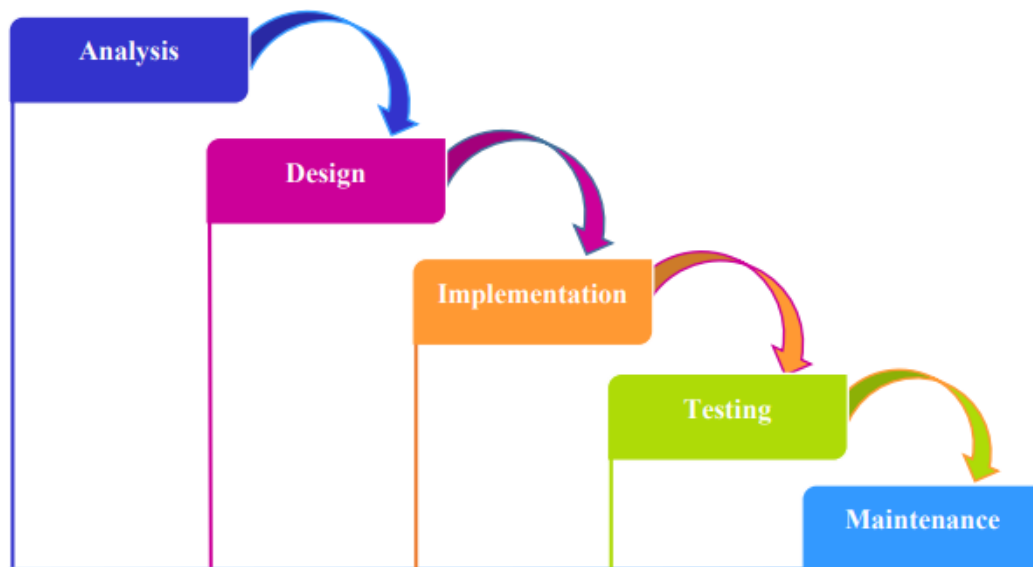


## BAB III METODOLOGI PENYELESAIAN MASALAH

### III.1 Metodologi Penelitian

Pada bab ini akan dibahas mengenai metode pelaksanaan dalam rancang bangun mesin penyortir warna tomat cherry berbasis *computer vision* dengan menggunakan metode Waterfall.



Gambar III.1 Metode Waterfall

Metode Waterfall merupakan salah satu metode pengembangan perangkat lunak yang paling umum digunakan. Pendekatan ini mengikuti alur yang bersifat linear, dimulai dari tahap awal seperti perencanaan siste hingga tahap akhir yaitu pemeliharaan. Setiap tahap harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya, dan tidak memungkinkan untuk kembali ke tahap sebelumnya setelah tahap tersebut selesai [25].

## **III.2 Analisis Kebutuhan**

### **III.2.1 Kebutuhan Mekanisme Penyortiran**

Pada proses penyortiran tomat berdasarkan tingkat kematangan (matang, setengah matang, mentah, dan busuk), dibutuhkan mekanisme fisik yang mampu mengalihkan tomat secara otomatis ke jalur yang sesuai. Mekanisme ini harus mampu memisahkan empat jenis tomat dengan cepat, selaras dengan laju konveyor, serta meminimalkan kerusakan fisik pada tomat, khususnya yang sudah matang. Selain itu, dibutuhkan sistem yang efisien secara ruang dan energi, mudah dirawat atau dimodifikasi, serta menggunakan aktuator seminimal mungkin namun tetap presisi. Berdasarkan kebutuhan tersebut, mekanisme sortir masing-masing dibandingkan dari segi efisiensi dan kompleksitas.

Mekanisme *rotary* dan *pusher* dipilih untuk penyortiran tomat pada penelitian ini. Desain ini hanya memerlukan dua aktuator servo, yaitu satu servo rotary untuk mengarahkan tomat mentah dan setengah matang ke wadah masing-masing, serta satu servo pusher untuk mendorong tomat busuk ke wadah pembuangan. Sementara itu, tomat matang dibiarkan meluncur langsung ke wadah utamanya tanpa intervensi. Mekanisme ini tidak hanya sederhana dan hemat ruang, dan tetap menjaga kondisi fisik tomat. Selain itu, sistem ini mudah dikembangkan apabila di masa mendatang diperlukan penambahan kategori sortir lainnya.

### **III.1.2 Kebutuhan Perangkat Keras**

- 1) Webcam : Berfungsi untuk menangkap citra atau video secara real-time dari alur tomat cherry di atas konveyor. Kamera ini menjadi sensor utama yang memberikan input visual ke laptop agar sistem dapat melakukan deteksi objek, segmentasi, serta penghitungan luas atau ukuran tomat cherry.
- 2) Motor DC : digunakan untuk menggerakkan belt konveyor sehingga tomat cherry dapat berjalan secara kontinu melewati area deteksi kamera. Kecepatan motor DC dapat diatur sesuai kebutuhan, sehingga sistem dapat menyesuaikan

dengan kecepatan deteksi agar tidak terjadi kesalahan sortir.

- 3) Motor Servo : Motor servo digunakan untuk menggerakkan mekanisme pemisah (separator) yang bertugas memindahkan tomat cherry ke jalur yang sesuai berdasarkan hasil klasifikasi. Servo bergerak presisi ke arah tertentu untuk memastikan tomat dapat dipisahkan ke wadah sesuai klasifikasi kematangannya.
- 4) Tomat Cherry : Tomat cherry berperan sebagai objek nyata yang diamati oleh sistem. Sistem akan mendeteksi tingkat kematangan, menghitung jumlah, serta mengukur ukuran atau luas tomat. Dan memisahkan ke wadah sesuai klasifikasi.
- 5) Arduino : Berfungsi sebagai pengendali utama perangkat keras. Arduino menerima perintah dari laptop untuk mengatur motor DC (konveyor) dan motor servo (separator). Dengan demikian, Arduino menjadi penghubung antara hasil pemrosesan data (*software*) dengan pergerakan fisik mesin sortir (*hardware*).
- 6) Laptop : Digunakan sebagai pusat pemrosesan data yang diterima dari kamera (webcam). Laptop menjalankan software berbasis computer vision (misalnya Python + YOLOV8) untuk mendeteksi kondisi tomat cherry, menghitung jumlah, dan menentukan kategori (matang, setengah matang, atau busuk).

### **III.2.3 Kebutuhan Perangkat Lunak**

- 1) Microsoft Excel : Digunakan untuk menyimpan, mengelola, serta menganalisis data hasil pengujian sistem penyortir. Excel sebagai catatan hasil deteksi.
- 2) Website Roboflow : Berfungsi sebagai platform untuk membuat, mengelola, dan mengolah dataset gambar tomat cherry. Di Roboflow, dataset dapat diberi anotasi (labeling), diatur ukurannya, serta dipersiapkan dengan kompatibel.

- 3) Google Drive : Digunakan sebagai media penyimpanan berbasis cloud yang menghubungkan dataset dari Roboflow dengan Google Colab. Dengan Google Drive, file dataset dapat diakses dengan mudah saat proses training model.
- 4) Google Colab : Menjadi platform utama untuk melatih dataset gambar yang berasal dari Roboflow. Colab menyediakan lingkungan komputasi berbasis cloud dengan dukungan GPU, sehingga pelatihan model computer vision dapat dilakukan lebih cepat tanpa memerlukan perangkat keras lokal yang mahal.
- 5) Visual Studio Code (VS Code) : Digunakan sebagai Integrated Development Environment (IDE) untuk menulis program Python. VS Code digunakan untuk memprogram sistem sortir, menampilkan antarmuka Tkinter dan OpenCV.
- 6) Arduino IDE : Berfungsi untuk menulis, mengunggah, dan menguji program ke papan Arduino. Arduino IDE juga menjadi perantara komunikasi antara kode yang dijalankan di Visual Studio Code dengan perangkat keras (motor DC, servo, dan sensor). Output dari pemrosesan laptop kemudian diwujudkan dalam bentuk gerakan fisik melalui Arduino.

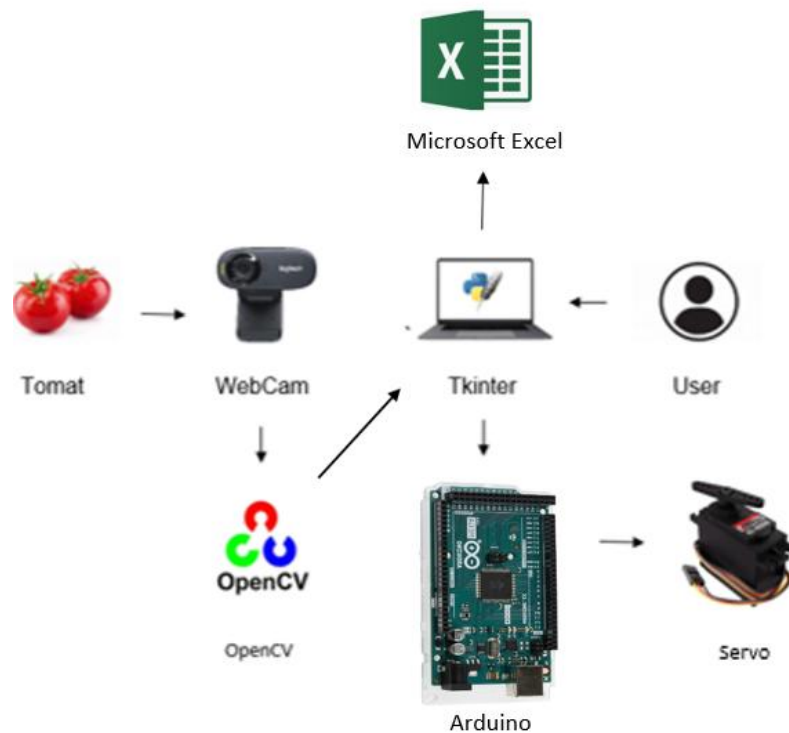
#### **III.2.4 Fungsi Sistem**

- 1) Pemantauan Visual : Webcam mengambil gambar/video tomat cherry di atas konveyor lalu mengirimkan data visual ke mini laptop. Proses ini menjadi input utama sistem.
- 2) Pemrosesan Data: Data visual diproses di Visual Studio Code (Python + YOLOv8 / segmentasi). Hasil analisis berupa deteksi jenis, ukuran, atau kondisi tomat, kemudian dikirim sebagai instruksi ke Arduino untuk dieksekusi.
- 3) Kontrol Motor Servo: Berdasarkan instruksi, Arduino menggerakkan motor servo untuk menyortir tomat cherry ke jalur tertentu sesuai kategori.
- 4) Kontrol Motor DC: Arduino juga mengatur kecepatan motor DC yang menggerakkan konveyor agar sesuai dengan laju deteksi dan penyortiran.

- 5) Penyimpanan Data: Semua hasil deteksi (jenis tomat, jumlah, hasil sortir) disimpan secara otomatis ke dalam file Microsoft Excel sebagai dokumentasi
- 6) Akses dan Kontrol Pengguna: Pengguna berinteraksi melalui antarmuka Tkinter di laptop untuk mengakses data dan memantau proses.

### III.3. Desain

#### III.3.1 Gambaran Umum Sistem



Gambar III. 2 Gambaran umum mesin penyortir warna tomat cherry berbasis computer vision.

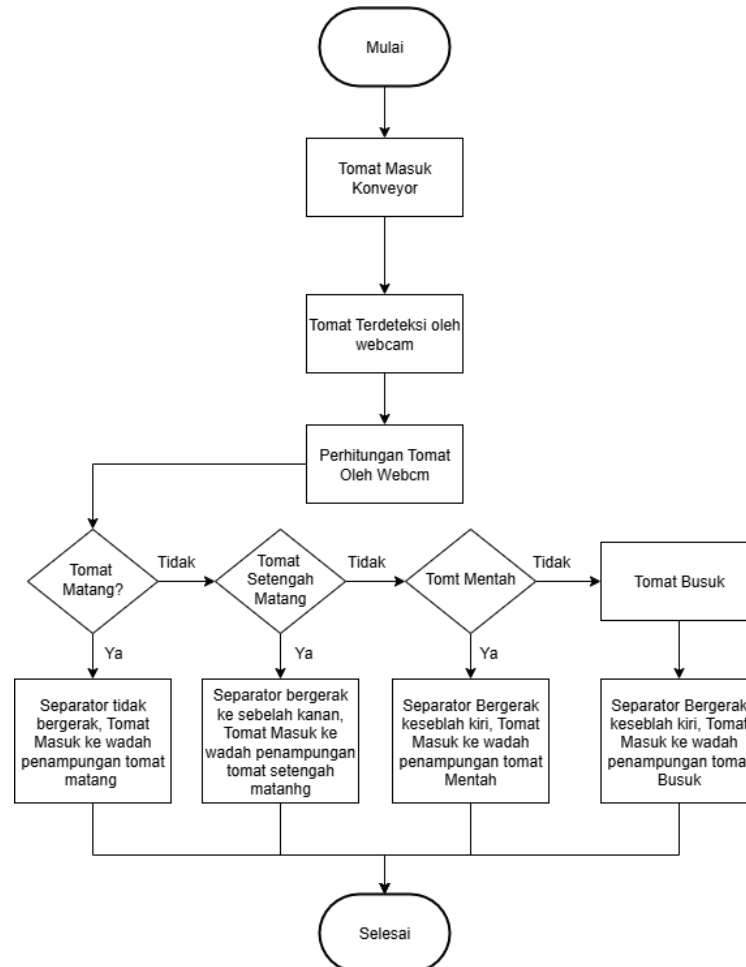
Gambar III.2 Gambar tersebut merupakan desain perancangan umum dari sistem mesin penyortir warna tomat cherry berbasis teknologi *computer vision*, yang dirancang untuk mengotomatiskan proses pemisahan tomat berdasarkan tingkat kematangan, yaitu matang, setengah matang, busuk dan mentah. Proses diawali saat

user mengaktifkan sistem melalui antarmuka GUI berbasis Tkinter yang dijalankan. Gambar dari webcam selanjutnya diproses secara real-time menggunakan OpenCV, dengan dukungan model deteksi objek seperti YOLOv8 untuk mengenali dan mengklasifikasikan tomat berdasarkan warna dan kematangannya. Hasil dari proses ini mencakup tomat matang, mentah. Informasi ini kemudian ditampilkan secara langsung melalui GUI Tkinter, dan dapat disimpan oleh user dalam bentuk file Excel untuk keperluan dokumentasi atau analisis, menggunakan aplikasi Microsoft Excel. Selain itu, hasil klasifikasi juga digunakan untuk memberi perintah ke mikrokontroler Arduino untuk output tegangan melalui komunikasi serial.

### **III.3.2 Diagram Alir**

Diagram alir merupakan representasi dari langkah-langkah penyelesaian suatu masalah yang disusun secara berurutan dan digambarkan menggunakan simbol-simbol khusus. Diagram ini menggambarkan alur logika yang membentuk suatu program. Diagram alir (flowchart) adalah diagram yang menggunakan simbol atau gambar tertentu beserta garis alur untuk menggambarkan urutan langkah-langkah dari awal hingga akhir dalam suatu proses, kegiatan, atau program[26].

### III.3.2.1 Diagram alir proses sistem

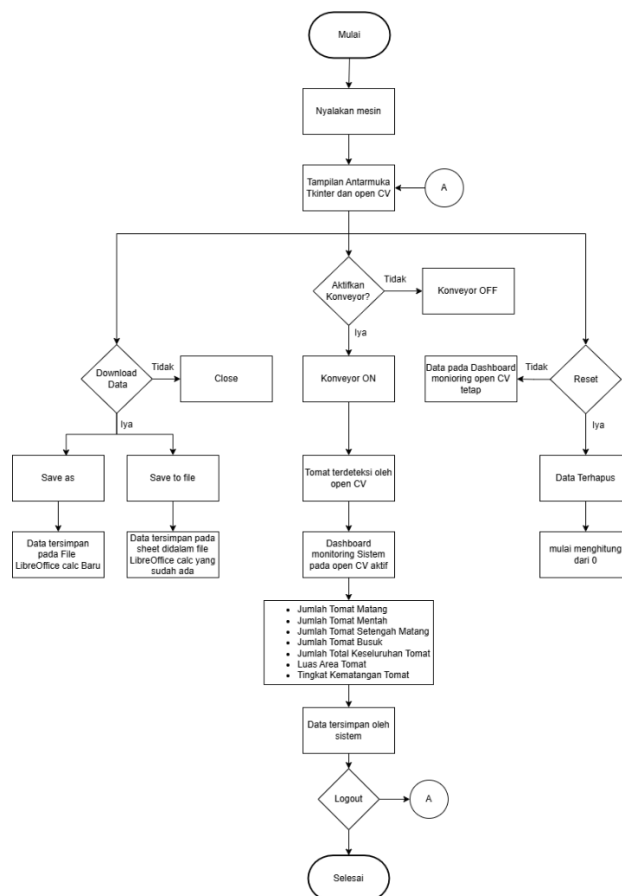


Gambar III.3 Diagram alir proses sistem

Gambar III.3 merupakan flowchart proses kerja sistem penyortiran otomatis tomat cherry berdasarkan tingkat kematangan dengan bantuan webcam OpenCV dan aktuator. Proses dimulai ketika tomat masuk ke atas konveyor kemudian webcam menangkap citra tomat dan sistem mendeteksinya secara otomatis untuk melakukan penghitungan jumlah serta analisis warna tomat melalui pemrosesan citra. Deteksi ini dilakukan untuk mengklasifikasikan tomat berdasarkan tingkat kematangan yang meliputi kategori matang, setengah matang, mentah, dan busuk.

Setelah proses klasifikasi dilakukan sistem akan menentukan pergerakan separator sesuai dengan hasil identifikasi tomat. Jika tomat terdeteksi matang maka separator tidak bergerak dan tomat diarahkan langsung oleh konveyor menuju wadah tomat matang. Jika tomat terdeteksi setengah matang maka separator bergerak ke kanan untuk mengarahkan tomat ke wadah setengah matang. Jika tomat terdeteksi mentah maka separator bergerak ke kiri untuk mengarahkan tomat ke wadah mentah. Sedangkan jika tomat tidak termasuk dalam tiga kategori tersebut maka diklasifikasikan sebagai tomat busuk dan separator juga akan bergerak ke kiri untuk mengarahkannya ke wadah tomat busuk.

**III.3.2.2 Diagram Alir Antarmuka**



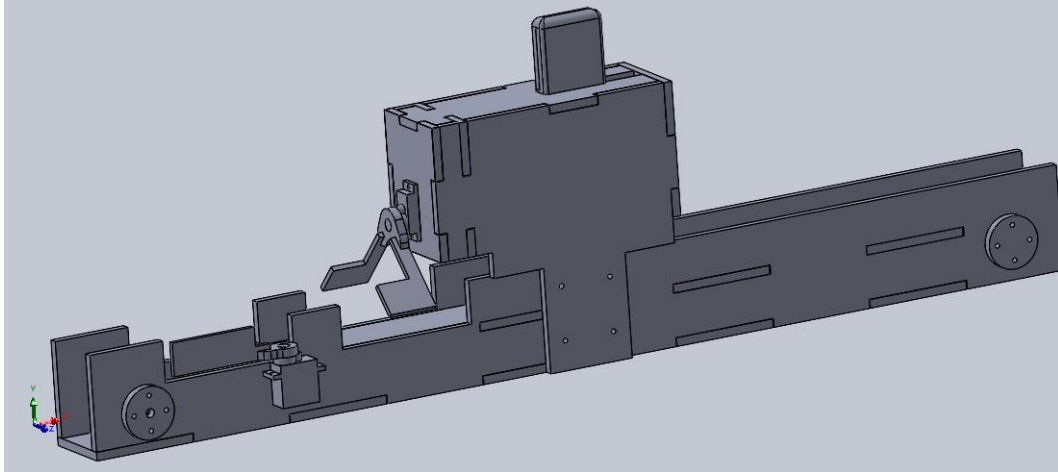
Gambar III.4 Diagram alir Antarmuka Tkinter dan Open CV

Gambar III.4 menunjukkan diagram alir (flowchart) sistem antarmuka berbasis Tkinter dan OpenCV yang berfungsi untuk mengatur proses pendeteksian, pemantauan, penyimpanan, dan pengunduhan data tomat pada mesin penyortir otomatis. Proses diawali dengan menyalakan mesin, yang kemudian akan menjalankan program utama dan menampilkan antarmuka grafis pengguna berbasis Tkinter serta jendela pemantauan berbasis OpenCV. Tampilan ini menjadi pusat kontrol dan monitoring sistem, memungkinkan pengguna untuk mengatur konveyor dan mengamati proses klasifikasi tomat secara *real-time*.

Setelah antarmuka aktif, pengguna dapat memilih untuk menghidupkan atau mematikan konveyor melalui tombol yang tersedia. Jika konveyor diaktifkan, maka kamera akan mulai mendeteksi tomat yang melintas, dan sistem secara otomatis mengklasifikasikan tomat berdasarkan warna dan ukuran menggunakan algoritma deteksi berbasis OpenCV. Data yang diperoleh dari proses ini ditampilkan langsung pada dashboard sistem, meliputi jumlah tomat matang, mentah, setengah matang, busuk, total keseluruhan tomat, luas area tomat, serta tingkat kematangannya.

Selanjutnya, sistem menyimpan hasil klasifikasi secara otomatis, dan pengguna memiliki opsi untuk mengunduh data tersebut melalui fitur Download Data, baik dengan membuat file baru maupun menambahkan data ke file yang telah ada. Terdapat pula fitur Reset, yang berfungsi menghapus seluruh data deteksi sebelumnya dan memulai penghitungan dari nol. Jika proses telah selesai, pengguna dapat menekan tombol Logout, yang akan menutup antarmuka dan mengakhiri seluruh proses sistem secara otomatis.

### III.3.3 Perancangan Mekanik



Gambar III.5 Perancangan Mekanik mesin penyortir warna tomat cherry berbasis computer vision

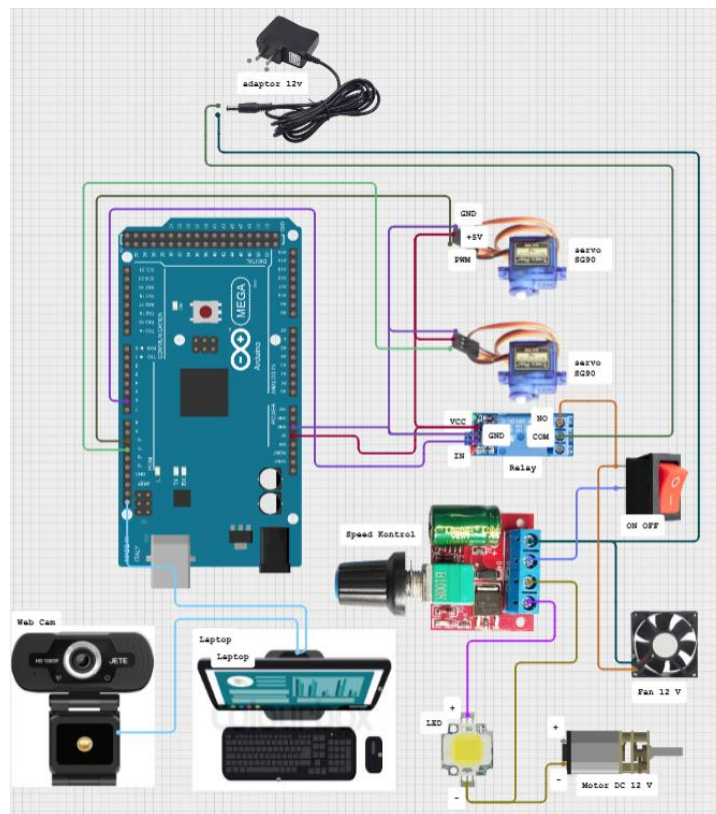
Gambar III.5 merupakan Perancangan Mekanik dari mesin penyortir warna tomat cherry berbasis *computer vision* konveyor di desain berukuran lebar 3.5 Cm untuk memberikan jalur untuk satu persatu tomat berada diatas nya. Konveyor tersebut diberikan ruang kosong pada sisi nya di saat penyortiran agar tomat dapat keluar dari konveyor dan berpindah pada tempat penampungan. Mekanik bekerja berawal dari Bergeraknya konveyor untuk membawa tomat menuju area inspeksi. Di area ini, webcam memindai warna tomat dan mengirimkan data ke sistem kontrol untuk proses penyortiran.

Webcam diletakkan di atas kotak tertutup agar cahaya tidak mempengaruhi proses penangkapan gambar oleh webcam. Sistem kontrol kemudian memproses data tersebut untuk menentukan apakah tomat masuk dalam kategori matang, mentah, busuk atau setengah matang. Berdasarkan keputusan ini, jika tomat mentah maka mekanik dari separator satu berputar ke ke kiri dan memindahkan tomat masuk ke wadah penampungan, jika tomat setengah matang di geser ke kanan oleh separator satu, jika tomat busuk maka separator dua akan berputar ke sebelah kiri dan jika

tomat matang maka kedua separator akan diam agar tomat lolos dan menuju wadah penampungannya.

Desain ini bertujuan meningkatkan efisiensi dan akurasi dalam proses penyortiran tomat secara otomatis, sehingga dapat mengurangi kesalahan manual dan mempercepat proses produksi. Sistem ini juga dapat diintegrasikan dengan teknologi canggih lainnya untuk memantau data secara *real-time* dan menyesuaikan kriteria penyortiran sesuai kebutuhan.

### III.3.4 Perancangan Elektrik



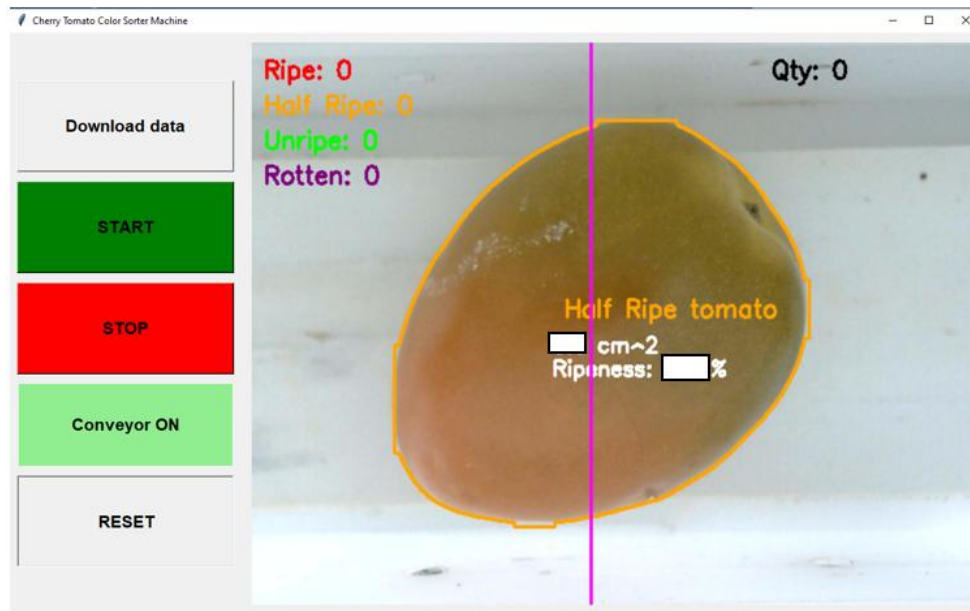
Gambar III. 6 Perancangan elektrik mesin penyortir warna tomat cherry berbasis *computer vision*

Gambar III.4 menunjukkan rangkaian sistem elektrik dari mesin penyortir tomat cherry berbasis computer vision yang mengintegrasikan Arduino Mega 2560, kamera webcam, motor DC, dua buah servo motor SG90, modul relay, pengatur kecepatan motor (speed control), dan lampu LED 12V. Proses dimulai saat webcam yang terhubung ke laptop atau mini PC menangkap gambar tomat yang berjalan di atas konveyor. Gambar tersebut dianalisis secara real-time menggunakan OpenCV dan model deteksi objek dari Python di dalam laptop. Hasil analisis ini menentukan klasifikasi tomat, yang nantinya mengontrol pergerakan servo untuk proses sortir. Motor DC 12V berfungsi sebagai penggerak konveyor. Motor ini diatur kecepatannya menggunakan modul speed control dan diaktifkan melalui modul relay yang dikendalikan oleh pin digital Arduino Mega. Modul relay bekerja sebagai saklar otomatis yang menghubungkan sumber daya 12V dari adaptor ke motor.

Saat relay aktif, lampu LED 12V menyala bersamaan sebagai pencahayaan tambahan untuk meningkatkan akurasi deteksi kamera. Aktifnya relay juga mengaktifkan kipas 12V untuk mendinginkan controller, agar controller tidak cepat rusak jika terlalu panas. Sistem penyortiran dilakukan dengan dua buah servo SG90 yang dikendalikan langsung oleh Arduino Mega melalui sinyal PWM. Servo akan bergerak ke arah tertentu berdasarkan sinyal dari laptop sesuai klasifikasi tomat (matang, mentah, setengah matang, atau busuk). Arduino juga menerima data klasifikasi dari laptop melalui komunikasi serial, dan mengatur pergerakan servo secara otomatis. Keseluruhan sistem ini bekerja secara terintegrasi, mulai dari deteksi, klasifikasi, penyortiran, hingga kontrol motor dan pencahayaan, sehingga memungkinkan proses penyortiran tomat berlangsung secara otomatis, akurat, dan efisien tanpa intervensi manual.

### III.3.5 Perancangan Informatika

#### III.3.5.1 Perancangan Tampilan User Interface



Gambar III.7 Tampilan Open CV dan *interface* pada Tkinter

Gambar III.8 merupakan rancangan Tampilan Open CV dan *interface* pada Tkinter. Pada tampilan Open CV terdapat pendeteksian jenis kematangan tomat dengan ditandai dengan garis yang menyelimuti warna tomat atau disebut garis segmentasi. Untuk warna tomat matang berwarna merah, untuk tomat mentah berwarna hijau dan untuk tomat setengah matang berwarna orange. Didalam garis segmentation terdapat jenis kematangan tomat dan pengukuran luas. Di atas open CV terdapat rekapitulasi jumlah tomat sesuai jenis dan total keseluruhan. Garis biru di tengah open CV berfungsi sebagai garis penghitung jika melewatinya.

Di sisi kiri GUI terdapat beberapa tombol fungsional yang dapat digunakan untuk mengontrol sistem secara manual. Tombol "Download data" untuk mengunduh hasil deteksi ke file, "START" untuk memulai menjalankan konveyor, "STOP" untuk menghentikan konveyor sementara, "konveyor OFF/ON" memberikan indikator konveyor sedang OFF atau ON, dan "RESET" untuk mengulang counting

menjadi 0 lagi. Tampilan ini menggambarkan kemampuan sistem dalam mengintegrasikan pemrosesan visual, analisis objek, serta kontrol perangkat keras dalam satu antarmuka yang terstruktur dan informatif.

### III.3.5.2 Perancangan Tabel Data

Tabel III.1 Tampilan tabel data tomat pada Microsoft Excel

NO	REAL TIME DETECT	COLOR TOMATO	DIMENTION TOMATO (Cm <sup>2</sup> )	RIPENESS RASIO TOMATO (%)
1	Y/M/D H:M:S	Rotten tomato	... Cm <sup>2</sup>	.... %
2	Y/M/D H:M:S	Half Ripe tomato	... Cm <sup>2</sup>	.... %
3	Y/M/D H:M:S	Ripe tomato	... Cm <sup>2</sup>	.... %
4	Y/M/D H:M:S	Unripe tomato	... Cm <sup>2</sup>	.... %
5	Y/M/D H:M:S	Unripe tomato	... Cm <sup>2</sup>	.... %

Color Tomato	Qty Tomato	Average Ripeness Ratio
Ripe	....	.... %
Unripe	....	.... %
Half ripe	....	.... %
Rotten	....	
Qty Tomatoes	....	

Tabel III.1 Merupakan tampilan perancangan tabel data di dalam aplikasi Microsoft Excel, data tomat diperoleh dari download yang berada di interface tkiner. Tabel tersebut merupakan perencanaan input data dari rancang bangun mesin penyortir tomat cherry berbasis computer vision. Terdapat data tomat yang masing-masing dicatat, waktu deteksi secara real-time, kematangan tomat, dan dimensi luasnya dalam satuan cm<sup>2</sup>, rasio kematangan. Rekapitulasi total banyaknya tomat, dan rata rata rasio dari semua tomat dalam jenis tomat tertentu. Data ini menunjukkan bahwa sistem berhasil mendeteksi warna, jumlah dan ukuran tomat secara otomatis dan akurat berdasarkan citra kamera.

## III.4 Pengujian dan Pemeliharaan

### III.4.1 Skema Pengujian Sistem

Skema Pengujian Sistem merujuk pada strategi yang direncanakan untuk menguji sistem yang telah dikembangkan dalam suatu penelitian atau proyek. Tujuannya

adalah untuk memverifikasi bahwa sistem tersebut beroperasi sesuai dengan spesifikasi yang diinginkan, serta untuk mengidentifikasi dan mengatasi potensi masalah yang mungkin timbul. Skema ini mencakup penetapan tujuan pengujian, pemilihan metode pengujian yang sesuai, pengembangan skenario uji yang representatif, serta prosedur untuk mengumpulkan dan menganalisis data uji. Pentingnya skema ini terletak pada validasi yang komprehensif terhadap kualitas dan kinerja sistem sebelum diimplementasikan, sehingga memastikan keefektifan dan kehandalan sistem dalam penggunaan praktis

Tabel III.2 Skema pengujian sistem









No	Pengujian	Fungsi
1	Pengujian Elektrik dan Mekanik	Memeriksa fungsi rangkaian serta memastikan bahwa motor servo bergerak sesuai dengan kategori cacat yang terdeteksi.
2	Pengujian Pengaruh Ketinggian Kamera	Pengujian pengaruh ketinggian kamera terhadap deteksi benda dilakukan untuk memahami bagaimana variasi posisi kamera memengaruhi kemampuan sistem dalam mengenali dan memproses objek yang terdeteksi.
3	Pengujian Deteksi Menggunakan Penutup	Pengujian deteksi menggunakan penutup bertujuan mengevaluasi seberapa baik sistem dapat mendeteksi dan mengenali objek dalam kondisi yang menghalangi mengaburkan penglihatan.




4	Pengujian data yang ditampilkan di User Interface	Pengujian data yang ditampilkan dari server ke User Interface berfungsi untuk memastikan bahwa informasi yang disajikan kepada pengguna akurat, up-to date, dan mudah dipahami, sehingga pengguna dapat mengambil keputusan yang tepat berdasarkan data tersebut.
5	Pengujian Fitur di User Interface	Pengujian perangkat lunak, khususnya fungsi-fungsi layar yang ada pada UI (User Interface), berfungsi untuk memastikan bahwa setiap elemen dan fitur tampilan berfungsi dengan benar, memberikan pengalaman pengguna yang optimal dan bebas dari kesalahan
6	Pengujian Download Data	Pengujian Download data bertujuan memastikan data yang ditampilkan pada user interface dapat di download dan ditampilkan pada aplikasi serta sesuai antara data pada user interface dengan data pada aplikasi.

#### III.4.2 Skema Pengujian Objek

Objek yang akan digunakan dalam penelitian ini adalah tomat *cherry*. Penelitian ini akan fokus pada pendeteksian berbagai jenis kematangan dan warna tomat. Beberapa jenis tomat *cherry* yang akan diidentifikasi dan dianalisis meliputi:

Tabel III.3 Klasifikasi Tomat

No	Klasifikasi Jenis Kematangan Tomat Cherry	Gambar Tomat Cherry
1	Tomat Matang	
2	Tomat Matang	
3	Tomat Matang	
4	Tomat Matang	
5	Tomat Setengah Matang	
6	Tomat Setengah Matang	
7	Tomat Setengah Matang	
8	Tomat Setengah Matang	

9	Tomat Mentah	
10	Tomat Mentah	
11	Tomat Mentah	

#### III.4.2 Pengelompokan jenis warna tomat

Tomat cherry dalam sistem ini diklasifikasikan menjadi empat kategori berdasarkan tingkat kematangan dan kondisi fisiknya, yaitu tomat matang, tomat setengah matang, tomat mentah, dan tomat busuk. Tomat matang memiliki warna dari merah terang hingga merah tua merata di seluruh permukaan kulit, menandakan bahwa tomat telah mencapai kematangan penuh dan siap dikonsumsi. Tomat setengah matang menunjukkan gradasi warna dari hijau kekuningan hingga kemerahan sebagian, di mana sebagian permukaan masih berwarna hijau atau kuning sementara bagian lainnya mulai memerah. Tomat mentah ditandai dengan warna hijau pekat atau hijau muda secara menyeluruh yang menunjukkan bahwa tomat belum mengalami proses pematangan. Sedangkan tomat busuk umumnya memiliki warna tidak normal seperti coklat gelap, kehitaman, abu-abu kusam, atau menunjukkan bercak-bercak aneh yang dapat disertai dengan perubahan tekstur dan tanda-tanda pembusukan lainnya