

SIMULASI APLIKASI PENDETEKSI PENGGUNAAN MASKER MUKA DALAM MEMBANTU PENGAWASAN PROGRAM 3M

Arnold Nasir

Prodi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Atma Jaya Makassar
Alamat e-mail:arnold_nasir@outlook.com

ABSTRACT

The Coronavirus has made severe damage to the society around the world. Every aspect of human life was affected. As the result, the government propose the “new normal” policy to be applied as the new standards for regulating people in doing their activities. One of the rules within the new normal policy is each people must wear a face mask when they are in a public space. In order to enforce the rule, the implementation of TensorFlow in a face detection technology becomes one of the solutions, especially in the university environment.

Keywords: *Face detection, face mask, new normal, TensorFlow*

1. PENDAHULUAN

Novel coronavirus atau yang lebih dikenal dengan istilah Virus Corona (COVID-19) telah mengakibatkan kerusakan yang cukup parah bagi masyarakat di seluruh dunia. Sejak kemunculannya di awal tahun 2020, setidaknya 170 negara ekonominya diprediksi akan mengalami resesi [1], sehingga diperlukan upaya-upaya guna mencegah dampak ekonomi yang lebih fatal kedepannya. Salah satu bentuk upaya yang dilakukan adalah dengan menerapkan kebijakan “new normal” yang merupakan standar baru dalam mengatur cara hidup masyarakat pasca merebaknya wabah corona.

Menurut Juru Bicara Penanganan COVID-19, Achmad Yurianto, mengatakan bahwa new normal merupakan adaptasi cara hidup baru masyarakat dimana masyarakat diharapkan menjaga pola hidup bersih dan sehat untuk menunjang produktivitas masyarakat tetap produktif meskipun ancaman virus corona ditengah-tengah masyarakat masih ada [2]. Salah satu pola hidup sehat yang dimaksud sebagai panduan new normal ini adalah penggunaan masker muka saat keluar rumah.

Dengan pemberlakuan new normal tentunya diharapkan penyebaran virus corona dapat ditekan. Namun, pengawasan pemberlakuan new normal ditengah-tengah masyarakat tidaklah mudah, misalnya proses pengawasan terhadap penggunaan masker muka ketika keluar rumah. Menurut Direktur Pusat Studi Psikologi Bencana dari Universitas Surabaya, Listyo Yuwanto,

mengatakan bahwa ada sejumlah faktor yang membuat seseorang mengabaikan menggunakan masker muka, yaitu perilaku sosial, persepsi masyarakat terhadap ancaman penyakit, persepsi terhadap variabel demografi, mementingkan unsur kenyamanan dibandingkan kesehatan, dan ketidaktegasan penerapan protokoler pencegahan [3]. Lebih lanjut dijelaskan bahwa masyarakat Indonesia pada umumnya tidak terbiasa menggunakan masker muka karena merasa kesulitan dalam bernafas, suara yang dikeluarkan ketika berbicara tidak jelas, serta ketidaknyamanan yang diakibatkan oleh tali masker yang dikaitkan pada telinga. Hal ini diperparah dengan ketidaktegasan penerapan protokoler pencegahan, sehingga mengakibatkan jumlah masyarakat yang terinfeksi virus corona meningkat tajam. Oleh karena itu, diperlukan sebuah mekanisme pengawasan yang dapat membantu proses monitoring terhadap penggunaan masker muka di tempat umum, misalnya pada lingkungan universitas.

Salah satu metode yang saat ini marak digunakan adalah dengan menggunakan teknologi computer vision, dimana komputer diajar untuk mengenali berbagai objek secara visual. Dalam mengembangkan teknologi computer vision dengan tingkat akurasi yang tinggi maka diperlukan integrasi dengan teknologi machine learning. Salah satu platform yang umum digunakan adalah TensorFlow.

Dengan mengimplementasikan TensorFlow terhadap teknologi computer

vision diharapkan dapat menghasilkan aplikasi yang dapat membantu proses monitoring penggunaan masker muka pada lingkungan universitas. Selain itu, adanya aplikasi pendeteksi masker muka dapat mencegah penyebaran dan pembentukan klaster virus corona yang baru.

2. TINJAUAN PUSTAKA

2.1 Kajian Penelitian Sejenis

Adapun beberapa kajian penelitian sejenis yang digunakan sebagai acuan dalam kegiatan penelitian ini, diantaranya:

1. Xin-Yi Peng, Jun Cao, dan Fu-Yuan Zhang dengan judul penelitian: “*Masked Face Detection Based on Locally Nonlinear Feature Fusion*” [4]. Penelitian ini berfokus pada proses ekstraksi fitur wajah agar lebih akurat dan efektif dengan cara mengusulkan teknik LNFF-Net (*Locally Nonlinear Feature Fusion-based Network*). Teknik LNFF-Net ini bekerja dengan cara menggunakan dataset FCN yang telah dilatih sebelumnya untuk mengekstraksi data pembelajaran terlebih dahulu, kemudian menggabungkannya dengan fitur visual tradisional, misalnya *saliency map* (sebuah gambar yang memperlihatkan kualitas unik setiap pixelnya).
2. A. Nieto-Rodriguez, M. Mucientes, dan V.M. Brea dengan judul penelitian: “*Mask and maskless face classification system to detect breach protocols in the operating room*” [5]. Penelitian ini berfokus pada proses klasifikasi wajah menjadi dua kategori: wajah menggunakan masker bedah dan wajah tanpa menggunakan masker bedah. Penelitian ini menggunakan dataset LFW dengan 5.000 *positive images* dan 10.000 *negative images* untuk tipe *gray-scale*, sementara untuk tipe berwarna digunakan 4.000 *positive images* dan *negative images*. Penelitian ini berhasil untuk mendeteksi wajah yang berjarak 5 meter dari kamera dengan tingkat akurasi untuk menentukan wajah yang menggunakan masker bedah atau tidak sebesar 95%.

2.2 Metode Deteksi Wajah

Beberapa metode pendekatan deteksi wajah yang diajukan dalam penelitian ini antara lain:

1. *Appearance-Based method*: umumnya dalam metode pencocokan template, template tersebut telah ditentukan sebelumnya oleh para ahli. Namun, pada *Appearance-Based method* template dipelajari dari contoh gambar menggunakan analisis statistik atau pembelajaran mesin untuk mendapatkan karakteristik yang relevan dari sebuah wajah.
2. *Feature-Based method*: fitur invariant dari sebuah wajah digunakan untuk mendeteksi tekstur dan warna kulit. Namun fitur wajah tersebut dapat rusak akibat penerangan yang berlebihan, noise, dan oklusi.
3. *Knowledge-Based method*: metode ini berasal dari pengetahuan terhadap fitur wajah yang telah diketahui. Akan tetapi masalah yang dihadapi dengan metode ini adalah kesulitan dalam menerjemahkan pengetahuan manusia terhadap fitur-fitur wajah kedalam sebuah aturan yang tepat dan mudah dimengerti.

2.3 TensorFlow

TensorFlow merupakan salah satu open-source machine-learning library yang umum digunakan dalam melakukan proses analitik berskala besar dan komputasi machine learning. TensorFlow dikembangkan oleh tim peneliti dan arsitek Google Brain dari Google’s Machine Learning Research Organisation dan bertujuan untuk membantu kegiatan penelitian di bidang machine learning dan deep neural network. Beberapa contoh penggunaan TensorFlow antara lain melakukan partisi digit tulisan tangan, identifikasi gambar, pencocokan kata, natural language processing (NLP), dan sequence-to-sequence method untuk interpretasi mesin.

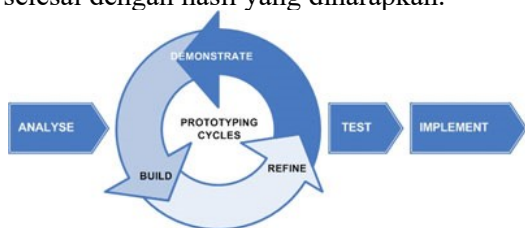
Dalam melakukan proses analitik, TensorFlow menggunakan grafik aliran data yang bertujuan untuk merepresentasikan komputasi, shared state, dan perhitungan operasi yang mempengaruhi keadaan tertentu. Sementara itu, untuk proses komputasi TensorFlow menggabungkan

beberapa model dan algoritma machine learning dan neural networks.

3. METODOLOGI PENELITIAN

Penelitian ini menggunakan *Rapid Application Development* (RAD) dalam mengembangkan aplikasi yang direncanakan. RAD merupakan metode pengembangan software yang lebih cepat dibandingkan metode pengembangan software tradisional karena RAD menerapkan penggunaan teknik yang berulang seperti yang dinyatakan oleh Murch [6]. Selain itu, James Martin berpendapat bahwa RAD tidak hanya membutuhkan waktu yang singkat dalam mengembangkan sebuah software tetapi juga memberikan kualitas end-product yang lebih baik bila dibandingkan metodologi tradisional seperti metode Waterfall.

Menurut Watkins [7], RAD memiliki kerangka waktu yang tetap dalam mengembangkan sebuah aplikasi ataupun sistem, mulai dari 30, 60, hingga 90 hari. Dengan demikian, apabila peneliti ingin mengembangkan suatu software yang ternyata waktu penyelesaiannya melebihi kerangka waktu terlama yang dimiliki RAD maka proyek tersebut dipastikan tidak akan selesai dengan hasil yang diharapkan.



Gambar 1 Bagan Metode RAD

Gambar 1 menunjukkan fase-fase pengembangan yang terdapat dalam RAD, antara lain:

1. Analisa

Pada fase ini peneliti mengumpulkan data yang diperlukan dalam mengembangkan aplikasi yang dirancangkan. Observasi serta studi literatur merupakan instrumen penelitian yang dipilih untuk digunakan dalam mengumpulkan data. Dari kegiatan observasi diperoleh data yang diperlukan dalam membantu proses kegiatan analisa perbandingan kedua metode distance classifier. Setelah data-data telah dikumpulkan, kemudian dilakukan analisa terkait waktu yang dibutuhkan agar penelitian ini dapat diselesaikan sesuai

dengan target waktu yang telah ditentukan.

2. Siklus Prototyping

Pada fase ini, peneliti mulai mengembangkan aplikasi secara bertahap berdasarkan hasil analisa terhadap data-data yang telah dikumpulkan. Proses pengembangan dimulai dengan pemodelan terhadap fitur yang didukung oleh aplikasi yang dirancangkan menggunakan diagram use case maupun diagram aktivitas. Selanjutnya, dengan menggunakan sistem yang telah ada sebagai pedoman dasar dalam merancang aplikasi yang diinginkan, peneliti kemudian mengubah atau menambahkan beberapa fitur baru ke dalam sistem yang akan menghasilkan sebuah prototipe. Dengan menggunakan prototipe sebagai acuan dalam pengembangan aplikasi proses pengembangan dapat berjalan lebih singkat. Hal ini dikarenakan peneliti hanya melakukan perubahan maupun perbaikan pada prototipe sehingga tidak perlu merubah rancangan desain aplikasi secara keseluruhan. Setelah peneliti selesai membangun dan menyempurnakan prototipe, maka selanjutnya prototipe tersebut akan didemonstrasikan. Pada tahapan ini, prototipe akan diuji oleh beberapa pengguna, kemudian pengguna memberikan evaluasi terhadap prototipe, dengan memberikan beberapa masukan berupa perbaikan-perbaikan kecil maupun besar. Namun, jika pengguna berpendapat bahwa prototipe tidak memenuhi ekspektasi calon pengguna, maka peneliti harus mengulang proses mengembangkan prototipe.

3. Pengujian

Setelah prototipe telah memenuhi kriteria yang telah ditetapkan, maka prototipe akan masuk dalam fase pengujian. Pada fase ini, prototipe akan diuji dengan serangkaian uji kelayakan yang telah ditentukan oleh peneliti untuk mengetahui apakah masih terdapat kecacatan yang mungkin tidak nampak pada saat proses pengembangan. Hasil dari setiap uji kelayakan akan dicatat dan dirangkum sehingga dapat digunakan untuk pengembangan prototipe yang serupa dikemudian hari.

4. Implementasi

Fase implementasi merupakan fase terakhir dari proses pengembangan sebuah aplikasi menggunakan metode RAD. Setelah prototipe berhasil melewati fase pengujian maka prototipe telah siap sepenuhnya untuk digunakan atau dioperasikan.

4. HASIL DAN PEMBAHASAN

4.1 Analisa Kebutuhan

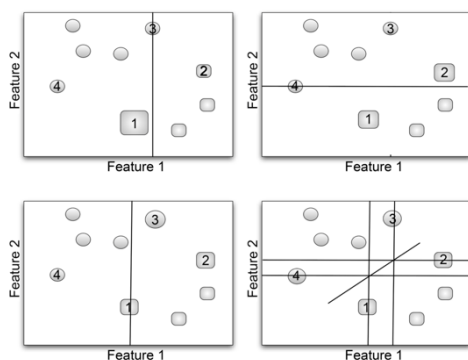
4.1.1 Observasi

Pada kegiatan penelitian ini wilayah umum yang digunakan sebagai studi kasus adalah daerah yang memiliki fasilitas CCTV. Hal ini bertujuan untuk memudahkan pengambilan gambar-gambar yang akan digunakan dalam membentuk dataset. Disamping itu, dengan adanya bantuan dari CCTV akan membantu proses observasi di masa pandemi.

4.1.2 Deteksi Fitur Wajah

Klasifikasi Haar atau sering pula dikenal dengan sebutan Haar-Cascade adalah teknik deteksi wajah yang berbasis pohon di mana dalam tahap pelatihan, dibuatlah gelombang kecil penolakan statistik. Singkatnya, salah satu classifier yang besar dibuat berdasarkan classifier yang lebih lemah. Proses ini dikenal dengan istilah boosting.

Dengan teknik tersebut, classifier yang lebih baik dapat dikembangkan dari beberapa classifier yang menghasilkan output lemah. Selain itu, dengan meningkatkan berat penalti terhadap sampel *misclassified* dapat melatih hipotesis pada iterasi berikutnya sehingga menjadi lebih akurat seperti yang dapat terlihat pada Gambar 2.



Gambar 2 Contoh ilustrasi boosting effect pada Haar-Cascade

Secara umum, ada empat (4) jenis metode boosting yang dapat digunakan secara umum:

1. Real Adaboost
2. Discrete Adaboost
3. Logitboost; dan,
4. Gentle Adaboost

Secara umum, Adaboost sendiri merupakan sebuah algoritma yang dibuat dengan tujuan untuk menciptakan sebuah strong classifier sebagai kombinasi linear yang dapat ditulis dalam persamaan sebagai berikut:

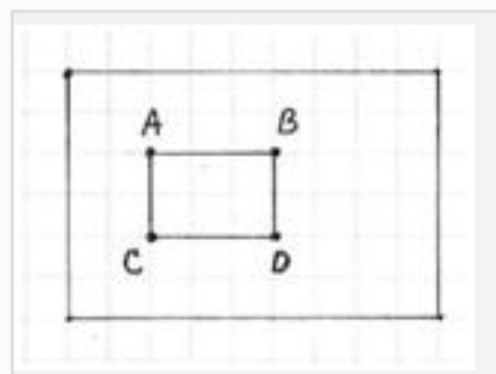
$$f(x) = \sum_{t=1}^T \alpha_t h_t(x)$$

Algoritma Viola-Jones adalah mekanisme yang banyak digunakan untuk deteksi objek. Properti utama dari algoritma ini adalah pelatihan yang lambat, tetapi pendeteksiannya cepat. Algoritma ini menggunakan filter fitur Haar, sehingga tidak menggunakan perkalian.

Efisiensi algoritma Viola-Jones dapat ditingkatkan secara signifikan dengan terlebih dahulu menghasilkan gambar integral yang dalam persamaannya dapat ditulis sebagai berikut:

$$II(y, x) = \sum_{p=0}^y \sum_{q=0}^x Y(p, q)$$

Gambar integral memungkinkan integral untuk Haar extractors dihitung dengan menambahkan hanya empat angka. Sebagai contoh, gambar integral area ABCD (Gambar 2) dihitung sebagai $II(y_A, x_A) - II(y_B, x_B) - II(y_C, x_C) + II(y_D, x_D)$.



Gambar 3 Integrasi area gambar menggunakan gambar integral

Deteksi terjadi di dalam jendela deteksi. Ukuran jendela minimum dan maksimum dipilih, dan untuk setiap ukuran ukuran

langkah geser dipilih. Kemudian jendela deteksi dipindahkan ke seluruh gambar sebagai berikut:

1. Atur ukuran jendela minimum, dan geser langkah yang sesuai dengan ukuran itu. Untuk ukuran jendela yang dipilih, geser jendela secara vertikal dan horizontal dengan langkah yang sama. Pada setiap langkah, satu set filter pengenalan wajah N diterapkan. Jika satu filter memberikan jawaban positif, wajah terdeteksi pada janda saat ini. Jika ukuran jendela adalah ukuran maksimum menghentikan prosedur. Jika tidak, perbesar ukuran jendela dan langkah geser yang sesuai ke ukuran yang dipilih berikutnya dan lanjutkan ke langkah 2.
2. Setiap filter pengenalan wajah (dari set filter N) berisi satu set pengklasifikasi yang terhubung dengan kaskade. Setiap penggolong melihat pada bagian persegi panjang dari jendela deteksi dan menentukan apakah itu tampak seperti wajah. Jika ya, classifier berikutnya diterapkan. Jika semua pengklasifikasi memberi jawaban positif, filter memberikan jawaban positif dan wajah dikenali. Kalau tidak, filter berikutnya di set filter N dijalankan.

Proses pengujian terhadap kedua metode tersebut adalah dengan mengimplementasikan kedalam sebuah sistem pengenalan wajah dari penelitian sebelumnya.

Jumlah training set yang dipakai dalam penelitian ini berjumlah 2,808 citra wajah. Setiap citra memiliki ukuran 180 x 220 dan telah mengalami proses perubahan dari citra berwarna menjadi citra greyscale. Setiap citra akan diubah dari sebuah array menjadi sebuah vector dengan panjang $N = 180 \times 220$ dengan menggabungkan setiap baris dari array gambar.

Setelah persiapan tersebut dilakukan maka selanjutnya peneliti melakukan 5 tahapan proses sebagai berikut:

1. Menghitung dan mempertimbangkan perhitungan citra rata-rata wajah dari set pelatihan.

2. Menghitung vector eigen dan nilai eigen dari matriks kovariansi.
3. Menentukan jumlah eigenface yang saat ini dimiliki.
4. Memproyeksikan setiap gambar dalam pelatihan yang ditetapkan ke dalam ruang wajah.
5. Memperhitungkan satu gambar dalam pengujian yang ditetapkan pada saat itu.

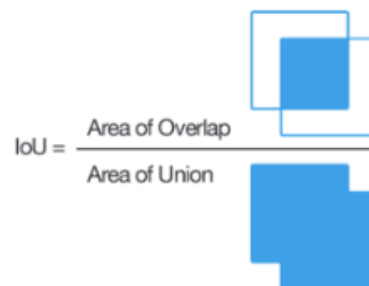
4.1.3 Metode Pendeteksi Wajah

Pada kegiatan penelitian ini yang menjadi fokus utama adalah kemampuan aplikasi dalam mendeteksi wajah manusia, lalu selanjutnya melakukan penyesuaian agar aplikasi mampu menentukan wajah yang menggunakan masker wajah atau tidak. Dalam mengembangkan metode pendeteksi wajah yang sesuai terhadap tujuan penelitian ini, maka peneliti menggunakan metode TFD (Tiny Face Detector).

TFD merupakan metode pengenalan objek wajah dalam sebuah gambar berukuran lebih kecil dari 300x300 px. Metode TFD dapat mendeteksi pola wajah dengan ukuran gambar yang kecil dikarenakan pendekatan yang dilakukan ialah dengan membuat sebuah batas area berbentuk kotak yang berasal dari rumus jarak Jaccard seperti berikut:

$$d(s_i, s_j) = 1 - J(s_i, s_j) \quad (1)$$

$S_i = (h_i, w_i)$ dan $S_j = (h_j, w_j)$ merupakan sepasang bentuk kotak pembatas dan J mewakili kesamaan koefisien Jaccard[8]. Koefisien kesamaan Jaccard atau yang juga dikenal dengan sebutan istilah Indeks Jaccard merupakan statistik yang digunakan dalam memahami kesamaan antara set sampel seperti yang dapat dilihat pada gambar 4.



Gambar 4. Ilustrasi kesamaan koefisien Jaccard atau Indeks Jaccard

Setelah menentukan metode deteksi wajah yang akan digunakan maka selanjutnya peneliti melakukan implementasi TensorFlow.js kedalam prototipe. Pada TensorFlow.js juga telah tersedia library khusus yang menangani masalah verifikasi, pengenalan, dan pengelompokan wajah yang dikenal dengan sebutan FaceNet.

FaceNet merupakan Convolutional Neural Network (CNN) yang dirancang oleh Google dan dilatih untuk menyelesaikan masalah yang berkaitan dengan pendeteksian dan pengenalan wajah dengan skala yang lebih efisien. Dalam pengembangan FaceNet, dataset yang digunakan untuk melatih FaceNet hingga mencapai tingkat akurasi diatas 85% adalah dataset Labeled Faces in The Wild (LFW) dan Youtube Faces DB. Dengan menggabungkan metode TFW dan TensorFlow.js maka kekurangan dari pendeteksi wajah yang sulit dilakukan ketika pengguna melampaui jarak optimal dan menyebabkan wajah pengguna berukuran kecil dapat teratasi. Hal ini disebabkan LFW pada FaceNet dioptimalkan untuk bekerja ketika wajah pengguna masih dapat terdeteksi dengan jelas pada perangkat komputer. Akan tetapi jika pengguna melampaui jarak optimal dari perangkat komputer sehingga terlihat kecil maka TFW yang akan menggantikan posisi LFW dalam mendeteksi wajah pada sebuah gambar atau video.

4.2 Pembentukan dan Pelatihan Dataset

Ada banyak dataset wajah yang dapat digunakan sebagai tolok ukur algoritma deep learning. Berikut adalah beberapa kumpulan data populer untuk pelatihan pengenalan wajah: Wajah Berlabel di Alam Liar: 13.000 gambar wajah dikumpulkan dari web, diberi label dengan nama orang yang digambarkan, Toronto Face Dataset, Olivetti: beberapa gambar dari beberapa orang yang berbeda, Multi-Pie: CMU Multi-PIE Face Database, Face-in-Action, JACFEE: Ekspresi Emosi Wajah Jepang dan Kaukasia, FERET: Database Teknologi Pengenalan Wajah, mmifacedb: MMI, IndianFaceDatabase, dan banyak sumber lainnya.

Untuk keperluan penelitian ini, untuk menangani tugas pengenalan wajah bertopeng, terdapat tiga jenis dataset wajah yang dimasker, yaitu Dataset Deteksi Wajah

Bertopeng (MFDD), Set Dataset Pengenalan Wajah Bertopeng Dunia Nyata (RMFRD) dan Set Data Pengenalan Wajah Bertopeng Tersimulasi. (SMFRD).

Ada juga set data publik yang tersedia, yaitu VGGFace2 adalah set data pengenalan wajah berskala besar. Gambar diunduh dari Penelusuran Gambar Google dan memiliki variasi besar dalam pose, usia, iluminasi, etnis, dan profesi. Dataset berisi 3,31 juta gambar dari 9131 subjek, dengan rata-rata 362,6 gambar untuk setiap subjek [5]. Dataset pengenalan wajah bertopeng harus menyertakan beberapa gambar wajah bertopeng dan tidak bertopeng dari subjek yang sama

Untuk tujuan kesederhanaan, kami memilih kumpulan data 5.000 wajah bertopeng dan 90.000 wajah non-bertopeng. Oleh karena itu, saat membagi dataset menjadi train / validation, kita perlu menjaga proporsi sampel yang sama dalam train dan validasi sebagai keseluruhan dataset. Kami melakukannya dengan menggunakan fungsi `train_test_split` dari `sklearn` dan kami meneruskan label kumpulan data ke parameter `stratify`, dan sisanya akan kami lakukan.

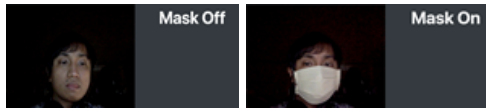
Kami menggunakan dua skenario test split yaitu:

Skenario 1: Non-GPU dengan 10 dan 20 epoch, 70% set data untuk pelatihan dan 30% untuk validasi.

Skenario 2: GPU tunggal dengan 10 dan 20 epoch, 70% dari kumpulan data untuk pelatihan dan 30% untuk validasi.

4.3 Hasil Pengujian

Setelah prototipe melalui proses penyempurnaan maka selanjutnya prototipe akan diuji untuk melihat kemampuannya dalam mendeteksi masker wajah terpasang atau tidak. Proses pengujian dilakukan sebanyak 20 kali dengan mengatur jarak pengguna dengan perangkat komputer sejauh 1.5 – 3 meter. Dari hasil pengujian mendapatkan tingkat akurasi prototipe dalam mengenali wajah yang menggunakan masker wajah diatas 90%. Adapun setelah dilakukan pengujian stress testing, prototipe menggunakan memori RAM sebanyak 261MB.



Gambar 5. Hasil pengujian deteksi wajah yang menggunakan masker wajah dan wajah yang tidak menggunakan masker wajah

5. KESIMPULAN

Dari hasil kegiatan penelitian ini dapat disimpulkan sebagai berikut:

1. Implementasi TensorFlow.js dapat digunakan dalam membantu proses deteksi penggunaan masker wajah.
2. Aplikasi yang dirancang berbasis web sehingga ringan untuk dijalankan pada perangkat komputer dengan spesifikasi yang minim.
3. Pencahayaan tetap memegang peranan penting dalam perhitungan akurasi deteksi wajah. Pada kondisi pencahayaan ruangan yang kurang terang, akurasi deteksi wajah dapat berkurang karena adanya daerah wajah yang tidak terdeteksi dengan baik sehingga menghasilkan deteksi wajah menggunakan masker wajah.
4. Adapun saran terhadap untuk penelitian selanjutnya, penulis berpendapat bahwa penelitian ini dapat diarahkan dengan menggunakan algoritma yang berbeda, seperti penggunaan Linear Discriminant Analysis (LDA) untuk mempercepat proses pengenalan wajah. Disamping itu, penelitian dapat juga diarahkan untuk diintegrasikan dengan berbagai macam sistem informasi.

6. DAFTAR PUSTAKA

- [1] Liputan6. Dampak Corona, Pertumbuhan Ekonomi 170 Negara Diprediksi Negatif. Dapat diakses pada: <https://www.liputan6.com/bisnis/read/4225298/dampak-corona-pertumbuhan-ekonomi-170-negara-diprediksi-negatif>.

- [2] tirto.id. Arti New Normal Indonesia: Tatanan Baru Beradaptasi dengan COVID-19. Dapat diakses pada: <https://tirto.id/arti-new-normal-indonesia-tatanan-baru-beradaptasi-dengan-covid-19-fDB3>.
- [3] baya.co.id. Berkaca Dari Indira Khalista, Mengapa Masih Banyak Masyarakat Yang Ogah Pakai Masker?. Dapat diakses pada : https://www.ubaya.ac.id/2018/content/news_detail/2903/Berkaca-dari-Indira-Khalista--Mengapa-Masih-Banyak-Masyarakat-yang-Ogah-Pakai-Masker-.html
- [4] Ji, Yangfeng, Tong Lin, Hongbin Zha. no date. Mahalanobis Distance Based Non-Negative Sparse Representation for Face Recognition [online]. Dapat diakses pada http://www.cis.pku.edu.cn/faculty/vision/lintong/papers/ICMLA09_Ji.pdf
- [5] A. Nieto-Rodríguez, M. Mucientes, dan V. M. Brea, "Mask and maskless face classification system to detect breach protocols in the operating room", In Proceedings of the 9th International Conference on Distributed Smart Cameras (ICDSC '15), pp. 207–208, 2015.
- [6] Murch, R.2012, Project Management: Best Practices for IT Professionals, Ed.1, Prentice Hall, New Jersey.
- [7] Watkins, J. 2009, Agile Testing: How To Succeed in an Extreme Testing Environment, Cambirdge University Press, Cambridge
- [8] H. Peiyun and Deva R. Finding Tiny Faces. Dapat diakses pada: <https://arxiv.org/pdf/1612.04402v2.pdf>.

