

PENYALA MESIN OTOMATIS PADA BOAT BERBASIS BARCODE

Automatic Start Engine on Boat based on Barcode

Dinda Dewi Kumalasari ¹, Suci Aulia, S.T., M.T ², Syahban Rangkuti, S.T., M.T ³

^{1,2,3}Prodi D3 Teknologi Telekomunikasi, Universitas Telkom

¹dindadewikumalasari@gmail.com, ²sucia@tass.telkomuniversity.ac.id, ³syahkti@yahoo.com

Abstrak

Alat transportasi pada zaman sekarang sudah menjadi kebutuhan pokok bagi manusia. *Boat* adalah salah satunya, yaitu alat transportasi air yang masih diminati terkhususnya oleh para nelayan di pesisir pantai. Para nelayan menempatkan *boat* mereka dipinggiran pantai. Karena banyaknya *boat* yang ada dan untuk menyalakan mesin *boat* masih manual, maka keamanan pada *boat* itu sendiri perlu dipertimbangkan.

Berdasarkan permasalahan diatas maka perlu dibuat sistem *barcode* sebagai penyala mesin otomatis pada *boat* sebagai solusi. Pada Proyek Akhir ini akan diimplementasikan suatu sistem penyala mesin otomatis berbasis *image processing*. Menuju Era Revolusi Industri 4.0, *image processing* digunakan untuk kemajuan teknologi atau *visual computer*. Pada perancangan sistem keamanan ini penulis menggunakan *barcode* sebagai kode *password* yang berfungsi untuk mengoperasikan *boat* yang sudah memenuhi aspek – aspek keamanan.

Dengan adanya sistem pengecekan terlebih dahulu pada aspek keamanan menggunakan *barcode*, akan dapat meningkatkan sistem keamanan pada *boat*. Serta memiliki performansi yang baik, dengan tingkat akurasi 85%, waktu proses yang dibutuhkan rata-rata selama 38.31 detik dengan persentase 83.3%.

Kata Kunci: *Barcode, Image Processing, Boat.*

Abstract

Transportation has become a basic necessity for humans. Boat is one of them, namely water transportation that is still in demand especially by fishermen on the coast. The fishermen put their boat on the shore. Because there are many boats and the start engines are still manual, then the safety of the boat itself needs to be considered.

Based on the above problems, it is necessary to make a barcode system as an automatic start engine on a boat as a solution. In this Final Project, an image processing-based automatic start engine system will be implemented. Towards the Age of Industrial Revolution 4.0, image processing is used for advancing technology or visual computers. In this designed security system the author uses a barcode as a password code to operate a boat that has met security aspects.

With the existence of first checking system on the security aspects for using a barcode, it will improve the security system on the boat. As well as having a good performance, with the rate of accuration is 85%, the required processing time is 38.31 seconds with the percentration is 83.3%.

Keyword: *Barcode, Image Processing, Boat.*

1. Pendahuluan

Teknik *barcode* telah banyak digunakan untuk mengidentifikasi produk, barang atau pengiriman dalam kehidupan kita sehari – hari. Secara tradisional, *barcode scanners* untuk membaca kode gambar yang dicetak. Dalam beberapa tahun terakhir, kamera digital menjadi lebih murah dan banyak tersedia pada perangkat pribadi seperti *smartphone*. Akan lebih mudah jika gambar *barcode* dapat dikodekan dengan menggunakan kamera digital yang ada pada *smartphone*. Namun, pada *image processing*, hal tersebut merupakan tantangan untuk mengenali *barcode* dengan akurasi tinggi dalam kondisi yang sulit[9].

Dimana – mana kamera tertanam pada ponsel yang memungkinkan pengguna untuk menangkap gambar *barcode* sebagai metode input dari pada mengetikan nomor *barcode* secara manual, yang mana untuk menghilangkan data masukan secara manual dari rawan kelambatan dan data yang salah. *Barcode reader* telah dikembangkan di berbagai bidang seperti bank daerah, produk yang direkomendasi, logistik, dan informasi halal pada umat muslim[4]. Dalam beberapa tahun terakhir, dengan perkembangan teknologi Internet, berbagai simbol *barcode* telah dikembangkan dengan cepat. Terutama *barcode* 1D yang banyak digunakan sebagai pembelian langsung di supermarket dan toko serba ada[10].

Di Jepang, sebagian besar ponsel dengan perangkat kamera yang mendukung pengenalan simbol, seperti EAN *barcode* dan QR code (*barcode* 2D), dan simbol kode ini digunakan untuk mengakses layanan jaringan

yang disediakan telepon secara mudah dengan membaca URL atau karakter dari alamat tersebut[5]. Dapat dilihat bahwa manfaat *barcode* dengan *image processing* sangatlah banyak, selain kepraktisannya, *barcode* juga dinilai keamanannya yang terjamin karena mempunyai tingkat akurasi tinggi melalui *image processing* dengan waktu proses yang cepat.

2. Dasar Teori

2.1. Barcode

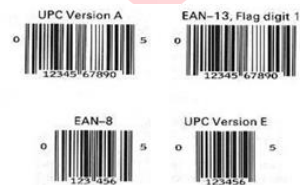
Barcode adalah gambar persegi atau persegi panjang yang terdiri dari serangkaian garis hitam paralel dan spasi putih dengan berbagai lebar yang dapat dibaca oleh pemindai. *Barcode* diterapkan pada sebuah produk sebagai alat identifikasi yang cepat. *Barcode* digunakan di toko eceran sebagai bagian dari proses pembelian, digudang untuk melacak inventaris, dan pada faktur untuk membantu dalam akuntansi, dan masih banyak kegunaan lainnya.



Gambar 2. 1 Barcode

Konsep *barcode* dikembangkan oleh Norman Joseph Woodland dan Bernard Silver, yang menggambar serangkaian garis dipasir untuk merepresentasikan kode Morse. Paten diberikan pada tahun 1966 dan NCR menjadi perusahaan pertama yang mengembangkan pemindai komersial untuk membaca simbologi *barcode*. Beberapa jenis *barcode* adalah sebagai berikut:

2.1.1. UPC/EAN



Gambar 2. 2 UPC/EAN

Format *barcode* paling populer adalah Format UPC (*Universal Product Code*) yang kita temukan di semua produk supermarket. Tersedia sejak awal tahun 1970-an, format ini dikenal diseluruh dunia dan diakui secara universal. Kode UPC normal berisi 12 digit angka. Digit pertama mengidentifikasi jenis produk kode apa yang digunakan (eceran, farmasi, dll). Kelima digit berikutnya mengidentifikasi produk tertentu yang diproduksi oleh pabrik. Digit terakhir adalah digit cek yang digunakan untuk mengetahui apakah pemindai kode *barcode* membaca sebelas digit pertama dengan benar. Produsen produk eceran harus mengajukan permohonan *barcode* UPC untuk produk tersebut dengan menghubungi *Uniform Code Council* (UCC).

2.1.2. Code 39



Gambar 2. 3 Code 39

Kode 39 adalah kode dua tingkat yang dirancang untuk menyediakan huruf dan angka. Versi standar mengkodekan huruf – huruf besar A – Z, angka dan beberapa tanda baca. Karakter asterisk (*) selalu digunakan sebagai karakter awal dan penghentian. *Extended Code 39* mengkodekan 128 karakter ASCII. Kode 39 telah menjadi standar untuk pemerintahan, pabrik, *barcode* industri, pendidikan, dan aplikasi bisnis.

Disebut kode 39 karena setiap karakter terdiri dari sembilan elemen, lima bar, dan empat spasi. Tiga dari sembilan elemen tersebut yaitu lebar, sedangkan enam sisanya yaitu sempit. *Extended Code 39* menggunakan pasangan karakter tertentu untuk mewakili karakter yang biasanya tidak ada dalam kode 39. Karakter yang ditambahkan ini mengambil ruang yang biasanya menghasilkan dua karakter, sehingga kode yang dihasilkan lebih panjang daripada kode 39 biasa untuk sejumlah karakter tertentu.

2.1.3. Code 128



Gambar 2. 4 Code 128

Kode 128 adalah kode terbaik untuk digunakan ketika semua 128 karakter ASCII diperlukan. Kode 128 ini adalah kode empat tingkat, yang berarti bar dan spasi dapat memiliki empat lebar yang berbeda. Sebenarnya ada tiga versi dari kode 128. Versi A mengkodekan semua karakter alfanumerik huruf besar ditambah semua karakter kontrol ASCII. Versi B mengkodekan semua karakter alfanumerik huruf besar dan kecil. Versi C hanya mengkodekan angka. Dimungkinkan untuk beralih di antara set karakter dalam kode dengan menggunakan karakter shift. Keuntungan kode 128 adalah dapat menyandikan semua karakter ASCII dalam panjang kode sesingkat mungkin. Kerugiannya adalah, karena memiliki empat lebar bar dan ruang yang berbeda, bukan dua, lebih banyak tuntutan ditempatkan pada teknologi pencetakan dan pengukuran.

2.1.4. PDF 417



Gambar 2. 1 PDF 417

Kode PDF 417 dan Data Matrix adalah simbol dua dimensi yang paling umum digunakan saat ini. PDF 417 adalah simbol bertumpuk dan ditemukan oleh Ynjiun Wang tahun 1991 di Symbol Technologies. PDF adalah singkatan dari Portable Data File, dan simbol terdiri dari 17 modul yang masing – masing berisi 4 bar dan spasi (dengan demikian angka “417”). Struktur kode memungkinkan untuk kepadatan informasi antara 100 dan 340 karakter. Kode berada di domain publik. General Motors mengumumkan pada bulan Februari 2000 bahwa semua pemasoknya harus mengkonversi ke standar PDF 417 untuk semua bagian dan pengiriman.

2.1.5. Data Matrix



Gambar 2. 2 Data Matrix

Data Matrix dari CiMatrix adalah kode matriks 2D yang dirancang untuk mengemas banyak informasi dalam ruang yang sangat kecil. Simbol Data Matrix dapat menyimpan antara 1 hingga 500 karakter. Simbol ini juga terukur antara persegi 1-mil ke persegi 14 inci. Karena informasi dikodekan oleh posisi dot absolut dan posisi dot relatif, itu tidak rentan terhadap cacat pencetakan seperti barcode tradisional. Skema pengkodean memiliki tingkat redundansi tinggi dengan data “tersebar” di seluruh simbol. Menurut perusahaan, ini memungkinkan simbol untuk dibaca dengan benar bahkan jika sebagian dari itu hilang.

2.1.6. 3-DI



Gambar 2. 3 barcode 3-DI

3 – DI dikembangkan oleh Lynn Ltd. dan merupakan kode kepemilikan. Barcode ini paling cocok untuk tanda identifikasi pada permukaan logam yang mengkilap dan melengkung seperti instrumen.

2.1.7. Array Tag



Gambar 2. 4 Array Tag

ArrayTag diciptakan oleh Dr. Warren D. ArrayTag dapat menyandikan ratusan karakter dan dapat dibaca pada jarak hingga 50 meter dan dioptimalkan untuk membaca pada jarak jauh atau dalam situasi pencahayaan variabel. Pengaplikasian prinsip dari kode ini adalah untuk melacak log.

2.1.8. Aztec Code



Gambar 2. 5 Aztec Code

Aztec Code ditemukan oleh Andy Longacre dari Welch Allyn Inc. pada tahun 1995 dan berada di area publik. Aztec Code dirancang untuk kemudahan pencetakan dan kemudahan decoding. Aztec Code terkecil mengkodekan 13 karakter angka atau 12 karakter alfabet, sedangkan simbol terbesar menyandikan 3832 numerik atau 3067 karakter alfabet[2] [7].

2.2 Pengolahan Citra Digital

2.2.1 Akuisi Citra

Akuisisi citra merupakan proses menangkap (*capture*) atau memindai (*scan*) suatu citra analog sehingga diperoleh citra digital. Beberapa faktor yang perlu diperhatikan dalam proses akuisisi citra antara lain adalah: jenis alat akuisisi, resolusi kamera, teknik pencahayaan, perbesaran atau zooming, jarak, dan sudut pengambilan citra.

2.2.2 Preprocessing

Pada tahap ini data citra masukan akan diubah menjadi data citra yang lebih sesuai untuk diproses. *Preprocessing* yang dilakukan meliputi beberapa tahap mulai dari *input* data citra, konversi RGB ke *grayscale*, konversi *grayscale* ke *BW (black and white)*, *resize* dan *cropping* sehingga menghasilkan citra sesuai dengan standar yang diharapkan dan dapat memudahkan proses selanjutnya.

1. Resize

Proses pengubahan *size* file citra yang awalnya *size* file citra tersebut besar akan di kompresi menjadi *size* file yang lebih kecil dari ukuran yang semula.

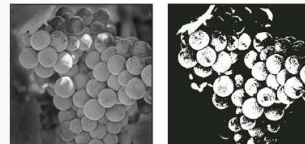
2. RGB – Grayscale



Gambar 2. 6 RGB-Grayscale

RGB adalah singkatan dari *Redn Green, Blue*. Warna piksel dalam gambar adalah kombinasi dari tiga warna Merah, Hijau, dan Biru (RGB). Nilai warna RGB adalah diwakili dalam tiga dimensi XYZ, diilustrasikan oleh atribut cahaya, kroma, dan rona. Kualitas gambar warna tergantung pada warna yang ditunjukkan oleh jumlah bit yang dapat didukung oleh perangkat digital. [3] *Gray* adalah warna gambar dimana hanya terdapat satu warna saja, yaitu warna yang mendominasi terhadap abu-abu (*gray*). *RGB to Grayscale* adalah sebuah proses pengubahan citra RGB ke citra *grayscale* (abu-abu).

3. Grayscale – BW



Gambar 2. 7 Grayscale-BW

Grayscale to BW adalah suatu proses pengubahan citra/gambar dari yang berwarna abu-abu atau dalam bahasa inggrisnya *gray* menjadi sebuah citra/gambar berwarna hitam dan putih (*black and white*).

4. Cropping



Gambar 2. 8 Cropping

Cropping adalah proses penghapusan bagian sudut dari suatu gambar untuk memotong/mengambil/mengeluarkan sebagian isi dari gambar guna memperoleh hasil yang diinginkan. Tujuan dari *cropping* adalah untuk mendapatkan foto dengan komposisi yang lebih enak untuk dilihat atau agar objek terlihat lebih dekat.

2.2.3 Image Processing

Image processing meningkatkan kualitas pemrosesan gambar. Teknik Digital *Image Processing* dapat diterapkan di berbagai bidang yang berbeda seperti analisis gambar diagnostik, perencanaan bedah, deteksi dan pencocokan objek, latar belakang pengurangan dalam video, mengukur volume jaringan, mencari benda di satelit gambar (jalan, hutan, dll.), sistem kontrol lalu lintas, menemukan objek dalam pengenalan wajah,

iris pengakuan, pencitraan pertanian, dan pencitraan medis. *Image processing* mengatasi masalah seperti hilangnya kualitas gambar, untuk meningkatkan citra yang terdegradasi[11].

2.3 Noise

Noise adalah sebuah istilah dalam dunia fotografi untuk menyebut titik – titik berwarna yang biasanya mengganggu hasil foto sehingga membuat foto menjadi nampak tidak halus. Adapun jenis – jenis noise sebagai berikut:

2.3.1 Gaussian Noise



Gambar 2. 9 Gaussian Noise

Gaussian noise juga disebut *normal noise*, karena disebabkan oleh sumber alami seperti getaran termal atom dan sifat khusus dari radiasi benda hangat. *Gaussian noise* umumnya mengganggu nilai abu – abu dalam gambar digital. Oleh sebab itu model *gaussian noise* pada dasarnya dirancang dan karakteristik dengan PDF (*Probability Density Function*) atau histogram yang dinormalisasi sehubungan dengan nilai abu – abu.[1]

2.3.2 Salt & Pepper



Gambar 2. 10 Salt and Pepper Noise

Seperti namanya *noise* jenis ini terlihat seperti *salt and pepper*. Namun gambar tidak sepenuhnya rusak oleh *noise* tersebut alih – alih beberapa nilai piksel diubah dalam gambar. *Noise* ini terlihat pada pengiriman data. Nilai piksel gambar digantikan dengan nilai piksel rusak baik maksimum atau nilai piksel minimum yaitu, 255 atau 0 masing – masing, jika jumlah bit adalah 8 untuk transmisi[1].

2.3.3 Poisson



Gambar 2. 11 Poisson Noise

Poisson noise bukan merupakan noise buatan. Poisson merupakan noise yang ditambahkan langsung pada citra tanpa kita menambahkan parameter apapun, sehingga efeknya pada citra pun tetap. Pada MatLab, jika matrik citra adalah *double precision*, maka nilai piksel inputnya dianggap sebagai mean dari distribusi *Poisson* dengan skala 10^{12} . Sebagai contoh jika piksel inputnya mempunyai nilai $5.5 \cdot 10^{-12}$, maka piksel keluaran akan dibangkitkan dari distribusi *Poisson* dengan mean 5.5 di kembalikan lagi ke skala 10^{12} . Jika matrik citra adalah *single precision*, maka factor skalanya menggunakan 10^6 . Jika matrik citra adalah uint8 atau uint16, maka nilai input piksel digunakan langsung tanpa faktor skala. Sebagai contoh jika masukan piksel uint8 mempunyai nilai 10, maka piksel keluaran akan dibangkitkan dari distribusi *Poisson* dengan mean 10 [1].

2.4 Segmentasi Karakter

Menyegmentasikan karakter pada *image* dengan mengelompokkan ke setiap karakter untuk mempermudah pencocokan di tahap selanjutnya[6][8]. Ketika melakukan segmentasi dilakukan juga penyeleksian area objek yang dipilih.

2.5 Character Recognition

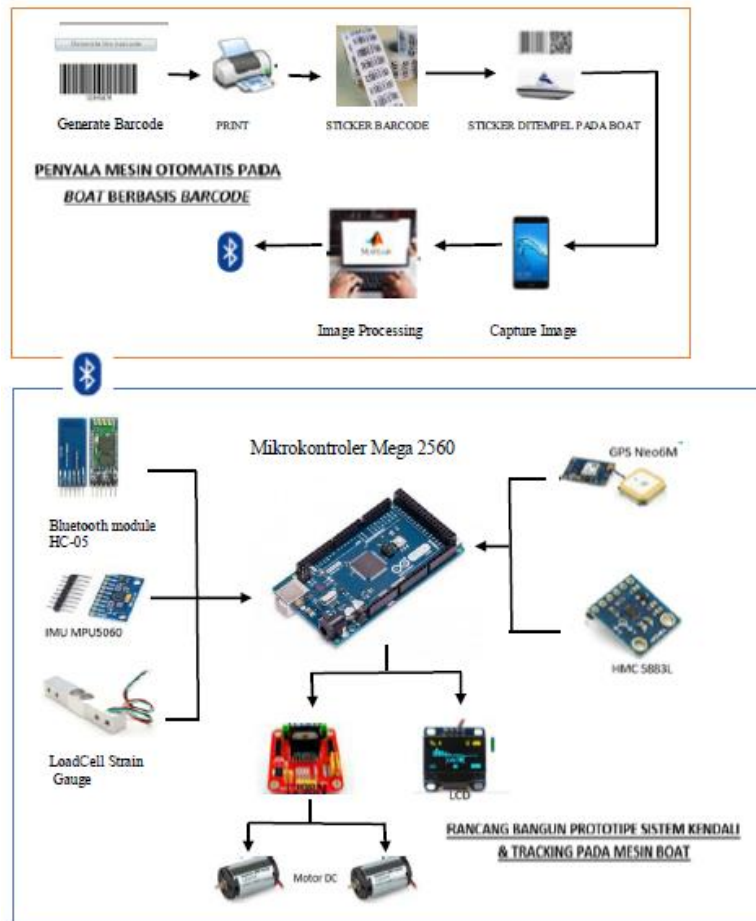
Gambar yang sudah disegmentasi kemudian disimpan dengan format tipe data *string* dalam sebuah variable. Selanjutnya pada tahap OCR (*Optical Character Recognition*) yang digunakan untuk mencocokkan

setiap karakter dengan database yang sudah disimpan, berupa file gambar alphabet dan angka yang sudah dijadikan bentuk bit yang dimana akan dikonfirmasi pada database. Selanjutnya jika sudah sesuai maka data akan ditampilkan[6].

3. Perancangan

Pada bab ini akan dijelaskan diagram sistem yang digunakan dengan hasil keluaran yaitu dapat menyalakan mesin pada sebuah *boat*. Mikrokontroler akan menyalakan *boat* sesuai dengan data *barcode reader* yang telah dikonversi pada MATLAB dan data – data yang terbaca pada sensor MU – 6050, sensor *Loadcell Strain gauge*, HMC 5883L, serta GPS Neo6m yang terpasang pada lantai kapal. Apabila *barcode* yang dibaca oleh MATLAB sudah sesuai maka MATLAB mengirimkan data serial ke mikrokontroler, kemudian mikrokontroler akan mengeksekusi data sesuai program yang dibuat.

Barcode akan difoto melalui akses dari kamera *smartphone* yang menggunakan operasi sistem *android*, menggunakan aplikasi *ipcam (Ip Webcam)* yang dapat diinstall melalui *play store*. Dan kemudian foto tersebut akan muncul pada axes GUI Matlab, pada Matlab akan dilakukan *preprocessing* kemudian melalui metode OCR (*Optical Character Recognition*) yang akan mendeteksi dan menyeleksi karakter dengan *bounding box*, lalu dari *bounding box* tersebut akan melalui proses *cropping*, dilanjutkan dengan proses *image to text* dimana pada proses tersebut karakter yang terdapat pada gambar yang telah dipotong dan terdeteksi akan diubah menjadi karakter teks dengan tipe data string. Kemudian akan dicocokkan dengan database dari ms.excel dengan format .csv. Jika *match* (cocok) akan mengirimkan data serial bit “1” dan pada GUI akan muncul keterangan ‘*boat on*’ sehingga *boat* akan menyala, jika tidak *unmatch* (tidak cocok) maka akan mengirimkan data serial bit “0” dan pada GUI akan muncul keterangan ‘*Off*’ sehingga *boat* tidak akan menyala. Adapun model sistem perancangan yang telah dibuat, dapat dilihat pada Gambar 3.1 dibawah ini.



Gambar 3. 1 Blok Diagram Ilustrasi Barcode Reader

3.2 Tahapan Perancangan

Adapun tahap proses perancangan yang dilakukan sebagai berikut :

3.2.1 Akuisisi Citra

Akuisisi citra merupakan langkah awal dari proses identifikasi *barcode*. Untuk memperoleh citra label *barcode* dilakukan pengambilan gambar menggunakan kamera *smartphone* dengan pengaruh jarak dan

pengaruh cahaya saat pagi atau malam hari. Dan hasil dari citra yang diambil berupa citra RGB. Berikut daftar waktu untuk pengambilan citra *barcode*:

1. Pukul 07.00 WIB
2. Pukul 12.00 WIB
3. Pukul 17.00 WIB

3.2.2 Preprocessing

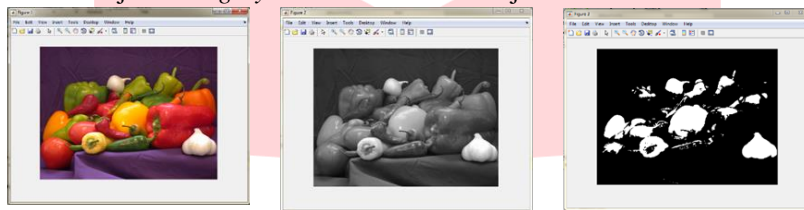
Pada tahap ini data citra masukan akan diubah menjadi data citra yang lebih sesuai untuk diproses. *Preprocessing* yang dilakukan meliputi beberapa tahap mulai dari masukan data citra, konversi RGB ke *grayscale*, konversi *grayscale* ke *BW* (*black and white*), *resize* dan *cropping* sehingga menghasilkan citra sesuai dengan standar yang diharapkan dan dapat memudahkan proses selanjutnya.

1. *Resize*

Proses pengubahan *size* file citra yang awalnya *size* file citra tersebut besar akan di kompresi menjadi *size* file yang lebih kecil dari ukuran yang semula.

2. Pengubahan Warna

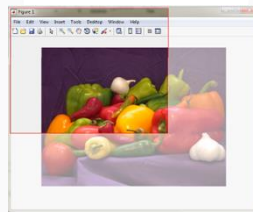
Pengubahan warna adalah mengubah warna citra yang awalnya citra RGB diubah menjadi citra *grayscale*, setelah menjadi citra *grayscale* akan diubah menjadi citra BW.



Gambar 3. 1 RGB-Grayscale-BW

3. *Cropping*

Cropping adalah proses penghapusan bagian sudut dari suatu gambar untuk memotong/mengambil/mengeluarkan sebagian isi dari gambar guna memperoleh hasil yang diinginkan. Tujuan dari *cropping* adalah untuk mendapatkan foto dengan komposisi yang lebih enak untuk dilihat atau agar objek terlihat lebih dekat.



Gambar 3. 2 *Cropping*

3.2.3 Image Processing

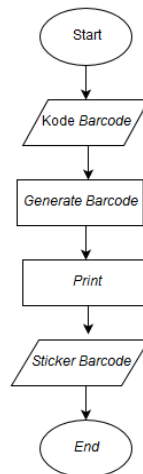
Image processing meningkatkan kualitas pemrosesan gambar. Teknik digital *image processing* dapat diterapkan diberbagai bidang yang berbeda seperti analisis gambar diagnostik, perencanaan bedah, deteksi dan pencocokan objek, latar belakang pengurangan dalam video, pelokalan tumor, mengukur volume jaringan, Cari benda di satelit gambar (jalan, hutan, dll.), sistem kontrol lalu lintas, menemukan objek dalam pengenalan wajah, iris pengakuan, pencitraan pertanian, dan pencitraan medis. *Image processing* mengatasi masalah seperti hilangnya kualitas gambar, untuk meningkatkan citra yang terdegradasi.

3.3 Blok Diagram Pengerjaan

Pada bab ini akan dijelaskan mengenai perancangan sistem yang akan dibuat adalah alur pembuatan *barcode*, pembacaan *barcode*, dan alur penyalaan *boat*. Adapun perancangan sistem yang dibuat sebagai berikut:

3.3.1 Diagram Alur Pembuatan *Barcode*

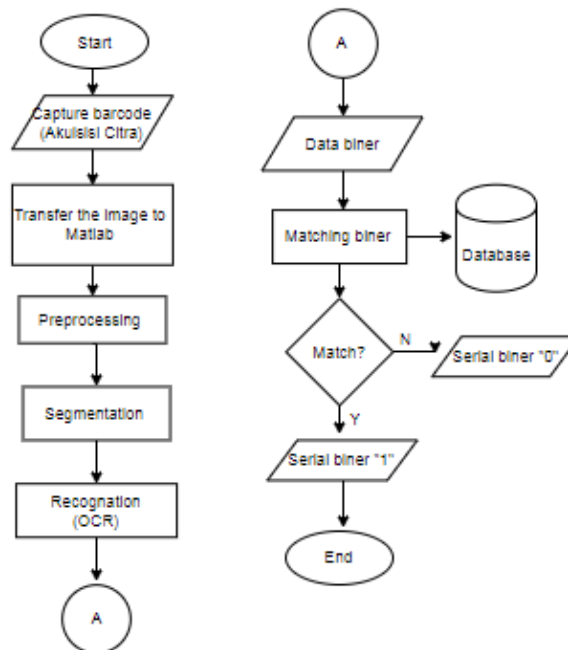
Diagram alir pada gambar 3.4 merupakan sebuah konsep pembuatan *barcode*, mulai dari masukan kode yang akan dibuat menjadi *barcode*, dimana *barcode* tersebut akan dicetak menjadi kertas tempel lalu selesai.



Gambar 3. 3 Diagram Alir Pembuatan Barcode

3.3.2 Diagram Alur Pembacaan Barcode

Diagram alir pada gambar 3.5 merupakan sebuah proses dari pembacaan *barcode*, mulai dari mengambil foto *barcode* menggunakan kamera *smartphone* dan masuk ke Matlab untuk diproses. Pada saat pemrosesan, terjadi proses akuisisi citra lalu masuk ke tahap *preprocessing* dengan melalui proses *resize* citra, pengubahan citra *RGB* ke *grayscale*, pengubahan citra *grayscale* ke citra *BW*, dan proses *cropping*. Selanjutnya masuk ke tahap *image processing* dengan menggunakan metode OCR (*Optical Character Recognition*), pada tahap ini terjadi proses segmentasi yang mana pada tahap tersebut penyeleksian lagi dengan area yang berukuran sama dengan angka dan huruf sehingga nantinya area yang dipilih adalah area yang berukuran sama angka dan huruf pada *barcode*, yang dimana diluar area itu akan diabaikan sehingga mempermudah untuk tahap selanjutnya. Pada tahap recognasi merupakan tahap pencocokan. Data yang dicocokkan dengan *bit database* huruf dan angka pada matlab akan diinisialisasikan dan diubah menjadi sebuah text. Dari text akan dicocokkan kembali dengan database csv, bila mana text yang terbaca cocok dengan data yang ada pada csv, maka akan mengirimkan srial biner “1”, jika tidak cocok akan mengirimkan serial biner “0”.

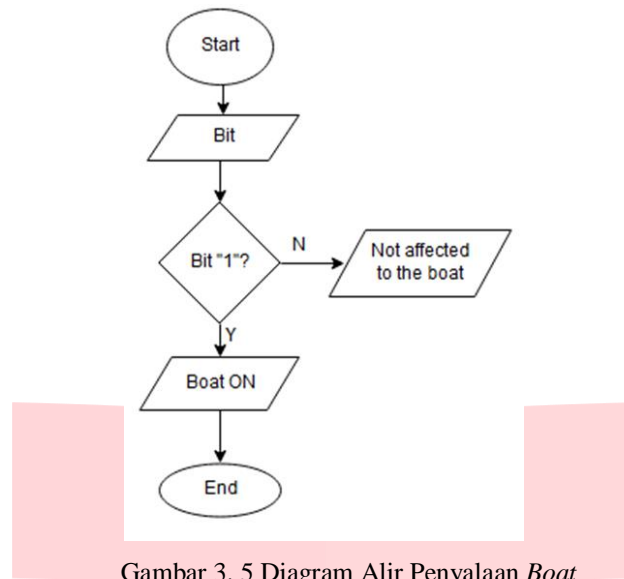


Gambar 3. 4 Diagram Alir pembacaan Barcode

3.3.3 Diagram Alur Penyalaan Boat

Diagram alir dibawah ini pada Gambar 3.6 menjelaskan tentang bagaimana alur dari sebuah proses pembacaan *barcode*, berawal dari masukan bit yang diperoleh dari hasil pencocokan dengan database, dan

bilamana bit yang diterima adalah bit “1” maka akan mengirimkan data serial “1” sehingga *boat* akan menyala, jika bit yang diterima adalah bit “0” maka akan mengirimkan data serial “0” sehingga *boat* tidak menyala.



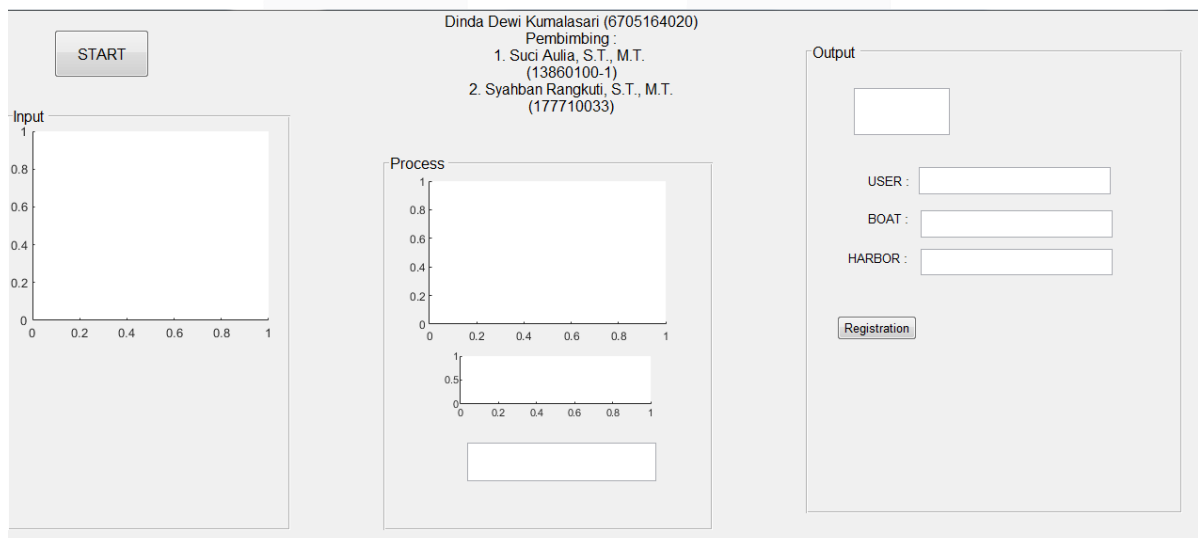
Gambar 3. 5 Diagram Alir Penyalaan *Boat*

3.4 Realisasi Sistem *Barcode Reader*

Pada bagian ini akan dijelaskan mengenai realisasi sistem Penyala Mesin Otomatis pada *Boat* Berbasis *Barcode*. *Barcode reader* yang telah terintegrasi dengan MATLAB akan diolah sesuai instruksi – instruksi pemrograman yang telah dibuat. Perancangan sistem ini menggunakan Arduino Mega 2560 sebagai sistem kontrol yang dapat merespon semua data input yang didapat dari sensor – sensor serta dari *barcode reader*.

3.4.1 Pembuatan Desain GUI pada Matlab

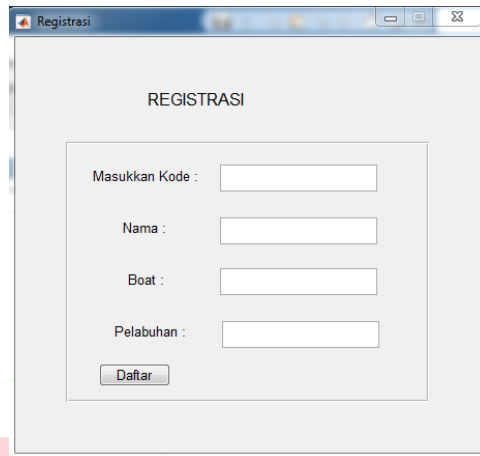
Dibawah ini merupakan beberapa tampilan GUI (*guide user interface*) yang telah dibuat dan digunakan pada sistem penyalat mesin otomatis pada *boat* berbasis *barcode* :



Gambar 3. 6 Desain Tampilan GUI

Gambar 3. 7 merupakan tampilan *guide* sistem Penyala Mesin Otomatis pada *Boat* Berbasis *Barcode*. Inputan berupa citra yang akan ditampilkan pada axes 1 dengan pilihan tombol *browse* atau *real time*. Pada kolom process selanjutnya citra *barcode* akan di *preprocessing* kemudian akan diproses *object detection*, lalu pada tombol *image to text* akan mengubah *image text* menjadi teks/karakter string. Setelah melalui proses pengolahan citra digital akan menghasilkan outputan berupa bit. Citra *barcode* yang sudah diproses akan dicocokkan dengan database yang sudah dibuat, database tersebut berisi kode-kode *barcode* yang sudah terdaftar untuk menyalakan *boat*. Jika pada tombol sinkronisasi muncul bit “1” maka *barcode* yang diproses sudah cocok

dengan database, jika pada tombol sinkronisasi muncul bit “0” maka barcode yang diproses tidak cocok dengan database.



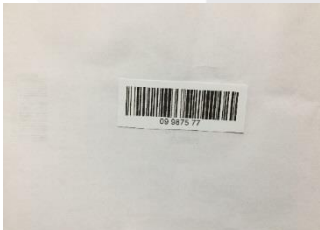
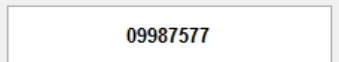



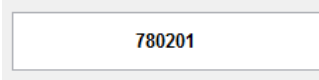

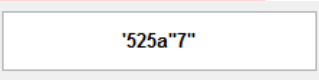
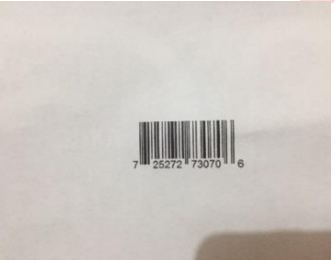
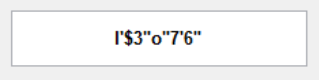
Gambar 3. 7 Desain GUI Registrasi

Gambar 3. 8 merupakan tampilan *guide* dari menu registrasi, yang dimana menu tersebut untuk mendaftarkan user baru atau kode *barcode* baru. Dengan cara masukan kode – kode yang akan didaftarkan dan masuk ke dalam database, dan masukan nama pengguna, tipe *boat*, dan pelabuhan dimana *boat* tersebut berlabuh.

3.5 Data Barcode

Tabel 4. 1 Pengujian pada bermacam jenis *barcode*

No	Citra Barcode	Hasil OCR
1.	(Barcode 39) 	
2.	(Barcode 128) 	

3.	 <p>(Barcode EAN 8)</p>	 <p>780201</p>
4.	 <p>(Barcode EAN 13)</p>	 <p>'525a"7"</p>
5.	 <p>(Barcode UPC)</p>	 <p>!\$3"o"7'6"</p>

Dapat dilihat pada Tabel 4.1 dari berbagai jenis barcode 1 dimensi, yang dapat terbaca oleh sistem menggunakan metode OCR (*Optical Character Recognition*) yaitu *barcode* dengan jenis *code 39* dan *code 128*.

3.6 Skenario pengujian

Dalam skenario ini dilakukan simulasi *barcode reader* menggunakan citra *barcode* yang akan dibaca sebagai kode *password* yang berfungsi untuk mengoperasikan *boat* yang sudah memenuhi aspek – aspek keamanan.

Pilih tombol *start* pada kolom *input* dan langsung terhubung dengan *Ipcam* untuk mengambil gambar secara langsung menggunakan kamera *smartphone*. Gambar yang berupa foto *barcode* akan muncul pada axes 1. Setelah mendapatkan citra yang akan diproses selanjutnya citra *barcode* tersebut akan melalui tiga tahapan proses yaitu *preprocessing*, *object detection*, dan *image to text*. Dimana proses tersebut digunakan untuk mengintegrasikan *barcode* yang dimasukkan.

Pada kolom *output* terdapat tombol registrasi, pilih tombol registrasi jika hendak mendaftarkan kode baru untuk menyalakan *boat* dan kode yang didaftarkan akan otomatis masuk kedalam database. Selanjutnya akan di sinkronisasi untuk memastikan apakah *barcode* yang dibaca sesuai dengan kode-kode yang terdapat pada database. Jika pada sinkronisasi muncul teks '*Boat On*' maka *barcode* yang dibaca sesuai dengan kode-kode pada database. Jika pada tombol sinkronisasi muncul teks '*Off*' maka *barcode* yang dibaca tidak sesuai dengan kode-kode pada database.

3.6.1 Parameter Pengujian

1. Akurasi

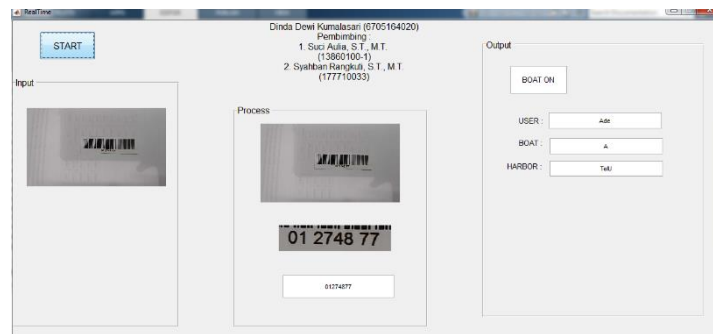
Pengujian tingkat akurasi deteksi angka pada *image* yang akan dikonverensikan menjadi angka dalam bentuk tipe data *string*.

2. Waktu proses
Pengujian lama waktu pemrosesan *barcode reader*.
3. Keandalan terhadap *noise*
Pengujian dimana objek yang terdapat *noise* dapat diproses dengan baik atau tidak.

4. Pengujian

4.1 Pengujian Sistem Simulasi *Barcode Reader*

Untuk mengetahui apakah sistem *barcode reader* dapat berfungsi sebagaimana mestinya, maka dilakukan pengujian dan analisis sesuai teori yang ada.

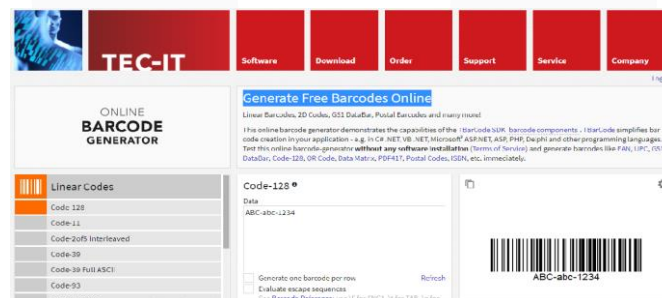


Gambar 4. 1 Hasil Simulasi *Barcode Reader*

Dari Gambar 4.1 terlihat hasil pengujian simulasi sistem *barcode reader* dengan masukan berupa *image barcode* yang diperoleh dari hasil foto menggunakan kamera *smartphone* dan diproses secara *realtime*. Pada hasil keluaran yang ditampilkan yaitu keterangan apakah *boat* berhasil menyala atau tidak, dan jika *image barcode* yang dibaca telah sesuai dengan database maka akan ditampilkan pula data *user* seperti nama *user*, nama *boat*, dan nama pelabuhan dimana *boat* tersebut dilabuhkan.

4.2.3 Generate Barcode

Generate barcode ini dilakukan secara online. Untuk generate barcode online ini dapat diakses pada <https://barcode.tec-it.com/en/Code128>. Jenis barcode yang disediakan sangat bermacam – macam dapat dipilih sesuai keinginan dan kebutuhan.



Gambar 4. 2 tampilan generate barcode

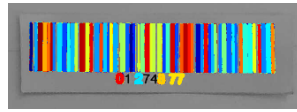
Cara penggunaannya cukup mudah, masukan kode yang diinginkan ke dalam kolom data, kemudian klik *Refresh* dan hasil generate barcode dapat dilihat pada kolom samping kolom data.

4.2.3 Akuisisi Citra

Citra barcode yang akan diambil menggunakan kamera *smartphone android* operating system, dengan kamera yang fokus dan jelas untuk mempermudah proses selanjutnya. Pengambilan citra barcode dengan jarak ketinggian 15cm dan sudut 90°. Kondisi cahaya yang diperlukan yaitu dengan kondisi cahaya yang cukup, pada pagi, siang, dan sore hari.

4.2.3 Segmentasi

Segmentasi citra merupakan proses yang ditunjukkan untuk mendapatkan objek-objek yang terkandung di dalam citra atau membagi citra ke dalam beberapa daerah dengan setiap objek atau daerah memiliki kemiripan atribut.



Gambar 4. 3 Hasil Segmentasi

Pada gambar 4.3 menunjukkan gambar hasil deteksi wilayah teks dengan segmentasi. Ini berfungsi dengan baik untuk teks karena konsistensi warna dan kontras teks yang tinggi mengarah ke profil intensitas yang stabil. Source Code sbb:

```
img = handles.data1;
I = rgb2gray(img);
% Detect segmentation regions.
[mserRegions, mserConnComp] = detectMSERFeatures(I, ...
'RegionAreaRange',[200 8000],'ThresholdDelta',4);
```

4.2.3 Thresholding



Gambar 4. 4 Penyeleksian dan *removeing non-text*

Memisahkan antara teks dan non-teks, yaitu menghapus daerah non-teks. Penghapusan daerah non-teks dapat digunakan untuk menyaring wilayah non-teks menggunakan *thresholding*/ambang batas sederhana.

4.2.3 Segmentasi Kedua

Memisahkan setiap karakter yang terdapat pada gambar. Dan mengidentifikasi satu persatu gambar yang sudah dipisahkan, dan melakukan pengecekan terhadap database.



Gambar 4. 5 Pemisahan setiap karakter

Source code sebagai berikut :

```
% Dapatkan kotak pembatas untuk semua wilayah.
bboxes = vertcat(mserStats.BoundingBox);

% Konversikan dari format kotak pembatas [x y lebar tinggi] ke [xmin ymin
xmax ymax] format untuk memudahkan.
xmin = bboxes(:,1);
ymin = bboxes(:,2);
xmax = xmin + bboxes(:,3) - 1;
ymax = ymin + bboxes(:,4) - 1;

% Perluas kotak pembatas dengan jumlah kecil.
expansionAmount = 0.02;
xmin = (1-expansionAmount) * xmin;
ymin = (1-expansionAmount) * ymin;
xmax = (1+expansionAmount) * xmax;
ymax = (1+expansionAmount) * ymax;

% Klip kotak pembatas agar berada dalam batas gambar.
xmin = max(xmin, 1);
ymin = max(ymin, 1);
xmax = min(xmax, size(I,2));
```



```
ymin = min(ymin, size(I,1));
```

```
% Tampilkan kotak pembatas yang diperluas.
```

```
expandedBBoxes = [xmin ymin xmax-xmin+1 ymax-ymin+1];
```

```
IEExpandedBBoxes = insertShape(colorImage, 'Rectangle', expandedBBoxes, 'LineWidth', 3);
```

4.2.3 Bounding Box



Gambar 4. 6 *Bounding Boxes Text*

Gambar 4.3 merupakan langkah keempat yaitu mengabungkan daerah teks untuk hasil deteksi akhir, semua hasil deteksi terdiri dari karakter teks individual. Supaya teks dapat dikenali pada metode OCR, karakter teks individu digabungkan menjadi kata-kata atau baris teks. Yang membawa informasi yang lebih bermakna daripada hanya karakter individu. Salah satu cara untuk menggabungkan masing-masing wilayah teks menjadi kata atau baris teks adalah yang pertama menemukan wilayah teks yang berdekatan dan kemudian membentuk kotak pembatas di sekitar wilayah tersebut. Hal ini membuat kotak pembatas wilayah teks tetangga tumpang tindih sehingga wilayah teks yang merupakan bagian dari kata atau garis teks yang sama membentuk rantai kotak pembatas yang tumpang tindih. Pada *detected text* kotak pembatas yang tumpang tindih dapat digabung bersama untuk membentuk kotak pembatas tunggal di sekitar kata atau garis teks individual. Kemudian hasil dari bounding box akan dipotong, berikut source code cropping bounding text :

```
croppedImage = imcrop(img, textBBoxes);
```

```
% subplot(3,3,4);
```

```
guidata(hObject,handles);
```

```
axes(handles.axes3);
```

```
imshow(croppedImage)
```



Gambar 4. 7 *Cropped Bounding Boxes Text*

Langkah kelima yaitu mengenali teks yang terdeteksi menggunakan OCR (*Optical Character Recognition*). Setelah mendeteksi daerah teks, gunakan fungsi OCR untuk mengenali teks di dalam setiap kotak pembatas. Dan mengubah ke dalam karakter string.

```
01274877
```

Gambar 4. 8 *Image to Text*

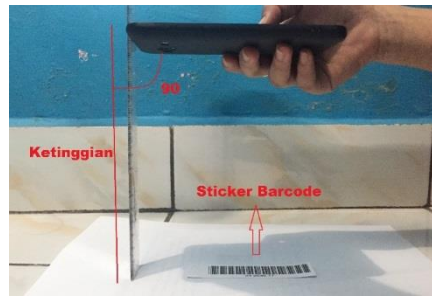
4.2 Hasil pengujian

Sesudah dilakukannya pengujian simulasi sistem penyala mesin otomatis pada *boat* berbasis *barcode*, maka hasil dari pengujian tersebut akan dianalisis sesuai parameter pengujian yang telah ditentukan, dan dari hasil pengujian tersebut akan diketahui karakteristik *image barcode* yang dapat di proses oleh sistem dengan benar.

4.2.1 Pengujian Berdasarkan Ketinggian dan Sudut

1. Spesifikasi Kamera 12 MP

Pengujian ini merupakan pengujian tahap awal untuk mencari kondisi terbaik pengambilan *image barcode* yang dapat diproses dengan baik dan benar oleh sistem menggunakan spesifikasi kamera 12 megapiksel. Pada Tabel 4.2 merupakan hasil dari pengujian mulai dari pengambilan *image barcode* sampai dengan berhasilnya mengirimkan data serial dengan parameter ketinggian dan sudut pengambilan gambar. Pengujian ini dilakukan sebanyak dua ratus kali, yang dimana tiap – tiap jarak ketinggian diujikan sebanyak empat puluh kali dengan dua sudut yang berbeda, dan dari tiap – tiap sudut tersebut akan di ujicobakan masing – masing sebanyak dua puluh kali.



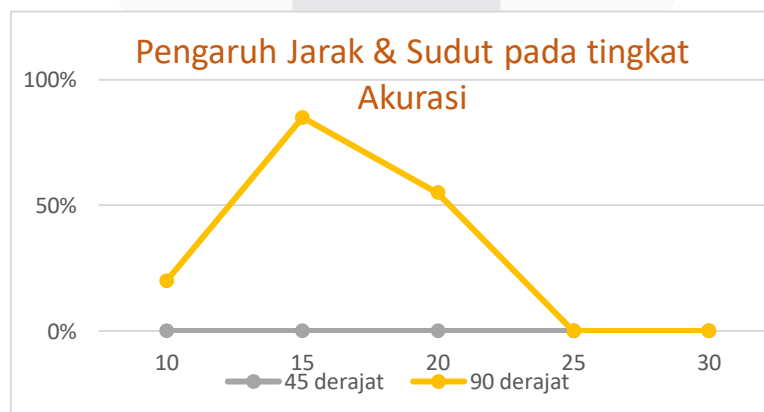
Gambar 4. 9 ketinggian dan sudut

Dari pengujian tersebut didapatkan hasil pada tabel sebagai berikut :

Tabel 4. 2 Uji Barcode Reader pada Gambar

No.	Parameter		Banyak Pengujian	Terbaca	Persentase
	Ketinggian (cm)	Sudut			
1.	10	90°	20	4	20%
		45°	20	0	0%
2.	15	90°	20	17	85%
		45°	20	0	0%
3.	20	90°	20	11	55%
		45°	20	0	0%
4.	25	90°	20	0	0%
		45°	20	0	0%
5.	30	90°	20	0	0%
		45°	20	0	0%

Dari data yang diujikan dengan beberapa parameter tersebut bahwa pada jarak ketinggian 15 cm dengan sudut 0 derajat mendapatkan persentase paling besar yaitu 85%. Pada jarak ketinggian > 20cm image barcode sudah tidak dapat dibaca oleh sistem. Dan pada pengambilan foto image barcode dengan sudut 45 derajat tidak dapat dibaca oleh sistem, yang semua hasil presentasinya mendapatkan 0%.



Gambar 4. 10 Grafik Pengujian Jarak dan Sudut

2. Spesifikasi Kamera 8 MP

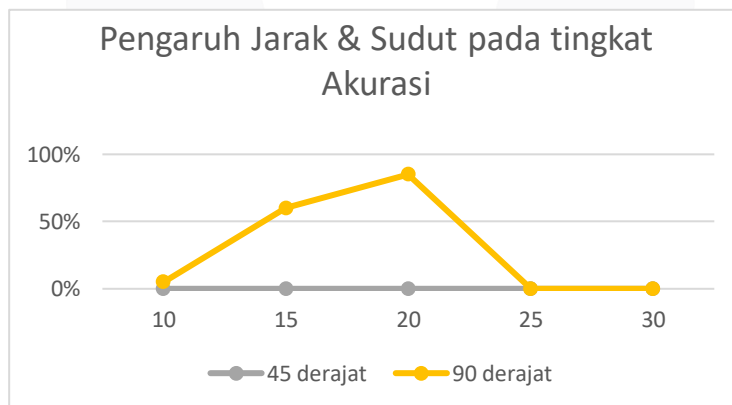
Pengujian berdasarkan jarak ketinggian dan sudut menggunakan spesifikasi kamera 8 megapiksel. Pada Tabel 4.3 merupakan hasil dari pengujian mulai dari pengambilan image barcode sampai dengan berhasilnya

mengirimkan data serial dengan parameter ketinggian dan sudut pengambilan gambar. Pengujian ini dilakukan sebanyak dua ratus kali, yang dimana tiap – tiap jarak ketinggian diujikan sebanyak empat puluh kali dengan dua sudut yang berbeda, dan dari tiap – tiap sudut tersebut akan di ujicobakan masing – masing sebanyak dua puluh kali.

Tabel 4. 3 Uji *Barcode Reader* pada Gambar

No.	Parameter		Banyak Pengujian	Terbaca	Persentase
	Ketinggian (cm)	Sudut			
1.	10	90°	20	1	5%
		45°	20	0	0%
2.	15	90°	20	12	60%
		45°	20	0	0%
3.	20	90°	20	17	85%
		45°	20	0	0%
4.	25	90°	20	0	0%
		45°	20	0	0%
5.	30	90°	20	0	0%
		45°	20	0	0%

Dari data yang diujikan dengan beberapa parameter tersebut bahwa pada jarak ketinggian 20 cm dengan sudut 90 derajat mendapatkan persentase paling besar yaitu 85%. Pada jarak ketinggian > 20cm *image barcode* sudah tidak dapat dibaca oleh sistem. Dan pada pengambilan foto *image barcode* dengan sudut 45 derajat tidak dapat dibaca oleh sistem, yang semua hasil presentasenya mendapatkan 0%.



Gambar 4. 11 Grafik Pengujian Jarak dan Sudut

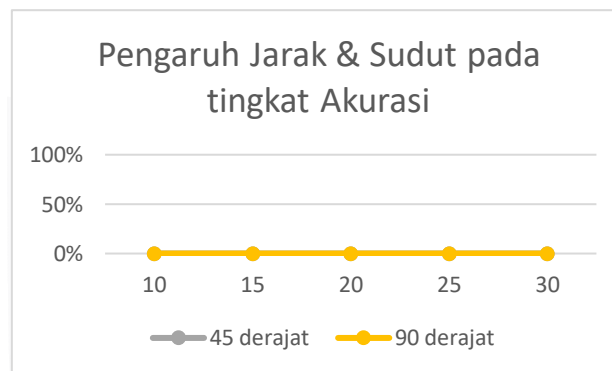
3. Spesifikasi Kamera 5 MP

Pengujian berdasarkan jarak ketinggian dan sudut menggunakan spesifikasi kamera 5 megapiksel. Pada Tabel 4.4 merupakan hasil dari pengujian mulai dari pengambilan *image barcode* sampai dengan berhasilnya mengirimkan data serial dengan parameter ketinggian dan sudut pengambilan gambar. Pengujian ini dilakukan sebanyak dua ratus kali, yang dimana tiap – tiap jarak ketinggian diujikan sebanyak empat puluh kali dengan dua sudut yang berbeda, dan dari tiap – tiap sudut tersebut akan di ujicobakan masing – masing sebanyak dua puluh kali.

Tabel 4. 4 Uji *Barcode Reader* pada Gambar

No.	Parameter		Banyak Pengujian	Terbaca	Persentase
	Ketinggian (cm)	Sudut			
1.	10	90°	20	0	0%
		45°	20	0	0%
2.	15	90°	20	0	0%
		45°	20	0	0%
3.	20	90°	20	0	0%
		45°	20	0	0%
4.	25	90°	20	0	0%
		45°	20	0	0%
5.	30	90°	20	0	0%
		45°	20	0	0%

Dari data yang diujikan dengan beberapa parameter tersebut bahwa pada semua jarak ketinggian dan sudut yang telah diuji cobakan tidak ada yang berhasil dengan keseluruhan persentase keberhasilan sebesar 0%.



Gambar 4. 12 Grafik Pengujian Jarak dan Sudut

4.2.4 Pengujian Berdasarkan Perubahan Kondisi Cahaya

Pada pengujian ini dilakukan ujicoba untuk mengetahui bagaimana performansi *image barcode* yang diproses terhadap pengaruh perubahan kondisi cahaya diluar ruangan pada pagi hari, siang hari, dan sore hari. *Image barcode* yang diambil dengan parameter jarak ketinggian 15 cm, sudut 90°, dan pengujian dilakukan sebanyak dua puluh kali. Hasil pengujian dapat dilihat pada tabel dibawah ini:

Tabel 4. 5 Uji pada Gambar Kondisi Cahaya

No.	Terbaca			No.	Terbaca		
	Pagi	Siang	Sore		Pagi	Siang	Sore
1.	√	√	√	11.	√	√	√
2.	√	√	√	12.	√	√	√
3.	√	√	√	13.	√	√	√
4.	√	√	√	14.	√	√	√
5.	√	√	√	15.	√	√	√
6.	√	√	√	16.	√	√	√
7.	√	√	√	17.	√	√	√
8.	√	√	√	18.	√	√	√
9.	√	√	√	19.	√	√	√
10.	√	√	√	20.	√	√	√
Persentase					100%	100%	100%





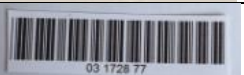














Hasil analisa bahwa pengaruh cahaya pada pagi hari, siang hari, dan sore hari tidak mempengaruhi performansi kerja sistem *barcode reader*. Jika masih terdapat cahaya yang menerangi *image barcode* yang akan diambil maka *image* tersebut masih bisa diproses oleh sistem.

4.2.3 Tingkat Akurasi Barcode Reader

4.2.3.1 Pengujian Barcode 39

Pada pengujian ini dilakukan ujicoba akurasi *barcode reader*. *Image barcode* yang diambil dengan parameter jarak ketinggian 15 cm dan sudut 90°. Dapat dilihat pada tabel berikut ini:



Tabel 4. 6 Akurasi Barcode 39





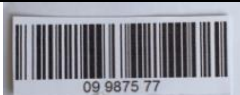






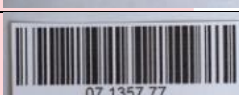


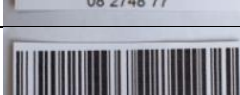
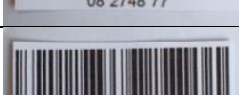


No.	Serial Barcode	Akurasi	No.	Serial Barcode	Akurasi
1.		01274877	11.		01274877
2.		02172877	12.		02172877
3.		03172877	13.		03172877
4.		04264677	14.		04264677
5.		05013877	15.		05013877
6.		01274877	16.		01274877
7.		02172877	17.		02172877
8.		03172877	18.		03172877
9.		04264677	19.		04264677
10.		05013877	20.		05013877
Persentase					100%

Pengujian ini dilakukan untuk mengetahui tingkat akurasi *barcode reader*. Tingkat akurasi data dari pembacaan *barcode 39* yang diperoleh yaitu sebesar 100% dari uji coba yang telah dilakukan sebanyak 20 kali.

4.2.3.2 Pengujian Barcode 128

Tabel 4. 7 Akurasi Barcode 128

No.	Serial Barcode	Akurasi	No.	Serial Barcode	Akurasi
1.		06123477	11.		06123477

2.		07135777	12.		07135777
3.		08274877	13.		08274877
4.		09987577	14.		09987577
5.		10987577	15.		10987577
6.		06123477	16.		06123477
7.		07135777	17.		07135777
8.		08274877	18.		08274877
9.		09987577	19.		09987577
10.		10987577	20.		10987577
Persentase					100%

Pengujian ini dilakukan untuk mengetahui persentase dari tingkat akurasi *barcode reader* menggunakan metode OCR (*Optical Character Recognition*). Tingkat akurasi data dari pembacaan *barcode* 128 yang diperoleh yaitu sebesar 100% dari uji coba yang telah dilakukan sebanyak dua puluh kali.

4.2.4 Tingkat Akurasi Database

Tabel 4. 8 Akurasi Database

No	Serial	Akurasi	No	Serial	Akurasi
1.	01274877	√	11.	01274877	√
2.	02172877	√	12.	02172877	√
3.	03172877	√	13.	03172877	√
4.	04264677	√	14.	04264677	X
5.	05013877	√	15.	05013877	√
6.	06123477	X	16.	06123477	√
7.	07135777	√	17.	07135777	√
8.	08274877	√	18.	08274877	√
9.	09987577	√	19.	09987577	X
10.	10987577	√	20.	10987577	√

Pengujian ini merupakan pengujian bahwa *image barcode* dengan kondisi yang terbaik yaitu dengan jarak ketinggian sebesar 15 cm dan sudut sebesar 90 derajat menggunakan kamera dengan kualitas 12 MP yang dapat dibaca dengan baik oleh sistem dan juga oleh database dengan presentase 85%. Dan kondisi terbaik dengan parameter jarak ketinggian sebesar 20 cm dan sudut sebesar 90 derajat menggunakan kamera dengan kualitas 8 MP.

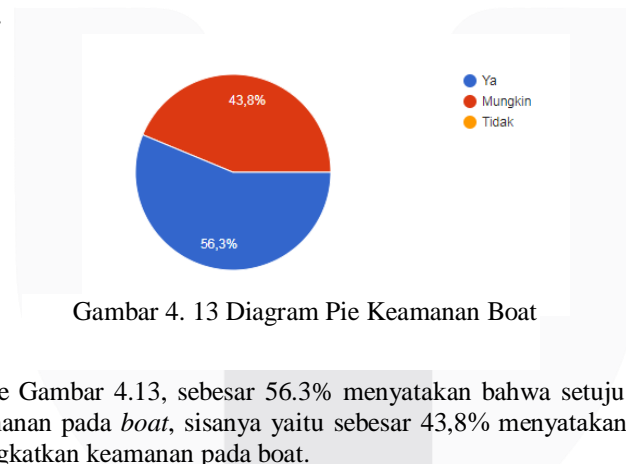
4.2.5 Waktu Pemrosesan

Tabel 4. 9 Delay

No.	Delay	No.	Delay	No.	Delay
1.	74.39 s	11.	72.30 s	21.	71.70 s
2.	75.01 s	12.	72.38 s	22.	72.00 s
3.	72.02 s	13.	72.12 s	23.	71.64 s
4.	72.64 s	14.	72.16 s	24.	72.17 s
5.	72.07 s	15.	72.15 s	25.	72.20 s
6.	77.38 s	16.	73.47 s	26.	72.28 s
7.	70.37 s	17.	73.05 s	27.	72.23 s
8.	74.88 s	18.	73.07s	28.	73.52 s
9.	72.02 s	19.	72.70 s	29.	71.44 s
10.	72.23 s	20.	74.45s	30.	74.37 s
Rata – rata					72.81 s

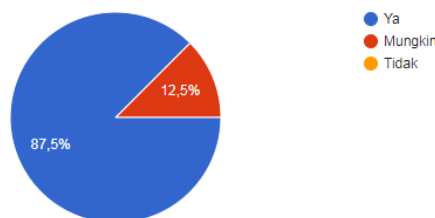
Pada pengujian ini dengan menggunakan parameter ketinggian, sudut, dan kondisi cahaya terbaik sehingga mendapatkan delay yang dihitung mulai awal *capture image barcode* sampai dengan baling – baling *boat* menyala rata – rata sebesar 72.81 detik.

4.2.6 Hasil Kuesioner



Gambar 4. 13 Diagram Pie Keamanan Boat

Pada diagram pie Gambar 4.13, sebesar 56,3% menyatakan bahwa setuju jika sistem *keyless* dapat lebih meningkatkan keamanan pada *boat*, sisanya yaitu sebesar 43,8% menyatakan ‘mungkin’ bahwa sistem *keyless* dapat lebih meningkatkan keamanan pada *boat*.



Gambar 4. 14 Diagram Pie Barcode Sebagai Pengganti Kunci

Dari hasil kuesioner yang sudah didapat, seperti pada Gambar 4.14, menunjukkan bahwa sebesar 87,5% menyatakan setuju penggunaan *barcode* berfungsi sebagai pengganti kunci. Dan sisanya sebesar 12,5% menyatakan ‘mungkin’ bahwa penggunaan *barcode* berfungsi sebagai pengganti kunci.

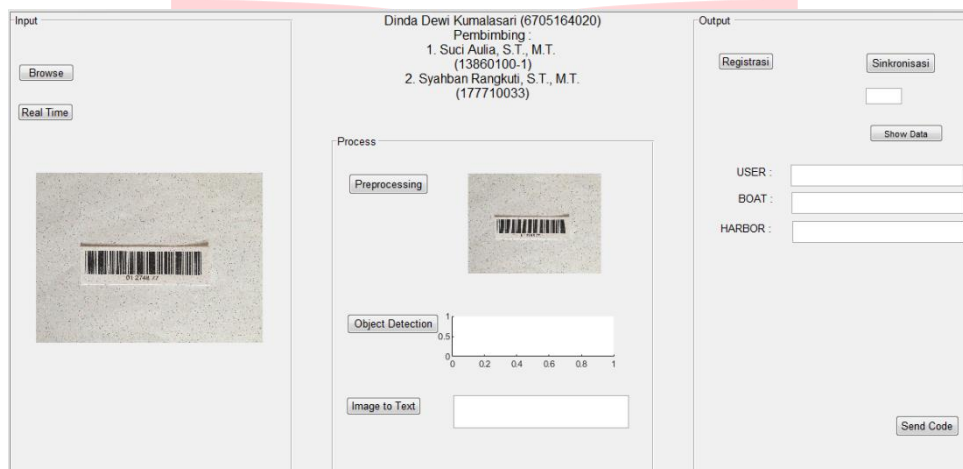
4.3. Pengujian Terhadap Noise

4.3.1 Gaussian Noise



Gambar 4. 15 *Image barcode* dengan *Gaussian noise*

Gambar 4.15 merupakan tampilan dari *image barcode* yang telah ditambahkan dengan *noise* Gaussian sebesar 3%. *Image barcode* yang telah ditambahkan dengan *noise* akan di uji cobakan pada sistem *barcode reader*.



Gambar 4. 16 uji coba Gaussian noise

Dapat dilihat pada Gambar 4.16 *image barcode* yang diproses dengan sistem ini berhenti pada proses *object detection*. Dan pada Matlab terdapat indikasi *error* seperti pada gambar 4.16 dibawah ini.

```

Index in position 2 exceeds array bounds.

Error in GUI_Proyek_Akhir>pushbutton5_Callback (line 222)
xmin = bboxes(:,1);

Error in gui_mainfcn (line 95)
feval(varargin{:});
    
```

Gambar 4. 17 indikasi error

Error terjadi pada proses *thresholding* yaitu proses dimana mendeteksi karakter – karakter yang terdapat pada sebuah *image*. Proses *thresholding* ini tidak berjalan lancar, seperti pada gambar 4.18 tidak semua karakter dapat di deteksi oleh sistem.



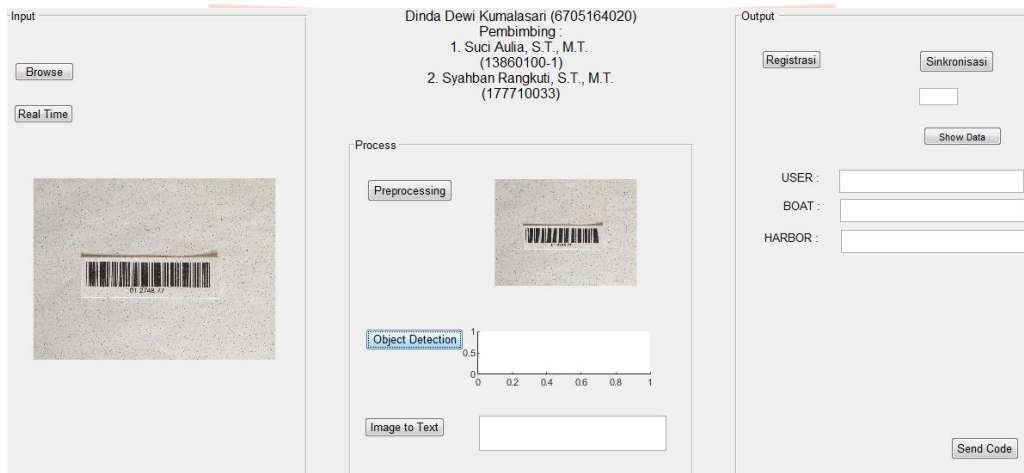
Gambar 4. 18 error pada *image barcode* noise Gaussian

4.3.2 Salt & Pepper Noise



Gambar 4. 19 Image barcode dengan noise salt & pepper

Gambar 4.19 merupakan tampilan dari *image barcode* yang telah ditambahkan dengan *noise salt & pepper* sebesar 3%. *Image barcode* yang telah ditambahkan dengan *noise* akan di uji cobakan pada sistem *barcode reader*.



Gambar 4. 20 uji coba salt & pepper

Dapat dilihat pada Gambar 4.20 *image barcode* yang diproses dengan sistem ini berhenti pada proses object detection. Dan pada Matlab terdapat indikasi error seperti pada Gambar 4.21 dibawah ini.

```

Index in position 2 exceeds array bounds.

Error in GUI_Projek_Akhir>pushbutton5_Callback (line 222)
xmin = bboxes(:,1);

Error in gui_mainfcn (line 95)
feval(varargin{:});
    
```

Gambar 4. 21 indikasi error

Error terjadi pada proses *thresholding* yaitu proses dimana mendeteksi karakter – karakter yang terdapat pada sebuah *image*. Proses *thresholding* ini tidak berjalan lancar, seperti pada gambar 4.22 tidak semua karakter dapat di deteksi oleh sistem.



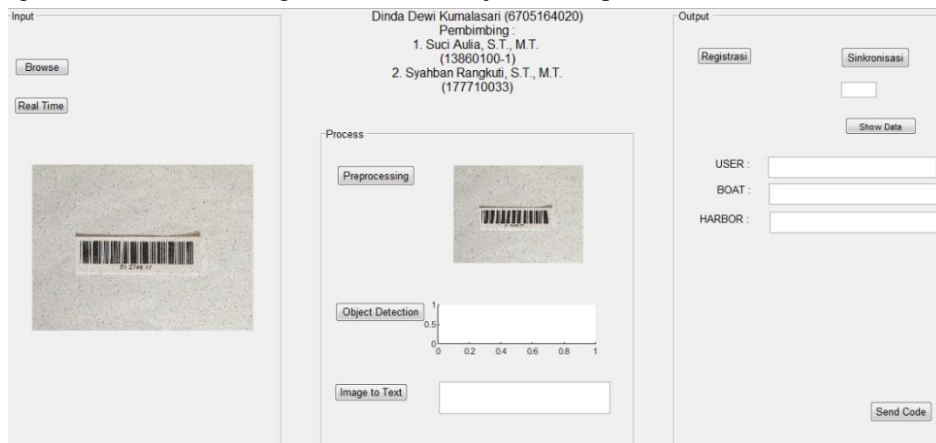
Gambar 4. 22 error pada image barcode noise Salt & Pepper

4.3.3 Poisson Noise



Gambar 4. 23 Image barcode dengan noise poisson

Gambar 4.23 merupakan tampilan dari *image barcode* yang telah ditambahkan dengan *noise poisson*. *Image barcode* yang telah ditambahkan dengan *noise* akan di uji cobakan pada sistem *barcode reader*.



Gambar 4. 24 uji coba poisson noise

Dapat dilihat pada Gambar 4.24 *image barcode* yang diproses dengan sistem ini berhenti pada proses *object detection*. Dan pada Matlab terdapat indikasi *error* seperti pada gambar 4.24 dibawah ini.

```

Index in position 2 exceeds array bounds.

Error in GUI_Proyek_Akhir>pushbutton5_Callback (line 222)
xmin = bboxes(:,1);

Error in gui_mainfcn (line 95)
feval(varargin{:});
    
```

Gambar 4. 25 Indikasi error

Error terjadi pada proses *thresholding* yaitu proses dimana mendeteksi karakter – karakter yang terdapat pada sebuah *image*. Proses *thresholding* ini tidak berjalan lancar, seperti pada gambar 4.26 tidak semua karakter dapat di deteksi oleh sistem.



Gambar 4. 26 error pada image barcode noise poisson

5. Kesimpulan dan Saran

Kesimpulan

Berdasarkan paparan yang telah dijelaskan pada bab – bab sebelumnya, bahwa sistem simulasi *barcode reader* menggunakan aplikasi Matlab, dapat ditarik kesimpulan:

1. Pada sistem simulasi ini diperlukannya kondisi gambar dengan parameter jarak ketinggian 15 cm, sudut 90°, dan cahaya yang cukup agar dapat diproses.

2. Tingkat akurasi *barcode reader* pada jenis *barcode* 39 dan 128 yaitu 100%.
3. Waktu proses tidak membutuhkan waktu yang lama yaitu rata – rata 38.31 detik.
4. Pada sistem ini diperlukannya kamera *smartphone android* yang fokus dan tanpa filter tambahan.
5. *Image barcode* dapat dibaca dengan oleh sistem dan juga oleh database dengan presentase sebesar 85%.
6. Data yang sudah diproses pada *barcode reader*, berhasil terkirim melalui komunikasi serial wireless menggunakan *Bluetooth*.
7. *Image Barcode* dengan *noise* tambahan (*noise* buatan) tidak dapat diproses oleh sistem.

Saran

Berdasarkan hasil pembuat simulasi sistem penyalah mesin otomatis pada *boat* berbasis, maka penulis dapat mengambil saran sebagai berikut:

1. Untuk perancangan simulasi selanjutnya perlu adanya pengujian terhadap spesifikasi kamera *smartphone* yang digunakan untuk akuisisi citra.

DAFTAR PUSTAKA

- [1] A. K. Boyat and B. K. Joshi. 2015. *A Review Paper : Noise Models in Digital Image Processing*. Signal & Image Processing : An International Journal. Vol. 6. No.2. Page:63-75
- [2] American Barcode and RFID. 2014. Barcode Basics, Basic concept for barcode and automatic identification beginners. Page: 1-20.
- [3] C. Saravanan. 2010. *Color image to grayscale image conversion*. 2010 2nd International Conference on Computer Engineering and Applications, ICCEA 2010. Vol. 2. Page:196-199.
- [4] D. Addiati, U. U. Sheikh and S. A.R. Abu Bakar. 2010. *Restoration of out of focus barcode images using Wiener filter*. Proceedings - 5th International Conference on Signal Image Technology and Internet Based Systems, SITIS 2009. Page: 133-137.
- [5] E. Ohbuchi, H. Hanaizumi and L. Hock. 2004. *Barcode readers using the camera device in mobile phones*. Proceedings - 2004 International Conference on Cyberworlds, CW 2004. Page: 260-265.
- [6] G. T. Sutar, A. V Shah, and P. G. Student, 2007. "Number Plate Recognition Using an Improved Segmentation," *International Journal of Innovative Research in Science, Engineering and Technology (An ISO Certified Organization)*, vol. 3297, no. 5, pp. 12360–12368.
- [7] J. Z. Gao, L. Prakash and R. Jagatesan. 2007. *Understanding 2D-BarCode technology and applications in M-commerce - Design and implementation of A 2D barcode processing solution*. Proceedings - International Computer Software and Applications Conference. Vol. 2. No. Compsac. Page: 49-56.
- [8] T. Hu and T. Yang. 2013. *Application of computational verb image processing to 1-D barcode localization*. Proceedings of the International Conference on Anti-Couterfeiting, Security and Identification, ASID. Page: 3-6.
- [9] T. S. Gunawan, A. Mutholib, and M. Kartiwi, Aug. 2017. "Performance Evaluation of Automatic Number Plate Recognition on Android Smartphone Platform," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 4, p. 1973,
- [10] T. Wakahara and N. Yamamoto. 2011. *Image Processing of 2-dimensional barcode*. Proceedings - 2011 International Conference on Network-Based Information Systems, NBIS 2011. No. 1. Page: 484-490.