

## SISTEM *LOW-COST* UNTUK MENGHITUNG JUMLAH PENGUNJUNG PADA RUANGAN MENGGUNAKAN METODE *BACKGROUND SUBTRACTION*

Mochamad Daffa Ramadhana<sup>1</sup>, Radimas Putra Muhammad<sup>2</sup>, F. Yudi Limpraptono<sup>3</sup>

Teknik Elektro S1, ITN Malang, Malang Indonesia

<sup>1</sup>dphormdhno@gmail.com, <sup>2</sup>radimas@lecturer.itn.ac.id, <sup>3</sup>fyudil@lecturer.itn.ac.id

**Abstrak**— Pusat keramaian, seperti pertokoan, perpustakaan, dan mall, sering dikunjungi oleh orang-orang. Mengumpulkan data tentang jumlah pengunjung sangat penting karena dapat menunjukkan minat masyarakat terhadap lokasi dan mengoptimalkan ketersediaan sumber daya untuk memenuhi kebutuhan operasional. Akibatnya, penelitian ini mengembangkan sebuah sistem yang dapat menghitung jumlah pengunjung yang melewati pintu untuk membantu pengelola memantau kondisi tempat. Image processing atau computer vision adalah teknologi yang digunakan dalam pembuatan sistem ini. Hasil pengujian menunjukkan bahwa metode Background Subtraction, yang digunakan dalam komputer vision ini, dapat dengan akurat mendeteksi objek yang bergerak dalam gambar. Selain itu, teknik opening dan closing dapat dengan efektif meningkatkan kualitas gambar.

Sistem *Low-Cost* sendiri adalah solusi yang dirancang untuk mencapai tujuan tertentu dengan meminimalkan biaya tanpa mengorbankan fungsi dan efektivitas secara signifikan. Sistem ini melibatkan penggunaan komponen yang lebih murah, perangkat lunak *open-source*, dan inovasi dalam desain serta implementasi untuk menciptakan sistem yang terjangkau namun tetap efektif dan terjamin penggunaannya. Sistem *low-cost* biasanya menggunakan komponen dengan harga terjangkau yang tetap memiliki kemampuan dasar yang diperlukan untuk menjalankan fungsi utama, seperti halnya : Raspberry Pi, Arduino, dsb. Dan untuk perangkat lunak biasanya menggunakan *open-source* yang gratis dan dapat disesuaikan, karena memungkinkan pengguna untuk memodifikasi perangkat lunak sesuai kebutuhan tanpa perlu membayar biaya tambahan.

Desain yang sederhana juga mengurangi biaya produksi tetapi juga memudahkan proses instalasi dan pemeliharaan. Penggunaan dari sistem *low-cost* ini juga berfokus pada fungsi dasar yang dibutuhkan sehingga biaya produksi bisa ditekan seminimal mungkin sementara sistem tetap dapat menjalankan tugas yang diperlukan.

Pada proyek ini terdapat 5 pengujian dari penghitungan pengunjung yang masuk maupun keluar ruangan, berikut rata-rata nilai presisi dari pengujian sebesar 75,7%, rata-rata nilai recall sebesar 100%, dan rata-rata nilai akurasi sebesar 75,8%.

**Kata Kunci:** *Pengolahan Citra, Computer Vison, People Counter, Background Subtraction, Sistem Low-Cost.*

### I. PENDAHULUAN

Perkembangan teknologi sampai saat ini telah mempengaruhi banyak bidang, dimana salah satu perkembangannya adalah dimana sebuah sistem dapat melihat serta mengenali sebuah objek pada sebuah gambar atau video layaknya sebuah pengelihatan yang ada pada manusia. Tentunya apabila kemampuan sistem tersebut dapat diimplementasikan akan sangat membantu berbagai pekerjaan manusia agar lebih efektif dan efisien[1].

Saat ini, video digital digunakan secara luas untuk berbagai aplikasi. Ini telah menjadi salah satu teknologi terpenting dalam komunikasi multimedia dan digunakan dalam berbagai bidang, termasuk pendidikan, kesehatan, hiburan, bisnis, dan lain-lain. Penggunaan video tidak terbatas pada kebutuhan komunikasi, tetapi juga digunakan dalam berbagai bidang [1].

Salah satu sistem yang diperlukan untuk mengembangkan teknologi yang dapat digunakan dalam berbagai bidang kebutuhan adalah pendeteksian objek manusia. Secara umum, cara kerja deteksi objek manusia ini mengadaptasi cara indra penglihatan manusia mengamati apa yang ada di dunia nyata, mulai dari analisis gambar dan video untuk mengumpulkan informasi tentang aktivitas yang biasa dilakukan manusia [2]. Dengan kata lain, pendeteksian objek manusia mengadaptasi cara pengelihatan manusia (*human vision*) yang saat ini dikenal sebagai teknologi penglihatan komputer [2].

Sistem penghitung pengunjung membantu manajemen mengoptimalkan ketersediaan sumber daya dan memenuhi kebutuhan operasional di kemudian hari. Jika pengelola dapat memberikan pelayanan yang baik kepada pengunjung dan pelanggan dan memenuhi kebutuhan

mereka, efek positif juga akan dirasakan, seperti kepercayaan (trust), kesetiaan (loyalty), keyakinan, marketing mulut (mouth marketing), profit, dan pertumbuhan bisnis (growth) [2].

Teknologi multimedia dapat membantu menghitung jumlah pengunjung ruangan. Teknologi multimedia adalah kombinasi gambar, suara, animasi, dan video yang disimpan, diproses, dan disajikan di dalam komputer. Ini memungkinkan interaksi yang inovatif antara komputer dan manusia. Contohnya, jumlah piksel gambar video yang direkam oleh kamera digital, juga dikenal sebagai webcam, di suatu ruangan dapat dihitung dengan menghitung berapa banyak orang yang masuk ke ruangan tersebut [3].

Sistem Low-Cost sendiri adalah solusi yang dirancang untuk mencapai tujuan tertentu dengan meminimalkan biaya tanpa mengorbankan fungsi dan efektivitas secara signifikan. Sistem ini melibatkan penggunaan komponen yang lebih murah, perangkat lunak open-source, dan inovasi dalam desain serta implementasi untuk menciptakan sistem yang terjangkau namun tetap efektif dan terjamin penggunaannya[3].

#### *B. Rumusan Masalah*

Menurut penjelasan latar belakang sebelumnya bahwa:

1. Bagaimana membangun Prototype Sistem Low-Cost untuk Menghitung Jumlah Pengunjung pada Ruangan menggunakan Metode Background Subtract?
2. Bagaimana sistem kerja dari Background Subtraction untuk menghitung jumlah pengunjung pada ruangan?
3. Bagaimana cara pihak pengelola tempat untuk mengoptimalkan ketersediaan sumberdaya yang dapat menunjang kebutuhan operasional di kemudian hari?

#### *C. Tujuan*

Dari penjelasan rumusan masalah sebelumnya, maka dapat dibuat tujuan dari penelitian ini adalah sebagai berikut:

1. Mengembangkan solusi yang efektif dan ekonomis untuk memantau jumlah pengunjung pada ruangan, yang dapat diimplementasikan dengan biaya rendah dan tanpa memerlukan peralatan yang mahal.
2. Mendorong inovasi dalam bidang penghitungan jumlah pengunjung dengan memanfaatkan teknologi yang tersedia secara murah dan mudah diakses, sehingga dapat membuka potensi penggunaan sistem ini di berbagai industri dan sektor.
3. Meningkatkan efisiensi pengelolaan ruangan dengan menyediakan sistem yang dapat secara otomatis mengoptimalkan ketersediaan sumberdaya yang dapat menunjang kebutuhan operasional, dan mengurangi kebutuhan akan pengawasan manual yang intensif.

## II. KAJIAN PUSTAKA

### *A. Sistem Penghitungan Pengunjung pada Pusat Keramaian*

Pusat keramaian yaitu lokasi yang menjadi pusat aktivitas sosial, komersial, dan budaya di suatu wilayah. Tempat ini sering kali ramai dikunjungi oleh masyarakat untuk berbagai keperluan seperti belanja, rekreasi, hiburan, atau pertemuan sosial. Pusat keramaian memiliki peran penting dalam kehidupan kota atau wilayah tersebut, karena menjadi tempat berkumpulnya berbagai kalangan masyarakat dan menjadi pusat aktivitas ekonomi, sosial, dan budaya.

Sistem penghitungan dalam pusat keramaian merupakan sebuah sistem yang digunakan untuk menghitung jumlah pengunjung yang masuk dan keluar dari area atau ruangan tertentu di dalam pusat keramaian tersebut. Sistem ini dapat menggunakan berbagai teknologi dan metode untuk mengidentifikasi, melacak, dan menghitung orang yang masuk dan keluar, sehingga memberikan informasi tentang lalu lintas pengunjung di pusat keramaian tersebut.

Teknologi yang akan diterapkan pada sistem ini yaitu teknologi kamera yang dipasang di pintu masuk atau pintu keluar. Kamera ini akan merekam gambar atau video dari area tersebut, dan kemudian perangkat lunak khusus akan digunakan untuk menganalisis gambar atau video tersebut untuk mendeteksi dan menghitung jumlah orang yang masuk dan keluar.

Dengan menggunakan sistem penghitungan yang tepat, pengelola pusat keramaian dapat memantau dan mengelola lalu lintas pengunjung secara efektif, meningkatkan pengalaman pengunjung, serta membuat keputusan yang lebih baik terkait operasional dan strategi pemasaran.

### *B. Sistem Low-Cost*

Sistem Low-Cost adalah solusi yang dibuat untuk mencapai tujuan tertentu dengan biaya yang lebih rendah, tanpa mengurangi fungsi dasar dan tujuan sistem. Sistem ini menggunakan komponen yang lebih murah, dan perangkat lunak open-source.

Biasanya, sistem ini memakai komponen dengan harga yang terjangkau seperti Raspberry Pi atau Arduino. Perangkat lunak yang gratis sering dipilih karena bisa disesuaikan dengan kebutuhan tanpa biaya tambahan. Desainnya yang sederhana membantu mengurangi biaya pembuatan dan juga memudahkan pemasangan serta perawatan. Intinya, sistem low-cost fokus pada fungsi-fungsi utama, sehingga biaya bisa ditekan serendah mungkin, tapi tetap bisa bekerja dengan baik.

### C. Computer Vision

Computer vision adalah bidang ilmu yang mempelajari bagaimana merekonstruksi, menafsirkan, dan memahami representasi tiga dimensi gambar dua dimensi berdasarkan karakteristik strukturnya. Computer vision pada dasarnya berusaha untuk meniru dan memodelkan kemampuan penglihatan manusia dengan menggunakan perangkat lunak dan perangkat keras komputer. Studi ini menggabungkan ilmu dari berbagai bidang, seperti teknik elektro, ilmu komputer, matematika, fisiologi, biologi, dan ilmu kognitif. Memahami dan mensimulasikan proses penglihatan manusia membutuhkan keterlibatan berbagai disiplin ilmu tersebut. Untuk mendekati kemampuan manusia dalam menangkap informasi, teknologi computer vision harus memiliki berbagai fungsi pendukung.

Untuk mendekati kemampuan manusia dalam menangkap informasi, sebuah teknologi computer vision harus memiliki berbagai fungsi pendukung yang dapat beroperasi secara menyeluruh.



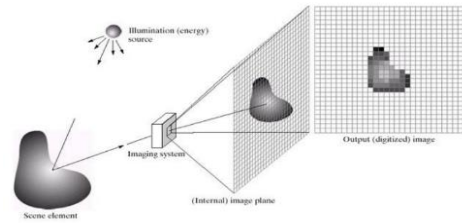
Gambar 1. Computer Vision

### D. Citra Digital

Citra digital adalah representasi data dua dimensi yang dapat dijelaskan dengan fungsi  $f(x, y)$ , di mana  $x$  dan  $y$  menunjukkan koordinat spasial dan intensitas cahaya pada setiap koordinat  $(x, y)$  dalam fungsi  $f$  menunjukkan amplitudo, yang menggambarkan warna cahaya pada citra analog. Karena citra digital telah mengalami proses digitalisasi kuantisasi dan spasial, baik koordinat  $(x, y)$  maupun intensitas  $f(x, y)$  terbatas pada kuantitas diskrit.

Perangkat yang menggunakan metode pengambilan sampel dan kuantisasi untuk mengambil gambar menciptakan citra digital. Jumlah kotak yang teratur dalam baris dan kolom, yang sebenarnya menentukan ukuran piksel pada gambar, ditentukan oleh proses sampling dan kuantisasi. Sementara itu, kuantisasi menentukan jumlah tingkat kecerahan yang dinyatakan dalam skala keabuan, yang sesuai dengan jumlah bit biner yang digunakan oleh perangkat, dan dengan demikian menentukan jumlah warna yang dapat diwakili pada gambar. Pada dasarnya, gambar digital dapat digambarkan sebagai matriks  $M \times N$ .

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



Berikut penjelasan variabel dari rumus di atas :

$f(x, y)$  : Nilai dari fungsi di posisi  $(x, y)$  dalam matriks atau grid.

$M$  : Jumlah baris dalam matriks.

$N$  : Jumlah kolom dalam matriks.

Piksel sendiri terdiri dari dua elemen: koordinat dan intensitas atau warna. Nilai  $f(x, y)$  pada koordinat  $(x, y)$  menunjukkan besar intensitas atau warna piksel pada titik tersebut. Karena itu, sebuah matriks dapat digunakan untuk menampilkan citra digital. Secara terstruktur, matriks dapat digambarkan sebagai fungsi intensitas  $f(x, y)$ , di mana nilai baris dan kolom menunjukkan koordinat posisi, dan nilai  $f(x, y)$  menunjukkan nilai intensitas pada setiap titik  $(x, y)$ , yang menunjukkan besar intensitas citra, tingkat keabuan, atau warna piksel pada titik tersebut. Selama proses digitalisasi (sampling dan kuantisasi), ukuran baris  $M$  dan kolom  $N$  diperoleh, yang menghasilkan matriks  $M \times N$  dan jumlah tingkat keabuan piksel  $G$ .

### E. True Color (RGB)

Gambar RGB, juga dikenal sebagai "true color", disimpan dalam format yang berukuran  $(m \times n) \times 3$ , di mana setiap piksel memiliki tiga warna berbeda: merah, hijau, dan biru. Campuran warna merah, hijau, dan biru ini menentukan warna setiap piksel, sehingga gambar RGB dapat menghasilkan 256 tingkat warna. Teori bahwa merah (630nm), hijau (530nm), dan biru (450nm) adalah panjang gelombang cahaya yang paling peka di mata manusia didasarkan pada RGB.



Gambar 2. Warna RGB

### F. Grayscale

Gambar berwarna gelap adalah gambar dengan satu saluran warna, sehingga seluruh gambar hanya terdiri dari warna abu-abu. Warna-warna ini berkisar dari hitam, yang menunjukkan intensitas terendah, hingga putih, yang menunjukkan intensitas tertinggi. Untuk mengubah gambar berwarna RGB menjadi gambar grayscale, rumus "luma" atau "luminance" digunakan untuk menghasilkan warna abu-abu. Gambar grayscale hanya memerlukan satu nilai kecerahan untuk setiap piksel, berbeda dengan gambar berwarna, di mana tiga

nilai kecerahan diperlukan untuk setiap piksel. Gambar grayscale disimpan dalam format 8-bit dengan 256 tingkat kecerahan yang berbeda, mulai dari 0 (hitam) hingga 255 (putih), dengan nilai yang berbeda untuk warna abu-abu. Rumus berikut digunakan untuk mengubah gambar menjadi grayscale.

$$y' = 0.299R' + 0.587G' + 0.114B'$$

Berikut penjelasan variabel dari rumus di atas :

$y'$  : Nilai luminance atau kecerahan dari piksel setelah dikonversi menjadi grayscale.

$R'$  : Nilai intensitas warna merah dari piksel dalam gambar RGB.

$G'$  : Nilai intensitas warna hijau dari piksel dalam gambar RGB.

$B'$  : Nilai intensitas warna biru dari piksel dalam gambar RGB.



Gambar 4. Grayscale level

#### G. Binerisasi

Binerisasi adalah proses mengubah gambar grayscale menjadi gambar biner, yang berarti setiap piksel pada gambar diubah dari nilai 0 dan 255 menjadi nilai 0 dan 1. Akibatnya, gambar hanya terdiri dari warna hitam dan putih. Selama proses ini, nilai ambang, juga dikenal sebagai ambang, digunakan untuk menentukan piksel mana yang akan menjadi 0 atau 1. Piksel hitam yang memiliki kecerahan di bawah ambang akan diubah menjadi 0, sedangkan piksel putih yang memiliki kecerahan yang sama atau lebih tinggi dari ambang akan diubah menjadi 1. Aplikasi yang membutuhkan pemisahan objek dari latar belakang sering menggunakan citra biner. Ini termasuk aplikasi seperti pengenalan karakter tulisan tangan, analisis bentuk, atau segmentasi gambar. Ciri biner sangat sederhana dan mudah diolah komputer karena hanya ada dua nilai potensial untuk setiap piksel. Proses thresholding berikut digunakan untuk mengubah gambar grayscale menjadi gambar biner:

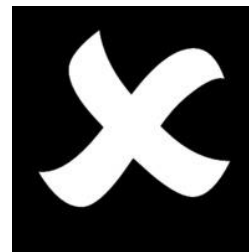
$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{if } f(x,y) < T \end{cases}$$

Berikut penjelasan variabel dari rumus di atas :

$g(x,y)$  : Nilai Biner

$f(x,y)$  : Nilai citra grayscale pada suatu pixel

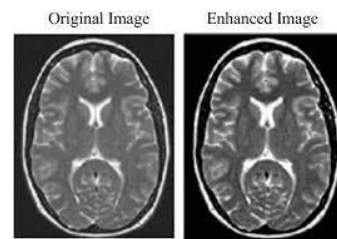
$T$  : Nilai ambang Threshold



Gambar 5. Citra biner

#### H. Digital Image Processing

Pengolahan gambar, atau juga disebut image processing, adalah suatu proses di mana gambar diubah atau diproses menggunakan metode tertentu sehingga menghasilkan gambar lain sebagai output. Pengolahan gambar digital adalah bidang ilmu yang mempelajari banyak hal tentang peningkatan kualitas gambar (seperti meningkatkan kontras, transformasi warna, dan restorasi gambar), transformasi gambar (seperti rotasi, translasi, dan skala), dan pemilihan fitur gambar terbaik untuk analisis, ekstraksi informasi atau deskripsi objek, dan pengenalan objek dalam gambar. Selain itu, data dikompresi atau dikurangi untuk mempercepat penyimpanan, transmisi, dan proses data. Gambar adalah input pengolahan gambar, dan gambar hasil pengolahan adalah outputnya.



Gambar 6. Contoh pengolahan citra digital

#### I. Threshold

Thresholding adalah teknik yang menetapkan nilai tertentu sebagai batas, di mana nilai-nilai di atas atau di bawah batas tersebut dipertahankan, sedangkan nilai-nilai lainnya diabaikan. Hasil dari proses thresholding adalah citra yang menunjukkan perbedaan intensitas yang jelas, biasanya dalam bentuk hitam dan putih. Metode ini umumnya digunakan untuk memisahkan objek dari latar belakang dalam suatu gambar, yang disebut segmentasi. Untuk menghitung jumlah piksel putih dalam setiap wilayah, citra dikonversi menjadi citra biner dengan menggunakan threshold. Proses thresholding mengubah gambar berwarna atau grayscale menjadi citra biner, memungkinkan identifikasi yang jelas antara objek dan latar belakang. Citra hasil thresholding sering digunakan untuk pengenalan objek dan ekstraksi fitur lebih lanjut. Hasil dari proses thresholding biasanya disimpan dalam tipe data float, yang memiliki nilai antara 0 hingga 1. Dengan parameter yang telah ditetapkan sebelumnya, piksel gambar akan diubah menjadi 1 atau putih jika berada di atas batas, dan 0 atau hitam jika berada di bawah batas.

$$f(x,y) = \begin{cases} 1, & f(x,y) > T \\ 0, & f(x,y) \leq T \end{cases}$$

Gambar 7. Nilai Ambang Thresold

Berikut penjelasan variabel dari rumus di atas :

$f(x,y)$  : Nilai dari fungsi di posisi  $(x,y)$  dalam matriks atau grid.

$T$  : Nilai Thresold

#### J. Background Subtraction

Dalam pemrosesan citra menggunakan metode background subtraction sering digunakan di berbagai proyek. Contohnya adalah untuk menghitung jumlah orang yang masuk atau keluar dari suatu area atau kendaraan yang melewati jalur tertentu dalam sistem informasi lalu lintas. Metode ini memungkinkan untuk memisahkan objek yang bergerak, seperti manusia atau kendaraan, dari latar belakangnya dengan cara membedakan antara latar belakang (background) dan objek yang bergerak (foreground).

Ketika hanya latar belakang dan objek yang bergerak yang diamati, proses background subtraction menjadi lebih mudah. Namun, jika ada objek lain yang bergerak dari satu tempat ke tempat lain, seperti bayangan dari objek tersebut, objek tersebut juga akan dianggap sebagai objek bergerak atau bagian dari foreground. Aplikasi berbasis pengolahan citra sering menggunakan algoritma background subtraction, yang ditunjukkan dalam persamaan berikut:

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

Keterangan :

$S(n)$  : Selisih frame ke-n

$F(n)$  : Frame ke-n

$B$  : Citra Background

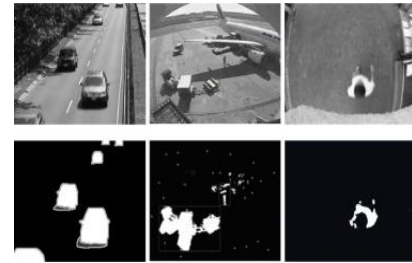
$T$  : Threshold

Metode background subtraction yang digunakan dalam penelitian ini untuk membedakan antara objek (foreground) dan latar belakang. Sebagai bagian dari proses segmentasi, gambar diubah menjadi skala keabuan (grayscale) dan diterapkan nilai ambang yang telah ditentukan. Daftar program dan hasilnya ditunjukkan di bawah ini :

```
cap = cv2.VideoCapture('Video1.mp4')
subtraction = cv2.createBackgroundSubtractorMOG2()

while True:
    ret, frame1 = cap.read()
    time = float(1/delay)
    sleep(time)
```

(a)



(b)

Gambar 6. (a) Program Background Subtraction, (b) Gambar sebelum dan sesudah proses Background Subtraction

#### K. Preprocessing

Preprocessing adalah langkah awal sebelum melakukan deteksi yang bertujuan untuk menyiapkan komponen-komponen yang diperlukan. Proses ini terdiri dari tiga bagian. Ini termasuk mengubah gambar RGB (merah, hijau, dan biru) menjadi gambar *grayscale*, mengidentifikasi wilayah yang relevan untuk fokus deteksi, dan membuat garis hitung untuk menghitung jumlah pengunjung yang masuk dan melewati garis tersebut.

##### a. Grayscale

Grayscale merupakan Sebuah representasi kanal piksel  $(x, y)$  yang menunjukkan derajat kegelapan dan kecerahan dikenal sebagai skala gelap. Memiliki 256 tingkat intensitas, skala abu-abu berkisar dari 0 hingga 255, di mana nilai 0 menunjukkan warna hitam dan nilai 255 menunjukkan warna putih. Persamaan berikut digunakan untuk mengubah gambar berwarna menjadi grayscale:

$$X = (0,299 * R) + (0,587 * G) + 0,114 * B$$

Keterangan :

$X$  : Nilai grayscale

$R$  : Nilai intensitas red

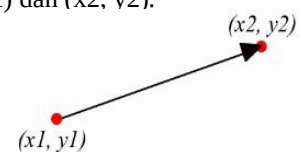
$G$  : Nilai intensitas green

$B$  : Nilai intensitas blue

Grayscale adalah langkah pertama dalam proses pra-pemrosesan yang bertujuan untuk mengonversi gambar berwarna menjadi gambar keabuan, memfasilitasi pemisahan antara objek (foreground) dan latar belakang (background).

##### b. Garis Bantu

Garis ini adalah kode khusus yang dibuat untuk membantu library OpenCV menghitung jumlah pengunjung yang masuk dan keluar. Ini menetapkan koordinat  $(x1, y1)$  dan  $(x2, y2)$ .



Gambar 7. Garis Bantu

Garis = (img, start\_point, end\_point, color, thickness)

Keterangan :

Img : Gambar/video untuk ditarik garis nantinya

Start point : Koordinat awal garis, oleh (x1, y1)

End\_point : Koordinat akhir garis, oleh (x2, y2)

Color : Warna direpresentasikan oleh RGB  
(255,0,0)

Thickness : Ketebalan garis

Langkah pertama dalam proses membuat garis bantu adalah mengetahui resolusi output kamera yang digunakan untuk menentukan titik awal (start\_point) dan titik akhir (end\_point). Setelah itu harus menetapkan warna dan ketebalan garis.

#### L. Region of Interest (ROI)

Region of Interest (ROI) merupakan segmen citra yang dipilih untuk pengolahan lebih lanjut. Dalam sistem penghitung jumlah pengunjung yang masuk dalam ruangan, penempatan ROI diletakkan di atas akses pintu masuk pada ruangan.

Ini memungkinkan sistem untuk menghitung dan mendeteksi pengunjung yang melewati area ROI tersebut tanpa perlu memisahkan area ROI dari frame aslinya. Dengan demikian, proses penghitungan pengunjung dapat dilakukan dengan lebih efisien dan tepat.

#### M. Tracking Object

Tracking Object dapat dilakukan dengan memilih Region of Interest (ROI). Namun, karena algoritma perhitungan menggunakan prinsip pemotongan garis, melacak objek dengan ROI sebagai indikator dapat menyebabkan berbagai kemungkinan yang dapat mengurangi tingkat akurasi sistem. Oleh karena itu, fungsi pelacakan penelitian ini menggunakan tanda titik sebagai indikator. Tanda titik ditempatkan di tengah kotak atau persegi yang telah digambar sebelumnya data yang diperlukan untuk menggambar tanda titik adalah koordinat nilai titik tengah (x, y) pada kotak atau persegi. Rumus nilai

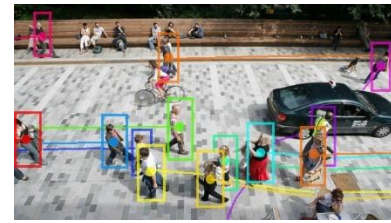
$$\text{tengah koordinat sumbu x: } X = \frac{X}{2}$$

$$\text{rumus nilai tengah koordinat sumbu y: } Y = \frac{Y}{2}$$

Program yang dapat menampilkan tracking berupa tanda titik pada objek ditunjukkan dalam daftar berikut bersama dengan hasilnya:

```
def pega_centro(x, y, w, h):  
    x1 = int(w / 2)  
    y1 = int(h / 2)  
    cx = x + x1  
    cy = y + y1  
    return cx, cy
```

(a)



(b)

Gambar 8. (a) Program Tracking Object, (b) Gambar Hasil Tracking Object

#### N. Algoritma Resize

Resize adalah proses untuk mengubah ukuran gambar digital, baik menjadi lebih kecil atau lebih besar dari ukuran aslinya. Caranya adalah dengan membandingkan ukuran gambar yang ada dengan ukuran gambar yang diinginkan, tanpa menggunakan metode khusus. Berikut urutan dari proses Resize :

1. Menentukan Ukuran Baru: Memutuskan ukuran akhir gambar, baik lebar dan tinggi, yang ingin dicapai.
2. Memilih Algoritma Interpolasi: Memilih metode untuk menentukan nilai piksel baru, yang bisa termasuk:
  - Nearest Neighbor: Menggunakan piksel terdekat.
  - Bilinear Interpolation: Menggunakan rata-rata dari empat piksel terdekat.
  - Bicubic Interpolation: Menggunakan rata-rata dari 16 piksel terdekat.
  - Lanczos Resampling: Menggunakan fungsi sinc untuk akurasi tinggi.
3. Menghitung Nilai Piksel Baru: Menghitung nilai warna atau intensitas untuk setiap piksel baru berdasarkan gambar asli.
4. Menyusun Gambar Baru: Menyusun piksel-piksel baru untuk membentuk gambar dengan ukuran yang diinginkan.

Di bawah ini adalah rumus yang digunakan untuk proses resize :

1. Menghitung koordinat garis :

$$X = \frac{pb \times pp}{pa}$$

Keterangan :

x : Nilai piksel baris baru

pb : Ukuran panjang matriks baru

pp : Posisi piksel baris

pa : Ukuran panjang matriks lama

2. Menghitung nilai koordinat kolom :

$$y = \frac{lb \times pp}{la}$$

Keterangan :

y : Nilai piksel kolom baru

lb : Ukuran lebar matriks baru



lp : Posisi piksel kolom  
la : Ukuran lebar matriks lama

#### O. Open CV

OpenCV (Open Source Computer Vision) adalah sebuah perpustakaan penglihatan komputer yang pertama kali diinisiasi oleh Intel pada tahun 1999. Dengan fokus pada pemrosesan gambar secara real-time, perpustakaan ini menyediakan sejumlah algoritma penglihatan komputer yang dapat dimodifikasi dan dikembangkan secara bebas, dan dimaksudkan untuk berfungsi di berbagai platform. OpenCV dapat digunakan dalam berbagai bahasa pemrograman, seperti C++, Python, dan Android. Salah satu keunggulan utama OpenCV adalah ketersediaannya secara bebas dan lisensi sumber terbuka, yang berarti siapa pun dapat mengunduh, menggunakan, memodifikasi, dan mendistribusikan perangkat lunaknya tanpa biaya atau batasan lisensi. Hal ini menyebabkan peneliti, pengembangan perangkat lunak, dan penggemar teknologi di seluruh dunia sangat menyukainya.



Gambar 9. Open CV

#### P. Python

Python adalah bahasa pemrograman yang sangat serbaguna, mudah dipelajari, dan sintaksisnya bersih dan mudah dibaca. Python, yang dibuat oleh Guido van Rossum pada tahun 1991, sekarang menjadi salah satu bahasa pemrograman paling populer di dunia. Python sangat fleksibel, yang merupakan keunggulan utamanya. Analisis data, kecerdasan buatan, pemrograman jaringan, pengembangan perangkat lunak, dan pengembangan web adalah beberapa industri di mana Python digunakan. Python memiliki banyak pustaka dan kerangka kerja yang tersedia dan dapat digunakan untuk berbagai tujuan.

Python terkenal karena kemampuan untuk membuat kode yang mudah dibaca dan dipahami. Sangat cocok untuk pengembangan perangkat lunak tim karena sintaksis dan struktur bahasanya yang bersih. Python juga lintas platform, yang berarti kode yang ditulis dalamnya dapat berjalan di berbagai sistem operasi seperti Windows, Linux, dan macOS tanpa mengalami banyak perubahan.



Gambar 10. Logo Python

#### Q. Raspberry Pi

Raspberry Pi, juga dikenal sebagai RasPi, merupakan sebuah Komputer Papan Tunggal (Single Board Computer) yang berukuran sebesar kartu kredit. Yayasan Raspberry Pi di Inggris mengembangkan perangkat ini untuk membantu pendidikan komputer dasar di sekolah. Meskipun tidak memiliki hard disk, Raspberry Pi menggunakan kartu SD untuk proses booting dan penyimpanan data jangka panjang. Raspberry Pi menggunakan sistem pada chip (SoC) Broadcom BCM2835 hingga BCM2837, serta prosesor ARM1176JZF-S MHz atau bahkan 1.2GHz 64-bit quad-core ARMv8 CPU pada Raspberry Pi 3.

- Pin 3.3V dan 5V : Ini terhubung ke pin VCC pada komponen yang bersangkutan untuk memberikan tegangan kepada sensor, LED, motor, dan relay.
- Pin GND (Ground) : Ini terhubung ke kaki tanah atau ground.
- Pin GPIO : Raspberry Pi 1 memiliki 26 pin GPIO, sementara Raspberry Pi 2 dan 3 memiliki 40 pin GPIO.



(a)

(b)

Gambar 11. (a) Raspberry Pi, (b) Pin GPIO

#### R. Webcam

Kamera web adalah kamera real-time yang memungkinkan gambar diakses atau ditampilkan secara langsung melalui World Wide Web, aplikasi pesan instan, atau program panggilan video. Istilah "web" kadang-kadang digantikan dengan kata yang menjelaskan konteks atau gambar yang direkam oleh kamera. Kamera web memungkinkan transfer media secara langsung dan memungkinkan komunikasi cepat melalui obrolan video dan pertemuan tatap muka secara langsung.

Webcam biasanya dilengkapi dengan perangkat lunak yang mengambil gambar dari kamera digital secara teratur atau pada interval tertentu dan menyiarkan gambar tersebut melalui koneksi internet. Meskipun ada banyak cara untuk menyiarkan, perangkat keras yang mengubah foto menjadi file JPG dan mengirimkannya ke server web menggunakan Protokol Transfer Berkas (FTP) adalah yang paling umum.

Tingkat bingkai, juga dikenal sebagai frame rate, mengacu pada jumlah gambar yang dapat diambil dan dikirim oleh perangkat lunak dalam satu detik. Untuk streaming video yang lancar, diperlukan setidaknya 15

frame per detik (FPS), dengan 30 FPS yang ideal. Untuk mencapai frame rate yang tinggi, diperlukan koneksi internet yang cepat.

Tidak semua kamera web membutuhkan koneksi langsung ke komputer, beberapa jenis dilengkapi dengan perangkat lunak kamera web dan server web bawaan, yang dikenal sebagai kamera jaringan. Selain itu, kamera web juga dapat menggunakan koneksi radio, Ethernet, atau WiFi untuk menghindari kabel.



Gambar 12. (a) Webcam, (b) Cara kerja webcam

III. METODOLOGI PENELITIAN

A. Waktu dan Tempat Pelaksanaan

Penelitian ini dilakukan di Laboratorium Komputer & Jaringan Teknik Elektro untuk perakitan alat dan pengambilan datanya.

B. Alat dan Bahan

Untuk alat dan bahan yang diperlukan dalam perakitan alat pada penelitian ini terdiri dari beberapa komponen yaitu:

Tabel 1. Kebutuhan Alat dan Bahan

| Nama Perangkat | Jenis         | Jumlah |
|----------------|---------------|--------|
| Raspberry Pi 3 | Model B+      | 1      |
| Webcam         | Xiaovv        | 1      |
| Micro SD       | SanDisk Ultra | 1      |

Untuk detail spesifikasi alat dan bahan yang dipakai pada penelitian ini bisa dilihat sebagai berikut:

1. Raspberry Pi 3 B+

Tabel 2. Spesifikasi Raspberry Pi 3 Model B+

| Spesifikasi            | Keterangan                                      |
|------------------------|-------------------------------------------------|
| SoC : Broadcom         | BCM2837B0<br>Cortex-A53 (ARMv8) 64-bit @ 1.4GHz |
| RAM : 1 GB             | LPDDR2 SDRAM                                    |
| WLAN : 2.4GHz and 5GHz | IEEE 802.11.b/g/n/ac                            |
| Bluetooth              | Bluetooth 4.2, BLE                              |
| Ethernet               | Gigabit Ethernet over USB 2.0 (maximum          |

|                |                             |
|----------------|-----------------------------|
|                | throughput 300 Mbps)        |
| GPIO           | 40 Pin                      |
| Display Output | HDMI                        |
| USB            | 4 USB 2.0                   |
| Interfaces     | CSI, DSI, 3.5 mm audio jack |
| Storage        | Micro SD                    |
| Input/Output   | 5V / 2,5A DC                |

2. Webcam

Tabel 3. Spesifikasi Webcam Xiaovv

| Spesifikasi         | Keterangan                                               |
|---------------------|----------------------------------------------------------|
| Input Voltage       | 5 V                                                      |
| Input Ampere        | 1 A                                                      |
| Output Voltage      | 1.25 – 30V (DC)                                          |
| Ouput Ampere        | 5 – 10A                                                  |
| Working Temperature | -10 °C~ +50 °C                                           |
| Resolution          | 1080P                                                    |
| Video Format        | H.265 _AVI                                               |
| Connection          | USB 2.0                                                  |
| Pixel               | 200W                                                     |
| Lens                | F2.4 4MM                                                 |
| Support Platforms   | Windows 7/8/10<br>Linux 2.4.6 above<br>Mac OS 10.5 above |

3. Micro SD

Tabel 4. Spesifikasi Micro SD SanDisk (Ultra)

| Spesifikasi       | Keterangan                                                  |
|-------------------|-------------------------------------------------------------|
| Capacity          | 32 GB                                                       |
| Speed Class       | Class 10                                                    |
| Speed Performance | 120 MB/s                                                    |
| Compatibility     | Micro SDHC & Micro SDXC (smart phone, tablets, camera, etc) |
| Colour            | Red and grey design for Ultras Series                       |

C. Perancangan Penelitian

Penelitian ini membahas tentang implementasi teknologi Computer Vision dalam menghitung jumlah pengunjung di sebuah ruangan dengan menerapkan metode Background Subtraction. Menggunakan perangkat seperti webcam dan Raspberry Pi 3 Model B+, penelitian ini bertujuan untuk merancang dan menerapkan sistem penghitung pengunjung dengan fokus pada aspek keamanan. Sistem ini diharapkan dapat membantu dalam berbagai hal seperti otomatisasi, keamanan, kesehatan, dan efisiensi energi.

Dirancang untuk operasi real-time, sistem ini ditempatkan di atas pintu dengan posisi kamera webcam yang diatur sedemikian rupa untuk mengambil gambar



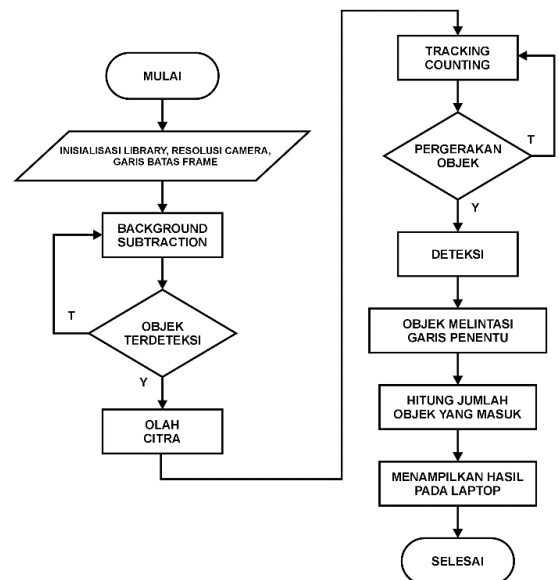
pengunjung dengan akurat dan jelas saat melewati. Prosesnya melibatkan pengolahan citra menggunakan metode Background Subtraction untuk mendeteksi objek bergerak, kemudian menghitung jumlah pengunjung berdasarkan objek yang melewati dua garis virtual yang menandakan masuk atau keluar dari ruangan.

#### D. Spesifikasi Sistem

Spesifikasi sistem yang dibuat pada penelitian ini adalah sebagai berikut :

- a. Untuk merencanakan dan mengimplementasikan sistem penghitung jumlah pengunjung dengan memperhatikan keamanan dengan mendeteksi jumlah pengunjung di suatu ruangan sehingga dapat membantu mencegah situasi berbahaya yang melibatkan kerumunan atau kepadatan massa.
- b. Mengontrol kepatuhan terhadap aturan dan regulasi kesehatan dan keselamatan dengan memantau jumlah orang di suatu ruangan hal ini dapat membantu untuk pengendalian penyebaran penyakit.
- c. Memantau jumlah pengunjung atau peserta dalam acara atau pertunjukan untuk tujuan keamanan dan perencanaan
- d. Mengoptimalkan penggunaan energi dengan menyesuaikan pencahayaan, pemanasan, atau pendinginan berdasarkan jumlah pengunjung dalam suatu ruangan.
- e. Meningkatkan respons terhadap situasi darurat dengan memberikan informasi real-time tentang jumlah pengunjung pada suatu ruangan

#### E. Flowchart Keseluruhan Sistem



Gambar 13. Flowchart Keseluruhan Sistem

Untuk penjelasan algoritma flowchart sistem pada gambar 13. adalah program ini dimulai dengan menginisialisasi library, resolusi kamera, dan garis virtual frame. Webcam kemudian merekam objek yang bergerak, sementara di sisi lain frame akan muncul gambar hasil background subtraction. Selanjutnya, program akan memproses gambar tersebut untuk memeriksa apakah telah menyerupai gambar aslinya; jika belum, program akan kembali ke proses background subtraction.

Dikarenakan beberapa gambar objek yang bergerak yang terekam oleh webcam memiliki beberapa noise, langkah berikutnya dari program adalah melakukan tracking dan counting untuk secara konsisten mengidentifikasi dan melacak posisi, kecepatan, dan arah pergerakan objek. Selain itu, program juga akan menghitung jumlah objek yang berada dalam jangkauan kamera. Setelah itu, objek yang bergerak akan dideteksi saat melewati garis virtual frame, dan program akan mulai menghitungnya. Hasilnya kemudian akan ditampilkan jumlah pengunjung ruangan pada laptop.

#### F. Flowchart Cara Kerja Garis Virtual Sistem

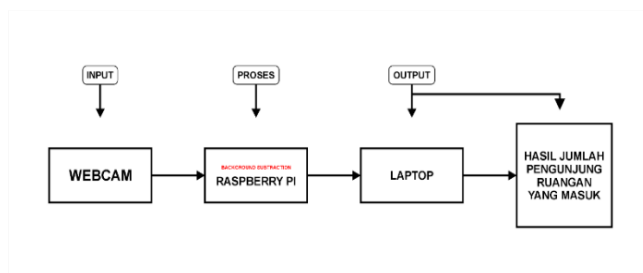
Untuk penjelasan flowchart cara kerja dari garis virtual pada gambar 14 yaitu garis virtual adalah garis yang mana berfungsi sebagai pembatas pergerakan objek yang akan masuk di wilayah penghitungan, dalam projek ini ada 2 garis virtual merah dan biru yang mana garis merah ini akan menjadi garis pembatas objek yang akan memasuki ruangan, dan garis biru sebagai pembatas yang mana objek akan keluar dari ruangan.



Gambar 14. Flowchart cara kerja garis virtual

Cara kerjanya hanya menunggu objek melewati garis tersebut, objek yang akan masuk supaya terhitung maka harus melewati garis merah dan sebelumnya berjalan dari bagian bawah frame yang tercapture kamera lalu berjalan ke arah atas bagian frame, begitupun objek yang akan keluar supaya terhitung maka harus melewati garis biru dan sebelumnya berjalan dari bagian atas frame lalu berjalan menuju ke bagian bawah frame.

#### G. Blok Diagram Alat

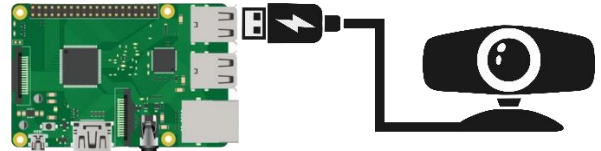


Gambar 15. Blok Diagram Alat

Pada awal proyek ini, dirancang alat untuk menghitung jumlah orang yang melewati pintu. Alat ini terdiri dari laptop, Raspberry Pi, dan webcam, yang berfungsi sebagai input. Raspberry Pi memproses gambar yang diambil oleh webcam dan menampilkannya pada

laptop. Gambar 3.3 menunjukkan bahwa rangkaian webcam dan Raspberry Pi terhubung melalui kabel USB. Ini memungkinkan sistem untuk menerima gambar video real-time, yang dapat diolah kemudian. Laptop dapat menampilkan jumlah orang yang masuk dan keluar ruangan.

#### D. Perancangan Perangkat Keras



Gambar 16. Perancangan Perangkat Keras

Gambar 16 menunjukkan perangkat keras yang dirancang untuk menghitung jumlah pengunjung yang datang ke suatu ruangan menggunakan metode Subtraction Background. Alat yang digunakan termasuk webcam dan Raspberry Pi 3 b+, yang dihubungkan menggunakan kabel USB untuk memungkinkan sistem menerima data video real-time. sehingga dapat diproses lebih lanjut. Selanjutnya, laptop menampilkan informasi tentang jumlah orang yang masuk dan keluar dari suatu ruangan.

#### E. Perancangan Sistem Penghitung Pengunjung Ruangan

Pada sistem penghitungan pengunjung pada penelitian ini menggunakan metode *background subtraction* kemudian untuk memperbaiki frame yang *noise* diperlukan tahap *tracking* dan *counting* untuk secara konsisten mengidentifikasi dan melacak posisi, kecepatan, dan arah pergerakan objek yang sudah di program menggunakan Python dan sudah terhubung dengan Raspberry pi 3 b+ yang dijalankan di OS Raspbian. Sistem penghitungan pengunjung ini akan menampilkan jumlah pengunjung yang masuk dan keluar ruangan pada tampilan laptop.



Gambar 17. Perancangan Perangkat Keras

#### F. Kapasitas kamera Low-Cost dalam FPS (Frame Per Second)

Webcam Xiaovv dalam proyek ini mampu menangani 30 frame per detik (FPS) pada resolusi 1080p. Artinya, webcam ini dapat menangkap video

dengan cukup lancar untuk aplikasi seperti video call, pengenalan wajah, dan pemantauan langsung. FPS mengukur seberapa banyak gambar ditampilkan dalam satu detik, semakin tinggi angkanya, semakin halus video yang ditampilkan. Hasil FPS juga tergantung pada kualitas perangkat keras dan perangkat lunak yang digunakan. Untuk menghitung FPS yaitu hasil frame dibagi dengan durasi dari video, seperti berikut :

$$\text{Frame} = \frac{\text{Total Frame}}{\text{Duration (s)}}$$

## V. HASIL DAN ANALISIS

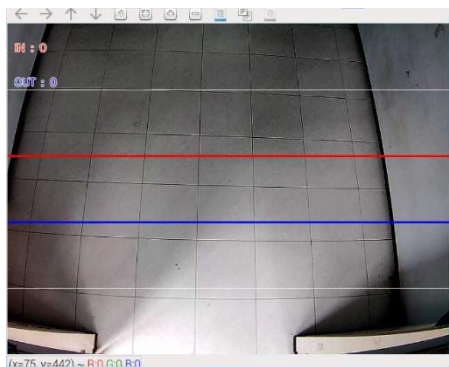
### A. Inisialisasi Webcam

Untuk menerapkan projek menghitung jumlah pengunjung dalam ruangan, alat yang perlu disiapkan adalah webcam, Raspberry Pi 3 model B+, SD Card yang sudah terpasang di Raspberry Pi dan sudah menginstal Raspi OS untuk mengoperasikan sistemnya.

Langkah berikutnya saya menghubungkan webcam dengan Raspberry Pi nya. Setelah kedua perangkat terhubung saya memerlukan kode program dengan Bahasa Python untuk mengatur webcam agar bisa mengambil gambar dan video pengunjung yang tertangkap dalam jangkauan. Setelah kode program siap harus dipastikan webcam bekerja dengan baik dengan melakukan pengujian dari kode program tersebut.

### B. Background Modelling

Pada tahap *background modelling* dalam pengolahan citra pada projek ini sangatlah penting yang bertujuan untuk memisahkan objek dari latar belakang dalam sebuah gambar atau video. Sistem ini akan menetapkan salah satu frame dari sebuah latar belakang sebagai pemisah antara objek manusia yang akan berjalan sehingga mempermudah dalam proses penghitungan jumlah pengunjung dalam ruangan.



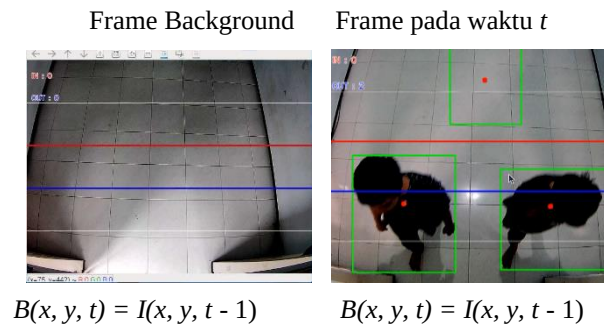
Gambar 18. Background Modelling

### C. Foreground Detection

Pada tahapan Foreground Detection ini, sistem melakukan penyimpanan setiap frame gambar ke dalam variabel yang terdiri dari tiga elemen, yaitu x, y, dan t.

Elemen x merepresentasikan sebagai ketinggian gambar, elemen y merepresentasikan sebagai lebar gambar,

dan elemen t merepresentasikan sebagai waktu dari frame gambar berikutnya.



Gambar 19. Perbedaan frame yang akan dibandingkan

Background dalam konteks ini didefinisikan sebagai frame dari gambar sebelumnya. Sistem akan membandingkan gambar atau frame berikutnya dengan background untuk menentukan perbedaan gambar dari sebelum dan sesudahnya. Perbedaan ini akan dihitung dengan cara membandingkan nilai piksel pada frame masukan gambar yang terekam dengan nilai piksel pada background model.

$$B(x, y, t) = I(x, y, t - 1)$$

Berikut penjelasan variabel dari rumus di atas :

B : Background

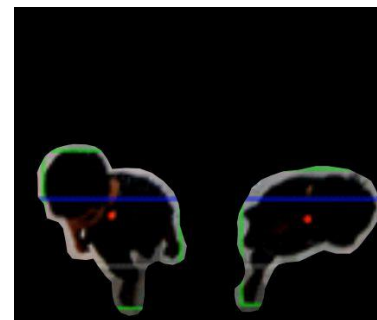
x : Koordinat horizontal piksel di dalam gambar atau video.

y : Koordinat vertikal piksel di dalam gambar atau video.

T : Menunjukkan kapan gambar atau frame tersebut direkam dalam urutan waktu.

I : Image atau gambar yang merujuk pada citra grayscale atau citra berwarna.

Th : Nilai ambang Thresold.



Gambar 20. Output Foreground Detection

Nilai piksel yang lebih dari nilai ambang, maka akan dikenali sebagai latar depan (objek bergerak) dan yang lebih kecil dari nilai ambang maka akan dikenali sebagai latar belakang (background).

### D. Citra RGB

Langkah subtraksi latar belakang dalam metode background subtraction dengan citra RGB melibatkan perbandingan gambar saat ini dengan gambar latar belakang untuk menemukan objek yang bergerak. Gambar latar belakang ini bisa dibuat dengan menghitung rata-rata

dari beberapa frame video atau menggunakan metode statistik yang lebih canggih.

#### E. Grayscale

Model latar belakang adalah representasi dari latar belakang tanpa objek yang bergerak. Dalam citra grayscale, ini hanya melibatkan satu saluran intensitas. Kecerahan dalam gambar grayscale disimpan dalam format 8-bit, yang memberikan 256 kemungkinan tingkat kecerahan, mulai dari 0 (hitam) hingga 255 (putih), dengan nilai di antaranya adalah berbagai tingkat warna abu-abu.

#### F. Binerisasi

Dalam proses ini, kita menggunakan nilai ambang untuk menentukan apakah piksel akan menjadi 0 atau 1. Piksel dengan kecerahan di bawah nilai ambang akan menjadi 0 (hitam), sedangkan piksel dengan kecerahan sama atau lebih tinggi dari nilai ambang akan menjadi 1 (putih). Citra biner sering digunakan untuk memisahkan objek dari latar belakang, misalnya dalam pengenalan tulisan tangan, analisis bentuk, atau segmentasi gambar. Karena citra biner hanya memiliki dua nilai (hitam dan putih), yang sangat sederhana dan mudah diproses oleh komputer.

#### G. Thresholding

Selanjutnya dalam tahapan ini diperlukan penetapan nilai Threshold (T). Nilai Threshold ini merupakan nilai ambang batas yang digunakan untuk membedakan antara piksel foreground (objek) dan piksel background (latar belakang) dalam gambar.

Penetapan nilai piksel ini hanya yang memiliki nilai lebih besar dari atau sama dengan Threshold yang akan diproses lebih lanjut. Sedangkan piksel-piksel yang memiliki nilai lebih kecil dari Threshold akan diabaikan.

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{if } f(x,y) < T \end{cases}$$

Berikut penjelasan variabel dari rumus di atas :

$g(x,y)$  : Nilai dari fungsi di posisi  $(x,y)$  dalam matriks atau grid.

T : Nilai Threshold

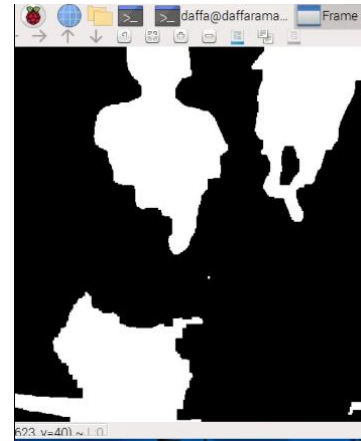
Fungsi  $g(x, y)$  dalam konteks ini mewakili citra biner yang dihasilkan dari citra grayscale  $f(x, y)$ . Citra biner ini hanya terdiri dari dua nilai, yaitu 0 (hitam) untuk piksel background dan 1 (putih) untuk piksel foreground.

Berikut penjelasan variabel dari rumus di atas :

$g(x,y)$  : Nilai dari fungsi di posisi  $(x,y)$  dalam matriks atau grid.

T : Nilai Threshold

Fungsi  $g(x, y)$  dalam konteks ini mewakili citra biner yang dihasilkan dari citra grayscale  $f(x, y)$ . Citra biner ini hanya terdiri dari dua nilai, yaitu 0 (hitam) untuk piksel background dan 1 (putih) untuk piksel foreground.



Gambar 21. Output Thresholding

Nilai Threshold (T) harus ditentukan dengan tepat agar citra biner yang dihasilkan dapat merepresentasikan objek yang diinginkan dengan baik. Besarnya nilai Threshold tergantung pada beberapa faktor, seperti karakteristik gambar, tingkat pencahayaan, dan jenis objek yang ingin dideteksi.

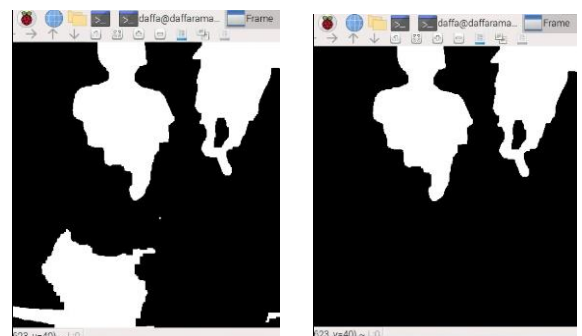
#### E. Operasi Morphology

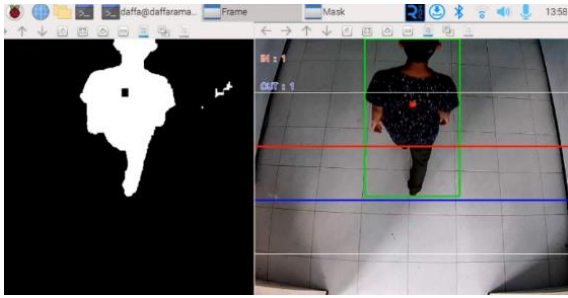
Pada tahap selanjutnya yaitu menggunakan dua operasi morphology, yakni adalah opening dan closing, untuk meningkatkan kualitas segmentasi gambar dalam proses deteksi.

1. Operasi opening akan bekerja sebagai pemutus bagian-bagian kecil dari objek manusia yang hanya terhubung dengan 1 atau 2 titik pada gambar biner yang dihasilkan sebelumnya. Hal ini bertujuan untuk menghilangkan objek-objek kecil yang bukan merupakan bagian dari objek manusia yang sebenarnya, dan memperhalus batas-batas objek tanpa mengubah area objek manusia sehingga menjadi lebih jelas dan rapi.

2. Operasi closing sendiri juga memiliki fungsi yang sangat penting untuk melengkapi sistem kerja dari operasi opening tadi, yaitu bekerja dengan cara menyambungkan bagian-bagian objek manusia yang terputus dan mengisi lubang-lubang kecil pada objek. Bertujuan untuk memperkuat struktur objek manusia dan menghilangkan celah-celah kecil yang mungkin terjadi pada hasil segmentasi.

Sebelum      Sesudah





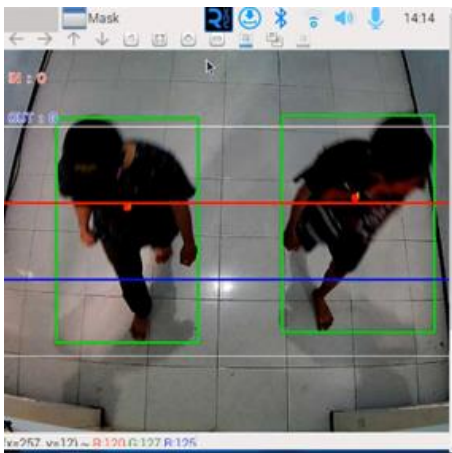
Gambar 22. Output Operasi Morphology (Opening & Closing)

#### F. Annotation

Pada tahap ini, sistem akan melakukan isolasi objek manusia yang terdeteksi dengan cara menggambar persegi panjang di sekeliling objek tersebut atau yang biasa disebut Bounding Box. Bounding Box ini digambarkan dengan menggunakan variabel (x, y, w, h). Berikut penjelasan makna dari masing-masing variabel :

- x : Nilai x mewakili koordinat garis atas persegi panjang.
- y : Nilai y mewakili koordinat garis kiri persegi panjang.
- w : Nilai w mewakili lebar persegi panjang.
- h : Nilai h mewakili tinggi persegi panjang.

Proses isolasi objek manusia ini dilakukan dengan cara mencari kontur (garis tepi) dari objek manusia yang terdeteksi. Dari hasil kontur tersebut, sistem kemudian akan menentukan koordinat (x, y, w, h) untuk menggambarkan persegi panjang yang mengelilingi objek manusia dengan tepat.



Gambar 23. Output Annotation

#### G. Proses Resize

Dalam metode background subtraction, algoritma resize dapat digunakan untuk menyesuaikan ukuran gambar atau video sebelum proses latar belakang dilakukan. Resize sering kali diperlukan jika ukuran citra video atau gambar yang diambil tidak sesuai dengan ukuran yang diinginkan atau jika perlu dilakukan untuk

efisiensi pemrosesan. Caranya adalah dengan membandingkan ukuran gambar yang ada dengan ukuran gambar yang diinginkan.

#### H. Pengukuran Presisi Akurasi Recall

Presisi, akurasi, dan recall adalah metrik yang digunakan dalam evaluasi kinerja sistem klasifikasi atau deteksi, terutama dalam konteks pembelajaran mesin dan visi komputer.

##### 1. Presisi (Precision) :

Presisi adalah ukuran seberapa akurat prediksi positif model. Ini dapat dihitung dengan membagi jumlah prediksi positif yang benar (True Positives) dengan jumlah total prediksi positif (True Positives + False Positives). Dengan kata lain, presisi menunjukkan persentase prediksi positif yang benar dari semua prediksi positif yang dibuat oleh model.

$$\text{Precision} = \frac{\text{True Positif}}{\text{True Positif} + \text{False Positive}}$$

##### 2. Recall :

Recall mengukur seberapa baik model dapat mendeteksi semua kejadian positif sebenarnya. Ini dihitung dengan membagi jumlah prediksi positif yang benar (True Positives) dengan jumlah kejadian positif sebenarnya (True Positives + False Negatives). Recall menunjukkan kemampuan model untuk menemukan semua kasus positif.

$$\text{Recall} = \frac{\text{True Positif}}{\text{True Positif} + \text{False Negative}}$$

##### 3. Akurasi (Accuracy):

Akurasi mengukur seberapa sering model membuat prediksi yang benar secara keseluruhan. Ini dapat dihitung dengan membagi jumlah prediksi yang benar (True Positives + True Negatives) dengan jumlah prediksi total yang dibuat.

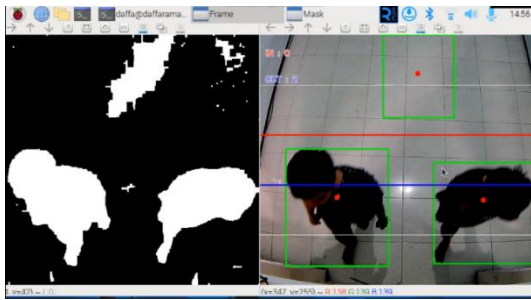
$$\text{Accuracy} = \frac{\text{True Positif} + \text{True Negative}}{\text{Total Prediction}}$$

#### H. Penghitungan Jumlah Pengunjung pada Ruangan

Pada tahap memasuki bagian inti dari penelitian, yaitu menghitung jumlah pengunjung pada ruangan. Penghitungan ini akan dibagi menjadi 5 bagian, yaitu :

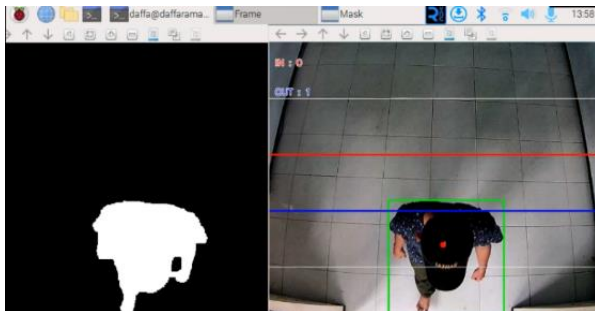
1. Menghitung 1 pengunjung (objek manusia) yang akan masuk ke dalam ruangan dengan cara menempatkan garis virtual (horizontal) berwarna merah pada posisi bagian atas frame yang terekam kamera. Apabila koordinat dari pengunjung tadi melewati garis virtual biru maka akan mulai terhitung sebagai pengunjung yang masuk ke dalam ruangan. Berikut contoh penerapannya :





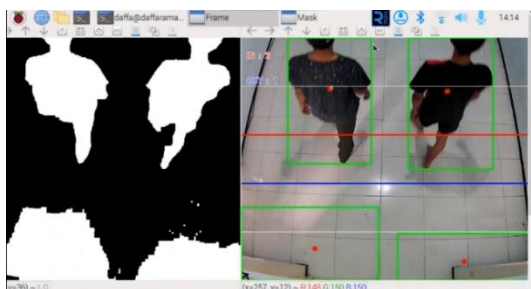
Gambar 24. Menghitung 1 pengunjung yang masuk

- Menghitung 1 pengunjung (objek manusia) yang akan keluar dari dalam ruangan dengan cara menempatkan garis virtual (horizontal) berwarna biru pada posisi bagian bawah frame yang terekam kamera. Apabila koordinat dari pengunjung tadi melewati garis virtual merah maka akan mulai dihitung sebagai pengunjung yang keluar dari dalam ruangan. Berikut contoh penerapannya :



Gambar 25. Menghitung 1 pengunjung yang keluar

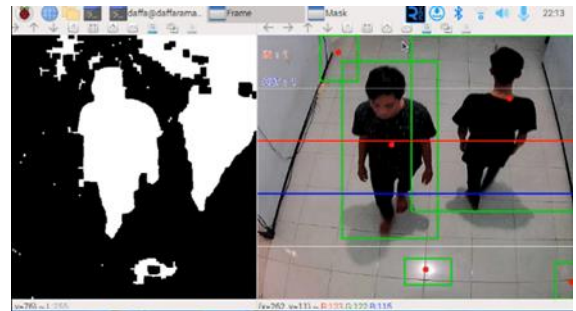
- Menghitung 2 pengunjung (objek manusia) yang akan masuk secara bersamaan ke dalam ruangan dengan cara menempatkan garis virtual (horizontal) berwarna merah pada posisi bagian atas frame yang terekam kamera. Apabila koordinat dari kedua pengunjung tadi melewati garis virtual biru maka akan mulai dihitung sebagai pengunjung yang masuk ke dalam ruangan. Tujuan dari pengujian ini adalah memastikan bahwa program berjalan dengan baik dan benar, dan hasil yang ditampilkan sesuai dengan apa yang diharapkan. Berikut contoh penerapannya :



Gambar 26. Menghitung 2 pengunjung yang masuk secara bersamaan

- Menghitung 2 pengunjung (objek manusia) yang akan keluar secara bersamaan dari dalam ruangan dengan cara menempatkan garis virtual (horizontal) berwarna biru pada posisi bagian bawah frame yang terekam kamera. Apabila koordinat dari kedua pengunjung tadi melewati garis

virtual merah maka akan mulai dihitung sebagai pengunjung yang keluar dari dalam ruangan. Tujuan dari pengujian ini adalah memastikan bahwa program berjalan dengan baik dan benar, dan hasil yang ditampilkan sesuai dengan apa yang diharapkan. Berikut contoh penerapannya :



Gambar 27. Menghitung 2 pengunjung yang keluar secara bersamaan

- Menghitung 2 pengunjung (objek manusia) yang akan masuk dan keluar secara bersamaan dari dalam ruangan dengan cara menempatkan garis virtual (horizontal) berwarna merah dan biru pada posisi bagian atas dan bawah frame yang terekam kamera. Apabila koordinat dari kedua pengunjung tadi melewati garis virtual merah dan biru maka akan mulai dihitung sebagai pengunjung yang keluar dari dalam ruangan dan masuk ke dalam ruangan. Tujuan dari pengujian ini adalah memastikan bahwa program berjalan baik dan benar, dan hasil yang ditampilkan sesuai dengan apa yang diharapkan. Berikut contoh penerapannya :

#### H. Evaluasi *Prototyping*

Setelah melakukan pengujian dari beberapa proses diatas, maka ada evaluasi terhadap hasil rancangan prototype tersebut. Berikut hasil rangkuman dari beberapa tahap dari proyek penghitungan pengunjung dalam ruangan pada tabel di bawah ini :

Tabel 5. Evaluasi Prototyping

| No | Kebutuhan Sistem     | Hasil Rancangan                                                                                        | Keterangan |
|----|----------------------|--------------------------------------------------------------------------------------------------------|------------|
| 1  | Inisialisasi Webcam  | Kamera dapat diakses secara baik dan dapat menampilkan hasil rekaman dengan normal                     | Berhasil   |
| 2  | Background Modelling | Pada tahap ini, frame yang terekam oleh webcam dapat dikenali sebagai latar belakang pengambilan objek | Berhasil   |
| 3  | Foreground Detection | Membandingkan antara frame background model dengan frame yang lain                                     | Berhasil   |
| 4  | Thresholding         | Mengubah bentuk frame asli dari rekaman webcam menjadi frame yang sudah berbentuk subtraction, dengan  | Berhasil   |

|   |                                       |                                                                                                                                                                                      |          |
|---|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|   |                                       | perbedaan latar belakang hitam dan objek yang berwarna putih                                                                                                                         |          |
| 5 | Operasi Morphology                    | Proses penyempurnaan frame thresholding yang kurang baik, yaitu dengan proses opening dan closing                                                                                    | Berhasil |
| 6 | Proses Annotation                     | Mendeteksi pergerakan objek dengan cara mengelilingi objek yang bergerak dengan bounding box warna hijau dan titik merah yang mengikuti pergerakan objek                             | Berhasil |
| 7 | Penghitungan pengunjung dalam ruangan | Pada tahap ini terdapat 5 pengujian dari beberapa pengunjung yang masuk dan keluar, hasilnya cukup akurat namun terdapat sedikit problem yaitu karena sensitifitas terhadap bayangan | Berhasil |

## I. Kode Program

Proyek ini bertujuan untuk membangun sistem penghitung jumlah pengunjung dalam suatu ruangan. Sistem ini menggunakan webcam untuk merekam dan menghitung jumlah orang yang masuk dan keluar ruangan. Bahasa pemrograman yang digunakan adalah Python dan library Open CV digunakan untuk memproses gambar dari kamera, kemudian untuk pengembangan sistem menggunakan Geany. Berikut adalah beberapa contoh kode program untuk proyek ini :

### 1. Source Code untuk untuk Inisialisasi Webcam

```
#Inisialisai Webcam
#cap = cv.VideoCapture(0)
cap = cv.VideoCapture(0)
#camera = PiCamera()
##camera.resolution = (160,120)
##camera.framerate = 5
##rawCapture = PiRGBArray(camera, size=(160,120))
##time.sleep(0.1)
```

Gambar 29. Kode Program untuk inisialisasi webcam

### 2. Source Code untuk proses Background Modelling

```
while(cap.isOpened()):
    ##for image in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
    ##tentukan Background Modelling
    ret, frame = cap.read()
    ## frame = image.array
```

Gambar 30. Kode Program untuk Background Modelling

### 3. Source Code untuk Foreground Detection

```
#Menerapkan Foreground Detection / Background Detection
fgmask = fgbg.apply(frame)
fgmask2 = fgbg.apply(frame)
```

Gambar 32. Kode Program untuk proses Foreground Detection

### 4. Source Code untuk proses Thresholding

```
h = 480
w = 640
frameArea = h*w
areaTH = frameArea/250
print('Area Threshold', areaTH)
```

Gambar 34. Kode Program untuk proses Thresholding

### 5. Source Code untuk operasi Morphology

```
#Matriks yang digunakan menghitung nilai setiap piksel
kernelOp = np.ones((3,3),np.uint8)
kernelOp2 = np.ones((5,5),np.uint8)
kernelCl = np.ones((11,11),np.uint8)
```

Gambar 34. Kode Program untuk menetapkan nilai Morphology

```
#Opening (pengikisan -> penghapusan) untuk menghilangkan noise
mask = cv.morphologyEx(imBin, cv.MORPH_OPEN, kernelOp)
mask2 = cv.morphologyEx(imBin2, cv.MORPH_OPEN, kernelOp)
#Closing (penghapusan -> pengikisan) untuk menggabungkan ke bagian putih
mask = cv.morphologyEx(mask, cv.MORPH_CLOSE, kernelCl)
mask2 = cv.morphologyEx(mask2, cv.MORPH_CLOSE, kernelCl)
```

Gambar 35. Kode Program untuk proses Morphology

### 6. Source Code untuk proses Annotation

```
##Melakukan Tracking Object
M = cv.moments(cnt)
cx = int(M['m10']/M['m00']) #mengambil titik bagian tengah objek
cy = int(M['m01']/M['m00'])
x,y,w,h = cv.boundingRect(cnt)
```

Gambar 36. Kode Program untuk proses Annotation

### 7. Source Code untuk tahap menghitung jumlah pengunjung

```
new = True
if cy in range(up_limit,down_limit):
    for i in persons:
        if abs(x-i.getx()) <= w and abs(y-i.gety()) <= h:
            # el objeto esta cerca de uno que ya se detecto antes
            new = False
            i.updateCoords(x,cy) #actualiza coordenadas en el objeto and resets age
            if i.going_UP(line_down,line_up) == True:
                cnt_up += 1;
                print("ID: ",i.getId(),',crossed going up at',time.strftime("%c"))
                log.write("ID: "+str(i.getId())+', crossed going up at ' + time.strftime("%c") + '\n')
            elif i.going_DOWN(line_down,line_up) == True:
                cnt_down += 1;
                print("ID: ",i.getId(),',crossed going down at',time.strftime("%c"))
                log.write("ID: " + str(i.getId()) + ', crossed going down at ' + time.strftime("%c") + '\n')
            break
    if i.getState() == '1':
        if i.getDir() == 'down' and i.gety() > down_limit:
            i.setDone()
        elif i.getDir() == 'up' and i.gety() < up_limit:
            i.setDone()
    if i.timeOut():
        #sacar 1 de la lista persons
        index = persons.index(i)
        persons.pop(index)
        del i #liberar la memoria de i
    if new == True:
        p = Person.MyPerson(pid,cx,cy, max_p_age)
        persons.append(p)
        pid += 1
```

Gambar 36. Kode Program untuk menghitung jumlah pengunjung

## J. Perhitungan Presisi *Recall* dan Akurasi Pegunjung

1. Menghitung 1 pengunjung (objek manusia) yang akan masuk ke dalam ruangan dengan cara menempatkan garis virtual (horizontal) berwarna merah pada posisi bagian atas frame yang terekam kamera. Berikut perhitungannya :

• Penghitungan Presisi :

$$Precision = \frac{TP}{TP + FP} = \frac{1}{1 + 0} = \frac{1}{1} \times 100\% = 100\%$$

• Penghitungan Recall :

$$Recall = \frac{TP}{TP + FN} = \frac{1}{1 + 0} = \frac{1}{1} \times 100\% = 100\%$$

• Penghitungan Akurasi :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{1 + 0}{1 + 0 + 0 + 0} = \frac{1}{1} \times 100\% = 100\%$$

2. Menghitung 1 pengunjung (objek manusia) yang akan keluar dari dalam ruangan dengan cara menempatkan garis virtual (horizontal) berwarna biru pada posisi bagian bawah frame yang terekam kamera. Berikut perhitungannya :

• Penghitungan Presisi :

$$Precision = \frac{TP}{TP + FP} = \frac{1}{1 + 0} = \frac{1}{1} \times 100\% = 100\%$$

• Penghitungan Recall :

$$Recall = \frac{TP}{TP + FN} = \frac{1}{1 + 0} = \frac{1}{1} \times 100\% = 100\%$$

• Penghitungan Akurasi :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{1 + 0}{1 + 0 + 0 + 0} = \frac{1}{1} \times 100\% = 100\%$$

3. Menghitung 2 pengunjung (objek manusia) yang akan masuk ke dalam ruangan dengan cara menempatkan garis virtual (horizontal) berwarna merah pada posisi bagian atas frame yang terekam kamera. Berikut perhitungannya :

• Penghitungan Presisi :

$$Precision = \frac{TP}{TP + FP} = \frac{2}{2 + 2} = \frac{2}{4} \times 100\% = 50\%$$

• Penghitungan Recall :

$$Recall = \frac{TP}{TP + FN} = \frac{2}{2 + 0} = \frac{2}{2} \times 100\% = 100\%$$

• Penghitungan Akurasi :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{2 + 2}{2 + 3 + 2 + 0} = \frac{5}{7} \times 100\% = 71\%$$

4. Menghitung 2 pengunjung (objek manusia) yang akan keluar dari dalam ruangan dengan cara menempatkan garis virtual (horizontal) berwarna biru pada posisi bagian bawah frame yang terekam kamera. Berikut perhitungannya :

• Penghitungan Presisi :

$$Precision = \frac{TP}{TP + FP} = \frac{2}{2 + 0} = \frac{2}{2} \times 100\% = 100\%$$

• Penghitungan Recall :

$$Recall = \frac{TP}{TP + FN} = \frac{2}{2 + 0} = \frac{2}{2} \times 100\% = 100\%$$

• Penghitungan Akurasi :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{2 + 0}{2 + 2 + 0 + 0} = \frac{2}{4} \times 100\% = 50\%$$

5. Menghitung 2 pengunjung (objek manusia) yang akan masuk dan keluar dari dalam ruangan dengan cara menempatkan garis virtual (horizontal) berwarna merah dan biru pada posisi bagian atas dan bawah frame yang terekam kamera. Berikut perhitungannya :

• Penghitungan Presisi :

$$Precision = \frac{TP}{TP + FP} = \frac{2}{2 + 5} = \frac{2}{7} \times 100\% = 28,5\%$$

• Penghitungan Recall :

$$Recall = \frac{TP}{TP + FN} = \frac{2}{2 + 0} = \frac{2}{2} \times 100\% = 100\%$$

• Penghitungan Akurasi :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{2 + 5}{2 + 5 + 5 + 0} = \frac{7}{12} \times 100\% = 58\%$$

## K. Tabel Jumlah Pengunjung pada setiap Percobaan

| No | Percobaan                                                        | Jumlah Pengunjung |
|----|------------------------------------------------------------------|-------------------|
| 1  | Penghitungan 1 Pengunjung yang masuk                             | 1                 |
| 2  | Penghitungan 1 Pengunjung yang keluar                            | 1                 |
| 3  | Penghitungan 2 Pengunjung yang masuk secara bersamaan            | 2                 |
| 4  | Penghitungan 2 Pengunjung yang keluar secara bersamaan           | 2                 |
| 5  | Penghitungan 2 Pengunjung yang masuk dan keluar secara bersamaan | 2                 |

Tabel 6. Jumlah Pengunjung pada setiap Percobaan

## L. Tabel Tingkat Akurasi Program pada Penghitungan Pengunjung

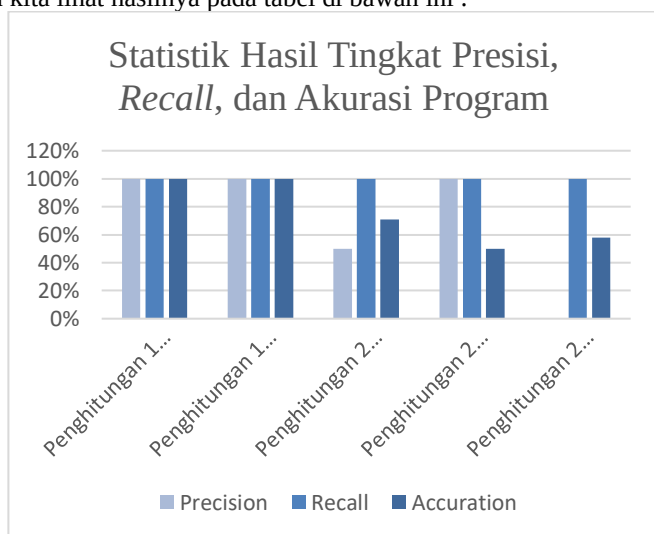
| No | Percobaan                                                        | Tingkat Akurasi Program |        |            |
|----|------------------------------------------------------------------|-------------------------|--------|------------|
|    |                                                                  | Precision               | Recall | Accuration |
| 1  | Penghitungan 1 Pengunjung yang masuk                             | 100%                    | 100%   | 100%       |
| 2  | Penghitungan 1 Pengunjung yang keluar                            | 100%                    | 100%   | 100%       |
| 3  | Penghitungan 2 Pengunjung yang masuk secara bersamaan            | 50%                     | 100%   | 71%        |
| 4  | Penghitungan 2 Pengunjung yang keluar secara bersamaan           | 100%                    | 100%   | 50%        |
| 5  | Penghitungan 2 Pengunjung yang masuk dan keluar secara bersamaan | 28,5%                   | 100%   | 58%        |

Tabel 7. Akurasi Program pada Penghitungan Pengunjung

## M. Tabel Statistik Hasil Tingkat Presisi, Recall, dan Akurasi Program

Pada tahap akhir ini yaitu melihat hasil dari penghitungan dari sistem yang sudah berjalan untuk menguji tingkat presisi, recall, dan akurasi program pada saat penghitungan pengunjung. Untuk di setiap percobaan ini terdapat 1 orang yang masuk, 1 orang yang keluar, 2 orang yang masuk secara bersamaan, 2 orang yang keluar secara bersamaan, dan 2 orang yang masuk dan keluar secara bersamaan. Tujuan dari pengujian tingkat presisi, recall, dan akurasi program pada saat penghitungan pengunjung ini adalah, supaya bisa memastikan seberapa jauh sistem ini berjalan secara maksimal.

Untuk hasil pengujian dari beberapa percobaan tersebut, bisa kita lihat hasilnya pada tabel di bawah ini :



## V KESIMPULAN DAN SARAN

### A. Kesimpulan

Berdasarkan progres yang telah dilakukan, terdapat beberapa kesimpulan terkait pengujian pada proyek ini, yaitu sistem ini menggunakan sistem pemrosesan Raspberry Pi 3 model b+, webcam sebagai sensor untuk mengambil citra dan micro sd sebagai penyimpanan Raspi OS.

Pada pengolahan citra ini terdapat kode program yang menampilkan 2 garis virtual yang bertujuan sebagai garis pembatas objek yang akan masuk atau keluar. Garis tersebut berwarna merah sebagai batas pengunjung yang akan terhitung ketika memasuki ruangan, dan garis warna biru sebagai batas pengunjung yang akan terhitung keluar dari dalam ruangan.

Kemudian ketika mengimplementasikan metode Background Subtraction berhasil mendeteksi pengunjung yang masuk maupun keluar dengan memisahkan antara objek bergerak dengan latar belakangnya. Objek yang bergerak akan dikenali sebagai warna putih dan latar belakang dikenali sebagai warna hitam.

Saat melakukan pengolahan citra terdapat beberapa masalah noise yang merusak kejernihan gambar sehingga membutuhkan suatu teknik Opening & Closing untuk menghilangkan dan mengisi lubang yang ada pada citra. Dalam menentukan masuk dan keluar manusia menggunakan tracking dengan adanya Bounding Box hijau yang mengelilingi dan ada 1 titik merah yang berada pada tengah objek yang terdeteksi. Bounding Box tersebut akan terus mengikuti kemana gerak objek yang terdeteksi.

Masuk ke dalam proses penghitungan pengunjung terdapat beberapa percobaan yang dilakukan, antara lain : menghitung 1 pengunjung yang masuk, menghitung 1 pengunjung yang keluar, menghitung 2 pengunjung yang masuk secara bersamaan, menghitung 2 pengunjung yang keluar secara bersamaan, menghitung 2 pengunjung yang masuk dan keluar secara bersamaan.

Hal ini mendapatkan beberapa hasil yang cukup akurat, berikut hasil dari pengujian penghitungan jumlah pengunjung :

1. Menghitung 1 pengunjung yang masuk, pada tahap ini sistem bekerja dengan sangat akurat dan berjalan sesuai perintah kode program. Hasil dari tingkat presisi sebesar 100%, recall sebesar 100%, dan akurasi sebesar 100% program ini bekerja ketika mendeteksi dan menghitung pengunjung yang masuk ke dalam ruangan, namun perlu digaris bawahi terdapat sedikit problem yang berasal dari gerak yang berlebihan dan bayangan. Kedua hal itu sedikit mempengaruhi kinerja sistem karena sistem yang sangat sensitive mendeteksi perubahan pada setiap frame, maka akan memunculkan bounding box dan titik merah yang mengikuti pergerakan melewati diantara kedua garis virtual tersebut.

2. Menghitung 1 pengunjung yang keluar, pada tahap ini sama dengan poin yang ke-1. Hasil dari tingkat presisi sebesar 100%, recall sebesar 100%, dan akurasi sebesar

100% namun ada beberapa masalah yang sedikit mempengaruhi kinerja dan hasil dari penghitungan pengunjung.

3. Menghitung 2 pengunjung yang masuk secara bersamaan, pada tahap pengujian ini hasil dari tingkat presisi sebesar 50%, recall sebesar 100%, dan akurasi sebesar 71%, hal ini dipengaruhi oleh objek yang berjalan secara bersamaan, sehingga muncul 2 bounding box sekaligus yang mengikuti arah kedua objek itu berjalan melewati garis virtual. Akan tetapi juga terdapat beberapa masalah yang tidak berbeda dari sebelumnya, yaitu gerakan berlebih dari pengunjung dan bayangan dari pengunjung sendiri, atau bisa juga karena cahaya matahari ataupun cahaya lampu yang memantul pada latar belakang sehingga memunculkan objek hitam yang ikut terdeteksi sebagai objek bergerak.

4. Menghitung 2 pengunjung yang keluar secara bersamaan, pada tahap ini hasil dari tingkat presisi sebesar 100%, recall sebesar 100%, dan akurasi sebesar 50% dan terdapat sedikit masalah pada bayangan dan gerakan pengunjung tersebut. Sedikit perbedaan dari pengujian sebelumnya, dalam percobaan 2 pengunjung ataupun lebih sedikit ada permasalahan pada atribut atau barang yang digunakan, seperti halnya topi, sarung, rok, tas dsb, ini juga sedikit mempengaruhi sensitivitas sistem yang bekerja.

5. Menghitung 2 pengunjung yang masuk dan keluar secara bersamaan, pada tahap pengujian ini tingkat ketepatan sistem dalam mendeteksi pergerakan pengunjung lebih sensitive lagi, dikarenakan dari arah berlawanan juga terdapat noise berupa bayangan, gerakan berlebihan dari pengunjung, dan atribut yang dibawa/digunakan. Dalam pengujian ini hasil dari tingkat presisi sebesar 28,5%, recall sebesar 100%, dan akurasi sebesar 58% karena banyak gangguan yang juga ikut terdeteksi, semisal ada 3 orang yang keluar dan masuk namun yang terhitung bisa sampai 7 orang.

#### B. Saran

Saran untuk kedepannya penelitian pengitungan pengunjung menggunakan pengolahan citra ini dapat dikembangkan lagi dari segi kode programnya supaya dapat mengurangi tingkat sensitifitas terhadap gangguan ringan seperti gerakan yang berlebihan, bayangan, maupun atribut yang digunakan, karena sangat berdampak sekali pada ketepatan penghitungan pengunjung. Serta pada bagian kamara sebagai perangkat inputan dari citra yang terekam supaya jelas dalam pengambilan gambar, lebih tepatnya yang beresolusi lebih tinggi dan gambar yang dihasilkan mudah dimengerti oleh kode programnya.

## DAFTAR PUSTAKA

- [1] Rizal, A. M. (2020) 'Implementation of a Ball Tracking System In Image Processing Using the Find Contour Method Implementasi Sistem Tracking Bola Pada Pengolahan Citra Menggunakan Metode Find Contour', *Journal of Electrical and Electronic Engineering UMSIDA*, 4(2), pp. 105–114.
- [2] Andrean, K., Armanto, H., & Pickerling, C. (2020). Sistem Tempat Parkir Terintegrasi yang Dilengkapi dengan Aplikasi Mobile dan Mikrokontroler.
- [3] Davies, E. R. (2022). The dramatically changing face of computer vision. *Advanced Methods and Deep Learning in Computer Vision*, <https://doi.org/10.1016/B978-0-12-822109-9.00010-2>.
- [4] Afriani, D., Ilham, D. N., Khairuman, Talib, M. S., and Harahap, M. K. (2022). Multidisciplinary Sciences and Arts Internet of Things Based Automatic Visitor Counter International Journal of Multidisciplinary Sciences and Arts. *International Journal of Multidisciplinary Sciences and Arts*, 1(1), 35–40. <https://doi.org/10.47709/ijmdsa.v1i1.1615>.
- [5] Garcia-Garcia, B., Bouwmans, T., and Silva, A. J. R. (2020). Background Subtraction in Real Applications: Challenges, Current Models and Future Directions. *Computer Science Review*, 35, 1–42. <https://doi.org/10.1016/j.cosrev.2019.100204>.
- [6] Hossein, S., Sarim Hasan, Z., Rollakanti, R., Arulananth, T. S., Prasad, S. V. S., Anand, J., Ravi, B., and Poliseti, M. (2022). Moving Object Detection using Background Subtraction. *ASSIC 2022 - Proceedings: International Conference on Advancements in Smart, Secure and Intelligent Computing*, August, 1–8. <https://doi.org/10.1007/978-3-319-07386-6>.
- [7] Khan, S. H., Yousaf, M. H., Murtaza, F., and Velastin, S. A. (2020). Passenger Detection and Counting during Getting on and off from Public Transport Systems . *NED University Journal of Research*, 2(March), 35–46. <https://doi.org/10.35453/NEDJR-ASCN-2019-0016>.
- [8] Mahamad, A. K., Saon, S., Hashim, H., Ahmadon, M. A., and Yamaguchi, S. (2020). Cloud-Based People Counter. *Bulletin of Electrical Engineering and Informatics*, 9(1), 284–291. <https://doi.org/10.11591/eei.v9i1.1849>.
- [9] Mukti, F. S., Farokhah, L., and Aqromi, N. L. (2021). Pemodelan Sistem Deteksi Wajah Sebagai Penghitung Jumlah Penumpang Transportasi Publik. *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, 4(1), 67–77. <https://doi.org/10.31598/jurnalresistor.v4i1.834>.
- [10] Murdan, A. P., Bucktowar, V., Oree, V., and Enoch, M. P. (2020). Low Cost Bus Seating Information



- [11] Agustina, F., & Ardiansyah Amri, Z. (2020). Identifikasi Citra Daging Ayam Kampung dan Broiler Menggunakan Metode GLCM dan Klasifikasi NN. INFOKAM.
- [12] Amatullah, I. N. (2021). Perbandingan Tingkat Akurasi Pengenalan Kadar Ikan pada Pempek Berdasarkan Resolusi Kamera dengan Metode Pengenalan Jaringan Syaraf Tiruan Backpropagation. Universitas multi data Palembang.
- [13] Zhao, Y., Zhang, Z., Zhu, H., & Ren, J. (2022). Quantitative Response of GrayLevel Co-Occurrence Matrix Texture Features to the Salinity of Cracked Soda Saline-Alkali Soil. International Journal of Environmental Research and Public Health.
- [14] M. Arsal, B. Agus Wardijono, and D. Anggraini, “Face Recognition Untuk Akses Pegawai Bank Menggunakan DeepLearning Dengan Metode CNN,” J. Nas. Teknol. dan Sist. Inf., vol. 6, no.1, pp. 55–63, 2020, doi:10.25077/teknosi.v6i1.2020.55-63.[5] Y. Achmad, R. C. Wihandika, and C. D.
- [15] K. S. D. S and R. Kamalraj, “Vehicle Detection and Counting of a Vehicle Using Opencv,” no. 05, pp. 1610–1613, 2021.
- [16] A. N. N. Afifah, Indrabayu, A. Suyuti, and Syafaruddin, “A review on image processing techniques for damage detection on photovoltaic panels,” ICIC Express Lett., vol. 15, no. 7, pp. 779–790, 2021, doi: 10.24507/icicel.15.07.779.
- [17] C. K. U. Nggiku, A. Rabi, and S. Subairi, “Deteksi Kantuk Untuk Keamanan Berkendara Berbasis Pengolahan Citra,” J. JEETech, vol. 4, no.1, pp. 48–56, 2023.

## VII. BIODATA PENULIS



Mochamad Daffa Ramadhana, lahir di Tulungagung, 19 November 2001. Penulis menyelesaikan Pendidikan Sekolah Menengah Kejuruan di SMK Queen Al-Falah Kediri. Setelah itu pada tahun 2020 penulis melanjutkan Pendidikan Studi diperguruan tinggi swasta Institut Teknologi Nasional Malang program studi Teknik Elektro S-1 dengan konsentrasi Teknik Komputer. Akhir kata dari penulis mengucapkan terimakasih dan rasa syukur yang sebanyak-banyaknya atas selesainya penelitian ini dengan judul “Sistem *Low-Cost* untuk Menghitung Jumlah Pengunjung pada Ruangan Menggunakan Metode *Background Subtraction*”.