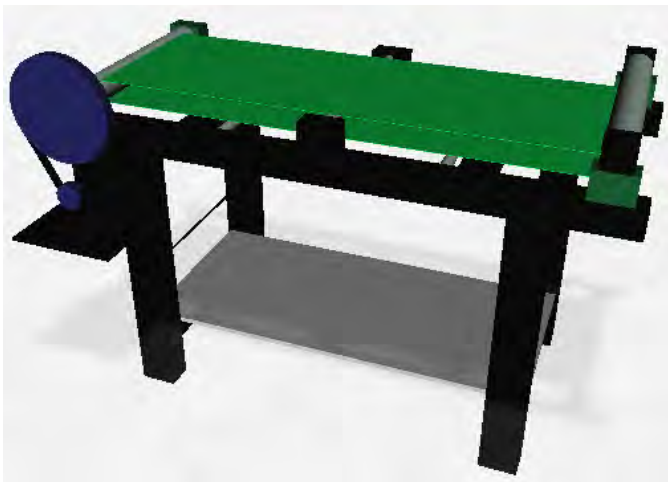






rol di ujungnya dengan *bearing* untuk meletakkan sabuk karet *conveyor*. Salah satu rol bersifat tetap dan tersambung dengan sistem *belt* dan *pulley* yang digunakan sebagai penggerak *conveyor*, sedangkan rol lainnya dapat digeser untuk mengatur kekencangan sabuk *conveyor*. Lalu ada tiga rol yang lebih kecil dengan *bearing* yang digunakan untuk memuluskan jalannya sabuk. Berikut ini merupakan gambar dari perancangan mekanik.

Lalu ditambahkan rak sistem penggerak, digunakan plat besi baja 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*. Kemudian ada juga rol tambahan untuk menjepit kertas tisu dan batang besi untuk meletakkan kertas tisu



Gambar 4. Rancangan mekanik keseluruhan.

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*. Kedua *pulley* ini berdiameter 20,32 cm dan 5,08 cm dengan jarak antara kedua porosnya sejauh 16.5 cm berasio sebesar 4:1.

#### C. Programmable Logic Controller (PLC)

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*. Konfigurasi dari I/O yang digunakan oleh sistem beserta perangkat apa saja yang terhubung dengan I/O tersebut dapat dilihat di tabel berikut.

Tabel 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

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.

#### D. Raspberry Pi

Raspberry Pi digunakan sebagai pusat proses utama dari keseluruhan sistem. 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 ini membutuhkan daya sebesar +5 VDC dan 1,2 A. Tetapi karena banyak perangkat yang terhubung ke Raspberry Pi dan menggunakan cukup banyak daya, maka digunakan adaptor daya AC yang memiliki spesifikasi daya +5 V dan 2,5 A.

Alat ini juga dilengkapi modul kamera Raspberry Pi yang digunakan untuk mengambil video yang akan diolah oleh program pendeteksi lubang kertas tisu. Modul kamera ini terhubung ke Raspberry Pi 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. Penggunaan modul ini jauh lebih efisien karena dapat mengoptimalkan *Graphic Processing Unit (GPU)* yang tidak dapat digunakan oleh *webcam*. Dengan modul ini program dapat menggunakan *Central Processing Unit (CPU)* dan GPU sekaligus [6].

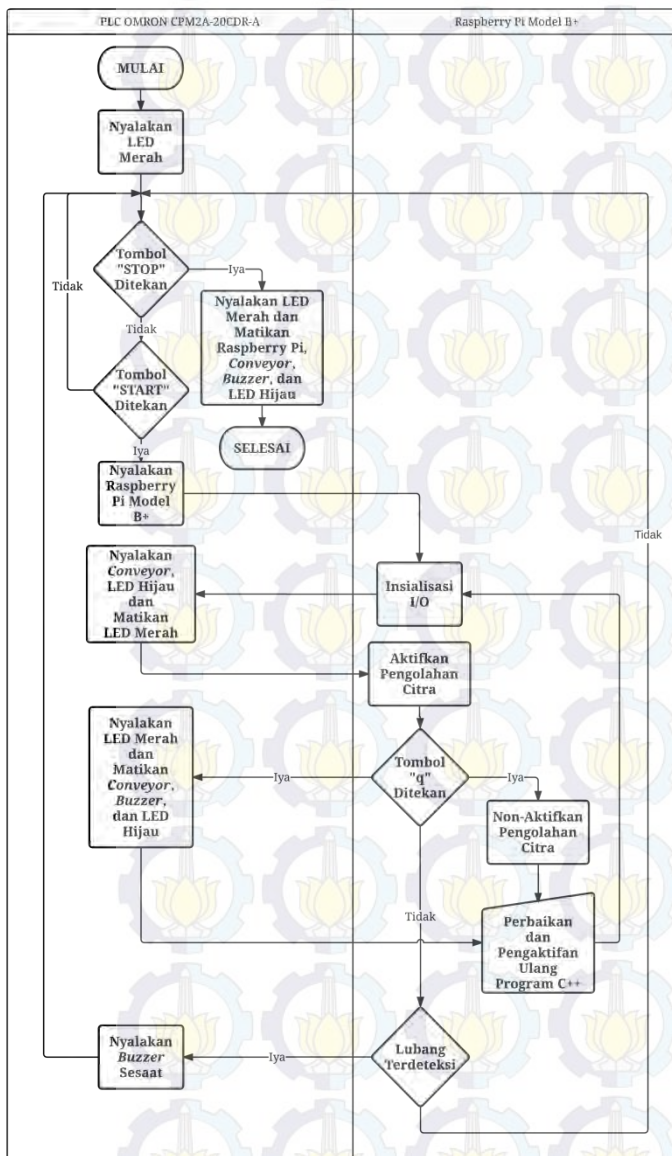
Tabel 3.  
Konfigurasi I/O Raspberry Pi Model B+.

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

IV. PERANCANGAN PERANGKAT LUNAK

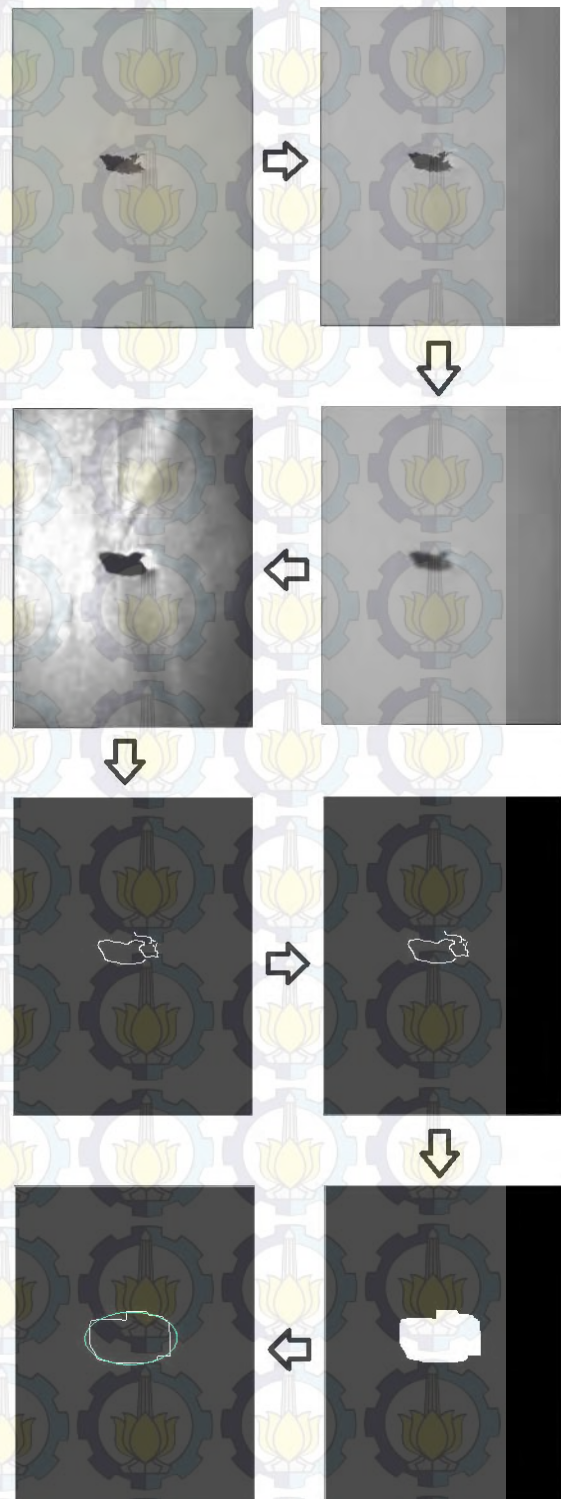
A. Program Raspberry Pi

Program di Raspberry Pi digunakan untuk mengolah citra dan berkomunikasi dengan PLC melalui komunikasi serial. Program ini memiliki cara kerja seperti gambar di bawah ini.



Gambar 5. Cara kerja program keseluruhan.

Pengolahan citra akan melakukan proses *grayscale*, yang diikuti oleh penghalusan, penyetaraan histogram, pendeteksian tepian Canny, *erode*, *dilate*, dan penemuan kontur [6]-[13]. Tujuh langkah pengolahan citra pada program ini akan menghasilkan pemrosesan citra seperti gambar berikut.



Gambar 6. Hasil pengolahan citra ketika terdeteksi lubang.

B. Program PLC

Pemrograman PLC ini dilakukan menggunakan CX-Programmer 9.5 di PC yang disambungkan ke PLC melalui

komunikasi serial RS-232. Program ini berbentuk *ladder diagram* dan digunakan untuk mengendalikan *buzzer* dan VFD. Berikut ini merupakan table konfigurasi alamat di PLC.

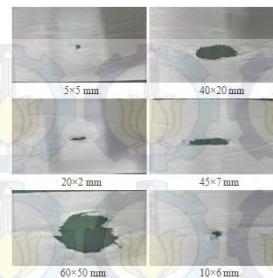
Tabel 4.  
Konfigurasi alamat PLC.

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>

## V. PENGUJIAN

Pengujian ini merupakan pengujian keseluruhan sistem yang disusun oleh berbagai perangkat. Pertama kita menyambungkan PLC dengan sumber daya DC, *pushbutton*, pembatas arus, VFD, *driver-driver*, dan Raspberry Pi yang sudah disertai modul kamera melalui RS-232. Kemudian tekan tombol "START" dan system akan berjalan.

Ketika Raspberry Pi sudah selesai *booting*, maka program akan berjalan dan pendeteksi lubang akan aktif. *Buzzer* akan menyala jika mendeteksi lubang. Pengujian ini dilakukan dengan menempelkan kertas tisu ke sabuk *conveyor* dan memutarinya sehingga menyerupai proses penggulungan. Dalam pengujian ini kita menguji dengan 6 ukuran lubang yang berbeda dan di 7 macam kecepatan *conveyor* yang berbeda.



Gambar 7. Jenis dan ukuran lubang.

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.

Tabel 5.  
Persentase *error* pengujian keseluruhan sistem.

Kecepatan <i>Conveyor</i>	<i>Error</i> (%)
25 RPM	0
50 RPM	0
75 RPM	0
100 RPM	5,55
150 RPM	11,11
200 RPM	38,89
250 RPM	72,22

Jika dilihat dari tabel di atas maka dapat dilihat bahwa *error* mulai terjadi di kecepatan 100 RPM. *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 100 RPM. Jika kecepatan penggulungan melewati 100 RPM, maka mulai terjadi *error* dan sistem tidak bisa bekerja secara *real-time*.

## VI. 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 *conveyor* berada di bawah 100 RPM.

## UCAPAN TERIMA KASIH

Penulis A.R.M. mengucapkan terima kasih kepada Bapak Suwito, S.T., M.T., Bapak Ir. Hendra Kusuma M.Eng.SC., dan Bapak Ir. Tasripan, M.T. atas saran dan bimbingannya. Penulis juga berterimakasih pada Imam Shalahuddin Tonaziz atas saran dan kerjasamanya.

## DAFTAR PUSTAKA

- [1] H. Purnomo, *Pengantar Teknik Industri*. Yogyakarta: Graha Ilmu (2004).
- [2] F. Bertini, and C. Rommel, "Can the Raspberry Pi Revolutionize the Low-End Industrial Computing Space?," *VDC Research* (2014, Apr.).
- [3] Raspberry Pi Foundation. (2014, Mei). Raspberry Pi FAQs [Online]. Available:  
<http://www.raspberrypi.org/help/faqs/>
- [4] *SYSMAC CPM1/CPM1A/CPM2A/CPM2C/SRM1(-V2) Programmable Controller PROGRAMMING MANUAL*. OMRON (2000).
- [5] Raspberry Pi Foundation. (2013, Juli). RaspiCam Documentation [Online]. Available:  
<http://www.raspberrypi.org/wp-content/uploads/2013/07/RaspiCam-Documentation.pdf>
- [6] OpenCV. (2014, Juli). Load, Modify, and Save an Image [Online]. Available:  
[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)
- [7] OpenCV. (2014, Juli). Smoothing Images [Online]. Available:  
[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)
- [8] OpenCV. (2014, September). Histogram Equalization [Online]. Available:  
[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)
- [9] OpenCV. (2014, Agustus). Canny Edge Detector [Online]. Available:  
[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)
- [10] OpenCV. (2014, Juli). Eroding and Dilating [Online]. Available:  
[http://docs.opencv.org/doc/tutorials/imgproc/erosion\\_dilatation/erosion\\_dilatation.html](http://docs.opencv.org/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html)
- [11] OpenCV. (2014, Juli). More Morphology Transformations [Online]. Available:  
[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)
- [12] OpenCV. (2014, September). Structural Analysis and Shape Descriptors [Online]. Available:  
[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)
- [13] J. Canny, "A Computational Approach to Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, Vol. PAMI-8 (1986, Nov.) 679-698.
- [14] S. Suzuki, and K. Abe, "Topological Structural Analysis of Digitized Binary Images by Border Following", *Computer Vision, Graphics, and Image Processing*, Vol. 30 (1985, Apr.) 32-46.