



**Sistem Pendeteksi Objek Berbasis Ruangan  
Warna YCbCr untuk *Human Machine Interface*  
Menggunakan Raspberry Pi**

**Tugas Akhir**

**Oleh:  
Naufal Hanif Marzukie (4212011001)**

**Program Studi Teknik Mekatronika  
Jurusan Teknik Elektro  
Politeknik Negeri Batam  
2024**

## Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan di bawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul: “*Human Machine Interface Menggunakan Raspberri PI Pada Mesin Deteksi Warna Menggunakan Metode Ruang Warna YCbCr*” adalah **hasil karya sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri**. Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Batam, 24 Juni 2024



---

Arizul pajri  
NIM: 4212011002



---

Naufal Hanif Marzukie  
NIM: 4212011001



---

Rifdah Nur Fauzlyah  
NIM: 4212011007

# Lembar Pengesahan

Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar  
Sarjana Terapan Teknik (S.Tr.T)  
di  
Politeknik Negeri Batam

Oleh:  
Naufal Hanif Marzukie (4212011001)  
Arizul Pajri (4212011002)  
Rifdah Nur Fauziyah (4212011007)

Tanggal Sidang: 17/07, 2024

Disetujui oleh:



1. Diono, S.Tr. T., M.Sc  
NIK: 120243



1. Ir. Indra Hardian Mulyadi, S.T.,  
M.Eng., Ph.D  
NIK:117179



2. Muhammad Naufal Airlangga  
Diputra, S.Pd.  
NIK: 122281

# [*Human Machine Interface* Menggunakan Raspberry Pi Pada Mesin Deteksi Warna Menggunakan Metode Ruang Warna YCbCr]

## Abstrak

Dengan perkembangan ilmu pengetahuan dan teknologi, banyak alat yang mempermudah pekerjaan manusia. Salah satu bidang yang mendukung hal ini adalah *computer vision*. Penelitian ini bertujuan menjadikan Raspberry Pi sebagai pilihan dalam pembuatan HMI (*Human Machine Interface*) pada mesin. Metode yang digunakan untuk pelacakan objek adalah ruang warna YCbCr. Raspberry Pi adalah komputer papan tunggal. Penelitian ini membangun aplikasi GUI (*Graphic User Interface*) yang menampilkan hasil pelacakan objek menggunakan metode YCbCr. YCbCr terdiri dari komponen luminansi yang merepresentasikan warna RGB, serta komponen Cb dan Cr yang merepresentasikan krominansi, yaitu corak warna dan saturasi. GUI pada HMI menggunakan *Windows form* dengan bahasa C#. Hasil pelacakan objek memberikan nilai input pada PLC sebagai pusat kendali mesin, menggunakan komunikasi TCP/IP. Selain mengendalikan, HMI juga menampilkan pelacakan objek dan menyimpan data jumlah produk terlacak. Ini memudahkan pengguna dalam mengoperasikan dan mendapatkan data terkait produk yang dikerjakan. Cara kerja mesin dan HMI ini adalah pengguna mengatur warna yang akan dilacak. Jika kamera mendeteksi objek dengan warna yang sesuai, mesin melanjutkan proses. Jika tidak, mesin tidak memproses objek ke tahap selanjutnya. Ini memastikan warna objek yang terdeteksi sesuai dengan yang diinginkan pengguna, memudahkan proses sortir warna objek yang sedang dikerjakan. Hasil penelitian ini menunjukkan bahwa Raspberry Pi dapat digunakan untuk merancang mesin otomatisasi. Penelitian ini menggunakan Windows pada Raspberry Pi karena Raspbian tidak mendukung *webcam* yang digunakan. Namun, penggunaan Windows menyebabkan penggunaan memori mencapai batas maksimum, mengurangi efisiensi Raspberry Pi sebagai HMI. Jika HMI tidak memerlukan kamera, Raspbian dapat digunakan untuk mengurangi penggunaan memori dibandingkan dengan Windows.

Kata kunci: HMI, PLC, Raspberry, YcbCr

# **[HUMAN MACHINE INTERFACE USING RASPBERRY PI ON COLOR DETECTION MACHINE USING YCbCr COLOR SPACE METHOD]**

## **Abstract**

*With the advancement of science and technology, many tools facilitate human work. One field that supports this is computer vision. This research aims to make Raspberry Pi a viable option for creating HMI (Human Machine Interface) on machines. The method used for object tracking is the YCbCr color space. Raspberry Pi is a single-board computer. This research develops a GUI (Graphic User Interface) application that displays the results of object tracking using the YCbCr method. YCbCr consists of a luminance component that represents RGB colors, as well as Cb and Cr components that represent chrominance, which includes hue and saturation. The HMI GUI uses Windows Forms with the C# language. The object tracking results provide input values to the PLC, the central control unit of the machine, using TCP/IP communication. In addition to controlling the machine, the HMI also displays object tracking and stores data on the number of tracked products. This facilitates users in operating and obtaining data related to the products being processed. The working mechanism of the machine and HMI is that the user sets the color to be tracked. If the camera detects an object with the appropriate color, the machine proceeds with the process. If not, the machine does not process the object to the next stage. This ensures that the detected object's color matches the user's requirements, simplifying the color sorting process for the objects being worked on. The results of this study show that Raspberry Pi can be used to design automation machines. This research uses Windows on Raspberry Pi because Raspbian does not support the webcam used. However, using Windows causes memory usage to reach the maximum limit, reducing the efficiency of Raspberry Pi as an HMI. If the HMI does not require a camera, Raspbian can be used to reduce memory usage compared to Windows.*

*Keywords: HMI, PLC, Raspberry, YCbCr*

## Kata Pengantar

Puji Syukur kami panjatkan kehadiran Allah SWT, Tuhan yang maha esa, yang telah memberikan Rahmat dan hidayah-Nya. Sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul “Human Machine Interface Menggunakan Raspberry Pi pada Mesin Deteksi Warna Menggunakan Metode Ruang Warna YCbCr”. Shalawat berangkaikan salam semoga senantiasa tercurahkan kepada nabi besar Muhammad SAW, yang telah menjadi suri tauladan bagi umat manusia. Penulisan laporan akhir ini memiliki tujuan untuk melengkapi persyaratan kelulusan tingkat Diploma IV Program Studi Mekatronika, Jurusan Teknik Elektro, Politeknik Negeri Batam.

Pada kesempatan ini penulis menyampaikan terima kasih yang sebesar-besarnya atas segala bantuan dan dukungan yang telah diberikan selama penyusunan Tugas Akhir ini hingga selesai kepada:

1. Kedua orang tua dan keluarga yang telah memberikan doa, semangat dan motivasi.
2. Bapak Uuf Brajawidagda, S.T., M.T., Ph.D. selaku Direktur Politeknik Negeri Batam.
3. Bapak Dr.Budi Sugandi, S.T., M.Eng. selaku ketua Jurusan Teknik Elektro Politeknik Negeri Batam sekaligus wali dosen kelas A malam.
4. Bapak Indra Hardian Mulyadi, S.T., M.Eng. selaku ketua Program Studi Mekatronika Politeknik Negeri Batam dan dosen pembimbing tugas akhir.
5. Bapak Muhammad Naufal Airlangga Diputra, S.Pd., M.P.H. selaku dosen pengampu mata kuliah Tugas Akhir.
6. Bapak Budi Sugandi, S.T., M.Eng. selaku wali dosen kelas A malam.
7. Bapak Diono, S.Tr.T., M.Sc dan Bapak Muhammad Naufal Airlangga Diputra, S.Pd., M.P.H. selaku dosen penguji.
8. Seluruh dosen-dosen Jurusan Teknik Elektro Politeknik Negeri Batam.
9. Seluruh teman-teman yang telah membantu proses penyelesaian Tugas Akhir ini.

Penulis menyadari dalam penyusunan Laporan Tugas Akhir ini masih banyak kekurangan, oleh karena itu penulis mengharapkan kritik dan saran yang membangun demi kesempurnaan Tugas Akhir ini. Semoga Laporan Tugas Akhir ini dapat bermanfaat dan berguna bagi kita semua. Terima kasih atas segala perhatiannya.

Batam, 26 Juni 2024

Naufal Hanif Marzukie  
Arizul Pajri  
Rifdah Nur Fauziyah

# Daftar Isi

Pernyataan Keaslian Tugas Akhir .....	i
Lembar Pengesahan .....	ii
Abstrak .....	iii
<i>Abstract</i> .....	iv
Kata Pengantar .....	v
Daftar Isi .....	vii
Daftar Gambar .....	ix
Daftar Tabel .....	xi
Bab 1. Pendahuluan .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan .....	2
1.4. Manfaat .....	3
1.5. Batasan .....	3
1.6. <i>Work Breakdown Structure</i> .....	3
Bab 2. Tinjauan Pustaka .....	5
2.1. Penelitian Terkait .....	5
2.2. Dasar Teori .....	19
2.2.1. Raspberry Pi .....	19
2.2.2. LCD .....	20
2.2.3. PLC TM241CE40R .....	21
2.2.4. Visual Studio 2022 .....	22
2.2.5. Ruang warna YCbCr .....	22
2.2.6. Komunikasi Modbus TCP/IP .....	25
2.2.7. <i>Webcam / Kamera</i> .....	31
2.2.8. <i>Operation Windows System</i> .....	32
Bab 3. Metodologi .....	33
3.1. Perancangan .....	33

3.1.1. Perancangan Desain <i>Graphic User Interface</i> (GUI) .....	37
3.1.2. Perancangan Mekanikal .....	46
3.1.3. Perancangan Elektrikal .....	51
3.2. Alat dan Bahan .....	56
3.3. Pengujian.....	57
3.3.1. Pengujian kamera terhadap benda .....	57
3.3.2. Pengujian komunikasi HMI & PLC.....	59
3.3.3. Pengujian <i>delay</i> waktu kamera.....	60
3.3.4. Pengujian data yang diolah oleh HMI.....	61
Bab 4. Hasil dan Pembahasan .....	62
4.1. Data Pengujian kamera terhadap benda.....	62
4.2. Data Pengujian komunikasi HMI & PLC .....	69
4.3. Data Pengujian <i>delay</i> waktu kamera .....	73
4.4. Data pengujian yang diolah oleh HMI .....	80
Bab 5. Kesimpulan dan Saran .....	86
5.1. Kesimpulan .....	86
5.2. Saran .....	87
Daftar Pustaka .....	88
Biodata .....	89
Lampiran .....	90

## Daftar Gambar

Gambar 1. Gambar Raspberry Pi.....	19
Gambar 2. Gambar LCD.....	20
Gambar 3. Gambar PLC.....	21
Gambar 4. Gambar <i>icon</i> visual studio.....	22
Gambar 5. Gambar konversi RGB ke YCbCr.....	22
Gambar 6. Gambar pengujian nilai YCbCr pada objek berwarna putih.....	23
Gambar 7. Gambar pengujian nilai YCbCr pada objek berwarna biru.....	24
Gambar 8. Gambar pengujian nilai YCbCr pada objek berwarna merah.....	24
Gambar 9. Gambar <i>icon</i> visual studio.....	26
Gambar 10. Konfigurasi IP Address PLC.....	28
Gambar 11. Konfigurasi IP Address modul wago.....	28
Gambar 12. <i>Setting</i> IP Address.....	29
Gambar 13. Gambar <i>webcam</i> .....	31
Gambar 14. Gambar <i>Windows 10 in Raspberry Pi</i> .....	32
Gambar 15. diagram alur.....	33
Gambar 16. Alur kerja mesin.....	35
Gambar 17. Ukuran <i>Size Form</i> Utama.....	38
Gambar 18. Tampilan awal GUI.....	39
Gambar 19. tampilan menu <i>login</i> .....	40
Gambar 20. Tampilan menu Autorun.....	41
Gambar 21. Tampilan menu Autorun.....	43
Gambar 22. Mesin tampak seluruhnya.....	46
Gambar 23. Mesin tampak diperbesar.....	46
Gambar 24. Mesin tampak samping.....	47
Gambar 25. Dimensi mesin.....	47
Gambar 26. <i>Sub-assembly</i> mesin.....	48
Gambar 27. HMI tampak seluruhnya.....	48
Gambar 28. HMI tampak belakang.....	49
Gambar 29. HMI tampak samping.....	49
Gambar 30. HMI tampak samping.....	50
Gambar 31. Dimensi <i>cover</i> HMI.....	50
Gambar 32. <i>Sub-assembly cover</i> HMI.....	51
Gambar 33. <i>Wiring diagram power-1</i> .....	51
Gambar 34. <i>Wiring diagram power-2</i> .....	52
Gambar 35. Topologi jaringan TCP/IP mesin.....	52
Gambar 36. <i>Wiring diagram field input</i> .....	53
Gambar 37. <i>Wiring diagram field output-1</i> .....	54
Gambar 38. <i>Wiring diagram field output-2</i> .....	55
Gambar 39. <i>Wiring diagram connection pin Raspberry to LCD</i> .....	56

Gambar 40. Pengujian menggunakan HMI.....	57
Gambar 41. Pengujian menggunakan laptop .....	58
Gambar 42. Pengujian waktu <i>delay</i> pembacaan kamera .....	60
Gambar 43. Pengujian <i>history alarm</i> pada mesin .....	81
Gambar 44. Pengujian <i>pop up alarm error connecting</i> .....	81
Gambar 45. Pengujian pop up akses terblokir .....	82
Gambar 46. Pengujian <i>history alarm</i> .....	82
Gambar 47. Pengujian <i>counter</i> .....	83

## Daftar Tabel

Tabel 1. Tabel <i>Work Breakdown Structure</i> (WBS) .....	3
Tabel 2. Tabel Literature Review Matrix.....	18
Tabel 3. Tabel Komunikasi Modbus .....	26
Tabel 4. Tabel alamat memori PLC.....	27
Tabel 5. Tabel <i>user guide</i> .....	40
Tabel 6. Tabel <i>user guide login</i> .....	41
Tabel 7. Tabel <i>user guide Autorun</i> .....	42
Tabel 8. Tabel <i>user guide MANUAL I/O</i> .....	44
Tabel 9. Tabel Alat dan Bahan .....	56
Tabel 10. Tabel pengujian komunikasi .....	59
Tabel 11. Tabel pengujian warna putih menggunakan HMI Raspberry Pi.....	62
Tabel 12. Tabel pengujian warna putih menggunakan laptop .....	63
Tabel 13. Tabel pengujian warna merah menggunakan HMI Raspberry Pi.....	64
Tabel 14. Tabel pengujian warna merah menggunakan laptop .....	65
Tabel 15. Tabel pengujian warna biru menggunakan laptop .....	66
Tabel 16. Tabel pengujian warna biru menggunakan laptop .....	67
Tabel 17. Tabel pengujian komunikasi HMI dan PLC.....	69
Tabel 18. Tabel kebenaran BCD.....	72
Tabel 19. Tabel <i>delay</i> waktu.....	74
Tabel 20. Pengujian Waktu <i>delay</i> laptop.....	77
Tabel 21. Pengujian Waktu <i>delay</i> Raspberry Pi .....	78
Tabel 22. Pengujian Waktu <i>delay</i> antara laptop dan Raspberry Pi.....	78
Tabel 22. Pengujian perekaman <i>alarm</i> pada mesin .....	84

# Bab 1. Pendahuluan

## 1.1. Latar Belakang

Dengan semakin berkembangnya ilmu pengetahuan dan teknologi semakin banyak alat-alat yang mempermudah pekerjaan manusia. Selain mengurangi tenaga yang dikeluarkan oleh manusia, alat-alat ini juga berguna untuk menghasilkan dan menjalankan sebuah produk dengan cepat dan tepat. Salah satu bidang yang mendukung untuk melakukan hal tersebut adalah computer vision. Salah satu perkembangan dari penggunaan teknik *computer vision* adalah *tracking object* (pelacakan objek). Pelacakan objek bertujuan untuk mendeteksi dan mengikuti posisi dari suatu objek bergerak yang diinginkan. Pelacakan objek banyak dibutuhkan oleh berbagai macam aplikasi *vision based* seperti *human computer interface*, kompresi/komunikasi video dan sistem keamanan. Pelacakan objek mampu mendeteksi objek yang bergerak, *filtering noise*, dan gerakan-gerakan lain yang tidak diperlukan [1].

Banyak cara yang dilakukan agar dapat melakukan pelacakan objek, metode yang digunakan pada penelitian kali ini adalah menggunakan ruang warna YCbCr, Ruang warna YCbCr terbagi atas 2 komponen yaitu komponen luminance yang merepresentasikan warna RGB. Komponen lainnya adalah Cb dan Cr yang merepresentasikan krominansi yang merepresentasikan corak warna dan saturasi. Komponen ini juga mengindikasikan banyaknya corak warna dan saturasi [2].

Untuk dapat menampilkan hasil pelacakan objek tersebut, maka dibutuhkan tempat untuk menampilkan hasil dari proses pelacakan objek yang dimana pada penelitian ini akan menggunakan HMI (*Human Machine Interface*) yang dibuat dari Raspberry Pi 4. Raspberry Pi merupakan serangkaian komputer papan tunggal dengan ukuran sebesar kartu kredit [3]. Hal ini memudahkan pengguna untuk mengintegrasikan ke dalam beberapa proyek dan juga perangkat yang memiliki ruang terbatas. Dalam proyek ini Raspberry Pi akan dijadikan sebuah HMI, hal ini melibatkan semua aspek interaksi dan komunikasi antara pengguna yaitu manusia dan mesin-mesin melalui HMI (*Human machine interface*). sistem lengkap yang terdiri dari pengguna manusia, HMI, dan mesin itu sendiri disebut sebagai sistem *human-machine system* *The term*. HMI sebagai artefak mediasi dalam sistem kompleks untuk menghubungkan otorasi/pengendalian mesin, memiliki kepentingan dalam kesadaran situasi, keandalan, efisiensi, efektivitas, ketahanan, dan keamanan [4]. Raspberry Pi ini akan dihubungkan dengan LCD berukuran 7 inch yang di mana ukuran ini juga mendukung manfaat ukuran dari Raspberry Pi.

Judul penelitian ini diambil berdasarkan pengalaman PBL (*project based learning*) dengan judul *Human Machine Interface Using Raspberry Pi 4* yang dikerjakan pada semester ini, yang dimana HMI pada Raspberry Pi bisa terhubung dengan PLC model TM241CE40R, hasil akhir yang diinginkan adalah hanya sebatas HMI pada Raspberry Pi dapat saling bertukar data atau informasi dengan PLC

menggunakan protokol Modbus TCP/IP. Ada beberapa improvisasi yang akan ditambahkan pada tugas akhir ini yaitu menghubungkan HMI dari Raspberry Pi langsung kepada mesin dengan tujuan untuk dapat mengendalikan, dan memantau kondisi dan proses yang sedang bekerja pada mesin tersebut, termasuk kondisi I/O pada mesin, jejak rekam alarm yang terjadi pada mesin. Mesin yang digunakan juga berdasarkan pengalaman dari PBL semester 4 yaitu CIM: *Station Fabrication Process*, hasil akhir dari PBL ini adalah membuat SCADA menggunakan aplikasi vijeo, Aplikasi ini merupakan aplikasi yang berhubungan dengan model PLC yang digunakan. Pada mesin juga dilakukan improvisasi yaitu menambahkan fungsi kamera untuk dapat melacak warna objek yang akan di proses pada mesin tersebut dan nantinya akan ditampilkan pada HMI yang sedang dikerjakan.

Dalam tugas akhir ini, tujuan penulisan ini bertujuan untuk menganalisis penggunaan Raspberry Pi sebagai Human machine interface pada mesin pendeteksi warna menggunakan metode ruang warna YCbCr. Seperti HMI mampu mengendalikan, mengoperasikan, dan memantau proses kerja yang terjadi pada mesin tersebut.

## 1.2. Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah:

- a. Bagaimana merancang sebuah Raspberry Pi agar dapat menjadi sebuah pusat kendali berupa HMI pada mesin *automation*?
- b. Bagaimana metode ruang warna YCbCr dapat mendeteksi sebuah objek?
- c. Bagaimana menghubungkan Raspberry Pi dengan sistem controller pada mesin otomasi?
- d. Bagaimana cara HMI pada Raspberry Pi dapat mengendalikan ,mengontrol dan mengolah data sebuah mesin?

## 1.3. Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut:

- a. Memprogram Raspberry Pi agar dapat menjadi pusat kendali berupa HMI pada mesin otomasi
- b. Menganalisis sistem pendeteksi warna menggunakan pelacakan objek dengan metode ruang warna YCbCr
- c. Menghubungkan Raspberry Pi dengan sistem kontroler pada mesin yaitu PLC yang ada pada mesin otomasi
- d. Menganalisis bagaimana HMI dapat mengendalikan, mengontrol dan mengolah data pada mesin

## 1.4. Manfaat

Manfaat dari pengerjaan TA ini adalah agar Raspberry Pi dapat digunakan sebagai perancangan sebuah mesin otomasi, karena dilihat dari banyaknya kelebihan yang dimiliki Raspberry Pi ini memiliki potensi lebih agar dapat dijadikan media untuk sebuah HMI.

## 1.5. Batasan

Untuk menjelaskan isi dari proyek ini agar tidak terlalu jauh melebar dari apa yang akan disampaikan, maka akan di buat batasan permasalahan yang akan dibahas. Adapun batasan masalahnya adalah sebagai berikut:

- a. Bahasa pemrograman yang akan digunakan dalam pembuatan GUI (*Graphical User Interface*) adalah Bahasa C#.
- b. Kontroler yang digunakan pada mesin ini adalah PLC modicon Schneider TM241.
- c. Komunikasi antara HMI dan PLC menggunakan Modbus TCP/IP (*Ethernet*).
- d. Parameter yang akan dikontrol adalah alamat yang ada pada GUI dan mesin, data yang diolah oleh HMI dapat di jadikan sebuah informasi untuk operator mesin.

## 1.6. Work Breakdown Structure

*Work Breakdown Structure* (WBS) ditampilkan jika TA dilakukan secara tim. Jelaskan peran masing-masing anggota dalam bentuk tabel (misal Tabel 1) dan teks.

**Tabel 1. Tabel *Work Breakdown Structure* (WBS)**

No	Nama	Tugas dan Tanggung Jawab dalam Tim
1	Naufal Hanif Marzukie	<ul style="list-style-type: none"><li>• Koordinator</li><li>• Pemrograman pelacakan objek berdasarkan warna</li><li>• Analisis dan pengambilan data warna pada objek</li><li>• Pengujian keakuratan pembacaan kamera</li></ul>
2	Arizul Pajri	<ul style="list-style-type: none"><li>• Sistem kendali mesin <i>detection objek fabrication</i> proses</li><li>• Desain mekanikal dan elektrik</li><li>• Komunikasi PLC dan HMI</li></ul>

		<ul style="list-style-type: none"> <li>• Pengujian program PLC dan komunikasi antara PLC,HMI dan mesin</li> </ul>
3	Rifdah Nur Fauziyah	<ul style="list-style-type: none"> <li>• Desain GUI (<i>Graphic User Interface</i>)</li> <li>• Pemrograman GUI pada HMI</li> <li>• Pengujian desain HMI terhadap PLC dan mesin</li> <li>• Pembuatan <i>Log History</i> terhadap <i>alarm</i> yang muncul pada mesin.</li> </ul>

## Bab 2. Tinjauan Pustaka

### 2.1. Penelitian Terkait

Teknologi yang terus berkembang membuat kualitas kehidupan manusia meningkat pada standar yang lebih tinggi. Teknologi otomatisasi yang saat ini berkembang mampu meningkatkan kualitas dan kuantitas produksi di dunia industri. Sistem otomasi adalah sistem yang bekerja dengan menggabungkan sistem mekanik, elektrik, hidrolis dan pneumatik. Umumnya untuk sistem otomasi menggunakan PLC yang terhubung dengan HMI sebagai media pengendali dan pemantauan untuk menggerakkan mesin-mesin sistem pneumatik dan lainnya [5]

Penggunaan teknologi otomatis sekarang ini telah berkembang hingga menjangkau teknologi sistem kendali otomatis berbasis *human machine interface (HMI)*. HMI merupakan perangkat dan sarana yang sangat penting pada suatu pusat sistem pengendalian suatu *plant* yang diperlukan sebagai media komunikasi antara pengguna (*operator*, teknisi, dan *engineer*) dengan komputer untuk memanfaatkan data dari sistem *real time*. Sistem kendali otomatis berbasis HMI memiliki keuntungan yaitu sesuai untuk kawasan lahan yang fleksibel, praktis dan tidak memerlukan tenaga kerja banyak karena menggunakan sistem otomatisasi, investasi lebih murah, lebih mudah dikendalikan karena menggunakan sistem komputerisasi dan lebih efisien dalam waktu pencarian untuk penyelesaian pekerjaan yang lebih cepat[6].

HMI merupakan sistem yang menghubungkan antara manusia dan mesin. HMI dapat berupa pengontrol yang mampu memvisualisasikan status manual, maupun melalui tampilan komputer yang bersifat nyata[5].

Penelitian terkait yang dilakukan oleh Dedy Agung Prabowo, Dedy Abdullah, dan Ari Manik dengan judul “DETEKSI DAN PERHITUNGAN OBJEK BERDASARKAN WARNA MENGGUNAKAN *COLOR OBJECT TRACKING*” yang diterbitkan pada tahun 2018. Penelitian ini dilakukan dengan sistem yang dibuat untuk mendeteksi dan menghitung warna yang telah ditentukan dengan kamera. Hasil dari penelitian tersebut menunjukkan bahwa perbedaan warna objek dan latar belakang sangat menentukan keberhasilan pendeteksian objek. Pencahayaan yang baik untuk pendeteksian objek adalah 31-15.000 lux.[1].

Penelitian terkait yang dilakukan oleh Nabila Yulianda Putri dan Riki Mukhaiyar dengan judul “*Control and Monitoring System Process Handling Production on SMI 4.0 Machines using PLC Controller Wirelessly Based on Human Machine Interface*” yang diterbitkan pada tahun 2022. Penelitian ini dilakukan dengan melakukan pengujian sistem, dan data yang diterima oleh PLC pada HMI seperti *control running station*, perhitungan *input*, *output* dapat memunculkan notifikasi error sehingga mampu mempersingkat waktu dalam analisis *troubleshoot* pada mesin produksi[7].

Penelitian terkait yang dilakukan oleh Bilal Naji Alhasnawi dan Basil H. Jasim dengan judul “*SCADA Controlled Smart Home Using Raspberry Pi3*” yang diterbitkan pada tahun 2019. Sistem rumah pintar yang dikendalikan dari jarak jauh memenuhi kebutuhan masyarakat. Pengoperasian kendali jarak jauh dilakukan oleh PC (*Personal Computer*) yang didukung oleh IGSS SCADA. Di PC dengan protokol TCP/IP bertindak sebagai pemancar, yang mengirimkan perintah ke modul Raspberry Pi3. Raspberry Pi3 menerima perintah dari SCADA. Raspberry Pi3 mengirimkan perintah dengan alamat ke unit terminal (papan Wemos-D1) melalui MQTT. Setiap unit terminal menerima alamat dari Raspberry Pi3, jika alamat unit terminal cocok dengan Raspberry Pi3, ia dapat mengontrol perangkat atau parameter yang diukur, kemudian mengirimkan parameter ini beserta alamatnya ke Raspberry Pi3. Raspberry Pi3 mengirimkan data ini ke PC pemilik. [5].

Penelitian terkait yang dilakukan oleh Helmy Rahadian dan M Ary Heryanto dengan judul “Pengembangan Human Machine Interface (HMI) pada *Simulator Sortir Bola* sebagai Media Pembelajaran Otomasi Industri” yang diterbitkan pada tahun 2020. Penelitian ini melibatkan *Programmable Logic Controller* (PLC) dan *Human Machine Interface* (HMI). HMI menjadi bagian penting dalam sistem otomasi industri. PLC merupakan piranti yang dapat diprogram untuk melakukan fungsi pengendalian pada sistem yang sederhana maupun kompleks. Di sisi lain, HMI berperan dalam membangun interaksi yang *user friendly* antara manusia sebagai pengendali dengan mesin. HMI digunakan untuk memonitor dan memahami jalannya suatu proses. HMI memiliki kapabilitas untuk memberikan notifikasi berupa *alarm audio* maupun *visual* apabila dalam proses produksi terjadi situasi yang tidak normal [8].

**Tabel 2. Tabel Literature Review Matix**

<i>Author/ Date</i>	<i>Theoretical/ Conceptual Framework</i>	<i>Research Question(s)/ Hypotheses</i>	<i>Methodology</i>	<i>Analysis &amp; Results</i>	<i>Conclusions</i>	<i>Implications for Future research</i>	<i>Implications For practice</i>
Nabila Yulianda Putri dan Riki Mukhaiyar (2022)	Perancangan <i>hardware</i> dan <i>software</i> dengan blok diagram menjelaskan kerja struktur sistem secara keseluruhan, serta komunikasi perangkat, pemrograman PLC dan HMI. Diharapkan bahwa operator produksi dapat menggunakan metode yang lebih efisien untuk menyelesaikan masalah pada proses produksi, yang akan mempercepat waktu target produksi.	Operator industri biasanya menggunakan saklar tombol tekan untuk mengontrol dan memantau proses industri, yang berarti mereka harus memastikan bahwa operator selalu ada di tempat selama proses produksi berlangsung. Selain itu, bagaimana meningkatkan kemudahan mendapatkan informasi untuk melakukan monitoring dan	Penelitian ini dilakukan menggunakan metode kuantitatif. Tahapan penelitian dimulai dengan penjelasan tentang konsep dan prinsip kerja sistem yang dirancang dan digambarkan dalam blok diagram. Selain itu, penelitian ini juga menjelaskan perancangan <i>hardware</i> dan <i>software</i> dengan memberikan penjelasan singkat tentang pemrograman PLC, komunikasi	Setelah data dikumpulkan, analisis data dilakukan untuk menilai hasil penelitian. Dalam pengujian sistem, PLC mengirimkan data ke HMI seperti kontrol running station, perhitungan input, dan output, dan memunculkan notifikasi error. Ini membuat	Diharapkan hasil penelitian ini akan membantu operator produksi menyelesaikan masalah dalam proses produksi yang akan mempercepat waktu target produksi.	Penelitian ini diharapkan dapat membantu operator produksi menyelesaikan masalah dalam proses produksi, yang akan mempercepat waktu yang ditargetkan.	Penggunaan sistem kontrol dan monitoring pada mesin produksi dengan PLC dan HMI dapat meningkatkan efisiensi kerja pada proses produksi dengan menambahkan perangkat HMI yang dapat mengontrol dan melihat sistem produksi secara virtual secara real-time. Dalam penelitian ini, data yang diterima oleh PLC dapat ditampilkan pada HMI, seperti input, output, perhitungan, dan notifikasi kesalahan. Hal ini dapat mengurangi waktu yang dihabiskan untuk mengevaluasi perbaikan masalah pada mesin produksi.

**Tabel 2. Tabel Literature Review Matix**

<i>Author/ Date</i>	<i>Theoretical/ Conceptual Framework</i>	<i>Research Question(s)/ Hypotheses</i>	<i>Methodology</i>	<i>Analysis &amp; Results</i>	<i>Conclusions</i>	<i>Implications for Future research</i>	<i>Implications For practice</i>
		kontrol sistem secara real time?	perangkat, dan HMI.	analisis masalah pada mesin produksi lebih cepat.			
Dedy Agung Prabowo , Dedy Abdullah , dan Ari Manik (2018)	Mendefinisikan pengolahan citra, teknologi visi komputer, dan teknik pelacakan objek untuk mendeteksi dan menghitung objek berdasarkan warna menggunakan teknik pelacakan objek warna.	mengembangkan metode untuk mendeteksi dan memantau objek berdasarkan warna yang menggunakan pengawasan objek warna. menggunakan <i>webcam</i> sebagai input gambar dan membagi gambar untuk membedakan warna yang ingin dideteksi dari	menggunakan metode penelitian, yang dimulai dengan membuat flowchart dan membuat aplikasi untuk mendukung penelitian. Selanjutnya, penulis melakukan pengujian dengan berbagai kondisi pencahayaan lingkungan dengan luminosity atau lux sebagai	Hasil studi menunjukkan bahwa perbedaan warna objek dan latar belakang sangat memengaruhi keberhasilan pendeteksian objek. 31-15.000 lux adalah pencahayaan objek yang baik. Objektif	Sistem yang dibangun memiliki kemampuan untuk mendeteksi dan menghitung objek yang telah ditentukan. Keberhasilan pendeteksian objek sangat bergantung pada perbedaan warna antara objek dan latar belakang.	menambahkan algoritme tambahan untuk meningkatkan stabilitas pelacakan objek dan menerapkan sistem pada mesin. Selain itu, dengan menggunakan teknik pengolahan gambar seperti perbaikan kontras, perbaikan tepian objek, dan penajaman, Anda dapat memperluas cakupan warna yang dapat dideteksi oleh sistem dan meningkatkan kualitas gambar. Mengembangkan sistem pelacakan objek	pengembangan sistem pelacakan objek yang dapat digunakan untuk berbagai aplikasi seperti sistem keamanan, pengawasan lalu lintas, dan pengenalan wajah.

**Tabel 2. Tabel Literature Review Matix**

<i>Author/ Date</i>	<i>Theoretical/ Conceptual Framework</i>	<i>Research Question(s)/ Hypotheses</i>	<i>Methodology</i>	<i>Analysis &amp; Results</i>	<i>Conclusions</i>	<i>Implications for Future research</i>	<i>Implications For practice</i>
		backgroundnya . Selain itu, untuk mengetahui bagaimana pencahayaan mempengaruhi sistem deteksi dan pelacakan objek	satuan cahaya. Selain menggunakan alat bantu android dengan aplikasi pengukur cahaya, teknik segmentasi, atau filtering, digunakan untuk membedakan warna yang ingin dideteksi dari background dan menentukan area tempat objek berada, sehingga dapat diketahui lokasinya, dan dapat ditandai dengan area kotak sebagai tanda terlacaknya	yang telah ditentukan dapat dideteksi dan dihitung oleh sistem yang telah dikembangkan.		yang lebih akurat dan cepat secara real-time.	

**Tabel 2. Tabel Literature Review Matix**

<i>Author/ Date</i>	<i>Theoretical/ Conceptual Framework</i>	<i>Research Question(s)/ Hypotheses</i>	<i>Methodology</i>	<i>Analysis &amp; Results</i>	<i>Conclusions</i>	<i>Implications for Future research</i>	<i>Implications For practice</i>
			objek, dan juga menghitung jumlah objek yang telah terdeteksi.				
Bilal Naji Alhasna wi dan Basil H. Jasim (2019)	Mengembangkan sistem rumah pintar dengan menggunakan SCADA, modul Raspberry Pi3, dan papan Wemos-D1. Penggunaan protokol SCADA, TCP/IP, dan MQTT. Tidak seperti saklar konvensional yang dipasang di dinding, rumah modern diharapkan memiliki sistem kendali terpusat. Pengguna harus bergerak ke dekat dinding untuk mengontrol sakelar. Orang lanjut usia dan penyandang cacat tidak suka cara ini. Sistem rumah	Tidak seperti saklar konvensional yang dipasang di dinding, rumah modern diharapkan memiliki sistem kendali terpusat. Pengguna harus bergerak ke dekat dinding untuk mengontrol sakelar. Orang lanjut usia dan penyandang cacat tidak suka cara ini. Sistem rumah	Komputer menggunakan jaringan Wi-Fi untuk mengirimkan perintah ke modul unit stasiun pangkalan (Raspberry Pi3) dan jaringan TCP/IP untuk mengirimkan perintah ke unit terminal (board Wemos-D1). Aliran ini juga berfungsi sebaliknya, yaitu unit terminal mengirimkan	Kebutuhan masyarakat dapat dipenuhi melalui sistem rumah pintar yang dikendalikan dari jarak jauh. Komputer pribadi yang didukung oleh IGSS SCADA berfungsi sebagai pemancar dan	Papan Wemos-D1, Raspberry Pi3, dan SCADA digunakan untuk membuat sistem rumah pintar. Ini adalah sistem otomasi yang hemat biaya, tidak membutuhkan perawatan, dan ramah pengguna untuk membantu orang lanjut usia dan	Dimungkinkan untuk mengembangkan sistem rumah pintar ini di berbagai lokasi, seperti industri, mal, dan rumah sakit.	Dengan Raspberry Pi3 sebagai pusat kontrol, teknologi SCADA dapat digunakan untuk mengontrol dan mengotomatisasi rumah pintar. Dalam hal ini, teknologi SCADA mengumpulkan data dari peralatan rumah tangga dan mengirimkannya ke aplikasi IGSS melalui protokol TCP, dan teknologi MQTT mengirimkan perintah ke peralatan rumah tangga dari aplikasi IGSS. Dengan demikian, teknologi SCADA dan MQTT

**Tabel 2. Tabel Literature Review Matix**

<i>Author/ Date</i>	<i>Theoretical/ Conceptual Framework</i>	<i>Research Question(s)/ Hypotheses</i>	<i>Methodology</i>	<i>Analysis &amp; Results</i>	<i>Conclusions</i>	<i>Implications for Future research</i>	<i>Implications For practice</i>
		pintar yang memiliki modul untuk mengontrol peralatan dari jarak jauh dapat membantu mereka mengatasi kesulitan mereka dalam mengoperasikannya.	informasi ke unit stasiun pangkalan dan unit stasiun pangkalan mengirimkan informasi ke komputer melalui jaringan Wi-Fi.	mengirimkan perintah ke modul Raspberry Pi3. SCADA menerima perintah dari modul Raspberry Pi3 dan menggunakan MQTT untuk mengirimkan perintah dengan alamat ke unit terminal (papan Wemos-D1). Jika alamat unit terminal cocok dengan	penyandang disabilitas. Sistem kendali terpusat memungkinkan pengoperasian peralatan melalui IGSS SCADA. Ini juga menghemat waktu konsumen untuk mengoperasikan peralatan. Untuk menghubungkan beban, saklar konvensional membutuhkan lebih banyak kabel, tetapi sistem yang diusulkan membutuhkan		dapat digunakan untuk mengotomatisasi rumah pintar.

**Tabel 2. Tabel Literature Review Matix**

<i>Author/ Date</i>	<i>Theoretical/ Conceptual Framework</i>	<i>Research Question(s)/ Hypotheses</i>	<i>Methodology</i>	<i>Analysis &amp; Results</i>	<i>Conclusions</i>	<i>Implications for Future research</i>	<i>Implications For practice</i>
				Raspberry Pi3, maka Raspberry Pi3 dapat melakukan Artikel ini menggunakan papan pengontrol Wemos-D1 yang dilengkapi dengan IC ESP8266. Relay elektromagnetik digunakan untuk menghubungkan beban ke pengontrol Wemos-D1.	lebih sedikit kabel. Peralatan di rumah dapat diatur dan statusnya dapat dipantau dari jarak jauh.		

**Tabel 2. Tabel Literature Review Matix**

<i>Author/ Date</i>	<i>Theoretical/ Conceptual Framework</i>	<i>Research Question(s)/ Hypotheses</i>	<i>Methodology</i>	<i>Analysis &amp; Results</i>	<i>Conclusions</i>	<i>Implications for Future research</i>	<i>Implications For practice</i>
Prof Ravi Kiran R dan Sparsha S Reddy  (2022)	Antarmuka Mesin Manusia berfungsi sebagai artefak mediasi dalam sistem kompleks untuk menjembatani otomatisasi dan kontrol mesin yang penting untuk kesadaran situasi, keandalan, efisiensi, efektivitas, ketahanan, dan keselamatan. Selain itu, dengan mempelajari pengalaman agen manusia selama fase perancangan dan pengujian, antarmuka ini dapat memberikan pemahaman yang lebih baik tentang	Saat ini, telah diakui bahwa interaksi manusia-mesin dengan pabrik industri dan sistem teknis dinamis lainnya sangat penting untuk keselamatan, kualitas, dan efisiensi proses. Ini mencakup semua aspek interaksi dan komunikasi antara manusia dan mesin melalui antarmuka manusia-mesin.	Antarmuka mesin manusia memiliki berbagai kelas pengguna manusia, seperti operator, insinyur, personel pemeliharaan, dan manajer. Meskipun berbeda, mereka tumpang tindih dengan informasi. Mesin sebagai sistem otomatis menjadi lebih kompleks daripada mesin dengan lebih sedikit otomatisasi,	Kompleksitas yang meningkat dan kendali yang lebih canggih menuntut kualitas komunikasi dan kerjasama antara manusia dan mesin yang lebih baik. Peran pengguna manusia bergeser dari pengontrol menjadi pengawas. Pengawas	Tidak bergantung pada heuristik "pengamat manusia – artefak antarmuka" untuk menangani masalah yang muncul sebagai akibat dari integrasi teknologi dan manusia. Akibatnya, tampilan ini benar-benar dapat membantu agen manusia beradaptasi secara dinamis terhadap variabilitas dan kontinjensi	Berkonsentrasi pada tujuh tantangan besar: simbiosis manusia-teknologi: interaksi manusia-lingkungan; etika, privasi, dan keamanan; kesejahteraan, kesehatan, dan eudaimonia; aksesibilitas universal dan aksesibilitas; kreativitas dan pembelajaran; dan skema dan demokrasi. Penelitian di masa depan memiliki kemampuan untuk meningkatkan pemahaman kita tentang masalah-masalah ini dan menghasilkan solusi yang lebih baik untuk masalah yang terkait	Tujuh tantangan besar—simbiosis manusia-teknologi; interaksi manusia-lingkungan; etika, privasi, dan keamanan; kesejahteraan, kesehatan, dan eudaimonia; aksesibilitas universal dan aksesibilitas; kreativitas dan pembelajaran; skema, dan demokrasi—harus dipertimbangkan saat merancang antarmuka mesin manusia. Dalam praktiknya, ini berarti bahwa ketika merancang antarmuka mesin manusia untuk memenuhi kebutuhan pengguna dan mengatasi masalah interaksi manusia-teknologi, elemen-

**Tabel 2. Tabel Literature Review Matix**

<i>Author/ Date</i>	<i>Theoretical/ Conceptual Framework</i>	<i>Research Question(s)/ Hypotheses</i>	<i>Methodology</i>	<i>Analysis &amp; Results</i>	<i>Conclusions</i>	<i>Implications for Future research</i>	<i>Implications For practice</i>
	kompleksitas sistem kompleks yang sangat terdistribusi.		yang menghasilkan struktur kendali pengawasan yang lebih canggih.	manusia berinteraksi dengan sistem melalui satu atau lebih lapisan komputer, di mana antarmuka manusia-mesin, otomatisasi, dan fungsi pendukung keputusan diterapkan. Oleh karena itu, interaksi manusia-mesin yang dirancang dengan baik	sistem yang sangat kompleks. Hal ini mungkin dianggap sebagai proses pemikiran yang keluar dari kotak dualisme konvensional konsep HMI, di mana antarmuka dan manusia adalah entitas yang terpisah. Namun, desain HMI yang berbeda dapat berdampak besar pada kinerja operator. Penelitian ini menyelidiki bagaimana	dengan interaksi manusia-teknologi.	elemen ini harus dipertimbangkan. Selain itu, diperlukan perbaikan kualitas komunikasi dan peningkatan kemampuan adaptif operator karena kompleksitas sistem otomatis yang meningkat dan struktur kendali yang semakin canggih.

**Tabel 2. Tabel Literature Review Matix**

<i>Author/ Date</i>	<i>Theoretical/ Conceptual Framework</i>	<i>Research Question(s)/ Hypotheses</i>	<i>Methodology</i>	<i>Analysis &amp; Results</i>	<i>Conclusions</i>	<i>Implications for Future research</i>	<i>Implications For practice</i>
				menjadi sangat penting dalam pengendalian proses tingkat lanjut, akibat dari tingkat otomatisasi yang tinggi dan munculnya fungsi otomatisasi yang lebih luas. Teknologi canggih dengan fungsi yang lebih luas mencakup sistem pakar	properti terdistribusi dari sistem dapat mempengaruhi desain HMI. Tujuan dari penelitian ini adalah untuk mengetahui apa yang membentuk pendukung keputusan yang andal. HMI membantu elemen manusia menyelesaikan masalah.		

**Tabel 2. Tabel Literature Review Matix**

<i>Author/ Date</i>	<i>Theoretical/ Conceptual Framework</i>	<i>Research Question(s)/ Hypotheses</i>	<i>Methodology</i>	<i>Analysis &amp; Results</i>	<i>Conclusions</i>	<i>Implications for Future research</i>	<i>Implications For practice</i>
				untuk kontrol dan diagnostik, kontrol proses fuzzy, dan jaringan saraf tiruan dalam kontrol proses.			
Helmy Rahadian dan M Ary Heryanto (2020)	<i>Human Machine Interface (HMI) dan programmable Logic Control (PLC)</i> merupakan komponen penting dari sistem otomasi industri. Piranti yang dapat diprogram yang dikenal sebagai PLC dapat melakukan fungsi pengendalian pada	Survei yang dilakukan menunjukkan bahwa banyak perusahaan telah mengetahui konsep industri 4.0 tetapi tidak tahu cara memanfaatkannya. Karena akan semakin banyak	pengembangan sistem antarmuka pengguna (HMI) pada simulator sortir bola, termasuk menentukan piranti I/O, mendesain layout HMI, memprogram PLC, menentukan	Dengan bantuan sistem HMI baru, pengujian telah dilakukan pada simulator sortir bola. Hasilnya ditunjukkan pada Tabel 3,	Seperti yang diharapkan, simulator sortir bola memenuhi semua kriteria pengujian dan panel monitor HMI telah ditambahkan.	Ada banyak cara untuk mengembangkan sistem HMI pada simulator, salah satunya adalah dengan menambahkan level akses pengguna, data logger, dan alarm. Level akses pengguna adalah sistem keamanan yang diterapkan untuk memberi pengguna akses ke sumber daya yang tersedia di HMI	Simulator sortir bola memiliki sistem HMI yang dapat digunakan sebagai alat untuk belajar teknologi otomasi industri, khususnya PLC dan HMI. Diharapkan bahwa pengembangan sistem HMI ini akan membantu proses pembelajaran dan menghasilkan profil lulusan yang

**Tabel 2. Tabel Literature Review Matix**

<i>Author/ Date</i>	<i>Theoretical/ Conceptual Framework</i>	<i>Research Question(s)/ Hypotheses</i>	<i>Methodology</i>	<i>Analysis &amp; Results</i>	<i>Conclusions</i>	<i>Implications for Future research</i>	<i>Implications For practice</i>
	<p>sistem yang kompleks maupun sederhana. Sebaliknya, Human-Machine Interface (HMI) bertanggung jawab untuk menciptakan interaksi yang mudah digunakan antara manusia sebagai operator dan mesin. HMI memungkinkan pemantauan dan pemahaman proses yang berjalan.</p> <p>Jika terjadi situasi yang tidak normal selama proses produksi, HMI dapat memberikan notifikasi melalui</p>	<p>penggunaan mesin otomatis dan modern, perusahaan akan banyak berinvestasi dalam pelatihan karyawannya. Di Indonesia sendiri, yang memiliki tenaga kerja terbesar keempat di dunia, masih banyak orang yang tidak memiliki kemampuan yang diperlukan. Dunia akademik menjadi sangat penting untuk</p>	<p>parameter proses, dan merancang layout HMI.</p>	<p>yang menunjukkan bahwa sistem HMI telah berhasil memenuhi semua poin pengujian. Selain itu, dibahas juga tahapan-tahapan pengembangan sistem HMI pada simulator sortir bola, termasuk menentukan piranti I/O, mengemban</p>		<p>sesuai dengan tingkat pengguna, seperti operator, pimpinan, dan administrator. Adanya data logger dan alarm membantu melakukan analisis dan troubleshooting masalah.</p>	<p>berpengalaman dalam teknologi otomasi industri. Pengembangan sistem HMI pada simulator sortir bola juga dapat membantu dalam menganalisis dan menyelesaikan masalah. Sistem HMI yang memiliki data logger, alarm, dan sistem keamanan akses tingkat user juga dapat meningkatkan keamanan dan efisiensi penggunaan simulator sortir bola.</p>

**Tabel 2. Tabel Literature Review Matix**

<i>Author/ Date</i>	<i>Theoretical/ Conceptual Framework</i>	<i>Research Question(s)/ Hypotheses</i>	<i>Methodology</i>	<i>Analysis &amp; Results</i>	<i>Conclusions</i>	<i>Implications for Future research</i>	<i>Implications For practice</i>
	alarm audio dan visual.	menyediakan tenaga kerja yang berkualitas, dan konsep Education 4.0 adalah upaya untuk mencapainya.		layout HMI, dan memprogram PLC.			

## 2.2. Dasar Teori

### 2.2.1. Raspberry Pi



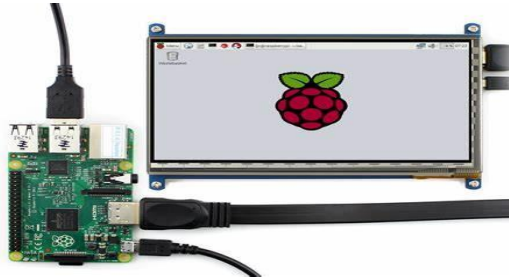
**Gambar 1. Gambar Raspberry Pi**

Raspberry Pi adalah komputer papan tunggal (SBC) seukuran kartu kredit yang dapat menjalankan program perkantoran, permainan komputer, dan pemutar media dengan resolusi tinggi. Sejumlah pengembang dan ahli komputer dari Universitas Cambridge, Inggris, bertanggung jawab atas pengembangan Raspberry Pi, yang didirikan oleh yayasan nirlaba Raspberry Pi *Foundation*[9].

Raspberry Pi menawarkan kemampuan komputasi yang luar biasa karena ia bukan mikrokontroler, tetapi komputer berukuran kecil yang berfungsi penuh. Raspberry Pi cukup kuat untuk setiap operasi kontrol kompleks yang diperlukan di node karena memiliki kecepatan CPU antara 700MHz dan 1.2GHz dan RAM sebesar 1GB. Kartu SD eksternal dengan kapasitas 8-16GB harus digunakan, meskipun tidak memiliki memori *Flash on-board*. Ini memungkinkan untuk menyimpan banyak program dan bahkan gambar dari modul kamera keamanan jika diperlukan. Raspberry Pi juga dapat memuat dan menjalankan Sistem Operasi Linux dari kartu micro SD. OS Linux memungkinkan Raspberry Pi melakukan multitasking dan memberikan banyak fleksibilitas dan kemudahan ketika menangani kebutuhan sistem secara *real-time*. Raspberry Pi juga memiliki empat puluh pin GPIO tujuan umum[3]

Penggunaan Raspberry Pi pada penelitian kali ini adalah berfungsi sebagai otak atau pusat pengendalian mesin. Alat ini juga berperan penting dalam memproses informasi, mengatur tindakan, dan mengendalikan berbagai aspek mesin untuk melakukan tugas-tugas tertentu, dengan ukuran sebesar kartu kredit, hal ini memudahkan pengguna untuk mengintegrasikan ke dalam beberapa proyek dan juga perangkat yang memiliki ruang terbatas.

### 2.2.2. LCD



**Gambar 2. Gambar LCD**

LCD (*Liquid Crystal Display*) adalah jenis layar yang menggunakan Liquid Crystal sebagai media reflektif. LCD dapat digunakan di berbagai bidang seperti: Monitor, TV, kalkulator. Monitor LCD berwarna memiliki puluhan ribu piksel. Piksel adalah unit terkecil dari LCD. 10.000 piksel ini digabungkan menjadi gambar dengan bantuan pengontrol yang ditempatkan di monitor. Dalam dunia elektronik, LCD digunakan sebagai tampilan atau layar hemat energi. LCD sendiri merupakan teknologi tampilan digital yang memiliki struktur molekul polar dan membentuk gambar pada permukaan datar dengan menyinari Liquid Crystal dan filter warna yang diapit di antara dua elektroda transparan. Namun, Liquid Crystal tidak memancarkan cahaya secara langsung. Ketika medan listrik diterapkan, molekul menyesuaikan posisinya di dalam medan dan membentuk struktur kristal yang mempolarisasi cahaya yang melewatinya.

Penggunaan LCD pada HMI (*Human Machine Interface*) menggunakan resolusi  $1024 \times 600$ , LCD Layar Sentuh Kapasitif 7 inci, antarmuka HDMI, mendukung berbagai sistem. Panel sentuh kapasitif kaca yang dikeraskan, kekerasan 6H (hanya versi Case yang memiliki kaca kekerasan). Ini mendukung Raspberry Pi dan juga dapat digunakan sebagai monitor komputer. Penggunaan LCD pada HMI di penelitian ini bertujuan untuk menampilkan tampilan dari hasil pelacakan objek dan juga GUI yang sudah dibuat.

### 2.2.3. PLC TM241CE40R



**Gambar 3. Gambar PLC**

PLC banyak dikenal sebagai akronim dari PC (*Personal Computer*). Dan ini menjadikan suatu hal yang membingungkan antara pengertian PLC dan PC, akhirnya sekarang PLC memiliki pengertian tersendiri yaitu Programmable Logic Controller. PLC adalah sebuah peralatan user friendly berbasis microprocessor. PLC merupakan suatu komputer khusus yang berisi fungsi kontrol dari berbagai jenis dan level secara kompleksitas. Sistem PLC pertama dikembangkan dari komputer konvensional pada akhir tahun 1960 dan awal 1970[10]

Secara definisi bahwa PLC atau *programmable logic controller* adalah suatu alat kontrol (kendali) yang dapat diprogram dengan menggunakan *console* atau komputer/laptop untuk mengendalikan logik-logik yang terdapat dalam CPU (*Central Processing Unit*). Bentuk program yang dihasilkan pada PLC dinamakan *ladder diagram* (diagram tangga) dan instruksi pada *ladder diagram* dapat dibaca dengan menggunakan *Mnemonic Programming*. Pada *ladder diagram* terdiri dari *input output* dengan sumber tegangan, yang tersusun seperti tangga, dan bekerja secara berurutan. PLC dengan berbagai macam *input* dan *output*, merupakan peralatan kendali yang tergolong kedalam arus kuat yang biasa digunakan di industri-industri yang menghasilkan produk secara kontinu dalam jumlah yang banyak. PLC merupakan alat kendali yang bekerja dengan prinsip pensaklaran, yaitu berupa *switch*, *timer* dan *counter* yang sudah ada di dalam PLC itu sendiri. PLC terdiri dari bagian masukan, proses dan keluaran. PLC ini dapat diprogram dan disesuaikan dengan proses yang diinginkan, serta dapat dikomunikasi dengan LCD sebagai tampilan HMI [11]

Penggunaan PLC TM241CE40R pada penelitian ini memiliki tujuan untuk mengontrol *conveyor* dan *cylinder pick and place*, untuk mensortir barang sesuai dengan warna produk yang ditentukan pada mesin yang digunakan [12].

## 2.2.4 Visual Studio 2022



**Gambar 4. Gambar icon visual studio**

Visual Studio 2022 merupakan suatu perangkat lunak yang digunakan untuk pengembangan aplikasi, baik aplikasi bisnis maupun personal dalam bentuk aplikasi *console*, aplikasi Windows, atau aplikasi web. Visual studio memiliki berbagai macam tipe antara lain aplikasi desktop (*Windows Form*, *Command Line (Console)*, Aplikasi Web, dan *Windows Mobile*). Visual studio pada penelitian ini digunakan untuk membuat desain *Graphic User Interface* (GUI), pemrograman pada tampilan layar di HMI, juga membuat pemrograman kamera pelacakan objek.

## 2.2.5. Ruang warna YCbCr

YCBCR adalah teknik ruang warna yang digunakan untuk sistem video digital dan fotografi. Luminansi (Luma Y) adalah kecerahan yang terjadi dengan menggunakan gradasi hitam dan putih. Krominansi (chroma [CB dan CR]) adalah informasi warna dalam sinyal yang terutama terkonsentrasi pada "YCBCR" baik dalam sinyal merah atau biru. Studi tentang mata manusia mengatakan bahwa luminansi sangat sensitif tetapi tidak sama dengan krominansi. Oleh karena itu, metode ruang warna YCBCR menggunakan efeknya untuk menunjukkan variasi luminansi dan krominansi dengan menyortir modul dari gambar yang ditentukan [13]. Konversi nilai dari warna RGB ke YCbCr digambarkan sebagai berikut:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

**Gambar 5. Gambar konversi RGB ke YCbCr**

Pada penelitian ini, ruang warna YCbCr digunakan sebagai metode untuk melakukan pelacakan objek dengan menggunakan *YCbCrColorFiltering* yang disediakan oleh pustaka AForge.NET untuk pemrosesan gambar. Penulisan program yang digunakan untuk metode tersebut adalah sebagai berikut:

```

private void YCbCrColorFiltering(Bitmap srcImage)
{
    YCbCrFiltering filter = new YCbCrFiltering();

    filter.Y = new Range(Ymin, Ymax);
    filter.Cb = new Range(Cbmin, Cbmax);
    filter.Cr = new Range(Crmin, Crmax);

    detectedImage = filter.Apply(srcImage);
    pictureBox1.Image = detectedImage;
}

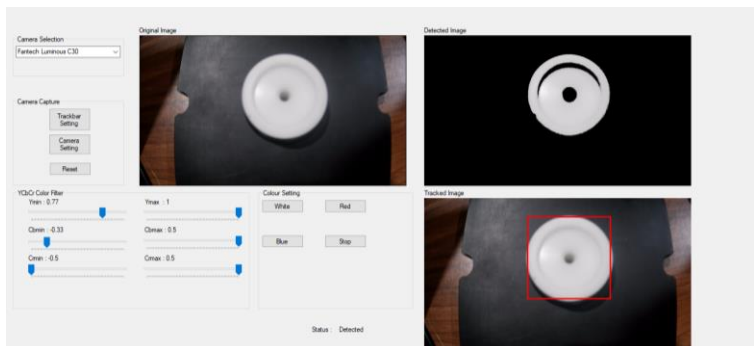
```

Metode *YCbCrColorFiltering* ini digunakan untuk memproses sebuah citra dengan menyaringnya berdasarkan komponen-komponen warna dalam model YCbCr dan menampilkan hasilnya dalam sebuah kontrol antarmuka pengguna. Rentang nilai untuk komponen Y, Cb, dan Cr yang digunakan dalam penyaringan warna dengan model warna YCbCr pada penelitian ini adalah:

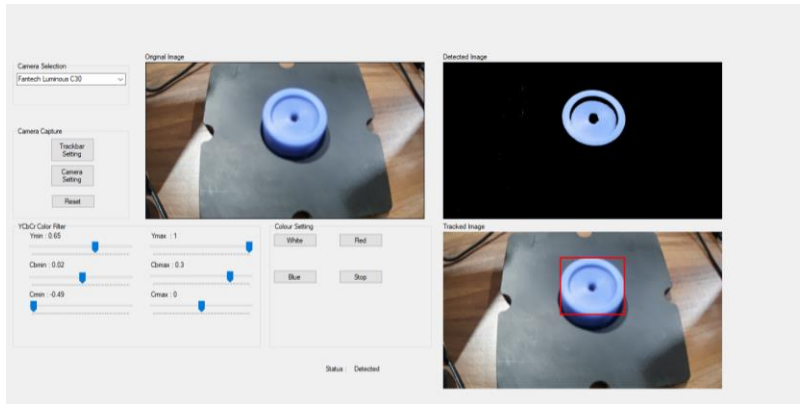
```

float Ymin = 0, Ymax = 1;
float Cbmin = -0.5f, Cbmax = 0.5f;
float Crmin = 0.5f, Crmax = 0.5f;

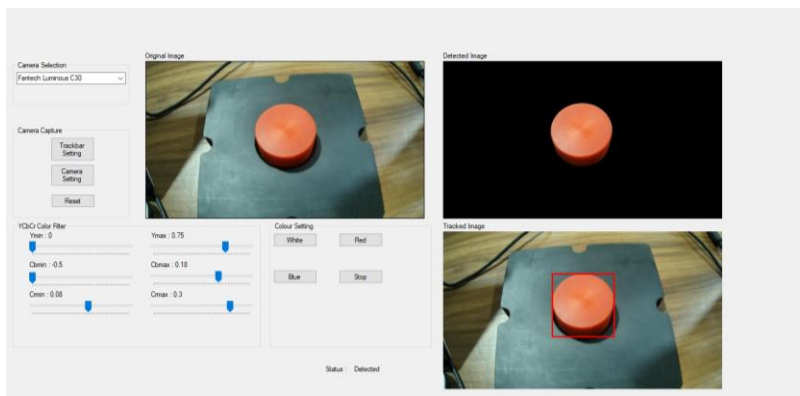
```



**Gambar 6.** Gambar pengujian nilai YCbCr pada objek berwarna putih



**Gambar 7. Gambar pengujian nilai YCbCr pada objek berwarna biru**



**Gambar 8. Gambar pengujian nilai YCbCr pada objek berwarna merah**

Berdasarkan rentang nilai YCbCr, diperoleh nilai dari tiap warna yang akan diuji setelah melakukan percobaan untuk mencari nilai yang paling sesuai saat mendeteksi objek seperti pada gambar 6, gambar 7, dan gambar 8. Pada kotak gambar bagian *detected image* terlihat bahwa metode *YCbCrColorFiltering* memisahkan objek yang akan dideteksi dengan latar belakang yang di tampilkan pada kotak gambar *tracked image*, kotak berwarna merah memperlihatkan objek yang telah dideteksi. Dalam penelitian ini, nilai tersebut dituliskan dalam pemrograman sebagai berikut:

```

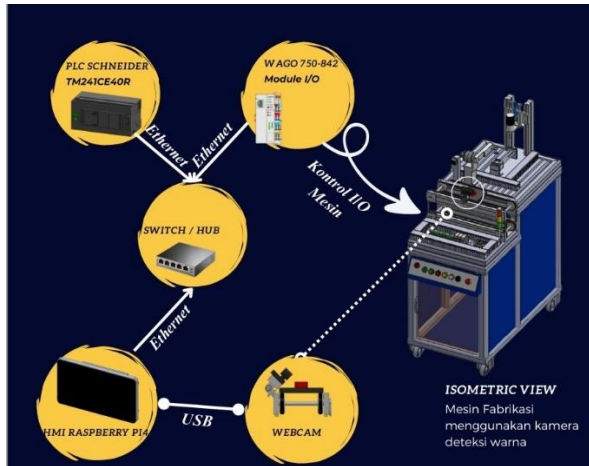
private List<ColorData> colors = new List<ColorData>
{
    new ColorData { Name = "WHITE", Ymin = 0.77f, Ymax = 1f, Cbmin = -
0.33f, Cbmax = 0.5f, Crmin = -0.5f, Crmax = 0.5f },
    new ColorData { Name = "BLUE", Ymin = 0.65f, Ymax = 1f, Cbmin = 0.02f,
Cbmax = 0.3f, Crmin = -0.49f, Crmax = 0f },
    new ColorData { Name = "RED", Ymin = 0f, Ymax = 0.75f, Cbmin = -0.5f,
Cbmax = 0.18f, Crmin = 0.08f, Crmax = 0.3f }
};
private int counter;

```

### 2.2.6. Komunikasi Modbus TCP/IP

Komunikasi Modbus TCP/IP digunakan sebagai protocol komunikasi yang berfungsi mengirimkan data dari perangkat pengirim seperti PC atau HMI yang mendukung Modbus TCP/IP ke perangkat penerima seperti PLC, Arduino dan perangkat penerima lainnya yang mendukung protocol tersebut melalui jaringan *ethernet*. Komunikasi Modbus TCP/IP merupakan varian komunikasi Modbus yang digunakan dalam komunikasi data di internet dan jaringan lokal.

Fungsi utama TCP adalah memastikan bahwa semua paket data diterima dengan benar. Sedangkan IP berfungsi untuk memastikan data atau pesan tersebut telah dialamatkan dan dirutekan dengan benar. Modbus TCP/IP mendukung berbagai tipe topologi jaringan seperti *Star*, *Tree*, dan *Daisy Chain*. Sehingga memungkinkan untuk perluasan dan modifikasi jaringan sesuai dengan kebutuhan. Perangkat seperti *switch/router ethernet* sangat membantu dalam hal perluasan jaringan serta untuk menjangkau jarak yang cukup jauh. Modbus TCP/IP dapat memiliki lebih dari satu *Server* yang dapat mengontrol lebih banyak perangkat sebanyak 247 *Client* pada jaringan *ethernet* yang sama, dengan catatan perangkat memiliki Alamat IP yang unik. *Server* memiliki fungsi untuk memulai inisiasi komunikasi seperti menulis dan membaca data, serta mengetahui status dan kondisi dari *Client*. Berikut gambar topologi jaringan Modbus TCP/IP antara HMI, PLC, dan juga Wago pada mesin deteksi warna objek.



**Gambar 9. Gambar Topologi Jaringan**

Untuk Alamat IP Address dari PLC TM241 adalah 192.168.1.4 sedangkan Address dari Wago/Modul *input* dan *output* adalah 192.168.1.5 dan untuk alamat IP Address dari HMI Raspberry Pi adalah 192.168.1.2. Topologi jaringan yang digunakan adalah tipe *star*/bintang terlihat pada gambar perangkat *switch* sebagai jembatan penghubung untuk menghubungkan beberapa perangkat tersebut melalui jaringan Modbus TCP/IP. HMI Raspberry Pi berperan sebagai *Server* yang dapat mengontrol dan memonitoring data dari beberapa *Client*nya yaitu PLC TM241 dan WAGO modul I/O (*input/output*).

Pada komunikasi Modbus TCP/IP terdapat 4 buah tipe penyimpanan/pengolahan data yang digunakan untuk proses *transfer* data melalui komunikasi Modbus TCP/IP:

**Tabel 3. Tabel Komunukasi Modbus**

Tipe Memori	Tipe Objek	Jenis Tugas	Keterangan
<i>Coils</i>	<i>Single Bit</i>	<i>Read-Write</i>	<i>Server</i> dan <i>Client</i> dapat merubah data <i>Coil</i>
<i>Discretes Input</i>	<i>Single Bit</i>	<i>Read-Only</i>	<i>Server</i> hanya dapat membaca data <i>Coil</i> sedangkan <i>Client</i> yang dapat merubah data <i>Coil</i>
<i>Input Registers</i>	<i>16-bit word</i>	<i>Read-Only</i>	<i>Server</i> hanya dapat membaca nilai data <i>Word</i> sedangkan <i>Client</i> yang

			dapat merubah nilai data <i>Word</i>
<i>Holding Registers</i>	16-bit <i>word</i>	<i>Read-Write</i>	<i>Server</i> dan <i>Client</i> dapat merubah nilai data <i>Word</i>

Untuk tipe penyimpanan/pengolahan data yang dapat digunakan pada komunikasi Modbus TCP/IP antara HMI dengan PLC TM241 hanya dapat menggunakan tipe *registers* saja. Hal ini dikarenakan tidak ada ruang penyimpanan untuk *Modbus.Coil* yang tersedia pada *software* SoMachine/EcoStruxure Machine Expert PLCs. Jadi jika ingin menggunakan *transfer* data *Coil/bit* dapat memanfaatkan nilai bilangan bit yang aktif pada saat nilai *word* memiliki nilai sesuai dengan inputan HMI. Berikut alamat memori *register* yang berlaku untuk semua seri PLC TM2XX:

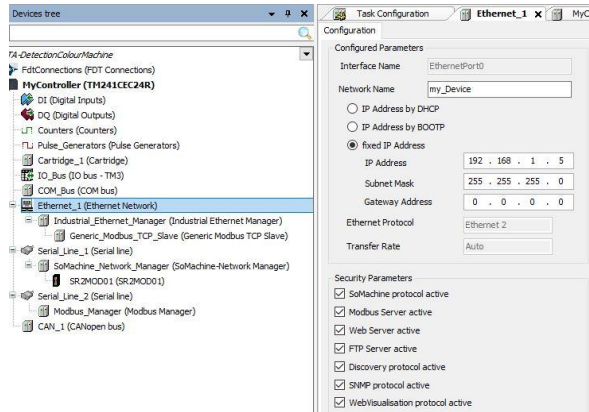
**Tabel 4. Tabel alamat memori PLC**

Double Words	Words	Bytes	Bits		
%MD0	%MW0	%MB0	%MX0.7	...	%MX0.0
		%MB1	%MX1.7	...	%MX1.0
	%MW1	%MB2	%MX2.7	...	%MX2.0
%MB3		%MX3.7	...	%MX3.0	
%MD1	%MW2	%MB4	%MX4.7	...	%MX4.0
		%MB5	%MX5.7	...	%MX5.0
	%MW3	%MB6	%MX6.7	...	%MX6.0
		%MB7	%MX7.7	...	%MX7.0
%MD2	%MW4	%MB8	%MX8.7	...	%MX8.0
		...	...	...	...
	...	...	...	...	

Berikut langkah-langkah pengaturan awal komunikasi Modbus TCP/IP antara HMI Raspberry Pi, PLC, dan Wago modul I/O:

1. Konfigurasi alamat IP dan *Port* komunikasi PLC dan Wago modul I/O.

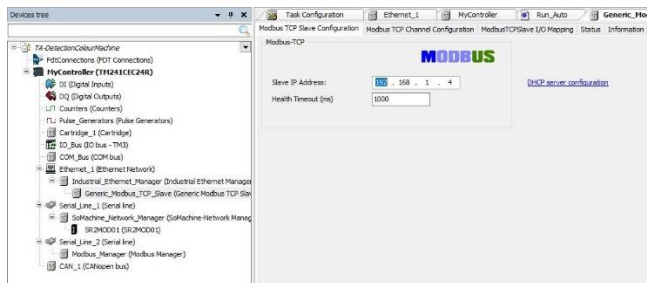
Untuk pengaturan alamat IP pada PLC dapat dilakukan dengan cara klik *Ethernet* pada menu Device Tree *software* Somachine dan selanjutnya mengisi alamat IP pada kolom *IP Address* dan mengisi subnet mask sesuai dengan alamat IP yang digunakan seperti pada gambar 10.



**Gambar 10. Konfigurasi IP Address PLC**

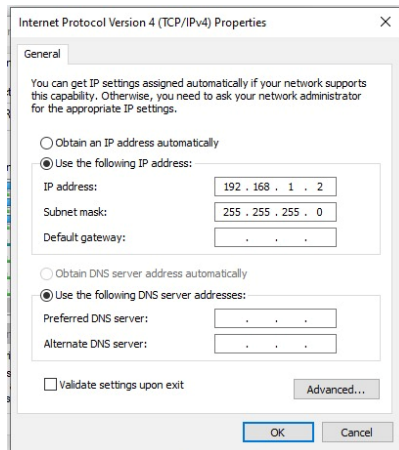
Sedangkan untuk pengaturan alamat IP pada modul Wago dapat dilakukan dengan cara:

- Klik tanda tambah disamping *Ethernet* pada menu *Device Tree* pada *software* Somachine
- Klik tanda tambah disamping *Industrial Ethernet*
- Dan tambahkan *Generic Modbus\_TCP\_Slave* sehingga muncul tampilan seperti nomor 11.
- Masukkan alamat IP pada kolom *Slave IP Address* dan mengisi waktu *timeout* agar PLC dan Modul I/O Wago dapat berkomunikasi melalui jaringan *Modbus TCP/IP*.



**Gambar 11. Konfigurasi IP Address modul wago**

2. Konfigurasi alamat IP dan Port komunikasi HMI Raspberry Pi  
Cara pengaturan alamat IP HMI Raspberry Pi sama persis dengan konfigurasi *ethernet* pada Laptop/PC jika berjalan pada sistem Windows 10. Yaitu dengan cara:
  - Klik pada *start button* lalu pilih *menu settings*
  - Selanjutnya pilih *network and internet*
  - Klik pada *ethernet* → *change adaptor options* → *ethernet* → *properties*
  - Klik *Internet Protocol Version 4 (TCP/IPv4)* lalu klik *properties* lalu selanjutnya pilih “*Use the following IP Address*”
  - Tuliskan alamat IP 192.168.1.2 pada kolom *IP Address* dan biarkan *subnet mask* terisi secara otomatis.



**Gambar 12. Setting IP Address**

3. Menghubungkan kabel *ethernet* setiap perangkat pada *switch*.
4. Menyalakan semua perangkat (PLC, HMI, dan juga Wago modul I/O) termasuk dengan *Switch/Hub*.
5. Menjalankan program komunikasi Modbus TCP/IP menggunakan library EasyModbus pada HMI Raspberry Pi4.

Berikut adalah program C# untuk menghubungkan HMI ke PLC melalui Modbus TCP/IP:

```
ModbusClient plc = new ModbusClient("192.168.1.5", 502); //IP Address PLC
private void button8_Click(object sender, EventArgs e)
{
    try
    {
        plc.Connect();
        if (plc.Connected == true)
        {
            label7.Text = "Connected";
            MessageBox.Show("Connected");
            int j = 1;
            int write = j;
            plc.WriteSingleRegister(9, write);
        }
        else
        {
            MessageBox.Show("Failed to connect to PLC!");
            WriteMessageToNotepad("Failed to connect to PLC");
            int j = 0;
            int write = j;
            plc.WriteSingleRegister(9, write);
        }
    }

    var readplcTimer = new Timer()
    {
        Interval = 1500,
        Enabled = true
    };

    var HistoryAlarm = new Timer()
    {
        Interval = 10,
        Enabled = true
    };

    readplcTimer.Tick += ReadplcTimer_Tick;
```

```

readplcTimer.Start();
HistoryAlarm.Tick += HistoryAlarm_Tick;
HistoryAlarm.Start();
Cameratriger();
Alarm();
AlarmStatus();
connect_indicator();
}
catch (Exception ex)
{
    MessageBox.Show("Error connecting to PLC: " + ex.Message);
    // Ensure lights indicate disconnection in case of error
}
}
}

```

### 2.2.7. Webcam / Kamera



**Gambar 13. Gambar webcam**

Penggunaan *webcam* pada penelitian tugas akhir ini adalah untuk proses pembacaan *object* benda yang akan diproses di Raspberry Pi, adapun spesifikasi *webcam* yang digunakan pada tugas akhir ini adalah:

*Video Resolution: 2560 x 1440 pixel*

*Camera: 4 Megapixel*

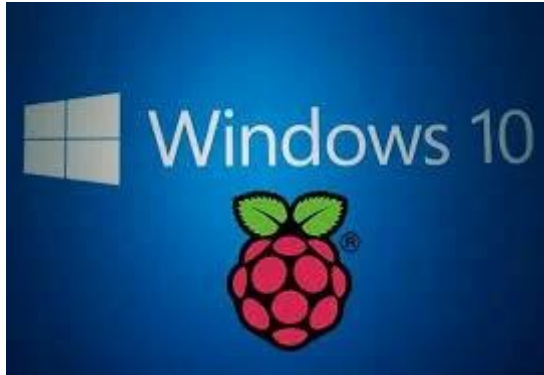
*Field of view: 106°*

*Frame Rate: 25 FPS LOCKED*

*Build in Microphone*

*Plug Type:* USB 2.0  
*Cord length:* 1.4 meter  
*Dimension:* 73 x 26 x33 mm  
*Weight:* 87 gr

### **2.2.8. Operation Windows System**



**Gambar 14. Gambar Windows 10 in Raspberry Pi**

Sistem operasi Windows adalah perangkat lunak yang mengatur penggunaan sumber daya komputer serta menyediakan antarmuka pengguna dan mengelola perangkat keras. Tujuan dari sistem operasi ini adalah memfasilitasi pengguna untuk mengeksekusi program, mengelola memori, mengatur file, dan menyediakan antarmuka pengguna. Dalam penelitian ini, Windows digunakan sebagai sistem operasi yang dijalankan oleh Raspberry Pi untuk mendukung komponen yang digunakan, seperti *webcam* dan library AForge.NET, agar mesin dapat mendeteksi warna. Pada umumnya, Windows tidak diperuntukkan untuk digunakan pada Raspberry Pi. Namun, proyek Windows pada Raspberry Pi ini adalah upaya komunitas yang dimulai pada tahun 2018 yang memungkinkan menjalankan versi desktop penuh Windows (tidak sama dengan IoT Core) pada komputer papan tunggal Raspberry Pi yang terjangkau.

## Bab 3. Metodologi

### 3.1. Perancangan

Untuk memudahkan proses pengerjaan, perlu adanya alur kerja agar proses pengerjaan dapat dilakukan secara terurut dan jelas, seperti yang ada pada Gambar 15.



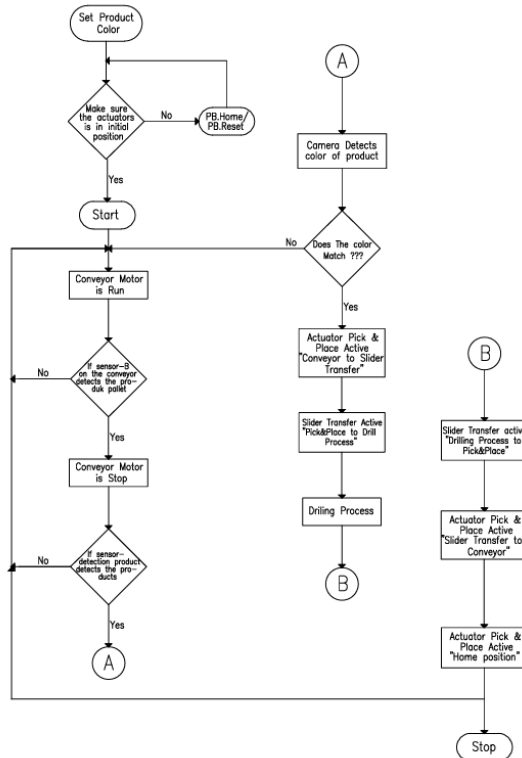
**Gambar 15. diagram alur**

Pada saat akan mengerjakan tugas akhir ini, studi literatur dilakukan terlebih dahulu agar penulis memiliki pengetahuan dan pandangan terhadap apa yang akan dikerjakan. Dimana studi literatur ini digunakan untuk mencari landasan teori, konsep, dan mencari hipotesis penelitian. Setelah mendapatkan landasan teori, konsep, dan hipotesis maka hal yang akan di kerjakan selanjutnya adalah perancangan sistem, perancangan adalah sekumpulan aktivitas yang menggambarkan secara rinci bagaimana sistem akan berjalan. Hal itu bertujuan

untuk menghasilkan produk perangkat lunak yang sesuai dengan kebutuhan pengguna.

Setelah perancangan sistem selesai dibuat, hal selanjutnya yang akan dikerjakan adalah pembuatan GUI menggunakan *Windows form*. *Windows form* atau yang biasa disebut *WinForm* adalah *class library* buatan Microsoft yang bersifat GUI dengan menggunakan Bahasa C#. GUI ini nantinya akan di tampilkan dalam sebuah layar HMI, isi dari GUI tersebut adalah akses masuk pengguna, proses kerja, *input output* dan juga data produk yang dikerjakan. Pada proses kerja dibuat *form* untuk menampilkan dan mengatur pelacakan objek menggunakan metode ruang YCbCr, langkah-langkah lebih jelas nya dalam proses pengerjaan adalah sebagai berikut:

1. Membuat GUI yang akan dijalankan dalam HMI
2. Mengarahkan objek warna pada kamera
3. Mengatur nilai Y, Cb, dan Cr pada *trackbar* yang telah disediakan
4. Membuat algoritma untuk mengklasifikasikan objek yang telah terdeteksi berdasarkan warna
5. Setelah kamera mendeteksi objek, sinyal akan dikirim menjadi nilai *input* pada PLC untuk dilakukan pemilahan



**Gambar 16. Alur kerja mesin**

- Pertama yang perlu diperhatikan saat akan memulai proses mesin *detection object* yaitu mengatur warna produk sesuai dengan kebutuhan output yang diinginkan. Caranya dengan klik *option list* pada HMI lalu *select* warna produk yang diinginkan.
- Selanjutnya pastikan semua posisi *actuator* seperti *rotary cylinder* dan *slider cylinder* sudah berada pada *initial position* atau *home position* agar mesin dapat dijalankan. Jika belum berada pada *initial position*, maka perlu untuk menekan PB.Home pada HMI agar semua *actuator* dapat kembali ke *initial position* secara otomatis.
- Setelah memperhatikan kedua faktor di atas. Selanjutnya mesin dapat dijalankan dengan cara klik *Run button* pada HMI.
- Selanjutnya *conveyor* akan hidup membawa *pallet* produk. Jika sensor A pada *conveyor* mendeteksi sebuah *pallet* produk, maka *conveyor*

akan mati. Jika tidak, maka *conveyor* akan terus bergerak sampai sensor A mendeteksi kembali adanya *pallet* produk.

- e) Jika sensor B / *detection product* mendeteksi adanya produk di atas *pallet* tersebut , maka kamera akan mulai mendeteksi warna dari produk tersebut. Sedangkan jika sensor B pada *conveyor* tidak mendeteksi adanya produk setelah sensor A mendeteksi sebuah *pallet* produk, maka motor *conveyor* akan hidup kembali membawa *pallet* tersebut ke *unloading station*.
- f) Jika hasil deteksi kamera sesuai dengan warna produk yang telah diatur pada pengaturan awal tadi. Maka *actuator pick and place* akan aktif memindahkan produk dari *conveyor* menuju silinder *transfer*. Sedangkan jika warna yang terdeteksi tidak sesuai dengan warna produk yang telah ditentukan. Maka *conveyor* akan hidup kembali untuk memindahkan produk tersebut langsung menuju *unloading station*.
- g) Selanjutnya silinder *transfer* akan memindahkan produk tersebut menuju proses pengeboran.
- h) Setelah proses pengeboran produk selesai. Produk akan dipindahkan kembali menuju silinder *pick and place* dan siap untuk dipindahkan kembali menuju *conveyor*.
- i) *Conveyor* akan memindahkan produk menuju *unloading station*. Dan *rotary cylinder* akan kembali secara bersamaan ke *initial position*.
- j) Selanjutnya *conveyor* pada mesin akan tetap hidup dan *sequence* pada proses mesin ini akan terus berulang sampai tombol *stop/emergency* ditekan dan jika terjadi alarm pada mesin.

### 3.1.1. Perancangan Desain *Graphic User Interface* (GUI)

GUI pada HMI ini menggunakan *Windows form*. *Windows form* atau sering disebut dengan *WinForms* adalah *user interface* (UI) *framework* untuk membuat sebuah aplikasi desktop [6]. Pada gambar nomor 9 merupakan tampilan awal GUI saat pertama kali pengguna membuka GUI tersebut.

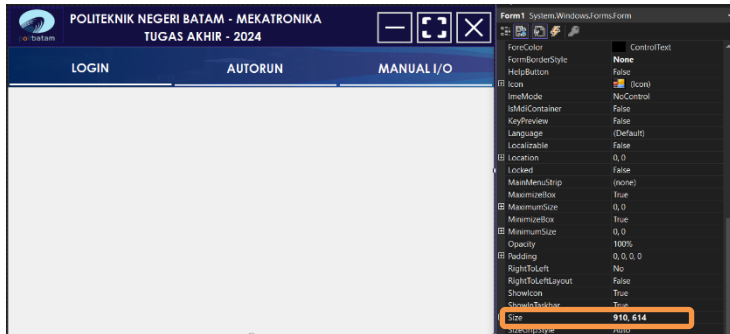
Pada pembuatan *Graphic User Interface* (GUI) terdapat beberapa aturan pokok, diantaranya [14]:

1. Memberikan nama suatu *form* dengan jelas agar dapat dipahami fungsinya oleh pengguna.
2. Penggunaan istilah yang tepat.
3. Menampilkan *form* atau Tampilan yang menarik sesuai dengan kebutuhan.
4. Menggunakan label yang cukup sering digunakan.
5. Penggunaan istilah dan singkatan yang harus dilakukan secara konsisten.
6. Pada penggunaan warna haruslah konsisten pada setiap *form* atau reportnya.
7. Pada saat menggunakan *interface*, *user* dapat dengan mudah menjalankan operasi dengan menggerakkan kursor pada *form*.
8. Berisi pesan kesalahan jika didapati kesalahan memasukkan data.

Untuk membuat desain *interface* yang ada beberapa langkah yang akan dilakukan:

1. Melakukan Riset

Diperlukan adanya riset untuk mengetahui apa saja kebutuhan yang diperlukan saat ingin membuat suatu desain *interface*. Pada penelitian kali ini, riset dilakukan pada alat yang digunakan yaitu LCD sebesar 7 inch. Untuk ukuran LCD sebesar 7 inch, maka didapati ukuran *interface* pada *form* utama sebesar 910 x 614 dan ukuran *form user control* sebesar 910 x 463. Hal ini perlu dilakukan agar nantinya ketika penempatan *tool* yang lain bisa dimaksimalkan pada ruang yang tersedia di dalam panel.



**Gambar 17. Ukuran *Size Form* Utama**

## 2. Panduan desain

Menyusun panduan desain merupakan langkah penting untuk menjaga konsistensi dan efektivitas UI. Panduan ini mencakup standar dan aturan desain yang harus ditaati. Berikut ini adalah komponen utama dari panduan desain:[15]

### a. Warna

Menentukan palet warna yang digunakan, yang mencakup warna aksen, warna pendukung, dan warna utama. Ini diperlukan untuk membuat identitas visual yang konsisten dan mengkomunikasikan suasana atau emosi tertentu.

### b. Tipografi

Menentukan gaya, ukuran, ketebalan, dan jenis font. Pilihan tipografi yang tepat dapat memengaruhi bagaimana pengguna melihat dan merasakan antarmuka.

### c. Ikonografi

Menentukan gaya ikon yang digunakan dan memastikan bahwa ikon mudah dikenali dan tampil konsisten di aplikasi atau situs web.

### d. *Layout* dan spasi

Untuk memastikan antarmuka yang terorganisir dan mudah dinavigasi, atur tata letak elemen seperti *header*, *footer*, dan konten utama, serta atur ruang antara elemen.

### e. Komponen UI

Membuat tampilan dan perilaku standar untuk elemen antarmuka pengguna seperti tombol, *form*, dan menu *dropdown*.

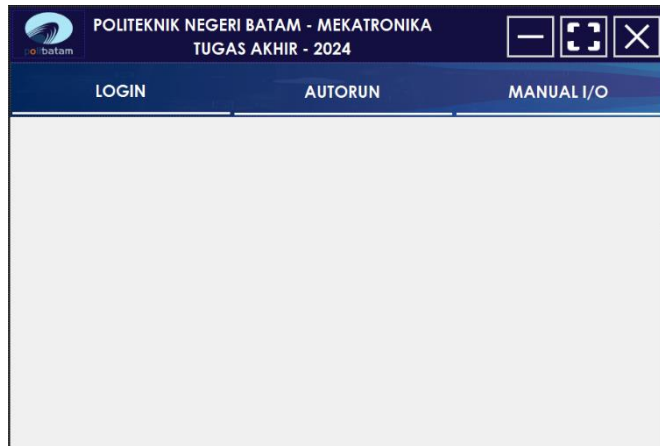
### f. Panduan *Responsif*

Selain itu, sangat penting untuk memiliki aturan desain yang kompatibel dengan berbagai ukuran layar dan perangkat supaya

antarmuka pengguna dapat bekerja dengan baik di berbagai perangkat.

Pembuatan Desain *interface* pada penelitian ini menyesuaikan dengan *device* yang tersedia dan menyesuaikan dengan kebutuhan pada saat penelitian ini dilakukan tanpa mengurangi panduan pokok dan langkah-langkah pada saat pembuatan desain *interface*. Terdapat beberapa fitur di dalam desain *interface* ini, diantaranya:

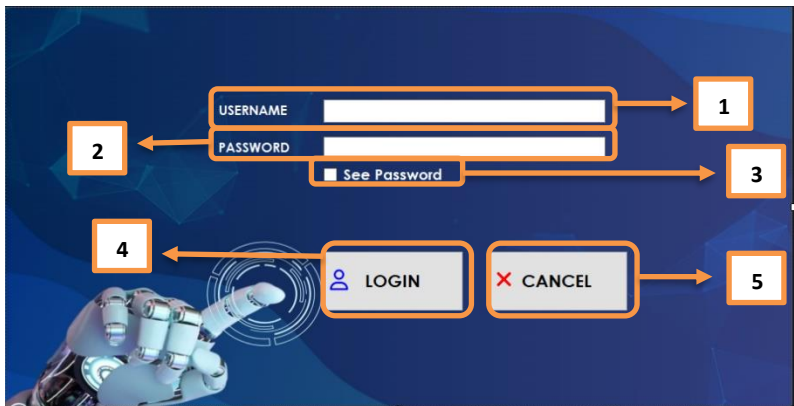
Pada ujung kiri atas terdapat logo Politeknik negeri batam, pada sudut kanan terdapat 3 tombol yaitu tombol *minimize*, *maximize*, dan yang terakhir adalah tombol *close*. Dibawahnya terdapat pilihan menu untuk *Login*, *Autorun* dan *Manual I/O*. Menu *login* berfungsi untuk fitur keamanan yang digunakan sebelum masuk pada menu *autorun*, pengguna harus memasukkan *id* dan *password* sebelum menggunakan menu selanjutnya, jika *id* dan *password* belum dimasukkan maka menu *autorun* dan *Manual I/O* tidak dapat diakses.



Gambar 18. Tampilan awal GUI

**Tabel 5. Tabel user guide**

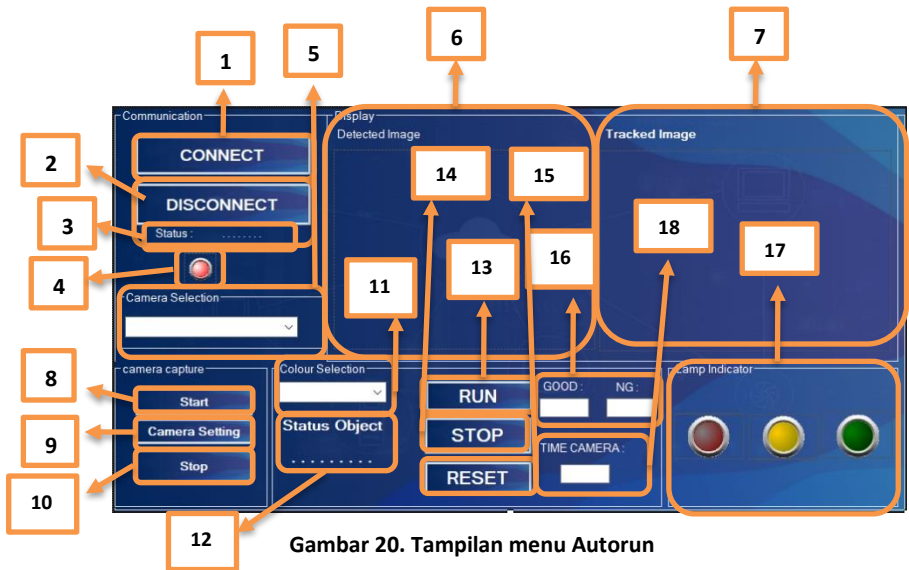
No	Fitur	Keterangan
1.	<i>Close, Minimize, and Maximize</i>	Tombol <i>Close</i> : Menutup jendela aplikasi. Tombol <i>Minimize</i> : Menyimpan jendela ke <i>taskbar/dock</i> . Tombol <i>Maximize</i> : Memperbesar jendela hingga penuh layar.
2.	<i>Login</i>	Meningkatkan Keamanan Sistem dan mengatur hak akses pengguna.
3.	<i>Autorun</i>	Kontrol mesin terhadap HMI dengan berbagai macam fitur dan tombol.
4.	<i>Manual I/O</i>	Mengontrol I/O secara manual



**Gambar 19. tampilan menu login**

Tabel 6. Tabel *user guide login*

No	Fitur	Keterangan
1.	<i>Username</i>	Memasukkan karakter <i>username</i> yang benar, dengan tujuan untuk mengatur hak akses pada pengguna.
2.	<i>Password</i>	Rangkaian karakter rahasia untuk mengamankan akses. Pengguna harus memasukkan <i>password</i> yang benar saat <i>login</i> atau mengakses sistem.
3.	<i>See Password</i>	Fitur ini Memungkinkan pengguna melihat karakter <i>password</i> yang dimasukkan dalam bentuk teks biasa.
4.	<i>Login</i>	Fitur tombol <i>login</i> pada HMI adalah untuk mengautentikasi pengguna dan memberikan akses ke sistem dengan memasukkan <i>username</i> dan <i>password</i> yang benar.
5.	<i>Cancel</i>	Fitur tombol " <i>Cancel</i> " pada HMI adalah untuk membatalkan aksi atau proses yang sedang berlangsung.

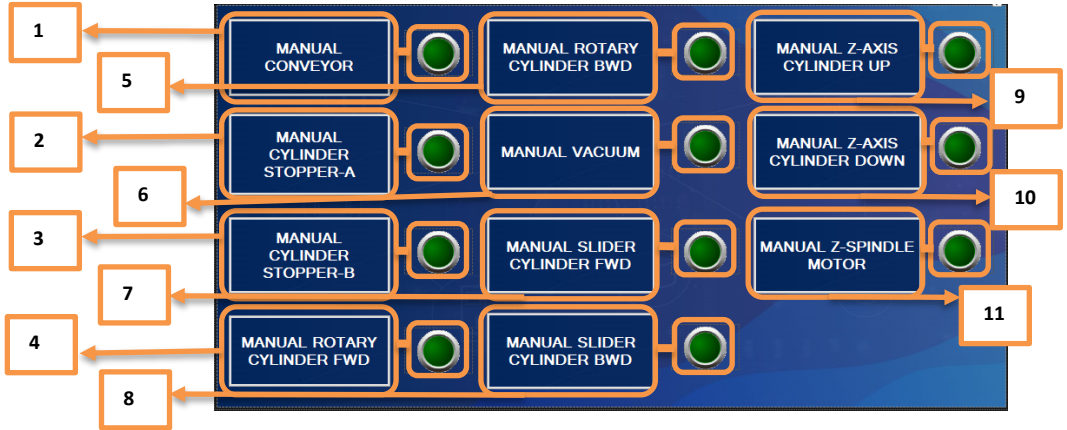


Gambar 20. Tampilan menu Aurorun

**Tabel 7. Tabel user guide Autorun**

No	Fitur	Keterangan
1.	<i>Connect</i>	Tombol connect yang berfungsi untuk mengkoneksikan komunikasi antara PLC dan HMI
2.	<i>Disconnect</i>	Tombol <i>Disconnect</i> digunakan untuk memutuskan komunikasi antara PLC dan HMI
3.	Status	Fitur ini berfungsi untuk menampilkan status dari pembacaan komunikasi antara PLC dan HMI
4.	Lampu indikator	Lampu indikator ini memunculkan warna hijau jika PLC terkoneksi dengan HMI, dan berwarna merah jika tidak terkoneksi atau terputus koneksi nya antara PLC dan HMI
5.	<i>Camera selection box</i>	Memungkinkan pengguna memilih kamera yang ingin digunakan untuk tampilan atau proses tertentu.
6.	<i>Detected image box</i>	Menampilkan gambar yang telah diidentifikasi atau terdeteksi dengan adanya modifikasi atau pemrosesan tertentu.
7.	<i>Tracked image box</i>	Menampilkan gambar yang sedang dilacak atau diikuti pergerakannya dalam waktu nyata.
8.	<i>Start</i>	Mengatur pengaturan pembacaan pada kamera secara manual.
9.	<i>Camera setting button</i>	Mengakses atau mengatur pengaturan kamera.
10.	<i>Stop camera button</i>	Memutus tampilan pada kamera secara manual
11.	<i>Colour selection box</i>	Mengatur box pengaturan warna yang akan ditangkap oleh kamera.
12.	<i>Status object</i>	Menampilkan pembacaan ketika kamera mendeteksi object atau tidak mendeteksi object
13.	<i>Run Button</i>	Tombol yang berfungsi untuk memulai operasi pada mesin
14.	<i>Stop Button</i>	Tombol yang berfungsi untuk menghentikan operasi pada mesin
15.	<i>Reset button</i>	Tombol yang berfungsi untuk mengembalikan kondisi mesin kembali ke awal dan untuk membersihkan alarm yang muncul
16.	<i>Indicator counter</i>	Menampilkan kondisi perhitungan pembacaan kamera ketika mendeteksi warna yang sesuai dan tidak sesuai

17.	<i>Lampu indicator</i>	Memberikan <i>informasi</i> secara visual tentang kondisi mesin
18.	<i>Indicator time camera</i>	Memberikan <i>informasi</i> secara visual tentang pembacaan waktu <i>delay</i> yang terjadi ketika camera membaca kondisi warna yang " <i>good</i> " dan " <i>not good</i> ".



Gambar 21. Tampilan menu Autorun

**Tabel 8. Tabel user guide MANUAL I/O**

No	Fitur	Keterangan
1.	<i>Manual Conveyor &amp; Lamp Indicator</i>	<ul style="list-style-type: none"> <li>• Tombol ini digunakan untuk mengontrol <i>conveyor</i> secara manual, memungkinkan pengguna untuk menyalakan atau mematikan <i>conveyor</i> sesuai kebutuhan.</li> <li>• Lamp indicator memberikan indikasi visual tentang status sistem <i>conveyor</i></li> </ul>
2.	<i>Manual Cylinder Stopper-A &amp; Lamp Indicator</i>	<ul style="list-style-type: none"> <li>• Tombol ini dapat digunakan untuk mengaktifkan atau menonaktifkan silinder stopper-A secara manual, menghentikan atau menggerakkan stopper sesuai kebutuhan.</li> <li>• Lampu indikator memberikan indikasi visual mengenai status silinder stopper-A, apakah sedang dalam posisi aktif atau non-aktif.</li> </ul>
3.	<i>Manual Cylinder Stopper-B &amp; Lamp Indicator</i>	<ul style="list-style-type: none"> <li>• Tombol ini dapat digunakan untuk mengaktifkan atau menonaktifkan silinder stopper-B secara manual, menghentikan atau menggerakkan stopper sesuai kebutuhan.</li> <li>• Lampu indikator memberikan indikasi visual mengenai status silinder stopper-A, apakah sedang dalam posisi aktif atau non-aktif.</li> </ul>
4.	<i>Manual Rotary Cylinder FWD &amp; Lamp Indicator</i>	<ul style="list-style-type: none"> <li>• Tombol ini digunakan untuk menggerakkan <i>rotary cylinder</i> ke arah maju (<i>Forward</i>) secara manual.</li> <li>• Lampu indikator memberikan sinyal visual mengenai status operasi <i>rotary cylinder</i>, seperti apakah silinder sedang bergerak maju atau tidak.</li> </ul>
5.	<i>Manual Rotary Cylinder BWD &amp; Lamp Indicator</i>	<ul style="list-style-type: none"> <li>• Tombol ini digunakan untuk menggerakkan <i>rotary cylinder</i> ke arah maju (<i>Backward</i>) secara manual.</li> <li>• Lampu indikator memberikan sinyal visual mengenai status operasi <i>rotary cylinder</i>, seperti apakah silinder sedang bergerak mundur atau tidak.</li> </ul>

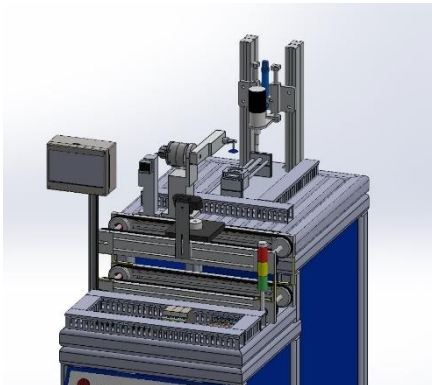
6.	<i>Manual Vacuum &amp; Lamp Indicator</i>	<ul style="list-style-type: none"> <li>• Tombol ini digunakan untuk mengontrol sistem vakum secara manual.</li> <li>• Lampu indikator memberikan indikasi visual mengenai status operasional sistem vakum, seperti apakah vakum sedang aktif atau tidak.</li> </ul>
7.	<i>Manual Slider Cylinder FWD &amp; Lamp Indicator</i>	<ul style="list-style-type: none"> <li>• Tombol ini digunakan untuk menggerakkan <i>slider cylinder</i> ke arah maju (<i>Forward</i>) secara manual.</li> <li>• Lampu indikator memberikan sinyal visual mengenai status operasi slider cylinder, seperti apakah silinder sedang bergerak maju atau tidak.</li> </ul>
8.	<i>Manual Slider Cylinder BWD &amp; Lamp Indicator</i>	<ul style="list-style-type: none"> <li>• Tombol ini digunakan untuk menggerakkan <i>slider cylinder</i> ke arah maju (<i>Backward</i>) secara manual.</li> <li>• Lampu indikator memberikan sinyal visual mengenai status operasi slider cylinder, seperti apakah silinder sedang bergerak mundur atau tidak.</li> </ul>
9.	<i>Manual Z-Axis Cylinder Up &amp; Lamp Indicator</i>	<ul style="list-style-type: none"> <li>• Tombol ini digunakan untuk menggerakkan silinder Z-Axis ke atas (<i>Up</i>) secara manual.</li> <li>• Lampu indikator memberikan sinyal visual mengenai status operasi silinder Z-Axis, seperti apakah silinder sedang bergerak ke atas atau tidak.</li> </ul>
10.	<i>Manual Z-Axis Cylinder Down &amp; Lamp Indicator</i>	<ul style="list-style-type: none"> <li>• Tombol ini digunakan untuk menggerakkan silinder Z-Axis ke atas (<i>Down</i>) secara manual.</li> <li>• Lampu indikator memberikan sinyal visual mengenai status operasi silinder Z-Axis, seperti apakah silinder sedang bergerak ke bawah atau tidak.</li> </ul>
11.	<i>Manual Z-Spindle Motor &amp; Lamp Indicator</i>	<ul style="list-style-type: none"> <li>• Tombol ini digunakan untuk mengaktifkan atau menonaktifkan motor spindle pada sumbu Z secara manual.</li> <li>• Lampu indikator memberikan sinyal visual mengenai status operasi motor</li> </ul>

		spindle pada sumbu Z, seperti apakah motor sedang aktif atau tidak.
--	--	---

### 3.1.2. Perancangan Mekanikal

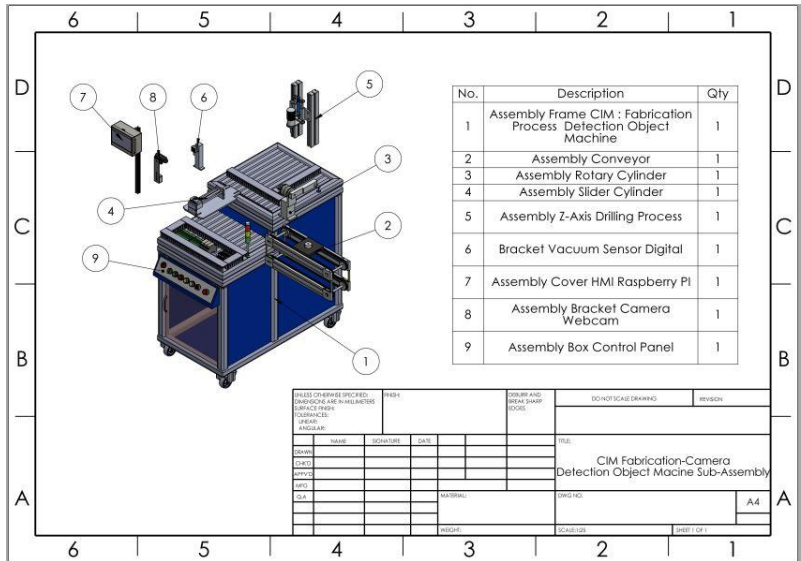


**Gambar 22. Mesin tampak seluruhnya**

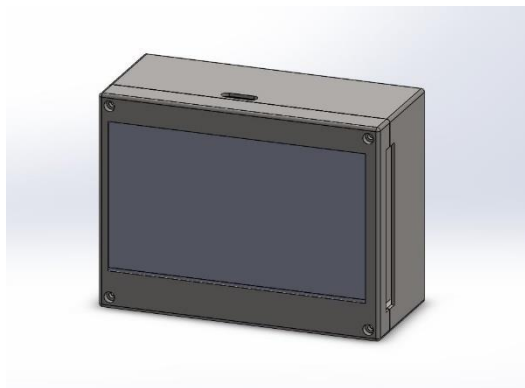


**Gambar 23. Mesin tampak diperbesar**





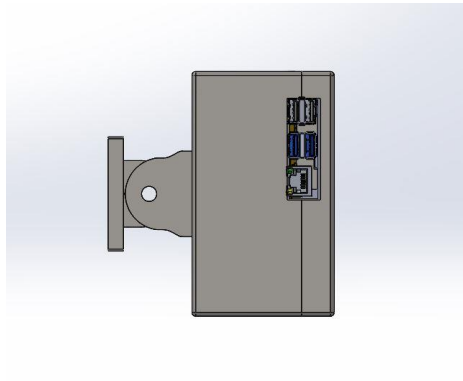
Gambar 26. Sub-assembly mesin



Gambar 27. HMI tampak seluruhnya

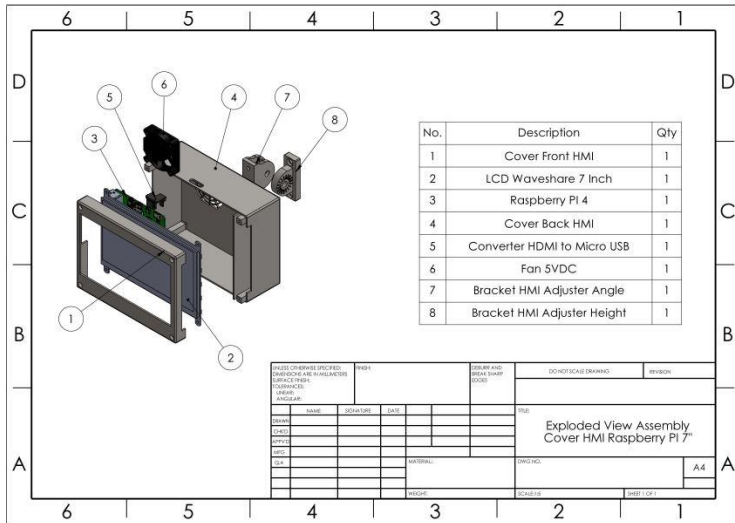


**Gambar 28. HMI tampak belakang**



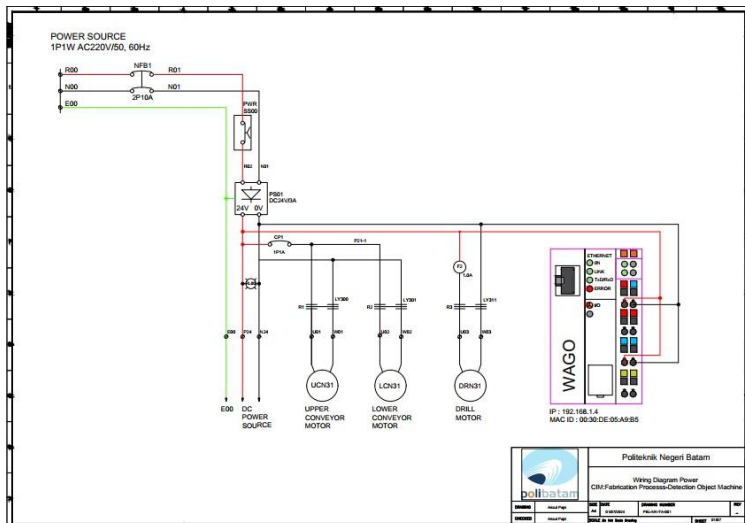
**Gambar 29. HMI tampak samping**





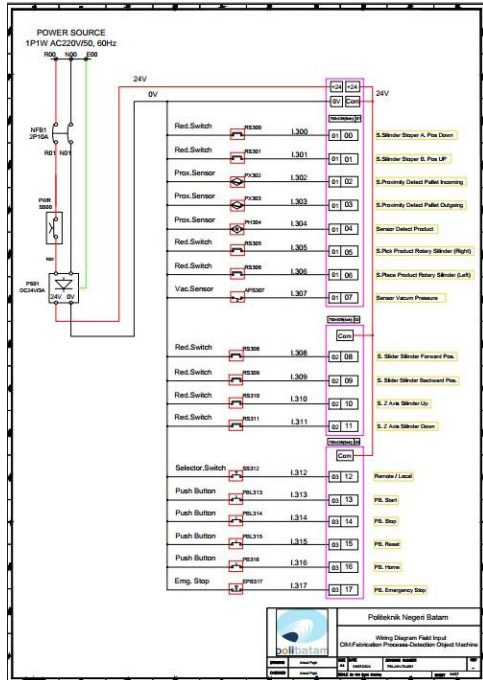
Gambar 32. Sub-assembly cover HMI

### 3.1.3. Perancangan Elektrikal



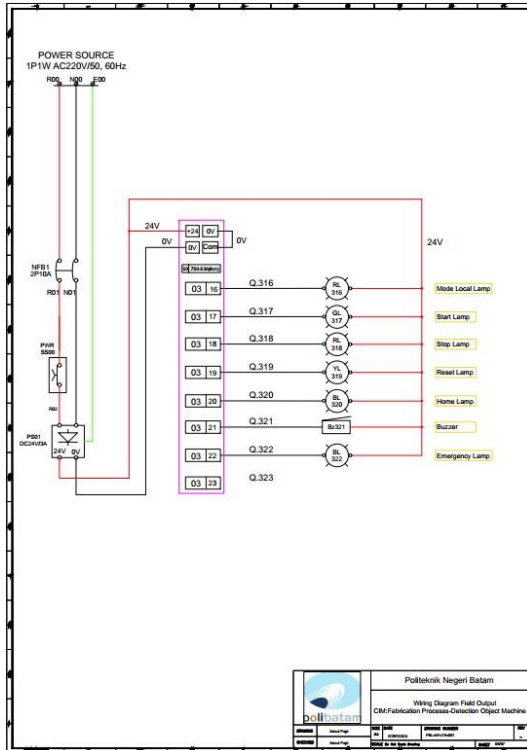
Gambar 33. Wiring diagram power-1



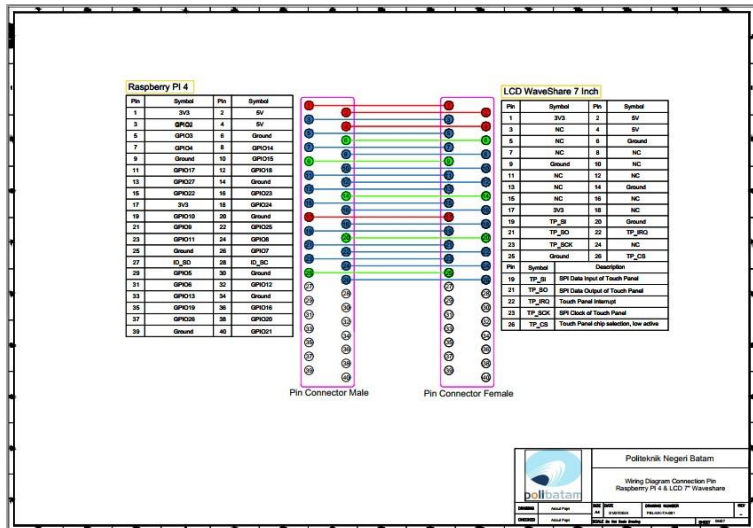


**Gambar 36. Wiring diagram field input**





**Gambar 38. Wiring diagram field output-2**



Gambar 39. Wiring diagram connection pin Raspberry to LCD

### 3.2. Alat dan Bahan

Tabel no 9 merupakan estimasi biaya yang dibutuhkan dalam merancang alat “Human machine interface dengan Raspberry Pi” diantaranya adalah:

Tabel 9. Tabel Alat dan Bahan

No.	Alat/bahan	Harga Satuan (Rp.)	Jumlah	Total (Rp.)	Keterangan
1.	Raspberry Pi	1,600,000	1	1.600,000	Disediakan oleh kampus
2.	LCD	850,000	1	850,000	Disediakan oleh kampus
3.	Webcam	150,000	1	150,000	
4.	PLC Modicon TM241CE40R	9,200,000	1	9,200,000	
5.	Kabel Ethernet 3 meter	15,000	1	15,000	
6.	Switch/Hub TP-Link	100,000	1	100,000	

	5 Port Gigabit (TLSF1005D)				
7.	Adaptor 5 volt	60,000	1	60,000	
8.	Kabel Converter mini usb to HDMI	140,000	1	140,000	
9.	Cover HMI	0	1	0	
10.	Bracket Kamera	250,000	1	250,000	
11.	Stand HMI	300,000	1	300,000	
	Total			12,665,000	

### 3.3. Pengujian

Dalam pengujian ini, data yang akan didapatkan adalah perbandingan lamanya waktu yang dibutuhkan oleh HMI saat dijalankan oleh Raspberry Pi ataupun laptop dalam mengirim sinyal hasil pendeteksian kamera yang akan dikirim ke PLC.

#### 3.3.1. Pengujian kamera terhadap benda

Pengujian kamera deteksi Warna terhadap latar belakang dan pencahayaan menggunakan Raspberry Pi (RASPI). Pengujian ini menggunakan kamera *webcam* untuk mendeteksi objek berwarna merah dengan latar belakang objek yang berwarna putih yang terhubung dengan Raspberry Pi yang bertugas untuk memproses gambar yang muncul dari kamera dan untuk mendeteksi warna.



**Gambar 40. Pengujian menggunakan HMI**

Pengujian kamera deteksi Warna terhadap latar belakang dan pencahayaan menggunakan Laptop ASUS TUF FX5601. Pengujian ini dilakukan dengan

menggunakan laptop yang terhubung dengan kamera *webcam* untuk dilakukan perbandingan pada 2 pengujian menggunakan *devices* yang berbeda



**Gambar 41. Pengujian menggunakan laptop**

### 3.3.2. Pengujian komunikasi HMI & PLC

Hasil Pengujian Komunikasi PLC dan HMI Raspberry Pi 4

Tabel 10. Tabel pengujian komunikasi

No	Mode	Description	Address			Value		Command	Status
			HMI Register	PLC Word	PLC Bit	HMI (Int)	PLC		
1	RUN AUTO	PB Run	[0]	%M0.0	%M0.0	1	TRUE	Write	Good
2		PB Stop	[0]	%M0.0	%M0.0	0	FALSE	Write	Good
3		Trigger Camera	[1]	%M1.1	%M2.0	1	TRUE	Read	Good
4		Camera Detected	[2]	%M2.2	%M4.0	1	TRUE	Write	Good
5		Camera Not Detected	[2]	%M2.2	%M4.1	2	TRUE	Write	Good
6		Reset Memory Camera	[3]	%M3.3	%M6.0	1	TRUE	Read&Write	Good
7		Indicator Stop Condition	[5]	%M5.5	%M0.0	1	TRUE	Read	Good
8		Indicator HomeReady Condition	[5]	%M5.5	%M0.1	2	TRUE	Read	Good
9		Indicator Run Condition	[6]	%M6.6	%M0.2	4	TRUE	Read	Good
10		Alarm Transfer Error	[7]	%M7.7	%M4.0	1	TRUE	Read	Good
11		Alarm Vacuum Failure	[7]	%M7.7	%M4.1	2	TRUE	Read	Good
12		Alarm Product no Detect	[7]	%M7.7	%M4.2	4	TRUE	Read	Good
13		PB Reset	[8]	%M8.8	%M6.0	1	TRUE	Write	Good
14		Indicator Connect Success	[9]	%M9.9	%M6.0	1	TRUE	Read	Good
15		Holding Alarm Status	[10]	%M10.10	%M20.0	1	TRUE	Read	Good
16		Counter Good	[11]	%M11.11	-	same with PLC	same with PLC	Read	Good
17		Counter Not Good	[12]	%M12.12	-	same with PLC	same with PLC	Read	Good
18		Delayed Time (ms)	[14]	%M14.14	-	same with PLC	same with PLC	Read	Good
19		Manual Conveyor	[15]	%M15.15	%M30.0	1	TRUE	Write	Good
20	Manual Cylinder Stopper-A	[16]	%M16.16	%M32.0	1	TRUE	Write	Good	
21	Manual Cylinder Stopper-B	[17]	%M17.17	%M34.0	1	TRUE	Write	Good	
22	Manual Rotary Cylinder FWD	[18]	%M18.18	%M36.0	1	TRUE	Write	Good	
23	Manual Rotary Cylinder BWD	[18]	%M18.18	%M38.1	2	TRUE	Write	Good	
24	Manual Vacuum	[19]	%M19.19	%M38.0	1	TRUE	Write	Good	
25	Manual Slider Cylinder FWD	[20]	%M20.20	%M40.0	1	TRUE	Write	Good	
26	Manual Slider Cylinder BWD	[20]	%M20.20	%M40.1	2	TRUE	Write	Good	
27	Manual 2-Axis Cylinder Up	[21]	%M21.21	%M42.0	1	TRUE	Write	Good	
28	Manual 2-Axis Cylinder Down	[21]	%M21.21	%M42.1	2	TRUE	Write	Good	
29	Manual Spindle Motor	[22]	%M22.22	%M44.0	1	TRUE	Write	Good	
30	RUN MANUAL	Status - Sensor Cylinder Stopper-A Pos Down	[23]	%M23.23	%M46.0	1	TRUE	Read	Good
31	Status - Sensor Cylinder Stopper-B Pos Up	[24]	%M24.24	%M48.0	1	TRUE	Read	Good	
32	Status - Sensor Proximity Detect Pallet Incoming	[25]	%M25.25	%M50.0	1	TRUE	Read	Good	
33	Status - Sensor Proximity Detect Pallet Outgoing	[26]	%M26.26	%M52.0	1	TRUE	Read	Good	
34	Status - Sensor Detect Product	[27]	%M27.27	%M54.0	1	TRUE	Read	Good	
35	Status - Sensor Rotary Cylinder Front	[28]	%M28.28	%M56.0	1	TRUE	Read	Good	
36	Status - Sensor Rotary Cylinder Back	[29]	%M29.29	%M58.0	1	TRUE	Read	Good	
37	Status - Sensor Vacuum Pressure	[30]	%M30.30	%M60.0	1	TRUE	Read	Good	
38	Status - Sensor Slider Cylinder Pos FWD	[31]	%M31.31	%M62.0	1	TRUE	Read	Good	
39	Status - Sensor Slider Cylinder Pos BWD	[32]	%M32.32	%M64.0	1	TRUE	Read	Good	
40	Status - Sensor Cylinder 2-Axis L/F	[33]	%M33.33	%M66.0	1	TRUE	Read	Good	
41	Status - Sensor Cylinder 2-Axis Down	[34]	%M34.34	%M68.0	1	TRUE	Read	Good	

Penjelasan dari tabel 10 adalah sebagai berikut:

*Description*: sinyal atau *variable* yang diuji

*Address*: alamat data yang digunakan oleh HMI dan PLC.

- *Word*: Alamat memori yang digunakan pada PLC untuk menyimpan data tersebut
- HMI: Menyatakan alamat memori pada HMI yang digunakan untuk membaca atau menulis perintah
- PLC: Menyatakan alamat memori pada PLC yang digunakan untuk membaca atau menulis perintah

*Value*: nilai pada data yang dikirimkan.

- HMI: Nilai data yang terbaca atau ditulis pada HMI
- PLC: Nilai data yang terbaca atau ditulis pada PLC

*Command*: perintah yang diberikan oleh HMI pada PLC.

Status: hasil dari pengujian yang dilakukan apakah bekerja dengan baik atau tidak.

Pengujian ini bertujuan untuk memastikan bahwa HMI dapat berkomunikasi dan bertukar data dengan PLC sehingga dapat melakukan pengontrolan dan *monitoring* pada mesin dengan benar. Selain itu juga untuk memvalidasi *Address/tag* yang digunakan itu sinkron sehingga tidak terjadi kesalahan pembacaan nilai data yang salah. Dari data *di atas* dapat dilihat bahwa pengujian yang dilakukan untuk memeriksa apakah data yang dikirim atau ditampilkan oleh HMI sesuai dengan data yang diproses oleh PLC. Hal ini juga dilakukan untuk memastikan bahwa tidak ada kesalahan data pada saat melakukan *trial & errors* pada program PLC yang dibuat.

### 3.3.3. Pengujian *delay* waktu kamera

Gambar nomor 42 merupakan hasil dari pengujian *delay* kamera ketika mendeteksi warna. Pengujian dilakukan dengan memilih mode deteksi berwarna biru dengan benda berwarna putih dan latar belakang benda berwarna hitam, didapati hasil "FALSE" dikarenakan kamera tidak mendeteksi warna yang dipilih sebelumnya yaitu warna biru, maka didapati waktu pembacaan kamera ketika mendeteksi sekitar 2,4 detik.



Gambar 42. Pengujian waktu *delay* pembacaan kamera

### **3.3.4. Pengujian data yang diolah oleh HMI**

Pada pengujian ini, HMI akan diuji apakah data yang ditampilkan sesuai dengan keadaan nyata atau tidak. Data yang akan diperoleh dari HMI berupa file yang akan merekam dan mencatat apabila terjadi kesalahan pada mesin. Selain itu, HMI dapat menampilkan data mengenai jumlah barang yang terdeteksi oleh kamera, apakah sesuai dengan pengaturan atau tidak. HMI juga dapat menampilkan data mengenai waktu tunda yang dibutuhkan kamera untuk mendeteksi benda.

## Bab 4. Hasil dan Pembahasan

### 4.1. Data Pengujian kamera terhadap benda

Pengujian ini bertujuan untuk mendapatkan hasil dari pendeteksian oleh kamera. Sebanyak 20 sampel diuji dari tiap warna untuk mengevaluasi kinerja kamera pada dua perangkat yang berbeda, yaitu Raspberry Pi sesuai dengan judul utama pada penelitian ini, dan juga laptop untuk membandingkan kinerja kamera antara Raspberry Pi dan laptop. Hasil pengujian tersebut kemudian dianalisis menggunakan statistik deskriptif untuk memberikan gambaran umum tentang tingkat keberhasilan deteksi benda.

Pengujian kamera deteksi warna putih menggunakan HMI Raspberry Pi.

**Tabel 11. Tabel pengujian warna putih menggunakan HMI Raspberry Pi**

No	pencahayaannya (Lux)		latar belakang				Hasil	
	4	185	Putih	Merah	Biru	Hitam	Detected	Not Detected
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Pengujian kamera deteksi warna putih menggunakan HMI Laptop.

**Tabel 12. Tabel pengujian warna putih menggunakan laptop**

No	pencahayaan (Lux)		latar belakang				Hasil	
	4	185	Putih	Merah	Biru	Hitam	Detected	Not Detected
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Pengujian kamera deteksi warna merah menggunakan HMI Raspberry Pi.

**Tabel 13. Tabel pengujian warna merah menggunakan HMI Raspberry Pi**

No	pencahayaan (Lux)		latar belakang				Hasil	
	4	185	Putih	Merah	Biru	Hitam	Detected	Not Detected
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Pengujian kamera deteksi warna merah menggunakan HMI laptop.

**Tabel 14. Tabel pengujian warna merah menggunakan laptop**

No	pencahayaan (Lux)		latar belakang				Hasil	
	4	185	Putih	Merah	Biru	Hitam	Detected	Not Detected
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
18	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Tabel 11 dan tabel 12 memperlihatkan hasil pengujian kamera terhadap benda berwarna putih, sedangkan tabel 13 dan tabel 14 memperlihatkan hasil pengujian kamera terhadap benda berwarna merah, kedua warna tersebut diuji berdasarkan parameter yang telah ditetapkan. Dari 20 sampel yang diuji dari tiap warna, semua pengujian berhasil mendeteksi objek. Tingkat keberhasilan dari pengujian yang telah dilakukan adalah 100%. Dengan tingkat keberhasilan 100%, kamera menunjukkan kinerja yang sangat baik dalam mendeteksi objek, untuk memberikan gambaran lebih lengkap mengenai keandalan tingkat keberhasilan yang dihitung, digunakan perhitungan *confidence interval* (CI) untuk proporsi binomial pada tingkat kepercayaan 95%.

$$CI = \hat{P} \pm Z \sqrt{\frac{\hat{P}(1 - \hat{P})}{n}} \tag{1}$$

Dimana:

$\hat{P}$  = proporsi sampel

Z = nilai z merupakan 95% confidence interval (1.96)

n = jumlah sampel

Dikarenakan dari pengujian kedua warna *di atas* mendapatkan hasil 100% atau  $\hat{P} = 1$ , istilah dari  $\hat{P}(1 - \hat{P})$  menjadi nol, sehingga *interval* CI tidak berpengaruh terhadap hasil pengujian dari kedua warna ini. Mengingat semua hasil pengujian yang telah dilakukan berhasil, dapat diyakini bahwa tingkat keberhasilan deteksi objek dari kedua warna di atas yaitu putih dan merah adalah sangat tinggi.

Pengujian kamera deteksi warna biru menggunakan HMI Raspberry Pi.

**Tabel 15. Tabel pengujian warna biru menggunakan Raspberry Pi**

No	pencahayaannya (Lux)		latar belakang				Hasil	
	4	185	Putih	Merah	Biru	Hitam	Detected	Not Detected
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
18	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Pengujian kamera deteksi warna biru menggunakan HMI laptop.

**Tabel 16. Tabel pengujian warna biru menggunakan laptop**

No	pencahayaan (Lux)		latar belakang				Hasil	
	4	185	Putih	Merah	Biru	Hitam	Detected	Not Detected
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
18	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Kedua tabel *di atas* memperlihatkan hasil pengujian kamera terhadap benda berwarna biru, warna tersebut diuji berdasarkan parameter yang telah ditetapkan sama seperti dua warna yang sudah diuji yaitu putih dan merah. Dari 20 sampel yang diuji hasil pengujian untuk objek berwarna biru berbeda dengan warna putih dan merah. Tingkat keberhasilan dari pengujian yang telah dilakukan adalah 70%. Dari 20 kali pengujian terlihat bahwa hasil benda tidak terdeteksi adalah ketika pencahayaan berada di 185 lux dan latar belakang berwarna putih dan merah

Dengan tingkat keberhasilan 70%, kamera menunjukkan kinerja yang baik dalam mendeteksi objek berwarna biru, untuk memberikan gambaran lebih lengkap mengenai keandalan tingkat keberhasilan yang dihitung, digunakan perhitungan *confidence interval* (CI) untuk proporsi binomial pada tingkat kepercayaan 95%.

Berdasarkan data yang telah dimiliki, didapatkan bahwa:

$\hat{P}$  = proporsi sampel (0.7 atau 70%)

Z = nilai z merupakan 95% confidence interval (1.96)

n = jumlah sampel (20)

Maka perhitungan dari *confidence interval* adalah sebagai berikut:

$$CI = 0.7 \pm 1.96 \sqrt{\frac{0.7(1 - 0.7)}{20}}$$

$$CI = 0.7 \pm 1.96 \sqrt{\frac{0.7 \times 0.3}{20}}$$

$$CI = 0.7 \pm 1.96\sqrt{0.021}$$

$$CI = 0.7 \pm 1.96 \times 0.145$$

$$CI = 0.7 \pm 0.284$$

Jadi, dengan tingkat kepercayaan 95% dapat diyakini bahwa proporsi sebenarnya dari keberhasilan deteksi objek oleh kamera pada warna biru tidak akan lebih rendah dari 41.6% dan tidak lebih dari 98.4%. hasil ini menunjukkan kinerja yang baik dari kamera.

## 4.2. Data Pengujian komunikasi HMI & PLC

Data yang didapatkan dari pengujian 2 adalah hasil pengujian komunikasi antara HMI dan juga PLC.

Tabel 17. Tabel pengujian komunikasi HMI dan PLC

No	Description	Address			Value		Command	Status
		HMI	PLC		HMI (Int)	PLC		
		Register	Word	Bit				
1	PB.Run	[0]	%MW0	%MX0.0	1	TRUE	Write	Good
2	PB.Stop	[0]	%MW0	%MX0.0	0	FALSE	Write	Good
3	Trigger Camera	[1]	%MW1	%MX2.0	1	TRUE	Read	Good
4	Camera Detected	[2]	%MW2	%MX4.0	1	TRUE	Write	Good
5	Camera Not Detected	[2]	%MW2	%MX4.1	2	TRUE	Write	Good
6	Reset Memory Camera	[3]	%MW3	%MX6.0	1	TRUE	Read&Write	Good
7	Indicator Stop Condition	[5]	%MW5	%MX0.0	1	TRUE	Read	Good
8	Indicator HomeReady Condition	[5]	%MW5	%MX0.1	2	TRUE	Read	Good
9	Indicator Run Condition	[5]	%MW5	%MX0.2	4	TRUE	Read	Good
10	Alarm Transfer Error	[7]	%MW7	%MX4.0	1	TRUE	Read	Good
11	Alarm Vacuum Failure	[7]	%MW7	%MX4.1	2	TRUE	Read	Good
12	Alarm Product no Detect	[7]	%MW7	%MX4.2	4	TRUE	Read	Good
13	PB.Reset	[8]	%MW8	%MX6.0	1	TRUE	Write	Good
14	Indicator Connect Success	[9]	%MW9	%MX8.0	1	TRUE	Read	Good
15	Holding Alarm Status	[10]	%MW10	%MX20.0	1	TRUE	Read	Good
16	Counter Good	[11]	%MW11	-	same with PLC	same with PLC	Read	Good
17	Counter Not Good	[12]	%MW12	-	same with PLC	same with PLC	Read	Good
18	Delayed Time (s)	[14]	%MW14	-	same with PLC	same with PLC	Read	Good
19	Manual Conveyor	[15]	%MW15	%MX30.0	1	TRUE	Write	Good
20	Manual Cylinder Stopper-A	[16]	%MW16	%MX32.0	1	TRUE	Write	Good
21	Manual Cylinder Stopper-B	[17]	%MW17	%MX34.0	1	TRUE	Write	Good
22	Manual Rotary Cylinder FWD	[18]	%MW18	%MX36.0	1	TRUE	Write	Good
23	Manual Rotary Cylinder BWD	[18]	%MW18	%MX36.1	2	TRUE	Write	Good
24	Manual Vacuum	[19]	%MW19	%MX38.0	1	TRUE	Write	Good
25	Manual Slider Cylinder FWD	[20]	%MW20	%MX40.0	1	TRUE	Write	Good
26	Manual Slider Cylinder BWD	[20]	%MW20	%MX40.1	2	TRUE	Write	Good
27	Manual Z-Axis Cylinder Up	[21]	%MW21	%MX42.0	1	TRUE	Write	Good
28	Manual Z-Axis Cylinder Down	[21]	%MW21	%MX42.1	2	TRUE	Write	Good
29	Manual Z-Spindle Motor	[22]	%MW22	%MX44.0	1	TRUE	Write	Good
30	Status : Sensor Cylinder Stopper-A Pos.Down	[23]	%MW23	%MX46.0	1	TRUE	Read	Good
31	Status : Sensor Cylinder Stopper-A Pos.Up	[24]	%MW24	%MX48.0	1	TRUE	Read	Good
32	Status : Sensor Proximity Detect Pallet Incoming	[25]	%MW25	%MX50.0	1	TRUE	Read	Good
33	Status : Sensor Proximity Detect Pallet Outgoing	[26]	%MW26	%MX52.0	1	TRUE	Read	Good
34	Status : Sensor Detect Product	[27]	%MW27	%MX54.0	1	TRUE	Read	Good
35	Status : Sensor Rotary Cylinder Front	[28]	%MW28	%MX56.0	1	TRUE	Read	Good
36	Status : Sensor Rotary Cylinder Back	[28]	%MW28	%MX56.1	1	TRUE	Read	Good
37	Status : Sensor Vacuum Pressure	[30]	%MW30	%MX60.0	1	TRUE	Read	Good
38	Status : Sensor Slider Cylinder Pos.FWD	[31]	%MW31	%MX62.0	1	TRUE	Read	Good
39	Status : Sensor Slider Cylinder Pos.BWD	[32]	%MW32	%MX64.0	1	TRUE	Read	Good
40	Status : Sensor Cylinder Z-Axis UP	[33]	%MW33	%MX66.0	1	TRUE	Read	Good
41	Status : Sensor Cylinder Z-Axis Down	[34]	%MW34	%MX68.0	1	TRUE	Read	Good

Pada table pengujian komunikasi PLC dan HMI Raspberry Pi dapat dijelaskan bahwa:

1. Pada saat menekan PB.Run, HMI akan menuliskan nilai *integer* 1 pada alamat *register* [0] yang artinya PLC akan membaca alamat bilangan *word* (%MW0) bernilai sama dengan 1. Sehingga jika digunakan bilangan *boolean/bit* maka alamat PLC %MX0.0 yang termasuk ke dalam bilangan *word* %MW0 akan memiliki nilai 'TRUE'. Alamat inilah yang digunakan untuk mulai mengoperasikan mesin deteksi warna.
2. Pada saat menekan PB.Stop, HMI akan menuliskan nilai *integer* 0 pada alamat *register* [0] yang artinya PLC akan membaca alamat bilangan *word* (%MW0) bernilai sama dengan 0. Sehingga jika digunakan bilangan *boolean/bit* maka alamat PLC %MX0.0 yang termasuk ke dalam bilangan

word %MW0 akan memiliki nilai 'FALSE'. Alamat inilah yang digunakan untuk menghentikan operasi mesin deteksi warna.

3. *Trigger Camera*, Pada saat sensor B / *Detection Object* mendeteksi adanya produk, maka PLC akan menulis nilai *integer* 1 pada alamat %MW1 dengan mengubah nilai %MX2.0 menjadi 'TRUE'. Artinya HMI akan membaca nilai *register* [1] memiliki nilai 1 dan mulai untuk mendeteksi warna produk.
4. *Camera Detected*, Jika hasil deteksi kamera ialah warna produk yang diinginkan, maka HMI akan menuliskan nilai *integer* 1 pada alamat *register* [2]. Artinya PLC akan membaca alamat bilangan *word* (%MW2) bernilai sama dengan 1. Sehingga jika digunakan bilangan *boolean*nya maka alamat PLC %MX4.0 yang termasuk ke dalam bilangan *word* %MW2 akan memiliki nilai 'TRUE'. Alamat inilah yang digunakan untuk melanjutkan proses fabrikasi terhadap objek yang berhasil dideteksi.
5. *Camera Not Detected*, Jika hasil deteksi kamera ialah warna produk yang tidak sesuai, maka HMI akan menuliskan nilai *integer* 2 pada alamat *register* [2]. Artinya PLC akan membaca alamat bilangan *word* (%MW2) bernilai sama dengan 2. Sehingga jika digunakan bilangan *boolean*nya maka alamat PLC %MX4.1 yang termasuk ke dalam bilangan *word* %MW2 akan memiliki nilai 'TRUE'. Alamat inilah yang digunakan untuk menghidupkan *conveyor* dan membawa produk tersebut menuju *unloading area*.
6. *Reset Memory Camera*, Jika proses pengeboran produk telah selesai maka PLC akan menuliskan nilai *integer* 1 pada alamat %MW3. Sehingga HMI akan membaca nilai *register* [3] dan menuliskan nilai 0 pada alamat *register* [2] untuk mereset status hasil deteksi kamera dan menutup tampilan screenshot pada GUI HMI nya.
7. *Indicator Stop Condition*, Jika HMI membaca nilai *register* [5] sama dengan 1 artinya alamat (%MW5) bernilai sama dengan 1 atau alamat %MX10.0 bernilai 'TRUE'.
8. *Indicator Home/Ready Condition*, Jika HMI membaca nilai *register* [5] sama dengan 2 artinya alamat (%MW5) bernilai sama dengan 2 atau alamat %MX10.1 bernilai 'TRUE'.
9. *Indicator Run Condition*, Jika HMI membaca nilai *register* [5] sama dengan 4 artinya alamat (%MW5) bernilai sama dengan 4 atau alamat %MX10.2 bernilai 'TRUE'.
10. *Alarm Transfer Error*, Jika HMI membaca nilai *register* [7] sama dengan 1 artinya alamat (%MW7) bernilai sama dengan 1 atau alamat %MX14.0 bernilai 'TRUE'.

11. *Alarm Vacuum Failure Error*, Jika HMI membaca nilai *register* [7] sama dengan 2 artinya alamat (%MW7) bernilai sama dengan 2 atau alamat %MX14.1 bernilai 'TRUE'.
12. *Alarm No Product Detected*, Jika HMI membaca nilai *register* [7] sama dengan 4 artinya alamat (%MW7) bernilai sama dengan 4 atau alamat %MX14.2 bernilai 'TRUE'.
13. Pada saat menekan PB.Home/Reset, HMI akan menuliskan nilai *integer* 1 pada Alamat *register* [8] yang artinya PLC akan membaca alamat bilangan *word* (%MW8) bernilai sama dengan 1. Sehingga jika digunakan bilangan *booleanya* maka alamat PLC %MX16.0 yang termasuk ke dalam bilangan *word* %MW8 akan memiliki nilai 'TRUE'. Alamat inilah yang digunakan untuk reset alarm status dan mengembalikan semua silinder ke *initial position*.
14. *Indicator Connect Succes*, Jika HMI membaca nilai *register* [9] sama dengan 1 artinya alamat (%MW9) bernilai sama dengan 1 atau alamat %MX18.0 bernilai 'TRUE'.
15. *Holding Alarm Status*, Jika HMI membaca nilai *register* [10] sama dengan 1 artinya alamat (%MW10) bernilai sama dengan 1 atau alamat %MX20.0 bernilai 'TRUE'.
16. *Counter Good*, HMI membaca jumlah produk yang sesuai warna menggunakan Alamat *register* [11] melalui program *counter* yang ada pada PLC dengan menggunakan alamat *word* (%MW11).
17. *Counter Not Good*, HMI membaca jumlah produk yang tidak sesuai warna menggunakan alamat *register* [12] melalui program *counter* yang ada pada PLC dengan menggunakan alamat *word* (%MW12).
18. *Delayed time Camera (s)*, HMI membaca jumlah produk yang tidak sesuai warna menggunakan Alamat *register* [14] melalui program *counter timer* yang ada pada PLC dengan menggunakan alamat *word* (%MW14).
19. *Manual Conveyor – Manual Spindle Motor*, HMI menulis nilai *integer* pada alamat *word* sesuai nilai dan alamat tabel pengujian di atas, sehingga PLC merespon untuk menggerakkan *actuator* secara manual.
20. *Status Sensor*, HMI membaca nilai *word* pada alamat *register* sesuai dengan data yang dikirimkan oleh PLC sesuai dengan kondisi aktual sensor yang ingin dimonitoring oleh HMI.

Proses pertukaran data yang digunakan adalah *Write/Read Holding Register* yaitu hanya dapat bertukar nilai data dengan memasukkan bilangan *word/integer*. Artinya perlu menggunakan metode BCD / *Binary Convert Decimal* untuk mengubah bilangan desimal/*integer* menjadi bilangan biner jika ingin menggunakan setiap alamat bit di dalam bilangan *word* yang digunakan. Sebagai

contoh alamat *word* yang digunakan yaitu %MW0.0 yang memiliki alamat bit %MX0.0 - %MX1.16. Lihat tabel kebenaran BCD berikut:

**Tabel 18. Tabel kebenaran BCD**

Angka Desimal	BCD	Bit (%MW0) Aktif / <i>True</i>
1	0001	%MX0.0
2	0010	%MX0.1
3	0011	%MX0.0 & %MX0.1
4	0100	%MX0.2
5	0101	%MX0.2 & %MX0.0
6	0110	%MX0.2 & %MX0.1
7	0111	%MX0.2 & %MX0.1 & %MX0.0
8	1000	%MX0.3
9	1001	%MX0.3 & %MX0.0

Dari table pengujian 17 dapat disimpulkan bahwa pengujian komunikasi data antara HMI dengan PLC dapat berkomunikasi dengan baik. Terbukti dari status atau hasil pengujian yang menyatakan bahwa hasil pertukaran data antara HMI dan PLC tidak terdapat kesalahan satupun. Berikut perhitungan persentase keberhasilan pada pengujian komunikasi Modbus TCP/IP:

Keterangan:

*G*: Status berhasil/baik

*NG*: Status gagal

*N*: Jumlah data pengujian

%*Good*: Persentase berhasil

$$\%Good = \frac{G-NG}{N} \times 100\% \quad (2)$$

$$\%Good = \frac{18-0}{18} \times 100\%$$

$$\%Good = 100\%$$

Jadi, Jika menggunakan perhitungan *confidence interval* (CI) yang telah dilakukan pada pengujian 1, untuk proporsi binomial pada tingkat kepercayaan 95%, maka hasil dari perhitungan CI tidak berpengaruh besar terhadap pengujian, dikarenakan hasil pengujian yang dilakukan berhasil dan tidak terdapat kesalahan satupun. Dapat dipastikan bahwa tingkat keberhasilan komunikasi Modbus TCP/IP antara HMI dan PLC sangat baik.

### 4.3. Data Pengujian *delay* waktu kamera

Pengujian ini dilakukan untuk mendapatkan data *delay* waktu yang dibutuhkan HMI untuk bisa mendeteksi warna. Dalam pengujian ini HMI Raspberry Pi akan dibandingkan dengan HMI yang akan dijalankan laptop, berikut perbandingan spesifikasi antara Raspberry Pi dan Juga laptop yang digunakan dalam pengujian ini.

#### Tabel spesifikasi Raspberry Pi

- Processor: Broadcom BCM2837B0, Cortex-A5364-bit SoC @ 1.4GHz.
- Memory: 1GB LPDDR2 SDRAM.
- OS: Windows10 on raspberry

#### Tabel spesifikasi Laptop TUF FA506\_FX506II

- Memory: 8 GB
- Processor: AMD ryzen5 4600H with radeon graphic
- Graphic card: Nvidia Geforce GTX 1650 TI
- OS: Windows 10

Tabel nomor 19 adalah hasil pengujian setelah dilakukan perhitungan waktu yang dibutuhkan oleh HMI untuk bisa mendeteksi warna

**Tabel 18. Tabel *delay* waktu**

No	Warna	Result	Delayed Time (s)		Comparison
			Laptop/PC	HMI Raspberry	
1	Putih	Detected	2.68	7.08	4.40
2	Putih	Detected	2.00	6.75	4.75
3	Putih	Detected	2.23	6.82	4.59
4	Putih	Detected	2.88	7.12	4.24
5	Putih	Detected	2.56	7.28	4.72
6	Putih	Not Detected	2.64	6.72	4.08
7	Putih	Not Detected	3.01	7.34	4.33
8	Putih	Not Detected	2.76	6.57	3.81
9	Putih	Not Detected	3.22	5.78	2.56
10	Putih	Not Detected	3.83	8.12	4.29
11	Merah	Detected	1.89	6.93	5.04
12	Merah	Detected	2.00	8.37	6.37
13	Merah	Detected	2.30	6.15	3.85
14	Merah	Detected	2.46	7.24	4.78
15	Merah	Detected	2.94	7.73	4.79
16	Merah	Not Detected	3.42	5.81	2.39
17	Merah	Not Detected	2.92	6.75	3.83
18	Merah	Not Detected	2.73	5.82	3.09
19	Merah	Not Detected	2.89	7.13	4.24
20	Merah	Not Detected	1.97	5.65	3.68
21	Biru	Detected	2.96	8.43	5.47
22	Biru	Detected	3.15	7.96	4.81
23	Biru	Detected	2.77	8.15	5.38
24	Biru	Detected	3.33	6.81	3.48
25	Biru	Detected	3.56	7.38	3.82
26	Biru	Not Detected	2.42	5.43	3.01
27	Biru	Not Detected	2.58	7.64	5.06
28	Biru	Not Detected	3.76	7.78	4.02
29	Biru	Not Detected	3.24	6.54	3.30
30	Biru	Not Detected	3.89	7.32	3.43
Average			<b>2.83</b>	<b>7.02</b>	<b>4.19</b>

Tabel *di atas* memperlihatkan *delay* waktu yang dibutuhkan oleh dua perangkat yang digunakan pada pengujian ini yaitu laptop dan juga Raspberry Pi untuk bisa mendeteksi warna, dengan data yang dimiliki maka akan dibuatkan perhitungan mengenai rata-rata, nilai tengah, dan juga standar deviasi.

Untuk mendapatkan nilai rata-rata dari data *di atas* maka digunakan rumus sebagai berikut:

**1. Perhitungan nilai rata – rata (*mean*)**

$$\text{Rata – rata (mean)} = \frac{\text{jumlah nilai data}}{\text{banyaknya data}} \quad (3)$$

a) Rata-rata *delay* waktu pada laptop:

$$\text{Rata - rata (mean)} = \frac{84.99}{30}$$

$$\text{Rata - rata (mean)} = 2.833$$

- b) Rata-rata *delay* waktu pada Raspberry Pi:

$$\text{Rata - rata (mean)} = \frac{210.6}{30}$$

$$\text{Rata - rata (mean)} = 7.02$$

- c) Rata-rata *delay* waktu perbandingan antara laptop dan Raspberry Pi:

$$\text{Rata - rata (mean)} = \frac{125.61}{30}$$

$$\text{Rata - rata (mean)} = 4.19$$

## 2. Perhitungan nilai *median*

Selanjutnya perhitungan yang akan digunakan adalah nilai tengah, untuk mendapatkan nilai tengah dari data *di atas* maka digunakan rumus sebagai berikut:

$$\text{Me} = [(n \div 2) + (n \div 2) + 1] \div 2 \quad (4)$$

Dengan menggunakan rumus median jumlah data genap, untuk menentukan dua nilai tengah dari 3 data *di atas* adalah sebagai berikut:

$$\text{Me} = (n \div 2)$$

$$\text{Me} = (30 \div 2)$$

$$\text{Me} = 15$$

Me 15 atau nilai tengah pertama dari ketiga data *di atas* setelah diurutkan dari nilai terendah sampai terbesar adalah:

- a) Laptop = 2.77
- b) Raspberry Pi = 7.08
- Comparison = 4.24

$$\text{Me} = (n \div 2) + 1$$

$$\begin{aligned}
 Me &= (30 \div 2) + 1 \\
 Me &= 15 + 1 \\
 Me &= 16
 \end{aligned}$$

Me 16 atau nilai tengah kedua dari ketiga data *di atas* setelah diurutkan dari nilai terendah sampai terbesar adalah:

- a) Laptop = 2.88
  - b) Raspberry Pi = 7.12
- Comparison* = 4.24

Setelah mendapatkan dua nilai tengah dari ketiga *di atas* maka akan didapatkan nilai tengah dengan cara sebagai berikut:

$$Me = (\text{data ke 15} + \text{data ke 16}) \div 2$$

Untuk mendapatkan nilai tengah dari laptop:

$$\begin{aligned}
 Me &= (2.77 + 2.88) \div 2 \\
 Me &= (5.65) \div 2 \\
 Me &= 2.825
 \end{aligned}$$

Untuk mendapatkan nilai tengah dari Raspberry Pi:

$$\begin{aligned}
 Me &= (7.08 + 7.12) \div 2 \\
 Me &= (14.2) \div 2 \\
 Me &= 7.1
 \end{aligned}$$

Dikarenakan nilai tengah pertama dan kedua pada *comparison* sama, maka nilai tengah dari *comparison* adalah 4.24

### 3. Perhitungan nilai standar deviasi

Selanjutnya adalah menentukan nilai standar deviasi dari tabel *di atas*, untuk mendapatkan standar deviasi maka dibutuhkan rumus sebagai berikut:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1}} \tag{5}$$

Yang dimana:

$\sigma$  = Standar Deviasi (SD)

$n$  = ukuran sampel

$X_i$  = Nilai  $x$  ke  $i$

$\bar{X}$  = Nilai rata-rata

Setelah mendapatkan nilai rata-rata dari ketiga data *di atas*, selanjutnya kurangi data waktu dengan rata-rata yang sudah didapatkan.

**Tabel 19. Pengujian Waktu *delay* laptop**

No	delayed time laptop	nilai rata-rata	time - rata-rata	kuadrat perbedaan
1	2.68	2.833	-0.15	0.023409
2	2.00	2.833	-0.83	0.693889
3	2.23	2.833	-0.60	0.363609
4	2.88	2.833	0.05	0.002209
5	2.56	2.833	-0.27	0.074529
6	2.64	2.833	-0.19	0.037249
7	3.01	2.833	0.18	0.031329
8	2.76	2.833	-0.07	0.005329
9	3.22	2.833	0.39	0.149769
10	3.83	2.833	1.00	0.994009
11	1.89	2.833	-0.94	0.889249
12	2.00	2.833	-0.83	0.693889
13	2.30	2.833	-0.53	0.284089
14	2.46	2.833	-0.37	0.139129
15	2.94	2.833	0.11	0.011449
16	3.42	2.833	0.59	0.344569
17	2.92	2.833	0.09	0.007569
18	2.73	2.833	-0.10	0.010609
19	2.89	2.833	0.06	0.003249
20	1.97	2.833	-0.86	0.744769
21	2.96	2.833	0.13	0.016129
22	3.15	2.833	0.32	0.100489
23	2.77	2.833	-0.06	0.003969
24	3.33	2.833	0.50	0.247009
25	3.56	2.833	0.73	0.528529
26	2.42	2.833	-0.41	0.170569
27	2.58	2.833	-0.25	0.064009
28	3.76	2.833	0.93	0.859329
29	3.24	2.833	0.41	0.165649
30	3.89	2.833	1.06	1.117249

**Tabel 20. Pengujian Waktu *delay* Raspberry Pi**

No	delayed time raspberry pi	nilai rata-rata	time - rata-rata	kuadrat perbedaan
1	7.08	7.02	0.06	0.0036
2	6.75	7.02	-0.27	0.0729
3	6.82	7.02	-0.20	0.04
4	7.12	7.02	0.10	0.01
5	7.28	7.02	0.26	0.0676
6	6.72	7.02	-0.30	0.09
7	7.34	7.02	0.32	0.1024
8	6.57	7.02	-0.45	0.2025
9	5.78	7.02	-1.24	1.5376
10	8.12	7.02	1.10	1.21
11	6.93	7.02	-0.09	0.0081
12	8.37	7.02	1.35	1.8225
13	6.15	7.02	-0.87	0.7569
14	7.24	7.02	0.22	0.0484
15	7.73	7.02	0.71	0.5041
16	5.81	7.02	-1.21	1.4641
17	6.75	7.02	-0.27	0.0729
18	5.82	7.02	-1.20	1.44
19	7.13	7.02	0.11	0.0121
20	5.65	7.02	-1.37	1.8769
21	8.43	7.02	1.41	1.9881
22	7.96	7.02	0.94	0.8836
23	8.15	7.02	1.13	1.2769
24	6.81	7.02	-0.21	0.0441
25	7.38	7.02	0.36	0.1296
26	5.43	7.02	-1.59	2.5281
27	7.64	7.02	0.62	0.3844
28	7.78	7.02	0.76	0.5776
29	6.54	7.02	-0.48	0.2304
30	7.32	7.02	0.30	0.09

**Tabel 21. Pengujian Waktu *delay* antara laptop dan Raspberry Pi**

No	perbandingan delayed time antara laptop dan raspberry pi	nilai rata-rata	time - rata-rata	kuadrat perbedaan
1	2.39	4.19	-1.80	3.24
2	2.56	4.19	-1.63	2.6569
3	3.01	4.19	-1.18	1.3924
4	3.09	4.19	-1.10	1.21
5	3.3	4.19	-0.89	0.7921
6	3.43	4.19	-0.76	0.5776
7	3.48	4.19	-0.71	0.5041
8	3.68	4.19	-0.51	0.2601
9	3.81	4.19	-0.38	0.1444
10	3.82	4.19	-0.37	0.1369
11	3.83	4.19	-0.36	0.1296
12	3.85	4.19	-0.34	0.1156
13	4.02	4.19	-0.17	0.0289
14	4.08	4.19	-0.11	0.0121
15	4.24	4.19	0.05	0.0025
16	4.24	4.19	0.05	0.0025
17	4.29	4.19	0.10	0.01
18	4.33	4.19	0.14	0.0196
19	4.4	4.19	0.21	0.0441
20	4.59	4.19	0.40	0.16
21	4.72	4.19	0.53	0.2809
22	4.75	4.19	0.56	0.3136
23	4.78	4.19	0.59	0.3481
24	4.79	4.19	0.60	0.36
25	4.81	4.19	0.62	0.3844
26	5.04	4.19	0.85	0.7225
27	5.06	4.19	0.87	0.7569
28	5.38	4.19	1.19	1.4161
29	5.47	4.19	1.28	1.6384
30	6.37	4.19	2.18	4.7524

Setelah mendapatkan kuadrat perbedaan dari tiap data yang dimiliki, masing-masing kuadrat perbedaan di jumlahkan dan total penjumlahan dari kuadrat perbedaan dibagi 30 sesuai dengan data hasil pengujian yang telah dimiliki, maka didapatkan hasil perhitungannya sebagai berikut:

- a) Laptop: 0.292561
  - b) Raspberry Pi: 0.64918
- Perbandingan kedua devices: 0.74709

Dikarenakan rumus standa deviasi adalah akar pangkat dua, maka standar deviasi dari ketiga data *di atas* adalah:

- a) Standar deviasi dari data *delayed time* laptop

$$\sigma = \sqrt{0.292561}$$

$$\sigma = 0.54$$

- b) Standar deviasi dari data *delayed time* Raspberry Pi

$$\sigma = \sqrt{0.64918}$$

$$\sigma = 0.8$$

Standar deviasi dari data perbandingan *delayed time* antara laptop dan Raspberry Pi

$$\sigma = \sqrt{0.74709}$$
$$\sigma = 0.86$$

Berdasarkan hasil analisis data mengenai waktu *delay* yang dibutuhkan oleh kedua perangkat untuk mendeteksi benda, dapat disimpulkan sebagai berikut:

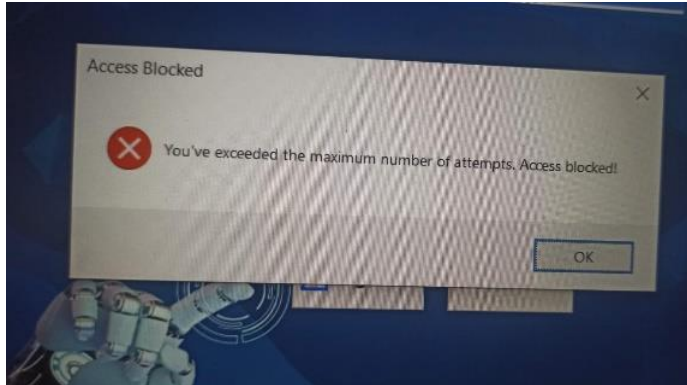
- Rata-rata waktu *delay* pada laptop sebesar 2.833 detik, sedangkan pada Raspberry Pi *delay* waktu yang dimiliki sebesar 7.02 detik. Dari hasil ini, terlihat bahwa waktu *delay* Raspberry Pi lebih tinggi dibandingkan dengan laptop, hal ini memperlihatkan respon yang dimiliki laptop lebih tinggi dibandingkan dengan Raspberry Pi.
- Mengenai nilai tengah dari data yang telah diuji, hasil ini menunjukkan bahwa separuh dari pengujian yang dilakukan pada laptop memiliki waktu *delay* kurang dari 2.825 detik, sementara separuh lainnya melebihi 2.825 detik, sementara pada Raspberry Pi, separuh pengujian memiliki waktu *delay* kurang dari 7.1 detik, separuh lainnya melebihi dari 7.1 detik. Hal ini menunjukkan bahwa konsistensi hasil pengujian dimana waktu dari Raspberry Pi tetap lebih tinggi dibandingkan dengan laptop.
- Standar deviasi dari data yang telah diuji mengukur seberapa jauh data tersebar dari nilai rata-rata. Pada laptop memiliki standar deviasi yang lebih kecil yaitu 0.54 dibandingkan dengan Raspberry Pi yaitu 0.8. hal ini menunjukkan bahwa laptop lebih konsisten dan memiliki variasi yang lebih kecil, sedangkan pada Raspberry Pi menunjukkan bahwa waktu *delay* pada Raspberry Pi lebih bervariasi.

Kesimpulan dari hasil pengujian ini adalah dari segi efisiensi laptop lebih efisien mendeteksi warna dikarenakan *delay* waktu yang lebih rendah dibandingkan dengan Raspberry Pi. Untuk keperluan pendeteksian warna penggunaan laptop lebih disarankan, namun apabila tetap membutuhkan Raspberry Pi, diperlukan optimasi atau pengaturan tambahan untuk mengurangi waktu *delay*, atau bisa mengganti kamera dengan alat pendeteksi warna lainnya.

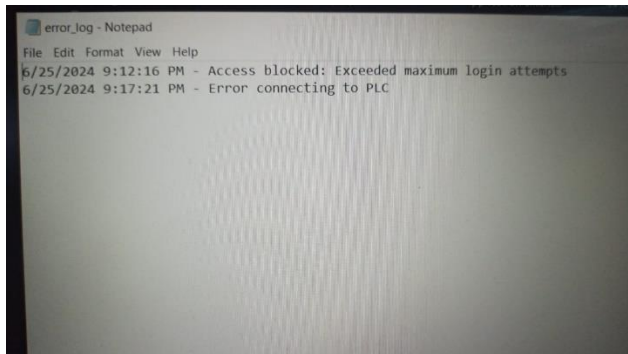
#### **4.4. Data pengujian yang diolah oleh HMI**

Pengujian ini dilakukan untuk melihat apakah data yang akan diolah oleh HMI adalah akurat, konsisten, dan dapat diandalkan oleh pengguna. Proses pengujian ini melibatkan beberapa langkah untuk memverifikasi dan memvalidasi data. Berikut hasil dari pengujian data yang diolah oleh HMI:





**Gambar 45. Pengujian pop up akses terblokir**



**Gambar 46. Pengujian *history alarm***

Gambar 44, 45, dan 46 menampilkan rekaman ketika terjadi sebuah kesalahan pada mesin. Kesalahan yang terjadi pada mesin akan direkam dan dicatat dalam sebuah *file* Notepad. Keterangan yang ditampilkan dalam *file* tersebut adalah waktu terjadinya kesalahan dan deskripsi kesalahan yang terjadi. Hal yang diuji pada pengujian ini adalah memastikan bahwa waktu yang dicatat dalam Notepad sesuai dengan waktu terjadinya kesalahan pada mesin.



**Gambar 47. Pengujian counter**

Pada gambar 47 terlihat data yang ditampilkan oleh HMI, yaitu jumlah benda yang terdeteksi dan tidak terdeteksi. Pengujian ini dilakukan untuk memastikan data yang ditampilkan aktual dan sesuai. Pada time camera ditampilkan berapa lama waktu yang dibutuhkan untuk kamera mendeteksi warna.

**Tabel 22. Pengujian perekaman *alarm* pada mesin**

No	Date	Time	Description	Status
1	7/19/2024	7:10:03	transfer error	succes
2	7/19/2024	7:10:15	vacuum failure	succes
3	7/19/2024	7:10:28	product not detected	succes
4	7/19/2024	7:10:41	vacuum failure	succes
5	7/19/2024	7:10:54	transfer error	succes
6	7/19/2024	7:11:06	transfer error	succes
7	7/19/2024	7:11:19	vacuum failure	succes
8	7/19/2024	7:11:32	product not detected	succes
9	7/19/2024	7:11:44	transfer error	succes
10	7/19/2024	7:11:57	product not detected	succes
11	7/19/2024	7:12:10	vacuum failure	succes
12	7/19/2024	7:12:22	vacuum failure	succes
13	7/19/2024	7:12:35	product not detected	succes
14	7/19/2024	7:12:48	transfer error	succes
15	7/19/2024	7:13:01	product not detected	succes
16	7/19/2024	7:13:13	vacuum failure	succes
17	7/19/2024	7:13:26	transfer error	succes
18	7/19/2024	7:13:39	product not detected	succes
19	7/19/2024	7:13:51	product not detected	succes
20	7/19/2024	7:14:04	vacuum failure	succes
21	7/19/2024	7:14:17	transfer error	succes
22	7/19/2024	7:14:30	vacuum failure	succes
23	7/19/2024	7:14:42	transfer error	succes
24	7/19/2024	7:14:55	product not detected	succes
25	7/19/2024	7:15:08	product not detected	succes
26	7/19/2024	7:15:20	vacuum failure	succes
27	7/19/2024	7:15:33	transfer error	succes
28	7/19/2024	7:15:46	vacuum failure	succes
29	7/19/2024	7:15:59	product not detected	succes
30	7/19/2024	7:16:11	transfer error	succes
31	7/19/2024	7:16:24	transfer error	succes
32	7/19/2024	7:16:37	product not detected	succes
33	7/19/2024	7:16:49	vacuum failure	succes
34	7/19/2024	7:17:02	vacuum failure	succes
35	7/19/2024	7:17:15	transfer error	succes
36	7/19/2024	7:17:27	product not detected	succes
37	7/19/2024	7:17:40	transfer error	succes
38	7/19/2024	7:17:53	product not detected	succes
39	7/19/2024	7:18:06	vacuum failure	succes
40	7/19/2024	7:18:18	transfer error	succes

Tabel 22 merupakan pengujian yang dilakukan untuk mengetahui apakah jika terjadi kesalahan pada mesin data dapat terekam secara *real time* atau tidak. Pengujian dilakukan dengan cara sengaja melakukan kesalahan untuk melihat apakah HMI dapat merekam dan mengolah data dari mesin, setelah dilakukan 40 kali pengujian, tingkat keberhasilan dari pengujian yang telah dilakukan adalah 100%. Dengan tingkat keberhasilan sebesar 100%, mengindikasikan bahwa HMI

dapat merekam dan mengolah data mengenai kesalahan yang terjadi pada mesin dengan sangat baik. Jika menggunakan perhitungan *confidence interval* (CI) yang telah dilakukan pada pengujian 1 untuk proporsi binomial pada tingkat kepercayaan 95%, hasil dari perhitungan CI tidak berpengaruh terhadap hasil pengujian pengolahan data pada HMI. Mengingat hasil pengujian yang telah dilakukan berhasil, dapat diyakini bahwa tingkat keberhasilan HMI dalam mengolah data dari mesin adalah sangat tinggi.

# Bab 5. Kesimpulan dan Saran

## 5.1. Kesimpulan

Berdasarkan hasil perancangan dan setelah dilakukan pengujian, kesimpulan yang dapat diambil dari tugas akhir yang berjudul *human machine interface* menggunakan Raspberry Pi pada mesin deteksi warna menggunakan metode ruang warna YCbCr berdasarkan tujuan dan manfaatnya adalah sebagai berikut:

1. Raspberry Pi dapat dijadikan sebagai pusat kendali atau HMI (*Human Machine Interface*) pada mesin otomasi. Hasil pengujian komunikasi HMI & PLC menunjukkan bahwa pertukaran data dari kedua perangkat bekerja dengan baik dan tidak ada kendala. Hal ini memperkuat alasan bahwa Raspberry Pi dapat digunakan sebagai HMI.
2. Analisis terhadap metode ruang warna YCbCr dapat digunakan untuk melacak objek sesuai dengan warna yang diinginkan. Hal ini terbukti dari hasil pengujian kamera terhadap benda, di mana percobaan dengan berbagai metode seperti pengaturan pencahayaan dan perubahan warna latar belakang menghasilkan data yang konsisten. Ini menunjukkan bahwa metode ruang warna YCbCr efektif untuk melacak objek pada mesin otomasi. Namun, terdapat kendala pada waktu tunda (*delay*) yang dibutuhkan kamera saat mendeteksi benda. Saat HMI dijalankan pada Raspberry Pi, waktu tunda kamera sekitar 4,19 detik lebih lama dibandingkan dengan menggunakan laptop. Hal ini dikarenakan Windows digunakan sebagai sistem operasi pada Raspberry Pi karena Raspbian, sistem operasi asli Raspberry Pi, tidak memiliki *software* untuk *webcam* yang digunakan sebagai alat pendeteksi benda. Namun, penggunaan Windows menyebabkan penggunaan memori mencapai batas maksimum, sehingga respon komputer menjadi lambat dan mempengaruhi efisiensi Raspberry Pi sebagai HMI. Apabila HMI yang dirancang tidak memerlukan kamera, Raspbian dapat digunakan sebagai sistem operasi untuk mengurangi penggunaan memori dibandingkan dengan Windows.
3. Raspberry Pi dapat terhubung dengan sistem pengendali seperti PLC pada mesin otomasi. Hal ini disimpulkan dari hasil pengujian komunikasi HMI dan PLC, di mana tidak terjadi hambatan dalam pertukaran data antara kedua perangkat, baik saat Raspberry Pi membaca maupun menulis data.
4. HMI yang dibuat dapat mengendalikan, mengontrol, dan mengolah data. Analisis terhadap data hasil pengujian menunjukkan bahwa HMI mencatat waktu kesalahan mesin dengan tepat, serta mampu mendeteksi jumlah produk sesuai dengan parameter yang diatur. Hal ini membuktikan bahwa HMI berfungsi sesuai dengan tujuan penelitian, yaitu untuk mengelola dan memproses data dengan akurat dalam konteks pengendalian mesin.

## 5.2. Saran

Berdasarkan hasil kesimpulan yang didapatkan setelah melakukan perancangan dan pengujian, ada beberapa hal yang perlu ditingkatkan dan ditindak lanjuti, berikut beberapa saran yang dapat dilakukan untuk penelitian lebih lanjut:

1. Pada saat melakukan perancangan, perhatikan apakah perangkat keras yang digunakan dapat berfungsi dengan baik pada Raspberry Pi tanpa memengaruhi segi fungsionalitas.
2. Untuk peneliti selanjutnya, usahakan meminimalisir waktu yang dibutuhkan kamera untuk mendeteksi benda agar tidak memerlukan waktu yang lama.
3. Untuk peneliti selanjutnya, lebih baik memaksimalkan penggunaan sistem operasi (OS) yang sudah terbukti berfungsi dengan baik pada Raspberry Pi, seperti Raspbian dan Ubuntu. Sistem operasi yang digunakan dalam penelitian ini adalah Windows yang masih dalam tahap *tester mode* dan tidak kompatibel dengan spesifikasi Raspberry Pi sehingga respon komputer menjadi lambat dan memengaruhi efisiensi.
4. Meningkatkan keamanan pada HMI dengan memberikan batasan akses tiap pengguna, menambahkan fitur pembuatan akun baru, dan memungkinkan penggantian sandi pada HMI tanpa perlu mengubah penulisan kode.
5. Saat merancang penggunaan alat kontrol, pastikan perangkat yang digunakan dapat mendukung fitur perintah *Coil* atau *boolean* sehingga tidak membutuhkan banyak memori *word* untuk membuat perintah.
6. Untuk peneliti selanjutnya, usahakan memiliki lebih banyak parameter dalam pengujian saat akan mendeteksi warna pada objek agar mendapatkan nilai akurat dari warna yang akan dideteksi oleh kamera, seperti penggunaan cahaya terang dan redup, serta penggunaan cahaya dengan warna yang berbeda.

## Daftar Pustaka

- [1] D. A. A. M. Dedy Agung Prabowo, "DETEKSI DAN PERHITUNGAN OBJEK BERDASARKAN WARNA MENGGUNAKAN COLOR OBJECT TRACKING," <https://ejournal.unib.ac.id>.
- [2] J. Y. S. I. P. N. Moh. La Andi Rais Imran Yatim, "Deteksi Area Wajah Manusia Pada Citra Berwarna Berbasis Segmentasi Warna YCbCr dan Operasi Morfologi Citra," <https://www.academia.edu>.
- [3] B. N. B. Alhasnawi, "SCADA controlled smart home using Raspberry Pi3," <https://www.researchgate.net/publication>. Accessed: Jun. 30, 2023. [Online]. Available: [https://www.researchgate.net/publication/327348520\\_SCADA\\_controlled\\_smart\\_home\\_using\\_Raspberry\\_Pi3](https://www.researchgate.net/publication/327348520_SCADA_controlled_smart_home_using_Raspberry_Pi3)
- [4] S. S. R. Prof. Ravi Kiran R, "Human Machine Interface ," <https://ijrpr.com>. Accessed: Jun. 30, 2023. [Online]. Available: <https://ijrpr.com/uploads/V3ISSUE7/IJRPR6098.pdf>
- [5] D. Hasbi Simanjuntak, "Sistem Monitoring Pada Sorting Machine dengan HMI Berbasis PLC," 2021.
- [6] M. R. Faisal and E. Kurniawan, "Seri Belajar Windows Forms: Membangun Aplikasi Desktop berbasis .NET Core 3.1 dengan Visual Studio 2019." [Online]. Available: <https://www.researchgate.net/publication/340133969>
- [7] N. Y. Putri and R. Mukhaiyar, "Control and Monitoring System Process Handling Production on SMI 4.0 Machines using PLC Controller Wirelessly Based on Human Machine Interface," *JTEV (Jurnal Teknik Elektro dan Vokasional)*, vol. 8, no. 1, p. 158, May 2022, doi: 10.24036/jtev.v8i1.116918.
- [8] H. Rahadian and M. A. Heryanto, "Pengembangan Human Machine Interface (HMI) pada Simulator Sortir Bola sebagai Media Pembelajaran Otomasi Industri," *JURNAL NASIONAL TEKNIK ELEKTRO*, vol. 9, no. 2, p. 84, Jul. 2020, doi: 10.25077/jnte.v9n2.766.2020.
- [9] T. Ade Mulyanto, M. Habiby, and R. Adam, "HOME AUTOMATION SYSTEM DENGAN MENGGUNAKAN RASPBERRY PI 4," 2021.
- [10] D. Yuhendri, "Penggunaan PLC Sebagai Pengontrol Peralatan Building Automatis," 2018.
- [11] Sadi Sumardi, "3.IMPLEMENTASI HUMAN MACHINE INTERFACE PADA MESIN HEEL," 2020.
- [12] Yesputra Rolly, "5. VISUAL STUDIO 2010," 2017.
- [13] Gopinathan Ssubramani, "A Study on Image Enhancement Techniques using YCbCr Color Space Methods." Accessed: Jul. 03, 2023. [Online]. Available:

[https://www.researchgate.net/publication/307527154\\_A\\_Study\\_on\\_Image\\_Enhancement\\_Techniques\\_using\\_YCbCr\\_Color\\_Space\\_Methods](https://www.researchgate.net/publication/307527154_A_Study_on_Image_Enhancement_Techniques_using_YCbCr_Color_Space_Methods)

- [14] Indrajani, *DATABASE DESIGN ALL IN ONE*. Elex Media Komputindo, 2018.
- [15] N. L. Kamila, "6 Tahapan Desain dalam Menciptakan UI yang Ramah Pengguna." Accessed: Jul. 19, 2024. [Online]. Available: <https://dibimbing.id/blog/detail/tahapan-desain-dalam-menciptakan-ui-yang-ramah-pengguna>

## Biodata



Nama : Arizul Pajri  
TTL : Payakumbuh,13 Juli 1997  
Agama : Islam  
Alamat : Nadim Raya Tahap 1 Blok B no 29

Email : arizulfajri@gmail.com  
Riwayat Pendidikan : SMA/SMK: SMK Negeri 1 Bukittinggi  
SMP : SMP Negeri 1 Tilatang Kamang



Nama : Naufal Hanif Marzukie  
TTL : Bandung,29 April 2000  
Agama : Islam  
Alamat : Gardan raya blok GA 5 no 7

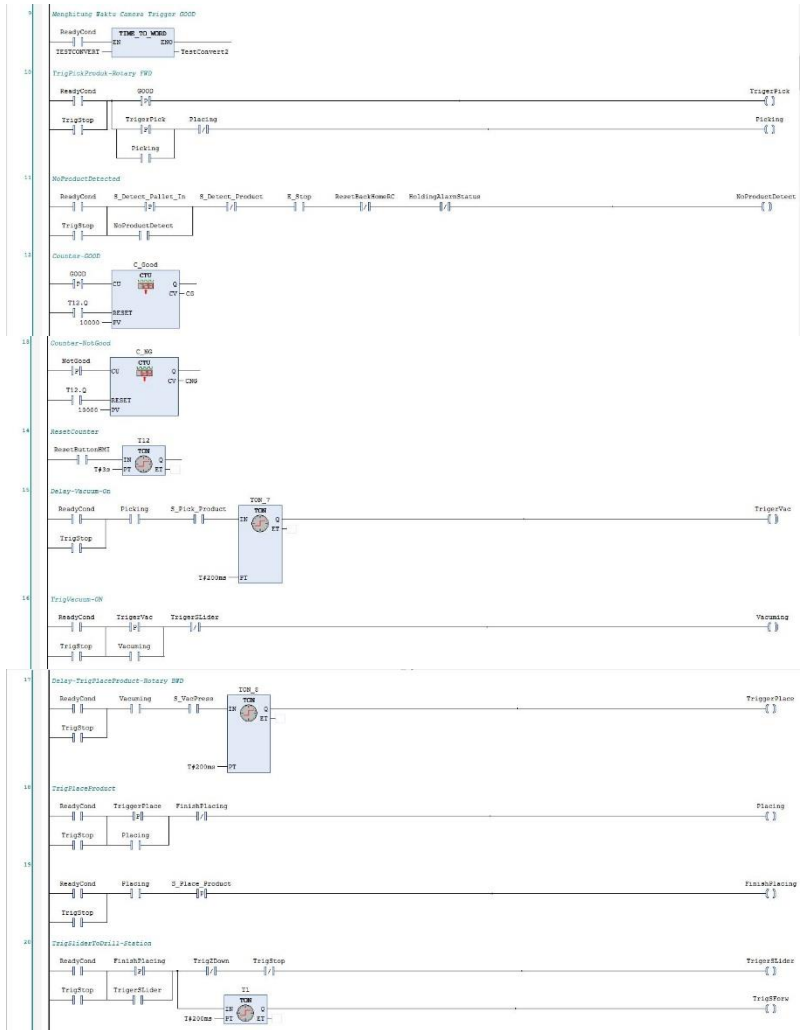
Email : Naufalhanif294@gmail.com  
Riwayat Pendidikan : SMA/SMK: SMK Negeri 1 Batam  
SMP : SMP Negeri 28 Batam

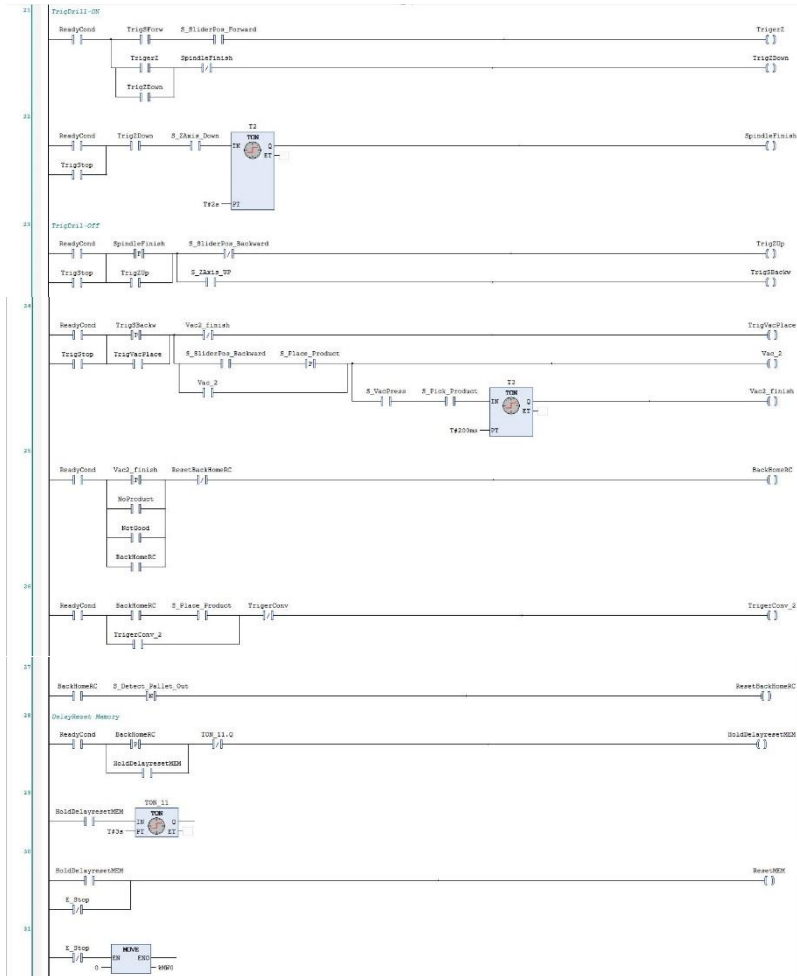


Nama : Rifdah Nur Fauziah  
TTL : Batam,11 Desember 2000  
Agama : Islam  
Alamat : Perum. Citra Pendawa Asri Blok E4 No 10

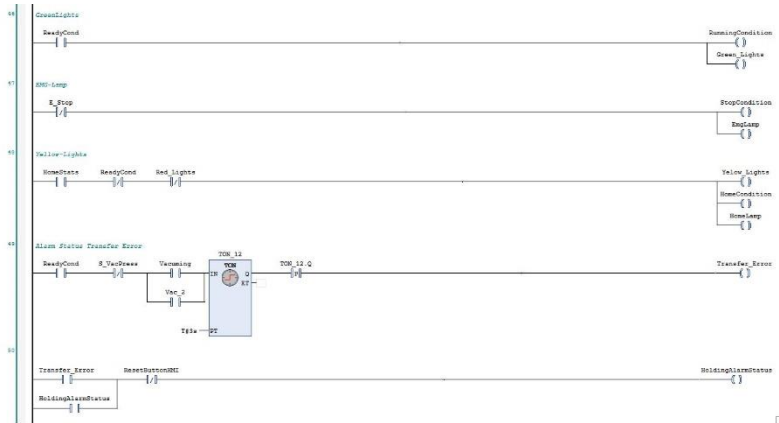
Email : rifdaahnurfauziah@gmail.com  
Riwayat Pendidikan : SMA/SMK: SMK Negeri 1 Batam  
SMP : SMP Negeri 11 Batam











## Text Program PLC:

```
1 PROGRAM Run_Auto
2 VAR
3     S_Stoper_A AT %IX14.0: BOOL;
4     S_Stoper_B AT %IX14.1: BOOL;
5     S_Detect_Pallet_In AT %IX14.2: BOOL;
6     S_Detect_Pallet_Out AT %IX14.3: BOOL;
7     S_Detect_Product AT %IX14.4: BOOL;
8     S_Pick_Product AT %IX14.5: BOOL;
9     S_Place_Product AT %IX14.6: BOOL;
10    S_VacPress AT %IX14.7: BOOL;
11    S_SliderPos_Forward AT %IX15.0: BOOL;
12    S_SliderPos_Backward AT %IX15.1: BOOL;
13    S_ZAxis_UP AT %IX15.2: BOOL;
14    S_ZAxis_Down AT %IX15.3: BOOL;
15    Remote_Local AT %IX15.4: BOOL;
16    Start AT %IX15.5: BOOL;
17    Stop AT %IX15.6: BOOL;
18    Reset AT %IX15.7: BOOL;
19    Home AT %IX16.0: BOOL;
20    E_Stop AT %IX16.1: BOOL;
21    ReadyCond: BOOL;
22    Conveyor_M_Atas AT %QX4.0: BOOL;
23    SV_Stoper_A AT %QX4.2: BOOL;
24    SV_Stoper_B AT %QX4.3: BOOL;
25    SV_Rotary_Pick AT %QX4.4: BOOL;
26    SV_Rotary_Place AT %QX4.5: BOOL;
27    SV_Vacuum_Press AT %QX4.6: BOOL;
28    SV_Slider_Forward AT %QX4.7: BOOL;
29    SV_Slider_Backward AT %QX5.0: BOOL;
30    SV_ZAxis_Up AT %QX5.1: BOOL;
31    SV_ZAxis_Down AT %QX5.2: BOOL;
32    SV_Motor_Spindle AT %QX5.3: BOOL;
33    Red_Lights AT %QX5.4: BOOL;
34    Yellow_Lights AT %QX5.5: BOOL;
35    Green_Lights AT %QX5.6: BOOL;
36    StartLamp AT %QX6.1: BOOL;
37    StopLamp AT %QX6.2: BOOL;
38    ResetLamp AT %QX6.3: BOOL;
39    HomeLamp AT %QX6.4: BOOL;
40    Buzzer AT %QX6.5: BOOL;
41    EmgLamp AT %QX6.6: BOOL;
42
```

```
43 TrigerConv: BOOL;
44 TrigerPick: BOOL;
45 Picking: BOOL;
46 TrigerVac: BOOL;
47 Vacuuming: BOOL;
48 TriggerPlace: BOOL;
49 Placing: BOOL;
50 FinishPlacing: BOOL;
51 TrigerSLider: BOOL;
52 T1: TON;
53 TrigSForw: BOOL;
54 TrigerZ: BOOL;
55 TrigZDown: BOOL;
56 T2: TON;
57 SpindleFinish: BOOL;
58 TrigZUp: BOOL;
59 HomeStats: BOOL;
60 TrigStop: BOOL;
61 HomeSearch: BOOL;
62 TrigSBackw: BOOL;
63 TrigVacPlace: BOOL;
64 Vac_2: BOOL;
65 Vac2_finish: BOOL;
66 T3: TON;
67 BackHomeRC: BOOL;
68 T4: TON;
69 TrigerConv_2: BOOL;
70 FinishCycle: BOOL;
71 T5: TON;
72 TrigBuzzer: BOOL;
73
74 StopConv: BOOL;
75 T6: TON;
76 NoProductDetect: BOOL;
77 NoProduct: BOOL;
78 AlarmNoProduct: BOOL;
79 T7: TON;
80
```

```

81 a AT %MW0: WORD;
82 b AT %MW1: WORD;
83 c AT %MW2: WORD;
84 d AT %MW3: WORD;
85 e AT %MW4: WORD;
86 f AT %MW5: WORD;
87 g AT %MW6: WORD;
88 h AT %MW7: WORD;
89 i AT %MW8: WORD;
90 j AT %MW9: WORD;
91 k AT %MW10: WORD;
92 CG AT %MW11: WORD;
93 CNG AT %MW12: WORD;
94 TestConvert2 AT %MW14: WORD;
95
96 StartHMI AT %MX0.0: BOOL; // MW0
97 StopHMI AT %MX0.1: BOOL; // MW0
98
99 TrigCamera AT %MX2.0: BOOL; // MW1
100 GOOD AT %MX4.0: BOOL; // MW2
101 NotGood AT %MX4.1: BOOL; // MW2
102
103 ResetMEM AT %MX8.0: BOOL; // MW3
104
105 StopCondition AT %MX10.0: BOOL; // MW5
106 HomeCondition AT %MX10.1: BOOL; // MW5
107 RunningCondition AT %MX10.2: BOOL; // MW5
108
109 Transfer_Error AT %MX14.0: BOOL; // MW7
110 Vacuum_Failure AT %MX14.1: BOOL; // MW7
111 ProductNot_Detected AT %MX14.2: BOOL; // MW7
112
113 ResetButtonHMI AT %MX16.0: BOOL; // MW8
114
115 HoldingAlarmStatus AT %MX20.0: BOOL; //MW10
116

```

```

117 // DelayRotaryFWD
118 TON_7: TON;
119 // DelayTrigPlace-Product
120 TON_8: TON;
121
122 cv_Var AT %MW5: WORD;
123 C0: CTU;
124 TON_20: TON;
125 ResetBackHomeRC: BOOL;
126 TrigCamera0: BOOL;
127 HoldDelayresetMEM: BOOL;
128 TON_11: TON;
129 HoldConv2: BOOL;
130 TON_12: TON;
131 C_Good: CTU;
132 C_NG: CTU;
133 T12: TON;
134 TESTCONVERT: TIME;
135 END_VAR

```

Program HMI:

**A. Form utama**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Web.UI.Design.WebControls;
using System.Windows.Forms;
using UPPERHMI1.UserControls;

```

```
namespace UPPERHMI1
```

```
{
```

```
    public partial class Form1: Form
    {
```

```
        private Panel panelContainer;
        private bool isLoggedIn = FALSE;
    }

```

```

private void ShowAutorun()
{
    Autorun uc = new Autorun();
    addUserControl(uc);
}
private void uc_LoginSuccessful(object
sender, EventArgs e)
{
    SetLoginStatus(true);
    ShowAutorun();
}

public Form()
{
    InitializeComponent();
    panelContainer = new Panel();
    panelContainer.Dock = DockStyle.Fill;
    this.Controls.Add(panelContainer);
    ShowLoginForm();
}
private void ShowLoginForm()
{
    Login uc = new Login();
    uc.LoginSuccessful += uc_LoginSuccessful;
    addUserControl(uc);
}

private void addUserControl(UserControl
userControl)
{
    userControl.Dock = DockStyle.Fill;
    userControl.Anchor = AnchorStyles.Bottom
| AnchorStyles.Left;
    panelContainer.Controls.Add(userControl);
    userControl.BringToFront();
    userControl.Location = new Point(0,
panelContainer.Height - userControl.Height);
}

```

```

sender, private void guna2Button1_Click(object
EventArgs e)
{
    Login uc = new Login();
    addUserControl(uc);
    ShowLoginForm();
}

private void BtnClose_Click(object
sender, EventArgs e)
{
    this.Close();
}

private void BtnMax_Click(object sender,
EventArgs e)
{
    if (this.WindowState ==
FormWindowState.Maximized)
    {
        this.WindowState =
FormWindowState.Normal; // Jika sudah maksimalkan,
kembalikan ke normal
    }
    else
        this.WindowState =
FormWindowState.Maximized;
}

private void BtnMin_Click(object sender,
EventArgs e)
{
    this.WindowState =
FormWindowState.Minimized;
}

sender, private void guna2Button3_Click(object
EventArgs e)
{
    // Memeriksa apakah pengguna telah login
    if (isLoggedIn)
    {

```

```

        Autorun uc = new Autorun();
        addUserControl(uc);
    }
    else
    {
        MessageBox.Show("Anda harus login
terlebih dahulu.");
        ShowLoginForm(); // Menampilkan form
Login jika pengguna belum login
    }
}
public void SetLoginStatus(bool status)
{
    isLoggedIn = status;
}

private void Form1_Load(object sender,
EventArgs e)
{
}

private void guna2Button2_Click(object
sender, EventArgs e)
{
    // Memeriksa apakah pengguna telah login
    if (isLoggedIn)
    {
        Manual_IO uc = new Manual_IO();
        addUserControl(uc);
    }
    else
    {
        MessageBox.Show("Anda harus login
terlebih dahulu.");
        ShowLoginForm(); // Menampilkan form
Login jika pengguna belum login
    }
}
}
}
}

```

```

    B. Form login
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace UPPERHMI1.UserControls
{
    public partial class Login: UserControl
    {
        private int attemptsLeft = 3; // Declare
attemptsLeft here
        public event EventHandler LoginSuccessful;

        public Login()
        {
            InitializeComponent();
            this.Height = 463; // Set the desired
height
            this.Width = 910; // Set the desired
width
            TxtUserName.Focus();
            TxtPassword.UseSystemPasswordChar = true;
            this.TxtPassword.KeyPress += new
System.Windows.Forms.KeyPressEventHandler(this.TxtPas
sword_KeyPress);
        }

        private void BtnLogin_Click(object sender,
EventArgs e)
        {
            if (TxtUserName.Text == "" &&
TxtPassword.Text == "")
            {

```

```

        MessageBox.Show("Username and
        Password cannot be empty!", "Fill Username and
        Password", MessageBoxButtons.OK,
        MessageBoxIcon.Exclamation);
    }
    else if (TxtUserName.Text == "1" &&
    TxtPassword.Text == "1")
    {
        MessageBox.Show("Login Success!",
        "Access Success", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        Manual_IO startMenu = new
        Manual_IO();
        startMenu.Show();
        LoginSuccessful?.Invoke(this,
        EventArgs.Empty);
        this.Hide();
    }
    else
    {
        attemptsLeft--;
        MessageBox.Show("Incorrect username
        and password", "Check Username and Password",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        if (attemptsLeft <= 0)
        {
            MessageBox.Show("You've exceeded
            the maximum number of attempts. Access blocked!",
            "Access Blocked", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
            WriteMessageToNotepad("Access
            blocked: Exceeded maximum login attempts");
            Application.Exit(); // Menutup
            aplikasi setelah akses terblokir
        }
        else
        {
            Clean();
            TxtUserName.Focus();
        }
    }
}

```

```

    }
    private void Clean()
    {
        TxtUserName.Text = "";
        TxtPassword.Text = "";
    }

    private void checkBox1_CheckedChanged(object
sender, EventArgs e)
    {
        if (checkBox1.Checked)
        {
            TxtPassword.UseSystemPasswordChar =
FALSE;
        }
        else
        {
            TxtPassword.UseSystemPasswordChar =
true;
        }
        TxtPassword.Focus();
    }

    private void WriteMessageToNotepad(string
message)
    {
        try
        {
            using (StreamWriter sw = new
StreamWriter("error_log.txt", true))
            {
                sw.WriteLine($"{DateTime.Now} -
{message}");
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error writing to
Notepad: " + ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

```

```

        }

        private void TxtPassword_KeyPress(object
sender, KeyPressEventArgs e)
        {
            if (e.KeyChar == (char)Keys.Enter)
            {
                BtnLogin_Click(sender,
EventArgs.Empty);
            }
        }

        private void BtnCancel_Click(object sender,
EventArgs e)
        {
            Clean(); // Clear the username and
password fields
            attemptsLeft = 3; // Reset the number of
attempts
            TxtUserName.Focus(); // Set focus to the
username field
        }

        private void TxtUserName_TextChanged(object
sender, EventArgs e)
        {
        }
    }
}

```

### C. Autorun:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

using System.Windows.Forms;
using System.Collections;
using AForge;
using AForge.Imaging;
using AForge.Imaging.Filters;
using AForge.Video;
using AForge.Video.DirectShow;
using EasyModbus;
using AForge.Math.Geometry;
using System.Reflection.Emit;
using System.IO;
using System.Security.Cryptography.X509Certificates;

namespace UPPERHMI1.UserControls
{
    public partial class Autorun: UserControl
    {
        private Panel container;
        public int a; //WriteInput_PB.Start
        public int b; //Trigger on camera
        public int c; //object good
        public int d; //object not good
        public int e; //reset all memori
        public int f; //alarm
        public int g; //ReadOutputMerah
        public int h; //ReadOutputKuning
        public int i; //ReadOutputHijau
        public int j; //Connect_indicator
        public int k; //Reset Button

        ModbusClient plc = new
ModbusClient("192.168.1.5", 502); //IP Address PLC

        Bitmap sourceImage;
        Bitmap detectedImage;

        float Ymin = 0, Ymax = 1;
        float Cbmin = -0.5f, Cbmax = 0.5f;
        float Crmin = 0.5f, Crmax = 0.5f;

        int TRACK_SPACE = 2;
    }
}

```

```

        private FilterInfoCollection videoDevices;
        private VideoCaptureDevice videoDevice;
        private ArrayList listCamera = new
ArrayList();

        public Autorun()
        {
            InitializeComponent();
            panelContainer = new Panel();
            panelContainer.Dock = DockStyle.Fill;
            this.Controls.Add(panelContainer);

            //TAMBAHIN INI
            comboBoxColors.DataSource = colors;
            comboBoxColors.DisplayMember = "Name";
        }

        private static string _usbcamera;
        public string usbcamera
        {
            get { return _usbcamera; }
            set { _usbcamera = value; }
        }

        private void OpenVideoSource(IVideoSource
source)
        {
            try
            {
                this.Cursor = Cursors.WaitCursor;
                CloseCurrentVideoSource();
                videoSourcePlayer1.VideoSource =
source;

                videoSourcePlayer1.Start();
                this.Cursor = Cursors.Default;
            }
            catch { }
        }

        public void CloseCurrentVideoSource()

```

```

        {
            try
            {
                if (videoSourcePlayer1.VideoSource !=
null)
                    {
                        videoSourcePlayer1.SignalToStop();

                            for (int i = 0; i < 30; i++)
                            {
                                if
(!videoSourcePlayer1.IsRunning)
                                    break;

                                System.Threading.Thread.Sleep(100);

                                    }
                                if (videoSourcePlayer1.IsRunning)
                                {
                                    videoSourcePlayer1.Stop();
                                }
                                videoSourcePlayer1.VideoSource =
null;
                            }
                        }
                    catch { }
                }

        private void YCbCrColorFiltering(Bitmap
srcImage)
        {
            YCbCrFiltering filter = new
YCbCrFiltering();

            filter.Y = new Range(Ymin, Ymax);
            filter.Cb = new Range(Cbmin, Cbmax);
            filter.Cr = new Range(Crmin, Crmax);

            detectedImage = filter.Apply(srcImage);
            pictureBox1.Image = detectedImage;

```

```

    }

    private bool objectDetected = FALSE;
    private Control panelContainer;

    private void ObjectTracking(Bitmap srcImage)
    {
        if (srcImage == null || detectedImage ==
null) return;

        Bitmap newImage =
(Bitmap)detectedImage.Clone();

        BlobCounter bc = new BlobCounter();
        bc.MinHeight = 20;
        bc.MinWidth = 20;
        bc.FilterBlobs = true;
        bc.ObjectsOrder = ObjectsOrder.Area;
        bc.ProcessImage(newImage);
        Rectangle[] rects =
bc.GetObjectsRectangles();

        objectDetected = rects.Length > 0;

        foreach (Rectangle recs in rects)
        {
            if (rects.Length > 0)
            {
                Rectangle objectRect = rects[0];
                Graphics graph =
Graphics.FromImage(srcImage);
                using (Pen pen = new
Pen(Color.FromArgb(255, 0, 0), 10))
                {
                    graph.DrawRectangle(pen,
objectRect);
                }
                graph.Dispose();
            }
        }
    }

```

```

    }

    pictureBox2.Image = srcImage;

    label4.BeginInvoke((MethodInvoker)(() =>
    {
        label4.Text = objectDetected ?
"Detected": "FALSE";
    }));
}

private void button1_Click(object sender,
EventArgs e)
{
    OpenCamera();
    ColorData selectedColor =
(ColorData)comboBoxColors.SelectedItem;

    // Gunakan nilai-nilai warna yang dipilih
    float Ymin = selectedColor.Ymin;
    float Ymax = selectedColor.Ymax;
    float Cbmin = selectedColor.Cbmin;
    float Cbmax = selectedColor.Cbmax;
    float Crmin = selectedColor.Crmin;
    float Crmax = selectedColor.Crmax;
}

private void button2_Click(object sender,
EventArgs e)
{
    if ((videoDevice != null))
    {
        try
        {

((VideoCaptureDevice)videoDevice).DisplayPropertyPage
(this.Handle);

        }
        catch (NotSupportedException ex)
        {

```

```

        MessageBox.Show(ex.Message,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void Resetallmemori()
{
    var value = plc.ReadHoldingRegisters(0,
5);

    if (value[4] == 1)
    {

        ResetTrackedImages();
        ResetLabel4();
        int c = 0;
        int write = c;
        plc.WriteSingleRegister(2, write);

    }

    else
    {

    }

    int[] read = plc.ReadHoldingRegisters(0,
5);
    e = read[4];
}

private void Cameratriger()
{
    try
    {
        var value =
plc.ReadHoldingRegisters(0, 2);

        if (value[1] == 1)
        {

```

```

        OpenCamera();
        CheckLabel();
    }

    else
    {
        CloseCamera();
        ResetDetectedImages();
    }

    int[] read =
plc.ReadHoldingRegisters(0, 2);
        b = read[1];
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error communicating
with PLC: " + ex.Message);
        WriteMessageToNotepad("Error
communicating with PLC: " + ex.Message);
        Application.Exit(); // Menutup
aplikasi karena kesalahan yang fatal
    }
}

private void Autorun_FormClosing(object
sender, FormClosingEventArgs e)
{
    // Pastikan kamera ditutup sebelum
menutup aplikasi
    CloseCamera();
    // Hentikan timer jika masih berjalan
}

private void button3_Click(object sender,
EventArgs e)
{
    CloseCamera();
    ResetDetectedImages();
    ResetLabel4();
    ResetTrackedImages();
}

```

```

    }

    private void Form1_Formclosed(object sender,
    FormClosedEventArgs e)
    {
        if (videoDevice != null &&
        videoDevice.IsRunning)
            videoDevice.Stop();
    }

    //untuk menjalankan sequence ketika objek
    telah terdeteksi
    public void CheckLabel()
    {
        if (label4.Text == "Detected")
        {
            CloseCamera();
            ResetDetectedImages();
            int c = 1;
            int write = c;
            plc.WriteSingleRegister(2, write);
        }

        else if (label4.Text == "FALSE")
        {
            CloseCamera();
            ResetDetectedImages();
            int c =2;
            int write = c;
            plc.WriteSingleRegister(2, write);
        }
    }

    private void button9_Click(object sender,
    EventArgs e)
    {
        try
        {
            if (plc.Connected)
            {
                plc.Disconnect();
            }
        }
    }

```

```

        label7.Text = "Disconnected";
        MessageBox.Show("Disconnected
from PLC");
        int j = 0;
        int write = j;
        plc.WriteSingleRegister(9,
write);
    }
    else
    {
        MessageBox.Show("Already
disconnected from PLC!");
        return;
    }
}
catch (Exception ex)
{
    MessageBox.Show("Error disconnecting
from PLC: " + ex.Message);
}
}

private void connect_indicator()
{
    var value = plc.ReadHoldingRegisters(0,
10);

    if (value[9] == 1)
    {
        REDCON.Visible = FALSE;
        GREENCON.Visible = true;
    }
    else if (value[9] == 0)
    {
        REDCON.Visible = true;
        GREENCON.Visible = FALSE;
    }
}

private void button8_Click(object sender,
EventArgs e)
{

```

```

try
{
    plc.Connect();
    if (plc.Connected == true)
    {
        label7.Text = "Connected";
        MessageBox.Show("Connected");
        WriteMessageToNotepad("Connected
to PLC");

        int j = 1;
        int write = j;
        plc.WriteSingleRegister(9,
write);
    }
    else
    {
        MessageBox.Show("Failed to
connect to PLC!");
        WriteMessageToNotepad("Failed to
connect to PLC");

        int j = 0;
        int write = j;
        plc.WriteSingleRegister(9,
write);
    }

    var readplcTimer = new Timer()
    {
        Interval = 1500,
        Enabled = true
    };

    var HistoryAlarm = new Timer()
    {
        Interval = 10,
        Enabled = true
    };

    readplcTimer.Tick +=
ReadplcTimer_Tick;
    readplcTimer.Start();
}

```

```

        HistoryAlarm.Tick +=
HistoryAlarm_Tick;
        HistoryAlarm.Start();
        Cameratrigger();
        Alarm();
        AlarmStatus();
        connect_indicator();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error connecting to
PLC: " + ex.Message);
        WriteMessageToNotepad("Error
connecting to PLC");
        // Ensure lights indicate
disconnection in case of error
    }
}

private int Delaytime;
private int goodCounter;
private int notGoodCounter;
private void Counter()
{
    int [] read = plc.ReadHoldingRegisters(0,
15);

    int goodCounter = read[11];
    int notGoodCounter = read[12];
    int Delaytime = read[14];
    textBox1.Text = goodCounter.ToString();
    textBox2.Text =
notGoodCounter.ToString();
    textBox3.Text = Delaytime.ToString();

}
private int lastAlarmState = 0;

private void Alarm()
{
    var value = plc.ReadHoldingRegisters(0,
8);

```

```

        int currentAlarmState = value[7];

        if (currentAlarmState != lastAlarmState)
        {
            switch (currentAlarmState)
            {
                case 1:
                    MessageBox.Show("Transfer
Error");
                    WriteMessageToNotepad("Transfer Error");
                    break;
                case 2:
                    MessageBox.Show("Vacuum
Failure");
                    WriteMessageToNotepad("Vacuum
Failure");
                    break;
                case 4:
                    MessageBox.Show("Product No
Detect");
                    WriteMessageToNotepad("Product No Detect");
                    break;
                default:
                    // Handle other cases or
                    default behavior if needed
                    break;
            }

            // Update the lastAlarmState to the
            current state
            lastAlarmState = currentAlarmState;
        }

        int[] read = plc.ReadHoldingRegisters(0,
8);
        f = read[7];
    }

    private void AlarmStatus ()
    {

```

```

9);

var value = plc.ReadHoldingRegisters(0,

if (value[5] == 1)
{
    RED1.Visible = true;
    RED0.Visible = FALSE;
    YELLOW1.Visible = true;
    YELLOW0.Visible = FALSE;
    GREEN1.Visible = FALSE;
    GREEN0.Visible = true;
}
else if (value[5] == 2)
{
    RED1.Visible = FALSE;
    RED0.Visible = true;
    YELLOW1.Visible = FALSE;
    YELLOW0.Visible = true;
    GREEN1.Visible = FALSE;
    GREEN0.Visible = true;
}

else if (value[5] == 4)
{
    RED1.Visible = FALSE;
    RED0.Visible = true;
    YELLOW1.Visible = true;
    YELLOW0.Visible = FALSE;
    GREEN1.Visible = true;
    GREEN0.Visible = FALSE;
}
}

private void WriteMessageToNotepad(string
message)
{
    try
    {
        using (StreamWriter sw = new
StreamWriter("error_log.txt", true))
        {

```

```

        sw.WriteLine($"{DateTime.Now} -
{message}");
    }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error writing to
Notepad: " + ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void button7_Click(object sender,
EventArgs e)
{
    int a = 0;
    int write = a;
    plc.WriteSingleRegister(0, write);

    CloseCamera();
    Resetallmemori();
}
private void timer1_Tick(object sender,
EventArgs e)
{
    Cameratriger();
}

private void ReadplcTimer_Tick(object sender,
EventArgs e)
{
    Cameratriger();
    Resetallmemori();
    connect_indicator();
    Counter();
}

private void ResetLabel4()
{
    label4.BeginInvoke((MethodInvoker)() =>
    {

```

```

        label4.Text = ". . . . .";
    }));
}

private void OpenCamera()
{
    try
    {
        usbcamera =
comboBox1.SelectedIndex.ToString();
        videoDevices = new
FilterInfoCollection(FilterCategory.VideoInputDevice)
;
        if (videoDevices.Count != 0)
        {
            foreach (FilterInfo device in
videoDevices)
            {
                listCamera.Add(device.Name);
            }
        }
        else
        {
            MessageBox.Show("Camera Devices
Found");
        }
        videoDevice = new
VideoCaptureDevice(videoDevices[Convert.ToInt32(usbca
mera)].MonikerString);
        OpenVideoSource(videoDevice);
    }
    catch (Exception err)
    {
        MessageBox.Show(err.ToString());
    }
}

private void CloseCamera()
{
    if (videoDevice != null &&
videoDevice.IsRunning)
    {

```

```

        videoDevice.SignalToStop();
        videoDevice.WaitForStop();
        videoDevice = null;
    }
}

private void
videoSourcePlayer1_NewFrame(object sender, ref Bitmap
image)
{
    try
    {
        Bitmap;
        sourceImage = image.Clone() as

        if (sourceImage != null)
        {
            YCbCrColorFiltering(sourceImage);

            ObjectTracking(sourceImage);
        }
    }
    catch { }
}

private void Autorun_Load_1(object sender,
EventArgs e)
{
    button9.Click += button9_Click;

    // TAMBAHIN INI
    comboBoxColors.DataSource = colors;
    comboBoxColors.DisplayMember = "Name";

    // Menambahkan event handler untuk
    perubahan pilihan di combo box
    comboBoxColors.SelectedIndexChanged +=
comboBoxColors_SelectedIndexChanged;

    videoDevices = new
FilterInfoCollection(FilterCategory.VideoInputDevice)
;

```

```

        if (videoDevices.Count != 0)
        {
            foreach (FilterInfo device in
videoDevices)
            {
                comboBox1.Items.Add(device.Name);
            }
        }
        else
        {
            comboBox1.Items.Add("No DirectShow
devices found");
        }
        comboBox1.SelectedIndex = 0;

comboBoxColors_SelectedIndexChanged(comboBoxColors,
EventArgs.Empty);
    }

    private void groupBox3_Enter(object sender,
EventArgs e)
    {

    }

    private void ResetDetectedImages()
    {
        pictureBox1.Image = null;
    }

    private void button10_Click(object sender,
EventArgs e)
    {
        int a = 1;
        int write = a;
        plc.WriteSingleRegister(0, write);
        ColorData selectedColor =
(ColorData)comboBoxColors.SelectedItem;

        // Gunakan nilai-nilai warna yang dipilih
        float Ymin = selectedColor.Ymin;
        float Ymax = selectedColor.Ymax;

```

```

        float Cbmin = selectedColor.Cbmin;
        float Cbmax = selectedColor.Cbmax;
        float Crmin = selectedColor.Crmin;
        float Crmax = selectedColor.Crmax;
    }

    private void ResetTrackedImages()
    {
        pictureBox2.Image = null;
        ResetLabel4();
    }

    private void Form1_FormClosing(object sender,
FormClosingEventArgs e)
    {
        // Pastikan kamera ditutup sebelum
        menutup aplikasi
        if (videoDevice != null &&
        videoDevice.IsRunning)
        {
            videoDevice.SignalToStop();
            videoDevice.WaitForStop();
            videoDevice = null;
        }
    }

    private void HistoryAlarm_Tick(object sender,
    EventArgs e)
    {
        Alarm();
        AlarmStatus();
    }

    private void
    comboBoxColors_SelectedIndexChanged_1(object sender,
    EventArgs e)
    {
    }

```

```

        private void button4_Click(object sender,
EventArgs e)
        {
            var value = plc.ReadHoldingRegisters(0,
11);

            if (value[10] == 0)
            {

            }

            else
            {
                int i = 1;
                int write = i;
                plc.WriteSingleRegister(8, write);
            }

            int[] read = plc.ReadHoldingRegisters(0,
11);
            i = read[10];

        }

        private void RED0_Click(object sender,
EventArgs e)
        {

        }

        private void label6_Click(object sender,
EventArgs e)
        {

        }

        private void label7_Click(object sender,
EventArgs e)
        {

        }

```

```

//TAMBAHIN INI SAMPAI TERAKHIR
public class ColorData
{
    public string Name { get; set; }
    public float Ymin { get; set; }
    public float Ymax { get; set; }
    public float Cbmin { get; set; }
    public float Cbmax { get; set; }
    public float Crmin { get; set; }
    public float Crmax { get; set; }

    public override string ToString()
    {
        return Name;
    }
}

private void GREEN0_Click(object sender,
EventArgs e)
{

}

private List<ColorData> colors = new
List<ColorData>
{
    new ColorData { Name = "WHITE", Ymin =
0.77f, Ymax = 1f, Cbmin = -0.33f, Cbmax = 0.5f, Crmin
= -0.5f, Crmax = 0.5f },
    new ColorData { Name = "BLUE", Ymin =
0.65f, Ymax = 1f, Cbmin = 0.02f, Cbmax = 0.3f, Crmin
= -0.49f, Crmax = 0f },
    new ColorData { Name = "RED", Ymin = 0f,
Ymax = 0.75f, Cbmin = -0.5f, Cbmax = 0.18f, Crmin =
0.08f, Crmax = 0.3f }
};
private int counter;

private void
comboBoxColors_SelectedIndexChanged(object sender,
EventArgs e)

```

```

        {
            // Mendapatkan warna yang dipilih dari
            combo box
            ColorData selectedColor =
            (ColorData)comboBoxColors.SelectedItem;

            // Memperbarui nilai-nilai batas warna
            berdasarkan warna yang dipilih
            Ymin = selectedColor.Ymin;
            Ymax = selectedColor.Ymax;
            Cbmin = selectedColor.Cbmin;
            Cbmax = selectedColor.Cbmax;
            Crmin = selectedColor.Crmin;
            Crmax = selectedColor.Crmax;
        }
    }
}

```

#### D. Manual I/O

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Collections;
using AForge;
using AForge.Imaging;
using AForge.Imaging.Filters;
using AForge.Video;
using AForge.Video.DirectShow;
using EasyModbus;
using AForge.Math.Geometry;
using System.Reflection.Emit;
using System.IO;
using System.Security.Cryptography.X509Certificates;

namespace UPPERHMI1.UserControls
{

```

```

public partial class Autorun: UserControl
{
    private Panel container;
    public int a; //WriteInput_PB.Start
    public int b; //Trigger on camera
    public int c; //object good
    public int d; //object not good
    public int e; //reset all memori
    public int f; //alarm
    public int g; //ReadOutputMerah
    public int h; //ReadOutputKuning
    public int i; //ReadOutputHijau
    public int j; //Connect_indicator
    public int k; //Reset Button

    ModbusClient plc = new
ModbusClient("192.168.1.5", 502); //IP Address PLC

    Bitmap sourceImage;
    Bitmap detectedImage;

    float Ymin = 0, Ymax = 1;
    float Cbmin = -0.5f, Cbmax = 0.5f;
    float Crmin = 0.5f, Crmax = 0.5f;

    int TRACK_SPACE = 2;

    private FilterInfoCollection videoDevices;
    private VideoCaptureDevice videoDevice;
    private ArrayList listCamera = new ArrayList();

    public Autorun()
    {
        InitializeComponent();
        panelContainer = new Panel();
        panelContainer.Dock = DockStyle.Fill;
        this.Controls.Add(panelContainer);

        //TAMBAHIN INI
        comboBoxColors.DataSource = colors;
        comboBoxColors.DisplayMember = "Name";
    }
}

```

```

private static string _usbcamera;
public string usbcamera
{
    get { return _usbcamera; }
    set { _usbcamera = value; }
}

private void OpenVideoSource(IVideoSource
source)
{
    try
    {
        this.Cursor = Cursors.WaitCursor;
        CloseCurrentVideoSource();
        videoSourcePlayer1.VideoSource = source;
        videoSourcePlayer1.Start();
        this.Cursor = Cursors.Default;
    }
    catch { }
}

public void CloseCurrentVideoSource()
{
    try
    {
        if (videoSourcePlayer1.VideoSource !=
null)
        {
            videoSourcePlayer1.SignalToStop();

            for (int i = 0; i < 30; i++)
            {
                if
(!videoSourcePlayer1.IsRunning)
                    break;

                System.Threading.Thread.Sleep(100);
            }
            if (videoSourcePlayer1.IsRunning)
            {
                videoSourcePlayer1.Stop();
            }
        }
    }
}

```



```

        foreach (Rectangle recs in rects)
        {
            if (rects.Length > 0)
            {
                Rectangle objectRect = rects[0];
                Graphics graph =
Graphics.FromImage(srcImage);
                using (Pen pen = new
Pen(Color.FromArgb(255, 0, 0), 10))
                {
                    graph.DrawRectangle(pen,
objectRect);
                }
                graph.Dispose();
            }
        }

        pictureBox2.Image = srcImage;

        label4.BeginInvoke((MethodInvoker)(() =>
        {
            label4.Text = objectDetected ?
"Detected": "FALSE";
        }));
    }

    private void button1_Click(object sender,
EventArgs e)
    {
        OpenCamera();
        ColorData selectedColor =
(ColorData)comboBoxColors.SelectedItem;

        // Gunakan nilai-nilai warna yang dipilih
        float Ymin = selectedColor.Ymin;
        float Ymax = selectedColor.Ymax;
        float Cbmin = selectedColor.Cbmin;
        float Cbmax = selectedColor.Cbmax;
        float Crmin = selectedColor.Crmin;
        float Crmax = selectedColor.Crmax;
    }

```

```

    }

    private void button2_Click(object sender,
EventArgs e)
    {
        if ((videoDevice != null))
        {
            try
            {

                ((VideoCaptureDevice)videoDevice).DisplayPropertyPage(this.Handle);

            }
            catch (NotSupportedException ex)
            {
                MessageBox.Show(ex.Message, "Error",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }

    private void Resetallmemori()
    {
        var value = plc.ReadHoldingRegisters(0, 5);

        if (value[4] == 1)
        {

            ResetTrackedImages();
            ResetLabel4();
            int c = 0;
            int write = c;
            plc.WriteSingleRegister(2, write);

        }

        else
        {

        }

        int[] read = plc.ReadHoldingRegisters(0, 5);

```

```

        e = read[4];
    }

    private void Cameratrigger()
    {
        try
        {
            var value = plc.ReadHoldingRegisters(0,
2);

            if (value[1] == 1)
            {
                OpenCamera();
                CheckLabel();
            }

            else
            {
                CloseCamera();
                ResetDetectedImages();
            }

            int[] read = plc.ReadHoldingRegisters(0,
2);

            b = read[1];
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error communicating
with PLC: " + ex.Message);
            WriteMessageToNotepad("Error
communicating with PLC: " + ex.Message);
            Application.Exit(); // Menutup aplikasi
            karena kesalahan yang fatal
        }
    }

    private void Autorun_FormClosing(object sender,
FormClosingEventArgs e)
    {
        // Pastikan kamera ditutup sebelum menutup
        aplikasi
        CloseCamera();
        // Hentikan timer jika masih berjalan
    }

```

```

    }

    private void button3_Click(object sender,
EventArgs e)
    {
        CloseCamera();
        ResetDetectedImages();
        ResetLabel4();
        ResetTrackedImages();
    }

    private void Form1_FormClosed(object sender,
FormClosedEventArgs e)
    {
        if (videoDevice != null &&
videoDevice.IsRunning)
            videoDevice.Stop();
    }

    //untuk menjalankan sequence ketika objek telah
terdeteksi
    public void CheckLabel()
    {
        if (label4.Text == "Detected")
        {
            CloseCamera();
            ResetDetectedImages();
            int c = 1;
            int write = c;
            plc.WriteSingleRegister(2, write);
        }

        else if (label4.Text == "FALSE")
        {
            CloseCamera();
            ResetDetectedImages();
            int c =2;
            int write = c;
            plc.WriteSingleRegister(2, write);
        }
    }
}

```

```

        private void button9_Click(object sender,
EventArgs e)
        {
            try
            {
                if (plc.Connected)
                {
                    plc.Disconnect();
                    label7.Text = "Disconnected";
                    MessageBox.Show("Disconnected from
PLC");
                    int j = 0;
                    int write = j;
                    plc.WriteSingleRegister(9, write);
                }
                else
                {
                    MessageBox.Show("Already
disconnected from PLC!");
                    return;
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Error disconnecting
from PLC: " + ex.Message);
            }
        }

        private void connect_indicator()
        {
            var value = plc.ReadHoldingRegisters(0, 10);

            if (value[9] == 1)
            {
                REDCON.Visible = FALSE;
                GREENCON.Visible = true;
            }
            else if (value[9] == 0)
            {
                REDCON.Visible = true;
                GREENCON.Visible = FALSE;
            }
        }
    }

```

```

        private void button8_Click(object sender,
EventArgs e)
        {
            try
            {
                plc.Connect();
                if (plc.Connected == true)
                {
                    label7.Text = "Connected";
                    MessageBox.Show("Connected");
                    WriteMessageToNotepad("Connected to
PLC");

                    int j = 1;
                    int write = j;
                    plc.WriteSingleRegister(9, write);
                }
                else
                {
                    MessageBox.Show("Failed to connect
to PLC!");
                    WriteMessageToNotepad("Failed to
connect to PLC");

                    int j = 0;
                    int write = j;
                    plc.WriteSingleRegister(9, write);
                }

                var readplcTimer = new Timer()
                {
                    Interval = 1500,
                    Enabled = true
                };

                var HistoryAlarm = new Timer()
                {
                    Interval = 10,
                    Enabled = true
                };

                readplcTimer.Tick += ReadplcTimer_Tick;
                readplcTimer.Start();
                HistoryAlarm.Tick += HistoryAlarm_Tick;
                HistoryAlarm.Start();
            }
        }
    }
}

```

```

        Cameratrigger();
        Alarm();
        AlarmStatus();
        connect_indicator();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error connecting to
PLC: " + ex.Message);
        WriteMessageToNotepad("Error connecting
to PLC");
        // Ensure lights indicate disconnection
in case of error
    }
}

private int Delaytime;
private int goodCounter;
private int notGoodCounter;
private void Counter()
{
    int [] read = plc.ReadHoldingRegisters(0,
15);

    int goodCounter = read[11];
    int notGoodCounter = read[12];
    int Delaytime = read[14];
    textBox1.Text = goodCounter.ToString();
    textBox2.Text = notGoodCounter.ToString();
    textBox3.Text = Delaytime.ToString();

}
private int lastAlarmState = 0;

private void Alarm()
{
    var value = plc.ReadHoldingRegisters(0, 8);
    int currentAlarmState = value[7];

    if (currentAlarmState != lastAlarmState)
    {
        switch (currentAlarmState)
        {
            case 1:

```

```

        MessageBox.Show("Transfer
Error");
        WriteMessageToNotepad("Transfer
Error");
        break;
    case 2:
        MessageBox.Show("Vacuum
Failure");
        WriteMessageToNotepad("Vacuum
Failure");
        break;
    case 4:
        MessageBox.Show("Product No
Detect");
        WriteMessageToNotepad("Product
No Detect");
        break;
    default:
        // Handle other cases or default
        behavior if needed
        break;
    }

    // Update the lastAlarmState to the
current state
    lastAlarmState = currentAlarmState;
}

int[] read = plc.ReadHoldingRegisters(0, 8);
f = read[7];
}

private void AlarmStatus ()
{
    var value = plc.ReadHoldingRegisters(0, 9);

    if (value[5] == 1)
    {
        RED1.Visible = true;
        RED0.Visible = FALSE;
        YELLOW1.Visible = true;
        YELLOW0.Visible = FALSE;
        GREEN1.Visible = FALSE;
        GREEN0.Visible = true;
    }
}

```

```

    }
    else if (value[5] == 2)
    {
        RED1.Visible = FALSE;
        RED0.Visible = true;
        YELLOW1.Visible = FALSE;
        YELLOW0.Visible = true;
        GREEN1.Visible = FALSE;
        GREEN0.Visible = true;
    }

    else if (value[5] == 4)
    {
        RED1.Visible = FALSE;
        RED0.Visible = true;
        YELLOW1.Visible = true;
        YELLOW0.Visible = FALSE;
        GREEN1.Visible = true;
        GREEN0.Visible = FALSE;
    }
}

private void WriteMessageToNotepad(string
message)
{
    try
    {
        using (StreamWriter sw = new
StreamWriter("error_log.txt", true))
        {
            sw.WriteLine($"{DateTime.Now} -
{message}");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error writing to
Notepad: " + ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    }
}

private void button7_Click(object sender,
EventArgs e)

```

```

    {
        int a = 0;
        int write = a;
        plc.WriteSingleRegister(0, write);

        CloseCamera();
        Resetallmemori();
    }
    private void timer1_Tick(object sender,
EventArgs e)
    {
        Cameratriger();
    }

    private void ReadplcTimer_Tick(object sender,
EventArgs e)
    {
        Cameratriger();
        Resetallmemori();
        connect_indicator();
        Counter();
    }

    private void ResetLabel4()
    {
        label4.BeginInvoke((MethodInvoker)(() =>
        {
            label4.Text = ". . . . .";
        }));
    }

    private void OpenCamera()
    {
        try
        {
            usbcamera =
comboBox1.SelectedIndex.ToString();
            videoDevices = new
FilterInfoCollection(FilterCategory.VideoInputDevice);
            if (videoDevices.Count != 0)
            {
                foreach (FilterInfo device in
videoDevices)

```

```

        {
            listCamera.Add(device.Name);
        }
    }
    else
    {
        MessageBox.Show("Camera Devices
Found");
    }
    videoDevice = new
VideoCaptureDevice(videoDevices[Convert.ToInt32(usbcamer
a)].MonikerString);
        OpenVideoSource(videoDevice);
    }
    catch (Exception err)
    {
        MessageBox.Show(err.ToString());
    }
}
private void CloseCamera()
{
    if (videoDevice != null &&
videoDevice.IsRunning)
    {
        videoDevice.SignalToStop();
        videoDevice.WaitForStop();
        videoDevice = null;
    }
}

private void videoSourcePlayer1_NewFrame(object
sender, ref Bitmap image)
{
    try
    {
        sourceImage = image.Clone() as Bitmap;

        if (sourceImage != null)
        {
            YCbCrColorFiltering(sourceImage);

            ObjectTracking(sourceImage);
        }
    }
}

```

```

    }
    catch { }
}

private void Autorun_Load_1(object sender,
EventArgs e)
{
    button9.Click += button9_Click;

    // TAMBAHIN INI
    comboBoxColors.DataSource = colors;
    comboBoxColors.DisplayMember = "Name";

    // Menambahkan event handler untuk perubahan
    pilihan di combo box
    comboBoxColors.SelectedIndexChanged +=
    comboBoxColors_SelectedIndexChanged;

    videoDevices = new
    FilterInfoCollection(FilterCategory.VideoInputDevice);
    if (videoDevices.Count != 0)
    {
        foreach (FilterInfo device in
        videoDevices)
        {
            comboBox1.Items.Add(device.Name);
        }
    }
    else
    {
        comboBox1.Items.Add("No DirectShow
        devices found");
    }
    comboBox1.SelectedIndex = 0;

    comboBoxColors_SelectedIndexChanged(comboBoxColors,
    EventArgs.Empty);
}

private void groupBox3_Enter(object sender,
EventArgs e)
{
}

```

```

private void ResetDetectedImages()
{
    pictureBox1.Image = null;
}

private void button10_Click(object sender,
EventArgs e)
{
    int a = 1;
    int write = a;
    plc.WriteSingleRegister(0, write);
    ColorData selectedColor =
(ColorData)comboBoxColors.SelectedItem;

    // Gunakan nilai-nilai warna yang dipilih
    float Ymin = selectedColor.Ymin;
    float Ymax = selectedColor.Ymax;
    float Cbmin = selectedColor.Cbmin;
    float Cbmax = selectedColor.Cbmax;
    float Crmin = selectedColor.Crmin;
    float Crmax = selectedColor.Crmax;

}

private void ResetTrackedImages()
{
    pictureBox2.Image = null;
    ResetLabel4();

}

private void Form1_FormClosing(object sender,
FormClosingEventArgs e)
{
    // Pastikan kamera ditutup sebelum menutup
    aplikasi
    if (videoDevice != null &&
videoDevice.IsRunning)
    {
        videoDevice.SignalToStop();
        videoDevice.WaitForStop();
        videoDevice = null;
    }
}

```

```

    }

    private void HistoryAlarm_Tick(object sender,
EventArgs e)
    {
        Alarm();
        AlarmStatus();
    }

    private void
comboBoxColors_SelectedIndexChanged_1(object sender,
EventArgs e)
    {

    }

    private void button4_Click(object sender,
EventArgs e)
    {
        var value = plc.ReadHoldingRegisters(0, 11);

        if (value[10] == 0)
        {

        }

        else
        {
            int i = 1;
            int write = i;
            plc.WriteSingleRegister(8, write);
        }

        int[] read = plc.ReadHoldingRegisters(0,
11);
        i = read[10];

    }

    private void RED0_Click(object sender, EventArgs
e)
    {

```

```

    }

    private void label6_Click(object sender,
EventArgs e)
    {

    }

    private void label7_Click(object sender,
EventArgs e)
    {

    }

    //TAMBAHIN INI SAMPAI TERAKHIR
    public class ColorData
    {
        public string Name { get; set; }
        public float Ymin { get; set; }
        public float Ymax { get; set; }
        public float Cbmin { get; set; }
        public float Cbmax { get; set; }
        public float Crmin { get; set; }
        public float Crmax { get; set; }

        public override string ToString()
        {
            return Name;
        }
    }

    private void GREEN0_Click(object sender,
EventArgs e)
    {

    }

    private List<ColorData> colors = new
List<ColorData>
    {
        new ColorData { Name = "WHITE", Ymin =
0.77f, Ymax = 1f, Cbmin = -0.33f, Cbmax = 0.5f, Crmin =
-0.5f, Crmax = 0.5f },

```

```

        new ColorData { Name = "BLUE", Ymin = 0.65f,
Ymax = 1f, Cbmin = 0.02f, Cbmax = 0.3f, Crmin = -0.49f,
Crmax = 0f },
        new ColorData { Name = "RED", Ymin = 0f,
Ymax = 0.75f, Cbmin = -0.5f, Cbmax = 0.18f, Crmin =
0.08f, Crmax = 0.3f }
    };
    private int counter;

    private void
comboBoxColors_SelectedIndexChanged(object sender,
EventArgs e)
    {
        // Mendapatkan warna yang dipilih dari combo
box
        ColorData selectedColor =
(ColorData)comboBoxColors.SelectedItem;

        // Memperbarui nilai-nilai batas warna
berdasarkan warna yang dipilih
        Ymin = selectedColor.Ymin;
        Ymax = selectedColor.Ymax;
        Cbmin = selectedColor.Cbmin;
        Cbmax = selectedColor.Cbmax;
        Crmin = selectedColor.Crmin;
        Crmax = selectedColor.Crmax;
    }
}
}

```