



Sistem Penghitung Jumlah Orang Menggunakan Metode SSD-MobileNet dan Centroid Tracking

People Counting System Using SSD-MobileNet and Centroid Tracking Methods

Afandi Nur Aziz Thohari *¹⁾, Aisyatul Karima²⁾, Angga Wahyu Wibowo³⁾, Kuwat Santoso⁴⁾

^{1,2,3)}Program Studi Teknik Informatika, Jurusan Teknik Elektro, Politeknik Negeri Semarang
Jl. Prof. Soedarto, SH, Semarang, Indonesia 50275

⁴⁾Program Studi Teknologi Rekayasa Komputer, Jurusan Teknik Elektro, Politeknik Negeri Semarang
Jl. Prof. Soedarto, SH, Semarang, Indonesia 50275

Cara sitasi: N. P. Pertama and N. P. Kedua, "Petunjuk penulisan dan kirim artikel Jurnal Teknologi dan Sistem Komputer mulai penerbitan nomor 7(4) tahun 2019," *Jurnal Teknologi dan Sistem Komputer*, vol. x, no. x, pp. xx-xx, 202x. doi: [10.14710/jtsiskom.x.x.202x.xx-xx](https://doi.org/10.14710/jtsiskom.x.x.202x.xx-xx), [Online].

Abstract – One of the applications of artificial intelligence to prevent the spread of coronavirus is to create an automatic people counting system to avoid crowds in the room. This research discusses making a prototype system for counting people using a deep learning algorithm on a single board computer. The purpose of this research is to calculate the number of people in a room, so that the occupancy of the room can be reduced. This research aims to combine two computer vision methods, namely SSD-MobileNet, for object identification of people and centroid tracking to count the number of people. Based on the tests carried out, it shows that the system has been able to count people objects with 100% accuracy if the number of people entering the room is one, two, or three together. Then the system can detect objects with a maximum distance of 10 meters, and the light intensity is dim or less than 100 lux. The computational test shows that the system can process video with a frame count of 30 fps and high definition (HD) video quality.

Keywords – object detection, people counting, object tracking, prototype

Abstrak – Salah satu penerapan kecerdasan buatan untuk mencegah penyebaran virus corona adalah dengan membuat sistem penghitung jumlah orang otomatis untuk mencegah kerumunan di dalam ruangan. Penelitian ini membahas mengenai pembuatan prototipe sistem penghitung jumlah orang menggunakan algoritma deep learning pada single board computer. Tujuan dari penelitian ini adalah untuk menghitung jumlah orang dalam suatu ruangan agar okupansi ruangan dapat ditekan. Kontribusi dari penelitian ini adalah mengkombinasikan dua metode visi komputer yaitu SSD-MobileNet untuk identifikasi objek orang dan centroid tracking untuk menghitung jumlah orang. Berdasarkan pengujian yang telah dilakukan menunjukkan bahwa sistem telah dapat menghitung objek orang dengan akurasi 100% apabila jumlah orang yang memasuki ruangan berjumlah satu, dua, atau tiga secara bersama-sama. Kemudian sistem

dapat mendeteksi objek dengan jarak maksimal 10 meter dan intensitas cahaya redup atau kurang dari 100 lux. Pada pengujian komputasi menunjukkan bahwa sistem dapat memproses video dengan jumlah frame 30 fps dan kualitas video high definition (HD).

Kata kunci – deteksi objek, menghitung jumlah orang, pelacakan objek, prototipe

I. PENDAHULUAN

Perkembangan teknologi informasi dan kecerdasan buatan telah sangat masif digunakan diberbagai bidang. Tujuan dari teknologi tersebut adalah untuk meringankan pekerjaan manusia dan juga untuk otomatisasi pekerjaan. Adanya kecerdasan buatan memungkinkan pekerjaan yang semula dilakukan oleh manusia, sekarang dapat dilakukan oleh mesin. Beberapa pekerjaan seperti penjaga pembayaran tol dan resepsionis, sekarang digantikan oleh mesin pembayaran tol dan email [1].

Teknologi kecerdasan buatan dapat dimanfaatkan untuk bidang kesehatan, terutama pada masa pandemi Covid-19 saat ini. Salah satu pemanfaatan teknologi kecerdasan buatan untuk mencegah penyebaran Covid-19 adalah membangun sistem penghitung jumlah orang otomatis. Sistem ini dibangun sebagai usaha mengurangi kerumunan sesuai dengan himbuan pemerintah mengenai protokol kesehatan. Salah satu bentuk penegakan protokol kesehatan adalah dengan membatasi kerumunan di tempat-tempat publik seperti mall, rumah makan, tempat wisata, dan lain sebagainya.

Pembatasan okupansi ruangan ini telah diatur dalam peraturan pemerintah melalui instruksi Menteri Dalam Negeri Republik Indonesia Nomor 6 Tahun 2020 tentang penegakan protokol kesehatan untuk pengendalian penyebaran Covid-19 [2]. Oleh karena itu, para pemilik mall atau cafe wajib membatasi jumlah pengunjung sebanyak 50 persen dari kapasitas ruangan. Umumnya pemilik tempat usaha tersebut menghitung jumlah pengunjung secara manual untuk mengetahui kapasitas ruangan. Namun melalui teknologi kecerdasan buatan



yaitu visi komputer, jumlah orang yang memasuki ruangan dapat diketahui secara otomatis.

Kajian mengenai implementasi visi komputer untuk menghitung jumlah orang sebenarnya sudah pernah dilakukan. Salah satunya adalah penelitian yang dilakukan oleh [3], menghitung jumlah orang menggunakan metode HOG dan Kalman Filter. Video diambil dari kamera CCTV secara *real-time*. Kemudian penelitian dari [4] dan [5] yang membangun sistem penghitung jumlah orang menggunakan metode *background subtraction*. Terakhir penelitian dari [6] dan [7] yang mendeteksi jumlah orang menggunakan metode pengenalan wajah. Penggunaan metode pengenalan wajah ini bertujuan agar tidak terjadi duplikasi saat melakukan perhitungan jumlah orang.

Pada penelitian [4] dan [5] objek dapat dihitung apabila tertangkap kamera, namun sistem belum dapat membedakan objek yang terdeteksi. Jadi semua objek akan dihitung tidak peduli objek tersebut manusia atau ukan. Kemudian penelitian yang dikerjakan oleh [8] telah dapat membedakan objek orang, tetapi memiliki akurasi perhitungan orang belum akurat apabila jumlah orang yang melintas lebih dari satu. Oleh karena itu, pada penelitian ini akan menyelesaikan kekurangan yang terdapat pada sistem penghitung jumlah orang.

Pada penelitian ini dibangun prototipe sistem yang mampu melakukan klasifikasi objek manusia dan bukan manusia. Prototipe ini juga mampu menghitung jumlah orang yang melintasi kamera secara bersama-sama. Kedua fitur tersebut dapat bekerja secara efektif, apabila kondisi cahaya di sekitar objek terang. Kemudian objek manusia yang terangkap kamera harus utuh sebagian anggota tubuhnya. Jadi meskipun objek manusia berjalan beresama-sama, asalkan terdapat minimal setengah bagian anggota tubuh terangkap kamera, maka sistem akan dapat mengklasifikasi sebagai objek manusia.

Metode yang digunakan untuk mendeteksi objek orang adalah *Single Shot Detector* (SSD) [9]. Sedangkan metode untuk menghitung jumlah orang adalah *centroid tracking* [10]. Fitur yang terdapat pada sistem penghitung jumlah orang ini adalah dapat mengirim notifikasi email apabila jumlah pengunjung telah mencapai batas okupansi ruangan. Dapat merekap kehadiran pengunjung masuk dan keluar kedalam berkas CSV, dapat mengatur penjadwalan dari sistem. Kemudian video dapat diakses secara *real-time* menggunakan kamera pengawas yang terhubung ke jaringan intranet.

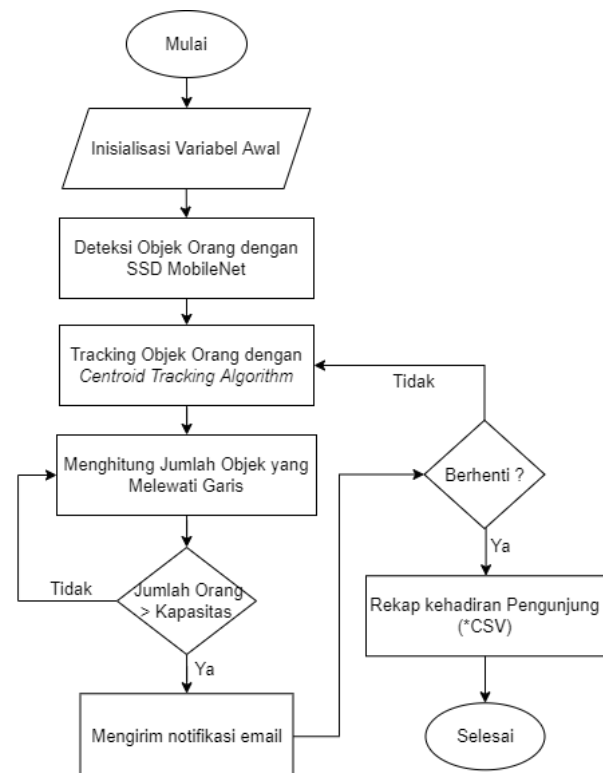
II. Metode penelitian

Penelitian yang dilakukan adalah penelitian terapan berupa prototipe sistem penghitung jumlah orang otomatis. Metode yang dipakai merupakan kombinasi antara deteksi objek dan pelacakan objek. Metode deteksi objek yang dipakai adalah *Single Shot Detector* (SSD) untuk mendeteksi kehadiran orang. Sedangkan metode pelacakan objek yang dipakai pada penelitian ini adalah *Centroid Tracking*.

Pengambilan data dilakukan dengan menggunakan kamera secara *real-time*. Peletakan kamera dilakukan dengan menggunakan teknik *top down view* sehingga

dapat mengambil pemandangan suatu objek dari atas. Data yang diambil adalah data orang yang melewati garis penghitung. Jumlah orang yang digunakan sebagai sample adalah 6, dengan durasi kurang lebih selama 3 jam atau 180 menit.

Mekanisme penghitungan orang dengan cara memberikan ID unik pada setiap orang yang muncul di kamera. Kemudian melakukan pelacakan pada ID tersebut sampai melewati garis. Apabila melewati garis masuk, maka akan dicatat masuk. Sebaliknya apabila melewati garis keluar maka dicatat keluar. Rekap jumlah orang yang masuk dan keluar ruangan akan dicatat pada bekas *Comma Separated Values* (CSV) ketika sistem dimatikan. Sistem juga dapat mengirim notifikasi email apabila jumlah orang yang dicatat melebihi kapasitas ruangan. Adapun alur kerja dari sistem dapat dilihat pada diagram alir Gambar 1.



Gambar 1. Alur Kerja Sistem

A. Persiapan Alat dan Bahan

Proses komputasi dari sistem dilakukan oleh komputer papan tunggal yaitu Raspberry Pi 4 Model B. Kamera yang dipakai untuk mengambil video adalah Rasbpery Pi Camera Versi 2 dengan resolusi 8 Megapiksel. Memori yang dipakai adalah MicroSD dengan kapasitas 128 GB. Alasan menggunakan Raspberry Pi 4 Model B adalah karena ukurannya yang kecil dan memiliki spesifikasi perangkat keras yang tinggi seperti Quad Core CPU dan 4 GB RAM memori [11]. Tampilan dari perangkat keras yang dipakai ditunjukkan pada Gambar 2.



Gambar 2. Tampilan Perangkat Keras Sistem

Selain perangkat keras, terdapat juga perangkat lunak berupa pustaka-pustaka dari bahasa pemrograman python yang perlu dipasang. Pustaka python yang digunakan untuk membangun sistem penghitung orang otomatis ini ditunjukkan pada Tabel 1.

Tabel 1. Pustaka yang dibutuhkan

No	Pustaka	Fungsi
1	OpenCV	Untuk <i>capture</i> video, membuat tulisan dan garis.
2	Dlib	Untuk deteksi pola wajah
3	Imutils	Untuk proses rotasi dan translasi
4	Scipy	Untuk membuat fungsi matematika
5	Numpy	Untuk perhitungan matriks
6	Schedule	Untuk melakukan penjadwalan

B. Object Detection

Pada penelitian ini digunakan kerangka kerja *Single Shot Detector* (SSD) untuk mengidentifikasi objek orang. SSD merupakan kerangka kerja untuk deteksi objek yang hanya mengambil satu tangkapan layar untuk dapat mendeteksi beberapa objek dalam satu gambar [9]. SSD memiliki akurasi yang lebih baik dibandingkan kerangka kerja Faster-RCNN, serta memiliki nilai *Frame Per Second* (FPS) yang lebih besar dibanding kerangka kerja YOLO versi 4 [12].

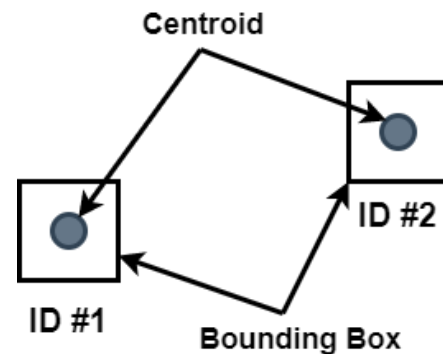
Masalah yang umumnya muncul dalam deteksi objek adalah dari sumber daya yang dipakai. Umumnya untuk melakukan deteksi objek dibutuhkan spesifikasi komputer yang tinggi karena mengolah banyak data gambar [13]. Untuk mengatasi permasalahan tersebut, digunakan arsitektur MobileNet untuk mengurangi komputasi yang berlebih. MobileNet merupakan salah satu arsitektur dari *Convolutional Neural Network* (CNN) yang digunakan untuk deteksi objek pada perangkat dengan sumber daya yang rendah seperti ponsel [14]. Arsitektur MobileNet dipakai, sebab pada penelitian ini menggunakan raspberry pi yang memiliki *resource* yang rendah.

Hal yang membuat MobileNet lebih ringan dibanding arsitektur CNN lainnya adalah pada ketebalan *filter* yang dapat menyesuaikan dengan ketebalan masukan yang berupa gambar. Pada umumnya arsitektur CNN memiliki empat atau lima lapisan konvolusi, namun pada MobileNet lapisan konvolusi dibagi menjadi *depthwise*

dan *pointwise* [15]. Oleh karena itu lapisan dari layer konvolusi MobileNet menjadi lebih sedikit. Pada penelitian ini dilakukan kombinasi dari kerangka kerja SSD dan arsitektur MobileNet untuk mendeteksi objek orang.

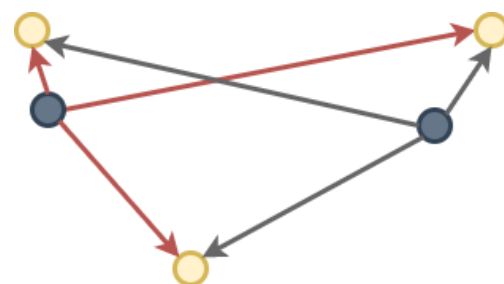
C. Object Tracking

Tujuan dari pelacakan objek adalah untuk dapat menghitung jumlah orang yang memasuki ruangan. Proses dari pelacakan objek dilakukan dengan cara mengidentifikasi koordinat dari objek. Koordinat ini diperoleh dari titik tengah *bouding box* objek. Oleh karena itu teknik pelacakan ini dinamakan *centroid tracking*. Setelah mendapatkan koordinat titik tengah objek, selanjutnya memberikan ID unik untuk setiap objek yang terdeteksi. Terakhir dilakukan pelacakan selama objek bergerak didalam *frame* video. Ketika objek keluar dari frame maka ID yang telah dipakai akan dihapus. Lebih detail mengenai penggunaan *centroid tracking*, ditunjukkan pada Gambar 3.



Gambar 3. Menentukan Titik Tengah dari Objek

Setelah mendapatkan koordinat dari masing-masing titik, langkah selanjutnya adalah menghitung jarak antara objek dengan garis pembatas (portal) menggunakan jarak *euclidean*. Tujuan dilakukannya pengukuran jarak ini untuk mendapatkan nilai *update* dari pergerakan objek. Algoritma *centroid tracking* membuat asumsi bahwa pasangan titik tengah (*centroid*) yang memiliki nilai jarak *euclidean* kecil, maka akan memiliki ID objek yang sama.



Gambar 4. Ilustrasi Perhitungan Jarak *Euclidean* dari masing-masing titik

Seperti ditunjukkan pada Gambar 4 bahwa terdapat dua pasang *centroid* hitam dan kuning yang saling berdekatan, kemudian muncul satu *centroid* baru berwarna kuning. Agar mudah menghitung jarak



euclidean, maka ditambahkan panah yang mengarah ke semua titik. Panah mewakili penghitungan jarak *euclidean* antara semua *centroid* ungu dan semua *centroid* kuning. Jarak *euclidean* dapat menghitung kemiripan suatu data, sehingga dapat digunakan untuk melacak sebuah objek [16]. Persamaan dari jarak *euclidean* ditunjukkan dengan persamaan (1).

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

Menghitung jarak *euclidean* dalam dua dimensi (d), memerlukan koordinat kartesian yang diwakili oleh sumbu x dan y. Koordinat x_1 dan y_1 menunjukkan lokasi awal dari objek. Sedangkan koordinat x_2 dan y_2 menunjukkan lokasi pergerakan objek. Menggunakan persamaan (1) dapat diketahui pergerakan objek, dan juga dapat mendeteksi adanya objek baru.

Apabila objek telah melewati garis pembatas maka, sistem akan menambah jumlah orang yang masuk. Sebaliknya jika objek orang melewati garis pembatas dari arah berlawanan, maka sistem akan menambah jumlah orang yang keluar. Lebih jelas mengenai proses menghitung jumlah orang keluar dan masuk ruangan, dapat dibaca pada Algoritma 1.

Algoritma 1 : Menghitung Jumlah Objek

```
INPUT : Gerakan Objek Manusia
OUTPUT : Jumlah Orang dalam Ruang
1: SET totalDown TO 0
2: SET totalUp TO 0
3: SET totalIn TO 0
4: f.write(str(totalIn))
5: f.close()

6: DEFINE FUNCTION countUp():
7:   global totalUp
8:   totalUp += 1
9:   SET totalIn TO totalDown - totalUp
10:  IF totalIn <= 0:
11:    SET totalIn TO 0

12:  f.write(str(totalIn))
13:  f.close()
14:  SET to.counted TO True

15: DEFINE FUNCTION countDown()
16:  global totalDown
17:  totalDown += 1
18:  SET totalIn TO totalDown - totalUp
19:  IF totalIn <= 0:
20:    SET totalIn TO 0
21:  f.write(str(totalIn))
22:  f.close()
23:  SET to.counted TO True
```

D. Menghitung Jumlah Orang

Cara menghitung jumlah orang yang masuk dan keluar menggunakan ID yang ditangkap oleh kamera. ID akan didaftarkan jika objek yang muncul adalah orang. Selain objek orang, maka tidak akan muncul ID. Hal ini

dikarenakan saat mendefinisikan objek hanya *class* person saja yang dipakai. Kemudian untuk mengetahui orang tersebut masuk atau keluar maka dipakai teknik pelacakan (*tracking*). Apabila orang yang muncul bergerak dari atas ke bawah maka akan dihitung masuk ruangan. Namun jika orang bergerak dari bawah ke atas maka akan dihitung keluar ruangan. ID, arah, dan jumlah orang diinisialisasi pada variabel-variabel yang ada pada program. Proses perhitungan jumlah orang hanya perlu menambahkan kondisi atau percabangan IF dan Else seperti ditunjukkan algoritma 1.

Sedangkan untuk merekam data pada berkas CSV, digunakan fungsi `open()` dengan mode `write` untuk mengisi data. Nilai yang terekam memiliki ambang batas (*threshold*) yang ditentukan oleh pengguna. Apabila telah melewati nilai ambang batas, maka sistem akan mengirimkan notifikasi email yang berisi pesan peringatan. Untuk dapat mengirimkan pesan peringatan ini digunakan pustaka `mailer` yang ada di `python`. Email yang digunakan adalah `google mail`, dimana sebelumnya telah memberikan perizinan agar pearangkat lain dapat mengirim email [17]. Terakhir untuk durasi perekaman sistem dapat diatur menggunakan pustaka `datetime` dan `scheduler` yang ada di `python` [18].

E. Skenario Pengujian

Pengujian adalah proses yang sangat penting untuk mengetahui keberhasilan dari prototipe yang dibangun. Terdapat beberapa skenario pengujian yang dilakukan untuk memastikan prototipe bekerja dengan baik, yaitu pengujian menghitung jumlah orang yang keluar dan masuk ruangan, pengujian kehadiran pengunjung untuk mengetahui rekapitulasi jumlah orang dalam berkas CSV. Pengujian notifikasi email untuk mengetahui keberhasilan pengiriman email dari sistem. Pengujian jarak dan intensitas cahaya untuk mengetahui jarak maksimal yang dapat dibaca prototipe. Terakhir pengujian jumlah *frame* untuk menguji kinerja CPU pada jumlah *frame* dan resolusi yang berbeda-beda.

Data yang dihasilkan dari proses pengujian akan dianalisis untuk mendapatkan akurasi atau prosentasi keberhasilan. Proses analisis dilakukan dengan mengamati hasil pendeteksian dari sistem dengan hasil yang diamati oleh penulis. Sehingga dari proses tersebut dapat diketahui tingkat keberhasilan dari sistem untuk menentukan jumlah orang. Untuk menghitung tingkat keberhasilan atau akurasi sistem, digunakan perhitungan seperti ditunjukkan persamaan (2).

$$p = \frac{n}{m} \times 100\% \quad (2)$$

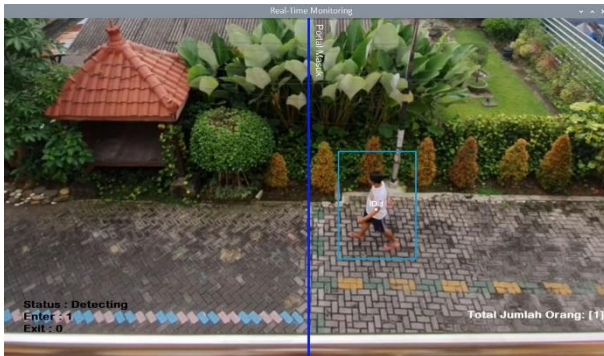
Menghitung persentase keberhasilan (p) dilakukan dengan membandingkan jumlah data percobaan yang berhasil (n) dengan jumlah data pengamatan (m). Berdasarkan hasil perhitungan keberhasilan sistem, dapat diketahui keterbatasan atau spesifikasi sistem yang dapat ditingkatkan pada penelitian berikutnya.



III. HASIL DAN PEMBAHASAN

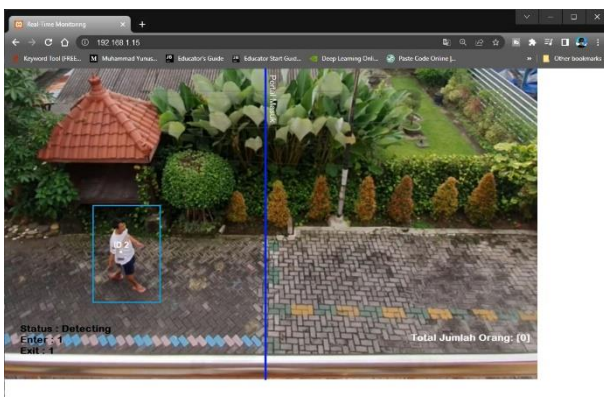
A. Pengujian Perhitungan Jumlah Orang

Perhitungan jumlah orang dilakukan secara otomatis menggunakan prototipe yang dibangun. Pengambilan *frame* dilakukan secara *real-time* dengan menempatkan prototipe berada diatas menggunakan teknik *top down view*. Untuk menghitung jumlah orang yang masuk dan keluar ruangan, dibuat sebuah garis portal. Apabila orang melewati garis portal bagian atas maka secara otomatis sistem akan menghitung bahwa orang tersebut masuk ruangan. Namun jika orang melewati dari bawah garis maka dihitung keluar ruangan. Tampilan video dari program ditunjukkan pada Gambar 5.



Gambar 5. Pengujian Sistem Deteksi Jumlah Orang

Pada sistem pemantauan terdapat keterangan total jumlah orang yang berada di ruangan. Perhitungan total jumlah orang di ruangan didapatkan dari pengurangan jumlah orang yang masuk dan jumlah orang yang keluar. Sistem pemantauan juga dapat diakses secara *real-time* menggunakan alamat ip lokal. Melalui jaringan intranet pemantauan dapat dilakukan melalui PC dekstop maupun ponsel. Adapaun hasil pemantauan dari prototipe ditunjukkan pada Gambar 6.



Gambar 6. Pengujian Monitoring Jumlah Orang Melalui Web Browser

Seperti yang telah ditunjukkan pada Gambar 6 dan 7 bahwa sistem telah dapat menampilkan pemantauan orang yang keluar dan masuk ruangan. Melalui sistem yang dibangun, diperoleh *real-time* video yang dipakai untuk pengujian akurasi perhitungan jumlah orang. Pengujian bertujuan untuk melihat kemampuan sistem dalam menghitung jumlah orang yang masuk/keluar

ruangan. Mekanisme pengujian dilakukan dengan memasuki atau meninggalkan ruangan bersama-sama. Hasil dari pengujian jumlah orang yang masuk ruangan disajikan pada Tabel 2, sedangkan jumlah orang yang keluar ruangan disajikan pada Tabel 3.

Tabel 2. Pengujian Jumlah Orang Memasuki Ruangan

Percobaan	Realisasi Jumlah Orang Masuk	Jumlah Orang Pada Program	Akurasi
1	1	1	100%
2	2	2	100%
3	3	3	100%
4	4	3	80%
5	5	3	60%
6	6	2	30%

Tabel 3. Pengujian Jumlah Orang Keluar Ruangan

Percobaan	Realisasi Jumlah Orang Keluar	Jumlah Orang Pada Program	Akurasi
1	1	1	100%
2	2	2	100%
3	3	3	100%
4	4	4	100%
5	5	3	60%
6	6	2	30%

Berdasarkan data Tabel 2 dan Tabel 3 diketahui bahwa sistem dapat menghitung jumlah orang dengan tepat apabila orang yang melintas berjumlah 1 atau 2 atau 3. Akurasi sistem apabila orang yang melintas secara bersama-sama kurang dari 4 adalah 100%. Sebaliknya jika jumlah orang yang melintas lebih dari 3 maka sistem tidak dapat menghitung jumlahnya secara akurat. Kesalahan sistem dalam menghitung objek yang melintas bersama-sama adalah karena algoritma SSD belum mampu mendeteksi objek yang bagiannya tidak utuh [19]. Saat bergerak bersamaan lebih dari tiga orang, maka akan ada anggota tubuh yang tertutup oleh objek lain sehingga sistem tidak mengklasifikasikan sebagai objek orang.

B. Pengujian Kehadiran Jumlah Orang

Sistem telah berhasil untuk mendeteksi objek dan melakukan *tracking* menggunakan ID. Selanjutnya dilakukan pengujian untuk monitoring jumlah orang dalam area. Hasil monitoring disajikan dalam berkas CSV. Berkas csv akan keluar setelah program selesai dijalankan. Isi dari berkas csv adalah data jumlah orang yang masuk dan keluar dari ruangan. Pada data tersebut juga dapat diketahui tanggal dan waktu saat program berjalan, serta total pengunjung/orang dalam suatu ruangan. Isi dari berkas CSV setelah program dimatikan, ditunjukkan pada Tabel 4.



Tabel 4. Rekapitulasi Jumlah Orang yang Masuk dan Keluar Ruangan

Time	In	Out	Total Inside
2021-04-24 11:02:33	1	0	1
2021-04-24 11:21:44	2	0	2
2021-04-24 11:55:45	3	0	3
2021-04-24 12:04:12	4	1	3
2021-04-24 12:15:08	5	1	4
2021-04-24 12:26:31	6	1	5
2021-04-24 12:33:32	7	1	6
2021-04-24 12:41:56	8	1	7
2021-04-24 13:02:34	9	2	7
2021-04-24 13:12:06	10	2	8
2021-04-24 13:21:13	11	2	9
2021-04-24 13:40:59	12	2	10

C. Pengujian Notifikasi E-mail

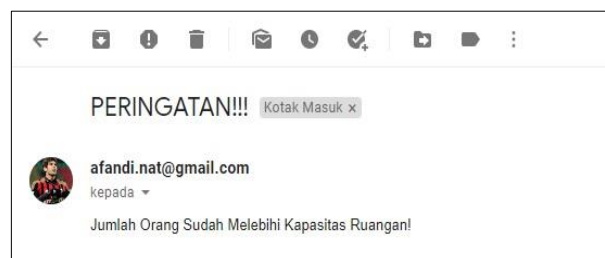
Selain monitoring jumlah orang melalui berkas CSV, terdapat juga notifikasi email apabila pengunjung telah melampaui kapasitas ruangan yang ditetapkan. Pada program yang dibangun, ditetapkan jumlah orang yang diperbolehkan adalah 10 orang. Apabila jumlah melebihi itu maka sistem akan mengirimkan email, sesuai dengan alamat yang telah ditentukan. Email yang dipakai adalah email milik google, setelah sebelumnya dilakukan pengaturan untuk mengizinkan mengirim email melalui aplikasi yang belum terdaftar.

Melalui pengujian yang dilakukan, diperoleh hasil bahwa sistem berhasil mengirimkan email apabila batas maksimal okupansi ruangan terpenuhi. Tabel 5 menunjukkan bahwa siswa akan mengirimkan notifikasi email pada saat total pengunjung di dalam ruangan berjumlah 10 atau lebih. Sistem akan terus mengirimkan email sampai jumlah pengunjung kurang dari 10 orang.

Tabel 5. Pengujian Notifikasi Email

Waktu	Masuk	Keluar	Total	Notifikasi Email
11:02:33	1	0	1	Tidak Ada
11:21:44	2	0	2	Tidak Ada
11:55:45	3	0	3	Tidak Ada
12:04:12	4	1	3	Tidak Ada
12:15:08	5	1	4	Tidak Ada
12:26:31	6	1	5	Tidak Ada
12:33:32	7	1	6	Tidak Ada
12:41:56	8	1	7	Tidak Ada
13:02:34	9	2	7	Tidak Ada
13:12:06	10	2	8	Tidak Ada
13:21:13	11	2	9	Tidak Ada
13:40:59	12	2	10	Terkirim
13:58:11	13	2	11	Terkirim
14:25:27	14	2	12	Terkirim
14:46:16	14	3	11	Terkirim
14:47:23	14	4	10	Terkirim
15:34:52	14	5	9	Tidak Ada

Isi dari email peringatan yang masuk tidak dianggap sebagai spam, dan berisi kalimat peringat seperti ditunjukkan pada Gambar 7.



Gambar 7. Pengujian Pemberitahuan Lewat Email

D. Pengujian Jarak Objek dan Intensitas Cahaya

Sebuah objek dapat ditangkap oleh kamera apabila objek tersebut mendapatkan cukup cahaya dari lingkungan sekitarnya. Oleh karena itu intensitas cahaya dapat mempengaruhi kinerja sistem dalam mendeteksi objek. Pada penelitian ini dilakukan pengujian intensitas cahaya menggunakan parameter yang sesuai dengan standart *Illuminating Engineering Society* (IES). Berdasarkan pada standar IES terdapat tiga tingkatan intensitas cahaya yaitu terang, redup, dan gelap [20]. Keadaan cahaya terang memiliki nilai intensitas lebih dari 100 lux. Sedangkan redup memiliki nilai 20 sampai 99 lux, dan lingkungan gelap memiliki nilai intensitas kurang dari 20 lux. Adapun hasil pengujian deteksi objek orang pada intensitas cahaya yang berbeda-beda ditunjukkan pada Tabel 6.

Tabel 6. Pengujian Intensitas Cahaya

Pengujian	Terang > 100 lux	Redup 20 - 99 lux	Gelap < 20 lux
5x	100%	80%	20%
10x	100%	60%	20%
20x	90%	55%	25%

Data pada Tabel 6 menunjukkan bahwa kondisi cahaya paling ideal untuk deteksi objek adalah pada saat kondisi pencahayaan terang atau intensitas cahaya lebih dari 100 lux. Hal ini ditunjukkan dengan tingkat akurasi 100% pada 10 kali percobaan. Sedangkan pada kondisi redup memiliki akurasi 80% saat dilakukan 5 kali pengujian. Namun tidak disarankan untuk mengambil data pada saat kondisi cahaya redup karena memiliki akurasi deteksi objek yang kecil. Pada kondisi gelap, sistem tidak dapat mendeteksi adanya objek. Oleh sebab itu perlu menempatkan alat pada lingkungan dengan intensitas cahaya tinggi, atau menggunakan bantuan penerangan lampu.

Selain intensitas cahaya, parameter lain yang perlu diperhatikan dalam memasang alat penghitung jumlah orang ini adalah jarak dengan objek manusia yang ingin dipantau. Jangan sampai memasang ditempat yang terlalu tinggi sehingga objek manusia yang ingin dihitung tidak terdeteksi. Pengujian jarak dilakukan beberapa kali untuk mendapatkan nilai hasil yang akurat. Adapun data



hasil pengujian jarak objek dengan kamera dengan intensitas cahaya terang ditunjukkan pada Tabel 7.

Tabel 7. Pengujian Jarak dengan Intensitas Cahaya Terang

Jarak	Pengujian		
	5x	10x	20x
1 meter	100%	100%	100%
2 meter	100%	100%	100%
3 meter	100%	100%	100%
4 meter	100%	100%	100%
5 meter	100%	100%	100%
6 meter	100%	100%	100%
7 meter	100%	100%	80%
8 meter	100%	70%	60%
9 meter	100%	70%	60%
10 meter	50%	50%	50%
11 meter	50%	50%	20%

Pada data Tabel 7 diketahui bahwa pada jarak 1 sampai 6 meter sistem dapat mendeteksi semua objek manusia dengan akurasi 100%. Jarak diatas 7 meter masih dapat mendeteksi objek, namun pada 20 kali pengujian akurasi dari sistem cenderung menurun hingga dibawah 60%. Untuk mengatasi hal tersebut dapat mengganti kamera dengan resolusi yang lebih tinggi.

Berdasarkan hasil pengujian yang disajikan pada Tabel 5 dan Tabel 6, diketahui bahwa sistem yang dibangun mampu mengidentifikasi objek yang berupa manusia pada jarak maksimal 10 meter. Hal ini membuktikan bahwa sistem mampu mengidentifikasi perubahan nilai piksel pada *frame*, meskipun piksel tersebut sangat kecil nilainya.

Sistem mampu mendeteksi objek manusia dengan intensitas cahaya terang. Namun untuk lingkungan yang kurang cahaya atau redup sistem tidak mampu mengidentifikasi objek manusia pada jarak lebih dari 7 meter. Pada lingkungan yang gelap sistem hanya dapat mendeteksi objek pada jarak kurang dari 4 meter. Hal ini terjadi karena sistem mengubah citra pada tiap *frame* menjadi citra *grayscale*. Saat keadaan lingkungan gelap objek tidak teridentifikasi karena warnanya menyatu dengan lingkungan disekitar [21].

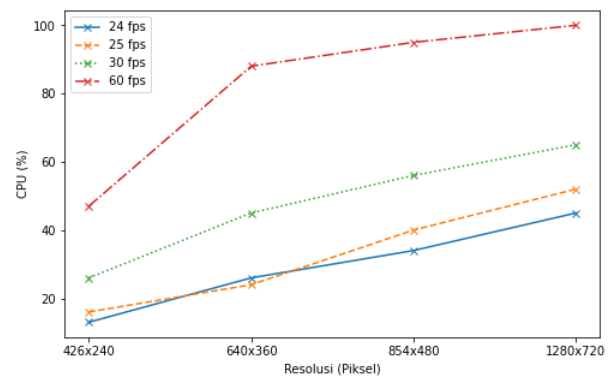
E. Pengujian Jumlah *Frame*

Pengujian ini bertujuan untuk mengetahui kemampuan dari sistem pada saat memproses masukan berupa video secara *real-time* [22]. Semakin besar jumlah *frame* akan membuat kualitas video semakin baik. Namun dengan banyaknya jumlah *frame* yang diproses, maka akan membebani kinerja dari prosesor dan memori. Pengujian ini dilakukan dengan membandingkan nilai *Frame Per Second* (FPS) dari video standart movie (*moving picture*) dan televisi, yaitu 24, 25, 30, dan 60 fps. Kemudian untuk resolusi layar yang pakai sebagai pembandingan adalah layar ukuran NTSC hingga *High Definition* (HD). Adapun hasil pengujian jumlah *frame* disajikan pada Tabel 8.

Tabel 8. Pengujian Jumlah *Frame*

FPS	Resolusi	Kinerja CPU	Keterangan
24	426x240	13%	Lancar
	640x360	26%	Lancar
	854x480	34%	Lancar
	1280x720	45%	Lancar
25	426x240	16%	Lancar
	640x360	24 %	Lancar
	854x480	40%	Lancar
	1280x720	52%	Lancar
30	426x240	26%	Lancar
	640x360	45%	Lancar
	854x480	56%	Lancar
	1280x720	65%	Lancar
60	426x240	47%	Lancar
	640x360	88%	Patah-Patah
	854x480	95%	Patah-Patah
	1280x720	100%	<i>Freeze</i>

Berdasarkan data pada Tabel 8, dapat diketahui kenaikan kinerja CPU pada saat jumlah fps semakin besar. Kenaikan yang signifikan terjadi pada saat *frame* berjumlah 60 fps. Agar lebih jelas mengenai kenaikan kinerja CPU, dapat dilihat grafik pada Gambar 8.



Gambar 8. Perbandingan Kenaikan Kinerja CPU Pada Tiap *Frame*

Data pada Tabel 8 juga menunjukkan bahwa keluaran video yang dihasilkan sistem dapat berjalan dengan lancar pada jumlah *frame* 24, 25 dan 30. Pada jumlah *frame* 60 fps video yang dihasilkan tidak dapat dinikmati karena patah-patah, bahkan pada saat resolusi HD video tidak dapat dijalankan. Penggunaan jumlah *frame* 60 fps dapat membebani kinerja CPU dan GPU pada Raspberry Pi 4 Model B [23]. Agar dapat menjalankan 60 fps, dapat mengganti perangkat pemroses yang memiliki spesifikasi lebih tinggi seperti intel galileo, jetson nano, dll.

IV. KESIMPULAN

Pada penelitian ini dibangun sebuah sistem penghitung orang otomatis menggunakan metode SSD untuk mengidentifikasi objek orang dan metode *centroid tracking* untuk menghitung jumlah orang. Penelitian ini hanya menghitung objek yang berupa manusia, sedangkan objek selain manusia tidak akan dihitung.



Pengujian perhitungan jumlah orang yang telah dilakukan menunjukkan bahwa sistem memiliki akurasi 100% apabila jumlah orang yang lewat secara bersamaan kurang dari 4. Sedangkan apabila jumlah orang yang lewat bersamaan berjumlah 5 sistem tidak dapat menghitung dengan akurat dikarenakan terdapat bagian tubuh yang tidak terlihat oleh kamera. Solusi untuk mengatasi hal tersebut adalah dengan menempatkan kamera diatas orang yang melintas, dengan sudut 60° dan jarak 10 meter. Melalui skenario tersebut, setiap orang yang melintas akan terlihat dengan jelas anggota tubuhnya.

Sistem dapat mengirimkan notifikasi email apabila jumlah orang yang masuk melebihi kapisitas yang ditentukan. Berdasarkan pengujian sistem akan terus mengirim email bila jumlah orang terus bertambah. Rekapitulasi jumlah orang yang masuk dan keluar akan disimpan dalam file *.csv saat program dimatikan. Pengujian lainnya yang dilakukan adalah pengujian jarak dan intensitas cahaya. Pada pengujian tersebut menunjukkan bahwa jarak maksimal objek dapat di deteksi sistem adalah 10 meter. Sedangkan intensitas cahaya yang kurang atau gelap akan membuat sistem tidak dapat mengidentifikasi objek karena nilai perubahan piksel yang terlalu kecil. Terakhir pada pengujian jumlah frame, sistem dapat menampilkan hasil video dengan lancar pada 24, 25, 30 fps. Namun apabila jumlah frame lebih dari 30 maka hasil video akan patah-patah atau cenderung freeze.

DAFTAR PUSTAKA

- [1] R. Girasa, "AI as a Disruptive Technology," in *Artificial Intelligence as a Disruptive Technology*, Palgrave Macmillan, Cham, 2020, pp. 3–21.
- [2] Kemendagri, *Penegakan Protokol Kesehatan Untuk Pengendalian Penyebaran Corona Virus Disease 2019*. Indonesia: Kementerian Dalam Negeri Republik Indonesia, 2020, pp. 1–5.
- [3] F. D. Adhinata, M. Ikhsan, and W. Wahyono, "People counter on CCTV video using histogram of oriented gradient and Kalman filter methods," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 3, pp. 222–227, 2020, doi: 10.14710/jtsiskom.2020.13660.
- [4] D. Y. Setiawan, H. Fitriyah, and I. Arwani, "Sistem Penghitung Jumlah Orang Melewati Pintu Menggunakan Metode Background Subtraction Berbasis Raspberry Pi," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 2, pp. 2105–2113, 2019.
- [5] A. K. Mahamad, S. Saon, H. Hashim, M. A. Ahmadon, and S. Yamaguchi, "Cloud-based people counter," *Bulletin Electrical Engineering and Informatics*, vol. 9, no. 1, pp. 284–291, 2020, doi: 10.11591/eei.v9i1.1849.
- [6] T. Parthornratt, N. Burapanonte, and W. Gunjarueg, "People identification and counting system using raspberry Pi (AU-PiCC: Raspberry Pi customer counter)," in *International Conference on Electronics, Information, and Communication (ICEIC)*, Da Nang, Vietman, Jan. 2016, doi: 10.1109/ELINFOCOM.2016.7563020.
- [7] T. Y. Chen, C. H. Chen, D. J. Wang, and Y. L. Kuo, "A people counting system based on face-detection," in *4th International Conference on Genetic and Evolutionary Computing (ICGEC)*, Shenzhen, China, Dec. 2010, pp. 699–702, doi: 10.1109/ICGEC.2010.178.
- [8] D. W. Nugraha, Y. Anshori, and N. K. Candriasih, "Rancang Bangun Sistem Penghitung Jumlah Pengunjung Perpustakaan Menggunakan Metode Haar like Features (Studi Kasus Pada Perpustakaan Universitas Tadulako)," *ScientiCO : Computer Science and Informatics Journal*, vol. 1, no. 1, p. 57, 2019, doi: 10.22487/j26204118.2018.v1.i1.11902.
- [9] W. Liu *et al.*, "SSD: Single shot multibox detector," *Lecturer Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence Lecture Notes in Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2.
- [10] J. C. Nascimento, A. J. Abrantes, and J. S. Marques, "Algorithm for centroid-based tracking of moving objects," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Phoenix, AZ, USA, August. 1999, vol. 6, no. 1, pp. 3305–3308, doi: 10.1109/icassp.1999.757548.
- [11] A. A. Suzen, B. Duman, and B. Sen, "Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN," in *2nd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Ankara, Turkey, Jun. 2020, pp. 3–7, doi: 10.1109/HORA49412.2020.9152915.
- [12] J. A. Kim, J. Y. Sung, and S. H. Park, "Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition," in *International Conference on Consumer Electronics (ICCE)*, Taoyuan, Taiwan, Sep. 2020, pp. 8–11, doi: 10.1109/ICCE-Asia49877.2020.9277040.
- [13] E. Cengil, A. Çinar, and Z. Güler, "A GPU-based convolutional neural network approach for image classification," in *International Artificial Intelligence and Data Processing Symposium (IDAP)*, Malatya, Turkey, Sep. 2017, doi: 10.1109/IDAP.2017.8090194.
- [14] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," in *International Conference on Learning Representations*, Toulon, France, April. 2017, pp. 1–9, [Online]. Available: <http://arxiv.org/abs/1704.04861>.
- [15] D. Sinha and M. El-Sharkawy, "Thin MobileNet:



- An Enhanced MobileNet Architecture,” in *10th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, New York, USA, Oct. 2019, pp. 0280–0285, doi: 10.1109/UEMCON47517.2019.8993089.
- [16] H.-T. Kim, Gyu-yeong; Kim, Jae-Ho; Park, Jang-Sik; Yu, Yun-Sik; Kim, “Vehicle Tracking using Euclidean Distance,” *Journal Korea Institute Electronic Communication Sciences*, vol. 7, no. 6, pp. 1293–1299, 2012, doi : <https://doi.org/10.13067/JKIECS.2012.7.6.1293>.
- [17] S. N. Jyothi and K. V. Vardhan, “Design and implementation of real time security surveillance system using IoT,” in *International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, Oct. 2016, doi: 10.1109/CESYS.2016.7890003.
- [18] D. Bader, “Schedule,” *readthedocs.io*, 2020. <https://schedule.readthedocs.io/en/stable/> (accessed May 12, 2021).
- [19] Y. Li, H. Huang, Q. Xie, L. Yao, and Q. Chen, “Research on a surface defect detection algorithm based on MobileNet-SSD,” *Applied Sciences Journal*, vol. 8, no. 9, 2018, doi: 10.3390/app8091678.
- [20] David L. DiLaura, *An Introduction to The IES Lighting Handbook 10 th*. Forgotten Books, 2017.
- [21] N. N. Putri, “Aplikasi Pendeteksi Objek bergerak pada Image Sequence Dengan Metode Background Substraction,” *Jurnal Teknologi Rekayasa.*, vol. 21, no. 3, pp. 162–172, 2016.
- [22] R. D. Ramadhani, A. N. A. Thohari, and N. A. Nugraha, “Sistem Keamanan Ruang Berbasis Internet of Things Menggunakan Single Board Computer,” *Jurnal Nasional Informatika dan Teknologi Jaringan*, vol. 1, no. 2, pp. 0–5, 2020, doi: <https://doi.org/10.30743/infotekjar.v4i2.2338>.
- [23] E. Rohadi *et al.*, “Internet of Things: CCTV Monitoring by Using Raspberry Pi,” in *the first International Conference on Applied Science and Technology (iCAST)*, Manado, Indonesia, Oct, 2018, pp. 454–457, doi: 10.1109/iCAST1.2018.8751612.

