

IMPLEMENTASI *MOBILENETV2* DAN *FRAME DIFFERENCE* UNTUK PENENTUAN KECEPATAN KENDARAAN

Ridho Tadjudin¹, Dewi Rosmala²

Program Studi Informatika, Fakultas Teknologi Industri
Institut Teknologi Nasional

Jl. PKH Mustofa No.23, Neglasari, Cibeunying Kaler, Kota Bandung
ridhotadjudin@gmail.com¹, d_rosmala@itenas.ac.id²

ABSTRAK

Perangkat kamera di jalan dapat digunakan untuk mendeteksi masalah seperti pelanggaran batas kecepatan yang dapat menimbulkan kecelakaan berakibat pada luka serius hingga kematian. Penelitian ini berfokus pada sistem yang dapat melakukan penghitungan kecepatan kendaraan menggunakan metode *Frame Difference* dari data video yang terlebih dahulu telah diekstraksi menjadi rangkaian citra dan dilakukan proses segmentasi objek menggunakan *MobileNetV2*. *Frame Difference* digunakan untuk menghitung nilai perpindahan objek mobil pada setiap rangkaian citra. Nilai perpindahan lalu digunakan untuk menentukan kecepatan perpindahan mobil pada video. Dari hasil yang didapatkan, sistem memiliki akurasi masing-masing 98% pada citra segmentasi, 74,81% untuk penentuan titik acuan, dan 83,84% untuk titik acuan kanan dan 79,81% untuk titik acuan kiri pada perhitungan kecepatan mobil.

Kata kunci: smart city, data video, frame difference, *MobileNetV2*, mobil.

ABSTRACT

Camera devices on the road can be used to detect problems such as violations of speed limits which can lead to accidents that can be serious injury or death. This research focuses on a system that can calculate vehicle speed using the Frame Difference method from the video data that has been extracted first into a series of images and the object segmentation process is carried out using the MobileNetV2. Frame Difference is used to calculate the displacement value of the car object in each series of images. The displacement value is then used to determine the displacement speed of the car in a video. From the results obtained, the system has an accuracy each of 98% for car segmented image, 74.81% for determining

the reference point, and 83.84% for the right reference point and 79.81% for the left reference point is obtained in determining the speed of the car.

Keywords: smart city, video data, frame difference, MobileNetV2, car.

I. PENDAHULUAN

Kecelakaan lalu lintas menjadi penyebab utama kematian pada rentang usia 15-29 tahun di seluruh dunia. Salah satu cara untuk mengurangi angka kecelakaan lalu lintas adalah dengan mengatur dan menegakkan penggunaan batas kecepatan kendaraan pada tipe jalan tertentu (World Health Organization, 2010). Sistem pengukuran kecepatan berbasis video diketahui sebagai implementasi *Intelligent Transportation System (ITS)* yang bagus dikembangkan karena kapabilitas dan biaya murah. Penentuan pelanggaran kecepatan kendaraan juga dapat dikembangkan melalui deteksi gerakan (Tourani, Shahbami, Akoushideh, Khazae, & Suen, 2019).

Kemunculan metode-metode pengenalan pola, bentuk benda, dan pengukuran berbasis pengolahan citra telah dimanfaatkan pada beragam aplikasi kehidupan (Reddy, Priya, & Neelima, 2015). Sejumlah sistem pelacakan objek tunggal muncul, tetapi lebih kompleks lagi berupa pelacakan objek untuk mengetahui kecepatan pergerakan dari target dan pencocokkan bingkai ke bingkai dalam urutan video (Walia & Kapoor, 2016).

Untuk dapat menghitung kecepatan kendaraan yang melewati suatu jalan maka sistem harus mampu melakukan deteksi terhadap kendaraan yang terekam kamera. Sistem deteksi gerakan merupakan suatu sistem yang mampu mendeteksi gerakan yang terjadi di dalam video (Firmansyah & Supriana, 2015).

Deteksi gerakan merupakan subjek penting dalam bidang computer vision yang digunakan oleh banyak sistem pada aplikasi video pengawas, monitoring trafik, kompresi video, perhitungan kecepatan dan sebagainya (Luvizon, Nassu, & Minetto, 2017). Pengukuran kecepatan kendaraan berbasis pengolahan citra digital telah banyak dilakukan. Metode yang umum digunakan untuk pengukuran kecepatan kendaraan dengan citra video diantaranya *corner detection*, *texture analysis*, *video tracking*, dan seterusnya (Goyal & Singhai, 2018).

Teknologi pengenalan objek dan pemrosesan citra dapat digunakan untuk mendeteksi bentuk kendaraan, salah satunya adalah mobil. *Convolutional Neural Network* (CNN) merupakan metode yang dapat digunakan untuk pemrosesan citra. Salah satunya ialah *DeeplabV3* dengan arsitektur *pre-trained MobileNetV2* untuk mendeteksi piksel mobil pada sebuah citra (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018). Model ini memiliki kinerja yang baik dengan tingkat komputasi kecil (Howard, et al., 2017) sehingga dapat mengoptimalkan kinerja sistem. *Output* yang dihasilkan adalah piksel yang sudah terdapat kode piksel mobil hasil deteksi. Setelah diketahui nilai piksel mobil, maka dilakukan pencarian ujung kanan (Tissainayagam & Suter, 2005) sebagai titik acuan mobil. Pergerakan objek yang horizontal memungkinkan proses perhitungan menggunakan acuan sumbu *x* searah (Xiang & Xiaoqing, 2011).

Setelah diketahui titik kanan pada setiap citra dalam rangkaian kemudian digunakan metode *Frame Difference* untuk mengukur perpindahan objek mobil antar frame. Nilai perpindahan kemudian dikalkulasi untuk mendapatkan nilai kecepatan objek dalam video dan kemudian dikonversi ke dalam kecepatan Kilometer per Jam (Km/Jam).

Rumusan masalah pada penelitian ini dirumuskan sebagai berikut :

1. Bagaimana mengukur kinerja *frame difference* untuk menentukan nilai perpindahan objek pada rangkaian citra.
2. Bagaimana menghitung kecepatan berdasarkan nilai perpindahan objek menggunakan faktor skala.

Tujuan penelitian ini untuk mengukur kinerja sistem dalam memproses video dan penggunaan *Frame Difference* untuk menentukan kecepatan objek mobil dalam video.

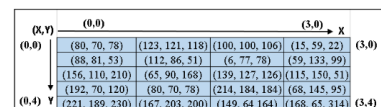
Untuk memfokuskan pembahasan pada penelitian ini, maka perlu diambil beberapa batasan masalah sebagai berikut.

1. Objek penelitian yaitu kecepatan mobil.
2. Sistem menganalisa objek tunggal dalam video.
3. Video berformat mp4.
4. Panjang lintasan dalam video uji adalah 11,57 meter.
5. Perekaman video uji dilakukan dalam 10 kecepatan berbeda yaitu 5 km/jam, 10 km/jam, 15 km/jam, 20 km/jam, 25 km/jam, 30km/jam, 35 km/jam, 40 km/jam, 45 km/jam, 50 km/jam dengan masing-masing kecepatan diambil 3 buah video uji.
6. Perekaman video uji menggunakan kamera digital Canon 1200D dengan resolusi video 1280x720 dan framerate 60 fps.
7. Model segmentasi menggunakan pretrained MobileNetV2.

II. LANDASAN TEORI

II.1 Citra

Citra adalah suatu nilai yang menciptakan suatu imitasi dari obyek seperti sebenarnya. Komputer membaca citra berisi *array* dengan bilangan *real* yang direpresentasikan dengan deretan bit, nilai berisi koordinat *x* dan *y* yaitu baris dan kolom yang disebut piksel. Nilai piksel ini bisa mewujudkan warna, titik, garis, dan bidang (Mirah, 2018). Gambar 1 merupakan contoh representasi koordinat dan piksel pada citra yang dibaca oleh komputer.



	(0,0)			(3,0)
(0,0)	(80, 70, 78)	(123, 121, 118)	(100, 100, 106)	(15, 59, 22)
	(88, 81, 53)	(112, 86, 51)	(6, 77, 78)	(59, 133, 99)
	(156, 110, 210)	(65, 90, 168)	(139, 127, 126)	(115, 150, 51)
	(192, 70, 120)	(80, 70, 78)	(214, 184, 184)	(68, 145, 95)
(0,4)	(221, 189, 230)	(167, 203, 200)	(149, 64, 164)	(168, 65, 314)
				(3,4)

Gambar 1. Koordinat dan piksel pada citra

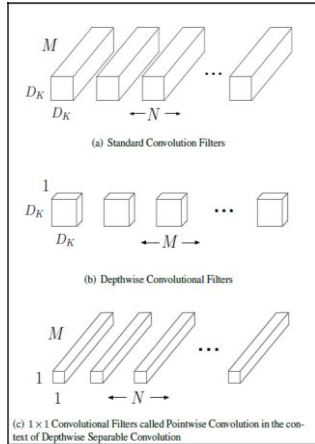
(Sumber: (Mirah, 2018))

Piksel pada citra ditunjukkan oleh kolom dari koordinat (0,0) hingga (3,4) dengan isi sejumlah bilangan. Koordinat dimulai dari titik (0,0) pada sudut kiri atas dari sebuah citra.

II.2 Pre-trained MobileNetV2

Pre-Trained adalah suatu model deep learning yang telah dilatih dengan sebuah data latih yang bertujuan mendapatkan model untuk deteksi sebuah objek. Penelitian ini menggunakan arsitektur

DeeplabV3 dengan backbone MobileNetV2 (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018) yang telah di-*training* terlebih dahulu menggunakan dataset Pascal VOC2012 (Visual Object Classes 2012). Pascal VOC adalah kumpulan dataset untuk mengenali objek dari sejumlah kelas objek visual dalam adegan realistik.



Gambar 2. Konvolusi Standar

(Sumber : (Howard, et al., 2017))

MobileNetV2 mengganti lapisan konvolusional standard dengan blok *depthwise convolution* dimana setiap blok terdiri dari lapisan konvolusional 3x3 yang menyaring input, diikuti oleh pointwise convolution 1x1 yang menggabungkan nilai-nilai yang difilter untuk menciptakan sebuah fitur baru.

Batch Normalization adalah teknik untuk *deep neural network* untuk membuat standard inputan kepada *layer* untuk setiap batch kecil (Brownlee, 2019). Fungsi ini memberikan efek menstabilkan proses pembelajaran sistem dan sangat berpengaruh untuk mengurangi epoch training yang dibutuhkan untuk melatih model *deep learning*.

ReLU (*Rectified Linear Unit*) merupakan fungsi linear yang digunakan untuk mengubah nilai x menjadi 0 jika x bernilai asal negatif dan tetap mempertahankan nilai x jika bernilai lebih dari 0. Fungsi ReLU pertama kali diperkenalkan oleh Geoffrey Hinton dan Vinod Nair untuk menggantikan fungsi aktivasi sigmoid (Pedomanti, 2018).

Operasi akan berulang sehingga setiap piksel pada citra input akan mempunyai output berupa nilai matriks kode hasil klasifikasi berdasarkan Pascal VOC2012. Hasil keluaran citra pada sistem ini memiliki tinggi 512 piksel dan lebar citra mengikuti lebar input citra.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

Gambar 3. Arsitektur MobileNetV2

(Sumber : (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018))

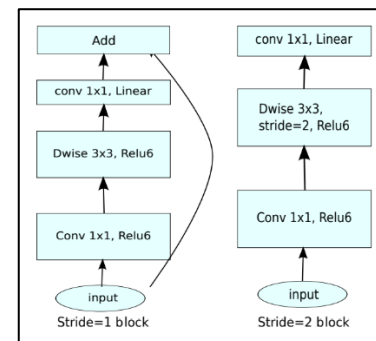
Pada *MobileNetV2*, terdapat input dan output antara model sedangkan lapisan atau layer bagian dalam model mengenkapsulasi kemampuan model untuk mengubah input dari konsep tingkat piksel hingga dapat mengetahui klasifikasi gambar. Sehingga yang akan digunakan pada penelitian ini yaitu *Pre-Trained MobileNetV2*.

Input	Operator	Output
$h \times w \times k$	1x1 conv2d, ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwse s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Gambar 4. Blok Penyusun MobileNetV2

(Sumber : (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018))

Berdasarkan blok penyusun *MobileNetV2* pada Gambar 4, dicantumkan detail arsitektur yang memiliki 32 filter pada fully convolutional layer, diikuti dengan 19 layer *residual bottleneck*. Kemudian digunakan ReLU6 sebagai variabel non-linear karena keunggulannya pada komputasi rendah. Standar kernel 3x3 digunakan, juga batch normalization pada tahap *training*. *Expansion factor t* digunakan pada nilai ukuran input (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018).



Gambar 5. Building Block MobileNetV2

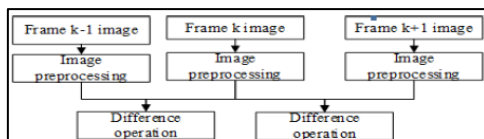
(Sumber : (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018))

Depthwise separable convolutions dapat menampilkan lebih banyak fitur dari parameter konvolusi yang lebih sedikit. *Width multipliers* merupakan parameter desimal yang mampu menskalakan total jumlah parameter dalam model keatas atau ke bawah, memungkinkan untuk menyeimbangan kekuatan model dengan biaya komputasi. *Linear bottlenecks* digunakan antara layer dan *shortcut connections* antara tiap *bottlenecks*. *Linear bottlenecks* berfungsi untuk mencegah kerusakan informasi, sedang *shortcut connections* memungkinkan proses *training* lebih cepat dan meningkatkan akurasi.

II.3 Frame Difference

Konsep dasar metode *frame difference* berupa perbandingan rangkaian frame citra yang ditangkap sesuai dengan urutan waktu. Rangkaian citra yang mengandung fitur geometri diekstrak, untuk kemudian dilakukan analisa (Singla, 2014). Frame pertama dipanggil sebagai frame referensi dan frame kedua dipanggil sebagai frame input. Keduanya dibandingkan dan nilai perbedaan dalam piksel ditentukan. Bisa juga dilakukan optimasi dengan menggabungkan algoritma lain kedalam *frame difference*, salah satunya adalah *joint difference*. *Joint difference* dikembangkan untuk mempercepat proses klasifikasi piksel. Menggunakan informasi frame difference untuk memperbaiki klasifikasi piksel sekaligus melakukan pembaruan klasifikasi (Migliore, Matteucci, & Naccari, 2006).

Dengan demikian, dalam metode *frame difference*, fitur objek yang diambil sesuai dengan perbedaan antara dua atau tiga frame secara terus menerus.



Gambar 6. Three Frame Difference

(Sumber : (Singla, 2014))

Menurut penelitian Rosin dan Ellis, Algoritma frame differences dilakukan melalui 3 tahap. Tahap pertama adalah proses penentuan citra referensi atau citra latar (Singla, 2014). Tahap kedua adalah proses operasi aritmatika *subtraction*, dan yang terakhir pengaturan ambang batas (*threshold*). Threshold merupakan bagian penting untuk penentuan ketelitian dari pendeteksian gerak. Menurut penelitian, nilai

threshold berbanding terbalik dengan jumlah noise yang dideteksi setelah dilakukan proses perbandingan. Semakin kecil nilai *threshold* suatu algoritma *frame difference*, maka semakin besar jumlah *noise* yang terdeteksi. Penelitian yang dilakukan oleh Rosin dan Ellis melakukan pengaturan *threshold* dengan menggunakan algoritma *automatic threshold* (Affrudin, 2017).

Sebuah nilai *threshold* dinyatakan berbanding terbalik terhadap jumlah *noise* yang dapat dideteksi setelah dibandingkan, hal tersebut menjelaskan bahwa jika nilai *threshold* kecil, maka akan semakin besar pula jumlah *noise* yang dapat terdeteksi (Yuha, Al Fiqri, Ashari, Pratama, & Harahap, 2019). Untuk bisa menghitung sebuah *frame difference* dapat dilakukan dengan melakukan substraksi antara dua buah frame. Citra hasil substraksi akan digunakan untuk menentukan dimana posisi objek yang bergerak (Affrudin, 2017).

Agar dapat membandingkan sebuah citra dengan waktu t, dengan citra t-1, dapat pula dilakukan dengan algoritma *Double Differences*, selanjutnya akan dilakukan perbandingan antara kedua citra t-1 dengan citra t-2 (Gupta, Singh, & Gupta, 2014).

II.4 Scale Factor Conversion

Scale factor conversion merupakan pendekatan konversi yang dikembangkan dengan mempertimbangkan arah kamera tegak lurus dengan jalur yang dilalui objek dan gerak objek dalam garis yang hampir lurus. Ini memungkinkan konversi bisa terjadi tanpa harus memiliki jarak minimal antara objek dan kamera (Khan, Porta, Vizzari, & Bandini, 2014). Faktor skala digunakan untuk melakukan konversi jarak yang dinyatakan dalam piksel menjadi jarak aktual yang dinyatakan dalam meter.

$$f_{skala} = \frac{\text{panjang sebenarnya}}{\text{panjang dalam citra}} \dots \dots \dots [1]$$

Berdasarkan Persamaan 1, maka untuk mendapatkan nilai faktor skala diperlukan nilai variabel panjang yang telah diketahui. Jika panjang sebenarnya dinyatakan dalam satuan meter maka hasil pengolahan menggunakan faktor skala tersebut akan dinyatakan dalam satuan meter (Lugianti, Sari, & Ningrum, 2019).

II.5 Penghitungan Kecepatan

Untuk mendapatkan hasil kecepatan objek bergerak dapat dihitung dengan mencari total jarak yang ditempuh dari awal objek bergerak sampai akhir

objek bergerak dibagi dengan waktu yang ditempuh dalam melakukan perpindahan (Lugianti, Sari, & Ningrum, 2019) Kecepatan kendaraan diperoleh dari citra hasil segmentasi yang telah ditentukan titik yang menjadi acuan pengukuran. Pada citra yang berurutan, setelah proses segmentasi piksel dan memberikan titik acuan maka harus ditentukan posisi awal dan akhir dengan referensi yang valid. Selama objek yang memiliki titik acuan berada dalam area yang telah ditentukan mulai dari titik awal sampai akhir maka jumlah citra yang diperlukan objek mobil selama melewati area dapat diketahui.

$$v_c = (\Delta d \times f_{skala}) \times fps \dots\dots\dots [2]$$

$$kec_{sistem} = v_c \times 3,6 \dots\dots\dots [3]$$

Keterangan:

- v_c : hasil kecepatan mobil (m/s)
- Δd : perpindahan rata-rata mobil
- f_{skala} : nilai faktor skala konversi
- fps : framerate video
- kec_{sistem} : hasil kecepatan mobil (km/jam)

Pada Persamaan 2 ditampilkan rumus untuk menghitung nilai kecepatan objek dalam rangkaian citra dengan menggunakan nilai perpindahan rata-rata dari titik acuan objek, dinyatakan dalam Δd . Variabel fps kemudian digunakan untuk mendapatkan nilai waktu dalam satuan detik.

Konversi nilai kecepatan meter per sekon ke kilometer per jam dapat dilakukan dengan melakukan pembagian nilai antara jam ke detik dan kilometer ke meter, yang kemudian didapatkan variabel 3,6 sebagai faktor pengali (Khan, Porta, Vizzari, & Bandini, 2014). Persamaan 2 digunakan untuk mengkonversikan nilai kecepatan yang didapatkan sistem dari kecepatan meter per sekon (m/s) menjadi kilometer per jam (km/jam).

III. ANALISIS DAN PERANCANGAN

III.1 Pengumpulan Data

Dalam penelitian ini data yang digunakan merupakan video rekaman kendaraan mobil yang diambil berdasarkan kecepatan tertentu.

Dalam pengambilan data untuk penelitian ini, lokasi yang dipilih adalah Lapangan MTQ Kota Salakan Sulawesi Tengah. Adapun alasan pemilihan lokasi ini yaitu merupakan area terbuka dengan sedikit

aktivitas manusia disekitarnya, lokasi umum yang bebas digunakan oleh masyarakat, dan kontur lokasi datar.

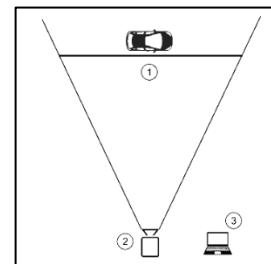


Gambar 7. Lokasi Lapangan MTQ Kota Salakan

(Sumber: (www.google.com/maps/@-1.3246071,123.2881956,16.25z))

Waktu pengambilan data pada penelitian ini dilakukan pada hari Jumat tanggal 12 Juni 2020 pukul 14.00 – 15.00 WITA.

Rekaman video kendaraan yang diambil memuat pergerakan satu objek mobil. Rekaman kemudian diklasifikasikan berdasarkan kecepatan mobil nyata mulai dari 5 km/jam, 10 km/jam, 15 km/jam, 20 km/jam, 25 km/jam, 30 km/jam, 35 km/jam, 40 km/jam, 45 km/jam, dan 50 km/jam. Pada masing-masing klasifikasi kecepatan diambil 3 video yang berbeda.



Gambar 8. Ekosistem Perekaman

Pada Gambar 8 ditampilkan beberapa komponen pendukung pada saat dilakukan perekaman video. Nomor 1 menandakan lintasan yang sudah ditandai yang menjadi lintasan mobil, nomor 2 merupakan kamera yang merekam video mobil bergerak, dan nomor 3 merupakan perangkat laptop digunakan untuk memproses video dan menampilkan hasil penghitungan kecepatan yang dilakukan oleh sistem.

Peralatan yang digunakan dalam pengambilan data pada penelitian ini antara lain:

1. Kamera Canon 1200D
2. Tripod
3. Alat pengukur panjang (meteran)

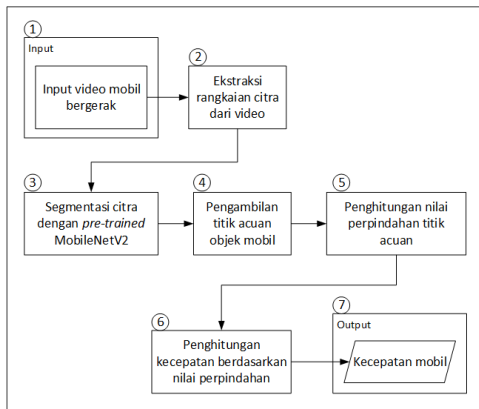
4. Objek penanda berwarna cerah
5. Mobil

III.2 Perancangan Sistem

Pada bagian ini menjelaskan mengenai prinsip kerja secara keseluruhan dari penelitian penentuan kecepatan kendaraan menggunakan metode *frame difference* berbasis video ini. Pada bagian perancangan ini diberikan diagram blok yang merepresentasikan tahapan proses pada Gambar 9.

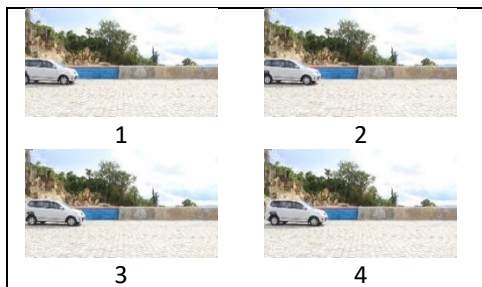
Sistem yang dibangun adalah penerapan *frame difference* untuk menentukan kecepatan objek mobil dalam video yang terdiri dari beberapa tahapan seperti ditampilkan pada Gambar 9.

Pada tahap 1 merupakan proses *input* video ke dalam sistem. Sistem yang telah dibangun mengambil data video dari akun *Google Drive*. Pada tahap 2 dilakukan proses ekstraksi rangkaian citra oleh sistem dengan menggunakan



Gambar 9. Blok Diagram

metode *keyframe extraction* melalui modul yang ditambahkan ke dalam sistem. Pada Gambar 9 ditampilkan contoh rangkaian citra hasil ekstraksi dari video uji 50_3.mp4.



Gambar 10. Contoh Rangkaian Citra Hasil Ekstraksi

Pada tahap 3 dilakukan proses segmentasi piksel mobil dengan menggunakan *pre-trained MobileNetV2*. Proses ini memberikan hasil keluaran berupa piksel 7 pada piksel hasil segmentasi yang terdapat objek mobil.



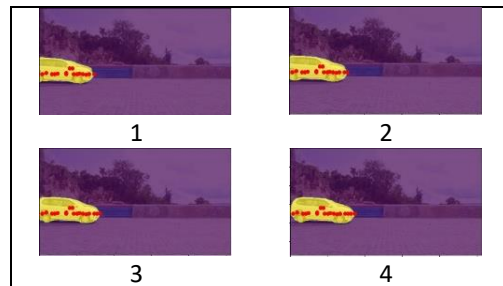
Gambar 11. Contoh Proses Segmentasi (kiri *input*, kanan *output*)

Pada tahap 4 sistem memproses citra hasil segmentasi untuk dilakukan pengambilan titik acuan dalam bentuk koordinat piksel citra, baik ujung kanan maupun ujung kiri objek mobil.



Gambar 12. Contoh Proses Penentuan Titik Acuan

Titik acuan yang telah didapatkan nilai koordinatnya kemudian dihitung nilai besaran perpindahan antar citra dalam rangkaian, proses ini dilakukan pada tahap 5.



Gambar 13. Contoh Rangkaian Citra Hasil dan Titik Acuan

Pada tahap 6 dilakukan proses perhitungan kecepatan objek mobil dalam rangkaian citra dengan menggunakan faktor skala untuk mendapatkan nilai kecepatan dengan satuan yang baku.

Hasil kecepatan: 46.46425781250001 km/jam

Gambar 14. Contoh Tampilan Hasil Kecepatan

Pada tahap 7 sistem menampilkan hasil *output* yang didapatkan dari keseluruhan proses dalam format kecepatan kilometer per jam (Km/Jam).

III.3 Deteksi Piksel Mobil

Pada proses ini digunakan model *pre-trained MobileNetV2* hasil penelitian Sandler dkk. (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018). Model ini bersumber dari Pascal VOC 2012 dan telah diuji dengan mengenali 21 objek. Model ini memiliki nilai *mean Intersection-over-Union* (mIOU) sebesar 75,32% dan akurasi sebesar 80,25%.

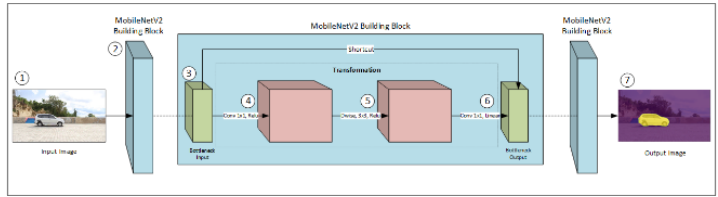
Proses ini digunakan untuk melakukan perubahan citra input menjadi *array* piksel yang telah ditandai oleh kode piksel objek pada citra hasil segmentasi.

Tabel 1. Kode Deteksi Objek Pascal VOC 2012

Kode Piksel	Nama Objek
0	Background
1	Pesawat Terbang
2	Sepeda
3	Burung
4	Kapal Laut
5	Botol
6	Bis
7	Mobil
8	Kucing
9	Kursi
10	Sapi
11	Meja Makan
12	Anjing
13	Kuda
14	Sepeda Motor
15	Manusia
16	Pot Tanaman
17	Domba
18	Sofa
19	Kereta Api
20	Monitor TV

Pada Tabel 1. terdapat 21 jenis benda yang dapat dideteksi beserta kode piksel yang menjadi *output* pada sistem yaitu background, pesawat terbang, sepeda, burung, kapal laut, botol, bis, mobil, kucing, kursi, sapi, meja makan, anjing, kuda, sepeda motor, manusia, tanaman di pot, domba, sofa, kereta api dan televisi (Everingham, 2012). Kode piksel 7 menjadi penanda bahwa terdapat piksel mobil pada suatu array piksel output pada citra hasil proses *pre-trained MobileNetV2*.

Tahapan proses deteksi piksel mobil yang dilakukan oleh sistem menggunakan model *pre-trained MobileNetV2* dapat dilihat pada Gambar 15. Pada bagian 1 merupakan citra hasil ekstraksi video uji yang menjadi citra *input* pada proses deteksi ini.

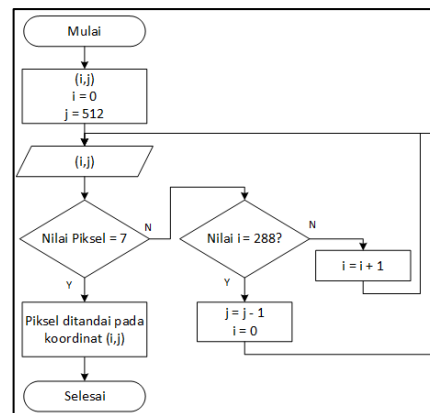


Gambar 15. Flowchart Segmentasi *MobileNetV2*

Bagian 2 merupakan *building block* model *MobileNetV2* yang digunakan. *Building block* ini terdiri dari dua jenis seperti yang ditampilkan pada Gambar 3 dan berjumlah 32 filter. Bagian 3 merupakan tahapan untuk mengoptimalkan kinerja memori pada sistem sembari tetap menjaga keefektifan proses deteksi. Bagian 5 merupakan operator pertama pada bagian transformasi nilai citra, dilakukan proses konvolusi matriks 1×1 dan ReLU (*Rectified Linear Unit*) komputasi rendah. Pada bagian 6 merupakan operator kedua yang terdapat proses *depthwise separable convolutions* untuk menampilkan fitur dari parameter konvolusi yang dilanjutkan dengan konvolusi matriks 3×3 dan ReLU. Bagian 7 merupakan proses konvolusi linear 1×1. Kemudian terdapat proses *shortcut* untuk mempercepat proses pada saat *training* dan meningkatkan akurasi. Kemudian pada bagian 9 merupakan citra hasil segmentasi menggunakan model *pre-trained MobileNetV2*.

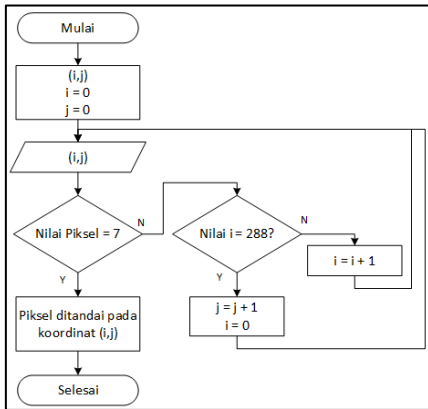
III.4 Penentuan Titik Acuan Mobil

Proses penentuan titik acuan menggunakan pencarian secara iteratif. Proses ini dilakukan untuk mengetahui titik-titik tertentu pada matriks segmentasi citra. Adapun titik yang digunakan adalah titik kanan objek dan titik kiri objek, keduanya digunakan sebagai acuan pergerakan.



Gambar 16. Flowchart Penentuan Ujung Kanan

Proses yang ditampilkan pada Gambar 16, dimulai dari koordinat (i,j) dimana j adalah lebar piksel citra dimulai dari koordinat 512, sedangkan i adalah tinggi piksel citra dimulai dari koordinat 0. Untuk sumbu i dilakukan iterasi + 1 jika pada sumbu (i_n,j) , apabila nilai j telah sama dengan 0 maka nilai j dilakukan iterasi -1 sampai terdapat piksel mobil. Piksel kemudian ditandai sebagai acuan kanan objek mobil.



Gambar 17. Flowchart Penentuan Ujung Kiri

Pada Gambar 17 ditampilkan proses penentuan ujung kiri objek, dimulai dari koordinat (i,j) dimana j adalah lebar piksel citra dimulai dari koordinat 0, sedangkan i adalah tinggi piksel citra dimulai dari koordinat 0. Untuk sumbu i dilakukan iterasi + 1 jika pada sumbu (i_n,j) , apabila nilai j telah sama dengan 0 maka nilai j dilakukan iterasi + 1 sampai terdapat piksel mobil. Piksel kemudian ditandai sebagai acuan kiri objek mobil.

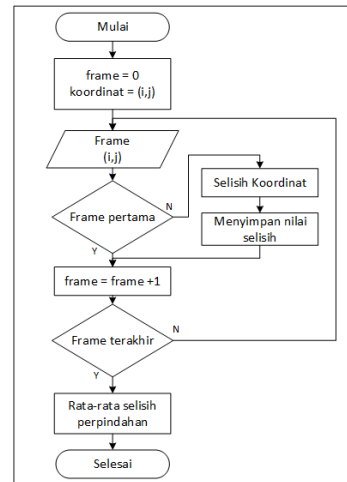
III.5 Frame Difference

Proses ini menggunakan metode *frame difference*. Proses ini dilakukan untuk mengetahui nilai perpindahan objek pada setiap citra hasil deteksi dalam rangkaian citra.

Dimulai dari frame 0 diambil titik koordinat (i,j) sebagai acuan awal. Kemudian dilakukan iterasi +1 pada frame citra untuk diambil nilai koordinat $(i,j)_n$. Selanjutnya dilakukan penghitungan selisih panjang antara dua titik pada kedua frame.

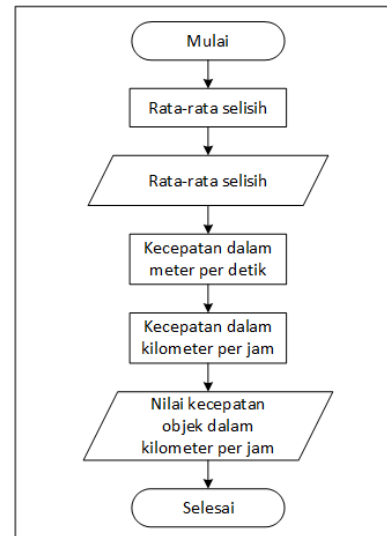
III.6 Penghitungan Kecepatan

Proses ini menggunakan metode *frame difference*. Proses ini dilakukan untuk mengetahui nilai perpindahan objek pada setiap citra hasil segmentasi.



Gambar 18. Flowchart Mendapatkan Nilai Selisih Perpindahan

Dimulai dari citra pertama dalam rangkaian citra yang memiliki piksel mobil diambil titik koordinat (i,j) sebagai acuan awal. Kemudian dilakukan iterasi +1 pada frame citra setelahnya untuk diambil nilai koordinat $(i,j)_n$.



Gambar 19. Flowchart Penghitungan Nilai Kecepatan

Selanjutnya dilakukan penghitungan selisih panjang antara kedua frame. Proses ini dilakukan berulang hingga tidak lagi terdapat citra yang memiliki piksel mobil di dalam rangkaian citra.

IV. HASIL DAN PEMBAHASAN

Pengujian dilakukan untuk mengetahui apakah setiap komponen dari sistem berfungsi sesuai dengan yang diharapkan. Pengujian sistem dilakukan dengan 3 skenario pengujian yang berbeda dengan tujuan yang berbeda. Ketiga jenis pengujian tersebut ditampilkan pada Tabel 2.

Tabel 2. Skenario Pengujian

Pengujian	Skenario Pengujian	Hasil
1	Melakukan pengujian deteksi mobil pada sistem dengan input citra hasil ekstraksi video	Mendapatkan nilai akurasi, presisi, recall, dan F1-score
2	Melakukan pengujian penentuan titik acuan pada citra hasil	Mendapatkan nilai akurasi penentuan titik acuan
3	Melakukan penghitungan jarak perpindahan mobil untuk mendapatkan nilai kecepatan dengan menggunakan dua titik acuan	Mendapatkan nilai hasil penghitungan kecepatan mobil dan akurasi

Data uji video yang digunakan didapatkan melalui rekaman langsung dengan total sebanyak 30 video pada saat dilakukan ekstraksi berjumlah setara dengan 5570 citra uji. Detail mengenai keseluruhan video uji ditampilkan pada Tabel 3.

Tabel 3. Detail Data Uji

No.	Nama Video	Kecepatan asli (km/jam)	Jumlah frame citra	Durasi (detik)
1	5_1.mp4	5	347	5.78
2	5_2.mp4	5	290	4.8
3	5_3.mp4	5	378	6.3
4	10_1.mp4	10	201	3.35
5	10_2.mp4	10	335	5.58
6	10_3.mp4	10	314	5.23
7	15_1.mp4	15	223	3.72
8	15_2.mp4	15	278	4.63
9	15_3.mp4	15	260	4.33
10	20_1.mp4	20	210	3.5
11	20_2.mp4	20	215	3.58
12	20_3.mp4	20	173	2.88
13	25_1.mp4	25	194	3.23
14	25_2.mp4	25	179	2.98
15	25_3.mp4	25	175	2.92
16	30_1.mp4	30	132	2.2
17	30_2.mp4	30	161	2.68
18	30_3.mp4	30	164	2.73
19	35_1.mp4	35	134	2.23
20	35_2.mp4	35	91	1.52
21	35_3.mp4	35	126	2.1
22	40_1.mp4	40	118	1.96
23	40_2.mp4	40	119	1.98

No.	Nama Video	Kecepatan asli (km/jam)	Jumlah frame citra	Durasi (detik)
24	40_3.mp4	40	105	1.75
25	45_1.mp4	45	112	1.86
26	45_2.mp4	45	114	1.9
27	45_3.mp4	45	104	1.73
28	50_1.mp4	50	123	2.05
29	50_2.mp4	50	99	1.65
30	50_3.mp4	50	96	1.6

Berdasarkan hasil pengujian deteksi mobil, didapatkan variabel yang digunakan untuk mengukur kinerja sistem yang ditampilkan pada Tabel 4.

Tabel 4. Hasil Pengujian Segmentasi Piksel Mobil (1)

No.	Nama Video	Nilai				Akurasi
		TP	TN	FP	FN	
1	5_1.mp4	308	37	0	2	99.42%
2	5_2.mp4	205	82	0	3	98.96%
3	5_3.mp4	232	138	0	8	97.88%
4	10_1.mp4	172	26	0	3	98.5%
5	10_2.mp4	165	165	0	5	98.5%
6	10_3.mp4	186	122	0	6	98.09%
7	15_1.mp4	140	79	0	4	98.2%
8	15_2.mp4	151	125	0	2	99.28%
9	15_3.mp4	149	108	0	3	98.84%
10	20_1.mp4	127	80	0	3	98.57%
11	20_2.mp4	132	80	0	3	98.6%
12	20_3.mp4	120	45	0	8	95.37%
13	25_1.mp4	117	76	0	1	99.48%
14	25_2.mp4	104	73	0	2	98.88%
15	25_3.mp4	107	65	0	3	98.28%
16	30_1.mp4	87	43	0	2	98.48%
17	30_2.mp4	91	68	0	2	98.75%
18	30_3.mp4	93	68	0	3	98.17%
19	35_1.mp4	75	55	0	4	97.01%
20	35_2.mp4	74	15	0	2	97.8%
21	35_3.mp4	76	48	0	2	98.41%
22	40_1.mp4	64	52	0	2	98.3%
23	40_2.mp4	62	54	0	3	97.47%
24	40_3.mp4	66	37	0	2	98.09%
25	45_1.mp4	56	54	0	2	98.21%
26	45_2.mp4	57	53	0	4	96.49%
27	45_3.mp4	58	44	0	2	98.07%
28	50_1.mp4	51	69	0	3	97.56%
29	50_2.mp4	51	46	0	2	97.97%
30	50_3.mp4	51	43	0	2	97.91%

Dari pengujian yang ditampilkan pada Tabel 4 dan diperoleh akurasi rata-rata keseluruhan 98,1% dari keseluruhan video uji.

Pengujian titik acuan mobil dilakukan untuk mendapatkan nilai akurasi pemberian titik acuan yang diproses oleh sistem. Pengujian ini dilakukan pada seluruh rangkaian citra pada kedua titik acuan yang digunakan. Pada Tabel 5 ditampilkan hasil pengujian

penentuan titik acuan kanan pada seluruh rangkaian citra.

Tabel 5. Hasil Pengujian Penentuan Titik Acuan Kanan

Nama Video	Jumlah citra terdeteksi	Penentuan Titik Acuan		Akurasi (%)
		Tepat	Salah	
5_1.mp4	310	188	122	60.64
5_2.mp4	208	135	73	64.9
5_3.mp4	240	148	92	61.66
10_1.mp4	175	127	48	72.57
10_2.mp4	170	108	62	63.52
10_3.mp4	192	137	55	71.35
15_1.mp4	144	114	30	79.16
15_2.mp4	153	127	26	83
15_3.mp4	152	128	24	84.21
20_1.mp4	131	101	30	77.09
20_2.mp4	135	101	34	74.81
20_3.mp4	128	100	28	78.12
25_1.mp4	118	92	26	77.96
25_2.mp4	106	78	28	73.58
25_3.mp4	110	86	24	78.18
30_1.mp4	89	67	22	75.28
30_2.mp4	93	74	19	79.56
30_3.mp4	97	79	18	81.44
35_1.mp4	79	50	29	63.29
35_2.mp4	76	57	19	75
35_3.mp4	78	57	21	73.07
40_1.mp4	66	47	19	71.21
40_2.mp4	65	47	18	72.3
40_3.mp4	68	52	16	76.47
45_1.mp4	58	41	17	70.68
45_2.mp4	61	45	16	73.77
45_3.mp4	60	47	13	78.33
50_1.mp4	54	39	15	72.22
50_2.mp4	53	44	9	83.01
50_3.mp4	53	41	12	77.35

Hasil pengujian penentuan titik acuan titik kiri pada seluruh rangkaian citra hasil deteksi ditampilkan pada Tabel 6.

Tabel 6. Hasil Pengujian Perhitungan Kecepatan Mobil Acuan Titik Kiri

Nama Video	Jumlah citra terdeteksi	Penentuan Titik Acuan		Akurasi (%)
		Tepat	Salah	
5_1.mp4	165	117	48	70.9
5_2.mp4	210	147	63	70
5_3.mp4	246	169	77	68.69
10_1.mp4	115	84	31	73.04
10_2.mp4	169	131	38	77.51
10_3.mp4	181	136	45	75.13
15_1.mp4	143	115	28	80.41
15_2.mp4	157	124	33	78.98
15_3.mp4	147	120	27	81.63
20_1.mp4	125	94	31	75.2

20_2.mp4	129	103	26	79.84
20_3.mp4	112	83	29	74.1
25_1.mp4	121	96	25	79.33
25_2.mp4	105	87	18	82.85
25_3.mp4	107	78	29	72.89
30_1.mp4	83	62	21	74.69
30_2.mp4	92	69	23	75
30_3.mp4	93	74	19	79.56
35_1.mp4	79	61	18	77.21
35_2.mp4	54	40	14	74.07
35_3.mp4	78	65	13	83.33
40_1.mp4	65	46	19	70.76
40_2.mp4	67	50	17	74.62
40_3.mp4	64	49	15	76.56
45_1.mp4	58	43	15	74.13
45_2.mp4	61	45	16	73.77
45_3.mp4	61	43	18	70.49
50_1.mp4	54	38	16	70.37
50_2.mp4	54	41	13	75.92
50_3.mp4	54	40	14	74.07

Hasil pengujian pada Tabel 5 dan Tabel 6, didapatkan nilai akurasi sebesar 74,81% pada kemampuan sistem untuk menentukan titik acuan.

Setelah proses penentuan posisi objek dilakukan, dilakukan pengujian nilai perpindahan yang didapatkan dari proses *frame difference*. Hasil pengujian perhitungan kecepatan dengan acuan titik kanan objek mobil ditampilkan pada Tabel 7.

Tabel 7. Hasil Pengujian Perhitungan Kecepatan Mobil Acuan Titik Kanan

Nama Video	Δd	Kecepatan Hasil	
		m/s	Km/Jam
5_1.mp4	1.52	2.07	7.45
5_2.mp4	1.59	2.16	7.78
5_3.mp4	1.71	2.31	8.32
10_1.mp4	2.89	3.92	14.13
10_2.mp4	2.98	4.05	14.58
10_3.mp4	2.65	3.59	12.95
15_1.mp4	3.53	4.78	17.23
15_2.mp4	3.31	4.49	16.18
15_3.mp4	3.33	4.52	16.29
20_1.mp4	3.92	5.31	19.14
20_2.mp4	3.77	5.11	18.43
20_3.mp4	3.95	5.35	19.29
25_1.mp4	4.29	5.81	20.94
25_2.mp4	4.83	6.55	23.61
25_3.mp4	4.63	6.28	22.61
30_1.mp4	5.7	7.73	27.84
30_2.mp4	5.51	7.47	26.89
30_3.mp4	5.51	7.47	26.92
35_1.mp4	6.5	8.81	31.72

35_2.mp4	6.65	9.02	32.47
35_3.mp4	6.54	8.87	31.94
40_1.mp4	7.75	10.51	37.84
40_2.mp4	7.82	10.61	38.2
40_3.mp4	7.5	10.17	36.64
45_1.mp4	8.87	11.63	41.87
45_2.mp4	8.3	11.25	40.51
45_3.mp4	8.49	11.51	41.44
50_1.mp4	9.45	12.81	46.14
50_2.mp4	9.48	12.85	46.27
50_3.mp4	9.51	12.9	46.46

Hasil pengujian perhitungan kecepatan dengan acuan titik kiri objek mobil ditampilkan pada Tabel 8.

Tabel 8. Hasil Pengujian Perhitungan Kecepatan Mobil Acuan Titik Kiri

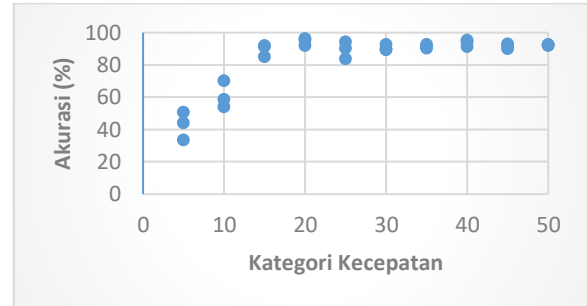
Nama Video	Δd	Kecepatan Hasil	
		m/s	Km/Jam
5_1.mp4	1.87	2.54	9.16
5_2.mp4	2.02	2.74	9.89
5_3.mp4	1.95	2.65	9.54
10_1.mp4	3.09	4.19	15.11
10_2.mp4	3.01	4.09	14.73
10_3.mp4	2.8	3.79	13.66
15_1.mp4	3.56	4.83	17.39
15_2.mp4	3.26	4.43	15.95
15_3.mp4	3.45	4.68	16.88
20_1.mp4	4.03	5.46	19.68
20_2.mp4	3.93	5.33	19.21
20_3.mp4	4.04	5.48	19.74
25_1.mp4	4.16	5.64	20.33
25_2.mp4	4.82	6.54	23.06
25_3.mp4	4.72	6.4	23.07
30_1.mp4	5.71	7.75	27.91
30_2.mp4	5.54	7.52	27.08
30_3.mp4	5.47	7.42	26.73
35_1.mp4	6.52	8.84	31.85
35_2.mp4	6.69	9.08	32.69
35_3.mp4	6.5	8.82	31.75
40_1.mp4	7.71	10.46	37.67
40_2.mp4	7.63	10.35	37.27
40_3.mp4	7.38	10	36.02
45_1.mp4	8.59	11.65	41.96
45_2.mp4	8.28	11.23	40.43
45_3.mp4	8.5	11.52	41.48
50_1.mp4	9.39	12.73	45.86
50_2.mp4	9.45	12.81	46.14
50_3.mp4	9.5	12.89	46.41

Setelah dilakukan penghitungan kecepatan mobil, sistem juga akan mengeluarkan output berupa nilai kecepatan mobil hasil perhitungan sistem.

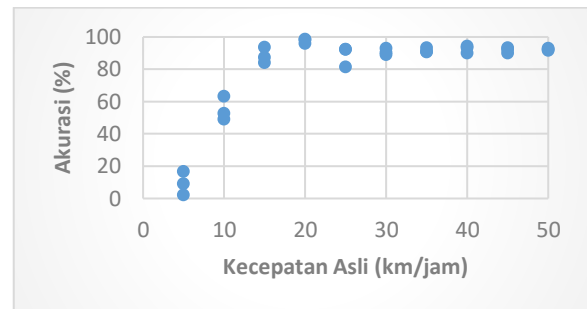
Kemudian, didapatkan nilai hasil akurasi yang dapat dilihat pada Gambar 20 dan Gambar 21.

V. KESIMPULAN

Dari hasil penelitian, sistem memiliki akurasi 98% pada segmentasi citra dengan menggunakan model pre-trained *MobileNetV2* dan *Frame Difference*.



Gambar 20. Grafik Akurasi Hitung Kecepatan Titik Acuan Kanan



Gambar 21. Grafik Akurasi Hitung Kecepatan Titik Acuan Kiri

Pada penentuan titik acuan didapatkan akurasi sebesar 74,81% pada seluruh rangkaian citra uji. Nilai akurasi yang didapatkan untuk perhitungan kecepatan cukup memuaskan dengan akurasi rata-rata pada nilai kecepatan dengan titik acuan kanan 83,84% dan untuk titik acuan kiri sebesar 79,81%.

Nilai kesalahan paling tinggi untuk penghitungan kecepatan terdapat pada titik acuan kiri video 5_2.mp4 dengan nilai kesalahan relatif 97,8%. Nilai kesalahan paling rendah penghitungan kecepatan terdapat pada titik acuan kiri video 20_3.mp4 dengan nilai kesalahan relatif 1,3%.

Beberapa masalah pada metode deteksi dan penghitungan kecepatan ini, salah satu contoh adalah permasalahan pada segmentasi dapat mempengaruhi akurasi nilai yang dihasilkan oleh proses penghitungan karena titik acuan mobil bergantung pada piksel hasil segmentasi dan mobil yang tidak mencapai kecepatan aktual saat perekaman.

REFERENSI

- World Health Organization. (2010). *Injuries and Violence: The Facts*. Geneva: World Health Organization.
- Tourani, A., Shahbrami, A., Akoushideh, A., Khazae, S., & Suen, C. Y. (2019). Motion-based Vehicle Speed Measurement for Intelligent Transportation Systems. *I.J. Image, Graphics and Signal Processing*, 4, 42-54.
- Reddy, K. R., Priya, K. H., & Neelima, N. (2015). Object Detection and Tracking - A Survey. ICC-ICN.
- Walia, G. S., & Kapoor, R. (2016). Recent Advances on Multicue Object Tracking: A Survey. *Artif Intell Rev*.
- Firmansyah, M. A., & Supriana, I. (2015). Multiple Object Tracking dan Estimasi Posisi untuk Menunjang Sistem Keamanan Rumah. *Research Gate*.
- Luvizon, D. C., Nassu, B. T., & Minetto, R. (2017). A Video-Based System for Vehicle Speed Measurement in Urban Roadways. *IEEE Transactions on Intelligent Transportation Systems*, 18, 1393-1404.
- Goyal, K., & Singhai, J. (2018). Review of Background Subtraction Methods using Gaussian Mixture Model for Video Surveillance Systems. *Artif Intell Rev*, 50, 241-259.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *CVPR Computer Vision Foundation*, 4510-4520.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.
- Tissainayagam, P., & Suter, D. (2005). Object tracking in image sequences using point features. *Pattern Recognition - Elsevier*, 105-113.
- Xiang, L., & Xiaoqing, W. (2011). Motion Segmented Based on Edged Detection. *Procedia Engineering - Elsevier*, 74-78.
- Mirah, S. (2018). Pengenalan NIK pada E-KTP menggunakan Segmentasi Profil Proyeksi dan Ekstraksi Ciri menggunakan Invarian Momen Hu dan Intensity of Character. *Teknik Informatika Universitas Sanata Dharma*.
- Brownlee, J. (2019). *A Gentle Introduction to Batch Normalization for Deep Neural Network*.
- Pedomanti, D. (2018). Comparison of non-linear activation functions for deep neural networks on MNIST classification task.
- Singla, N. (2014). Motion Detection based on Frame Difference Method. *International Journal of Information & Computation Technology*, 4(15), 1559-1565.
- Migliore, D., Matteucci, M., & Naccari, M. (2006). A Reevaluation of Frame Difference in Fast and Robust Motion Detection. *VSSN*, 215-218.
- Affrudin, D. R. (2017). Penerapan Metode Frame Difference Pada Sistem Deteksi Gerakan Berbasis Android. *Jurnal Mahasiswa Teknik Informatika JATI*, 144-148.
- Yuha, R. A., Al Fiqri, M. D., Ashari, Pratama, R., & Harahap, M. (2019). Deteksi Gerakan pada Kamera CCTV dengan Algoritma Frame Difference dan Frame Substraction. *Seminar Nasional APTIKOM*, 503-511.
- Gupta, P., Singh, Y., & Gupta, M. (2014). Moving Object Detection Using Frame Difference, Background Substraction And SOBS For Video Surveillance Application. *System Modelling & Advancement in Research Trends (SMART)*, 151-156.
- Khan, S. D., Porta, F., Vizzari, G., & Bandini, S. (2014). Estimating Speeds of Pedestrians in Real-World Using Computer Vision. *Cellular Automata*, 526-535.
- Lugianti, I., Sari, J. Y., & Ningrum, I. P. (2019). Deteksi Kecepatan Kendaraan Bergerak Berbasis Video Menggunakan Metode Frame Difference. *Seminar Nasional Teknologi Terapan Berbasis Kearifan Lokal*, 324-332.
- Everingham, M. (2012). Visual Object Classes Challenge 2012 (VOC2012).
- Zhang, J., Cao, J., & Mao, B. (2016). Moving Object Detection based on Non-parametric methods and Frame Difference for Traceability Video Analysis. *Procedia Computer Science*, 91, 995-1000.