

BAB IV

PEMBUATAN ALAT

4.1. Pembuatan *prototype*

Pada tahap ini dilakukan pembuatan prototipe sebagai bentuk implementasi dari hasil perancangan sistem yang telah disusun pada tahap sebelumnya. Pembuatan prototipe dilakukan secara terstruktur untuk memastikan bahwa seluruh komponen dapat terintegrasi dan bekerja sesuai dengan fungsi yang direncanakan. Tahap ini mencakup proses perakitan, konfigurasi, dan pengujian awal terhadap setiap komponen yang digunakan sehingga sistem dapat beroperasi secara optimal sesuai dengan tujuan penelitian.

Proses pembuatan prototipe melibatkan dua tahapan utama, yaitu pembuatan perangkat keras (*hardware*) dan pembuatan perangkat lunak (*software*). Pembuatan perangkat keras dilakukan dengan merakit seluruh komponen sesuai dengan desain sistem yang telah dirancang. Tahap ini bertujuan untuk membangun bentuk fisik sistem sehingga seluruh komponen dapat saling terhubung dan berfungsi sebagai satu kesatuan.

Selanjutnya dilakukan pengembangan perangkat lunak yang berfungsi sebagai pengendali utama sistem. Pada tahap ini, program dirancang dan diimplementasikan untuk mengatur proses kerja sistem, mengolah data yang diperoleh dari sensor, serta mengendalikan keluaran sesuai dengan logika dan algoritma yang telah ditentukan. Selain itu, perangkat lunak juga dikembangkan untuk mendukung proses komunikasi data, pemantauan sistem, dan pengujian kinerja prototipe. Dengan integrasi antara perangkat keras dan perangkat lunak, prototipe yang dihasilkan diharapkan dapat merepresentasikan sistem yang dirancang serta menjadi dasar dalam pelaksanaan pengujian dan evaluasi pada tahap berikutnya.

4.2. Peralatan dan Bahan Sistem

Dalam pelaksanaan penelitian dan pengembangan sistem pada tugas akhir ini, diperlukan berbagai peralatan dan bahan yang berperan sebagai komponen utama maupun pendukung dalam proses perancangan, implementasi, dan pengujian sistem. Ketersediaan peralatan dan bahan yang sesuai sangat penting untuk memastikan seluruh tahapan penelitian dapat berjalan dengan baik serta menghasilkan sistem yang berfungsi sesuai dengan tujuan yang telah ditetapkan. Oleh karena itu, dilakukan identifikasi kebutuhan sebagai langkah awal untuk menentukan spesifikasi alat dan bahan yang diperlukan dalam pembangunan sistem. Identifikasi kebutuhan tersebut bertujuan untuk memastikan bahwa seluruh komponen yang digunakan mampu mendukung proses pengembangan sistem secara optimal. Selain itu, hasil identifikasi juga menjadi acuan dalam proses pengadaan, perakitan, konfigurasi, dan pengujian sistem sehingga dapat meminimalkan kendala yang mungkin terjadi selama pelaksanaan penelitian. Dengan adanya perencanaan kebutuhan yang baik, diharapkan sistem yang dikembangkan dapat bekerja secara terintegrasi, andal, dan sesuai dengan kebutuhan penelitian. Adapun peralatan dan bahan yang digunakan dalam penelitian ini disajikan pada tabel 4.1 dan 4.2 berikut.

Tabel 4.1 Bahan Penelitian

No.	Komponen	Jumlah
1.	ESP32 DevKit V1	1 buah
2.	<i>Module Sensor TCS34725</i>	1 buah
3.	<i>Driver Motor Stepper A4988</i>	1 buah
4.	<i>Motor Stepper Nema17</i>	1 buah
5.	Motor Servo MG996R	4 buah
6.	<i>Sensor Load Cell</i>	3 set
7.	<i>Power Supply (PSU) 12V 5A</i>	1 buah
8.	<i>DC Buck Converter</i>	1 buah
9.	<i>Emergency Stop Button</i>	1 buah

No.	Komponen	Jumlah
10.	MCB	1 buah
11.	<i>Push Button</i>	2 buah
12.	Box Panel 20x30	1 buah
13.	<i>Terminal Block</i>	1 buah
14.	<i>Conveyor Belt</i> PVC Hijau	2 meter
15.	Lembaran <i>Acrylic</i> 1,5mm	7 lembar
16.	Fiber PCB Plat 10x20cm	1 buah
17.	FeCL ₃ 100g	2 bungkus
17.	<i>Bearing Shaft</i> 10mm	4 buah
18.	<i>Timing Belt and Pulley</i> 10mm	1 set
19.	<i>Gravity Roller Conveyor</i> 50mmx400mm	2 buah
20.	<i>Alumunium Profile</i> 2020 series	2 meter
21.	<i>Alumunium Profile</i> 2040 series	4 meter

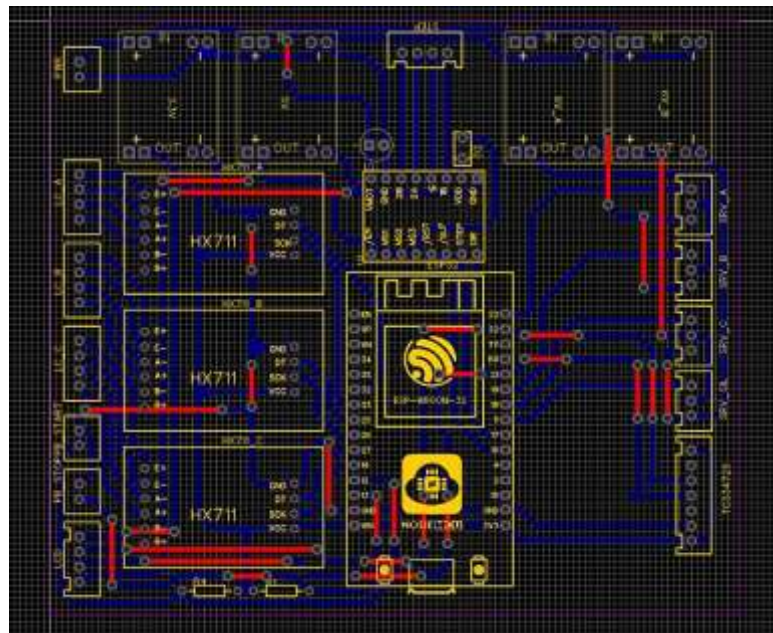
Tabel 4.2 Peralatan Penelitian

No.	Peralatan	Fungsi/Penggunaan
1.	Laptop	Digunakan untuk pemrograman ESP32, pengolahan data, perancangan sistem, serta pembuatan <i>website</i> monitoring dan penyusunan laporan penelitian.
2.	<i>Smartphone</i>	Digunakan untuk mengakses halaman monitoring berbasis <i>web</i> , memantau hasil klasifikasi tingkat kematangan buah kopi, melihat data berat secara <i>real-time</i> , serta menguji komunikasi data antara ESP32 dan <i>web server</i> melalui jaringan Wi-Fi
4.	Multimeter Digital	Digunakan untuk mengukur tegangan, arus, hambatan, serta memeriksa kontinuitas rangkaian elektronik.

No.	Peralatan	Fungsi/Penggunaan
5.	Perkakas	Digunakan untuk membantu proses pemasangan, pelepasan, dan pengencangan komponen mekanik pada sistem <i>conveyor</i> .

4.3. Perancangan dan Pembuatan PCB

Perancangan PCB dilakukan untuk mengintegrasikan hubungan antar-komponen elektronik pada sistem *conveyor* penyortir buah kopi agar rangkaian menjadi lebih rapi, stabil, dan mudah dalam proses pemasangan serta pemeriksaan. PCB digunakan sebagai media penghubung antara mikrokontroler, sensor, *driver* motor, dan komponen pendukung lainnya. Perancangan *layout* PCB dibuat berdasarkan rangkaian dan hubungan pin antar-komponen yang telah ditentukan pada tahap perancangan perangkat keras.

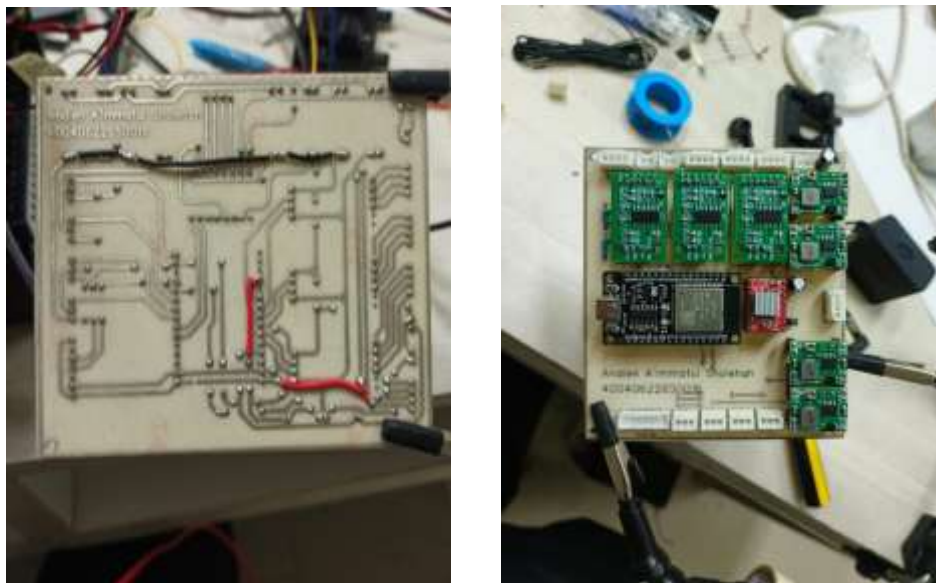


Gambar 4.1 *Layout Design* PCB pada EasyEDA

Perancangan *Layout Design* PCB pada Gambar 4.1 dilakukan dengan memperhatikan penempatan setiap komponen dan jalur koneksi antar-komponen. ESP32 ditempatkan pada bagian tengah PCB sebagai pusat pengendalian sistem. Posisi tersebut dipilih agar koneksi menuju komponen *input* dan *output* dapat

disusun dengan lebih terorganisir. Beberapa modul HX711 ditempatkan pada bagian kiri PCB yang digunakan sebagai penguat dan pengubah sinyal dari sensor *load cell* sebelum data diproses oleh ESP32. Penempatan beberapa modul HX711 dilakukan agar masing-masing *load cell* dapat terhubung dan diproses secara terpisah oleh sistem.

Pada bagian atas PCB terdapat jalur dan konektor yang digunakan untuk *driver motor stepper*, sedangkan bagian sisi kanan digunakan untuk koneksi aktuator dan komponen keluaran sistem lainnya. Konektor pada setiap bagian disusun untuk mempermudah pemasangan dan pelepasan komponen ketika dilakukan pemeriksaan atau perawatan sistem. Penempatan komponen juga disesuaikan dengan kebutuhan ruang dan arah jalur sehingga rangkaian dapat ditempatkan secara efisien pada papan PCB.



Gambar 4.2 Realisasi *Layout Design* PCB EasyEDA pada papan PCB

Jalur PCB pada Gambar 4.2 dirancang dengan memperhatikan hubungan setiap pin berdasarkan rangkaian yang telah dibuat. Jalur catu daya dan jalur sinyal disusun menuju komponen yang sesuai sehingga tidak terjadi kesalahan koneksi. Proses *routing* dilakukan dengan mempertimbangkan posisi komponen, arah jalur, serta kebutuhan koneksi antar-komponen. Setelah seluruh jalur selesai dibuat,

layout diperiksa kembali untuk memastikan setiap jalur telah terhubung sesuai dengan rancangan dan tidak terdapat jalur yang saling bertabrakan.

4.4. Implementasi Perangkat Keras (*Hardware*)

Implementasi perangkat keras pada penelitian ini dilakukan untuk merealisasikan hasil perancangan sistem *conveyor* penyortir buah kopi berdasarkan tingkat kematangan menggunakan sensor warna TCS34725 berbasis ESP32. Sistem yang dikembangkan terdiri dari beberapa subsistem utama, yaitu sistem *conveyor* sebagai media pemindahan buah kopi, sistem sensor sebagai pendeteksi karakteristik warna buah kopi, sistem aktuator sebagai mekanisme penyortiran, serta sistem kontrol berbasis ESP32 yang berfungsi sebagai pusat pengolahan data dan pengendalian seluruh perangkat. Integrasi seluruh komponen tersebut dirancang agar mampu melakukan proses identifikasi dan penyortiran buah kopi secara otomatis berdasarkan tingkat kematangannya.

Pada tahap implementasi, setiap komponen ditempatkan sesuai dengan hasil perancangan mekanik dan elektrik yang telah disusun pada Bab III. Penempatan komponen dilakukan dengan mempertimbangkan kestabilan sistem, kemudahan proses pemeliharaan, serta akurasi

pembacaan sensor. Selain itu, tata letak komponen juga dirancang untuk meminimalkan gangguan yang dapat mempengaruhi proses identifikasi warna buah kopi, seperti getaran *conveyor*, perubahan intensitas cahaya lingkungan, maupun gangguan elektromagnetik dari motor penggerak.

4.4.1. Implementasi Rangka dan Sistem *Conveyor*

Sistem *conveyor* merupakan bagian utama yang berfungsi sebagai media transportasi buah kopi dari area pemasukan menuju area pembacaan sensor dan selanjutnya menuju area penyortiran. *Conveyor* dirancang menggunakan mekanisme *belt conveyor* karena memiliki konstruksi yang sederhana, mudah diimplementasikan, serta mampu memindahkan objek berukuran kecil secara kontinu. Pemilihan *conveyor* jenis ini juga didasarkan pada karakteristik buah kopi

yang relatif ringan sehingga tidak memerlukan sistem pemindahan dengan kapasitas beban besar.

Pada tahap implementasi, rangka *conveyor* dibuat berdasarkan komponen mekanis yang telah dirancang pada Bab III. Rangka berfungsi sebagai struktur utama untuk menopang *belt conveyor*, *roller*, motor penggerak, serta komponen pendukung lainnya. Pemasangan setiap bagian dilakukan dengan memperhatikan kesesuaian ukuran dan posisi agar rangka memiliki kestabilan yang baik selama *conveyor* beroperasi.



Gambar 4.3 Rangka Sistem *Drive Pulley Belt Conveyor*

Dapat dilihat pada Gambar 4.3, *Belt conveyor* dipasang pada bagian atas rangka dan dihubungkan dengan *roller* sebagai media penggerak dan penopang *belt*. *Roller* dipasang secara presisi pada rangka sehingga *belt* dapat bergerak secara kontinu mengikuti putaran motor. Untuk menjaga agar *belt* tetap berada pada posisi yang sesuai, dilakukan penyetelan terhadap posisi dan kekencangan *belt* setelah seluruh komponen terpasang. Penyetelan tersebut dilakukan untuk mengurangi kemungkinan *belt* bergeser atau keluar dari jalur (*tracking issue*) selama *conveyor* beroperasi.

Motor *stepper* NEMA17 dipasang pada bagian penggerak *conveyor* dan dihubungkan dengan mekanisme transmisi menggunakan *pulley* dan *timing belt*. Putaran motor diteruskan menuju *roller* sehingga menghasilkan gerakan *belt conveyor*. Pemasangan motor dan mekanisme transmisi dilakukan dengan memastikan posisi *pulley* berada pada satu garis agar perpindahan putaran dapat berlangsung dengan baik dan tidak menyebabkan *belt* transmisi mudah terlepas.

Penggerak *conveyor* menggunakan motor *stepper* NEMA17 yang dikendalikan melalui *driver* A4988. Kombinasi kedua komponen tersebut memungkinkan pengaturan arah putaran, kecepatan, dan jumlah langkah motor secara presisi. Dengan karakteristik tersebut, *conveyor* dapat bergerak secara stabil sehingga proses identifikasi tingkat kematangan buah kopi menggunakan sensor TCS34725 dapat dilakukan dengan lebih akurat. Pengaturan kecepatan *conveyor* menjadi faktor penting karena mempengaruhi waktu pembacaan sensor terhadap objek yang melintas. Oleh sebab itu, kecepatan motor diatur pada nilai yang optimal agar diperoleh keseimbangan antara kapasitas penyortiran dan kualitas hasil identifikasi.

4.4.2. Implementasi Sistem Penyortiran

Sistem penyortiran berfungsi untuk memisahkan buah kopi berdasarkan hasil identifikasi tingkat kematangan yang diperoleh dari sensor warna. Mekanisme penyortiran menggunakan motor servo yang dikendalikan langsung oleh mikrokontroler ESP32. Pemilihan motor servo digunakan sebagai aktuator pengarah yang mengatur jalur pergerakan buah kopi menuju wadah berdasarkan kategori hasil klasifikasi.

Prinsip kerja sistem penyortiran dimulai ketika sensor TCS34725 berhasil mengidentifikasi tingkat kematangan buah kopi yang sedang berada pada area pembacaan. Hasil identifikasi tersebut kemudian digunakan sebagai dasar pengambilan keputusan oleh ESP32. Berdasarkan kategori yang diperoleh, mikrokontroler akan mengirimkan sinyal PWM kepada motor servo untuk menggerakkan mekanisme pengarah menuju jalur yang sesuai. Algoritma

Pengendalian servo dan nilai *step conveyor* dapat dilihat pada Gambar 4.4 sebagai berikut.

```
else {
    if (
        currentClassification == CLASS_MENTAH) {
        servoA.write(
            SORTER_SERVO_IDLE_ANGLE);
    }

    else if (
        currentClassification == CLASS_SEDANG) {
        servoB.write(
            SORTER_SERVO_IDLE_ANGLE);
    }

    else if (
        currentClassification == CLASS_MATANG) {
        servoC.write(
            SORTER_SERVO_IDLE_ANGLE);
    }
}
```

```
//=====
// JUMLAH STEP SORTIR
//=====

#define STEP_MENTAH 27550UL
#define STEP_SEDANG 41800UL
#define STEP_MATANG 53250UL
```

Gambar 4.4 Algoritma Pengendalian Servo dan Nilai *Step Conveyor*

Penggunaan motor servo memungkinkan proses perpindahan jalur sortir dilakukan secara cepat dan akurat. Selain itu, servo juga memiliki konsumsi daya yang relatif rendah sehingga sesuai untuk diterapkan pada sistem otomatis skala laboratorium. Dengan adanya sistem penyortiran otomatis ini, proses pemisahan buah kopi dapat dilakukan tanpa intervensi operator sehingga meningkatkan efisiensi dan konsistensi hasil sortir.

4.4.3. Implementasi Sistem Pengukuran dan Pengolahan Data *Load Cell*

Implementasi sistem pengukuran berat dilakukan untuk memperoleh informasi massa buah kopi hasil penyortiran. Sistem ini menggunakan sensor *load cell* yang dikombinasikan dengan modul penguat HX711 sehingga perubahan gaya yang diterima sensor dapat dikonversi menjadi data digital yang dapat diproses oleh mikrokontroler.

Sensor ditempatkan pada wadah penampungan hasil sortir sehingga berat buah kopi yang terkumpul dapat diukur secara otomatis. Ketika jumlah buah kopi pada wadah bertambah, gaya yang diterima sensor juga meningkat dan menghasilkan perubahan sinyal listrik yang proporsional terhadap beban yang diberikan.

Data berat yang diperoleh dari sensor kemudian dibaca oleh ESP32 dan dikonversi ke dalam satuan gram melalui proses kalibrasi. Nilai hasil pengukuran ditampilkan pada LCD serta dikirimkan ke *website* monitoring sehingga pengguna dapat mengetahui berat hasil penyortiran secara *real-time*.

Penerapan sistem pengukuran berat memberikan informasi tambahan mengenai hasil penyortiran yang dilakukan. Selain digunakan sebagai parameter monitoring, data berat juga dapat dimanfaatkan untuk mengevaluasi produktivitas dan performa sistem penyortiran yang telah dirancang.

4.4.4. Implementasi Sistem Kontrol

Implementasi sistem kontrol merupakan tahap penerapan perangkat lunak pada mikrokontroler ESP32 untuk mengendalikan keseluruhan proses kerja sistem *conveyor* penyortir buah kopi. Sistem kontrol mengintegrasikan pembacaan sensor, pengolahan data, pengendalian motor *conveyor* dan aktuator penyortiran, pengukuran berat, serta penyajian data hasil penyortiran. Setiap proses dikendalikan oleh ESP32 berdasarkan kondisi dan data yang diperoleh dari masing-masing perangkat sehingga sistem dapat menjalankan proses pemindahan, identifikasi, penyortiran, pengukuran, dan monitoring secara otomatis.

4.4.4.1 *Flow* Program ESP32

ESP32 digunakan sebagai pusat kendali utama pada sistem *conveyor* penyortir buah kopi. Program yang dijalankan pada ESP32 mengatur proses kerja alat mulai dari inisialisasi perangkat, pembacaan sensor, pengolahan data, penentuan tingkat kematangan, pengendalian *conveyor* dan aktuator penyortiran, hingga pembacaan berat hasil penyortiran dan pengiriman data ke sistem monitoring. Seluruh proses tersebut dijalankan secara berurutan dan berulang selama sistem berada dalam kondisi operasi.



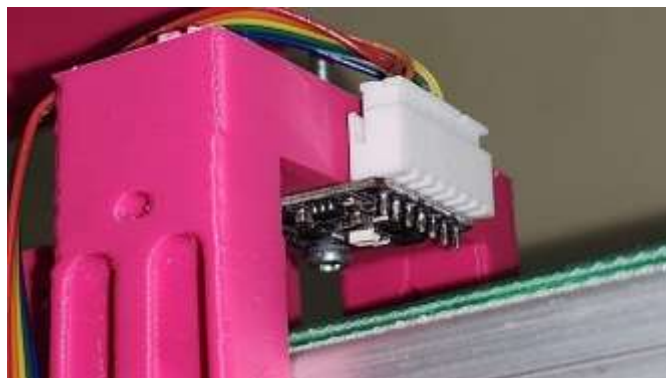
Gambar 4.5 Penempatan ESP32

Pada Gambar 4.5, ESP32 melakukan inisialisasi terhadap perangkat yang digunakan, meliputi sensor warna TCS34725, sensor *load cell*, *driver* motor, motor servo, LCD, tombol kontrol, serta koneksi Wi-Fi. Setelah seluruh perangkat berhasil diinisialisasi, sistem berada pada kondisi siap bekerja. Ketika sistem dijalankan, motor *stepper* menggerakkan *conveyor* untuk membawa buah kopi dari area pemasukan menuju area pembacaan sensor. Saat buah kopi berada pada posisi pembacaan, sensor warna TCS34725 membaca karakteristik warna permukaan buah dan menghasilkan data warna yang kemudian diproses oleh ESP32.

Data hasil pembacaan sensor digunakan untuk menentukan kategori tingkat kematangan buah kopi. Setelah kategori diperoleh, ESP32 memberikan perintah kepada aktuator penyortiran untuk mengarahkan buah menuju wadah yang sesuai. Setelah buah masuk ke wadah hasil penyortiran, sensor *load cell* melakukan pengukuran berat buah yang terkumpul. Hasil pengukuran kemudian diproses dan ditampilkan melalui LCD serta dikirimkan ke halaman monitoring berbasis web. Setelah satu proses penyortiran selesai, sistem kembali ke tahap berikutnya untuk menerima dan mengolah buah kopi selanjutnya. Alur kerja tersebut dilakukan secara berulang sehingga proses pemindahan, identifikasi, penyortiran, pengukuran, dan monitoring dapat berlangsung secara otomatis.

4.4.4.2 Sistem Sensor Warna TCS34725

Sensor warna TCS34725 digunakan sebagai komponen input pada sistem untuk memperoleh informasi warna dari buah kopi yang melewati *conveyor*. Sensor ditempatkan pada bagian pengamatan *conveyor* sehingga dapat membaca warna permukaan buah kopi sebelum buah diarahkan menuju proses penyortiran seperti pada Gambar 4.5. Pada implementasinya, sensor TCS34725 dihubungkan dengan mikrokontroler ESP32 dan digunakan untuk memperoleh data warna melalui empat kanal pembacaan, yaitu *Red* (R), *Green* (G), *Blue* (B), dan *Clear* (C). Kanal *Red* digunakan untuk memperoleh informasi intensitas warna merah, kanal *Green* untuk memperoleh informasi intensitas warna hijau, kanal *Blue* untuk memperoleh informasi intensitas warna biru, sedangkan kanal *Clear* digunakan untuk memperoleh nilai intensitas cahaya secara keseluruhan yang diterima oleh sensor.



Gambar 4.6 Penempatan Pengamatan Sensor

Sensor ini dipasang pada bagian atas lintasan *conveyor* sehingga buah kopi yang bergerak dapat melewati area pembacaan sensor secara berurutan. Pemasangan sensor pada Gambar 4.6 dibuat juga dengan jarak tertentu terhadap permukaan buah agar kondisi pembacaan dapat dibuat lebih konsisten pada setiap objek yang melewati area deteksi. Posisi sisi sensor dibuat dengan jarak tertentu terhadap permukaan buah agar kondisi pembacaan dapat dibuat lebih konsisten pada setiap objek yang melewati area deteksi. Pemasangan sensor juga memperhatikan kondisi pencahayaan pada area pembacaan. Pencahayaan yang relatif seragam diperlukan agar perubahan intensitas cahaya lingkungan tidak terlalu memengaruhi hasil pembacaan warna. Sensor dihubungkan dengan ESP32 sebagai mikrokontroler utama sehingga data yang diperoleh dari sensor dapat diteruskan untuk diproses lebih lanjut. Sensor TCS34725 digunakan untuk memperoleh data kanal warna *Red*, *Green*, *Blue*, dan *Clear* yang menjadi dasar dalam proses identifikasi tingkat kematangan buah kopi.

Dengan pemasangan tersebut, setiap buah kopi yang melewati area pembacaan akan memperoleh kondisi pengukuran yang relatif sama. Data hasil pembacaan sensor tersebut kemudian dikirimkan ke mikrokontroler ESP32 melalui pin digital untuk diproses lebih lanjut. ESP32 membandingkan nilai RGB tersebut dengan parameter klasifikasi yang telah ditentukan sebelumnya sehingga dapat mengidentifikasi tingkat kematangan buah kopi. Hasil identifikasi selanjutnya digunakan sebagai dasar pengendalian motor servo untuk mengarahkan buah kopi ke wadah penyortiran yang sesuai.

4.4.4.3 Sistem *Load Cell*

Sensor *Load cell* digunakan untuk mengukur berat buah kopi pada masing-masing wadah hasil penyortiran. Pada implementasinya, *load cell* dipasang pada bagian bawah wadah penampungan sehingga beban buah kopi yang masuk ke dalam wadah dapat diterima oleh sensor. *Load cell* dihubungkan dengan modul HX711 sebagai penguat dan pembaca sinyal, kemudian data hasil pengukuran diteruskan ke ESP32 untuk diproses menjadi nilai berat.



Gambar 4.7 Sistem *Load Cell*

Berdasarkan Gambar 4.7, penempatan *load cell* pada bagian bawah wadah dilakukan agar berat buah yang terkumpul dapat diteruskan secara langsung sebagai beban pada sensor. Ketika jumlah buah kopi pada wadah bertambah, beban yang diterima *load cell* juga meningkat sehingga menghasilkan perubahan sinyal yang dapat dibaca oleh sistem. Setiap *load cell* digunakan untuk memantau berat pada kategori hasil penyortiran yang telah ditentukan.

Data hasil pembacaan *load cell* kemudian digunakan sebagai informasi berat hasil penyortiran. Nilai tersebut tidak hanya digunakan untuk ditampilkan pada sistem secara lokal melalui LCD, tetapi juga menjadi data yang dikirimkan ke halaman monitoring. Dengan demikian, implementasi *load cell* memungkinkan sistem memberikan informasi mengenai berat buah kopi pada setiap kategori hasil penyortiran secara otomatis.

4.4.4.4 *Driver Motor A4988 Nema17 dan Servo MG996R*

Pada sistem *conveyor* penyortir buah kopi, motor *stepper* NEMA17 digunakan sebagai penggerak utama *belt conveyor*, sedangkan motor servo MG996R digunakan sebagai aktuator pada mekanisme penyortiran buah berdasarkan tingkat kematangan. Pengendalian kedua motor dilakukan oleh ESP32 sesuai dengan proses kerja sistem yang telah dirancang. Motor *stepper* NEMA17 tidak dihubungkan secara langsung dengan ESP32, tetapi menggunakan *driver* A4988 sebagai pengendali motor. Penggunaan *driver* diperlukan untuk mengatur pemberian pulsa dan arah putaran motor *stepper* sehingga motor dapat bergerak

sesuai dengan kebutuhan sistem *conveyor*. Pada Gambar 4.8 menunjukkan konfigurasi pin pada *Driver* A4988 dan Motor Servo MG996R.

```
// A4988
#define STEP_PIN 16
#define DIR_PIN 17

// Servo
// A = Mentah
// B = Setengah Matang
// C = Matang
// D = Silo

#define SERVO_A_PIN 23
#define SERVO_B_PIN 19
#define SERVO_C_PIN 18
#define SERVO_D_PIN 5
```

Gambar 4.8 Konfigurasi Pin *Driver* A4988 dan Motor Servo pada ESP32

Pada implementasi sistem kontrol, ESP32 digunakan sebagai pengendali *driver* A4988 dan motor servo. *Driver* A4988 digunakan untuk mengendalikan motor *stepper* NEMA17 sebagai penggerak *conveyor*. Pada program, pin STEP *driver* A4988 dihubungkan dengan GPIO 16 ESP32, sedangkan pin DIR dihubungkan dengan GPIO 17. Pin STEP digunakan sebagai masukan pulsa untuk mengatur pergerakan motor *stepper*, sedangkan pin DIR digunakan untuk menentukan arah putaran motor. Konfigurasi tersebut memungkinkan ESP32 mengendalikan pergerakan motor *stepper* sesuai dengan kebutuhan sistem *conveyor*.

Selain motor *stepper*, sistem menggunakan motor servo MG996R sebagai aktuator pada mekanisme penyortiran. Berdasarkan konfigurasi program, terdapat empat servo yang digunakan pada sistem, yaitu Servo A untuk kategori buah mentah, Servo B untuk kategori buah setengah matang, Servo C untuk kategori buah matang, dan Servo D yang digunakan pada bagian silo. Masing-masing servo dihubungkan pada pin ESP32 yang berbeda, yaitu GPIO 23 untuk Servo A, GPIO 19 untuk Servo B, GPIO 18 untuk Servo C, dan GPIO 5 untuk Servo D. Penggunaan

beberapa servo tersebut memungkinkan mekanisme penyortiran bekerja sesuai dengan kategori kematangan buah yang telah ditentukan oleh sistem.

4.4.4.5 Sistem Catu Daya

Sistem catu daya berfungsi sebagai sumber energi listrik bagi seluruh komponen yang digunakan pada *prototype conveyor* penyortir buah kopi. Pada penelitian ini digunakan *power supply* AC-DC 12V 5A yang mengubah tegangan AC dari jaringan listrik menjadi tegangan DC 12V sebagai sumber tegangan utama sistem. Pemilihan *power supply* tersebut didasarkan pada kebutuhan daya seluruh komponen agar dapat bekerja secara stabil selama proses pengujian.

Tegangan keluaran sebesar 12V didistribusikan ke beberapa komponen sesuai dengan kebutuhan operasinya. Motor *stepper* NEMA17 beserta *driver* A4988 memperoleh suplai tegangan langsung dari *power supply* 12V karena memerlukan tegangan dan arus yang lebih besar untuk menggerakkan *conveyor*. Sementara itu, komponen yang bekerja pada tegangan lebih rendah, seperti ESP32, sensor warna TCS34725, LCD I2C, dan sensor *load cell*, memperoleh suplai tegangan melalui DC *Buck Converter* yang menurunkan tegangan 12V menjadi 5V sesuai kebutuhan masing-masing perangkat.

Selain berfungsi sebagai penurun tegangan, penggunaan DC *Buck Converter* juga bertujuan menjaga kestabilan tegangan keluaran sehingga mikrokontroler dan sensor dapat bekerja secara optimal. Tegangan yang stabil sangat penting untuk mencegah terjadinya kesalahan pembacaan sensor maupun gangguan komunikasi antarperangkat yang dapat memengaruhi proses identifikasi tingkat kematangan buah kopi.

Secara keseluruhan, implementasi sistem catu daya mampu menyediakan suplai tegangan yang sesuai bagi setiap komponen sehingga seluruh subsistem dapat bekerja secara terintegrasi. Dengan pengaturan distribusi daya yang tepat, sistem *conveyor* penyortir buah kopi dapat beroperasi secara stabil selama proses identifikasi, penyortiran, pengukuran berat, dan monitoring berlangsung.

4.5. Implementasi Perangkat Lunak (*Software*)

Implementasi perangkat lunak dilakukan untuk merealisasikan logika kerja sistem yang telah dirancang pada tahap perancangan. Program dikembangkan menggunakan Arduino IDE dengan bahasa pemrograman C++. Perangkat lunak berfungsi sebagai penghubung antara sensor, mikrokontroler, dan aktuator sehingga seluruh subsistem dapat bekerja secara terintegrasi.

Struktur program disusun secara modular untuk memudahkan proses pengembangan dan pemeliharaan sistem. Setiap fungsi utama seperti pembacaan sensor, pengolahan data, klasifikasi tingkat kematangan, serta pengendalian aktuator ditempatkan pada bagian program yang terpisah sehingga memudahkan proses *debugging* maupun pengembangan fitur tambahan di masa mendatang.

4.5.1. Implementasi Algoritma Pembacaan Data Sensor

Implementasi algoritma pembacaan data sensor dilakukan untuk memperoleh informasi warna permukaan buah kopi sebagai dasar dalam proses identifikasi tingkat kematangan pada sensor warna TCS34725. Sensor warna TCS34725 digunakan sebagai komponen *input* pada sistem untuk memperoleh informasi warna dari buah kopi yang melewati *conveyor*. Sensor ditempatkan pada bagian pengamatan *conveyor* sehingga dapat membaca warna permukaan buah kopi sebelum buah diarahkan menuju proses penyortiran. Pada implementasinya, sensor TCS34725 dihubungkan dengan mikrokontroler ESP32 dan digunakan untuk memperoleh data warna melalui empat kanal pembacaan, yaitu *Red* (R), *Green* (G), *Blue* (B), dan *Clear* (C). Kanal *Red* digunakan untuk memperoleh informasi intensitas warna merah, kanal *Green* untuk memperoleh informasi intensitas warna hijau, kanal *Blue* untuk memperoleh informasi intensitas warna biru, sedangkan kanal *Clear* digunakan untuk memperoleh nilai intensitas cahaya secara keseluruhan yang diterima oleh sensor.

Implementasi sensor dilakukan pada bagian atas jalur *conveyor* sehingga setiap buah kopi yang melintas dapat melewati area pembacaan secara bergantian. Posisi sensor dirancang dengan jarak yang tetap terhadap permukaan buah kopi agar

nilai RGB yang diperoleh lebih konsisten. Selain itu, kecepatan *conveyor* diatur sedemikian rupa sehingga sensor memiliki waktu yang cukup untuk melakukan pembacaan warna sebelum buah bergerak menuju proses penyortiran. Untuk meningkatkan kestabilan hasil pembacaan, area sensor dilengkapi dengan pencahayaan yang seragam serta dirancang agar tidak terpengaruh oleh cahaya dari lingkungan sekitar. Kondisi pencahayaan yang stabil sangat penting karena perubahan intensitas cahaya luar dapat memengaruhi nilai RGB yang dibaca oleh sensor TCS34725. Oleh karena itu, pengujian dilakukan pada kondisi pencahayaan yang relatif konstan sehingga hasil identifikasi tingkat kematangan menjadi lebih akurat.

```

tcs.getRawData(
    &r,
    &g,
    &b,
    &c
);

xSemaphoreGive(
    i2cMutex
);

Loading...

sumR +=
    r;

sumG +=
    g;

sumB +=
    b;

sumC +=
    c;

validSamples++;
}

```

Gambar 4.9 Implementasi Program Pembacaan Data Sensor TCS34725

Proses pembacaan data sensor pada Gambar 4.9 dilakukan menggunakan fungsi `getRawData()` yang terdapat pada program. Fungsi tersebut digunakan untuk mengambil nilai dari masing-masing kanal sensor dan menyimpannya ke dalam

variabel *r*, *g*, *b*, dan *c*. Dengan demikian, setiap proses pembacaan menghasilkan empat nilai yang merepresentasikan kondisi warna dan intensitas cahaya pada objek yang sedang diamati. Nilai *Red*, *Green*, dan *Blue* selanjutnya menjadi parameter utama dalam proses identifikasi tingkat kematangan buah kopi, sedangkan nilai *Clear* digunakan sebagai data pendukung dalam proses pembacaan sensor.

Untuk memperoleh data pembacaan yang lebih representatif, sistem tidak hanya menggunakan satu hasil pembacaan sensor. Setiap nilai hasil pembacaan ditambahkan secara bertahap ke variabel akumulasi, yaitu *sumR*, *sumG*, *sumB*, dan *sumC*. Nilai pada kanal *Red* ditambahkan ke *sumR*, nilai *Green* ditambahkan ke *sumG*, nilai *Blue* ditambahkan ke *sumB*, sedangkan nilai *Clear* ditambahkan ke *sumC*. Selain itu, variabel *validSamples* digunakan untuk mencatat jumlah sampel pembacaan sensor yang berhasil diperoleh. Proses akumulasi tersebut dilakukan selama proses pengambilan sampel sehingga data dari beberapa pembacaan dapat digunakan pada tahap pengolahan selanjutnya.

Data hasil pembacaan sensor tersebut kemudian dikirimkan ke mikrokontroler ESP32 melalui pin digital untuk diproses lebih lanjut. ESP32 membandingkan nilai RGB tersebut dengan parameter klasifikasi yang telah ditentukan sebelumnya sehingga dapat mengidentifikasi tingkat kematangan buah kopi. Hasil identifikasi selanjutnya digunakan sebagai dasar pengendalian motor servo untuk mengarahkan buah kopi ke wadah penyortiran yang sesuai.

Sebelum proses pembacaan dilakukan, ESP32 menginisialisasi sensor TCS34725 melalui antarmuka komunikasi I2C. Sensor kemudian melakukan pembacaan terhadap empat kanal, yaitu *Red* (R), *Green* (G), *Blue* (B), dan *Clear* (C). Data hasil pembacaan tersebut diperoleh melalui fungsi *getRawData()* dan disimpan pada variabel *R*, *G*, *B*, dan *C*. Nilai *Red*, *Green*, dan *Blue* digunakan sebagai parameter utama dalam proses klasifikasi tingkat kematangan, sedangkan nilai *Clear* digunakan sebagai data pendukung untuk menunjukkan intensitas cahaya yang diterima sensor.

Pada proses pembacaan, program mengambil data RGB secara berulang untuk memperoleh hasil pengukuran yang lebih stabil. Data tersebut selanjutnya dibandingkan dengan parameter referensi yang telah diperoleh melalui proses kalibrasi. Sebelum digunakan pada proses klasifikasi, setiap data hasil pembacaan diperiksa untuk memastikan nilainya berada dalam rentang yang telah ditentukan berdasarkan proses kalibrasi. Tahap ini bertujuan untuk mengurangi kemungkinan kesalahan pembacaan akibat perubahan kondisi pencahayaan maupun gangguan selama proses pengukuran.

Dengan algoritma pembacaan tersebut, ESP32 mampu memperoleh informasi warna permukaan buah kopi secara berurutan dan konsisten. Data RGB yang telah diperoleh kemudian diteruskan ke proses klasifikasi tingkat kematangan untuk menentukan kategori buah kopi, yaitu matang, setengah matang, atau mentah.

4.5.2. Implementasi Algoritma Klasifikasi Tingkat Kematangan

Implementasi algoritma klasifikasi tingkat kematangan dilakukan untuk mengidentifikasi kondisi buah kopi berdasarkan nilai komponen warna merah (*Red*), hijau (*Green*), dan biru (*Blue*) atau RGB yang diperoleh dari sensor warna TCS34725. Proses klasifikasi merupakan inti dari sistem penyortiran karena hasil identifikasi digunakan sebagai dasar pengambilan keputusan dalam mengendalikan mekanisme penyortiran secara otomatis. Pada penelitian ini, klasifikasi dilakukan dengan membandingkan nilai RGB hasil pembacaan sensor terhadap rentang nilai yang telah ditentukan melalui proses kalibrasi dan pengujian.

Ketika buah kopi melewati area pembacaan, sensor TCS34725 mendeteksi warna permukaan buah dan menghasilkan data kanal *Red*, *Green*, *Blue*, dan *Clear*. Data hasil pembacaan sensor TCS34725 kemudian dibaca oleh ESP32 melalui komunikasi I2C dan digunakan sebagai parameter warna dalam proses identifikasi. Nilai RGB tersebut dibandingkan dengan batas klasifikasi yang telah ditentukan berdasarkan hasil kalibrasi dan pengujian.

Berdasarkan hasil pengolahan tersebut, sistem mengelompokkan buah kopi ke dalam tiga kategori, yaitu matang, setengah matang, dan mentah. Buah kopi dikategorikan sebagai matang apabila hasil pembacaan menunjukkan dominasi komponen warna merah (*Red*). Apabila komponen warna hijau (*Green*) lebih dominan, buah kopi dikategorikan sebagai mentah. Sementara itu, buah kopi dengan nilai RGB yang berada di antara kedua kondisi tersebut diklasifikasikan sebagai setengah matang. Rentang nilai setiap kategori diperoleh melalui proses kalibrasi menggunakan beberapa sampel buah kopi dengan tingkat kematangan yang berbeda.

Hasil klasifikasi yang diperoleh selanjutnya digunakan sebagai acuan dalam proses penyortiran otomatis. ESP32 mengirimkan sinyal kendali kepada motor servo MG996R untuk mengarahkan buah kopi ke wadah sesuai dengan kategori kematangannya. Selain digunakan sebagai dasar pengendalian aktuator, informasi hasil klasifikasi juga ditampilkan pada LCD dan dikirimkan ke halaman *website* berbasis HTML melalui jaringan Wi-Fi sehingga pengguna dapat memantau hasil penyortiran secara *real-time*.

4.5.3. Implementasi Algoritma Penyortiran

Setelah Implementasi algoritma penyortiran dilakukan untuk memisahkan buah kopi ke dalam wadah yang berbeda sesuai dengan hasil klasifikasi tingkat kematangannya. Algoritma ini bekerja setelah proses identifikasi tingkat kematangan selesai dilakukan oleh sistem kendali utama. Tujuan dari algoritma penyortiran adalah memastikan setiap buah kopi diarahkan ke wadah yang sesuai secara otomatis tanpa memerlukan campur tangan operator sehingga proses penyortiran dapat berlangsung secara cepat, konsisten, dan berkesinambungan.

Setelah ESP32 menerima hasil klasifikasi berdasarkan nilai RGB yang diperoleh dari sensor warna TCS34725, mikrokontroler akan menentukan posisi motor servo MG996R sesuai dengan kategori tingkat kematangan buah kopi. Buah kopi yang diklasifikasikan sebagai matang diarahkan ke wadah pertama, buah kopi setengah matang diarahkan ke wadah kedua, sedangkan buah kopi mentah diarahkan ke wadah ketiga. Penentuan posisi servo dilakukan berdasarkan sudut

putar yang telah diprogram sebelumnya sehingga setiap kategori memiliki jalur penyortiran yang berbeda.

Selama proses penyortiran berlangsung, *conveyor* tetap bergerak dengan kecepatan yang telah ditentukan sehingga buah kopi dapat berpindah dari area pembacaan sensor menuju area penyortiran secara berurutan. ESP32 menghitung waktu tempuh buah kopi dari posisi pembacaan sensor hingga mencapai mekanisme penyortiran. Setelah buah kopi berada pada posisi yang telah ditentukan, ESP32 mengirimkan sinyal PWM kepada motor servo untuk menggerakkan mekanisme pengarah sesuai dengan hasil klasifikasi. Setelah proses penyortiran selesai, motor servo kembali ke posisi awal dan bersiap untuk menerima perintah penyortiran berikutnya.

Implementasi algoritma penyortiran yang terintegrasi dengan sensor warna TCS34725, motor *stepper* NEMA17 sebagai penggerak *conveyor*, dan motor servo MG996R sebagai aktuator memungkinkan proses pemisahan buah kopi berlangsung secara otomatis dengan tingkat konsistensi yang lebih baik dibandingkan penyortiran manual. Integrasi tersebut juga mendukung proses monitoring secara *real-time*, karena setiap hasil klasifikasi dan penyortiran dikirimkan oleh ESP32 ke halaman *website* berbasis HTML sehingga pengguna dapat memantau kinerja sistem selama proses berlangsung.

4.6. Implementasi Antarmuka Monitoring

Implementasi antarmuka monitoring dilakukan untuk memberikan kemudahan dalam memantau hasil penyortiran buah kopi secara *real-time*. Antarmuka monitoring dikembangkan menggunakan *web server* yang dijalankan oleh mikrokontroler ESP32 sehingga informasi hasil identifikasi tingkat kematangan dan pengukuran berat dapat diakses melalui *browser* pada komputer maupun *smartphone* yang terhubung ke jaringan Wi-Fi yang sama.

4.6.1. Website HTML

Website monitoring dikembangkan menggunakan bahasa *markup* HTML (*HyperText Markup Language*) sebagai struktur utama halaman *web* yang

dijalankan pada ESP32. Halaman *web* dirancang dengan tampilan yang sederhana sehingga informasi hasil penyortiran dapat ditampilkan secara jelas dan mudah dipahami oleh pengguna. *Website* dapat diakses melalui alamat IP *Address* yang diberikan oleh ESP32 setelah perangkat berhasil terhubung ke jaringan Wi-Fi.



Gambar 4.10 Tampilan *Website* Monitoring

Halaman monitoring pada Gambar 4.10 menampilkan beberapa informasi utama, yaitu status koneksi sistem, hasil identifikasi tingkat kematangan buah kopi, data berat hasil penyortiran yang diperoleh dari sensor *load cell*, serta jumlah buah kopi pada masing-masing kategori penyortiran. Informasi tersebut ditampilkan secara sistematis menggunakan teks, tabel, maupun elemen antarmuka lainnya sehingga memudahkan dalam memantau kondisi alat selama proses penyortiran berlangsung.

Setiap kali sensor warna TCS34725 melakukan identifikasi terhadap buah kopi dan sensor *load cell* mengukur berat hasil penyortiran, ESP32 akan memperbarui data yang ditampilkan pada halaman *website*. Dengan mekanisme tersebut, dapat diperoleh informasi hasil penyortiran secara *real-time* tanpa perlu melakukan pembaruan halaman secara manual.

Penggunaan *website* berbasis HTML memberikan beberapa keuntungan, di antaranya tidak memerlukan instalasi aplikasi tambahan, dapat diakses melalui berbagai perangkat yang memiliki *browser*, serta memudahkan proses monitoring selama alat beroperasi. Selain itu, tampilan *website* dapat dikembangkan lebih

lanjut dengan menambahkan fitur visualisasi data maupun penyimpanan riwayat hasil penyortiran.

4.7. Implementasi Keseluruhan Sistem

Implementasi keseluruhan sistem dilakukan dengan mengintegrasikan seluruh perangkat keras dan perangkat lunak yang telah dirancang menjadi satu kesatuan sistem *conveyor* penyortir buah kopi berdasarkan tingkat kematangan. Integrasi ini meliputi sensor warna TCS34725 sebagai pendeteksi tingkat kematangan, sensor *load cell* sebagai pengukur berat hasil penyortiran, motor stepper NEMA17 sebagai penggerak *conveyor* melalui *driver* A4988, motor servo MG996R sebagai mekanisme penyortiran, mikrokontroler ESP32 sebagai pusat kendali, LCD I2C sebagai media tampilan lokal, serta *web server* berbasis HTML sebagai antarmuka monitoring.

Proses kerja sistem diawali ketika buah kopi dimasukkan ke dalam *hopper*, kemudian dibawa oleh *belt conveyor* menuju area pembacaan sensor. *Conveyor* digerakkan oleh motor *stepper* NEMA17 dengan kecepatan yang telah diatur sehingga setiap buah kopi melewati sensor secara bergantian. Pengaturan kecepatan *conveyor* dilakukan untuk memberikan waktu yang cukup bagi sensor TCS34725 dalam membaca warna permukaan buah secara optimal.

Ketika buah kopi berada pada area pembacaan, sensor warna TCS34725 mendeteksi intensitas komponen warna merah (*Red*), hijau (*Green*), dan biru (*Blue*) pada permukaan buah kopi. Nilai RGB hasil pembacaan kemudian dikirimkan ke mikrokontroler ESP32 untuk diproses menggunakan algoritma klasifikasi yang telah dirancang. Berdasarkan hasil pengolahan tersebut, ESP32 menentukan kategori tingkat kematangan buah kopi menjadi matang, setengah matang, atau mentah.

Setelah proses identifikasi selesai, ESP32 mengirimkan sinyal PWM kepada motor servo MG996R untuk menggerakkan mekanisme penyortiran sesuai dengan kategori yang telah ditentukan. Servo mengarahkan buah kopi menuju wadah penampungan yang sesuai, sedangkan *conveyor* tetap bergerak untuk membawa

buah kopi berikutnya menuju area pembacaan sensor. Dengan mekanisme tersebut, proses identifikasi dan penyortiran dapat berlangsung secara berkesinambungan.

Selanjutnya, sensor *load cell* mengukur berat buah kopi yang telah masuk ke masing-masing wadah penyortiran. Data hasil pengukuran berat diproses oleh ESP32 dan dikombinasikan dengan hasil identifikasi tingkat kematangan. Informasi tersebut kemudian ditampilkan pada LCD I2C sebagai tampilan lokal serta dikirimkan ke halaman monitoring berbasis HTML melalui jaringan Wi-Fi sehingga pengguna dapat memantau kondisi sistem selama proses penyortiran berlangsung.

Melalui integrasi seluruh komponen tersebut, sistem mampu melakukan identifikasi tingkat kematangan buah kopi berdasarkan karakteristik warna, melakukan penyortiran secara otomatis, mengukur berat hasil penyortiran, serta menyajikan informasi hasil proses secara *real-time* pada halaman monitoring. Integrasi antara perangkat keras dan perangkat lunak tersebut menjadikan sistem bekerja secara otomatis, konsisten, dan mampu meningkatkan efisiensi proses penyortiran buah kopi dibandingkan dengan metode manual.