

# REKOGNISI PELAT KENDARAAN BERMOTOR DALAM KEADAAN BERGERAK



## TUGAS AKHIR

*Disusun dalam rangka memenuhi salah satu persyaratan*

*Untuk menyelesaikan program Strata-1 Departemen Teknik Informatika*

*Fakultas Teknik Universitas Hasanuddin*

*Makassar*

**Disusun Oleh :**

**RAHMIYANTI RUSLI**

**D421 14 315**

**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2019**



## LEMBAR PENGESAHAN SKRIPSI

**-REKOGNISI PELAT KENDARAAN BERMOTOR DALAM KEADAAN BERGERAK-**

**OLEH:**

**RAHMIYANTI RUSLI**

**NIM D42114315**

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 21 Januari 2019.  
Diterima dan disahkan sebagai salah satu syarat memperoleh gelar Sarjana Teknik (ST.)  
pada Program Strata-1 Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Gowa, 1 Maret 2019

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Dr. Indrabayu, S.T., M.T., M.Bus.Sys.  
NIP 19750716 200212 1 004

Dr. Eng. Intan Sari Areni, S.T., M.T.  
NIP 19750203 200012 2 002

Diterima dan disahkan oleh:  
Ketua Departemen Teknik Informatika



Dr. Amif Ahmad Ilham, S.T., M.IT.  
NIP: 19731010 199802 1 001



## ABSTRAK

Pengendara bermotor yang melewati batas kecepatan kendaraan di atas 40 km/jam dapat mengganggu pengguna jalan lain bahkan dapat mengakibatkan kecelakaan lalu lintas. Nomor pelat kendaraan dapat menjadi salah satu bukti dari pelanggaran lalu lintas, dikarenakan dari pelat kendaraan dapat diperoleh informasi dari kepemilikan kendaraan tersebut.

Dari kasus ini, maka dikembangkan sistem yang mampu merekognisi pelat nomor kendaraan pada jalan jendral sudirman dengan menggunakan data video lalu lintas dengan resolusi 1920 x 1080 *pixel* yang diambil dari arah depan kendaraan dan menggunakan perangkat kamera statis yang dipasang pada jembatan penyeberangan dengan sudut kamera 60°.

Dalam penelitian ini, metode yang digunakan yaitu algoritma *Kalman Filter* untuk *tracking* kendaraan, *Cascade Classifier* dengan deskripsi LBP untuk mendeteksi lokasi pelat dan metode OCR untuk merekognisi karakter pelat dengan membandingkan 2 metode *pre processing* yaitu *Top Hat Transform* dan *Bottom Hat Transform* dengan menggunakan matlab 2016a. Dengan metode yang digunakan dapat merekognisi pelat selain pelat modifikasi dan pelat yang tidak simetris. Hasil dari penelitian menunjukkan bahwa tingkat akurasi pendeteksian dan rekognisi karakter pelat kendaraan bermotor dengan menggunakan metode *Top Hat Transform* yaitu sebesar 83%, menggunakan metode *Bottom Hat Transform* sebesar 50%, dan menggunakan metode *Top Bottom Hat Transform* sebesar 70%.

**Kata Kunci :** Nomor Pelat Kendaraan, Rekognisi Pelat, *Kalman Filter*, *Cascade Classifier*, LBP, *Top Hat Transform*, *Bottom Hat Transform*, OCR.



## KATA PENGANTAR

Assalamu Alaikum Wr. Wb.

Segala puji dan syukur atas kehadiran Allah SWT yang telah melimpahkan Rahmat dan Karunia-Nya sehingga Tugas Akhir dengan judul “**REKOGNISI PELAT KENDARAAN BERMOTOR DALAM KEADAAN BERGERAK**” ini dapat diselesaikan sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Pada penyusunan kali ini disajikan hasil penelitian menyangkut judul yang telah diangkat dan telah melalui proses pencarian dari berbagai sumber baik jurnal penelitian, prosiding pada seminar-seminar nasional/internasional, buku maupun dari situs-situs di internet, selain dari hasil-hasil penelitian sebelumnya.

Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan Tugas Akhir, sangatlah sulit untuk menyelesaikan Tugas Akhir ini. Oleh karena itu, penulis berterima kasih kepada:

- 1) Kedua Orang tua penulis, Bapak **Rusli** dan Ibu **Rahmatang** serta saudara-saudara penulis kakak **Rahmawati Rusli**, kakak **Ridwan Rusli**, kakak **Rustan Rusli**, dan adik **Ramla Melani Putri** yang selalu memberikan dukungan, doa, dan semangat;
- 2) Bapak **Dr. Indrabayu ST., MT., M.Bus.Sys.**, selaku pembimbing 1 dan Ibu **Dr.Eng. Intan Sari Areni, ST, MT.**, selaku pembimbing II yang selalu

menyediakan waktu, tenaga, pikiran dan perhatian yang luar biasa untuk mengarahkan penulis dalam penyusunan Tugas Akhir;



- 3) Bapak **Dr. Amil Ahmad Ilham, ST., M.IT., Ph.D** selaku Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas bimbingannya selama masa perkuliahan penulis;
- 4) Terkhusus **Ilham Dwi Putra** yang selalu memberikan kasih sayang tiada hentinya dan selalu memberikan dorongan serta motivasi dalam penyusunan Tugas Akhir ini;
- 5) Para Sahabat, partner tim ITS, teman-teman dan kakak-kakak AIMP *Research Group* Unhas yang telah memberikan begitu banyak bantuan selama penelitian, pengambilan data dan diskusi progress penyusunan Tugas Akhir;
- 6) Segenap Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu penulis;
- 7) Seluruh teman-teman **angkatan 2014** Departemen Teknik Informatika FT UH atas semua bantuan dan semangat yang diberikan selama ini;
- 8) Orang-orang berpengaruh lainnya yang tanpa sadar telah menjadi inspirasi penulis.

Akhir kata, penulis berharap semoga Allah SWT. berkenan membalas segala kebaikan dari semua pihak yang telah banyak membantu. Semoga Tugas Akhir ini dapat memberikan manfaat bagi pengembangan ilmu. Aamiin.

*Wassalam*

Makassar, 22 Desember 2018

Penulis



## DAFTAR ISI

HALAMAN JUDUL .....	i
LEMBAR PENGESAHAN .....	
ABSTRAK .....	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	viii
DAFTAR TABEL .....	x
BAB I.....	1
PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian .....	3
1.4 Manfaat Penelitian .....	3
1.5 Batasan Masalah .....	3
1.6 Sistematika Penulisan .....	4
BAB II .....	6
TINJAUAN PUSTAKA .....	6
2.1 Pengertian Pelat Nomor Kendaraan .....	6
2.2 Citra Digital .....	7
2.3 Pengolahan Citra Digital .....	8
4 Interpolasi .....	8
5 <i>Threshold</i> .....	9
6 Morfologi Citra.....	11



2.7 Analisis Blob .....	15
2.8 Local Binary Pattern .....	15
2.9 Optical Character Recognition .....	16
2.10 Tracking.....	19
2.11 Kalman Filter .....	19
2.12 Gassians Maxture Models.....	20
<b>BAB III.....</b>	<b>22</b>
<b>METODOLOGI PENELITIAN .....</b>	<b>22</b>
3.1 Tahapan Penelitian .....	22
3.2 Waktu dan Lokasi Penelitian.....	24
3.3 Instrumen Penelitian.....	25
3.4 Teknik Pengambilan Data.....	25
3.5 Perancangan Implementasi Sistem.....	27
3.5.1 Input Video.....	29
3.5.2 Deteksi Kendaraan .....	31
3.5.3 Multiobject Tracking .....	35
3.5.4 Deteksi Pelat Kendaraan .....	39
3.5.5 Pre Processing .....	41
3.5.6 Membagi 3 bagian Pelat.....	49
3.5.7 Rekognisi Teksi Karakter Pelat Kendaraan .....	51
3.5.8 Graphic User Interface (GUI) Sistem .....	53
3.6 Analisis Kerja Sistem.....	56
.....	57
DAN PEMBAHASAN .....	57
1 Hasil Penelitian.....	57



4.1.1 Hasil Operasi Data Citra .....	57
4.1.2 Hasil Pengujian Sistem.....	70
4.2 Pembahasan .....	94
<b>BAB V</b> .....	<b>97</b>
<b>PENUTUP</b> .....	<b>97</b>
5.1 Kesimpulan .....	97
5.2 Saran .....	97
<b>DAFTAR PUSTAKA</b> .....	<b>99</b>
<b>LAMPIRAN</b> .....	<b>103</b>





## DAFTAR GAMBAR

Gambar 2.1 Citra <i>Grayscale</i> 4 x 4 <i>Pixel</i> .....	10
Gambar 2.2 Citra Biner 4 x 4 <i>Pixel</i> .....	10
Gambar 2.3 Proses Dilasi dengan SE Berukuran 3 x 3 .....	12
Gambar 2.4 Proses Erosi dengan SE Berukuran 3 x 3.....	14
Gambar 2.5 Analisis Luasan Blob Sebuah Objek Kendaraan .....	15
Gambar 2.6 Contoh Implementasi LBP .....	16
Gambar 2.7 Gambar Alur Kerja OCR.....	17
Gambar 3.1 Diagram Tahapan Penelitian.....	22
Gambar 3.2 Lokasi Pemantauan dan Pengambilan Data .....	24
Gambar 3.3 Ilustrasi Pengambilan Data .....	25
Gambar 3.4 Deskripsi Posisi Kamera dan Jalan yang Dideteksi .....	26
Gambar 3.5 Flowchart Perancangan Sistem .....	28
Gambar 3.6 Frame RGB .....	30
Gambar 3.7 Contoh Sampel Positif .....	30
Gambar 3.8 Contoh Sampel Negatif .....	31
Gambar 3.9 Blok Diagram Tahapan Deteksi Kendaraan.....	31
Gambar 3.10 Citra Hasil Konversi ke <i>Grayscale</i> .....	32
Gambar 3.11 Hasil Inisialisasi <i>Foreground</i> dan <i>Background</i> .....	34
Gambar 3.12 Citra dengan Kendaraan Terdeteksi .....	35
Gambar 3.13 Contoh ROI Sampel Positif .....	39
3.14 Blok Diagram Tahapan <i>Pre Processing</i> .....	41
3.15 Citra Kandidat Pelat .....	42
3.16 Citra Hasil Konversi RGB ke <i>Grayscale</i> .....	43



Gambar 3.17 Citra Hasil <i>Contrast Enhancement</i> .....	43
Gambar 3.18 Citra Hasil Konversi <i>Grayscale</i> ke <i>Binary</i> .....	44
Gambar 3.19 Citra Setelah <i>Top Hat Transform</i> .....	45
Gambar 3.20 Citra Hasil <i>Bottom Hat Transform</i> .....	46
Gambar 3.21 Citra Hasil <i>Top Hat</i> dan <i>Bottom Hat</i> .....	47
Gambar 3.22 Citra Hasil Olah ROI .....	48
Gambar 3.23 Citra Pelat Hasil <i>Pre Processing</i> .....	49
Gambar 3.24 Bagian Pertama Pelat .....	50
Gambar 3.25 Bagian Kedua Pelat .....	50
Gambar 3.26 Bagian Ketiga Pelat .....	50
Gambar 3.27 <i>Indonesian License Plate Font</i> .....	52
Gambar 3.28 Tampilan Fitur <i>OcrTrainer</i> .....	52
Gambar 3.29 Hasil Visualisasi Sistem Berupa <i>Output Pelat</i> .....	53
Gambar 3.30 Tampilan <i>List File Video</i> , Tombol <i>Play</i> .....	54
Gambar 3.31 Tampilan <i>Frame per Frame</i> dan Pelat yang <i>Tercapture</i> .....	55
Gambar 4.1 Konversi Citra.....	57
Gambar 4.2 Citra Hasil Inisialisasi <i>Foreground</i> dan <i>Background</i> .....	58
Gambar 4.3 Citra Hasil Morfologi .....	59
Gambar 4.4 Citra Hasil Deteksi.....	60
Gambar 4.5 Hasil dari Detektor Pelat Menggunakan LBP .....	61
Gambar 4.6 <i>Cropping</i> Citra.....	62
Gambar 4.7 <i>Grayscale</i> Citra Pelat.....	62
4.8 Proses Penyalinan Nilai Tiap Pixel pada Proses <i>Resize</i> .....	63
4.9 <i>Resize</i> Citra Pelat .....	64
4.10 Konversi Citra ke Biner .....	64



<b>Gambar 4.11 Citra Hasil <i>Top Hat Transform</i> .....</b>	<b>65</b>
<b>Gambar 4.12 Citra Hasil <i>Bottom Hat Transform</i> .....</b>	<b>66</b>
<b>Gambar 4.13 Hasil Citra <i>Top Bottom Transform</i> .....</b>	<b>66</b>
<b>Gambar 4.14 Citra Hasil Olah ROI &gt; 800 pixel.....</b>	<b>67</b>
<b>Gambar 4.15 ROI dalam Citra .....</b>	<b>68</b>
<b>Gambar 4.16 Citra Hasil <i>Masking Region Of Interest (ROI)</i>.....</b>	<b>68</b>
<b>Gambar 4.17 Citra Hasil Bagi 3 Bagian .....</b>	<b>69</b>
<b>Gambar 4.18 <i>Indonesian License Plate Font</i>.....</b>	<b>70</b>
<b>Gambar 4.19 Hasil <i>Output</i> dari Sistem .....</b>	<b>70</b>
<b>Gambar 4.20 Pelat Kendaraan yang Gagal Direkognisi dengan Baik.....</b>	<b>95</b>



## DAFTAR TABEL

<b>Tabel 3.1 Pengaturan <i>Property</i> pada <i>Blob Analysis</i>. .....</b>	<b>34</b>
<b>Tabel 4.1 Perbandingan Menggunakan <i>Kalman Filter</i> dan tidak Menggunakan <i>Kalman Filter</i> .....</b>	<b>71</b>
<b>Tabel 4.2 Hasil Pengukuran Akurasi Rekognisi Pelat Kendaraan Bermotor Menggunakan Metode <i>Top Hat Transform</i>.....</b>	<b>74</b>
<b>Tabel 4.3 Hasil Pengukuran Akurasi Rekognisi Pelat Kendaraan Bermotor Menggunakan Metode <i>Bottom Hat Transform</i> .....</b>	<b>81</b>
<b>Tabel 4.4 Hasil Pengukuran Akurasi Rekognisi Pelat Kendaraan Bermotor Menggunakan Metode <i>Top Bottom Hat Transform</i>.....</b>	<b>87</b>



# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Salah satu karakteristik dari kendaraan yaitu pelat nomor kendaraan. Pelat nomor kendaraan merupakan identitas utama dari sebuah kendaraan bermotor yang layak jalan/beroperasi di jalan raya, sehingga pelat nomor polisi umumnya diletakkan pada posisi yang mudah dilihat. Nomor polisi dari pelat kendaraan pula dapat diketahui informasi kepemilikan dari kendaraan tersebut.

Data dari Badan Pusat Statistik tahun 2017 yang bersumber dari Kantor Kepolisian Republik Indonesia menuliskan bahwa jumlah sepeda motor untuk wilayah Indonesia yaitu sebanyak 98,88 Juta atau 81,5% dari total jumlah kendaraan yang ada di Indonesia. Sedangkan untuk mobil 22,48 juta atau sekitar 18,55% dari total keseluruhan. Untuk wilayah Sulawesi Selatan berdasarkan data dari Badan Pusat Statistik tahun 2017 menuliskan jumlah sepeda motor yaitu 2.701.739 atau 85,3% dari jumlah kendaraan yang ada di wilayah Sulawesi Selatan, sedangkan sisanya yaitu sebesar 464.340 atau sekitar 14,7% jumlah mobil di Sulawesi Selatan (Badan Pusat Statistik, 2017).

Di kota-kota besar seperti Makassar, banyak jalan-jalan utama yang padat kendaraan seperti pada jalan A.P Pettarani dan jalan Jend. Sudirman. Dengan kepadatan lalu lintas, sering kali terjadi pelanggaran lalu lintas di mana pengendara yang ugal-ugalan melewati batas kecepatan yang telah ditentukan.

Ini pun menjadi masalah yang sangat serius ketika kurangnya bukti



pelanggaran lalu lintas sehingga pengendara yang melakukan pelanggaran tidak dapat ditindaklanjuti.

Berdasarkan peraturan Menteri no. 111 tahun 2015 tentang batas kecepatan kendaraan, kecepatan kendaraan pada daerah pemukiman yaitu 30km/jam, kemudian batas kecepatan kendaraan di jalanan perkotaan yaitu 50km/jam, dan batas kecepatan kendaraan pada jalan antar kota yaitu 80km/jam (Ppid Dephub, 2015).

Berdasarkan data dari Humas Polda Sulsel pada tanggal 30 November 2017 tercatat selama 2 minggu pelaksanaan operasi zebra di sebuah lokasi ditemukan 504 pelanggaran lalu lintas yaitu 430 pengendara sepeda motor dan sisanya 74 pengendara mobil (News Rakyatku, 2017).

Nomor polisi yang tercantum pada pelat kendaraan bisa digunakan oleh pihak berwajib sebagai barang bukti pengendara melakukan pelanggaran lalu lintas. Jumlah kendaraan yang banyak mengakibatkan aparat kepolisian sulit untuk mengenali pelat nomor kendaraan secara manual, sehingga dibutuhkan suatu sistem yang dapat mengenali pelat nomor kendaraan secara otomatis.

Dengan perkembangan teknologi informasi, hal tersebut dapat ditangani dengan sistem yang terhubung dengan beberapa kamera di beberapa titik. Kamera berfungsi untuk merekam dan memantau aktifitas pengendara, dan dapat disediakan sebuah komputer untuk menganalisa dan mengolah data video yang dihasilkan oleh kamera tersebut. Pengolahan video tersebut

manfaat untuk mengetahui pelat kendaraan yang melewati ruas jalan  
tu. Kamera yang digunakan dapat berupa static kamera yang memiliki



kualitas video yang cukup baik, serta sudut pengambilan gambar juga harus tepat, supaya mendapatkan gambar yang optimal.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang, maka rumusan masalah pada tugas akhir ini adalah:

1. Bagaimana cara mendeteksi pelat kendaraan bermotor dalam keadaan bergerak?
2. Bagaimana analisis akurasi sistem pendeteksian pelat dari arah depan kendaraan?

## 1.3 Tujuan Penelitian

Tujuan dari tugas akhir ini adalah :

1. Untuk mengetahui cara mendeteksi pelat kendaraan bermotor dalam keadaan bergerak.
2. Untuk mengetahui analisis akurasi sistem pendeteksian pelat dari arah depan kendaraan.

## 1.4 Manfaat Penelitian

Manfaat dari tugas akhir ini adalah :

### a) **Bagi Masyarakat**

Mengurangi tindak pelanggaran lalu lintas karena sistem akan mudah mengenali pelanggar lalu lintas.

### b) **Bagi Dinas Perhubungan dan Korps Lalu Lintas**

Memantau pelanggaran lalu lintas



### c) Bagi Instansi Penelitian

Menjadi salah satu sistem yang dapat dikembangkan di Sulawesi Selatan.

## 1.5 Batasan Masalah Penelitian

Yang menjadi batasan masalah dalam tugas akhir ini adalah :

1. Font pelat standar.
2. Dudukan pelat dari sudut  $105^\circ$  sampai  $135^\circ$ .
3. Data yang diolah berupa data video dengan resolusi  $1920 \times 1080$  *pixel* dengan format *.mov*.
4. Data video diambil menggunakan *static camera* dengan sudut kamera  $60^\circ$  dan ketinggian 8,75 meter.
5. Waktu pengambilan data dilakukan pada siang hari.
6. Pengambilan video dilakukan dari arah depan kendaraan.
7. Kondisi pelat yang direkognisi tampil secara keseluruhan dalam *frame* (tidak terpotong karena objek lain atau tidak terpotong karena sebagian objek pelat keluar dari *frame*).
8. Posisi pelat simetris.
9. Tidak terdapat noise pada pelat.

## 1.6 Sistematika Penulisan



Untuk memberikan gambaran singkat mengenai isi tulisan secara ringkas, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu :



## **BAB I PENDAHULUAN**

Bab ini menguraikan secara umum mengenai hal yang menyangkut latar belakang, perumusan masalah dan batasan masalah, tujuan, manfaat, dan sistematika penulisan.

## **BAB II TINJAUAN PUSTAKA**

Bab ini berisi teori-teori tentang hal-hal yang berhubungan dengan Pengolahan Citra, Pemrosesan Citra dan metode yang digunakan.

## **BAB III METODOLOGI PENELITIAN**

Bab ini berisi tentang perencanaan dan penerapan algoritma serta teknik pengolahan data.

## **BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil pengolahan data serta pembahasan yang disertai tabel hasil penelitian.

## **BAB V PENUTUP**

Bab ini berisi tentang kesimpulan yang didapatkan berdasarkan hasil penelitian yang telah dilakukan serta saran-saran untuk pengembangan lebih lanjut.



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Pengertian Pelat Nomor Kendaraan**

Tanda Nomor Kendaraan Bermotor (TNKB) atau yang biasa disebut dengan pelat nomor polisi adalah pelat terbuat dari alumunium yang menunjukkan tanda kendaraan bermotor di Indonesia yang telah didaftarkan pada Kantor Samsat. Samsat atau Sistem Administrasi Manunggal Satu Atap merupakan suatu sistem administrasi terpadu antara Polri, Dinas Pendapatan Provinsi dan PT Jasa Raharja yang memberikan pelayanan pencatatan kendaraan bermotor. Tanda nomor kendaraan bermotor berbentuk plat alumunium terdiri dari 2 (dua) baris. Baris pertama menunjukkan kode wilayah (huruf), nomor polisi (angka), dan kode/seri akhir wilayah (huruf), sementara baris kedua menunjukkan bulan dan tahun masa berlaku. Nomor polisi biasanya diberikan sesuai dengan urutan pendaftaran kendaraan. Ada beberapa warna tanda kendaraan bermotor berdasarkan sifat dan kepemilikan kendaraannya tersebut (Helmy Fitriawan, 2014).

Indonesia memiliki karakteristik pelat nomor tersendiri. Biasanya, format pada pelat nomor di Indonesia adalah **HH AAAA HH**, dimana **H** merepresentasikan huruf dan **A** merepresentasikan angka 0-9. Satu atau dua huruf pertama merepresentasikan area dimana kendaraan tersebut didaftarkan.

diikuti dengan satu sampai empat angka, kemudian diakhiri oleh satu huruf. Misal, **BM 6015 TV** merupakan kendaraan dari Pekanbaru, karena



dua huruf pertamanya, **BM**, merupakan kode untuk Pekanbaru. Sejak 2008, daerah Jakarta dan sekitarnya memperkenalkan format baru untuk pelat mobil, yaitu **HH AAAA HHH**. Sebelumnya, format ini telah diperkenalkan terlebih dahulu untuk pelat motor (R. Aditya, 2016).

Terdapat beberapa kelompok pelat nomor yang dibedakan melalui warnanya, yaitu:

1. Warna putih pada latar hitam: Kendaraan pribadi
2. Warna merah pada latar putih: Kendaraan yang belum terdaftar atau kendaraan baru yang belum ada pemiliknya
3. Warna hitam pada latar kuning: Kendaraan untuk transportasi umum seperti bus, taksi, dan angkutan kota
4. Warna putih pada latar merah: Kendaraan institusi pemerintahan Militer, polisi, dan pemadam kebakaran memiliki warna sendiri, dan biasanya terdapat lambang pangkat pemilik kendaraan tersebut.

## 2.2 Citra Digital

Pengolahan citra digital merupakan metode yang digunakan untuk mengolah citra pada komputer sehingga dapat menghasilkan gambar sesuai dengan yang dibutuhkan. Misalnya terdapat sebuah citra digital berwarna dengan ukuran 1280x720 piksel. Dengan pengolahan citra digital, gambar tersebut dapat diubah ukurannya menjadi 640x360 tanpa mengurangi kualitas gambar. Citra

adalah fungsi kontinu dari intensitas cahaya dalam bidang dua dimensi. Secara matematis, fungsi ini dapat dilambangkan dengan  $f(x,y)$ , dimana  $(x,y)$  adalah koordinat pada bidang dua dimensi tersebut, dan  $f(x,y)$  merupakan



intensitas cahaya pada titik  $(x,y)$ . Ukuran terkecil dalam citra digital biasa disebut *picture element* atau *pixel* (Syawaluddin, 2016).

Citra bergerak (*moving images*) adalah rangkaian citra diam yang ditampilkan secara beruntun sehingga memberi kesan pada mata kita sebagai gambar yang bergerak. Setiap citra di dalam rangkaian itu disebut *frame*. Gambar-gambar yang tampak pada film layar lebar atau televisi pada hakikatnya terdiri dari ratusan hingga ribuan *frame*. Meskipun sebuah citra kaya informasi, namun seringkali citra yang kita miliki mengalami penurunan mutu (degradasi), misalnya mengandung cacat atau derau (*noise*), warnanya terlalu kontras, kurun tajam, kabur (*blurring*), dan sebagainya. Tentu saja citra semacam ini menjadi lebih sulit diinterpretasi karena informasi yang disampaikan oleh citra tersebut menjadi berkurang. Agar citra yang mengalami gangguan mudah diinterpretasikan (baik oleh manusia maupun mesin), maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik (Syawaluddin, 2016).

### 2.3 Pengolahan Citra Digital

Pengolahan citra digital merupakan metode yang digunakan untuk mengolah citra pada komputer sehingga dapat menghasilkan gambar sesuai dengan yang dibutuhkan. Misalnya terdapat sebuah citra digital berwarna dengan ukuran 1280x720 piksel. Dengan pengolahan citra digital, gambar tersebut dapat diubah ukurannya menjadi 640x360 piksel tanpa mengurangi kualitas gambar

(Syawaluddin, 2016).

Citra merupakan fungsi kontinu dari intensitas cahaya dalam bidang dua



dimensi. Secara matematis, fungsi ini dapat dilambangkan dengan  $f(x,y)$ , dimana  $(x,y)$  merupakan koordinat pada bidang dua dimensi tersebut, dan  $f(x,y)$  merupakan intensitas cahaya pada titik  $(x,y)$ . Ukuran terkecil dalam citra digital biasa disebut *picture element* atau *pixel* (Syawaluddin, 2016).

## 2.4 Interpolasi

Interpolasi itu sendiri merupakan kegiatan *resampling* sebuah citra untuk menentukan nilai-nilai antara piksel-piksel yang ditetapkan . Misalnya, pada proses *image resizing*, jumlah piksel pada citra baru dapat lebih banyak (jika ingin memperbesar gambar) atau lebih sedikit (jika memperkecil gambar). Nilai piksel pada gambar baru ini didapatkan dengan cara interpolasi. Pada MATLAB, terdapat tiga metode interpolasi yang dapat digunakan, yaitu (Syawaluddin, 2016):

1. Interpolasi tetangga piksel terdekat (*nearest neighbor interpolation*)
2. Interpolasi *bilinear*
3. Interpolasi *bicubic*

Secara fundamental, ketiga metode interpolasi tersebut bekerja dengan cara yang sama. Untuk menentukan nilai piksel baru, akan ditentukan suatu titik di citra lama yang berkorespondensi dengan titik di citra baru. Nilai piksel pada citra baru dapat ditentukan dengan menghitung rata-rata pembobotan dari kumpulan piksel pada area yang telah ditentukan. Perbedaan pada ketiga metode interpolasi tersebut adalah jumlah piksel yang diperhitungkan untuk mencari

ai pada suatu titik di citra baru tersebut, yaitu (Syawaluddin, 2016) :



1. Pada metode *nearest neighbor interpolation*, nilai piksel yang diperhitungkan adalah nilai piksel yang terdekat, dan nilai piksel lain tidak ditentukan.
2. Pada metode *bilinear*, nilai yang diperhitungkan adalah 4 nilai terdekat (2x2)
3. Pada metode *bicubic*, nilai yang diperhitungkan adalah 8 nilai terdekat (3x3) Semakin banyak piksel yang dihitung, makin lama waktu yang dibutuhkan, namun hasilnya akan lebih akurat.

### 2.5 Threshold

Metode ini menggunakan nilai ambang  $T$  sebagai patokan untuk memutuskan *pixel* diubah menjadi hitam atau putih. Biasanya  $T$  dihitung dengan persamaan 1 (T. Sutoyo, dkk, 2009).

$$T = \frac{f_{maks} + f_{min}}{2} \dots\dots\dots(1)$$

Di mana  $f_{maks}$  adalah nilai intensitas *pixel* maksimum pada citra dan  $f_{min}$  adalah nilai intensitas minimum *pixel* pada citra. Sebagai contoh citra grayscale 4 x 4 *pixel* dengan kedalaman 8 bit pada **gambar 2.1**:

200	230	150	75
240	50	170	90
210	100	120	80
100	90	200	230



**Gambar 2.1.** Citra *Grayscale* 4 x 4 *Pixel* 1 (T. Sutoyo, dkk, 2009)

Dengan metode ini, nilai *threshold*  $T$  adalah :

$$T = \frac{f_{maks} + f_{min}}{2} = \frac{240 + 50}{2} = 145 \dots \dots \dots (2)$$

Bila nilai  $T = 145$ , maka diterapkan untuk citra tersebut diperoleh pada **gambar 2.2:**

255	255	255	0
255	0	255	0
255	0	0	0
0	0	255	255

**Gambar 2.2.** Citra Biner 4 x 4 *pixel* 1 (T. Sutoyo, dkk, 2009)

## 2.6 Morfologi Citra

Salah satu operasi morfologi dalam citra *grayscale* adalah *opening*. Operasi *opening* pada sebuah citra  $f$  oleh *strel*  $b$  dinotasikan dengan  $f \circ b$  dan didefinisikan sebagai operasi erosi yang dilanjutkan dengan operasi dilasi, kedua operasi tersebut dilakukan secara berulang (Prasetyo, 2011).

Morfologi di dunia digital dapat diartikan sebuah cara untuk

tipsikan ataupun menganalisa bentuk dari objek digital. Morfologi dalam

tal adalah suatu *tool* untuk ekstraksi komponen *image* yang berguna dalam



representasi dan deskripsi dari bentuk daerah (*region shape*) dengan *structuring element* (SE) untuk menentukan *properties of interest* dari *image* (Putra, 2010).

Pada Morfologi, suatu citra dinyatakan dengan himpunan koordinat diskrit (kontinu). Dalam hal ini, himpunan tersebut berhubungan dengan *point* atau *pixel* objek pada citra. Karena objek dianggap sebagai suatu himpunan maka operasi-operasi himpunan seperti gabungan (*union*), irisan (*intersection*), dan komplemen (*complement*) dapat dilakukan (Putra, 2010).

Operasi morfologi menggunakan dua input himpunan yaitu suatu citra (pada umumnya citra biner) dan suatu kernel. Khusus dalam morfologi, istilah kernel biasa disebut *structuring elements* (elemen pembentuk struktur). *SE* merupakan suatu matrik yang pada umumnya berukuran kecil. Elemen dari *SE* dapat bernilai 1, 0, dan *don't care*. Nilai *don't care* biasanya ditandai dengan nilai elemen dikosongkan atau diberi tanda silang. Terdapat dua operasi dasar morfologi yaitu dilasi (*dilation*) dan erosi (*erosion*). Operasi-operasi ini menjadi dasar untuk membuat berbagai operasi morfologi yang sangat berguna untuk pengolahan citra digital, seperti *opening*, *closing*, *hit and miss transform*, *thinning*, dan *thickening* (Putra, 2010).

Jika suatu objek (citra input) dinyatakan dengan  $A$  dan  $SE$  dinyatakan dengan  $B$  serta  $Bx$  menyatakan translasi  $B$  sedemikian sehingga pusat  $B$  terletak pada  $x$ . Operasi dilasi  $A$  dan  $B$  dapat dinyatakan sebagai berikut.

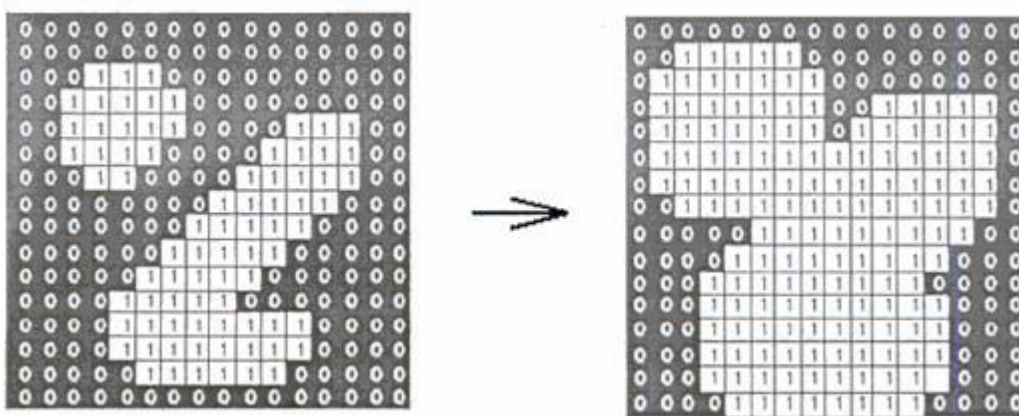
$$D(A, B) = A \oplus B = \{x: Bx \cap A \neq \emptyset\} \dots \dots \dots (3)$$

$\emptyset$  menyatakan himpunan kosong.





Dilasi (*dilation*) berguna untuk memperluas atau menebalkan objek pada *image* biner. Proses dilasi dilakukan dengan membandingkan setiap *pixel* citra input dengan nilai pusat *SE* dengan cara melapiskan (*superimpose*) *SE* dengan citra sehingga pusat *SE* tepat dengan posisi *pixel* citra yang diproses. Jika paling sedikit ada 1 *pixel* pada *SE* sama dengan nilai *pixel* objek (*foreground*) citra maka *pixel* input diset nilainya dengan nilai *pixel foreground* dan bila semua *pixel* yang berhubungan adalah *background* maka input *pixel* diberi nilai *pixel background* (Putra, 2010).



**Gambar 2.3.** Proses dilasi dengan *SE* berukuran  $3 \times 3$  dengan semua elemen *SE* bernilai 1 (Putra, 2010)

Semakin besar ukuran *SE* maka semakin besar perubahan yang terjadi. *SE* berukuran kecil juga dapat memberikan hasil yang sama dengan *SE* berukuran besar dengan cara melakukan dilasi berulang kali. Efek dilasi terhadap citra biner adalah memperbesar batas dari objek yang ada sehingga objek terlihat semakin besar dan

bag yang terdapat di tengah objek akan tampak mengecil (Putra, 2010).

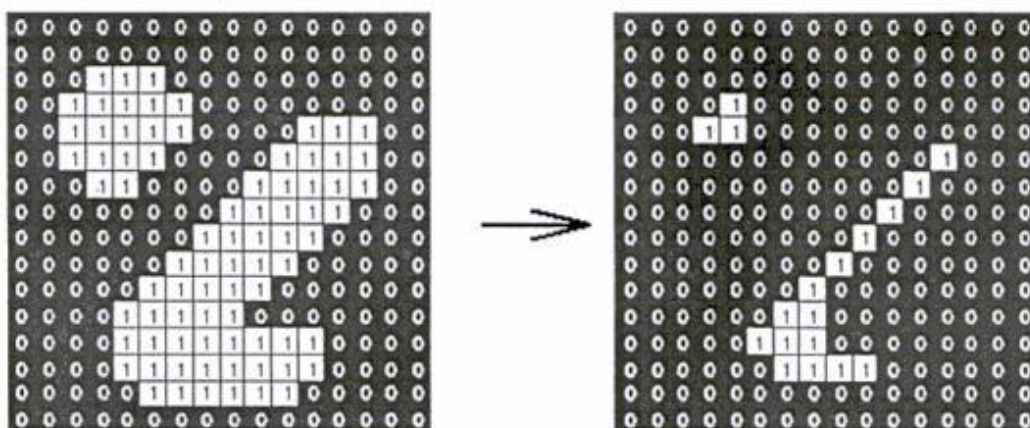
Matematis, operasi erosi dapat dinyatakan sebagai berikut.

$$A \ominus B = \{x: Bx \subset X\} \dots \dots \dots (4)$$



Sama seperti dilasi, proses erosi dilakukan dengan membandingkan setiap *pixel* citra input dengan nilai pusat *SE* dengan cara melapiskan *SE* dengan citra sehingga pusat *SE* tepat dengan posisi *pixel* citra yang diproses. Jika semua *pixel* pada *SE* tepat sama dengan semua nilai *pixel* objek (*foreground*) citra maka *pixel* input diset nilainya dengan nilai *pixel foreground*, jika tidak maka input *pixel* diberi nilai *pixel background*. Proses serupa dilanjutkan dengan menggerakkan *SE pixel* demi *pixel* pada citra input (Putra, 2010).

Proses erosi merupakan kebalikkan dari proses dilasi. Jika dalam proses dilasi menghasilkan objek yang lebih luas maka dalam proses erosi akan menghasilkan objek yang menyempit (mengecil). Lubang pada objek juga akan membesar seiring menyempitnya objek tersebut (Putra, 2010).



**Gambar 2.4.** Proses erosi dengan *SE* berukuran  $3 \times 3$  dengan semua elemen *SE* bernilai 1. (Putra, 2010)

Dari **gambar 2.4** terlihat hasil proses erosi menyebabkan objek mengecil.

semakin besar kernel yang digunakan maka hasil yang akan didapatkan akan semakin

kecil. Begitu pula juga apabila proses erosi dilakukan berulang-ulang akan



terus mengecilkan objek walaupun hanya menggunakan *SE* berukuran kecil (Putra, 2010).

Di samping itu operasi morfologi *grayscale*, terdapat juga algoritma dasar morfologi *grayscale* adalah *Top-Hat Transform* dan *Bottom Hat Transform*. Teknik dengan menggunakan *Top Hat Transform* dan *Bottom Hat Transform* telah banyak digunakan dalam penelitian *image processing*, khususnya pada penelitian yang memiliki pencahayaan tidak teratur. Teknik *Top Hat Transform* ini mengombinasikan pengurangan citra *grayscale* dengan *opening* sedangkan teknik *Bottom Hat Transform* mengombinasikan pengurangan citra *grayscale* dengan *closing* (Putra,2010).

a. *Top Hat Transform*

Transformasi *top hat* merupakan perbedaan antara citra masukan (citra *grayscale*) dan citra setelah mengalami operasi *opening* (Solomon dan Breckon, 2011). Secara matematis seperti berikut (Gonzalez dan Woods, 2002):

$$T_{hat}(f) = f - (f \circ b) \dots \dots \dots (5)$$

Pada rumus diatas, *f* adalah citra masukan (citra *grayscale*) dan *(f ∘ b)* adalah citra hasil operasi *opening*. Transformasi ini berguna untuk mendapatkan bentuk global suatu objek yang mempunyai intensitas yang bervariasi.

b. *Bottom Hat Transform*

Transformasi *bottom hat* didefinisikan sebagai berikut (Gonzalez dan Woods, 2002):

$$B_{hat}(f) = (f \circ b) - f \dots \dots \dots (6)$$



Transformasi *bottom hat* merupakan perbedaan antara citra setelah mengalami operasi *closing* dan citra masukan (citra *grayscale*). Pada rumus diatas,  $f$  adalah citra masukan (citra *grayscale*) dan  $(f \cdot b)$  adalah citra hasil operasi *closing*.

c. *Top Bottom Hat Transform*

Pada MATLAB, *top hat* dan *bottom hat* dapat digunakan bersama-sama untuk meningkatkan kontras dalam foto.

*Syntax*:

$J = \text{imsubtract}(\text{imadd}(I, \text{tophat}), \text{bottomhat});$

Maksud *syntax* di atas citra masukan ditambah dengan citra hasil transformasi *top hat*, kemudian dikurangi dengan citra hasil transformasi *bottom hat*.

Keterangan :

$I$  = citra masukan.

$SE$  = struktur elemen.

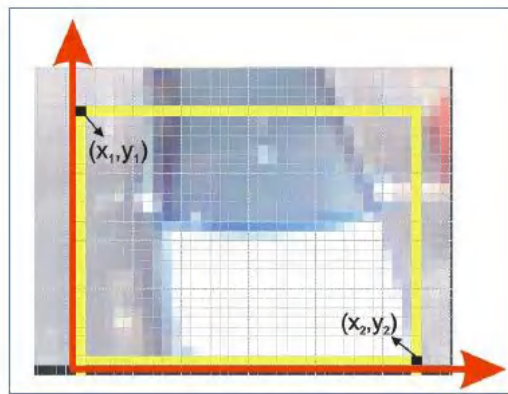
*imadd* digunakan untuk menambah dua gambar atau konstan gambar. *imsubtract* digunakan untuk mengurangi satu gambar atau konstan gambar.

## 2.7 Analisis Blob

Analisis blob merupakan teknik yang digunakan untuk menyatakan luas  $A$  dari suatu *image* yang menjadi fokus deteksi. Untuk menentukan nilai  $A$  beberapa hal yang harus diketahui untuk menghasilkan sebuah blob yang Penentuan luas blob pada setiap objek pada proses segmentasi *foreground*



perlu dianalisis karena nilai blob pada tiap objek akan berbeda. Hal ini dipengaruhi oleh fitur objek seperti ukuran, jenis, dan teknik pengambilan data video. Prosesnya dimulai dari penandaan area *foreground* yang dianggap objek, kemudian pengumpulan data area menjadi blob seperti posisi *pixel* awal, panjang terhadap sumbu  $x$  dan sumbu  $y$ , dan luas area *pixel* (Basri, 2015).



**Gambar 2.5.** Analisis Luasan Blob Sebuah Objek Kendaraan (Basri, 2015)

## 2.8 Local Binary Pattern (LBP)

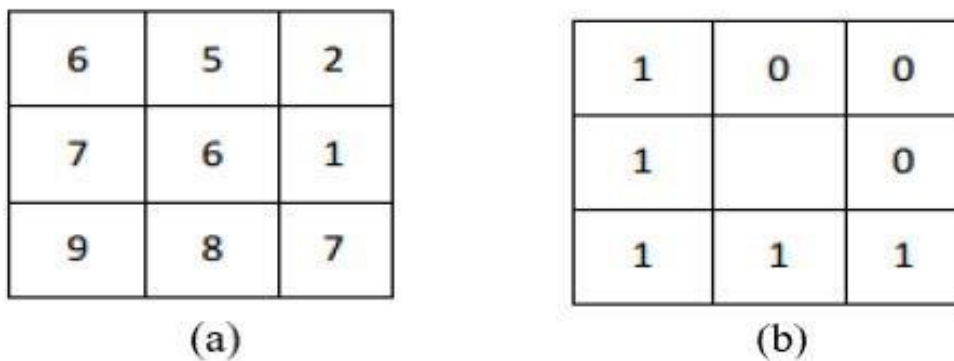
*Local Binary Pattern (LBP)* merupakan suatu operasi gambar yang mentransformasikan sebuah citra menjadi sebuah susunan label *integer* yang menggambarkan kenampakan skala kecil dari suatu citra. Label atau statistik tersebut, biasanya merupakan sebuah histogram, kemudian digunakan lagi untuk analisis citra yang lebih lanjut (Afrizal Firdaus, 2016).

Tahapan dalam mendapatkan nilai *LBP* adalah dengan mendapatkan nilai transformasi biner yang akan disusun menjadi desimal. Jika intensitas tetangga

bandingkan lebih besar atau sama dengan intensitas pusatnya maka nilai transformasi binernya adalah 1. Sebaliknya, jika intensitas tetangga yang



dibandingkannya lebih kecil daripada intensitas pusatnya maka nilai transformasi binernya adalah 0. Setelah ke-8 tetangganya dibandingkan, nilai transformasi tersebut akan disusun dan dikonversikan ke dalam bentuk desimal (Afrizal Firdaus, 2016).



**Gambar 2.6.** Contoh Implementasi *LBP* (a) Nilai Pixel pada Citra, (b) Nilai *LBP* dari Citra (a) (Afrizal Firdaus, 2016)

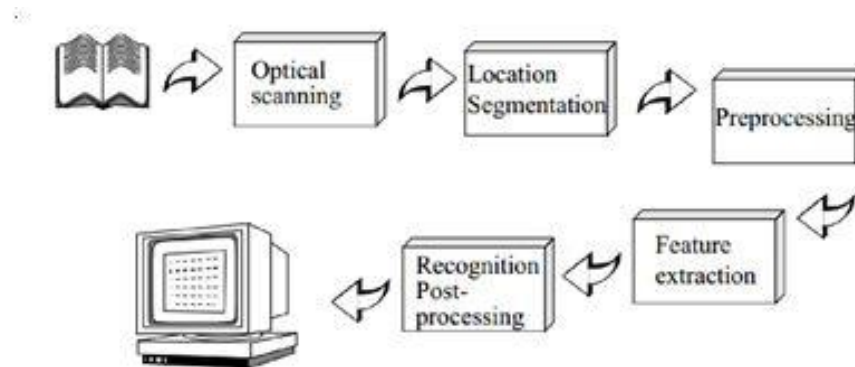
## 2.9 Optical Character Recognition

*Optical Character Recognition (OCR)* adalah sebuah sistem yang berfungsi untuk men-*scan* gambar pada sebuah citra yang memiliki konten di dalamnya akan diubah menjadi format teks. *OCR* dapat memudahkan untuk melakukan manipulasi terhadap citra yang bertulisan tangan, tulisan mesin ketik atau teks komputer (Devi Alfiani, 2017).

Teknologi *OCR* telah digunakan secara intensif sejak tahun 1970 oleh sebuah perusahaan di Dallas, Texas, yang bernama *Recognition Equipment, Inc.*, perusahaan tersebut membuat sistem berkecepatan tinggi untuk membaca tanda

atau kredit dari pembelian bahan bakar (Devi Alfiani, 2017).





**Gambar 2.7.** Gambar Alur Kerja *OCR* (Devi Alfiani, 2017)

Pada **gambar 2.7** dapat dilihat bahwa untuk membuat sistem *OCR* dibutuhkan tahapan-tahapan yang harus dilewati terlebih dahulu, Tahap pertama adalah *optical scanning* yaitu untuk menjadikan dokumen asli menjadi dokumen digital. Tahap kedua adalah *location segmentation* yaitu mencari daerah citra yang memiliki obyek berupa teks yang akan dibaca. Tahap ketiga adalah *preprocessing*, pada tahap ini akan dilakukan perbaikan terhadap citra untuk menghilangkan kecacatan. Tahap keempat adalah *feature extraction*, terdapat beberapa teknik yang digunakan antara lain *template matching*, *transformation*, *distribution of point zoning*, *moments*, *n-tupe*, *crossing*, dan *structural feature*. Tahap akhir adalah *recognition postprocessing* dimana akan dilakukan klasifikasi sehingga menghasilkan keluaran berupa teks dalam bentuk digital (Devi Alfiani, 2017).

*OCR* selalu difungsikan sebagai alternatif penyimpanan untuk kunci utama, teks dalam bentuk digital. Sebagai definisi, *OCR* adalah metode pemasukan data

puter yang digunakan adalah teknologi *scan* dan analisa gambar. Untuk si atau pembaca karakter dalam bentuk teks sebagai kunci atau *keyword*. de tersebut juga dilihat bentuk digital jika ingin menyiapkan dokumen





yang akan diubah bentuknya menjadi bentuk digital. Terlebih dahulu dokumen itu di-*scan* dan dicetak dahulu agar tidak terjadi kesalahan (Devi Alfiani, 2017).

Kecepatan *scan* adalah satu detik perhalaman atau lima belas detik perhalaman atau lebih cepat lagi tergantung dari teknologi *scan* yang ada dan tipe atau *merk scan* itu sendiri. Kinerja *OCR software* dapat diproses 7000 atau lebih dari ukuran *letter*, spasi ganda halaman perjam dengan 1000 *time* lebih cepat memasukkan data dengan operator yang siap dan pengalaman (Devi Alfiani, 2017).

Salah satu contoh dari *OCR* adalah *Digital Image Processing*, yaitu penggunaan algoritma komputer untuk membuat, memproses, komunikasi dan menampilkan gambar digital. *Digital Image Processing* dapat digunakan untuk:

- a. Mengkonversi sinyal dari sensor gambar ke dalam gambar digital.
- b. Meningkatkan kejernihan, menghilangkan *noise* dan benda lainnya.
- c. Mengekstrak ukuran, skala, atau jumlah objek dari suatu suatu kejadian.
- d. Mempersiapkan gambar untuk menampilkan atau mencetak.

## 2.10 Tracking

Proses mencari objek bergerak dalam urutan *frame* yang dikenal sebagai pelacakan (*tracking*). Pelacakan ini dapat dilakukan dengan menggunakan ekstraksi ciri benda dan mendeteksi objek/benda bergerak di urutan *frame*. Di dalam *computer vision*, *object tracking* adalah sebuah proses yang bertujuan untuk mengikuti pergerakan suatu objek. Dengan adanya *tracking*, nantinya dapat

ng kendaraan yang terdeteksi.





## 2.11 Kalman Filter

Pengertian umum dari *Kalman Filter* adalah solusi rekursif yang menggunakan teknik asimilasi data. Asimilasi data adalah salah satu teknik estimasi yang banyak digunakan untuk menyelesaikan berbagai masalah sistem dinamik (Maryam, 2016).

*Kalman Filter* merupakan suatu algoritma yang menggabungkan model dan pengukuran. Data pengukuran terbaru menjadi bagian penting dari algoritma *Kalman Filter* karena data terakhir akan mengoreksi hasil prediksi pengukuran, sehingga hasil estimasi selalu mendekati kondisi yang sebenarnya (Maryam, 2016).

*Kalman Filter* mengestimasi proses dengan menggunakan kontrol umpan balik dalam bentuk pengukuran *noise*. Dengan demikian, persamaan untuk *kalman filter* dibagi menjadi dua kelompok: persamaan *update* waktu dan persamaan *update* pengukuran. Persamaan *update* waktu bertanggung jawab untuk memproyeksikan kedepan (waktu). keadaan saat ini dan estimasi kovarian *error* mengestimasi untuk mendapatkan sebuah estimasi apriori untuk langkah waktu berikutnya. Persamaan *update* pengukuran bertanggung jawab atas umpan balik untuk menggabungkan pengukuran baru ke dalam estimasi apriori untuk mendapatkan estimasi aposteriori yang lebih baik (Maryam, 2016).

*Kalman Filter* terdiri dari dua tahap yaitu :

### 1. Tahap prediksi

Tahap prediksi dipengaruhi oleh dinamika sistem dengan memprediksi keadaan dengan menggunakan persamaan estimasi variabel keadaan dan kurusnya dihitung menggunakan kovarian *error* (Maryam, 2016).



## 2. Tahap koreksi

Pada tahap koreksi hasil estimasi variabel keadaan yang diperoleh pada tahap prediksi dikoreksi menggunakan model pengukuran. Salah satu bagian dari tahap ini yaitu matriks *Kalman Gain* yang digunakan untuk meminimumkan kovarian *error* (Maryam, 2016).

### 2.12 *Gaussians Mixture Model (GMM)*

GMM merupakan fungsi *density* probabilitas parametrik yang direpresentasikan sebagai jumlah komponen *gaussian*. GMM umumnya digunakan sebagai model parametrik dari distribusi probabilitas suatu pengukuran kontinyu atau fitur dari suatu sistem biometrik, seperti pelacakan berdasarkan warna suatu objek dalam video (Santosh et al., 2013), sedangkan pada citra RGB digunakan nilai vektornya. Penggunaan model yang lebih banyak pada setiap piksel akan menyebabkan proses *background extraction* lebih bersifat adaptif karena lebih banyak komponen warna yang dapat dimodelkan dalam setiap pikselnya. Jika suatu piksel tidak cocok dengan semua model distribusi *Gaussian*, maka model *Gaussian* yang memiliki probabilitas terkecil akan dihapus dan digantikan dengan model *Gaussian* untuk warna piksel yang baru. Langkah selanjutnya adalah menentukan piksel mana yang termasuk dalam objek *background* dan *foreground*, maka dilakukan seleksi. Jika warna piksel masuk dalam kategori salah satu kandidat *background*, maka piksel tersebut akan dianggap sebagai *background* (piksel diberi

warna hitam). Di luar itu, piksel yang tidak termasuk dalam kategori *background* akan dianggap sebagai *foreground* (piksel diberi nilai 1/putih) (Maryam et al., 2015).



*Background* adalah objek yang tidak bergerak di dalam *frame* video. Sedangkan *foreground* adalah objek bergerak dalam *frame* video. Objek yang terdeteksi sebagai *foreground* ditandai dengan blob yang merupakan sekumpulan piksel yang memiliki hubungan tetangga. Akurasi sistem deteksi bisa ditingkatkan dengan melakukan optimalisasi parameter blob dan penentuan *Region of Interest* (*ROI*) objek yang dideteksi (Indrabayu dkk., 2015).

### 2.13 State of The Art

Beberapa penelitian terkait yang membahas tentang deteksi pelat telah dilakukan dari tahun ke tahun. Penelitian-penelitian sebelumnya merekognisi pelat pada bagian belakang kendaraan dengan menggunakan data video dengan berbagai skenario.

**Tabel 2.1.** *State of The Art*

Penelitian yang telah Dilakukan				Penelitian saat ini
<b>Judul</b>	Deteksi Nomor Pelat Kendaraan Bergerak Menggunakan <i>Metode Local Binary Pattern</i> dan <i>Optical Character Recognition</i>	<i>Optical Character Recognition</i> untuk Deteksi Pelat Mobil dan Motor Kendaraan pada Kampus Teknik Gowa	Rekognisi Pelat Mobil yang Bergerak pada Kondisi Hujan Malam Hari dengan Teknik <i>Top Hat Transform</i>	Rekognisi Pelat Kendaraan Bermotor dalam Keadaan Bergerak
<b>Tahun</b>	2017	2018	2018	2019



Penelitian yang telah Dilakukan					Penelitian saat ini
Skenario Pengambilan Data	Objek	Mobil dan Motor	Mobil dan Motor	Mobil	Motor
	Tinggi Kamera	3,5 meter	2,3 meter; 2,5 meter; 3 meter; 4 meter	8,4 meter	8,75 meter
	Arah Kendaraan	Belakang Kendaraan	Belakang Kendaraan	Belakang Kendaraan	Depan Kendaraan
Metode	Deteksi Kendaraan	-	<i>Gaussians Mixture Models</i>	<i>Gaussians Mixture Models</i>	<i>Gaussians Mixture Models dan Kalman Filter</i>
	Deteksi Lokasi Pelat	<i>Local Binary Patern</i>	<i>Local Binary Patern</i>	<i>Local Binary Patern</i>	<i>Local Binary Patern</i>
	Rekognisi Karakter	<i>Optical Character Recognition</i>	<i>Optical Character Recognition</i>	<i>Optical Character Recognition</i>	<i>Optical Character Recognition</i>
	<i>Image Enhancement</i>	-	-	<i>Top Hat Transform</i>	<i>Top Hat Transform, Bottom Hat Transform, Top Bottom Hat Transform</i>
Hasil	Akurasi 80,3%	Akurasi 80,9%	Akurasi 54,6%	Akurasi 83%	

