

**PENENTUAN REGION OF INTEREST (ROI) UNTUK  
MENGHITUNG JUMLAH KENDARAAN PADA  
JALAN RAYA MENGGUNAKAN  
FRAME SUBSTRACTION**

**SKRIPSI**

**Diajukan Untuk Memenuhi Persyaratan Meraih Gelar Sarjana  
Komputer Pada Program Studi Teknik Informatika Fakultas  
Teknik dan Komputer Universitas Harapan Medan**



**HENDRIZAL SAPUTRA  
202353002**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK DAN KOMPUTER  
UNIVERSITAS HARAPAN MEDAN  
MEDAN  
2023**

## **PERNYATAAN PEMBIMBING**

Saya/Kami dengan ini menyatakan bahwa saya/kami telah memeriksa Skripsi Mahasiswa ini dan menurut pendapat saya/kami, Skripsi Mahasiswa ini telah mencukupi untuk ruang lingkup dan kualitas untuk dianugerahkan gelar Sarjana Teknik/Komputer dalam bidang

Tanggal, 25 Juli 2023

Pembimbing 1

Pembimbing 2

**Dr. Sayuti Rahman, S.T., M.Kom.**

**Dr. Arnes Sembiring, S.T., M.Kom.**

## PERNYATAAN MAHASISWA

Saya yang bertanda tangan di bawah ini:

Nama : Hendrizal Saputra  
Nama Orang Tua : Nasarman  
Program Studi : Teknik Informatika  
Jenjang Studi : Sarjana (S-1)  
NPM : 202353002

Menyatakan bahwa :

1. Skripsi ini merupakan gagasan, rumusan dan ide saya sendiri, tanpa bantuan dari pihak lain kecuali arahan dari Tim Dosen Pembimbing.
2. Skripsi ini belum pernah diajukan untuk mendapat gelar sarjana, baik di Fakultas Teknik dan Komputer Universitas Harapan Medan maupun Perguruan Tinggi lain.
3. Dalam Skripsi ini tidak terdapat karya atau pendapat yang ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan mencantumkan sebagai acuan dalam naskah dengan menyebut nama pengarang dan dicantumkan dalam daftar pustaka.

Demikianlah pernyataan ini saya perbuat dengan sesungguhnya dan apabila dikemudian hari terbukti pernyataan ini tidak benar, maka saya bersedia menerima sanksi akademik yang berlaku, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi ini.

Medan, 25 Juli 2023  
Yang Menyatakan

<i>Mate rai 600</i>
-----------------------------

HENDRIZAL SAPUTRA  
202353002

## **PERSETUJUAN**

**JUDUL** : **PENENTUAN REGION OF INTEREST (ROI) UNTUK MENGHITUNG JUMLAH KENDARAAN PADA JALAN RAYA MENGGUNAKAN FRAME SUBSTRACTION**

**KATEGORI** : **SKRIPSI**  
**NOMOR POKOK MAHASISWA** : **202353002**  
**FAKULTAS** : **TEKNIK DAN KOMPUTER**  
**PROGRAM STUDI** : **TEKNIK INFORMATIKA**  
**TAHUN TAMAT** : **2023**

**DISETUJUI OLEH**  
Komisi Pembimbing

Pembimbing 1

Pembimbing 2

**Dr. Sayuti Rahman, S.T., M.Kom.**

**Dr. Arnes Sembiring, S.T., M.Kom.**

Mengetahui,  
Ketua Program Studi Teknik Informatika

**Ilham Faisal, S.T., M.Kom.**

## KATA PENGANTAR

Puji dan syukur penulis ucapkan kepada Allah SWT atas limpahan berkat, rahmat, serta kemudahan yang telah diberikan sehingga penulis dapat menyelesaikan Skripsi ini yang merupakan syarat untuk mendapatkan gelar Sarjana Teknik/Komputer pada Program Studi Teknik Informatika, Fakultas Teknik dan Komputer, Universitas Harapan Medan. Tak lupa juga shalawat serta salam kepada Nabi Muhammad SAW yang telah memberi petunjuk kepada kita ke jalan yang lurus.

Dalam kurun waktu pengerjaan Skripsi ini penulis menyadari bahwa sangat banyak pihak yang berjasa turut membantu penulis dalam penyelesaian Skripsi ini. Dalam kesempatan ini penulis mengucapkan terima kasih kepada :

1. Ayahanda Nasarman serta Ibunda Sarinah atas doa dan kasih sayangnya yang tulus dan tak terhingga kepada penulis.
2. Bapak Dodi Syahputra, S.T., M.Kom selaku Dekan Fakultas Teknik dan Komputer
3. Bapak Ilham Faisal, S.T., M.Kom. selaku Ketua Program Studi Teknik Informatika Fakultas Teknik dan Komputer .
4. Bapak Dr. Sayuti Rahman, S.T., M.Kom. selaku pembimbing I dan Bapak Dr. Arnes Sembiring, S.T., M.Kom. selaku pembimbing II yang telah meluangkan waktu membimbing penulis selama pengerjaan Skripsi ini.
5. Ibu Haida Dafitri, S.T., M.Kom. selaku Sekretaris Program Studi Teknik Informatika
6. Adik-adik tercinta serta keluarga besar.
7. Teman-teman seperjuangan mahasiswa Program Studi Fakultas Teknik dan Komputer angkatan yang telah memberikan motivasi dan perhatiannya.
8. Semua pihak yang terlibat baik langsung maupun tidak langsung dalam pengerjaan Skripsi ini yang tidak penulis sebutkan satu persatu diucapkan terima kasih.

Penulis menyadari bahwa Skripsi ini masih belum sempurna. Oleh karena itu untuk menyempurnakan Skripsi ini, kritik dan saran yang membangun sangat penulis harapkan. Akhir kata penulis berharap semoga Skripsi ini dapat bermanfaat

Medan, 25 Juli 2023  
Penulis

HENDRIZAL SAPUTRA  
202353002

## ABSTRAK

Semakin tinggi kualitas suatu citra maka semakin detail informasi yang akan di peroleh. Tetapi, tidak semua wilayah citra memungkinkan untuk dilakukan analisis dengan kecepatan proses yang tinggi. Pemilihan algoritma yang tepat berpengaruh terhadap kecepatan waktu pemrosesan. Apabila tidak ada pembatasan untuk area yang akan di proses mengakibatkan waktu pemrosesan secara realtime melebihi waktu pemrosesan maksimal yang seharusnya. Tingginya waktu pemrosesan yang terjadi mengakibatkan aliran data menjadi kurang cepat. Sarana/processor yang digunakan juga mampu mempengaruhi kecepatan pemrosesan. Region Of Interest (ROI) adalah cara yang tepat untuk mengurangi tingginya waktu pemrosesan tersebut. ROI mampu menandai area tertentu sehingga dapat digunakan untuk mengoptimalkan kinerja sistem untuk mendeteksi, menghitung dan mengklasifikasi kendaraan secara realtime. Tanpa adanya ROI, pemrosesan dilakukan pada seluruh piksel citra tanpa terkecuali. Terdapat beberapa tahapan yang dilakukan di dalam penelitian yaitu menganalisis masalah yang ada, penentuan wilayah ROI, aplikasi ROI sebelum proses pengolahan citra dan menganalisis hasil yang didapatkan.

**Kata Kunci :** *Pengolahan citra, klasifikasi kendaraan, region of interest, waktu pemrosesan*

## ABSTRACT

Increasing resolution of an image is more detailed information will be obtained especially in the image used to detect vehicles. But, every single areas are not allow to analyze with higher speed process. If there are no restrictions for the area to be processed, the processing time in real time exceeds the maximum processing time that should be. The high processing time that occurs make less rapid data flow. The high processing time can affect to processing speed. Region Of Interest (ROI) is the right wayto reduce the high processing time. ROI is able to mark certain areas so that it can be used to optimize system performance to detect, calculate and classify vehicles in realtime. Without ROI, processing is carried out on all pixels without exception. There are several steps taken in the research, namely analyzing existing problems, determining the ROI area, application of ROI before the image processing and analyzing the results obtained. The results obtained are by using ROI image processing time can be faster than the processing time when not using ROI.

**Keywords** : *image processing, car classification, region of Interest (ROI), processing time*

## DAFTAR ISI

<b>PERNYATAAN PEMBIMBING</b>	<b>i</b>
<b>PERNYATAAN MAHASISWA</b>	<b>ii</b>
<b>PERSETUJUAN</b>	<b>iii</b>
<b>KATA PENGANTAR</b>	<b>iv</b>
<b>ABSTRAK</b>	<b>vi</b>
<b>ABSTRACT</b>	<b>vii</b>
<b>DAFTAR ISI</b>	<b>viii</b>
<b>DAFTAR TABEL</b>	<b>xi</b>
<b>DAFTAR GAMBAR</b>	<b>xii</b>
<b>BAB 1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Batasan Masalah	3
1.6 Sistematika Penulisan	4
<b>BAB 2 DASAR TEORI</b>	<b>5</b>
2.1 Citra	5
2.2 Pengolahan Citra Digital	5
2.3 Konsep Dasar Video Digital	6
2.4 Karakteristik Video Digital	6
2.4.1 Resolusi	6
2.4.2 Laju Frame	9
2.4.3 Aktivitas Gerak	9
2.4.4 Kedalaman Bit	12
2.4.6 Ruang Warna Red Green Blue	12
2.4.7 Format Piksel YUV	13
2.4.8 Kontras Gambar	14

2.5 Kamera CCTV (Closed Circuit Television)	15
2.6 Computer Vision	15
2.7 Frame Substraction	17
2.8 Region Of Interest	17
2.9 Phyton	18
2.9.1 Python Imaging Library	18
2.9.2 Pyhton Time Library	18
2.9.3 Pyhton Py2Exe	20
2.10 Object Tracking	20
2.11 Penelitian Terkait	21
<b>BAB 3 ANALISIS DAN PERANCANGAN SISTEM</b>	<b>22</b>
3.1 Penentuan Region Of Interest Untuk Menghitung Jumlah Kendaraan Pada Jalan Raya Menggunakan Frame Substraction	22
3.1.1 Analisis Kebutuhan Sistem	22
3.2 Perancangan Sistem	22
3.2.1 Perancangan Flowchart	23
3.3 Algoritma Region Of Interest (ROI)	24
3.3.1 OpenCV Contour Approximation	25
3.3.2 Intensitas Kontur	26
3.4 Analisis Metode Frame Substraction	27
3.5 Perancangan Perangkat Lunak	40
3.5.1 Penentuan Region Of Interest Untuk Menghitung Jumlah Kendaraan Pada Jalan Raya Menggunakan Frame Substraction	40
3.5.1.1 Blok 1	40
3.5.1.2 Blok 2	40
3.5.1.3 Blok 3	40
3.5.1.4 Blok 4	41

<b>BAB 4 HASIL DAN PEMBAHASAN</b>	<b>42</b>
4.1 Analisis Data	42
4.2 Pengujian	42
4.3 Analisa Hasil	46
<b>BAB 5 KESIMPULAN DAN SARAN</b>	<b>50</b>
5.1 Kesimpulan	50
5.2 Saran	51
<b>DAFTAR PUSTAKA</b>	

## DAFTAR TABEL

<b>Tabel 2.1</b> Model sequence video	10
<b>Tabel 2.2</b> 16 Bit truecolor	12
<b>Tabel 3.1</b> Intensitas Kontur ROI	26
<b>Tabel 3.2</b> Algoritma Intensitas Sudut ROI	27
<b>Tabel 3.3</b> Nilai Piksel Grayscale Frame Background	28
<b>Tabel 3.4</b> Nilai Pikse Grayscale Frame Foreground	29
<b>Tabel 3.5</b> Perhitungan selisih background dan foreground	29
<b>Tabel 3.6</b> Hasil Frame Substraction	34
<b>Tabel 3.7</b> Ekstraksi Background	34
<b>Tabel 3.8</b> Hasil Eksraksi Background	39
<b>Tabel 4.1</b> Hasil Eksraksi Background	49

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Video dengan tinggi N Pixel dan lebar M Pixel	7
<b>Gambar 2.2</b> Ruang warna RGB	12
<b>Gambar 2.3</b> Model warna RGB	12
<b>Gambar 2.4</b> YUV Sampling 4:4:4	13
<b>Gambar 2.5</b> YUV Sampling 4:2:2	14
<b>Gambar 2.6</b> YUV Sampling 4:2:0	14
<b>Gambar 2.7</b> Kontras Gambar	15
<b>Gambar 2.8</b> Hirarki proses Computer Vision	16
<b>Gambar 3.1</b> Diagram Flowchart Sistem Keseluruhan	23
<b>Gambar 3.2</b> Area ROI	24
<b>Gambar 3.3</b> Garis Area ROI	24
<b>Gambar 3.4</b> Kurva Area ROI	26
<b>Gambar 3.5</b> Frame Video	27
<b>Gambar 3.6</b> Frame Background	28
<b>Gambar 3.7</b> Frame Foreground	28
<b>Gambar 3.8</b> Hasil Frame Substraction	39
<b>Gambar 3.9</b> Kode Program untuk membaca file	40
<b>Gambar 3.10</b> Kode Program untuk menandai ROI	40
<b>Gambar 3.11</b> Kode Program untuk mendeteksi objek	40
<b>Gambar 3.12</b> Kode Program membaca frame-frame video	41
<b>Gambar 4.1</b> Kode Program untuk membaca file	42
<b>Gambar 4.2</b> Kode Program untuk menandai ROI	42
<b>Gambar 4.3</b> Kode Program untuk mendeteksi objek	43
<b>Gambar 4.4</b> Kode Program membaca frame-frame video	44
<b>Gambar 4.5</b> Feed Videotes	45
<b>Gambar 4.6</b> Feed Frame Videotes	45
<b>Gambar 4.7</b> Feed ROI Videotes	46
<b>Gambar 4.8</b> ROI Video	46
<b>Gambar 4.9</b> ROI Video2	47
<b>Gambar 4.10</b> ROI Video3	47

<b>Gambar 4.11</b> ROI Video4	47
<b>Gambar 4.12</b> ROI Video5	48
<b>Gambar 4.13</b> ROI Video6	48
<b>Gambar 4.14</b> ROI Video7	48
<b>Gambar 4.15</b> ROI Video8	49

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Perkembangan teknologi telah semakin maju, baik dalam bidang hardware maupun *software*. Banyak perusahaan asing maupun dalam negeri yang memproduksi dan menjual produk elektronik berbasis multimedia seperti *smarthphone*, *notebook*, tablet, televisi, kamera, dan lain sebagainya. Perkembangan industri multimedia ini tidak terlepas dari citra sebagai salah satu parameter penting dalam kualitas teknologi multimedia. Selain itu, kebutuhan akan citra dimanfaatkan dalam bidang medis sebagai media pembantu diagnosa penyakit, peningkatan resolusi citra satelit, pengenalan pola, sistem keamanan berbasis citra, dan media pembelajaran visual yang efektif dalam bidang pendidikan. Dalam bidang-bidang tersebut citra dengan resolusi tinggi sangat dibutuhkan, sebab semakin tinggi resolusi citra maka informasi yang diperoleh dari citra tersebut lebih detail, sehingga analisis terhadap citra menjadi lebih akurat.

Penelitian yang dilakukan oleh Khairul Umam dan Benny Sukma Negara untuk mendeteksi pergerakan objek telah menggunakan metode *frame subtraction* menghasilkan tingkat keberhasilan sebesar 80%. Penelitian yang dilakukan Karina Mariane Kaloh, Vecky C. Poekoel dan Muhamad Dwisnanto Putro menghasilkan akurasi pendeteksian menggunakan metode *frame subtraction* sebesar 80.56%.

Penelitian yang dilakukan Muhammad Affandes dan Afdi Ramadani untuk mendeteksi gerak banyak objek dengan metode *frame subtraction* menghasilkan akurasi 88.3%. Berdasarkan penelitian terdahulu, implementasi *frame subtraction* dinilai berguna untuk mendeteksi pergerakan objek dalam sistem monitoring sebab menghasilkan akurasi yang dinilai besar.

Rangkaian citra biasanya diperoleh dari rangkaian *frame* tanpa adanya segmentasi menggunakan ROI (Region Of Interest). Super resolusi pada rangkaian *frame* ini umumnya dilakukan pada keseluruhan piksel *frame*, padahal kebutuhan informasi dari suatu citra hanya terdapat pada suatu bagian tertentu dari *frame* tersebut bukan pada keseluruhan piksel frame. Oleh karena itu, dibutuhkan cara untuk mendapatkan bagian frame tersebut agar proses ekstraksi dapat lebih efektif yaitu menggunakan ROI (Region Of Interest) dan

agar proses ekstraksi dapat lebih efektif yaitu menggunakan ROI (Region Of Interest) dan melakukan proses tracking dalam rangkaian citra.

Dalam mengurai persentase kemacetan dan perbandingan jumlah laju kendaraan setiap hari nya pada lalu lintas jalan raya, institusi terkait memerlukan waktu yang cukup lama dalam pelaksanaannya. Perhitungan jumlah kendaraan menjadi pekerjaan yang memerlukan konsentrasi dan waktu seiring dengan bertambahnya jumlah kendaraan yang beredar. Perhitungan kendaraan secara manual oleh manusia dalam jangka waktu lama dapat menimbulkan kesalahan perhitungan dikarenakan fokus manusia dapat berkurang apabila melakukan suatu hal dalam waktu yang lama. Diperlukan sistem hitung kendaraan yang bekerja secara otomatis untuk membantu manusia dalam menghitung jumlah kendaraan. Penelitian ini menawarkan sistem yang dapat menghitung jumlah kendaraan berdasarkan jenis secara otomatis menggunakan *Metode Frame Subtraction*. Diharapkan dengan adanya metode ini dapat memudahkan pekerjaan institusi dinas perhubungan dalam melakukan perbandingan jumlah kendaraan setiap hari nya dan menentukan persentase kemacetan di lalulintas jalan raya.

Citra objek dalam *frame* akan dipilih menggunakan ROI. Selanjutnya proses tracking pada tiap *frame* citra dilakukan untuk mendapatkan posisi objek pada serangkaian frame video. Seperti yang di uraikan oleh Hendrianto, berdasarkan hasil dari penelitian yang telah dilakukan dapat diambil kesimpulan bahwa penerapan proses ROI dapat digunakan sebagai solusi untuk mengatasi masalah waktu pemrosesan. Hal ini dibuktikan dengan pengujian yang dilakukan sebelum dan setelah pengimplementasian proses ROI.

Pada penelitian ini bertujuan untuk mengatasi masalah waktu pemrosesan yang terjadi ketika proses deteksi kendaraan secara realtime. ROI ditempatkan pada proses sebelum pengolahan citra, hal ini berguna agar daerah yang dilakukan proses pengolahan citra berfokus pada frame ROI dan bukan pada seluruh frame. Sehingga, ROI dapat digunakan untuk membatasi kawasan pengamatan suatu kendaraan. Dengan adanya latar belakang di atas, maka akan dilakukan penelitian mengenai **“Penentuan Region Of Interest (ROI) Untuk Menghitung Jumlah Kendaraan Menggunakan Frame Substraction”**.

## 1.2 Rumusan Masalah

Setelah permasalahannya dapat diidentifikasi maka dapat diambil suatu rumusan masalah yaitu :

1. Bagaimana menghitung jumlah kendaraan pada lalu lintas agar memudahkan institusi terkait dapat menentukan jumlah laju kendaraan dan mengurai persentase kemacetan yang ada di jalan raya?
2. Bagaimana mengimplementasikan algoritma *Frame Substraction* untuk proses penentuan ROI pada sebuah video?

## 1.3 Tujuan Penelitian

Tujuan yang akan dicapai dalam pembangunan aplikasi Penentuan Region Of Interest (ROI) Untuk Menghitung Jumlah Kendaraan Menggunakan Frame Substraction adalah:

1. Untuk menghitung jumlah kendaraan pada lalu lintas agar memudahkan institusi terkait dapat menentukan jumlah laju kendaraan dan mengurai persentase kemacetan yang ada di jalan raya.
2. Memberikan bentuk sistem yang dapat menentukan ROI-nya sendiri dalam proses *object tracking*.

## 1.4 Manfaat Penelitian

Manfaat yang akan dicapai dalam pembangunan aplikasi Penentuan Region Of Interest (ROI) Untuk Menghitung Jumlah Kendaraan Menggunakan Frame Substraction adalah:

1. Memudahkan institusi terkait dalam melakukan perhitungan jumlah kendaraan di Jalan raya dan mengurai persentase kemacetan yang ada
2. Dapat menentukan ROI objek atau kendaraan yang bergerak di jalan raya

## 1.5 Batasan Masalah

Dalam melakukan penelitian ini terdapat batasan masalah yang membatasi ruang lingkup penelitian. Batasan masalah tersebut yaitu :

1. Aplikasi yang dibangun menggunakan bahasa pemrograman Python
2. Menggunakan metode ROI untuk penandaan objek pada video.

## **1.6 Sistematika Penulisan**

Sistematika penulisan skripsi ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan dan memudahkan pembaca dalam memahaminya. Sistematika penulisan skripsi ini adalah sebagai berikut:

### **BAB 1 PENDAHULUAN**

Bab 1, menguraikan tentang latar belakang yang mendasari dibuatnya skripsi ini, identifikasi masalah, rumusan masalah dan batasan masalah, tujuan, metodologi penelitian, dan sistematika penulisan.

### **BAB 2 DASAR TEORI**

Bab 2, membahas tentang teori yang digunakan dalam analisa permasalahan yang ada yaitu multimedia, , teori dasar Object Extraction, video, format *file*, *ROI*, *Frame Extraction*, dan metode pendekatan sistem.

### **BAB 3 ANALISIS DAN PERANCANGAN SISTEM**

Bab 3, menguraikan tentang analisis dan perancangan aplikasi yang dibangun, yaitu analisis yang meliputi analisis kebutuhan fungsional dan non fungsional, arsitektur sistem.

### **BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM**

Bab 4, berisi tentang implementasi *Frame Substraction* ke dalam perangkat lunak, implementasi antarmuka dan pengujian perangkat lunak.

### **BAB 5 KESIMPULAN DAN SARAN**

Bab 5 berisi kesimpulan berdasarkan tujuan yang ingin dicapai dan saran untuk hasil penelitian ini.

## **BAB II**

### **DASAR TEORI**

#### **2.1 Citra**

Berdasarkan pembuatannya citra dapat dikelompokkan menjadi dua bagian, yaitu citra kontinu dan citra diskrit. Citra kontinu merupakan citra yang dihasilkan dari sistem optik yang menerima sinyal analog. Sedangkan citra diskrit merupakan citra kontinu yang melalui proses digitalisasi, jadi dapat dikatakan bahwa citra diskrit merupakan citra digital (Munir, 2004)

Dari sudut pandang matematis citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra (dua dimensi). Secara matematis citra dapat didefinisikan sebagai fungsi  $f(x, y)$  berukuran M baris dan N kolom, yang dalam hal ini  $(x, y)$  merupakan koordinat spasial pada bidang dwimatra sedangkan amplitudo  $f$  dititik koordinat  $(x, y)$  dinamakan intensitas atau tingkat keabuan dari sebuah citra pada titik tersebut. Apabila nilai dari  $x$ ,  $y$  dan nilai amplitudo  $f$  secara keseluruhan berhingga dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut merupakan citra digital (Putra 2010).

#### **2.2 Pengolahan Citra Digital**

Citra merupakan sesuatu yang memiliki banyak informasi. Namun seringkali citra tersebut mengalami penurunan mutu (degradasi), seperti mengandung cacat atau noise. Memiliki warna yang terlalu kontras. Kurang tajam, kabur (*Blurring*), dan lain sebagainya. Tentunya hal tersebut akan menyulitkan untuk menginterpretasikan citra yang ada karena informasi yang terdapat pada citra akan menurun. Untuk itu perlu dilakukan manipulasi sehingga citra tersebut kualitasnya dapat lebih baik dari sebelumnya. Proses memanipulasi citra tersebut dapat dilakukan dengan bantuan komputer, dengan kata lain proses tersebut dapat dikatakan dengan pengolahan citra digital (Munir, 2004).

Citra digital direpresentasikan dengan matriks karena itu pada dasarnya operasi pada citra digital adalah memanipulasi setiap elemen yang terdapat matriks. Elemen-elemen yang ingin dimanipulasi dapat berupa elemen tunggal, sekumpulan elemen yang berdekatan, atau seluruh elemen yang terdapat pada matriks (Munir, 2004)

## 2.3 Konsep Dasar Video Digital

Video digital adalah teknologi untuk menangkap, merekam, memproses, mentransmisikan dan menata ulang gambar bergerak. Biasanya menggunakan sinyal elektronik, atau media digital. Jenis sistem video recording yang bekerja menggunakan sistem digital dibandingkan dengan analog dalam hal representasi videonya (Nurhayati, Universitas Diponegoro, 2010). Video digital pada dasarnya merupakan susunan atas serangkaian frame dengan kecepatan tertentu, tergantung pada frame rate yang diberikan (dalam frame/second) yang bekerja menggunakan sistem digital yang diperoleh dari hasil direkam, kemudian diproses, dan disimpan. Representasi visual tujuan utamanya adalah agar manusia sebagai audience yang melihat merasa berada di scene (lokasi) atau ikut berpartisipasi dalam kejadian yang ditampilkan. Oleh sebab itu, suatu gambar harus dapat menyampaikan informasi spatial dan temporal dari suatu scene (Ramayanti, adoc.tips, 2009). Video digital meliputi rangkaian frame dan redundansi, masing - masing frame merupakan gambar atau citra (image) digital.

## 2.4 Karakteristik Video Digital

Karakteristik video digital ditentukan oleh resolusi (resolution) atau dimensi frame (frame dimension), kedalaman pixel (pixel depth), dan laju frame (frame rate). Karakteristik – karakteristik video digital ini digunakan untuk menentukan kualitas video dan jumlah bit yang dibutuhkan saat menyimpan maupun pentransmisian video digital tersebut.

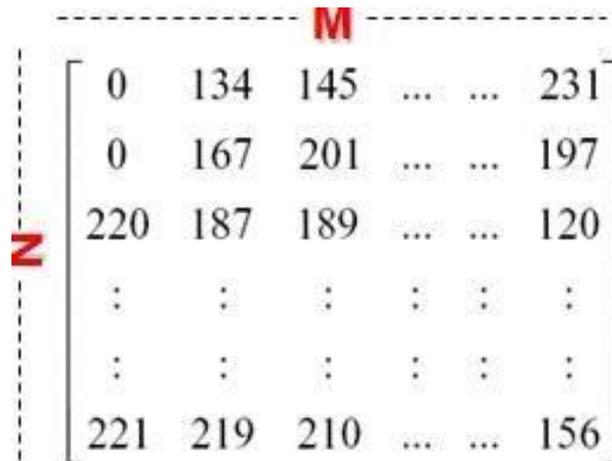
### 2.4.1 Resolusi

Resolusi video merupakan tingkat detil suatu video digital tersebut. Semakin tinggi resolusi video maka akan semakin tinggi pula tingkat detil dari video tersebut. Satuan dalam pengukuran resolusi video digital dapat berupa ukuran fisik (jumlah garis per mm dibagi jumlah garis per inchi) ataupun dapat juga berupa ukuran video menyeluruh (jumlah garis per tinggi citra).

Menurut Saputra, unikom (2013) resolusi video dapat diukur dengan berbagai cara sebagai berikut:

#### a. Resolusi pixel

Resolusi pixel merupakan perhitungan jumlah pixel dalam sebuah video digital. Sebuah video dengan tinggi N pixel dan lebar M pixel, berarti memiliki resolusi sebesar M x N. Resolusi pixel akan memberikan dua buah angka integer yang secara berurutan akan mewakili jumlah pixel lebar dan jumlah pixel tinggi dari video tersebut.



**Gambar 2.1** Video dengan tinggi N pixel dan lebar M pixel (Rijal,2013)

Resolusi pixel juga merupakan hasil perkalian jumlah pixel lebar dan tingginya dan kemudian dibagi dengan 1 juta. Jenis resolusi pixel seperti ini sering ditemui pada kamera digital. Suatu video yang memiliki lebar 2.048 pixel dan tinggi 1.536 pixel, memiliki total pixel sebanyak  $2.048 \times 1.536 = 3.145.728$  pixel = 3.1 mega pixel. Perhitungan lainnya menyatakan dalam satuan pixel per inchi, yang berarti banyaknya pixel yang ada sepanjang 1 inchi baris dalam video.

#### b. Resolusi spasial

Resolusi spasial menunjukkan seberapa dekat jarak setiap garis pada video. Jarak tersebut tergantung dari sistem yang menciptakan video tersebut. Resolusi spasial menghasilkan jumlah pixel per satuan panjang. Resolusi spasial dari sebuah monitor computer adalah 72 hingga 100 garis per inchi atau dalam resolusi pixel 72 hingga 100 ppi. Contoh lain yaitu video digital dengan ukuran HD beresolusi 720p, 1080p, dan 1080i, semua dengan piksel persegi. Huruf tersebut menunjukkan interlaced dan progressive scanning. Jadi resolusi 720p memiliki ukuran  $1280 \times 720$  piksel, resolusi 1080p memiliki ukuran  $1920 \times 1080$  piksel, dan resolusi 1080i memiliki ukuran  $1920 \times 1080$  piksel pada frekuensi 50 HZ.

#### c. Resolusi spectrum

Sebuah video digital membedakan intensitas kedalam beberapa spectrum. Video multi spectrum akan memberikan spectrum atau panjang gelombang yang lebih baik untuk menampilkan warna.

#### d. Resolusi temporal

Resolusi temporal berkaitan dengan video atau disebut juga frame rate. Suatu video merupakan kumpulan frame statis yang berupa gambar yang berurutan dan ditampilkan secara cepat. Resolusi temporal memberikan jumlah frame yang dapat ditampilkan setiap detik dengan satuan frame per second (fps).

#### e. Resolusi spectrum

Sebuah video digital membedakan intensitas kedalam beberapa spectrum. Video multi spectrum akan memberikan spectrum atau panjang gelombang yang lebih baik untuk menampilkan warna.

#### f. Resolusi temporal

Resolusi temporal berkaitan dengan video atau disebut juga frame rate. Suatu video merupakan kumpulan frame statis yang berupa gambar yang berurutan dan ditampilkan secara cepat. Resolusi temporal memberikan jumlah frame yang dapat ditampilkan setiap detik dengan satuan frame per second (fps). Penggunaan frame rate disesuaikan dengan aktifitas gerak dari video, untuk video dengan aktifitas gerak tinggi atau high motion perlu menggunakan frame rate yang besar, tetapi jika video dengan aktifitas gerak rendah atau low motion maka lebih cocok dan efisien menggunakan frame rate yang rendah.

#### g. Resolusi radiometrik

Resolusi ini memberikan nilai atau tingkat kehalusan video yang dapat ditampilkan dan biasanya ditampilkan dalam satuan bit. Contoh video 8 bit dan video 256 bit. Semakin tinggi resolusi radiometrik ini maka semakin baik perbedaaan intensitas yang ditampilkan. Menurut Nurhayati (2010) berdasarkan ukuran video digital, beberapa ukuran video dari yang paling rendah hingga paling tinggi digunakan saat ini yaitu :

##### a. SQCIF (Sub Quarter Common Intermediate Format)

SQCIF merupakan salah satu jenis dari resolusi CIF, standar resolusi ini memiliki ukuran yang lebih kecil yaitu 128x96, frame rate 30 fps. SQCIF digunakan untuk mendukung kamera dengan resolusi yang lebih kecil dan transmisi video dengan bandwidth yang lebih rendah dan dirancang untuk menampilkan video dengan ukuran kecil seperti tampilan di handphone dan untuk video conference melalui internet.

##### b. QCIF (Quarter Common Intermediate Format)

QCIF ini merupakan varian yang didukung oleh resolusi CIF, QCIF memiliki ukuran resolusi 176x144 dengan mengurangi setengah dari kedua resolusi horizontal dan vertikal dari ukuran resolusi CIF dasar.

c. CIF (Common Intermediate Format)

CIF adalah sebuah format standarisasi resolusi horizontal dan vertikal dalam piksel yang biasa digunakan dalam sistem video teleconference. CIF memiliki ukuran resolusi 352 x 288. CIF menggunakan frame rate 30 fps.

d. SD (Standard Definition)

SD (Standard Definition) memiliki kualitas gambar sedikit lebih baik dari video sebelumnya karena ukurannya lebih besar, memiliki resolusi 720x576. SD membutuhkan piksel yang kecil, dan membutuhkan frekuensi 50 Hz, namun dari segi kualitas lebih baik dari CIF. Dalam komunikasi menggunakan video resolusi SD saat ini sudah sangat banyak bahkan, video dengan SD digunakan pada saat melakukan streaming dengan didukung bandwidth yang memadai.

e. HD (High Definition)

HD merupakan video yang secara gambaran umum beresolusi 720p, 1080p, dan 1080i, semua dengan piksel persegi. Huruf tersebut menunjukkan interlaced dan progressive scanning. Jadi resolusi 720p memiliki ukuran 1280 x 720 piksel, resolusi 1080p memiliki ukuran 1920 x1080 piksel, dan resolusi 1080i memiliki ukuran 1920x1080 piksel pada frekuensi 50 HZ .

#### 2.4.2 Laju Frame (frame rate)

Menurut Haslinda (2012) laju frame (frame rate) menunjukkan jumlah frame yang digambarkan tiap detik dinyatakan dalam frame/detik. Sehubungan dengan laju frame ini, ada dua hal yang perlu diperhatikan, yaitu kehalusan gerakan (smooth motion) dan kilatan (flash). Kehalusan gerakan ditentukan oleh jumlah frame yang berbeda per detik. Untuk mendapatkan gerakan yang halus, video digital setidaknya harus berada pada 25 frame/detik. Kilatan ditentukan oleh jumlah berapa kali layar digambar per detik. Dengan 20 frame/detik, kilatan sudah dapat dlenyapkan.

#### 2.4.3 Aktivitas Gerak (Motion Activity) Pada Video

Menurut Peker (2001) Motion Activity digunakan untuk menggambarkan tingkat atau intensitas kegiatan, tindakan, atau gerakan pada suatu urutan video. Model sequence video diklasifikasikan dalam tiga tingkat motion activity yaitu low motion, medium motion, dan high motion seperti pada tabel berikut:

**Table 2.1** Model sequence video

No.	<i>Low Motion</i>	<i>Medium Motion</i>	<i>High Motion</i>
1.	Video <i>conference</i> : Seseorang yang sedang berbicara di depan kamera dimana hanya terdapat pergerakan kepala dan bahu dengan detail pergerakan yang tidak banyak	Kerumunan : Pergerakan beberapa orang di keramaian dengan tingkat detail pergerakan yang banyak terdapat pergerakan kamera.	<i>Sport</i> : Sebuah tembakan tajam pada permainan <i>football</i> dimana terdapat perpindahan pergerakan pemain dan <i>background</i> yang sangat banyak.
2.	<i>News</i> : Satu atau lebih reporter berita yang sedang berbicara ke arah kamera dimana terdapat pergerakan kepala dan bahu.	Seni bela diri : Sebuah adegan pertarungan yang melibatkan beberapa orang dengan pergerakan yang cepat.	<i>Car chase</i> : Sebuah adegan balap mobil dengan kecepatan yang sangat tinggi dan detail pergerakan yang sangat banyak.
3.	<i>Ship</i> : Kapal yang melaju dengan kecepatan rendah.	<i>Airport</i> : Adegan beberapa orang berjalan dikeramaian dan terdapat pergerakan kamera.	
4.		Animasi : Beberapa adegan karakter animasi komputer dengan adanya pergerakan kamera	

Sumber (Yadavalli,2003)

#### 2.4.4 Kedalaman Bit

Menurut Rohman (2011) kedalaman bit menentukan jumlah bit yang digunakan untuk mempresentasikan tiap piksel pada sebuah frame dan dinyatakan dalam bit/piksel. Semakin banyak jumlah bit yang digunakan untuk mempresentasikan sebuah piksel, yang berarti semakin tinggi kedalaman pikselnya, maka semakin tinggi pula kualitasnya. Kedalam piksel paling rendah terdapat pada binary-value image yang hanya menggunakan 1 bit untuk tiap piksel, sehingga hanya ada dua kemungkinan bagi tiap piksel, yaitu 0 (hitam) atau 1 (putih).

Nilai 1 byte (8 bit) untuk tiap piksel, diperoleh 28 atau 256 level intensitas. Kemudian video dengan kedalam 16 bit biasanya disebut video high color, dimana setiap pikselnya diwakili oleh 2 byte atau 16 bit dengan memiliki 65.356 warna. Dalam formasi bitnya, nilai merah dan biru mengambil tempat di 5 bit di kanan dan kiri. Komponen hijau memiliki 5 bit ditambah 1 bit ekstra, pemilihan komponen hijau dengan deret 6 bit dikarenakan penglihatan manusia lebih sensitif terhadap warna hijau untuk lebih jelasnya dapat dilihat pada Tabel 2.2.

**Tabel 2.2** 16 Bit truecolor

| Bit |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| R   | R   | R   | R   | R   | G   | G   | G   | G   | G   | G   | B   | B   | B   | B   | B   |

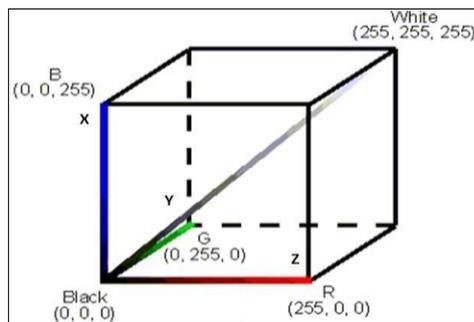
Sumber (Rijal,2013)

#### 2.4.5 Representasi Warna

Dalam video digital representasi warna merupakan cara standar untuk menspesifikasikan suatu warna tertentu dengan mendefinisikan suatu sistem koordinat 3D dan suatu ruang bagian yang mengandung semua warna yang dapat dibentuk ke dalam suatu model tertentu. Suatu warna yang dapat dispesifikasikan menggunakan suatu model akan terhubung ke suatu titik tunggal dalam suatu ruang bagian yang disefinisikannya. Pada umumnya, data video dipisahkan menjadi komponen warna (chrominance) dan komponen kecerahan (luminance). Komponen tersebut dipisahkan dengan beberapa cara, yaitu RGB dan YUV.

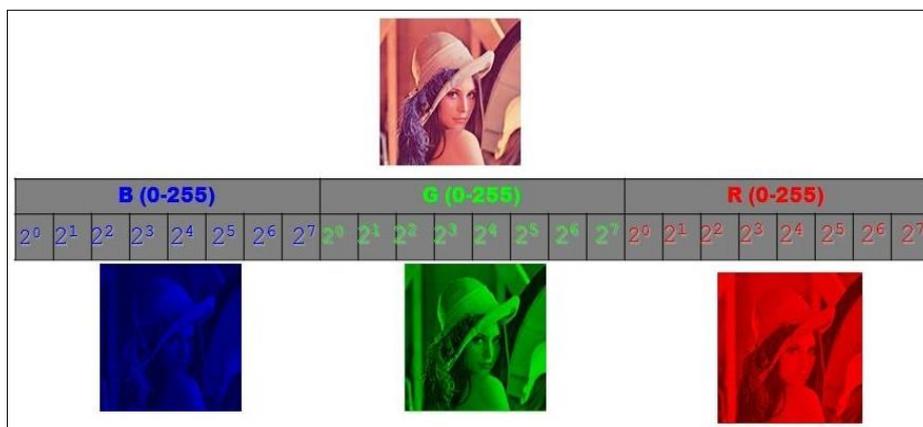
### 2.4.6 Ruang Warna Red Green Blue (RGB)

Menurut Putranto (2010) kombinasi dari komponen warna dasar yaitu merah, hijau, dan biru dapat disebut juga ruang warna. Untuk 24 bit per piksel, kisaran R, G, B bervariasi dari 0 sampai 255. Apabila R, G, B adalah semua 0 maka warna yang dihasilkan akan menjadi hitam. Jika R, G, B adalah semua 255 maka warna keluaran akan menjadi putih. Konsep ruang warna RGB ditentukan pada gambar 2.5.



**Gambar 2.2** Ruang warna RGB (Rijal, 2013)

Sumbu x mewakili rentang warna biru, sumbu y mewakili rentang warna hijau dan sumbu z mewakili rentang warna merah. Seperti pada Gambar 2.6, terlihat bahwa warna hitam diwakili di asal dan warna putih diwakili di sudut lain, dimana warna merah, hijau, dan biru adalah masing masing 255. Demikian pula kita dapat mendapatkan nilai warna lain di berbagai penjuru kubus sesuai dengan nilai-nilai RGB yang berbeda.



**Gambar 2.3** Model Warna RGB (Rijal, 2013)

### 2.4.7 Format Piksel YUV

Menurut Wang (2012) sistem warna YUV adalah pengkodean warna skema untuk memisahkan komponen video ataupun gambar digital yaitu komponen kecerahan atau cahaya (luminance) dan komponen warna sinyal (chrominance). Mata manusia kurang peka terhadap variasi warna daripada intensitas variasi, jadi YUV memungkinkan pengkodean informasi cahaya (Y) bandwidth penuh dan informasi krominan (UV) setengah bandwidth atau komponen kecerahan dinyatakan dengan Y, sedangkan dua komponen warna dinyatakan dengan U dan V. Komponen luminan dapat diperlakukan sebagai skala abu-abu dari suatu RGB image. Ruang warna berbasis YUV diperoleh dari komponen RGB yang telah mengalami koreksi gamma, sehingga disebut R'G'B dengan nilai sebagai berikut:

$$Y = 0,299R + 0,587G + 0,114B$$

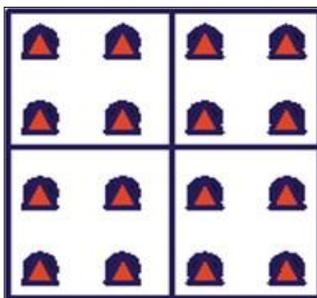
$$U = 0,147R - 0,289G + 0,436B = 0,492 (B-Y)$$

$$V = 0,615R - 0,515G - 0,100B = 0,877 (R-Y)(2.1)$$

Mata manusia lebih sensitif terhadap luminance daripada chrominance pada komponen gambar, itu sebabnya pada aplikasi pengkodean seperti motion estimation, suatu block dikodekan atau tidak, didasarkan pada komponen kecerahan (luminance). Dalam video coding komponen luminance dihitung untuk setiap piksel, tetapi dua komponen chrominance (U dan V) adalah rata-rata dari 4 piksel, ini membagi dua jumlah data yang dikirimkan untuk setiap piksel.

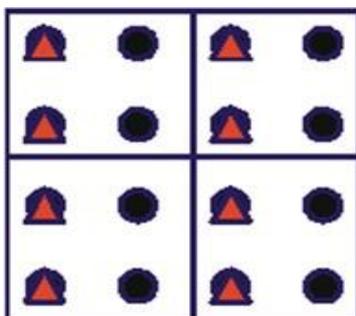
Menurut Sullivan (2008) notasi “A : B : C” adalah notasi yang digunakan untuk menggambarkan seberapa sering U dan V adalah sampel relative terhadap Y.

1. 4:4:4 berarti untuk setiap 2 x 2 Y piksel terdapat 4 U dan 4 V piksel, seperti pada gambar 2.8.



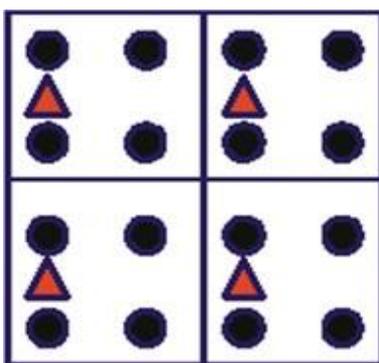
**Gambar 2.4** YUV Sampling 4:4:4(Rijal,2013)

2. 4:2:2 berarti 2:1 downsampling arah horizontal, tanpa downsampling arah vertical, dimana setiap 2 x 2 Y piksel terdapat 2 U dan 2 V. Seperti yang terlihat pada gambar 2.9



**Gambar 2.5** YUV Sampling 4:2:2 (Rijal, 2013)

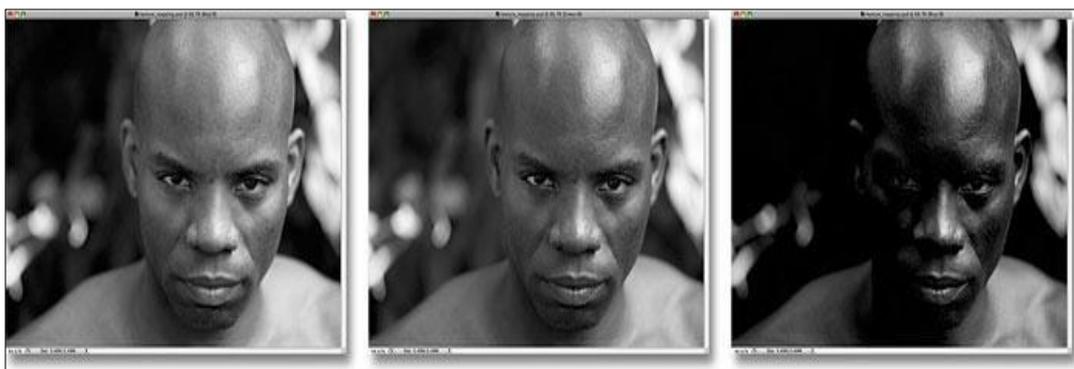
3. 4:2:0 berarti 2:1 downsampling arah horizontal dan arah vertical, dimana setiap 2 x 2 Y piksel terdapat 1 U dan 1 V piksel. Seperti pada gambar 2.10.



**Gambar 2.6** YUV Sampling 4:2:0 (Rijal, 2013)

#### 2.4.8 Kontras Gambar

Menurut Peli (1990) kontras merupakan perbedaan luminance dan warna pada sebuah objek gambar. Kontras suatu video adalah distribusi piksel terang dan gelap. Video grayscale dengan nilai kontras rendah maka akan terlihat lebih gelap, terlalu terang, atau abu-abu. Video dengan kontras tinggi memiliki daerah gelap dan terang yang luas. Semakin tinggi kontras video, akan semakin tajam perbedaan antara warna-warna yang terang dan warna-warna yang gelap. Dalam persepsi visual kontras, kontras ditentukan berdasarkan perbedaan pada warna dan kecerahan pada objek atau objek lainnya dengan bidang yang sama. Karena visual manusia lebih sensitif terhadap kontras daripada warna. Perbedaan tingkat kontras pada suatu video terlihat pada Gambar 2.11, dimana daerah kiri pada gambar dengan kontras yang rendah dan daerah kanan dengan kontras yang tinggi.



**Gambar 2.7** Kontras gambar  
(<http://desaindotgrafis.blogspot.com/2014>)

## 2.5 Kamera CCTV (Closed Circuit Television)

CCTV (Closed Circuit Television) adalah suatu alat yang yang dapat mengirimkan data berupa video melalui transmisi kabel coaxial, FO atau UTP bahkan tanpa kabel ke lokasi tertentu untuk dimonitor, direkam, atau untuk dianalisa. Saat ini penggunaan CCTV sudah mengarah ke IP network camera (IP CCTV), walaupun di beberapa tempat masih ada yang menggunakan analog karena disesuaikan dengan kebutuhan aplikasi pengguna. Jadi CCTV berfungsi untuk memonitor suatu ruangan melalui layar televisi / monitor, dengan menampilkan gambar dari kamera yang dipasang di setiap ruangan (biasanya tersembunyi) yang diinginkan oleh bagian keamanan atau yang berkepentingan. Semua kegiatan di dalamnya dapat dimonitor di suatu ruangan atau secara remote (Suharto Cahyadi, docplayer.info, 2015).

## 2.6 Computer Vision

Di dalam bukunya (Munir 2018) menjelaskan bahwa *computer vision* adalah salah satu terminologi yang berkaitan dengan pengolahan citra. Pada dasarnya *computer vision* mencoba untuk meniru sistem kerja dari visual manusia. *Computer vision* merupakan

proses otomatis yang mengintegrasikan beberapa proses untuk persepsi *visual*, misalnya akuisisi citra, pengolahan citra, klasifikasi, pengenalan, dan pengambilan keputusan.

Pada dasarnya, terdapat tiga aktifitas utama yang terjadi dalam *computer vision*, yakni :

1. Akuisisi citra.
2. Memproses dan memodifikasi data citra.
3. Menganalisis dan menginterpretasi citra, kemudian menggunakan hasil pemrosesan untuk tujuan tertentu. Digambarkan dalam bentuk hirarki (Munir 2018) mengutip dari Schalkoff, Robert J mengklasifikasikan setiap proses yang terdapat di dalam *computer vision* terlihat seperti berikut:

Hirarki Pemrosesan	Contoh Algoritma
<i>preprocessing</i>	<i>noise removal</i> <i>contrast enhancement</i>
<i>lowest-level feature extraction</i>	<i>edge detection</i> <i>texture detection</i>
<i>intermediate-level feature identification</i>	<i>connectivity</i> <i>pattern matching</i> <i>boundary coding</i>
<i>high-level scene interpretation via images</i>	<i>model-base recognition</i>

**Gambar 2.8** Hirarki Proses *Computer Vision*  
(Munir, 2004)

## 2.7 *Frame Subtraction*

*Frame subtraction* atau nama lainnya adalah *foreground detection* merupakan salah satu teknik pada pengolahan citra dan *computer vision*. *Frame subtraction* biasanya digunakan untuk melakukan pendeteksian terhadap objek seperti manusia, kendaraan, tulisan, ataupun bagian dari manusia itu sendiri. Secara umum metode ini bekerja dengan memisahkan objek yang akan dideteksi (*foreground*) dengan latar belakang (*background*). Teknik *Frame subtraction* merupakan metode yang digunakan untuk mendeteksi objek bergerak dari sebuah video yang memiliki sebuah *background* yang statis (Lavanya 2014). *Background* merupakan sejumlah piksel dari sebuah citra yang statis tak bergerak di depan kamera, sedangkan *foreground* merupakan semua objek yang ada selain dari *background*. *Foreground* didapatkan hanya saat *background* telah ditentukan. (Irianto dkk. 2009)

Di dalam jurnalnya, (Umam dan Negara 2016) menjabarkan tahapan- tahapan dari metode *frame subtraction* dalam mendeteksi obyek bergerak, yaitu :

1. Menentukan citra referensi (*Frame*).
2. Menangkap citra objek yang akan di deteksi (*foreground*)
3. Membandingkan *Frame* dengan *foreground*
4. Menentukan hitam atau putih dari citra hasil *background subtraction*.

## 2.8 *Region Of Interest (ROI)*

*Region of Interest (ROI)* merupakan salah satu fitur yang tersedia dalam JPEG2000. ROI memungkinkan dilakukannya pengkodean secara berbeda pada area tertentu dari citra digital, sehingga mempunyai kualitas yang lebih baik dari area sekitarnya. Fitur ini menjadi sangat penting, bila terdapat bagian tertentu dari citra digital yang dirasakan lebih penting dari bagian yang lainnya (Agustina Linda, Universitas Negeri Semarang, 2018).

Untuk melakukan ROI, perlu diidentifikasi koefisien–koefisien yang termasuk dalam ROI. Hal tersebut dimaksudkan untuk menghasilkan sebuah ROI mask, yang merupakan indikator dari koefisien yang termasuk ROI atau bukan.

Umumnya yang digunakan sebagai penanda merupakan sebuah angka biner, yang dimiliki oleh piksel-piksel yang termasuk area ROI. Penanda tersebut berperan sebagai pemetaan area dari citra digital yang termasuk ROI. Pada proses DWT pun, mask tersebut akan berubah, mengikuti transformasi yang dilakukan pada citra digital. Sehingga pada akhirnya akan diperoleh informasi mengenai koefisien yang termasuk ROI.

Bentuk lain atau area yang berbentuk tidak beraturan yang ingin dipisahkan dari image induk maka didefinisikan sebagai ROI (Region of Interest) dimana di MATLAB terdapat banyak sekali fungsi yang bisa digunakan, salah satunya yaitu `roipoly(I,c,r)` dimana `I` adalah matrik gambar, `c` adalah matrik titik kolom daerah yang menjadi ROI dan `r` adalah matrik titik baris daerah yang menjadi ROI. Proses ROI yang terjadi adalah dengan memilih area pada frame video. Proses ROI dalam penelitian ini digunakan untuk membatasi atau memperkecil area pemrosesan. Pembatasan area pemrosesan dilakukan dengan cara menentukan area pada video. Pembatasan area pada video dilakukan agar objek- objek yang berada diluar area tersebut tidak menjadi penambahan noise pada citra yang akan di proses. Tujuan daripada pengimplementasian ROI dalam penelitian ini adalah untuk meningkatkan waktu pemrosesan sehingga dapat diimplementasikan pada mini PC(Personal Computer) seperti Raspberry Pi dan Odroid UX4. Implementasi ROI di dalam penelitian ini terdapat dua tahapan, pertama menentukan titik penentuan untuk daerah ROI dan kedua mengaplikasikan ROI dalam program sistem, sehingga mampu mempercepat waktu pemrosesan.

ROI digunakan untuk membatasi area deteksi objek, video yang digunakan. Waktu pengambilan data dilakukan pada waktu pagi, siang, sore dan malam hari pada titik yang sama, dengan catatan pengambilan data dilakukan dalam kondisi cuaca cerah. Pengolahan data citra dilakukan dengan bahasa pemrograman Python. Pengujian dilakukan dengan menggunakan video.

## 2.9 Python

Python merupakan salah satu bahasa pemrograman yang banyak digunakan oleh perusahaan besar maupun para developer untuk mengembangkan berbagai macam aplikasi berbasis desktop, web dan mobile. Python diciptakan oleh Guido van Rossum di Belanda pada tahun 1990 dan namanya diambil dari acara televisikesejukan Guido Monty Python's Flying Circus. Van Rossum mengembangkan Python sebagai

hobi, kemudian Python menjadi bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintak intuitif dan memiliki pustaka yang luas (Schuerer & Maufrais 2010).Kepopuleran penggunaan Python menempatkannya menjadi bahasa pemrograman yang mulai banyak dipelajari oleh mahasiswa terutama mahasiswa di kampus yang berbasis IT, guna menyelesaikan tugas kuliah, tugas akhir maupun tugas penelitian, untuk dapat menyelesaikan berbagai tugas pemrograman, seseorang perlu memahami algoritma, karena pada dasarnya program computer adalah implementasi dari algoritma.

Menurut pengertian dari Python Software Foundation (2016), Python adalah bahasa pemrograman interpretatif, berorientasi objek dan semantik yang dinamis. Python memiliki high-level struktur data, dynamic typing dan dynamic binding. Python memiliki sintaks sederhana dan mudah dipelajari untuk penekanan pada kemudahan membaca dan mengurangi biaya perbaikan program. Python mendukung modul dan paket untuknya tersedia secara gratis untuk semua platform dan dapat secara bebas disebar.

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python bisa dibilang bahasa pemrograman dengan tujuan umum yang dikembangkan secara khusus untuk membuat source code mudah dibaca. Python juga memiliki library yang lengkap sehingga memungkinkan programmer untuk membuat aplikasi yang mutakhir dengan menggunakan source code yang tampak sederhana (Ljubomir Perkovic, 2012).

### **2.9.1 Python Imaging Library**

Python Imaging Library atau disingkat PIL adalah Library gratis untuk bahasa pemrograman python untuk menambahkan kemampuan pengolahan citra untuk interpreter python. Library ini mendukung banyak format file, dan memberikan kuat pengolahan gambar dan kemampuan grafis. Library ini sendiri tersedia untuk Windows, Mac OS X dan Linux.

### **2.9.2 Python Time Library**

Python Time Library adalah Library yang menyediakan fungsi yang berhubungan dengan waktu. Untuk fungsi terkait, lihat juga datetime dan calendar modules. Meski modul ini

selalu tersedia, tidak semua fungsi yang teredia pada semua platform. Sebagian besar fungsi yang di tetapkan dalam panggilan modul ini fungsi platform seperti Library C dengan nama yang sama. Kadang – kadang mungkin membantu untuk berkonsultasi dokumentasi platform, karena semantik fungsi-fungsi ini bervariasi antara platform.

### **2.9.3 Python Py2Exe**

py2exe adalah ekstensi Python yang mengubah Python script (py) ke Microsoft Windows executable (exe). executables ini dapat berjalan pada sistem tanpa Python diinstal. Ini adalah alat yang paling umum untuk melakukannya. py2exe digunakan untuk mendistribusikan resmi BitTorrent client (sebelum versi 6.0) dan masih digunakan untuk mendistribusikan SpamBayes serta proyekproyek lainnya. Sejak Mei 2014, ada versi py2exe tersedia untuk Python 3. Sebelum itu, py2exe dibuat hanya untuk Python 2, dan itu perlu untuk menggunakan alternatif seperti cx\_Freeze untuk Python 3 kode. Meskipun program ini mengubah file Py ke exe, itu tidak membuatnya berjalan lebih cepat sebagaipy2exe hanya bundel bytecode Python daripada mengubahnya ke mesin-kode. Bahkan mungkin berjalan lebih lambat daripada menggunakan interpreter Python langsung karena startup overhead.

## **2.10 Object Tracking**

Object tracking adalah salah satu metode pengolahan citra digital yang penting dalam bidang visi komputer [3]. Dalam perkembangan teknologi perangkat kamera dan semakin meningkatnya minat dan kebutuhan sistem analisa video otomatis dengan banyaknya menghasilkan algoritma object tracking. Penggunaan sistem object tracking sering dimanfaatkan untuk tugas-tugas yang berkaitan dengan, sistem keamanan, surveillance, pemantau lalu lintas, sistem navigasi dan interaksi manusia-komputer [4-8]. Untuk mendukung sistem keamanan yang membutuhkan pengawasan secara real time. Dari subjek pengawasan umumnya mengarah pada penjejakan dan analisa aktivitas manusia di lingkungan luar maupun sudut pandang di dalam ruang. Ada tiga langkah dalam membangun sebuah sistem pengawasan, yaitu mendeteksi objek yang ditentukan, menjejaki posisi objek setiap frame, dan menganalisa lintasan objek untuk mengenali prilakunya.

Untuk ruangan dengan ukuran besar maka dapat menggunakan kamera memiliki kemampuan bergerak dengan sistem mekanik pan and tilt. Pan-tilt camera ini dapat bergerak rotasi secara horisontal dan vertikal. Dalam aplikasinya, untuk memantau ruangan atau objek, kamera dapat digerakkan secara manual oleh petugas keamanan.

Dengan bantuan visualisasi di layar monitor, objek dapat diikuti pergerakannya dengan menggunakan pengendali jarak jauh tanpa harus ke lokasi. Pada penelitian yang telah dibuat oleh Hu, merancang sistem pendeteksi gerakan menggunakan teknik pengolahan citra digital, namun pada perancangan tersebut hanya dapat mendeteksi keberadaan suatu benda yang bersifat spesifik, ini kurang efisien jika digunakan untuk pemantauan ruangan sebagai sistem pengawas terhadap pergerakan objek. Pada implementasinya penggunaan perangkat kamera statis kurang efisien sebagai alat pemantauan karena tidak didukung dengan sistem otomatis untuk mengikuti pergerakan. Adapun implementasi lain dari Yunardi yang merancang pengendalian kamera webcam untuk aplikasi sistem robotika untuk mendeteksi objek dengan warna permukaan objek yang spesifik. Bhowmik membahas sistem object tracking dengan menggunakan kamera yang diaplikasikan pada virtual reality. Sistem yang telah dibuat dilengkapi dengan joystick untuk menggerakkan arah hadap pandangan kamera untuk memilih target yang akan dipantau. Dari hasil yang dilakukan terbatas pada mengikuti posisi obyek tersebut dengan menggunakan kendali yang diperintah oleh user. Pada sebuah sistem robot penyelamat, Setyawan menggunakan kamera untuk menemukan korban di tempat terjadi bencana.

## **2.11 Penelitian Terkait**

Penelitian yang dilakukan oleh Awang Hendrianto Pratomo, Wilis Kaswidjanti dan Siti Muarifah pada tahun 2020 bahwa Penerapan proses ROI dapat digunakan sebagai solusi untuk mengatasi masalah waktu pemrosesan. Hal ini dibuktikan dengan pengujian yang dilakukan sebelum dan setelah pengimplementasian proses ROI. Hasil pengujian waktu pemrosesan menggunakan laptop sebelum pengimplementasian ROI memiliki waktu pemrosesan sebesar 35 ms sedangkan setelah diimplementasikannya proses ROI memiliki waktu pemrosesan sebesar 9,5 ms. Pada pengujian yang dilakukan menggunakan Raspberry Pi3 sebelum diimplementasikannya proses ROI memiliki waktu pemrosesan yaitu sebesar 248 ms sedangkan setelah diimplementasikannya proses ROI.

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **3.1 Penentuan Region Of Interest Untuk Menghitung Jumlah Kendaraan Pada Jalan Raya Menggunakan Frame Substraction**

Pada bab tiga ini akan dibahas perancangan Penentuan Region Of Interest Pada Object Tracking Menggunakan Frame Substraction yang merupakan pokok pembahasan dari Skripsi ini.

##### **3.1.1 Analisis Kebutuhan Sistem**

Dalam pembuatan program Penentuan Region Of Interest Untuk Menghitung Jumlah Kendaraan Pada Jalan Raya Menggunakan Frame Substraction yang telah dirancang, dibutuhkan perangkat keras (*hardware*), perangkat lunak (*software*) dan perangkat operator (*brainware*). Adapun *hardware*, *software* dan *brainware* yang dibutuhkan adalah sebagai berikut :

###### 1. *Hardware* (Perangkat Keras)

Spesifikasi perangkat keras yang digunakan untuk membangun aplikasi ini sebagai berikut :

- a. Processor Intel Core i5
- b. Ram 2 GHz
- c. Harddisk 500 Gb
- d. Laptop Acer 14 Inch
- e. Mouse

###### 2. Perangkat Lunak (*Software*)

Spesifikasi perangkat lunak yang digunakan untuk membangun aplikasi ini sebagai berikut :

- a. Sistem Operasi *Windows 7 Professional Ultimate*
- b. Bahasa Pemrograman *Python, Anaconda dan Open CV*.

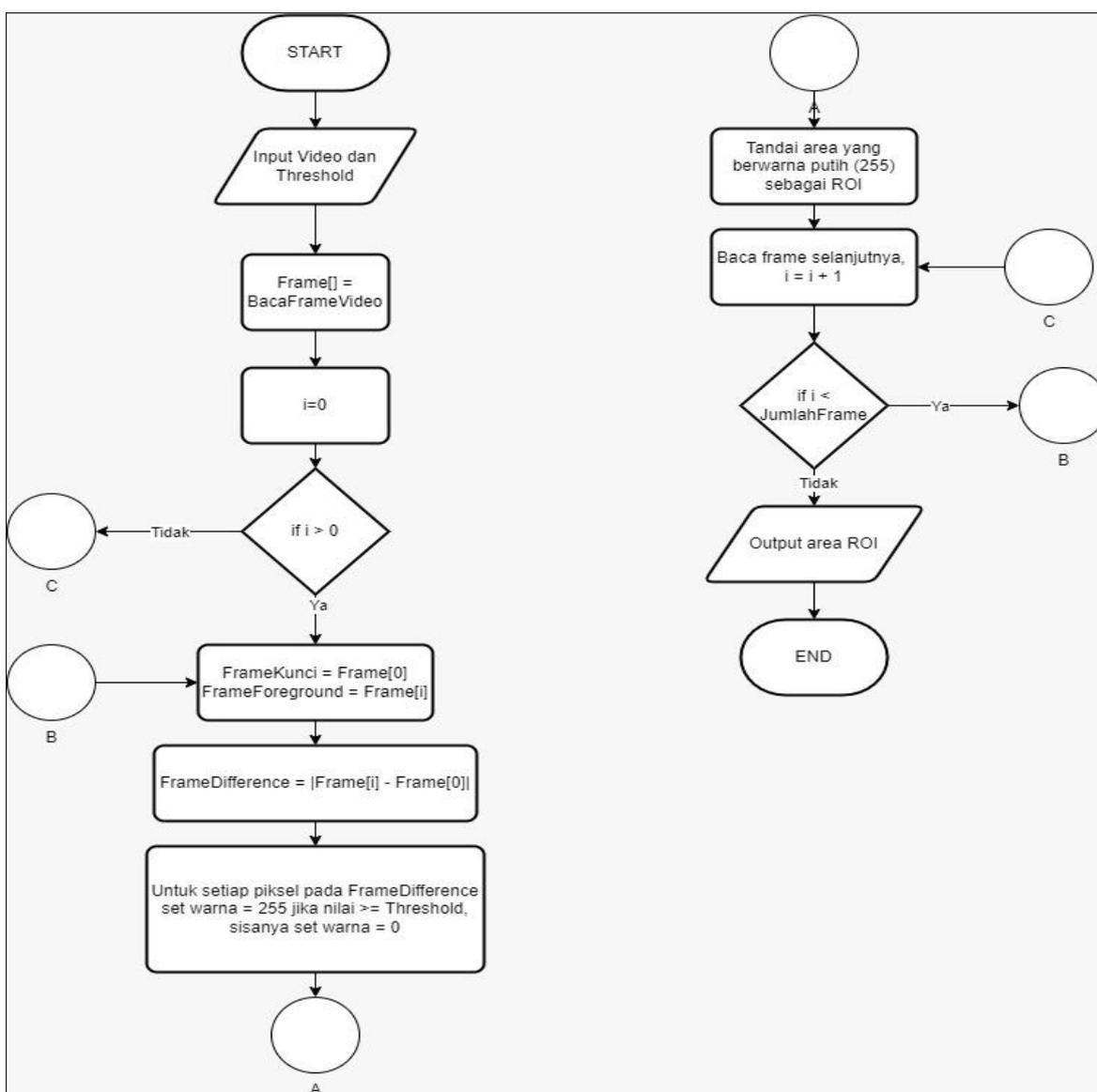
Perangkat Pemikir (*Brainware*), yaitu orang yang mengerti sistem operasi komputer yang memiliki keahlian yang baik untuk menjalankan program.

#### **3.2 Perancangan Sistem**

Perancangan sistem yang akan dilakukan pada penelitian ini terbagi menjadi dua tahap, yaitu perancangan *flowchart diagram* bertujuan untuk memberikan gambaran mengenai proses dan alur penggunaan dari sistem yang dikembangkan. Sedangkan perancangan antarmuka dilakukan untuk memberikan gambaran tampilan antar muka dari sistem yang dikembangkan.

### 3.2.1 Perancangan *Flowchart*

Flowchart adalah penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafis dari langkah – langkah dan urutan – urutan prosedur dari suatu program. Flowchart menolong analis dan pemogrammer untuk memecahkan masalah kedalam segmen – segmen yang lebih kecil dan menolong dalam menganalisis alternatif – alternatif lain dalam mengoprasian. Adapun *flowchart* proses pengenalan menggunakan isodata *clustering* dapat dilihat pada gambar berikut.



**Gambar 3.1** Diagram *Flowchart* Sistem Keseluruhan

Rancangan diagram alir seperti yang terlihat pada gambar 3.5 memperlihatkan aliran proses pelatihan yang dilakukan oleh sistem.



Perhitungan statistik area dan intensitas untuk ROI umum yang terdiri dari segmen garis lurus adalah sebagai berikut:

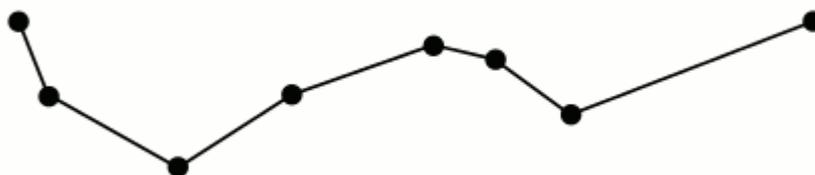
1. Setiap baris gambar dipertimbangkan secara bergantian.
2. Jika baris memotong ROI di mana saja, maka setiap piksel dalam baris gambar tersebut akan dipertimbangkan secara bergiliran.
3. Setiap piksel berpotongan dengan ROI.
4. Total area ROI adalah jumlah area dari semua bentuk yang berpotongan. Untuk ROI Teks, Penanda, dan Garis, areanya adalah nol.
5. Biarkan  $I_i$  menjadi intensitas piksel  $i$  dengan area berpotongan bukan nol; biarkan  $a_i$  menjadi area berpotongan dari piksel  $i$ ; dan biarkan  $A$  menjadi luas total dari semua bentuk yang berpotongan (yaitu,  $\sum(a_i)$ ).
6. Intensitas Pixel rata-rata adalah  $\sum(a_i \times I_i) / A$ .
7. Untuk ROI Teks atau Penanda, rata-rata intensitas piksel adalah intensitas piksel di tengah teks atau di bawah penanda. Misalkan  $\mu$  adalah intensitas piksel rata-rata.
8. Deviasi standar dalam intensitas piksel adalah:  $\sqrt{((\sum(a_i \times I_i \times I_i) - \mu \times \mu \times A) / A)}$ .  
Pengecualian untuk hal ini adalah untuk ROI Teks dan Penanda, dengan deviasi standar dalam intensitas piksel selalu nol.
9. Intensitas piksel minimum adalah intensitas minimum dalam setiap bentuk berpotongan dengan area lebih besar dari nol. Pengecualian untuk hal ini adalah untuk ROI Teks dan Penanda, dengan intensitas piksel minimum adalah intensitas piksel di tengah teks atau di bawah penanda.
10. Intensitas piksel minimum adalah intensitas minimum dalam setiap bentuk berpotongan dengan area lebih besar dari nol. Pengecualian untuk hal ini adalah untuk ROI Teks dan Penanda, dengan intensitas piksel minimum adalah intensitas piksel di tengah teks atau di bawah penanda.

### 3.3.1 OpenCV Contour Approximation

Pendekatan Kontur OpenCV atau lebih tepat disebut dengan Algoritma Ramer-Douglas-Peucker. Algoritma Douglas-Peucker adalah algoritma yang dapat digunakan untuk optimasi saat pengolahan citra. Menggunakan fungsi `findContours` OpenCV, kita dapat memilih semua kemungkinan kontur pada gambar yang diberikan. Dalam penentuan ROI pada video yg diambil berfungsi sebagai penentuan garis atau warna dalam

penghasilan ROI yang dilakukan. Waktu berjalan algoritma ini ketika dijalankan pada polyline yang terdiri dari  $n - 1$  segmen dan  $n$  simpul diberikan oleh perulangan  $T(n) = T(i + 1) + T(n - i) + O(n)$  dimana  $i = 1, 2, \dots, n - 2$  adalah nilai indeks pada pseudocode.

Dimulai dari titik awal dan akhir dari sebuah kurva, pertama-tama algoritma akan menemukan simpul pada jarak maksimum dari garis yang menghubungkan dua titik referensi. Kita sebut sebagai  $\text{max\_point}$ . Jika  $\text{max\_point}$  terletak pada jarak kurang dari ambang batas, kami secara otomatis mengabaikan semua simpul antara titik awal dan akhir dan membuat kurva menjadi garis lurus.



**Gambar 3.4** Kurva Area ROI

### 3.3.2. Intensitas Kontur

Cara intensitas (level) kontur ditentukan dari piksel awal dijelaskan di sini. Piksel awal. Pixel awal pada lokasi  $(x, y)$  ditunjukkan di bawah ini, dan memiliki intensitas  $X_{x, y}$ .

**Tabel 3.1** Intensitas Kontur ROI

	$X_{x-x,y-1}$	$X_{x,y-1}$	$X_{x+1,y-1}$
	$X_{x+1,y-1}$	$X_{x,y}$	$X_{x+1,y}$
	$X_{x-x,y+1}$	$X_{x,y+1}$	

Jika pusat piksel dianggap sebagai lokasi nilai intensitas, maka intensitas interpolasi pada keempat sudut piksel adalah tingkat kontur diatur ke rata-rata intensitas empat sudut.

**Tabel 3.2** Algoritma Intensitas Sudut ROI

Intensitas Sudut 1	$(Xx-1, y-1 + Xx, y-1 + Xx, y + Xx-1, y) / 4$
Intensitas Sudut 2	$(Xx, y-1 + Xx+1, y-1 + Xx+1, y + Xx, y) / 4$
Intensitas Sudut 3	$(Xx, y + Xx+1, y + Xx+1, y+1 + Xx, y+1) / 4$
Intensitas Sudut 4	$(Xx-1, y + Xx, y + Xx, y+1 + Xx-1, y+1) / 4$

### 3.4 Analisis Metode Frame Substraction

Pada penelitian tugas akhir ini dilakukan pencarian *Region of Interest* dari citra yang diperoleh dari frame video menggunakan Teknik frame subtraction. Secara garis besar, pencarian ROI menggunakan frame subtraction adalah pencarian area pada frame atau citra yang memiliki perbedaan seiring berjalannya waktu sehingga proses – proses seperti object detection atau object tracking dapat lebih efisien dengan hanya berfokus pada area tersebut (ROI).

Sebagai ilustrasi diperoleh terdapat frame dari video sebagai berikut :

**Gambar 3.5** Frame Video

Dengan menggunakan persamaan frame subtraction yaitu :

$$S(n) = |F(n) - B| > T$$

Maka dibutuhkan beberapa variable seperti frame background, frame foreground dan Threshold. Sebagai ilustrasi threshold atau nilai ambang yang digunakan adalah 128. Sedangkan frame background yang digunakan adalah gambar pertama sedangkan frame foreground yang digunakan

adalah gambar kedua. Sehingga nilai S(5) dapat dihitung dengan menggunakan tahapan berikut :

1. Membaca nilai piksel dari frame background, yang dalam ilustrasi ini menggunakan frame pada t=0.



**Gambar 3.6** Frame Background

**Tabel 3.3** Nilai Piksel Grayscale Frame Background

228	230	231	231	231	232	232	232	233	234	236	234	229	241	235	235	240	240	242	242	244	245	246	246	246	245	245	245	243	239
223	224	225	225	225	226	228	227	228	228	233	208	183	219	197	194	240	239	242	244	244	244	242	241	241	240	239	239	237	233
215	216	217	218	222	221	221	222	224	223	224	225	231	214	228	234	233	236	239	240	239	239	238	237	234	231	230	228	225	223
223	224	224	227	228	230	229	230	231	234	233	236	237	218	232	234	232	230	225	226	226	224	224	224	227	226	222	221	217	220
189	191	192	190	196	185	190	196	203	195	206	193	202	195	184	195	189	191	199	190	192	206	208	206	183	182	197	200	208	195
81	84	85	74	70	68	92	115	101	91	109	124	135	134	123	116	124	135	88	73	75	84	110	145	76	63	79	99	111	103
91	101	100	84	84	85	81	95	177	194	179	171	157	135	118	129	141	180	143	67	85	97	107	140	82	79	64	108	152	129
93	95	88	75	75	105	150	173	175	149	128	127	131	109	107	134	133	141	185	193	118	109	139	136	151	145	124	119	151	119
115	132	149	174	190	189	164	132	120	121	126	131	131	123	112	125	117	122	125	159	181	169	185	130	122	173	170	173	166	155
211	228	228	199	149	112	109	118	122	122	122	128	115	125	109	115	116	117	124	116	134	192	210	158	76	131	166	181	198	207
219	183	130	99	103	112	117	118	119	118	120	125	102	136	107	115	121	120	116	119	117	106	146	206	202	189	197	203	190	186
117	94	105	110	108	112	113	117	113	113	124	110	115	145	97	109	116	118	115	112	111	119	111	113	167	209	208	190	184	186
105	111	106	107	108	110	113	109	110	117	127	99	112	135	97	105	109	120	116	111	114	107	114	118	101	120	176	211	197	188
106	102	112	112	108	112	116	109	109	119	120	90	112	136	99	99	103	111	114	106	109	109	103	108	118	109	99	143	200	197
105	107	105	107	109	109	108	108	112	122	114	83	99	123	97	99	104	106	114	114	104	108	109	107	104	116	114	97	112	139
105	105	108	107	108	107	107	104	120	120	102	83	93	104	98	92	106	102	110	115	104	103	108	111	101	102	112	121	105	103

2. Membaca nilai piksel dari frame foreground, yang dalam ilustrasi ini menggunakan frame t = 5.



**Gambar 3.7** Frame Foreground

**Tabel 3.4** Nilai Pixel Grayscale Frame Foreground

223	225	226	226	227	228	229	230	230	231	233	231	184	207	225	184	226	242	240	242	244	244	244	243	243	243	243	241	238	
215	218	218	219	221	223	222	223	223	225	226	225	219	217	214	220	230	239	241	244	244	243	242	240	239	237	237	235	234	231
218	220	220	222	223	225	225	226	226	228	227	230	234	225	220	236	230	229	230	229	229	228	228	227	226	227	223	220	218	217
209	210	216	214	218	220	216	219	227	228	234	232	231	226	211	227	224	222	222	223	222	225	224	222	223	217	224	224	226	227
93	101	103	101	94	93	97	129	129	108	116	125	130	151	136	134	133	138	136	124	113	132	150	165	147	99	123	130	151	146
76	89	96	86	77	78	82	86	120	161	154	168	163	148	132	124	133	159	141	55	67	68	71	130	114	51	64	70	119	121
95	101	103	82	77	78	105	132	181	192	161	143	137	127	101	123	135	144	191	162	110	108	135	124	130	106	102	84	139	162
103	108	116	131	146	169	188	172	141	122	123	128	134	126	110	124	129	131	134	180	196	114	145	156	146	176	170	157	152	141
188	205	221	227	204	155	120	113	120	123	124	128	125	125	117	113	125	128	126	115	138	204	218	169	77	113	160	167	185	195
227	221	187	132	105	104	114	120	119	122	119	130	110	122	121	110	118	122	125	123	132	128	163	212	169	134	177	193	196	193
174	114	98	105	111	113	114	115	119	112	119	123	104	140	117	106	117	122	106	112	111	129	109	131	196	223	205	196	193	188
96	107	111	106	107	110	111	113	111	113	126	116	100	144	108	102	111	122	90	69	101	99	119	114	107	143	200	211	188	184
114	106	105	116	109	110	117	111	110	114	127	103	99	136	116	98	107	113	117	100	114	114	105	113	119	105	110	169	212	208
103	107	106	107	109	110	112	110	108	119	125	95	94	126	113	93	108	104	115	115	108	110	109	105	107	119	109	98	135	177
107	106	105	107	107	110	107	107	111	125	116	88	90	106	107	91	107	104	111	119	111	105	109	116	104	105	116	116	100	106
102	106	108	107	111	107	107	103	119	124	107	82	91	95	106	88	104	102	108	111	107	101	104	107	103	102	104	113	121	107

## 3. Menghitung nilai (S) :

Tahapan selanjutnya adalah menghitung nilai selisih antara piksel citra background dengan piksel citra foreground :

**Tabel 3.5** Perhitungan selisih pixel background dan foreground

$P(0,0) =  228-223 =5$	$P(3,7) =  75-131 =56$	$P(6,14) =  108-107 =1$
$P(0,1) =  223-215 =8$	$P(3,8) =  174-227 =53$	$P(6,15) =  107-107 =0$
$P(0,2) =  215-218 =3$	$P(3,9) =  199-132 =67$	$P(7,0) =  232-230 =2$
$P(0,3) =  223-209 =14$	$P(3,10) =  99-105 =6$	$P(7,1) =  227-223 =4$
$P(0,4) =  189-93 =96$	$P(3,11) =  110-106 =4$	$P(7,2) =  222-226 =4$
$P(0,5) =  81-76 =5$	$P(3,12) =  107-116 =9$	$P(7,3) =  230-219 =11$
$P(0,6) =  91-95 =4$	$P(3,13) =  112-107 =5$	$P(7,4) =  196-129 =67$
$P(0,7) =  93-103 =10$	$P(3,14) =  107-107 =0$	$P(7,5) =  115-86 =29$
$P(0,8) =  115-188 =73$	$P(3,15) =  107-107 =0$	$P(7,6) =  95-132 =37$
$P(0,9) =  211-227 =16$	$P(4,0) =  231-227 =4$	$P(7,7) =  173-172 =1$
$P(0,10) =  219-174 =45$	$P(4,1) =  225-221 =4$	$P(7,8) =  132-113 =19$
$P(0,11) =  117-96 =21$	$P(4,2) =  222-223 =1$	$P(7,9) =  118-120 =2$
$P(0,12) =  105-114 =9$	$P(4,3) =  228-218 =10$	$P(7,10) =  118-115 =3$
$P(0,13) =  106-103 =3$	$P(4,4) =  196-94 =102$	$P(7,11) =  117-113 =4$
$P(0,14) =  105-107 =2$	$P(4,5) =  70-77 =7$	$P(7,12) =  109-111 =2$
$P(0,15) =  105-102 =3$	$P(4,6) =  84-77 =7$	$P(7,13) =  109-110 =1$
$P(1,0) =  230-225 =5$	$P(4,7) =  75-146 =71$	$P(7,14) =  108-107 =1$
$P(1,1) =  224-218 =6$	$P(4,8) =  190-204 =14$	$P(7,15) =  104-103 =1$

**Lanjutan Tabel 3.5**  
Perhitungan selisih pixel background dan foreground

$P(1,2) =  216-220 =4$	$P(4,9) =  149-105 =44$	$P(8,0) =  233-230 =3$
$P(1,3) =  224-210 =14$	$P(4,10) =  103-111 =8$	$P(8,1) =  228-223 =5$
$P(1,4) =  191-101 =90$	$P(4,11) =  108-107 =1$	$P(8,2) =  224-226 =2$
$P(1,5) =  84-89 =5$	$P(4,12) =  108-109 =1$	$P(8,3) =  231-227 =4$
$P(1,6) =  101-101 =0$	$P(4,13) =  108-109 =1$	$P(8,4) =  203-129 =74$
$P(1,7) =  95-108 =13$	$P(4,14) =  109-107 =2$	$P(8,5) =  101-120 =19$
$P(1,8) =  132-205 =73$	$P(4,15) =  108-111 =3$	$P(8,6) =  177-181 =4$
$P(1,9) =  228-221 =7$	$P(5,0) =  232-228 =4$	$P(8,7) =  175-141 =34$
$P(1,10) =  183-114 =69$	$P(5,1) =  226-223 =3$	$P(8,8) =  120-120 =0$
$P(1,11) =  94-107 =13$	$P(5,2) =  221-225 =4$	$P(8,9) =  122-119 =3$
$P(1,12) =  111-106 =5$	$P(5,3) =  230-220 =10$	$P(8,10) =  119-119 =0$
$P(1,13) =  102-107 =5$	$P(5,4) =  185-93 =92$	$P(8,11) =  113-111 =2$
$P(1,14) =  107-106 =1$	$P(5,5) =  68-78 =10$	$P(8,12) =  110-110 =0$
$P(1,15) =  105-106 =1$	$P(5,6) =  85-78 =7$	$P(8,13) =  109-108 =1$
$P(2,0) =  231-226 =5$	$P(5,7) =  105-169 =64$	$P(8,14) =  112-111 =1$
$P(2,1) =  225-218 =7$	$P(5,8) =  189-155 =34$	$P(8,15) =  120-119 =1$
$P(2,2) =  217-220 =3$	$P(5,9) =  112-104 =8$	$P(9,0) =  234-231 =3$
$P(2,3) =  224-216 =8$	$P(5,10) =  112-113 =1$	$P(9,1) =  228-225 =3$
$P(2,4) =  192-103 =89$	$P(5,11) =  112-110 =2$	$P(9,2) =  223-228 =5$
$P(2,5) =  85-96 =11$	$P(5,12) =  110-110 =0$	$P(9,3) =  234-228 =6$
$P(2,6) =  100-103 =3$	$P(5,13) =  112-110 =2$	$P(9,4) =  195-108 =87$
$P(2,7) =  88-116 =28$	$P(5,14) =  109-110 =1$	$P(9,5) =  91-161 =70$
$P(2,8) =  149-221 =72$	$P(5,15) =  107-107 =0$	$P(9,6) =  194-192 =2$
$P(2,9) =  228-187 =41$	$P(6,0) =  232-229 =3$	$P(9,7) =  149-122 =27$
$P(2,10) =  130-98 =32$	$P(6,1) =  228-222 =6$	$P(9,8) =  121-123 =2$
$P(2,11) =  105-111 =6$	$P(6,2) =  221-225 =4$	$P(9,9) =  122-122 =0$
$P(2,12) =  106-105 =1$	$P(6,3) =  229-216 =13$	$P(9,10) =  118-112 =6$
$P(2,13) =  112-106 =6$	$P(6,4) =  190-97 =93$	$P(9,11) =  113-113 =0$
$P(2,14) =  105-105 =0$	$P(6,5) =  92-82 =10$	$P(9,12) =  117-114 =3$
$P(2,15) =  108-108 =0$	$P(6,6) =  81-105 =24$	$P(9,13) =  119-119 =0$
$P(3,0) =  231-226 =5$	$P(6,7) =  150-188 =38$	$P(9,14) =  122-125 =3$
$P(3,1) =  225-219 =6$	$P(6,8) =  164-120 =44$	$P(9,15) =  120-124 =4$
$P(3,2) =  218-222 =4$	$P(6,9) =  109-114 =5$	$P(10,0) =  236-233 =3$
$P(3,3) =  227-214 =13$	$P(6,10) =  117-114 =3$	$P(10,1) =  233-226 =7$
$P(3,4) =  190-101 =89$	$P(6,11) =  113-111 =2$	$P(10,2) =  224-227 =3$
$P(3,5) =  74-86 =12$	$P(6,12) =  113-117 =4$	$P(10,3) =  233-234 =1$
$P(3,6) =  84-82 =2$	$P(6,13) =  116-112 =4$	$P(10,4) =  206-116 =90$
$P(10,5) =  109-154 =45$	$P(13,12) =  135-136 =1$	$P(17,3) =  230-222 =8$
$P(10,6) =  179-161 =18$	$P(13,13) =  136-126 =10$	$P(17,4) =  191-138 =53$

**Lanjutan Tabel 3.5**  
Perhitungan selisih pixel background dan foreground

$P(10,7) =  128-123 =5$	$P(13,14) =  123-106 =17$	$P(17,5) =  135-159 =24$
$P(10,8) =  126-124 =2$	$P(13,15) =  104-95 =9$	$P(17,6) =  180-144 =36$
$P(10,9) =  122-119 =3$	$P(14,0) =  235-225 =10$	$P(17,7) =  141-131 =10$
$P(10,10) =  120-119 =1$	$P(14,1) =  197-214 =17$	$P(17,8) =  122-128 =6$
$P(10,11) =  124-126 =2$	$P(14,2) =  228-220 =8$	$P(17,9) =  117-122 =5$
$P(10,12) =  127-127 =0$	$P(14,3) =  232-211 =21$	$P(17,10) =  120-122 =2$
$P(10,13) =  120-125 =5$	$P(14,4) =  184-136 =48$	$P(17,11) =  118-122 =4$
$P(10,14) =  114-116 =2$	$P(14,5) =  123-132 =9$	$P(17,12) =  120-113 =7$
$P(10,15) =  102-107 =5$	$P(14,6) =  118-101 =17$	$P(17,13) =  111-104 =7$
$P(11,0) =  234-231 =3$	$P(14,7) =  107-110 =3$	$P(17,14) =  106-104 =2$
$P(11,1) =  208-225 =17$	$P(14,8) =  112-117 =5$	$P(17,15) =  102-102 =0$
$P(11,2) =  225-230 =5$	$P(14,9) =  109-121 =12$	$P(18,0) =  242-240 =2$
$P(11,3) =  236-232 =4$	$P(14,10) =  107-117 =10$	$P(18,1) =  242-241 =1$
$P(11,4) =  193-125 =68$	$P(14,11) =  97-108 =11$	$P(18,2) =  239-230 =9$
$P(11,5) =  124-168 =44$	$P(14,12) =  97-116 =19$	$P(18,3) =  225-222 =3$
$P(11,6) =  171-143 =28$	$P(14,13) =  99-113 =14$	$P(18,4) =  199-136 =63$
$P(11,7) =  127-128 =1$	$P(14,14) =  97-107 =10$	$P(18,5) =  88-141 =53$
$P(11,8) =  131-128 =3$	$P(14,15) =  98-106 =8$	$P(18,6) =  143-191 =48$
$P(11,9) =  128-130 =2$	$P(15,0) =  235-184 =51$	$P(18,7) =  185-134 =51$
$P(11,10) =  125-123 =2$	$P(15,1) =  194-220 =26$	$P(18,8) =  125-126 =1$
$P(11,11) =  110-116 =6$	$P(15,2) =  234-236 =2$	$P(18,9) =  124-125 =1$
$P(11,12) =  99-103 =4$	$P(15,3) =  234-227 =7$	$P(18,10) =  116-106 =10$
$P(11,13) =  90-95 =5$	$P(15,4) =  195-134 =61$	$P(18,11) =  115-90 =25$
$P(11,14) =  83-88 =5$	$P(15,5) =  116-124 =8$	$P(18,12) =  116-117 =1$
$P(11,15) =  83-82 =1$	$P(15,6) =  129-123 =6$	$P(18,13) =  114-115 =1$
$P(12,0) =  229-184 =45$	$P(15,7) =  134-124 =10$	$P(18,14) =  114-111 =3$
$P(12,1) =  183-219 =36$	$P(15,8) =  125-113 =12$	$P(18,15) =  110-108 =2$
$P(12,2) =  231-234 =3$	$P(15,9) =  115-110 =5$	$P(19,0) =  242-242 =0$
$P(12,3) =  237-231 =6$	$P(15,10) =  115-106 =9$	$P(19,1) =  244-244 =0$
$P(12,4) =  202-130 =72$	$P(15,11) =  109-102 =7$	$P(19,2) =  240-229 =11$
$P(12,5) =  135-163 =28$	$P(15,12) =  105-98 =7$	$P(19,3) =  226-223 =3$
$P(12,6) =  157-137 =20$	$P(15,13) =  99-93 =6$	$P(19,4) =  190-124 =66$
$P(12,7) =  131-134 =3$	$P(15,14) =  99-91 =8$	$P(19,5) =  73-55 =18$
$P(12,8) =  131-125 =6$	$P(15,15) =  92-88 =4$	$P(19,6) =  67-162 =95$
$P(12,9) =  115-110 =5$	$P(16,0) =  240-226 =14$	$P(19,7) =  193-180 =13$
$P(12,10) =  102-104 =2$	$P(16,1) =  240-230 =10$	$P(19,8) =  159-115 =44$
$P(12,11) =  115-100 =15$	$P(16,2) =  233-230 =3$	$P(19,9) =  116-123 =7$
$P(12,12) =  112-99 =13$	$P(16,3) =  232-224 =8$	$P(19,10) =  119-112 =7$
$P(12,13) =  112-94 =18$	$P(16,4) =  189-133 =56$	$P(19,11) =  112-69 =43$
$P(12,14) =  99-90 =9$	$P(16,5) =  124-133 =9$	$P(19,12) =  111-100 =11$

**Lanjutan Tabel 3.5**  
Perhitungan selisih pixel background dan foreground

$P(12,15) =  93-91 =2$	$P(16,6) =  141-135 =6$	$P(19,13) =  106-115 =9$
$P(13,0) =  241-207 =34$	$P(16,7) =  133-129 =4$	$P(19,14) =  114-119 =5$
$P(13,1) =  219-217 =2$	$P(16,8) =  117-125 =8$	$P(19,15) =  115-111 =4$
$P(13,2) =  214-225 =11$	$P(16,9) =  116-118 =2$	$P(20,0) =  244-244 =0$
$P(13,3) =  218-226 =8$	$P(16,10) =  121-117 =4$	$P(20,1) =  244-244 =0$
$P(13,5) =  134-148 =14$	$P(16,12) =  109-107 =2$	$P(20,3) =  226-222 =4$
$P(13,6) =  135-127 =8$	$P(16,13) =  103-108 =5$	$P(20,4) =  192-113 =79$
$P(13,7) =  109-126 =17$	$P(16,14) =  104-107 =3$	$P(20,5) =  75-67 =8$
$P(13,8) =  123-125 =2$	$P(16,15) =  106-104 =2$	$P(20,6) =  85-110 =25$
$P(13,9) =  125-122 =3$	$P(17,0) =  240-242 =2$	$P(20,7) =  118-196 =78$
$P(13,10) =  136-140 =4$	$P(17,1) =  239-239 =0$	$P(20,8) =  181-138 =43$
$P(13,11) =  145-144 =1$	$P(17,2) =  236-229 =7$	$P(20,9) =  134-132 =2$
$P(20,10) =  117-111 =6$	$P(24,1) =  241-239 =2$	$P(27,8) =  173-167 =6$
$P(20,11) =  111-101 =10$	$P(24,2) =  234-226 =8$	$P(27,9) =  181-193 =12$
$P(20,12) =  114-114 =0$	$P(24,3) =  227-223 =4$	$P(27,10) =  203-196 =7$
$P(20,13) =  109-108 =1$	$P(24,4) =  183-147 =36$	$P(27,11) =  190-211 =21$
$P(20,14) =  104-111 =7$	$P(24,5) =  76-114 =38$	$P(27,12) =  211-169 =42$
$P(20,15) =  104-107 =3$	$P(24,6) =  82-130 =48$	$P(27,13) =  143-98 =45$
$P(21,0) =  245-244 =1$	$P(24,7) =  151-146 =5$	$P(27,14) =  97-116 =19$
$P(21,1) =  244-243 =1$	$P(24,8) =  122-77 =45$	$P(27,15) =  121-113 =8$
$P(21,2) =  239-228 =11$	$P(24,9) =  76-169 =93$	$P(28,0) =  243-241 =2$
$P(21,3) =  224-225 =1$	$P(24,10) =  202-196 =6$	$P(28,1) =  237-234 =3$
$P(21,4) =  206-132 =74$	$P(24,11) =  167-107 =60$	$P(28,2) =  225-218 =7$
$P(21,5) =  84-68 =16$	$P(24,12) =  101-119 =18$	$P(28,3) =  217-226 =9$
$P(21,6) =  97-108 =11$	$P(24,13) =  118-107 =11$	$P(28,4) =  208-151 =57$
$P(21,7) =  109-114 =5$	$P(24,14) =  104-104 =0$	$P(28,5) =  111-119 =8$
$P(21,8) =  169-204 =35$	$P(24,15) =  101-103 =2$	$P(28,6) =  152-139 =13$
$P(21,9) =  192-128 =64$	$P(25,0) =  245-243 =2$	$P(28,7) =  151-152 =1$
$P(21,10) =  106-129 =23$	$P(25,1) =  240-237 =3$	$P(28,8) =  166-185 =19$
$P(21,11) =  119-99 =20$	$P(25,2) =  231-227 =4$	$P(28,9) =  198-196 =2$
$P(21,12) =  107-114 =7$	$P(25,3) =  226-217 =9$	$P(28,10) =  190-193 =3$
$P(21,13) =  109-110 =1$	$P(25,4) =  182-99 =83$	$P(28,11) =  184-188 =4$
$P(21,14) =  108-105 =3$	$P(25,5) =  63-51 =12$	$P(28,12) =  197-212 =15$
$P(21,15) =  103-101 =2$	$P(25,6) =  79-106 =27$	$P(28,13) =  200-135 =65$
$P(22,0) =  246-244 =2$	$P(25,7) =  145-176 =31$	$P(28,14) =  112-100 =12$
$P(22,1) =  242-242 =0$	$P(25,8) =  173-113 =60$	$P(28,15) =  105-121 =16$
$P(22,2) =  238-228 =10$	$P(25,9) =  131-134 =3$	$P(29,0) =  239-238 =1$
$P(22,3) =  224-224 =0$	$P(25,10) =  189-223 =34$	$P(29,1) =  233-231 =2$
$P(22,4) =  208-150 =58$	$P(25,11) =  209-143 =66$	$P(29,2) =  223-217 =6$
$P(22,5) =  110-71 =39$	$P(25,12) =  120-105 =15$	$P(29,3) =  220-227 =7$
$P(22,6) =  107-135 =28$	$P(25,13) =  109-119 =10$	$P(29,4) =  195-146 =49$
$P(22,7) =  139-145 =6$	$P(25,14) =  116-105 =11$	$P(29,5) =  103-121 =18$

**Lanjutan Tabel 3.5**  
Perhitungan selisih pixel background dan foreground

$P(22,8) =  185-218 =33$	$P(25,15) =  102-12 =0$	$P(29,6) =  129-162 =33$
$P(22,9) =  210-163 =47$	$P(26,0) =  245-243 =2$	$P(29,7) =  119-141 =22$
$P(22,10) =  146-109 =37$	$P(26,1) =  239-237 =2$	$P(29,8) =  155-195 =40$
$P(22,11) =  111-119 =8$	$P(26,2) =  230-223 =7$	$P(29,9) =  207-193 =14$
$P(22,12) =  114-105 =9$	$P(26,3) =  222-224 =2$	$P(29,10) =  186-188 =2$
$P(22,13) =  103-109 =6$	$P(26,4) =  197-123 =74$	$P(29,11) =  186-184 =2$
$P(22,14) =  109-109 =0$	$P(26,5) =  79-64 =15$	$P(29,12) =  188-208 =20$
$P(22,15) =  108-104 =4$	$P(26,6) =  64-102 =38$	$P(29,13) =  197-177 =20$
$P(23,0) =  246-244 =2$	$P(26,7) =  124-170 =46$	$P(29,14) =  139-106 =33$
$P(23,1) =  241-240 =1$	$P(26,8) =  170-160 =10$	$P(29,15) =  103-107 =4$
$P(23,2) =  237-227 =10$	$P(26,9) =  166-177 =11$	
$P(23,3) =  224-222 =2$	$P(26,10) =  197-205 =8$	
$P(23,4) =  206-165 =41$	$P(26,11) =  208-200 =8$	
$P(23,5) =  145-130 =15$	$P(26,12) =  176-110 =66$	
$P(23,6) =  140-124 =16$	$P(26,13) =  99-109 =10$	
$P(23,7) =  136-156 =20$	$P(26,14) =  114-116 =2$	
$P(23,8) =  130-169 =39$	$P(26,15) =  112-104 =8$	
$P(23,9) =  158-212 =54$	$P(27,0) =  245-243 =2$	
$P(23,10) =  206-131 =75$	$P(27,1) =  239-235 =4$	
$P(23,11) =  113-114 =1$	$P(27,2) =  228-220 =8$	
$P(23,12) =  118-113 =5$	$P(27,3) =  221-224 =3$	
$P(23,13) =  108-105 =3$	$P(27,4) =  200-130 =70$	
$P(23,14) =  107-116 =9$	$P(27,5) =  99-70 =29$	
$P(23,15) =  111-107 =4$	$P(27,6) =  108-84 =24$	
$P(24,0) =  246-243 =3$	$P(27,7) =  119-157 =38$	

Sehingga diperoleh hasil operasi frame subtraction adalah sebagai berikut :

**Tabel 3.6** Hasil Frame Substraction

5	5	5	5	4	4	3	2	3	3	3	3	45	34	10	51	14	2	2	0	0	1	2	2	3	2	2	2	2	1
8	6	7	6	4	3	6	4	5	3	7	17	36	2	17	26	10	0	1	0	0	1	0	1	2	3	2	4	3	2
3	4	3	4	1	4	4	4	2	5	3	5	3	11	8	2	3	7	9	11	10	11	10	10	8	4	7	8	7	6
14	14	8	13	10	10	13	11	4	6	1	4	6	8	21	7	8	8	3	3	4	1	0	2	4	9	2	3	9	7
96	90	89	89	102	92	93	67	74	87	90	68	72	44	48	61	56	53	63	66	79	74	58	41	36	83	74	70	57	49
5	5	11	12	7	10	10	29	19	70	45	44	28	14	9	8	9	24	53	18	8	16	39	15	38	12	15	29	8	18
4	0	3	2	7	7	24	37	4	2	18	28	20	8	17	6	6	36	48	95	25	11	28	16	48	27	38	24	13	33
10	13	28	56	71	64	38	1	34	27	5	1	3	17	3	10	4	10	51	13	78	5	6	20	5	31	46	38	1	22
73	73	72	53	14	34	44	19	0	2	2	3	6	2	5	12	8	6	1	44	43	35	33	39	45	60	10	6	19	40
16	7	41	67	44	8	5	2	3	0	3	2	5	3	12	5	2	5	1	7	2	64	47	54	93	3	11	12	2	14
45	69	32	6	8	1	3	3	0	6	1	2	2	4	10	9	4	2	10	7	6	23	37	75	6	34	8	7	3	2
21	13	6	4	1	2	2	4	2	0	2	6	15	1	11	7	5	4	25	43	10	20	8	1	60	66	8	21	4	2
9	5	1	9	1	0	4	2	0	3	0	4	13	1	19	7	2	7	1	11	0	7	9	5	18	15	66	42	15	20
3	5	6	5	1	2	4	1	1	0	5	5	18	10	14	6	5	7	1	9	1	1	6	3	11	10	10	45	65	20
2	1	0	0	2	1	1	1	1	3	2	5	9	17	10	8	3	2	3	5	7	3	0	9	0	11	2	19	12	33
3	1	0	0	3	0	0	1	1	4	5	1	2	9	8	4	2	0	2	4	3	2	4	4	2	0	8	8	16	4

#### 4. Melakukan ekstraksi background.

Menggunakan nilai  $T$  sebesar 80, maka diperoleh hasil ekstraksi background adalah sebagai berikut, dimana jika nilai piksel hasil frame subtraction lebih kecil dari 80 maka akan dianggap background (0) sedangkan lebih besar dari 80 maka akan dianggap foreground (1) :

**Tabel 3.7** Ekstraksi Background

$P(0,0) = (5 < 80) = 0$	$P(3,7) = (56 < 80) = 0$	$P(6,14) = (1 < 80) = 0$
$P(0,1) = (8 < 80) = 0$	$P(3,8) = (53 < 80) = 0$	$P(6,15) = (0 < 80) = 0$
$P(0,2) = (3 < 80) = 0$	$P(3,9) = (67 < 80) = 0$	$P(7,0) = (2 < 80) = 0$
$P(0,3) = (14 < 80) = 0$	$P(3,10) = (6 < 80) = 0$	$P(7,1) = (4 < 80) = 0$
$P(0,4) = (96 \geq 80) = 1$	$P(3,11) = (4 < 80) = 0$	$P(7,2) = (4 < 80) = 0$
$P(0,5) = (5 < 80) = 0$	$P(3,12) = (9 < 80) = 0$	$P(7,3) = (11 < 80) = 0$
$P(0,6) = (4 < 80) = 0$	$P(3,13) = (5 < 80) = 0$	$P(7,4) = (67 < 80) = 0$
$P(0,7) = (10 < 80) = 0$	$P(3,14) = (0 < 80) = 0$	$P(7,5) = (29 < 80) = 0$
$P(0,8) = (73 < 80) = 0$	$P(3,15) = (0 < 80) = 0$	$P(7,6) = (37 < 80) = 0$
$P(0,9) = (16 < 80) = 0$	$P(4,0) = (4 < 80) = 0$	$P(7,7) = (1 < 80) = 0$
$P(0,10) = (45 < 80) = 0$	$P(4,1) = (4 < 80) = 0$	$P(7,8) = (19 < 80) = 0$
$P(0,11) = (21 < 80) = 0$	$P(4,2) = (1 < 80) = 0$	$P(7,9) = (2 < 80) = 0$
$P(0,12) = (9 < 80) = 0$	$P(4,3) = (10 < 80) = 0$	$P(7,10) = (3 < 80) = 0$
$P(0,13) = (3 < 80) = 0$	$P(4,4) = (102 \geq 80) = 1$	$P(7,11) = (4 < 80) = 0$
$P(0,14) = (2 < 80) = 0$	$P(4,5) = (7 < 80) = 0$	$P(7,12) = (2 < 80) = 0$
$P(0,15) = (3 < 80) = 0$	$P(4,6) = (7 < 80) = 0$	$P(7,13) = (1 < 80) = 0$
$P(1,0) = (5 < 80) = 0$	$P(4,7) = (71 < 80) = 0$	$P(7,14) = (1 < 80) = 0$

**Lanjutan Tabel 3.7** Ekstraksi Background

$P(1,1) = (6 < 80) = 0$	$P(4,8) = (14 < 80) = 0$	$P(7,15) = (1 < 80) = 0$
$P(1,2) = (4 < 80) = 0$	$P(4,9) = (44 < 80) = 0$	$P(8,0) = (3 < 80) = 0$
$P(1,3) = (14 < 80) = 0$	$P(4,10) = (8 < 80) = 0$	$P(8,1) = (5 < 80) = 0$
$P(1,4) = (90 \geq 80) = 1$	$P(4,11) = (1 < 80) = 0$	$P(8,2) = (2 < 80) = 0$
$P(1,5) = (5 < 80) = 0$	$P(4,12) = (1 < 80) = 0$	$P(8,3) = (4 < 80) = 0$
$P(1,6) = (0 < 80) = 0$	$P(4,13) = (1 < 80) = 0$	$P(8,4) = (74 < 80) = 0$
$P(1,7) = (13 < 80) = 0$	$P(4,14) = (2 < 80) = 0$	$P(8,5) = (19 < 80) = 0$
$P(1,8) = (73 < 80) = 0$	$P(4,15) = (3 < 80) = 0$	$P(8,6) = (4 < 80) = 0$
$P(1,9) = (7 < 80) = 0$	$P(5,0) = (4 < 80) = 0$	$P(8,7) = (34 < 80) = 0$
$P(1,10) = (69 < 80) = 0$	$P(5,1) = (3 < 80) = 0$	$P(8,8) = (0 < 80) = 0$
$P(1,11) = (13 < 80) = 0$	$P(5,2) = (4 < 80) = 0$	$P(8,9) = (3 < 80) = 0$
$P(1,12) = (5 < 80) = 0$	$P(5,3) = (10 < 80) = 0$	$P(8,10) = (0 < 80) = 0$
$P(1,13) = (5 < 80) = 0$	$P(5,4) = (92 \geq 80) = 1$	$P(8,11) = (2 < 80) = 0$
$P(1,14) = (1 < 80) = 0$	$P(5,5) = (10 < 80) = 0$	$P(8,12) = (0 < 80) = 0$
$P(1,15) = (1 < 80) = 0$	$P(5,6) = (7 < 80) = 0$	$P(8,13) = (1 < 80) = 0$
$P(2,0) = (5 < 80) = 0$	$P(5,7) = (64 < 80) = 0$	$P(8,14) = (1 < 80) = 0$

Lanjutan Tabel 3.7 Ekstraksi Background

$P(2,1) = (7 < 80) = 0$	$P(5,8) = (34 < 80) = 0$	$P(8,15) = (1 < 80) = 0$
$P(2,2) = (3 < 80) = 0$	$P(5,9) = (8 < 80) = 0$	$P(9,0) = (3 < 80) = 0$
$P(2,3) = (8 < 80) = 0$	$P(5,10) = (1 < 80) = 0$	$P(9,1) = (3 < 80) = 0$
$P(2,4) = (89 \geq 80) = 1$	$P(5,11) = (2 < 80) = 0$	$P(9,2) = (5 < 80) = 0$
$P(2,5) = (11 < 80) = 0$	$P(5,12) = (0 < 80) = 0$	$P(9,3) = (6 < 80) = 0$
$P(2,6) = (3 < 80) = 0$	$P(5,13) = (2 < 80) = 0$	$P(9,4) = (87 \geq 80) = 1$
$P(2,7) = (28 < 80) = 0$	$P(5,14) = (1 < 80) = 0$	$P(9,5) = (70 < 80) = 0$
$P(2,8) = (72 < 80) = 0$	$P(5,15) = (0 < 80) = 0$	$P(9,6) = (2 < 80) = 0$
$P(2,9) = (41 < 80) = 0$	$P(6,0) = (3 < 80) = 0$	$P(9,7) = (27 < 80) = 0$
$P(2,10) = (32 < 80) = 0$	$P(6,1) = (6 < 80) = 0$	$P(9,8) = (2 < 80) = 0$
$P(2,11) = (6 < 80) = 0$	$P(6,2) = (4 < 80) = 0$	$P(9,9) = (0 < 80) = 0$
$P(2,12) = (1 < 80) = 0$	$P(6,3) = (13 < 80) = 0$	$P(9,10) = (6 < 80) = 0$
$P(2,13) = (6 < 80) = 0$	$P(6,4) = (93 \geq 80) = 1$	$P(9,11) = (0 < 80) = 0$
$P(2,14) = (0 < 80) = 0$	$P(6,5) = (10 < 80) = 0$	$P(9,12) = (3 < 80) = 0$
$P(2,15) = (0 < 80) = 0$	$P(6,6) = (24 < 80) = 0$	$P(9,13) = (0 < 80) = 0$
$P(3,0) = (5 < 80) = 0$	$P(6,7) = (38 < 80) = 0$	$P(9,14) = (3 < 80) = 0$
$P(3,1) = (6 < 80) = 0$	$P(6,8) = (44 < 80) = 0$	$P(9,15) = (4 < 80) = 0$
$P(3,2) = (4 < 80) = 0$	$P(6,9) = (5 < 80) = 0$	$P(10,0) = (3 < 80) = 0$
$P(3,3) = (13 < 80) = 0$	$P(6,10) = (3 < 80) = 0$	$P(10,1) = (7 < 80) = 0$
$P(3,4) = (89 \geq 80) = 1$	$P(6,11) = (2 < 80) = 0$	$P(10,2) = (3 < 80) = 0$
$P(3,5) = (12 < 80) = 0$	$P(6,12) = (4 < 80) = 0$	$P(10,3) = (1 < 80) = 0$
$P(3,6) = (2 < 80) = 0$	$P(6,13) = (4 < 80) = 0$	$P(10,4) = (90 \geq 80) = 1$
$P(10,5) = (45 < 80) = 0$	$P(13,12) = (1 < 80) = 0$	$P(17,3) = (8 < 80) = 0$
$P(10,6) = (18 < 80) = 0$	$P(13,13) = (10 < 80) = 0$	$P(17,4) = (53 < 80) = 0$
$P(10,7) = (5 < 80) = 0$	$P(13,14) = (17 < 80) = 0$	$P(17,5) = (24 < 80) = 0$
$P(10,8) = (2 < 80) = 0$	$P(13,15) = (9 < 80) = 0$	$P(17,6) = (36 < 80) = 0$
$P(10,9) = (3 < 80) = 0$	$P(14,0) = (10 < 80) = 0$	$P(17,7) = (10 < 80) = 0$
$P(10,10) = (1 < 80) = 0$	$P(14,1) = (17 < 80) = 0$	$P(17,8) = (6 < 80) = 0$
$P(10,11) = (2 < 80) = 0$	$P(14,2) = (8 < 80) = 0$	$P(17,9) = (5 < 80) = 0$
$P(10,12) = (0 < 80) = 0$	$P(14,3) = (21 < 80) = 0$	$P(17,10) = (2 < 80) = 0$
$P(10,13) = (5 < 80) = 0$	$P(14,4) = (48 < 80) = 0$	$P(17,11) = (4 < 80) = 0$
$P(10,14) = (2 < 80) = 0$	$P(14,5) = (9 < 80) = 0$	$P(17,12) = (7 < 80) = 0$
$P(10,15) = (5 < 80) = 0$	$P(14,6) = (17 < 80) = 0$	$P(17,13) = (7 < 80) = 0$
$P(11,0) = (3 < 80) = 0$	$P(14,7) = (3 < 80) = 0$	$P(17,14) = (2 < 80) = 0$
$P(11,1) = (17 < 80) = 0$	$P(14,8) = (5 < 80) = 0$	$P(17,15) = (0 < 80) = 0$
$P(11,2) = (5 < 80) = 0$	$P(14,9) = (12 < 80) = 0$	$P(18,0) = (2 < 80) = 0$
$P(11,3) = (4 < 80) = 0$	$P(14,10) = (10 < 80) = 0$	$P(18,1) = (1 < 80) = 0$
$P(11,4) = (68 < 80) = 0$	$P(14,11) = (11 < 80) = 0$	$P(18,2) = (9 < 80) = 0$
$P(11,5) = (44 < 80) = 0$	$P(14,12) = (19 < 80) = 0$	$P(18,3) = (3 < 80) = 0$
$P(11,6) = (28 < 80) = 0$	$P(14,13) = (14 < 80) = 0$	$P(18,4) = (63 < 80) = 0$
$P(11,7) = (1 < 80) = 0$	$P(14,14) = (10 < 80) = 0$	$P(18,5) = (53 < 80) = 0$
$P(11,8) = (3 < 80) = 0$	$P(14,15) = (8 < 80) = 0$	$P(18,6) = (48 < 80) = 0$

Lanjutan Tabel 3.7 Ekstraksi Background

$P(11,9) = (2 < 80) = 0$	$P(15,0) = (51 < 80) = 0$	$P(18,7) = (51 < 80) = 0$
$P(11,10) = (2 < 80) = 0$	$P(15,1) = (26 < 80) = 0$	$P(18,8) = (1 < 80) = 0$
$P(11,11) = (6 < 80) = 0$	$P(15,2) = (2 < 80) = 0$	$P(18,9) = (1 < 80) = 0$
$P(11,12) = (4 < 80) = 0$	$P(15,3) = (7 < 80) = 0$	$P(18,10) = (10 < 80) = 0$
$P(11,13) = (5 < 80) = 0$	$P(15,4) = (61 < 80) = 0$	$P(18,11) = (25 < 80) = 0$
$P(11,14) = (5 < 80) = 0$	$P(15,5) = (8 < 80) = 0$	$P(18,12) = (1 < 80) = 0$
$P(11,15) = (1 < 80) = 0$	$P(15,6) = (6 < 80) = 0$	$P(18,13) = (1 < 80) = 0$
$P(12,0) = (45 < 80) = 0$	$P(15,7) = (10 < 80) = 0$	$P(18,14) = (3 < 80) = 0$
$P(12,1) = (36 < 80) = 0$	$P(15,8) = (12 < 80) = 0$	$P(18,15) = (2 < 80) = 0$
$P(12,2) = (3 < 80) = 0$	$P(15,9) = (5 < 80) = 0$	$P(19,0) = (0 < 80) = 0$
$P(12,3) = (6 < 80) = 0$	$P(15,10) = (9 < 80) = 0$	$P(19,1) = (0 < 80) = 0$
$P(12,4) = (72 < 80) = 0$	$P(15,11) = (7 < 80) = 0$	$P(19,2) = (11 < 80) = 0$
$P(12,5) = (28 < 80) = 0$	$P(15,12) = (7 < 80) = 0$	$P(19,3) = (3 < 80) = 0$
$P(12,6) = (20 < 80) = 0$	$P(15,13) = (6 < 80) = 0$	$P(19,4) = (66 < 80) = 0$
$P(12,7) = (3 < 80) = 0$	$P(15,14) = (8 < 80) = 0$	$P(19,5) = (18 < 80) = 0$
$P(12,8) = (6 < 80) = 0$	$P(15,15) = (4 < 80) = 0$	$P(19,6) = (95 \geq 80) = 1$
$P(12,9) = (5 < 80) = 0$	$P(16,0) = (14 < 80) = 0$	$P(19,7) = (13 < 80) = 0$
$P(12,10) = (2 < 80) = 0$	$P(16,1) = (10 < 80) = 0$	$P(19,8) = (44 < 80) = 0$
$P(12,11) = (15 < 80) = 0$	$P(16,2) = (3 < 80) = 0$	$P(19,9) = (7 < 80) = 0$
$P(12,12) = (13 < 80) = 0$	$P(16,3) = (8 < 80) = 0$	$P(19,10) = (7 < 80) = 0$
$P(12,13) = (18 < 80) = 0$	$P(16,4) = (56 < 80) = 0$	$P(19,11) = (43 < 80) = 0$
$P(12,14) = (9 < 80) = 0$	$P(16,5) = (9 < 80) = 0$	$P(19,12) = (11 < 80) = 0$
$P(12,15) = (2 < 80) = 0$	$P(16,6) = (6 < 80) = 0$	$P(19,13) = (9 < 80) = 0$
$P(13,0) = (34 < 80) = 0$	$P(16,7) = (4 < 80) = 0$	$P(19,14) = (5 < 80) = 0$
$P(13,1) = (2 < 80) = 0$	$P(16,8) = (8 < 80) = 0$	$P(19,15) = (4 < 80) = 0$
$P(13,2) = (11 < 80) = 0$	$P(16,9) = (2 < 80) = 0$	$P(20,0) = (0 < 80) = 0$
$P(13,3) = (8 < 80) = 0$	$P(16,10) = (4 < 80) = 0$	$P(20,1) = (0 < 80) = 0$
$P(13,4) = (44 < 80) = 0$	$P(16,11) = (5 < 80) = 0$	$P(20,2) = (10 < 80) = 0$
$P(13,5) = (14 < 80) = 0$	$P(16,12) = (2 < 80) = 0$	$P(20,3) = (4 < 80) = 0$
$P(13,6) = (8 < 80) = 0$	$P(16,13) = (5 < 80) = 0$	$P(20,4) = (79 < 80) = 0$
$P(13,7) = (17 < 80) = 0$	$P(16,14) = (3 < 80) = 0$	$P(20,5) = (8 < 80) = 0$
$P(13,8) = (2 < 80) = 0$	$P(16,15) = (2 < 80) = 0$	$P(20,6) = (25 < 80) = 0$
$P(13,9) = (3 < 80) = 0$	$P(17,0) = (2 < 80) = 0$	$P(20,7) = (78 < 80) = 0$
$P(13,10) = (4 < 80) = 0$	$P(17,1) = (0 < 80) = 0$	$P(20,8) = (43 < 80) = 0$
$P(13,11) = (1 < 80) = 0$	$P(17,2) = (7 < 80) = 0$	$P(20,9) = (2 < 80) = 0$
$P(20,10) = (6 < 80) = 0$	$P(24,1) = (2 < 80) = 0$	$P(27,8) = (6 < 80) = 0$
$P(20,11) = (10 < 80) = 0$	$P(24,2) = (8 < 80) = 0$	$P(27,9) = (12 < 80) = 0$
$P(20,12) = (0 < 80) = 0$	$P(24,3) = (4 < 80) = 0$	$P(27,10) = (7 < 80) = 0$
$P(20,13) = (1 < 80) = 0$	$P(24,4) = (36 < 80) = 0$	$P(27,11) = (21 < 80) = 0$
$P(20,14) = (7 < 80) = 0$	$P(24,5) = (38 < 80) = 0$	$P(27,12) = (42 < 80) = 0$
$P(20,15) = (3 < 80) = 0$	$P(24,6) = (48 < 80) = 0$	$P(27,13) = (45 < 80) = 0$
$P(21,0) = (1 < 80) = 0$	$P(24,7) = (5 < 80) = 0$	$P(27,14) = (19 < 80) = 0$

Lanjutan Tabel 3.7 Ekstraksi Background

$P(21,1) = (1 < 80) = 0$	$P(24,8) = (45 < 80) = 0$	$P(27,15) = (8 < 80) = 0$
$P(21,2) = (11 < 80) = 0$	$P(24,9) = (93 \geq 80) = 1$	$P(28,0) = (2 < 80) = 0$
$P(21,3) = (1 < 80) = 0$	$P(24,10) = (6 < 80) = 0$	$P(28,1) = (3 < 80) = 0$
$P(21,4) = (74 < 80) = 0$	$P(24,11) = (60 < 80) = 0$	$P(28,2) = (7 < 80) = 0$
$P(21,5) = (16 < 80) = 0$	$P(24,12) = (18 < 80) = 0$	$P(28,3) = (9 < 80) = 0$
$P(21,6) = (11 < 80) = 0$	$P(24,13) = (11 < 80) = 0$	$P(28,4) = (57 < 80) = 0$
$P(21,7) = (5 < 80) = 0$	$P(24,14) = (0 < 80) = 0$	$P(28,5) = (8 < 80) = 0$
$P(21,8) = (35 < 80) = 0$	$P(24,15) = (2 < 80) = 0$	$P(28,6) = (13 < 80) = 0$
$P(21,9) = (64 < 80) = 0$	$P(25,0) = (2 < 80) = 0$	$P(28,7) = (1 < 80) = 0$
$P(21,10) = (23 < 80) = 0$	$P(25,1) = (3 < 80) = 0$	$P(28,8) = (19 < 80) = 0$
$P(21,11) = (20 < 80) = 0$	$P(25,2) = (4 < 80) = 0$	$P(28,9) = (2 < 80) = 0$
$P(21,12) = (7 < 80) = 0$	$P(25,3) = (9 < 80) = 0$	$P(28,10) = (3 < 80) = 0$
$P(21,13) = (1 < 80) = 0$	$P(25,4) = (83 \geq 80) = 1$	$P(28,11) = (4 < 80) = 0$
$P(21,14) = (3 < 80) = 0$	$P(25,5) = (12 < 80) = 0$	$P(28,12) = (15 < 80) = 0$
$P(21,15) = (2 < 80) = 0$	$P(25,6) = (27 < 80) = 0$	$P(28,13) = (65 < 80) = 0$
$P(22,0) = (2 < 80) = 0$	$P(25,7) = (31 < 80) = 0$	$P(28,14) = (12 < 80) = 0$
$P(22,1) = (0 < 80) = 0$	$P(25,8) = (60 < 80) = 0$	$P(28,15) = (16 < 80) = 0$
$P(22,2) = (10 < 80) = 0$	$P(25,9) = (3 < 80) = 0$	$P(29,0) = (1 < 80) = 0$
$P(22,3) = (0 < 80) = 0$	$P(25,10) = (34 < 80) = 0$	$P(29,1) = (2 < 80) = 0$
$P(22,4) = (58 < 80) = 0$	$P(25,11) = (66 < 80) = 0$	$P(29,2) = (6 < 80) = 0$
$P(22,5) = (39 < 80) = 0$	$P(25,12) = (15 < 80) = 0$	$P(29,3) = (7 < 80) = 0$
$P(22,6) = (28 < 80) = 0$	$P(25,13) = (10 < 80) = 0$	$P(29,4) = (49 < 80) = 0$
$P(22,7) = (6 < 80) = 0$	$P(25,14) = (11 < 80) = 0$	$P(29,5) = (18 < 80) = 0$
$P(22,8) = (33 < 80) = 0$	$P(25,15) = (0 < 80) = 0$	$P(29,6) = (33 < 80) = 0$
$P(22,9) = (47 < 80) = 0$	$P(26,0) = (2 < 80) = 0$	$P(29,7) = (22 < 80) = 0$
$P(22,10) = (37 < 80) = 0$	$P(26,1) = (2 < 80) = 0$	$P(29,8) = (40 < 80) = 0$
$P(22,11) = (8 < 80) = 0$	$P(26,2) = (7 < 80) = 0$	$P(29,9) = (14 < 80) = 0$
$P(22,12) = (9 < 80) = 0$	$P(26,3) = (2 < 80) = 0$	$P(29,10) = (2 < 80) = 0$
$P(22,13) = (6 < 80) = 0$	$P(26,4) = (74 < 80) = 0$	$P(29,11) = (2 < 80) = 0$
$P(22,14) = (0 < 80) = 0$	$P(26,5) = (15 < 80) = 0$	$P(29,12) = (20 < 80) = 0$
$P(22,15) = (4 < 80) = 0$	$P(26,6) = (38 < 80) = 0$	$P(29,13) = (20 < 80) = 0$
$P(23,0) = (2 < 80) = 0$	$P(26,7) = (46 < 80) = 0$	$P(29,14) = (33 < 80) = 0$
$P(23,1) = (1 < 80) = 0$	$P(26,8) = (10 < 80) = 0$	$P(29,15) = (4 < 80) = 0$
$P(23,2) = (10 < 80) = 0$	$P(26,9) = (11 < 80) = 0$	
$P(23,3) = (2 < 80) = 0$	$P(26,10) = (8 < 80) = 0$	
$P(23,4) = (41 < 80) = 0$	$P(26,11) = (8 < 80) = 0$	
$P(23,5) = (15 < 80) = 0$	$P(26,12) = (66 < 80) = 0$	
$P(23,6) = (16 < 80) = 0$	$P(26,13) = (10 < 80) = 0$	
$P(23,7) = (20 < 80) = 0$	$P(26,14) = (2 < 80) = 0$	
$P(23,8) = (39 < 80) = 0$	$P(26,15) = (8 < 80) = 0$	
$P(23,9) = (54 < 80) = 0$	$P(27,0) = (2 < 80) = 0$	
$P(23,10) = (75 < 80) = 0$	$P(27,1) = (4 < 80) = 0$	



frame foreground apakah itu dikarenakan cahaya atau ada pergerakan.

### 3.5 Perancangan Perangkat Lunak

Dalam perancangan perangkat lunak pada sistem ini secara keseluruhan sistem dibuat menggunakan Bahasa pemrograman Python dengan beberapa library yaitu openCV dan Anaconda.

#### 3.5.1 Penentuan Region Of Interest Pada Object Tracking Menggunakan Frame Substraction

Feed video CCTV sebagai input utama pada sistem ini menggunakan metode Frame Substraction. Berikut adalah beberapa tahapan tersebut :

##### 3.5.1.1 Blok 1

Perintah berikut digunakan untuk mengimport lybrary open cv yang digunakan untuk membaca file video memutar dan mengekstraksi frame frame yang ada pada video. ditambah lybrari fungsi tambahan untuk menandai area ROI pada video

```
import cv2 as cv2
from tracker import *
```

**Gambar 3.9** Kode Program untuk membaca file

##### 3.5.1.2 Blok 2

perintah berikut digunakan untuk menginisialisasi variabel library open cv dan variabel tracker untuk menandai area ROI

```
tracker = EuclideanDistTracker()
cap = cv2.VideoCapture("video1.mp4")
```

**Gambar 3.10** Kode Program untuk menandai area ROI

##### 3.5.1.3 Blok 3

```
object_detector = cv2.createBackgroundSubtractorMOG2()
object_detector.setDetectShadows(False)
object_detector.setVarThreshold(300)
```

**Gambar 3.11** Kode Program untuk mendeteksi objek

### 3.5.1.4 Blok 4

Perintah berikut merupakan sub rutin utama yang mana isinya berupa proses pembacaan frame-frame pada video melakukan frame difference yang dilanjutkan dengan menandai area-area ROI yang mana outputnya akan ditampilkan pada tiga feed video berbeda yaitu video hasil frame difference (hitam putih), video ROI yang telah ditandai (blok roi), dan video input asli

```
Total_ROI = 0

while True:
    ret, frame = cap.read()
    if frame is None:
        break

    roi = frame.copy()
    fgMask = object_detector.apply(frame)

    contours, _ = cv2.findContours(fgMask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    detections = []
    for cnt in contours:
        area = cv2.contourArea(cnt)

        if area > 100:
            #cv2.drawContours(roi, [cnt], -1, (0, 255, 0), 2)
            x, y, w, h = cv2.boundingRect(cnt)

            detections.append([x, y, w, h])
            #cv2.drawContours(frame, [cnt], -1, (0, 255, 0), 2)

    boxes_ids = tracker.update(detections)
    jumlah_ROI = 0
    for box_id in boxes_ids:
        x, y, w, h, id = box_id
        #cv2.putText(roi, str(id), (x, y - 15), cv2.FONT_HERSHEY_PLAIN, 2, (255, 0, 0), 2)
        Total_ROI = Total_ROI+1
        jumlah_ROI = jumlah_ROI+1

        cv2.putText(roi, "ROI-" + str(id), (x, y - 15), cv2.FONT_HERSHEY_PLAIN, 2, (255, 0, 0), 2)
        cv2.rectangle(roi, (x, y), (x + w, y + h), (0, 255, 0), 3)

    cv2.putText(roi, "ROI Terdeteksi : " + str(jumlah_ROI), (15, 15),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0))
    cv2.rectangle(frame, (10, 2), (100,20), (255,255,255), -1)
    cv2.putText(frame, "Frame Video : " + str(cap.get(cv2.CAP_PROP_POS_FRAMES)), (15, 15),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0))

    cv2.imshow("roi", roi)
    cv2.imshow("Frame", frame)
    cv2.imshow("Frame Difference", fgMask)

    keyboard = cv2.waitKey(30)
    if keyboard == 'q' or keyboard == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

**Gambar 3.12** Kode program pembacaan frame-frame video

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Analisis Data

Pada tahapan analisis, data yang dibutuhkan dalam penelitian ini adalah berupa video CCTV dari berbagai macam kebutuhan atau area, kemudian diolah menggunakan pengolahan citra yaitu menggunakan algoritma *Region Of Interest (ROI)* dan *Frame Substraction* untuk mendeteksi objek-objek yang bergerak pada sebuah video. Pada tahap analisis data dilakukan penentuan video yang akan digunakan. Kemudian mengimplementasikan system yang menggunakan algoritma *Region Of Interest (ROI)* dan *Frame Substraction* untuk mendeteksi objek-objek yang bergerak menggunakan python. Ada beberapa langkah dalam program ini, yaitu:

- a. Memanggil video yang akan digunakan.
- b. Pembagian frame-rame video.
- c. Melakukan dan *Frame Substraction* ROI pada frame .
- d. Kemudian menandai dan mengitung objek yang bergerak (area) pada frame dan menampilkannya dalam video

#### 4.2 Pengujian

Setelah merancang dan membuat sistem, selanjutnya dilakukan pengujian. Pengujian bertujuan untuk melihat sejauh mana sistem yang telah dibangun sesuai dengan yang diharapkan, contoh hasil penerapan *Region Of Interest (ROI)* dan *Frame Substraction* mendeteksi objek-objek bergerak dalam video dapat dilihat sebagai berikut ini :

##### 1. Blok 1

Perintah berikut digunakan untuk mengimport lybrary open cv yang digunakan untuk membaca file video memutar dan mengekstraksi frame frame yang ada pada video. ditambah lybrari fungsi tambahan untuk menandai area ROI pada video

```
import cv2 as cv2
from tracker import *
```

**Gambar 4.1** Kode Program untuk membaca file

## 2. Blok 2

perintah berikut digunakan untuk menginisialisasi variabel library open cv dan variabel tracker untuk menandai area ROI

```
tracker = EuclideanDistTracker()
cap = cv2.VideoCapture("video1.mp4")
```

**Gambar 4.2** Kode Program untuk menandai area ROI

## 3. Blok 3

perintah berikut digunakan untuk menginisialisasi objek open cv yang digunakan untuk melakukan operasi pembacaan frame dan proses frame difference dengan parameter threshold yang ditentukan

```
object_detector = cv2.createBackgroundSubtractorMOG2()
object_detector.setDetectShadows(False)
object_detector.setVarThreshold(300)
```

**Gambar 4.3** Kode Program untuk mendeteksi objek

Ketiga blok diatas difokuskan untuk melakukan proses pemecahan frame-frame pada video dan kemudian membaca piksel piksel pada frame-frame tersebut untuk melakukan proses pembacaan frame difference dan penentuan parameter threshold pada masing-masing frame.

## 4. Blok 4

Perintah berikut merupakan sub rutin utama yang mana isinya berupa proses pembacaan frame-frame pada video melakukan frame difference yang dilanjutkan dengan menandai area-area ROI yang mana outputnya akan ditampilkan pada tiga feed video berbeda yaitu video hasil frame difference (hitam putih), video ROI yang telah ditandai (blok roi), dan video input asli

```

Total_ROI = 0
while True:
    ret, frame = cap.read()
    if frame is None:
        break

    roi = frame.copy()
    fgmask = object_detector.apply(frame)

    contours, _ = cv2.findContours(fgmask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    detections = []
    for cnt in contours:
        area = cv2.contourArea(cnt)

        if area > 100:
            #cv2.drawContours(roi, [cnt], -1, (0, 255, 0), 2)
            x, y, w, h = cv2.boundingRect(cnt)

            detections.append([x, y, w, h])
            #cv2.drawContours(frame, [cnt], -1, (0, 255, 0), 2)

    boxes_ids = tracker.update(detections)
    jumlah_ROI = 0
    for box_id in boxes_ids:
        x, y, w, h, id = box_id
        #cv2.putText(roi, str(id), (x, y - 15), cv2.FONT_HERSHEY_PLAIN, 2, (255, 0, 0), 2)
        Total_ROI = Total_ROI+1
        jumlah_ROI = jumlah_ROI+1

        cv2.putText(roi, "ROI-" + str(id), (x, y - 15), cv2.FONT_HERSHEY_PLAIN, 2, (255, 0, 0), 2)
        cv2.rectangle(roi, (x, y), (x + w, y + h), (0, 255, 0), 3)

    cv2.putText(roi, "ROI Terdeteksi : " + str(jumlah_ROI), (15, 15),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0))
    cv2.rectangle(frame, (10, 2), (100,20), (255,255,255), -1)
    cv2.putText(frame, "Frame Video : " + str(cap.get(cv2.CAP_PROP_POS_FRAMES)), (15, 15),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0))

    cv2.imshow("roi", roi)
    cv2.imshow("Frame", frame)
    cv2.imshow("Frame Difference", fgmask)

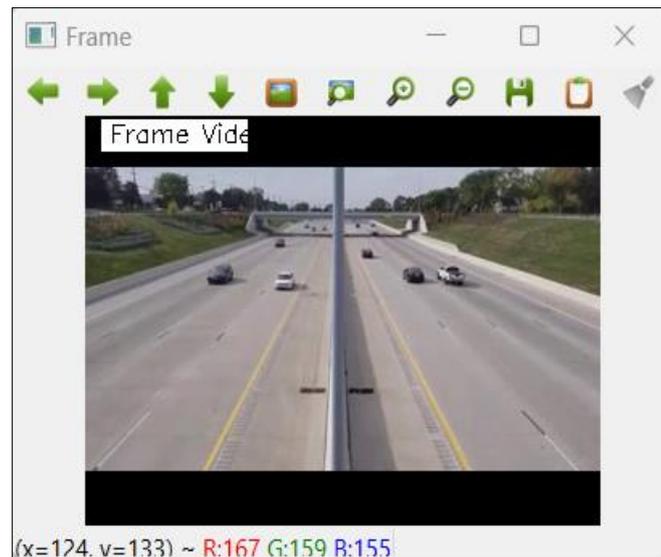
    keyboard = cv2.waitKey(30)
    if keyboard == 'q' or keyboard == 27:
        break

cap.release()
cv2.destroyAllWindows()

```

**Gambar 4.4** Kode program pembacaan frame-frame video

Dalam pengujiannya, penulis menggunakan dua buah video dengan nama videotes dengan format MP4. Berikut hasil pengujian yang dilakukan pada video tersebut:



**Gambar 4.5** Feed Videotes

Pada gambar 4.5 ditampilkan *feed* dari video yang digunakan yaitu videotes. Feed ini hanya menampilkan video asli tanpa adanya penerapan metode apapun.



**Gambar 4.6** Feed Frame Differences Videotes

Pada gambar 4.6 ditampilkan *feed Frame Differences* dari video yang digunakan yaitu videotes. Feed ini hanya menampilkan video hasil penerapan frame difference untuk memberikan hambaran yang lebih jelas dari perbedaan area yang merupakan latar dan objek.



**Gambar 4.7** Feed ROI Videotes

Pada gambar 4.7 ditampilkan feed Roi dari video yang digunakan yaitu videotes. Feed ini hanya menampilkan video hasil penerapan frame difference untuk memberikan hambaran yang lebih jelas dari perbedaan area yang merupakan latar dan objek.

Pada bagian akhir dari koding akan ditampilkan jumlah ROI yang didapatkan dari videotes, yaitu berjumlah 2431 ROI.

### 4.3 Analisa Hasil

Menggunakan model KNN yang telah dibangun kemudian telah dilakukan pengujian identifikasi terhadap 10 citra uji menggunakan model KNN yang dapat dilihat pada gambar dan tabel berikut :

```

{1: (60, 107), 3: (103, 81), 4: (293, 102)}
{1: (60, 110), 4: (293, 102), 3: (103, 81)}
In [13]: print("Total ROI yang terdeteksi : ",Total_ROI)
Total ROI yang terdeteksi : 1922
In [ ]:

```

**Gambar 4.8** ROI Video

```

{3: (62, 180), 5: (265, 139)}
{3: (62, 180), 5: (262, 141)}
{3: (57, 185), 5: (262, 141)}
{3: (57, 185), 5: (262, 142)}
{3: (55, 191), 5: (262, 142)}
{3: (55, 191), 5: (262, 142)}

In [11]: print("Total ROI yang terdeteksi : ",Total_ROI)

Total ROI yang terdeteksi : 31102

In [ ]:

```

Gambar 4.9 ROI Video2

```

{9: (243, 177), 8: (69, 147), 5: (225, 129)}
{9: (243, 177), 8: (69, 147), 5: (222, 128)}
{9: (239, 171), 8: (69, 147), 5: (222, 128)}
{9: (239, 171), 8: (69, 147), 5: (222, 128)}

In [21]: print("Total ROI yang terdeteksi : ",Total_ROI)

Total ROI yang terdeteksi : 2474

In [ ]:

```

Gambar 4.10 ROI Video3

```

{5: (245, 195), 6: (158, 180)}
{5: (245, 198), 6: (158, 180)}
{5: (245, 198), 6: (156, 182)}
{5: (245, 199), 6: (156, 182)}
{5: (245, 199), 6: (156, 184)}
{5: (246, 201), 6: (156, 184)}
{5: (246, 201), 6: (155, 185)}
{5: (246, 201), 6: (155, 186), 7: (97, 72)}
{5: (246, 201), 6: (155, 186), 7: (97, 72)}

In [31]: print("Total ROI yang terdeteksi : ",Total_ROI)

Total ROI yang terdeteksi : 3433

In [ ]:

In [ ]:

```

Gambar 4.11 ROI Video4

```

{9: (299, 175)}
{9: (303, 179)}
{9: (305, 184)}
{9: (308, 186)}
{9: (311, 188)}
{9: (313, 190)}

In [36]: print("Total ROI yang terdeteksi : ",Total_ROI)

Total ROI yang terdeteksi : 2004

```

Gambar 4.12 ROI Video5

```

{10: (230, 197), 11: (278, 187), 7: (74, 168)}
{10: (234, 201), 11: (278, 187), 7: (74, 168)}
{10: (234, 201), 11: (287, 197), 7: (74, 168)}
{10: (234, 201), 11: (287, 197), 7: (74, 168)}
{10: (234, 201), 11: (287, 197), 7: (74, 168)}

In [41]: print("Total ROI yang terdeteksi : ",Total_ROI)

Total ROI yang terdeteksi : 2168

In [ ]:

```

Gambar 4.13 ROI Video6

```

{8: (41, 195), 4: (20, 169), 9: (298, 170)}
{8: (38, 195), 4: (20, 169), 9: (298, 170)}
{8: (38, 195), 4: (20, 169), 9: (300, 175)}
{8: (38, 195), 4: (20, 169), 9: (300, 175)}
{8: (38, 195), 4: (20, 169), 9: (300, 175)}

In [46]: print("Total ROI yang terdeteksi : ",Total_ROI)

Total ROI yang terdeteksi : 1897

In [ ]:

```

Gambar 4.14 ROI Video7

```

{15: (92, 200), 11: (28, 201), 16: (247, 172), 13: (27,
{15: (92, 200), 11: (28, 201), 16: (263, 189), 13: (27,
{15: (92, 200), 11: (40, 194), 16: (263, 189), 13: (27,
{15: (92, 200), 11: (30, 198), 16: (263, 189), 13: (27,
{15: (92, 200), 11: (30, 198), 16: (263, 189), 13: (27,
In [51]: print("Total ROI yang terdeteksi : ",Total_ROI)

Total ROI yang terdeteksi : 2329

In [ ]:

```

**Gambar 4.15 ROI Video8**

No	Nama Video	Durasi	Total ROI yang terdeteksi
1	video1	1 menit	1922
2	video2	1 menit	31102
3	video3	1 menit	2474
4	video4	1 menit	3433
5	video5	1 menit	2004
6	video6	1 menit	2168
7	video7	1 menit	1897
8	video8	1 menit	2329

**Tabel 4.1 Hasil Pengujian ROI**

Dari table diatas, dapat diambil kesimpulan bahwa system yang dibuat oleh penulis berhasil menerapkan proses frame differences dengan baik dan sekaligus mampu memberikan penandaan ROI dengan cukup deta

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Berdasarkan penelitian yang dilakukan penulis dalam penerapan algoritma *Region Of Interest (ROI)* dan *Frame Substraction* untuk Menghitung Jumlah Kendaraan Pada Jalan Raya

1. Menghitung jumlah kendaraan di jalan raya dengan penentuan Region Of Interest (ROI) menggunakan metode Frame Substraction, dimana proses ROI dilakukan dengan penandaan kotak merah pada kendaraan dalam video berjalan.
2. Dalam hal ini frame subtraction memberikan deretan gambar yang digunakan dalam video kemudian algoritma ROI akan mengolah frame frame tersebut dan membandingkannya satu dengan yang lain untuk menandakan area mana yang merupakan area yang mengalami perubahan nilai dan ditandai sebagai Region Of Interest
3. Kelemahan dari Program aplikasi ROI ini adalah pendeteksian yang dilakukan dalam ROI tidak hanya terfokus ke kendaraan saja, pendeteksian juga terdapat pada bagian objek yang bergerak karena angin yang membuat objek tersebut terdeteksi oleh ROI
4. Kelebihan dari Program Aplikasi ROI ini adalah pendeteksian yang dilakukan untuk mendeteksi kendaraan di jalan raya dapat terlihat jelas dan jumlah kendaraan yang melintas di jalan raya dapat terlihat jumlahnya pada ROI

#### **5.2. Saran**

Adapun saran penulis dalam penelitian ini sebagai berikut.

1. Perlunya pengambilan video dengan memanfaatkan Gimbal, atau bahkan menggunakan video stabilizer untuk memberikan hasil video yang stabil.
2. Perlu algoritma tambahan untuk memberikan identitas pada objek untuk lebih menonjolkan efek dan kegunaan dari ROI.
3. Penulis menyadari dalam pengolahan data, ada banyak kesalahan yang terjadi seperti

pemilihan video yang masih memanfaatkan video dari Youtube, pengujian yang sedikit, dan kekurangan lainnya. Sehingga penulis mengharapkan adanya perbaikan 1 pemanfaatan atau algoritma pendung lainnya yang dapat digunakan pada penelitian lanjutan.

## DAFTAR PUSTAKA

- Hendrianto Pratomo, A, Kaswidjanti, W, dan Mu'arifah, S. *Jurnal Teknologi Informasi dan ilmu komputer*. "Implementasi Algoritma Region Of Interest (ROI) Untuk meningkatkan performa algoritma deteksi dan klasifikasi kendaraan". 2020.
- Arby Yuha, R, Danu Al Fiqri, M, Ashari, Pratama, R, dan Harahap, M. *Seminar Nasional APTIKOM*. "Deteksi gerakan pada kamera CCTV dengan Algoritma Frame Difference dan Frame Substraction". 2019.
- <http://desaindotgrafis.blogspot.com/>. 2014
- Mercu Buana. <http://digilib.mercubuana.ac.id>. 2019
- Md Jan, M, Zainal, N, dan Jamaludin, S. *IAES Internasional Journal of Artificial Intelligence*. "Region of interest-based image retrieval techniques : a review". 2020.
- Laksmi Dewi, AI. *Tugas Akhir Jurusan Teknik Elektro Fakultas Teknik Universitas Udayana Jimbaran Bali*. "Frame Rate Minimum pada Video tanpa kompresi menggunakan Normalized Frame Difference Sebagai Pendeskripsi Intensitas Gerak". 2015.
- Romzi, M, dan Kurniawan, B. *Jurnal Teknik Informatika Mahakarya*. "Pembelajaran Pemrograman Python dengan pendekatan Algoritma". 2020.
- Tri Yunardi, R, W.Mardhiya, A, Hafi Yahya, M, dan Satria Arisgraha, FC. *ELKHA*. "Desain dan Implementasi Visual Object Tracking Menggunakan Pan and Tilt Vision System". 2019.