

SKRIPSI

**SISTEM DETEKSI KECEPATAN KENDARAAN SECARA
REAL-TIME DENGAN METODE GAUSSIAN MIXTURE
MODELS (GMM)**

Disusun dan diajukan oleh:

**MUHAMMAD IKHWAN RAMADHANI
D121 17 1512**



**PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2024**

LEMBAR PENGESAHAN SKRIPSI

SISTEM DETEKSI KECEPATAN KENDARAAN SECARA REAL-TIME DENGAN METODE GAUSSIAN MIXTURE MODELS (GMM)

Disusun dan diajukan oleh

Muhammad Ikhwan Ramadhani
D121 17 1512

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin Pada tanggal 22 Januari 2024 dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama,



Prof. Dr. Ir. Indrabayu S.T., M.T.,
M.Bus.Sys., IPM., ASEAN.Eng.
NIP 19750716 2002121 004

Pembimbing Pendamping,



Elly Warni S.T., M.T.
NIP 19820216 2008122 001

Ketua Program Studi,



Prof. Dr. Ir. Indrabayu S.T., M.T., M.Bus.Sys., IPM., ASEAN.Eng.
NIP 19750716 2002121 004

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini ;

Nama : Muhammad Ikhwan Ramadhani

NIM : D121171512

Program Studi : Teknik Informatika

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

SISTEM DETEKSI KECEPATAN KENDARAAN SECARA REAL-TIME DENGAN METODE GAUSSIAN MIXTURE MODELS (GMM)

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

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

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

Gowa, 19 Februari 2024

Yang Menyatakan



Muhammad Ikhwan Ramadhani

ABSTRAK

MUHAMMAD IKHWAN RAMADHANI. *Sistem Deteksi Kecepatan Kendaraan Secara Real-Time Dengan Metode Gaussian Mixture Models (GMM)* (dibimbing oleh Indrabayu dan Elly Warni)

Dalam era modern, pemantauan kecepatan kendaraan di jalan raya menjadi semakin penting untuk menjaga keselamatan dan ketertiban lalu lintas. Namun, pengawasan ini masih menjadi tantangan di berbagai kota, termasuk Kota Makassar. Meskipun telah terpasang sistem CCTV di sepanjang jalan, pemanfaatannya belum optimal dalam mengendalikan pelanggaran batas kecepatan. Ini menyebabkan risiko pelanggaran lalu lintas yang berbahaya dan mengancam keselamatan pengguna jalan serta ketertiban umum.

Penelitian ini bertujuan untuk mengatasi permasalahan tersebut dengan mengembangkan dan menguji sebuah sistem deteksi kecepatan kendaraan secara real-time menggunakan teknologi CCTV. Sistem ini dirancang untuk memantau kecepatan kendaraan secara akurat dan efisien, dengan harapan dapat meningkatkan pengawasan lalu lintas di Kota Makassar. Selain itu, penelitian ini juga bertujuan untuk mengevaluasi tingkat akurasi sistem dalam mendeteksi kecepatan kendaraan.

Penelitian ini memanfaatkan kumpulan data berupa 21 rekaman video yang terbagi dalam tujuh skenario pengambilan data yang berbeda, serta tiga variasi skenario kecepatan. yang kemudian dilakukan *background subtraction* dengan GMM, lalu dilanjut perhitungan kecepatan dengan cara menghitung perpindahan centroidnya menggunakan metode *euclidean distance*. Yang selanjutnya sistem akan dievaluasi menggunakan metode MAPE.

Hasil penelitian menunjukkan keberhasilan sistem dalam mendeteksi kecepatan kendaraan pada berbagai skenario kecepatan dan skenario jumlah objek. Evaluasi sistem mengungkapkan tingkat akurasi yang sangat baik, dengan MAPE gabungan sebesar 6.78%.

Kata Kunci: CCTV, Kecepatan kendaraan, GMM

ABSTRACT

MUHAMMAD IKHWAN RAMADHANI. *Real-Time Vehicle Speed Detection System Using Gaussian Mixture Models (GMM) Method* (supervised by Indrabayu dan Elly Warni)

In the modern era, monitoring vehicle speed on the road has become increasingly important to maintain traffic safety and order. However, monitoring is still a challenge in various cities, including Makassar City. Even though a CCTV system has been installed along the road, its use is not optimal in controlling speed limit violations. This causes the risk of dangerous traffic violations and threatens the safety of road users and public order.

This research aims to overcome this problem by developing and testing a real-time vehicle speed detection system using CCTV technology. This system is designed to monitor vehicle speed accurately and efficiently, with the hope of improving traffic control in Makassar City. Apart from that, this research also aims to evaluate the system's level of accuracy in detecting vehicle speed.

This research utilizes a dataset comprising 21 video recordings divided into seven distinct data collection scenarios, along with three variations of speed scenarios. Subsequently, background subtraction is performed using Gaussian Mixture Model (GMM), then continues with speed calculations by calculating the centroid displacement using the Euclidean distance method. Next, the system will be evaluated using the MAPE method.

The research results show the success of the system in detecting vehicle speed in various speed scenarios and number of objects scenarios. System evaluation revealed an excellent accuracy rate, with a combined MAPE of 6.78%.

Keywords: CCTV, Vehicle Speed, GMM

DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI.....	i
PERNYATAAN KEASLIAN.....	ii
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR	vi
DAFTAR TABEL.....	viii
DAFTAR SINGKATAN DAN ARTI SIMBOL	x
DAFTAR LAMPIRAN.....	xi
KATA PENGANTAR	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian/Perancangan.....	2
1.4 Manfaat Penelitian/Perancangan.....	2
1.5 Batasan Masalah.....	2
1.6 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Video Digital.....	4
2.2 Citra Digital.....	6
2.3 Visi Komputer.....	8
2.4 <i>Background Subtraction</i>	9
2.5 <i>Metode Gaussian Mixture Model</i>	11
2.6 <i>Median Filtering</i>	13
2.7 <i>Shadow Removal</i>	14
2.8 Morfologi Citra	15
2.9 <i>Contour</i>	17
2.10 <i>Tracking</i>	19
2.11 <i>Speed</i> Estimasi dengan kamera diatas.....	20
2.12 <i>Mean Absolute Percentage Error (MAPE)</i>	23
2.13 <i>Video Streaming Digital</i>	23
BAB 3 METODE PENELITIAN.....	28
3.1 Tahapan Penelitian.....	28
3.2 Waktu dan Lokasi Penelitian	30
3.3 Instrumen Penelitian.....	30
3.4 Teknik Pengambilan Data.....	31
3.5 Perancangan Sistem	34
BAB 4 HASIL DAN PEMBAHASAN.....	41
4.1 Hasil Penelitian	41
BAB 5 KESIMPULAN DAN SARAN.....	84
5.1 Kesimpulan	84
5.2 Saran.....	85
DAFTAR PUSTAKA	86

DAFTAR GAMBAR

Gambar 1 Struktur video digital.....	4
Gambar 2 Koordinasi Citra Digital (Putra, 2010).....	7
Gambar 3 Rangkaian proses utama pada Computer Vision.....	9
Gambar 4 Proses <i>background subtraction</i> untuk mendapatkan <i>foreground</i> image	10
Gambar 5 Diagram Alir Metode GMM	12
Gambar 6 Contoh <i>Erosion</i> pada Morfologi Citra	16
Gambar 7 Contoh <i>Dilation</i> pada Morfologi Citra.....	16
Gambar 8 Contoh Operasi <i>Opening</i> dan <i>Closing</i>	17
Gambar 9 Representasi Posisi Kamera di Atas.....	20
Gambar 10 Tendensi Kamera dan Tampilan Grid Hasil Rekaman dari Atas	22
Gambar 11 Alur diagram Video Streaming	25
Gambar 12 Alur Penelitian.....	28
Gambar 13 Lokasi penelitian pada ruas jalanan utama fakultas teknik unhas.....	30
Gambar 14 Deskripsi posisi kamera dan kendaraan yang dideteksi	31
Gambar 15 Tahapan Perhitungan Kecepatan	34
Gambar 16 Layout Pengambilan Video	35
Gambar 17 alur GMM secara singkat	36
Gambar 18 <i>Background subtraction</i> pada skenario 1	42
Gambar 19 <i>Background subtraction</i> pada skenario 2	42
Gambar 20 <i>Background subtraction</i> pada skenario 3	42
Gambar 21 <i>Background subtraction</i> pada skenario 4	42
Gambar 22 <i>Background subtraction</i> pada skenario 5	43
Gambar 23 <i>Background subtraction</i> pada skenario 6	43
Gambar 24 <i>Background subtraction</i> pada skenario 7	43
Gambar 25 <i>Smoothing</i> pada skenario 1.....	44
Gambar 26 <i>Smoothing</i> pada skenario 2.....	44
Gambar 27 <i>Smoothing</i> pada skenario 3.....	44
Gambar 28 <i>Smoothing</i> pada skenario 4.....	45
Gambar 29 <i>Smoothing</i> pada skenario 5.....	45
Gambar 30 <i>Smoothing</i> pada skenario 6.....	45
Gambar 31 <i>Smoothing</i> pada skenario 7.....	45
Gambar 32 <i>Shadow Removal</i> pada skenario 1	46
Gambar 33 <i>Shadow Removal</i> pada skenario 2	46
Gambar 34 <i>Shadow Removal</i> pada skenario 3	47
Gambar 35 <i>Shadow Removal</i> pada skenario 4	47
Gambar 36 <i>Shadow Removal</i> pada skenario 5	47
Gambar 37 <i>Shadow Removal</i> pada skenario 6	47
Gambar 38 <i>Shadow Removal</i> pada skenario 7	48
Gambar 39 <i>Fill Holes</i> pada skenario 1	48
Gambar 40 <i>Fill Holes</i> pada skenario 2.....	49
Gambar 41 <i>Fill Holes</i> pada skenario 3.....	49
Gambar 42 <i>Fill Holes</i> pada skenario 4.....	49
Gambar 43 <i>Fill Holes</i> pada skenario 5.....	49
Gambar 44 <i>Fill Holes</i> pada skenario 6.....	50

Gambar 45 <i>Fill Holes</i> pada skenario 7.....	50
Gambar 46 Hasil deteksi kontur pada skenario 1.....	51
Gambar 47 Hasil deteksi kontur pada skenario 2.....	51
Gambar 48 Hasil deteksi kontur pada skenario 3.....	51
Gambar 49 Hasil deteksi kontur pada skenario 4.....	52
Gambar 50 Hasil deteksi kontur pada skenario 5.....	52
Gambar 51 Hasil deteksi kontur pada skenario 6.....	52
Gambar 52 Hasil deteksi kontur pada skenario 7.....	53
Gambar 53 Hasil deteksi yang telah dibounding-box pada skenario 1.....	53
Gambar 54 Hasil deteksi yang telah dibounding-box pada skenario 2.....	53
Gambar 55 Hasil deteksi yang telah dibounding-box pada skenario 3.....	54
Gambar 56 Hasil deteksi yang telah dibounding-box pada skenario 4.....	54
Gambar 57 Hasil deteksi yang telah dibounding-box pada skenario 5.....	54
Gambar 58 Hasil deteksi yang telah dibounding-box pada skenario 6.....	55
Gambar 59 Hasil deteksi yang telah dibounding-box pada skenario 7.....	55
Gambar 60 hasil deteksi gagal	55
Gambar 61 Kecepatan kendaraan pada skenario 1 dengan kecepatan aktual 20 ..	59
Gambar 62 Kecepatan kendaraan pada skenario 1 dengan kecepatan aktual 30 ..	60
Gambar 63 Kecepatan kendaraan pada skenario 1 dengan kecepatan aktual 40 ..	61
Gambar 64 Kecepatan kendaraan pada skenario 2 dengan kecepatan aktual 20 ..	62
Gambar 65 Kecepatan kendaraan pada skenario 2 dengan kecepatan aktual 30 ..	63
Gambar 66 Kecepatan kendaraan pada skenario 2 dengan kecepatan aktual 40 ..	64
Gambar 67 Kecepatan kendaraan pada skenario 3 dengan kecepatan aktual 20 ..	66
Gambar 68 Kecepatan kendaraan pada skenario 3 dengan kecepatan aktual 30 ..	67
Gambar 69 Kecepatan kendaraan pada skenario 3 dengan kecepatan aktual 40 ..	68
Gambar 70 Kecepatan kendaraan pada skenario 4 dengan kecepatan aktual 20 ..	69
Gambar 71 Kecepatan kendaraan pada skenario 4 dengan kecepatan aktual 30 ..	70
Gambar 72 Kecepatan kendaraan pada skenario 4 dengan kecepatan aktual 40 ..	71
Gambar 73 Kecepatan kendaraan pada skenario 5 dengan kecepatan aktual 20 ..	73
Gambar 74 Kecepatan kendaraan pada skenario 5 dengan kecepatan aktual 30 ..	74
Gambar 75 Kecepatan kendaraan pada skenario 5 dengan kecepatan aktual 40 ..	75
Gambar 76 Kecepatan kendaraan pada skenario 6 dengan kecepatan aktual 20 ..	76
Gambar 77 Kecepatan kendaraan pada skenario 6 dengan kecepatan aktual 30 ..	77
Gambar 78 Kecepatan kendaraan pada skenario 6 dengan kecepatan aktual 40 ..	78
Gambar 79 Kecepatan kendaraan pada skenario 7 dengan kecepatan aktual 20 ..	79
Gambar 80 Kecepatan kendaraan pada skenario 7 dengan kecepatan aktual 30 ..	80

DAFTAR TABEL

Tabel 1 Kriteria nilai MAPE	23
Tabel 2 Skenario pengujian sistem	33
Tabel 3 Menunjukkan hasil Labelling dan <i>Tracking</i> pada skenario 1	56
Tabel 4 Menunjukkan hasil Labelling dan <i>Tracking</i> pada skenario 2	56
Tabel 5 Menunjukkan hasil Labelling dan <i>Tracking</i> pada skenario 3	56
Tabel 6 Menunjukkan hasil Labelling dan <i>Tracking</i> pada skenario 4	57
Tabel 7 Menunjukkan hasil Labelling dan <i>Tracking</i> pada skenario 5	57
Tabel 8 Menunjukkan hasil Labelling dan <i>Tracking</i> pada skenario 6	58
Tabel 9 Menunjukkan hasil <i>Labelling</i> dan <i>Tracking</i> pada skenario 7	58
Tabel 10 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 20	59
Tabel 11 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 30	60
Tabel 12 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 40	61
Tabel 13 Menunjukkan hasil MAPE pada skenario 1	62
Tabel 14 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 20	63
Tabel 15 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 30	64
Tabel 16 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 40	65
Tabel 17 Menunjukkan hasil MAPE pada skenario 2	65
Tabel 18 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 20	66
Tabel 19 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 30	67
Tabel 20 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 40	68
Tabel 21 Menunjukkan hasil MAPE pada skenario 3	69
Tabel 22 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 20	70
Tabel 23 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 30	71
Tabel 24 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 40	72
Tabel 25 Menunjukkan hasil MAPE pada skenario 4	72
Tabel 26 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 20	73
Tabel 27 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 30	74
Tabel 28 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 40	75
Tabel 29 Menunjukkan hasil MAPE pada skenario 5	76

Tabel 30 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 20	77
Tabel 31 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 30	77
Tabel 32 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 40	78
Tabel 33 Menunjukkan hasil MAPE pada skenario 6.....	79
Tabel 34 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 20	80
Tabel 35 Menunjukkan hasil deteksi kecepatan oleh sistem dengan kecepatan aktual 30	81
Tabel 36 Menunjukkan hasil MAPE pada skenario 7.....	81
Tabel 37 Menunjukkan hasil MAPE pada keseluruhan skenario	81
Tabel 38 Menunjukkan perbandingan hasil MAPE sebelum dan sesudah <i>resizing</i>	83

DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/Singkatan	Arti dan Keterangan
P	Bidang tegak lurus pandang di layar kamera
θ_1 / Tv	Sudut yang dibentuk oleh kemiringan kamera
θ_2	Sudut pandang yang tidak dicangkup oleh kamera
θ_3 / Tc	Sudut pandang yang dicangkup oleh kamera
H	Tinggi kamera dari permukaan jalan
D	Jarak horizontal antara kamera dan kendaraan
h	Tinggi kendaraan
L	Jarak nyata antara kamera dan kendaraan
d	<i>Euclidean distance</i>
V	<i>Velocity / Kecepatan</i>
k	Koefisien kalibrasi / parameter geometris
t	<i>time / waktu</i>
f	<i>focal length</i>
v	<i>vertical dimension of 35mm image format</i>
RGB	<i>Red Green Blue</i>
CCTV	<i>Closed Circuit Television</i>
ITS	<i>Intelligent Transport System</i>
fps	<i>frame per second</i>
MAPE	<i>Mean Absolute Percentage Error</i>
CV	<i>Computer Vision</i>
BS	<i>Background Subtraction</i>
GMM	<i>Gaussian Mixture Models</i>
ROI	<i>Region Of Interest</i>

DAFTAR LAMPIRAN

Lampiran 1 Lembar Perbaikan Skripsi	88
Lampiran 2 Source Code 1.....	89
Lampiran 3 Source Code 2.....	92

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa, yang telah memberikan rahmat dan karunia-Nya, sehingga dapat menyelesaikan tugas akhir dengan judul “Sistem Deteksi Kecepatan Kendaraan Secara real-time berbasis *network traffic camera* dengan metode *Gaussian Mixture Models (GMM)*” sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 di Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin.

Penulis menyadari bahwa penyusunan dan penulisan tugas akhir ini tidak dapat terselesaikan dengan baik tanpa adanya bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih banyak kepada:

1. Kedua orang tua penulis, Bapak Syafiuddin Hamid dan Ibu Hikmatul Azizah yang selalu memberikan dukungan, doa, dan semangat yang tiada hentinya, serta selalu sabar dalam mendidik penulis sejak kecil.
2. Bapak Prof. Dr. Ir. Indrabayu S.T., M.T.,M.Bus.Sys., IPM., ASEAN.Eng, selaku pembimbing I dan Bu Elly Warni S.T., M.T. selaku pembimbing II, yang senantiasa menyediakan waktu, tenaga, pikiran, dan perhatian yang luar biasa dalam mengarahkan penulis untuk menyelesaikan tugas akhir.
3. Bapak Prof. Dr. Ir. Indrabayu S.T., M.T.,M.Bus.Sys., IPM., ASEAN.Eng selaku ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.
4. Segenap keluarga *AIMP Research Group* Universitas Hasanuddin yang telah memberikan begitu banyak bantuan selama penelitian, pengambilan data dan diskusi progres penyusunan tugas akhir serta selalu memberikan motivasi.
5. Muhammad Fadhil, Muhammad Hidayat, Ahmad Reza, Muhammad Zulfahmi Sadrah, dan Fauzan Alif Anwar serta seluruh anggota *Recognizer* yang telah membantu penulis sejak awal perkuliahan dan selalu membantu dalam penyelesaian tugas akhir.
6. Segenap staf dan dosen Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin yang telah membantu kelancaran penyelesaian tugas akhir.
7. Serta pihak-pihak lain yang tidak disebutkan dan tanpa sadar telah menjadi inspirasi dan membantu penulis dalam menyelesaikan tugas akhir.

Penulis berharap semoga Tuhan Yang Maha Esa berkenan membalas segala kebaikan serta jasa dari semua pihak yang telah banyak membantu penulis dalam menyelesaikan tugas akhir ini. Penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna, oleh karena itu penulis mengharapkan segala bentuk saran serta masukan yang membangun dari berbagai pihak. Semoga tugas akhir ini dapat memberi manfaat bagi para pembaca dan semua pihak. Aamiin.

BAB I PENDAHULUAN

1.1 Latar Belakang

Saat ini, di berbagai kota telah dipasang CCTV pada setiap ruas jalan di Kota Makassar. Namun, petugas lalu lintas belum sepenuhnya memanfaatkan kegunaan CCTV. Petugas lalu lintas hanya dapat mengetahui kondisi pada suatu jalan, seperti kepadatan, kecelakaan dan lain-lain. Data mengenai kecepatan seringkali diabaikan. Padahal, Kecepatan kendaraan sangat berguna untuk diketahui, karena dapat ditentukan apakah kecepatan suatu kendaraan di atas hukum batas kecepatan yang berlaku (Andrew et al., 2017).

Adapun undang-undang yang mengatur batas kecepatan yakni tercantum pada pasal 21 UU No.22 Tahun 2009 yang berbunyi “setiap jalan memiliki batas kecepatan paling tinggi yang ditetapkan secara nasional.” Serta pada pasal 287 UU No.22 Tahun 2009 Tentang sanksi pelanggaran yang berbunyi “Setiap orang yang mengemudikan Kendaraan Bermotor di jalan yang melanggar aturan batas kecepatan paling tinggi atau paling rendah bisa kena pidana kurungan paling lama 2 bulan atau denda paling banyak Rp500.000,00-.”.

Maka perlu dilakukan upaya untuk menertibkan pengemudi melebihi batas maksimum. Penentuan batas kecepatan maksimum yang ditunjukkan dengan rambu lalu lintas adalah salah satu aturan efektif untuk menanggulangi kebut-kebutan. Namun tanpa ada petugas lalu-lintas yang berjaga, pelanggaran tetap akan terjadi. Maka deteksi kecepatan sangat penting untuk membantu pengawasan para pengguna lalu lintas (Fajriyah & Setiyono, 2016).

Saat ini, pengawasan pengguna lalu lintas sudah memakai CCTV di beberapa titik serta *Intelligent Transport System* (ITS) dan konsep *Smart City* memungkinkan pemerintah untuk menggunakan CCTV juga. Berhubungan dengan tiga hal tersebut, maka pemanfaatan hasil rekaman dan/atau *real-time video* untuk memperkirakan atau mendeteksi kecepatan pengemudi. Serta, sejauh ini upaya pemerintah kota Makassar dalam memanfaatkan fitur CCTV tersebut hanya terbatas, yakni pengenalan plat kendaraan.

Oleh karena itu, penulis mengusulkan suatu sistem yang dapat memprediksi serta menghitung kecepatan kendaraan secara akurat, agar dapat membantu pengawasan pengguna lalu lintas yang dimana melanggar aturan tentang maksimal kecepatan yang berlaku serta menambah fitur-fitur yang belum ada di CCTV Kota Makassar.

1.2 Rumusan Masalah

Adapun rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Bagaimana cara mengembangkan sistem deteksi kecepatan kendaraan dengan CCTV secara *real-time*?
2. Bagaimana tingkat akurasi sistem dalam mendeteksi kecepatan kendaraan dengan menggunakan metode MAPE ?

1.3 Tujuan Penelitian/Perancangan

Adapun tujuan dari penelitian ini ialah sebagai berikut:

1. Mengembangkan sistem deteksi kecepatan kendaraan dengan CCTV secara *real-time*.
2. Menunjukkan keakuratan sistem dalam mendeteksi kecepatan kendaraan dengan menggunakan metode MAPE.

1.4 Manfaat Penelitian/Perancangan

Adapun manfaat yang dapat diperoleh dari penelitian ini sebagai berikut:

1. Membantu penertib ketertiban untuk mengetahui dan/atau menertibkan pemakai kendaraan yang melanggar aturan yang berlaku.
2. Membantu Pemerintah Kota Makassar mewujudkan program *smart city*.
3. Memberikan informasi terhadap bidang ilmu pengetahuan yang relevan sesuai tema.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut

1. Data yang digunakan berupa video yang kemudian diekstrak menjadi beberapa *frame* gambar.

2. Video diambil dari ketinggian yang sudah ditentukan serta pengambilan video mencakup seluruh ruas jalanan 1 arah dan posisi kamera tidak bergerak.
3. Video diambil pada saat cuaca cerah serta pada siang hari dengan ketentuan objek tidak memiliki bayangan yang tebal ke samping objek tersebut.
4. Validasi hasil estimasi kecepatan dilakukan dengan uji empiris yaitu dengan menggunakan rekaman video kendaraan yang telah diatur kecepatannya yang selanjutnya diuji keakuratannya dengan menggunakan metode MAPE.

1.6 Sistematika Penulisan

Berikut adalah gambaran singkat mengenai isi tulisan secara keseluruhan:

BAB I PENDAHULUAN

Pada bab ini diuraikan mengenai latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah penelitian, metode penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini membahas landasan teori yang digunakan untuk menganalisis masalah yang akan diteliti serta hal-hal lain yang berhubungan dengan variabel-variabel data yang akan digunakan, seperti tinggi posisi kamera, kemiringan kamera dll.

BAB III METODOLOGI PENELITIAN

Pada bab ini berisi mengenai tahapan penelitian, instrumen penelitian, pengumpulan data, penerapan algoritma, teknik pengolahan data.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini berisi tentang sistem yang telah dibangun serta pembahasan dari hasil penelitian.

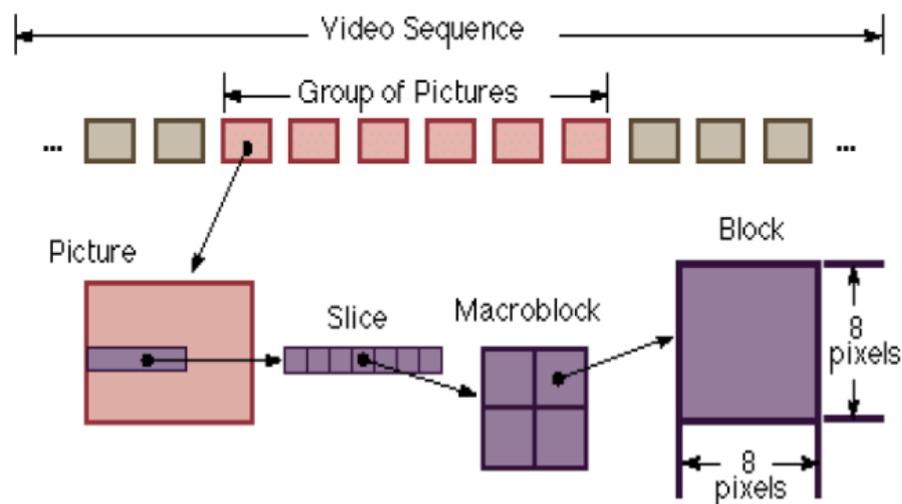
BAB V KESIMPULAN DAN SARAN

Pada 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 Video Digital

Video adalah bentuk penerapan teknologi untuk menangkap, merekam, memproses, menyimpan, dan merekonstruksi kumpulan citra yang saling berurutan. Alan C. Bovik dalam bukunya “*Handbook of Image and Video Processing*” menjelaskan bahwa video digital merupakan hasil sampling dan kuantisasi dari video analog. Secara mendasar, tidak ada perbedaan proses sampling dan kuantisasi antara citra digital dan video digital (Bovik, 2010).



Gambar 1 Struktur video digital

<http://www.hongik.edu/~sjpark/mpeg.html>

Dari Gambar 1, struktur yang menyusun video yaitu (Mukhopadhyay, 2011) :

1. *Video Sequence*, diawali dengan *sequence header*, berisi satu grup gambar atau lebih, diakhiri dengan kode *end-of-sequence*.
2. *Group of Pictures (GOP)*, *Header* dan serangkaian satu atau lebih gambar yang dimaksudkan untuk memungkinkan akses secara acak menjadi beruntun.
3. *Picture*, Unit pengkodean utama dari urutan video. Gambar terdiri dari tiga matriks segi empat yang mewakili nilai pencahayaan (Y) dan dua nilai krominasi (Cb dan Cr).

4. *Slice*, satu atau beberapa *macroblocks* bersebelahan. Urutan *macroblocks* dalam *slice* adalah dari kiri ke kanan dan dari atas ke bawah.
5. *Macroblocks*, komponen pencahayaan dan komponen krominasi sesuai dengan urutan blok pada aliran data. Video digital pada dasarnya adalah sekumpulan citra digital yang disusun secara teratur dan berurutan sehingga menyebabkan efek objek yang ada di dalam citra digital tersebut bergerak, karena adanya citra yang saling berurutan. Video digital terdiri dari beberapa *frame* yang ditampilkan dengan kecepatan tertentu. Setiap *frame* merupakan citra kontinu dan kecepatan untuk menampilkan citra-citra yang ada disebut sebagai *frame rate* dengan satuan fps (*frame per second*). Jika *frame rate* pada suatu video digital cukup tinggi, maka video akan terlihat semakin halus, dikarenakan banyaknya citra yang menyusun sebuah video tersebut. Kualitas suatu video sangat dipengaruhi oleh besarnya nilai karakteristik yang dimiliki oleh sebuah video digital, sehingga sensitivitas mata manusia terhadap video yang dilihat dipengaruhi oleh besarnya nilai-nilai karakteristik dari video itu sendiri.

Adapun karakteristik yang dimiliki oleh sebuah video digital adalah :

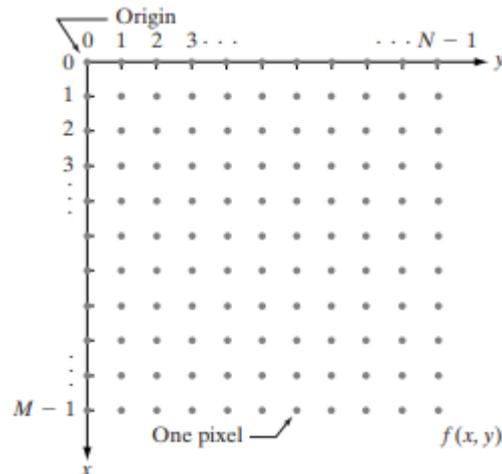
1. Resolusi. Resolusi atau dimensi *frame* merupakan ukuran sebuah *frame* yang dinyatakan dengan *pixel x pixel*. Semakin tinggi resolusi maka semakin baik tampilan video tersebut, namun resolusi yang tinggi membutuhkan jumlah bit yang besar, sehingga memiliki ukuran *file* yang besar.
2. Kedalaman Bit. Kedalaman bit akan menentukan jumlah bit yang digunakan untuk merepresentasikan tiap piksel pada sebuah *frame*. Sama halnya dengan resolusi, semakin besar kedalaman bit yang digunakan, maka semakin besar jumlah bit yang dibutuhkan.
3. *Frame rate*. *Frame rate* merupakan banyaknya jumlah *frame* yang bergerak tiap detik yang kerap dikenal sebagai *frame per second* (fps). Karakteristik ini berkaitan dengan kehalusan gerakan (*smoothness of motion*) sebuah objek di dalam video. Beberapa nilai standar *frame per second* (fps) yang umum digunakan adalah 30 fps dan 25 fps.

2.2 Citra Digital

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra digital merupakan kumpulan piksel (*picture element*) dengan suatu intensitas tertentu. Resolusi atau dimensi citra merupakan ukuran dari sebuah citra yang dinyatakan dengan *pixel's pixel*. Semakin tinggi resolusi suatu citra, maka akan semakin baik tampilan dari citra digital tersebut. Bagian terkecil dari suatu citra disebut dengan *pixel*. Sedangkan intensitas (kedalaman bit) dari masing-masing *pixel*, secara keseluruhan akan menggambarkan terang atau gelapnya citra digital tersebut.

Citra sebagai keluaran suatu sistem perekaman data dapat bersifat *optic* berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan.

Citra terbagi menjadi dua yaitu citra diam (*still image*) dan citra bergerak (*moving image*). Citra diam adalah citra tunggal yang tidak bergerak. Sedangkan citra bergerak adalah rangkaian citra diam yang ditampilkan secara beruntun (*sequentiali*) sehingga memberi kesan pada mata sebagai gambar yang bergerak. Pengolahan citra digital menunjuk pada pemrosesan gambar dua dimensi menggunakan komputer. Dalam konteks yang lebih luas, pengolahan citra digital, mengacu pada pemrosesan setiap data dua dimensi. Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu. Suatu citra dapat didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitude f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x, y, dan nilai amplitude f secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital.



Gambar 2 Koordinasi Citra Digital

Sumber : (Gonzalez & Woods, 2002)

Nilai pada suatu irisan antar baris dan kolom (pada posisi x,y) disebut dengan *picture elements*, *image elements*, *pels*, atau *pixels*. Istilah terakhir (*pixel*) paling sering digunakan pada citra digital. Citra digital tersusun atas titik-titik yaitu dapat berbentuk persegi panjang dan secara beraturan membentuk baris-baris dan kolom-kolom. Setiap titik memiliki koordinat dan dapat dinyatakan dalam bilangan bulat positif, yaitu 0 atau 1 bergantung pada sistem yang digunakan. Format nilai *pixel* sama dengan format citra keseluruhan. Pada kebanyakan sistem pencitraan, nilai ini dapat berupa bilangan bulat positif. Format citra digital yang banyak digunakan, yaitu:

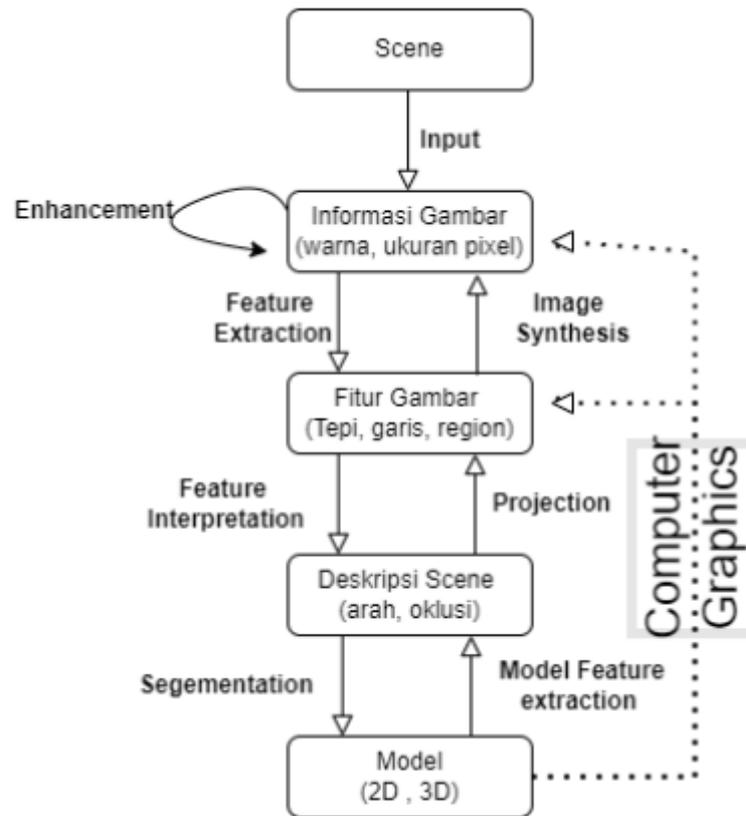
1. Citra biner (*Monokrom*). Citra monokrom atau citra hitam putih merupakan citra yang setiap pikselnya hanya mempunyai dua kemungkinan nilai, seperti *on* dan *off*, disimpan dalam matriks dengan nilai 0 (*off*) dan 1 (*on*).
2. Citra skala keabuan (*Grayscale*). Citra grayscale dikatakan format citra skala keabuan karena pada umumnya warna yang dipakai adalah warna hitam sebagai warna minimum dan warna putih sebagai warna maksimumnya, sehingga warna antara keduanya adalah abu-abu.
3. Citra berwarna, dimana citra warna terdiri atas 3 layer matriks, yaitu *R layer*, *G-layer*, *B-layer*. Sistem warna RGB (*Red Green Blue*) menggunakan sistem tampilan grafik kualitas tinggi (*High quality raster graphic*) yaitu mode 24 bit. Setiap komponen warna merah, hijau, biru masing-masing mendapatkan

alokasi 8 bit untuk menampilkan warna. Pada sistem warna RGB, setiap *pixel* akan dinyatakan dalam 3 parameter dan bukan nomor warna. Setiap warna mempunyai range nilai 00 (angka desimalnya adalah 0) dan f (angka desimalnya 255) atau mempunyai nilai derajat keabuan $256 = 28$. Dengan demikian, range warna yang digunakan adalah $(28) (28) (28) = 2$ (atau dikenal dengan istilah *true color* pada *Windows*). Nilai warna yang digunakan merupakan gabungan warna cahaya merah, hijau dan biru.

2.3 Visi Komputer

Computer vision (CV) atau visi komputer dikembangkan dengan memiliki dua tujuan ganda. Dari sudut ilmu sains biologi, visi komputer dimaksudkan untuk memahami model komputasi dari sistem visual manusia. Dari sudut ilmu rekayasa titik pandang, CV bertujuan membangun sistem otonom yang dapat melakukan beberapa tugas yang dapat dilakukan oleh sistem visual manusia (bahkan diharapkan dapat melampauinya dalam beberapa kasus). Banyaknya tantangan di bidang ini terkait dengan ekstraksi informasi 3D dan temporal dari dari 2D dengan waktu pengambilan yang variatif sebagaimana satu atau lebih kamera memperoleh informasi yang terangkum dalam adegan dinamis (Huang, 1996). Tentu saja, kedua tujuan tersebut berkaitan erat. Keberagaman mata manusia dan sifat sistem visual manusia sering memberikan inspirasi kepada para insinyur yang merancang sistem visi komputer. Sebaliknya, algoritma visi komputer dapat menawarkan wawasan tentang bagaimana sistem visual manusia bekerja dalam sekian baris kode.

Pemrosesan CV pada lingkup objek 3D terdiri dari banyak komponen yang melakukan fungsi tertentu dalam koordinasi dengan komponen lain. Gambar 3 merupakan blok rangkaian utama dan tipe informasi atau data yang terlibat dari proses tersebut. (Shirai, 2012)



Gambar 3 Rangkaian proses utama pada Computer Vision

Sumber : (Shirai, 2012)

Sebagaimana yang tersaji pada Gambar 3, Ketika sebuah adegan/*scene* masuk ke dalam program, informasi dasar dari gambar tersebut akan diekstraksi baik komponen warna pada gambar, ukuran tiap objek dalam *pixel* serta informasi *pixel* keseluruhan adegan itu sendiri. Informasi yang diekstraksi bisa berupa raw gambar yang original, maupun *scene* yang telah mengalami beberapa *enhancement* baik itu kontras, saturasi, dll. Setelah itu *scene* tersebut akan diinterpelasi untuk mendapatkan data tepi, garis, region, dan data apapun yang berkaitan dengan pembentukan dasar tiap objek pada *scene* tersebut. Langkah terakhir merupakan segmentasi untuk mendapatkan deskripsi mengenai *scene* tersebut yang selanjutnya akan terbentuk matriks model 2D maupun 3D dari satu *scene* tersebut.

2.4 Background Subtraction

Background subtraction (BS) digunakan untuk mendapatkan objek yang bergerak pada serangkaian image. Objek yang bergerak dapat diidentifikasi dengan

melakukan pengurangan antara *frame* pada waktu t dengan *background model* (model latar belakang). Kemudian nilai citra hasil pengurangan tersebut dibandingkan dengan nilai ambang batas (*threshold*) sesuai dengan metode yang dipakai untuk membangun *foreground image*.

Secara matematis proses tersebut dapat ditulis sebagai berikut:

$$|I(x, y, t) - B(x, y, t)| = d(x, y, t) \quad (1)$$

$$f(x, y, t) = \begin{cases} 1 & \text{jika } d(x, y, t) \geq \tau \\ 0 & \text{jika } d(x, y, t) < \tau \end{cases} \quad (2)$$

Dimana:

$I(x, y, t)$ adalah citra pada waktu t ,

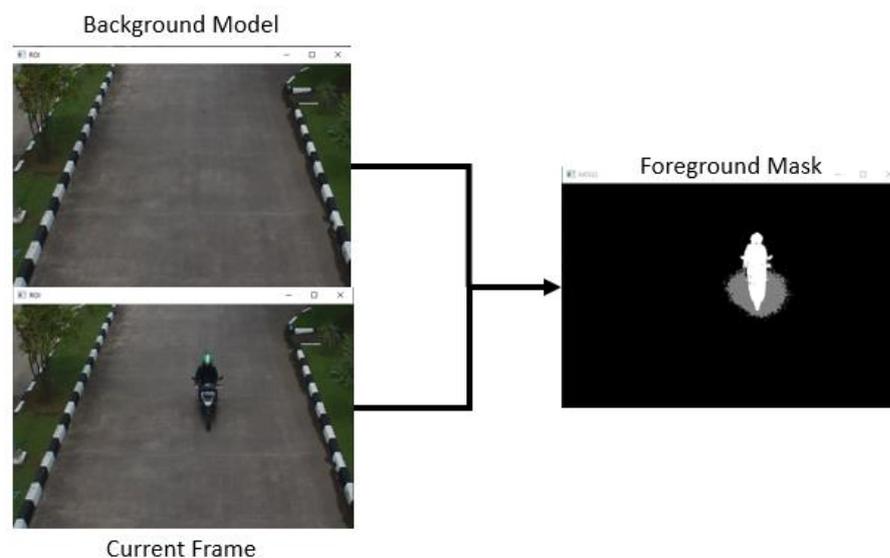
$B(x, y, t)$ adalah *background model* pada waktu t ,

$d(x, y, t)$ adalah citra hasil pengurangan I ,

B pada waktu t , $f(x, y, t)$ adalah *foreground image* (gerak yang dideteksi),

τ adalah *threshold*.

Ada berbagai macam metode untuk menginisialisasi *background model*. Salah satunya adalah metode *Gaussian Mixture Model* atau GMM (Amaluddin et al., 2015). Adapun contoh dari background subtraction oleh GMM ditunjukkan oleh Gambar 4.



Gambar 4 Proses *background subtraction* untuk mendapatkan *foreground image*

2.5 Metode *Gaussian Mixture Model*

Gaussian Mixture Model (GMM) adalah salah satu metode dari *Background Subtraction*. GMM merupakan tipe *density* model yang terdiri dari komponen fungsi-fungsi gaussian. Komponen fungsi tersebut terdiri dari *weight* yang berbeda untuk menghasilkan multi model *density*. Model-model GMM terbentuk dari data warna piksel berdasarkan waktu. Hasil model tersebut akan menjadi 2 bagian, yaitu model yang mencerminkan *background* dan model *non-background*.

Jumlah model GMM yang digunakan mempengaruhi jumlah model *background*. Semakin besar jumlah model GMM yang dipakai semakin banyak model *background* yang dimiliki suatu piksel. GMM memproses tiap piksel pada citra, baik citra berupa skalar (citra *grayscale*) maupun vektor (citra berwarna). Prosedur dari metode *Gaussian Mixture Model* disajikan pada Gambar 5 Diagram Alir Metode GMM.

Terdapat beberapa tahapan dalam pemilihan distribusi yang mencerminkan *background* (Amaluddin et al., 2015). Tahapan-tahapan tersebut adalah:

1. Pencocokan input terhadap distribusi

Pada tahap ini input dicocokkan dengan semua distribusi sampai ditemukan distribusi yang paling cocok. Suatu piksel dikatakan masuk dalam suatu distribusi jika nilai piksel tersebut masuk dalam jarak 2.5 standar deviasi dari sebuah distribusi. Untuk pencocokan input digunakan persamaan (3).

$$\mu_{i,j,k} - 2.5 * \sigma_{i,j,k} < X_{i,j,t} < \mu_{i,j,k} + 2.5 * \sigma_{i,j,k} \quad (3)$$

Dimana:

X_t adalah nilai intensitas dari suatu piksel (i,j) pada *frame* ke- t ,

μ_k adalah nilai *mean* pada piksel (i,j) dari *gaussian* ke- k ,

σ_k sebagai standar deviasi pada piksel (i,j) dari *gaussian* ke- k .

2. Update parameter

Pada tahap ini dilakukan update terhadap nilai dari parameter-parameter GMM yang nantinya digunakan untuk mengolah input selanjutnya. Nilai yang di-*update* terdiri dari *weight*, *means*, dan *standard deviation*. Nilai *weight* di-*update* menggunakan persamaan (4). Nilai *means* di-*update* menggunakan persamaan (6) dengan ρ diberikan pada persamaan (5). Nilai *standard deviation* di-*update*

menggunakan Persamaan (7). Setiap persamaan tersebut berlaku pada setiap piksel (i,j) .

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \quad (4)$$

$$\rho = \alpha/(\omega_{k,t}) \quad (5)$$

$$\mu_{k,t} = (1 - \rho)\mu_{k,t-1} + \rho X_t \quad (6)$$

$$\sigma_{k,t}^2 = (1 - \rho)\sigma_{k,t-1}^2 + \rho(X_t - \mu_{k,t})(X_t - \mu_{k,t}) \quad (7)$$

Dimana:

$\omega_{k,t}$ adalah bobot dari *gaussian* ke-k pada *frame* t,

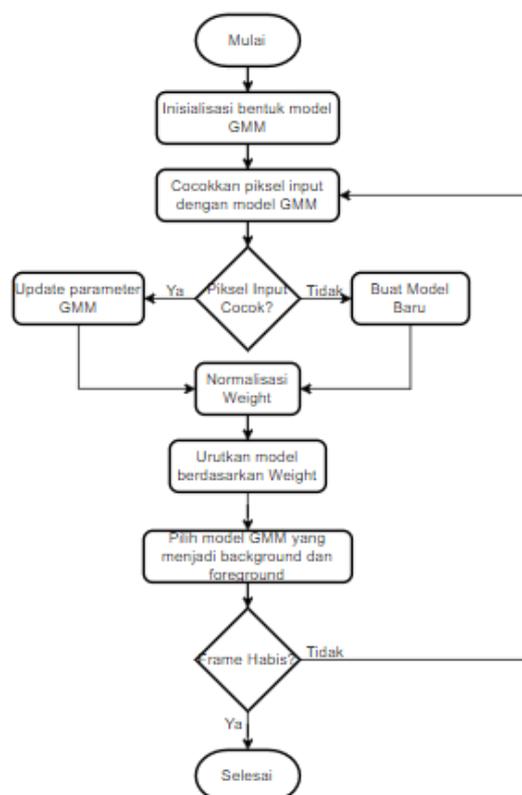
$\mu_{k,t}$ adalah *mean* dari *gaussian* ke-k pada *frame* t,

$\sigma_{k,t}$ adalah standar deviasi dari *gaussian* ke-k pada *frame* ke-t,

α adalah *learning rate*, dan nilai M_k ,

t adalah 1 untuk model yang cocok dan 0 untuk model yang tidak cocok

Setelah nilai weight di-update dilakukan normalisasi sehingga total bobot dari semua distribusi tepat 1. Sementara *means* dan standar deviasi di- update hanya jika ada nilai piksel yang cocok dengan distribusi tersebut (Wicaksono, 2017).



Gambar 5 Diagram Alir Metode GMM

Sumber : (Wicaksono, 2017)

3. Pemilihan Distribusi *Background*

Pada tahap ini dipilih model-model yang mencerminkan *background*. Pertama model-model diurutkan berdasarkan ω σ^2 sehingga distribusi yang paling mencerminkan *background* tetap di atas dan yang tidak mencerminkan *background* ada di bawah yang nantinya digantikan oleh distribusi yang lain. Untuk memilih B distribusi pertama yang dijadikan distribusi *background* digunakan persamaan (8), dengan T adalah nilai ambang batas yang telah ditentukan sebelumnya.

$$B = \operatorname{argmin}_b(\sum w_k > T) \quad (8)$$

2.6 Median Filtering

Median adalah nilai tengah dari kumpulan data. Untuk mencari *median* dari kumpulan data yang ganjil maka persamaan (9) menunjukkan cara mencarinya.

$$x = \frac{n+1}{2} \quad (9)$$

dimana:

x = Nilai baru *median*

n = Jumlah data

Untuk median filtering ini, data yang digunakan untuk menghitung median terdiri dari kumpulan data yang ganjil. Hal ini disebabkan dengan jumlah data yang ganjil maka piksel yang akan diproses dapat berada ditengah. Pada *median filtering* digunakan matrik berdimensi N X N. Dari matrik tersebut, kemudian data yang ada diurutkan dan dimasukkan dalam sebuah matrik berukuran 1X (N X N). Hal ini berguna untuk mempermudah menemukan median dari kumpulan data yang telah urut tersebut.

Sebagai contoh Gambar 6 sebagai suatu matrik berdimesi 3X3 yang berisi piksel utama dan piksel-piksel disekitarnya :

9	5	5
3	8	5
2	1	4

Gambar 6 Matrik untuk *median filtering*

Sumber: (Yuwono, 2015)

Matrik diatas harus diurutkan terlebih dahulu dan dimasukkan dalam sebuah matrik yang berukuran 1X (3X3) atau 1X9.

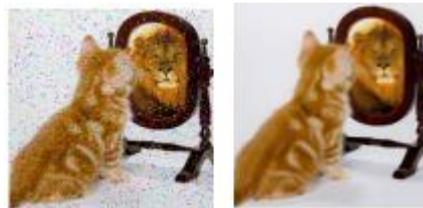
1	2	3	4	5	5	5	8	9
---	---	---	---	---	---	---	---	---

Gambar 7 Matrik untuk median filtering setelah diurutkan

Sumber:(Yuwono, 2015)

Dari Gambar 7 dapat dicari nilai piksel yang baru dengan menggunakan perhitungan median, maka nilai mediannya adalah $x = 5$. Nilai 5 ini akan menggantikan nilai 8 sehingga piksel utamanya akan memiliki warna yang berbeda dengan sebelumnya.(Yuwono, 2015)

Adapun contoh median filtering ditunjukkan oleh Gambar 8.



Gambar 8 Gambar asli (kiri), Gambar hasil median filtering (kanan)

Sumber: (Yuwono, 2015)

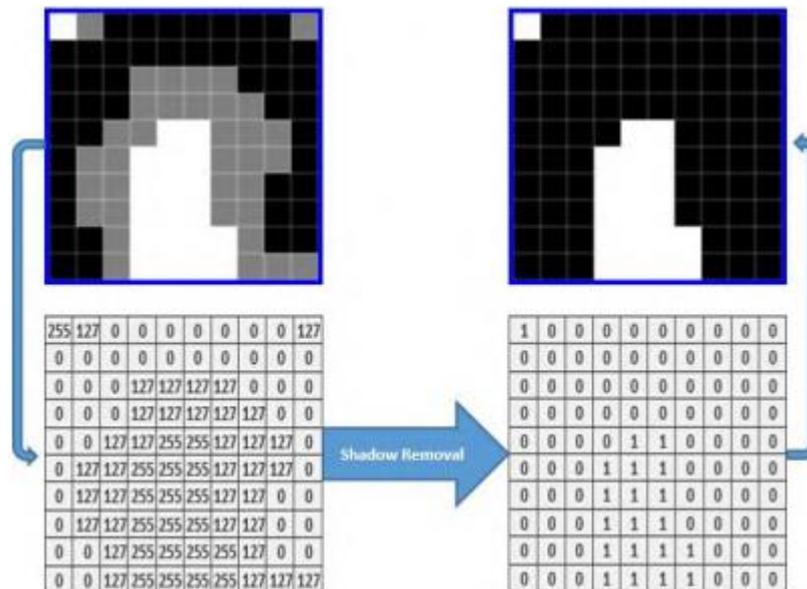
2.7 Shadow Removal

Shadow removal adalah proses dimana bayangan yang terdeteksi di foreground akan dihilangkan. Pada shadow removal output dari GMM yang berupa citra grayscale yang hanya memiliki 3 jenis intensitas yaitu 0 untuk obyek diam, 127 untuk obyek bayangan, dan 255 untuk obyek bergerak, akan diubah menjadi hanya 2 jenis intensitas atau biasa disebut biner yaitu 0 untuk obyek diam dan bayangan, dan 1 untuk obyek bergerak. Shadow removal adalah proses dimana nilai-nilai tersebut dipetakan ke angka biner yaitu 0 dan 1 seperti yang dijelaskan pada persamaan (10)

$$I_{biner}(x,y) = \begin{cases} 0, & I_{gray}(x,y) = 0 \text{ atau } I_{gray}(x,y) = 127 \\ 1, & I_{gray}(x,y) = 256 \end{cases} \quad (10)$$

Proses *shadow removal* berlaku pada seluruh piksel dari *image*. Proses *shadow removal* akan mempermudah proses dari deteksi obyek bergerak karena hasil dari

proses ini adalah *image biner* yang hanya mempunyai dua nilai, yaitu 0 atau hitam yang merepresentasikan *background* dan 1 atau putih yang merepresentasikan foreground atau obyek bergerak. contoh proses dari *shadow removal* dijelaskan pada Gambar 9.(Putra, 2016)



Gambar 9 Proses *shadow removal*

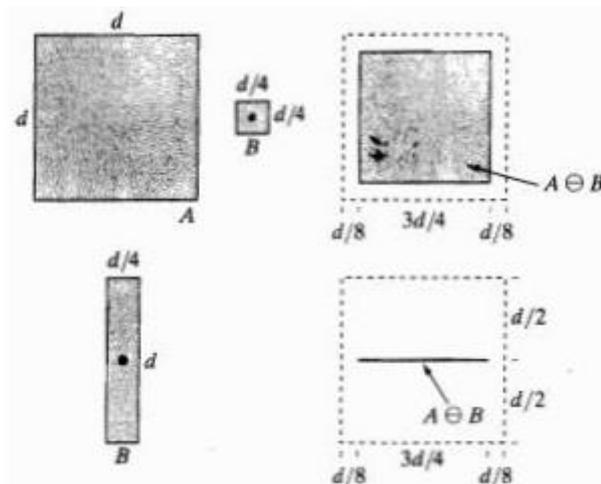
Sumber:(Putra, 2016)

2.8 Morfologi Citra

Morfologi dalam citra digital adalah suatu *tool* untuk ekstraksi komponen image yang berguna dalam representasi dan deskripsi dari bentuk daerah (*region shape*) dengan *structuring element* (SE) untuk menentukan *properties of interest* dari *image*. Di dalam morfologi, terdapat dua operasi fundamental yaitu *erosion* dan *dilation*. Secara matematis, dengan A dan B adalah himpunan pada Z^2 , *erosion* dari A oleh B, dinotasikan dengan $A \ominus B$ didefinisikan oleh persamaan (11).

$$A \ominus B = \{z | (B)z \subseteq A\} \quad (11)$$

Dengan kata lain, *erosion* dari A oleh B adalah himpunan semua titik z sedemikian hingga semua titik z pada B termuat didalam A. Dalam hal ini, B diasumsikan sebagai *structuring element*. *Erosion* berguna untuk menyempitkan atau menipiskan objek pada *image biner*. Contoh *erosion* dapat dilihat pada Gambar 10.



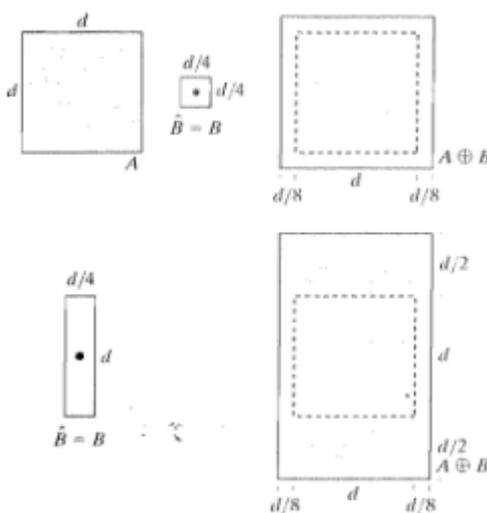
Gambar 10 Contoh *Erosion* pada Morfologi Citra

Sumber : (Gonzalez & Woods, 2002)

Dengan asumsi yang sama dengan yang digunakan oleh persamaan (11). *Dilation* dari A oleh B, dinotasikan dengan $A \oplus B$, didefinisikan oleh Persamaan berikut.

$$A \oplus B = \{z|(B)z \cap A \neq \emptyset\} \tag{12}$$

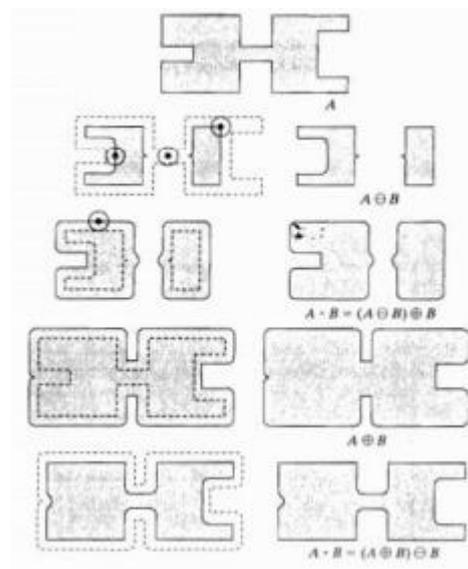
Dilation berguna untuk memperluas atau menebalkan objek pada image biner. Contoh dilation dapat dilihat pada Gambar 11



Gambar 11 Contoh *Dilation* pada Morfologi Citra

Sumber:(Gonzalez & Woods, 2002)

Selain dua operasi fundamental *erosion* dan *dilation*, terdapat dua operasi lain yang merupakan kombinasi dari dua operasi fundamental tersebut, yaitu *opening* dan *closing*. Operasi *opening* adalah operasi yang melakukan proses *erosion* terlebih dahulu kemudian hasilnya akan dilakukan *dilation*. Sedangkan operasi *closing* adalah operasi yang melakukan proses *dilation* terlebih dahulu kemudian hasilnya akan dilakukan *erosion*. Operasi *opening* secara umum digunakan untuk memperhalus kontur objek, menghilangkan gap tipis dan tonjolan pada objek. Sedangkan operasi *closing* digunakan untuk menghubungkan gap tipis, menghilangkan lubang kecil (*small holes*), dan mengisi gap pada kontur objek. Contoh operasi *opening* dan *closing* dapat dilihat pada Gambar 12. (Wicaksono, 2017)



Gambar 12 Contoh Operasi *Opening* dan *Closing*

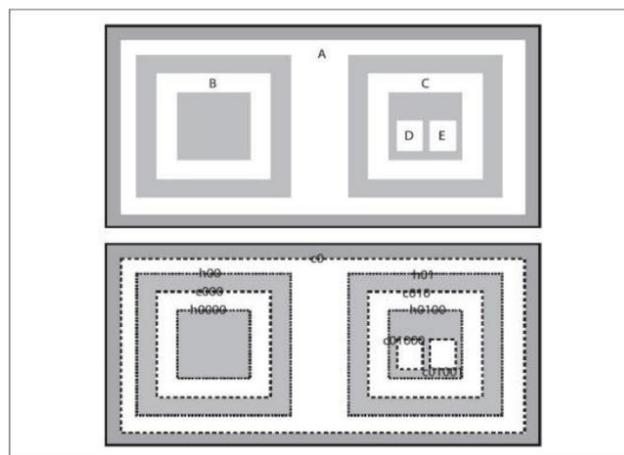
Sumber:(Gonzalez & Woods, 2002)

2.9 Contour

Kontur (*contour*) adalah suatu kumpulan *point* atau titik yang didapatkan dari suatu komputasi atau perhitungan yang mewakili bentuk dari suatu batas objek pada suatu citra yang nantinya dihubungkan dengan garis dari setiap titik atau *point* yang telah didapat, sehingga menghasilkan kontur dari objek tersebut. Kontur biasa dipakai untuk melihat garis-garis batas dari suatu objek, mirip dengan tepi (*edge*) hanya saja kontur lebih detil, tegas dan jelas. Misalkan melakukan pencarian kontur terhadap suatu citra untuk melihat apakah ada kontur yang berbentuk, seperti

sebuah mobil dalam suatu citra. Kontur didalam openCV membutuhkan suatu tempat *memory* khusus untuk menampung *sequence* yang berisikan titik atau *point-point* yang ditemukan. Maka di dalam openCV dibuat suatu struktur data yang merupakan *linkedlist* yang *sequential* tersendiri untuk menampung titik-titik kontur tersebut yang disebut dengan *sequence* atau disingkat **cvSeq** (*computer vision Sequential*).

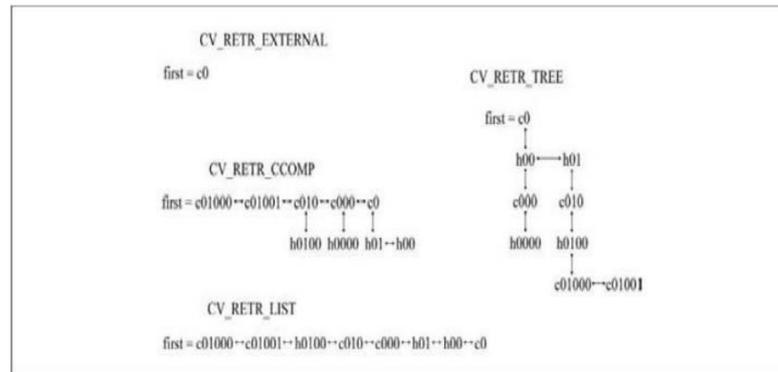
Dalam pembuatan kontur diperlukan rutin untuk melakukan proses pencarian kontur. Citra yang dipakai adalah citra hasil konversi menjadi *grayscale* dan dikenakan *threshold* sehingga menghasilkan citra *bi-level*. Dalam Gambar 13 dapat terlihat ilustrasi dari pencarian kontur. Terdapat gambar yang memiliki daerah putih dan abu-abu (dengan tingkat ketebalan yang berbeda) dan merupakan *background*, daerah putih diberi label A sampai E dengan background gelap. Di bagian gambar paling bawah terdapat hasil dari rutin pencarian kontur dan diberi label cX dan hX (X merupakan nilai biner iterator). Untuk label cX merupakan kontur dari setiap objek yang ada pada gambar sedangkan label hX merupakan hole atau lubang yang nantinya merupakan bagian yang dilingkari oleh kontur, yang didalam gambar kontur di perlihatkan sebagai garis yang putus-putus.(Gibran, 2012)



Gambar 13 Contoh Tes Pencarian Kontur Pada Suatu Citra

Sumber:(Gibran, 2012)

Dalam pencarian kontur ada 4 metode yang dapat digunakan dalam pengambilan kontur pada suatu citra yaitu *Retrieval External*, *Retrieval Components*, *Retrieval List*, dan *Retrieval Tree*.



Gambar 14 Bentuk Tipe Contour

Sumber : (Gibran, 2012)

1. *Retrieval External* adalah Pengambilan kontur yang didasarkan hanya pada kontur-kontur yang terlihat jelas dan berada di bagian luar objek yang ada dalam citra.
2. *Retrieval List* adalah Pengambilan seluruh kontur dan menyimpannya di dalam list. Tersimpan sebagai linkedlist.
3. *Retrieval Components* adalah Pengambilan seluruh kontur dan menyimpannya pada dua level hirarki dimana hirarki paling atas/ *top level boundaries* (kontur yang paling luar yang melingkari *hole*) adalah *external boundaries* dari komponen sedangkan level kedua adalah *hole boundaries*.
4. *Retrieval Tree* adalah Pengambilan seluruh kontur dan merekonstruksikannya ke dalam hirarki (*full hierarchy*) menjadi kontur yang bersarang

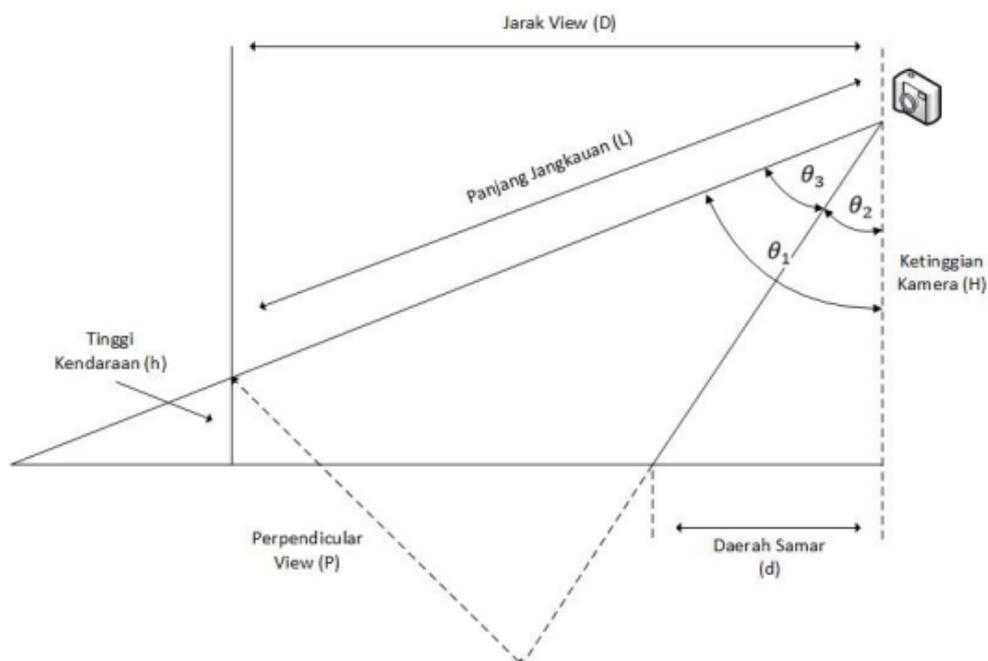
2.10 Tracking

Proses mencari objek bergerak dalam urutan *frame* yang dikenal sebagai pelacakan (*tracking*). Pelacakan ini dapat dilakukan dengan menggunakan ekstraksi ciri benda dan mendeteksi objek/benda bergerak di urutan *frame*. Dengan menggunakan nilai posisi objek di setiap *frame*, bisa digunakan untuk menghitung posisi dan kecepatan objek bergerak (Yilmaz et al., 2006). Tujuan dari proses *tracking* adalah untuk mengetahui objek yang sama pada urutan *frame*.

2.11 Speed Estimasi dengan kamera diatas

Kecepatan kendaraan diperoleh dari *frame* hasil deteksi *foreground*, yaitu dengan menentukan posisi kendaraan pada setiap *frame*. Jadi perlu ditentukan *bounding box* dan *centroid* dari hasil *foreground* yang telah dilakukan sebelumnya. *Centroid* disini sangat penting untuk mengetahui jarak kendaraan yang bergerak dalam *frame* yang berurutan. Pada *frame* berurutan, setelah dideteksi objek kendaraan bergerak dan memberikan *bounding box* maka harus ditentukan posisi objek pada *frame* ke t dan $t + 1$ dengan proses *tracking*.

Pada penelitian ini, digunakan posisi kamera di atas yang dapat direpresentasikan pada Gambar 15.



Gambar 15 Representasi Posisi Kamera di Atas

Sumber:(Wicaksono, 2017)

Seperti yang ditunjukkan pada Gambar 15, kamera diatur pada ketinggian H diatas permukaan jalan dengan sumbu kemiringan optiknya yaitu θ_1 dari jalan. Hubungan antara sudut lensa kamera dan domain dari kamera dapat ditentukan dengan menggunakan persamaan geometris. Dari Gambar 15, persamaan yang sesuai dengan bidang tegak lurus bidang dapat ditulis sebagai berikut:

$$P = 2L \tan (\theta_3/2) \quad (13)$$

$$L = \sqrt{(H - h)^2 + D^2} \quad (14)$$

$$P = 2 \tan (\theta_3/2) \sqrt{(H - h)^2 + D^2} \quad (15)$$

dimana:

P adalah bidang tegak lurus pandang di layar kamera;

θ_3 adalah sudut pandang yang dicakup oleh kamera,

H adalah tinggi kamera dari permukaan jalan,

D adalah jarak horizontal antara kamera dan kendaraan,

h adalah tinggi dari kendaraan,

L adalah jarak nyata antara kamera dan kendaraan.

Jika $\theta_1 \rightarrow \angle 90^\circ$ kemudian $L \rightarrow D$, maka persamaan diatas dapat disederhanakan menjadi:

$$P = 2D \tan (\theta_3/2) \quad (16)$$

Selain itu, θ_1 bisa didapatkan jika diasumsikan tidak ada kendaraan yang lewat, dengan kata lain $h \rightarrow 0$, maka dapat ditulis :

$$\theta_1 = \arctan (D/H) \quad (17)$$

dan sudut untuk daerah yang samar adalah :

$$\theta_2 = \arctan (d/H) \quad (18)$$

, d adalah daerah yang samar seperti yang ditunjukkan pada Gambar 15.

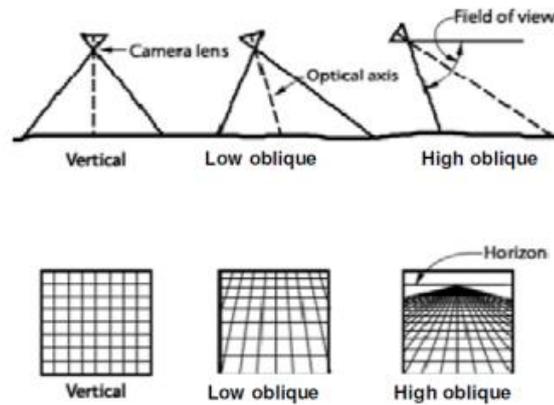
Selain itu juga, diketahui bahwa :

$$\theta_3 = \theta_1 - \theta_2 \text{ sehingga } \theta_2 = \theta_1 - \theta_3. \quad (19)$$

Daerah samar d bisa diperoleh dengan :

$$d = H \tan(\theta_1 - \theta_3) = H \tan \theta_2 \quad (20)$$

Kecenderungan dari kamera dengan posisi seperti Gambar 15 akan menghasilkan gambar dari tiga jenis area seperti yang ditunjukkan pada Gambar 16 yang bagian atas. Sedangkan Gambar 16 yang bagian bawah merupakan grid bagian garis yang terlihat pada berbagai jenis sudut kamera (Karim & Dehghani, 2010).



Gambar 16 Tendensi Kamera dan Tampilan Grid Hasil Rekaman dari Atas

Sumber:(Karim & Dehghani, 2010)

Untuk mengetahui jarak yang ditempuh dalam piksel, dapat dimisalkan koordinat suatu objek seperti berikut:

$$ct(a, b) \text{ dan } ct + 1(c, d)$$

Dimana ct dan $ct+1$ adalah posisi titik centroid pada *frame* t dan $t + 1$ untuk satu objek yaitu dengan koordinat (a, b) dan koordinat (c, d) .

Perbedaan jarak jauh untuk kendaraan dengan *Euclidean Distance* (Gokule & Kulkarni, 2014) yaitu:

$$d = \sqrt{(a - c)^2 + (b - d)^2} \quad (21)$$

Sedangkan, untuk kecepatan (speed) dapat dihitung dengan:

$$V = k.d/t \quad (22)$$

$$k = \frac{\text{actual_height}}{\text{image_height}} \quad (23)$$

$$t = 1/fps \quad (24)$$

Dengan k adalah koefisien kalibrasi, t adalah waktu antar 2 *frame* yang berurutan.

2.12 Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) adalah nilai rata – rata perbedaan absolut yang ada diantara nilai dari prediksi dan nilai realisasi yang disebutkan sebagai hasil persenan dari nilai realisasi. Penggunaan Mean Absolute Percentage Error (MAPE) pada evaluasi dari hasil peramalan dapat melihat tingkat akurasi terhadap angka peramalan dan angka realisasi. Nilai Mean Absolute Percentage Error (MAPE) dapat dihitung dengan menggunakan persamaan (25) berikut:

$$MAPE = \frac{100}{n} \sum \frac{|At-Ft|}{At} \quad (25)$$

dimana :

At= nilai aktual pada data t

Ft = nilai peramalan pada data t

n= jumlah periode data

Penggunaan MAPE pada evaluasi hasil prediksi dapat menghindari pengukuran akurasi terhadap besarnya nilai aktual dan nilai prediksi. Kriteria nilai MAPE ditunjukkan pada Tabel 1. (Putro et al., 2018)

Tabel 1 Kriteria nilai MAPE

Nilai MAPE	Kriteria
<10%	Sangat Baik
10%-20%	Baik
20%-50%	Cukup
>50%	Buruk

2.13 Video Streaming Digital

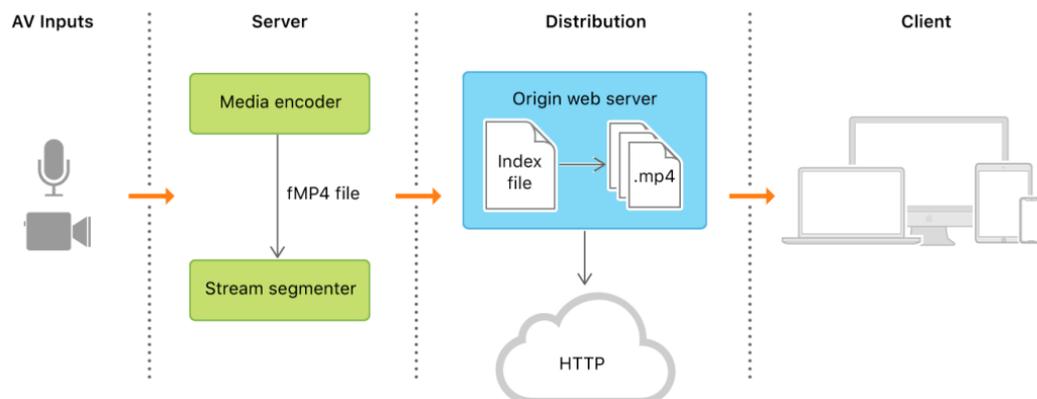
Video merupakan suatu media yang sangat penting untuk komunikasi dan hiburan selama puluhan tahun ini. Pada pertama kalinya, video diolah dan ditransmisikan dalam bentuk analog. Perkembangan di bidang komputer telah

membantu terbentuknya video dalam bentuk digital. Video digital merupakan representasi elektronik dari gambar visual bergerak dalam bentuk data digital yang dikodekan. Berbeda dengan video analog yang merepresentasikan gambar visual bergerak dengan sinyal analog. Video digital terdiri dari serangkaian gambar digital yang ditampilkan secara beruntun dan berurutan. Video digital juga dapat diartikan sebagai teknologi pengiriman sinyal elektronik dari suatu gambar yang bergerak. Aplikasi umum dari sinyal video digital yaitu seperti televisi, namun ia bisa juga digunakan dalam aplikasi lain di dalam bidang teknik, saintifik, produksi dan juga keamanan. Berdasarkan bahasa, kata video ini berasal dari kata latin. Salah satu penerapan video digital yang digunakan dalam transmisi data adalah video streaming.

Video Streaming adalah teknologi pengiriman data, video atau audio dalam bentuk yang telah dikompresi melalui jaringan internet yang ditampilkan oleh suatu *player* secara *realtime*. Pengguna memerlukan *player* yang merupakan aplikasi khusus untuk melakukan dekompresi dan mengirimkan data berupa video ke tampilan layar monitor dan data berupa suara ke *speaker*. Sebuah *player* dapat berupa suatu bagian dari *browser* atau sebuah perangkat lunak. Inti dari *streaming* adalah membagi data dan *encoding*, kemudian mengirimkannya melalui jaringan dan pada saat data sampai pada pengguna maka akan dilakukan *decoding* serta pembacaan data. Ciri-ciri aplikasi *streaming* yaitu distribusi audio, video, dan multimedia pada jaringan secara *realtime* atau *on demand*. Setelah *ter-streaming*, *transfer* media data digital dari *server* dan diterima oleh pengguna sebagai *realtime* stream simultan sehingga pengguna tidak perlu menunggu keseluruhan data di-download karena server mengirimkan data yang diperlukan setiap selang waktu tertentu. Hal ini memungkinkan pengguna untuk menjalankan *file content* seketika dengan periode *buffer* pendek.

Ada beberapa video *streaming* antara lain *webcast*, dimana tayangan yang ditampilkan merupakan siaran langsung (*live*) dan *Video on Demand* (VoD), dimana tayangan yang ditampilkan sudah terlebih dahulu disimpan dalam *server*. Faktor-faktor yang mempengaruhi distribusi *video streaming* melalui jaringan antara lain: besarnya *bandwidth*, waktu tunda (*delay*), *lost packet*, dan juga teknik mendistribusikan video tersebut ke beberapa tujuan secara merata dan efisien

(Apostolopoulos et al., 2002). Berikut ilustrasi alur diagram video *streaming* pada Gambar 17.



Gambar 17 Alur diagram Video Streaming

Sumber: (Apple, n.d.)

Ada tiga cara umum yang biasa digunakan dalam menerima stream data sebagai berikut.

1. *Download*

Pada penerimaan *stream* data dengan cara *download*, akses video dilakukan dengan cara melakukan *download* terlebih dahulu suatu *file* multimedia dari server. Penggunaan cara ini mengharuskan keseluruhan suatu *file* multimedia harus diterima secara lengkap pada pengguna. *File* multimedia yang sudah diterima kemudian disimpan pada tempat penyimpanan yang ada di komputer. Pengguna baru dapat mengakses video tersebut setelah berhasil menerima *file* multimedia tersebut secara lengkap. Keuntungan dari penggunaan cara *download* ini adalah akses yang lebih cepat ke salah satu bagian dari *file* tersebut. Sedangkan kekurangannya adalah pengguna yang ingin mengakses video tersebut harus menunggu terlebih dahulu sampai keseluruhan *file* multimedia tersebut diterima secara lengkap.

2. *Streaming*

Pada penerimaan video secara *streaming*, pengguna dapat melihat suatu *file* multimedia hampir bersamaan ketika *file* tersebut mulai diterima. Penggunaan cara ini mengharuskan pengiriman suatu *file* multimedia ke pengguna secara konstan

(*realtime*). Hal ini bertujuan agar pengguna dapat menyaksikan video yang diterima secara langsung tanpa ada bagian yang hilang. Keuntungan dari cara ini adalah pengguna tidak perlu menunggu hingga suatu *file* multimedia dikirimkan secara lengkap. Dengan demikian, penggunaan cara ini memungkinkan sebuah *server* untuk melakukan pengiriman siaran secara langsung kepada pengguna.

3. *Progressive Downloading*

Progressive downloading adalah suatu metode *hybrid* yang merupakan hasil penggabungan antara metode *download* dan metode *streaming*, dimana video yang sedang diakses dapat diterima dengan cara *download* sehingga *player* yang ada pada pengguna sudah dapat mulai menampilkan video tersebut sejak sebagian dari *file* tersebut diterima walaupun *file* tersebut belum diterima secara lengkap

Secara umum, terdapat 4 komponen dari *streaming* yaitu:

1. *Input*

Sumber dari video yang akan di-*streaming*. Sumber tersebut dapat berupa *file* video, DVD, MPEG, dan seterusnya.

2. *Encoder*

Bagian dari aplikasi *server* yang bertugas untuk mengubah video sumber menjadi sebuah format yang sesuai dengan transmisi *streaming*, dimana format ini umumnya memiliki tingkat kompresi tinggi sehingga dapat ditransmisikan dengan baik pada suatu media jaringan.

3. *Server*

File hasil *encoding* kemudian didistribusikan oleh *server* kepada pengguna. Pada aplikasi yang digunakan, *encoder* dan *server* berada pada satu aplikasi yang sama yang terintegrasi satu sama lain.

4. *Player / output*

Player berfungsi untuk melakukan *decoding* terhadap *file* hasil *streaming* dan menampilkannya pada pengguna.

Penerapan teknologi video *streaming* mengharuskan dilakukannya perancangan sistem dan jaringan secara matang untuk memungkinkan pengiriman video *streaming* yang berkualitas baik. Faktor yang memengaruhi proses video *streaming* pada jaringan adalah *Bandwidth*.

Bandwidth didefinisikan sebagai jumlah bit informasi yang melalui suatu jaringan dalam periode waktu tertentu. *Bandwidth* yang tersedia di internet pada umumnya tidak dapat diketahui secara pasti dan sangat bervariasi terhadap waktu. Besarnya *bandwidth* yang tersedia pada jaringan sangat memengaruhi proses kerja suatu video *streaming*. Jika *server* melakukan pengiriman sebuah video dengan bit *rate* tinggi yang melebihi kapasitas *bandwidth* yang tersedia, maka akan terjadi kemacetan sehingga paket-paket tersebut akan di-drop. Hal ini akan menyebabkan terjadinya penurunan kualitas video yang diterima. Hal ini juga berlaku apabila *server* melakukan pengiriman dengan bit *rate* yang lebih rendah. Oleh karena itu, seorang perancang jaringan harus mampu memperkirakan besar kapasitas *bandwidth* yang tersedia dan menyesuaikannya dengan bit *rate* video yang dikirimkan.