

## IMPLEMENTASI SUPPORT VECTOR MACHINE UNTUK PENGENALAN PLAT NOMOR KENDARAAN

**Tholib Hariono**

Universitas KH. A. Wahab Hasbullah Jombang

Email: hariono@unwaha.ac.id



©2019 –EPiC Universitas KH. A. Wahab Hasbullah Jombang ini adalah artikel dengan akses terbuka dibawah lisensi CC BY-NC-4.0 (<https://creativecommons.org/licenses/by-nc/4.0/>).

### Abstract

*License plate recognition has a very important role in developing better transport systems such as electronic tolls, electronic parking, traffic monitoring activities and others. Number plate recognition process is carried out through four main stages (plate detection, segmentation, feature extraction and classification). Plate detection is done to obtain the location of license plate using the Viola Jones algorithm. The segmentation process to separate the characters from the image of the license plate area-based color using labeling techniques. Feature extraction is based on the division of the image into several blocks and then calculates the average value of pixels every block using the integral image method to obtain characteristics. In classification using SVM with 3 kernels: linear, RBF and polynomials. The test results character acquired 93.75% accuracy is best in the division images of 50 blocks with polynomial kernel, and 82.86% accuracy in testing plate.*

**Keywords** :Support Vector Machine, License Plate, Viola Jones, image block

### Abstrak

Pengenalan plat nomor kendaraan memiliki peranan yang sangat penting dalam mengembangkan sistem transportasi yang lebih baik seperti tol elektronik, parkir elektronik, pemantauan aktifitas lalu lintas dan lain-lain. Proses pengenalan plat nomor dilakukan melalui empat tahap utama (deteksi plat, segmentasi, ekstraksi ciri dan klasifikasi). Deteksi plat dilakukan untuk memperoleh lokasi plat nomor menggunakan algoritma *Viola Jones*. Proses segmentasi untuk memisahkan karakter dari citra plat nomor berbasis area warna menggunakan teknik *labeling*. Ekstraksi ciri didasarkan pada pembagian citra menjadi beberapa blok kemudian menghitung rata-rata nilai piksel per blok menggunakan metode citra integral untuk memperoleh ciri. Pada tahap klasifikasi menggunakan SVM dengan 3 kernel: *linear*, *rbf* dan *polynomial*. Hasil pengujian per karakter diperoleh akurasi terbaik 93.75% pada pembagian citra 50 blok dengan *kernel polynomial*, dan akurasi 82.86% pada pengujian per plat.

**Kata kunci** :Support Vector Machine, Plat Nomor, Viola Jones, Blok Citra

### 1. Pendahuluan

Pengenalan plat nomor (PPN) kendaraan secara otomatis memiliki peranan yang sangat penting dalam mengembangkan sistem transportasi yang lebih baik. Plat nomor kendaraan bersifat unik pada setiap kendaraan bermotor. Plat tersebut dapat

digunakan untuk berbagai kebutuhan pencatatan data kendaraan, misalnya diterapkan pada berbagai layanan seperti tol elektronik, parkir elektronik, pemantauan aktifitas lalu lintas, mendeteksi kendaraan yang tersangkut masalah kriminal dan lain-lain (Runmin Wang, dkk, 2014).

Pada umumnya proses pengenalan plat nomor kendaraan dilakukan melalui dua langkah utama, yaitu deteksi lokasi objek (plat nomor) dan pembacaan karakter plat nomor tersebut. Deteksi lokasi objek digunakan untuk mengetahui posisi lokasi plat nomor serta memisahkannya dari gambar yang kompleks untuk memudahkan proses berikutnya karena objek yang ditangani terfokus pada citra plat nomor saja. Pembacaan karakter plat nomor untuk mengetahui karakter apa saja yang ada pada citra plat nomor tersebut. Pada tahapan ini memiliki tiga proses yaitu segmentasi karakter, ekstraksi ciri dan klasifikasi.

Pada penelitian ini penulis menggunakan metode *Viola Jones* untuk mendeteksi lokasi plat nomor kendaraan karena penelitian menunjukkan selain hasil deteksi yang baik metode ini juga mampu bekerja pada kondisi gambar yang memiliki kompleksitas tinggi.

Pada bagian pembacaan karakter plat nomor dilakukan melalui beberapa proses yaitu segmentasi karakter, ekstraksi ciri dan klasifikasi. Segmentasi untuk memisahkan atau mengambil tiap-tiap citra karakter yang ada di dalam plat nomor untuk memudahkan dalam mengenali karakter apa saja dalam plat tersebut. Ekstraksi ciri untuk mendapatkan informasi yang penting dari citra karakter sehingga informasi yang akan diproses lebih sederhana. Ekstraksi ciri pada penelitian ini didasarkan pada pembagian blok atau zona, yaitu membagi citra menjadi beberapa blok kemudian menghitung nilai rata-rata pixel masing-masing area blok dengan metode citra integral. Hasil perhitungan piksel pada masing-masing blok akan menjadi sebuah data ciri yang dipakai dalam klasifikasi untuk mengenali karakter.

Pada tahap klasifikasi menggunakan metode SVM. SVM berada dalam satu kelas dengan Jaringan Saraf Tiruan (JST) dalam hal fungsi dan kondisi permasalahan yang dapat diselesaikan, keduanya termasuk *supervised learning*, namun dalam implementasi menunjukkan SVM lebih baik dari JST. JST menemukan solusi melalui *local optimal* sehingga solusi yang dicapai tidak selalu sama, sedangkan SVM menemukan solusi melalui *global optimal* dan mencapai solusi yang sama dalam setiap runningnya. Selain itu pada JST semua data latih akan dipelajari selama proses *training*, sehingga waktu yang dibutuhkan relatif lama, hal ini berbeda dengan SVM yang hanya menggunakan beberapa data *training* yang memiliki kontribusi terhadap pembentukan model klasifikasi. Dalam proses pembelajarannya, SVM berusaha untuk mencari *hyperplane* yang terbaik yang berfungsi sebagai pemisah dua buah kelas data pada ruang input (Vapnik, 1995).

## 2. Kerangka Teori

### 2.1. Support Vector Machine (SVM)

#### 2.1.1 Konsep Dasar SVM

Berbeda dengan strategi *neural network* yang berusaha mencari *hyperplane* pemisah antar class, SVM berusaha menemukan *hyperplane* yang terbaik pada ruang input. Prinsip dasar SVM adalah *linear classifier*, dan selanjutnya dikembangkan agar dapat bekerja pada problem *non-linear* dengan memasukkan konsep *kernel trick* pada ruang kerja berdimensi tinggi. *Hyperplane* (pemisah) terbaik antara kedua kelas dapat ditemukan dengan mengukur margin *hyperplane* tersebut dan mencari titik maksimalnya. *Margin* merupakan jarak antara *hyperplane* dengan data yang terdekat pada tiap-tiap kelas. Data yang masuk dalam area *margin* dan *hyperplane* ini disebut dengan *support vector*.

*Hyperplane* yang baik adalah *hyperplane* yang terletak di tengah-tengah antara dua set objek dari dua kelas. Mencari *hyperplane* terbaik berarti mencari margin atau jarak antara dua set objek dari kelas yang berbeda. Jika  $w x_1 + b = +1$  adalah *hyperplane* pendukung dari kelas  $+1$  ( $w x_1 + b = +1$ ) dan  $w x_2 + b = -1$  *hyperplane* pendukung dari kelas  $-1$  ( $w x_2 + b = -1$ ), margin antara dua kelas dapat dihitung dengan mencari jarak antara kedua *hyperplane* pendukung dari dua kelas. Secara spesifik, margin dihitung dengan cara berikut :

$$(w x_1 + b = +1) - (w x_2 + b = -1) \Rightarrow w(x_1 - x_2) = 2 \Rightarrow$$

$$\left( \frac{w}{\|w\|} (x_1 - x_2) \right) = \frac{2}{\|w\|} . \quad (1)$$

Pada dasarnya jumlah fungsi pemisah ini tidak terbatas jumlahnya. Untuk membuktikan bahwa memaksimalkan margin antara dua set obyek akan meningkatkan probabilitas pengelompokkan secara benar dari data testing misalnya dari jumlah yang tidak terbatas ini diambil dua, yaitu  $f_1(x)$  dan  $f_2(x)$ . Fungsi  $f_1$  mempunyai margin yang lebih besar dari pada fungsi  $f_2$ . Setelah menemukan dua fungsi ini, kemudian suatu data baru masuk dengan keluaran  $-1$ . Disini harus dikelompokkan apakah data ini berada dalam kelas  $-1$  atau  $+1$  menggunakan fungsi pemisah yang sudah ditemukan. Dengan menggunakan fungsi  $f_1$ , akan dikelompokkan data baru ini di kelas  $-1$  yang berarti data benar dalam pengelompokannya. Kemudian gunakan  $f_2$  untuk menempatkannya di kelas  $+1$  yang berarti salah.

### 2.1.2 KernelTrick Dan Non Linear Classification Pada SVM

Untuk menangani data non linier digunakan model kernel. Dalam non linear SVM, pertamanya data  $\vec{x}$  dipetakan oleh fungsi  $\Phi$  ke ruang vektor yang berdimensi lebih tinggi agar *hyperplane* yang memisahkan dapat dikonstruksikan.

Pemetaan ini dilakukan dengan menjaga topologi data, dalam artian dua data yang berjarak dekat pada *input space* akan berjarak dekat juga pada *feature space*, sebaliknya dua data yang berjarak jauh pada *input space* akan juga berjarak jauh pada *feature space*.

Selanjutnya proses pembelajaran pada SVM dalam menemukan titik-titik *support vectormenggunakan* fungsi Kernel  $K(\vec{x}_i, \vec{x}_j)$  yang mendefinisikan secara implisit transformasi  $\Phi$ . Hal ini disebut sebagai *Kernel Trick*, yang dirumuskan :

$$K(\vec{x}_i, \vec{x}_j) = \Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j) \quad (2)$$

Tabel 1. Kernel Yang Umum Digunakan Dalam SVM

Jenis Kernel	Definisi
Polynomial	$K(\vec{x}_i, \vec{x}_j) = K(\vec{x}_i, \vec{x}_j + 1)^P$
Rbf	$K(\vec{x}_i, \vec{x}_j) = \exp\left(-\frac{\ (\vec{x}_i, \vec{x}_j)\ ^2}{2\sigma^2}\right)$
Sigmoid	$K(\vec{x}_i, \vec{x}_j) = \tanh(\alpha\vec{x}_i, \vec{x}_j\beta)$

### 2.1.3 Multi Class SVM

SVM saat pertama kali diperkenalkan oleh Vapnik, hanya dapat mengklasifikasikan data ke dalam dua kelas (klasifikasi biner). Namun, penelitian lebih lanjut untuk mengembangkan SVM sehingga bisa mengklasifikasi data yang memiliki lebih dari dua kelas, terus dilakukan. Ada dua pilihan untuk mengimplementasikan *multi class SVM*, pertama menemukan dan menggabung beberapa fungsi pemisah persoalan klasifikasi dua kelas untuk menyelesaikan persoalan klasifikasi multi kelas. Kedua, secara langsung menggunakan semua data dari semua kelas dalam satu formulasi persoalan optimasi.

Berikut ini adalah metode yang umum digunakan untuk mengimplementasikan *multi class SVM* :

#### 1. Metode One-Against-All

Metode ini membangun k buah model SVM biner (k merupakan jumlah kelas). Setiap k berisi data latih yang dibagi menjadi dua kelas. Data latih

yang mewakili dari kelas (k) diberi label +1 dan data latih selain kelas (k) diberi label -1.

Proses klasifikasi dilakukan dari k-1, jika hasil pada k-1 adalah +1 maka data data tersebut termasuk dalam kelas (k-1), jika hasilnya -1 maka akan dilanjutkan klasifikasi ke k-2 dan seterusnya sampai didapat nilai +1.

#### 2. Metode One-Against-One

Dengan menggunakan metode ini, dibangun  $\frac{k(k-1)}{2}$  buah model klasifikasi biner (k adalah jumlah kelas). Setiap model klasifikasi (k) dilatih pada data dari dua kelas (k) yang lain. Penentuan hasil klasifikasi dilakukan melalui teknik voting, kelas atau (k) yang paling sering muncul merupakan hasil klasifikasi, sedangkan jika terjadi jumlah voting yang sama maka hasil klasifikasi didasarkan pada urutan kelas yang paling atas.

### 2.2 Algoritma Viola Jones

Perkembangan aplikasi yang menggunakan sistem deteksi objek saat ini sangat banyak. Deteksi objek bisa dilakukan dengan berbagai cara salah satunya metode *Viola-Jones*. Metode ini memiliki algoritma yang efisien, sehingga tidak memerlukan waktu lama dalam melakukan proses pendeteksian objek.

Ada tiga kontribusi utama yang menjadi kerangka deteksi dari metode ini :

#### 2.2.1 Citra integral

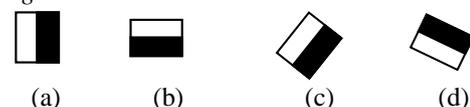
Citra integral memungkinkan untuk evaluasi fitur haar like yang sangat cepat, karena sistem perhitungannya tidak bekerja secara langsung dengan intensitas gambar.

Fitur Haar-like dapat didefinisikan sebuah persegi panjang sederhana yang didalamnya memiliki perbedaan dari jumlah piksel. Perbedaan jumlah piksel tersebut menunjukkan adanya batasan area gelap dan terang pada suatu gambar.

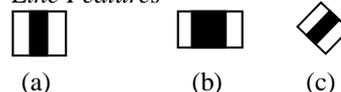
Fitur *haar like* bekerja dengan menganggap daerah persegi panjang yang bedekatan di lokasi tertentu di jendela deteksi, meringkas intensitas pixel di setiap daerah dan menghitung perbedaan antara jumlah tersebut. Perbedaan ini kemudian digunakan untuk mengkategorikan sub bagian dari suatu gambar.

Model fitur *haar like* pada viola jones seperti di bawah ini:

#### 1) Edge Features



#### 2) Line Features



3) Center Surround Features



Gambar 1 model fitur Haar Like

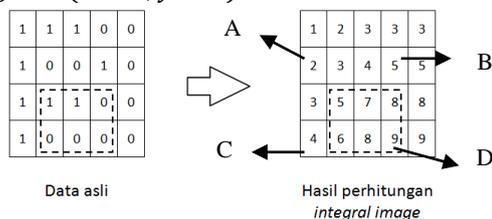
Jumlah fitur *haar like* yang harus dihitung nilai pikselnya sangatlah banyak jumlahnya, bahkan dalam penelitian viola jones, sebuah objek wajah dengan ukuran 24 x 24 piksel memuat sampai 160.000 fitur di dalamnya, sehingga apabila fitur-fitur tersebut dihitung dengan cara klasik akan menyebabkan perhitungan komputasi menjadi berat dan mahal. Metode citra integral akan menjadi sangat cepat karena proses perhitungannya hanya melalui penjumlahan dan pengurangan pada empat titik (A, B, C dan D) yang mewakili area fitur *haar like*.

Proses perhitungan citra integral dimulai berdasar pada citra yang nilai tiap pixel-nya merupakan akumulasi dari nilai pixel atas dan kirinya. Sebagai contoh, pixel(a,b) memiliki nilai akumulatif untuk semua pixel(x,y) dimana  $x \leq a$  dan  $y \leq b$

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (3)$$

Dimana  $ii(x,y)$  adalah citra *integral* dan  $i(x,y)$  adalah citra asli dengan kondisi:

$$ii(x, y) = i(x, y) + ii(x - 1, y) + ii(x, y - 1) - ii(x - 1, y - 1) \quad (4)$$



Gambar 2 contoh perhitungan citra integral

Pada gambar 2 diilustrasikan matrix di sebelah kiri adalah citra asli dan sebelah kanan adalah hasil perhitungan citra integral. Kotak dengan garis putus-putus adalah area yang akan dihitung jumlah pikselnya. Rumus yang digunakan adalah :

$$\begin{aligned} \sum \text{piksel} &= D + A - C - B \\ &= 9 + 2 - 4 - 5 = 2 \end{aligned}$$

Hasil perhitungan tersebut sama dengan hasil perhitungan pada citra asli.

2.2.2 Adaptive Boosting (AdaBoost)

Algoritma *AdaBoost* berfungsi melakukan pemilahan fitur-fitur dalam jumlah banyak dengan hanya memilih fitur-fitur tertentu. *Boosting* merupakan meta-algoritma dalam *machine*

*learning* untuk melakukan *supervised learning*. Kebanyakan algoritma *boosting* mengikuti sebuah rancangan. Secara umum *boosting* terjadi dalam iterasi, secara *incremental* menambahkan *weak learner* ke dalam satu *strong learner*. Pada setiap iterasi, satu *weak learner* belajar dari suatu data latihan, kemudian *weak learner* ditambahkan ke dalam *strong learner*. Setelah itu data-data kemudian diubah masing-masing bobotnya, data-data yang mengalami kesalahan klasifikasi akan mengalami penambahan bobot dan data-data yang terklasifikasi dengan benar akan mengalami pengurangan bobot, sehingga *weak learner* pada iterasi berikutnya menjadi lebih terfokus pada data-data yang mengalami kesalahan klasifikasi oleh *weak learner* yang sebelumnya.

2.2.3 Classifier of Cascade

*Classifier of Cascade* adalah metode untuk menggabungkan berturut-turut pengklasifikasi yang lebih kompleks dalam struktur *cascade* yang secara drastis meningkatkan kecepatan detektor dengan memfokuskan perhatian pada area yang berpeluang pada suatu gambar. Hal ini dilakukan untuk menentukan dimana letak objek yang dicari pada suatu citra. *Classifier of Cascade* ini merupakan tahap terakhir dalam metode *Viola-Jones*.

Adanya klasifikasi bertingkat merupakan karakteristik dari algoritma Viola Jones. Klasifikasi pada algoritma ini terdiri dari beberapa tingkatan dimana setiap tingkatan mengeluarkan sub window yang diyakini bukan objek. Hal ini dilakukan karena lebih mudah untuk menilai sub window tersebut bukan objek yang ingin dideteksi ketimbang menilai apakah sub window tersebut merupakan objek yang ingin dideteksi (Dzulkarnain, 2011).

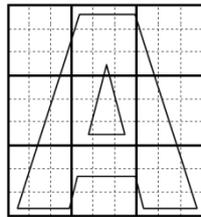
2.4 Segmentasi Citra

Segmentasi citra merupakan proses yang ditujukan untuk mendapatkan objek-objek yang terkandung di dalam citra atau membagi citra ke dalam beberapa daerah dengan setiap objek atau daerah memiliki kemiripan atribut.

Teknik yang digunakan adalah labeling yaitu pemberian label pada sekumpulan citra yang berwarna putih. Hasil dari segmentasi citra adalah sekumpulan wilayah yang melingkupi citra tersebut, atau sekumpulan kontur yang diekstrak dari citra (pada deteksi tepi). Tiap piksel dalam suatu wilayah mempunyai kesamaan karakteristik atau properti yang dapat dihitung.

### 2.2.3.1 Ekstraksi ciri Blok

Metode ekstraksi blok akan membagi citra menjadi beberapa bagian (blok) yang berukuran sama (gambar 3) kemudian dari setiap blok akan diambil cirinya. Ada beberapa algoritma untuk mengambil ciri pada area blok ini, pada penelitian ini penulis menggunakan metode citra integral karena sifatnya yang unik dalam setiap perubahan nilai piksel.



Gambar 3 Contoh pembagian blok citra karakter

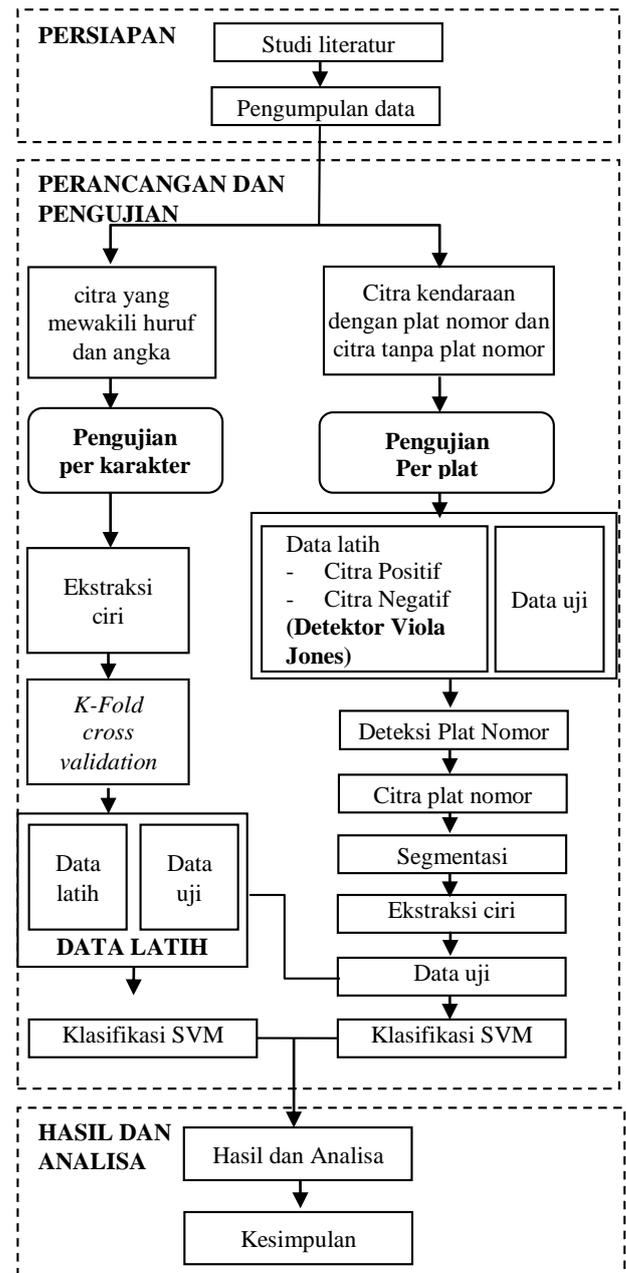
Algoritma dalam penerapan metode pembagian blok adalah :

1. Citra masukan dalam format biner
2. Bagi citra tersebut ke dalam  $n$  blok yang sama
3. Hitung nilai piksel pada masing-masing blok menggunakan citra integral
4. Hitung rata-rata hasil perhitungan point 3
5. Hasilnya adalah  $n$  fitur yang akan digunakan dalam klasifikasi pengenalan karakter.

### 3. Metodologi

Bahan yang digunakan pada penelitian ini terdiri dari pertama bahan untuk klasifikasi yaitu citra per karakter sebanyak 720, kedua bahan pembuatan deteksi objek yaitu citra positif (plat nomor) sebanyak 200 dan citra negatif sebanyak 300, ketiga bahan pengujian per plat sebanyak 70 gambar plat nomor dengan kendaraannya.

Adapun gambaran dari prosedur penelitian ditunjukkan pada gambar 4.



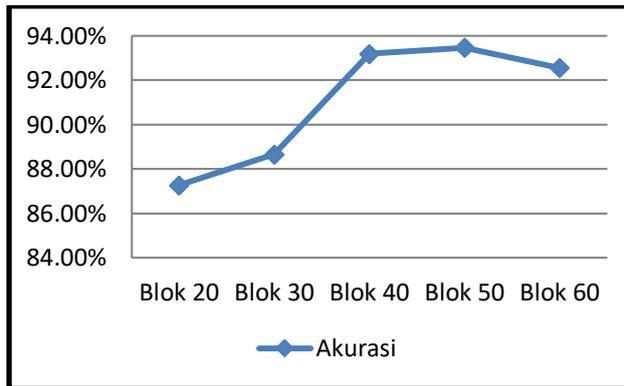
Gambar 4 Prosedur penelitian

### 4. Hasil dan Pembahasan

#### 4.1. Hasil pengujian per karakter

Dari hasil pengujian per karakter diperoleh hasil akurasi pada setiap pembagian blok yang berbeda-beda. Pembagian citra 20 blok diperoleh akurasi terendah dengan rata-rata 87.27%, pada pengujian berikutnya 30 blok diperoleh akurasi sebesar 88.66% terjadi peningkatan sebesar 1,39%, pada pengujian 40 blok diperoleh akurasi sebesar 93.38% terjadi peningkatan yang cukup besar yaitu 4.72%, pada pengujian 50 blok diperoleh akurasi tertinggi yaitu 93.75% terjadi peningkatan sebesar 0.37%, namun pada pengujian terakhir 60 blok

terjadi penurunan 1.2 % dengan akurasi sebesar 92.55%.



Gambar 5 Grafik rata-rata akurasi tiap-tiap blok

Hasil pengujian masing-masing kernel pada pembagian citra 50 blok didapatkan hasil terbaik pada kernel polynomial sebesar 93,89% dan hasil terendah pada kernel rbf sebesar 92,92% sebagaimana ditunjukkan pada tabel 2.

Tabel 2 Hasil pengujian kernel pada pembagian citra 50 blok.

Kernel Function	Akurasi				Rata2
	K1 (%)	K2 (%)	K3 (%)	K4 (%)	
Linear	90.00	93.33	93.33	97.78	93.61
Rbf	91.67	90.00	91.11	98.89	92.92
Polynomial	90.56	92.22	94.44	98.33	93.89

Hasil pengujian pada tiap-tiap karakter menunjukkan beberapa karakter memperoleh akurasi 100%, karakter tersebut adalah “9, C, J, L, O, P, S”, karakter dengan akurasi terendah adalah “Q” sebagaimana ditunjukkan pada tabel 3.

Tabel 3 hasil pengujian tiap-tiap karakter

Char	Kernel function			Char	Kernel function		
	Linear (%)	Rbf (%)	Polynomial (%)		Linear (%)	Rbf (%)	Polynomial (%)
0	90	90	85	I	75	75	95
1	80	85	85	J	100	100	100
2	95	85	95	K	90	90	85
3	90	90	90	L	100	100	100
4	100	95	100	M	75	95	90
5	100	95	95	N	90	95	90
6	100	90	100	O	100	100	100
7	100	95	95	P	100	100	100
8	95	90	100	Q	90	60	80
9	100	100	100	R	90	90	90
A	95	90	95	S	100	100	100
B	85	95	85	T	100	95	100
C	100	100	100	U	95	95	95
D	100	100	95	V	85	90	95

Char	Kernel function			Char	Kernel function		
	Linear (%)	Rbf (%)	Polynomial (%)		Linear (%)	Rbf (%)	Polynomial (%)
E	90	90	90	W	95	100	95
F	95	90	90	X	95	95	95
G	85	95	90	Y	95	95	95
H	100	95	90	Z	95	100	95

4.2 Hasil Pengujian per plat

Pada pengujian ini digunakan sampel sebanyak 70 plat nomor yang tersebar di 59 file gambar, sehingga ada beberapa gambar yang berisi lebih dari satu plat nomor. Cara melakukan pengujian seperti pada gambar 7.



Gambar 7 form pengujian plat nomor

Dari 70 sampel yang dibuat pengujian plat nomor, sebanyak 58 berhasil dibaca dengan baik (82,86%), sisanya sebanyak 12 sampel (17,14%) gagal dibaca. Faktor kegagalan dalam pengujian bervariasi, sebagian besar kegagalan terjadi pada proses deteksi objek sebanyak 6 (8,57%), proses segmentasi sebanyak 5 (7,14%) dan kesalahan klasifikasi sebanyak 2 (2,86%).

4.3 Hasil pengujian deteksi plat nomor berdasarkan jarak, sudut dan resolusi

Pengujian berdasarkan jarak dilakukan dengan mengambil sampel gambar melalui berbagai jarak yaitu 30 cm, 50 cm, 150 cm, 300 cm, 450 cm dan 600 cm dengan resolusi setiap gambar sebesar 2560 x 1920 piksel. Hasil pengujian menunjukkan jarak pengambilan gambar terdekat yang dapat dibaca dengan baik adalah 30 cm, sedangkan jarak pengambilan gambar terjauh yang dapat dideteksi dengan baik adalah 450 cm.

Pengujian berdasarkan sudut dengan menggunakan sampel gambar yang diambil dengan kamera melalui berbagai arah dan sudut yaitu 0°, 20°, 40°, 50° dan 65°. Hasil pengujian menunjukkan

sudut pengambilan gambar yang dapat diproses dengan baik adalah  $0^{\circ}$  sampai  $40^{\circ}$  dengan arah menyamping, kemudian dari arah bawah dan atas dengan sudut  $40^{\circ}$  juga dapat diproses dengan baik. Pada sudut  $50^{\circ}$ , objek masih dapat terdeteksi tetapi tidak sempurna sedangkan mulai sudut  $65^{\circ}$  ke atas plat nomor tidak dapat terdeteksi.

Pengujian berdasarkan resolusi menggunakan sampel dengan ukuran resolusi yang berbeda-beda yaitu (2560 x 1920 piksel), (1000 x 750 piksel), (500 x 375 piksel). Gambar diambil dari jarak  $\pm 50$  cm, resolusi maksimal gambar adalah 2560 x 1920 piksel sesuai dengan kemampuan maksimal dari kamera. Hasil pengujian menunjukkan resolusi maksimal (2560 x 1920 piksel) masih dapat dideteksi dengan baik lokasi plat nomor, sedangkan resolusi terendah yang dapat dideteksi adalah 250 x 188 piksel.

## 5. Kesimpulan

Berdasarkan hasil penelitian dan analisa hasil penelitian dapat diambil beberapa kesimpulan yaitu :

1. Penentuan jumlah blok citra berpengaruh pada hasil pengujian. Jumlah blok yang diuji adalah 20, 30, 40, 50 dan 60. hasil penelitian menunjukkan akurasi terbaik pada 50 blok yaitu 93.47%.
2. Penggunaan Fungsi Kernel *polynomial* memiliki akurasi terbaik dibanding dengan kernel *linear* dan *rbf* yaitu 93.89%.
3. Pengujian plat nomor melalui tiga proses (deteksi, segmentasi dan klasifikasi), hasil pengujian didapatkan akurasi sebesar 82.86%, sedangkan 8.57% kegagalan pada proses deteksi objek, 7.14% kegagalan pada proses segmentasi dan 2.86% kegagalan pada proses klasifikasi.
4. Deteksi lokasi plat mampu bekerja pada gambar yang diambil dari jarak 30 cm sampai 450 cm, pada gambar yang diambil dari arah depan dengan sudut  $0^{\circ}$  sampai arah samping dengan sudut  $\pm 40^{\circ}$ , juga dari arah bawah dan atas dengan sudut  $40^{\circ}$ , dan pada gambar dengan resolusi 2560 x 1920 piksel sampai 250 x 188 piksel.
5. SVM mampu membedakan citra karakter dan citra bukan karakter, oleh karena itu kesalahan dalam segmentasi yang memasukkan citra bukan karakter tidak akan diklasifikasi oleh SVM sehingga hasil akhir pembacaan plat nomor tetap akurat.

## Daftar Pustaka

- Chang, C., dan Lin, Chih-Jen, 2011, A Library for Support Vector Machines, *ACM Transactions on Intelligent Systems and Technology*, 2:27:1-27:27.
- Cheng, Z.Z., 2012, Combining SVM and CHMM Classifiers For Porno Video Recognition, *The Journal of China Universities of Posts and Telecommunications*, 19(3): 100–106.
- Christianini, N., dan Taylor, J. S., 2000, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, USA.
- Du, H., Liu, Z., Jiang, J., Shen, L., 2013, Stretchability-aware block scaling for image retargeting, *Journal of Visual Communication and Image Representation*, 4, 499–508.
- Hadi, S., 2014, *Sistem Pengenalan Plat Nomor Kendaraan dengan Thinning dan Chain Code*, Thesis Magister Sistem Informasi UNDIP, Semarang.
- He, Z.F., Sun, Z.A., Tan, T.N., Wei, Z.S., 2009a. *Efficient Iris Spoof Detection Via Boosted Local Binary Patterns*. In: Tistarelli, M., Nixon, M.S. (Eds.), *Advances in Biometrics, Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1080–1090.
- Hsu, C.W., dan Lin, C.J., 2002, A Comparison of Methods for Multi-class Support Vector Machines. *IEEE Transactions on Neural Networks*, 13(2):415-425
- Jiedi, S., Qiyang, X., Jiangtao, W., Fei, W., 2014, Natural Gas Pipeline Small Leakage Feature Extraction And Recognition Based On LMD Envelope Spectrum Entropy and SVM, *Measurement*, 55, 434-443.
- Sinha, G., Rani, A., Dhir, R., Rani, R., 2012, Zone-Based Feature Extraction Techniques and SVM for Handwritten Gurmukhi Character Recognition, *International Journal of Advanced Research in Computer Science and Software Engineering*, 2, 106-111.
- Viola, P., dan Jones, M.J., 2004, Robust Real-Time Face Detection, *International Journal of Computer Vision*, 57(2), 137–154.
- Wang, W., Q.J. Jiang, X. Zhou, 2011, Car License Plate Detection Based On MSER, in: *Proceedings of the International Conference on Consumer Electronics, Communications and Networks*, 3973–3976.

- Wang, R., Sang, N., Huang, R. Wang, Y. 2013, License Plate Detection Using Gradient Information And Cascade Detectors, *Optik*, 125, 186– 190.
- Yoav, F dan Robert, E. Schapire, 1999, A Short Introduction to Boosting, *Journal of Japanese Society for Artificial Intelligence* 14(5):771-780.
- Yuan, L., Wu, Ming C., Yi, Z., 2013, Facial Expression Recognition Based On Fusion Feature of PCA and LBP with SVM, *Optik* 124, 2767-2770.
- Zhao, Y., Yuan, Y., S.B. Bai, dkk., 2011, Voting-Based License Plate Location, in: *Proceedings of the 14th International Conference on Intelligent Transportation Systems*, 314–317.
- Zheng, D., Y. Zhao, J. Wang, 2005, An Efficient Method Of License Plate Location, *Pattern Recognition Letters*, 26 (15) 2431–2438.
- Zhu, W.G. , G.J. Hou, J. Xing, 2002, A Study Of Locating Vehicle License Plate Based On Color Feature And Mathematical Morphology., in: *Proceedings of the 6th International Conference on Signal Processing*, 748–751.