

Perbandingan Metode *CF* dan *k-NN* untuk Identifikasi Warna pada Robot *Soccer*

Syaiful Amri, Hadi Suyono, dan Onny Setyawati

Abstrak—Robot *soccer* adalah salah satu divisi dari Kontes Robot Cerdas Indonesia (KRCI). Tugas robot ini adalah bermain bola berdasarkan *rule* standar internasional yang telah ditetapkan organisasi RoboCup. Sesuai dengan *rule* pertandingan robot *soccer*, bahwa warna setiap objek yang berada dalam lapangan pertandingan memiliki warna tertentu. Objek-objek yang dimaksud adalah bola, gawang, garis-garis putih, dan warna lapangan itu sendiri. Pada penelitian ini objek yang diteliti adalah bola (*orange*) dan gawang (*kuning*). Tingkat akurasi dan waktu yang dibutuhkan untuk mengidentifikasi menjadi pertimbangan dalam mendisain fungsi *vision* robot *soccer*. Penelitian ini digunakan dua metode untuk mengidentifikasi warna objek pada robot *soccer* yaitu *Color Filtering (CF)* dan *k-Nearest Neighbor (k-NN)* untuk mengklasifikasi warna *per-pixel* dalam citra, kemudian dilanjutkan deteksi objek, mencari titik pusat koordinat dan pengukuran jarak objek dengan aksi *scanning* dan *tracking*. Dari hasil penelitian ini didapatkan waktu komputasi dengan *CF* adalah 0,097 detik dan *k-NN* adalah 0,27 detik, sedangkan rerata *error* estimasi jarak bola dengan rentang pengukuran 10 cm sampai dengan 360 cm menggunakan *CF* adalah 6,02% dan *k-NN* 7,19%.

Kata Kunci—*Robot Soccer*, *k-Nearest Neighbor (k-NN)*, *Color Filtering*.

I. PENDAHULUAN

KONTES Robot Sepak Bola Indonesia (KRSBI) merupakan salah satu divisi dari KRCI yang diselenggarakan sejak tahun 2009 dengan nama awal KRCI Expert Battle. KRSBI merupakan kontes robot sepak bola nasional yang sekaligus menjadi ajang kualifikasi nasional untuk mewakili Indonesia dalam RoboCup yang merupakan kompetisi robot sepak bola resmi tingkat dunia di bawah organisasi RoboCup. Kontes robot nasional divisi sepak bola ini (selanjutnya disebut robot *Soccer*) adalah salah satu program kreatifitas mahasiswa yang sangat strategis yang kedepannya didisain untuk dapat menjadi unggulan obyek penelitian dan edukasi rekayasa robotika tingkat dunia dengan standar internasional [1].

Untuk proses identifikasi objek menggunakan sensor kamera telah banyak dilakukan penelitian yang tidak hanya pada bidang robotika saja (robot *Soccer*

misalnya), termasuk di bidang kedokteran, industri, pertanian, geologi, kelautan dan lain sebagainya. Adapun penelitian terdahulu yang berkaitan dengan penelitian ini adalah yang menggunakan *stereo vision* untuk pengukuran jarak objek, namun dengan kondisi intensitas cahaya ruang yang sudah diatur agar tidak mempengaruhi pengukuran jarak objek [2], karena secara matematis, citra objek merupakan fungsi kontinu dengan intensitas cahaya pada bidang dua dimensi. Peneliti lainnya menggunakan *mono vision* modul kamera CmuCam3.0 dengan hasil yaitu kemampuan robot sangat berpengaruh terhadap proses identifikasi objek, dengan menentukan *range* nilai RGB (warna dasar yang dikenal perangkat komputer *Red Green Blue*) suatu objek harus sesuai dengan intensitas cahaya ruangan, karena jika terlalu besar dalam menentukan *range* nilai RGB suatu objek, maka objek yang lain juga dapat dianggap sebagai objek yang sama, namun jika *range* terlalu kecil, maka sangat sulit untuk mendeteksi objek yang dimaksud [3].

Sebelum mengidentifikasi objek berwarna, dilakukan proses segmentasi. Ada beberapa cara untuk segmentasi berdasarkan warna, salah satunya adalah dengan klasifikasi warna menjadi beberapa wilayah. Pada penelitian ini digunakan dua algoritma yaitu algoritma *Color Filtering* (algoritma konvensional untuk klasifikasi warna) dan *k-Nearest Neighbor (k-NN)*, kemudian dibandingkan kinerja antara kedua algoritma tersebut. Algoritma klasifikasi *k-Nearest Neighbor* mempunyai tingkat akurasi yang lebih baik dibandingkan dengan algoritma klasifikasi *Support Vector Machine (SVM)* dalam kasus klasifikasi gambar berwarna [4]. Adapun variabel yang akan digunakan ketiga algoritma tersebut untuk melakukan klasifikasi warna yaitu nilai HSV (*Hue Saturation Value*) yang didapatkan dari hasil konversi nilai RGB citra.

II. WARNA DAN METODE KLASIFIKASI

A. Atribut Warna

Selain model warna RGB, warna juga dapat dimodelkan menjadi sistem ruang HSV, dimana model ini terdiri dari tiga komponen yaitu *Hue (H)*, *Saturation (S)* dan *Value (V)*.

▪ *Hue*

Komponen *Hue* menyatakan warna sebenarnya, seperti merah, biru atau kuning. *Hue* digunakan untuk membedakan warna-warna dan menentukan kemerahan (*redness*), kehijauan (*greenness*), dan sebagainya.

▪ *Saturation*

Syaiful Amri adalah Dosen Teknik Elektro Politeknik Negeri Bengkalis dan Mahasiswa Program Studi Magister Teknik Elektro Universitas Brawijaya, Malang, Indonesia (email: syaifulAmri_ske@yahoo.com).

Hadi Suyono adalah Dosen Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Malang, Indonesia.

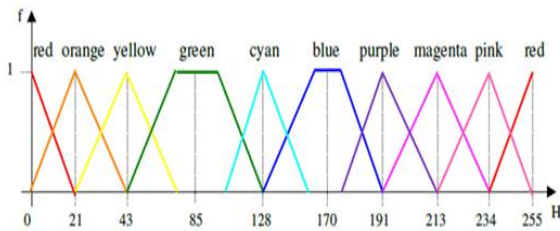
Onny Setyawati adalah Dosen Teknik Elektro Universitas Brawijaya, Malang, Indonesia.

Adalah ukuran seberapa besar tingkat kemurnian warna tersebut, yaitu mengindikasikan seberapa banyak warna putih diberikan pada warna.

▪ *Value/Intensity*

Adalah atribut yang menyatakan ukuran seberapa besar kecerahan dari suatu warna atau banyaknya cahaya yang diterima oleh mata tanpa mempedulikan warna. Nilainya adalah kisaran antara gelap (hitam) dan terang (putih) yang dapat bernilai dari 0 sampai 100 %.

Pada sistem HSV, komponen *hue* sebenarnya menyatakan warna seperti yang biasa dipahami manusia.



Gambar. 1. Partisi dimensi H dengan fungsi keanggotaan fuzzy [6]

Younes dkk membuat model fuzzy untuk menyatakan warna seperti pada Gambar 1 [5]. Warna tanpa *hue* jika suatu warna mempunyai *saturation* = 0, yaitu dibuat dari warna putih saja. Jika *saturation* = 255, maka tidak ada warna putih yang ditambahkan pada warna tersebut.

B. *Klasifikasi Color Filtering*

Algoritma dapat dilakukan dengan melakukan *thresholding* yang digunakan untuk menyaring suatu nilai tertentu dengan menggunakan dua batas, yaitu nilai minimal dan maksimal [6]. Dengan demikian, nilai yang akan digunakan adalah nilai yang berada pada rentang batas minimal dan maksimal yang diizinkan.

Adapun formula untuk *Color Filtering* adalah :

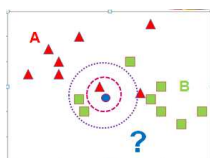
If ($H > H_{min}$) & ($H < H_{max}$) &
 ($S > S_{min}$) & ($S < S_{max}$) &
 ($V > V_{min}$) & ($V < V_{max}$)
Pixel = 255; //putih

Else

Pixel = 0; //hitam

C. *Klasifikasi k-Nearest Neighbor (k-NN)*

Klasifikasi k-Nearest Neighbor (*k-NN*) merupakan sebuah metode untuk melakukan klasifikasi suatu data baru berdasarkan data *learning* yang jarak paling dekat dengan data baru tersebut [7].



Gambar. 2. Contoh klasifikasi dengan *k-NN* [13]

Ilustrasi klasifikasi metode *k-NN* ditunjukkan pada Gambar 3. Data *learning* diproyeksikan ke ruang dimensi banyak, dimana masing-masing dimensi mempresentasikan fitur dari data. Pada klasifikasi *k-NN*

dibagi menjadi bagian-bagian berdasarkan klasifikasi data *learning*.

Untuk mengetahui dekat atau jauhnya tetangga bisa dihitung berdasarkan jarak *euclidean* seperti pada Persamaan 1 [8].

$$d_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \quad (1)$$

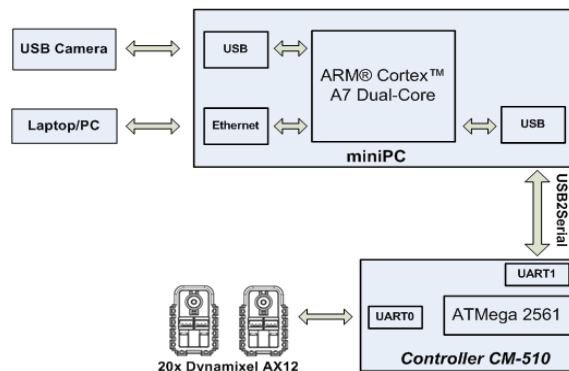
Dimana,

- x_{ik} = Nilai x pada data *training*
- x_{jk} = Nilai x pada data *testing*
- n = Batas jumlah banyaknya data

III. PERANCANGAN DAN IMPLEMENTASI

A. *Perancangan Hardware*

Secara umum ada dua perancangan, yaitu perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*). Adapun perangkat kerasnya meliputi : usb webcam, miniPC (mini personal computer), *controller* CM510, aktuator dynamixel AX12 dan laptop. Interkoneksi masing-masing perangkat keras dari satu perangkat ke perangkat lainnya seperti yang ditunjukkan pada Gambar 3.



Gambar. 3. Interkoneksi *hardware* pada robot soccer

Adapun penjelasan interkoneksi masing-masing perangkat keras adalah sebagai berikut :

- Citra yang dihasilkan dari *webcam* merupakan citra digital yang selanjutnya akan diproses.
- Untuk menghubungkan laptop ke miniPC yaitu dengan menggunakan kabel jaringan atau yang biasa dikenal dengan kabel UTP.
- *Controller* CM-510 yang digunakan pada penelitian ini adalah berbasis ATMega2561, dimana pada *controller* ini terdapat dua jalur komunikasi UART. Pada UART1 digunakan untuk jalur komunikasi *controller* dengan miniPC. Dimana pada *port* USB miniPC ditambah dengan USB2Serial *converter* agar bisa terhubung secara serial dengan UART1 *controller* CM-510.

Hasil identifikasi objek yang diproses dalam miniPC akan ditransfer yaitu berupa data nilai posisi dan jarak objek yang dibutuhkan oleh *controller* CM-510. Sementara pada UART0 digunakan untuk jalur komunikasi *controller* dengan aktuator dynamixel AX12 dengan mode *half duplex*.

Terdapat 2 subsistem seperti yang terlihat pada Gambar 3 dimana tugas masing-masing subsistem

adalah :

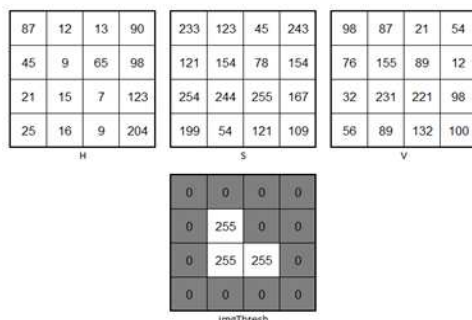
- Tugas mengidentifikasi objek, mengukur jarak objek, mengetahui posisi objek serta mengirimkan data tersebut melalui *port* serial ke *controller* CM-510 yang diproses di miniPC.
- Sedangkan yang kedua yaitu *controller* CM5-510 yang bertugas menerima data dari *port* serial yaitu berupa informasi posisi dan jarak objek yang dikirim dari miniPC. Setelah informasi ini didapatkan, *controller* akan memerintahkan aktuatorX (gerak horizontal) dan aktuatorY (gerak vertikal) untuk bergerak sesuai dengan posisi objek.

B. Perancangan Algoritma

Terdapat 3 algoritma, dimana 2 algoritma yang berfungsi untuk melakukan proses segmentasi citra yaitu dengan mengklasifikasi warna per-*pixel*. Sedangkan algoritma berikutnya adalah jika objek teridentifikasi dari hasil segmentasi, maka proses berikutnya adalah mencari posisi objek yaitu mencari titik pusat massa dengan menggunakan nilai rerata koordinat setiap *pixel* yang menyusun objek dengan aksi *scanning* dan *tracking*.

1. Algoritma Color Filtering

Penerapan algoritma *Color Filtering* pada robot *Soccer* diawali dengan kalibrasi terlebih dahulu. Kalibrasi yang dimaksud digunakan untuk mengetahui nilai batas minimal dan nilai batas maksimal dari komponen warna [12]. Warna yang digunakan adalah model warna HSV (*Hue Saturation Value*) yang dikonversi dari komponen warna R (*Red*) G (*Green*) dan B (*Blue*).

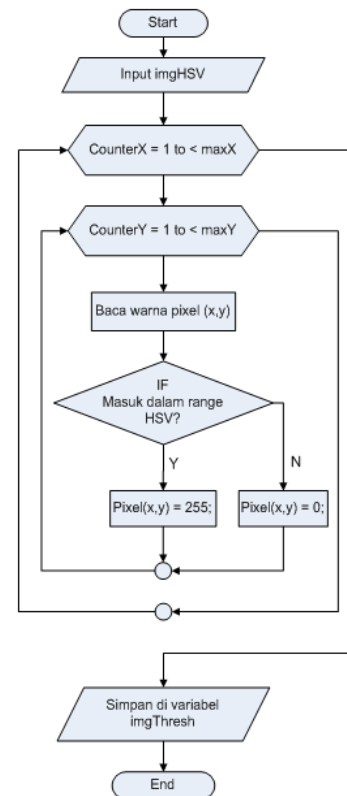


Gambar. 4. Ilustrasi konversi citra HSV ke *grayscale* dengan *Color Filtering*

Tujuan dari kalibrasi ini adalah untuk mencari nilai batas minimal dan maksimal masing-masing komponen warna HSV. Beberapa variabel dan konstanta pada algoritma *Color Filtering* seperti pada Gambar 5. Adapun penjelasan variabel dan konstanta yang terdapat pada algoritma *Color Filtering* Gambar 5 adalah:

- *imgHSV* : Merupakan citra HSV yang sudah dikonversi dari model citra RGB sebagai masukan untuk dikonversi menjadi citra biner.
- *maxX* : jumlah maksimal resolusi citra X (kolom) .
- *maxY* : jumlah maksimal resolusi citra Y (baris).
- nilai 0 : warna hitam
- nilai 255 : warna putih

- *imgThresh* : citra biner hasil konversi yang didapat berdasarkan dari *Color Filtering*.



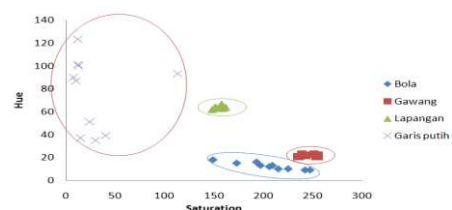
Gambar. 5. Algoritma *Color Filtering*

Pada Gambar 4 diperlihatkan ilustrasi segmentasi dengan metode *Color Filtering* dengan nilai kalibrasi yang didapat adalah:

$H_{min} = 0$, $H_{max} = 17$, $S_{min} = 123$, $S_{max} = 256$, $V_{min} = 105$, $V_{max} = 256$.

2. Algoritma k-Nearest Neighbor (k-NN)

Pada Gambar 6 diilustrasikan 4 kelas yang terbentuk dari data *training* berdasarkan model warna *Hue* (H) dan *Saturation* (S) yaitu kelas bola, gawang, warna lapangan dan garis putih.



Gambar. 6. Kelas dari *training* berdasarkan warna H dan S

Adapun Algoritma *k-Nearest Neighbor* (*k-NN*) adalah:

- Menentukan jumlah kelas yang ingin dibentuk.
- Membangkitkan *centroids* (titik pusat kelas).
- Hitung jarak antara data baru ke setiap label kelas data.
- Tentukan k label kelas data yang mempunyai jarak yang paling minimal.
- Klasifikasikan data baru ke dalam label data yang mayoritas.

Dari data *training* didapatkan *centroids* bola ([205,6], (12,5)), *centroids* gawang ([247,7], [2,17]), *centroids*

lapangan ([155,6], [64]) dan *centroids* garis putih ([205,6], [12,5]).

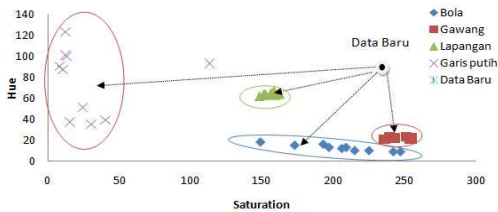
87	12	13	90
45	9	65	98
21	15	7	123
25	16	9	204

233	123	45	243
121	154	78	154
254	244	255	167
199	54	121	109

98	87	21	54
76	155	89	12
32	231	221	98
56	89	132	100

Gambar. 7. Contoh data baru yang ingin di klasifikasikan berdasarkan warna H dan S

Ilustrasi cara menentukan kelas warna objek dari data baru seperti pada Gambar 8. Misal pada *pixel*(1,1), dengan nilai HS(87,233) seperti pada Gambar 7, maka dihitung jarak antara data baru ke masing-masing kelas. Setelah jarak didapatkan, diklasifikasikan data baru tersebut dengan jarak kelas yang paling minimal (tingkat smiliaritas yang tinggi). Ilustrasi data baru yang akan diukur jarak ke masing-masing kelas diperlihatkan pada Gambar 8.



Gambar. 8. Contoh klasifikasi data baru dengan *k-NN*

Untuk menghitung jarak antara data baru dengan masing-masing kelas dengan menggunakan Persamaan 1 yaitu :

Jarak ke kelas bola :

$$a^2 = b^2 + c^2$$

$$a = \sqrt{b^2 + c^2}$$

$$= \sqrt{(233 - 205,6)^2 + (87 - 12,5)^2}$$

$$= 79,38$$

Jarak ke kelas gawang :

$$a^2 = b^2 + c^2$$

$$a = \sqrt{b^2 + c^2}$$

$$= \sqrt{(233 - 247,7)^2 + (87 - 21,7)^2}$$

$$= 66,93$$

Jarak ke kelas lapangan :

$$a^2 = b^2 + c^2$$

$$a = \sqrt{b^2 + c^2}$$

$$= \sqrt{(233 - 155,6)^2 + (87 - 64)^2}$$

$$= 80,75$$

Jarak ke kelas garis putih :

$$a^2 = b^2 + c^2$$

$$a = \sqrt{b^2 + c^2}$$

$$= \sqrt{(233 - 205,6)^2 + (87 - 12,5)^2}$$

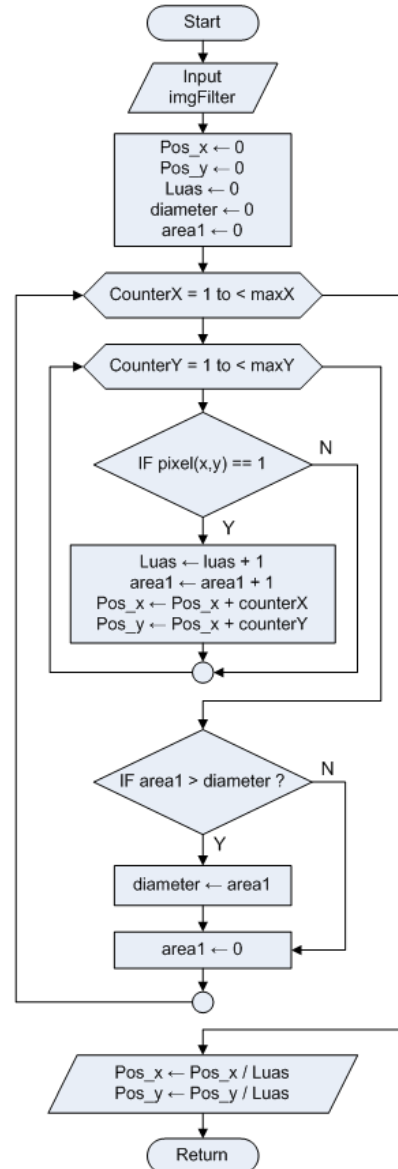
$$= 205,62$$

Setelah didapatkan jarak data baru ke masing-masing kelas, maka diklasifikasikan data baru tersebut dengan jarak kelas yang paling minimal (tingkat smiliaritas yang tinggi) adalah jarak ke kelas gawang yaitu 66,93. Jika objek yang ingin di-track adalah bola, maka hasil

segmentasi dengan *k-NN*, *pixel* tersebut menjadi 0 (warna hitam).

3. Algoritma Mencari Posisi Objek

Setelah kontur di-filter, proses berikutnya adalah mencari posisi (titik koordinat) objek yaitu dengan mencari pusat massa (*centroid*) dengan menggunakan nilai rerata koordinat setiap yang menyusun objek. Pada citra biner yang mana a_{xy} akan bernilai 0 atau 1, *moment* tingkat ke-0 (m_{00}) adalah sama dengan area dari objek.



Gambar. 11. Algoritma mencari posisi koordinat objek

Adapun penjelasan variabel dan konstanta yang terdapat pada algoritma pada *flowchart* Gambar 11 adalah.

- counterX = variabel iterasi X (horizontal/kolom).
 - counterY = variabel iterasi Y (vertikal/baris).
 - maxX = nilai maksimal X (jumlah maksimal kolom).
 - maxY = nilai maksimal Y (jumlah maksimal baris).
 - Pos_x = variabel untuk m_{10} .
 - Pos_y = variabel untuk m_{01} .
 - Luas = variabel untuk luas area objek (m_{00}).
- Pusat dari area atau massa (*moment*) adalah parameter untuk menyatakan lokasi dari objek. Pusat

area dari suatu objek didefinisikan sebagai Persamaan 2 [9].

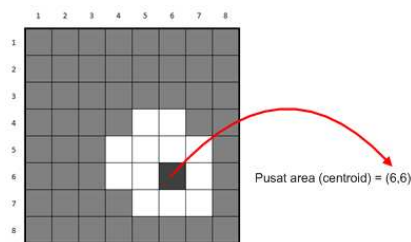
$$x' = \frac{m_{10}}{m_{00}} \text{ dan } y' = \frac{m_{01}}{m_{00}} \quad (2)$$

Dimana (x', y') merupakan pusat koordinat dari objek. Pada Gambar 13 adalah contoh hasil dari threshold yang akan dicari pusat koordinat sehingga akan diketahui lokasi objek pada citra.

$$x' = \frac{73}{13} = 5,6 \rightarrow 6$$

$$y' = \frac{73}{13} = 5,6 \rightarrow 6$$

Sehingga posisi objek adalah pada koordinat *pixel* (6,6) seperti yang ditunjukkan pada Gambar 10.



Gambar 10. Contoh titik pusat objek

IV. HASIL DAN PEMBAHASAN

A. Waktu Komputasi

Pengujian waktu yang dibutuhkan untuk identifikasi adalah dengan cara mengukur durasi waktu yang dimulai dari *capture image*, segmentasi dengan cara mengklasifikasikan per-*pixel* sampai dengan mencari *moment* objek dari hasil *filter* (operasi *opening*). Data hasil dari pengujian didapatkan seperti pada Tabel I.

TABEL I
HASIL PENGUJIAN WAKTU DURASI PROSES IDENTIFIKASI

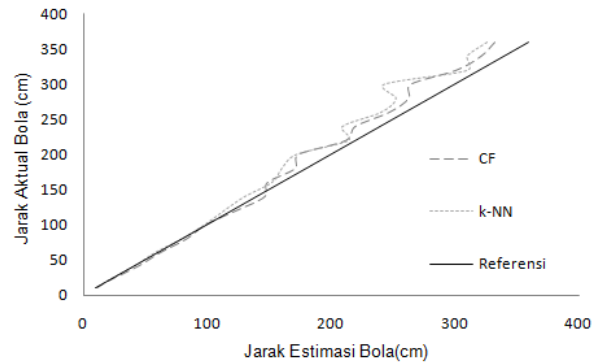
No Uji	Color Filtering	k-NN
1	0,11	0,27
2	0,1	0,25
3	0,1	0,28
4	0,09	0,28
5	0,1	0,29
6	0,09	0,25
7	0,12	0,28
8	0,08	0,27
9	0,09	0,27
10	0,09	0,28
Rerata	0,097	0,27

Tabel I merupakan hasil pengujian durasi waktu identifikasi dalam satu fase. Hasil yang didapatkan bahwa waktu komputasi untuk proses identifikasi warna dengan *CF* lebih kecil (0,097 detik) dibandingkan dengan *k-NN* (0,27 detik), dikarenakan metode *CF* menggunakan logika yang sederhana untuk mengklasifikasi.

B. Error estimasi jarak

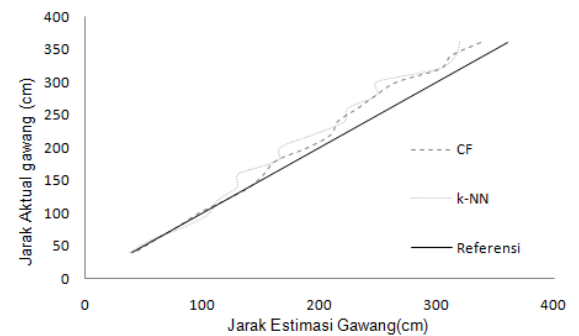
Perbandingan *error* estimasi jarak bola diantara ketiga metode segmentasi dengan klasifikasi warna dari hasil pengujian diperlihatkan pada Gambar 11. sedangkan *error* estimasi jarak gawang diperlihatkan pada Gambar 12.

Dari hasil pengujian, *error* estimasi jarak pada bola maupun gawang didapatkan bahwa metode *k-NN* mempunyai rerata *error* estimasi jarak yang lebih besar dibandingkan dengan metode *CF*.



Gambar 11. Error estimasi jarak bola dengan metode *CF* dan *k-NN*

Ini dikarenakan bahwa pada metode *k-NN* sangat rentan terhadap *noise* pada data *training* yaitu pada objek bola (*orange*), gawang (*kuning*) dan garis putih yang mempunyai nilai *hue* yang saling terdistribusi.



Gambar 12. Error estimasi jarak gawang dengan metode *CF* dan *k-NN*

C. Perbandingan Metode Klasifikasi

Setelah dilakukan pengujian segmentasi dengan metode *Color Filtering* dan *k-NN*, maka terdapat beberapa perbandingan masing-masing metode seperti pada Tabel II.

TABEL II
PERBANDINGAN METODE KLASIFIKASI

No Uji	Rerata Error Estimasi Jarak (%)		Waktu Komputasi (detik)
	Bola	Gawang	
Color Filtering (CF)	6,02	5,76	0,097
k-Nearest Neighbor (k-NN)	7,19	9,78	0,27

Dilihat dari dua parameter yaitu Rerata *error* estimasi jarak dan waktu komputasi, bahwa metode *CF* lebih baik dari *k-NN* pada kasus klasifikasi warna pada robot *soccer*.

V. KESIMPULAN

Metode klasifikasi warna untuk melakukan segmentasi objek (bola dan gawang) pada robot *Soccer* ini dihasilkan rerata *error* estimasi jarak bola yang terkecil dengan rentang pengukuran 10 cm sampai dengan 360 cm adalah pada metode klasifikasi *CF* dengan rerata *error* 6,02%. Sedangkan waktu komputasi tercepat yang dibutuhkan untuk identifikasi warna objek adalah pada metode *Color Filtering* dengan waktu 0,097 detik.

Waktu komputasi yang dibutuhkan untuk identifikasi warna pada metode *k-NN* adalah 0,27 detik dan metode *CF* adalah 0,097 detik, dengan *hardware* yang digunakan adalah miniPC cubieboard2 (Pocessor Arm Cortex dual core, RAM 1Gb).

Agar waktu yang dibutuhkan untuk mengidentifikasi dan *tracking* objek berwarna lebih cepat dengan resolusi *frame* citra 640x480 *pixel* (resolusi *standard*), sebaiknya menggunakan algoritma *Color Filtering* berdasarkan model warna *hue* yang sudah tersedia di *library* *opencv* yang dilengkapi dengan pengenalan bentuk objek sebagai *filter*.

DAFTAR PUSTAKA

- [1] Tim KRCI, 2013. Buku Panduan buku panduan KRCI Robo *Soccer* Humanoid League 2013 versi 1.0, Jakarta.
- [2] Zheng, Chang and Li. 2010. A Study of 3D Feature Tracking and Localization Using A Stereo Vision System, IEEE, pp.402-407.
- [3] Ramadhan . T. 2009. Aplikasi Kontroler Fuzzy *Q-Learning* Pada Robot Mandiri yang Terintegrasi Kamera Cmcumcam 3.0. ITS, Surabaya.
- [4] Farsiah., Fuadi dan Munadi., 2013. Klasifikasi Gambar Berwarna Menggunakan *k-Nearest Neighbor* dan Support Vector Machine, Universitas Syiah Kuala, Banda Aceh.
- [5] Younes, dkk. 2007. *Color Image* Profiling Using Fuzzy Sets. Turkish Journal of Electric Engineering & Computer Sciences. 13(3): 343-359.
- [6] Jatmiko. Wisnu Dkk . 2012. Robotika Teori dan Aplikasi. Perpustakaan Nasional Fakultas Ilmu Komputer UI Depok.
- [7] Rhido. A, 2006. *k-Nearest Neighbor*. Soft Computing Research Group. EEPIS-ITS.
- [8] Khamar, K. 2013. Short Text Classification Using *kNN* Based on Distance Function. IJARCCCE International Journal of Advanced Research in Computer and Communication Engineering. Government Engineering College, Modasa.(ISSN Print : 2319-5940 ISSN Online : 2278-1021).
- [9] Kadir. Abdul., Susanto, Adhi., 2013. Teori dan Aplikasi Pengolahan Citra. Yogyakarta: Penerbit ANDI.
- [10] Basuki, Ahmad. 2006. Metode Bayes . PENS - ITS. Surabaya.
- [11] Putra. Dharma. 2010. Pengolahan Citra Digital. Yogyakarta.
- [12] Rohman. F, 2012. Implementasi Deteksi Citra Berwarna menggunakan *OpenCV* Library Pada Perangkat Keras Embedded System. Tesis Unibraw, Malang.
- [13] McAndrew, Alasdair. 2004. An Introduction to Digital *Image* Processing with MATLAB. Melbourne: Victoria University Of Technology.
- [14] M. N. A. Wahab., N. Sivadev and K. Sundaraj. 2011. Target Distance Estimation Using Monocular Vision System for Mobile Robot. IEEE Conference on Open System (ICOS2011).