Deteksi Mobil Menggunakan Histogram of Oriented Gradient

Cahyo Permata,I Ketut Eddy Purnama dan Muhtadin
Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111
E-mail: ketut@ee.its.ac.id, muhtadin_s@elect-eng.its.ac.id

Sistem deteksi mobil merupakan suatu teknologi yang sangat membantu dalam berbagai bidang. Dalam bidang lalulintas, sistem deteksi mobil dipakai untuk pemantauan traffic lalulintas. Selain itu, deteksi mobil juga digunakan pada sistem parkir untuk menghitung jumlah mobil. Bebagai metode dikembangkan untuk memaksimalkan hasil deteksi. Salah satu metode yang dibahas dalam tugas akhir ini adalah metode Histogram of Oriented Gradient (HOG) yang digunakan untuk mendeteksi mobil pada image statis. Karakteristik feature HOG dari sebuah mobil ditunjukkan oleh distribusi gradien yang berupa garis vektor. hasil feature HOG dilearning menggunakan Support Vector Machine (SVM) untuk menghasilkan sebuah model yang digunakan sebagai acuan deteksi. Pada tugas akhir ini metode HOG digunakan untuk mendeteksi mobil pada image statis. Learning dilakukan dengan menggunakan 2462 data image positif (image mobil) dan 4627 data image negatif (image bukan mobil) yang didapatkan dari chenghak dataset. pengujian yang dilakukan ada dua, yaitu pengujian dengan mengubah jumlah training data dan pengujian dengan mengubah nilai threshold dari bobot feature HOG pada area yang dideteksi. hasil pengujian menunjukkan tingkat akurasi tes model dengan image tes sebesar 99,10% dengan image tes sebanyak 1028 image positif dan 1978 image negatif. Sedangkan hasil deteksi mobil menunjukkan 76,17% dari 196 image.

Kata Kunci—deteksi obyek, deteksi mobil, Histogram of Oriented Gradient, SVM.

I. PENDAHULUAN

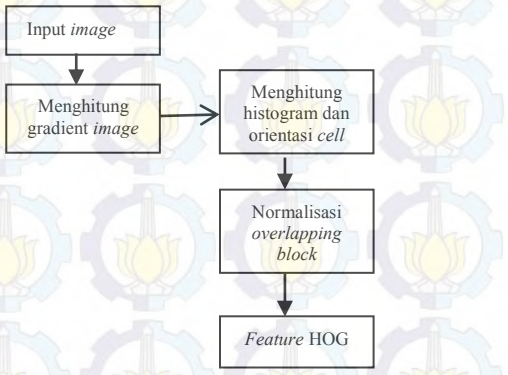
SEIRING dengan perkembangan jaman, teknologi tidak hanya dipakai dalam bidang komputer saja, namun hampir semua aspek kehidupan tidak lepas dari penggunaan teknologi. Pemantauan lalulintas, pengaturan tempat parkir merupakan sedikit dari sekian banyak bidang memanfaatkan teknologi. Dalam penerapannya, teknologi sangat membantu pekerjaan-pekerjaan diatas. Sebagai contoh adalah teknologi car counting pada sistem lalulintas dan teknologi deteksi lahan parkir yang. Berbagai metode diterapkan untuk dapat mendeteksi mobil dengan hasil yang maksimal. Salah satu metode deteksi obyek yang dibahas dalam penelitian ini adalah metode Histogram of Oriented Gradient (HOG). HOG dikembangkan oleh Navneet Dalal dan bill trigs pada tahun 2005[1]. Pengujian awal yang dilakukan Navneet Dalal dan Bill Triggs terfokus pada deteksi pejalan kaki saja. Namun dalam perkembangannya, penelitian dilakukan untuk mendeteksi obyek yang beragam. HOG merupakan metode *image processing* untuk tujuan deteksi obyek. Pada penelitian ini, metode HOG diterapkan untuk mendeteksi obyek mobil dalam suatu *image* statis. Untuk mengklasifikasikannya, dibutuhkan sebuah classifier, dalam penelitian ini classifier yang digunakan adalah *Support Vector machine* (SVM). Pada learning SVM tahapan yang dipakai ada dua, yaitu training SVM untuk menghasilkan model, dan *classify* SVM untuk *testing* model. Sebagai bahan *training* maka dibutuhkan *dataset* yang dimbil dari dataset Chenghak[2]. Dataset ini memiliki ukuran yang sama tiap *image* , yaitu 128x64 kecuali image untuk proses deteksi, yaitu berukuran 680x480. Hasil learning ini menghasilkan sebuah model yang dipakai sebagai acuan untuk mendeteksi mobil dalam suatu *image*.

II. METODE PENELITIAN

A. *Histogram of Oriented Gradient*

Histogram of Oriented Gradient (HOG) adalah sebuah metode yang digunakan dalam *image processing* untuk tujuan deteksi obyek . Teknik ini menghitung nilai gradien dalam daerah tertentu pada suatu *image* . Tiap *image* mempunyai karakteristik yang ditunjukkan oleh distribusi gradien. Karakteristik ini diperoleh dengan membagi *image* kedalam daerah kecil yang disebut *cell*. Tiap *cell* disusun sebuah histogram dari sebuah gradient. Kombinasi dari histogram ini dijadikan sebagai deskriptor yang mewakili sebuah obyek.

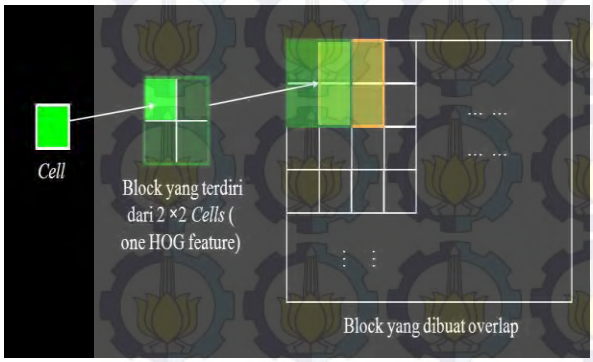
Secara keseluruhan algoritma HOG ditunjukkan pada gambar 1.



Gambar 1. Algoritma *Histogram of Oriented Gradient*

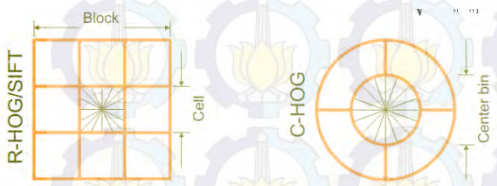
Dari gambar 1, Tahap awal dari HOG adalah menghitung nilai gradien dari input *image* . Metode yang paling umum untuk menghitung besarnya gardien adalah dengan menggunakan *sobel filter* dalam satu atau dua arah, baik secara horisontal atau vertikal. Selanjutnya adalah membuat

bagian-bagian yang dinamakan *cell*. Setiap piksel dalam sebuah *cell* mempunyai nilai histogram sendiri-sendiri berdasarkan nilai yang dihasilkan dalam perhitungan gradien. *Cell* memiliki ukuran 4x4 pixel pada sebuah *image* sedangkan block memiliki ukuran 2x2 *cell* atau 8x8 pixel.penjelasan diatas ditunjukkan pada gambar 2.



Gambar 2. Cell yang menyusun sebuah *block*

Pada gambar 3. *Cell* dalam HOG dapat berupa persegi panjang (R-HOG) atau setengah lingkaran(C-HOG). R-HOG diwakili oleh tiga parameter yaitu jumlah sel per blok, jumlah pixel per *cell*, dan jumlah *bin* per histogram. Sedangkan C-HOG memiliki empat parameter yaitu jumlah sudut dan *radial bin*, jari-jari *center bin*, dan faktor ekspansi untuk radius tambahan dari *radial bin*.

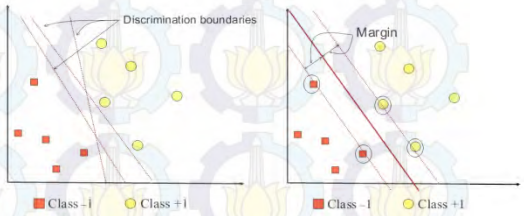


Gambar 3. Cell yang menyusun sebuah *block*[1]

Hasil *feature* HOG dirubah menjadi *feature vector* untuk diproses kedalam SVM *classifier*. SVM *classifier* digunakan untuk menemukan garis pemisah (*hyperplane*) dari class +1 yang berisi *feature vector* positif dengan class -1 yang berisi *feature vector* negatif. Setelah dilatih model hasil *training* dites untuk diketahui tingkat akurasi dalam mengenali obyek yang di tes[1].

B. *Support Vector Machine (SVM)*

Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah *class* pada input space. *Support Vector Machine* (SVM) dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan pada tahun 1992 di Annual Workshop on Computational Learning Theory[3]. Konsep dasar SVM sebenarnya merupakan kombinasi harmonis dari teori-teori komputasi yang telah ada puluhan tahun sebelumnya, seperti margin *hyperplane* (Duda & Hart tahun 1973, Cover tahun 1965,Vapnik 1964, dsb.), *kernel* diperkenalkan oleh Aronszajn tahun 1950, dan demikian juga dengan konsep-konsep pendukung yang lain. Akan tetapi hingga tahun 1992, belum pernah ada upaya merangkaikan komponen-komponen tersebut. Berbeda dengan strategi *neural network* yang berusaha mencari *hyperplane* pemisah antar *class*, SVM berusaha menemukan *hyperplane* yang terbaik pada *input space*. 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. Perkembangan ini memberikan rangsangan minat penelitian di bidang *pattern recognition* untuk investigasi potensi kemampuan SVM secara teoritis maupun dari segi aplikasi.



Gambar 4. Class dalam SVM[5]

Gambar 4a memperlihatkan beberapa *pattern* yang merupakan anggota dari dua buah *class* : +1 dan –1. *Pattern* yang tergabung pada *class* –1 disimbolkan dengan warna merah (kotak), sedangkan *pattern* pada *class* +1, disimbolkan dengan warna kuning(lingkaran). Problem klasifikasi dapat diterjemahkan dengan usaha menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut. Berbagai alternatif garis pemisah (*discrimination boundaries*) ditunjukkan pada gambar 4a. *Hyperplane* pemisah terbaik antara kedua *class* dapat ditemukan dengan mengukur *margin hyperplane* tsb. dan mencari titik maksimalnya. *Margin* adalah jarak antara h*yperplane* tersebut denga n *pattern* terdekat dari masing-masing *class*. *Pattern* yang paling dekat ini disebut sebagai *support vector*. Garis solid pada gambar 4b menunjukkan *hyperplane* yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua *class*, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah support vector. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM.

Jika x adalah vektor pada ruang vektor, make fungsi *hyperplane* dapat ditulis sebagai berikut :

$$|\overrightarrow{w}.\vec{x}_i + b| \qquad (2.1)$$

Dengan w adalah koefisien vektor *weight* dan b adalah *bias*. Sedangkan label masing-masing dinotasikan $yi \in \{-1, +1\}$ dengan $i = 1,2,3, \ldots, l$ dimana l adalah banyaknya data.

Diasumsikan kedua *class* -1 dan +1 dapat terpisah secara sempurna oleh *hyperplane*. *Pattern* $\vec{x}_i$ yang termasuk *class* -1 (sampel negatif) dapat dirumuskan sebagai *pattern* yang memenuhi formula

$$\overrightarrow{w}.\vec{x}_i + b \leq -1 \qquad (2.2)$$

Sedangkan *pattern* xi yang termasuk dalam *class* +1 (sampel positif) dapat dirumuskan

$$\overrightarrow{w}.\vec{x}_i + b \leq +1 \qquad (2.3)$$

Jarak antara vektor *training* xi dan *hyperplane* disebut margin yang didefinisikan sebagai :

$$\frac{|\vec{w}.\vec{x}_i + b|}{||\vec{w}||} \quad (2.4)$$

Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara *hyperplane* dan titik terdekatnya, yaitu $1/||\vec{w}||$. Hal ini dapat dirumuskan sebagai *Quadratic Programming (QP) problem* yaitu mencari titik minimal persamaan (2.4) dengan memperhatikan *constraint* persamaan (2.2).

$$\min_{\vec{w}} \tau(w) = 1/2\,||\vec{w}||^2 \quad (2.5)$$

$$y_i\,(\vec{x_i}.\vec{w} + b) - 1 \geq 0 \quad (2.6)$$

Problem ini dapat dipecahkan dengan berbagai teknik komputasi, diantaranya *Lagrange Multiplier*

$$L(\vec{w}, b, a) = \tfrac{1}{2}||\vec{w}||^2 - \textstyle\sum_{i=1}^{l} a_i \left(y_i\left((\vec{x_i}.\vec{w} + b) - 1 \geq 0\right)\right)$$

$$(i = 1,2,3, \dots, l) \quad (2.7)$$

a_i adalah *Langrange multipliers*, yang bernilai nol atau positif ($a_i \geq 0$). NIlai optimal dari persamaan (2.7) dapat dihitung dengan meminimalkan L terhadap $\vec{w}$ dan , dan memaksimalkan L terhadap a_i. Dengan memperhatikan sifat bahwa pada titik optimal gradient $L = 0$, persamaan (2.6) dapat dimodifikasi sebagai maksimalisasi problem yang hanya mengandung $a_i$saja, sebagaimana persamaan (2.8) dibawah ini.

Maximize :

$$\textstyle\sum_{i=1}^{l} a_i - \frac{1}{2}\sum_{i,j=1}^{l} a_i\, a_j y_i y_j \vec{x}_i.\vec{x}_j \quad (2.8)$$

Subject to :

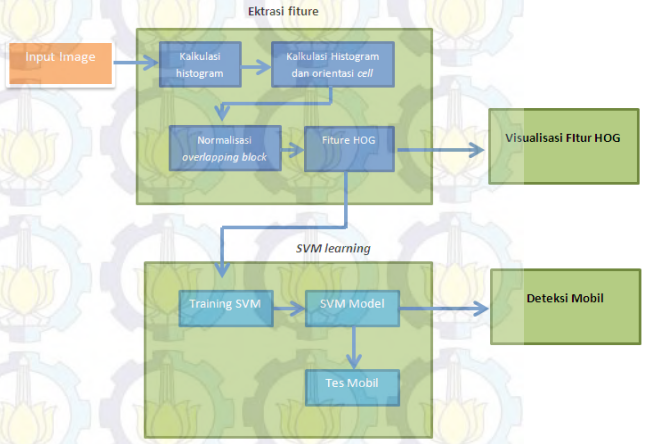
$$a_i \geq 0\ (i = 1,2,3, \dots, l)$$
$$\sum_{i=1}^{l} a_i\, y_i = 0 \quad (2.9)$$

Dari hasil perhitungan ini diperoleh a_i yang kebanyakan bernilai positif. Data yang berkorelasi dengan a_i yang positif inilah yang disebut *support vector*[.5]

C. Desain Sistem

Bagian ini merupakan Desain sistem dari deteksi mobil menggunakan *Histogram of Oriented Gradient* (HOG). Sistem ini digunakan untuk mendeteksi *image* dengan berbagai obyek yang ada didalamnya, dalam hal ini *image* yang digunakan adalah *image* statis. Input ada dua macam, yaitu input yang akan digunakan untuk *learning* SVM dan input *image* yang akan dites apakah didalam *image* tersebut terdapat obyek mobil atau tidak.

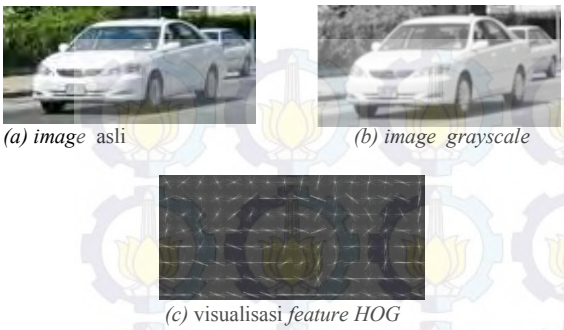
Sistem dibagi menjadi empat bagian yakni ekstrasi *feature*, visualisasi *feature*, *learning* SVM dan deteksi mobil. Secara keseluruhan, desain sistem ditampilkan pada gambar 5.



Gambar 5. Desain sistem

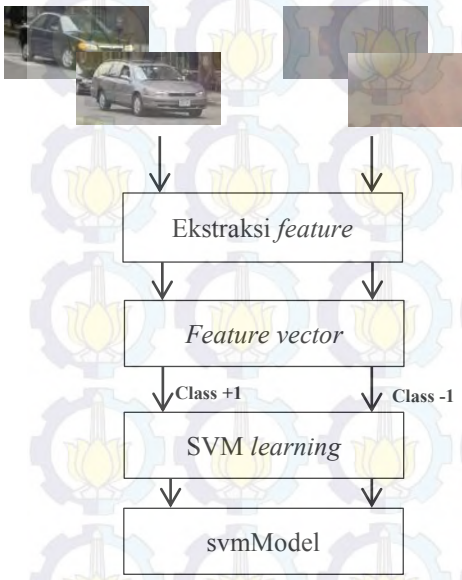
Pada gambar 5, Input *image* yang pertama merupakan *image* yang diambil dari *dataset* yang sudah disiapkan (isi *dataset* akan dijelaskan pada bagian selanjutnya) dengan ukuran 128x64. Tahap kedua adalah ekstrasi *feature*, dalam proses ekstrasi *feature*, input *image* dirubah kedalam *image grayscale* untuk memudahkan penghitungan *gradient image* . Penghitungan gradien *image* dimaksudkan untuk mendapatkan karakteristik mobil. gradient sebuah *image* didapatkan dengan mencari garis tepi dimana antara daerah satu dengan daerah sebelahnya memiliki perbedaan nilai yang tinggi. Untuk mencarinya digunakan *sobel filter* yang mengektimasi gradient arah sumbu x dan arah sumbu y. setelah didapatkan gradien *image* , langkah selanjutnya adalah membagi *image* kedalam daerah-daerah kecil yang disebut *cell* dengan ukuran 4x4 piksel. Histogram dihitung pada Masing-masing *cell* dengan jumlah *bin* sebanyak 9 dalam sudut 180 derajat. Nilai histogram masing-masing *cell* ini didistribusikan kedalam orientasi *cell* yang membentuk garis vektor dengan jumlah garis sama dengan jumlah bin yang dibentuk, yaitu 9 garis dengan sudut 20 derajat antara garis satu dengan garis yang selanjutnya sehingga total garis yang dibentuk dalam sudut 360 derajat sebanyak 18 garis. Untuk memaksimalkan *feature* HOG maka dibentuk sebuah block yang merupakan gabungan dari beberapa *cell* dengan ukuran 2x2 *cell* atau 8x8 piksel. teknik *overlapping block* dilakukan dengan membuat *overlap block* yang bersebelahan dan menghitung kembali bagian *block* yang *overlap*.

Bagian sistem selanjutnya adalah visualisasi *feature* HOG. Visualisasi ini berupa garis vektor yang secara keseluruhan menunjukkan karakteristik dari obyek, yaitu mobil. hasil visualisasi ditunjukkan pada gambar 6.



Gambar 6. Hasil Visualisasi *feature* HOG

Feature HOG juga dipakai sebagai input dari *learning* SVM. *Feature* HOG dirubah kedalam *feature vector* dengan ukuran 4608x1. Ukuran *feature vector* dihasilkan dari perkalian dari ukuran block (2x2 *cell*) , jumlah bin 9, dan banyaknya block yang terbentuk dari *image* dengan ukuran 128x64. *Feature vector* inilah yang dipakai sebagai input untuk proses *learning* SVM. Dalam proses learning dilakukan 2 tahap, yaitu *training* data dan tes model. Pada *training* data, *feature* HOG dari *image* positif diberi label +1 yang menandakan bahwa *image* yang dilatih merupakan *feature image* positif dan *feature image* negatif diberi label -1 yang menandakan *image* yang dilatih merupakan *feature image* negatif. *Training* SVM ini menghasilkan sebuah model yang akan dites dengan menggunakan *classify* SVM. Gambar 7 menunjukkan proses *learning* SVM



Gambar 7. Learning SVM

Model hasil *learning* SVM ini digunakan sebagai acuan dalam mendeteksi mobil. proses deteksi dilakukan dengan melakukan *filter* dari pojok kiri atas sampai pojok kanan bawah. Apabila dalam proses *filter* tersebut terdapat bagian *image* yang sesuai dengan model, maka obyek dianggap sebagai *image* positif yang berarti merupakan obyek mobil dengan memberi kotak pada obyek tersebut.

D. *Data set*

Data set merupakan komponen yang dibutuhkan untuk data *learning* dengan menggunakan SVM. Data *learning* yang dibutuhkan berupa *training* data positif yaitu data yang berisi obyek mobil, *training* data negatif yang berisi obyek bukan mobil (*background*), data tes positif, data tes negatif, serta *image* yang akan dideteksi. *Image* yang dipakai semuanya berukuran 128x64, kecuali *image* yang akan dideteksi, memiliki ukuran 680x480, lebih besar dari *image training* dan *image* tes. Data set diambil dari Chenghak dataset yang didalamnya berisi :

a) 196 *testing image*
b) 196 *annotations files* untuk *testing image*
c) 2462 *training car samples*
d) 1028 *testing car samples*

Namun dataset diatas tidak disertakan *image* negatif untuk *training* dan *testing*. Untuk *image* negatif didapatkan dengan *crop image* acak dengan ukuran 128x64. Banyaknya *image* negatif yang digunakan adalah sebagai berikut.

a) 4627 *training image* negatif
b) 1978 *testing image* negatif

Secara keseluruhan komposisi *image data set* ditunjukkan dapa tabel 1.

Table 1. data learning

Image Positif		*Image* Negatif	
Training image	2462	*Training image*	4627
Test image	1028	*Test image*	1978
Image global = 196			

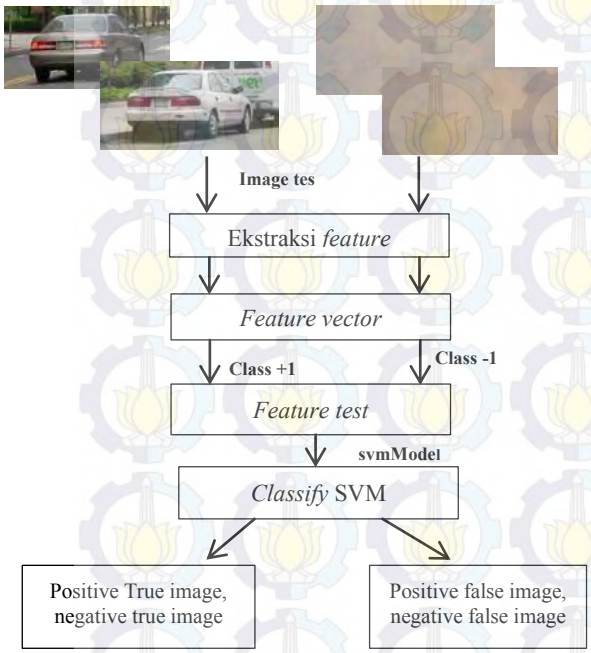
E. *Testing model*

Testing *image* merupakan pengujian model hasil learning untuk diketahui tingkat akurasi model dalam mengenali *image* tes. *Image* tes memiliki besar piksel yang sama dengan data *training*, namun memiliki variasi *image* yang berbeda. Dari gambar 8, sebelum melakukan *testing* model terlebih dilakukan ekstraksi feature HOG dari *image* yang dites. Selanjutnya feature tersebut dirubah kedalam bentuk *feature* vector yang ukurannya sama dengan *feature vector* data *training* yaitu 4608x1. *Feature vector* dari *image* positif digabung dengan *feature vector image* negatif yang menghasilkan sebuah *feature* tes. *Feature vector* dari *image* tes ini diklasifikasikan menggunakan *classify* SVM dengan model *training* sebagai acuannya. Dari tes ini diketahui banyaknya *image* yang berhasil dideteksi sesuai dengan *class* yang ditentukan sebelumnya (*image* positif dideteksi sebagai *image* positif, *image* negatif dideteksi sebagai *image* negatif) dan juga diketahui banyaknya *image* yang tidak berhasil dideteksi sesuai dengan *class* yang ditentukan.

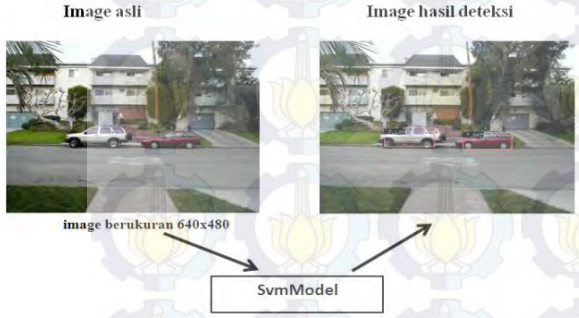
F. Deteksi Mobil

Pada gambar 9, Proses deteksi dilakukan dengan melakukan *filtering* pada *image* yang dideteksi dengan model (svmModel) sebagai *filter*, yaitu dengan cara menggerakkan model dari pojok kiri atas dari *image* yang dideteksi sampai pojok kanan bawah. Apabila dalam proses *fiter* terdapat *feature* HOG yang bobotnya (*weight*) diatas nilai *threshold* yang ditentukan maka *feature* tersebut dideteksi sebagai *image* positif dengan memberi kotak pada bagian tersebut. Namun

apabila bobot *feature* yang dideteksi dibawah nilai *threshold* yang ditentukan maka *feature* tersebut danggap sebagai *image* negatif.



Gambar 8. Testing *image*



Gambar 9. Proses deteksi mobil

III. HASIL DAN DISKUSI

A. Pengujian dengan jumlah data *training* yang berbeda-beda

Pada pengujian ini, jumlah *training* data yang digunakan dibuat berbeda-beda. Hasil *training* disimpan dalam bentuk model kemudian dites dengan menggunakan SVM classify.

Pengujian dilakukan dengan komposisi data *training* seperti tabel 2 dengan jumlah tes *image* positif sebanyak 1028 tes *image* positif dan 1978 tes *image* negatif. Pengujian dilakukan sebanyak 11 kali dengan data *training* awal sebanyak 200 *image* positif dan 400 *image* negatif.

Untuk pengujian pertama, menghasilkan data sebagai berikut.

Pengujian ke-1
Data *image* *training* :
a) *Image* positif 200 *image*
b) *Image* negatif 400 *image*
Hasil Tes :
a) Tes *image* positif : 965 correct, 63 incorrect, 1028 total
b) Tes *Image* negatif : 1970 correct, 8 incorrect, 1978 total
c) Tes *Image* positif dan negatif : 2965correct, 41 incorrect,3006 total

Pengujian ke-2
Data *image* *training* :
a) *Image* positif 400 *image*
b) *Image* negatif 800 *image*
Hasil Tes :
a) Tes *image* positif : 981 correct, 47 incorrect, 1028 total
b) Tes *Image* negatif : 1970 correct, 8 incorrect, 1978 total
c) Tes *Image* positif dan negatif : 2955 correct, 51 incorrect,3006 total

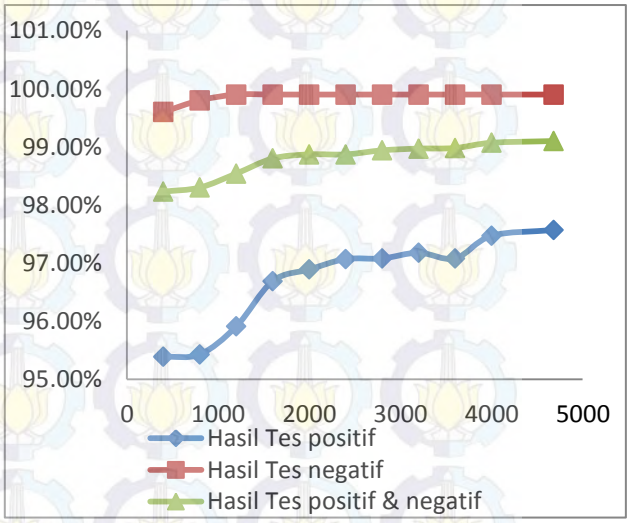
Pengujian dilanjutkan sampai 11 kali pengujian dengan menambah 200 *image* untuk *image* positif dan menambah 400 *image* untuk *image* negatif dari data pengujian sebelumnya.

Dari hasil pengujian tersebut didapatkan tabel 2 seperti dibawah ini.

Tabel 2. Hasil pengujian *training* data

Training positif	*Training* negatif	Hasil Tes positif	Hasil Tes negatif	Hasil Tes positif & negatif
200	400	95.39%	99.60%	98.23%
400	800	95.43%	99.80%	98.30%
600	1200	95.91%	99.90%	98.54%
800	1600	96.69%	99.90%	98.80%
1000	2000	96.89%	99.90%	98.87%
1200	2400	97.47%	99.90%	98.87%
1400	2800	97.08%	99.90%	98.94%
1600	3200	97.18%	99.90%	98.97%
1800	3600	97.08%	99.90%	98.98%
2000	4000	97.47%	99.90%	99.07%
2462	4680	97.57%	99.90%	99.10%

Dari tabel diatas didapatkan grafik hubungan antara jumlah data *training* dengan hasil tes.



Gambar 10. Grafik hubungan antara jumlah data *training* dengan hasil tes.

Dari grafik pada gambar 10, hasil tes positif menunjukkan peningkatan apabila jumlah data *training* ditambah. Prosentase

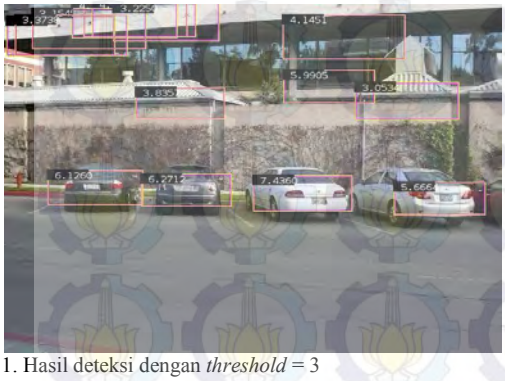
terakhir menunjukkan bahwa tingkat keberhasilan tes dengan *classify* SVM sebesar 99.10% dengan data *training* positif sebanyak 2462 *image* dan data *training* negative sebanyak 4680 *image* . Namun pada saat tes ke-2, dengan jumlah *training* data 400 *image* positif dan 800 *image* negative mengalami penurunan, dikarenakan model yang dihasilkan dari *training* ke-2 mampu mendeteksi *image* dengan komposisi 981 correct, 47 incorrect, 1028 total.

Sedangkan hasil tes negatif menunjukkan tingkat keberhasilan sebesar 99,80% dari tes *image* sebanyak 1978 *image* . Prosentase mengalami peningkatan mulai saat tes ke-1sampai tes ke-3 dan selanjutnya prosentasenya tetap sampai pada tes ke-11.

Untuk tes *image* positif dan negatif mengalami peningkatan mulai saat tes ke-3 sampai tes ke-11. Namun hasil tes menurun saat tes ke-2 akan tetapi hasilnya meningkat kembali sampai tes ke-11. Secara *linear*, hasil tes menunjukkan peningkatan dengan ditambahnya jumlah *training* data.

B. Pengujian dengan *threshold* yang berbeda-beda

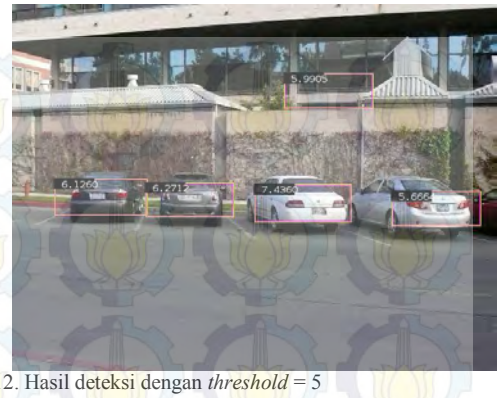
Pengujian yang kedua adalah pengujian deteksi mobil dengan mengubah nilai *threshold* . pengujian awal dilakukan dengan nilai *threshold* = 3. Hasil deteksi ditunjukkan pada gambar 11.



Gambar 11. Hasil deteksi dengan *threshold* = 3

Obyek yang mempunyai nilai *weight* dibawah 3 dideteksi sebagai *image* negatif sedangkan *image* dengan *weight* diatas 3 dideteksi sebagai *image* positif. Dari gambar 11 Banyak obyek bukan mobil yang dideteksi sebagai mobil. hal ini dikarenakan nilai *threshold* yang terlalu kecil. Hasilnya akan berbeda apabila nilai *threshold* pada deteksi *image* sebesar 5 (gambar 12). Empat obyek terdeteksi sebagai *image* positif, namun ada satu obyek yang seharusnya *image* negatif terdeteksi sebagai *image* positif.

Hasil pengujian dengan 196 *image* menunjukkan nilai *threshold* maksimal antara *image* satu dengan *image* lain tidak sama. Dengan memaksimalkan nilai *threshold* pada *image* yang dideteksi, tingkat keberhasilan sistem dalam mendeteksi mobil sebesar 76,17% dengan jumlah 403 *image* mobil dan yang terdeteksi sebanyak 307 *image* .



Gambar 12. Hasil deteksi dengan *threshold* = 5

IV. KESIMPULAN

Dari hasil perancangan dan pengujian seluruh sistem dalam dapat ditarik beberapa kesimpulan. Hasil pengujian tes model menunjukkan tingkat keberhasilan mencapai 99,10% dalam mengenali *image* tes. Pada pengujian image tes positif, saat data training ditambah, tingkat keberhasilannya meningkat dari 95.39% menjadi 97,57 % . untuk pengujian image tes negatif, hasilnya meningkat dari 99,60% menjadi 99,90% pada pengujian ketiga dan konstan untuk pengujian berikutnya. sedangkan untuk pengujian image gabungan (tes image positif dan negatif) tingkat keberhasilannya naik 98.23% menjadi 99,10% .

UCAPAN TERIMA KASIH

Penulis C.P mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu dalam menyelesaikan penelitian ini, diataranya keluarga dan orang-orang terdekat yang selalu memberikan doa dan dukungannya serta dosen pembimbing dan pengajar Bidang Studi Sistem Komputer Jurusan Elektro IT atas saran dan bimbingannya

DAFTAR PUSTAKA

[1] Navneet Dalal and Bill Triggs, *Histograms of Oriented Gradients for Human Detection,* INRIA Rhˆ one-Alps, 655 avenue de l'Europe, Montbonnot 38334, France, 2005.

[2] Cheng-Hao Kuo and Ramakant Nevatia, *Robust Multi-View Car Detection using Unsupervised Sub-Categorization,* University of Southern California, Institute for Robotics and Intelligent Systems Los Angeles, CA 90089, USA, 2009.

[3] B. Boser, I. Guyon, and V. Vapnik, *An training algorithm for optimal margin classifiers,* Fifth Annual Workshop on Computational Learning Theory, 1992.

[4] Steve R. Gunn, *Support Vector Machines for Classification and Regression*, , Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science, Univerity of Southampton, 1998.

[5] Susanti Wijaya Kusuma "penerapan teknik SVM (Support Vector Machine) untuk Pendeteksian pengaruh kestabilan enzim termutasi", Universitas Indonesia, Depok, 2009.