diperlukan pada *driver* motor. Sistem kemudian menambahkan nilai penghitung jumlah pembersihan (*totalClean*) sebagai data operasional alat yang nantinya digunakan untuk keperluan monitoring dan pencatatan data.

```

DataSet.stepPos = ON_TOP;
St.enableStepper(OFF_STEPPER);

```

Gambar 4. 49 Program update posisi ketika pembersih dari bawah ke atas

```

DataSet.stepPos = ON_BOTTOM;
St.enableStepper(OFF_STEPPER);

```

Gambar 4. 50 Program update posisi ketika pembersih dari atas ke bawah

```

DataSet.totalClean++;
saveData(systemFile,&DataSet,sizeof(DataSet));

```

Gambar 4. 51 Program update total pembersihan dan menyimpan parameter data ke littleFS

Setelah proses pembersihan selesai, sistem akan mengumpulkan data operasional yang meliputi waktu pelaksanaan pembersihan, posisi cleaner, dan jumlah total pembersihan yang telah dilakukan.

4.3.5 Pengiriman Data ke Spreadsheet

Sistem monitoring pada alat dilengkapi dengan fitur pengiriman data secara daring menggunakan layanan Google Spreadsheet. Fitur ini bertujuan untuk menyimpan data operasional alat secara otomatis sehingga dapat digunakan sebagai dokumentasi maupun pemantauan kinerja sistem. Komunikasi data dilakukan menggunakan koneksi WiFi yang tersedia pada ESP32 serta memanfaatkan Google Sheets API sebagai media pertukaran data antara perangkat dan spreadsheet.

Sebelum proses pengiriman data dilakukan, ESP32 terlebih dahulu melakukan koneksi ke jaringan WiFi dan melakukan sinkronisasi waktu menggunakan layanan Network Time Protocol (NTP). Waktu hasil sinkronisasi kemudian digunakan untuk memperbarui RTC DS3231 sehingga data yang dikirim ke spreadsheet memiliki informasi waktu yang akurat. Setelah koneksi internet dan autentikasi Google Spreadsheet berhasil dilakukan, sistem akan memasuki kondisi siap untuk melakukan pengiriman data.

```
void setupWiFi(){
    WiFi.setAutoReconnect(true);
    WiFi.begin(WIFI_SSID,WIFI_PASSWD);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Connect To WI – FI");
    Serial.print("Connecting to Wi – Fi");
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(1000);
        disconnectWifi ++;
        if (disconnectWifi > 3) break;
    }
}
```

```

}

if(disconnectWifi > 3 && WiFi.status() != WL_CONNECTED){
  Serial.println();
  Serial.println("Failed to connect to Wi-Fi");
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" Failed To Connect.");
}else if(WiFi.status() == WL_CONNECTED){
  Serial.println();
  Serial.println("Wi-Fi connected successfully");
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" Wi-Fi Connected.");
  delay(1000);
}
lcd.clear();
}

```

Gambar 4. 52 Program koneksi ESP32 dengan wifi

```

configTime(gmtOffset_sec, daylightOffset_sec, ntpServer2);
waitNtpTime();
rtcAdjustTime();

```

Gambar 4. 53 Program set NTP server dan adjust waktu RTC berdasarkan waktu server

Proses pengiriman data dilakukan setiap kali mekanisme pembersihan sleeve lampu UV selesai dijalankan. Pada tahap ini sistem akan mengumpulkan data operasional yang meliputi waktu pencatatan, posisi terakhir cleaner, serta jumlah total pembersihan yang telah dilakukan alat. Data tersebut kemudian disusun ke dalam format yang sesuai dengan Google Sheets API dan dikirimkan ke spreadsheet yang telah ditentukan sebelumnya.

```

collectData();
if(WiFi.status() != WL_CONNECTED) WiFi.reconnect();
if(WiFi.status() == WL_CONNECTED){
  if(configExists(pendingFile)){
    checkPendingData();
    pendingQueueSend();
    collectData();
    sendToGoogleSheets();
  }else{
    sendToGoogleSheets();
  }
}else{
  checkPendingData();
  checkPendingSlot();
}
cleanTime = runTime + TimeSpan(0,0,DataSet.cleanInterval,0);
lcd.clear();

```

Gambar 4. 54 Program pengambilan data monitoring dan proses pengiriman data ke spreadsheet

Untuk meningkatkan keandalan sistem, perangkat lunak dilengkapi mekanisme *failsafe* pengiriman data. Ketika koneksi internet tidak tersedia atau proses pengiriman data mengalami kegagalan, data operasional tidak langsung dihapus melainkan disimpan sementara ke dalam memori LittleFS sebagai data *pending*. Setiap data yang gagal dikirim akan ditempatkan pada antrean (*queue*) sehingga informasi operasional tetap tersimpan meskipun terjadi gangguan jaringan komunikasi.

```

checkPendingData();
checkPendingSlot();

```

Gambar 4. 55 Fungsi failsafe pengiriman data ketika sistem tidak terhubung ke wifi

```

void checkPendingSlot(){
    int pendingIndex = 0;
    for (int i = 0; i < 50; i ++){
        if(GsPend[i].pending == false){
            pendingIndex = i;
            break;
        }
    }
    GsPend[pendingIndex].pending = true;
    GsPend[pendingIndex].data = GsData;
    saveData(pendingFile,&GsPend,sizeof(GsPend));
}

```

Gambar 4. 56 Program failsafe data dan disimpan sementara ke littleFS

Pada saat koneksi internet kembali tersedia, sistem akan memeriksa keberadaan data *pending* yang tersimpan pada memori internal. Selanjutnya seluruh data yang berada dalam antrian akan dikirimkan kembali ke Google Spreadsheet secara berurutan sesuai urutan penyimpanannya. Setelah seluruh data berhasil dikirim, status data *pending* akan dihapus sehingga ruang penyimpanan dapat digunakan kembali untuk proses pencatatan berikutnya.

```

if(configExists(pendingFile)){
    checkPendingData();
    pendingQueueSend();
    collectData();
    sendToGoogleSheets();
}

```

Gambar 4. 57 Program cek data sementara yang tertinggal untuk dikirim ke spreadsheet

Penerapan mekanisme *failsafe* ini memungkinkan sistem tetap menjaga integritas data monitoring meskipun terjadi gangguan jaringan internet maupun kegagalan komunikasi dengan server Google Spreadsheet. Dengan demikian seluruh aktivitas pembersihan yang dilakukan alat tetap dapat terdokumentasi dengan baik tanpa kehilangan data operasional yang penting.