



**TUGAS AKHIR - TE 141599**

**RANCANG BANGUN SISTEM PENDETEKSI LUBANG PADA  
KERTAS TISU BERBASIS INFORMASI VISUAL DENGAN  
RASPBERRY PI**

Andre Raditya Mutalip  
NRP 2210100143

Dosen Pembimbing  
Suwito, S.T., M.T.  
Ir. Tasripan, M.T.

JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015



**FINAL PROJECT - TE 141599**

**DESIGN OF HOLES DETECTION SYSTEM ON TISSUE PAPER  
BASED ON VISUAL INFORMATION WITH RASPBERRY PI**

Andre Raditya Mutalip  
NRP 2210100143

*Advisor*  
Suwito, S.T., M.T.  
Ir. Tasripan, M.T.

*DEPARTMENT OF ELECTRICAL ENGINEERING  
Faculty of Industrial Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2015*

**RANCANG BANGUN SISTEM PENDETEKSI LUBANG  
PADA KERTAS TISU BERBASIS INFORMASI VISUAL  
DENGAN RASPBERRY PI**

**TUGAS AKHIR**

**Diajukan Untuk Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik**

**Pada**

**Bidang Studi Elektronika**

**Jurusan Teknik Elektro**

**Institut Teknologi Sepuluh Nopember**

**Menyetujui :**

**Dosen Pembimbing I**



**Suwito, S.T., M.T.**

**NIP. 19810105200501104**

**Dosen Pembimbing II**



**Ir. Tasripan, M.T.**

**NIP. 196204181990031004**



**SURABAYA**

**JANUARI, 2015**

# **RANCANG BANGUN SISTEM PENDETEKSI LUBANG PADA KERTAS TISU BERBASIS INFORMASI VISUAL DENGAN RASPBERRY PI**

Andre Raditya Mutalip  
2210100143

Dosen Pembimbing I : Suwito, S.T., M.T.

Dosen Pembimbing II : Ir. Tasripan, M.T.

## **ABSTRAK**

Industri kertas tisu menggunakan sistem pendeteksi visual untuk mengendalikan kualitas seperti cacat berupa lubang dari produknya, tetapi sistem berupa kamera-kamera industri ini begitu mahal harganya. Maka dari itu, Tugas Akhir ini mengembangkan sistem pendeteksi lubang pada kertas tisu berbasis informasi visual dengan Raspberry Pi yang ekonomis. Hasilnya adalah sebuah sistem yang mengambil citra kertas tisu dari kamera, lalu diolah di Raspberry Pi dengan sebuah program yang disokong OpenCV. Jika terdeteksi lubang di kertas tisu, maka *buzzer* berbunyi melalui PLC. Berikutnya, dilakukan simulasi dan pengujian mekanisme penggulungan kertas tisu dengan lubang paling kecil 25 mm<sup>2</sup> dan ditemukan kecepatan optimal sistem 0,27 m/s. Hasilnya adalah sistem dengan nilai batas bawah 90° dan nilai batas atas 255° untuk programnya dan dapat mendeteksi lubang pada kertas tisu dengan kecepatan penggulungan maksimal 0,27 m/s. Sistem ini dapat ditingkatkan kecepatannya dengan penggunaan nilai batas adaptif, penggunaan dua atau lebih SBC, dan sistem pendingin. Sistem pun akan lebih mudah digunakan jika ditambahkan HMI, *data logger*, dan penanda kertas tisu.

**Kata kunci: Kertas tisu, pengendalian kualitas, pengolahan citra, Raspberry Pi, PLC**

# **DESIGN OF HOLES DETECTION SYSTEM ON TISSUE PAPER BASED ON VISUAL INFORMATION WITH RASPBERRY PI**

Andre Raditya Mutalip  
2210100143

1<sup>st</sup> Advisor : Suwito, S.T., M.T.

2<sup>nd</sup> Advisor : Ir. Tasripan, M.T.

## **ABSTRACT**

*Tissue paper industries uses visual detection system to control the quality of defects such as holes from the product, but the system in the form of industrial cameras is very expensive. Therefore, this Final Project developed a holes detection system on tissue paper based on visual information with Raspberry Pi which is economical. The result is a system that captures the image of the tissue paper from the camera, and then processed in the Raspberry Pi with a program that is supported by OpenCV. If a hole is detected in the tissue paper, then the buzzer that is controlled by the PLC sounds. Next, the simulation and the testing of tissue paper winding mechanism with the smallest hole at 25 mm<sup>2</sup> found the optimal system speed of 0.27 m/s. The result is a system with a lower threshold of 90° and an upper threshold of 255° for the program and can detect holes in tissue paper with a maximum winding speed of 0.27 m/s. We can increase its speed with the use of adaptive threshold, the use of two or more SBC, and cooling systems. The system will be easier to use if we added HMI, data logger, and markers for tissue paper.*

**Keywords:** *tissue paper, quality control, image processing, Raspberry Pi, PLC*

## KATA PENGANTAR

Ucap puji dan syukur kepada Tuhan Yang Maha Kuasa atas rahmat dan karunia-Nya yang tiada akhirnya terutama berupa kekuatan, kesabaran, dan kelancaran dalam menyelesaikan Tugas Akhir dengan judul “Rancang Bangun Sistem Pendeteksi Lubang pada Kertas Tisu Berbasis Informasi Visual dengan Raspberry Pi”.

Tujuan utama dalam penulisan Tugas Akhir ini adalah sebagai persyaratan untuk menyelesaikan jenjang pendidikan S1 di Bidang Studi Elektronika, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.

Dalam kesempatan ini penulis ingin mengucapkan terimakasih kepada pihak-pihak yang berjasa dalam penulisan Tugas Akhir ini, yaitu:

1. Sege nap keluarga, terutama Papa dan Mama tercinta yang selalu memberikan dukungan dan doa bagi putranya.
2. Bapak Suwito, S.T., M.T., Bapak Ir. Hendra Kusuma M.Eng.Sc., dan Bapak Ir. Tasripan, M.T. selaku dosen pembimbing yang banyak memberikan saran dan bimbingan bagi Tugas Akhir ini.
3. Elfrida Maharani yang banyak memberikan dukungan dalam pengerjaan Tugas Akhir ini.
4. Imam Shalahuddin Tonaziz atas saran dan kerjasamanya selama pembuatan Tugas Akhir ini.
5. Sahabat-sahabat yang sudah banyak memberi dukungan dan hiburan selama pengerjaan Tugas Akhir ini.
6. Seluruh keluarga besar Jurusan Teknik Elektro ITS, para dosen, karyawan, dan kawan-kawan atas dukungan, masukan, dan kerjasamanya selama proses perkuliahan dan pengerjaan Tugas Akhir.

Surabaya, Januari 2015

Penulis

# DAFTAR ISI

	HALAMAN
<b>JUDUL</b>	
<b>LEMBAR PERNYATAAN KEASLIAN</b>	
<b>LEMBAR PENGESAHAN</b>	
<b>ABSTRAK</b> .....	i
<b>ABSTRACT</b> .....	iii
<b>KATA PENGANTAR</b> .....	v
<b>DAFTAR ISI</b> .....	vii
<b>DAFTAR GAMBAR</b> .....	xi
<b>DAFTAR TABEL</b> .....	xiii
<b>BAB 1 PENDAHULUAN</b>	
1.1 Latar Belakang Permasalahan .....	1
1.2 Permasalahan .....	2
1.3 Tujuan .....	3
1.4 Metodologi .....	3
1.5 Sistematika Penulisan .....	4
1.6 Relevansi .....	5
<b>BAB 2 DASAR TEORI</b>	
2.1 Kertas Tisu .....	7
2.2 Robekan .....	9
2.3 Pengendalian Kualitas .....	10
2.4 <i>Single Board Computer</i> (SBC) .....	10
2.4.1 Raspberry Pi .....	11
2.4.2 <i>Secure Digital</i> (SD) .....	12
2.5 Kamera Digital .....	13
2.5.1 Modul Kamera Raspberry Pi .....	14
2.6 <i>Programmable Logic Controller</i> (PLC) .....	15
2.6.1 PLC OMRON CPM2A-20CDR-A .....	16

2.6.2	<i>Ladder Diagram</i> .....	16
2.7	<i>Variable Frequency Drive (VFD)</i> .....	19
2.7.1	VFD Fuji Electric FRENIC-Mini FRN075CIS-7A .....	19
2.8	Motor Induksi 3 Fasa .....	21
2.9	Sistem <i>Conveyor</i> .....	22
2.10	<i>Relay</i> .....	23
2.11	<i>Buzzer</i> .....	25
2.12	Sumber Daya DC .....	26
2.13	Pengolahan Citra.....	26
2.13.1	<i>Open Source Computer Vision (OpenCV)</i> .....	27
2.13.2	<i>Grayscaleing</i> .....	27
2.13.3	Penghalusan .....	28
2.13.4	Penyetaraan Histogram.....	29
2.13.5	Pendeteksian Tepian Canny.....	30
2.13.6	Operasi Morfologi .....	32
2.13.7	Penemuan Kontur .....	34
2.14	Komunikasi Serial RS-232 .....	34
2.14.1	<i>Command Block dan Response Block PLC</i> .....	35
2.14.2	<i>Frame Check Sequence (FCS)</i> .....	36

## **BAB 3 PERANCANGAN SISTEM**

3.1	Perencanaan dan Penyusunan Konsep Sistem .....	38
3.1.1	Kebutuhan Sistem.....	39
3.1.2	Pertimbangan Perencanaan Sistem .....	40
3.1.3	Tuntutan Perencanaan Sistem.....	41
3.1.4	Penyusunan Konsep Sistem .....	43
3.2	Perancangan Perangkat Keras.....	43
3.2.1	Perancangan Rangkaian Elektrik .....	45
3.2.1.1	Perancangan Rangkaian Pembatas Arus .....	45
3.2.1.2	Perancangan Rangkaian <i>Pushbutton</i> .....	46
3.2.1.3	Perancangan Rangkaian <i>Driver Raspberry Pi Model B+</i> .....	47
3.2.1.4	Perancangan Rangkaian <i>Driver Light Emitting Diode (LED)</i> .....	48

3.2.1.5	Perancangan Rangkaian <i>Driver Buzzer</i> .....	49
3.2.2	Perancangan Mekanik.....	50
3.2.2.1	Perancangan Mekanik <i>Conveyor</i> .....	51
3.2.2.2	Perancangan Mekanik Sistem <i>Belt</i> dan <i>Pulley</i> .....	53
3.2.3	<i>Variable Frequency Drive</i> (VFD).....	54
3.2.3.1	VFD Fuji Electric FRENIC-Mini FRN075CIS- 7A.....	54
3.2.3.2	Motor Induksi 3 Fasa .....	57
3.2.4	<i>Programmable Logic Controller</i> (PLC).....	58
3.2.4.1	PLC OMRON CPM2A-20CDR-A .....	59
3.2.4.2	Sambungan Komunikasi Serial RS-232 .....	60
3.2.5	Raspberry Pi.....	61
3.2.5.1	Layar.....	62
3.2.5.2	<i>Keyboard</i> dan <i>Mouse</i> .....	62
3.2.5.3	Modul Kamera Raspberry Pi .....	63
3.2.6	Perangkat Tambahan.....	64
3.2.6.1	Sumber Daya DC.....	64
3.2.6.2	Lampu Penerangan .....	65
3.2.6.3	Lengan Mekanik Raspberry Pi Model B+ Beserta Modul Kamera.....	65
3.3	Perancangan Perangkat Lunak .....	66
3.3.1	PC .....	68
3.3.1.1	Program Komunikasi Serial RS-232.....	69
3.3.1.2	Program Pendeteksi Lubang pada Kertas Tisu .....	70
3.3.1.3	Program PLC .....	72
3.3.2	Raspberry Pi Model B+ .....	74
3.4	Cara Kerja Keseluruhan Sistem .....	74
 <b>BAB 4 PENGUJIAN SISTEM</b>		
4.1	Pengujian Bagian-Bagian Penyusun Sistem.....	77
4.1.1	Pengujian Sumber Daya DC .....	77
4.1.2	Pengujian Pembatas Arus .....	78

4.1.3	Pengujian <i>Pushbutton</i> .....	78
4.1.4	Pengujian <i>Driver</i> Raspberry Pi Model B+.....	79
4.1.5	Pengujian <i>Driver</i> LED.....	79
4.1.6	Pengujian <i>Driver Buzzer</i> .....	80
4.1.7	Pengujian Komponen <i>Buzzer</i> .....	80
4.1.8	Pengujian PLC.....	81
4.1.9	Pengujian VFD.....	82
4.1.10	Pengujian Komunikasi Serial RS-232.....	83
4.2	Pengujian Keseluruhan Sistem Terhadap Kecepatan <i>Conveyor</i> .....	84
<b>BAB 5 PENUTUP</b>		
5.1	Kesimpulan.....	89
5.2	Saran.....	89
<b>DAFTAR PUSTAKA</b> .....		91
<b>LAMPIRAN</b> .....		95
<b>BIODATA PENULIS</b>		

## DAFTAR TABEL

	HALAMAN
Tabel 3.1	Konfigurasi I/O VFD..... 55
Tabel 3.2	Konfigurasi I/O PLC ..... 59
Tabel 3.3	Konfigurasi I/O Raspberry Pi Model B+..... 61
Tabel 3.4	<i>Setting</i> komunikasi serial RS-232..... 69
Tabel 3.5	Konfigurasi alamat PLC ..... 73
Tabel 4.1	Hasil pengujian sumber daya DC ..... 77
Tabel 4.2	Hasil pengujian pembatas arus ..... 78
Tabel 4.3	Hasil pengujian <i>pushbutton</i> ..... 78
Tabel 4.4	Hasil pengujian <i>driver</i> Raspberry Pi..... 79
Tabel 4.5	Hasil pengujian <i>driver</i> LED..... 79
Tabel 4.6	Hasil pengujian <i>driver buzzer</i> ..... 80
Tabel 4.7	Hasil pengujian komponen <i>buzzer</i> ..... 80
Tabel 4.8	Hasil pengujian keluaran PLC ..... 81
Tabel 4.9	Hasil pengujian VFD ..... 82
Tabel 4.10	Hasil pengujian komunikasi serial RS-232..... 83
Tabel 4.11	Hasil pengujian keseluruhan sistem terhadap kecepatan <i>conveyor</i> ..... 86
Tabel 4.12	Persentase <i>error</i> pengujian keseluruhan sistem terhadap kecepatan <i>conveyor</i> ..... 88

## DAFTAR GAMBAR

	HALAMAN
Gambar 2.1 Proses pabrikasi kertas tisu .....	8
Gambar 2.2 Raspberry Pi Model B+ .....	11
Gambar 2.3 Ukuran <i>Secure Digital</i> (SD) .....	13
Gambar 2.4 Perintah <i>LOAD ladder diagram</i> .....	17
Gambar 2.5 Perintah <i>LOAD NOT ladder diagram</i> .....	17
Gambar 2.6 Perintah <i>AND ladder diagram</i> .....	17
Gambar 2.7 Perintah <i>AND NOT ladder diagram</i> .....	18
Gambar 2.8 Perintah <i>OR ladder diagram</i> .....	18
Gambar 2.9 Perintah <i>OR NOT ladder diagram</i> .....	18
Gambar 2.10 Perintah <i>OUT ladder diagram</i> .....	18
Gambar 2.11 Perintah <i>OUT NOT ladder diagram</i> .....	19
Gambar 2.12 Perintah <i>END ladder diagram</i> .....	19
Gambar 2.13 Grafik hubungan kecepatan dan torsi motor listrik .....	20
Gambar 2.14 Ilustrasi motor Induksi 3 fasa .....	22
Gambar 2.15 <i>Relay</i> HRS4H-S-DC24V .....	24
Gambar 2.16 <i>Buzzer</i> SFM-27 .....	25
Gambar 2.17 Blok diagram sumber daya DC .....	26
Gambar 2.18 Contoh proses <i>grayscale</i> .....	27
Gambar 2.19 Grafik <i>kernel</i> Gaussian 1 dimensi .....	28
Gambar 2.20 Contoh histogram suatu citra .....	29
Gambar 2.21 Contoh proses penyetaraan histogram dan hasilnya .....	30
Gambar 2.22 Contoh hasil proses pendeteksian tepian Canny .....	31
Gambar 2.23 Contoh proses <i>dilation</i> .....	33
Gambar 2.24 Contoh proses <i>erosion</i> .....	33
Gambar 2.25 Contoh proses <i>opening</i> .....	33
Gambar 2.26 Contoh proses <i>closing</i> .....	34
Gambar 2.27 Contoh hasil proses penemuan kontur .....	34
Gambar 2.28 <i>Format command block</i> .....	35
Gambar 2.29 <i>Format response block</i> .....	36
Gambar 3.1 Alur perancangan sistem .....	37
Gambar 3.2 Konfigurasi perangkat keras .....	44

Gambar 3.3	Rancangan skematik pembatas arus .....	46
Gambar 3.4	Rancangan skematik <i>pushbutton</i> .....	46
Gambar 3.5	Rancangan skematik <i>driver</i> Raspberry Pi Model B+ .....	47
Gambar 3.6	Rancangan skematik <i>driver Light Emitting Diode (LED)</i> .....	48
Gambar 3.7	Rancangan skematik <i>driver buzzer</i> .....	49
Gambar 3.8	Rancangan mekanik keseluruhan .....	50
Gambar 3.9	Rancangan mekanik <i>conveyor</i> sebelumnya.....	51
Gambar 3.10	Rancangan mekanik <i>conveyor</i> .....	52
Gambar 3.11	Rancangan mekanik sistem <i>belt</i> dan <i>pulley</i> .....	53
Gambar 3.12	Rancangan skematik kontrol <i>sourcing</i> VFD dari PLC....	55
Gambar 3.13	<i>Wiring</i> VFD Fuji Electric FRENIC-Mini FRN001C1□-2U .....	56
Gambar 3.14	VFD Fuji Electric FRENIC-Mini FRN075CIS-7A.....	57
Gambar 3.15	<i>Wiring</i> motor induksi 3 fasa .....	57
Gambar 3.16	Motor induksi 3 fasa .....	58
Gambar 3.17	PLC OMRON CPM2A-20CDR-A.....	59
Gambar 3.18	<i>Wiring</i> PLC OMRON CPM2A-20CDR-A.....	60
Gambar 3.19	<i>Wiring</i> komunikasi serial RS-232 .....	60
Gambar 3.20	<i>Port</i> Raspberry Pi Model B+.....	62
Gambar 3.21	<i>Keyboard</i> dan <i>mouse</i> Logitech MK220 .....	63
Gambar 3.22	Modul kamera Raspberry Pi.....	63
Gambar 3.23	Sumber daya DC HILED FX150-H1V24 .....	64
Gambar 3.24	Lampu LED Kris.....	65
Gambar 3.25	Lengan Raspberry Pi Model B+ berserta modul kamera	65
Gambar 3.26	Cara kerja program keseluruhan .....	67
Gambar 3.27	Cara kerja program C++ .....	68
Gambar 3.28	<i>Format command block</i> penulisan area IR/SR.....	70
Gambar 3.29	Hasil proses pengolahan citra .....	72
Gambar 3.30	Keseluruhan sistem .....	75
Gambar 4.1	Jenis dan ukuran lubang.....	85
Gambar 4.2	Grafik persentase <i>error</i> pengujian keseluruhan sistem terhadap kecepatan <i>conveyor</i> .....	88

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang Permasalahan

Produksi merupakan kegiatan manusia untuk mengolah barang atau jasa dan meningkatkan kualitas atau kuantitas dari barang atau jasa tersebut untuk memenuhi kebutuhan manusia. Banyak faktor yang berperan dalam memperoleh kuantitas dan kualitas produksi yang baik.

Dalam pemilihan setiap produk yang akan dikonsumsi, konsumen seringkali mempertimbangkan kualitas dari produk tersebut dan sama halnya dengan perusahaan dalam memproduksi dan menyalurkan suatu produk selalu mengaitkan dengan kualitas. Kualitas merupakan derajat atau tingkat karakteristik yang melekat pada produk sehingga dapat memenuhi persyaratan dan keinginan. Hal ini berarti produsen harus dapat mengenali karakteristik dari suatu produk yang berhubungan dengan kualitas, lalu membuat suatu tolak ukur untuk mengendalikannya.

Pengertian ini jelas berhubungan dengan kepuasan pelanggan saat memakai produk, dalam hal ini kertas tisu. Pelanggan akan berharap produk yang ia beli bisa digunakan dengan baik sesuai ketentuannya. Diharapkan produk dapat memenuhi permintaan baik dari segi kinerja, keistimewaan tambahan, kehandalan, kesesuaian, daya tahan, kemampuan melayani, estetika, dan ketepatannya.

Hanya sedikit perangkat keras elektronik yang mengundang antusiasme sebesar Raspberry Pi yang dirancang untuk membantu pelajaran teknologi komputer di sekolah-sekolah. Pada tahun 2013 saja saat Raspberry Pi merayakan ulang tahunnya yang pertama, perangkat ini sudah terjual lebih dari satu juta buah. Hal ini mengindikasikan bahwa perangkat ini sudah mulai dieksploitasi oleh para pengembang sebagai alternatif yang lebih murah dan efektif untuk kepentingan komersial<sup>[1]</sup>.

Sistem baru seperti Raspberry Pi akan mengubah pasar komputer industri murah ketika penghobi, pengajar, peneliti, dan pengembang mulai berpindah ke Raspberry Pi. Pabrikasi yang berkompetisi dalam pasar kontrol industri, kontrol otomatisasi, dan

SCADA harus mengawasi perkembangan Raspberry Pi yang terus berevolusi. Beberapa organisasi, mulai dari organisasi kecil sampai perusahaan besar seperti Intel telah memasuki pasar ini. *Raspberry Pi Foundation* pun adalah institusi yang tidak berfokus pada keuntungan dan dapat menurunkan harga tanpa perlu memperhatikan jumlah pendapatan. Pengembang-pengembang pun mulai mendapatkan pengalaman di bidang arsitektur ARM dengan sistem operasi seperti Linux. Inovasi ini didukung oleh sifatnya yang fleksibel dan *open source* sehingga Raspberry Pi akan terus berevolusi<sup>[2]</sup>.

Raspberry Pi yang diluncurkan pada tahun 2012 yang hanya berharga \$35 sudah terjual sebanyak dua juta buah pada tahun 2014 karena ledakan pengguna. Proyek yang dapat dilakukan oleh sistem hanya dibatasi oleh ambisi pengembangnya. Raspberry Pi bahkan digunakan sebagai proyek *Unmanned Aerial Vehicle* (UAV) sampai superkomputer. Sistem ini sukses karena sifatnya yang hanya butuh daya rendah, ringan, kecil, modular, fleksibel, dan interoperabilitas yang membuat pengembang makin kreatif<sup>[2]</sup>.

Maka dalam Tugas Akhir ini akan dibuat sebuah sistem sederhana yang digunakan untuk mengecek kualitas kertas tisu. Sistem ini diharapkan dapat mengecek adanya lubang atau tidak pada kertas tisu berdasarkan informasi visual yang didapat. Sistem ini merupakan integrasi dari kamera, Raspberry Pi, PLC, dan motor penggulung kertas tisu.

## **1.2 Permasalahan**

Permasalahan yang dibahas dalam Tugas Akhir ini adalah:

1. Pendeteksian lubang pada kertas tisu berbasis informasi visual dengan Raspberry Pi.
2. Pencarian kecepatan optimal di mana sistem masih dapat mendeteksi lubang dengan baik.

Dalam tugas ini ada beberapa asumsi yang membatasi masalah yang akan diuji di sistem, yaitu:

1. Kertas tisu berjenis kertas tisu wajah yang berbentuk gulungan dengan lebar 10 cm.
2. Kertas tisu tidak memiliki cacat berupa noda dan bagian yang terlipat.

3. Proses penggulungan di sistem ini hanya berupa simulasi dengan menggunakan kertas tisu yang ditempelkan ke *conveyor* yang diputar.
4. Lubang yang terdapat di kertas tisu paling kecil berukuran 25 mm<sup>2</sup>.
5. Biaya tidak dibahas dalam penulisan ini.

### **1.3 Tujuan**

Tujuan dari Tugas Akhir ini adalah merancang suatu sistem untuk mendeteksi ada atau tidaknya lubang pada kertas tisu dengan Raspberry Pi sebagai alternatif yang lebih murah sehingga mengetahui produk mana yang mengalami kecacatan. Sistem ini merupakan integrasi dari kamera, Raspberry Pi, dan PLC dan bekerja dengan mengolah informasi visual.

### **1.4 Metodologi**

Dalam penyelesaian Tugas Akhir ini digunakan metodologi sebagai berikut:

#### **1. Studi Literatur**

Pada tahap ini dilakukan pengumpulan dasar teori yang menunjang dalam penulisan Tugas Akhir. Dasar teori ini dapat diambil dari buku-buku, jurnal, *proceeding*, artikel-artikel di internet, dan forum-forum diskusi internet.

#### **2. Perencanaan dan Penyusunan Konsep Sistem**

Dari tahap sebelumnya, akan dilakukan perencanaan dan penyusunan konsep untuk menyusun kebutuhan dari sistem ini yang akan direalisasikan di tahap-tahap berikutnya.

#### **3. Perancangan Perangkat Keras**

Pada tahap berikut ini akan dirancang perangkat keras yang terintegrasi antara kamera, Raspberry Pi, PLC, VFD, dan motor *conveyor*.

Motor akan digunakan sebagai simulasi penggulungan kertas tisu. Lalu kamera akan terus merekam proses penggulungan dan menangkap citra dari kertas tisu yang lewat. Kemudian citra akan diproses oleh Raspberry Pi melalui program yang sudah dibuat. Citra akan diolah melalui program yang menggunakan *library* OpenCV 2.4.9. dan dari sini akan diketahui ada atau tidaknya lubang pada kertas tisu.

Sedangkan keluaran dari Raspberry Pi akan digunakan sebagai masukan ke PLC di mana keluaran PLC akan digunakan sebagai peringatan pada pekerja untuk mengecek jika terdeteksi lubang pada

kertas tisu. Apabila terdeteksi lubang pada kertas tisu, maka program akan menyalakan *buzzer*. Selain untuk mengaktifkan *buzzer*, keluaran Raspberry Pi ke PLC juga akan digunakan untuk mengaktifkan *conveyor* melalui sebuah VFD.

#### 4. Perancangan Perangkat Lunak

Pada tahap ini akan dilakukan instalasi perangkat lunak dan program yang dikehendaki pada Raspberry Pi untuk membuat sistem pendeteksi lubang pada kertas tisu. Pertama, akan dipilih *operating system* untuk Raspberry Pi, seperti NOOBS, Raspbian, Pidora, dan lain-lain. Lalu melakukan instalasi OpenCV 2.4.9., dan lain-lain di Raspberry Pi. Terakhir adalah membuat program di PC agar dapat dijalankan di Raspberry Pi dan melaksanakan pendeteksian lubang pada kertas tisu dan juga dapat berkomunikasi dengan PLC.

#### 5. Pengujian Sistem

Proses pengujian dilakukan dengan menghubungkan sistem yang terintegrasi antara kamera, Raspberry Pi, dan PLC. Setelah itu, menjalankan sistem dan melakukan proses pengujian dengan menggunakan sebuah motor sebagai simulasi penggulangan gulungan kertas tisu. Proses pengujian akan dilaksanakan secara *real-time* dengan kecepatan simulasi penggulangan yang berbeda-beda. Dari data yang diperoleh akan dipilih solusi yang terbaik untuk mengatasi permasalahan yang dihadapi.

#### 6. Penulisan Laporan Tugas Akhir

Tahap penulisan laporan Tugas Akhir dilakukan pada saat tahap pengujian sistem dimulai serta setelahnya.

### 1.5 Sistematika Penulisan

Untuk mempermudah pembahasan Tugas Akhir ini, maka laporan ini akan dibagi menjadi lima bab dengan sistematika penulisan sebagai berikut:

#### BAB 1: PENDAHULUAN

Bab ini berisi latar belakang permasalahan, perumusan masalah, tujuan, metodologi, sistematika penulisan, dan manfaat dari Tugas Akhir ini.

#### BAB 2: DASAR TEORI

Dalam bab ini akan dijelaskan mengenai teori-teori yang digunakan untuk menyelesaikan permasalahan dalam Tugas Akhir ini secara garis besar.

#### BAB 3: PERANCANGAN SISTEM

Bab ini mencakup penjabaran mengenai rancangan sistem yang telah dibuat beserta penjelasannya.

#### BAB 4: PENGUJIAN SISTEM

Dalam bab ini akan diadakan pengujian terhadap alat dengan beberapa syarat untuk mengetahui apakah alat telah bekerja dengan baik dengan melakukan beberapa analisis terhadap cara kerja alat.

#### BAB 5: PENUTUP

Bab ini mencakup kesimpulan dan saran berdasarkan hasil pengujian dan analisis yang telah dilakukan.

### **1.6 Relevansi**

Tugas Akhir ini diharapkan memberikan relevansi atau manfaat sebagai berikut:

1. Dapat menjadi referensi bagi perkembangan teknologi pengolahan citra dalam industri khususnya industri pengolahan kertas tisu.
2. Dapat menjadi referensi bagi perkembangan penggunaan Raspberry Pi di bidang industri.
3. Dapat menjadi referensi bagi orang lain yang hendak mempelajari masalah yang serupa.

## **BAB 2**

### **DASAR TEORI**

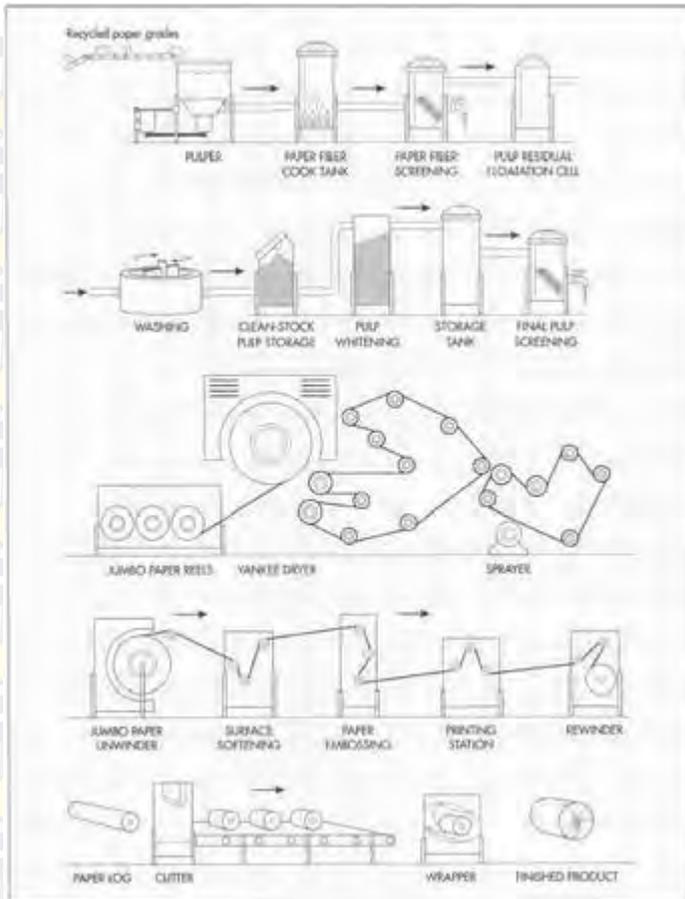
Bab ini akan membahas mengenai teori-teori yang digunakan dalam sistem yang dirancang dalam Tugas Akhir ini. Selain dari penelitian terkait yang dilakukan sebelumnya, landasan teori ini akan memaparkan tentang Kertas Tisu, Robekan, Pengendalian Kualitas, Raspberry Pi, Modul Kamera Raspberry Pi, *Programmable Logic Controller* (PLC), *Variable Frequency Driver* (VFD), Motor Induksi 3 Fasa, *Conveyor*, *Relay*, *Buzzer*, *DC Power Supply*, Pengolahan Citra, dan Komunikasi Serial RS-232.

#### **2.1 Kertas Tisu**

Kertas tisu adalah salah satu jenis kertas yang sangat tipis dan ringan. Kertas tisu yang akan dievaluasi dalam Tugas Akhir ini adalah kertas tisu wajah. Kertas tisu wajah mengacu pada kelas kertas lembut, menyerap, dan dapat dibuang yang sesuai untuk digunakan di area wajah. Produk ini merupakan alternatif dari sapu tangan yang dapat dibuang sehabis digunakan. Jenis kertas tisu ini biasanya dijual dalam kotak dan digunakan untuk membantu pengeluaran lendir dari hidung atau hal lainnya.

Kertas tisu wajah dibuat dari kertas tisu yang teringan (14-18 g/m<sup>2</sup>). Permukaannya dibuat lebih mulus dengan *calendering* ringan. Tipe kertas ini biasanya terdiri dari 1-2 lapis. Karena dituntut untuk memiliki kualitas tinggi dasar tisu ini dibuat dari bubur kertas murni, tetapi bisa juga ditambahi serat hasil daur ulang<sup>[3]</sup>. Kertas tisu bisa ditambahkan pelembut, losion, atau parfum untuk mendapatkan karakteristik yang diinginkan. Kertas tisu wajah yang telah selesai diproses akan digulung atau dilipat dan dimasukkan ke bungkus sebesar kantong atau kotak.

Kertas tisu biasa dibuat dari kertas murni, kombinasi dari kayu lunak dan kayu keras. Kayu lunak memiliki serat yang panjang dan member kekuatan pada kertas, sedangkan kayu keras memiliki serat pendek yang membuat kertas lebih lembut. Kertas tisu biasa memiliki kombinasi 70% kayu keras dan 30% kayu lunak. Material lainnya adalah air, bahan kimia, dan pemutih.



**Gambar 2.1** Proses pabrik kertas tisu

Proses pabrik dari kertas tisu pada Gambar 2.1 dapat dijabarkan sebagai berikut<sup>[4]</sup>:

1. Pohon akan tiba di pabrik dan dihilangkan kulitnya menjadi kayu.
2. Kayu akan dipotong-potong menjadi kecil agar lebih mudah diubah menjadi bubur kertas.

3. Kepingan kayu ini akan dicampur dengan bahan kimia dan dimasak untuk mengurangi kadar airnya hingga 50% menjadi bubur kertas.
4. Bubur kertas akan dicuci untuk melepaskan *lignin* dan bahan kimia.
5. Bubur kertas akan diputihkan dengan melepas warna dari seratnya. *Lignin* yang tersisa, perekat serat, akan membuat kertas menguning seiring waktu dan harus diputihkan.
6. Bubur kertas akan dicampur air dan disemprotkan pada jala-jala yang mengeringkannya dan menghasilkan lembaran serat.
7. Lembaran serat ini akan dipanaskan hingga kelembapannya berkurang hingga 5%.
8. Lalu kertas akan melalui proses *creping*, ditipiskan dengan bilah tajam dan digulung dengan kecepatan bervariasi. Tergantung tipe mesin dan jenis kertas yang digulung, kecepatannya bervariasi dari 5 m/s hingga 35 m/s.
9. Gulungan kertas ini akan dibuka kembali, dipotong, dan digulung ulang pada tabung karton panjang. Lalu kertas tisu yang digulung pada tabung karton ini akan dipotong-potong dan dikemas menjadi kertas tisu gulung yang biasa kita lihat di pasaran.

## **2.2 Robekan**

Robekan adalah aksi merusak suatu material dengan paksa, tanpa bantuan perkakas untuk memotong. Robekan pada kertas mungkin saja bersifat disengaja atau tidak disengaja. Berbeda dengan potongan, yang biasanya berbentuk lurus atau berpola yang terkontrol oleh perkakas seperti gunting, robekan berbentuk tidak rata dan tidak terencana.

Tiap material berbeda ketahanannya terhadap robekan. Beberapa material mungkin lebih tahan terhadap robekan ketika kondisinya masih utuh, tetapi ketika muncul potongan atau robekan kecil, material ini menjadi lebih mudah dirobek.

Material dapat diukur kekuatan robeknya dengan beberapa cara. Nilai dari kekuatan robek suatu material tergantung dari bentuk, kecepatan peregangan, dan suhu. Robekan yang dideteksi oleh sistem ini berbentuk lubang.

### 2.3 Pengendalian Kualitas

Definisi dari kualitas adalah faktor kunci yang dapat membawa keberhasilan, pertumbuhan, dan peningkatan posisi bersaing dalam bisnis. Kualitas dari suatu produk diartikan sebagai derajat di mana produk atau jasa tersebut mampu memuaskan keinginan dari konsumen. Kualitas menjadi faktor dasar keputusan konsumen untuk mendapatkan suatu produk, karena konsumen akan membandingkan kualitas suatu produk dengan produk-produk lain dari saingan<sup>[5]</sup>.

Pengendalian kualitas adalah aktivitas pengendalian proses untuk mengukur ciri-ciri kualitas produk, membandingkan dengan persyaratan, dan mengambil tindakan yang perlu dilakukan jika terdapat perbedaan. Tujuan dari pengendalian kualitas adalah mengatur kualitas produk atau jasa yang dapat memuaskan konsumen tetap sesuai persyaratan. Dua tahap awal pengendalian kualitas di Tugas Akhir ini akan dilakukan terhadap produk berupa kertas tisu. Salah satu persyaratan kualitas dari kertas tisu adalah bebas dari lubang. Dua tahap pengendalian kualitas akan dilakukan untuk memeriksa ciri-ciri kertas tisu yang bergulir di *conveyor* dan mengecek apakah produk sudah sesuai dengan persyaratan bebas lubang.

### 2.4 Single-Board Computer (SBC)

*Single-Board Computer* (SBC) adalah sebuah rangkaian komputer lengkap yang dibuat dalam satu papan rangkaian, sistem ini memiliki mikroprosesor, *memory*, I/O, dan fitur lainnya yang diperlukan sebuah komputer. Komputer jenis ini dirancang untuk demonstrasi, sistem perkembangan, sistem pendidikan, atau *controller*.

SBC dapat dibuat dengan meningkatkan kepadatan dari rangkaiannya. Perancangan SBC menurunkan biaya untuk membangun sebuah sistem dengan menurunkan jumlah papan rangkaian yang diperlukan, dan mengeliminasi *connector* dan *bus driver* yang biasa digunakan. Dengan menempatkan seluruh fungsi di satu papan, sistem yang lebih kecil dapat dibuat<sup>[6]</sup>.

SBC dibagi dua menurut bisa atau tidaknya SBC menyokong *slot* tambahan. SBC yang tidak menyokong *slot* tambahan biasanya sudah memiliki I/O yang dibutuhkan. Sedangkan yang menyokong *slot*

tambahan dapat menambahkan I/O tertentu yang dibutuhkan melalui *slot* tadi. SBC banyak digunakan di situasi industri di mana mereka digunakan untuk mengatur jalannya proses atau ditanam di dalam perangkat lain untuk menyediakan pengendalian dan interaksi antar muka. Karena tingginya tingkat integrasi, sedikitnya jumlah komponen dan *connector*, SBC seringkali lebih kecil, ringan, efisien, dan tahan banting dibandingkan *multi-board computer*.

Dalam perancangan sistem Tugas Akhir ini, akan digunakan SBC Raspberry Pi yang merupakan SBC yang menyokong *slot* tambahan yang digunakan untuk memasang *Random Access Memory* (RAM) dalam bentuk *SD Card*.

#### 2.4.1 Raspberry Pi

Raspberry Pi merupakan *single-board computer* (SBC) yang memiliki ukuran seperti kartu ATM dan menyokong LINUX sebagai sistem operasi utamanya. Raspberry Pi dikembangkan oleh Raspberry Pi Foundation yang berdomisili di Britania Raya. Perancangan Raspberry Pi ini awalnya dimaksudkan untuk menyediakan komputer yang murah untuk anak-anak sebagai sarana untuk mempelajari bahasa pemrograman komputer. Raspberry Pi ini diluncurkan pada 29 Februari 2012 untuk pertama kalinya. Raspberry Pi memiliki dua model utama, Model A dan Model B. Perbedaan dari kedua model ini adalah besarnya *memory* yang disokong dan ada atau tidaknya *Ethernet port*. Model A memiliki *memory* 256 MB dan tidak menyokong *Ethernet port*, sedangkan Model B memiliki *memory* 512MB dan menyokong *Ethernet port*<sup>[7]</sup>. Raspberry Pi dapat dilihat di Gambar 2.2 berikut ini.



**Gambar 2.2** Raspberry Pi Model B+

Sebenarnya Raspberry Pi berfungsi layaknya komputer *desktop* lengkap. Sistem ini menggunakan ARM *System-on-a-Chip* (SoC) yang telah dikemas dan diintegrasikan dalam sebuah papan rangkaian dengan dimensi 5,5 cm x 8,5 cm dengan ketebalan 2 cm. Karakteristiknya yang ringkas, lengkap, multi-guna, mudah digunakan, dan didukung oleh komunitas aktif yang tersebar di penjuru dunia, menjadi keunggulan Raspberry Pi yang membuatnya menjadi salah satu SBC yang paling sukses di pasaran.

Raspberry Pi Model B+ menawarkan fitur-fitur sebagai berikut<sup>[8]</sup>:

- 4 *USB 2.0 Port*
- 1 *HDMI Port*
- 1 *RCA Port* untuk keluaran video
- 1 *Audio Port* 3,5 mm
- 1 *Micro SD Card Port*
- 1 *RJ-45 Port (Ethernet LAN Port)*
- 17 GPIO

Raspberry Pi Model B+ akan digunakan sebagai sarana pengolahan citra dari produk kertas tisu dalam Tugas Akhir ini. Citra dari produk kertas tisu akan dioleh melalui program C++ dengan *library* OpenCV 2.4.9. yang dijalankan di dalam Raspberry Pi. Dengan program ini akan dilakukan pengendalian kualitas terhadap produk kertas tisu, memilah antara kertas tisu yang memenuhi persyaratan kualitas dan tidak.

#### **2.4.2 *Secure Digital (SD)***

*Secure Digital (SD)* adalah kartu *non-volatile memory* yang sangat banyak digunakan di perangkat yang mudah dipindahkan, layaknya telepon genggam, kamera digital, perangkat GPS, dan lain-lain. Standar SD dikenalkan pada tahun 2000 sebagai peningkatan dari MultiMediaCard (MMC). Dikembangkan oleh *Secure Digital Association (SDA)* yang didirikan oleh Panasonic, SanDisk, dan Toshiba. Standar ini sudah digunakan oleh ratusan merk di dunia dengan ribuan jenis produk.

SD memiliki 4 jenis *format* dengan 3 ukuran yang berbeda. 4 jenis ini, yaitu *Standard-Capacity* (SDSC), *High-Capacity* (SDHC), *Extended-Capacity* (SDXC), dan SDIO yang menyokong fungsi I/O. sedangkan 3 jenis yang dibedakan melalui ukuran, yaitu ukuran asli (32 mm x 24 mm), ukuran mini (21,5 mm x 20 mm), dan ukuran mikro (15 mm x 11 mm). Adaptor pasif dapat menyokong kartu yang berukuran lebih kecil untuk perangkat yang menggunakan kartu yang lebih besar. Ukuran SD dapat dilihat di Gambar 2.3.



**Gambar 2.3** Ukuran *Secure Digital* (SD)

Pada Tugas Akhir ini, SD akan digunakan sebagai RAM dari Raspberry Pi. SD akan digunakan sebagai tempat menyimpan sistem operasi, program-program, dan data-data yang dibutuhkan.

## 2.5 Kamera Digital

Kamera digital adalah kamera yang menyandikan citra dan video secara digital dan menyimpannya untuk pemrosesan selanjutnya. Kebanyakan kamera yang dijual sekarang adalah kamera digital, kamera ini juga banyak diterapkan di telepon genggam hingga kendaraan.

Kamera digital dan analog menggunakan sistem optik yang serupa, menggunakan lensa dengan diafragma yang bervariasi untuk memfokuskan cahaya ke penangkap citra. Diafragma dan rana kamera menangkap cahaya yang cukup ke penangkap citra selayaknya *film* tetapi secara elektronik. Kamera digital dapat menampilkan gambar

yang baru ditangkap langsung ke layar, kamera ini juga mampu menyimpan dan menghapus gambar dari *memory*-nya. Kamera digital di masa kini juga banyak yang dapat merekam video dengan suara.

Dua tipe utama sensor citra dalam kamera adalah *Charge-Coupled Device* (CCD) dan *Complementary Metal–Oxide–Semiconductor* (CMOS). Sensor CCD memiliki satu *amplifier* untuk seluruh piksel, sedangkan tiap piksel di sensor CMOS memiliki *amplifier* sendiri. Dibandingkan dengan sensor CCD, sensor CMOS menggunakan daya yang lebih kecil dan lebih cepat. Kualitas gambar yang ditangkap tergantung dari kemampuan pemrosesan citra dari kamera, daripada tipe sensornya.

Dalam Tugas Akhir ini akan digunakan modul kamera yang dikembangkan oleh Raspberry Pi Foundation sendiri khusus penggunaan dengan Raspberry Pi. Kamera ini dipilih karena menyediakan waktu pemrosesan yang lebih cepat karena dirancang khusus untuk Raspberry Pi.

### 2.5.1 Modul Kamera Raspberry Pi

Modul kamera Raspberry Pi menyokong sensor 5MP dan sambungan dengan kabel pita ke *CSI connector* di Raspberry Pi<sup>[9]</sup>. Modul kamera ini mampu menghasilkan gambar dan video dengan kualitas yang lebih baik dibandingkan dengan *webcam* yang berharga sama. Modul kamera Raspberry Pi dirilis ke pasar pada tanggal 14 Mei 2013<sup>[10]</sup> dan sudah terjual lebih dari sepuluh ribu buah.

Berikut ini adalah spesifikasi teknik dari Modul Kamera Raspberry Pi<sup>[11]</sup>:

- Tipe sensor: OmniVision OV5647 Color CMOS QSXGA (5MP)
- Ukuran sensor: 3,67 mm x 2,74 mm
- Jumlah piksel: 2592 x 1944
- Ukuran piksel: 1,4  $\mu\text{m}$  x 1,4  $\mu\text{m}$
- Lensa:  $f=3,6$  mm,  $f/2,9$
- Sudut pandang:  $54^\circ$  x  $41^\circ$
- Bidang pandang: 2 m x 1.33 m di 2 m
- Fokus tetap: 1m -  $\infty$
- Video: 1080p di 30 fps dengan *codec* H.264 (AVC)
- Ukuran: 25 mm x 24 mm

## 2.6 *Programmable Logic Controller (PLC)*

*Programmable Logic Controller (PLC)* adalah komputer digital yang digunakan dalam otomatisasi dari proses elektromekanik pada industri. PLC banyak digunakan di industri dan mesin. PLC dirancang untuk banyak masukan dan keluaran, kisaran suhu yang lebih luas, ketahanan terhadap gangguan elektrik, dan ketahanan terhadap getaran dan benturan. Program yang digunakan untuk mengatur operasi mesin biasanya disimpan dalam *Non-Volatile Memory (NVM)*. PLC merupakan contoh dari sistem *real-time* di mana keluaran harus diproduksi sebagai respon dari kondisi masukan dalam jangka waktu yang terbatas. Jika tidak terpenuhi, maka operasi yang tidak diinginkan akan terjadi.

Sebelum adanya PLC proses di industri banyak diatur menggunakan sistem *relay*, tetapi sistem ini sangat susah diperbarui karena sangat mahal dan menghabiskan waktu, setiap *relay*-nya harus dihubungkan ulang sesuai karakteristik operasionalnya. Karena kelebihanannya yang sangat menguntungkan maka PLC lebih banyak digunakan daripada sistem *relay* yang lebih tua. PLC berukuran fisik lebih kecil daripada sistem *relay*, tidak diperlukannya pengubahan *hardware* jika ingin menerapkan perubahan kecil terhadap suatu proses dan hanya perlu mengubah programnya.

Perbedaannya dari komputer lain adalah PLC memiliki perlindungan untuk kondisi kerja yang lebih berat (seperti debu, kelembaban, panas, dingin) dan memiliki masukan dan keluaran yang dapat diperbanyak. Masukan dan keluaran dari PLC dapat dihubungkan ke sensor dan aktuator. Program PLC biasanya dijalankan selama sistem yang dikontrol tetap berjalan. Status dari masukan disalin ke *memory* yang dapat diakses prosesor. Lalu program dijalankan dari instruksi di jenjang pertama hingga jenjang terakhir. Proses ini akan memakan waktu tertentu untuk mengevaluasi seluruh jenjang dan memperbarui keluarannya. Waktu untuk melaksanakan program ini dapat memakan waktu sebesar beberapa milisekon untuk program yang kecil atau prosesor yang cepat. Jika waktu untuk melaksanakan program terlalu lama, respon dari PLC untuk proses akan terlalu lambat dan tidak berguna<sup>[12]</sup>.

Umumnya ada dua jenis PLC berdasarkan terpisahnya modul-modulnya (I/O, CPU, dan lain-lain). Jika modulnya terpisah-pisah, maka dinamakan PLC modular. Sedangkan jika semua modul tergabung menjadi satu, maka dinamakan PLC *compact*. Beda kedua PLC ini terletak di jumlah I/O yang dapat dimasukkan. Pada tipe modular jika menginginkan I/O yang lebih banyak, analog maupun digital, maka bisa ditambahkan melalui modul. Sedangkan tipe *compact*, I/O hanya bisa mengerti sinyal digital dan terbatas, jika ingin I/O analog, maka diperlukan PLC tambahan.

PLC biasanya memiliki sambungan untuk komunikasi, seperti RS-232 atau Ethernet. Kebanyakan PLC masa kini dapat berkomunikasi melalui sebuah jaringan dengan sistem lain, seperti komputer yang menjalankan SCADA. Jaringan komunikasi ini juga digunakan untuk memprogram sebuah PLC. Program PLC biasanya ditulis di sebuah aplikasi khusus di komputer biasa dan diunggah ke PLC melalui komunikasi tadi. Program akan disimpan di *memory* PLC. Ada beberapa Bahasa pemrograman yang dapat digunakan untuk memprogram PLC, tetapi kita menggunakan *ladder diagram* (LD) di Tugas Akhir ini.

Pada sistem ini akan digunakan PLC OMRON CPM2A-20CDR-A.

### **2.6.1 PLC OMRON CPM2A-20CDR-A**

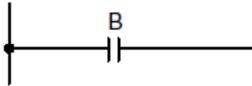
PLC OMRON CPM2A-20CDR-A termasuk jenis PLC *compact* karena memiliki modul I/O, CPU, dan lain-lain yang tergabung menjadi satu dalam sebuah modul. PLC ini memiliki 12 masukan digital dan 8 keluaran digital dan perlu dipasok dengan sinyal AC. Keluaran yang digunakan oleh PLC ini berjenis *relay output*. PLC tipe ini diprogram dengan bahasa pemrograman *ladder diagram* yang akan diunggah ke PLC melalui PC<sup>[13][28]</sup>.

### **2.6.2 Ladder Diagram**

*Ladder diagram* merupakan bahasa pemrograman yang merepresentasikan sebuah program dengan diagram grafik berdasarkan *relay logic*. Bahasa pemrograman ini banyak digunakan oleh PLC sebagai aplikasi kontrol industri. Instruksi-intruksi dasar *ladder diagram* yang akan digunakan di sistem akan dijelaskan lebih lanjut di bawah<sup>[13]</sup>.

1. *LOAD* (LD)

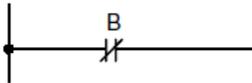
Perintah ini digunakan jika suatu sistem kontrol hanya membutuhkan satu logika untuk bekerja. Perintah ini serupa dengan kontak NO *relay*. Simbol perintah ini dapat dilihat di Gambar 2.4.



**Gambar 2.4** Perintah *LOAD* ladder diagram

2. *LOAD NOT*

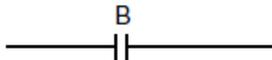
Perintah ini digunakan jika suatu sistem kontrol hanya membutuhkan satu logika untuk bekerja. Perintah ini serupa dengan kontak NC *relay*. Simbol perintah ini dapat dilihat di Gambar 2.5.



**Gambar 2.5** Perintah *LOAD NOT* ladder diagram

3. *AND*

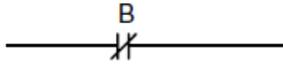
Perintah ini digunakan jika suatu sistem kontrol membutuhkan lebih dari satu logika untuk bekerja. Perintah ini serupa dengan kontak NO *relay*. Simbol perintah ini dapat dilihat di Gambar 2.6.



**Gambar 2.6** Perintah *AND* ladder diagram

4. *AND NOT*

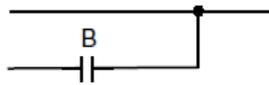
Perintah ini digunakan jika suatu sistem kontrol membutuhkan lebih dari satu logika untuk bekerja. Perintah ini serupa dengan kontak NC *relay*. Simbol perintah ini dapat dilihat di Gambar 2.7.



**Gambar 2.7** Perintah *AND NOT* ladder diagram

5. *OR*

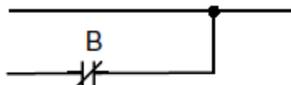
Perintah ini digunakan jika suatu sistem kontrol membutuhkan satu atau lebih dari satu logika untuk bekerja. Perintah ini serupa dengan kontak NO *relay*. Simbol perintah ini dapat dilihat di Gambar 2.8.



**Gambar 2.8** Perintah *OR* ladder diagram

6. *OR NOT*

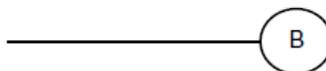
Perintah ini digunakan jika suatu sistem kontrol membutuhkan satu atau lebih dari satu logika untuk bekerja. Perintah ini serupa dengan kontak NC *relay*. Simbol perintah ini dapat dilihat di Gambar 2.9.



**Gambar 2.9** Perintah *OR NOT* ladder diagram

7. *OUT*

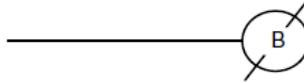
Perintah ini digunakan untuk mengeluarkan suatu keluaran jika kondisi logikanya terpenuhi. Perintah ini serupa dengan kontak NO *relay*. Simbol perintah ini dapat dilihat di Gambar 2.10.



**Gambar 2.10** Perintah *OUT* ladder diagram

#### 8. *OUT NOT*

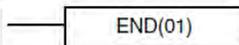
Perintah ini digunakan untuk mengeluarkan suatu keluaran jika kondisi logikanya terpenuhi. Perintah ini serupa dengan kontak NC *relay*. Simbol perintah ini dapat dilihat di Gambar 2.11.



**Gambar 2.11** Perintah *OUT NOT* ladder diagram

#### 9. *END*

Perintah ini digunakan untuk menutup program. Simbo perintah ini dapat dilihat di Gambar 2.12.



**Gambar 2.12** Perintah *END* ladder diagram

### 2.7 *Variable Frequency Drive (VFD)*

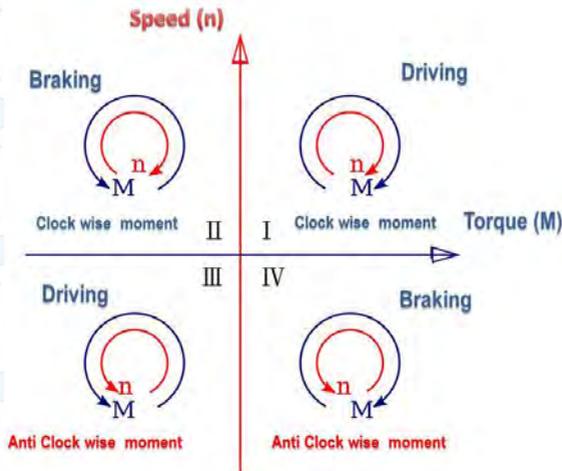
*Variable Frequency Drive (VFD)* adalah istilah untuk penggerak kecepatan elektromekanik yang digunakan untuk mengendalikan kecepatan dan torsi motor AC dengan mengubah-ubah frekuensi dan tegangan dari masukan motor<sup>[14]</sup>. VFD digunakan secara luas mulai dari perangkat kecil hingga bor-bor di pertambangan. Sebuah VFD adalah sebuah sistem penggerak yang terdiri dari 3 sistem yang lebih kecil, yaitu motor AC, pengendali gerak utama, dan operator antarmuka penggerak.

Sistem VFD banyak menggunakan motor induksi 3 fasa sebagai motor AC-nya. Beberapa tipe motor 1 fasa dapat digunakan, tetapi motor 3 fasa lebih diutamakan. Motor induksi 3 fasa lebih sesuai untuk sistem dan lebih ekonomis.

Pengendali VFD adalah sistem pengubah daya *solid-state* yang terdiri dari *rectifier bridge converter*, sambungan DC, dan *inverter*. Kebanyakan dari VFD adalah penggerak AC-AC yang membutuhkan masukan AC dan mengeluarkan keluaran AC juga, tetapi ada juga VFD berjenis DC-AC. *Rectifier converter* yang paling umum digunakan di

penggerak *Voltage-Source Inverter (VSI)* adalah *full-wave diode bridge* 6-pulsa 3-fasa Sambungan DC-nya terdiri dari kapasitor untuk memuluskan keluaran DC dari *converter* dan memberikan masukan yang tetap ke *inverter*. Tegangan DC yang telah disaring ini akan diubah ke tegangan AC *quasi-sinusoidal* dengan melalui *switching inverter*.

Operator antarmuka menyediakan operator untuk menjalankan, memberhentikan, dan mengatur kecepatan motor. Fungsi kendali tambahan dalam VSD adalah membalik arah putaran motor, mengubah beberapa tingkat kecepatan yang sudah ditetapkan, dan menyediakan kendali otomatis dari sinyal kontrol proses eksternal. Operator antarmuka ini bisa menggunakan layar *alphanumeric*, lampu indikasi, dan pengukur untuk memberikan informasi mengenai operasi gerak. Seringkali *keypad* dan layar dipasang di depan VFD. Banyak juga VFD yang menyediakan I/O untuk tombol, saklar, dan sinyal kontrol eksternal lainnya. Beberapa VFD juga menyediakan *port* komunikasi serial agar VFD dapat dikendalikan melalui komputer.



**Gambar 2.13** Grafik hubungan kecepatan dan torsi motor listrik

Melihat dari Gambar 2.13 sistem penggerak dapat dibedakan menjadi empat kuadran, yaitu:

- Kuadran I: Menggerakkan ke arah depan dengan kecepatan dan torsi positif.
- Kuadran II: Mengerem ketika arah gerak ke depan dengan kecepatan positif dan torsi negatif.
- Kuadran III: Menggerakkan ke arah belakang dengan kecepatan negatif dan torsi positif.
- Kuadran IV: Mengerem ketika arah gerak ke belakang dengan kecepatan negatif dan torsi negatif.

VFD menggerakkan motor dengan memasok motor dengan frekuensi dan tegangan rendah dan meningkatkannya hingga sesuai dengan kecepatan yang diinginkan, hal ini berlaku sebaliknya ketika mengerem.

Pada Tugas Akhir ini akan digunakan VFD Fuji Electric FRENIC-Mini yang merupakan VFD berjenis VSI untuk mengendalikan kecepatan motor induksi 3 fasa yang digunakan untuk menggerakkan *conveyor*. VFD ini akan dikendalikan melalui sinyal kontrol eksternal dari PLC.

### **2.7.1 VFD Fuji Electric FRENIC-Mini FRN075CIS-7A**

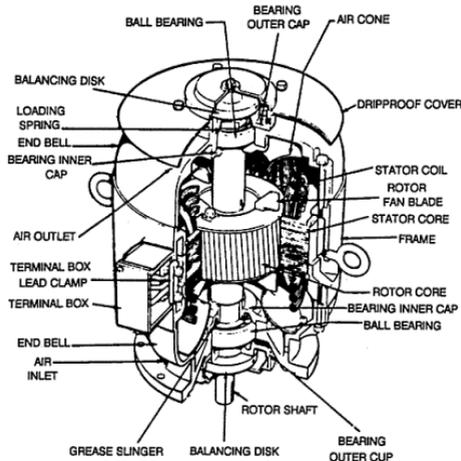
Fuji Electric merupakan salah satu penyedia VFD utama di dunia untuk VFD dengan kelas 5HP dan di bawahnya. Seri FRENIC-Mini merupakan salah satu VFD terbaik dalam hal performa untuk aplikasi 1 fasa dan 3 fasa. VFD ini merupakan perangkat yang ringkas dan menggabungkan operasi yang mudah dengan performa industri yang baik<sup>[15]</sup>.

FRENIC-Mini merupakan VFD yang sesuai untuk digunakan dengan *conveyor*. VFD ini memiliki torsi awal yang tinggi, stabil pada kecepatan rendah, dan operasi stabil meskipun ada fluktuasi beban. VFD ini juga menyediakan mode hemat energi otomatis dan kipas pendingin. Ringkas dan sesuai dengan banyak frekuensi merupakan salah satu fitur utama dari FRENIC-Mini.

## **2.8 Motor Induksi 3 Fasa**

Motor induksi atau motor tidak sinkron merupakan motor listrik AC yang membutuhkan arus listrik di rotornya untuk menghasilkan torsi dengan induksi elektromagnetik dari medan magnet

di kumparan stator. Motor induksi tidak membutuhkan komutasi mekanik dan eksitasi seperti motor DC dan motor sinkron. Motor induksi bisa berjenis kumparan atau sangkar. Ilustrasi motor ini dapat dilihat di Gambar 2.14.



**Gambar 2.14** Ilustrasi motor induksi 3 fasa

Motor induksi 3 fasa berjenis sangkar banyak digunakan sebagai penggerak di industri karena sifatnya yang tahan banting, dapat diandalkan, dan ekonomis. Sedangkan motor induksi 1 fasa banyak digunakan di perangkat berbeban kecil seperti kipas angin. Meskipun banyak digunakan untuk kecepatan tetap, sekarang motor ini banyak dipasangkan dengan VFD untuk penggunaan dengan kecepatan yang bervariasi.

Medan putar stator akan memotong batang konduktor rotor dan menimbulkan GGL induksi dan menghasilkan arus. Arus di dalam medan magnet akan menimbulkan gaya pada rotor untuk menggerakkan beban dan berputar searah dengan medan putar stator.

## 2.9 Sistem Conveyor

Sistem *conveyor* merupakan perangkat mekanik yang umum digunakan untuk memindahkan materi dari satu tempat ke tempat

lainnya. *Conveyor* sangat berguna saat diterapkan untuk memindahkan materi yang besar dan berat. Sistem ini memudahkan perpindahan bermacam-macam materi secara mudah dan cepat, sehingga sangat banyak digunakan di industri.

Berikut ini beberapa keuntungan dari sistem *conveyor*:

- Sistem ini dapat memindahkan materi dengan aman dari satu tempat ke tempat lainnya. Jika proses ini dilakukan oleh tenaga kerja manusia, maka proses menjadi berat dan mahal.
- Sistem ini dapat dipasang hampir di mana saja dan lebih aman dibandingkan menggunakan alat lainnya untuk memindahkan materi.
- Sistem ini dapat memindahkan beban dengan segala bentuk, ukuran, dan berat.
- Ada beberapa pilihan untuk menggerakkan sistem ini, seperti hidrolik, mekanik, hingga sistem terotomatisasi.

Tugas Akhir ini menggunakan *belt conveyor* untuk mensimulasikan proses penggulangan kertas tisu secara berulang kali. *Conveyor* jenis ini menggunakan sabuk yang terpasang di dua katrol atau lebih, sehingga dapat menghasilkan putaran terus-menerus. Salah satu atau lebih katrol ini akan diberi daya gerak, menggerakkan sabuk dan material di atas sabuknya.

Sabuk dari sistem ini biasanya terdiri dari beberapa lapis materi. Banyak sabuk terdiri dari dua lapis materi. Lapisan bawah biasanya terdiri dari kain tenun seperti poliester, nilon, atau katun dan berfungsi untuk menghantarkan daya yang merata. Sedangkan lapisan atasnya biasa menggunakan berbagai macam karet atau plastik sesuai penggunaannya.

Sabuk karet paling umum digunakan, biasanya sabuk ini disimpulkan di dua rol yang salah satunya diberi daya gerak, sabuk ini biasa bergerak menggelincir di rangka metal atau rol-rol lain. Rol lebih efisien karena meminimalkan daya gesek karena berat beban. Sistem *conveyor* jenis ini juga paling serba guna dan murah.

## **2.10 Relay**

*Relay* merupakan saklar yang dioperasikan secara elektrik. Kebanyakan relay menggunakan elektromagnet untuk menggerakkan

saklarnya, tetapi ada juga yang berjenis *solid-state*. *Relay* dibutuhkan jika ada rangkaian yang membutuhkan sinyal yang berbeda dengan yang disediakan. Berikut ini *relay* yang digunakan oleh sistem di Gambar 2.15.



**Gambar 2.15** *Relay* HRS4H-S-DC24V

*Relay* elektromagnetik sederhana memiliki gulungan kawat membalut inti besi, kuk besi yang menyediakan jalur rendah reluktansi untuk fluks magnet, lengan yang dapat bergerak, dan satu atau lebih kontak. Lengan ini terhubung dengan pegas yang membuat celah udara ketika tidak aliran listrik di gulungan.

Ketika arus listrik melewati gulungan, maka gulungan akan menghasilkan medan magnet dan mengaktifkan lengannya dan dapat menyambung atau memutus sambungan kontak. Lalu jika arus listrik berhenti, pegas akan membalik kembali sambungan kontak tadi. Ketika gulungan dipasok dengan arus DC, dioda dipasang paralel dengan gulungan untuk mendisipasi daya dari medan magnet yang gagal ketika deaktivasi, di mana hal ini dapat memberikan lonjakan tegangan yang berbahaya untuk komponen semikonduktor<sup>[16]</sup>.

Ada empat jenis relay dilihat dari jumlah kontakannya, yaitu:

- *Single Pole Single Throw* (SPST): Memiliki dua terminal yang bisa tersambung atau tidak. Bisa dalam bentuk *Normally Open* (NO) *Normally Closed* (NC).
- *Single Pole Double Throw* (SPDT): Sebuah terminal tersambung dengan salah satu dari dua terminal.
- *Double Pole Single Throw* (DPST): Seperti dua SPST yang tergabung menjadi satu tetapi diaktifkan oleh satu gulungan.
- *Double Pole Double Throw* (DPDT): Seperti dua SPDT yang tergabung menjadi satu tetapi diaktifkan oleh satu gulungan.

Pada Tugas Akhir ini akan digunakan *relay* SPDT untuk mengatur jalannya *buzzer*.

### 2.11 *Buzzer*

*Buzzer* merupakan perangkat sinyal audio yang dapat bekerja secara mekanik, elektromekanik, atau pizoelektrik. Perangkat ini biasa digunakan sebagai alarm atau konfirmasi masukan. Kebanyakan *buzzer* di masa kini sudah menggunakan pizoelektrik. Berikut ini *buzzer* yang digunakan oleh sistem di Gambar 2.16.



**Gambar 2.16** *Buzzer* SFM-27

Pengeras suara pizoelektrik memiliki keuntungan, yaitu tahan terhadap kelebihan beban. Ada juga beberapa kerugiannya, yaitu: dapat terjadi osilasi di beberapa *amplifier* ketika diberi beban kapasitif yang menyebabkan distorsi dan kerusakan di *amplifier*. Pizoelektrik juga lemah dalam bidang respon frekuensi, dan hanya digunakan sebagai dalam satu frekuensi saja seperti *buzzer*.

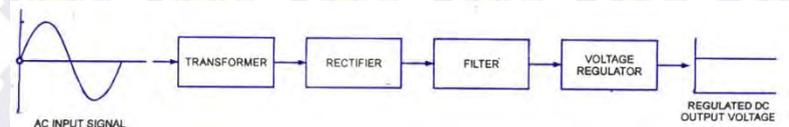
Pengeras suara pizoelektrik menggunakan kristal pizoelektrik yang disambungkan ke diafragma mekanik. Sinyal audio akan diberikan ke kristal, yang bereaksi dengan melebar seiring besarnya tegangan yang diberikan, dan mengubah energi elektrik menjadi energi mekanik. Elemen aktif ini merupakan inti dari transduser ini yang mengubah energi elektrik ke energi akustik. Elemen aktif ini merupakan materi yang terpolarisasi oleh elektroda yang dipasang di kedua sisi elemen. Ketika ada medan listrik di materi, molekul yang terpolarisasi akan mensejajarkan diri sesuai medan listrik, menghasilkan dipol yang terinduksi di dalam struktur kristal materi tersebut. Pensejajaran molekul ini menyebabkan materi berubah dimensi.

## 2.12 Sumber Daya DC

Sumber daya DC adalah suatu perangkat yang memasok tegangan dengan polaritas tetap (positif atau negatif) ke beban. Sumber daya DC dapat memiliki masukan dari sumber DC atau sumber AC.

Beberapa sumber daya DC menggunakan sumber AC sebagai masukannya. Beberapa sumber daya menggunakan transformator untuk mengubah tegangan masukan ke tegangan AC yang lebih rendah atau lebih tinggi. *Rectifier* atau penyearah digunakan untuk mengubah tegangan AC tadi menjadi tegangan DC yang berubah-ubah, lalu dilewatkan ke *filter* yang mengubahnya menjadi tegangan DC yang tak teratur.

Tegangan DC yang tak teratur tadi akan dilewatkan ke regulator tegangan linear untuk mengubahnya ke tegangan DC tetap yang diinginkan. Regulator ini juga dapat membatasi arus untuk melindungi sumber daya dan beban dari kelebihan arus. Blok diagram selengkapnyanya dapat dilihat di Gambar 2.17 berikut ini.



**Gambar 2.17** Blok diagram sumber daya DC

Tegangan keluaran yang tetap banyak digunakan di berbagai aplikasi elektronik. Kita akan menggunakan sumber daya DC ini untuk menyediakan tegangan DC yang tetap sebagai sinyal kontrol untuk VFD dan *buzzer*.

## 2.13 Pengolahan Citra

Pengolahan citra adalah berbagai bentuk pemrosesan sinyal dengan masukan berupa citra, seperti foto atau video. Keluaran dari pengolahan citra bisa berupa citra juga atau karakteristik dan parameter yang terkait dengan citra masukan. Kebanyakan teknik pengolahan citra menafsirkan citra sebagai sinyal dua dimensi dan menerapkan teknik pemrosesan sinyal umum. Pengolahan citra akan dilakukan untuk mendeteksi ada atau tidaknya lubang pada produk kertas tisu.

Pengolahan citra akan dilakukan melalui program di Raspberry Pi yang ditulis dengan bahasa C++ dengan *library Open Source Computer Vision* (OpenCV) 2.4.9.

### 2.13.1 **Open Source Computer Vision (OpenCV)**

*Open Source Computer Vision* (OpenCV) adalah *library* fungsi pemrograman yang khusus mengakomodasi *computer vision*, dikembangkan oleh pusat penelitian Intel Rusia di Nizhny Novgorod, dan didukung oleh Willow Garage dan Itseez. OpenCV bersifat gratis bebas digunakan siapa saja. *Library* ini dapat digunakan di banyak bahasa pemrograman dan berfokus pada pengolahan citra secara *real-time*.

### 2.13.2 **Grayscale**

Citra *grayscale* adalah citra di mana nilai dari tiap pikselnya adalah *sample* tunggal dan hanya berisi informasi intensitas. Citra jenis ini terdiri dari berbagai tingkat warna abu-abu, mulai dari hitam yang paling lemah intensitasnya sampai putih yang terkuat. Hasil proses ini dapat dilihat di Gambar 2.18.



**Gambar 2.18** Contoh proses *grayscale*<sup>[17]</sup>

Citra berwarna terdiri dari tiga lapisan warna berupa *red*, *green*, dan *blue*. Jika citra tidak diubah ke bentuk *grayscale* maka pemrosesan selanjutnya harus dilakukan terhadap ketiga lapisan tersebut. Maka dengan proses *grayscale* ketiga lapisan itu dijadikan satu lapisan yang berwarna abu-abu saja.

Suatu citra *grayscale* dapat direpresentasikan sebagai *array* dua dimensi di mana tiap pikselnya menunjukkan intensitas dari citra pada koordinat yang sesuai. Suatu citra 8 bit biasa memiliki 256 nilai intensitas ( $0^{\circ}$ - $255^{\circ}$ ), di mana  $0^{\circ}$  merepresentasikan intensitas paling gelap dan  $255^{\circ}$  paling terang.

Biasanya proses ini dilakukan dengan mengambil rata-rata dari nilai lapisan *red*, *green*, dan *blue*. Tetapi di OpenCV mereka menggunakan rumus yang berbeda dengan proses *grayscale* normal, yaitu<sup>[17]</sup>:

$$Y = 0,299R + 0,587G + 0,114B$$

Keterangan:

Y = nilai *grayscale*

R = nilai lapisan *red*

G = nilai lapisan *green*

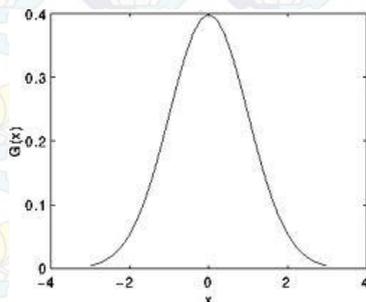
B = nilai lapisan *blue*

### 2.13.3 Penghalusan

*Smoothing* merupakan merupakan proses yang sederhana dan banyak digunakan di pengolahan citra. *Smoothing* banyak dilakukan untuk mengurangi *noise* pada suatu citra. Operasi ini dilakukan dengan melewati citra ke suatu filter.

Ada beberapa jenis filter yang digunakan di dalam proses ini, tetapi Tugas Akhir ini menggunakan Filter Gaussian untuk melakukan proses *smoothing* terhadap citra yang sudah berbentuk *grayscale*. Proses ini akan menghilangkan *noise* kecil pada citra dan membuatnya lebih kabur. Filter ini bekerja dengan menggabungkan tiap titik di *array* masukan dengan *kernel* Gaussian dan menjumlahkannya untuk menghasilkan *array* keluaran<sup>[18]</sup>.

Berikut ini merupakan *kernel* Gaussian 1D yang direpresentasikan sebagai grafik di Gambar 2.19.



**Gambar 2.19** Grafik *kernel* Gaussian 1 dimensi<sup>[18]</sup>

Jika sebuah citra didefinisikan sebagai citra 1 dimensi, maka piksel yang terletak di tengah grafik memiliki bobot terbesar. Bobot dari piksel tetangga akan berkurang seiring meningkatnya jarak spasial antara mereka dan piksel yang di tengah. Sedangkan Gaussian 2D dapat direpresentasikan sebagai di rumus berikut<sup>[18]</sup>.

$$G_0(x,y) = A e^{-\frac{(x - \mu_x)^2}{2 \sigma_x^2} - \frac{(y - \mu_y)^2}{2 \sigma_y^2}}$$

Keterangan:

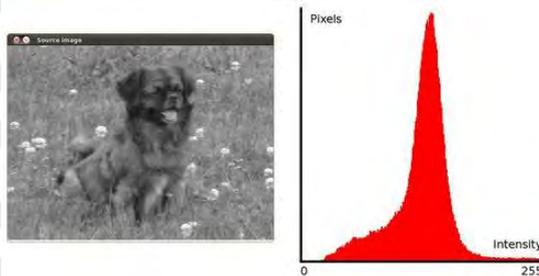
A = tinggi puncak

$\mu$  = posisi puncak

= variasi dari tiap variabel x dan y

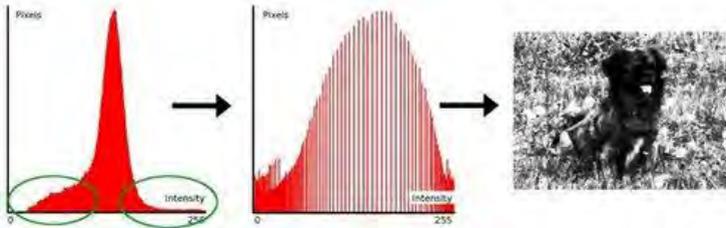
#### 2.13.4 Penyetaraan Histogram

Histogram adalah representasi grafik dari distribusi intensitas piksel dari sebuah citra. Histogram menghitung jumlah piksel yang jatuh dalam setiap nilai intensitas yang ada dalam sebuah citra.



**Gambar 2.20** Contoh histogram suatu citra<sup>[19]</sup>

Penyetaraan histogram adalah sebuah metode untuk meningkatkan kontras dari suatu citra, untuk meningkatkan jarak antar intensitasnya. Jika kita melihat Gambar 2.20, maka bisa dilihat bahwa banyak piksel yang jatuh di daerah tengah dari banyaknya nilai intensitas yang ada. Metode ini akan meregangkan daerah ini seperti Gambar 2.21 di bawah ini.



**Gambar 2.21** Contoh proses penyetaraan histogram dan hasilnya<sup>[19]</sup>

Penyetaraan dilakukan dengan memetakan ulang distribusi satu (histogram awal) ke distribusi lain (histogram yang lebih lebar dan merata nilai intensitasnya) sehingga nilai intensitasnya lebih merata dengan rumus distribusi konvolusi dan rumus pemetaan di bawah<sup>[19]</sup>.

$$H(i) = \sum_{0 \leq j < i} H(j)$$

Keterangan:

$H'(i)$  = distribusi kumulatif

$$\text{equalized}(x, y) = H(\text{src}(x, y))$$

Keterangan:

equalized = gambar yang sudah disetarakan histogramnya

src = citra yang akan disetarakan histogramnya

Metode ini dilakukan untuk menghilangkan *noise* yang masih ada setelah dilakukan proses penghalusan citra dan meningkatkan kontras citra sehingga pendeteksian lubang di kertas tisu akan lebih akurat.

### 2.13.5 Pendeteksian Tepian Canny

Detektor tepian Canny dikembangkan oleh John Canny di tahun 1986. Detektor ini juga disebut detektor optimal, algoritma Canny ini memiliki sifat error yang rendah, lokalisasi yang bagus, dan respon minimal. Contoh pemrosesan ini dapat dilihat di Gambar 2.22.



**Gambar 2.22** Contoh hasil proses pendeteksian tepian Canny<sup>[20]</sup>

Pendeteksi tepian Canny adalah proses dengan empat langkah sebagai berikut<sup>[20][21]</sup>:

1. Citra akan kembali melewati filter Gaussian dengan *kernel* di bawah ini untuk mengurangi *noise*.

$$K = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Keterangan:  
K = *kernel*

2. Proses ini mencari gradien intensitas dari citra dengan proses serupa dengan Sobel. Pertama akan dikonvolusikan masker di bawah ini dalam arah x dan y.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Lalu magnitudo gradien dan arah gradien dari tiap piksel akan dihitung dengan rumus berikut. Arahnya akan dibulatkan ke 4 sudut ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ).

$$G = \sqrt{G_x^2 + G_y^2}$$
$$= \tan^{-1} \frac{G_y}{G_x}$$

Keterangan:

$G$  = magnitudo gradien piksel  
= arah gradien piksel

3. Supresi non-maksimal menentukan piksel mana yang lebih sesuai sebagai tepian jika dibandingkan dengan piksel tetangganya. Jika magnitudo gradien lebih besar dari dua piksel tetangganya dalam arah gradien yang sama, maka piksel akan diidentifikasi sebagai tepian. Jika tidak, maka piksel akan diidentifikasi sebagai latar belakang dan diberi nilai  $0^\circ$ .
4. Pembatasan (*thresholding*) histerisis akan menerapkan dua nilai batas (*threshold*) dalam satuan derajat ( $^\circ$ ) yang akan mengeliminasi tepian yang lemah dengan ketentuan sebagai berikut.
  - a. Jika nilai gradien suatu piksel berada di atas batas atas, piksel akan terdeteksi sebagai tepian.
  - b. Jika nilai gradien suatu piksel berada di bawah batas bawah, piksel tidak terdeteksi sebagai tepian.
  - c. Jika nilai gradien suatu piksel berada di antara dua nilai batasan, maka piksel tersebut akan dideteksi sebagai bagian dari tepian jika menempel dengan piksel lain yang nilainya berada di atas batas atas.

### 2.13.6 Operasi Morfologi

Operasi morfologi ini terdiri dari dua operasi utama, yaitu *dilation* dan *erosion*. *Dilation* dilakukan dengan membuat suatu *kernel* dengan bentuk apapun dengan inti di tengahnya. Lalu *kernel* ini dipindaikan ke citra dan menghitung nilai maksimum piksel yang

terkena *kernel* dan mengganti nilai piksel yang berada di inti *kernel* dengan nilai maksimum tersebut. Sedangkan proses *erosion* merupakan proses yang serupa dengan *dilation* tetapi yang dicari adalah nilai minimumnya<sup>[22]</sup>. Kedua proses ini memiliki rumus di bawah ini.

$$dst_d(x, y) = \max_{(x, y): \text{elemen}(x, y) \neq 0} src(x + x, y + y)$$

$$dst_e(x, y) = \min_{(x, y): \text{elemen}(x, y) \neq 0} src(x + x, y + y)$$

Keterangan:

$dst_d$  = citra hasil operasi *dilation*

$dst_e$  = citra hasil operasi *erosion*

$src$  = citra awal

Contoh kedua proses ini dapat dilihat di Gambar 2.23 dan Gambar 2.24.



**Gambar 2.23** Contoh proses *dilation*<sup>[22]</sup>



**Gambar 2.24** Contoh proses *erosion*<sup>[22]</sup>

Penggabungan dari kedua operasi utama ini menghasilkan operasi *opening* dan *closing*. *Opening* dapat didapat dengan melakukan *erosion* terlebih dahulu diikuti dengan *dilation*. Operasi ini berguna untuk menghapus objek kecil (daerah terang) seperti *noise*. Sedangkan *closing* merupakan kebalikan dari *opening*. Operasi ini berguna untuk menutup lubang-lubang kecil (daerah gelap) di citra<sup>[23]</sup>. Contoh kedua proses ini dapat dilihat di Gambar 2.25 dan Gambar 2.26.



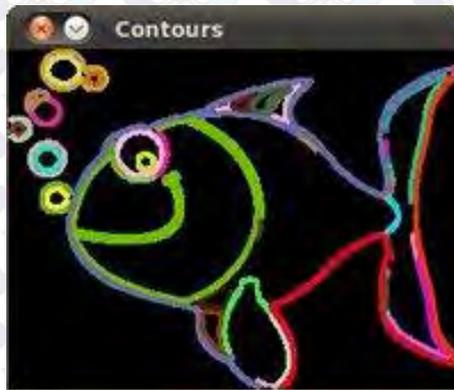
**Gambar 2.25** Contoh proses *opening*<sup>[23]</sup>

2.13

**Gambar 2.26** Contoh proses *closing*<sup>[23]</sup>

### 2.13.7 Penemuan Kontur

Fungsi di OpenCV ini menemukannya kontur pada suatu citra biner dengan algoritma dari S. Suzuki dan K. Abe. Fungsi hanya dapat digunakan di citra biner dan mengaplikasikan algoritma yang ada. Tiap kontur yang ditemukan akan disimpan menjadi vektor poin-poin<sup>[24][25]</sup>. Contoh penggunaan fungsi ini dapat dilihat di Gambar 2.27 di mana kontur yang ditemukan ditandai warna.



**Gambar 2.27** Contoh hasil proses penemuan kontur<sup>[24]</sup>

### 2.14 Komunikasi Serial RS-232

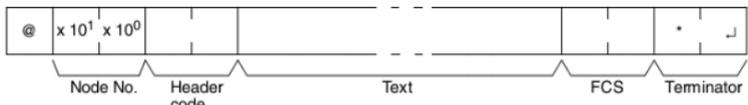
Dalam telekomunikasi, RS-232 adalah standar transmisi data melalui komunikasi serial. Standar ini mendefinisikan sinyal yang menghubungkan *Data Terminal Equipment* (DTE) seperti terminal komputer, dan sebuah *Data-Circuit Terminating Equipment* (DCE) seperti *modem*. Standar RS-232 banyak digunakan di *port* serial komputer. Standar ini mendefinisikan karakteristik elektrik, *timing* dari sinyal, arti sinyal, ukuran pin sambungan<sup>[26]</sup>.

RS-232 dulu banyak digunakan di PC dan perangkat lainnya. Tetapi RS-232 dibatasi oleh kecepatannya yang rendah, tegangan yang besar, dan sambungan yang besar. PC di jaman sekarang sudah menggunakan USB sebagai ganti dari RS-232. Banyak komputer sekarang memerlukan adaptor USB ke RS-232 jika ingin menggunakan komunikasi serial ini. Tetapi RS-232 masih banyak digunakan di industri, jaringan, dan perangkat sains.

### 2.14.1 Command Block dan Response Block PLC

PLC dapat mengirim dan menerima data dari PC dengan komunikasi *hostlink*, biasanya PLC menerima perintah untuk menulis atau membaca alamat di dalamnya. Terdapat aturan-aturan tersendiri dalam penulisan dan pembacaan data dari PLC OMRON CPM2A-20CDR-A ini. Aturan-aturan ini diatur dengan *format command block* dan *response block*. *Command block* merupakan keadaan ketika PC menjadi *host* dan mengirimkan data ke PLC *Format* dari *command block* dan *response block* akan dibahas lebih lanjut di bawah ini.

#### 1. Command Block

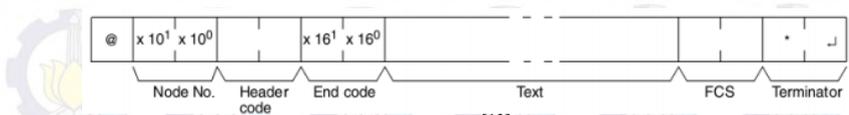


**Gambar 2.28** Format command block<sup>[13]</sup>

Keterangan:

- @: Kode awal yang harus diberikan.
- Node No.: Nomor identitas PLC.
- Header code: Menunjukkan operasi (*read* atau *write*) di area tertentu.
- Text: *Read*: word awal yang dituju dan jumlah word yang akan dibaca.  
*Write*: word awal yang dituju dan jumlah word yang akan ditulis.
- FCS: Untuk mengecek adanya kesalahan transmisi.
- Terminator: Kode yang menandakan *block* sudah berakhir.

## 2. Response Block



**Gambar 2.29** Format response block<sup>[13]</sup>

Keterangan:

@: Kode awal yang harus diberikan.

Node No.: Nomor identitas PLC.

Header code: Menunjukkan operasi (*read* atau *write*) di area tertentu yang tadi dikirim.

End code: Penanda komunikasi berjalan normal atau tidak. Jika tidak kode ini juga menandakan terjadi kesalahan di bagian mana.

Text: *Read: word* yang terbaca.

FCS: Untuk mengecek adanya kesalahan transmisi.

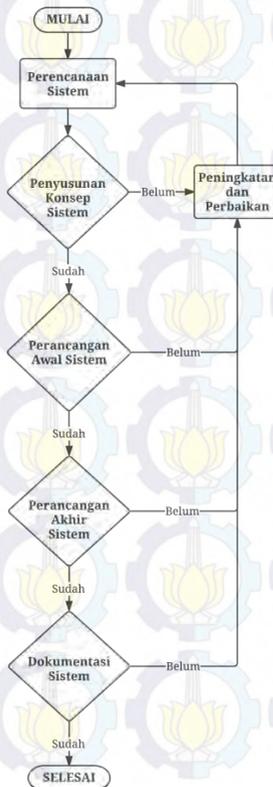
Terminator: Kode yang menandakan *block* sudah berakhir.

### 2.14.2 Frame Check Sequence (FCS)

FCS adalah teknik pemeriksaan kesalahan dalam transmisi sebuah data. Pengecekan dilakukan dengan mengirimkan sandi berukuran 8 bit sebelum *terminator*. FCS merupakan data 8 bit yang terdiri dari hasil operasi “XOR” yang dilakukan pada tiap data dari awal *frame* sampai akhir *frame*. Lalu data 8 bit ini dikonversikan menjadi 2 karakter ASCII. FCS akan dicek ulang dengan oleh penerima untuk mendeteksi kesalahan di *frame*. Jika isi FCS yang dikirim dengan hasil perhitungan FCS ulang oleh penerima tidak sama, maka dapat dipastikan telah terjadi kesalahan transmisi data.

## BAB 3 PERANCANGAN SISTEM

Bab ini akan membahas mengenai perancangan sistem secara keseluruhan untuk sistem pendeteksi lubang pada kertas tisu berbasis informasi visual dengan Raspberry Pi. Pada perancangan sebuah sistem, dasar-dasar alur pengerjaan sangat diperlukan. Tahap-tahap perencanaan dan perancangan sistem ini digambarkan dalam sebuah diagram alur<sup>[27]</sup>.



**Gambar 3.1** Alur perancangan sistem

Tahap-tahap perancangan sistem ini dapat dilihat di Gambar

3.1. Perancangan sistem ini dibagi menjadi lima tahap, yaitu:

1. Perencanaan Sistem  
Pada tahap ini akan dilakukan pengumpulan informasi dan memilih ide dari sistem ini. Kemudian dilanjutkan dengan membuat kebutuhan-kebutuhan yang harus dipenuhi.
2. Penyusunan Konsep Sistem  
Tahap ini mengidentifikasi masalah yang penting dari ide sistem ini dan menyusun struktur sistem. Pada tahap ini dilakukan pengumpulan informasi tentang prinsip dan struktur kerja dari sistem dan digabungkan menjadi suatu konsep.
3. Perancangan Awal Sistem  
Pada tahap ini dilakukan perancangan bentuk awal, pemilihan material yang akan digunakan, dan perhitungan yang sudah dievaluasi dalam segi teknik.
4. Perancangan Akhir Sistem  
Pada tahap ini perancangan sistem sudah selesai dan dilakukan pengecekan untuk mengeliminasi titik lemah dan *error* di sistem.
5. Dokumentasi Sistem  
Dilakukan dokumentasi lengkap berisi gambar dari sistem dan daftar bagiannya. Lalu dilakukan penulisan instruksi pengoperasian sistem dan kemampuan sistem.

Setiap tahapan proses selesai, maka hasil proses dari tahapan tersebut menjadi masukan untuk tahapan selanjutnya dan menjadi umpan balik terhadap tahapan sebelumnya. Perancangan dengan metode ini dapat berubah-ubah setiap saat berdasarkan umpan balik yang diberikan ke tahapan sebelumnya.

### **3.1 Perencanaan dan Penyusunan Konsep Sistem**

Perencanaan dan penyusunan konsep dari sistem akan membuat kebutuhan dari sistem setelah dilakukan studi literature untuk mengumpulkan informasi. Dari kebutuhan ini akan dianalisa dan ditimbang ulang dengan beberapa pertimbangan untuk sistem. Dari

kedua tahapan sebelumnya maka akan disusun tuntutan dari sistem ini yang nanti akan dikembangkan ke konsep pemecahan masalah sistem.

### **3.1.1 Kebutuhan Sistem**

Sistem pendeteksi lubang pada kertas tisu ini diperlukan di industri-industri kertas tisu untuk melakukan pengendalian kualitas. Industri kertas tisu perlu mengetahui adanya cacat atau tidak dalam suatu gulungan kertas tisu. Saat proses penggulungan berjalan, sistem ini akan mengawasi jalannya proses dan memberi tanda jika kertas tisu berlubang.

Alat-alat yang digunakan oleh industri kertas tisu memiliki harga yang sangat mahal hingga puluhan juta. Oleh karena itu dirancang sebuah sistem yang diusahakan menjadi alternatif yang lebih murah bagi kamera-kamera industri terutama di bidang industri kertas tisu. Alat ini merupakan sebuah prototip yang diharapkan dapat ditingkatkan lagi kemampuannya ke depannya. Raspberry Pi menjadi pilihan utama dari sistem ini karena kemampuan dan harganya yang masih terjangkau. Sistem ini pun membutuhkan alat yang mampu mensimulasikan proses penggulungan di industri.

Berdasarkan kebutuhan di atas, maka diperlukan beberapa langkah analisis untuk memperjelas tugas perancangan sistem ini. Ada dua bagian sistem yang akan dirancang dalam Tugas Akhir ini, yaitu:

1. Sistem Pendeteksi Lubang pada Kertas Tisu Berbasis Informasi Visual dengan Raspberry Pi
  - a. Spesifikasi Prosesor  
Prosesor akan menggunakan Raspberry Pi yang lebih murah sehingga dapat menekan ongkos ekonomi pada penelitian ini. Sistem ini juga harus fleksibel dan dapat diubah-ubah sesuai konfigurasi industri.
  - b. Spesifikasi Kamera  
Kamera menggunakan kamera digital yang sesuai dengan Raspberry Pi dan spesifikasinya cukup baik. Kamera ini pun fleksibel dan dapat diubah-ubah lensanya untuk memenuhi permintaan industri.

c. Spesifikasi Penampilan

Sistem pendeteksi ini memiliki konstruksi yang telah disesuaikan dengan keamanan dan kenyamanan dalam pemrosesan. Sistem ini memiliki ukuran kecil sehingga dapat ditempatkan di mana saja untuk memantau proses penggulungan.

2. Sistem Simulator Penggulungan Kertas Tisu

a. Spesifikasi Prosesor

Prosesor yang digunakan oleh sistem ini adalah PLC yang banyak digunakan untuk industri-industri yang ada. Maka sistem pendeteksi harus dapat berkomunikasi dengan PLC untuk kepentingan pengendalian.

b. Spesifikasi Tenaga Penggerak

Tenaga penggerak di sini menggunakan motor listrik yang memiliki kemudahan perawatan dan memiliki tingkat polusi rendah. Motor ini harus bisa dikendalikan kecepatannya untuk melakukan simulasi dan pengujian.

c. Spesifikasi Sistem

Alat ini menggunakan *conveyor* yang sudah dimodifikasi sehingga dapat melakukan simulasi penggulungan kertas tisu.

d. Spesifikasi Penampilan

Sistem ini memiliki konstruksi yang sudah disesuaikan keamanan dan nyaman dalam bekerja. Sistem ini memiliki ukuran yang tidak terlalu besar.

### 3.1.2 Pertimbangan Perencanaan Sistem

Didasari dari kebutuhan sistem di atas, maka pertimbangan yang dilakukan pada sistem ini adalah:

1. Sistem Pendeteksi Lubang pada Kertas Tisu Berbasis Informasi Visual dengan Raspberry Pi

a. Pertimbangan Teknis

- Konstruksi kuat dan baik.
- Kemudahan dalam pengoperasian.
- Proses pemasangan dan perawatan mudah dilakukan.

- b. Pertimbangan Ergonomis
    - Sistem lebih murah.
    - Sistem cukup sederhana.
    - Sistem hemat daya.
  - c. Pertimbangan Lingkungan
    - Sistem hemat daya.
  - d. Pertimbangan Keselamatan Kerja
    - Sistem cukup aman digunakan oleh operator.
2. Sistem Simulator Penggulungan Kertas Tisu
- a. Pertimbangan Teknis
    - Kontruksi kuat dan baik.
    - Kemudahan dalam pengoperasian.
    - Proses pemasangan mudah dilakukan.
  - b. Pertimbangan Ergonomis
    - Sistem mudah digunakan untuk pengujian.
  - c. Pertimbangan Lingkungan
    - Sistem tidak bising.
    - Sistem tidak menimbulkan polusi berlebih.
  - d. Pertimbangan Keselamatan Kerja
    - Sistem cukup aman digunakan oleh operator.

### **3.1.3 Tuntutan Perencanaan Sistem**

Berdasarkan pertimbangan sebelumnya, maka kita dapat menguraikan tuntutan untuk perencanaan sistem ini yang terdiri dari:

- 1. Sistem Pendeteksi Lubang pada Kertas Tisu Berbasis Informasi Visual dengan Raspberry Pi
  - a. Tuntutan Konstruksi
    - Konstruksi dapat menahan beban dan getaran sistem ketika beroperasi.
    - Perawatan mudah dilakukan.
  - b. Tuntutan Fungsi
    - Sistem dapat mendeteksi lubang dengan baik pada kertas tisu berbasis informasi visual.

- Sistem dapat member tanda jika terdeteksi lubang pada kertas tisu.
- c. Tuntutan Pengoperasian
    - Pengoperasian cukup mudah.
    - Operator tidak memerlukan keahlian khusus untuk mengoperasikannya.
  - d. Tuntutan Keamanan
    - Komponen-komponen sistem yang berpotensi menyebabkan kecelakaan diberi perlindungan.
  - e. Tuntutan Ergonomis
    - Sistem berukuran kecil.
    - Sistem mudah dipasang dan dipindahkan.
    - Sistem ini memerlukan satu operator.
    - Sistem lebih murah dan hemat daya.
2. Sistem Simulator Penggulungan Kertas Tisu
    - a. Tuntutan Konstruksi
      - Rangka dapat menahan beban dan getaran sistem ketika beroperasi.
      - Dapat melakukan proses penggulungan kertas tisu dengan baik.
      - Pengelasan harus kuat.
    - b. Tuntutan Fungsi
      - Sistem diharapkan dapat mensimulasikan penggulungan kertas tisu dengan baik.
      - Sistem dapat diubah-ubah kecepatannya.
      - Sistem menggunakan PLC sebagai standar industri.
    - c. Tuntutan Pengoperasian
      - Pengoperasian cukup mudah.
    - d. Tuntutan Keamanan
      - Komponen-komponen sistem yang berpotensi menyebabkan kecelakaan diberi perlindungan.
    - e. Tuntutan Ergonomis
      - Sistem tidak berukuran terlalu besar.

- Sistem dapat dipindah-pindahkan.
- Sistem hanya membutuhkan satu operator.

### 3.1.4 Penyusunan Konsep Sistem

Peyusunan konsep sistem ini berguna untuk memberi solusi atas tuntutan sistem di atas. Berdasarkan tahapan sebelumnya kita dapat menyusun konsep bagia kedua sistem seperti di bawah ini:

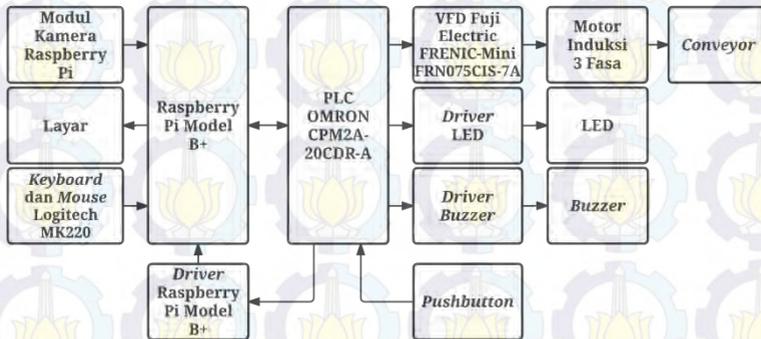
1. Sistem Pendeteksi Lubang pada Kertas Tisu Berbasis Informasi Visual dengan Raspberry Pi
  - a. Menggunakan Raspberry Pi sebagai prosesor utama.
  - b. Menggunakan modul kamera Raspberry Pi untuk menangkap informasi visual.
  - c. Memberi *casing* akrilik untuk perlindungan dan keamanan sistem.
  - d. Hanya dioperasikan oleh *pushbutton*.
  - e. Menggunakan *buzzer* sebagai tanda terdeteksinya lubang pada kertas tisu.
2. Sistem Simulator Penggulungan Kertas Tisu
  - a. Menggunakan motor induksi 3 fasa sebagai penggerak utama.
  - b. Menggunakan PLC sebagai prosesor utama.
  - c. Menggunakan VFD untuk mengatur kecepatan motor induksi 3 fasa.
  - d. Perangkat-perangkat dan kabel-kabel dirapikan demi perlindungan dan kemanan sistem.

### 3.2 Perancangan Perangkat Keras

Perancangan perangkat keras sistem ini didasarkan pada kebutuhan sistem untuk bekerja secara lancar. Pengerjaan kedua sistem yang sudah direncanakan sebelumnya akan dikerjakan secara bersamaan dan dianggap sebagai suatu integritas. Secara keseluruhan, perancangan perangkat keras ini dapat dilihat di Gambar 3.2 dan dibagi menjadi beberapa bagian, yaitu:

1. Perancangan rangkaian elektrik
2. Perancangan mekanik

3. Sumber daya DC
4. Motor induksi 3 fasa
5. *Variable Frequency Drive* (VFD)
6. *Programmable Logic Controller* (PLC)
7. Raspberry Pi



**Gambar 3.2** Konfigurasi perangkat keras

Rangkaian elektrik digunakan sebagai pembatas arus dari sumber daya DC dan sebagai hasil keluaran akhir dari sistem. Perancangan mekanik merupakan rancangan dari perangkat keras yang digunakan untuk mendukung kerjanya sistem. Seluruh sumber daya DC yang diperlukan sistem akan disuplai oleh satu sumber daya DC. Motor induksi 3 fasa digunakan sebagai penggerak utama *conveyor* yang dikendalikan oleh VFD. PLC akan menunjang segala logika yang digunakan untuk mengendalikan segala keluaran akhir dari sistem ini. Seluruh pemrosesan citra dan kerja sistem akan ditunjang oleh Raspberry Pi.

Selain terdapat perancangan perangkat keras keseluruhan yang mencakup konfigurasi dan hubungan dari tiap perangkat keras, juga terdapat perancangan perangkat lunak yang nanti akan dijelaskan di *flowchart* program.

Pada perancangan ini terdapat beberapa asumsi yang terdapat di sistem ini, yaitu:

1. *Conveyor* merupakan hasil perancangan ulang dari perancangan yang telah dilakukan oleh mahasiswa ITS sebelumnya.
2. *Conveyor* tidak digunakan sebagai sarana pemindah materi, tetapi digunakan sebagai sarana simulasi proses penggulungan kertas tisu.

### **3.2.1 Perancangan Rangkaian Elektrik**

Rangkaian elektrik dirancang di atas sebuah *printed circuit board* (PCB) dan terdiri dari beberapa komponen rangkaian elektrik yang digunakan oleh sistem pendeteksi ini. Rangkaian elektrik ini diimplementasikan sebagai pembatas arus, masukan, dan keluaran dari sistem ini. Meskipun terdapat lima rangkaian elektrik di sistem ini tetapi kedua rangkaian ini diimplementasikan di tiga PCB saja yang diletakkan di dalam kotak plastik.

#### **3.2.1.1 Perancangan Rangkaian Pembatas Arus**

Rangkaian pembatas arus di sistem ini digunakan untuk membatasi arus yang digunakan untuk mengendalikan VFD. Spesifikasi arus yang digunakan sebagai pengendali VFD dan spesifikasi arus yang dihasilkan sumber daya DC berbeda. VFD membutuhkan arus dengan spesifikasi maksimal 50 mA untuk pengendalinya, sedangkan sumber daya DC yang digunakan menghasilkan tegangan +24 V dengan arus 6,25 A. Maka kita dapat menghitung tahanan internal sebesar 3,84  $\Omega$  dari sumber daya DC ini dengan rumus di bawah ini.

$$V = I \times R$$

Keterangan:

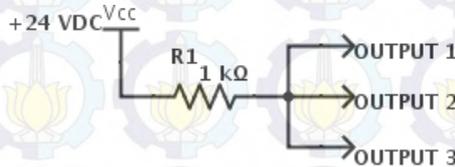
V = tegangan (V)

I = arus (A)

R = tahanan ( $\Omega$ )

Lalu tahanan ditambahkan 1 k $\Omega$  sehingga menjadi 1003,84  $\Omega$ . Lalu dengan rumus sebelumnya didapatkan tiga keluaran +24 VDC dengan arus sebesar 23,9 mA  $\approx$  24 mA. Ketiga keluaran ini akan digunakan untuk menyuplai keluaran '00'-'04' dari PLC. Sedangkan

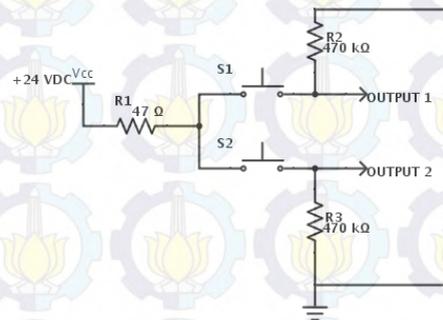
untuk mengendalikan VFD, sistem ini menggunakan keluaran '00'-'04' dari PLC. Rangkaian ini hanya berisi resistor sebesar 1 k $\Omega$  yang digunakan untuk mengurangi arus. Keseluruhan rangkaian ini digambarkan di Gambar 3.3.



**Gambar 3.3** Rancangan skematik pembatas arus

### 3.2.1.2 Perancangan Rangkaian Pushbutton

*Pushbutton* digunakan sebagai masukan untuk menyalakan dan mematikan seluruh sistem. Dengan ditekannya *pushbutton* ini akan ada masukan ke PLC sebesar +24 VDC. Rangkaian ini mengambil sumber daya dari PLC ini sendiri yang memiliki sumber daya DC sebesar +24 VDC. Berikut ini merupakan rancangan rangkaian di Gambar 3.4.



**Gambar 3.4** Rancangan skematik *pushbutton*

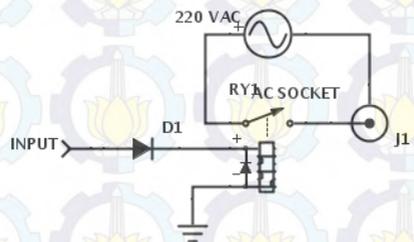
Rangkaian ini memiliki beberapa komponen sebagai berikut:

1. Resistor 47  $\Omega$  yang berfungsi sebagai hambatan agar rangkaian dan PLC tidak *short* ketika *pushbutton* ditekan.

2. Dua *pushbutton* yang digunakan sebagai masukan ke PLC untuk menyalakan dan mematikan sistem.
3. Dua resistor 470 k $\Omega$  yang digunakan untuk membuang arus listrik yang tersisa ketika *pushbutton* tidak ditekan.

### 3.2.1.3 Perancangan Rangkaian Driver Raspberry Pi Model B+

*Driver buzzer* pada sistem ini akan digunakan untuk memisahkan antara keluaran PLC dan Raspberry Pi yang memiliki persyaratan spesifikasi tegangan yang berbeda dengan spesifikasi tegangan dari keluaran PLC. Raspberry Pi membutuhkan tegangan 0 VAC untuk mati dan 220 VAC untuk menyala. Sedangkan PLC tetap menyalurkan keluaran sebesar +24 VDC. Maka dari itu dirancang *driver* ini dengan *relay* eksternal untuk mengendalikan Raspberry Pi. Keluaran PLC yang memiliki spesifikasi tegangan berbeda dengan Raspberry Pi akan digunakan untuk mengaktifkan gulungan *relay* yang selanjutnya akan mengaktifkan kontak *relay*. Kontak *relay* akan menyalurkan arus listrik dengan spesifikasi tegangan yang sesuai untuk Raspberry Pi dan menyalakannya. Berikut ini merupakan rancangan rangkaian *driver* Raspberry Pi yang tergambar di Gambar 3.5.



**Gambar 3.5** Rancangan skematik *driver* Raspberry Pi Model B+

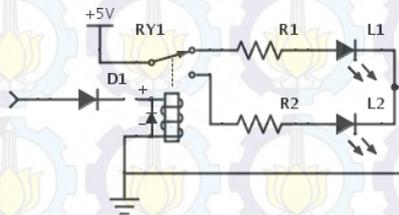
Rangkaian ini memiliki beberapa komponen, yaitu:

1. Dioda yang berfungsi untuk membatasi arus yang melewati *relay*.
2. Dioda yang berfungsi untuk membuang arus listrik ketika gulungan *relay* di-non-aktifkan.

3. *Relay* yang berfungsi sebagai saklar elektrik antara PLC dan Raspberry Pi.
4. Stopkontak yang terhubung dengan sumber daya 220 VAC melewati *relay*.

#### 3.2.1.4 Perancangan Rangkaian Driver Light Emitting Diode (LED)

*Driver* LED pada sistem ini akan digunakan untuk memisahkan antara keluaran PLC dan LED yang memiliki persyaratan spesifikasi tegangan yang berbeda dengan spesifikasi tegangan dari keluaran PLC. LED membutuhkan tegangan 0 VDC untuk mati dan +5 VDC untuk menyala. Sedangkan PLC tetap menyalurkan keluaran sebesar +5 VDC bahkan ketika berada di logika '0' dan +24 VDC di logika '1'. Maka dari itu dirancang *driver* ini dengan *relay* eksternal untuk mengendalikan LED. Keluaran PLC yang memiliki spesifikasi tegangan berbeda dengan LED akan digunakan untuk mengaktifkan gulungan *relay* yang selanjutnya akan mengaktifkan kontak *relay*. Kontak *relay* akan menyalurkan arus listrik dengan spesifikasi tegangan yang sesuai dari GPIO Raspberry Pi untuk LED dan menyalakannya. Berikut ini merupakan rancangan rangkaian *driver* LED yang tergambar di Gambar 3.6.



**Gambar 3.6** Rancangan skematik *driver* LED

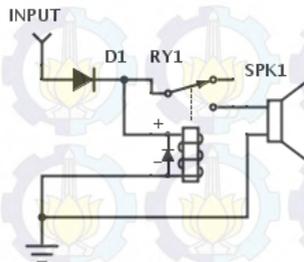
Rangkaian ini memiliki beberapa komponen, yaitu:

1. Dioda yang berfungsi untuk membatasi arus yang melewati *relay*.
2. Dioda yang berfungsi untuk membuang arus listrik ketika gulungan *relay* di-non-aktifkan.

3. *Relay* yang berfungsi sebagai saklar elektrik antara PLC dan *buzzer*.
4. Dua resistor yang berfungsi sebagai pembatas arus untuk LED.
5. Dua LED yang digunakan untuk menandakan siap atau tidaknya sistem.

### 3.2.1.5 Perancangan Rangkaian Driver Buzzer

*Driver buzzer* pada sistem ini akan digunakan untuk memisahkan antara keluaran PLC dan aktuator berupa *buzzer* yang memiliki persyaratan spesifikasi tegangan yang berbeda dengan spesifikasi tegangan dari keluaran PLC. *Buzzer* membutuhkan tegangan 0 VDC untuk mati dan +1-24 VDC untuk menyala. Sedangkan PLC tetap menyalurkan keluaran sebesar +5 VDC bahkan ketika berada di logika '0'. Maka dari itu dirancang *driver* ini dengan *relay* eksternal untuk mengendalikan *buzzer*. Keluaran PLC yang memiliki spesifikasi tegangan berbeda dengan *buzzer* akan digunakan untuk mengaktifkan gulungan *relay* yang selanjutnya akan mengaktifkan kontak *relay*. Kontak *relay* akan menyalurkan arus listrik dengan spesifikasi tegangan yang sesuai untuk *buzzer* dan menyalakannya. Berikut ini merupakan rancangan rangkaian *driver buzzer* yang tergambarkan di Gambar 3.7.



**Gambar 3.7** Rancangan skematik *driver buzzer*

Rangkaian *driver buzzer* ini mencakup beberapa komponen, yaitu:

1. Dioda yang berfungsi untuk membatasi arus yang melewati *relay*.

2. Dioda yang berfungsi untuk membuang arus listrik ketika gulungan *relay* di-non-aktifkan.
3. *Relay* yang berfungsi sebagai saklar elektrik antara PLC dan *buzzer*.
4. *Buzzer* yang berfungsi sebagai alarm ketika terdeteksi lubang di kertas tisu.

### 3.2.2 Perancangan Mekanik

Perancangan mekanik merupakan rancangan sistem penunjang dari keseluruhan sistem agar proses berjalan secara lancar. Perancangan bagian mekanik dari sistem ini meliputi perancangan mekanik *conveyor* dan perancangan mekanik sistem *belt* dan *pulley* yang dapat dilihat di Gambar 3.8.



**Gambar 3.8** Rancangan mekanik keseluruhan

Perancangan mekanik *conveyor* adalah penambahan rancangan dari rancangan sebelumnya agar sesuai dengan sistem ini. *Conveyor* dalam sistem ini hanya digunakan sebagai perangkat untuk melakukan simulasi proses penggulungan kertas tisu dan tidak berfungsi untuk memindahkan barang. Sedangkan perancangan sistem *belt* dan *pulley* merupakan rancangan sebagai penunjang pergerakan *conveyor* melalui sebuah motor induksi 3 fasa. Seluruh sistem penggerak berupa

VFD, motor induksi 3 fasa, dan sistem *belt* dan *pulley* diletakkan di rak tambahan di samping *conveyor*. Sedangkan lampu penerangan dan lengan kamera dan raspberry pi akan dipasang di pinggir *conveyor*. Rak tambahan juga diletakkan di bawah *conveyor* untuk meletakkan stop kontak, sumber daya DC, PLC, dan kotak rangkaian elektrik.

### 3.2.2.1 *Perancangan Mekanik Conveyor*

Dalam perancangan *conveyor* ini, saya menggunakan *conveyor* yang sudah ada sebelumnya. *Conveyor* yang telah ada sebelumnya tergambar di Gambar 3.9. *Conveyor* terbuat dari plat besi setebal 2 mm. Keseluruhan sistem *conveyor* ini memiliki panjang 103 cm, lebar 47 cm, dan tinggi 65,5 cm dengan luas penampang sabuk *conveyor* sebesar 94×30 cm. Sistem ini dilengkapi dengan dua rol berdiameter 5,08 cm di ujungnya dengan *bearing* untuk meletakkan sabuk karet *conveyor*. Salah satu rol bersifat tetap dan tersambung dengan sistem *belt* dan *pulley* yang digunakan sebagai penggerak *conveyor*, sedangkan rol lainnya dapat digeser untuk mengatur kekencangan sabuk *conveyor*. Lalu ada tiga rol yang lebih kecil berdiameter 3,175 cm dengan *bearing* yang digunakan untuk memuluskan jalannya sabuk.



**Gambar 3.9** Rancangan mekanik *conveyor* sebelumnya

Dari rancangan yang telah ada, saya menambahkan beberapa rak dari plat besi setebal 2 mm, rak dari plat akrilik setebal 5 mm pada

*conveyor*, dan rol. Untuk menambahkan rak sistem penggerak, digunakan plat besi setebal 2 mm seluas 25×20 cm yang dilas dan dibaut di samping *conveyor* dan dapat diatur tingginya. Sedangkan untuk rak sistem-sistem lainnya, digunakan akrilik setebal 5 mm seluas 70×39 cm yang dipasang di rangka kaki-kaki *conveyor*. Lalu ditambahkan juga batang besi yang berfungsi untuk meletakkan gulungan kertas tisu dan satu rol lagi berdiameter 3,81 cm di ujung yang dapat diatur ketinggiannya yang berfungsi untuk menjepit kertas tisu. Seluruh penambahan rancangan dengan materi plat besi dilakukan dengan teknik pengelasan. Keseluruhan sistem ini tergambar di Gambar 3.10.



**Gambar 3.10** Rancangan mekanik *conveyor*

Kecepatan translasi dari sabuk *conveyor* dapat dihitung dengan rumus berikut ini.

$$V = \frac{\pi \times d \times N_2}{6000}$$

Keterangan:

V = kecepatan translasi sabuk *conveyor* (m/s)

d = diameter rol penggerak *conveyor* (cm)

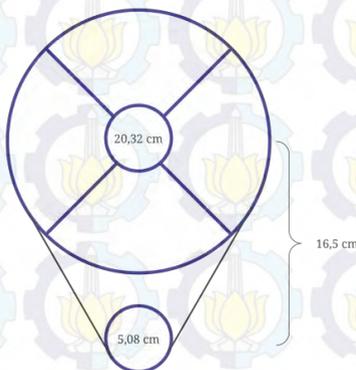
N<sub>2</sub> = kecepatan rotasi rol penggerak *conveyor* (RPM)

### 3.2.2.2 Perancangan Mekanik Sistem Belt dan Pulley

Sistem ini merupakan bagian dari sistem penggerak yang berfungsi untuk menstramisikan daya gerak dari motor induksi 3 fasa ke *conveyor*. Sistem ini memiliki *pulley* yang berupa roda besi dengan celah di pinggirnya di mana *belt* bisa diletakkan. *Belt* berbentuk lingkaran terbuat dari karet dengan tebal dan lebar 1cm. *Belt* dan dua *pulley* yang digunakan bersama-sama memiliki keuntungan mekanis tergantung dari rasio diameter kedua *pulley* dan jarak antara kedua poros *pulley* yang dihubungkan dengan *belt*.

Ketika dua *pulley* dengan diameter yang tidak sama disambungkan dengan *belt*, kecepatan dan torsi dari sebuah sistem penggerak dapat dimanipulasi. *Pulley* yang lebih besar dapat mengurangi kecepatan dan meningkatkan torsi, sedangkan *pulley* yang lebih kecil dapat meningkatkan kecepatan dan mengurangi torsi.

Ada dua *pulley* yang digunakan di sistem ini, *pulley* yang lebih kecil disambungkan dengan motor induksi 3 fasa dan yang lebih besar disambungkan ke *conveyor*. Sistem ini dirancang agar motor dapat berputar di frekuensi yang stabil tetapi tetap bisa memutar *conveyor* cukup pelan agar sistem pendeteksi lubang berjalan sempurna. Dengan kecepatan yang pelan dan torsi yang lebih besar di *conveyor*, motor juga lebih ringan memutar *conveyor*.



**Gambar 3.11** Rancangan mekanik sistem *belt* dan *pulley*

Kedua *pulley* ini berdiameter 20,32 cm dan 5,08 cm dengan jarak antara kedua porosnya sejauh 16,5 cm. Jika putaran dari motor dimisalkan sebesar 100, maka putaran di *pulley conveyor* akan sebesar 25 karena sistem ini memiliki rasio sebesar 4:1 yang didapatkan dengan rumus di bawah ini.

$$\frac{D_1}{D_2} = \frac{N_2}{N_1}$$

Keterangan:

$D_1$  = diameter *pulley* motor (cm)

$D_2$  = diameter *pulley conveyor* (cm)

$N_1$  = kecepatan rotasi *pulley* motor (RPM)

$N_2$  = kecepatan rotasi *pulley conveyor* (RPM)

Gambar 3.11 menggambarkan selengkapnya mengenai sistem *belt* dan *pulley* ini.

### 3.2.3 *Variable Frequency Drive (VFD)*

VFD digunakan sebagai pengendali motor induksi 3 fasa sebagai penggerak *conveyor*. Keluaran dari VFD ini berupa daya dengan tegangan dan frekuensi yang bervariasi untuk mengendalikan kecepatan motor. VFD sendiri dikendalikan oleh PLC untuk memiliki beberapa kecepatan yang sudah ditentukan di program VFD. VFD yang digunakan di sistem ini adalah VFD Fuji Electric FRENIC-Mini FRN075CIS-7A.

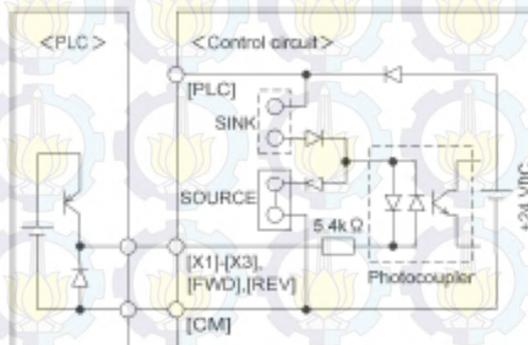
#### 3.2.3.1 *VFD Fuji Electric FRENIC-Mini FRN075CIS-7A*

VFD Fuji Electric FRENIC-Mini FRN075CIS-7A adalah VFD digunakan untuk mengendalikan motor induksi 3 fasa. VFD ini bekerja dengan suplai daya satu fasa 200-240 VAC. VFD ini dapat dikontrol dengan sinyal analog maupun digital, tetapi di sistem ini saya menggunakan sinyal digital dari PLC untuk mengontrolnya. Keluaran dari VFD ini merupakan daya tiga fasa yang disuplai ke motor untuk

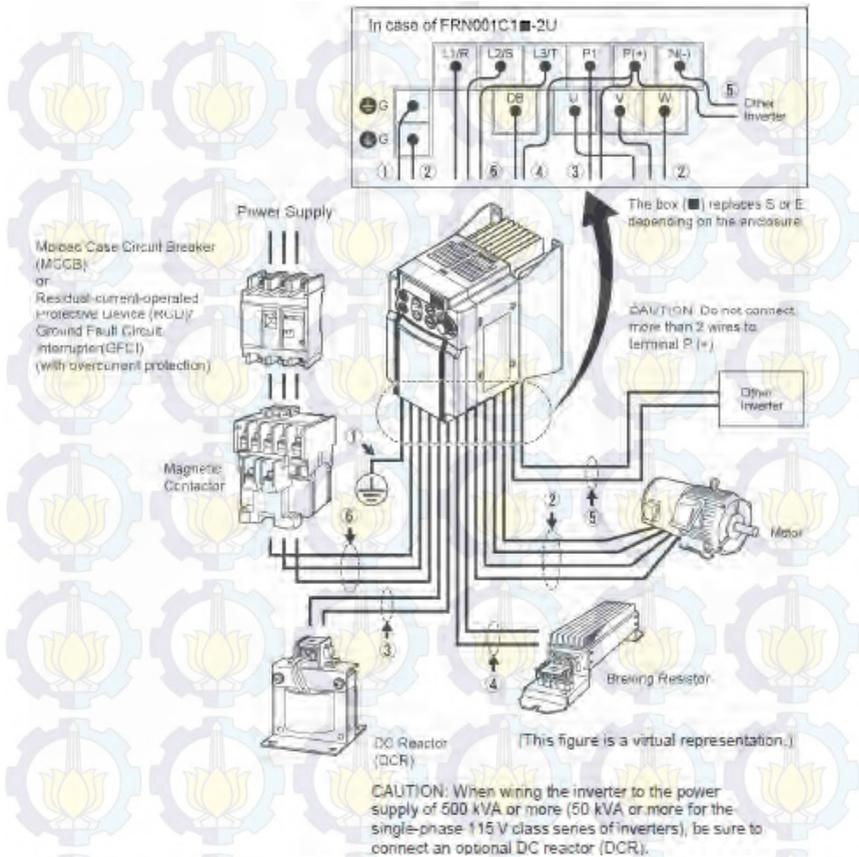
mengatur kecepatannya. Berikut ini merupakan konfigurasi I/O dari VFD di Tabel 3.1. Lalu *wiring* sumber daya dan motor induksi 3 fasa di Gambar 3.13 dan rangkaian skematik untuk mengendalikan VFD dari PLC dengan konfigurasi *sourcing* di Gambar 3.12. Pada Gambar 3.13 digunakan VFD Fuji Electric FRENIC-Mini FRN001C1□-2U yang disuplai daya 3 fasa, tetapi kita menggunakan daya 1 fasa untuk VFD Fuji Electric FRENIC-Mini FRN075CIS-7A sehingga hanya perlu 2 kabel dari sumber daya bukan 3 kabel.

**Tabel 3.1** Konfigurasi I/O VFD

Terminal	Perangkat Keras	Sambungan
L	Stop Kontak	L
N	Stop Kontak	N
G	Stop Kontak	GND
X1	PLC	10.00
X2	PLC	10.01
X3	PLC	10.02
FWD	PLC	10.03
REV	PLC	10.04
CM	Sumber Daya DC	V-
U	Motor Induksi 3 Fasa	U
V	Motor Induksi 3 Fasa	V
W	Motor Induksi 3 Fasa	W



**Gambar 3.12** Rancangan skematik kontrol *sourcing* VFD dari PLC<sup>[15]</sup>



**Gambar 3.13** Wiring VFD Fuji Electric FRENIC-Mini FRN001C1□-2U<sup>[15]</sup>

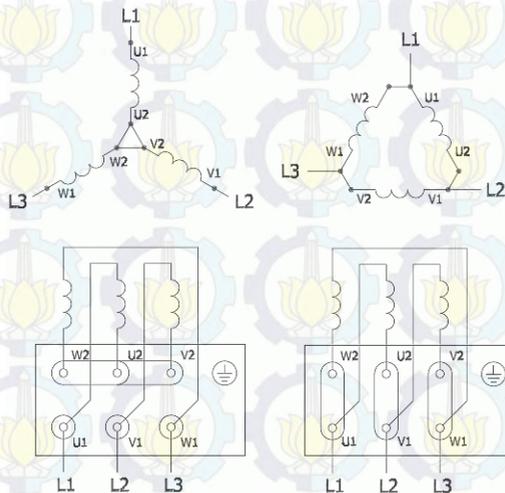
VFD dapat diprogram dengan menggunakan antarmuka VFD berupa layar dan tombol-tombol yang terdapat di VFD. VFD ini diprogram sesuai dengan *rating* motor, kecepatan yang kita butuhkan, dan dengan apa kita mengendalikannya, dan lain-lain. Berikut ini merupakan gambaran dari VFD yang digunakan di Gambar 3.14.



**Gambar 3.14** VFD Fuji Electric FRENIC-Mini FRN075CIS-7A

### 3.2.3.2 *Motor Induksi 3 Fasa*

Motor induksi 3 fasa digunakan sebagai penggerak utama dari sistem *conveyor*. Motor ini menggunakan konfigurasi  $\Delta$  (delta atau segitiga) karena motor terus bekerja di kecepatan rendah sehingga membutuhkan daya lebih. Konfigurasi  $\Delta$  pada motor menghasilkan daya yang besar dan torsi yang besar pula. *Wiring* ini dapat dilihat di Gambar 3.15.



**Gambar 3.15** *Wiring* motor induksi 3 fasa

Motor ini diatur kecepatannya oleh VFD dengan mengubah frekuensi, arus, dan tegangannya. Kecepatan dapat dihitung dengan rumus sebagai berikut.

$$N_1 = \frac{120 \times f}{p}$$

Keterangan:

$N_1$  = kecepatan rotasi motor (RPM)

$f$  = frekuensi tegangan (Hz)

$p$  = jumlah kutub magnetik motor

Berikut ini merupakan gambar dari motor induksi 3 fasa di Gambar 3.16.



**Gambar 3.16** Motor induksi 3 fasa

### 3.2.4 *Programmable Logic Controller (PLC)*

PLC digunakan sebagai pengendali keluaran akhir dari sistem ini. Perangkat-perangkat yang dikendalikan oleh PLC adalah:

1. Sebuah *driver buzzer* yang memerlukan satu keluaran dari PLC.
2. Sebuah VFD yang memerlukan lima keluaran dari PLC.

PLC yang digunakan di sistem ini adalah PLC OMRON CPM2A-20CDR-A.

### 3.2.4.1 PLC OMRON CPM2A-20CDR-A

PLC OMRON CPM2A-20CDR-A adalah PLC berjenis *compact* dengan 12 masukan digital dan 8 keluaran digital. PLC ini berkerja dengan suplai daya 100-240 VAC untuk menjalankannya. Tegangan yang digunakan oleh keluaran dari PLC ini adalah +24 VDC. Keluaran dari PLC ini adalah keluaran *relay* dan disetel dengan konfigurasi *sourcing*. Berikut ini merupakan gambar dari PLC OMRON CPM2A-20CDR-A di Gambar 3.17.

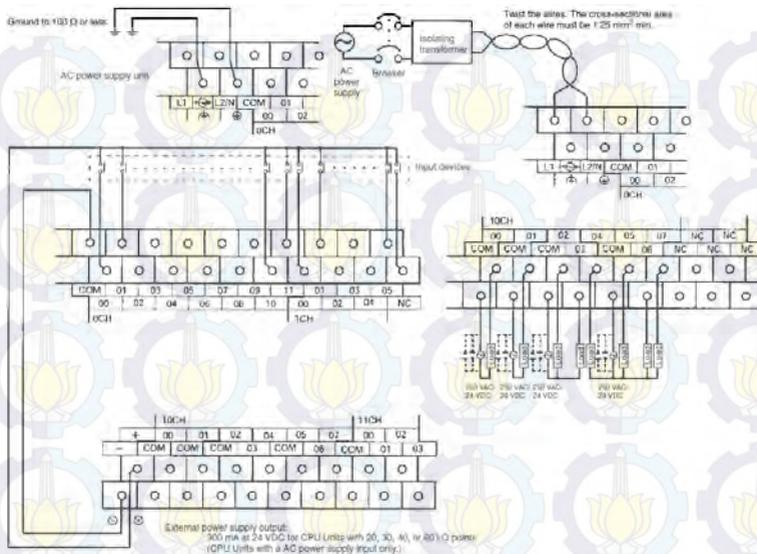


**Gambar 3.17** PLC OMRON CPM2A-20CDR-A

Konfigurasi dari I/O yang digunakan oleh sistem beserta perangkat apa saja yang terhubung dengan I/O tersebut dapat dilihat di Tabel 3.2 dan *wiring* dapat dilihat di Gambar 3.18.

**Tabel 3.2** Konfigurasi I/O PLC

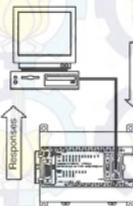
Alamat	Jenis	Perangkat Keras	Sambungan
0.00	Masukan	<i>Pushbutton</i> "START"	+24 VDC <i>Common</i>
0.01	Masukan	<i>Pushbutton</i> "STOP"	+24 VDC <i>Common</i>
10.00	Keluaran	VFD (X1)	<i>Common</i> +24VDC
10.01	Keluaran	VFD (X2)	<i>Common</i> +24VDC
10.02	Keluaran	VFD (X3)	<i>Common</i> +24VDC
10.03	Keluaran	VFD (FWD)	<i>Common</i> +24VDC
10.04	Keluaran	VFD (REV)	<i>Common</i> +24VDC
10.05	Keluaran	<i>Driver Raspberry Pi</i>	<i>Common</i> +24VDC
10.06	Keluaran	<i>Driver LED</i>	<i>Common</i> +24VDC
10.07	Keluaran	<i>Driver Buzzer</i>	<i>Common</i> +24VDC



**Gambar 3.18** Wiring PLC OMRON CPM2A-20CDR-A<sup>[28]</sup>

### 3.2.4.2 Sambungan Komunikasi Serial RS-232

Komunikasi pada PLC menggunakan komunikasi *hostlink* di mana PC akan mengirim perintah ke PLC, perintah ini dapat digunakan untuk membaca atau menulis data di dalam PLC, PLC pun akan memberikan respon ke PC sesuai dengan perintah yang terkirim. Komunikasi *hostlink* ini menggunakan *port* RS-232C dari PLC dengan *wiring* seperti Gambar 3.19.



**Gambar 3.19** Wiring komunikasi serial RS-232

Komunikasi ini tersambung menggunakan *port* RS-232C yang menggunakan sambungan DE-9 dan disambungkan dengan adaptor DE-9 ke USB karena PC dan Raspberry Pi tidak memiliki *port* DE-9, dari adaptor ini lalu langsung terhubung ke PC atau Raspberry Pi. Komunikasi ini digunakan untuk menjalankan sistem, di mana PLC mendapat perintah dari Raspberry Pi untuk mengatur keluarannya. Selain itu komunikasi ini juga digunakan untuk mengunggah program dari PC ke PLC.

### 3.2.5 Raspberry Pi Model B+

Raspberry Pi Model B+ digunakan sebagai pusat proses utama dari keseluruhan sistem. Raspberry Pi Model B+ dipilih karena produk ini mudah didapatkan di Indonesia dan relatif murah. SBC ini digunakan untuk menjalankan program utama dan mengendalikan seluruh keluaran. Program akan mendapatkan masukan dari modul kamera Raspberry Pi untuk diproses dan menghasilkan keluaran yang diteruskan ke PLC untuk diteruskan ke perangkat lainnya. Raspberry Pi Model B+ ini membutuhkan daya sebesar +5 VDC dan 1,2 A. Tetapi karena banyak perangkat yang terhubung ke Raspberry Pi Model B+ dan menggunakan cukup banyak daya, maka saya menggunakan adaptor daya AC yang memiliki spesifikasi daya +5 V dan 2,5 A. Perangkat-perangkat yang terhubung ke Raspberry Pi Model B+ dapat dilihat di Tabel 3.3 dengan *port* di Gambar 3.20.

**Tabel 3.3** Konfigurasi I/O Raspberry Pi Model B+

<i>Port</i>	Jenis	Perangkat Keras
Micro-USB	Masukan	Adaptor Daya AC
CSI	Masukan	Modul Kamera Raspberry Pi
USB	Masukan	<i>Keyboard</i> dan <i>Mouse</i>
USB	Keluaran	PLC
HDMI	Keluaran	Layar
GPIO 4	Keluaran	+5 VDC <i>Driver</i> LED
GPIO 6	Keluaran	GND <i>Driver</i> LED



**Gambar 3.20** Port Raspberry Pi Model B+

### 3.2.5.1 *Layar*

Layar digunakan untuk sebagai tampilan dari Raspberry Pi Model B+. Raspberry Pi Model B+layaknya sebuah PC memerlukan layar agar dapat digunakan. Layar ini memberikan tampilan agar kita dapat mengamati jalannya program pendeteksi lubang kertas tisu. Layar ini terhubung ke Raspberry Pi Model B+dengan sambungan HDMI. Layar apa saja dapat digunakan sebagai tampilan Raspberry Pi Model B+. Jika layar hanya memiliki layar dengan masukan VGA, maka dapat digunakan adaptor HDMI ke VGA.

### 3.2.5.2 *Keyboard dan Mouse*

Perangkat ini berfungsi sebagai masukan Raspberry Pi Model B+. Kedua perangkat ini digunakan untuk melakukan *troubleshooting* di Raspberry Pi Model B+. *Keyboard* dan *mouse* yang bejenis Logitech MK220 yang merupakan satu set *keyboard*, *mouse*, dan *wireless receiver*. *Keyboard* dan *mouse* terhubung secara nirkabel ke *wireless*

*receiver* yang dipasang di *port* USB Raspberry Pi. Satu set *keyboard*, *mouse*, dan *wireless receiver* dapat dilihat di Gambar 3.21.



**Gambar 3.21** *Keyboard* dan *mouse* Logitech MK220

### 3.2.5.3 *Modul Kamera Raspberry Pi*

Modul kamera Raspberry Pi digunakan untuk mengambil video yang akan diolah oleh program pendeteksi lubang kertas tisu. Modul kamera ini terhubung ke Raspberry Pi Model B+ melalui kabel pita ke *Camera Serial Interface* (CSI) *port* dari Raspberry Pi. Modul kamera ini dapat menangkap video 1080p hingga 30 fps, 720p hingga 60 fps, dan VGA hingga 90 fps. Modul kamera dapat diakses menggunakan *library Multi-Media Abstraction Layer* (MMAL), *Video for Linux* (V4L), dan lain-lain, tetapi sistem ini menggunakan V4L untuk mengakses modul kamera. Modul kamera Raspberry Pi dapat dilihat di Gambar 3.22.



**Gambar 3.22** Modul kamera Raspberry Pi

Keuntungan dari penggunaan modul kamera Raspberry Pi ini adalah penggunaan prosesor Raspberry Pi yang lebih maksimal. Jika kita menggunakan *webcam* maka seluruh proses kamera dan proses dari program seluruhnya dijalankan oleh *Central Processing Unit* (CPU), sedangkan dengan menggunakan modul kamera Raspberry Pi, proses kamera akan dikerjakan oleh *Graphic Processing Unit* (GPU) dan

proses dari program dikerjakan oleh CPU. Pembagian kerja ini akan memaksimalkan kecepatan jalannya program. Dengan program yang berjalan lancar, maka *delay* akan berkurang dan sistem *real-time* dapat direalisasikan.

### 3.2.6 Perangkat Tambahan

Pada sistem ini, selain perangkat yang dirancang sendiri, terdapat juga perangkat tambahan yang sudah jadi. Perangkat tambahan ini sudah dirancang dengan bentuk dan fungsi tertentu juga, di mana fungsi ini digunakan untuk mendukung kinerja sistem ini.

#### 3.2.6.1 Sumber Daya DC

Sumber daya DC ini mengubah tegangan 220 VAC menjadi 24 VDC. Perangkat ini dapat menghasilkan keluaran +24 VDC ataupun -24 VDC dengan arus 6,25 A. Ada dua pasang keluaran yang terdapat di sumber daya. Setiap pasangannya dapat menghasilkan keluaran +24 VDC atau -24 VDC tergantung pemasangan. Sistem ini menggunakan dua pasang keluaran dari sumber daya. Keduanya diatur untuk menghasilkan +24 VDC. Kedua keluaran ini akan dihubungkan ke PLC yang berfungsi sebagai *relay* otomatis bagi perangkat lain. Satu keluaran digunakan untuk memasok *buzzer*, *driver* Raspberry Pi Model B+, *driver* LED, *driver buzzer*, sedangkan yang lainnya akan dikurangi arusnya dan digunakan untuk memasok VFD. Sumber daya DC ini dapat dilihat di Gambar 3.23.



**Gambar 3.23** Sumber daya DC HILED FX150-H1V24

### 3.2.6.2 *Lampu LED*

Lampu LED digunakan sebagai penerang sistem. Proses pengolahan citra membutuhkan tingkat pencahayaan yang tetap agar proses berjalan lancar. Berbedanya tingkat pencahayaan dapat membuat hasil yang berbeda di keluaran. Maka proses pengolahan citra diterangi oleh lampu agar tingkat pencahayaannya stabil dan mengurangi kesalahan. Lampu ini berjenis LED putih yang memiliki *driver* sendiri yang menghasilkan +5-12 VDC untuk lampu. Sementara itu *driver*-nya membutuhkan pasokan daya 100-240 VAC. Lampu ini dipasangkan ke badan *conveyor* dengan klem. Lampu ini dapat diaktifkan dengan menekan sebuah tombol yang terdapat di kepala lampu. Lampu ini dapat dilihat di Gambar 3.24.



**Gambar 3.24** Lampu LED Kris

### 3.2.6.3 *Lengan Raspberry Pi Model B+ Beserta Modul Kamera*

Lengan ini merupakan lengan mekanik yang terbuat dari potongan besi dan pegas yang disusun sedemikian rupa. Lengan ini dapat diatur-atur posisinya sesuai dengan kebutuhan sistem. Satu klem

digunakan untuk menjepit rak sistem penggerak, sedangkan klem lainya digunakan untuk menjepit Raspberry Pi dan modul kameranya. Lengan ini dapat dilihat di Gambar 3.25.

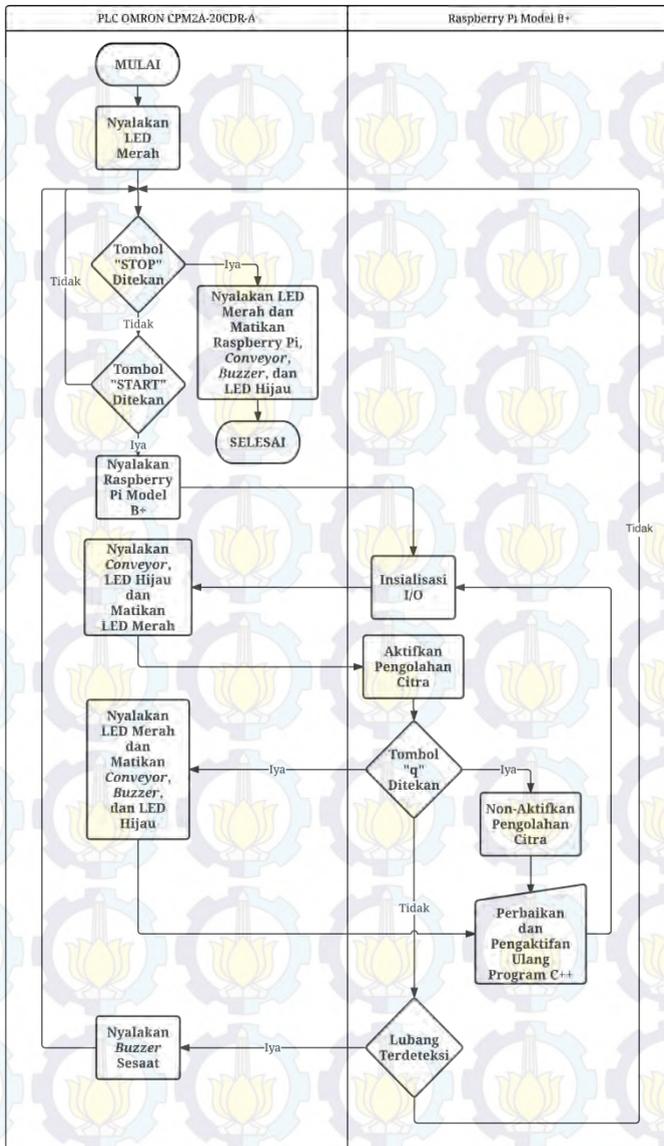


**Gambar 3.25** Lengan Raspberry Pi Model B+ beserta modul kamera

### **3.3 Perancangan Perangkat Lunak**

Perancangan perangkat lunak merupakan perancangan program yang menyatukan keseluruhan perangkat keras sehingga dapat bekerja bersama dan menjalankan tugas tertentu. Program untuk sistem ini dikembangkan di PC dengan menggunakan CX-Programmer 9.5 dan Code::Blocks, tetapi program akan dijalankan di Raspberry Pi yang sudah dilengkapi dengan beberapa *library* yang dibutuhkan.

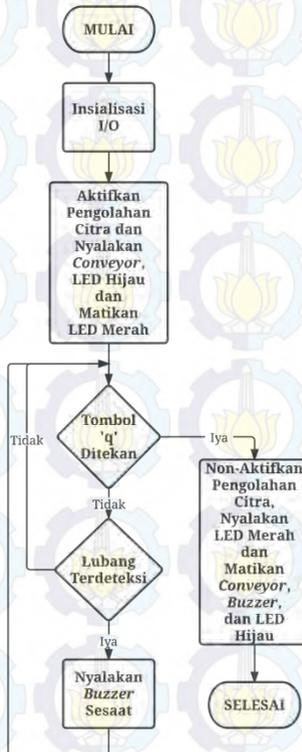
Keseluruhan cara kerja sistem memiliki logika pengerjaan seperti di Gambar 3.26 Program dibagi dua, ada program C++ yang dijalankan di Raspberry Pi dan program *ladder diagram* yang dijalankan di PLC. Program C++ di Raspberry Pi berfungsi untuk menjalankan pengolahan citra yang menentukan adanya lubang atau tidak dan memberi perintah ke PLC untuk menyalakan keluaran-keluaran melalui komunikasi serial. Sedangkan PLC digunakan untuk menyalakan Raspberry Pi yang mengontrol sistem dengan menekan *pushbutton*. PLC juga digunakan untuk menyalurkan perintah Raspberry Pi untuk mengendalikan keluaran-keluaran yang diinginkan.



**Gambar 3.26** Cara kerja program keseluruhan

### 3.3.1 PC

PC digunakan untuk mengembangkan program dari sistem ini. Untuk pengembangan program digunakan aplikasi CX-Programmer 9.5. dan Code::Blocks. CX-Programmer 9.5. digunakan untuk pembuatan logika kerja yang berada di PLC berupa *ladder diagram* dan mengunggahnya ke PLC. Program utama keseluruhan akan ditulis di Code::Blocks dalam bahasa C++ dengan *library* OpenCV 2.4.9. untuk pengolahan citra dan untuk komunikasi serial. Program C++ secara keseluruhan dapat dilihat di lampiran yang disediakan dengan cara kerja seperti di Gambar 3.27.



**Gambar 3.27** Cara kerja program C++

### 3.3.1.1 Program Komunikasi Serial RS-232

Program ini menggunakan *library* “rs232.h” yang dibuat oleh Teunis van Beelen<sup>[29]</sup>. Program ini terdiri dari dua tahap yang akan dijelaskan lebih lanjut sebagai berikut:

1. Mendefinisikan Variabel

Variabel yang didefinisikan di bagian ini adalah variabel *setting* komunikasi serial RS-232 dan seluruh *command block* yang akan digunakan untuk mengendalikan *buzzer* dan VFD. *Setting* komunikasi serial RS-232 yang digunakan dapat dilihat di Tabel 3.4.

**Tabel 3.4** *Setting* komunikasi serial RS-232

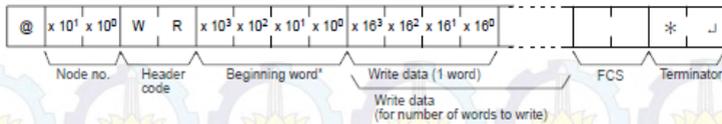
Parameter	Disarankan	Pilihan
Tipe PLC	OMRON CPM2A	
I/F PLC	RS-232	RS-232, RS-422, atau RS-485
<i>Baud Rate</i>	9600	9600 atau 19200
<i>Data Bits</i>	7	7 atau 8
<i>Parity</i>	<i>Even</i>	<i>Even, Odd, atau None</i>
<i>Stop Bits</i>	2	2 atau 1

2. Membuka Komunikasi Serial RS-232

Tahap ini dilakukan dengan fungsi “RS232\_OpenComport” dan memasukkan *setting* komunikasi serial RS-232 ke dalam fungsi tersebut dan *port* yang digunakan oleh sambungan komunikasi serial RS-232.

3. Mengirimkan *Command Block* ke PLC

Tahap ini dilakukan dengan menggunakan fungsi “RS232\_cputs” dan mengirimkan data dalam bentuk *string* yang sebelumnya sudah didefinisikan. *Format command block* dan *response block* selengkapnya dapat dilihat bab 2. Sedangkan *format* yang digunakan di program ini dapat dilihat di Gambar 3.28.



**Gambar 3.28** Format command block penulisan area IR/SR<sup>[13]</sup>

Keterangan:

@:

Kode awal yang harus diberikan

Node No.:

Nomor identitas PLC

Header code:

Penunjuk operasi yang dilakukan (*read* atau *write*) dan di area mana operasi tersebut dilakukan. Pada program ini digunakan kode “WR” yang melakukan operasi *write* di area IR/SR.

Beginning word:

Menunjukkan alamat yang dituju di area IR/SR di PLC.

Write data:

*word* awal yang dituju dan data yang akan ditulis.

FCS:

Untuk mengecek ada atau tidaknya kesalahan pada *frame* data.

Terminator:

Harus diberikan sebagai indikasi akhir dari *command block*.

### 3.3.1.2 Program Pendeteksi Lubang pada Kertas Tisu

Program ini menggunakan *library* dari OpenCV 2.4.9. yang sudah memiliki fungsi-fungsi tersendiri untuk melakukan pengolahan citra. Program ini terdiri dari beberapa tahap pengolahan citra untuk menentukan adanya lubang atau tidak yang akan dijelaskan di bawah ini:

1. Mendefinisikan Variabel

Pada bagian ini kita mendefinisikan jenis variabel-variabel kosong atau terisi dengan nilai yang nantinya akan digunakan oleh proses pengolahan citra.

2. Mengakses Modul Kamera Raspberry Pi

Melakukan akses ke modul kamera dan merekam video dengan fungsi “VideoCapture” dan mendefinisikan kamera mana yang digunakan. Lalu mengatur resolusinya dengan ukuran 256x208 dengan fungsi “set” berektensi “CV\_CAP\_PROP\_FRAME\_WIDTH” dan “set” berektensi “CV\_CAP\_PROP\_FRAME\_HEIGHT” tanpa tanda petik.

3. Mengambil *Frame*

Melakukan pengambilan *frame* dari video yang ditangkap oleh modul kamera dengan fungsi “>>” dan memasukkannya ke variabel “src”.

4. *Grayscale*

Melakukan proses *grayscale* pada variabel “src” menggunakan fungsi “cvtColor” berektensi “CV\_BGR2GRAY” dan menempatkannya di variabel “src\_gray”.

5. Menghaluskan

Variabel “src\_gray” akan dihaluskan dengan metode Gaussian menggunakan fungsi “GaussianBlur” dan hasilnya ditempatkan di variabel “src\_blur”.

6. Menyetarakan Histogram

Lalu melakukan penyetaraan histogram pada variabel “src\_blur” dengan fungsi “equalizeHist” dan hasilnya dimasukkan ke variabel “src\_hist”

7. Mendeteksi Tepian Canny

Melakukan pendeteksian tepian Canny terhadap variabel “src\_hist” dengan fungsi “Canny” menggunakan nilai batas bawah  $90^\circ$  dan nilai batas atas  $255^\circ$  untuk gradien tiap piksel dan mendeteksi tepian lubang. Hasil dari fungsi ini akan diletakkan di variabel “src\_canny”

8. Melakukan Operasi Morfologi

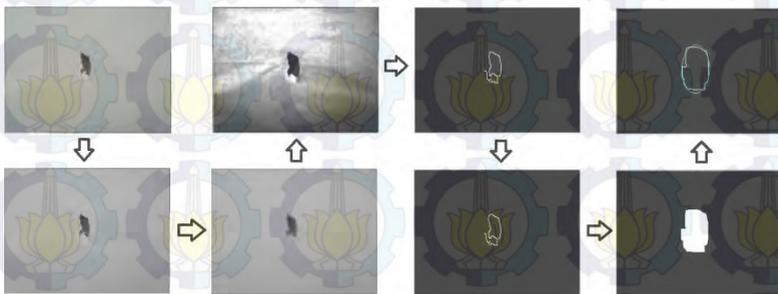
Pada tahap ini dilakukan operasi morfologi *opening* yang akan mengeliminasi *noise* yang masih tersisa di variabel “src\_canny”. Operasi ini dilakukan dengan melakukan operasi *erode* dan dilanjutkan dengan operasi *dilate*. Variabel “src\_canny akan melewati fungsi “erode” dengan elemen 1x1 dan menghasilkan

variabel “src\_erode”. Kemudian variabel “src\_erode” akan melewati fungsi “dilate” dengan elemen 10x10 dan menghasilkan variabel “src\_dilate”.

#### 9. Menemukan Kontur

Tahap akhir ini menggunakan fungsi “findContours” pada variabel “src\_dilate” untuk menemukan lubang dan menampilkan pendeteksian dengan fungsi “drawContours” dan menghasilkan variabel “drawing” yang ditampilkan dengan fungsi “imshow”. Jika ditemukan lubang juga akan dilakukan pengiriman *command block* ke PLC untuk menyalakan *buzzer*.

Hasil dari proses pengolahan citra ini dapat dilihat di Gambar 3.29 berikut.



**Gambar 3.29** Hasil proses pengolahan citra

#### 3.3.1.3 Program PLC

Pemrograman PLC ini dilakukan menggunakan CX-Programmer 9.5 di PC yang disambungkan ke PLC melalui komunikasi serial RS-232. Pemrograman ini terdiri dari beberapa tahap, yaitu:

1. Melakukan *Setting* Komunikasi Serial RS-232  
Lakukan pemilihan PLC dan *setting* komunikasi serial RS-232 seperti pada Tabel 3.4 sebelumnya.
2. Melakukan *Setting* PLC  
Kemudian *setting* PLC agar selalu berada di “*Monitor Mode*” sehingga bisa dikendalikan melalui *command block* yang dikirimkan dari Raspberry Pi melalui komunikasi serial RS-232.

### 3. Membuat Program *Ladder Diagram*

Lalu buat program *ladder diagram* dengan blok-blok fungsi yang sudah dijelaskan di bab 2. Program ini dapat dilihat selengkapnya di lampiran, sedangkan konfigurasi alamatnya dapat dilihat di Tabel 3.5.

**Tabel 3.5** Konfigurasi alamat PLC

Alamat	Keterangan
200.00	Menyalakan alamat “10.07”
200.01	Mematikan alamat “10.07”
201.00	Menyalakan alamat “10.06”
201.01	Mematikan alamat “10.06”
202.00	Menyalakan alamat “10.00”
202.01	Mematikan alamat “10.00”
203.00	Menyalakan alamat “10.01”
203.01	Mematikan alamat “10.01”
204.00	Menyalakan alamat “10.02”
204.01	Mematikan alamat “10.02”
205.00	Menyalakan alamat “10.03”
205.01	Mematikan alamat “10.03”
206.00	Menyalakan alamat “10.04”
206.01	Mematikan alamat “10.04”
0.00	Menyalakan alamat “10.05”
0.01	Mematikan alamat “10.05”
10.00	Mengatur terminal “X1” VFD
10.01	Mengatur terminal “X2” VFD
10.02	Mengatur terminal “X3” VFD
10.03	Mengatur terminal “FWD” VFD
10.04	Mengatur terminal “REV” VFD
10.05	Mengatur Raspberry Pi
10.06	Mengatur LED
10.07	Mengatur <i>buzzer</i>

### 4. Mengunggah Program *Ladder Diagram* ke PLC

Lakukan pengunggahan program *ladder diagram* ke PLC. Proses pengunggahan akan mengubah PLC ke ”*Program Mode*” dan

setelah proses pengunggahan selesai, PLC akan dikembalikan ke “*Monitor Mode*”.

### 3.3.2 Raspberry Pi Model B+

Raspberry Pi merupakan komputer yang digunakan untuk menjalankan program utama. Raspberry Pi memerlukan *library* OpenCV 2.4.9. untuk menjalankan program. Selain itu juga dibutuhkan *driver* V4L modul kamera Raspberry Pi, dan *driver* sambungan komunikasi serial RS-232. Kemudian pindahkan *file* C++ program yang sudah dibuat dan diuji sebelumnya di PC ke Raspberry Pi. Lalu lakukan *compiling* dengan hasil *file* “TA.exe”.

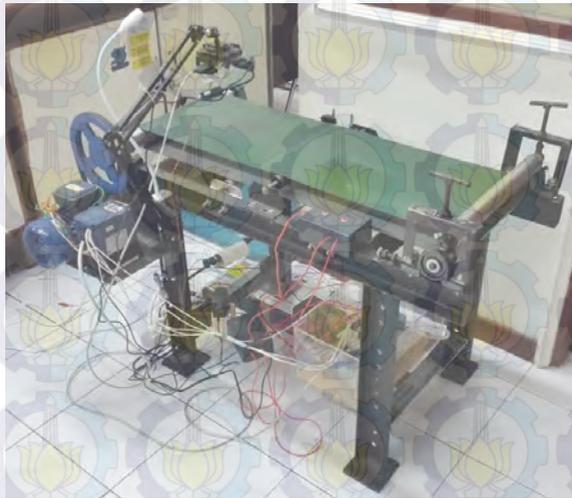
Raspberry Pi ini sudah diatur agar menjalankan program pendeteksi lubang pada kertas tisu langsung setelah *booting*. Operator hanya perlu menekan *pushbutton* “START” untuk menyalakan Raspberry Pi dan “STOP” untuk mematikan Raspberry Pi. Operasi *booting* sudah diubah sehingga setelah selesai *booting*, Raspberry Pi akan menjalankan sebuah *script* bernama “launcher.sh” yang akan mengaktifkan *driver* V4L modul kamera Raspberry Pi, membuka *folder* program, dan menjalankan program. *Script* ini dapat dilihat langsung di lampiran.

### 3.4 Cara Kerja Keseluruhan Sistem

Keseluruhan sistem dapat dilihat di Gambar 3.30 dan memiliki cara kerja keseluruhan sistem seperti di bawah ini.

1. Operator memasang kertas tisu di *conveyor* dan menjepitnya di antara dua rol.
2. Lalu aktifkan seluruh sumber daya untuk perangkat keras.
3. PLC akan menyalakan LED merah sebagai tanda bahwa sistem belum siap.
4. Kemudian tekan *pushbutton* “START” yang akan menyalakan Raspberry Pi melalui PLC.
5. Tunggu Raspberry Pi selesai *booting* dan program pendeteksi lubang pada kertas tisu akan berjalan sendiri.

6. Program ini akan melakukan inialisasi I/O, lalu menyalakan LED hijau dan mematikan LED merah melalui PLC sebagai tanda bahwa sistem sudah berjalan.
7. *Conveyor* akan menyala melalui PLC dan VFD dan kertas tisu akan tertarik.
8. Pengolahan citra pun diaktifkan.
9. Jika terdeteksi lubang, maka *buzzer* akan dinyalakan sesaat melalui PLC.
10. Untuk mematikan sistem ini cukup menekan *pushbutton* “STOP” yang mematikan Raspberry Pi dan seluruh proses.
11. Untuk keperluan lain, seperti memperbaiki program C++ di Raspberry Pi, operator dapat menekan tombol “q” di *keyboard* yang hanya mematikan program di Raspberry Pi saja tetapi sistem tidak mati seluruhnya.
12. Dengan menekan tombol “q”, maka LED hijau dan *conveyor* akan mati dan LED merah akan menyala sebagai tanda bahwa sistem belum siap digunakan.



**Gambar 3.30** Keseluruhan sistem

## BAB 4

### PENGUJIAN SISTEM

Pembahasan pada bab ini mencakup pengujian sistem pendeteksi lubang pada produk industri kertas tisu berbasis informasi visual dengan Raspberry Pi. Dalam bab ini pengujian akan dibagi menjadi dua bagian, yaitu pengujian masing-masing komponen dan perangkat penyusun sistem dan pengujian keseluruhan sistem.

#### 4.1 Pengujian Bagian-Bagian Penyusun Sistem

Pengujian ini menguji tiap bagian penting penyusun sistem. Pengujian ini dilakukan untuk memastikan tiap bagian sistem bekerja dengan baik sebelum dijadikan satu kesatuan. Dengan menguji tiap bagian penyusun sistem, kita dapat memastikan tiap bagian sistem bekerja dengan baik dan tidak akan merusak sistem ketika menguji keseluruhan sistem. Pengujian yang akan dilakukan di sub-bab ini adalah pengujian sumber daya DC, pembatas arus, *pushbutton*, *driver* Raspberry Pi Model B+, *driver Light Emitting Diode (LED)*, *driver buzzer*, *buzzer*, PLC, VFD, dan komunikasi serial RS-232.

##### 4.1.1 Pengujian Sumber Daya DC

Pengujian sumber daya DC merupakan pengujian tegangan dan arus yang dihasilkan oleh kedua pasang keluaran yang diatur agar menghasilkan keluaran positif. Sumber daya ini merupakan perangkat tambahan di sistem ini dan tidak dirancang sendiri. Hasil pengujian ini dapat dilihat di Tabel 4.1.

**Tabel 4.1** Hasil pengujian sumber daya DC

Keluaran	Tegangan	Arus
1	24 VDC	6,25 A
2	24 VDC	6,25 A

Tegangan dan arus yang dihasilkan oleh sumber daya DC sudah memenuhi permintaan sistem, sehingga sumber daya DC ini dapat digunakan.

#### 4.1.2 Pengujian Pembatas Arus

Pada pengujian pembatas arus dilakukan pengujian tegangan dan arus yang sudah dibatasi dari sumber daya DC agar tidak melebihi 50 mA. Hasil pengujian ini dapat dilihat selengkapnya di Tabel 4.2.

**Tabel 4.2** Hasil pengujian pembatas arus

Keluaran	Tegangan	Arus
1	24 VDC	24 mA
2	24 VDC	24 mA
3	24 VDC	24 mA

Arus yang dihasilkan oleh pembatas arus sudah memenuhi permintaan VFD yang mempunyai persyaratan arus masukan digital maksimal sebesar 50 mA. Arus ini juga tidak terlalu kecil dan masih cukup kuat memenuhi permintaan sistem.

#### 4.1.3 Pengujian *Pushbutton*

Pengujian *pushbutton* digunakan untuk melihat keluaran dari rangkaian ini yang akan digunakan sebagai masukan ke PLC. Pada pengujian ini digunakan sumber daya DC variable untuk menguji. Hasil pengujian selengkapnya dapat dilihat di Tabel 4.3.

**Tabel 4.3** Hasil pengujian *pushbutton*

Tegangan Masukan	<i>Pushbutton</i>	Kondisi <i>Pushbutton</i>	Tegangan Keluaran
24 VDC	1	Ditekan	24 VDC
24 VDC	1	Tidak Ditekan	0 VDC
24 VDC	2	Ditekan	24 VDC
24 VDC	2	Tidak Ditekan	0 VDC

Pengujian di atas sudah menunjukkan bahwa *pushbutton* sudah berjalan baik dan dapat digunakan sebagai masukan ke PLC di sistem.

#### 4.1.4 Pengujian *Driver* Raspberry Pi Model B+

Pengujian *driver* Raspberry Pi digunakan untuk melihat keadaan *relay* yang menyalakan atau mematikan Raspberry Pi. Pada pengujian ini digunakan sumber daya DC variabel. *Relay* memiliki dua kontak, yaitu *Normally Open* (NO) dan *Normally Closed* (NC), tetapi *driver* ini hanya menggunakan kontak NO saja. Hasil pengujian selengkapnya dapat dilihat di Tabel 4.4.

**Tabel 4.4** Hasil pengujian *driver* Raspberry Pi Model B+

Tegangan <i>Trigger</i>	Kondisi <i>Relay</i>	Kontak NO <i>Relay</i>	Kontak NC <i>Relay</i>	Keluaran <i>Driver</i>
0-14 VDC	Mati	Membuka	Menutup	0 VAC
15-24 VDC	Menyala	Menutup	Membuka	220 VAC

Pengujian ini sudah memuaskan kebutuhan untuk mematikan dan menyalakan Raspberry Pi pada sistem dan rangkaian dapat langsung digunakan.

#### 4.1.5 Pengujian *Driver* LED

Pengujian *driver* LED digunakan untuk melihat keadaan *relay* dan LED. Pada pengujian ini digunakan sumber daya DC variabel. *Relay* memiliki dua kontak, yaitu *Normally Open* (NO) dan *Normally Closed* (NC), *driver* ini hanya menggunakan kontak NC untuk LED merah dan kontak NO untuk LED hijau. Hasil pengujian selengkapnya dapat dilihat di Tabel 4.5.

**Tabel 4.5** Hasil pengujian *driver* LED

Tegangan <i>Trigger</i>	Kondisi <i>Relay</i>	Kontak NO <i>Relay</i>	Kontak NC <i>Relay</i>	LED yang Menyala
0-14 VDC	Mati	Membuka	Menutup	Merah
15-24 VDC	Menyala	Menutup	Membuka	Hijau

Pengujian menyatakan bahwa rangkaian ini sudah berfungsi dengan baik dan dapat digunakan sistem.

#### 4.1.6 Pengujian *Driver Buzzer*

Pengujian *driver buzzer* digunakan untuk melihat keadaan *relay* yang mengatur kerja *buzzer*. Pada pengujian ini digunakan sumber daya DC variabel. *Relay* memiliki dua kontak, yaitu *Normally Open* (NO) dan *Normally Closed* (NC), tetapi *driver* ini hanya menggunakan kontak NO saja. Hasil pengujian selengkapnya dapat dilihat di Tabel 4.6.

**Tabel 4.6** Hasil pengujian *driver buzzer*

Tegangan <i>Trigger</i>	Kondisi <i>Relay</i>	Kontak NO <i>Relay</i>	Kontak NC <i>Relay</i>	Keluaran <i>Driver</i>
0-14 VDC	Mati	Membuka	Menutup	0 VDC
15-24 VDC	Menyala	Menutup	Membuka	24 VDC

Pengujian di atas mendapatkan bahwa *driver buzzer* sudah berjalan baik dan dapat digunakan untuk mengendalikan *buzzer* di sistem.

#### 4.1.7 Pengujian Komponen *Buzzer*

Pada pengujian *buzzer* ini digunakan untuk menguji komponen *buzzer* yang memiliki dua kondisi *on* dan *off*. *Buzzer* ini akan diintegrasikan menjadi satu rangkaian dengan *driver*-nya. Pengujian ini menggunakan sumber daya DC variabel, untuk lebih lengkapnya dapat dilihat di Tabel 4.7.

**Tabel 4.7** Hasil pengujian komponen *buzzer*

Tegangan <i>Trigger</i>	Kondisi <i>Buzzer</i>
0 VDC	Mati
1-24 VDC	Menyala

Pada pengujian di atas kita mendapatkan bahwa *buzzer* yang diuji tidak sesuai dengan spesifikasi yang tertera. Spesifikasi yang tertera menyebutkan bahwa *buzzer* menyala dengan masukan 6-24 V, tetapi *buzzer* sudah menyala meskipun hanya diberi tegangan 1 V. Maka dari itu, *buzzer* dihubungkan dengan *driver buzzer* yang berisi *relay* sehingga dapat dimatikan dengan memutus aliran listriknya.

#### 4.1.8 Pengujian PLC

Sistem ini hanya menggunakan keluaran dari PLC sebagai fungsi pengendali sistem dan tidak menggunakan masukannya. PLC ini dikendalikan melalui komunikasi serial dari Raspberry Pi yang akan diuji di sub-bab 4.1.7. Pada uji coba PLC diatur dengan konfigurasi *sourcing* dan pin *common*-nya dihubungkan dengan tegangan +24 V. Hasil pengujian dapat dilihat di Tabel 4.8.

**Tabel 4.8** Hasil pengujian PLC

Tegangan <i>Common</i>	Arus <i>Common</i>	Alamat Keluaran	Logika PLC	Tegangan Keluaran	Arus Keluaran
24 VDC	24 mA	10.00	0	5 VDC	10 mA
24 VDC	24 mA	10.01	0	5 VDC	10 mA
24 VDC	24 mA	10.02	0	5 VDC	10 mA
		10.03	0	5 VDC	10 mA
		10.04	0	5 VDC	10 mA
		10.05	0	5 VDC	10 mA
24 VDC	6,25 A	10.06	0	5 VDC	10 mA
		10.07	0	5 VDC	10 mA
		10.00	1	24 VDC	24 mA
24 VDC	24 mA	10.01	1	24 VDC	24 mA
24 VDC	24 mA	10.02	1	24 VDC	24 mA
		10.03	1	24 VDC	24 mA
		10.04	1	24 VDC	24 mA
		10.05	1	24 VDC	2 A
24 VDC	6,25 A	10.06	1	24 VDC	2 A
		10.07	1	24 VDC	2 A

Dari hasil pengujian yang diambil dapat disimpulkan bahwa spesifikasi keluaran PLC sudah sesuai dengan yang diinginkan dan dapat digunakan untuk mengendalikan perangkat-perangkat lain pada sistem.

#### 4.1.9 Pengujian VFD

Pada pengujian ini VFD sudah diatur agar dapat menghasilkan keluaran dengan beberapa frekuensi dan arah putaran dengan sinyal kontrol digital dari PLC. Masukan digital ini berlogika '0' saat keluaran PLC bertegangan 5 V dan berlogika '1' saat keluaran PLC bertegangan 24 V. Hasil dari pengujian ini dapat dilihat di Tabel 4.9.

**Tabel 4.9** Hasil pengujian VFD

Logika Terminal Digital VFD					Frekuensi Keluaran	Arah Putaran Motor	Kecepatan Motor
FWD	REV	X1	X2	X3			
0	0	0	0	0	0 Hz	Diam	0 RPM
0	0	0	0	1	0 Hz	Diam	0 RPM
0	0	0	1	0	0 Hz	Diam	0 RPM
0	0	0	1	1	0 Hz	Diam	0 RPM
0	0	1	0	0	0 Hz	Diam	0 RPM
0	0	1	0	1	0 Hz	Diam	0 RPM
0	0	1	1	0	0 Hz	Diam	0 RPM
0	0	1	1	1	0 Hz	Diam	0 RPM
0	1	0	0	0	0 Hz	Diam	0 RPM
0	1	0	0	1	5 Hz	Mundur	100 RPM
0	1	0	1	0	10 Hz	Mundur	200 RPM
0	1	0	1	1	15 Hz	Mundur	300 RPM
0	1	1	0	0	20 Hz	Mundur	400 RPM
0	1	1	0	1	30 Hz	Mundur	600 RPM
0	1	1	1	0	40 Hz	Mundur	800 RPM
0	1	1	1	1	50 Hz	Mundur	1000 RPM
1	0	0	0	0	0 Hz	Diam	0 RPM
1	0	0	0	1	5 Hz	Maju	100 RPM
1	0	0	1	0	10 Hz	Maju	200 RPM

Logika Terminal Digital VFD					Frekuensi Keluaran	Arah Putaran Motor	Kecepatan Motor
FWD	REV	X1	X2	X3			
1	0	0	1	1	15 Hz	Maju	300 RPM
1	0	1	0	0	20 Hz	Maju	400 RPM
1	0	1	0	1	30 Hz	Maju	600 RPM
1	0	1	1	0	40 Hz	Maju	800 RPM
1	0	1	1	1	50 Hz	Maju	1000 RPM
1	1	0	0	0	0 Hz	Diam	0 RPM
1	1	0	0	1	0 Hz	Diam	0 RPM
1	1	0	1	0	0 Hz	Diam	0 RPM
1	1	0	1	1	0 Hz	Diam	0 RPM
1	1	1	0	0	0 Hz	Diam	0 RPM
1	1	1	0	1	0 Hz	Diam	0 RPM
1	1	1	1	0	0 Hz	Diam	0 RPM
1	1	1	1	1	0 Hz	Diam	0 RPM

Dari hasil pengujian di atas didapatkan bahwa hasil keluaran dari VFD sudah memenuhi ketentuan yang diinginkan dan dapat digunakan di sistem.

#### 4.1.10 Pengujian Komunikasi Serial RS-232

Pengujian ini menggunakan aplikasi Code::Blocks dengan program tersendiri yang dapat dilihat di lampiran dari PC untuk mengirimkan *command block* ke PLC serta mendengarkan *response block*-nya. Pada pengujian ini kita menguji apakah proses penulisan di area IR pada PLC berhasil atau tidak dengan melihat *response block* dari PLC. Hasil pengujian selengkapnya dapat dilihat di Tabel 4.10.

**Tabel 4.10** Hasil pengujian komunikasi serial RS-232

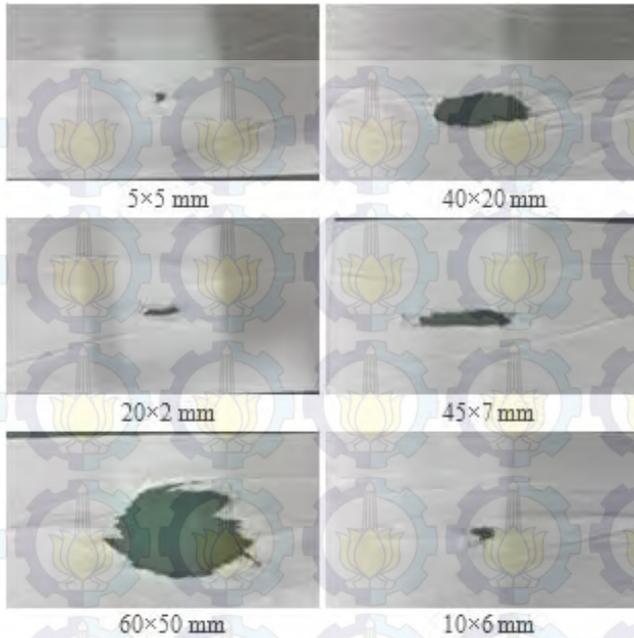
Alamat PLC	Command Block	Response Block
IR 200	@00WR0200000146*↵	@00WR0045*↵
IR 200	@00WR0200000245*↵	@00WR0045*↵
IR 201	@00WR0201000046*↵	@00WR0045*↵
IR 201	@00WR0201000147*↵	@00WR0045*↵
IR 201	@00WR0201000244*↵	@00WR0045*↵

Alamat PLC	Command Block	Response Block
IR 202	@00WR0202000045*↵	@00WR0045*↵
IR 202	@00WR0202000144*↵	@00WR0045*↵
IR 202	@00WR0202000247*↵	@00WR0045*↵
IR 203	@00WR0203000044*↵	@00WR0045*↵
IR 203	@00WR0203000145*↵	@00WR0045*↵
IR 203	@00WR0203000246*↵	@00WR0045*↵
IR 204	@00WR0204000043*↵	@00WR0045*↵
IR 204	@00WR0204000142*↵	@00WR0045*↵
IR 204	@00WR0204000241*↵	@00WR0045*↵
IR 205	@00WR0205000042*↵	@00WR0045*↵
IR 205	@00WR0205000143*↵	@00WR0045*↵
IR 205	@00WR0205000240*↵	@00WR0045*↵
IR 206	@00WR0206000041*↵	@00WR0045*↵
IR 206	@00WR0206000140*↵	@00WR0045*↵
IR 206	@00WR0206000243*↵	@00WR0045*↵

Dari hasil pengujian di atas didapatkan bahwa komunikasi serial berjalan dengan sempurna dan sudah dapat digunakan untuk keperluan sistem. Seperti pada *command block* @00WR020000146\*↵, instruksi 00WR02000001 ini berarti kita akan mengakses PLC pertama dan menulis data di area IR 200 sebesar 1 bit. *Response block* yang didapatkan berupa @00WR0045\*↵ di mana instruksi 00WR00 berarti pengaksesan PLC pertama dan proses penulisan berjalan lancar.

#### 4.2 Pengujian Keseluruhan Sistem Terhadap Kecepatan Conveyor

Pengujian ini merupakan pengujian keseluruhan sistem yang disusun oleh berbagai perangkat. Pertama kita menyambungkan PLC dengan sumber daya DC, pembatas arus, VFD, dan *driver buzzer*. Lalu sambungkan Raspberry Pi yang sudah disertai modul kamera dengan PLC melalui RS-232. Kemudian aktifkan seluruh sistem dan jalankan *driver* modul kamera di Raspberry Pi Model B+. Lalu mulai program pendeteksi lubang di Raspberry Pi.



**Gambar 4.1** Jenis dan ukuran lubang

Ketika program dijalankan, maka pendeteksi lubang akan aktif. Jika kita menekan tombol ‘a’ di *keyboard* maka *conveyor* akan bergulir dan *buzzer* akan menyala jika mendeteksi lubang. Dalam pengujian ini kita menguji dengan 6 ukuran lubang yang berbeda dan di 7 macam kecepatan *conveyor* yang berbeda. Untuk rincian bentuk lubangnya dapat Gambar 4.1.

Dari pengujian ini kita melakukan pengujian terhadap keenam lubang di atas dan dijalankan di tujuh kecepatan yang berbeda. Pengujian dilakukan tiga kali untuk tiap jenis pengujian sehingga didapatkan data yang lebih akurat. Pengujian ini dimaksudkan untuk mengukur akurasi sistem di berbagai kecepatan dan jenis lubang sehingga kita mendapatkan kecepatan yang optimal untuk sistem. Hasil pengujian dapat dilihat di Tabel 4.11.

**Tabel 4.11** Hasil pengujian keseluruhan sistem terhadap kecepatan conveyor

Kecepatan Motor	Kecepatan Conveyor	Ukuran Lubang (mm <sup>2</sup> )	Pengujian 1	Pengujian 2	Pengujian 3
100 RPM	0,07 m/s	≈25	✓	✓	✓
		≈800	✓	✓	✓
		≈40	✓	✓	✓
		≈315	✓	✓	✓
		≈3000	✓	✓	✓
		≈60	✓	✓	✓
200 RPM	0,13 m/s	≈25	✓	✓	✓
		≈800	✓	✓	✓
		≈40	✓	✓	✓
		≈315	✓	✓	✓
		≈3000	✓	✓	✓
		≈60	✓	✓	✓
300 RPM	0,2 m/s	≈25	✓	✓	✓
		≈800	✓	✓	✓
		≈40	✓	✓	✓
		≈315	✓	✓	✓
		≈3000	✓	✓	✓
		≈60	✓	✓	✓
400 RPM	0,27 m/s	≈25	✓	✓	✓
		≈800	✓	✓	✓
		≈40	✓	✓	✓
		≈315	✓	✓	✓
		≈3000	✓	✓	✓
		≈60	✓	✓	✗
600 RPM	0,4 m/s	≈25	✓	✓	✓
		≈800	✓	✗	✓
		≈40	✓	✓	✓
		≈315	✓	✓	✓
		≈3000	✓	✓	✓

Kecepatan Motor	Kecepatan Conveyor	Ukuran Lubang (mm)	Pengujian 1	Pengujian 2	Pengujian 3
600 RPM	0,4 m/s	≈60	✓	✓	✗
		≈25	✓	✓	✓
800 RPM	0,53 m/s	≈800	✓	✗	✗
		≈40	✓	✓	✓
		≈315	✗	✗	✗
		≈3000	✓	✓	✓
		≈60	✗	✓	✗
		≈25	✗	✗	✗
1000 RPM	0,66 m/s	≈800	✓	✗	✗
		≈40	✗	✓	✗
		≈315	✗	✗	✗
		≈3000	✓	✓	✓
		≈60	✗	✗	✗
		≈25	✗	✗	✗

Dari hasil pengujian dapat disimpulkan bahwa saat kecepatan conveyor ditingkatkan maka *error* yang didapat makin banyak pula jumlahnya. Demi kemudahan pembacaan data, maka kita bisa melihat tingkat *error* dalam persentase. Persentase *error* dapat dihitung dengan rumus:

$$error = \frac{n_{error}}{n_{total}} \times 100\%$$

Keterangan:

*error* = persentase *error* (%)

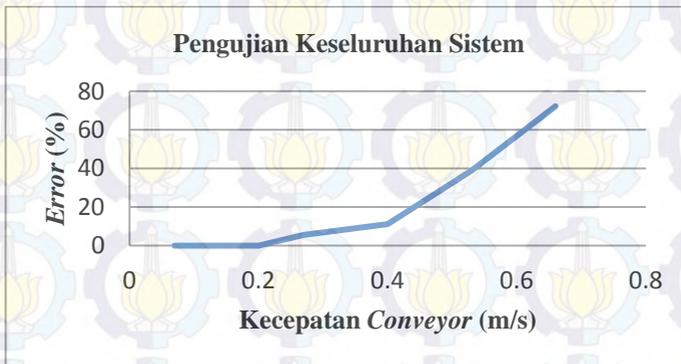
$n_{error}$  = jumlah *error* pendeteksian

$n_{total}$  = jumlah pendeteksian seharusnya

Dari rumus di atas maka persentase *error* pengujian di tiap kecepatan dapat dihitung dan dibuat grafik demi kemudahan pembacaan seperti pada Gambar 4.2. Hasil persentase *error* selengkapnya dapat dilihat di Tabel 4.12.

**Tabel 4.12** Persentase *error* pengujian keseluruhan sistem terhadap kecepatan *conveyor*

Kecepatan Motor	Kecepatan <i>Conveyor</i>	<i>Error</i> (%)
100 RPM	0,07 m/s	0
200 RPM	0,13 m/s	0
300 RPM	0,2 m/s	0
400 RPM	0,27 m/s	5,55
600 RPM	0,4 m/s	11,11
800 RPM	0,53 m/s	38,89
1000 RPM	0,66 m/s	72,22



**Gambar 4.2** Grafik persentase *error* pengujian keseluruhan sistem terhadap kecepatan *conveyor*

Jika dilihat dari tabel di atas maka dapat dilihat bahwa *error* mulai terjadi di kecepatan 0,27 m/s. *Error* ini terjadi karena *delay* dari terbatasnya kekuatan pemrosesan di Raspberry Pi sehingga membatasi kecepatan pendeteksian sistem. Sistem dapat bekerja optimal jika kertas tisu digulung dengan kecepatan di bawah 0,27 m/s. Jika kecepatan penggulungan melewati 0,27 m/s, maka mulai terjadi *error* dan sistem tidak bisa bekerja secara *real-time*.

## BAB 5

### PENUTUP

#### 5.1 Kesimpulan

Setelah pengujian dilakukan, dari hasilnya dapat ditarik kesimpulan sebagai berikut:

1. Nilai batas (*threshold*) gradien yang digunakan di fungsi pendeteksi tepian Canny pada program untuk mendeteksi lubang adalah  $90^\circ$  untuk nilai batas bawah dan  $255^\circ$  untuk nilai batas atas.
2. Sistem dapat mendeteksi lubang dengan optimal secara *real-time* jika kecepatan translasi *conveyor* berada di bawah 0,27 m/s.

#### 5.2 Saran

Ada pula saran mengenai Tugas Akhir ini untuk pengembangan ke depannya, yaitu:

1. Penggunaan nilai batas adaptif (*adaptive threshold*) di fungsi pendeteksi tepian Canny untuk mendeteksi lubang<sup>[30]</sup>.
2. Penggabungan lebih dari satu SBC sehingga menjadi komputer yang lebih kuat dan dapat mempercepat kecepatan pemrosesan.
3. Penambahan *heatsink* dan kipas angin pada Raspberry Pi agar kinerja sistem tidak melambat ketika prosesor menjadi terlalu panas akibat *overclock*.
4. Penambahan *Human-Machine Interface* (HMI) sebagai tampilan program agar lebih mudah digunakan oleh operator.
5. Penambahan program *data logger* agar memudahkan operator untuk mengecek ulang produk kertas tisu.
6. Penambahan perangkat yang dapat menandai gulungan kertas tisu setelah proses penggulungan selesai jika terdeteksi lubang pada satu gulungan kertas tisu tersebut.

## DAFTAR PUSTAKA

1. Edwards, C., "Not-So-Humble Raspberry Pi Gets Big Ideas", *Engineering & Technology*, Vol. 8, No. 3, pp. 30-33, April, 2013
2. Bertini, F. dan Rommel, C., "Can the Raspberry Pi Revolutionize the Low-End Industrial Computing Space?", VDC Research, April, 2014
3. Paulapuro, H., "*Paper and Board Grades*", TAPPI, Atlanta, 2000
4. Muir, F., "*An Irrelevant and Almost Complete Social History of the Bathroom*", Stein and Day, New York, 1983
5. Purnomo, H., "*Pengantar Teknik Industri*", Graha Ilmu, Yogyakarta, 2004
6. Rosch, W.L., "*Hardware Bible, Sixth Edition*", Que, Indianapolis, Ch. 2, 2003
7. Raspberry Pi Foundation, "Raspberry Pi FAQs" <URL: <http://www.raspberrypi.org/help/faqs/>>, Mei, 2014
8. Upton, E., "New Product Launch! Introducing Raspberry Pi Model B+" <URL: <http://www.raspberrypi.org/introducing-raspberry-pi-model-b-plus/>>, Juli, 2014
9. Raspberry Pi Foundation, "RaspiCam Documentation" <URL: <http://www.raspberrypi.org/wp-content/uploads/2013/07/RaspiCam-Documentation.pdf>>, Juli, 2013
10. Upton, E., "Camera Board Available for Sale!" <URL: <http://www.raspberrypi.org/camera-board-available-for-sale/>>, Mei, 2013
11. eLinux.org, "RPi Camera Module" <URL: [http://elinux.org/Rpi\\_Camera\\_Module](http://elinux.org/Rpi_Camera_Module)>, Juni, 2014
12. Maher, M.J., "Real-Time Control and Communications", Proc. of the 18th Annual ESD/SMI International Programmable Controllers Conference and Exposition, pp. 431-436, Detroit, April, 1989
13. OMRON, "**SYSMAC CPM1/CPM1A/CPM2A/CPM2C/SRM1(-V2) Programmable Controller Programming Manual**", OMRON, 2000

14. NEMA Standards Publication, “*Application Guide for AC Adjustable Speed Drive System*”, National Electrical Manufacturers Association, Rosslyn, 2007
15. Fuji Electric, “**FRENIC-Mini Compact Inverter Instruction Manual**”, Fuji Electric, 2002
16. Mason, C. R., “Art and Science of Protective Relaying, Ch. 2” <URL: <http://www.gedigitalenergy.com/multilin/notes/artsci/>>, November, 2014
17. OpenCV, “Load, Modify, and Save an Image” <URL: [http://docs.opencv.org/doc/tutorials/introduction/load\\_save\\_image/load\\_save\\_image.html](http://docs.opencv.org/doc/tutorials/introduction/load_save_image/load_save_image.html)>, Juli, 2014
18. OpenCV, “Smoothing Images” <URL: [http://docs.opencv.org/doc/tutorials/imgproc/gaussian\\_median\\_blur\\_bilateral\\_filter/gaussian\\_median\\_blur\\_bilateral\\_filter.html](http://docs.opencv.org/doc/tutorials/imgproc/gaussian_median_blur_bilateral_filter/gaussian_median_blur_bilateral_filter.html)>, Juli, 2014
19. OpenCV, “Histogram Equalization” <URL: [http://docs.opencv.org/doc/tutorials/imgproc/histograms/histogram\\_equalization/histogram\\_equalization.html](http://docs.opencv.org/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html)>, September 2014
20. OpenCV, “Canny Edge Detector” <URL: [http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny\\_detector/canny\\_detector.html](http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html)>, Agustus, 2014
21. Canny, J., “A Computational Approach to Edge Detection”, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Vol. PAMI-8, No. 6, pp. 679-698, Massachusetts, November, 1986
22. OpenCV, “Eroding and Dilating” <URL: [http://docs.opencv.org/doc/tutorials/imgproc/erosion\\_dilatation/erosion\\_dilatation.html](http://docs.opencv.org/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html)>, Juli, 2014
23. OpenCV, “More Morphology Transformations” <URL: [http://docs.opencv.org/doc/tutorials/imgproc/opening\\_closing\\_hats/opening\\_closing\\_hats.html](http://docs.opencv.org/doc/tutorials/imgproc/opening_closing_hats/opening_closing_hats.html)>, Juli, 2014
24. OpenCV, “Structural Analysis and Shape Descriptors” <URL: [http://docs.opencv.org/modules/imgproc/doc/structural\\_analysis\\_and\\_shape\\_descriptors.html](http://docs.opencv.org/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html)>, September, 2014

25. Suzuki, S. dan Abe, K., "Topological Structural Analysis of Digitized Binary Images by Border Following", *Computer Vision, Graphics, and Image Processing*, Vol. 30, No. 1, pp. 32-46, Hamamatsu, April, 1985
26. Engineering Dept., "*EIA Standard RS-232-C: Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange*", Electronic Industries Association, Washington, 1969
27. Pahl, G., Beitz, W., Feldhusen, J., dan Groten K.H., "*Engineering Design: A Systematic Approach, Third Edition*", Diterjemahkan dan diedit oleh Wallace, K. dan Blessing, L., Springer Verlag, London, 2007
28. OMRON, "**SYSMAC CPM2A Programmable Controller Operation Manual**", OMRON, 2005
29. Van Beelen, T., "RS-232 for Linux, FreeBSD, and Windows" <URL: <http://www.teuniz.net/RS-232/>>, September, 2014
30. Otsu, N., "A Threshold Selection Method from Gray-Level Histograms", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, pp. 62-66, Tokyo, Januari, 1979

## LAMPIRAN

### Program C++ Raspberry Pi

```
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>

#include <iostream>
#include <stdlib.h>
#include <stdio.h>
#include "rs232.h"
#include "rs232.c"

using namespace cv;
using namespace std;

///Image Processing Program Variables
Mat src, src_gray, src_blur, src_hist, src_erode, src_dilate, src_canny;
int lowThreshold = 90;
int highThreshold = 255;
int erosion_elem = 0;
int erosion_size = 0;
int erosion_type;
int dilation_elem = 0;
int dilation_size = 10;
int dilation_type;
vector<vector<Point>> contours;
vector<Vec4i> hierarchy;
int k = 0;

///Serial Communication Program Variables
int i, n, cport_nr=16, bdrate=9600;
unsigned char buf[4096];
```

```
char mode[]={'7','E','2',0},str[20][512];
```

```
int main(int, char**)
{
    strcpy(str[0], "@00WR0200000146*r");//It turns the buzzer on.
    strcpy(str[1], "@00WR0200000245*r");//It turns the buzzer off.
    strcpy(str[2], "@00WR0201000147*r");//It turns the Green LED on.
    strcpy(str[3], "@00WR0201000244*r");//It turns the Green LED
off.
    strcpy(str[4], "@00WR0202000144*r");//It turns the X1 on.
    strcpy(str[5], "@00WR0202000247*r");//It turns the X1 off.
    strcpy(str[6], "@00WR0203000145*r");//It turns the X2 on.
    strcpy(str[7], "@00WR0203000246*r");//It turns the X2 off.
    strcpy(str[8], "@00WR0204000142*r");//It turns the X3 on.
    strcpy(str[9], "@00WR0204000241*r");//It turns the X3 off.
    strcpy(str[10], "@00WR0205000143*r");//It runs the motor
forward.
    strcpy(str[11], "@00WR0205000240*r");//It stops the motor.
    strcpy(str[12], "@00WR0206000140*r");//It runs the motor
reversed.
    strcpy(str[13], "@00WR0206000243*r");//It stops the motor.
    strcpy(str[14], "@00WR0201000046*r");
    strcpy(str[15], "@00WR0202000045*r");
    strcpy(str[16], "@00WR0203000044*r");
    strcpy(str[17], "@00WR0204000043*r");
    strcpy(str[18], "@00WR0205000042*r");
    strcpy(str[19], "@00WR0206000041*r");

    ///Open Serial Communication
    if(RS232_OpenComport(cport_nr, bdrate, mode))
    {
        printf("Can Not Open COM Port\n\r");
    }

    return 1;
}
```

```
}
```

```
///Turn On Optimization  
setUseOptimized(true);
```

```
///Capture Video  
VideoCapture cap(0);
```

```
cap.set(CV_CAP_PROP_FRAME_WIDTH, 256);  
cap.set(CV_CAP_PROP_FRAME_HEIGHT, 208);
```

```
if(!cap.isOpened())  
    return 2;
```

```
///Turn Green LED On  
RS232_cputs(cport_nr, str[2]);  
RS232_cputs(cport_nr, str[14]);  
///Run Motor Forward at 10 Hz  
RS232_cputs(cport_nr, str[5]);  
RS232_cputs(cport_nr, str[15]);  
RS232_cputs(cport_nr, str[6]);  
RS232_cputs(cport_nr, str[16]);  
RS232_cputs(cport_nr, str[9]);  
RS232_cputs(cport_nr, str[17]);  
RS232_cputs(cport_nr, str[10]);  
RS232_cputs(cport_nr, str[18]);  
RS232_cputs(cport_nr, str[13]);  
RS232_cputs(cport_nr, str[19]);
```

```
for (;;)   
{  
    ///Retrieve Frame  
    cap >> src;  
    //imshow("Source", src);
```

```

///Convert to Gray
cvtColor( src, src_gray, CV_BGR2GRAY );
//imshow("Gray",src_gray);

///Smooth
GaussianBlur( src_gray, src_blur, Size(9, 9), 0, 0);
//imshow("Blurred",src_blur);

///Equalize Histogram
equalizeHist( src_blur, src_hist );
//imshow("Histogram Equalized",src_hist);

///Apply Canny Edge Detection
Canny(src_hist, src_canny, lowThreshold, highThreshold, 3,
true);
//imshow( "Canny Edge Detected", src_canny );

///Erode
if( erosion_elem == 0 ){ erosion_type = MORPH_RECT; }
else if( erosion_elem == 1 ){ erosion_type = MORPH_CROSS;
}
else if( erosion_elem == 2) { erosion_type =
MORPH_ELLIPSE; }
Mat element1 = getStructuringElement( erosion_type, Size(
2*erosion_size + 1, 2*erosion_size+1 ), Point( erosion_size,
erosion_size ) );
erode( src_canny, src_erode, element1 );
//imshow("Eroded", src_erode);

///Dilate
if( dilation_elem == 0 ){ dilation_type = MORPH_RECT; }
else if( dilation_elem == 1 ){ dilation_type =
MORPH_CROSS; }

```

```

else if( dilation_elem == 2) { dilation_type =
MORPH_ELLIPSE; }
Mat element2 = getStructuringElement( dilation_type,Size(
2*dilation_size + 1, 2*dilation_size+1 ),Point( dilation_size,
dilation_size ));
dilate(src_erode,src_dilate,element2);
//imshow("Dilated", src_dilate);

//Find Contours
findContours( src_dilate, contours, hierarchy,
CV_RETR_EXTERNAL, CV_CHAIN_APPROX_SIMPLE, Point(0, 0)
);

// Find the Ellipses for Each Contour & Send Data to PLC for
the Buzzer System
vector<RotatedRect> minEllipse( contours.size() );
int count=0;

for( int i = 0; i < contours.size(); i++ )
{
int a=0;
if( contours[i].size() > 6)
{
minEllipse[i] = fitEllipse( Mat(contours[i]) );
imwrite(format("defect_data/defect%02d.jpg", k), src);
cout<<"Number of Contour = "<<count<<endl;
cout<<"Defects Detected"<<endl;
cout<<"Contour Size = "<<contours[i].size()<<endl;
RS232_cputs(cport_nr, str[0]);///Buzzer On
printf("Sent: %s\n\r", str[0]);
RS232_cputs(cport_nr, str[1]);///Buzzer Off
printf("Sent: %s\n\r", str[1]);
k++;
a=1;
}
}

```

```

        count++;
    }
    if(a==0)
    {
        count=0;
        cout<<"Number of Contour = "<<count<<endl;
        cout<<"No Defects Detected"<<endl;
        RS232_cputs(cport_nr, str[1]);///Buzzer Off
        printf("Sent: %s\n\r", str[1]);
    }
}

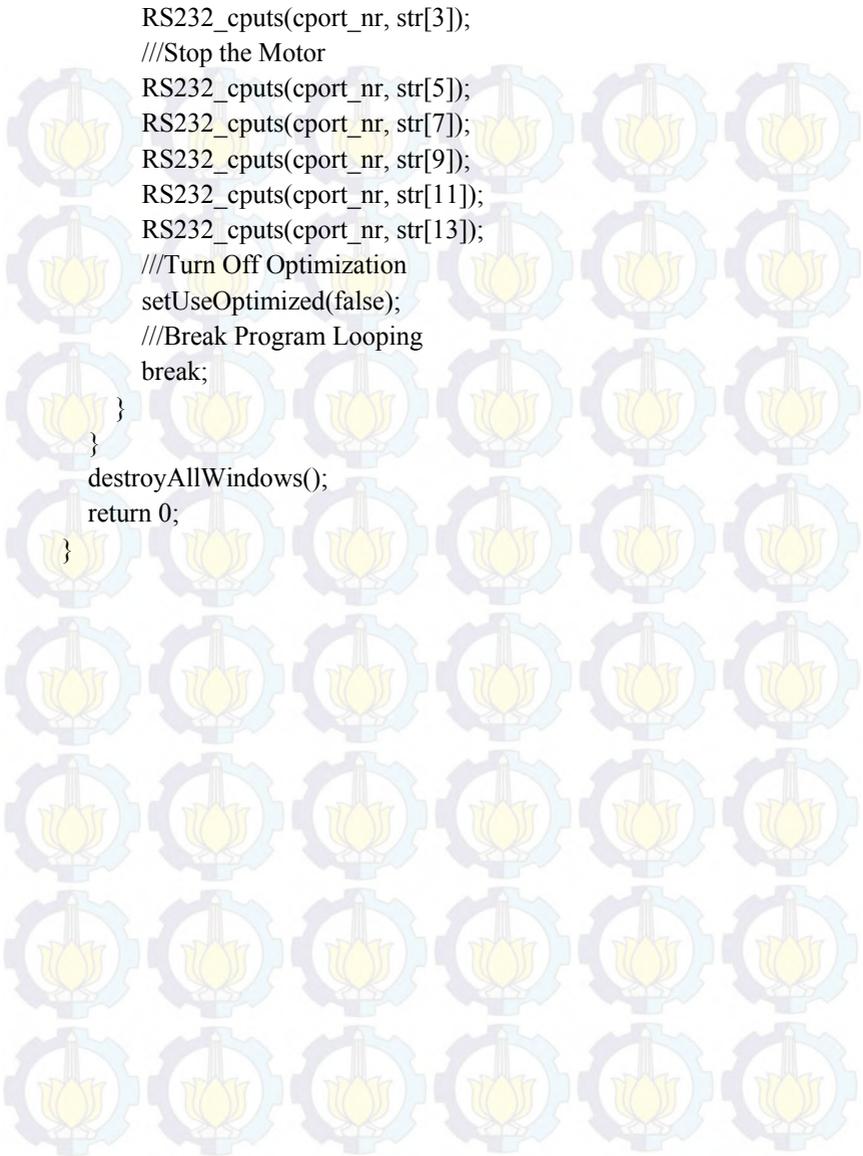
///Draw Contours and Ellipses
Mat drawing = Mat::zeros( src_dilate.size(), CV_8UC3 );
for( int i = 0; i< contours.size(); i++ )
{
    Scalar color = Scalar(255, 255, 0);
    Scalar color1 = Scalar(255, 255, 255);
    ///Contours
    drawContours( drawing, contours, i, color1, 1, 8, hierarchy, 0,
Point() );
    ///Ellipses
    ellipse( drawing, minEllipse[i], color, 1, 8 );
}

///Show Contours and Ellipses
imshow( "Contours Found", drawing );

///Break the Program
if( waitKey( 10 ) == 'q' )
{
    ///Turn the Buzzer Off
    RS232_cputs(cport_nr, str[1]);
    ///Turn the Green LED Off
}

```

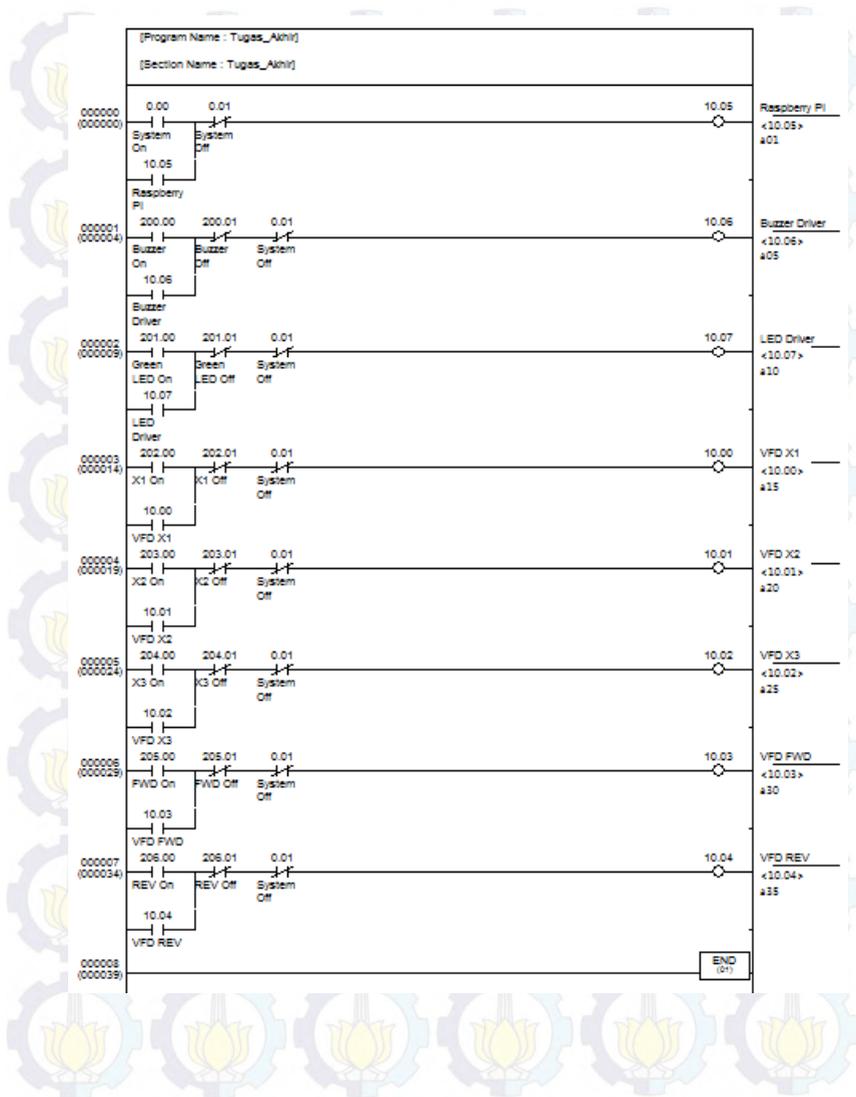
```
RS232_cputs(cport_nr, str[3]);
//Stop the Motor
RS232_cputs(cport_nr, str[5]);
RS232_cputs(cport_nr, str[7]);
RS232_cputs(cport_nr, str[9]);
RS232_cputs(cport_nr, str[11]);
RS232_cputs(cport_nr, str[13]);
//Turn Off Optimization
setUseOptimized(false);
//Break Program Looping
break;
}
}
destroyAllWindows();
return 0;
}
```



## ***Script Raspberry Pi***

```
sudo pkill uv4l && uv4l --driver raspicam --auto-video_nr --encoding  
yuv420 --width 256 --height 208 --nopreview && export  
LD_PRELOAD=/usr/lib/uv4l/uv4ltext/armv6l/libuv4ltext.so  
cd /  
cd /home/pi/TA  
./TA
```

## Program Ladder Diagram PLC



## Program C++ Pengujian Komunikasi Serial RS-232

```
#include <stdlib.h>
#include <stdio.h>

#ifdef _WIN32
    #include <Windows.h>
#else
    #include <unistd.h>
#endif

#include "rs232.h"
#include "rs232.c"

using namespace std;

int main()
{
    int i, n,
        cport_nr=5, /* /dev/ttyS5 (COM6 on windows) */
        bdrate=9600; /* 9600 baud */

    unsigned char buf[4096];
    char mode[]={'7','E','2',0},
        str[4][512];
    strcpy(str[0], "@00WR0200000146*r");

    if(RS232_OpenComport(cport_nr, bdrate, mode))
    {
        printf("Can not open comport\n");

        return(0);
    }
}
```

```
}
```

```
RS232_cputs(cport_nr, str[0]);  
printf("sent: %s\n", str[0]);
```

```
while(1)
```

```
{
```

```
    n = RS232_PollComport(cport_nr, buf, 4095);
```

```
    if(n > 0)
```

```
    {
```

```
        buf[n] = 0; /// always put a "null" at the end of a string!
```

```
        for(i=0; i < n; i++)
```

```
        {
```

```
            if(buf[i] < 32) /// replace unreadable control-codes by dots
```

```
            {
```

```
                buf[i] = '.';
```

```
            }
```

```
        }
```

```
        printf("received %i bytes: %s\n", n, (char *)buf);
```

```
    }
```

```
    #ifdef _WIN32
```

```
        Sleep(100);
```

```
    #else
```

```
        usleep(100000); /// sleep for 100 milliSeconds
```

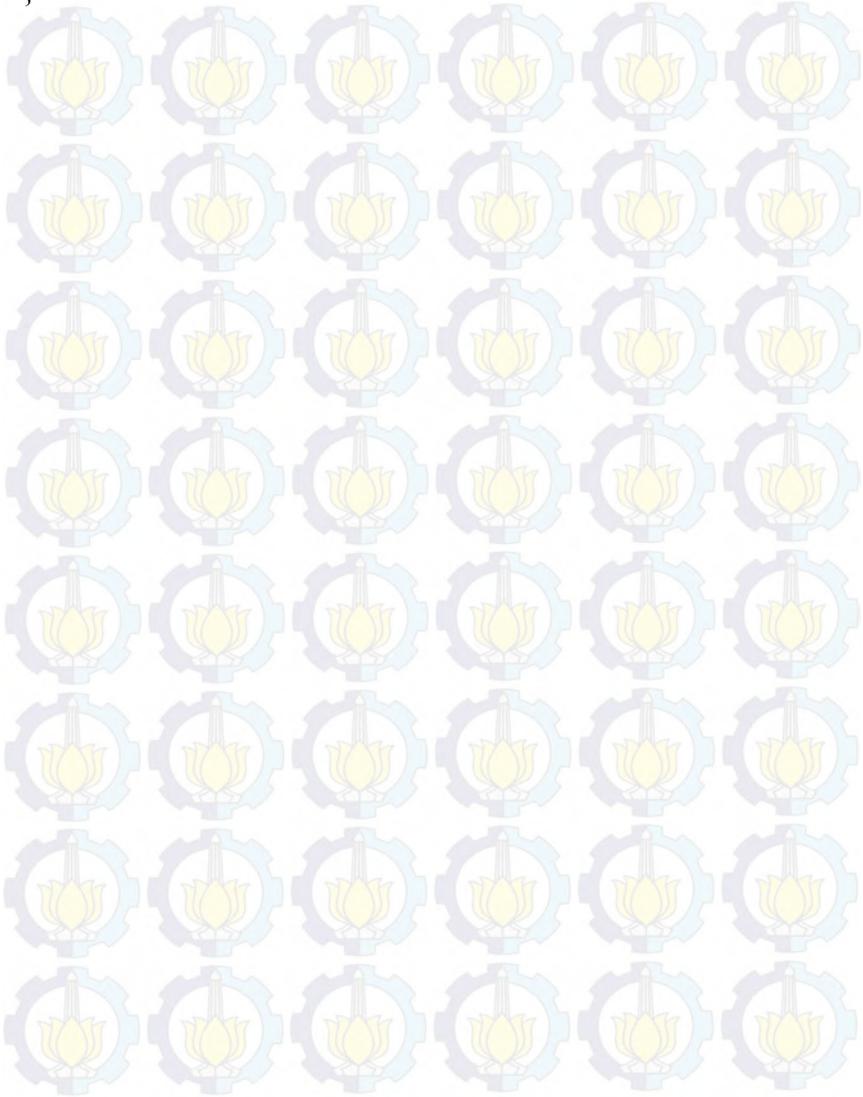
```
    #endif
```

```
    i++;
```

```
    i %= 2;
```

```
}
```

```
return(0);  
}
```



## BIODATA PENULIS



Penulis bernama Andre Raditya Mutalip. Lahir di kota Malang pada tanggal 20 Agustus 1992. Penulis merupakan anak pertama dari pasangan Abdul Mutalip dan Tinamarinda Noegroho. Beralamat di Jl. Venice 5 No. 6, Pantai Indah Kapuk, Jakarta Utara. Mengawali pendidikan di PG Kusuma Bangsa Surabaya, dilanjutkan ke TK Santa Maria Surabaya, lalu SD Santa Maria Surabaya, dan SMP Santa Maria Surabaya. Kemudian pindah ke Jakarta dan melanjutkan pendidikan di SMP Tarakanita 2 Jakarta Utara, dan dilanjutkan ke SMA Tarakanita 2 Jakarta Utara. Sekarang penulis masih menempuh pendidikan S1 di Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya dengan bidang studi Elektronika.