

Deteksi Ketersediaan Slot Parkir Berbasis Pengolahan Citra Digital Menggunakan *Metode Histogram of Oriented Gradients* dan *Support Vector Machine*

Aditya Riska Putra*¹, Ika Candradewi²

¹Program Studi Elektronika dan Instrumentasi, FMIPA UGM, Yogyakarta

²Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: *¹aditya.riska.putra@gmail.com, ²ika.candradewi@ugm.ac.id

Abstrak

Penelitian ini bertujuan untuk mengimplementasikan metode berbasis pengolahan citra digital untuk memproses status slot parkir pada area parkir mobil dengan menggunakan metode ekstraksi ciri HOG (*Histogram of Oriented Gradients*) pada setiap region area parkir. Hasil ekstraksi ciri diklasifikasikan menggunakan SVM (*Support Vector Machine*) dengan membandingkan kernel Linear, RBF (*Radial Basis Function*), Poly, dan Sigmoid. Hasil klasifikasi SVM dianalisis menggunakan *confusion matrix* melalui parameter: *accuracy*, *specificity*, *sensitivity*, dan *precision*.

Pada pengujian data utama dengan kernel Linear didapatkan akurasi kondisi sunny 98.0%, rainy 98.8%, dan cloudy 99.2%. Pengujian kernel Poly didapatkan akurasi kondisi sunny 99.2%, rainy 98.9%, dan cloudy 99.4%. Pengujian RBF kernel didapatkan akurasi kondisi sunny 97.9%, rainy 98.7%, dan cloudy 99.6%. Pada pengujian data tambahan, didapatkan akurasi menggunakan kernel Linear sebesar 97.7%, kernel RBF sebesar 97.9%, dan kernel Poly sebesar 97.4%. Pengujian kernel Sigmoid menggunakan default grid tidak didapatkan model yang optimal melalui validasi sehingga tidak dapat digunakan dalam pengujian.

Kata Kunci—slot parkir, ekstraksi fitur, histogram of oriented gradient, klasifikasi, support vector machine, kernel.

Abstract

This research aims to implement method based on digital image processing to inform the status of parking slots at the car parking area by using a feature extraction HOG (*Histogram of Oriented Gradients*) method in every region of the parking area. Feature extraction results are classified using SVM (*Support Vector Machine*) by comparing the Linear, RBF (*Radial Basis Function*), Poly, and Sigmoid kernels. SVM classification results were analyzed using the *confusion matrix* with *accuracy*, *specificity*, *sensitivity*, and *precision* parameters.

In terms of accuracy, system obtained with Linear kernel in sunny conditions shows 98.0% accuracy; rainy 98.8% accuracy; cloudy 99.2% accuracy. Obtained accuracy using Poly kernel test in sunny conditions shows 99.2%; rainy 98.9%; cloudy 99.4%. Obtained accuracy using RBF kernel in sunny conditions shows 97.9%; rainy 98.7%; cloudy 99.6%. In terms of accuracy using additional data testing obtained with Linear kernel shows accuracy of 97.7%; RBF kernel 97.9% accuracy; Poly kernel 97.4% accuracy. Sigmoid kernel testing can't be used because the optimal model did not obtained by using default grid.

Keyword— parking slot, feature extraction, histogram of oriented gradient, classification, support vector machine, kernel.

1. PENDAHULUAN

Sistem informasi mengenai dinamika area parkir mobil sangat dibutuhkan bagi para pengendara untuk dapat mengetahui lokasi parkir yang masih kosong sehingga pengendara mobil tidak harus berkeliling dan banyak membuang waktu. Beberapa solusi untuk mengatasi masalah tersebut adalah dengan menggunakan alat khusus yang bekerja atas dasar prinsip jaringan sesor (*sensor network*), dimana sensor diletakkan pada setiap slot parkir untuk mendeteksi keberadaan mobil pada slot parkir, seperti yang telah dibuat dan diproduksi oleh ST Elektronik dan Mesh Netic[1]. Dari hasil analisis beberapa metode pendeteksian kendaraan pada sistem parkir mengenai penggunaan *inductive loop*, sensor ultrasonik, infra merah, dan *computer vision*, memiliki kesimpulan bahwa teknologi *computer vision* memiliki banyak keuntungan diantaranya: praktis, ekonomis, akurat dan cerdas[2].

Adapun beberapa penelitian mengenai sistem deketsi slot parkir menggunakan teknik pengolahan citra digital telah dilakukan dengan berbagai metode berbeda diantaranya *motion tracking*[1], pemodelan warna[2][3], penggunaan fitur lokal[4], fitur global [5]. Dengan mengacu pada penelitian sebelumnya maka penulis akan mengimplementasikan metode ekstraksi ciri HOG (*Histogram of Oriented Gradient*) dan mekanisme klasifikasi ketersediaan slot parkir menggunakan SVM (*Support Vector Machine*). Fokus pengujian ini adalah mengetahui performa sistem dengan memvariasikan parameter terkait metode HOG dan SVM yang digunakan.

2. METODE PENELITIAN

2.1 Analisis Sistem

Pada penelitian ini, peralatan yang digunakan berupa komponen hardware dan software. Untuk hardware digunakan komputer sebagai pengolah data, sedangkan *software*/perangkat lunak berupa pustaka/*library* OpenCV 2.4.13 dan Visual Studio 2013 dengan pemrograman C++. Spesifikasi *hardware* yang digunakan berupa laptop dengan OS berupa Windows 8.1, RAM sebesar 4 Gigabyte, NVIDIA Geforce GT 720M 2 GB VRAM, Processor Intel i3-3217U (1,8Ghz, 3MB L3 Cache).

Data penelitian terdiri dari data utama dan data tambahan. Data utama diambil dari citra dataset *Pontifical Catholic University of Parana (PUCPR)*[6], sedangkan data tambahan diambil dari citra parkir mobil Fakultas Matematika dan Ilmu pengetahuan Alam Universitas Gadjah Mada Yogyakarta. Citra uji utama terdiri dari tiga kondisi berbeda yaitu *sunny*, *rainy*, dan *cloudy*, dimana pada masing-masing kondisi terdapat 26 citra uji, sedangkan citra uji tambahan terdiri dari 43 citra. Data pelatihan disusun dengan menggunakan data utama[6] yang telah dipisahkan dari citra uji sebanyak 3340 citra positif dan 7940 citra negatif.

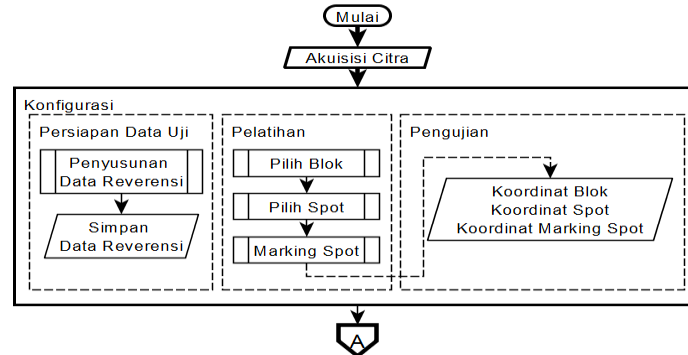
Metode HOG sebagai metode ekstraksi ciri global dipilih karena metode ini terbukti *robust* dan digunakan dalam berbagai penelitian yaitu *tracking* objek[7]. Selain itu metode HOG memiliki mekanisme normalisasi gradient vektor sehingga membuat metode tersebut lebih *invariant* (setidaknya lebih *robust*) terhadap perubahan kontras citra[8]. SVM sebagai mekanisme klasifikasi dipilih karena SVM selain meminimalkan *error* pada set-pembelajaran, juga memiliki strategi yang disebut SRM (*Structural Risk Minimization*), dimana diwujudkan dengan memilih *hyperplane* dengan *margin* terbesar, sehingga memiliki generalisasi data yang baik terhadap data yang belum pernah dijumpai sebelumnya[9].

2.2 Rancangan Arsitektur Sistem

Implementasi sistem terdiri dari tahap konfigurasi, ekstraksi ciri HOG, dan klasifikasi SVM. Pada tahap konfigurasi dilakukan proses pemilihan blok pengamatan, pemilihan spot parkir mobil/ROI (*Region of Interest*), penandaan/*marking* spot pada citra uji, dan pembentukan data reverensi. Proses ekstraksi ciri HOG pada tahap pelatihan berfungsi untuk menghimpun fitur citra yang digunakan pada tahap pelatihan model SVM, sedangkan ekstraksi ciri pada tahap pengujian berfungsi mengesktrak ciri citra sebagai data input pengujian. Klasifikasi SVM

diawali dengan melatih model SVM untuk mendapatkan model optimal melalui *k-fold cross validation*[10] dan melakukan pengujian menggunakan model SVM optimal yang didapatkan.

2.2.1 Tahap Konfigurasi



Gambar 1 Diagram alur tahap konfigurasi

Tahap konfigurasi merupakan tahap yang dijalankan satu kali proses dimana pada tahap ini output proses berupa koordinat blok, spot, dan data aktual citra uji, yang dapat dilihat pada Gambar 1. Pemilihan blok dilakukan pada citra uji untuk melokalisir area parkir mobil dengan melakukan transformasi perspektif/geometri. Transformasi perspektif dilakukan dengan tujuan membuat citra blok lebih perpendikular dan mengkompensasi dampak perspektif akibat sudut pengambilan citra. Hasil transformasi perspektif dapat dilihat pada gambar 2.

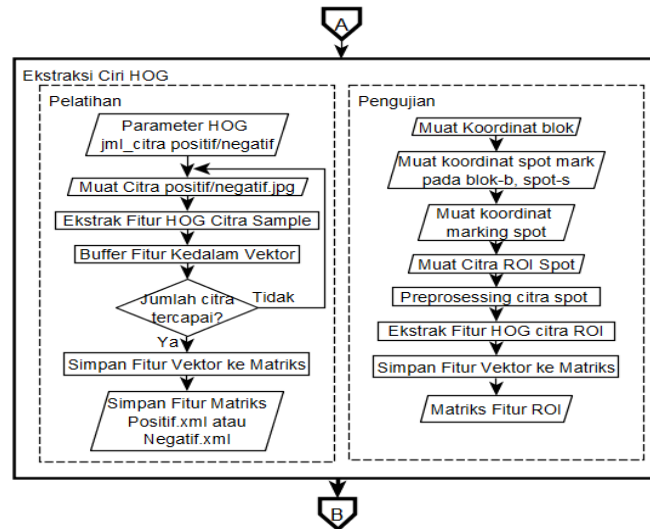


Gambar 2 Transformasi perspektif/geometri

Pemilihan spot parkir melalui *rectangle ROI*, dilakukan pada citra blok hasil transformasi perspektif. Data reverensi pada tahap konfigurasi digunakan sebagai data pembandingan yang akan dibandingkan dengan data hasil uji sistem untuk kemudian dianalisis melalui *confusion matriks*[11].

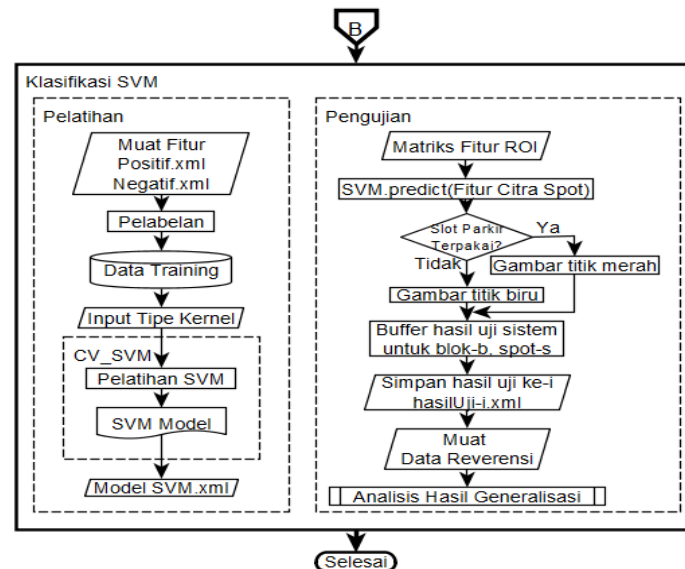
2.2.2 Tahap Ekstraksi Ciri

Ekstraksi ciri dilakukan untuk mengekstrak ciri dan membentuk data latih pada tahap pelatihan. Ekstraksi ciri pada tahap pengujian dilakukan pada citra ROI/slot parkir untuk kemudian digunakan sebagai data input pengujian. Pada tahap pelatihan ekstraksi ciri dilakukan secara bergantian antara data positif dan negatif. Pada penelitian ini proses ekstraksi ciri dilakukan dengan memvariasikan jumlah bin dari nilai 1 hingga 9 dan blok *stride* 8x8 dan 16x16 piksel. Ukuran *detection window*, blok, dan sel dibuat tetap pada nilai 64x128, 16x16, dan 8x8 piksel. Tahap ekstraksi ciri dapat dilihat pada Gambar 3.



Gambar 3 Diagram alur ekstraksi cir HOG

2.2.3 Klasifikasi SVM



Gambar 4 Diagram alur klasifikasi SVM

Sebelum klasifikasi dilakukan dengan menggunakan model SVM, model SVM perlu dilakukan pelatihan menggunakan data HOG yang telah terbentuk pada proses ekstraksi ciri citra sampel. Model SVM pada tahap pelatihan dilakukan *cross validation* untuk mengetahui nilai validitasnya. Pada tahap pelatihan model dan pengujian klasifikasi SVM, dilakukan perbandingan performa sistem menggunakan kernel Linear, RBF, Poly, dan Sigmoid. Variasi parameter HOG dan pengujian menggunakan Linear kernel SVM digunakan sebagai data acuan atau data pembanding terhadap kernel SVM lainnya. Tahap pelatihan dan pengujian klasifikasi SVM dapat dilihat ada Gambar 4.

3. HASIL DAN PEMBAHASAN

Tahap pengujian merupakan tahap akhir yang digunakan untuk mengetahui performa atau kinerja sistem. Pengujian dilakukan dengan melakukan pelatihan secara berulang-ulang fitur HOG pada SVM untuk mendapatkan parameter optimal dalam lingkup pengujian.

3.1 Pengujian Peletakan ROI Spot Parkir

Peletakan ROI pada citra area parkir PUCPR dilakukan dengan cara estimasi mengingat dampak perspektif terjadi pada area parkir yang terletak jauh dari kamera. Estimasi dilakukan dengan menggunakan citra reverensi area parkir yaitu ketika semua slot parkir mobil terpakai, sehingga peletakan ROI didasarkan atas lokasi mobil. Hasil uji *cropping* dengan reverensi lokasi mobil dapat dilihat pada gambar 6. Pada kondisi dimana ROI diletakan berdasarkan acuan garis pembatas parkir, hasil pengujian akuisisi citra dalam cakupan ROI pada blok yang terletak jauh dari kamera menunjukkan bahwa objek mobil tidak ter-*cover* dengan baik. Peletakan ROI dengan acuan garis parkir dilakukan pada citra area parkir PUCPR yang masih kosong. Berikut adalah citra area parkir dataset PUCPR blok 1.



Gambar 5 Peletakan ROI pada blok pengamatan dengan acuan garis parkir dan pengujian (PUCPR)

Terlihat pada Gambar 5 pengujian *cropping* citra ROI, koordinat yang didasarkan pada garis parkir tidak dapat melingkupi objek mobil dengan sempurna.

Keberadaan mobil dalam setiap slot parkir membantu dalam memperkirakan peletakan ROI pada masing-masing spot parkir.



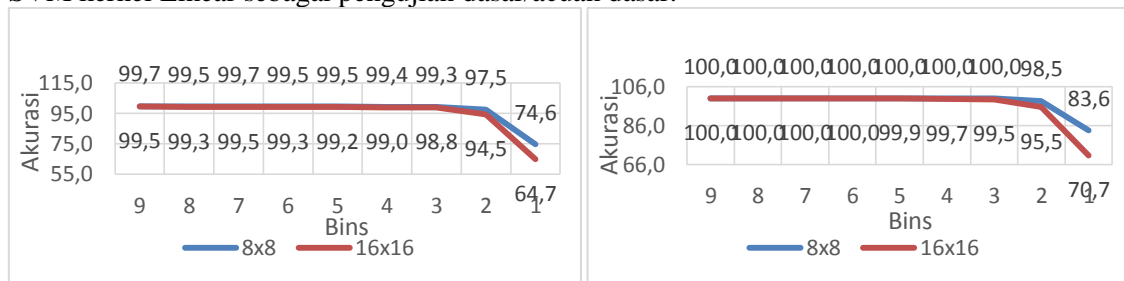
Gambar 6 Peletakan ROI pada blok pengamatan dengan acuan lokasi mobil dan pengujian (PUCPR)

3.2 Pengujian Data PUCPR

Pengujian pertama dilakkan dengan data utama PUCPR dimana antara data latih dan data uji telah dilakukan pemisahan data.

3.2.1 Pengujian Fitur HOG Menggunakan SVM kernel Linear

Pada pengujian ini, nilai bin divariasikan dari angka 9 hingga 1. Selain jumlah bin, ukuran pergeseran blok (*Block Stride*) dilakukan variasi dengan ukuran 8x8 serta 16x16 piksel dimana kedua variasi tersebut dilakukan guna mengetahui pengaruh jumlah bin dan pergeseran blok terhadap jumlah fitur yang dapat diekstrak serta performa sistem pada tahap pengujian SVM kernel Linear sebagai pengujian dasar/acuan dasar.



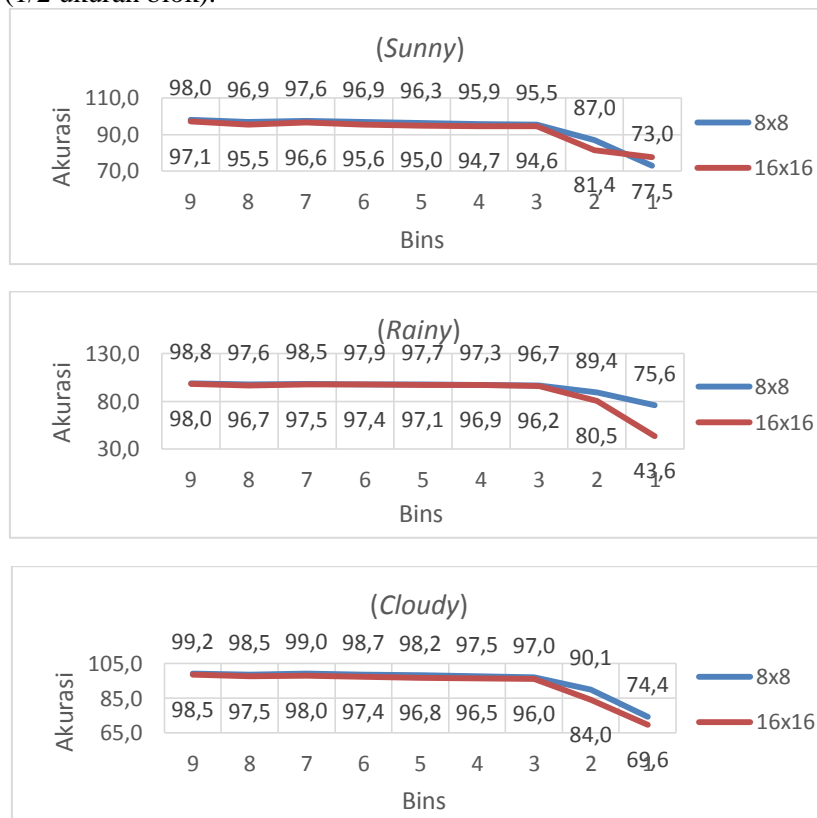
Gambar 7 Grafik validasi model SVM dengan *cross validation* (a) dan dengan data latih (b)

Sebelum pengujian dilakukan, model SVM dilakukan pelatihan berulang untuk masing-masing konfigurasi menggunakan Linear SVM. Setelah dilakukan pelatihan, model SVM yang

terbentuk dilakukan validasi melalui mekanisme *5-fold cross validation* dan pengujian dengan data latih secara langsung. Terlihat bahwa akurasi pada Gambar 7 (a) dan (b) berbeda dan cenderung lebih tinggi pada mekanisme validasi dengan data latih (*self validation*). Hal ini terjadi karena pada mekanisme *cross validation*, dataset dibagi ke dalam subset pelatihan dan pengujian dimana data pengujian pada masing-masing subset tidak digunakan dalam pelatihan model SVM. Pada mekanisme *self validation*, data pengujian dan data pelatihan tidak dilakukan pemisahan, sehingga data pengujian merupakan data yang pernah dijumpai pada tahap pelatihan. Setelah masing-masing model terbentuk pada setiap konfigurasi, maka tahap selanjutnya adalah melakukan pengujian. Hasil pengujian dapat dilihat pada Gambar 8.

Dari hasil pengujian pada ke-tiga kondisi berbeda tersebut, akurasi optimal dicapai pada bins dengan nilai 9 dengan pergeseran blok sebesar 8x8 piksel. Akurasi sistem secara umum semakin turun dengan berkurangnya nilai bins serta secara signifikan mulai turun pada konfigurasi bins 2 dan 1. Bin pada ekstraksi ciri HOG, berfungsi dalam melakukan kuantisasi arah *magnitude* gradien piksel kedalam histogram, dimana dengan semakin berkurangnya bins pada konfigurasi pengujian, jumlah fitur yang berhasil diekstrak dari suatu citra akan semakin berkurang dan berdampak pada turunya performa sistem.

Selain jumlah bins, blok *stride* atau pergeseran blok juga berdampak pada jumlah fitur yang dapat diekstrak. Pada pemilihan ukuran blok *stride* 8x8 piksel serta ukuran blok 16x16 piksel, terdapat area *overlapping* normalisasi pada tahap ekstraksi sebesar 8 piksel pada masing-masing blok (1/2 ukuran blok).



Gambar 8 Grafik akurasi variasi bin dan blok stride kondisi *sunny*, *rainy*, dan *cloudy*

Berbeda halnya ketika blok *stride* berukuran 16x16 piksel serta ukuran blok 16x16 piksel, konfigurasi tersebut tidak memungkinkan adanya *overlapping* normalisasi antar blok. Sehingga fitur yang berhasil diekstrak pada konfigurasi blok *stride* 8x8 piksel lebih besar daripada ukuran 16x16 piksel. Dari hasil pengujian menunjukkan bahwa akurasi blok *stride* 8 x 8 piksel lebih tinggi daripada blok *stride* 16 x 16 piksel. Rangkuman pengujian dapat dilihat pada Tabel 1.

Tabel 1 *Confusion matriks* pengujian bins 9, blok *stride* 8 x 8 SVM Linear

<i>Confusion Matriks</i>	<i>Sunny</i>	<i>Rainy</i>	<i>Cloudy</i>
<i>True Positive (TP)</i>	1.463	3.765	2.013
<i>False Negative (FN)</i>	37	48	38
<i>True Negative (TN)</i>	5.035	2.787	4.563
<i>False Positive (FP)</i>	95	30	16
<i>Sensitifity/Recall/TP Rate</i>	97,5 %	98,7 %	98,1 %
<i>PPV/ Precision</i>	93,9 %	99,2 %	99,2 %
<i>Specificity</i>	98,1 %	98,9 %	99,67 %
Akurasi	98,0 %	98,8 %	99,2 %
<i>Error</i>	2,0 %	1,2 %	0,8 %

Terlihat bahwa akurasi optimal dicapai pada data *cloudy*, dimana pada kondisi *cloudy* akurasi mencapai 99.2%. Pada data *rainy*, akurasi yang didapatkan lebih tinggi dibanding data *sunny*, yaitu 98.8%, sehingga data *sunny* memiliki akurasi terendah dibanding kedua data tersebut.

3.2.2 Pengujian SVM Kernel Poly

Pada pengujian SVM Poly kernel, menggunakan data optimal HOG yang sudah didapatkan pada pengujian SVM Linear yaitu dengan konfigurasi jumlah bin 9 dan blok *stride* 8x8. Terdapat tiga buah parameter yang perlu dilakukan *setting/tunning* atau pemilihan guna mendapatkan model SVM Poly kernel. Ketiga parameter SVM Poly kernel tersebut adalah *gamma*, *degree*, dan *coef0*. Ketiga nilai tersebut divariasikan menggunakan *default grid* OpenCV. Berikut adalah variasi nilai untuk ketiga nilai tersebut:

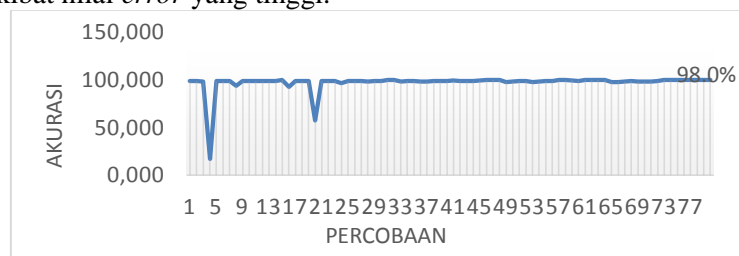
gamma grid : 0.00005, 0.00015, 0.00225, 0.03375, 0.50625.

degree grid : 0.01, 0.07, 0.49, 3.43

coef0 grid : 0.1, 1.4, 19.6, 274.4

Ketiga nilai parameter *gamma*, *degree*, dan *coef0* kemudian dikombinasikan dalam pengujian berulang sehingga terdapat 80 kombinasi percobaan. Grafik pengujian *cross validation* kernel Poly dapat dilihat pada Gambar 9.

Dari 80 kombinasi percobaan *default grid* Poly kernel adalah 99,8%, dengan nilai *gamma* = 0.5062, *degree* = 3.43, dan *coef0* = 0,1. Terdapat dua buah pengujian yang menghasilkan akurasi buruk yaitu ditandai dengan pergerakan turun tajam grafik pengujian. Pergerakan turun tersebut diakibatkan oleh besarnya nilai *error* pengujian. *Error* pengujian terjadi pada pengujian nomor 4 dan 20. Pada pengujian model nomor 4 SVM Poly kernel, didapatkan nilai *error* sebesar 83.0% serta akurasi sebesar 17.0% dengan parameter *gamma* = 0.00005, *degree* = 0.01, dan *coef0* = 274,4. Pada pengujian nomor 20, didapatkan akurasi sebesar 43.0% serta akurasi sebesar 57.0% dengan parameter *gamma* = 0.00015, *degree* = 0.01, dan *coef0* = 274,4. Hal tersebut menunjukkan bahwa dengan nilai parameter tersebut, *hyperplane* sebagai bidang pemisah fitur SVM dalam *feature space* tidak dapat diletakkan secara sempurna memisahkan kedua buah kelas, sehingga kesalahan klasifikasi banyak yang terjadi dan berakibat nilai *error* yang tinggi.



Gambar 9 Akurasi optimal pengujian parameter kernel Poly

Dari model SVM optimal yang telah terbentuk dengan kombinasi *gamma*, *degree*, dan *coef0*, kemudian digunakan untuk melakukan pengujian data uji pada ketiga kondisi berbeda

yaitu *sunny*, *rainy*, dan *cloudy*. Tabel 2 berikut adalah hasil pengujian yang didapatkan dengan kernel Poly.

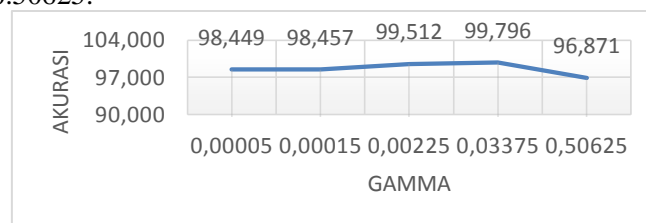
Tabel 2 *Confusion matriks* pengujian bins 9, blok *stride* 8 x 8 SVM Poly

Confusion Matriks	Sunny	Rainy	Cloudy
<i>True Positive (TP)</i>	1.470	3.748	2.013
<i>False Negative (FN)</i>	30	65	38
<i>True Negative (TN)</i>	5.109	2.811	4.578
<i>False Positive (FP)</i>	21	6	1
<i>Sensitifity/Recall/TP Rate</i>	98,0 %	98,3 %	98,1 %
<i>PPV/ Precision</i>	98,6 %	99,8 %	100,0 %
<i>Specificity</i>	99,6 %	99,8%	100,0 %
Akurasi	99,2 %	98,9 %	99,4 %
Error	0,8 %	1,1 %	0,6 %

Akurasi data *cloudy* pada Tabel 2 terlihat lebih tinggi dibandingkan data *sunny* dan *rainy* yaitu 99.4%. Namun pada pengujian SVM Poly terlihat bahwa data *sunny* memiliki nilai akurasi yang lebih tinggi yaitu 99.2% dibanding *rainy* dengan akurasi 98.9%.

3.2.3 Pengujian SVM Kernel RBF

Parameter yang berkaitan dengan kernel RBF adalah *gamma*. adalah: 0.00005, 0.00015, 0.00225, 0.03375, 0.50625.



Gambar 10 Gamma optimal pada kernel RBF

Dari grafik pengujian *cross validation* kernel RBF Gambar 10, akurasi optimal pengujian model, jatuh pada nilai *gamma* 0.03375 dengan akurasi sebesar 99,8%. Nilai *gamma* optimal dari pengujian *default grid* dan *train auto* sama yaitu 0.03375. Dengan model optimal yang sudah terbentuk kemudian digunakan untuk melakukan pengujian terhadap data uji dengan tiga kondisi berbeda yaitu *sunny*, *rainy*, dan *cloudy* yang dapat dilihat pada Tabel 3.

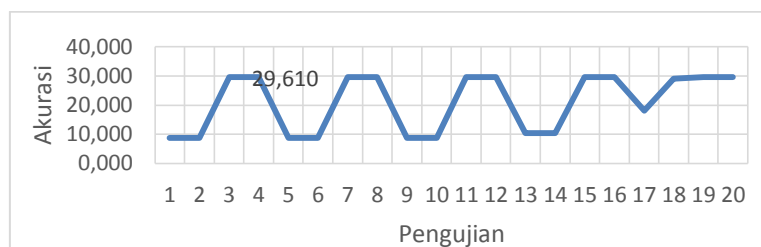
Tabel 3 *Confusion matriks* pengujian bins 9, blok *stride* 8 x 8 SVM RBF

Confusion Matriks	Sunny	Rainy	Cloudy
<i>True Positive (TP)</i>	1.489	3.775	2.037
<i>False Negative (FN)</i>	11	38	14
<i>True Negative (TN)</i>	5.002	2.766	4.564
<i>False Positive (FP)</i>	128	51	15
<i>Sensitifity/Recall/TP Rate</i>	99,3 %	99,0 %	99,3 %
<i>PPV/ Precision</i>	92,1 %	98,7 %	99,3 %
<i>Specificity</i>	97,5 %	98,2 %	99,7 %
Akurasi	97,9 %	98,7 %	99,6 %
Error	2,1 %	1,3 %	0,4 %

Pada Tabel 3 kembali terlihat bahwa data *cloudy* menghasilkan akurasi yang lebih tinggi dibandingkan data *sunny* dan *rainy* yaitu 99.6%. Begitu pula data *rainy* bila dibandingkan dengan data *sunny*, data *rainy* menghasilkan akurasi yang lebih tinggi yaitu 98.6% bila dibanding data *sunny* dengan akurasi sebesar 97.9%.

3.2.4 Pengujian SVM Kernel Sigmoid

Kernel Sigmoid memiliki dua parameter utama yaitu γ dan coef0 . Seperti pada pengujian SVM kernel RBF dan Poly, *default grid gamma* adalah: 0.00005, 0.00015, 0.00225, 0.03375, 0.50625, sedangkan *default grid coef0* adalah: 0.1, 1.4, 19.6, 274.4. Kedua parameter Sigmoid kernel tersebut, kemudian dikombinasikan kedalam 20 pasangan nilai γ dan coef0 untuk diujicobakan ke dalam algoritma pelatihan SVM dan pengujian *5 fold cross validation* sebagai upaya mengetahui nilai akurasinya. Dari hasil pengujian *cross validation* didapatkan hasil sebagai berikut:



Gambar 11 Akurasi optimal pengujian parameter Sigmoid Kernel

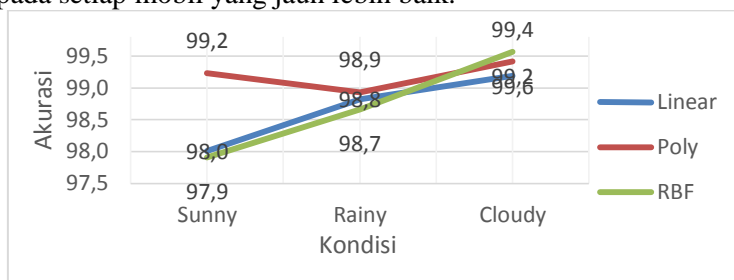
Dari grafik pengujian *cross validation* kernel Sigmoid Gambar 11, dapat dilihat bahwa nilai akurasi validasi model optimum menggunakan *default grid* adalah 29.6% dan dicapai pada pengujian ketiga dengan kombinasi nilai γ dan coef0 yaitu 0.00005 dan 19.6, dimana parameter tersebut sesuai dengan hasil *train auto* pada algoritma OpenCV. Hasil pengujian *default grid* menunjukkan bahwa kernel Sigmoid tidak dapat digunakan dalam pengujian selanjutnya karena menghasilkan akurasi yang rendah dalam taraf model. Sehingga kernel Sigmoid tidak cocok digunakan untuk membentuk model dengan data yang dimiliki. Hal tersebut menunjukkan bahwa dalam rentang *default grid* yang ada terhadap nilai γ dan coef0 , model optimal tidak tercapai dikarenakan algoritma SVM gagal dalam meletakkan *hyperplane* atau pemisah terbaik terhadap data yang dimiliki. Kesalahan klasifikasi terjadi cukup besar sehingga mengakibatkan tingginya nilai *error* dan turunya nilai akurasi. Hal tersebut ditunjukkan dengan pergerakan naik turun nilai akurasi pada grafik Gambar 11 yang hanya maksimal pada nilai 29.6%.

Secara umum Sigmoid kernel tidak dapat menghasilkan hasil optimal jika dibandingkan dengan RBF kernel. Dijelaskan pula bahwa kernel Sigmoid dapat berpotensi tidak PSD (*positive semi-definite*) terhadap pemilihan nilai γ dan coef0 . Dampak penggunaan kernel yang bersifat Non-PSD jika dipandang dari segi pemisahan data adalah pada masalah optimasi konstrain, dimana solusi *Lagrangian primal problem* tidak dapat dilakukan transfer kedalam *dual problem* atau hubungan antar keduanya tidak ada. Dengan demikian bidang *hyperplane*/pemisah kelas data tidak dapat diletakkan secara optimal untuk memisahkan data fitur kelas yang ada dan berdampak pada tingginya nilai *error*[12]. Syarat sebuah fungsi untuk dapat menjadi fungsi kernel adalah memenuhi teorema Mercer yang menyatakan bahwa matriks kernel yang dihasilkan harus bersifat *positive semi-definite*[9].

Kernel RBF dan Sigmoid memiliki kesamaan sifat untuk nilai $\gamma > 0$ dan nilai $\text{coef0} < 0$ pada kernel RBF dengan γ kecil, namun performa tidak sebaik kernel RBF. Pada kasus dimana nilai $\gamma > 0$ dan nilai $\text{coef0} > 0$, performa yang dihasilkan tidak sebaik jika dibanding pada kombinasi $\text{coef0} < 0$. *Library* OpenCV dengan nilai negatif pada setting parameter coef0 guna menyamakan sifat dengan kernel RBF seperti yang dipaparkan[12], tidak dimungkinkan karena menghasilkan *error* pada sitaks pemrograman ketika diujicobakan, dimana pada pesan *error* tersebut menyatakan bahwa parameter coef0 harus bersifat positif.

Dari pengujian menggunakan SVM Linear, RBF, dan Poly yang telah dilakukan dengan parameter optimal, dirangkum dan didapatkan data perbandingan antar ketiga kondisi. Dari grafik Gambar 12 dapat dilihat bahwa secara umum data *cloudy* mendapatkan akurasi paling tinggi diikuti dengan data *rainy*. Hal ini terjadi karena pada data *sunny*, mobil yang berada pada masing-masing spot parkir mengalami *over expose*, dimana terdapat *flare* pada atap mobil akibat pantulan sinar matahari, sehingga menurunkan kontras pada analisis spot parkir. Pada

kondisi *rainy* dan *cloudy*, pencahayaan lebih merata atau homogen. Hal ini terlihat pada data uji dimana kontras pada setiap mobil yang jauh lebih baik.

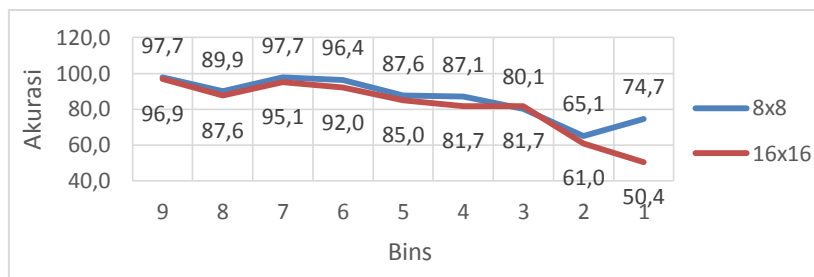


Gambar 12 Data perbandingan akurasi masing-masing kernel

3.3 Pengujian Data Suplemen (MIPA)

Pengujian sistem dengan data MIPA dilakukan dengan menggunakan skenario pengujian yang sama dengan data PUCPR serta menggunakan dataset PUCPR. Data MIPA diambil dari pukul 07.00 hingga 13.00 WIB dengan jumlah citra sebanyak 43 citra uji secara random dengan memperhatikan perubahan dinamika kendaraan yang ada.

Pada Gambar 13 mengenai pengujian bins dan blok *stride* data MIPA, terlihat bahwa akurasi pengujian optimal menggunakan SVM Linear dicapai pada konfigurasi jumlah bins 9 dan 7 dan blok *stride* 8 x 8 dengan akurasi sebesar 97,7% terhadap 43 citra data uji. Kesamaan akurasi pada bin 9 dan 7 dapat terjadi karena fluktuasi pengujian akibat data pengujian dan data latih menggunakan set berbeda, baik berbeda data lokasi maupun berbeda sudut pengambilan citra. Sesuai dengan teori dan pengujian yang telah dilakukan bahwa nilai bin pada metode HOG merepresentasikan kuantisasi data, dimana semakin berkurangnya nilai bin maka semakin sedikit fitur yang dapat diekstrak dari suatu citra, sehingga pengujian untuk membandingkan performa kernel SVM dilakukan menggunakan data HOG dengan konfigurasi bin 9 dan blok *stride* 8 x 8.



Gambar 13 Grafik akurasi variasi bin dan blok *stride* data MIPA

Tabel 4 berikut merupakan perbandingan pengujian menggunakan ketiga kernel tersebut:

Tabel 4 *Confusion matriks* pengujian bins 9, blok *stride* 8 x 8 (MIPA)

Confusion Matriks	Linear	Poly	RBF
True Positive (TP)	191	183	185
False Negative (FN)	1	9	7
True Negative (TN)	187	194	194
False Positive (FP)	8	1	1
Sensitifity/Recall/TP Rate	99.5 %	95.3 %	96.4 %
PPV/ Precision	96.0 %	99.5 %	99.5 %
Specificity	95.9 %	99.5 %	99.5 %
Akurasi	97.7 %	97.4 %	97.9 %
Error	2.3 %	2.6 %	2.1 %

Dari Tabel 4 dapat dilihat bahwa akurasi optimum dicapai pada kernel RBF dengan nilai akurasi sebesar 97.9% dengan akurasi terbesar kedua adalah kernel Linear dengan akurasi 97.7% serta kernel Poly dengan akurasi 97.4%.

4. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan terhadap sistem menggunakan data utama PUCPR dan data suplemen/tambahan MIPA, telah berhasil dibuat suatu sistem yang dapat mengklasifikasikan status slot parkir menggunakan metode HOG dan SVM, serta dapat diambil beberapa kesimpulan sebagai berikut:

1. Pengujian parameter HOG dan SVM Linear dengan data input PUCPR dan MIPA didapatkan akurasi optimum pada pengujian dengan jumlah bins 9 dan ukuran blok *stride* 8x8 piksel. Pengujian tersebut dilakukan dengan menggunakan ukuran *detection window*, ukuran blok, dan sel yang tetap secara berurutan yaitu 64x128, 16x16, dan 8x8 piksel.
 2. Pada pengujian kernel RBF dan Poly menggunakan data fitur HOG optimal, parameter optimum RBF dicapai dengan nilai $\gamma = 0.03375$. Parameter optimum Poly dicapai dengan nilai $\gamma = 0.50625$, $\text{degree} = 3.43$, dan $\text{coef0} = 0.1$. Pada pengujian kernel Sigmoid menggunakan *default grid* tidak didapatkan model yang optimal melalui validasi sehingga tidak digunakan dalam pengujian.
 3. Pada pengujian data PUCPR
 - a. Menggunakan kernel Linear didapatkan hasil sebagai berikut:
 - ✓ Kondisi *Sunny*: akurasi 98.0%, *sensitivity* 97.5%, *precision* 93.9%, *specificity* 98.1 %.
 - ✓ Kondisi *Rainy*: akurasi 98.8%, *sensitivity* 98.7%, *precision* 99.2%, *specificity* 98.9 %.
 - ✓ Kondisi *Cloudy*: akurasi 99.2%, *sensitivity* 98.1%, *precision* 99.2%, *specificity* 99.7 %.
 - b. Menggunakan kernel RBF didapatkan hasil sebagai berikut:
 - ✓ Kondisi *Sunny*: akurasi 97.9%, *sensitivity* 99.3%, *precision* 92.1%, *specificity* 97.5%.
 - ✓ Kondisi *Rainy*: akurasi 98.7%, *sensitivity* 99.0%, *precision* 98.7%, *specificity* 98.2%.
 - ✓ Kondisi *Cloudy*: akurasi 99.6%, *sensitivity* 99.3%, *precision* 99.3%, *specificity* 99.7%.
 - c. Menggunakan kernel Poly didapatkan hasil sebagai berikut:
 - ✓ Kondisi *Sunny*: akurasi 99.2%, *sensitivity* 98.0%, *precision* 98.6%, *specificity* 99.6%.
 - ✓ Kondisi *Rainy*: akurasi 98.9%, *sensitivity* 98.3%, *precision* 99.8%, *specificity* 99.8%.
 - ✓ Kondisi *Cloudy*: akurasi 99.4%, *sensitivity* 98.1%, *precision* 100.0%, *specificity* 100.0%.
- Bila dilakukan perataan nilai antara kondisi *sunny*, *rainy*, dan *cloudy* pada ketiga kernel tersebut, kernel Poly memiliki performa akurasi paling tinggi yaitu 99.191%, dibanding RBF 98.708% dan linear 98.673%.
4. Pada pengujian data suplemen (MIPA)
 - a. Menggunakan Linear kernel didapatkan akurasi 97.7%, *sensitivity* 99.5%, *precision* 96.0%, *specificity* 95.9%.
 - b. Menggunakan RBF kernel didapatkan hasil akurasi 97.9%, *sensitivity* 96.4%, *precision* 99.5%, *specificity* 99.5%.
 - c. Menggunakan Poly kernel didapatkan hasil akurasi 97.4%, *sensitivity* 95.3%, *precision* 99.5%, *specificity* 99.5%.

Untuk data MIPA, RBF kernel memiliki performa akurasi paling tinggi yaitu 97.9%.

5. SARAN

Beberapa hal yang perlu disempurnakan sehingga perlu dilakukan penelitian lebih lanjut untuk memperbaiki kualitas dan akurasi sistem dalam penelitian ini adalah:

1. Menambahkan dan memperkaya variasi data latih untuk meningkatkan akurasi sistem pada lokasi pengujian berbeda terutama variasi citra sampel negative.
2. Menggunakan analisis wilayah warna yang lebih invarian terhadap perubahan pencahayaan daripada menggunakan wilayah warna *grayscale*.

3. Perbaikan dari sisi teknik dalam penentuan lokasi slot parkir dan mengatasi *overlapping* citra pada slot parkir akibat dampak perspektif.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam mengerjakan penelitian ini.

DAFTAR PUSTAKA

- [1] Torres, F.2013. "*Automatic Parking Lot Occupancy Computation Using Motion Tracking*". Faculty of The College of Engineering and Computer Science: Atlantic University Boca Raton Florida.
- [2] Modi, P., Morellas, V., dan Papanikolopoulos, N. 2011. "*Counting Empty Parking Spots at Truck Stops Using Computer Vision*". Department of Computer Science and Engineering University of Minnesota: Whasington.
- [3] Tschentscher, M dan Neuhausen, M.2012. "*Video-based parking space detection*". Institute for Neural Computation, Ruhr-Universität Bochum.
- [4] True, N.2011. "*Vacant Parking Space Detection in Static Images*". University of California, San Diego 9500 Gilman Drive, La Jolla, CA 92093.
- [5] Almeida, P., Olivera, L., Silva,E., Britto, A., dan Koerich, A .2013. "*Parking Space Detection using Textural Descriptors*". Federal University of Paran´ a Department of Informatics Curitiba, PR, Brazil - 81531-990.IEEE International Conference on Systems, Man, and Cybernetics.
- [6] Almeida, P., Oliveira, L. S., Silva Jr, E., Britto Jr, A., Koerich, A.,2015. "*PKLot - A robust dataset for parking lot classification, Expert Systems with Applications*". Federal University of Paran´ a Department of Informatics Curitiba, PR, Brazil. 42(11):4937-4949.
- [7] Dalal, N dan Triggs, B .2005." Histograms of Oriented Gradients for Human Detection". INRIA Rhoˆne-Alps, 655 avenue de l'Europe, Montbonnot 38334, France.
- [8] Cormic,C. 2013. "*HOG Person Detector Tutorial*". Sumber: <http://mccormickml.com/2013/05/09/hog-person-detector-tutorial/>. Diakses : 7 Mei 2016.
- [9] Sembiring,K., 2007. Tutorial SVM. Sumber: <http://sutikno.blog.undip.ac.id/files/2011/11/tutorial-svm-bahasaindonesia-oleh-krisantus.pdf>, diakses : 21 September 2016.
- [10] Wihardi, Y. 2013. "K-fold Cross Validation Generator in Cpp" Sumber: <http://blog.yayaw.web.id/programming/k-fold-cross-validation-generator-in-cpp>. diakses : 12 Juni 2016.
- [11] Han, J., dan Kamber, M. 2006. Data Mining Concept and Tehniques. San Fransisco: Morgan Kauffman. ISBN 13: 978-1-55860-901-3.
- [12] Lin, H., dan Lin, C.2003."A Study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-type Methods". Department of Computer Science and Information Engineering National Taiwan University Taipei 106, Taiwan.