

ANALISIS KINERJA *HAND-TRACKING-BY-DETECTION* UNTUK TEKNOLOGI HOLOGRAM INTERAKTIF MENGGUNAKAN *SUPPORT VECTOR MACHINE*

PERFORMANCE ANALYSIS OF HAND-TRACKING-BY-DETECTION FOR INTERACTIVE HOLOGRAM TECHNOLOGY USING SUPPORT VECTOR MACHINE

Fajar Frimadana¹, Suryo Adhi Wibowo², Koredianto Usman³

^{1,2,3}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom, Bandung

¹fajardana@student.telkomuniversity.ac.id, ²suryoadhiwibowo@telkomuniversity.ac.id,

³korediantousman@telkomuniversity.ac.id

Abstrak

Hologram merupakan salah satu penerapan dari *Human Computer Interaction* yang dapat membuat penyampaian informasi secara 2D menjadi lebih menarik dan lebih efisien. Hologram dapat digerakkan menggunakan prinsip kerja *Hand Gesture Detection*. Prinsip kerja dari *Hand Gesture Detection* adalah sebuah sistem yang memungkinkan komputer dapat mendeteksi gerakan tangan dari manusia. Banyak metode yang dapat digunakan untuk menerapkan *Hand Gesture Detection* ini, tetapi masih terdapat parameter-parameter yang mengurangi tingkat akurasi deteksi seperti proses pengambilan citra atau *gesture* yang kurang tepat. Pada tugas akhir ini merancang sebuah sistem yang dapat mendeteksi gerak tangan dengan menggunakan metode *Histogram of Oriented Gradient* (HOG) untuk ekstraksi ciri citra dan *Support Vector Machine* (SVM) untuk klasifikasi citra. Tugas akhir ini menggunakan metode SVM karena pada penelitian sebelumnya yang menggunakan metode *adaboost* terdapat parameter-parameter yang membuat tingkat akurasi deteksi gerak tangan menurun, dan terdapat kondisi dimana tangan tidak terdeteksi oleh sistem. Tugas Akhir ini menggunakan *Cell Size*, *Block Size*, dan kernel linear yang digunakan sebagai konfigurasi untuk menganalisis hasil deteksi tangan. *Dataset* yang digunakan pada penelitian ini berupa 600 citra latihan, dan 300 citra yang diambil dari video berdurasi 10 detik sebagai data uji. Pada penelitian ini parameter yang digunakan untuk analisis adalah akurasi, *Intersection of Union* (IoU) dan waktu komputasi. Penelitian ini mendapatkan konfigurasi terbaik dengan menggunakan *Cell Size* 4×4, *Block Size* 7×7, dan kernel linear. Akurasi yang didapatkan yaitu 98%, IoU 0,88 dan waktu komputasi 44,95 milidetik.

Kata Kunci: *Hologram, Human Computer Interaction, Hand Gesture Detection, Histogram of Oriented Gradients, Support Vector Machine.*

Abstract

Hologram is one of the applications of *Human Computer Interaction* that can make delivering information in 2D more attractive and more efficient. Holograms can be moved using the working principle of *Hand Gesture Detection*. The working principle of *Hand Gesture Detection* is a system that allows computers to detect hand movements from humans. Many methods can be used to implement this hand gesture detection, but there are still parameters can reduce the level of accuration, such as the process of image capture or inaccurate gestures. This final project designs a system that can detect hand movements using the *Histogram of Oriented Gradient* (HOG) method for image feature extraction and *Support Vector Machine* (SVM) for image classification. This final project uses the SVM method because in previous research using the *Adaboost* method there were parameters that made the accuracy of detection from hand motion decreased, and there were conditions where the hand was not detected by the system. This Final Project uses *Cell Size*, *Block Size* and linear kernel which is used as a configuration to analyze the results of hand detection. The dataset used in this study was 600 training images, and 300 images taken from 10-second videos as test data. In this study the parameters used for analysis are accuracy, *Intersection of Union* (IoU) and computational time. This research gets the best configuration using *Cell Size* 4×4, *Block Size* 7×7, and linear kernel. The accuracy obtained is 98%, IoU 0.88 and computing time 44.95 milliseconds.

Keywords: *Hologram, Human Computer Interaction, Hand Gesture Detection, Histogram of Oriented Gradients, Support Vector Machine.*

1. Pendahuluan

Perkembangan teknologi yang terjadi di era globalisasi saat ini sangat pesat. Media pertukaran informasi yang dahulu memakai surat, sekarang *email* banyak digunakan karena lebih praktis dan efisien. Hologram dapat digunakan sebagai media pertukaran informasi yang lebih menarik karena pengguna dapat berinteraksi secara langsung dengan gambar 2D tanpa menggunakan *mouse* atau menyentuh layar komputer. Objek hologram dapat digerakkan secara *realtime* dengan menggunakan prinsip kerja deteksi gerak tangan (*hand gesture detection*). *Hand*

gesture detection menjadi salah satu cara yang banyak digunakan di bidang teknologi informasi yang membutuhkan interaksi antara manusia dan komputer, dan memberikan pengalaman yang menarik bagi pengguna [1]. *Hand gesture detection* banyak diimplementasikan pada teknologi-teknologi saat ini karena dianggap lebih efisien. Meskipun deteksi tangan banyak diteliti pada akhir-akhir ini, banyak faktor yang menyebabkan penurunan performa dari deteksi tangan, seperti pemilihan *gesture* yang kurang tepat atau terlalu kompleks, kualitas citra yang diambil dan sebagainya.

Sudah banyak penelitian mengenai penggunaan pola gerak tangan dalam berinteraksi terhadap komputer. Pada penelitian pengenalan pola gerak tangan dengan menggunakan metode *Hidden Markov Model* (HMM) dan memakai dataset yang cukup besar didapatkan akurasi yang tinggi, tetapi algoritma ini hanya bisa digunakan untuk pengenalan gerak tangan yang membutuhkan gerak minimum saja [2]. Pada penelitian yang lain, Li, L. Yang, dkk, penulis menggunakan algoritma *adaboost* untuk melakukan deteksi tangan, deteksi yang didapatkan cukup baik karena dapat mengatasi gangguan yang disebabkan oleh masalah kulit, tetapi untuk situasi seperti benda yang mirip dengan tangan belum sepenuhnya dapat diatasi [3].

Tugas Akhir ini penulis mengimplementasikan *hand gesture detection* pada hologram dengan menggunakan metode *Histogram of Oriented Gradient* sebagai ekstraksi citra dan *Support Vector Machine* untuk proses klasifikasi citra. Parameter penting yang mempengaruhi performansi sistem yaitu *cell size*, *block size*, dan kernel yang digunakan saat pengujian. Pada Tugas Akhir ini dilakukan uji coba pada setiap nilai parameter yang sudah ditentukan. Hasil dari pengujian akan dibandingkan dan dianalisis untuk menentukan parameter terbaik sehingga dapat diterapkan pada sistem.

2. Tinjauan Pustaka

Pada bab ini menjelaskan tentang tinjauan pustaka dan konsep dasar mengenai Hologram, Pengolahan citra, HOG, dan SVM.

2.1 Hologram

Istilah *Holografi* merupakan bahasa dari Yunani yang terdiri dari kata *Holo* yang berarti keseluruhan dan kata *Grafik* yang berarti catatan. Secara keseluruhan kata *Holografi* memiliki arti seluruh catatan, dan *Holografi* mengacu pada semua informasi yang ada pada suatu objek. Hologram merupakan gambar tiga dimensi yang terjadi karena terdapat pantulan sinar laser terhadap bahan fotografi tanpa menggunakan kamera [4].

2.2 Pengolahan Citra (*Image Processing*)

Image Processing adalah analisis dan manipulasi citra digital, *image processing* digunakan untuk meningkatkan kualitas citra. Teknik ini dapat diterapkan dalam berbagai bidang seperti deteksi dan pencocokan objek, pengurangan latar belakang dalam suatu video, mengukur volume jaringan, sistem kontrol lalu lintas, pengenalan wajah, pengenalan iris, dan pencitraan medis. Teknik *image processing* dapat mengatasi masalah seperti hilangnya kualitas citra [5].

Untuk mendapatkan dan mengolah sebuah informasi yang ada pada citra, *computer vision* harus terdiri dari banyak fungsi pendukung antara lain, proses penangkapan citra (*image acquisition*), pengolahan citra (*image processing*), analisis data citra (*image analysis*), dan proses pemahaman citra (*image understanding*).

2.3 Color Space

2.3.1 Red Green Blue (RGB)

Citra dengan ruang warna RGB adalah citra yang paling banyak digunakan pada gambar digital. RGB adalah ruang warna yang paling umum digunakan untuk representasi gambar digital karena sesuai dengan tiga warna utama yang dicampur untuk ditampilkan pada monitor atau perangkat serupa. Ruang warna RGB terdiri dari 3 warna primer yaitu *red* (merah), *green* (hijau), dan *blue* (biru). Setiap piksel dibentuk dari tiga komponen tersebut. Penggabungan dari tiga warna primer akan didapatkan warna sekunder lainnya [6]. Setiap komponen warna RGB menggunakan 8 bit, dimana nilai warna setiap komponen antara 0 sampai 255. Sehingga citra RGB seringkali disebut sebagai citra dengan intensitas 24 bit. Jumlah warna RGB dapat dihitung $256^3 = 16.777.216$ warna.

2.3.2 Citra Keabuan (*Grayscale*)

Citra *grayscale* atau citra keabuan adalah citra yang hanya memiliki satu nilai kanal pada setiap pikselnya. Nilai kanal tersebut menunjukkan intensitas warna dari merah, hijau, dan biru. Nilai intensitas warna pada citra *grayscale* didasarkan pada derajat keabuan. Jumlah warna pada citra *grayscale* adalah 256 karena jumlah bit pada citra *grayscale* adalah 8, sehingga jumlah warnanya yaitu $2^8 = 256$ (rentan warna dari 0 – 255). Nilai 0 pada citra *grayscale* menyatakan warna hitam dan 255 menyatakan warna putih [7]. Perhitungan untuk mengubah citra RGB menjadi citra *grayscale* dapat dilihat pada persamaan (2.1).

$$gryascale = (0,2989 \times R) + (0,8570 \times G) + (0,1140 \times B), \quad (2.1)$$

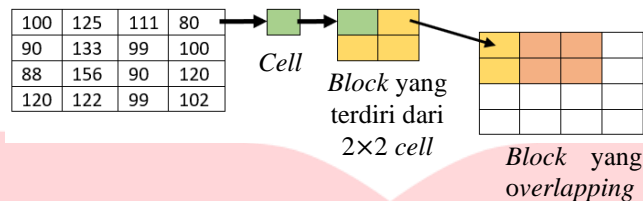
2.4 Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) adalah salah satu metode ekstraksi citra yang sering digunakan untuk mendeteksi suatu objek. Prinsip kerja dari HOG adalah menghitung nilai gradien setiap piksel dalam daerah tertentu. Setiap citra memiliki karakteristik yang ditunjukkan oleh distribusi gradien. Untuk mendapatkan karakteristik tersebut tahap awal dari metode HOG adalah menghitung nilai gradien dari setiap piksel dari citra,

kemudian membagi citra menjadi *cell*, membentuk histogram dari *cell* yang sudah didapatkan, membentuk blok dari setiap histogram, dan terakhir melakukan normalisasi pada setiap blok [8]. Tahapan algoritma HOG adalah sebagai berikut [8] :

1. Menentukan ukuran *cell* dan *block*

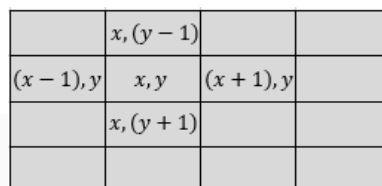
Proses ini menentukan blok dan *cell* pada suatu citra. *Cell* dapat terdiri dari beberapa piksel misalnya 2×2 , 4×4 , 8×8 , 16×16 , dan lainnya. Sedangkan *block* terdiri dari beberapa *cell* dan antar *block* saling *overlapping* [9].



Gambar 2.1 Cell Yang Menyusun Block [9].

2. Menghitung nilai gradien (G) dan orientasi gradien (θ)

Gradien merupakan perubahan setiap fungsi intensitas dari citra. Sebelum menghitung gradien, citra akan diubah terlebih dahulu ke bentuk *grayscale* agar tidak memperhatikan intensitas yang berbeda dari ruang warna (RGB). Metode yang digunakan adalah metode 1-D *centered*, dengan matriks $[-1,0,1]$ yang diterapkan pada arah *vertical* dan *horizontal* di suatu piksel [10]. Gambar 2.2 menunjukkan visualisasi perhitungan gradien.



Gambar 2.2 Visualisasi Perhitungan Gradien [10].

3. Menentukan orientasi bin

Pada metode HOG, citra dibagi menjadi beberapa *cell* yang sudah ditentukan. Selanjutnya membuat histogram untuk setiap *cell* agar dapat diketahui berapa nilai dalam setiap *cell*. Untuk membuat sebuah histogram diperlukan nilai gradien yang didapatkan dari perhitungan di setiap piksel. Karena di setiap *cell* memiliki gradien yang berbeda-beda maka diperlukan bin orientasi. Besarnya nilai gradien dari piksel dalam sel, jika nilai bin = 9 maka terdapat 9 daerah pembagian pada histogram dalam rentang $0^\circ - 180^\circ$ ($0^\circ - 20^\circ$, $20^\circ - 40^\circ$, $40^\circ - 60^\circ$, $60^\circ - 80^\circ$, $80^\circ - 100^\circ$, $100^\circ - 120^\circ$, $120^\circ - 140^\circ$, $140^\circ - 160^\circ$, $160^\circ - 180^\circ$) setiap daerah memiliki selisih 20° [11].

4. Menghitung vektor HOG

Untuk mendapatkan nilai ekstraksi, dilakukan perhitungan vektor pada keseluruhan. Vektor pada setiap *block* digabungkan dengan *block-block* yang lain.

5. Melakukan Normalisasi Block

Untuk memaksimalkan *feature* HOG maka dibutuhkan *block* yang terdiri dari gabungan beberapa *cell* dengan ukuran 2×2 atau 8×8 piksel. *Block* ini berguna untuk menghindari perubahan intensitas dan kontras pada sebuah citra. Terdapat beberapa skema yang dapat digunakan untuk normalisasi histogram level *block*. Salah satu rumus yang digunakan untuk normalisasi blok adalah *L1-sqrt* seperti pada persamaan (2.2).

$$v = \sqrt{\frac{v}{\|v\| + \epsilon}}, \quad (2.2)$$

Dengan v adalah nilai fitur yang mengandung histogram dan ϵ adalah konstanta yang bernilai 0,1. Nilai ϵ digunakan untuk menghindari terjadinya pembagian dengan nilai 0.

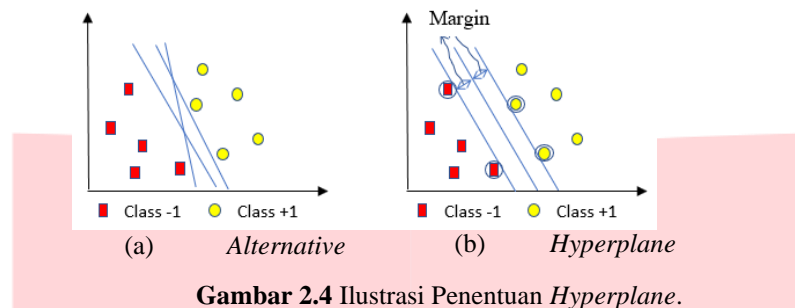
2.5 Support Vector Machine

Support Vector Machine (SVM) adalah salah satu metode dari *machine learning* yang memiliki tujuan untuk mendapatkan *hyperplane* terbaik yang memisahkan dua buah *class* di *input space*. Prinsip dasar dari SVM sendiri adalah *linear classifier*, SVM juga dapat digunakan untuk memisahkan *class* yang memiliki sifat *non-linear* dengan menggunakan *kernel trick*. *Hyperplane* adalah garis pemisah antara *class* yang satu dengan yang lain.

Hyperplane didapatkan dengan cara mengukur *margin hyperplannya* dan mencari titik maksimalnya. *Margin* merupakan jarak antara titik *class* yang paling dekat dengan *hyperplane* [12].

2.5.1 Linear Classification SVM

Konsep SVM yaitu mencari *hyperplane* terbaik yang memisahkan dua buah *class* pada input space [12]. Pada gambar 2.4 terdapat beberapa patern yang menunjukkan anggota dari dua buah *class*, pattern yang terdapat di *class* -1 disimbolkan dengan warna merah yang berbentuk kotak, sedangkan pada *class* +1 disimbolkan dengan warna kuning yang berbentuk lingkaran.



Gambar 2.4 Ilustrasi Penentuan *Hyperplane*.

Berbagai alternative *hyperplane* yang terdapat pada gambar 2.6(a). *Hyperplane* pemisah terbaik antara *class* -1 dan *class* +1 didapatkan dengan cara mengukur *margin hyperplane* dan mencari titik maksimalnya. Garis solid yang terdapat pada gambar 2.6(b) merupakan *hyperplane* terbaik yang didapatkan, sedangkan titik *class* yang dilingkari merah merupakan *support vector*. Usaha yang dilakukan untuk mendapatkan *hyperplane* ini merupakan inti proses dari SVM [12]. *Hyperplane* dapat ditemukan dengan persamaan

$$\vec{w} \cdot \vec{x} + b = 0, \quad (2.3)$$

dengan \vec{x} adalah data *input*, \vec{w} dan b adalah parameter yang nanti akan dicari nilainya.

Pattern yang termasuk pada *class* -1 dapat dirumuskan dengan persamaan

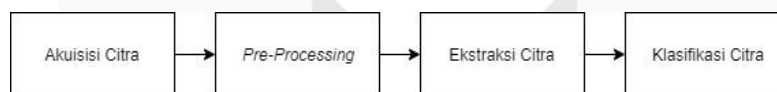
$$\vec{w} \cdot \vec{x} + b \leq -1, \quad (2.4)$$

Sedangkan *pattern* yang termasuk *class* +1 dirumuskan dengan persamaan

$$\vec{w} \cdot \vec{x} + b \geq +1, \quad (2.5)$$

3. Desain Model Sistem

Perancangan pada penelitian ini menggunakan metode *Histogram of Oriented Gradients* untuk ekstraksi citra dan menggunakan metode *Support Vector Machine* untuk mengklasifikasi citra. Seperti pada gambar 3.1, tangan dapat dideteksi melalui tahapan sebagai berikut: *input* citra atau akuisisi citra, tahap *pre-processing* untuk meningkatkan kualitas citra, kemudian ekstraksi ciri citra menggunakan HOG untuk mendapatkan ciri-ciri dari citra yang nantinya digunakan sebagai *input* pada proses klasifikasi, dan tahap terakhir adalah proses klasifikasi citra menggunakan metode SVM untuk menentukan pola tangan yang digunakan pada sistem.



Gambar 3.1 Desain Perancangan Sistem.

3.1 Pengambilan Citra atau Akuisisi Citra

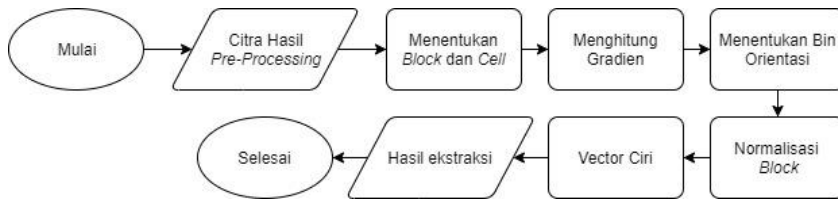
Pada Tugas akhir ini dataset yang dipakai terdiri dari 2 *class*, yaitu telapak tangan dan kepalan tangan. Dataset terdiri dari 600 citra latih dan 300 citra uji. Citra telapak tangan berfungsi sebagai *pointer* untuk menggerakkan *cursor* pengganti *mouse*. Sedangkan citra kepalan tangan berfungsi sebagai *pointer* untuk melakukan *drag*, *drop*, dan *click*.

3.2 Pre-Processing

Alur pada tahap *pre-processing* yaitu: proses *cropping* citra pada bagian objek yang ingin dideteksi, setelah *crop* citra di *resize* menjadi menjadi ukuran 200×200 piksel, kemudian setelah *resize* citra diubah menjadi citra *grayscale*.

3.3 Ekstraksi Ciri

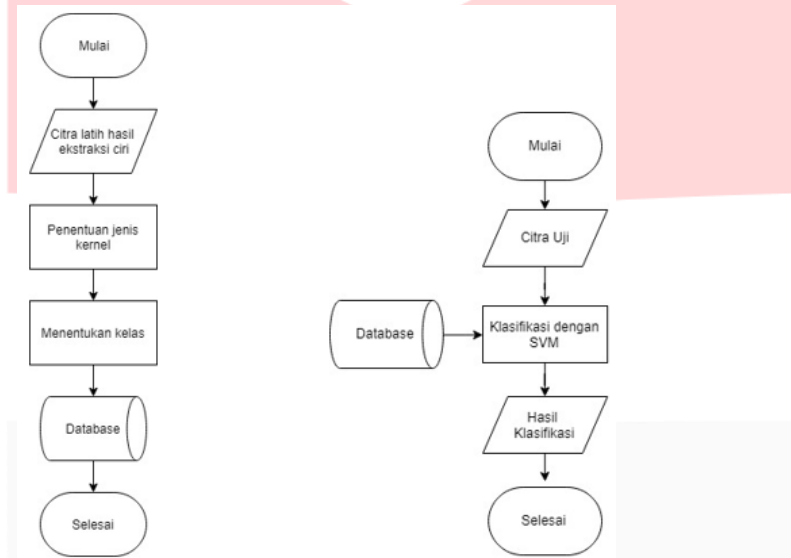
Ekstraksi citra diperlukan untuk mendapatkan ciri ciri dari suatu citra yang membedakan dengan citra lainnya. Pada penelitian ini ekstraksi citra dilakukan dengan menggunakan metode *Histogram of Oriented Gradients*. Diagram alir proses ekstraksi ciri citra menggunakan HOG dapat dilihat pada gambar 3.2.



Gambar 3.2 Diagram Alir Ekstraksi Ciri HOG.

3.4 Klasifikasi

Pada proses klasifikasi, dibagi menjadi klasifikasi SVM untuk data latih dan klasifikasi SVM untuk data uji. Gambar 3.3(a) merupakan diagram alir klasifikasi SVM untuk data latih dan gambar 3.3(b) merupakan diagram alir klasifikasi SVM untuk data uji.



(a) Diagram Alir SVM data

(b) Diagram Alir SVM data

Gambar 3.3 Diagram Alir Ekstraksi Ciri HOG.

4. Analisis Parameter Kinerja

Desain sistem akan dirancang untuk melakukan *hand-tracking-by-detection* dengan metode SVM. Parameter kinerja akan dianalisis dan diperhitungkan untuk menilai kinerja sistem. Tugas Akhir akan menguji 3 parameter kinerja, yaitu akurasi dan waktu komputasi.

1. *Intersection of Union*

Intersection of Union (IoU) adalah parameter yang digunakan untuk menghitung akurasi ketepatan *bounding box* dengan *groundtruth box* dari objek. IoU mempengaruhi akurasi dari sistem, sistem akan dikatakan akurat untuk mendeteksi objek jika $IoU \geq 0,5$. IoU dapat dihitung menggunakan persamaan 4.1.

$$D = \frac{B_{GT} \cap B_{AC}}{B_{GT} \cup B_{AC}}, \tag{4.1}$$

dengan B_{GT} merupakan *groundtruth*, dan B_{AC} merupakan *bounding box actual*.

2. Akurasi

Salah satu parameter kinerja sistem yaitu akurasi. Parameter ini akan mengukur kebenaran dalam mendeteksi bentuk jari. Akurasi dapat dihitung menggunakan persamaan 4.2.

$$\eta = \frac{C}{T} \times 100\%, \tag{4.2}$$

dengan η merupakan akurasi, C adalah jumlah data yang akurat, dan T adalah total keseluruhan data.

3. Waktu Komputasi

Waktu komputasi merupakan waktu yang dibutuhkan sistem dalam menjalankan pengujian. Waktu komputasi dapat dihitung menggunakan persamaan 4.3.

$$W = W_{akhir} - W_{awal}, \tag{4.3}$$

dengan W merupakan waktu komputasi sistem, W_{akhir} adalah waktu akhir pengujian sistem, W_{awal} adalah waktu awal pengujian sistem.

5. Hasil dan Analisis

Bagian ini berisi tentang pengujian terhadap algoritma HOG dan SVM menggunakan parameter performansi akurasi dan waktu komputasi. Pengujian menggunakan 3 konfigurasi parameter, diantaranya: *cell size*, *block size* dan kernel linear.

5.1 Skenario Pengujian Sistem

Tugas Akhir ini menggunakan 600 citra latih dan video 10 detik sebagai data uji. Pengujian menggunakan 3 skenario yaitu:

1. Skenario 1: pengujian terhadap parameter IoU
Pada skenario pertama menunjukkan perbedaan IoU yang didapatkan dari *Cell Size* 2×2, 4×4, dan 8×8 piksel dengan menggunakan *Block size* 3×3 *cell*, 5×5 *cell*, 7×7 *cell*, *Bin Number* 9, dan kernel linear. Pengujian dilakukan untuk menentukan IoU dari citra pola tangan dengan menggunakan video berukuran 1280×720 piksel berdurasi 10 detik.
2. Skenario 2: pengaruh *cell size* 2×2, 4×4, dan 8×8 pada *block size* 5×5
Pada skenario kedua menunjukkan perbedaan akurasi yang didapatkan pada *Cell Size* 2×2, 4×4, dan 8×8 dengan menggunakan *Block size* 3×3, 5×5, 7×7, *Bin Number* 9, dan kernel linear. Pengujian dilakukan untuk menentukan akurasi dari citra pola tangan dengan menggunakan video berukuran 1280×720 piksel berdurasi 10 detik.
3. Skenario 3: pengaruh *cell size* 2×2, 4×4, dan 8×8 pada *block size* 7×7
Pada Skenario ketiga menunjukkan perbedaan waktu komputasi yang didapatkan pada *Cell Size* 2×2, 4×4, dan 8×8 dengan menggunakan *Block size* 3×3, 5×5, 7×7, *Bin Number* 9, dan kernel linear. Pengujian dilakukan untuk menentukan waktu komputasi dari citra pola tangan dengan menggunakan video berukuran 1280×720 piksel berdurasi 10 detik.

6. Data Hasil Pengujian Sistem

Setelah dilakukan pengujian maka didapatkan hasil dari seluruh konfigurasi sistem. Hasil pengujian skenario pertama dapat dilihat pada tabel 6.1 dan 6.2.

Tabel 6.1 Hasil Pengujian Skenario Pertama Citra Pola 1.

Ukuran Cell	Ukuran Block	IoU
2×2	3×3	0,697
4×4		0,713
8×8		0,705
2×2	5×5	0,738
4×4		0,771
8×8		0,745
2×2	7×7	0,759
4×4		0,881
8×8		0,872

Tabel 6.2 Hasil Pengujian Skenario Pertama Citra Pola 2.

Ukuran Cell	Ukuran Block	IoU
2×2	3×3	0,836
4×4		0,848
8×8		0,841
2×2	5×5	0,852
4×4		0,87
8×8		0,86
2×2	7×7	0,885
4×4		0,89
8×8		0,89

Hasil skenario kedua dapat dilihat pada tabel 6.3 dan 6.4.

Tabel 6.3 Hasil Pengujian Skenario Kedua Citra Pola 1.

Ukuran Cell	Ukuran Block	Akurasi (%)
2×2	3×3	82,3
4×4		84,3
8×8		83
2×2	5×5	87,66
4×4		92
8×8		88,66
2×2	7×7	96
4×4		98
8×8		96

Tabel 6.4 Hasil Pengujian Skenario Kedua Citra Pola 2.

Ukuran Cell	Ukuran Block	Akurasi (%)
2×2	3×3	88
4×4		88,3
8×8		88
2×2	5×5	88,3
4×4		92
8×8		90
2×2	7×7	94
4×4		98
8×8		96

Hasil skenario ketiga dapat dilihat pada tabel 6.5 dan 6.6.

Tabel 6.5 Hasil Pengujian Skenario Ketiga Citra Pola 1.

Ukuran Cell	Ukuran Block	Waktu Komputasi (ms)
2×2	3×3	504,7
4×4		144,49
8×8		45,66
2×2	5×5	250,4
4×4		51,3
8×8		25,13
2×2	7×7	241,85
4×4		44,95
8×8		24,28

Tabel 6.6 Hasil Pengujian Skenario Ketiga Citra Pola 2.

Ukuran Cell	Ukuran Block	Waktu Komputasi (ms)
2×2	3×3	682,2
4×4		174,38
8×8		45,51
2×2	5×5	586,82
4×4		150,85
8×8		38,72
2×2	7×7	235,04
4×4		51,67
8×8		22,42

Berdasarkan hasil uji coba pada skenario pertama yang dapat dilihat pada tabel 6.1 dan 6.2 IoU tertinggi terdapat pada konfigurasi *cell size* 4×4 dan *block size* 7×7 dengan nilai 0.881 dan 0,89. Hal ini menunjukkan bahwa konfigurasi tersebut memiliki jarak yang tidak jauh antara *bounding box groundtruth* dengan *predicted bounding boxes* jika dibandingkan dengan konfigurasi lainnya. Rentang nilai IoU adalah 0 sampai 1. Berdasarkan hasil uji coba pada skenario kedua yang dapat dilihat pada tabel 6.3 dan 6.4, akurasi tertinggi yang dicapai oleh sistem adalah 98%. tertinggi didapatkan dengan konfigurasi *cell size* 4×4 dan *block size* 7×7. Sistem dikatakan akurat dalam mendeteksi objek jika memiliki $\text{IoU} \geq 0.5$. Setelah dilakukan uji coba pada skenario ketiga, konfigurasi *cell size* 8×8 dan *block size* 7×7 memiliki waktu komputasi sistem paling cepat karena semakin besar ukuran *cell* membuat proses ekstraksi semakin sedikit, sehingga sistem dapat melakukan pengujian dengan cepat.

7. Kesimpulan

Setelah dilakukan pengujian terhadap parameter yang ditentukan, didapatkan hasil analisa dari setiap parameter yang diuji. Hasil yang memiliki akurasi paling tinggi terdapat pada pengujian dengan *cell size* 4×4 dengan *block size* 7×7 yaitu 98%. Waktu komputasi tercepat terdapat pada pengujian *cell size* 8×8 dengan *block size* 7×7 yaitu 24,28 mili detik. Data akurasi dan waktu terbaik dapat dilihat pada tabel 6.3.

Dari semua hasil pengujian, *cell size* dengan ukuran 4×4 piksel memiliki tingkat akurasi yang paling tinggi jika dibandingkan dengan ukuran lainnya, karena dengan ukuran tersebut proses ekstraksi ciri tidak membuang banyak informasi penting yang terkandung dalam citra. Tetapi dengan ukuran *cell size* 2×2 yang seharusnya memiliki tingkat akurasi lebih baik dari 4×4 didapatkan tingkat akurasi yang lebih rendah, hal tersebut dapat terjadi karena proses ekstraksi ciri citra sangat detail sehingga sistem memiliki acuan bahwa bentuk pola tangan harus sepenuhnya sesuai dengan citra yang dilatih. *Cell size* dengan ukuran 8×8 piksel memiliki waktu komputasi yang lebih cepat jika dibandingkan dengan ukuran lainnya karena dengan ukuran tersebut proses ekstraksi berjalan dengan cepat tetapi membuang banyak informasi penting yang terkandung dalam citra.

8. Daftar Pustaka:

- [1] T. Q. Vinh and N. T. Tri, "Hand Gesture Recognition Based on Depth Image Using Kinect Sensor," in *2015 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)*, 2015.
- [2] A. Ghotkar, P. Vidap, and K. Deo, "Dynamic Hand Gesture Recognition using Hidden Markov Model by Microsoft Kinect Sensor," *Int. J. Comput. Appl.*, vol. 150, no. 5, pp. 5–9, 2016.
- [3] H. Li, L. Yang, X. Wu, and J. Zhai, "Hands Detection Based on Statistical Learning," 2012.
- [4] A. H. Awad, "The First Design of a Smart Hologram for Teaching," *Adv. Sci. Eng. Technol. Int. Conf.*, pp. 1–4, 2018.
- [5] P. Paulraj, "Digital Image Processing Techniques," vol. 5, no. May, 2016.
- [6] S. K. Singh, D. S. Chauhan, M. Vatsa, and R. Singh, "A Robust Skin Color Based Face Detection Algorithm," vol. 6, no. 4, pp. 227–234, 2003.
- [7] C. Solomon and T. Breckon, *Fundamentals of Digital Image Processing*. 2011.
- [8] C. Q. Lai and S. S. Teoh, "An Efficient Method of HOG Feature Extraction Using Selective Histogram Bin and PCA Feature Reduction," vol. 16, no. 4, pp. 101–108, 2016.
- [9] C. Permata and I. K. Eddy, "Deteksi Mobil Menggunakan Histogram of Oriented Gradient," pp. 1–6, 2013.
- [10] C. Li, "A New Method Combining HOG and Kalman Filter for Video-based Human Detection and Tracking," pp. 290–293, 2010.
- [11] I. Conference and S. Processing, "Influence of binning in Histograms of Oriented Gradients method

representation,” pp. 15–19, 2017.

[12] A. S. Nugroho, A. B. Witarto, and D. Handoko, “Support Vector Machine,” 2003.

[13] J. Shukla, “A Method for Hand Gesture Recognition,” 2014.

