

MONITORING PENGGUNAAN MASKER PADA PENGUNJUNG STMIK “AMIKBANDUNG” MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN)

Fajar Nazmi Fadillah¹, Khoirida Aelani²

¹Program Studi Teknik Informatika, STMIK “AMIKBANDUNG”, Indonesia

²Program Studi Teknik Informatika, STMIK BANDUNG, Indonesia

Email: fajar.nazmif@gmail.com¹, khoirida@stmik-bandung.ac.id²

Abstrak

Sistem monitoring yang digunakan di STMIK “AMIKBANDUNG” saat ini adalah berbasis web tanpa adanya dukungan atau dokumentasi untuk memiliki modul deteksi masker yang berbasis machine learning. Machine learning merupakan cabang ilmu kecerdasan buatan yang meliputi pembangunan sistem yang berdasarkan pada data. Dalam penelitian ini, machine learning diterapkan dalam pengembangan deteksi masker. Hasil akhir dalam penelitian ini adalah sistem deteksi masker yang lebih efektif dan efisien dari segi program, didukung dengan antarmuka dan user experience yang interaktif, dan disertai dokumentasi supaya program dapat dikembangkan lebih lanjut. Dengan memanfaatkan kecerdasan buatan, pihak kampus dapat membuat keputusan lebih mudah berkat dukungan statistik yang menunjukkan jumlah pelanggaran melalui antarmuka situs web.

Kata kunci: COVID-19, Machine Learning, Convolutional Neural Network

Abstract

The monitoring system used at STMIK “AMIKBANDUNG” is currently web-based without any support or documentation for having a machine learning-based mask detection module.. Machine learning itself is a branch of artificial intelligence which includes building systems based on gathered dataset. In this research, machine learning will be studied and applied in the development of mask detection. The final result in this study is a new mask detection system that is more effective and efficient in terms of program, supported by an interactive interface and user experience, accompanied by

documentation so that the program can be developed further. By utilizing artificial intelligence, the campus can make decisions easier with the support of statistics that show the number of violations viewed from the web interface.

Keywords: COVID-19, Machine Learning, Convolutional Neural Network

I. PENDAHULUAN

1.1. Latar Belakang

Penyakit COVID-19 (*coronavirus disease 2019*) merupakan penyakit menular yang disebabkan oleh SARS-CoV-2. Kasus COVID-19 pertama ditemukan di Wuhan, Cina, Desember 2019 lalu. Terdapat beberapa variasi pengenalannya gejala COVID-19, beberapa yang dapat diklasifikasikan adalah demam, batuk, merasa kelelahan, kesulitan bernapas, dan hilangnya kemampuan untuk mencium bau dan rasa. Gejala mulai tampak dari satu hingga empat belas hari setelah terpapar virus [1].

Salah satu sektor yang terdampak dari pandemi global ini adalah pendidikan, dengan pengalihan sistem belajar menjadi daring membuat kampus sepi pengunjung tetapi tidak menutup kemungkinan dalam beberapa kesempatan untuk belajar secara luring belum lagi tamu yang hendak berkunjung ke kampus. Hal ini menimbulkan semua lingkungan harus menerapkan protokol kesehatan yang disarankan oleh WHO dan pemerintah setempat, terutama pemeriksaan suhu yang saat ini sudah ditangani oleh satpam kampus dan pemakaian masker ketika memasuki lingkungan. Namun hingga saat ini belum ada sistem

yang menangani kedua protokol kesehatan ini secara eksklusif. Hal tersebut membuat kampus memiliki kesulitan dalam membuat aturan protokol kesehatan untuk menentukan apakah protokol saat ini sudah baik atau belum dikarenakan kurangnya informasi pengunjung.

Dari permasalahan yang terjadi, maka akan dibuatkan sebuah sistem deteksi masker menggunakan *machine learning* dan *dashboard* berbasis web. Sistem ini dapat menampilkan dan mengklasifikasi ketika pengunjung menggunakan masker, dan kampus dapat mendapat informasi mengenai jumlah pengunjung kampus. Dengan bantuan sistem pelaporan ini, kampus diharapkan terbantu dalam memantau jumlah pengunjung yang memasuki lingkungan kampus STMIK “AMIKBANDUNG”.

1.2. Rumusan Penyelesaian Masalah

Berdasarkan latar belakang permasalahan yang telah diuraikan di atas maka didapatkan rumusan penyelesaian masalah sebagai berikut:

- a. Saat ini belum ada sistem khusus yang dapat melaporkan jumlah pengunjung di STMIK “AMIKBANDUNG” berikut prosentasi validitas penggunaan masker, sehingga perlu dibuat aplikasi pemantauan berbasis web.
- b. Sistem yang dibuat menggunakan algoritma CNN, dengan tampilan antar muka berupa *dashboard* yang mudah dibaca disertai dengan penyimpanan basis data yang tertstruktur.

1.3. Ruang Lingkup

Batasan ruang lingkup penelitian adalah seputar pengembangan deteksi masker di STMIK “AMIKBANDUNG”. Untuk keluaran yang diharapkan adalah sistem klasifikasi masker dan *dashboard* yang bertujuan untuk menjadi sistem pelaporan yang dapat melihat statistik kunjungan STMIK “AMIKBANDUNG” dan dapat menguji 10 (sepuluh) jenis masker yang diklasifikasikan sesuai standar masker WHO dan Kementerian Kesehatan Republik Indonesia disertai perbandingan dari dua kamera berbeda (kamera laptop Dell Inspiron 7567 0,92MP dan kamera Logitech C270 0,9MP).

1.4. Tujuan Penelitian

Berikut merupakan tujuan dari penelitian ini adalah sebagai berikut:

- a. Secara akademis, penelitian ini diharapkan memberi kontribusi ilmiah pada kajian tentang

penggunaan Python sebagai bahasa pemrograman untuk melakukan *machine learning*.

- b. Secara praktis, penelitian ini diharapkan memberi manfaat melalui analisis yang dipaparkan pada pihak-pihak yang berkepentingan dalam analisis data dan *machine learning*.
- c. Memberikan pandangan bagi pembaca maupun pada pihak-pihak yang hendak mencoba Python untuk keperluan *data science* dan *machine learning*.
- d. Membuat sistem pelaporan untuk STMIK “AMIKBANDUNG” untuk memudahkan pihak kampus dalam melacak pengunjung yang masuk ke lingkungan kampus.

II. METODE PENELITIAN DAN PEMBAHASAN HASIL

2.1 Tahapan Penelitian

Metode yang digunakan untuk penelitian ini adalah *Extreme Programming* (XP). Menurut Kent Beck [9], merupakan metode pengembangan perangkat lunak yang cepat, efisien, berisiko rendah, fleksibel, terprediksi, *scientific*, dan menyenangkan. XP juga adalah suatu model yang termasuk dalam pendekatan *agile*, metode pengembangan yang identic dengan memecah pengembangan menjadi beberapa iterasi. Metode pengembangan XP dipilih karena perangkat lunak yang akan dibuat tidak terlalu kompleks dan tergolong berskala kecil dan juga membutuhkan waktu pengembangan yang tidak terlalu lama. Penjelasan dari tiap fase yang terjadi pada metode ini terdiri dari beberapa tahapan, yaitu:

- a. Perancangan (*Planning*)

Langkah awal yang dilakukan yaitu analisa kebutuhan dan pengumpulan data yang akan digunakan dalam pengembangan aplikasi, hal ini meliputi sumber informasi terkait STMIK “AMIKBANDUNG” baik tertulis dari buku, jurnal, atau akses dari situs web maupun yang tidak tertulis seperti hasil dari wawancara dan observasi.

- b. Desain (*Design*)

Tahapan ini diawali dengan membuat perancangan algoritma yang akan digunakan dalam pengembangan, rancangan antarmuka

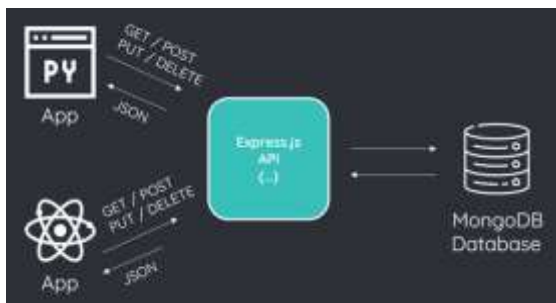
menggunakan Figma dan Adobe Illustrator sebagai media aplikasi *vector*.

c. Pengembangan (*Coding*)

Pada tahap ini pelatihan mesin akan dilakukan supaya dapat memiliki model yang akurat dengan tingkat kegagalan rendah menggunakan bahasa pemrograman Python. Setelah itu hasil pengembangan berupa model akan digunakan untuk menjadi tolak ukur ketika klasifikasi masker dilakukan. Terakhir membuat *dashboard* yang dapat digunakan kampus untuk melihat statistik pengunjung secara interaktif.

d. Pengujian (*Testing*)

Tahap ini dilakukan pengujian menggunakan *unit test* dan metode pengujian yang telah ditentukan sebelumnya. Karena pembuatan *unit test* adalah pendekatan utama dari XP. Dalam melakukan pengujian, penulis menggunakan Teknik pengujian *black box*. Pengujian ini bermaksud untuk melakukan pengujian integrasi antara masukan dan keluaran yang sesuai semestinya terjadi tanpa mengetahui struktur sistem. Untuk gambaran besar bagaimana alur sistem bekerja, perinciannya ditunjukkan pada Gambar 1.



Gambar 1 Alur Kerja Sistem

Terdapat dua modul yang dikembangkan masing-masing menggunakan bahasa pemrograman Python dan Javascript, untuk menyimpan data penelitian ini menggunakan basis data MongoDB. Untuk berinteraksi dengan basis data, penelitian menggunakan perantara API supaya data tidak bisa diakses secara langsung dengan alasan keamanan, fleksibilitas, dan skalabilitas program ketika hendak dikembangkan lebih lanjut. Maka dari itu untuk mengakses data diperlukan interaksi menggunakan *HTTP Request* untuk mengirim permintaan dan *HTTP Response* untuk menerima permintaan. API yang dikembangkan hanya menyajikan keluaran dalam

bentuk JSON (*Javascript Object Notation*) yang lalu diproses oleh masing-masing bahasa.

2.2 Perangkat Yang Digunakan

Kebutuhan perangkat yang diperlukan untuk menunjang penelitian ini terbagi dalam beberapa bagian.

a. Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan aplikasi ini adalah:

1. Sistem operasi Microsoft Windows 10 Versi 21H1 (OS Build 19043.1165)
2. Visual Studio Code dan Jupyter Notebook sebagai editor teks
3. Affinity Designer sebagai perangkat lunak olah citra berbasis *vector*
4. Figma sebagai perancang antarmuka
5. Google Chrome 88.0.4.4324.182 sebagai peramban pengembangan *dashboard*

b. Perangkat Keras

Perangkat keras yang digunakan dalam pembuatan aplikasi ini adalah laptop Dell Inspiron 7567 dengan spesifikasi berikut:

1. CPU Intel® Core™ i7-7700HQ CPU @2.80GHz (4CPUs)
2. GPU Nvidia GeForce GTX 1050Ti, 768 CUDA Core, 4GB GDDR5 128 Bit VRAM
3. Memori RAM 16GB DDR4 2400MHz
4. SSD SATA berkapasitas 240GB
5. Kamera Dell Inspiron 7567 0,92MP dan Kamera Logitech C270 0,9MP

c. Peladen

Ada juga peladen yang dijalankan dengan spesifikasi sebagai berikut:

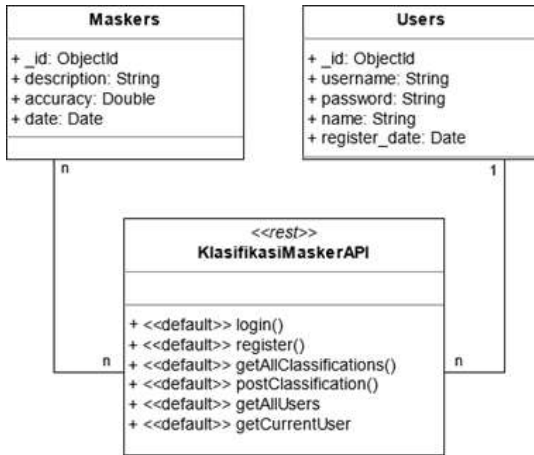
1. Peladen: 127.0.0.1 via TCP/IP
2. Porta: 27017
3. Versi peladen: 4.2.0
4. *Charset* peladen: UTF-8 Unicode (utf8)

2.3 Desain

Tahap ini menjelaskan desain aplikasi mulai dari rancangan algoritma hingga rancangan antarmuka yang nantinya akan digunakan dan dikembangkan pada aplikasi.

2.3.3 Class Diagram

Class Diagram menggambarkan struktur dan deskripsi class, package, dan objek satu sama lain di dalam sistem, dapat dilihat pada Gambar 7.



Gambar 7 Class Diagram Sistem Klasifikasi Masker

2.3.4 Rancangan Antarmuka

Menurut O'Brien, rancangan antarmuka merupakan suatu sistem potensial yang disediakan bagi pengembang dan calon pengguna yang dapat memberikan gambaran kira-kira sistem akan berfungsi bila telah disusun dalam bentuk yang lengkap. Biasanya proses ini dilakukan dengan membuat komponen dasar dari sistem yang akan dikembangkan. Rancangan antarmuka untuk penelitian ini dibangun dengan bantuan Figma dan Affinity Designer. Berikut merupakan rancangan yang telah dibuat (Gambar 8-10).



Gambar 8 Rancangan Aplikasi Klasifikasi Masker



Gambar 9 Rancangan Halaman Login Admin



Gambar 10 Rancangan Halaman Dashboard Admin

2.4 Pengembangan (Coding)

Pada tahap pengembangan akan dilakukan implementasi dari perancangan yang telah dibuat pada tahap sebelumnya. Implementasi dilakukan untuk mengubah model rancangan ke dalam bahasa mesin/kode.

2.4.1 Implementasi API

API digunakan untuk menjadi media komunikasi antar modul, tahapan pengembangan ini ditulis menggunakan Express.js sebagai kerangka kerja Node.js. Untuk basis data menggunakan MongoDB ketika telah melakukan HTTP request yang langsung berkaitan dengan basis data. Semua respon yang diberikan menggunakan JSON, yang alasan penggunaannya atas popularitasnya dibanding XML (Extensible Markup Language) yang memiliki kesamaan dengan semua bahasa pemrograman mengenai bagaimana data terstruktur.

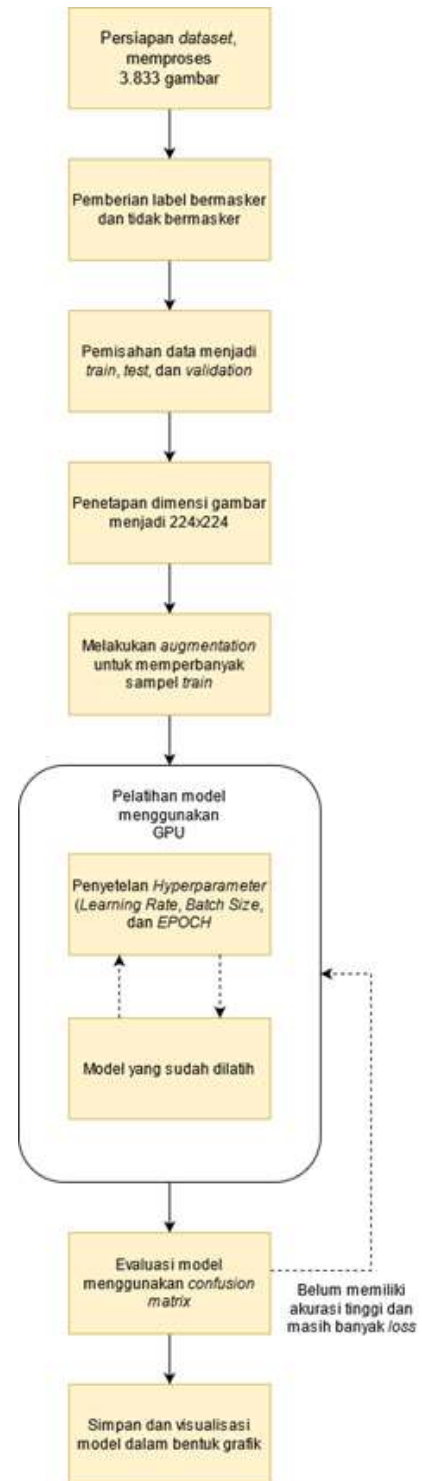
Tabel 1 Dokumentasi API Klasifikasi Masker

Metode	Endpoint	Fungsi	Auth
GET	/api/users	Dapatkan semua pengguna	JWT
GET	/api/masker	Dapatkan semua klasifikasi	JWT
GET	/api/auth/user	Dapatkan pengguna yang saat ini login	JWT
POST	/api/masker	Tambah klasifikasi baru	JWT
POST	/api/auth/register	Tambah pengguna baru (registrasi)	-
POST	/api/auth/login	Login	-

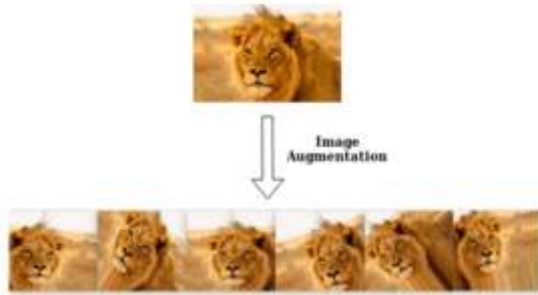
Untuk menambah keamanan dan otorisasi, penulis menggunakan JWT (*Javascript Web Token*). Informasi ini disimpan menggunakan fitur *cookies* yang ada pada peramban. Token dimasukkan ke dalam *heade HTTP request* dengan *key* “x-auth-token”. Selain meringankan beban kerja peladen, dengan menggunakan *stateless* jika pengembangan dari perangkat lunak ini akan dilanjutkan skalabilitasnya juga tidak terlalu kompleks. Data dokumentasi API klasifikasi masker ada pada Tabel 1.

2.4.2 Implementasi Sistem Klasifikasi Masker

Alur kerja pelatihan dan pembuatan model bermasker seperti Gambar 11.



Gambar 11 Alur Kerja Pelatihan & Pembuatan Model

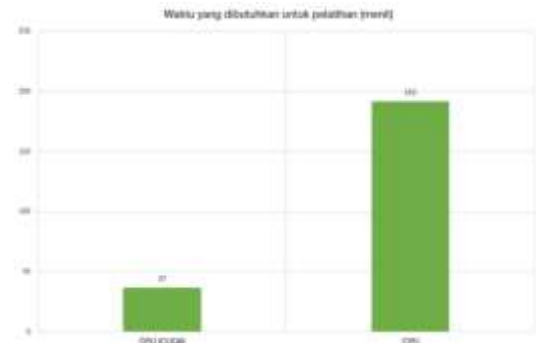


Gambar 12 Proses Transformasi Gambar

Transformasi yang dimaksud adalah memanipulasi gambar ke berbagai bentuk dan mengubahnya menjadi *tensor*, fitur ini disediakan pustaka *tensorflow.keras*. Beberapa manipulasi dilakukan adalah memutar secara horizontal maupun vertikal, memutar gambar (*rotation*), memperbesar gambar (*zoom*), dan menggeser posisi gambar (*shear*) seperti terlihat pada Gambar 16. Dengan memanipulasi keberbagai bentuk, maka dalam penelitian ini dapat memberikan data lebih banyak yang dapat mesin pelajari demi mencapai tingkat akurasi tinggi dengan tingkat *loss* rendah.

Di tahap selanjutnya setelah gambar dimanipulasi, maka data sudah siap digunakan untuk dilatih dan dikenali oleh mesin. Proses yang harus dilewati adalah meng-*compile* kanvas yang sudah dibuat. Proses kompilasinya menggunakan algoritma Adam yang sudah disediakan yang sebelumnya telah diinisialisasi yaitu *learning rate / epoch*. Setelah *model* di-*compile* baru dapat memasuki proses pelatihan, untuk memulainya menggunakan fungsi *model fit* lalu dengan memberi *parameter* tambahan untuk menyesuaikan *weight*, data validasi, dan *epoch*.

Untuk memaksimalkan *training*, algoritma Adam digunakan karena metode perhitungan *learning rate*-nya yang adaptif terhadap parameter berbeda di setiap *batch size* menghasilkan model dapat dilatih lebih mudah diterapkan dengan persyaratan memori yang lebih kecil.



Gambar 13 Rentang Waktu Pelatihan Menggunakan CPU dan GPU

Dalam penelitian ini proses pelatihan *model* menggunakan spesifikasi perangkat yang telah ditulis sebelumnya dan memanfaatkan teknologi Nvidia CUDA. Dengan kartu gratis yang lebih kuat dan telah didukung oleh CUDA, terdapat 768 *core* yang membantu proses pelatihan lebih cepat memakan waktu rata-rata 111 detik per *epoch*. Proses pelatihan juga dapat menggunakan CPU tetapi dikarenakan CPU memiliki *core* yang lebih sedikit dibandingkan *core* yang dimiliki kartu grafis dalam penelitian ini maka akan lebih efektif dan efisien jika pelatihan dilakukan menggunakan kartu grafis. Dengan *epoch* senilai 20, penulis dapat menyimpulkan rentang waktu yang digunakan untuk mendapat model ditunjukkan pada Gambar 13, menunjukkan proses pelatihan yang lebih cepat sebanyak 418% pada pelatihan menggunakan GPU CUDA.

	precision	recall	f1-score	support
bermasker	0.99	0.99	0.99	383
tidak_bermasker	0.99	0.99	0.99	384
accuracy			0.99	767
macro avg	0.99	0.99	0.99	767
weighted avg	0.99	0.99	0.99	767

Gambar 14 Evaluasi Model Klasifikasi

Setelah model dilatih di tahap selanjutnya model harus dipastikan mendapat tingkat prediksi yang cukup baik. Sesuai Gambar 14 di atas untuk evaluasinya penelitian ini menggunakan fungsi *classification_report()* yang merupakan sebuah metode dari pustaka *scikit-learn* yang fungsinya untuk membuat laporan teks yang menunjukkan metrik hasil klasifikasi. Pertama dimulai dari hasil *precision* yang diperoleh dari perhitungan dengan formula sebagai berikut:

$$precision = \frac{true\ positive}{predicted\ yes}$$

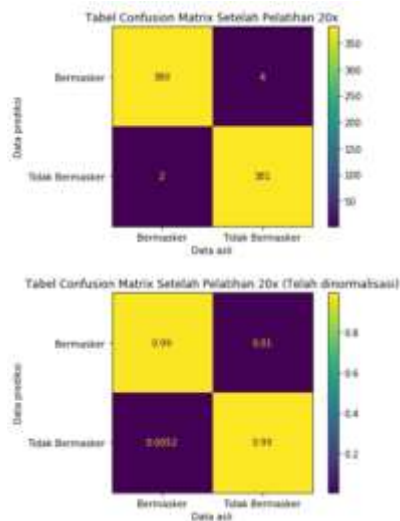
Didapatkan hasil sebesar 0,99 yang dengan menggunakan skala biner (rentang 0 dan 1) merupakan hasil yang tinggi hanya terjadi 6 kesalahan dari 767 data yang diproses berasal dari kedua label. Selanjutnya perhitungan untuk *recall* didapatkan dari hasil perhitungan dengan formula sebagai berikut:

$$recall = \frac{true\ positive}{actual\ yes}$$

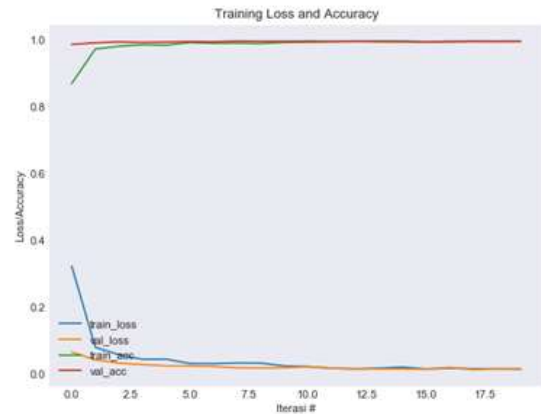
Untuk label bermasker mendapatkan hasil *recall* sebesar 0,99 karena mendapat hasil *true positive* dan *actual yes* dengan nilai sama. Sedangkan untuk label tidak bermasker mendapatkan hasil *recall* sebesar 0,99.

$$f1\ score = \frac{recall}{precision}$$

Untuk penilaian *f1-score* atau *f score*, dihasilkan dari formula yang ditunjukkan di atas dan kedua nilai yang diuji memiliki nilai sama 0,99. Nilai *support* adalah jumlah data yang diuji terhadap masing-masing label, jadi total terdapat 767 data dan terjadi 1 kegagalan di label bermasker sehingga akhirnya dapat disimpulkan dengan model yang sudah dibuat dan dilatih, akan terdapat tingkat kesalahan sebesar 0,26% dan tingkat akurasi 99,74% untuk sistem dapat mengklasifikasikan dengan benar dan sesuai dengan yang diberikan.

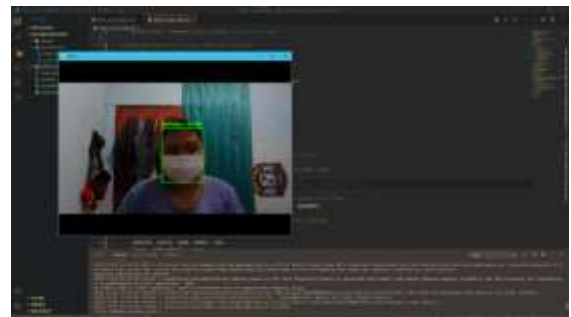


Gambar 15 Tabel Confusion Matrix Model Yang Telah Dilatih

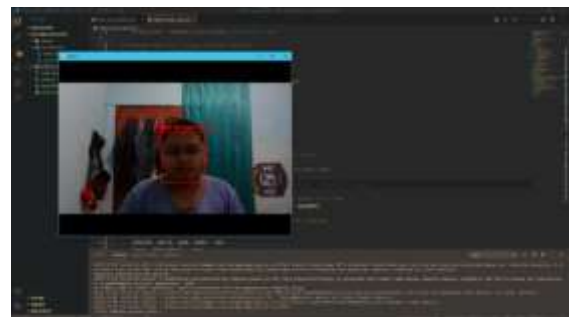


Gambar 16 Hasil Pelatihan Model

Gambar 16 menunjukkan hasil dan spesifikasi model yang telah dilatih sebelumnya. Grafik ini menunjukkan pada iterasi awal model masih memperlihatkan tingkat akurasi yang tidak terlalu tinggi namun seiring pelatihan berlangsung dan berulang maka mesin telah dilatih untuk mengenali pola yang sama berulang kali, hasilnya seiring berjalannya iterasi Latihan menjadi lebih terlatih hingga mendapat tingkat akurasi tertinggi dengan tingkat *loss* rendah untuk memastikan tidak salah ketika klasifikasi sebenarnya dilakukan.



Gambar 17 Hasil Klasifikasi Jika Bermasker



Gambar 18 Hasil Klasifikasi Jika Tidak Bermasker

2.4.3 Implementasi Dashboard



Gambar 19 Implementasi Halaman Login



Gambar 20 Implementasi Halaman Dashboard Utama

Penulisan kode antarmuka *dashboard* seluruhnya menggunakan Javascript dengan kerangka kerja React.js. Javascript di zaman kini dapat digunakan sebagai bahasa penuh yang selain dapat dijalankan di peramban, bisa juga berjalan lewat *desktop* berkat bantuan *Chrome Engine*. React.js digunakan atas fiturnya yang dapat menggunakan komponen kembali yang dinamakan *components*, dengan fitur ini penelitian dapat menghemat waktu penulisan skrip dan meringankan beban program yang dikembangkan. Gambar 19-20 menunjukkan implementasi dashboard.

2.5 Pengujian (Testing)

Pengujian sistem adalah aktivitas yang sudah direncanakan secara sistematis bertujuan untuk menguji fungsionalitas sistem sesuai kriteria yang ditentukan. Pengujian yang dilakukan pada klasifikasi masker yang standarnya bersumber dari WHO untuk menguji mana masker yang baik dan layak digunakan [10].

Pengujian dilakukan menggunakan kamera dari laptop Dell Inspiron 7567 (0,92MP) dan kamera Logitech C270 (0,9MP) dengan intensitas cahaya yang beragam supaya dapat memaksimalkan kemampuan sistem dalam mengklasifikasikan objek dan masker yang akan dipindai. Dari pengamatan yang dilakukan, didapat hasil pengujian intensitas cahaya sebagai berikut (Tabel 2-4) :

Tabel 2 Pengujian Klasifikasi Terhadap Intensitas Cahaya

Intensitas Cahaya	Jarak Maksimal	Tingkat Akurasi
Masker Disposable Non-Medis		
Dalam ruangan (minim cahaya)	0,82 meter	98,11%
Dalam ruangan (cahaya cukup)	1,35 meter	99,52%
Luar ruangan	1,80 meter	98,52%

Pengujian dilakukan untuk mengetahui apakah sistem deteksi dapat ditaruh di berbagai tempat dengan intensitas cahaya yang beragam dan mendapat tingkat prediksi 90 – 100% dapat mengenali pengunjung yang bermasker dan tidak bermasker dengan melakukan 10 kali pengujian menggunakan jenis masker *Disposable Non-Medis* di jarak 1 meter di dalam ruangan dengan pencahayaan baik. Ketika telah melewati jarak maksimal maka program hingga gagal mendeteksi adanya wajah dalam jangkauan kamera.

Tabel 3 Pengujian Klasifikasi Berdasarkan Jenis Masker Di Kamer Dell Inspiron 7567

Jenis Perlindungan	Jenis Masker	Tingkat Akurasi
Jarak 1 meter – Intensitas cahaya cukup di dalam ruangan		
1 Ply civil protective mask	1 ply paper face mask	98,92%
2 Ply civil protective mask	2 ply paper face mask	98,76%
3 Ply civil protective mask	Masker 3 ply (Berwarna)	98,85%
Type II ASTM LEVEL 1	Surgical green mask	98,47%
Type IIR ASTM LEVEL 2	Surgical green mask	98,63%
TYPE IIR ASTM LEVEL 3	Surgical green mask	98,59%
FFP1	Dust mask	98,12%
FFP2 & N95	Respirator mask	98,74%
FFP3 & N99	Respirator mask	97,92%
N100	Respirator mask	96,34%

Type II ASTM LEVEL 1	Surgical green mask	98,57%
Type IIR ASTM LEVEL 2	Surgical green mask	98,33%
TYPE IIR ASTM LEVEL 3	Surgical green mask	97,85%
FFP1	Dust mask	99,12%
FFP2 & N95	Respirator mask	98,84%
FFP3 & N99	Respirator mask	98,33%
N100	Respirator mask	94,09%

Pengujian dilakukan menggunakan standar yang telah ditetapkan oleh WHO menyesuaikan dengan masker yang paling sering digunakan di Indonesia untuk menutupi standar local dari Kementerian Kesehatan Republik Indonesia.

Perhitungan tingkat akurasi diperoleh dengan melakukan 10 kali pengujian dengan masing-masing masker lalu diambil rata-ratanya menghasilkan formula sebagai berikut:

$$\text{Tingkat akurasi} = \frac{\sum x_i}{n}$$

Di mana:

- x_i merupakan sampel jenis perlindungan masker yang diuji
- n merupakan jumlah percobaan yang dilakukan

Tabel 4 Pengujian Klasifikasi Berdasarkan Jenis Masker Di Kamera Logitech C270

Jenis Perlindungan	Jenis Masker	Tingkat Akurasi
Jarak 1 meter – Intensitas cahaya cukup di dalam ruangan		
1 Ply civil protective mask	1 ply paper face mask	99,12%
2 Ply civil protective mask	2 ply paper face mask	99,16%
3 Ply civil protective mask	Masker 3 ply (Berwarna)	99,03%

III. KESIMPULAN

Berdasarkan hasil pengembangan *monitoring* penggunaan masker pada pengunjung STMik “AMIKBANDUNG” menggunakan algoritma *Convolutional Neural Network* (CNN) berbasis web yang dijelaskan sebelumnya, diperoleh kesimpulan sebagai berikut:

Implementasi klasifikasi mendapat tingkat akurasi rata-rata di atas 90% dari dua kamera berbeda, kamera laptop Dell Inspiron 7567 (0,92MP) dan kamera Logitech C270 (0,9MP). Kualitas kamera mempengaruhi kualitas deteksi dari sistem yang dibangun. Jarak efektif pendeteksian saat ini berada dalam jangkauan 0,82 meter hingga 1,80 meter dengan

pertimbangan intensitas cahaya yang cukup dari pemasangan kamera. Jika melewati jarak efektif, sistem klasifikasi akan menampilkan tingkat akurasi prediksi lebih rendah hingga tidak dapat mendeteksi wajah sama sekali.

REFERENSI

- I. C. o. T. o. Viruses, “Virus Taxonomy: Classification and Nomenclature of Viruses,” 2012.
- B. Budiman, “PENDETEKSIAN PENGGUNAAN MASKER WAJAH DENGAN METODE CONVOLUTIONAL NEURAL NETWORK,” 2021.
- D. Kingma dan J. Ba, “ADam: A Method for Stochastic Optimization,” dalam *International Conference on Learning Representations*, 2014.
- D. Li dan Y. Dong, “Deep Learning: Methods and Applications,” vol. 7, no. 3-4, pp. 197-387, 2014.
- D. M. W. Powers, “Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation,” vol. 2, 2008.
- D. Kuhlman, *A Python Book: Beginning Python, Advanced Python, and Python Exercises*, Platypus Global Media, 2011.
- I. C. Education, “Application Programming Interface (API),” IBM, 19 August 2020. [Online]. Available: <https://www.ibm.com/cloud/learn/api>. [Diakses 11 February 2021].
- K. Beck dan C. Andres, *Extreme Programming Explained: Embrace Change*, Pearson Education, 2004.
- OpenCV, “About OpenCV,” OpenCV, 2021. [Online]. Available: <https://opencv.org/about>. [Diakses 23 February 2021].
- W. H. Organization, “Advice on the use of masks in the context of COVID-19: interim guidance,” 6 April 2020. [Online]. Available: <https://apps.who.int/iris/handle/10665/331693>.