

Identifikasi Garis pada Lapangan Sepak Bola **Berbasis *OpenCV***

TUGAS AKHIR

Oleh :

Dewa Puja Pratama 3310801008

Rahmatul Hidayah 3310801021

Disusun untuk memenuhi syarat kelulusan Program Diploma III



PROGRAM STUDI TEKNIK INFORMATIKA

POLITEKNIK NEGERI BATAM

BATAM

2011

LEMBAR PENGESAHAN

Batam, 22 Agustus 2011

Pembimbing,

Riwinoto, M. Kom

NIK. 103025

LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3310801008

Nama : Dewa Puja Pratama

adalah mahasiswa Teknik Informatika Politeknik Negeri Batam yang menyatakan bahwa tugas akhir dengan judul:

IDENTIFIKASI GARIS PADA LAPANGAN SEPAK BOLA BERBASIS OPENCV

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Negeri Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 22 Agustus 2011

Dewa Puja Pratama
3310801008

LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : 3310801021

Nama : Rahmatul Hidayah

adalah mahasiswa Teknik Informatika Politeknik Negeri Batam yang menyatakan bahwa tugas akhir dengan judul:

IDENTIFIKASI GARIS PADA LAPANGAN SEPAK BOLA BERBASIS OPENCV

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Negeri Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 22 Agustus 2011

Rahmatul Hidayah
3310801021

HALAMAN PERSEMBAHAN

Assalamualaikum Wr.Wb....

1. TA ini Q persembahkan untuk kedua orang tua , nenek q, adik q, serta keluarga ku yang telah memberikan doa, semangat, dan motivasi
2. Buat dosen pembimbing q Bpk.Riwinoto.M.Kom yang telah membantu ku, mendengarkan keluhan-keluhan q untuk menyelesaikan TA ini, maafkan ya pak klo slama ini udah merepotkan
3. Ucapan terimakasih buat Bpk Toar yang telah mengajari q, membantu ku dalam pembuatan Program TA ini
4. Buat jenggo dan jingga yang telah menemani ku, setia pada q sampai sekarang ini seneng memiliki kalian.....
5. Buat my Patner TA q Rahma yang suka buat q emosi hahahaha.... tapi seneng q punya partner seperti kamu penuh semangat ☺
6. Buat temen-temen team futsal....(“afri, adli, iqbal, antok, ucok, bayu, wira, arman, fitra, iwan, dandre, fatah, handa, takim, iksan”) klo dah pada lulus smua kapan-kapan kita main futsal bareng lagi ya hahahaha.....
7. Buat Guru Spritual q Bpk Said Nurinsani hahaha... serta ke 4 sahabat q (“Dedi, Ikin, Kiply, Erwan”) kalian sahabat terbaik q thanks buat semuanya.....
8. Terakhir buat semua pihak yang telah mendukung terlaksananya tugas akhir ini, maaf tidak bisa disebutin satu persatu, tapi q bertrimakasih untuk kalian semua....

Wassalamu’alaikum Wr.Wb....

DEWA PUJA PRATAMA ☺

HALAMAN PERSEMBAHAN

↳ Kepada Allah,,

Zat tunggal penguasa isi alam semesta dan segala pergerakannya yang penuh kejutan. Terima kasih untuk segala rupa dalam hidupku. Kau Maha Segalanya.

↳ Kepada Bapak dan Emak,,

Semua perjalanan hidup tentunya bermula dan akan selalu disyukuri atas bapak dan emakku (M.Saleh & Ratnalita) yang Amah sayangi. Karena mereka senyum ini hidup dan tangis ini berarti dalam karya ini. Terima kasih untuk segala percaya yang tak ada habisnya.

↳ Kepada Saudara Sedarahku,,

Adik-adikku tersayang, (Syahidah) moga betah di kampus barunya, (Abdul Aziz) moga juga betah disekolah barunya, (Siti Shaleha) semangat belajarnya, dan raih cita-citamu. Sadar atau tidak kalian berarti dalam perjalanan hidupku.

↳ Kepada Sahabat,,

Buat My Partner Tugas Akhir ini (Dewa Puja Pratama) yang selalu memberi supportnya, Para pemberi semangat tanpa jemu Farah, Anto, Fita, Tika, Sarah, Fitra, Prima, Arman, Mariaty, Ipit, Ferri, Lis, Fiza, Afriz, dll. Terima kasih untuk hari-hari aneka rasa.

↳ Kepada Dosen,,

Manusia-manusia terpilih tanpa pamrih Bpk. Riwinoto, Bpk. Uuf Brajawidagda, Bpk. Toar, Bpk. Afdhol Zikri, Ibu Riska, dll. Terima kasih atas semua jawaban akan tanda tanyaku. Atas semua ikhlas yang tak bisa kubalas.

↳ Kepada Kamu,,

Yang selalu memberiku semangat untuk menyelesaikan Tugas Akhir ini. Terima kasih atas segala perhatianmu. Serta semua jiwa yang singgah dalam hidupku.

Rahmatul Hidayah

KATA PENGANTAR

Segala puji dan syukur kita panjatkan kepada Allah SWT yang telah memberikan Rahmat dan Karunia-Nya, penulis bersyukur kepada Allah SWT atas selesainya Tugas Akhir yang berjudul "IDENTIFIKASI GARIS PADA LAPANGAN SEPAKBOLA BERBASIS OPENCV".

Dalam kesempatan ini pula penulis mengucapkan terima kasih kepada:

1. Allah SWT atas terselesainya Tugas Akhir ini
2. Kedua orang tua yang telah memberikan dukungan moril maupun materil
3. Bapak DR.Ir.Priyono Eko Sanyoto, M. Sc, selaku Direktur Politeknik Negeri Batam
4. Bapak Uuf Brajawidagda selaku koordinator Tugas Akhir
5. Bapak Riwinoto selaku dosen Pembimbing Tugas Akhir ini. Terima kasih atas bimbingan dan motivasi yang bapak berikan dalam penyelesaian Tugas Akhir ini
6. Keluarga yang telah memberikan do'a, dukungan dan semangat.
7. Bapak/ Ibu dosen program studi Teknik Informatika Politeknik Negeri Batam atas arahan dan bimbingannya
8. Teman-teman Teknik Informatika angkatan 2008 yang memberikan dukungan, semangat dan do'a.

Penulis menyadari bahwa masih banyak kekurangan dalam penyusunan laporan ini. Oleh karena itu, penulis sangat mengharapkan bantuan dari beberapa pihak baik berupa kritik maupun saran yang membangun dari pembaca. Akhir kata penulis mengucapkan terima kasih, semoga penulisan laporan ini dapat bermanfaat bagi pembaca yang ingin mengembangkan sebuah penelitian yang serupa.

Batam, Agustus 2011

Penulis

ABSTRAK

IDENTIFIKASI GARIS PADA LAPANGAN SEPAK BOLA BERBASIS *OPENCV*

Aplikasi identifikasi garis pada lapangan sepakbola membutuhkan *library OpenCV*. Pengenalan objek garis dimulai dengan pengambilan image per frame format RGB. Image RGB yang diambil di konversi dalam format grayscale dimana hasilnya adalah hitam dan putih, selanjutnya untuk deteksi tepi image menggunakan metode canny, kemudian pada proses deteksi akhir garis menggunakan metode Hough Line Transform dengan menggunakan ruang parameter koordinat polar yaitu sudut dan radius pada beberapa titik yang diambil.

Hasil deteksi garis pada image frame video menunjukkan hasil tidak stabil. Sedangkan untuk hasil deteksi garis pada file image menunjukkan hasil stabil.

Kata Kunci : Identifikasi Garis, Image RGB, Grayscale, Deteksi Tepi, Hough Line Transform.

ABSTRACT

IDENTIFICATION OF LINES IN THE SOCCER FIELDS BASED OPENCV

Application identification line on the football field requires OpenCV library. The introduction begins with the retrieval object image lines per frame RGB format. RGB image taken in grayscale format conversion in which the result is a black and white, next to the image edge detection using Canny method, then at the end of line detection using Hough Line Transform method using the parameter space which is polar coordinate angle and radius at some point taken.

The results of line detection in a video frame image shows the results unstable. As for the detection of lines in the image file shows stable results.

Keywords: Line Identification, Image RGB, Grayscale, Edge Detection, Hough Line Transform.

DAFTAR ISI

LEMBAR PENGESAHAN.....	i
LEMBAR PERNYATAAN	ii
LEMBAR PERSEMBAHAN.....	iv
KATA PENGANTAR.....	vi
ABSTRAKSI.....	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xiv
Bab I Pendahuluan.....	1
I.1 Latar Belakang.....	1
I.2 Rumusan Masalah	3
I.3 Batasan Masalah	3
I.4 Tujuan	3
I.5 Sistematika Penulisan	3
Bab II Landasan Teori	5
II.1 Pengenalan Objek Garis pada Image untuk Setiap Frame File Video	5
II.1.1 Proses RGB ke Grayscale Image.....	6
II.1.2 Proses dari Grayscale ke Edge Binary Image.....	7
II.1.3 Proses Mendeteksi Garis pada Edge Binary Image	10
II.2 Hough Line Transform.....	11
II.3 Pengenalan Garis pada Lapangan Sepakbola.....	17
II.4 Hubungan Antar Referensi	18
Bab III Analisis	19
III.1 Deskripsi Umum Sistem	19
III.2 Fitur Utama.....	20
III.3 Perangkat Lunak	20
III.3.1 Kebutuhan Fungsional	20
III.4 Use Case.....	20

III.4.1	Diagram <i>Use Case</i>	20
III.4.2	Skenario <i>Use Case</i>	21
III.5	Analisis Kelas.....	22
Bab IV	Perancangan dan Implementasi.....	24
IV.1	Diagram Interaksi.....	24
IV.1.1	Diagram Interaksi <i>Use Case</i> Pemutar Video.....	24
IV.1.2	Diagram Interaksi <i>Use Case</i> Pemutar Webcam.....	25
IV.1.3	Diagram Interaksi <i>Use Case</i> Ambil Image per Frame Video	26
IV.1.4	Diagram Interaksi <i>Use Case</i> Konversi ke Grayscale	26
IV.1.5	Diagram Interaksi <i>Use Case</i> Deteksi Edge	27
IV.1.6	Interaksi Diagram <i>Use Case</i> Deteksi Informasi Garis	27
IV.2	Diagram Kelas.....	28
IV.2.1	Rancangan Kelas Rinci	28
IV.3	Algoritma.....	30
IV.3.1	Algoritma pada Kelas Pemutar Video	30
IV.3.2	Algoritma pada Kelas Pemutar Webcam	31
IV.3.3	Algoritma pada Kelas Image per Frame	31
IV.3.4	Algoritma pada Kelas Grayscale	31
IV.3.5	Algoritma pada Kelas Edge Binary Image.....	31
IV.3.6	Algoritma pada Kelas Identifikasi Garis	32
IV.4	Perancangan Antarmuka.....	32
IV.4.1	Rancangan Tampilan.....	32
Bab V	Implementasi dan Pengujian	33
V.1	Implementasi Kelas	33
V.2	Implementasi Antarmuka.....	34
V.3	Pengujian	34
V.3.1	Skenario pengujian	34
V.3.2	Hasil Pengujian.....	35
Bab VI	Kesimpulan dan Saran.....	40
VI.1	,Kesimpulan	40

VI.2	Saran	40
	DAFTAR PUSTAKA.....	xv

DAFTAR GAMBAR

Gambar II-1. Proses pengenalan objek line pada image	5
Gambar II-2. Konversi ke Grayscale	6
Gambar II-3. Edge Detection	7
Gambar II-4. Contoh piksel sebelum dideteksi edge	8
Gambar II-5. Contoh piksel sesudah dideteksi edge	8
Gambar II-6. Koordinat polar	14
Gambar II-7. Grafik ruang Hough	15
Gambar II-8. Contoh hasil hough transform	15
Gambar II-9. Contoh garis pada lapangan sepak bola	16
Gambar II-10. Hasil akhir dari garis yang sudah dideteksi	16
Gambar II-11. Gambar lapangan sepak bola	18
Gambar III-1. Deskripsi umum sistem	19
Gambar III-2. Diagram use case	20
Gambar III-3. Analisis kelas	22
Gambar IV-1. Diagram interaksi <i>use case</i> pemutar video	24
Gambar IV-2. Diagram interaksi use case pemutar webcam	25
Gambar IV-3. Diagram interaksi use case ambil image per frame video	26
Gambar IV-4. Diagram interaksi use case konversi ke grayscale	26
Gambar IV-5. Diagram interaksi use case deteksi edge	27
Gambar IV-6. Diagram interaksi <i>use case</i> deteksi informasi garis	27
Gambar IV-7. Diagram kelas	28
Gambar IV-8. Rancangan tampilan	32
Gambar V-1. Deteksi garis dengan objek penggaris	35
Gambar V-2. Gambar garis putus-putus	35
Gambar V-3. Pergerakan frame pada video garis putus-putus	35
Gambar V-4. Gambar garis lurus berwarna	36
Gambar V-5. Pergerakan frame di video garis lurus satu warna pada lapangan sepakbola.	36

Gambar V-6. Garis lurus beda warna.....	37
Gambar V-7. Pergerakan frame di video garis lurus beda warna.....	37
Gambar V-8. Perbandingan hasil deteksi garis lurus.....	38
Gambar V-9. Pergerakan frame pada video pertandingan sepak bola.....	38
Gambar V-10. Image asli sebelum deteksi garis	39
Gambar V-11. Image asli setelah deteksi garis	39

DAFTAR TABEL

Tabel III-1. Spesifikasi analisis kelas.....	23
Tabel V-1. Implementasi kelas.....	33
Tabel V-2. Implementasi antarmuka	34

Bab I Pendahuluan

Pada bab ini akan dijelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan dan sistematika penulisan.

I.1 Latar Belakang

Teknologi informasi merupakan *icon* ilmu pengetahuan yang hangat dibicarakan akhir abad ini dan telah menyebar hampir di semua bidang tidak terkecuali di bidang olahraga sepak bola.

Sepak bola merupakan olahraga paling populer dan digemari bukan hanya di Indonesia, bahkan juga di dunia saat ini. Pada waktu piala dunia 2006 yang diselenggarakan di negara Jerman, media massa banyak sekali memberitakan tentang sepak bola seakan-akan sepak bola mengalahkan cabang olahraga lainnya. Sepak bola kini bukan lagi sekadar olahraga atau permainan. Ia telah menjadi komoditas yang tidak berbeda dengan produk lain yang beredar dalam sistem pasar. (*Media Indonesia, Minggu 15 Juni 2008*).^[1]

Setiap pertandingan sepak bola, dipimpin oleh seorang wasit yang memiliki kewenangan penuh untuk menegakkan peraturan permainan sehubungan dengan pertandingan dimana dia telah ditunjuk untuk itu, dimulai saat dia masuk sampai dia meninggalkan lapangan permainan. Wasit menjadi elemen paling penting karena segala keputusannya akan mempengaruhi jalannya pertandingan. Kadangkala dalam suatu pertandingan, terjadi kesalahpahaman yang di latar belakang oleh wasit yang tidak tepat dalam membuat keputusan. Seperti pada saat sebuah tendangan secara langsung ke arah gawang lawan dan menghasilkan gol, tetapi karena keterbatasan penglihatan wasit dianggap tidak gol. Hal ini tentu saja menimbulkan

^[1] http://eprints.undip.ac.id/1535/1/Hamdi_Zubaidi.pdf

kontroversi dan wacana untuk menerapkan teknologi pembantu dalam menyelesaikan persoalan ini.

Salah satu teknologi yang dapat digunakan untuk membuat aplikasi ini, yaitu software yang dinamakan *OpenCV*. *OpenCV* adalah singkatan dari *Open Computer Vision*, yaitu *library-library open source* yang di khususkan untuk melakukan *image processing*.

OpenCV itu sendiri merupakan software yang dapat mendeteksi sebuah objek melalui kamera atau webcam yang dikelola menggunakan komputer. Sehingga objek tersebut dapat dideteksi oleh program ini dengan menggunakan bahasa pemrograman *C#* pada *Microsoft Visual Studio 2010 Express Edition* sebagai editornya. Tujuannya adalah agar komputer mempunyai kemampuan yang mirip dengan cara pengolahan visual pada manusia.

Oleh karena itu, dibutuhkan tahapan-tahapan dalam pembuatan e-wasit ini antara lain yaitu memantau arah pergerakan bola, mengidentifikasi garis pada lapangan sepak bola, menentukan rule-rule pada garis dan bola, serta mengidentifikasi para pemain sepak bola. Selain pembuatan e-wasit juga ada alternatif lain dalam menangani masalah tentang garis yaitu dengan pembuatan sensor garis namun ada kelemahannya yaitu hanya bisa mengidentifikasi garis – garis yang ada namun tidak dapat menentukan arah pergerakan bola, rule-rule pada garis dan bola, dan mengidentifikasi para pemain sepak bola.

Tugas Akhir sebelumnya telah ada penelitian oleh Farah Lestari dan Erni Purnamasari (2010-2011) tentang identifikasi pada bola (lingkaran).^[2] Pada pembuatan e-wasit memerlukan beberapa tahapan mulai dari memantau arah pergerakan bola, mengidentifikasi garis, menentukan rule- rule pada garis dan bola,

^[2] Tugas Akhir Erni Purnamasari & Farah Lestari (2010-2011), “*Aplikasi Deteksi Posisi Bola Lingkaran Berbasis OpenCV.doc*”. Politeknik Negeri Batam.

serta mengidentifikasi para pemain sepak bola. Kemudian pada penelitian tugas akhir ini hanya pada tahapan mengidentifikasi garis yang berguna untuk mendeteksi garis-garis yang ada pada file video.

I.2 Rumusan Masalah

Rumusan masalah dalam aplikasi ini adalah:

1. Bagaimana mengidentifikasi garis dari file video.
2. Bagaimana menampilkan posisi garis yang telah diidentifikasi.

I.3 Batasan Masalah

Batasan masalah dalam aplikasi ini adalah:

1. Identifikasi garis dari file video.
2. Data yang diambil adalah file video.

I.4 Tujuan

Tujuan dari aplikasi ini adalah:

1. Mengidentifikasi garis dari file video.
2. Menampilkan posisi garis yang telah diidentifikasi.

I.5 Sistematika Penulisan

Adapun sistematika penulisan dalam aplikasi ini adalah:

Bab I Pendahuluan, berisi tentang penjelasan mengenai latar belakang masalah dalam proses penelitian, perumusan masalah, batasan masalah dalam proses penelitian, tujuan penelitian, dan sistematika penulisan yang memberikan gambaran isi laporan tugas akhir.

Bab II Tinjauan Pustaka, berisi mengenai studi literatur yang digunakan sebagai referensi dalam proses pembuatan aplikasi ini.

Bab III Analisis, berisi tentang hubungan antara perangkat keras dan perangkat lunak yang terdiri dari deskripsi umum sistem, fitur utama perangkat lunak, kebutuhan fungsional, diagram *usecase* beserta skenarionya serta analisis kelas.

Bab VI Perancangan, berisi tentang diagram interaksi, diagram kelas, rancangan kelas rinci dan perancangan antarmuka.

Bab V Implementasi dan pengujian, berisi mengenai proses implementasi hasil dari rancangan berupa produk jadi aplikasi deteksi garis berbasis *OpenCV*, yang akan dilakukan validasi apakah aplikasi sesuai dengan ketentuan atau tidak.

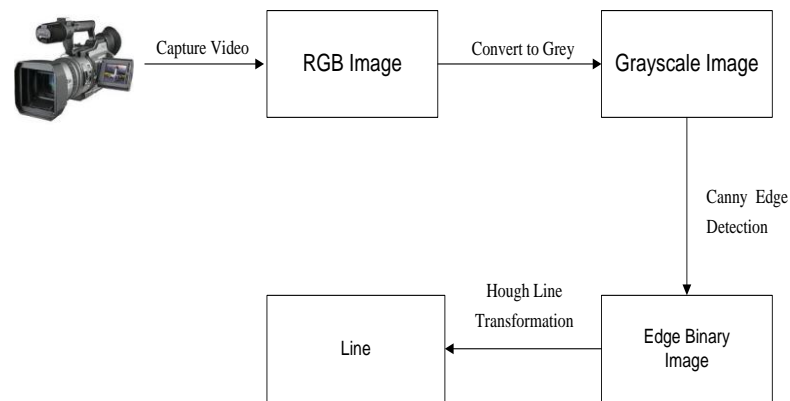
Bab VI Kesimpulan dan saran, berisi mengenai penyimpulan hasil dari proses implementasi dan pengujian pada bab sebelumnya, saran sebagai bahan pertimbangan untuk pengembangan penelitian selanjutnya.

Bab II Landasan Teori

Pada bab ini akan dijelaskan mengenai pengenalan objek garis pada image untuk setiap frame file video, hough line transform, pengenalan garis pada lapangan sepak bola, hubungan antar referensi.

II.1 Pengenalan Objek Garis pada Image untuk Setiap Frame File Video

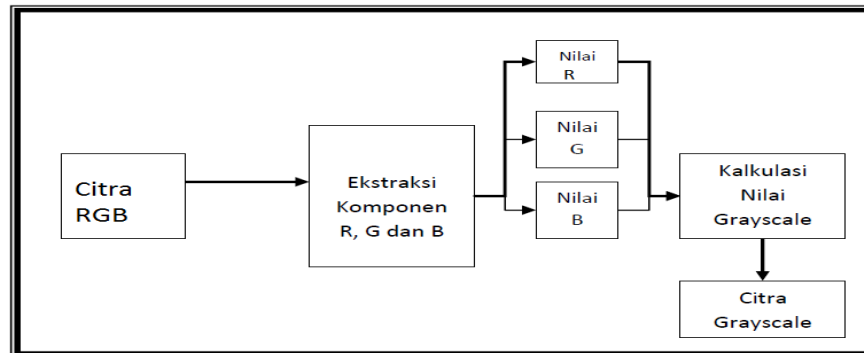
Ada beberapa keterangan dasar yang akan dijelaskan tentang proses image. Proses yang pertama yaitu mengambil image dari file video dengan format RGB. Image RGB yang diambil di konversi ke dalam format grayscale dimana hasilnya adalah hitam dan putih. Proses selanjutnya yaitu mendeteksi tepi image menggunakan metode Canny. Pada proses akhir deteksi menggunakan metode Hough Line Transform yang merupakan suatu teknik untuk menentukan lokasi suatu bentuk dalam citra. Untuk lebih jelas perhatikan gambar dibawah ini:



Gambar II-1. Proses pengenalan objek line pada image

II.1.1 Proses RGB ke Grayscale Image

Konversi citra dari format RGB ke grayscale sangat diperlukan untuk keperluan pemrosesan citra, diantaranya untuk memperkecil ukuran file dan untuk memenuhi syarat beberapa algoritma identifikasi yang memerlukan citra matriks tunggal.



Gambar II-2. Konversi ke Grayscale

Proses pertama adalah mengambil nilai R, G dan B dari suatu citra bertipe RGB. Pada tipe bmp citra direpresentasikan dalam 24 bit, sehingga diperlukan proses untuk mengambil masing-masing 3 kelompok 8 bit dari 24 bit tadi. Sebagai contoh suatu pixel memiliki nilai RGB 24 bit sebagai berikut :

111100001111000011111111, untuk mendapatkan masing-masing nilai R, G dan B dilakukan operasi-operasi sebagai berikut. Untuk mendapatkan nilai R dilakukan operasi modul dengan bilangan 256 sebagai berikut :

$$\begin{aligned}\text{Nilai R} &= 111100001111000011111111 \bmod 10000000 \\ &= 11111111\end{aligned}$$

Sedangkan untuk nilai G, dapat dicari dengan cara sebagai berikut:

$$\begin{aligned}\text{Nilai G} &= (111100001111000011111111 \text{ dan } 1111111100000000) / 100000000 \\ &= 11110000\end{aligned}$$

Untuk Nilai B, dapat dicari dengan menggunakan rumus

$$\begin{aligned} \text{Nilai B} &= (111100001111000011111111 \text{ dan } 111111110000000000000000) \\ &/1000000000000000000 \\ &= 11110000 \end{aligned}$$

Sehingga dari nilai pixel 1111000011110000111111112 atau 15790335 diperoleh nilai R = 11111111 = 255

$$G = 11110000 = 240$$

$$B = 11110000 = 240$$

Sehingga diperoleh triplet RGB = (255,240,240).

Setelah nilai triplet RGB kita peroleh, maka kita bisa mendapatkan nilai grayscale dari pixel tersebut. Ide dasarnya sebenarnya adalah membuat band tunggal dari 3 band RGB tadi dengan rumus tertentu. Pada penelitian ini digunakan rumus :

$$\text{red} = (\text{red} * 5) \setminus 10$$

$$\text{green} = (\text{green} * 8) \setminus 10$$

$$\text{blue} = (\text{blue} * 3) \setminus 10$$

$$\text{gray} = ((\text{red} + \text{green} + \text{blue}) * 10) \setminus 16$$

Dengan mengaplikasikan prosedur tadi pada semua pixel akan kita dapatkan citra dengan format grayscale.

II.1.2 Proses dari Grayscale ke Edge Binary Image

Kelebihan dalam menggunakan metode canny yaitu dalam mengekstrak tepian dengan kebebasan pemilihan parameter yang digunakan.^[3]

Contoh gambar :

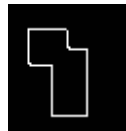


Gambar II-3. Edge Detection

^[3] <http://p3m.amikom.ac.id/p3m/51%20-%20KONVERSI%20FORMAT%20CITRA%20RGB%20KE%20FORMAT%20GRAYSCALE.pdf>



Gambar II-4. Contoh piksel sebelum dideteksi edge



Gambar II-5. Contoh piksel sesudah dideteksi edge

Dengan operator Canny diformulasikan dengan tiga tujuan utama, yaitu:

1. Deteksi optimal tanpa respons yang salah.
2. Lokalisasi yang baik dengan jarak minimal antara edge terdeteksi dan edge sebenarnya.
3. Respons tunggal untuk mengeliminasi respons ganda terhadap suatu edge.

Kriteria pertama bermaksud untuk mengurangi respons terhadap noise. Hal ini bisa dicapai dengan melakukan smoothing, dan canny adalah orang pertama yang menunjukkan bahwa Gaussian filtering dapat memberikan hasil yang optimal untuk tujuan ini.

Kriteria kedua bermaksud untuk meningkatkan akurasi, sehingga edge terdeteksi pada lokasi yang benar. Hal ini dapat dicapai dengan menerapkan proses non-maximum suppression, untuk mempertahankan edge yang berada pada peak/local maximum dan menghapuskan edge yang lain.

Kriteria ketiga berhubungan dengan lokasi suatu titik edge sebagai respons dari perubahan brightness, karena operator deteksi edge yang sebelumnya masih memberikan respons ganda terhadap suatu titik edge.

Algoritma Canny Edge Detection

Algoritma ini berjalan dalam 4 langkah sebagai berikut:

- Gaussian Smoothing

Langkah pertama dalam deteksi edge dengan menggunakan operator Canny adalah dengan melakukan smoothing terhadap citra dengan tujuan untuk mengurangi respons system terhadap noise serta melakukan control terhadap detail yang muncul pada citra edge.

- Konvolusi citra dengan suatu operator turunan pertama

Konvolusi citra dengan suatu operator turunan pertama dilakukan untuk memperoleh tingkat akurasi deteksi edge yang baik serta mengurangi kemungkinan terjadinya respons ganda terhadap suatu edge. Contoh turunan pertama diantaranya adalah Sobel, Prewitt, dan Robert.

- *Non-maxima Suppression*

Non-maxima Suppression atau *peak-detection* hanya mempertahankan titik-titik yang gradiennya merupakan local maxima dari suatu kumpulan data. Proses ini dilakukan dengan memanfaatkan informasi arah dari suatu edge.

Misalkan pada suatu daerah 3x3 suatu titik disebut maksimum bila gradien titik-titik pada kedua sisi titik tersebut lebih kecil dari gradien titik yang bersangkutan.

- Hysteresis Thresholding

Walaupun Gaussian Smoothing sudah dilakukan pada tahap pertama deteksi edge, citra output dari nn-maxima suppression masih mungkin mengandung potongan-potonga edge yang tidak benar karena adanya noise.

Solusi yang biasa digunakan untuk mengatasi masalah ini adalah thresholding, dimana semua edge yang bernilai dibawah nilai ambang (threshold) tertentu diabaikan. Tetapi dengan metode ini, penentuan nilai ambang yang tepat sulit dilakukan.

Akibatnya, masih akan ada edge yang salah (noise) terdeteksi jika nilai ambang terlalu rendah, atau edge yang hilang jika nilai ambang yang digunakan terlalu besar. Untuk mengatasi masalah tersebut, Canny menggunakan 2 nilai ambang, yaitu ambang atas dan ambang bawah.

Proses thresholding dimulai dengan titik yang nilainya lebih besar dari ambang atas. Titik ini dikenali sebagai edge, kemudian digunakan sebagai seed untuk tahap berikutnya. Dari seed tersebut lalu dilakukan analisa keterhubungan/linking, dimana 8 piksel tetangga dari seed dicek apakah nilainya melebihi ambang bawah atau tidak. Jika iya, maka piksel tersebut dikenali sebagai edge dan digunakan sebagai seed berikutnya. Proses ini dilakukan secara rekursif, sampai tidak ada lagi piksel yang memiliki nilai diatas nilai ambang bawah. Dengan demikian, hysteresis thresholding akan mengurangi kemungkinan adanya noise pada edge maupun hilangnya edge. ^[4]

II.1.3 Proses Mendeteksi Garis pada Edge Binary Image

Hough Transform (HT) merupakan suatu teknik untuk menentukan lokasi suatu bentuk dalam citra. HT dicetuskan pertama kali oleh P.V.C hough (1862), dilihat potensinya sebagai salah satu algoritma dalam pemrosesan citra oleh Rosenfeld (1969), kemudian diimplementasikan untuk mendeteksi garis dalam citra Duda (1972), dan sejak itu mengalami perkembangan yang sangat luas karena banyaknya keunggulan dan besarnya potensi untuk pengembangan lebih lanjut yang ditawarkan oleh algoritma ini. Salah satu keunggulannya adalah bahwa HT dapat memberikan hasil yang sama baiknya dengan algoritma *template matching*, tetapi lebih cepat. Dalam implementasinya, HT melakukan pemetaan terhadap titik-titik pada citra ke dalam parameter space (HT space) berdasar kan suatu fungsi yang mendefinisikan bentuk yang ingin dideteksi. HT umumnya digunakan untuk melakukan ekstrasi garis, lingkaran atau elips pada citra, namun dalam

[⁴] http://en.wikipedia.org/wiki/Canny_edge_detector

perkembangannya, HT juga telah dapat digunakan untuk melakukan ekstraksi bentuk-bentuk yang lebih kompleks.

II.2 Hough Line Transform

Hough transform adalah ekstraksi fitur teknik yang digunakan dalam analisis citra, visi komputer, dan pengolahan gambar digital. Tujuan dari teknik ini adalah untuk menemukan contoh sempurna dari objek di dalam kelas tertentu dari bentuk oleh prosedur pemungutan suara. Ini prosedur pemungutan suara yang dilakukan dalam ruang parameter, dari mana kandidat objek diperoleh sebagai maxima lokal dalam ruang akumulator disebut yang eksplisit dibangun oleh algoritma untuk menghitung transformasi Hough.

Hough klasik mengubahnya dengan identifikasi garis dalam gambar, namun kemudian mengubah Hough jadi diperpanjang untuk mengidentifikasi posisi bentuk sewenang-wenang, seperti lingkaran atau elips. Hough transform yang universal yang digunakan adalah diciptakan oleh Richard Duda dan Peter Hart pada tahun 1972, yang menyebutnya sebagai "Hough transform", setelah paten 1962 terkait Paulus Hough. Transformasi dipopulerkan di visi komputer masyarakat dengan Dana H. Ballard melalui artikel jurnal 1981 berjudul "Generalisasi Hough transform untuk mendeteksi bentuk berbeda".

Dalam analisis otomatis dari gambar digital, masalah sering muncul yaitu mendeteksi bentuk sederhana, seperti garis lurus, lingkaran atau elips. Dalam banyak kasus sebuah deteksi tepi dapat digunakan sebagai tahap pra-pengolahan untuk mendapatkan poin gambar atau piksel gambar yang diinginkan pada kurva dalam ruang gambar. Karena ketidaksempurnaan baik dalam data gambar atau deteksi tepi, dikarenakan ada poin yang hilang atau piksel pada kurva yang diinginkan serta deviasi spasial antara garis yang ideal / lingkaran / elips dan titik tepi yang banyak karena diperoleh dari detekso tepi.

Tujuan dari Hough transform adalah untuk mengatasi masalah ini dengan melakukan pengelompokan titik ujung ke objek kandidat dengan melakukan prosedur pemungutan suara yang eksplisit lebih dari satu set parameter objek gambar (Shapiro dan Stockman, 304).

Kasus paling sederhana dari transformasi Hough adalah transformasi linier untuk mendeteksi garis lurus. Dalam ruang gambar, garis lurus dapat digambarkan sebagai $y = mx + b$ dan dapat grafis untuk setiap pasangan titik gambar (x, y) . Dalam transformasi Hough, ide utama adalah untuk mempertimbangkan karakteristik garis lurus bukan sebagai poin gambar $(x_1, y_1), (x_2, y_2)$, dll, tetapi, dalam hal parameter, yaitu, parameter kemiringan m dan b parameter. Berdasarkan fakta bahwa, garis lurus $y = mx + b$ dapat direpresentasikan sebagai titik (b, m) di ruang parameter. Namun, salah satu masalah yang menimbulkan garis vertikal ke nilai tak terbatas dari m parameter dan b . Untuk alasan komputasi, oleh karena itu lebih baik menggunakan parameter yang berbeda, dinotasikan r dan θ (*theta*), untuk baris di Hough transform.

R Parameter mewakili jarak antara garis dan asal, sementara θ adalah sudut dari vektor dari titik asal ke titik terdekat (lihat Koordinat). Menggunakan parametrisasi ini, persamaan garis dapat ditulis sebagai berikut:

$$y = \left(-\frac{\cos \theta}{\sin \theta} \right) x + \left(\frac{r}{\sin \theta} \right)$$

yang dapat disusun kembali untuk $r = x \cos \theta + y \sin \theta$. (Shapiro dan Stockman, 304). Oleh karena itu mungkin untuk mengasosiasikan dengan setiap baris dari gambar pasangan (r, θ) yang unik jika $\theta \in [0, \pi)$ and, Atau jika $\theta \in [0, 2\pi)$ dan $r \geq 0$. (r, θ) kadang-kadang disebut sebagai *ruang Hough* untuk set garis-garis lurus dalam dua dimensi.

Representasi ini membuat Hough transform konseptual sangat dekat dengan dua dimensi transformasi Radon. (Dapat dilihat sebagai cara yang berbeda dalam memandang transformasi yang sama).

Untuk titik sembarang pada bidang gambar dengan koordinat, misalnya, (x_0, y_0) , garis yang melalui itu adalah $r(\theta) = x_0 \cdot \cos \theta + y_0 \cdot \sin \theta$ dimana r (jarak antara garis dan asal) ditentukan oleh θ .

Hal ini terkait dengan sinusoidal kurva di (r, θ) pesawat, yang unik ke titik itu. Jika kurva yang sesuai dengan dua poin dipindahkan, lokasi (di *ruang Hough*) di mana sesuai dengan garis (dalam ruang gambar asli) yang melewati kedua titik. Lebih umum, satu set poin yang membentuk garis lurus akan menghasilkan sinusoid pada parameter. Dengan demikian, masalah mendeteksi poin collinear dapat diubah menjadi masalah menemukan persamaan kurva.

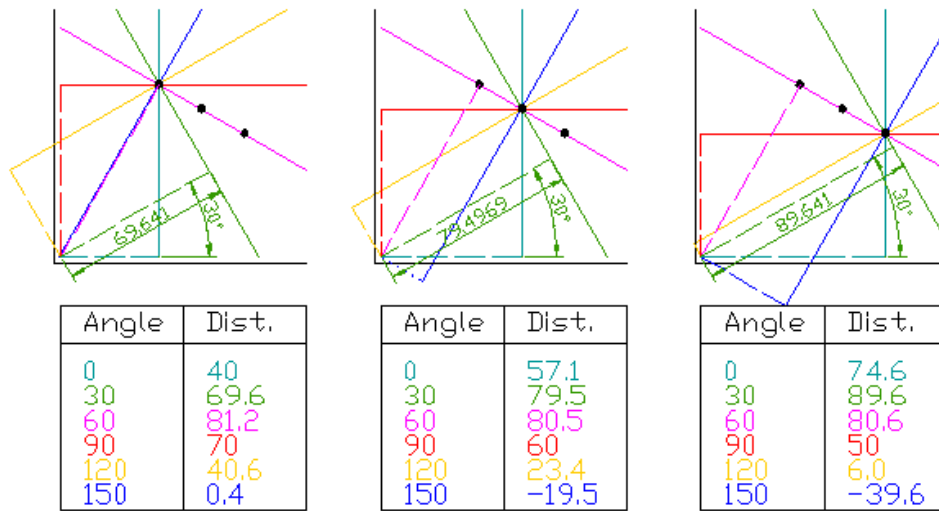
Algoritma Hough transform menggunakan sebuah array, disebut akumulator, untuk mendeteksi keberadaan garis $y = mx + b$. Para Dimensi akumulator sama dengan jumlah parameter tidak diketahui dari masalah transformasi Hough. Misalnya, masalah transformasi Hough linier memiliki dua parameter yang tidak diketahui: m dan b . Dua dimensi dari array akumulator akan sesuai dengan nilai-nilai terkuantisasi untuk m dan b . Untuk setiap pikselnya, algoritma transformasi Hough menentukan apakah ada tepi pada pixel yang. Jika demikian, itu akan menghitung parameter baris tersebut, dan kemudian mencari bin akumulator bahwa parameter ke dalam, dan meningkatnya nilai bin tersebut. Dengan menemukan nilai tertinggi, biasanya dengan mencari maximal lokal dalam ruang akumulator, garis kemungkinan besar dapat diekstraksi, dan (perkiraan) dan didefinisi geometris. (Shapiro and Stockman, 304).

Cara termudah untuk menemukan sudut adalah dengan menerapkan beberapa bentuk ambang batas, tapi teknik yang berbeda dapat menghasilkan hasil yang lebih baik dalam situasi yang berbeda - yang menentukan garis ditemukan serta berapa banyak. Karena garis tidak mengandung informasi panjang, seringkali yang diperlukan untuk menemukan bagian mana dari gambar yang cocok dengan garis-garis. Selain itu, karena kesalahan ketidaksempurnaan pada langkah deteksi tepi,

biasanya akan ada kesalahan dalam ruang akumulator, menemukan puncak yang tepat, dan dengan garis-garis yang sesuai.

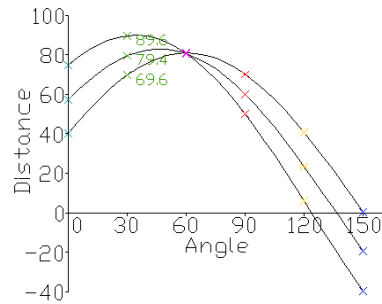
Contoh:

Pertimbangkan titik tiga data, ditampilkan sebagai titik hitam.



Gambar II-6. Koordinat polar

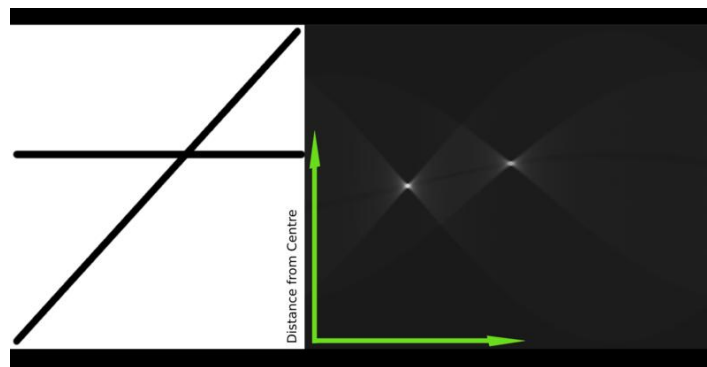
- Untuk setiap titik data, jumlah baris yang akan diplot melalui itu, semua pada sudut yang berbeda. Ini adalah ditampilkan di sini sebagai garis padat.
- Untuk setiap baris baris padat diplot yang tegak lurus untuk itu dan yang memotong asal . Ini adalah ditampilkan sebagai garis putus-putus.
- Panjang (yaitu tegak lurus jarak ke asal) dan sudut dari setiap garis putus-putus diukur. Dalam diagram di atas, hasilnya ditunjukkan pada tabel.
- Ini diulang untuk setiap titik data.
- Sebuah grafik dari panjang baris untuk setiap sudut, yang dikenal sebagai grafik ruang Hough.



Gambar II-7. Grafik ruang Hough

Titik di mana kurva berpotongan memberikan jarak dan sudut. Jarak dan sudut ini menunjukkan garis yang memotong titik-titik yang diuji. Dalam grafik yang ditunjukkan garis berpotongan pada titik ungu; ini sesuai dengan garis ungu solid dalam diagram di atas, yang melewati tiga poin.

Berikut ini adalah contoh yang berbeda menunjukkan hasil dari transformasi Hough pada gambar raster yang mengandung dua garis tebal. ^[5]



Gambar II-8. Contoh hasil hough transform

Hasil ini mengubah disimpan dalam matriks. Nilai sel merupakan jumlah kurva melalui titik apapun. Nilai-nilai sel yang lebih tinggi diberikan cerah. Dua titik terang yang jelas parameter Hough dari dua baris. Dari posisi tempat-tempat ini, sudut dan jarak dari pusat gambar dari dua baris dalam gambar input dapat ditentukan.

^[5] http://en.wikipedia.org/wiki/Hough_transform

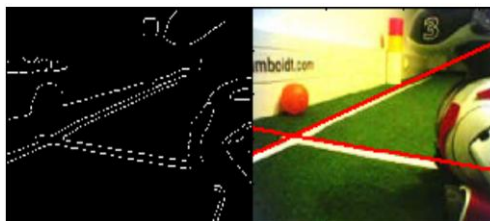
Contoh aplikasi hough transform pada robot sepak bola:

Beberapa penerapan transformasi hough dalam robot sepak bola. Hugh transformasi adalah sebuah metode untuk mendeteksi garis itu menyediakan beberapa informasi yang berguna untuk robot yang dapat digunakan dalam mengetahui lokalisasi dan garis lapangan misalnya dalam situasi yang ditunjukkan pada Gambar II.9.



Gambar II-9. Contoh garis pada lapangan sepak bola

Masalah utama dalam deteksi line di sini bukan ketika melakukan Hough Transform. Tetapi jauh lebih sulit menentukan berapa banyak garis dalam gambar dan apakah dua titik di ruang Hough milik garis yang sama atau tidak. Hal ini membutuhkan deteksi maximal lokal dalam ruang Hough dan clustering poin untuk menghindari mendeteksi yang sama lebih dari satu kali. Sehingga proses akhir diperlukan untuk dapat mendeteksi benda nyata. Dapat melihat pada gambar II.10 bahwa Hough Line Deteksi Algoritma dapat bekerja di RoboCup.^[6]



Gambar II-10. Hasil akhir dari garis yang sudah dideteksi

^[6] www.math.tau.ac.il/~turkel/notes/HoughTransform.pdf

II.3 Pengenalan Garis pada Lapangan Sepakbola

Lapangan sepak bola berbentuk persegi panjang, dengan perbandingan panjang dan lebar sama dengan dua banding satu ($P \times L = 2:1$), atau kurang sedikit. Sebagai contoh, misalnya panjang lapangan sepak bola tersebut adalah 90 meter, maka lebarnya harus berkisar 45 meter atau tidak lebih dari 55 meter.

Panjang lapangan sepak bola antara 90 meter (minimal) sampai 120 meter (maksimal). Lebar lapangan sepak bola antara 45 meter (minimal) sampai 90 meter (maksimal). Gawang dalam permainan sepak bola berbentuk persegi panjang dengan perbandingan 3:2. Ukuran ideal yang dilansir FIFA adalah lebarnya 7,3 meter dengan tinggi 2,4 meter.

Kotak penalti adalah area penjaga gawang bebas menyentuh bola dengan tangan. Kotak penalti adalah daerah rawan. Jika pemain lawan dilanggar dalam kotak penaltinya sendiri, maka tim lawan akan mendapat hadiah penalti, yaitu tendangan bebas berjarak 12 meter.

Pemain penendang hanya akan berhadapan dengan seorang penjaga gawang. Ada dua kotak penalti dalam lapangan sepak bola, yaitu:

- *Kotak penalti kecil (6-yard box)*

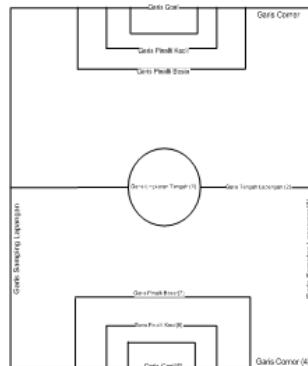
Kotak penalti kecil adalah area penjaga gawang yang mempunyai kekuasaan mutlak dan tidak boleh diganggu oleh pemain lawan. Kotak penalti kecil ini berada dalam kotak penalti besar dan berbentuk persegi panjang. Ukurannya adalah panjang 18 meter dengan lebar 5,5 meter.

- *Kotak penalti besar (18-yard box)*

Kotak penalti besar adalah area rawan, karena jika pemain lawan dilanggar dalam kotak penaltinya sendiri, maka tim lawan akan mendapat hadiah penalti. Panjangnya adalah 40-45 meter dengan lebar 16-19 meter. Garis lingkaran tengah (*Kick Off Area*) lapangan sepak bola berdiameter 9,15

meter. Garis lingkaran tersebut berada tepat di tengah lapangan. Di tengah lingkaran tersebut terdapat titik yang digunakan pemain untuk memulai *kick off*.^[7]

Garis - garis yang diakui pada lapangan bola :



Gambar II-11. Gambar lapangan sepak bola

Keterangan :

1. Garis Tengah Lapangan
2. Garis Lingkaran Tengah
3. Garis Samping Lapangan
4. Garis Corner
5. Garis Goal
6. Garis Pinalti Kecil
7. Garis Pinalti Besar

II.4 Hubungan Antar Referensi

Hubungan antar referensi untuk menyelesaikan Tugas Akhir ini yaitu proses pertama mengambil *image* dari file video dengan format RGB, kemudian format RGB dikonversi ke dalam format *grayscale* dimana hasilnya berupa *image* hitam putih, proses selanjutnya yaitu mendeteksi tepi *image* menggunakan metode *Canny*. Pada proses akhir yaitu untuk mendeteksi garis pada file video menggunakan metode *hough line transform* yang merupakan suatu teknik yang menentukan lokasi suatu bentuk dalam citra.

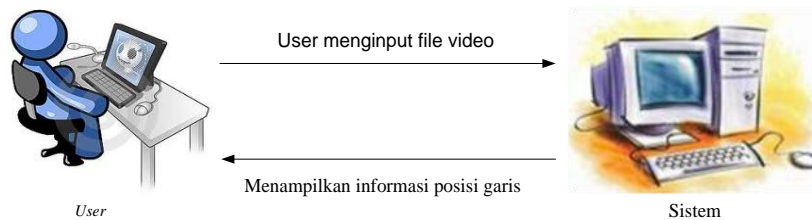
^[7]<http://talam.web44.net/?p=21>

Bab III Analisis

Pada bab ini akan dijelaskan mengenai deskripsi umum sistem, fitur utama perangkat lunak, use case dan analisis kelas.

III.1 Deskripsi Umum Sistem

Aplikasi ini akan menerima inputan data dari *user* berupa file video pertandingan sepak bola. Sistem akan mengidentifikasi posisi garis yang ada di lapangan sepak bola. Kemudian aplikasi akan menampilkan hasil posisi garis yang telah diidentifikasi.



Gambar III-1. Deskripsi umum sistem

Keterangan Gambar III.1 Deskripsi umum sistem:

1. User : Melakukan masukan data berupa file video pertandingan sepak bola.
2. Sistem : Menerima inputan data berupa file video pertandingan sepak bola dan akan mengidentifikasi posisi garis. Kemudian akan menampilkan hasil informasi garis yang telah diidentifikasi.

III.2 Fitur Utama

III.3 Perangkat Lunak

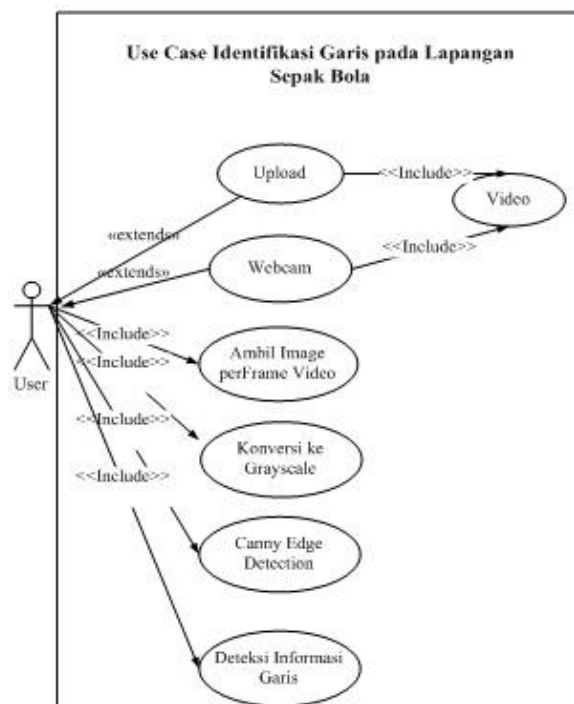
Aplikasi ini telah disesuaikan dengan kebutuhan pokok permasalahan yang telah diangkat. Aplikasi dapat digunakan untuk mengidentifikasi garis yang ada dilapangan sepak bola.

III.3.1 Kebutuhan Fungsional

- F-001 Sistem menerima inputan file video pertandingan sepak bola dari *user*
- F-002 Sistem mampu mengidentifikasi posisi garis
- F-003 Sistem akan menampilkan informasi posisi garis tersebut

III.4 Use Case

III.4.1 Diagram Use Case



Gambar III-2. Diagram use case

III.4.2 Skenario *Use Case*

III.4.2.1 Skenario *Use Case Upload File Video*

- Kondisi awal : File video belum di upload
- Kondisi akhir : Memainkan video
- Skenario : *User* dapat menginput file video pertandingan sepak bola melalui buka file video dengan menekan angka 1, maka akan muncul perintah masukkan nama file video pertandingan sepak bola dan video berputar dengan mengidentifikasi garis yang ada.

III.4.2.2 Skenario *Use Case Webcam*

- Kondisi awal : Video belum berputar
- Kondisi akhir : Video telah berputar
- Skenario : *User* dapat mengambil file video melalui buka webcame dengan menekan angka 2, maka image yang dianggap garis akan terdeteksi garis lurus secara otomatis.

III.4.2.3 Skenario *Use Case Ambil Image Per frame Video*

- Kondisi awal : Sistem mempunyai frame sebelumnya
- Kondisi akhir : Image dalam format RGB
- Skenario : Mengambil image berupa frame by frame pada video.

III.4.2.4 Skenario *Use Case Konversi ke Grayscale*

- Kondisi awal : Image dalam format RGB
- Kondisi akhir : Image dalam format Grayscale
- Skenario : Image RGB dikonversi kedalam format Grayscale oleh sistem.

III.4.2.5 Skenario *Use Case* Deteksi Edge

Kondisi awal : Image dalam format Grayscale

Kondisi akhir : Image telah dideteksi tepi

Skenario : Image Grayscale dikonversi dalam format Deteksi Edge oleh sistem.

III.4.2.6 Skenario *Use Case* Deteksi Informasi Garis

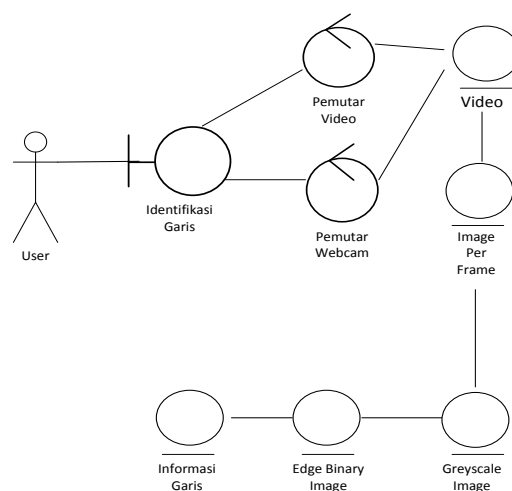
Kondisi awal : Image dalam format edge binary image, tidak ada informasi garis.

Kondisi akhir : Image dalam format edge binary image, ada informasi garis.

Skenario : Sistem melakukan deteksi tepi pada image edge dan menampilkan informasi garis yang telah terdeteksi.

III.5 Analisis Kelas

Berikut ini adalah analisis kelas yang dibuat berdasarkan identifikasi objek yang ada pada skenario *use case*.



Gambar III-3. Analisis kelas

Gambar III-3 menjelaskan tentang analisis kelas pada aplikasi ini. Adapun rincian dari kelas-kelas pada gambar tersebut dijelaskan pada tabel III-1, seperti berikut:

Tabel III-1. Spesifikasi analisis kelas

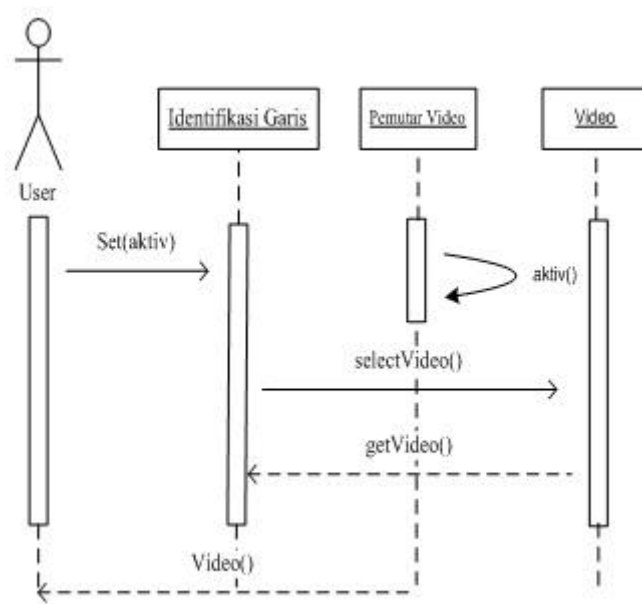
Jenis Kelas	Nama Kelas	Deskripsi
Kelas Boundary	Identifikasi	Kelas untuk menampilkan interface menu utama dari aplikasi.
	Garis	
Kelas Control	Pemutar Video	Kelas untuk mengambil file video melalui upload file video
	Pemutar Webcam	Kelas untuk memutar video melalui webcam.
Kelas Entity	Video	Kelas yang berisi file video pertandingan sepak bola
	Image per Frame	Kelas yang berisi frame-frame image.
	Image Grayscale	Kelas image yang berformat grayscale
	Edge Binary Image	Kelas image yang berformat edge binary image
	Informasi Garis	Kelas yang berisi informasi garis, yaitu start point dan end point

Bab IV Perancangan dan Implementasi

Pada bab ini akan dijelaskan mengenai diagram interaksi, diagram kelas dan rancangan kelas rinci, algoritma, perancangan antarmuka.

IV.1 Diagram Interaksi

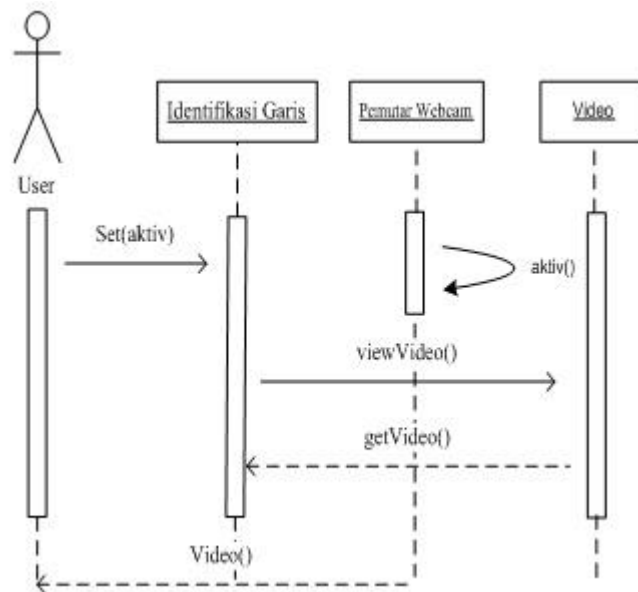
IV.1.1 Diagram Interaksi *Use Case* Pemutar Video



Gambar IV-1. Diagram interaksi *use case* pemutar video

Pada Gambar IV-1 menjelaskan proses memutar file video oleh *user* yang menggunakan upload file video. Pertama kali *user* membuka layar identifikasi garis, setelah aktif harus memilih file video, kemudian file video yang di pilih akan berputar dan menampilkan file video tersebut.

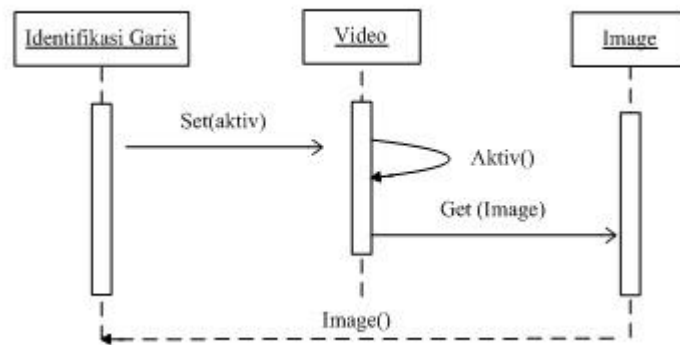
IV.1.2 Diagram Interaksi Use Case Pemutar Webcam



Gambar IV-2. Diagram interaksi use case pemutar webcam

Pada Gambar IV-2 menjelaskan proses memutar file video oleh *user* yang menggunakan pemutar webcam. *user* dapat membuka layar identifikasi garis, setelah aktif aplikasi akan menampilkan video menggunakan webcam, kemudian file video akan berputar dan menampilkan file video tersebut.

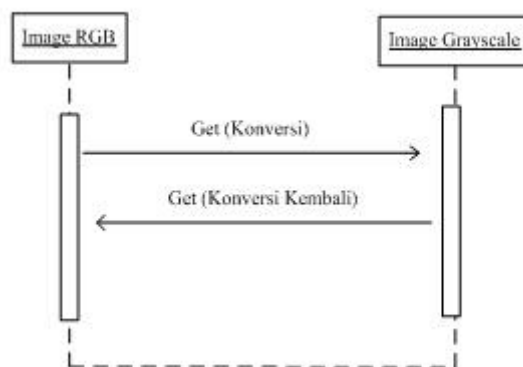
IV.1.3 Diagram Interaksi Use Case Ambil Image per Frame Video



Gambar IV-3. Diagram interaksi use case ambil image per frame video

Pada gambar IV-3 menjelaskan proses mengambil *image* per frame video yaitu kelas identifikasi garis akan mengatur video yang aktif dengan membuka video dari kelas video, kelas *image* akan memperoleh *image* per frame dari sebuah video dan selanjutnya akan menampilkan *image* yang telah dihasilkan.

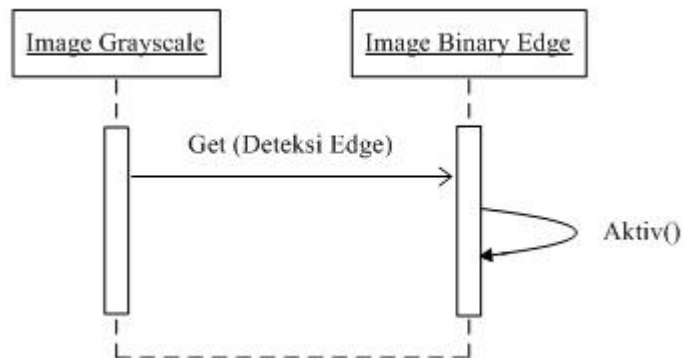
IV.1.4 Diagram Interaksi Use Case Konversi ke Grayscale



Gambar IV-4. Diagram interaksi use case konversi ke grayscale

Pada gambar IV-4 menjelaskan proses konversi *image* ke *grayscale* yaitu pada kelas *image* RGB di konversi ke *grayscale*, kemudian *image* yang telah dihasilkan di konversi kembali ke kelas *image* RGB.

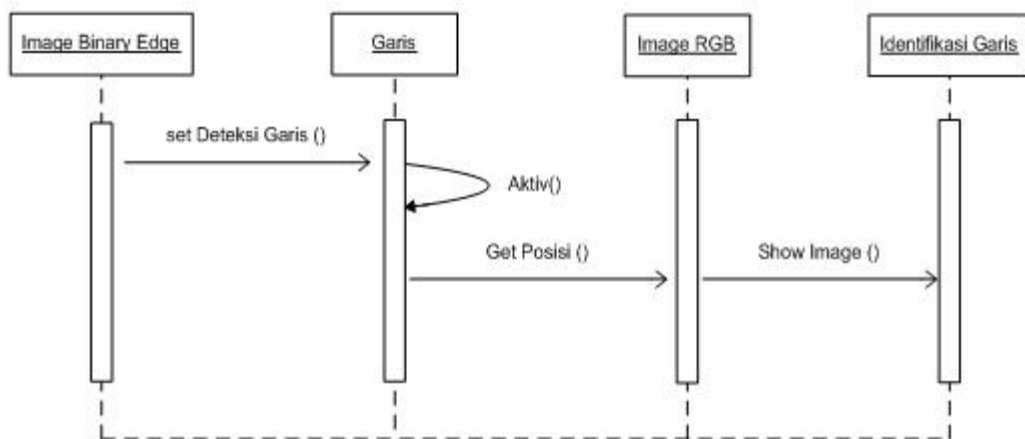
IV.1.5 Diagram Interaksi Use Case Deteksi Edge



Gambar IV-5. Diagram interaksi use case deteksi edge

Pada gambar IV-5 menjelaskan proses deteksi *edge* yaitu pada kelas *image grayscale* akan memperoleh deteksi tepi di kelas *image binary edge* yang aktif.

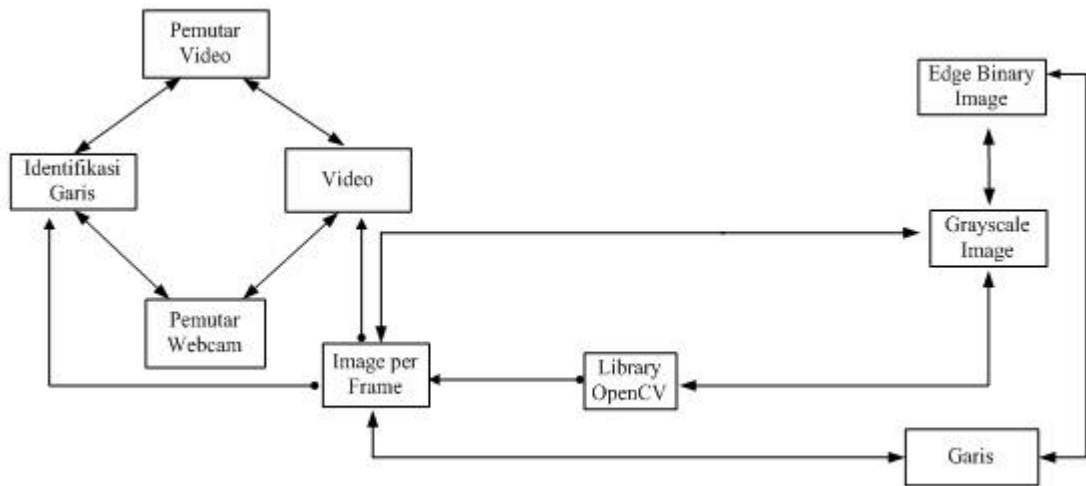
IV.1.6 Interaksi Diagram Use Case Deteksi Informasi Garis



Gambar IV-6. Diagram interaksi use case deteksi informasi garis

Pada gambar IV-6 menjelaskan proses deteksi garis yaitu pada kelas *Image binary edge* akan menetapkan garis yang telah di deteksi dari kelas *garis*, sehingga akan dapat menentukan posisi garis dari kelas *image RGB*. Kemudian *image* garis yang telah terdeteksi akan di tampilkan pada kelas layar menu.

IV.2 Diagram Kelas



Gambar IV-7. Diagram kelas

IV.2.1 Rancangan Kelas Rinci

Rancangan rinci setiap kelas disajikan pada gambar dibawah ini:

IV.2.1.1 Kelas Pemutar Video

Pemutar Video		
+ set(aktif)	:	Void
+ selectVideo()	:	Void
+ getVideo()	:	Void

IV.2.1.2 Kelas Pemutar Webcam

Pemutar Webcam		
+ set(aktif)	:	Void
+ selectVideo()	:	Void
+ showVideo()	:	Void

IV.2.1.3 Kelas Video

Video		
- Frame size	:	Int
+Set (Aktiv)	:	Void
+Show (Image)	:	Void

IV.2.1.4 Kelas Image per Frame

Image per Frame		Extends from OpenCV.image
- Size	:	Int
- Height	:	Int
- Width	:	Int
- PropertyItem	:	Int
- Format Pikel	:	Teks
+ Get (Konversi)	:	Void
+ Get (Posisi)	:	Void

IV.2.1.5 Kelas Grayscale

Grayscale Image		Extends from OpenCV.image
- Size	:	Int
- Height	:	Int
- Width	:	Int
- PropertyItem	:	Int
- Format Pikel	:	Teks
+ Get (Konversi)	:	Void

IV.2.1.6 Kelas Deteksi Edge

Edge Binary Image		Extends from OpenCV.image
- Size	:	Int
- Height	:	Int
- Width	:	Int
- PropertyItem	:	Int
- Format Pikel	:	Teks
+ Get (Deteksi Edge)	:	Void

IV.2.1.7 Kelas Deteksi Informasi Garis

Informasi Garis		
- Radius	:	Int
- Titik Pusat	:	Point
+ Get (Deteksi Edge)	:	Void

IV.3 Algoritma

IV.3.1 Algoritma pada Kelas Pemutar Video

Initial State : File belum terisi Final State : File telah terisi
If buka file video Than masukkan nama file video Else Sistem tidak menghasilkan apa-apa

IV.3.2 Algoritma pada Kelas Pemutar Webcam

```
Initial State : webcam belum terbuka  
Final State  : webcam telah terbuka
```

```
If buka webcam  
Than webcam terbuka  
Else  
Sistem tidak menghasilkan apa-apa
```

IV.3.3 Algoritma pada Kelas Image per Frame

```
Initial State : image per frame  
Final State  : menampilkan video
```

```
CvCapture *capture  
  
IplImage *frame  
  
frame = cvQueryFrame(capture);  
  
cvShowImage
```

IV.3.4 Algoritma pada Kelas Grayscale

```
Initial State : Video RGB  
Final State  : Video Grayscale (hitam putih)
```

```
IplImage* gray = cvCreateImage  
  
(cvGetSize(frame)  
  
cvCvtColor( frame, gray, CV_BGR2GRAY )
```

IV.3.5 Algoritma pada Kelas Edge Binary Image

```
Initial State : Video Graycle (hitam putih)  
Final State  : Video telah di deteksi tepi
```

```
IplImage* color_dst = cvCreateImage  
  
(cvGetSize(gray)  
  
cvCanny( gray, color_dst, 70, 200, 3 );
```

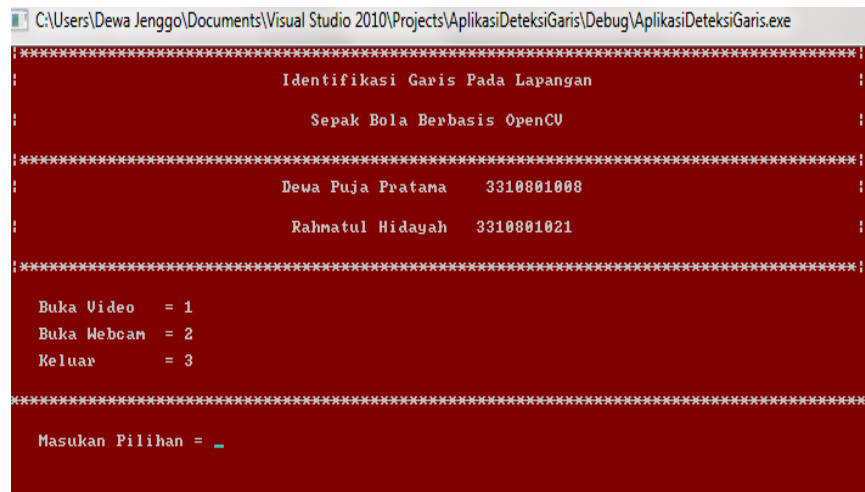
IV.3.6 Algoritma pada Kelas Identifikasi Garis

Initial State : Video Garis RGB
Final State : Video Garis RGB telah terdeteksi

- Buat 1 garis
- Ambil 2/3 titik sembarang pada garis
- Cari persamaan garis untuk ruang parameter
- Buat titik temu dari persamaan garis tersebut

IV.4 Perancangan Antarmuka

IV.4.1 Rancangan Tampilan



```
C:\Users\Dewa Jenggo\Documents\Visual Studio 2010\Projects\AplikasiDeteksiGaris\Debug\AplikasiDeteksiGaris.exe
*****
:                               Identifikasi Garis Pada Lapangan                               :
:                               Sepak Bola Berbasis OpenCV                               :
:*****
:                               Dewa Puja Pratama   3310801008                               :
:                               Rahmatul Hidayah   3310801021                               :
:*****
:
: Buka Video   = 1
: Buka Webcam  = 2
: Keluar       = 3
:
:*****
:
: Masukkan Pilihan = _
```

Gambar IV-8. Rancangan tampilan

Bab V Implementasi dan Pengujian

Setelah dilakukan tahap perancangan maka tahap selanjutnya adalah implementasi dan pengujian pada perangkat lunak. Implementasi akan menghasilkan aplikasi yang dapat dijalankan di lingkungan operasional. Untuk mengetahui apakah aplikasi tersebut dapat melakukan fungsi sesuai deskripsi perencanaan maka perlu dilakukan pengujian. Pada perangkat keras akan dilakukan pengukuran dan pengujian untuk mengetahui kinerja sistem.

V.1 Implementasi Kelas

Implementasi kelas berisi tentang nama kelas, nama file fisik dan nama file executable, implementasi kelas disajikan pada tabel V.I.

Tabel V-1. Implementasi kelas

No	Nama Kelas	Nama File Fisik	Nama File Executable
1	Pemutar Video	cvCreateFileCapture.cpp	cvCreateFileCapture.lib
2	Pemutar Webcam	cvCaptureFromCam.cpp	cvCaptureFromCam.lib
3	Video	cvQueryFrame.cpp	cvQueryFrame.lib
4	Image perFrame	IplImage.cpp	IplImage.lib
5	Image Grayscale	Iplgray.cpp	Iplgray.lib
6	Edge Binary Image	cvCanny.cpp	cvCanny.lib
7	Informasi Garis	cvHoughLines2.cpp	cvHoughLines2.lib

V.2 Implementasi Antarmuka

Implementasi antarmuka dari aplikasi Identifikasi Garis Pada Lapangan SepakBola ini dapat dilihat pada table V-2.

Tabel V-2. Implementasi antarmuka

No	Antar Kelas	Nama File Fisik	Nama File Executable
1	Layar Menu	AplikasiDeteksiGaris.cpp	VisualStudio2010\Projects\AplikasiDeteksiGaris\Debug\AplikasiDeteksiGaris.exe

V.3 Pengujian

V.3.1 Skenario pengujian

Pada tahapan yang dilakukan dalam pengujian ini yaitu:

Langkah pertama yang dilakukan yaitu persiapan file video yang mempunyai image garis. File video berisi pergerakan penggaris secara horizontal atau vertikal dengan warna penggaris yang berbeda dengan latar. Jika pergerakan penggaris mendekati kamera maka ukuran penggaris yang terdeteksi akan membesar, ketika penggaris menjauh dari kamera maka ukuran penggaris yang terdeteksi akan mengecil.

Langkah kedua, pengujian dengan merekam 3 macam garis yaitu garis putus-putus, garis berwarna dan garis beda warna.

1. Pada pendeteksian garis putus-putus, merekam garis putus-putus berwarna hitam dengan background putih, garis terdeteksi sesuai dengan posisi garis.
2. Pada pendeteksian garis lurus satu warna, merekam garis lurus dengan warna putih dengan background hijau, garis terdeteksi sesuai dengan posisi garis.
3. Pada pendeteksian garis lurus beda warna, merekam garis lurus dengan warna yang berbeda yaitu hitam dan merah dengan background putih, garis terdeteksi sesuai dengan posisi garis.

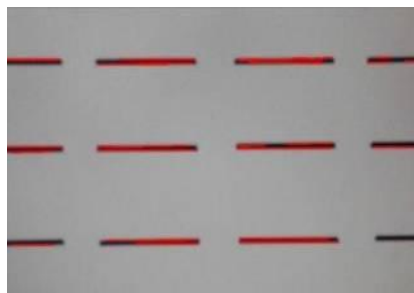
V.3.2 Hasil Pengujian

- Deteksi garis menggunakan webcam dengan mengambil objek penggaris



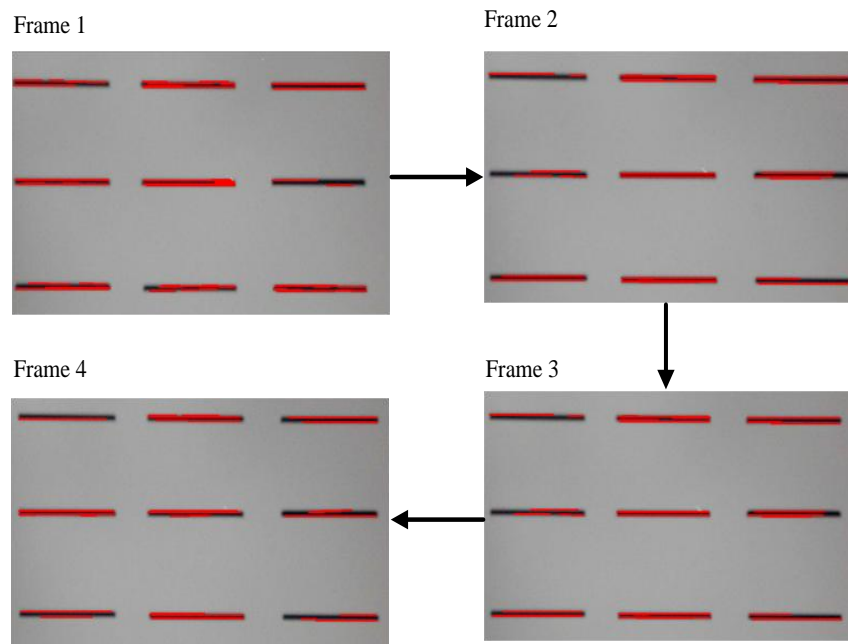
Gambar V-1. Deteksi garis dengan objek penggaris

- Deteksi garis putus-putus



Gambar V-2. Gambar garis putus-putus

Hasil dari pergerakan frame pada video garis putus-putus



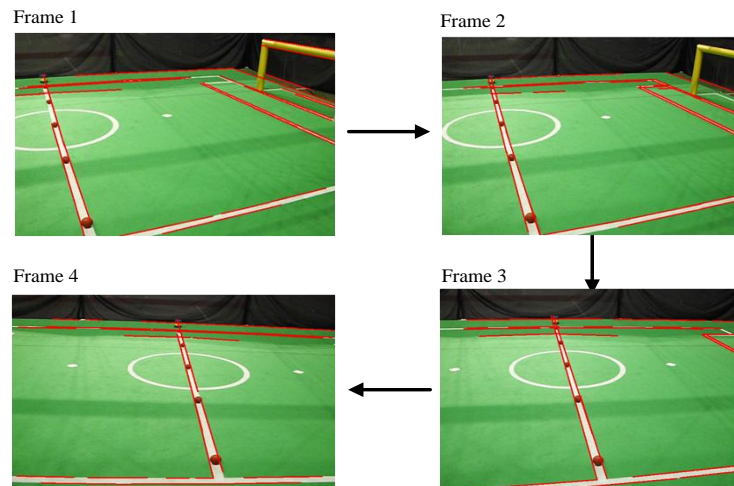
Gambar V-3. Pergerakan frame pada video garis putus-putus

- Deteksi garis lurus satu warna



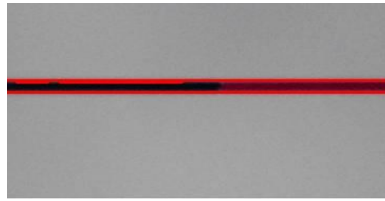
Gambar V-4. Gambar garis lurus berwarna

Hasil dari pergerakan frame di video garis lurus satu warna pada lapangan sepakbola.



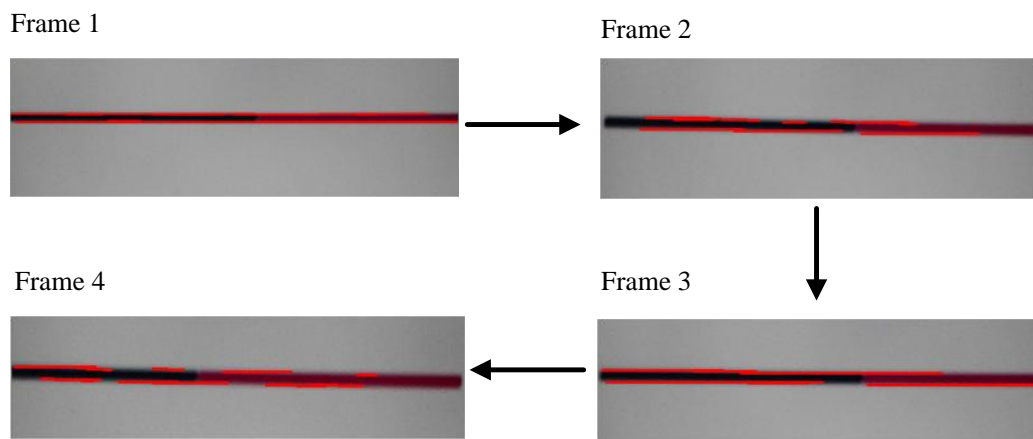
Gambar V-5. Pergerakan frame di video garis lurus satu warna pada lapangan sepakbola.

- Deteksi garis lurus beda warna



Gambar V-6. Garis lurus beda warna

Hasil dari pergerakan frame di video garis lurus beda warna



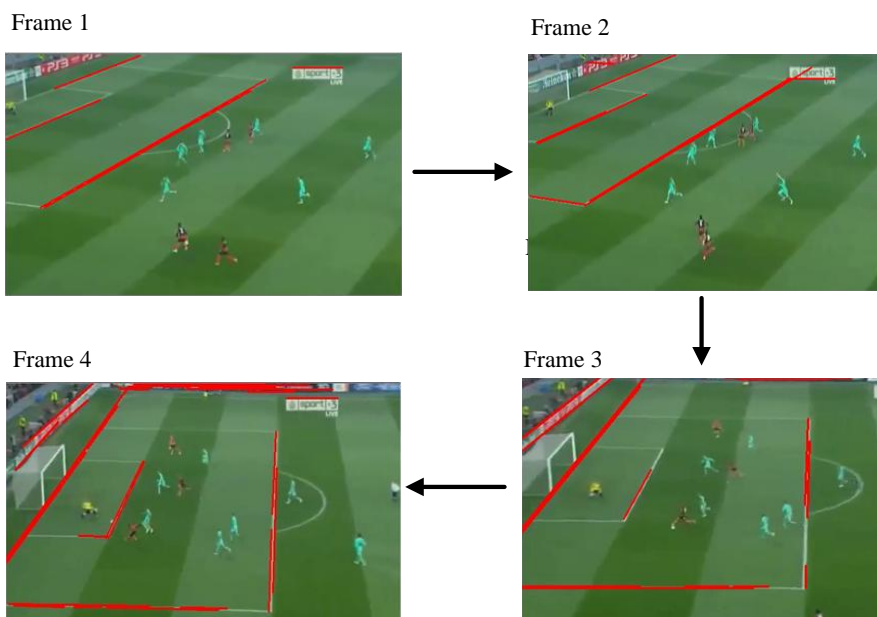
Gambar V-7. Pergerakan frame di video garis lurus beda warna

Perbandingan hasil deteksi garis lurus pada pertandingan sepak bola



Gambar V-8. Perbandingan hasil deteksi garis lurus

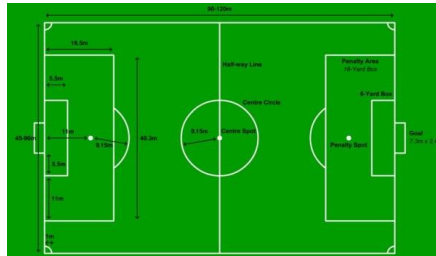
Hasil dari pergerakan frame pada video pertandingan sepak bola



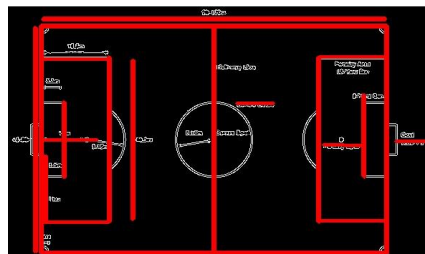
Gambar V-9. Pergerakan frame pada video pertandingan sepak bola

Dari gambar hasil pengujian diatas menunjukan bahwa aplikasi sudah benar dapat menampilkan posisi garis. Pada deteksi garis putus-putus, garis lurus satu warna dan garis lurus beda warna sistem secara keseluruhan dapat mendeteksi garis yang ada. Namun pada saat tertentu deteksi garis bisa tidak stabil disebabkan oleh perputaran frame pada image video yang cepat.

Eksekusi aplikasi dilakukan dengan menggunakan rekaman video dari lapangan sepak bola. Tujuannya agar aplikasi bisa mendeteksi garis-garis yang ada pada lapangan sepak bola tersebut. Namun pada saat tertentu deteksi garis bisa tidak stabil disebabkan oleh perputaran frame pada image video yang cepat.



Gambar V-10. Image asli sebelum deteksi garis



Gambar V-11. Image asli setelah deteksi garis

Eksekusi aplikasi dilakukan dengan menggunakan image garis lurus. Tujuannya sebagai bukti bahwa aplikasi dapat mendeteksi garis-garis yang ada secara stabil, karena tidak adanya perputaran frame seperti pada video. Dapat dilihat pada gambar V- 11.

Bab VI Kesimpulan dan Saran

VI.1 ,Kesimpulan

Setelah di lakukan tahap analisis, perancangan, implementasi dan pengujian pada aplikasi Identifikasi Garis pada Lapangan Sepakbola Berbasis OpenCV, maka kesimpulan yang didapat adalah:

1. Aplikasi mendeteksi garis lurus pada file video dalam keadaan tidak stabil, dan identifikasi garis lurus pada image dalam keadaan stabil.
2. Aplikasi dapat mengidentifikasi garis lurus menggunakan file video (berupa garis lurus, garis putus-putus dan garis lurus beda warna).
3. Aplikasi dapat mengidentifikasi garis lurus menggunakan webcam dengan contoh objek penggaris.
4. Dapat mendeteksi garis lurus dengan mengkonversi video ke grayscale, deteksi tepi dan menampilkan informasi garis dengan format RGB.

VI.2 Saran

Penyusun menyadari masih terdapat kekurangan-kekurangan dalam aplikasi Identifikasi Garis pada Lapangan Sepakbola Berbasis OpenCV ini. Oleh karena itu, aplikasi ini dapat dikembangkan kembali. Adapun saran yang dapat diberikan untuk penyempurnaan dari aplikasi ini adalah:

1. Diharapkan aplikasi ini dapat mendeteksi garis pada file video secara stabil.
2. Diharapkan aplikasi ini dapat dikembangkan untuk penyempurnaan deteksi garis pada lapangan sepakbola.

DAFTAR PUSTAKA

1. Zubaidi, Hamdi, “*Landasa Program Perencanaan dan Perancangan Arsitektur Stadion Utama Gelora Bung Tomo di Surabaya*”, (2009, April). Tersedia: http://eprints.undip.ac.id/1535/1/Hamdi_Zubaidi.pdf, diakses pada tanggal 23 Februari 2011.
2. Tugas Akhir Erni Purnamasari & Farah Lestari, “*Aplikasi Deteksi Posisi Pola Lingkaran Berbasis OpenCV.doc*”, (2010-2011). Politeknik Negeri Batam, diakses pada tanggal 23 Februari 2011.
3. Evan, “*Studi Digital Watermarking Citra Bitmap dalam Mode Warna Hue Saturation Lightness*”, (2009). Tersedia: <http://www.scribd.com/doc/51538669/MakalahIF30581-2009-a014>, diakses pada tanggal 20 Januari 2011.
4. Al Fatta, Hanif, “*Konversi Format Citra RGB ke Format Grayscale Menggunakan Visual Basic*”, (2007, Nov 24). Tersedia: <http://p3m.amikom.ac.id/p3m/51%20%20KONVERSI%20FORMAT%20CITRA%20RGB%20KE%20FORMAT%20GRAYSCALE.pdf>, diakses pada tanggal 09 Maret 2011.
5. Wikipedia The Free Encyclopedia, “*Canny Edge Detector*”. Tersedia: http://en.wikipedia.org/wiki/Canny_edge_detector, diakses pada tanggal 10 Maret 2011.
6. Departmen Of Computer Engineering University of California at Santa Cruz, “*Hough Transform for Lines and Curves*”. Tersedia: <http://www.docstoc.com/docs/47349074/hought-transform-for-lines-and-Curves.pdf>, diakses pada tanggal 31 Maret 2011.
7. Wojciech Wojcikiewicz, “*Hough Transform Line Detection in Robot Soccer*”. Tersedia : www.math.tau.ac.il/~turkel/notes/HoughTransform.pdf, diakses pada tanggal 04 April 2011.

8. Ilham Gustian, "*Ukuran Lapangan Sepak Bola*". Tersedia: <http://talam.web44.net/?p=21>, diakses pada tanggal 23 Februari 2011.
9. Budiharto Widodo, "*Robotika Teori + Implementasi*", (2010). Penerbit: Andi. Yogyakarta. Hal 130.
10. ITS, "*Teori Penunjang*". Tersedia: <http://digilib.its.ac.id/public/ITS-NonDegree-7518-74050470036-bab2.pdf>, diakses pada tanggal 28 Februari 2011.
11. Wikipedia The Free Encyclopedia "*Hough Line Transform*" Tersedia: http://en.wikipedia.org/wiki/Hough_transform diakses pada tanggal 19 Agustus 2011.