

**SKRIPSI**  
**SISTEM DETEKSI HAMBATAN PADA *AUTONOMOUS DRIVING***  
**MENGGUNAKAN METODE *SINGLE SHOT MULTIBOX DETECTOR***  
**(SSD)**

**Disusun dan diajukan oleh:**

**TASLINDA**

**D121171010**



**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2022**

**LEMBAR PENGESAHAN SKRIPSI**

**SISTEM DETEKSI HAMBATAN PADA AUTONOMOUS DRIVING  
MENGUNAKAN METODE SINGLE SHOT MULTIBOX DETECTOR  
(SSD)**

**Disusun dan diajukan oleh**

**TASLINDA  
D121171010**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin pada tanggal 12 Oktober 2022 dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,

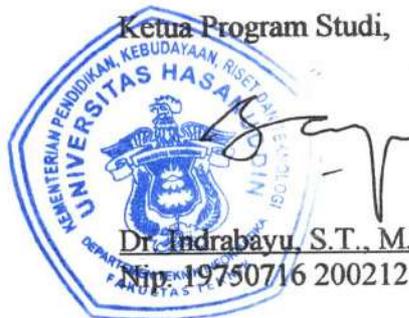
Pembimbing Utama,

Dr. Indrabayu, ST., MT., M.Bus.Sys.  
Nip. 197507162002121004

Pembimbing Pendamping,

Anugrayani Bustamin, S.T., M.T.  
Nip. 199012012018074001

Ketua Program Studi,



Dr. Indrabayu, S.T., M.T., M.Bus.Sys.  
Nip. 19750716 200212 1 004

## PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Taslinda  
NIM : D121171010  
Departemen : Teknik Informatika  
Jenjang : S1

Menyatakan dengan ini karya tulisan saya berjudul:

SISTEM DETEKSI HAMBATAN PADA *AUTONOMOUS DRIVING*  
MENGUNAKAN *METODE SINGLE SHOT MULTIBOX DETECTOR (SSD)*

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilalihan tulisan orang lain bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 12 Oktober 2022

Yang menyatakan,



TASLINDA

## KATA PENGANTAR

Puji syukur penulis panjatkan atas kehadiran Allah SWT karena berkat Rahmat dan Karunia-Nya sehingga Tugas Akhir yang berjudul **“SISTEM DETEKSI HAMBATAN PADA *AUTONOMOUS DRIVING* MENGGUNAKAN METODE *SINGLE SHOT MULTIBOX DETECTOR (SSD)*”** ini dapat terselesaikan dengan baik sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari bahwa banyak kendala yang dihadapi dalam menyelesaikan skripsi ini, namun berkat dorongan, dukungan, bimbingan serta motivasi yang diberikan oleh berbagai pihak, sehingga penyusunan skripsi ini dapat terselesaikan dengan baik. Ucapan terima kasih serta penghargaan yang setinggi-tingginya penulis sampaikan kepada:

1. Orang tua penulis, Bapak Aripin dan Ibu Jumase, yang selalu memberikan dukungan, doa, semangat dan kasih sayang serta selalu sabar dalam mendidik penulis sejak kecil.
2. Bapak Dr. Indrabayu S.T., M.T., M. Bus.Sys., selaku pembimbing I dan Ibu Anugrayani Bustamin, S.T., M.T., selaku pembimbing II yang selalu menyediakan waktu, tenaga, pikiran dan memberikan bimbingan dalam penyusunan skripsi.
3. Saudara penulis, Lisa dan Tirpa yang telah banyak memberikan doa, dukungan serta *mood booster* penulis selama ini, serta kak Ani dan kak Ilham yang selalu

memberikan nasehat dan masukan selama penyusunan tugas akhir dan membantu penulis di masa-masa sulit.

4. Keluarga besar Tamping dan Umareng *Family* yang telah banyak memberikan dukungan, motivasi, dan doa kepada penulis selama menempuh pendidikan sejak kecil.
5. Teman-teman tim *autonomous*, Muh. Fadhil, Nublan Azqalani, dan Khairul Hidayat yang telah memberikan begitu banyak bantuan selama penelitian, pengambilan data, diskusi progres serta dorongan selama ini. Serta tim *tuberculosis* Alfarabi dan Irzam, yang juga banyak membantu, memberikan masukan dan dorongan kepada penulis selama pengerjaan tugas akhir.
6. Teman-teman dan kakak-kakak Laboratorium AIMP *Research Group* FT-UH yang telah memberikan bantuan selama penelitian dan diskusi mengenai penyusunan tugas akhir.
7. Teman-teman RECOGN17ER atas dukungan dan semangat yang telah diberikan kepada penulis selama awal perkuliahan sampai penyelesaian tugas akhir.
8. Fitriani Nasir, Herlina, Devy, Rieka, Irma, Jumraini dan Wahyudi yang telah banyak memberikan dukungan dan bantuan, serta memberikan semangat kepada penulis di masa-masa sulit.
9. Segenap Dosen dan Staff Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah banyak membantu penulis selama masa perkuliahan.

10. Serta berbagai pihak atas segala dukungan dan bantuannya yang tidak dapat penulis tuliskan satu persatu.

Akhir kata, penulis berharap semoga Allah SWT membalas segala kebaikan dari semua pihak yang telah membantu. Semoga Tugas Akhir ini dapat memberikan manfaat serta menambah wawasan ilmu untuk pembaca dan juga bagi penulis sendiri.

Makassar, September 2022

Penulis

## ABSTRAK

Teknologi transportasi cerdas menjadi bagian yang penting dalam pengembangan *autonomous driving*, salah satunya adalah dalam mendeteksi objek hambatan. Pada saat mengemudikan mobil, akan ada banyak hambatan yang dapat menghalangi kelancaran saat berkendara dan bahkan dapat mengakibatkan kecelakaan. Terdapat beberapa faktor penyebab terjadinya kecelakaan di jalan saat berkendara, yaitu faktor kendaraan, faktor manusia, dan faktor kondisi jalan. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun sistem deteksi objek hambatan *car*, *motorcycle*, *person* dan *pothole* dengan metode *Single Shot Multibox Detector* (SSD) dan arsitektur *mobilenet v2*. Data input berupa data video yang di ekstrak menjadi *frame*, dimana data tersebut diambil menggunakan sebuah *dash camera* yang dipasang di mobil. Jumlah dataset yang digunakan pada proses *training* yaitu 720 gambar untuk setiap objek hambatan. Adapun parameter latih pada proses *training* yaitu *num steps* 20000 dan *batch size* 16. Pada penelitian ini, metode SSD dapat diimplementasikan dengan baik dalam mendeteksi dan mengklasifikasikan objek hambatan secara *real time*, dimana hasil akurasi yang didapatkan pada kecepatan 10 km/jam, 20 km/jam, 30 km/jam, dan 40 km/jam masing-masing sebesar 93.88%, 97.22%, 95.83%, dan 94.44%. Adapun hasil performa model di beberapa resolusi gambar 2560x1600 piksel, 1280x800 piksel, 640x400 piksel, dan 320x200 piksel, didapatkan bahwa akurasi tertinggi di resolusi 2560x1600, dimana akurasi pada masing-masing objek hambatan *car*, *motorcycle*, *person*, dan *pothole* yaitu sebesar 100.00%, 66.37%, 100.00%, dan 100.00%.

**Kata kunci:** *autonomous driving*, hambatan, SSD, *mobilenet v2*, *real time*

## DAFTAR ISI

KATA PENGANTAR .....	ii
ABSTRAK.....	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR .....	viii
DAFTAR TABEL.....	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA .....	6
2.1 <i>Autonomous Driving</i> .....	6
2.2 <i>Computer Vision</i> .....	7
2.3 <i>Object Detection</i> .....	9
2.4 Convolutional Neural Network.....	10
2.5 <i>Single Shot Multibox Detector (SSD)</i> .....	14
2.6 <i>MobileNet V2</i> .....	17
2.7 <i>Confusion Matrix</i> .....	19
2.8 <i>Penelitian Terkait</i> .....	22
BAB III METODE PENELITIAN .....	25
3.1 Tahapan Penelitian .....	25

3.2	Waktu dan Lokasi Penelitian .....	26
3.3	Instrumen Penelitian.....	27
3.4	Teknik Pengambilan Data .....	27
3.5	Perancangan Sistem.....	28
3.6	Analisis Kerja Sistem .....	44
BAB IV HASIL DAN PEMBAHASAN .....		46
4.1	Hasil Penelitian .....	46
4.2	Pembahasan.....	67
BAB V PENUTUP .....		76
5.1	Kesimpulan .....	76
5.2	Saran .....	77
DAFTAR PUSTAKA .....		78
LAMPIRAN.....		80

## DAFTAR GAMBAR

Gambar 2.1 <i>Autonomous Driving</i> (Azani, 2018).....	7
Gambar 2.2 Arsitektur <i>Convolutional Neural Network</i> (Nadhifa, 2018).....	10
Gambar 2.3 Proses pada <i>Convolutional Layer</i> (Samuel, 2017).....	12
Gambar 2.4 Contoh Perhitungan REL-U.....	13
Gambar 2.5 Proses matematis pooling layer (a) <i>max pooling</i> menggunakan stride 2, (b) <i>average pooling</i> menggunakan stride 2 .....	13
Gambar 2.6 <i>Fully-Connected Layer</i> (Grattarola et al., 2017).....	14
Gambar 2.7 Arsitektur SSD (Liu et al., 2016) .....	16
Gambar 3.1 Diagram Alur Penelitian .....	25
Gambar 3.2 Ilustrasi Pengambilan Data .....	28
Gambar 3.3 <i>Flowchart</i> Perancangan Sistem.....	28
Gambar 3.5 Augmentasi <i>Horizontal Flip</i> .....	32
Gambar 3. 7 Augmentasi <i>Scale Zoom In</i> .....	33
Gambar 3.8 Augmentasi <i>Brightness</i> .....	33
Gambar 3.9 Augmentasi <i>Contrast</i> .....	33
Gambar 3.10 Arsitektur <i>MobileNetV2</i> .....	35
Gambar 3.11 Proses Konvolusi .....	36
Gambar 3.12 Proses Perpindahan dengan <i>Stride 1</i> .....	37
Gambar 3. 13 Proses Konvolusi Menggunakan <i>Padding</i> .....	38
Gambar 3.14 <i>Bottleneck Block MobileNet V2</i> .....	39
Gambar 3.15 Contoh proses konvolusi pada salah satu <i>blok bottleneck</i> .....	42
Gambar 4. 1 Grafik Performa Sistem di Beberapa Resolusi .....	62
Gambar 4. 2 Performa dari <i>Processor</i> di Resolusi 2560x1600 Piksel.....	63
Gambar 4. 3 Grafik Performa dari <i>Processor</i> di Resolusi 2560x1600 Piksel	63
Gambar 4. 4 Performa dari <i>Processor</i> di Resolusi 1280x800 Piksel.....	64
Gambar 4. 5 Grafik Performa dari <i>Processor</i> di Resolusi 1280x800 Piksel ..	64
Gambar 4. 6 Performa dari <i>Processor</i> di Resolusi 640x400 Piksel.....	65
Gambar 4. 7 Grafik Performa dari <i>Processor</i> di Resolusi 640x400 Piksel ...	65
Gambar 4. 8 Performa dari <i>Processor</i> di Resolusi 320x200 Piksel.....	66

Gambar 4. 9 Grafik Performa dari <i>Processor</i> di Resolusi 320x200 Piksel ....	66
Gambar 4. 10 Gambar <i>Pothole</i> pada Resolusi 320x200 Piksel (a) gambar asli (b) gambar <i>edge detection</i> .....	69
Gambar 4. 11 Contoh <i>Feature Map</i> Objek <i>Pothole</i> di Resolusi 320x200 Piksel.....	70

## DAFTAR TABEL

Tabel 2. 1 Garis Besar <i>Human Vision</i> dan <i>Computer Vision</i> .....	8
Tabel 2. 2 Perbandingan performansi metode DNN (Liu et al., 2016).....	15
Tabel 2. 3 <i>Multi-Class Confusion Matrix</i> .....	20
Tabel 3. 1 Data gambar dalam format CSV .....	31
Tabel 3.2 Parameter latih model SSD.....	34
Tabel 3.3 Konvolusi Layer pada <i>MobileNetV2</i> .....	39
Tabel 4. 1 <i>Confusion Matrix Skenario 1</i> dengan Kecepatan 10km/jam .....	47
Tabel 4. 2 Hasil Pengujian Skenario 1 dengan Kecepatan 10km/jam .....	47
Tabel 4. 3 <i>Confusion Matrix Skenario 2</i> dengan Kecepatan 20km/jam .....	48
Tabel 4. 4 Hasil Pengujian Skenario 2 dengan Kecepatan 20km/jam .....	49
Tabel 4. 5 <i>Confusion Matrix Skenario 3</i> dengan Kecepatan 30km/jam .....	50
Tabel 4. 6 Hasil Pengujian Skenario 3 dengan Kecepatan 30km/jam .....	50
Tabel 4. 7 <i>Confusion Matrix Skenario 4</i> dengan Kecepatan 40km/jam .....	51
Tabel 4. 8 Hasil Pengujian Skenario 4 dengan Kecepatan 40km/jam .....	52
Tabel 4. 9 Hasil Pengujian dengan Ukuran Gambar 2560x1600 Piksel.....	53
Tabel 4. 10 Hasil Pengujian dengan Ukuran Gambar 1280x800 Piksel.....	55
Tabel 4. 11 Hasil Pengujian dengan Ukuran Gambar 640x400 Piksel.....	57
Tabel 4. 12 Hasil Pengujian dengan Ukuran Gambar 320x200 Piksel.....	60
Tabel 4. 13 Nilai <i>Loss</i> pada <i>Num Steps</i> 20000 <i>tiap Batch Size</i> .....	68
Tabel 4. 14 Nilai <i>Loss</i> pada <i>Num Steps</i> 50000 <i>tiap Batch Size</i> .....	68
Tabel 4. 15 Hasil Pengujian Kesalahan Klasifikasi .....	72
Tabel 4. 16 Hasil Pengujian Klasifikasi Benar .....	73

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

*Autonomous Driving* yang juga dikenal sebagai mobil tanpa pengemudi adalah mobil yang memiliki kemampuan untuk berkendara layaknya dikemudikan oleh orang. Mobil ini bergerak dengan menggunakan rangkaian AI (*Artificial Intelligent*) serta rangkaian sensor yang dipasang pada mobil. “*Autonomous Driving* merupakan teknologi baru yang berpotensi untuk mengubah pola hidup masyarakat, dimana membantu orang dalam kehidupan sehari-hari dengan menyediakan layanan transportasi yang andal dan aman” (Azani, 2018).

*Autonomous driving* meringankan beban pengemudi dengan melakukan operasi cerdas, salah satunya dalam mendeteksi hambatan. Pada saat mengemudikan mobil, akan ada banyak hambatan yang dapat menghalangi kelancaran saat berkendara. Menurut Dirjen Perhubungan Darat, Budi Setyadi memaparkan “ada beberapa faktor penyebab terjadinya kecelakaan, yaitu faktor manusia (61%), faktor kendaraan (9%) dan faktor kondisi jalan (30%)” dari total kecelakaan sebanyak 109.244 berdasarkan data Integrated Road Safety Management System (IRSMS) Korlantas RI pada tahun 2019 (Raharja, 2021).

Dari ketiga faktor tersebut, faktor manusia dan kondisi jalan memiliki persentase paling tinggi, dimana hambatan yang paling umum penyebab terjadinya kecelakaan di jalan antara lain orang atau pejalan kaki dan lubang, ini semua membuat pengemudi tidak secara bebas dalam berkendara dan bahkan menyebabkan potensi bahaya tabrakan. Mendeteksi hambatan secara akurat dan

cepat sangat penting untuk kelancaran berkendara dan mencegah kecelakaan lalu lintas.

Algoritma *Convolutional Neural Network* (CNN) memiliki kinerja terbaik dalam klasifikasi citra secara umum dan telah terbukti keakuratan dalam klasifikasi gambar dan pendeteksian objek. CNN juga memperoleh tingkat kesalahan lebih rendah dalam kompetisi *ImageNet*. Berdasarkan pengembangan algoritma CNN, terdapat tiga algoritma yang memiliki performansi tinggi, yaitu algoritma *Faster RCNN*, *You Only Look Once (YOLO)*, dan *Single Shot Multibox Detector (SSD)* (Gianani et al., 2018)

SSD merupakan algoritma *single deep learning neural network* yang menggunakan *bounding boxes* untuk memperkirakan lokasi objek yang akan dideteksi. Menurut (Ryu and Kim, 2018; Nur Latifah, 2019) algoritma SSD yang paling kompatibel dengan kapasitas memori dan pemrosesan data yang terbatas. “Algoritma SSD juga mampu menghasilkan model data dengan kecepatan, akurasi, dan kapasitas yang memadai meskipun menggunakan dataset dalam jumlah sedikit. SSD memiliki komputasi dan nilai kecepatan deteksi yang tinggi dibanding metode lainnya, Metode SSD ini juga sangat cocok digunakan untuk mendeteksi objek secara *real time*” (Nur Latifah, 2019)

Berdasarkan latar belakang yang telah dipaparkan, maka di masa depan teknologi *autonomous driving* untuk mendeteksi hambatan-hambatan penyebab kecelakaan sangat dibutuhkan. Pada penelitian ini, penulis mengusulkan penggunaan metode *Single Shot Multibox Detector* untuk mendeteksi objek

hambatan yang ada di jalan, dengan harapan akan memberikan akurasi yang lebih baik agar tepat dalam mendeteksi objek.

## **1.2 Rumusan Masalah**

1. Bagaimana cara merancang dan membuat sistem deteksi dan klasifikasi hambatan pada pengembangan teknologi *Autonomous Driving*?
2. Bagaimana performa sistem dalam mendeteksi dan mengklasifikasikan hambatan dengan menggunakan metode *Single Shot Multibox Detector (SSD)*?

## **1.3 Tujuan Penelitian**

Tujuan yang ingin dicapai dari penelitian ini sebagai berikut:

1. Merancang dan membangun sistem deteksi dan klasifikasi hambatan pada pengembangan teknologi *Autonomous Driving*.
2. Menunjukkan performa sistem dalam mendeteksi dan mengklasifikasikan hambatan menggunakan metode *Single Shot Multibox Detector (SSD)*

## **1.4 Manfaat Penelitian**

Adapun manfaat yang dapat diperoleh dari penelitian ini sebagai berikut:

1. Sebagai pengembangan di dunia otomotif demi terwujudnya program *smart city* di kota Makassar.
2. Sebagai acuan untuk penelitian lebih lanjut pada pengembangan teknologi *autonomous driving* mengenai *object detection*.

## **1.5 Batasan Masalah**

1. Objek hambatan yang dideteksi adalah mobil, motor, orang, dan lubang.
2. Data diambil pada pukul 13.00 sampai 15.00 dengan kondisi cuaca cerah.

3. Menggunakan 1 buah *dashcam* dengan posisi kamera terletak pada kaca mobil bagian depan.
4. Sudut kemiringan kamera yaitu 90°.
5. Sistem dibuat menggunakan arsitektur *MobileNet*.
6. Sistem hanya fokus mendeteksi objek yang ada di jalan.
7. Kondisi objek lubang yang dideteksi tidak tergenang air.

## **1.6 Sistematika Penulisan**

Untuk memberikan gambaran singkat mengenai isi tulisan secara keseluruhan, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu:

### **BAB I PENDAHULUAN**

Bab ini menguraikan secara umum mengenai hal yang menyangkut latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, dan sistematika penulisan.

### **BAB II TINJAUAN PUSTAKA**

Bab ini berisi teori-teori tentang hal-hal yang mendasari dan yang berhubungan dengan penelitian, termasuk di dalamnya tentang *autonomous driving* dan metode-metode yang digunakan dalam penelitian.

### **BAB III METODOLOGI PENELITIAN**

Pada bab ini berisi tentang apa saja yang akan dilakukan pada saat penelitian, yang meliputi tahapan, waktu dan lokasi, instrumen penelitian, perancangan sistem, dan analisis kerja sistem

#### **BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil pengolahan data serta pembahasan yang disertai tabel hasil penelitian.

#### **BAB V PENUTUP**

Bab ini berisi tentang kesimpulan yang didapatkan berdasarkan hasil penelitian yang telah dilakukan serta saran-saran untuk pengembangan lebih lanjut.

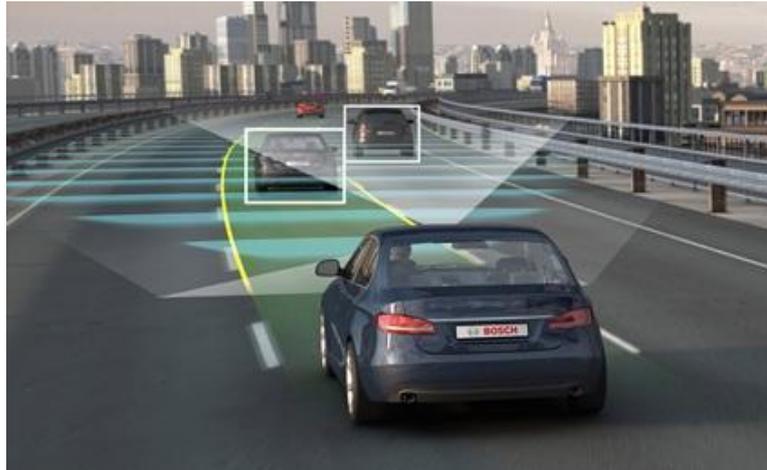
## BAB II

### TINJAUAN PUSTAKA

#### *2.1 Autonomous Driving*

*Autonomous Driving* atau dikenal sebagai mobil tanpa pengemudi. *Autonomous Driving* adalah mobil yang memiliki kemampuan untuk berkendara layaknya dikendalikan seperti manusia padahal mobil tersebut bergerak dengan rangkaian *Artificial Intelligence* (AI) serta terdiri dari kumpulan sistem-sistem yang saling bekerja sama untuk memungkinkan kendaraan tersebut berkendara (Azani, 2018).

Pada prinsipnya, *autonomous driving* menggunakan sejumlah sensor dan prosesor untuk mendeteksi kondisi yang ada di depan mobil serta mengambil keputusan layaknya manusia yang berkendara. Sensor paling utama yang digunakan adalah GPS (*Global Positioning System*), dimana digunakan untuk mengetahui posisi persis dari mobil. Kamera dan radar yang dipasang di sekitar mobil digunakan untuk memantau keadaan sekitar layaknya mata. Gelombang *infrared* serta *ultrasonic* yang digunakan untuk menghitung jarak secara akurat ke kendaraan lain. Sistem *autonomous* mengambil alih seluruh pengoperasian kendaraan, dari mulai menginjak gas, rem hingga putaran setir, dapat dilihat pada **Gambar 2.1** berikut:



**Gambar 2.1 *Autonomous Driving* (Azani, 2018)**

Proses-proses yang dilakukan dalam *self-driving* sangat bergantung pada teknologi *machine learning* dan *artificial intelligence*. Walaupun jalan yang ditempuh dalam pengembangan mobil *self-driving* ini masih panjang dan masih perlu banyak perbaikan di sana-sini, tapi *autonomous driving* ini merupakan solusi yang cukup logis untuk mengatasi masalah pengemudi yang ada di jalan. Hal ini dapat dilihat bahwa mayoritas kecelakaan di Indonesia disebabkan oleh kesalahan pengemudi, untuk itu dengan mengurangi control fisik seorang pengemudi, *autonomous driving* dapat mengurangi angka kecelakaan yang terjadi di jalan (Suhartono, 2017).

## **2.2 *Computer Vision***

*Computer vision* adalah ilmu dan teknologi mesin yang melihat dimana mesin mampu mengekstrak informasi dari gambar yang diperlukan untuk menyelesaikan tugas tertentu. Pada hakikatnya, *computer vision* mencoba meniru cara kerja visual manusia. Dalam proses penglihatan manusia, manusia melihat objek menggunakan indera penglihatan yaitu mata, lalu citra objek diteruskan ke

otak untuk diinterpretasikan sehingga manusia mengerti objek apa yang tampak. Hasil interpretasi ini kemudian digunakan untuk pengambilan keputusan (Munir & Yuliani, 2019).

**Tabel 2. 1** Garis Besar *Human Vision* dan *Computer Vision*

<i>Human Vision</i>	<i>Computer Vision</i>
Menggunakan mata dan visual <i>cortex</i> di dalam otak.	Menggunakan kamera-kamera yang terhubung pada sistem komputer
Menemukan dari gambar objek apa yang ada dalam penglihatan, dimana posisinya, bagaimana mereka bergerak, dan apa bentuknya.	Secara otomatis menginterpretasi gambar-gambar dan mencoba untuk mengerti isinya seperti pada <i>human vision</i>

**Tabel 2.1** menunjukkan gambaran umum antara *human vision* dan *computer vision*. Untuk itu *computer vision* mampu mengenali objek yang diamati. Kemampuan untuk mengenali ini merupakan kombinasi dari pengolahan citra dan pengenalan pola. Pengolahan citra adalah tahap awal dalam *computer vision* untuk menghasilkan citra yang lebih baik atau lebih mudah diinterpretasikan, sedangkan pengenalan pola adalah proses identifikasi objek pada citra. Proses-proses dalam *computer vision* secara garis besar dapat dibagi menjadi (Basuki & Yuliani, 2019):

- a. Proses mengakuisisi citra digital (*image acquisition*)
- b. Proses pengolahan citra (*image processing*)
- c. Proses analisis data citra (*image analysis*)
- d. Proses pemahaman data citra (*image understanding*)

*Computer vision* merupakan kombinasi antara *image processing* dan *pattern recognition*. Pengolahan citra (*image processing*) merupakan bagian dimana kualitas citra ditingkatkan. Pengenalan pola (*pattern recognition*) merupakan proses untuk mengelompokkan data masukan baik berupa data numerik maupun citra secara otomatis berdasarkan fitur yang dimiliki.

*Computer vision* adalah pembangunan deskripsi objek fisik yang eksplisit dan gamblang dari sebuah gambar. *Output* dari *computer vision* adalah deskripsi atau interpretasi atau beberapa pengukuran kuantitatif struktur dalam adegan 3D (Le & Yuliani, 2019)

### **2.3 Object Detection**

*Objek detection* merupakan tahap penting yang banyak diterapkan dalam proses pencarian gambar, *auto-annotation* gambar dan pemahaman gambar. *Object detection* atau deteksi objek dimana menentukan keberadaan suatu objek, ruang lingkungannya dan lokasi pada gambar. Deteksi objek dibagi menjadi dua yaitu *soft detection* dan *hard detection*. *Soft detection* hanya mendeteksi adanya objek sedangkan *hard detection* mendeteksi keberadaan objek dan lokasi objek (Jalled, 2016).

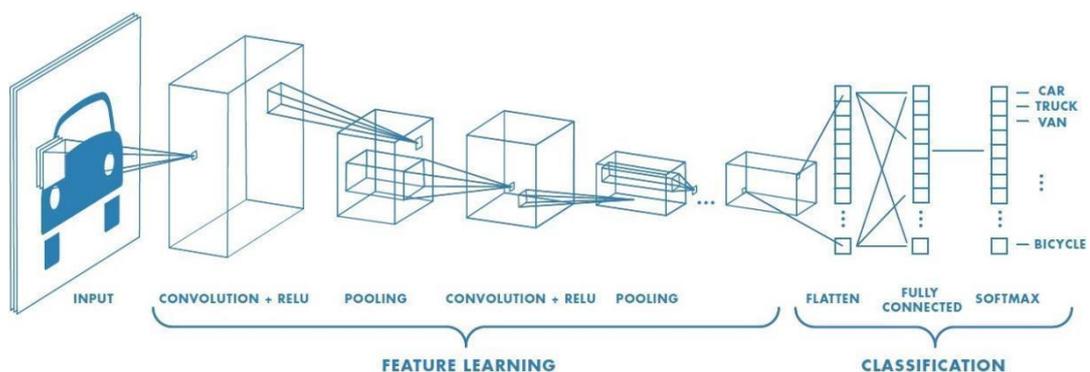
Sistem deteksi objek bekerja melalui tahapan *preprocessing*, ekstraksi fitur, dan klasifikasi. *Preprocessing* adalah pengolahan data asli untuk meningkatkan kualitas dari data sebelum diolah ke tahap berikutnya. Ekstraksi dan klasifikasi adalah sebuah proses untuk mendapatkan fitur dari data yang digunakan agar objek dapat dikategorikan jenisnya.

Proses pendekatan yang dapat digunakan untuk proses deteksi objek, yaitu *top-down* dan *bottom-up* atau kombinasi dari keduanya. Pendekatan *top-down* dilakukan dengan menyertakan tahap pelatihan untuk mendapatkan fitur-fitur kelas atau objek. Sedangkan pendekatan *bottom-up* dilakukan dengan menganalisis fitur gambar tingkat rendah atau menengah seperti tepi atau segmen (Al Kadafi, 2017).

## 2.4 Convolutional Neural Network

*Convolutional Neural Network* (CNN) merupakan salah satu algoritma dari *deep learning* yang merupakan hasil pengembangan dari *Multi-Layer Perceptron* (MLP) yang dirancang untuk melakukan olah data menjadi bentuk dua dimensi, misalnya gambar atau suara. CNN digunakan untuk melakukan klasifikasi data yang berlabel dengan menggunakan metode *supervised learning* yang dimana terdapat data yang dilatih dan terdapat data variabel yang ditargetkan sehingga tujuan dari metode yaitu mengelompokkan suatu data yang sudah ada.

Cara kerja dari CNN yaitu meniru cara kerja otak pada manusia yang terdiri dari berbagai *neuron*.



**Gambar 2.2** Arsitektur *Convolutional Neural Network* (Nadhifa, 2018)

**Gambar 2.2** menunjukkan bahwa arsitektur CNN memiliki beberapa lapisan tersembunyi (*hidden layer*) dimana masukan dari CNN berupa vektor

tunggal dan data yang dipropagasikan adalah data dua dimensi sehingga operasi linear pada CNN menggunakan konvolusi. Arsitektur CNN dibagi menjadi dua fitur yaitu: (Vivian dkk, 2020).

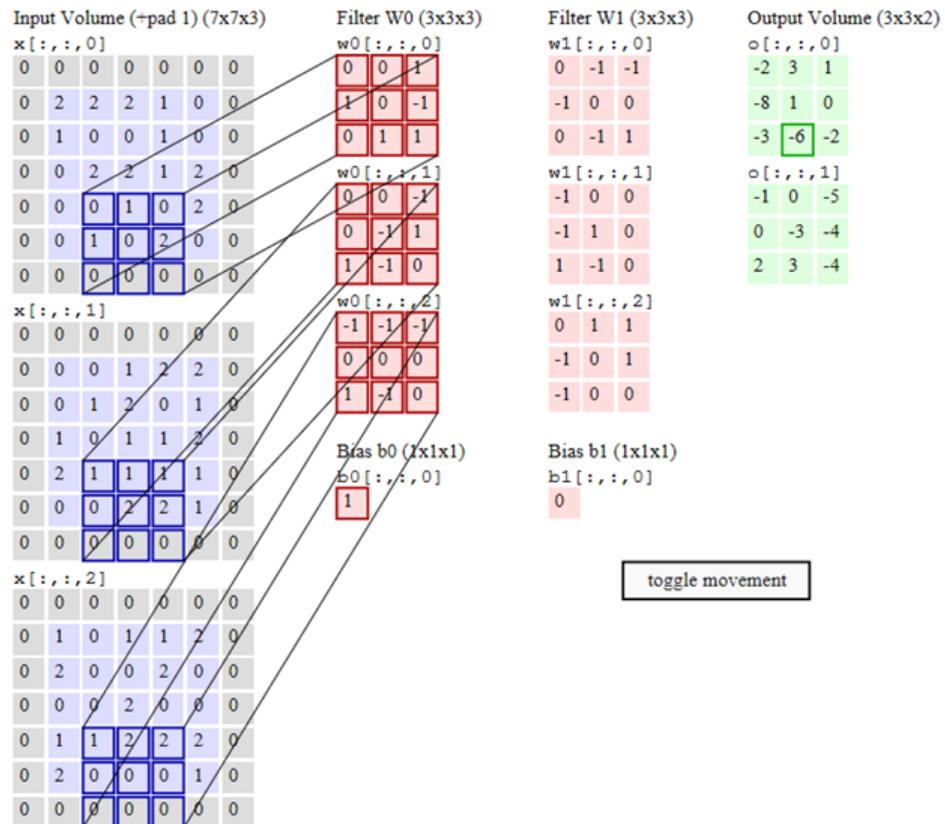
### 1. *Feature Extraction Layer*

Pada *feature extraction layer* terdapat dua lapisan, lapisan pertama adalah *convolutional layer* dan lapisan kedua adalah *pooling layer*. Lapisan ini menerima input gambar secara langsung dan memprosesnya sampai menghasilkan *output* berupa vektor untuk diolah pada lapisan berikutnya.

#### a. *Convolutional layer*

*Convolutional layer* adalah *layer* yang terdiri dari neuron yang tersusun sehingga membentuk sebuah *filter* yang berfungsi untuk deteksi dan mengekstraksi citra pada *input* yang diberikan. Cara kerja dari *convolutional layer* ini adalah melakukan konvolusi blok citra dengan matriks filter. Kemudian, hasilnya akan berupa angka yang merupakan *dot product* antara bagian *input* dengan *filter* yang digunakan.

Setelah itu, dengan menggunakan konvolusi *filter* pada kemungkinan posisi *filter* tersebut akan menghasilkan sebuah *activation map*. Proses kerja pada *Convolutional Layer* dapat dilihat pada **Gambar 2.3**



**Gambar 2.3** Proses pada *Convolutional Layer* (Samuel, 2017)

b. *Rectified Linear Unit (REL-U)*

*Rectified Linear Unit (REL-U)* adalah fungsi aktivasi pada jaringan *neural network*. Fungsi aktivasi ini terdapat fungsi *non-linier* yang menentukan nilai keluaran ketika mendapat masukan tertentu. Nilai keluaran yang didapat akan bernilai 0 jika negatif dan bernilai sama dengan nilai keluarannya apabila positif. Secara matematis dapat dituliskan sebagai berikut:

$$REL - U(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (2.1)$$

dimana variabel x adalah hasil keluaran dari fungsi aktivasi REL-U. Contoh perhitungan pada REL-U dapat dilihat pada **Gambar 2.4**

17	-3	10	7
-3	-6	9	-10
-9	2	-9	-14
0	-6	-7	-2

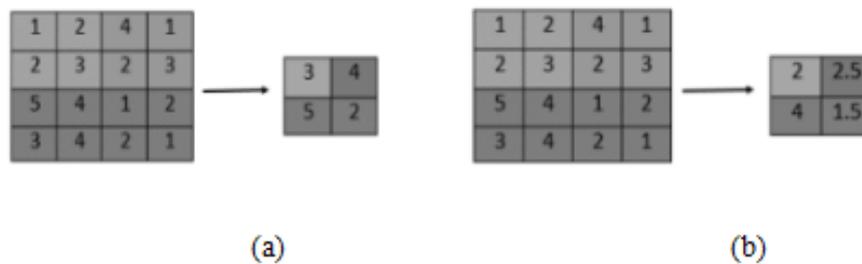
→

17	0	10	7
0	0	9	0
0	2	0	0
0	0	0	0

**Gambar 2.4 Contoh Perhitungan REL-U**

c. *Pooling Layer*

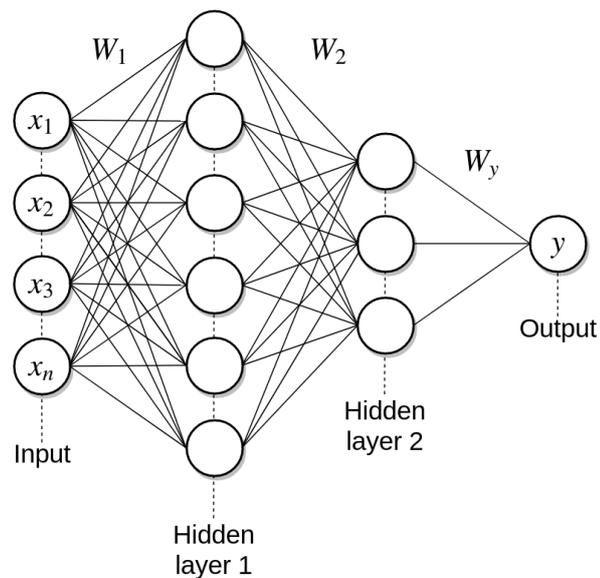
*Pooling layer* adalah lapisan yang mereduksi dimensi dari *feature map* secara parsial menggunakan operasi *downsampling* sehingga dapat mempercepat komputasi dan mengatasi *overfitting*. Fungsi *pooling* yang digunakan ada 2, yaitu *max pooling* dan *average pooling*. Konsep *max pooling* ini adalah dengan menentukan nilai maksimum pada tiap pergeseran *filter* dan mengambil *pixel* terbesar untuk menjadi matriks baru. Sedangkan *average pooling* mereduksi dengan menentukan rata-rata pada tiap pergeseran filternya. Proses matematis pada *pooling layer* dapat dilihat pada **Gambar 2.5**



**Gambar 2.5 Proses matematis pooling layer (a) *max pooling* menggunakan stride 2, (b) *average pooling* menggunakan stride 2**

## 2. Classification

Pada fitur *classification* terdapat *Fully-Connected Layer*, yang merupakan lapisan yang berfungsi dalam penerapan *multi perceptron* untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linier. Cara kerja layer ini sama seperti jaringan saraf manusia. Lapisan ini beroperasi pada layer yang disebut *hidden state* atau *hidden layer*. **Gambar 2.6** berikut merupakan *fully connected layer*:



**Gambar 2.6 Fully-Connected Layer (Grattarola et al., 2017)**

### 2.5 Single Shot Multibox Detector (SSD)

*Single shot multibox detector* (SSD) merupakan algoritma pengembangan *Deep Neural Network* (DNN) yang bertujuan untuk mendeteksi objek. Algoritma SSD ini bekerja berdasarkan pendekatan pada *feed forward convolutional network* yang nantinya menghasilkan kumpulan *bounding box* (*bbox*) dan skor untuk menentukan kategori dari objek pada masing-masing *bounding-box* tersebut.

Jaringan dasar yang digunakan pada SSD yaitu *VGG16* dan diikuti beberapa jaringan konvolusi. Adapun arti dari nama SSD itu sendiri yaitu:

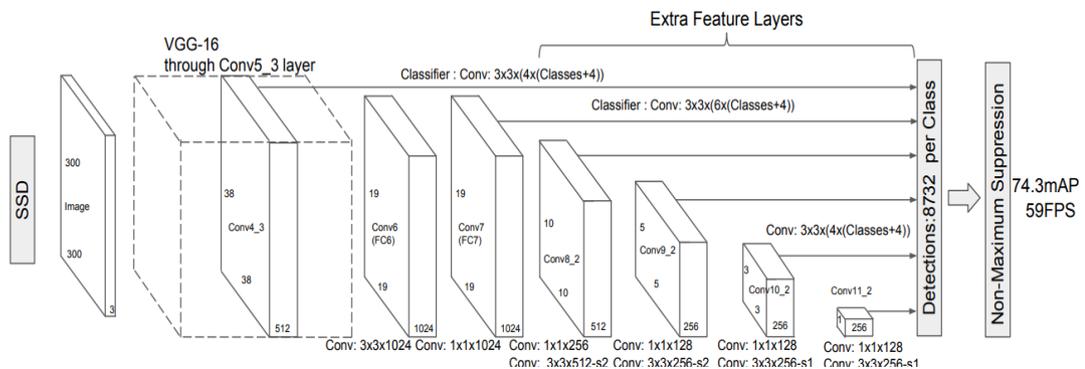
- *Single shot*, ini berarti bahwa tugas pelokalan dan klasifikasi objek dilakukan dalam *single forward pass* pada jaringan.
- *Multibox*, adalah nama teknik untuk regresi kotak pembatas.
- *Detector*, jaringan adalah *detector* objek yang juga mengklasifikasikan objek yang terdeteksi.

Berdasarkan penelitian Liu et al., terdapat tiga metode *deep learning* yang memiliki performansi tertinggi, yaitu *Faster RCNN*, *You Only Look Once (YOLO)*, dan *Single Shot Multibox Detector (SSD)*. Pada penelitian tersebut menguji waktu dan akurasi deteksi menggunakan dataset PASCAL VOC, COCO, dan ILSVRC dengan ukuran masukan yang bervariasi. Ketiga metode tersebut dapat mendeteksi objek secara *real time*, namun memiliki mAP, FPS, dan jumlah *bounding box* hasil deteksi untuk objek yang berbeda. *Mean Average Precision (mAP)* merupakan akurasi rata-rata presisi ukuran *bounding box* dengan ukuran objek pada citra. Penelitian ini dapat mengukur kecepatan deteksi objek dalam *frame per second (fps)*. Perbandingan ketiga metode DNN ditunjukkan pada **Tabel 2.2**.

**Tabel 2. 2 Perbandingan performansi metode DNN (Liu et al., 2016)**

<b>Metode</b>	<b><i>Bounding box</i></b>	<b><i>Fps</i></b>	<b><i>mAP (%)</i></b>
Faster RCNN	300	7	73.2
Yolo	98	45	63.4
SSD	8732	46	74.3

Berdasarkan **Tabel 2.2** metode SSD memiliki akurasi tertinggi dibanding dengan metode *Yolo* dan *Faster RCNN*, dimana memperoleh mAP sebesar 74.3%, kecepatan deteksi 46fps dan jumlah hasil deteksi maksimal sebanyak 8732 *bounding box*.



**Gambar 2.7** Arsitektur SSD (Liu et al., 2016)

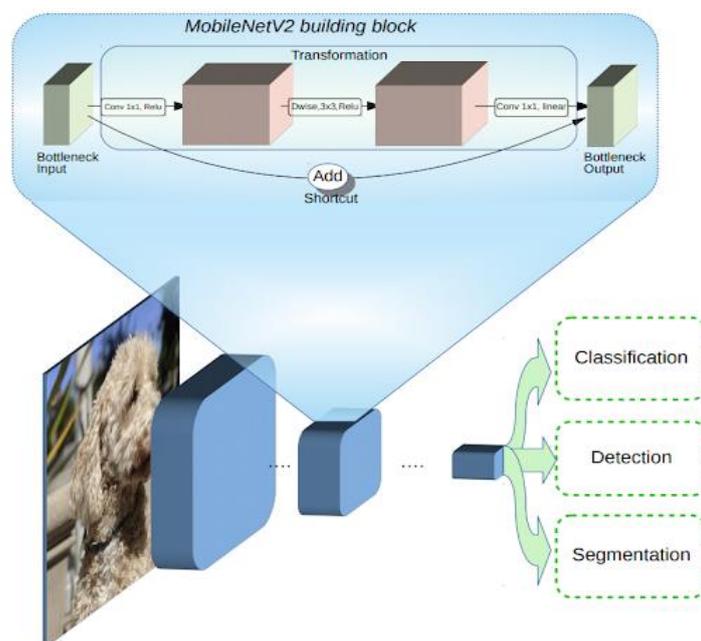
**Gambar 2.7** merupakan arsitektur dari SSD. SSD tersusun dalam beberapa proses konvolusi. Bagian awal SSD menggunakan arsitektur dari *VGG16* (Ranalli et al., 2018; Yuliani, 2019) yang merupakan salah satu algoritma klasifikasi citra berkualitas tinggi. SSD memiliki 3 fitur utama yaitu *Multiscale Feature maps*, *Convolutional Predictor*, *Default Box and Aspect Ratio*. Alasan *VGG16* digunakan sebagai jaringan dasar adalah karena kinerjanya yang kuat dalam tugas klasifikasi gambar berkualitas tinggi.

*VGG16* pada SSD terdiri dari 5 lapisan konvolusi yang masing-masing menghasilkan *feature map* yang menjadi masukan pada lapisan setelahnya. Pada layer konvolusi keempat, *feature map* yang dihasilkan digunakan sebagai masukan untuk proses *localization* dan *classification* pertama. Setiap *cell* (lokasi, koordinat) pada *feature map* digunakan untuk membentuk 4 objek prediksi. Setiap prediksi terdiri dari satu *bbox* dan  $N + 1$  skor untuk tiap kelas (tambahan 1 untuk kelas

‘bukan objek’) dimana  $N$  adalah jumlah kelas. SSD menggunakan *filter* konvolusi  $3 \times 3$  untuk melakukan prediksi tiap *cell* pada *feature map*. Tiap *filter* tersebut menghasilkan *output* dengan *channel*  $N + 1 + 4$  dimana tambahan 4 adalah lokasi dari *bbox* ( $x, y, w, h$ ). Misalkan pada lapisan konvolusi keempat dari *VGG16* menghasilkan *feature map* dengan ukuran  $38 \times 38 \times 512$ , maka SSD akan menghasilkan output sebesar  $38 \times 38 \times 4 \times (N+1+4)$ .

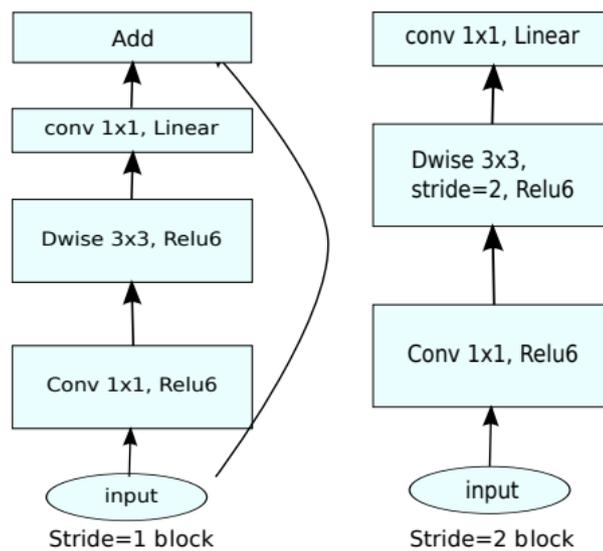
## 2.6 MobileNet V2

*Mobile Net* merupakan salah satu arsitektur *Convolutional Neural Network* (CNN) yang dapat digunakan untuk mengatasi kebutuhan akan *computing resource* berlebih. Kelebihan yang dimiliki oleh arsitektur *mobilenetv2* ini yaitu memiliki ketebalan dari *filter* konvolusi yang sesuai dengan gambar, menjadikannya lebih hemat ukuran dari model yang dibuat. Struktur dasar dari arsitektur ini dapat dilihat pada **Gambar 2.8**.



**Gambar 2.8** Arsitektur *MobileNet V2* (Ekoputris, 2018)

*MobileNet V2* merupakan pengembangan dari *MobileNet V1* sebelumnya yang mendukung untuk pengenalan visual termasuk klasifikasi, deteksi dan segmentasi. *MobileNet V2* dibangun menggunakan konvolusi yang dapat dipisahkan secara mendalam sebagai blok bangunan yang efisien. Terdapat dua komponen utama yang terdapat pada *mobilenetv2*, yaitu *Inverted Residual Block with Stride of 1* dan *Bottleneck Residual Block with Stride of 2* (Open Genus, 2022). *MobileNet V2* juga memiliki dua jenis lapisan konvolusi yaitu *1x1 Convolution* dan *3x3 Depthwise Convolution*. Setiap blok memiliki 3 lapisan yang berbeda, diantaranya *1x1 Convolution with Relu6*, *Depthwise Convolution*, dan *1x1 Convolution without any linearity*. **Gambar 2.9** berikut merupakan dua komponen model pada *mobilenet v2*:



**Gambar 2.9** Dua Komponen Model *MobileNet V2* (Sandler, 2018)

***Stride 1 Block:***

a. *Input*

b. *1x1 Convolution with Relu6*

- c. *Depthwise Convolution with Relu6*
- d. *1x1 Convolution without any linearity*
- e. *Add*

***Stride 2 Block:***

- a. *Input*
- b. *1x1 Convolution with Relu6*
- c. *Depthwise Convolution with stride 2 and Relu6*
- d. *1x1 Convolution without any linearity*

Perbedaan model *MobileNet V1* dan *MobileNet V2* yaitu algoritma serta pada struktur dari masing-masing model, untuk model *MobileNet V1* hanya terdapat satu layer, sedangkan pada model *MobileNet V2* terdapat dua layer yang membuat waktu proses pada model ini lebih cepat dibandingkan dengan versi satu. Secara keseluruhan, model *MobileNet V2* lebih cepat untuk akurasi yang sama di seluruh spektrum *latency*. Sehingga, *MobileNet V2* sangat efektif untuk deteksi dan segmentasi objek. (Nufus dkk, 2021)

## ***2.7 Confusion Matrix***

Berdasarkan jumlah keluaran kelas nya, sistem klasifikasi dibagi menjadi 4 jenis, yaitu klasifikasi *binary*, *multi-class*, *multi-label* dan *hierarchical*. Secara umum penentuan baik tidaknya performa suatu model klasifikasi dapat dilihat dari parameter pengukuran performanya, yaitu *recall*, *precision*, *f1-score*, dan *accuracy*. Untuk menghitung faktor-faktor tersebut diperlukan sebuah matriks yang biasa disebut dengan *confusion matrix*.

Terdapat empat kondisi untuk mempresentasikan hasil proses klasifikasi, yaitu (Mohajon & Tedy, 2021):

- a. *True Positive* (TP), keadaan dimana *classifier* memprediksi dengan benar kelas positif sebagai positif.
- b. *True Negative* (TN), keadaan dimana *classifier* memprediksi dengan benar kelas negative sebagai negative.
- c. *False Positive* (FP), keadaan dimana *classifier* salah, yaitu memprediksi kelas negative sebagai positif.
- d. *False Negative* (FN), keadaan dimana *classifier* salah, yaitu memprediksi kelas positif sebagai negative

Pada **Tabel 2.3** merupakan contoh ilustrasi *multi-class confusion matrix* ditinjau pada performa sistem dalam mengenali kelas B.

**Tabel 2. 3 Multi-Class Confusion Matrix**

		<i>PREDICTED</i>			
		<i>Classes</i>	<b>A</b>	<b>B</b>	<b>C</b>
<i>ACTUAL</i>	<b>A</b>	<b>TN</b>	<b>FP</b>	<b>TN</b>	<b>TN</b>
	<b>B</b>	<b>FN</b>	<b>TP</b>	<b>FN</b>	<b>FN</b>
	<b>C</b>	<b>TN</b>	<b>FP</b>	<b>TN</b>	<b>TN</b>
	<b>D</b>	<b>TN</b>	<b>FP</b>	<b>TN</b>	<b>TN</b>

Dalam *multi-class classification*, kelas-kelas dicantumkan dalam urutan yang sama dalam baris. Kelas-kelas yang diklasifikasikan dengan benar terletak

pada diagonal utama dari kiri atas ke kanan bawah dan merupakan titik pertemuan antara prediksi dan aktual (Grandini et al., 2020).

Metrik kinerja dari sebuah algoritma adalah *recall*, *precision*, *f1-score*, dan *accuracy*, yang dihitung berdasarkan nilai dari TP, TN, FP, dan FN (Singh et al., 2021).

a. *Recall*

*Recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang benar positif dan salah negatif. Data positif yang diklasifikasikan dengan benar dibagi dengan total data positif.

$$Recall = \frac{(TP)}{(TP+FN)} \times 100\% \quad (2.2)$$

b. *Precision*

*Precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Data positif yang diklasifikasikan dengan benar dibagi dengan total data yang diklasifikasikan sebagai data positif.

$$Precision = \frac{(TP)}{(TP+FP)} \times 100\% \quad (2.3)$$

c. *F1-Score*

*F1-Score* adalah nilai yang di dapat untuk menunjukkan performa algoritma yang diterapkan. *F1-Score* menunjukkan keseimbangan antara *recall* dan *precision*.

$$F1 - Score = 2X \frac{(Precision \times Recall)}{(Precision+Recall)} \times 100\% \quad (2.4)$$

d. *Accuracy*

*Accuracy* adalah rasio prediksi benar (positif) dengan keseluruhan data.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100\% \quad (2.5)$$

## **2.8 Penelitian Terkait**

Berikut ini merupakan beberapa penelitian terkait dengan penelitian yang dilakukan:

**1. *Obstacle Classification and Detection for Vision Based Navigation for Autonomous Driving* (Deepika dkk. 2017)**

Pada penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) dan arsitektur *SegNet encoder decoder*. Dimana metode yang diusulkan telah berhasil mengedepankan algoritma yang efisien dalam hal akurasi. Algoritma tersebut mampu mencetak rata-rata *Intersection Over Union* (IoU) sebesar 0.7631.

**2. *Real-Time Category-Based and General Obstacle Detection for Autonomous Driving* (Noa dkk. 2017)**

Penelitian ini menggunakan metode *Automated Ground Truth* (AGT). Hasil yang di dapat peneliti berhasil meningkatkan deteksi *obstacle* secara umum. Dimana metode AGT yang dikombinasikan dengan arsitektur jaringan dan kumpulan data versi *StixelNet*.

**3. *Obstacle Detection and Classification using Deep Learning for Tracking in Hight-Speed Autonomous Driving* (Prabhakar dkk. 2017)**

Prabhakar dkk melakukan penelitian pada tahun 2017, dimana dalam penelitiannya menggunakan metode *Region based Convolutional Neural*

*Network (R-CNN)* dengan hasil implementasi sistem *GPU Titan X* mencapai kecepatan pemrosesan 10 fps untuk citra resolusi *VGA*. *Frame rate* yang cukup tinggi menggunakan *GPU* yang kuat menunjukkan kesesuaian sistem untuk *autonomous driving* di jalan raya. Hasil deteksi dan klasifikasi citra dari *KITTI* dan *iRoads* serta jalan raya di india menunjukkan kinerja sistem yang tidak berubah-ubah terhadap bentuk dan tampilan objek serta pencahayaan dan kondisi iklim berbeda.

**4. *Multi Task Learning for Dangerous Object Detection in Autonomous Driving (Chen dkk. 2018)***

Pada penelitian Chen dkk menggunakan metode *Cartesian Product-Multi Task Learning (CP-MTL)*. Dimana hasil yang diperoleh secara matematis membuktikan bahwa *cartesian product-based* yang diusulkan lebih optimal daripada *linear multi-task*.

**5. *Obstacle Detection and Tracking Method for Autonomous Vehicle Based on Three-Dimensional LiDAR (Xie dkk. 2019)***

Xie dkk dalam penelitiannya pada tahun 2019, menggunakan metode *Random Sample and Consensus (RANSAC)*. Algoritma yang digunakan berhasil mengelompokkan *obstacle* secara efisien. Berdasarkan hasil *clustering*, *obstacle* statis dideteksi dengan metode *fusi multi frame* dan *obstacle* yang bergerak akan dideteksi lebih lanjut.

**6. *Detection and Distance Estimation against Motorcycles as Navigation Aids for Visually-impaired People* (Nur Latifah dkk 2019).**

Penelitian ini pada tahun 2019, peneliti menggunakan metode *Single Shot Multibox Detector* (SSD) dan *Pinhole Model*. Hasil pengujian deteksi motor tertinggi diperoleh pada *learning rate*  $10^{-3}$  dengan akurasi 95.04%, RMSE terendah estimasi jarak sebesar 0.12, dan akurasi estimasi posisi relatif sebesar 80% pada jarak 2 meter.