



# Penghitung Pengunjung dan Deteksi Masker Menggunakan OpenCV dan YOLO

Agung Wibowo Ardiyanta Surbakti<sup>1</sup>, Rahmi Eka Putri.<sup>2</sup>

<sup>1,2</sup> Jurusan Teknik Komputer, FTI Universitas Andalas Limau Manis Kec. Pauh, Kota Padang, Sumatera Barat, 25163, Indonesia

### ARTICLE INFORMATION

#### Sejarah Artikel:

Diterima Redaksi: 8 September 2021

Revisi: 14 September 2022

Ditebitkan Online: 31 Oktober 2022

### KEYWORDS

Visitor, YOLO, Face Mask, Covid-19

### CORRESPONDENCE

Phone: +62 (0751) 12345678

E-mail: agungardiyanta@gmail.com

### A B S T R A C T

The spread of COVID-19 that occurs through droplets can be avoided by reducing contact between individuals, so it is necessary to limit visitors, especially in crowded places such as shopping centers to avoid transmission between visitors. This study utilizes YOLOv3 object detection to recognize objects from camera image input, which is implemented on the Raspberry Pi 4, to identify visitors and use masks. The results of the identification of human objects will be calculated to determine the number of visitors at the shopping center. Then a buzzer sound warning is given when visitors are not wearing masks, if visitors exceed the capacity limit, they are also given a warning via an android application to the building manager. The results of the model detection show the mAP value of 77.92% for 3 classes of mask objects, without masks and humans.

## PENDAHULUAN

Penyebaran dari COVID-19 yang disebabkan oleh coronavirus dimulai pada bulan Desember tahun 2019 di Wuhan China dan telah meluas penyebarannya sampai keluar wilayah China. Meluasnya penyebaran dari COVID-19 di dunia, hal ini dimudahkan oleh cara penularan virus itu sendiri yang dapat berpindah antara manusia melalui medium droplet, tepat pada tanggal 11 Maret 2020 World Health Organization (WHO) menetapkan wabah COVID-19 menjadi pandemi global [1]. Kejadian ini telah menyebabkan keadaan baru dimasyarakat dalam berkegiatan sehari-hari yang tidak dapat kembali leluasa seperti biasanya dikarenakan munculnya kekhawatiran akan kemungkinan penyebaran virus ini. Sedangkan di Indonesia sendiri pemerintah dengan resmi memberikan pengumuman kasus pertama COVID-19 pada tanggal 2 Maret 2020. Lalu munculnya korban jiwa pertama pada tanggal 11 Maret 2020 dari kasus di Solo Jawa Tengah [2].

Kementerian Kesehatan (Kemenkes) Melalui Peraturan Menteri Kesehatan Nomor 9 Tahun 2020 memberikan panduan tata cara pencegahan penularan Covid-19 di tempat umum seperti memakai masker, mencuci tangan dan juga tetap menjaga jarak antar tiap individu. Lalu Kementerian Perdagangan melalui Surat Edaran Nomor 12 tahun 2020 Tentang Pemulihan Aktivitas Perdagangan Yang Dilakukan Pada Masa Pandemi Corona Virus Disease 2019 (COVID-19) Dan New Normal juga mengeluarkan

<https://doi.org/10.25077/chipset.3.02.83-93.2022>

peraturan yang harus dilaksanakan pelaku bisnis pusat perbelanjaan dalam membatasi jumlah pengunjung untuk menghindari kontak fisik dan tetap menerapkan pencegahan dan pengendalian penyebaran Covid-19 di tempat keramaian.

Berdasarkan aturan yang berlaku itulah maka diperlukan cara untuk menghitung pengunjung pusat perbelanjaan dan mendeteksi penggunaan masker untuk mendukung penerapan pencegahan penyebaran COVID-19 di tempat umum, pada penelitian yang terdahulu [3] [4] telah dilakukan penghitungan pada jumlah orang yang melewati pintu pada gedung namun hanya menghasilkan keluaran berupa jumlah pengunjung pada LCD matriks tanpa adanya tampilan lanjutan dan belum tersedianya fitur pemberitahuan pembatasan kapasitas serta deteksi penggunaan masker sesuai dengan keadaan yang merujuk peraturan Menteri Kesehatan dan surat edaran Menteri Perdagangan mengenai pengunjung di pusat perbelanjaan pada masa pandemik COVID-19.

Berdasarkan paparan di atas maka penulis ingin memunculkan opsional cara lain yang dapat digunakan pengelola gedung untuk mengontrol jumlah pengunjung dan mendeteksi penggunaan masker, berupa sistem yang mampu menghitung jumlah pengunjung yang masuk dan keluar gedung dan mendeteksi penggunaan masker untuk itu dibuatlah sistem pada penelitian ini.

## LANDASAN TEORI

### Coronavirus

Coronavirus adalah virus yang menyerang saluran pernapasan manusia dari masalah seperti flu sampai penyakit yang lebih serius seperti Middle East Respiratory Syndrome (MERS) dan Severe Acute Respiratory Syndrome (SARS), juga sering dikenal dengan sindrom pernapasan akut berat. Pada bulan Desember tahun 2019 di Wuhan China telah ditemukan jenis baru dari COVID-19 yang diberi nama dengan Severe Acute Respiratory Syndrome COVID-19 2 (SARS-COV2) virus jenis baru ini menyebabkan penyakit Corona Virus Disease 2019 (COVID-19) [5]. Infeksi pernapasan COVID-19 dapat ditularkan melalui droplet yang merupakan tetesan kecil saat seseorang berbicara, batuk dan bersin. Penularan melalui droplet dapat terjadi jika seseorang berada sangat dekat (1 meter) dengan seseorang yang memiliki gejala pernapasan terinfeksi COVID-19. Gejala umum dari COVID-19 berupa demam  $\geq 38^{\circ}\text{C}$ , batuk kering, dan sesak napas. Namun beberapa kasus ditemukan tanpa gejala sama sekali atau hanya gejala batuk ringan saja. COVID-19. Bahaya yang disebabkan COVID-19 seperti penyakit pernapasan lainnya, COVID-19 dapat menyebabkan gejala ringan termasuk pilek, sakit tenggorokan, batuk, dan demam. Sekitar 80% kasus dapat pulih tanpa perlu perawatan khusus. Sekitar 1 dari setiap 6 orang mungkin akan menderita sakit yang parah, seperti disertai pneumonia atau kesulitan bernapas, yang biasanya muncul secara bertahap dan juga kasus kematian. Walaupun angka kematian penyakit ini masih rendah (sekitar 3%), namun bagi orang yang berusia lanjut, dan orang-orang dengan kondisi medis yang sudah ada sebelumnya (seperti diabetes, tekanan darah tinggi dan penyakit jantung), biasanya lebih rentan untuk menjadi sakit parah. Pada masa sekarang ilmuwan di dunia sedang mengerjakan vaksin untuk mengatasi COVID-19 yang masih berada pada tahap uji coba [6].

### COVID-19 di Indonesia

Kasus penyebaran COVID-19 pertama kali di Indonesia resmi diumumkan oleh pemerintah pada tanggal 2 Maret 2020, berjumlah 2 orang yang diperkirakan tertular melalui seorang WN Jepang. Lalu pada tanggal 11 Maret kasus kematian pertama yang disebabkan COVID-19 terjadi di Solo Jawa Tengah [7]. Kasus penyebaran COVID-19 semakin meningkat di Indonesia data terakhir pada tanggal 27 Juli 2020, telah terkonfirmasi 102.051 kasus dengan jumlah dalam perawatan 36.611 jiwa, jumlah sembuh 60.539 jiwa dan jumlah kematian 4.901 jiwa [8]. Sebelumnya pemerintah telah mengambil kebijakan untuk menekan penyebaran COVID-19 pada tanggal 3 April 2020 dengan ditetapkannya Peraturan Kementerian Kesehatan (PMK) nomor 9 tahun 2020 tentang pedoman pemberlakuan Pembatasan Sosial Berskala Besar (PSBB) yang bertujuan membatasi ruang gerak di masyarakat untuk mencegah perluasan penyebaran COVID-19. PSBB ini dapat diterapkan oleh tiap-tiap kepala daerah dengan mengajukan izin kepada kementerian kesehatan terlebih dahulu, pemerintah juga menghimbau agar masyarakat menerapkan protokol kesehatan seperti mencuci tangan, menggunakan masker, serta jaga jarak saat berada di tempat umum untuk mencegah penularan COVID-19. Lalu pada tanggal 28 Mei 2020 Kementerian Perdagangan mengeluarkan Surat Edaran nomor 12 tahun 2020 tentang Pemulihan Aktivitas

Perdagangan yang Dilakukan pada Masa Pandemi Corona Virus Disease 2019 (COVID-19).

### Surat Edaran Menteri Perdagangan Nomor 12 tahun 2020

Melalui surat edaran ini diatur tata cara pelaksanaan kembali aktivitas perdagangan di masa pandemi COVID-19 dengan memperhatikan protokol kesehatan COVID-19. Ruang lingkup dari aturan ini berupa tempat-tempat kegiatan perdagangan yang menyelenggarakan transaksi perdagangan untuk bahan pokok dan barang penting, kebutuhan pangan, bahan bakar minyak, gas, dan energi, fasilitas pelayanan kesehatan atau fasilitas lain dalam rangka pemenuhan pelayanan kesehatan, fasilitas pelayanan pariwisata. Salah satu yang menjadi bahasan pada surat edaran ini adalah jumlah pengunjung yang harus dibatasi untuk tetap menerapkan protokol kesehatan COVID-19. Masing-masing tempat memiliki persentase batasan pengunjung dari jumlah kunjungan pada saat kondisi normal. Nilai batas itu dibedakan sesuai jenis tempat/fasilitas perdagangan Pasar Rakyat (30%), Toko Swalayan (Minimarket, Supermarket, Hypermarket Department Store) (40%), Restoran/atau rumah Makan/ Warung Makan, Kafe (40%), Toko obat/ Farmasi dan Alat Kesehatan (40%), Mall atau Pusat Perbelanjaan (35%), Restoran di Rest area (40%) dan Tempat Hiburan tertentu seperti kebun binatang, Museum, Galeri Seni (50%), pembatasan jumlah pengunjung ini bertujuan untuk mencegahnya terjadi kepadatan yang dapat menyulitkan untuk menerapkan jaga jarak sesuai dengan protokol kesehatan yang berlaku

### OpenCV



Gambar 1. Logo OpenCV

OpenCv (Open Source Computer Vision Library) adalah sebuah library open source untuk computer vision dan machine learning. OpenCV dibangun untuk menyediakan infrastruktur umum untuk aplikasi computer vision dan mempercepat penggunaan machine perception pada produk komersil. Menjadi produk yang berlisensi BSD, OpenCV memudahkan bisnis untuk memanfaatkan dan memodifikasi kode. Library ini memiliki lebih dari 2500 algoritma yang dioptimalkan, yang mencakup serangkaian komprehensif computer vision dan algoritma mesin dan pembelajaran mesin yang mutakhir. Algoritma ini dapat digunakan untuk mendeteksi dan mengenali wajah, mengidentifikasi objek, mengklasifikasikan tindakan manusia dalam video, melacak pergerakan kamera, melacak objek bergerak, mengekstraksi model objek 3D, menghasilkan awan titik 3D dari kamera stereo, menjahit gambar bersama untuk menghasilkan resolusi tinggi gambar seluruh adegan, temukan gambar serupa dari basis data gambar, hapus mata merah dari gambar yang diambil menggunakan flash, ikuti gerakan mata, kenali pemandangan dan membuat penanda untuk melapisinya dengan augmented reality dan lainnya. OpenCV memiliki lebih dari 47 ribu orang pengguna komunitas dan perkiraan jumlah unduhan melebihi 18 juta. Perpustakaan digunakan secara luas di perusahaan, kelompok penelitian dan oleh badan pemerintah [9].

**YOLO**

You Only Look Once (YOLO) pertama kali diciptakan oleh Joseph Redmon pada tahun 2015 adalah sistem deteksi objek secara realtime berdasarkan CNN (Convolutional Neural Network). Pada konferensi CVPR (Conference on Computer Vision and Pattern Recognition) pada 2017, Joseph Redmon dan Ali Farhadi merilis YOLO v2 telah meningkatkan akurasi dan kecepatan algoritma. Pada April 2018, Joseph Redmon dan Ali Farhadi merilis YOLO v3 terbaru di mana memiliki performance / kinerja yang semakin meningkat pada deteksi objek [10]. YOLO bekerja dengan menerapkan jaringan saraf tunggal ke gambar penuh. Jaringan ini membagi gambar menjadi daerah dan memprediksi kotak pembatas dan probabilitas untuk setiap wilayah. Kotak pembatas diberikan nilai bobot oleh probabilitas yang diprediksi. YOLO memiliki keunggulan dibandingkan sistem berbasis classifier. YOLO menggunakan prediksi dengan evaluasi jaringan saraf tunggal inilah yang membedakan dari sistem R-CNN yang membutuhkan ribuan untuk satu gambar. Perbedaan ini membuat YOLO sangat cepat 1000 kali lebih cepat dari R-CNN dan 100 kali lebih cepat dari Fast R-CNN [11].

**Pengujian Kualitas Model dengan mean Average Precisions (mAP)**

Mean average precisions merupakan salah satu pengukuran yang digunakan untuk mengevaluasi kemampuan dari model machine learning. Berikut beberapa parameter untuk pengukuran performa dalam mendapatkan nilai mAP model:

Precision merupakan nilai prediksi yang benar positif dibandingkan dengan seluruh hasil yang diprediksi positif oleh model perhitungan untuk mendapatkan nilai precision dapat dilihat berikut ini

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

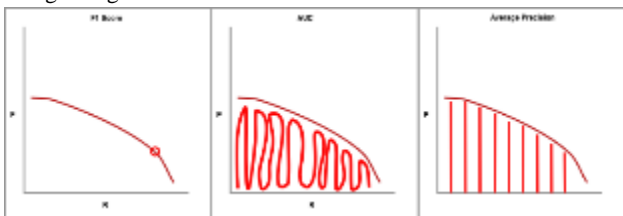
Recall merupakan tingkat sensitifitas model dimana recall merupakan hasil dari rasio prediksi benar positif dibandingkan dengan seluruh data yang benar positif untuk mencari nilai recall kita menggunakan rumus sebagai berikut:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

F-1 score merupakan nilai perbandingan rata-rata presisi dan recall yang dibobotkan nilai ini menunjukkan titik keseimbangan antara nilai precision dan recall yang memiliki rumus sebagai berikut:

$$F - 1 = \frac{2 * Recall * Precision}{Recall + Precision} \quad (3)$$

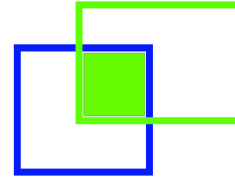
Dari 3 parameter di atas, untuk menghitung Average Precisions (AP) digunakan kurva dari precision-recall (kurva F1) untuk menghitung area di bawah kurva tersebut.



Gambar 2. Kurva precisions-recall

Dalam deteksi objek, prediksi dilakukan dalam representasi bounding box dan kelas label. Dalam penerapannya, penentuan bounding box melalui koordinat X, Y, h, dan w akan menghasilkan sedikit ketidakcocokan dengan objek aslinya (ground truth). Untuk mengatasi hal tersebut, dikenalkan metode

IOU (Intersection Over Union) yang menyediakan metrik untuk menyesuaikan batasan bounding box dan menentukan bounding box yang pas dari beberapa bounding boxes yang saling tumpang tindih. IoU biasanya diatur pada threshold 60 sampai 75 persen. Jadi, IoU digunakan untuk validasi deteksi apakah hasil prediksi benar atau tidak [12].



Gambar 3. Sketsa proses penentuan bounding boxes dengan metrik IoU

Setelah didapatkan hasil perhitungan metrik-metrik di atas, selanjutnya dilakukan Perhitungan AP atau tingkat akurasi prediksi rata-rata atau area di bawah kurva precision-recall. Hal ini didapatkan dengan persamaan:

$$AP = \frac{1}{11} \sum_{r \in \{0,0.1, \dots, 1\}} P_{interp}(r) \quad (4)$$

Dimana  $AP_i$  merupakan skor AP (Average Precisions) untuk masing-masing kelas objek (N) [12].

**Raspberry Pi**



Gambar 4. Raspberry Pi 4

Raspberry Pi yang biasa dikenal dengan raspi adalah Single board computer (SBC) berukuran kartu kredit berbiaya rendah yang dapat dihubungkan ke monitor komputer atau TV, dan menggunakan keyboard dan mouse standar. Raspi mampu melakukan semua yang bisa dikerjakan pada komputer desktop, mulai dari menjelajah internet dan memutar video definisi tinggi, hingga membuat spreadsheet, pemrosesan kata, dan bermain game. Raspi dikembangkan oleh Yayasan Raspberry Pi yang berlokasi di Inggris [13]. Raspi memiliki beberapa model yaitu zero, A, B, A+ dan B+ yang memiliki kelebihan masing-masing. Raspi memiliki beberapa versi seperti Raspberry Pi 1 sampai yang terbaru Raspberry Pi 4.

**Modul Kamera Raspberry Pi V2**



Gambar 5. Modul Kamera Raspberry Pi v2

Modul Kamera Raspberry Pi v2 menggantikan Modul Kamera asli pada bulan April 2016. Modul Kamera v2 memiliki sensor Sony IMX219 8-megapiksel (dibandingkan dengan sensor OmniVision OV5647 5-megapiksel dari kamera asli). Modul

Kamera dapat digunakan untuk mengambil video definisi tinggi, dan juga foto. Modul kamera ini mendukung mode 1080p30, 720p60 dan VGA90. Modul kamera ini dapat dihubungkan dengan kabel pada port CSI di Raspberry Pi [14].

## METODE PENELITIAN

### Rancangan Umum Sistem

Rancangan umum sistem menunjukkan bagaimana tiap komponen dari sistem dapat terintegrasi dalam satu kesatuan. Rancangan yang akan diterapkan dapat dilihat pada diagram gambar berikut ini:

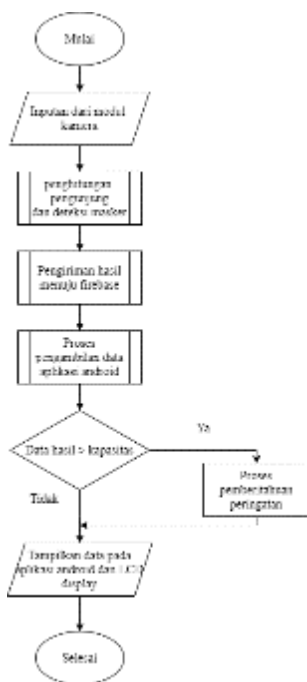


Gambar 6. Rancangan umum sistem

rancangan umum sistem penghitung pengunjung ini adalah modul kamera yang akan mengambil video akan diarahkan pada pintu masuk pengunjung, hasil tangkapan kamera diproses pada Raspberry Pi untuk dikenali sebagai objek manusia juga mendeteksi penggunaan masker jika terdeteksi tidak menggunakan masker maka modul buzzer akan berbunyi sebagai peringatan, kemudian objek manusia yang dikenali akan dihitung jumlahnya hasil akan disimpan. Hasil dari perhitungan akan menjadi nilai perubahan pengunjung yang masuk dan keluar melalui pintu masuk. Kemudian hasil perhitungan akan dikirim menuju database Firebase dan disimpan dan akan terus diperbarui saat terjadi perubahan data jumlah pengunjung di dalam gedung.

### Rancangan Umum Proses

Rancangan proses secara umum yang akan digunakan pada sistem penghitung pengunjung ini dapat dilihat pada gambar 15



Gambar 7. Rancangan umum proses

Sistem ini akan bekerja dengan tahapan dimulai dari proses inputan video dari modul kamera yang terhubung dengan Raspberry Pi, video tangkapan kamera akan diproses pada Raspberry Pi untuk dikenali sebagai objek manusia hasil dan mendeteksi penggunaan masker pada pengunjung dalam tahap proses ini jika pengunjung tidak menggunakan masker maka buzzer akan berbunyi. Hasil deteksi kembali diproses untuk dilakukan perhitungan jumlah pengunjung. Hasil hitung akan dikirim menuju Firebase melalui jaringan internet. Aplikasi android pengguna yang terhubung ke Firebase akan mengakses data hasil yang terdapat di Firebase. Hasil akan dibandingkan dengan kapasitas sesuai batas aman yang telah diinputkan sebelumnya oleh pengelola pada aplikasi android untuk jumlah pengunjung sesuai edaran menteri perdagangan. Jika hasil melewati dari batas aman maka sistem akan mengirimkan pemberitahuan peringatan pada aplikasi android pengguna, data yang akan ditampilkan pada aplikasi android seperti jumlah orang dan keterangan jumlah masuk dan keluar pengunjung gedung. Jika hasil tidak melewati batas aman maka data hanya ditampilkan pada halaman aplikasi android dan juga pada modul LCD pada sistem. Sistem akan memperbarui data pada Firebase saat adanya perubahan data pengunjung sehingga pengguna diharapkan mendapatkan data secara tepat.

## HASIL DAN ANALISA

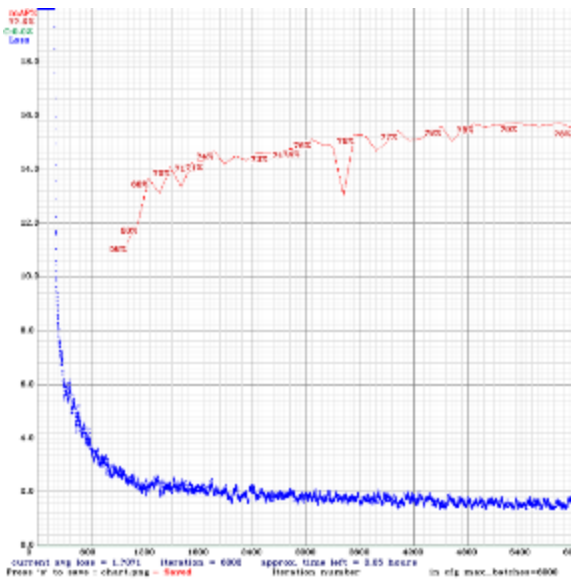
### Implementasi Perangkat Keras

Komponen yang digunakan pada implementasi perangkat keras sistem yaitu:

1. Modul kamera raspi v2 dengan resolusi 8 *MegaPixel* (MP), berfungsi menangkap citra pada pintu masuk secara *realtime*, hasil tangkapan diolah pada raspi, untuk dilakukan deteksi pengunjung dan masker menggunakan YOLOv3 tiny.
2. Raspberry Pi 4 Model B 4GB RAM, sebagai pusat proses pada sistem mengolah tangkapan citra dan memproses *output* yang dikirim menuju database pada Firebase.
3. Modul LCD display 16x2, berfungsi sebagai tampilan keluaran jumlah pengunjung yang telah terdeteksi.
4. Buzzer berfungsi sebagai keluaran suara peringatan saat pengunjung tidak menggunakan masker terdeteksi.

Komponen-komponen tersebut dihubungkan menjadi satu kesatuan, diletakan di atas menghadap pintu masuk gedung.



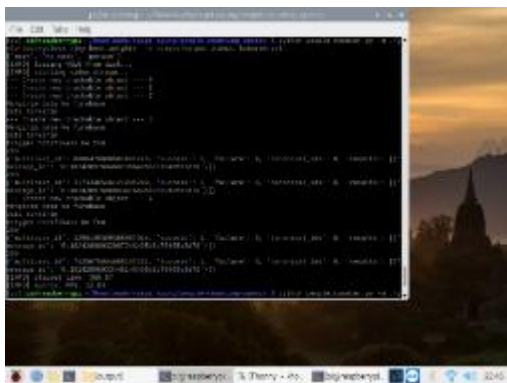


Gambar 12. Grafik hasil pelatihan model dan perhitungan mAP

**Modul implementasi model YOLO pada Raspberry Pi 4**

Hasil model YOLO yang telah dilatih untuk mendeteksi objek masing-masing kelas tersebut akan diterapkan pada raspberry pi 4 untuk mendapatkan deteksi secara langsung melalui *input* pada modul kamera. Alur proses pada implementasi model yolo mengikuti rancangan alur proses poin utama dalam implementasi modul terbagi beberapa bagian yaitu:

1. Deteksi gambar untuk objek manusia, masker dan tanpa masker dari masukan kamera
2. Memberikan *id* objek dan *tracking* objek manusia, ditentukan perubahan arah gerak objek untuk menentukan objek pada posisi masuk atau keluar gedung
3. Menampilkan data pengunjung pada LCD dan mengirim data ke firebase untuk dapat diakses pada aplikasi android
4. Menyalakan *buzzer* saat terdeteksi objek tanpa masker dan mengirim notifikasi pada layanan firebase saat jumlah pengunjung melebihi batasan gedung



Gambar 13. Tampilan pada terminal



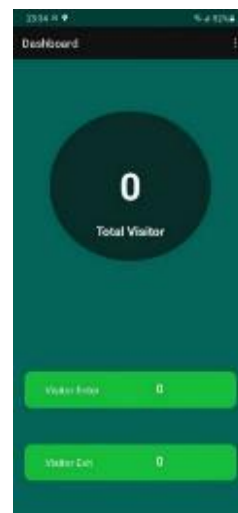
Gambar 14. Deteksi objek



Gambar 15. pemberian id dan tracking

**Implementasi Aplikasi Android**

Aplikasi android dibagi dua tampilan utama pada implementasinya. Pada halaman pertama menjadi tampilan keluaran hasil deteksi dan halaman kedua sebagai antarmuka pengaturan masukan nilai. Implementasi aplikasi android sesuai dengan rancangan aplikasi.



Gambar 16. Tampilan aplikasi android halaman



Gambar 17. Tampilan aplikasi android pengaturan  
Semua nilai yang didapat pada aplikasi disimpan dan diambil dari database firebase. Implementasi pada database firebase dapat dilihat pada gambar berikut.



Gambar 18. Implementasi database aplikasi pada firebase  
Database yang tersimpan pada firebase menggunakan format JSON dengan nama objek yolov3 yang didalamnya terdapat tiga key. Pada key pkeluar memiliki value dengan format integer untuk menerima nilai pengunjung yang keluar, key pmasuk memiliki value dengan format integer untuk menerima nilai pengunjung yang masuk dan key rCap memiliki value dengan format integer untuk menerima nilai batasan pengunjung yang telah dimasukan melalui halaman pengaturan aplikasi, nilai yang tersimpan merupakan nilai bulat hasil presentase dari total kapasitas gedung. Nilai rCap ini menjadi batasan pembanding pada program agar sistem dapat memicu cloud messaging pada firebase untuk mengirim pemberitahuan peringatan pada aplikasi android. Berikut tampilan dari pemberitahuan peringatan pada aplikasi android.

**Pengujian dan Analisa**

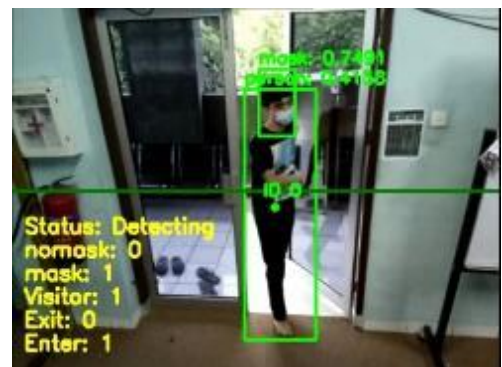
Pengujian dan analisa dilakukan untuk mengukur serta menganalisis hasil dari implementasi perangkat keras dan perangkat lunak, pada penelitian ini dilakukan pengujian pada perangkat keras, perangkat lunak dan fungsional alat atau sistem secara keseluruhan

**Pengujian Modul Kamera**

Modul kamera akan menangkap citra gambar yang menjadi inputan pada sistem maka diperlukan untuk memastikan kondisi pada modul kamera, untuk itu dilakukan pengujian modul kamera dengan melakukan tangkapan gambar pada waktu siang dan malam pada hari yang berbeda dengan hasil uji seperti berikut:

Tabel 1. Hasil Pengujian Pada Kamera

NO	Kondisi	Percobaan-1	Percobaan-2
1	Siang	Terdeteksi	Terdeteksi
2	Malam	Terdeteksi	Terdeteksi
3	Siang	Terdeteksi	Terdeteksi
4	Malam	Terdeteksi	Terdeteksi



Gambar 19. Deteksi pada siang



Gambar 20. Deteksi pada malam

Pada hasil pengujian yang dilakukan pada hari yang berbeda didapatkan bahwa proses deteksi dapat dilakukan baik pada waktu siang hari dan malam hari. Hasil deteksi dipengaruhi oleh pencahayaan yang baik dan posisi objek tidak membelakangi atau

melawan sumber cahaya yang lebih dikenal dengan istilah *backlight* yang akan menyebabkan bayangan di depan kamera.

**Pengujian kemampuan deteksi pada variasi intensitas cahaya**

Pengujian kemampuan deteksi sistem terhadap objek pada nilai intensitas cahaya yang berbeda, untuk mendapatkan intensitas cahaya yang bervariasi dilakukan pengkondisian sumber cahaya pada tempat pengujian, hasil dari pengujian dapat dilihat pada tabel berikut:

Tabel 2. Hasil Pengujian Intensitas Cahaya

NO	Intensitas Cahaya (Lux)	Keterangan
1	443	Objek dapat terdeteksi dan dilakukan tracking
2	368	Objek dapat terdeteksi dan dilakukan tracking
3	220	Objek masker/tanpa masker mulai sulit dideteksi, proses tracking masih dapat dilakukan
4	156	objek masker/tanpa masker tidak terdeteksi tracking tidak berjalan baik
5	60	deteksi tidak dapat dipertahankan untuk dilakukan tracking arah gerak objek
6	18	tidak ada objek yang terdeteksi

Hasil pengukuran intensitas cahaya didapatkan menggunakan sensor LDR diletakan di sebelah posisi kamera. Dari hasil pengujian pada tabel 4.2 menunjukkan pengaruh eksternal intensitas cahaya terhadap kemampuan deteksi sistem, untuk mendapatkan hasil deteksi yang optimal dibutuhkan pencahayaan yang terang pada ruangan sistem.

**Pengujian Koneksi Raspberry Pi**



Gambar 21. Plot latensi pada LAN



Gambar 22. Plot latensi pada Wi-Fi



Gambar 23. Plot latensi pada Hotspot 4G GSM

Hasil pengujian dari tiga jalur konektivitas didapatkan rata-rata latensi masing masing jalur pada LAN 17,95 ms lalu pada WiFi 23,46 ms dan pada Hotspot 4G GSM 68,91 ms dari ketiga perbandingan tersebut dapat dilihat nilai latensi rata-rata terendah didapat pada jalur konektivitas LAN dengan hasil latensi terendah dan lebih stabil. Selain menguji nilai latensi pengiriman juga dilakukan pengujian keberhasilan pengiriman data dari raspberry pi ke database firebase. Langkah pengujian yang dilakukan dengan memberikan input nilai acak yang dikirim pada firebase melihat keberhasilan pengiriman data dan menghitung waktu pengiriman tiap data yang dilakukan otomatis menggunakan program python. Pada hasil pengujian pengiriman data dari raspi ke firebase dapat dilihat keberhasilan pada tiap pengiriman. Waktu tercepat pengiriman data pada 1.42 second dan waktu terlama yang dibutuhkan untuk mengirim data pada 2.12 second lalu dari seluruh percobaan didapat waktu rata-rata pengiriman yaitu 1.53 second.

**Pengujian Model YOLO**

Pengujian model YOLO dilakukan untuk melihat kemampuan deteksi pada tiap kelas mask, nomask, dan person selama proses pada sistem. Pengujian yang dilakukan pertama mengukur performa dari model yolo. Pengukuran yang digunakan sebagai indikator performa model meliputi nilai precision, recall, dan F-1 score merujuk parameter performa model pada subbab 2.4. Precision merupakan nilai prediksi yang benar positif dibandingkan dengan seluruh hasil yang diprediksi positif oleh



model perhitungan untuk mendapatkan nilai precision dapat dilihat berikut ini

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{686}{686 + 190}$$

$$Precision = 0,78$$

Recall merupakan tingkat sensitifitas model dimana recall merupakan hasil dari rasio prediksi benar positif dibandingkan dengan seluruh data yang benar positif untuk mencari nilai recall kita menggunakan rumus sebagai berikut:

$$Recall = \frac{TP}{TP + FN}$$

$$Recall = \frac{686}{686 + 285}$$

$$Recall = 0.71$$

F-1 score merupakan nilai perbandingan rata-rata presisi dan recall yang dibobotkan nilai ini menunjukkan titik keseimbangan antara nilai precision dan recall yang memiliki rumus sebagai berikut:

$$F - 1 = \frac{2 * Recall * Precision}{Recall + Precision}$$

$$F - 1 = \frac{2 * 0,71 * 0,78}{0,71 + 0,78}$$

$$F - 1 = 0,74$$

Pada pengujian performa model yolo kita menggunakan fungsi bawaan yang langsung mendapatkan seluruh nilai indicator performa beserta nilai mean average precision (mAP) dari model yang telah dibuat.

Hasil dari rangkuman indikator performa model dapat dilihat pada gambar 28:

```

[info] Format: box_label, box_conf, box_cls, box_cls_prob, box_cls_prob, box_cls_prob, box_cls_prob, box_cls_prob, box_cls_prob
Total: 80/80, 5.421
avg_nms_iou = 0.7406
WARNING: Additional namespaces: size = 62, 64 MB
Loading weights from https://github.com/yololab/yolo-v5/blob/master/weights/...
name: 64, trainId: 284, G:Gauging (64x64x64x64x64x64)
From: 16x16x16, 14 layers, 1280x1280x1280
...
calculation: mAP (mean average precision): ...
detector layer: 10 - type: 10
detector layer: 22 - type: 22
280
40 - class: 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000
class_10 = 1, name = Mask, ap = 0.78, (TP = 686, FP = 190)
class_11 = 1, name = NoMask, ap = 0.71, (TP = 686, FN = 285)
class_12 = 2, name = Person, ap = 0.978, (TP = 272, FP = 129)
...
mAP (mean average precision) = 0.74, precision = 0.78, recall = 0.71, F1 score = 0.74
(mAP) (mean average precision) = 0.74, (TP = 686, FP = 190, FN = 285), Average (mAP) = 0.74
Dok threshold = 0.5, used Area-Under-Curve for each unique recall
mean average precision (mAP@0.50) = 0.7406, (TP = 686)
Total: 80/80, 5.421

```

Gambar 24. Hasil pengukuran performa model

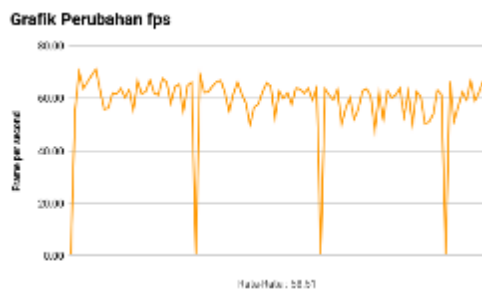
Hasil dari perhitungan mAP menggunakan fungsi darknet menunjukkan hasil yaitu precision model 78% nilai precision menunjukkan seberapa akurat prediksi yang dilakukan oleh model, nilai recall model 71% recall atau sensitivitas merupakan seberapa baik model menemukan nilai deteksi yang benar-benar positif dari seluruh nilai positif, F-1 score 74% merupakan perbandingan rata-rata presisi dan recall yang dibobotkan,serta memiliki mAP sebesar 77,92 %. Nilai pada mAP merupakan rata-rata dari AP tiap kelas objek dari ketiga kelas yang dimiliki model

nilai AP pada kelas nomask yang memiliki nilai tertinggi diikuti kelas mask dan person.

Tabel 3. Hasil Pengujian Deteksi Model

NO	Jarak	Kelas	Gambar Terdeteksi	Rata-rata Confidence Score Model (%)
1	1 m	Masker	20/20	44
		Tanpa Masker	17/20	46
		Orang	40/40	77
2	2 m	Masker	20/20	77
		Tanpa Masker	20/20	72
		Orang	40/40	94
3	3 m	Masker	20/20	55
		Tanpa Masker	20/20	72
		Orang	40/40	89
4	4 m	Masker	20/20	60
		Tanpa Masker	20/20	60
		Orang	40/40	92
5	5 m	Masker	20/20	67
		Tanpa Masker	20/20	79
		Orang	40/40	84

Pada pengujian tabel 4 diberikan data uji orang menggunakan masker dan tanpa masker sebanyak 40 gambar tiap jarak yang diujikan. Dengan pembagian 10 gambar menggunakan masker menghadap ke depan kamera, 10 gambar tanpa masker menghadap ke kamera, 10 gambar menggunakan masker menghadap samping dari kamera, serta 10 gambar tanpa masker menghadap samping dari kamera. Rata-rata confidence score dari hasil deteksi perubahan jarak posisi objek menunjukkan jarak optimal yang menghasilkan deteksi dengan akurasi yang tinggi cenderung berada pada jarak 2 meter.



Gambar 25. Grafik Hasil Perubahan Fps

Hasil pengujian fps dari sistem menunjukkan nilai tertinggi fps yang didapatkan 70,99 fps dengan rata-rata sebesar 58.51 fps, nilai terendah yang didapatkan sebesar 0.77 fps.

## KESIMPULAN

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan pada sistem penghitung pengunjung dan deteksi masker dapat ditarik beberapa kesimpulan sebagai berikut:

1. Metode yang diterapkan pada sistem mampu melakukan deteksi pada tiga kelas objek mask, nomask, dan person menggunakan model YOLO versi 3 tipe tiny pada raspberry pi 4 dengan mAP 77,92%
2. Penghitungan jumlah pengunjung dapat dilakukan pada siang dan malam hari, keberhasilan penghitungan bergantung terhadap performa model dan kemampuan komputasi sistem, dikarenakan proses deteksi dan *tracking* yang memakan sumber daya sistem dilihat dari nilai fps terendah 0,77 fps saat proses deteksi dan tracking.
3. Aplikasi android dan dapat menerima dan menampilkan informasi hasil dari sistem dengan waktu rata-rata yang 2,58 detik untuk menyesuaikan data antara keduanya. Sedangkan notifikasi peringatan diterima dan ditampilkan pada aplikasi android dengan rata-rata waktu 1,76 detik.

## SARAN

Mengingat masih adanya keterbatasan pada penelitian yang dilakukan untuk sistem ini maka diusulkan beberapa saran pengembangan dalam penelitian selanjutnya antara lain yaitu:

1. Menggunakan *single board computer* dengan kartu grafis tambahan untuk meningkatkan kemampuan komputasi sistem dalam melakukan deteksi
2. Memperbanyak data latih untuk meningkatkan akurasi dan performa dari model deteksi yang digunakan

3. Menerapkan model deteksi YOLO tipe normal versi terbaru atau mempertimbangkan algoritma model deteksi lainnya

## REFERENSI

- [1] WHO, "who.int," World Health Organization, 11 03 2020. [Online]. Available: <https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020>. [Accessed 28 06 2020].
- [2] B. Baskara, "Kompas.id," Kompas, 18 04 2020. [Online]. Available: <https://bebas.kompas.id/baca/riset/2020/04/18/rangkaian-peristiwa-pertama-covid-19/>. [Accessed 28 06 2020].
- [3] D. Krisrenanto, M. Rivai and d. F. Budiman, "Identifikasi Jumlah dan Tingkat Aktivitas Orang," *JURNAL TEKNIK ITS*, vol. 6, pp. 2337-3539, 2017.
- [4] D. Y. Setiawan, H. Fitriyah and I. Arwani, "Sistem Penghitung Jumlah Orang Melewati Pintu Menggunakan Metode," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, pp. 2105-2113, 2019.
- [5] M. A. Shereen, S. Khan, A. Kazmi and N. Bashir, "COVID-19 infection: Origin, transmission, and characteristics of human," *Journal of Advanced Research*, vol. 24, pp. 91-98, 2020.
- [6] "stoppneumonia.id," Yayasan Sayangi Tunas Cilik (YSTC), 2020. [Online]. Available: <https://stoppneumonia.id/informasi-tentang-virus-corona-novel-coronavirus/>. [Accessed 28 06 2020].
- [7] Kompas.id, "bebas.kompas.id," Kompas, 18 04 2020. [Online]. Available: <https://bebas.kompas.id/baca/riset/2020/04/18/rangkaian-peristiwa-pertama-covid-19/>. [Accessed 28 07 2020].
- [8] Gugus Tugas Percepatan Penanganan COVID-19, "covid19.go.id/," Gugus Tugas Percepatan Penanganan COVID-19, 28 07 2020. [Online]. Available: <https://covid19.go.id/peta-sebaran>. [Accessed 28 07 2020].
- [9] OpenCV team, "opencv.org," OpenCV team, 2020. [Online]. Available: <https://opencv.org/about/>. [Accessed 28 07 2020].
- [10] R. F. Kevin Rahmat Trisnoyo, "Tabungan Pintar Berbasis Single Board Computer," *CHIPSET (Journal on Computer Hardware, Signal Processing Embedded System and Networking)*, vol. 1, no. Vol. 1 No. 02 (2020): Journal on Computer Hardware, Signal Processing, Embedded System and Networking, pp. 53-60, 2020.
- [11] J. a. F. A. Redmon, "YOLOv3: An Incremental Improvement," *arXiv*, 2018.
- [12] R. F. A. Muhammad Abdul Hadi, "Klasifikasi Tingkat Ancaman Kriminalitas Bersenjata Menggunakan Metode You Only Look Once (YOLO)," *Journal on Computer*

*Hardware, Signal Processing, Embedded System and Networking*, vol. 2, no. Vol. 2 No. 01 (2021): Journal on Computer Hardware, Signal Processing, Embedded System and Networking, pp. 33-40, 2021.

- [13] T. R. P. Foundation, "raspberrypi.org," The Raspberry Pi Foundation, [Online]. Available: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>. [Accessed 25 07 2020].
- [14] Raspberry Foundation, "raspberrypi.org," Raspberry Foundation, 2020. [Online]. Available: <https://www.raspberrypi.org/products/camera-module-v2/>. [Accessed 04 08 2020].
- [15] T. A. Guy, "Youtube," 28 January 2020. [Online]. Available: <https://www.youtube.com/watch?v=10joRJt39Ns>. [Accessed Januari 2021].
- [16] AlexeyAB, "Github," [Online]. Available: <https://github.com/AlexeyAB/darknet#how-to-train-to-detect-your-custom-objects>. [Accessed Januari 2021].