

# PENERAPAN ALGORITMA BLOB DETECTION DAN OPERATOR SOBEL PADA APLIKASI PENGENALAN OBJEK CITRA PLAT NOMOR KENDARAAN

Riyadi J. Iskandar<sup>1</sup>, Kartono<sup>2</sup>, Agusalim<sup>3</sup>

<sup>1,2,3</sup>Informatika Fakultas Teknologi Informasi Universitas Widya Dharma, Pontianak  
e-mail: <sup>1</sup>riyadijiskandar@gmail.com, <sup>2</sup>kartono@widyadharm.ac.id, <sup>3</sup>lucius.agus@gmail.com

## Abstract

*Object recognition is a technique for the identification of a specific object from an image by using a specific algorithm. This object recognition techniques can be applied to many things, one of which is to recognize the image of the vehicle license plates. Vehicle license plate is an identity for each vehicle. Some of the Supporting methods that can be used to perform object recognition i.e. Sobel operator and Blob detection. The algorithm will analyze images captured by a camera that can recognize objects such as license plate numbers and then the feature in the image blob detection results will be classified so it can be identified. Through this study, the authors wanted to design an application that apply image processing techniques using Sobel operator and blob detection algorithm to detect objects contained in a license plate.*

**Keywords :** *Object Recognition, Vehicle License Plate, Software, Algorithm, Sobel Operator, Blob Detection*

## Abstrak

Pengenalan objek merupakan sebuah teknik untuk melakukan identifikasi terhadap sebuah objek tertentu dari sebuah citra dengan menggunakan algoritma tertentu. Teknik pengenalan objek ini dapat diterapkan pada berbagai hal, salah satunya adalah untuk mengenali citra plat nomor kendaraan. Plat nomor kendaraan merupakan sebuah identitas bagi setiap kendaraan. Beberapa metode pendukung yang dapat digunakan untuk melakukan pengenalan objek yaitu operator sobel dan blob detection. Algoritma tersebut akan melakukan analisa citra yang ditangkap oleh kamera sehingga dapat mengenali objek seperti plat nomor kendaraan dan kemudian fitur pada citra hasil blob detection akan diklasifikasikan sehingga dapat dikenali. Melalui penelitian ini, penulis ingin merancang sebuah aplikasi yang menerapkan teknik pengolahan citra dengan menggunakan operator sobel dan algoritma blob detection untuk mendeteksi objek yang terdapat pada suatu plat nomor kendaraan.

**Kata Kunci :** *Pengenalan Objek, Plat Nomor, Perangkat Lunak, Algoritma, Operator Sobel, Blob Detection.*

## 1. PENDAHULUAN

Pada era globalisasi seperti sekarang ini, kebutuhan masyarakat akan teknologi cukup tinggi terutama komputer. Komputer adalah suatu perangkat elektronik yang dapat digunakan untuk menghitung suatu bentuk operasi aritmatika, mengakses suatu informasi, maupun menyimpan suatu informasi ke dalam memori. Komputer banyak dibutuhkan masyarakat sebagai sarana untuk mengolah berbagai jenis data seperti gambar, suara, video, dan lain sebagainya. Hal ini berdampak terhadap tingginya kebutuhan masyarakat terhadap komputer.

Salah satu manfaat dari komputer yang sangat penting yaitu sebagai media pengenalan objek (*object recognition*). Pengenalan objek adalah kemampuan untuk memahami sifat fisik suatu objek (seperti bentuk, warna dan tekstur) dan menerapkan atribut semantik ke objek, yang meliputi pemahaman penggunaannya, pengalaman sebelumnya dengan objek dan bagaimana kaitannya dengan orang lain. Objek yang dimaksud dapat berupa suara, gambar, tulisan, dan lain sebagainya.

Salah satu penerapan dari pengenalan objek dalam suatu perangkat lunak adalah perangkat lunak pengenalan objek terhadap citra plat nomor kendaraan bermotor. Berbagai metode dan algoritma telah dibangun untuk mendukung proses pengenalan objek diantaranya adalah *sobel operator* dan *blob detection*.

Pada penelitian ini, penulis akan membahas penggunaan *sobel operator* dan *blob detection* untuk mengenali objek yang terdapat pada suatu citra plat nomor kendaraan dimana pengenalan akan melalui tahapan pemrosesan citra, tahapan segmentasi dan tahapan pengenalan objek.

## 2. METODE PENELITIAN

### 2.1. Rancangan Penelitian, Teknik Analisis, dan Perancangan Sistem

#### 2.1.1. Rancangan Penelitian

Rancangan penelitian yang dilakukan oleh penulis yaitu berupa desain penelitian deskriptif. Penelitian deskriptif merupakan penelitian yang dilakukan dengan cara menggambarkan komponen apa saja yang diperlukan untuk menyajikan aplikasi kepada pengguna komputer. Dengan menggunakan bentuk rancangan penelitian ini, penulis dapat diketahui apa yang menjadi kelebihan dan kekurangan dari aplikasi yang dirancang.

#### 2.1.2. Teknik Analisis Sistem

Teknik analisis data yang digunakan penulis untuk menggambarkan jalannya aliran data ke dalam sistem yaitu dengan menggunakan *Unified Modeling Language* (UML)

#### 2.1.3. Teknik Perancangan Sistem

Teknik perancangan sistem yang digunakan penulis untuk merancang perangkat lunak pemindai plat nomor menggunakan algoritma *blob detection* adalah bahasa pemrograman *java* dengan menggunakan IDE NetBeans.

### 2.2. Landasan Teori

#### 2.2.1. Data

Data merupakan kenyataan yang menggambarkan suatu kejadian dan merupakan kesatuan nyata yang nantinya akan digunakan sebagai bahan dasar suatu informasi<sup>[1]</sup>. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata<sup>[2]</sup>.

#### 2.2.2. Sistem

Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variable-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung sama lain<sup>[3]</sup>. Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu<sup>[4]</sup>.

#### 2.2.3. Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah sebuah profesi yang dilakukan oleh seorang perancang perangkat lunak yang berkaitan dengan pembuatan dan pemeliharaan aplikasi perangkat lunak dengan menerapkan teknologi dan praktisi dari ilmu komputer, manajemen proyek, dan bidang-bidang lainnya<sup>[5]</sup>. Proses pengembangan/rekayasa perangkat lunak sesungguhnya merupakan aktivitas-aktivitas yang diperlukan untuk menerjemahkan 'kebutuhan dan harapan pengguna' menjadi sebuah sistem perangkat lunak<sup>[6]</sup>.

#### 2.2.4. Analisis Sistem

Analisis sistem adalah penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan yang diharapkan sehingga diusulkan perbaikan perbaikannya<sup>[7]</sup>. Analisis sistem didefinisikan sebagai bagaimana memahami dan menspesifikasi dengan detail apa yang harus dilakukan oleh sistem<sup>[8]</sup>.

#### 2.2.5. Perancangan Sistem

Perancangan sistem adalah proses pengembangan spesifikasi sistem baru berdasarkan hasil rekomendasi analisis sistem<sup>[9]</sup>. Tujuan rancangan sistem adalah menggambarkan sistem yang baru sebagai sebuah kumpulan modul atau sub sistem<sup>[10]</sup>.

#### 2.2.6. Perancangan Masukan

Perancangan input merupakan bentuk perancangan *form* pengisian yang berkaitan dengan sistem pengolahan data dalam program<sup>[11]</sup>. Desain input dirancang sebagai tampilan antarmuka di mana pengguna bisa memasukkan input baik berupa data melalui keyboard maupun perangkat lain<sup>[12]</sup>.

#### 2.2.7. Perancangan Keluaran

Perancangan *output* merupakan perancangan yang berkaitan dengan informasi hasil pengolahan data<sup>[13]</sup>. Piranti keluaran (alat output) dirancang untuk memudahkan pengguna membaca atau melihat hasil yang dikerjakan oleh mesin tersebut<sup>[14]</sup>.

#### 2.2.8. Perangkat Lunak atau Aplikasi

Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus)<sup>[15]</sup>. Program aplikasi merupakan perangkat lunak (*software*) yang digunakan untuk tujuan tertentu seperti mengolah dokumen, mengatur Windows 7, permainan (*game*), dan sebagainya<sup>[16]</sup>.

#### 2.2.9. Plat Nomor

Setiap Kendaraan Bermotor yang dioperasikan di jalan wajib dilengkapi dengan dengan Surat Tanda Kendaraan Bermotor dan Tanda Nomor Kendaraan Bermotor. Tanda Nomor Kendaraan Bermotor (TNKB) tersebut harus memuat kode wilayah, nomor registrasi dan masa berlaku. Tanda Nomor Kendaraan Bermotor ini sering disebut dengan Nomor Polisi (Nopol) atau Plat Nomor. Bahan yang digunakan untuk Plat Nomor Kendaraan Bermotor di Indonesia adalah terbuat dari Aluminium dan wajib dipasangkan di setiap kendaraan bermotor yang dioperasikan di Jalan<sup>[17]</sup>. Plat Nomor/ Nomor Polisi (Nopol) adalah plat aluminium Tanda Nomor Kendaraan Bermotor (TNBK) di Indonesia yang telah didaftarkan di kantor bersama SAMSAT (Sistim Administrasi Manunggal Satu Atap/ One Roff System). Penggunaan tanda nomor kendaraan bermotor di Indonesia merupakan warisan Hindia Belanda<sup>[18]</sup>.

#### 2.2.10. Citra Digital

*A digital image is a set of bits (zeroes and ones) that represents an image. That set of bits is created following a specific digital image format so that different programs can understand it.* (Sebuah citra digital terdiri dari serangkaian bit (nol dan satu) yang mewakili sebuah citra. Rangkaian dari bit tersebut dibuat berdasarkan format spesifik citra digital sehingga dapat dipahami oleh program.)<sup>[19]</sup> *A digital image is composed of picture*

*elements called Pixels. Pixels are the smallest sample of an Image. A pixel represents the brightness at one point. (Sebuah citra digital terbentuk dari elemen-elemen gambar yang disebut dengan Pixel. Pixel adalah sampel terkecil dari sebuah citra. Sebuah pixel mewakili kecerahan pada satu titik.)*<sup>[20]</sup>.

#### 2.2.11. Citra Grayscale

Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixelnya, dengan kata lain nilai bagian RED = GREEN = BLUE. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas<sup>[21]</sup>. *In a gray-scale image, each pixel is represented by a byte or word, the value of which represents the light intensity at that point in the image. An 8-bit image will have a brightness variation from 0 to 255 where '0' represents black and '255' represents white.* (Dalam citra keabuan, setiap piksel diwakili oleh *byte* atau kata, yang nilainya mewakili intensitas cahaya saat itu. Sebuah citra 8-bit akan memiliki variasi tingkat kecerahan dari 0 sampai 255 dimana '0' mewakili hitam dan '255' mewakili putih.)<sup>[22]</sup>.

#### 2.2.12. Operator Sobel

*Sobel edge detection which, again, consists of two masks to determine the edge in vector form. the Sobel operator was the most popular edge detection techniques with a theoretical basis. It proved popular because it gave, overall, a better performance than other contemporaneous edge detection operators, such as the Prewitt Operator.* (Deteksi tepi Sobel yang, terdiri dari dua penutup untuk menentukan tepi dalam bentuk vektor. Operator Sobel merupakan teknik deteksi tepi yang paling populer berbasis teoritis. Terbukti populer karena secara keseluruhan, ia memberikan kinerja yang lebih baik dari operator deteksi tepi kontemporer lainnya, seperti operator Prewitt.)<sup>[23]</sup>.

#### 2.2.13. Blob Detection

*There are many methods for extracting objects (blobs) from images. In this paper we use methods implemented in the AForge.NET library. These algorithms are described by Andrew Kirillov. There are four types of the algorithms: Convex full, Left/Right Edges, Top/Bottom Edges, Quadrilateral. Fig.1 describes these blob detection methods.* (Terdapat berbagai macam metode untuk mengekstraksi objek (blob) dari suatu gambar. Dalam makalah ini kami menggunakan metode yang diterapkan dalam AForge.Net. Algoritma-algoritma ini dijelaskan oleh Andrew Kirillov. Terdapat empat jenis algoritma: Cembung penuh, Tepi Kiri/Kanan, Tepi Atas/Bawah, Persegi<sup>[24]</sup>. *Blob detection and extraction proves to be a useful tool in many areas; one main application is to provide complementary information about regions, which is not obtained from edge detectors or corner detectors. In early work in the area, blob detection was used to obtain regions of interest for further processing.* (Blob detection dan extraction terbukti menjadi alat yang berguna dalam berbagai bidang; salah satu aplikasi utamanya adalah untuk memberikan informasi pelengkap tentang wilayah, yang tidak diperoleh dari detektor tepi atau detektor sudut. Dalam pengerjaan awal, *blob detection* digunakan untuk mendapatkan daerah yang menarik untuk diproses lebih lanjut.)<sup>[25]</sup>.

#### 2.2.14. Feature Extraction

*Feature extraction means identifying certain attributes of interest. The feature extraction is usually a non-invertible transformation that loses some information.* (Ekstraksi fitur berarti mengidentifikasi atribut-atribut tertentu yang menarik. Ekstraksi fitur biasanya merupakan sebuah transformasi yang mengalami proses kehilangan informasi dan tidak dapat dibalik.)<sup>[26]</sup>.

#### 2.2.15. Pengertian UML

UML (*Unified Modeling Language*) adalah 'bahasa' pemodelan untuk sistem atau perangkat lunak yang berparadigma 'berorientasi objek'. Pemodelan sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami<sup>[27]</sup>.

#### 2.2.16. Java

Bahasa pemrograman Java merupakan bahasa pemrograman berorientasi masa depan atau bias jadi, Java adalah bahasa masa depan itu sendiri<sup>[28]</sup>. Java adalah program yang sudah termasuk termasuk dalam *repository* dalam setiap versi dari ubuntu, tinggal kita hubungkan ke *repository* lalu melakukan instalasi java<sup>[29]</sup>.

#### 2.2.17. NetBeans

*NetBeans is a modular, standards-based integrated development environment (IDE) that software developers use to create and modify applications written in Java.* (NetBeans adalah sebuah modular, integrated development environment (IDE) berbasis standar yang digunakan oleh pengembang perangkat lunak untuk membuat dan memodifikasi aplikasi yang tertulis dalam Java.)<sup>[30]</sup>. *NetBeans is a Java Integrated Development Environment, IDE, which enables fast application development with the most adopted frameworks, technologies, and servers.* (NetBeans merupakan sebuah *integrated development environment (IDE)* berbasis java yang memungkinkan pengembangan aplikasi yang cepat dengan kerangka kerja, teknologi, dan server yang paling banyak diadopsi.)<sup>[31]</sup>.

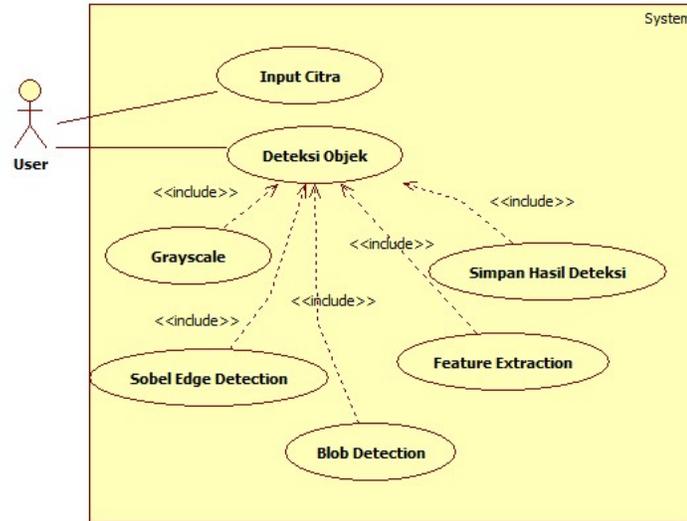
### 3. HASIL DAN PEMBAHASAN

#### 3.1. Gambaran Umum Rancangan Perangkat Lunak melalui Use Case Diagram

*Use Case* adalah sebuah kumpulan dari diagram dan teks yang mendeskripsikan bagaimana keinginan pengguna berinteraksi dengan sistem. Dengan *Use Case* mengidentifikasi fungsionalitas yang disediakan oleh

sistem (*Use Case*) dan pengguna (*Actor*) yang berinteraksi dengan sistem. *Use Case* program ini memiliki langkah-langkah sebagai berikut:

- a. Pengguna dapat melakukan *input* gambar dengan memilih tombol “*Load Image*”.
- b. Pengguna dapat melakukan pendeteksian objek dengan memilih menu proses yang telah disediakan dan kemudian menekan tombol “*Process*” untuk memulai proses yang dipilih.
- c. Pengguna dapat menyimpan hasil deteksi objek dengan memilih tombol “*Save Image*”.



Gambar 1. Use Case Diagram

### 3.2. Tampilan Aplikasi

#### 3.2.1. Form Utama

Pada *form* Utama, pengguna dapat berinteraksi langsung dengan perangkat lunak menggunakan tombol yang tersedia. Menu *save image* akan menyimpan citra hasil deteksi, menu *load image* akan memanggil *form* pilih gambar untuk membantu pengguna memilih gambar untuk di deteksi, dan menu *exit* akan mengeluarkan pengguna dari perangkat lunak.

Menu proses memiliki tujuh buah menu yang berbentuk *checkbox*. Menu *grayscale* berfungsi untuk mengubah citra berwarna menjadi citra keabuan. Menu *mean filter* dan *median filter* berfungsi untuk menghilangkan *noise* yang terdapat pada citra keabuan. Menu *sobel edge detection* akan melakukan proses deteksi tepi terhadap citra. Menu *blobbing* akan melakukan proses *blob detection* pada citra. Menu *region splitting* akan melakukan pemisahan *region of interest* yang terdeteksi. Menu *feature extraction* akan melakukan proses ekstraksi fitur untuk mengklasifikasikan objek yang terdeteksi.

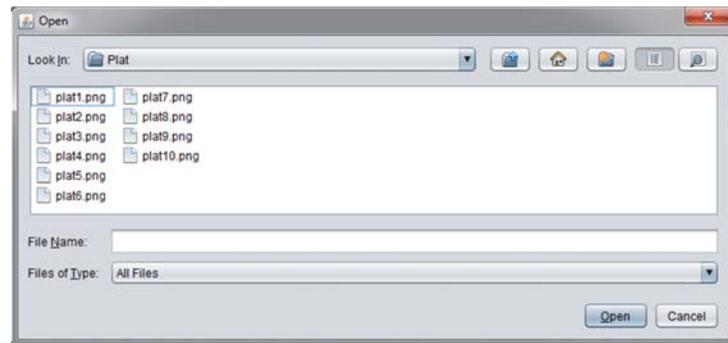
*Listbox* hasil akan menampilkan hasil berupa objek yang terdeteksi pada citra, tombol *process* memulai fungsi proses deteksi objek, dan tombol *reset* untuk membersihkan tampilan *form* dan membawa pengguna kembali ke tampilan awal.



Gambar 2 Menu Utama

### 3.2.2. Form Pilih Gambar

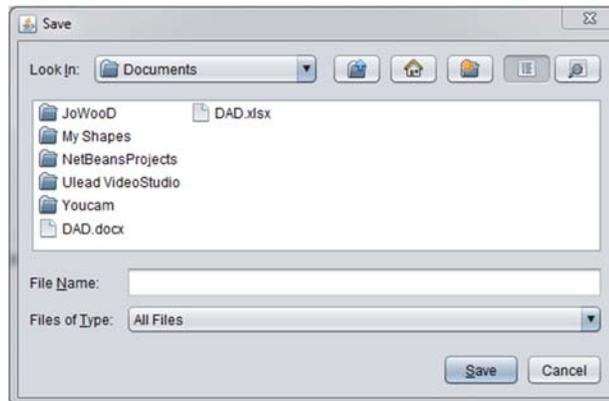
Pada *form* ini pengguna dapat memilih gambar plat nomor kendaraan bermotor yang akan digunakan untuk proses deteksi objek. Karakteristik *form* pilih gambar sama dengan *form open file* yang terdapat dalam perangkat lunak lainnya. Menu ini dapat diakses melalui menu *file* pada *form* utama kemudian pilih menu *load image*.



Gambar 3 Form Load Image

### 3.2.3. Form Simpan Hasil Deteksi

*Form* simpan hasil deteksi memiliki karakteristik yang hampir sama dengan *form* pilih gambar sebelumnya. *Form* simpan memiliki rancangan yang hampir sama dengan *form* pilih gambar, yang membedakannya hanya tombol *save* yang menggantikan tombol *open* pada *form* pilih gambar. Menu ini dapat diakses melalui menu *file* yang terdapat pada *form* utama kemudian pilih menu *save image*.



Gambar 4 Form Save Image

### 3.2.4. Form About

*Form about* berfungsi untuk menampilkan data diri pembuat atau perancang perangkat lunak. *Form about* memiliki *textbox* yang akan menampilkan data diri perancang perangkat lunak, dan tombol ok yang akan membawa pengguna kembali ke *form* utama ketika dipilih. Menu ini dapat diakses melalui menu *about* yang terdapat pada menu utama.



Gambar 5 Form About

### 3.2.5. Pengujian Perangkat Lunak

Pada bagian ini penulis akan menampilkan beberapa hasil pengujian yang dilakukan selama proses penelitian dalam bentuk tabel. Berikut tabel pengujian yang dibuat oleh penulis:

Tabel 1 Tabel Pengujian

Plat	Intensitas Cahaya	Blobbing	Jumlah Objek Terdeteksi	Jumlah Objek Terklasifikasi	Karakteristik Plat	Keterangan
	Kurang	Gagal	5 (K,9,3,N,H)	5 (K,9,3,N,H)	Miring	Objek tidak terdeteksi dan terklasifikasi dengan baik.
	Terlalu Terang	Gagal	-	-	Miring, blur	Tidak terdeteksi apa-apa.
	Terang	Kurang berhasil (ada blob yang tergabung)	7 (K,B,1,77,6,H,E)	7 (K,B,1,77,6,H,E)	Normal	Ada blob yang tergabung menjadi 1 objek mengakibatkan objek tersebut tidak dapat diklasifikasikan.
	Terang	Berhasil namun kurang sempurna	6 (K,B,9,0,N,S)	5 (K,B,9,0,S)	Miring, blur	Objek yang teridentifikasi tidak sama dengan objek yang terdeteksi.
	Cukup	Berhasil dan hampir sempurna	8 (K,B,1,3,5,1,Q,A)	8 (K,B,1,3,5,1,Q,A)	Normal	Sesuai dengan yang diharapkan.
	Ada pantulan cahaya	Berhasil dan hampir sempurna	8 (K,B,1,2,6,5,A,T)	8 (K,B,1,2,6,5,A,T)	Normal	Blobbing ada yang gagal namun berhasil diidentifikasi dalam feature extraction.

Dari tabel pengujian di atas, terdapat beberapa kasus dengan plat yang memiliki kondisi yang berbeda pula. Seperti misalnya plat dengan kondisi cahaya yang terlalu dan plat yang gambarnya *blur* sehingga perangkat lunak tidak dapat mendeteksi apa-apa. Setelah melalui beberapa proses pengujian, dapat disimpulkan bahwa perangkat lunak mampu bekerja dengan baik apabila plat dalam kondisi yang cukup baik dan dengan tingkat intensitas cahaya yang baik pula.

### 3.3. Spesifikasi Sistem yang Disarankan

Agar perangkat lunak pemindai plat ini dapat berjalan dengan baik, dibutuhkan perangkat dengan spesifikasi tertentu. Spesifikasi ini disarankan agar pengguna dapat terhindar dari berbagai masalah pada saat melakukan proses deteksi, seperti misalnya terjadi *hang* pada komputer atau laptop. Berikut ini merupakan beberapa komponen yang diperlukan serta spesifikasi sistem yang disarankan.

1. Prosesor AMD Athlon(tm) II X2 270 CPU @ 3.40GHz
2. Memori (RAM) 4 GB
3. Keyboard dan Mouse
4. Sistem Operasi Windows 7 Ultimate 64-bit
5. Java 7 dan versi di atasnya

## 4. KESIMPULAN

Berdasarkan pembahasan yang telah diuraikan pada bab sebelumnya, dapat diambil kesimpulan tentang aplikasi pengenalan objek terhadap citra plat nomor kendaraan sebagai berikut:

- a. Aplikasi ini digunakan untuk mengetahui cara kerja proses pengenalan objek menggunakan algoritma *sobel* operator dan *blob detection*.
- b. Besarnya tingkat keakuratan mengenali objek yang terdapat pada plat tergantung pada kondisi yang dimiliki plat yang digunakan sebagai objek deteksi.
- c. Untuk mendapatkan hasil yang akurat selama proses pengenalan objek, harus diperhatikan data *input* yang digunakan, proses yang dipilih saat pengujian dan kondisi saat plat selesai dikenali.

## 5. SARAN

Setelah melakukan pengujian pada aplikasi pengenalan objek terhadap citra plat nomor kendaraan menggunakan operator *sobel* dan algoritma *blob detection*, penulis menyadari bahwa aplikasi ini masih memiliki

banyak kelemahan dan jauh dari sempurna. Oleh sebab itu penulis memberikan saran yang dapat bermanfaat kepada pembaca serta tidak menutup kemungkinan untuk mengembangkan dan memperbaiki kekurangan yang ada adalah sebagai berikut:

- a. Untuk mendapatkan hasil deteksi yang akurat, penulis menyarankan untuk menggunakan foto atau gambar plat yang sesuai dengan kriteria yang terdapat pada pembatasan masalah.
- b. Perlu dilakukan pengembangan algoritma lebih lanjut sehingga dapat mengenali plat dengan berbagai kondisi serta menghasilkan *output* yang lebih baik.
- c. Untuk menjalankan aplikasi ini, disarankan memiliki spesifikasi komputer minimum yang seperti digunakan penulis atau lebih, dikarenakan aplikasi ini masih terhitung berat dikarenakan adanya beberapa proses pengolahan citra yang membutuhkan spesifikasi *hardware* yang mendukung sehingga tidak mengakibatkan komputer pengguna mengalami *hang* saat proses deteksi objek dijalankan.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada civitas akademika Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak atas segala dukungan terhadap penelitian ini.

## DAFTAR PUSTAKA

- [1] Anhar. (2010). *Cara Mudah Mengamankan Data Komputer & Laptop*. Mediakita. Jakarta.
- [2] Jogiyanto. H.M. (2009). *Analisis dan Desain Sistem Informasi Edisi III*. Andi Offset. Yogyakarta.
- [3] Al Fatta, Hanif. (2007). *Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*. Andi Offset. Yogyakarta. Sutabri, T., (2014), *Pengantar Teknologi Informasi*, Andi Offset, Yogyakarta.
- [4] Jogiyanto. H.M. (2009). *Analisis dan Desain Sistem Informasi Edisi III*. Andi Offset. Yogyakarta.
- [5] Simarmata, Janner. (2010). *Rekayasa Perangkat Lunak*. Andi Offset. Yogyakarta.
- [6] Nugroho, A., (2010), *Rekayasa Perangkat Lunak Berbasis Objek dengan Metode USDP*, Andi Offset, Yogyakarta.
- [7] Jogiyanto. H.M. (2009). *Analisis dan Desain Sistem Informasi Edisi III*. Andi Offset. Yogyakarta.
- [8] Al Fatta, Hanif. (2007). *Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*. Andi Offset. Yogyakarta.
- [9] Kusrini, dan Andri K., (2007), *Tuntunan Praktis Membangun Sistem Informasi Akuntansi dengan Visual Basic dan Microsoft SQL Server*, Andi Offset, Yogyakarta.
- [10] Gaol, J., (2008), *Sistem Informasi Manajemen Pemahaman dan Aplikasi*, Grasindo, Jakarta.
- [11] Kusrini, dan Andri K., (2007), *Tuntunan Praktis Membangun Sistem Informasi Akuntansi dengan Visual Basic dan Microsoft SQL Server*, Andi Offset, Yogyakarta.
- [12] Al Fatta, H., (2009), *Rekayasa Sistem Pengenalan Wajah*, Andi Offset, Yogyakarta.
- [13] Kusrini, dan Andri K., (2007), *Tuntunan Praktis Membangun Sistem Informasi Akuntansi dengan Visual Basic dan Microsoft SQL Server*, Andi Offset, Yogyakarta.
- [14] Zakaria, Teddy Marcus, dan Agus Prijono. (2007). *Perancangan Antarmuka untuk Interaksi Manusia dan Komputer*. Informatika. Bandung.
- [15] Hendrayudi. (2009). *VB 2008 untuk Berbagai Keperluan Programming*. PT Elex Media Komputindo. Jakarta.
- [16] Hakim S, R., (2010), *Buku Pintar Windows 7*, PT Elex Media Komputindo, Jakarta.
- [17] Dickson. (2014). *Kode Plat Nomor / Tanda Nomor Kendaraan Bermotor di Indonesia*. Diambil dari: <http://ilmupengetahuanumum.com/kode-plat-nomor-tanda-nomor-kendaraan-bermotor-di-indonesia/> (27 Maret 2015).
- [18] REP. (2012). *Plat Nomor Cantik di Indonesia*. Diambil dari: [http://www.kompasiana.com/beecool/plat-nomor-cantik-di-indonesia\\_55101d67813311d738bc6141](http://www.kompasiana.com/beecool/plat-nomor-cantik-di-indonesia_55101d67813311d738bc6141) (27 Maret 2015).
- [19] Montabone, Sebastian (2010). *Beginning Digital Image Processing: Using Free Tools for Photographers*. Apress. United States.
- [20] Jayaraman S. (2011). *Digital Image Processing*. Tata McGraw-Hill Education. India.
- [21] Putra, Darma. (2010). *Pengolahan Citra Digital*. Andi Offset. Yogyakarta.
- [22] Jayaraman S. (2011). *Digital Image Processing*. Tata McGraw-Hill Education. India.
- [23] Nixon, Mark., dan Alberto Aguado. (2013). *Feature Extraction and Image Processing*. Newnes. Oxford.
- [24] Rutkowski, Leszek. (2014). *Artificial Intelligence and Soft Computing*. Springer. Germany.
- [25] Damiani, Ernesto. dan Jechang Jeong (2009). *Multimedia Techniques for Device and Ambient Intelligence*. Springer. Germany.
- [26] Ritcher, Michael M. dan Rosina Weber. (2013) *Case-Based Reasoning: A Textbook*. Springer Science & Business Media. Germany.
- [27] Nurgoho, Adi. (2010). *Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP*. Andi Offset. Yogyakarta.
- [28] Nugroho, Adi. (2009). *Rekayasa Perangkat Lunak Menggunakan UML dan Java*. Edisi 1. Andi. Yogyakarta.

- [29] Safaat, Nazruddin H. (2012). *Pemrograman Aplikasi Mobile Smartphone Dan Tablet PC Berbasis Android*. Informatika. Bandung.
- [30] Goldman, Ron. dan Richard P. Gabriel. (2005). *Innovation Happens Elsewhere: Open Source as Business Strategy*. Morgan Kaufmann. United States.
- [31] Dantas, Rhawi. (2011). *NetBeans IDE 7 Cookbook*. Packt Publishing Ltd. United Kingdom.