



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EE 184801

**IDENTIFIKASI BARCODE MENGGUNAKAN METODE
OBJECT DETECTION UNTUK PENGAPLIKASIAN PADA
BAGGAGE HANDLING SYSTEM BANDAR UDARA**

Alif Fathsal Muttaqin
NRP 07111640000023

Dosen Pembimbing
Fajar Budiman, ST., M.Sc.
Ronny Mardiyanto, ST., MT., Ph.D.

DEPATERMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EE 184801

**IDENTIFIKASI BARCODE MENGGUNAKAN METODE
OBJECT DETECTION UNTUK PENGAPLIKASIAN PADA
BAGGAGE HANDLING SYSTEM BANDAR UDARA**

Alif Fathsal Muttaqin
NRP 0711164000023

Dosen Pembimbing
Fajar Budiman, ST., M.Sc.
Ronny Mardiyanto, ST., MT., Ph.D.

DEPATERMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



FINAL PROJECT - EE 184801

***BARCODE IDENTIFICATION WITH OBJECT DETECTION
METHOD FOR APPLICATION IN AIRPORT BAGGAGE
HANDLING SYSTEM***

Alif Fathsal Muttaqin
NRP 0711164000023

Supervisor

Fajar Budiman, ST., M.Sc.
Ronny Mardiyanto, ST., MT., Ph.D.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Intelligent Informatics and Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2020

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “*Identifikasi Barcode menggunakan metode object detection untuk pengaplikasian pada Baggage Handling System Bandar Udara*” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 8 Juni 2020

Alif Fathsal Muttaqin
NRP 0711 16 4000 0023

**LEMBAR PENGESAHAN
IDENTIFIKASI BARCODE MENGGUNAKAN
METODE OBJECT DETECTION UNTUK
PENGAPLIKASIAN PADA *BAGGAGE
HANDLING SYSTEM* BANDAR UDARA**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I,



Fajar Budiman, S.T., M.Sc.

NIP. 19860707 201404 1 001

**SURABAYA
JULI, 2020**

LEMBAR PENGESAHAN
IDENTIFIKASI BARCODE MENGGUNAKAN
METODE OBJECT DETECTION UNTUK
PENGAPLIKASIAN PADA *BAGGAGE*
***HANDLING SYSTEM* BANDAR UDARA**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing II,



Ronny Mardiyanto, S.T., M.T., Ph.D.

NIP. 19810118 200312 1 003

SURABAYA
JULI, 2020

Identifikasi Barcode Menggunakan Metode Object Detection untuk Pengaplikasian pada *Baggage Handling System* Bandar Udara

Nama : Alif Fathsal Muttaqin
Pembimbing : 1. Fajar Budiman, ST., M.Sc.
2. Ronny Mardiyanto, S.T., M.T., Ph.D

ABSTRAK

Baggage Handling System (BHS) merupakan salah satu cara dalam proses sekuritas bandara. Dengan BHS, bagasi penumpang dapat dimonitor serta ditracking posisinya secara *realtime*. Umumnya proses pembacaan posisi bagasi ini menggunakan *barcode scanner object* (BSO). Salah satu bandara di Indonesia yang telah menerapkan sistem BHS adalah Terminal 3 Bandar Udara Internasional Soekarno Hatta. Pada Unit BHS T3 Bandara Soekarno-Hatta sering proses pembacaan barcode ini mengalami kegagalan pembacaan.

Pengunaan kamera sebagai instrument penunjang identifikasi barcode pada bagasi bisa menjadi salah satu solusi untuk meminimalisir proses kegagalan identifikasi bagasi. Tugas akhir ini menggunakan kamera webcam sebagai *prototype*. Kamera yang dipasang akan mengcapture bagasi sehingga bisa dideteksi dengan menggunakan metode pengujian *object detection* untuk kemudian bisa diidentifikasi isi informasi pada barcode yang menempel pada bagasi.

Hasil pengujian memperlihatkan bahwa sistem untuk mendeteksi secara optimal pada saat pengaturan sudut kamera pada sudut 45^o dan jarak objek dengan kamera berada pada jarak antar 20 s.d. 30 cm. Kecepatan konveyor yang menggerakkan objek berada pada kecepatan 0,013-0,039 m/s . Hasil pengujian dari metode ini adalah sistem mampu mendeteksi serta mengekstraksi isi informasi pada barcode dengan tingkat akurasi 86%. *Error* diakibatkan oleh pencahayaan dan warna objek. Penggunaan kamera resolusi tinggi dapat diaplikasikan pada penelitian selanjutnya.

Kata Kunci : *Baggage Handling System*, Barcode, Bandar Udara, Object Detection

-----*Halaman ini sengaja dikosongkan*-----

Barcode Identification with Object Detection Method for Application in Airport *Baggage Handling System*

Name : Alif Fathsal Muttaqin
Supervisor : 1. Fajar Budiman, ST., M.Sc.
2. Ronny Mardiyanto, S.T., M.T., Ph.D

ABSTRACT

Baggage Handling System (BHS) is one of the ways in the airport security process. With BHS, passenger baggage can be monitored and tracked in real time. Generally the process of reading baggage position uses a barcode scanner (BSO). One of the airports in Indonesia that has implemented the BHS system is Terminal 3 of Soekarno Hatta International Airport. At the BHS T3 Unit of Soekarno-Hatta Airport, the barcode reading process often fails to read.

The use of the camera as an instrument supporting the identification of barcodes on the trunk can be one solution to minimize the process of baggage identification failure. This final project uses a webcam camera as the prototype. The camera that is installed will capture the luggage so that it can be detected using the object detection testing method to then identify the contents of the information on the barcode attached to the trunk.

The test results show that the system for optimally detecting when setting the camera angle at an angle of 45° and the distance of the object to the camera is at a distance between 20 s.d. 30 cm. The speed of the conveyor moving the object is at a speed of 0.013-0.039 m / s. The test results of this method are the system is able to detect and extract information content on barcodes with an accuracy rate of 86%. Error caused by lighting and color of the object. The use of high resolution cameras can be applied in further research.

Keywords : *Baggage Handling System, Barcode, Airport, Object Detection*

-----*Halaman ini sengaja dikosongkan*-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan tugas akhir yang berjudul “Identifikasi Barcode menggunakan metode object detection untuk pengaplikasian pada *Baggage Handling System* Bandar Udara” Pada kesempatan ini penulis menyampaikan terima kasih atas segala bantuan dan dukungannya yang telah diberikan selama proses pembuatan tugas akhir ini kepada:

1. Kedua orang tua tercinta, Bapak Poermintojo dan Ibu Ida Kuswandari yang selalu memberikan do’a dan dukungan yang sangat berarti dalam keadaan apapun. Semoga Allah selalu memberikan kesehatan.
2. Bapak Fajar Budiman, ST., M.Sc. dan Bapak Ronny Mardiyanto, S.T., M.T.,Ph.D selaku Dosen Pembimbing atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya tugas akhir,
3. Seluruh Staff/Karyawan/Dosen Departemen Teknik Elektro yang telah memberikan banyak ilmu dan menciptakan suasana belajar yang mendukung.
4. Teman-teman e56 yang telah berjuang bersama semenjak awal masuk perkuliahan.
5. Novia Anantya Nugroho, yang selalu memberikan support dan dukungan kepada penulis pada segala kondisi
6. Semua pihak yang telah membantu penulis dalam menyelesaikan tugas akhir ini yang tidak dapat disebutkan satu per satu.

Penulis berharap semoga tugas akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 8 Juni 2020

Penulis

-----*Halaman ini sengaja dikosongkan*-----

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR.....	vii
LEMBAR PENGESAHAN	ix
LEMBAR PENGESAHAN	x
ABSTRAK.....	xi
ABSTRACT	xiii
KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL.....	xxi
BAB I.....	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Permasalahan	3
1.3. Tujuan	3
1.4. Batasan Masalah	3
1.5. Manfaat	4
1.6. Metodologi	4
1.7. Sistematika Penulisan.....	5
1.8. Relevansi.....	6
BAB II	7
TINJAUAN PUSTAKA	7
2.1. <i>Baggage Handling System</i>	7
2.2. Barcode Tag	11
2.3. Video Processing.....	12
2.4. Human Machine Interface – Supervisory Control and Data Acquisition (HMI-SCADA).....	14
2.5. OpenCV	15
2.5.1. Contours	15
2.5.2. Mode Contour	17
2.5.3. Metode Contour	18
2.6. Zbar.....	18
2.7. Logitech C270.....	19
2.8. Arduino Uno	20
2.9. Stepper motor driver A4988.....	21
2.10. Motor stepper 17HS4401	21
BAB III	23

PERANCANGAN SISTEM	23
3.1. Flowchart Sistem	24
3.2. Perancangan Area Pengujian Sistem.....	25
3.3. Perancangan Informasi dalam Barcode.....	26
3.4. Perancangan Sistem Deteksi Bagasi	27
3.4.1. Konversi warna menjadi grayscale	28
3.4.2. Threshold citra menjadi biner	28
3.4.3. Mencari contour object	29
3.4.4. Mendeteksi dan menggambar area contour.....	29
3.5. Perancangan Sistem Identifikasi Barcode	30
3.5.1. Set frame dan image scanner	32
3.5.2. Konversi Warna Menjadi Grayscale	33
3.5.3. Scan Image Barcode	33
3.5.4. Deteksi Jenis Barcode	33
3.5.5. Ekstraksi Informasi yang Terkandung Dalam Barcode	34
BAB IV.....	37
PENGUJIAN DAN ANALISA DATA	37
4.1. Pengujian Detektor Objek Barcode.....	37
4.1.1. Pengujian Berdasarkan Jarak	37
4.1.2. Pengujian Berdasarkan Sudut Kamera Terhadap Objek	39
4.1.3. Pengujian Berdasarkan Ketinggian Kamera	40
4.1.4. Pengujian Berdasarkan Kecepatan Pergerakan Objek	40
4.2. Analisa Data.....	40
BAB V.....	49
PENUTUP	49
5.1. Kesimpulan	49
5.2. Saran	49
DAFTAR PUSTAKA	51
LAMPIRAN	53
BIODATA PENULIS	63

DAFTAR GAMBAR

Gambar 1. 1 Pie Chart Permasalahan BHS pada Juli 2019	2
Gambar 2. 1 Skema perjalanan bagasi dari penumpang hingga ke pesawat pada unit BHS Terminal 3 bandar Udara Soekarno Hatta	8
Gambar 2. 2 Area Check-in Terminal 3 Bandar Udara Soekarno Hatta	8
Gambar 2. 3 Area Identification Unit <i>Baggage Handling System</i> Terminal 3 Bandar Udara Soekarno Hatta	9
Gambar 2. 4 Barcode Scanner Object pada area Identification	9
Gambar 2. 5 Mesin X-Ray Scanner pada unit BHS Terminal 3 Bandar Udara Soekarno Hatta	10
Gambar 2. 6 Mesin Helixorter dan BSO pada area sorting	10
Gambar 2. 7 Area akhir bagasi yang telah berhasil disortir	11
Gambar 2. 8 Area Manual Coding Station	11
Gambar 2. 9 Jenis Barcode 1 Dimensi	12
Gambar 2. 10 Jenis Barcode 2 Dimensi	12
Gambar 2. 11 Contoh Object Detection	13
Gambar 2. 12 Skema HMI-SCADA Unit BHS T3 Bandar Udara Internasional Soekarno-Hatta	14
Gambar 2. 13 Tampilan area monitoring pada unit BHS	15
Gambar 2.14. Hierarki contour dan contour tree	16
Gambar 2.15. Program mencari contour	17
Gambar 2.16. konsep kontur tiap mode	17
Gambar 2. 17 Logitech C270 webcam camera	19
Gambar 2. 18 Arduino Uno	20
Gambar 2.19. A4988 stepper motor driver with dengan heatsink	21
Gambar 2.20. Motor Stepper 17HS4401	21
Gambar 3. 1 Skema Perancangan sistem	23
Gambar 3. 2 Flowchart sistem	24
Gambar 3. 3 Area Identification unit BHS Terminal 3 Bandar Udara Soekarno Hatta	25
Gambar 3. 4 Perancangan area pengujian penelitian tugas akhir	25
Gambar 3. 5 Tampilan website QRCode Generator	26
Gambar 3. 6 Output QRCode dari website	26
Gambar 3.7. Flowchart pendeteksian objek bagasi	27
Gambar 3.8. Program konversi warna menjadi <i>grayscale</i>	28
Gambar 3.9. Program threshold menjadi biner	28
Gambar 3.10. Program pencarian contour	29
Gambar 3.11. Program pendeteksian dan penggambaran contour	30
Gambar 3.12. Hasil deteksi objek bagasi	30

Gambar 3.13	Flowchart sistem identifikasi barcode	31
Gambar 3.14	(a) Barcode yang berhasil di deteksi, (b) Barcode yang tidak berhasil dideteksi	31
Gambar 3.15	Program set image scanner dan frame	32
Gambar 3.16	Hasil set frame pada ukuran 640 x 480 piksel	32
Gambar 3.17	Program konversi warna ke dalam bentuk grayscale	32
Gambar 3.18	Hasil konversi warna ke bentuk grayscale	33
Gambar 3.19	Program scan image barcode.....	34
Gambar 3.20	Program deteksi jenis barcode.....	34
Gambar 3.21	Program decode barcode	35
Gambar 3.22	Hasil deteksi dan ekstraksi informasi barcode	35
Gambar 4.1	Objek yang akan dideteksi	37
Gambar 4.2	Metode pengukuran jarak.....	38
Gambar 4.3	Grafik terdeteksi barcode pada jarak tertentu.....	38
Gambar 4.4	Pengambilan sudut kamera terhadap objek yang berbeda ..	39
Gambar 4.5	Grafik deteksi barcode pada sudut kamera yang berbeda.....	39
Gambar 4.6.	contoh sampel objek dengan warna yang berbeda	41
Gambar 4.7.	Deteksi Bagasi dan Barcode yang benar	41
Gambar 4.8.	Deteksi bagasi dan barcode yang salah	41
Gambar 4.9.	notifikasi kegagalan deteksi sistem	46
Gambar 4.10.	(a) deteksi barcode, (b) deteksi objek yang kurang sempurna, (c) hasil identifikasi isi informasi pada kondisi objek yang kurang sempurna	46

DAFTAR TABEL

Tabel 4. 1 Hasil Pengujian deteksi barcode dalam kondisi bergerak	40
Tabel 4.2. Hasil pengujian pada 15 sampel.....	42
Tabel 4.3. Analisa gambar tidak sempurna.....	43
Tabel 4.4. Analisa hasil gambar >50% sempurna.....	44
Tabel 4.5. Analisa proses gambar sempurna.....	47

\

-----*Halaman ini sengaja dikosongkan*-----

BAB I

PENDAHULUAN

1.1. Latar Belakang

Bandara merupakan kawasan dimana pesawat udara mendarat serta lepas landas. Seiring perkembangan zaman, dengan keuntungan mampu menempuh jarak yang jauh melintasi benua dan samudra dengan waktu yang singkat moda transportasi pesawat udara saat ini makin diminati oleh khalayak masa kini. Bertambahnya jumlah pengguna moda transportasi pesawat udara pun berbanding lurus dengan aktivitas pada bandara. Orang yang ingin berergian selalu membawa barang untuk dimasukkan dalam bagasi. Jumlah bagasi yang makin banyak tentunya juga memunculkan kemungkinan segala bentuk interferensi yang melanggar hukum dan membahayakan keselamatan serta keamanan para penumpang. Untuk itulah diperlukan *Airport Security System* (ASS) yaitu suatu kontrol keamanan yang memiliki proses kompleks dengan beragam divais yang efektivitas serta kecepatannya mampu dirasakan dampaknya oleh para pengguna moda transportasi pesawat. [1].

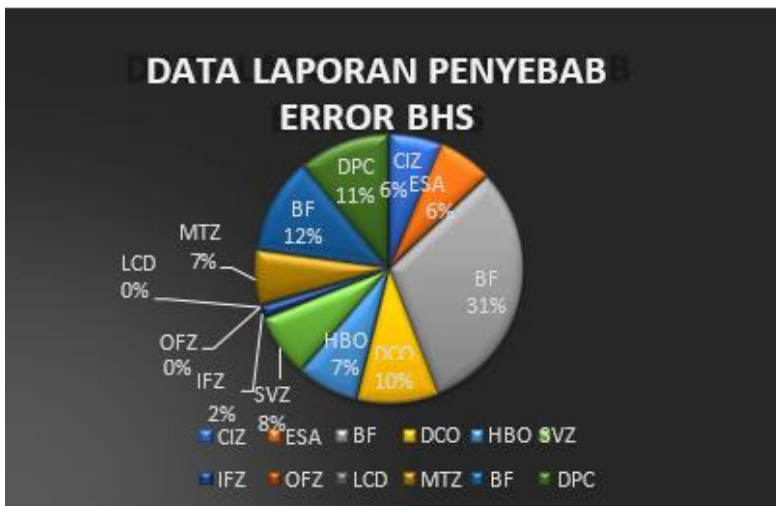
Baggage Handling System (BHS) merupakan salah satu cara dalam proses sekuritas bandara. Dengan BHS, bagasi penumpang dapat dimonitor serta ditracking posisinya secara realtime. Proses pada BHS ini dipengaruhi oleh berbagai macam factor mulai dari kompleksitas sistem, variabel stokastik, serta kendala yang saling bertentangan membuat perlunya analisa terhadap data yang diterima oleh sistem. [2] Umumnya proses pembacaan posisi bagasi ini menggunakan barcode scanner. Barcode yang telah dibaca nantinya akan masuk ke database dan akan ditrack posisinya tiap waktu. Setelah barcode terbaca, bagasi akan disorting sesuai dengan penerbangan masing-masing dan dibawa manual oleh porter ke pesawat.

Salah satu bandara yang telah menerapkan sistem BHS adalah Terminal 3 Bandar Udara Internasional Soekarno Hatta. Pada Unit BHS T3 Bandara Soekarno-Hatta sering proses pembacaan barcode ini mengalami kegagalan pembacaan sehingga mengharuskan bagasi melalui mekanisme manual coding station (MCS). Hal ini bisa jadi disebabkan adanya bagasi yang memiliki banyak barcode (multiple barcode tag) sehingga membuat sistem sulit membedakan barcode yang digunakan pada saat ini dengan barcode penerbangan sebelumnya yang tidak dilepas. Dalam jangka waktu satu hari saja bisa ratusan kasus bagasi yang harus

masuk ke stasiun identifikasi manual. Hal ini tentunya menghambat proses kerja dari sistem.

Setiap harinya ada ribuan hingga puluhan ribu bagasi yang berlalu lintas melalui Unit *Baggage Handling System* Terminal 3 Bandar Udara Internasional Soekarno-Hatta. Lalu lintas bagasi yang cukup padat pada satu hari ini tentunya tak jarang diikuti pula oleh beberapa kendala yang melanda selama proses perjalanan bagasi. Dari data yang didapatkan selama 20 hari terjadi 3275 kasus *Baggage Handling System* mengalami masalah. Dari gambar ditunjukkan penyebab BHS mengalami gangguan sesuai klasifikasinya.

Salah satu permasalahan yang melatar belakangi tugas akhir ini adalah Missing Item. Missing Item adalah kondisi dimana bagasi tidak dapat diidentifikasi sehingga dianggap hilang datanya. Salah satu penyebabnya adalah pemindahan urutan bagasi serta kesalahan pembacaan bagtag pada bagasi. Dari permasalahan ini dari Unit BHS T3 Bandara Soekarno-Hatta telah mengantisipasinya dengan memberikan solusinya adalah dengan cara mengecek barcode bagasi secara manual. Pengecekan bagasi secara manual ini masuk dalam prosedur Manual Coding Station (MCS).



Gambar 1. 1 Pie Chart Permasalahan BHS pada Juli 2019

Pengaplikasian MCS ini tentu sangat membantu identifikasi bagasi yang tidak dapat terlacak barcode tag nya[5]. Namun, penggunaan MCS ini baru dapat digunakan dalam tahap Sorting. Apabila bagasi tidak dapat terbaca pada tahap Identification maka akan dilakukan proses secara mendalam menggunakan mesin khusus dan dicek manual. Hal ini tentunya menurunkan efisiensi waktu serta efektivitas penggunaan mesin barcode scanner yang digunakan pada tahap Identification ini.

Tujuan penggunaan kamera webcam sebagai instrument penunjang identifikasi barcode pada bagasi bisa menjadi salah satu solusi untuk meminimalkan proses kegagalan identifikasi bagasi. Kamera yang dipasang akan mengcapture bagasi pada posisi tertentu yang kemudian akan terlihat secara jelas barcode dari bagasi tersebut sehingga bisa diidentifikasi dengan lebih presisi. [3] [12] Proses deteksi objek merupakan salah satu metode identifikasi yang bisa digunakan. Dalam hal ini barcode yang sesuai dengan jadwal penerbangan akan menjadi objek yang akan kita deteksi. [4]

1.2. Permasalahan

Perumusan masalah pada Tugas Akhir ini adalah sebagai berikut :

1. Bagaimana sistem dapat mendeteksi bagasi
2. Bagaimana sistem dapat mendeteksi *QR code*
3. Bagaimana sistem dapat mengidentifikasi jenis *QR code*
4. Bagaimana sistem bisa melakukan update data secara real time

1.3. Tujuan

Tujuan dari Pembuatan Tugas Akhir ini adalah :

1. Memunculkan metode baru dalam proses identifikasi bagasi
2. Membantu unit *Baggage Handling System* di Bandara untuk meminimalkan salah satu permasalahan yang sering terjadi

1.4. Batasan Masalah

Pembuatan Tugas Akhir ini memiliki beberapa topik serta batasan masalah dalam proses pengerjaannya. Batasan masalah dalam usulan Tugas Akhir ini terletak pada skala pengerjaan yang hanya berupa prototype. Prototype yang akan dirancang berupa pemodelan sistem dari *Baggage Handling System* dalam skala kecil. Beberapa instrument serta variabel yang digunakan belum berada dalam level skala industri.

Adapun instrument serta variabel yang akan diatur sesuai kebutuhan Tugas Akhir ini, antara lain :

1. Penggunaan kamera sebagai instrument penangkap gambar berada pada range frame per second maksimal 30 fps
2. Pergerakan objek akan disimulasikan dengan menggunakan mini conveyor berukuran 40x20 cm dengan kecepatan conveyor maksimal pada 300 rpm
3. Bagasi yang digunakan adalah bagasi berukuran standar dengan ukuran maksimum 1000 x 800 x 600 mm dan berat maksimum 50 kg
4. Proses pengujian secara langsung di Bandara belum dapat dilakukan dikarenakan pandemi COVID-19

1.5. Manfaat

Manfaat dalam pembuatan Tugas Akhir ini adalah sebagai berikut :

1. Dapat memudahkan proses identifikasi bagasi
2. Dapat memberikan informasi serta data yang lebih akurat serta update data secara real time bagi monitoring sistem
3. Referensi metode yang digunakan untuk pengembangan sistem pada unit *Baggage Handling System* di Bandara

1.6. Metodologi

Metodologi yang digunakan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut :

1. Studi Literatur

Studi Literatur sebagai referensi awal sebelum kita mulai untuk merakit alat. Literatur yang digunakan bersumber dari paper, jurnal, serta buku. Literatur yang akan dipelajari ini akan berfokus pada pengkodean barcode, pengolahan objek hasil tangkapan gambar oleh kamera, serta pembuatan sistem monitoring pada Single board computer

2. Perakitan Alat

Tugas Akhir ini akan direalisasikan sebagai prototype atau mini system dari area Identification unit *Baggage Handling System*. Pertama, kamera akan membaca dan mengidentifikasi barcode. Kemudian, hasil pembacaan akan terhubung kepada server utama yang kita definisikan sebagai laptop user disini.

3. Pengambilan Data

Alat yang telah siap kemudian dijalankan untuk kemudian diambil informasi datanya sebagai penunjang untuk proses selanjutnya. Pengambilan data ini akan berkaitan dengan data hasil pembacaan object detection barcode pada bagasi serta proses pengambilan informasi pada barcode apakah sudah bisa membaca

4. Analisis Data dan Evaluasi

Data yang telah diambil kemudian dianalisa untuk diketahui apakah sistem yang dibuat ini telah sesuai dengan percangan awal yang kita inginkan. Bila belum maka kita dapat memperbaikinya sampai mendapat hasil yang kita inginkan. Hasil yang kita inginkan adalah sistem mampu mengidentifikasi bentuk barcode, selanjutnya sistem mampu membedakan barcode yang digunakan pada penerbangan saat itu dengan barcode yang sudah lama tertempel. Dan, terakhir sistem dapat mengekstrasi informasi di dalam barcode dan menampilkannya pada sistem monitoring

5. Penyusunan Buku Laporan

Buku Laporan Tugas Akhir akan disusun begitu semua apa yang kita kerjakan telah selesai serta sesuai dengan apa yang kita inginkan sedari awal.

1.7. Sistematika Penulisan

Penulis membagi laporan penelitian ini menjadi lima bab yang terhubung satu sama lain. Hal ini untuk menghindari kesalahan interpretasi terhadap isi di dalam laporan. Penjelasan tentang masing-masing bab dibuat dengan sistematika penulisan sebagai berikut:

BAB 1 PENDAHULUAN

Pada bab ini dijelaskan tentang latar belakang, permasalahan, batasan masalah, tujuan, metodologi, sistematika penulisan, serta relevansi dari penelitian yang dilakukan.

BAB II TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai tinjauan pustaka yang membantu penelitian. Teori yang dapat membantu penelitian ini antara lain adalah *Baggage Handling System*, object detection.

BAB III PERANCANGAN SISTEM

Pada bab ini membahas tentang perancangan sistem yang akan digunakan pada tugas akhir ini. Perancangan sistem ini meliputi perancangan hardware dan software.

BAB IV PENGUJIAN DAN ANALISIS

Pada bab ini dibahas hasil deteksi Barcode dan juga rekognisi atau identifikasi data barcode. Selanjutnya dilakukan analisa berdasarkan data hasil pengujian.

BAB V PENUTUP

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.8. Relevansi

Hasil dari Tugas Akhir ini merupakan suatu prototype yang kemudian diharapkan bisa diterapkan pada sistem *Baggage Handling System* khususnya pada proses screening identifikasi bagasi tiap Bandar Udara.

BAB II

TINJAUAN PUSTAKA

Bandara merupakan kawasan dimana transportasi udara berlangsung. Kawasan dimana pesawat udara mendarat dan lepas landas. Pesawat komersil pengangkut orang yang mampu membawa orang menuju suatu tempat dengan jarak yang jauh namun mampu ditempuh dalam waktu yang relative cepat membuat pesawat saat ini diminati para traveler. Bertambahnya jumlah penumpang ini tentu berbanding lurus dengan banyaknya lalu lintas bagasi.

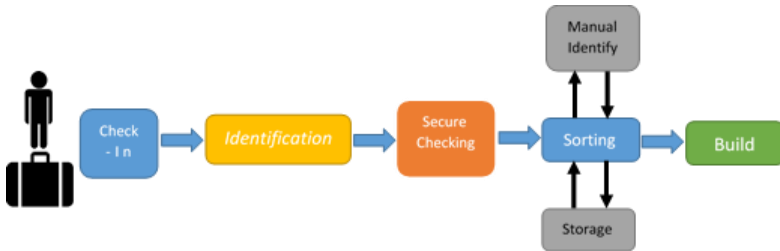
Baggage Handling System (BHS) merupakan suatu unit pada Terminal 3 Bandar Udara Internasional Soekarno-Hatta yang memonitor serta mengatur jalannya lalu lintas bagasi penumpang. Bagasi penumpang yang dibawa saat proses check-in kemudian akan dibawa oleh conveyor melalui beberapa tahapan sebelum bagasi bisa tiba di pesawat sesuai dengan tujuan.

Pada bab ini akan dijelaskan beberapa pustaka penunjang pengerjaan tugas akhir ini mulai dari *Baggage Handling System*, metode pendeteksian yang akan digunakan dan instrumen-instrumen penunjang pengerjaan

2.1. *Baggage Handling System*

Baggage Handling System (BHS) merupakan unit pada Bandar Udara yang memonitor serta mengatur jalannya lalu lintas bagasi penumpang. Seperti diilustrasikan pada gambar BHS memiliki beberapa tahapan perjalanan bagasi mulai dari proses check-in hingga sampai proses pemuatan bagasi ke dalam pesawat.

Check-in adalah proses asal mula penumpang menyerahkan bagasi mereka ke *Baggage Handling System*. Pada saat check-in, sebuah *Baggage Source Message IATA* (BSM) akan dihasilkan oleh DCS maskapai untuk setiap bagasi yang masuk. BSM ini berisi informasi tentang nama penumpang, 10-digit License Plate Number (LPN) dan nomor penerbangan. Informasi BSM ini akan ditransformasikan menjadi sebuah barcode yang kemudian akan ditempelkan pada body bagasi dalam bentuk sebuah bagtag. Ilustrasi area check in pada unit BHS Terminal 3 Bandar Udara Soekarno Hatta dapat dilihat pada gambar 2.2. Bagasi yang telah melalui proses check-in kemudian meneruskan perjalanannya menuju tahap selanjutnya yaitu Identification.



Gambar 2. 1 Skema perjalanan bagasi dari penumpang hingga ke pesawat pada unit BHS Terminal 3 bandar Udara Soekarno Hatta



Gambar 2. 2 Area Check-in Terminal 3 Bandar Udara Soekarno Hatta

Banyak se instrumen untuk mengidentifikasi informasi bisa menggunakan barcode scanner yang melepaskan sinar infrared atau bisa juga menggunakan metode RFID [13]. Unit BHS Terminal 3 Bandar Udara Soekarno Hatta menggunakan instrument *Barcode Scanners Object* (BSO) yang berbasis sinar inframerah untuk mendeteksi serta mengekstrak informasi dari barcode yang menempel pada bagasi. Gambar 2.3. dan Gambar 2.4 menunjukkan area identification beserta alat BSO yang berada pada area tersebut.

Bagasi yang telah berhasil diidentifikasi selanjutnya akan dibawa menuju proses secure checking. Secure checking unit BHS T3 Bandar Udara Soekarno Hatta memiliki 3 level keamanan untuk memeriksa isi bagasi apakah membawa barang yang berbahaya atau tidak. Proses secure checking ini menggunakan mesin X-Ray yang diilustrasikan pada gambar 2.5.

Pada level 1 gambar bagasi yang telah tertangkap oleh mesin X-Ray akan dievaluasi secara otomatis oleh system. Bagasi yang telah lolos proses evaluasi maka akan masuk ke tahap sorting namun apabila bagasi tidak dapat dievaluasi pada level 1 maka akan menuju level 2. Pada level 2, gambar dari mesin X-Ray akan dievaluasi manual oleh operator dan apabila masih belum lolos lagi maka bagasi akan menuju level 3 untuk dicek isinya secara langsung oleh petugas.



Gambar 2. 3 Area Identification Unit *Baggage Handling System* Terminal 3 Bandar Udara Soekarno Hatta



Gambar 2. 4 Barcode Scanner Object pada area Identification



Gambar 2. 5 Mesin X-Ray Scanner pada unit BHS Terminal 3 Bandar Udara Soekarno Hatta



Gambar 2. 6 Mesin Helixorter dan BSO pada area sorting

Sorting merupakan proses terakhir yang akan dilalui bagasi sebelum menuju pesawat masing-masing. Di tahap ini sebuah bagasi akan diklasifikasikan dengan menggunakan mesin helixorter dan BSO. Mesin Helixorter ini dapat dilihat pada gambar 2.6.

Bagasi akan diklasifikasikan menuju areanya masing-masing yang dapat dilihat pada gambar 2.7. Namun apabila bagtag tidak dapat diidentifikasi maka bagasi akan melalui Manual Coding Station (MCS) untuk membaca barcode secara manual agar dapat diidentifikasi lokasi area akhir bagasi. Area manual coding station ini diilustrasikan pada gambar 2.8. Fokus dari penelitian tugas akhir ini berada pada tahap identifikasi *Baggage Handling System*.



Gambar 2. 7 Area akhir bagasi yang telah berhasil disortir



Gambar 2. 8 Area Manual Coding Station

2.2. Barcode Tag

Data informasi merupakan suatu hal penting dalam proses suatu industri. Banyaknya aktivitas suatu perusahaan tentunya makin banyak pula data informasi yang harus dihimpun. Pada masa kini informasi sudah mampu disimpan dalam bentuk yang lebih ringkas. Sebagai contohnya informasi sudah dapat disimpan pada suatu barcode.

Barcode merupakan suatu objek yang mampu menyimpan informasi dalam bentuk kode bergaris. Untuk mengakses informasi yang disimpan dalam suatu barcode dapat digunakan barcode scanner untuk memecahkan kodenya. Banyaknya informasi yang tersimpan dalam barcode bisa kita atur tergantung dengan keinginan pengguna.



Gambar 2. 9 Jenis Barcode 1 Dimensi



Gambar 2. 10 Jenis Barcode 2 Dimensi

Umumnya barcode menggunakan tipe 1 Dimensi (1D Barcode) yang mana barcode akan berorientasi ke arah vertical atau horizontal saja seperti diilustrasikan pada gambar 2.9. Namun, seiring perkembangan informasi yang dibutuhkan maka muncullah 2D Barcode yang mana akan menampilkan barcode dalam orientasi vertical serta horizontal sehingga terlihat barcode berbentuk T. Penggunaan Barcode 2D ini memiliki kelebihan kapasitas informasi yang dibawanya menjadi lebih banyak. [6]. Contoh dari barcode 2 dimensi diilustrasikan pada gambar 2.10.

Baggage Handling System mengidentifikasi bagasi normal dengan menggunakan jenis barcode 2 dimensi yaitu label barcode sesuai IATA resolusi 740. Label bagasi mempunyai kode bentuk-T di mana ada dua barcode identik yang membentuk T. Dasar T mengarah ke tengah label bagasi. Pada penelitian ini barcode yang digunakan adalah barcode 2 dimensi jenis lain yaitu QRCode.

2.3. Video Processing

Dewasa ini manusia semakin dimudahkan dalam mengidentifikasi objek. Manusia yang awalnya harus memerhatikan setiap aktivitas dan melakukan identifikasi melalui kedua bola matanya kini mulai digantikan oleh suatu metode pemrosesan citra. Pemrosesan citra ini mempunyai banyak pilihan dan metode mulai dari berbasis gambar (image processing) atau berbasis video (video processing) sudah dapat diaplikasikan tiap orang yang ingin mengidentifikasi suatu objek.

Video processing merupakan suatu proses pengolahan digital suatu objek video. Video processing membutuhkan gambar bergerak atau video sebagai input untuk kemudian dapat dilakukan proses pengolahan. Untuk

dapat dilakukan proses digitasi maka gambar perlu disampling serta dikuantisasi dalam setaip bit bilangan tetap.[7]

Beragam se metode yang dapat digunakan dalam image processing salah satunya adalah object detection. Object detection merupakan suatu aktivitas memetakan (localizing) serta mengidentifikasi suatu objek dalam sebuah gambar. Object detection ini sudah banyak diaplikasikan salah satunya dalam industri otomotif sebagai deteksi bangunan untuk perancangan sistem assisted and autonomus driving. Variabel pengukuran untuk melihat efisiensi deteksi menggunakan matrik konvesional seperti tingkat presisi serta recall.

Dalam mengontrol performansi dalam input video kita menggunakan 4 indikator kemungkinan hasil yaitu, True Positive (TP), False Positives (FP), True Negatives (TN), serta False Negative (FN). Menggunakan teknologi deep networks seperti Inception yang mempunyai ratusan lapisan konvolusioal, menjadi salah satu kekurangan utama pada teknik baru sehubungan dengan algoritma tradisional yang kebutuhan daya untuk komputasi serta memorinya secara signifikan bertambah. Untuk mengurangi kebutuhan hardware yang beragam para peneliti menemukan beragam metode kompresi model seperti quantization, hashing, dan lain sebagainya bahkan bisa sepenuhnya membuat arsitektur yang berbeda.[4]



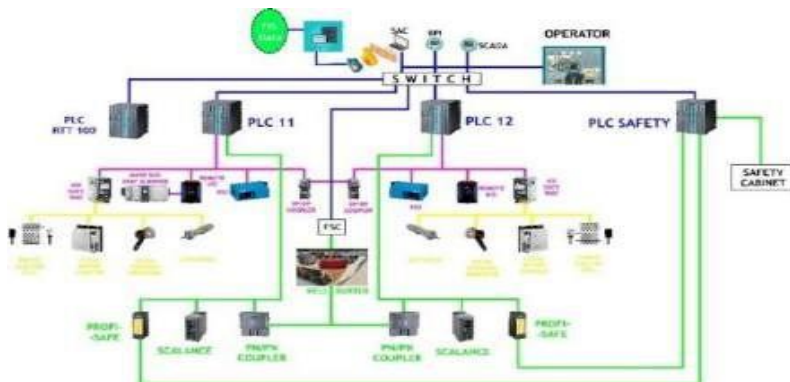
Gambar 2. 11 Contoh Object Detection

2.4. Human Machine Interface – Supervisory Control and Data Acquisition (HMI-SCADA)

Perkembangan teknologi dalam suatu industri masa kini yang semakin pesat mengakibatkan industri terus berkembang. Penggunaan sistem kontrol, visualisasi data operasi serta pelaporan kondisi operasional menjadi pusat perhatian seluruh industri di dunia. Sistem yang serba otomatis serta monitoring sistem melalui jarak jauh menjadi fokus yang dilakukan oleh Human Machine Interface (HMI) / Supervisory Control and Data Acquisition (SCADA). Dengan menggunakan sistem HMI-SCADA perkembangan teknologi beserta implementasinya menjadi lebih kompleks, dimana kita memerlukan pula standar industri untuk meningkatkan keamanan sistem.

Secara umum HMI-SCADA terdapat dua komponen utama yaitu Sistem Komunikasi dan Teknologi software sistem. Perkembangan sistem komunikasi turut serta mengambil peran besar dalam implementasi sistem HMI/SCADA pada industri. Terdapat dua level komunikasi yang digunakan pada sistem komunikasi ini yaitu, teknologi informasi dan field.

Pada tahap Identification, saat bagasi masuk ke area pembacaan barcode akan muncul suatu notifikasi pada sistem yang memperlihatkan bahwa bagasi telah terbaca dan bisa melanjutkan ke proses selanjutnya yaitu Security Checking. Pada tampilan SCADA akan memunculkan pemberitahuan bahwa bagasi ini memuat informasi apapun mulai dari nama penumpang, maskapai, hingga kode penerbangan.



Gambar 2. 12 Skema HMI-SCADA Unit BHS T3 Bandar Udara Internasional Soekarno-Hatta



Gambar 2. 13 Tampilan area monitoring pada unit BHS

2.5. OpenCV

OpenCV (Open Source Computer Vision Library) merupakan sebuah pustaka opensource untuk pengolahan computer vision maupun machine learning. OpenCV diciptakan untuk menyediakan sebuah infrastruktur dasar untuk aplikasi visi computer dan untuk meningkatkan penggunaan dari persepsi mesin pada produk komersial. OpenCV merupakan produk berlisensi BSD sehingga mudah untuk digunakan serta modifikasi kode.[8]

OpenCV mempunyai lebih dari 2500 algoritma yang meliputi sebuah set komprehensif dari algoritma machine learning serta visi computer. Algoritma pada openCV bisa digunakan untuk beragam fungsi seperti deteksi objek, rekognisi wajah, maupun tracking pergerakan kamera. OpenCV mampu digunakan dalam beragam bahasa pemrograman seperti C++ ataupun python. Pada penelitian tugas akhir ini digunakan openCV versi 4.1.2 dan bahasa pemrograman C++.

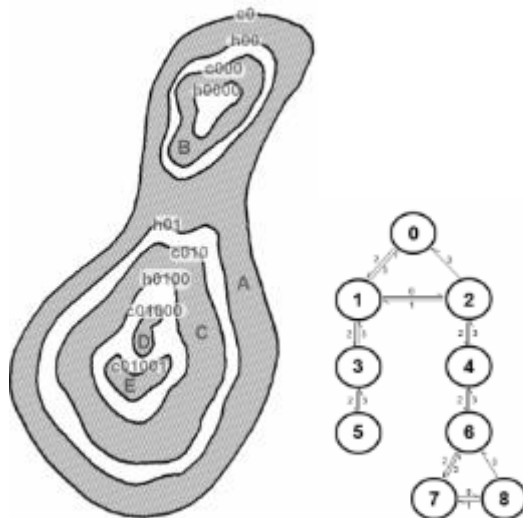
2.5.1. Contours

Contour adalah sebuah kumpulan titik yang merepresentasikan dalam satu arah atau yang lainnya menjadi sebuah kurva gambar. Representasi ini dapat berbeda-beda tergantung pada keadaan yang didapatkan. [14] Terdapat banyak cara dalam merepresentasikan sebuah kurva. Pada OpenCV, contours direpresentasikan dalam bentuk template

objek STL `vector< >` yang membuat setiap data yang masuk pada vector akan diekstraksi kode informasi tentang lokasi dari titik selanjutnya pada kurva tersebut. Kita dapat merepresentasikan menjadi sebuah urutan titik 2 dimensi dengan menggunakan perintah (`vector<cv::Point>` atau `vector<cv::Point2f>`).

Saat mendeklarasikan fungsi `cv::findContours()` akan mengkomputasi kontur dari sebuah gambar biner. Gambar biner dapat dikonversi dengan menggunakan perintah `cv::Canny()` yang mempunyai batas di tiap piksel. Selain itu, bisa juga menggunakan fungsi `cv::threshold` atau `cv::adaptiveThreshold()` yang tiap tepinya secara implisit membatasi antara daerah positif dan negatif.

Hierarki dari sebuah kontur dapat direpresentasikan dalam sebuah pohon kontur seperti ditunjukkan dalam gambar 2.14. Konsep dari pengurungan daerah daerah ini bisa dikatakan seperti mengumpulkan kontur dalam sebuah *contour tree* (pohon kontur). Pohon kontur ini akan mengekstraksi kode yang membatasi hubungan di tiap strukturnya. Pohon kontur ini bergantung pada gambar yang diujikan dan tiap contour disebut sebagai `c0` di tiap *root node*. Dalam mencari contour kita dapat mendeklarasikannya sesuai dalam penjelasan di gambar 2.15.



Gambar 2.14. Hierarki contour dan contour tree

```

Void cv::findContours (
    cv::InputOutputArray image => input biner channel tunggal 8 bit
    cv::OutputArrayOfArrays contours => output vektor dari sebuah
    vektor atau titik

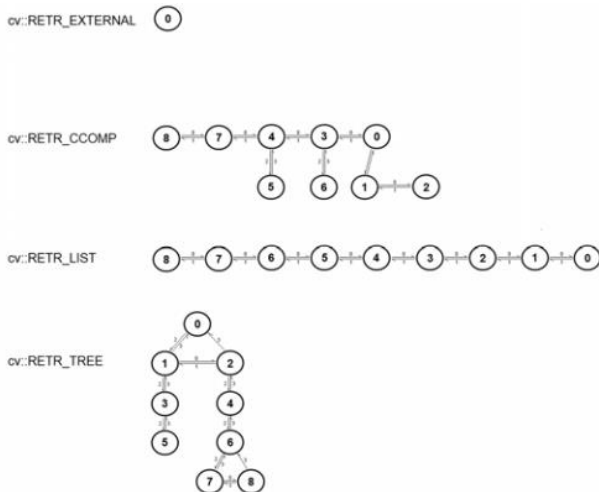
    int mode => mode pengambilan contour

    int method => metode penaksiran contour

    cv::Point offset =cv::Point() => offset setiap titik

```

Gambar 2.15. Program mencari contour



Gambar 2.16. konsep kontur tiap mode

2.5.2. Mode Contour

RETR merupakan mode dalam mengambil contour. Terdapat 4 mode yang dapat kita pilih sesuai kebutuhan yaitu, `cv::RETR_TREE`, `cv::RETR_EXTERNAL`, `cv::RETR_LIST`, dan `cv::RETR_CCOMP`. Setiap mode memiliki keunggulan dan juga cara dalam mengambil kontur yang berbeda-beda pula

RETR_CCOMP merupakan mode yang mengambil semua kontur dan mengaturnya menjadi hierarki dua tingkat, di mana batas-batas tingkat atas berada batas eksternal komponen dan batas tingkat kedua adalah batas lubang. RETR_TREE merupakan mode yang mengambil semua kontur dan merekonstruksi hierarki penuh kontur bersarang.

RETR_LIST merupakan mode yang mengambil semua kontur dan menempatkannya dalam daftar. RETR_EXTERNAL merupakan mode yang mengambil hanya bagian terluar dari contour. Secara sederhana konsep dari tiap contour dapat dilihat pada gambar 2.16.

2.5.3. Metode Contour

Metode merupakan cara dari program untuk menaksir atau memperkirakan area-area mana saja yang merupakan sebuah contour. Beberapa metodenya seperti CHAIN_APPROX_NONE merupakan metode yang menerjemahkan seluruh poin dari kode kontur menjadi sebuah titik. Operasi ini akan menghasilkan jumlah titik yang cukup besar dan setiap titiknya akan menjadi satu dari delapan tetangga titik sebelumnya. Tidak ada percobaan yang dilakukan untuk mengurangi jumlah dari simpul yang dikembalikan.

Metode selanjutnya adalah CHAIN_APPROX_SIMPLE yang akan mengkompresi segmen horizontal, vertikal, dan diagonal dari input dan metode ini akan menyisakan tiap titik akhirnya. Untuk beberapa kasus khusus, metode ini dapat menghasilkan sebuah pengurangan substansial dari jumlah titik yang kembali. Contoh ekstrim adalah menghasilkan sebuah kotak berapapun ukurannya berorientasi sepanjang sumbu x dan sumbu y yang akan mengembalikan 4 titik.

Metode terakhir adalah metode CHAIN_APPROX_TC89_L1 atau CHAIN_APPROX_TC89_KCOS yang akan mengaplikasikan salah satu dari algoritma pendekatan rantai Teh-Chin. Algoritma T-C ini merupakan metode yang lebih canggih dan juga mengkomputasikan secara intensif untuk mengurangi jumlah titik yang dikembalikan. Algoritma T-C tidak membutuhkan parameter tambahan untuk dijalankan.

2.6. Zbar

Zbar merupakan salah satu jenis pustaka pada openCV yang bersifat opensource. Zbar digunakan sebagai tambahan pustaka apabila pengguna ingin mendeteksi atau mengekstraksi informasi sebuah barcode. Pustaka ZBar dapat diunduh secara gratis melalui website penyediannya yaitu sourceforge.

2.7. Logitech C270

Kamera menjadi instrument penangkap citra objek sebagai input system. Logitech merupakan salah satu perusahaan yang bergerak pada bidang penyedia produk kamera. Variasi produk dari Logitech sangatlah beragam dan mengharuskan kita benar benar mencermati setiap spesifikasi pada masing-masing kamera produksi dari perusahaan ini.

Pada penelitian ini kamera yang digunakan adalah tipe webcam Logitech C270. Kamera ini merupakan tipe kamera webcam yang mempunyai keunggulan pada bentuknya yang ramping serta fleksibel sehingga mampu untuk diletakkan pada beragam posisi. Dengan menggunakan kabel USB sepanjang 1,5 meter kamera ini sangatlah fleksibel untuk diletakkan pada beragam posisi. Kamera ini memiliki keunggulan pada poros kamera yang mampu digerakkan sehingga dapat diatur sendiri sudut pengambilan gambar sesuai kebutuhan.

Layar lensa kamera ini mampu diatur kedalam mode widescreen yang berarti cakupan sudut pandang dari kamera ini cukup luas. Logitech webcam C270 mempunyai spesifikasi Field of View (FoV) sebesar 60° yang berarti cukup luas daya cakupan kamera. Kamera ini merupakan jenis kamera fixed-focus. [9]

Kamera ini memiliki kelebihan kelebihan built-in audio microphone tipe mono sehingga bisa digunakan untuk keperluan perekaman video yang membutuhkan input audio. Kamera ini memiliki resolusi maksimum sebesar 720p serta penangkapan video maksimum berada pada angka 30 frame per second.[9]. Gambar dari logitech C270 dapat dilihat pada gambar 2.17



Gambar 2. 17 Logitech C270 webcam camera



Gambar 2. 18 Arduino Uno

2.8. Arduino Uno

Arduino merupakan platform elektronik *open-source* yang mudah digunakan untuk hardware ataupun software. Board arduino mempunyai kemampuan untuk membaca beragam input seperti sentuhan jari pada sebuah tombol, dan juga bisa mengubahnya menjadi sebuah output seperti untuk menggerakkan motor, menyalakan lampu LED dan mengirimkan data secara online. Kita dapat mengirimkan sebuah set instruksi kepada mikrokontroler pada board arduino dengan menggunakan bahasa pemrograman arduino dan aplikasi software Arduino IDE. [10]

Pada masa kini, arduino telah berkembang secara pesat untuk berkontribusi kepada para pengguna di seluruh dunia. Hal ini dikarenakan banyak se keunggulan arduino dibandingkan system yang lain antara lain harganya yang tidak mahal, softwarentya yang simple dan mudah untuk dipelajari. Keunggulan tambahan dari arduino ini dikarenakan bersifat *opensource* sehingga mudah untuk ditambahkan sebuah hardware ataupun software eksternal.

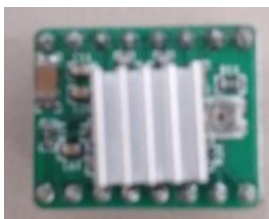
Arduino Uno merupakan salah satu tipe board arduino yang paling banyak digunakan dan mudah untuk dipelajari bagi para pemula. Arduino Uno merupakan board mikrokontroler yang berbasis pada ATmega328P. Spesifikasi dari tipe ini adalah mempunyai 14 pin digital input/output yang 6 diantaranya bisa digunakan sebagai output PWM, 6 pin analog input, 16 MHz resonator keramik, koneksi USB, penghubung daya DC, Header ICSP dan sebuah tombol reset. Gambar dari arduino uno dapat dilihat pada gambar 2.18

2.9. Stepper motor driver A4988

A4988 merupakan sebuah driver motor microstepping yang telah dilengkapi built-in translator sehingga mudah untuk dioperasikan. Ilustrasi dari stepper motor ini dapat dilihat pada gambar 2.19 Driver ini didesain untuk mengoperasikan motor stepper bipolar dalam beragam mode mulai dari fullstep-mode, halfstep-mode, bahkan sixteenth step-mode.¹

2.10. Motor stepper 17HS4401

Motor stepper merupakan penggerak conveyor pada pengerjaan tugas akhir ini. Motor stepper tipe 17HS4401 (gambar 2.20 menjadi pilihan dalam penelitian ini. Tipe 17HS4401 merupakan sebuah motor 2 fasa bipolar. Motor stepper ini mempunyai torsi tahanan sebesar 40 Ncm. Memiliki arus rata-rata sebesar 1.7 A motor ini mampu menggerakkan mini conveyor.²



Gambar 2.19. A4988 stepper motor driver with dengan heatsink



Gambar 2.20. Motor Stepper 17HS4401

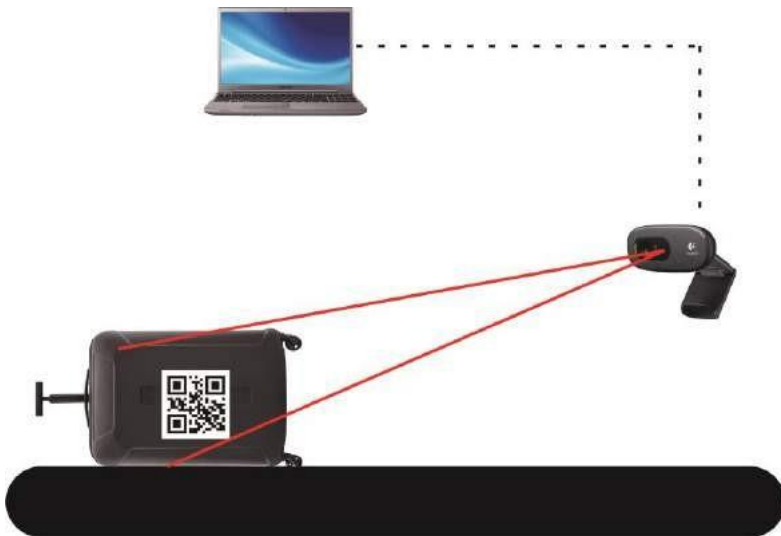
¹, DMOS Microstepping Driver with Translator and Overcurrent Protection, Allegro MicroSystems, 2014, hal.1

², 2 Phase Hybrid Stepper Motor 17HS Series, MotionKing (China) Motor Industry Co., Ltd,

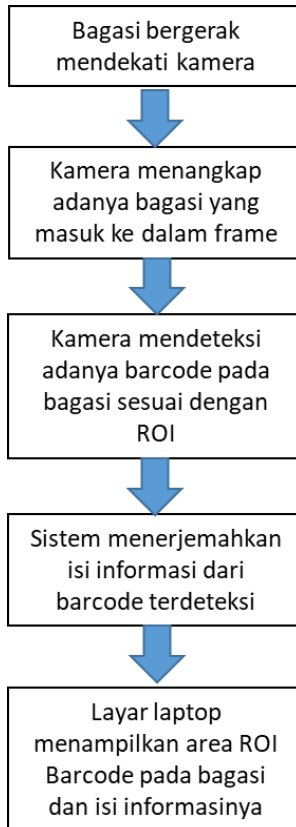
-----*Halaman ini sengaja dikosongkan*-----

BAB III PERANCANGAN SISTEM

Bab ini akan menjelaskan tentang perancangan system untuk skenario pembacaan bagtag pada sebuah bagasi. System yang dirancang pada penelitian tugas akhir ini merupakan prototype dari unit BHS Terminal 3 Bandar Udara Soekarno Hatta. Skema keseluruhan prototype ini dapat dilihat pada gambar.3.1. Sistem ini menggunakan kamera webcam serta laptop. Kamera webcam digunakan sebagai pengambil gambar citra bagasi yang telah ditempeli oleh bagtag. Laptop digunakan sebagai pemroses pengolahan citra untuk deteksi serta mengekstraksi isi informasi pada bagasi. Laptop akan menampilkan kondisi bagasi yang berhasil ditangkap kamera serta akan menunjukkan area dimana barcode berhasil dideteksi. Sehingga operator dapat memantau kondisi bagasi serta hasil decode dari layar laptop.



Gambar 3. 1 Skema Perancangan sistem



Gambar 3. 2 Flowchart sistem

3.1. Flowchart Sistem

Pada bagian ini akan ditampilkan flowchart system yang digunakan dalam penelitian tugas akhir ini seperti yang ditampilkan pada gambar 3.2.

Penelitian ini dengan menggunakan kamera yang didapatkan frame input gambar sebesar 640x480 piksel. Setelah gambar ditangkap kemudian system mengolah gambar tersebut menggunakan bantuan pustaka openCV dan zbar. Sistem kemudian akan mendeteksi serta mengekstraksi isi informasi pada barcode. Hasil dari identifikasi ini akan ditampilkan pada layar laptop operator.

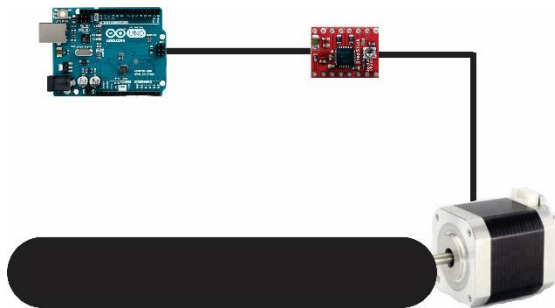
3.2. Perancangan Area Pengujian Sistem

Penelitian tugas akhir ini dilakukan beberapa penyesuaian dalam menentukan area pengujian system dikarenakan ini merupakan sebuah prototype ataupun mini system dari unit BHS Terminal 3 Bandar Udara Soekarno Hatta. Kondisi sebenarnya dilakukan pada area identification unit BHS seperti diilustrasikan pada gambar 3.3. sedangkan pada penelitian ini area diset seperti pada gambar 3.4.

Pada perancangan area ini objek akan digerakkan melalui sebuah mini conveyor yang menggunakan bantuan tenaga penggerak dari sebuah motor stepper. Jarak pergerakan objek menuju kamera dimulai dari jarak 40 cm yang akan terus bergerak hingga objek telah melewati kamera. Kamera diset setinggi 30 cm diatas permukaan tanah.



Gambar 3. 3 Area Identification unit BHS Terminal 3 Bandar Udara Soekarno Hatta



Gambar 3. 4 Perancangan area pengujian penelitian tugas akhir



Gambar 3. 5 Tampilan website QRCode Generator



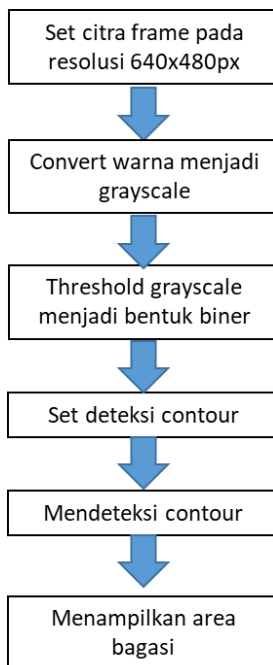
Gambar 3. 6 Output QRCode dari website

3.3. Perancangan Informasi dalam Barcode

Barcode merupakan objek paling penting dalam penelitian tugas akhir ini. Barcode sebagai objek pembawa informasi yang akan diidentifikasi oleh system. Pada unit BHS Terminal 3 Bandar Udara Soekarno Hatta menggunakan jenis barcode sesuai resolusi IATA 740 yang didapatkan dari system DCS masing-masing maskapai. Dikarenakan barcode sesuai resolusi IATA 740 ini mempunyai system tersendiri sehingga terdapat beberapa penyesuaian dalam penelitian tugas akhir ini. Barcode resolusi IATA merupakan barcode 2 dimensi sehingga pada penelitian ini kami menggunakan jenis barcode 2 dimensi yang lain yaitu QR Code. QR Code sendiri diproduksi sendiri dengan menggunakan bantuan halaman website QCode-generator yang dapat diakses secara gratis.

3.4. Perancangan Sistem Deteksi Bagasi

Deteksi merupakan proses mencari dan menemukan keberadaan objek sesuai kriteria yang telah ditetapkan. Pada penelitian ini bagasi menjadi objek yang akan dideteksi untuk diketahui kemudian apakah ada barcode yang berada pada bagasi tersebut. Tujuan dari pendeteksian objek bagasi ini selain untuk mengetahui apakah ada objek yang melewati conveyor adalah untuk mengantisipasi kemungkinan kegagalan identifikasi barcode sehingga operator bisa lebih mudah dalam memonitor bagasi mana yang gagal teridentifikasi. Pada penelitian tugas akhir ini disesuaikan dengan kemampuan *Field of View* (FOV) dari kamera sehingga menggunakan pustaka openCV *findcontour*. Alur dari sistem pendeteksian bagasi ini akan dijelaskan pada subbab berikut. Flowchart pendeteksian bagasi dapat dilihat pada gambar 3.7.



Gambar 3.7. Flowchart pendeteksian objek bagasi

3.4.1. Konversi warna menjadi grayscale

Proses pertama dalam pendeteksian bagasi ini adalah mengkonversi warna menjadi *grayscale* untuk memudahkan proses pendeteksian dan penguatnisasi bit menjadi biner pada proses selanjutnya. Program dari pengkonversian warna menjadi *grayscale* dapat dilihat pada gambar 3.8. pada program ini kita menggunakan fungsi `cv::cvtColor` untuk mengkonversi warna. Masukkan input image yang akan diproses yaitu `detek`. Lalu kita mendeklarasikan output array dari hasil pemrosesan ini yaitu `grey`. Dan terakhir kita memasukkann metode konversi warnanya karena kita akan mengubahnya menjadi grayscale maka kodenya adalah `COLOR_BGR2GRAY`

3.4.2. Threshold citra menjadi biner

Proses selanjutnya adalah menentukan threshold gambar yang ditangkap oleh kamera menjadi bentuk yang lebih sederhana. Citra gambar yang telah menjadi *grayscale* kemudian diubah menjadi bentuk biner sehingga lebih memudahkan dalam proses pencarian contour nantinya. Pada program ini nilai maksimal threshold diatur pada angka 255 dan juga nilai threshold diatur pada 110 agar tidak terlalu jauh perbedaan kuantisasi binernya sehingga pendeteksi objek nanti tidak salah mendeteksi hal yang tidak sesuai dengan tujuan kita. Program dari threshold citra ini dapat dilihat pada gambar 3.9.

```
cap >> detek;  
if (detek.empty()) break;  
cvtColor(detek, grey, COLOR_BGR2GRAY);
```

Gambar 3.8. Program konversi warna menjadi *grayscale*

```
threshold(grey, biner, 110, 255, THRESH_BINARY);  
int edgeThresh = 1;
```

Gambar 3.9. Program threshold menjadi biner

```
vector<vector<Point>>contours;  
Mat finderCount = biner.clone();  
findContours(finderCount, contours, RETR_TREE,  
CHAIN_APPROX_SIMPLE);
```

Gambar 3.10. Program pencarian contour

3.4.3. Mencari contour object

Proses setelah gambar diubah menjadi biner adalah pencarian contour dari objek yang kita dideteksi. Langkah pertama adalah kita mendefinisikan contour sebagai vector point. Selanjutnya kita menkloning hasil biner tadi untuk kemudian dicari contournya menggunakan metode RETR_TREE. Metode RETR_TREE adalah proses pengambilan semua kontur lalu merekonstruksi seluruh hierarkinya berdasar kontur bersarang. Program dari pencarian contour ini dapat dilihat pada gambar 3.10.

3.4.4. Mendeteksi dan menggambar area contour

Langkah terakhir dari pendeteksi objek ini adalah mendeteksi serta menggambar area contour. Pada program ini kita akan mengatur area minimum contour baru kita bisa menggambar contour pada gambar. Pendeteksian ditunjukkan dengan fungsi DetectContour. Kita mendaklarasikan minimum area contournya adalah 0 dan akan terus disesuaikan dengan countours size yang dideteksi. Kemudian setelah contour berhasil dideteksi sesuai areanya kita akan menggambar hasil pendeteksiannya menggunakan fungsi drawContours. Pada proses penggambaran ini menggunakan warna hijau Program bisa dilihat pada gambar 3.11. Hasil dari pendeteksian contour ditunjukkan oleh gambar 3.12.

```

Mat DetectContour(detek.size(), CV_8UC3, Scalar(0,
0, 0));

int areamin = 0;
int dtcup = 0;
for (int a = 0; a < contours.size(); a++)
{
    if (contourArea(contours[a]) > areamin)
    {
        areamin = contourArea(contours[a]);
        dtcup = a;
    }
}

drawContours(detek, contours, dtcup, Scalar(0, 255,
0), 5, 8);

```

Gambar 3.11. Program pendeteksian dan penggambaran contour



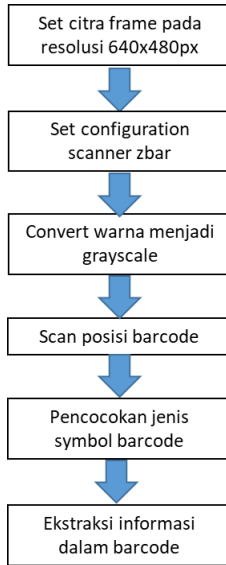
(a)

(b)

Gambar 3.12. Hasil deteksi objek bagasi

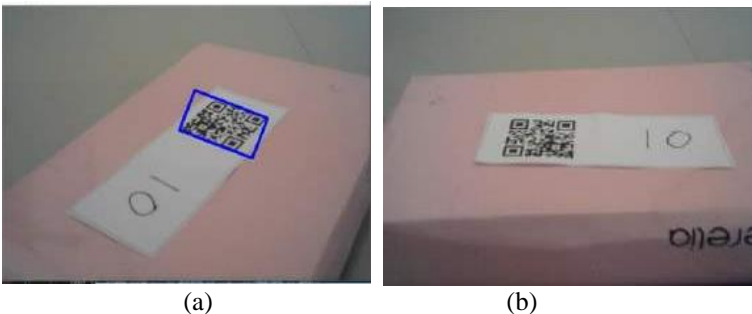
3.5. Perancangan Sistem Identifikasi Barcode

Sistem Identifikasi barcode pada penelitian tugas akhir ini menggunakan pustaka zbar dan openCV. Pustaka openCV digunakan untuk pemrosesan citra gambar dari webcam. Pustaka zbar digunakan untuk deteksi jenis barcode serta mengekstrak informasi yang ada di dalam barcode. Flowchart dari sistem identifikasi barcode dapat dilihat pada gambar 3.13.



Gambar 3.13 Flowchart sistem identifikasi barcode

Hasil identifikasi barcode ini memiliki beragam hasil. Contoh dari gambar yang berhasil di deteksi diilustrasikan pada gambar 3.14(a). sedangkan gambar yang tidak berhasil dideteksi pada gambar 3.14(b)



Gambar 3.14 (a) Barcode yang berhasil di deteksi, (b) Barcode yang tidak berhasil dideteksi

```
ImageScanner scanner;  
scanner.set_config(ZBAR_NONE, ZBAR_CFG_ENABLE, 1);  
double dWidth = cap.get(CAP_PROP_FRAME_WIDTH);  
double dHeight = cap.get(CAP_PROP_FRAME_HEIGHT);
```

Gambar 3. 15 Program set image scanner dan frame

3.5.1. Set frame dan image scanner

Pada penelitian tugas akhir ini langkah pertama yang dilakukan adalah menentukan frame yang akan digunakan sebagai area untuk menangkap citra objek. Image scanner yang digunakan adalah bagian dari pustaka zbar. Set program dapat dilihat pada gambar 3.15. dan hasil set frame pada ukuran 640x480 piksel diilustrasikan pada gambar 3.16



Gambar 3. 16 Hasil set frame pada ukuran 640 x 480 piksel

```
Mat frame;  
Mat gray;  
cap.read(frame);  
cvtColor(frame, gray, COLOR_BGR2GRAY);
```

Gambar 3. 17 Program konversi warna ke dalam bentuk grayscale



Gambar 3. 18 Hasil konversi warna ke bentuk grayscale

3.5.2. Konversi Warna Menjadi Grayscale

Untuk memudahkan proses pendeteksian serta ekstraksi isi informasi maka citra gambar yang didapatkan dari kamera diubah ke dalam bentuk grayscale. Program pengubahan ini dapat dilihat pada gambar 3.17. Kita menggunakan fungsi `cv::cvtColor` untuk mengkonversi warna. Masukkan input image yang akan diproses yaitu `frame`. Lalu kita mendeklarasikan output array dari hasil pemrosesan ini yaitu `gray`. Dan terakhir kita memasukkann metode konversi warnanya karena kita akan mengubahnya menjadi grayscale maka kodenya adalah `COLOR_BGR2GRAY`

3.5.3. Scan Image Barcode

Gambar yang telah dikonversi menjadi grayscale ini kemudian akan di-wrap agar sesuai dengan set frame baris dan kolom yang telah kita tentukan. Deklarasikan sebuah fungsi untuk menentukan lebar dan tinggi dari kolom frame gambar input. Gambar tersebut kemudian akan dibuat dalam bentuk *uchar* untuk memudahkan proses scanning. Lalu gambar akan discan untuk dideteksi jenis serta lokasi barcode berada. Program pada tahap ini ditunjukkan oleh gambar 3.19.

3.5.4. Deteksi Jenis Barcode

Objek yang sudah discan pada tahap sebelumnya kemudian akan di deteksi jenis/symbol barcodenya. Penentuan jenis barcode ini akan memudahkan proses ekstraksi informasi yang terkandung di dalam barcode. Gambar 3.20. menunjukkan program deteksi jenis barcode. Program deteksi jeni barcode ini akan dicocokkan dengan library header `imagescanner` yang telah dideklarasikan di awal. Fungsi `symbolIterator` ini untuk membuat sistem terus berjalan selama bentuk barcode yang terdeteksi belum mengalami kecocokan. Apabila suda cocok maka `symbol vector` akan disimpan dalam `vector<Point> vp`

3.5.5. Ekstraksi Informasi yang Terkandung Dalam Barcode

Barcode yang telah berhasil dideteksi jenisnya kemudian akan didecode atau diekstraksi informasi yang terkandung di dalamnya. Seperti ditunjukkan pada gambar 3.21. yang menunjukkan program ekstraksi informasi. Pada program tersebut fungsi *symbol* menunjukkan symbol yang didapatkan pada gambar 3.20. Barcode yang telah berhasil diidentifikasi jenis simbolnya kemudian akan didecode untuk diekstraksi isi informasi yang dibawanya melalui fungsi *vp.pushback*. Fungsi *vp.pushback* juga akan menunjukkan lokasi dari barcode berada yang kemudian akan digambar sebuah kotak 2 dimensi berwarna biru melalui fungsi *RotatedRect r= minAreaRect(vp)*. Hasil dari ekstraksi informasi dan penggambaran lokasi barcode ditampilkan pada gambar 3.22.

```
int width = frame.cols;
int height = frame.rows;
uchar* raw = (uchar*)gray.data;

Image image(width, height, "Y800", raw, width * height);

// scan the image for barcodes
int n = scanner.scan(image);
```

Gambar 3. 19 Program scan image barcode

```
for (Image::SymbolIterator symbol = image.symbol_begin();
     symbol != image.symbol_end();
     ++symbol)
    {
        vector<Point> vp;
```

Gambar 3. 20 Program deteksi jenis barcode


```

cout << "decoded " << symbol->get_type_name() << " symbol "
<< symbol->get_data() << endl;
int n = symbol->get_location_size();
for (int i = 0; i < n; i++)
{
vp.push_back(Point(symbol->get_location_x(i),          symbol-
>get_location_y(i)));
}
RotatedRect r = minAreaRect(vp);
Point2f pts[4];
r.points(pts);
for (int i = 0; i < 4; i++)
{
line(frame, pts[i], pts[(i + 1) % 4], Scalar(255, 0, 0), 3);
}

```

Gambar 3. 21 Program decode barcode



Gambar 3. 22 Hasil deteksi dan ekstraksi informasi barcode

-----*Halaman ini sengaja dikosongkan*-----

BAB IV

PENGUJIAN DAN ANALISA DATA

Bab ini akan membahas tentang metode pengujian serta analisa data yang digunakan pada penelitian ini yaitu identifikasi barcode menggunakan metode object detection untuk pengaplikasian pada unit *Baggage Handling System* Bandar Udara. Tujuan dari pembahasan pada bab ini yaitu untuk mengetahui apakah system yang telah dirancang sedemikian rupa ini mampu menyelesaikan permasalahan atau tidak.

4.1. Pengujian Detektor Objek Barcode

Pengujian detector objek barcode ini untuk menentukan seberapa akurat system mampu mendeteksi barcode pada suatu objek. Penelitian ini saya menggunakan kardus sebagai pengganti bagasi serta untuk barcode tag menggunakan hasil print gambar QRCode yang telah dibuat. Ilustrasi objek yang dideteksi dapat dilihat pada gambar 4.1.



Gambar 4. 1 Objek yang akan dideteksi

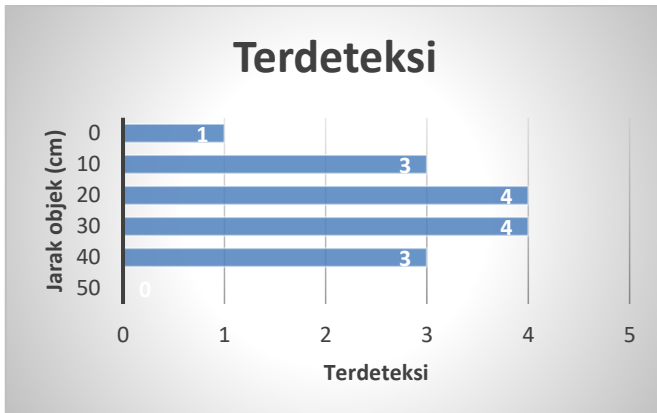
4.1.1. Pengujian Berdasarkan Jarak

Jarak menjadi salah satu parameter penentuan deteksi objek. Penelitian ini diukur dalam jarak antara 0 s.d. 100 cm. Pengukuran dilakukan setiap 10 cm untuk mengetahui titik dimana kamera mampu mendeteksi barcode pada sebuah objek. Ilustrasi dari metode pengujian jarak ini ditampilkan pada gambar 4.2.

Dari hasil pengujian seperti diilustrasikan pada gambar 4.3. didapati bahwa system mampu mendeteksi system pada jarak antara 0 s.d. 40 cm dengan memperhatikan sudut kamera terhadap objek. Hasil terbaik ketika barcode berada pada jarak 20-30 cm dari kamera.



Gambar 4. 2 Metode pengukuran jarak



Gambar 4. 3 Grafik terdeteksi barcode pada jarak tertentu

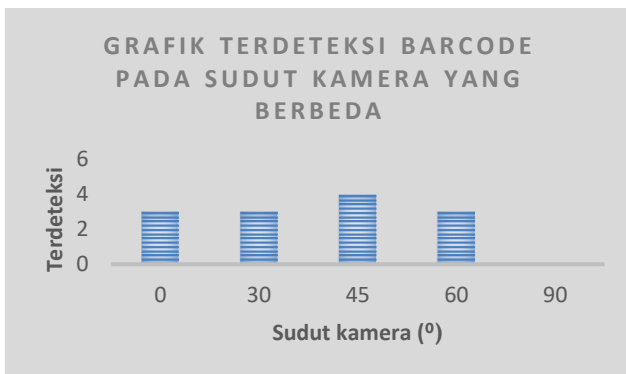
4.1.2. Pengujian Berdasarkan Sudut Kamera Terhadap Objek

Sudut pengambilan kamera pada penelitian ini menggunakan 5 variabel sudut yang berbeda yaitu 0° , 30° , 45° , 60° , dan 90° terhadap objek. sudut pengambilan kamera yang berbeda ini ditunjukkan pada gambar 4.3. Dari hasil pengujian didapatkan bahwa system tidak mampu membaca barcode apabila sudut pengambilan kamera terhadap objek $>60^\circ$.

Dari gambar 4.5. didapati bahwa sistem ini paling baik untuk mendeteksi barcode dengan set sudut kamera sebesar 45° terhadap objek. Hal ini dikarenakan kamera lebih mampu mendeteksi barcode pada jarak yang lebih banyak sehingga area pendeteksian menjadi lebih luas.



Gambar 4. 4 Pengambilan sudut kamera terhadap objek yang berbeda



Gambar 4. 5 Grafik deteksi barcode pada sudut kamera yang berbeda

4.1.3. Pengujian Berdasarkan Ketinggian Kamera

Ketinggian kamera terhadap objek diukur dari permukaan tanah hingga ke lensa kamera. Jarak ketinggian kamera terhadap objek diukur mulai dari 20 cm diatas permukaan tanah hingga 50 cm diatas permukaan tanah, Hasil pengujian menunjukkan kamera tidak dapat membaca jenis barcode apabila jarak kamera terhadap objek lebih dari 30 cm.

Tabel 4. 1 Hasil Pengujian deteksi barcode dalam kondisi bergerak

Kecepatan	Terdeteksi Barcode
Lambat (0,013 m/s)	True
Menengah (0,026 m/s)	True
Cepat (0,039 m/s)	False

4.1.4. Pengujian Berdasarkan Kecepatan Pergerakan Objek

Kecepatan pergerakan objek dalam penelitian ini dipengaruhi oleh pergerakan motor conveyor. Objek diletakkan diatas sabuk conveyor dan akan bergerak mendekati kamera. Kecepatan conveyor dipengaruhi oleh pengaturan motor stepper. Motor stepper diatur dalam kecepatan konstan.

Pada penelitian ini, dikarenakan ada beberapa penyesuaian pengaturan kecepatan conveyor maksimal pada 0,039 m/s yang dikategorikan sebagai kecepatan cepat. Kecepatan lambat pada kecepatan 0,013 m/s. Dan kecepatan menengah pada 0,026 m/s. Dari setiap kategori kecepatan akan dilihat kemampuan kamera dalam menangkap dan mendeteksi barcode pada objek. Tabel 4.1. menunjukkan hasil pengujian deteksi objek pada kondisi bergerak

4.2. Analisa Data

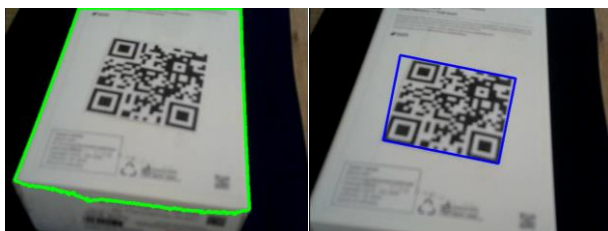
Dari hasil pengujian dengan beragam variable didapati bahwa setting posisi kamera terbaik berada pada sudut 45° terhadap objek. Ketinggian kamera diatur pada jarak 30 cm terhadap alas objek. Objek akan bergerak dari jarak 40 cm sampai mendekati kamera pada kecepatan 0,013 m/s.

Pada pengambilan data ini penulis menggunakan 15 sampel objek dengan warna yang berbeda. Tujuan dari penggunaan warna objek yang berbeda ini untuk melihat akurasi sistem apakah masih bisa mendeteksi objek apabila backgroundnya berbeda-beda karena dalam kondisi

sebenarnya pasti setiap bagasi mempunyai warna yang berbeda-beda. Contoh dari sampel yang berbeda ini dapat dilihat pada gambar 4.6. Hasil dari pembacaan yang tepat dapat dilihat pada gambar 4.7. sedangkan untuk pembacaan yang salah dapat dilihat pada gambar 4.8



Gambar 4.6. contoh sampel objek dengan warna yang berbeda



Gambar 4.7. Deteksi Bagasi dan Barcode yang benar



Gambar 4.8. Deteksi bagasi dan barcode yang salah

Tabel 4.2. Hasil pengujian pada 15 sampel

WARNA	BENAR	SALAH	AKURASI
Putih	8	0	100%
Biru muda	8	0	100%
Hitam	7	1	88%
Putih	8	0	100%
Kuning	8	0	100%
Biru laut	6	2	75%
Merah cerah	8	0	100%
Merah tua	6	2	75%
Biru muda	8	0	100%
Biru tua	7	1	88%
Hijau gelap	7	1	88%
Hijau cerah	8	0	100%
Coklat glossy	1	7	13%
Kuning	6	2	75%
Oranye	7	1	88%
AKURASI TOTAL			86%






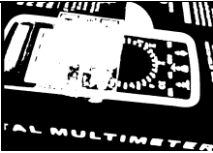



Pada penelitian ini, orientasi objek akan diubah-ubah sebanyak 8 arah sesuai arah mata angin. Orientasi objek yang diubah-ubah ini juga untuk melihat kemampuan sistem dalam mendeteksi objek. Hasil dari pendeteksian sampel yang berbeda ini membuat total ada 120 data yang masuk. Hasil dari penelitian ini bisa dilihat pada tabel 4.2. Hasil True menandakan objek berhasil dideteksi dan barcode berhasil diidentifikasi. Sedangkan, hasil false menandakan objek tidak dapat dideteksi dan tidak dapat diidentifikasi.

Dari tabel dapat dilihat bahwa total akurasi sistem sebesar 86%. Namun terdapat warna yang tingkat akurasinya dibawah 50% yaitu warna coklat glossy yang hanya mampu menangkap dengan benar 1 dari 8 orientasi. Dari hasil analisa didapatkan bahwa warna yang bersifat memantulkan cahaya membuat sistem tidak dapat membaca ataupun mendeteksi objek dengan akurat.

Warna-warna yang bersifat cerah seperti putih, hijau cerah, biru muda lebih mampu menghasilkan tingkat akurasi lebih tinggi dibandingkan warna gelap. Sistem mampu mendeteksi objek lebih baik dalam background cerah dikarenakan tingkat kontrasannya lebih tinggi sehingga pendeteksian lebih bisa mendeteksi bagian yang sesuai.

Analisa dari hasil pengujian tiap objek mulai dari gambar asli, gambar setelah grayscale, dan gambar saat proses threshold ke bentuk biner akan ditunjukkan melalui tabel 4.3. , tabel 4.4., dan tabel 4.5. Tiap tabel akan menampilkan salah satu orientasi dari 8 orientasi tiap objek yang diuji dalam penelitian ini. Pemrosesan gambar asli menuju ke bentuk threshold biner perlu dilakukan konversi warna ke dalam bentuk grayscale agar lebih memudahkan pemrosesan citra. Semua objek yang bergerak di konveyor akan terus mendekati kamera dan menampilkan perubahan terhadap tingkat pencahayaan serta kekontrasan yang ditangkap oleh kamera.












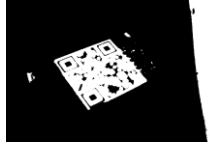
Tabel 4.3. Analisa gambar tidak sempurna

No	Asli	Grayscale	Threshold Biner
1			
2			
3			

Dari Tabel 4.3. menunjukkan beberapa contoh gambar yang tidak sempurna. Terlihat bahwa saat gambar dibuat dalam grayscale dan dilanjutkan threshold biner terdapat bentuk objek yang tidak sempurna sehingga menyebabkan sistem tidak dapat mendeteksi secara sempurna objek yang tertangkap. Tidak sempurnanya bentuk objek ini salah satunya disebabkan oleh warna dan bahan objek yang bersifat memantulkan cahaya.

Ketidaktepatan dalam membaca kontur objek ini mengakibatkan kontur yang dideteksi dan akan digambar oleh sistem mengalami kecacatan dalam pemrosesan. Alhasil bentuk barcode dan juga objek yang digambar berbentuk abstrak dan sulit untuk diidentifikasi lebih lanjut.. Mode RETR_TREE dan juga metode CHAIN_APPROX_SIMPLE yang digunakan dalam pencarian kontur di penelitian kali ini bisa diganti dengan mode atau metode yang lain untuk mengantisipasi kegagalan proses deteksi objek.

Tabel 4.4. Analisa hasil gambar >50% sempurna

No	Asli	Grayscale	Threshold Biner
1			
2			
3			
4			



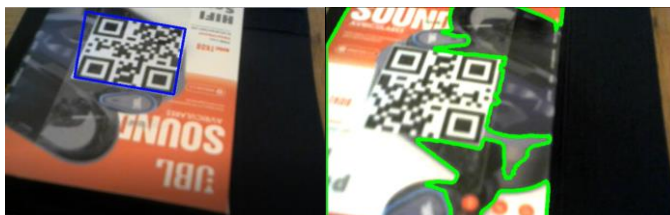
Apabila kita mencocokkan data pada tabel 4.2. yang menampilkan tingkat error sebesar 14% dan juga hasil pemrosesan gambar pada tabel 4.3. dan 4.4. dapat diketahui beberapa poin permasalahan yang menyebabkan kegagalan deteksi dan identifikasi pada penelitian ini.

Warna dan bahan objek yang memantulkan cahaya ini menyebabkan kamera mengalami ketidaksempurnaan dalam menangkap keseluruhan citra objek tersebut. Apabila objek tidak dapat terdeteksi dan teridentifikasi maka akan muncul notifikasi pada sistem seperti yang diilustrasikan pada gambar 4.9.

Tabel 4.4. menampilkan beberapa objek yang kurang sempurna dalam pemrosesan konversi ke bentuk grayscale dan biner namun masih dapat dideteksi serta diidentifikasi. Bisa terlihat bahwa objek terdeteksi dan teridentifikasi namun masih menyisakan beberapa bagian yang tidak dapat digambar oleh contour. Hasil dari pemrosesan gambar yang kurang sempurna ini ditunjukkan pada gambar 4.10. yang menampilkan pula hasil output dari identifikasi sistem.

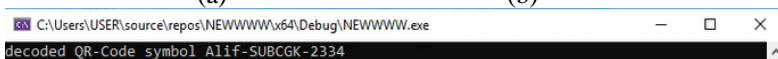
```
WARNING: .\zbar\decoder\pdf417.c:89: <unknown>: Assertion "g[0] >= 0 && g[1] >= 0 && g[2] >= 0" failed.
dir=0 sig=8704 k=3 g0=2f8 g1=ffffffff g2=000 buf[0000]=
WARNING: .\zbar\decoder\pdf417.c:89: <unknown>: Assertion "g[0] >= 0 && g[1] >= 0 && g[2] >= 0" failed.
dir=0 sig=cad8 k=0 g0=dd9 g1=ffffffff g2=cef buf[0000]=
```

Gambar 4.9. notifikasi kegagalan deteksi sistem



(a)

(b)



(c)













Gambar 4.10. (a) deteksi barcode, (b) deteksi objek yang kurang sempurna, (c) hasil identifikasi isi informasi pada kondisi objek yang kurang sempurna

Pada 2 kasus pemrosesan objek yang kurang sempurna seperti ditampilkan pada tabel 4.3. dan tabel 4.4. didapatkan bahwa objek yang mengganggu pencahayaan dalam hal ini memantulkan cahaya kembali ke kamera akan mengakibatkan proses deteksi dan identifikasi yang tidak terlalu baik. Penggunaan objek dengan warna bersifat *doff* akan memudahkan sistem dalam melakukan prosesnya.

Objek yang secara sempurna dalam proses deteksi dan identifikasi dapat dilihat pada tabel 4.5. Objek yang berada di tiap nomer tersebut menampilkan hasil pemrosesan konversi ke dalam bentuk grayscale dan biner yang sempurna. Objek bagasi dapat dideteksi secara sempurna keseluruhan bentuknya. Begitu pula dengan barcode yang bentuknya tidak mengalami kecacatan sehingga sistem bisa mendeteksi dan mengidentifikasi isi informasi pada barcode tersebut.

Pada tabel 4.5. dapat dilihat bahwa gambar yang terproses sempurna akan menampilkan bentuk objek yang terblok putih saat dithreshold biner. Barcode yang akan diidentifikasi juga terlihat bentuknya tidak mengalami kecacatan dan masih menyerupai bentuk aslinya sehingga bisa memudahkan sistem dalam mengidentifikasi isi informasi yang dibawa oleh barcode tersebut

Tabel 4.5. Analisa proses gambar sempurna

No	Asli	Grayscale	Threshold Biner
1			
2			
3			
4			

Dari hasil yang ditunjukkan dapat dianalisa bahwa objek dan barcode yang kesempurnaan objeknya diatas 50% masih dapat untuk dideteksi dan diidentifikasi. Warna dari objek yang akan dideteksi dan diidentifikasi perlu diperhatikan pula agar sistem dapat memproses citra yang didapatkan dari kamera secara sempurna.

-----*Halaman ini sengaja dikosongkan*-----

BAB V

PENUTUP

5.1. Kesimpulan

Penelitian tugas akhir yang telah dilakukan ini memunculkan beberapa kesimpulan yang dapat diambil oleh penulis, antara lain :

- Variable yang paling mempengaruhi pembacaan kamera terhadap objek adalah jarak kamera terhadap objek, sudut pengambilan kamera, kecepatan objek, dan warna background objek
- Sudut pengambilan kamera untuk mendapatkan hasil yang optimal berada pada sudut 45° terhadap objek
- Jarak antara objek dengan kamera paling optimal saat berada dalam range jarak 20-30 cm dari kamera
- Kecepatan objek mempengaruhi proses pendeteksian barcode pada bagasi
- Warna background objek paling baik apabila berwarna cerah namun tidak memantulkan cahaya
- Tingkat akurasi pendeteksian berada pada angka 86%
- Tingkat akurasi tanpa pendeteksian bagasi juga menghasilkan tingkat akurasi pendeteksian barcode yang sama sehingga pendeteksian cukup menggunakan *barcode detection*
- Hasil penelitian belum 100% akurat dikarenakan belum diaplikasikan langsung pada unit *Baggage Handling System*

5.2. Saran

Penelitian tugas akhir yang telah dilakukan penulis memiliki beberapa bagian yang masih dapat disempurnakan. Oleh sebab itu, penulis memiliki beberapa saran untuk kelanjutan penelitian ini antara lain

- Penggunaan kamera yang mampu auto-focus sehingga dapat menangkap serta mendeteksi objek pada jarak yang lebih jauh
- Pengaplikasian sistem langsung pada unit BHS Bandara agar hasil yang didapatkan lebih akurat
- Objek yang akan dideteksi memiliki variasi warna dan bentuk sehingga dapat dilihat keandalan sistem

-----*Halaman ini sengaja dikosongkan*-----

DAFTAR PUSTAKA

- [1] K. K. K and K. V. M. Prashanth, "Techniques for Detecting and Tracking of Baggages in Airports," in 2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT), 2017, pp. 333–338.
- [2] V. T. Le, J. Zhang, M. Johnstone, S. Nahavandi, and D. Creighton, "A generalised data analysis approach for *Baggage Handling Systems* simulation," in 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2012, pp. 1681–1687.
- [3] Y. Ebrahim, W. Abdelsalam, M. Ahmed, and Siu-Cheung Chau, "Proposing a hybrid tag-camera-based identification and navigation aid for the visually impaired," in Second IEEE Consumer Communications and Networking Conference, 2005. CCNC. 2005, 2005, pp. 172–177.
- [4] A. Sobti, C. Arora, and M. Balakrishnan, "Object Detection in Real-Time Systems: Going Beyond Precision," in 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, 2018, pp. 1020–1028.
- [5] R. Zocca, F. D'Ercoli, and D. Vincenzi, "A statistical tool for Barcode misread analysis," in 2007 IEEE Workshop on Automatic Identification Advanced Technologies, 2007, pp. 156–160.
- [6] K. Qian, Y. Yu, D. Wang, X. Yu, and Z. Zhao, "Design for two-dimensional barcode dynamic recognition system in the environment of large-scale logistics," in 2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2015, pp. 878–882.
- [7] B. Basavaprasad, M. Ravi, "A study on the importance of image processing and its applications", in 2014 National Conference on Recent Innovations in Engineering and Technology, 2014, pp.155-160
- [8] "About" [Daring]. Tersedia pada : <https://opencv.org/about/> [Diakses pada 8 Juni 2020]
- [9] "Logitech C270 HD Webcam." [Daring]. Tersedia pada : <https://www.logitech.com/en-us/product/hd-webcam-c270> [Diakses pada 8 Juni 2020]
- [10] "What is Arduino." [Daring]. Tersedia pada : <https://www.arduino.cc/en/guide/introduction> [Diakses pada 22 Juni 2020]
- [11] "Arduino Uno REV3" [Daring]. Tersedia pada : <https://store.arduino.cc/usa/arduino-uno-rev3> [Diakses pada 22 Juni 2020]

- [12] X. Deng, J.P.M.G Linnartz, X.Long, and G. Zhou, “Reading Analysis for Barcode Scanner With Interference from LED-Based Lighting,” in 2019 IEEE Access Volume 7, 2019, pp.96787-96798.
- [13] H.Baskoro, H. Prabowo, A. Trisetyarso, Meyliana, A.N. Hidayanto, “Design Considerations of RFID Based *Baggage Handling System*, A Literature Review”, in 2017 International Conference on Information Management and Technology (ICIMTech), 2017, pp. 210-214.
- [14] A.Kahler, G.Bradsky. “Learning OpenCV : Computer Vision in C++ with the OpenCV Library”, 2018, O’Reilly Media

LAMPIRAN

Lampiran 1 (Source code program)

```
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/opencv.hpp>
#include <opencv2/zbar.h>
#include <iostream>

using namespace cv;
using namespace std;
using namespace zbar;

int main()
{
    VideoCapture cap;
    cap.open(0);
    Mat grey;
    Mat detek;
    Mat biner;
    namedWindow("Asli", WINDOW_AUTOSIZE);
    //namedWindow("grayscale", WINDOW_AUTOSIZE);
    namedWindow("Deteksi", WINDOW_AUTOSIZE);
    if (!cap.isOpened()) // exit program if unsuccessful
    {
        cout << "Cannot open the video cam" << endl;
        return -1;
    }
    int width_frame = cap.get(CAP_PROP_FRAME_WIDTH);
    int height_frame = cap.get(CAP_PROP_FRAME_HEIGHT);
    VideoWriter video("bag15.avi", zbar_fourcc('M', 'J', 'P', 'G'), 10, Size(width_frame, height_frame), true);
```

```

ImageScanner scanner;
scanner.set_config(ZBAR_NONE, ZBAR_CFG_ENABLE, 1);
double dWidth = cap.get(CAP_PROP_FRAME_WIDTH);
double dHeight = cap.get(CAP_PROP_FRAME_HEIGHT);
VideoWriter video2("bar15.avi", zbar_fourcc('M', 'J',
'P', 'G'), 10, Size(width_frame, height_frame), true);
cout << "Frame size : " << dWidth << " x " << dHeight
<< endl;

    while (1)
    {
        cap >> detek;
        if (detek.empty()) break;
        cvtColor(detek, grey, COLOR_BGR2GRAY);
        threshold(grey,      biner,      110,      255,
THRESH_BINARY);
        int edgeThresh = 1;

        vector<vector<Point>>contours;
        Mat finderCount = biner.clone();
        findContours(finderCount, contours, RETR_TREE,
CHAIN_APPROX_SIMPLE);
Mat DetectContour(detek.size(), CV_8UC3, Scalar(0, 0,
0));

        int areamin = 0;
        int dtcup = 0;
        for (int a = 0; a < contours.size(); a++)
        {
            if (contourArea(contours[a]) > areamin)
            {
                areamin = contourArea(contours[a]);
                dtcup = a;
            }
        }
    }

```

```

drawContours(detek, contours, dtcup, Scalar(0, 255, 0),
5, 8);

    imshow("Deteksi", detek);
    video.write(detek);

    Mat frame;
    bool bSuccess = cap.read(frame);
    if (!bSuccess)
    {
        cout << "Cannot read a frame from video
stream" << endl;
        break;
    }

    Mat gray;
    cvtColor(frame, gray, COLOR_BGR2GRAY);
    int width = frame.cols;
    int height = frame.rows;
    uchar* raw = (uchar*)gray.data;

    Image image(width, height, "Y800", raw, width *
height);

    int n = scanner.scan(image);
for (Image::SymbolIterator symbol =
image.symbol_begin();
    symbol != image.symbol_end();
    ++symbol) {
    vector<Point> vp;

    cout << "decoded " << symbol->get_type_name()
<< " symbol " << symbol->get_data() << endl;
    int n = symbol->get_location_size();
    for (int i = 0; i < n; i++) {
        vp.push_back(Point(symbol->get_location_x(i),
symbol->get_location_y(i)));
    }
}

```

```

RotatedRect r = minAreaRect(vp);
    Point2f pts[4];
    r.points(pts);
    for (int i = 0; i < 4; i++) {
        line(frame, pts[i], pts[(i + 1) %
4], Scalar(255, 0, 0), 3);
    }
}

    imshow("Asli", frame);
    video2.write(frame);

    if (waitKey(30) == 27)
    {
        cout << "esc key is pressed by user" <<
endl;
        break;
    }
}
return 0;
}

```

Lampiran 2 (source code program kecepatan motor)

```
#include <AccelStepper.h>
const int dirPin = 12;
const int stepPin = 11;

#define motorInterfaceType 1

AccelStepper myStepper(motorInterfaceType, stepPin, dirPin);

void setup() {
  myStepper.setMaxSpeed(1000);
  myStepper.setSpeed(100); //set kecepatan
}

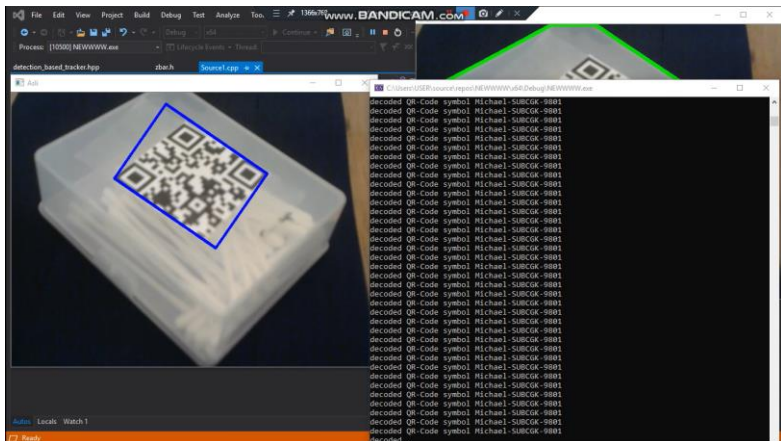
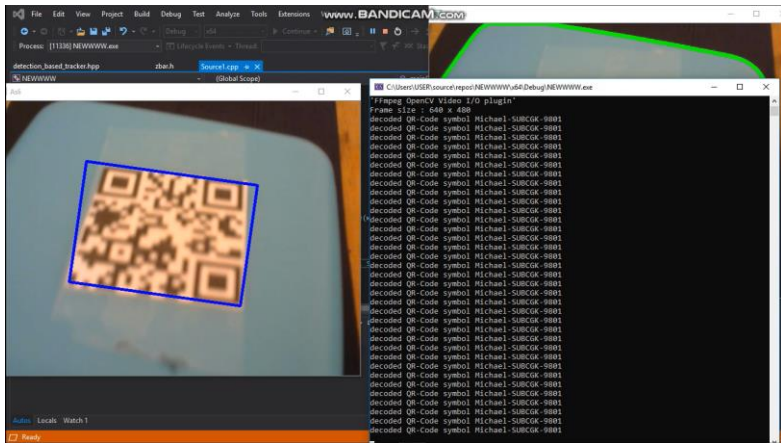
void loop() {
  myStepper.runSpeed();
}
```

Lampiran 3 (15 Sampel yang digunakan)



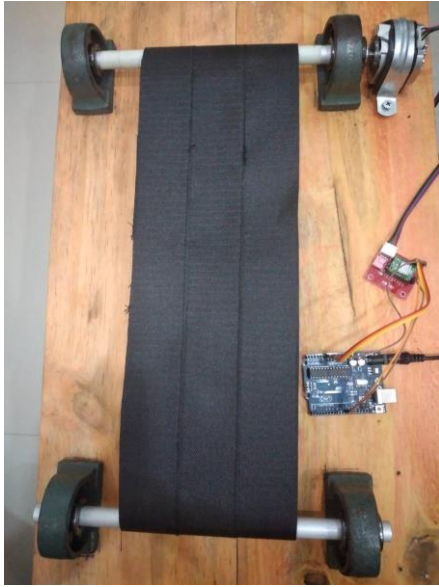


Lampiran 4 (tampilan interface saat pengujian)



Lampiran 5 (area pengujian)





BIODATA PENULIS



Penulis bernama lengkap Alif Fathsal Muttaqin yang lahir pada tanggal 24 Januari 1998 di Kota Madiun. Penulis memulai perkuliahan di Departemen Teknik Elektro, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember semenjak tahun 2016. Selama perkuliahan penulis aktif dalam kegiatan kemahasiswaan khususnya dalam bidang manajerial. Penulis aktif dalam organisasi Himpunan Mahasiswa Teknik Elektro (HIMATEKTRO ITS). Selain itu penulis juga aktif sebagai asisten laboratorium mikroelektronika dan system tertanam. Penulis dapat dihubungi melalui email : alifathsal@gmail.com