

BAB IV

PEMBUATAN ALAT

Proses pembuatan alat *dehydrator* listrik dilakukan berdasarkan hasil perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*) yang telah disusun pada bab sebelumnya. Tahap ini bertujuan untuk merealisasikan rancangan menjadi suatu sistem yang mampu mengendalikan temperatur ruang pengering secara otomatis serta melakukan pemantauan kondisi alat secara *real-time* melalui media monitoring berbasis internet yaitu Google Spreadsheet dan Telegram Bot.

Tahap pembuatan alat diawali dengan pemasangan elemen pemanas (*heater*) dan sensor thermocouple pada *dehydrator*, dilanjutkan dengan pemasangan komponen kelistrikan serta perakitan rangkaian elektronika lainnya pada box panel sesuai dengan wiring diagram yang telah dirancang. Setelah seluruh komponen terpasang, dilakukan proses pemrograman ESP32 untuk mengatur fungsi monitoring, pencatatan data, serta komunikasi dengan Google Spreadsheet dan Telegram Bot. Selanjutnya, dilakukan pengaturan parameter pada *Temperature Controller* TC4S agar sistem pengendalian suhu dapat bekerja sesuai dengan karakteristik proses pengeringan.

Sebelum seluruh sistem dioperasikan, dilakukan pengujian secara bertahap untuk memastikan seluruh perangkat berfungsi sesuai dengan perancangannya. Pengujian meliputi pemeriksaan catu daya, pembacaan suhu oleh thermocouple, kinerja *Temperature Controller* TC4S, proses pemanasan *heater*, tampilan LCD, komunikasi ESP32 dengan Google Spreadsheet, serta pengiriman notifikasi melalui Telegram Bot. Tahapan pengujian tersebut bertujuan untuk memastikan seluruh komponen dapat terintegrasi dengan baik sehingga sistem *dehydrator* listrik mampu beroperasi sesuai dengan fungsi yang direncanakan.

4.1 Alat dan Bahan

Pembuatan sistem *dehydrator* listrik memerlukan berbagai alat kerja dan bahan atau komponen yang digunakan selama proses perakitan, dan pengujian alat. Seluruh alat dan bahan dipersiapkan sesuai dengan kebutuhan agar setiap tahapan pembuatan dapat dilaksanakan secara sistematis dan menghasilkan sistem yang

sesuai dengan spesifikasi yang telah ditentukan. Alat yang digunakan berfungsi untuk mendukung proses pengerjaan mekanik dan perakitan rangkaian kelistrikan. Sementara itu, bahan yang digunakan terdiri atas komponen mekanik, komponen kelistrikan, dan komponen elektronika yang saling terhubung sehingga membentuk sistem pengendalian suhu dan monitoring pada *dehydrator* listrik.

Rincian alat dan bahan atau komponen yang digunakan selama proses pembuatan sistem disajikan pada Tabel 4- 1 dan Tabel 4- 2.

Tabel 4- 1 Daftar Alat yang Digunakan

No.	Alat	Fungsi
1	Obeng (+) dan (-)	Memasang dan melepas baut pada komponen serta box panel
2	Tang kombinasi	Menjepit, membengkokkan, dan memotong kabel
4	Tang pengupas kabel (<i>Wire Stripper</i>)	Mengupas isolasi kabel listrik
5	Multimeter digital	Mengukur tegangan, kontinuitas, dan sambungan rangkaian
6	Solder	Menyolder komponen elektronika
7	Timah solder	Bahan untuk membuat sambungan antar komponen elektronika
8	Bor listrik	Membuat lubang pada loyang, box panel, dll
9	Gerinda tangan	Memotong atau merapikan box panel untuk tempat LCD, keypad, dan TC4S
10	Laptop	Pemrograman ESP32, konfigurasi sistem, dan pengujian komunikasi IoT
11	Kabel USB	Menghubungkan ESP32 dengan laptop saat melakukan pemograman

Tabel 4- 2 Daftar Bahan yang Digunakan

No.	Bahan	Jumlah
1	ESP32 DevKit V1	1 buah
2	<i>Temperature Controller</i> TC4S R1TC	1 buah
3	Sensor Thermocouple Tipe-K	2 buah
4	Modul MAX6675	1 buah
5	<i>Solid State Relay</i> (SSR) 10 DA	1 buah
6	<i>Heater</i> Tubular 500 Watt	1 buah
7	Relay 1 Channel	1 buah
8	LCD I2C 16 x 2	1 buah
9	Keypad Matriks 4 x 4	1 buah
10	Adaptor <i>Power Supply</i> 5V-2A	1 buah
11	<i>Miniature Circuit Breaker</i> (MCB) 2A	1 buah
12	Box Panel 25 x 25 x 12	1 buah
14	Terminal Block	1 buah
15	Kabel NYAF 1,5 mm	2 meter
16	Kabel Jumper	2 meter
17	Baut, Mur, Spacer dan Ring	8 set
18	PCB	1 buah

4.2 Perancangan Perangkat Keras (*Hardware*)

Pada tahap pembuatan perangkat keras (*hardware*) dilakukan proses perakitan dan integrasi seluruh komponen yang digunakan pada sistem kontrol suhu *dehydrator* listrik. Perangkat keras yang digunakan meliputi bagian mekanik yaitu *dehydrator* listrik, serta bagian elektronika dan komponen pendukung lainnya. Seluruh komponen dirakit sesuai dengan rancangan sehingga membentuk satu

kesatuan sistem yang mampu menjalankan fungsi pengendalian suhu dan monitoring.

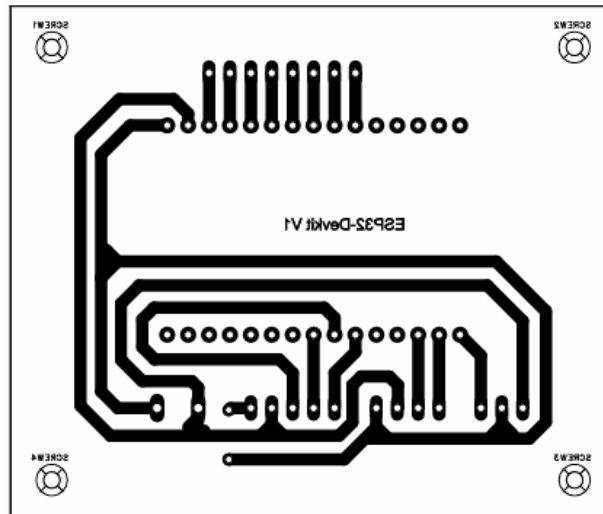
Proses pembuatan perangkat keras diawali dengan pemasangan elemen pemanas (*heater*) dan sensor thermocouple, kemudian dilanjutkan dengan perakitan komponen pada box panel. Setelah proses perakitan selesai, dilakukan pemeriksaan terhadap setiap jalur untuk memastikan seluruh koneksi telah terpasang dengan benar sebelum sistem dioperasikan. Tahapan tersebut dilakukan agar perangkat keras yang telah dirakit dapat berfungsi sesuai dengan kebutuhan sistem.

4.2.1 Perancangan Skematik, Layout, dan Pembuatan PCB

Perancangan skematik dan layout PCB dilakukan sebagai bagian dari perancangan perangkat keras untuk menentukan hubungan antarbagian rangkaian serta tata letak komponen pada papan rangkaian. Perancangan ini bertujuan agar pemasangan komponen dan pengkabelan sistem dapat dilakukan secara lebih terstruktur serta memudahkan proses perakitan perangkat keras.

Pada rancangan PCB yang dibuat, ESP32-DevKit V1 digunakan sebagai modul mikrokontroler utama. Papan rangkaian juga dilengkapi dengan terminal penghubung yang digunakan untuk menghubungkan rangkaian PCB dengan komponen pendukung lainnya pada sistem. Penempatan komponen disusun dengan mempertimbangkan kebutuhan koneksi dan kemudahan pemasangan pada perangkat, sehingga rangkaian dapat terintegrasi dengan bagian perangkat keras lainnya.

Layout PCB kemudian dirancang berdasarkan hasil perancangan skematik dengan memperhatikan posisi komponen, jalur koneksi, serta kerapian tata letak pada papan. Hasil rancangan layout tersebut digunakan sebagai acuan dalam proses pembuatan PCB dan perakitan rangkaian elektronik pada sistem *dehydrator* listrik. Rancangan PCB yang dihasilkan ditunjukkan pada Gambar 4- 1.



Gambar 4- 1 Desain PCB

Setelah proses perancangan skematik dan layout selesai, dilakukan proses pencetakan PCB berdasarkan desain yang telah dibuat. Pola jalur rangkaian dicetak pada permukaan PCB sesuai dengan layout yang telah dirancang. Setelah pola jalur terbentuk, PCB diproses untuk menghasilkan jalur penghantar sesuai dengan rancangan rangkaian.

Selanjutnya dilakukan proses pengeboran pada titik-titik yang telah ditentukan sebagai tempat pemasangan kaki komponen dan terminal penghubung. Posisi pengeboran disesuaikan dengan layout PCB agar komponen dapat dipasang sesuai dengan rancangan.

Setelah proses pembuatan jalur dan pengeboran selesai, dilakukan pemasangan komponen elektronika pada PCB. Komponen dipasang sesuai dengan posisi dan fungsi masing-masing berdasarkan rancangan yang telah dibuat. Selanjutnya dilakukan proses penyolderan pada kaki komponen dan titik sambungan untuk membentuk hubungan listrik antar komponen.

Setelah seluruh komponen terpasang dan disolder, dilakukan pemeriksaan terhadap jalur dan sambungan PCB untuk memastikan setiap koneksi telah terhubung dengan baik serta tidak terdapat hubungan singkat antarjalur yang tidak seharusnya terhubung. PCB yang telah selesai dirakit kemudian digunakan sebagai bagian dari rangkaian kontrol pada sistem *dehydrator* listrik.

4.2.2 Pemasangan *Heater* dan Sensor Thermocouple pada *Dehydrator*

Pada tahap ini dilakukan pemasangan elemen pemanas (*heater*) dan sensor thermocouple tipe-K yang bertujuan untuk membentuk sistem *dehydrator* yang mampu menghasilkan suhu sesuai dengan kebutuhan proses pengeringan serta menyediakan data suhu sebagai masukan bagi sistem kontrol suhu. Tahap ini menjadi bagian penting dalam proses pembuatan perangkat keras karena posisi pemasangan *heater* dan sensor thermocouple akan mempengaruhi distribusi panas di dalam ruang pengering serta hasil pembacaan suhu selama sistem beroperasi.

Sebelum proses pemasangan dilakukan, terlebih dahulu ditentukan posisi *heater* dan sensor thermocouple di dalam ruang pengering. Penentuan posisi dilakukan dengan mempertimbangkan distribusi panas di dalam ruang pengering serta posisi pembacaan agar suhu yang terukur dapat merepresentasikan kondisi sebenarnya. Selain itu, penempatan kedua komponen juga disesuaikan dengan konstruksi oven sehingga proses pemasangan dapat dilakukan dengan mudah tanpa mengganggu fungsi ruang pengering.

Selanjutnya dilakukan pembuatan lubang pada bagian belakang *dehydrator* sebagai jalur pemasangan *heater* dan sensor thermocouple. Ukuran lubang disesuaikan dengan dimensi masing-masing komponen sehingga proses pemasangan dapat dilakukan dengan kuat dan presisi. Setelah proses tersebut selesai, *heater* tubular dipasang pada bagian bawah ruang pengering sebagai elemen pemanas, sedangkan sensor thermocouple dipasang pada bagian tengah ruang pengering sebagai pendeteksi suhu. Penempatan sensor pada bagian tengah ruang pengering bertujuan agar suhu yang terbaca di dalam ruang pengering lebih akurat serta mengurangi pengaruh radiasi panas secara langsung dari *heater*.



Gambar 4- 2 Proses Pemasangan Elemen Pemanas (*Heater*) pada *Dehydrator*



Gambar 4- 3 Proses Pemasangan Sensor Thermocouple pada *Dehydrator*

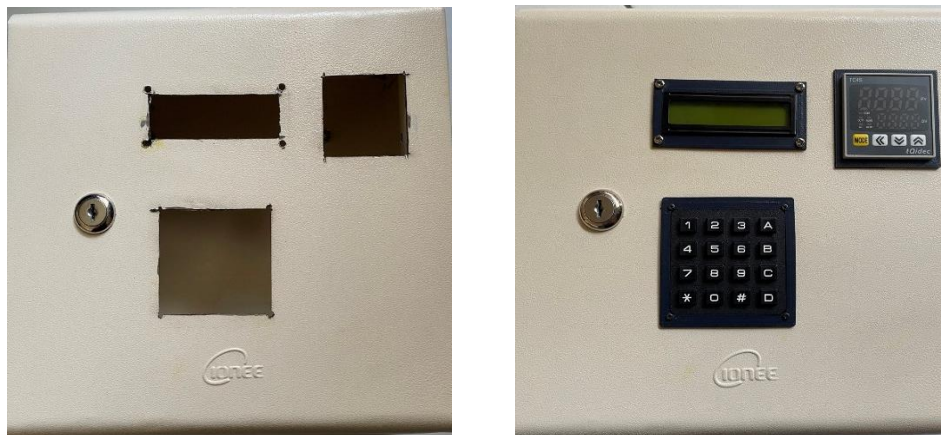
Setelah kedua komponen tersebut terpasang, dilakukan pemeriksaan terhadap posisi *heater* dan sensor thermocouple serta kekuatan pemasangannya untuk memastikan kedua komponen telah terpasang dengan baik dan aman sebelum dihubungkan dengan sistem kontrol. Hasil pemasangan *heater* dan sensor thermocouple pada *dehydrator* ditunjukkan pada Gambar 4- 2 dan Gambar 4- 3 diatas.

4.2.3 Perakitan Komponen pada Box Panel

Tahap selanjutnya yaitu proses perakitan seluruh komponen elektronik pada box panel sebagai pusat pengendali sistem *dehydrator*. Box panel dirancang untuk menempatkan seluruh komponen kontrol dalam satu tempat sehingga memudahkan proses perakitan maupun pengoperasian. Penataan posisi setiap komponen

disesuaikan dengan rancangan yang telah dibuat agar proses pemasangan dan penyambungan antar komponen dapat dilakukan dengan lebih mudah serta ruang di dalam box panel dapat dimanfaatkan secara optimal.

Proses perakitan diawali dengan pembuatan lubang pada tutup box panel sebagai tempat pemasangan *Temperature Controller* TC4S, LCD I2C, dan keypad. Pembuatan lubang dilakukan menggunakan gerinda tangan dengan ukuran lubang yang disesuaikan terhadap dimensi masing-masing komponen sehingga komponen dapat terpasang dengan presisi. Setelah proses pemotongan selesai, dilakukan perapian pada sisi lubang untuk menghilangkan bagian yang tajam. Selanjutnya dipasang bracket yang dirancang menggunakan *software* AutoCAD dan dicetak menggunakan printer 3D sebagaiudukan sekaligus penutup bekas pemotongan sehingga tampilan box panel menjadi lebih rapi serta memastikan setiap komponen dapat dipasang dengan baik. Hasil pemasangan komponen pada tutup box panel ditunjukkan pada Gambar 4- 4.



Gambar 4- 4 Hasil Pemasangan Komponen pada Tutup Box Panel

Selanjutnya dilakukan proses pengeboran pada plat mounting yang berada di bagian dalam box panel. Lubang dibuat sesuai dengan posisi baut pada masing-masing komponen sehingga pemasangan dapat dilakukan dengan kuat dan rapi. Setelah proses pengeboran selesai, dipasang berbagai komponen yang terdiri atas adaptor *power supply*, ESP32, SSR, relay, modul MAX6675, terminal block, dan MCB pada posisi yang telah ditentukan. ESP32 dipasang pada PCB untuk mempermudah proses perakitan dan penyambungan rangkaian. Penempatan setiap

komponen mempertimbangkan ukuran fisik serta kemudahan penyambungan antar komponen.

Setelah seluruh komponen terpasang, dilakukan penyambungan rangkaian sesuai dengan diagram wiring yang telah dirancang. Tahap ini meliputi penyambungan sumber catu daya AC dan DC serta koneksi antar komponen sehingga seluruh perangkat dapat bekerja sebagai satu kesatuan sistem. Setiap sambungan dibuat dengan memperhatikan kesesuaian terminal, polaritas, dan spesifikasi kabel yang digunakan agar sistem dapat beroperasi dengan aman dan andal. Seluruh kabel kemudian dirapikan dan disusun di dalam box panel untuk memudahkan proses pemeriksaan, serta mengurangi kemungkinan terjadinya kesalahan sambungan.

Tahap akhir dari proses perakitan adalah melakukan pemeriksaan terhadap seluruh sambungan dan fungsi setiap komponen. Pemeriksaan dilakukan untuk memastikan tidak terdapat kesalahan pada rangkaian, hubung singkat, maupun polaritas yang terbalik sebelum sistem diberi sumber tegangan. Setelah proses pemeriksaan selesai, dilakukan pengujian awal dengan menyalakan sistem untuk memastikan seluruh komponen dapat berfungsi dengan baik, *Temperature Controller* TC4S dapat beroperasi sesuai fungsinya, serta LCD I2C mampu menampilkan informasi sesuai dengan program yang telah dibuat. Hasil akhir perakitan komponen pada box panel ditunjukkan pada Gambar 4- 5.



Gambar 4- 5 Hasil Akhir Perakitan Komponen pada Box Panel

4.3 Perancangan Perangkat Lunak (*Software*)

Perangkat lunak (*software*) merupakan bagian yang berperan dalam mendukung proses monitoring pada *dehydrator* listrik. Program dikembangkan menggunakan Arduino IDE dengan bahasa pemrograman Arduino dan diunggah pada mikrokontroler ESP32 sebagai pusat pengolahan data. Melalui perangkat lunak ini, ESP32 dapat berkomunikasi dengan seluruh komponen yang digunakan sehingga proses pembacaan data, pengolahan informasi, penampilan data, dan pengiriman hasil monitoring dapat berlangsung secara terintegrasi.

Perangkat lunak yang dikembangkan berfungsi untuk membaca data suhu dari sensor thermocouple melalui modul MAX6675, menerima masukan waktu pengeringan melalui keypad, menampilkan informasi suhu dan waktu pada LCD I2C 16×2, serta mengirimkan data hasil monitoring ke Google Spreadsheet dan Telegram Bot melalui jaringan Wi-Fi. Penyusunan program dilakukan dengan mengatur alur kerja sistem mulai dari proses inisialisasi perangkat, pembacaan data sensor, pengolahan data, hingga komunikasi data dengan media monitoring berbasis *Internet of Things* (IoT).

Selain mendukung proses monitoring, perangkat lunak juga dirancang untuk memungkinkan pemantauan alat dari jarak jauh. Data hasil pembacaan suhu, status sistem, dan sisa waktu pengeringan dikirimkan secara berkala ke Google Spreadsheet sebagai media pencatatan data, sedangkan Telegram Bot digunakan untuk memberikan informasi mengenai kondisi alat selama proses pengeringan berlangsung. Dengan demikian, pengguna dapat memantau kinerja *dehydrator* secara *real-time* tanpa harus berada di dekat alat.

4.3.1 Pembuatan Program ESP32 pada Arduino IDE

Setelah perancangan perangkat keras selesai dilakukan, tahap berikutnya adalah membuat program pada mikrokontroler ESP32 menggunakan Arduino IDE. Program ditulis menggunakan bahasa pemrograman arduino dalam satu berkas utama atau *sketch* yang memuat fungsi monitoring suhu, pengaturan waktu pengeringan, pengendalian relay, tampilan LCD, serta pengiriman data melalui Telegram dan Google Spreadsheet.

Program menggunakan sistem operasi waktu nyata FreeRTOS untuk membagi proses menjadi dua *task* yang dijalankan pada inti ESP32 yang berbeda. Proses pembacaan sensor, pembacaan *keypad*, perhitungan waktu, pembaruan LCD, dan pengendalian relay dijalankan melalui *TaskControl*. Sementara itu, proses pengiriman data melalui Telegram dan Google Spreadsheet dijalankan melalui *TaskInternet*. Pembagian tersebut dilakukan agar komunikasi internet tidak menghambat proses kontrol dan monitoring selama pengeringan berlangsung.

Pembuatan program dilakukan secara bertahap, dimulai dari penambahan *library*, konfigurasi pin ESP32, serta deklarasi variabel dan parameter program. Penyusunan program secara terstruktur bertujuan agar kode lebih mudah dipahami, diuji, dan dikembangkan.

a. Penambahan *Library* pada Program

Tahap pertama dalam pembuatan program adalah menambahkan *library* yang diperlukan agar ESP32 dapat menjalankan seluruh fungsi sistem. *Library* *WiFi.h*, *HTTPClient.h*, dan *WiFiClientSecure.h* digunakan untuk menghubungkan ESP32 ke jaringan Wi-Fi serta melakukan komunikasi menggunakan protokol HTTP dan HTTPS. *Library* *UniversalTelegramBot.h* dan *ArduinoJson.h* digunakan untuk mendukung proses pengiriman pesan melalui Telegram Bot. Komunikasi dengan LCD I2C menggunakan *library* *Wire.h* dan *LiquidCrystal_I2C.h*, sedangkan pembacaan masukan pengguna melalui *keypad* menggunakan *Keypad.h*. Pembacaan suhu dari sensor *thermocouple* melalui modul MAX6675 menggunakan *max6675.h*. Selanjutnya, *library* *freertos/FreeRTOS.h* dan *freertos/task.h* digunakan untuk membuat serta menjalankan dua *task* pada inti ESP32 yang berbeda.

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
#include "max6675.h"
#include <freertos/FreeRTOS.h>
```

```
#include <freertos/task.h>
```

Gambar 4- 6 Penambahan *Library* pada Program

Gambar 4- 6 menunjukkan seluruh *library* yang digunakan untuk mendukung komunikasi Wi-Fi, pengiriman data, pembacaan sensor, penggunaan LCD dan *keypad*, serta pembagian proses menggunakan FreeRTOS.

b. Konfigurasi Pin ESP32

Tahap berikutnya adalah melakukan konfigurasi pin ESP32 yang terhubung dengan setiap komponen. Konfigurasi pin bertujuan untuk mendefinisikan fungsi masing-masing pin sehingga komunikasi antara mikrokontroler dan perangkat keras dapat berjalan sesuai perancangan. LCD menggunakan komunikasi I2C dengan alamat 0x27, sedangkan jalur SDA dan SCL masing-masing menggunakan GPIO21 dan GPIO22 yang diinisialisasi pada fungsi `setup()`. Modul MAX6675 berkomunikasi dengan ESP32 melalui pin SCK, CS, dan SO. Relay dikendalikan melalui GPIO23 sebagai keluaran digital untuk menghubungkan atau memutuskan jalur pemanas. Sementara itu, *keypad* matriks 4×4 menggunakan empat pin baris dan empat pin kolom sebagai jalur pembacaan tombol.

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
#define MAX6675_SO 19
#define MAX6675_CS 5
#define MAX6675_SCK 18
```

```
MAX6675 thermocouple(
    MAX6675_SCK,
    MAX6675_CS,
    MAX6675_SO
);
```

```
#define RELAY_PIN 23
#define RELAY_ON HIGH
#define RELAY_OFF LOW
```

```
const byte ROWS = 4;
const byte COLS = 4;
```

```
char keys[ROWS][COLS] =
```

```

{
  {'1','2','3','*'},
  {'4','5','6','0'},
  {'7','8','9','#'},
  {'A','B','C','D'}
};

byte rowPins[ROWS] =
{
  13, 12, 14, 27
};

byte colPins[COLS] =
{
  26, 25, 33, 32
};

Keypad keypad = Keypad(
  makeKeymap(keys),
  rowPins,
  colPins,
  ROWS,
  COLS
);

```

Gambar 4- 7 Konfigurasi Pin ESP32

Gambar 4- 7 menunjukkan konfigurasi pin ESP32 yang digunakan untuk menghubungkan LCD I2C, modul MAX6675, relay, dan *keypad*. Konfigurasi tersebut disesuaikan dengan wiring sistem agar setiap komponen dapat berkomunikasi dengan ESP32 sesuai fungsi masing-masing.

c. Deklarasi Variabel dan Parameter Program

Tahap selanjutnya adalah mendeklarasikan variabel dan parameter yang digunakan selama program berjalan. Variabel tersebut digunakan untuk menyimpan status proses, hasil pembacaan suhu, masukan waktu, perhitungan *timer*, kondisi koneksi Wi-Fi, serta permintaan pengiriman data melalui internet. Beberapa variabel dideklarasikan menggunakan kata kunci *volatile* karena nilainya digunakan oleh *TaskControl* dan *TaskInternet* yang berjalan pada inti berbeda. Variabel *requestTelegram* dan *requestSpreadsheet* digunakan sebagai penanda adanya permintaan pengiriman data. Pesan Telegram disimpan

sementara pada variabel telegramMessage sebelum diproses oleh *TaskInternet*. Program juga mendeklarasikan ControlTask dan InternetTask sebagai *handle* untuk kedua *task* FreeRTOS. Interval pengiriman Telegram dan Google Spreadsheet ditetapkan sebesar 60.000 milidetik atau 60 detik. Pengolahan data suhu menggunakan filter IIR dengan nilai koefisien alpha sebesar 0,3 untuk mengurangi perubahan nilai suhu yang terlalu cepat.

```
volatile bool requestTelegram = false;
volatile bool requestSpreadsheet = false;
volatile bool prosesRunning = false;
volatile bool wifiConnected = false;
volatile float suhu = 0;

String telegramMessage = "";
String inputMenit = "";

unsigned long waktuMulai = 0;
unsigned long durasiDetik = 0;
unsigned long elapsed = 0;
long sisaDetik = 0;

bool stopConfirmation = false;
bool tunggu3Detik = false;

const unsigned long intervalReadTemp = 1000;
const unsigned long lcdInterval = 300;
const unsigned long telegramInterval = 60000;
const unsigned long spreadsheetInterval = 60000;
const unsigned long debounceTime = 200;

float suhuFilter = 0.0;
bool firstRead = true;
const float alpha = 0.3;

TaskHandle_t ControlTask;
TaskHandle_t InternetTask;
```

Gambar 4- 8 Deklarasi Variabel Program dan Parameter Program

Gambar 4- 8 menunjukkan variabel dan parameter utama yang digunakan dalam sistem. Variabel status digunakan untuk menentukan kondisi kerja alat, variabel waktu digunakan untuk menjalankan proses *countdown*, sedangkan variabel permintaan internet digunakan sebagai penghubung antara proses

kontrol dan proses komunikasi. Penjelasan mengenai mekanisme kerja setiap fungsi program dibahas pada subbab 4.3.2.

4.3.2 Cara Kerja Program

Pada subbab ini dijelaskan cara kerja program yang digunakan pada sistem monitoring *dehydrator* listrik berbasis ESP32. Program disusun ke dalam beberapa fungsi yang saling terintegrasi sehingga setiap fungsi memiliki tugas yang berbeda dalam menjalankan sistem. Pembagian program ke dalam beberapa fungsi bertujuan untuk mempermudah penyusunan kode, meningkatkan keterbacaan program, serta memudahkan proses pengujian dan perbaikan apabila terjadi kesalahan pada sistem.

Secara umum, program bekerja dengan melakukan inisialisasi seluruh perangkat keras saat ESP32 dinyalakan, kemudian menerima masukan waktu pengeringan melalui keypad. Selama proses pengeringan berlangsung, ESP32 membaca data suhu dari sensor thermocouple melalui modul MAX6675, menampilkan informasi pada LCD I2C, mengendalikan relay berdasarkan waktu yang telah ditentukan, serta mengirimkan data monitoring ke Telegram Bot dan Google Spreadsheet melalui jaringan Wi-Fi. Penjelasan mengenai fungsi dan mekanisme kerja setiap bagian program akan diuraikan pada pembahasan berikut sesuai dengan fungsi yang digunakan dalam program.

a. Fungsi Inisialisasi Sistem `setup()`

```
pinMode(RELAY_PIN, OUTPUT);
digitalWrite(RELAY_PIN, RELAY_OFF);

Wire.begin(21, 22);
lcd.init();
lcd.backlight();

xTaskCreatePinnedToCore(
    TaskControl, "Control", 8192,
    NULL, 2, &ControlTask, 1
);

xTaskCreatePinnedToCore(
    TaskInternet, "Internet", 8192,
```

```

    NULL, 1, &InternetTask, 0
);

```

Gambar 4- 9 Fungsi setup()

Gambar 4- 9 menunjukkan fungsi setup() yang dijalankan satu kali ketika ESP32 pertama kali diaktifkan. Fungsi ini digunakan untuk mengatur komunikasi serial, komunikasi I2C, LCD, relay, koneksi Wi-Fi, serta task FreeRTOS yang digunakan dalam program. Komunikasi serial diaktifkan menggunakan Serial.begin(115200) untuk menampilkan informasi proses pada Serial Monitor. Komunikasi I2C diinisialisasi menggunakan Wire.begin(21,22), dengan GPIO21 sebagai jalur SDA dan GPIO22 sebagai jalur SCL. LCD kemudian diaktifkan menggunakan lcd.init() dan lcd.backlight().

Pada tahap awal, LCD menampilkan tulisan SMART DRYER dan BOOTING... sebagai indikator proses inisialisasi. Pin relay diatur sebagai keluaran menggunakan pinMode(RELAY_PIN, OUTPUT). Relay kemudian diatur dalam kondisi OFF menggunakan digitalWrite(RELAY_PIN, RELAY_OFF) sehingga *heater* tidak langsung bekerja ketika alat dinyalakan. Selanjutnya, ESP32 mencoba terhubung ke jaringan Wi-Fi selama maksimal 10 detik. Apabila koneksi berhasil, variabel wifiConnected bernilai true, komunikasi aman untuk Telegram diaktifkan, dan sistem mengirimkan notifikasi bahwa perangkat telah terhubung ke jaringan. Apabila koneksi gagal, sistem tetap dapat bekerja dalam mode *offline*, tetapi pengiriman data ke Telegram dan Google Spreadsheet tidak dilakukan.

Setelah proses inisialisasi selesai, LCD menampilkan menu SET TIME(MIN) dan tanda > sebagai indikator bahwa sistem siap menerima masukan waktu pengeringan. Pada tahap akhir fungsi setup(), program membuat dua task FreeRTOS, yaitu TaskControl dan TaskInternet.

b. Fungsi Program Utama loop() dan Pembagian Task FreeRTOS

```

void loop()
{
    vTasDerivative Timeelay(portMAX_DELAY);
}

void TaskControl(void *pv)

```

```

{
    while (true)
    {
        if (prosesRunning)
        {
            prosesPengeringan();
        }
        else
        {
            menuInputWaktu();
        }

        vTasDerivative Timeelay(pdMS_TO_TICKS(20));
    }
}

void TaskInternet(void *pv)
{
    while (true)
    {
        prosesInternet();

        vTasDerivative Timeelay(pdMS_TO_TICKS(100));
    }
}

```

Gambar 4- 10 Fungsi loop(), TaskControl() dan TaskInternet()

Gambar 4- 10 menunjukkan fungsi loop() pada program hanya digunakan untuk menahan proses utama dengan perintah vTasDerivative Timeelay(portMAX_DELAY). Pengendalian sistem tidak dilakukan secara langsung melalui fungsi loop(), tetapi dialihkan ke dua task FreeRTOS. Task TaskControl dijalankan pada core 1 ESP32 dengan prioritas lebih tinggi. Task ini bertugas menjalankan fungsi kontrol utama, yaitu membaca *keypad*, menjalankan menu input waktu, membaca suhu, menghitung waktu pengeringan, memperbarui LCD, serta mengendalikan relay. Task TaskInternet dijalankan pada core 0 ESP32. Task ini bertugas menjalankan fungsi prosesInternet() untuk mengirimkan pesan Telegram dan data ke Google Spreadsheet. Task kontrol dijalankan setiap 20 milidetik, sedangkan task internet dijalankan setiap 100 milidetik. Pembagian tugas tersebut membuat proses

komunikasi internet berjalan terpisah dari proses kontrol sehingga pembacaan sensor dan pengendalian alat dapat tetap berjalan.

c. Fungsi Pembacaan dan Penyaringan Suhu `bacaSuhuIIR()`

```
float bacaSuhuIIR()
{
    float suhuRaw = thermocouple.readCelsius();

    if (firstRead)
    {
        suhuFilter = suhuRaw;
        firstRead = false;
    }

    suhuFilter =
        (alpha * suhuRaw) +
        ((1.0 - alpha) * suhuFilter);

    return suhuFilter;
}
```

Gambar 4- 11 Fungsi `bacaSuhuIIR()`

Gambar 4- 11 menunjukkan fungsi `bacaSuhuIIR()` yang digunakan untuk membaca suhu dari sensor *thermocouple* melalui modul MAX6675. Nilai suhu mentah diperoleh menggunakan perintah `thermocouple.readCelsius()`. Pada pembacaan pertama, nilai suhu digunakan sebagai nilai awal filter. Pembacaan berikutnya diolah menggunakan filter IIR dengan koefisien α sebesar 0,3. Nilai suhu baru dikombinasikan dengan nilai suhu hasil penyaringan sebelumnya. Proses tersebut bertujuan untuk mengurangi perubahan nilai yang terlalu cepat sehingga data suhu yang ditampilkan pada LCD dan dikirimkan melalui internet menjadi lebih stabil. Pembacaan suhu dilakukan secara berkala setiap satu detik sesuai dengan nilai `intervalReadTemp = 1000`. Nilai hasil penyaringan disimpan pada variabel suhu dan digunakan untuk tampilan LCD, notifikasi Telegram, serta pencatatan pada Google Spreadsheet.

d. Fungsi Input Waktu Pengeringan (menuInputWaktu())

```

if (key >= '0' && key <= '9')
    inputMenit += key;

if (key == '*')
    inputMenit = "";

if (key == '#' && inputMenit.length() > 0) {
    durasiDetik = inputMenit.toInt() * 60UL;
    waktuMulai = millis();
    digitalWrite(RELAY_PIN, RELAY_ON);
    prosesRunning = true;
}

```

Gambar 4- 12 Fungsi menuInputWaktu()

Gambar 4- 12 menunjukkan fungsi menuInputWaktu() yang digunakan untuk menerima masukan durasi pengeringan dari *keypad* matriks 4×4. Program membaca tombol menggunakan perintah keypad.getKey(). Untuk menghindari pembacaan tombol secara berulang akibat *bouncing*, digunakan *debounce* dengan waktu 200 milidetik. Tombol angka 0 sampai 9 digunakan untuk memasukkan waktu pengeringan dalam satuan menit dengan batas maksimal tiga digit. Tombol * digunakan untuk menghapus seluruh masukan. Tombol # digunakan untuk memulai proses apabila nilai waktu telah tersedia. Ketika tombol # ditekan, nilai waktu dikonversi dari menit menjadi detik menggunakan inputMenit.toInt()*60UL. Program kemudian menyimpan waktu mulai pada variabel waktuMulai, mengatur nilai awal sisaDetik, mengaktifkan relay, dan mengubah nilai prosesRunning menjadi true. Perubahan tersebut menyebabkan TaskControl berpindah dari fungsi menuInputWaktu() ke fungsi prosesPengeringan(). Apabila Wi-Fi tersedia, program menyusun pesan mengenai dimulainya proses pengeringan. Pesan tersebut disimpan pada variabel telegramMessage, kemudian program mengaktifkan requestTelegram agar pesan dapat dikirim oleh TaskInternet.

e. Fungsi Proses Pengeringan (prosesPengeringan())

```

if (millis() - lastReadTemp >= intervalReadTemp) {
    suhu = bacaSuhuIIR();
}

```

```

    lastReadTemp = millis();
}

elapsed = (millis() - waktuMulai) / 1000;
sisadetik = durasiDetik - elapsed;

if (sisadetik <= 0) {
    digitalWrite(RELAY_PIN, RELAY_OFF);
    prosesRunning = false;
    lcd.clear();
    lcd.print("PROCESS DONE");
}    lcd.print("REMOVE ITEM");
}

```

Gambar 4- 13 Fungsi prosesPengerinan()

Gambar 4- 13 menunjukkan fungsi prosesPengerinan() yang merupakan fungsi utama selama proses pengeringan berlangsung. Fungsi ini membaca suhu secara berkala, menghitung waktu yang telah berjalan, menentukan sisa waktu, memperbarui LCD, memeriksa tombol penghentian, serta mengatur relay. Perhitungan waktu dilakukan dengan membandingkan waktu saat ini dengan waktu mulai menggunakan fungsi millis(). Hasil perhitungan digunakan untuk menentukan nilai elapsed dan sisadetik. Nilai sisa waktu kemudian dikonversi menjadi jam, menit, dan detik untuk ditampilkan pada LCD. Selama proses berjalan, LCD menampilkan suhu dalam format T:xx.x C pada baris pertama dan sisa waktu dalam format HH:MM:SS pada baris kedua. Pembaruan LCD dilakukan setiap 300 milidetik sesuai dengan nilai lcdInterval.

Fungsi ini juga memeriksa tombol C sebagai perintah penghentian manual. Apabila tombol C ditekan, sistem menampilkan konfirmasi STOP PROCESS? dengan pilihan #=YES dan *=NO. Tombol * digunakan untuk membatalkan penghentian dan melanjutkan proses, sedangkan tombol # digunakan untuk mematikan relay dan menghentikan proses pengeringan. Apabila nilai sisadetik telah mencapai nol, relay dimatikan dan nilai prosesRunning diubah menjadi false. LCD menampilkan PROCESS DONE dan REMOVE ITEM sebagai tanda bahwa proses pengeringan telah selesai. Program juga menyusun notifikasi penyelesaian apabila koneksi Wi-Fi tersedia.

f. Fungsi Pengolahan Internet prosesInternet()

```
void prosesInternet() {
  if (!wifiConnected) return;

  if (requestTelegram) {
    requestTelegram = false;
    bot.sendMessage(CHAT_ID, telegramMessage, "");
  }

  if (requestSpreadsheet) {
    requestSpreadsheet = false;
    kirimSpreadsheet(suhu);
  }
}
```

Gambar 4- 14 Perintah Pengiriman Data Internet

Gambar 4- 14 menunjukkan Fungsi prosesInternet() dijalankan oleh TaskInternet. Fungsi ini memeriksa nilai requestTelegram dan requestSpreadsheet untuk menentukan jenis data yang harus dikirimkan. Apabila requestTelegram bernilai true, program mengirimkan isi telegramMessage melalui perintah bot.sendMessage(). Setelah proses pengiriman dilakukan, nilai requestTelegram dikembalikan menjadi false. Apabila requestSpreadsheet bernilai true, program memanggil fungsi kirimSpreadsheet() untuk mengirimkan data monitoring melalui HTTP GET. Pemisahan proses internet ke dalam TaskInternet membuat pengiriman data tidak dijalankan langsung di dalam fungsi kontrol. Dengan demikian, proses komunikasi jaringan dapat berlangsung tanpa menghentikan pembacaan suhu, perhitungan waktu, pembaruan LCD, dan pengendalian relay.

g. Fungsi Pengiriman Notifikasi Telegram

```
if (wifiConnected && requestTelegram) {
  bot.sendMessage(CHAT_ID, telegramMessage, "");
  requestTelegram = false;
}
```

Gambar 4- 15 Fungsi kirimSpreadsheet()

Gambar 4- 15 menunjukkan fungsi pengiriman notifikasi ke telegram. Notifikasi Telegram hanya dikirimkan apabila nilai wifiConnected bernilai true. Pengiriman data monitoring dilakukan setiap 60 detik sesuai dengan nilai

telegramInterval. Informasi yang dikirimkan meliputi suhu aktual, status relay atau *heater*, sisa waktu pengeringan, dan status koneksi Wi-Fi. Selain pengiriman berkala, Telegram juga digunakan untuk mengirimkan informasi ketika sistem berhasil terhubung ke Wi-Fi, proses pengeringan dimulai, proses dihentikan secara manual, dan proses pengeringan selesai. Pesan tidak dikirimkan secara langsung oleh fungsi kontrol. Fungsi kontrol terlebih dahulu menyusun isi pesan dan mengaktifkan requestTelegram, kemudian proses pengiriman dilakukan oleh TaskInternet.

h. Fungsi Pengiriman Data ke Google Spreadsheet

```
void kirimSpreadsheet(float suhu) {
    if (!wifiConnected) return;

    String url = GOOGLE_SCRIPT_URL;
    url += "?temperature=" + String(suhu, 1);
    url += "&heater=" +
        String(digitalRead(RELAY_PIN) == RELAY_ON ? "ON" : "OFF");

    HTTPClient http;
    http.begin(url);
    http.GET();
    http.end();
}
```

Gambar 4- 16 Fungsi kirimSpreadsheet()

Gambar 4- 16 yang menunjukkan fungsi kirimSpreadsheet() digunakan untuk mengirimkan data monitoring ke Google Spreadsheet melalui alamat Web App Google Apps Script. Data dikirim menggunakan metode HTTP GET melalui objek HTTPClient. Parameter yang dikirimkan meliputi suhu, status *heater*, waktu pengeringan, dan status proses. Pengiriman data hanya dilakukan ketika ESP32 terhubung ke Wi-Fi. Data dikirimkan secara berkala setiap 60 detik sesuai nilai spreadsheetInterval. Setelah permintaan HTTP dikirimkan, kode respons dari server ditampilkan pada Serial Monitor. Setelah proses selesai, koneksi HTTP ditutup menggunakan http.end(). Data yang tersimpan pada Google Spreadsheet dapat digunakan sebagai riwayat monitoring dan bahan analisis proses pengeringan.

4.3.3 Pembuatan Google Spreadsheet

Google Spreadsheet digunakan sebagai media penyimpanan data hasil monitoring yang dikirimkan oleh ESP32 melalui jaringan internet. Agar data yang dikirimkan dapat diterima dan disimpan secara otomatis, diperlukan konfigurasi Google Apps Script sebagai penghubung antara ESP32 dan Google Spreadsheet. Tahapan pembuatan Google Spreadsheet pada penelitian ini adalah sebagai berikut.

1. Membuat lembar kerja (Google Spreadsheet) baru sebagai media penyimpanan data hasil monitoring.
2. Menambahkan nama kolom (*header*) yang terdiri atas *Date*, *Time*, *Temperature* (°C), *Heater*, *Timer*, dan Status sesuai dengan parameter yang akan direkam.
3. Membuka menu Extensions → Apps Script untuk membuat Google Apps Script yang berfungsi sebagai penghubung antara ESP32 dan Google Spreadsheet.
4. Menuliskan program Google Apps Script untuk menerima data yang dikirimkan oleh ESP32 dan menyimpannya ke dalam Google Spreadsheet.
5. Menyimpan program Google Apps Script yang telah dibuat.
6. Melakukan proses Deploy dengan memilih jenis Web App sehingga Google Apps Script dapat diakses melalui jaringan internet.
7. Mengatur hak akses (*Who has access*) menjadi *Anyone* atau *Anyone with the link* agar layanan dapat diakses oleh ESP32 tanpa memerlukan autentikasi tambahan.
8. Menyalin alamat Web App URL yang dihasilkan setelah proses deployment selesai.
9. Mengonfigurasi alamat Web App URL tersebut ke dalam program ESP32 melalui variabel `GOOGLE_SCRIPT_URL` sebagai alamat tujuan pengiriman data monitoring menuju Google Apps Script.
10. Mengunggah program ke ESP32 dan melakukan pengujian untuk memastikan data monitoring berhasil dikirim serta tersimpan pada Google Spreadsheet.

```
//=====
// GOOGLE SPREADSHEET
//=====
```

```
String GOOGLE_SCRIPT_URL =
"https://script.google.com/macros/s/AKfycbx0_9EpD4_ahZX1qwUMCxQP0os_EfVJ
P6_GgTk-3OdXw-OZk3tix182GixinSlejiZ8GQ/exec";
```

Gambar 4- 17 Inisialisasi URL Google Apps Script

Gambar 4- 17 menunjukkan proses inisialisasi alamat Web App URL yang diperoleh dari Google Apps Script. Alamat tersebut disimpan pada variabel `GOOGLE_SCRIPT_URL` sebagai tujuan pengiriman data monitoring dari ESP32. Selanjutnya, variabel ini akan digunakan oleh fungsi `kirimSpreadsheet()` untuk mengirimkan parameter suhu, status *heater*, sisa waktu, dan status proses ke Google Spreadsheet melalui metode HTTP GET.

4.3.4 Pembuatan Telegram Bot

Telegram Bot digunakan sebagai media monitoring jarak jauh yang berfungsi untuk mengirimkan informasi kondisi dehydrator listrik secara otomatis kepada pengguna. Notifikasi yang dikirim meliputi status koneksi sistem, dimulainya proses pengeringan, kondisi proses selama pengeringan berlangsung, proses dihentikan, serta proses pengeringan selesai. Agar ESP32 dapat mengirimkan pesan ke aplikasi Telegram, diperlukan pembuatan bot menggunakan layanan BotFather serta konfigurasi Bot Token dan Chat ID pada program ESP32. Tahapan pembuatan Telegram Bot pada penelitian ini adalah sebagai berikut.

1. Menginstal aplikasi Telegram pada perangkat smartphone atau komputer, kemudian membuat akun Telegram apabila belum memiliki akun.
2. Mencari akun `@BotFather` melalui kolom pencarian pada aplikasi Telegram.
3. Memulai percakapan dengan BotFather menggunakan perintah `/start`.
4. Membuat bot baru dengan mengirimkan perintah `/newbot`, kemudian memberikan nama bot (*Bot Name*) dan nama pengguna (*Username*) sesuai ketentuan Telegram.
5. Menyalin Bot Token yang diberikan oleh BotFather sebagai identitas bot yang akan digunakan oleh ESP32 untuk mengakses Telegram Bot API.

telah dibuat sehingga membentuk sebuah prototipe *dehydrator* listrik. Hasil realisasi alat ditunjukkan pada Gambar 4- 19.



Gambar 4- 19 Hasil Akhir Alat *Dehydrator* Listrik

Prototipe yang telah dirancang terdiri atas ruang pengering yang dilengkapi elemen pemanas (*heater*) serta panel kontrol yang ditempatkan di bagian luar ruang pengering. Panel kontrol berisi mikrokontroler ESP32, modul MAX6675, relay, LCD I2C, dan keypad 4×4 yang digunakan sebagai media pengaturan waktu pengeringan dan monitoring sistem. Selain itu, sistem juga dilengkapi dengan monitoring berbasis *Internet of Things* (IoT). ESP32 dirancang untuk membaca data suhu dari modul MAX6675, menampilkan informasi pada LCD, serta mengirimkan data monitoring ke Google Spreadsheet dan Telegram Bot melalui jaringan Wi-Fi.

Susunan komponen dan hubungan antar perangkat telah disesuaikan dengan hasil perancangan perangkat keras dan perangkat lunak yang telah dijelaskan pada subbab sebelumnya. Prototipe yang telah dirancang selanjutnya digunakan sebagai objek pada tahap pengujian untuk mengetahui kesesuaian fungsi setiap komponen serta kinerja sistem secara keseluruhan. Hasil pengujian tersebut akan dibahas pada bab berikutnya.