



# Sistem Pendeteksi Pejalan Kaki Di Lingkungan Terbatas Berbasis SSD *MobileNet V2* Dengan Menggunakan Gambar 360° Ternormalisasi

*(MobileNet V2 SSD Based Pedestrian Detection System Using Normalized 360° Image)*

N. Nufus<sup>1\*</sup>, D.M. Arifin<sup>2</sup>, A. S. Satyawan<sup>3</sup>, R. A. S. Nugraha<sup>4</sup>, M.I. Asysyakuur<sup>5</sup>, N. N. A. M. Santi<sup>6</sup>, C.H. Parangin<sup>7</sup>, Ema<sup>8</sup>

<sup>1,3,4,5,6,7,8</sup>Teknik Elektro, Universitas Nurtanio

E-mail: nufus.te20@gmail.com, radityasn.te16@student.unnur.ac.id, ikromasy.te16@gmail.com, marlina.te20@student.unnur.ac.id, candra.te16@student.unnur.ac.id, ema@unnur.ac.id

<sup>2</sup>Prodi Teknik Aeronautika Pertahanan, Akademi Angkatan Udara, Yogyakarta  
denden8552@aau.ac.id

<sup>3</sup>Badan Riset dan Inovasi Nasional, Universitas Jambi  
E-mail: arief.suryadi@akane.waseda.jp

**Abstract**— *The growth of vehicular traffic in various areas such as urban and rural areas is getting higher due to the increasing need for transportation equipment. This condition causes many scientists to continuously improve the quality of the mode of transportation so that it is easier, safer, and more practical to use. As a result, the idea of driverless transportation has emerged to meet this need. The form of transportation is private vehicles, and it is expected to be in the form of mass vehicles such as buses or trains. Several aspects must be considered in designing autonomous transportation or autonomous electric vehicles so as not to cause accidents that can endanger the driver and the surrounding environment. Among them is the existence of a pedestrian detection system. This system is critical because, like conventional vehicles, autonomous vehicles must avoid pedestrians, but they work with no driver assistance. In this study, a software system that can detect pedestrians from all directions using a 360° camera was developed to overcome the above. This system also utilizes deep learning technology. The design and realization of this system went through several stages. The stages include installing supporting software on the NVIDIA Jetson AGX Xavier, capturing video data with a 360° camera for composing a dataset of 19,038 images, training the MobileNet V2 SSD with the dataset, and testing the trained model with the real-time and offline testing process. As a result, by testing 548 normalized 360° images offline for daytime conditions, 60.40% of images were perfectly detectable, while for 514 normalized 360° images for evening conditions, 62.25% of images were ideally detected.*

**Keywords**— *Object Detection, SSD, MobileNetV2, Autonomous, Mobility In Society 5.0*

**Abstrak**— *Pertumbuhan arus lalu lintas kendaraan di berbagai wilayah seperti perkotaan dan pedesaan semakin tinggi diakibatkan oleh semakin meningkatnya kebutuhan alat transportasi. Pertumbuhan arus lalu lintas kendaraan di berbagai wilayah seperti perkotaan dan pedesaan semakin tinggi diakibatkan oleh semakin meningkatnya kebutuhan alat transportasi. Kondisi tersebut menyebabkan banyak para ilmuwan berupaya untuk meningkatkan kualitas dari moda transportasi secara terus-menerus sehingga semakin mudah, aman dan praktis untuk digunakan. Akibatnya saat*

\*Penulis Korespondensi (Nafisun Nufus)

E-mail: nufus.te20@student.unnur.ac.id

ini telah muncul ide transportasi tanpa pengemudi untuk memenuhi kebutuhan tersebut. Bentuk alat transportasi tersebut tidak hanya kendaraan pribadi, namun lebih diharapkan berupa kendaraan masal seperti bus atau kereta. Hal ini tentunya akan menjadi bentuk tantangan baru dalam memasuki era *Mobility In Society 5.0* yang diperkirakan akan segera dihadapi juga oleh negara kita. Ada beberapa aspek yang harus diperhatikan dalam mendesain transportasi otonom atau kendaraan listrik otonom agar tidak menimbulkan kecelakaan yang dapat membahayakan pengendara dan lingkungan sekitar. Diantaranya adalah keberadaan sistem pendeteksi para pejalan kaki. Sistem ini sangat penting karena seperti halnya kendaraan konvensional, kendaraan otonom juga harus menghindari pejalan kaki, namun kali ini dengan tanpa bantuan pengemudi. Untuk mengatasi hal tersebut di atas, maka pada penelitian ini sistem software yang memiliki fungsi untuk dapat mendeteksi pejalan kaki dari segala arah menggunakan kamera 360° dikembangkan. Sistem ini juga memanfaatkan teknologinya *deep learning*. Desain dan realisasi sistem ini melalui beberapa tahapan yang dimulai dari instalasi software pendukung pada NVIDIA Jetson AGX Xavier, pengambilan data video dengan kamera 360° untuk membuat dataset sebanyak 19.038 gambar, melatih SSD MobileNet V2 dengan dataset tersebut, hingga proses pengujian secara real-time dan offline. Hasilnya, dengan mengujikan 548 gambar 360° yang ternormalisasi secara offline untuk kondisi siang hari terdapat 60,40% gambar yang dapat terdeteksi sempurna, sedangkan untuk 514 gambar 360° ternormalisasi untuk kondisi sore hari diperoleh 62,25% gambar terdeteksi sempurna. Sedangkan pada pengujian real-time diperoleh *confident level* 90% untuk pendeteksian pejalan kaki pada siang hari, dan 85% pada sore hari.

**Kata Kunci**— *Object Detection, SSD, MobileNetV2, Autonomous, Mobility In Society 5.0*

## I. PENDAHULUAN

Peningkatan arus lalu lintas kendaraan dan pergerakan orang di atas prasarana transportasi pada suatu kota seperti prasarana jalan raya perkotaan maupun pedesaan sangat tergantung pada pesatnya pertumbuhan ekonomi suatu daerah atau wilayah kota karena transportasi sendiri merupakan kebutuhan turunan (*derived demand*) oleh jasa transportasi bertambah dengan meningkatnya kegiatan ekonomi dan penurunan perekonomian.

Kondisi peningkatan tersebut seperti berkesan bahwa keselamatan pejalan kaki di perkotaan dan pedesaan cenderung terabaikan dan kebijakan-kebijakan yang diambil cenderung berpihak kepada pemilik kendaraan, utamanya kendaraan pribadi. Hasil studi dari Institut Studi Transportasi (INSTRAN) mendapatkan bahwa 65% kecelakaan lalu lintas berakibat kematian adalah pejalan kaki, dan kecenderungannya melibatkan kelalaian pengendara kendaraan bermotor termasuk mobil. Kondisi ini menjadi pertimbangan berbagai pihak untuk menghadirkan kendaraan yang dapat dikendalikan tanpa pengemudi atau menggunakan komputer sehingga meminimalisir terjadinya kelalaian yang diakibatkan pengendara [1].

Saat ini diberbagai belahan dunia telah mulai dikembangkan kendaraan listrik tanpa campur tangan pengemudi atau biasa disebut kendaraan listrik otonom (KLO). Kendaraan tersebut diharapkan dapat mengurangi kelalaian manusia dalam mengemudi, disamping itu dapat memberi kemudahan bagi anak-anak hingga dewasa untuk dapat bepergian tanpa dibebani keharusan mengemudi.

Agar KLO dapat terwujud maka sistem pendukung otonomnya harus bekerja dengan baik. Para pengembang kendaraan KLO seperti Waymo dan Tesla [2] telah berlomba-lomba mengembangkan sistem pendeteksian objek, sistem tersebut berbasis LiDAR, kamera maupun radar. Masing-masing memiliki keunggulan tersendiri. Pada penelitian ini sistem pendeteksi pejalan kaki di lingkungan terbatas berbasis SSD MobileNet V2 dengan menggunakan gambar 360° ternormalisasi yang akan dikembangkan berbasis kamera. Hal ini dikarenakan kemampuan kamera semakin berkembang dan ekonomis.

Selain itu, beragam cara telah dikembangkan orang untuk menghasilkan konsep deteksi objek berbasis kamera yang memiliki akurasi tinggi. Jika pada awalnya deteksi objek sangat

bergantung pada fitur-fitur objek dalam domain gambar dua dimensi (2D), maka pada saat ini perkembangannya sudah sangat pesat, yaitu dengan memanfaatkan teknologi *Artificial Intelligence* (AI) khususnya *Deep Learning* (DL) [3]. Konsep DL ini selanjutnya menjadi konsep yang akan dikembangkan pada penelitian ini karena memiliki akurasi yang semakin tinggi.

## II. LANDASAN TEORI

Pendeteksian objek adalah salah satu bagian ilmu dari computer vision dalam memproses gambar yang bertujuan dapat mendeteksi objek yang ada pada suatu gambar. Objek yang dimaksud pada topik ini adalah manusia. Setiap objek memiliki fitur khusus yang membedakan objek pada suatu citra. Ada banyak bermacam teknik untuk mendapatkan fitur pada suatu citra sesuai kebutuhan objek yang akan dideteksi. Pada topik ini digunakan metode pengujian *white box* yaitu suatu metode yang menggunakan *software* yang sudah ada dilatihkan pada objek yang diinginkan dalam hal ini menggunakan teknologi *deep learning*.

### A. Artificial Intelligence

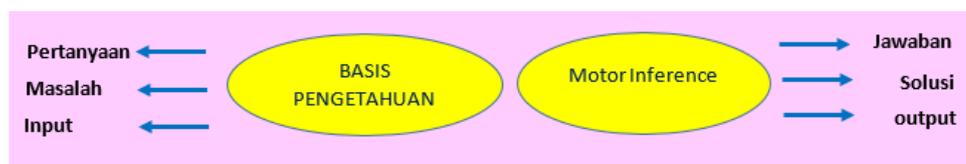
*Artificial Intelligence* (AI) merupakan bagian dari ilmu komputer yang mempelajari teknik agar mesin (komputer) dapat meniru pekerjaan yang dilakukan manusia dengan performa yang mampu menyainginya. AI menurut John McCarthy (1956) yang dikutip dari jurnal penelitian (Pannu, 2015) [4] mengatakan bahwa AI bertujuan untuk mengetahui atau memodelkan proses berpikir manusia dan mendesain mesin sehingga bisa menirukan perilaku manusia. Dalam pembuatan aplikasi kecerdasan buatan terdapat dua hal yang menjadi bagian utama yang dibutuhkan yaitu :

1. *Knowledge Base* (Basis Pengetahuan)

Bagian ini berisi tentang fakta-fakta, teori, pemikiran dan hubungan antara satu dengan yang lainnya.

2. *Inference Engine* (Motor Inferensi)

Kemampuan menarik kesimpulan berdasarkan pengalaman. Setiap paragraf minimal harus terdiri dari dua kalimat. Gunakan kalimat yang padat dan jelas. Perhatikan selalu tentang jarak dan margin. Perhatikan lanjutan contoh paragraf berikut. Penerapan konsep kecerdasan buatan pada komputer dapat dilihat pada Gambar 1.



Gambar 1 Penerapan Konsep Kecerdasan Buatan

AI merupakan salah satu disiplin ilmu yang luas, beberapa lingkup utama AI antara lain adalah Sistem Pakar (*Expert System*), Pengolahan Bahasa Alami (*Natural Language Processing/NLP*), Pengenalan Ucapan (*Speech Recognition*), *Computer Vision*, *Intelligent Computer-Aided Instruction*, dan lainnya. Sistem pakar adalah usaha untuk menirukan seorang pakar. Tujuan dari sistem pakar yaitu untuk mentransfer kepakaran dari seorang pakar ke komputer, kemudian ke orang lain (orang yang bukan pakar).

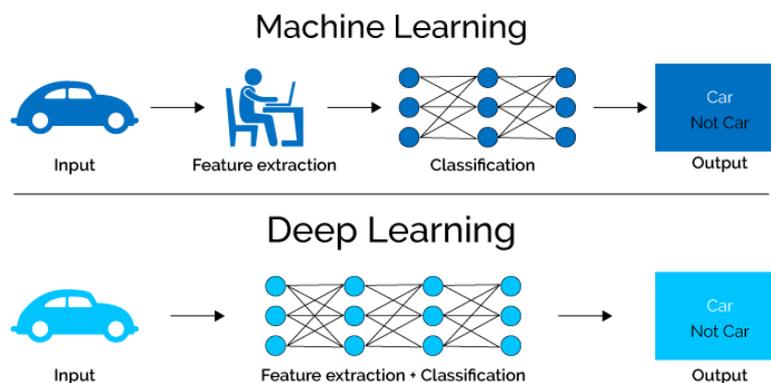
Pengolahan bahasa alami yaitu dimana pengguna bisa melakukan komunikasi dengan bahasa sehari-hari. Pengenalan ucapan yaitu dimana manusia dapat melakukan komunikasi dengan komputer menggunakan suara. *Computer vision* yaitu dalam hal menginterpretasikan

objek atau gambar yang tampak melalui komputer. *Intelligent Computer-Aided Instruction* yaitu bagaimana komputer dapat berperan sebagai tutor yang dapat mengajar atau melatih.

### B. Deep Learning

*Deep Learning* atau sering dikenal dengan istilah Pembelajaran Struktural Mendalam (*Deep Structured Learning*) atau Pembelajaran Hierarki (*Hierarchical learning*) adalah salah satu cabang dari ilmu pembelajaran mesin (*Machine Learning*) yang terdiri algoritma pemodelan abstraksi tingkat tinggi pada data menggunakan sekumpulan fungsi transformasi non-linear yang ditata berlapis-lapis dan mendalam. Teknik dan algoritma dalam *deep learning* dapat digunakan baik untuk kebutuhan pembelajaran terarah (*supervised learning*), pembelajaran tak terarah (*unsupervised learning*) dan semi-terarah (*semi-supervised learning*) dalam berbagai aplikasi seperti pengenalan citra, pengenalan suara, klasifikasi teks, dan sebagainya [5]. *Deep learning* disebut sebagai *Deep* (dalam) karena struktur dan jumlah jaringan saraf pada algoritmanya sangat banyak bisa mencapai hingga ratusan lapisan.

*Deep learning* merupakan salah satu jenis algoritma jaringan saraf tiruan yang menggunakan data yang memberikan informasi sebagai masukan dan memprosesnya menggunakan sejumlah lapisan tersembunyi (*hidden layer*) transformasi *non linier* dari data masukan untuk menghitung nilai luaran. Algoritma pada *deep learning* memiliki fitur yang unik yaitu sebuah fitur yang mampu mengekstraksi secara otomatis. Hal ini berarti algoritma yang dimilikinya secara otomatis dapat menangkap fitur yang relevan sebagai keperluan dalam pemecahan suatu masalah. Algoritma semacam ini sangat penting dalam sebuah kecerdasan buatan karena mampu mengurangi beban pemrograman dalam memilih fitur yang eksplisit atau dapat tersampaikan secara langsung. Algoritma ini dapat digunakan untuk memecahkan permasalahan yang perlu pengawasan (*supervised*), tanpa pengawasan (*unsupervised*), dan semi terawasi (*semi supervised*). Pemodelan jaringan *deep learning* ditunjukkan pada Gambar 2.



Gambar 2. *Machine Learning dan Deep Learning* [5]

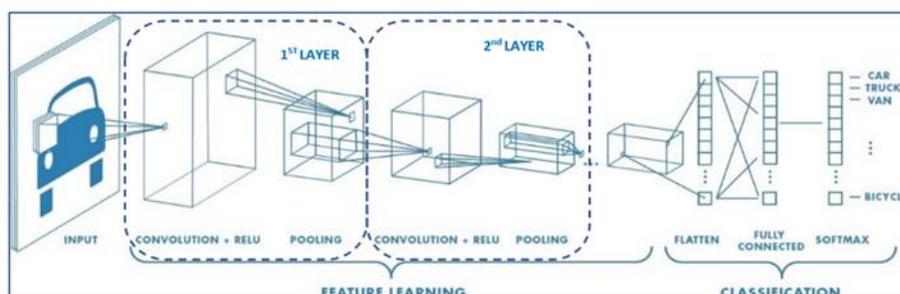
### C. Convolutional Neural Network

*Convolutional Neural Network* adalah salah satu *metode Deep Learning* (DL) yang dapat digunakan untuk mendeteksi dan mengenali sebuah objek pada sebuah citra digital (gambar). *Stride* dan *padding* terdapat di dalam *convolutional neural network*. *Stride* adalah jumlah *piksel* yang digunakan untuk menggeser *filter* secara horizontal atau vertikal. Dimana, dalam hal ini *stride* memindahkan filter satu piksel pada setiap langkah untuk menghitung *output* konvolusi berikutnya. Sedangkan *padding* adalah modifikasi yang umum digunakan yang memungkinkan ukuran *input* disesuaikan dengan kebutuhan [6].

*Convolutional Neural Network* (CNN) merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi dalam bentuk citra (gambar). CNN ini termasuk ke dalam jenis *deep neural network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada dasarnya klasifikasi citra dapat

digunakan dengan MLP, akan tetapi dengan metode MLP kurang sesuai untuk digunakan karena tidak menyimpan informasi spesial dari data citra dan menganggap setiap piksel adalah fitur yang independent sehingga menghasilkan hasil yang kurang baik [7].

Pada dasarnya CNN merupakan sebuah arsitektur ANN yang memiliki karakteristik khusus, dan dapat dilatih serta memiliki beberapa tahapan yang terdiri dari masukan (*input*) dan keluaran (*output*). Masukan dan keluaran pada setiap tahapan terdiri dari beberapa array yang berfungsi untuk menghasilkan *feature map*. Setiap tahap terdiri dari tiga proses utama yaitu konvolusi, fungsi aktivasi dan *pooling*. Konsep dari CNN dapat dilihat pada Gambar 3.



Gambar 3. Struktur Dasar *Convolutional Neural Network* [7]

Pada Gambar 3 dapat dilihat bahwa tahap pertama dari arsitektur CNN merupakan tahap konvolusi. Pada tahap ini masukan gambar dicari fitur-fiturnya dengan menggunakan sebuah kernel (*filter*) dengan ukuran tertentu. Ukuran dan jumlah kernel yang digunakan bergantung pada jumlah fitur yang ingin dihasilkan. Setelah tahap tersebut dilanjutkan dengan tahap yang melibatkan fungsi aktivasi. Fungsi aktivasi dapat berupa proses linierisasi seperti fungsi aktivasi ReLU (*Rectifier Linear Unit*). Setelah proses fungsi aktivasi kemudian diikuti dengan proses *pooling*. Pada proses *pooling* jumlah *feature map* akan berkurang secara signifikan sehingga jumlah data tidak membengkak dan mengakibatkan waktu proses meningkat. Keseluruhan proses tersebut dapat diulang secara berurutan berkali-kali sampai didapatkan *feature map* yang cukup untuk dilanjutkan ke proses *fully connected neural network*, sehingga didapatkan *output class*.

#### D. Deteksi Objek

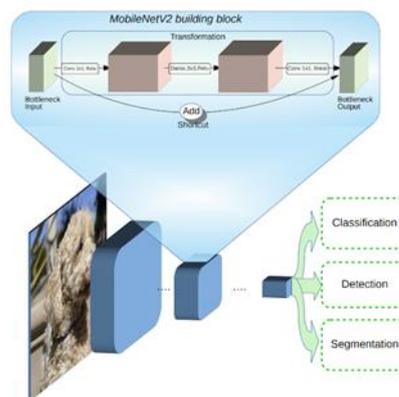
Deteksi objek (*Object Decton*) adalah teknik visi komputer untuk menemukan contoh objek dalam gambar atau video. Algoritma deteksi objek biasanya memanfaatkan *machine learning* dan *deep learning* untuk menampilkan hasil yang bermakna. Ketika manusia melihat gambar atau video, manusia dapat mengenali dan menemukan objek dalam beberapa saat, berbeda dengan komputer yang masih memerlukan komputasi yang kompleks. Tujuan deteksi objek adalah untuk mereplikasi kecerdasan yang dimiliki manusia dalam melihat benda menggunakan komputer. Cara kerja deteksi objek adalah dengan menempatkan keberadaan objek dalam gambar dan menggambarkan kotak pembatas disekitar objek tersebut [8].

Proses deteksi tersebut dapat dilakukan dengan berbagai macam metode yang umumnya melakukan pembacaan fitur-fitur dari seluruh objek pada citra *input*. Fitur dari objek pada citra input tersebut akan dibandingkan dengan fitur dari model yang digunakan atau *template*. Hasil perbandingan tersebut dapat digunakan untuk menentukan apakah suatu objek terdeteksi sebagai *template* yang dimaksud atau tidak. Sistem deteksi objek perlu melatih dan menguji SSD mobileNet V2 dengan dataset yang merupakan data yang diambil dengan kamera 360° ternormalisasi sebanyak 19.038 dengan *bounding box* dan diberi label untuk kelas pada setiap objek untuk proses pengenalan. Demi mencapai tujuan ini, ada banyak dataset untuk menghasilkan model *Deep Learning* salah satunya seperti Pascal-VOC.

### E. MobileNet V2

*MobileNet* adalah model kecil latensi rendah daya rendah yang diukur untuk memenuhi batasan sumber daya dari berbagai kasus penggunaan. Menurut makalah penelitian, MobileNet V2 meningkatkan kinerja model seluler yang canggih pada banyak tugas dan tolak ukur serta pada seluruh spektrum ukuran model yang berbeda. MobileNet V2 merupakan ekstraktor fitur yang sangat efektif untuk deteksi dan segmentasi objek. Misalnya untuk deteksi, saat dipasangkan dengan *Single Shot Detector Lite*, MobileNetV2 sekitar 35 persen lebih cepat dengan akurasi yang sama dari MobileNet V1.

Kemacetan dari MobileNetV2 mengkodekan *input* dan *output* perantara sementara lapisan dalam merangkul kemampuan model untuk mengubah dari konsep tingkat yang lebih rendah seperti piksel ke deskriptor tingkat yang lebih tinggi seperti kategori gambar. Dengan koneksi residual tradisional, pintasan memungkinkan pelatihan yang lebih cepat dan akurasi yang lebih baik. MobileNet V2 masih menggunakan *depthwise* dan *pointwise convolution*. MobileNet V2 menambahkan dua fitur baru yaitu: 1) *linear bottleneck*, dan 2) *shortcut connections* antar *bottlenecks* struktur dasar dari arsitektur ini ditunjukkan pada Gambar 4.



Gambar 4. Arsitektur Mobilenet V2 Kotak Biru Menunjukkan Blok Pembentukan Konvolusi *Linear* [9] *Bottleneck*

Pada bagian *bottleneck* terdapat *input* dan *output* antara model sedangkan lapisan atau layer bagian dalam meng-enkapsulasi kemampuan model untuk mengubah input dari konsep tingkat yang lebih rendah (piksel) ke deskriptor tingkat yang lebih tinggi.

Pada akhirnya, seperti halnya koneksi residua pada CNN tradisional, *shortcut* antar *bottlenecks* memungkinkan *training* atau pelatihan yang lebih cepat dan akurasi yang lebih baik. Arsitektur model pada MobileNet V2 yaitu terdapat blok bangunan dasar yang merupakan konvolusi yang dapat dipisahkan kedalamannya dengan residu. Arsitektur MobileNetV2 berisi lapisan konvolusi penuh awal dengan 32 filter, diikuti oleh 19 lapisan *bottleneck residual*. Para peneliti telah menyesuaikan arsitektur ke titik kinerja yang berbeda, dengan menggunakan resolusi gambar input dan pengganda lebar sebagai hyperparameter yang dapat disetel, yang dapat disesuaikan tergantung pada akurasi yang diinginkan atau pertukaran kinerja. Jaringan utama (pengganda lebar  $1,224 \times 224$ ), memiliki biaya komputasi 300 juta perkalian-tambahan dan menggunakan 3,4 juta parameter. Biaya komputasi jaringan berkisar dari 7 perkalian-tambahan hingga 585 juta, sedangkan ukuran model bervariasi antara parameter 1,7 juta dan 6,9 juta.

Perbedaan Model MobileNetV1 dan MobileNetV2 yaitu algoritma serta pada struktur dari masing-masing model, untuk MobileNetV1 hanya terdapat satu layer dan untuk MobileNetV2 terdapat dua layer yang membuat waktu proses MobileNetV2 lebih cepat dibandingkan dengan MobileNetV1, MobileNetV2 menggunakan operasi 2 kali lebih sedikit, memiliki akurasi lebih tinggi, membutuhkan parameter 30 persen lebih sedikit dan sekitar 30-40 persen lebih cepat [9].

*F. Insta 360° One R*

Pada penelitian ini kamera yang digunakan adalah kamera Insta 360° *one R* yang sering disebut dengan kamera 360° yang berfungsi untuk mendeteksi objek berupa pejalan kaki dengan keadaan secara *real-time* dan *offline*. Pada dasarnya teknologi kamera 360° adalah kamera yang memungkinkan untuk mengambil gambar dari semua sudut (360°) sehingga mampu menghasilkan video yang dapat dilihat dari berbagai sisi-depan, belakang, samping, atas dan bawah [10]. Pada Gambar 5 hasil dari pengambilan data menggunakan kamera 360° yang umumnya kamera 360° mempunyai dua lensa yaitu lensa depan 180° (gambar bagian atas) dan lensa belakang 180° (gambar bagian bawah) yang ternormalisasi untuk mengambil dan menggabungkan gambar secara bersamaan. Hasilnya kamera mampu menghasilkan gambar yang tajam dan akurat pada kondisi cerah maupun redup.



Gambar 5. Hasil Kamera 360°

III. MODEL YANG DIUSULKAN

*A. Tahapan Penelitian*

Metodelogi penelitian ini dilakukan secara sistematis dari mulai tahapan studi literatur hingga pembuatan laporan penelitian Secara lebih detil tahapan, metode dan hasil pelaksanaan ini dapat dilihat pada Tabel 1 berikut.

TABEL I  
TAHAPAN PENELITIAN

No	Tahapan	Metode	Hasil
1.	Studi literatur terkait sistem pendeteksi pejalan kaki berbasis SSD MobileNet V2, dan teknologi gambar 360° .	a. Survei internet. b. Studi buku.	Pemahaman: a. Mengenai teknologi mengenai teknologi deep learning berbasis SSD MobileNet V2 dan teknologi gambar 360°. b. Pemrograman dengan bahasa <i>Python</i> pada PC dan NVIDIA Jetson

TABLE 1 LANJUTAN

No	Tahapan	Metode	Hasil
1.	Studi literatur terkait sistem pendeteksi pejalan kaki berbasis SSD MobileNet V2, dan teknologi gambar 360 <sup>0</sup> .	c. Survei internet. d. Studi buku.	Pemahaman: c. Mengenai teknologi mengenai teknologi deep learning berbasis SSD MobileNet V2 dan teknologi gambar 360 <sup>0</sup> . d. Pemrograman dengan bahasa <i>Python</i> pada PC dan NVIDIA Jetson
2.	Desain dan instalasi <i>software</i> sistem pendeteksi pejalan kaki berbasis SSD MobileNet V2 dengan menggunakan gambar 360 <sup>0</sup> ternormalisasi.	a. Menggunakan <i>software</i> pemrograman <i>Python</i> . b. Mengambil data gambar pejalan kaki dengan kamera 360 <sup>0</sup> c. Memberi label pada gambar 360 <sup>0</sup> menggunakan CVAT.	a. Desain dan modifikasi <i>software</i> aplikasi pendeteksian pejalan kaki berbasis SSD MobileNet V2. b. Dataset gambar 360 <sup>0</sup> ternormalisasi berisi informasi pejalan kaki.
3.	Realisasi <i>software</i> sistem pendeteksi pejalan kaki berbasis SSD MobileNet V2 dengan menggunakan gambar 360 <sup>0</sup> ternormalisasi pada Jetson AGX xavier	Melatih <i>software</i> sistem pendeteksi pejalan kaki berbasis SSD MobileNet V2 dengan dataset gambar 360 <sup>0</sup> ternormalisasi berisi informasi pejalan kaki.	<i>Software</i> aplikasi sistem pendeteksi pejalan kaki berbasis SSD MobileNet V2 dengan menggunakan gambar 360 <sup>0</sup> ternormalisasi.
4.	Pengukuran kinerja <i>software</i> teknologi deteksi objek berbasis <i>deep learning</i>	Pengukuran skalalaboratorium dan lapangan.	Kinerja <i>software</i> aplikasi sistem pendeteksi pejalan kaki berbasis SSD MobileNet V2 dengan menggunakan gambar 360 <sup>0</sup> ternormalisasi pada jetson AGX xavier.
5	Melakukan analisis kinerja <i>software</i> deteksi pejalan kaki berbasis SSD MobileNet V2 pada Jetson AGX xavier.	Analisis kuantitatif dan analisis kualitatif	Data dan kesimpulan penelitian

#### IV. IMPLEMENTASI MODEL DAN PEMBAHASAN

Pada bab ini akan dijelaskan tahapan pengujian sistem deteksi pejalan kaki berbasis metode MobileNetV2. Berdasarkan dua kondisi waktu yang mewakili kecerahan gambar, yaitu gambar yang diambil pada waktu siang hari dan sore hari.

##### A. Kondisi Siang Hari Dan Sore Hari

Pengujian pertama dari penelitian ini adalah pengujian pejalan kaki pada siang hari dan sore hari. Hasil pengujian ini dibagi menjadi 3 katagori yaitu deteksi sempurna, deteksi tidak sempurna dan deteksi gagal.

##### 1. Kategori Deteksi Sempurna

Pada Gambar 6.a untuk kondisi siang hari dan 6.b untuk kondisi sore hari diperlihatkan hasil deteksi pejalan kaki dengan kategori sempurna. Objek pejalan kaki yang dideteksi ditunjukkan oleh *bounding box* (kotak) yang berwarna hijau, disertai dengan persentase tingkat keakuratan (*confident level*) dari pejalan kaki yang dideteksi.



Gambar 6.a. Deteksi Sempurna Siang Hari



Gambar 6.b. Deteksi Sempurna Sore Hari

Gambar 6.a dan 6.b dikategorikan deteksi sempurna karena setiap pejalan kaki yang terdapat pada gambar tersebut sudah dipindai *bounding box* dan disertai dengan adanya *confident level* yang menunjukkan tingkat keakuratan deteksi masing-masing objek pejalan kaki.

### 2. Deteksi Tidak Sempurna

Pada Gambar 7.a kondisi siang hari dan 7.b kondisi sore hari diperlihatkan hasil deteksi pejalan kaki dengan kategori tidak sempurna. Objek pejalan kaki yang dideteksi ditunjukkan oleh *bounding box* yang berwarna hijau. disertai dengan persentase tingkat keakuratan hasil deteksi.



Gambar 7.a. Deteksi Tidak Sempurna Siang Hari



Gambar 7.b. Deteksi Tidak Sempurna Sore Hari

Gambar 7.a dikategorikan deteksi tidak sempurna karena ada satu pejalan kaki yang terdapat pada gambar tersebut tidak memperoleh *bounding box*. Gambar 7.b dikategorikan deteksi tidak sempurna karena adanya motor yang merupakan objek selain pejalan kaki yang memperoleh *bounding box* serta adanya *confident level* dengan nilai 56.6%.

### 3. Deteksi Gagal

Pada Gambar 8.a kondisi siang hari dan 8.b kondisi sore hari diperlihatkan hasil deteksi pejalan kaki dengan kategori gagal karena tidak ditemukan *bounding box* pada objek yang seharusnya dideteksi sebagai pejalan kaki.



Gambar 8.a Deteksi Gagal Siang Hari



Gambar 8.b Deteksi Gagal Sore Hari

Pada Gambar 8.a diperlihatkan kategori pendeteksian gagal karena semua pejalan kaki yang terdapat pada gambar tersebut tidak dapat dideteksi oleh *software* yang ditunjukkan oleh tidak adanya *bounding box* serta *confident level* pada setiap gambar tersebut. Untuk persentase keakuratan dari pengujian dengan kondisi siang hari.

Sedangkan Gambar 8.b dikategorikan deteksi gagal karena semua pejalan kaki yang terdapat pada gambar tersebut tidak dapat dideteksi oleh *software* yang ditunjukkan oleh tidak adanya *bounding box*. Contoh lainnya adalah *bounding box* yang muncul pada objek yang bukan pejalan kaki yaitu motor.

TABEL II  
PERSENTASE KEAKURATAN PENGUJIAN SIANG HARI

Kategori	Jumlah	Persentase
Deteksi Sempurna	331	$\frac{331}{548} \times 100\% = 60,40\%$
Deteksi Tidak Sempurna	134	$\frac{134}{548} \times 100\% = 24,45\%$
Deteksi Gagal	83	$\frac{83}{548} \times 100\% = 15,14\%$

TABEL III  
PERSENTASE KEAKURATAN PENGUJIAN SORE HARI

Kategori	Jumlah	Persentase
Deteksi Sempurna	320	$\frac{320}{514} \times 100\% = 62,25\%$
Deteksi Tidak Sempurna	120	$\frac{120}{514} \times 100\% = 23,34\%$
Deteksi Gagal	74	$\frac{74}{514} \times 100\% = 14,39\%$

Untuk persentase keakuratan dari pengujian yang dilakukan kondisi siang hari dengan jumlah gambar sebanyak 548 diperlihatkan pada Tabel 2, sedangkan untuk kondisi sore hari dengan jumlah gambar sebanyak 514 dapat dilihat pada Tabel 3.

### B. Waktu Pemrosesan

Pengujian waktu proses dimaksudkan untuk mengetahui berapa lama waktu yang diperlukan bagi model *software* pendeteksian pejalan kaki dalam sebuah gambar. Pada penelitian ini dilakukan pengujian terhadap waktu proses terhadap 1 gambar yang mewakili kondisi siang hari, dan 1 gambar yang mewakili kondisi sore hari. Waktu proses dihitung baik menggunakan *Central Processing Unit* (CPU) maupun *Graphical Processing Unit* (GPU). Data waktu proses pada kondisi siang hari dapat dilihat pada Tabel 4, dan pada Gambar 9 diperlihatkan gambar yang diuji untuk kondisi siang hari.

TABEL IV  
DATA WAKTU PROSES PADA KONDISI SIANG HARI

No	Nomor Gambar	Waktu Proses	
		CPU ( <i>millisecond</i> )	GPU ( <i>millisecond</i> )
1	Gambar 9	34,04	34,03



Gambar 9. Gambar Yang Diuji pada Kondisi Siang Hari

Waktu proses yang diambil dari 1 sampel pada kondisi siang hari adalah 34,04 ms untuk dapat bekerja pada CPU, dan 34,03 ms untuk dapat bekerja pada GPU. Waktu proses pada gambar pertama lebih besar dari pada gambar berikutnya dikarenakan pencarian nilai bobot awal dari setiap node pada jaringan.

Sementara itu, data kecepatan waktu proses pada kondisi sore hari dapat dilihat pada Tabel 5, dan pada Gambar 10 diperlihatkan gambar yang diuji pada sore hari.

TABEL V  
DATA WAKTU PROSES PADA KONDISI SORE HARI

NO	NOMOR GAMBAR	WAKTU PROSES	
		CPU ( <i>millisecond</i> )	GPU ( <i>millisecond</i> )
1	Gambar 10	37,47	37,52



Gambar 10. Gambar Yang Diuji pada Kondisi Sore Hari

Waktu proses yang diambil dari 1 sampel pada kondisi sore hari adalah 37,47 ms untuk dapat bekerja pada CPU, dan 37,52 ms untuk dapat bekerja pada GPU. Selain itu, waktu proses pada kondisi gambar sore hari cenderung lebih lama dari kondisi gambar siang hari dikarenakan intensitas cahaya yang lebih rendah mengharuskan pencarian nilai bobot yang lebih lama.

## V. KESIMPULAN

Pada penelitian tugas akhir ini, telah dilakukan rancang bangun *software* pendeteksian pejalan kaki berbasis *deep learning*, yang nantinya akan dimanfaatkan untuk aplikasi kendaraan listrik otonom. Adapun hasilnya dapat disimpulkan sebagai berikut:

*Software* pendeteksi pejalan kaki berbasis *deep learning* dengan *network SSD MobilenetV2* dapat berfungsi untuk mendeteksi pejalan kaki baik *offline* maupun secara *real-time*. Model *software* ini telah diujikan pada mini komputer jenis NVIDIA jetson xavier.

Pengukuran kinerja dari sistem pendeteksian pejalan kaki berbasis *deep learning* ini dilakukan setelah model *software* dilatih dengan menggunakan 19.038 dataset gambar 360° ternormalisasi yang diambil dengan menggunakan kamera 360°. Kinerja sistem selanjutnya diperoleh dengan mengujikan 548 gambar 360° ternormalisasi secara *offline* untuk kondisi gambar siang hari, dan 514 gambar 360° ternormalisasi secara *offline* untuk kondisi sore hari. Pada kondisi gambar siang hari, 60,40% gambar dapat terdeteksi sempurna dengan *confident level* maksimum 90%. Sedangkan untuk kondisi gambar sore, 62,25% gambar dapat terdeteksi sempurna dengan *confident level* maksimum 85%. Pada penelitian ini dilakukan pengujian terhadap waktu proses terhadap satu gambar yang mewakili kondisi siang hari, dan satu gambar yang mewakili kondisi sore hari. Pada penelitian ini dilakukan pengujian terhadap waktu proses terhadap satu gambar yang mewakili kondisi siang hari, dan satu gambar yang mewakili kondisi sore hari. Waktu proses dihitung baik menggunakan CPU maupun GPU. Waktu proses pendeteksian pejalan kaki pada kondisi siang hari pada sebuah gambar adalah 34.04 ms saat *software* bekerja pada CPU, dan 34.03 ms pada GPU. Sedangkan untuk pengujian gambar pada sore hari diperlukan waktu proses pendeteksian pejalan kaki pada sebuah gambar 37.47 ms saat *software* bekerja pada CPU, dan 37.52 ms pada GPU.

### UCAPAN TERIMA KASIH

Ucapan terima kasih kami sampaikan kepada pihak-pihak yang telah berkontribusi dalam penelitian ini hingga makalah ini dapat dipublikasikan pada Seminar Nasional Sains Teknologi dan Inovasi Indonesia 2021, diantaranya kepada:

- a. LPDP (Lembaga Pengelola Dana Pendidikan) sebagai sponsor penelitian.
- b. BRIN (Badan Riset dan Inovasi Nasional) sebagai penyedia prasarana penelitian.
- c. Universitas Nurtanio sebagai penyedia prasarana penelitian.
- d. AAU (Akademi Angkatan Udara) sebagai perguruan tinggi kerjasama dengan Yasau untuk kolaborator penulisan naskah dan penyiapan konferensi serta prosiding.
- e. Serta rekan rekan – rekan peneliti lainnya.

### REFERENSI

- [1] Waymo, <https://id.wikipedia.org/wiki/Waymo>, 21 June 2021.
- [2] Tesla, [https://id.wikipedia.org/wiki/Tesla,\\_Inc.](https://id.wikipedia.org/wiki/Tesla,_Inc.), 21 June 2021.
- [3] Artificial Intelligence, [https://id.wikipedia.org/wiki/Kecerdasan\\_buatan](https://id.wikipedia.org/wiki/Kecerdasan_buatan), 21 June 2021.
- [4] J. McCarthy, "What Is Artificial Intelligence Anyway," Computer Science Department, pp. 1–15, 2007.
- [5] XenonStack. 2017. "Log Analytics With Deep Learning And Machine Learning". Medium, 13 Mei 2017. (<https://medium.com/@xenonstack/log-analytics-with-deep-learning-and-machine-learning-20a1891ff70e>) diakses pada 25 Agustus 2021.
- [6] Sahasri, M., & Gireesh, C. (2017). Object Motion Detection and Tracking for Video Surveillance. International Journal of Engineering Trends and Technology, (April), 161–164.
- [7] Hubel & Wiesel, T, 1968.
- [8] Sahasri, M., & Gireesh, C. (2017). Object Motion Detection and Tracking for Video Surveillance. International Journal of Engineering Trends and Technology, (April), 161–164.
- [9] MobileNet V2, <https://medium.com/nodeflux/mobilenet-deteksi-objek-pada-platform-mobile-bbbf3806e4b3>.
- [10] A.S. Satyawan, J. Hara, H. Watanabe, "Automatic Self-Improvement Scheme in Optical Flow-Based Motion Estimation for Sequential Fisheye Images," ITE Transactions on Media Technology and Applications, Vol. 7, No. 1, pp. 20-35, Jan. 2019.