

**PENGENALAN PLAT NOMOR KENDARAAN PADA CITRA
HASIL CAPTURE KAMERA MENGGUNAKAN METODE
MORFOLOGI DAN TEMPLATE MATCHING**

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Ahmad Rifqie Ilham F.

NIM: 125150300111006



**PROGRAM STUDI TEKNIK INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016**

PENGESAHAN

PENGENALAN PLAT NOMOR KENDARAAN PADA CITRA HASIL CAPTURE KAMERA
MENGUNAKAN METODE MORFOLOGI DAN TEMPLATE MATCHING

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Ahmad Rifqie Ilham F.

NIM: 125150300111006

Skripsi ini telah diuji dan dinyatakan lulus pada
25 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing

Fitri Utamingrum, Dr. Eng., S.T, M.T.

NIP: 19820710 200812 2 001

Mengetahui

Ketua Program Studi Informatika

Drs. Marji, M.T.

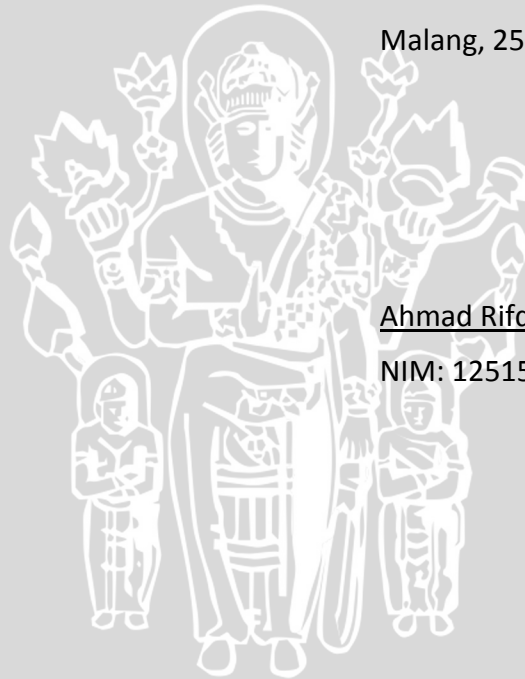
NIP: 19670801 199203 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 25 Januari 2016



Ahmad Rifqie Ilham F.

NIM: 125150300111006

KATA PENGANTAR

Puji syukur kehadiran Tuhan YME yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Pengenalan Plat Nomor Kendaraan Pada Citra Hasil Capture Kamera Menggunakan Metode Morfologi Dan Template Matching” ini dapat terselesaikan.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Kedua orang tua penulis, abah Adenan dan ibu Mutmainah yang selalu memberikan doa, motivasi, kasih sayang, dukungan moril dan materil sebagai penyemangat dalam menyelesaikan skripsi ini.
2. Ibu Fitri Utaminingrum, Dr. Eng., S.T, M.T. selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
3. Bapak Drs. Marji, M.T. selaku ketua Program Studi Teknik Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Bapak Ir. Sutrisno, M.T. selaku Ketua Program Teknologi Informasi dan Ilmu Komputer.
5. Bapak Wijaya Kurniawan, ST., MT. selaku dosen pembimbing dan penasihat akademik penulis.
6. Segenap Bapak dan Ibu dosen Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya atas segenap ilmu pengetahuan dan perhatian yang diberikan.
7. Saudara seperjuangan “Tekkotech”, Made manik, Faisal Amir, Benny Adiwijaya, Hilmy Firdaus, Aulia Arif, Dicky Sudaryatmo, Rohmatullah, Rizal Zakariah, Rishang Asmadi, Eliana Putri dan Azka Azzahidin. Terimakasih atas segala bantuan dan perjuangan kita selama ini.
8. Partner seperjuangan M. Rizki Firman Habibie dan Rizki Ardiansyah, terimakasih telah menemani dan berjuang bersama menjalani hari-hari dan tugas selama ini dikosan tercinta.
9. Saudaraku tercinta Mohammad Adilla Khoir, mas Amicka, Cak Halim, Cak Audie, Qurotul ayunin, Mazidah, Khoirur Roziq, Durri Ashfa, Sholihul Ibad, Syahrul, mba Sulistin Agustin, mba Anik, mba lailatul fitriah, mba udah, mas Udin. Terimakasih atas semua motivasi dan doa-daoanya.
10. Bapak Ahmad dan ibu Sumiati selaku pemilik kos di Jl. Terusan piranha atas no.44 yang telah memberikan kasih sayang layaknya anak sendiri.
11. Mas Viel, mba Anggi, Ahmad Purwanto, Rizal Pamungkas, Guntur, Fajar dan seluruh rekan semua terima kasih atas bantuan bantuannya selama ini.
12. Abdurrahman Hidayat, Tama, dan rekan seperjuangan skripsi dan rekan riset vision atas segala dukungan dan bantuannya.

13. Sahabat-sahabatku Khalus Dino, M. Faried, Sulistiyono, Bintang Wahyu, Erlita Dian, Widiya Putri, Uwak Ciwik, Eva, Ubed Firmansyah, Lukman Hakim, Faris Syahrizal, Baskoro, Ardi, Kiki, Lilis Suryani, Amrin Rosada, Arie SWK, dan Aditya Sasmito terimakasih atas bantuan dan persahabatan selama ini.
14. M. Wildan Alauddin, Rusyadi, Haris, Bagas, Heri, Galih, Hadi, Syafiq, Anggi, Elis, Winda, Nurlaila, Fierda, dan Rizky Eka atas bantuan dan pertemanan yang tak ternilai selama ini.
15. Segenap staf dan karyawan di Program Teknologi Informasi dan Ilmu Komputer yang telah banyak membantu penulis dalam pelaksanaan penyusunan skripsi ini.
16. Keluarga besar penulis nenek, pakde, budhe, dan semua keluarga besar yang tidak dapat disebutkan satu persatu terimakasih atas dukungan motivasi dan doanya.
17. Seluruh teman-teman SISKOM angkatan 2012 atas dukungan dan kebersamaan dari awal perkuliahan sampai akhir kelulusan.
18. Semua pihak yang telah membantu terselesaikannya skripsi ini yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 25 Januari 2016

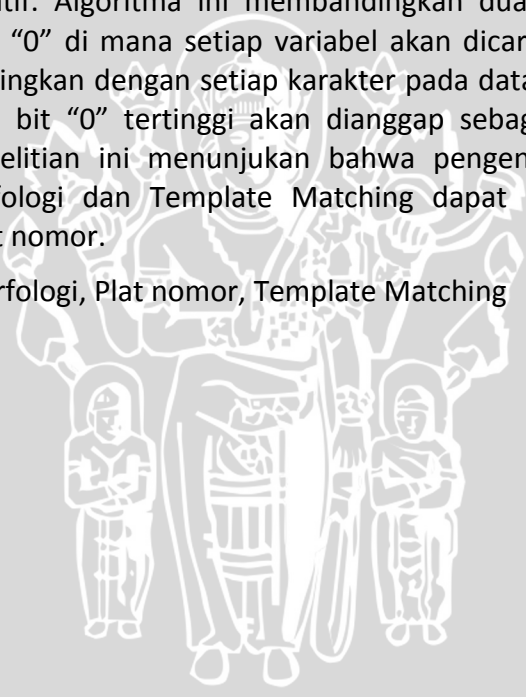
Penulis

rifqieahmad@gmail.com

ABSTRAK

Penelitian tentang pengenalan plat nomor kendaraan banyak digunakan untuk mendapatkan informasi dari plat nomor kendaraan. Ada banyak metode yang digunakan untuk mengenali pola plat nomor kendaraan salah satunya metode Template Matching yang merupakan metode sederhana untuk mengenali pola pada sebuah citra. Metode Template Matching sering digunakan untuk mengenali citra karakter sidik jari, angka dan huruf. Pada project akhir ini dibuat sebuah aplikasi pengenalan plat nomor kendaraan yang digunakan untuk tujuan mempermudah input data plat nomor pada sistem akses parkir. Metode yang digunakan pada aplikasi ini adalah metode Morfologi untuk segmentasi citra karakter dengan menggunakan operasi erosi dan dilasi dan untuk klasifikasi karakter menggunakan metode Template Matching dengan algoritma bit comparing yang digunakan untuk mencari dua variable atau matriks yang sifatnya kuantitatif. Algoritma ini membandingkan dua variabel matriks dengan acuan nilai bit "0" di mana setiap variabel akan dicari nilai bit "0" nya kemudian akan dibandingkan dengan setiap karakter pada data template acuan, yang mempunyai nilai bit "0" tertinggi akan dianggap sebagai karakter yang cocok. Hasil dari penelitian ini menunjukkan bahwa pengenalan plat nomor dengan metode Morfologi dan Template Matching dapat mengenali setiap karakter pada citra plat nomor.

Kata Kunci— Citra, Morfologi, Plat nomor, Template Matching



ABSTRACT

Research on license plate recognition is widely used to obtain information from the vehicle license plate. There are many methods that used to recognize the pattern of license plate. One of the methods used is Template Matching. Template Matching is a simple method to identify patterns in an image. Template Matching method is often used to identify a character of fingerprint image, numbers and letters. In this study we create a license plate recognition application that is used for facilitating purpose in license plate data input on the parking access systems. The method used in this application is a Morphology method for character image segmentation using erosion and dilation operations. For the character classification we use Template Matching method using a bit comparing algorithm that used to searching the two variables or matrix that are quantitative. This algorithm compares two matrix variables with bit "0" value as a reference. Each variables would searching for their bit "0" value and then being compared with each characters in the data template reference, which has a highest bit "0" value will be considered as matching characters. The results from this study showed that the license plate recognition using Template Matching method with Morphology can recognize each character in the image of the license plate.

Keyword: Image, Morphology, License plate, Template Matching

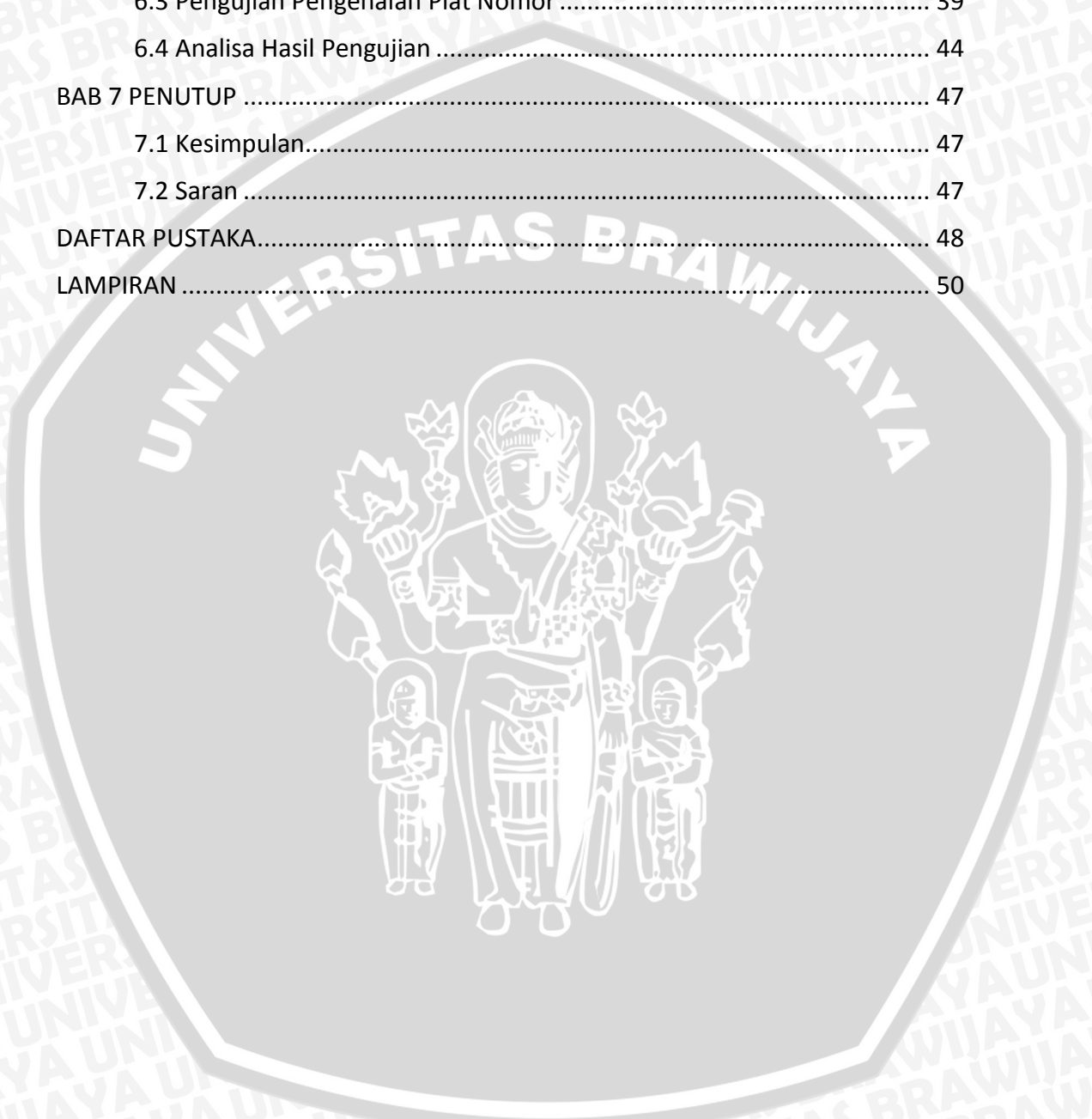


DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
<i>ABSTRACT</i>	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
1.7 Jadwal Penelitian	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka.....	5
2.2 Dasar Teori.....	6
2.2.1 Tanda Nomor Kendaraan Bermotor (TNKB)	6
2.2.2 Pengertian Citra Digital	8
2.2.3 Jenis Citra	8
2.2.4 Thresholding.....	10
2.2.5 Morfologi.....	10
2.2.6 Template Matching.....	12
2.2.7 MATLAB.....	13
2.2.8 Webcam (Web Camera).....	14
BAB 3 METODOLOGI	15
3.1 Alur Metode Penelitian.....	15

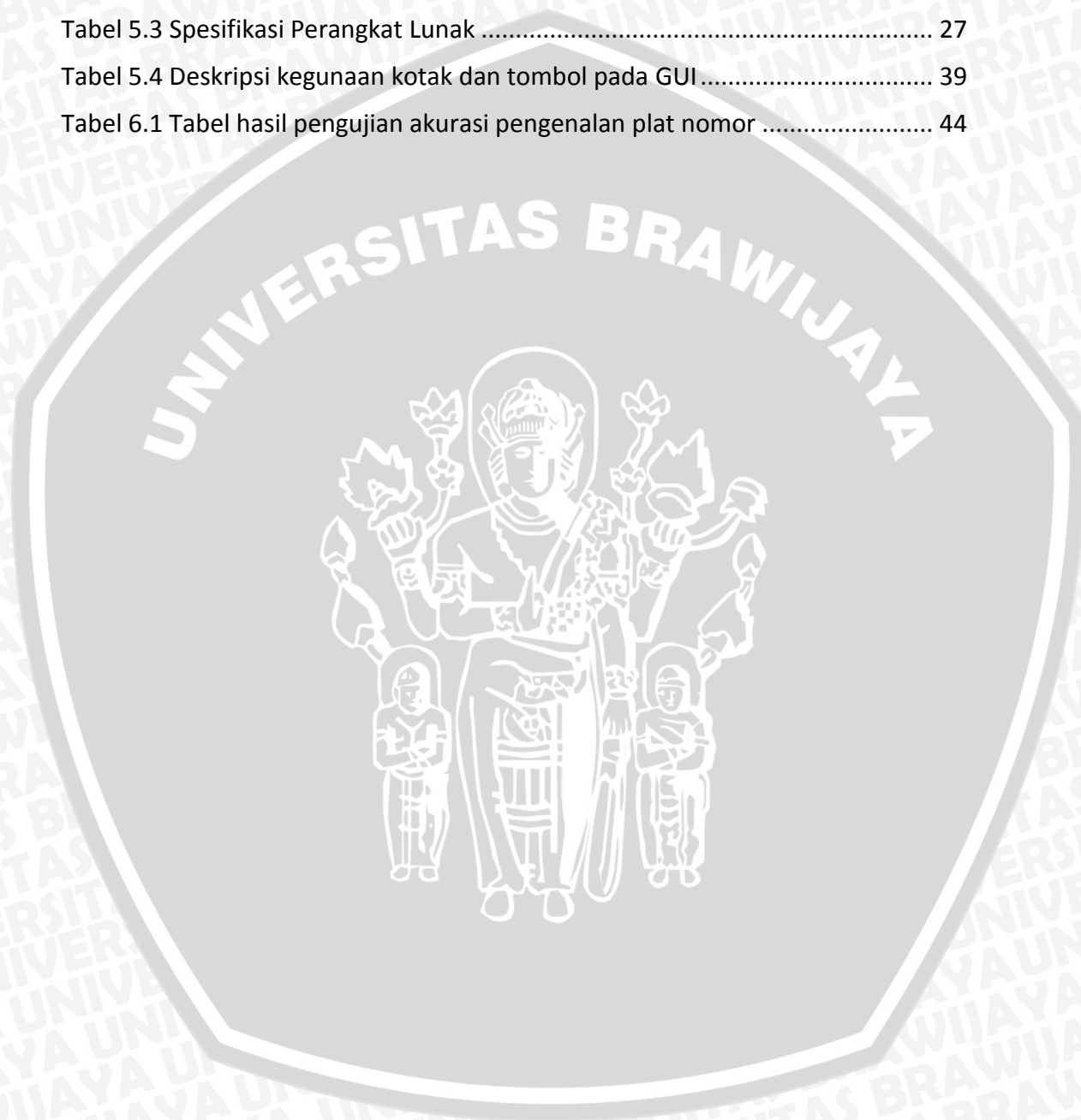
3.2 Studi Literatur	15
3.3 Analisa Kebutuhan	16
3.3.1 Kebutuhan Pengguna	16
3.3.2 Kebutuhan Sistem	16
3.3.3 Kebutuhan Perangkat Keras	16
3.3.4 Kebutuhan Perangkat Lunak	16
3.4 Perancangan Sistem.....	16
3.5 Implementasi Sistem	17
3.6 Pengujian dan Analisis	17
3.7 Kesimpulan.....	17
BAB 4 Rekayasa Persyaratan.....	18
4.1 Pendahuluan	18
4.1.1 Tujuan.....	18
4.1.2 Kegunaan.....	18
4.1.3 Karkteristik Pengguna	18
4.1.4 Lingkungan Operasi.....	18
4.2 Kebutuhan Sistem	19
4.2.1 Kebutuhan Fungsional.....	19
4.2.2 Kebutuhan Non-Fungsional	20
4.2.3 Kebutuhan Antarmuka Perangkat Keras.....	20
4.2.4 Kebutuhan Antarmuka Perangkat Lunak.....	20
4.3 Diagram Blok.....	20
BAB 5 PEMBAHASAN.....	21
5.1 Perancangan Sistem.....	21
5.1.1 Perancangan Perangkat Keras	21
5.1.2 Perancangan Perangkat Lunak.....	21
5.2 Implementasi Sistem	26
5.2.1 Spesifikasi Perangkat Keras	26
5.2.2 Spesifikasi Perangkat Lunak	27
5.2.3 Batasan Implementasi.....	27
5.2.4 Implementasi Perangkat Keras	28
5.2.5 Implementasi Perangkat Lunak.....	28

BAB 6 PENGUJIAN	38
6.1 Pengujian Integrasi Kamera dengan Komputer	38
6.2 Pengujian Integrasi Kamera dengan MATLAB	38
6.3 Pengujian Pengenalan Plat Nomor	39
6.4 Analisa Hasil Pengujian	44
BAB 7 PENUTUP	47
7.1 Kesimpulan.....	47
7.2 Saran	47
DAFTAR PUSTAKA.....	48
LAMPIRAN	50



DAFTAR TABEL

Tabel 5.1 Spesifikasi Webcam Havit HV-V612	26
Tabel 5.2 Spesifikasi Komputer	27
Tabel 5.3 Spesifikasi Perangkat Lunak	27
Tabel 5.4 Deskripsi kegunaan kotak dan tombol pada GUI	39
Tabel 6.1 Tabel hasil pengujian akurasi pengenalan plat nomor	44



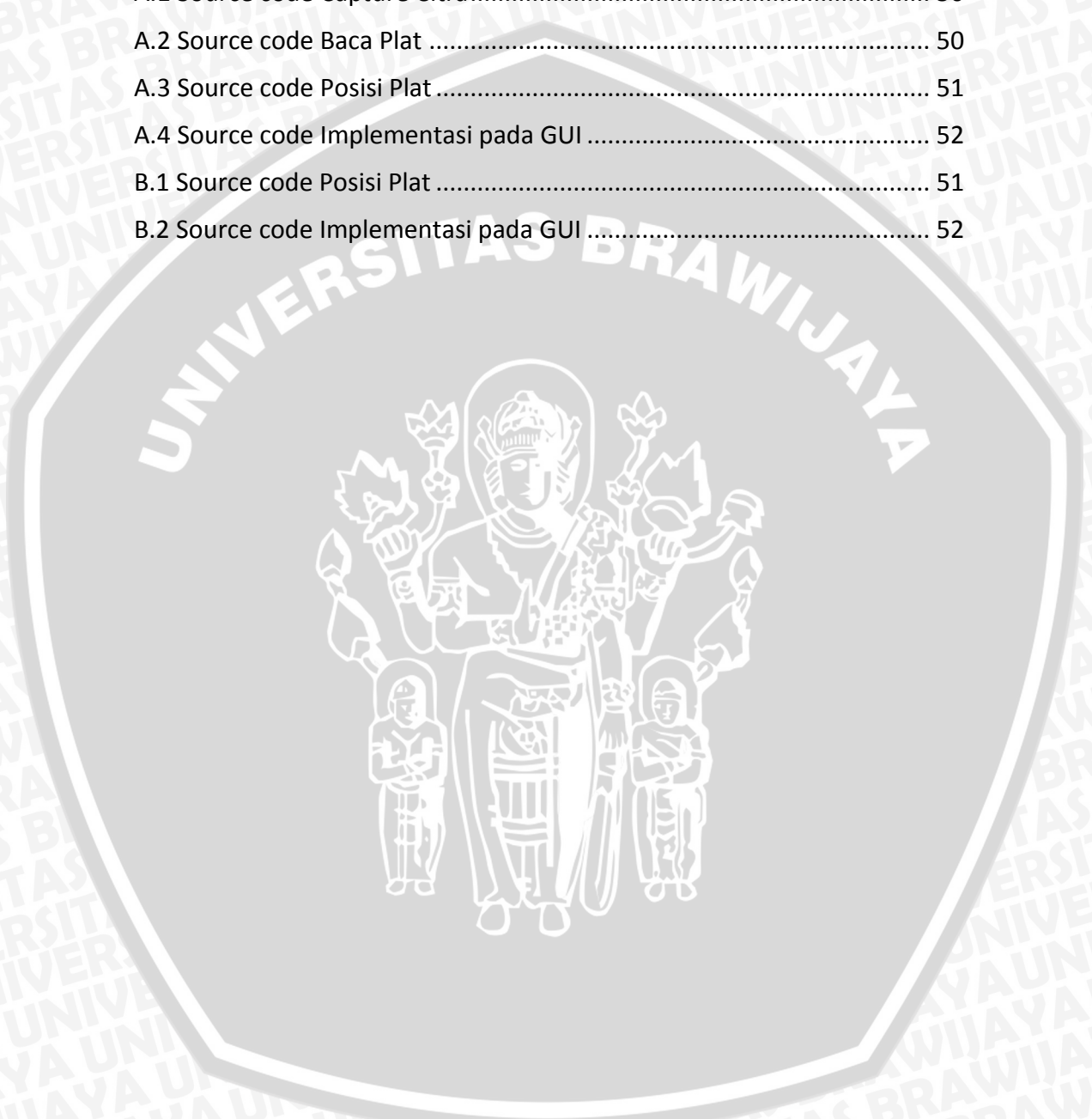
DAFTAR GAMBAR

Gambar 2.1 Diagram blok sistem penelitian Erdinc dan Kursat	5
Gambar 2.2 Perancangan analisi sistem penelitian Taufik	6
Gambar 2.3 Tanda Nomor Kendaraan Bermotor Indonesia	7
Gambar 2.4 Koordinat citra digital	8
Gambar 2.5 Citra Warna RGB	9
Gambar 2.6 Citra <i>Grayscale</i>	9
Gambar 2.7 Citra Biner	9
Gambar 2.8 Beberapa contoh struktur elemen	11
Gambar 2.9 Ilustrasi proses erosi citra	11
Gambar 2.10 Ilustrasi proses dilasi citra	12
Gambar 2.11 Ilustrasi template matching	13
Gambar 2.12 Webcam	14
Gambar 3.1 Diagram alir penelitian	15
Gambar 3.2 Gambaran umum sistem	17
Gambar 4.1 Diagram Blok Sistem	20
Gambar 5.1 Skema Perancangan Perangkat Keras	21
Gambar 5.2 Diagram Alir Umum Sistem	22
Gambar 5.3 Diagram alir proses pembuatan template	23
Gambar 5.4 Karakter template	24
Gambar 5.5 Diagram alir proses pencarian plat nomor	24
Gambar 5.6 Diagram alir proses Pengenalan plat nomor	25
Gambar 5.7 Kamera Webcam Havit HV-V612	27
Gambar 5.8 Perancangan GUI	28
Gambar 5.9 Proses capture pada GUI aplikasi	29
Gambar 5.10 Source code capture citra	30
Gambar 5.11 Citra karakter plat nomor (a), citra setelah dilakukan cropping untuk database template (b)	30
Gambar 5.12 Source code untuk preprocessing	31
Gambar 5.13 Citra RGB (a), citra setelah proses grayscale (b), citra setelah proses thresholding(c)	32

Gambar 5.14 Hasil Proses Morfologi dari citra (a) menjadi citra (b)	33
Gambar 5.15 Source code proses morfologi dan median filter	33
Gambar 5.16 Citra hasil segmentasi	34
Gambar 5.17 Source code program untuk segmentasi	34
Gambar 5.18 Matrix karakter U pada database template	35
Gambar 5.19 Pengurangan matrix, matrix (a) dikurangi matrix (b), hasilnya menjadi matrix (c)	36
Gambar 5.20 Source code program untuk mencari jumlah bit 0	36
Gambar 5.21 Source code mencari hasil 0 terbesar	37
Gambar 5.22 Array penyimpan jumlah bit 0	37
Gambar 6.1 Notifikasi instalasi kamera telah berhasil	38
Gambar 6.2 Kamera telah terhubung dengan komputer	38
Gambar 6.3 Kamera telah terhubung dengan MATLAB	39
Gambar 6.4 Citra capture pada siang hari cerah	39
Gambar 6.5 Citra capture pada sore hari cerah	40
Gambar 6.6 Thresholding plat nomor kondisi terlalu banyak pantulan cahaya ...	40
Gambar 6.7 Thresholding plat nomor tidak terdapat pantulan cahaya	41
Gambar 6.8 Hasil cropping plat citra terdapat banyak noise	41
Gambar 6.9 Hasil cropping plat secara rapih	42
Gambar 6.10 Segmentasi citra yang terdapat banyak noise	42
Gambar 6.11 Segmentasi citra sesuai karakter objek	42
Gambar 6.12 Tampilan GUI aplikasi pengenalan plat nomor	43

DAFTAR LAMPIRAN

LAMPIRAN	50
A.1 Source code Capture Citra.....	50
A.2 Source code Baca Plat	50
A.3 Source code Posisi Plat	51
A.4 Source code Implementasi pada GUI	52
B.1 Source code Posisi Plat	51
B.2 Source code Implementasi pada GUI	52



BAB 1 PENDAHULUAN

Pendahuluan menjelaskan pentingnya penelitian dan tahapan-tahapan pengenalan plat nomor kendaraan menggunakan metode morfologi dan *template matching*.

1.1 Latar belakang

Perkembangan teknologi pada berbagai macam bidang kehidupan manusia semakin berkembang pesat, salah satunya teknologi komputer. Pada era globalisasi sekarang ini peran teknologi komputer tidak dapat dipisahkan dari kehidupan manusia karena telah menjadi suatu kebutuhan yang dapat meningkatkan produktifitas manusia dan dapat membantu dalam berbagai macam bidang pekerjaan manusia. Salah satu kemajuan bidang teknologi tersebut adalah visi komputer, teknologi tersebut digunakan sebagai pengolahan citra digital untuk mendapatkan suatu informasi. Jadi, di sini mesin komputer mampu mengekstrak informasi dari suatu gambar yang telah diambil dari sebuah kamera baik berupa foto maupun video (Munawaroh dan Felix, 2010). Teknologi visi komputer ini mempunyai banyak kegunaan diantaranya sebagai sistem pengendalian proses pada robot, sistem deteksi slot parkir, pengenalan sidik jari, dan pengenalan plat nomor kendaraan. Berbagai macam inovasi tersebut telah banyak diaplikasikan dalam kehidupan sehari-hari dan telah banyak membantu pekerjaan manusia. Melihat keadaan yang terjadi saat ini, khususnya di kota-kota besar yang terdapat banyak perkantoran, pasar, dan tempat wisata akses parkir menjadi hal penting. Banyaknya kendaraan bermotor dapat mempengaruhi sistem akses parkir, apabila terdapat banyak kendaraan yang akan masuk area parkir maka petugas parkir harus bisa cepat dan efisien dalam mendata nomor kendaraan yang masuk area parkir tersebut karena jika tidak bisa cepat mendata maka akan berakibat antrian yang panjang.

Selama ini akses parkir kendaraan bermotor baik roda dua maupun roda empat masih menggunakan sistem parkir manual di mana pencatatan nomor polisi kendaraan bermotor dilakukan dengan cara memasukkan nomor polisi kendaraan ke dalam komputer yang kemudian diproses untuk dicetak atau dengan cara mencatat manual. Dengan sistem seperti ini dapat menimbulkan beberapa permasalahan seperti terjadinya antrian yang panjang dan biasanya terjadi salah input atau salah pencatatan nomor kendaraan. Untuk mengatasi permasalahan tersebut telah berbagai cara dilakukan seperti pemasangan kamera pada area masuk parkir untuk menangkap gambar dari plat nomor kendaraan. Namun, sejauh ini pengambilan gambar tersebut hanya sebatas database berupa gambar saja. Hal itu membuat pencatatan plat nomor juga masih harus dilakukan secara manual dengan memasukan data nomor kendaraan berdasar gambar hasil jepretan kamera menjadi data teks oleh petugas parkir. Dengan memanfaatkan teknologi pengolahan citra digital maka data berupa gambar yang mengandung suatu karakter dapat diekstrak untuk diambil informasinya dan dikonversikan ke dalam data teks.

Pada saat ini telah ada beberapa penelitian mengenai sistem pengenalan plat nomor kendaraan seperti pengenalan plat nomor kendaraan berbasis jaringan syaraf tiruan. Sistem dapat mendeteksi dan mengenali plat nomor melalui tiga tahapan utama yaitu lokalisasi area plat nomor, segmentasi setia karakter dari citra plat nomor dan pengenalan hasil segmentasi karakter. Pada proses pengenalan angka dan huruf dilakukan menggunakan dua jaringan syaraf tiruan terpisah untuk meningkatkan keberhasilan proses pengenalan. Untuk pelatihan jaringan syaraf tiruan algoritma yang dipakai adalah back-propagasi karena dapat menangani masalah pengenalan pola kompleks (Erdinc dan Kursat, 2011). Kemudian penelitian berjudul pengenalan plat nomor otomatis menggunakan principal component analysis (PCA) dan learning vector quatization (LVQ) . Pada penelitian tersebut sistem pengenalan plat nomor menggunakan metode PCA sebagai metode dalam mengekstraksi ciri dan menggunakan jaringan syaraf tiruan learning vector quantization untuk mengklasifikasikan karakter. PCA mampu mereduksi data dan mempertahankan karakteristiknya. Sedangkan JST LVQ mampu melakukan pembelajaran pada lapisan kompetitif yang terawasi (Taufik, 2010). Pada dasarnya pengolahan suatu citra dapat menggunakan banyak metode sehingga dapat terus berkembang sesuai dengan tujuan yang ingin dicapai.

Berdasarkan kajian dari beberapa jurnal penelitian-penelitian di atas penulis mengusulkan pengembangan sistem akases parkir. Sistem ini menggunakan pengolahan citra digital menggunakan metode morfologi dan template matching yang nantinya dapat mempermudah akses parkir dan mencegah terjadinya kesalahan input data. Sistem ini dirancang untuk meringankan beban dari sisi pengemudi dan petugas parkir, dimana memanfaatkan webcam untuk mengambil gambar plat nomor, kemudian gambar tersebut akan diolah dan outputnya akan ditampilkan pada layar monitor dalam bentuk teks. Sehingga informasi mengenai data plat nomor dapat diolah dengan efektif.

1.2 Rumusan masalah

Rumusan masalah dalam penelitian ini yaitu:

1. Bagaimana menerapkan metode morfologi dan template matching ke dalam sistem pengenalan plat nomor kendaraan?
2. Bagaimana tingkat akurasi data hasil pengenalan plat nomor kendaraan menggunakan metode morfologi dan template matching?

1.3 Tujuan

Tujuan yang ingin dicapai dari penelitian ini yaitu:

1. Menerapkan metode morfologi dan template matching ke dalam sistem pengenalan plat nomor kendaraan
2. Menguji tingkat akurasi sistem pengenalan plat nomor kendaraan menggunakan metode morfologi dan template matching

1.4 Manfaat

Dari penelitian ini diharapkan dapat memberikan solusi dalam mengatasi masalah akses parkir, untuk membantu kinerja petugas parkir dalam memasukkan data dalam rangka mencegah terjadinya kemacetan di area masuk parkir akibat terlalu lama memasukkan data nomor kendaraan bermotor dan untuk mencegah terjadinya kesalahan memasukkan data kendaraan. Penulis juga berharap dari hasil penelitian ini dapat digunakan sebagai acuan ilmiah dalam bidang visi komputer dan pengolahan citra digital. Dengan demikian para pembaca dapat memahami dan mengerti perancangan sistem pengenalan plat nomor kendaraan. Kemudian dapat dikembangkan lagi dan diterapkan dalam kehidupan sehari-hari.

1.5 Batasan masalah

Agar pembahasan yang dilakukan pada penelitian ini dapat dilakukan lebih fokus dan terarah, maka perlu ditetapkan beberapa batasan permasalahan, antara lain :

1. Pembahasan terfokus untuk mengenali plat nomor kendaraan pribadi yang telah diatur dalam Undang-Undang No. 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan(UU LLAJ) pasal 68 ayat 2-3, dan spesifikasi oleh Korps Lantas Mabes Polri tahun 2011
2. Plat nomor berlatar belakang warna hitam dengan tulisan berwarna putih
3. Pengujian menggunakan hasil capture citra yang mengandung plat nomor
4. Pengambilan citra menggunakan webcam dengan resolusi 640x480 pixel
5. Proses pengolahan citra menggunakan software MATLAB R2013a
6. Pengambilan citra plat motor dengan jarak 30cm di depan kendaraan dan tinggi 100cm dia atas permukaan tanah dengan sudut kamera 75⁰. Pengambilan citra plat mobil dengan jarak 60cm di depan kendaraan dan tinggi 50cm dia atas permukaan tanah dengan sudut kamera 90⁰
7. Jenis kendaraan yang digunakan pada pengujian adalah motor matic dan motor bebek. Sedangkan untuk mobil digunakan pada mobil jenis sedan dan mobil keluarga

1.6 Sistematika pembahasan

Uraian singkat mengenai struktur penulisan pada masing-masing bab adalah sebagai berikut:

BAB I Pendahuluan

Menjelaskan tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan dari "PENGENALAN PLAT NOMOR KENDARAAN PADA CITRA HASIL CAPTURE KAMERA MENGGUNAKAN METODE MORFOLOGI DAN TEMPLATE MATCHING".

BAB II Landasan Kepustakaan

Pada bab ini menjelaskan tentang landasan teori yang terkait dengan penelitian. Pada bab ini juga dijelaskan tentang penelitian serupa yang pernah dilakukan.

BAB III Metode Penelitian

Membahas tentang alur kerja yang dilakukan dalam penulisan diantaranya studi literatur, analisis kebutuhan sistem, perancangan, implementasi dan pengujian sistem suhu dan kelembaban.

BAB IV Rekayasa Persyaratan

Membahas tentang semua kebutuhan atau *requirements* system, serta kebutuhan fungsional dan non fungsional system.

BAB V Perancangan dan Implementasi

Menjelaskan tentang perancangan system seperti diagram blok system, perancangan database system serta hasil implementasi perancangan system.

BAB VI Pengujian

Pengujian dilakukan apabila system telah dilakukan perancangan dan implementasi, pengujian dilakukan untuk melakukan uji coba system serta melakukan analisis.

BAB VII Penutup

Penutup berisi tentang saran dan kesimpulan dari hasil pembuatan system.

1.7 Jadwal Penelitian

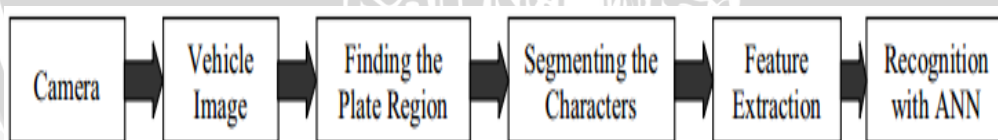
KEGIATAN	Oktober 2015				Nopember 2015				Desember 2015				Januari 2016	
	Minggu				Minggu				Minggu				Minggu	
	1	2	3	4	1	2	3	4	1	2	3	4	1	2
Studi Kepustakaan	■													
Metode penelitian sistem		■												
Analisis kebutuhan sistem			■											
Perancangan sistem				■										
Penulisan Proposal					■									
Pengumpulan kebutuhan sistem						■								
Pengumpulan data							■							
Pembuatan sistem								■						
Pembuatan template database									■					
Pembuatan template GUI										■				
Implementasi sistem											■			
Pengujian sistem												■		
Penulisan Laporan													■	■

BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepustakaan berisi uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan tema, masalah, atau pertanyaan penelitian. Dalam landasan kepustakaan terdapat landasan teori dari berbagai sumber pustaka yang terkait dengan teori dan metode yang digunakan dalam penelitian. Jika dibutuhkan sesuai dengan karakteristik penelitiannya dan syarat kecukupan khusus keminatan tertentu, bisa juga terdapat kajian pustaka yang menjelaskan secara umum penelitian-penelitian terdahulu yang berhubungan dengan topik skripsi dan menunjukkan persamaan dan perbedaan skripsi tersebut terhadap penelitian terdahulu yang dituliskan.

2.1 Tinjauan Pustaka

Penelitian dari Erdinc dan Kursat membahas mengenai pengenalan plat nomor kendaraan menggunakan metode jaringan syaraf tiruan. Terdapat tiga proses utama dalam membangun sistem ini yaitu proses lokalisasi plat nomor merupakan proses untuk mendapatkan kandidat plat nomor kendaraan dari citra yang mengandung plat nomor. Kemudian proses segmentasi karakter merupakan proses untuk memisahkan karakter dengan latar belakang dengan melakukan sub proses contrast extension, median filtering, dan blob coloring method. Tahap terakhir yaitu pengenalan karakter dari hasil segmentasi dengan menggunakan metode jaringan syaraf tiruan dan untuk pelatihannya menggunakan algoritma back-propagation (Erdinc dan Kursat, 2011). Diagram blok sistem dapat dilihat pada Gambar 2.1.

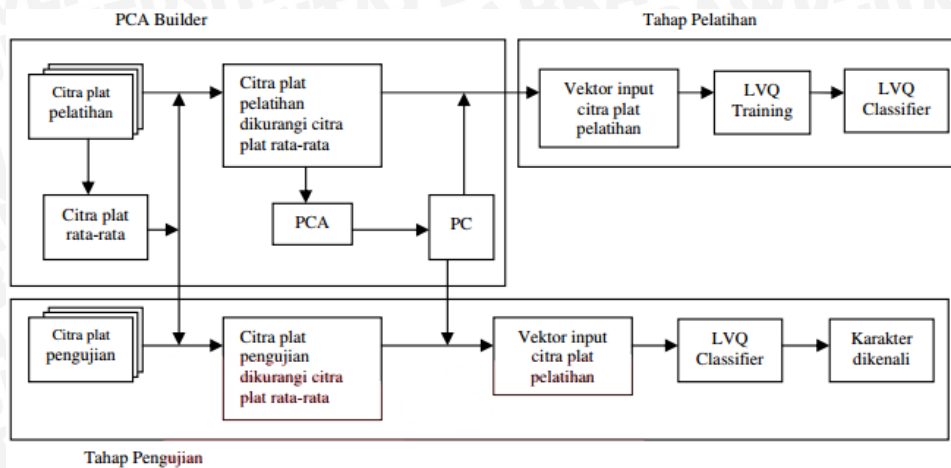


Gambar 2.1 Diagram blok sistem penelitian Erdinc

Sumber: (Erdinc dan Kursat, 2011)

Dalam penelitian ini angka dan huruf diklasifikasikan dengan menggunakan dua jaringan syaraf tiruan terpisah untuk mencegah pengenalan angka dan huruf yang hampir sama seperti "0" dengan "O", "2" dengan "Z", dan "8" dengan "B".

Pada penelitian selanjutnya dari Taufik, Budi, dan Retno tentang pengenalan plat nomor otomatis menggunakan principal component analysis (PCA) dan learning vector quantization (LVQ). Citra yang diambil untuk pengujian adalah plat nomor kendaraan mobil pribadi Negara Indoneisa dan hanya menggunakan plat nomornya saja dengan latar belakang putih dengan ukuran 1600x1200 pixel, kemudian citra masing-masing karakter diolah berukuran 50x50 pixel.



Gambar 2.2 Perancangan analisi sistem penelitian Taufik

Sumber: (Taufik, et al., 2010)

Pada Gambar 2.2 dapat dilihat perancangan sistem dari citra input akan dilakukan beberapa proses utama, untuk mengekstraksi ciri menggunakan metode PCA yang menghasilkan output berupa eigenvector yang sudah terurut berdasarkan eigenvalue atau disebut juga Principal Component (PC). Setiap image dikalikan dengan PC sehingga mendapatkan ekstraksi dengan ukuran yang lebih kecil untuk selanjutnya dijadikan input klasifikasi JST. Kemudian untuk melakukan klasifikasi menggunakan Learning Vector Quantization (LVQ) yang merupakan metode klasifikasi pola yang masing-masing unitnya mewakili kategori atau kelas tertentu. Hasil output klasifikasi LVQ akan mengenali karakter pada plat nomor.

2.2 Dasar Teori

Pada bagian dasar teori menjelaskan tentang beberapa definisi tentang teori yang digunakan dalam pengolahan citra digital, undang-undang mengenai spesifikasi plat nomor resmi Indonesia, komponen yang digunakan untuk membangun sistem, dan aplikasi pendukung pemroses citra digital.

2.2.1 Tanda Nomor Kendaraan Bermotor (TNKB)

Tanda Nomor Kendaraan Bermotor (TNKB) atau pada masyarakat umumnya disebut Plat Nomor Kendaraan adalah tanda registrasi identifikasi kendaraan bermotor yang berfungsi sebagai bukti legitimasi pengoperasian kendaraan bermotor berupa pelat atau berbahan lain dengan spesifikasi tertentu yang diterbitkan Polri dan berisikan kode wilayah, nomor registrasi, serta masa berlaku dan dipasang pada kendaraan bermotor (Peraturan KAPOLRI, 2012). Berdasarkan aturan Korps Lantas Mabes Polri tahun 2011 mengatur spesifikasi tanda nomor kendaraan bermotor di Indonesia sebagai berikut:



Gambar 2.3 Tanda Nomor Kendaraan Bermotor Indonesia

Sumber: (www.oto.detik.com, 2011)

Spesifikasi Teknis Tanda Nomor Kendaraan Bermotor:

- 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),
- 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 275×110 mm, sedangkan untuk kendaraan bermotor roda 4 atau lebih adalah 430×135 mm,
- Terdapat cetakan garis lurus pembatas lebar 5 mm di antara ruang nomor polisi dengan ruang angka masa berlaku,
- Pada sudut kanan atas dan sudut kiri bawah terdapat tanda khusus (security mark) cetakan lambang Polisi Lalu Lintas; sedangkan pada sisi sebelah kanan dan sisi sebelah kiri ada tanda khusus cetakan "KORLANTAS POLRI" (Korps Lalu Lintas Kepolisian RI) yang merupakan hak paten pembuatan TNKB oleh Polri.

Warna Tanda Nomor Kendaraan Bermotor:

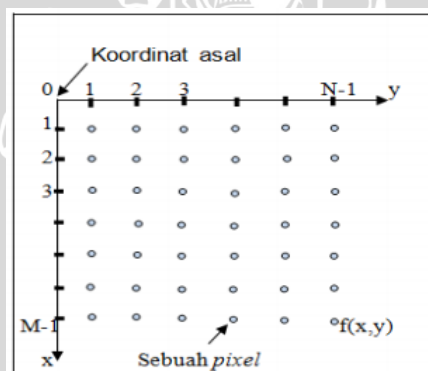
- Kendaraan bermotor perseorangan dan sewa: warna dasar hitam dengan tulisan berwarna putih.
- Kendaraan bermotor umum: warna dasar kuning dengan tulisan berwarna hitam.
- Kendaraan bermotor milik pemerintah: warna dasar merah dengan tulisan berwarna putih.
- Kendaraan bermotor korps diplomatik negara 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.

- f. 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.

2.2.2 Pengertian Citra Digital

Secara umum pengolahan citra digital menunjuk pada pemrosesan gambar 2 dimensi menggunakan komputer. Dalam konteks yang lebih luas, pengolahan citra digital mengacu pada pemrosesan setiap data 2 dimensi. Citra digital merupakan sebuah larik (array) yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu (Putra, 2010).

Suatu citra dapat didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spesial, dan amplitudo f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan citra pada titik tersebut. Apabila nilai x , y , dan nilai amplitudo f secara keseluruhan berhingga (finite) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital. Koordinat citra digital terhadap sumbu (x,y) dapat dilihat pada Gambar 2.4.



Gambar 2.4 Koordinat citra digital

Sumber: (Putra, 2010)

2.2.3 Jenis Citra

Suatu pixel memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimal. Jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Namun secara umum jangkauannya adalah 0 – 255. Citra dengan penggambaran seperti ini digolongkan ke dalam citra integer. Berikut adalah jenis-jenis citra berdasarkan nilai pixelnya:

- 1) Citra warna (RGB)

Citra berwarna, atau biasa dinamakan citra RGB, merupakan jenis citra yang menyajikan warna dalam bentuk komponen R (merah), G (hijau), dan B (biru) (Rizal, 2013).



Gambar 2.5 Citra Warna RGB

Sumber: (www.wallpaperhere.com, 2015)

2) Citra grayscale

Citra grayscale merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixelnya. Nilai tersebut dapat digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, abu, dan putih. Tingkat keabuan disini warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih (Putra, 2010).



Gambar 2.6 Citra *Grayscale*

Sumber: (www.wallpaperhere.com, 2015)

3) Citra biner (binary image)

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai pixel yaitu hitam dan putih. Citra biner juga disebut sebagai citra B&W (black and white) atau citra monokrom. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap pixel dari citra biner (Putra, 2010).



Gambar 2.7 Citra Biner

Sumber: (www.wallpaperhere.com, 2015)

2.2.4 Thresholding

Metode thresholding (pengambangan) merupakan suatu metode yang sering digunakan dalam pengolahan citra digital atau image processing. Thresholding citra adalah suatu metode yang digunakan untuk memisahkan antara obyek dan backgroundnya (Fauzi, 2010).

Secara umum proses thresholding terhadap citra grayscale bertujuan untuk menghasilkan citra biner, secara matematis dapat ditulis sebagai berikut:

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) \geq T \\ 0 & \text{if } f(x, y) < T \end{cases} \dots\dots\dots(2.1)$$

Dengan $g(x,y)$ adalah citra biner dari citra grayscale $f(x,y)$, dan T menyatakan nilai threshold (ambang). Nilai T memegang peranan yang sangat penting dalam proses pengambangan. Kualitas hasil citra biner sangat bergantung pada nilai T yang digunakan (Kumaseh, 2013). Terdapat dua jenis pengambangan, yaitu:

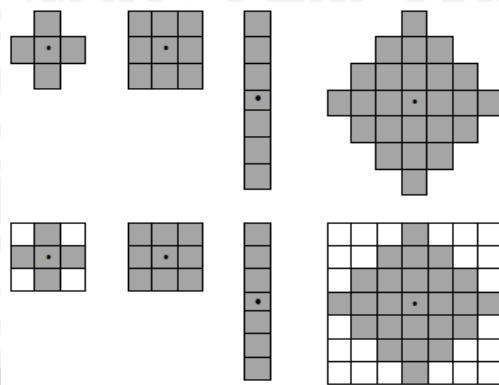
- 1) Pengambangan global (global thresholding).
 Pada pengambangan global, seluruh pixel citra dikonversikan menjadi hitam atau putih dengan satu nilai ambang T . kemungkinan besar pada pengambangan global akan banyak informasi yang hilang karena hanya menggunakan satu nilai T untuk keseluruhan pixel.
- 2) Pengambangan local (local threshloding)
 Pada pengambangan lokal, suatu citra dibagi menjadi blok-blok kecil dan kemudian dilakukan pengambangan lokal pada setiap blok dengan nilai T yang ditentukan.

2.2.5 Morfologi

Morfologi digital adalah kenyataan bahwa pada sebuah citra digital mengandung serangkaian piksel-piksel yang membentuk sekumpulan data dua dimensi. Persamaan matematika tertentu pada serangkaian piksel dapat digunakan untuk meningkatkan aspek dari bentuk dan struktur sehingga dapat lebih muda dikenali (Setiawan, 2014).

Operasi morfologi menggunakan dua input himpunan yaitu suatu citra (pada umumnya citra biner) dan suatu kernel. Khusus dalam morfologi, istilah kernel biasa disebut dengan SE atau structuring elements (elemen pembentuk struktur). SE merupakan suatu matix dan umumnya berukuran kecil (Rizal, 2015).





Gambar 2.8 Beberapa contoh struktur elemen

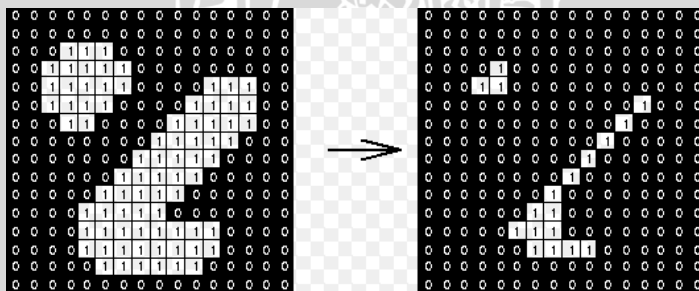
Pada morfologi citra terdapat dua operasi dasar, yaitu:

1) Erosi

Erosi merupakan proses mengecilkan atau menipiskan obyek citra biner. Dengan melakukan perpaduan sebuah citra asli dengan struktur elemen. Operasi erosi umumnya digunakan untuk mengikis atau menghilangkan piksel atau memperkecil ukuran suatu objek citra. Operasi erosi dapat dilakukan pada citra biner dengan strel tertentu disesuaikan dengan hasil citra yang ingin dicapai.

$$E(A, B) = A \ominus B = \{Z : B_z \subset X\} \dots \dots \dots (2.2)$$

Pada persamaan 2.2 maksudnya adalah dilakukan erosi citra A oleh struktur elemen (strel) B yang terdiri atas semua titik $z = (x,y)$ di mana (B_z) adalah bagian dari himpunan A. Pada proses erosi dilakukan dengan cara proses piksel per piksel yang termasuk anggota A. sehingga dapat menghasilkan citra yang berbeda yang lebih kecil dari citra aslinya



Gambar 2.9 Ilustrasi proses erosi citra dengan SE 3x3

Sumber: (Rizal, 2015)

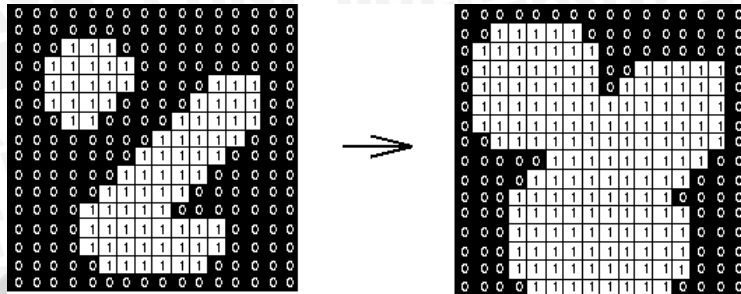
2) Dilasi

Dilasi merupakan operasi kebalikan dari erosi operasi dilasi adalah proses menebalkan suatu objek dengan cara menggabungkan titik piksel latar menjadi bagian suatu objek sesuai dengan struktur elemen yang digunakan. Biasanya operasi dilasi digunakan untuk menambal bagian kecil citra yang tidak terhubung agar citra tampak lebih jelas.

$$D(A, B) = A \oplus B = \{Z : B_z \cap A \neq \phi\} \dots \dots \dots (2.3)$$



Pada persamaan 2.3 merupakan dilasi citra A oleh struktur elemen B di mana A dan B adalah himpunan dari Z, untuk setiap area di luar tepi citra A akan dilakukan proses pergeseran dan kemudian menggabungkan seluh hasilnya dengan struktur elemen B.



Gambar 2.10 Ilustrasi proses dilasi citra dengan SE 3x3

Sumber: (Rizal, 2015)

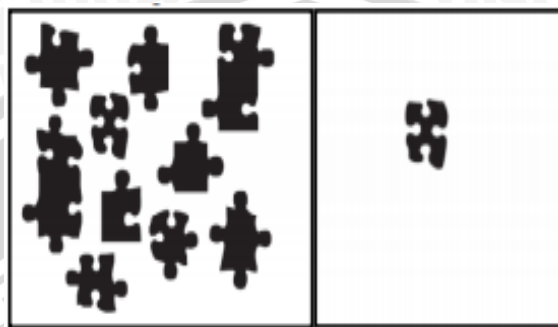
2.2.6 Template Matching

Template Matching merupakan proses pengenalan citra yang cukup sederhana dalam pengolahan citra digital yaitu dengan cara menemukan bagian-bagian terkecil dari gambar yang cocok dengan template gambar. Template matching merupakan salah satu ide yang digunakan untuk menjelaskan bagaimana otak kita mengenali kembali bentuk-bentuk atau pola-pola. Template dalam konteks rekognisi pola menunjuk pada konstruk internal yang jika cocok (match) dengan stimulus penginderaan mengantar pada rekognisi suatu objek. Atau pengenalan pola terjadi jika terjadi kesesuaian antara stimulus indera dengan bentuk mental internal. Gagasan ini mendukung bahwa sejumlah besar template telah tercipta melalui pengalaman hidup kita. Tiap-tiap template berhubungan dengan suatu makna tertentu.

Template Matching dapat dibagi antara dua pendekatan, yaitu :

- 1) Pendekatan berbasis fitur, menggunakan fitur pencarian dan template gambar seperti tepi atau sudut, sebagai pembanding pengukuran matrik untuk menemukan lokasi template matching yang terbaik di sumber gambar. Pendekatan dapat membuktikan lebih baik, jika template gambar memiliki fitur yang kuat jika pencocokan di pencarian gambar bisa diubah dengan cara tertentu. Karena pendekatan ini tidak mempertimbangkan keseluruhan dari template gambar, komputasi dapat lebih efisien ketika bekerja dengan sumber gambar beresolusi lebih besar, sebagai pendekatan alternatif, berbasis template, mungkin memerlukan pencarian titik – titik yang berpotensi untuk menentukan lokasi pencocokan yang terbaik.
- 2) Pendekatan berbasis template, pendekatan ini lebih efektif digunakan apabila template tanpa fitur yang kuat, atau ketika sebagian besar template gambar merupakan gambar yang cocok. Karena berbasis

template, template matching berpotensi memerlukan sampling dari sejumlah besar poin, untuk mengurangi jumlah sampling poin dengan mengurangi resolusi pencarian dan template gambar oleh faktor yang sama dan melakukan operasi pada perampingan gambar yang dihasilkan (multiresolusi, atau piramida, pengolahan citra), menyediakan pencarian titik data dalam pencarian gambar sehingga template tidak harus mempunyai pencarian titik data, atau kombinasi keduanya.



Gambar 2.11 Ilustrasi template matching

Sumber: (Bahri dan Maliki, 2012)

Gambar 2.11 bagian kiri merupakan citra yang mengandung objek yang sama dengan objek pada template yang ada di sebelah kanan. Template diposisikan pada citra yang akan dibandingkan dan dihitung derajat kesesuaian pola pada citra masukan dengan pola pada citra template. Tingkat kesesuaian antara citra masukan dan citra template bisa dihitung berdasarkan nilai eror terkecil (Sofian dan Irfan, 2012).

2.2.7 MATLAB

MATLAB® (Matrix Laboratory) adalah salah satu software atau aplikasi yang digunakan untuk pengolahan citra digital yang dikembangkan oleh Mathworks, Inc. MATLAB merupakan sebuah lingkungan komputasi yang interaktif dengan bahasa tingkat tinggi untuk perhitungan numerik, visualisasi, dan pemrograman. MATLAB juga dapat digunakan untuk menganalisis data, mengembangkan algoritma, dan membuat aplikasi. Dengan bahasa, tools, dan fungsi-fungsi built-in pada MATLAB, memungkinkan pengguna untuk dapat mengeksplorasi beberapa pendekatan untuk mencapai sebuah solusi lebih cepat dibandingkan dengan spreadsheets atau bahasa pemrograman tradisional, seperti C/C++ atau Java (Mathworks, 2014).

Fitur utama pada MATLAB:

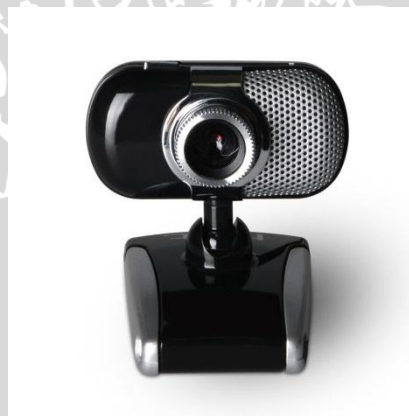
- 1) Bahasa tingkat tinggi untuk perhitungan numerik, visualisasi, dan pengembangan aplikasi (application development)
- 2) Lingkungan komputasi yang interaktif untuk eksplorasi, desain, dan pemecahan masalah
- 3) Fungsi matematika untuk aljabar linier, statistik, analisis fourier, penyaringan (filtering), optimasi, integrasi numerik (numerical

integration), memecahkan persamaan diferensial biasa (ordinary differential equations)

- 4) Built-in grafis untuk memvisualisasikan data dan tools untuk membuat costum plots
- 5) Pengembangan tools untuk meningkatkan kualitas kode (code quality), pemeliharaan (maintainability), dan memaksimalkan kinerja (maximizing performance)
- 6) Fungsi-fungsi untuk mengintegrasikan algoritma berbasis MATLAB dengan aplikasi eksternal seperti Microsoft® Excel®, dan bahasa pemrograman seperti C, Java, .NET

2.2.8 Webcam (Web Camera)

Pada dasarnya webcam (web camera) dan PC camera memiliki pengertian yang sama, yakni sebuah kamera video digital dengan bentuk yang relatif kecil yang dapat dihubungkan dengan komputer melalui port USB ataupun port COM. Pada umumnya tipe kamera ini tidak membutuhkan tempat penyimpanan data, karena data hasil penangkapan (capture) atau perekaman (record) yang didapat langsung ditransfer ke komputer. Namun istilah webcam mengarah pada jenis kamera yang digunakan untuk kebutuhan layanan berbasis web (Rizal, 2014).



Gambar 2.12 Webcam

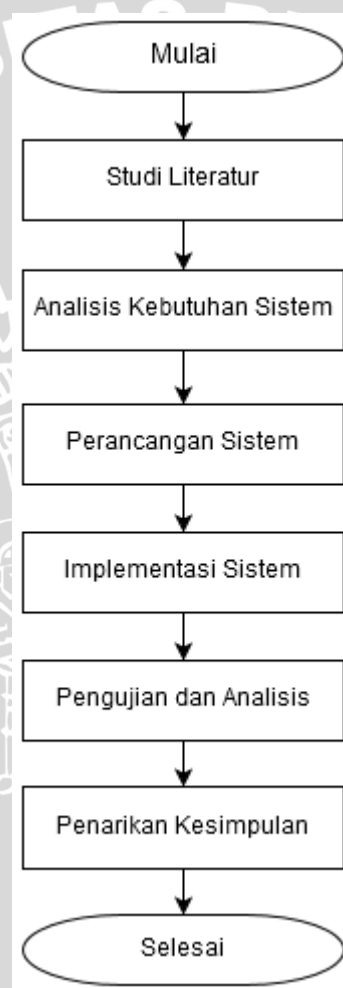
Sumber: (<http://www.amazon.com>, 2015)

BAB 3 METODOLOGI

Bab ini menjelaskan metode yang digunakan dalam melakukan penelitian tentang sistem pengenalan pola plat nomor kendaraan pada citra hasil capture kamera menggunakan metode morfologi dan template matching. Tipe penelitian ini adalah implementatif dengan pendekatan perancangan (*design*) melalui proses yang utuh mulai dari analisis perancangan hingga pengujiannya.

3.1 Alur Metode Penelitian

Alur metode penelitian yang dilakukan secara umum dapat dilihat dari diagram alir pada gambar 3.1.



Gambar 3.1 Diagram alir penelitian

3.2 Studi Literatur

Studi literatur dilakukan untuk mencari referensi dari penelitian-penelitian sebelumnya sebagai pendukung penelitian yang akan dilakukan. Dari penelitian yang dibuat referensi dilakukan penyusunan teori-teori yang akan digunakan dalam penelitian dan perancangan sistem. Teori-teori yang dapat dimanfaatkan

pada penelitian ini meliputi teori mengenai pengolahan citra digital, metode morfologi citra, dan metode klasifikasi template matching.

3.3 Analisa Kebutuhan

Analisa kebutuhan ditujukan untuk melakukan analisis pada beberapa kebutuhan yang diperlukan sistem pada penelitian. Analisa kebutuhan pada penelitian dibagi menjadi 2, yaitu kebutuhan Fungsional dan Non-fungsional.

3.3.1 Kebutuhan Pengguna

Kebutuhan Pengguna pada penelitian ini adalah sebagai berikut :

- Sistem mampu menampilkan citra yang mengandung plat nomor
- Sistem mampu menampilkan hasil pengenalan plat nomor kendaraan

3.3.2 Kebutuhan Sistem

Kebutuhan sistem yang diperlukan untuk membangun pada sistem ini adalah sebagai berikut :

- Sistem dapat mengcapture citra yang mengandung plat nomor kendaraan
- Sistem dapat mendeteksi plat nomor kendaraan
- Sistem dapat mengenali pola plat nomor kendaraan

3.3.3 Kebutuhan Perangkat Keras

Kebutuhan Perangkat Keras yang diperlukan untuk membangun sistem ini adalah sebagai berikut :

- Perangkat Komputer/ Laptop
- Kamera Webcam
- LCD Monitor

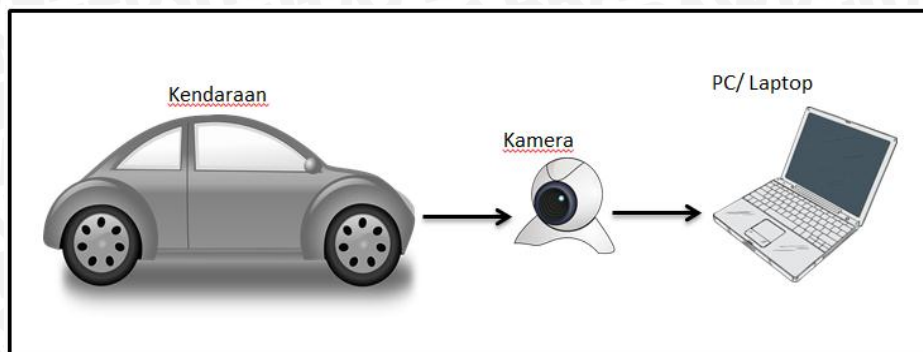
3.3.4 Kebutuhan Perangkat Lunak

Kebutuhan Perangkat Lunak yang diperlukan untuk membangun sistem ini adalah sebagai berikut :

- MATLAB R2013a
- OS Windows 7
- Image processing library
- Bahasa Pemrograman Matlab

3.4 Perancangan Sistem

Perancangan sistem dilakukan apabila semua kebutuhan yang diperlukan oleh sistem sudah terpenuhi. Perancangan dilakukan sesuai dengan gambaran umum sistem yang dapat dilihat pada Gambar 3.2.



Gambar 3.2 Gambaran umum sistem

Dalam sistem ini pengolahan data menggunakan Komputer atau Laptop. Citra kendaraan dicapture menggunakan kamera webcam dengan jarak, ketinggian dan sudut yang telah ditentukan. Hasil capture dikirim ke Komputer/ Laptop untuk dilakukan proses pengolahan citra merubah data citra digital menjadi data alfanumerik. Kemudian hasil pengolahan akan ditampilkan pada monitor dalam bentuk GUI dan menampilkan data berupa citra plat nomor dan data teks hasil pengenalan citra.

3.5 Implementasi Sistem

Pada tahap implementasi sistem, hal pertama yang dilakukan adalah pengerjaan perangkat keras. Setelah semua kebutuhan perangkat keras terpenuhi selanjutnya adalah pengerjaan perangkat lunak untuk mengolah data. Software dan library yang dibutuhkan diinstall pada Komputer/Laptop kemudian dilakukan pembuatan aplikasi berbasis GUI untuk dapat mengcapture citra yang mengandung plat nomor dengan kamera. Pada tahap akhir dilakukan integrasi antara perangkat keras dengan perangkat lunak dan dilakukan konfigurasi sistem sehingga bisa saling terhubung.

3.6 Pengujian dan Analisis

Pada tahapan pengujian sistem, parameter yang dilakukan disesuaikan dengan perancangan sistem. Beberapa perancangan untuk skenario pengujian adalah sebagai berikut :

1. Pengujian akurasi pendeteksian plat nomor kendaraan dari hasil capture kamera.
2. Pengujian terhadap akurasi ketepatan pengenalan pola karakter pada plat nomor kendaraan
3. Pengujian terhadap peformansi yang mampu dilakukan sistem dalam mengolah data

3.7 Kesimpulan

Kesimpulan sistem pada penelitian ini didapat apabila seluruh pengujian sistem dari perancangan telah dilakukan. Selanjutnya hasil dari pengujian sistem secara garis besar dijadikan sebagai kesimpulan.

BAB 4 REKAYASA PERSYARATAN

Dalam bab ini menjelaskan persyaratan minimal yang harus dipenuhi untuk perancangan hingga implementasi. Dengan harapan perancangan dan implementasi sistem pengenalan plat nomor ini bias berjalan dengan baik.

4.1 Pendahuluan

Pada penelitian mengenai sitem pengenalan plat nomor ini telah ditentukan tujuan dari sistem, kegunaan dari dibuatnya sistem, karakteristik pengguna serta lingkungan pengoprasiaanya.

4.1.1 Tujuan

Perancangan sistem ini bertujuan untuk mendeteksi dan mengenali plat nomor kendaraan pada akses masuk area parkir menggunakan pengolahan citra digital. Pada sistem ini dilakukan capture citra yang mengandung plat nomor menggunakan kamera webcam, dengan posisi kamera yang diletakkan pada jarak , tinggi, dan sudut yang telah ditentukan didepan kendaraan. Data citra hasil capture kamera dikirim ke komputer/laptop untuk dilakukan proses pengolahan citra. Pada proses pengolahan citra dilakukan beberapa proses secara garis besar terdapat tiga proses utama yaitu pendeteksian plat nomor kendaraan, segmentasi karakter pada plat nomor, dan klasifikasi atau pengenalan plat nomor kendaraan juga bisa diartikan sistem melakukan proses konversi citra digital menjadi alfanumerik. Dari hasil pengenalan citra ditampilkan pada layar monitor berupa data teks.

4.1.2 Kegunaan

Sistem ini berguna untuk mengatasi masalah input data plat nomor kendaraan bermotor pada akses masuk area parkir. Dengan menggunakan pengolahan citra digital hasil capture plat nomor akan diolah pada sistem aplikasi untuk mendapatkan hasil berupa data teks. Dengan menggunakan sistem ini akan mempermudah pekerjaan petugas parkir dan mempersingkat waktu input data kendaraan sehingga dapat mencegah antrian masuk area parkir juga data yang diinput lebih terjamin.

4.1.3 Karkteristik Pengguna

Pengguna sistem bertindak sebagai pemberi input untuk memulai berjalannya sitem aplikasi pengenalan plat nomor. Pengguna dapat sebagi pemonitor masuknya data hasil capture dari kamera setelah program dijalankan dan dapat melihat hasil konversi dari data citra digital menjadi data teks.

4.1.4 Lingkungan Operasi

Pada persyaratan kebutuhan lingkungan yang mendukung kebutuhan sistem yaitu:

1. Pengambilan citra sampel plat mobil dengan jarak 60cm di depan kendaraan dan tinggi 50cm dengan sudut 90° .
2. Pengambilan citra sampel plat motor dengan jarak 30cm di depan kendaraan dan tinggi 100cm dengan sudut kamera 75° .
3. Pengambilan sampel dilakukan pada sore hari dengan kondisi cuaca cerah

4.2 Kebutuhan Sistem

Analisis kebutuhan bertujuan untuk menggambarkan kebutuhan-kebutuhan dari sistem. Kebutuhan fungsional dan non-fungsional akan dianalisis sesuai dengan kebutuhan sistem sehingga akan mempermudah dalam mendesain dan mengimplementasikan sistem.

4.2.1 Kebutuhan Fungsional

- a. Komputer/Laptop dapat menerima data citra hasil capture kamera.
Pada fungsi ini komputer/laptop dapat menerima setiap data hasil capture kamera yang mengandung plat nomor kendaraan dan digunakan sebagai input sistem.
- b. Sistem dapat mendeteksi plat nomor kendaraan
Sebelum masuk tahap pengenalan diperlukan tahap deteksi plat nomor, ini berpengaruh pada tahap selanjutnya karena jika salah mendeteksi plat nomor maka hasil output tidak akan sesuai dengan citra plat nomor.
- c. Sistem dapat melakukan proses morfologi pada citra input.
Setiap citra hasil input tidak selalu mempunyai komposisi citra yang sempurna dengan menggunakan metode morfologi bagian-bagian dari karakter yang diperlukan dapat direayasa untuk mendapatkan citra yang sesuai dan lebih mudah untuk proses klasifikasi nantinya.
- d. Sistem mampu melakukan segmentasi setiap karakter
Proses segmentasi dilakukan untuk memisahkan setiap karakter juga memisahkan dengan latar belakang citra. Didapatkan karakter yang siap untuk dilakukan klasifikasi.
- e. Dapat melakukan proses klasifikasi menggunakan metode template matching
Pada citra hasil segmentasi dilakukan proses pengenalan atau klasifikasi citra dengan membandingkan dengan citra template.
- f. Dapat menampilkan hasil klasifikasi pada GUI
Setelah didapatkan hasil pengenalan plat nomor selanjutnya hasil pengenalan berupa teks ditampilkan pada GUI.
- g. Template citra huruf dan angka.
Digunakan pada tahap klasifikasi. Semakin banyak template cita huruf dan angka maka semakin baik untuk pengklasifikasian citra.

4.2.2 Kebutuhan Non-Fungsional

Pada kebutuhan non-fungsional sebaiknya menggunakan kamera dengan resolusi 640x480 karena dapat mempercepat proses pengolahan citra digital dan pengenalan plat nomor kendaraan. Jika resolusi kamera semakin besar maka proses pengolahan citra digital akan semakin lambat.

4.2.3 Kebutuhan Antarmuka Perangkat Keras

Antarmuka perangkat keras pada sistem ini yaitu:

1. Kamera webcam
2. Komputer/Laptop

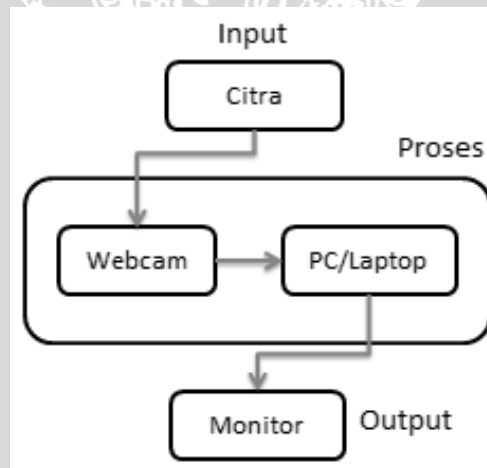
4.2.4 Kebutuhan Antarmuka Perangkat Lunak

Antarmuka perangkat lunak pada sistem ini yaitu:

1. OS Windows 7
2. MATLAB R2013a
3. GUI MATLAB

4.3 Diagram Blok

Diagram blok sistem terdiri dari citra yang mengandung plat nomor sebagai input, webcam untuk mengcapture citra yang mengandung plat nomor, PC/Laptop sebagai pemroses data citra digital, dan monitor untuk menampilkan hasil pengenalan citra plat nomor kendaraan.



Gambar 4.1. Diagram Blok Sistem

Berdasarkan Gambar 4.1 perancangan sistem yang dibuat dapat dijelaskan sebagai berikut:

- a. Input berupa citra yang mengandung plat nomor kendaraan.
- b. Webcam sebagai sensor untuk mengcapture citra
- c. PC/Laptop sebagai pengolah data citra digital untuk dirubah ke data teks.
- d. Hasil pengenalan ditampilkan pada monitor dalam bentuk GUI berupa hasil capture kamera dan hasil pengenalan citra data teks.

BAB 5 PEMBAHASAN

Bab ini menjelaskan tentang perancangan sistem pengenalan plat nomor kendaraan pada citra hasil capture kamera menggunakan metode morfologi dan template matching, serta implementasi sistem dari hasil perancangan berupa implementasi perangkat keras dan perangkat lunak.

5.1 Perancangan Sistem

Pada tahapan perancangan sistem menjelaskan tentang konsep dasar dan gambaran umum, diagram blok, perancangan perangkat keras, perancangan perangkat lunak sistem.

5.1.1 Perancangan Perangkat Keras

Perancangan perangkat keras terdiri dari beberapa komponen, kamera sebagai sensor untuk mengcapture citra yang mengandung plat nomor, PC/Laptop sebagai pengolah data citra digital dan LCD Monitor sebagai output untuk menampilkan hasil proses. Skema perancangan perangkat keras dapat dilihat pada Gambar 5.1.

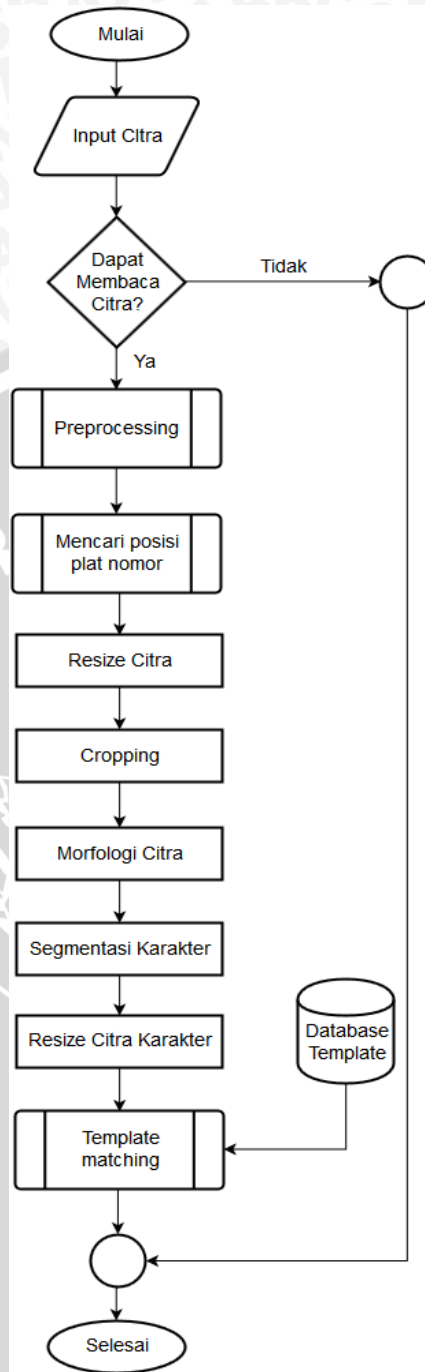


Gambar 5.1 Skema Perancangan Perangkat Keras

Pada penerapan sistem pengenalan plat nomor kendaraan, kamera dihubungkan dengan laptop menggunakan kabel USB. Output ditampilkan pada layar laptop atau LCD Monitor.

5.1.2 Perancangan Perangkat Lunak

Pada perancangan perangkat lunak membahas tentang diagram alir yang akan digunakan pada sistem agar dapat mendeteksi dan mengenali pola plat nomor kendaraan menggunakan pengolahan citra digital. Perancangan diagram alir umum pengolahan citra digital secara garis besar pada sistem ini dapat dilihat pada Gambar 5.2.



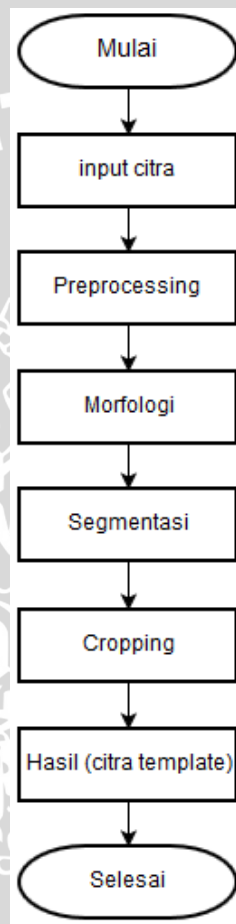
Gambar 5.2 Diagram Alir Umum Sistem

Berdasarkan diagram alir di atas, tahap pertama yang dilakukan saat program mulai dijalankan adalah mengcapture citra yang mengandung plat nomor kendaraan. Kemudian mulai dilakukan preprocessing untuk merubah citra warna RGB menjadi citra keabuan kemudian menjadi citra biner. Dilakukan proses pencarian plat nomor untuk dipisahkan dari objek lain agar lebih mudah dalam pengenalan karakter plat nomor. Morfologi untuk memperbaiki citra agar sesuai dengan karakter angka dan huruf sebenarnya. Setelah itu dilakukan proses segmentasi karakter atau pemisahan karakter untuk membedakan karakter satu

dengan yang lainnya. Pada proses template matching hasil dari segmentasi diklasifikasikan dibandingkan dengan template database untuk mendapatkan hasil yang sesuai.

5.1.2.1 Perancangan Proses Pembuatan Template

Template di sini berfungsi sebagai data pengklasifikasian yang digunakan untuk mencocokkan karakter input. Database template terdiri dari karakter citra angka 0-9 dan huruf A-Z. Setiap karakter terdapat 3 macam template, jadi jumlah total 108 template citra karakter dan ukuran citra template 26x18 pixel.



Gambar 5.3 Diagram alir proses pembuatan template

Pada Gambar 5.3 diagram alir di atas proses dimulai dari input berupa citra warna RGB dilakukan preprocessing untuk merubah ke dalam bentuk citra biner. Kemudian dilakukan proses morfologi opening dengan operasi erosi dan selanjutnya operasi dilasi bertujuan untuk memperbaiki kondisi citra. Pada tahap selanjutnya untuk melakukan pemisahan setiap karakter dilakukan proses segmentasi dengan menggunakan boundingbox. Setelah setiap karakter di segmentasi dilakukan proses cropping objek untuk memotong bagian objek sesuai boundingbox dan menghasilkan citra karakter yang disimpan pada database template.

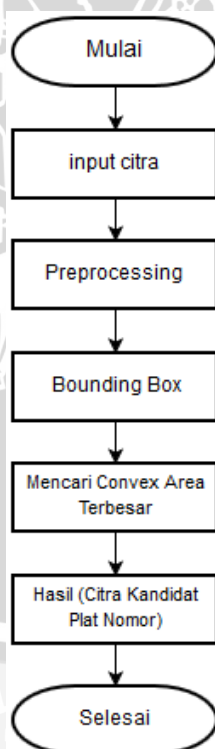


Gambar 5.4 Karakter template

Pada Gambar 5.4 merupakan citra plat nomor yang telah diproses menjadi data citra template yang kemudian digunakan pada proses template matching. Proses pembuatan template diambil dari beberapa sampel plat nomor yang terdiri dari 36 karakter huruf A-Z dan angka 0-9 dikali 3 setiap karakter menjadi 108 karakter template database.

5.1.2.2 Perancangan Proses Pencarian Kandidat Plat Nomor

Pencarian kandidat plat nomor dilakukan untuk memisahkan plat nomor dengan latar belakang atau objek lainnya. Setelah hasil kandidat plat nomor ditemukan dapat dilakukan proses segmentasi karakter.



Gambar 5.5 Diagram alir proses pencarian kandidat plat nomor

Dari Gambar 5.5 diagram di atas untuk mencari kandidat plat nomor, proses pertama adalah input citra kemudian dilakukan preprocessing merubah

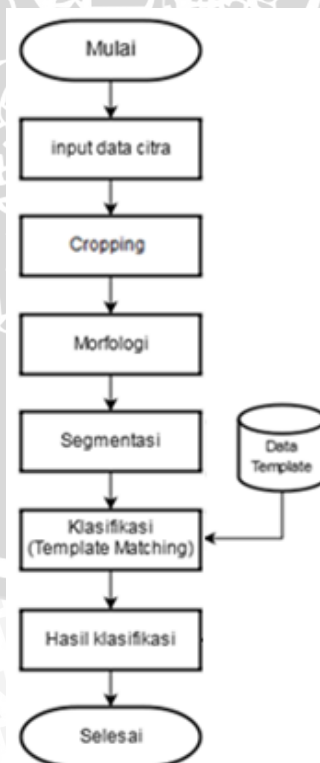
citra RGB menjadi citra biner sekaligus dilakukan proses pengambangan citra (thresholding) dengan persamaan berikut:

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) \geq 190 \\ 0 & \text{if } f(x, y) < 190 \end{cases} \dots\dots\dots 5.1$$

Apabila nilai citra lebih dari sama dengan 190 maka akan dirubah menjadi citra berwarna putih bernilai biner (1). Jika nilai citra kurang dari 190 maka bernilai biner (0) menjadi citra berwarna hitam. Kemudian dilakukan proses boundingbox yaitu mengkotak atau membatasi setiap citra bernilai 1 yang terhubung, dari proses ini dicarilah kandidat plat nomor dengan memanfaatkan fungsi regionprops dengan atribut ConvexArea yaitu suatu atribut di dalam regionprops yang dapat menghitung luas area citra di dalam boundingbox. Dari luas area yang didapat dipilih area yang terluas karena pada citra hasil capture diasumsikan area kotak plat nomor adalah area yang terbesar.

5.1.2.3 Perancangan Proses Pengenalan Plat Nomor

Proses pengenalan plat nomor menggunakan metode template matching, di mana citra input yang telah diproses akan dicocokkan dengan citra template mencari tingkat kecocokan yang paling tinggi. Pada penelitian ini algoritma template matching yang dipakai adalah perbandingan bit "0". Jadi untuk menentukan kandidat karakter yang paling cocok dengan cara membandingkan jumlah bit "0" pada karakter input dengan setiap karakter template jika didapatkan jumlah bit "0" terbesar pada perbandingan suatu karakter maka karakter itulah yang akan dipilih dan dianggap paling cocok.



Gambar 5.6 Diagram alir proses Pengenalan plat nomor



Pada Gambar 5.6 proses pengenalan plat nomor dimulai dari input data citra plat nomor dilakukan proses cropping untuk mendapatkan citra angka dan huruf plat nomornya saja. Kemudian dilakukan operasi morfologi untuk meningkatkan aspek bentuk dan struktur karakter dan dilakukan proses segmentasi untuk membedakan setiap karakter. Dari hasil segmentasi dilakukan pengenalan dengan metode template matching menggunakan pendekatan bit. Jadi, matrix citra akan dijadikan satu kolom kemudian dilakukan pengurangan matrik dengan citra template setelah itu akan dicari nilai bit "0" sebagai acuan untuk dibandingkan hasilnya. Dengan jumlah bit "0" yang terbesar menjadikan karakter tersebut dianggap paling cocok dalam proses klasifikasinya.

5.2 Implementasi Sistem

Pada tahapan implementasi sistem menjelaskan tentang spesifikasi perangkat keras dan spesifikasi perangkat lunak, batasan-batasan implementasi, pembuatan GUI untuk kebutuhan interface dalam memudahkan pengguna dalam menjalankan program, serta implementasi algoritma pengolahan citra.

5.2.1 Spesifikasi Perangkat Keras

Pada implementasi sistem terdapat dua perangkat keras yang digunakan, yaitu Komputer/Laptop dan Kamera Webcam. Perangkat keras dihubungkan melalui kabel USB.

1. Kamera Webcam

Kamera Webcam yang digunakan untuk mendapatkan gambar lalu lintas dijalan menggunakan kamera Webcam Havit HV-V612. Spesifikasi kamera webcam adalah sebagai berikut :

Tabel 5.1 Spesifikasi Webcam Havit HV-V612

Webcam Havit HV-V612	
<i>Interface to PC</i>	<i>High Speed USB 2.0 (Compatible with USB 1.1)</i>
<i>Resolution</i>	<i>0.48 Megapixels (640x480 pixels)</i>
<i>Frame Rate</i>	<i>800 x 480 pixels (VGA), 600 x 640, 288 x 320, 320 x 240 (CIF)</i>
<i>Built-In Microphone</i>	<i>Available with 3.5mm Audio Port</i>
<i>Operating System Support</i>	<i>PC with Windows® (2000 / XP /Vista / 7)</i>
<i>Dimension (WHD)</i>	<i>54.1(W) x 54(H) x 54.9(D) mm, Cable Length : Approx. 140cm</i>
<i>Weight</i>	<i>95 gram</i>



Gambar 5.7 Kamera Webcam Havit HV-V612

2. Komputer

Komputer berfungsi sebagai media pengolahan input (citra) menjadi sebuah output yang ditampilkan pada monitor. Spesifikasi komputer yang digunakan dijelaskan pada tabel 5.2.

Tabel 5.2 Spesifikasi Komputer

Acer Aspire 4741	
<i>Processor Onboard</i>	<i>Intel Core i3-2330M CPU @2.27GHz 2.27 GHz</i>
<i>Standard Memory</i>	<i>2 GB DDR3</i>
<i>Display</i>	<i>Intel HD Graphic</i>
<i>Hard Drive Type</i>	<i>320 GB Serial ATA 5400 RPM</i>

5.2.2 Spesifikasi Perangkat Lunak

Pengembangan sistem ini menggunakan sebuah perangkat lunak yang dipaparkan pada tabel 5.3.

Tabel 5.3 Spesifikasi Perangkat Lunak

Acer 4741	
<i>Operating System</i>	<i>Windows 7 Ultimate 32-bit</i>
<i>Programming Tool</i>	<i>MATLAB R2013a</i>

5.2.3 Batasan Implementasi

Beberapa batasan implementasi sistem akan dijelaskan sebagai berikut:

1. Pengambilan citra dilakukan dengan menggunakan kamera Webcam Havit HV-V612
2. Citra yang diolah merupakan citra yang telah diambil dari lokasi tempat parkir pada sore hari



3. Aplikasi pemrograman untuk pengolahan citra menggunakan software MATLAB R2013a
4. Penempatan dan sudut kamera telah ditentukan yaitu, kamera di depan kendaraan dengan jarak untuk mobil 60 cm dan tinggi 50cm dengan sudut 90° . Untuk motor jarak kendaraan dengan kamera 30cm, tinggi 100cm dengan sudut 75°
5. Sampel motor yang digunakan adalah motor matic dan motor bebek standar belum dimodifikasi. Sampel mobil yang digunakan adalah mobil sedan dan mobil keluarga
6. Plat nomor dengan latar belakang warna hitam dan tulisan berwarna putih
7. Kondisi plat nomor standar belum dimodifikasi

5.2.4 Implementasi Perangkat Keras

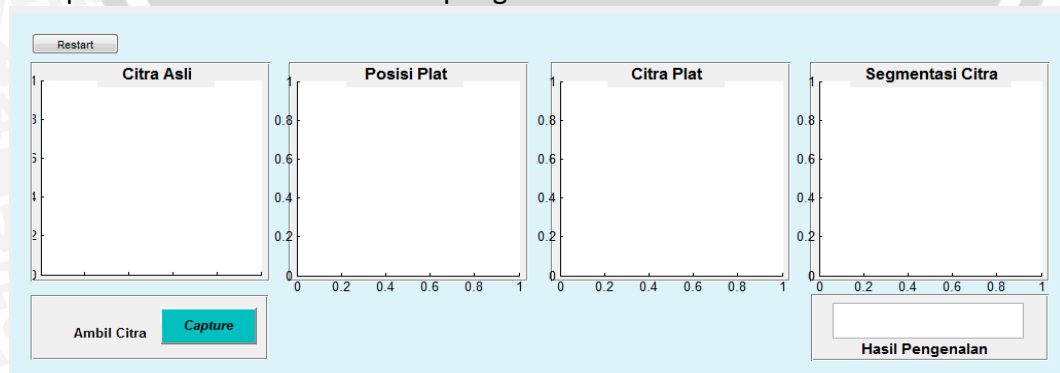
Pada implementasi perangkat keras sesuai dengan perancangan sistem terdiri dari kamera webcam untuk mengcapture citra sebagai input dan Komputer sebagai pengolah data citra digital. Digunakan kamera webcam dengan resolusi 640x480 agar proses pengolahan citra lebih cepat. Hasil capture kamera webcame dapat disimpan dan ditampilkan pada komputer dan LCD monitor. Menggunakan kabel USB untuk menghubungkan kamera dengan komputer.

5.2.5 Implementasi Perangkat Lunak

Pada implementasi perangkat lunak terdapat beberapa tahapan proses dimulai dari proses pembuatan program yang dapat melakukan pengolahan citra. Langkah pertama dilakukan pembuatan template dari citra data latih dilakukan proses pendeteksian karakter kemudian dilakukan *cropping* hasilnya dijadikan sebagai database template. Untuk pengujian dilakukan dengan beberapa tahapan penting yaitu proses pencarian obyek plat nomor, proses segmentasi karakter, dan proses pengklasifikasian karakter.

5.2.5.1 Graphical User Interface (GUI)

Graphical user interface (GUI) merupakan suatu tampilan dari program yang dibuat untuk mempermudah user dalam menggunakan sistem aplikasi. Pada gambar 5.8 ditampilkan rancangan GUI yang terdiri dari empat kolom axes dua push button dan kolom hasil pengenalan.



Gambar 5.8 Perancangan GUI

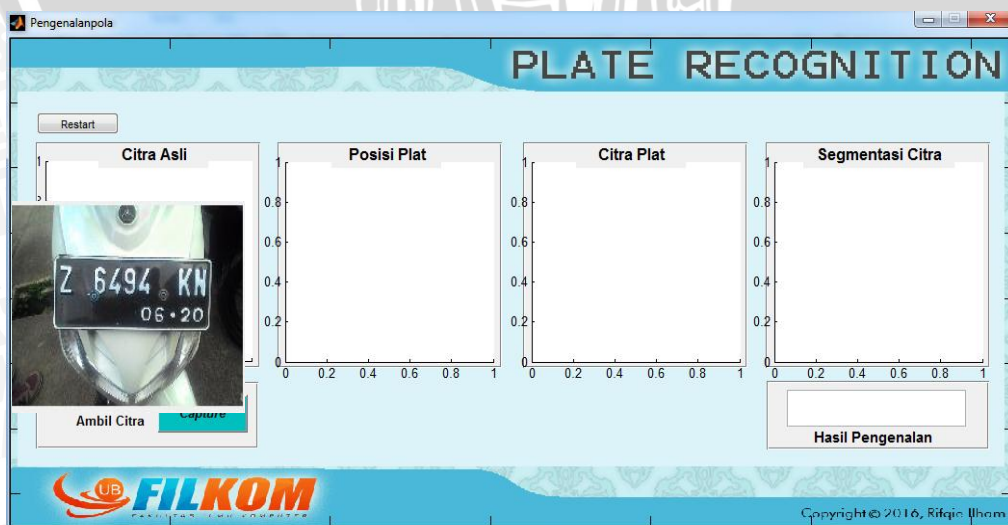
Tujuan dibuatnya GUI ini untuk mempermudah user dalam menjalankan program yang telah dibuat pada sistem tanpa harus melihat banyak baris program. Deskripsi kegunaan dari atribut kotak dan tombol pada gambar 5.7 dapat dilihat pada Table 5.4

Tabel 5.4 Deskripsi kegunaan kotak dan tombol pada GUI

Nama Atribut	Fungsi
Citra Asli	Menampilkan citra hasil capture kamera untuk diproses
Posisi Plat	Menampilkan di mana letak posisi obyek plat nomor
Citra Plat	Menampilkan citra obyek plat nomornya saja yang telah dicrop
Segmentasi Citra	Menampilkan Hasil proses citra berupa segmentasi citra setiap karakter yang ditemukan
Ambil Citra (capture)	Memberikan perintah untuk mengcapture citra yang mengandung plat nomor
Hasil Pengenalan	Menampilkan hasil pengenalan berupa teks (angka dan huruf)

5.2.5.2 Mengcapture Citra yang Mengandung Plat Nomor

Proses awal yang dilakukan pada sistem pengenalan plat nomor adalah melakukan proses *capture* atau pengambilan citra yang mengandung plat nomor kendaraan dengan jarak yang telah ditentukan. Pada gambar 5.9 ditunjukkan hasil kamera yang mengambil citra dari plat nomor kendaraan roda dua. Perintah *capture* akan dilakukan ketika menekan tombol *capture* untuk kemudian data citra akan ditampilkan citra aslinya dan dilakukan proses pengolahan citra hasil *capture* tersebut.



Gambar 5.9 Proses capture pada GUI aplikasi

Untuk mendapatkan hasil capture yang maksimal pengambilan citra plat nomor kendaraan dilakukan pada area yang mempunyai penerangan yang cukup atau pada area *outdoor* dengan kondisi cuaca yang cerah dan tidak terlalu banyak cahaya yang memantul pada plat nomor kendaraan.

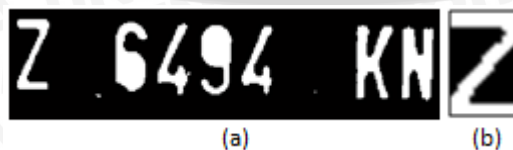
```
guidata(hObject, handles);
axes(handles.axes1);
vid= videoinput('winvideo',2,'YUY2_640x480');
% preview(vid);
hImage = imagesc(zeros(400,300), 'Parent', handles.axes1);
preview(vid, hImage);
% set(vid,'ReturnedColorSpace','rgb');
set(vid, 'FramesPerTrigger', Inf);
set(vid, 'ReturnedColorspace', 'rgb');
%snapshot citra
capture= getsnapshot(vid);
% captureRGB= ycbcr2rgb(capture);
guidata(hObject, handles);
%memilih axes sebagai letak gambar yang dicapture
axes(handles.axes1);
imshow(capture);
```

Gambar 5.10 Source code capture citra

Pada source code yang ditunjukkan gambar 5.10 inisial variable 'vid' sebagai fungsi untuk memanggil kamera. Format ukuran citra yang dihasilkan kamera havit yang terdeteksi pada matlab adalah 'YUY_640x480'. Variabel 'hImage' digunakan untuk menampilkan area untuk mengcapture kamera dengan ukuran 400x300 pixel. Perintah getsnapshot digunakan untuk mengcapture citra yang disimpan pada variable capture dan hasilnya diletakkan pada axes1 untuk ditampilkan.

5.2.5.3 Pembuatan Template

Template yang digunakan pada sistem ini adalah menggunakan data citra karakter hasil proses pengolahan citra data latih yang berupa citra *capture* dari 30 plat nomor kendaraan. Pada setiap plat nomor yang diolah akan diambil karakter yang diperlukan yang terdiri dari karakter huruf A-Z dan angka 0-9 yang berjumlah 36 sampel citra karakter. Dari ke-36 karakter tersebut dilakukan *editing* menggunakan software paint untuk menambah variasi setiap karakter, untuk setiap karakter akan diberikan masing-masing sebanyak tiga variasi. Total keseluruhan databasetemplate yang digunakan pada sistem ini berjumlah 108 data citra karakter.



Gambar 5.11 Citra karakter plat nomor (a), citra setelah dilakukan cropping untuk database template (b)

5.2.5.4 Implementasi *Preprocessing*

Pada tahap *preprocessing* dilakukan proses merubah citra RGB menjadi citra biner dan merubah format yang sebelumnya uint8 menjadi double agar dapat dilakukan proses operasi matematika. Terdapat beberapa sub proses pada tahap ini diantaranya yaitu:

1. Citra RGB dirubah dahulu menjadi citra keabuan (grayscale) dengan perintah *rgb2gray* pada *MATLAB*. Dari proses ini didapatkan citra dengan skala hitam dan putih dan terdiri dari satu layer.
2. Dari citra keabuan dirubah menjadi citra black and white (biner) dengan melakukan proses *thresholding*. Pada proses *thresholding* ini dilakukan pengambangan citra dengan batas ambang 190. Jadi apabila nilai suatu piksel adalah kurang dari 190 maka akan dirubah menjadi piksel bernilai 0 atau warna hitam. Apabila nilai piksel citra bernilai lebih dari 190 maka piksel tersebut akan dirubah menjadi bernilai 1 atau berwarna putih. Dari proses pengambangan akan didapatkan citra yang terlihat obyek plat nomor dan obyek lainnya yang memenuhi persyaratan batas ambang *thresholding*.

```
function crop=posisi(citra);  
f2=rgb2gray(citra);  
[jmlB jmlK]=size(f2);  
for x=1:jmlB  
    for y=1:jmlK  
        if f2(x,y) > 190  
            g(x,y)=255;  
        else  
            g(x,y)=0;  
        end  
    end  
end  
final=bwareaopen(g,400);
```

Gambar 5.12 *Source code* untuk *preprocessing*

Pada *source code* yang ditunjukkan Gambar 5.12 merupakan *source code* fungsi dari beberapa proses yang terdiri dari input yang memanggil data citra dari fungsi *posisi*. Dilakukan proses merubah citra menjadi skala keabuan dengan perintah *rgb2gray* dan disimpan pada variable *f2*. Kemudian dilakukan proses *thresholding* dengan batas ambang lebih dari 190 nilai citra akan menjadi warna putih dan nilai citra yang kurang dari 190 akan menjadi citra berwarna hitam. Dengan perintah *bwareaopen* dilakukan binerisasi dari citra sebelumnya dan hanya mengambil luas area yang bernilai 400 piksel (area yang hanya bernilai 1 saja) untuk disimpan pada variable *final*.



Gambar 5.13 citra RGB (a), citra setelah proses *grayscale* (b), citra setelah proses *thresholding*(c)

Pada Gambar 5.13 dapat dilihat perubahan dari setiap proses pengolahan citra. Gambar (a) merupakan citra berwarna atau citra asli dari hasil capture, citra (b) hasil dari proses *grayscale* yang nampak berwarna keabuan, dan citra (c) adalah hasil dari proses *thresholding* yang terlihat berwarna hitam dan putih atau citra biner.

5.2.5.5 Mencari Posisi Plat Nomor

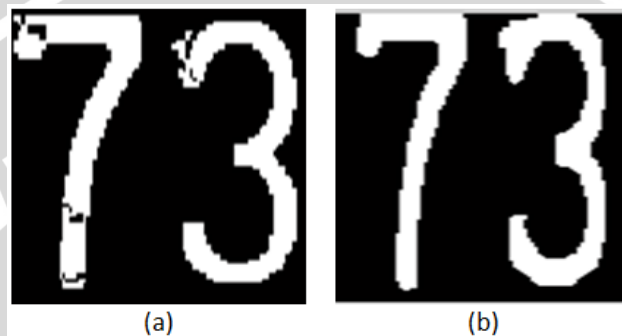
Pada proses pencarian plat nomor bertujuan untuk mendapatkan obyek plat nomornya saja agar lebih muda dalam mendeteksi setiap karakter angka dan huruf dari plat nomor tersebut. Dalam mencari plat nomor ada beberapa tahapan proses yang dilakukan yaitu:

1. Melakukan proses merubah citra warna menjadi citra biner seperti yang dilakukan pada poin 5.2.5.4
2. Melakukan proses *bounding box*, pada tahap ini bounding box dicari dengan cara melakukan pencarian setiap piksel yang berwarna putih atau bernilai 1 yang saling terhubung, dimana setiap ditemukan awal piksel putih maka itu akan mengawali proses penandaan titik awal bounding box. Selanjutnya dilakukan proses pencarian piksel putih pada koordinat selanjutnya sampai ditemukan piksel yang bernilai 0 maka proses penandaan bounding box akan berhenti dan akan dilakukan plot antar poin sehingga dihasilkan suatu box yang telah mengelilingi obyek karakter berwarna putih tersebut. Hal itu dilakukan sampai seluruh kumpulan piksel berwarna putih sudah ditandai dan dibounding box. Pada program MATLAB dapat digunakan fungsi "*regionprops*" dengan atribut "*BoundingBox*" untuk dapat mendapatkan informasi dan penandaan *bounding box*.
3. Mencari area terbesar dari *bounding box*, untuk mencari posisi dari plat nomor telah ditetapkan bahwa dengan batasan pengambilan gambar yang sesuai batasan masalah akan didapatkan plat nomor dengan area terluas pada citra gambar. Jadi dari citra yang telah dilakukan bounding box akan dipilih citra yang area box nya terbesar dan itu dianggap sebagai obyek plat nomor kendaraan yang telah terdeteksi.

- Setelah proses bounding box dilakukan dan telah dipilih area terbesar dari box langkah selanjutnya adalah memotong citra obyek yang dianggap plat nomor kendaraan.

5.2.5.6 Implementasi Proses Morfologi Citra

Proses morfologi citra dilakukan bertujuan untuk memperkuat hasil dari citra yang telah diolah menjadi output yang lebih akurat, oleh karena itu dilakukan proses *opening*, dimana menggunakan dua proses morfologi yakni terlebih dahulu dilakukan proses erosi kemudian proses dilasi. Proses ini bertujuan untuk mengilangkan/meminimalkan titik-titik kecil pada citra.



Gambar 5.14 Hasil Proses Morfologi dari citra (a) menjadi citra (b)

Dari Gambar 5.14 dapat dilihat citra yang sebelumnya terdapat bercak-bercak dapat diminimalisasi hingga mendapatkan hasil yang lebih baik. Karena nanti akan berpengaruh pada hasil klasifikasinya.

```
% median filtering untuk menghapus noise
g=medfilt2(croprxy,[3 3]);
% figure(4), imshow(g);
% logic 0/1
B=logical(g);
%proses opening
mor=imerode(B,strel('square',3));
mor2=imdilate(mor,strel('rectangle',[3 4]));
fil=imfill(mor,'holes');
```

Gambar 5.15 Source code proses morfologi dan median filter

Pada source code di atas tahap pertama melakukan proses median filter untuk menghapus noise atau objek kecil yang tidak dibutuhkan pada citra lalu dilakukan proses logical untuk merubah format sebelumnya double menjadi logical, matrix citra hanya berisi 0 dan 1 hasilnya disimpan pada variable 'B'.

Pada tahap ini dilakukan operasi morfologi opening yang pertama adalah citra B dierosi oleh struktur elemen berbentuk square 3 maksudnya mask berbentuk matrix $[1 \ 1 \ 1; 1 \ 1 \ 1; 1 \ 1 \ 1]$ hasilnya disimpan pada variable mor, selanjutnya dilakukan dilasi citra mor oleh struktur elemen rectangle $[3 \ 4]$ atau bisa dituliskan $[1 \ 1 \ 1 \ 1; 1 \ 1 \ 1 \ 1; 1 \ 1 \ 1 \ 1]$ hasilnya disimpan pada variable mor2. Hasil tadi dilakukan proses filling objek yaitu mengisi bagian objek yang terdapat lubang sampai akhirnya tertutupi.

5.2.5.7 Implementasi Proses Segmentasi Karakter

Pada tahap ini dilakukan proses pemisahan karakter dengan menganalisa untuk mencari kumpulan pixel putih atau pixel yang bernilai 1 dengan area minimal 400 pixel kemudian dilakukan boundingbox untuk mengetahui batas area yang akan dipisahkan selanjutnya dilakukan proses cropping untuk memotong citra objek dan dilakukan proses resize citra untuk menormalisasi citra agar ukuran citra semuanya sama.



Gambar 5.16 Citra hasil segmentasi

Pada Gambar 5.16 dapat dilihat garis boundingbox yang sesuai membatasi karakter untuk dapat dipisahkan (cropping). Pixel yang berwarna putih dan berukuran lebih dari batasan area tertentu yang dapat dilakukan bounding box. Dari citra segmentasi ini dapat dilakukan proses cropping, memotong setiap karakter yang telah dibounding box. Untuk kebutuhan input pada proses perbandingan dengan template.

```
fil=imfill(mor,'holes');
%mengambil citra bernilai satu yang lebih dari 400 pixel
final=bwareaopen(fil,400);
%untuk mencari berapa banyak kandidat plat nomor
%dengan mencari dan menandai piksel putih/ 1
%banyak kandidat disimpan pada total
[Label,Total]=bwlabel(final,4);
%fungsi regionprops untuk mencari atribut boundingbox
Iprops=regionprops(Label,'BoundingBox');
obj=floor(cat(1,Iprops.BoundingBox));
finis=[];
for i=1:Total
    crop=imcrop(fil,Iprops(i).BoundingBox);
    crop=imresize(crop,[26 18]);
    cin= crop(:);
    hasil= klasifikasi(cin);
    finis=[finis hasil];
disp(hasil);
end
```

Gambar 5.17 Source code program untuk segmentasi

Pada program untuk segmentasi citra pada Gambar 5.11 ditunjukkan dengan menggunakan perintah `bwareaopen` dengan dengan nilai 400 maksudnya adalah pada citra yang tersimpan pada variable `fil` dilakukan seleksi hanya citra putih

yang luas area nya lebih dari 400 pixel saja yang dapat diambil dan disimpan pada variable final. Selanjutnya variable final sebagai acuan untuk mendapat kandidat citra segmentasi yaitu dengan perintah `bwlabel` yang berfungsi untuk mencari kumpulan pixel berwarna putih sekaligus mengelompokannya dan menghitung banyaknya kumpulan citra putih yang ditemukan dan disimpan pada array Total. Langkah selanjutnya pada program yaitu melakukan boundingbox untuk membatasi citra yang telah dikelompokan sebelumnya menggunakan fungsi `regionprops` dengan atribut 'BoundingBox'. Untuk mengambil nilai boundingboxnya saja dapat dilakukan dengan perintah `cat`.

Proses berikutnya adalah cropping obyek memotong setiap karakter yang terdeteksi. Cropping menggunakan perintah `imcrop` pada MATLAB hasilnya disimpan pada variable `crop` dan proses ini diulang sampai sebanyak nilai variable "Total". Hasil cropping dilakukan proses normalisasi citra agar ukuran citra semuanya sama dilakukan `resize` dapat dilakukan dengan perintah `imresize(crop,[26 18])` yang berarti ukuran dirubah menjadi 26 baris dan 18 kolom.

5.2.5.8 Implementasi Proses Template Matching

Pada proses implementasi template matching terdapat beberapa pendekatan umumnya pendekatan fitur dan pendekatan template. Pada penelitian ini dilakukan klasifikasi citra karakter menggunakan pendekatan template karena lebih efektif untuk mengenali citra dan sebagian besar template merupakan gambar yang cocok.

1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1
1	1	1	0	0	0	0	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	0	0	0	0	0

Gambar 5.18 Matrix karakter U pada database template

Setelah seluruh karakter sudah disegmentasi dilakukan proses klasifikasi dengan template matching di sini algoritma yang dipakai adalah algoritma pembandingan bit "0". Pada algoritma ini terdapat beberapa tahap yaitu:



1. Melakukan operasi pengurangan matrix antara citra karakter input dengan citra karakter pada template
2. Melakukan proses matematika pengurang matrix, citra input akan dikurangi dengan citra yang ada pada template.

a =	b =	c =																																				
<table border="1"> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> </table>	1	1	1	0	1	0	0	1	0	0	1	0	<table border="1"> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	1	0	1	1	0	1	1	0	1	1	1	1	<table border="1"> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>-1</td><td>1</td><td>-1</td></tr> <tr><td>-1</td><td>1</td><td>-1</td></tr> <tr><td>-1</td><td>0</td><td>-1</td></tr> </table>	0	1	0	-1	1	-1	-1	1	-1	-1	0	-1
1	1	1																																				
0	1	0																																				
0	1	0																																				
0	1	0																																				
1	0	1																																				
1	0	1																																				
1	0	1																																				
1	1	1																																				
0	1	0																																				
-1	1	-1																																				
-1	1	-1																																				
-1	0	-1																																				
(a)	(b)	(c)																																				

Gambar 5.19 Pengurangan matrix, matrix (a) dikurangi matrix (b), hasilnya menjadi matrix (c)

3. Melakukan pencarian nilai nilai bit "0" di dalam variable JN, variable yang menyimpan data hasil pengurangan matrix input dengan template. Setelah ditemukan hasil keseluruhan bit 0 kemudian disimpan ke dalam array untuk dibandingkan pada proses selanjutnya.

```

%-----
% matrix citra input (cin) dikurangi dengan setiap karakter template
% template disimpan pd variabel PT1 sampai n
% var Pengurangan dengan Template (PT) menyimpan hasil penguran matrix
% mencari nilai '0' pd variabel PT hasilnya disimpan pd JN (jumlah Nol)
% hasilnya akan disimpan pada var b (lokasi nilai '0' berada)
for PT1 = cin-karakter0 ;
    JN= find(PT1==0);
    [b1,n]= size(JN);
end
for PT2 = cin-karakter1 ;
    JN= find(PT2 ==0);
    [b2,n]= size(JN);
end
for PT3 = cin-karakter2 ;
    JN= find(PT3 ==0);
    [b3,n]= size(JN);
end
for PT4 = cin-karakter3 ;
    JN= find(PT4 ==0);
    [b4,n]= size(JN);
end
    
```

Gambar 5.20 Source code program untuk mencari jumlah bit 0

Pada program di atas citra input (cin) dikurangi dengan citra karakter template, hasilnya disimpan pada variable PT proses pengurangan dilakukan dengan seluruh karakter template. Dari hasil dari pengurangan dicari nilai 0 nya disimpan pada variable JN. Setelah itu jumlah dari karakter 0 akan disimpan pada sebuah array.



- Tahap membandingkan , akan dilakukan perbandingan nilai untuk mencari jumlah bit terbesar yang disimpan pada array HS. Setelah itu ditentukan mana hasil yang cocok.

```

hmax= max(HS);           % mencari nilai HS terbesar
phs= find(HS==hmax);    % mencari posisi max di dalam PT HS
% membandingkan nilai phs untuk menentukan karakter hasil phs=posisi di hs
if phs==1;
    hasil='0';
else if phs==2;
    hasil='1';
else if phs==3;
    hasil='2';
else if phs==4;
    hasil='3';
else if phs==5;
    hasil='4';
else if phs==6;
    hasil='5';

```

Gambar 5.21 Source code mencari hasil 0 terbesar

Pada ada source code Gambar 5.21 dilakukan proses pencarian nilai maksimum pada array HS yang sebelumnya telah menyimpan nilai bit 0 dari pengurangan citra input dengan citra database template. Dengan menggunakan perintah “max(HS)” maka nilai terbesar dapat ditemukan. Setelah nilai terbesar dari array telah ditemukan sekarang mencari posisi di array nomor berapakah nilai tersebut didapatkan. Dengan menggunakan perintah find(HS==hmax) yang berarti mencari nilai hmax pada HS. Kemudian dilakukan pencocokan dengan menggunakan sistem percabangan kondisi if else.

HS <1x51 double>								
	1	2	3	4	5	6	7	8
1	211	233	214	318	196	281	327	309
2								
3								

Gambar 5.22 Array penyimpan jumlah bit 0

Pada gambar 5.22 merupakan array yang menyimpan nilai dari jumlah bit 0 dari hasil operasi pengurangan dan pencarian nilai 0 pada matrix citra satu kolom.

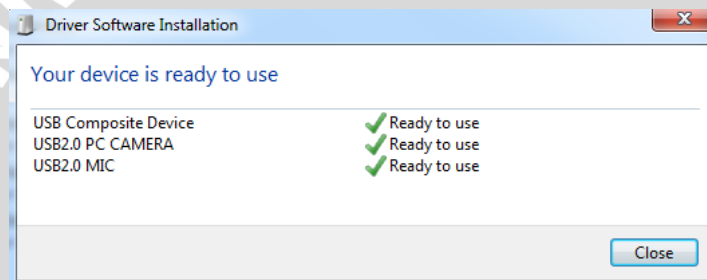


BAB 6 PENGUJIAN

Bab ini membahas tentang proses pengujian dan analisis terhadap sistem setelah proses perancangan dan implementasi telah dilakukan. Dalam penelitian ini proses pengujian terdiri dari beberapa proses pengujian baik dari pengujian perangkat keras maupun pengujian perangkat lunak. Pengujian dilakukan untuk mengetahui fungsionalitas dari sistem yang telah dibuat. Sedangkan analisis dilakukan untuk mendapatkan kesimpulan hasil pengujian yang telah dilakukan.

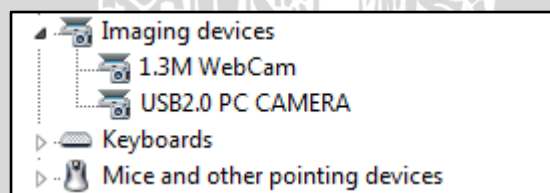
6.1 Pengujian Integrasi Kamera dengan Komputer

Pada pengujian ini dilakukan prosedur untuk mengetahui apakah PC camera sudah terhubung dengan komputer. Pertama yang dilakukan adalah menghubungkan Kamera webcam dengan komputer melalui kabel USB. Setelah saling terhubung, maka keluar notifikasi bahwa device telah terhubung dan melakukan instalasi software.



Gambar 6.1 Notifikasi instalasi kamera telah berhasil

Untuk memastikan kamera telah terhubung dengan komputer dapat dilihat pada computer management, tab device manager.

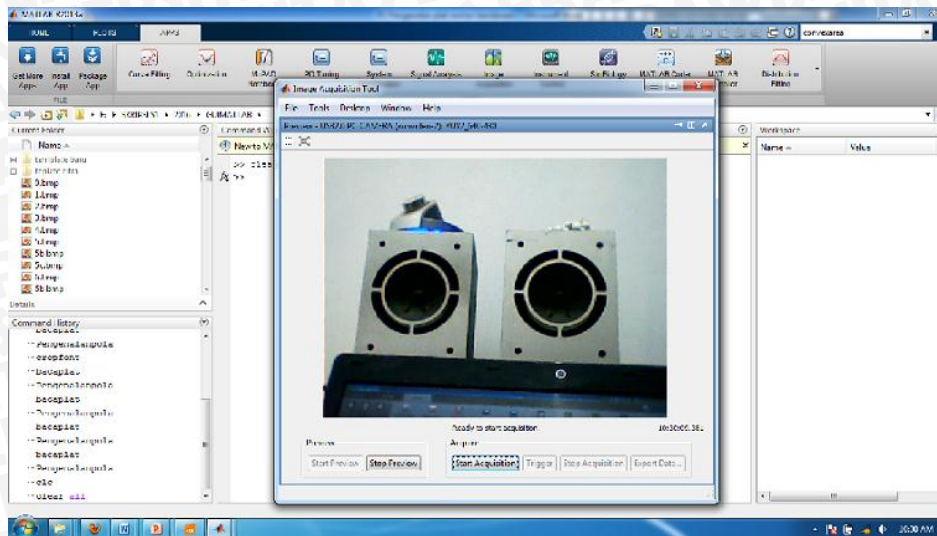


Gambar 6.2 Kamera telah terhubung dengan komputer

Setelah dipastikan kamera telah terhubung dengan komputer selanjutnya kamera dapat dioperasikan dengan memakai aplikasi bawaan komputer atau dengan aplikasi yang lain.

6.2 Pengujian Integrasi Kamera dengan MATLAB

Pada pengujian ini dilakukan untuk mengetahui apakah PC camera sudah terhubung dengan MATLAB. Untuk melihat konfigurasi data dari device(webcam) yang telah terpasang dapat menggunakan *imaqtool* atau dengan perintah *imaqhwinfo*. Pada Gambar 6.3 meunjukkan bahwa kamera telah terhubung dengan MATLAB dan sudah dapat digunakan.



Gambar 6.3 Kamera telah terhubung dengan MATLAB

6.3 Pengujian Pengenalan Plat Nomor

Pengujian ini dilakukan untuk mengetahui tingkat keakuratan metode template matching dengan algoritma pembandingan bit "0" dalam mengenali citra plat nomor kendaraan. Ditentukan posisi kamera depan kendaraan dengan jarak untuk mobil 60 cm dan tinggi 50cm dengan sudut 90^o. Untuk motor jarak kendaraan dengan kamera 30cm, tinggi 100cm dengan sudut 75^o. Pengambilan citra dilakukan pada sore hari cerah.

Langkah-langkah dalam pengujian ini adalah sebagai berikut:

- a. Pengambilan beberapa sampel citra yang mengandung plat nomor dengan menggunakan kamera webcam resolusi 640x480 pixel



Gambar 6.4 Citra capture pada siang hari cerah

Pada Gambar 6.4 merupakan citra yang dicapture pada siang hari dengan kondisi cuaca cerah dan terlalu banyak cahaya yang memantul pada plat nomor kendaraan. Sehingga dihasilkan citra plat nomor kendaraan yang terlihat berwarna keputihan dibagian tengah.



Gambar 6.5 Citra capture pada sore hari cerah

Pada Gambar 6.5 merupakan citra yang dicapture pada sore hari dengan kondisi cuaca cerah dihasilkan citra capture yang sesuai dengan harapan tidak terlalu banyak cahaya yang memantul pada objek plat nomor kendaraan.

b. Mengolah citra menggunakan komputer dan software MATLAB.



Gambar 6.6 Thresholding plat nomor kondisi terlalu banyak pantulan cahaya

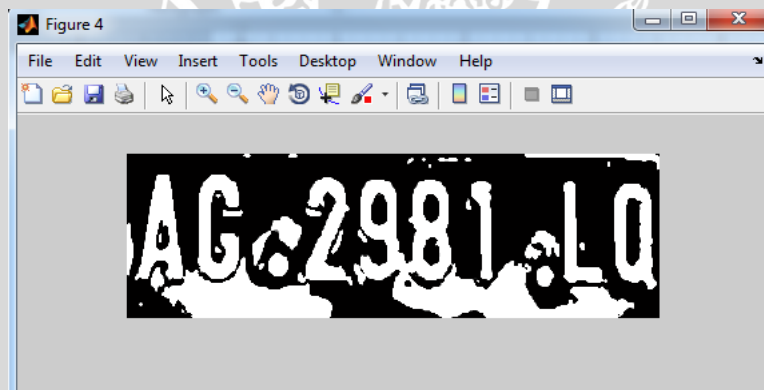
Terlihat pada Gambar 6.6 citra plat nomor hasil thresholding yang terlalu banyak noise cukup mengganggu dalam proses segmentasi. Bila dirubah batas

ambang threshold nya pun akan mempengaruhi objek yang lain sehingga tidak memungkinkan untuk dilakukan proses pengenalan karakter.



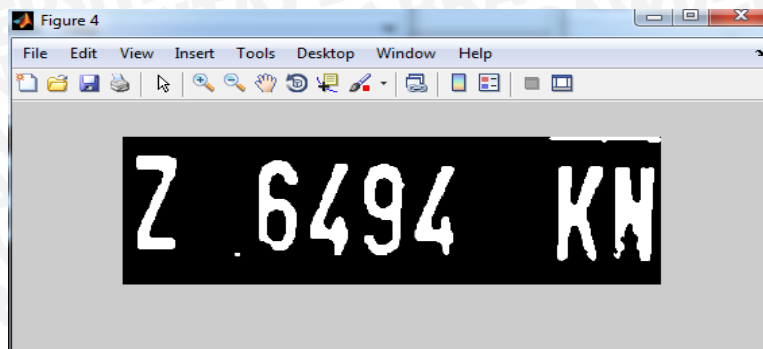
Gambar 6.7 Thresholding plat nomor tidak terdapat pantulan cahaya

Citra hasil cropping pada Gambar 6.7 terlihat hasil tresholding cukup rapih karena pengaruh dari pencahayaannya citra diambil pada sore hari dengan cahaya tidak terlalu terang. Semua kandidat karakter dapat terlihat jelas meskipun ada bagian yang kurang sempurna.



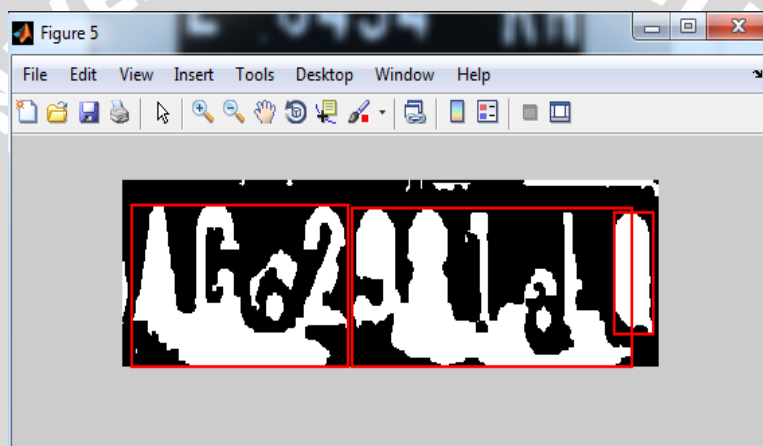
Gambar 6.8 Hasil cropping plat citra terdapat banyak noise

Pada Gambar 6.8 hasil cropping sudah tepat namun untuk masalah noise pada citra tidak dapat dihilangkan karena terlalu banyak dan sudah seperti objek karakter plat nomor.



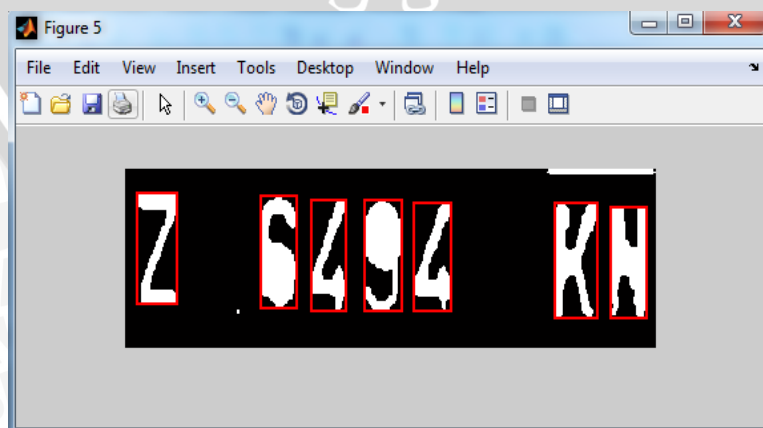
Gambar 6.9 Hasil cropping plat secara rapih

Hasil cropping pada Gambar 6.9 sudah sesuai dengan yang diharapkan dan citra objek karakter juga dapat terlihat. Jika citra sudah pada hasil seperti ini akan lebih muda dalam proses segmentasinya karena hanya ada sedikit noise dan citra karakter tampak lebih menonjol dan jelas.



Gambar 6.10 Segmentasi citra yang terdapat banyak noise

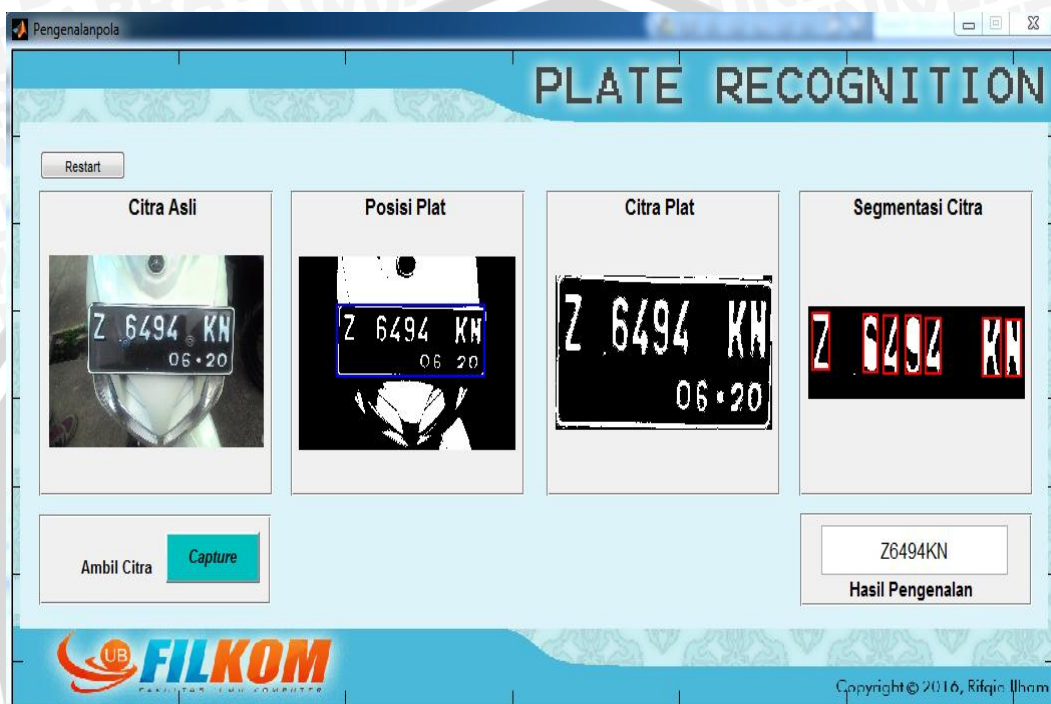
Citra Gambar 6.10 merupakan hasil segmentasi citra yang tidak sesuai dengan yang diharapkan karena bounding box tidak dapat mengkotaki karakter yang seharusnya menjadi kandidat plat nomor, melainkan mengkotaki obyek lain yang tidak sesuai dengan karakter plat nomor.



Gambar 6.11 Segmentasi citra sesuai karakter objek

Hasil segmentasi citra pada Gambar 6.11 sudah sesuai karena dapat melakukan bounding box pada setiap karakter yang memang menjadi kandidat plat nomor. Dengan hasil proses pengolahan citra yang seperti ini dapat mempermudah proses pengenalan plat nomor kendaraan dan hasilnya juga dapat lebih baik.

- c. Menampilkan hasil pengolahan citra dalam GUI berupa citra capture dan hasil pengenalan citra.



Gambar 6.12 Tampilan GUI aplikasi pengenalan plat nomor

Pada Gambar 6.12 merupakan bentuk GUI dari sistem pengenalan plat nomor kendaraan, pada GUI ini dirancang sedemikian rupa untuk mempermudah pengguna dalam menjalankan program tanpa harus eksekusi perintah program melalui perintah teks. Pada tampilan GUI di atas terdapat beberapa citra yang ditampilkan yaitu citra asli yang merupakan citra hasil capture, kemudian sebelah kanan ada tab posisi plat yang menampilkan hasil pencarian posisi plat nomor kendaraan yang telah terdeteksi dengan diberi garis kotak berwarna biru. Tab selanjutnya adalah citra plat menampilkan hasil *cropping* obyek plat nomornya saja yang telah terdeteksi. Kemudian tab sebelah kanan ada satu lagi yaitu yang menampilkan hasil segmentasi citra berupa citra biner yang telah ditandai dengan garis berwarna merah untuk setiap karakter yang dideteksi. Untuk memberi perintah menjalankan program yaitu dengan menekan tombol capture, sistem akan berjalan mulai dari mengcapture citra yang mengandung plat nomor kemudian diproses dengan pengolahan citra digital untuk dikenali. Hasil pengenalan disajikan dalam kolom hasil pengenalan dalam format teks. Untuk mereset program dengan menekan tombol restart program akan berjalan dari awal kembali.

6.4 Analisa Hasil Pengujian

Pada pengujian pengenalan plat nomor dilakukan pada sepuluh sampel pengujian dari berbagai jenis kendaraan dengan plat nomor standar nasional indonesia dan didapatkan hasil yang dapat dilihat pada table 6.1.









Akurasi pengenalan karakter pada setiap plat nomor kendaraan:












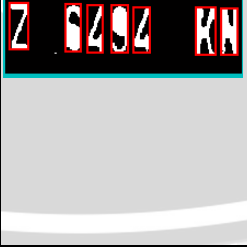
$$AP = \frac{K_d}{K_t} \times 100\%$$

Keterangan:

- AP : Akurasi pengenalan plat nomor kendaraan
- K_d : Jumlah karakter dikenali dengan benar
- K_t : Jumlah karakter terdeteksi

Tabel 6.1 Tabel hasil pengujian akurasi pengenalan plat nomor

No	Citra Mengandung Plat nomor	Segmentasi karakter	Hasil Pengenalan	Akurasi
1			N473AD	100%
2			DA7716AR	100%
3			W1945RB	100%
4			NE354TAI	87.5%

5			N1865AD	100%
6			S4255JU	100%
7			AG2402BJ	100%
8			N2925EV	100%
9			W6#89MY	85.7%
10			Z6494KN	100%

Dari hasil pengujian pada setiap plat nomor kendaraan didapatkan 72 karakter yang terdeteksi, dari hasil tersebut karakter yang dapat dikenali dengan benar sejumlah 70 karakter dan 2 karakter tidak dapat dikenali dengan benar.

Akurasi pengenalan karakter keseluruhan:

$$APS = \frac{S_{kd}}{S_{kt}} \times 100\%$$

Keterangan:

- APS : Akurasi pengenalan plat nomor kendaraan secara keseluruhan
- S_{kd} : Jumlah seluruh karakter dikenali dengan benar
- S_{kt} : Jumlah karakter terdeteksi

Berdasarkan data keseluruhan yang diperoleh akurasi dari sistem pengenalan plat nomor menggunakan metode morfologi dan template matching ini adalah sebesar 97,2 %. Hal ini menunjukkan bahwa metode pengenalan sudah cukup baik dalam mengenali plat nomor kendaraan. Pada plat nomor yang tidak dapat terbaca terdapat beberapa dugaan kendala seperti kurangnya template database plat nomor, bisa juga karena pantulan cahaya yang terlalu banyak pada objek plat nomor, dan pengambilan citra yang tidak sesuai. Untuk dapat mengatasi permasalahan plat nomor yang tidak bisa dikenali dari uji coba tersebut.



BAB 7 PENUTUP

Kesimpulan yang diperoleh dari hasil pengujian dan analisa hingga saran yang perlu ditambahkan untuk penelitian selanjutnya.

7.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, pengujian dan analisis yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Algoritma template matching dapat melakukan proses pencocokan citra karakter dengan cukup baik.
2. Dengan metode morfologi dapat memperbaiki kondisi bentuk objek citra sehingga dapat lebih muda dalam proses pengenalan karakternya.
3. Pada metode template matching semakin banyak jenis citra yang dilatih untuk dijadikan template maka kemungkinan akurasi pengenalan karakter menjadi semakin tinggi.
4. Dengan metode template matching menggunakan algoritma perbandingan bit 0 memiliki tingkat keberhasilan yang cukup tinggi terhadap pencocokannya dari hasil pengujian didapatkan akurasi pengenalan karakter sebesar 97.2%.
5. Kelemahan pada algoritma template matching yaitu perhitungan dan penyimpanan data yang besar untuk proses pencocokkan

7.2 Saran

Adapun saran yang dapat saya berikan, pada penelitian ini masih cukup sederhana. Diharapkan untuk pengembangannya tidak hanya pada bentuk simulasi, melainkan dapat diterapkan pada aplikasi atau perangkat yang digunakan sehari-hari seperti aplikasi mobile maupun aplikasi PC. Bisa juga diimplementasikan pada mikro komputer agar lebih fleksibel dalam penerapannya. Pada penentuan acuan perbandingan template masih menggunakan operasi yang cukup sederhana mungkin dapat ditambahkan lagi menggunakan pendekatan fitur untuk memperdetail proses pengenalan karakter obyek. Untuk dapat dipergunakan pada jarak yang lebih jauh bisa ditambahkan algoritma tambahan seperti feature extraction sebagai tambahan acuan fiturnya. Sebagai pengembangan teknologi, mungkin dapat menggunakan metode Morfologi dan template Matching pada penelitian yang lain dan dapat diimplementasikan pada suatu perangkat.

DAFTAR PUSTAKA

- Admin, 2011. *Beautiful Butterfly*. [image online] Tersedia melalui: <http://www.wallpaperhere.com/Beautiful_Butterfly_64895> [Diakses 23 September 2015]
- Andika, M.L., 2011. *Desain Baru Pelat Nomor Kendaraan*. [image online] Tersedia melalui: <<http://oto.detik.com/read/2011/05/09/094600/1635217/648/ini-dia-desain-baru-pelat-nomor-kendaraan>> [Diakses 23 September 2015]
- Bahri, R.S., dan Maliki, I., 2012 *Perbandingan Algoritma Template Matching Dan Feature Extraction Pada Optical Character Recognition*. [e-journal]. Tersedia melalui: Jurnal Informatika UNIKOM< <http://komputa.if.unikom.ac.id/jurnal/perbandingan-algoritma.5>> [diakses 20 November 2015]
- Dhiraj Y. L., Pramod, B. B., 2014. *A Review Paper on Automatic Number Plate Recognition (ANPR) System*. [e-journal] (2014) Vol 1 Issue 1. Tersedia melalui: International Journal of Innovative Research in Advanced Engineering<<http://http://ijirae.com/images/downloads/vol1issue2/AEE10084.April14.16.pdf>> [Diakses 25 September 2015]
- Edrinc H.K., dan Kursat. K.C., 2011. *Artificial Neural Networks Based Vehicle License Plate Recognition* [e-journal]. Tersedia melalui: sciencedirect <<http://www.sciencedirect.com/science/article/pii/S1877050910005442>> [Diakses 25 Desember 2015]
- Fauzi, M.H. dan Tjandrasa H., 2012 *Implementasi Thresholding Citra menggunakan Algoritma Hybrid Optimal Estimation*. [e-journal]. Tersedia melalui: <<http://digilib.its.ac.id/public/ITS-Undergraduate-12935-Paper.pdf>> [diakses 18 November 2015]
- Havit, 2015. *HAVIT® HV-V612 HD Camera and Webcam with Microphone*. [image online] Tersedia melalui: <<http://www.amazon.com/HV-V612-Microphone-Messenger-Windows-Perfect/dp/B00SINI864>> [Diakses 26 Desember 2015]
- KAPOLRI, 2012. *Registrasi dan Identifikasi Kendaraan Bermotor*. Tersedia melalui: <<https://www.polri.go.id/pustaka/pdf/PERATURAN%20KAPOLRI%20NOMOR%205%20TAHUN%202012%20TENTANG%20REGISTRASI%20DAN%20IDENTIFIKASI%20KENDARAAN%20BERMOTOR.pdf>> [Diakses 25 November 2015]
- Kumaseh, M.R., Latumakulita, L., dan Nainggolan, N., 2013. *Segmentasi Citra Digital Ikan Menggunakan Metode Thresholding*. [e-journal]. Tersedia melalui: Jurnal Ilmiah Sains Universitas Sam Ratulangi <<http://ejournal.unsrat.ac.id/index.php/JIS/article/view/2057/1631>> [diakses 8 Desember 2015]
- Mathworks. 2014. *Getting Started Guide*. [pdf]. Tersedia di: <https://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf>, [diakses 18 November 2015]

- Michael, R., dan Novita. H., 2015. *An Examination of Character Recognition on ID card using Template Matching Approach* [e-journal]. Tersedia melalui: sciencedirect < <http://www.sciencedirect.com/science/article/pii/S1877050915020633>> [Diakses 25 Desember 2015]
- Munawaroh, S. dan Sutanto, F.A., 2010 *Pengolah Citra Digital untuk Identifikasi Uang Kertas*. [e-journal]. Tersedia melalui: Jurnal Teknologi Informasi DINAMIK Universitas Stikubank <<http://www.unisbank.ac.id/ojs/index.php/fti1/article/view/109>> [diakses 16 November 2015]
- Naviri, G.D., 2015. *pengenalan alphabet bahasa isyarat tangan menggunakan smart k-nearest neighbor (smart knn) untuk tuna rungu dan tuna wicara*. S1. Universitas Brawijaya. Tersedia di < <http://ptiik.ub.ac.id/skripsi> >, [diakses 20 Desember 2015]
- Putra, D., 2010. *Pengolahan Citra Digital*, Yogyakarta : Penerbit Andi
- Rizal, F.M., kurniawan, W., Muttaqin, A., 2014. *Sistem Deteksi Slot Parkir Menggunakan Pengolahan Citra Digital Dengan Metode Thresholding*. [e-journal] (2014) Vol. 6 No. 1. Tersedia melalui: jurnal Filkom Universitas Brawijaya <<http://filkom.ub.ac.id/doro/archives/detail/DR00011201512>> [Diakses 22 September 2015]
- Rohmatullah, M.I., 2014. *Identifikasi Citra pada Plat Nomor Kendaraan Mobil Pribadi Menggunakan Metode K-Nearest Neighbour*. S1. Universitas Dian Nuswantoro. Tersedia di <http://eprints.dinus.ac.id/13315/1/jurnal_13890.pdf>, [diakses 16 November 2015]
- Setiawan, A., Suryani, E., dan Wiharto., 2014 *Segmentasi citra sel darah merah berdasarkan morfologi sel untuk mendeteksi anemia defisiensi besi*. [e-journal]. Tersedia melalui: UNS-F. MIPA Jur. Informatika < <http://dglib.uns.ac.id/>> [diakses 7 November 2015]
- Sofian R.B. dan Irfan M., 2012 *Perbandingan Algoritma Template Matching dan Feature Extraction pad Optical Character Recognition*. [e-journal]. Tersedia melalui: Jurnal Komputer dan Informatika UNIKOM <<http://komputa.if.unikom.ac.id/jurnal/perbandingan-algoritma.5>> [diakses 18 November 2015]
- Taufik, M.S., Wirayuda, T.A., dan Dayawati. R.N., 2010 *Pengenalan Plat Nomor Otomatis Menggunakan Principal Component Analysis (Pca) Dan Learning Vector Quantization (Lvq)*. [e-journal]. Tersedia melalui: Jurnal Informatika Telkom University <<https://repository.telkomuniversity.ac.id/home/catalog/id/94884/slug/pengenalan-plat-nomor-otomatis-menggunakan-principal-component-analysis-pca-dan-learning-vector-quantization-lvq-.html>> [diakses 7 November 2015]

LAMPIRAN

A.1 Source code Capture Citra

```

1  % --- Executes on button press in pushbutton1.
2  function pushbutton2_Callback(hObject, eventdata, handles)
3  % hObject    handle to pushbutton1 (see GCBO)
4  % eventdata  reserved - to be defined in a future version of MATLAB
5  % handles    structure with handles and user data (see GUIDATA)
6  guidata(hObject, handles);
7  axes(handles.axes1); %memilih axes1 sebagai letak gambar yang
8  dimunculkan
9  vid= videoinput('winvideo',2,'YUY2_640x480');
10 % preview(vid);
11 hImage = imagesc(zeros(400,300), 'Parent', handles.axes1);
12 preview(vid, hImage);
13 % set(vid,'ReturnedColorSpace','rgb');
14 set(vid, 'FramesPerTrigger', Inf);
15 set(vid, 'ReturnedColorspace', 'rgb');
16 pause(5);
17 capture= getsnapshot(vid);
18 % captureRGB= ybcr2rgb(capture);
19 guidata(hObject, handles);
20 axes(handles.axes2); %memilih axes2 sebagai letak gambar yang
   dicapture
   imshow(capture);

```

A.2 Source code Baca Plat

```

1  clc;
2  close all;
3  clear all;
4  %-----
5  % load citra plat nomor dari hasil kamera
6  citra=image;
7  crop=posisiplat(citra);
8  croprz= imresize(crop,[220 415]);
9  croprxy= imcrop(croprz,[10,20,398,125]);
10 % median filtering untuk menghapus noise
11 g=medfilt2(croprxy,[3 3]);
12 % merubah format menjadi biner (logical)

```

```

13 B=logical(g);
14 % proses morfologi citra untuk memperbaiki bentuk dan komposisi
15 mor=imerode(B,strel('square',3));
16 mor2=imdilate(mor,strel('rectangle',[3 4]));
17 % operasi untuk mengisi lubang pada tengah objek
18 fil=imfill(mor,'holes');
19 final=bwareaopen(fil,400);
20 % Proses segmentasi karakter
21 [Label,Total]=bwlabel(final,4);
22 Iprops=regionprops(Label,'BoundingBox','Image');
23 obj=floor(cat(1,Iprops.BoundingBox));
24
25 % plot bounding box, untuk menampilkan garis bounding box
26 hold on
27 for n=1:size(Iprops,1)
28
29     rectangle('Position',Iprops(n).BoundingBox,'EdgeColor','r','LineWid
30     th',2);
31 end
32 hold off
33
34 % proses looping untuk cropping citra segmentasi
35 finis=[];
36 for i=1:Total
37     crop=imcrop(fil,Iprops(i).BoundingBox);
38     crop=imresize(crop,[26 18]);
39     out= crop(:);
40     % out = im2double(out2);
41     hasil= klasifikasi(out);
42     finis=[finis hasil];
43     % disp(finis);
44 end

```

A.3 Source code Posisi Plat

```

1 function crop=posisi(citra);
2 f2=rgb2gray(citra);
3 f2=imresize(f2,[360 540]);
4
5 % proses thresholding
6 [jmlB jmlK]=size(f2);
7 for x=1:jmlB

```



```

8     for y=1:jmlK
9         if f2(x,y) > 190
10            g(x,y)=255;
11        else
12            g(x,y)=0;
13        end
14    end
15 end
16 % proses binerisasi dan seleksi citra putih
17 final=bwareaopen(g,450);
18 stats=regionprops(final,'all');
19 stats1=floor(cat(1,stats.BoundingBox));
20 figure(1), imshow(final);
21 hold on
22 for n=1:size(stats,1)
23
24     rectangle('Position',stats(n).BoundingBox,'EdgeColor','b','LineWidth',2);
25 end
26 hold off
27
28 %proses cropping area plat nomor
29 N_maks=0;
30 Area_maks=0;
31 for i=1:length(stats)
32     if stats(i).ConvexArea > Area_maks
33         Area_maks= stats(i).ConvexArea;
34         N_maks=i;
35     end
36 end
37 crop=imcrop(g,stats(N_maks).BoundingBox);
38

```

A.4 Source code Implementasi pada GUI

```

1     function varargout = Pengenalanpola(varargin)
2     % Begin initialization code - DO NOT EDIT
3     gui_Singleton = 1;
4     gui_State = struct('gui_Name',       mfilename, ...
5                       'gui_Singleton',  gui_Singleton, ...
6                       'gui_OpeningFcn', @Pengenalanpola_OpeningFcn,
7                       ...
8                       'gui_OutputFcn',  @Pengenalanpola_OutputFcn, ...

```

```

9         'gui_LayoutFcn', [] , ...
10        'gui_Callback', []);
11    if nargin && ischar(varargin{1})
12        gui_State.gui_Callback = str2func(varargin{1});
13    end
14
15    if narginout
16        [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
17    else
18        gui_mainfcn(gui_State, varargin{:});
19    end
20    % End initialization code - DO NOT EDIT
21    % --- Executes just before Pengenalanpola is made visible.
22    function Pengenalanpola_OpeningFcn(hObject, eventdata, handles,
23    varargin)
24    % This function has no output args, see OutputFcn.
25    % hObject    handle to figure
26    % eventdata  reserved - to be defined in a future version of MATLAB
27    % handles    structure with handles and user data (see GUIDATA)
28    % varargin   command line arguments to Pengenalanpola (see
29    VARARGIN)
30    % Choose default command line output for Pengenalanpola
31    handles.output = hObject;
32    guidata(hObject, handles);
33    %-----
34    hback= axes('unit','normalized','position',[0 0 1 1]);
35    uistack(hback,'bottom');
36    [back map] = imread('bgmatlab.png') ;
37    image(back)
38    % Update handles structure
39    guidata(hObject, handles);
40
41    % --- Outputs from this function are returned to the command line.
42    function varargout = Pengenalanpola_OutputFcn(hObject, eventdata,
43    handles)
44    % varargout  cell array for returning output args (see VARARGOUT);
45    % hObject    handle to figure
46    % eventdata  reserved - to be defined in a future version of MATLAB
47    % handles    structure with handles and user data (see GUIDATA)
48    % Get default command line output from handles structure
49    varargout{1} = handles.output;

```

```

50
51 % --- Executes on button press in pushbutton1.
52 function pushbutton1_Callback(hObject, eventdata, handles)
53 % hObject     handle to pushbutton1 (see GCBO)
54 % eventdata   reserved - to be defined in a future version of MATLAB
55 % handles     structure with handles and user data (see GUIDATA)
56 [namafilename, formatfile] = uigetfile({'*.jpg'}, 'membuka gambar');
57 %memilih
58 gambar
59 image = imread([formatfile, namafilename]); %membaca gambar
60 imageres= imresize(image,[400 600]);
61 bacaplat;      %untuk menyambungkan program ke GUI
62
63 guidata(hObject, handles);
64
65 axes(handles.axes1); %memilih axes1 sebagai letak gambar yang
66 dimunculkan
67 imshow(imageres); %memunculkan gambar
68
69 % axes(handles.axes7);
70 % imshow('fil3.gif')
71 axes(handles.axes2);
72 % filres= imresize(fil,[400 600]);
73 imshow(fil);
74 hold on
75 for n=1:size(Iprops,1)
76 rectangle('Position',Iprops(n).BoundingBox,'EdgeColor','r','LineWid
77 th',2);
78
79 end
80 hold off
81 ;
82 set(handles.plat, 'String',finis);
83
84 function plat_Callback(hObject, eventdata, handles)
85 % hObject     handle to plat (see GCBO)
86 % eventdata   reserved - to be defined in a future version of MATLAB
87 % handles     structure with handles and user data (see GUIDATA)
88
89 % --- Executes during object creation, after setting all
90 properties.
91 function plat_CreateFcn(hObject, eventdata, handles)









































































```



```
91 % hObject handle to plat (see GCBO)
92 % Hint: edit controls usually have a white background on Windows.
93 % See ISPC and COMPUTER.
94 if ispc && isequal(get(hObject,'BackgroundColor'),
95 get(0,'defaultUicontrolBackgroundColor'))
96     set(hObject,'BackgroundColor','white');
97 end
98
99 % --- Executes during object creation, after setting all
100 properties.
101 function axes6_CreateFcn(hObject, eventdata, handles)
102 % hObject handle to axes6 (see GCBO)
103 % eventdata reserved - to be defined in a future version of MATLAB
104 % handles empty - handles not created until after all CreateFcns
105 called
106
107 % --- Executes on mouse press over figure background, over a
108 disabled or
109 % --- inactive control, or over an axes background.
110 function figure1_WindowButtonDownFcn(hObject, eventdata, handles)
111 % hObject handle to figure1 (see GCBO)
112 % eventdata reserved - to be defined in a future version of MATLAB
113 % handles structure with handles and user data (see GUIDATA)
114
115 % --- Executes on mouse motion over figure - except title and menu.
116 function figure1_WindowButtonMotionFcn(hObject, eventdata, handles)
117 % hObject handle to figure1 (see GCBO)
118 % eventdata reserved - to be defined in a future version of MATLAB
119 % handles structure with handles and user data (see GUIDATA)
```

B.1 Citra Template

Database citra template terdiri dari 108 karakter huruf , setiap huruf mempunyai 3 variasi bentuk. Dapat dilihat pada Gambar 7.1

 0 BMP File 166 bytes	 0b BMP File 166 bytes	 0c BMP File 166 bytes	 1 BMP File 166 bytes
 1b BMP File 166 bytes	 1c BMP File 166 bytes	 2 BMP File 166 bytes	 2b BMP File 166 bytes
 2c BMP File 166 bytes	 3 BMP File 166 bytes	 3b BMP File 166 bytes	 3c BMP File 166 bytes
 4 BMP File 166 bytes	 4b BMP File 166 bytes	 4c BMP File 166 bytes	 5 BMP File 166 bytes
 5b BMP File 166 bytes	 5c BMP File 166 bytes	 6 BMP File 166 bytes	 6b BMP File 166 bytes
 6c BMP File 166 bytes	 7 BMP File 166 bytes	 7b BMP File 166 bytes	 7c BMP File 166 bytes
 8 BMP File 166 bytes	 8b BMP File 166 bytes	 8c BMP File 166 bytes	 9 BMP File 166 bytes
 9b BMP File 166 bytes	 9c BMP File 166 bytes	 A BMP File 166 bytes	 A2 BMP File 166 bytes
 A3 BMP File 166 bytes	 B BMP File 166 bytes	 B2 BMP File 166 bytes	 B3 BMP File 166 bytes
 C BMP File 166 bytes	 C2 BMP File 166 bytes	 C3 BMP File 166 bytes	 D BMP File 166 bytes
 D2 BMP File 166 bytes	 D3 BMP File 166 bytes	 E BMP File 166 bytes	 E2 BMP File 166 bytes
 E3 BMP File 166 bytes	 F BMP File 166 bytes	 F2 BMP File 166 bytes	 F3 BMP File 166 bytes
 G BMP File 166 bytes	 G2 BMP File 166 bytes	 G3 BMP File 166 bytes	 H BMP File 166 bytes
 H2 BMP File 166 bytes	 H3 BMP File 166 bytes	 I BMP File 166 bytes	 I2 BMP File 166 bytes
 I3 BMP File 166 bytes	 J BMP File 166 bytes	 J2 BMP File 166 bytes	 J3 BMP File 166 bytes
 K BMP File 166 bytes	 K2 BMP File 166 bytes	 K3 BMP File 166 bytes	 L BMP File 166 bytes
 L2 BMP File 166 bytes	 L3 BMP File 166 bytes	 M BMP File 166 bytes	 M2 BMP File 166 bytes
 M3 BMP File 166 bytes	 N BMP File 166 bytes	 N2 BMP File 166 bytes	 N3 BMP File 166 bytes

	O BMP File 166 bytes		O2 BMP File 166 bytes		O3 BMP File 166 bytes		P BMP File 166 bytes
	P2 BMP File 166 bytes		P3 BMP File 166 bytes		Q BMP File 166 bytes		Q2 BMP File 166 bytes
	Q3 BMP File 166 bytes		R BMP File 166 bytes		R2 BMP File 166 bytes		R3 BMP File 166 bytes
	S BMP File 166 bytes		S2 BMP File 166 bytes		S3 BMP File 166 bytes		T BMP File 166 bytes
	T2 BMP File 166 bytes		T3 BMP File 166 bytes		U BMP File 166 bytes		U2 BMP File 166 bytes
	U3 BMP File 166 bytes		V BMP File 166 bytes		V2 BMP File 166 bytes		V3 BMP File 166 bytes
	W BMP File 166 bytes		W2 BMP File 166 bytes		W3 BMP File 166 bytes		X BMP File 166 bytes
	X2 BMP File 166 bytes		X3 BMP File 166 bytes		Y BMP File 166 bytes		Y2 BMP File 166 bytes
	Y3 BMP File 166 bytes		Z BMP File 166 bytes		Z2 BMP File 166 bytes		Z3 BMP File 166 bytes

Gambar 7.1 Citra template database

B.2 Citra Data Latih

Citra yang digunakan untuk membuat database template terdiri dari beberapa plat nomor yang diekstrak diambil setiap karakter yang diperlukan.

Tabel 7.1 Citra data latih





