



***PROTOTYPE SISTEM DETEKSI PLAT NOMOR KENDARAAN PADA
SISTEM PERPARKIRAN BERBASIS *IMAGE PROCESSING*
MENGUNAKAN METODE *OPTICAL
CHARACTER RECOGNITION (OCR)****

SKRIPSI

Oleh

**Rizaldy Rohimawan Santoso
NIM 121910201027**

**PROGRAM STUDI STRATA-1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2019**



***PROTOTYPE SISTEM DETEKSI PLAT NOMOR KENDARAAN PADA
SISTEM PERPARKIRAN BERBASIS IMAGE PROCESSING
MENGUNAKAN METODE OPTICAL
CHARACTER RECOGNITION (OCR)***

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Teknik Elektro (S1)
dan mencapai gelar Sarjana Teknik

oleh

Rizaldy Rohimawan Santoso
NIM 121910201027

**PROGRAM STUDI STRATA-1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2019**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang telah memberikan limpahan berkah dan nikmat yang luar biasa kepadaku baik kesempatan dan kekuatan lahir batin untuk menuntut ilmu dan menyelesaikan tahap ini beserta Nabi Muhammad SAW yang selalu menjadi panutan dalam setiap langkah;
2. Orang tua tercinta yang telah memberikan doa, bimbingan, kasih sayang, dukungan, kerja keras, perjuangan dan pengorbanan kepada saya dalam meraih segala impian dan cita-cita;
3. Adik tersayangku atas doa, semangat, dan dukungan yang diberikan kepadaku;
4. Semua guru dan dosen sejak taman kanak-kanak hingga perguruan tinggi yang telah ikhlas mendidik dan memberikan ilmu pengetahuan dengan penuh kasih sayang dan kesabaran kepada penulis;
5. Almamater tercinta, Fakultas Teknik Universitas Jember yang saya banggakan sebagai tempat bagi saya untuk menimba ilmu khususnya dibidang teknik elektro.

MOTTO

“Urusan seorang mukmin patut dikagumi. Semua urusannya merupakan kebaikan bagi dirinya dan tidak terdapat kecuali pada diri seorang mukmin. Apabila memperoleh kesenangan dia bersyukur dan itu baik untuk dirinya. Dan bila ditimpa kesusahan dia bersabar dan itu baik untuk dirinya”. (HR.Imam Muslim)

“Allah akan meninggikan orang-orang yang beriman diantara kamu dan orang-orang yang berilmu pengetahuan beberapa derajat. Dan Allah Maha Mengetahui apa yang kamu kerjakan” (Al-Mujadillah:11)



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Rizaldy Rohimawan Santoso

Nim : 121910201027

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul: “*Prototype Sistem Deteksi Plat Nomor Kendaraan Pada Sistem Perparkiran Berbasis Image Processing Menggunakan Metode Optical Character Recognition (OCR)*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 29 Mei 2019

Yang Menyatakan,

Rizaldy Rohimawan Santoso

NIM 121910201027

SKRIPSI

***PROTOTYPE SISTEM DETEKSI PLAT NOMOR KENDARAAN PADA
SISTEM PERPARKIRAN BERBASIS IMAGE PROCESSING
MENGUNAKAN METODE OPTICAL
CHARACTER RECOGNITION (OCR)***

Oleh

Rizaldy Rohimawan Santoso

NIM 121910201027

Pembimbing:

Dosen Pembimbing Utama : Sumardi S.T., M.T.

Dosen Pembimbing Anggota : Widya Cahyadi S.T., M.T.

PENGESAHAN

Skripsi berjudul “*Prototype Sistem Deteksi Plat Nomor Kendaraan Pada Sistem Perparkiran Berbasis Image Processing Menggunakan Metode Optical Character Recognition (OCR)*” telah diuji dan disahkan pada:

hari, tanggal : Rabu, 29 Mei 2019

tempat : Fakultas Teknik Universitas Jember

Tim Penguji

Pembimbing Utama,

Sumardi S.T., M.T.
NIP. 19670113 199802 1 001

Penguji I,

Khairul Anam S.T.,M.T., Ph.D.
NIP. 19780405 200501 1 002

Pembimbing Anggota,

Widya Cahyadi S.T., M.T.
NIP. 19851110 201404 1 001

Penguji II,

Catur Suko Sarwono S.T., M.Si
NIP. 19680119 199702 1 001

Mengesahkan
Dekan Fakultas Teknik Universitas Jember,

Dr. Ir. Entin Hidayah, M. U. M.
NIP 19661215 199503 2 001

Prototype sistem deteksi plat nomor kendaraan pada sistem perparkiran berbasis *image processing* menggunakan metode *optical character recognition* (OCR)

Rizaldy Rohimawan Santoso

Program Studi Teknik Elektro, Fakultas Teknik, Universitas Jember

ABSTRAK

Image Processing atau sering disebut dengan pengolahan citra digital merupakan metode yang digunakan untuk mengolah atau memproses dari gambar asli sehingga menghasilkan gambar lain yang sesuai dengan kebutuhan. Pada penelitian ini, kita menggunakan salah satu metode *image processing* yaitu OCR (*Optical Character Recognition*) merupakan metode *image processing* yang berfungsi sebagai pembacaan bentuk *image* menjadi karakter. Metode OCR (*Optical Character Recognition*) di dalamnya terdapat sebuah arsitektur *tesseract* OCR yang berfungsi sebagai *database* karakter. Pada penelitian ini, di lakukan proses uji di mana setelah pengambilan gambar plat nomor akan di proses oleh aplikasi untuk pembacaan karakter plat nomor, setelah mendapatkan karakter plat nomor proses selanjutnya akan di aplikasikan pada sistem perparkiran yaitu di proses untuk pembukaan pintu masuk dan pintu keluar. Hasil pengujian dari sistem deteksi plat nomor kendaraan yaitu beberapa plat nomor kendaraan bisa dilakukan pembacaan karakter dengan metode OCR (*Optical Character Recognition*) dan pada metode OCR untuk karakter Q diidentifikasi jadi karakter 0, sedangkan pada karakter L diidentifikasi jadi karakter 1 akan tetapi jika hasil proses *image processing* akhir yaitu karakter citra terhapus ataupun terdapat banyak noise pada karakter citra maka susah untuk pembacaan karakter.

Kata kunci: *Image Processing*, OCR (*Optical Character Recognition*), Sistem Perparkiran

The vehicle number plate detection system prototype on image processing based parking systems uses the method of optical character recognition (OCR)

Rizaldy Rohimawan Santoso

*Departement of Electrical Engineering, Faculty of Engineering, University of
Jember*

ABSTRACT

Image Processing or often referred to as digital image processing is a method used to process or process the original image so that it produces another image that fits your needs. In this study, we use one method of image processing or image processing, namely OCR (Optical Character Recognition) is an image processing method that functions as a reading of the image shape into characters. The OCR (Optical Character Recognition) method in which there is a tesseract OCR architecture that functions as a character database. In this study, a test process was carried out in which after taking the license plate image the process will be processed for the reading of number plate characters, after obtaining the number plate character the next process will be applied to the parking system, which is the opening and exit doors. The test results from the vehicle number plate detection system, which are several vehicle number plates, and can be read by OCR (Optical Character Recognition) method, the OCR method for Q characters is identified to be 0, while L character is identified to be 1 but if the final image processing process, namely the image character is deleted or there is a lot of noise in the image character, it is difficult to read characters.

Keywords: *Image Processing, OCR (Optical Character Recognition), Parking systems*

RINGKASAN

Prototype Sistem Deteksi Plat Nomor Kendaraan Pada Sistem Perparkiran Berbasis Image Processing Menggunakan Metode Optical Character Recognition (OCR); Rizaldy Rohimawan Santoso, 121910201027; 2019; 72 halaman; Fakultas Teknik Universitas Jember.

Image Processing atau sering disebut dengan pengolahan citra digital merupakan metode yang digunakan untuk mengolah atau memproses dari gambar asli sehingga menghasilkan gambar lain yang sesuai dengan kebutuhan. Pada penelitian ini, kita menggunakan salah satu metode *image processing* atau pengolahan citra yaitu OCR (*Optical Character Recognition*) merupakan metode *image processing* yang berfungsi sebagai pembacaan bentuk *image* menjadi karakter. Metode OCR (*Optical Character Recognition*) di dalamnya terdapat sebuah arsitektur *tesseract* OCR yang berfungsi sebagai *database* karakter. Pada penelitian ini, di lakukan proses uji di mana setelah pengambilan gambar plat nomor akan di proses oleh aplikasi untuk pembacaan karakter plat nomor, setelah mendapatkan karakter plat nomor proses selanjutnya akan di aplikasikan pada sistem perparkiran yaitu di proses untuk pembukaan pintu masuk dan pintu keluar. Selain itu, pada penelitian ini juga dilakukan pengambilan waktu kendaraan selama parkir pada saat kendaraan masuk ataupun keluar dan akan dihasilkan jumlah pembayaran sesuai dengan lama waktunya kendaraan di parkir, selanjutnya penelitian ini juga akan disimpan sebagai *database* sebagai hasil plat nomor kendaraan yang telah melewati pintu masuk dan pintu keluar.

Hasil pengujian dari sistem deteksi plat nomor kendaraan yaitu beberapa plat nomor kendaraan bisa dilakukan pembacaan karakter dengan metode OCR (*Optical Character Recognition*) akan tetapi jika hasil proses *image processing* akhir yaitu karakter citra terhapus ataupun terdapat banyak noise pada karakter citra maka susah untuk pembacaan karakter.

PRAKATA

Puji syukur ke hadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “*Prototype Sistem Deteksi Plat Nomor Kendaraan Pada Sistem Perparkiran Berbasis Image Processing Menggunakan Metode Optical Character Recognition (OCR)*”. Skripsi ini disusun guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan studi strata satu (S1) pada Fakultas Teknik Universitas Jember.

Penulis menyadari sepenuhnya bahwa dalam penulisan skripsi ini tidak lepas dari dukungan, bantuan serta motivasi dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terimakasih kepada:

1. Allah SWT atas semua karunia yang diberikan, sehingga penulis dapat menyelesaikan tugas akhir untuk mencapai gelar Sarjana Teknik;
2. Ibu Dr. Ir. Entin Hidayah M.U.M selaku Dekan Fakultas Teknik Universitas Jember;
3. Bapak Sumardi S.T., M.T. selaku Dosen Pembimbing Utama dan Bapak Dr. Widya Cahyadi S.T., M.T. selaku Dosen Pembimbing Anggota yang telah banyak membantu, meluangkan waktu, pikiran, dan perhatiannya untuk membimbing penulis untuk menyelesaikan penelitian dan skripsi ini;
4. Bapak Khairul Anam S.T.,M.T., Ph.D. selaku Dosen Penguji I dan Bapak Catur Suko Sarwono S.T., M.Si. selaku Dosen Penguji II yang telah meluangkan waktu untuk menguji, mengevaluasi, memberikan saran dan penilaian terhadap skripsi ini;
5. Bapak Suprihadi Prasetyono, S.T., M.T. selaku Dosen Pembimbing Akademik atas segala bimbingan dan kesabarannya dalam membimbing penulis selama masa studi sampai tugas akhir;
6. Bapak dan Ibu Dosen Fakultas Teknik Universitas Jember yang telah mengamalkan ilmunya dan memperluas ilmu pengetahuan serta wawasan penulis selama menempuh masa kuliah;

7. Ayahanda Bagong Budi Santoso, SP. dan Ibunda Dra. Susila serta adikku Ainisatur Rohmaning Budi tercinta yang sejak aku lahir sampai dengan sekarang dan kelak akan senantiasa memberikan dukungan, doa dan pengorbanan serta menjadi sumber motivasi dan inspirasi;
8. Keluarga Besar Elektro Universitas Jember yang telah mendukung dengan segenap jiwa , cinta dan rasa;
9. Teman-teman seperjuangan Angkatan 2012 Teknik Elektro Universitas Jember “SATE UJ” semuanya atas kebaikan dan bantuan selama kuliah ini;
10. Dan semua pihak yang tidak dapat disebutkan satu per satu atas bantuan, perhatian, dan inspirasi bagi penulis dalam menyelesaikan penelitian ini dan mendoakan kesuksesan ujian skripsi ini.

Penulis menerima saran dan kritik yang membangun dari pembaca sekalian demi kesempurnaan skripsi ini. Penulis juga berharap semoga skripsi ini dapat memberikan manfaat bagi perkembangan ilmu pengetahuan.

Jember, 29 Mei 2019

Penulis

DAFTAR ISI

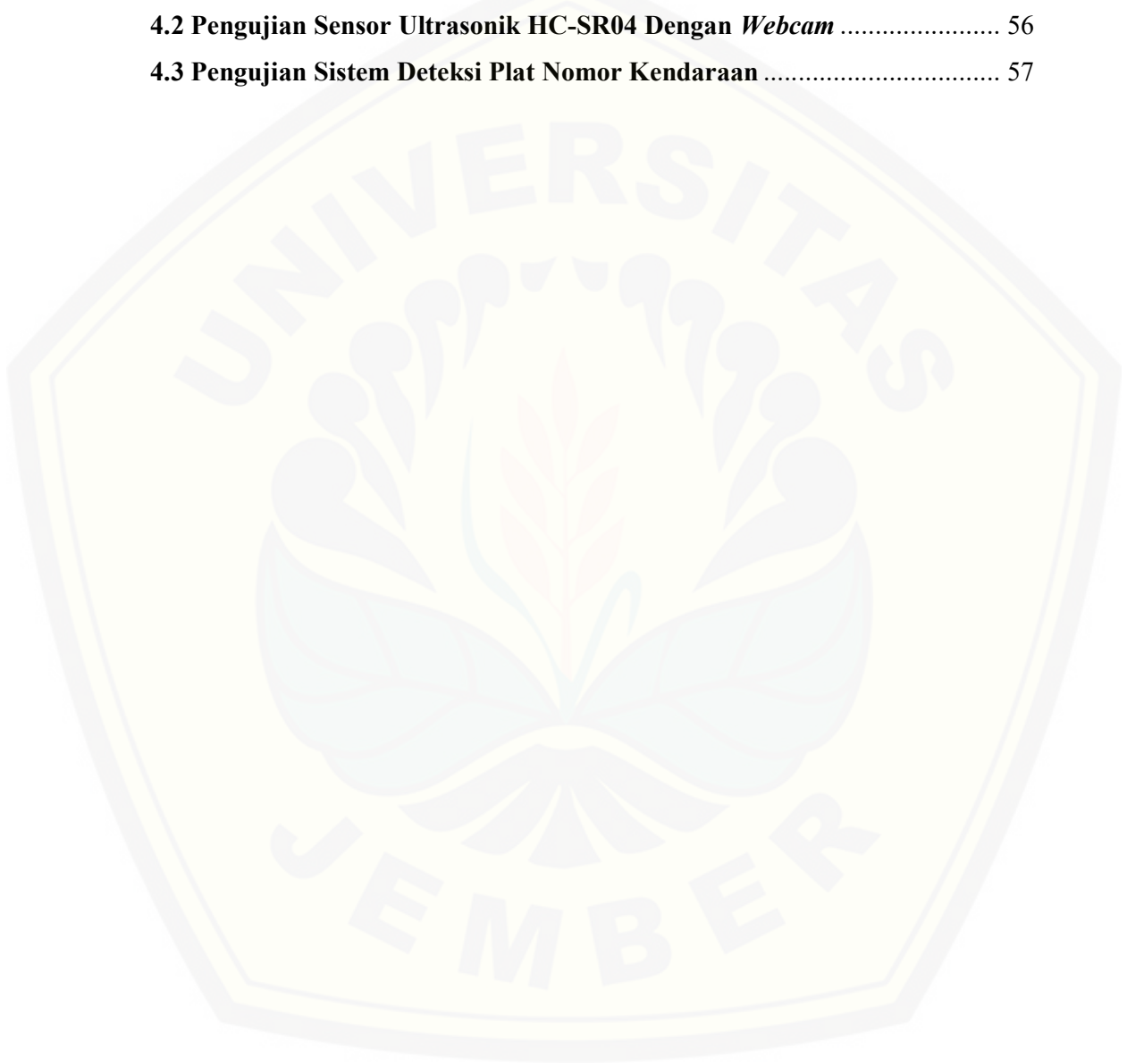
	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
HALAMAN MOTTO	iii
HALAMAN PERNYATAAN	iv
HALAMAN PEMBIMBINGAN	v
HALAMAN PENGESAHAN	vi
ABSTRAK	vii
<i>ABSTRACT</i>	viii
RINGKASAN	ix
PRAKATA	x
DAFTAR ISI	xii
DAFTAR TABEL	xv
DAFTAR GAMBAR	xvi
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
BAB 2. TINJAUAN PUSTAKA	5
2.1 Plat Nomor Kendaraan Bermotor di Indonesia	5
2.1.1 Sejarah Penggunaan Plat Nomor Kendaraan Bermotor di Indonesia	5
2.1.2 Spesifikasi Teknis	5
2.1.3 Warna	6
2.2 Webcam	7
2.3 Image Processing	8
2.1.2 Pengertian <i>Image Processing</i>	8

2.1.3 Representasi Citra Digital.....	9
2.4 Proses Pengolahan Citra Warna	9
2.5 Citra <i>Grayscale</i> (Keabuan)	12
2.6 <i>Thresholding</i>	13
2.7 Citra Biner	13
2.8 <i>Erosion</i>.....	14
2.9 <i>Dilation</i>	15
2.10 <i>Software Visual Studio</i>	15
2.11 EmguCV	16
2.12 <i>Optical Character Recognition (OCR)</i>	17
2.13 <i>Tesseract OCR</i>	19
2.13.1 Arsitektur <i>Tesseract</i>	19
2.14 Arduino Uno.....	20
2.14.1 Catu Daya	21
2.15 Motor DC	23
2.15.1 <i>Power Windows</i>	23
2.16 Sensor <i>Hall effect</i>.....	24
2.17 Relay	26
2.18 Sensor Ultrasonik HC-SR04	28
BAB 3. METODOLOGI PENELITIAN	32
3.1 Tempat Dan Waktu Penelitian	32
3.1 Tahapan Penelitian	32
3.3 Perancangan Sistem	33
3.3.1 Diagram Blok Deteksi Plat Nomor	33
3.3.2 Diagram Blok Alat Deteksi Plat Nomor Pada Sistem Perparkiran	34
3.4 Perancangan Sistem	36
3.4.1 Desain Alat Sistem Perparkiran	36
3.4.2 Rancangan Keseluruhan Alat	37
3.5 Perancangan <i>Hardware</i>	38
3.5.1 Perancangan Rangkaian <i>Power Supply</i>	38

3.5.2 <i>Layout PCB</i>	39
3.5.3 Arduino Uno Dengan Sensor HC-SR04	39
3.5.4 Arduino Uno Dengan Sensor Magnetik	40
3.5.5 Arduino Uno Dengan Motor DC.....	41
3.5.6 <i>Webcam</i>	42
3.6 <i>Flowchart</i>	43
3.7 Perancangan <i>Software</i>	38
3.7.1 Proses Deteksi Plat Nomor Kendaraan.....	46
3.7.2 <i>Input Citra</i>	46
3.7.3 <i>Filtering Citra</i>	47
3.7.4 Metode <i>Optical Character Recognition (OCR)</i>	51
3.7.5 Aplikasi Prototype Sistem Deteksi Plat Nomor.....	52
3.7.5 <i>Database Software Visual Studio</i>	53
3.7.5 <i>Database Microsoft Excel</i>	53
BAB 4. PEMBAHASAN DAN ANALISIS	55
4.1 Analisa Perangkat Keras <i>Prototype</i> Sistem Deteksi Plat Nomor Kendaraan Menggunakan Metode <i>Optical Character Recognition (OCR)</i> Pada Sistem Perparkiran	55
4.1.1 Pengujian Sensor Ultrasonik HC-SR04 Yang Terhubung Antara Visual Studio Dengan Arduino.....	55
4.2 Analisa Perangkat Lunak <i>Prototype</i> Sistem Deteksi Plat Nomor Menggunakan Metode <i>Optical Character Recognition (OCR)</i> Pada Sistem Perparkiran	57
4.2.1 Hasil Pengujian Data <i>Prototype</i> Sistem Deteksi Plat Nomor Kendaraan Menggunakan Metode <i>Optical Character Recognition (OCR)</i> Pada Sistem Perparkiran.....	57
BAB 5. KESIMPULAN DAN SARAN	71
5.1 Kesimpulan	71
5.2 Saran	71
DAFTAR PUSTAKA	73

DAFTAR TABEL

	Halaman
4.1 Pengujian Sensor Ultrasonik HC-SR04	55
4.2 Pengujian Sensor Ultrasonik HC-SR04 Dengan <i>Webcam</i>	56
4.3 Pengujian Sistem Deteksi Plat Nomor Kendaraan	57



DAFTAR GAMBAR

	Halaman
2.1 Plat Nomor Kendaraan Bermotor di Indonesia.....	5
2.2 <i>Webcam</i>	8
2.3 Nilai warna RGB dalam <i>hexadecimal</i>	10
2.4 Komposisi warna RGB	10
2.5 Contoh konversi citra ke <i>Gray</i>	12
2.6 Contoh Derajat Kualitas <i>Tresholding</i>	13
2.7 Citra biner dan representasi citra binernya	14
2.8 Contoh Hasil Gambar Yang Di Proses Menggunakan Operasi Morphologi Erosi.....	15
2.9 Contoh Hasil Gambar Yang Di Proses Menggunakan Operasi Morphologi Dilasi	16
2.10 Tampilan dinding kerja Microsoft Visual Studio 2015 Enterprise	17
2.11 Arsitektur <i>tesseract</i>	19
2.12 Arduino UNO	20
2.13 Deskripsi Arduino Uno R3.....	22
2.14 Spesifikasi Motor <i>Power Windows</i>	24
2.15 Motor <i>Power Windows</i>	24
2.16 Cara Kerja Sensor <i>Hall Effect</i>	25
2.17 Konstruksi Relai Elektro Mekanik Posisi NC (Normally Close).....	27
2.18 Konstruksi Relai Elektro Mekanik Posisi NO (Normally Open)	27
2.19 Relay	29
2.20 Sensor Ultrasonik HC-SR04	29
3.1 Blok diagram Tahap Penelitian.....	32
3.2 Diagram Blok Alat Deteksi Plat Nomor Pada Sistem Perparkiran	33
3.3 Diagram Blok Alat Deteksi Plat Nomor Pada Sistem Perparkiran	34
3.4 Alat Sistem Perparkiran Untuk Pintu Masuk	36
3.5 Alat Sistem Perparkiran Untuk Pintu Keluar	36

3.6 Gambar keseluruhan alat	37
3.7 Rancangan Rangkaian <i>Power Supply</i> Simetris	38
3.8 <i>Layout PCB Power Supply</i> Simetris, <i>Power Supply Regulator</i> , Dan <i>Driver Motor</i>	39
3.9 Rangkaian Arduino Dengan HC-SR04	39
3.10 Rangkaian Arduino Dengan Sensor <i>Magnetic</i>	40
3.11 Rangkaian Arduino Dengan <i>Driver Motor</i> dan Motor DC	41
3.12 <i>Webcam Logitech C270</i>	41
3.13 Flowchart Perancangan Sistem Deteksi Nomor Plat Nomor Pada Pintu Masuk Dan Keluar	43
3.14 Flowchart <i>Save Data Output</i> Plat Nomor ke DataBase Pada Pintu Masuk Dan Keluar	44
3.15 Flowchart Sistem Perparkiran	45
3.16 Input Citra RGB 1	47
3.17 Input Citra RGB 2	47
3.18 Citra <i>Image 1</i> Dengan <i>Filtering Grayscale</i>	48
3.19 Citra <i>Image 2</i> Dengan <i>Filtering Grayscale</i>	48
3.20 Citra <i>Image 1</i> Dengan <i>Filtering Thresholding</i>	49
3.21 Citra <i>Image 2</i> Dengan <i>Filtering Thresholding</i>	49
3.22 Citra <i>Image 1</i> dengan Proses <i>Erosion</i>	49
3.23 Citra <i>Image 1</i> dengan Proses <i>Erosion</i>	50
3.24 Citra <i>Image 1</i> dengan Proses <i>Dilate 1x</i>	50
3.25 Citra <i>Image 1</i> dengan Proses <i>Dilate 2x</i>	50
3.26 Citra <i>Image 2</i> dengan Proses <i>Dilate 1x</i>	51
3.27 Citra <i>Image 2</i> dengan Proses <i>Dilate 2x</i>	51
3.28 Proses Metode OCR Pada Citra <i>Image 1</i> Dengan Hasil P 1124 SJ	51
3.29 Proses Metode OCR Pada Citra <i>Image 2</i> Dengan Hasil P 1124 SJ	52
3.30 Aplikasi Dapat Berfungsi	52
3.31 Aplikasi Tidak Dapat Berfungsi	53
3.32 Hasil <i>Database Software Visual Studio</i>	53
3.33 Hasil <i>Databse Software Visual Studio</i>	54

4.1 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1285 QC Pada Pintu Masuk	59
4.2 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 375 NL Pada Pintu Masuk	59
4.3 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1285 QC Pada Pintu Keluar	60
4.4 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 375 NL Pada Pintu Keluar	60
4.5 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1226 QI Pada Pintu Masuk	61
4.6 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1226 QI Pada Pintu Keluar	61
4.7 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1501 KX Pada Pintu Masuk	62
4.8 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1501 KX Pada Pintu Keluar	62
4.9 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1146 TY Pada Pintu Masuk	63
4.10 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1146 TY Pada Pintu Keluar	63
4.11 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1124 SJ Pada Pintu Masuk	64
4.12 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1124 SJ Pada Pintu Keluar	65
4.13 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1124 SJ Total Pembayaran	65
4.14 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1086 MW Pada Pintu Masuk	65
4.15 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1086 MW Pada Pintu Keluar	66

4.16 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1086 MW	
Total Pembayaran	66
4.17 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1081 RU	
Pada Pintu Masuk	66
4.18 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1081 RU	
Pada Pintu Keluar	67
4.19 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1081 RU	
Total Pembayaran	67
4.21 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 960 ML	
Pada Pintu Masuk	67
4.22 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 960 ML	
Pada Pintu Keluar	68
4.23 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 960 ML	
Total Pembayaran	68
4.24 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1285 QC	
Pada Pintu Masuk	68
4.25 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1285 QC	
Pada Pintu Keluar	69
4.16 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1285 QC	
Total Pembayaran	69
4.26 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1147 TV	
Pada Pintu Masuk	69
4.27 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1147 TV	
Pada Pintu Keluar	70
4.28 Proses Pengenalan Karakter Pada Plat Nomor Kendaraan P 1147 TV	
Total Pembayaran	70

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi di Indonesia yang kini terus semakin berkembang pesat yang menyebabkan banyak kemudahan dalam melakukan kegiatan sehari – hari. Hal ini menimbulkan dampak kepada sistem alat – alat elektronika yang kegunaannya menjadi serba otomatis dan masyarakat yang lebih memilih menggunakan cara praktis dan efisien. Akan tetapi masih ada beberapa sistem yang digunakan secara manual misalnya pada sistem perparkiran sehingga masih dilakukan dengan bantuan manusia untuk melakukan mencatatkan plat nomor. Dengan cara ini sangat memiliki banyak kelemahan yaitu terletak pada manusia. Manusia yang memiliki sifat bosan dan lelah dapat menimbulkan terjadinya kesalahan dan bahkan faktor lainnya yaitu proses mencatatkan memerlukan waktu yang lama, Sedangkan untuk teknologi yang sudah berkembang di luar Negara Indonesia pada system perparkiran sudah menggunakan metode sistem *card* dan juga sistem sudah banyak yang dijalankan secara otomatis dan bahkan pada sistem perparkiran di luar Negara Indonesia sudah menggunakan metode pembayaran tiap jam dan plat nomor khusus untuk area *free* parkir.

Image Processing atau sering disebut dengan pengolahan citra digital merupakan metode yang digunakan untuk mengolah atau memproses dari gambar asli sehingga menghasilkan gambar lain yang sesuai dengan kebutuhan. Pada penelitian ini, kita menggunakan salah satu metode *image processing* atau pengolahan citra yaitu OCR (*Optical Character Recognition*) merupakan metode *image processing* yang berfungsi sebagai pembacaan bentuk *image* menjadi karakter. Metode OCR (*Optical Character Recognition*) di dalamnya terdapat sebuah arsitektur *tesseract* OCR yang berfungsi sebagai *database* karakter. Pada metode ini menggunakan kamera digital yang berfungsi sebagai deteksi plat nomor pada sistem perparkiran sangatlah efektif, karena dengan *image processing* menggunakan metode OCR (*Optical Character Recognition*) langsung diproses

oleh computer dimana proses ini bisa dilakukan secara *real time* maupun non *real time* sehingga dengan metode ini dapat mengurangi kesalahan-kesalahan dan menjadi solusi sistem perparkiran secara manual yang ada di Indonesia.

Pada penelitian sebelumnya yang berjudul “*Deteksi Plat Nomor Kendaraan Bermotor Berdasarkan Area pada Image Segmentation*” dari penelitian tersebut menggunakan metode interpolasi yaitu menghitung bobot rata-rata beberapa pixel terhadap satu titik, selanjutnya akan dilakukan penggabungan pixel dengan jarak tertentu dan proses lanjutan dilakukan tahap segmentasi dari area yang sudah didapatkan (Wakhidah, Nur. 2012). Penelitian selanjutnya yang berjudul “*Perancangan Program Deteksi Dan Pengenalan Plat Nomor Kendaraan Menggunakan Algoritma Ekstraksi Kontur Dan OCR*” dari penelitian tersebut dilakukan proses deteksi kontur yaitu suatu proses yang menghasilkan tepi-tepi dari objek citra, keberadaan tepi tersebut ditandai dengan tingginya perubahan nilai pixel dan proses pengenalan karakter menggunakan metode OCR dengan cara pembacaan tiap karakter (Setiadi, Himawan. 2012). Dari kedua penelitian tersebut hanya fokus pada pendeteksian nomor plat kendaraan tanpa dilanjutkan proses pada sistem perparkiran. Pada penelitian ini dilakukan secara *realtime* dan proses uji di mana setelah pengambilan gambar plat nomor akan di proses pengurangan dan penambahan pada pixel dan proses selanjutnya oleh untuk pembacaan karakter menggunakan metode OCR (*Optical Character Recognition*), setelah mendapatkan karakter plat nomor proses selanjutnya akan di aplikasikan pada sistem perparkiran yaitu di proses untuk pembukaan pintu masuk dan pintu keluar. Selain itu, pada penelitian ini juga dilakukan pengambilan waktu kendaraan selama parkir pada saat kendaraan masuk ataupun keluar dan akan dihasilkan jumlah pembayaran sesuai dengan lama waktunya kendaraan di parkir, selanjutnya penelitian ini juga akan disimpan sebagai *database* sebagai hasil plat nomor kendaraan yang telah melewati pintu masuk dan pintu keluar.

1.2 Rumusan Masalah

Mengacu pada permasalahan yang telah diuraikan pada latar belakang masalah, maka rumusan masalah dapat ditekankan pada:

1. Bagaimana cara analisa sistem pengenalan plat nomor kendaraan berdasarkan metode OCR (*Optical Character Recognition*) ?
2. Bagaimana cara menentukan posisi *webcam* yang tepat terhadap plat nomor yang sedang diamati dan akan ditampilkan ke komputer?
3. Bagaimana cara pengaplikasian plat nomor kendaraan pada sistem perparkiran ?

1.3 Batasan Masalah

Untuk menyederhanakan masalah maka dibuat batasan-batasan masalah sebagai berikut :

1. Bahasa pemrograman yang digunakan adalah visual basic.NET.
2. Bentuk fisik plat nomor sesuai dengan prosedur yang ada di negara Indonesia yaitu dengan latar belakang plat berwarna hitam dan nomor berwarna putih.
3. Plat nomor kendaraan yang digunakan adalah plat nomor mobil (kendaraan roda 4) yang berukuran 395 x 135 mm dan 430 x 135 mm.
4. Pembacaan karakter pada plat nomor kendaraan berhuruf kapital.
5. Kesalahan pembacaan karakter plat nomor kendaraan yaitu Q untuk 0 dan L untuk 1
6. Latar belakang setelah plat nomor berwarna merah.
7. Sistem yang dibuat pada penelitian akan bekerja secara *realtime*.
8. Posisi *webcam* pada palang untuk pembacaan plat nomor harus tetap dan tepat.
9. Mikrokontroler yang digunakan pada sistem perparkiran yaitu arduino.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang diuraikan di atas, penelitian ini dilakukan untuk :

1. Menganalisa sistem pengenalan plat nomor kendaraan berdasarkan metode OCR (*Optical Character Recognition*).
2. Menentukan posisi *webcam* yang tetap dan tepat terhadap plat nomor yang sedang diamati dan akan ditampilkan ke komputer.
3. Mengetahui cara pengaplikasian plat nomor kendaraan pada sistem perpajakan

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari alat hasil penelitian ini yaitu pendeteksi plat nomor kendaraan yang akan diaplikasikan pada sistem perpajakan.

BAB 2. TINJAUAN PUSTAKA

2.1 Plat Nomor Kendaraan Bermotor di Indonesia

Tanda Nomor Kendaraan Bermotor (TNKB), atau seringkali disebut plat nomor, adalah plat aluminium tanda kendaraan bermotor di Indonesia yang telah didaftarkan pada kantor bersama samsat.



Gambar 2.1 Plat Nomor Kendaraan Bermotor di Indonesia

(Sumber : http://id.wikipedia.org/wiki/Tanda_Nomor_Kendaraan_Bermotor)

2.1.1 Sejarah Penggunaan Plat Nomor Kendaraan Bermotor di Indonesia

Penggunaan tanda nomor kendaraan bermotor di Indonesia, terutama di Jawa, merupakan warisan sejak zaman Hindia Belanda, yang menggunakan kode wilayah berdasarkan pembagian wilayah karesidenan.

2.1.2 Spesifikasi Teknis

Tanda Nomor Kendaraan Bermotor berbentuk plat aluminium dengan cetakan tulisan dua baris. Baris pertama menunjukkan: kode wilayah (huruf), nomor polisi (angka), dan kode/seri akhir wilayah (huruf) dan baris kedua menunjukkan bulan dan tahun masa berlaku.

Bahan baku TNKB adalah aluminium dengan ketebalan 1 mm. Ukuran TNKB untuk kendaraan bermotor roda 2 dan roda 3 adalah 250x105 mm, sedangkan untuk kendaraan bermotor roda 4 atau lebih adalah 395x135 mm. Terdapat cetakan garis lurus pembatas lebar 5 mm diantara ruang nomor polisi dengan ruang angka masa berlaku. Pada sudut kanan atas dan sudut kiri bawah

terdapat tanda khusus (*security ark*) cetakan lambang Polisi Lalu Lintas; sedangkan pada sisi sebelah kanan dan sisi sebelah kiri ada tanda khusus cetakan "DITLANTAS POLRI" yang merupakan hak paten pembuatan TNKB oleh Polri.

Pada pertengahan tahun 2014 terjadi perubahan tampilan. Plat nomor kini sedikit diperpanjang dari ukuran semula (untuk roda empat). Selain itu, terdapat perubahan posisi lambang Polantas dan tulisan "Korlantas Polri", yakni, lambang Polantas kini berada di sudut kiri atas dan kanan bawah, sedangkan tulisan "Korlantas Polri" berada pada sudut kiri bawah dan kanan atas. Ukuran TNKB untuk kendaraan roda 2 dan 3 sekarang menjadi 275 mm dengan lebar 110 mm, sedangkan untuk kendaraan roda 4 atau lebih adalah panjang 430 mm dengan lebar 135 mm. Sementara ini, plat resmi yang lama masih berlaku (apalagi terkadang sejumlah Samsat di berbagai daerah sering memanfaatkan plat jenis lama untuk kendaraan yang plat nomornya diperpanjang setelah tahun 2011). Selain itu, pada spesifikasi teknis baru ini plat nomor menggunakan rupa huruf (*font*) yang sama.

2.1.3 Warna

Warna TNKB (tanda nomor kendaraan bermotor) ditetapkan sebagai berikut:

- Kendaraan bermotor perseorangan dan sewa: warna dasar hitam dengan tulisan berwarna putih.
- Kendaraan bermotor umum: warna dasar kuning dengan tulisan berwarna hitam.
- Kendaraan bermotor milik pemerintah: warna dasar merah dengan tulisan berwarna putih.
- Kendaraan bermotor korps diplomatik negara asing: warna dasar putih/merah dengan tulisan berwarna hitam.
- Kendaraan bermotor staf operasional korps diplomatik negara asing: warna dasar hitam dengan tulisan berwarna putih serta terdiri dari lima angka dan kode angka negara yang dicetak lebih kecil dengan format sub-bagian.

- Kendaraan bermotor di kawasan perdagangan bebas (*Free Trade Zone*) yang mendapatkan fasilitas pembebasan bea masuk (berdasarkan Peraturan Menteri Keuangan, kendaraan bermotor ini tidak boleh dioperasikan/dimutasikan ke wilayah Indonesia lainnya): warna dasar hijau dengan tulisan hitam.
- Kendaraan tidak bermotor di Surabaya: warna dasar biru dengan tulisan putih.

2.2 Webcam

Webcam atau *web camera* adalah sebuah kamera *video* digital kecil yang dihubungkan ke komputer melalui *port USB* ataupun *port COM*. Sebuah *web camera* yang sederhana terdiri dari sebuah lensa standar, dipasang di sebuah papan sirkuit untuk menangkap sinyal gambar, *casing (cover)*, termasuk *casing* depan dan *casing samping* untuk menutupi lensa standar dan memiliki sebuah lubang lensa di *casing* depan yang berguna untuk memasukkan gambar, kabel support, yang dibuat dari bahan yang fleksibel, salah satu ujungnya dihubungkan dengan papan sirkuit dan ujung satu lagi memiliki penghubung, kabel ini dikontrol untuk menyesuaikan ketinggian, arah dan sudut pandang *web camera*. *Web camera* memiliki fitur-fitur dan setting yang bermacam-macam, diantaranya adalah:

- 1) *Motion sensing* – *web camera* akan mengambil gambar ketika kamera mendeteksi gerakan.
- 2) *Image archiving* – pengguna dapat membuat sebuah *archive* yang menyimpan semua gambar dari *web camera* atau hanya gambar-gambar tertentu saat *interval pre-set*.
- 3) *Video messaging* – beberapa program messaging mendukung fitur ini.
- 4) *Advanced connections* – menyambungkan perangkat *home theater* ke *web camera* dengan kabel maupun nirkabel.
- 5) *Automotion* – kamera robotik yang memungkinkan pengambilan gambar secara *pan* atau *tilt* dan *setting* program pengambilan *frame* berdasarkan posisi kamera.

- 6) *Streaming media* – aplikasi profesional, setup *web camera* dapat menggunakan kompresi MPEG4 untuk mendapatkan *streaming audio* dan *video* yang sesungguhnya.
- 7) *Custom coding* – mengimport kode komputer pengguna untuk memberitahu *web camera* apa yang harus dilakukan (misalnya *automatically refresh*).
- 8) *AutoCam* – memungkinkan pengguna membuat *web page* untuk *web camera* secara gratis di *server* perusahaan pembuat *web camera*.

Ada beberapa permasalahan yang dihadapi *web camera*. Secara fisik, kamera-kamera yang beredar di pasaran memiliki kesulitan untuk memenuhi kebutuhan personal pengguna karena desainnya yang cukup bergaya namun hanya memiliki sedikit variasi. Lalu, sudut pandang *web camera* disesuaikan tidak langsung dalam cara yang tidak nyaman. Dan juga pengguna banyak menemui kesulitan ketika menyesuaikan posisi *web camera* untuk menangkap gambar (Riki Setian, 2014).



Gambar 2.2 *Webcam*.
(Sumber: Riki Setian, 2014)

2.3 *Image Processing*

2.3.1 *Pengertian Image Processing*

Citra dapat berbentuk foto hitam putih atau berwarna, sinyal-sinyal *video* seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu pita magnetik. Menurut presisi yang digunakan untuk menyatakan titik-titik koordinat pada ranah waktu atau bidang dan untuk menyatakan nilai keabuan atau warna suatu citra, maka secara teoritis citra dapat dikelompokkan menjadi empat kelas citra, yaitu cara kontinu-kontinu, kontinu-diskret, diskret-kontinu dan diskret-diskret.

Dengan label pertama menyatakan presisi dari titik-titik koordinat pada bidang citra sedangkan label kedua menyatakan presisi nilai keabuan atau warna. Kontinu dinyatakan dengan presisi tak hingga, sedangkan diskret dinyatakan dengan presisi angka berhingga. Pengubahan citra yang bersifat kontinu menjadi citra yang bersifat diskret memerlukan pembuatan kisi-kisi arah vertikal dan horisontal, sehingga diperoleh citra dalam bentuk larik dua dimensi. Proses tersebut dikenal sebagai proses digitisasi atau pencuplikan (*sampling*). Setiap elemen larik tersebut dikenal sebagai elemen gambar atau piksel.

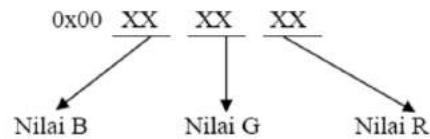
2.3.2 Representasi Citra Digital

Citra didefinisikan sebagai suatu fungsi dua dimensi $f(x, y)$, dengan x dan y merupakan koordinat spasial, dan f disebut sebagai kuantitas bilangan skalar positif yang memiliki maksud secara fisik ditentukan oleh sumber citra. Suatu citra digital yang diasumsikan dengan fungsi $f(x, y)$ direpresentasikan dalam suatu fungsi koordinat berukuran $M \times N$. Variabel M adalah baris dan N adalah kolom sebagaimana ditunjukkan pada Gambar 2.7. Setiap elemen dari *array* matriks disebut *image element*, *picture element*, atau *pixel* (Gonzales et al. 2003).

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \dots & \dots & \dots & \dots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \dots\dots\dots(2.1)$$

2.4 Proses pengolahan citra warna

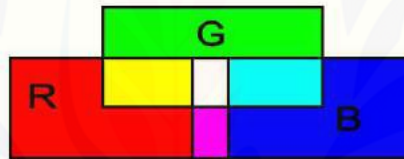
Dasar dari pengolahan citra adalah pengolahan warna RGB pada posisi tertentu. Dalam pengolahan citra warna dipresentasikan dengan nilai hexadesimal dari 0x00000000 sampai 0x00ffffff. Warna hitam adalah 0x00000000 dan warna putih adalah 0x00ffffff. Variabel 0x00 menyatakan angka dibelakangnya adalah *hexadecimal*.



Gambar 2.3 Nilai warna RGB dalam *hexadecimal*.

(Sumber: Riki Setian, 2014)

Gambar 2.2 Nilai warna RGB dalam hexadesimal Terlihat bahwa setiap warna mempunyai range nilai 00 (angka desimalnya adalah 0) dan ff (angka desimalnya adalah 255), atau mempunyai nilai derajat keabuan $256 = 2^8$. Dengan demikian *range* warna yang digunakan adalah $(2^8)(2^8)(2^8) = 2^{24}$ (atau yang dikenal dengan istilah *True Colour* pada Windows). Nilai warna yang digunakan di atas merupakan gabungan warna cahaya merah, hijau dan biru seperti yang terlihat pada gambar 2.2. Sehingga untuk menentukan nilai dari suatu warna yang bukan warna dasar digunakan gabungan skala kecerahan dari setiap warnanya.



Gambar 2.4 Komposisi warna RGB.

(Sumber: Riki Setian, 2014)

Dari definisi diatas untuk menyajikan warna tertentu dapat dengan mudah dilakukan, yaitu dengan mencampurkan ketiga warna dasar RGB. Pengolahan citra dimulai dengan proses *thresholding*, yaitu proses pemisahan citra berdasarkan batas nilai tertentu, dalam proses *thresholding* citra warna diubah menjadi citra biner. Tujuan proses *thresholding* adalah untuk membedakan objek dengan latar belakangnya. Setelah proses *thresholding* proses selanjutnya adalah proses penghitungan nilai-nilai parameter antara lain R, G, B, RGB rata-rata (*color value*), dan indeks R (*Ired*), indeks G (*Igreen*), indeks B (*Iblue*), dari tiap-tiap *pixel*

a. Pengukuran Parameter RGB (*Red*, *Green* dan *Blue*)

Paramater RGB diperoleh dari tiap-tiap *pixel* warna pada citra yang merupakannilai intensitas untuk masing-masing warna merah, hijau, dan biru.

Nilai rata-rata dari R, G dan B dijumlahkan untuk mendapatkan *color value* atau RGB rata-rata.

b. Pengukuran parameter Indeks R, Indeks G dan Indeks B

Perhitungan indeks warna merah/indeks R (*Ired*), indeks warna hijau/indeks G (*Igreen*), dan indeks warna biru/indeks B (*Iblue*) menggunakan rumus pada persamaan (1), (2) dan (3). Intensitas warna merah dibagi dengan penjumlahan dari nilai intensitas warna merah, hijau, dan biru sehingga menghasilkan nilai parameter indeks R. Intensitas warna hijau dibagi dengan penjumlahan dari nilai intensitas warna merah, hijau, dan biru sehingga menghasilkan nilai parameter indeks G. Intensitas warna biru dibagi dengan penjumlahan dari nilai intensitas warna merah, hijau, dan biru sehingga menghasilkan nilai parameter indeks B. Perhitungan parameter Indeks R, G, dan B diperoleh dari tiap-tiap *pixel* pada citra tepi.

Model warna RGB dapat juga dinyatakan dalam bentuk indeks warna RGB dengan rumus sebagai berikut (Ahmad, 2005; Arimurthy, dkk., 1992):

Indeks warna merah:

$$(Ired) = \frac{R}{R+G+B} \dots\dots\dots(2.2)$$

Indeks warna hijau:

$$(Igreen) = \frac{G}{G+B+R} \dots\dots\dots(2.3)$$

Indeks warna biru:

$$(Iblue) = \frac{B}{B+G+R} \dots\dots\dots(2.4)$$

Dimana : R = *Red* (Merah)
G = *Green*(Hijau)
B = *Blue* (Biru)

Dengan R, G dan B masing-masing merupakan besaran yang menyatakan nilai intensitas warna merah, hijau, dan biru.

2.5 Citra *Grayscale* (Keabuan)

Citra grayscale merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, artinya nilai dari Red = Green = Blue. Nilai-nilai tersebut digunakan untuk menunjukkan intensitas warna.

Proses awal yang banyak dilakukan dalam *image processing* adalah mengubah citra berwarna menjadi citra *grayscale*, hal ini digunakan untuk menyederhanakan model citra. Seperti telah dijelaskan didepan, citra berwarna terdiri dari 3 layer matrik yaitu *R-layer*, *G-layer* dan *B-layer*. Sehingga untuk melakukan proses-proses selanjutnya tetap diperhatikan tiga layer di atas. Bila setiap proses perhitungan dilakukan menggunakan tiga layer, berarti dilakukan tiga perhitungan yang sama. Sehingga konsep itu diubah dengan mengubah 3 layer di atas menjadi 1 layer matrik *grayscale* dan hasilnya adalah citra *grayscale*. Dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan.

Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing *r*, *g* dan *b* menjadi citra *grayscale* dengan nilai *s*, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai *r*, *g* dan *b* sehingga dapat dituliskan menjadi:

$$S = \frac{R+G+B}{3} \dots\dots\dots(2.5)$$

Dimana :
r = red
g = green
b = blue
s = rata- rata



Gambar 2.5 Contoh konversi citra ke *Gray*.

(Sumber: Riki Setian, 2014)

2.6 Thresholding

Thresholding digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. Dengan menggunakan *thresholding* maka derajat keabuan bisa diubah sesuai keinginan, misalkan diinginkan menggunakan derajat keabuan 2, maka tinggal membagi nilai derajat keabuan dengan 2 (Riki Setian, 2014). Proses *thresholding* ini pada dasarnya adalah proses pengubahan kuantisasi pada citra, sehingga untuk melakukan *thresholding* dengan derajat keabuan dapat digunakan rumus:

$$X = b.int(w.b) \dots \dots \dots (2.6)$$

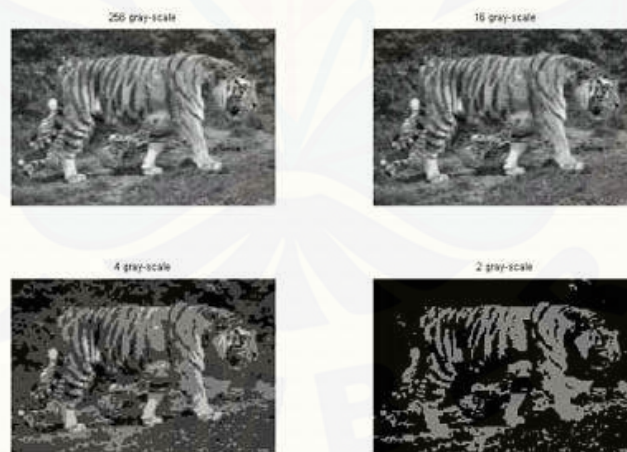
dimana :

w adalah nilai derajat keabuan sebelum *thresholding*

x adalah nilai derajat keabuan setelah *thresholding*

$$b = int(256/a) \dots \dots \dots (2.7)$$

Berikut ini contoh *thresholding* mulai di 256, 16, 4 dan 2.

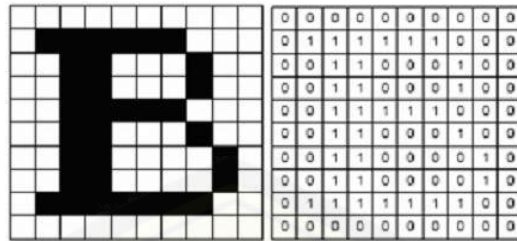


Gambar 2.6 Contoh Derajat Kualitas *Thresholding*.

(Sumber: Riki Setian, 2014)

2.7 Citra Biner

Citra biner, yaitu citra yang hanya terdiri atas dua warna, yaitu hitam dan putih. Oleh karena itu, setiap *pixel* pada citra biner cukup direpresentasikan dengan 1 bit.



Gambar 2.7 Citra Biner Dan Representasi Citra Binernya.

(Sumber: Riki Setian, 2014)

Meskipun saat ini citra berwarna lebih disukai karena memberi kesan yang lebih kaya dari citra biner, namun tidak membuat citra biner mati. Pada beberapa aplikasi citra biner masih tetap dibutuhkan, misalkan citra logo instansi (yang hanya terdiri dari warna hitam dan putih), citra kode barang (*barcode*) yang tertera pada label barang, citra hasil pemindaian dokumen teks, dan sebagainya. Seperti yang sudah disebutkan diatas, citra biner hanya mempunyai dua nilai derajat keabuan: hitam dan putih. *Pixel – pixel* objek bernilai 1 dan *pixel – pixel* latar belakang bernilai 0. Pada waktu menampilkan gambar, 0 adalah putih dan 1 adalah hitam. Jadi pada citra biner, latar belakang berwarna putih sedangkan objek berwarna hitam seperti tampak pada gambar 2.3 diatas. Meskipun komputer saat ini dapat memproses citra hitam-putih (*grayscale*) maupun citra berwarna, namun citra biner masih tetap di pertahankan keberadaannya.

Alasan penggunaan citra biner adalah karena citra biner memiliki sejumlah keuntungan sebagai berikut:

- 1) Kebutuhan memori kecil karena nilai derajat keabuan hanya membutuhkan representasi 1 bit.
- 2) Waktu pemrosesan lebih cepat dibandingkan dengan citra hitam-putih ataupun warna. (Marvin Chandra Wijaya,2007)

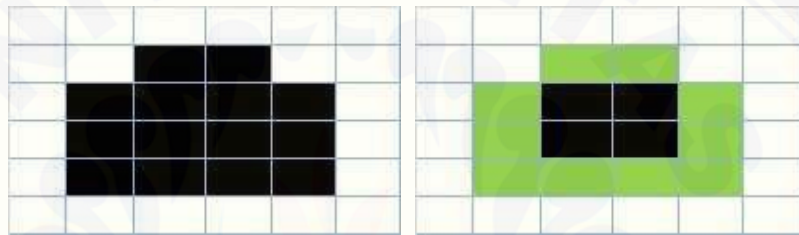
2.8 Erosion

Erosi merupakan operasi morfologi yang akan memperkecil pixel pada batas antar objek dalam suatu citra digital. Apabila erosi dilakukan

maka yang dikerjakan adalah memindahkan piksel pada batasan-batasan objek yang akan di erosi. Jumlah dari piksel yang ditambah atau dihilangkan bergantung pada ukuran dan bentuk dari *structuring element* yang digunakan untuk memproses image tersebut.

Contoh :

Di bawah ini merupakan hasil image setelah dilakukan proses Erosi dari Original Image dengan Stucturing Element yang telah ditentukan di atas. Warna hijau merupakan pixel dihilangkan setelah dilakukan proses Erosi, sehingga pixel hasilnya hanya yang berwarna hitam.



Gambar 2.8 Contoh Hasil Gambar Yang Di Proses Menggunakan Operasi Morphologi Erosi

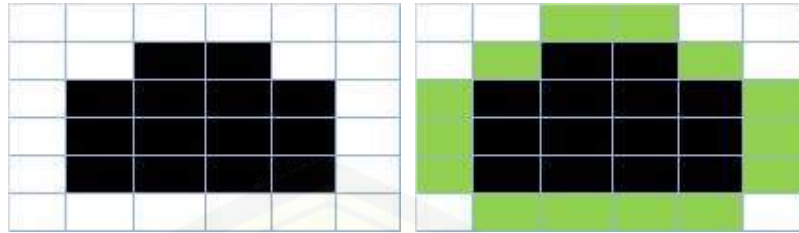
(Sumber: <https://mochamadyagi.wordpress.com/2012/04/17/pengolahan-citra-tugas-4/>)

2.9 Dilation

Dilasi merupakan operasi morfologi yang akan memperbesar pixel pada batas antar objek dalam suatu citra digital. Secara rinci Dilasi merupakan suatu proses menambahkan piksel pada batasan dari objek dalam suatu image sehingga nantinya apabila dilakukan operasi ini maka image hasilnya lebih besar ukurannya dibandingkan dengan image aslinya.. Jumlah dari piksel yang ditambah atau dihilangkan bergantung pada ukuran dan bentuk dari *structuring element* yang digunakan untuk memproses image tersebut.

Contoh :

Di bawah ini merupakan hasil image setelah dilakukan proses Dilasi dari Original Image dengan Stucturing Element yang telah ditentukan di atas. Warna hijau merupakan pixel tambahan setelah dilakukan Dilasi



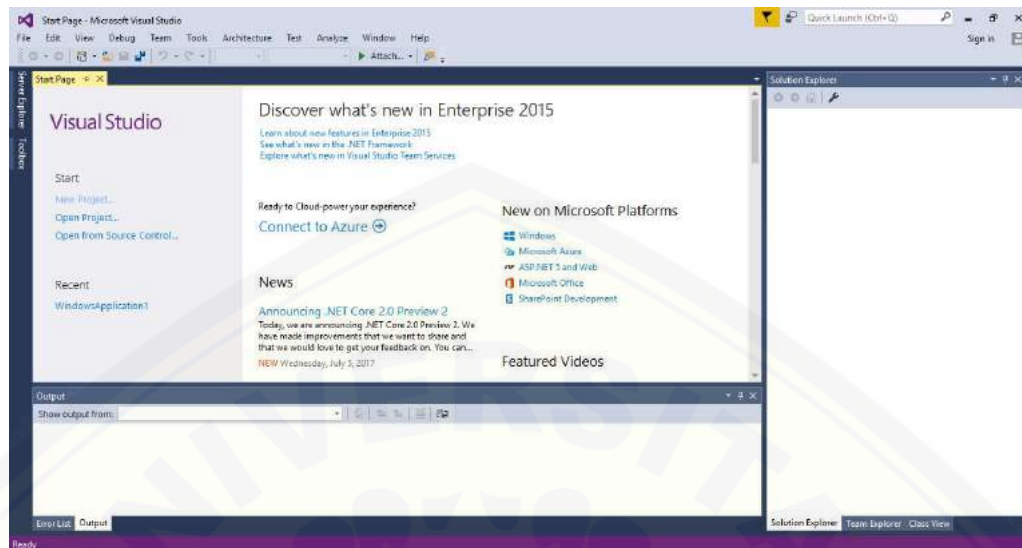
Gambar 2.9 Contoh Hasil Gambar Yang Di Proses Menggunakan Operasi Morphologi Dilasi

(Sumber: <https://mochamadyagi.wordpress.com/2012/04/17/pengolahan-citra-tugas-4/>)

2.10 Software Visual Studio 2015

Microsoft Visual Studio 2015 merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment (IDE)*, dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi *Windows Mobile* (yang berjalan di atas .NET Compact Framework). Visual studio yang digunakan tugas akhir ini adalah visual studio 2015 yang mendukung *library up to date computer vision* dalam proses pengolahan citra. Microsoft Visual Studio C# 2015 Ultimate dipilih karena dapat digunakan dan diintegrasikan dengan *library* OpenCV terbaru.



Gambar 2.10 Tampilan dinding kerja Microsoft Visual Studio 2015 Enterprise.

2.11 EmguCV

OpenCV merupakan *library* yang paling terkenal dan paling banyak digunakan hingga saat ini. OpenCV dikembangkan oleh Intel. Mulanya ditulis dalam bahasa C dan sekarang juga menyertakan *interface* C++ sejak versi 2.0. Tentunya *library* ini secara *default* hanya bisa digunakan oleh program yang ditulis dengan C atau C++.

EmguCV adalah *wrapper* .Net untuk OpenCV. Dengan EmguCV, fungsi-fungsi dalam OpenCV bisa dipanggil melalui bahasa pemrograman yang *compatible* dengan .NET seperti C#, VB, dan VC++. Keuntungan menggunakan EmguCV yang paling utama adalah *library* ini sepenuhnya ditulis dengan C# yang mana tentunya lebih aman karena pembuatan *object* ataupun *reference* diatur oleh *garbage collector*. Selain itu, EmguCV juga *cross platform* sehingga dapat di *compile* lewat Mono dan dijalankan di atas sistem operasi Linux atau Mac OS.

Dengan EmguCV, kita dapat membuat aplikasi apapun seperti layaknya menggunakan OpenCV. Namun paling tidak ada dua konsep penting yang perlu diketahui terlebih dahulu sebelum menggunakan Emgu CV. Konsep yang pertama mengenai layer pada EmguCV, EmguCV terdiri dari 2 layer yaitu *basic layer* dan *second layer*. *Basic layer* mengandung fungsi, struktur, dan enumerasi yang

secara langsung merefleksikan apa yang ada di OpenCV. Dengan adanya layer inilah kita bisa memanggil fungsi-fungsi pada OpenCV dengan bahasa pemrograman C#. Sedangkan *second layer* mengandung kelas-kelas yang memanfaatkan keunggulan teknologi .NET.

2.12 Optical Character Recognition (OCR)

Manusia mengenali objek-objek yang ada di sekelilingnya, dengan cara ini mata menerapkan mekanisme optik. Namun sementara otak melihat input, kemampuan untuk memahami sinyal-sinyal ini bervariasi pada setiap orang sesuai dengan banyak faktor.

Begitu halnya dengan OCR (*Optical Character Recognition*). Teknologi ini memungkinkan mesin untuk secara otomatis mengenali karakter melalui mekanisme optik. (AIM.Inc, 2000) *AIM. Inc* dalam artikel berjudul *Optical Character Recognition* pada tahun 2000 juga menjelaskan, OCR banyak

Digunakan untuk mengkonversi buku-buku dan dokumen ke dalam bentuk file elektronik, misalnya dalam bentuk PDF dan lain sebagainya, untuk mengkomputerisasi sistem pencatatan misalnya di kantor, atau untuk mempublikasikan teks pada website.

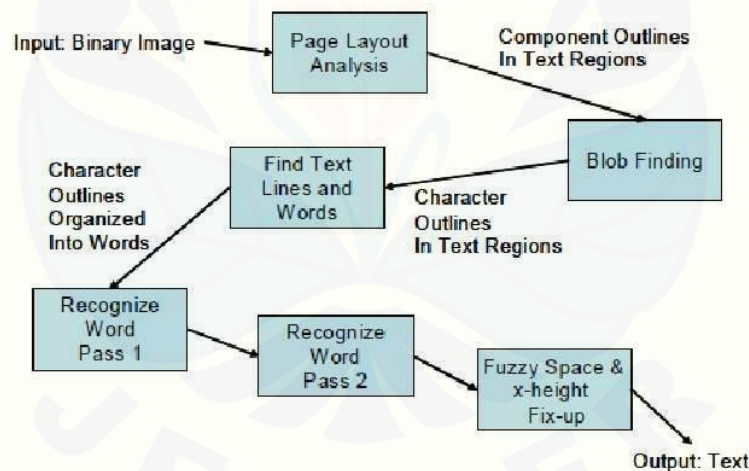
OCR memungkinkan untuk mengedit teks, mencari kata atau frase, dan menerapkan teknik seperti mesin penerjemahan, *text-to-speech* dan *text mining*. OCR biasa digunakan untuk bidang penelitian dalam pengenalan pola, kecerdasan buatan (*artificial intelligent*) dan *computer vision*. Sistem OCR memerlukan kalibrasi untuk membaca font yang spesifik, versi awal harus diprogram dengan gambar karakter masing-masing, dan bekerja pada satu font pada suatu waktu. Sistem cerdas (*intelligent system*) dengan tingkat akurasi yang tinggi pengakuan untuk font yang paling sekarang umum. Beberapa sistem mampu mereproduksi output diformat yang erat mendekati halaman yang dipindai asli termasuk gambar, kolom dan non-tekstual komponen. (Nelson, 2007)

2.13 Tesseract OCR

Tesseract adalah *engine open source* OCR yang awalnya dikembangkan HP (*Hewlett-Packard*) antara tahun 1984 dan 1994. *Tesseract* dimulai dari sebuah proyek penelitian PhD di HP Laboratorium, Bristol oleh Ray Smith. Setelah penelitian bersama antara HP Labs Bristol dan Scanner HP divisi di Colorado, *tesseract* secara signifikan memimpin dalam akurasi atas mesin komersial tetapi tidak menjadi produk. (Ray Smith, 2005, p1)

Tahap perkembangan berikutnya kembali di HP Lab Bristol sebagai investigasi OCR untuk kompresi. Pada akhir proyek ini, pada tahun 1994, pengembangan berhenti sepenuhnya. Mesin ini dikirim ke UNLV (University Nevada Las Vegas) pada tahun 1995 menjalani tes akurasi tahunan OCR. (Ray Smith, 2005)

2.13.1 Artitektur *Tesseract*



Gambar 2.11 Arsitektur *tesseract*
(Sumber: Himawan Setiadi, 2011)

Tesseract OCR mengasumsikan input yang diterima berupa sebuah *binary image*. Pertama, analisis dilakukan pada komponen terhubung/*Connected Component (CC)* untuk menemukan di mana *outline* komponen disimpan. Pada tahap ini *outlines* dikumpulkan bersama menjadi *blob*. *Blob* disusun menjadi baris teks, sedangkan garis dan *region* dianalisis untuk *pitch* tetap dan teks *proporsional*. Baris teks dipecah menjadi kata-kata berbeda menurut jenis spasi

karakter. Teks dengan *pitch* tetap dibagi menjadi sel-sel karakter. Teks proporsional dipecah menjadi kata-kata dengan menggunakan ruang pasti dan ruang *fuzzy*. Pengenalan kata pada *image* dilakukan pada dua tahap proses yang disebut *pass-two* (Smith, 2009).

Pada *pass* pertama dilakukan untuk mengenali masing-masing kata pada gilirannya. Kata-kata yang sukses pada *pass* pertama yaitu kata-kata yang terdapat di kamus dan tidak ambigu kemudian diteruskan ke *adaptive classifier* sebagai data pelatihan. Begitu *adaptive classifier* memiliki sampel yang cukup, *adaptive classifier* ini dapat memberikan hasil klasifikasi bahkan pada *pass* pertama. Proses *pass* kedua dilakukan untuk mengenali kata-kata yang mungkin saja kurang dikenali atau terlewatkan pada *pass* pertama, pada tahap ini *adaptive classifier* telah memperoleh informasi lebih dari *pass* pertama. Tahap terakhir menyelesaikan ruang *fuzzy* dan memeriksa hipotesis alternatif pada ketinggian-x untuk mencari teks dengan *smallcap*. *Tesseract* dirancang untuk mengenali teks putih di atas latar hitam dan teks hitam di atas latar putih. Hal ini menyebabkan rancangan mengarah pada analisis komponen terhubung/*connected component* (CC) dan operasi pada *outline* komponen. Langkah pertama setelah analisis CC ialah menemukan *blob* pada *region* teks. Sebuah *blob* merupakan unit putatif yang dapat diklasifikasikan, yang mana bisa satu atau lebih komponen-komponen yang saling tumpang tindih secara horizontal.

2.14 Arduino Uno



Gambar 2.12 Arduino UNO

(Sumber: <http://aozon.blogspot.co.id/2014/03/mengenal-arduino-uno-lebih-rinci.html>)

Arduino Uno adalah board mikrokontroler berbasis ATmega328. Uno memiliki 14 pin digital input / output (dimana 6 dapat digunakan sebagai output PWM), 6 input analog, resonator keramik 16 MHz, koneksi USB, jack listrik, header ICSP, dan tombol reset. Uno dibangun berdasarkan apa yang diperlukan untuk mendukung mikrokontroler, sumber daya bisa menggunakan power USB (jika terhubung ke komputer dengan kabel USB) dan juga dengan adaptor atau baterai. Arduino Uno berbeda dari semua papan sebelumnya dalam hal tidak menggunakan FTDI chip driver USB-to-serial. Sebaliknya, fitur Atmega16U2 (Atmega8U2 sampai versi R2) diprogram sebagai konverter USB-to-serial. Revisi 2 dari Uno memiliki resistor pulling 8U2 HWB yang terhubung ke tanah, sehingga lebih mudah untuk menggunakan mode DFU.

Board Arduino Uno memiliki fitur – fitur baru sebagai berikut :

- Pinout : menambahkan SDA dan SCL pin yang dekat ke pin aref dan dua pin baru lainnya ditempatkan dekat ke pin RESET, dengan I/O REF yang memungkinkan sebagai buffer untuk beradaptasi dengan tegangan yang disediakan dari board sistem. Pengembangannya, sistem akan lebih kompatibel dengan prosesor yang menggunakan AVR, yang beroperasi dengan 5V dan dengan Arduino karena beroperasi dengan 3,3V. Yang kedua adalah pin yang tidak terhubung, yang disediakan untuk tujuan pengembangannya.
- Sirkuit reset
- ATmega 16U2 ganti 8U yang digunakan sebagai konverter USB-to-serial.

Mikrokontroler	ATMega 328
<i>Operating Voltage</i>	5V
<i>Input Voltage (recommended)</i>	7 – 12V
<i>Input Voltage (limit)</i>	6 – 20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

Gambar 2.13 Deskripsi Arduino Uno R3

(Sumber: <http://aozon.blogspot.co.id/2014/03/mengenal-arduino-uno-lebih-rinci.html>)

2.14.1 Catu Daya

Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber listrik dipilih secara otomatis. Eksternal (non-USB) daya dapat berupa baik AC-DC adaptor atau baterai. Adaptor ini dapat dihubungkan dengan cara menghubungkan plug pusat – positif 2.1mm ke dalam board colokan listrik. Sedangkan untuk baterai dapat dihubungkan kedalam header pin GND dan Vin dari konektor Power.

Board dapat beroperasi pada pasokan daya dari 6 – 20 volt. Jika diberikan dengan kurang dari 7V, bagaimanapun, pin 5V dapat menyeluplai kurang dari 5 volt dan board mungkin tidak stabil. Jika menggunakan lebih dari 12V, regulator bisa panas dan merusak board. Rentang yang dianjurkan adalah 7V – 12V. Selain itu, beberapa pin memiliki fungsi khusus :

- Serial : 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirim (TX) data TTL serial. Pin ini terhubung ke pin yang sesuai dari chip ATmega8U2 USB-to-Serial TTL.
- Eksternal Interupsi : 2 dan 3. Pin ini dapat dapat dikonfigurasi untuk memicu interupsi pada nilai yang rendah, tepi naik atau jatuh, atau perubahan nilai. Lihat `attachInterrupt()` fungsi untuk rincian.
- PWM : 3,5,6,9,10, dan 11. Menyediakan 8-bit output PWM dengan `analogWrite()` fungsi.
- SPI : 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan library SPI.
- LED : 13. Ada built-in LED terhubung ke pin digital 13. Ketika pin adalah nilai TINGGI. LED menyala, ketika pin adalah RENDAH, itu off.

Arduino Uno R3 memiliki 6 input analog diberi label A0 sampai A5, masing – masing menyediakan 10-bit resolusi (yaitu 1024 nilai yang berbeda). Secara default sistem mengukur dari ground sampai 5 volt, meskipun mungkin untuk mengubah ujung atas rentang menggunakan pin AREF dan fungsi `analogReference()`. Selain itu, beberapa pin memiliki fungsi khusus :

- TWI : A4 atau SDA pin dan A5 atau SCL pin. Mendukung komunikasi TWI
- AREF : Referensi tegangan untuk input analog. Digunakan dengan `analogReference()`
- RESET

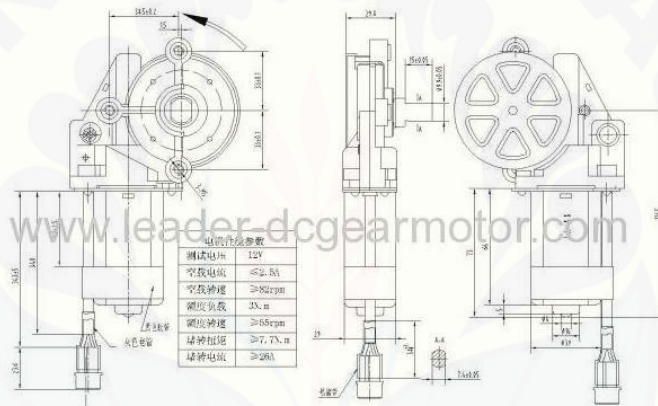
2.15 Motor DC

Motor DC adalah motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Kumparan medan pada motor DC disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Motor arus searah, sebagaimana namanya, menggunakan arus langsung yang tidak langsung atau *direct-unidirectional*.

2.15.1 Motor Power Windows

Motor *power windows* merupakan motor dc yang cara kerjanya yaitu dinamo akan berputar bila menerima arus listrik. Lalu, arus listrik tersebut diubah menjadi tenaga mekanis melalui gigi, motor *power windows* dapat berputar searah jarum jam maupun sebaliknya. Spesifikasi untuk motor *power windows* sebagai berikut :

1. Tegangan 12 volt
2. Arus max ≤ 3.5
3. Speed max 90 – 110 Rpm
4. Torque max ≤ 8



Gambar 2.14 Spesifikasi Motor *Power Windows*.

(Sumber: leader-dcgearmotor.com)

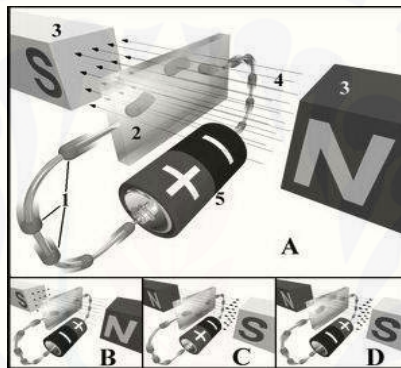


Gambar 2.15 Motor *Power Windows*.

(Sumber: leader-dcgearmotor.com)

2.16 Sensor *Hall Effect*

Sensor *Hall Effect* digunakan untuk mendeteksi kedekatan (*proximity*), kehadiran atau ketidakhadiran suatu objek magnetis (yang) menggunakan suatu jarak kritis. Pada dasarnya ada dua tipe Half-Effect Sensor, yaitu tipe linear dan tipe on-off. Tipe linear digunakan untuk mengukur medan magnet secara linear, mengukur arus DC dan AC pada konduktordan fungsi-fungsi lainnya. Sedangkan tipe on-off digunakan sebagai limit switch, sensor keberadaan (*presence sensors*), dsb. Sensor ini memberikan logika output sebagai interface gerbang logika secara langsung atau mengendalikan beban dengan *buffer amplifier*.



Gambar 2.16 Cara Kerja Sensor *Hall Effect*.

(Sumber: <http://elektronika-dasar.web.id/definisi-dan-fungsi-sensor-efek-hall/>)

Keterangan gambar :

1. Elektron
2. Sensor Hall atau Elemen Hall
3. Magnet
4. Medan Magnet
5. *Power Source*

Gambar diagram *hall effect* tersebut menunjukkan aliran elektron. Dalam gambar A menunjukkan bahwa elemen Hall mengambil kutub negatif pada sisi atas dan kutub positif pada sisi bawah. Dalam gambar B dan C, baik arus listrik ataupun medan magnet dibalik, menyebabkan polarisasi juga terbalik. Arus dan medan magnet yang dibalik ini menyebabkan sensor Hall mempunyai kutub negatif pada sisi atas.

Hall Effect tergantung pada beda potensial (tegangan Hall) pada sisi yang berlawanan dari sebuah lembar tipis material konduktor atau semikonduktor dimana arus listrik mengalir, dihasilkan oleh medan magnet yang tegak lurus dengan elemeh Hall. Perbandingan tegangan yang dihasilkan oleh jumlah arus dikenal dengan tahanan Hall, dan tergantung pada karakteristik bahan. Dr. Edwin Hall menemukan efek ini pada tahun 1879.

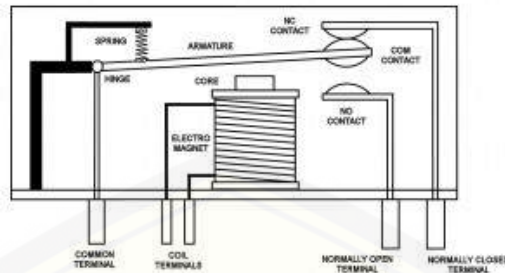
Hall Effect dihasilkan oleh arus pada konduktor. Arus terdiri atas banyak beban kecil yang membawa partikel-partikel (biasanya elektron) dan membawa gaya Lorentz pada medan magnet. Beberapa beban ini berakhir di sisi – sisi konduktor. Ini hanya berlaku pada konduktor besar dimana jarak antara dua sisi cukup besar.

Salah satu yang paling penting dari *Hall Effect* adalah perbedaan antara beban positif bergerak dalam satu arah dan beban negatif bergerak pada kebalikannya. *Hall Effect* memberikan bukti nyata bahwa arus listrik pada logam dibawa oleh elektron yang bergerak, bukan oleh proton. Yang cukup menarik, Hall Effect juga menunjukkan bahwa dalam beberapa substansi (terutama semikonduktor), lebih cocok bila kita berpikir arus sebagai “holes” positif yang bergerak daripada elektron.

2.17 Relay

Relay dibutuhkan dalam rangkaian elektronika sebagai eksekutor sekaligus interface antara beban dan sistem kendali elektronik yang berbeda sistem power supplynya. Secara fisik antara saklar atau kontaktor dengan elektromagnet relay terpisah sehingga antara beban dan sistem kontrol terpisah. Bagian utama relay elektro mekanik adalah sebagai berikut.

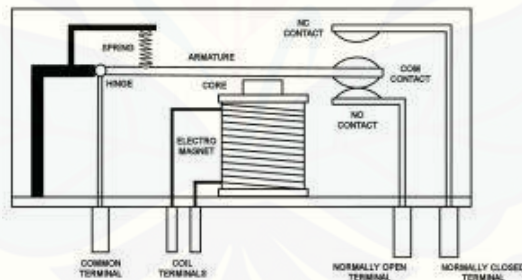
- Kumparan elektromagnet
- Saklar atau kontaktor
- Swing Armatur
- Spring (Pegas)



Gambar 2.17 Konstruksi Relai Elektro Mekanik Posisi NC (Normally Close)

(Sumber: <http://elektronika-dasar.web.id/teori-relay-elektro-mekanik/>)

Dari konstruksi relai elektro mekanik diatas dapat diuraikan sistem kerja atau proses relay bekerja. Pada saat elektromagnet tidak diberikan sumber tegangan maka tidak ada medan magnet yang menarik armature, sehingga saklar relay tetap terhubung ke terminal NC (Normally Close) seperti terlihat pada gambar konstruksi diatas. Kemudian pada saat elektromagnet diberikan sumber tegangan maka terdapat medan magnet yang menarik armature, sehingga saklar relay terhubung ke terminal NO (Normally Open) seperti terlihat pada gambar dibawah.



Gambar 2.18 Konstruksi Relai Elektro Mekanik Posisi NO (Normally Open)

(Sumber: <http://elektronika-dasar.web.id/teori-relay-elektro-mekanik/>)

Relay elektro mekanik memiliki kondisi saklar atau kontaktor dalam 3 posisi. Ketiga posisi saklar atau kontaktor relay ini akan berubah pada saat relay mendapat tegangan sumber pada elektromagnetnya. Ketiga posisi saklar relay tersebut adalah :

- **Posisi Normally Open (NO)**, yaitu posisi saklar relay yang terhubung ke terminal NO (Normally Open). Kondisi ini akan terjadi pada saat relay mendapat tegangan sumber pada elektromagnetnya.

- **Posisi Normally Close (NC)**, yaitu posisi saklar relay yang terhubung ke terminal NC (Normally Close). Kondisi ini terjadi pada saat relay tidak mendapat tegangan sumber pada elektromagnetnya.
- **Posisi Change Over (CO)**, yaitu kondisi perubahan armatur saklar relay yang berubah dari posisi NC ke NO atau sebaliknya dari NO ke NC. Kondisi ini terjadi saat sumber tegangan diberikan ke elektromagnet atau saat sumber tegangan diputus dari elektromagnet relay.

Relay yang ada dipasaran terdapat beberapa jenis sesuai dengan desain yang ditentukan oleh produsen relay. Dilihat dari desai saklar relay maka relay dibedakan menjadi :

- **Single Pole Single Throw (SPST)**, relay ini memiliki 4 terminal yaitu 2 terminal untuk input kumparan elektromagnet dan 2 terminal saklar. Relay ini hanya memiliki posisi NO (Normally Open) saja.
- **Single Pole Double Throw (SPDT)**, relay ini memiliki 5 terminal yaitu terdiri dari 2 terminal untuk input kumparan elektromagnetik dan 3 terminal saklar. relay jenis ini memiliki 2 kondisi NO dan NC.
- **Double Pole Single Throw (DPST)**, relay jenis ini memiliki 6 terminal yaitu terdiri dari 2 terminal untuk input kumparan elektromagnetik dan 4 terminal saklar untuk 2 saklar yang masing-masing saklar hanya memiliki kondisi NO saja.
- **Double Pole Double Throw (DPDT)**, relay jenis ini memiliki 8 terminal yang terdiri dari 2 terminal untuk kumparan elektromagnetik dan 6 terminal untuk 2 saklar dengan 2 kondisi NC dan NO untuk masing-masing saklarnya.

Relay dapat digunakan untuk mengontrol motor AC dengan rangkaian kontrol DC atau beban lain dengan sumber tegangan yang berbeda antara tegangan rangkaian kontrol dan tegangan beban. Diantara aplikasi relay yang dapat ditemui diantaranya adalah :

- Relay sebagai kontrol ON/OFF beban dengan sumber tegang berbeda.
- Relay sebagai selektor atau pemilih hubungan.
- Relay sebagai eksekutor rangkaian delay (tunda)

- Relay sebagai protektor atau pemutus arus pada kondisi tertentu.
Berikut ini memperlihatkan beberapa bentuk kontak dari sebuah *relay* :



Gambar 2.19 Relay

(Sumber: www.omron.com)

2.18 Sensor Ultrasonik HC-SR04

Sensor ultrasonik adalah sebuah sensor yang mengubah besaran \wedge isis(bunyi) menjadi besaran listrik. Pada sensor ini gelombang ultrasonik dibangkitkan melalui sebuah benda yang disebut *piezoelektrik*. *Piezoelektrik* ini akan menghasilkan gelombang ultrasonik dengan frekuensi 40 kHz ketika sebuah osilator diterapkan pada benda tersebut. Sensor ultrasonik secara umum digunakan untuk suatu pengungkapan tak sentuh yang beragam seperti aplikasipengukuran jarak. Alat ini secara umum memancarkan gelombang suara ultrasonik menuju suatu target yang memantulkan balik gelombang kearah sensor. Kemudian system mengukur waktu yang diperlukan untuk pemancaran gelombang sampai kembali ke sensor dan menghitung jarak target dengan menggunakan kecepatan suara dalam medium.



Gambar 2.20 Sensor Ultrasonik HC-SR04

(Sumber: www.electronics.com)

Karakteristik sensor ultrasonic HC-SR04 (www.electfreaks.com, 2013)

- a. Tegangan operasi DC-5V
- b. Arus operasi 15mA
- c. Frekuensi operasi 40KHZ
- d. Jarak terjauh 4m
- e. Jarak menengah 2cm
- f. Sudut dalam mengukur 15°
- g. *Input Trigger Signal* 10us TTL pulse

Rangkaian penyusun sensor ultrasonik ini terdiri dari *transmitter*, *Receiver*, dan komparator. Selain itu, gelombang ultrasonik dibangkitkan oleh sebuah kristal tipis bersifat *piezoelektrik*. Bagian-bagian dari sensor ultrasonik adalah sebagai berikut:

a) *Piezoelektrik*

Peralatan *piezoelektrik* secara langsung mengubah energi listrik menjadi energi mekanik. Tegangan *input* yang digunakan menyebabkan bagian keramik meregang dan memancarkan gelombang ultrasonik. Tipe operasi transmisi elemen *piezoelektrik* sekitar frekuensi 32 kHz. Efisiensi lebih baik, jika frekuensi osilator diatur pada frekuensi resonansi *piezoelektrik* dengan sensitifitas dan efisiensi paling baik. Jika rangkaian pengukur beroperasi pada mode pulsa elemen *piezoelektrik* yang sama dapat digunakan sebagai *transmitter* dan *receiver*. Frekuensi yang ditimbulkan tergantung pada osilatornya yang disesuaikan frekuensi kerja dari masing-masing transduser. Karena kelebihan ini maka transduser *piezoelektrik* lebih sesuai digunakan untuk sensor ultrasonik.

b) *Transmitter*

Transmitter adalah sebuah bagian yang berfungsi sebagai pemancar gelombang ultrasonik dengan frekuensi sebesar 40 kHz yang dibangkitkan dari sebuah osilator. Untuk menghasilkan frekuensi 40 KHz, harus di buat sebuah rangkaian osilator dan keluaran dari osilator dilanjutkan menuju penguat sinyal. Besarnya frekuensi ditentukan oleh komponen RLC / kristal tergantung dari desain osilator yang digunakan. Penguat sinyal akan memberikan sebuah sinyal listrik yang diumpankan ke *piezoelektrik* dan terjadi reaksi mekanik

sehingga bergetar dan memancarkan gelombang yang sesuai dengan besar frekuensi pada osilator.

c) Receiver

Receiver terdiri dari transduser ultrasonik menggunakan bahan *piezoelektrik*, yang berfungsi sebagai penerima gelombang pantulan yang berasal dari *transmitter* yang dikenakan pada permukaan suatu benda atau gelombang langsung LOS (*Line of Sight*) dari *transmitter*. Oleh karena bahan *piezoelektrik* memiliki reaksi yang *reversible*, elemen keramik akan membangkitkan tegangan listrik pada saat gelombang datang dengan frekuensi yang resonan dan akan menggetarkan bahan *piezoelektrik* tersebut. Penerima ultrasonik ini akan menerima sinyal ultrasonik yang dipancarkan oleh pemancar ultrasonik dengan karakteristik frekuensi yang sesuai. Sinyal yang diterima tersebut akan melalui proses *filterisasi frekuensi* dengan menggunakan rangkaian *band pass filter* (penyaring pelewat pita), dengan nilai frekuensi yang dilewatkan telah ditentukan. Kemudian sinyal keluarannya akan dikuatkan dan dilewatkan ke rangkaian komparator (pembanding) dengan tegangan referensi ditentukan berdasarkan tegangan keluaran penguat pada saat jarak antara sensor kendaraan mini dengan sekat/dinding pembatas mencapai jarak minimum untuk berbelok arah. Dapat dianggap keluaran komparator pada kondisi ini adalah *high* (logika "1") sedangkan jarak yang lebih jauh adalah *low* (logika "0"). Logika-logika biner ini kemudian diteruskan ke rangkaian pengendali (mikrokontroler).

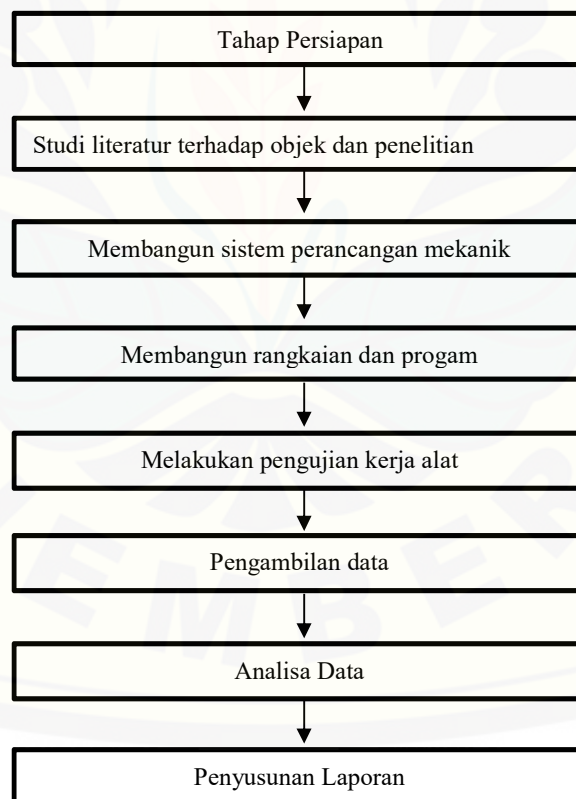
BAB 3. METODELOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian ini, tentang “*Prototype Sistem Deteksi Plat Nomor Kendaraan Pada Sistem Perparkiran Berbasis Image Processing Menggunakan Metode Optical Character Recognition (OCR)*“ ini dilakukan pada bulan Agustus 2017 di Laboratorium Jaringan Komunikasi, Jalan Slamet Riyadi No. 62 Patrang, Jember 68111

3.2 Tahapan Penelitian

Dalam pembuatan skripsi dan penelitian ini, dibuat langkah-langkah atau prosedur penelitian sebagai berikut:



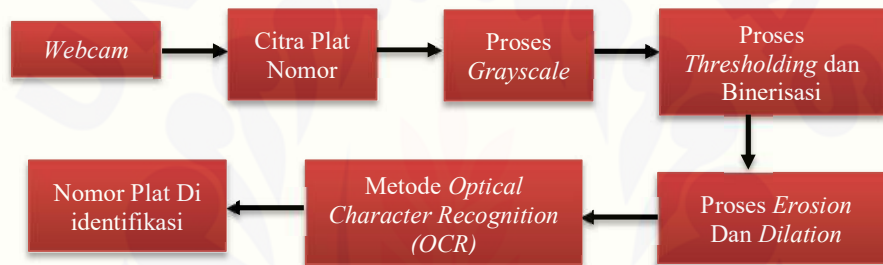
Gambar 3.1 Blok diagram Tahap Penelitian

Tahap penelitian ini meliputi beberapa tahap mulai dari tahap persiapan alat dan bahan yang nantinya akan digunakan dalam proses penelitian. Selanjutnya

mengumpulkan study literature terhadap objek penelitian tentang destilasi air laut. Setelah literature sudah mencukupi, dilakukan pembuatan mekanik alat, rangkaian, dan program sebagai sarana awal untuk pengambilan data yang mana mekanik alat, rangkaian, dan program harus dilakukan pengujian terlebih dahulu untuk menentukan kelayakan alat dan bahan yang digunakan dalam pengambilan data percobaan. Setelah itu baru dilakukan pengambilan data yang akan digunakan untuk proses analisis dan pembuatan laporan penelitian.

3.3 Perancangan Sistem

3.3.1 Diagram Blok Deteksi Plat Nomor

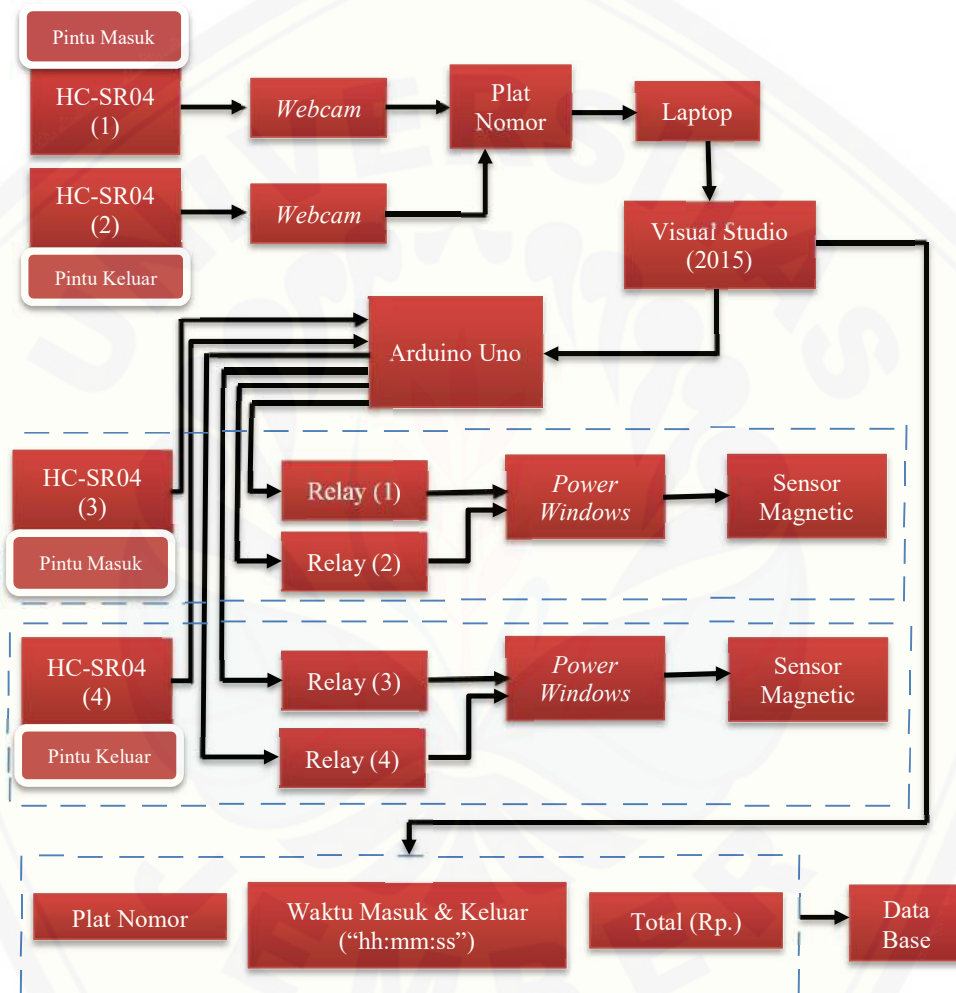


Gambar 3.2 Diagram Blok Alat Deteksi Plat Nomor Pada Sistem Perparkiran

Pada penelitian ini citra plat nomor akan diambil menggunakan *logitech HD webcam* dengan resolusi kamera sebesar 3 MP. Pada diagram blok deteksi plat nomor menggunakan 3 proses *filtering* dan untuk pengindetifikasi karakter plat nomor menggunakan metode *optical character recognition* (OCR). Pada saat *webcam* pengambilan citra gambar plat nomor atau bisa disebutkan sebagai citra RGB, selanjutnya dilakukan proses *filtering grayscale* di mana fungsi dari *filtering* ini mengubah citra gambar RGB menjadi citra keabu-abuan dan kemudian dilakukan proses *thresholding* yang berfungsi untuk mengubah derajat keabu-abuan dan citra gambar akan memiliki 2 warna yaitu hitam dan putih. Pada *pixel – pixel* objek bernilai 1 dan *pixel – pixel* latar belakang bernilai 0. Pada citra dapat di sebutkan 0 adalah putih dan 1 adalah hitam, proses ini dapat disebutkan sebagai proses binerisasi. Proses Selanjutnya yaitu *filtering erosion* yaitu proses *filtering* yang berfungsi untuk memperkecil *pixel* pada citra gambar dan *filtering dilation* yaitu proses *filtering* yang berfungsi untuk memperbesar *pixel* pada citra gambar.

Proses berikutnya dilakukan pengidentifikasi citra gambar untuk pembacaan karakter plat nomor yaitu menggunakan metode *optical character recognition* (OCR).

3.3.2 Diagram Blok Alat Deteksi Plat Nomor Pada Sistem Perparkiran



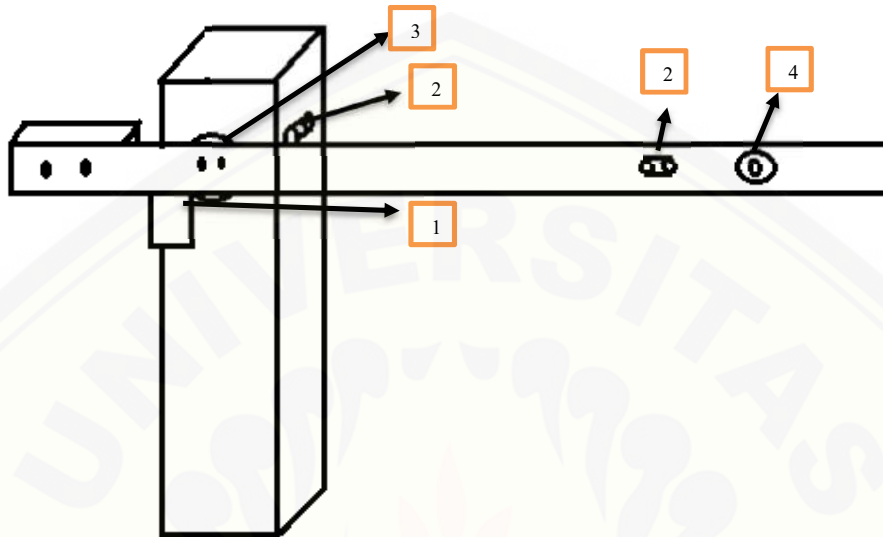
Gambar 3.3 Diagram Blok Alat Deteksi Plat Nomor Pada Sistem Perparkiran

Pada penelitian ini citra plat nomor akan diambil menggunakan *logitech HD webcam* dengan resolusi kamera sebesar 3 MP. Pada diagram blok perancangan ini terdiri dari *webcam*, sebuah perangkat laptop dan software Visual Studio. Ada 2 proses yaitu pada pintu masuk dan keluar, 2 proses tersebut memiliki alur yang sama dan di proses otomatis melalui 1 laptop. Plat nomor yang dipakai

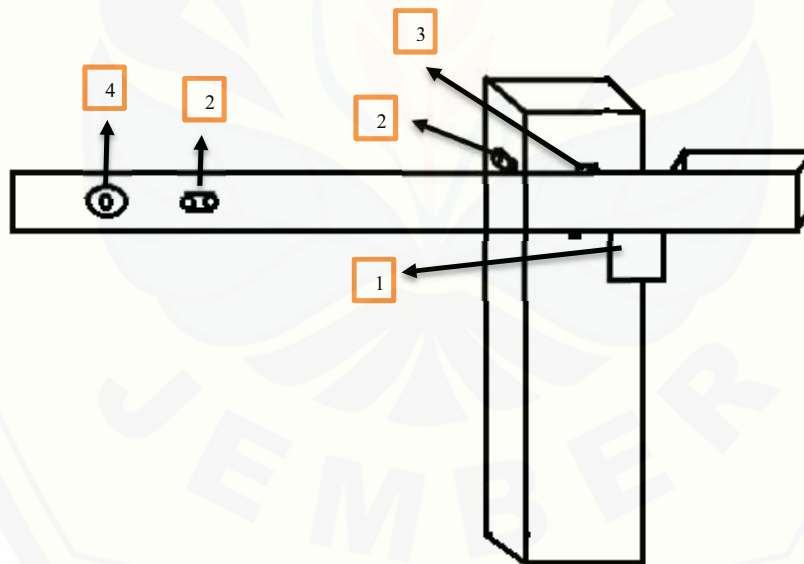
dengan prosedur yang ada di negara Indonesia yaitu dengan latar belakang plat berwarna hitam dan nomor berwarna putih, *WebCam* berfungsi sebagai pengambilan gambar pada plat nomor dan sebuah perangkat laptop berfungsi untuk mengolah hasil citra gambar plat nomor yang didapatkan dari *WebCam* dengan menggunakan software Visual Studio. Pada pintu masuk HC-SR04(1) dengan jarak tertentu akan mengaktifkan *webcam* kemudian dilakukan pengambilan gambar pada plat nomor dan akan diolah di laptop menggunakan software visual studio (visual basic.NET) dan hasil *output* berupa plat nomor dan waktu masuk, proses selanjutnya akan mengirimkan *serial port* dari visual studio ke Arduino, selanjutnya akan memicu Arduino untuk mengaktifkan *relay* dan menggerakkan motor *power windows* searah jarum jam dan pada sensor *magnetic* berfungsi untuk menghentikan motor *power windows* , selanjutnya pada HC-SR04(3) dengan jarak tertentu akan mengaktifkan *relay* dan menggerakkan motor *power windows* berlawanan jarum jam dan pada sensor *magnetic* berfungsi untuk menghentikan motor *power windows*, sedangkan pada pintu keluar HC-SR04(2) dengan jarak tertentu akan mengaktifkan *webcam* kemudian dilakukan pengambilan gambar pada plat nomor dan akan diolah di laptop menggunakan software visual studio (visual basic.NET) dan hasil *output* berupa plat nomor dan waktu keluar kemudian dihasilkan selisih dari waktu keluar dengan waktu masuk dikalikan dengan harga perwaktu. Pada saat bersamaan plat nomor, waktu masuk dan waktu keluar akan di simpan pada database, proses selanjutnya akan mengirimkan *serial port* dari visual studio ke Arduino, selanjutnya akan memicu Arduino untuk mengaktifkan *relay* dan menggerakkan motor *power windows* berlawanan jarum jam dan pada sensor *magnetic* berfungsi untuk menghentikan motor *power windows* , selanjutnya pada HC-SR04(4) dengan jarak tertentu akan mengaktifkan *relay* dan menggerakkan motor *power windows* searah jarum jam dan pada sensor *magnetic* berfungsi untuk menghentikan motor *power windows*.

3.4 Perancangan Mekanik

3.4.1 Desain Alat Sistem Perparkiran



Gambar 3.4 Alat Sistem Perparkiran Untuk Pintu Masuk



Gambar 3.5 Alat Sistem Perparkiran Untuk Pintu Keluar

Dalam perancangan *design* mekanik pada gambar diatas dibutuhkan bahan-bahan sebagai berikut :

1. Motor *Power Windows*
2. HC-SR04

3. Sensor *Magnetic*
4. *Webcam Logitech* HD
5. Besi
6. Plat Aluminium
7. mur baut

Dari semua bahan yang ada disusun hingga menjadi satu kesatuan seperti pada gambar diatas. Menggunakan *power Window* karena untuk menggerakkan plat aluminium yang memerlukan motor dengan torsi besar. HC-SR04 berfungsi menghidupkan *webcam Logitech* HD dan untuk menggerakkan motor *power windows*, sedangkan untuk sensor *magnetic* berfungsi untuk menghentikan pergerakan motor *power windows*.

3.4.2 Rancangan Keseluruhan Alat



Gambar 3.6 Gambar keseluruhan alat

Keterangan :

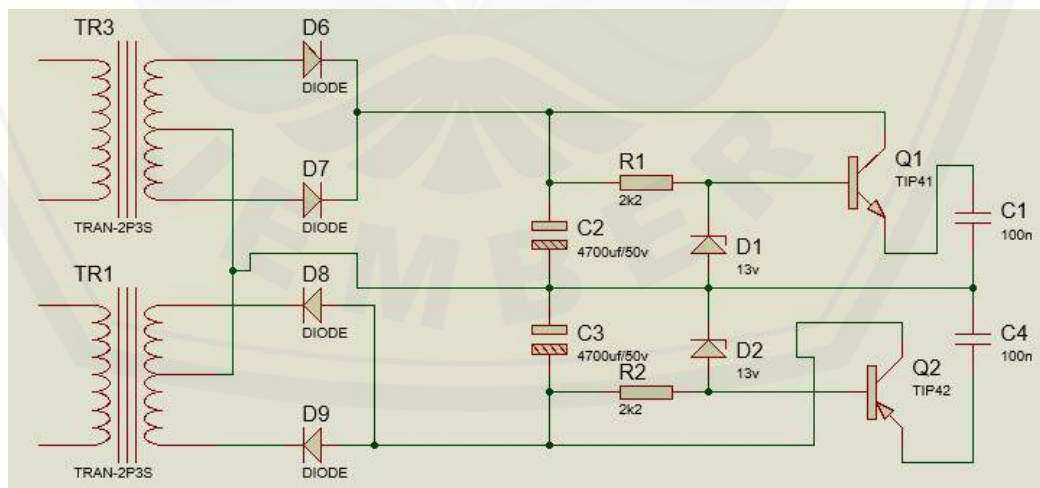
1. Webcam Logitech HD 1
2. Webcam Logitech HD 2
3. Sensor Ultrasonic HC-SR04 (1)
4. Sensor Ultrasonic HC-SR04 (2)
5. Sensor Ultrasonic HC-SR04 (3)
6. Sensor Ultrasonic HC-SR04 (4)
7. Sensor *Magnetic*
8. Sensor *Magnetic*
9. Motor *Power Windows*
10. Motor *Power Windows*

Motor *Power windows* merupakan alat untuk menggerakkan palang pintu

3.5 Perancangan *Hardware*

3.5.1 Perancangan Rangkaian *Power supply*

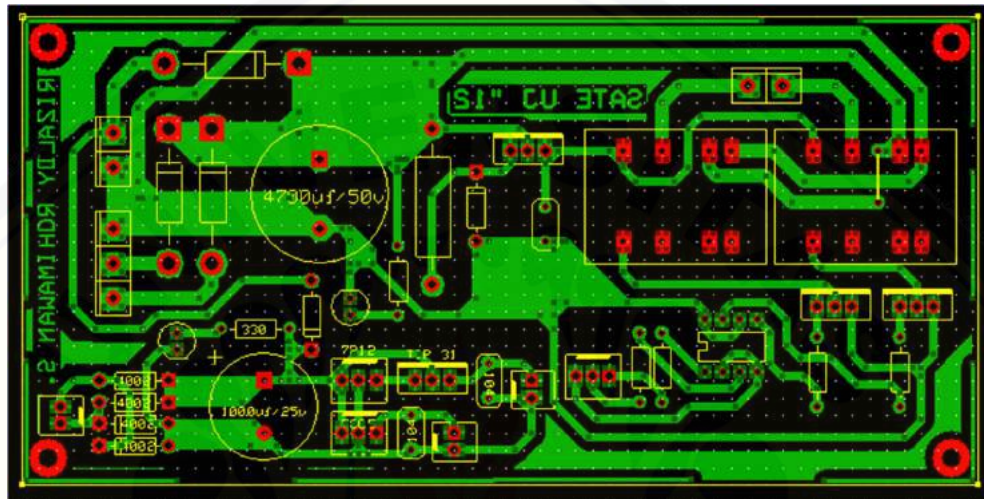
Pada alat ini, *power supply* digunakan sebagai sumber tegangan untuk 2 motor *power windows*. Dalam perancangan *power supply* ini terdapat *power supply* simetris di mana digunakan sebagai sumber tegangan *driver* motor. Berikut ini adalah skema rancangan *power supply* yang digunakan sesuai dengan gambar 3.8



Gambar 3.7 Rancangan Rangkaian *Power Supply* Simetris

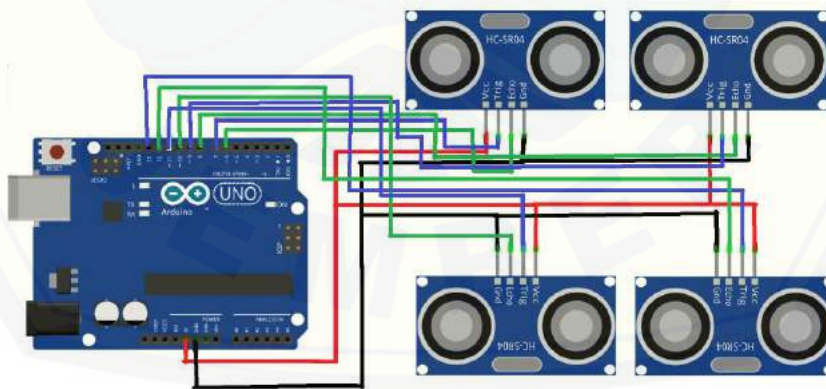
3.5.2 Layout PCB

Pembuatan *layout* PCB didesign menggunakan *software* ExpressPCB. Pada 1 *layout* PCB terdapat rangkaian *power supply* simetris dan rangkaian *relay* berfungsi sebagai *driver* motor. Rangkaian ini dijadikan 1 *layout* karena untuk menghemat ruang sehingga lebih terlihat praktis.



Gambar 3.8 *Layout* PCB *Power Supply* Simetris, *Power Supply* Regulator, Dan *Driver* Motor

3.5.3 Arduino Uno dengan Sensor HC-SR04

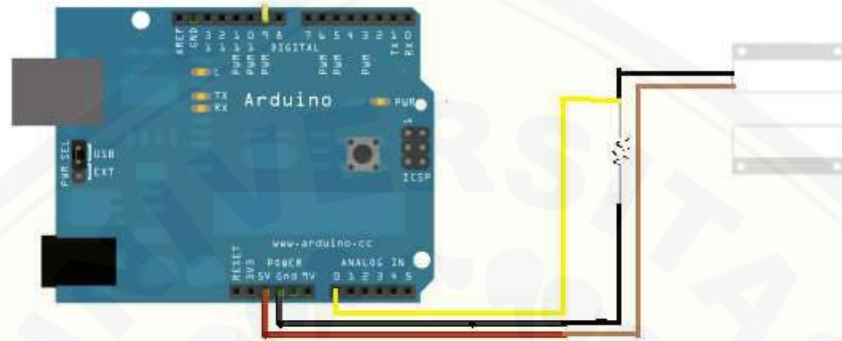


Gambar 3.9 Rangkaian Arduino Dengan HC-SR04

Pada rangkaian sensor HC-SR04 ini digunakan untuk Menghidupkan *webcam* dan menggerakkan motor *power windows*. Pada sensor HC-SR04 pin *trig*

dihubungkan pada pin (7,9,11,13) arduino , sedangkan pin *echo* dihubungkan pada pin (6,8,10,12) arduino.

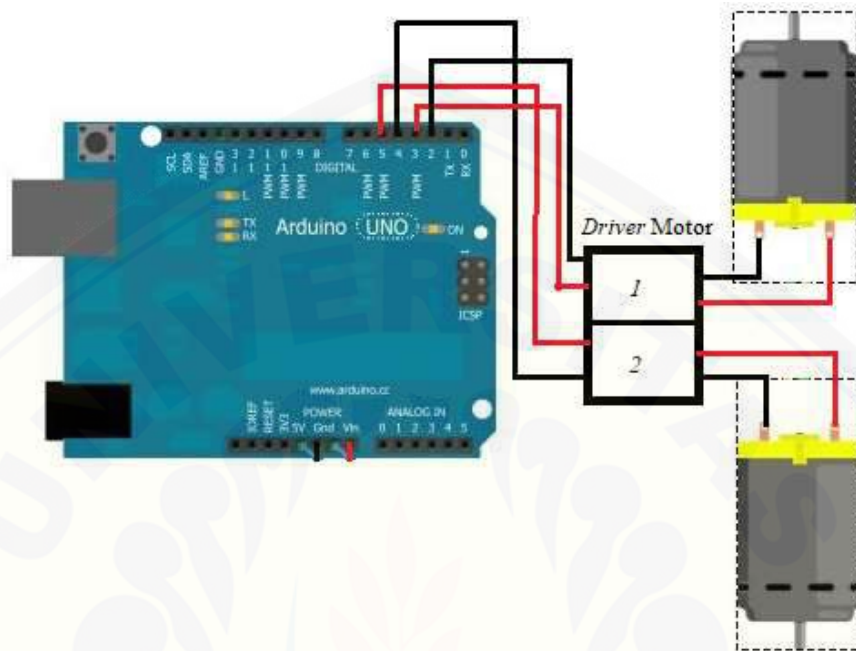
3.5.4 Arduino Uno dengan Sensor *Magnetic*



Gambar 3.10 Rangkaian Arduino Dengan Sensor *Magnetic*

Rangkaian sensor *magnetic* ini berfungsi untuk menghentikan pergerakan *power windows*, pada sensor *magnetic* ini bias dikatakan fungsinya sebagai limit *switch*. Untuk rangkaiannya sensor *magnetic* sisi satu komponen dihubungkan pada pin 5 V sedangkan lainnya di hubungkan dengan resistor dan selanjutnya dihubungkan pada pin GND, pada pin *analog in* (A0) dihubungkan pada kaki sensor *magnetic* sebelum di hubungkan dengan resistor. Pada penelitian ini di gunakan 4 sensor *magnetic* sehingga pada kaki sensor *magnetic* dihubungkan pada pin *analog in* (A0-A3)

3.5.5 Arduino Uno dengan Motor DC



Gambar 3.11 Rangkaian Arduino Dengan *Driver* Motor dan Motor DC

Pada rangkaian arduino ini berfungsi untuk menggerakkan motor DC dengan memberikan logika pada program arduino *High* dan *Low*. Untuk *driver* motornya digunakan rangkaian *relay* dimana fungsi *relay* ini digunakan untuk mengatur nilai *high* dan *low* secara bergantian sehingga dapat menggerakkan motor DC secara kiri atau kanan. Pada kaki motor DC (1) dihubungkan pada pin (2,3) arduino, sedangkan kaki motor DC (2) dihubungkan pada pin (4,5).

3.5.6 Webcam



Gambar 3.12 Webcam Logitech C270

Pada *webcam* berfungsi untuk pengambilan input citra gambar pada plat nomor dan *webcam* dapat berfungsi jika sensor ultrasonik HCR-SR04 membaca jarak di bawah sebesar 20 cm. Pada program visual studio (VB.NET) untuk pengambilan input citra gambar menggunakan *webcam* sebagai berikut :

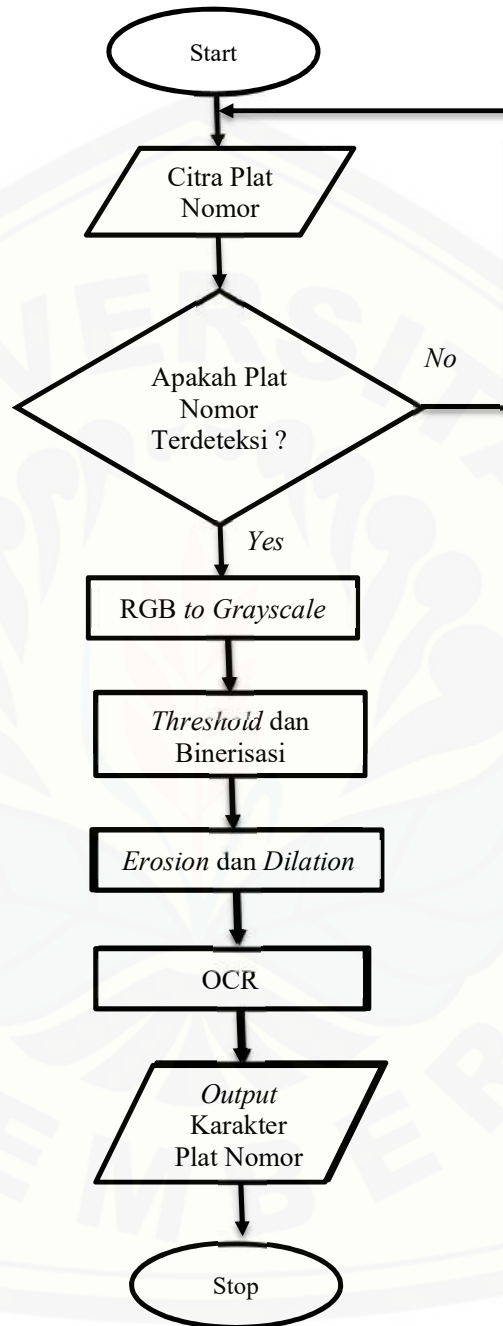
```
CamDevice.SetCaptureProperty(Emgu.CV.CvEnum.CAP_PROP.CV_CAP_PROP_FRAME_WIDTH,320)
CamDevice.SetCaptureProperty(Emgu.CV.CvEnum.CAP_PROP.CV_CAP_PROP_FRAME_HEIGHT,240)
ImgRGB = CamDevice.QueryFrame
```

Pada program visual studio (VB.NET) untuk pengaktifan *webcam* sebagai berikut :

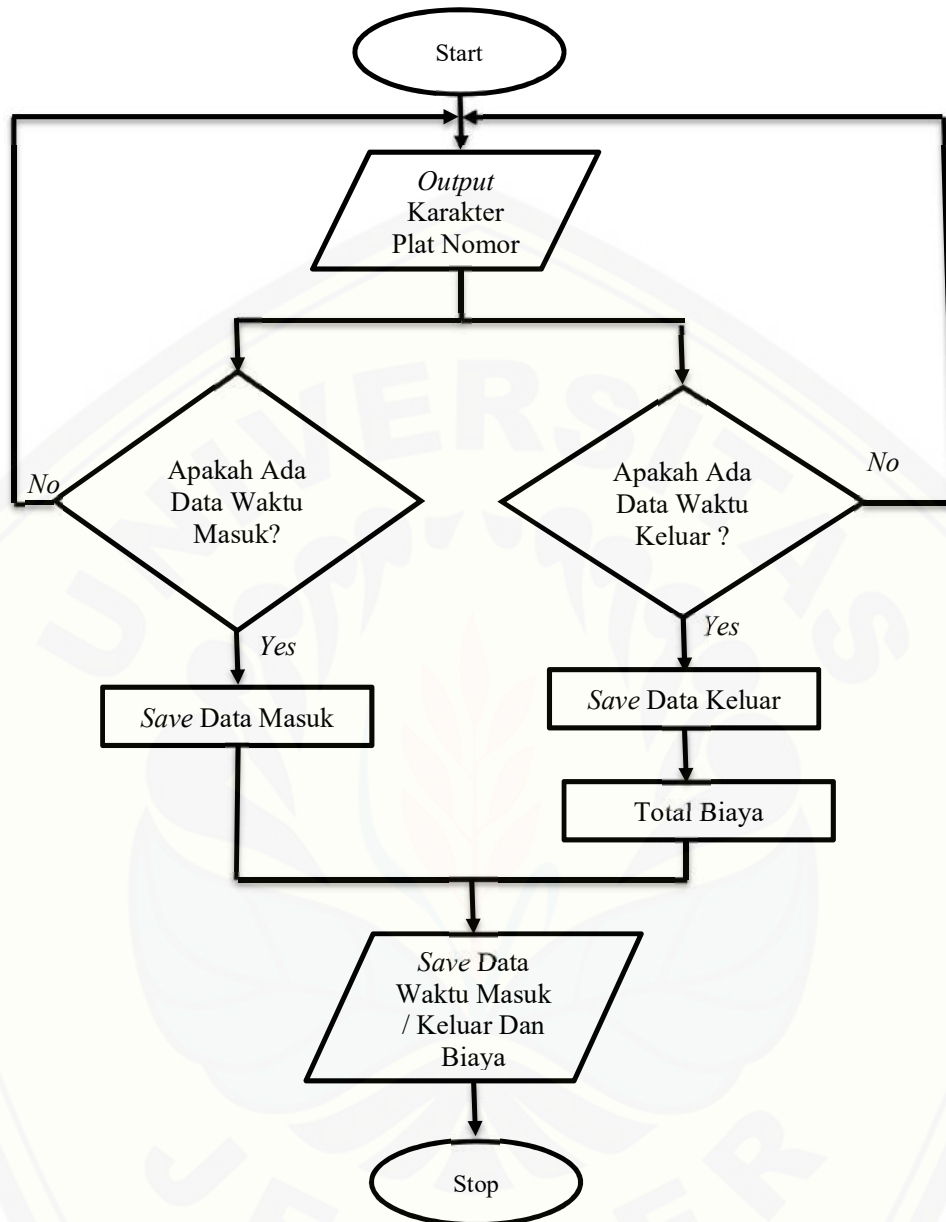
```
read = readbuffer.Replace(vbCr, "").Replace(vbLf, "")
aryTextFile = read.Split("|")
TextBox6.Text = aryTextFile(0)
TextBox1.Text = aryTextFile(1)

If TextBox6.Text <= 20 Then
    Timer1.Enabled = True
End If
If TextBox1.Text <= 20 Then
    Timer2.Enabled = True
End If
```

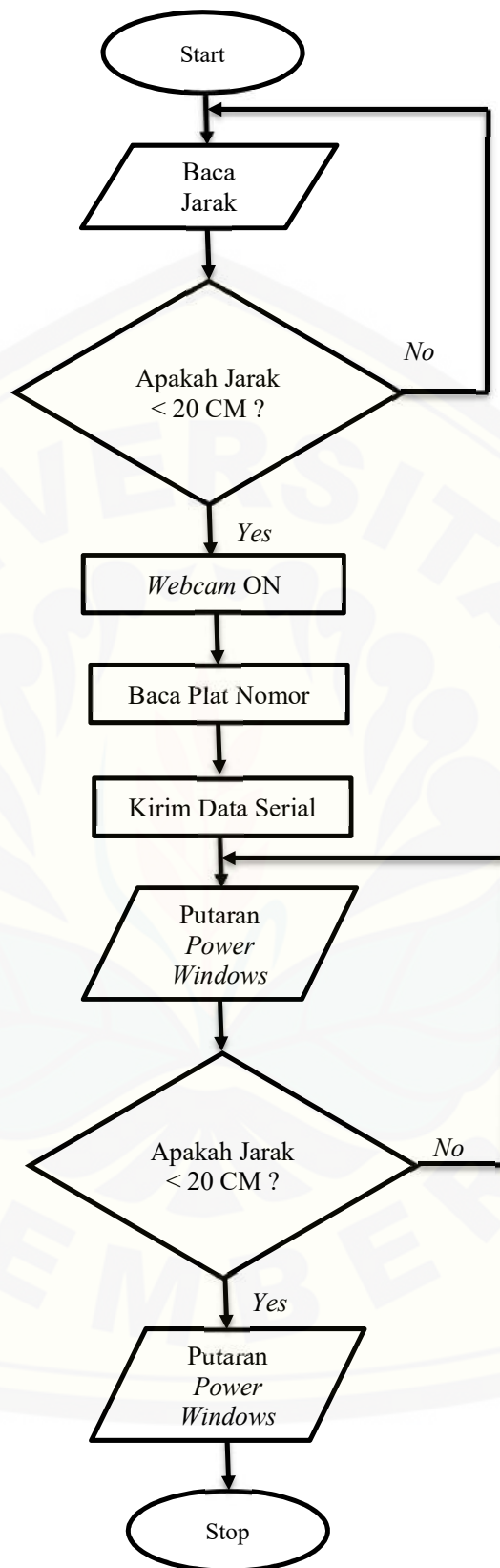
3.6 FlowChart



Gambar 3.13 Flowchart Perancangan Sistem Deteksi Nomor Plat Nomor Pada Pintu Masuk Dan Keluar



Gambar 3.14 Flowchart *Save Data Output* Plat Nomor ke DataBase Pada Pintu Masuk Dan Keluar



Gambar 3.15 Flowchart Sistem Perparkiran

3.7 Perancangan *Software*

Pada penelitian ini dilakukan tahap pengambilan data terdiri dari 10 citra plat nomor. Citra plat nomor pada pintu masuk diambil dengan posisi pengambilan citra lurus sejajar dengan jarak sekitar 57 cm antara *webcam* dengan plat nomor, sedangkan citra plat nomor pada pintu masuk diambil dengan posisi pengambilan citra lurus sejajar dengan jarak sekitar 52 cm antara *webcam* dengan plat nomor. Masing-masing plat nomor akan di uji pada system perparkiran dengan kondisi di palang pintu masuk dan keluar. Aplikasi pada penelitian ini dibuat dengan memanfaatkan *library* EmguCV pada Visual Studio C+. Hasil pengujian pada pengambilan data antara plat nomor akan di samakan dengan hasil pembacaan karakter plat nomor pada *database* di *visual studio*.

3.7.1 Proses Deteksi Plat Nomor Kendaraan

Pada penelitian ini proses deteksi plat nomor dilakukan beberapa tahap pengolahan citra yaitu pengambilan input citra gambar, *filtering citra* yaitu *grayscale*, *thresholding* dan binerisasi, *erosion* dan *dilate*. Setelah proses pengolahan citra dilakukan proses deteksi karakter plat nomor dengan metode *optical character recognition* (OCR). Pada program visual studio (VB.NET) untuk tahapan pengolahan citra dan metode *optical character recognition* (OCR) sebagai berikut :

```
ImgRGB = CamDevice.QueryFrame
ImageBox1.Image = ImgRGB

ImgGray = ImgRGB.Convert(Of Gray, Byte)()
ImgBiner = ImgGray.ThresholdBinary(New Gray(200), New Gray(255))

erode = ImgBiner2.Erode(1)
dilate = erode1.Dilate(2)
ocr.Recognize(dilate)
TextBox2.Text = ocr.GetText
```

3.7.2 Input Citra

Proses pertama pada pengenalan karakter plat nomor adalah inialisasi variabel dan *input* citra pada sistem. Pada penelitian ini citra yang digunakan berwarna RGB dengan format bmp dan ukuran *pixel* yaitu 320x240 *pixel*. Pada

citra berwarna memiliki kombinasi warna *red green* dan *blue* pada setiap *pixel*-nya, setiap warna dasar tersebut memiliki nilai minimum 0 dan nilai maksimum 255 (8 bit). Pada gambar 4.5 merupakan citra plat nomor pada pintu masuk yang diambil dengan jarak antara *webcam* dengan plat nomor sejauh 62 cm serta posisi *webcam* sejajar dengan plat nomor, sedangkan pada gambar 4.6 merupakan citra plat nomor pada pintu keluar yang diambil dengan jarak antara *webcam* dengan plat nomor sejauh 52 cm serta posisi *webcam* sejajar dengan plat nomor.



Gambar 3.16 Input Citra RGB 1



Gambar 3.17 Input Citra RGB 2

3.7.3 Filtering Citra

Citra masukan selanjutnya akan dilakukan proses *filtering* pada citra. *Filtering* pada citra ini terdiri dari beberapa proses pengolahan citra diantaranya *grayscale*, *thresholding*, dan *erosion*.

a. Grayscale

Pada proses *grayscale*, citra masukan yang berwarna RGB (*Red Green Blue*) dikonversi kedalam bentuk citra berderajat keabuan (*grayscale*). Pada citra *grayscale* hanya memiliki satu nilai kanal pada setiap *pixel*-nya, artinya nilai *red*, *green*, dan *blue* adalah sama besar. Proses mengubah citra berwarna menjadi citra berderajat keabuan ini dapat dilakukan dengan cara mengambil nilai rata-rata dari nilai R, G, dan B. Sehingga dapat dituliskan menjadi $R+G+B/3$. Jika dilihat dari gambar 4.2, citra *grayscale* tersebut terdiri dari warna abu-abu yang bervariasi pada warna hitam sebagai intensitas yang terendah dan warna putih sebagai intensitas tertinggi.



Gambar 3.18 Citra *Image 1* Dengan *Filtering Grayscale*



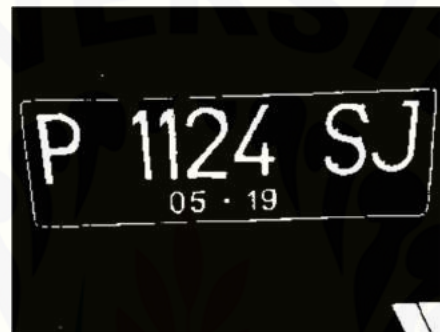
Gambar 3.19 Citra *Image 2* Dengan *Filtering Grayscale*

b. *Thresholding* dan Binerisasi

Pada proses binerisasi, citra masukan dari proses *grayscale* akan dikonversi kedalam bentuk citra biner yaitu citra dalam hitam dan putih. Tiap *pixel* dalam citra levelnya dirubah melalui nilai dari *thresholding*, apabila *pixel* nilai diatas *thresholding* maka *pixel* citra *image* akan diubah ke warna putih, sedangkan pada level di bawah nilai *thresholding* maka *pixel* citra *image* akan diubah ke warna hitam. Jika dilihat dari gambar 4.9 dan gambar 4.10, citra *image* dengan proses *thresholding* tersebut memiliki nilai *thresholding* 200.



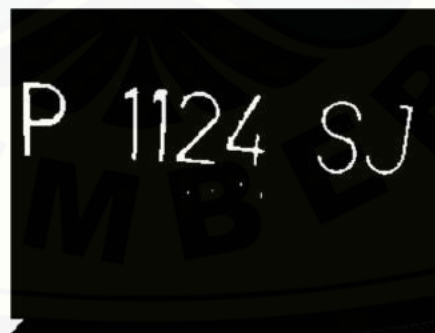
Gambar 3.20 Citra *Image 1* Dengan *Filtering Thresholding*



Gambar 3.21 Citra *Image 2* Dengan *Filtering Thresholding*

c. *Erosion*

Pada proses *erosion*, dimana citra masukan dari proses *thresholding* dan binerisasi akan dilakukan proses perkecilan pada *pixel*. Jika dilihat dari gambar 4.11 dan gambar 4.12, citra *image* dengan proses *erosion* terdapat proses perkecilan *pixel* pada citra *image*.



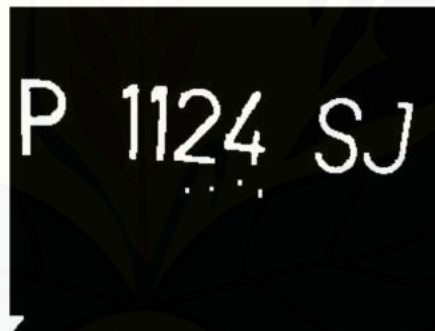
Gambar 3.22 Citra *Image 1* dengan Proses *Erosion*



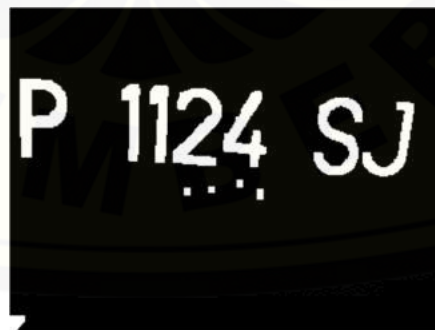
Gambar 3.23 Citra *Image 2* dengan Proses *Erosion*

d. Dilate

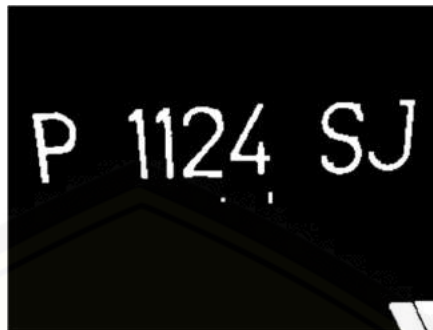
Pada proses *dilate*, dimana citra masukan dari proses *erosion* akan dilakukan proses pembesaran pada *pixel*. Pada proses *dilate* akan dilakukan proses 2x pembesaran pada *pixel*. Jika dilihat dari gambar 4.13, gambar 4.14, gambar 4.15 dan gambar 4.16, citra *image* dengan proses *dilate* terdapat proses pembesaran *pixel* pada citra *image*.



Gambar 3.24 Citra *Image 1* dengan Proses *Dilate 1x*

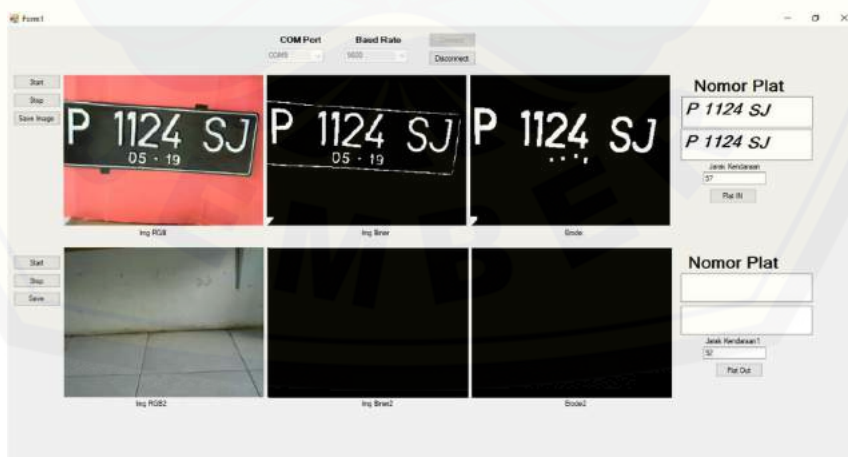


Gambar 3.25 Citra *Image 1* dengan Proses *Dilate 2x*

Gambar 3.26 Citra *Image 2* dengan Proses *Dilate 1x*Gambar 3.27 Citra *Image 2* dengan Proses *Dilate 2x*

3.7.4 Metode *Optical Character Recognition (OCR)*

Metode *optical character recognition (OCR)*, di mana sebuah proses metode *scanning* untuk pembacaan karakter pada citra *image*. Metode OCR merupakan *datasheet library emguCV*.

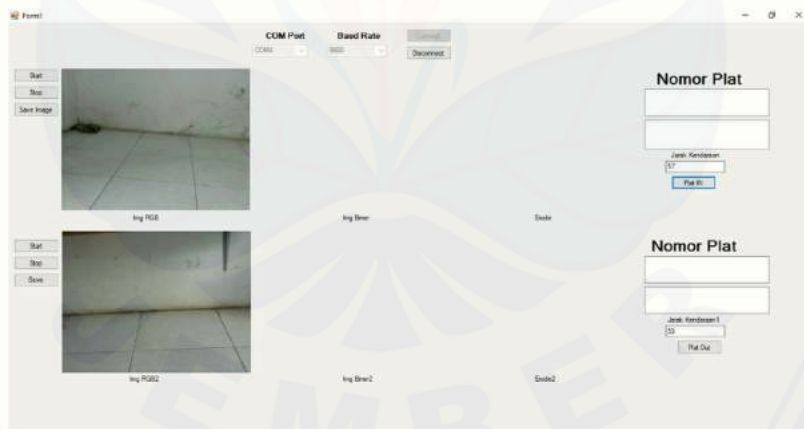
Gambar 3.28 Proses Metode OCR Pada Citra *Image 1* Dengan Hasil P 1124 SJ



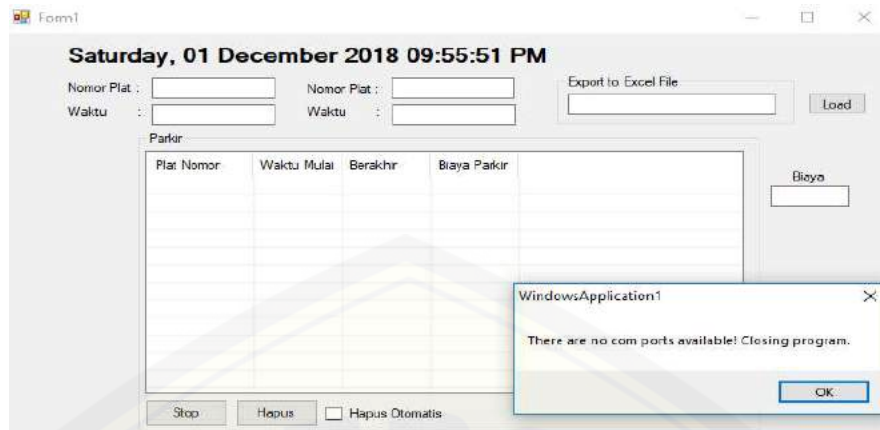
Gambar 3.29 Proses Metode OCR Pada Citra *Image 2* Dengan Hasil P 1124 SJ

3.7.5 Aplikasi Prototype Sistem Deteksi Plat Nomor

Program aplikasi yang dibuat pada penelitian ini dapat berfungsi jika *port* arduino telah dibaca oleh komputer, sedangkan jika *port* Arduino tidak terbaca oleh komputer maka saat dibuka aplikasi ada sebuah peringatan “ *There Are No Com Ports Available! Closing Program* “ dan selanjutnya aplikasi akan *close*.



Gambar 3.30 Aplikasi Dapat Berfungsi



Gambar 3.31 Aplikasi Tidak Dapat Berfungsi

3.7.6 Database Software Visual Studio

Setelah proses metode *optical character recognition* (OCR), proses selanjutnya adalah menampilkan proses *database* tiap selesai *scanning* pembacaan karakter pada citra *image*. Proses penyimpanan *database* ini dilakukan hasil dari deteksi plat nomor dan selanjutnya dengan menghitung nilai waktu tiap menit dan total pembayaran dari plat nomor pada saat keadaan antara pintu masuk dan pintu keluar.

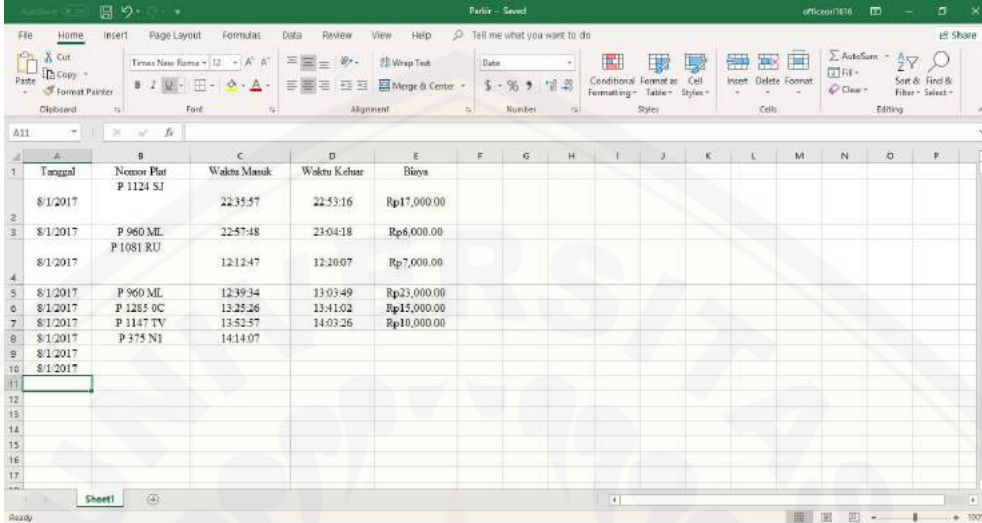


Gambar 3.32 Hasil Database Software Visual Studio

3.7.7 Database Microsoft Excel

Setelah proses metode *optical character recognition* (OCR) dan menampilkan proses *database* tiap selesai *scanning* pembacaan karakter pada citra

image, selanjutnya proses penyimpanan *database* keseluruhan pada *microsoft excel*.



The screenshot shows a Microsoft Excel spreadsheet with the following data:

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Tanggal	Nomor Plat	Waktu Masuk	Waktu Keluar	Biaya										
2	8/1/2017	P 1124 SJ	22:35:57	22:53:16	Rp17,000.00										
3	8/1/2017	P 960 ML	22:57:48	23:04:18	Rp6,000.00										
4	8/1/2017	P 1081 RU	12:12:47	12:20:07	Rp7,000.00										
5	8/1/2017	P 960 ML	12:39:34	13:03:49	Rp23,000.00										
6	8/1/2017	P 1285 0C	13:25:26	13:41:02	Rp15,000.00										
7	8/1/2017	P 1147 TV	13:52:57	14:03:26	Rp10,000.00										
8	8/1/2017	P 375 N1	14:14:07												
9	8/1/2017														
10	8/1/2017														
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															

Gambar 3.33 Hasil *Database Software Visual Studio*

BAB 5. PENUTUP

Dari hasil-hasil pengujian dan pembahasan yang telah dilakukan dari penelitian ini, dapat diambil beberapa kesimpulan dan saran yang dapat digunakan sebagai dasar untuk penelitian selanjutnya.

5.1 Kesimpulan

Berdasarkan hasil analisis prototype sistem deteksi plat nomor menggunakan metode *optical character recognition* (OCR) pada sistem parkir, dapat diambil kesimpulan sebagai berikut.

1. Berdasarkan tabel 4.2 (halaman 54) data hasil pengujian sistem deteksi plat nomor kendaraan menggunakan *optical character recognition* (OCR) yaitu pada plat nomor kendaraan P 1285 QC baik pada pintu masuk maupun pintu keluar teridentifikasi P 1285 0C, jadi metode OCR mengidentifikasi karakter Q pada karakter 0, sedangkan pada plat nomor kendaraan P 375 NL di pintu masuk teridentifikasi P 375 N1, jadi metode OCR mengidentifikasi karakter L dengan 1.
2. Berdasarkan tabel 4.2 (halaman 54) data hasil pengujian sistem deteksi plat nomor kendaraan menggunakan *optical character recognition* (OCR) yaitu tidak dapat dideteksi pada plat nomor kendaraan P 1226 QI dan P 1146 TY karena pada *image processing* akhir citra karakter terhapus, sedangkan pada plat nomor P 1501 KX terdapat noise di tiap karakter citra.

5.2 Saran

Sebagai upaya pengembangan penelitian ke arah lebih baik lagi, dibawah ini terdapat saran yang dapat diberikan dari hasil penelitian pada tugas akhir ini.

1. Perbandingan ukuran plat nomor kendaraan dari pembacaan karakter menggunakan metode OCR (*Optical Character Recognition*) dapat digunakan sebagai referensi untuk menggunakan plat nomor kendaraan di Indonesia.

2. Penggunaan perbandingan metode yang digunakan pada proses pengenalan karakter untuk mengetahui kelemahan dan kelebihan masing-masing metode.
3. Penambahan proses citra *image processing* yang berfungsi untuk memperhalus citra gambar sehingga mudah dalam pembacaan karakter.

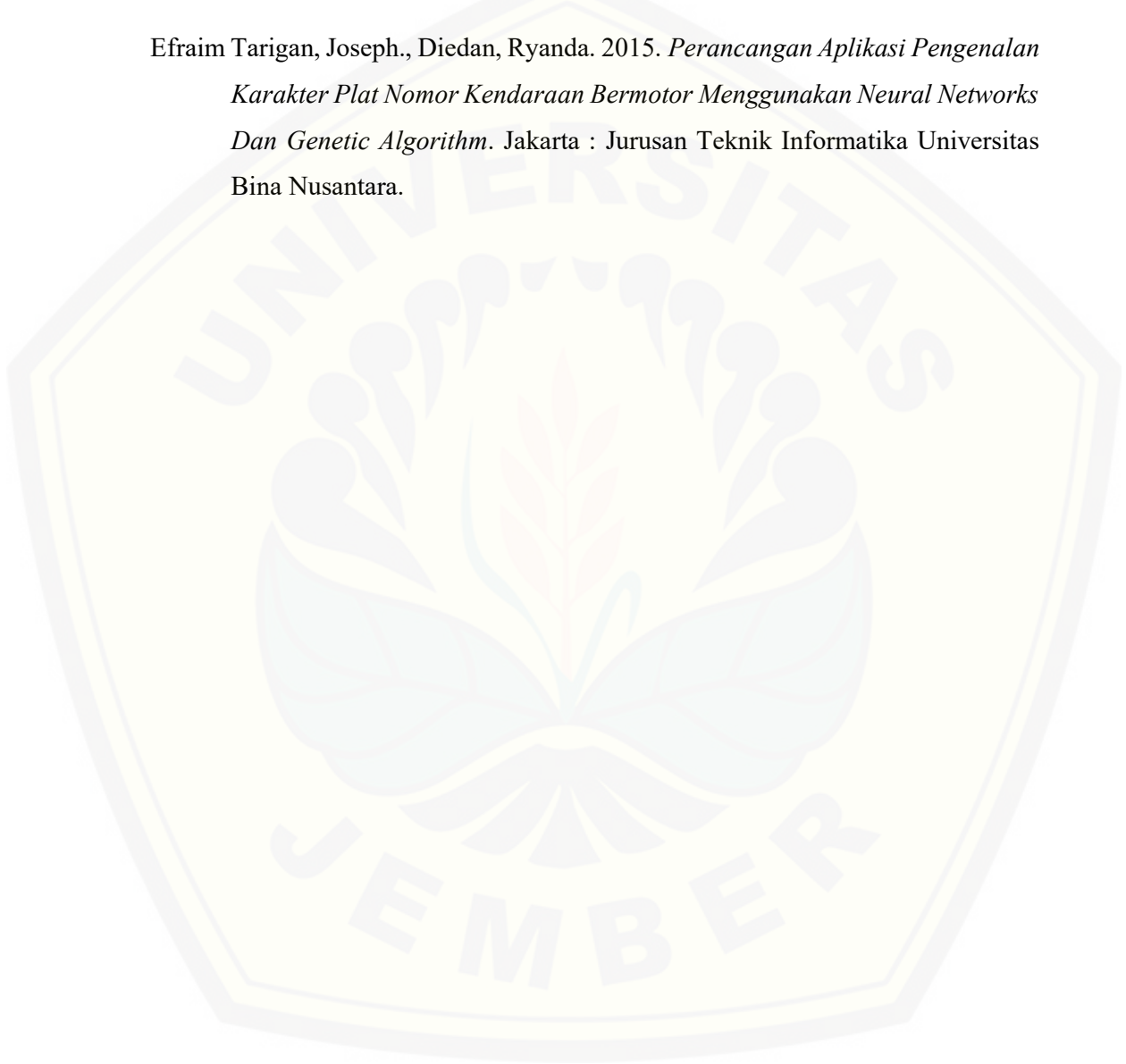


Daftar Pustaka

- Asyra., Anton., Ade., Dwi., Dkk. 2010. *Connected Component Analysis Sebagai Metode Pencarian Karakter Plat Dalam Sistem Pengenalan Plat Nomor Kendaraan*. Jakarta : Jurusan Teknik Informatika Universitas Al Azhar Indonesia.
- Andy., Sigit., dan Dwi. *Sistem Pengenalan Plat Nomor Mobil Untuk Aplikasi Informasi Karcis Parkir*. Surabaya : Jurusan Teknik Komputer Universitas Politeknik Elektronika Negeri Surabaya – Institut Teknologi Sepuluh Nopember (ITS).
- Sri Wahyono, Eko., Ernastuti. *Identifikasi Nomor Polisi Mobil Menggunakan Metode Jaringan Saraf Buatan Learning Vector Quantizatio*. Depok: Jurusan Teknik Informatika, Universitas Gunadarma.
- Wakhidah, Nur. 2012 *Deteksi Plat Nomor Kendaraan Bermotor Berdasarkan Area pada Image Segmentation*. Semarang: Fakultas Teknologi Informasi dan Komunikasi Universitas Semarang.
- Prasetya Wicaksana, Riza. *Pengenalan Plat Nomor Kendaraan Secara Otomatis Untuk Pelanggaran Lalu Lintas*. Surabaya: Teknik Komputer dan Telematika Jurusan Teknik Elektro Insitut Teknologi Sepuluh Nopember Suarabaya.
- Setian, Riki. 2014. *Rancang Bangun Gerakan Robot 2 Dof Menggunakan Image Processing Dengan Metode Fuzzy Adaptive*. Jember: Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.
- Setiadi, Himawan. 2012. *Perancangan Program Deteksi Dan Pengenalan Plat Nomor Kendaraan Menggunakan Algoritma Ekstraksi Kontur Dan Ocr*. Jakarta : Jurusan Teknik Informatika Dan Matematika Universitas Bina Nusantara.

Napitupula, Josua., Marshall, Daniel., Mahargono. 2008. *Analisis Dan Perancangan Sistem Pengenal Karakter Pada Plat Nomor Kendaraan Bermotor Dengan Pendekatan Logika Fuzzy*. Jakarta : Jurusan Teknik Informatika Universitas Bina Nusantara.

Efrain Tarigan, Joseph., Diedan, Ryanda. 2015. *Perancangan Aplikasi Pengenalan Karakter Plat Nomor Kendaraan Bermotor Menggunakan Neural Networks Dan Genetic Algorithm*. Jakarta : Jurusan Teknik Informatika Universitas Bina Nusantara.



Lampiran

List Main Program

```
Imports Emgu.CV
Imports Emgu.Util
Imports Emgu.CV.OCR
Imports Emgu.CV.Structure

Imports System.IO
Imports System.IO.Ports
Imports System.Threading

Public Class Form1
    Dim comOpen As Boolean      'Keeps track of the port status. True =
Open; False = Closed
    Dim readbuffer As String   'Buffer of whats read from the serial port
    Dim CamDevice As New Capture(0)
    Dim CamDevice1 As New Capture(1)
    Dim ImgGray As New Image(Of Gray, Byte)(320, 240)
    Dim ImgBiner As New Image(Of Gray, Byte)(320, 240)
    Dim ImgGray2 As New Image(Of Gray, Byte)(320, 240)
    Dim ImgBiner2 As New Image(Of Gray, Byte)(320, 240)
    Dim ImgRGB As New Image(Of Bgr, Byte)(320, 240)
    Dim ImgRGB2 As New Image(Of Bgr, Byte)(320, 240)

    Dim ocr As Tesseract = New Tesseract("tessdata", "eng",
Tesseract.OcrEngineMode.OEM_TESSERACT_ONLY)
    Dim ocr1 As Tesseract = New Tesseract("tessdata", "eng",
Tesseract.OcrEngineMode.OEM_TESSERACT_ONLY)

    Dim flag As Integer = 0
    Dim x As Integer
    Dim y As Integer
    Dim lebar As Integer
    Dim tinggi As Integer

    Dim aryTextFile() As String

    Dim comPORT As String
    Dim receivedData As String = ""

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        Parkir.Show()
        ImageBox1.Visible = True
        ImageBox2.Visible = True
        ImageBox3.Visible = True
        ImageBox4.Visible = True
        ImageBox5.Visible = True
        ImageBox6.Visible = True
        Timer1.Enabled = False
        Timer2.Enabled = False
    End Sub
End Class
```

```

    Dim comPorts As String() = System.IO.Ports.SerialPort.GetPortNames

    If comPorts.Count < 1 Then
        'If thereThenThen Then are Not Ports connected, Show an Error
        And close the program.
        MsgBox("There are no com ports available! Closing program.")
        Me.Close()
    Else
        ComboBox1.Items.AddRange(comPorts)
        ComboBox1.Text = comPorts(0)
    End If
End Sub
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    ImageBox1.Visible = True
    ImageBox2.Visible = True
    ImageBox3.Visible = True
    .

    ImgGray = ImgRGB.Convert(Of Gray, Byte)()
    ImgBiner = ImgGray.ThresholdBinary(New Gray(200), New Gray(255))

    Dim erode As Image(Of Gray, Byte)
    Dim dilate As Image(Of Gray, Byte)
    erode = ImgBiner.Erode(1)
    dilate = erode.Dilate(2)

    ImageBox2.Image = ImgBiner
    ImageBox3.Image = dilate

    '-----OCR-----
    ocr.Recognize(dilate)
    TextBox9.Text = ocr.GetText

End Sub

Private Sub Button6_Click(sender As Object, e As EventArgs) Handles
Button6.Click
    'SerialPort1.WriteLine("1")
    Parkir.TextBox1.Text = TextBox9.Text
    Parkir.Show()

End Sub

Private Sub Button10_Click(sender As Object, e As EventArgs) Handles
Button10.Click
    SaveFileDialog1.FileName = "Gambar1"
    SaveFileDialog1.Filter = "Gambar1| *.bmp"
    If SaveFileDialog1.ShowDialog() = DialogResult.OK Then
        ImageBox1.Image.Save(SaveFileDialog1.FileName)
    End If
End Sub

```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Timer1.Enabled = True
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Timer1.Enabled = False
End Sub

Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
    ImageBox4.Visible = True
    ImageBox5.Visible = True
    ImageBox6.Visible = True

    ImgGray2 = ImgRGB2.Convert(Of Gray, Byte)()
    ImgBiner2 = ImgGray2.ThresholdBinary(New Gray(200), New Gray(255))

    Dim erode1 As Image(Of Gray, Byte)
    Dim dilate1 As Image(Of Gray, Byte)
    erode1 = ImgBiner2.Erode(1)
    dilate1 = erode1.Dilate(2)

    ImageBox5.Image = ImgBiner2
    ImageBox6.Image = dilate1

'-----OCR-----
    ocr1.Recognize(dilate1)
    TextBox10.Text = ocr1.GetText

End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click
    Timer2.Enabled = True
End Sub

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles
Button4.Click
    Timer2.Enabled = False
End Sub

Private Sub Button9_Click(sender As Object, e As EventArgs)
    ImageBox1.Visible = False
    ImageBox2.Visible = False
    ImageBox3.Visible = False

    TextBox9.Visible = False
    TextBox10.Visible = False

    ImageBox4.Visible = False
    ImageBox5.Visible = False
```



```

        ImageBox6.Visible = False
    End Sub

    Private Sub TextBox9_TextChanged(sender As Object, e As EventArgs)
Handles TextBox9.TextChanged
        Timer1.Enabled = False
        SerialPort1.WriteLine("2")
        Parkir.TextBox1.Text = TextBox9.Text
        Parkir.Show()

    End Sub

    Private Sub TextBox10_TextChanged(sender As Object, e As EventArgs)
Handles TextBox10.TextChanged
        Timer2.Enabled = False
        SerialPort1.WriteLine("1")
        Parkir.TextBox2.Text = TextBox10.Text
        Parkir.Show()

    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles
Button5.Click
        'Connect the serial port
        DoConnect()
    End Sub

    Private Sub Button7_Click(sender As Object, e As EventArgs) Handles
Button7.Click
        DoDisconnect()
    End Sub
    Public Sub DoDisconnect()
        'Graceful disconnect if port is open
        If comOpen Then
            SerialPort1.DiscardInBuffer()
            SerialPort1.Close()

            'Reset our flag and controls
            comOpen = False
            Button7.Enabled = False
            Button5.Enabled = True
            cmbBaud.Enabled = True
            ComboBox1.Enabled = True
        End If
    End Sub

    Public Sub DoConnect()
        'Setup the serial port connection
        With SerialPort1()
            .PortName = ComboBox1.Text                'Selected Port
            .BaudRate = CInt(cmbBaud.Text)            'Baud Rate. 9600 is
default.
            .Parity = IO.Ports.Parity.None
            .DataBits = 8
            .StopBits = IO.Ports.StopBits.One
            .Handshake = IO.Ports.Handshake.None
        End With
    End Sub

```

```

        .RtsEnable = False
        .ReceivedBytesThreshold = 1
        .NewLine = vbCr
        .ReadTimeout = 10000
    End With

    'Try to open the selected port...
    Try
        SerialPort1.Open()
        comOpen = SerialPort1.IsOpen
    Catch ex As Exception
        comOpen = False
        MsgBox("Error Open: " & ex.Message)
    End Try

    Button7.Enabled = True
    Button5.Enabled = False
    cmbBaud.Enabled = False
    ComboBox1.Enabled = False
End Sub

Private Sub Button12_Click(sender As Object, e As EventArgs) Handles
Button12.Click
    SaveFileDialog2.FileName = "Gambar1"
    SaveFileDialog2.Filter = "Gambar1| *.bmp"
    If SaveFileDialog2.ShowDialog() = DialogResult.OK Then
        ImageBox4.Image.Save(SaveFileDialog2.FileName)
    End If
End Sub

Private Sub Button13_Click(sender As Object, e As EventArgs) Handles
Button13.Click
    Timer3.Enabled = False
    'SerialPort1.WriteLine("2")
    Parkir.TextBox2.Text = TextBox10.Text
    Parkir.Show()

End Sub

Private Sub Timer3_Tick(sender As Object, e As EventArgs) Handles
Timer3.Tick

CamDevice.SetCaptureProperty(Emgu.CV.CvEnum.CAP_PROP.CV_CAP_PROP_FRAME_WIDT
H, 320)

CamDevice.SetCaptureProperty(Emgu.CV.CvEnum.CAP_PROP.CV_CAP_PROP_FRAME_HEIG
HT, 240)

    ImgRGB = CamDevice.QueryFrame
    ImageBox1.Image = ImgRGB

    ImgGray = ImgRGB.Convert(Of Gray, Byte)()
    ImgBiner = ImgGray.ThresholdBinary(New Gray(200), New Gray(255))

    Dim erode As Image(Of Gray, Byte)
    Dim dilate As Image(Of Gray, Byte)
    erode = ImgBiner.Erode(1)
    dilate = erode.Dilate(2)

```

```

'=====OCR=====
    ocr.Recognize(dilate)
    TextBox2.Text = ocr.GetText

End Sub

Private Sub Timer4_Tick(sender As Object, e As EventArgs) Handles
Timer4.Tick

CamDevice1.SetCaptureProperty(Emgu.CV.CvEnum.CAP_PROP.CV_CAP_PROP_FRAME_WID
TH, 320)

CamDevice1.SetCaptureProperty(Emgu.CV.CvEnum.CAP_PROP.CV_CAP_PROP_FRAME_HEI
GHT, 240)
    ImgRGB2 = CamDevice1.QueryFrame
    ImageBox4.Image = ImgRGB2

    ImgGray2 = ImgRGB2.Convert(Of Gray, Byte)()
    ImgBiner2 = ImgGray2.ThresholdBinary(New Gray(200), New Gray(255))

    Dim erode1 As Image(Of Gray, Byte)
    Dim dilate1 As Image(Of Gray, Byte)
    erode1 = ImgBiner2.Erode(1)
    dilate1 = erode1.Dilate(2)

'=====OCR=====
    ocr1.Recognize(dilate1)
    TextBox3.Text = ocr1.GetText

End Sub

Private Sub SerialPort1_DataReceived(ByVal sender As System.Object, ByVal e
As System.IO.Ports.SerialDataReceivedEventArgs) Handles
SerialPort1.DataReceived
    If comOpen Then

        Try
            'Send data to a new thread to update the ph display
            readbuffer = SerialPort1.ReadLine()
            Me.Invoke(New EventHandler(AddressOf updateData))
        Catch ex As Exception
            'Otherwise show error. Will display when disconnecting.
            'MsgBox(ex.Message)
        End Try
    End If
End Sub

Public Sub updateData(ByVal sender As Object, ByVal e As
System.EventArgs)
    'Update data
    Dim read As String

```

```

read = readbuffer.Replace(vbCr, "").Replace(vbLf, "")
aryTextFile = read.Split("|")
TextBox6.Text = aryTextFile(0)
TextBox1.Text = aryTextFile(1)

If TextBox6.Text <= 20 Then
    Timer1.Enabled = True
End If
If TextBox1.Text <= 20 Then
    Timer2.Enabled = True
End If

End Sub

End Class

List Program DataBase

Imports Excel = Microsoft.Office.Interop.Excel
Public Class Parkir

    Dim BiayaPerMenitRodaDua As Double = (1000 / 60) ' Sama saja Rp 1000
per jam
    Dim BiayaPerMenitRodaEmpat As Double = (2000 / 60) ' Sama saja Rp 2000
per jam
    Dim li As New ListViewItem
    Dim BiayaAbsRodaDua As Integer
    Dim BiasaAbsRodaEmpat As Integer
    Dim timedate As Date
    Dim timedate1 As Date
    Dim excelready As Integer
    Dim strFileName As String

    Dim BiayaDua As String

    Dim APP As Excel.Application
    Dim worksheet As Excel.Worksheet
    Dim workbook As Excel.Workbook
    Dim y As Integer = 2
    Dim lRow As Long = 0

    Dim cari As Integer = 0

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        InitializeView()
        Dim timemont As String
        timemont = Now.ToString("dddd, dd MMMM yyyy hh:mm:ss tt")

```

```

        Label5.Text = timemont
    End Sub

    Private Sub InitializeView()

        ' Mengatur listview

        With lstDua
            .FullRowSelect = True
            .GridLines = True
            .Columns.Add("Plat Nomor").Width = 85
            .Columns.Add("Waktu Mulai").Width = 72
            .Columns.Add("Berakhir").Width = 72
            .Columns.Add("Biaya Parkir").Width = 71
        End With

    End Sub

    Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs)
Handles TextBox1.TextChanged
        timedate = Now.ToLongTimeString
        TextBox3.Text = timedate

        If IsExists("2", UCase(Trim(TextBox1.Text))) Then
            MsgBox("Maaf Plat Nomor Sudah Ada",
                MsgBoxStyle.Information, "Masih Ada")
            Exit Sub
        End If

        li = lstDua.Items.Add(TextBox1.Text)
        li.SubItems.Add(Now.ToString("hh':'mm':'ss")).Tag = Now
        worksheet.Cells(lstDua.Items.Count + 1, 3).value = TimeString

        workbook.Save()

    End Sub

    Private Sub TextBox2_TextChanged(sender As Object, e As EventArgs)
Handles TextBox2.TextChanged
        timedate1 = Now.ToLongTimeString
        TextBox4.Text = timedate1

        If TextBox2.Text = "" Then
            MsgBox("Masukkan plat nomor dan jenis kendaraan yang akan
dicari", MsgBoxStyle.Critical, "Cari")
            Form1.Timer1.Enabled = False
            Exit Sub

        End If
        lstDua.SelectedItems.Clear()
        If IsExists("2", UCase(Trim(TextBox2.Text))) = False Then

```



```

        MsgBox("Kendaraan tidak ditemukan atau mungkin kendaraan telah
keluar", MsgBoxStyle.Information, "Tidak ditemukan")
        Form1.Timer2.Enabled = False
    End If

    With lstDua
        If .SelectedItems.Count > 0 Then
            'BiayaDua = DateDiff(DateInterval.Second,
            .Items(.SelectedItems(0).Index).SubItems(1).Tag, Now) * 1000
            BiayaDua = Val(Format(DateDiff(DateInterval.Minute,
            .Items(.SelectedItems(0).Index).SubItems(1).Tag, Now) * TextBox5.Text,
            "#"))
            MsgBox("Nomor Plat : " & TextBox2.Text &
            Environment.NewLine & "Biaya parkir Rp." & BiayaDua)

            worksheet.Cells(cari, 5).Value = BiayaDua
            workbook.Save()

            .SelectedItems(0).BackColor = Color.Red

            .SelectedItems(0).SubItems.Add(Now.ToString("hh':'mm':'ss"))
            .SelectedItems(0).SubItems.Add("Rp." & BiayaDua)

        End If
    End With

End Sub

Private Sub cmdDel12_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdDel12.Click
    With lstDua
        If .SelectedItems.Count > 0 Then
            If MsgBox("Hapus dari daftar?", MsgBoxStyle.YesNo +
            MsgBoxStyle.Question, "Hapus") = MsgBoxResult.Yes Then
                .Items.Remove(.SelectedItems(0))
            End If
        End If
    End With
End Sub

Private Function IsExists(ByVal j As Char, ByVal Plat As String) As
Boolean
    If j = "2" Then
        For i = 0 To lstDua.Items.Count - 1

            If UCase(lstDua.Items(i).Text) = UCase(Plat) And
            lstDua.Items(i).BackColor <> Color.Red Then
                ' Jika sudah ada dan belum di stop

                lstDua.Items(i).Selected = True

                worksheet.Cells(i + 2, 1).value = DateString

```

```

        worksheet.Cells(i + 2, 2).Value = lstDua.Items(i).Text
        'worksheet.Cells(i + 2, 3).value = TimeString

        worksheet.Cells(i + 2, 4).value = TimeString

        cari = i + 2

        workbook.Save()

        lstDua.Focus()
        IsExists = True
    End If
Next
Else
    End If

End Function

Private Sub chkAutoDel2_CheckedChanged(sender As Object, e As
EventArgs) Handles chkAutoDel2.CheckedChanged
    lstDua.Items.Clear()
End Sub

Private Sub LoadExcel_Click(sender As Object, e As EventArgs) Handles
LoadExcel.Click
    OpenFileDialog1.Filter = "Excel (*.xlsx)|*.xlsx"
    If OpenFileDialog1.ShowDialog() = DialogResult.OK Then
        namafile.Text = OpenFileDialog1.FileName
    End If

    If namafile.Text <> "" Then
        APP = New Excel.Application
        workbook = APP.Workbooks.Open(namafile.Text)
        worksheet = workbook.Worksheets("sheet1")
    End If
End Sub
End Class

```

List Program Arduino

```

#include <Wire.h>

int MMA = 2; //Motor masuk arah Atas
int MMB = 3; //Motor masuk arah Bawah
int MKA = 4; //Motor keluar arah Atas
int MKB = 5; //Motor keluar arah Bawah
int LMA = A0; //Limit masuk atas
int LMB = A1; //Limit masuk bawah
int LKA = A2; //Limit keluar atas
int LKB = A5; //Limit keluar atas

int echoPin0 = 10;
int trigPin0 = 11;

```

```
int echoPin1 = 12;
int trigPin1 = 13;
int echoPin2 = 6;
int trigPin2 = 7;
int echoPin3 = 8;
int trigPin3 = 9;

long duration0, distance0;
long duration1, distance1;
long duration2, distance2;
long duration3, distance3;

void baca0() {
  digitalWrite(trigPin0, LOW); // Added this line
  delayMicroseconds(10); // Added this line
  digitalWrite(trigPin0, HIGH);
  delayMicroseconds(10); // Added this line
  digitalWrite(trigPin0, LOW);
  duration0 = pulseIn(echoPin0, HIGH);
  distance0 = (duration0 / 2) * 0.03448;
}

void baca1() {
  digitalWrite(trigPin1, LOW); // Added this line
  delayMicroseconds(10); // Added this line
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10); // Added this line
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);
  distance1 = (duration1 / 2) * 0.03448;
}

void baca2() {
  digitalWrite(trigPin2, LOW); // Added this line
  delayMicroseconds(10); // Added this line
  digitalWrite(trigPin2, HIGH);
  delayMicroseconds(10); // Added this line
  digitalWrite(trigPin2, LOW);
  duration2 = pulseIn(echoPin2, HIGH);
  distance2 = (duration2 / 2) * 0.03448;
}

void baca3() {
  digitalWrite(trigPin3, LOW); // Added this line
  delayMicroseconds(10); // Added this line
  digitalWrite(trigPin3, HIGH);
  delayMicroseconds(10); // Added this line
  digitalWrite(trigPin3, LOW);
  duration3 = pulseIn(echoPin3, HIGH);
  distance3 = (duration3 / 2) * 0.03448;
}

void MMU() {
  digitalWrite(MMA, LOW);
  digitalWrite(MMB, HIGH);
}
```

```
void MMD() {
    digitalWrite(MMA, HIGH);
    digitalWrite(MMB, LOW);
}

void MKU() {
    digitalWrite(MKA, LOW);
    digitalWrite(MKB, HIGH);
}

void MKD() {
    digitalWrite(MKA, HIGH);
    digitalWrite(MKB, LOW);
}

void MMS() {
    digitalWrite(MMA, LOW);
    digitalWrite(MMB, LOW);
}

void MKS() {
    digitalWrite(MKA, LOW);
    digitalWrite(MKB, LOW);
}

void setup() {
    Serial.begin(9600);
    //Serial.println("--- Start Serial Monitor SEND_RCVE ---");

    Wire.begin();

    pinMode(trigPin0, OUTPUT);
    pinMode(echoPin0, INPUT);
    pinMode(trigPin1, OUTPUT);
    pinMode(echoPin1, INPUT);
    pinMode(trigPin2, OUTPUT);
    pinMode(echoPin2, INPUT);
    pinMode(trigPin3, OUTPUT);
    pinMode(echoPin3, INPUT);

    for (int a = 2; a <= 5; a++)
        pinMode(a, OUTPUT);

    //for (int b = 6; b <= 11; b++)
    //pinMode(b, INPUT);

    pinMode(A0, INPUT);
    pinMode(A1, INPUT);
    pinMode(A2, INPUT);
    pinMode(A3, INPUT);

    if (digitalRead(LMB) == LOW) {
        MMD();
    }

    if (digitalRead(LMB) == HIGH)
```

```
MMS();

if (digitalRead(LKB) == LOW) {
    MKD();
}

if (digitalRead(LKB) == HIGH)
    MKS();
}

void loop()
{
    digitalWrite(MMA, LOW);
    digitalWrite(MMB, LOW);
    digitalWrite(MKA, LOW);
    digitalWrite(MKB, LOW);

    baca0();
    baca1();
    baca2();
    baca3();
    Serial.print(distance2);
    Serial.print("|");
    Serial.println(distance3);

    if (Serial.available())
    {
        char number = Serial.read();
        if (number == '1') {
            MMU();
            delay(300);
            MMS();
            delay(300);
            MMU();
            delay(300);
            MMS();
            delay(300);
            MMU();

            //Serial.println("masuk");
            for (;;) {
                if (digitalRead(LMA) == HIGH) {
                    //Serial.println("Pintu Terbuka");
                    MMS();
                    break;
                }
            }
        }

        for (;;) {
            boolean masuk = false;

            baca0();
            if (distance0 <= 50) {
```



```
for (;;) {
  //Serial.print("Jarak sensor dengan benda = ");
  //Serial.print(distance1);
  //Serial.println(" cm");
  baca0();
  if (distance0 > 50) {
    delay(500);
    //Serial.print("Jarak sensor dengan benda = ");
    //Serial.print(distance1);
    //Serial.println(" cm");
    masuk = true;
    break;
  }
}
}

//BacaSerial();

if (masuk == true) {
  MMD();

  //Serial.println("masuk");
  break;
}
}

for (;;) {
  if (digitalRead(LMB) == HIGH) {
    //Serial.println("Pintu Tertutup");
    MMS();
    break;
  }
}
}

if (number == '2') {
  MKU();
  delay(300);
  MKS();
  delay(300);
  MKU();
  delay(300);
  MKS();
  delay(300);
  MKU();

  //Serial.println("keluar");
  for (;;) {
    if (digitalRead(LKA) == HIGH) {
      MKS();
      break;
    }
  }
}

for (;;) {
```

```
boolean keluar = false;

bacal();
if (distancel <= 50) {
  for (;;) {
    //Serial.print("Jarak sensor dengan benda = ");
    //Serial.print(distancel);
    //Serial.println(" cm");
    bacal();
    if (distancel > 50) {
      delay(500);
      //Serial.print("Jarak sensor dengan benda = ");
      //Serial.print(distancel);
      //Serial.println(" cm");
      keluar = true;
      break;
    }
  }
}

//BacaSerial();
if (keluar == true) {
MKD();

  Serial.println("keluar");
  break;
}
}

for (;;) {
  if (digitalRead(LKB) == HIGH) {
    MKS();
    break;
  }
}
}
```