

## THE CALCULATION SYSTEM OF NUMBER OF PEOPLE IN A ROOM BASED ON HUMAN DETECTION USING HAAR-CASCADE CLASSIFIER

Gusti Ngurah Rama Putra Atmaja<sup>1</sup>, Koredianto Usman<sup>2</sup>, Muhammad Ary Murti<sup>3</sup>

<sup>1,2</sup>Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom, Indonesia

<sup>3</sup>Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom, Indonesia

Email: [ramaputraatmaja62@gmail.com](mailto:ramaputraatmaja62@gmail.com), [korediantousman@telkomuniversity.ac.id](mailto:korediantousman@telkomuniversity.ac.id),  
[arymurti@telkomuniversity.ac.id](mailto:arymurti@telkomuniversity.ac.id)

(Naskah masuk: 7 Februari 2021, diterima untuk diterbitkan: 14 Februari 2021)

### Abstract

Data of number of people in the room, calculations are usually carried out by assigning someone to oversee a room. In this final project, a system for calculating the number of people in the room is designed with image processing based on human detection that can be used in rooms, both for commercial applications and for security. This system uses Raspberry Pi device that already has an image processing method Haar-Cascade Classifier. Input data is in the form of video taken directly via webcam to be captured into a frame so that it can be used as an input the Haar-Cascade Classifier method and perform the counting process will be sent to the Antares platform. The system design has been tested with five scenarios. Scenario 1 the effect of the distance of the object, scenario 2 the effect of the pose of the object, scenario 3 the effect of the amount the object in the frame, scenario 4 affects the scale factor and scenario 5 measurement computation time. Scenarios 1 to 3 will do the best configuration for minimum neighbour. The system gets the best accuracy of 98,5% when the object distance 4 meters, the best accuracy of 96,6% when the object is facing forward and accuracy the best is 97,7% when the object in the frame is more than two objects with the best configuration use the minimum neighbour 5. Scenario 4 gets accuracy the best is 76,2% when using the scale factor 1.1. Scenario 5 gets the average computation time of the system is under one second, meaning the detection process done pretty fast.

**Keywords:** image processing, Haar-Cascade Classifier, human detection.

## SISTEM PERHITUNGAN JUMLAH ORANG DALAM RUANGAN BERDASARKAN DETEKSI MANUSIA MENGGUNAKAN HAAR-CASCADE CLASSIFIER

### Abstrak

Data jumlah orang dalam ruangan biasanya diperoleh dengan menugaskan seseorang untuk mengawasi sebuah ruangan. Pada penelitian ini dirancang sistem perhitungan jumlah orang dalam ruangan dengan *image processing* berdasarkan deteksi manusia yang dapat digunakan pada ruangan, baik untuk aplikasi komersial maupun untuk keamanan. Sistem ini menggunakan perangkat *Raspberry Pi* yang sudah terdapat proses pengolahan citra dengan metode *Haar-Cascade Classifier*. Data masukan berupa *video* yang diambil langsung melalui *webcam* yang akan ditangkap menjadi *frame* sehingga bisa digunakan sebagai *input* metode *Haar-Cascade Classifier* dan melakukan proses perhitungan yang akan dikirim ke *platform Antares*. Rancangan sistem yang dibuat telah diujikan dengan lima skenario. Skenario 1 pengaruh jarak objek, skenario 2 pengaruh pose objek, skenario 3 pengaruh jumlah objek dalam *frame*, skenario 4 pengaruh *scale factor* dan skenario 5 pengukuran waktu komputasi. Skenario 1 sampai 3 akan dilakukan konfigurasi terbaik terhadap *minimum neighbour*. Sistem mendapat akurasi terbaik sebesar 98,5% saat jarak objek 4 meter, akurasi terbaik sebesar 96,6% saat objek menghadap depan dan akurasi terbaik sebesar 97,7% saat objek dalam *frame* lebih dari dua dengan konfigurasi terbaik menggunakan nilai *minimum neighbour* 5. Skenario 4 mendapat akurasi terbaik sebesar 76,2% saat menggunakan nilai *scale factor* 1,1. Skenario 5 mendapatkan waktu komputasi rata-rata sistem dibawah satu detik, artinya proses deteksi dilakukan cukup cepat.

**Kata kunci:** image processing, Haar-Cascade Classifier, human detection.

### 1. PENDAHULUAN

Perkembangan teknologi yang pesat membuat manusia tidak kehabisan ide untuk membuat

perangkat yang dapat memecahkan masalah sehari-hari dengan memanfaatkan fungsi dari sebuah perangkat yang sudah ada seperti kamera, komputer

dan perangkat teknologi lainnya. Perangkat tersebut dapat menyelesaikan masalah kecil maupun besar, baik dalam bidang keamanan, bidang bisnis maupun bidang manajemen. Salah satu contoh adalah alat yang mempermudah kehidupan manusia adalah alat pengumpulan data jumlah orang dalam ruangan pada periode waktu tertentu. Data jumlah orang dalam ruangan bermanfaat untuk sistem pengawasan, analisis data dan lain sebagainya. Untuk memperoleh data tersebut biasanya perhitungan dilakukan secara manual dengan menugaskan seseorang untuk mengawasi ruangan. Cara tersebut tentu saja sangat tidak efektif dan efisien. Untuk menyelesaikan permasalahan *human error* atau kelalaian manusia dalam pengawasan disuatu ruangan, sebuah ruangan dipasang kamera pengawas menggunakan *Closed Circuit Television* (CCTV). Hal ini juga masih kurang efektif, karena CCTV hanya mengambil gambar dan *video* yang mampu memantau secara pasif tanpa adanya pemberitahuan lebih lanjut [1].

Dari permasalahan tersebut, perlu dikembangkan teknologi *image processing* yang mampu mendeteksi objek secara *realtime* [2]. *Object detection* merupakan sebuah proses pengenalan objek yang diamati pada tiap *frame* dari pengambilan gambar atau *video* yang dilakukan oleh *computer vision*. Diperlukan suatu pendekatan untuk dapat melakukan *object detection*. Pendekatan yang dapat dilakukan adalah melakukan pendeteksian objek manusia melalui *frame* yang ditangkap oleh sebuah *webcam*. Pada Penelitian ini akan dirancang sebuah sistem perhitungan jumlah orang dalam ruangan berdasarkan deteksi tubuh bagian atas manusia berbasis *Raspberry Pi*. Deteksi tubuh bagian atas manusia merupakan salah satu proses untuk melakukan perhitungan jumlah orang yang berada dalam sebuah ruangan dalam jangkauan kamera. Keunggulan mendeteksi tubuh bagian atas manusia merupakan bagian tubuh yang mudah diidentifikasi baik dalam kondisi perorangan maupun sekumpulan orang. Tubuh bagian atas manusia juga mudah dipetakan dimana terdapat kepala yang berada antara bahu kiri dan bahu kanan.

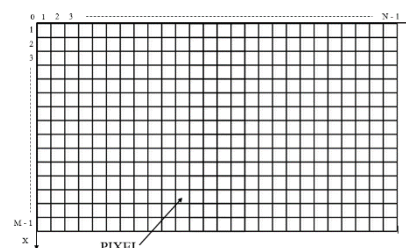
Pengembangan mengenai deteksi objek manusia selalu diteliti. Pada penelitian [3] menggunakan metode *Background Substraction* didapat hasil akurasi tertinggi sebesar 75% untuk satu orang masuk bersamaan dan 71% untuk beberapa orang yang masuk secara bersamaan sedangkan pada penelitian [4] menggunakan metode *You Only Look Once* untuk mendeteksi jumlah orang dalam transportasi umum didapatkan hasil akurasi deteksi tertinggi 69%. Hal ini yang menjadi latar belakang penelitian ini karena penulis ingin meningkatkan akurasi deteksi objek manusia dalam ruangan. Oleh karena itu, pada penelitian ini akan dirancang sebuah sistem perhitungan jumlah orang dalam sebuah ruangan menggunakan metode *Haar-Cascade Classifier* sebagai algoritma pendeteksi objek yang sudah pernah diimplementasikan pada citra *thermal*

[5]. Pada penelitian [5] dijelaskan bahwa metode *Haar-Cascade Classifier* merupakan algoritma yang memiliki waktu komputasi yang cukup cepat dan memiliki nilai presisi dan *recall* yang cukup baik. Berdasarkan hasil tersebut, penelitian ini akan menggunakan metode *Haar-Cascade Classifier* yang akan diimplementasikan untuk sistem deteksi objek secara *realtime*.

## 2. METODE PENELITIAN

### 2.1. Citra Digital

Citra digital adalah suatu gambar yang dapat diolah oleh komputer. Citra digital tersusun dari beberapa *pixel* yang mana tiap *pixel* mempresentasikan warna yang berbeda di suatu titik tertentu pada citra. *Pixel* ialah elemen terkecil dari suatu citra untuk menentukan resolusi dari citra tersebut [6]. Citra digital disimpan dalam berbagai format *file* berbeda. Format *file* tersebut antara lain jpg, jpeg, png, bmp dan lain-lain.



Gambar 1. Representasi Citra Digital

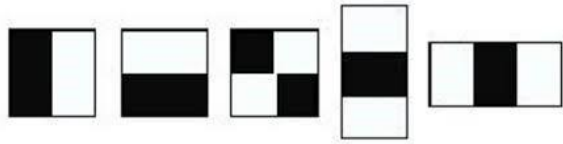
Citra digital merupakan matriks berukuran  $M \times N$  dimana elemen dari matriks fungsi intensitas cahaya. Kumpulan dari intensitas cahaya tersebut dinyatakan sebagai suatu fungsi kontinu  $f(x,y)$  dimana  $x$  dan  $y$  menyatakan indeks koordinat dari elemen matriks yang bersangkutan [7]. Matriks citra digital berukuran  $M \times N$  direpresentasikan pada Gambar 2.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Gambar 2. Matriks Citra Digital

### 2.2. Haar-Like Feature

*Haar-Like Feature* merupakan fitur yang memiliki dasar dari fungsi matematika *Haar-Wavelet*. *Haar-Wavelet* adalah gelombang yang memiliki satu interval tinggi dan satu interval rendah yang dikombinasikan untuk pendeteksian objek visual [8]. Dalam dua dimensi, gelombang digambarkan dengan sepasang persegi yang berdekatan seperti Gambar 3.



Gambar 3. Haar-Wavelet dalam Dua Dimensi

Pada umumnya *Haar-Like Feature* yang terdiri dari dua persegi panjang digunakan untuk *edge detection* (deteksi tepi) dan yang terdiri dari tiga persegi panjang akan digunakan untuk *line detection* (deteksi garis). Pada Gambar 4 merupakan ilustrasi citra masukan yang akan diekstraksi fitur dengan menggunakan *Haar-Like Feature*.



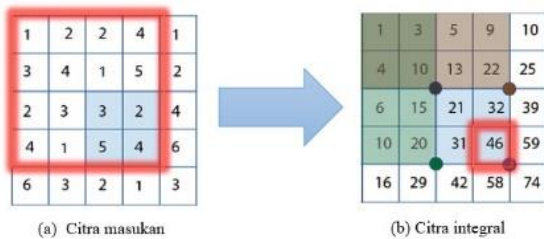
Gambar 4. Ilustrasi Ekstraksi Fitur Tubuh Manusia

Nilai fitur yang terhitung pada fitur *haar* akan menentukan daerah tersebut memiliki fitur tubuh bagian atas manusia atau tidak. Apabila nilai fitur *haar* lebih besar dari pada nilai *threshold* [9], maka daerah tersebut dinyatakan memiliki fitur tubuh bagian atas manusia.

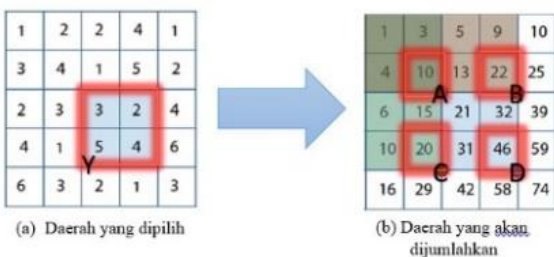
### 2.3. Integral Image

Teknik *integral image* dapat mempermudah proses citra yang diolah untuk menghitung hasil penjumlahan nilai *pixel* pada daerah yang terdeteksi oleh fitur *haar* [10]. Dari Gambar 5 ditunjukkan cara kerja *integral image*. Untuk menghitung matriks hasil *integral image* dapat digunakan Persamaan 1.

$$(x, y) = \sum_{x', y'}^{x, y} (x', y') \quad (1)$$



Gambar 5. Cara Kerja *Integral Image*



Gambar 6. Area *Pixel* yang Dihitung

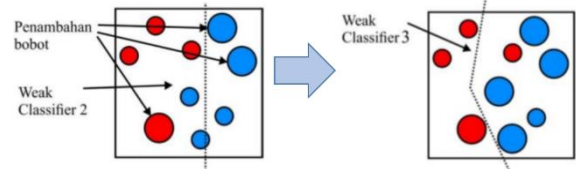
$(x, y)$  adalah nilai dari koordinat baris dan kolom suatu matriks. Sehingga dalam penggunaan *integral image* kinerja menjadi lebih efisien yang dijelaskan pada Gambar 6. Untuk menghitung nilai *pixel* dalam area yang sudah ditentukan digunakan Persamaan 2 [11].

$$Y = A + D - (B + C) \quad (2)$$

$Y$  adalah nilai area yang didapat menggunakan metode *integral image* dari variabel  $A, B, C,$  dan  $D$  yang merupakan koordinat pembatas area tersebut.

### 2.5. Adaptive Boost

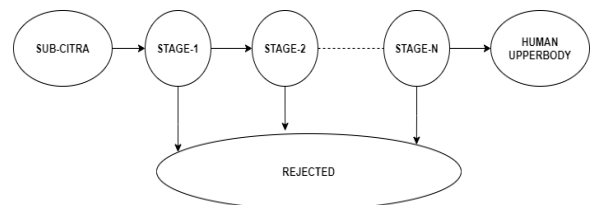
*AdaBoost* bertujuan untuk melakukan seleksi secara spesifik terhadap fitur yang dianggap penting pada beberapa *classifier* yang telah dibentuk [12]. *AdaBoost* memiliki rangkaian filter yang cukup efisien untuk menggolongkan daerah pada suatu gambar. Rangkaian filter tersebut terdiri dari *AdaBoost classifier* yang terbentuk dari gabungan *classifier* lemah. Suatu *classifier* dikatakan lemah jika tidak dapat memenuhi target klasifikasi yang telah ditentukan sebelumnya. *Classifier* lemah tersebut menetapkan suatu bobot sehingga apabila digabungkan akan menjadi satu *classifier* yang kuat [13].



Gambar 7. Proses Penambahan Bobot

### 2.4. Cascade Classifier

*Cascade Classifier* merupakan metode klasifikasi bertingkat yang bertugas untuk menolak area gambar yang tidak terdeteksi manusia dengan menggunakan *classifier* tiap tingkatan klasifikasinya [14]. Gambar 8 merupakan skema klasifikasi bertingkat pada *Cascade Classifier*.



Gambar 8. Skema *Cascade Classifier*

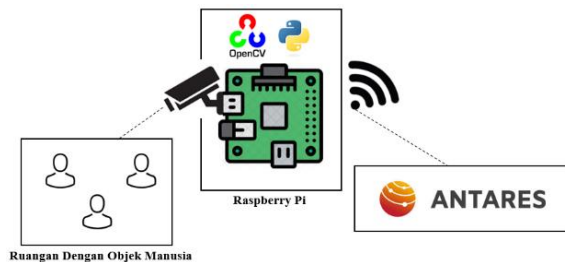
Pada klasifikasi tingkat pertama, tiap sub-citra akan diklasifikasi secara sederhana. Seiring dengan bertambahnya tingkat klasifikasi, maka diperlukan syarat yang lebih spesifik sehingga *classifier* yang digunakan menjadi lebih kompleks. Hal ini ditujukan agar dapat mengurangi kemungkinan

salah deteksi yang dilakukan oleh sistem (*False Positive*). Apabila terdapat sub-citra yang gagal dilewatkan pada salah satu tingkat klasifikasi, maka sub-citra tersebut digolongkan sebagai bukan manusia. Namun apabila semua tingkat klasifikasi yang ada dalam rangkaian *cascade classifier* terlewati, maka area sub-citra tersebut dianggap memiliki fitur atau ciri sebagai objek manusia.

### 3. ALUR KERJA PENELITIAN

#### 3.1. Desain Sistem

Sistem yang akan dibangun merupakan sistem pendeteksi tubuh bagian atas manusia dengan *image processing* dan menggabungkannya dengan konsep *Internet of Things* yang berbasis pada *Raspberry Pi*. Sistem ini menggunakan metode *Haar-Cascade Classifier* sebagai algoritma dalam *image processing*. Pemodelan mengenai sistem yang akan dirancang pada Penelitian dapat dilihat pada Gambar 9.



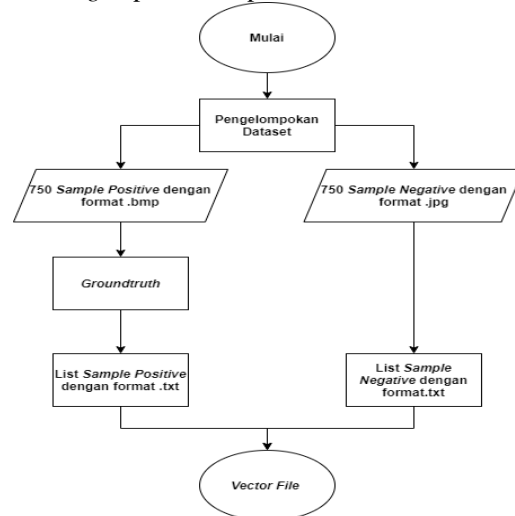
Gambar 9. Desain Sistem

*Webcam* yang dihubungkan dengan *Raspberry Pi* berfungsi untuk merekam keadaan ruangan dan melakukan *capture frame* sebagai *input* pengolahan citra pada *Raspberry Pi*. *Raspberry Pi* berfungsi sebagai pusat dari seluruh kerja sistem ini. Semua *frame* yang didapat dari *webcam* akan masuk ke tahap pengolahan citra dengan menggunakan algoritma *Haar-Cascade Classifier*. Selanjutnya data hasil olahan ini akan dikirimkan ke server dengan koneksi internet. *Antares* merupakan sebuah platform *Internet of Things* yang berfungsi sebagai *cloud* atau penampung data dari *Raspberry Pi* dalam bentuk *string*.

#### 3.2. Pre-processing

Pada tahap ini dilakukan persiapan untuk proses *training* dataset. Proses *training* memerlukan kumpulan *sample positive* dan *sample negative*. *Sample positive* merupakan kumpulan gambar yang berisikan objek manusia didalamnya sedangkan *sample negative* merupakan kumpulan gambar *background* atau latar tempat yang tidak terdapat objek manusia didalamnya. Dalam Penelitian ini penulis menggunakan 750 *sample positive* dan 750 *sample negative*. Pada *sample positive* akan dilakukan pemetaan daerah yang terdapat objek manusia (*groundtruth*) agar objek manusia lebih

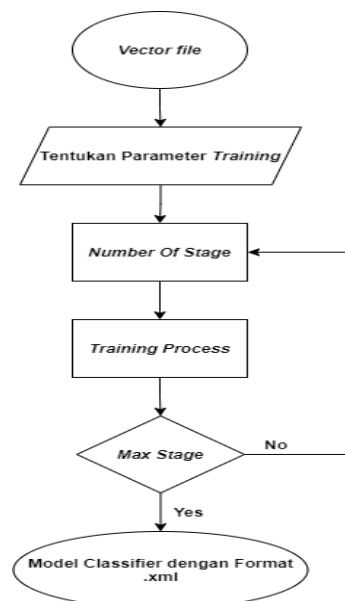
mudah dikenali. Setelah penentuan *groundtruth* pada *sample positive* maka seluruh *sample* data akan diubah kedalam bentuk *file .txt*. Skema proses *pre-processing* dapat dilihat pada Gambar 10.



Gambar 10. Skema Pre-Processing

#### 3.3. Training Dataset

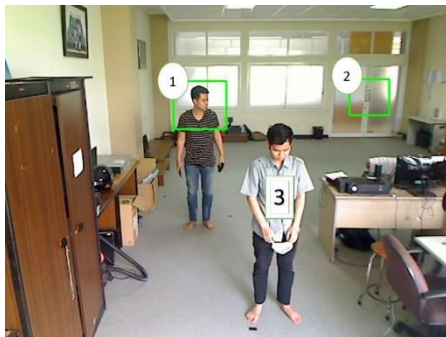
Pada tahap ini akan dilakukan proses *training* data *sample positive* dan *sample negative* yang telah dikumpulkan dalam *vector file*. Proses *training* memerlukan beberapa parameter seperti ukuran lebar dan tinggi *bounding box*, jumlah *sample* dataset yang digunakan dan jumlah *stage cascade classifier* yang akan digunakan. Hasil dari proses *training* merupakan model *classifier* yang memiliki spesifikasi sesuai parameter yang telah ditentukan. Model *classifier* terdiri dari kumpulan nilai *threshold* yang membentuk sebuah *file .xml* yang nantinya akan diujikan. Pada Penelitian ini model *classifier* menggunakan 18 tingkat klasifikasi dengan ukuran lebar dan tinggi *bounding box* adalah 20x22. Skema proses *training* dapat dilihat pada Gambar 11.



Gambar 11. Skema Training

### 3.4. Uji Hasil Prediksi

Setelah file .xml atau model classifier hasil dari proses training sistem diperoleh, maka langkah selanjutnya adalah melakukan prediksi menggunakan model classifier tersebut [15]. File .xml berisi semua informasi tentang fitur yang harus terpenuhi dalam setiap classifier. Pada proses ini model classifier akan dimasukkan kedalam sistem uji. Pada Gambar 12 ditunjukkan hasil pengujian dari model classifier. Kotak hijau dengan label nomor satu pada Gambar 12 adalah objek manusia yang terdeteksi oleh sistem (*True Positive*). Kotak hijau dengan label nomor dua pada Gambar 12 adalah salah deteksi yang dilakukan oleh sistem (*False Positive*) sedangkan objek dengan label nomor 3 adalah objek manusia yang tidak terdeteksi oleh sistem (*False Negative*).



Gambar 12. Frame Hasil Uji Prediksi

### 3.5. Parameter Performansi

#### 3.5.1 Akurasi

Akurasi didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual [16]. Akurasi merupakan rasio prediksi benar *True Positive* (TP) dengan keseluruhan data yaitu *True Positive* (TP), *False Positive* (FP) dan *False Negative* (FN). Akurasi secara matematis dapat dituliskan pada Persamaan 3.

$$A = \frac{TP}{TP+FP+FN} \times 100\% \quad (3)$$

#### 3.5.2 Presisi

Presisi merupakan rasio prediksi benar *True Positive* (TP) dibandingkan dengan keseluruhan hasil yang diprediksi positif *True Positive* (TP) dan *False Positive* (FP). Semakin besar nilai presisi maka nilai *False Positive* (FP) atau kemungkinan salah deteksi yang dilakukan oleh sistem semakin kecil. Presisi secara matematis dapat dituliskan pada Persamaan 4.

$$P = \frac{TP}{TP+FP} \times 100\% \quad (4)$$

#### 3.5.3 Recall

*Recall* merupakan rasio prediksi benar *True Positive* (TP) dibandingkan dengan keseluruhan data yang benar positif *True Positive* (TP) dan *False Negative* (FN). Semakin besar nilai *recall* maka nilai *False Negative* (FN) atau kemungkinan objek manusia yang tidak terdeteksi oleh sistem semakin kecil. *Recall* secara matematis dapat dituliskan pada Persamaan 5.

$$R = \frac{TP}{TP+FN} \times 100\% \quad (5)$$

#### 3.5.4 Waktu Komputasi

Waktu komputasi merupakan perhitungan seberapa cepat sistem melakukan pengenalan objek dalam satuan detik. Parameter ini sangat penting karena semakin sedikit waktu yang dibutuhkan sistem dalam mengenali objek, maka dapat diartikan sistem tersebut semakin mendekati waktu *realtime*. Waktu komputasi secara matematis dapat dituliskan pada Persamaan 6.

$$t_k = t_e - t_s \quad (6)$$

$t_s$  : Waktu saat proses sistem dimulai

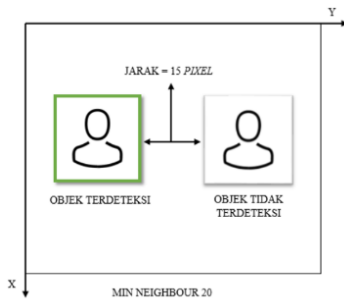
$t_e$  : Waktu saat proses sistem selesai

### 3.6. Parameter Optimalisasi

Parameter optimalisasi adalah suatu nilai yang dapat diubah pada suatu sistem yang dapat mempengaruhi hasil performa sehingga dapat diatur nilainya untuk mendapatkan hasil performa yang terbaik. Adapun 2 parameter optimalisasi yang dapat diatur yaitu nilai *minimum neighbour* dan *scale factor*.

#### 3.6.1 Minimum Neighbour

*Minimum Neighbour* merupakan parameter yang menentukan jarak yang harus dimiliki antar *bounding box*. *Minimum neighbour* merupakan jarak minimal antar *bounding box* yang diperlukan untuk mendeteksi objek [17]. Artinya, jika jarak *bounding box* objek terdeteksi kurang dari *minimum neighbour* maka objek yang terdeteksi pada posisi representasi *pixel* yang lebih besar akan diabaikan. Semakin besar nilai *minimum neighbour* maka nilai akurasi akan menurun sedangkan nilai presisi akan meningkat. Pada Gambar 13 merupakan contoh hasil deteksi saat menggunakan *minimum neighbour* 20 dengan jarak antara kotak hijau pendeteksi objek (*bounding box*) adalah 15 *pixel*.



Gambar 13. Perbandingan Penggunaan *Minimum Neighbour*

### 3.6.2 Scale Factor

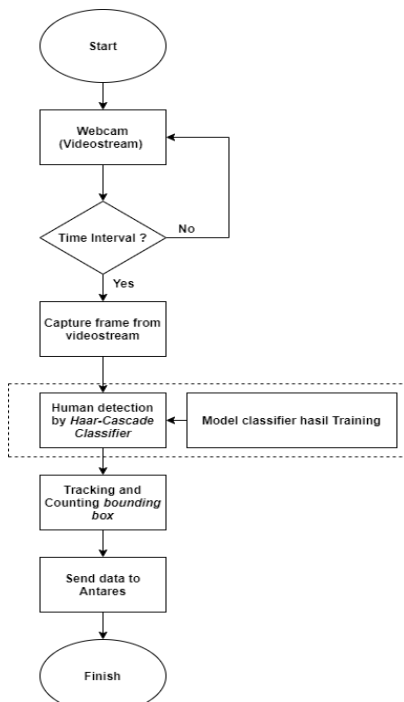
*Scale factor* merupakan parameter optimasi yang dapat menentukan seberapa besar ukuran citra dapat berkurang pada setiap skala [18]. Semakin tinggi nilai *scale factor* maka semakin kecil ukuran *frame* yang masuk sebagai *input* pengolahan citra. Pada Gambar 14 merupakan perubahan skala citra yang dipengaruhi oleh *Scale Factor*.



Gambar 14. Pengaruh *Scale Factor* Terhadap Skala Gambar

## 4. HASIL DAN PEMBAHASAN

Pada Gambar 15 merupakan alur kerja dari sistem yang dimulai dari *videostream webcam* dan akan dilakukan *capture frame* setelah waktu *interval* terpenuhi. Hasil akhir akan berupa *frame hasil capture* yang akan digunakan untuk analisis dan hasil perhitungan yang akan ditampilkan pada *Antares IoT Platform*.



Gambar 15. Alur Kerja Sistem

Perangkat lunak yang digunakan dalam rancangan sistem dapat dilihat pada Tabel 1 dan untuk spesifikasi dari perangkat keras yang digunakan dalam rancangan sistem dapat dilihat pada Tabel 2.

Tabel 1. Kebutuhan Perangkat Lunak

Software	Fungsi
Raspbian	Sebagai sistem operasi yang digunakan pada <i>Raspberry Pi</i>
Python	Bahasa pemrograman yang digunakan dalam pengolahan citra
OpenCV	Library yang menyediakan fungsi <i>Haar-Cascade Classifier</i>
Virtual Network Computing(VNC)	Aplikasi <i>remote</i> yang akan menghubungkan <i>Raspberry Pi</i> dengan Personal Computer

Tabel 2. Kebutuhan Perangkat Keras

Hardware	Spesifikasi
<i>Raspberry Pi 3B+</i>	ARM Cortex-A53,1,2GHz RAM 1 Gb 4 buah port USB 1 buah port MicroSD
Webcam	5Megapixels
Logitech C310	Output USB 2.0
MicroSD	Kapasitas 16 Gb
Personal Computer	Windows 10

### 4.1. Skenario Pengujian Sistem

Pada penelitian ini citra yang akan diujikan adalah citra dengan ukuran dimensi 600x400 pada ruang warna RGB yang diperoleh melalui proses *capture* dari *videostream* oleh *Raspberry Pi*. Terdapat 5 skenario pengujian, dimana setiap skenario akan melakukan analisis untuk mencari konfigurasi terbaik agar hasil skenario tersebut maksimal. Pada skenario 1, 2, 3 dan 5 akan menggunakan nilai *minimum neighbour* 3, 5, 10, 20 dan 50. Pada skenario 4 dan 5 akan menggunakan nilai *scale factor* 1,05; 1,1; 1,3 dan 1,5. Rincian dari skenario yang diujikan :

- Skenario 1 : Pengaruh jarak objek terhadap Akurasi, *Recall* dan Presisi. Terdapat 90 citra dalam data uji yang terbagi kedalam 3 kelas, yaitu 30 data dengan jarak objek 2 meter, 30 data dengan jarak objek 4 meter dan 30 data dengan jarak objek 6 meter. Skenario ini bertujuan untuk mengetahui pengaruh jarak objek terhadap kemampuan deteksi oleh sistem.
- Skenario 2 : Pengaruh pose objek terhadap Akurasi, *Recall* dan Presisi . Terdapat 90 citra dalam data uji yang terbagi kedalam 3 kelas, yaitu 30 data dengan objek hadap depan, 30 data dengan objek hadap samping dan 30 data dengan objek hadap belakang. Skenario ini bertujuan untuk mengetahui pengaruh pose objek dalam *frame* terhadap kemampuan deteksi oleh sistem.
- Skenario 3 : Pengaruh jumlah objek terhadap Akurasi, *Recall* dan Presisi. Terdapat 90 citra dalam data uji yang terbagi kedalam 3 kelas, yaitu 30 data dengan 1 objek di dalam *frame*, 30 data dengan 2 objek di dalam

frame dan 30 data dengan lebih dari 2 objek di dalam frame. Skenario ini bertujuan untuk mengetahui pengaruh jumlah objek yang terdapat dalam frame terhadap kemampuan deteksi oleh sistem.

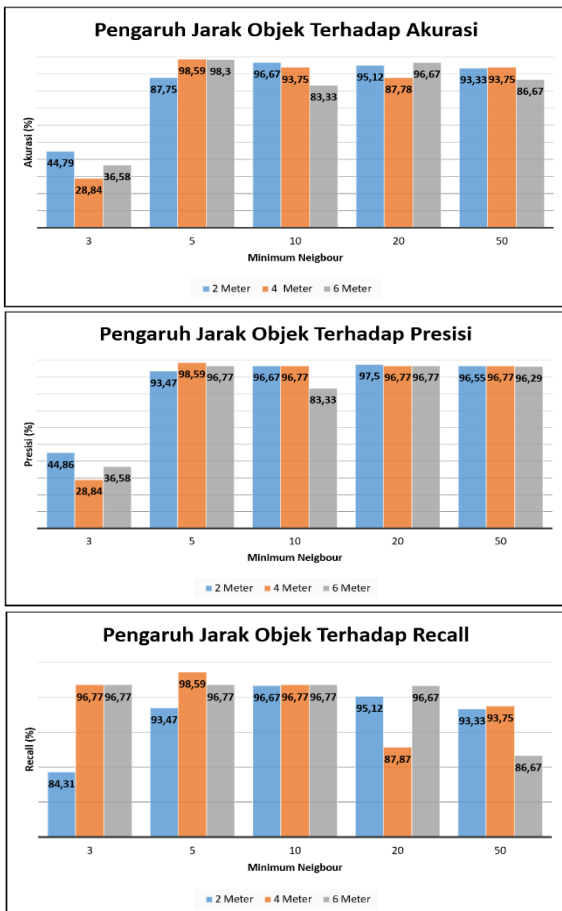
- Skenario 4 : Pengaruh *Scale factor* terhadap Akurasi, Presisi dan *Recall*. Terdapat 200 data uji yang diambil secara terus-menerus dan dihitung seberapa besar pengaruh *scale factor* dalam proses mendeteksi sebuah objek.
- Skenario 5 : Pengujian terhadap waktu komputasi. Terdapat 10 data uji yang diambil secara terus-menerus dan akan dihitung waktu komputasi rata-rata dari 10 data uji tersebut. Pengujian ini bertujuan untuk menguji seberapa cepat sistem dapat mendeteksi sebuah objek. Semakin kecil waktu yang diperlukan dalam mengenali objek, maka sistem akan semakin mendekati *realtime*.

4.2. Hasil Pengujian Sistem

Pada sub-bab ini akan diperlihatkan hasil dari 5 skenario yang telah diujikan.

4.2.1 Skenario 1

Akurasi, presisi dan *recall* yang dihasilkan pada skenario 1 dapat dilihat pada Gambar 16.

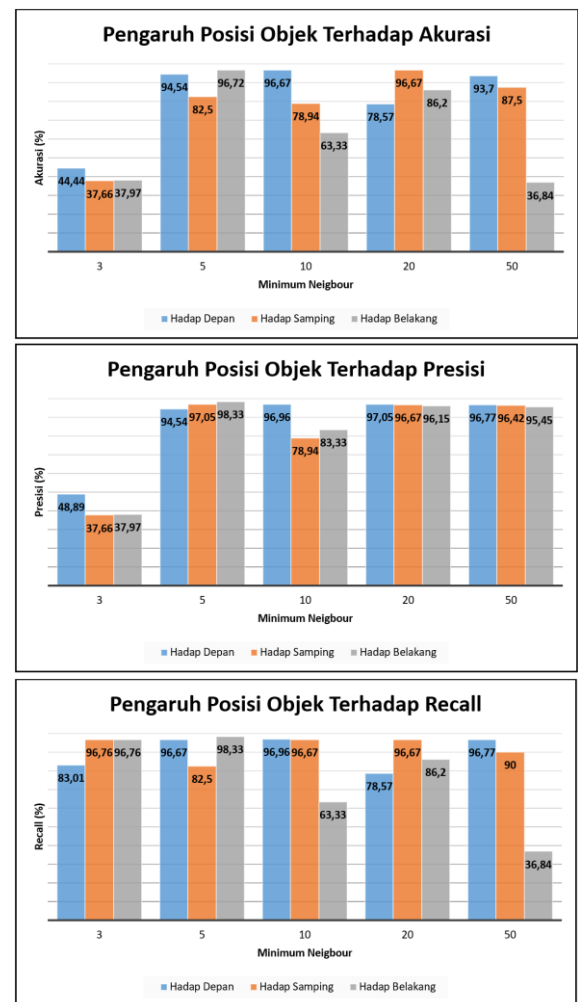


Gambar 16. Hasil Skenario 1

*Minimum neighbour 5* memiliki rata-rata akurasi 95,4%. Presisi dan *recall* saat menggunakan *minimum neighbour 5* mencapai presentase melebihi 95% saat objek berada pada jarak 4 meter dan 6 meter. Dari pengujian skenario 1 dapat dilihat bahwa sistem bekerja optimal pada jarak 4 meter dengan nilai *minimum neighbour 5* karena memiliki nilai akurasi 98,59%, presisi dan *recall* berada pada presentase yang melebihi 96%. Sistem juga dapat bekerja optimal pada jarak objek 6 meter dengan nilai *minimum neighbour 5* karena hasil pengujian akurasi 98,3%, presisi dan *recall* memiliki presentase 96%.

4.2.2 Skenario 2

Akurasi, presisi dan *recall* yang dihasilkan pada skenario 2 dapat dilihat pada Gambar 17.



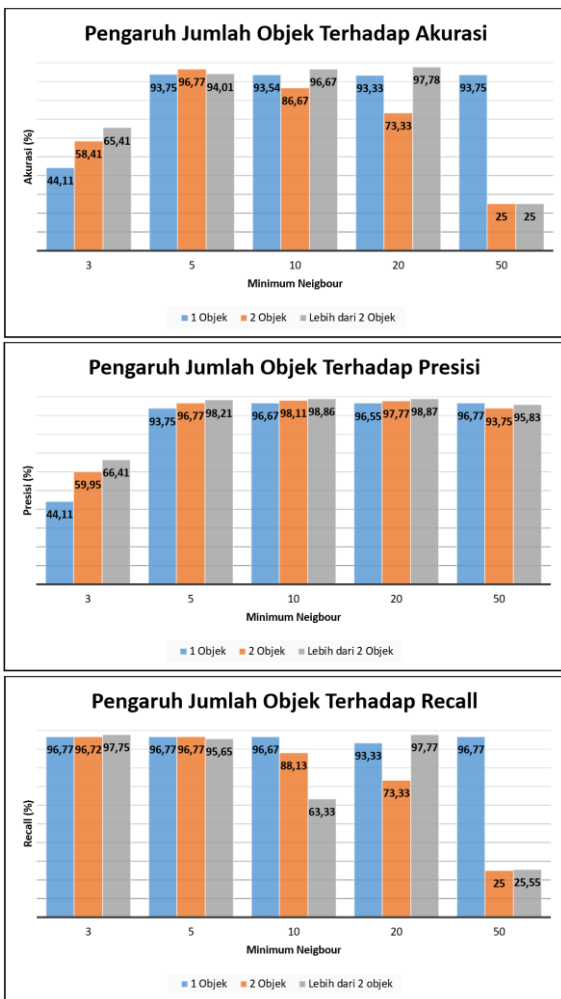
Gambar 17. Hasil Skenario 2

Setelah mendapatkan hasil dari akurasi, presisi dan *recall* dari pengujian skenario 2 dapat dilihat bahwa sistem bekerja optimal pada saat objek menghadap depan dengan menggunakan nilai *minimum neighbour 5*. Hal tersebut karena hasil akurasi, presisi dan *recall* saat menggunakan nilai *minimum neighbour 5* sangat konsisten dari pada nilai

*minimum neighbour* lain. Walaupun nilai *minimum neighbour* 5 tidak pernah mencapai akurasi optimal, rata-rata akurasi yang dihasilkan lebih besar dari *minimum neighbour* lain yaitu sebesar 91,2%.

**4.2.3 Skenario 3**

Akurasi, presisi dan *recall* yang dihasilkan pada skenario 3 dapat dilihat pada Gambar 18.



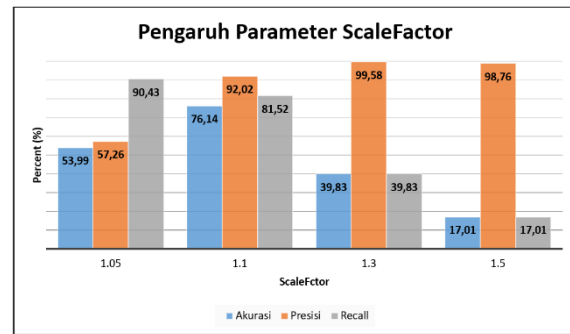
Gambar 18. Hasil Skenario 3

Setelah mendapatkan hasil dari akurasi, presisi dan *recall* dari pengujian skenario 3 dapat dilihat bahwa sistem bekerja optimal pada saat menggunakan nilai *minimum neighbour* 5. Pada skenario 3 membuktikan bahwa nilai *minimum neighbour* 5 memiliki akurasi yang konsisten dalam mendeteksi objek dalam *frame* dengan rata-rata presentase akurasi mencapai 94,8%, rata-rata presentase presisi mencapai 96,2% dan rata-rata presentase *recall* mencapai 98,5%.

Skenario 3 merupakan skenario paling penting pada penelitian ini. Hasil dari skenario 3 sangat berpengaruh untuk menentukan performansi terbaik sistem dalam mendeteksi objek manusia dalam sebuah ruangan. Hal tersebut karena kualitas perhitungan sangat berpengaruh pada kualitas deteksi

**4.2.4 Skenario 4**

Akurasi, presisi dan *recall* yang dihasilkan pada skenario 4 dapat dilihat pada Gambar 19.



Gambar 19. Hasil Skenario 4

Dapat dilihat bahwa nilai *scale factor* 1,1 menghasilkan akurasi paling baik. Presisi dan *recall* yang dihasilkan saat menggunakan nilai *scale factor* 1,1 dirasa paling konsisten dari nilai *scale factor* lain karena memiliki nilai presisi dan *recall* diatas 80%. Menggunakan nilai *scale factor* yang lebih besar dari 1,1 akan meningkatkan nilai presisi namun menurunkan nilai *recall* sedangkan jika menggunakan nilai *scale factor* yang lebih kecil dari 1,1 akan meningkatkan nilai *recall* dan menurunkan nilai presisi. Sehingga penggunaan *scale factor* 1,1 akan membuat sistem dapat bekerja secara optimal.

**4.2.5 Skenario 5**

Waktu komputasi rata-rata terhadap perubahan parameter *minimum neighbour* dapat dilihat pada Tabel 3.

<i>Minimum Neighbour</i>	Rata-rata Waktu Komputasi
3	0,7669 detik
5	0,7603 detik
10	0,7558 detik
20	0,5812 detik

Waktu komputasi rata-rata terhadap perubahan parameter *scale factor* dapat dilihat pada Tabel 4.

<i>Scale Factor</i>	Rata-rata Waktu Komputasi
1,05	0,8099 detik
1,1	0,4846 detik
1,3	0,3240 detik
1,5	0,2808 detik

Pada Tabel 3 dan Tabel 4 dapat dilihat bahwa rata-rata waktu komputasi memiliki nilai yang lebih kecil dari satu detik sehingga sistem dianggap mampu mengeluarkan data secara *realtime*. Waktu komputasi tercepat terjadi saat menggunakan *minimum neighbour* 50 dan *scale factor* 1,5, hal tersebut terjadi sebab banyaknya *True Positive* yang dihilangkan karena tidak memenuhi syarat minimal ketetapan sehingga mempersingkat waktu deteksi. Semakin kecil nilai *minimum neighbour* dan *scale factor* yang



digunakan maka waktu komputasi akan semakin besar. Waktu komputasi juga dapat dipengaruhi oleh spesifikasi komponen yang digunakan.

## 5. KESIMPULAN

Berdasarkan pada hasil simulasi dan analisis penelitian yang telah dilakukan dapat ditarik kesimpulan. Sistem dapat mengenali objek dalam jarak 2 meter, 4 meter dan 6 meter dengan konfigurasi terbaik menggunakan nilai *minimum neighbour* 5 dengan akurasi rata-rata 95,4%. Sistem dapat mengenali objek saat menghadap depan, samping dan belakang dengan konfigurasi terbaik menggunakan nilai *minimum neighbour* 5 dengan akurasi rata-rata 91,2%. Sistem dapat mengenali 1 objek, 2 objek dan lebih dari 2 objek dalam *frame* dengan konfigurasi terbaik menggunakan nilai *minimum neighbour* 5 dengan akurasi rata-rata 94,8%. Sistem bekerja optimal saat menggunakan nilai *scale factor* 1,1 dengan akurasi 76,1%. Sistem memiliki rata-rata waktu komputasi 0,62 detik. Pengenalan objek dapat menghasilkan akurasi yang optimal saat hanya ada 1 objek dalam *frame* dengan pose menghadap depan dan berjarak 4 meter. Konfigurasi terbaik untuk realisasi sistem adalah dengan menggunakan nilai *minimum neighbour* 5 dan nilai *scale factor* 1,1.

Metode *Haar-Cascade Classifier* sudah sering digunakan untuk mendeteksi objek wajah dalam foto. Berdasarkan hasil penelitian ini dapat disimpulkan bahwa metode *Haar-Cascade Classifier* dapat diimplementasikan untuk sistem *realtime* karena memiliki nilai akurasi yang cukup baik dan waktu komputasi yang cukup singkat.

## DAFTAR PUSTAKA

- [1] M. H. B. Pratama, A. Hidayatno, and A. A. Zahra, "Aplikasi Deteksi Gerak Pada Kamera Keamanan Menggunakan Metode Background Subtraction Dengan Algoritma Gaussian Mixture Model," *Transient J. Ilm. Tek. Elektro*, vol. 6, no. 2, pp. 246–253, 2017.
- [2] H. Mulyawan, "Identifikasi dan Tracking Objek Berbasis Image Processing secara Real Time," *EEPIS Final Proj.*, 2011.
- [3] D. S. dan Hurriyatul Fitriyah dan Issa Arwani, "Sistem Penghitung Jumlah Orang Melewati Pintu Menggunakan Metode Background Subtraction Berbasis Raspberry Pi," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 2, pp. 2105–2113, 2018.
- [4] T. A. A. H. Kusuma, K. Usman, S. Saidah, "PEOPLE COUNTING FOR PUBLIC TRANSPORTATIONS USING YOU ONLY LOOK ONCE METHOD," *J. Tek. Inform.*, vol. 2, no. 1, pp. 57–66, 2021.
- [5] C. H. Setjo, B. Achmad, and others, "Thermal image human detection using Haar-cascade classifier," in *2017 7th International Annual Engineering Seminar (InAES)*, 2017, pp. 1–6.
- [6] Jonathan Sachs, "Digital Image Basics," 1996.
- [7] D. I. S. Saputra, M. A. Triwibowo, M. F. Noeris, and M. Alasad, "Pengolahan Citra Negatif Klise Menjadi Citra True Color Dengan Matlab," *Sisfotenika*, vol. 7, no. 1, pp. 85–95, 2017.
- [8] M. Ketcham and V. Inmoonnoy, "The message notification for patients care system using hand gestures recognition," in *2017 International Conference on Digital Arts, Media and Technology (ICDAMT)*, 2017, pp. 412–416.
- [9] F. Umam and others, "Implementasi Sistem Pendeteksian Target Berdasarkan Upper Body Dan Warna Pada Robot Pengikut Manusia," *J. Mikrotek*, vol. 1, no. 1, pp. 11–16, 2018.
- [10] W. Sulistiyo, B. Suyanto, I. Hestningsih, and others, "Rancang Bangun Prototipe Aplikasi Pengenalan Wajah untuk Sistem Absensi Alternatif dengan Metode Haar Like Feature dan Eigenface," *JTET (Jurnal Tek. Elektro Ter.)*, vol. 3, no. 2, 2016.
- [11] R. Lumaris and E. Setyati, "DETEKSI DAN REPRESENTASI FITUR MATA PADA SEBUAH CITRA WAJAH MENGGUNAKAN HAAR CASCADE DAN CHAIN CODE," *IDeaTech 2016*, 2016.
- [12] A. Zhandos and J. Guo, "An approach based on decision tree for analysis of behavior with combined cycle power plant," in *2017 International Conference on Progress in Informatics and Computing (PIC)*, 2017, pp. 415–419.
- [13] E. Y. Puspaningrum and W. S. J. Saputra, "DETEKSI WAJAH DENGAN BOOSTED CASCADE CLASSIFIER," *SCAN-Jurnal Teknol. Inf. dan Komun.*, vol. 13, no. 3, pp. 15–18, 2018.
- [14] A. Obukhov, "Haar classifiers for object detection with cuda," in *GPU Computing Gems Emerald Edition*, Elsevier, 2011, pp. 517–544.
- [15] F. Timbane, S. Du, and R. Aylward, "The Experimental Comparison of Features for Hand Detection," in *2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)*, 2018, pp. 1–7.
- [16] N. D. Miranda, L. Novamizanti, and S. Rizal, "Convolutional Neural Network Pada Klasifikasi Sidik Jari Menggunakan Resnet-50," *J. Tek. Inform.*, vol. 1, no. 2, pp. 61–68, 2020.

- [17] S. Guennouni, A. Ahaitouf, and A. Mansouri, "A comparative study of multiple object detection using Haar-like feature selection and local binary patterns in several platforms," *Model. Simul. Eng.*, vol. 2015, 2015.
- [18] A. Dutta, "Object Detection and Facial Features Identification in Python using OpenCV," 2019.