

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Sistem Deteksi Gerakan Berbasis *Computer Vision*

2.1.1.1 *Computer Vision*

Computer Vision merupakan ilmu dan teknologi mesin yang melihat. Dalam hal ini, arti «melihat» yaitu dimana mesin dapat mengekstrak informasi melalui gambar yang digunakan untuk berbagai macam fungsi tertentu. Sebagai disiplin teknologi, *computer vision* berusaha untuk menerapkan teori dan model untuk pembangunan sistem *computer vision*. *Computer vision* diartikan sebagai teknologi yang mempelajari bagaimana komputer dapat melihat dan mengenali obyek yang diamati (Setiawan dkk., 2021).

2.1.1.2 *Artificial Intelligence (AI)*

Artificial Intelligence (AI), atau dalam bahasa Indonesia dikenal sebagai Kecerdasan Buatan, adalah cabang ilmu komputer yang bertujuan untuk mengembangkan sistem dan mesin yang mampu melakukan tugas yang biasanya memerlukan kecerdasan manusia. *AI* melibatkan penggunaan algoritma dan model matematika untuk memungkinkan komputer dan sistem lainnya untuk belajar dari data, mengenali pola, dan membuat keputusan yang cerdas (Eriana dkk., t.t.).

2.1.1.3 Tarung Derajat

Tarung Derajat adalah Ilmu, tindakan moral dan sikap hidup yang memanfaatkan kemampuan daya gerak otot, otak, dan nurani secara realistis dan rasional, terutama pada upaya penguasaan dan penerapan 5 (lima) daya gerak moral, yaitu : kekuatan – kecepatan – ketepatan – keberanian – keuletan pada sistem ketahanan dan pertahanan diri yang agresif dan dinamis pada bentuk – bentuk gerakan pukulan, tendangan, tangkisan, bantingan, kuncian, hindaran, dan gerakan anggota tubuh penting lainnya yang terpola pada teknik, taktik, dan strategi bertahan dan menyerang yang praktis dan efektif bagi suatu ilmu olahraga seni beladiri (Setiawan, 2021).

2.1.1.4 Pose Estimation

Pose Estimation adalah sebuah teknik pada *computer vision* yang digunakan untuk mendeteksi gerakan dari seseorang ataupun sebuah objek dengan mengidentifikasi lokasi *landmarks* atau titik tubuh seorang (Romindo dkk., 2022).

Tujuannya adalah untuk membentuk representasi seperti kerangka dari tubuh manusia dan kemudian memprosesnya lebih lanjut untuk aplikasi khusus tugas (Pringsewu dkk., t.t.).

2.1.1.5 Convolutional Neural Network (CNN)

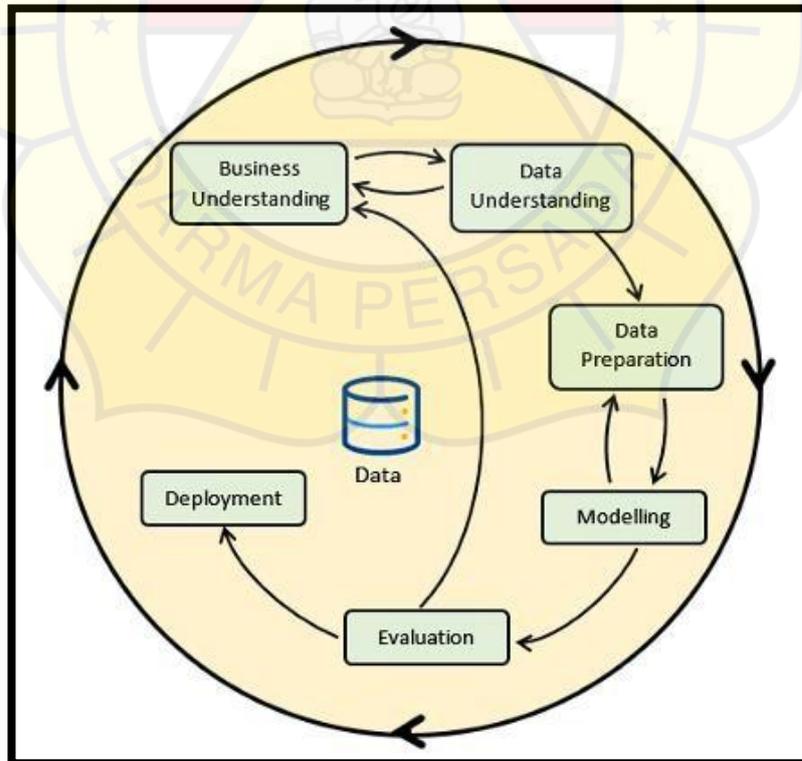
CNN adalah model *deep learning* yang dirancang khusus untuk pengolahan citra dan data berstruktur grid, seperti gambar dan video. *CNN* menggunakan lapisan konvolusi untuk mengekstraksi fitur dari gambar, yang kemudian digunakan

untuk pengenalan objek, klasifikasi gambar, dan deteksi wajah (Budiman, Lestanti, & Yuana, 2023).

2.1.1.6 CRISP-DM

Metodologi yang digunakan dalam penelitian ini adalah *CRISP-DM* (*Cross Industry Standard Process for Data Mining*), yang didirikan pada akhir 1990-an oleh empat perusahaan: *Integral Solutions Ltd.* (penyedia solusi data mining komersial), *NCR* (penyedia database), *DaimlerChrysler* (penyedia mobil produsen), dan *OHRA* (perusahaan asuransi). Ini adalah standarisasi proses untuk menyelesaikan masalah dalam penelitian *data mining*.

CRISP-DM menggunakan 6 tahapan, tahapan-tahapan *CRISP-DM* sebagai berikut :



Gambar 2. 1 Alur Proses CRISP-DM

1. Business Understanding

Tahap pemahaman terhadap substansi dari penelitian yang dilakukan dengan menentukan tujuan dan Batasan dari penelitian ini.

2. Data Understanding

Tahap ini dimulainya proses pengumpulan data, mempelajari data yang didapatkan terkait kualitas data, kapan data didapatkan dan menghilangkan duplikasi data.

3. Data Preparation

Tahap ini data yang telah di review, kemudian dilakukan proses perbaikan struktur data dan datasetnya, dengan menormalisasikan data menggunakan cara :

a. Case Folding

Yaitu menjadikan semua kata menjadi huruf kecil.

b. Cleaning

Merupakan aktifitas meniadakan tanda baca, numeric, link url, username dan simbol.

c. Stopword-Removal

Merupakan aktifitas meniadakan kata-kata yang tidak penting seperti, di, akan, dll.

d. Stemming

Yaitu menormalisasikan kata ke kata dasar.

4. Modeling

Tahapan penentuan Teknik data mining yang akan digunakan pada penelitian ini, pada tahap ini aplikasi data mining yang digunakan serta parameter yang akan digunakan sudah di tentukan sebelumnya untuk mendapatkan hasil yang maksimal.

5. Evaluation

Tahap ini evaluasi dilakukan terhadap model yang telah di tentukan sebelumnya, untuk mendapatkan hasil apakah model tersebut dapat memenuhi tujuan yang ditetapkan pada tahap *Business Understanding*.

6. Deployment

Tahap ini merupakan representasi dari proses yang telah dilakukan sebelumnya, yang bisa berupa laporan atau sebagai format yang bisa digunakan secara berulang pada penelitian selanjutnya.

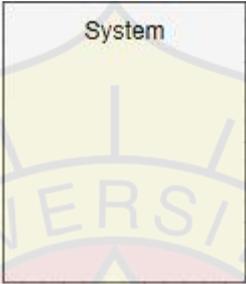
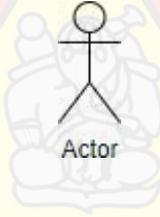
2.1.1.6 Pemodelan Sistem UML

Ini adalah alat diagram standar untuk mendesain. Ini adalah bahasa grafis untuk menentukan, memvisualisasikan, membangun dan mendokumentasikan komponen sistem perangkat lunak. Ini membantu dalam pemahaman yang lebih baik tentang perangkat lunak atau sistem atau produk yang akan dikembangkan di antara pengembang dan pelanggan (Sundaramoorthy, 2022:2).

1. Use Case Diagram

(Sundaramoorthy, 2022) menyatakan bahwa, Ini berfokus pada identifikasi persyaratan fungsional sistem yang sedang dipertimbangkan. Tabel 2.1 menjelaskan rincian komponen-komponen dari *use case* diagram.

Tabel 2. 1 Rincian Komponen *Use case* Diagram

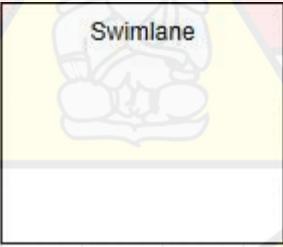
| No | Komponen | Simbol | Tujuan |
|----|-------------------------------|---|--|
| 1 | <i>System Boundary</i> |  | <ul style="list-style-type: none"> • Mempresentasikan Ruang Lingkup Sistem • Merangkum serangkaian fungsi lengkap sistem. |
| 2 | <i>Actor</i> |  | <ul style="list-style-type: none"> • Pengguna yang berinteraksi dengan suatu sistem. • Seorang aktor dalam <i>use case diagram</i> berinteraksi dengan <i>use case</i>. • Pengembang harus mempertimbangkan dampak apa saja yang akan diberikan aktor terhadap fungsi yang ingin mereka modelkan. |
| 3 | <i>Use case</i> |  | <ul style="list-style-type: none"> • <i>Use case</i> adalah representasi visual dari fungsi bisnis yang berbeda dalam suatu sistem. • Memastikan bahwa proses bisnis bersifat diskrit. • Buat daftar fungsi bisnis yang terpisah dalam rumusan |

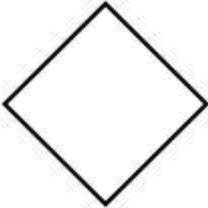
| | | | |
|---|-----------------------|------------------------|---|
| | | | <p>masalah yang diberikan.</p> <ul style="list-style-type: none"> • Mengidentifikasi kasus penggunaan adalah suatu penemuan. |
| 4 | Association | ————— | <ul style="list-style-type: none"> • Merepresentasikan hubungan antara aktor dan <i>use case</i>. |
| 5 | Include | -----<<include>>-----> | <ul style="list-style-type: none"> • Relasi <i>include</i> merepresentasikan situasi dimana suatu <i>use case</i> menyertakan fungsionalitas dari satu <i>use case</i> lainnya. |
| 6 | Extend | -----<<extend>>-----> | <ul style="list-style-type: none"> • <i>Extend relationship</i> antara dua <i>use case</i> mana pun menyiratkan hubungan yang berarti bahwa <i>use case</i> yang diperluas menambahkan perilaku tambahan terhadap fungsionalitas <i>use case</i> dasar yang ada. |
| 7 | Generalization | —————> | <ul style="list-style-type: none"> • <i>Generazation</i> adalah hubungan antara <i>use case</i> induk dan satu atau lebih <i>use case</i> anak. |
| 8 | Dependency | -----> | <ul style="list-style-type: none"> • <i>Dependency</i> mendefinisikan hubungan di mana keberadaan satu <i>use case</i> bergantung pada keberadaan <i>use case</i> lainnya. |

2. Activity Diagram

(Sundaramoorthy, 2022) menyatakan bahwa, *Activity Diagram* berfokus pada aktivitas berurutan dan paralel yang terlibat dalam setiap kebutuhan fungsional sistem. Tabel 2.2 menjelaskan rincian komponen-komponen dari *activity diagram*.

Tabel 2. 2 Rincian Komponen *Activity Diagram*

| No | Komponen | Simbol | Tujuan |
|----|----------------------|---|---|
| 1 | <i>Initial State</i> |  | <ul style="list-style-type: none"> Merepresentasikan keadaan awal sistem sedang dipertimbangkan |
| 2 | <i>Final State</i> |  | <ul style="list-style-type: none"> Merepresentasikan status penghentian sistem yang sedang dipertimbangkan. |
| 3 | <i>Swimlane</i> |  | <ul style="list-style-type: none"> <i>Swimlane</i> terdiri dari dua jenis yaitu <i>Swimlane</i> vertikal dan <i>Swimlane</i> horizontal. Vertical <i>Swimlane</i> – mewakili aktivitas paralel dari skenario tertentu dari sistem yang sedang dipertimbangkan. Horizontal <i>Swimlane</i> – mewakili aktivitas berurutan dari skenario tertentu dari sistem yang sedang dipertimbangkan. |

| | | | |
|---|------------------------|---|--|
| 4 | <i>Action State</i> |  | <ul style="list-style-type: none"> • Status tindakan mewakili operasi atau aktivitas bisnis atau proses. |
| 5 | <i>Decision</i> |  | <ul style="list-style-type: none"> • Sebuah simpul keputusan mempunyai satu masukan atau dua atau lebih keluaran tergantung pada kondisi yang dirancang. • Setiap aliran keluaran memiliki kondisi yang melekat padanya. • Jika suatu kondisi terpenuhi, aliran akan berlanjut seiring dengan keluaran yang sesuai. • Keluaran 'lainnya' dapat didefinisikan sepanjang aliran dapat ditentukan lanjutkan jika tidak ada kondisi lain yang terpenuhi. |
| 6 | <i>Synchronization</i> |  | <ul style="list-style-type: none"> • Mewakili dua atau lebih aktivitas yang terjadi pada waktu atau kecepatan yang sama |
| 7 | <i>Flow Final</i> |  | <ul style="list-style-type: none"> • Mewakili penghentian abnormal suatu jalur dalam activity diagram yang tidak dianggap sebagai bagian dari sistem |

| | | | |
|--|--|--|---------------------------|
| | | | yang sedang dikembangkan. |
|--|--|--|---------------------------|

2.1.1.7 Software dan Tools Terkait

1. Visual Studio Code

Visual Studio Code (juga disebut sebagai *VS Code*) adalah editor kode gratis, sumber terbuka, dan lintas platform yang dikembangkan oleh *Microsoft*. Diberi peringkat sebagai Lingkungan Pengembangan Paling Populer dalam Survei Pengembang *Stack Overflow 2019*, *Visual Studio Code* adalah editor kode kaya fitur yang sangat dapat disesuaikan yang tidak hanya bagus untuk mengedit kode sumber tetapi juga memiliki dukungan bawaan untuk kolaborasi dan lingkungan yang dihosting di *cloud* (Speight, 2021).

2. TensorFlow

TensorFlow merupakan *library open source* yang disediakan untuk pembelajaran mesin. *TensorFlow* berfungsi untuk membangun model pembelajaran mesin dan dapat menyediakan komputansi numerik. *Library TensorFlow* sangat populer dan banyak digunakan dalam melakukan pembelajaran mendalam (*Deep Learning*). Untuk menjalankan *library TensorFlow* ini bisa dilakukan pada tools *Jupyter Notebook* dan *Google Colaboratory*. (Noviyanti et al., 2023).

3. Javascript

Javascript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen *HTML*, sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama untuk web. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa *HTML* dengan

mengijinkan pekekseskusion perintah perintah di sisi *user*, yang artinya di sisi *browser* bukan di sisi server web (Wahyudi, 2022).

2.2 Kajian Penelitian Terdahulu

Tabel di bawah ini menyajikan ringkasan referensi penelitian sebelumnya yang terkait dengan pembuatan *computer vision* serta penggunaan metode *CRISP-DM*.

Tabel 2. 3 *Paper* penelitian terkait

| No. | Penulis | Tahun | Judul | Metode | Publikasi |
|---|---|-------|--|----------------------|---|
| 1. | Afika Rianti, Nuur Wachid Abdul Majid, Ahmad Fauzi | 2023 | <i>CRISP-DM:Metodologi Proyek Data Science</i> | <i>CRISP- DM</i> | Prosiding Seminar Nasional Teknologi Informasi dan Bisnis (SENATIB) 2023 |
| <p>Hasil:</p> <p>Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa metodologi <i>CRISP-DM</i> dapat digunakan dalam proyek bidang data science tepatnya dalam area <i>data mining, artificial intelligence, machine learning, deep learning, big data, data analysis, dan data analytics.</i></p> | | | | | |
| <p>Keterbatasan penelitian:</p> | | | | | |

| | | | | | |
|---|--|------|---|--------------------------------|--|
| | <p>Kemudian untuk tantangan dalam menggunakan metodologi ini antara lain adalah <i>outdated</i>, awalan yang lambat, bukan merupakan kerangka koordinasi kelompok, tidak ada kerangka yang jelas untuk berkomunikasi, serta tahap <i>deployment</i> tidak ada. Secara keseluruhan, semua pertanyaan dalam penelitian ini telah terjawab. Untuk penelitian selanjutnya, sebaiknya menggunakan referensi yang lebih banyak serta dapat membandingkan dengan metodologi lain.</p> | | | | |
| 2. | Syefrida Yulina | 2021 | Penerapan <i>Haar Cascade Classifier</i> dalam Mendeteksi Wajah dan Transformasi <i>Citra Grayscale</i> Menggunakan <i>OpenCV</i> | <i>Haar Cascade Classifier</i> | Jurnal Komputer Terapan Vol. 7, No. 1, Mei 2021, 100 – 109 |
| <p>Hasil:</p> <p>Berdasarkan hasil pengujian pada deteksi wajah menggunakan metode <i>Haar Cascade Classifier</i>, maka didapatkan total akurasi adalah 100%., jika citra inputan memiliki objek wajah dengan posisi frontal, sedangkan akurasi dibawah 50% didapatkan dari citra inputan yang memiliki objek wajah frontal / tidak frontal sama sekali. Hasil ini dipengaruhi oleh posisi wajah yang ada pada citra inputan. Sedangkan untuk kesalahan deteksi, didapatkan hasil sebesar 24%. Pada sistem ini juga telah berhasil mengubah citra inputan berwarna menjadi citra keabuan dengan menggunakan <i>library OpenCV</i>.</p> | | | | | |
| <p>Keterbatasan penelitian:</p> <p>Sistem ini tidak dapat mendeteksi wajah yang memiliki posisi tidak frontal.</p> | | | | | |

| | | | | | |
|--|---|------|---|----------------------|--|
| 3. | Samuel Anaya Zai, Sindy Fitriani Margaret, Yohanna Permata Putri | 2023 | Sistem Pendeteksi Kecepatan Kendaraan dengan Menggunakan Metode <i>Deep Learning</i> | <i>Deep Learning</i> | Jurnal Ilmiah Dan Karya Mahasiswa Vol. 1 No. 6 Desember 2023 |
| <p>Hasil:</p> <p>Penelitian ini bertujuan untuk mengembangkan sistem pendeteksi kecepatan kendaraan menggunakan metode <i>deep learning</i>. Metode ini digunakan untuk memproses dan menganalisis informasi visual dalam gambar atau video dengan tujuan memberikan komputer kemampuan untuk "melihat" dan memahami dunia visual seperti manusia. Penelitian ini menggunakan data berupa video <i>real-time</i> yang diambil dari sebuah website pemantauan CCTV jalanan di kota Bekasi. Metode <i>deep learning</i> yang digunakan dalam penelitian ini menggunakan <i>OpenCV (Open Source Computer Vision)</i> sebagai alat untuk memproses video dalam sistem <i>computer vision</i>.</p> | | | | | |
| <p>Keterbatasan penelitian:</p> <p>Optimasi Sistem dan penyesuaian lebih lanjut disarankan untuk meningkatkan kinerja sistem deteksi kecepatan kendaraan . Hal ini mencakup penyempurnaan model dan parameter, serta pengujian dan peningkatan implementasi. Dengan optimasi yang tepat, sistem dapat mencapai hasil yang lebih baik dalam mendeteksi dan melacak kendaraan serta mengukur kecepatan kendaraan.</p> | | | | | |