

**PENERAPAN ALGORITMA MEANS SHIFT DALAM
LOKALISASI CITRA PLAT NOMOR POLISI DAN
SEGMENTASI KARAKTER**

SKRIPSI

oleh :

Muhammad Firdaus Ardiansyah

0510963033 - 96



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU
PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2010**

**PENERAPAN ALGORITMA MEANS SHIFT DALAM
LOKALISASI CITRA PLAT NOMOR POLISI DAN
SEGMENTASI KARAKTER**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh
gelar Sarjana dalam bidang Ilmu Komputer

oleh :

Muhammad Firdaus Ardiansyah

0510963033 - 96



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2010**

UNIVERSITAS BRAWIJAYA



LEMBAR PENGESAHAN SKRIPSI

**PENERAPAN ALGORITMA MEAN SHIFT DALAM
LOKALISASI CITRA PLAT NOMOR POLISI DAN
SEGMENTASI KARAKTER**

Oleh:

Muhammad Firdaus Ardiansyah

0510963033 - 96

Setelah dipertahankan di depan Majelis Penguji
Pada tanggal 9 November 2010
dan dinyatakan memenuhi syarat untuk memperoleh
gelar Sarjana dalam bidang Ilmu Komputer

Pembimbing I

Pembimbing II

Edy Santoso S.Si., M.Kom.
NIP. 197404142003121004

Drs. Marji, MT
NIP. 196708111992031001

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Agus Suryanto, M.Sc
NIP. 196908071994121001

UNIVERSITAS BRAWIJAYA



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Muhammad Firdaus Ardiansyah
NIM : 0510963033-96
Jurusan : Matematika
Penulis Tugas Akhir berjudul :

**PENERAPAN ALGORITMA MEANS SHIFT DALAM
LOKALISASI CITRA PLAT NOMOR POLISI DAN
SEGMENTASI KARAKTER**

Dengan ini menyatakan bahwa :

1. Isi dari Tugas Akhir yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Tugas Akhir ini.
2. Apabila dikemudian hari ternyata Tugas Akhir yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 23 November 2010
Yang menyatakan,

(Muhammad Firdaus Ardiansyah)
NIM. 0510963033

UNIVERSITAS BRAWIJAYA



Penerapan Algoritma Means Shift dalam Lokalisasi Citra Plat Nomor Polisi dan Segmentasi Karakter

ABSTRAK

Sistem pengenalan nomor polisi kendaraan bermotor banyak dilakukan oleh negara maju untuk melacak pelanggaran lalu lintas, penggunaan sistem parkir otomatis dan lain sebagainya. Dalam sistem pengenalan plat nomor, lokalisasi dan segmentasi karakter plat nomor merupakan hal yang harus dilakukan agar memperoleh kinerja yang sesuai. Hal tersebut dikarenakan dalam sebuah citra yang akan dianalisa terdapat bermacam-macam kandidat huruf atau angka yang akan dicari oleh sistem, sehingga tidaklah mungkin menganalisa semua objek tersebut untuk dikenali sebagai bagian dari karakter citra plat nomor.

Metode yang pernah dilakukan oleh Wenjing Jia, Huaifeng Zhang, dan Xiangjian He pada tahun 2005 dalam melakukan lokalisasi plat nomor, menggunakan metode *mean shift algorithm* untuk mengatasi masalah seperti kualitas gambar buruk karena kondisi pencahayaan sekitarnya, gambar distorsi perspektif, gangguan karakter, dan lain sebagainya dalam sistem pengenalan plat nomor yang ada waktu itu.

Sistem lokalisasi dan segmentasi karakter plat nomor polisi kendaraan bermotor di Indonesia dalam penelitian ini menggunakan *mean shift algorithm*, deteksi tepi sobel, dan klasifikasi mahalanobis dengan data yang telah diinputkan sebagai data latih. Kendaraan yang diuji sebanyak 20 buah mobil pribadi dengan plat hitam, dengan masing-masing mobil diambil 5 buah citra dalam beberapa sudut pengambilan.

Hasil penelitian menunjukkan bahwa sistem dapat melakukan deteksi lokasi plat nomor dengan akurasi 78% dan menunjukkan pengaruh segmentasi dengan metode *mean shift algorithm* yang cukup signifikan dalam proses penentuan lokasi plat nomor.

Kata kunci : *Mean Shift Algorithm*, lokalisasi dan segmentasi plat nomor

UNIVERSITAS BRAWIJAYA



Means Shift Algorithm Implementation in Police License Plate Localization and Character Segmentation

ABSTRACT

License plate recognition systems in vehicles is mostly done by developed countries to keep track of traffic violations, the use of automated parking systems and so forth. In a number plate recognition system, localization and segmentation of license plate characters needed in order to obtain an appropriate performance. That is because in an image to be analyzed contained a variety of candidate letters or numbers that will be searchable by the system, so it is impossible to analyze all these objects to be recognized as part of license plate character image.

Methods that have been done by Wenjing Jia, Huaifeng Zhang, and Xiangjian He in 2005 in conducting license plate localization, using the mean shift algorithm to solve problems such as poor image quality due to ambient lighting conditions, image distortion of perspective, character disorders, etc in number plate recognition systems existing at that time.

License plate localization and segmentation system in Indonesia vehicles in this research using the mean shift algorithm, Sobel edge detection, and Mahalanobis classification with data that has been input as training data. Vehicles are tested as many as 20 private cars with a black plate, with each car taken 5 pieces of the image in several angles.

The results showed that the system can perform license plate location detection with an accuracy of 78% and shows the influence of segmentation with method mean shift algorithm is quite significant in the process of determining the location of license plates.

Keywords: Mean Shift Algorithm, localization and segmentation of license plate

UNIVERSITAS BRAWIJAYA



KATA PENGANTAR

Alhamdulillah rabbil 'alamin. Puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat, taufik serta hidayah-Nya sehingga Proposal Skripsi yang berjudul “Lokalisasi dan Segmentasi Karakter Plat Nomor Polisi” dapat diselesaikan. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada program studi Ilmu Komputer, jurusan Matematika, fakultas MIPA, Universitas Brawijaya. Banyak pihak yang berperan atas terselesainya penelitian dan penulisan Skripsi ini. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Edy Santoso, S.Si., M.Kom. selaku dosen pembimbing pertama yang telah memberikan saran dan bimbingan atas penelitian skripsi.
2. Drs Marji MT. selaku dosen pembimbing kedua yang telah memberikan saran dan bimbingan atas penelitian skripsi.
3. Dr. Agus Suryanto, Msc., selaku ketua jurusan Matematika FMIPA UB Malang.
5. Segenap bapak dan ibu dosen yang telah mendidik, dan mengamalkan ilmunya kepada penulis.
6. Segenap staf dan karyawan di Jurusan Matematika FMIPA UB.
7. Ibu, Bapak, Kakak, Adik, dan Istriku yang telah memberikan semangat dan mendoakan selama belajar di Program Studi Ilmu Komputer FMIPA UB dan dalam mengerjakan skripsi.
8. Pak Wahyunanto, Rahmat, dan mas Arif, atas kesediaanya meminjamkan kamera dan lensanya,
9. Andi Permana SP S.Kom atas pinjaman jas dan dasinya.
10. Saudara-saudara seperjuangan dalam dakwah di Badan Kordinasi Lembaga Kampus dan Hizbut Tahrir atas

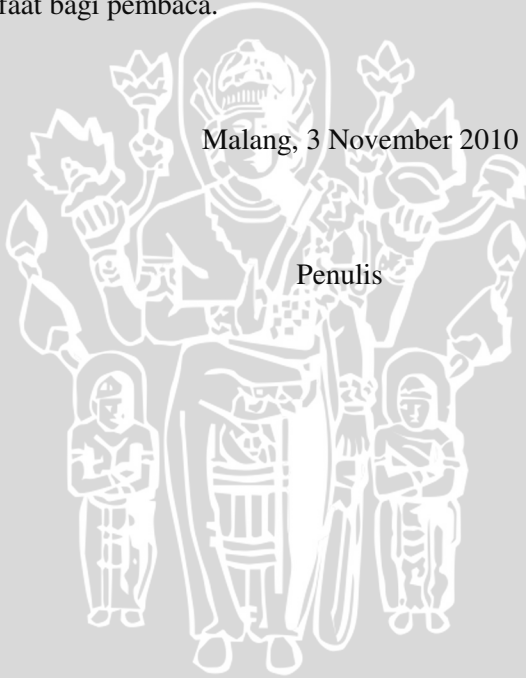
amanahnya, semoga menjadi amalan yang kelak mendatangkan keridoan Allah SWT.

11. Serta teman-teman *ilkomers* '05 yang telah memberikan semangat dan dukungan pada penulis.
12. Semua pihak lain yang telah membantu terselesaikannya skripsi ini yang tidak bisa penulis sebutkan satu-persatu.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, dan memiliki banyak kekurangan, sehingga dengan segala kerendahan hati penulis mengharapakan kritik dan saran yang membangun dari pembaca. Semoga penulisan skripsi ini bermanfaat bagi pembaca.

Malang, 3 November 2010

Penulis



DAFTAR ISI

Judul	i
Lembar Pengesahan Skripsi	iii
Lembar Pernyataan.....	v
Abstrak	vii
Abstract	ix
Kata Pengantar	xi
Daftar Isi.....	xiii

BAB I PENDAHULUAN	1
1.1 Latar belakang.....	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Metodologi Penelitian.....	4
1.7 Sistematika Penulisan	5

BAB II TINJAUAN PUSTAKA.....	7
2.1. Tanda Nomor Kendaraan Bermotor (plat).....	7
2.2. Sistem Pengenalan Plat Nomor Polisi Otomatis	8
2.3. Algoritma Pergeseran Nilai Rataan (Mean Shift Algorithm).....	8
2.4. Kepersegipanjang (<i>Rectangularity</i>).....	10
2.4.1. Rasio Persegi Panjang Plat Nomor	11
2.5. Deteksi Tepi.....	11
2.5.1. Deteksi Tepi dengan Metode Sobel	12
2.5.2. Kepadatan Tepi (<i>edge density</i>).....	13
2.6. Jarak Mahalanobis	13
2.7. Pengujian Statistik	14
2.8. Metode evaluasi	15

BAB III METODE DAN PERANCANGAN SISTEM	17
3.1. Pengumpulan serta pemilihan citra yang akan digunakan.	17
3.2. Analisa dan perancangan perangkat lunak.	18

3.2.1.	Proses pengolahan citra	18
3.2.1.1.	Segmentasi Citra dengan Algoritma Pergeseran Rataan (<i>Means Shift Algorithm</i>).	19
3.2.1.2.	Deteksi Tepi dengan Metode Sobel.....	24
3.2.1.3.	Binerisasi citra	25
3.2.1.4.	Lokalisasi Citra	26
3.2.1.5.	Klasifikasi Mahalanobis	29
3.2.1.6.	Kliping Citra Plat Nomor.....	30
3.3.	Pengujian perangkat lunak untuk menguji ketepatan penggunaan algoritma untuk penyelesaian masalah....	31
3.4.	Rancangan basis data.....	31
BAB IV METODE HASIL DAN PEMBAHASAN		33
4.1.	Lingkungan Implementasi	33
4.2.	Implementasi Basis Data	33
4.3.	Implementasi Program.....	34
4.3.1.	Segmentasi Citra.....	35
4.3.2.	Deteksi Tepi.....	37
4.3.3.	Binarisasi	39
4.3.4.	Lokalisasi.....	40
4.3.5.	Klasifikasi Mahalanobis	42
4.3.6.	Pemotongan Citra (kliping) Plat Nomor	43
4.4.	Pelatihan	46
4.5.	Implementasi Antar Muka	47
4.6.	Implementasi Uji Coba	48
4.7.	Analisa Hasil Pengujian	48
BAB V PENUTUP		55
5.1.	Kesimpulan.....	55
5.2.	Saran	55
DAFTAR PUSTAKA.....		57
LAMPIRAN		59

DAFTAR GAMBAR

Gambar 2.1. Ilustrasi Algoritma dalam pencarian titik yang sesuai.....	9
Gambar 2.2. Hasil perulangan pada citra kastil	10
Gambar 3.1. Citra yang memenuhi kriteria.....	17
Gambar 3.2. Ilustrasi pengambilan citra	18
Gambar 3.3. Alur Perangkat Lunak.	19
Gambar 3.4. Citra mobil	20
Gambar 3.5. Algoritma <i>Mean Shift</i> pada citra awal.....	20
Gambar 3.6. Diagram Alir Algoritma <i>Mean Shift</i>	21
Gambar 3.7. Citra yang telah disegmentasi	22
Gambar 3.8. Komposisi warna pada Citra Awal	22
Gambar 3.9. Matriks Citra Tersegmentasi baris 1	23
Gambar 3.10. Matriks Citra Tersegmentasi baris 8	23
Gambar 3.11. Citra yang telah dideteksi tepinya dan mengalami binerisasi	24
Gambar 3.12. Citra yang Dideteksi tepinya	25
Gambar 3.13. Algoritma binerisasi	25
Gambar 3.14. Tepian Objek dalam Citra yang Telah Mengalami Deteksi Tepi dan Lokalisasi.....	26
Gambar 3.15. Algoritma Region Growing	27
Gambar 3.16. Algoritma lokalisasi hasil	27
Gambar 3.17. Diagram alir lokalisasi citra.....	28
Gambar 3.18. Algoritma klasifikasi Mahalanobis.....	30
Gambar 3.19. Citra Plat nomor yang telah dikliping	30
Gambar 4.1 Struktur Basis Data.....	34
Gambar 4.2. Hasil Pemrosesan Citra	46
Gambar 4.3. Antarmuka program	47
Gambar 4.4. Antarmuka program (Data Pelatihan)	47
Gambar 4.5. Kandidat plat nomor L 1533 CA	50
Gambar 4.6. Kandidat plat nomor B 1472 BJA	51
Gambar 4.7. Grafik akurasi	51

DAFTAR TABEL

Tabel 3.1. Vektor uji klasifikasi	30
Tabel 3.2. Rancangan Tabel Pengujian Citra	31

Tabel 3.3. Tabel DataLatih.....	32
Tabel 3.4. Tabel DataUji	32
Tabel 3.5. Tabel DataKlasifikasi	32
Tabel 4.1. Data Hasil Pelatihan	47
Tabel 4.2. Pengujian pada citra mobil L 1533 CA	49

Daftar Source Code

Source Code 4.1. Ical Class	35
Source Code 4.2. Msa <i>Filtering</i>	36
Source Code 4.3. <i>Sobel Edge Detection</i>	38
Source Code 4.4. binarisasi	40
Source Code 4.5. Scan Image Hasil Binarisasi, dan Melakukan Penelusuran Warna Putih	41
Source Code 4.6. Deteksi Warna Putih yang Tidak Terputus dan Disimpan pada Variabel untuk Lokalisasi.....	41
Source Code 4.7. Klasifikasi Mahalanobis Distance.....	42
Source Code 4.8. Menghitung Titik Citra yang Dipotong.....	44
Source Code 4.9. Lokalisasi Karakter	45
Source Code 4.10. Mencatat data pelatihan	46

DAFTAR LAMPIRAN

Lampiran 1 data hasil penelitian	59
Lampiran 2 data klasifikasi plat nomor kelas 1	62
Lampiran 3 data uji non plat.....	62
Lampiran Citra yang diujikan dalam skripsi	63

BAB I PENDAHULUAN

1.1 Latar belakang

Sistem pengenalan nomor polisi kendaraan bermotor banyak dilakukan oleh negara maju untuk melacak pelanggaran lalu lintas, penggunaan sistem parkir otomatis dan lain sebagainya. Sistem ini cukup efektif dalam melaksanakan fungsinya sebagai alat bantu untuk mengefisienkan kinerja kepolisian mereka. Alat ini biasa diletakkan bersamaan dengan kamera khusus yang digunakan untuk mengambil citra kendaraan bermotor yang melanggar lalu lintas atau memasuki sistem parkir. Pada sistem tersebut, sebuah citra kendaraan bermotor akan dianalisa apakah plat nomor kendaraan tersebut dapat dianalisa atau tidak.

Dalam sistem pengenalan plat nomor, lokalisasi dan segmentasi karakter plat nomor merupakan hal yang harus dilakukan agar memperoleh kinerja yang sesuai. Hal tersebut dikarenakan dalam sebuah citra yang akan dianalisa terdapat bermacam-macam kandidat huruf atau angka yang akan dicari oleh sistem, sehingga tidaklah mungkin menganalisa semua objek tersebut untuk dikenali sebagai bagian dari karakter citra plat nomor.

Proses dari sistem tersebut dapat disederhanakan jika objek plat nomor yang akan dikenali telah ditentukan lokasi karakter plat nomornya. Hal ini dapat dimungkinkan jika dilakukan proses pengenalan plat nomor terlebih dahulu dengan mendeteksi benda-benda yang berbentuk persegi panjang dengan rasio tertentu. Kandidat plat nomor tersebut dianalisa kemudian dilakukan segmentasi karakternya.

Metode yang pernah dilakukan oleh Wenjing Jia, Huaifeng Zhang, dan Xiangjian He pada tahun 2005 dalam melakukan lokalisasi plat nomor, menggunakan metode *mean shift algorithm* untuk mengatasi masalah seperti kualitas gambar buruk karena kondisi pencahayaan sekitarnya, gambar perspektif distorsi, gangguan karakter, dan lain sebagainya dalam sistem pengenalan plat nomor yang ada waktu itu.

Kemudian melakukan deteksi tepi, serta pengenalan geometri citra plat nomor. Hasil penelitian mereka menunjukkan hal yang cukup baik dalam melakukan lokalisasi citra plat nomor.

Selain algoritma *mean shift*, pada citra yang telah disederhanakan warnanya, dilakukan deteksi tepi. Kegunaan deteksi tepi adalah melakukan *high pass filter* untuk menampilkan detail dari sebuah citra. Dari penelitian yang dilakukan tersebut, maka deteksi tepi sobel cukup bagus untuk diterapkan dalam sistem pengenalan plat nomor, karena dapat mengatasi masalah terlalu banyaknya gangguan citra.

Berdasarkan latar belakang yang telah dijabarkan, maka penelitian ini mengambil judul **“Penerapan Algoritma Means Shift dalam Lokalisasi Citra Plat Nomor Polisi dan Segmentasi Karakter”** untuk menguji keakuratan metode tersebut dalam melakukan segmentasi karakter pada citra plat nomor.

1.2 Perumusan Masalah

Berdasar latar belakang masalah, masalah yang akan dirumuskan adalah mengenai pemisahan gambar plat nomor polisi pada citra yang berisi kendaraan bermotor dan objek lain di sekitarnya menggunakan metode algoritma pergeseran nilai rata-rata (*mean shift algorithm*) dan deteksi tepi sobel (*sobel edge detection*).

Secara lebih lengkap, masalah dapat didefinisikan sebagai berikut :

1. Bagaimana memisahkan citra plat nomor pada citra yang berisi kendaraan bermotor, lengkap dengan plat nomor kendaraan tersebut.
2. Bagaimana penggunaan metode *mean shift algorithm* dan deteksi tepi sobel (*sobel edge detection*) sebagai solusi pemisahan gambar plat nomor tersebut dari citra.
3. Bagaimana keakuratan penggunaan kedua metode tersebut dalam melakukan lokalisasi dan segmentasi karakter plat nomor.

1.3 Batasan Masalah

Penyelesaian dari masalah yang telah didefinisikan membutuhkan batasan tertentu agar dapat masalah yang ada dapat diselesaikan dengan efektif sehingga mencapai tujuan yang diinginkan. Batas masalah yang diberikan adalah :

1. Citra plat nomor kendaraan, adalah mobil pribadi yang beroperasi di Indonesia.
2. Citra plat nomor diambil beberapa contoh mobil, dengan jarak dan ukuran yang konstan sehingga citra plat nomor memiliki dimensi 2% dari resolusi citra.
3. Citra yang diambil jelas dan tidak lebih dari satu citra plat nomor kendaraan dalam satu citra.
4. Algoritma yang digunakan adalah algoritma pergeseran nilai rata-rata (*mean shift algorithm*) dan deteksi tepi sobel (*sobel edge detection*) sebagai pemisah objek benda. Algoritma lokalisasi citra yang telah tersegmentasi dan algoritma deteksi area plat nomor.
5. Citra yang digunakan memiliki format *windows jpeg* dan resolusi 400x300 pixel.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai adalah

1. Pengembangan perangkat lunak untuk melakukan lokalisasi dan segmentasi karakter pada citra berisi plat nomor kendaraan bermotor yang berasal dari citra yang berisi objek lainnya.
2. Uji keakuratan dari algoritma pergeseran nilai rata-rata dan deteksi tepi sobel (*sobel edge detection*) untuk lokalisasi dan segmentasi karakter citra plat nomor mobil yang beroperasi di Indonesia.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini antara lain :

1. Tersedianya perangkat lunak untuk melakukan lokalisasi dan segmentasi karakter citra plat nomor kendaraan bermotor pada citra kompleks yang berisi objek lainnya

dengan metode algoritma pergeseran nilai rata-rata dan deteksi tepi sobel (*sobel edge detection*).

2. Tersedianya referensi sebagai acuan pengembangan lokalisasi dan segmentasi karakter citra plat nomor dengan metode pergeseran nilai rata-rata dan deteksi tepi sobel (*sobel edge detection*).
3. Tersedianya informasi tentang keakuratan metode pergeseran nilai rata-rata dan deteksi tepi sobel (*sobel edge detection*) dalam segmentasi citra digital.
4. Tersedianya perangkat lunak yang dapat dikembangkan menjadi pendeteksi nomor polisi kendaraan bermotor yang dapat dimanfaatkan pada sistem parkir, sistem ketertiban lalu-lintas dan lain sebagainya di Indonesia.

1.6 Metodologi Penelitian

Metodologi yang digunakan :

1. Studi Literatur

Mempelajari teori-teori yang berhubungan dengan konsep citra digital, metode segmentasi dengan menggunakan teori *mean shift algorithm* serta teori mengenali bentuk fisiologis dari citra digital.

2. Pendefinisian dan analisis masalah

Mendefinisikan dan menganalisis masalah untuk mencari solusi yang tepat.

3. Perancangan dan implementasi sistem

Membuat rancangan model perangkat lunak dengan analisis terstruktur dan mengimplementasikan hasil rancangan tersebut yaitu membuat piranti lunak segmentasi dan lokalisasi citra plat nomor kendaraan bermotor.

4. Uji coba dan analisa hasil implementasi

Menguji perangkat lunak dengan berbagai citra kendaraan bermotor dengan batasan masalah yang telah disebutkan. Hasil dari implementasi tersebut dianalisa kesesuaiannya dengan tujuan yang telah dirumuskan sebelumnya, untuk kemudian dievaluasi dan disempurnakan.

1.7 Sistematika Penulisan

1. BAB I PENDAHULUAN

Berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Menguraikan teori-teori yang berhubungan plat nomor polisi, teori yang berkaitan dengan algoritma segmentasi dan lokalisasi citra digital.

3. BAB III METODE DAN PERANCANGAN SISTEM

Pada bab ini akan dijelaskan mengenai metode-metode yang digunakan dalam segmentasi dan lokalisasi citra plat nomor kendaraan bermotor.

4. BAB IV HASIL DAN PEMBAHASAN

Pada bab ini akan dilakukan implementasi sistem, pengujian dan analisa model sistem perangkat lunak yang dibangun.

5. BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan dari seluruh rangkaian penelitian serta saran kemungkinan pengembangan selanjutnya.

UNIVERSITAS BRAWIJAYA



BAB II

TINJAUAN PUSTAKA

2.1. Tanda Nomor Kendaraan Bermotor (Plat)

Di Indonesia, bukti bahwa kendaraan bermotor telah didaftarkan, diberikan buku pemilik kendaraan bermotor, surat tanda nomor kendaraan bermotor serta tanda nomor kendaraan bermotor (plat).

Bentuk, ukuran, bahan, warna, dan cara pemasangan tanda nomor kendaraan bermotor (plat) sebagaimana dimaksud harus memenuhi syarat-syarat sebagai berikut :

a.berbentuk lempengan tipis persegiempat, dengan ukuran panjang 250 milimeter dan lebar 105 milimeter untuk sepeda motor dan ukuran panjang 395 milimeter serta lebar 135 milimeter untuk kendaraan jenis lainnya serta ditambahkan tempat untuk pemasangan tanda uji

b.terbuat dari bahan yang cukup kuat serta tahan terhadap cuaca, yang pada permukaannya berisi huruf dan angka yang dibuat dari bahan yang dapat memantulkan cahaya

c.tinggi huruf dan angka pada tanda nomor kendaraan bermotor yang dituliskan pada lempengan sebagaimana dimaksud dalam huruf a, sekurang-kurangnya 45 milimeter untuk sepeda motor dan 70 milimeter untuk kendaraan bermotor jenis lainnya

d.warna tanda nomor kendaraan bermotor adalah sebagai berikut :

1)dasar hitam, tulisan putih untuk kendaraan bermotor bukan umum dankendaraan bermotor sewa

2)dasar kuning, tulisan hitam untuk kendaraan umum

3)dasar merah, tulisan putih untuk kendaraan bermotor dinas Pemerintah

4)dasar putih, tulisan hitam untuk kendaraan bermotor Korps Diplomatik negara asing.

e.Tanda nomor kendaraan bermotor dipasang pada tempat yang disediakan bagian depan dan belakang kendaraan bermotor. (Soeharto, 1993)

2.2. Sistem Pengenalan Plat Nomor Polisi Otomatis

Pengenalan plat nomor otomatis (*automatic license plate recognition*) adalah sebuah metode pengawasan yang menggunakan pengenalan karakter optik (*optical character recognition*) pada gambar untuk membaca plat nomor pada kendaraan. Sistem ini dapat digunakan dengan cara pemasangan kamera pada jalan yang rawan terjadi pelanggaran peraturan atau digunakan oleh pihak kepolisian sebagai metode pengumpulan pembayaran tol elektronik per penggunaan lalu lintas jalan dan monitoring kegiatan, seperti kepatuhan lampu merah di persimpangan.

Sistem ini dapat digunakan untuk menyimpan gambar yang diambil oleh kamera serta teks dari plat nomor, dengan beberapa konfigurasi, foto pengemudi dapat disimpan. Sistem pencahayaan biasanya menggunakan inframerah untuk memungkinkan kamera untuk mengambil gambar kapan saja hari. Kilat yang kuat termasuk dalam setidaknya satu versi dari persimpangan-pemantauan kamera, melayani baik untuk menerangi gambar dan membuat pelaku menyadari kesalahannya (Constant, 1999).

Pergeseran nilai rata-rata (*mean*) atau lebih dikenal dengan istilah *mean shift* adalah analisis nonparametrik umum teknik clustering berdasarkan kepadatan estimasi untuk analisis spasial fitur kompleks. Algoritma ini terdiri dari prosedur iteratif sederhana yang menggeser masing-masing fitur menunjuk ke titik stasioner terdekat sepanjang gradien arah dari perkiraan fungsi kepadatan. Ini telah berhasil diterapkan pada banyak aplikasi seperti segmentasi dan pelacakan.

2.3. Algoritma Pergeseran Nilai Rataan (*Mean Shift Algorithm*)

Prosedur pergeseran rata-rata terdiri dari dua langkah: perkiraan gradien dari fungsi kepadatan, dan pemanfaatan hasilnya untuk membentuk kelompok. Gradien dari fungsi kepadatan diperkirakan oleh densitas estimator nonparametrik.

Kemudian mulai dari masing-masing titik sampel, prosedur *mean-shift* secara berulang menemukan jalan sepanjang gradien arah menjauh dari lembah dan menuju puncak terdekat (Yang, Duraiswami, DeMenthon, Davis, 2003).

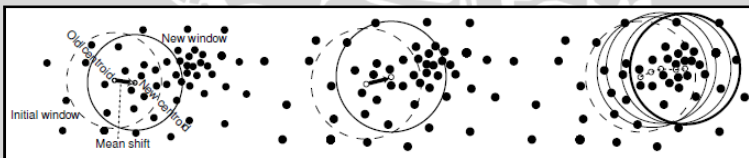
Secara matematis, algoritma *mean shift* dinotasikan pada persamaan 2.1.

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right), \dots (2.1)$$

di mana h (disebut parameter bandwidth) mendefinisikan radius kernel. Sedangkan n adalah jumlah titik-titik yang digunakan sebagai jumlah pixel anggota kernel, dan K adalah kernel yang diperoleh dari persamaan 2.2.

$$K(x) = C_{k,d} k(|x|^2) \dots (2.2)$$

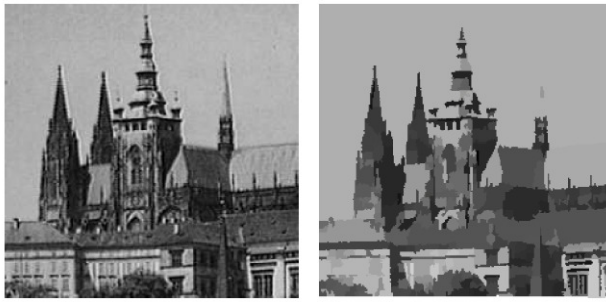
dan gambar 2.1. memberikan ilustrasi pergeseran titik-titik yang dapat menyebabkan terjadinya perubahan nilai rataan sehingga menghasilkan nilai warna yang sesuai dengan nilai warna-warna yang lain.



Gambar 2.1. Ilustrasi Algoritma dalam pencarian titik yang sesuai

Algoritma tersebut dijalankan dengan urutan sebagai berikut :

1. Dimulai dengan memilih wilayah untuk memulai algoritma
2. Menentukan pusat massa dari data.
3. Pindahkan daerah ke lokasi pusat massa baru.
4. Terus diulang hingga konvergen sehingga menghasilkan citra pada gambar 2.2.



Gambar 2.2. Hasil algoritma pada citra kastil

Menurut analisis statistik, dibandingkan dengan daerah non pelat nomor, daerah pelat kombinasi fitur yang bersifat unik, seperti bentuk yang selalu persegi panjang, ukuran persegi panjang tersebut memiliki aspek rasio (panjang : lebar) tertentu. Fitur tersebut didefinisikan dan diekstraksi dalam rangka untuk mendeteksi daerah plat dari daerah yang telah tersegmentasi.

2.4. Kepersegipanjang (*Rectangularity*)

Plat nomor polisi selalu berbentuk persegi panjang dengan aspek rasio tertentu. Ini adalah bentuk yang paling penting dari ciri plat nomor. Aspek persegi panjang (*rectangularity*) adalah suatu pengukuran yang menunjukkan bagaimana objek tersebut dibandingkan dengan aspek rasio minimum (*minimum enclosing ratio*) yang telah didefinisikan dalam persamaan 2.3.

$$R = \frac{A_0}{A_{MER}} \dots (2.3)$$

di mana A_0 adalah luas objek dan A_{MER} adalah luas pendekatan ukuran rasio objek plat nomor tersebut. Dalam kasus tertentu, citra plat nomor yang dianalisa mempunyai kemiringan tertentu, sehingga mengharuskan adanya rotasi

untuk mendapat rasio aspek kepersegipanjang yang tepat (Wenjing Jia, Huaifeng Ze, Xiangjian He, 2005).

2.4.1. Rasio Persegi Panjang Plat Nomor

Tanda nomor kendaraan bermotor sebagaimana dimaksud dalam Pasal 175 PP No.44/1993 berisi data mengenai kode wilayah pendaftaran, nomor pendaftaran kendaraan bermotor dan masa berlaku. Tanda Nomor Kendaraan Bermotor berbentuk plat aluminium dengan cetakan tulisan dua baris yakni dengan ketentuan:

- a. Baris pertama menunjukkan: kode wilayah (huruf), nomor polisi (angka), dan kode/seri akhir wilayah (huruf).
- b. Baris kedua menunjukkan bulan dan tahun masa berlaku
- c. Ukuran panjang 250 milimeter dan lebar 105 milimeter untuk sepeda motor dan ukuran panjang 395 milimeter serta lebar 135 milimeter untuk kendaraan jenis lainnya. (Soeharto, 1993). Berdasarkan ukuran plat nomor yang disebutkan, terdapat kepastian rasio ukuran plat nomor polisi kendaraan bermotor roda dua yaitu 5 : 2 dan 4 : 1 untuk roda empat atau lebih.

2.5. Deteksi Tepi

Deteksi tepi (*Edge Detection*) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra, tujuannya adalah :

- a. Untuk menandai bagian yang menjadi detail citra
- b. Untuk memperbaiki detail dari citra yang kabur, yang terjadi karena error atau adanya efek dari proses akuisisi citra

Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya (Riyanto).

Terdapat beberapa teknik yang digunakan untuk mendeteksi tepi, antara lain :

- a. Operator gradien pertama, contoh beberapa gradien pertama yang dapat digunakan untuk mendeteksi tepi di dalam citra, yaitu operator gradien selisih-terpusat,

operator Sobel, operator Prewitt, operator Roberts, operator Canny.

- b. Operator turunan kedua, disebut juga operator Laplace. Operator Laplace mendeteksi lokasi tepi khususnya pada citra tepi yang curam. Pada tepi yang curam, turunan keduanya mempunyai persilangan nol, yaitu titik di mana terdapat pergantian tanda nilai turunan kedua, sedangkan pada tepi yang landai tidak terdapat persilangan nol. Contohnya adalah operator Laplacian Gaussian, operator Gaussian.
- c. Operator kompas, digunakan untuk mendeteksi semua tepi dari berbagai arah di dalam citra. Operator kompas yang dipakai untuk deteksi tepi menampilkan tepi dari 8 macam arah mata angin yaitu Utara, Timur Laut, Timur, Tenggara, Selatan, Barat, Barat Daya, dan Barat Laut. Deteksi tepi dilakukan dengan mengkonvolusikan citra dengan berbagai mask kompas, lalu dicari nilai kekuatan tepi (*magnitude*) yang terbesar dan arahnya. Operator kompas yang dipakai untuk deteksi tepi menampilkan tepi dari 8 macam arah mata angin, yaitu Utara, Timur Laut, Timur, Tenggara, Selatan, Barat, Barat Daya, dan Barat Laut.

Selain operator gradien yang sudah disebutkan, masih ada beberapa operator gradien yang lain yang dapat digunakan untuk mendeteksi tepi di dalam citra, yaitu selisih terpusat, sobel, prewitt, Roberts, dan Canny.

2.5.1. Deteksi Tepi dengan Metode Sobel

Tinjau pengaturan pixel di sekitar pixel (x,y) :

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_3 & (x, y) & a_4 \\ a_5 & a_6 & a_7 \end{bmatrix}$$

Operator Sobel adalah magnitudo dari gradien yang dihitung dengan persamaan 2.4.

$$M = \sqrt{S_x^2 + S_y^2} \dots (2.4)$$

yang dalam hal ini, turunan parsial dihitung dengan

$$\begin{aligned} Sx &= (a_2 + ca_2 + a_4) - (a_0 + ca_7 + a_6) \\ Sy &= (a_0 + ca_1 + a_{22}) - (a_6 + ca_5 + a_4) \end{aligned}$$

Persamaan 2.5. Perhitungan Nilai Magnitudo

dengan konstanta $c = 2$. Dalam bentuk mask, s_x dan s_y dapat dinyatakan sebagai

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ dan } S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

(Dewi Agushinta R, Alina Diyanti)

2.5.2. Kepadatan tepi (*edge density*)

Kepadatan tepi adalah nilai dari penjumlahan setiap pixel dari tepian objek dalam citra yang telah dideteksi tepinya. Deteksi tepi ini merupakan salah satu variabel yang digunakan dalam pengenalan objek.

2.6. Jarak Mahalanobis

Jarak Mahalanobis merupakan ukuran jarak yang diperkenalkan oleh Prasanta Chandra Mahalanobis pada tahun 1936, seorang ilmuwan India (Mahalanobis, 1932).

Jarak Mahalanobis didasarkan pada korelasi antar variabel-variabel, khususnya invers matriks kovariansi. Kuadrat jarak Mahalanobis antara dua vektor acak x dan y , yang berdistribusi sama dengan matriks kovariansi Σ , adalah

$$d_m(x, y) = \sqrt{(x - y)^t \Sigma^{-1} (x - y)} \dots (2.6)$$

Jarak Mahalanobis banyak digunakan dalam analisa kluster (*cluster analys*). Misalkan kita memiliki dua kelompok yang berbeda (populasi) yang kita beri label sebagai G1 dan G2. Sebagai contoh, di beberapa komunitas, G1 dan G2 mungkin mewakili anak perempuan dan anak laki-laki,

masing-masing atau, dalam diagnosa misalnya orang normal dan berpenyakit. Misal X menandakan (Random) vektor yang berisi pengukuran yang dilakukan pada diberikan orang pribadi atau badan yang diteliti. Asumsi umum adalah untuk mengambil vektor acak p -dimensi X sebagai memiliki variasi yang sama tentang artinya dalam salah kelompok. Kemudian perbedaan antara kelompok-kelompok dapat dianggap dalam hal perbedaan antara vektor mean dari X , di setiap kelompok relatif terhadap variasi dalam kelompok umum. (McLachlan, 1999).

2.7. Pengujian Statistik

Ada dua metode pengujian dalam statistik yaitu pengujian parametrik dan non parametrik.

2.7.1. Pengujian Parametrik ialah suatu uji yang menghendaki syarat - syarat spesifik mengenai parameter dan populasi dari mana cuplikan diambil. semua syarat ini tidak akan diuji, hanya diduga saja bahwa syarat tersebut dapat dipenuhi. hasil dari uji parametrik dapat diterima berdasarkan suatu harga penerimaan dari asumsi yang diajukan. selain itu, dalam uji parametrik membutuhkan suatu tingkat pengukuran yang teliti paling tidak dalam skala interval.

2.7.2. Uji statistika nonparametrik ialah uji dengan model yang tidak membutuhkan suatu parameter khusus dari populasi yang diamati. beberapa asumsi yang berhubungan erat dengan uji statistika nonparametrik adalah bahwa bahwa pengamatan tersebut bebas dan variabel yang diamati kontinu, tetapi asumsi yang dibuat adalah lebih lemah dan kurang teliti jika dibandingkan dengan uji parametrik. oleh karena itu, uji nonparametrik tidak membutuhkan suatu pengukuran dengan tingkat ketelitian yang tinggi seperti uji parametrik. biasanya uji non parametrik dipakai untuk menganalisis data dalam skala ordinal dan nominal. (siegel, 1994)

Skala nominal yang membedakan benda atau peristiwa yang satu dan yang lainnya berdasarkan nama (predikat). jadi, boleh mengklasifikasikan barang-barang yang dihasilkan pada suatu proses dengan predikat cacat atau tidak cacat. tidak jarang digunakan angka2 yang dipilih sembarang, sebagai pengganti nama-nama atau sebutan 2 untuk membedakan benda atau karakteristik tertentu. sebagai contoh, dapat digunakan pada angka 1 untuk menyebut barang yang cacat, dan angka 0 untuk barang tidak cacat. skala nominal juga digunakan terhadap sejumlah benda atau peristiwa yang termasuk dalam kategori nominal. (Daniel, 1989)

2.8. Metode Evaluasi.

Untuk mengevaluasi performansi model yang dibangun oleh algoritma klasifikasi dapat dilakukan dengan menghitung jumlah dari test record yang diprediksi benar (akurasi) atau salah (error rate) oleh model tersebut. akurasi dan error rate didefinisikan sebagai berikut (tan., dkk, 2004)

$$\text{Akurasi} = \frac{\text{jumlah prediksi benar}}{\text{jumlah total prediksi}} \times 100\%.$$

$$\text{Error rate} = \frac{\text{jumlah prediksi salah}}{\text{jumlah total prediksi}} \times 100\%.$$

algoritma klasifikasi berusaha untuk mencari model yang mempunyai akurasi yang tinggi atau error rate yang rendah ketika model diterapkan pada test set.

UNIVERSITAS BRAWIJAYA



BAB III

METODE DAN PERANCANGAN SISTEM

Pada bab ini akan dijelaskan secara detail mengenai proses perancangan sistem serta metode yang akan digunakan dalam menyelesaikan masalah yang telah dirumuskan.

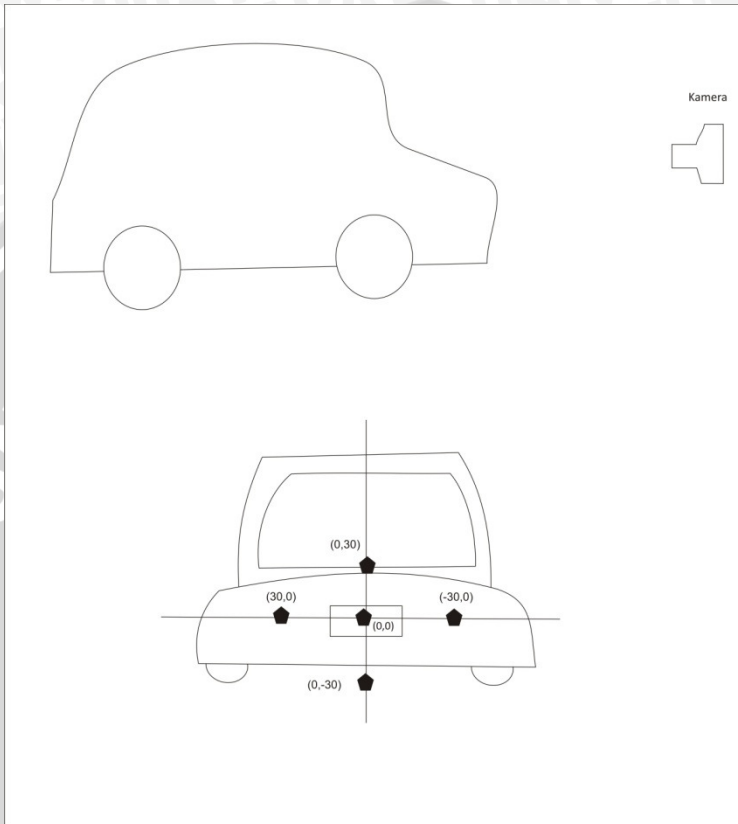
3.1. Pengumpulan serta pemilihan citra yang akan digunakan.

Agar diperoleh hasil dari citra yang sesuai, perlu diperhatikan beberapa hal, diantaranya adalah jarak serta fokus kamera yang akan mengambil gambar tersebut.



Gambar 3.1. citra yang memenuhi kriteria

Dimensi citra konstan sebesar 400x300 pixel dan dimensi plat nomor sebesar ± 2 persen dari ukuran citra. Jumlah sampel citra yang akan mengalami pemrosesan diambil 5 citra tiap 20 mobil dengan sudut berbeda, $(0^\circ, 0^\circ)$, $(0^\circ, 30^\circ)$, $(0^\circ, -30^\circ)$, $(30^\circ, 0^\circ)$, $(-30^\circ, 0^\circ)$. Di mana posisi kamera terhadap mobil tetap. Sementara sudut pengambilan berubah-ubah. Ilustrasi sudut pengambilan citra ditampilkan pada gambar 3.2.



Gambar 3.2. Ilustrasi pengambilan citra

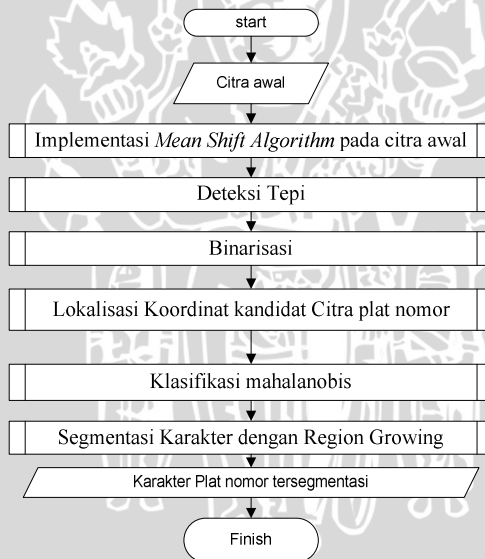
Agar diperoleh data yang valid mengenai keakuratan metode pergeseran nilai rata-rata (*mean shift algorithm*) dan deteksi tepi sobel (*sobel edge detection*) dalam melakukan lokalisasi karakter citra plat nomor dari citra, dilakukan uji keakuratan dengan mengelompokkan pada masing-masing sudut pengambilan citra.

3.2. Analisa dan perancangan perangkat lunak.

3.2.1. Proses pengolahan citra

Tahap pengolahan citra dijelaskan pada gambar 3.3. Citra awal disegmentasikan dengan menggunakan *mean shift*

algorithm, sehingga memperoleh hasil citra dengan warna yang lebih sederhana, hasil dari segmentasi tersebut dapat diamati histogram warnanya. Dari citra yang telah tersegmentasi, maka akan didapat beberapa objek, dengan warna yang telah homogen. Dari citra tersebut, warna-warna homogen yang tersebar dalam citra akan dideteksi tepi. Hasil akhir deteksi tepi akan dilokalisasi dengan mengabaikan noise (citra homogen dengan luas $< \pm 175$ pixel). Citra yang telah dilokalisasi akan diuji bentuk geometrisnya, dengan klasifikasi *mahalanobis* sehingga dapat disimpulkan bagian mana dari citra yang telah tersegmentasi tersebut merupakan citra plat nomor. Kemudian pada area plat nomor yang ditemukan, dilakukan binerisasi dan sekali lagi dilakukan *region growing* untuk menentukan bagian mana dari plat nomor tersebut yang merupakan karakter.



Gambar 3.3. Alur perangkat lunak

3.2.1.1. Segmentasi Citra dengan Algoritma Pergeseran Rataan (*Means Shift Algorithm*).

Tahap segmentasi citra dimulai dengan memproses citra dengan algoritma pada gambar 3.5. serta gambar 3.6. yaitu

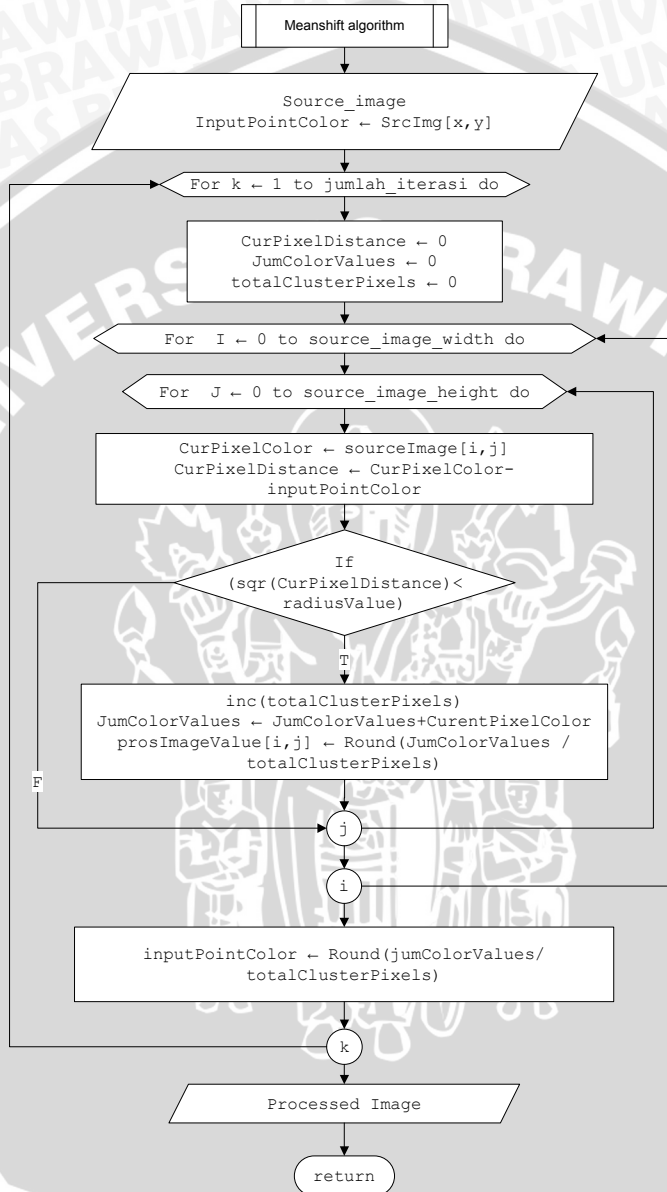
diagram alir algoritma *mean shift*, dari algoritma tersebut diperoleh citra dengan warna sederhana dari citra yang sebelumnya berwarna kompleks (gambar 3.4 , 3.7).



Gambar 3.4. Citra mobil

1. InputWarna $\leftarrow$ Ambil titik pada citra
2. Tentukan jumlah perulangan algoritma
3. Definisikan jumlahNilaiWarna dengan nilai awal 0
4. Definisikan jumlahPixelCluster dengan nilai awal 0
5. Scan titik pada citra seluruhnya
6. JarakPiksel $\leftarrow$ nilaiWarnaCitraSekarang - NilaiAwalWarna
7. Jika jarakPiksel < Radius maka
 - 8.1. inc(jumlahPixelCluster)
 - 8.2. JumColorValues $\leftarrow$ JumColorValues + CurPixelColor
 - 8.3. NilaiWarnaCitraSekarang $\leftarrow$ jumColorValues / totalClusterPixels
 - 8.4. inputWarna $\leftarrow$ NilaiWarnaCitraSekarang

Gambar 3.5. Algoritma *mean shift* pada citra awal



Gambar 3.6. Diagram Alir Algoritma *Mean Shift*



Gambar 3.7. Citra yang telah disegmentasi

Pelaksanaan algoritma ini adalah dengan memberikan 9 titik dengan membagi citra menjadi 9 bagian dan mengambil titik tengahnya sebagai titik mula atau input koordinat awal citra, sehingga dari citra tersebut diambil 9 sampel warna dari 9 titik tersebut. Dari setiap titik tersebut akan diolah dengan algoritma yang dijelaskan dalam Algoritma 3.1 sehingga memperoleh hasil seperti gambar 3.7.

Contoh perhitungan pada citra 8x8 berikut, kernel bergerak pada indeks matriks 0,0. Citra awal, memiliki variasi warna sebanyak 20, perhitungan akan ditunjukkan pada 3 pixel di baris pertama, dimulai dengan menampilkan komposisi warna pada citra awal yang ditunjukkan pada gambar 3.8 dan kernel bergerak dari titik (0,0), serta nilai radius sebesar 10.

13	16	8	20	2	7	8	13
1	14	18	13	13	18	16	14
18	7	10	17	11	8	15	14
19	20	17	2	17	11	9	7
10	5	13	7	2	16	18	18
18	8	20	12	5	8	15	1
10	13	3	12	8	1	15	8
10	20	3	3	14	5	19	19

Gambar 3.8. Komposisi warna pada Citra Awal

13	14	12	14	2	13	12	12
1	14	18	13	13	18	16	14
18	7	10	17	11	8	15	14
19	20	17	2	17	11	9	7
10	5	13	7	2	16	18	18
18	8	20	12	5	8	15	1
10	13	3	12	8	1	15	8
10	20	3	3	14	5	19	19

Gambar 3.9. Matriks Citra Tersegmentasi baris 1

Segmentasi dilakukan pada baris pertama, menghasilkan perhitungan 7 pixel yang mendekati titik pertama (0,0) dengan nilai rata-rata 12 dengan perhitungan sebagai berikut :

Pada pixel (0,1), nilai 16 menjadi 14, diperoleh dari $(13+16)/2 \approx 14$. Pada pixel (0,2), nilai 8 menjadi 12, diperoleh dari $(13+16+8)/3 \approx 12$. Pada pixel (0,3) dari bernilai 20 menjadi 14 diperoleh dari $(13+16+8+20)/4 \approx 14$. Semua nilai yang masih masuk dalam radius, memperoleh perlakuan yang sama, sedangkan nilai yang di luar radius, tidak diperdulikan. Dan proses ini berlanjut dengan pengulangan yang diperlukan sampai pada citra menjadi citra sebagai dengan komposisi warna yang ditunjukkan pada gambar 3.9.

14	14	13	14	13	13	13	13
13	14	14	2	14	14	14	13
13	13	13	13	2	13	13	13
13	13	13	13	13	13	13	1
13	13	3	13	13	1	13	13
13	13	3	3	13	13	13	13

Gambar 3.10. Matriks Citra Tersegmentasi baris 8

Segmentasi dilakukan pada baris ke delapan, menghasilkan perhitungan 55 pixel yang mendekati nilai titik

pertama (0,0) dengan nilai rata-rata 13. Nilai yang mendekati nilai rata-rata akan mengalami perubahan nilai sesuai pergeseran rata-rata, sehingga, jika perulangan ini dilakukan berulang, ulang baik pada titik yang belum disegmentasi, ataupun yang telah mengalami segmentasi, akan menghasilkan citra dengan warna yang lebih sederhana.

3.2.1.2. Deteksi Tepi dengan Metode Sobel.

Proses yang dialami citra untuk dapat diambil bagian yang diinginkan adalah dengan melakukan deteksi tepi sobel. Metode ini dilakukan dengan cara membuat *subimage* dengan ukuran tertentu dikalikan dengan konstanta sobel untuk memperoleh hasil seperti yang ditunjukkan dalam gambar 3.11 menjadikan citra yang telah mengalami deteksi tepi dapat diproses bentuk geometrisnya.



Gambar 3.11. Citra yang telah dideteksi tepinya dan mengalami binerisasi

Contoh berikut memperlihatkan deteksi tepi dengan operator Sobel. Konvolusi pertama dilakukan terhadap pixel yang bernilai 2 (di titik pusat *mask*) yang diperlakukan pada contoh gambar 3.10 :

*	*	*	*	*	*	*	*
*	22	3	16	22	17	3	*
*	20	13	20	13	3	0	*
*	3	24	17	33	16	2	*

*	3	14	30	14	22	20	*
*	14	20	1	16	17	17	*
*	32	32	42	17	0	17	*
*	*	*	*	*	*	*	*

Gambar 3.12. Citra yang Dideteksi tepinya

Nilai 22 pada gambar 3.12 citra hasil konvolusi diperoleh dengan perhitungan berikut :

$$S_x = (13)(1) + (1)(2) + (14)(1) + (12)(-1) + (13)(-2) + (13)(-1) = -22$$

$$S_y = (13)(1) + (14)(2) + (12)(1) + (14)(-1) + (14)(-2) + (13)(-1) = -2$$

$$M = \sqrt{22^2 + (-2)^2} \cong 22$$

3.2.1.3. Binerisasi citra

Citra yang telah mengalami deteksi tepi, masih mempunyai permasalahan yang menjadikan lokalisasi citra sukar dilakukan. Pada citra, masih banyak tersebar pixel-pixel keabuan akibat proses deteksi tepi. Hal ini dapat dihilangkan dengan melakukan binerisasi (*threshold limitation*) dengan algoritma pada gambar 3.13, yakni memberikan nilai batas tertentu untuk membulatkan bilangan penyusun warna. Sehingga warna yang dapat dimunculkan dari citra hasil image segmentation berupa citra hitam – putih murni. Proses ini merupakan proses yang sangat penting, karena dapat menghilangkan titik-titik piksel yang mengganggu proses lokalisasi citra.

1. Scan seluruh pixel gambar
2. Simpan nilai warna suatu titik dalam variabel
3. Pisahkan nilai R,G,B variabel tersebut
4. Jika $R \leq TressHoldvalue$
5. Jika benar maka $R \leftarrow 0$
6. Jika Salah maka $R \leftarrow 255$
7. Jika $G \leq TressHoldvalue$
8. Jika benar maka $G \leftarrow 0$
9. Jika Salah maka $G \leftarrow 255$
10. Jika $B \leq TressHoldvalue$
11. Jika benar maka $B \leftarrow 0$
12. Jika Salah maka $B \leftarrow 255$
13. Nilai akhir citra $\leftarrow RGB(R,G,B)$

Gambar 3.13. Algoritma binerisasi

3.2.1.4. Lokalisasi Citra

Citra yang telah mengalami deteksi tepi dan perbaikan kontras selanjutnya dipindahkan, dengan menelusuri pixel-pixel yang berwarna putih. Setiap pixel yang tidak terputus akan dicatat dalam sebuah variabel yang menyimpan jumlah luasan serta koordinat tiap titik yang diuji. Selain itu, diberikan nilai tertentu yang akan menyatakan luas minimal citra yang tidak dianggap sebagai pixel pengganggu (*noise*).

Proses lokalisasi gambar dilakukan dengan metode *region growing* atau deteksi tetangga dengan warna yang sama seperti yang telah didefinisikan pada gambar algoritma 3.18. Citra yang telah terbinerisasi tentunya hanya memiliki dua macam warna, yaitu hitam dan putih. Proses *region growing* ini dimulai dengan mencari titik putih sebagai objek yang akan dicari. Selanjutnya selama masih menemukan tetangga yang berwarna putih akan dianggap sebagai tetangga dan tiap koordinat tetangga tersebut disimpan dalam array koordinat hingga tidak menemukan tetangga lagi, setiap titik yang terdeteksi tentunya memiliki luasan tertentu, sehingga luasan yang lebih kecil dari batasan *noise* akan dianggap sebagai *noise* dan diabaikan pada gambar 3.14. terdapat 13 objek yang dianggap bukan *noise*, sehingga dapat dilokalisasi pada citra yang lain menggunakan algoritma pada gambar 3.15, 3.16. 3.17,



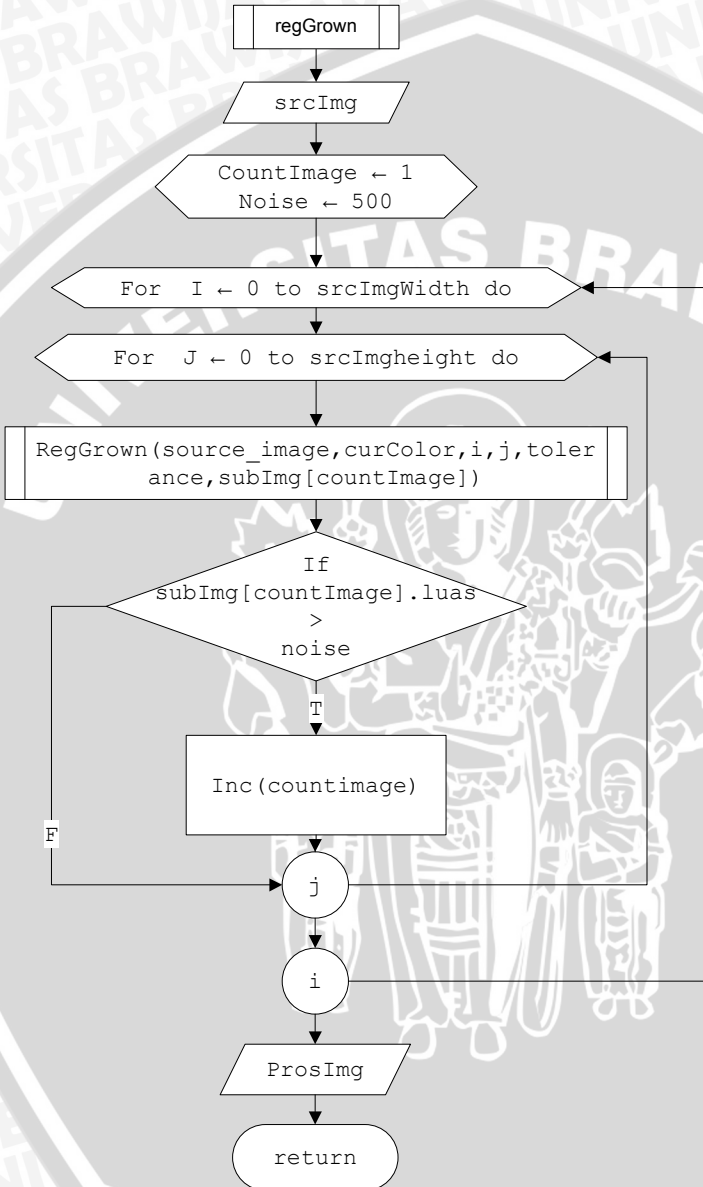
Gambar 3.14. Tepian Objek dalam Citra yang Telah Mengalami Deteksi Tepi dan Lokalisasi

1. Pada Gambar awal, pilih titik, simpan sebagai koordinatcitra
2. Jika koordinatcitra[x+1,y+1] memenuhi nilai toleransi, maka luas pixel bertambah 1, simpan lokasi koordinatnya
3. Jika koordinatcitra[x+1,y] memenuhi nilai toleransi, maka luas pixel bertambah 1, simpan lokasi koordinatnya
4. Jika koordinatcitra[x+1,y-1] memenuhi nilai toleransi, maka luas pixel bertambah 1, simpan lokasi koordinatnya
5. Jika koordinatcitra[x-1,y+1] memenuhi nilai toleransi, maka luas pixel bertambah 1, simpan lokasi koordinatnya
6. Jika koordinatcitra[x-1,y+1] memenuhi nilai toleransi, maka luas pixel bertambah 1, simpan lokasi koordinatnya
7. Jika koordinatcitra[x-1,y+1] memenuhi nilai toleransi, maka luas pixel bertambah 1, simpan lokasi koordinatnya
8. Jika koordinatcitra[x,y+1] memenuhi nilai toleransi, maka luas pixel bertambah 1, simpan lokasi koordinatnya
9. Jika koordinatcitra[x,y-1] memenuhi nilai toleransi, maka luas pixel bertambah 1, simpan lokasi koordinatnya

Gambar 3.15 Algoritma *Region Growing*

1. Inisiasi JumGbr $\leftarrow 0$
2. Scan seluruh titik pada citra
3. Jika suatu titik tidak berwarna hitam
4. Lakukan *Region Growing* dan simpan semua titik
5. Jika Luas Pixel yang dideteksi lebih > dari batasNoise
6. Jika benar, maka simpan semua kordinat
7. Jika tidak, buang nilai yang telah disimpan
8. Lanjutkan pencarian berikutnya

Gambar 3.16. Algoritma lokalisasi hasil



Gambar 3.17. Diagram alir lokalisasi citra

3.2.1.5. Klasifikasi Mahalanobis

Setiap objek yang telah dilokalisasi, diekstrak fiturnya. Dalam penelitian ini, ada tiga variabel yang akan diujikan dalam klasifikasi mahalanobis menjadi vektor x . Yaitu rasio, kepersegi panjang, dan kepadatan tepi. Ketiga variabel dibentuk menjadi sebuah vektor, yang diuji dengan metode jarak mahalanobis.

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} aspek_rasio(Ar) \\ kepersegi_panjang(R) \\ kepadatan_tepi(Ed) \end{bmatrix}$$

$$d_k = \sqrt{(x - mk)^t C^{-1} (x - mk)}$$

dengan

$$k = 1, 2, 3$$

Vector M dan C diperoleh dari data latih yang diambil dari penelitian. Matriks M merupakan nilai rata-rata dari fitur 10 citra plat nomor dalam data latih, sedangkan vektor C merupakan nilai varians dari nilai fitur 10 data latih.

Dalam penelitian, terdapat dua fitur yang menandakan objek tersebut sebagai sebuah citra plat nomor, oleh karena itu, data pengujian fitur memiliki tiga buah vektor yang akan diujikan.

Setiap objek hasil lokalisasi sebelumnya akan dipilah untuk memperkecil besar komputasi. Citra lokalisasi tidak akan dihitung dengan *mahalanobis distance*, jika :

1. Objek terlalu besar atau terlalu kecil.
2. Nilai kepersegi panjang kurang dari 1400 piksel
3. Aspek rasio di luar batas 2 sampai 4

Data latih dari penelitian dibentuk sebagai berikut:

Tabel 3.1. Vektor uji klasifikasi

Kelas	Fitur	<i>Ar</i>		<i>R</i>		<i>Ed</i>	
		<i>mean</i>	<i>C</i>	<i>mean</i>	<i>C</i>	<i>mean</i>	<i>C</i>
Kelas1							
Kelas2							
Bukan Plat							

Jika $d1 > d3$ atau $d2 > d3$ maka dianggap sebagai citra plat nomor. Algoritma klasifikasi mahalanobis ditunjukkan pada gambar 3.18

```

1.  Inisialisasi nilai vector X,M1,M2,M3,C1,C2,C3
2.  Hitung nilai D1,D2,D3 dengan rumus
    RC[0],RC[1],RC[2] ← 0;
    For i←0 to 2 do
        For j ← 0 to 2 do
            R[j] ← X-M[i];
            RC[i] ←RC[i] + X[i]*C[i]/SQRT(C[0]2+ C[1]2+ C[2]2)
            D[i] ←
            SQRT((RC[i]*R[0])2+(RC[i]*R[1])2+(RC[i]*R[j])2)
3.  Jika D1>D3 atau D2>D3 maka
    3.1 vektor tersebut adalah plat nomor
    3.2 vektor tersebut bukan plat nomor

```

Gambar 3.18. Algoritma klasifikasi Mahalanobis

3.2.1.6. Kliping Citra Plat Nomor

Kandidat-kandidat citra plat nomor, yang telah diuji kelayakan geometrisnya, diambil titik koordinat ujung-ujungnya untuk melakukan pemotongan (kliping) citra awal, sehingga citra yang dihasilkan menjadi seperti gambar 3.19. pada citra yang telah diklip tersebut dapat dilakukan segmentasi karakter.



Gambar 3.19. Citra Plat nomor yang telah diklipping

3.2.1.7. Lokalisasi Karakter

Citra hasil Kliping sekali lagi dilakukan binerisasi, beserta deteksi tepi kemudian dilakukan lokalisasi. Pada akhir proses, hanya karakter nomor kendaraan bermotor yang telah

disegmentasi dari citra awalnya yang menjadi hasil akhir perangkat lunak.

3.3. Pengujian perangkat lunak untuk menguji ketepatan penggunaan algoritma dalam penyelesaian masalah.

3.3.1. Pengukuran Tingkat Kesalahan dan uji proporsi

Pengukuran tingkat kesalahan dilakukan untuk memperoleh ukuran ketepatan sistem dalam mengenali citra plat nomor yang terdapat dalam citra. Tingkat kesalahan yang diukur dinyatakan dengan perhitungan pada rumus 3.5. dan tabel pengujian seperti tabel 3.2

$$error = \frac{|jumlah_sampel - jumlah_dikenali|}{jumlah_sampel} \quad (3.5)$$

Tabel 3.2. Rancangan Tabel Pengujian Citra

No	Input	kontras	Ar	R	Ed	Hasil
error						

Pengukuran tingkat kesalahan ini menggunakan data sampel citra yang diujikan. Setiap kendaraan yang diuji, diambil lima sampel citra, kemudian citra tersebut, dianalisa tingkat keakuratan dengan nilai batasan ketika hendak dilakukan binerisasi, sehingga menghasilkan kesimpulan dari data yang ada.

3.4. Rancangan Basis Data

Dalam penelitian ini, untuk dapat mengenali fitur dari plat nomor citra yang hendak dikenali, perlu penyimpanan data latih ke dalam *database*. *Database* yang digunakan terdiri 3 tabel yaitu tabel DataLatih, DataUji, DataKlasifikasi.

Tabel 3.3. Tabel DataLatih

<i>Field</i>	<i>Panjang</i>	<i>Allow Null</i>
NamaCitra	Text(255)	No
EdgeDensity	Number	yes
Ratio	Number	yes
Ao	Number	yes
MER	Number	yes
LuasMinimal	Number	yes
AmbangKontras	Number	yes
InputPoint	Number	yes
tanggalLatih	date/time	yes

Tabel 3.4. Tabel DataUji

<i>Field</i>	<i>Panjang</i>	<i>Allow Null</i>
NamaCitra	Text(255)	No
EdgeDensity	Number	Yes
Ratio	Number	Yes
Ao	Number	Yes
MER	Number	Yes
LuasMinimal	Number	Yes
AmbangKontras	Number	Yes
InputPoint	Number	Yes
tanggalUji	date/time	Yes
Benar	Yes/no	No

Tabel 3.5. Tabel DataKlasifikasi

<i>Field</i>	<i>Panjang</i>	<i>Allow Null</i>
NamaKelas	Text(255)	No
EdgeDensity	Number	Yes
Ratio	Number	Yes
Ao	Number	Yes
C_EdgeDensity	Number	Yes
C_Ratio	Number	Yes
C_Ao	Number	Yes

BAB IV

HASIL DAN PEMBAHASAN

4.1. Lingkungan Implementasi

Perangkat keras yang digunakan dalam pembuatan aplikasi penjadwalan sumber daya proyek adalah sebagai berikut :

1. Prosesor Intel Atom 1.6 Ghz
2. Memori 2 GB
3. Harddisk dengan kapasitas 320 GB
4. Monitor LCD 10.1''
5. Keyboard
6. Mouse

Perangkat lunak yang digunakan dalam pembuatan aplikasi penjadwalan sumber daya proyek adalah sebagai berikut :

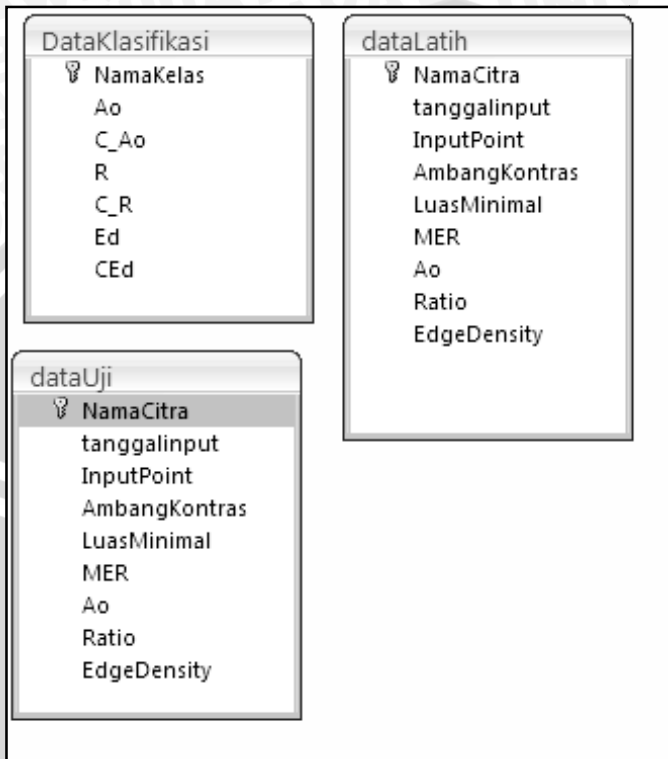
1. Sistem Operasi Windows XP
2. Borland Delphi 7
3. Microsoft Access 2003
4. ACD See 7.0

Kamera yang digunakan dalam pengumpulan data adalah sebagai berikut :

1. Canon Power Shot SX 1 IS
2. Canon EOS 300D
3. Canon EOS 1000D
4. Nikon D60

4.2. Implementasi Basis Data

Pada bab III telah dijelaskan rancangan basis data yang berisi tabel Data untuk menyimpan data latih dan data pengujian serta data klasifikasi. Tabel tersebut diimplementasikan ke dalam *Microsoft Access 2003*.



Gambar 4.1 Struktur Basis Data

4.3. Implementasi Program

Sesuai yang telah dibahas di bab III, tahap pengolahan citra dibagi menjadi tujuh tahap dan masing-masing tahap ditulis dalam sebuah procedure yang didefinisikan dalam sebuah kelas yang diberi nama “Tical”.

```

type TIcal=Class
    mImage : TBitmap;
    Mask : T3x3FloatArray;
    MaskX, MaskY : T3x3FloatArray;
    posX, posY:integer;
    warnaSkr:tcolor;
    arrayKordinatImage:array[0..999]of
        tkordinatimage;
    jumArrayKI:integer;
    constructor Create;
    destructor free;
    function msa(const nInputPoint, iterasi,
        radius:integer; const img:timage):
        Tbitmap;
    function ColorDetection(const cl1,cl2:
        TColor;tolerance:real):boolean;
    function lokalisasi(ToleransiLokalisasi,
        minpixelsvalue:Integer;const
        img:timage):TBitmap;
    procedure countRecursive(var imageI3: TImage;
        const curcolor:TColor;
        const X,Y,Tolerance : integer;
        var area : tkordinatimage);
    procedure binarisasi(top:integer);
    function MClassification(x_uji: TVector;
        statistik,covariance:T3x3FloatArray):
        boolean;
    function
        SobelEdge2(Sg1,Sg2:TstringGrid;ABias,contr
        ass : integer;const img:Timage) : TBitmap;
end;

```

Source Code 4.1. Ical Class

4.3.1. Segmentasi Citra

Segmentasi dilakukan pada citra, sebagaimana telah dijelaskan pada bab III, kode program berikut merupakan implementasi dari algoritma *meanshift*, dengan parameter jumlah input point, jumlah iterasi, radius, dan citra asal. Input poin merupakan nilai titik awal di aman kernel akan bergerak.

Pada program, pemilihan koordinat titik tersebut dilakukan secara acak.

```
msa(const nInputPoint,iterasi,Radius: integer; const
    img: timage): TBitmap;
var
    temp2,temp1:tbitmap;
    inputPointColor: TColor;
    inputPointRed : integer;
    currentPixelRedDistance : integer;
    currentPixelColorR:byte;
    i,j,a,b,c,d,e : integer;
    accumulatedRedValues : double;
    totalClusterPixels : integer;
    row1,row2:PRGBTripleArray;
begin
    temp1:=img.Picture.Bitmap;
    temp2:=tbitmap.create;
    temp2.Width:=temp1.Width;
    temp2.Height:=temp1.Height;
    temp2.PixelFormat:=pf24bit;
    mImage:=img.Picture.Bitmap;
    for i:=0 to temp1.Height-1 do
        begin
            row1:=Mimage.ScanLine[i];
            row2:=temp2.ScanLine[i];
            for j:=0 to MImage.Width-1 do
                begin
                    row2[j].rgbtRed:=row1[j].rgbtRed;
                end;
            end;
        end;

    for c := 0 to nInputPoint-1 do
        begin

            inputPointColor :=
                TColor(img.Canvas.Pixels[random(widht),random(hei
                    ght)]);
            // given input point updated and processed
            prop.numberOfIterations times
            for d := 1 to iterasi do
                begin
                    inputPointRed := GetRValue(inputPointColor);

                    accumulatedRedValues := 0;
                    totalClusterPixels := 0;
```

Source Code 4.2. Msa Filtering

```

for a := 0 to temp1.height-1 do
begin
row1:=temp1.ScanLine[a];
row2:=temp2.ScanLine[a];
  for b := 0 to temp1.width-1 do
    begin
      currentPixelColorR := row1[b].rgbtRed;
      currentPixelRedDistance := currentPixelColorR -
inputPointRed;
      if (((currentPixelRedDistance *
currentPixelRedDistance) < Radius) then
        begin
          inc(totalClusterPixels);
          accumulatedRedValues := accumulatedRedValues +
currentPixelColorR;

          end;
        end;
      end;

      inputPointColor := rgb(Round(accumulatedRedValues /
totalClusterPixels)
      end;
    end;

mImage:=temp2;
result:=mImage;
end;

```

Source Code 4.2. Msa Filtering (lanjutan)

Hasil citra yang telah disegmentasi warnanya akan dilanjutkan dengan pendeteksian tepi.

4.3.2. Deteksi Tepi

Deteksi diimplementasikan dalam program, dengan membuat fungsi dalam kelas, yang memiliki parameter matriks *maskx*, *masky*, bias, kontras dan citra asal, menghasilkan citra yang telah dideteksi tepinya.


```

//berikut nilai citra dikalikan matriks y
LNewBlue2 :=
    (LRow1[LCol-1].rgbtBlue*MaskY[0,0]) +
    (LRow1[LCol].rgbtBlue*MaskY[1,0]) +
    (LRow1[LCol+1].rgbtBlue*MaskY[2,0]) +
    (LRow2[LCol-1].rgbtBlue*MaskY[0,1]) +
    (LRow2[LCol].rgbtBlue*MaskY[1,1]) +
    (LRow2[LCol+1].rgbtBlue*MaskY[2,1]) +
    (LRow3[LCol-1].rgbtBlue*MaskY[0,2]) +
    (LRow3[LCol].rgbtBlue*MaskY[1,2]) +
    (LRow3[LCol+1].rgbtBlue*MaskY[2,2]);

LNewBlue2 := (LNewBlue2 / LCoef) + ABias;
if LNewBlue2 > 255 then LNewBlue2 := 255;
if LNewBlue2 < 0 then LNewBlue2 := 0;

LRowOut[LCol-1].rgbtBlue :=
trunc(sqrt(LNewBlue*LNewBlue+LNewBlue2*LNewBlue2));
end;
end;

binarisasi(contrast);
result:=mImage;
end;

```

Source Code 4.3. Sobel Edge Detection (lanjutan)

4.3.3. Binarisasi

Hasil dari deteksi tepi, menghasilkan citra dengan tepian yang memiliki warna bermacam-macam (walaupun lebih sederhana), sementara untuk proses selanjutnya adalah proses lokalisasi, yang berarti memindahkan pixel-pixel dengan warna putih yang tidak terputus untuk kemudian dikenali. Pada proses ini, citra mengalami binerisasi, agar menjadi mutlak hitam dan putih, dengan batasan tertentu, sehingga pada warna yang di bawah batas, akan dianggap hitam, dan sebaliknya dianggap putih. Berikut adalah fungsi binarisasi citra dalam program.

```

for i:=0 to MImage.Height-1 do
begin
row1:=mImage.ScanLine[i];
row2:=temp.ScanLine[i]; //mImage.ScanLine[i];
for j:=0 to MImage.Width-1 do
begin
if row1[j].rgbtRed < bot then r1:=0
else if row1[j].rgbtRed >= top then r1:= 255 ;
if row1[j].rgbtGreen < bot then g1:=0
else if row1[j].rgbtGreen >= top then g1:= 255 ;
if row1[j].rgbtBlue < bot then b1:=0
else if row1[j].rgbtBlue >= top then b1:= 255 ;

if r1+g1+b1 > 0 then
begin
row2[j].rgbtRed:=255;
row2[j].rgbtGreen:=255;
row2[j].rgbtBlue:=255;
end
else
begin
row2[j].rgbtRed:=0;
row2[j].rgbtGreen:=0;
row2[j].rgbtBlue:=0;
end;
end;
end;
end;

```

Source Code 4.4. binarisasi

4.3.4. Lokalisasi

Setiap pixel yang tidak terputus akan dicatat dalam sebuah variabel yang menyimpan jumlah luasan serta koordinat tiap titik yang diuji. Selain itu, diberikan nilai tertentu yang akan menyatakan luas minimal citra yang tidak dianggap sebagai pixel pengganggu (*noise*). Proses lokalisasi gambar dilakukan dengan metode *region growing* atau deteksi tetangga dengan warna yang sama seperti yang didefinisikan pada program.

```

jumArrayKI:=0;
temp:=TImage.Create(nil);
temp.Picture.Bitmap:=img.Picture.Bitmap;

for i:=0 to temp.Picture.Width-1 do
  for j:=0 to temp.Picture.Height-1 do
    begin
      if (temp.Canvas.Pixels[i,j]= CLWhite) then
        begin
          arrayKordinatImage[jumArrayKI].luas:=0;
          countRecursive(temp,
                        temp.Canvas.Pixels[i,j],i,j,ToleransiLo
                        kalisasi,arrayKordinatImage[jumArrayKI]
                        );
          if arrayKordinatImage[jumArrayKI].luas >=
            minpixelsvalue then
            inc(jumArrayKI);
          end;
        end;
      end;
    end;
  end;
end;

```

Source Code 4.5. Scan Image Hasil Binarisasi, dan Melakukan Penelusuran Warna Putih

```

procedure Tical.countRecursive(var image13: TImage; const
  curcolor: TColor;
  const X, Y, Tolerance: integer; var area:
  tkordinatimage);
begin
  if (X+1 >=image13.Picture.Bitmap.Width) or
    (Y+1 >= image13.Picture.Bitmap.Height) or
    (X-1<0) or
    (Y-1<0)
  then exit;
  image13.Canvas.Pixels[X,Y]:=clFuchsia;
  inc(area.luas);
  area.posisi[area.luas].x:=x;
  area.posisi[area.luas].y:=y;

```

Source Code 4.6. Deteksi Warna Putih yang Tidak Terputus dan Disimpan pada Variabel untuk Lokalisasi

```

if(image13.Canvas.Pixels[X+1,Y]=clwhite) then
    countRecursive(image13,curColor,X+1,Y,area);
if (image13.Canvas.Pixels[X-1,Y]=clwhite) then
    countRecursive(image13,curColor,X-1,Y,area);
if (image13.Canvas.Pixels[X,Y-1]=clwhite) then
    countRecursive(image13,curColor,X,Y-1,area);
if (image13.Canvas.Pixels[X,Y+1]=clwhite) then
    countRecursive(image13,curColor,X,Y+1,area);
if (image13.Canvas.Pixels[X+1,Y+1]=clwhite) then
    countRecursive(image13,curColor,X+1,Y+1 ,area);
if (image13.Canvas.Pixels[X-1,Y+1]=clwhite) then
    countRecursive(image13,curColor,X-1,Y+1,area);
if (image13.Canvas.Pixels[X-1,Y-1]=clwhite) then
    countRecursive(image13,curColor,X-1,Y-1,area);
if (image13.Canvas.Pixels[X+1,Y-1]=clwhite) then
    countRecursive(image13,curColor,X+1,Y-1,area); end;

```

Source Code 4.6. Deteksi Warna Putih yang Tidak Terputus dan Disimpan pada Variabel untuk Lokalisasi (lanjutan)

4.3.5. Klasifikasi Mahalanobis

Pada citra yang mengalami lokalisasi, diekstrak 3 buah fitur untuk dianalisa dalam *mahalanobis distance*, dalam program, diimplementasikan sebagaimana *source code 4.7*.

```

function TIcal.MClassification(x_uji:Tvector;
statistik,
    covariance: T3x3FloatArray): boolean;
var i:integer;
c1,c2,c3,d1,d2,d3:extended;
begin
c1:=sqrt(sqr(covariance[0,0])+sqr(covariance[1,0])+sqr
(covariance[2,0]));
c2:=sqrt(sqr(covariance[0,1])+sqr(covariance[1,1])+sqr
(covariance[2,1]));
c3:=sqrt(sqr(covariance[0,2])+sqr(covariance[1,2])+sqr
(covariance[2,2]));
d1:=0;d2:=0;d3:=0;
for i:=0 to 2 do
begin
    d1:=d1+(x_uji[i]-
        statistik[i,0])*(covariance[i,0]/c1);
    d2:=d2+(x_uji[i]-
        statistik[i,1])*(covariance[i,1]/c2);
    d3:=d3+(x_uji[i]-
        statistik[i,2])*(covariance[i,2]/c3);
end;

```

Source Code 4.7. Klasifikasi Mahalanobis Distance

```

d1:=
sqrt(sqr(d1*x_uji[0])+sqr(d1*x_uji[1])+sqr(d1*x_uji[2]
))
d2:=
sqrt(sqr(d2*x_uji[0])+sqr(d2*x_uji[1])+sqr(d2*x_uji[2]
))
d3:=
sqrt(sqr(d3*x_uji[0])+sqr(d3*x_uji[1])+sqr(d3*x_uji[2]
))
if (d1>d3)or(d2>d3) then
result:=true
else Result := false; end;

```

Source Code 4.7. Klasifikasi Mahalanobis Distance (lanjutan)

4.3.6. Pemotongan Citra (*klipling*) Plat Nomor

Citra yang telah dianalisa dan dianggap sebagai citra plat nomor, koordinatnya disimpan dan digunakan untuk melakukan pemotongan pada citra asal. Sehingga menghasilkan output file dalam format jpg yang berisi plat nomor kendaraan yang telah dideteksi sekaligus menghitung Ao, R, dan Ed.

```

tested:=arrayKordinatImage[index];
a.x:=10000;a.Y:=10000;b.X:=0;b.Y:=10000;
c.Y:=0;c.X:=0;d.X:=10000;d.Y:=0;
for j:=1 to tested.luas-1 do
begin
if tested.posisi[j].X<a.X then //xmin
begin
a.X:=tested.posisi[j].X;
d.X:=tested.posisi[j].X; end;
if tested.posisi[j].y<a.Y then //ymin
begin
a.y:=tested.posisi[j].y;
b.y:=tested.posisi[j].y; end;
if tested.posisi[j].X>b.X then //xmax
begin
b.X:=tested.posisi[j].X;
c.X:=tested.posisi[j].X; end;
if tested.posisi[j].y>d.y then //ymax
begin
d.y:=tested.posisi[j].y;
c.y:=tested.posisi[j].y; end;
end;
end;

```

Source Code 4.8. Menghitung Titik Citra yang Dipotong

```

{
    A(xmin,ymin)                                B(xmax,ymin)
    *-----*
    |               |
    |               |
    *-----*
    D(xmin,ymax)                                C(xmax,ymax)
}
// gambarpersegi(image13,a,b,c,d);

temp:=timage.Create(nil);
temp.Picture.Bitmap:=tbitmap.Create;
temp.Picture.Bitmap.Width:=round(nilaivector(a,b));
temp.Picture.Bitmap.Height:=round(nilaivector(a,d));
temp.Picture.Bitmap.PixelFormat:=pf24bit;

for i:=a.X to b.X do
    for j:=a.Y to d.Y do
        begin
            temp.Canvas.Pixels[i-a.X,j-a.Y]:=
sourceimg.Canvas.Pixels[i,j];
        end;
        tested.R:=nilaivector(a,b)/nilaivector(a,d);
        tested.Ao:=(nilaivector(a,b)*nilaivector(a,d));
        tested.Ed:=2*nilaivector(a,b)+2*nilaivector(a,d);
        arrayKordinatImage[index]:=tested;
    result:=temp.picture.bitmap;
}

```

Source Code 4.8. Menghitung Titik Citra yang Dipotong

Pemotongan citra akan diteruskan sampai pada karakter yang hendak dicari, dengan melakukan proses lokalisasi ulang dengan melakukan binarisasi pada citra hasil pemotongan. Kemudian dilakukan penelusuran warna putih, yang merupakan daerah karakter plat nomor. Dengan mengulang metode *region growing* maka akan didapatkan bagian dari citra plat nomor yang merupakan daerah karakter. Karakter hasil segmentasi tersebut merupakan *pre processing* untuk dapat dilakukan pengenalan karakter. Proses tersebut ditunjukkan pada *Source Code 4.9*, serta hasil dari seluruh proses pengolahan citra ditunjukkan pada gambar 4.2.

```

IcalImage.lokalisasi(0, SeMin2.Value, CFoundedPlat);
j:=0;
for i:=0 to 10 do
begin
  bmp[i]:=tbitmap.Create;
  bmp[i].PixelFormat:=pf24bit;
  bmp[i].Width:=1;
  bmp[i].Height:=1;
end;
for i:=0 to LBChar.Items.Count-1 do
begin
  if icalimage.arrayKordinatImage[i].R < 1 then
  begin
    bmp[j]:=
    icalImage.analisaBentuk(chasil, clblue, 9, i);
    inc(j);
  end;
end;
cchar0.Picture.Bitmap:= bmp[0];

```

Source Code 4.9. Lokalisasi Karakter

Semua proses akan menghasilkan citra sebagai berikut.



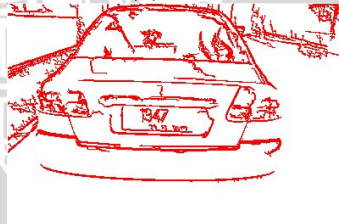
(a)



(b)



(c)



(d)



(e)

N I 9 4 7 C J

(f)

(a)Citra awal (b)Citra tersegmentasi (c)Citra deteksi tepi
(d)citra terlokalisasi (e)citra plat nomor yang telah terdeteksi
dan dilokalisasi (f)Lokalisasi karakter dalam citra plat nomor

Gambar 4.2. Hasil Pemrosesan Citra

4.4. Pelatihan

Proses pelatihan digunakan untuk mencari data kelas plat nomor sebagaimana telah dijelaskan di Bab III, sehingga diperoleh data yang diinginkan untuk melakukan klasifikasi *mahalanobis*. Proses ini menyimpan tiga fitur utama plat nomor beserta fitur non plat nomor. Hal ini dilakukan secara manual, dengan memilih daerah yang dianggap plat nomor. Proses penyimpanan data latih ditunjukkan pada source code 4.10

```
qInsert.ParamByName('namaCitra').Value:=NamaFileBmp;  
qInsert.ParamByName('inputpoint').Value:=seInput.Valu  
e;  
qInsert.ParamByName('ambangkontras').Value:=SeKontras  
s.Value;  
qInsert.ParamByName('luasminimal').Value:=SeMinPixels  
.Value;  
qInsert.ParamByName('mer').Value:=SeAmer.Value;  
qInsert.ParamByName('ao').Value:=IcalImage.arrayKordi  
natImage[listbox1.ItemIndex].Ao;  
qInsert.ParamByName('ratio').Value:=IcalImage.arrayKo  
rdinatImage[listbox1.ItemIndex].R;  
qInsert.ParamByName('edgedensity').Value:=IcalImage.a  
rrayKordinatImage[listbox1.ItemIndex].Ed;  
qInsert.Parameters.ParamByName('tanggalinput').Value:  
=now;  
QInsert.ExecSQL;
```

Source Code 4.10. Mencatat data pelatihan

Data hasil pelatihan pada citra latih menghasilkan data sebagai berikut, mAo merupakan rata-rata dari jumlah pixel area plat nomor, C_mAo merupakan covarian dari mAo, mR

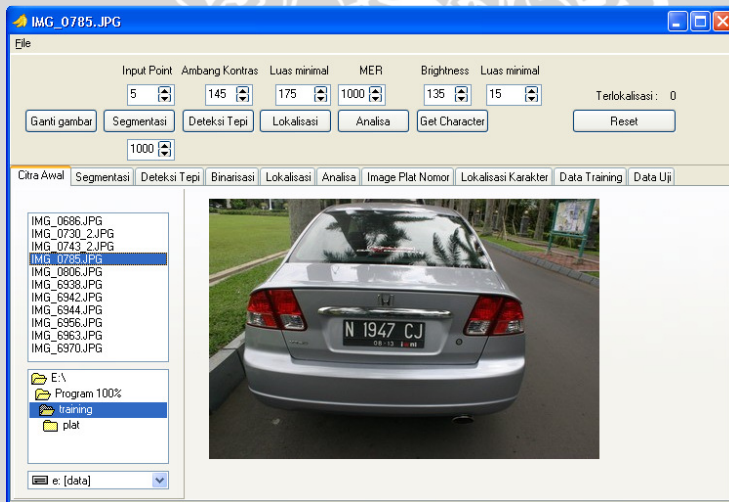
merupakan nilai rata-rata perbandingan panjang dengan lebar, C_mR merupakan nilai covarian dari mR, dan mEd merupakan nilai rata-rata dari kepadatan tepi, C_mEd merupakan nilai covarian dari mEd.

Tabel 4.1. Data Hasil Pelatihan

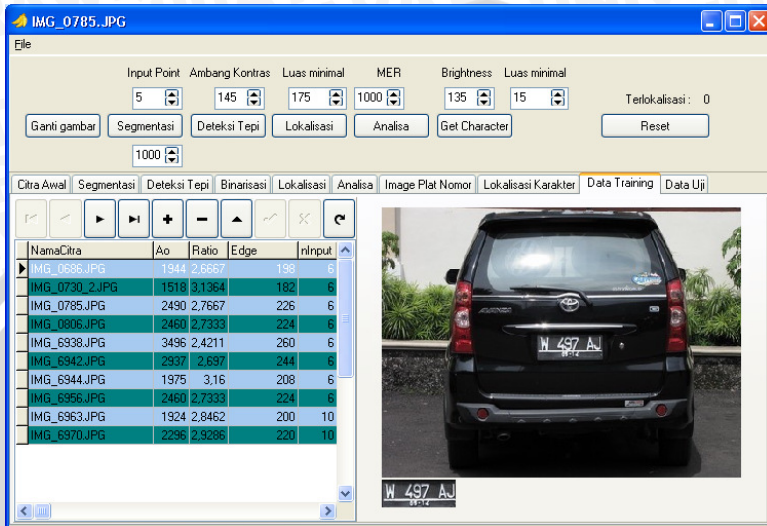
Nama Kelas	mAo	C_mAo	mR	C_mR	mEd	mCEd
NONPLAT	2600	950,7072	3,2141	0,61948	200,56	45,8015
PLAT1	2400	570,054	2,85982	0,230930	220,62	23,89835
PLAT2	2350	566,0547	2,80891	0,2217609	218,6	22,843428

4.5. Implementasi Antar Muka

Berdasarkan rancangan antarmuka pada bab sebelumnya, maka dibuatlah antarmuka seperti ditunjukkan pada gambar 4.3 dan 4.4.



Gambar 4.3. Antarmuka program



Gambar 4.4. Antarmuka program (Data Pelatihan)

4.6. Implementasi Uji Coba

Pada subbab ini akan dilakukan pembahasan mengenai pengujian yang telah dilakukan pada sistem dan hasil pengukuran tingkat kesalahan lokalisasi karakter plat nomor. Serta melakukan uji keakuratan terhadap data yang telah berhasil melakukan deteksi dan segmentasi citra plat nomor dengan mengelompokkan masing-masing sudut pengambilan citra.

4.7. Analisa Hasil Pengujian

Histogram warna mewakili nilai tiap *range* warna di antara 0-255. Semakin besar frekuensi warna dengan nilai tertentu, menunjukkan pixel-pixel pada gambar memiliki warna dengan nilai tersebut dalam jumlah yang lebih besar dari warna lainnya. Secara matematis, algoritma *mean shift* yang berhasil diterapkan, akan membuat nilai warna tertentu menjadi lebih banyak daripada yang lainnya dan sebaliknya. Sehingga pada citra yang telah mengalami algoritma, histogramnya akan menjadi berbentuk gelombang, dengan lebih dari 1 puncak dan 1 lembah.

Pada tahap pengujian, setiap kendaraan yang diuji, diambil 5 (lima) citra dengan perbedaan sudut pengambilan. Pada tabel 4.2. menunjukkan program mendeteksi citra plat nomor L1533CA sebanyak 5 sudut pengambilan, dan mampu melokalisasi plat nomornya.

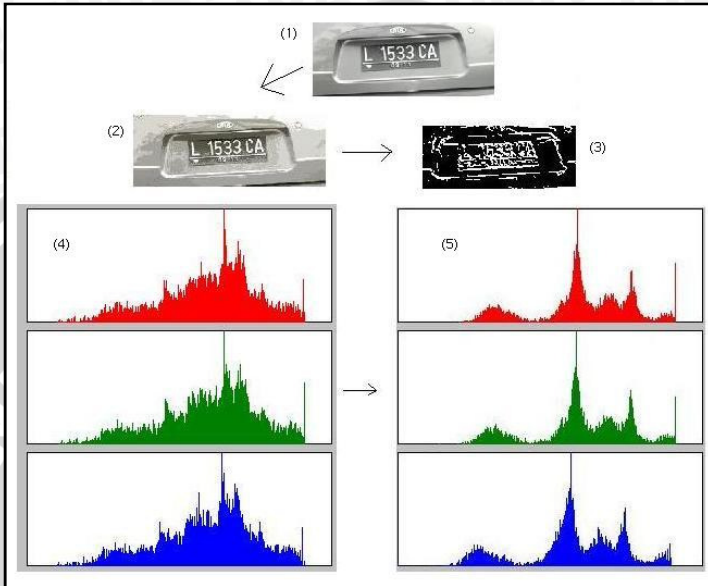
Tabel 4.2 Pengujian pada citra mobil L 1533 CA

File	Citra	Terdeteksi
L1533CA(1) .jpg		Ya
L1533CA (2) .jpg		Ya
L1533CA (3) .jpg		Ya

L1533CA (4).jpg		Ya
L1533CA.jpg		Ya

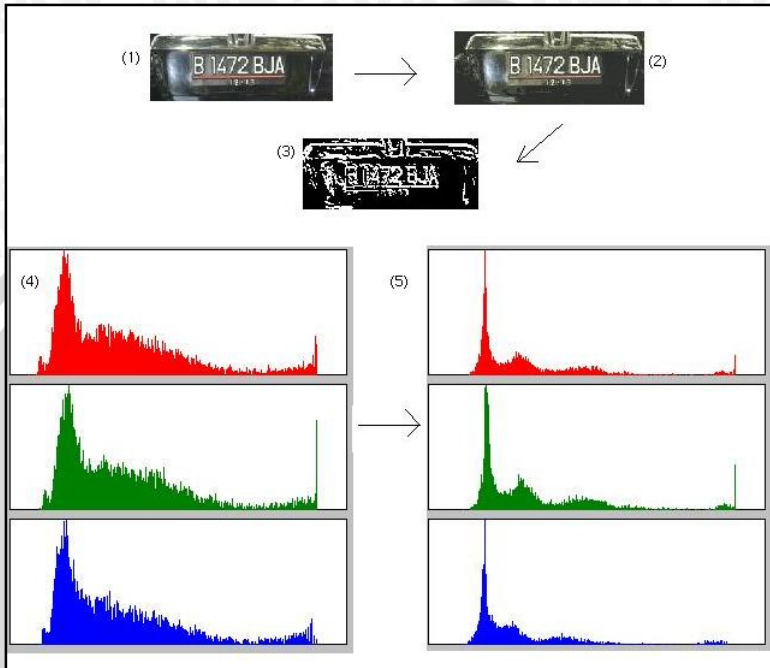
Analisa pada proses plat nomor L 1533 CA terlihat bahwa segmentasi pada citra L1533CA berhasil, hal ini dapat dianalisa pada histogram warna kandidat citra pada gambar 4.5 (1), dan (2) serta histogram (4) dan (5), di mana distribusi warna pada kandidat citra dapat memberikan hasil deteksi tepi yang baik, yaitu berupa hasil deteksi tepi kandidat daerah citra memiliki tepian citra yang tidak terputus (3) dan dapat ditelusuri serta dianalisa dengan klasifikasi mahalanobis.

Hasil pengujian pada mobil L 1533 CA berhasil sepenuhnya dalam mendeteksi plat nomor pada ke-5 citra.



Gambar 4.5. kandidat plat nomor L 1533 CA. (1) citra awal, (2) citra tersegmentasi, (3) citra tepi, (4) histogram citra awal, (5) histogram citra tersegmentasi

Analisa lainnya pada proses plat nomor B 1472 BJA bahwa segmentasi pada citra nomor B 1472 BJA mengalami kegagalan, hal ini dapat dianalisa pada histogram warna kandidat citra pada gambar 4.6 (1), dan (2) serta histogram (4) dan (5), di mana distribusi warna pada kandidat citra tidak dapat memberikan hasil deteksi tepi yang baik, yaitu memiliki tepian citra kandidat plat nomor yang terputus sehingga hasil deteksi tepi kandidat daerah citra memiliki tepian citra yang tidak dapat dianalisa (3). Hasil pengujian pada mobil B 1472 BJA berhasil hanya pada 1 citra dari 5 citra yang diujikan. Hal ini menunjukkan bahwa segmentasi warna pada citra mempengaruhi akurasi dari proses deteksi tepi.



Gambar 4.6. kandidat plat nomor B 1472 BJA. (1) citra awal, (2) citra tersegmentasi, (3) citra tepi, (4) histogram citra awal, (5) histogram citra tersegmentasi

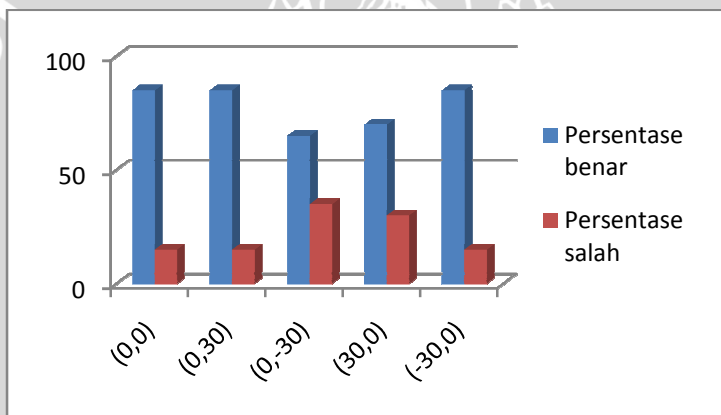
Dari pengujian terhadap 5 citra untuk setiap kendaraan masing-masing file dengan sudut (0,0) diberi nama file yang memiliki ketentuan xxxxxxxx(1).jpg, dengan xxxxxxxx merupakan plat nomor kendaraan yang diuji. Untuk sudut (0,30) diberi nama xxxxxxxx(2).jpg. Untuk sudut (0,-30) diberi nama xxxxxxxx (3).jpg. Untuk sudut (30,0) diberi nama xxxxxxxx (4).jpg. Untuk sudut (-30,0) diberi nama xxxxxxxx (5).jpg. Data lengkap hasil pengujian dapat dilihat di lampiran 1.

Pada 20 mobil, program dapat mendeteksi dan melokalisasi citra plat nomor. Analisa terhadap data sebanyak 100 mobil menunjukkan program mampu mendeteksi plat nomor dengan klasifikasi mahalanobis dengan

mengelompokkan data uji masing-masing pada sudut pengambilan diperoleh nilai akurasi yang dijelaskan pada tabel 4.3. dan gambar 4.7,

Tabel 4.3. Hasil Pengujian

Sudut citra	Persentase benar	Persentase salah
(0,0)	85	15
(0,30)	85	15
(0,-30)	65	35
(30,0)	70	30
(-30,0)	85	15
Rata-rata	78	22



Sehingga dapat disimpulkan dalam penelitian bahwa faktor yang mempengaruhi akurasi adalah :

1. Pengaruh warna latar belakang mobil.
2. Tepian citra plat nomor yang tidak terputus pada waktu dilakukan deteksi tepi.
3. Sudut pengambilan citra

UNIVERSITAS BRAWIJAYA



BAB V PENUTUP

5.1. Kesimpulan

Dari pengerjaan serta penelitian yang telah dilakukan, serta mengacu pada tujuan dari penulisan skripsi, maka dapat disimpulkan beberapa hal, diantaranya:

1. Algoritma Mean Shift dan Deteksi Tepi Sobel dapat diterapkan dalam perangkat lunak untuk melakukan deteksi plat nomor mobil pribadi di Indonesia.
2. Pada tahap pengujian aplikasi dengan 20 mobil, dengan masing-masing mobil diambil 5 citra memberikan hasil kesalahan rata-rata sebesar 22% sehingga tingkat ketepatan aplikasi dalam mendeteksi citra plat nomor dari sebuah citra yang berisi obyek lain sebesar 78%.

5.2. Saran

1. Perlu perbandingan metode lain dalam melakukan lokalisasi citra plat nomor dan segmentasi karakter sehingga dapat diperoleh data yang terukur tentang efektifitas metode ini dalam melakukan fungsinya.
2. Penelitian ini dapat diteruskan hingga menjadi sebuah aplikasi yang dapat melakukan deteksi dan pengenalan plat nomor yang efektif dalam penggunaannya pada kendaraan pribadi maupun kendaraan lainnya di Indonesia.

UNIVERSITAS BRAWIJAYA



DAFTAR PUSTAKA

- Agushinta, R. D. Diyanti, A. *Perbandingan Kinerja Metode Deteksi Tepi pada Citra Wajah*. Dewiar @staff.gunadarma.ac.id. Universitas Gunadarma. Tersedia pada : <http://repository.gunadarma.ac.id:8000/browse.php?nfile=785> [28 desember 2009]
- Anonim. *Deteksi Tepi Menggunakan Metode Robert, Sobel, Prewitt, dan Laplacian*. Tersedia pada <http://balqsilmi.blogspot.com/2009/11/deteksi-tepi-menggunakan-metode-robert.html> [4 januari 2010]
- Daniel, Wayne W. 1989. *Statistika Non Parametrik Terapan*. Gramedia. Jakarta.
- Jia Wenjing. Zhang, Huaifeng. He, Xian. 2005. *Mean Shift for Accurate Number Plate Detection*. Department of Computer Systems Faculty of Information Technology University of Technology : Sydney.
- Mahalanobis, P C (1936). "On the generalised distance in statistics". Proceedings of the National Institute of Sciences of India.
- Mclahan, GJ.1999.Mahalanobis Distance. Departement of Mathematics University of Queenslane. Australia.
- Mike Constant.1999.An Introduction to ANPR. Tersedia pada:http://www.cctvinformation.co.uk/i/An_Introduction_to_ANPR [18 November 2010]
- Riyanto. *Praktikum Citra – Deteksi Tepi (Edge Detection)*. Tersedia di <http://www.eepis-its.edu/~riyanto/citra-bab8.pdf> [28 desember 2009]
- Siegel, Sidney. 1994. *Statistika NonParametrik untuk Ilmu-Ilmu Sosial*. PT. Gramedia Pustaka Utama. Jakarta.

Soeharto.1993.Peraturan Pemerintah No.44 tahun 1993
Tentang Kendaraan dan Pengemudi.Jakarta

Tan and Kumar. 2004. *Introduction to Data Mining : lecture
note for chapter 1.*

[Http://www.cse.msu.edu/~cse980](http://www.cse.msu.edu/~cse980)



LAMPIRAN

Lampiran 1. Data hasil penelitian

NamaCitra	benar	MER	Ao	Ratio	EdgeDensity
B 1472 BJA(1).jpg	TRUE	1000	1992	3,458333	214
B 1472 BJA(2).jpg	FALSE	1000	3468	3	272
B 1472 BJA(3).jpg	FALSE	1000	1508	2,230769	168
B 1472 BJA(4).jpg	TRUE	1000	2187	3	216
B 1472 BJA(5).jpg	TRUE	1000	1512	3,428571	186
B 2925 QF(1).jpg	TRUE	1000	1775	2,84	192
B 2925 QF(2).JPG	TRUE	1000	1960	2,5	196
B 2925 QF(3).jpg	TRUE	1000	2280	2,533333	212
B 2925 QF(4).JPG	TRUE	1000	2325	2,419355	212
B 2925 QF(5).JPG	TRUE	1000	1704	2,958333	190
BA 2596 TI(1).jpg	TRUE	1000	3255	2,657143	256
BA 2596 TI(2).jpg	TRUE	1000	2050	3,28	214
BA 2596 TI(3).jpg	TRUE	1000	2752	2,6875	236
BA 2596 TI(4).jpg	TRUE	1000	3298	2,852941	262
BA 2596 TI(5).jpg	TRUE	1000	2436	3,107143	230
KT 1489 P(1).jpg	FALSE	1000	1680	2,916667	188
KT 1489 P(2).jpg	FALSE	1000	1680	2,916667	188
KT 1489 P(3).jpg	FALSE	1000	1680	2,916667	188
KT 1489 P(4).jpg	FALSE	1000	1680	2,916667	188
KT 1489 P(5).jpg	FALSE	1000	1680	2,916667	188
L 1244 ZJ(1).jpg	FALSE	1000	2900	3,448276	258
L 1244 ZJ(2).jpg	TRUE	1000	1950	2,884615	202
L 1244 ZJ(3).jpg	TRUE	1000	1540	3,181818	184
L 1244 ZJ(4).jpg	FALSE	1000	2900	3,448276	258
L 1244 ZJ(5).jpg	TRUE	1000	1680	2,916667	188
L 1533 CA(1).jpg	TRUE	1000	1800	3,125	198

L 1533 CA(2).jpg	TRUE	1000	1512	3,428571	186
L 1533 CA(3).jpg	TRUE	1000	2146	2,551724	206
L 1533 CA(4).jpg	TRUE	1000	2310	2,566667	214
L 1533 CA(5).jpg	TRUE	1000	1800	3,125	198
L 1597 LC(1).jpg	TRUE	1000	1428	3,238095	178
L 1597 LC(2).jpg	TRUE	1000	1428	3,238095	178
L 1597 LC(3).jpg	FALSE	1000	1428	3,238095	178
L 1597 LC(4).jpg	FALSE	1000	1800	3,125	198
L 1597 LC(5).jpg	TRUE	1000	1584	3,272727	188
L 1747 PD(1).JPG	TRUE	1000	2268	3,111111	222
L 1747 PD(2).JPG	TRUE	1000	1474	3,045455	178
L 1747 PD(3).jpg	TRUE	1000	2494	2,965517	230
L 1747 PD(4).JPG	FALSE	1000	2268	3,111111	222
L 1747 PD(5).JPG	TRUE	1000	2494	2,965517	230
N 1127 CW(1).JPG	TRUE	1000	2028	3	208
N 1127 CW(2).JPG	TRUE	1000	2944	2,875	248
N 1127 CW(3).JPG	TRUE	1000	2400	2,666667	220
N 1127 CW(4).JPG	TRUE	1000	2214	3,037037	218
N 1127 CW(5).JPG	TRUE	1000	2160	2,962963	214
N 1258 DQ(1).jpg	TRUE	1000	1890	2,592593	194
N 1258 DQ(2).jpg	FALSE	1000	2944	2,875	248
N 1258 DQ(3).jpg	TRUE	1000	1988	2,535714	198
N 1258 DQ(4).jpg	TRUE	1000	2790	3,1	246
N 1258 DQ(5).jpg	TRUE	1000	2944	2,875	248
N 1398 CL(1).jpg	TRUE	1000	1872	3,25	204
N 1398 CL(2).jpg	TRUE	1000	2184	2,785714	212
N 1398 CL(3).jpg	FALSE	1000	1950	2,884615	202
N 1398 CL(4).jpg	FALSE	1000	2314	3,423077	230
N 1398 CL(5).jpg	TRUE	1000	1950	2,884615	202
N 1632 CT(1).jpg	TRUE	1000	1794	2,653846	190

N 1632 CT(2).jpg	TRUE	1000	1820	2,692308	192
N 1632 CT(3).jpg	TRUE	1000	2268	2,892857	218
N 1632 CT(4).jpg	TRUE	1000	1750	2,8	190
N 1632 CT(5).jpg	FALSE	1000	1750	2,8	190
N 1747 DN(1).jpg	TRUE	1000	2214	3,037037	218
N 1747 DN(2).jpg	TRUE	1000	1452	3	176
N 1747 DN(3).jpg	TRUE	1000	1449	3,285714	180
N 1747 DN(4).jpg	TRUE	1000	2262	2,689655	214
N 1747 DN(5).jpg	TRUE	1000	2050	3,28	214
N 1798 CQ(1).jpg	FALSE	1000	2436	2,896552	226
N 1798 CQ(2).jpg	TRUE	1000	2184	2,785714	212
N 1798 CQ(3).jpg	TRUE	1000	2322	3,185185	226
N 1798 CQ(4).jpg	TRUE	1000	2550	2,833333	230
N 1798 CQ(5).jpg	TRUE	1000	2494	2,965517	230
N 1840 CK(1).jpg	TRUE	1000	2790	3,1	246
N 1840 CK(2).jpg	TRUE	1000	3264	3,1875	268
N 1840 CK(3).jpg	TRUE	1000	2214	3,037037	218
N 1840 CK(4).jpg	TRUE	1000	3300	3,030303	266
N 1840 CK(5).jpg	TRUE	1000	2852	2,967742	246
N 335 AQ(1).jpg	TRUE	1000	2752	2,6875	236
N 335 AQ(2).jpg	TRUE	1000	2296	2,928571	220
N 335 AQ(3).jpg	FALSE	1000	2296	2,928571	220
N 335 AQ(4).jpg	TRUE	1000	2465	2,931034	228
N 335 AQ(5).jpg	TRUE	1000	2688	2,625	232
N 506 BT(1).jpg	TRUE	1000	2835	2,314286	232
N 506 BT(2).jpg	TRUE	1000	2739	2,515152	232
N 506 BT(3).jpg	TRUE	1000	1900	3,04	202
N 506 BT(4).jpg	TRUE	1000	3115	2,542857	248
N 506 BT(5).jpg	TRUE	1000	2607	2,393939	224
N 598 GK(1).jpg	TRUE	1000	2370	2,633333	218

N 598 GK(2).jpg	TRUE	1000	1898	2,807692	198
N 598 GK(3).jpg	FALSE	1000	1800	2,88	194
N 598 GK(4).jpg	TRUE	1000	2146	2,551724	206
N 598 GK(5).jpg	FALSE	1000	2370	2,633333	218
N 682 C(1).jpg	TRUE	1000	3094	2,676471	250
N 682 C(2).jpg	TRUE	1000	3384	2,611111	260
N 682 C(3).jpg	TRUE	1000	3276	2,527778	254
N 682 C(4).jpg	FALSE	1000	3300	3,030303	266
N 682 C(5).jpg	TRUE	1000	2940	3,266667	256
W 1406 AC(1).jpg	TRUE	1000	2528	2,46875	222
W 1406 AC(2).jpg	TRUE	1000	2480	2,580645	222
W 1406 AC(3).jpg	FALSE	1000	2480	2,580645	222
W 1406 AC(4).jpg	TRUE	1000	2624	2,5625	228
W 1406 AC(5).jpg	TRUE	1000	2325	2,419355	212

Lampiran 2. data klasifikasi plat nomor kelas 1

NamaCitra	Ao	Ratio	EdgeDensity
IMG_0686.JPG	1944	2,666666667	198
IMG_0730_2.JPG	1518	3,136363636	182
IMG_0785.JPG	2490	2,766666667	226
IMG_0806.JPG	2460	2,733333333	224
IMG_6938.JPG	3496	2,421052632	260
IMG_6942.JPG	2937	2,696969697	244
IMG_6944.JPG	1975	3,16	208
IMG_6956.JPG	2460	2,733333333	224
IMG_6963.JPG	1924	2,846153846	200
IMG_6970.JPG	2296	2,928571429	220

Lampiran 3. data uji non plat

NamaCitra	Ao	Ratio	EdgeDensity
IMG_0686.JPG	3200	2	240

IMG_6963.JPG	1250	2	150
IMG_6970.JPG	4070	2,972972973	294
IMG_6963.JPG	2679	1,212765957	208
IMG_0704_2.JPG	2790	2,903225806	242
IMG_0704_2.JPG	1325	2,12	156
IMG_0705_2.JPG	1407	3,19047619	176
IMG_0705_2.JPG	2044	2,607142857	202
IMG_0707.JPG	2418	2,516129032	218
Rata-rata	2353,666667	2,391412535	209,5555556
StDev	950,7072367	0,619479815	45,8015041

Lampiran Citra yang diujikan dalam skripsi dapat dilihat dalam CD skripsi + aplikasi lengkap



UNIVERSITAS BRAWIJAYA

