

Rancang Bangun Conveyor Penyortir Mur Berbasis Raspberry Pi Menggunakan Metode *Contour Area*

Dimas Surya Pratama

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
e-mail : dimassurya.18011@mhs.unesa.ac.id

Lilik Anifah, Lusia Rakhmawati, Rr. Hapsari Peni Agustin T.

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
e-mail : lilikanifah@unesa.ac.id, lusiarakhmawati@unesa.ac.id, hapsaripeni@unesa.ac.id

Abstrak

Kebutuhan akan penggunaan alat penyortir barang di industri secara otomatis, cepat, serta presisi dalam pendistribusian barang semakin tinggi sehingga diperlukannya suatu sistem *computer* yang dapat digunakan untuk memisahkan barang. Dengan adanya berbagai jenis ukuran dari mur, tentu industri pembuatan mur membutuhkan suatu alat penyortir otomatis untuk menyortir ukuran dari mur. Pada penelitian ini membahas tentang perancangan sebuah sistem *conveyor* yang digerakkan dengan motor dc 6V dengan pengolahan citra menggunakan metode *contour area* untuk menghitung jumlah *pixel* dari mur. *Mini PC* yang digunakan sebagai kontrol pusat yaitu Raspberry Pi 3 Model B dengan menggunakan bahasa pemrograman *Python*. Pengambilan citra dilakukan secara *real time* dengan menggunakan kamera *webcam* Logitech C922 Pro. Ukuran mur yang digunakan sebagai bahan pengujian yaitu mur dengan ukuran 7mm sebanyak 50 buah dan 10mm sebanyak 50 buah. Hasil citra yang didapatkan berupa jumlah *pixel* dari *contour area* citra biner pada objek mur yang digunakan untuk menentukan ukuran dari mur yang selanjutnya akan dilakukan proses penyortiran secara otomatis. Hasil dari penelitian penyortiran mur menggunakan metode *contour area*, memperoleh nilai presentase tingkat keberhasilan sebesar 99% dari 50 kali pengujian masing-masing ukuran mur dan didapatkan *error* sebesar 1%. Sistem bekerja optimal dengan kecepatan *belt* pada 82 rpm dan intensitas cahaya ruangan berkisar 125 Lux.

Kata kunci: *Mur, Raspberry Pi, Pengolahan citra, Contour area.*

Abstract

The need for the use of nuts sorting equipment in the industry automatically, quickly, and with precision in the distribution of nuts is getting higher so a computer system is needed that can be used to separate nuts. With the various sizes of nuts, the nut-making industry needs an automatic sorter to sort the sizes of nuts. This study discusses the design of a conveyor system driven by a 6V dc motor with image processing using the contour area method to calculate the number of pixels from the nuts. The mini PC used as the central control is the Raspberry Pi 3 Model B using the Python programming language. Image capture is done in real-time using the Logitech C922 Pro webcam camera. The size of the nuts used as testing material is 50 nuts with a size of 7mm and 50 pieces of 10mm. The image results obtained are in the form of the number of pixels from the contour area of the binary image on the nut object which is used to determine the size of the nut which will then be automatically sorted. The results of the research of sorting nuts using the contour area method obtained a percentage value of 99% success rate from 50 times of testing each nut size and obtained an error of 1%. The system works optimally with a belt speed of 82 rpm and a room light intensity of around 125 Lux.

Keywords: *Nuts, Raspberry Pi, Image processing, Contour area.*

PENDAHULUAN

Mur merupakan salah satu bagian yang tidak terpisahkan dengan kebutuhan sehari-hari mulai dari keperluan rumah tangga maupun profesional yang bekerja dengan alat teknik yang membutuhkan mur (Natalia, 2020). Terdapat banyak ukuran mur di pasaran, mulai dari ukuran kecil, menengah hingga berukuran besar. Dengan adanya berbagai jenis ukuran dari mur, tentu berpengaruh

terhadap proses penyortiran dipabrik mur. Industri membutuhkan suatu peralatan yang dapat bekerja secara otomatis, kontinu, teliti, dan presisi (Wicaksono, dkk, 2018). Kemajuan teknologi saat ini dapat menjadi solusi atau jawaban atas kebutuhan industri maupun manusia tersebut.

Alat penyortir merupakan sebuah alat untuk menyortir barang yang digunakan pada suatu industri, dimana barang tersebut disortir berdasarkan kriteria yang

diinginkan oleh industri, kriteria yang diinginkan tersebut biasanya berdasarkan ukuran, warna, tekstur, maupun jenis dari barang tersebut (Safaris dan Hansi, 2020). Kebutuhan akan penggunaan alat penyortir barang di industri secara otomatis, cepat, serta presisi dalam pendistribusian barang semakin tinggi sehingga diperlukannya suatu sistem computer yang dapat digunakan untuk memisahkan barang tersebut. Selain itu, penggunaan alat ini dapat menghemat biaya produksi karena proses penyortiran dilakukan secara otomatis dan tidak memerlukan banyak pekerja, di lain sisi apabila proses penyortiran dilakukan secara manual maka tidak lepas dengan adanya *human error* seperti pekerja yang kelelahan maupun kurang teliti dalam menyortir barang (Tarigan, dkk, 2018).

Berdasarkan kondisi tersebut alat penyortir otomatis dibuat dengan memanfaatkan kemajuan teknologi dan informasi yang mengalami perkembangan cukup pesat saat ini, salah satunya yaitu *computer vision* sebagai pusat berjalannya program. *Computer vision* merupakan bidang pengetahuan yang memungkinkan suatu *computer* dapat melihat lingkungan disekitarnya kemudian mengolah citra yang dilihat menjadi sebuah perintah untuk menyelesaikan tugas tertentu (Faradila, dkk, 2019).

Pengolahan citra digital (*image processing*) merupakan suatu metode untuk melakukan konversi masukan berupa citra menjadi bentuk digital dengan tujuan untuk mendapatkan kualitas citra yang diinginkan maupun mengambil informasi yang ada pada citra tersebut (Putri, 2016). Pengolahan citra digital merupakan pengembangan metode untuk menganalisis informasi intelektual *visual* yang memiliki dampak yang sangat besar pada kehidupan (Lyakhov, dkk. 2020). Saat ini pengaplikasian pengolahan citra digital sudah banyak digunakan diberbagai bidang seperti bidang kedokteran, astronomi, industri dan masih banyak lagi.

Salah satu bahasa pemrograman yang digunakan untuk mengolah citra yaitu bahasa pemrograman *python* seperti yang dipakai pada penelitian ini. Hampir di semua *platform*, bahasa pemrograman *python* dapat dioperasikan antara lain yaitu seperti Linux, Windows, Mac dan *platform* lainnya. Penggunaan *source code* aplikasi pada bahasa pemrograman *Python* biasanya akan dikompilasi menjadi format perantara yang dikenal sebagai *bytecode* yang selanjutnya akan dieksekusi oleh perangkat atau mesin (Ratna, 2020). Sedangkan untuk *library* yang digunakan diperuntukkan untuk melakukan pengolahan citra atau *image processing* adalah OpenCV yang dikembangkan oleh Intel Corporation. Dimana *library* OpenCV merupakan sebuah *library* gratis yang memiliki tujuan agar komputer mempunyai kemampuan

yang hampir sama dengan cara pengolahan visual pada manusia (Sulistiyanti, 2019).

Berkembangnya sistem kecerdasan buatan saat ini menjadikan bidang pengolahan citra digital semakin berkembang. Telah banyak riset yang membahas segmentasi video berdasarkan suatu kriteria tertentu. Penelitian mengenai alat penyortir barang otomatis pada *conveyor* dengan pengolahan citra telah dilakukan oleh (Wicaksono, dkk, 2018) menggunakan sebuah kamera untuk menangkap warna dari barang yang akan disortir. Selanjutnya ada penelitian yang dilakukan (Yunardi, dkk, 2015) dengan mendeteksi objek berupa kotak parcel berbasis *contour* untuk mengukur luas dan volume objek. Selain itu juga terdapat pada tugas akhir yang dilakukan oleh (Natalia, 2020) yang menggunakan *conveyor* dengan metode *Euler Number* untuk mencacah dan menyortir baut dengan mur.

Penelitian ini hampir sama dengan penelitian sebelumnya yang sudah dilakukan oleh (Natalia, 2020). Pada penelitian tersebut menggunakan *conveyor* dengan metode *Euler Number* dan *contour area* untuk mencacah dan menyortir baut dengan mur (Natalia, 2020). Sedangkan pada penelitian ini menggunakan metode *contour area* untuk menyortir mur berdasarkan ukurannya. Sistem pada penelitian (Natalia, 2020). masih terdapat kekurangan yaitu mur dan baut terpisah pada dua tempat akan tetapi dalam satu wadah tersebut masih terdapat berbagai ukuran sehingga menambah waktu untuk memilah mur dan baut tersebut sesuai ukurannya. Oleh karena itu pada penelitian ini hanya membahas tentang bagaimana merancang sebuah sistem *conveyor* menggunakan metode *contour area* dengan mendeteksi jumlah *pixel* pada mur untuk dilakukan proses penyortiran mur berbasis Raspberry Pi dengan bahasa pemrograman *Python* secara *real time*. Tujuan dari penelitian ini adalah untuk mengetahui hasil pengujian dari sistem *conveyor* dengan pengolahan citra menggunakan metode *contour area* sebagai penyortir mur berbasis Raspberry Pi. *Contour area* merupakan metode untuk mengetahui jumlah *pixel* yang direpresentasikan dari suatu objek yang ditangkap atau di *capture* oleh kamera. Selain itu, pada penelitian ini juga menggunakan *Region of Interest* untuk mengurangi *noise* pada *background* sekaligus mengurangi tingginya waktu pemrosesan.

Belt conveyor digerakkan oleh motor DC 6V sebagai media pengangkut atau pembawa mur. Kemudian mur akan dipisahkan pada wadah yang terbagi menjadi dua bagian dan terdapat papan sortir yang digerakkan oleh motor servo sebagai media penyortir mur. Dengan adanya alat ini, sistem penyortiran mur dapat dikerjakan secara otomatis berdasarkan ukurannya

sehingga pengguna hanya meletakkan mur pada *belt conveyor* dengan ukuran yang berbeda dan alat akan berjalan sesuai perintah yang sudah diprogram sebelumnya. Untuk ukuran dari mur yang digunakan sebagai bahan pengujian yaitu mur dengan ukuran 7 mm dan 10 mm. *Conveyor* akan menjalankan mur pada *belt* dan akan dipisahkan berdasarkan ukurannya dengan pengolahan citra yang ditangkap oleh *webcam*.

METODE

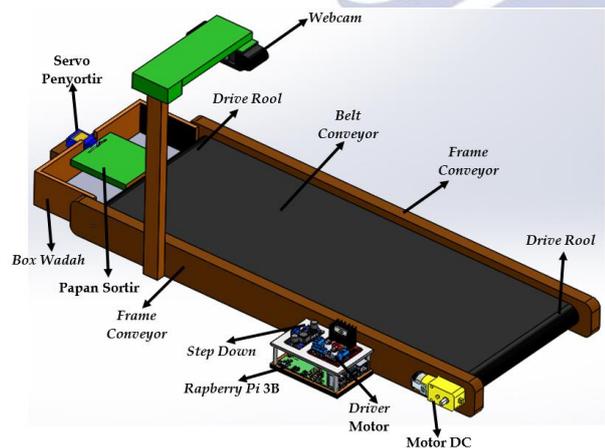
Tahapan Penelitian

Instrument pengambilan data dalam penilitan ini menggunakan Raspberry Pi 3 Model B dengan *software* Thonny IDE. Data akan ditampilkan pada *frame* yang dijalankan menggunakan bahasa pemrograman *Python* 3.9 dan library *OpenCV* untuk mendapatkan citra dan data berupa nilai *contour area* dari mur. Tahapan perancangan dari penelitian ini ditunjukkan dengan diagram alur pada Gambar 1.



Gambar 1. Tahapan penelitian

Perancangan Mekanik



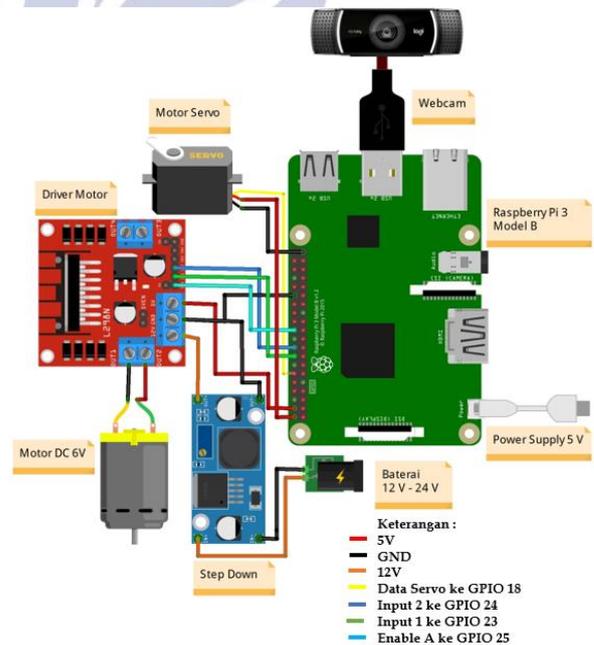
Gambar 2. Rancangan mekanik conveyor

Visualisasi terkait perancangan bentuk sistem dari alat pensortir mur yang akan dikerjakan ditampilkan secara isometri pada Gambar 2. *Conveyor* terdiri dari penampang *belt*, *frame* atau dinding *conveyor*, *box wadah* yang terbuat dari kayu dan terdapat 2 *drive roll* untuk menggerakkan *belt*. Sedangkan untuk bahan *belt*

dari spon karet dengan dilapisi kain hitam untuk membedakan warna objek dengan warna *background*. *Belt conveyor* memiliki panjang 46 cm. Peletakan *webcam* berada diatas *belt conveyor* dengan ketinggian 25 cm. Salah satu *drive roll conveyor* digerakkan menggunakan motor dc 6V. Untuk wadah penampung mur terdapat papan sortir yang terbuat dari bahan akrilik dengan ukuran 7 X 9 cm yang digerakkan oleh motor servo Tower Pro SG90 dan *Box wadah* berukuran 18 X 10 cm.

Perancangan Hardware

Perancangan *Hardware* terdiri dari *wiring* motor DC, *driver* motor, modul *step down*, motor servo, dan *webcam* yang dihubungkan pada *Mini PC* (Raspberry Pi 3 Model B). *Wiring* keseluruhan ditunjukkan pada Gambar 3. *Mini PC* (Raspberry Pi) sebagai alat untuk penerima data dan pemrosesan serta sebagai pengiriman program ke perangkat atau komponen lain yaitu *webcam*, motor DC, dan motor servo. *Mini PC* terhubung oleh adaptor dengan *output* sebesar 5V dan untuk interface dari Raspberry Pi pada penelitian ini menggunakan *VNC* yang dijalankan pada laptop.

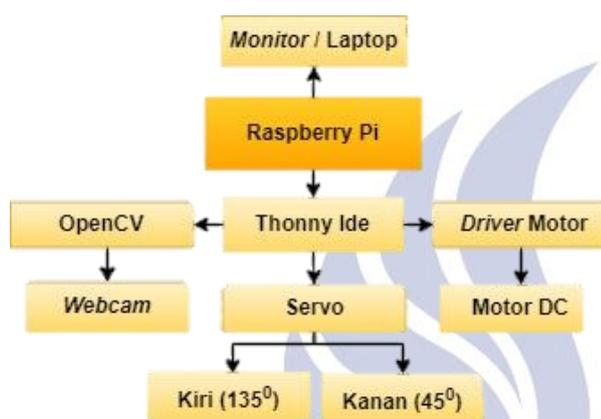


Gambar 3. Wiring diagram

Webcam terhubung melalui kabel USB untuk meng-*capture* atau menangkap citra secara *real time* diatas *belt conveyor* yang berjalan. Modul *step down* yang terhubung dengan baterai berfungsi untuk *supply* tegangan 6V dengan menghubungkan pin 6V dan GND pada *driver* motor. Pada bagian GND *driver* motor juga dihubungkan pada pin GND Raspberry Pi atau yang disebut dengan *common ground* agar kedua perangkat dapat saling berkomunikasi.

PWM motor DC diatur melalui pin GPIO 25 (pin 22) yang dihubungkan dengan pin ENA driver motor sebagai penggerak motor untuk *belt conveyor*, pin IN1 dihubungkan dengan pin GPIO24 (pin 18) dan pin IN2 dihubungkan dengan pin GPIO23 (pin 16). Pin IN1 dan IN2 digunakan untuk menentukan kondisi *HIGH* atau *LOW* sebagai pengatur arah putaran dari motor DC. Sedangkan data motor servo dihubungkan dengan pin GPIO 18 (pin 11). Pin 5V dan GND servo dihubungkan dengan 5V (pin 1) dan GND (pin 3) Raspberry Pi .

Perancangan Software



Gambar 4. Blok diagram rancangan software

Perancangan *software* ditunjukkan menggunakan blok diagram yang ditunjukkan pada Gambar 4. Komponen *hardware* dan pengolahan citra pada penelitian ini di jalankan menggunakan program *python* dengan *software* Thonny Ide dan *library* OpenCV untuk meng-ekstrak informasi dari sebuah citra. *Webcam*, motor dc dan servo kondisi *standby* saat program *python* pada Raspberry Pi dijalankan. Kemudian motor dc yang terpasang dengan *drive roll* menggerakkan *belt* untuk menjalankan mur satu per satu. *Webcam* meng-*capture live video* pada *belt conveyor* untuk mendeteksi objek atau mur yang akan di sortir. Posisi awal servo diatur berada pada 90° sehingga posisi papan berada pada keadaan lurus sejajar dengan *conveyor*. Ukuran objek berupa nilai *pixel* dari mur, nilai dengan *range* 400 sampai 730, maka Raspberry Pi akan mengirimkan perintah pada servo dengan sudut 135° (ke kiri). Apabila mur dengan *range* nilai *pixel* 780 sampai 1000 maka Raspberry Pi akan mengirimkan perintah pada servo dengan sudut 45° (ke kanan). Range tersebut ditentukan berdasarkan nilai dari area mur saat berjalan diatas *belt conveyor* dengan kisaran nilai kurang lebih 656 untuk ukuran 7mm dan ukuran 10mm dengan kisaran nilai kurang lebih 913. Acuan kisaran *range* nilai tersebut didapat dari pengujian yang ditunjukkan pada Gambar 11 (c), dan (d).

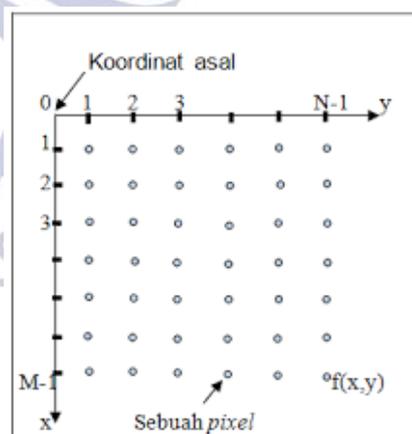
Metode Image Processing

Metode yang digunakan dalam pengolahan citra pada penelitian ini ditunjukkan pada Gambar 5. Dimana sebuah citra perlu dikonversi dahulu ke dalam bentuk nilai-nilai numerik agar dapat diolah oleh pustaka.



Gambar 5. Flowchart metode image processing

Nilai *x*, *y*, dan *f* pada citra digital yang berhingga dan diskrit dengan nilai *M* baris dan *N* kolom. Nilai tersebut terdapat nilai elemen yang berhingga atau sering disebut dengan *image element pels*, *picture element* dan *pixels* (Simanjuntak, 2019). Sebuah *pixel* dapat mewakili sebuah bagian berupa kotak yang merupakan bagian terkecil dari citra (Putri, 2016). Rentang nilai pada suatu *pixel* berbeda-beda tergantung dari jenis warnanya. Jangkauan rentang nilainya adalah 0-255. Pada persamaan (1) disajikan persamaan dari citra digital dalam bentuk matriks yang ditunjukkan berupa titik-titiknya pada Gambar 6.



Gambar 6. Koordinat citra diskrit (Sumber :Putri, Asti Riani, 2016)

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0, N - 1) \\ f(1,0) & f(1,1) & f(1, N - 1) \\ f(M - 1,0) & f(M - 1,1) & f(M - 1, N - 1) \end{bmatrix} \dots(1)$$

Keterangan :

M = Banyak *pixel* per baris

N = Banyak *pixel* per kolom

Raspberry Pi akan memberikan perintah *capture* dengan fungsi *cv2.VideoCapture* untuk memindai objek atau mur tersebut. Proses akuisisi citra dan *image processing* yang diambil masih dalam bentuk citra berwarna atau RGB. Pada perancangan, citra diberikan batasan *frame* atau ROI (*Region of Interest*) untuk mengurangi banyaknya *noise* yang tertangkap oleh *webcam*. Selain itu juga digunakan untuk mengurangi waktu pengolahan citra. Ukuran yang digunakan dalam *Region of Interest* yaitu ($x_1= 140$, $x_2= 320$ dan $y_1= 1200$, $y_2= 220$).

Selanjutnya citra RGB dikonversi menjadi citra *greyscale* menggunakan fungsi dari *library* OpenCV yaitu *cv2.cvtColor*. Yang semula warna awal citra terdiri dari 3 layer matrik disederhanakan menjadi model citra 1 layer matrik yaitu ruang warna *greyscale* atau keabuan saja (Santi, 2011).

Citra *greyscale* dikonversi ke citra biner dengan fungsi *cv2.threshold*. Cara untuk mengkonversi citra *greyscale* ke citra biner yaitu dengan menetapkan nilai ambang (*threshold*). Penggunaan nilai ambang yaitu untuk menentukan suatu intensitas yang akan dikonversi menjadi nilai 0 atau menjadi 1. Secara matematis, konversi dinyatakan dengan rumus pada persamaan:

$$g(x,y) = \begin{cases} 0 & \text{jika } f(x,y) < T \\ 1 & \text{jika } f(x,y) \geq T \end{cases} \dots\dots\dots(2)$$

Bila $g(x,y)$ adalah citra biner dari citra *greyscale* $f(x,y)$ dan T merupakan nilai ambang. Kualitas yang dihasilkan dari cira biner bergantung pada nilai T yang digunakan. Citra yang akan diolah dengan memfokuskan pada kontur objek yaitu dengan fungsi *cv2.findContours*. Citra biner kemudian disegmentasi untuk mengenali bentuk maupun ukurannya agar dapat mempermudah proses pengenalan citra (Lestari, dkk. 2019). Pada dasarnya segmentasi citra merupakan masalah mendasar dari *computer vision* dan memiliki peran penting dalam beragam pengaplikasiannya seperti pengambilan citra dan deteksi objek (Xiang, dkk. 2017). Proses segmentasi pada penelitian ini menggunakan *Bounding Box* dengan fungsi *cv2.boundingRect* untuk menandai objek yang dideteksi. Dari hasil segmentasi objek tersebut, Raspberry Pi mengirimkan perintah ke servo dengan menggerakkan papan sortir dengan sudut 45° atau 135° untuk mensortir mur sesuai dengan ukuran yang dihitung berdasarkan nilai area *pixel* dari hasil citra biner objek tersebut menggunakan fungsi *cv2.contourArea*. Pada citra biner, luas area objek berwarna putih. Warna putih pada citra biner bernilai 1 *pixel*. Total dari *pixel* dalam sebuah citra merupakan nilai areanya (Chaohui, dkk. 2010). Rumus area ditunjukkan pada persamaan (3).

Hasil dari luas area objek ditampilkan pada citra asli atau citra RGB seperti pada Gambar 11 (a), dan (b).

$$A = \sum_{x=1}^M \sum_{y=1}^N f(x,y) \dots\dots\dots(3)$$

Keterangan :

A = Jumlah *pixel* dari luas objek (area)

M = Jumlah kolom *pixel*

N = Jumlah baris *pixel*

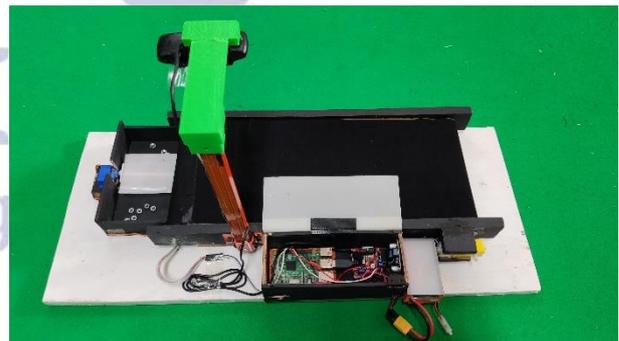
(x,y) = Koordinat spasial *pixel*

$f(x,y) = 1$, jika (x,y) adalah *pixel* objek dalam citra biner

HASIL DAN PEMBAHASAN

Hasil Perancangan *Hardware* dan *Software*

Hasil perancangan terdiri dari alat penyortir mur ditampilkan secara fisik pada Gambar 7 dari hasil perancangan terdapat beberapa pembagian mulai dari *input*, proses, dan *output*. Dimana komponen input yaitu *webcam* yang meng-*capture* objek atau mur untuk mendapatkan ukurannya dari nilai area *pixel*. *Webcam* yang digunakan pada penelitian ini adalah *webcam* Logitech C922 Pro yang terhubung langsung dengan Raspberry Pi melalui port USB-A. Mikrokontroler atau mini PC yang digunakan yaitu Raspberry Pi 3 Model B. Raspberry Pi berfungsi sebagai mikrokontroler dan juga sebagai pengolah data *webcam* ditempatkan disebelah *frame conveyor*. Sedangkan untuk komponen *output* berupa 1 buah motor servo Tower Pro SG90 dipasangkan dengan papan sortir yang terdapat di *box* wadah sebagai penyortir mur berdasarkan ukurannya. Selain itu juga terdapat motor DC yang digunakan untuk memutar *drive roll conveyor*.



Gambar 7. Tampilan *conveyor* secara fisik

Integrasi *Hardware* & *Software*

Dari perancangan *hardware* dan *software*, dilakukan proses integrasi secara keseluruhan untuk menentukan apakah pada sistem tersebut berjalan sesuai dengan apa yang diharapkan atau tidak (Ariansyah, 2019). integrasi tersebut dilakukan dengan tujuan untuk melihat kemungkinan terjadinya kesalahan atau *error* pada sistem dengan melakukan beberapa pengujian pada

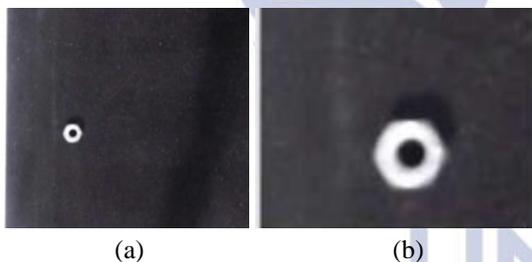
sistem. Apabila masih terjadi kesalahan maka dilakukan pengecekan ulang terhadap rancangan *hardware* dan *software* kemudian di integrasikan ulang sampai terdapat kesalahan kecil atau bahkan sampai tidak terjadi kesalahan. Pengujian sistem dan pengambilan data dilakukan di Laboratorium Robotika, Fakultas Teknik, Universitas Negeri Surabaya.

Pengujian Webcam

Pengujian *webcam* dilakukan secara *real time* untuk meng-*capture* objek. Pengujian dilakukan pada setiap proses *image processing* dengan kamera *webcam* Logitech C922 Pro dengan resolusi full HD 1920 x 1080 *pixel*, 30 fps dengan posisi kamera berjarak 25 cm diatas objek mur dengan nilai intensitas cahaya sekitar 125 Lux. Pengujian yang dilakukan mulai dari pengaruh penggunaan ROI, konversi citra RGB ke *grayscale*, konversi citra *grayscale* ke citra biner atau *thresholding*, segmentasi dengan *bounding box* dan menentukan nilai *pixel* dari *contour area* pada mur.

Hasil Pengujian ROI

Penggunaan ROI bertujuan untuk menghilangkan sebagian *background* yang ditangkap kamera sehingga memudahkan objek yang dianalisa (Widiyanto, 2020). Pengujian dilakukan dengan membedakan saat sebelum menggunakan ROI dan sesudah menggunakan ROI.

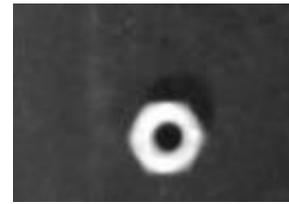


Gambar 8. (a) *Frame* asli (b) *Frame* setelah menggunakan ROI

Dari hasil pengujian seperti pada Gambar 8 dapat diketahui bahwa hasil citra dari objek mur dengan ROI lebih tampak jelas dan fokus sehingga dapat mengurangi jumlah *noise* dari *background* saat dilakukan tahap pengolahan citra selanjutnya.

Hasil Konversi Citra RGB ke *Greyscale*

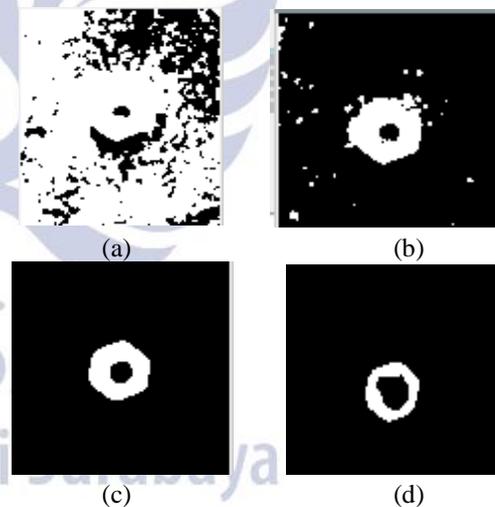
Konversi citra berwarna atau RGB menjadi citra *greyscale* atau derajat keabuan seperti yang ditunjukkan pada Gambar 9 merupakan proses awal yang dilakukan dalam *image processing* untuk memperoleh nilai warna yang sederhana.



Gambar 9. Citra *grayscale*

Hasil Pengujian Segmentasi Menggunakan *Thresholding*

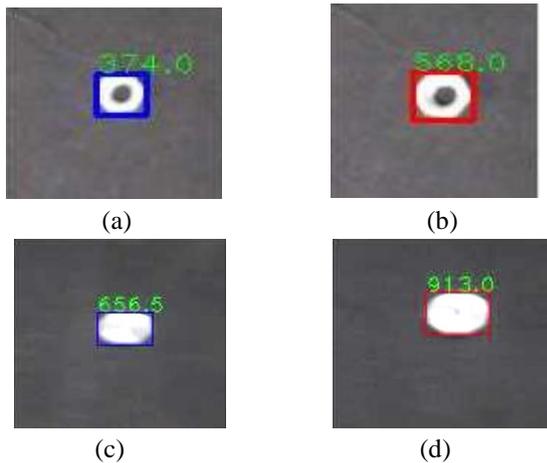
Terjadi perubahan tingkat iluminasi dari citra *output* yang dihasilkan dengan metode *thresholding* sesuai dengan besar kecilnya nilai ambang atau nilai T yang ditentukan. Adapun kisaran nilai ambang yang dipakai adalah dari 0 sampai 255. Dari hasil pengujian dapat diketahui bahwa nilai ambang yang semakin mendekati nilai 0 atau 255 akan membuat objek yang disegmentasi sulit dibedakan atau sama dengan *background*. Kontras warna yang dihasilkan dari proses *thresholding* memudahkan dalam membedakan batas citra dari objek dengan *background* (Xiaoyu, dkk. 2016). Nilai yang digunakan untuk pengujian adalah 75, 90, 130 dan 240 seperti yang ditampilkan pada Gambar 10. Sedangkan nilai ambang yang digunakan untuk pengambilan data pada penelitian ini yaitu 130.



Gambar 10. *Thresholding* dengan nilai ambang (a) 75, (b) 90, (c) 130, dan (d) 240

Pengujian *Bounding Box* pada Objek

Bounding box merupakan salah satu cara untuk menandai citra dengan metode *region of interest* (ROI) dengan cara menemukan koordinat terdalam dan terluar dari objek mur, kemudian dibuat *rectangle* dan juga untuk menentukan ukuran dari mur berdasarkan jumlah *pixel*. Contoh *bounding box* ditampilkan pada Gambar 11 yang diambil saat *conveyor* diam (a),(b) dan jalan (c),(d).



Gambar 11. (a) dan (b) *Bounding box* objek saat diam, (c) dan (d) *Bounding box* objek saat *conveyor* berjalan

Pengujian Kecepatan Motor DC

Kecepatan motor DC sebagai penggerak *belt conveyor* sangat berpengaruh terhadap penyesuaian kecepatan pembacaan *webcam* dalam mendeteksi keberadaan mur dan nilai *pixel* dari mur. Dengan begitu dilakukan pengujian sebanyak 100 kali terdiri dari 50 mur berukuran 7mm dan 50 mur berukuran 10mm terhadap kecepatan motor DC mulai dari kecepatan 54 rpm, 65 rpm dan 82 rpm. Dari hasil pengujian ditunjukkan dengan *confusion matriks* pada Tabel 1, 2, dan 3 sebagai berikut.

Tabel 1. Pengujian dengan kecepatan motor 54 rpm

<i>True</i> \ <i>Predicated</i>	Mur Ukuran	
	7 mm	10 mm
Mur Ukuran 7 mm	50 Mur	0 Mur
Mur Ukuran 10 mm	11 Mur	39 Mur

Tabel 2. Pengujian dengan kecepatan motor 65 rpm

<i>True</i> \ <i>Predicated</i>	Mur Ukuran	
	7 mm	10 mm
Mur Ukuran 7 mm	50 Mur	0 Mur
Mur Ukuran 10 mm	5 Mur	45 Mur

Tabel 3. Pengujian dengan kecepatan motor 82 rpm

<i>True</i> \ <i>Predicated</i>	Mur Ukuran	
	7 mm	10 mm
Mur Ukuran 7 mm	50 Mur	0 Mur
Mur Ukuran 10 mm	1 Mur	49 Mur

Pada penelitian ini menggunakan kecepatan motor DC yaitu 82 rpm, penggunaan rpm tinggi disesuaikan pada proses pensortiran di industri yang memerlukan proses cepat dengan akurasi yang tinggi. Akan tetapi semakin tinggi rpm yang digunakan dengan penggunaan metode *contour area* secara *real time* maka objek yang terdeteksi akan melebar sehingga ukuran yang terbaca menjadi lebih besar dari ukuran objek aslinya. Selain itu fps *webcam* yang digunakan terbatas, maka dari itu pada penelitian ini sudah disesuaikan terlebih dahulu *range* nilai area dari objek mur dengan kecepatan motor yang digunakan.

Pengujian Sistem dengan Lux yang Berbeda

Tabel 4. Pengujian terhadap intensitas cahaya yang berbeda

Intensitas Cahaya (Lux)	Pengujian (Jumlah Mur)	Jumlah mur yang sesuai ukuran	Akurasi
70	100	80	80%
125	100	99	99%
145	100	96	96%
170	100	82	82%

Pengujian sistem terhadap intensitas cahaya (Lux) dilakukan pada ruang tertutup dengan cahaya lampu TL dengan jumlah lampu sebanyak 5. Lux bernilai 70 ketika lampu menyala 2, 125 ketika menyala 3 lampu, 145 ketika menyala 4 lampu dan 170 ketika menyala 5 lampu. Berdasarkan Tabel 4 dapat diketahui sistem dapat bekerja secara optimal dalam menyortir mur sesuai ukuran pada intensitas cahaya ruangan berkisar 125 Lux dengan nilai batas ambang *threshold* sebesar 130. Dimana saat nilai Lux rendah, sistem sulit mendeteksi objek karena cahaya yang kurang terang membuat objek memiliki warna yang hampir sama dengan *background* atau hitam. Sedangkan saat nilai Lux semakin tinggi warna dari *backgorund* akan terang dan membuat objek terdeteksi semakin besar.

Pengujian Sistem Secara Keseluruhan

Pengujian dilakukan pada sistem secara keseluruhan untuk melihat proses yang terjadi, mulai dari pergerakan mur yang dijalankan pada *belt conveyor*, selanjutnya mur akan dideteksi oleh *webcam* dengan metode *image processing* untuk mensortir mur sampai pada *box* wadah pemisah mur sesuai ukuran yang telah ditentukan.

Berdasarkan hasil pengujian yang dilakukan terhadap seluruh komponen alat menunjukkan bahwa seluruh perancangan alat dapat berjalan sesuai dengan fungsi yang peneliti harapkan. Pengambilan data dilakukan sebanyak 100 kali percobaan dengan ukuran mur yang berbeda yaitu 50 mur berukuran 7mm dan 50

mur berukuran 10mm, berdasarkan hasil pengujian sistem secara keseluruhan dengan kecepatan motor DC berkisar 82 rpm dan Lux bernilai 125. Adapun hasil dari pengujian sistem alat penyortir mur sesuai dengan ukurannya berdasarkan metode *contour area* ditunjukkan menggunakan *confusion matriks* pada Tabel 5.

Tabel 5. Hasil uji sistem secara keseluruhan

True \ Predicated	Mur Ukuran	
	7 mm	10 mm
Mur Ukuran 7 mm	50 Mur	0 Mur
Mur Ukuran 10 mm	1 Mur	49 Mur

Pada *confusion matriks* pengujian sistem secara keseluruhan dapat diketahui dari 100 kali pengujian dengan mur berukuran 7mm dan 10mm menunjukkan bahwa tingkat akurasi yang didapatkan dari respon *webcam* dengan metode *contour area* dalam menentukan nilai *pixel* atau ukuran dari mur sebesar 99% sehingga sistem dapat dikatakan bekerja cukup bagus dan cukup sesuai dengan penyortiran yang dilakukan. Pada pengujian masih terdapat *error* sebesar 1% dikarenakan terdapat sedikit cacat pada fisik objek mur seperti mur yang terdapat karat, sehingga warna dari karat tersebut terbaca oleh sistem bernilai 0 seperti nilai pada *backgroundnya*. Selain itu fps yang turun saat sistem mendeteksi objek mengakibatkan terjadinya *delay* sehingga kontur dari objek mur akan melebar atau membesar yang mengakibatkan jumlah *pixel* objek lebih besar atau banyak.

Pada penelitian ini memiliki akurasi 99%, akurasi tersebut hampir sama dengan penelitian yang sudah dilakukan oleh (Natalia, 2020) yaitu memiliki akurasi 99,02% dalam menyortir mur dan baut, akan tetapi terdapat perbedaan pada penggunaan tegangan kerja dari motor dan kecepatan motor yang diatur. Pada penelitian ini menggunakan kecepatan motor maksimum sebesar 82 rpm disesuaikan dengan kondisi yang dibutuhkan industri saat ini yaitu kebutuhan akan alat penyortir yang cepat dan presisi. Selain itu pada penelitian (Yunardi, dkk, 2015) memiliki akurasi rata-rata 87,5% dalam menyortir kotak parcel berdasarkan luas *contour* dan volumenya. *Error* pada penelitian (Yunardi, dkk, 2015) bisa juga dikarenakan pencahayaan yang kurang maksimal sehingga sebagian *contour* dari kotak parcel tidak terdeteksi. Pada Tabel 6 menunjukkan hasil dari 10 pengujian pertama dengan tingkat akurasi sebesar 90% menyatakan kebenaran dan 10% terjadi *error*.

Tabel 6. Hasil dari 10 pengujian pertama

Citra dari Objek Mur Terdeteksi	Ukuran Mur Hasil Pengamatan	Keterangan
	M7	Benar
	M10	Benar
	M7	Benar
	M10	Benar
	M7	Benar
	M10	Benar
	M7	Benar
	M7	Benar
	M10	Benar
	M10	Salah

PENUTUP

Simpulan

Hasil pengujian secara *real time* menunjukkan bahwa *webcam* dapat mendeteksi mur yang berjalan diatas *belt conveyor* dengan kecepatan motor 82 rpm dan dapat menghitung nilai *pixel* untuk menunjukkan ukuran dari mur yang akan disortir, dengan tingkat keberhasilan sebesar 99 % dan *error* 1% yang diperoleh dari 100 kali pengujian terdiri dari 50 buah mur dengan ukuran 7mm dan 50 buah mur dengan ukuran 10mm. *Error* yang terjadi pada saat pengujian dikarenakan keadaan fisik dari objek mur dan penggunaan *hardware* yang terbilang masih memiliki spesifikasi rendah sehingga saat proses berlangsung terdapat *delay* yang mengakibatkan area dari objek tidak sesuai. Sistem bekerja dengan baik pada tingkat intensitas cahaya bernilai sekitar 125 Lux.

Saran

Berdasarkan hasil pengujian, terdapat beberapa saran yang bisa dikembangkan pada sistem agar didapatkan hasil yang lebih maksimal seperti menggunakan *mini pc*

dengan spesifikasi yang lebih tinggi agar dapat mempercepat proses pengolahan data terutama pada saat pengolahan citra, penggunaan motor DC diatas 6V agar pergerakan *drive roll conveyor* lebih stabil dengan torsi yang lebih tinggi, pemberian *gate* pada *conveyor* agar objek lebih tepat atau fokus dalam cakupan *webcam* selain itu juga menggunakan dataset dari objek yang akan dipakai untuk mengenali objek sehingga dapat mengurangi kesalahan pembacaan atau pengenalan objek apabila jenis objek yang disortir berbeda.

REFRENSI

- Ariansyah, Randi. 2019. *Rancang Bangun Alat Sortir Jeruk Berbasis Mikrokontroler*. Makasar. Universitas Islam Negeri Makasar.
- Chaohui Lu, Hui Ren, Yibin Zhang, dan Yinhua Shen. 2010. *Leaf Area Measurement Based on Image Processing*. IEEE International Conference on Measuring Technology and Mechatronics Automation. 978-0-7695-3962-1/10.
- Fharadila, Bayu, dan Feri Candra. 2019. *Identifikasi Kematangan Buah Nanas Menggunakan Metode Jaringan Syaraf Tiruan*. Jom FTEKNIK. Vol.6 (1).
- Gusa, Rika Favoria. 2013. *Pengolahan Citra Digital untuk Menghitung Luas Daerah Bekas Penambang Timah*. Jurnal Nasional Teknik Elektro. Vol.2 No.2.
- Lestari, Puji, dan Hans-Peter Schade. 2019. *Boundary Matched Human Area Segmentation for Chroma Keying Using Hybrid Depth-Color Analysis*. IEEE 4th International Conference on Signal and Image Processing. 978-1-7281-3660-8/19
- Lyakhof, P.A, A.S.Abdulsalyamova, M.R. Kiladze, D.I. Kaplun, dan A.S. Voznesensky. 2020. *Method of Oriented Contour Detection on Image Using Lorentz Function*. Mediterranean Conference on Embedded Computing. 978-1-7281-6949-1/20 IEEE.
- Natalia, Lidya. 2020. *Pencacah dan Penyortir Baut dan Mur Menggunakan Raspberry Pi*. Tugas Akhir. Yogyakarta: Universitas Sanata Dharma.
- Putri, Asti Riani. 2016. *Pengolahan Citra dengan Menggunakan Webcam pada Kendaraan Bergerak di Jalan Raya*. Jurnal Ilmiah Pendidikan Informatika. Vol.1 (1) pp.1-6.
- Ratna, Silvia. 2020. *Pengolahan Citra Digital dan Histogram dengan Python dan Text Editor Pycharm*. JurnalTechnologia. Vol. 11, No.3.
- Safaris, Ahmad dan Hansi Efendi. 2020. *Rancang Bangun Alat Kendali Sortir Barang Berdasarkan Empat Kode Warna*. Jurnal Teknik Elektro dan Vokasional. Volume 06 No.02.
- Santi, Candra Noor. 2011. *Mengubah Citra Berwarna Menjadi Gray-Scale dan Citra Biner*. Jurnal Teknologi Informasi DINAMIK. Volume 16, No.1.
- Simanjuntak, Openyear P. 2019. *Aplikasi Computer Vision Pada Lengan Robot Pemindah Bneda Berdasarkan Ukuran dan Posisi*. Tugas Akhir. Yogyakarta: Universitas Sanata Dharma.
- Sulistiyanti, Sri Ratna, F.X.Arianto Setyawan, Kris Sivam, dan Sri Purwiyanti.. 2019. *Alat Identifikasi Jenis Daging dengan Pengolahan Citra Digital Menggunakan Python 2.7 dan OpenCV Berbasis Raspberry Pi 3*. Jurnal Matrix. Vol.9 (2) pp.54-60.
- Tarigan, Agri Denada Br, dan Iman Setiono. 2018. *Rancang Bangun Sistem Kendali Alat Penyortir Barang Berwarna Merah dan Hijau dengan Sensor TCS230 Berbasis PLC Schneider*. Gema Teknologi. Vol. 20, No.1.
- Wicaksono, Fajar Ridho, Angga Rusdinar, dan Prasetya Dwi Wibawa. 2018. *Perancangan dan Implementasi Alat penyortir Barang pada Konveyor dengan Pengolahan Citra*. e-Proceeding of Engineering. Vol.5 (1) pp. 40-47.
- Widiyanto, Didit. 2020. *Tinjauan Algoritma ROI (Region of Interest) dengan Metode Pengembangan Otsu dan Klasterisasi K-Mean; Hasil dan Tantangannya*. Jurnal Informatik. Edisi ke-16, No.2.
- Xiang Fu, Chen Chen, Jian Li, Changhu Wang, dan C. - C Jay Kuo. 2017. *Image Segmentation using contour, surface, and depth cues*. ICIP. 978-1-5090-2175-8/17.
- Xiaoyu Zhang, Qiuyuan Zhong, Jing Yuan, Hui Gong, dan Qingming Luo. 2016. *Accelerating Sectioning-Based Whole-Brain Imaging by Real-Time Contour Recognition of Cytoarchitecture*. Asia Communications and Photonics Conferene (ACP).
- Yunardi, Riky Tri, Winarno, dan Pujiyanto. 2015. *Contour-based Object Detection in Automatic Sorting System for a parcel Boxes*. ICAMIMIA.