

Deteksi dan Tracking Pemain Sepakbola menggunakan *Histogram of Oriented Gradients (HOG)* dan *Kalman Filter*

Kurniawan Nur Ramadhani ^{#1}, Ade Saepul Mugni ^{#2}, Mohamad Syahrul Mubarak ^{#3}

[#] Fakultas Informatika, Universitas Telkom

Jl. Telekomunikasi No. 01, Terusan Buah Batu, Bandung 40257

¹ kurniawannr@telkomuniversity.ac.id

² @ademugni893@gmail.com

³ msyahrulmubarak@gmail.com

Abstract

In this research, we built a football player tracking system on video. This research used Histogram of Oriented Gradient (HOG) feature extraction because it performs well in unstable lighting intensity condition. In addition to detecting football players, the system also performed classification on player team. Team classification is performed using clustering on color moment feature vector. To maintain the detection performance, the system conducted tracking evaluation using Kalman Filter. From the testing process, the system performed the highest F1-score of 0.87 (0-1 scale) with various video lighting conditions.

Keywords: *Football Player, Tracking, Histogram of Oriented Gradient, Kalman Filter, Color Moment, K-Means, Support Vector Machine.*

Abstrak

Dalam penelitian ini, dibangun sebuah sistem untuk melakukan *tracking* pemain sepakbola pada data video. Penelitian ini menggunakan ekstraksi ciri *Histogram of Oriented Gradient (HOG)* yang cocok untuk digunakan pada kondisi intensitas pencahayaan tidak stabil. Selain melakukan deteksi pemain bola, dalam penelitian ini dilakukan pengklasifikasian tim menggunakan *clustering* pada vektor ciri *color moment*. Untuk menjaga performansi deteksi, dilakukan evaluasi *tracking* menggunakan *Kalman Filter*. Berdasarkan hasil penelitian, sistem *tracking* yang dibangun memberikan performansi F1-score tertinggi mencapai 0.87 (skala 0-1) dengan berbagai kondisi pencahayaan video.

Kata Kunci: *Pemain bola, Tracking, Histogram of Oriented Gradient, Kalman Filter, Color Moment, K-Means, Support Vector Machine.*

I. PENDAHULUAN

Data statistik pemain sepakbola di lapangan sangat dibutuhkan bagi sebuah tim sepakbola untuk menentukan strategi permainan. Beberapa data statistik yang dapat diperoleh dari hasil pertandingan di lapangan adalah area jangkauan pemain dan pola pergerakan suatu tim di lapangan. Beberapa teknologi telah diusulkan untuk tujuan ini, seperti penggunaan sensor [1]. Penggunaan sensor mampu mendeteksi kebutuhan data statistik pemain yang detail. Namun, teknologi sensor ini juga harus dituntut agar tidak mengubah lingkungan atau aktivitas yang lazim dilakukan pada pertandingan olah raga, contohnya penggunaan *device* sensor posisi yang tertanam pada pemain harus disesuaikan agar tidak mengganggu aktivitas di lapangan [1]. Solusi lain yang dapat dijadikan pilihan adalah dengan bantuan teknologi *computer vision*, khususnya mengenai deteksi objek melalui citra. Hal ini tidak akan mempengaruhi sedikit pun aktivitas para pemain bola di lapangan. Lebih jauh lagi, kemampuan sensor dan kemampuan *computer vision* dapat diintegrasikan, seperti pada penelitian [2], sebagai sebuah sistem untuk meningkatkan performansi analisis aktivitas pemain sepakbola di lapangan.

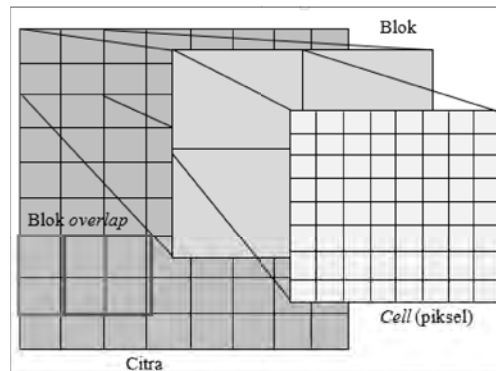
Teknologi *computer vision* untuk mendeteksi sekaligus *tracking* objek sudah banyak diusulkan dan diterapkan pada berbagai permasalahan, seperti orang, kendaraan, dan objek-objek lainnya. Pemanfaatan deteksi dan *tracking* pada orang dapat diterapkan pada permasalahan olahraga, salah satunya olahraga sepakbola, seperti pada penelitian [3-7]. Deteksi dan *tracking* pemain sepakbola ini berasal dari video hasil rekaman pertandingan. Secara umum, metode yang digunakan pada kasus ini dapat dipecah menjadi dua tahapan, deteksi pemain kemudian *tracking* pemain. Deteksi objek melalui sebuah video dapat dibagi menjadi dua cara, yaitu *background subtraction* dan metode berbasis ciri. Pada penelitian sebelumnya [3, 4], digunakan metode berbasis ciri, yaitu ciri *Histogram of Oriented Gradients* (HOG) dengan *classifier Support Vector Machine* (SVM) dan ciri Haar. Sedangkan pada penelitian sebelumnya [5, 6] menggunakan metode *background subtraction*. Namun, deteksi objek pada video berbasis *background subtraction* ini bisa saja rentan terhadap perubahan intensitas cahaya yang terjadi pada video sepakbola. Hal ini dapat diatasi dengan beralih ke metode berbasis ciri, seperti pada penelitian [3, 4] yang telah melakukan pengujian dengan menggunakan berbagai macam kondisi video sepakbola, termasuk variasi pencahayaan dengan nilai performansi lebih dari 90 %. Di samping deteksi objek pemain, penggunaan ciri warna, baik untuk membedakan tim satu dengan tim lain maupun untuk meningkatkan akurasi deteksi cocok untuk diterapkan pada kasus sepakbola, seperti pada penelitian [4-7]. Hal ini karena kasus sepakbola identik dengan penggunaan warna kostum. Metode HOG dapat dikombinasikan dengan ciri warna, seperti pada penelitian [3], namun ciri warna ini digunakan untuk meningkatkan akurasi deteksi. Lain hanya dengan penelitian [4-6], ciri warna tidak digunakan untuk meningkatkan akurasi deteksi melainkan digunakan untuk membedakan antara pemain dari tim satu dengan tim lain.

Proses lain dalam mendeteksi objek adalah *tracking*. Untuk *tracking* objek, khususnya kasus pemain sepakbola, penelitian sebelumnya [4] menggunakan metode Kalman Filter. Metode ini dapat meningkatkan kemampuan dalam hal deteksi maupun *tracking*. Dalam prosesnya, metode Kalman melakukan prediksi dan koreksi terhadap objek yang terdeteksi. Pergerakan objek pada video dapat diatasi menggunakan Kalman yang berbasis prediksi dan koreksi. Hal ini dilakukan agar kemampuan deteksi dan *tracking* dapat terjaga melalui proses prediksi dan koreksi.

II. DASAR TEORI

A. *Histogram of Oriented Gradients* (HOG)

Histogram of Oriented Gradients (HOG) merupakan sebuah *feature descriptor* berbasis bentuk orientasi gradien dari suatu *region*. Ciri ini menghasilkan persebaran nilai gradien berdasarkan arah tepi (*edge direction*) dari suatu citra. Pembuatan deksripsi ciri HOG diawali membagi citra ke dalam *cells* dengan ukuran piksel *cell* tertentu. Beberapa *cells* dikelompokkan ke dalam satu blok. Setiap satu blok menghasilkan ciri HOG.



Gambar 1 Ilustrasi Blok dan Cell HOG

HOG dapat memproses citra dalam format *Red Green Blue* (RGB) atau *grayscale*. Selain dari segi ruang warna, ada hal yang perlu diperhatikan pula dalam menentukan citra masukan, yaitu ukuran dari citra tersebut. Ukuran citra ini nantinya akan berhubungan dengan ukuran *detection window* pada saat *testing*. *Detection window* ini yang nantinya berfungsi sebagai *template* pencarian objek pada saat proses deteksi sistem berjalan. Ukuran yang lazim digunakan pada deteksi orang adalah 64×128 piksel. Namun, tidak menutup kemungkinan ukuran yang digunakan berbeda, tergantung dengan kasus yang diangkat. Selain ruang warna dan ukuran, normalisasi *gamma* pun dapat menjadi pertimbangan ketika membangun deksripsi ciri HOG. Normalisasi ini dilakukan agar pencahayaan pada citra masukan dikoreksi. Hal ini dikarenakan nilai *gamma* di setiap mesin berbeda.

1. Perhitungan Gradien

Perhitungan gradien dilakukan dengan cara mengurangi nilai piksel selanjutnya dengan nilai piksel sebelumnya pada masing-masing sumbu x dan y menggunakan operator $[-1,0,1]$ dan $[-1,0,1]^T$, $[-1,1]$, $[1,-8,0,8,1]$ [5]. Selain dengan operator 1-D tersebut, dapat juga menggunakan operator 2-D, seperti operator Sobel, dan lain-lain. Dari operator tersebut, dapat dicari gradien terhadap x , $px(x,y)$ (horizontal) dan gradien terhadap y , $py(x,y)$ (vertikal) dengan (x,y) merupakan nilai koordinat piksel pada gambar. Berikut perhitungan matematis menggunakan operator 1-D *centered* [6].

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

$$py(x,y) = I(x,y+1) - I(x,y-1) \quad (2)$$

I merupakan matriks dari citra, dengan $I(x,y)$ merupakan nilai *gray* pada piksel. Dari kedua persamaan di atas, didapat *magnitude*, $G(x,y)$ dan orientasi dari gradien, dapat dilihat pada persamaan berikut.

$$G(x,y) = \sqrt{px^2(x,y) + py^2(x,y)} \quad (3)$$

$$\text{orientasi}(x,y) = \arctan \frac{py(x,y)}{px(x,y)} \quad (4)$$

2. Orientation Binning

Tahap ini merupakan penempatan nilai gradien sesuai dengan arah orientasinya untuk membuat distribusi histogram. Orientasi *bins* yang digunakan pada histogram dapat memiliki *angle* antara $0^\circ - 180^\circ$ (*unsigned gradient*) atau $0^\circ - 360^\circ$ (*signed gradient*) [5]. Jumlah orientasi *bins* yang memiliki tingkat performansi yang baik ketika digunakan pada deteksi orang adalah 9 *bin* dengan *range* $0^\circ - 180^\circ$, seperti pada penelitian [5]. Pembangunan histogram berdasarkan orientasi dari nilai gradien. Jika orientasinya sesuai dengan *bin* yang dituju, maka nilai gradien pada saat itu diakumulasikan ke *bin* yang dituju tersebut. Pembuatan histogram dilakukan pada area *cell* yang berada pada blok yang terbagi di dalam ukuran *detection window* (ukuran citra). Setelah histogram dibuat, dilakukan normalisasi. Normalisasi dilakukan baik pada blok, maupun pada keseluruhan deskriptor HOG. Ada beberapa normalisasi yang dapat digunakan, seperti pada penelitian sebelumnya [5].

$$L2norm: v = \frac{v}{\sqrt{\|v\|_2^2 + e^2}} \quad (5)$$

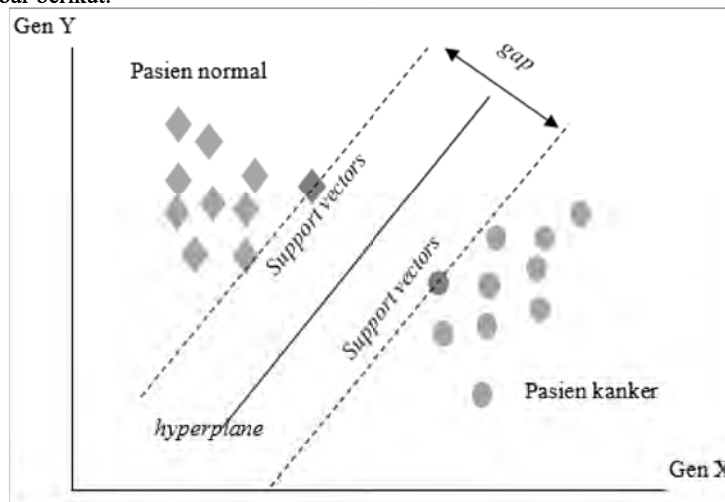
$$L1norm: v = \frac{v}{(\|v\|_1 + e)} \quad (6)$$

$$L1sqrt: v = \sqrt{\frac{v}{(\|v\|_1 + e)}} \quad (7)$$

Nilai v dalam persamaan ini merupakan nilai histogramnya atau nilai yang akan dinormalisasi. Sedangkan e merupakan konstanta yang digunakan agar tidak terjadi pembagian dengan bilangan nol.

B. Support Vector Machine (SVM)

SVM merupakan algoritma klasifikasi dengan skema pembelajaran terbimbing yang bertujuan untuk mencari *hyperplane* yang memisahkan kelas-kelas dengan jarak (*margin/gap*) maksimal antara *border-line* (*support vectors*) kelas satu dengan *border-line* kelas lain [7]. Misalkan *dataset* dideskripsikan dengan dua gen, gen X dan gen Y. Dari *dataset* tersebut, pasien normal dan pasien kanker direpresentasikan sebagai vektor, seperti pada gambar berikut.



Gambar 2 Ilustrasi SVM

Untuk membangun suatu klasifier SVM, perlu diperhatikan beberapa hal, diantaranya sebagai berikut [7].

1. *Dataset* latih, asumsi *dataset linearly separable*.

$$Dataset = \{(x_i, y_i) | x_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\}_{i=1}^n \quad (8)$$

2. *Margin* atau *gap* merupakan jarak antar dua *hyperplane*

$$w \cdot x + b = 1 \quad (9)$$

$$w \cdot x + b = -1 \quad (10)$$

Diketahui bahwa jarak antar *hyperplane*

$$D = 2/\|w\| \quad (11)$$

Dari persamaan di atas didapatkan bahwa untuk memaksimalkan jarak berarti kita meminimalkan nilai $\|w\|$.

3. Kriteria *hyperplane* yang berusaha meminimalkan nilai $\|w\|$ serta memenuhi persamaan

$$y_i(w \cdot x_i + b) \geq 1, i = 1, \dots, n \quad (12)$$

4. Jika diberikan sebuah data baru x , maka persamaan *classifier*-nya adalah

$$f(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) \quad (13)$$

C. Color Moment

Color moment merupakan ciri yang digunakan sebagai ciri suatu objek berdasarkan warnanya. Metode ini menghitung distribusi warna pada sebuah gambar. *Color moment* dihitung berdasarkan masing-masing *layer* citra, misal *layer red* (R), *green* (G), dan *blue* (B) pada citra RGB. Ada tiga ciri yang dimiliki oleh metode ini, yaitu *mean*, *standard deviation* dan *skewness*. Ketiga ciri tersebut dihitung di masing-masing *layer*, R, G dan B, sehingga menghasilkan 9 ciri. Ciri ini tidak hanya dapat digunakan untuk *color spaces* RGB saja, melainkan *color space* lainnya, seperti HSV, CMYK, LAB, dan lain-lain. Berikut adalah persamaan yang digunakan untuk mencari ciri *color moment* [8].

$$E_i = \sum_{j=1}^N \frac{1}{N} p_{ij} \quad (14)$$

$$\sigma_i = \sqrt{\left(\frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^2 \right)} \quad (15)$$

$$s_i = \sqrt[3]{\left(\frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^3 \right)} \quad (16)$$

Persamaan tersebut menghitung *mean*, *standar deviation*, dan *skewness*. Dari persamaan tersebut, N merupakan jumlah piksel dari gambar, p_{ij} merupakan nilai piksel ke- j pada *layer* ke- i .

D. K-Means Clustering

K-Means Clustering adalah algoritma yang mengelompokkan data menjadi beberapa kelompok (*cluster*) berdasarkan kemiripan antar *datapoint*. Algoritma k-Means dimulai dengan menginisialisasi titik *centroid* sejumlah k titik pada sekumpulan data, baik secara acak ataupun tidak. Setiap *datapoint* yang ada, dilakukan perhitungan kedekatan ciri antara masing-masing *datapoint* dengan ciri titik *centroid* yang dituju. Apabila *datapoint* tersebut memiliki jarak yang kecil terhadap titik *centroid* yang dituju, maka *datapoint* tersebut termasuk ke dalam *cluster* titik *centroid* tersebut. Setelah masing-masing titik *centroid* atau *cluster* memiliki sejumlah data, dilakukan perhitungan kembali titik *centroid* di tiap *cluster* tersebut. Proses pemasukan data pada tiap *cluster* dan proses perhitungan ulang *centroid* ini terus dilakukan sampai nilai titik *centroid* pada tiap *cluster* tidak berubah lagi. Berikut adalah algoritma k-Means [9].

Input : data $D \subseteq R^d$; number of clusters $K \in N$.

Output : K cluster means $\mu_1, \dots, \mu_K \in R^d$.

randomly initialise K vectors $\mu_1, \dots, \mu_K \in R^d$;

repeat

assign each $x \in D$ *to argmin*; $Dis_2(x, \mu_j)$;

for $j = 1$ *to* K **do**

$D_j \leftarrow \{x \in D \mid x \text{ assigned to cluster } j\}$;

$\mu_j = \frac{1}{|D_j|} \sum_{x \in D_j} x$

end

until no change in μ_1, \dots, μ_K ;

return μ_1, \dots, μ_K ;

E. Kalman Filter

Kalman Filter merupakan *estimator* untuk proses diskrit yang linear [4]. *Estimator* ini memprediksi *state* berikutnya berdasarkan *state* saat ini secara rekursif. Hasil *output* dari Kalman merupakan sebuah estimasi yang

selanjutnya akan dikoreksi sesuai hasil observasi di *state* berikutnya. Secara garis besar, Kalman filter dibagi menjadi dua persamaan, yaitu *estimation* dan *measurement*.

1. Estimation Equation

Persamaan ini dilakukan untuk memprediksi *state* berikutnya berdasarkan *state* saat ini. Persamaan ini dibagi menjadi dua, yaitu *state* prediksi dan *covariance* prediksi.

$$\mathbf{x}_k = A\mathbf{x}_{k-1} \quad (17)$$

$$P_k = AP_{k-1}A^T + Q \quad (18)$$

Persamaan 17 ditujukan untuk memprediksi *state* berikutnya berdasarkan *state* sebelumnya yang ditransformasikan menggunakan matriks transisi A [4]. Sedangkan persamaan 18 melakukan prediksi *covariance* dengan menambahkan matriks *process noise* Q [6]. Inisialisasi *state* biasanya diawali dengan nol pada setiap komponen *state vector*. Untuk inisialisasi nilai P dan Q disesuaikan dengan keadaan sistem. Bersamaan dengan jalannya sistem, nilai P dan Q ini akan berubah sesuai dengan proses estimasi dan koreksi. Proses estimasi menghasilkan nilai prediksi *state* \mathbf{x} dan *covariance* P .

2. Measurement Equation

Persamaan ini dilakukan untuk mengobservasi langsung objek. Hasil observasi ini dikalkulasikan dengan hasil estimasi yang sudah diprediksi sebelumnya untuk menghasilkan nilai *state* dan *covariance* baru yang digunakan pada estimasi selanjutnya. Hal ini terus dilakukan secara rekursif. Berikut adalah persamaan *measurement* yang digunakan [6].

$$K_k = P_{k-1}H^T(HP_{k-1}H^T + R)^{-1} \quad (19)$$

$$\mathbf{x}_k = \mathbf{x}_{k-1} + K(\mathbf{z}_k - H\mathbf{x}_{k-1}) \quad (20)$$

$$P_k = (I - KH)P_{k-1} \quad (21)$$

K merupakan *Kalman Gain* yang digunakan sebagai koefisien ketika melakukan koreksi *state* hasil estimasi terhadap hasil observasi. Sama halnya dengan proses estimasi, pada proses koreksi ini juga memiliki matriks *noise* yaitu *measurement noise* R . Observasi dituliskan dalam bentuk vektor kolom \mathbf{z} . Komponen-komponen yang terdapat pada vektor tersebut bergantung pada sistem yang dibuat. Pada video biasanya komponen berisi lokasi objek yang ditandai dengan *bounding box* serta ada yang menambahkan komponen lebar dan tinggi *bounding box* pada vektor \mathbf{z} . Matriks H merupakan matriks transisi yang digunakan untuk mentransformasikan *state* hasil estimasi ke dalam bentuk vektor \mathbf{z} . Hasil dari proses *measurement* ini adalah nilai *state vector* \mathbf{x} dan *covariance* P yang baru, seperti pada persamaan 20 dan 21. Kedua nilai ini yang nantinya digunakan kembali untuk melakukan estimasi selanjutnya [6].

III. PERANCANGAN SISTEM

Sistem yang dibangun pada penelitian ini adalah sistem yang dapat mendeteksi pemain sekaligus membedakan pemain berdasarkan tim masing-masing. Secara umum sistem akan memproses video masukkan yang berisi rekaman pertandingan sepakbola. Sistem melakukan proses deteksi, klasifikasi tim, serta *tracking* pada setiap *frame*-nya. Sistem kemudian menampilkan hasil deteksi pemain yang ditandai dengan *bounding box* dengan label di setiap pemain yang terdeteksi, serta warna *bounding box* yang berbeda. Secara garis besar, skema sistem dibagi menjadi dua tahap, yaitu *training* dan *testing video*.



Gambar 3 Contoh Hasil Deteksi

A. Training

Proses ini untuk melatih sistem agar dapat mendeteksi objek pemain sepakbola. *Training* dilakukan menggunakan sejumlah data citra positif dan data citra negatif. Data citra positif merupakan objek yang akan dikenali, dalam kasus ini yaitu pemain sepakbola. Data ini berupa model citra *player action*, seperti berlari, berjalan, berdiri, dan aksi-aksi lain yang berhubungan dengan aktivitas pemain sepakbola di lapangan. Sedangkan untuk data citra negatif, data berupa citra bukan pemain sepakbola, seperti rumput, garis lapangan, gawang, dan objek-objek lain yang kemungkinan dapat berada di lapangan. Data positif atau data gambar pose pemain yang digunakan dalam penelitian ini berjumlah 771 gambar sedangkan data negatif merupakan gambar suasana lapangan dengan ukuran 128x64 piksel sebanyak 1507 buah.



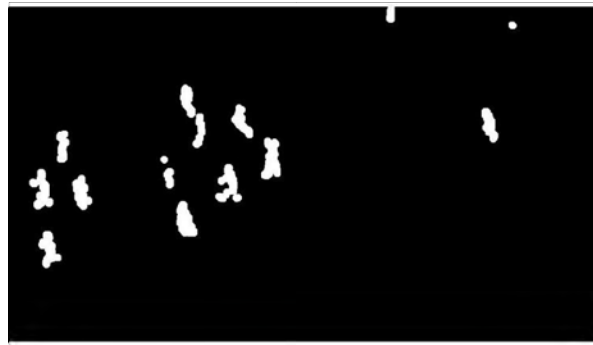
Gambar 4 Contoh Dataset

Proses ekstraksi ciri menggunakan HOG dengan beberapa ukuran *detection window*, dari ukuran citra 32x16, 64x32, dan 128x64 piksel. Ukuran *cell* dibuat menjadi tiga, yaitu 4x4, 6x6 dan 8x8 piksel dengan ukuran blok 2x2 *cell*. Sedangkan untuk histogram, digunakan 9 *bin* dengan rentang 0-180 derajat. Pembangunan HOG dilakukan pada setiap blok dengan *overlap* sebesar 50%. Jumlah dimensi vektor ciri HOG bergantung pada

konfigurasi ukuran cell dan blok HOG. Hasil ekstraksi ciri HOG diklasifikasi menggunakan SVM dengan *linear kernel*. Hasil klasifikasi model disimpan untuk digunakan pada tahap klasifikasi objek dalam proses *testing*.

B. Testing

Testing bertujuan untuk mengetahui performansi dari sistem yang dibangun. Langkah pertama adalah tahap segmentasi lapangan. Hal ini dilakukan dengan asumsi pemain sepakbola selalu berada di area lapangan. Segmentasi lapangan dilakukan dengan cara mengeliminasi piksel yang bukan berwarna hijau (asumsi lapangan berwarna hijau). Segmentasi dilakukan dengan membandingkan piksel pada tiap *channel* RGB [10]. Jika nilai piksel pada G lebih besar dari nilai piksel pada R dan B, maka piksel tersebut dianggap sebagai lapangan dan diberi nilai 1. Jika tidak, maka piksel tersebut bukan lapangan dan diberi nilai 0. Setelah melakukan pengeliminasian piksel, dilakukan proses morfologi untuk menghasilkan daerah atau *region of interest* (ROI) yang akan digunakan untuk proses selanjutnya. Karena video dihasilkan dari kamera yang statis, maka untuk mengurangi waktu komputasi proses pencarian objek, dilakukan deteksi objek bergerak. Tahap deteksi objek bergerak menggunakan algoritma *three frame difference*. Metode ini membandingkan *frame* saat ini dengan *frame* sebelumnya dan *frame* setelahnya. Melalui proses morfologi, piksel-piksel yang bernilai 1 (warna putih) akan dikenali sebagai *foreground* (objek).



Gambar 5 Hasil Deteksi Objek Bergerak

Pada objek-objek bergerak yang terdeteksi, dilakukan ekstraksi ciri menggunakan HOG. Hasil ekstraksi HOG tersebut diklasifikasikan menggunakan SVM yang telah dibangun pada tahap *training*. Setiap objek (pemain) yang terdeteksi di setiap *frame* diberi *bounding box* sebagai ciri bahwa pemain terdeteksi.

Pada area-area *bounding box* hasil deteksi, dilakukan ekstraksi *color moment*. Dari hasil ekstraksi, masing-masing area *bounding box* (*datapoint*) tersebut memiliki ciri warna dengan 9 nilai, masing-masing *channel* R, G dan B memiliki 3 nilai. Semua hasil ekstraksi *color moment* dikelompokkan ke dalam *cluster* menggunakan algoritma *k-means*. Jumlah *cluster* yang ditentukan berjumlah dua. Setelah *cluster* dari masing-masing objek didapatkan, *bounding box* dengan dua jenis warna ditampilkan sebagai ciri bahwa objek pemain terdeteksi dan telah dikelompokkan sesuai *cluster* masing-masing. Warna dari *bounding box* merupakan visualisasi untuk membedakan antar tim.

Proses deteksi objek ini dilakukan pada setiap *frame*. Dari hasil deteksi, diperoleh informasi seperti lokasi serta ukuran *bounding box*. Informasi lokasi digunakan sebagai komponen *tracking* untuk dievaluasi pada setiap *frame* selanjutnya. Untuk meningkatkan performansi deteksi atau mempertahankan proses deteksi objek dilakukan metode Kalman Filter. Metode ini melakukan prediksi atau estimasi *state* berikutnya berdasarkan *state* saat ini dan melakukan koreksi dari hasil estimasi yang didapat dengan hasil observasi pada *state* saat itu. Hasil observasi berupa informasi objek yang telah terekstrak dari HOG dan ciri warna pada *frame* saat itu. Proses estimasi didapatkan dari hasil perhitungan informasi objek (posisi *bounding box*) pada *frame* saat ini untuk menghasilkan informasi objek pada *frame* selanjutnya. Setelah melakukan estimasi, *frame* berubah dari *frame* sebelumnya menjadi *frame* saat ini. Pada *frame* saat ini dilakukan proses koreksi dari hasil estimasi *frame* sebelumnya. Informasi objek hasil estimasi akan dibandingkan dengan hasil observasi (informasi nyata hasil deteksi HOG) pada *frame* saat itu untuk mengurangi tingkat *error* pada saat deteksi antar *frame*.

IV. HASIL PENGUJIAN DAN ANALISIS

Pengujian dilakukan untuk mengukur tingkat performansi sistem. Pada sistem ini, dilakukan dua pengujian, yaitu pengujian deteksi sekaligus *tracking* dan pengujian klasifikasi tim. Pengujian dilakukan pada 7 video dengan berbagai kondisi, seperti video pertandingan sepakbola dengan penerangan alami atau buatan, tingkat perbedaan warna kostum pemain, serta perbedaan tingkat intensitas pencahayaan pada video. Semua video diambil menggunakan kamera statis dari samping lapangan (tribun penonton). Pengujian dilakukan menggunakan 30 *frame* yang diambil dengan interval konstan di tiap video uji. Nilai interval setiap video berbeda, tergantung dari durasi atau total *frame* pada video. Nilai performansi pengujian menggunakan skala dari 0 sampai 1. Pengujian deteksi dilakukan dengan membandingkan lokasi deteksi pemain hasil sistem dengan lokasi pemain nyata di setiap *frame* pada video uji. Pengujian akan dilakukan pada konfigurasi yang berbeda, yaitu ukuran *cell* HOG yang digunakan. Performansi sistem ditentukan dari nilai *F1-score* yang diperoleh dari hasil *precision* dan *recall* [11].

$$recall = \frac{TP}{TP + FN} \quad (22)$$

$$precision = \frac{TP}{TP + FP} \quad (23)$$

$$F1 - score = \frac{2 \times precision \times recall}{precision + recall} \quad (24)$$

Dalam kasus ini, *true positive* (TP) mengindikasikan jumlah lokasi pemain hasil deteksi sistem yang sesuai dengan lokasi pemain asli di video. Sedangkan *false positive* (FP) mengindikasikan jumlah lokasi hasil deteksi yang menunjukkan bukan pemain. Kemudian *false negative* (FN) mengindikasikan jumlah pemain yang tidak terdeteksi oleh sistem.

Pengujian klasifikasi tim dilakukan untuk mengetahui seberapa baik ciri warna *color moment* yang digunakan untuk membedakan antara satu tim dengan tim lain. Performansi dihitung menggunakan persamaan akurasi sebagai berikut.

$$akurasi = \frac{\sum \text{pemain tim } x \text{ yang benar}}{\sum \text{pemain tim } x \text{ yang terdeteksi}} \quad (25)$$

A. Hasil Pengujian Deteksi

Deteksi yang digunakan pada sistem ini dikombinasikan dengan *tracking* Kalman, sehingga kemampuan deteksi akan dibantu oleh *tracking* Kalman. Pada proses ini dilakukan pengujian menggunakan ukuran *cell* HOG yang berbeda. Hasil pengujian dapat dilihat pada tabel berikut.

Tabel 1 Hasil Pengujian Deteksi Pemain Sepakbola

Video	Ukuran Detection Window	Ukuran <i>cell</i> HOG (piksel)		
		4x4	6x6	8x8
A	128x64	0.82/0.83 (0.82)	0.93/0.76 (0.84)	0.92/0.82 (0.87)
B	64x32	0.70/0.80 (0.75)	0.67/0.84 (0.75)	0.76/0.67 (0.71)
C	32x16	0.93/0.75 (0.83)	0.84/0.83 (0.83)	0.80/0.82 (0.81)
D	32x16	0.90/0.76 (0.82)	0.86/0.79 (0.82)	0.80/0.70 (0.75)
E	64x32	0.88/0.87 (0.87)	0.83/0.87 (0.85)	0.84/0.88 (0.86)

F	32x16	0.85/0.80 (0.82)	0.82/0.76 (0.79)	0.73/0.68 (0.75)
G	32x16	0.78/0.61 (0.68)	0.75/0.69 (0.72)	0.68/0.55 (0.61)
Performansi		Precision/Recall (F1-score)		

Dari hasil pengujian dapat dilihat bahwa ukuran *cell* dan ukuran *sliding window* saling berpengaruh. Ketika ukuran *sliding window* 32x16 piksel maka ukuran *cell* yang optimal adalah 4x4 dan 6x6 piksel. Sementara untuk ukuran *sliding window* 64x32 dan 128x64, ukuran *cell* sangat berpengaruh dalam jumlah FP dan FN. Semakin kecil ukuran *cell* maka semakin besar kemunculan FP dan FN.

B. Pengujian Klasifikasi Tim

Pada pengujian ini, dipilih konfigurasi HOG yang memiliki nilai performansi paling tinggi pada saat deteksi. Dengan konfigurasi tersebut, dihitung performansi klasifikasi pada setiap video uji. Wasit, penjaga gawang, dan *bounding box* lain selain pemain tim A dan B tidak akan dihitung. Di bawah ini merupakan tabel hasil pengujian klasifikasi tim.

Tabel 2 Hasil Pengujian Klasifikasi Tim

Video	Kondisi	Akurasi	
		Tim A	Tim B
A	<i>Detection window</i> 128x64 piksel, Perbedaan kostum jauh, <i>background</i> pemain <i>uniform</i>	0.95	0.97
B	<i>Detection window</i> 64x32 piksel, Perbedaan kostum jauh, beberapa <i>background</i> pemain tidak <i>uniform</i>	0.98	0.82
C	<i>Detection window</i> 32x16 piksel, Perbedaan kostum dekat, <i>background</i> pemain <i>uniform</i>	1.00	0.14
D	<i>Detection window</i> 32x16 piksel, perbedaan kostum jauh, <i>background</i> pemain <i>uniform</i>	0.98	0.96
E	<i>Detection window</i> 64x32 piksel, perbedaan kostum sedang serta pencahayaan pada video tidak seragam, beberapa <i>background</i> pemain tidak <i>uniform</i>	0.70	0.41
F	<i>Detection window</i> 32x16 piksel, perbedaan kostum jauh, <i>background</i> pemain <i>uniform</i>	0.86	0.86
G	<i>Detection window</i> 32x16 piksel, perbedaan kostum jauh, beberapa <i>background</i> pemain tidak <i>uniform</i>	0.87	0.98

Berdasarkan tabel hasil pengujian di atas, terlihat bahwa video C dan Video E memiliki tingkat akurasi yang kecil. Pada video C, tingkat akurasi yang kecil disebabkan karena perbedaan warna kostum yang dekat.. Selain itu juga pengambilan gambar yang jauh menyebabkan ukuran pemain yang terdeteksi kecil, hal ini tentunya akan mempengaruhi ukuran gambar *input* saat melakukan ekstraksi ciri warna.

V. KESIMPULAN

Berdasarkan hasil pengujian yang telah diketahui pada bab sebelumnya, maka kesimpulan yang dapat ditarik dari penelitian ini adalah sebagai berikut.

1. Algoritma ekstraksi ciri HOG dapat digunakan pada sistem deteksi dan *tracking* pemain sepakbola pada data video pertandingan sepakbola dengan kondisi perbedaan tingkat intensitas pencahayaan dengan performansi terbaik adalah F1-Score=0.87.
2. Penentuan ukuran *detection window* dan *cell* HOG mempengaruhi performansi sistem. Ukuran *cell* HOG harus proporsional dengan ukuran *detection window*. Jika ukuran *detection window* sebesar 128x64 piksel, maka ukuran *cell* HOG minimal memiliki ukuran 1/8 dari lebar *detection window* untuk lebar dan tinggi *cell* agar histogram yang dibuat proporsional membentuk suatu objek yang dicari.

3. Penggunaan Kalman *filter* dikombinasikan dengan HOG memberikan performansi yang baik dalam melakukan *tracking* pemain. Ketika suatu objek pemain tidak terdeteksi oleh HOG pada *frame* selanjutnya, sistem tetap akan mendeteksi objek tersebut menggunakan hasil *tracking* Kalman.
4. Penggunaan *color moment* sebagai ciri warna pada pengklasifikasian tim sepakbola dapat digunakan pada kasus sepakbola. Dalam penelitian ini, klasifikasi tim menggunakan *color moment* menghasilkan akurasi tertinggi sebesar 0.98 dan 0.96 pada tim A dan tim B pada video D.

DAFTAR PUSTAKA

- [1] Leser, Roland, Arnold Baca, and Georg Ogris. *Local Positioning Systems in (Game) Sports*. Sensors 11.10 (2011): 9778-9797.
- [2] Halvorsen, P., Sægrov, S., Mortensen, A., Kristensen, D. K., Eichhorn, A., Stenhaus, M., ... & Johansen, D. *Bagadus: An Integrated System for Arena Sports Analytics: A Soccer Case Study*. Proceedings of the 4th ACM Multimedia Systems Conference. ACM, 2013.
- [3] Gerke, S., Singh, S., Linnemann, A., & Ndjiki-Nya, P. *Unsupervised Colour Classifier Training for Soccer Player Detection*. Visual Communications and Image Processing (VCIP), 2013. IEEE, 2013.
- [4] Najafzadeh, Nima, Mehran Fotouhi, and Shohreh Kasaei. *Multiple Soccer Players Tracking*. Artificial Intelligence and Signal Processing (AISP), 2015 International Symposium on. IEEE, 2015.
- [5] Dalal, Navneet, and Bill Triggs. *Histograms of Oriented Gradients for Human Detection*. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.
- [6] Li, Changyan, Lijun Guo, and Yichen Hu. *A New Method Combining HOG and Kalman Filter for Video-Based Human Detection and Tracking*. Image and Signal Processing (CISP), 2010 3rd International Congress on. Vol. 1. IEEE, 2010.
- [7] Statnikov Alexander, Hardin Douglas, Guyon Isabelle, Aliferis Constantin F. *A Gentle Introduction to Support Vector Machine in Biomedicine*. Biomedical and Health Informatics: From Foundations to Applications to Policy. AMIA San Francisco November 14-15, 2009.
- [8] Keen Noah. *Color Moments*. School Of Informatics, University of Edinburgh. 2005.
- [9] Flach, Peter. *Machine Learning The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press, New York. 2012
- [10] Grissom Daniel, Tarango Joseph. *Field Detection Using Matlab*. Department of Computer Science and Engineering, University of California, Riverside. 2007.
- [11] Van Asch, Vincent. *Macro-and micro-averaged evaluation measures* [\[\[basic draft\]\]](#). 2013.

