

**DETEKSI GARIS LAPANGAN SEPAKBOLA ROBOT DENGAN METODE  
*HOUGH TRANSFORM* DAN *SSD MOBILENET V2***

**SKRIPSI**

**ADE RIZKY**

**171402134**



**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

DETEKSI GARIS LAPANGAN SEPAKBOLA ROBOT DENGAN METODE  
*HOUGH TRANSFORM* DAN *SSD MOBILENET V2*  
SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah Sarjana  
Teknologi Informasi

ADE RIZKY  
171402134



PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024

## PERSETUJUAN

Judul : Deteksi Garis Lapangan Sepakbola Robot Dengan  
Metode *Hough Transform* dan *Ssd Mobilenet v2*  
Kategori : Skripsi  
Nama Mahasiswa : Ade Rizky  
Nomor Induk Mahasiswa : 171402134  
Program Studi : Sarjana (S-1) Teknologi Informasi  
Fakultas : Ilmu Komputer dan Teknologi Informasi  
Universitas Sumatera Utara

Medan, 11 Juli 2024

Komisi Pembimbing

Pembimbing 2,

Dedy Arisandi S.T., M.Kom

NIP. 197908312009121002

Pembimbing 1,

Seniman S.Kom., M.Kom

NIP. 198705252014041001

Diketahui/disetujui oleh

Program Studi SI Teknologi Informasi

Ketua,

Dedy Arisandi S.T., M.Kom

NIP. 197908312009121002

**PERNYATAAN**

DETEKSI GARIS LAPANGAN SEPAKBOLA ROBOT DENGAN METODE  
*HOUGH TRANSFORM* DAN *SSD MOBILENET V2*

**SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 11 Juli 2024

Ade Rizky

171402134

## UCAPAN TERIMA KASIH

Puji syukur kita panjatkan kepada Allah Yang Maha Kuasa atas kesehatan, rahmat, dan petunjuk-Nya yang telah memungkinkan penulis menyelesaikan skripsi ini dengan sebaik-baiknya. Shalawat dan salam semoga tercurahkan kepada Nabi Muhammad SAW, penutup para nabi, yang syafaatnya kita harapkan di akhirat kelak.

Skripsi ini disusun sebagai bagian dari penyelesaian pendidikan penulis dalam Program Studi Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara, sebagai syarat untuk memperoleh gelar Sarjana Komputer. Skripsi ini berjudul "Deteksi Garis Lapangan Sepakbola Robot Dengan Metode *Hough Transform*".

Penulis ingin menyampaikan penghargaan yang tulus kepada semua pihak yang telah mendukung dalam perjalanan perkuliahan dan penyusunan skripsi ini. Tanpa dukungan dan doa dari rekan-rekan, penulis merasa sulit untuk menyelesaikan skripsi ini. Terima kasih kepada:

1. Kedua orang tua penulis, Herman dan Wartiani selaku orang tua yang selalu memberikan kasih sayang, doa, nasehat, serta kesabarannya dalam setiap langkah hidup penulis.
2. Kakak penulis, Suci Heriani yang selalu memberikan dan membantu penulis dalam menyelesaikan skripsi.
3. Bapak Prof. Dr. Muryanto Amin, S.Sos., M.Si selaku Rektor Universitas Sumatera Utara.
4. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc. sebagai Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
5. Bapak Dedy Arisandi ST., M.Kom. sebagai Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
6. Bapak Ivan Jaya, S.T., M.Kom. sebagai Sekretaris Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
7. Bapak Seniman S.Kom., M.Kom. sebagai Dosen Pembimbing I dan Bapak Dedy Arisandi ST., M.Kom. sebagai Dosen Pembimbing II, yang telah memberikan bimbingan dan saran yang berharga.

8. Dosen Penguji I dan Dosen Penguji II yang memberikan masukan penting dalam menyempurnakan skripsi ini.
9. Semua tenaga pengajar dan staf dari Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara atas ilmu yang telah diberikan.
10. Teman-teman di PCS yaitu Fajar, Ibnu, Jackie, Rafif, Rizky, Dinul, Fakhri, Gilbert, Muharis, Preston, Rafid, Rezky, Prima dan Haqi yang menemani selama di kontrakan, masa perkuliahan hingga masa penulisan Tugas Akhir perkuliahan.
11. Teman-teman satu bimbingan, Yonadab, Alfi, dan Maulana, yang memberikan dukungan dan saran selama proses penyusunan skripsi.
12. Partner, Sahabat, serta teman-teman dekat, Muhammad Ulwan Azmi, Deo Pranata Silitonga, Muhammad Farras Siraj Polem, dan Taufik Baskoro, Nabila Sagita, Lisa, Safrizal, Fikri, Adrian, dan Ravel yang turut serta dalam perjalanan perkuliahan.
13. Teman-teman angkatan 2017 Teknologi Informasi yang telah membantu dan berjuang bersama penulis dalam menghadapi perkuliahan.
14. Semua individu yang telah membantu dan mendampingi penulis selama proses penelitian ini.

Semoga Allah Yang Maha Esa memberkahi semua individu yang terlibat dalam dukungan, doa, dan bantuan kepada penulis dalam menyelesaikan skripsi ini.

Medan, 11 Juli 2024

Penulis

## ABSTRAK

Dalam memainkan pertandingan robot sepak bola beroda dengan baik, sebuah robot minimal harus memiliki beberapa kemampuan dasar seperti mengikuti pergerakan bola, menggiring bola, dan menendang bola kearah gawang lawan. Untuk mencapai hal itu maka robot harus bisa mendeteksi empat buah komponen dasar dalam permainan sepak bola, salah satu hal terpenting adalah mendeteksi garis-garis lapangan yang menjadi pembatas. Untuk itu dilakukan penelitian deteksi garis lapangan pada citra permainan sepak bola RoboCup Soccer SPL dengan mengimplementasikan metode *Hough Transform*. Dataset yang digunakan dalam penelitian ini adalah *Robocup SPL Instance Segmentation Dataset* sebanyak 987 citra uji. Hasil penelitian ini menunjukkan bahwa Metode *Hough Transform* dapat melakukan pendeteksian garis pada citra RoboCup Soccer SPL dengan implementasi pada sistem. Deteksi garis pada citra permainan sepak bola RoboCup Soccer SPL menunjukkan nilai akurasi pendeteksian garis sebesar 96%.

Kata Kunci : Deteksi Garis, Metode Hough Transform, RoboCup Soccer SPL, Touch Line.

**DETECTION OF ROBOTS FOOTBALL FIELD LINES USING HOUGH  
TRANSFORM AND SSD MOBILENET V2 METHOD**

**ABSTRACT**

*When playing a wheeled soccer robot match properly, a robot must at least have some basic skills such as following the movement of the ball, dribbling, and kicking the ball towards the opponent's goal. To achieve this, the robot must be able to detect the four basic components in a soccer game, one of the most important things is detecting the field lines that are the boundaries. For this reason, a field line detection study was carried out in the image of a RoboCup Soccer SPL soccer game by implementing the Hough Transform method. The dataset used in this research is the Robocup SPL Instance Segmentation Dataset with 987 test images. The results of this study indicate that the Hough Transform method can perform line detection in RoboCup Soccer SPL images with implementation on the system. Line detection in the image of the RoboCup Soccer SPL soccer game shows a line detection accuracy value of 96%.*

Keywords: Line Detection, Hough Transform Method, RoboCup Soccer SPL, Touch Line.



## DAFTAR ISI

PERSETUJUAN	ii
PERNYATAAN	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	4
1.4 Batasan Masalah	4
1.5 Manfaat Penelitian	4
1.6 Metodologi Penelitian	4
1.7 Sistematika Penulisan	5
BAB 2 LANDASAN TEORI	7
2.1 Region Based Image Segmentation	7
2.2 Gaussian Blur	8
2.3 Bitwise Operation	8
2.4 Citra Digital	9
2.4.1. Representasi Citra Warna	10
2.4.2. Representasi Citra Keabuan	10
2.4.3. Representasi Citra Biner	11
2.5 Deteksi Tepi	12
2.6 Transformasi Hough	14
2.7 OpenCV	16
2.8 Single Shot Multibox Detector (SSD)	18
2.9 MobileNet	18
2.10 Penelitian Terkait	20

BAB 3 ANALISIS DAN PERANCANGAN	22
3.1    Metode Penelitian	22
3.1.1.    Metode Hough Transform	22
3.1.2.    Metode SSD MobileNet V2	23
3.2    Arsitektur Umum	23
3.3    Proses Penelitian	24
3.3.1.    Pengumpulan Dataset	24
3.3.2.    Labelling Data	25
3.3.3.    Identifikasi Garis Pinggir Pada Lapangan Sepakbola Robot	26
3.3.4.    Convert Xml to Csv	28
3.3.5.    Membuat Labelmap	29
3.3.6.    Membuat TFRecord	29
3.3.7.    Identifikasi Garis Tengah Pada Lapangan Sepakbola Robot	29
BAB 4 IMPLEMENTASI DAN PENGUJIAN	35
4.1    Implementasi Sistem	35
4.2    Perangkat Penelitian	35
4.2.1.    Kebutuhan Perangkat Keras	35
4.2.2.    Implementasi Perangkat Lunak	36
4.1    Pengujian	38
4.2    Hasil Deteksi	40
4.3    Perhitungan <i>Confusion Matrix</i>	48
BAB 5 KESIMPULAN DAN SARAN	51
5.1    Saran	51
5.2    Kesimpulan	51
DAFTAR PUSTAKA	52

**DAFTAR TABEL**

<b>Tabel 2. 1</b> Penelitian Terdahulu	20
<b>Tabel 4. 1</b> Contoh Citra Pengujian <i>RoboCup Soccer</i>	38
<b>Tabel 4. 2</b> Hasil Deteksi Garis Lapangan	40
<b>Tabel 4. 3</b> TP, TN, FP dan FN Confusion Matrix	48
<b>Tabel 4. 4</b> Perhitungan Confusion Matrix Deteksi Garis	48

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> <i>Region Based Image Segmentation</i>	7
<b>Gambar 2. 2</b> <i>Gaussian Blur</i>	8
<b>Gambar 2. 3</b> <i>Bitwise Operation</i>	9
<b>Gambar 2. 4</b> Representasi Citra Warna	10
<b>Gambar 2. 5</b> Representasi Citra Keabuan	11
<b>Gambar 2. 6</b> Representasi Citra Biner	11
<b>Gambar 2. 7</b> Proses Deteksi Tepi Citra	12
<b>Gambar 2. 8</b> Model Tepi Satu Citra	13
<b>Gambar 2. 9</b> Jenis-Jenis Tepi Citra	14
<b>Gambar 2. 10</b> Citra sebuah garis dalam koordinat Kartesian dengan satu titik uji.	15
<b>Gambar 2. 11</b> Hasil transformasi dari satu titik uji dalam ruang parameter $r - \theta$ .	15
<b>Gambar 2. 12</b> Klasifikasi Haar	17
<b>Gambar 2. 13</b> Arsitektur SSD	18
<b>Gambar 2. 14</b> <i>MobileNet V2</i>	19
<b>Gambar 2. 15</b> <i>MobileNet V</i>	20
<b>Gambar 3. 1</b> Arsitektur Umum	24
<b>Gambar 3. 2</b> Sampel Citra Uji	25
<b>Gambar 3. 3</b> Dataset yang Telah Diberikan Label	25
<b>Gambar 3. 4</b> <i>File xml</i> yang sudah dilabeling	26
<b>Gambar 3. 5</b> Definisikan Interval Warna Hijau dan Putih Pada Mode Gambar HSV	26
<b>Gambar 3. 6</b> Penerapan <i>Binary Masking</i> dan <i>Smoothing Image</i>	27
<b>Gambar 3. 7</b> Penerapan <i>Hough Line Transformation</i>	27
<b>Gambar 3. 8</b> Rumus Konversi Citra Keabuan	28
<b>Gambar 3. 9</b> <i>Convert xml to csv</i>	28
<b>Gambar 3. 10</b> Membuat Labelmap	29
<b>Gambar 3. 11</b> Membuat TFRecord	29
<b>Gambar 3. 12</b> Jumlah step yang digunakan	30
<b>Gambar 3. 13</b> Jumlah <i>class object</i> yang digunakan	31
<b>Gambar 3. 14</b> Jenis Fitur <i>Ekstrasi</i>	31
<b>Gambar 3. 15</b> Model Hasil <i>Training</i>	32

<b>Gambar 3. 16</b> Pembuatan <i>Inference</i> Model	33
<b>Gambar 3. 17</b> Pendeteksian Garis Tengah dengan Metode <i>SSD MobileNetV2</i>	33
<b>Gambar 3. 18</b> <i>Finalisasi Overlay</i>	34
<b>Gambar 4. 1</b> Tampilan Utama Sistem	36
<b>Gambar 4. 2</b> Tampilan Aplikasi Saat Menjalankan Program	37
<b>Gambar 4. 3</b> Tampilan pendeteksian objek garis koordinat garis tengah lapangan	37
<b>Gambar 4. 4</b> Visualisasi Citra Hasil Pengujian	49

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Sepak bola adalah olahraga yang paling digemari di seluruh dunia, termasuk juga di Indonesia. Dengan cepatnya perkembangan teknologi, banyak orang yang mulai tertarik untuk mengembangkan robot sepakbola. Dampak dari hal tersebut adalah diadakan nya turnamen robot sepakbola untuk pertama kali yang dinamakan RoboCup SPL. Kompetisi ini memanfaatkan lapangan sepak bola dalam ruangan yang telah dimodifikasi ukurannya menjadi lebih kecil, dengan setiap tim terdiri dari 5 robot beroda yang sepenuhnya otonom, artinya tidak ada intervensi manusia dalam pengoperasiannya.

Di Indonesia juga diadakan kontes robot yang diadakan oleh Kementrian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia (Kemenristek Dikti) dimana diikuti oleh seluruh Perguruan Tinggi dibawah naungan Kemenristek Dikti. Awalnya RISTEKDIKTI Mengadakan kompetisi "Kompetisi Robot Pemadam Api Indonesia Beroda" (KRPAI-Beroda) yang menampilkan robot beroda yang dirancang untuk mencari jalur dan ruangan guna memadamkan api. Juara nasional dari acara ini akan mewakili Indonesia dalam kompetisi serupa di Trinity College. Oleh karena itu untuk meningkatkan keilmuan dan kreatifitas mahasiswa di bidang robotika diadakanlah perlombaan Robot Sepakbola Beroda Indonesia. Dalam kontes ini, mahasiswa diharapkan untuk mengembangkan kemampuan di berbagai bidang, termasuk mekanika, manufaktur, elektronika, pemrograman, kecerdasan buatan, pemrosesan gambar, komunikasi digital, dan strategi. Selain itu, pengembangan dalam hal disiplin, toleransi, sportivitas, kerjasama, saling menghargai, pengendalian emosi, dan kemampuan soft skill lainnya juga sangat diperlukan.

Kontes Robot Sepakbola Beroda Indonesia diadakan sesuai dengan regulasi yang diterapkan dalam RoboCup *Middle Size League* (MSL), dengan penyesuaian

tertentu sesuai dengan kondisi di Indonesia, seperti ukuran lapangan. Aturan dalam RoboCup MSL didasarkan pada modifikasi aturan FIFA untuk sepakbola manusia.. Untuk dapat memainkan pertandingan robot sepakbola beroda dengan baik, sebuah robot minimal harus memiliki beberapa keahlian dasar seperti mengetahui pergerakan bola, menggiring bola, dan menendang bola kearah gawang lawan. Untuk mencapai hal itu maka robot harus bisa mendeteksi empat buah komponen dasar dalam permainan sepakbola, yaitu: Robot lawan, bola, lapangan (garis-garis lapangan yang menjadi pembatas), dan gawang lawan. Adapun beberapa penelitian yang telah dilakukan sebelumnya, diantaranya penelitian yang dilakukan Susianto, Budi (2018) Mendeteksi objek gawang menggunakan metode *HoughTransform* dengan tingkat keberhasilan 79 dari 116 kali percobaan. Pendeteksian gawang tampak jelas secara visual pada jarak di atas 50 cm dan posisi antara  $-50^{\circ}$  sampai  $50^{\circ}$ . Penelitian yang dilakukan oleh Hapsari, Widi (2016) menggunakan metode *Hough Transform*, digunakan untuk Menganalisis karakteristik terhadap motif batik parang dari sebaran garis pada sudut kemiringan tertentu dan panjang garis dalam kelompok kemiringan sudut. Bahan penelitian adalah 50 citra dari sampel data dan 30 citra data dan penelitian masing masing parang dan non parang. Baik akurasi batik parang maupun non parang. Selanjutnya ada juga penelitian yang dilakukan oleh Sa'adiyah, Halimatus, et al (2011) dengan judul “Aplikasi Transformasi Hough Untuk Deteksi Garis Lurus”. Keberhasilan program transformasi Hough dalam mendeteksi dan merekonstruksi garis lurus pada suatu citra bervariasi, dengan tingkat keberhasilan yang mencapai 100% untuk beberapa kasus dan hanya mencapai 20% untuk yang lainnya. Varian ini disebabkan oleh garis lurus yang tidak terdeteksi karena memiliki nilai R (yang dihasilkan setelah transformasi ke koordinat polar) di bawah nilai R maksimum yang telah dikalikan dengan nilai ambang tertentu, sehingga garis tersebut tidak terdeteksi sebagai garis lurus. Secara keseluruhan, hasil rata-rata persentase keberhasilan total adalah sebesar 90%.

Selanjutnya ada juga penelitian yang dilakukan Lubis, Zulfadli F (2018) dengan judul “Penentuan Pelanggaran Garis Pembatas Parkir Kendaraan Roda Empat Menggunakan Algoritma Hough Transfrom”. Hasil penelitian menunjukkan bahwa dengan menggunakan 50 data citra uji, rata-rata tingkat ketepatan (precision) mencapai 88% dan rata-rata tingkat perolehan (recall) mencapai 92%, dengan nilai akurasi keseluruhan mencapai 90%.

Berikutnya ada penelitian yang dilakukan oleh Rizaldy, Aulia R (2020) dengan judul “Deteksi Ruang Garis Khusus (RHK) Sepeda Motor Menggunakan *Hough Transform* “. Hasil dari penelitian ini berupa citra yang menampilkan garis-garis yang terdeteksi secara visual menggunakan metode *Hough Transform*. Penelitian ini bertujuan untuk mengoptimalkan algoritma pendeteksian garis pada citra radiografi kepala (RHK) agar mendeteksi garis-garis dengan lebih tepat. Proses dengan hasil ketepatan secara visual adalah hasil dengan menghilangkan proses blur dan melakukan pengaturan terhadap variable threshold pada pemrosesan *Hough transform*.

Penelitian yang dilakukan Sumin, Agus, et al (2020) dengan judul Pembuatan Aplikasi Deteksi Objek Menggunakan TensorFlow Object Detection API dengan Memanfaatkan SSD MobileNet V2 Sebagai Model Pra-Terlatih. Penelitian ini menggunakan dataset sebanyak 500 gambar dengan membagi menjadi tiga bagian, yaitu train set, validation set, dan test set dengan masing - masing perbandingan sebesar 70% : 20% : 10%. Pelatihan dilakukan dengan bantuan Google Research Colaboratory sebagai virtual machine. Penelitian ini menggunakan Python 3.6.8 sebagai bahasa pemrograman dan memakai beberapa library yang disediakan oleh Python. Dari uji coba yang dilakukan, aplikasi ini memiliki tingkat akurasi sebesar 93.02% pada test set.

Berdasarkan latar belakang tersebut maka dilakukan penelitian dengan judul “DETEKSI GARIS LAPANGAN SEPAKBOLA ROBOT DENGAN METODE *HOUGH TRANSFORM* DAN *SSD MOBILENET V2* ”.

## 1.2 Rumusan Masalah

Dengan mulai banyaknya pertandingan sepakbola robot yang diadakan untuk tingkat nasional maupun internasional, membuat banyak pihak berlomba lomba untuk mengikuti dan mencoba memenangkan pertandingan ini. Namun, pengembangan sistem maupun alat yang digunakan masih menjadi tantangan. Oleh karena itu, penelitian ini bertujuan agar mengetahui seberapa efektif metode *hough transform* dalam mendeteksi garis lapangan sepakbola robot.



### 1.3 Tujuan Penelitian

Adapun Tujuan dari penelitian tugas akhir, mengimplementasikan metode *hough transform* pada sistem untuk mendeteksi garis lapangan pada citra permainan sepakbola robot.

### 1.4 Batasan Masalah

Dalam penelitian pada tugas akhir ini, terdapat batasan masalah antara lain :

1. Input Citra berupa lapangan permainan sepakbola robot.
2. Citra memiliki format ekstensi \*.png dengan dimensi 300 x 300 pixels.
3. Sistem dirancang dengan menggunakan algoritma *hough transform* dan *SSD Mobilenet V2* untuk mendeteksi garis lapangan pada citra sepakbola robot.

### 1.5 Manfaat Penelitian

Diharapkan manfaat penelitian pada tugas akhir, sebagai berikut :

1. Mengetahui tingkat akurasi algoritma *hough transform* mendeteksi garis lapangan pada citra permainan sepakbola robot.
2. Memberikan gambaran dan inspirasi pada penelitian selanjutnya untuk mendeteksi garis lapangan sepakbola robot.

### 1.6 Metodologi Penelitian

Kerangka Metodologi penelitian dalam tugas akhir sebagai berikut :

1. Penelitian Pustaka  
Merupakan proses mengidentifikasi dan mengumpulkan informasi dari berbagai sumber seperti buku, jurnal, artikel, dan referensi lainnya mengenai penggunaan metode deteksi menggunakan algoritma transformasi Hough.
2. Analisis  
Merupakan proses mengidentifikasi dan mengumpulkan informasi dari berbagai sumber seperti buku, jurnal, artikel, dan referensi lainnya mengenai penggunaan metode deteksi menggunakan algoritma transformasi Hough.
3. Perancangan

Tahap di mana arsitektur sistem direncanakan, termasuk pengumpulan data, proses pelatihan, dan desain antarmuka. Perancangan ini didasarkan pada hasil analisis dari penelitian pustaka yang telah dilakukan.

#### 4. Implementasi

Melibatkan pengkodean program menggunakan lingkungan pengembangan berbasis *Hough Transform* dan *Single Shot Multibox Detector (SSD)* untuk menerapkan sistem yang telah direncanakan sebelumnya.

#### 5. Pengujian

Proses verifikasi untuk memastikan bahwa aplikasi yang dikembangkan beroperasi sesuai dengan ekspektasi yang telah ditetapkan.

#### 6. Dokumentasi

Langkah terakhir dalam penelitian tugas akhir, yang mencakup dokumentasi dari tahap analisis hingga pengujian dalam format tugas akhir.

### 1.7 Sistematika Penulisan

Sistematika dalam penulisan tugas akhir akan dibagi menjadi beberapa bab sebagai berikut:

#### **BAB 1 : PENDAHULUAN**

Pada bab ini diuraikan hal-hal yang merupakan latar belakang masalah, pembuatan rumusan masalah, pembatasan masalah yang ada, pembuatan tujuan dan manfaat dari penelitian, metodologi penelitian yang dilakukan serta sistematika penulisan.

#### **BAB 2 : LANDASAN TEORI**

Berisi dasar teori pendeteksian citra menggunakan metode *hough transform* dan *ssd mobilenet v2*.

#### **BAB 3 : ANALISIS DAN PERANCANGAN**

Dari Bab ini berisikan analisis terhadap deteksi garis lapangan pada citra permainan sepakbola robot dengan algoritmahough transform serta pemodelan sistem yang akan dirancang.

#### **BAB 4 : IMPLEMENTASI DAN PENGUJIAN**

Berisi bentuk jadi dari sistem yang telah dibangun berupa tampilan-tampilan serta hasil pengujian terhadap data serta nilai akurasi pengenalan.

## **BAB 5 : KESIMPULAN DAN SARAN**

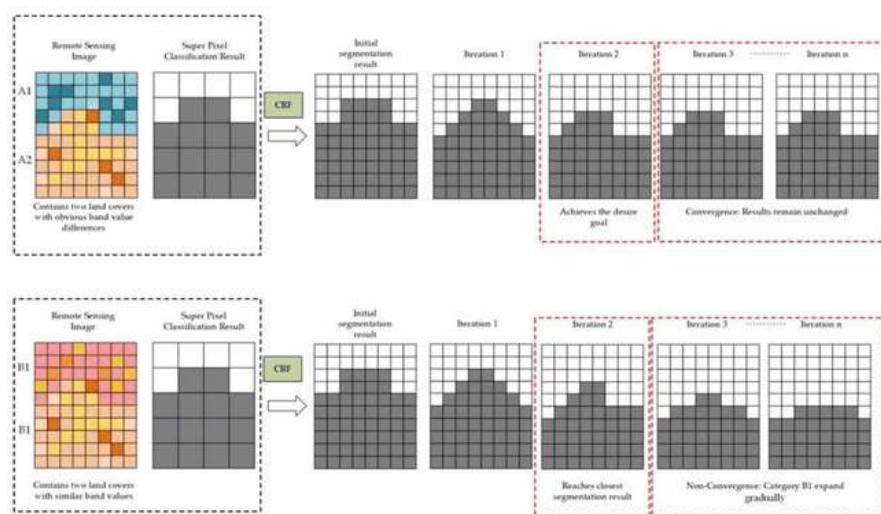
Bab ini berisi pemaparan hasil kesimpulan dari hasil pengujian yang telah dilakukan serta saran-saran untuk perbaikan dan pengembangan lebih lanjut.

## BAB 2

### LANDASAN TEORI

#### 2.1 Region Based Image Segmentation

Algoritma segmentasi rentang berbasis kawasan dapat dikategorikan lebih lanjut menjadi dua kelompok besar: algoritme segmentasi rentang berbasis model parametrik dan algoritme pertumbuhan kawasan. Algoritma kelompok pertama didasarkan pada asumsi model permukaan parametrik dan pengelompokan titik data sehingga semuanya dapat dianggap sebagai titik permukaan dari model parametrik yang diasumsikan (contoh dari model tersebut). Algoritme pertumbuhan wilayah dimulai dengan mensegmentasi gambar ke wilayah awal. Wilayah-wilayah ini kemudian digabungkan atau diperluas dengan menggunakan strategi pertumbuhan wilayah. Daerah awal dapat diperoleh dengan menggunakan metode yang berbeda, termasuk metode iteratif atau acak. Kelemahan dari algoritma kelompok ini adalah bahwa pada umumnya mereka menghasilkan batas yang terdistorsi karena segmentasi biasanya dilakukan pada tingkat wilayah daripada tingkat piksel.



**Gambar 2. 1** *Region Based Image Segmentation* (Pan, X, 2018)

## 2.2 Gaussian Blur

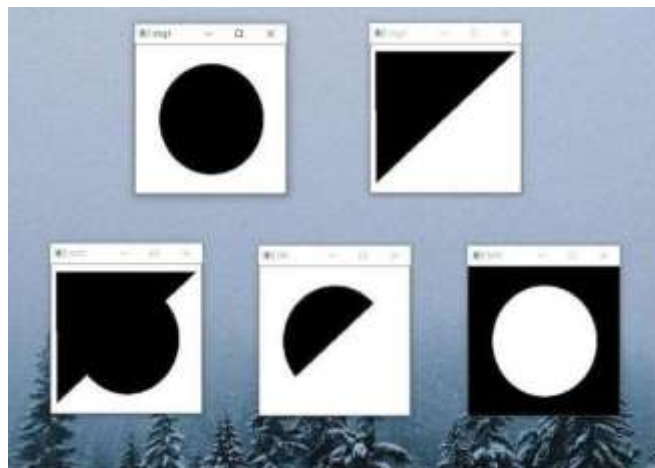
Dalam pemrosesan gambar, Gaussian blur atau dikenal juga sebagai Gaussian smoothing adalah efek yang dihasilkan dari memburamkan gambar menggunakan fungsi Gaussian, yang dinamai sesuai dengan ahli matematika dan ilmuwan Carl Friedrich Gauss. Teknik ini banyak digunakan dalam perangkat lunak grafis, terutama untuk mengurangi noise gambar dan menghilangkan detail yang tidak diinginkan. Efek visual dari Gaussian blur adalah kehalusan buram yang menyerupai melihat gambar melalui layar transparan, berbeda dengan efek bokeh yang dihasilkan oleh lensa di luar fokus atau bayangan objek di bawah cahaya biasa. Penghalusan Gaussian juga sering digunakan sebagai tahap pra-pemrosesan dalam algoritme visi komputer untuk meningkatkan struktur gambar pada berbagai skala—lihat representasi ruang skala dan implementasi ruang skala.



**Gambar 2. 2** *Gaussian Blur* ([www.digitalbunker.dev/understanding-gaussian-blurs](http://www.digitalbunker.dev/understanding-gaussian-blurs))

## 2.3 Bitwise Operation

Operasi bitwise pada gambar sangat berguna ketika kita perlu mengekstrak objek yang tidak beraturan dari latar belakang dalam sebuah gambar. Misalnya, dalam situasi di mana kita ingin mengekstrak objek tertentu dari gambar dan menempelkannya pada gambar lain, operasi bitwise dapat digunakan untuk memisahkan latar depan (objek yang ingin diekstrak) dari latar belakang.



**Gambar 2. 3** *Bitwise Operation* ([www.pyimagesearch.com](http://www.pyimagesearch.com))

## 2.4 Citra Digital

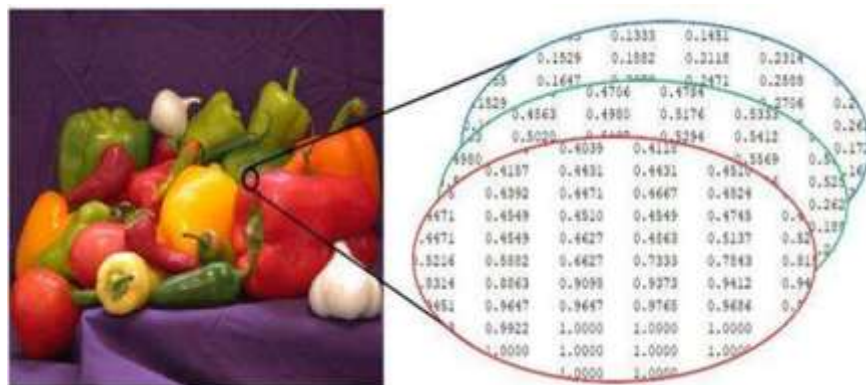
Pengolahan citra digital merupakan ruang lingkup pengembangan sistem yang berbagai sudut pandang dari segi keamanan, Gambar dua dimensi yang ditampilkan pada layar komputer direpresentasikan sebagai himpunan nilai digital yang disebut piksel (picture elements). Secara matematis, citra dianggap sebagai fungsi kontinu dari intensitas cahaya dalam bidang dua dimensi. Citra digital, di sisi lain, adalah representasi diskrit dari citra tersebut, dinyatakan sebagai fungsi  $f(x,y)$  di mana koordinat spasial ( $x$  dan  $y$ ) telah di-diskritisasi untuk sampling, dan tingkat kecerahan (intensitas) telah di-diskritisasi juga dengan fungsi intensitas cahaya  $f(x,y)$ . Setiap titik ( $x, y$ ) dalam citra digital memiliki nilai yang mewakili tingkat kecerahan pada titik tersebut. Konsep ini mengacu pada diskritisasi koordinat spasial dan diskritisasi tingkat kecerahan (keabuan) dalam citra, yang merupakan prinsip dasar dalam pemrosesan citra digital (Gonzales & Wood, 2003). Citra digital terdiri dari kumpulan titik yang disebut piksel (pixel atau picture element), di mana setiap piksel direpresentasikan sebagai sebuah kotak kecil. Setiap piksel memiliki koordinat yang dinyatakan dalam bentuk ( $y, x$ ), di mana  $y$  menunjukkan baris dan  $x$  menunjukkan kolom. Secara umum, koordinat pojok kiri atas biasanya dinyatakan sebagai  $(0, 0)$ . Dengan demikian, dalam citra berukuran  $M$  baris dan  $N$  kolom, koordinat piksel terbawah dan terkanan akan berada pada koordinat  $(M-1, N-1)$  (Kadirss, 2013).

Elemen-elemen yang terdiri dari kecerahan (brightness), kontras (contrast), kontur (contour), warna (color), bentuk (shape), dan tekstur (texture) merupakan komponen-komponen penting dalam citra digital. Secara umum, citra digital dapat

dibagi menjadi dua jenis, yaitu citra diam (*still image*) dan citra bergerak (*motion image*) (Gonzalez, 1992). Banyak peralatan elektronik yang menghasilkan citra digital, seperti scanner, kamera digital, mikroskop digital, dan pembaca sidik jari (*fingerprint reader*). Untuk mengolah citra digital sesuai kebutuhan, tersedia berbagai perangkat lunak seperti Adobe Photoshop dan GIMP (*GNU Image Manipulation Program*), yang menyediakan berbagai fitur untuk pengolahan citra digital (Kadir et al., 2016).

#### 2.4.1. Representasi Citra Warna

Citra warna adalah hasil dari penggabungan tiga warna primer, yaitu merah, hijau, dan biru (RGB). Nilai intensitas untuk setiap komponen RGB berkisar antara 0 hingga 255, di mana setiap komponen memiliki nilai 8 bit atau  $2^8$ . Dengan demikian, jumlah kemungkinan warna yang dapat dihasilkan oleh komponen RGB adalah  $2^{24}$  atau  $256 \times 256 \times 256$ , yang setara dengan 16.777.216 kombinasi warna.



**Gambar 2. 4** Representasi Citra Warna (MathWork, 2022)

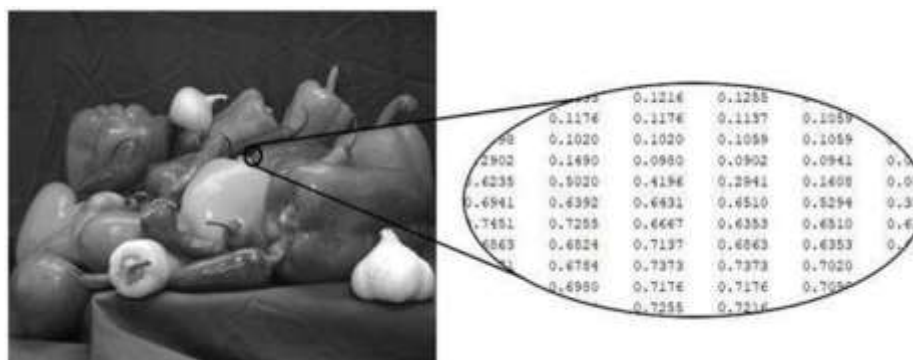
#### 2.4.2. Representasi Citra Keabuan

Citra digital keabuan memiliki gradasi warna dari putih hingga hitam. Rentang warna pada citra grayscale banyak digunakan dalam berbagai aplikasi, termasuk dalam dunia kedokteran seperti pada gambar sinar-X (X-ray). Grayscale adalah hasil rata-rata dari citra warna dengan menggunakan persamaan sebagai berikut:

$$\text{Grayscale} = (0.2989 * R) + (0.5870 * G) + (0.1140 * B)$$

Di mana R, G, dan B adalah nilai intensitas warna merah, hijau, dan biru dari piksel tersebut dalam citra warna asli. Persamaan ini memberikan bobot yang berbeda pada

setiap saluran warna berdasarkan kontribusi relatif mereka terhadap kecerahan keseluruhan citra grayscale.

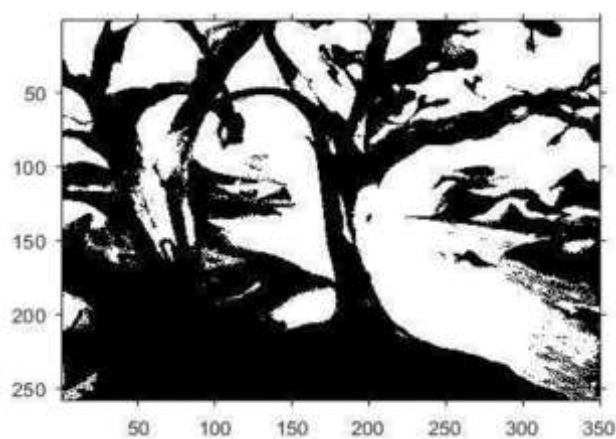


**Gambar 2. 5** Representasi Citra Keabuan (MathWork, 2022)

#### 2.4.3. Representasi Citra Biner

Citra binary merupakan citra yang setiap piksel dinyatakan dengan 2 kemungkinan nilai yakni 0 dan 1. Nilai 0 mewakili warna putih dan 1 mewakili warna hitam. Proses perhitungan citra ke binary setelah didapatkan hasil melalui thresholding citra *grayscale* pada persamaan sebagai berikut :

$$g(x, y) = \begin{cases} 1 & \text{jika } f(x, y) \geq T \\ 0 & \text{jika } f(x, y) < T \end{cases}$$



**Gambar 2. 6** Representasi Citra Biner (Hayat, F et al., 2021)

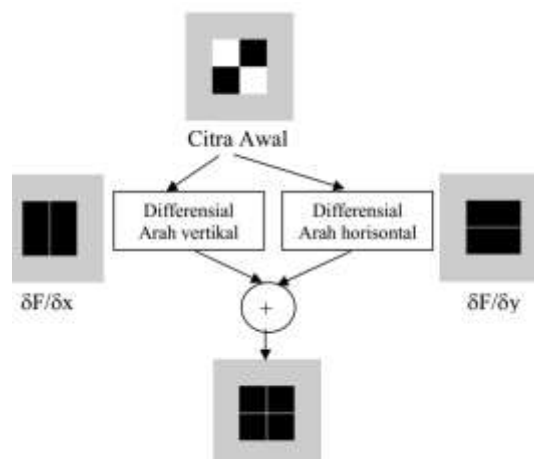


## 2.5 Deteksi Tepi

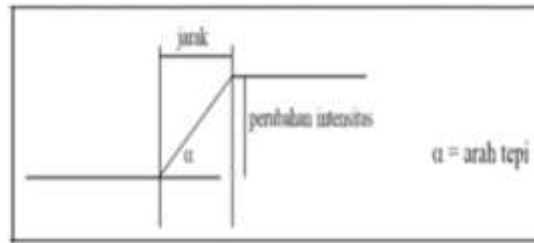
Tepi citra (edge) merupakan perubahan yang tiba-tiba dalam nilai intensitas derajat keabuan dalam jarak yang singkat. Deteksi tepi (Edge Detection) pada suatu citra adalah proses untuk mengidentifikasi dan menandai garis tepi yang membatasi dua wilayah citra homogen yang memiliki tingkat kecerahan yang berbeda. Tujuan dari deteksi tepi pada citra adalah sebagai berikut:

1. Untuk menandai bagian yang memiliki detail dalam citra, sehingga memudahkan dalam analisis atau pengolahan lebih lanjut.
2. Untuk memperbaiki detail dari citra yang kabur, yang mungkin terjadi karena kesalahan atau efek dari proses akuisisi citra.
3. Untuk mengubah citra 2D menjadi bentuk kurva, yang memungkinkan untuk mendapatkan representasi yang lebih ringkas atau lebih mudah diproses.

Dengan mendeteksi tepi, informasi penting tentang struktur objek dalam citra dapat diekstraksi dengan lebih efektif, membantu dalam berbagai aplikasi termasuk penglihatan komputer, pengolahan citra medis, dan pengenalan pola. Suatu titik (x,y) dikatakan sebagai tepi (edge) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya.



**Gambar 2. 7** Proses Deteksi Tepi Citra ( Liantoni, 2015)

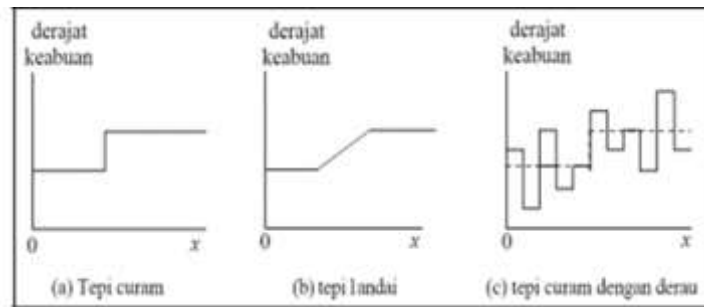


**Gambar 2. 8** Model Tepi Satu Citra ( Ridwan, et al., 2020)

Tiga jenis tepi yang terdapat dalam citra digital, sebagaimana dijelaskan oleh Panetta et al. (2015), adalah sebagai berikut:

1. Tepi Curam : Tepi curam adalah tepi dengan perubahan intensitas yang tajam. Artinya, terjadi perubahan signifikan dalam tingkat kecerahan piksel dari satu sisi tepi ke sisi lainnya. Arah tepi biasanya berkisar sekitar  $90^\circ$ .
2. Tepi Landai : Tepi landai, juga dikenal sebagai tepi lebar, adalah tepi dengan sudut arah yang kecil. Ini berarti perubahan dalam intensitas cahaya terjadi secara bertahap, dan tepi landai dapat dianggap terdiri dari sejumlah tepi lokal yang berdekatan.
3. Tepi yang Mengandung Derau (Noise) : Tepi yang mengandung derau adalah tepi yang terpengaruh oleh derau atau noise dalam citra. Noise dapat menyebabkan gangguan atau perubahan yang tidak diinginkan pada tepi, membuatnya sulit untuk diidentifikasi atau diproses dengan benar.

Umumnya tepi yang terdapat dalam aplikasi computer vision seringkali mengandung derau atau noise yang dapat mempengaruhi kualitas tepi yang dideteksi. Oleh karena itu, seringkali dilakukan operasi peningkatan kualitas citra (image enhancement) sebelum melakukan deteksi tepi. Operasi peningkatan kualitas citra dapat dilakukan dengan berbagai teknik, termasuk filtrasi spasial atau frekuensi untuk mengurangi noise, penajaman kontras, dan normalisasi histogram, antara lain. Tujuan dari operasi ini adalah untuk meningkatkan kualitas citra dan membuat tepi lebih jelas dan terdefinisi dengan baik sebelum melakukan deteksi tepi.



**Gambar 2. 9** Jenis-Jenis Tepi Citra (Arifin, 2011)

## 2.6 Transformasi Hough

Transformasi Hough diperkenalkan pertama kali oleh Paul Hough pada tahun 1962 sebagai sebuah teknik untuk mendeteksi garis lurus dalam citra. Ini adalah sebuah teknik transformasi citra yang berguna untuk mengisolasi atau mengekstraksi fitur dari sebuah citra. Transformasi Hough bertujuan untuk mendapatkan fitur-fitur spesifik dalam citra, dan salah satu teknik yang paling umum digunakan adalah Classical Hough Transform, yang dapat digunakan untuk mendeteksi objek berbentuk kurva seperti garis, lingkaran, elips, dan parabola.

Keuntungan utama dari Transformasi Hough adalah kemampuannya untuk mendeteksi tepi yang tidak terputus atau memiliki celah pada batas fitur. Selain itu, transformasi Hough relatif tidak terpengaruh oleh derau atau noise dalam citra, membuatnya lebih robust dalam mendeteksi fitur-fitur penting.

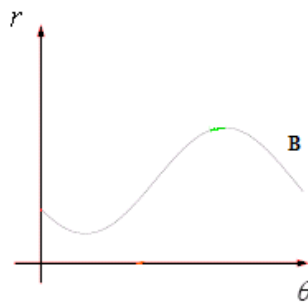
Dengan demikian, transformasi Hough telah menjadi alat yang sangat berguna dalam berbagai aplikasi pengolahan citra, termasuk dalam pengenalan pola, visi komputer, dan analisis citra medis.

Transformasi Hough menggunakan representasi parametrik untuk menggambarkan garis-garis dalam citra. Dalam representasi ini, kita menggunakan persamaan garis umum  $y=mx+c$  di mana  $m$  adalah gradien (slope) garis dan  $c$  adalah perpotongan dengan sumbu  $y$ . Ketika titik-titik dalam citra diubah menjadi ruang parameter  $m-c$ , sebuah garis lurus dalam citra akan diwakili oleh titik dalam ruang parameter yang saling berpotongan pada suatu titik. Sebaliknya, jika dalam citra terdapat beberapa garis yang saling berpotongan pada suatu titik, maka dalam ruang parameter  $m-c$ , mereka akan diwakili oleh sebuah garis lurus dengan persamaan  $y=mx+c$ . Namun, terdapat masalah ketika menghadapi garis vertikal, di mana gradien

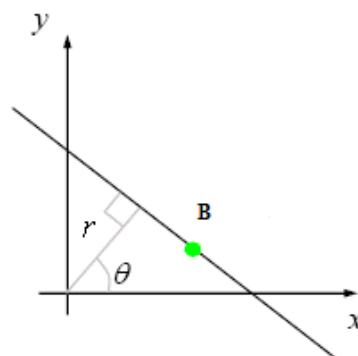
(m) garis vertikal adalah tak berhingga ( $\infty$ ). Untuk mengatasi hal ini, digunakan representasi alternatif dalam bentuk persamaan parametrik:

$$r = x \cos \theta + y \sin \theta$$

Di sini,  $r$  adalah jarak titik dari titik asal (0,0) ke garis yang mempunyai sudut kemiringan  $\theta$  dengan sumbu  $x$ . Persamaan ini lebih fleksibel dan dapat mengatasi masalah ketika menghadapi garis vertikal. Dimana gambar koordinat kartesiannya ditunjukkan pada Gambar 2.10. Sebuah titik B dalam gambar tersebut apabila ditransformasi ke dalam ruang parameter  $r$  - maka akan menjadi seperti Gambar 2.11. Apabila di dalam citra terdapat suatu garis lurus, maka jika garis lurus citra ditransformasi kedalam ruang parameter – akan terjadi suatu titik penumpukan antar kurva sinusoida hasil pentransformasian masing-masing komponen piksel garis yang membentuk garis lurus tersebut.



**Gambar 2. 10** Citra sebuah garis dalam koordinat Kartesian dengan satu titik uji. (Sunarya, U, et al., 2015)



**Gambar 2. 11** Hasil transformasi dari satu titik uji dalam ruang parameter  $r - \theta$ . (Sunarya, U, et al., 2015)

## 2.7 OpenCV

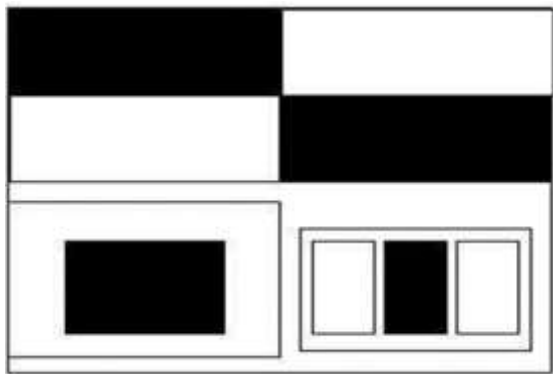
OpenCV merupakan singkatan dari "Intel Open Source Computer Vision Library" yang terdiri dari sekurang-kurangnya 300 fungsi C atau bahkan lebih. Ini adalah perangkat lunak sumber terbuka (open source) yang gratis dan dapat digunakan dalam konteks komersial maupun non-komersial tanpa perlu membayar lisensi kepada Intel. OpenCV dapat dijalankan pada komputer berbasis Windows maupun Linux. Library OpenCV menyediakan berbagai fungsi-fungsi untuk pengolahan citra dan penglihatan komputer, memungkinkan pengguna untuk mengimplementasikan berbagai teknik dalam domain tersebut. Ini memberikan akses ke berbagai fungsi pemrosesan gambar, analisis pola, deteksi objek, pelacakan objek, dan banyak lagi. Dengan fitur-fitur yang luas dan dukungan komunitas yang kuat, OpenCV telah menjadi salah satu alat yang sangat berharga dalam pengembangan aplikasi pengolahan citra dan penglihatan komputer. Ini memungkinkan komunitas open source untuk mengikuti perkembangan terbaru dalam teknologi penglihatan mesin seiring dengan pertumbuhan komputer pribadi yang terus berkembang.

Metode Viola-Jones yang digunakan dalam pengenalan wajah pada OpenCV merupakan pendekatan yang efektif dan efisien untuk mendeteksi objek dalam gambar, seperti wajah, menggunakan klasifikasi bertingkat. Metode ini melibatkan beberapa konsep kunci:

1. Haar Feature: Haar feature adalah bentuk sederhana yang digunakan untuk mengidentifikasi pola dalam citra. Ini melibatkan pengukuran perbedaan intensitas antara area piksel di bawah kotak tertentu dalam citra. Fitur-fitur Haar ini digunakan sebagai deskriptor yang berguna untuk membedakan antara berbagai objek dalam citra.
2. Integral Image: Integral image adalah representasi citra yang digunakan untuk mempercepat proses ekstraksi fitur Haar. Ini memungkinkan penghitungan fitur Haar di berbagai skala dan posisi dengan lebih efisien.
3. Metode Machine Learning AdaBoost: Metode AdaBoost digunakan untuk melatih klasifikasi untuk mengenali wajah berdasarkan fitur Haar yang diekstraksi dari citra pelatihan. Ini adalah teknik machine learning yang digunakan untuk meningkatkan kinerja klasifikasi dengan memilih subset fitur yang paling informatif.

4. **Klasifikasi Bertingkat:** Klasifikasi bertingkat digunakan untuk menggabungkan banyak fitur Haar secara efisien dan efektif untuk membedakan antara objek yang ingin dideteksi (misalnya, wajah) dan latar belakang.

Bentuk Haar wavelets yang digunakan dalam metode Viola-Jones adalah segi empat tunggal dalam dua dimensi. Gelombang persegi ini terdiri dari pasangan segiempat yang berdekatan, satu berwarna putih dan satu berwarna hitam, yang digunakan sebagai fitur untuk membedakan antara objek dan latar belakang dalam citra.



**Gambar 2. 12** Klasifikasi Haar ( Ali, et al., 2015)

Pengukuran kinerja sistem merupakan langkah penting dalam evaluasi dan pengembangan sistem, terutama dalam konteks pengolahan data dan pengenalan pola. Confusion matrix merupakan salah satu metode yang sangat berguna untuk mengukur kinerja suatu metode klasifikasi atau sistem pengenalan pola.

Confusion matrix adalah tabel yang digunakan untuk membandingkan hasil klasifikasi yang dilakukan oleh suatu sistem dengan klasifikasi yang seharusnya. Tabel ini menggambarkan jumlah true positive (TP), false positive (FP), true negative (TN), dan false negative (FN) dari hasil klasifikasi. Dari informasi ini, berbagai metrik kinerja seperti akurasi, presisi, recall, dan F1-score dapat dihitung.

Secara umum, confusion matrix memiliki empat sel yang masing-masing mewakili:

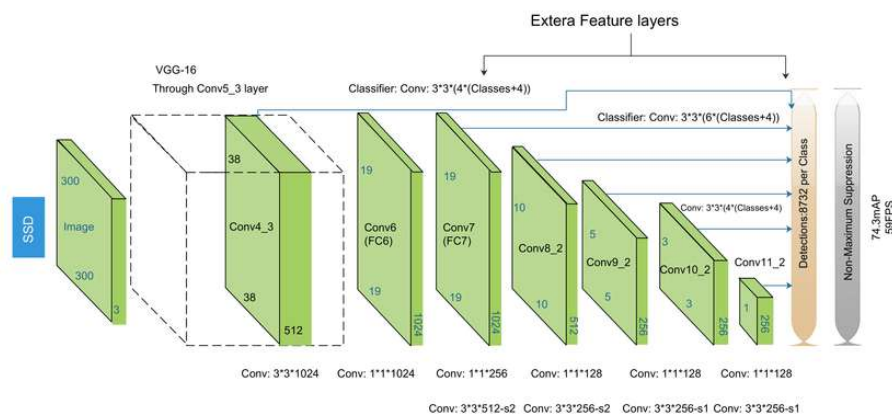
1. True Positive (TP): Jumlah sampel positif yang diklasifikasikan dengan benar oleh sistem.
2. False Positive (FP): Jumlah sampel negatif yang salah diklasifikasikan sebagai positif oleh sistem.
3. True Negative (TN): Jumlah sampel negatif yang diklasifikasikan dengan benar oleh sistem.

4. False Negative (FN): Jumlah sampel positif yang salah diklasifikasikan sebagai negatif oleh sistem.

Dengan menggunakan informasi dari confusion matrix, kita dapat menghitung berbagai metrik evaluasi seperti akurasi, presisi, recall, dan F1-score, yang membantu dalam memahami kinerja sistem secara lebih mendalam. Confusion matrix sangat penting dalam pengujian dan evaluasi model klasifikasi dalam berbagai bidang, termasuk pengenalan pola, pemrosesan bahasa alami, dan pengolahan citra.

## 2.8 Single Shot Multibox Detector (SSD)

Single Shot Multibox Detector (SSD) adalah algoritma deteksi objek yang cepat dan efisien yang digunakan dalam pengolahan citra dan visi komputer. Keunggulan SSD adalah kecepatannya dan efisiensinya dalam mendeteksi objek, karena menggabungkan langkah-langkah deteksi menjadi satu proses. Hal ini membuatnya cocok untuk aplikasi real-time yang memerlukan deteksi objek cepat, seperti pengawasan lalu lintas, pengenalan wajah, atau pengenalan objek dalam video..Arsitektur SSD bisa dilihat pada gambar berikut ini.



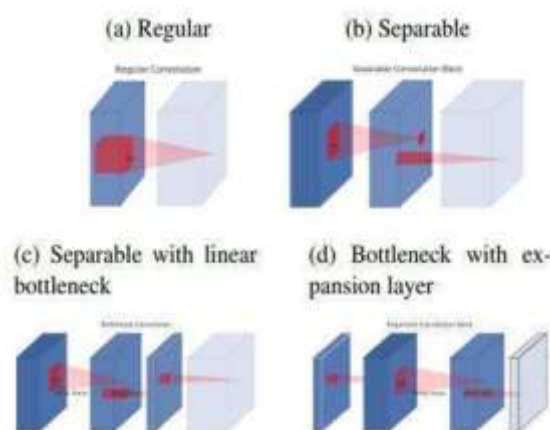
**Gambar 2. 13** Arsitektur SSD (Xin, et al., 2020)

## 2.9 MobileNet

*MobileNet* merupakan suatu metode yang merupakan bagian dari arsitektur jaringan syaraf tiruan yang dapat memenuhi kebutuhan pengolahan data yang berlebihan. Mobilenet dikembangkan oleh Google Architecture berdasarkan pemenuhan kebutuhan

CNN untuk membangun arsitektur yang dapat digunakan pada sistem mobile. Perbedaan mendasar antara arsitektur *MobileNet* dan *Convolutional Neural Network*(CNN) adalah penggunaan lapisan convolutional. Ketebalan filter tersebut dibuat sesuai dengan ketebalan input citra daun yang ada. *MobileNet* saat ini memiliki V1, V2, dan V3.

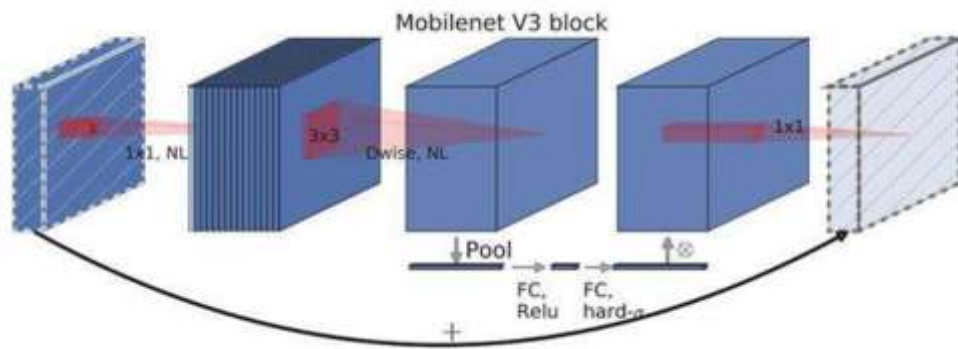
*MobileNetV1* mengenalkan *depthwise separable convolutions* sebagai pengganti *layer* konvolusi tradisional. Lapisan ini membagi filter spasial dari mekanisme penghasil fitur. *Depthwise separable convolutions* dibagi menjadi dua bagian, yaitu *light weight depthwise convolution* dan *heavier 1 X 1 pointwise convolutions* untuk penghasil fitur. *MobileNetV2* menggunakan *linear bottleneck* di antara lapisan dan struktur *invertedresidual* di antara bottleneck.



**Gambar 2. 14** *MobileNet V2* ([www. deep-learning-study.tistory.com](http://www.deep-learning-study.tistory.com))

*MobileNetV3* diatur ke CPU ponsel melalui kombinasi *network architecture search* (NAS) yang beradaptasi dengan *hardware* dan dilengkapi algoritma NetAdapt serta ditingkatkan dengan arsitektur yang dimodifikasi. Peningkatan *network* di *MobileNetV3* ini ada 2 cara, yaitu menghapus lapisan dan *swish non-linearity*. *MobileNetV3* memiliki dua model, yaitu *MobileNetV3-Large* dan *MobileNetV3-Small*.





**Gambar 2. 15** *MobileNet V* (Yuan, et al 2022)

## 2.10 Penelitian Terkait

Adapun penelitian terdahulu dapat dilihat pada **Tabel 2.1** berikut ini.

**Tabel 2. 1** Penelitian Terdahulu

No	Peneliti	Judul Penelitian	Keterangan
1	Susianto, Budi (2018)	DETEKSI FITUR GAWANG PADA ROBOT BERBASIS <i>HOUGH TRANSFORM</i> .	Mendeteksi objek gawang menggunakan metode hough transform dengan tingkat keberhasilan 79 dari 116 kali percobaan. Pendeteksian gawang secara visual terlihat jelas pada jarak di atas 50 cm dan posisi antara $-50^{\circ}$ sampai $50^{\circ}$ .
2	Hapsari, Widi (2016)	TRANSFORMASI <i>HOUGH LINIER</i> UNTUK ANALISIS DAN PENGENALAN BATIK MOTIF PARANG	Hough Transform digunakan untuk menganalisis karakteristik motif batik parang dari distribusi garis pada sudut kemiringan tertentu dan panjang garis dalam kelompok sudut kemiringan tersebut. Bahan penelitian terdiri dari citra sampel data dan citra data untuk motif parang dan non-parang masing-masing. Penelitian ini bertujuan untuk meningkatkan akurasi dalam mengidentifikasi motif batik parang dan non-parang.
3	Sa'diyah, Halimatus, et al (2011)	APLIKASI TRANSFORMASI <i>HOUGH</i> UNTUK DETEKSI GARIS LURUS	Tingkat keberhasilan program transformasi Hough dalam mendeteksi dan merekonstruksi garis lurus pada suatu citra dapat bervariasi, ada yang mencapai 100%, sementara ada pula yang hanya mencapai 20%. Perbedaan ini disebabkan oleh garis lurus yang tidak terdeteksi memiliki nilai besar R, yang merupakan hasil dari transformasi ke koordinat polar, di bawah besar R

			<p>maksimum yang telah dikalikan dengan nilai ambang. Akibatnya, garis lurus tersebut tidak terdeteksi sebagai garis lurus. Namun, hasil rata-rata persentase keberhasilan total adalah sebesar 90%.</p>
4	Lubis, Zulfadli F (2018)	<p>PENENTUAN PELANGGARAN GARIS PEMBATAK PARKIR KENDARAAN RODA EMPAT MENGGUNAKAN ALGORITMA <i>HOUGH TRANSFORM</i></p>	<p>Hasil penelitian dari studi ini, dengan menggunakan 50 data citra sebagai uji, menunjukkan bahwa nilai rata-rata tingkat ketepatan (precision) mencapai 88%, sementara nilai rata-rata tingkat perolehan (recall) mencapai 92%. Selain itu, nilai akurasi keseluruhan mencapai 90%.</p>
5	Rizaldy, Aulia R (2020)	<p>DETEKSI RUANG GARIS KHUSUS (RHK) SEPEDA MOTOR MENGGUNAKAN <i>HOUGH TRANSFORM</i></p>	<p>Hasil dari penelitian ini adalah citra RHK yang terdeteksi secara visual. Citra tersebut akan menampilkan garis yang terdeteksi oleh Hough Transform. Dalam penelitian ini, dilakukan optimasi algoritma untuk meningkatkan ketepatan dalam pendeteksian garis pada citra RHK. Proses tersebut mencakup penghilangan blur dan penyesuaian variabel threshold pada pemrosesan Hough Transform.</p>
6	Sumin, Agus, et al (2020)	<p>PEMBUATAN APLIKASI DETEKSI OBJEK MENGGUNAKAN <i>TENSORFLOW OBJECT DETECTION API</i> DENGAN MEMANFAATKAN <i>SSD MOBILENET V2</i> SEBAGAI MODEL PRA-TERLATIH.</p>	<p>Penelitian ini menggunakan dataset sebanyak 500 gambar dengan membagi menjadi tiga bagian, yaitu train set, validation set, dan test set dengan masing - masing perbandingan sebesar 70% : 20% : 10%. Pelatihan dilakukan dengan bantuan Google Research Colaboratory sebagai virtual machine. Penelitian ini menggunakan Python 3.6.8 sebagai bahasa pemrograman dan memakai beberapa library yang disediakan oleh Python. Dari uji coba yang dilakukan, aplikasi ini memiliki tingkat akurasi sebesar 93.02% pada test set.</p>

## **BAB 3**

### **ANALISIS DAN PERANCANGAN**

#### **3.1 Metode Penelitian**

Dalam penelitian ini penulis menggunakan dua metode untuk mendeteksi garis lapangan sepakbola robot. Metode yang digunakan adalah *Hough Transform* dan *SSD MobileNet V2*.

##### *3.1.1. Metode Hough Transform*

Pada metode Hough Transform, dimana algoritma berfungsi sebagai pendeteksi garis pada lapangan. Sistem dirancang dengan mendeteksi citra sebagai bahan yang akan diproses dalam mendeteksi garis pada citra. Dalam tahap prapemrosesan citra, dilakukan analisis nilai RGB setiap piksel yang direpresentasikan dalam bentuk matriks. Selanjutnya, data ini diproses untuk menentukan perbedaan nilai yang mengindikasikan wilayah gelap dan terang. Nilai-nilai tersebut menjadi dasar untuk langkah-langkah pemrosesan selanjutnya. Metode penghitungan fitur ini melibatkan pengurangan nilai piksel di area yang berwarna putih dengan nilai piksel di area yang berwarna hitam. Proses selanjutnya adalah melakukan konversi nilai piksel citra warna (RGB) menjadi citra keabu-abuan (grayscale) dan selanjutnya dilakukan binerisasi untuk dilakukan pendeteksian tepi Sobel (edge detection). Hasil deteksi tepi diproses untuk deteksi helm dengan algoritma Hough Transform untuk mendeteksi ada atau tidak garis pada citra. Adapun hal-hal yang mempengaruhi dalam mendeteksi garis yaitu :

1. Posisi

Posisi citra garis lapangan dapat bervariasi tergantung pada kondisi saat pengambilan gambar, termasuk pergeseran, perpindahan, dan rotasi. Oleh

karena itu, jika posisi citra tidak akurat, perbaikan perlu dilakukan untuk menyesuaikan posisi citra.

## 2. Kondisi Citra

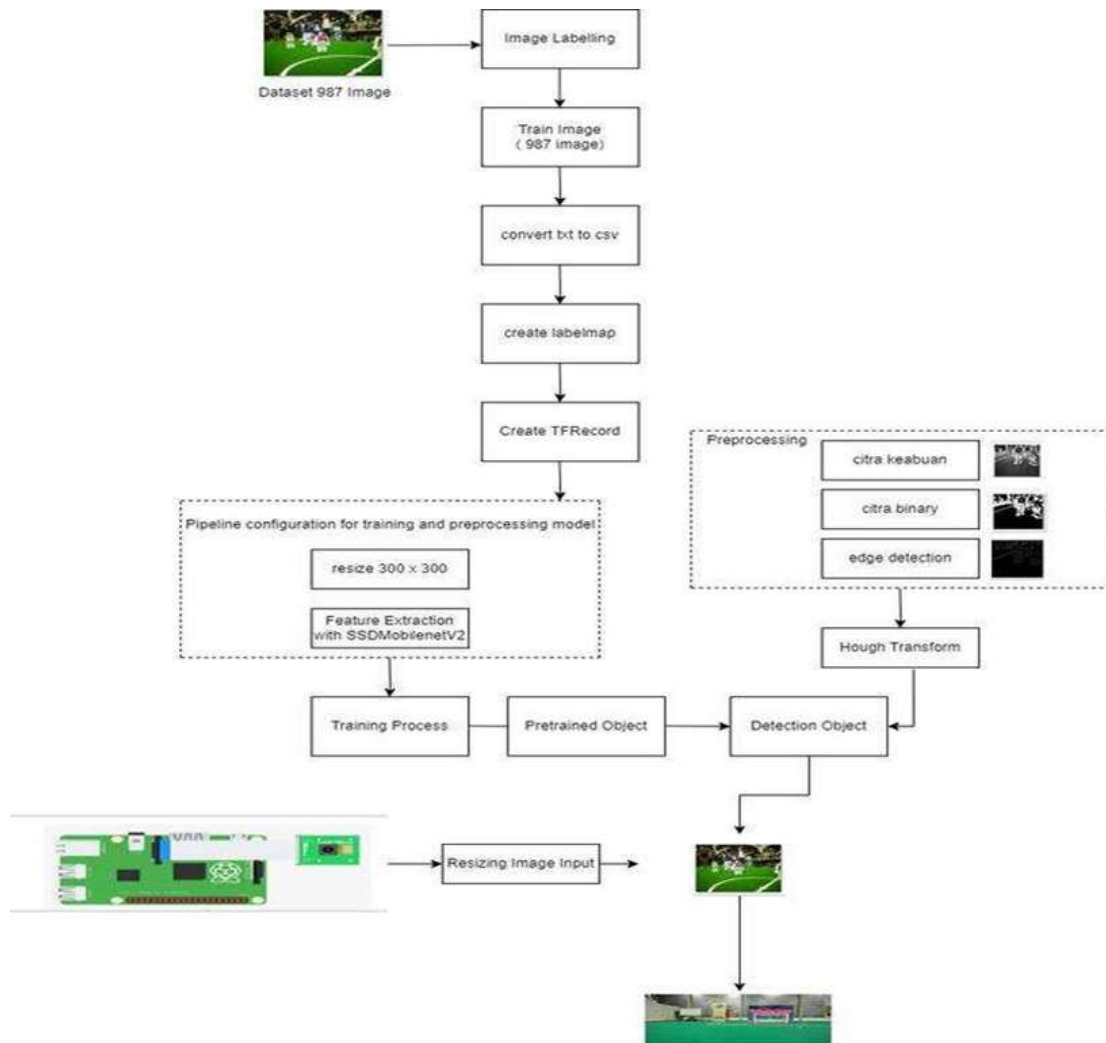
Ketika sebuah citra dibangun, pencahayaan dan karakteristik kamera menentukan hasil dari citra.

### 3.1.2. Metode SSD MobileNet V2

Pada pendekatan menggunakan metode SSD MobileNetV2 untuk merancang sistem pendeteksian objek pada robot sepak bola, langkah-langkah umum dimulai dengan pengumpulan dataset sebanyak 987 citra. Kemudian, citra-citra tersebut diubah ukurannya menjadi 300 x 300 piksel dan dilakukan ekstraksi fitur. Proses ekstraksi fitur dilakukan dengan membatasi piksel objek sehingga hanya objek terbesar yang dipilih. Pembatasan citra dilakukan berdasarkan lebar kontur yang diperoleh dari setiap segmen citra yang dihasilkan sebelumnya. Setelah mendapatkan segmen citra dengan kontur terbesar, langkah selanjutnya adalah mencari titik tengah dari kontur tersebut.

## 3.2 Arsitektur Umum

Adapun arsitektur umum yang digunakan pada penelitian ini dapat dilihat seperti pada gambar dibawah ini.



**Gambar 3. 1** Arsitektur Umum

### 3.3 Proses Penelitian

#### 3.3.1. Pengumpulan Dataset

Dalam penelitian ini, data yang digunakan berupa dokumen gambar, Yang bersumber dari [https://www.kaggle.com/datasets/pietbroemmel/naodevils-segmentation-upper-camera Robocup SPL Instance Segmentation Dataset](https://www.kaggle.com/datasets/pietbroemmel/naodevils-segmentation-upper-camera-Robocup-SPL-Instance-Segmentation-Dataset), selanjutnya dilakukan pemilihan yang akan dijadikan data pengujian sebanyak 987 citra uji. Sampel citra uji yang digunakan pada penelitian ini dapat dilihat pada gambar dibawah ini.



**Gambar 3. 2** Sampel Citra Uji

### 3.3.2. Labelling Data

Setelah mengumpulkan dataset, kemudian dataset diberikan label dan disimpan dalam bentuk file xml yang dapat dilihat seperti gambar dibawah ini.



**Gambar 3. 3** Dataset yang Telah Diberikan Label

Setelah dataset dilabelling maka dataset tersebut disimpan dalam file xml yang dapat dilihat pada gambar dibawah ini.

Name	Date modified	Type	Size
00002_RoboCup2019_AHSWH_Frogther...	18/10/2022 22:13	XML Document	1 KB
00003_GermanOpen2019_HUKs_Quickid...	18/10/2022 22:13	XML Document	1 KB
00007_RoboCup2019_AHSWH_Gaedeid...	18/10/2022 22:13	XML Document	1 KB
00008_RoboCup2019_AHSWH_Frogther...	18/10/2022 22:54	XML Document	1 KB
00011_RoboCup2019_AHSWH_Frogther...	18/10/2022 22:53	XML Document	1 KB
00013_RoboCup2019_AHSWH_Elektre-2...	18/10/2022 22:56	XML Document	1 KB
00014_GermanOpen2019_HUKs_Thames...	18/10/2022 22:56	XML Document	1 KB
00016_RoboCup2019_SandB_Cyclops-1st...	18/10/2022 22:56	XML Document	1 KB
00021_RoboCup2019_AHSWH_Frogther...	18/10/2022 22:56	XML Document	1 KB
00030_GermanOpen2019_HUKs_RedSkul...	18/10/2022 22:56	XML Document	1 KB
00031_RoboCup2019_SandB_Cyclops-1st...	18/10/2022 00:00	XML Document	1 KB
00033_RoboCup2019_SandB_Venom-1st...	18/10/2022 00:00	XML Document	1 KB
00037_RoboCup2019_AHSWH_Frogther...	18/10/2022 00:01	XML Document	1 KB
00038_GermanOpen2019_HUKs_Quickid...	18/10/2022 00:01	XML Document	1 KB
00042_RoboCup2019_AHSWH_Frogther...	18/10/2022 00:01	XML Document	1 KB
00043_GermanOpen2019_HUKs_RedSkul...	18/10/2022 00:02	XML Document	1 KB
00047_RoboCup2019_AHSWH_Elektre-2...	18/10/2022 00:02	XML Document	1 KB
00048_RoboCup2019_AHSWH_Frogther...	18/10/2022 00:02	XML Document	1 KB
00050_RoboCup2019_SandB_Cyclops-2nd...	18/10/2022 00:06	XML Document	1 KB
00051_RoboCup2019_Team-Team_Cyclop...	18/10/2022 00:06	XML Document	1 KB
00052_GermanOpen2019_HUKs_RedSkul...	18/10/2022 00:06	XML Document	1 KB
00053_GermanOpen2019_HUKs_Saboteu...	18/10/2022 00:06	XML Document	1 KB
00055_RoboCup2019_SandB_Cyclops-2nd...	18/10/2022 00:06	XML Document	1 KB
00057_RoboCup2019_SandB_Cyclops-1st...	18/10/2022 00:10	XML Document	1 KB
00060_GermanOpen2019_HUKs_Thames...	18/10/2022 00:10	XML Document	1 KB
00062_GermanOpen2019_HUKs_Quickid...	18/10/2022 00:10	XML Document	1 KB
00063_RoboCup2019_SandB_Venom-2nd...	18/10/2022 00:12	XML Document	1 KB
00064_RoboCup2019_SandB_Venom-2nd...	18/10/2022 00:12	XML Document	1 KB

**Gambar 3. 4** File xml yang sudah dilabeling

### 3.3.3. Identifikasi Garis Pinggir Pada Lapangan Sepakbola Robot

Pada penelitian ini identifikasi garis pinggir (garis putih) lapangan sepak bola robot menggunakan metode *Hough Transform*. Pada penelitian ini digunakan sampel data dari suatu video youtube dengan ekstraksi frame by frame sebagai landasan perancangan image processing yang diterapkan. Proses Identifikasi garis pinggir (garis putih) lapangan sepak bola robot dengan metode *Hough Transform* adalah sebagai berikut :

#### 1. Prepare Image Processing

Mendefinisikan interval warna Hijau dan Putih pada mode gambar HSV. Interval lower dan higher dari warna yang didapatkan berasal dari file `_tuning_color.py`.

```
# define global variable
image = img.copy()
# (hMin = 44 , sMin = 59, vMin = 100), (hMax = 70 , sMax = 130, vMax = 150)
lower_rgb = np.array([45, 50, 100])
upper_rgb = np.array([70, 130, 150])
# (hMin = 37 , sMin = 50, vMin = 111), (hMax = 63 , sMax = 73, vMax = 255)
lower_hsv = np.array([35, 50, 110])
upper_hsv = np.array([65, 75, 255])
```

**Gambar 3. 5** Definiskan Interval Warna Hijau dan Putih Pada Mode Gambar HSV

## 2. Image Processing

### a. Penerapan binary masking dan smoothing image

Pada bagian ini merupakan bagian paling penting dimana suatu frame gambar akan di ekstrak area hijau dan putih yang disertai smoothing method (gaussian, erotion, dan image morfologi), sehingga suatu segmentasi warna tertentu menjadi lebih dominan maupun terhindar dari noise (bintik-bintik kecil dari gambar).

```
image_rgb = cv.cvtColor(img, cv.COLOR_BGR2HSV)
image_rgb = cv.GaussianBlur(image_rgb, _kernel, _iter)
image_rgb = cv.inRange(image_rgb, lower_rgb, upper_rgb)
image_rgb = cv.cvtColor(image_rgb, cv.COLOR_GRAY2RGB)
image_rgb = cv.erode(image_rgb, (33, 33), 15)

image_hsv = cv.cvtColor(img, cv.COLOR_BGR2HSV)
image_hsv_blur = cv.GaussianBlur(image_hsv, _kernel, _iter)
image_masking = cv.inRange(image_hsv_blur, lower_hsv, upper_hsv)

kernel_morph = cv.getStructuringElement(cv.MORPH_RECT, (3, 3))
image_masking_close = cv.morphologyEx(image_masking, cv.MORPH_OPEN, kernel_morph, 1)
image_masking_close = cv.cvtColor(image_masking_close, cv.COLOR_GRAY2RGB)

image_gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
image_gray_blur = cv.GaussianBlur(image_gray, _kernel, _iter)
image_edge = cv.Canny(image_gray_blur, low_canny, high_canny)
line_image = np.zeros_like(img)
```

**Gambar 3. 6** Penerapan *Binary Masking* dan *Smoothing Image*

### b. Penerapan Hough Line Transform

Untuk memperbesar kemungkinan terdeteksi garis putih pada area sepak bola, maka penulis juga mengimplementasikan metode hough line transformation jenis “garis” dengan interval trigonometri tertentu sehingga membantu proses pendeteksian garis dari aspek matematis, diluar dari aspek segmentasi warna.

```
lines = cv.HoughLinesP(image_edge, 0.5, np.pi / 180, 15, np.array([]), 150, 7)

try:
    for line in lines:
        for x1, y1, x2, y2 in line:
            cv.line(line_image, (x1, y1), (x2, y2), (255, 255, 255), 3)
except:
    pass

masking_line = cv.bitwise_or(image_masking_close, line_image)

# HOFF: overlay for masking mid line.
result_overlay = overlay_image(masking_line, MODEL)
masking_line = cv.bitwise_and(masking_line, result_overlay)

if color is not False:
    masking_line[np.where((masking_line == [255, 255, 255]).all(axis = 2))] = color
```

**Gambar 3. 7** Penerapan *Hough Line Transformation*



### 3. Citra Keabuan (*Grayscale*)

File citra masukan dari dataset Robocup Soccer SPL selanjutnya dilakukan proses konversi ke ruang pewarnaan keabuan. Tahapan perhitungan citra RGB ke citra keabuan (*grayscale*) dengan menggunakan rumus pada Gambar 3.8.

$$I_{\text{gr}}(x, y) = \frac{I_R(x, y) + I_G(x, y) + I_B(x, y)}{3}$$

**Gambar 3. 8** Rumus Konversi Citra Keabuan

### 4. Citra Biner

Setelah file citra diubah ke pewarnaan grayscale maka tahapan selanjutnya adalah melakukan proses thresholding citra ke bentuk biner dengan menggunakan persamaan dibawah ini, dimana T merupakan nilai ambang yakni 128.

$$g(x, y) = \begin{cases} 1 & \text{jika } f(x, y) \geq T \\ 0 & \text{jika } f(x, y) < T \end{cases}$$

#### 3.3.4. Convert Xml to Csv

Mengubah anotasi hasil labelling txt ke csv yang berguna untuk meningkatkan efisiensi dalam penyimpanan, transfer, analisis, dan pengelolaan data.

1	Filename	width	height	class	xmin	ymin	xmax	ymax
2	00002_RoboCup201	640	480	garis tengah	252	232	548	290
3	00003_GemanOper	640	480	garis tengah	143	230	640	380
4	00007_RoboCup201	640	480	garis tengah	1	399	638	480
5	00008_RoboCup201	640	480	garis tengah	1	350	637	479
6	00011_RoboCup201	640	480	garis tengah	2	78	638	478
7	00012_RoboCup201	640	480	garis tengah	2	220	639	261
8	00014_GemanOper	640	480	garis tengah	2	349	639	477
9	00016_RoboCup201	640	480	garis tengah	5	333	633	480
10	00021_RoboCup201	640	480	garis tengah	1	201	619	260
11	00030_GemanOper	640	480	garis tengah	318	280	638	440
12	00031_RoboCup201	640	480	garis tengah	43	328	638	480
13	00033_RoboCup201	640	480	garis tengah	1	128	639	174
14	00037_RoboCup201	640	480	garis tengah	1	131	441	188
15	00038_GemanOper	640	480	garis tengah	1	274	638	480
16	00042_RoboCup201	640	480	garis tengah	1	305	640	480
17	00045_GemanOper	640	480	garis tengah	184	35	637	250
18	00047_RoboCup201	640	480	garis tengah	1	272	628	480
19	00048_RoboCup201	640	480	garis tengah	2	136	405	186
20	00058_RoboCup201	640	480	garis tengah	1	85	635	136
21	00051_RoboCup201	640	480	garis tengah	372	332	640	479

**Gambar 3. 9** Convert xml to csv

### 3.3.5. Membuat Labelmap

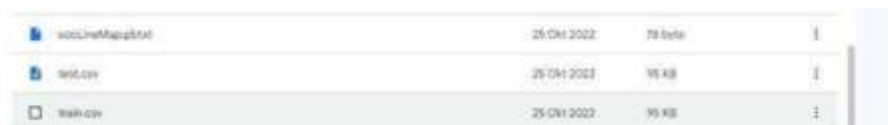
Pembuatan map label untuk class object. berguna untuk memberi tahu nama objek yg terdeteksi.

```
item {
    id: 1
    name: "garis tengah"
    display_name : "Garis Tengah"
}
```

**Gambar 3. 10** Membuat Labelmap

### 3.3.6. Membuat TFRecord

Kegunaan TFRecord adalah format data yang digunakan dalam TensorFlow untuk menyimpan data dalam bentuk serial binary. Pada metode Single Shot Multibox Detector (SSD), TFRecord digunakan penulis untuk menyimpan dataset yang digunakan untuk pelatihan model.



tfrecordMapLabel	25 Okt 2022	78 byte
tfrecord	25 Okt 2022	95 KB
tfrecord	25 Okt 2022	95 KB

**Gambar 3. 11** Membuat TFRecord

### 3.3.7. Identifikasi Garis Tengah Pada Lapangan Sepakbola Robot

Pada penelitian ini identifikasi garis tengah lapangan sepak bola robot menggunakan metode Single Shot Multibox Detector (SSD MobileNetV2). Pada penelitian ini digunakan sampel data dari suatu video youtube dengan ekstraksi frame by frame sebagai landasan perancangan image processing yang diterapkan. Proses Identifikasi garis tengah lapangan sepak bola robot dengan metode SSD MobileNetV2 adalah sebagai berikut :

### 1. Jumlah Steps yang digunakan

Pada penelitian ini jumlah step yang digunakan untuk mengidentifikasi garis tengah lapangan sepak bola robot menggunakan metode SSD MobileNetV2 dapat dilihat pada Gambar 3.12 berikut.

```

}
fine_tune_checkpoint: "nn-models/model.ckpt"
fine_tune_checkpoint_type: "detection"
# Note: The below line limits the training process to 200K steps, which we
# empirically found to be sufficient enough to train the pets dataset. This
# effectively bypasses the learning rate schedule (the learning rate will
# never decay). Remove the below line to train indefinitely.
num_steps: 10000
data_augmentation_options {
  random_horizontal_flip {
  }
}
data_augmentation_options {
  ssd_random_crop {
  }
}
}

train_input_reader: {
  tf_record_input_reader {
    input_path: "data/train.record"
  }
  label_map_path: "data/soclineMap.pbtxt"
}

eval_config: {
  num_examples: 8000
  # Note: The below line limits the evaluation process to 10 evaluations.
  # Remove the below line to evaluate indefinitely.
  max_evals: 10
}

```

**Gambar 3. 12** Jumlah step yang digunakan

### 2. Jumlah Class Object

Pada penelitian ini jumlah class objek yang digunakan untuk mengidentifikasi garis tengah lapangan sepak bola robot menggunakan metode SSD MobileNetV2, nilai num\_classes diatur ke 1. Ini berarti model akan dilatih untuk mendeteksi objek dari satu kelas saja. Seperti pada gambar berikut :

```

model {
  ssd {
    num_classes: 1
    box_coder {
      faster_rcnn_box_coder {
        y_scale: 10.0
        x_scale: 10.0
        height_scale: 5.0
        width_scale: 5.0
      }
    }
    matcher {
      argmax_matcher {
        matched_threshold: 0.5
        unmatched_threshold: 0.5
        ignore_thresholds: false
        negatives_lower_than_unmatched: true
        force_match_for_each_row: true
      }
    }
    similarity_calculator {
      iou_similarity {
      }
    }
    anchor_generator {
      ssd_anchor_generator {
        num_layers: 6
        min_scale: 0.2
        max_scale: 0.95
        aspect_ratios: 1.0
        aspect_ratios: 2.0
        aspect_ratios: 0.5
        aspect_ratios: 3.0
        aspect_ratios: 0.3333
        aspect_ratios: 0.15
      }
    }
  }
}

```

**Gambar 3. 13** Jumlah *class object* yang digunakan

### 3. Jenis Fitur Ekstraksi

Pada penelitian ini jenis fitur ekstraksi yang digunakan untuk mengidentifikasi garis tengah lapangan sepak bola robot menggunakan metode SSD MobileNetV2 dapat dilihat pada Gambar 3.14 berikut.

```

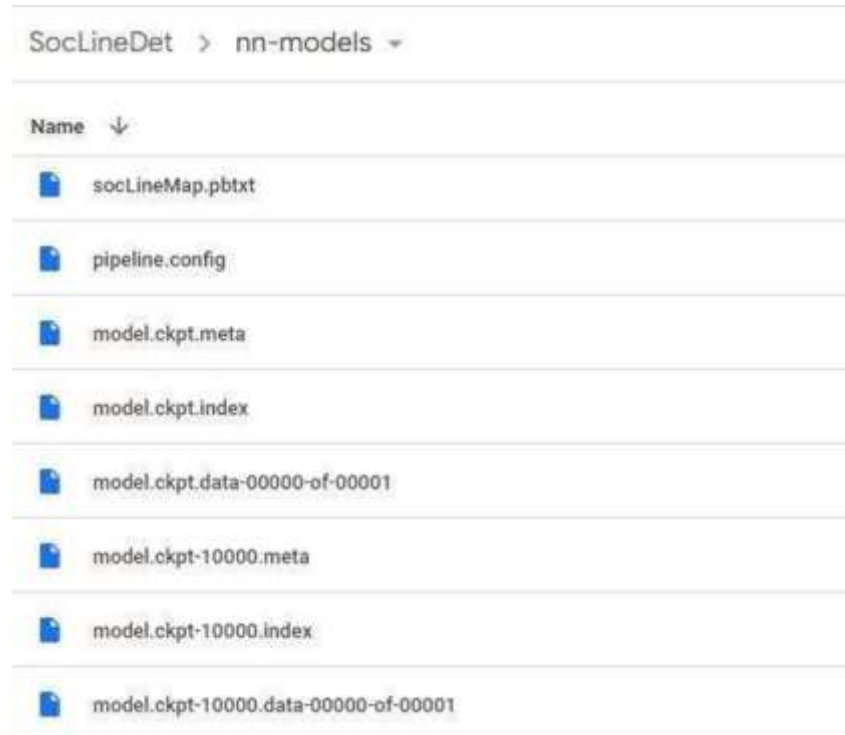
feature_extractor {
  type: 'ssd_mobilenet_v2'
  min_depth: 16
  depth_multiplier: 1.0
  conv_hyperparams {
    activation: RELU_6,
    regularizer {
      l2_regularizer {
        weight: 0.0004
      }
    }
  }
}

```

**Gambar 3. 14** Jenis Fitur *Ekstraksi*

#### 4. Model Hasil Training

Pada penelitian ini model hasil training yang digunakan untuk mengidentifikasi garis tengah lapangan sepak bola robot menggunakan metode SSD MobileNetV2 dapat dilihat pada Gambar 3.15 berikut.



**Gambar 3. 15** Model Hasil *Training*

#### 5. Pembuatan Inference Model Hasil Training

Pada penelitian ini pembuatan inference model hasil training yang digunakan untuk mengidentifikasi garis tengah lapangan sepak bola robot menggunakan metode SSD MobileNetV2 dapat dilihat pada Gambar 3.16 berikut.



#### 7. Finalisasi Overlay Pendeteksian Dengan Gambar

Binary masking overlay yang telah diterapkan berbagai metodologi, sesuai penjelasan di subbab sebelumnya, akan digabungkan dan ditimpa sebagai overlay pada suatu frame gambar dengan interval warna yang telah terdefinisikan dengan contoh hasil frame seperti dibawah (versi sebelum penerapan model mobilenet guna menghilangkan overlay “area tengah” lapangan).

```
final_overlay = cv.dilate(cv.medianBlur(cv.cvtColor(image_rgb, cv.COLOR_RGB2GRAY), 55), None, 15)
masking_line = np.bitwise_and(masking_line, cv.cvtColor(final_overlay, cv.COLOR_GRAY2RGB))
result = cv.addWeighted(image, 0.8, masking_line, 1, 0)
```

**Gambar 3. 18** *Finalisasi Overlay*

## **BAB 4**

### **IMPLEMENTASI DAN PENGUJIAN**

#### **4.1 Implementasi Sistem**

Pada bagian ini akan dijabarkan hasil akhir sistem yang diperoleh dari penerapan dan pengujian perangkat lunak deteksi garis lapangan dengan dataset RoboCup Soccer menggunakan Metode *Hough Transform* dan *SSD Mobilenet v2* yang sudah dijelaskan sebelumnya. Adapun data yang diolah adalah citra RoboCup Soccer SPL memiliki ukuran 300 x 300 piksel dengan format ekstensi file \*.png.

#### **4.2 Perangkat Penelitian**

##### *4.2.1. Kebutuhan Perangkat Keras*

Sebagai alat bantu dalam penelitian ini, keperluan perangkat keras dengan spesifikasi minimum, sebagai berikut :

Operating System	: Windows 10
Processor	: Intel® Core™ i5-7200U 2.5 GHz up to 3.1 GHz
Kartu Grafis	: NVIDIA GEFORCE 940 MX
RAM	: 4+4 GB DDR4
Penyimpanan	: 1TB
Resolusi	: 1366 x 768 (60Hz)



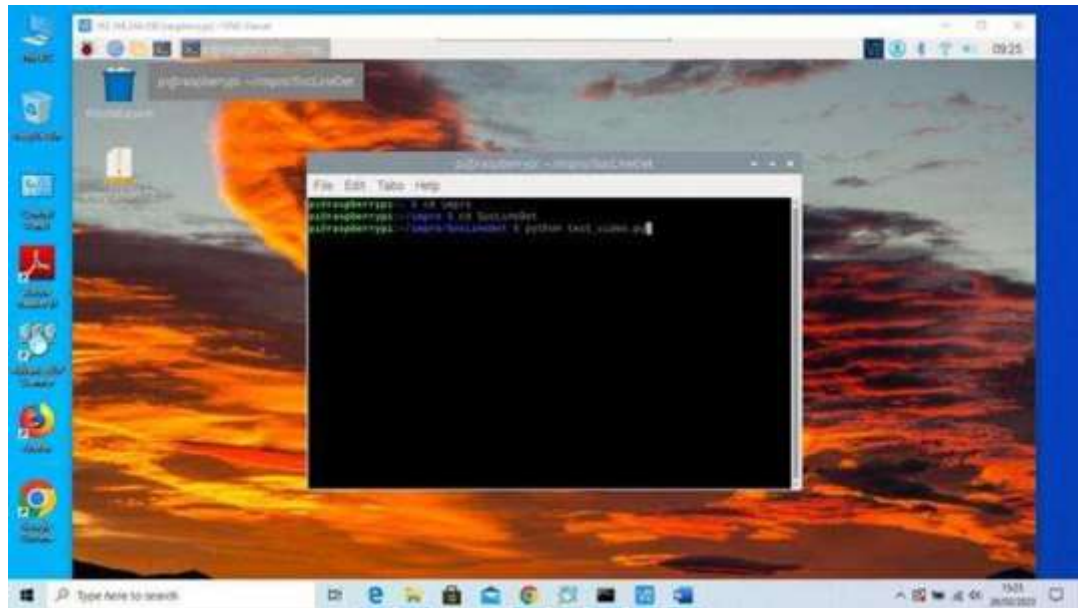
#### 4.2.2. Implementasi Perangkat Lunak

Implementasi Perangkat Lunak Data dan model config dibuat dalam format yaml, kemudian disimpan dalam dua file. Tampilan Sistem Implementasi dalam proses pengujian metode *hough transform* dan *ssd mobilenet v2* diperlukan untuk memperoleh hasil kinerja sistem yang disesuaikan untuk mendapatkan hasil akhir deteksi garis pada citra lapangan Robocup Soccer SPL. Tampilan utama aplikasi dapat dilihat pada Gambar 4.1 dibawah ini :



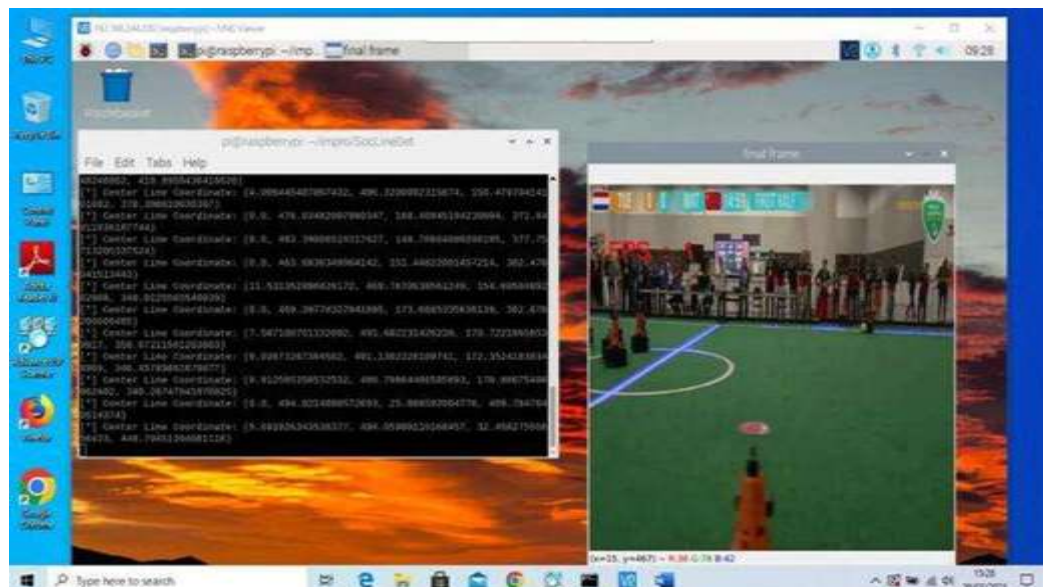
**Gambar 4. 1** Tampilan Utama Sistem

Ketika semua sudah siap maka aplikasi dapat dijalankan dan aplikasi akan menampilkan jendela tanda proses berjalan seperti gambar dibawah ini.



**Gambar 4. 2** Tampilan Aplikasi Saat Menjalankan Program

Setelah jendela terbuka dan menampilkan kamera maka akan langsung mendeteksi objek garis.. Tempat pengujian dilakukan pada karpet hijau yang sudah di bentuk menjadi lapangan tanding robot sepak bola beroda yang sesuai dengan peraturan Kontes Robot Sepak bola beroda (KRSBIB).









**Gambar 4. 3** Tampilan pendeteksian objek garis koordinat garis tengah lapangan

#### 4.1 Pengujian

Berdasarkan dataset citra yang sudah ada, maka akan dilakukan proses pengujian terhadap objek citra RoboCup Soccer SPL. Pada tahap ini akan dilakukan pengujian terhadap aplikasi yang telah dibangun dengan cara menguji 50 data citra. Dapat dilihat pada **Tabel 4.1** di bawah ini.

**Tabel 4. 1** Contoh Citra Pengujian *RoboCup Soccer*



No	Nama Citra	Ukuran	Pewarnaan
1.		300 x 300	RGB
2.		300 x 300	RGB
3.		300 x 300	RGB
4.		300 x 300	RGB

5.		300 x 300	RGB
6.		300 x 300	RGB
7.		300 x 300	RGB
8.		300 x 300	RGB
9.		300 x 300	RGB
10.		300 x 300	RGB








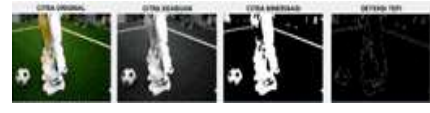
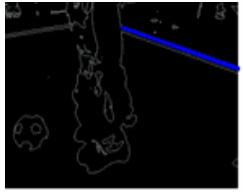


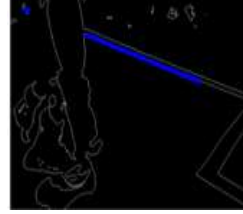


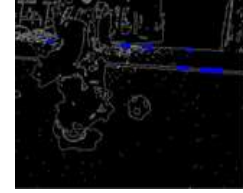





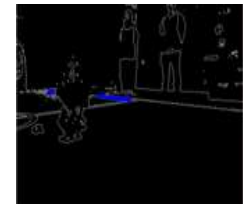
## 4.2 Hasil Deteksi






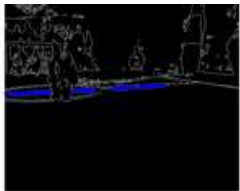


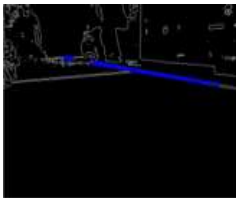


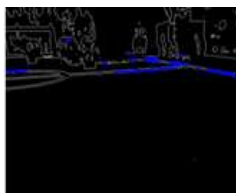






Jumlah dataset citra yang akan dilakukan pengujian untuk deteksi garis lapangan pada dataset RoboCup Soccer SPL sebanyak 50 citra, contoh citra hasil deteksi garis dapat dilihat pada tabel dibawah ini.






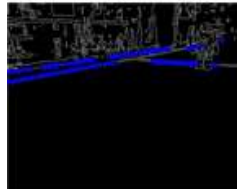


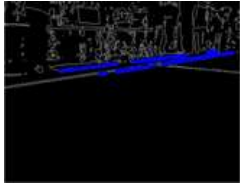


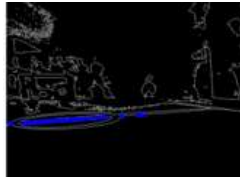


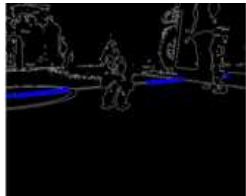


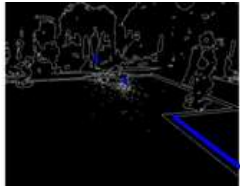
**Tabel 4. 2** Hasil Deteksi Garis Lapangan

No	Citra	Proses Deteksi	Hasil Deteksi
1.			
2.			
3.			
4.			
5.			



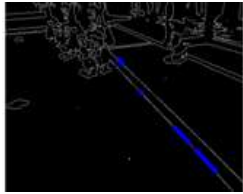





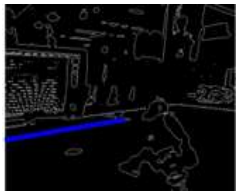





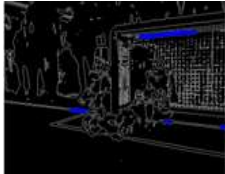


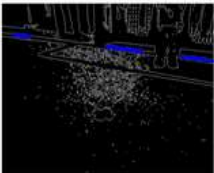



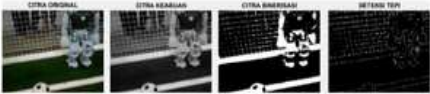
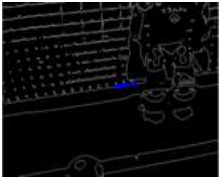

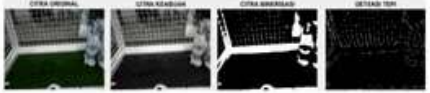
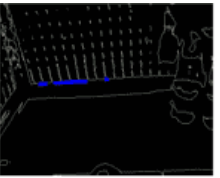





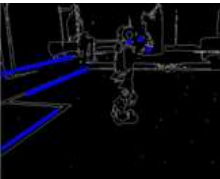


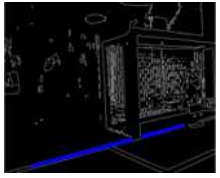


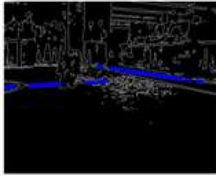
No	Citra	Proses Deteksi	Hasil Deteksi
6.			
7.			
8.			
9.			
10.			
11.			
12.			





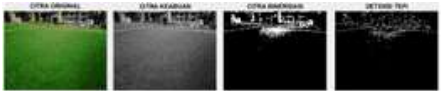
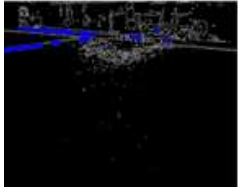


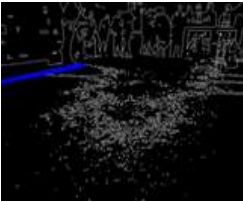








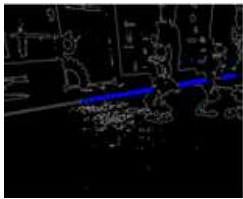
No	Citra	Proses Deteksi	Hasil Deteksi
13.			
14.			
15.			
16.			
17.			
18.			



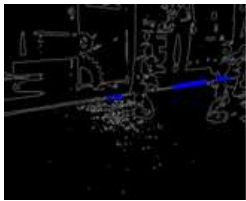


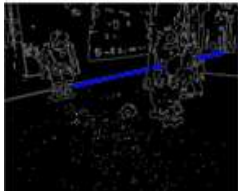


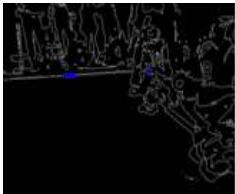

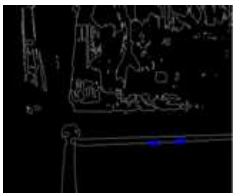


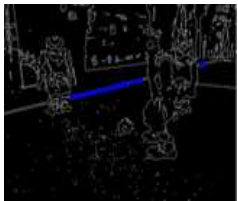


No	Citra	Proses Deteksi	Hasil Deteksi
19.			
20.			
21.			
22.			
23.			
24.			



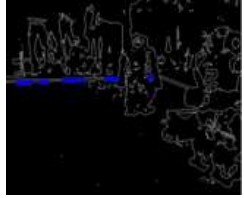

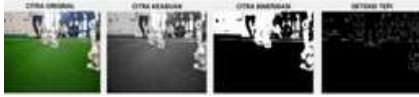



No	Citra	Proses Deteksi	Hasil Deteksi
25.			
26.			
27.			
28.			
29.			
30.			

No	Citra	Proses Deteksi	Hasil Deteksi
31.			
32.			
33.			
34.			
35.			
36.			

No	Citra	Proses Deteksi	Hasil Deteksi
37.			
38.			
39.			
40.			
41.			
42.			

No	Citra	Proses Deteksi	Hasil Deteksi
43.			
44.			
45.			
46.			
47.			
48.			

No	Citra	Proses Deteksi	Hasil Deteksi
49.			
50.			

### 4.3 Perhitungan *Confusion Matrix*

*Confusion Matrix* merupakan teknik atau cara yang dilakukan untuk dapat mengukur kinerja sistem dalam hal penelitian ini berkaitan dengan deteksi garis pada citra RoboCup Soccer SPL, sehingga dapat dilakukan proses perhitungan dengan menggunakan rumus sebagai berikut :

**Tabel 4. 3** TP, TN, FP dan FN *Confusion Matrix*

Kelas	Terklasifikasi Positif	Terklasifikasi Negatif
Positif	TP (True Positive)	FN (False Negative)
Negatif	FP (False Positive)	TN (True Negative)

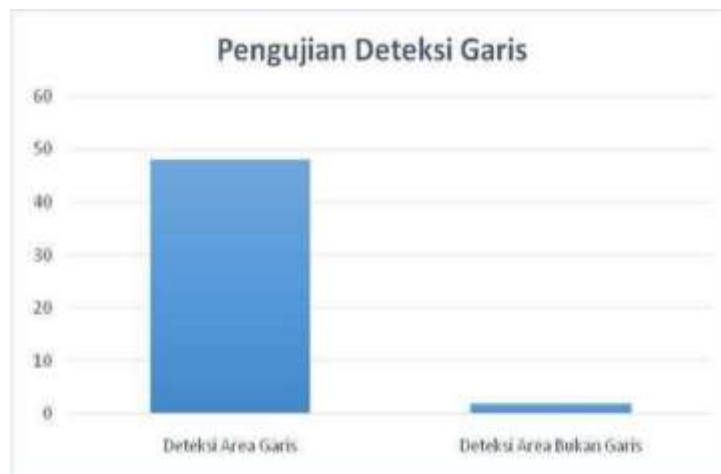
**Tabel 4. 4** Perhitungan *Confusion Matrix* Deteksi Garis

Kelas	Terklasifikasi Positif	Terklasifikasi Negatif
Positif	Terdeteksi Sebagai Garis Tepi Lapangan	Tidak Ada Hasil Terdeteksi
Negatif	2 atau Lebih Terdeteksi Sebagai Garis	Terdeteksi Sebagai Garis, Tapi Bukan Tepi Lapangan



**a) Accuracy**

$$\begin{aligned}
 Accuracy &= \frac{\text{jumlah data yang benar}}{\text{jumlah keseluruhan data uji}} \times 100\% \\
 &= \frac{48}{50} \times 100 \\
 &= 96\%
 \end{aligned}$$



**Gambar 4. 4** Visualisasi Citra Hasil Pengujian

**b) Precision**

$$\begin{aligned}
 Precision &= \frac{TP}{TP + FP} \times 100\% \\
 &= \frac{32}{32 + 12} \times 100\% \\
 &= \frac{32}{44} \times 100\% = 72\%
 \end{aligned}$$

**c) Recall**

$$\begin{aligned}
 Recall &= \frac{TP}{TP + FN} \times 100\% \\
 &= \frac{32}{32 + 0} \times 100\% \\
 &= \frac{32}{32} \times 100\% = 100\%
 \end{aligned}$$

**d) *F1 Score***

$$\begin{aligned} F1\ Score &= 2 \times \left( \frac{0,72 \times 1,00}{0,72 + 1,00} \right) \times 100\% \\ &= 83\% \end{aligned}$$

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1 Saran**

Proses pengujian aplikasi untuk mendeteksi garis pada citra RoboCup Soccer SPL dengan metode *hough transform* dan *ssd mobilenet v2* telah dilaksanakan dalam penelitian tugas akhir ini, sehingga dapat diambil kesimpulan sebagai berikut :

1. Metode *Hough Transform* dan SSD Mobilenet v2 dapat melakukan pendeteksian garis pada citra RoboCup Soccer SPL dengan implementasi pada sistem.
2. Nilai Akurasi pendeteksian garis sebesar 96 %.

#### **5.2 Kesimpulan**

Saran yang dapat diberikan untuk penelitian selanjutnya perlu dilakukan kombinasi metode yang lain dalam pendeteksian citra garis dan peningkatan aplikasi pendeteksian garis pada RoboCup Soccer secara real-time, agar dapat meminimalisir kesalahan pada aturan permainan sepak bola. Dapat juga dilakukan kombinasi antara pendeteksian objek bola dan garis lapangan agar membuat system seperti *VAR (Video Assistant Referee)*.



## DAFTAR PUSTAKA

- Aningtiyas, P. R., Sumin, A., & Wirawan, S. (2020). Pembuatan Aplikasi Deteksi Objek Menggunakan TensorFlow Object Detection API dengan Memanfaatkan SSD MobileNet V2 Sebagai Model Pra-Terlatih: Array. *Jurnal Ilmiah Komputasi*, 19(3), 421-430.
- Ahmad, S., Hayat, M. F., Qureshi, M. A., Asef, S., & Saleem, Y. (2021). Enhanced halftone-based secure and improved visual cryptography scheme for colour/binary Images. *Multimedia Tools and Applications*, 80(21), 32071-32090.
- Ardianto, V.E.J. (2018). “ROBOT SEPAKBOLA BERODA BERBASIS *RASPBERRY PI 3*”. Skripsi. Fakultas Sains dan Teknologi Jurusan Teknik Elektro Universitas Sanata Dharma Yogyakarta.
- Arifin, A., & Budiman, B. (2011). “Edge Detection Menggunakan Metode Roberts”. Cross. *Jurnal SIFO Mikroskil*, 12(1), 39-48.
- Bahaghighat, M., Xin, Q., Motamedi, S. A., Zanjireh, M. M., & Vacavant, A. (2020). Estimation of wind turbine angular velocity remotely found on video mining and convolutional neural network. *Applied Sciences*, 10(10), 3544.
- Budiono, I., Ilham, L.H., Muhtadin, Muhammad, A. (2015). “Penentuan Posisi dalam Lapangan KRSBI 2015 Menggunakan Platform Darwin-OP Berbasis Pendeteksian Gawang dan Garis”. Conference: The 3rd Indonesian Symposium On Robot Soccer CompetitionAt: Universitas Muhammadiyah Yogyakarta.
- Fayyadh, M., Suryana, U.(2015). “PERANCANGAN SISTEM OTOMATISASI PALANG PINTU KERETA API BERBASIS MOTION DETECTION ” . *ISSN : 2355-9365 e-Proceeding of Engineering : Vol.2, No.1 April 2015 : 291*
- Guennouni, S., Ahaitouf, A., & Mansouri, A. (2014, October). Multiple object detection using OpenCV on an embedded platform. In *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)* (pp. 374-377). IEEE.

- Hapsari, W. (2015). “TRANSFORMASI HOUGH LINEAR UNTUK ANALISIS DAN PENGENALAN BATIK MOTIF PARANG”. JURNAL INFORMATIKA Vol. 11, No. 2, November 2015.
- Indriani, F., Fitri, U., Yuita, A.S. (2018). “Deteksi *Zebra Cross* Pada Citra Digital Dengan Menggunakan Metode *Hough Transform*”. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 2, No. 6, Juni 2018, hlm. 2374-2380
- Kobayashi, Shinsuke., Muramatsu, Shogo., Kikuchi, Hisakazu., Iwahashi, Masahiro. (2007). “*WATER LEVEL TRACKING WITH CONDENSATION ALGORITHM*”. Conference: IWAIT 2007 , International Workshop on Advanced Image Technology.
- Liantoni, F. (2015). “ Deteksi tepi citra daun mangga menggunakan algoritma Ant Colony Optimization “. *Seminar Nasional Sains dan Teknologi Terapan III* (Vol. 3, pp. 411-418).
- Lubis, Z.F. (2018). “PENENTUAN PELANGGARAN GARIS PEMBATAS PARKIR KENDARAAN RODA EMPAT MENGGUNAKAN ALGORITMA *HOUGH TRANSFORM*”. Skripsi. Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
- Marbun, R.E. (2021). “ Segmentasi Citra Spektrum Menggunakan Algoritma Region Growing “. Vol 1, No 9, Februari 2021, Hal 424-429.
- Pan, X., & Zhao, J. (2018). High-resolution remote sensing image classification method based on convolutional neural network and restricted conditional random field. *Remote Sensing*, 10(6), 920.
- Pranata, A., & Astuti, E. Z. (2017). Pengolahan Citra Berbasis Deteksi Tepi Prewitt Pada Gambar Gigi Manusia. *Eksplora Informatika*, 6 (2), 98–105.
- Ridwan., Sitorus, S.H., Midyanti, D.M. (2020). “ Penerapan Metode Edge Detection Kirsch dan Robinson Untuk Mendeteksi Keaslian Uang Kertas Rupiah “. *Jurnal Komputer dan Aplikasi* Volume 08, No. 01 (2020), hal 23-33.
- Rizaldy, A.R. (2020). “DETEKSI GARIS RUANG HENTI KHUSUS (RHK) SEPEDA MOTOR MENGGUNAKAN *HOUGH TRANSFORM*“. Skripsi. Fakultas Teknik Telekomunikasi dan Elektro Institut Teknologi Telkom Purwokerto.

- Sa'adiyah, H., Rizal, R.I., Achmad, H. (2011). "APLIKASI *TRANSFORMASI HOUGH* UNTUK DETEKSI GARIS LURUS". Skripsi. Fakultas Teknik Universitas Diponegoro.
- Susianto, Budi.(2018). "DETEKSI FITUR GAWANG PADA ROBOT BERBASIS *HOUGH TRANSFORM*". Skripsi. Fakultas Teknologi dan Informatika Institut Bisnis dan Informatika STIKOM Surabaya.
- Wakhidah, N. (2011). "Implementasi Metode Hough Transform pada Image Segmentation". JURNAL TRANSFORMATIKA, Volume 9, No.1, Juli 2011 : 46 – 54.
- Zhang, Y., Yuan, B., Wang, Y., Wang, F., & Guo, J. (2022). A Real-Time Oil Content Analysis Method of Cuttings Based on Deep Learning. *IEEE Access*, 10, 132083-132094.



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,  
RISET, DAN TEKNOLOGI  
UNIVERSITAS SUMATERA UTARA  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

Jalan Universitas No. 9A Gedung A, Kampus USU Medan 20155, Telepon: (061) 821007  
Laman: <http://Fasilkomti.usu.ac.id>

KEPUTUSAN  
DEKAN FAKULTAS ILMU KOMPUTER  
DAN TEKNOLOGI INFORMASI  
NOMOR : 2717/UN5.2.14.D/SK/SPB/2024  
DEKAN FAKULTAS ILMU KOMPUTER  
DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA

- Membaca : Surat Permohonan Mahasiswa Fasilkom-TI USU tanggal 11 Juli 2024 perihal permohonan ujian skripsi:  
Nama : ADE RIZKY  
NIM : 171402134  
Program Studi : Sarjana (S-1) Teknologi Informasi  
Judul Skripsi : Deteksi Garis Lapangan Sepakbola Robot Dengan Metode Hough Transform dan Ssd mobilenet v2
- Memperhatikan : Bahwa Mahasiswa tersebut telah memenuhi kewajiban untuk ikut dalam pelaksanaan Meja Hijau Skripsi Mahasiswa pada Program Studi Sarjana (S-1) Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara TA 2023/2024.
- Menimbang : Bahwa permohonan tersebut diatas dapat disetujui dan perlu ditetapkan dengan surat keputusan
- Mengingat : 1. Undang-undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional.  
2. Peraturan Pemerintah Nomor 17 tahun 2010 tentang pengelolaan dan penyelenggara pendidikan.  
3. Keputusan Rektor USU Nomor 03/UN5.1.R/SK/SPB/2021 tentang Peraturan Akademik Program Sarjana Universitas Sumatera Utara.  
4. Surat Keputusan Rektor USU Nomor 1876/UN5.1.R/SK/SDM/2021 tentang pengangkatan Dekan Fasilkom-TI USU Periode 2021-2026

MEMUTUSKAN

- Menetapkan :  
Pertama : Membentuk dan mengangkat Tim Penguji Skripsi mahasiswa sebagai berikut:  
Ketua : Baihaqi Siregar S.Si., MT.  
NIP: 197901082012121002  
Sekretaris : Niskarto Zendrato S.Kom., M.Kom  
NIP: 198909192018051001  
Anggota Penguji : Seniman S.Kom., M.Kom.  
NIP: 198705252014041001  
Anggota Penguji : Dedy Arisandi ST., M.Kom.  
NIP: 197908312009121002  
Moderator : -  
Panitera : -
- Kedua : Segala biaya yang diperlukan untuk pelaksanaan kegiatan ini dibebankan pada Dana Penerimaan Bukan Pajak (PNPB) Fasilkom-TI USU Tahun 2024.
- Ketiga : Keputusan ini berlaku sejak tanggal ditetapkan dengan ketentuan bahwa segala sesuatunya akan diperbaiki sebagaimana mestinya apabila dikemudian hari terdapat kekeliruan dalam surat keputusan ini.

- Tembusan :
- 1. Ketua Program Studi Sarjana (S-1) Teknologi Informasi
  - 2. Yang bersangkutan
  - 3. Arsip

Medan  
Ditandatangani secara elektronik oleh:  
Dekan



Maya Silvi Lydia  
NIP 197401272002122001