

Pendeteksian Penggunaan Masker Berbasis Android dan YOLOv5 untuk Media Video Realtime pada Ruang Perkantoran

Trisna Gelar¹, Mugi Pangestu², Muhammad Fikri³, Naufal Taufik⁴, Urip Teguh⁵, Jonner Hutahaean⁶

^{1,2,3,4,5,6} Jurusan Teknik Komputer dan Informatika, DIII Teknik Informatika, Politeknik Negeri Bandung

Jl. Gegerkalong Hilir, Ciwaruga, Kec. Parongpong, Kabupaten Bandung Barat, Jawa Barat 40559

Email: trisna.gelar@polban.ac.id, mugipangestu41@gmail.com, naufaltaufik04@gmail.com, mfikri1689@gmail.com, urip@jtk.polban.ac.id

ABSTRAK

Relaksasi penggunaan masker untuk pencegahan penyebaran COVID-19 telah diberlakukan, namun untuk kegiatan dalam ruangan dan wilayah perkantoran kebijakan tersebut masih wajib dilaksanakan. Sistem pengawasan otomatis komersial memerlukan perangkat tambahan seperti kamera khusus, CCTV atau sistem berbasis *Internet of Things* (IoT). Alternatif lain dengan menggunakan perangkat Android yang mudah digunakan dan biaya inisiasi serta pemeliharannya lebih rendah. Maka dari itu pengembangan pendeteksian masker berbasis *Android* perlu dikembangkan. Pada penelitian ini fitur deteksi masker dikembangkan menggunakan algoritma *YOLOv5* pada 1200 gambar campuran dari data *Face Mask Detection* dengan *Face Mask KoTA205*. Dengan menambahkan distribusi kelas *without mask* dan *with mask*, Model *YOLOv5* yang dihasilkan mencapai akurasi *mAP* 91,24% dan *F1-Score* 87,78%. Selain itu model sudah dikonversi berformat *PyTorch Lite* dan dapat digunakan pada perangkat Android untuk mendeteksi masker secara *realtime*. Sistem telah terimplementasi memiliki tiga fitur utama, inisiasi perangkat, pendeteksian masker dan peringatan suara bila terjadi pelanggaran dan *dashboard* untuk bahan *monitoring* (data pendeteksian diolah menggunakan *Firestore*). Khusus fitur pendeteksian masker, telah diuji delapan *test case unit test* dan tiga *test case integration testing*, dengan keseluruhan *test case* berjalan sukses.

Kata kunci: Computer Vision, Dalam Ruangan, Pendeteksian Masker, YOLOv5

ABSTRACT

The necessity for wearing masks to prevent the spread of COVID-19 has been reduced, however it is still in recommended for indoor activities and businesses. Commercial automated surveillance systems require additional equipment, including specialized cameras, CCTV, or Internet of Things-(IoT) systems. Utilizing an Android device, which would be simple to use, has minimal setup and maintenance costs, is an alternative. Therefore, it is essential to construct an Android-based mask detection system. In this study, 1200 mixed images from the Face Mask Detection dataset with Face Mask KoTA205 were being used to develop the mask detection feature using the YOLOv5 algorithm. The resulting model gets an F1-Score of 87.78% and mAP accuracy of 91.24% by adding the class distribution without mask and with mask. The model has been translated to PyTorch Lite format so that it could be used for Mask Detection on Android-based platforms. The deployed system has three primary features: device initiation, mask detection and audible alerts in the event of a violation, and a dashboard for monitoring (detection data is processed by Firestore). For the mask detection feature, 8 test case unit tests and three test case integration tests were conducted successfully, with all test cases passing.

Keywords: Computer Vision, Indoor, Mask Detection, YOLOv5

1. Pendahuluan

Coronavirus Disease 2019 (COVID-19) menyerang sistem pernapasan dan dapat menyebabkan kematian, ditularkan melalui percikan (droplets) batuk atau bersin orang yang terinfeksi (Orang Bergejala) kemudian terhirup oleh orang sehat melalui mata, mulut dan hidung (Amalia, et al, 2020). Salah satu upaya pencegahan virus COVID-19, direkomendasikan menggunakan masker kain atau medis untuk melakukan proteksi pada area hidung dan mulut (Herdiana, 2020). Selain itu penggunaan masker dipahami secara luas oleh masyarakat dan mendapatkan respon positif sebagai bentuk pencegahan penyebaran virus corona untuk diri sendiri (Rachmaniar, 2020).

Kementerian Kesehatan mengeluarkan surat edaran terkait kewajiban penggunaan masker ketika berada di area publik, ditujukan kepada setiap kalangan masyarakat, terutama karyawan perkantoran sebagai

perlindungan diri terhadap COVID-19 (Kementerian Kesehatan, 2020). Perusahaan skala kecil dan menengah melakukan pengecekan penggunaan masker secara manual oleh petugas namun pada waktu-waktu tertentu pengecekan tidak dilakukan. Sehingga diperlukan sistem yang dapat mendeteksi penggunaan masker secara otomatis dan dapat membantu mendisiplinkan pekerja selalu menggunakan masker sesuai standar protokol kesehatan.

Sistem pendeteksi masker telah dikembangkan menggunakan dua pendekatan metode yaitu two stage dan one stage deteksi objek. Pada pendekatan two stage, posisi dan area wajah pada objek dideteksi lebih dahulu menggunakan algoritma ekstraksi fitur Viola Jones/Haar Cascade atau berbasis deep learning Multi- Task Cascaded Convolutional Neural Network (MTCNN), setelah itu dilakukan klasifikasi penggunaan masker menggunakan CNN atau MobileNetv2 (Joshi *et al.*, 2020; Anarki, *et al.*, 2021; Hermawati *et al.*, 2021). Selain memiliki waktu runtime yang tinggi, akurasi pendeteksian berpengaruh terhadap variasi dari pose wajah seperti tampak kiri, tampak kanan atau pose wajah yang memiliki sudut kemiringan tertentu.

Sedangkan pada pendekatan one stage, pendeteksian penggunaan masker dilaksanakan dalam satu tahap sehingga mempercepat waktu deteksi, salah satu algoritma yang digunakan adalah YOLO (You Only Look Once). Implementasi YOLOv4 untuk pendeteksian masker dilakukan pada kamera smartphone dengan objek kerumunan masyarakat pada tiga tempat yaitu Pasar Belawan, Lapangan Merdeka dan Pasar Universitas Sumatera Utara. Terkumpul 37 video dengan resolusi 1280x720 yang dikurasi menjadi 1.326 gambar dengan dua label mask dan non-mask. Akurasi klasifikasi yang dihasilkan mencapai 90.3% dengan kecepatan deteksi 15.44ms (Harahap *et al.*, 2021).

Selain mendeteksi masker, YOLO, telah diimplementasikan pada tiga kasus pemantauan penghitung jarak dan jumlah orang Pertama, Pendeteksian objek manusia berbasis YOLOv3 Tiny dan jarak antar manusia berbasis Euclidean Distance dengan akurasi masing-masing 78.42% dan 87.82% untuk data video diaplikasikan pada Nvidia Jetson (Hasani, *et al.*, 2022). Kedua, penggunaan YOLOv3 untuk lima citra kamera CCTV dengan akurasi rata-rata 90.04% (Indaryanto, *et al.*, 2021).

Ketiga, prototipe sistem pemantauan keramaian pada berbasis YOLOv4, data diambil menggunakan webcam pada suasana perkantoran Kelurahan Tanah Baru, Depok. Sistem akan memberikan peringatan jika terjadi kerumunan pada suatu pos pelayanan. Sistem akan memberikan peringatan melalui email sebagai sarana pengambilan keputusan strategis (Muharram, *et al.*, 2022).

Meskipun aturan penggunaan masker di luar ruangan sedang direlaksasi (Kementerian Kesehatan, 2022). Masyarakat masih diwajibkan menggunakan masker di dalam ruangan perkantoran, sekolah dan universitas. Berbagai solusi/prototipe yang sudah dikembangkan masih menggunakan kamera khusus dan perangkat tambahan seperti CCTV, Komputer, Web Camera dan perangkat berbasis IoT sehingga menambah beban pembiayaan dan pemasangan saat diaplikasikan.

Penelitian ini bertujuan untuk membangun pendeteksian dan monitoring pelanggaran masker yang terdiri dari tiga kondisi: tepat, tidak tepat dan tidak menggunakan masker berbasis YOLOv5 dengan akurasi yang baik pada skenario dalam ruangan perkantoran/universitas serta menggunakan perangkat android, tanpa perangkat tambahan lainnya.

2. Metode Penelitian

Pengembangan sistem pendeteksian dan monitoring pelanggaran penggunaan masker dikembangkan menggunakan model deteksi YOLOv5. Sumber citra berasal dari video kamera perangkat Android dipasangkan menggunakan tripod pada ruang perkantoran, data pelanggaran berbasis Firebase secara realtime akan dikirimkan dan divisualisasikan menggunakan Google Data Studio sebagai bahan pengambilan keputusan. Pendekatan metode penelitian deskriptif digunakan dengan tujuan menjabarkan kasus, mengumpulkan data, membuat deskripsi fakta-fakta yang ditemukan saat melakukan penelitian.

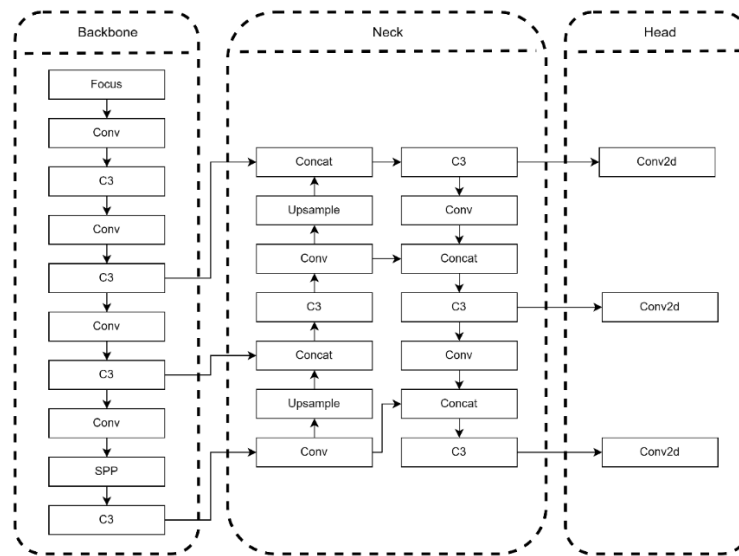
2.1 Landasan Teori

2.1.1 YOLOv5

YOLO adalah arsitektur dan model deteksi objek yang telah dilatih sebelumnya pada data COCO, dan merupakan penelitian open-source yang terus dikembangkan. YOLO merupakan salah satu algoritma paling akurat di bidang computer vision dengan metode deep learning (Bochkovskiy, Wang and Liao, 2020).

Gambar 1 menunjukkan arsitektur dari YOLOv5 yang terdiri dari tiga bagian yaitu Backbone menggunakan arsitektur CSPDarknet digunakan sebagai Spatial Pyramid Pooling (SPP) untuk mengekstraksi feature maps dengan ukuran yang berbeda dari inputan citra. Selanjutnya bagian kedua adalah bagian Neck yang menggunakan arsitektur FPN dan PAN yang memperkuat ekstraksi fitur dan meningkatkan kemampuan deteksi. Bagian terakhir adalah bagian Head yang merupakan lapisan dari YOLO yang digunakan untuk memprediksi target dengan ukuran berbeda pada feature map (Li et al., 2022).

YOLOv5 akan digunakan sebagai teknologi dalam melakukan pendeteksian penggunaan masker. Dipilihnya YOLO sebagai algoritma pendeteksian objek karena dapat memproses input berupa objek penggunaan masker pada wajah dengan jaringan konvolusional dalam sekali proses. YOLO tidak harus melakukan deteksi wajah terlebih dahulu lalu melakukan deteksi masker.



Gambar 1: Arsitektur YOLOv5 diadaptasi dari (Li et al., 2022)

2.1.2 YOLOv5 Mobile Development

PyTorch Lite merupakan framework open-source deep learning yang dapat mempercepat pembuatan prototipe terutama bagian deployment. PyTorch Lite mengkonversi model YOLOv5 yang telah dilatih untuk diterapkan pada platform seluler yaitu Android dan IOS. Semantik model dipertahankan dalam format TorchScript dan Lite Interpreter.

Minimal Android SDK yang dapat digunakan untuk melakukan inferensi model PyTorch Lite adalah SDK 21 [20]. Library yang digunakan untuk membangun sistem pendeteksian masker ini yaitu CameraXii, Dialogiii, Bitmapiv, dan Media Playerv.

2.1.3 Dashboard pada Google Data Studio

Google Data Studio adalah cara yang tepat untuk dijadikan alat yang populer untuk memvisualisasikan data (Hurst, 2020). Fitur Explorer yang berfungsi untuk memanipulasi data menggunakan rumus, filter, dan divisualisasikan sesuai kebutuhan. Kelebihan Google Data Studio yang lainnya yaitu data yang sudah divisualisasikan tersebut dapat dibagikan kepada pengguna lainnya dengan menggunakan fitur data sharing yang bersifat real time (Purnadi, 2021). Google Data Studio digunakan untuk memvisualisasikan data yang disimpan pada database Firebase. Data tersebut nantinya akan diproses terlebih dahulu ke dalam Google Spreadsheet dikarenakan proses pengiriman data ke Google Data Studio tidak dapat langsung dikirim melalui database Firebase.

2.2 Alat dan Bahan

2.2.1 Dataset Face Mask

Terdapat dua sumber data gambar yang digunakan pada penelitian ini yaitu dataset Face Mask Detection (Andrewmvd, 2021) yang terdiri dari 853 gambar dan Dataset Face Mask KoTA 205 (Fikri, 2022) yang terdiri dari 1200 gambar. Kedua dataset memiliki tiga kelas yaitu without mask, with mask dan incorrect mask, sedangkan perbedaan utama dari kedua sumber data tersebut adalah karakteristik perbedaan warna kulit, jenis masker, asesoris wajah (kumis, jenggot, dan kacamata), kepala (topi, hijab dan lain-lain) dari objek orang yang menggunakan masker.

Dataset Face Mask KoTA 205 dikumpulkan karena terdapat ketimpangan distribusi kelas, terutama kelas incorrect mask dan with mask pada dataset primer, serta menambahkan objek orang Indonesia yang khas, karena target sistem akan digunakan secara lokal. Penambahan jumlah kelas without mask dan incorrect mask sebesar 550 gambar dengan format anotasi PASCAL VOC XML. Penambahan kelas bertujuan mengurangi bias klasifikasi pada model YOLOv5 yang akan dibangun.

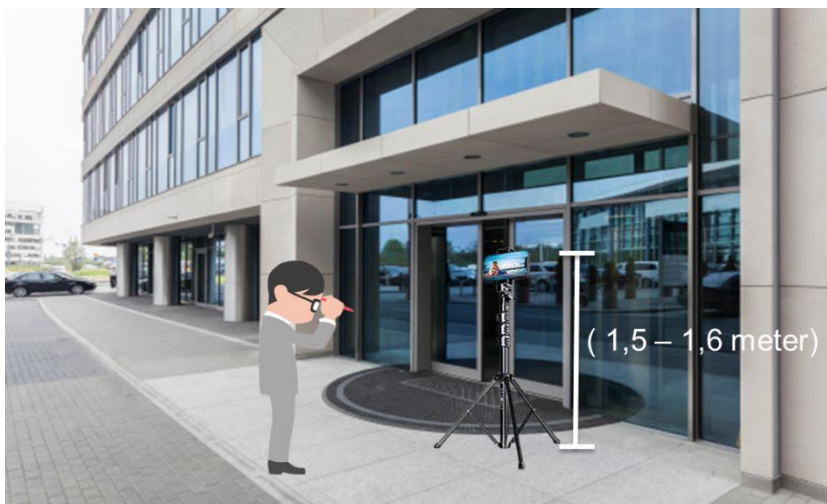
Pada dataset perlu disiapkan dalam set yang sudah dipisahkan untuk training, validation, dan testing. Presentase dari pemisahan data harus sesuai aturan dimana set pelatihan tidak boleh terlalu kecil karena jika terlalu kecil maka model tidak akan memiliki cukup data untuk dipelajari. Di sisi lain, jika set validasi terlalu kecil, maka matriks evaluasi seperti skor mAP dan F1-score akan memiliki varian yang besar dan tidak akan menghasilkan model yang optimal.

2.2.2 Perangkat Keras Pendukung

Google Colaboratory dengan spesifikasi Processor Intel(R) Xeon(R) CPU 2.20GHz, GPU Tesla T4 14G, RAM 13G dan Hardisk 40G. Perangkat Android memiliki kamera minimal kualitas video 480p, memiliki speaker dengan kondisi bekerja dengan baik, dan koneksi internet.

2.3 Perancangan Skenario Sistem

Gambar 2. menunjukkan skenario dari sistem yang akan dibangun. Aplikasi yang sudah dilakukan instalasi pada perangkat android selanjutnya ditempatkan pada tripod dengan tinggi tripod sesuai rata-rata kepala orang asia yaitu sekitar 1,5-1,6 meter. Kemudian diletakkan pada pintu masuk bangunan seperti kantor, tempat ibadah, ataupun tempat lainnya. Dengan itu sistem sudah dapat menjalankan pendeteksian penggunaan masker. Perangkat yang melakukan pendeteksian dibuat lebih dari satu dengan tempat penyimpanan yang berbeda-beda.



Gambar 2: Skenario sistem

Dari sisi aplikasi android, saat sistem mendeteksi seseorang maka sistem akan memberikan peringatan sesuai jika melanggar prokes. Data hasil pendeteksian akan disimpan pada database lokal, jika sudah tertampung seratus data maka akan disimpan ke database online. Data yang tersimpan pada database digunakan untuk visualisasi dalam bentuk grafik. Grafik yang dibuat dapat digunakan sebagai rekomendasi untuk Owner

melakukan tindakan secara langsung. Contoh tindakan yang bisa dilakukan jika grafik menunjukkan banyak orang melanggar di tempat atau di jam tertentu, owner dapat mengambil tindakan untuk menyiapkan orang untuk mengawasi tempat tersebut.

Sistem ini memiliki dua stakeholder yaitu Owner dan Developer. Dengan adanya sistem pendeteksian penggunaan masker dapat membantu owner untuk memberikan peringatan kepada pengunjung yang masuk ke area perkantoran atau yang lainnya. Owner juga akan mendapat data statistik penggunaan masker dari pengunjung yang masuk ke area perkantorannya.

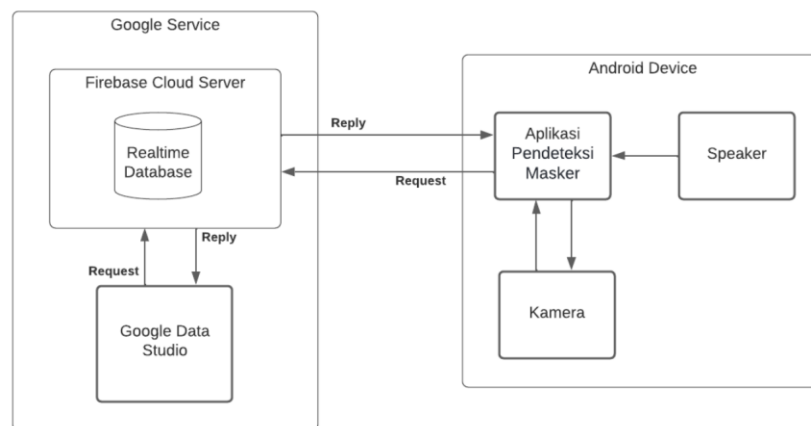
Terdapat empat ruang lingkup dari pengembangan sistem. Pertama, pengoperasian pendeteksian dikhususkan secara perorangan, Kedua, dioperasikan pada pintu masuk ruangan seperti kantor dan masjid, Ketiga, mendeteksi objek wajah dengan minimal jarak sebesar 20 cm dan maksimal jarak sebesar 350 cm sesuai dengan jangkauan ideal dari jarak pandang kamera dan Keempat, hanya berupa agen yang berperan untuk mengirim data untuk digunakan membuat dashboard pendeteksian.

2.4 Perancangan Pengembangan Sistem

Gambar 3 mengilustrasikan arsitektur sistem dan keterhubungannya. Terdapat tiga fitur utama yang akan diterapkan pada sistem ini, sebagai berikut.

2.4.1 Fitur Inisiasi Perangkat

Pembuatan fitur ini diperuntukkan bagi *owner* untuk melakukan pengisian nama dan memilih lokasi dari perangkat yang digunakan. Setup identitas ini hanya dilakukan sekali setelah proses instalasi aplikasi. Pada fitur lokasi ini tidak menggunakan sistem koordinat, dikarenakan jika beberapa perangkat diletakkan pada lokasi yang berjarak kurang lebih 5-10 meter, dan elevasi lantai pada wilayah yang sama maka sistem koordinat ini akan menunjukkan lokasi yang sama, sehingga untuk pengaturan lokasi merujuk pada nama tempat. Dengan melakukan setup identitas perangkat maka perangkat akan terdaftar dan setiap data hasil pendeteksian akan tersimpan di penyimpanan data sesuai perangkatnya.



Gambar 3: Arsitektur sistem

2.4.2 Fitur Pendeteksian

Pada fitur ini dilakukan proses pendeteksian terhadap pengunjung yang ingin memasuki pintu ruangan dengan menggunakan kamera *smartphone*, lalu data hasil pendeteksian tersebut disimpan melalui penyimpanan secara lokal. Setelah mendapatkan hasil pendeteksian, maka sistem akan memberikan suara peringatan yang ditujukan untuk pengunjung sesuai kelas dari hasil pendeteksian tersebut. Jika terdeteksi menggunakan masker akan berupa suara notifikasi selesai, kemudian jika tidak menggunakan masker maka berupa suara peringatan untuk menggunakan masker, dan jika tidak tepat penggunaannya akan berupa suara peringatan untuk menggunakan masker sesuai ketentuan.

2.4.3 Fitur Dashboard

Setelah data hasil pendeteksian tersimpan secara lokal telah tertampung seratus data, selanjutnya dikirimkan ke dalam *database Firebase*. Kemudian data tersebut digunakan untuk visualisasi yang ditampilkan pada dashboard menggunakan Google Data Studio. Pembuatan *dashboard* ini berguna bagi *Owner* bila ingin mengetahui data statistik hasil pendeteksian dan juga sebagai rekomendasi awal untuk memberikan tindakan lebih lanjut.

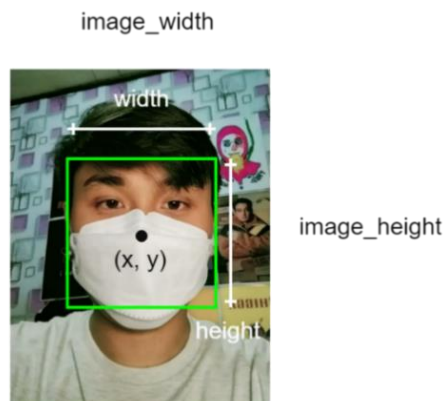
2.5 Perancangan Model Deteksi Masker YOLOv5

2.5.1 Perancangan Alur Kegiatan Pembuatan Dataset

Proses pembuatan dataset KoTA205 dimulai dari pemilihan/kurasi video. Setiap video yang dipilih terdapat objek orang Indonesia, berada pada lingkungan indoor atau outdoor, kondisi latar belakang video terjadi pada pagi, siang, sore, dan malam hari, ramai atau sepi, selain itu terdapat orang yang memakai masker, tidak memakai masker dan memakai masker tetapi penggunaannya belum tepat. Pada tiap video frame yang terdapat objek Orang akan dikumpulkan secara manual menjadi sebuah gambar dan diskalakan dalam ukuran 640x640 pixel.

Proses selanjutnya adalah anotasi data, setiap gambar akan dianotasi secara manual sesuai kondisi penggunaan masker, yaitu without mask, with mask dan incorrect mask dibantu dengan LabelImg dengan format VOC Pascal. Supaya anotasi dapat dibaca oleh YOLO maka dilakukan konversi menjadi format YOLO Darknet.

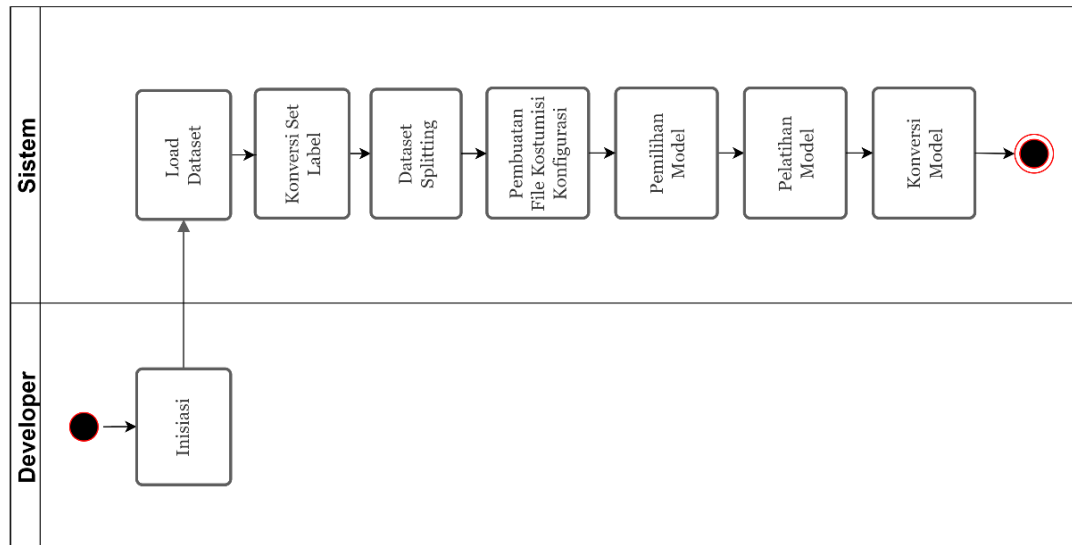
Gambar 4 merupakan ilustrasi anotasi yang terdiri dari nilai x dan nilai y yang merepresentasikan koordinat x dan y yang tepat pada tengah bounding box. Kemudian width merupakan panjang dari bounding box, dan height merupakan lebar dari bounding box. Data anotasi tersebut disimpan pada file dengan format txt dengan berisi anotasi yang sesuai data image, seperti kelas objek, koordinat x, koordinat y, width dan height.



Gambar 4: Anotasi dataset YOLO

2.5.2 Perancangan Alur Kegiatan Pelatihan Model

Pada YOLOv5 terdapat empat jenis model yaitu YOLOv5s, YOLOv5m, YOLOv5l, dan YOLOv5x yang memiliki keunggulan dan kekurangan masing-masing. Semakin besar model seperti YOLOv5l dan YOLOv5x akan menghasilkan model dengan performa yang lebih baik. Akan tetapi memiliki lebih banyak parameter dan memerlukan lebih banyak memori untuk dilatih, selain itu juga akan lebih lambat untuk dijalankan (Ultralytics, 2021).



Gambar 5: Alur aktifitas pelatihan model *YOLOv5*

Gambar 5 merupakan Alur aktivitas pelatihan model. Pertama adalah melakukan *Load Dataset*, yang disimpan pada Google Drive dan dimuat ke *Google Colab* untuk dimanipulasi. Lalu anotasi data dikonversi dari format *XML (Pascal VOC XML)* ke format *TXT (YOLO Darknet TXT)*. Selanjutnya sistem melakukan *Dataset Splitting*, dimana sistem membagi data menjadi *train, validation* dan *test* dengan perbandingan 80:10:10. Setelah itu, dilakukan kostumisasi konfigurasi, meliputi perubahan parameter path dataset, jumlah kelas dan nama kelas, pemilihan pretrained model. Jenis model yang akan digunakan yaitu *YOLOv5s* dikarenakan jenis model ini mendukung penerapan pada perangkat *mobile* dengan inferensi yang lebih singkat daripada jenis lainnya. Selain itu diatur juga *hyperparameter* lain seperti jumlah *epoch, batch size*, dan *image size*. Model hasil pelatihannya kemudian dilakukan konversi ke dalam format *Pytorch Lite (.ptl)*.

3. Hasil dan Pembahasan

3.1 Model Deteksi Masker Yolov5

3.1.1 Konversi Anotasi dan Splitting Dataset

Anotasi data dataset *Face Mask Detection* dikonversi menjadi data anotasi ke bentuk *YOLO Darknet*. Selanjutnya melakukan *splitting dataset* dengan Proporsi 70% data *training*, 15% data *validation*, dan 15% data *testing*.

Tabel 1 merupakan hasil *dataset* yang sudah dilakukan *splitting* yang menunjukkan jumlah data dari setiap *directory* hasil proses *splitting* dataset.

Tabel 1: Jumlah Hasil Splitting Data

Train	Validation	Test
1437	308	308

3.1.2 Konfigurasi File YOLOv5

File konfigurasi berisi jumlah kelas yang digunakan beserta dengan list dari nama kelas secara berurutan. Urutan kelas *pada* list akan menjadi representasi index dari kelas. Selain itu juga berisi pendefinisian *path* ke *directory dataset training, testing* dan juga *validation*.

Tabel 2: Hasil Pembuatan File Konfigurasi

train	/content/images/train
val	/content/images/val
test	/content/images/test

nc names	3 ['without mask', 'incorrect mask', 'with mask']
-------------	--

Tabel 2. menunjukkan contoh hasil dari pembuatan *file* konfigurasi dalam *format file yaml* yang tersimpan dalam *directory* YOLOv5.

3.1.3 Training Custom Dataset ke YOLOv5

YOLOv5 sudah menyediakan abstraksi yang bisa digunakan untuk proses pelatihan sehingga dilakukan beberapa pengaturan parameter. Pada pelatihan model, spesifikasi parameter yang diimplementasikan adalah sebagai berikut.

Tabel 3: Pengaturan Parameter Pelatihan

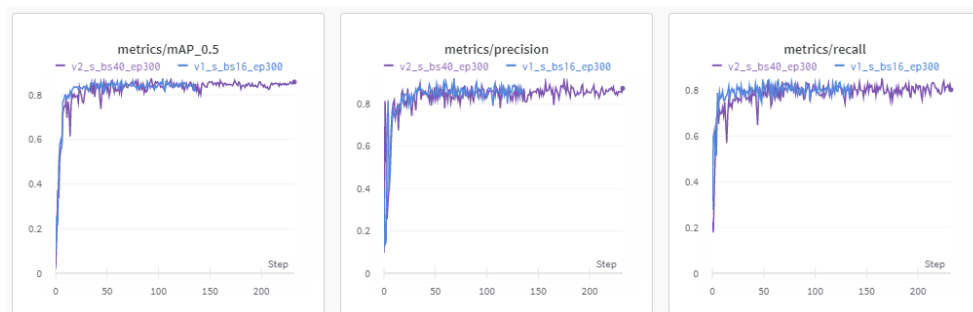
Parameter	Set
<i>weight</i>	<i>yolov5s</i>
<i>Img</i>	640
<i>batch size</i>	40
<i>epochs</i>	300
<i>file configuration</i>	<i>custom_data.yaml</i>

Tabel 3 menunjukkan parameter yang digunakan untuk pelatihan, selain itu juga terdapat parameter lain yang tidak diubah dan menggunakan yang sudah di atur secara default

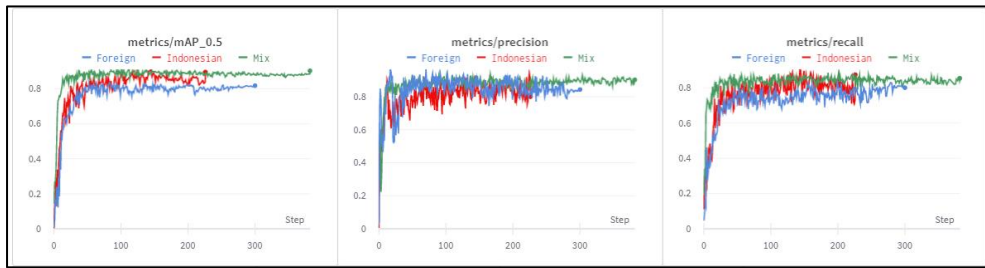
Pada pelatihan ini diatur weight dengan menggunakan pre-trained weight YOLOv5 dengan tipe s, ukuran input image adalah 640, batch size sebesar 16, jumlah epochs sebanyak 300, dan juga file konfigurasi menggunakan file konfigurasi yang sudah dibuat sebelumnya. Nilai parameter yang diset merupakan nilai terakhir atau terbaik yang didapat dari hasil evaluasi.

Gambar 6. menunjukkan perbandingan penggunaan batch size yang ditunjukkan dengan warna biru dengan menggunakan batch size enam belas dan warna ungu dengan menggunakan 40 batch. Pada pelatihan dengan menggunakan 16 batch, model yang dihasilkan memiliki mAP 82%. Sedangkan dengan menggunakan batch 40, model yang dihasilkan memiliki mAP 85%, Jika dibandingkan maka dari nilai matriks tersebut, penggunaan batch yang lebih tinggi menghasilkan model yang lebih baik.

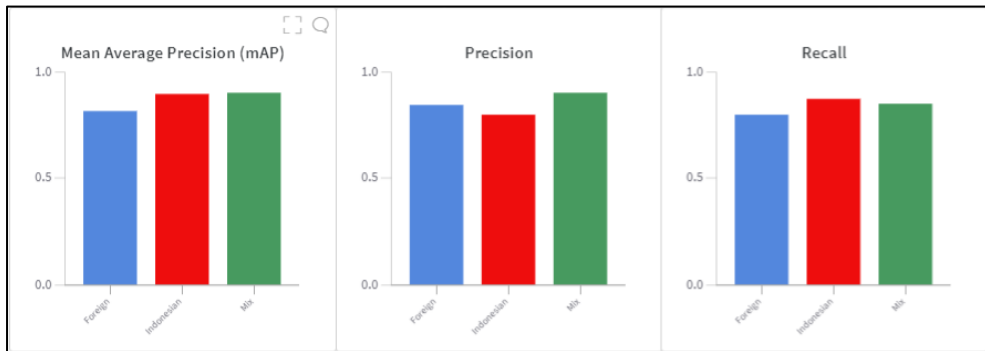
Selanjutnya saat jumlah epoch yang digunakan sebanyak 100 epoch, pelatihan selesai hingga mencapai epoch ke 99. Sedangkan dengan menggunakan 300 epochs, pelatihan terhenti pada kisaran 200 epochs. Hal ini dikarenakan pelatihan diberhentikan karena model tidak memiliki peningkatan performa. Hasil dari pelatihan model ini yang dilakukan dalam tiga skenario berdasarkan dataset yang digunakan yaitu Face Mask Detection, dataset Face Mask KoTA 205, dan *dataset* Campuran.



Gambar 6: Perbandingan Penggunaan Batch Size



Gambar 7: Grafik Proses Nilai Pelatihan Model oleh W&B



Gambar 8: Grafik Hasil Nilai Pelatihan Model oleh W&B

Gambar 7 menunjukkan hasil dari pelatihan model yang menunjukkan beberapa nilai dari kemampuan model. Perhitungan tersebut dihitung dan disimpulkan oleh *YOLOv5* kemudian perhitungan tersebut dikirimkan ke platform *W&B*(wandb, 2019). Gambar 8 menunjukkan hasil pelatihan yang diambil untuk mendapatkan kesimpulan model yang akan digunakan adalah dari nilai *mAP* dan nilai *F1*.

Tabel 4: Hasil Pelatihan Model Tiap Kategori

Kategori Dataset	Epoch Terbaik	Durasi (Menit)	<i>mAP</i>	<i>F1</i>
Face Mask Detection	287	150	81,72 %	82,17%
Face Mask KoTA 205	125	226	89,52 %	83,67%
Campuran	176	582	90,24 %	87,78%

Tabel 4 menunjukkan hasil dari pelatihan model terhadap tiga jenis kategori *dataset* yang dilatih dengan jumlah total pelatihan adalah kurang lebih 16 jam. Hasil terbaik adalah pada skenario dengan kategori *dataset* campuran yang mendapatkan skor *F1* tertinggi dengan nilai 87,78%. Begitu pula dengan skor *mAP* mendapatkan skor paling tinggi dibandingkan yang lain. Dari hasil *training* pada proses sebelumnya, model dengan dataset campuran terpilih untuk dikonversi. Model tersebut dikonversi untuk perangkat *mobile* yaitu dengan melakukan konversi menjadi format *torchscript PyTorch Lite*.

3.2 Pengujian Aplikasi

Pada penelitian ini dilakukan pengujian terhadap fitur pendeteksian masker dengan dua jenis pengujian, diantaranya adalah unit testing dan integration testing.

A. Unit Testing

Tabel 5: Matriks Keterhubungan Unit Testing

Test Function	Result
Fungsi <i>load dataset</i>	Pass
Fungsi konversi <i>dataset</i>	Pass
Fungsi <i>splitting dataset</i>	Pass
Fungsi <i>intersection of union</i>	Pass
Fungsi <i>non-max suppression</i>	Pass

Fungsi <i>image to Bitmap</i>	Pass
Fungsi <i>menghitung jarak deteksi</i>	Pass
Fungsi <i>memvalidasi jumlah data pendeteksian</i>	Pass

Tabel 6 merupakan ringkasan unit testing dengan menggunakan metode *black box* testing yang dilakukan pada pengembangan aplikasi.

B. Integrated Testing

Tabel 6: Matriks Keterhubungan Unit Testing

Test Function	Result
Fitur pendeteksian penggunaan masker	Pass
Fitur mengakhiri pendeteksian penggunaan masker	Pass
Fitur memberikan peringatan	Pass

Tabel 7 merupakan ringkasan matriks keterhubungan integration dengan menggunakan metode black box testing yang dilakukan pada pengembangan aplikasi. Demonstrasi aplikasi dapat dilihat pada tautan <https://www.youtube.com/watch?v=Ums6f0tQ9xg>

4. Simpulan

Terdapat dua simpulan utama pada penelitian yaitu terkait pemodelan deteksi masker berbasis YOLOv5 dan pengembangan sistem pendeteksian masker berbasis Android. Pertama, dengan mencampurkan data karakteristik orang Indonesia dan berfokus pada penambahan kelas *without mask* dan *incorrect mask* yang distribusinya tidak seimbang pada data primer. Dari hasil eksperimen menghasilkan model dengan nilai *mAP* dan *F1-score* terbaik. Kedua, Model deteksi masker dapat di-*deploy* pada perangkat Android dengan baik menggunakan *PyTorch Lite* sistem memiliki tiga fitur utama yang terimplementasi yaitu inisiasi perangkat, deteksi masker yang terdiri dari tiga klasifikasi (*with mask*, *without mask* dan *incorrect mask*), dan fitur *dashboard*. Pengujian secara *unit testing* dan *integration testing* telah dilakukan pada fitur pendeteksian dengan sukses. Adapun saran pengembangan sistem yaitu pengintegrasian dengan sistem deteksi kerumunan berbasis YOLOv5. Selain itu perlu dikembangkannya *controller* yang dapat mengakses sistem secara jarak jauh agar pengoperasian sistem menjadi lebih praktis.

Ucapan Terima Kasih

Kami ucapkan terima kasih atas partisipasi semua pihak atas terlaksananya penelitian dengan baik, dana berasal dari Skema Bantuan Tugas Akhir bersumber DIPA Politeknik Negeri Bandung.

Daftar Pustaka

- Amalia, L., Hiola, F. and Kesehatan Masyarakat, J. (2020) 'Analisis Gejala Klinis Dan Peningkatan Kekebalan Tubuh Untuk Mencegah Penyakit Covid-19', *Jambura Journal of Health Sciences and Research*, 2(2), pp. 71–76. doi: 10.35971/JJHSR.V2I2.6134.
- Anarki, G. A., Auliasari, K. and Orisa, M. (2021) 'Penerapan Metode Haar Cascade Pada Aplikasi Deteksi Masker', *JATI (Jurnal Mahasiswa Teknik Informatika)*, 5(1), pp. 179–186. doi: 10.36040/JATI.V5I1.3214.
- Andrewmvd (2021) *Face Mask Detection | Kaggle*. Available at: <https://www.kaggle.com/datasets/andrewmvd/face-mask-detection> (diakses: Juli 9 2022).
- Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y. M. (2020) 'YOLOv4: Optimal Speed and Accuracy of Object Detection'. Available at: <http://arxiv.org/abs/2004.10934> (diakses: Juni 17 2022).
- Fikri, M. (2022) *Dataset Face Mask KoTA 205 | Kaggle*. Available at: <https://www.kaggle.com/datasets/muhammadfikri16/dataset-face-mask-kota-205> (Accessed: 9 July 2022).
- Harahap, M. *et al.* (2021) 'Identification of Face Mask With YOLOv4 Based on Outdoor Video', *Sinkron: jurnal dan penelitian teknik informatika*, 6(1), pp. 127–134. doi: 10.33395/SINKRON.V6I1.11190.
- Hasani, M. C., Milenasari, F. and Setyawan, N. (2022) 'Pemantauan Physical Distance Pada Area Umum Menggunakan YOLO Tiny V3', *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 6(1), pp. 146–152. doi: 10.29207/RESTI.V6I1.3808.

8. Herdiana, D. (2020) 'Penanggulangan COVID-19 Tingkat Lokal Melalui Kebijakan Adaptasi Kebiasaan Baru (AKB) di Provinsi Jawa Barat', *Journal of Governance Innovation*, 2(2), pp. 131–156. doi: 10.36636/JOGIV.V2I2.442.
9. Hermawati, F. A. and Zai, R. A. (2021) 'Konferensi Nasional Ilmu Komputer (KONIK) 2021 Sistem Deteksi Pemakaian Masker Menggunakan Metode Viola-Jones dan Convolutional Neural Networks (CNN)', pp. 182–187.
10. Hurst, L. (2020) *Hands On With Google Data Studio: A Data Citizen's Survival Guide*. John Wiley and Sons.
11. Indaryanto, F., Nugroho, A. and Suni, A. F. (2021) 'Aplikasi Penghitung Jarak dan Jumlah Orang Berbasis YOLO Sebagai Protokol Kesehatan Covid-19', *Edu Komputika Journal*, 8(1), pp. 31–38. Available at: <https://journal.unnes.ac.id/sju/index.php/edukom/article/view/47837>.
12. Joshi, A. S. *et al.* (2020) 'Deep Learning Framework to Detect Face Masks from Video Footage', in *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*. IEEE, pp. 435–440. doi: 10.1109/CICN49253.2020.9242625.
13. Kementerian Kesehatan (2020) *Surat Edaran HK.02.02/I/385/2020*. Indonesia: Kementerian Kesehatan Republik Indonesia. Available at: <https://covid19.go.id/p/regulasi/surat-edaran-nomor-hk0202i3852020>. (diakses Juli 7, 2022)
14. Kementerian Kesehatan (2022) *Transisi Pandemi ke Endemi: Diperbolehkan Tidak Memakai Masker di Ruang Terbuka – Sehat Negeriku*. Available at: <https://sehatnegeriku.kemkes.go.id/baca/rilis-media/20220517/0739878/transisi-pandemi-ke-endemi-diperbolehkan-tidak-memakai-masker-di-ruang-terbuka/> (diakses: Juli 7 2022).
15. Li, Z. *et al.* (2022) 'A Two-Stage Industrial Defect Detection Framework Based on Improved-YOLOv5 and Optimized-Inception-ResnetV2 Models', *Applied Sciences*, 12(2), p. 834. doi: 10.3390/app12020834.
16. Muharram, A. T., Suhandana, A. A. and Marcheta, N. (2022) 'Rancang Bangun Sistem Deteksi Keramaian Berbasis Internet Of Things dalam mencegah penyebaran Covid-19', *Jurnal Teknologi Informatika dan Komputer*, 8(1), pp. 40–52. Available at: <http://journal.thamrin.ac.id/index.php/jtik/article/view/733>.
17. Purnadi, H. (2021) 'Pemanfaatan Google Spreadsheet Dan Google Data Studio Sebagai Dashboard Suhu Dan Kelembaban Di Laboratorium', *Insan Metrologi PPSDK*, 1(1), pp. 28–33. doi: 10.55101/PPSDK.V1I1.639.
18. Rachmaniar, R. (2020) 'Keberadaan Video Coronavirus di Channel Youtube', *Edsence: Jurnal Pendidikan Multimedia*, 2(1), pp. 11–18. doi: 10.17509/edsence.v2i1.25044.
19. Ultralytics (2021) 'Yolov5 Object Detections'. Available at: <https://github.com/ultralytics/yolov5/>. (diakses Agustus 31 2022)
20. wandb (2019) *Weights and Biases*. Available at: <https://docs.wandb.ai/>. (diakses Agustus. 31, 2022).

