

2017

Pengenalan Plat Nomor Kendaraan Indonesia Menggunakan Principal Component Analysis dan Metode K-Nearest Neighbor

Rohimah, Wudda

<http://repositori.usu.ac.id/handle/123456789/2378>

Downloaded from Repositori Institusi USU, Universitas Sumatera Utara

**PENGENALAN PLAT NOMOR KENDARAAN INDONESIA
MENGUNAKAN *PRINCIPAL COMPONENT ANALYSIS* DAN
*METODE K-NEAREST NEIGHBOR***

SKRIPSI

WUDDA ROHIMAH

121402005



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

MEDAN

2017

**Pengenalan Plat Nomor Kendaraan Indonesia
Menggunakan *Principal Component Analysis* dan
*Metode K-Nearest Neighbor***

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Teknologi Informasi

**WUDDA ROHIMAH
121402005**



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2017**

PERSETUJUAN

Judul : PENGENALAN PLAT NOMOR KENDARAAN
INDONESIA MENGGUNAKAN *PRINCIPAL
COMPONENT ANALYSIS* DAN *METODE
K-NEAREST NEIGHBOR*

Kategori : SKRIPSI

Nama : WUDDA ROHIMAH

Nomor Induk Mahasiswa : 121402005

Program Studi : SARJANA (S1) TEKNOLOGI INFORMASI

Departemen : TEKNOLOGI INFORMASI

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

Komisi Pembimbing :

Pembimbing 2 Pembimbing 1

Romi Fadillah Rahmat, B.Comp.Sc., M.Sc
NIP. 19860303 201012 1 004

Dani Gunawan, S.T., M.T
19820915 201212 1 002

Diketahui / Disetujui oleh
Program Studi S1 Teknologi Informasi
Ketua,

Romi Fadillah Rahmat, B.Comp.Sc., M.Sc
NIP. 19860303 201012 1 004

PERNYATAAN**Pengenalan Plat Nomor Kendaraan Indonesia
Menggunakan *PRINCIPAL COMPONENT ANALYSIS* dan
*METODE K-NEAREST NEIGHBOR*****SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, Oktober 2017

WUDDA ROHIMAH
121402005

UCAPAN TERIMA KASIH

Puji dan syukur kehadirat Allah SWT, karena rahmat dan izin-Nya penulis dapat menyelesaikan penyusunan skripsi ini, sebagai syarat untuk memperoleh gelar Sarjana Komputer, pada Program Studi S1 Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara. Selama penyelesaian tugas akhir ini, banyak bantuan dan kerja sama serta doa dan dukungan dari berbagai pihak, oleh karena itu penulis sampaikan ucapan terima kasih kepada :

1. Allah SWT, karena Rahmat dan Ridho-Nya penulis dapat menyelesaikan penyusunan tugas akhir ini.
2. Kedua Orang tua penulis Ayahanda Suryanto,S.Pd., dan Ibunda Yusrinani yang telah memberikan dukungan, semangat, kasih sayang, nasehat, dan doa semangat yang tiada putusnya kepada penulis sehingga penulis dapat menyelesaikan skripsi ini.
3. Adik penulis yaitu Rizky Abdillah dan Rifqah Afifah yang memberikan semangat dan dukungan kepada penulis.
4. Seluruh keluarga besar dari Ayahanda dan Ibunda Farsiati,S.Ag., dan Pakde Ayah Ucok Nasution yang telah memberikan dukungan, semangat dan doa semangat yang tiada putusnya kepada penulis.
5. Bapak Dani Gunawan,S.T.,M.T selaku Dosen Pembimbing I dan Bapak Romi Fadillah Rahmat, B.Comp.Sc, M.Sc selaku Dosen Pembimbing II yang telah membimbing penulis dalam menyelesaikan skripsi ini.
6. Bapak Baihaqi Siregar,S.Si., MT selaku Dosen Pembimbing I, Bapak Dr. Sawaluddin,M.IT selaku Dosen Pembimbing II yang telah memberikan kritik dan saran dalam penyempurnaan skripsi ini.
7. Bapak Muhammad Anggia Muchtar, ST., MM.IT selaku dosen Pembimbing akademik yang telah membimbing penulis dari semester awal hingga semester akhir.
8. Ketua dan Sekretaris Program Studi S1 Teknologi Informasi, Dekan dan Wakil Dekan Fakultas Ilmu Komputer dan Teknologi Informasi, dan seluruh dosen

serta staff kepegawaian di lingkungan Program Studi S1 Teknologi Informasi yang telah membantu dan membimbing penulis selama waktu perkuliahan.

9. Syafrizal Lubis selaku pria yang selalu memberikan doa, nasehat, dukungan dan semangat juga selalu mendengarkan keluh kesah, membantu, dan menemani penulis saat pengerjaan skripsi ini.
10. Sahabat penulis Renato Rashidi Siahaan, yang selalu memberikan dukungan dan semangat kepada penulis saat pengerjaan skripsi ini.
11. Sahabat penulis Ainul Husna, yang selalu mendengarkan curahan hati, dan sebagai teman seperjuangan dalam berbagai pengerjaan tugas selama perkuliahan yang telah memberikan dukungan dan semangat kepada penulis saat pengerjaan skripsi ini
12. Sahabat penulis, Kesebelasan Mayya Noor Lubis, Chairina Ulfa, Hasna Susanti Sitompul, Zahara Putri Sakila, Siti Hasanah, Misbah Hasugian, Kania Arfianti, Nani Sylviana Pasi, Ida Mayasari yang selalu memberikan dukungan, memberikan nasehat dan sebagai saudara diperantauan.
13. Sahabat Penulis, Hijab Traveller Ade Ayu Wardani, Suri Eka Pratiwi dan Citra Rezeki Kusuma yang selalu memberikan semangat dan dukungan kepada penulis.
14. Sahabat Dubes Igen, Nisa, Utiya, Azra, Andika dan teman-teman yang lain yang selalu memberi dukungan kepada penulis.
15. Teman-teman penulis ANAK PERTAMA, KEDE-IT, HIMATIF 2016-2017, GENBI-SUMUT, BIMBINGAN PAK DANI, BIMBINGAN PAK ROMI, GRUP S.KOM dan Kelompok-kelompok belajar lain saat perkuliahan, Terimakasih atas kerjasamanya selama ini.
16. Teman-teman Teknologi Informasi USU Novira, Mutiara, Willa, Ipat, Novia, Hendro, Tommy, Wandy terkhusus angkatan 2012, abangnda Akira, dan kakanda senior dan adik-adik junior Zaki, yang juga memberikan semangat kepada penulis.

ABSTRAK

Plat nomor adalah salah satu jenis identifikasi kendaraan bermotor bermotor. Plat nomor juga disebut plat registrasi kendaraan. Bentuknya berupa potongan plat logam atau plastik yang dipasang pada kendaraan bermotor sebagai identifikasi resmi. Kendaraan bermotor di Indonesia terus meningkat tiap tahunnya, juga berbanding lurus dengan meningkatnya tingkat kecelakaan yang disebabkan pelanggaran lalu lintas di jalan raya. Pada saat ini pengguna jalan di kota-kota besar di Indonesia khususnya kendaraan roda dua memerlukan program yang dapat memantau para pengendara kendaraan yang tidak disiplin dalam berkendara, seperti menerobos lampu merah. Sehingga diperlukannya sistem yang dapat mendeteksi plat nomor kendaraan untuk membantu proses pengolahan data plat nomor kendaraan secara otomatis. Metode yang digunakan dalam penelitian yaitu *K-Nearest Neighbor*. Plat nomor kendaraan digunakan sebagai masukan untuk proses pengolahan citra. Tahapan-tahapan yang dilakukan sebelum identifikasi yaitu proses pengolahan citra dan *feature extraction* dengan menggunakan metode *Principal Component Analysis*. Setelah dilakukan pengujian pada penelitian ini, didapatkan kesimpulan bahwa metode yang diajukan memiliki kemampuan dalam mendeteksi plat nomor kendaraan dengan akurasi sebesar 88,22%.

Kata Kunci : Plat nomor, *K-Nearest Neighbor*, *Principal Component Analysis*

LICENSE PLATE IMAGE DETECTION USING PRINCIPAL COMPONENT ANALYSIS AND K-NEAREST NEIGHBOR IN INDONESIA

ABSTRACT

License plate is one type of identification method for vehicle. License plate is also called vehicle registration plate. It comes in a form of flat metal plate or plastic which is attached to the vehicle as a formal identifier. Vehicles in Indonesia keeps on increasing every year, and it is directly proportional to the increase of accident reports which are caused by traffic violations. Today, a system which can be used to monitor these violation is needed in order to detect these undisciplined drivers. The system will detect the license plate number of the vehicle who violates the traffic and process the data of the license plate's owner automatically. The method which is used in this research is *K-Nearest Neighbor*. License plate number is used as the input for the system as an image, the process done before identification includes image processing and feature extraction with *Principal Component Analysis* method. After calibration and research was done, it can be concluded that the method proposed have the capability to detect the license plate number with an accuracy of 88,22%.

Keywords : License plate, K-Nearest Neighbor, Principal Component Analysis

DAFTAR ISI

	Halaman
Persetujuan	ii
Pernyataan	iii
Ucapan Terima Kasih	iv
Abstrak	vii
<i>Abstract</i>	viii
Daftar Isi	ix
Daftar Tabel	xii
Daftar Gambar	xiii
 Bab 1 Pendahuluan	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian	3
1.4. Batasan Masalah	3
1.5. Manfaat Penelitian	3
1.6. Metodologi Penelitian	3
1.7. Sistematika Penulisan	4
 Bab 2 Landasan Teori	
2.1. Plat Nomor Kendaraan	5
2.2. Citra	6
2.2.1. Pengertian Citra	6
2.2.2. Citra Digital	7
2.2.3. Pengolahan Citra Digital	8
2.2.3.1. <i>Resizing</i>	9
2.2.3.2. <i>Grayscale</i>	9
2.2.3.3. Deteksi tepi <i>Canny</i>	10

2.2.3.4.	<i>Binarization (Thresholding)</i>	11
2.2.3.5.	<i>Gaussian blur</i>	11
2.2.3.6.	Deteksi kontur	12
2.3.	Principal Component Anaysis (PCA)	12
2.4.	Pengenalan Pola	13
2.4.1.	Pengertian pola	13
2.4.2.	Pemisahan ciri pola	16
2.4.3.	Struktur sistem pengenalan pola	16
2.5.	Jaringan saraf tiruan	17
2.6.	Algoritma K-Nearest Neighbor (KNN)	18
2.6.	Penelitian Terdahulu	20

Bab 3 Analisis dan Perancangan Sistem

3.1.	Arsitektur Umum	24
3.2.	Dataset	25
3.2.	<i>Pre-Processing</i>	26
3.3.1.	<i>Cropping</i>	27
3.3.2.	<i>Resizing</i>	27
3.3.3.	<i>Grayscale</i>	27
3.3.3.	<i>Deteksi Tepi Canny</i>	28
3.4.	<i>Segmentation</i>	29
3.4.1.	<i>Thresholding</i>	29
3.4.2.	<i>Gaussian Blur</i>	30
3.5.	Ekstraksi Fitur	31
3.5.1.	<i>Deteksi Contour</i>	31
3.5.2.	<i>Principal Component Analysis</i>	32
3.6.	Klasifikasi	33
3.7.	Perancangan Sistem	34
3.7.1.	Perancangan Menu Sistem	34
3.7.2.	Perancangan Antarmuka	35
3.7.2.1.	Perancangan Tampilan Halaman Awal	35
3.7.2.2.	Perancangan Tampilan Halaman Utama Sistem	35
3.7.2.3.	Perancangan Tambah Karakter Data Latih	37

3.7.2.4. Perancangan Database Captured	38
Bab 4 Implementasi dan Pengujian	
4.1. Implementasi Sistem	40
4.1.1. Spesifikasi Perangkat Keras dan Perangkat Lunak	40
4.1.2. Implementasi Perancangan Antarmuka	41
4.1.3. Implementasi Data	44
4.2. Prosedur Operasional	44
4.3. Pengujian Sistem	50
4.4. Analisis	53
4.4. <i>Precision</i> dan <i>Recall</i>	55
4.4. Implementasi sistem lapangan	57
Bab 5 Kesimpulan dan Saran	
5.1. Kesimpulan	58
5.2. Saran	58
Daftar Pustaka	59

DAFTAR TABEL

	Halaman
Tabel 2.1. Contoh Fitur Objek	14
Tabel 2.2. Penelitian Tedahulu	22
Tabel 3.1. Pembagian data latih dan data uj	26
Tabel 4.1. Hasil Pengujian Citra	50

DAFTAR GAMBAR

	Halaman
Gambar 2.1. Plat Nomor Kendaraan Indonesia	6
Gambar 2.2. Citra RGB (kiri) dan <i>grayscale</i> (kanan)	6
Gambar 2.3. Kontur	12
Gambar 3.1. Arsitektur Umum	25
Gambar 3.2. Citra <i>Original</i>	26
Gambar 3.3. <i>Cropping</i>	27
Gambar 3.4. <i>Grayscale</i>	27
Gambar 3.5. <i>Canny</i>	29
Gambar 3.6. <i>Binarization (Thresholding)</i>	29
Gambar 3.7. Gaussian	30
Gambar 3.8. Deteksi Kontur	32
Gambar 3.9. Pseudocode proses pelatihan metode KNN	33
Gambar 3.10. Struktur Menu Sistem	34
Gambar 3.11. Rancangan halaman awal	35
Gambar 3.12. Rancangan halaman utama	36
Gambar 3.13. Rancangan Halaman Tambah Karakter Baru	37
Gambar 3.14. Rancangan Halaman Database Captured	39
Gambar 4.1. Tampilan halaman awal	41
Gambar 4.2. Tampilan halaman utama sistem	42
Gambar 4.3. Tampilan halaman tambah karakter data latih	42
Gambar 4.4. Tampilan halaman database captured	43
Gambar 4.5. Tampilan halaman simpan data	43
Gambar 4.6. Tampilan saat tombol deteksi diklik	44
Gambar 4.7. Tampilan ketika <i>Button Start</i> diklik	46
Gambar 4.8. Citra hasil <i>cropping</i>	46
Gambar 4.9. Citra hasil <i>grayscale</i>	46
Gambar 4.10. Citra hasil deteksi tepi <i>canny</i>	47
Gambar 4.11. Citra hasil Citra hasil <i>Binarization (Thresholding)</i>	47

Gambar 4.12. Citra hasil gaussian	47
Gambar 4.13. Citra hasil Deteksi Kontur	47
Gambar 4.14. Hasil kasifikasi Metode <i>K-Nearest Neighbor</i>	48
Gambar 4.15. Tampilan halaman ketika diklik New CHAR	49
Gambar 4.16. Tampilan halaman New Data	49
Gambar 4.17. Grafik hasil akurasi pengujian sistem	50
Gambar 4.18. Deteksi plat kendaraan dengan akurasi 100%	54
Gambar 4.19. Hasil pengujian 87,5%	55
Gambar 4.20. Hasil pengujian 75%	56
Gambar 4.21. Citra plat yang telah terjadi kerusakan	56

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Transportasi merupakan bidang kegiatan yang sangat penting dalam kehidupan masyarakat Indonesia. Transportasi merupakan hal yang sudah tidak langka lagi untuk saat ini. Semua kalangan hampir mempunyai dan membutuhkan transportasi. Nomor polisi kendaraan bermotor merupakan ciri atau tanda pengenal dari suatu kendaraan yang diberikan oleh kepolisian. Peningkatan jumlah kendaraan bermotor yang terus terjadi di Indonesia tiap tahunnya, juga berbanding lurus dengan meningkatnya tingkat kecelakaan yang disebabkan pelanggaran lalu lintas di jalan raya. Dengan meningkatnya jumlah kendaraan tersebut dibutuhkan suatu sistem untuk mendeteksi plat nomor kendaraan yang melintas di jalan raya. Nantinya data hasil pendeteksian tersebut dapat digunakan oleh pihak-pihak yang membutuhkan seperti, pihak kepolisian, dinas perhubungan, maupun masyarakat untuk mengetahui pelanggaran lalu lintas maupun tindak kriminal di jalan raya.

Penelitian mengenai pengenalan plat kendaraan ini sudah pernah diterapkan sebelumnya untuk pembayaran tiket parkir secara otomatis. Wong (2013) membaca karakter pada plat nomor kendaraan untuk menemukan lokasi plat kendaraan. Nomor kendaraan yang telah terdeteksi akan dilanjutkan tahap segmentasi tiap karakternya, tiap karakter ini akan melalui tahap ekstraksi ciri menggunakan *Directional Feature Extraction* dan akan dikenali menggunakan metode *Learning Vector Quantization*. Berdasarkan hasil pengujian, keberhasilan pengenalan rata-rata 87,093%, rata-rata waktu eksekusi 4,583 detik, dan rata-rata waktu pengenalan karakter 0,28 detik. Marasi (2007) pengenalan plat nomor mobil menggunakan jaringan kompetitif dan jaringan kohonen. Dengan mengimplementasikan perangkat lunak yang dapat mengenali karakter (berupa angka dan huruf) pada plat nomor dari citra hasil pemotretan kamera digital dengan menggunakan jaringan saraf tiruan tipe kohonen

dan kompetitif sebagai metode pengenalan. Berdasarkan hasil simulasi, perangkat lunak pengenalan plat nomor yang dibuat telah berhasil mengenali pola huruf dengan tingkat akurasi 88,89%. Chaudhary (2016) menggunakan perkiraan lokasi atau gambar segmentasi metode seperti *image binary*, Metode *otsu* atau segmentasi warna. Peningkatan segmentasi karakter dilakukan dengan menghitung komponen, penggalian masing-masing karakter dan tanam karakter individu. OCR digunakan untuk mengenali karakter. Tingkat akurasi 93%. Pada penelitian-penelitian sebelumnya terdapat beberapa kekurangan diantara adalah plat yang diambil tidak *real time*, menyebabkan pendeteksian plat menjadi kurang efektif. Tidak menggunakan metode untuk menghilangkan noise pada gambar masih. Tingkat akurasi yang masih kurang.

Pada penelitian ini, deteksi pengenalan plat kendaraan dilakukan menggunakan webcam. saat plat bergerak didepan kamera, kemudian camera akan membaca plat kendaraan yang melintas. Metode yang digunakan pada penelitian ini yaitu metode *K-Nearest Neighbor*. Yang digunakan untuk meningkatkan kecepatan dan akurasi identifikasi karakter plat kendaraan dengan mengklasifikasikan objek baru berdasarkan atribut dan data training.

Berdasarkan latar belakang di atas, maka penulis mengajukan penelitian dengan judul “PENGENALAN PLAT KENDARAAN INDONESIA MENGGUNAKAN PRINCIPAL COMPONENT ANALYSIS DAN METODE *K-NEAREST NEIGHBOR*”. Hasil yang diterapkan dari penelitian ini adalah plat kendaraan dapat terbaca dan terdeteksi dengan baik di saat kendaraan bergerak maupun tidak sehingga data yang dihasilkan bagus.

1.2 Rumusan Masalah

Peningkatan jumlah kendaraan bermotor yang terus terjadi di Indonesia tiap tahunnya, berbanding lurus dengan meningkatnya tingkat kecelakaan yang disebabkan pelanggaran lalu lintas di jalan raya. Dengan meningkatnya jumlah kendaraan tersebut dibutuhkan suatu sistem untuk mendeteksi plat nomor kendaraan yang melintas di jalan raya. Nantinya data hasil pendeteksian tersebut dapat digunakan oleh pihak-pihak yang membutuhkan seperti pihak kepolisian, dinas perhubungan, tempat parkir maupun masyarakat untuk mengetahui pelanggaran lalu lintas maupun tindak kriminal di jalan raya. Oleh karena itu penelitian ini dibuat untuk mendeteksi plat kendaraan yang

melintas di jalan raya menggunakan kamera dengan menggunakan Metode *K-Nearest Neighbor*.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk pengenalan plat nomor kendaraan menggunakan Metode *K-Nearest Neighbor* agar plat kendaraan yang melintas tersimpan didatabase.

1.4 Batasan Masalah

Untuk membatasi cakupan permasalahan yang akan dibahas dalam studi ini, penulis membuat batasan :

1. Data yang diolah berasal dari citra plat kendaraan hasil capture IP Camera.
2. Citra yang diambil plat berwarna dasar hitam dan tulisan berwarna putih.
3. Teknik pengenalan menggunakan Metode *K-Nearest Neighbor*.
4. Output yang dihasilkan yaitu nomor plat kendaraan di database.
5. Plat kendaraan yang diambil hanya yang terlihat jelas

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah :

1. Mengetahui plat nomor kendaraan yang melintas.
2. Membantu polisi lalu lintas untuk mengetahui kendaraan yang melanggar lalu lintas.
3. Membantu pihak kepolisian dalam menyelediki tindak kriminal di jalan raya.
4. Mengurangi pelanggaran lalu lintas karena adanya camera yang mendeteksi di lampu merah.

1.6 Metode Penelitian

Dalam penelitian ini, tahapan-tahapan yang akan dilalui adalah sebagai berikut:

a. Studi Literatur

Metode ini diawali dengan terlebih dahulu melakukan pembelajaran literatur pada sejumlah buku, artikel, paper, jurnal, makalah, maupun situs internet mengenai pembahasan pengenalan objek, citra, algoritma *K-Nearest Neighbor*.

b. Analisis

Pada tahap ini dilakukan analisis terhadap pengenalan citra plat nomor polisi.

c. Perancangan Perangkat Lunak

Pada tahap ini, digunakan seluruh hasil analisa terhadap studi literatur yang dilakukan untuk merancang perangkat lunak yang akan dihasilkan. Dalam tahapan ini juga dilakukan perancangan model antarmuka serta proses kerja sistem untuk memudahkan dalam proses implementasi.

d. Implementasi dan Pengujian Sistem

Pada tahap ini dilakukan pemasukan data serta memproses data untuk mendapatkan hasil apakah sudah sesuai dengan yang diharapkan.

1.7 Sistematika Penulisan

Dalam penulisan skripsi ini penulis membuat sistematika sebagai berikut:

BAB 1 Pendahuluan

Bab ini membahas tentang Latar Belakang, Perumusan Masalah, Batasan Masalah, Tujuan penelitian, Manfaat Penelitian, Metode Penelitian yang dilakukan serta Sistematika penulisan.

BAB 2 Landasan Teori

Bab ini membahas tentang citra, nomor polisi kendaraan bermotor, jaringan syaraf tiruan serta algoritma *K-Nearest Neighbor*.

BAB 3 Analisis Dan Perancangan Sistem

Membahas mengenai cara kerja jaringan syaraf tiruan dalam pengenalan citra nomor plat kendaraan bermotor, arsitektur umum sistem dan perancangan antar muka pengguna.

BAB 4 Implementasi Dan Pengujian Sistem

Membahas tentang implementasi dan pengujian sistem.

BAB 5 Penutup

Bab ini membahas kesimpulan dari semua pembahasan yang ada dengan saran saran yang ditujukan bagi para pembaca atau pengembang.

BAB 2

LANDASAN TEORI

2.1 Plat Nomor Kendaraan

Plat nomor adalah salah satu jenis identifikasi kendaraan bermotor bermotor. Plat nomor juga disebut plat registrasi kendaraan, atau di Amerika Serikat dikenal sebagai plat izin (*license plate*). Bentuknya berupa potongan plat logam atau plastik yang dipasang pada kendaraan bermotor sebagai identifikasi resmi. Biasanya plat nomor jumlahnya sepasang, untuk dipasang di depan dan belakang kendaraan. Namun ada yurisdiksi tertentu atau jenis kendaraan tertentu yang hanya membutuhkan satu plat nomor, biasanya untuk dipasang di bagian belakang.

Berdasarkan Peraturan Kepala Kepolisian RI Nomor 5 Tahun 2012 Tentang Registrasi dan Identifikasi Kendaraan Bermotor, Tanda Nomor Kendaraan Bermotor (TNKB) atau sering disebut plat nomor merupakan tanda regident Ranmor yang berfungsi sebagai bukti legitimasi pengoperasian Ranmor 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 (Perkap, 2012).

Secara teknis, spesifikasi TNKB berupa 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 250×105 mm, sedangkan untuk kendaraan bermotor roda 4 atau lebih adalah 395×135 mm. Terdapat cetakan garis lurus pembatas lebar 5 mm di antara ruang nomor polisi dengan ruang angka masa berlaku (yang lama), sedangkan yang baru terdapat garis putih di sekitar TNKB dan tidak ada batas pemisah antara nomor polisi dan masa berlaku dari tahun 2011 (Perkap, 2012).

Contoh plat nomor kendaraan yang ada di Indonesia dapat dilihat pada Gambar 2.1.



Gambar 2.1. Plat Nomor Kendaraan Indonesia

2.2 Citra

2.2.1 Pengertian Citra

Citra adalah suatu fungsi intensitas warna dua dimensi $f(x,y)$, dimana x & y mewakili lokasi koordinat suatu titik dan nilai dari fungsi yang merupakan tingkat intensitas warna / tingkat keabu-abuan dari titik tersebut. (Schalkoff, 1989).

Pada citra berwarna satu nilai *pixel* terdiri atas 3 *layer* yaitu *red*, *green*, dan *blue*. Rentang nilai dari tiap *layer* berkisar antara 0-255. Sedangkan pada citra *grayscale* representasi citranya bernilai sama untuk tiap *layer*. Citra yang dihasilkan dari proses *grayscale* berupa citra abu-abu. Citra *grayscale* sangat berbeda dari citra biner karena citra biner merupakan citra yang hanya memiliki dua variasi nilai, yaitu gelap dengan nilai piksel 0 dan terang dengan nilai piksel 255 (Shapiro dan Stockman, 2001), sedangkan citra *grayscale* warnanya beragam yang merupakan rentang antara putih dan hitam. Perbandingan gambar RGB dan hasil konversinya ke citra *grayscale* dapat dilihat pada Gambar 2.2.



Gambar 2.2. Citra RGB (kiri) dan *grayscale* (kanan)

Citra dikelompokkan menjadi citra tampak dan citra tak tampak. Citra tampak dalam kehidupan sehari-hari adalah foto keluarga, lukisan, apa yang tampak di layar monitor. Sedangkan citra tak tampak misalnya gambar dalam file (citra digital), citra yang direpresentasikan dengan fungsi matematis. Agar dapat dilihat manusia, citra tak

nampak ini harus dirubah dulu menjadi citra tampak, misalnya dengan menampilkannya di monitor atau dicetak diatas kertas. Di antara citra-citra tersebut, hanya citra digital yang dapat diolah menggunakan komputer. Jenis citra lain, jika hendak diolah dengan komputer, harus diubah terlebih dahulu menjadi citra digital (Wakhidah, 2012)

2.2.2 Citra Digital

Proses perubahan citra menjadi citra digital disebut digitalisasi. Digitalisasi dapat dilakukan dengan menggunakan alat atau sensor seperti kamera, *webcam scanner* dan sebagainya. Pada umumnya citra digital berbentuk empat persegi panjang, dengan dimensi ukurannya dinyatakan sebagai lebar x tinggi (*width x height*) (Wakhidah, 2012).

Citra digital memiliki koordinat spasial, dengan tingkat kecerahan atau intensitas cahaya (skala keabu-abuan) yang memiliki numerik yang diskrit direpresentasikan dalam bentuk fungsi matematis $f(x,y)$ yang menyatakan intensitas cahaya pada titik (x,y) . Citra digital disimpan didalam berkas (file) dengan format tertentu. Format citra yang baku di lingkungan sistem operasi *Microsoft Windows* adalah berkas bitmap (bmp). Format bmp mempunyai kelebihan dari segi kualitas gambar apabila citra ditampilkan pada layar monitor, karena citra dalam format BMP umumnya tidak dimampatkan, sehingga tidak ada informasi yang hilang, walaupun akibatnya ukuran berkasnya relatif besar (Wakhidah, 2012).

Citra dalam format bmp ada 3 macam yaitu citra biner, citra berwarna dan hitam putih (*grayscale*). Citra biner hanya memiliki dua nilai keabuan, 0 dan 1. Oleh karena itu, 1 bit sudah cukup untuk merepresentasikan nilai piksel. Sedangkan citra yang lebih umum adalah citra berwarna. Adapun warna yang terlihat pada gambar dengan format bmp merupakan kombinasi dari tiga warna dasar yaitu Red, Green, dan Blue (RGB). Setiap piksel pada layar monitor disusun oleh tiga komponen warna tersebut. Kombinasi dari ketiga warna tersebut menghasilkan warna yang khas untuk satu piksel bersangkutan.

Salah satu citra warna adalah citra 24 bit, dimana setiap nilai piksel direpresentasikan dengan 24 bit. Setiap piksel langsung menyatakan kombinasi komponen warna RGB. Masing-masing komponen warna direpresentasikan dengan 8 bit. 8 bit ini merepresentasikan nilai intensitas piksel. Dengan demikian ada sebanyak 2^8 adalah 256 derajat keabuan (0-255) untuk masing-masing komponen warna. Citra

24 bit disebut juga citra 16 juta warna, karena mampu menghasilkan 224 adalah 16.777.216 kombinasi warna (Wakhidah, 2012).

2.2.3 Pengolahan Citra Digital

Pengolahan Citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Istilah pengolahan citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Dalam definisi yang lebih luas, pengolahan citra digital juga mencakup semua data dua dimensi. Meskipun sebuah citra kaya informasi, namun seringkali citra yang kita miliki mengalami penurunan intensitas mutu, misalnya mengandung cacat atau derau (noise), warnanya terlalu kontras atau kabur tentu citra seperti ini akan sulit direpresentasikan sehingga informasi yang ada menjadi berkurang (R Munir, 2004).

Agar citra yang mengalami gangguan mudah direpresentasikan maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik. Pengolahan citra adalah pemrosesan citra khususnya dengan menggunakan komputer menjadi citra yang lebih baik (R Munir, 2004). Umumnya operasi-operasi pengolahan citra diterapkan pada citra apabila :

1. Perbaikan atau memodifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau untuk menonjolkan beberapa aspek informasi yang terkandung di dalam citra.
2. Elemen di dalam citra perlu dikelompokkan, dicocokkan dan diukur.
3. Sebagian citra perlu digabung dengan bagian citra yang lain.

Pengolahan citra adalah suatu metode yang digunakan untuk mengolah citra (gambar/*image*) sehingga menghasilkan gambar lain yang sesuai dengan kebutuhan khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Meskipun sebuah citra kaya informasi, tetapi seringkali citra yang kita miliki mengalami penurunan mutu, seperti mengandung cacat atau derau, warnanya terlalu kontras, kurang tajam, kabur dan sebagainya. Sebuah citra diartikan sebagai suatu fungsi kontinyu dalam dua dimensi dari intensitas cahaya (x,y) . Dimana x dan y menyatakan suatu koordinat ruang dan f pada suatu koordinat (x,y) yang menyatakan intensitas atau tingkat kecerahan atau derajat keabuan (brightness/gray level) dan informasi warna citra pada titik koordinat x dan y .

Citra digital adalah citra *Continue* yang diubah ke dalam bentuk diskrit, baik koordinat ruang maupun nilai intensitas cahayanya. Dengan kata lain, citra digital dibuat dengan cara mencuplik suatu citra kontinyu dengan jarak seragam. Suatu titik terkecil pada citra digital sering disebut sebagai *picture element* atau *pixel*. Citra dapat berupa citra vektor ataupun citra bitmap.

a. *Resizing*

Resizing merupakan proses mengubah ukuran citra, baik memperbesar ataupun memperkecil resolusi citra (Pratt, 2007). *Resizing* juga dapat digunakan untuk menormalisasi ukuran semua citra sehingga memiliki ukuran yang sama.

b. *Grayscale*

Grayscale merupakan proses mengubah citra warna (RGB) menjadi citra keabuan. *Grayscale* digunakan untuk menyederhanakan model citra RGB yang memiliki 3 layer matriks, yaitu layer matriks *red*, *green*, dan *blue* menjadi 1 layer matriks keabuan. *Grayscale* dilakukan dengan cara mengalikan masing-masing nilai *red*, *green*, dan *blue* dengan konstanta yang jumlahnya 1, seperti ditunjukkan pada persamaan 2.1.

$$I(x,y) = \alpha.R + \beta.G + \gamma.B \quad (2.1)$$

Dimana :
 $I(x,y)$ = piksel citra hasil *grayscale*
 α, β, γ = konstanta yang hasil penjumlahannya 1
 R = nilai *red* dari sebuah piksel
 G = nilai *green* dari sebuah piksel
 B = nilai *blue* dari sebuah piksel

Green channel merupakan salah satu jenis *grayscale* yang mengganti nilai setiap piksel pada citra hanya dengan nilai *green* dari piksel citra tersebut, seperti ditunjukkan pada persamaan 2.2.

$$I(x,y) = 0.R + 1.G + 0.B = G \quad (2.2)$$

Grayscale pada citra retina menggunakan *green channel* dikarenakan citra *green channel* memiliki *contrast* yang lebih baik (Putra, 2010).

c. *Deteksi tepi Canny*

Deteksi tepi adalah operasi yang dijalankan untuk mendeteksi garis tepi (*edges*) yang membatasi dua wilayah citra yang homogen yang memiliki tingkat kecerahan yang berbeda. Metode deteksi tepi yang memiliki jangkauan deteksi yang lebih luas. Metode ini mampu mendeteksi tepi sebanyak mungkin, menentukan tepi sedekat mungkin dengan tepi asli, dan tidak melakukan deteksi tepi secara berulang sehingga mengurangi kesalahan pada saat mengidentifikasi (Suciati et al. 2013).

Terdapat beberapa tahapan dalam deteksi tepi canny, yaitu:

1. Menghilangkan noise yang ada pada citra dengan menerapkan metode Gaussian.
2. Menggabungkan metode deteksi tepi (sobel, pewan, roberts) dengan cara pencarian horizontal (G_x) dan vertikal (G_y).
3. Tentukan arah tepian.
4. Membagi kedalam empat warna sehingga arah garis yang berbeda memiliki warna yang berbeda.
5. Terapkan *nonmaximum suppression* yang bertujuan menghilangkan kemungkinan gradien suatu piksel dari edge jika piksel tersebut bukan nilai maksimum di arah *edge* tersebut sehingga diperoleh garis tepi yang lebih baik.

Salah satu algoritma deteksi tepi modern adalah deteksi tepi dengan menggunakan metode *Canny*. Deteksi tepi *Canny* ditemukan oleh Marr dan Hildreth yang meneliti pemodelan persepsi visual manusia. Pendekatan algoritma canny dilakukan dengan konvolusi fungsi gambar dengan operator gaussian dan turunan-turunannya. Turunan pertama dari fungsi citra yang dikonvolusikan dengan fungsi Gaussian (Winarno,E , 2011).

d. *Binarization (Thresholding)*

Thresholding merupakan teknik binerisasi yang digunakan untuk mengubah citra keabuan menjadi citra biner. Thresholding dapat digunakan dalam proses segmentasi citra untuk mengidentifikasi dan memisahkan objek yang diinginkan dari *background* berdasarkan distribusi tingkat keabuan atau tekstur citra (Liao, 2001). Proses thresholding menggunakan nilai batas (*threshold*) untuk mengubah nilai piksel pada *grayscale image* menjadi hitam atau putih.

Citra hasil berupa citra hitam dan putih, citra hitam (0) diperoleh apabila nilai piksel citra keabuan lebih kecil dari nilai *threshold*, sebaliknya citra putih (1) diperoleh jika nilai piksel citra keabuan lebih besar dari nilai *threshold*. Pada proses *thresholding* citra dibagi ke dalam beberapa bagian, masing-masing dari bagian tersebut memiliki ukuran yang sama besar dengan sebuah citra dibagi dengan derajat keabuan. Persamaan yang digunakan terdapat pada persamaan 2.3 (Aruan, 2017).

$$g(x, y) \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \leq T \end{cases} \quad (2.3)$$

Dimana:

$g(x, y)$ = piksel citra hasil binerisasi

$f(x, y)$ = piksel citra asal

T = nilai threshold

e. *Gaussian Blur*

Blurring (pengaburan), sering juga disebut *smoothing*, merupakan pemerataan nilai piksel-piksel tetangga yang menghilangkan detail halus dari suatu citra. Pada penelitian ini, teknik *blurring* yang digunakan adalah Gaussian Blur. Gaussian Blur merupakan teknik pengaburan suatu citra menggunakan fungsi Gaussian, yang digunakan untuk mengurangi *noise* pada citra dan mengurangi detail (Nayakwadi & Pokale, 2014). Persamaan fungsi Gaussian untuk satu dimensi ditunjukkan pada persamaan 2.4.

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (2.4)$$

Maka untuk citra dua dimensi, persamaan fungsi Gaussian berubah menjadi seperti ditunjukkan pada persamaan 2.5.

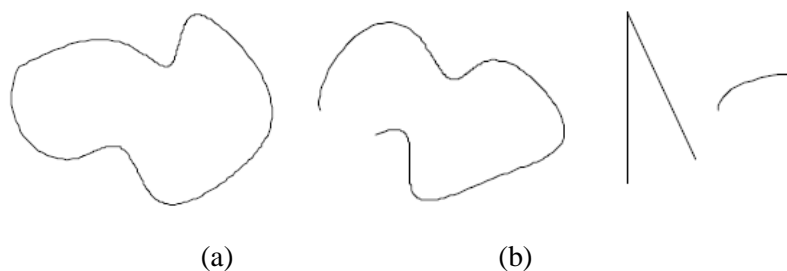
$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.5)$$

Dimana : $G(x)$ = citra hasil Gaussian Blur

f. *Deteksi Kontur*

Rangkaian *pixel-pixel* tepi yang membentuk batas daerah (*region boundary*) disebut **kontur** (*contour*) (JAI,1995). Kontur (*contour*) dapat didefinisikan sebagai suatu keadaan yang terjadi karena adanya perubahan intensitas pada piksel-piksel yang bertetangga. Dengan adanya perubahan intensitas ini, maka titik-titik tepi (*edge*) pada citra dapat terdeteksi. Kontur juga dapat didefinisikan sebagai urutan titik yang dapat menguraikan bentuk atau *region* (Williamson, A, 2014).

Kontur dapat terbuka atau tertutup. Kontur tertutup berkoresponden dengan batas yang mengelilingi suatu daerah seperti pada Gambar 2.2. (a). *Pixel-pixel* di dalam daerah dapat ditemukan dengan algoritma pengisian (*filling algorithm*). Batas daerah berguna untuk mendeskripsikan bentuk objek dalam tahap analisis citra (misalnya untuk mengenali objek). Kontur terbuka dapat berupa fragmen garis atau bagian dari batas daerah yang tidak membentuk sirkuit (Gambar 2.3.(b)).



Gambar 2.3. Kontur

2.3 Principal Component Analysis (PCA)

Ekstraksi ciri merupakan suatu proses untuk mendapatkan ciri dari sebuah objek atau citra. Untuk mendapatkan ciri dari suatu citra memerlukan suatu metode khusus. Salah satu metode yang digunakan dalam ekstraksi ciri sebuah citra adalah *Principal Component Analysis* (PCA). PCA adalah sebuah metode untuk mengambil ciri-ciri penting dari sekumpulan data dengan melakukan dekomposisi terhadap data tersebut, sehingga menghasilkan koefisien-koefisien yang tidak saling berkorelasi. PCA juga dikenal dengan transformasi Kauhunen-Loeve atau transformasi Hotelling atau teknik Eigenface (Taningrum, 2016).

PCA merupakan alat teoritis yang penting dalam teori deteksi, pengenalan pola dan image coding. Tujuan dari PCA adalah mengambil variasi total pada karakter *alphanumeric* yang dilatihkan dan menjelaskan variasi tersebut dengan variable yang jumlahnya lebih sedikit. Dengan kata lain PCA digunakan untuk merepresentasikan data dalam dimensi yang rendah sehingga waktu komputasi dapat dikurangi dan kompleksitas dari karakter *alphanumeric* yang tidak perlu dapat dihilangkan. PCA menghasilkan vektor-vektor eigen atau vector-vektor karakteristik yang kemudian akan digunakan untuk membentuk ruang eigen.

Dalam mereduksi dimensi data menjadi lebih rendah maka diperlukan penentuan vektor -vektor eigen yang dapat direduksi maupun vector-vektor eigen yang tidak dapat direduksi dengan mengurutkan vector-vektor eigen tersebut dari vektor eigen yang mempunyai nilai eigen terbesar sampai yang terkecil. Vektor-vektor eigen yang dapat direduksi adalah vektor yang mempunyai nilai eigen yang kecil. Hal ini dikarenakan nilai eigen yang kecil menandakan informasi yang dibawa oleh *vector eigen* tersebut tidak terlalu penting sehingga dapat direduksi tanpa mempengaruhi informasi penting karakteristik *alphanumeric* tersebut (Taningrum, 2016).

2.4 Pengenalan Pola

2.4.1 Pengertian Pola

Secara umum, pengertian pola (*pattern*) dapat dibedakan dengan istilah ciri fitur sebagai berikut pola adalah komposit, gabungan atau himpunan dari fitur yang merupakan sifat dari suatu objek. Ciri fitur adalah segala jenis aspek pembeda atau ciri-ciri yang membedakan. Ciri inilah yang digunakan dalam melakukan identifikasi terhadap objek yang dikenali. Oleh karena itu pemilihan fitur sangat menentukan keberhasilan dalam pengenalan pola (Putra, 2009).

Ukuran fitur diperoleh dari hasil ekstraksi fitur pada objek. Ukuran fitur bisa berwujud simbolik (misalnya warna) atau numerik (misalnya tinggi). Fitur yang bagus adalah fitur yang memiliki daya pembeda yang tinggi, sehingga pengenalan/ pengelompokkan pola berdasarkan fitur yng dapat dilakukan dengan keakuratan yang tinggi (Putra, 2009).

Contoh fitur objek terlihat pada table 2.1.

Table 2.1. Contoh Fitur Objek

Objek	Ciri / Fitur
Huruf	Tinggi, tebal, titik sudut, lengkungan garis, dll
Suara	Amplitudo, frekuensi, nada, intonasi, dll
Tanda Tangan	Kerumitan, tekanan, dll
Sidik Jari	Lengkungan, jumlah garis, dll
Buah	Bentuk, warna, dll

Pengenalan pola merupakan suatu proses yang dilakukan untuk mengelompokkan atau mengklasifikasi data numerik dan simbol. Banyak teknik statistik dan sintaksis yang telah dikembangkan untuk keperluan klasifikasi pola dan teknik-teknik ini dapat memainkan peran yang penting dalam sistem visual untuk pengenalan objek yang biasanya memerlukan banyak teknik. Bentuk-bentuk objek tertentu dalam dunia nyata yang sangat kompleks dapat dibandingkan dengan pola-pola dasar di dalam citra sehingga penggolongan objek yang bersangkutan dapat dilakukan lebih mudah

Pola adalah entitas yang terdefinisi dan dapat didefinisikan melalui ciri - cirinya (*feature*). Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola yang lain. Pengenalan pola bertujuan untuk menentukan kelompok untuk kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Dengan kata lain pengenalan pola membedakan objek dengan objek lain. Suatu sistem pengenalan pola melakukan:

1. Proses akuisisi data melalui sejumlah alat pengindraan atau sensor,
2. Mengatur bentuk representasi data,
3. Melakukan proses analisis dan klasifikasi data.

Tiga pendekatan pembuatan sistem pengenalan pola adalah Statistik (*statistical*), Sintaksis (*syntactic*) dan Jaringan Saraf Tiruan (*neural network*) (Schalkoff, 1992). Suatu aplikasi pengenalan pola bertujuan untuk melakukan proses pengenalan terhadap suatu objek (misalnya citra), salah satu hasilnya adalah mengklasifikasikan objek tersebut dalam kategori tertentu, berdasarkan pola yang dimilikinya.

1. Statistik

Semakin banyak pola yang disimpan, maka sistem akan semakin cerdas. Salah satu contoh penerapannya banyak pada pola pengenalan iris scan. Kelemahannya: hanya bergantung pada data yang disimpan saja, tidak memiliki sesuatu struktur yang unik yang dapat menjadi kunci pengenalan pola.

2. Sintaksis (*rule*)

Dengan *rule*/aturan maka sistem yang lebih terstruktur sehingga memiliki sesuatu ciri yang unik. Salah satu contoh penerapannya pada pola pengenalan sidik jari (*fingerprint*).

3. Jaringan Syaraf Tiruan (JST)

Merupakan gabungan dari pendekatan statistik dan pendekatan sintaks. Dengan gabungan dari dua metode maka JST merupakan pengenalan pola yang lebih akurat. Salah satu contoh penerapannya pada pola pengenalan suatu citra JST merupakan suatu sistem yang dapat memroses informasi dengan meniru cara kerja jaringan syaraf otak manusia.

Struktur sistem pengenalan pola ditunjukkan pada gambar 2.1 pengenalan pola ini terdiri dari suatu sensor (misalnya kamera, dan *scanner*), teknik prapengolahan, suatu algoritma atau mekanisme ekstraksi ciri dan algoritma untuk klasifikasi atau pengenalan (bergantung pada pendekatan yang dilakukan). Sebagai tambahan, biasanya beberapa data yang sudah diklasifikasikan diasumsikan telah tersedia untuk melatih sistem.

Prapengolahan adalah transformasi input (masukan) data mentah untuk membantu kemampuan komputasional dan pencarian ciri serta untuk mengurangi *noise* (derau). Pada prapengolahan citra (sinyal) yang ditangkap oleh sensor akan dinormalisasi agar citra menjadi lebih siap untuk diolah pada tahap pemisahan ciri. Kualitas ciri yang dihasilkan pada proses pemisahan ciri sangat bergantung pada hasil prapengolahan.

Klasifikasi merupakan tahap untuk mengelompokkan input data pada satu atau beberapa kelas berdasarkan hasil pencarian beberapa ciri yang signifikan dan pemrosesan atau analisis terhadap ciri itu. Setiap kelas terdiri dari sekumpulan objek yang memiliki kedekatan (kemiripan) ciri. (Munir, 2004).

2.4.2 Pemisahan Ciri Pola

Pengenalan pola digunakan untuk mengklasifikasikan atau menggambarkan sesuatu berdasarkan pengukuran kuantitatif ciri atau sifat dari objek. Pola merupakan suatu entitas yang terdefinisi dan dapat diidentifikasi dan diberi nama. Pola merupakan kumpulan hasil pengukuran yang bisa dinyatakan dengan notasi vektor atau matriks. Pemisahan ciri pada citra dilakukan berdasarkan blok. Pemisahan ciri berdasarkan blok biasanya digunakan setelah pemrosesan sebelumnya seperti proses deteksi tepi atau deteksi garis.

Citra yang telah mengalami proses deteksi tepi dibagi menjadi beberapa blok. Pembagian blok dapat dilakukan dengan dua model yaitu pembagian blok saling tumpang tindih dan pembagian blok tidak saling tumpang tindih. Pada penelitian ini digunakan model pembagian blok tidak saling tumpang tindih. Vektor dihitung dengan cara membagi citra kedalam blok berukuran tertentu sehingga akan menghasilkan blok dengan ukuran kecil. Setiap blok dilakukan penghitungan piksel. Jika dalam blok terdapat piksel dengan nilai 1 dengan jumlah minimal yang ditentukan, maka blok itu akan diberi nilai 1. Dan jika jumlah piksel bernilai 1 sebanyak kurang dari jumlah yang ditentukan, maka blok itu akan diberi nilai 0. Vektor yang dihasilkan memiliki elemen dengan nilai 0 atau 1. Proses ini bertujuan untuk menurunkan keragaman ukuran karakter angka pada nomor polisi yang disebabkan karena jarak pengambilan foto atau bentuk *style* karakter yang berbeda.

2.4.3 Struktur Sistem Pengenalan Pola

Secara umum, struktur sistem pengenalan pola adalah sebagai berikut:

- a. Sensor berfungsi untuk menangkap objek dari dunia nyata dan selanjutnya diubah menjadi sinyal digital melalui proses digitalisasi.
- b. Pra-pengolahan berfungsi mempersiapkan citra atau sinyal agar dapat menghasilkan ciri yang lebih baik pada tahap berikutnya. Pada tahap ini sinyal informasi ditonjolkan dan sinyal pengganggu (derau) diminimalisasi.
- c. Pencari dan seleksi fitur berfungsi menemukan karakteristik pembeda yang mewakili sifat utama sinyal sekaligus mengurangi dimensi sinyal menjadi sekumpulan bilangan yang lebih sedikit tetapi representatif.

- d. Algoritma klasifikasi berfungsi untuk mengelompokkan fitur ke dalam kelas yang sesuai.

2.5 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) atau *neural network* adalah suatu metode komputasi yang meniru sistem jaringan syaraf biologis. Metode ini menggunakan elemen perhitungan non-linier dasar yang disebut neuron yang diorganisasikan sebagai jaringan yang saling berhubungan, sehingga mirip dengan jaringan saraf manusia. Jaringan saraf tiruan dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran (YAN,2005).

JST ditentukan oleh 3 hal:

1. Pola hubungan antarneuron (disebut arsitektur jaringan).
2. Metode untuk menentukan bobot penghubung (disebut metode *training/learning*).
3. Fungsi aktivasi, yaitu fungsi yang digunakan untuk menentukan keluaran suatu neuron.

Jaringan syaraf tiruan telah dikembangkan sebagai generalisasi model matematika dari kognisi manusia atau saraf biologis, berdasarkan asumsi bahwa:

1. Pengolahan informasi terjadi pada elemen-elemen sederhana yang disebut neuron.
2. Sinyal dilewatkan di antara neuron melalui link koneksi.
3. Sambungan setiap link memiliki bobot terkait, dalam jaringan saraf yang khas, mengalikan sinyal yang ditransmisikan.
4. Setiap neuron menerapkan fungsi aktivasi (biasanya nonlinier) untuk input (jumlah bobot sinyal input) untuk menentukan sinyal output (Fausett, 1994).

2.6 Algoritma K-Nearest Neighbor (KNN)

Algoritma *K-Nearest Neighbor* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut (Widiarsana, O et al., 2011). KNN adalah metode yang digunakan untuk mengelompokkan objek berdasarkan contoh pelatihan terdekat di

ruang fitur. K-Nearest Neighbor merupakan jenis yang paling dasar dari contoh based learning atau lazy learning juga termasuk kelompok instance-based learning. K-Nearest Neighbor dilakukan dengan mencari kelompok objek dalam data training yang paling dekat (mirip) dengan objek pada data baru atau data testing. (Krati, 2014).

Tujuan dari metode ini yaitu untuk mengklasifikasikan objek baru melalui atribut dan contoh dari data training. Algoritma ini dihitung berdasarkan jarak minimum dari data baru ke data latih (Chandra et al. 2013). Kemudian diperoleh nilai yang paling sering muncul sebagai hasil prediksi dari data tersebut. Jauh dekat nilai ketetangga tersebut diperoleh melalui Euclidian.

K-Nearest Neighbor memiliki beberapa kelebihan yaitu tangguh terhadap training data yang noise dan efektif apabila data latih nya besar. Pada fase training, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data training sample. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk testing data atau yang klasifikasinya tidak diketahui. Jarak dari vector baru yang ini terhadap seluruh vector training sample dihitung dan sejumlah k buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut. Ketepatan algoritma K-Nearest Neighbor sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. (Alfian, 2014).

Algoritma K-Nearest Neighbor adalah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama dengan berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada yang memiliki kesamaan (similarity). Tujuan dari algoritma ini untuk mengklasifikasikan objek baru berdasarkan atribut dan training sample. *Classifier* tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori (Emma, 2009).

Ketepatan algoritma *K-Nearest Neighbor* sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur agar performa klasifikasi menjadi lebih baik.

Langkah-langkah untuk menghitung metode *K-Nearest Neighbor* antara lain:

1. Tentukan parameter K
2. Hitung jarak antara data yang akan dievaluasi dengan semua pelatihan

3. Urutkan jarak yang terbentuk
4. Tentukan jarak terdekat sampai urutan K
5. Pasangkan kelas yang bersesuaian
6. Cari jumlah kelas dari tetangga yang terdekat dan tetapkan kelas tersebut sebagai kelas data yang akan dievaluasi

$$d_i = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2} \quad (2.6)$$

Keterangan:

$x1$ = Sampel data

$x2$ = Data uji atau data testing

i = Variabel data

d = Jarak

p = Dimensi data

Nilai k yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai k yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur. Nilai k yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus khusus dimana klasifikasi diprediksikan berdasarkan training data yang paling dekat (dengan kata lain, $k=1$) disebut algoritma *Nearest Neighbor*.

Kelebihan algoritma *K-Nearest Neighbor* yaitu:

1. Tangguh terhadap *training* data yang memiliki banyak *noise*.
2. Efektif apabila *training* datanya besar.

2.7 Penelitian Terdahulu

Beberapa penelitian sebelumnya telah dilakukan, diantaranya pengenalan karakter pada plat nomor kendaraan bermotor dengan *learning vector quantization*. Pada penelitian tersebut aplikasi akan mengenali karakter dengan membaca setiap karakter pada nomor kendaraan. Dalam mengenali karakter, dimanfaatkan deteksi tepi sobel untuk menemukan lokasi plat kendaraan.

Nomor kendaraan yang telah terdeteksi akan dilanjutkan tahap segmentasi tiap karakternya, tiap karakter ini akan melalui tahap ekstraksi ciri menggunakan *Directional Feature Extraction* dan akan dikenali menggunakan metode *Learning Vector Quantization*. Untuk melakukan pengujian terhadap aplikasi ini dilakukan dengan pengujian *K-Fold Cross Validation*. Berdasarkan hasil pengujian, aplikasi mampu mengenali karakter dengan tingkat keberhasilan pengenalan rata-rata 87,093%, rata-rata waktu eksekusi 4,583 detik, dan rata-rata waktu pengenalan karakter 0,28 detik.

Penelitian selanjutnya yaitu perangkat lunak pengenalan plat nomor mobil menggunakan jaringan kompetitif dan jaringan kohonen. Dengan mengimplementasikan perangkat lunak yang dapat mengenali karakter (berupa angka dan huruf) pada plat nomor dari citra hasil pemotretan kamera digital dengan menggunakan jaringan saraf tiruan tipe kohonen dan kompetitif sebagai metode pengenalan. Berdasarkan hasil simulasi, perangkat lunak pengenalan plat nomor yang dibuat telah berhasil mengenali pola huruf dengan tingkat akurasi 88,89% dan pola angka dengan tingkat akurasi 98,3% dan pengenalan plat dengan tingkat akurasi 60%.

Penelitian berikutnya menggunakan algoritma pengenalan segmentasi karakter untuk pengenalan plat nomor kendaraan. Algoritma yang dibuat dalam tulisan ini dapat mengelompokkan karakter gambar secara akurat. *Preprocessing* dapat meningkatkan keakuratan segmentasi. Algoritma untuk segmentasi horisontal, menggunakan transformasi Hough, dapat memecahkan masalah rotasi, dan iluminasi varians. Penelitian sebelumnya dibatasi dengan segmentasi vertikal. Tingkat akurasinya mencapai 95%.

Penelitian selanjutnya menyajikan metode yang menggunakan segmentasi yang ditingkatkan dengan OCR untuk mengenali plat nomor kendaraan. Ekstraksi ini dilakukan menggunakan perkiraan lokasi atau gambar segmentasi. Metode seperti *image binary*, metode otsu atau segmentasi warna. Peningkatan segmentasi karakter dilakukan dengan menghitung komponen, penggalan masing-masing karakter dan tanam karakter individu. OCR digunakan untuk mengenali karakter. Tingkat akurasi 93%.

Penelitian berikutnya Patel, Seneha G, pada tahun 2013 menggunakan jaringan saraf tiruan *backpropagation* untuk pengenalan plat nomor kendaraan. Metode segmentasi karakter plat kendaraan dan ekstraksi diuji berdasarkan pada deteksi tepi

dan morfologi matematika yang digunakan untuk mendapatkan kontur gambar yang halus. Deteksi tepi pada penelitian ini menggunakan metode sobel dan pengenalan karakter menggunakan jaringan saraf tiruan *backpropagation*.

Di tahun 2013, Er. Kavneet Kaur et al, melakukan penelitian tentang pengenalan plat nomor kendaraan menggunakan teknik OCR (Optical Character Recognition). OCR digunakan untuk mengenali optikal hasil plat nomor karakter yang didasarkan pada pencocokan template. Algoritma ini diuji pada gambar kendaraan. OCR adalah tahap terakhir dalam pengenalan plat nomor kendaraan. Dalam tahap ini karakter pada nomor plat akan dikonversi ke teks.

Penelitian selanjutnya pada tahun 2002 Halina Kwaśnicka et al, melakukan penelitian pengelompokkan dan pengenalan plat kendaraan menggunakan gambar dari kamera. Pada penelitian ini peneliti menggunakan jaringan syaraf tiruan. Ekstraksi fitur pada penelitian ini menggunakan Connected component analysis yang akan melakukan proses filter, yang akan menentukan objek terlabel mana yang termasuk karakter pada plat nomor yang dicari. Proses klasifikasinya menggunakan OCR (Object character Recognition).

Penelitian lainnya Pada Tahun 2015, Dinesh Bhardwaj et al. melakukan penelitian tentang pengenalan plat nomor kendaraan secara real-time yang bertujuan untuk mengidentifikasi karakter secara langsung. Deteksi tepi pada penelitian ini menggunakan canny, kemudian dilanjutkan dengan Hough transform untuk mendeteksi garis pada citra dan *BLOB detection* untuk mendeteksi daerah dengan tingkat kecerahan yang lebih tinggi. Pada penelitian tersebut peneliti menggunakan ekstraksi fitur *Connected Component Analysis* (CCA) dan menggunakan perhitungan metode *mathematical morphology*. Penelitian ini berhasil mencapai akurasi 93% dalam pengenalan plat kendaraan ini.

Di Tahun 2017, Masood, Syein Z, et al. Melakukan penelitian mengenai deteksi plat nomor kendaraan and pengenalan menggunakan *Convolutional Neural Networks* dengan efisiensi akurasi yang bagus. Penelitian ini berhasil mendapatkan akurasi sebesar 94,55%.

Dari penelitian-penelitian yang telah dilakukan sebelumnya terdapat beberapa kelemahan dari penelitian tersebut diantaranya adalah plat yang diambil tidak *real time*, menyebabkan pendeteksian plat menjadi kurang efektif. Tidak menggunakan metode untuk menghilangkan noise pada gambar masih. Tingkat akurasi yang masih kurang.

Penelitian terdahulu yang telah dipaparkan diatas akan diuraikan secara singkat pada Tabel 2.2.

Tabel 2.2. Penelitian Terdahulu

No	Peneliti	Tahun	Judul	Akurasi
1	Halina Kwaśnicka et al	2002	<i>License plate localization and recognition in camera pictures</i>	-
2	Mellolo, Ottopianus	2012	<i>Pengenalan plat nomor polisi kendaraan bermotor</i>	79,43%.
3	Wong Ng Poi, et al.	2013	<i>Aplikasi pengenalan karakter pada plat nomor kendaraan bermotor dengan learning vector quantization</i>	87,093%,
4	Patel,Seneha G	2013	<i>Vehicle license plate recognition using Morphology and neural network</i>	-
5	Er. Kavneet Kaur et al.	2013	<i>Number plate recognition using ocr technique</i>	-
6	Siddhartha Choubey, et al	2015	<i>License Plate Recognition for High Security Registration Plates : A Review</i>	95%
7	Dinesh, Bhardwaj et al	2015	<i>Automated Number Plate Recognition Techniques</i>	93%
8	Chaudhary et al	2016	<i>Automatic License Plate Recognization System Using LabVIEW: Review</i>	93%.
9	Panchal Tejendra, et al	2016	<i>License Plate Detection using Harris Corner and CharacterSegmentation by Integrated Approach from an Image</i>	93,84%
10	Masood, Syed Z, et al	2017	<i>License Plate Detection and Recognition Using Deeply Learned Convolutional Neural Networks</i>	94,55%

Pada penelitian ini, penulis mengajukan untuk membuat *Pengenalan Plat Nomor Kendaraan* menggunakan *Principal Component Analysis* dengan *Metode K-Nearest Neighbor*.

BAB 3

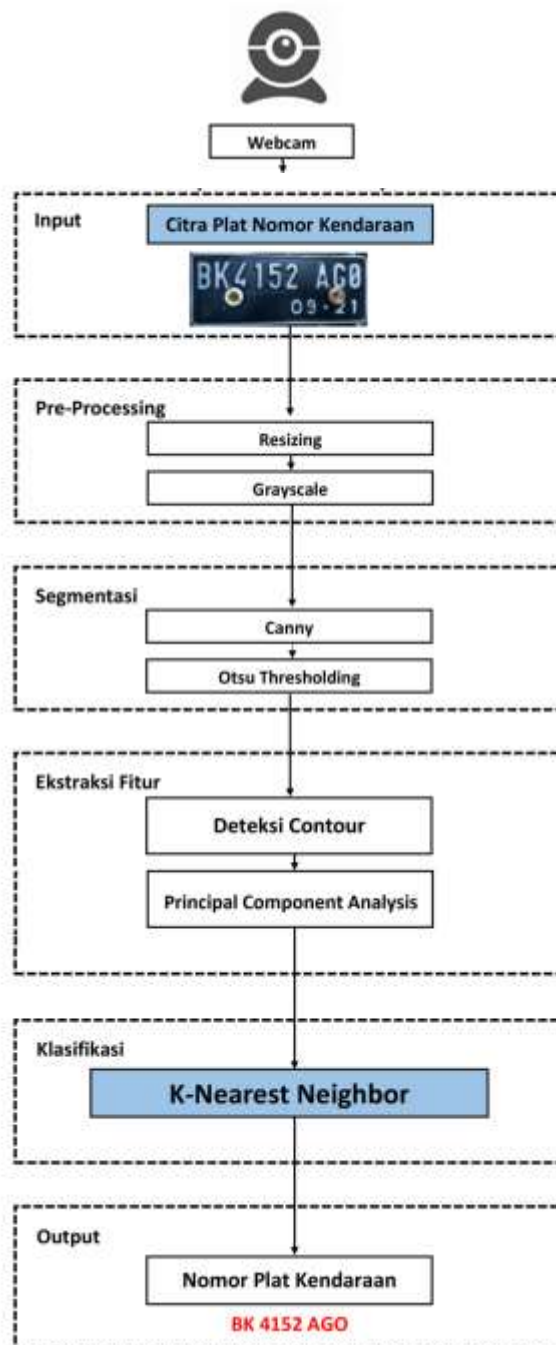
ANALISIS DAN PERANCANGAN SISTEM

Bab ini akan membahas tentang analisis dan perancangan pada sistem. Tahap pertama yaitu analisis data yang digunakan, analisis terhadap tahapan-tahapan pengolahan citra yang diterapkan, *feature extraction*, beserta Algoritma *K-Nearest Neighbor (KNN)* yang digunakan dalam mendeteksi plat kendaraan pada image . Pada tahapan selanjutnya yaitu dilakukan perancangan tampilan antarmuka sistem.

3.1. Arsitektur Umum

Metode yang akan digunakan untuk mendeteksi plat kendaraan pada *image* terdapat beberapa tahapan. Tahapan-tahapan tersebut dimulai dari pengambilan gambar yang masih original yang akan digunakan sebagai data latih maupun data uji, kemudian setelah citra plat sebagai data latih dan uji didapat maka dilakukan tahap *preprocessing* yang akan dimulai dari proses *grayscaling* yang bertujuan untuk mengkonversi citra berwarna menjadi citra skala keabuan, setelah citra berubah menjadi citra keabuan kemudian dilakukan proses selanjutnya yaitu proses pendeteksian tepi citra menggunakan deteksi tepi *canny* yang bertujuan untuk mendeteksi garis tepi (*edges*) yang membatasi dua wilayah citra yang homogen yang memiliki tingkat kecerahan yang berbeda, kemudian dilakukan proses *binarization (thresholding)*, *binarization* digunakan untuk membedakan gambar teks dengan latar belakang pada gambar huruf atau angka. Proses ini bertujuan untuk menghasilkan citra hitam putih dengan kata lain metode ini mengkonversi citra *gray-level* (citra keabuan) menjadi citra *bilevel* (citra biner). Tahap selanjutnya yaitu deteksi kontur, dengan mencari kontur yang memiliki 4 titik sudut kemudian dihitung nilai rasio dari kontur tersebut, kontur yang memiliki rasio antara 1.5 sampai 3.0 tersebut dinyatakan sebagai plat. Tahap selanjutnya ekstraksi fitur menggunakan *Principal Component Analysis (PCA)*, yang bertujuan untuk

mendapatkan ciri dari citra plat kendaraan. Tahap selanjutnya melakukan *cropping* dan pendeteksian huruf-huruf yang ada pada plat menggunakan algoritma *K-Nearest Neighbor (KNN)*. Setelah tahapan-tahapan tersebut dilakukan sehingga akan didapatkan hasil deteksi plat kendaraan menggunakan *image*. Adapun tahapan-tahapan diatas dapat dilihat dalam bentuk arsitektur umum pada Gambar 3.1.



Gambar 3.1 Arsitektur Umum

3.2. Dataset

Data citra yang digunakan dalam penelitian ini adalah citra yang diambil secara langsung di tempat parkir. Citra yang diambil dari tempat parkir merupakan citra yang digunakan untuk data latih dan data uji. Citra data uji yang digunakan ini diambil menggunakan kamera digital Samsung 12 MP. Dengan resolusi gambar 1080x810 pixel dan dengan format jpg.

Data citra yang digunakan untuk data uji sebanyak 25 citra, dan data latih sebanyak 100 citra dengan kualitas gambar yang baik. Seluruh data berjumlah 120 citra dengan pembagian citra ditunjukkan pada Tabel 3.1.

Tabel 3.1. Pembagian data latih dan data uji

No	Dataset	Jumlah Data
1	Data Pelatihan	100
2	Data Pengujian	25

Adapun contoh data uji dapat dilihat dari Gambar 3.2. Citra Original



Gambar 3.2 Citra *Original*

3.3. Pre-processing

Tahapan ini merupakan tahap pengolahan citra yang bertujuan untuk menghasilkan citra yang lebih baik untuk diproses ketahapan selanjutnya. Tahapan *preprocessing* ini terdiri dari *resizing*, *grayscale*, deteksi tepi *canny*, *thresholding*.

3.3.1. Cropping

Cropping merupakan proses pemotongan citra pada koordinat tertentu pada area citra. Untuk memotong bagian dari citra digunakan dua koordinat, yaitu koordinat awal yang merupakan awal koordinat bagi citra hasil pemotongan dan koordinat akhir yang merupakan titik koordinat akhir dari citra hasil pemotongan. Pada penelitian ini *cropping* dilakukan secara manual dengan mengcrop plat kendaraan yang memiliki ratio 1:3. Proses pengcropan dapat dilihat pada gambar 3.3.



Gambar. 3.3 Citra hasil *cropping*

3.3.2. Resizing

Resizing merupakan proses mengubah ukuran citra, baik memperbesar ataupun memperkecil resolusi citra, dalam hal ini citra plat kendaraan diubah ukurannya menjadi 216 x 106 piksel.

3.3.3. Grayscale

Citra plat yang diinput awal yang merupakan citra original dengan resolusi 1080x810 dengan format jpg. Kemudian dilakukan proses tahap awal yaitu *grayscale*. *Grayscale* dilakukan dengan tujuan untuk mengkonversi citra berwarna menjadi citra skala keabuan. Seperti yang terlihat pada Gambar 3.4 *Grayscale*



Gambar 3.4 *Grayscale*

Pada gambar 3.4 terlihat gambar yang sama seperti pada Gambar 3.2 telah di ubah menjadi *grayscale* seperti yang terlihat di Gambar 3.4. Citra berwarna telah diubah menjadi citra berskala keabuan. Pada penelitian ini, digunakan fungsi *cvtColor()* yang tersedia pada library OpenCV untuk mengubah citra asli menjadi citra grayscale. Berikut ini adalah penjelasan tentang fungsi tersebut.

Imgproc.cvtColor(source, destination, Imgproc.COLOR_RGB2GRAY);

Parameter :

- *Source* : Citra masukan, yaitu citra hasil *Resizing*
- *Destination* : Citra keluaran, yaitu citra hasil *grayscale*
- *Imgproc.COLOR_RGB2GRAY* : Fungsi untuk mengconvert citra asli menjadi citra keabuan.

3.3.4 Deteksi tepi Canny

Setelah melakukan proses *grayscale*, proses berikutnya yaitu melakukan deteksi tepi menggunakan metode *canny*. Deteksi tepi menggunakan metode *canny* bertujuan untuk mendeteksi garis tepi (*edges*) yang membatasi dua wilayah citra yang homogen yang memiliki tingkat kecerahan yang berbeda. Metode ini digunakan karena memiliki kemampuan untuk meletakkan dan menandai semua tepi yang ada sesuai dengan pemilihan parameter-parameter konvolusi yang dilakukan. Sekaligus juga memberikan fleksibilitas yang sangat tinggi dalam hal menentukan tingkat deteksi ketebalan tepi sesuai yang diinginkan.

Pada penelitian ini, digunakan fungsi *Canny()* yang tersedia pada library OpenCV untuk menentukan tingkat deteksi ketebalan tepi dari citra plat. Berikut ini adalah penjelasan tentang fungsi tersebut.

Canny(destination, destination, 0, 255)

Parameter :

- *Destination* : Citra masukan, yaitu citra hasil proses *grayscale*
- *Destination* : Citra keluaran, yaitu citra hasil *canny*
- *0,255* : Interval pencarian nilai thresholding tepi untuk pembentukan garis tepi *canny*

Berikut ini merupakan hasil dari deteksi tepi *canny*, terlihat pada Gambar 3.5.



Gambar 3.5 Canny

Pada Gambar 3.5 terlihat hasil proses deteksi tepi menggunakan metode *canny*. Metode ini dapat meningkatkan penampakan garis batas suatu daerah atau objek dalam sebuah citra seperti yang terlihat pada gambar. *Canny* juga mencirikan batas objek dan berguna untuk proses segmentasi dan identifikasi objek. Dengan *canny* dimungkinkan dapat mendeteksi tepian yang sebenarnya dengan tingkat *error* yang minimum dengan kata lain, operator *canny* di desain untuk menghasilkan citra tepian yang optimal.

3.4. Segmentation

Segmentasi merupakan proses partisi gambar digital ke beberapa daerah citra dengan tujuan untuk menyederhanakan ataupun merubah representasi citra menjadi lebih mudah dianalisa untuk ketahap selanjutnya

3.4.1. Thresholding

Tahap selanjutnya yaitu membedakan gambar text dengan latar belakang pada gambar huruf atau angka tersebut menggunakan *binarization (thresholding)*. Dengan tujuan untuk menghasilkan citra hitam putih yang bersih dari tingkat keabuan (*grayscale*), atau dengan kata lain metode ini mengkonversi citra *gray-level* ke citra *bilevel (binary image)*. Hasil dari proses ini dapat terlihat pada Gambar 3. 6



Gambar 3. 6 Binarization (Thresholding)

Pada Gambar 3.5 terlihat hasil dari proses *thresholding* dimana proses *thresholding* dapat disajikan dalam histogram citra untuk mengetahui penyebaran nilai-

nilai intensitas piksel pada suatu citra/bagian tertentu dalam citra sehingga untuk citra bimodal, histogram dapat dipartisi dengan baik (segmentasi objek dengan *background*) dan dapat ditentukan nilai *threshold*-nya. Pada penelitian ini, digunakan fungsi *threshold()* yang tersedia pada library OpenCV untuk mengubah citra menjadi citra *biner*. Berikut ini adalah penjelasan tentang fungsi tersebut.

*Imgproc.threshold(destination, destination, 0, 255, Imgproc.THRESH_BINARY +
Imgproc.THRESH_OTSU);*

Parameter :

- *Destination* : Citra masukan, yaitu citra hasil proses *Canny*
- *Destination* : Citra keluaran, yaitu citra hasil *threshold*
- 0,255 : Interval pencarian nilai *thresholding*
- *Imgproc.THRESH_BINARY + Imgproc.THRESH_OTSU* : Menerapkan ambang batas elemen array interval nilai *thresholding*.

3.4.2. Gaussian Blur

Proses *gaussian blur* dilakukan untuk menghilangkan *noise* yang masih terdapat pada citra setelah proses dilasi, agar tidak terdefinisi sebagai titik objek. Proses gaussian dapat terlihat pada Gambar 3.7.



Gambar 3.7 *Gaussian*

Gaussian blur dapat dilakukan dengan persamaan fungsi *gaussian* untuk citra dua dimensi yaitu persamaan 2.6 yang sudah termasuk dalam fungsi *GaussianBlur()* pada library OpenCV. Berikut ini adalah penjelasan tentang penggunaan fungsi tersebut pada penelitian ini.

Imgproc.GaussianBlur(destination, destination, new Size(5, 5), 0)

Parameter :

- *Destination* : Citra masukan, yaitu citra hasil proses *Thresholding*
- *Destination* : Citra keluaran, yaitu citra hasil *GaussianBlur*
- *newSize (5,5)* : Ukuran kernel *Gaussian*
- *Core.BORDER_DEFAULT* :

3.5. Ekstraksi Fitur

Tahapan setelah pengolahan citra yaitu mengekstraksi fitur atau ciri dari hasil tahapan akhir pengolahan citra.

3.5.1. Deteksi Contour

Rangkaian *pixel* *pixel* tepi yang membentuk batas daerah (*region boundary*) disebut kontur (*contour*) (JAI 1995). Kontur dapat terbuka atau tertutup. Kontur tertutup berkoresponden dengan batas yang mengelilingi suatu daerah. Batas daerah berguna untuk mendeskripsikan bentuk objek dalam tahap analisis citra (misalnya untuk mengenali objek). Kontur terbuka dapat berupa fragmen garis atau bagian dari batas daerah yang tidak membentuk sirkuit. Pada penelitian ini, digunakan fungsi *findContours()* untuk melakukan deteksi contour. Berikut ini adalah penjelasan tentang fungsi tersebut.

findContours(destination, contours, hierarchy, Imgproc.RETR_EXTERNAL,
Imgproc.CHAIN_APPROX_SIMPLE);

Parameter :

- *Destination* : Citra masukan, yaitu citra hasil proses *GaussianBlur*
- *Contours* : Titik kontur yang terdeteksi. Setiap titik kontur yang terdeteksi akan disimpan didalam list titik vektor
- *Hierarchy* : Keluaran berupa titik vektor yang bersifat opsional. Dalam parameter ini disimpan berbagai informasi dari setiap titik kontur, misalnya titik kontur sebelumnya, titik kontur selanjutnya, titik kontur utama, dsb.
- *ImgProc* : Citra masukan, yaitu citra hasil proses perbaikan citra

Pada tahap ini, dilakukan pencarian bentuk maupun kontur yang memiliki 4 titik sudut. Kemudian dilakukan perhitungan rasionya. Rasio dengan nilai > 1.0 dan ≤ 3.0 , maka akan dideteksi sebagai plat. Ketika kriteria rasio memenuhi apa yang telah ditentukan plat akan dikelilingi garis merah sebagai penanda plat pertama, garis biru plat kedua dan garis hijau plat ketiga. Hasil dari proses tersebut dapat dilihat dari Gambar 3.8.



Gambar 3.8. Deteksi Kontur

Setelah didapat plat yang sesuai dengan rasionya, kemudian masuk ketahap selanjutnya yaitu ekstraksi fitur menggunakan *Principal Component Analysis*.

3.5.2. *Principal Component Analysis*

Setelah tahap deteksi contour tahap selanjutnya yaitu ekstraksi fitur. Ekstraksi ciri merupakan suatu proses untuk mendapatkan ciri dari sebuah objek atau citra. Untuk mendapatkan ciri dari suatu citra memerlukan suatu metode khusus. Salah satu metode yang digunakan dalam ekstraksi ciri sebuah citra adalah *Principal Component Analysis* (PCA).

PCA merupakan sebuah metode untuk mengambil ciri-ciri penting dari sekumpulan data dengan melakukan dekomposisi terhadap data tersebut, sehingga menghasilkan koefisien-koefisien yang tidak saling berkorelasi. Ekstraksi fitur ini dilakukan untuk mengurangi atau mereduksi dimensi data. Dengan mempertahankan sebanyak mungkin informasi dari dataset. PCA mempertahankan sebanyak mungkin informasi dari dataset yang asli. PCA memproyeksikan data sepanjang suatu arah dimana data tersebut memiliki varians yang tinggi di tentukan oleh eigenvectors.

Pada penelitian ini, digunakan fungsi *Core.PCACompute()* yang tersedia pada *library* OpenCV untuk mendapatkan ciri dari sebuah objek atau citra. Untuk mendapatkan ciri dari suatu citra memerlukan suatu metode khusus. Salah satu metode yang digunakan dalam ekstraksi ciri sebuah citra adalah *Principal Component Analysis* (PCA). Berikut ini adalah penjelasan tentang fungsi tersebut.

Core.PCACompute(mat, mean, eigenVectors);

Parameter :

- *Mat* : Nilai masukan yang berasal dari *mean* dan *eigenvector* yang akan diproses ke tahap selanjutnya
- *Mean* : Keluaran nilai eigen vector yang dimaksud dan akan proses

ketahap selanjutnya.

- *Eigenvector* : Citra masukan, yaitu citra hasil proses dari *contour* yang akan diubah menjadi nilai eigen (*eigenvalue*).

3.6. Klasifikasi

Tahap selanjutnya setelah didapatkan nilai ekstraksi ciri yaitu klasifikasi dengan menggunakan metode *K-Nearest Neighbor* (KNN). Proses ini bertujuan untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Fitur yang diperoleh dari ekstraksi ciri tahap sebelumnya digunakan pada tahap ini. Tahapan ini dibedakan menjadi dua tahap yaitu pelatihan dan pengujian. Tahap pelatihan merupakan tahap validasi data. Pada proses pelatihan ditentukan sejumlah data latih dari total keseluruhan citra sebesar 60%. Adapun proses pelatihan metode *K-Nearest Neighbor* terlihat pada Gambar 3.9.

```

Klasifikasi (X,Y,x) // X: training data, Y: class dari X, x: contoh sample
for i = 1 sampai ke m
    hitung jarak  $d(X_i, x)$ 
end for
hitung I yang mengandung jarak k tetangga minimum ke  $d(X_i, x)$ 
return mayoritas kelas label untuk  $\{d(Y_i, \text{dimana } i \in I)\}$ 

```

Gambar 3.9. Pseudocode proses pelatihan metode KNN

Setelah semua data pelatihan divalidasi, proses identifikasi data pengujian dapat dilakukan sebagai berikut:

1. Tentukan parameter k = jumlah tetangga terdekat.
2. Masukkan data pengujian pertama.
3. Hitung jarak antara data baru dengan semua data yang ada didalam data latih.
4. Urutkan jarak tersebut dan tentukan tetangga mana yang terdekat berdasarkan jarak minimum ke- K .
5. Tentukan kategori dari nilai tetangga terdekat.
6. Gunakan kategori mayoritas dari tetangga terdekat tersebut sebagai nilai prediksi dari data yang baru.

Pada penelitian ini, digunakan fungsi *knn.train()* yang tersedia pada *library* OpenCV untuk melakukan training data menggunakan algoritma K-Nearest Neighbor. Berikut ini adalah penjelasan tentang fungsi tersebut.

knn.train(trainMat, ML.ROW_SAMPLE, Converters.vector_int_to_Mat(urutan));

Parameter :

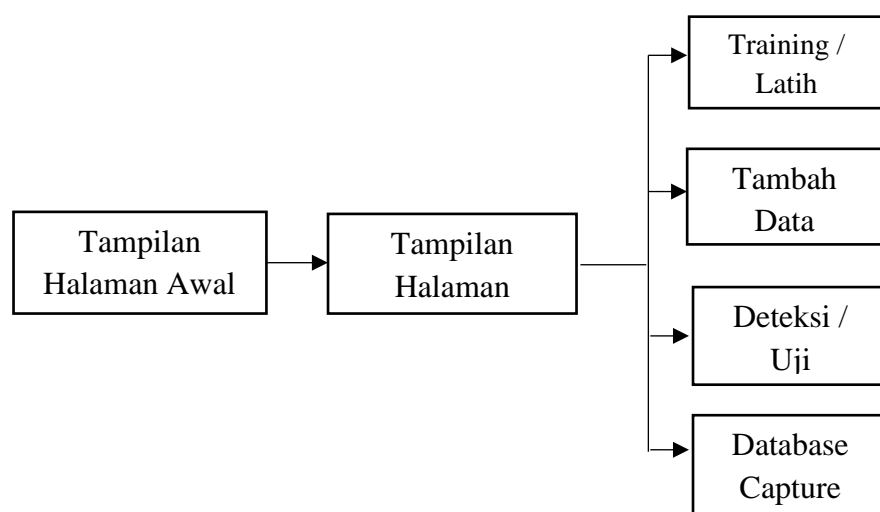
- TrainMat : Kumpulan matriks plat yang sudah di *learning*
- ML.ROW_SAMPLE : Jenis tampilan data yang tersusun seperti baris-baris/row seperti sebuah *list box*
- Converter.vector_int_to_Mat(urutan) : Mengconvert dari int ke *vector* dari urutan gambar 1 sampai 0, sebanyak gambar yang yang dilearning.

3.7. Perancangan Sistem

Pada tahapan perancangan sistem ini akan dijelaskan tentang perancangan menu sistem dan perancangan antarmuka aplikasi Deteksi Plat Kendaraan .Perancangan ini bertujuan agar pengguna dapat mudah menjalankan aplikasi.

3.7.1. Perancangan Menu Sistem

Struktur menu pada sistem dapat dilihat pada arsitektur di Gambar 3.10



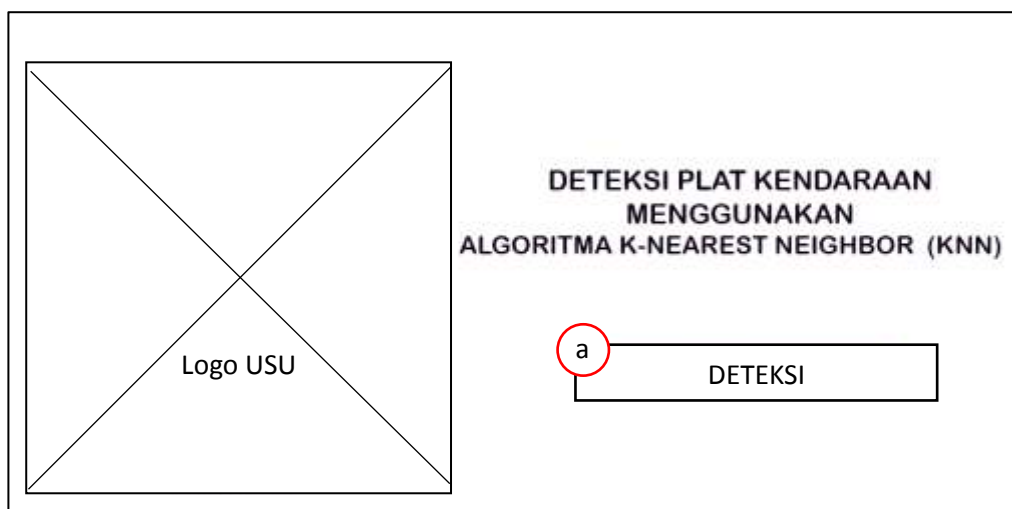
Gambar 3.10 . Struktur Menu Sistem

3.7.2. Perancangan Antarmuka

Perancangan antarmuka terdiri atas rancangan tampilan halaman awal serta tampilan halaman uji dan latih.

3.7.2.1. Perancangan Tampilan Halaman Awal

Halaman awal merupakan tampilan pertama kali ketika sistem dijalankan. Rancangan tampilan halaman awal dapat dilihat pada Gambar 3.11.

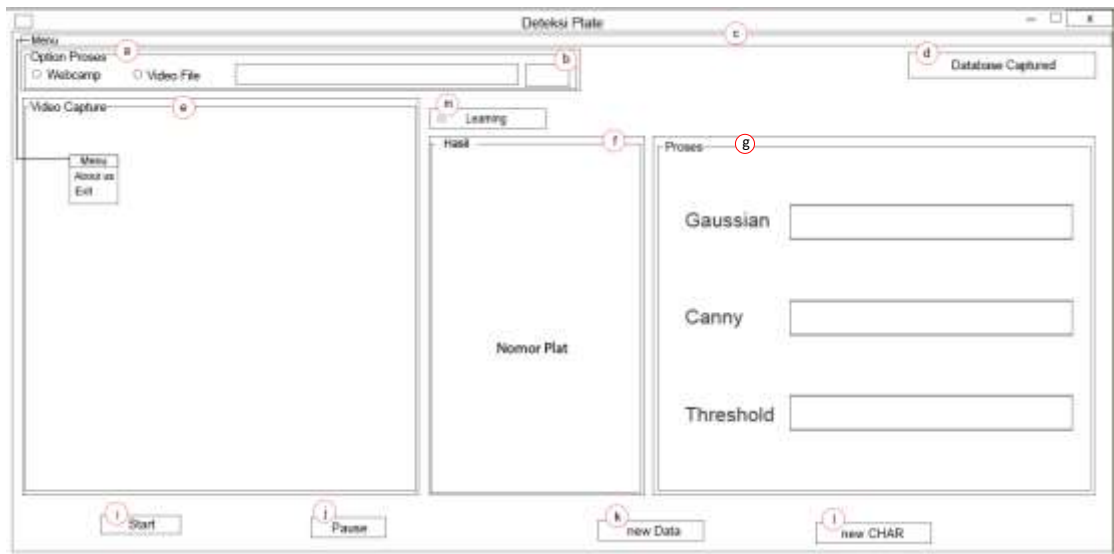


Gambar 3.11 Rancangan Halaman Awal

Keterangan : a. Tombol “Deteksi” memungkinkan pengguna untuk masuk ke halaman pengujian plat dan pelatihan plat

3.7.2.2. Perancangan Tampilan Halaman Utama Sistem

Pada tampilan halaman utama sistem merupakan tampilan untuk testing aplikasi. Ada beberapa fasilitas yang terdapat pada halaman ini yaitu pemilihan inputan citra berupa citra dari *webcam*, *video file*, maupun *select image*, pemrosesan citra plat, penambahan karakter baru, penambahan data baru, *database capture*, dan hasil deteksi plat kendaraan, juga terdapat *button start* dan *pause*. Dimana *button start* berfungsi untuk melakukan uji maupun latih. Perancangan tampilan tersebut dapat dilihat pada Gambar 3.12.



Gambar 3.12. Rancangan halaman utama

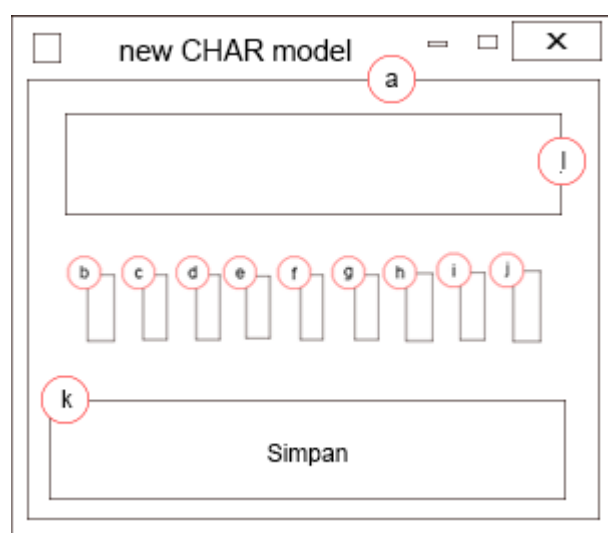
Keterangan :

- Option Proses* yang terdiri dari dua option radio button yaitu *webcam* dan *video file* yang digunakan untuk memilih proses memasukkan citra.
- Textfield* yang merupakan tempat hasil *url location file*. Dibagian *Textfield* ini juga terdapat select button yang dapat diklik sehingga akan menampilkan kotak dialog untuk memilih file citra plat kendaraan.
- Frame utama yang menampung keseluruhan panel yang terdiri dari judul frame, tombol *minimize* yang akan memperkecil ukuran jendela aplikasi, tombol *maximize* yang *disabled* karena frame sudah diatur sebesar ukuran layar, jadi tidak bisa diperbesar lagi, dan terakhir tombol *close* yang akan menutup aplikasi tersebut.
- Database Capture* yang akan menampilkan database dari data plat yang telah terdaftar yang terdiri dari nama pemilik, nomor plat, alamat, merk kendaraan, nomor mesin, nomor rangka dan tahun pembuatan.
- Video Capture* merupakan tampilan yang disediakan untuk memunculkan hasil webcam maupun *select image* yang merupakan citra plat kendaraan yang akan di proses.
- Bagian ini merupakan tampilan plat yang akan dideteksi. Yang nantinya berisi plat yang terdeteksi dan nomor plat yang terdeteksi terdapat dibagian bawah citra plat dan tersimpan di database.

- g. Bagian ini merupakan tampilan proses plat yang akan dideteksi yang terdiri dari proses Gaussian, canny dan thresholding.
- h. *Start* yang merupakan button untuk memulai melakukan proses deteksi plat baik menggunakan *webcam* maupun *video file*.
- i. *Pause* yang merupakan button untuk memberhentikan proses yang dilakukan webcam yang digunakan saat plat yang akan dideteksi telah didapatkan citranya.
- j. *new Data* merupakan *button* yang digunakan ketika ingin menambahkan data plat baru yang akan dimasukkan kedalam database *capture*.
- k. *new CHAR* merupakan *button* yang digunakan untuk melatih data dengan menambahkan karakter baru yang belum pernah dilatih sebelumnya.
- l. *Learning* merupakan button untuk melakukan learning saat *webcam* mulai aktif. Yang dapat membedakan saat melakukan *learning* maupun uji.

3.7.2.3 Perancangan Tampilan Halaman Tambah Karakter Data Latih

Tampilan halaman *new CHAR model* merupakan tampilan yang bertujuan untuk menginput karakter plat yang akan dilatih dengan memasukkan karakter ke 7 *textfield* yang ada berdasarkan citra plat yang sudah tercrop secara otomatis. Perancangan tampilan tersebut dapat dilihat pada Gambar 3.13.



Gambar 3.13. Rancangan Halaman Tambah Karakter Baru

Keterangan :

- Frame utama yang menampung keseluruhan panel yang terdiri dari judul frame, tombol *minimize* yang akan memperkecil ukuran jendela aplikasi, tombol *maximize* yang *disabled* karena frame sudah diatur sebesar ukuran layar, jadi tidak bisa diperbesar lagi, dan terakhir tombol *close* yang akan menutup halaman tersebut tersebut.
- Textfield* yang memungkinkan pengguna untuk mengisi kolom yang ada berdasarkan citra plat yang telah didapatkan dikolom b,c,d,e,f,g,h,i,j.
- Button* Simpan merupakan button yang akan memproses penyimpanan karkater yang telah diinputkan.
- Merupakan hasil citra yang telah tercrop dan akan diisi kekolom yang ada dibawahnya.

3.7.2.4 Perancangan Tampilan Halaman Database Captured

Tampilan halaman *Database Captured* merupakan tampilan yang bertujuan untuk menampilkan halaman database yang berisi data-data plat kendaraan yang telah dilakukan pendeteksian menggunakan aplikasi ini. Data-data yang dimasukkan berdasarkan data yang ada di nomor plat kendaraan. Perancangan tampilan tersebut dapat dilihat pada Gambar 3.14.

The image shows a window titled "Data Capture". Inside the window, there is a form with two main columns. The left column is labeled "Tanggal" and the right column is labeled "No Plat". At the bottom left of the form, there is a button labeled "Clear". Three red circles with letters are used as annotations: circle 'a' is around the window's title bar, circle 'b' is around the "No Plat" column header, and circle 'c' is around the "Clear" button.

Gambar 3.14. Rancangan Halaman *Database Captured*

Keterangan :

- a. Kolom tanggal merupakan kolom yang berisi tanggal pengguna saat melakukan input data
- b. Kolom ketiga berisi No plat kendaraan yang sebelumnya di input melalui halaman tambah data baru.
- c. Dibagian bawah halaman terdapat *button clear* untuk menghapus data yang ada di halaman *database captured*.

BAB 4

IMPLEMENTASI DAN PENGUJIAN

Bab ini akan membahas hasil dari implementasi algoritma *K-Nearest Neighbor* dalam mendeteksi citra plat kendaraan melalui citra dari webcam, citra plat maupun video dan pengujian sistem sesuai dengan analisis data dan perancangan yang telah dibahas pada Bab 3.

4.1 Implementasi Sistem

Pada tahap implementasi sistem, proses untuk mendeteksi citra plat kendaraan dimulai dari *preprocessing*, *segmentation*, *feature extraction*, dan hingga akhir yaitu pengklasifikasian yang diimplementasikan kedalam bahasa pemrograman Java berdasarkan perancangan yang telah dilakukan.

4.1.1. Spesifikasi Perangkat Keras dan Perangkat Lunak

Spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk membangun sistem ini adalah sebagai berikut.

1. *Processor Intel(R) Core™ i3-3217U CPU @ 1.80GHz (4CPUs), ~1.7GHz*
2. Kapasitas *hard disk* 500GB.
3. Memori RAM yang digunakan 2,00 GB.
4. Sistem operasi yang digunakan *Windows 8.1 Pro 64-bit (6.3, Build 9600)*.
5. *Eclipse Luna 3.2.0*
6. *Library* yang digunakan adalah OpenCV 3.2.0 – vc14 dan *SQLite JDBC 3.14.2.1*

4.1.2 Implementasi perancangan antarmuka

Implementasi perancangan antarmuka sistem berdasarkan analisis dan perancangan pada bab 3, yaitu sebagai berikut.

1. Tampilan halaman awal sistem

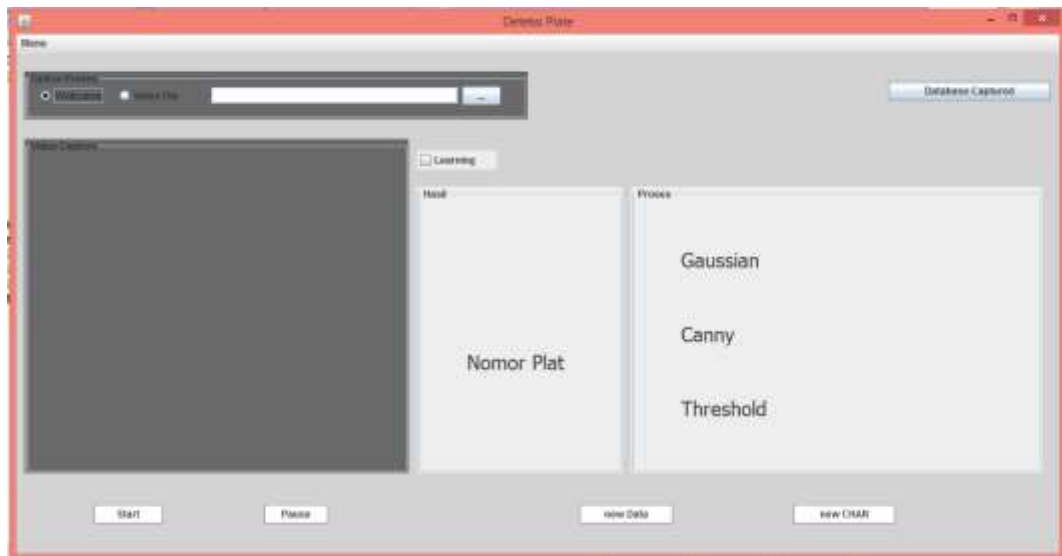
Tampilan halaman awal sistem merupakan tampilan yang pertama kali muncul ketika aplikasi dijalankan. Halaman awal dapat dilihat pada Gambar 4.1. Pada halaman ini terdapat button deteksi untuk menuju kehalaman utama sistem.



Gambar 4.1. Tampilan halaman awal

2. Tampilan Halaman Utama Sistem

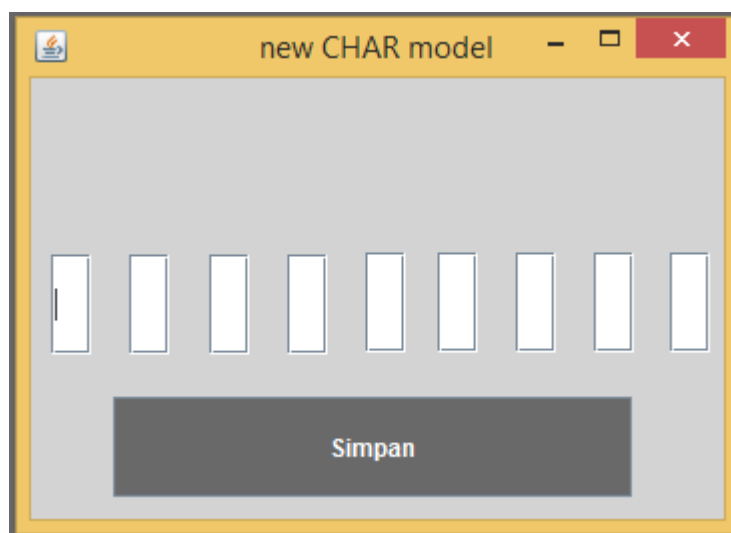
Halaman utama sistem merupakan halaman untuk uji dan latih sistem yang bertujuan untuk mendeteksi plat kendaraan melalui citra plat tersebut menggunakan algoritma *K-Nearest Neighbor*. Pada tampilan tersebut terdapat pilihan untuk melakukan training pada sistem, dapat dilakukan dengan memasukkan citra melalui webcam maupun berdasarkan video maupun foto yang telah disimpan. Di halaman tersebut juga dapat dilakukan pendeteksian plat yang berarti menguji sistem tersebut untuk mendeteksi citra plat. Tampilan tersebut dapat dilihat pada Gambar 4.2.



Gambar 4.2. Tampilan halaman utama sistem

3. Tampilan Halaman Tambah Karakter Data Latih

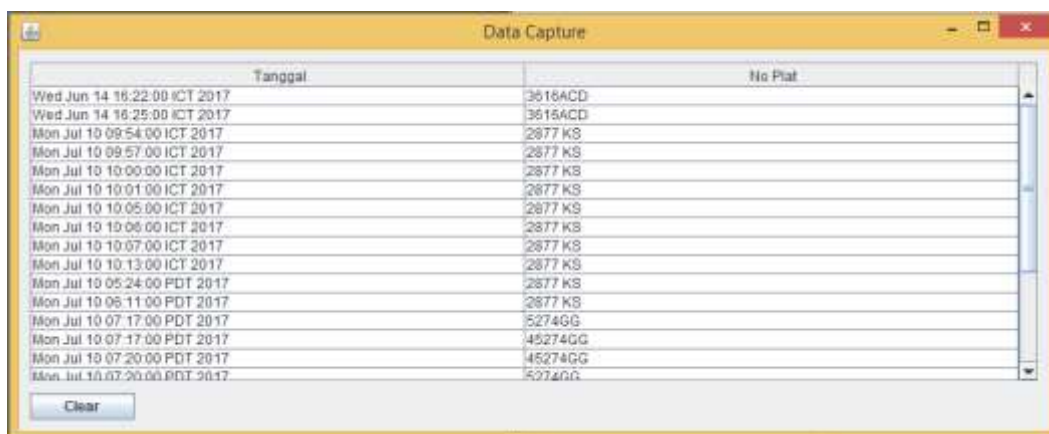
Tampilan halaman *new CHAR model* merupakan tampilan yang bertujuan untuk menginput karakter plat yang akan dilatih dengan memasukkan karakter ke *9textfield* yang ada berdasarkan citra plat yang sudah tercrop secara otomatis. 9 karakter tersebut akan diinput dan dimasukkan kedalam database chr.sqt yang disimpan per karakter sebagai platcash. Tampilan tersebut dapat dilihat pada Gambar 4.3.



Gambar 4.3 Tampilan halaman tambah karakter data latih

4. Tampilan Halaman Database Captured

Tampilan halaman *Database Captured* merupakan tampilan yang bertujuan untuk menampilkan halaman database yang berisi nomor plat kendaraan yang telah dilakukan pendeteksian menggunakan aplikasi ini. Nomor plat kendaraan tersebut dihasilkan dari plat kendaraan yang terdeteksi. Data *capture* tersebut berisi tanggal pendeteksian nomor plat yang terdeteksi. Tampilan tersebut dapat dilihat pada Gambar 4.4.



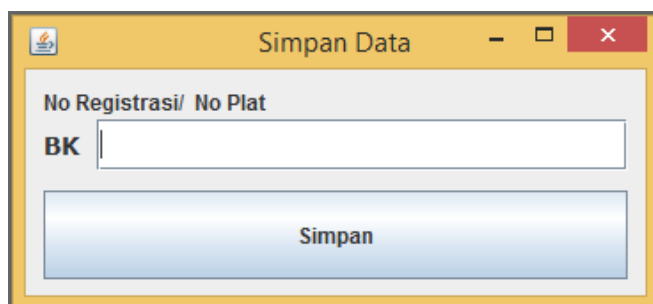
Tanggal	No Plat
Wed Jun 14 16:22:00 ICT 2017	3616ACD
Wed Jun 14 16:25:00 ICT 2017	3616ACD
Mon Jul 10 09:54:00 ICT 2017	2877 KS
Mon Jul 10 09:57:00 ICT 2017	2877 KS
Mon Jul 10 10:00:00 ICT 2017	2877 KS
Mon Jul 10 10:01:00 ICT 2017	2877 KS
Mon Jul 10 10:05:00 ICT 2017	2877 KS
Mon Jul 10 10:06:00 ICT 2017	2877 KS
Mon Jul 10 10:07:00 ICT 2017	2877 KS
Mon Jul 10 10:13:00 ICT 2017	2877 KS
Mon Jul 10 05:24:00 PDT 2017	2877 KS
Mon Jul 10 05:11:00 PDT 2017	2877 KS
Mon Jul 10 07:17:00 PDT 2017	5274GG
Mon Jul 10 07:17:00 PDT 2017	45274GG
Mon Jul 10 07:20:00 PDT 2017	45274GG
Mon Jul 10 07:20:00 PDT 2017	5274GG

Clear

Gambar 4.4. Tampilan halaman database captured

5. Tampilan Halaman Simpan Data

Halaman simpan data merupakan halaman untuk menyimpan nomor plat kendaraan yang akan disimpan ke database. Data yang disimpan merupakan nomor plat kendaraan yang didapatkan dari deteksi plat. Tampilan tersebut dapat dilihat pada Gambar 4.5.



Simpan Data

No Registrasi/ No Plat

BK

Simpan

Gambar 4.5. Tampilan Halaman Simpan Data

4.1.3 Implementasi Data

Data yang dimasukkan kedalam sistem adalah citra plat kendaraan yang diambil langsung ke jalan raya dan tempat parkir. Citra plat yang diambil merupakan citra plat yang masih bagus dan data terbaca dengan jelas. Data citra yang digunakan untuk data uji sebanyak 9 citra, dengan kualitas gambar yang baik. Seluruh data berjumlah 30 Citra.

Citra yang telah dikumpulkan kemudian dibagi menjadi dua, yakni: *training dataset* atau dataset pelatihan yang akan digunakan sebagai pembanding dalam klasifikasi dan *testing dataset* atau dataset pengujian yang digunakan untuk mengetahui akurasi dari proses klasifikasi. Dataset pelatihan berjumlah 70% dari data keseluruhan dan dataset pengujian berjumlah 30% dari data keseluruhan.

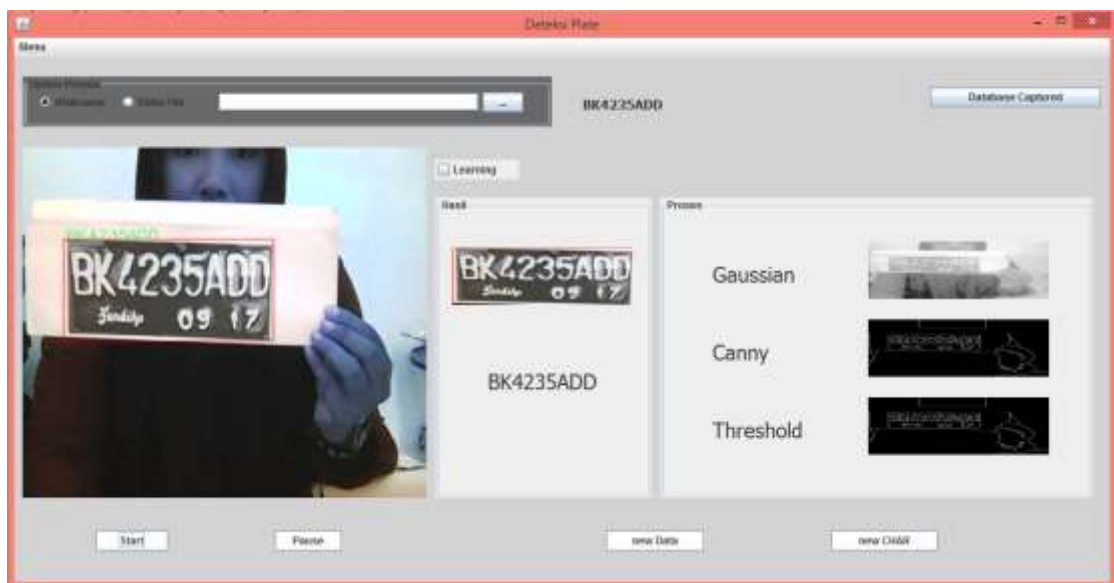


Gambar 4.6. Contoh data citra plat kendaraan

4.2 Prosedur Operasional

Tampilan awal aplikasi merupakan *splash*, setelah splash maka akan tampil halaman utama sistem. Tampilan halaman utama sistem ditunjukkan pada Gambar 4.1 yang terdapat satu menu yaitu menu deteksi yang berfungsi sebagai button untuk memulai sistem dan masuk ke tampilan utama sistem. Menu deteksi merupakan link untuk ke halaman utama. Selanjutnya, setelah masuk ke tampilan halaman utama sistem, dilakukanlah proses training data plat kendaraan. Seperti yang ditunjukkan pada

Gambar 4.5.pada halaman tersebut terdapat pilihan proses menggunakan webcam. Dan terdapat 5 button yang terdiri dari *Button Database capture, start, pause, new data* dan button *new char*. Button webcam berfungsi untuk input dengan melakukan proses pengambilan gambar menggunakan kamera external maupun internal. *Button database* memiliki fungsi untuk menampilkan data plat kendaraan yang tersimpan didalam database. Selanjutnya button *start* memiliki fungsi untuk melakukan memulai pendeteksian plat. Kemudian button *pause* berfungsi untuk memberhentikan proses deteksi yang dilakukan. Button *new data* merupakan button yang digunakan untuk menambahkan nomor plat kendaraan yang baru. yang akan dimasukkan kedalam database. Dan button *new char* yang berfungsi untuk melakukan training data dengan memasukkan tiap karakter yang ada didalam sebuah plat. Berikut tampilan ketika *Button Start* diklik dapat dilihat pada Gambar 4.7.



Gambar 4.7. Tampilan ketika *Button Start* diklik

Pada hasil dibagian sebelah kanan akan menghasilkan citra plat yang dideteksi, dan bagian proses merupakan proses pendeteksian citra plat. Proses pendeteksian tersebut didapat sebagai berikut.

1. *Cropping* merupakan proses pemotongan citra pada koordinat tertentu pada area citra. Hasil dari proses *cropping* dapat dilihat pada gambar 4.8.



Gambar. 4.8 Citra hasil *cropping*

2. *Grayscale* merupakan proses untuk mengkonversi citra berwarna menjadi citra skala keabuan. Hasil dari proses *Grayscale* dapat dilihat pada gambar 4.9.



Gambar. 4.9 Citra hasil *grayscale*

3. *Deteksi tepi Canny* merupakan proses untuk mendeteksi garis tepi (*edges*) yang membatasi dua wilayah citra yang homogen yang memiliki tingkat kecerahan yang berbeda. Hasil dari proses *Grayscale* dapat dilihat pada gambar 4.10.



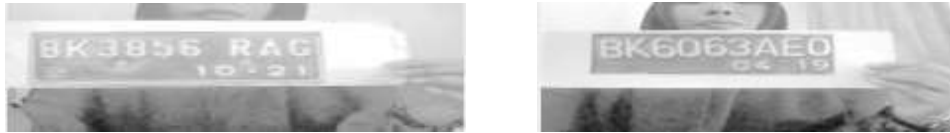
Gambar 4.10 Citra hasil deteksi tepi *canny*

4. *Thresholding* merupakan proses untuk mengkonversi citra *graylevel* ke citra *bilevel* (*binary image*). Hasil dari proses ini dapat dilihat pada Gambar 4.11.



Gambar 4.11 Citra hasil *Binarization (Thresholding)*

5. *Gaussian Blur* merupakan proses menghilangkan *noise* yang masih terdapat pada citra. Hasil dari proses ini dapat dilihat pada Gambar 4.12.



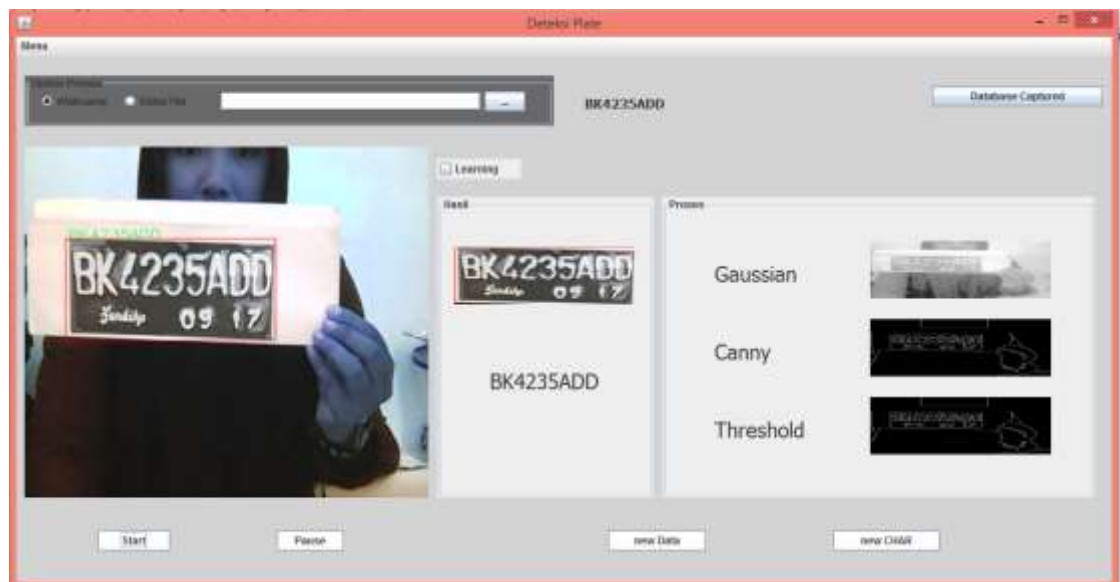
Gambar 4.12 Citra hasil *Gaussian*

6. Deteksi Contour merupakan proses pengambilan gambar berdasarkan contour yang telah ditetapkan oleh sistem. Hasil dari proses ini dapat dilihat pada Gambar 4.13.



Gambar 4.13 Citra hasil Deteksi Kontur

7. Klasifikasi menggunakan *K-Nearest Neighbor* merupakan proses klasifikasi terhadap plat berdasarkan data pembelajaran yang jaraknya paling dekat dengan karakter plat tersebut. Setelah dilakukan proses *preprocessing*, segmentasi dan ekstraksi fitur didapatkan hasil dari proses klasifikasi tersebut. Hasil dari klasifikasi dapat dilihat pada Gambar 4.14.



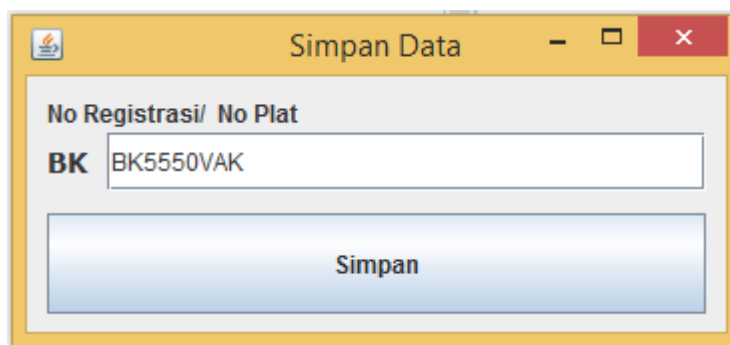
Gambar 4.14 Hasil kasifikasi Metode *K-Nearest Neighbor*

Setelah melakukan pendeteksian menggunakan *webcam*, selanjutnya melakukan pelatihan karakter yang ada diplat. Dengan mengklik button new char yang ada dihalaman utama sistem. Terdapat 9 kolom pada halaman ini yang akan diisi dengan nomor plat kendaraan yang sudah tercrop secara otomatis. Tampilan halaman tersebut dapat dilihat pada Gambar 4.15.



Gambar. 4.15. Tampilan Halaman ketika diklik New CHAR

Setelah dilakukan training dengan memasukkan plat cash 9 karakter tiap 1 plat. Selanjutnya juga terdapat button new data yang ketika diklik akan menampilkan halaman untuk menambahkan data nomor plat yang baru dideteksi. Data plat nomor tersebut selanjutnya dimasukkan ke database sebagai nomor plat yang terdaftar. Tampilan halaman tersebut dapat dilihat pada Gambar 4.16.



Gambar. 4.16. Tampilan Halaman New Data

Setelah dilakukan proses training dan penyimpanan data ke database, maka selanjutnya dapat dilakukan pengujian sistem. Pada halaman utama sistem terdapat

webcam yang juga digunakan sebagai training juga digunakan sebagai uji. Lakukan pendeteksian plat kendaraan. Plat tersebut akan muncul dikolom pertama untuk plat pertama jika rasio yang didapat sesuai dengan rasio yang telah ditetapkan. Setelah didapat citra yang sesuai dengan rasionya. Kemudian sistem yang menggunakan algoritma *K-Nearest Neighbor* akan melakukan fungsinya dengan mengklasifikasi objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Setelah hasil didapatkan kemudian nomor plat yang terdeteksi akan muncul dibagian atas citra plat. Kemudian jika nomor plat terdaftar didatabase nomor plat tersebut akan masuk kedalam kolom proses 1, yang menandakan bahwa plat tersebut sudah terdaftar.

4.3 Pengujian Sistem

Pada tahap ini akan dilakukan pengujian terhadap data dan sistem. Pengujian data dilakukan terhadap 10 citra plat kendaraan dengan menggunakan data training sebanyak 30 citra. Pengujian dilakukan dengan plat kendaraan yang bervariasi. Hasil pengujian aplikasi ini dilakukan dengan berbagai contoh variasi plat nomor kendaraan. Hasil dari pengujian dapat dilihat pada Tabel 4.1.

Tabel 4.1. Tabel pengujian sistem

No	Gambar	Hasil Deteksi	%	No	Gambar	Hasil Deteksi	%
1		BK3856RAG	100%	14		B45626ZAG	89 %
2		BK4235ADD	100%	15		BB5038HI	87,5 %
3		KK4152AGO	89%	16		BK6750MZ	100%

4		BK4452AGD	100%	17		B6 6892XZ	87,5 %
5		BK5274GG	100%	18		B94442AEW	89%
6		BK6808AGF	89%	19		BK3597AGN	100 %
7		BK6130CW	100%	20		B55550VAL	89%
8		BK6063AEO	100%	21		B02763ACB	89%
9		P1905PZB	87,5%	22		BK5700AGB	100%
10		B6703WJF	100%	23		B26889 NS	87,5 %
11		B15362QI	87,5 %	24		BZ3069 GP	100%
12		B05966KF	62,5 %	25		B26545ABW	100%
13		BK4560FF	75 %				

Berdasarkan data hasil uji yang telah dilakukan pada aplikasi Deteksi Plat Kendaraan pada image menggunakan algoritma *K-Nearest Neighbor*, dapat diperoleh nilai akurasi dengan rata-rata 92,86 %.

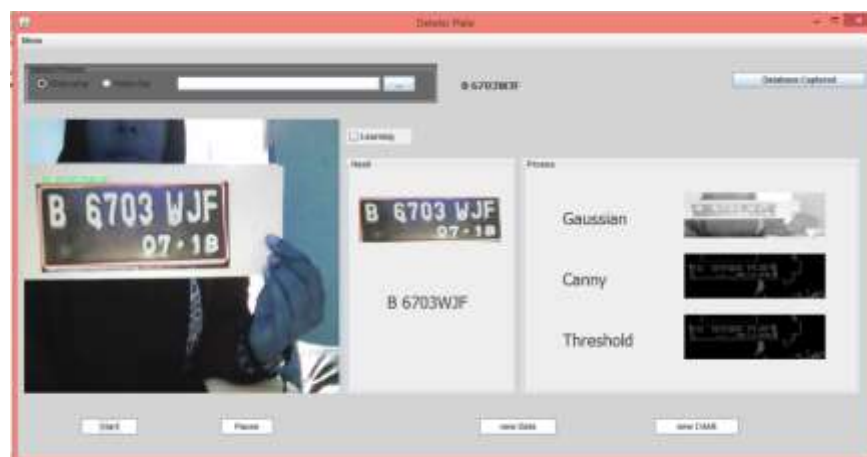
$$\begin{aligned}
 \text{Persentase akurasi} &= \frac{\text{Jumlah citra data uji benar}}{\text{Jumlah keseluruhan citra data uji}} \times 100\% \\
 &= \frac{2321.5}{25} \times 100\% \\
 &= 92,86 \%
 \end{aligned}$$

Dari perhitungan diatas dapat diketahui bahwa tingkat akurasi dari metode *K-Nearest Neighbor* dalam mendeteksi plat nomor kendaraan mencapai 92,86 %

4.4. Analisis

Berdasarkan pengujian dengan data yang bervariasi, penyebab dari rendahnya tingkat akurasi pada sistem adalah sebagai berikut :

1. Akurasi 100% yang didapatkan dari No 1 di tabel 4.1 dikarenakan pinggiran plat yang masih terlihat jelas maka proses deteksi tepi canny berjalan terdeteksi dengan baik. Tulisan pada karakter yang berwarna putih tebal dan masih terlihat cerah, Dibagian proses, Gaussian terproses dengan baik, juga karakter huruf dan angka yang terlihat merupakan tulisan standart plat yang dikeluarkan yang belum di modifikasi. Jadi, hal tersebutlah yang menjadi penyebab dapat terdeteksinya plat tersebut dengan baik. Hasil pengujian tersebut dapat dilihat pada gambar 4.17.

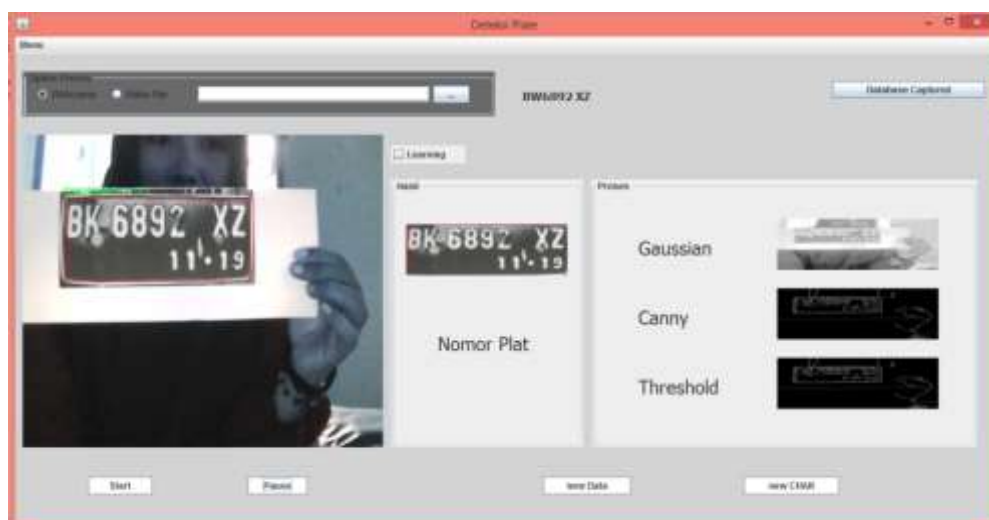




Gambar 4.17. Hasil pengujian 100%

Begitu juga dengan nomor 1,2,4,5,7,8,9,10,16,19,22,24 dan 25 yang hasil pendeteksiannya 100% dengan ciri plat yang hampir sama dengan no 1 di tabel 4.2.

2. Pada no 17 di tabel 4.2, dengan nomor plat BK6892XZ dengan akurasi yang didapatkan 87,5%. Dikarenakan plat tersebut sedikit bergelombang dibagian platnya. Sehingga terdapat cahaya yang mengganggu pendeteksian. Yang seharusnya BK6892XZ namun sistem membaca plat tersebut dengan nomor BW6892XZ. Hasil pengujian tersebut dapat dilihat pada gambar 4.18.

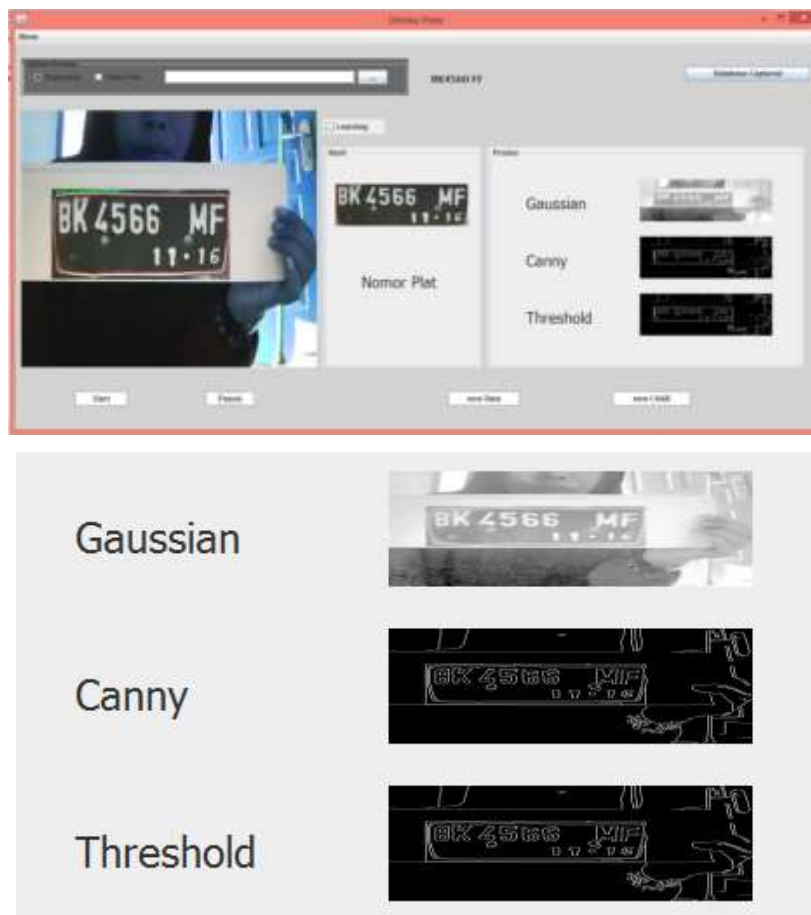




Gambar 4.18 Hasil pengujian 87,5%

Terjadi kesalahan dalam pendeteksian huruf K yang dibaca sebagai W, karena dibagian huruf K terdapat baut yang menutupi karakter tersebut. Sehingga huruf K tidak terlihat jelas ketika dideteksi. Kemudian, ketika masuk kedalam proses thresholding yaitu mengubah citra keabuan dari Gaussian menjadi citra *biner* terlihat huruf K yang tidak terlihat dengan jelas. Maka system membaca karakter tersebut dengan huruf W.

3. Di nomor 13 pada tabel 4.2, didapatkan akurasi 75% dari pengujian plat bernomor kendaraan BK4566MF, namun sistem membaca plat tersebut dengan nomor BK4560FF, hal tersebut dikarenakan pencahayaan dan kecerahan pada plat tersebut yang agak rendah yang menyebabkan sulitnya sistem dalam membaca karakter yang ada pada plat tersebut. Pencahayaan pada saat pendeteksian juga menjadi masalah utama dalam pendeteksian plat kendaraan ini, semakin gelap cahaya pada saat pendeteksian maka semakin rendah juga tingkat akurasi pendeteksian plat tersebut. Hasil pengujian tersebut dapat dilihat pada gambar 4.19.



Gambar 4.19 Hasil pengujian 75%

Jarak pada plat tersebut juga terlalu rapat dibagian karakter B dan K juga M dan F. jarak antara karakter angka 4566 ke karakter huruf MF terlalu jauh dibandingkan dengan plat standart yang ada. Karakter yang seharusnya angka 6 namun sistem membaca 0 dikarenakan angka 6 dan 0 yang terlalu rapat dan memiliki kemiripan. Begitu juga dengan karakter huruf M yang dibaca F oleh sistem dikarena kerapatan huruf dan pencahayaan pada bagian karakter huruf M. Maka dari itu dua huruf tersebut dideteksi sebagai angka 0, dan huruf F dan hal tersebutlah yang menyebabkan rendahnya tingkat akurasi pada plat nomor 13 tersebut.

4. BK4766KF yang terdapat pada nomor 12 di tabel 4.2 dengan akurasi yang didapat 62,5% merupakan contoh plat yang sudah tidak sesuai dengan ukuran yang sebenarnya, karena sudah terjadi kerusakan pada plat tersebut. Plat tersebut dapat dilihat dari gambar 4.20.



Gambar 4.20. Citra plat yang telah terjadi kerusakan

Peotnya plat tersebut yang menyebabkan gangguan pada saat pendeteksian karena rasio dari plat tersebut yang sudah berubah. Juga lekukan-lekukan pada plat yang menyebabkan masuknya cahaya dibagian lekukan tersebut. Pantulan sinar matahari pada plat yang berwarna putih yang tidak hilang selamaproses pendeteksian dan akhirnya ikut tersegmentasi dan dianggap sebagai karakter oleh sistem. Hal tersebutlah yang menyebabkan menurunnya tingkat akurasi pada pendeteksian plat.

5. Dari 25 plat yang telah diuji diatas, sebagian besar kesalahan terjadi pada pembacaan kode daerah seperti BK,BL,BB,BM,B dan sebagainya. Juga pada bagian tersebut karakter huruf kebanyakan sangat berdekatan, sehingga sistem salah dalam mendeteksi huruf tersebut.

Dari beberapa pemaparan di atas kegagalan dalam pendeteksian plat nomor kendaraan disebabkan oleh beberapa hal yaitu :

- Keadaan plat yang kurang bagus, dan sudah berubah dari bentuk aslinya.
- Pencahayaan yang tidak merata
- Jarak antara karakter plat nomor yang terlalu rapat antara satu dan yang lain
- Sulitnya membedakan karakter yang mirip, seperti angka 6 dan huruf G, angka 0 dan huruf O dan karakter-karakter lainnya.
- Posisi plat yang miring saat pendeteksian.

4.5. Analisis *Precision* dan *Recall*

Pada *pattern recognition* dan *information retrieval*, *precision* dan *recall* adalah dua perhitungan yang banyak digunakan untuk mengukur kerja dari sistem yang digunakan. *Precision* merupakan tingkat ketepatan antara informasi yang diminta oleh pengguna

dengan jawaban yang diberikan sistem. *Recall* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. *Precision* dan *recall* pada penelitian ini digunakan untuk mengukur kinerja algoritma *K-Nearest Neighbor* (KNN) yang berfungsi untuk melakukan pengenalan karakter plat nomor kendaraan. Untuk menguji sistem digunakan 25 data yang merupakan citra plat nomor kendaraan.

Tabel 4.2. Analisis Hasil Penelusuran

No	Relevan (a)	Tidak Relevan (b)	Total (a+b)	Tidak ditemukan (c)	Total (a+c)	Recall [a/(a+c)] x100%	Precision [a/(a+b)] x100%	F-Score Recall +Precision/ 2
1	15	10	25	0	15	100%	60%	80%
2	92,86 %	7. 14%	100%	0	92,86 %	100%	92,86 %	96,43%
Rata-Rata						100%	76, 34 %	88,22%

Keterangan:

- a : Relevan (plat yang didapat dan benar)
- b : Tidak relevan (plat yang didapat dan tidak benar)
- c : Tidak ditemukan (plat relevan yang tidak ditemukan)

Berdasarkan Tabel 4.5 di baris nomor 1 nilai *recall* adalah 100% dan nilai *precision* adalah 60 % dengan hasil akhir 80% berdasarkan kategori terbaca dengan benar atau tidak. Di baris nomor 2, nilai *recall* adalah 100% dan nilai *precision* adalah 92,86% dengan hasil akhir 96,43% berdasarkan kategori perhitungan per karakter. Berdasarkan tabel 4.5 rata-rata nilai *precision* adalah sebesar 76, 34% dan nilai *recall* sebesar 100% serta *F-Score* sebesar 88,22% dari skala 0%-100%.

Walaupun nilai *precision* lebih rendah dari pada nilai *recall*, tingkat keefektifan dari sistem pada penelitian ini sudah dikatakan efektif. Keefektifan suatu sistem dinilai berdasarkan teori yang dicetuskan oleh Lancaster (1991) yaitu relevan dan tidak efektif jika nilai dibawah 50%. tidak relevan. Efektifitas dibedakan menjadi dua bagian, yakni efektif jika nilai di atas 50%

4.6. Implementasi sistem dilapangan

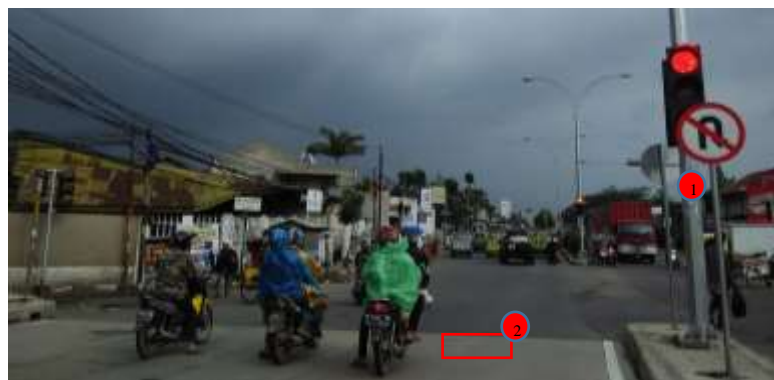
Sistem pengenalan plat kendaraan ini akan diimplementasikan ditempat parkir, yang bertujuan untuk mendeteksi plat kendaraan yang masuk ketempat parkir secara otomatis. Pembacaan plat nomor kendaraan secara otomatis membantu pegawai parkir dalam menjalani tugasnya. Karena dapat mempermudah dalam menangani para pemarkir yang akan memarkirkan kendaraannya. Berikut merupakan gambaran tempat parkir dan kamera pendeteksi plat nomor kendaraan. Gambar 4.15.



Gambar 4.15. Implementasi sistem di lapangan

Keterangan :

1. Letak kamera pendeteksi plat nomor
2. Plat nomor yang akan dideteksi



Gambar 4.16. Implementasi sistem di lapangan

Keterangan :

1. Letak kamera pendeteksi plat nomor
2. Plat nomor yang akan dideteksi

BAB V

KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan dari metode yang digunakan untuk mendeteksi plat kendaraan menggunakan citra plat serta saran yang diperlukan untuk pengembangan penelitian berikutnya.

5.1. Kesimpulan

Kesimpulan yang dapat diambil berdasarkan penelitian terhadap identifikasi jenis tanaman jamur beracun menggunakan *K-Nearest Neighbor* adalah sebagai berikut:

1. Penggunaan metode *K-Nearest Neighbor* dapat mendeteksi citra plat nomor kendaraan dengan nilai akurasi mencapai 92,86%.
2. Berdasarkan pengujian sistem, tinggi dan rendahnya tingkat akurasi ditentukan dari cahaya pada saat pendeteksian plat.
3. Deteksi tepi *canny* dan deteksi *contour* merupakan perpaduan yang baik untuk mendeteksi sebuah plat berbentuk rectangle.

5.2. Saran

Adapun saran yang diberikan oleh penulis untuk penelitian selanjutnya adalah sebagai berikut:

1. Menggunakan citra plat yang lebih baik dari segi bentuk maupun pencahayaan untuk *dataset* yang akan digunakan agar mempermudah proses pengcropan dan pendeteksian saat training data.
2. Menggunakan data pelatihan yang lebih banyak sehingga ketika di uji mampu mendapatkan akurasi yang lebih tinggi.

DAFTAR PUSTAKA

- Adfriansyah. 2012. *Pengenalan Karakter Pada Plat Nomor Kendaraan dengan Metode Backpropagation*. Medan. Universitas Sumatera Utara.
- Avianto, Donny. 2016. *Pengenalan Pola Karakter Plat Nomor Kendaraan Menggunakan Algoritma Momentum Backpropagation Neural Network*. Yogyakarta. Jurnal Informatika Vol. 10 No. 1.
- Baba, N & Sato, K. 1998. *A Consideration on the Learning Algorithm of Neural Network*. Kashiwara, Osaka Prefectur. IEEE Journal
- Bhushan, B., Singh, S. and Singla, R., 2013. License Plate Recognition System using Neural Networks and Multithresholding Technique. *International Journal of Computer Applications*, 84(5), pp.45-50.
- Chaudary, P., Dhaka, V.S. & Kumar, M. 2016. *Automatic License Plate Recognition System Using LabVIEW: Review*. Rajasthan. International Journal of Advance Research in Computer Science and Software Engineering.
- Choubey, S. & Rai, A. 2015. *License Plate Recognition for High Security Registration Plates: A Review*. Bhilai. International Journal of Computer Science and Information Technologies.
- Dinesh, Bhardwaj. & Sunil, M. 2015. *Review Paper on Automated Number Plate Recognition Techniques*. Punjab, India. International Journal of Emerging Research in Management & Technology.
- Er. K Kaur. & Vijay K, B. 2013. *Number Plate Recognition Using Ocr Technique*. Punjab, India. IJRET: International Journal of Research in Engineering and Technology.
- Fatihah, N. 2016. *Identifikasi Jenis Kayu Tropis Menggunakan Backpropagation Neural Network*. Skripsi. Universitas Sumatera Utara.
- Febriani, A. 2014. *Identifikasi Diabetic Retinopathy Melalui Citra Retina Menggunakan Modified K-Nearest Neighbor*. Skripsi. Universitas Sumatera Utara.

- Kamiyama, Naoki. 2004. *"Tuning of Learning rate, Momentum and Network scale on Backpropagation"*. Musashi Institute of Technology.
- Kwasnicka, Halina. & Bartosz Wawryzniak. 2002. *License plate localization and recognition in camera pictures*. Poland. Faculty Division of Computer Science, Wrocław University of Technology.
- Masood, Syed Z., Guang, Shu ., Afshin, D., Enrique G. Ortiz. 2017. *Lincense Plate Detection and Recognition Using Deeply Learned Convolutional Neural Networks*. Winter Park, FL. Computer Vision Lab, Sightbound Inc.
- Mellolo, Ottopianus. 2012. *Pengenalan Plat Nomor Polisi Kendaraan Bermotor*. Manado. Jurnal Ilmiah Sains Vol. 12 No. 1.
- Musoromy, Z. & Ramalingam, S. 2010. *Edge Detection Comparison for License Plate Detection*. Hatfield. ICARCV2010.
- Nurrahmadayeni. 2017. *Identifikasi Penyakit Hypertensive Retinopathy Melalui Citra Fundus Retina Menggunakan Probabilistic Neural Network*. Skripsi. Universitas Sumatera Utara.
- Munir, R. 2004. *Pengolahan Citra Digital Dengan Pendekatan Algoritmik*. Bandung. Informatika Bandung.
- Pao, Miranda Lee. 1989. Concepts of information retrieval. pp. 253-269. Libraries Unlimited Inc: United States.
- Patel, Sneha G. 2013. *Vehicle License Plate Recognition Using Morphology And Neural Network*. Gujarat, India. International Journal on Cybernetics & Informatics (IJCI) Vol.2, No.1.
- Putra, D. 2010. *Pengolahan Citra Digital*. Yogyakarta : Andi.
- Sahar S. Tabrizi., & Nadire, C. 2016. *A hybrid KNN-SVM model for Iranian license plate recognition*. Vienna, Austria. International Conference on Application of Fuzzy Systems and Soft Computing, ICAFS.
- Saunshi, Shrutika., Vishal, S., Juhi, P., Abhishek, Y., Dr. Sheetal, R. 2016. *License Plate Recognition Using Convolutional Neural Network*. Mumbai, India. IOSR Journal of Computer Engineering (IOSR-JCE).
- Sotirov, S & Krawczak, M. 2010. *"Generalized Net Model for Parallel Optimization of Multilayer Perceptron with Momentum Backpropagation Algorithm"*. Sofia. IEEE Journal.

Wong, N.P, Hardy. & Maulana A. 2013. *Aplikasi Pengenalan Karakter Pada Plat Nomor Kendaraan Bermotor dengan Learning Vector Quantization*. Medan. Sistem Informasi Indonesia.