



TESIS - TE142599

***TRACKING PEMAIN SEPAKBOLA MENGGUNAKAN METODE
KALMAN FILTER BERBASIS TWO-STAGES HUNGARIAN
ALGORITHM***

ATYANTA NIKA RUMAKSARI
2215206001

DOSEN PEMBIMBING
Dr. Surya Sumpeno, ST., M.Sc.
Dr. Adhi Dharma Wibawa, ST., MT.

PROGRAM MAGISTER
BIDANG KEAHLIAN TELEMATIKA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017



TESIS - TE142599

***TRACKING* PEMAIN SEPAKBOLA MENGGUNAKAN
METODE KALMAN FILTER BERBASIS *TWO-
STAGES HUNGARIAN ALGORITHM***

ATYANTA NIKA RUMAKSARI
2215206001

DOSEN PEMBIMBING
Dr. Surya Sumpeno, ST., M.Sc.
Dr. Adhi Dharma Wibawa, ST., MT.

PROGRAM MAGISTER
BIDANG KEAHLIAN TELEMATIKA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)

di
Institut Teknologi Sepuluh Nopember

oleh:

Atyanta Nika Rumaksari
NRP. 2215206001

Tanggal Ujian : 5 Juni 2017
Periode Wisuda : September 2017

Disetujui oleh:

1. Dr. Surya Sumpeno, ST., M.Sc. (Pembimbing I)
NIP: 196906131997021003
2. Dr. Adhi Dharma Wibawa, ST., MT. (Pembimbing II)
NIP: 197605052008121003
3. Prof. Dr. Ir. Mauidhi Hery Purnomo, M.Eng. (Penguji)
NIP: 195809161986011001
4. Dr. Ir. Wirawan, DEA (Penguji)
NIP: 196311091989031011
5. Dr. Ir. Achmad Affandi, DEA (Penguji)
NIP: 196510141990021001
6. Dr. I Ketut Eddy Purnama, ST., MT (Penguji)
NIP: 196907301995121001

Dekan Fakultas Teknologi Elektro



Dr. Tri Arief Sardjono, S.T., M.T.
NIP: 197002121995121001

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul “***TRACKING PEMAIN SEPAKBOLA MENGGUNAKAN METODE KALMAN FILTER BERBASIS TWO-STAGES HUNGARIAN ALGORITHM***” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 6 Mei 2017



Atyanta Nika Rumaksari

NRP: 2215206001

Halaman ini sengaja dikosongkan

TRACKING PEMAIN SEPAKBOLA MENGGUNAKAN METODE KALMAN FILTER BERBASIS TWO-STAGES HUNGARIAN ALGORITHM

Nama mahasiswa : Atyanta Nika Rumaksari
NRP : 2215206001
Pembimbing : 1. Dr. Surya Sumpeno, ST., M.Sc.
2. Dr. Adhi Dharma Wibawa, ST., MT.

ABSTRAK

Tracking atau pelacakan pemain dalam video sepakbola menjadi bagian penting dalam aplikasi cerdas yang berbasis interaksi manusia dan computer. Dimana dalam proses ini obyek diberikan label sesuai dengan identitasnya atau peran dalam pertandingan. Proses ini bertujuan untuk membantu mencari obyek (pemain atau bola) secara efisien. Kemudian hasil lokasi pergerakan obyek tersebut disimpan dalam media, agar dapat digunakan untuk merekonstruksi pola tersembunyi dari pergerakan obyek. Tantangan utama dalam penelitian *tracking* ini adalah ketika obyek yang diambil memiliki ukuran kecil, jumlahnya banyak dan memiliki pergerakan yang random, terdapat bayangan oleh karena sistem pencahayaan dan posisi pemain sering berhimpitan (oklusi) sehingga menyulitkan detektor mendeteksi pemain. Peneliti telah berhasil menghadapi tantangan tersebut dengan membuat sistem komprehensif gabungan dua proses, yaitu proses deteksi obyek dan proses *tracking*. Pada proses deteksi obyek peneliti menggunakan metode *background subtraction* dengan menambahkan filter operasi *bitwise* sebagai perekonstruksi obyek dan penghilang bayangan. Selanjutnya, pada proses *tracking*, metode Kalman filter diaplikasikan dengan menambahkan metode penugasan *dual-Hungarian* pada proses estimasi obyek dan pemulihan garis trayektori akibat dari oklusi.

Kata kunci: Ilmu olahraga, subtraksi *background*, *multi-tracking*, visi komputer

Halaman ini sengaja dikosongkan

TRACKING PLAYERS USING KALMAN FILTER METHOD BASED ON TWO-STAGES HUNGARIAN ALGORITHM

By : Atyanta Nika Rumaksari
Student Identity Number : 2215206001
Supervisor(s) : 1. Dr. Surya Sumpeno, ST., M.Sc.
2. Dr. Adhi Dharma Wibawa, ST., MT.

ABSTRACT

Tracking of players becomes an important part of intelligent applications based on human and computer interactions on soccer video. In this process the object is labeled according to its identity or role in the match. This process aims to efficiently locate the players automatically. It is also being used for storing the location for semantic approach. Thus, it can use to reconstruct the hidden patterns of object movement. The main challenge in this tracking are the object taken has a small size, number is large and has a random movement, there are shadows because of lighting system and position of players often coincide (occlusion). Thus, it is difficult for detectors to detect the players. We have successfully faced these challenges by creating a comprehensive system of combined two processes, namely object detection and tracking process. In object detection process, we used background subtraction method by adding bitwise operation filter as object reconstruction and shadow removal. Furthermore, in the tracking process, the Kalman filter method was applied using dual-Hungarian assignment method on the object estimation process and the result of occluded trajectory line.

Key words: *Sports science, background subtraction, multi-tracking, computer vision*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kepada Tuhan Yesus Kristus karena telah menyertai dan memampukan saya menyelesaikan tesis dengan baik. Saya ucapkan terimakasih kepada Bapak Dr. Surya Sumpeno selaku pembimbing pertama, dan Bapak Dr. Adhi Dharma Wibawa selaku pembimbing kedua. Tidak lupa saya ucapkan terima kasih kepada teman-teman di lab HCCV Bp. Adri Gabriel, Bp. Khamid, Bp. Adi Falih, Ib. Masfulatul, Bp. Didin, Bp. Mamat. Teman-teman Telematika 2015 yaitu Fahima, Jananta, Dhanu, dan Pak Candra. Teman-teman CIO 2015 mbak Erna, mbak Intan, mbak Indah, mbak Lia, mbak Erlyna, mbak Asri, mbak Nafik dan mohon maaf kepada teman-teman yang lain yang tidak saya sebutkan.

Terima kasih untuk LPDP yang berkenan memberikan beasiswa kepada saya. Kepada jajaran direksi LPDP Bp. Eko Prasetyo, Bp. Syahrul Elly, Bp. Mokhamad Mahdum, Bp. Abdul Kahar, dan Ibu Ratna Prabandari, saya ucapkan terima kasih banyak. Serta teman-teman *awardee* LPDP PK28, dan *awardee* lain di ITS, Om Oddy, Faisal, Nia, Aviv, dan banyak lagi yang tidak dapat saya sebut satu-per satu.

Terima kasih untuk Universitas Kristen Satya Wacana Salatiga, Bp. Dr. Iwan Setyawan, Bp. Saptadi, Bp. Andreas, dan rekan-rekan lainnya karena memberikan kesempatan kepada saya sebagai staf pengajar di Fakultas Teknik Elektro dan Komputer.

Terakhir, sporter utama saya, terima kasih atas doa dan kasihnya, Istri saya Putri Hergianasari, Anak saya Aleiteo Kinaratama, Ibu dan bapak saya Sri Sulandjari dan Mardi Yuwono, mertua Heroe Poerwanto dan Sugiati, keluarga kakak ipar Mas Panji, Mbak Selvi dan Nirma.

Semoga tesis yang saya tulis ini dapat berguna dan menjadi berkat bagi yang membacanya.

Surabaya, 19 Juni 2017

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

LEMBAR PENGESAHAN	iii
PERNYATAAN KEASLIAN TESIS	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan	4
1.4 Batasan Masalah	4
1.5 Kontribusi	5
BAB 2 KAJIAN PUSTAKA	7
2.1 Kajian Penelitian terkait	7
2.1.1 Penelitian Menggunakan Visi Komputer	7
2.1.2 Deteksi <i>Obyek dengan Background Subtraction</i>	8
2.1.3 <i>Tracking</i> Obyek	9
2.1.4 Aplikasi Kalman Filter sebagai Metode <i>Tracking</i>	10
2.2 Dasar Teori	12
2.2.1 Deteksi Obyek dengan <i>Background Subtraction</i>	12
2.2.2 Teori Optimasi Linier	14
2.2.3 Morfologi	18
2.2.4 Komponen Warna	21
2.2.5 Kalman Filter	23
2.2.6 Penugasan dalam Multi Point <i>Tracking</i>	26
BAB 3 METODOLOGI PENELITIAN	29
3.1 Preprocessing	29
3.1.1 Input Video	30
3.1.2 Ekualisasi Hitogram	32

3.1.3	Perancangan Metode.....	32
3.2	Proses Deteksi Obyek.....	35
3.2.1	<i>Background Subtraction</i>	35
3.2.2	Filter dan Morfologi Citra	41
3.2.3	Proses Penentuan Lokasi	43
3.3	Proses Tracking	44
3.3.1	Kalman Filter untuk <i>Multi-Tracking</i>	47
3.3.2	<i>First-Stage</i> Hungarian	48
3.3.3	Normalisasi Trayektori	49
3.3.4	<i>Second-Stage</i> Hungarian.....	50
BAB 4	HASIL DAN PEMBAHASAN	53
4.1	Dataset Uji Coba	53
4.2	Preprosesing	58
4.3	Proses Deteksi Obyek.....	59
4.3.1	Pemilihan <i>Background Sintesis</i>	59
4.3.2	Optimasi <i>Threshold</i> Menggunakan Algoritma GDLS.....	65
4.3.3	Perbandingan dengan Metode Deteksi Obyek Lain	66
4.4	Proses <i>Tracking</i>	72
4.4.1	Optimasi Trayektori	74
4.4.2	Perbandingan dengan Metode <i>Tracking</i> Lain.....	80
BAB 5	KESIMPULAN	83
	DAFTAR PUSTAKA.....	87
	LAMPIRAN 1 Matrix Biaya Antar Frame t dan $t - 1$	93
	LAMPIRAN 2 Contoh Hasil Tracking Sebagai Analisa Statistik.....	94
	LAMPIRAN 3 Posisi Penelitian Terhadap Penelitian Lain	95
	BIOGRAFI PENULIS	97

DAFTAR GAMBAR

Gambar 2-1. Gambar grafik contoh fungsi konveks $f(x) = x^2$	15
Gambar 2-2. Area diatas kurva fungsi $f(x)$ disebut sebagai kelompok konveks	16
Gambar 2-3. Grafik perubahan obyek tiap waktu	17
Gambar 2-4. Ilustrasi step tunggal	18
Gambar 2-5. Visualisasi proses morfologi: Dilasi	19
Gambar 2-6. [44] Input (a) dilanjutkan proses opening (<i>pembukaan</i>) (b) dan closing (<i>penutupan</i>) (c)	20
Gambar 2-7. Representasi visual perhitungan kontur biner	21
Gambar 2-8. Perbandingan format warna: RGB(a), HSV(b), dan YcbCr(c) [33].	22
Gambar 2-9. Diagram alir algoritma Hungarian [32]	28
Gambar 3-1. Diagram alur perancangan sistem	29
Gambar 3-2. Layout Kamera pada Stadion Alfheim [7]	31
Gambar 3-3. Proses subtraksi background	36
Gambar 3-4.. Inisialisasi background	37
Gambar 3-5. Background model pada $N=1000$	38
Gambar 3-6. (a) adalah contoh hasil subtraksi citra pada komponen Hue, (b) hasil subtraksi pada komponen Saturation, dan (c) hasil subtraksi pada komponen Value.	38
Gambar 3-7. Gerbang logika filter bitwise spasial.....	42
Gambar 3-8. Representasi pusat <i>massa</i> sebagai perwakilan keberadaan pemain. .	44
Gambar 3-9. Ilustrasi pembuatan jarak obyek referensi (merah) dengan estimasinya (biru, hijau).	48
Gambar 4-1. Diagram proses perancangan sistem.	53
Gambar 4-2. Pada frame 525 menunjukkan pemain memiliki bayangan	54
Gambar 4-3. Diambil dari frame pertama yang merupakan contoh: (a) Degradasi warna lapangan oleh karena distribusi pencahayaan tidak merata dan (b) warna seragam pemain memiliki kemiripan dengan <i>background</i>	56
Gambar 4-4. Frame ke-25 (a) <i>ground truth</i> (b) Gambar input.....	56
Gambar 4-5. (a) Hasil sebelum dan (b) Hasil sesudah proses ekualisasi histogram	58
Gambar 4-6. Histogram Komponen Merah Background Sintetis	62
Gambar 4-7. Histogram Komponen Hijau Background Sintetis.....	62
Gambar 4-8. Histogram Komponen Biru Background Sintetis	62
Gambar 4-9. Grafik nilai F1 lawan jumlah iterasi	65
Gambar 4-10. Gambar hasil deteksi metode usulan dibandingkan dengan metode-metode pembanding. (1) Gambar input, (2) Ground Truth, Gambar (3) GMM, (5) PCASM memiliki karakteristik rentan terhadap pengenalan bayangan. (4) DTSR dan (6) POBS tahan terhadap bayangan akan tetapi hasilnya terdapat noise. (7) adalah metode usulan.	67
Gambar 4-11. Gambar hasil filter bitwise spasial yang tidak sempurna.....	68
Gambar 4-12. Grafik Area Dalam Kurva (AUC) antar metode pembanding.	70

Gambar 4-13. Hasil Tracking menggunakan Kalman Filter pada 1900 frame.....	72
Gambar 4-14. Hasil proses optimasi Trayektori pada 1900 frame	75
Gambar 4-15. Garis tak beraturan pada awal tracking	76
Gambar 4-16. Gambar proses optimasi perbaikan eror hasil tracking	80
Gambar 4-17. Hasil Tracking dengan metode partikel filter dengan representasi sparse	81

DAFTAR TABEL

Tabel 3-1. Tabel karakteristik kamera yang digunakan untuk input	30
Tabel 3-2.. Parameter yang digunakan untuk meng-encode H264	30
Tabel 3-3. Tabel tantangan yang dihadapi pada dataset Alfheim [34].....	33
Tabel 3-4. Tabel tantangan yang dihadapi pada dataset Alfheim [34] (lanjutan) ..	34
Tabel 3-5. Tabel kebenaran filter bitwise spasial.....	42
Tabel 4-1. Tabel penjelasan variabel confusion matriks	57
Tabel 4-2. Hasil pemilihan background	60
Tabel 4-3. Perbandingan Warna RGB dan HSV dalam lapangan sepakbola.....	64
Tabel 4-4. Tabel perbandingan metode usulan dan metode lain	69
Tabel 4-5. Tabel TPR dan FPR untuk grafik AUC	70
Tabel 4-6. Tabel akurasi dan presisi antar metode	71
Tabel 4-7. Tabel jumlah obyek ideal yang harus tertampil	72
Tabel 4-8. Tabel rata-rata hasil tracking obyek yang terdeteksi	73
Tabel 4-9. Tabel kesalahan identifikasi pemain	77
Tabel 4-10. Jarak Euclidean antara obyek frame t-1 dan t.....	79

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam analisa video sepakbola, mengetahui lokasi pemain saat bertanding adalah hal yang utama. Para analis pertandingan saat ini menggunakan cara manual dalam melakukan pelabelan pemain, yaitu dengan memastikan identitas pemain dengan melihat seragam dan nomor punggung pemain. Ditunjang dengan perkembangan teknologi yang mutakhir saat ini, rekaman pertandingan sepakbola dapat dimungkinkan untuk memiliki pelabelan otomatis yang dibuat *on spot* di tengah pertandingan berlangsung. Tim perekam menggunakan peralatan kamera dan *cpu* yang canggih sehingga detail tekstur pemain, lapangan dan bola dapat terlihat dengan jelas walaupun direkam dari jarak yang jauh. Pelatih dan analis pertandingan menggunakan data rekaman ini sebagai referensi untuk membuat keputusan yang ada kaitannya dengan pembuatan strategi. Pelatih dalam olahraga sepakbola professional, tidak hanya bertindak sebagai pendamping teknis pemain akan tetapi bertanggung jawab juga terhadap kondisi fisik, psikis, infrastruktur dan keorganisasian pemain dalam tim. Adakalanya, pelatih bertindak sebagai penyambung lidah kebijakan manajemen. Oleh karena kompleksitas proses pengambilan keputusan dalam pertandingan, maka pengerjaan analisa statistik menjadi rumit dan memerlukan waktu lama dalam pengerjaannya. Analisa statistik didefinisikan sebagai sebuah hasil analisa dari data peubah random pergerakan pemain saat bertanding yang memiliki pola tertentu. Tantangannya adalah bagaimana pola tertentu tersebut dapat diketahui secara tepat. Pola tersebut didefinisikan sebagai karakteristik tim saat bertanding. Pemain memainkan perannya berdasarkan instruksi strategi dari pelatih. Peran pemain diciptakan pelatih sesuai dengan strategi yang digunakan, dengan kata lain dapat dijelaskan bahwa karakteristik tim relatif terhadap strategi. Untuk dapat melihat pergerakan pemain dibutuhkan kejelian pengamatan. Oleh karena kemampuan pengamatan manusia terbatas dalam hal pekerjaan *multitasking*, misalnya pada saat yang bersamaan melakukan aktivitas menghitung berapa frekuensi pemain melakukan

pelanggaran, melakukan offside, mendapatkan kesempatan merebut bola dan mengoperkan kepada penyerang, banyaknya kesempatan melakukan penyelamatan yang seharusnya dapat dilakukan akan tetapi tidak dilakukan dan lain sebagainya. Maka demi membantu mempercepat analisis pertandingan melacak pergerakan pemain sepakbola supaya dapat meningkatkan akurasi analisa pertandingan, otomatisasi pelacakan pemain dibutuhkan.

Pelacakan dengan menggunakan teknologi visi computer dipilih karena memperhatikan faktor kenyamanan pemain saat bertanding. Dengan kata lain saat mengumpulkan data, pemain tidak direpotkan dengan sensor atau alat apapun yang menempel pada tubuhnya. Para peneliti melakukan pelacakan karena mereka ingin mengamati dan mengenali pola perilaku obyek [1]. Dalam melakukan pelacakan video sepakbola, video memiliki tantangan yang harus dihadapi agar pelacakan dapat akurat. Pertama, karakteristik banyak obyek yang harus di kenali sehingga pergerakan random dengan kecepatan yang selalu berubah-ubah. Kedua, para pemain memiliki kecenderungan untuk berhimpitan (oklusi) sehingga *region of interest* (ROI) mengecil atau obyek menjadi *intermittent*. Ketiga, obyek yang dilacak memiliki luas yang kecil (*pixel-wise*) sehingga bagian tubuh obyek ada yang hilang sehingga pembentukan pusat massa sebagai titik lokasi tidak merepresentasikan secara presisi tentang lokasi obyek. Keempat, tiap-tiap obyek memiliki bayangan yang ikut bergerak sehingga dikenali sebagai bagian tubuh pemain.

Penelitian [2] [3] [4] hanya fokus pada hasil setelah *tracking*, mereka belum mengaplikasikan metode pemulihan lokasi obyek oleh karena himpitan. Padahal pemulihan lokasi obyek adalah hal yang sangat penting karena pada aplikasi riil obyek dapat tidak lengkap atau bayangan diidentifikasi sebagai obyek yang membuat metode *tracking* menjadi tidak maksimal. Deteksi obyek memiliki beberapa dasar pembentukan, secara matematis deteksi obyek dibuat dari memodelkan *background* dan tau *foreground*. Pemodelan ini bertujuan untuk meramalkan *pixel* mana yang akan menjadi *foreground-background*. Di dalam algoritma pemodelan *foreground-background* metode subtraksi atau selisih dari frame masukan dengan *background* untuk memperoleh *foreground* (obyek) dilakukan. Selanjutnya dijelaskan berdasarkan dasar pemodelan matematis dari

penelitian, deteksi obyek [5] dibagi menjadi empat besaran pokok yaitu menggunakan metode statistik untuk memisahkan *background* dan *foreground*, pencarian vector yang merepresentasikan *background* dan *foreground* tersebut, pengenalan *foreground* (obyek) dengan template matching, dan mengaplikasikan pengenalan fitur *foreground* berdasarkan keunikan hasil transformasi ranah frekuensi. Empat besaran pokok penelitian deteksi obyek, semuanya menggunakan asumsi format warna standard RGB, walaupun demikian, kita mengetahui bahwa setiap representasi warna dari format warna memiliki keunikan sendiri-sendiri, hal ini ditunjukkan dari penelitian Perez [6].

Berdasarkan analisa yang didapatkan dari hasil penelitian pendahulu yang disebut diatas, dalam kaitannya dengan penelitian guna menyelesaikan tantangan, penulis untuk membuat sistem multi-*tracking* pemain sepakbola, peneliti membagi solusi menjadi dua sistem utama, yaitu sistem deteksi obyek dan sistem *tracking*. Hal ini ditujukan untuk mendapatkan hasil yang maksimal dengan memvalidasi berdasarkan pembagian tersebut. Dalam sistem deteksi obyek, berbeda dengan penelitian terdahulu yang lebih mengacu pada kompleksitasnya [2] [3] [4], metode usulan dibuat didasarkan pada kesederhanaan aplikasi perhitungan ini. Kesederhanaan yang dimaksud adalah dalam pembuatan masking dan sekaligus memfilter bayangan, metode usulan tidak menggunakan proses operasi perulangan akan tetapi operasi aritmatika biner antara *pixel* referensi dengan *pixel* filter sehingga beban komputasi oleh karena perulangan dapat dihilangkan pada proses ini. Pada sisi pelacakan (*tracking*) metode usulan dibuat menggunakan metode Kalman filter yang dimodifikasi untuk *tracking* multi-obyek yang ditambahkan dua metode penugasan Hungarian agar label pemain dapat dipulihkan kembali berdasarkan peran yang dimainkan. Dari metode usulan ini peneliti berhasil menyelesaikan tantangan-tantangan seperti pengenalan obyek banyak (*multi-object tracking*) yang bergerak secara random dan termasuk obyek berukuran kecil karena direkam dengan menggunakan kamera bersudut pandang jauh (*panoramic*), pemulihan himpitan oleh oklusi, dan bayangan pemain oleh karena pencahayaan.

1.2 Rumusan Masalah

Belum tersedianya informasi tentang data statistik pergerakan pemain yang dihasilkan oleh proses *tracking* secara otomatis yang didapat dari video pertandingan sepakbola dalam rangka data statistic tersebut digunakan sebagai bahan pelatih/analisis untuk mendapatkan informasi dari perhitungan statistic. Hasil tersebut digunakan sebagai referensi pelatih untuk menghasilkan strategi pertandingan.

1.3 Tujuan

Berdasarkan inisialisasi penelitian awal dapat dikemukakan bahwa, dalam penelitian ini terdapat tiga tujuan utama yang dijelaskan sebagai berikut ini.

- Menyediakan informasi tentang data statistic yaitu koordinat pergerakan pemain dalam video sepakbola dengan menggunakan Kalman Filter sebagai metode *trackingnya*.
- Menghilangkan bayangan yang dihasilkan oleh proses deteksi pemain
- Memulihkan garis trayektori para pemain akibat terjadinya oklusi

1.4 Batasan Masalah

Penelitian ini berhubungan dengan penelitian lanjutan, yaitu melakukan aplikasi nyata pada sebuah sistem uji, sehingga dalam proses penelitian ini citra uji dan sistem proses haruslah sedekat mungkin dengan proses aplikasi lanjutan. Pertama, karakteristik sistem *robust* dimana dataset haruslah direkam dengan menggunakan kamera pengawas (*surveillance*) stasioner yang terpasang dipinggir lapangan. Kedua, selama pertandingan berlangsung tidak ada intervensi pengarahan oleh operator karena keseluruhan perekaman dilakukan secara otomatis. Berkenaan dengan batasan tersebut maka penulis memilih menggunakan dataset uji resmi [6]. Dataset ini dikeluarkan oleh Pal Halvorsen pada tahun 2013, dimana saat itu ia aktif bekerja pada Departemen informatika Universitas Oslo Norwegia.

1.5 Kontribusi

Kontribusi penelitian dibagi menjadi dua, yaitu dalam bidang keilmuan dan dalam bidang aplikasi kemasyarakatan. Kontribusi peneliti secara keilmuan yaitu memperkenalkan dua metode baru, yaitu: pembentukan masking sekaligus menyaring bayangan obyek oleh karena pencahayaan dan pemulihan garis trayektori proses *tracking* dengan menggunakan metode Hungarian ganda. Selain secara keilmuan, penelitian ini juga memiliki kontribusi dalam bidang ilmu olahraga untuk digunakan sebagai alat bantu mempermudah pekerjaan analis dan pelatih sepakbola dalam menganalisa video pertandingan sepakbola.

Halaman ini sengaja dikosongkan

BAB 2

KAJIAN PUSTAKA

Dalam Bab 2 ini dijelaskan mengenai kajian yang dilakukan oleh peneliti dalam membuat sistem *tracking* pemain sepakbola pada video pertandingan sepakbola. Dalam kajian ini peneliti membagi menjadi dua, yaitu saat peneliti melakukan kajian penelitian terkait dari peneliti lain dan kajian mengenai dasar teori yang digunakan untuk menyusun sistem tersebut.

2.1 Kajian Penelitian terkait

2.1.1 Penelitian Menggunakan Visi Komputer

Para peneliti pendahulu telah banyak melakukan penelitian tentang aplikasi visi komputer pada bidang olahraga menggunakan bola, maka dari itulah para peneliti membentuk penelitian secara resmi terpublikasi di Springer tentang aplikasi visi komputer dalam bidang olahraga yang diprakarsai oleh organisasi *Advances in Computer Vision and Pattern Recognition* [7]. Secara garis besar penelitian olahraga menggunakan bola dibagi menjadi dua hal, yaitu: melacak dan mengenali. Melacak berarti menunjukkan label pada waktu t dari label pada waktu $t - \alpha$. Mengenali berarti melakukan klasifikasi sesuatu berdasarkan hasil pembelajaran fitur dataset. Para peneliti telah melakukan penelitian dengan menggunakan komputer visi pada olahraga. Kajian tentang aplikasi computer visi dalam olahraga dibagi menjadi empat bidang penelitian yaitu:

- Pelacakan terhadap pergerakan bola [8] [9] [10],
- Pelacakan terhadap pergerakan pemain [11] [12],
- Pengenalan kegiatan yang dimainkan oleh pemain dengan mengenali kegiatan yang sedang mereka mainkan hal ini berarti melakukan klasifikasi otomatis tipe olahraga yang didapat dari video olahraga sembarang [13] dan mengenali nama aksi yang sedang berlangsung [14].
- Klasifikasi pola permainan yang mereka mainkan [4]. Hasil analisa pola permainan ini sangat berguna bagi pelatih dalam mempertimbangkan efektifitas strategi yang digunakan.

Dalam penelitian kali ini, penulis meneliti tentang *tracking*, dimana dalam aktivitas tersebut sistem harus dapat melacak label pemain dari waktu t yang sama dengan label pada waktu $t - \alpha$. Penulis mengkaji beberapa kajian mengenai pelacakan obyek yang diawali dengan deteksi obyek dan digabung dengan metode pelacakan pemain dan diakhiri dengan penggunaan Kalman filter sebagai metode *tracking*. Berikut adalah penjelasannya.

2.1.2 Deteksi Obyek dengan *Background Subtraction*

Deteksi obyek dengan *background subtraction* berarti melakukan pelabelan obyek (*singular* atau *plural*) sebagai *True* apabila pixel obyek ditemukan sebagai *foreground* dan label *False* bila pixel obyek ditemukan sebagai *background*. Metode *background subtraction* menggunakan perhitungan statistik sangat umum digunakan pada citra *background* yang tidak mengalami perubahan alias *statis*. Memodelkan *background* statis ini dengan menggunakan metode komponen statistik seperti *running average* [17] [18] [19], median [20] [21], atau analisa histogram [22] [23]. Ada pula metode statistik lain yang digunakan sebagai solusi terhadap bentuk *background* yang dinamis misalnya Model Gabungan Gaussian [24] [25] [18]. *Background* dinamis diartikan sebagai pixel bukan obyek yang diinginkan untuk dideteksi, dan pixel tersebut dapat berubah secara dinamis. Hal ini merupakan tantangan yang berat dalam metode deteksi. Diperlukan metode deteksi yang dapat mengklasifikasi obyek sebagai *foreground* dan non-obyek sebagai *background*. Selain menggunakan metode berdasar statistik, terdapat metode berdasarkan kluster [26] [27] [28]. Model ini mengupayakan tiap-tiap nilai *pixel* dalam frame untuk direpresentasikan berdasarkan klasternya (kelompok). Dewasa ini ditemukan beberapa metode yang menggunakan Neural Network, PCA, dan transformasi kuantisasi vector untuk menyelesaikan permasalahan pemodelan obyek dengan mengeliminasi *background*. Penelitian yang menggunakan Neural Network sebagai deteksi obyek dijelaskan dari ketika membuat representasi *background* berdasarkan bobot dari *Node* metode Neural Network (NN) [29] [30]. Metode lainnya seperti RPCA yang menggabungkan PCA dengan solusi representasi sparse [31], Estimasi

menggunakan filter *multiclass* statistical method Ekstraksi *foreground* berdasar area dengan menggunakan LVQ [32].

2.1.3 Tracking Obyek

Tracking [8] meneliti tentang pembentukan garis trayektori atau lintasan bola dari video pertandingan tenis. Pelacakan bola pada pertandingan tenis ditujukan untuk memberikan informasi yang berupa catatan kaki (anotasi) pada video, dengan cara merekonstruksi garis trayektori dari pergerakan bola tersebut [8]. Metode ini menggunakan algoritma asosiasi informasi data berlapis dimulai dari pembuatan lokasi kandidat bola dari frame sebelumnya. Kemudian dilanjutkan dengan *sliding window* melakukan search pada tiap-tiap *pixel*. Selanjutnya, melakukan perhitungan rekursi terhadap kandidat yang telah diketahui sampai mencapai kondisi konvergen (selisih nilai kandidat frame sebelum dan frame setelah memiliki nilai minimum). Terakhir, melacak posisi bola dengan persamaan kecepatan dengan kecepatan konstan. Lokasi bola yang ditemukan diproyeksikan kedalam ranah 3D dengan sumbu x adalah kolom, sumbu y adalah baris frame dan sumbu z adalah waktu (frame). *Tracking* obyek disini adalah *tracking* obyek tunggal. Deteksi obyek menggunakan teknik rekursif dan memiliki perulangan pada tiap-tiap *pixel*, akibatnya apabila video berukuran besar (Kualitas HD) maka beban komputasinya akan besar dibandingkan dengan metode deteksi obyek non rekursif. Asosiasi informasi diperoleh berdasarkan suatu obyek adalah obyek dianalisa dalam bentuk garis trayektori yang menghubungkan antara obyek estimasi pada frame t dan obyek yang dianalisa pada frame selanjutnya $t + \alpha$. Untuk memvalidasi hasil trayektori dilakukan dengan algoritma *Dijkstra*.

Terdapat suatu penelitian dengan menggunakan metode rekonstruksi trayektori lain dengan menggunakan pemodelan 3D. Rekonstruksi 3D pada trayektori bola tenis meja dengan menggunakan teknik aproksimasi bidang yang dilakukan oleh [9]. Hal ini menggunakan model aproksimasi validitas bidang planar. Pembuatan metode ini, diawali dengan pemodelan pergerakan dengan menggunakan persamaan pergerakan obyek (persamaan Newton) dengan mengasumsikan percepatan konstan dan diakhiri dengan memodelkan bidang lintasan dengan menghitung jarak *least square* terhadap lintasan trayektori bola.

Untuk menghasilkan posisi lokasi 3D, obyek pada penelitian ini direkam dengan menggunakan kamera stereo sehingga fitur jarak kedalaman bola sebagai pembentuk 3D dapat diketahui.

Rekonstruksi trayektori gerakan bola dengan menggunakan 6 buah kamera *long shoot* HD tersinkronisasi merupakan teknologi yang sedang berkembang pada akhir-akhir ini [10]. Multi metode diaplikasikan dari deteksi obyek sampai pelacakan obyek. Segmentasi obyek dengan menggunakan subtraksi *background*. Proses penghilangan *background* digunakan metode *sliding windows* seperti pada penelitian [8]. Langkah-langkah prosesnya dijelaskan sebagai berikut, yaitu dalam satu pengambilan akan dihasilkan 4 buah frame yang berkorespondensi dengan 6 kamera yang terpasang, dimana dua frame dari satu kamera utara dan satu kamera selatan, dan dua frame dari dua kamera timur dan dua kamera barat. Selanjutnya proses deteksi bola secara khusus menggunakan metode Hough transform. Dalam pemilihan metode *tracking*-nya menggunakan metode *Supervisor Node*. Metode ini berusaha melakukan koneksi keberadaan bola antar frame dan menyimpan lokasi keberadaan bola tersebut sebagai garis trayektori. Karena perekaman obyek dilakukan menggunakan multi kamera yang terpasang pada empat posisi mata angin, maka ketepatan posisi bola dapat diketahui dengan baik. Selanjutnya dijelaskan sebuah metode *tracking* dengan menggunakan Kalman Filter yang digunakan oleh penulis sebagai sistem pelacak label otomatis.

2.1.4 Aplikasi Kalman Filter sebagai Metode *Tracking*

Kalman Filter merupakan metode *tracking* yang awalnya ditujukan untuk melacak sebuah obyek. Metode ini mengaplikasikan persamaan hukum kedua Newton tentang persamaan percepatan dan kecepatan. Beberapa metode pada subbab 2.1.3 telah dijelaskan mengenai asumsi bahwa percepatan dan kecepatan awal suatu obyek adalah nol. Hal ini dimungkinkan karena pada saat awal *tracking* obyek berada pada kondisi diam di frame ke- t . Dengan analisa *pre-processing* ini digunakan sebagai suatu bentuk awal pembelajaran sistem (patokan) *tracking* agar dapat secara otomatis melakukan deteksi dan prediksi tentang lokasi obyek berada pada frame selanjutnya. Sehingga secara intuitif dapat dikatakan bahwa Kalman Filter akan melakukan pembelajaran pada frame-frame awal dan tingkat presisi

metode ini ditentukan oleh seberapa presisi deteksi obyek tersebut berada. Oleh sebab itu deteksi obyek mempunyai peran yang sangat penting dalam aplikasi *tracking*. Berdasarkan persamaan state matriks Kalman, metode ini termasuk metode *tracking* linier. Oleh karena itu dalam proses *tracking*nya, pembentukan garis trayektori hasil *tracking*nya dapat digambar dengan menggunakan teknik estimasi segitiga *pytagoras*. Selanjutnya, dalam aplikasi penelitian ini, peneliti menggunakan metode Kalman filter untuk melacak multi obyek. Peneliti, menggunakan satu metode kalman dengan mengasumsikan input matriks sebagai persamaan koordinat para pemain sepakbola. Alasan menggunakan satu metode Kalman dibanding dengan multi-Kalman filter adalah, dengan satu metode Kalman untuk matriks inputan dimensi n pemain sepakbola dapat dimungkinkan untuk diaplikasikan penugasa satu pemain sebagai satu peran, dan tidak ada prediksi lebih dari satu penugasan pada keluaran proses prediksi. Berdasarkan hal tersebut maka, metode Hungarian sebagai penugas peran dipekerjakan. Berikut adalah penelitian-penelitian *tracking* dengan menggunakan metode Kalman.

Penelitian yang dilakukan oleh [15] tentang aplikasi Kalman Filter dan algoritma Hungarian untuk multi-*tracking*. Data yang digunakan pada penelitian ini adalah data buatan, yaitu ketika obyek dilihat dari kamera *top view*. Proses deteksi oklusi menggunakan karakteristik obyek pada transformasi blob dengan menambahkan estimasi titik pusat masa benda. Fokus penelitian ini adalah untuk menguji performansi dari metode Kalman Filter dengan algoritma Hungarian. Hasil yang didapatkan dari penelitian ini adalah rata-rata sistem memerlukan 26 KB memori untuk menghitung dalam tiap frame.

Kasus oklusi yang terjadi telah diteliti lebih lanjut oleh [16] berdsarkan modifikasi Kalman Filter dengan algoritma pendeteksi oklusi. Proses pengenalan obyek paska oklusi dengan menggunakan asumsi bahwa data diasosiasikan dengan informasi obyek frame sebelum dan frame sesudah. Strategi yang dilakukan untuk pengenalan obyek pembentuk informasi asosiasi adalah dengan menghitung jarak *euclidean* antara frame referensi dengan frame sebelum dan jarak *euclidean* antara frame referensi dengan frame sesudahnya. Apabila hasil perhitungan antara frame acuan masa kini dan frame referensi masa lalu kurang dari nilai acuan maka obyek acuan frame pada masa kini dikenali sebagai obyek referensi frame masa lalu.

Strategi kedua yang dilakukan ialah menganalisa luas area obyek, apabila luas area obyek sesuai dengan nilai *threshold* maka obyek tersebut dikenali sebagai oklusi.

2.2 Dasar Teori

Dasar teori yang digunakan untuk mengerjakan penelitian ini dibagi menjadi enam subbab yaitu diawali dengan menjelaskan mengenai deteksi obyek dengan menggunakan teknik *background Subtraction*. Kemudian penulis menjelaskan mengenai teori optimasi linier yang nantinya digunakan untuk membentuk *threshold* dalam proses binerisasi untuk *masking*. Ketiga, penulis menjelaskan mengenai teknik morfologi untuk merekonstruksi obyek. Pemahaman komponen warna merupakan dasar yang penting karena dengan memahami komponen warna, penulis dapat menentukan analisa *threshold* yang sesuai untuk digunakan sebagai filter pemisah antara bayangan dan obyek. Kemudian dilanjutkan dengan, dasar teori pembentukan modifikasi Kalman Filter untuk *tracking* multi obyek. Dalam penjelasannya, penulis akan menuliskan langkah-demi-langkah untuk merancang pemodelan matriks state dengan menggunakan hukum kedua Newton. Terakhir, yang tidak kalah pentingnya tentang teori penugasan multi obyek. Metode ini digunakan oleh peneliti berdasarkan hasil penelitian bahwa, terdapat kemungkinan suatu obyek melakukan oklusi atau himpitan dengan metode lain, dan oleh karena proses deteksi obyek dan atau keadaan oklusi obyek dapat terjadi *intermitten*, sehingga mengganggu kesalahan pelabelan. Secara visual kesalahan pelabelan ini dapat terlihat dari garis trayektori yang putus. Berdasarkan kondisi ini maka metode Hungarian diaplikasikan dua kali sebagai solusi permasalahan tersebut.

2.2.1 Deteksi Obyek dengan *Background Subtraction*

Deteksi obyek menggunakan *masking* dalam computer visi adalah suatu metode untuk menunjukkan obyek dengan cara memberi label benar (*True*) pada *pixel* obyek citra dan label salah (*False*) pada *pixel* bukan obyek. Dimana obyek didefinisikan sebagai kumpulan *pixel* yang membentuk suatu daerah yang diminati yaitu ROI (*region of interest*) untuk dipilih, dimana dalam kasus ini daerah yang

diminati sama dengan para pemain yang sedang bermain. Pembentukan proses ini dengan cara mengenali karakteristik obyek dan non obyek. Suatu obyek dalam video sepakbola juga dapat didefinisikan sebagai nilai *pixel* yang bergerak selama waktu bermain dan sepanjang area frame. Dalam permainan bola ideal, dari definisi yang telah dikonstruksi maka dapat disimpulkan bahwa, nilai *pixel* yang bergerak adalah para pemain, bola, wasit, dan hakim garis. Secara matematis dapat dimodelkan suatu posisi obyek $O_i(x, y)$, dimana O_i adalah $\{x_{oi}, y_{oi}\}$ dengan l adalah jumlah piksel yang merupakan anggota dari O_i dan $O_i(x, y) = centroid\{O_i\}$. Hal ini berarti centroid merupakan titik berat luas area O_i . Dengan kata lain apabila detektor mendeteksi O_i dengan tidak lengkap yaitu $l' < l$ dan atau $l' > l$ maka O_i tetap dianggap obyek dengan label i . Dalam penelitian kali ini penulis tidak mendefinisikan obyek secara spesifik, memisahkan jenis obyeknya akan tetapi merujuk pada keseluruhan $O_i(x, y)$ pada $i < i < l$ yang bergerak pada frame I . Seperti yang kita telah ketahui diatas bahwa arti melacak adalah memberikan label yang berkesinambungan antara frame sebelumnya dan frame saat ini. Sehingga dapat dituliskan sebagai $\mathcal{L}\{O_{it-1}\}$, yaitu label obyek- i pada waktu ke- t adalah sama dengan label obyek i pada waktu $t - 1$. Dari kumpulan label pada waktu akhir pertandingan usai T inilah mesin dapat memproses garis trayektori yang merupakan gambar yang merepresentasikan dimana obyek sedang bergerak.

Sementara itu deteksi *background* didefinisikan sebagai $\phi = I - O_i$ dimana I adalah frame dan O_i adalah kumpulan *pixel* obyek. *Background* memiliki karakteristik nilai *pixel* $P(x, y)$ yang statis dalam setiap waktu. Hal ini diakibatkan oleh karena kamera merekam keseluruhan gambar dengan statis maka $\phi_{t=1} = \phi_{t=2} = \phi_{t=3}$, dst. Oleh karena $O_i(x, y)$ selalu bergerak maka $l_{oi} \ll l_{\phi}$, sehingga hasil detektor merupakan kumpulan nilai *pixel* pada lokasi (x, y) yang mengisyaratkan posisi obyek. Dalam penelitian kali ini nilai *pixel* dikodekan menjadi tiga komponen format warna HSV. Alasan pemilihan komponen warna HSV dapat dijelaskan pada subbab 2.2.4. Hasil akhir dari *background* adalah variabel ϕ dimana variabel ϕ perlu dilakukan pemeliharaan agar $|I - O_i|$ dapat optimal.

Setelah selesai menghitung *background*, langkah selanjutnya adalah menghilangkan komponen *background* pada frame input. Proses menghilangkan

komponen *background* dengan cara mengurangkan citra input dengan hasil *background* tersebut, hasil pengurangan dilakukan pada tiap-tiap komponen warna. \mathbb{R}^3 . Sehingga proses tersebut mendapatkan nilai $f(\Gamma) = \|I_t - \phi_t\|$ dengan $\Gamma = [\gamma_1, \gamma_2, \gamma_3]$. Fungsi f merupakan *image difference*. Oleh karena nilai *pixel* tidak mungkin negative maka perlu hasil pengurangan tersebut diberi harga mutlak. Dan akhirnya dari proses inilah *masking* dibuat. Proses *masking* dalam visi computer adalah proses pemisahan antara *foreground* dan *background* seperti yang dijelaskan pada sesi sebelumnya, pembuatan mask disini sama dengan membinerkan citra f dengan suatu threshold T . Nilai T memiliki elemen-elemen sebanyak tiga buah sesuai dengan komponen warna. Pemilihan nilai T yang mengakibatkan optimalnya $G(.)$ akan dijelaskan pada sub bab 2.2.2 dibawah ini.

2.2.2 Teori Optimasi Linier

2.2.2.1 Definisi Fungsi Konveks

Dalam matematika, nilai riil dari sebuah fungsi dalam interval dinamakan konveks, bila segmen garis antara dua titik dari grafik fungsi berada diatas pada fungsi. Dalam bidang euclidean (atau disebut secara umum dalam bidang vektor) paling sedikit dua dimensi, disebut sebagai konveks apabila titik-titik *epigraf*¹-nya merupakan kumpulan nilai riil yang konveks. Secara visual dapat dilihat pada gambar dibawah ini, bahwa fungsi $f(x) = x^2$ merupakan fungsi konveks. Karena memenuhi persyaratan konveks. Definisi bahwa X adalah kumpulan konveks dalam bidang vektor riil dan $f: X \rightarrow \mathbb{R}$ adalah sebuah fungsi.

- f dapat dikatakan konveks bila memenuhi persamaan sebagai berikut:

$$\forall x_1, x_2 \in X, \forall t \in [0, 1]: \quad (2-1)$$

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2).$$

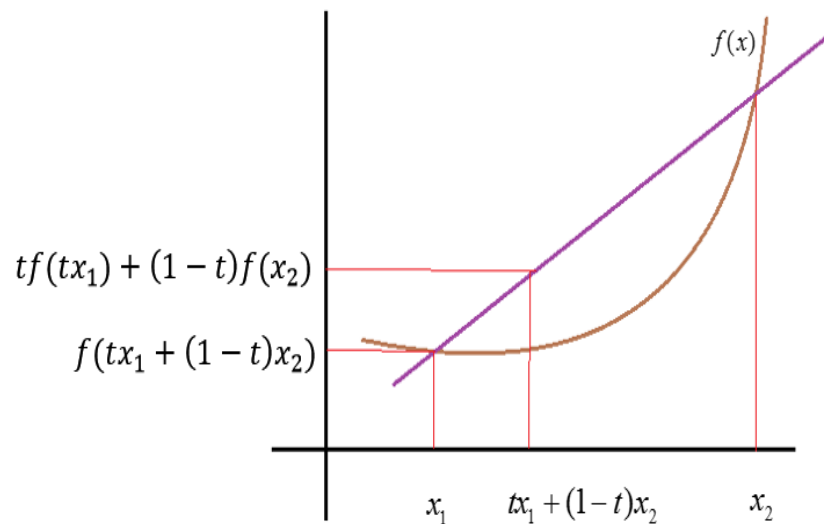
- f dapat dikatakan konveks ketat bila memenuhi persamaan sebagai berikut:

¹ Kumpulan dari nilainya berada diatas atau tepat berada nilai kurva fungsi tersebut

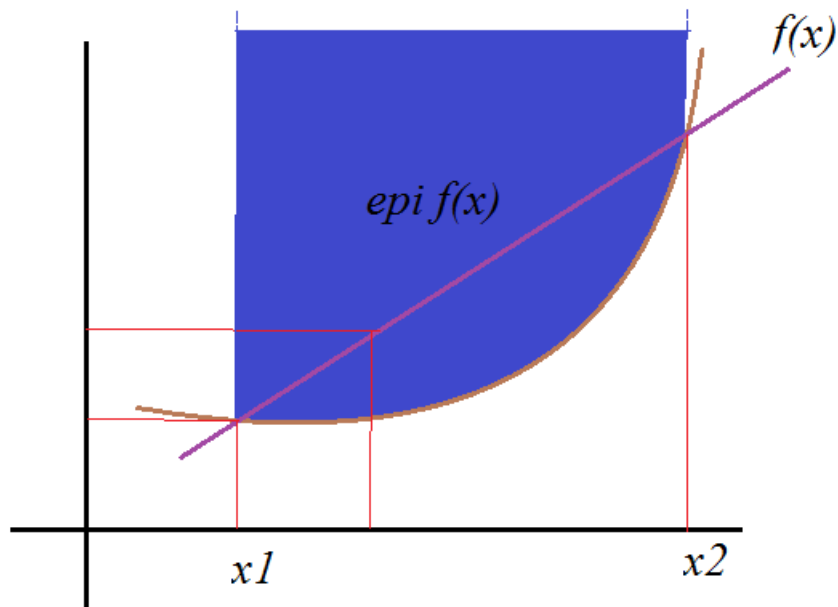
$$\forall x_1 \neq x_2 \in X, \forall t \in (0,1): \quad (2-2)$$

$$f(tx_1 + (1-t)x_2) < tf(x_1) + (1-t)f(x_2).$$

- Bila f adalah sebuah fungsi konveks maka jika dan hanya jika daerah diatas fungsi tersebut adalah kelompok konveks, yang memenuhi persamaan (2-1). Dari Gambar 2-2 dapat diketahui bahwa sebuah fungsi yang menghasilkan nilai selalu positif akan menghasilkan perubahan yang selalu menaik (meningkat), hal ini dapat dilihat pada garis ungu pada Gambar 2-2 dibawah ini. Dalam teori kalkulus, pencarian lokal minimum, global minimum adalah dengan menggunakan metode derivative. Dan oleh karena nilai $f(x) = x^2$ definisi diatas menyatakan bahwa local minimum dan global minimum adalah sama karena menempati lokasi dasar lembah yang sama.



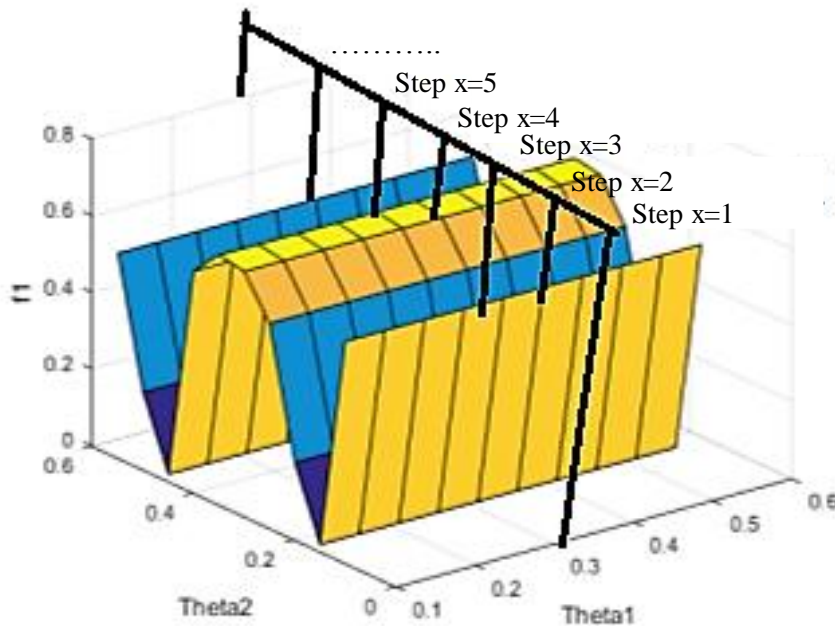
Gambar 2-1. Gambar grafik contoh fungsi konveks $f(x) = x^2$



Gambar 2-2. Area diatas kurva fungsi $f(x)$ disebut sebagai kelompok konveks

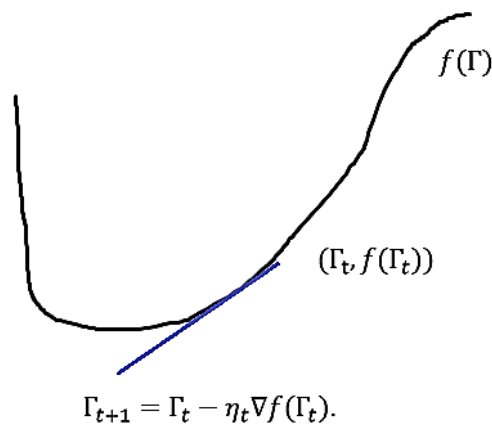
2.2.2.2 Optimasi Konveks Dimensi Bebas: Gradient Descend Line Search

Dalam optimasi, strategi *line search* adalah satu dari dua metode pencarian *local minimum* iterative (perulangan). Diketahui bila *local minimum* dinotasikan sebagai x^* dari sebuah fungsi obyektif $f: \mathbb{R}^n \rightarrow \mathbb{R}$. Metode *line search* berfungsi menemukan asal-usul sepanjang fungsi obyektif f , dimana fungsi f ini akan dikurangi dan kemudian dihitung langkah-langkahnya dan ukurannya yang menjelaskan seberapa jauh X dapat bergerak sesuai dengan arah langkah pencarian f nya. Gambar 2-3 menjelaskan bagaimana pencarian nilai f terhadap sumbu x dan y dengan ditampilkan step perubahan x pada sumbu x (theta 1).



Gambar 2-3. Grafik perubahan obyek tiap waktu

Suatu fungsi optimasi $f(\Gamma)$ didefinisikan sebagai pengurangan dua buah peubah $|I_t - \widehat{\phi}_t|$. Apabila $\Gamma = [\gamma_1 \gamma_2 \gamma_3]$ adalah nilai threshold warna H, S, dan V serta hasil dari fungsi f adalah seperti tertampil pada Gambar 2-1, maka berdasarkan karakteristik fungsi konveks yang menyatakan nilai dari kelompok f berada diatas kurva konveks dan bernilai riil, dapat disimpulkan bahwa kurva f diatas berjenis kurva konveks, dan metode optimasi konveks dapat diaplikasikan. Metode Gradient Descent sebagai solusi optimasi fungsi konveks dilakukan dengan $\underset{\Gamma}{\text{Minimalkan}} f(\Gamma)$ caranya: mengiterasi $\Gamma_{t+1} = \Gamma_t - \eta_t \nabla f(\Gamma_t)$. Dimana η_t adalah *stepsize*. Gambar 2-3 menjelaskan secara sederhana bagaimana step x=1, step x=2 dsb bergerak selaras dengan arah kurva f . Pergerakan step tersebut diatur oleh fungsi f_{t+1} , Hal ini dapat terlihat dengan jelas dalam proses step tunggal pada Gambar 2-4, bahwa f_{t+1} adalah garis biru yang mengikuti kurva $f(\Gamma)$ sepanjang Γ . Pada akhirnya akan didapatkan $\eta_t = \arg \min_{\eta} f(\Gamma - \eta \nabla f(\Gamma))$.



Gambar 2-4. Ilustrasi step tunggal

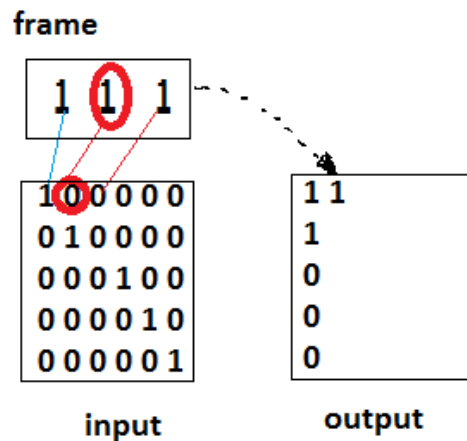
Setelah mendapatkan citra biner proses selanjutnya adalah memperbaiki bagian yang rusak akibat proses deteksi dengan cara menyambung daerah yang tak terhubung. Proses tersebut dinamakan morfologi citra.

2.2.3 Morfologi

Morfologi adalah sebuah cabang dari operasi pemrosesan citra yang berfungsi memproses bentuk dari sebuah obyek. Operasi morfologi menstrukturisasi elemen dari input citra dan menghasilkan output citra dimana ukuran matriks dari output citra tersebut memiliki ukuran yang sama dengan inputnya.

2.2.3.1 Dilasi dan Erosi

Operasi pemrosesan morfologi citra dibagi menjadi dua yaitu Dilasi dan Erosi, dimana dilasi adalah sebuah operasi dimana nilai dari output pixel keluaran adalah nilai max dari semua nilai pixel di frame inputnya. Bila dalam citra biner dilasi akan memaksa *pixel* dalam satu frame berharga satu walaupun pixel masukan tersebut bernilai nol. Proses ini dapat dilihat pada Gambar 2-5.



Gambar 2-5. Visualisasi proses morfologi: Dilasi

Secara matematis definisi dilasi dari sebuah citra I oleh struktur elemen H adalah :

$I \oplus H = \{(p + q) | p \in I, q \in H\}$. Dengan kata lain : Ambil dan kopi keseluruhan struktur elemen, H_p , pusatkan pada setiap lokasi pizel p di muka adalah:

$$I \oplus H = \bigcup_{p \in I} H_p \quad (2-3)$$

Sedangkan Erosi (kebalikan dari dilasi) adalah operasi nilai pixel frame output dipaksa bernilai minimum. Dalam gambar biner, jika salah satu *pixel* diatur ke 0, pixel keluaran diatur ke 0.

Pembentukan kontur atau biasa disebut sebagai penebalan tepi obyek adalah aplikasi dari proses morfologi. Erosi dari sebuah citra I oleh struktur elemen H adalah:

$$I \ominus H = \{p \in \mathbb{Z}^2 | (p + q) \in I, \text{ pada setiap } q \in H\} \quad (2-4)$$

Persamaan tersebut berarti: Simpan hanya pixel $p \in I$ dimana H_p cocok didalam I : $I \ominus H = \{p | H_p \subseteq I\}$.

2.2.3.2 Segmentasi Biner

Dalam proses segmentasi, penggunaan dualisme algoritma dilasi dan erosi diberlakukan. Erosi dapat dihitung sebagai dilasi dalam *background*:

$$I \ominus H = \overline{(I \oplus H^*)} \quad (2-5)$$

Sehingga dualisme pada dilasi adalah:

$$I \oplus H = \overline{(I \ominus H^*)} \quad (2-6)$$

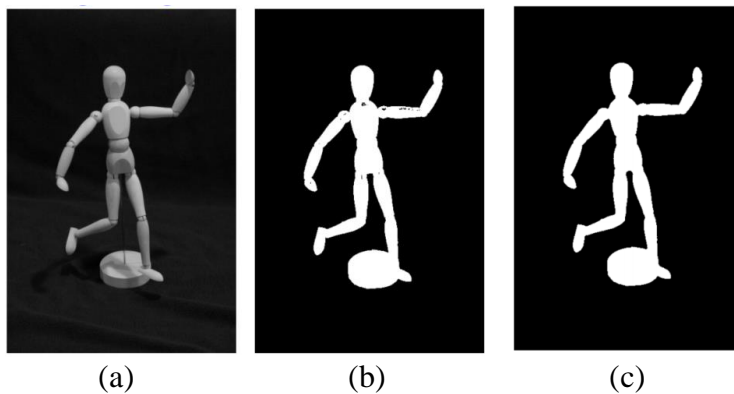
Operasi segmentasi sampai perhitungan kontur dimulai dari pembukaan (Gambar 2.4. (b)) yaitu teknik erosi kemudian diikuti dengan dilasi yang berguna untuk menyasar *foreground* yang lebih kecil dari struktur elemen H akan hilang, dan struktur yang lebih besar tetap tinggal.

$$I \circ H = (I \ominus H) \oplus H \quad (2-7)$$

Dilanjutkan dengan penutupan (Gambar 2-6. (c)) untuk melakukan pengisian total pada obyek, operasi dilakukan dari dilasi yang diikuti oleh erosi:

$$I \cdot H = (I \oplus H) \ominus H \quad (2-8)$$

Lubang yang berada di *foreground* yang lebih kecil dari H akan ditimpa, sehingga lubangnya hilang tertutupi dengan logika 1.



Gambar 2-6. [44] Input (a) dilanjutkan proses opening (*pembukaan*) (b) dan closing (*penutupan*) (c)

2.2.3.3 Kontur Biner

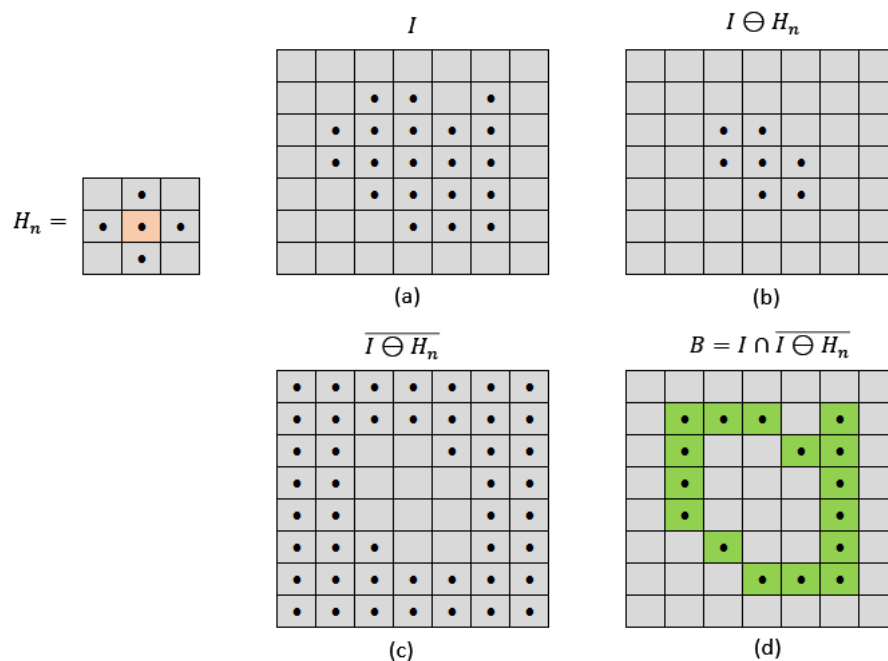
Dalam kasus morfologi *binary contour* sering disebut outline, artinya menggaris. Outline citra $B(u,v)$ dari obyek binary dapat dihitung dengan menggunakan dilasi yang diikuti oleh substraksi (atau operasi XOR). Gambar 2-7 menunjukkan representasi visual perhitungan kontur biner, diawali dengan H_n sebagai matriks $Se = \text{Structural element}$. Gambar 2-7 (a) adalah citra input, (b)

adalah proses XOR per pixel citra input dengan S_e . (c) adalah not adari (b), dan hasil akhir B adalah irisan I (AND) dengan hasil (c). Operasi kontur biner ini berfungsi untuk pembuatan bentuk (*shape*) dari sebuah obyek, koordinat bentuk ini kemudian disimpan dalam suatu *buffer* yang nantinya digunakan untuk melakukan *training* supaya pengenalan atas dasar bentuk unik dari suatu obyek dapat dilaksanakan. Penulis, menggunakan aplikasi kontur biner ini untuk memfilter bagian yang bukan merupakan tubuh pemain dan sekaligus menghilangkan bayangan. Proses ini terletak pada proses masking dengan menggunakan operasi aritmatika biner.

$$I' = I \ominus H$$

$$B(u, v) = XOR(I'(u, v), I(u, v))$$

(2-9)

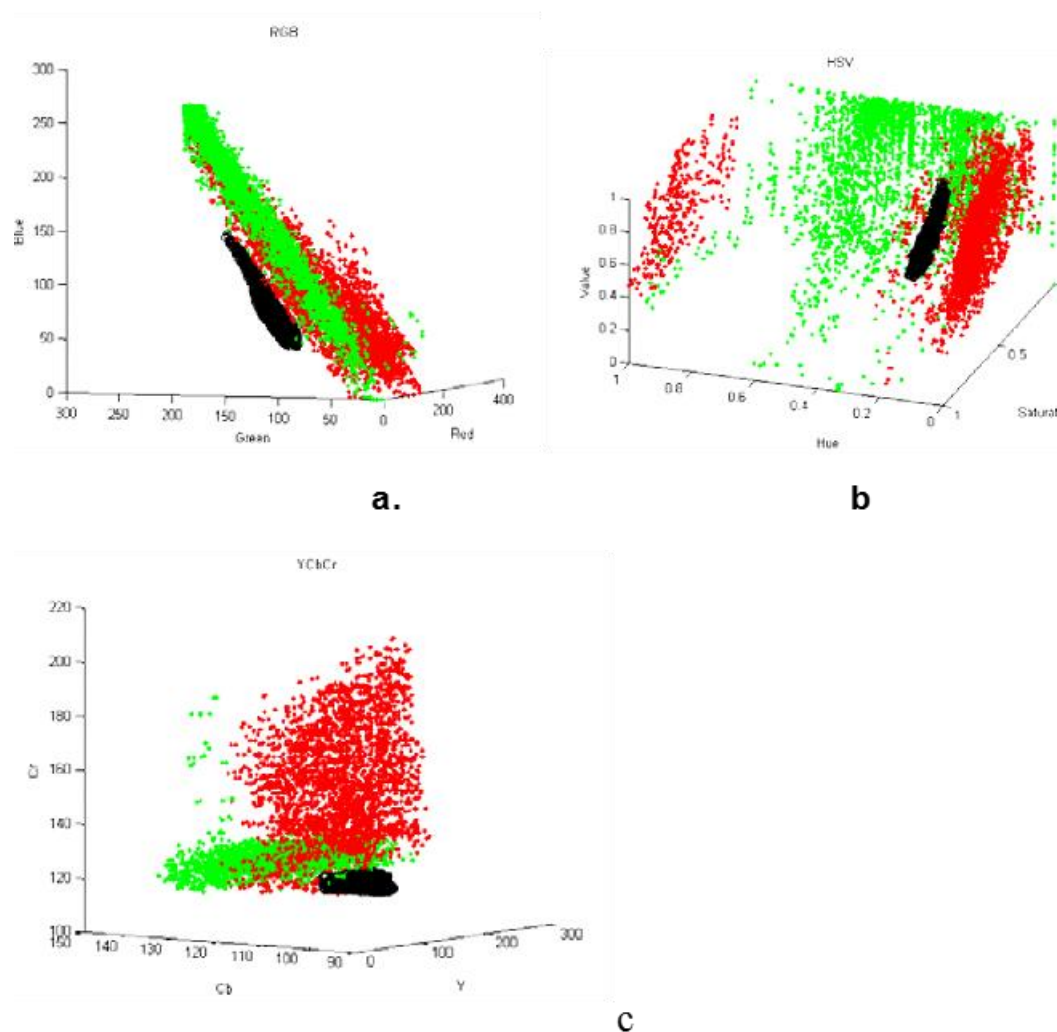


Gambar 2-7. Representasi visual perhitungan kontur biner

2.2.4 Komponen Warna

Sebelum masuk dalam permasalahan *tracking* pemain, sangatlah penting untuk memisahkan citra ke dalam bidang warna tertentu, karena pengenalan yang digunakan oleh penulis disini adalah pengenalan obyek (*tracking*) didasarkan pada warna. Bidang warna adalah model matematis yang menjelaskan bagaimana warna

dapat direpresentasikan dengan angka. Bidang warna yang paling umum ada tiga yaitu: RGB, HSV, dan YCbCr. Gambar 2-8 adalah perbandingan format warna yang dipilih untuk proses *tracking* pemain yang telah dilakukan penelitiannya oleh [33]. RBG (*Red, Blue, dan Green*) adalah bidang warna yang bersifat *additive* atau dengan kata lain bila kita ingin mendapatkan warna tertentu diluar dari warna yang disediakan maka tiga komponen warna primer: merah, biru dan hijau. HSV (*Hue, Saturation dan Value*) tidak bersifat *additive* seperti RGB karena semua informasi di enkapsulasi dalam tiga komponen variabel.



Gambar 2-8. Perbandingan format warna: RGB(a), HSV(b), dan YcbCr(c) [33]

Hue adalah panjang gelombang diantara spektrum cahaya tampak dimana output energi dari sumber ini adalah yang paling tinggi, *Saturation* adalah ekspresi dari *bandwith* relatif dari output sumber cahaya yang tampak mata, dan *Value* adalah ekspresi relatif dari intensitas energi output cahaya (disebut sebagai *brightness/kecerahan*). YCBCR bukan merupakan format bidang warna absolut, hal ini adalah salah satu cara RGB melakukan *encoding* dari informasinya. Y adalah bagian luma beresolusi tinggi bila ditransmisikan membutuhkan *bandwidth* tinggi. Cb dan Cr adalah perbedaan biru dan perbedaan merah pada komponen *chroma*, dimana resolusi dari bagian ini dapat diefisienkan. Oleh karena encoding yang berorientasi pada efisiensi maka format warna ini sering digunakan pada bidang fotografi digital.

2.2.5 Kalman Filter

Algoritma Kalman filter pada dasarnya menghitung besarnya *gain* Kalman, mengestimasi variabel *state* dan mengestimasi error kovarians setiap indeks waktu. Kalman filter ini dibentuk dengan menggunakan analisa *state-space* dengan hukum Newton ke dua. Dengan mengasumsikan bahwa proses pergerakan adalah sebuah translasi perpindahan posisi dengan jarak tertentu dan proses perpindahan tersebut bergerak relative terhadap waktu Δt . Proses pergerakan ini diasumsikan bahwa benda bergerak pada bidang permukaan datar. Akibatnya pergerakan benda dengan hambatan oleh karena permukaan tanah yang tidak rata sehingga mengakibatkan energi tiap-tiap pemain untuk bergerak terhambat diabaikan. Pemilihan asumsi ini didasarkan pada obyek yang akan dianalisa bergerak pada bidang datar yaitu lapangan sepakbola [4]. Asumsi kedua adalah pada lokasi pergerakan pemain bergerak dari kondisi inisial (awal) dari diam menuju bergerak dengan kecepatan awal adalah 0 dan percepatan adalah 0. Kalman filter pada hakekatnya adalah metode guna melacak obyek tunggal, apabila diaplikasikan untuk melacak obyek banyak maka harus disertai dengan algoritma penugasan (*assignment*), yaitu metode yang berfungsi sebagai pemberi label obyek referensi Q_{mea} , sebagai salah satu obyek hasil estimasi Q_{est} . Q adalah sebuah container matriks yang berisi lokasi para pemain. Jumlah obyek yang terdeteksi

adalah n dan Q_0 adalah inisial posisi obyek. Tiap-tiap Q_t merepresentasikan deteksi frame ke- t .

$$Q_t = A \cdot Q_0 + B \cdot \xi \quad (2-10)$$

Dimana dari hukum Newton yang kedua, didapatkan beberapa matriks state untuk kondisi lokasi (x, y) , kecepatan (\dot{x}, \dot{y}) , percepatan (\ddot{x}, \ddot{y}) dan jeda waktu Δt obyek bergerak dari suatu lokasi ke lokasi yang lain. Matriks A disebut sebagai matriks besaran dari matriks Q_0 dimana matriks ini berisi informasi lokasi obyek dan kecepatan obyek. Matriks B adalah matriks perubahan waktu Δt dan ξ adalah matriks percepatan obyek. Persamaan (2-10) merupakan hasil dari persamaan kinematika gerak lurus pada waktu kontinyu persamaan (2-11).

$$p(t) = p_0 + v_0 t + \frac{1}{2} \cdot a t^2 \quad (2-11)$$

$p(t)$ adalah posisi setiap saat, p_0 adalah posisi inialisasi awal pada nilai posisi satu dimensi, v_0 adalah kecepatan awal satu dimensi, a adalah percepatan pada waktu ke t satu dimensi.

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Q_t = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}; \quad (2-12)$$

$$B = \begin{bmatrix} \frac{1}{2} \Delta t^2 & \frac{1}{2} \Delta t^2 & \Delta t & \Delta t \end{bmatrix}; \quad \xi = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} \quad (2-13)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (2-14)$$

Prediksi kovarians Σ estimasi yang error, adalah sebagai berikut

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_x \sigma_y \\ \sigma_y \sigma_x & \sigma_y^2 \end{bmatrix} \quad (2-15)$$

Dimana $\sigma_x = \frac{1}{2} \Delta t^2$ dan $\sigma_y = \Delta t$, maka

$$\Sigma = \begin{bmatrix} \frac{1}{4} \Delta t^4 & \frac{1}{2} \Delta t^3 \\ \frac{1}{2} \Delta t^3 & \Delta t^2 \end{bmatrix} \text{ di dalam 2D} \quad (2-16)$$

Persamaan 3.11 bila diubah untuk memperoleh persamaan state lengkap menjadi:

$$\Sigma_{xy} = \begin{bmatrix} \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & 0 & \Delta t^2 & 0 \\ 0 & \frac{\Delta t^3}{2} & 0 & \Delta t^2 \end{bmatrix} \quad (2-17)$$

Dimana $\Sigma_{xy} = P_0$ adalah P , awal inialisasi. Untuk mengontrol penilaian. Sebagai pengontrol pengukuran, bila Kalman filter menggunakan Kalman gain $K_G = 1$, maka pengukuran akurat, akan tetapi estimasi menjadi tidak stabil, sebagai ganti bila $K_G = 0$, pengukuran menjadi tidak akurat, tetapi estimator stabil. Persamaan kalman gain adalah:

$$K_G = P_C \cdot C^T \cdot \text{inv}(C \cdot P_C \cdot C^T + E_Z) \quad (2-18)$$

P_C adalah proses kovarian matriks yang merepresentasikan error sebagai berikut:

$$\text{Dimana: } P_{ct} = A \cdot P_{c.0} \cdot A^T + \Sigma_{xy} \quad (2-19)$$

$$E_Z = \begin{bmatrix} \varepsilon_x & 0 \\ 0 & \varepsilon_y \end{bmatrix}$$

E_Z adalah konstanta eror pengukuran dalam state fungsi ukur. Penugasan pekerja dibutuhkan untuk memberikan label pada tiap-tiap obyek, ketika obyek bergerak maka label tersebut harus mengikutinya. Ketika lokasi pengukuran Q_{est} dari Q_t diperoleh maka algoritma penugasan Hungarian diaplikasikan untuk menempatkan lokasi pemain pada kolom label posisi. Setelah Q_{est} diperoleh dari penugasan algoritma Hungarian, Q_{est} diperbaharui dengan menggunakan persamaan sebagai berikut.

$$\hat{Q}_{est t} = Q_t + K_G (\hat{Q}_{mea}^T - C \cdot Q_t) \quad (2-20)$$

Pembaharuan perhitungan kovarians adalah sebagai berikut:

$$P_{ct} = (I - K_G \cdot C) P_C \quad (2-21)$$

2.2.6 Penugasan dalam Multi Point Tracking

Metode Hongaria adalah algoritma optimasi kombinatorial yang menyelesaikan masalah berdasarkan pembagian kerja dalam waktu polinomial. Algoritma ini mudah dimengerti dan diterapkan untuk menyelesaikan soal yang berupa penugasan dengan cara menemukan pemasangan sempurna. Algoritma Hungarian ini menghitung penugasan sebuah posisi kedalam matriks label. *Cost* matriks algoritma ini diperoleh dari jarak euklidian dari Q_{est} dan Q_{mea} . Keuntungan terbesar penggunaan algoritma Hungarian adalah kompleksitas algoritmanya yang polinomial. Metode yang digunakan dalam algoritma Hungarian dalam memecahkan masalah sangat sederhana dan mudah dipahami. Penerapannya bahwa setiap sumber daya harus ditugaskan hanya untuk satu pekerjaan. Untuk suatu masalah penugasan $n \times n$, jumlah penugasan yang mungkin dilakukan sama dengan $n!$ (n faktorial) karena berpasangan satu per satu. Penggunaan algoritma ini dikarenakan algoritma ini ditujukan untuk menutaskan satu peran terhadap satu pemain berdasarkan nilai matriks biaya terendah yang dimiliki antara tiap-tiap peran dan tiap-tiap pemain. Dalam penelitian ini, penulis menggunakan jarak euclidean untuk mencari kedekatan antara pemain dan peran.

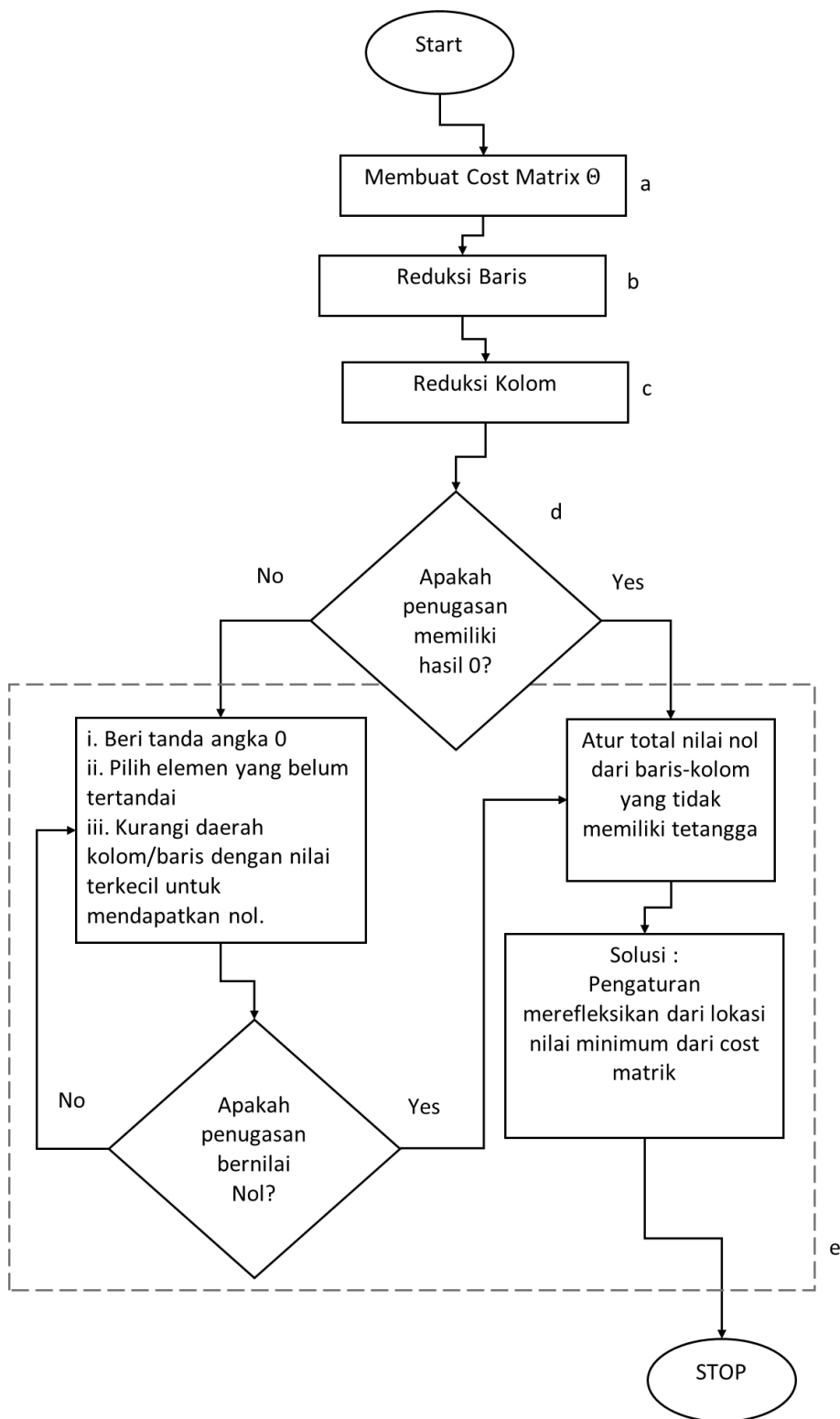
Algoritma Hungarian terdiri dari lima bagian penting, yaitu seperti yang tertampil pada Gambar 2-9 (a) pembuatan *cost* matriks Θ dengan mencari jarak euclidean Q_{est} dan Q_{mea} . Gambar 2-9 (b) Proses pereduksi baris dilakukan dengan mengurangi elemen yang masuk dari tiap-tiap baris dari semua elemen matriks Θ (c). Gambar 2-9 (d) adalah melakukan matriks pengecekan apakah penugasan memiliki jarak minimum =0 atau tidak. Gambar 2-9 (e) proses terakhir yang menampilkan pengaturan dan reduksi kolom-baris yang merefleksikan nilai lokasi minimum dari *cost* matriks (obyek baris ke- i , ditugaskan sebagai kolom ke- j).

Dalam penelitian ini penugasan Multi Point Tracking berfungsi untuk menyelaraskan urutan pemain hasil *tracking* dari frame sebelum ($t - 1$) dan frame saat ini (t). Dalam aplikasinya, peneliti menggunakan dua algoritma Hungarian untuk menugaskan pemain pada proses algoritma Kalman Filter (yaitu pada mengurutkan pemain berdasarkan matrik biaya antara lokasi pemain estimasi

dan lokasi pemain hasil prediksi) dan pada hasil keluaran algoritma Kalman Filter (mengurutkan lokasi pemain antara frame sebelum dan saat ini).

Algoritma Kalman filter sebenarnya merupakan algoritma *tracking* yang ditujukan untuk melakukan *tracking* satu obyek, akan tetapi pada perkembangannya algoritma ini dapat digunakan untuk melakukan pengenalan multi-obyek. Hal ini dilakukan dengan beranggapan bahwa masukan algoritma ini berupa matriks dengan ukuran $m \times n$ matriks. Nilai matriks ini haruslah kemudian dihitung untuk dengan step-step pada algoritma Kalman filter yang sama pada metode tracking satu obyek. Yang membedakan adalah, data masukan algoritma ini haruslah memiliki urutan yang harus sama antara frame sebelum dan sesudah. Untuk menjaga urutan inilah algoritma Hungarian (Munkres) dipekerjakan.

Tantangan dalam aplikasi algoritma Kalman filter multi-obyek adalah seberapa benar urutan data masukan (dimana data dibedakan dalam vector kolom). Sementara itu, pada praktiknya dilapangan, kondisi multi-obyek sering terjadi oklusi dan pergerakannya sangat cepat dan random. Untuk itulah maka, teknik estimasi masadepan perlu diberlakukan. Teknik ini digunakan dalam penelitian kali ini dengan mengasumsikan bahwa pergerakan antar frame memiliki selisih yang *affordable* sehingga jarak obyek frame saat ini yang terdekat dalam frame sebelumnya merupakan obyek yang dikehendaki. Proses pengambilan jarak ini, penulis menggunakan teknik penentuan perbedaan *Euclidean*, karena dipandang bahwa teknik ini cocok pada obyek dua dimensi (row-collumn). Selanjutnya asumsi yang digunakan dalam penelitian kali ini adalah semua citra yang direkam oleh video merupakan citra dua-dimensi, sehingga kondisi keadaan *depth* suatu obyek tidak dianalisa. Dengan kata lain, peneliti hanya mengambil keadaan obyek tersebut seperti yang tertampil pada hasil video saja.

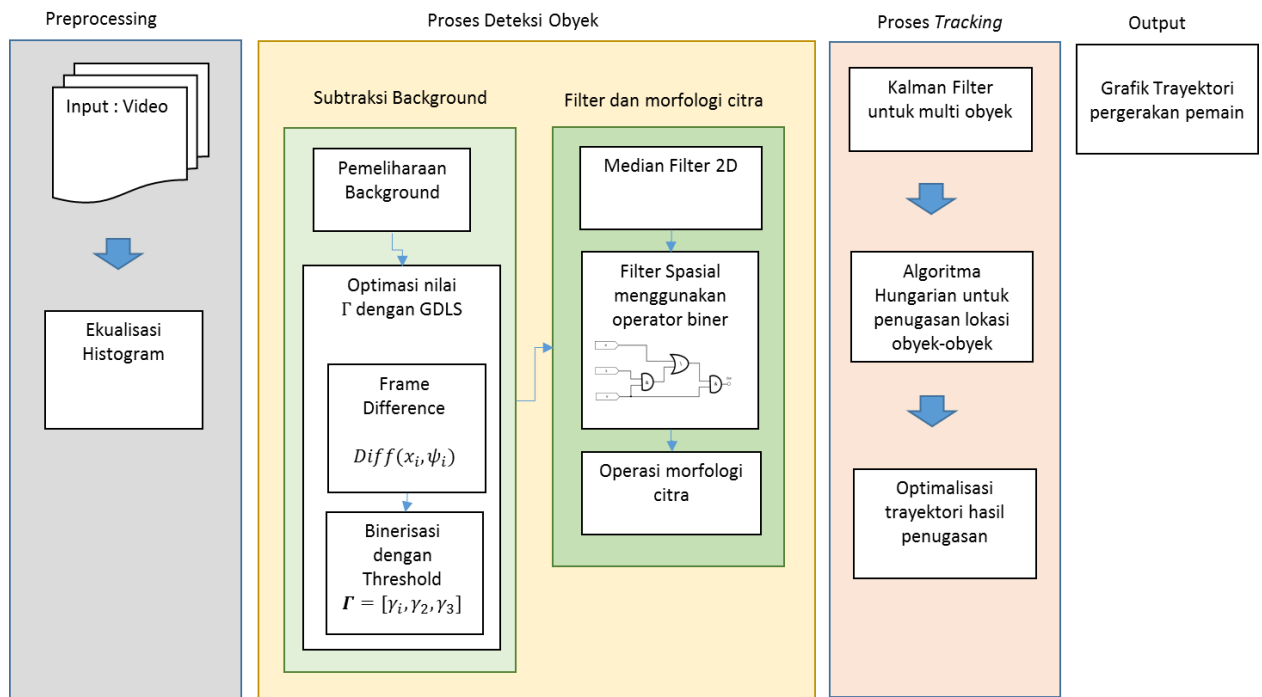


Gambar 2-9. Diagram alir algoritma Hungarian [32]

BAB 3

METODOLOGI PENELITIAN

Pada tahapan ini dijelaskan langkah-langkah metodologi penelitian secara sistematis. Hal ini dapat dilihat pada Gambar 3.1. Dalam penelitian kali ini, metodologi penyelesaian permasalahan terdapat tiga bagian utama yaitu preprocessing, deteksi obyek dan *tracking*. Berikut adalah tahapan-tahapan penjelasan ketiga tahapan tersebut.



Gambar 3-1. Diagram alur perancangan sistem

3.1 Preprocessing

Dalam tahapan ini diharapkan mendapatkan tiga syarat citra input ideal yang siap diolah kedalam proses deteksi obyek yaitu citra masukan yang seragam dalam tiap-tiap frame dalam tingkat terang-gelap (*brightness*), *error-free* pada hasil kompresi (avi, mpg, mp4, dsb), dan bebas dari *noise* perangkat keras. Walaupun demikian, tidak semua data ideal, ada beberapa data yang mengalami kerusakan akibat *noise*, sehingga mengakibatkan detektor obyek menganggap *noise* tersebut

sebagai obyek (*foreground*). Bagian 3.1.1 dan 3.1.2 menjelaskan tentang karakteristik input dan metode ekualisasi histogram untuk menangani permasalahan noise dan pencahayaan yang tidak merata.

3.1.1 Input Video

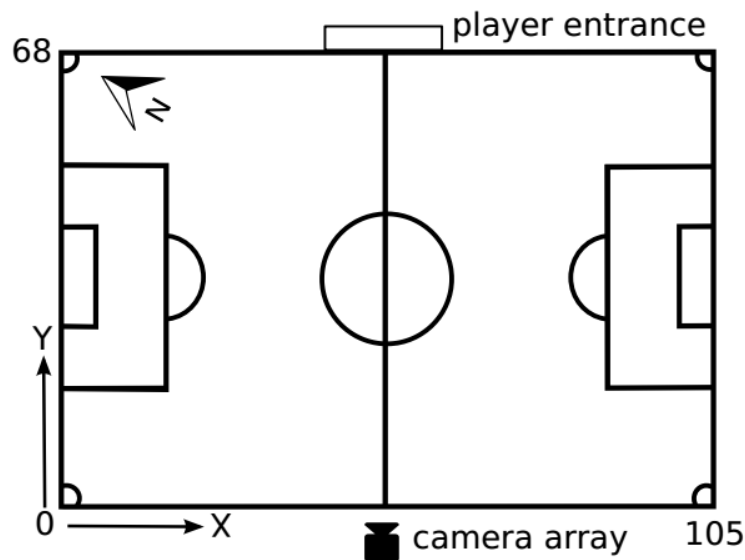
Input diperoleh dari dataset Alfheim [34] dimana dataset ini adalah dataset yang digunakan sebagai acuan pengujian metode *tracking* pada para pemain satu waktu permainan penuh. Karakteristik dataset ini adalah sebagai berikut yang tertampil pada tabel 3.1 sebagai karakteristik kamera yang digunakan untuk merekam dan tabel 3.2 untuk meng-encode H264. Dalam tabel ini juga dapat diketahui bahwa total jumlah frame yang dianalisa adalah 26.222 buah setara dengan 17,5 menit.

Tabel 3-1. Tabel karakteristik kamera yang digunakan untuk input

Kamera	<i>Basler acA1300-30gc</i>	<i>Basler acA200-50gc</i>
Resolusi	<i>1280 x 960</i>	<i>1920 x 1080</i>
Frame rate	<i>30 fps</i>	<i>25 fps</i>
Model Lensa	<i>3.5 mm Kouva-LM4NCL</i>	<i>8mm Azure-0814M5M</i>
Kegunaan	<i>Single-Wide-angle videos (1280x960)</i>	<i>Stitched Panoramic Video (4450x2000)</i>
Total jumlah frame	26.222 frame atau 17,4813 menit pertandingan dengan 25 fps.	

Tabel 3-2.. Parameter yang digunakan untuk meng-encode H264

Profile	Tinggi
Preset	Ultrafast
Tune	Zerolatency
Bidang warna	YUV 420 planar
GOP	3 x fps



Gambar 3-2. Layout Kamera pada Stadion Alfheim [7]

Kamera perekam menggunakan Basler yang ditambah dengan model lensa Azure 8mm. Kegunaan kamera tersebut adalah untuk merekam gambar *panoramic*. Secara software video diencode dengan menggunakan resolusi 4450x2000 pixel. Kamera panoramic ditempatkan pada satu tempat dimana lokasi pengamatan tersebut diatur sedemikian rupa sehingga semua pemain dapat terekam dengan baik. Citra video yang diambil dalam situasi yang khusus dimana citra tersebut didapatkan dari perekaman kamera yang statis pada pinggir lapangan, diperlihatkan pada Gambar 3-2 dibawah ini. Lebar lapangan bola adalah 105x68 m dan dalam lapangan inilah peneliti melakukan set *ROI*, dimana suatu obyek didefinisikan sebagai semua obyek yang bergerak dalam area *ROI* ini. Oleh karena menggunakan single kamera array Basler acA200-50gc (Tabel 3-1) maka efek dari pengambilan gambar akan memiliki efek melengkung, dan hal ini dapat dilihat pada Gambar 3-5 dibawah ini. Seperti yang telah dijelaskan diatas bahwa semua video dikodekan dengan menggunakan format H264 menggunakan library libx264. Kesemua parameter dipilih oleh profile kodek, tiap-tiap hasil rekaman dipisah dalam file segmen 3 detikan, yang memiliki kecepatan 3xFPS. Dalam hal penomoran file, digunakan empat digit sekuensial diikuti oleh waktu pengambilan. Dataset ini juga telah diteliti oleh Zivkovic untuk membuat subtraksi *background* yang efektif dan adaptif dengan menggunakan metode estimasi kerapatan distribusi *pixel* [6].

3.1.2 Ekualisasi Histogram

Seperti yang tertampil pada Gambar 3.2, metode ekualisasi histogram digunakan untuk menormalkan distribusi persebaran nilai *pixel* sehingga diharapkan gambar dapat memiliki persebaran nilai *pixel* yang merata. Dengan kata lain gambar yang memiliki kumpulan nilai *pixel* rendah dapat dipaksa untuk menaikkan nilai *pixel*-nya sehingga kecerahan gambar dapat ditingkatkan. Persamaan ekualisasi histogram adalah sebagai berikut:

$$P_r(r_k) = \frac{n_k}{n}; 0 \leq r_k \leq 1; k = 0, 1, \dots, L-1 \quad (3-1)$$

Dimana:

- L adalah jumlah level
- $p_r(r_k)$: jumlah probabilitas kemunculan level ke- k
- n_k : jumlah kemunculan level k pada citra
- n adalah total jumlah *pixel* pada citra

3.1.3 Perancangan Metode

Dalam tahapan ini dijelaskan mengenai proses pemilihan metode yang digunakan berdasarkan tantangan yang dihadapi pada dataset [34]. Daftar pengenalan tantangan didapatkan dari tabel Bouwmans [5]. Tabel 3.3 dibawah ini menjelaskan mengenai tantangan yang dihadapi tersebut. Dalam Tabel 3-3 terdapat 11 tantangan yang harus dihadapi dalam deteksi obyek dari sumber kamera. Tantangan-tantangan tersebut adalah: Kamera yang bergerak-gerak; Kualitas perekaman kamera yang buruk yang menghasilkan noise; Fitur pengaturan otomatis yang mengganggu deteksi obyek (otomatis zoom); Jeda waktu perekaman yang menghasilkan perubahan posisi yang ekstrim; Perubahan iluminasi yang menyebabkan tingkat kecerahan gambar menjadi lebih terang atau gelap; *Foreground aperture*, *background* yang bergerak; *Background* yang berubah secara dinamis misalkan obyek diambil didepan *digital signage*; Obyek yang aktif bergerak di awal, dan obyek yang bergerak pasif diawal dan tiba-tiba obyek bergerak aktif.

Tabel 3-3. Tabel tantangan yang dihadapi pada dataset Alfheim [34]

Tantangan	Penjelasan	Observasi
Kamera yang bergerak-gerak	Obyek dengan kamera bergerak-gerak menghasilkan komponen <i>background</i> yang tidak merata	Tidak
Kualitas perekaman kamera yang buruk	Kualitas kamera yang buruk akan menghasilkan <i>noise</i> visual.	Tidak
Fitur pengaturan otomatis kamera (fokus otomatis)	Fitur pengaturan otomatis mengakibatkan komponen <i>pixel</i> citra obyek mengalami perubahan mendadak sehingga detektor kesulitan beradaptasi.	Tidak
Jeda waktu perekaman	Jeda perekaman akan menghasilkan jeda perubahan posisi yang ekstrim dari posisi awal menuju selanjutnya	Tidak
Perubahan iluminasi	Perubahan distribusi pencahayaan dapat dilihat dari degradasi terang-gelap pada lapangan, dari daerah tengah lapangan yang mendapatkan banyak cahaya lampu sampai daerah pinggiran yang mendapatkan sedikit cahaya lampu.	Ya
Teknik <i>Bootstrapping</i> ²	Teknik untuk mencari pemain spesifik (satu dari banyak obyek)	Tidak

² *Bootstrapping* [45] adalah sebuah teknik pembelajaran klasifikasi obyek dengan menggunakan perpaduan banyak *classifier* yang dipasang seri atau parallel.

Tabel 3-4. Tabel tantangan yang dihadapi pada dataset Alfheim [34] (lanjutan)

Tantangan	Penjelasan	Observasi
<i>Foreground aperture</i>	Apabila <i>background</i> model beradaptasi terlalu cepat, maka perubahan pergerakan <i>pixel</i> latar depan (<i>foreground</i>) akan diserap sebagai latar belakang (<i>background</i>), sehingga mengakibatkan bentuk obyek tidak sempurna.	Ya
<i>Background</i> yang bergerak	<i>Background</i> bergerak	Tidak
<i>Background</i> yang berubah-ubah secara dinamis	Perubahan bentuk <i>background</i> secara dinamis	Tidak
Obyek yang aktif bergerak di awal	Pada awal obyek sudah terlihat aktif bergerak	Ya
Obyek yang bergerak pasif di awal	Pada awal obyek terlihat tidak active akan tetapi pada frame tertentu baru kemudian aktif bergerak	Ya
Bayangan	Tiap-tiap obyek mempunyai bayangan akibat dari cahaya lampu	Ya

Penulis mengerjakan update *background* subtraction guna menyelesaikan tantangan yang dihadapi. Update *background* ini berfungsi memperbaharui informasi perubahan *background* akibat pencahayaan dan proses encoding format video.

Dari hasil observasi yang dilakukan (Tabel 3-3) dapat disimpulkan bahwa, dataseet Alfheim [34] memiliki karakteristik sebagai berikut:

- Perubahan iluminasi pada lapangan olehkarena persebaran cahaya lampu
- Obyek bergerak secara dinamis dan ada saat dimana model *background* tidak dapat beradaptasi terhadap perubahan obyek.
- Obyek yang aktif bergerak sejak di awal frame

- Obyek yang saat diawal frame pasif tetapi pada waktu tertentu obyek baru bergerak
- Adanya bayangan pada tiap obyek akibat dari cahaya lampu.

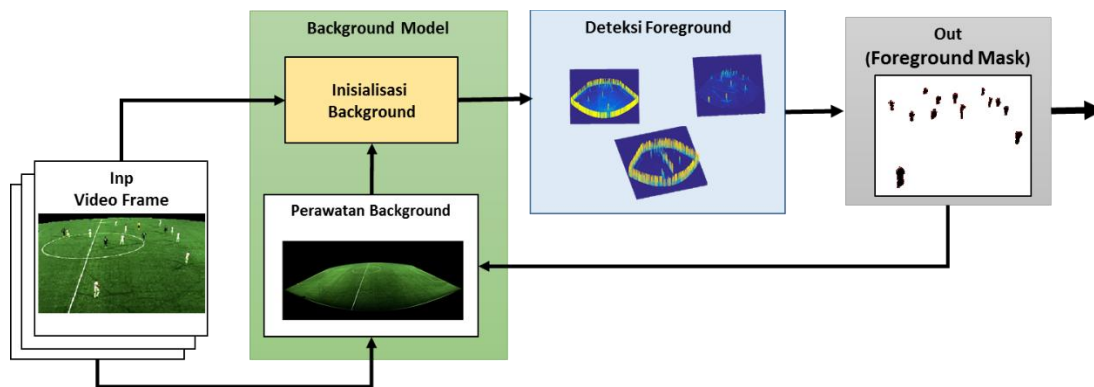
Menurut Bauwmans [5] subtraksi *background* berdasarkan statistik cocok untuk menangani perubahan model yang statis dan metode subtraksi ini termasuk metode yang mempunyai beban komputasi lebih rendah ketika memproses data gambar dengan luas (dataset ini memiliki frame luas dengan berukuran 4450x2000 pixel) dari pada jenis *background* subtraksi terkini. Jenis *background* subtraksi ini memiliki kelemahan pada pengenalan bayangan obyek. Karena bayangan bergerak menempel pada obyek, maka bayangan ini juga dianggap sebagai obyek. Oleh sebab itu, filter bayangan akan di tambahkan pada proses selanjutnya.

3.2 Proses Deteksi Obyek

Pada bagian ini dikerjakan proses deteksi obyek dari mulai dengan penentuan *background* dengan metode statistik, dilanjutkan dengan penentuan *threshold* menggunakan metode pembelajaran menggunakan algoritma GDLS (Gradient Decend with Linear Search), setelah itu, proses binerisasi dengan *threshold* menggunakan hasil proses pembelajaran GDLS dilakukan. Proses terakhir untuk memperbaiki hasil output yang terkena noise dan bayangan adalah mengaplikasikan proses filter ranah spasial median filter dan filter operasi biner. Efek samping dari hasil filter spasial tersebut adalah terkikisnya obyek sehingga patahan bagian obyek terjadi, untuk menanggulangi hal ini maka gabungan operasi morfologi dilasi dan closing diaplikasikan.

3.2.1 *Background Subtraction*

Background subtraction pada hakikatnya adalah memisahkan *background* dan *foreground* dalam suatu citra. Pemisahan ini dengan mengurangi suatu frame dengan *background* modelnya. *Background* model didapatkan dari proses rata-rata frame n -cuplik. Proses umum yang terjadi pada subtraksi *background* [5] adalah sebagai berikut ini.



Gambar 3-3. Proses subtraksi background

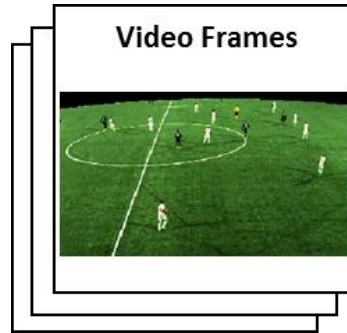
Dalam proses gambar 3.2 dapat dilihat bahwa data input dimasukkan kedalam dua proses sekaligus yaitu inisialisasi *background* dan bagian perawatan *background*, bagian inisialisasi *background* dalam mengerjakan keluaran untuk deteksi *foreground* membutuhkan keluaran dari perawatan *background*. Hasil dari deteksi *foreground* akan menghasilkan gambar biner atau disebut sebagai *foreground mask*. Detail penjelasan tiap-tiap proses yang terjadi pada empat tahapan-tahapan ini adalah sebagai berikut:

3.2.1.1 Tahapan Inisialisasi background

Dalam tahapan inisialisasi *background* ini (gambar 3.3) terdiri dari proses pemuatan frame-frame untuk membuat *background* buatan (*artificial background*). Perlu diketahui bahwa dalam dataset [34] tidak ditemukan satu frame yang memang digunakan sebagai *background* referensi. Jadi, *background* buatan dibuat dengan cara merata-rata cuplikan n frame. Keuntungan dari pembuatan *background* buatan ini adalah karena suatu sebab eksternal pencahayaan dari lampu sorot berubah-ubah, sehingga nilai pixel dinamis seiring dengan perubahan pencahayaan tersebut, maka perubahan nilai pixel tersebut akan mengakibatkan ketidak stabilan nilai *threshold* untuk deteksi *foreground*. Sebaliknya, jika penggunaan *background* buatan yang dapat mengikuti perubahan pencahayaan dipakai, maka nilai *threshold* dapat terjaga dengan baik.

Persamaan tahapan inialisasi *background* adalah sebagai berikut:

$$\phi = [I_1 \quad I_2 \quad \dots \quad I_n] \quad (3-2)$$



Gambar 3-4.. Inialisasi background

3.2.1.2 Tahapan Perawatan *background*

Tahapan perawatan *background* akan mengelola perubahan dari tahapan inialisasi dengan cara merata-rata cuplikan inialisasi *background* ϕ .

$$\phi_{t+1}(x, y) = (1 - \alpha)\phi_t(x, y) + \alpha I_t(x, y) \quad (3-3)$$

Dimana α adalah laju rata-rata dari proses pembelajaran *background*, dan nilai α bernilai: $0 \leq \alpha \leq 1$. Oleh karena *background* konstan maka nilai $\alpha = 0$, sehingga persamaan 3.3 menjadi:

$$\phi_{t+1}(x, y) = \phi_t(x, y) \quad (3-4)$$

Sehingga dalam semua sample frame n *background* akan seragam. *Background* buatan memiliki persamaan sebagai berikut:

$$\hat{\phi}_t = \frac{1}{N} \sum_{i=1}^N I_t(x, y) \quad (3-5)$$

Dan N akan diupdate setiap 50 frame untuk menjaga keseragaman *background*. Gambar 3.4 menunjukkan $\hat{\phi}_t$ pada $N = 1000$.

3.2.1.3 Tahapan deteksi *foreground*

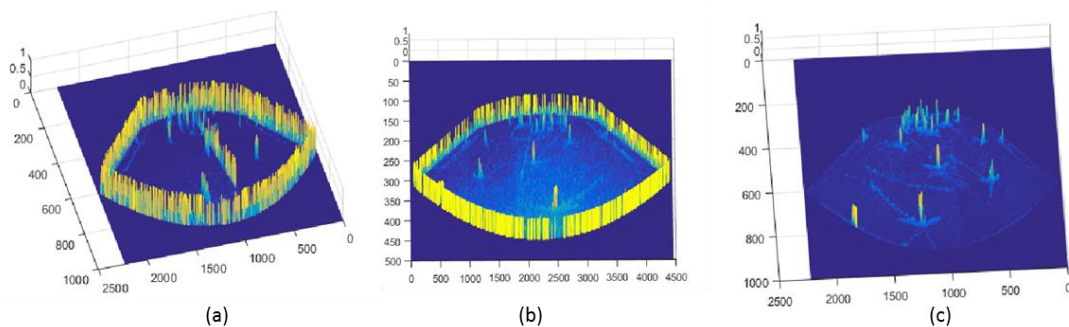
Foreground dideteksi dengan melakukan pengurangan antara *background* model ϕ dengan citra input I . Dan $f(\Gamma)$ adalah $|I_t - \hat{\phi}_t|$. Hasil dari pengurangan tersebut ditunjukkan pada Gambar 3-5. Sebelum menuju proses binerisasi,

penentuan *threshold* dimaksudkan untuk menentukan nilai ambang yang dapat memisahkan pixel *background* atau *foreground* dari sebuah citra.



Gambar 3-5. Background model pada $N=1000$.

Pada tahapan ini diperkenalkan algoritma optimisasi *Gradient Descend Line Search* (GDLS) untuk mencari nilai *threshold* yang membuat fungsi $\arg \min_{\Gamma} f(\Gamma)$. Nilai *threshold* ini akan digunakan pada tahapan binerisasi. Algoritma GDLS diaplikasikan dalam proses deteksi *foreground*. Algoritma ini merupakan algoritma orde satu yang memerlukan periode T iterasi. Dimana menemukan $\min f(\Gamma)$ berarti menemukan formasi terbaik dalam $\Gamma = [\gamma_1 \gamma_2 \gamma_3]$ dan $\Gamma \in \mathbb{R}^n$. Fungsi $f(\Gamma)$ memiliki karakteristik bahwa: $f(\Gamma) \rightarrow \|g(\Gamma) + \psi(\Gamma)\|$ dengan $\psi(\Gamma) = -g(\Gamma) + f(\Gamma)$. Dapat dikatakan bahwa $f(\Gamma)$ adalah sisa dari persamaan $\psi(\Gamma)$. Dalam penelitian ini, ditegaskan bahwa citra ψ adalah *ground truth* dari dataset [34].



Gambar 3-6. (a) adalah contoh hasil subtraksi citra pada komponen Hue, (b) hasil subtraksi pada komponen Saturation, dan (c) hasil subtraksi pada komponen Value.

Dari kalkulus, didapatkan bahwa mencari fungsi minimum $f(\Gamma)$ berarti mencari turunan pertama³ sama dengan nol $\frac{\delta f(\Gamma)}{d(\Gamma)} = 0$ dari fungsi tersebut dan hal ini secara otomatis memaksimalkan fungsi peluang jika dan hanya jika g terjadi apabila f terjadi: $p(g|f)$. Jadi, secara informal dapat dikatakan bahwa: karena Γ adalah unsur warna HSV maka Γ berdimensi tiga yaitu $[\gamma_1, \gamma_2, \gamma_3]$, maka iterasi algoritma ini dilakukan sebanyak $3 \times N$ kali. Pada tiap-tiap step fungsi hasil dimasukkan kedalam variabel δ dan kemudian fungsi $G(\cdot)$ diatur dengan memenuhi persamaan $G(\Gamma) < G(\Gamma_{step})$. Pemodelan bobot gradient $-\eta_t \nabla_{\Gamma(t-1)}$ dijelaskan oleh $-\eta_t \nabla_{\Gamma(t-1)} \approx \|f(\Gamma) - \psi(\Gamma)\|$ dengan $G(\Gamma) = \|f(\Gamma) - \psi(\Gamma)\| < G\Gamma_{step}$. Apabila total step telah dilakukan, hasil $\arg \min_{\Gamma} f(\Gamma)$ diperoleh dari melakukan penyortiran nilai δ . Dapat dilihat dari Algoritma 3.1 yang merupakan ringkasan dari tahapan proses pencarian nilai threshold. *Algoritma 3-1* dijelaskan sebagai berikut ini, diawali dengan inisialisasi input yaitu variabel $f_t(\Gamma)$ sebagai inputan, dimana $f_t(\Gamma)$ adalah urutan video hasil dari proses *image difference* atau sering disebut subtraksi *background*. Dimana dalam variabel f ini berdimensi tiga (\mathbb{R}^3). Dimensi dalam Γ ini merepresentasikan matriks H, S, dan V. Untuk inisialisasi awal variabel $G\Gamma_{step t}$ diberi nilai sembarang antara $0 < G\Gamma_{step t} < 1$. $t = 1, 2, 3, \dots, N$ sama dengan jumlah frame percobaan. Ekspektasi dari keluaran *Algoritma 3-1* adalah jarak minimal δ dari persamaan $\|f(\Gamma) - \psi(\Gamma)\|$ yang bersesuaian dengan variabel Γ , adapun variabel $\Gamma = [\gamma_1, \gamma_2, \gamma_3]$ adalah variabel threshold yang digunakan untuk threshold binerisasi obyek. Perulangan dilakukan dalam *nested looping* (perulangan bersarang) dengan mencari satu per satu nilai jarak $G(\cdot)$ yang kurang dari threshold $G\Gamma_{step}$

³ Dari persamaan (13,41) Bovik [47] dapat diartikan sebagai: fungsi turunan digunakan untuk memperoleh informasi perubahan relatif antara pixel-pixel. Kuantitas relatif ini dapat berupa posisi gambar, atau mungkin suatu parameter seperti faktor skala. Kuantitas relatif ini disebut sebagai gradien.

```

Input : Urutan Video  $\{f_t(\Gamma)\}$ 
          Urutan Video  $\{\psi_t(\Gamma)\}$ 
          inialisasi  $G\Gamma_{step} = rand(.)$ 
           $t = 1, \dots, N.d$ 
Output :  $arg\ min\ \delta$  pada variabel  $\gamma_1, \gamma_2, \gamma_3$ .
Inialisasi  $\gamma_1$  for 1:  $T_1$ ,
              $\gamma_2$  for 1:  $T_2$ ,
              $\gamma_3$  for 1:  $T_3$ ,
             Mencari step  $\eta_t$  sehingga:
              $G(\Gamma) = \|f(\Gamma) - \psi(\Gamma)\| < G\Gamma_{step}$ 
             Masukkan ke variabel  $(\delta \rightarrow \delta_t) \leftarrow -\eta_t \nabla_{\Gamma(t-1)} - \psi(\Gamma)$ 
             Perbaharui variabel:  $\delta_t \Gamma_t \leftarrow \Gamma_{t-1} + \delta_t$  .
             End of  $\gamma_3$ 
             End of  $\gamma_2$ 
             End of  $\gamma_1$ 

```

Algoritma 3-1. Aplikasi algoritma GDLS

Sehingga akan didapatkan matriks selisih δ sebesar $1 \times \text{jumla iterasi}$. Dari persamaan diatas dapat disimpulkan bahwa *Algoritma 3-1* adalah algoritma yang mencari nilai selisih δ terkecil untuk mendapatkan $\gamma_1, \gamma_2, \gamma_3$ untuk proses threshold selanjutnya seperti yang dijelaskan pada sub bab 3.2.1.4.

3.2.1.4 Tahapan binerisasi masking foreground

Pada tahapan ini dilakukan proses *threshold* dari hasil $[\gamma_1, \gamma_2, \gamma_3]$ pada δ minimum yang telah disorting pada proses sebelumnya. Proses *threshold* ini adalah sebagai berikut:

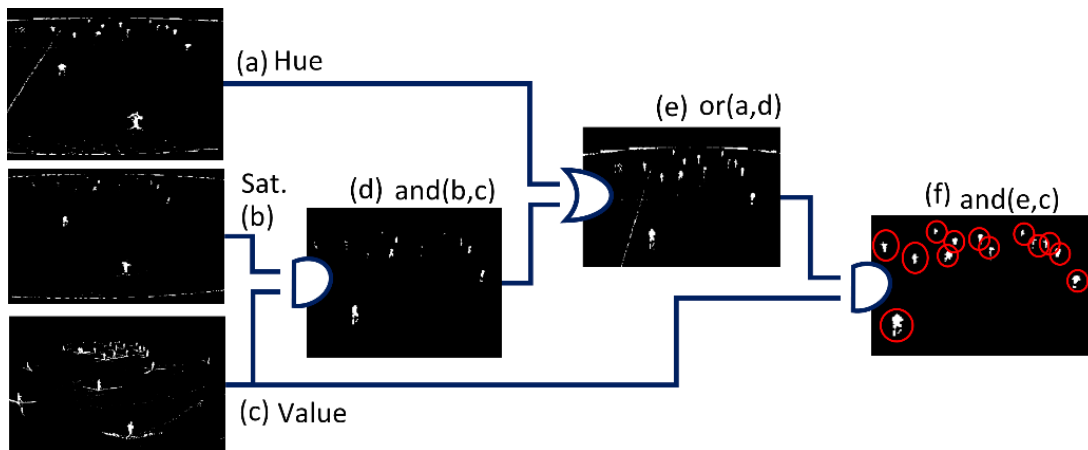
$$\begin{aligned}
 I_{B_Hue}(x, y) &= f_{Hue}(x, y) > \gamma_1 \rightarrow 1 \\
 I_{B_Sat}(x, y) &= f_{Sat}(x, y) > \gamma_2 \rightarrow 1 \quad , \text{ else : } 0 \\
 I_{B_Val}(x, y) &= f_{Val}(x, y) > \gamma_3 \rightarrow 1
 \end{aligned} \tag{3-6}$$

Dimana $I_{B_Hue}, I_{B_Sat}, I_{B_Val}$ adalah dimensi warna HSV (Hue Saturation Value) pada citra masukan. Hasil tahapan binerisasi ketika tiga kanal warna dapat dilihat sebagai berikut.

3.2.2 Filter dan Morfologi Citra

Bitwise operator didasarkan pada bentuk informasi spasial dari fungsi $f(\Gamma)$. Karena representasi format warna citra ada tiga dimensi (Hue, Saturation, dan Value) maka *threshold* harus diobservasi dalam ketiga format warna tersebut. Keluaran H,S, dan V diturunkan dengan menerapkan ambang batas T, untuk fungsi $f(\Gamma)$. Filter spasial tidak memerlukan perhitungan yang progresif, karena filter ini menerapkan perhitungan *array-ke-array*. Ke-progresif-an⁴ dalam komputasi akan menghasilkan alokasi memori lebih banyak, akibatnya akan menghasilkan kompleksitas yang tinggi dibanding dengan komputasi langsung (*direct computation*). Dalam Gambar 3-7 menunjukkan bagaimana filter dibentuk. Ini dimulai dengan fungsi AND dari komponen warna *saturation* (b) dan *value* (c) untuk menyaring bayangan pada *foreground*. Karena citra hasil memiliki bentuk yang tidak komplit maka operasi tambahan ditambahkan untuk mengembalikan informasi yang hilang dengan menambahkan informasi warna Hue (a). Jadi, operasi OR antara (d) dan (a) dihasilkan. Operasi ini menghasilkan bentuk obyek yang lengkap tetapi mendapat (menderita) bayangan sebagai obyek. Sehingga untuk final kita menggunakan operasi AND antara hasil sebelumnya (e) dengan komponen nilai (c) lagi untuk memperoleh output final (f). Tabel 3-5 menyajikan operasi detail sebagai tabel kebenaran gerbang logika filter.

⁴ Suatu kondisi komputasi dimana beban hasil komputasi berkembang secara bertahap seiring dengan jumlah penambahan iterasi. Hasil komputasi ini menyumbang beban komputasi *Processing Unit* mayoritas, sehingga apabila beban komputasi progressive ini dikurangi maka notasi big O berdasarkan kompleksitas algoritma yang dipakai akan menurun. Proses ini sering disebut optimasi secara process. Hasil yang didapatkan dari optimasi secara process ini adalah keadaan yang membutuhkan memory (cache) yang rendah, sehingga suatu process dapat diaplikasikan pada sistem yang rendah.



Gambar 3-7. Gerbang logika filter bitwise spasial

Tabel 3-5. Tabel kebenaran filter bitwise spasial

H	S	V	Out
0	0	0	0
0	0	1	0
...
0	1	1	1
1	1	1	1

Penulis melihat representasi warna dari Gambar 3.6 dan Tabel 4.3 saat pembentukan gerbang logika filter biner. Selama perhitungan filter ini, beban komputasi diukur sebesar $O(n)$ dengan n adalah rutine proses operasi elemen per elemen dalam matriks. Adapun demikian, dapat terlihat bahwa Tabel 3.5 merupakan tabel kebenaran operasi tersebut.

Obyek mengalami patahan akibat dari operasi filter ini, sehingga penambahan operasi morfologi dilakukan guna menyambung patahan dan membulatkan penipisan oleh karena operasi filter ini. Operasi morfologi yang diaplikasikan adalah dilasi dan *closing*. Algoritma closing pada sebuah citra f dengan elemen struktur s (disimbolkan dengan $f \circ s$ adalah dilasi \oplus yang diikuti dengan erosi \ominus).

$$f \bullet s = (f \oplus s_{rot})(-)s_{rot} \quad (3-7)$$

Sementara itu, proses dilasi adalah suatu proses penggabungan komponen 1 dengan berdasarkan operasi yang dikerjakan oleh matriks struktur elemen s . Representasi pusat masa didefinisikan setelah proses penghilangan noise yang diakibatkan oleh karena detektor memiliki false alarm saat melakukan pengenalannya. Dalam aplikasinya penulis selain menggunakan morfologi sebagai filter ranah spasial, juga menggunakan median filter 2D untuk memfilter derau (*noise*) akibat dari artefak *coding* format warna YUV menjadi JPEG. Walaupun demikian, hasil dari transformasi tersebut tidak terlalu menghambat akurasi pengenalan. Hasil akurasi pengenalan dapat dilihat pada Bab 4.

Proses filter biner ini didapatkan berdasarkan kondisi dataset yang unik, sehingga dapat dikatakan bahwa, tidak sembarang dataset dapat menggunakan filter ini. Kedepannya pengenalan *jersey* berbasis kecerdasan buatan terpimpin dapat diaplikasikan, guna mendapatkan hasil pengenalan yang tinggi. Proses ini merupakan proses krusial dalam melakukan pelacakan, karena akurasi yang tinggi dalam proses ini penentuan lokasi dengan menggunakan pusat masa diperoleh dari seberapa presisi luasan obyek terhadap bentuk riilnya. Subbab berikut akan menjelaskan secara teknis bagaimana proses penentuan lokasi diperoleh.

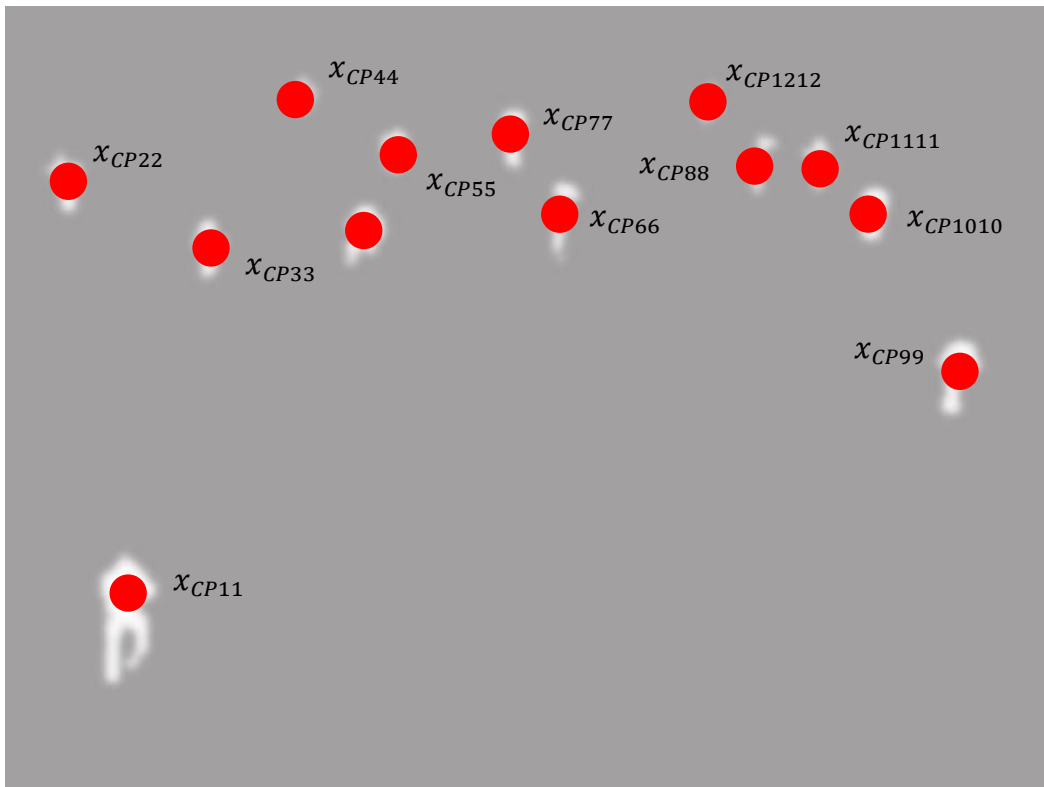
3.2.3 Proses Penentuan Lokasi

Penentuan titik koordinat pemain didapatkan dari menghitung pusat *massa* dengan menggunakan persamaan sebagai berikut.

$$x_{cm} = \frac{x_{max} + x_{min}}{2}$$

$$y_{cm} = \frac{y_{max} + y_{min}}{2} \quad (3-8)$$

Penentuan titik koordinat ini mewakili lokasi pemain keseluruhan, oleh sebab itu posisi pemain relative terhadap hasil pusat *massa*-nya.



Gambar 3-8. Representasi pusat *massa* sebagai perwakilan keberadaan pemain.

Representasi pusat *massa* ini dapat divisualisasikan sebagai berikut ini pada Gambar 3-8. Dalam penelitian kali ini peneliti menggunakan variabel Q_{mea} sebagai variabel untuk menyimpan lokasi pengukur (*measurement*). Q_{est} adalah variabel penyimpanan yang didapatkan dari matriks state hukum ke dua Newton. Selanjutnya pada subbab Proses Tracking berikut.

3.3 Proses Tracking

Menurut Lucey [35] [36] formasi adalah kumpulan dari label agnostic peran pemain. Dengan kata lain tiap-tiap peran (*role*) memiliki posisi relative terhadap peran lainnya (gelandang kiri bermain didepan bek kiri dan berada disebelah kiri dari gelandang tengah). Tiap peran dalam formasi adalah unik, yaitu tidak ada dua pemain memiliki peran yang sama pada suatu waktu, dan pemain dapat berganti peran selama pertandingan. Untuk melacak pemain dalam memainkan perannya dalam penelitian kali ini, digunakan metode *tracking* obyek banyak Kalman Filter yang digabung dengan algoritma penugasan Hungarian stage

pertama sebagai penugasan posisi pemain ukuran Q_{mea} menjadi Q_{est} dan Hungarian stage kedua sebagai penugasan posisi pemain frame referensi t terhadap frame sebelumnya ($t - 1$).

Proses *tracking* ini dimulai dari inialisasi parameter yang didasarkan pada persamaan 2.10, 2.11, 2.12, 2.13, 2.14, 2.15, 2.16, dan 2.17. Persamaan tersebut dinamakan persamaan state matriks. Referensi lokasi pemain estimasi Q_{est} pada proses tersebut di set pada koordinat (0,0). Jumlah obyek NF di set di awal sebanyak 25 obyek. Selanjutnya proses menggunakan algoritma Kalman filter yang dimulai dari menghitung lokasi perkiraan Q_{est} , dilanjutkan dengan prediksi kovarian error selanjutnya, dan perhitungan Kalman gain yang disimpan pada variabel Q_{est} . Seperti yang telah dijelaskan pada dasar teori diatas subbab 2.2.1 bahwa posisi pemain didefinisikan sebagai centroid luasan *pixel* yang terdeteksi, sehingga $Q_{est} = [centroid(:, x), centroid(:, y)]$. Apabila Q_{est} dan Q_{mea} telah diketemukan dari perhitungan, selanjutnya posisi Q_{est} yang berantakan akan diatur sedemikian rupa sehingga posisi Q_{est} sesuai dengan posisi Q_{mea} . Hal tersebut dikerjakan dengan mengaplikasikan Hungarian. Pada subbab berikut akan dijelaskan mengenai perancangan matriks state untuk membentuk Kalman Filter beserta algoritma Hungarian sebagai penugasan dari obyek referensi dan obyek acuan baik didalam proses algoritma Kalman filter ataupun pada hasil keluaran Kalman filter tersebut. Algoritma 3-2 menjelaskan secara detail *pseudocode* pembentukan algoritma Kalman Filter untuk *multi-tracking*. Dimana dalam algoritma ini dibedakan menjadi empat proses utama, yaitu pembentukan Kalman Gain, aplikasi Hungarian pada algoritma Kalman Filter tahap pertama, perbaharuan lokasi estimasi dengan matriks state, dan terakhir adalah recovery garis trayektori antara frame sebelum dan sesudah, agar kerusakan trayektori akibat oklusi dapat diperbaiki.

Inisialisasi parameter yang digunakan :	1
OUTPUT : Q_estimate(t)	2
	3
INPUT:	4
$Q = \text{zeros}(4,25);$	5
No_Sample = 2000;	6
$Q_{est} = \text{NaN}(4,\text{No_Sample});$ //membuat matriks NaN dengan ukuran	7
(4xNo_Sample)	8
$Q_{est}(:,1:\text{size}(Q,2)) = Q;$ //memasukkan nilai $Q = 0$ pada Q_{est} .	9
nF = 25; //Jumlah obyek yang ada dalam lapangan.	10
dt = 1; //Perubahan waktu	11
A,B,C // Sesuai dengan persamaan 2.10, 2.11, 2.12	12
u = 1; //Percepatan	13
Update Covariance untuk inisialisasi P sesuai dengan Persamaan 2.15	14
$P = \Sigma_x$	15
//Inisialisasi pengukuran eror sesuai dengan Persamaan 2.17	16
	17
PROSES :	18
//Menghitung lokasi perkiraan	19
$Q_{est} = A*Q_{est} + B*u;$	20
//Prediksi Kovarian perkiraan error selanjutnya	21
$P = A*P*A^T + \Sigma_x;$	22
	23
Kalman Gain	24
$K = P*C'/(C*P*C^T + Ez);$	25
//Mengambil lokasi pengukuran.	26
$Q_{mea} = [\text{centroids}(:,2) \text{centroids}(:,1)];$	27
	28
PROSES FIRST STAGE HUNGARIAN	29
//Proses ini Menugaskan lokasi Q_{loc_meas} sebagai $Q_{estimate}$.	30
$Q_{loc_meas} = \text{HUNGARIAN}(Q_{mea}, Q_{loc_meas})$	31
	32
Update Q_estimate	33
$Q_{estimate} = Q_{estimate} + K *(Q_{loc_meas}' - C * Q_{estimate});$	34
//Update the covariance estimation	35
$P = (\text{eye}(4) - K*C)*P;$	36
	37
PROSES SECOND STAGE HUNGARIAN	38
//Proses ini menugaskan $Q_{estimate}(t)$ sebagai $Q_{estimate}(t-1)$	39
$Q_{est} = \text{HUNGARIAN} (Q_{est}(t-1), Q_{est}(t))$	

Algoritma 3-2. Proses *tracking* metode usulan

3.3.1 Kalman Filter untuk *Multi-Tracking*

Diketahui bahwa lokasi para pemain disimpan pada variabel Q_{mea} . Dengan demikian apabila jumlah obyek yang diketahui adalah n maka Q_{lmea} adalah sebagai berikut.

$$Q_{mea} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \dots & \dots \\ x_n & y_n \end{bmatrix} \quad (3-9)$$

Lokasi awal (*initial location*) yang terkandung dalam Q_{lmea} berasal dari perhitungan pusat massa persamaan (3-8). Berdasarkan persamaan hukum ke-dua Newton state matrix didefinisikan seperti pada persamaan (2-10) sampai (2-14). Pertama-tama perhitungan Q_{est} sebagai inisialisasi ($Q_{est} \in \mathbb{R}^{4 \times n}$) didapatkan dari persamaan Kalman yang diambil dari elemen algoritma ke 19 adalah sebagai berikut ini.

$$Q_{est}(:, i) = A \cdot Q_{est}(:, i) + B \cdot u \quad (3-10)$$

Seperti yang dijelaskan pada dasar teori, matriks state A merupakan matriks nilai kecepatan dan percepatan suatu pemodelan matematis pergerakan pemain, dengan mengasumsikan pergerakan pemain inisial bergerak dari kondisi kecepatan nol dan percepatan nol menjadi kecepatan $v(t)$ dan percepatan $a(t)$. Selanjutnya, prediksi kesalahan estimasi Kalman didefinisikan sebagai P adalah sebagai berikut ini.

$$P = A \cdot P \cdot A^T + \Sigma_{xy} \quad (3-11)$$

Dimana Σ_{xy} didefinisikan sebagai kovarians dari pergerakan pemain pada lokasi (x, y) adalah sebagai persamaan (2-15), (2-16), dan (2-17). Persamaan Kalman Gain yang digunakan untuk menghitung Q_{est} didapatkan dari persamaan (2-18). Stage pertama Hungarian diberlakukan pada step ini, hal ini digunakan untuk memposisikan pemain Q_{est} hasil perhitungan (3-10) dengan lokasi titik pusat massa Q_{lmea} (persamaan (3-9)).

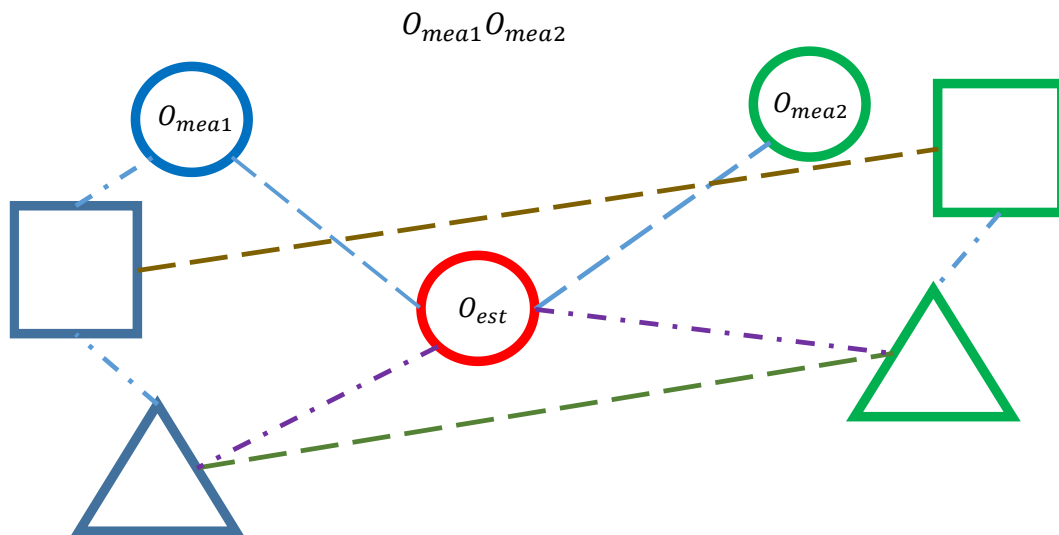
$$Q_{mea} = HUNGARIAN(Q_{mea}, Q_{lmea}) \quad (3-12)$$

Persamaan untuk menghitung Q_{est} yang diperoleh dari menambahkan perkalian Kalman Gain dengan lokasi hasil penugasan Hungarian tahap pertama yang dikurangi dengan nilai percepatan matriks state C adalah seperti yang dituliskan

pada persamaan (2-20). Selanjutnya estimasi kovarians diperbaharui seperti pada persamaan (2-19).

3.3.2 First-Stage Hungarian

Pada dasarnya permasalahan penugasan adalah bagaimana untuk membuat pemetaan maksimal pada tiap komponen observasi. Permasalahan ini merupakan permasalahan umum yang direpresentasikan dalam matriks penugasan. Penugasan berarti memberikan label pada obyek sesuai dengan bobot optimal perhitungan seleksi matriks biaya yang memuat skor obyek observasi dengan jenis pekerjaannya. Dalam tahapan observasi, posisi obyek dari Q_{est} dikenali dari pergerakan frame sebelumnya dan sesudahnya. Misalkan mengambil contoh Q_{est} dan $Q_{mea k}$ sebagai berikut ini. Jarak antara obyek pada frame sebelum O_{k-1} dan sesudah O_{k+1} didapatkan oleh jarak euclidean.



Gambar 3-9. Ilustrasi pembuatan jarak obyek referensi (merah) dengan estimasinya (biru, hijau).

Dalam penelitian ini, metode Hungarian digunakan untuk melakukan penyambungan lokasi obyek acuan pada frame k dengan lokasi obyek acuan pada frame $k - 1$ dan frame k . Sehingga obyek $O_{i,k}$ merepresentasikan identifikasi obyek dari obyek yang sama dari obyek pada frame sebelumnya ($k - 1$). Maka

matriks biaya (*cost matrix*) seperti yang dijelaskan pada persamaan (3-13) berikut ini.

$$D_{k,k-1} = \begin{bmatrix} d(11)_{k1,k-1} & d(12)_{k,k-1} & d(13)_{k,k-1} \\ d(21)_{k1,k-1} & d(22)_{k,k-1} & d(23)_{k,k-1} \\ d(31)_{k1,k-1} & d(32)_{k,k-1} & d(33)_{k,k-1} \end{bmatrix} \quad (3-13)$$

Dengan : $d(ij)_{k,k-1} = \text{jarak ekluidean}(O_{i,k-1}, O_{i,k})$

Hungarian akan mencari pemetaan obyek O_i dengan menemukan solusi nilai minimum pada matriks biaya D . Sehingga solusi dari permasalahan diatas adalah:

$$f = \text{Min}(\{D_{k,k-1}\}) \quad (3-14)$$

$$f: \{(O_{1,k-1} \rightarrow O_{1,k}), (O_{1,k-1} \rightarrow O_{3,k})\} | O_{i,k}, O_{i,k-1} \in R^2$$

f memiliki fungsi minimum D , jika dan hanya jika obyek O_{k-1} pada frame sebelumnya dikenali sebagai obyek pada frame saat ini $O_{i,k}$, dan obyek hasil pengenalan saat ini dikenali sebagai obyek frame berikutnya O_{k+1} . Dimana dimensi semua obyek yang terkenal sama yaitu berada pada dua dimensi $R^2(x, y)$.

3.3.3 Normalisasi Trayektori

Proses normalisasi trayektori untuk meniadakan jarak ekstrim posisi awal adalah suatu proses yang harus dilakukan ketika hasil trayektori menjadi tak beraturan ketika obyek tidak nampak (*intermitten*). Maka kesesuaian trayektori pada frame sebelum dan sesudah haruslah memiliki jarak tertentu, apabila jarak trayektori antara frame sebelum dan sesudah melebihi dari jarak yang diwajibkan akan menghasilkan situasi lecutan secara visual. Karena pada hakekatnya algoritma Kalman Filter menganggap segala sesuatu bergerak, apabila sesuatu tersebut diam algoritma Kalman Filter akan memprediksi bergerak terlebih dahulu dengan berdasar inisialisasi kecepatan dan percepatan matrik statenya baru kemudian mengkoreksi perubahan lokasi obyek pada frame berikutnya berdasarkan informasi lokasi frame sebelumnya.

Di dalam normalisasi trayektori juga menghadapi beberapa rencana apabila kondisi obyek tidak dapat terkenal oleh karena *intermitten*. Kondisi *intermitten* diakibatkan oleh ketika obyek dinyatakan hilang oleh penugasan Hungarian. Hilang deteksi diakibatkan oleh tiga hal, yaitu pertama ketika obyek sebgai besar berhimpitan dengan obyek lain, kedua ketika obyek keluar dari

lapangan, dan ketiga hasil dari filter morfologi malah mengerosi luasan obyek sehingga obyek dianggap noise lalu difilter. Oleh karena nF di set 25 maka keseluruhan obyek berjumlah 25 dan apabila terdapat deteksi baru maka detektor memposisikan kandidat deteksi tersebut sebagai salah satu dari ke-25 obyek.

3.3.4 Second-Stage Hungarian

Pada proses ini dijelaskan bagaimana proses pemulihan trayektori dilakukan.

Algoritma 3-3 memperlihatkan langkah-langkah pemulihan garis trayektori.

<i>Input</i> : $Q_{est} = [x, y]$	1
<i>Output</i> = \widehat{D}_Q Matriks output yang sudah mengalami penugasan ulang.	2
	3
	4
<i>Proses</i> :	5
$D_Q = euclid(Q_{est\ t}, Q_{est\ (t-1)})$	6
$if(D_{Qi} > T_{DQ}), \{Q_{est} = NaN\}$	7
$else \{$	8
$\widehat{D}_Q = Hungarian \{D_Q\}$	9
<i>End</i>	

Algoritma 3-3. Second-Stage Hungarian

Proses ini dimulai dari variabel lokasi estimasi Q_{est} yang berisi lokasi obyek dalam frame. Algoritma ini memerlukan dua buah inputan Q_{est} dari frame waktu sebelum $(t - 1)$ dan frame saat ini t . Perbedaan antara nilai obyek pada $Q_{est(t-1)}$ dan $Q_{est\ t}$ dihitung dengan menggunakan jarak euclidean D_Q . Perhitungan jarak tersebut dikerjakan pada semua obyek yang terkenal $Q_{est} = [q_1 q_2 q_3 \dots q_N]$. Proses perhitungan jarak D_Q dihitung sesuai dengan label obyek yang didapatkan dari proses *tracking* sebelumnya. Selanjutnya, apabila proses pembentukan D_Q selesai, tiap-tiap hasil jarak euclidean $euclid(.)$ dibandingkan dengan *threshold*. Nilai *threshold* ini didapatkan dari hasil percobaan. Nilai *threshold* ini berfungsi untuk memberikan penegasan kepada obyek supaya dalam frame selanjutnya apakah obyek tersebut layak untuk diberikan label sebagai obyek atau tidak.

Apabila diketahui terdapat elemen D_Q yang nilainya lebih dari $threshold T_{DQ}$ maka elemen D_{Qi} akan diberikan nilai NaN , sebaliknya nilai D_{Qi} akan dilewati. Proses terakhir pada sesi ini adalah mengaplikasikan algoritma Hungarian untuk menugaskan kembali, atau secara visual adalah menyambung patahan yang putus akibat dari proses *threshold* diatas. Contoh aplikasi proses ini dapat dilihat pada Gambar 4-16 (b) dan (c). Dimana gambar Gambar 4-16 (b) merupakan hasil proses *threshold* proses (a) dan (c) adalah hasil penyambungan patahan yang diakibatkan oleh (b).

Dalam penelitiannya, penulis menggunakan dua metode penugasan Hungarian. Berikut adalah penjelasan mengenai posisi algoritma Hungarian terhadap fungsinya dalam Kalman filter dan sebagai alat untuk memulihkan garis yang telah terputus. Pada hakekatnya algoritma Kalman filter mempekerjakan algoritma Hungarian stage pertama sebagai bentuk alat untuk memberikan label obyek sesuai dengan jarak kedekatan antara obyek pada obyek estimasi Q_{est} dan obyek ukur (referensi) Q_{mea} . Sedangkan algoritma Hungarian stage kedua berfungsi sebagai pemberi label obyek sesuai dengan jarak kedekatan antara $Q_{est(t-1)}$ dan $Q_{est t}$. Dengan mengasumsikan bahwa *cost matrix* dari matriks state posisi frame sebelum dan sesudah telah diperoleh dengan menggunakan jarak Euclidean, maka sekarang didapatkan matriks $M_{Cost} \mathbb{R}^{n \times n}$ dengan nilai $n = 25$ pemain. Matriks ini akan digunakan sebagai masukan pada algoritma Hungarian. Seperti yang telah diketahui bersama dalam subbab 2.2.6 penugasan suatu obyek merupakan obyek yang dikehendaki mengacu pada seberapa dekat obyek referensi tersebut dengan obyek yang dicari. Maka, dapat kita simpulkan bahwa disini tidak ada pelabelan khusus yang digunakan untuk menugaskan suatu obyek selain berdasarkan jarak *cost matrix* terdekat. Dengan mengasumsikan bahwa suatu pergerakan obyek tersebut memiliki jarak minimal \mathcal{E} terhadap obyek referensi pada frame sebelumnya maka penugasan Hungarian tersebut digunakan. Walaupun demikian suatu asumsi batasan akan dapat terjadi kesalahan deteksi apabila jarak minimal \mathcal{E} ternyata dimiliki oleh obyek yang lain yang melintasi daerah lintasan obyek referensi, hal ini akan menimbulkan kesalahan yang tak terelakkan. Hasil analisa dan penjelasan terhadap aplikasi sistem dapat dilihat pada subbab 4.4.

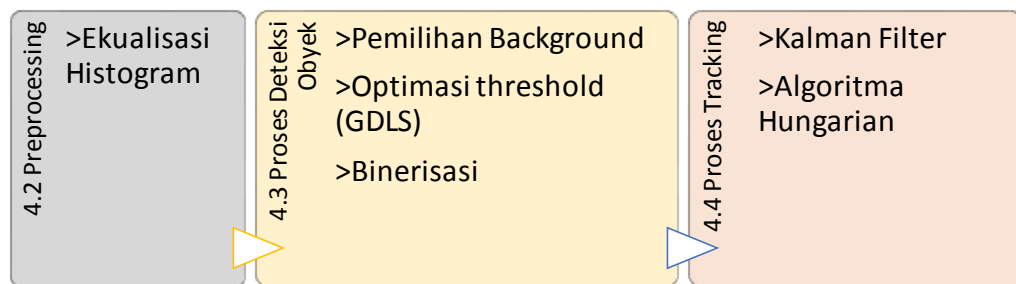
Halaman ini sengaja dikosongkan

BAB 4

HASIL DAN PEMBAHASAN

Dari Gambar 3-1 didapatkan diagram proses perancangan sistem sebagai berikut ini (Gambar 4-1). Dari percobaan yang dilakukan, sub-bab

Preprocessing, Proses Deteksi Obyek, dan Proses *Tracking*, menjelaskan hasil-hasil yang telah didapatkan dari percobaan.



Gambar 4-1. Diagram proses perancangan sistem.

Berikut akan dijelaskan mengenai hasil percobaan yang menggunakan data uji coba pada subbab 4.1, adapun Dataset [7] uji coba terbagi dua hal yaitu: Dataset video pertandingan setengah babak sepakbola (sekitar 27 menit) dan *Ground Truth*, dimana hal ini digunakan sebagai validator pengenalan obyek setelah proses deteksi obyek (*background subtraction*) dan *tracking* dilakukan.

4.1 Dataset Uji Coba

Data diambil dari hasil rekaman video yang direkam menggunakan kamera dengan spesifikasi tinggi (detail karakteristik kamera dapat dilihat pada Tabel 3-1) hasil pengambilan tersebut dipisahkan menjadi frame-per-frame oleh [6] sehingga menjadi dataset sebanyak 26.222 frame. Data memiliki karakteristik seperti yang ditunjukkan pada Tabel 3-3 dan ditunjukkan secara visual pada Gambar 4-2, yaitu disana diperlihatkan mengenai karakteristik dataset yaitu: Pertama, terdapat dimensi fisis data seperti: *Background* yang tak berubah (statis: diambil dari kamera stasioner). Kedua, terdapat suatu perubahan iluminasi. Misalnya dalam Gambar 4-2 diketahui terjadi degradasi warna level gelap-terang (*brightness*) yang direpresentasikan oleh degradasi skala keabuan/*grayscale* pada

pojok kanan dan kiri bawah menuju ke atas. Ketiga, terdapat *foreground aperture*, yaitu suatu kondisi rekaman data obyek aktif bergerak dari awal perekaman. Disamping ada beberapa obyek yang aktif bergerak, ada juga obyek pasif dari awal perekaman dan baru bergerak setelah beberapa saat kemudian. Selain itu, terdapat bayangan di obyek dalam setiap pergerakannya. Berdasarkan karakteristik ini dapat disimpulkan bahwa, pergerakan obyek dalam dataset ini dikategorikan sebagai pergerakan random. Dalam dataset tersebut muncul bayangan *penumbra* yang menempel pada pemain, bayangan tersebut diakibatkan oleh lampu-lampu sorot lapangan yang menyorot lapangan dari berbagai arah (terlihat dalam contoh Gambar 4-2). Penumbra tersebut seakan-akan berbentuk tanda silang dibawah kaki pemain. Bayangan ini harus dihilangkan karena bayangan ini membuat pemain satu dan lainnya mengalami oklusi dan perhitungan titik pusat massa lokasi pemain menjadi bergeser sehingga kemungkinan kondisi detektor mengalami *False Positive* tinggi.



Gambar 4-2. Pada frame 525 menunjukkan pemain memiliki bayangan

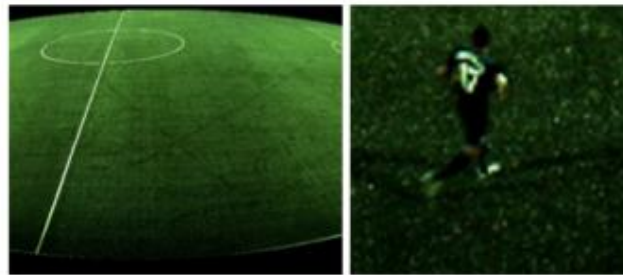
Disamping tantangan *penumbra* yang dihasilkan dari sistem pencahayaan, dalam penelitian ini terdapat pula tantangan dimana jersey pemain memiliki warna yang sama dengan area gelap dalam lapangan, sehingga hal tersebut mempersulit pengenalan berdasarkan warna. Contoh kasus tersebut dapat terlihat dalam Gambar

4-3. Dari gambar tersebut terlihat bahwa gambar memiliki tantangan pengenalan dari seragam pemain (*jersey*), yaitu terdapat salah satu tim dengan warna seragam hitam, dimana pada saat pemain berseragam hitam berada di daerah paling gelap (yaitu pinggir lapangan) karena distribusi cahaya tidak banyak, maka pemain tersebut memiliki karakteristik *pixel* RGB sama dengan *pixel background*. Akibatnya, detektor seringkali tidak dapat mengenali obyek saat kondisi tersebut terpenuhi.

Dalam penelitian ini, peneliti akan memperkenalkan sistem pendeteksi pemain yang *invariant* terhadap tantangan level terang-gelap dan bayangan *penumbra*, hal tersebut akan dijelaskan lebih lanjut pada subbab 4.3 dan 4.4. Selain itu, berdasarkan kegunaan sistem, suatu sistem *tracking* yang tepat guna hendaknya dapat melakukan pelacakan di dalam kondisi apapun dengan pergerakan pemain random. Percobaan yang dilakukan peneliti kali ini membuktikan bahwa keefektifan sistem deteksi dan *tracking* dalam melakukan tugasnya. Berikut adalah penjelasan dari analisa hasil penelitian.

Berdasarkan validasi ground truth dataset, total pemain dalam satu tim ada 22 pemain, akan tetapi dalam kenyataannya terdapat dua hakim garis, satu wasit, dan satu bola yang selalu bergerak. Berdasarkan tutorial dataset [7] pergerakan yang paling sulit untuk dilacak adalah pergerakan bola, hal ini dikarenakan bentuk bola yang kecil, pergerakannya cepat sehingga rawan terjadi oklusi, dan sering terjadi *intermitten* karena bola sering menjadi pusat perhatian dan diperebutkan oleh pemain. Untuk menyelesaikan hal-hal tersebut maka, penulis memilih metode deteksi obyek ini berdasarkan *background subtraction*. Peneliti mengakomodir semua obyek yang bergerak yang terekam dalam kamera dan berada pada lokasi daerah ROI (*Region of Interest*). Definisi ROI adalah suatu daerah di dalam lapangan, yaitu lapangan yang didefinisikan pada Gambar 3-2.

Dalam subbab 4.3 dan 4.4 akan menjelaskan secara lebih lanjut mengenai hasil yang diperoleh sistem peneliti. Seperti yang dijelaskan diatas jumlah yang selalu bergerak ada 25 obyek. Dalam penelitian ini dataset memiliki *ground truth* untuk deteksi obyek, secara visual hal ini ditunjukkan pada Gambar 4-4. *Ground truth* ini digunakan untuk melakukan validasi atas proses deteksi obyek dan sebagai inputan perhitungan training pada algoritma *GDLS*.



(a)

(b)

Gambar 4-3. Diambil dari frame pertama yang merupakan contoh: (a) Degradasi warna lapangan oleh karena distribusi pencahayaan tidak merata dan (b) warna seragam pemain memiliki kemiripan dengan *background*



(a)



(b)

Gambar 4-4. Frame ke-25 (a) *ground truth* (b) Gambar input

Hasil dari deteksi obyek kemudian divalidasi untuk mengukur akurasi proses deteksi. Adapun, cara proses validasinya adalah melakukan operasi masking pada citra hasil dan citra *ground truth* sehingga didapatkan hasil pencocokan tersebut, apabila hasil pencocokan tersebut cocok artinya keseluruhan *pixel* hasil proses metode usulan menempati area obyek (*masking*) *ground truth* maka di dalam tabel lokasi *pixel* dihitung 1, bila tidak dihitung 0. Jumlah nilai 1 di area obyek di total dan dibandingkan dengan jumlah seluruh area *ground truth* yang memiliki nilai 1. Hasil dari perhitungan tersebut dinamakan *True Positive*. Dengan kata lain, Tabel 4-1 yang menjelaskan bahwa true positive adalah nilai *pixel* obyek terdeteksi sebagai benar obyek adalah tepat. *True Negative* diartikan sebagai *pixel* obyek yang terdeteksi sebagai *background*, atau secara matematis dapat dijelaskan bahwa semua nilai piksel semesta S dikurangi dengan nilai piksel obyek yang telah terdeteksi benar sebagai obyek ($O \wedge \Omega$).

Tabel 4-1. Tabel penjelasan variabel confusion matriks

Nama Variabel	Penjelasan
<i>True Positive (TP)</i>	<i>Pixel</i> obyek (pemain) terdeteksi sebagai <i>foreground</i>
<i>True Negative (TN)</i>	<i>Pixel</i> obyek (pemain) terdeteksi sebagai <i>background</i>
<i>False Positive (FP)</i>	<i>Pixel</i> bukan obyek terdeteksi sebagai <i>foreground</i>
<i>False Negative (FN)</i>	<i>Pixel</i> bukan obyek terdeteksi sebagai <i>background</i>

False Positive diartikan sebagai *pixel* bukan obyek terdeteksi sebagai *foreground*, atau diartikan sebagai semua *pixel* yang seharusnya *background* yang salah dikenali sebagai *foreground*. Sementara itu *False Negative* diartikan sebagai *pixel* bukan obyek betul dikenali sebagai *background*.

Berikut (4-1), (4-2), (4-3) dan (4-4) adalah persamaan yang digunakan dalam Tabel 4-1. Tabel penjelasan variabel confusion matriks. Hasil perhitungan nyata persamaan tersebut dapat diketahui pada Tabel 4-4. Dengan X adalah *pixel* obyek, Ω adalah *pixel ground truth*, dan S adalah semesta.

- TP: *Pixel* obyek (pemain) terdeteksi sebagai *foreground*.

$$TP_N = \sum_{i=1}^N (X_i \wedge \Omega_i) \quad (4-1)$$

- FP: *Pixel* bukan obyek (pemain) terdeteksi sebagai *foreground*.

$$FP_N = \sum_{i=1}^N |(X_i \wedge \Omega_i) - X_i| \quad (4-2)$$

- TN: *Pixel* obyek terdeteksi sebagai *background*.

$$TN_N = \sum_{i=1}^N |S_i - (X_i \vee \Omega_i)| \quad (4-3)$$

- FN: *Pixel* bukan obyek terdeteksi sebagai *background*.

$$FN_N = \sum_{i=1}^N |(X_i \wedge \Omega_i) - \Omega_i| \quad (4-4)$$

4.2 Preprocessing

Dalam preprocessing terdapat proses ekualisasi histogram, hasil ekualisasi histogram dapat dilihat pada Gambar 4-5. Ekualisasi histogram berdasarkan pada asumsi bahwa persebaran data pada skala keabuan memiliki nilai bukan nol yang merata. Dengan kata lain, apabila pada hakikatnya secara fisis gambar memang cenderung condong menuju pada titik sebaran tertentu ($skewness > \text{or} < 0$) maka proses ekualisasi histogram ini tidak terlalu memiliki hasil yang signifikan. Peneliti mengungkapkan bahwa *Background* didefinisikan sebagai suatu kondisi *pixel* yang tidak mengalami perubahan nilai dari frame awal sampai frame akhir, secara visual dapat dikatakan bahwa citra yang disebut *background* adalah citra lapangan dengan garisnya. *Foreground* didefinisikan sebagai suatu *pixel* yang mengalami perubahan dari waktu frame awal sampai frame akhir, secara visual obyek yang merupakan *foreground* adalah pemain sepak bola, wasit dan bola.



Gambar 4-5. (a) Hasil sebelum dan (b) Hasil sesudah proses ekualisasi histogram

Berdasarkan hasil penelitian, bahwa dataset merupakan dataset yang diambil pada malam hari dan sistem pencahayaan menggunakan lampu sorot, sehingga dapat dipastikan perubahan iluminasi adalah statis, oleh sebab itu nilai iluminasi dinamis dapat diabaikan. Oleh karena itu pembaharuan *background* sintetis dapat dilakukan tanpa memasukkan asumsi perubahan iluminasi dinamis. Dari Gambar 4-5 diatas dapat dilihat bahwa setelah citra dinormalisasi citra menjadi lebih cerah. Tujuan preprocessing ini adalah menstandarisasi inputan yang masuk ke dalam proses sehingga nilai *pixel* dapat terdistribusi secara normal. Distribusi normal didefinisikan sebagai distribusi *Gaussian*. Walaupun demikian, dapat dilihat pada Gambar 4-5 (b) masih terdapat degradasi terang-gelap yang terjadi. Deteksi pemain

dengan menggunakan proses usulan terbukti dapat melakukan deteksi pemain walaupun dibatasi oleh masalah ketidak merataan distribusi cahaya. Tabel 4-2 dapat memperlihatkan bagaimana hasil perawatan *background*. *Background* sintetis ini dihasilkan dari perhitungan proses statistik yaitu rata-rata dari N frame. *Background* sintetis memiliki keluaran yang dapat memodelkan lapangan dengan distribusi iluminasi yang sesuai dengan sistem pencahayaan lapangan dengan tingkat degradasi tingkat terang-gelap yang mirip dengan lapangan real. Selanjutnya, pada subbab 4.3 akan menjelaskan mengenai proses-proses deteksi obyek, yang dimulai dari pemilihan *background*, pemilihan *threshold* guna pembentukan citra biner untuk masking, dan perbandingan metode lain guna membuktikan tingkat kesuksesan metode usulan dengan metode yang telah ada.

4.3 Proses Deteksi Obyek

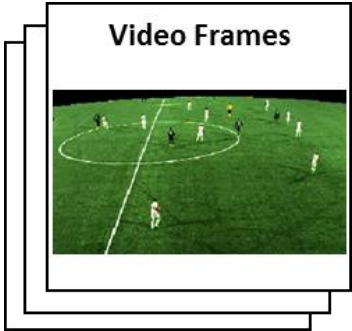

Hasil proses deteksi obyek dikelompokkan menjadi tiga yaitu hasil pada pemilihan proses *background*, optimasi *threshold* dengan menggunakan GDLS, dan binerisasi. Seperti yang telah dijelaskan pada subbab sebelumnya, bahwa penulis telah membandingkan metode usulan dengan metode lain yang sejenis guna mencari perbandingan tingkat keberhasilan metode usulan dengan metode lain. Metode-metode pembanding diperoleh dari rekomendasi [5], dengan mengacu pada kesamaan konsep penelitian pada pemodelan *background* menggunakan perhitungan statistik.

4.3.1 Pemilihan *Background Sintesis*

Dalam penelitian ini penulis menggunakan proses pemilihan *background* didapatkan dari rata-rata persamaan (3-5) Dari hasil percobaan yang dilakukan ditunjukkan pada Tabel 4-2. Proses ini dipilih karena obyek harus dapat dikenali walaupun dalam kondisi obyek tidak bergerak. Metode deteksi obyek yang berdasarkan pergerakan obyek tidak dipilih oleh karena dapat memungkinkan terjadinya kehilangan deteksi saat obyek tidak bergerak. Oleh karena penulis dalam penelitiannya menggunakan dasar *background subtraction*. Yang mana *background subtraction* adalah proses pemisahan *background* dan *foreground*.

Dalam penelitian ini, penulis menggunakan metode estimasi *background* untuk mendapatkan *foreground* daripada sebaliknya. Hal ini dikerjakan olehkarena, proses estimasi *background* atau sering disebut sebagai *background sintesis* memiliki tingkat komputasi yang rendah, karena hanya memerlukan perhitungan matematis matriks rata-rata dengan pengaturan kapan proses ini dilakukan dan berapa jumlah frame yang akan dimasukkan kedalam formula rata-rata.

Tabel 4-2. Hasil pemilihan background

Nama Variabel	Hasil dan Deskripsi
$I_t(x, y)$	<p data-bbox="582 779 1082 817">Input variabel citra dari tiap-tiap frame</p> 
N	<p data-bbox="389 1218 1157 1294">Jumlah Total frame yang dimasukkan dalam persamaan 3.5 =1000</p>
$\widehat{\Phi}_t(x, y)$	<p data-bbox="389 1294 1050 1332">Hasil tahap perawatan <i>background</i> pada $N = 1000$</p> 

Selanjutnya akan dijelaskan mengenai pembuatan *Background* sintesis. Setelah dihasilkan *background sintesis* akan selalu diperbaharui setiap 50 frame untuk menjaga ketahanan *background* dari perubahan iluminasi. Menurut hasil penelitian yang dilakukan perubahan setiap 50 frame sudah cukup untuk menghasilkan *background* sintesis yang terbaik, yakni mampu menghasilkan nilai score F1

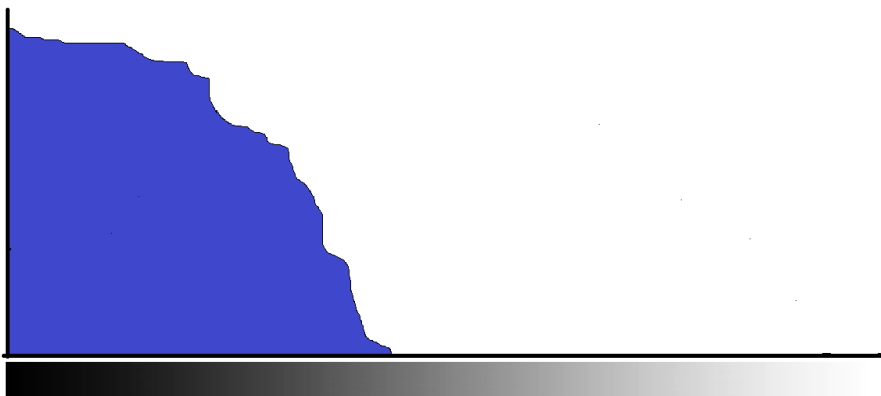
optimal. Dari tabel ini didapatkan hasil perawatan *background* $\hat{\phi}$ seperti yang tertampil pada baris ke tiga di kolom nilai variabel. Dapat dilihat dari tabel tersebut, bahwa hasil rata-rata ϕ memiliki bentuk *background* tanpa obyek yang mirip dengan citra asli *background*. Pemodelan *background* ini hanya merepresentasikan nilai *background* tanpa *foreground* (obyek). Perbaikan *background* (*background maintenance*) diartikan sebagai update variabel ϕ selama panjang waktu urutan (*sequence*) frame analisa. Pemodelan $\hat{\phi}$ tersebut memuat keseluruhan degradasi terang-gelap akibat distribusi cahaya yang tidak merata, sehingga diharapkan ketika proses subtraksi dilakukan nilai *pixel* yang berkorelasi terhadap *background* pada citra uji dapat dihilangkan seluruhnya. Walaupun demikian hasil ϕ tetap saja memiliki degradasi iluminasi, hal ini dapat terlihat pada distribusi terang-gelap pada Tabel 4-2 pada baris ketiga (ϕ). Adapun Gambar 4-6, Gambar 4-7, dan Gambar 4-8 adalah gambar dari histogram *background* sintesis Tabel 4-2 diatas. Dari gambar tersebut dapat terlihat dengan jelas bahwa Gambar 4-7 yang merupakan komponen warna hijau memiliki rentang persebaran yang lebih panjang dibanding dengan Gambar 4-6 yaitu komponen warna merah dan Gambar 4-8 komponen warna biru. Hal ini dikarenakan mayoritas gambar adalah warna hijau, yaitu rumput lapangan. Dari Gambar 4-7 juga dapat diketahui terdapat beberapa nilai *pixel* hijau yaitu dari rentan nilai pixel 50 sampai 0 yang memiliki nilai rata-rata jumlah *pixel* sebanyak 2000 buah. Komponen biru yang terdapat pada Gambar 4-8 terlihat memiliki kecenderungan menuju pada sisi kiri dimana sisi ini merupakan sisi nilai tingkat keabuan rendah sehingga representasi warna kebiruan rendah yaitu sekitar kurang dari 100 desimal. Sementara itu total pixel dalam satu frame adalah 8,9 juta *pixel* atau dengan kata lain terdapat peluang kemunculan *pixel* mengalami degradasi terang-gelap sebesar $2,24 \times 10^{-4}$. Nilai kecerahan pada sisi kanan bawah tidak ada yang bernilai lebih besar daripada 170 desimal. Hal ini terjadi karena distribusi warna hijau yang merata pada lapangan sepak bola, maka warna hijau tersebut digunakan sebagai obyek untuk melakukan *background subtraction* dengan cara mengenali tiap-tiap pixel apakah sebagai *pixel* berwarna hijau. Pengenalan berdasarkan warna seringkali mengalami permasalahan apabila kostum pertandingan memiliki warna yang sama terhadap lapangan.



Gambar 4-6. Histogram Komponen Merah Background Sintetis



Gambar 4-7. Histogram Komponen Hijau Background Sintetis











Gambar 4-8. Histogram Komponen Biru Background Sintetis

Komponen warna RGB rentan terhadap perubahan iluminasi, karena komponen warna ini mengisyaratkan tingkat level keabuan tiga komponen warna R, G, dan B, sehingga bila terdapat perubahan pencahayaan walaupun satu obyek yang sama nilai *pixel* komponen referensi akan mengalami perubahan drastis. Oleh sebab itu, perubahan level menuju komponen warna yang tidak terpengaruh terhadap

pencapaian (*lightness invariant*), dalam penelitian kali ini penulis menggunakan komponen warna HSV. Komponen warna HSV dipilih karena komponen ini memiliki kemampuan untuk mengkodekan level kromatisitas warna piksel, dengan kata lain secara intuitif penggunaan elemen luminasi (*brightness*) komponen warna terpisah dari nilai komponen piksel itu sendiri yaitu yang disimbolkan dengan Value. Tabel 4-3 dicuplik beberapa sampel bagian obyek dimana jenisnya adalah sebagai berikut ini: baris satu sampai baris enam adalah contoh degradasi warna lapangan, sementara itu baris selanjutnya sampai terakhir adalah contoh obyek yang terambil. Dari tabel tersebut dapat dilihat bahwa nilai komponen HSV memiliki nilai yang relatif stabil daripada RGB yang memiliki jarak perbedaan nilai *pixel* antara satu *pixel* dengan *pixel* lainnya. Selanjutnya pada subbab 4.3.2 akan membahas mengenai penentuan *threshold* hasil dari diferensiasi (*background subtraction*) yang akan digunakan untuk proses binerisasi. Dari Tabel 4-3 disampling 13 citra yang merepresentasikan keseluruhan warna dari obyek dan lapangannya. Sampling tersebut dibagi menjadi tiga besaran sampling utama, yaitu sampling lapangan (*sample field*), sampling pemain (*sample player*) tim A dan B, dan sampling wasit (*sample referee*). Analisa komponen warna ini pada pembuatan filter operasi biner akan sangat berguna sekali dalam memilih operasi aritmatika biner. Filter ini digunakan untuk memfilter bayangan yang menempel pada pemain. Selain bayangan yang menempel, berlandaskan analisa pada tabel Tabel 4-3 ini juga dapat digunakan sebagai detektor garis, dimana pada proses *background subtraction* garis lapangan tidak terfilter dengan baik, sehingga masih menyisakan sisa. Penjelasan proses ini dapat dilihat kembali pada Gambar 3-6, dimana diketahui terdapat tiga *image difference* berdasar komponen warna HSV. Proses penentuan *Threshold* adalah proses yang sangat penting sebelum melakukan pemfilteran terhadap bayangan dan garis lapangan, hal ini dikarenakan hasil dari proses pengurangan *background* sintesis dengan frame yang terjadi mengakibatkan nilai-nilai pixel bervariasi dan terdapat nilai negative. Proses *threshold* secara langsung akan mengakibatkan obyek akan hilang dan diklasifikasikan sebagai *background* sementara obyek dalam penelitian ini dihakimi sebagai *foreground*. Oleh karena ketidaksesuaian nilai negative hasil dari *background subtraction* muncul, maka

penulis menggunakan harga mutlak $|\cdot|$ untuk menaikkan atau mempositifkan nilai *background subtraction*

Tabel 4-3. Perbandingan Warna RGB dan HSV dalam lapangan sepakbola

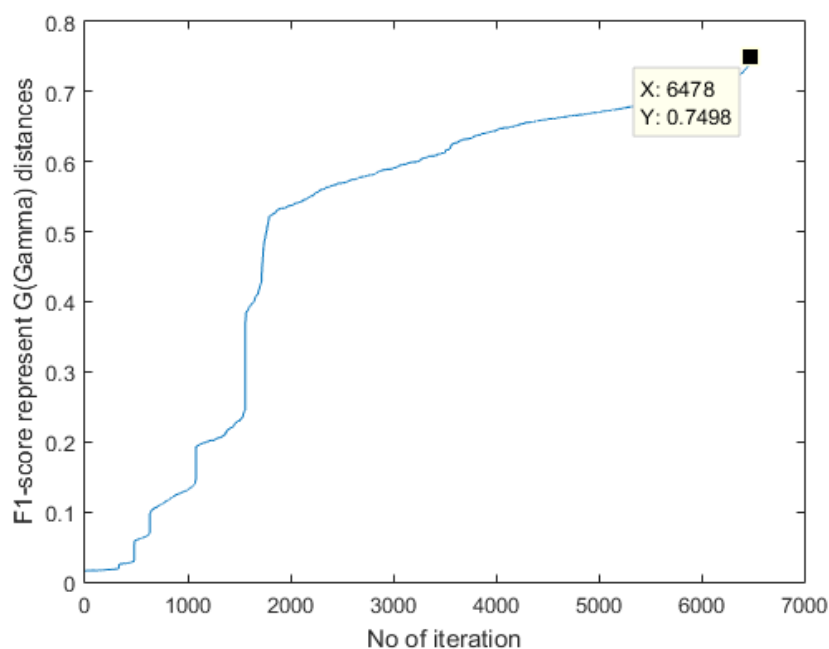
Name	RGB			HSV		
	R value	G value	B value	H value	S value	V value
 <i>Sample Field-1</i>	58	100	34	0,2833	0,5882	0,2667
 <i>Sample Field-2</i>	81	131	46	0,2743	0,267	0,494
 <i>Sample Field-3</i>	68	117	38	0,2786	0,6907	0,3804
 <i>Sample Field-4</i>	11	31	4	0,2901	0,6585	0,1608
 <i>Sample Field-5</i>	225	240	219	0,2722	0,1345	0,8745
 <i>Sample Field-6</i>	79	141	58	0,2798	0,6231	0,5098
 <i>Sample Player Black-1</i>	0	18	2	0,3205	0,52	0,1961
 <i>Sample Player Black-2</i>	0	2	7	0,2576	0,1341	0,6431
 <i>Sample Player Black-3</i>	6	31	0	0,187	0,1394	0,9843
 <i>Sample Referee-1</i>	224	241	129	0,1777	0,533	0,8902
 <i>Sample Player White-1</i>	159	164	132	0,1667	0,1098	1
 <i>Sample Player White-2</i>	220	240	189	0,2167	0,1688	0,9294
 <i>Sample Player White-3</i>	194	210	165	0,2259	0,2296	0,7686

sedemikian sehingga proses optimasi dapat bekerja dengan optimal. Oleh sebab itu, nilai harga mutlak yang dihasilkan membuat suatu fungsi menjadi fungsi yang selalu positif, dengan demikian kita dapat menginterpretasikan bahwa nilai suatu

fungsi positif tersebut sebagai hasil dari fungsi konveks (Detail dari penggunaan optimasi fungsi konveks dapat dilihat kembali pada Bab 3). Jadi, algoritma yang diberlakukan sebagai pemroses berdasarkan karakteristik fungsi konveks dapat dipekerjakan, adapun algoritma ini penulis memilih *Gradient Descent Line Search* (GDLS).

4.3.2 Optimasi *Threshold* Menggunakan Algoritma GDLS

Algoritma ini bekerja dalam tiap frame dengan melakukan iterasi sebanyak $6^3=216$ kali (*threshold* $\Gamma = \gamma_1, \gamma_2, \gamma_3$ masing-masing sebesar: $0 \leq \gamma \leq 5$ dengan step sebesar: 0.). Bila total 26.222 frame maka algoritma ini akan mempunyai beban iterasi sebanyak 216 kali x 26.222 frame yaitu sebanyak 5.663.952 kali. Berikut adalah grafik optimasi dari algoritma GDLS. Gambar 4-9 menunjukkan nilai maksimum F1-score sebesar 0,7498 pada 30 frame uji. Dimana nilai ini berkorelasi negatif dengan jarak $G(\Gamma)$.



Gambar 4-9. Grafik nilai F1 lawan jumlah iterasi

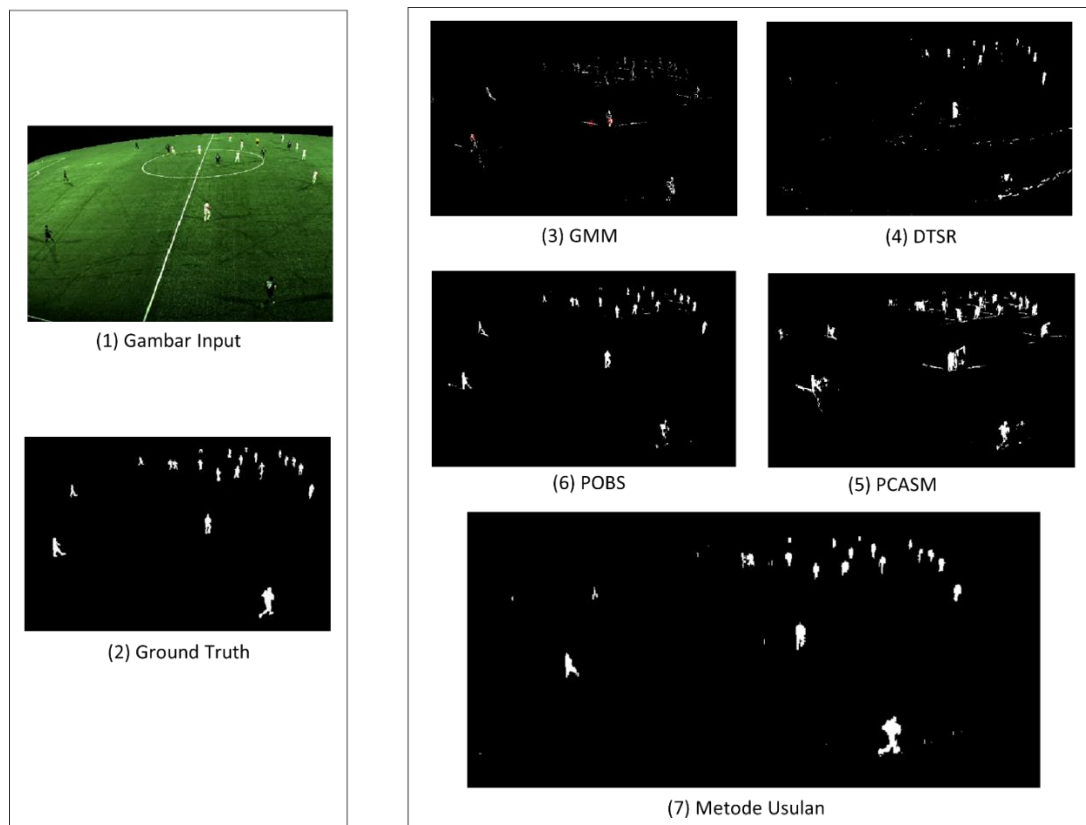
Melalui percobaan ini, didapatkan, $G(\Gamma)$ optimum pada komponen warna $H=0,1$, $S=0,5$, dan $V=0,1$. Oleh karena itu dipilih nilai $\Gamma = [0,1 \ 0,5 \ 0,1]$ sebagai *threshold*. Dari gambar tersebut juga dapat kita melihat pada iterasi ke 2.000 merupakan titik

belok, dimana sebelum iterasi ke 2.000 nilai vertikal F1 berada pada 0,5 secara signifikan naik kemudian mengalami kondisi steady konvergen pada iterasi > 2.000, dengan titik puncak nilai konvergensi pada iterasi ke 5.478. Pada penelitian ini, penulis mengaplikasikan GDLS hanya satu kali pada saat inisialisasi awal. Hal ini dikarenakan, dataset yang diambil tidak dinamis sehingga asumsi *background* dinamis dapat diabaikan. Berdasarkan beberapa asumsi yang telah dilakukan, maka dapat disimpulkan bahwa, sistem yang didesain oleh penulis hanya mampu menangani citra uji yang sesuai dengan dataset [35]. Adapun, berdasarkan beberapa penelitian yang telah dilakukan oleh penulis, citra *training* pada sistem pembelajaran terpimpin perlu dilakukan supaya sistem dapat belajar untuk mengoptimasi dirinya sendiri sehingga apapun input yang terjadi dapat dilakukan pengenalan. Setelah melakukan aplikasi algoritma optimasi, subbab 4.3.3 merupakan proses yang ditujukan untuk membandingkan metode usulan dengan metode perbandingan.

4.3.3 Perbandingan dengan Metode Deteksi Obyek Lain

Metode usulan dibandingkan dengan empat metode lain berdasarkan rekomendasi dari Bouwmans [5]. Pemilihan metode perbandingan tersebut, sama-sama merupakan golongan *Background subtraction* yang menggunakan dasar metode statistik, sehingga dapat dikatakan bahwa kemampuan empat metode perbandingan dengan usulan sebanding dalam penggunaan metode penyelesaiannya. Walaupun metode berdasar PCA merupakan metode transformasi vector kedalam kelasnya, dan metode GMM merupakan metode *probabilistic* dengan menggunakan distribusi gabungan dari nilai piksel yang dipandang sebagai variabel random, sementara itu DTSR menggunakan metode pemodelan median terhadap *background* dengan penambahan filter *Otsu* untuk penghilang bayangannya. Empat metode perbandingnya yaitu: Teknik PCA dan model penghalus spasial (PCASM) yang dibuat oleh Xue dkk [37], Gaussian Mixture Model (GMM) dibuat oleh Yao dkk [38] dan Kaewtrakulpong dkk [39], *Threshold* dinamis untuk penghilang bayangan (DTSR) oleh Taha dkk [40] dan operasi point subtraksi *background* (POBS). POBS adalah metode usulan tanpa operasi filter spasial bitwise. Khusus untuk pengujian metode usulan ini supaya menjaga

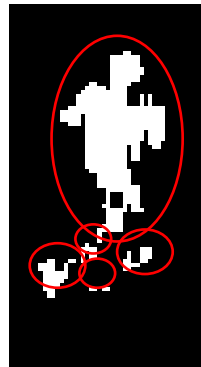
komprensivitasnya, peneliti menempatkan POBS sebagai metode lawan dengan tujuan mengetahui seberapa besar perbaikan metode usulan dari POBS tersebut.



Gambar 4-10. Gambar hasil deteksi metode usulan dibandingkan dengan metode-metode perbandingan. (1) Gambar input, (2) Ground Truth, Gambar (3) GMM, (5) PCASM memiliki karakteristik rentan terhadap pengenalan bayangan. (4) DTSR dan (6) POBS tahan terhadap bayangan akan tetapi hasilnya terdapat noise. (7) adalah metode usulan.

Gambar 4-10 menunjukkan perbandingan metode usulan dengan metode-metode GMM, DTSR, *Averaging Background Subtraction* (ABS), dan PCASM. Dalam gambar tersebut ditunjukkan juga ground truth sebagai acuan bentuk dan lokasi obyek yang harus dideteksi. Dapat diketahui bahwa, metode PCASM dan GMM masih mengenali bayangan pemain, sementara metode ABS dan DTSR sudah dapat menyaring bayangan yang mengganggu. Dari Gambar 4-10 (4) menunjukkan metode DTSR terlihat rentan terhadap *noise*. Filter bitwise spasial terbukti dapat

memfilter bayangan pemain dengan baik. Akan tetapi, ada saat dimana proses filter ini membuat bentuk obyek tererosi, sehingga menyebabkan patahan bagian obyek. Contoh patahan ini dapat dilihat pada Gambar 4-11, satu obyek terdeteksi sebagai lima obyek oleh karena ada bagian-bagian tubuh yang terputus.



Gambar 4-11. Gambar hasil filter bitwise spasial yang tidak sempurna

Hasil dari deteksi obyek dapat dilihat di dalam *confusion* matriks Tabel 4-4. Nilai pembandingan yang digunakan dari metode ini digunakan skor F1, TPR, dan FPR.

Berikut adalah persamaan TPR, dan FPR.

$$TPR = \frac{TP}{(TP + FN)} \quad (4-5)$$

$$FPR = \frac{FP}{(FP + TN)} \quad (4-6)$$

Pengujian metode usulan menggunakan grafik AUC [41], metode grafik AUC adalah grafik penguji sebuah metode apakah metode tersebut layak digunakan sebagai pengklasifikasi atau tidak, dan seberapa baik posisi pengklasifikasi tersebut terhadap metode lain.

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FN + FP} \quad (4-7)$$

Dalam pengembalian informasi, nilai prediktif positif disebut sebagai presisi (persamaan (4-8)) dan sensitifitas disebut *recall* (ekuivalen dengan TPR). Oleh karena asumsi jumlah TN / (TN+TP) besar maka penggunaan skor F1 digunakan

sebagai ukuran tunggal kinerja tes untuk kelas positif. Skor F adalah rata-rata dari presisi dan *recall* [41]. Dari perbandingan diatas dapat diketahui bahwa metode usulan memiliki nilai F1 sebesar 0,805 lebih besar daripada metode-metode lain seperti GMM, DTSR, PCASM, dan POBS sebesar 0,281, 0,456, 0,648, dan 0,754.

Tabel 4-4. Tabel perbandingan metode usulan dan metode lain

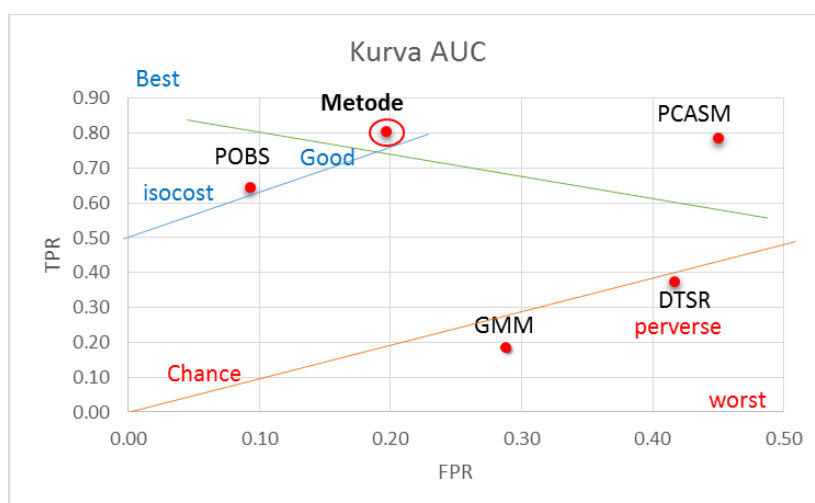
Metode	TP	FN	FP	F1
GMM [38] [39]	0,1679	0,7385	0,068	0,281
DTSR [40]	0,339	0,5675	0,2419	0,456
PCASM [37]	0,713	0,1934	0,5837	0,648
POBS	0,5856	0,3209	0,0602	0,754
Metode usulan	0,731	0,1754	0,1793	0,805

Tabel 4-4 dapat diketahui bahwa nilai TP metode usulan memiliki nilai yang tertinggi yaitu 0,73 dibanding dengan nilai metode-metode lain. Bahkan POBS hanya memiliki nilai 0,585 justru lebih kecil daripada PCASM yaitu sebesar 0,713, dimana nilai ini selisih sekitar 0,02 dengan metode usulan. Berdasarkan teori, semakin nilai FN kecil maka sistem akan semakin baik, karena bila nilai ini semakin kecil maka F1 akan semakin besar (membentuk relasi negative saling bertolak belakang antara FN dan F1). Nilai FP metode usulan kalah dengan POBS yaitu sebesar 0,0602 yang merupakan nilai paling unggul dibanding dengan nilai yang lain. Sedangkan pada Tabel 4-5 menunjukkan pembentuk grafik ROC atau grafik AUC (*Area Under Curve*). Dari tabel ini juga dapat diperoleh bahwa metode usulan memiliki nilai TPR yang tinggi yaitu 0,81 dibanding dengan metode-metode lain GMM 0,19, DTSR 0,37, PCASM mendekati metode usulan yaitu 0,79, dan POBS 0,65. Sedangkan FPR metode POBS memiliki nilai yang lebih baik yaitu 0,09 dibanding dengan metode usulan 0,20. Hal ini dikarenakan dalam metode POBS FP+TN memiliki nilai yang lebih kecil daripada FP sehingga

nilai FPR menjadi kecil. Garis hijau pada *Gambar 4-12* adalah garis yang menunjukkan posisi kelayakan sebuah metode [41].

Tabel 4-5. Tabel TPR dan FPR untuk grafik AUC

Metode	TPR	FPR
GMM [38] [39]	0,19	0,29
DTSR [40]	0,37	0,42
PCASM [37]	0,79	0,45
POBS	0,65	0,09
Metode Usulan	0,81	0,20



Gambar 4-12. Grafik Area Dalam Kurva (AUC) antar metode pembandingan.

Sumbu horizontal adalah FPR (*False Positive Rate*) atau laju pixel bukan obyek tetapi terdeteksi sebagai *foreground* dan sumbu vertical adalah TPR (*True Positive Rate*) atau laju pixel obyek tetapi terdeteksi sebagai *foreground*. TPR menunjukkan total jumlah pixel yang tepat terdeteksi sebagai pemain oleh metode usulan, sedangkan FPR menunjukkan total jumlah pixel bukan pemain yang salah tetapi tetap terdeteksi oleh metode usulan. Garis merah yang menunjuk kiri bawah merupakan daerah yang menunjukkan sebuah metode yang memiliki nilai FPR rendah akan tetapi TPR tinggi, dan metode yang berada pada daerah tersebut haruslah diganti. Garis merah yang menunjuk pada daerah kanan bawah merupakan

daerah terburuk yakni daerah yang memiliki nilai FPR tinggi dan TPR rendah, seperti keputusan yang diberikan oleh FPR rendah dan TPR tinggi, sebuah sistem haruslah dirubah dan tidak digunakan. Dengan kata lain hanya sistem yang berada pada *isocost* hijau saja yang dapat digunakan. Daerah merah tersebut disebut sebagai *preserve*.

Tabel 4-6. Tabel akurasi dan presisi antar metode

Metode	Akurasi	Presisi
GMM	0.29	0.71
DTSR	0.46	0.58
PCASM	0.65	0.55
POBS	0.75	0.91
Metode Usulan	0.80	0.80

Berdasarkan visualisasi, grafik AUC Gambar 4-12 hanya metode usulan sajalah yang layak digunakan sebagai detektor, karena ia berada pada perpotongan garis *isocost* biru dan *isocost* hijau.

Proses deteksi ini divalidasi dengan menggunakan membandingkan akurasi tiap-tiap metode pembanding, adapun proses validasinya dilakukan dengan menggunakan persamaan presisi (PPV) (4-7) dan akurasi (ACC) (4-8) metode usulan menempati nilai yang tinggi masing-masing yaitu 0,80.

$$PPV = \frac{TP}{TP + FP} \quad (4-8)$$

$$ACC = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (4-9)$$

Metode POBS memiliki tingkat presisi yang tertinggi dibanding yang lain yaitu 0,91, akan tetapi nilai akurasi metode usulan memiliki nilai tertinggi yaitu 0,80 dan tingkat presisi pengenalan obyek 0,80. Dari *Tabel 4-6* metode GMM walaupun memiliki nilai presisi yang tinggi yaitu 0,71 akan tetapi memiliki tingkat akurasi yang paling rendah yaitu 0,29.

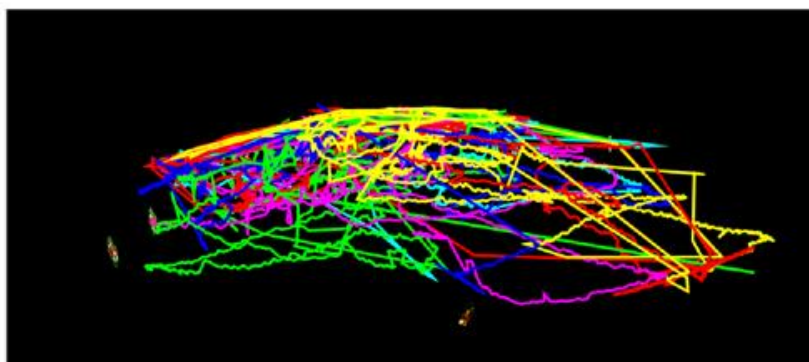
4.4 Proses Tracking

Proses *multi-tracking* menggunakan algoritma Kalman filter dengan menggunakan Hungarian untuk penugasan pemain. Penelitian ini membandingkan dua metode yaitu metode *tracking* menggunakan partikel filter dengan representasi matriks sparse dan metode usulan. Berikut adalah hasil dari penelitian tersebut. Tabel 4-7 menunjukkan bahwa *Ground Truth* pada database dapat diketahui bahwa total obyek yang tertampil adalah 25 obyek dengan obyek adalah pemain seragam putih sebanyak 11 orang, pemain seragam hitam sebanyak 11 orang, bola 1 buah, hakim garis 1 orang, dan wasit 1 orang.

Tabel 4-7. Tabel jumlah obyek ideal yang harus tertampil

Nama Obyek	Jumlah terdeteksi
Pemain seragam Putih	11
Pemain seragam Hitam	11
Bola	1
Hakim garis	1
Wasit	1

Berikut ini adalah percobaan proses *multi-tracking* metode usulan dengan menggunakan Kalman filter sebagai *trackingnya* dan algoritma Hungarian sebagai penugasan pengenalan multi obyek.



INPUT hasil Kalman Filter

Gambar 4-13. Hasil Tracking menggunakan Kalman Filter pada 1900 frame

Gambar hasil proses *tracking* adalah sebagai berikut ini seperti yang tertampil di Gambar 4-13. Diketahui bahwa trayektori masih rentan terhadap perubahan kesalahan pengenalan akibat dari oklusi, oleh sebab itu perlu adanya optimasi trayektori yang memperbaiki kesalahan pengenalan. Kesalahan pengenalan sebelum mengalami optimasi dapat dilihat pada Tabel 4-8. Dari tabel tersebut dapat diketahui bahwa sebelum optimasi para pemain tim seragam putih terkenali sebanyak 24 obyek, tim seragam hitam terkenali sebanyak 20 obyek, bola tidak terkenali, hakim garis terkenali sebagai 2 obyek, dan wasit terkenali sebanyak 2 obyek. Sedangkan setelah optimasi diberlakukan terjadi perbaikan, pemain seragam putih dikenali sebanyak 6 obyek, seragam hitam sebanyak 7 obyek, hakim garis sebanyak 1 obyek, dan wasit sebanyak 1 obyek. Proses pengenalan diatas diketahui bahwa bola tidak dapat terkenali dengan baik karena obyek memiliki luas yang terlalu kecil dan sering kondisinya intermiten yaitu rata-rata bila kelihatan bola memiliki luas kurang dari sepuluh pixel. Dengan kata lain dengan adanya optimasi pengenalan mengalami perbaikan sebesar 68 % yaitu dari 48 pengenalan menjadi 15 pengenalan. Sedangkan, setelah optimasi tingkat akurasi pengenalan menjadi 60% (ada 40% obyek yang tak terkenali), dibanding dengan sebelum optimasi tingkat akurasi pengenalan sebesar 192%.

Tabel 4-8. Tabel rata-rata hasil tracking obyek yang terdeteksi

Nama Obyek	Hasil Deteksi Sebelum Optimasi	Hasil Deteksi Sesudah Optimasi
Pemain seragam Putih	24	6
Pemain seragam Hitam	20	7
Bola	0	0
Hakim garis	2	1
Wasit	2	1
TOTAL	48/dari 25 Obyek (192% akurasi)	15/dari 25 Obyek (60% akurasi)

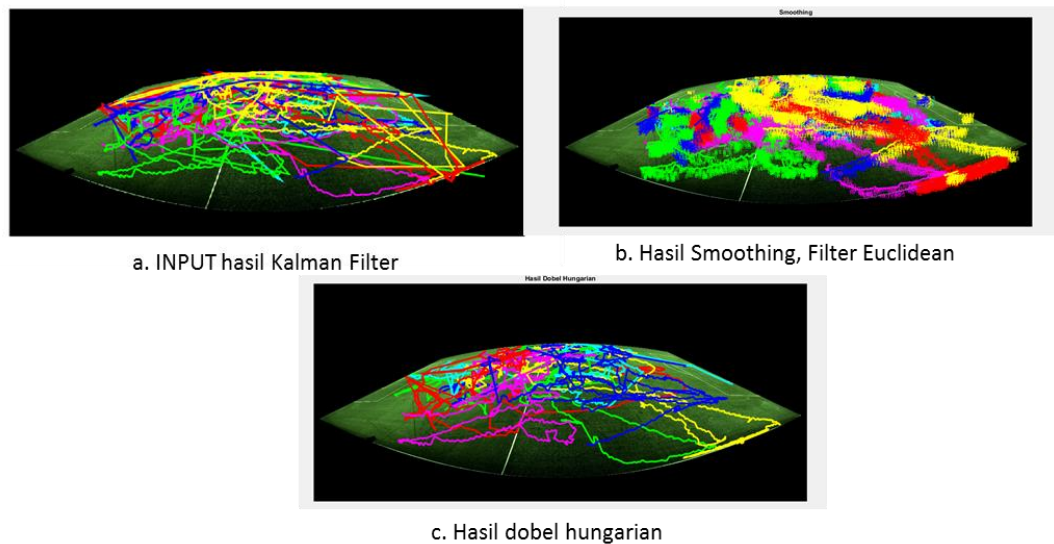
Artinya, setelah menggunakan metode optimasi terdapat 92% overlap obyek yang dikenali. Overlap terjadi disaat 15 obyek yang terdeteksi dilacak berkali-kali, sehingga satu obyek mempunyai beberapa garis trayektori. Dengan kata lain apabila 100% adalah 15 obyek maka $\frac{48}{15} = 3,20$ atau sekitar 320 %.

4.4.1 Optimasi Trayektori

Optimasi trayektori memiliki dua proses yaitu penghalusan sinyal menggunakan *threshold* yang berasal dari jarak euclidean dan penugasan kembali menggunakan algoritma Hungarian. Dari hasil pemrosesan metode usulan didapatkan bahwa hasil *tracking* setelah mengalami optimasi memiliki ketepatan sebesar 15 obyek dari 25 obyek atau tingkat akurasi *tracking* sebesar 60%. Pada subbab selanjutnya dijelaskan mengenai langkah-langkah proses optimasi, yaitu dimulai dari menyingkirkan nilai ekstrim yang lebih tinggi dari *threshold*, kemudian mengaplikasikan kembali algoritma Hungarian sebagai penugasan kembali.

4.4.1.1 Penghalusan Menggunakan Threshold

Penghalusan proses pertama ini dilakukan untuk menghilangkan lecutan yang terjadi oleh algoritma Kalman filter yang sedang melakukan pembelajaran menemukan lokasi obyek. Dalam metode usulan menggunakan inisialisasi lokasi awal pada posisi (0,0). Dalam melakukan pengenalanya Algoritma Kalman filter akan mengalami perbaikan semakin akurat, hal ini ditunjukkan pada Gambar 4-16 (a) dari garis hijau yang berasal pojok kiri atas (posisi lokasi awal (0,0)) menuju ke titik obyek kanan tengah bawah. Hasil penghalusan dapat dilihat pada Gambar 4-16 (b) dibawah ini. Gambar tersebut masih terpotong dan tujuan proses selanjutnya adalah menggabungkan kekosongan dalam satu jalur trayektori tersebut sebagai obyek yang sama.



Gambar 4-14. Hasil proses optimasi Trayektori pada 1900 frame

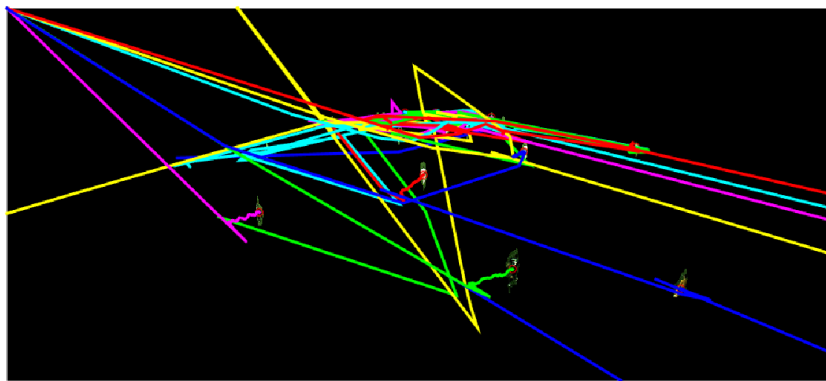
Gambar 4-14 (c) secara visual memperlihatkan bagaimana dalam satu jalur trayektori memiliki beberapa garis penyokong yang diperlihatkan oleh warna garis yang berbeda-beda dari garis trayektori input (a) dan (b).

4.4.1.2 Penugasan Kembali Menggunakan Hungarian

Proses setelah penghalusan pada poin 4.4.1.1 adalah melakukan penggabungan beberapa garis penyokong milik satu obyek yang sama. Acuan dari penggabungan tersebut berdasarkan pada posisi inialisasi obyek yang terkenal pada frame sebelumnya. Dengan menggunakan algoritma penugasan Hungarian digabung dengan pembuatan matriks biaya dari jarak euclidean diberlakukan. Hasil dari penugasan kembali ini dapat dilihat secara bentuk visual di Gambar 4-14 (c). Pada gambar ini masih terdapat overlap antara beberapa obyek walaupun penugasan sudah diberlakukan. Kesalahan pengenalan oleh karena penugasan ini diperoleh dari keterbatasan Hungarian yang harus memilih obyek dengan jarak optimal obyek-terhadap-pekerjaan (terdekat) pada matriks biaya. Contoh matriks biaya tersebut dapat diketahui pada Tabel 4-10 dan secara lengkap dapat dilihat pada Lampiran 1. Dari tabel ini diketahui bahwa threshold dipilih adalah 50 poin. Pada subbab selanjutnya akan dijelaskan mengenai strategi penulis untuk menganalisa error inialisasi pada hasil *tracking*.

4.4.1.3 Analisa Error Hasil Tracking

Oleh karena metode kalman filter adalah metode *tracking* yang berdasar persamaan linier sehingga adjustment terhadap pergerakan non linear memerlukan waktu, akibatnya apabila obyek intermiten (tiba-tiba hilang dan muncul kembali) akan mengakibatkan pelacakan akan tidak setabil, ketidakstabilan ini dapat terlihat dengan jelas dari gambar trayektori yang melebar tak beraturan, seperti yang terlihat pada gambar Gambar 4-15.



Gambar 4-15. Garis tak beraturan pada awal tracking

Tabel 4-9 dibawah ini menampilkan contoh kesalahan *tracking* yang dilakukan oleh metode usulan yang mengakibatkan hasil akurasi *tracking* sebesar 60% pada 1900 frame analisa. Dari tabel tersebut didapatkan bahwa frame terdapat saat dimana obyek tidak terkenali, yaitu dilambangkan dengan angka nol (warna merah). Perubahan obyek dapat terlihat dari berubahnya angka pengenalan (warna biru). Berdasarkan hasil percobaan, dapat disimpulkan bahwa pengenalan tidak stabil dikarenakan oleh dua hal yaitu: oklusi terhadap obyek yang lain dan kondisi *intermitten* hasil detektor obyek. Apabila garis trayektori yang didapatkan dari lokasi hasil *tracking* memiliki bentuk seperti lecutan yang besar (seperti yang tertampil pada Gambar 4-15), maka dapat dikatakan bahwa terdapat salah pengenalan dalam proses *tracking*. Metode normalisasi data trayektori menggunakan threshold dan penugasan kembali diberlakukan dengan tujuan untuk memperbaiki kesalahan hasil *tracking* tersebut. Gambar hasil penugasan kembali dapat dilihat pada Gambar

4-16. Walaupun demikian, dalam praktiknya tetap ada beberapa kesalahan *tracking* yang terjadi, oleh karena hasil optimasi yang masih menyisakan kesalahan.

Tabel 4-9. Tabel kesalahan identifikasi pemain

Frame	Ob 1	Ob 2	Ob 3	Ob 4	Ob 5	Ob 6	Ob 7	Ob 8	Ob 9
21	1	5	3	4	0	6	7	8	9
22	1	5	3	4	0	6	7	8	9
23	1	0	3	4	5	6	7	8	9
24	1	0	0	4	5	6	7	8	0
25	1	11	2	4	5	6	7	8	3
26	1	11	2	4	10	6	7	8	3
27	1	11	2	4	10	6	7	8	3
28	1	9	2	4	10	6	7	8	3
29	1	9	2	4	10	6	7	8	3
30	1	0	2	4	10	6	7	8	3

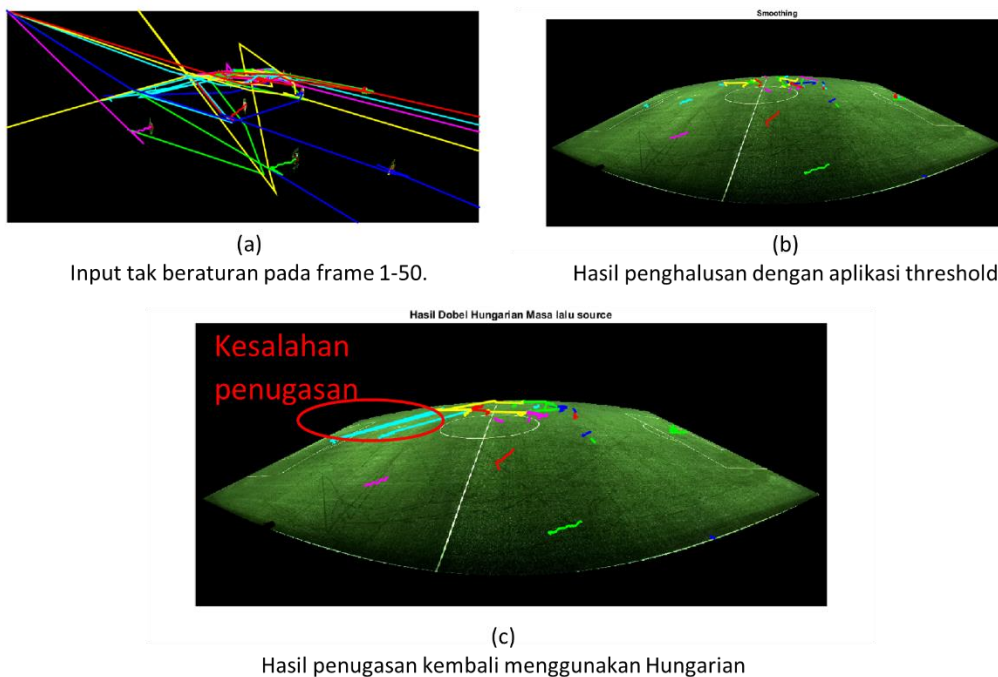
Dalam Gambar 4-16 (b) kesalahan tersebut dapat terlihat dari garis biru yang panjang linier yang overlap dengan garis kuning. Garis biru dalam lingkaran merah (Gambar 4-16 (c)) tersebut seharusnya tidak overlap, karena garis biru tersebut adalah obyek. Sementara itu metode usulan mendeteksi bahwa garis biru tersebut adalah bagian dari trayektori obyek lain. Sehingga secara visual dapat terlihat terjadinya penggabungan garis antara obyek satu dengan yang lain seolah-olah adalah satu obyek. Kesalahan tersebut dikarenakan oleh kesalahan penugasan algoritma Hungarian. Algoritma Hungarian bekerja melakukan penugasan berdasarkan nilai terkecil dari matriks biaya (*cost matrix*) jarak euclidean antara obyek satu dengan obyek yang lain (lihat Tabel 4-10). Oleh karena hanya berdasarkan jarak terkecil tanpa memandang bobot vector pergerakan obyek maka penugasan dapat terjadi kesalahan. Dalam Tabel 4-7 dapat dilihat bahwa dari 25 obyek yang tertangkap terdapat sekitar 10 obyek yang lolos tidak dapat terlacak

dengan baik. Hal ini terjadi karena pada saat awal proses deteksi obyek detektor tidak dapat mengenali obyeknya. Pengenalan obyek tersebut dibandingkan dengan ground truth Gambar 4-4 (a). Oleh karena dari awal obyek tak terkenali maka proses *tracking* tidak dapat dihasilkan, dalam Lampiran 1 terlihat bahwa obyek 18 keatas ditampilkan parameter *NaN* atau disebut juga sebagai *Not a Number* alias tidak dihasilkan suatu nilai. Pada frame ke-1 sampai 5, obyek terlihat masih belum sempat dikenali, akan tetapi dapat dilihat bahwa jarak Euclidean sangat besar yaitu melebihi ambang batas yang ditetapkan yakni 50. Sehingga selanjutnya difilter apabila jarak lebih dari 50 maka digantikan dengan *NaN*. Pemberian *NaN* pada aplikasi digunakan untuk memberi kode pada komputer bahwa saat proses perhitungan variabel tidak perlu diproses. Pada Tabel 4-10 dapat diketahui bahwa baris adalah frame t dan kolom adalah para obyek yang terkenali oleh detektor. Dimana tabel tersebut menunjukkan jarak obyek antar frame. Semakin besar jarak Euclidean berarti semakin cepat pulalah obyek tersebut bergerak. Walaupun demikian, jarak Euclidean yang besar menunjukkan pula terjadinya *intermitten* pada detektor obyek sehingga tracker sempat meloloskan obyek pada waktu tertentu. Kelolosan obyek pada detektor selain berakibat lebarnya jarak Euclidean antar frame, juga mengakibatkan kemungkinan tracker tidak dapat mempertahankan label obyek yang *intermitten* tersebut. Untuk menghadapi hal ini, diperlukan algoritma stage kedua Hungarian untuk melakukan penugasan kembali, dengan memastikan obyek pada frame sebelumnya adalah obyek referensi pada frame saat ini. Hasil penugasan diharapkan dapat merekonstruksi trayektori sehingga obyek yang terkenali pada frame saat ini merupakan obyek yang dikenali pada frame sebelumnya. Garis trayektori ini digunakan sebagai fitur dalam analisa *heat map* untuk menentukan karakteristik pergerakan pemain sepakbola. Hal ini dapat dikatakan bahwa, hasil *tracking* ini merupakan peubah random, oleh karena itu maka hasilnya dapat dimasukkan dalam fungsi distribusi probabilitas sebagai karakteristik khusus suatu pola tingkah-laku serangan tim dalam bertanding hal ini dikenal dengan istilah *adversarial behavior pattern*.

Tabel 4-10. Jarak Euclidean antara obyek frame t-1 dan t

	Ob 1	Ob 2	Ob 3	Ob 4	Ob 5	Ob 6	Ob 7	Ob 8	Ob 9	Ob 10	Ob 11	Ob 12	Ob 13	Ob 14	Ob 15	Ob 16	Ob 17
fr 1	103	44	129	93	116	214	63	34	62	64	66	68	78	69	71	74	335
fr 2	108	23	126	16	17	2	34	37	6	34	28	36	40	38	34	61	20
fr 3	91	6	98	56	12	19	10	42	15	29	3	41	11	11	19	57	29
fr 4	84	2	8	107	1	2	3	50	2	2	2	57	4	1	1	30	5
fr 5	38	1	8	15	1	2	2	5	1	3	1	41	3	29	2	3	3
fr 6	2	1	6	9	0	1	2	4	0	1	0	2	3	82	41	4	1
fr 7	4	2	1	4	1	1	1	2	1	35	1	4	1	12	42	1	0
fr 8	30	1	1	1	0	0	2	1	2	36	1	2	2	7	4	1	0
fr 9	5	1	1	1	3	0	2	0	1	3	1	1	2	2	4	1	1
fr 9	2	1	1	2	2	1	2	0	1	4	1	1	1	1	2	1	0
fr 10	1	1	1	1	0	0	1	0	0	1	1	1	3	1	0	1	0
fr 11	0	1	0	3	2	2	1	271	0	0	1	1	2	0	1	1	0
fr 12	0	92	1	1	1	0	1	254	1	1	1	1	1	1	0	1	0
fr 13	1	18	4	2	0	1	1	75	0	1	1	1	2	0	1	1	0
fr 14	1	6	3	1	1	0	1	15	0	1	1	1	2	1	1	1	0
fr 15	1	2	1	1	1	0	2	10	1	1	0	1	2	1	1	2	0
fr 16	1	1	1	1	0	1	1	2	1	0	1	1	1	1	1	1	1
fr 17	0	0	1	1	0	1	1	1	1	1	0	1	1	0	1	1	0
fr 18	0	0	1	1	0	0	1	2	1	1	0	1	2	1	1	1	1
fr 19	0	0	0	1	1	0	1	1	1	1	1	0	1	79	0	56	1
fr 20	0	89	1	180	278	1	1	2	1	1	0	29	2	15	1	39	0
fr 21	0	18	60	138	54	0	0	0	1	1	0	24	0	5	0	2	0
fr 22	0	6	54	7	19	0	0	0	1	1	0	43	0	2	1	4	0

Metode usulan dalam proses deteksi hanya memandangi posisi (lokasi, dan pergerakannya) obyek saja, tanpa memandangi fitur pola obyek misalkan warna baju, nomor punggung, dsb. Oleh sebab itu fitur yang digunakan untuk mengenali terbatas hanya pada nilai pergerakannya, apabila estimasi nilai pergerakan terdapat kesalahan, proses revisi menjadi sulit dilakukan karena tidak adanya kepastian bahwa hasil pengenalan merupakan benar-benar obyek dari hasil pengenalan obyek dari fitur lain. Hal ini dapat terlihat pada Gambar 4-14 (a), dimana pada gambar tersebut ditunjukkan terjadinya overlap pelacakan.

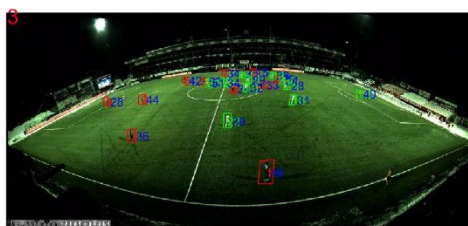


Gambar 4-16. Gambar proses optimasi perbaikan eror hasil tracking

4.4.2 Perbandingan dengan Metode *Tracking* Lain

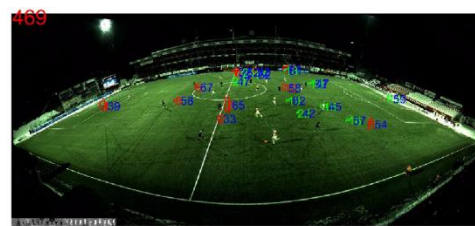
Metode lain yang dibandingkan adalah metode filter partikel dengan representasi matriks *sparse* berbasis pengenalan *template*, metode ini menggunakan komponen persebaran random pixel-pixel acuan sebagai indicator. Tiap-tiap pixel acuan tersebut berkorespondensi dengan citra yang berada di dalam *template*, apabila obyek pada frame saat ini cocok dengan lokasi pixel acuan pada frame sebelumnya, maka posisi pixel acuan dikatakan sama dengan posisi obyek pada frame selanjutnya, dan nilai perpindahan lokasi obyek pada frame sebelum dan

sesudah di simpan sebagai matriks trayektori. Metode ini dipilih sebagai metode pembandingan sebagai metode yang mewakili metode lain yang bekerja pada ranah operasi poin spasial langsung tanpa memodelkan *foreground* dan *background* terlebih dahulu. Berikut adalah gambar output hasil percobaan dengan menggunakan metode lawan dibandingkan dengan metode usulan. Gambar 4-17 (a) dan (b) adalah hasil dari percobaan metode lawan, sedangkan Gambar 4-17. (c) dan (d) adalah hasil percobaan metode usulan.



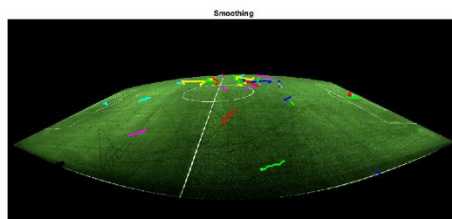
(a)

Citra awal pada frame ke-3 tracking masih dapat mengenali para pemain



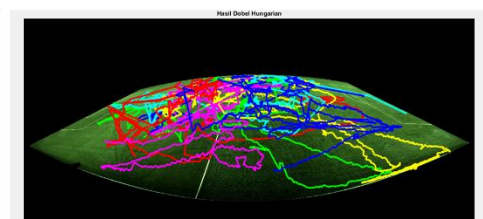
(b)

Citra pada frame ke-469 tracking sudah tidak dapat mengenali semua pemain



(c)

Pada frame ke-50 metode usulan masih dapat mengenali obyek dengan baik



(d)

Pada frame ke 1900 metode usulan tetap masih dapat mengenali obyek dengan akurasi pengenalan 19 obyek terkenal dari 25 obyek, dan 15 obyek dari 19 obyek dapat direkonstruksi trayektorinya dengan baik.

Gambar 4-17. Hasil Tracking dengan metode partikel filter dengan representasi sparse

Percobaan menggunakan metode lawan dihasilkan bahwa pada frame uji ke-469 pelacak telah kehilangan identifikasi terhadap total keseluruhan pemain sehingga dapat dilihat pada Gambar 4-17 (b) bahwa sistem kerangka obyek berada mengurungi obyek selain pemain. Hal ini diakibatkan oleh karena pergerakan pemain yang sangat cepat lebih cepat daripada kemampuan pelacak melacak pixel

yang dikategorikan sebagai pemain. Metode lawan menggunakan *template matching* sebagai sumber pengenalannya, apabila beberapa bagian tubuh pemain berada di beberapa bagian *template matching* maka detektor akan memiliki kesalahan pengenalan yang tinggi terhadap lokasi sebenarnya obyek tersebut. Berdasarkan hal ini, metode lawan memiliki tingkat pengenalan yang rendah disbanding dengan metode usulan. Berbeda dengan metode lawan, metode usulan menggunakan pemisah *background* dan *foreground* sebagai dasar deteksi obyek, pelacakan obyek dapat maksimal apabila semua *pixel background* dihilangkan pada frame, sehingga yang tersisa adalah pixel yang dihakimi sebagai *foreground*, oleh karena itu, sistem *tracking* selalu dapat mengenali obyek walaupun obyek tersebut bergerak dengan kecepatan random. Sehingga didapatkan hasil maksimal metode usulan dengan keseluruhan 1900 frame uji didapatkan rata-rata pengenalan sebesar 15 obyek terkenali dari 25 obyek.

BAB 5

KESIMPULAN

Kesimpulan yang didapatkan dari penelitian kali ini dapat dirumuskan sebagai berikut ini.

- Dimana penelitian ini terdiri dari dua proses utama yaitu deteksi obyek dan *tracking*. Proses deteksi obyek menggunakan metode *background subtraction* dengan mengasumsikan obyek (para pemain, bola dan wasit merupakan obyek) merupakan *foreground* dan lapangan beserta garis lapangan merupakan *background*. Sementara itu, *tracking* menggunakan metode *tracking* linier berbasis Kalman Filter.
- Metode Kalman Filter adalah metode *tracking* satu obyek yang dimodifikasi khusus untuk multi-obyek, dengan cara memasukkan obyek-obyek sebagai vector kolom pada matriks masukan. Agar terjaga perhitungan Kalmannya, metode usulan menggunakan dua step penugasan Hungarian.
- Dalam metode ini pada bagian deteksi obyek dikerjakan filter spasial bitwise yang berfungsi untuk menghilangkan bayangan pemain. Filter ini terbukti berhasil menghilangkan bayangan dan noise dengan efektif sebagai bentuk solusi dari tantangan tujuan poin ke-dua, hal ini dapat terlihat dari nilai F1-score terhadap ketepatan tertinggi yaitu 0.80 dibanding dengan empat metode-metode lain PCASM, GMM, DTSR, dan POBS yang memiliki F1-score sbagai berikut: 0.64, 0.281, 0.456, dan 0.754. Berkenaan dengan hasil tersebut maka tujuan penelitian poin ke-dua terpenuhi.
- Akan tetapi ada saat dimana pemain bergerak menyamping, filter ini menghilangkan sedikit bagian tubuh pemain dan menimbulkan patahan bagian tubuh pemain. Untuk menanggulangi hal itu maka operasi morfologi diaplikasikan untuk menyambung patahan tersebut. Efek dari operasi morfologi adalah membuat bentuk tubuh pemain sedikit berisi, akibatnya ketika dua atau lebih pemain bergerak beriringan rentan terhadap oklusi.
- Sedangkan pada proses *tracking*, metode usulan dibandingkan dengan metode partikel filter dengan representasi matriks sparse. Sebagai bentuk jawaban atas

tantangan pada tujuan penelitian poin pertama dan ketiga, peneliti membandingkan hasil metode usulan dengan metode lawan, dan hasilnya sampai pada akhir frame uji metode usulan tetap dapat mendeteksi dan melacak obyek dengan rata-rata 15 obyek terlacak dengan baik (60% akurasi *tracking*), dibanding dengan metode lawan yang pada frame ke 469 semua obyek kehilangan pelacak.

- Kesuksesan metode usulan ini diakarenakan mengaplikasikan dua kali algoritma Hungarian yaitu pertama saat penugasan lokasi pemain di dalam metode Kalman Filter dan terakhir di output Kalman filter setelah proses penghalusan sinyal trayektori dilakukan. Walaupun demikian, terdapat sekitar 40% dari obyek tidak dapat dipulihkan garis trayektorinya sehingga proses pelacakan pemain tidak dapat maksimal (hanya mendapatkan 15 pemain terlacak dari 25 keseluruhan obyek).

Terdapat empat hal catatan utama pada penelitian kali ini yang masih memerlukan perbaikan, dan hal ini digunakan sebagai saran untuk pengembangan penelitian berikutnya.

- Pertama, pengambilan obyek dilakukan menggunakan kamera statis, sehingga dimensi pengenalan obyek terbatas pada frame layar. Penelitian selanjutnya pengambilan dapat dilakukan oleh multi kamera sehingga dapat dilakukan pengenalan obyek melalui berbagai posisi dan hal ini dapat meningkatkan akurasi pelacakan pemain.
- Kedua, Oleh karena obyek deteksi hanya berdasarkan pemisahan *foreground-background* tanpa mempedulikan bentuk obyek individu pemain yang kukuh sehingga ketika rekonstruksi dari noise dan bayangan, titik tengah obyek yang mengisyaratkan lokasi meleset dikenali. Penelitian selanjutnya dapat dilakukan deteksi pemain berdasarkan pemodelan tulang penguat sehingga dengan tepat memprediksi kegiatan pemain sekaligus mendeteksi lokasinya.
- Ketiga, dalam penelitian kali ini pemisahan oklusi pemain dengan menggunakan penugasan identitas pemain menggunakan algoritma Hungarian, dimana dihasilkan bahwa metode ini masih menyisakan kemungkinan kesalahan penugasan. Penelitian selanjutnya, dapat dilakukan dengan

memisahkan pemain dengan melihat posisi pemain dari berbagai sumber kamera dengan dikuatkan oleh prediksi pergerakan pemain selanjutnya.

- Keempat, Analisa trayektori yang dilakukan oleh metode usulan digunakan untuk merekonstruksi arah gerak para pemain. Penelitian selanjutnya dapat menggunakan data trayektori ini untuk dihitung secara statistik untuk mendapatkan pola agresi pemain saat bertanding. Dari pola ini dapat digunakan untuk memprediksi taktik dan strategi tim saat sedang bertanding.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] T. Zhang, B. Ghanem, C. Xu and N. Ahuja, "Occlusion Detection via Structured Sparse Learning for Robust Object Tracking," in *Computer Vision in Sports*, Switzerland, Springer International Publishing, 2014, pp. 93-112.
- [2] J. Liu, X. Tong and W. Li, "Automatic Player Detectopm, Labeling and Tracking in Broadcast Soccer Video," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 103-113, 2009.
- [3] L. Sun and G. Liu, "FIELD LINES AND PLAYERS DETECTION AND RECOGNITION IN SOCCER VIDEO," in *ICASSP*, Taiwan, 2009.
- [4] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan and I. Matthews, "Large-Scale Analysis of Soccer Matches using Spatiotemporal Tracking Data," in *IEEE International Conference on Data Mining*, Senzhen, 2014.
- [5] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Computer Science Review*, Vols. 11-12, pp. 31-66, May 2014.
- [6] P. Perez, C. Hue, J. Vermaak and M. Gangnet, "Color-Based Probabilistic Tracking," in *ECCV 2002, LNCS 2350*, Berlin Heidelberg, 2002.
- [7] S. A. Pettersen, D. Johansen, H. Johansen, V. Berg-Johansen, V. R. Gaddam, A. Mortensen, R. Langseth, C. Griwodz, H. K. Stensland and H. P., "Soccer video and player position dataset," 2014. [Online]. Available: <http://home.ifi.uio.no/paalh/dataset/alfheim/>.
- [8] G. Thomas, T. B. Moeslund and A. Hilton, *Computer Vision in Sports*, New York: Springer, 2014.
- [9] F. Yan, W. Christmas and J. Kittler, "Ball Tracking for Tennis Video Annotation," in *Computer Vision in Sports, Advances in Computer Vision and Pattern Recognition*, Springer, 2014, pp. 25-46.
- [10] S. Tamaki and H. Saito, "Plane Approximation-Based Approach for 3D Reconstruction of Ball for Performance Analysis in Table Tennis," in *Computer VIsion in Sports, Advances in Computer VIsion and Pattern Recognition*, Springer, 2014, pp. 47-66.
- [11] P. Spagnolo, P. L. Mazzero, M. Leo, M. Nitti, E. Stella and A. DIstante, "On-Field Testing and Evaluation of a Goal-Line Technology System," in *Computer VIsion in Sports, Advances in Computer Vision and Pattern Recognition*, Springer, 2014, pp. 67-90.

- [12] T. Zhang, B. Ghanem, C. Xu and N. Ahuja, "Occlusion Detection via Structured Sparse Learning for Robust Object Tracking," in *Computer Vision in Sports, Advances in Computer Vision and Pattern Recognition*, Springer, 2014, pp. 93-112.
- [13] J. Liu and P. Carr, "Detecting and Tracking Sport Players with Random Forest and Context-Conditioned Motion Models," in *Computer Vision in Sports, Advances in Computer Vision and Pattern Recognition.*, Springer, 2014, pp. 113-132.
- [14] R. Gade and T. B. Moeslund, "Classification of Sport Types Using Thermal Imagery," in *Computer Vision in Sports, Advances in Computer Vision and Pattern Recognition*, Springer, 2014, pp. 209-228.
- [15] S. Wilson, C. K. Mohan and K. S. Murthy, "Event-Based Sports Video Classification Using HMM Framework," in *Computer Vision in Sports, Advances in Computer Vision and Pattern Recognition*, Springer, 2014, pp. 229-244.
- [16] B. Lee and M. Hedely, "Background estimation for video surveillance," in *Image and Vision Computing New Zealand (IVCNZ '02)*, New Zealand, 2002.
- [17] A. M. Rahman, B. Ahmed, A. M. Hossian and N. I. Mondal, "An adaptive background modeling based on modified running Gaussian average method," in *2017 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Bangladesh, 2017.
- [18] Z. Yi and F. Liangzhong, "Moving object detection based on running average background and temporal difference," in *2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering*, Hangzhou, 2010.
- [19] N. McFarlane and C. Schofield, "Segmentation and tracking of piglets in images," *Machine Vision and Applications*, vol. 8, pp. 187-193, 1995.
- [20] A. A. H. Mohamad and M. Osman, "Adaptive median filter background subtractions technique using fuzzy logic," in *2013 INTERNATIONAL CONFERENCE ON COMPUTING, ELECTRICAL AND ELECTRONIC ENGINEERING (ICCEEE)*, Khartoum, Sudan, 2013.
- [21] J. Zeng, Y. Wang, N. N. and H. E., "Extracting roadway background image: A mode based approach," *J. Transp. Res. Rep. 1994 (2006)*, pp. 82-88, 2006.
- [22] Z. Zeng, J. Jia, D. Yu, Y. Chen and Z. Z. Zhu, "Pixel modeling using histograms based on fuzzy partitions for dynamic background subtraction," *IEEE Transactions on Fuzzy Systems*, vol. PP, no. 99, pp. 1 - 1, 2016.

- [23] C. Richard Wren, A. Azarbayejani, T. Darrell and A. Paul Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Transaction of Pattern Analysis and Machine Intelligence*, vol. 19, pp. 780-785, 1997.
- [24] T. Bouwmans, F. El Baf and B. Vachon, "Background Modeling using Mixture of Gaussian for Foreground Detection - A Survey," *Recent Patents on Computer Science*, vol. 3, pp. 219-237, 2008.
- [25] D. Butler, V. Bove and S. Shridharan, "Real time adaptive foreground/background segmentation," in *EURASIP*, 2005.
- [26] M. Xiau, C. Han and X. Kang, "A Background reconstruction for dynamic scenes," in *International Conference on Information Fusion, ICIF 2006*, 2006.
- [27] Y. Zhang, Q. He, H. Wang, G. Guan, T. Xu and H. Chen, "Background subtraction based on pixel clustering," in *2016 IEEE International Conference on Information and Automation (ICIA)*, Zhejiang, 2016.
- [28] R. Luque, E. Dominguez, E. Palomo and J. Munoz, "An ART-type network approach for video object detection," in *European Symposium on Artificial Neural Networks*, 2010.
- [29] M. Brahmam and V. M. Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *2016 International Conference on Systems, Signals and Image Processing (IWSSIP)*, Bratislava, 2016.
- [30] C. Guyon, T. Bouwmans and E.-h. Zahzah, "Robust Principal Component Analysis for Background Subtraction: Systematic Evaluation and Comparison Analysis," in *INTECH, Principal Component Analysis, Book 1*, In Tech, March 2012, pp. 223-238.
- [31] S. Sumpeno, M. Hariadi and T. Aoki, "A Region-based Approach using LVQ for Semi-Automatic Video Object Extraction Technique," in *8th Seminar on Intelligent Technology and Its Applications*, Surabaya, Indonesia, 9-10 May 2007.
- [32] B. Sahbani and W. Adiprawita, "Kalman filter and Iterative-Hungarian Algorithm implementation for low complexity point tracking as part of fast multiple object tracking system," in *System Engineering and Technology (ICSET)*, Bandung, Indonesia, 2016.
- [33] J.-M. Jeong, T.-S. Yoon and J.-B. Park, "Kalman Filter Based Multiple Objects Detection-Tracking Algorithm Robust to Occlusion," in *SICE Annual Conference*, Sapporo, 2014.

- [34] A. Ribero, "Players Tracking in Football Game," University of Lisbon Portugal, Lisbon, Portugal, 2009.
- [35] P. Halvorsen, "Research Groups: Network and Distributed System: IFI Media," 2014. [Online]. Available: <http://home.ifi.uio.no/paalh/dataset/alfheim/>. [Accessed 14 March 2017].
- [36] P. Lucey, D. Oiver, P. Carr, Roth and I. Matthews, "Assessing team strategy using spatiotemporal data," in *KDD '13 Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013.
- [37] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan and I. Matthews, "Identifying Team Style in Soccer using Formations Learned from Spatiotemporal Tracking Data," in *IEEE International Conference on Data Mining Workshop*, 2014.
- [38] X. Gengjian, L. Song, J. Sun and J. Zhou, "Foreground detection: Combining background subspace learning with object smoothing model," in *2013 IEEE International Conference on Multimedia and Expo (ICME)*, San Jose, CA, 2013.
- [39] L. Yao and L. Miaogen, "An Improved Mixture-of-Gaussians Background Model with Frame Difference and Blob Tracking in Video Stream," *The Scientific World Journal*, vol. 2014, p. 9 pages, 2014.
- [40] P. Kaewtrakulpong and R. Bowden, "An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection," in *Video-Based Surveillance Systems*, Springer US, 2002, pp. 135-144.
- [41] M. Taha, H. H. Zayed, M. E. Khalifa and T. Nazmy, "Moving Shadow Removal for Multi-Objects Tracking in Outdoor Environments," *International Journal of Computer Applications*, vol. 97, no. 10, pp. 43-51, 2014.
- [42] M. W. D. Powers, "Evaluation: From Precision, Recall, and F-Factor to ROC, Informedness, Markedness and Correlation," *International Cognitive Science Conference*, 2003.
- [43] Feedback Instruments Ltd., *Digital Pendulum: Control in a Matlab Environment*, Sussex, UK: Feedback Instruments Ltd., 2006.
- [44] K. Tanaka and M. Sugeno, "Stability analysis and design of fuzzy control," *Fuzzy Sets and Systems*, vol. 45, pp. 135-156, 1992.

- [45] T. Fletcher, *CS4640: Image Processing Basics*, Utah: University of Utah, 2012.
- [46] P. M. Roth, H. Bischof, D. Skočaj and A. Leonardis, “Object Detection with Bootstrapped Learning,” in *Proceedings 10th Computer Vision Winter Workshop*, 2005.
- [47] A. Bovik, “Morphological Edge Detection,” in *The Essential Guide to Image Processing*, Elsevier, 2009, p. 311.

Halaman ini sengaja dikosongkan

LAMPIRAN 1 Matrix Biaya Antar Frame t dan $t - 1$

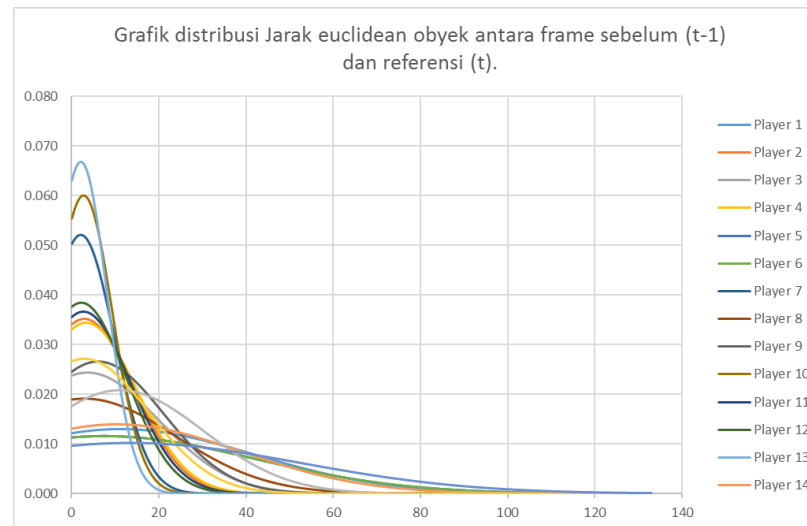
Tabel L-1 Cost Matrix antar frame

O1	O2	O3	O4	O5	O6	O7	O8	O9	O10	O11	O12	O13	O14	O15	O16	O17	O18	O19	O20	O21	O22	O23	O24	O25	
103	44	129	93	116	214	63	34	62	64	66	68	78	69	71	74	335	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	92
108	23	126	16	17	2	34	37	6	34	28	36	40	38	34	61	20	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	50
91	6	98	56	12	19	10	42	15	29	3	41	11	11	19	57	29	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	16
84	2	8	107	1	2	3	50	2	2	2	57	4	1	1	30	5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5
38	1	8	15	1	2	2	5	1	3	1	41	3	29	2	3	3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	3
2	1	6	9	0	1	2	4	0	1	0	2	3	82	41	4	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1
4	2	1	4	1	1	1	2	1	35	1	4	1	12	42	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0
30	1	1	1	0	0	2	1	2	36	1	2	2	7	4	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1
5	1	1	1	3	0	2	0	1	3	1	1	2	2	4	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0
2	1	1	2	2	1	2	0	1	4	1	1	1	1	2	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0
1	1	1	1	0	0	1	0	0	1	1	1	3	1	0	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0
0	1	0	3	2	2	1	271	0	0	1	1	2	0	1	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0
0	92	1	1	1	0	1	254	1	1	1	1	1	1	0	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0
1	18	4	2	0	1	1	75	0	1	1	1	2	0	1	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0
1	6	3	1	1	0	1	15	0	1	1	1	2	1	1	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1
1	2	1	1	1	0	2	10	1	1	0	1	2	1	1	2	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1
1	1	1	1	0	1	1	2	1	0	1	1	1	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2
0	0	1	1	0	1	1	1	1	1	0	1	1	0	1	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1
0	0	1	1	0	0	1	2	1	1	0	1	2	1	1	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5
0	0	0	1	1	0	1	1	1	1	1	0	1	79	0	56	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	237
0	89	1	180	278	1	1	2	1	1	0	29	2	15	1	39	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	46
0	18	60	138	54	0	0	0	1	1	0	24	0	5	0	2	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	40

LAMPIRAN 2 Contoh Hasil Tracking Sebagai Analisa Statistik

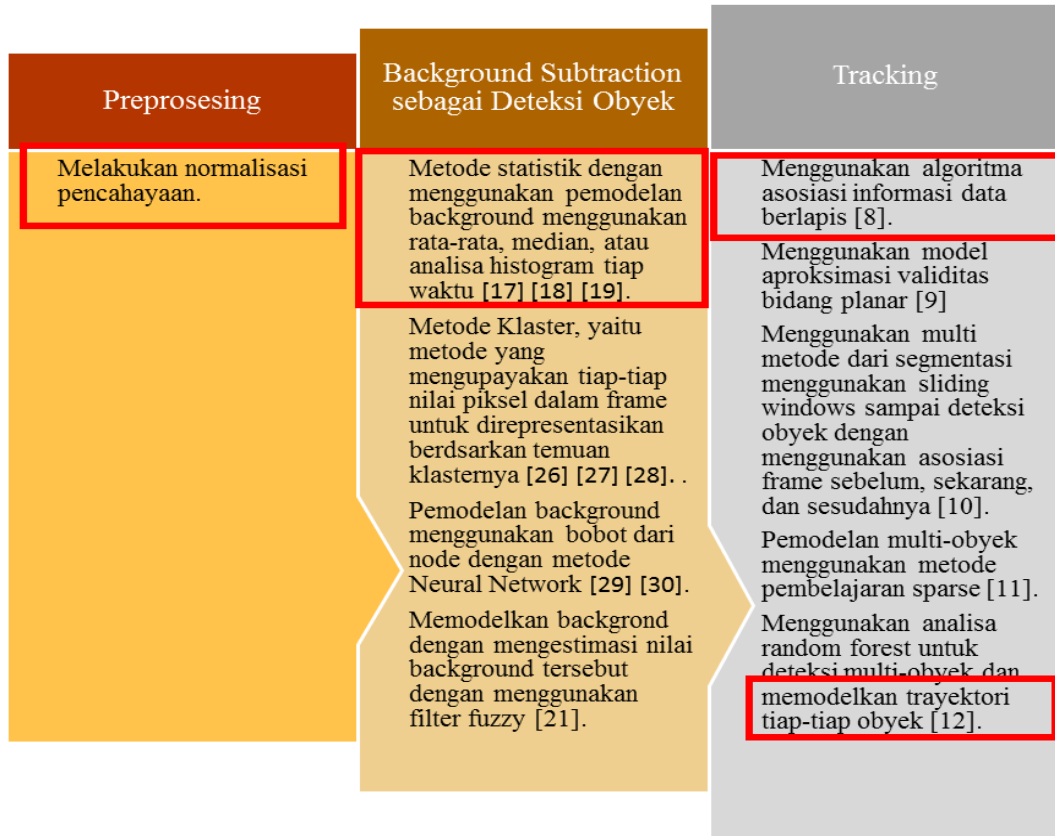
Tabel L-2. Statistik para pemain

MEAN	11.209	2.959	3.672	3.373	7.559	7.559	2.121	2.993	6.095	2.757	2.735	2.227	2.136	10.373	11.135	2.972	13.403
STD	30.678	11.332	16.363	11.591	34.437	34.437	7.654	20.848	14.991	6.643	10.883	10.374	5.969	28.569	19.148	14.677	39.083
0.1	0.012	0.034	0.024	0.033	0.011	0.011	0.050	0.019	0.025	0.055	0.036	0.038	0.063	0.013	0.018	0.027	0.010
0.2	0.012	0.034	0.024	0.033	0.011	0.011	0.051	0.019	0.025	0.056	0.036	0.038	0.063	0.013	0.018	0.027	0.010
0.3	0.012	0.034	0.024	0.033	0.011	0.011	0.051	0.019	0.025	0.056	0.036	0.038	0.064	0.013	0.018	0.027	0.010
0.4	0.012	0.034	0.024	0.033	0.011	0.011	0.051	0.019	0.025	0.056	0.036	0.038	0.064	0.013	0.018	0.027	0.010
0.5	0.012	0.034	0.024	0.033	0.011	0.011	0.051	0.019	0.025	0.057	0.036	0.038	0.064	0.013	0.018	0.027	0.010
0.6	0.012	0.034	0.024	0.033	0.011	0.011	0.051	0.019	0.025	0.057	0.036	0.038	0.065	0.013	0.018	0.027	0.010



Gambar L1. Representasi grafik distribusi pergerakan tiap-tiap pemain.

LAMPIRAN 3 Posisi Penelitian Terhadap Penelitian Lain



Gambar L2.Diagram posisi penelitian terhadap penelitian lain.

Dari diagram tersebut dapat diketahui bahwa penelitian yang dilakukan penulis, termasuk dalam perhitungan *background subtraction* dengan menggunakan perhitungan *statistic* dan dalam metode tracking, penelitian penulis termasuk pada menggunakan algoritma asosiasi berlapis dan pemodelan trayektori tiap-tiap obyek. Diagram ini sangat penting untuk menunjukkan *positioning* penelitian dibanding dengan metode yang telah ada.

Halaman ini sengaja dikosongkan

BIOGRAFI PENULIS



Atyanta Nika Rumaksari selesai menempuh gelar Master Teknik di Institut Teknologi Sepuluh Nopember Surabaya (ITS) pada tahun 2017, dengan jurusan Magister Telematika. Bidang yang ditekuninya adalah kecerdasan buatan, visi computer, komputasi paralel dan sistem embedded. Saat ini ia merupakan staf pengajar di fakultas Teknik Elektro program studi Teknik Komputer di Universitas Kristen Satya Wacana Salatiga (UKSW). Selain menempuh gelar Master di ITS ia pernah menempuh *Master of Business Administration* (MBA) di Institut Pengembangan Manajemen Indonesia Jakarta (IPMI) dan Strata 1 di UKSW.

Ia percaya bahwa inovasi produk yang membangun bangsa hanya dapat dicapai dengan melacak kebutuhan dan kesempatan. Dengan mengetahui kebutuhan kita dapat membuat suatu produk yang tepat guna, dan dengan mengetahui kesempatan kita dapat menentukan waktu yang tepat untuk melakukan rekayasa produk. Adapun, kesemuanya itu didasarkan dengan iman dan jiwa takut akan Tuhan, karena penulis sadar bahwa “*Takut akan Tuhan adalah permulaan pengetahuan*”.