

Face Detection System Menggunakan Metode Histogram of Oriented Gradients (HOG) dan Support Vector Machine (SVM)

Face Dtection System using Histogram of Oriented Gradients (HOG) Method amd Support Vector Machine(SVM)

Miftahul Bagus Pranoto¹, Kurniawan Nur Ramadhani,S.T.,M.T.², Anditya Arifianto,S.T.,M.T.³

Prodi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom
mbpranoto@gmail.com, andiess2006@gmail.com, anditya.arifianto@gmail.com

Abstrak

Pengembangan *face detection* dalam ilmu teknologi semakin dibutuhkan sekarang ini. Selain untuk *security system* yang membuat proses autentifikasi dan keamanan sistem semakin baik, *face detection* juga diperlukan untuk kebutuhan entertainment dan social media yang menambah nilai jual, dimana umumnya menerapkan teknologi *augmented reality*.

Pada penelitian ini akan dikembangkan metode yang digunakan untuk mendeteksi wajah menggunakan *Histogram of Oriented Gradients (HOG)* sebagai ekstraksi cirinya dan *Support Vector Machine (SVM)* sebagai klasifikatornya. Dataset yang digunakan sebanyak 644 citra positif dan 2.572 citra negatif untuk proses training, 110 citra positif untuk proses *testing*, dan 10 citra untuk proses deteksi. Sistem dibangun menggunakan HOG dengan *block* yang terdiri dari 2x2 *cell* dimana satu *cell* terdiri dari 8x8 *pixel* dan menggunakan *kernel linear* dalam *Support Vector Machine*. Sistem ini menghasilkan nilai *f-1 score* sebesar 71,42%.

Kata Kunci : *face detection, Histogram of Oriented Gradients, Support Vector Machine.*

Abstract

The development of face detection in the science of technology is increasingly needed nowadays. In addition to the security system that makes the process of authentication and system security getting better, face detection is also needed for the needs of entertainment and social media that add value selling, which generally apply augmented reality technology.

In this research, we will develop the method used to detect faces using Histogram of Oriented Gradients (HOG) as its extraction and Support Vector Machine (SVM) as its classifier. The dataset used was 644 positive images and 2,572 negative images for the training process, 110 positive images for the testing process, and 10 images for the detection process. The system is built using HOG with block consisting of 2x2 cell where one cell consists of 8x8 pixel and using linear kernel in Support Vector Machine. This system produces a f-1 score of 71.42%.

Keywords : *face detection, Histogram of Oriented Gradients, Support Vector Machine*

1. Pendahuluan

Ilmu pengetahuan dan teknologi begitu dinamis, berubah dan berkembang sangat pesat. Salah satu bagian dari teknologi yang juga tidak pernah berhenti berkembang adalah *computer vision* dan *image processing*. *Computer vision* adalah ilmu yang mempelajari bagaimana sebuah mesin mampu melihat dan mengenali sebuah objek yang diamati[9]. Sedangkan *image processing* merupakan jenis teknologi untuk melakukan pengolahan dan pemrosesan sebuah citra. Kedua cabang ilmu tersebut telah banyak diimplementasikan dalam berbagai aplikasi masa kini, misalnya untuk edukasi, *security* dan *entertainment*[10].

Di antara aplikasi-aplikasi yang terus dikembangkan dengan teknologi *computer vision* dan *image processing* adalah *face detection* (deteksi wajah). Implementasi nyata dari *face detection* ini adalah aplikasi-aplikasi hiburan berbasis *augmented reality*[11]. Belakangan, aplikasi berbasis *face*

processing, yang di antaranya adalah *face detection*, semakin berkembang seiring berkembangnya *social media*. Setelah Snapchat mempelopori aplikasi *face masking* pada salah satu fitur *social media*-nya, kini facebook juga telah mengakuisisi sebuah *startup* yang fokus pada *face masking* dan *face swapping*, MSQRD. Ada banyak metode yang digunakan para *developers* untuk membuat aplikasi *face detection*, di antaranya adalah Viola-Jones[12]. Viola-Jones sendiri sering dipakai karena impelementasinya yang cukup mudah[12]. Namun, Viola-Jones memiliki kelemahan, antara lain kemampuan diskriminatif tidak terlalu kuat untuk fitur tertentu, serta ambang batas yang digunakan untuk membedakan objek wajah dengan bukan wajah yang sulit beradaptasi dengan mode-distribusi yang kompleks[12].

Dari masalah tersebut, dibutuhkan sebuah metode yang lebih akurat dalam mendeteksi wajah. Maka dari itu, penulis akan menggunakan metode *Histogram of Oriented Gradients* (HOG) ini yang akan dikombinasikan dengan klasifikator *Support Vector Machine* (SVM) sebagai metode dalam melakukan *face detection*. HOG dikembangkan oleh Navneet Dalal dan Bill Trigs pada tahun 2005. Pada awalnya, pengujian yang dilakukan Navneet Dalal dan Bill Trigs adalah untuk mendeteksi penjalan kaki. Pada perkembangannya, metode ini dapat digunakan untuk mendeteksi objek lain. Sedangkan pada SVM, terdapat dua tahapan yang dipakai, yaitu *training* dan *classify*. *Training* pada SVM digunakan untuk menghasilkan model, sedangkan *classify* digunakan untuk melakukan pengujian[2].

2. Kajian Teori

2.1. Computer Vision

Computer vision merupakan salah satu bidang ilmu yang mempelajari bagaimana komputer dapat merekonstruksi, menginterpretasikan, memahami dan mengamati suatu objek. *Computer vision* juga mempelajari metode-metode dalam pemrosesan, analisis, pemahaman citra dan data-data lain dari dunia nyata secara umum untuk menghasilkan informasi numerik atau simbolik. *Computer vision* berkaitan dengan meniru penglihatan manusia menggunakan pendekatan *software* dan *hardware*[5].

Computer vision telah berkembang dalam berbagai bidang dengan tujuan yang bervariasi. Beberapa contoh dari implementasi *computer vision* adalah pemodelan medis, perangkat dalam interaksi manusia-komputer, peralatan navigasi pada sistem transportasi, pengontrolan di bidang industri, dan lain-lain[5].

Informasi simbolik dan numerik dari *computer vision* disebut visualisasi data. Bentuk visualisasi datanya adalah *image processing* (pengolahan citra) dan *pattern recognition* (pengenalan pola)[5].

2.2. Pengolahan Citra (*Image Processing*)

Image processing atau pengolahan citra adalah bidang yang berkenaan dengan proses transformasi gambar atau citra. Proses ini berfungsi untuk mendapatkan kualitas citra yang lebih baik[5]. Untuk menangkap dan mengolah sebuah informasi agar dapat mendekati kemampuan manusia, *computer vision* harus terdiri dari banyak fungsi pendukung, antara lain, proses penangkapan citra (*image acquisition*), pengolahan citra (*image processing*), analisa data citra (*image analysis*), serta proses pemahaman data citra (*image understanding*)[5].

2.3. Citra Digital

2.3.1. Definisi Citra Digital

Citra digital merupakan gambar yang berupa matriks fungsi 2 dimensi $f(x,y)$, dimana x dan y merupakan koordinat bidang dan f adalah definisi fungsi koordinat dari (x,y) [5]. Terdapat dua jenis citra yaitu citra diam atau citra statis (*still image*) dan citra bergerak (*moving image*). Citra diam merupakan citra tunggal yang tidak bergerak. Sedangkan citra bergerak adalah citra diam yang disusun secara berurutan sehingga terlihat seperti bergerak[5].

2.3.2. Citra Warna (*True Color Image*)

Citra berwarna atau dinamakan *true color image* atau juga biasa disebut citra RGB, merupakan jenis citra yang berbasis warna primer dalam bentuk komponen *red* (merah), *green* (hijau) dan *blue* (biru). Setiap komponen warna menggunakan 8bit, dan nilainya berkisar 0 sampai 255, sehingga kemungkinan warna yang dapat disajikan adalah $255^3 = 16.581.375$ warna. Dalam tugas akhir ini, penulis menggunakan citra jenis ini untuk digunakan dalam di dalam sistem sebagai bahan latihan dan bahan uji[9].

2.3.3. Citra Keabuan (*Grayscale Image*)

Citra keabuan atau *grayscale image* merupakan citra digital yang hanya memiliki satu nilai

kanal pada setiap pixel-nya, dengan kata lain nilai bagian $red = green = blue$. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari keabuan[9].

2.3.4. Citra Biner

Citra biner adalah citra digital yang hanya memiliki 2 kemungkinan nilai *pixel* yaitu 0 dan 1. Angka 0 merepresentasikan warna hitam dan angka 1 merepresentasikan warna putih sehingga citra ini pun sering disebut dengan citra B&W (*black & white*). Hanya dibutuhkan 1 *bit* untuk mewakili citra ini[9].

2.4. Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients (HOG) adalah salah satu metode ekstraksi ciri yang digunakan dalam *image processing* untuk mendeteksi suatu objek. HOG berasal dari sebuah asumsi yang menyatakan bahwa suatu objek dapat direpresentasikan dengan baik berdasarkan bentuk. Untuk memperoleh informasi pembeda maka gambar akan dibagi menjadi *cell* dan setiap *cell* akan dihitung sebagai *histogram of oriented gradients*. Setiap piksel dalam *cell* berkontribusi pada saat dilakukan voting bobot untuk membangun sebuah histogram yang berorientasi pada nilai-nilai gradien yang dihitung [8].

Proses awal untuk membangun HOG adalah dengan menghitung nilai gradien horizontal dan vertikal menggunakan deteksi tepi Sobel 1-D dengan rumus sebagai berikut :

$$dx = I(x + 1, y) - I(x - 1, y) \quad (2.1)$$

$$dy = I(x, y + 1) - I(x, y - 1) \quad (2.2)$$

Keterangan:

dx = nilai gradien citra secara vertical

dy = nilai gradien citra secara horizontal

$I(x,y)$ = nilai piksel pada baris x dan kolom y

Dari nilai gradien tersebut, akan dikalkulasikan *magnitude* dan orientasi-nya, dengan menggunakan rumus :

$$m(x, y) = \sqrt{dx^2 + dy^2} \quad (2.3)$$

$$\theta(x, y) = \arctan\left(\frac{dy}{dx}\right) \quad (2.4)$$

Setelah didapatkan nilai *magnitude* dan orientasinya, maka dilakukan *voting* histogram sesuai dengan pembagian nilai *bin*-nya dalam rentang 0-180 derajat. Jika nilai *bin* = 9 maka, terdapat 9 daerah pembagian pada histogram (0-10 derajat, 10-30 derajat, 30-50 derajat, 50-70 derajat, 70-90 derajat, 90-110 derajat, 110-130 derajat, 130-150 derajat, 150-170 derajat, dan 170-180 derajat).

Dari hasil perhitungan *magnitude* dan orientasi, maka dihitung nilai *bin*-nya dengan menggunakan rumus sebagai berikut :

$$hist(xA) = hist(xA) + mag(x, y) \times (B - \theta(x, y)) \quad (2.5)$$

$$hist(xB) = hist(xB) + mag(x, y) \times (\theta(x, y) - A) \quad (2.6)$$

Keterangan :

xA = nilai *bin* ke-A

xB = nilai *bin* ke-B

A = nilai sudut *bin* ke-A

B = nilai sudut *bin* ke-B

Setelah dilakukan penghitungan histogram untuk setiap *cell* selanjutnya dilakukan penggabungan histogram untuk seluruh *cell* dalam satu *block*. Hasil penggabungan tersebut akan dinormalisasi dengan menggunakan rumus sebagai berikut :

$$L_1 - norm : v' = \frac{v}{\sqrt{\|v\|^2 + \varepsilon}}$$

$$L_2 - norm : v'' = \frac{v'}{\sqrt{\|v'\|_2^2 + \varepsilon^2}}$$

Dengan :

v = vektor blok yang mengandung histogram

$\varepsilon = 0,1$ (*konstanta*)

$L - norm$ = normalisasi

(2.7)

(2.8)

Setiap *block* yang saling bertetangga dapat dijumlahkan dalam berbagai cara (*overlapping*) sehingga tidak menutup kemungkinan bahwa akan ada *cell* yang muncul pada *block* yang berbeda.

2.5. Support Vector Machine (SVM)

Support Vector Machine, secara sederhana dapat dijelaskan sebagai usaha mencari *hyperlane* atau garis pembatas yang paling optimal (terbaik) yang berfungsi sebagai pembeda dua buah *class* pada *input space*. SVM dikembangkan oleh Boser, Guyon dan Vapnik, serta pertama kali dipresentasikan pada tahun 1992 di *Annual Workshop on Computational Learning Theory*[3].

Pada *Support Vector Machine* terdapat beberapa *kernel*. Adapun dalam penelitian kali ini, penulis akan menggunakan tiga *kernel* sebagai bahan penelitian, yaitu *linear*, *polynomial* dan *gaussian*, yang masing-masing dapat didefinisikan dengan fungsi :

Tabel 2-1 Definisi Kernel SVM

No	Kernel	Definisi Fungsi
1	Linear	$K(x,y) = x y$ (2.9)
2	Polinomial	$K(x,y) = (x \cdot y + c)^d$ (2.10)
3	Radial (Gaussian)	$K(x,y) = \exp\left(\frac{- x - y ^2}{2\sigma^2}\right)$ (2.11)

2.6. Deteksi Wajah (Face Detection)

Deteksi wajah atau *face detection* merupakan teknologi komputer untuk mendeteksi wajah manusia, dengan cara menentukan posisi dan ukuran wajah manusia dalam suatu citra digital. Teknologi ini mampu mendeteksi wajah melalui sifat atau karakteristik wajah dengan mengabaikan objek lainnya, seperti badan manusia itu sendiri[7]. Bidang-bidang penelitian lain yang juga berkaitan dengan *face processing* (pemrosesan wajah) antara lain *face localization* (lokalisasi wajah), *face tracking* (penjejukan wajah), *facial expression recognition* (pengenalan ekspresi wajah), serta *face authentication* (autentifikasi wajah)[7].

2.7. Confusion Matrix

Confusion Matrix adalah tabel yang digunakan untuk mendeskripsikan performansi dari model klasifikasian pada sekumpulan data uji yang diketahui jumlah data yang benar. Terdapat empat ketentuan yang terdapat pada *confusion matrix* yaitu *True Positive*, *True Negative*, *False Positive*, dan *False Negative*[16].

Tabel 2-2. Confusion Matriks

	Predicted = Yes	Predicted = No
Actual = Yes	True Positive	False Negative
Actual = No	False Positive	True Negative

Perhitungan performansi dilakukan setelah proses klasifikasi selesai diproses. Performansi dari sistem menggunakan *f1-score*. Tabel di atas merupakan representasi *confusion matrix* setiap kelas, yang nantinya masing-masing komponen pada matriks yang bersesuaian di jumlah, dan masing-masing akan mendapatkan jumlah dari semua *True Positive*, *False Positive*, *True Negative*, dan *False Negative*. Setelah mendapatkan nilai-nilai pada *confusion matrix*, selanjutnya melakukan penghitungan *precision*, *recall*, dan *f1-score*.

$$Precision = \frac{\sum True Positive}{\sum True Positive + \sum False Positive} \quad (2.9)$$

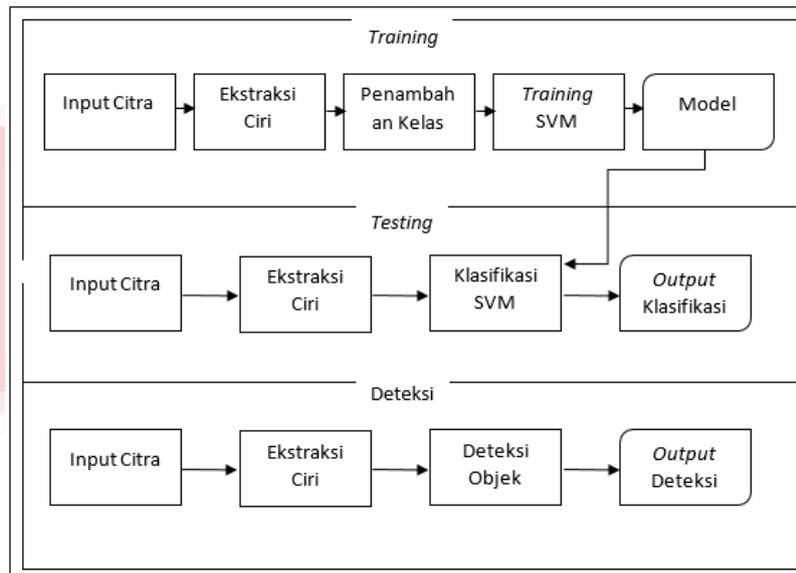
$$Recall = \frac{\sum True Positive}{\sum True Positive + \sum False Negative} \quad (2.10)$$

$$F1-Score = \frac{(2 \times Precision \times Recall)}{(Precision + Recall)} \quad (2.11)$$

Nilai *range* yang digunakan pada *precision*, *recall* dan *f1-score* ini adalah dari 0-1. Dalam kasus yang umum, biasanya *range* 0-1 dikalikan dengan angka 100% untuk melihat presentasi angka dalam suatu performansi. Nilai yang terbaik jika perhitungan *f1-score* mencapai nilai 1 atau dalam persen mencapai angka 100%. Dengan kata lain, nilai performansi terbaik, akan diperoleh jika nilai dari *True Positive* adalah 1 dan *False Positive* serta *False Negative* adalah 0.

3. Perancangan Sistem

Dalam penelitian tugas akhir ini, pemrosesan *face detection* akan melalui beberapa tahap yang dilakukan. Penulis membagi tahapan-tahapan tersebut menjadi 3 bagian, yaitu proses *training*, proses *testing*, dan proses deteksi. Berikut adalah gambaran umum dari sistem yang penulis kembangkan :



Gambar 3-1 Gambaran Umum Sistem

3.1. Pengumpulan Data

Dalam tugas akhir ini, penulis melakukan pengumpulan data dari berbagai sumber, agar representatif dan variatif. Adapun beberapa sumber yang penulis gunakan antara lain Yale Database[14], BioID[15], Google Image, dan lain-lain.

Sebagian data citra yang telah dikumpulkan tersebut, 110 citra di antaranya melalui proses *editing*, seperti *scaling*, rotasi dan pemotongan.

3.2. Ekstraksi Ciri

Ekstraksi ciri pada data citra yang telah dikumpulkan menggunakan metode *Histogram of Oriented Gradients (HOG)*. Ekstraksi ciri dilakukan terhadap citra berukuran 24×32 pixel. Potongan citra akan dibagi menjadi beberapa *block* yang saling *overlapping*, yang di dalamnya terdapat beberapa *cell*, dimana setiap *cell* disusun oleh beberapa *pixel*. Kemudian, pada setiap *cell* dihitung nilai gradien citra secara horizontal dan vertikal dengan menggunakan rumus pada persamaan (2.1) dan (2.2). Setelah itu, dihitung nilai *magnitude* dan orientasinya menggunakan rumus pada persamaan (2.3) dan (2.4). Selanjutnya, dilakukan normalisasi dengan menggunakan rumus pada persamaan (2.7) dan (2.8), untuk menghindari perbedaan nilai akibat perbedaan pencahayaan. Dari hasil perhitungan tersebut akan dibangun histogram dengan cara melakukan *voting* nilai θ sesuai dengan nilai *bin* yang telah ditentukan (*binning process*). Kemudian, akan dilakukan penggabungan histogram dari seluruh *cell* yang ada pada satu *block*.

3.3. Training

Data yang telah di ekstraksi ciri, kemudian di *training* menggunakan *Support Vector Machine (SVM)*, setelah sebelumnya ditambahkan kelas data, 1 untuk citra positif/wajah, dan -1 untuk citra negatif/bukan wajah. Dalam proses *training* kali ini, penulis melakukan observasi terhadap 3 *kernel SVM*, yaitu *Linear*, *Polynomial/Quadratic*, dan *Gaussian*. Keluaran dari proses *training* ini adalah model yang akan digunakan dalam memprediksi objek wajah.

3.4. Validasi

Proses validasi dilakukan setelah proses *training*. Validasi ini dilakukan terhadap parameter-parameter yang menentukan dalam proses deteksi, sehingga dalam proses deteksi nanti, sistem menggunakan parameter-parameter yang telah teruji memiliki akurasi yang tinggi. Dalam proses validasi ini, dilakukan pengujian terhadap 110 citra uji yang telah disiapkan, dan 644 citra positif yang digunakan sebagai data latih. Baik citra uji dan citra latih tersebut berukuran sama, yaitu 24×32 piksel, Proses validasi diawali dengan menginputkan citra, lalu mengekstraksinya dengan HOG, dan selanjut melakukan pengenalan/klasifikasi kelas data berdasarkan model yang telah dibangun dari proses *training*. Output dari proses validasi ini adalah 1 untuk citra yang dikenali sebagai wajah, dan -1 untuk citra yang tidak dikenali sebagai wajah.

3.5. Deteksi Objek

Proses deteksi wajah menggunakan parameter-parameter yang terbaik setelah melalui proses pengujian, yaitu ukuran *cell HOG*, jumlah dataset, dan *kernel SVM* yang digunakan. Dalam proses deteksi wajah ini sendiri

menggunakan *sliding window* dengan masing-masing window mengekstraksi ciri berukuran 24×32 *pixel*. Window bergerak dari sudut atas kiri ke kanan, dan selanjutnya ke bawah. Dengan proses deteksi seperti ini, wajah yang bisa dideteksi hanya berukuran 24×32 *pixel*.

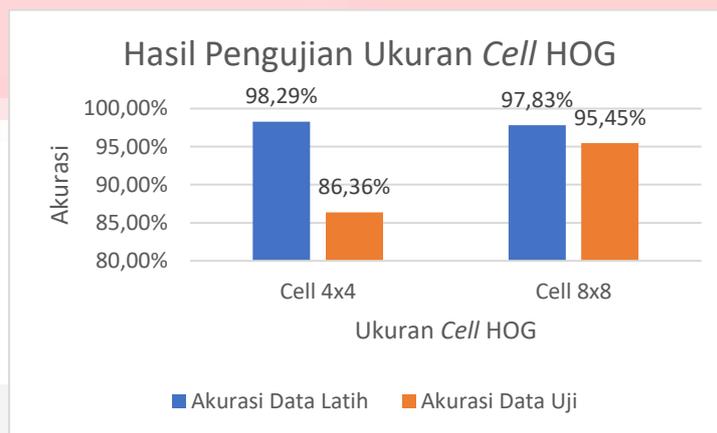
4. Pembahasan

4.1. Implementasi

Sesuai dengan rancangan sistem yang telah dijelaskan dalam bab tiga, sistem dibangun dengan bahasa *tool* Matlab. Dataset yang digunakan adalah 644 citra positif dan 2.572 citra negatif untuk proses *training*, 11 citra positif untuk proses validasi, 10 citra untuk proses deteksi. Sistem deteksi dibangun dengan menerapkan parameter-parameter terbaik dari proses validasi yang telah dilakukan sebelumnya.

4.2. Analisis Pengaruh Ukuran Cell HOG

Dalam skenario validasi pertama, penulis membandingkan parameter ukuran *cell* HOG yang akan digunakan. Adapun yang penulis bandingkan adalah ukuran *cell* 4×4 *pixel* dengan 8×8 *pixel* yang didapatkan hasil sebagai berikut :

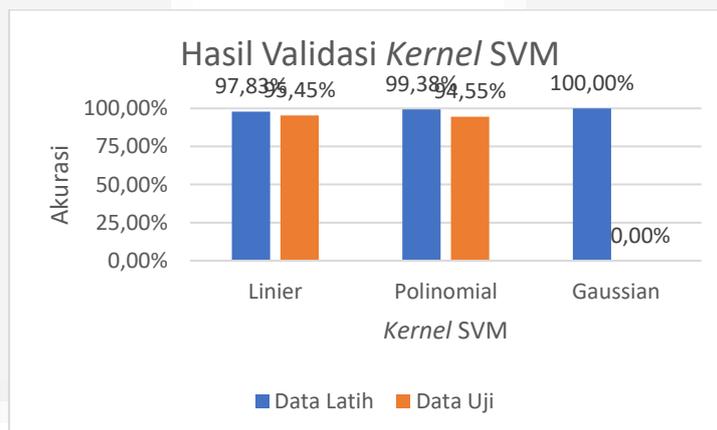


Gambar 2 : Hasil validasi ukuran *cell* HOG

Berdasarkan Gambar 9, dapat diketahui bahwa tingkat akurasi ukuran *cell* HOG 8×8 *pixel* lebih tinggi daripada 4×4 *pixel*, yaitu sebesar 95,45% dibandingkan dengan 86,36%. Dari hasil validasi pertama ini, diputuskan proses deteksi akan menggunakan ukuran *cell* HOG sebesar 8×8 *pixel*.

4.3. Analisis Pengaruh Kernel SVM

Pada scenario ini, penulis melakukan validasi terhadap tiga kernel SVM, yaitu *linier*, *polynomial*, dan *gaussian*. Hasilnya adalah sebagai berikut :



Gambar 3 : Hasil validasi *kernel* SVM

Dari Gambar 9 diketahui bahwa tingkat akurasi yang paling tinggi adalah menggunakan kernel linier (95,45%), disbanding dengan polynomial yang 94,55% dan Gaussian 0%. Maka, dari proses validasi yang telah dilakukan terhadap kernel SVM yang akan digunakan, dalam proses deteksi penulis akan menggunakan kernel linier.

4.3.1. Deteksi Wajah

Pada proses deteksi wajah ini, dilakukan pendeteksian terhadap 10 citra yang telah disiapkan dalam ukuran bervariasi. Masing-masing citra dideteksi menggunakan *sliding window* berukuran 24×32 *pixel* dan bergerak setiap

4 pixel baik secara vertikal maupun horizontal. Dari proses deteksi yang dilakukan, diukur nilai *precision*, *recall*, dan *f1-score* menggunakan *confusion matrix*.

Berikut hasil proses deteksi dari masing-masing citra :

Tabel 4.1 Hasil Deteksi Masing-Masing Citra

Nomor Citra	Jumlah Deteksi	True Positive	False Positive	False Negative
1	8	6	2	0
2	14	5	9	0
3	31	11	20	0
4	3	2	1	0
5	8	3	5	0
6	13	11	2	0
7	13	5	8	0
8	4	3	0	1
9	5	5	0	0
10	9	9	0	0
Jumlah	Deteksi = 108	TP = 60	FP = 47	FN = 1

Dari tabel hasil deteksi di atas, selanjutnya dilakukan penghitungan nilai *precision*, *recall*, dan *f-1 score*. Hasilnya adalah sebagai berikut :

Tabel 4.2 Nilai Precision, Recall, dan F1-Score

Precision	Recall	F1-Score
56,07 %	98,36 %	71,42 %

5. Penutup

5.1. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang sudah dilakukan, maka dapat disimpulkan bahwa:

1. Pada penelitian ini, didapatkan nilai *recall* sebesar 98,36%, *precision* sebesar 56,07%, dan *f1-score* sebesar 71,42%.
2. Ukuran *cell* dalam HOG mempengaruhi panjang histogram dari fitur suatu citra. Semakin kecil ukuran *cell* menyebabkan semakin banyak *block* yang dibentuk pada suatu citra. Semakin banyak *block* maka semakin panjang histogram yang dibentuk sehingga semakin lama pula proses eksekusi program. Ukuran *cell* juga mempengaruhi tingkat akurasi pendeteksian. Dalam penelitian ini, ukuran *cell* yang paling optimal adalah 8x8 *pixel*.
3. *Kernel* yang dipakai dalam klasifikator SVM sangat berpengaruh dalam akurasi deteksi. Dalam penelitian kali ini, *kernel* yang paling optimal adalah linear dengan tingkat akurasi 95,45%.

5.2. Saran

Saran yang bisa dilakukan untuk penelitian selanjutnya dengan topik yang sama yaitu deteksi wajah:

1. Apabila menggunakan metode *Histogram of Oriented Gradients* dalam ekstraksi citri, yang harus diperhatikan adalah ukuran citra yang digunakan pada saat pembangunan model. Sebaiknya citra berukuran tidak terlalu kecil dan citra tersebut menggambarkan ciri objek yang dideteksi.
2. Apabila menggunakan metode *Histogram of Oriented Gradients*, perlu mencoba ukuran *cell* yang lain.
3. Perlu mencoba kernel lain dalam klasifikator SVM yang dipakai.
4. Algoritma *sliding windows* dan proses deteksi perlu dibuat lebih optimal, fleksibel dan efisien.
5. Untuk meningkatkan *precision*, perlu membatasi dataset positif dan memperbanyak dataset negatif.

Daftar Pustaka

- [1] Mohsen Ghorbani, Alireza Tavakoli Targhi, and Mohammad Mahdi Dehshibi, *HOG and LBP : Towards a Robust Face Recognition System*, ISPR Lab., Department of Computer Science Faculty of Mathematics, Shahid Behsti University, Iran, 2015.
- [2] Navneet Dalal and Bill Triggs, *Histograms of Oriented Gradients for Human Detection*, INRIA Rh[^]one-Alps, 655 avenue de l'Europe, Montbonnot 38334, France, 2005
- [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, University of Southampton, 1998.
- [5] Kadir, A., & Susanto, A., *Teori dan Aplikasi Pengolahan Citra*, Yogyakarta, 2013.
- [6] Cahyo Permata, I Ketut Eddy Purnama dan Muhtadin, *Deteksi Mobil Menggunakan Histogram of Oriented Gradient*, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Surabaya (ITS), Surabaya, 2013.
- [7] Yang, M.H., Kriegman, D. Ahuja, N., *Detecting Faces in Images : A Survey*, *IEEE Trans, Pattern Analysis and Machine Intelligence*, vol 24, 2002.
- [8] Hua-chun Yang, Xu An Wang, *Cascade face detection based on Histogram of Oriented Gradients and Support Vector Machine*, School of Life Science and technology Xidian University, Department of Information Engineering, Engineering University of Armed Police Xi'an, China, 2015.
- [9] G.C.-C.W.R.S.D.M.R.F.Prates, "Brazilian License Plate Detection Using Histogram of Oriented Gradients and Sliding Windows," *IJCSIT*, vol.5, no.6, 2014.
- [10] C.Molder, "An Automatic License Plate Recognition (ALPR) System".
- [11] D.Kurnianto, "Empat Tipe Dasar Citra Digital," 4 June 2013. [Online]. Available : <https://catatanpeneliti.wordpress.com/2013/06/04/empat-tipe-dasar-citra-digital/>. [Diakses 20 January 2017].
- [12] PT.Skill, "Pengertian OCR dan OMR," 2015.[Online]. Available:<http://www.skill.co.id/front/index.php/jakarta-tech-update/296-pengertian-ocr-dan-omr-2>. [Diakses 30 Desember 2017].
- [13] L.Eikvil, "Optical Character Recognition," 1993.
- [14] A.Azma. [Online]. Available: <http://www.abuazmashare.id/2016/07/arti-kode-dan-warna-plat-nomer.html>. [Diakses 7 Januari 2017].
- [15] M.L.Andika, "detikOto," 9 Mei 2011. [Online]. Available: <http://oto.detik.com/read/2011/05/09/153252/1635629/648/ukuran-baru-plat-nomor-kendaraan>. [Diakses 7 Januari 2017].
- [16] J.Canny, "A computational Approach to Edge Detection." *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.8, no.6, pp.679-689, 1986.