

## Penerapan dan Analisis Perhitungan Orang dengan Chromatic Color Model Studi Kasus : Perhitungan orang dalam sebuah antrian

### Implementation and Analysis of People Counting with Chromatic Color Model Case Study: Calculation of People In a Queue

Imamul Akhyar<sup>1</sup>, Bedy Purnama<sup>2</sup>, Febryanti Sthevanie<sup>3</sup>

<sup>1,2,3</sup>Departemen Informatika, Telkom Engineering School, Telkom University  
Jalan Telekomunikasi No.1, Dayeuh Kolot, Bandung 40257 [imamul.akhyar4@gmail.com](mailto:imamul.akhyar4@gmail.com)<sup>1</sup>,  
[bedy.purnama@gmail.com](mailto:bedy.purnama@gmail.com)<sup>2</sup> [febryantisthevanie@gmail.com](mailto:febryantisthevanie@gmail.com)<sup>3</sup>

---

#### Abstrak

Saat ini, sistem *people counting* sudah diaplikasikan di berbagai tempat dan menjadi sebuah kebutuhan yang penting bagi beberapa kalangan. Seperti pengaplikasian sistem *people counting* di tempat pusat perbelanjaan baik besar maupun kecil, di museum, tempat pendidikan, kantor, dan masih banyak lagi. Terlebih lagi dengan semakin banyaknya minat masyarakat untuk menggeluti lebih dalam tentang *computer vision*, membuat *people counting* semakin berkembang seiring waktu, begitu pula dengan kebutuhan dari masyarakat sendiri.

Pada kasus antrian, sistem *people counting* juga bisa diterapkan. Tujuannya adalah agar pengawasan dalam sebuah antrian bisa dilakukan secara otomatis tanpa adanya bantuan operator. Dengan *people counting*, terhitung jumlah orang yang ada didalam antrian pada area jangkauan kamera.

Dalam Tugas Akhir ini penulis akan menggunakan studi kasus untuk perhitungan orang pada antrian yang *linier*. Pertama-tama dengan menggunakan *Chromatic Color Model* setelah menggunakan metode *Gaussian Mixture Model* pada tahap *background subtraction*, kendala dari *noise* bayangan pada objek yang akan dihitung akan terselesaikan. Dilakukan segmentasi objek dan *multiple person classification* membuat perhitungan untuk dua arah/*bi-directional* semakin akurat.

Dari hasil pengujian dan analisis terhadap sistem, kombinasi metode *Gaussian Mixture Model*, *Chromatic Color Model* dan *Multiple Person Classification* mampu menghasilkan akurasi sebesar 88%.

**Kata kunci** : *people counting*, antrian, *chromatic color model*, *multiple person*

---

#### Abstract

Nowadays, people counting system has been applied in various places and become an important requirement for some circles. Just like the application of people counting system in a shopping center both large and small, in museums, education, office, and many more. Moreover, with the increasing number of public interest to study more deeply about computer vision, makes people counting is growing over time, as well as the needs of the community itself.

In the case of the queue, people counting system can also be applied. The goal is to control in a queue can be performed automatically without operator assistance. With people counting, count the number of people who are in the queue in the area of camera range.

In this final project, writer will use a case study for the calculation of the linear queue. First by using Chromatic Color Model after using Gaussian Mixture Models on stage Substraction background, shadow noise constraints of the object to be counted will be resolved. Do multiple object segmentation and classification person making the calculations for two-way / bi-directional increasingly accurate.

From the test results and analysis of the system, a combination of methods Gaussian Mixture Models, Chromatic Color Classification Model and Multiple Person is able to produce an accuracy of 88%.

**Keywords** : people counting, queue, chromatic color model, multiple person classification

---

## 1. Pendahuluan

*People Counting* atau perhitungan orang terdiri dari semua teknik yang mampu mengekstrak jumlah orang yang ada pada daerah tertentu yang akan diamati pada dunia nyata yang memenuhi batasan akurasi dan kehandalan [17]. *People Counting* biasanya digunakan di tempat untuk mendapatkan sebuah informasi tertentu. Contohnya adalah penerapan *people counting* pada sebuah antrian baik antrian di kasir, pintu masuk sebuah tempat bermain, dsb. Orang dalam antrian cenderung berbaris/ *linier* sesuai jalur yang sudah disediakan dan memiliki jarak antara orang yang satu dengan yang lainnya. Seiring dengan berkembangnya teknologi, bidang *computer vision* seperti *people counting* pun semakin diminati.

Untuk penelitian *people counting*, diterapkan *people counting* dengan menggunakan *two stage segmentation* menggunakan kamera zenithal untuk mendapatkan hasil akurasi yang tinggi [4]. Penelitian berikutnya, digunakan metode *Curve Analysis* untuk menghitung jumlah orang yang ada secara *realtime* [1]. Dengan menggunakan *People Flow Counting System*, langkah pertama yang dilakukan adalah mendeteksi kepala terlebih dahulu agar pengenalan antara objek orang dan objek lain lebih jelas [5]. Berikutnya, dikembangkan teknik *people counting* berdasarkan aliran *bi-directional*, dengan menggunakan *flow* sebagai tahap *tracking*, perhitungan orang dinilai lebih baik [18]. *Kalman Filter* digunakan sebagai parameter objek dengan catatan pergerakannya stabil [12]. Dan dengan sudut kamera 90 derajat atau overhead, dikembangkan metode *Multiple Line* [2].

Di dalam Tugas Akhir ini, penulis mengambil studi kasus *people counting* pada antrian yang berada di luar ruangan (*outdoor*). Antrian yang dimaksud adalah antrian 1 baris dimana hanya ada 1 jalan masuk dan 1 jalan keluar (1 arah). Metode yang digunakan adalah pengkombinasian pada tahap *background subtraction* yaitu metode *Gaussian Mixture Model* dengan *Chromatic Color Model*. Diantara teknik pemisahan *background* yang lain, GMM dinilai sebagai metode yang paling handal, tetapi memiliki kelemahan apabila terdapat bayangan pada *foreground*. Dengan *Chromatic*

*Color Model* [6], *brightness* akan dipisahkan dengan komponen *chromaticity* dan akan dibandingkan nilai pikselnya antara *non-background* dengan *background component* nya. Dan untuk proses menghitung orang sebelumnya dilakukan *Multiple Person Classification* sehingga oklusi bisa diatasi dan perhitungan lebih akurat. Diharapkan dengan dibuatnya sistem ini, bisa dijadikan *prototype* untuk menghitung antrian seperti antrian di taman bermain yang bisa memberikan informasi berapa jumlah orang dalam antrian sebuah wahana bermain.

## 2. Landasan Teori

### 2.1 *People Counting*

*People Counting* atau perhitungan orang terdiri dari semua teknik yang mampu mengekstrak jumlah orang yang ada pada daerah tertentu yang akan diamati pada dunia nyata yang memenuhi batasan akurasi dan kehandalan [19]. *People Counting* biasanya digunakan di tempat yang tergolong ramai untuk mendapatkan sebuah informasi tertentu. Contohnya adalah penerapan *people counting* pada sebuah pusat perbelanjaan yang bertujuan untuk menganalisis perilaku pengunjung dan menentukan langkah-langkah yang harus dilakukan oleh pemilik pusat perbelanjaan tersebut untuk memaksimalkan usahanya.[7].

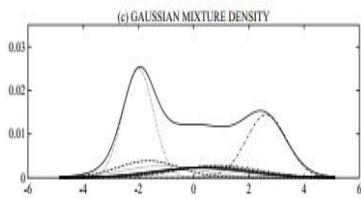
### 2.2 *Background Subtraction*

Sebelum menghitung orang, setiap objek yang bergerak harus terlebih dahulu diekstrak dari *background* [7]. *Background subtraction* merupakan proses manipulasi sebuah gambar *background* untuk mendapatkan objek yang diinginkan, umumnya objek yang bergerak atau dalam *people counting* merupakan orang yang akan dihitung. Terdapat beberapa metode untuk melakukan hal ini, seperti *frame difference* yang merupakan metode yang paling sederhana. Metode ini mengekstrak objek yang bergerak sebagai elemen *foreground* yang diambil dari perbedaan gambar antara setiap *frame* dan dikenal sebagai *background model* dari sebuah adegan.[10] Metode pada *background subtraction* sendiri cukup beragam dimulai dari *Gaussian Mixture Model*, *Frame Difference*, *Running Gaussian average*, dst.[9]

**2.2.1 Gaussian Mixture Model**

Gaussian Mixture Model merupakan salah satu metode dalam *background subtraction*. Metode ini digunakan untuk mendeskripsikan piksel dari *background*. Model ini dapat menerima *multimodal background*, sehingga merupakan model yang *robust* terhadap gerakan berulang dalam elemen latar, objek yang bergerak lambat, dan memperkenalkan atau menghapus objek dari latar [9].

GMM akan memberikan fungsi-fungsi komponen Gaussian untuk tiap piksel, dengan *input* adalah warna piksel dimana model-model GMM terbentuk berdasarkan waktu. Model akan membentuk 2 komponen utama, yakni model *background* dengan model *foreground*. Model *background* adalah model yang mencerminkan latar dari area yang diamati, sedangkan model *foreground* merupakan model yang mencerminkan objek yang bisa diamati.

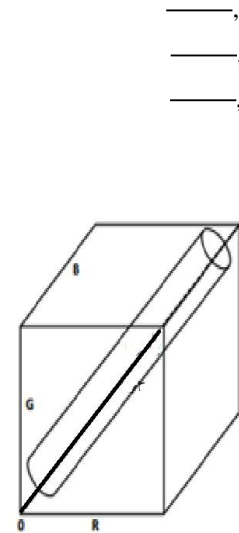


Gambar 2-1 Gaussian Mixture Model [15]

**2.3 Chromatic Color Model**

Chromatic color merupakan warna yang kurang memiliki hue/corak seperti putih, abu-abu, atau hitam. Dengan menggabungkan dengan metode GMM, bayangan yang ada pada objek bisa diatasi. *Brightness* dipisahkan dengan *chromaticity* dan membuatnya kompatibel dengan model campuran. Setelah menentukan *threshold* dan perbandingan *non-background* dengan *background*, maka piksel bisa dianggap sebagai bayangan. [14].

Ide untuk memisahkan bayangan dengan *foreground* adalah bayangan memiliki kemiripan dalam *chromaticity* tetapi memiliki *brightness* yang lebih rendah dalam RGB. Jadi nilai *brightness* dinaikkan. Rumus yang digunakan untuk mendapatkan koordinat dari *chromatic color model* adalah :

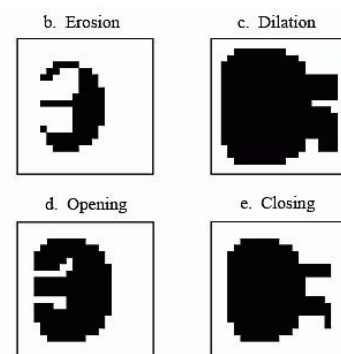


Gambar 2-2 Chromatic Color Model dengan Gaussian Mixture Model

**2.3 Morfologi Opening**

Operasi morfologi pada sebuah citra dilakukan agar citra yang diinginkan berkurang jumlah *noise* nya. Salah satu operasi morfologi adalah morfologi opening. Morfologi opening didefinisikan sebagai morfologi erosi yang diikuti oleh morfologi dilasi. Gambar 2.3 menunjukkan operasi kebalikan dari morfologi closing, yang didefinisikan sebagai dilatasi diikuti oleh erosi.

Seperti yang digambarkan oleh contoh dibawah ini, membuka menghilangkan pulau-pulau kecil dan filamen tipis objek piksel.



Gambar 2-3 Jenis Morfologi [17]

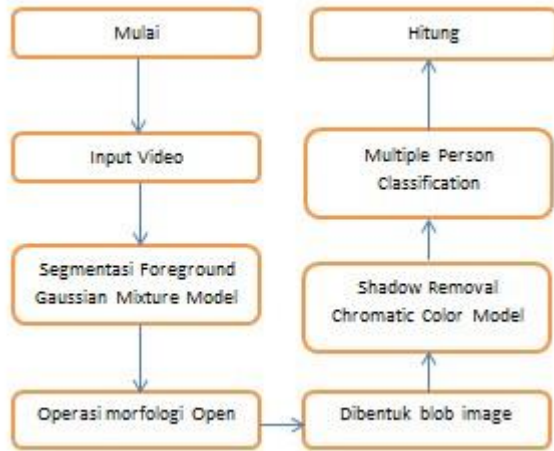
Demikian juga, penutupan menghilangkan pulau-pulau dan filamen tipis piksel latar belakang. Teknik ini

berguna untuk menangani gambar bising di mana beberapa piksel memiliki nilai biner yang salah. Misalnya, mungkin akan diketahui bahwa objek tidak dapat berisi "lubang", atau perbatasan objek harus mulus. [17]

### 3. Perancangan Sistem

#### 3.1 Gambaran Umum Sistem

Sistem yang dibangun dibuat menggunakan beberapa tahap. Yang pertama tahap *background subtraction*. Menggunakan metode *Gaussian Mixture Model* pada awalnya untuk memisahkan antara *foreground* dengan *background* pada setiap *frame*. Setelah didapatkan citra *foreground* dan *background*, maka dilanjutkan ke tahap berikutnya menggunakan *Chromatic Color Model* sehingga didapatkan *foreground* yang terbebas dari *noise* bayangan/ dijalankan *shadow removing algorithm*. Setelah didapatkan *foreground* yang bersih, akan ditemukan oklusi dari *foreground* tersebut apabila antara *blob foreground* terlalu berdekatan sehingga perlu dilakukan proses pemisahan dengan menggunakan *Multiple Person Classification*.



Gambar 0-1 Blok Proses Sistem

#### 3.2 Pengambilan Gambar

Dalam tahap pengambilan video, 1 set kamera digital diletakkan dengan posisi membentuk 45 derajat dan ketinggian sekitar 2,5 meter. Dilakukan percobaan percobaan untuk menentukan posisi dan ketinggian yang optimal. Ukuran/ resolusi dari video adalah 800 x 592 piksel.

Salah satu contoh video/ data yang digunakan adalah seperti dibawah ini :



Gambar 0-2 Data untuk learning background



Gambar 0-3 Data untuk learning ekstraksi ciri



Gambar 0-4 Data untuk fase testing Data set terdiri dari beberapa skenario, ada yang hanya 1 orang dalam antrian, skenario membedakan objek manusia dan bukan

manusia (kucing), skenario antrian dengan keadaan tertib (diam) dan skenario antrian dengan kebiasaan normal (berdekatan).

### 3.3 Segmentasi Foreground

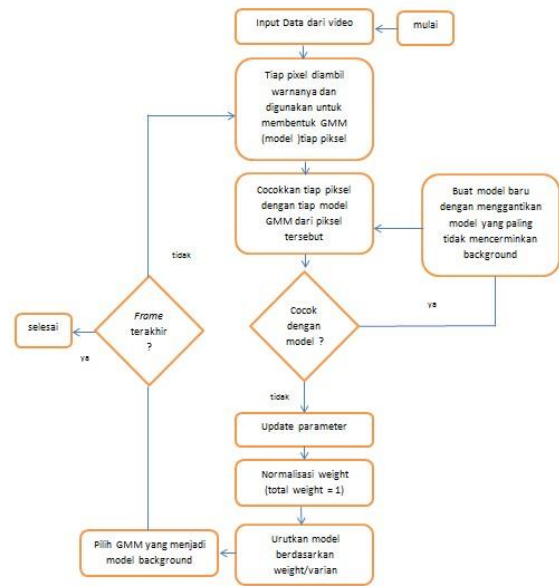
Pada *people counting*, sangat penting untuk mendeteksi objek bergerak yang akan dihitung. Apabila tidak akuratnya saat pengenalan dan pemisahan *background* dan *foreground* maka akurasi sistem pun akan berkurang. Tahap *background segmentation / moving object detection*, digunakan metode *Gaussian Mixture Model (GMM)* untuk mendapatkan *foreground*. Mengapa pada sistem ini menggunakan GMM adalah karena metode *Frame Difference* yang proses komputasinya lebih sederhana namun tidak cukup handal. GMM pun bisa mengatasi pergerakan yang lambat/ *slow moving* sehingga *foreground* tidak diklasifikasikan sebagai *background*.

Dengan GMM, sudah bisa menghasilkan *foreground* seperti gambar dibawah ini :



Gambar 0-5 Hasil Segmentasi Foreground

Beberapa *frame* diambil pikselnya untuk mendapat model GMM, lalu model GMM tersebut dicocokkan dengan model GMM dari tiap piksel pada *frame* berikutnya. Kemungkinan pertama jika model tidak cocok, maka dibuat model GMM baru untuk menggantikan model GMM yang tidak mencerminkan *background*, kemudian dianggap sebagai *foreground*. Kemungkinan kedua jika terdapat piksel yang cocok modelnya, maka dilakukan *update* terhadap parameter dalam model GMM pada piksel tersebut dan dikelompokkan sebagai *background*. *Frame* yang dikelompokkan sebagai *background* berwarna hitam, dan model *foreground* yang berwarna putih.



Gambar 0-6 Alur kerja GMM [16]

### 3.4 Operasi morfologi open

Dengan menggunakan *background* subtraction, sering kali masih menyisakan noise yang ditimbulkan dari gerakan halus seperti gerakan daun, rumput, dsb. Noise mengakibatkan kurang sempurnanya pemisahan *foreground*. Untuk mengatasi masalah tersebut, dilakukan morfologi open. Morfologi open melakukan erosi pada yang dikenali *foreground* dan dilakukan dilasi setelahnya agar *foreground* yang benar dikenali orang dikembalikan lagi.

### 3.5 Ekstraksi fitur

Setelah dipisahkan antara *background* dan *foreground*, tentu saja diperlukan identifikasi dan klasifikasi apakah objek yang dikenali sebagai *foreground* tersebut objek manusia atau tidak. Dengan menggunakan tinggi dan lebar dari objek yang ditemukan dan membandingkan piksel dari objek tersebut maka akan ditentukan klasifikasi objek. Selanjutnya akan ditentukan apakah objek tersebut berhimpitan/ dua orang atau hanya satu orang.

$$() \quad () \quad () \quad () \quad (3.1)$$

$$(3.2)$$

$$(3.3)$$

$$(3.4)$$

$$(3.5)$$

Dimana adalah Evaluation parameter.

A adalah piksel blob, H adalah Height, dan W adalah Width.

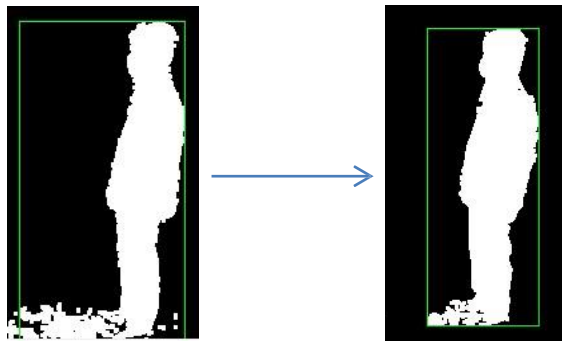
adalah jumlah orang, adalah *minimum quantity for single person* dan adalah *threshold* yang didapat dari *Evaluation parameter* untuk ukuran 1 orang.

Setelah dibuat rentang, diperlukan tambahan kondisi untuk menangani kasus dimana saat orang sedang berjalan, maka lebar dari blob pun akan semakin membesar dan dapat menimbulkan perhitungan orang yang salah. Disaat 1 orang sedang melangkah seharusnya dihitung 1, tetapi akan terhitung 2 karena memasuki rentang lebar 2 orang. Oleh karena itu, ditambahkan proses seleksi tiap *bounding box*, apabila memenuhi minimal *blob* untuk 1 orang dalam 1 *bounding box*, maka akan dihitung namun apabila dalam *bounding box* tersebut hanya menyisakan tangan atau kaki yang terpotong saat melangkah, maka mengakibatkan *blob* kurang dari minimal *blob* yang ditentukan dan tidak akan terhitung.

### 3.6 Shadow Removal

Dengan menggunakan GMM, akan terjadi kendala yaitu tidak bisa membedakan antara *piksel foreground* dengan bayangan sehingga *blob* akan membesar dan akurat. Untuk mengatasinya digunakan metode *Chromatic Color Model*. Ide utama dari metode ini adalah memisahkan antara *brightness/* piksel yang berwarna terang dengan *chromacity/* piksel yang berwarna gelap agar cocok dengan *mixture model*.

Piksel akan ditentukan apakah bayangan atau objek dengan ditentukan terlebih dahulu *threshold*.



Gambar 0-7 Hasil Shadow Removal

Pada tahap *shadow removal*, dilakukan perubahan nilai warna dari piksel yang telah dipisahkan menjadi

*foreground/ blob*. Dicari nilai dari koordinat *chromaticity* pada setiap piksel dengan rumus :

$$\begin{aligned} r &= \frac{R}{R+G+B}, \\ g &= \frac{G}{R+G+B}, \\ b &= \frac{B}{R+G+B}, \end{aligned}$$

Dimana  $r + g + b = 1$  [11]. Dengan begitu, piksel yang mengalami perubahan nilai karena adanya bayangan bisa diatasi.

### 3.7 Hitung

Pada proses terakhir yaitu penghitungan orang dilakukan dengan cara menghitung jumlah *bounding box* yang ada pada *frame*.

## 4. Pengujian Sistem dan Analisis

### 4.1 Pengujian Sistem

Pada tahap sebelumnya, perancangan sistem telah dilakukan untuk menunjang kebutuhan penelitian. Setelah sistem dirancang, maka sistem akan diuji untuk mengetahui peromansi sistem dari beberapa parameter uji yang disesuaikan dengan tujuan penelitian.

#### 4.1.1 Tujuan Pengujian

Pengujian pada Tugas Akhir ini dilakukan dengan tujuan yang merupakan pengembangan dari tujuan penelitian yang telah ditulis di bab pendahuluan, antara lain :

1. Mengukur akurasi sistem yang memiliki kecepatan komputasi tercepat dengan beberapa skenario yang akan diujikan
2. Menganalisis nilai parameter pada *Gaussian Mixture Model*
3. Menganalisis nilai parameter pada *multiple person classification*

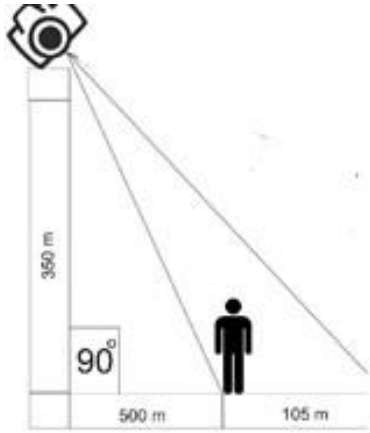
#### 4.1.2 Data Uji

Data uji dalam tugas akhir ini menggunakan data video yang diperoleh dari rekaman kamera di gedung LC dengan jarak dan sudut. Pengambilan video. Data video yang dihasilkan memiliki resolusi 800 x 592 piksel dengan format avi. Data uji yang digunakan pada penelitian ini adalah sebagai berikut :

Tabel 4-1 Data Uji

No.	Video	Data
1	Satu orang berjalan untuk mengantri, tidak ada antrian dibelakangnya	Skenario 1
2	Enam orang berjalan satu persatu untuk mengantri tanpa oklusi	Skenario II

3	Enam orang berjalan satu persatu untuk mengantri dengan dempetan	Skenario III
4	Kucing berjalan pada area perhitungan	Skenario IV



Gambar 4-1 Pengambilan video

### 4.1.3 Skenario Uji

Pada penelitian kali ini terdapat – sekenario uji yang akan dilakukan terhadap sistem perhitungan orang yaitu antara lain, pengujian GMM, pengujian penambahan shadow removal Chromatic Color Model, pengujian parameter *multiple person classification* sehingga didapat pengujian perhitungan orang secara keseluruhan di akhir.

### 4.1.4 Pengujian GMM

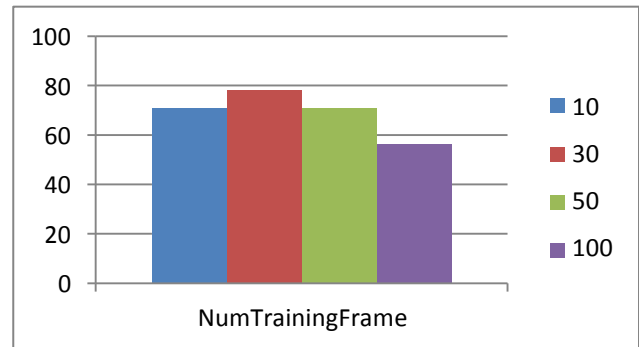
Tingkat hasil adaptasi terhadap *background* akan dilihat dari *foreground* yang terbentuk pada masing-masing skenario uji.

Tabel 4-2 Parameter GMM

Parameter	Keterangan
NumTrainingFrames	Jumlah <i>training frame</i>
NumGaussians	Jumlah distribusi <i>Gaussian</i>
MinimumBackgroundRatio	<i>Threshold</i> model <i>background</i>

Tabel 4-3 Hasil Uji NumTraingFrames

Data	NumTrainingFrame	Hasil Adaptasi
Skenario 1	10	Baik
	30	Baik
	50	Cukup
	100	Cukup

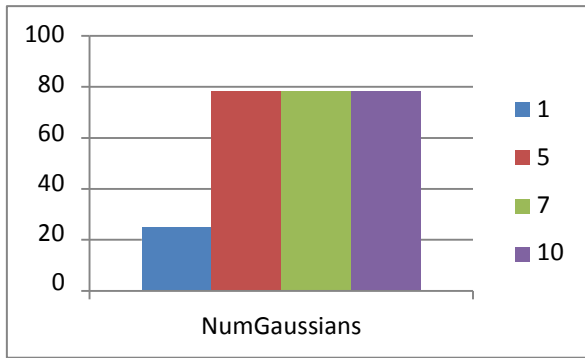


Gambar 4-2 Grafik NumTrainingFrame

Apabila dilihat dari hasil uji diatas, parameter *NumTrainingFrames* yang diambil adalah sebesar 30. Dapat disimpulkan bahwa pada kasus ini, parameter *numtrainingframes* diambil dengan nilai yang tidak terlalu kecil namun tidak terlalu besar pula. Semakin banyak frame yang diambil untuk training model background, maka *noise* berupa titik *foreground* yang ditimbulkan akan semakin banyak pula dikarenakan gerakan minor yang ada pada model *background*.

Tabel 4-4 Hasil Uji NumGaussians

Data	NumGaussians	Hasil Adaptasi
Skenario 1	1	Sangat buruk
	5	Baik
	7	Baik
	10	Baik

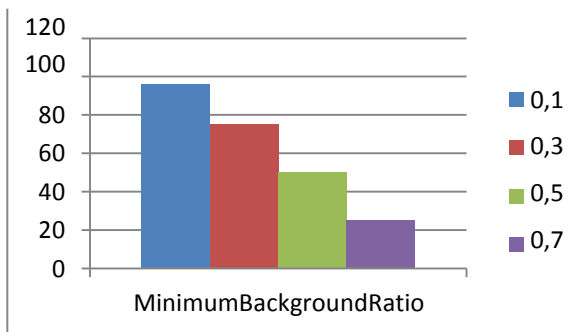


Gambar 4-3 Grafik NumGaussians

Apabila dilihat dari hasil uji diatas, parameter *NumGaussians* yang tergolong baik adalah 10. Nilai 10 diambil karena semakin besar nilai dari *numgaussians*, maka semakin baik GMM untuk memodelkan banyak *background*.

Tabel 4-5 Hasil Uji MinimumBackgroundRatio

Data	MinimumBackgroundRatio	Hasil Adaptasi
Skenario 1	0,1	Baik
	0,3	Cukup
	0,5	Buruk
	0,7	Sangat buruk



Gambar 4-4 Grafik MinimumBackgroundRatio

Apabila dilihat dari hasil uji diatas, parameter *MinimumBackgroundRatio*, hasil yang paling bagus adalah bernilai 0,1. Karena semakin kecil nilai *MinimumBackgroundRatio*, maka *foreground* yang diam akan semakin lama untuk menghilang karena dimodelkan sebagai *background*.

#### 4.1.5 Pengujian Chromatic Color Model

Untuk menguji apakah *shadow removal* dengan chromatic color model bisa mengurangi *noise* bayangan pada *foreground*, maka diuji perbandingan *width* dan *height* dari sebelum menggunakan *shadow removal* dengan setelah *shadow removal*.

Tabel 4-6 Perbandingan lebar boundingbox

frame	width dengan shadow removal	width tanpa shadow removal
261	64	93
269	66	130
400	67	129
474	109	144
508	88	100
524	99	124

Dengan menggunakan *shadow removal*, lebar *bounding box* yang terlalu besar karena *noise* bayangan bisa berkurang dan meminimalisasi terjadinya oklusi karena *bounding box* yang terlalu besar akibat bayangan yang ada pada *foreground*.

#### 4.1.6 Pengujian Multiple Person Classification

Berdasarkan persamaan 3.1, untuk menentukan apakah blob yang dihasilkan dari *background subtraction* menunjukkan 1 orang atau lebih, perlu diketahui blob yang diteliti, berapakah nilai *width* (W) minimum, *height* (H) minimum, *width* maksimum dan *height* maksimum sehingga blob yang diteliti berada pada range *Evaluation parameter*. Setelah diketahui berapa minimum dan maksimum dari *width* dan *height*, didapat persamaan 3,2 yang menunjukkan rentang untuk menghitung 1 orang.

Tabel 4-7 Ukuran untuk 1 orang

width		Height	
Minimum	maksimum	minimum	maksimum
51	125	215	250

Dari data diatas maka bisa ditentukan untuk  $=1$ , nilai dari  $= 51$  dan  $=125$  untuk *width*, dan  $= 215$  dan  $=250$  untuk *height*. *Evaluation parameter* berada diantara dan . Lalu untuk menentukan  $=2$  (2



orang), maka menurut persamaan 3.3, nilai dari diubah menjadi untuk rentang atas dan nilai sebelumnya untuk batas bawah sehingga didapat untuk =2, nilai dari = 125 dan =200 untuk *width*, dan = 250 dan = 400 untuk *height*. Namun apabila ada 2 blob orang bersatu, *height* tidak otomatis bertambah karena pada kasus ini, tinggi blob tetap sama sehingga *width* yang perlu diperhatikan. Dan terakhir untuk =3, nilai dari = 200 dan =350 untuk *width*.

Tabel 4-8 tabel uji parameter multiple person classification

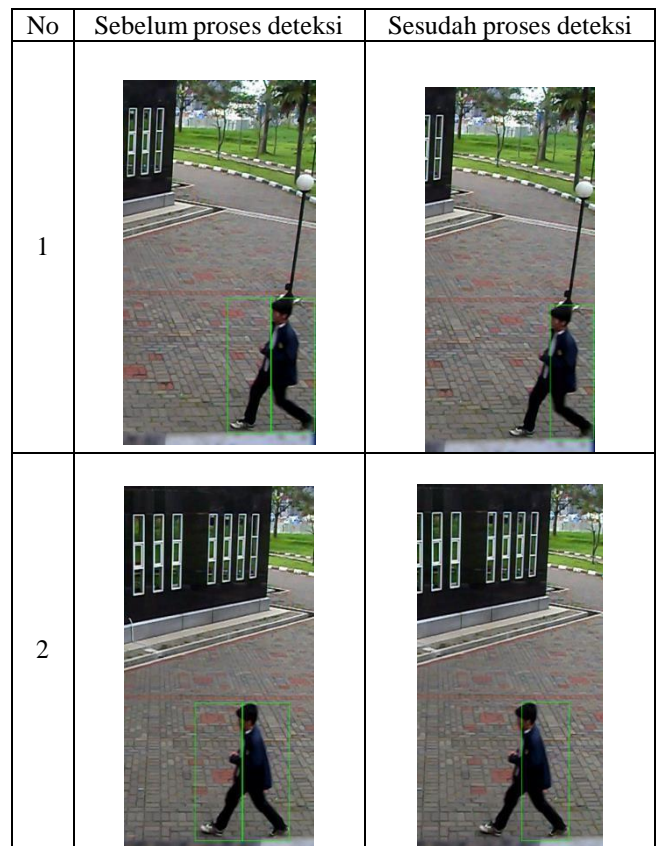
Video Skenario	Jumlah orang	Terhitung	Status	Persentase
1 orang dalam antrian	1	1	TRUE	100%
		1	TRUE	
		1	TRUE	
6 orang dalam antrian tanpa oklusi	1	1	TRUE	100%
		1	TRUE	
		1	TRUE	
	2	2	TRUE	100%
		2	TRUE	
		2	TRUE	
3	3	TRUE	100%	
	3	TRUE		
	3	TRUE		
6 orang dalam antrian ada oklusi	1 orang oklusi	1	TRUE	100%
		1	TRUE	
		1	TRUE	
	2 orang oklusi	2	TRUE	66,6%
		2	TRUE	
		1	FALSE	
	3 orang oklusi	3	TRUE	33,3%
		2	FALSE	
		2	FALSE	

Untuk menguji nilai rentang yang sudah ditentukan, maka diberikan uji pada video skenario. Dilihat dari tabel diatas maka pada percobaan dengan parameter yang ditentukan, untuk kasus tanpa oklusi atau *blob* tidak terlalu berdekatan maka menghasilkan nilai akurasi

pengenalan orang sebesar 100%. Namun saat uji parameter dengan kasus antrian ada oklusi, maka seringkali jika orang terlalu berdekatan mengakibatkan 2 orang yang dinilai bersatu dan masih masuk kedalam range 1 orang sehingga dinilai bahwa 2 orang tersebut hanya terhitung 1.

Dalam klasifikasi rentang orang, dilakukan proses menyeleksi kondisi untuk menangani kasus dimana saat orang sedang berjalan, maka lebar dari blob pun akan semakin membesar dan dapat menimbulkan perhitungan orang yang salah seperti yang dijelaskan pada bab 3 pada bagian *Multiple Person Classification*. Berikut gambar sebelum dan sesudah melakukan proses klasifikasi pada *bounding box*.

Tabel 4-9 Proses deteksi blob pada Multiple Person Classification





**4.1.7 Pengujian sistem keseluruhan**

Pada pengujian sistem secara keseluruhan, sistem dijalankan dengan semua komponen yang dibangun sehingga menghasilkan data sebagai berikut :

*Tabel 4-10 Tabel uji sistem tanpa oklusi*

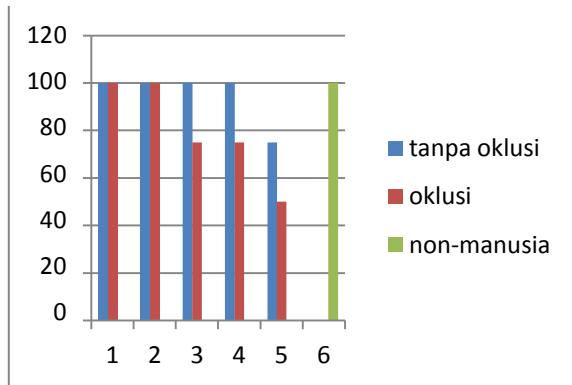
Jumlah orang	Frame	Terhitung	Status	Persentase
1	283	1	TRUE	100%
	329	1	TRUE	
	342	1	TRUE	
	359	1	TRUE	
2	476	2	TRUE	100%
	487	2	TRUE	
	490	2	TRUE	
	1471	2	TRUE	
3	568	3	TRUE	100%
	665	3	TRUE	
	705	3	TRUE	
	1361	3	TRUE	
4	632	4	TRUE	100%
	673	4	TRUE	
	692	4	TRUE	
	713	4	TRUE	
5	770	4	FALSE	75%
	932	5	TRUE	
	1005	5	TRUE	
	1104	5	TRUE	

*Tabel 4-11 Hasil uji objek non-manusia*

jumlah objek	frame	terhitung	status	Persentase
1 kucing	530	0	TRUE	100%
	580	0	TRUE	
	650	0	TRUE	
	800	0	TRUE	

*Tabel 4-12 Hasil uji sistem ada oklusi*

jumlah orang	Frame	terhitung	status	Persentase
1	237	1	TRUE	100%
	304	1	TRUE	
	358	1	TRUE	
	1270	1	TRUE	
2	405	2	TRUE	100%
	441	2	TRUE	
	381	2	TRUE	
	1161	2	TRUE	
3	491	3	TRUE	75%
	540	3	TRUE	
	752	2	FALSE	
4	1094	3	TRUE	75%
	575	4	TRUE	
	608	4	TRUE	
5	619	4	TRUE	50%
	917	3	FALSE	
	665	5	TRUE	
5	754	4	FALSE	50%
	819	4	FALSE	
	879	5	TRUE	



Gambar 4-5 Grafik hasil pengujian sistem keseluruhan

Adapun pengujian sistem dengan variasi jarak awal dari kamera ditambah 100 cm. Dan untuk data 1 orang didapat Saat jarak awal, didapat data untuk =1, nilai dari = 51 dan =125 untuk *width*, dan = 215 dan =250 untuk *height*. *Evaluation parameter* berada diantara dan . Lalu untuk menentukan =2 (2 orang), maka menurut persamaan 3.3, nilai dari diubah menjadi untuk rentang atas dan nilai sebelumnya untuk batas bawah sehingga didapat untuk =2, nilai dari = 125 dan =200 untuk *width*, dan = 250 dan = 400 untuk *height*. Namun apabila ada 2 blob orang bersatu, *height* tidak otomatis bertambah karena pada kasus ini, tinggi blob tetap sama sehingga *width* yang perlu diperhatikan. Dan terakhir untuk =3, nilai dari = 200 dan =350 untuk *width*.

Tabel 4-13 Tabel Width untuk 1 orang pada variasi jarak kamera

Jarak	Sample	L1	L2	L3	L4	L5	L6
600	1	40	37	38	44	45	50
	2	88	88	112	68	125	76
	3	44	43	43	44	43	43
	4	68	114	76	111	67	113
	5	46	46	46	46	46	46
700	1	40	43	41	42	38	37
	2	40	41	40	37	35	36
	3	42	42	42	41	42	42
	4	40	38	39	40	39	39
	5	41	39	39	41	42	42

Tabel 4-14 Hasil Pengujian Pada jarak

Jarak	Sample	Status
600	1	False
	2	True
	3	False
	4	True
	5	False

700	1	False
	2	False
	3	False
	4	False
	5	False

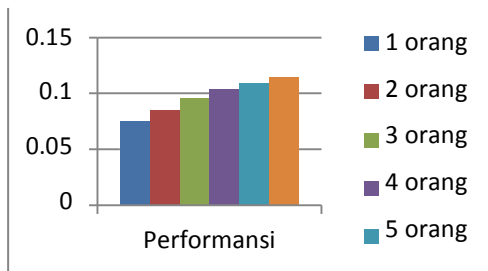
Setelah didapat data dari sample, bisa diambil kesimpulan bahwa untuk beberapa sample, rentang yang sudah ditentukan diawal masih bisa digunakan namun sudah tidak bisa digunakan untuk jarak 700 cm.

4.1.8 Pengujian performansi sistem

Berdasarkan pengujian performansi sistem menggunakan syntax tic tac pada matlab, maka didapatkan data sebagai berikut :

Tabel 4-15 Tabel Performa

No	Jumlah orang dalam video	Waktu proses (detik)
1	1	0,075
2	2	0,085
3	3	0,096
4	4	0,104
5	5	0,109
6	6	0,114



Gambar 4-6 Grafik performansi sistem

Dari tabel diatas, didapat bahwa untuk mengolah atau menghitung manusia, semakin banyak jumlah orang didalam video untuk dihitung, maka semakin lama atau meningkat waktu komputasi dari sistem.

## 5. Penutup

### 5.1 Kesimpulan

Setelah melakukan implementasi, pengujian dan analisis, maka dapat disimpulkan beberapa kesimpulan, yaitu :

1. Sistem menggunakan metode *Gaussian Mixture Model* untuk *background subtraction* untuk memisahkan *background* dengan *foreground*. Metode GMM bisa memisahkan dengan baik antara *foreground* dan *background* namun masih menyisakan *noise* bayangan pada *foreground* yang sudah dipisahkan.
2. Dengan menggunakan *shadow removal chromatic color model*, bayangan bisa diatasi meskipun tidak terlalu sempurna, namun sudah mengurangi lebarnya *blob* yang berlebihan dan menyebabkan oklusi.
3. Akurasi dari sistem sebesar 88%. Angka tersebut sudah mencapai angka yang diharapkan yaitu sebesar 85%. Alasan mengapa sistem tidak mencapai 100% karena masih ada masalah seperti oklusi dan *behavior* objek yang diteliti sehingga akurasi yang didapat hanya 88%.

### 5.2 Saran

Dalam kasus yang berkaitan dengan hal ini, maka peneliti mengajukan beberapa saran untuk pengembangan penelitian ini. Poin-poin saran yang diajukan oleh penulis adalah sebagai berikut:

1. Pada proses *background subtraction*, lebih baik menggunakan metode yang bisa mengatasi untuk *behavior foreground* yang diam, karena setelah dilakukan pada penelitian ini, dihasilkan bahwa dengan menggunakan *Gaussian Mixture Model*, model *background* akan selalu diupdate sehingga lama

kelamaan akan dikelompokkan kedalam *background* dan akurasi pun menurun.

2. Metode *multiple person classification* menggunakan aspek rasio untuk membedakan apakah terdapat satu orang atau lebih dalam suatu *bounding box*. Namun masalah timbul apabila dengan rasio yang digunakan, ada orang yang berdempetan dan menimbulkan oklusi sehingga perhitungan tidak akurat. Saran dari penulis adalah gunakan ekstraksi ciri lain sehingga pembagian lebih sempurna.

## Daftar Pustaka

- [1] Aik L.E., dan Zarita Zainuddin, 2009, *Real-Time People Counting System using Curve Analysis Method*, International Journal of Computer and Electrical Engineering Vol. 1 No.1
- [2] Barandiaran Javier, Berta Murguia, dan Fernando Boto, *Real-Time People Counting Using Multiple Lines*, Spain, Ninth International Workshop on Image Analysis for Multimedia Interactive Services
- [3] Bradsky, G., & Kaehler, A. 2008. *Learning OpenCV Computer Vision with the OpenCV Library*. O'Reilly Media.
- [4] Chen CH, Tsong-Yi Chen, Da-Jinn Wang, dan Tsang-Jie Chen, 2011, *A Cost-Effective People-Counter for a Crowd of Moving People Based on Two-Stage Segmentation*, Taiwan, Journal of Information Hiding and Multimedia Signal Processing Vol. 3 No. 1
- [5] Hu Yaowu, Ping Zhou, dan Hao Zhou, 2011, *A New Fast and Robust Method Based on Head Detection for People Flow Counting System*, Hangzhou, International Journal of Information Engineering Vol. 1 No.1
- [6] Horprasert T., David Harwood, dan Larry S., 1999, *A Statistical Approach for Real-time Robust Background Substraction and Shadow Detection*, IEEE
- [7] Hou, YL, G.K.H. Pang, 2011, *People Counting and Human Detection in a Challenging Situation*, in: IEEE Transactions volume 41 issue 1.
- [8] Jhonson, S. 2006. Stephen Johnson on Digital Photography. O' Reilly
- [9] Kalita Sishir, Raheja J.L., 2012, *An Insight into the Algorithms on Real-Time People Tracking and Counting System*, India, International Journal of Computer Application Vol.46 No.5
- [10] Kumar, Rakesh, Tapesh Parashar, Gopal Verma, 2012, *Background Modeling and Substraction Based People Counting for Real Time Video Surveillance*,

- in: International Journal of Soft Computing and Engineering.
- [11] Larry, Davis. 2000. *Nonparametric Model for Background Subtraction*. University of Maryland.
- [12] O.Masoud and N.P. Papanikolopoulos, *A Novel Method for Tracking and Counting Pedestrians in Real-Time Using a Single Camera*, IEEE Trans.on Vehicular Technology, vol.50,no.5,pp.1267-1278,2001.
- [13] Raheja J.L., Sishir Kalita, Pallab Jyoti Dutta, Solanki Lovendra, 2012, *A Robust Real Time People tracking and Counting incorporating shadow detection and removal*, India, International Journal of Computer Application Vol. 46 No. 4
- [14] Rainbowharmony. 2010. *Sistem Ruang Warna*. [Online] Available at : [http://digilib.itelkom.ac.id/index.php?option=com\\_content&view=article&id=668:sistem-ruang-warna&catid=15:pemrosesan-sinyal&Itemid=14/](http://digilib.itelkom.ac.id/index.php?option=com_content&view=article&id=668:sistem-ruang-warna&catid=15:pemrosesan-sinyal&Itemid=14/) [Accessed 28 Maret 2014]
- [15] Reynolds Douglas, *Gaussian Mixture Models*, USA, MIT Lincoln Laboratory
- [16] Rostianingsih, Silvia, Rudi Adipranata, Fredy S. 2009. *Adaptive Background dengan Metode Gaussian Mixture Models untuk Realtime Tracking*. Surabaya. Univ Kristen Petra
- [17] Smith, Steven W. 1997. *The Scientist and Engineer's Guide to Digital Signal Processing*
- [18] Schlogl T., B. Wachmann, W. Kropatsch, H. Bischof, 2001, *Evaluation of People Counting Systems*, Austria, Vienna Technical University
- [19] Yam Kin-Yi, Wan-Chi Siu, Ngai-Fong Law, Chok-Ki Chan, 2011, *Effective bi-directional people flow counting for real time surveillance system*, Hongkong, IEEE International C