

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi

Pada bab ini, akan dijelaskan langkah-langkah konkret dalam melaksanakan implementasi teknologi pengenalan wajah menggunakan atau memanfaatkan pustaka *face-api.js*. Sebagai catatan, implementasi di sini tidak melibatkan pembuatan model *CNN* secara mandiri. Mengikuti judul, abstrak penelitian dan batasan masalah yang telah disebutkan sebelumnya. Teknik *Convolutional Neural Network (CNN)* hanya sedekedar pemahaman mendasar tentang konsep dan cara mekanisme kerjanya, hal ini dimaksudkan agar pembaca, khususnya penulis, memiliki gambaran umum tentang hal-hal terkait teknik tersebut. Dengan begitu, ketika ingin membuat model sendiri di masa depan, sudah memiliki dasar-dasarnya dan tidak bingung. *Library face-api.js* telah mengimplementasikan model *CNN* yang telah dilatih sebelumnya untuk pendeteksian wajah, fitur wajah (*landmark*), dan pengenalan wajah. Berikut adalah beberapa penjelasan yang akan menjelaskan tentang implementasi *library face-api.js & silent-face-anti-spoofing* lalu penjelasan antar muka dan design sistem yang telah dibuat dan terakhir penjelasan tentang pengujian sistem.

4.2.1. Implementasi *Library face-api.js & silent-face-anti-spoofing*

Dalam pengimplementasian *library face-api.js & silent-face-anti-spoofing* ini, penulis lebih berfokus pada penggunaan yang sederhana daripada memberikan penjelasan rinci tentang proses pembuatan aplikasi yang telah dikembangkan. Hal ini dipilih karena jika penjelasan yang mendalam mengenai aplikasi yang telah dibuat ini, hal ini akan menjadi cukup panjang dan rumit. Sebagai gantinya, pada point ini hanya menjelaskan cara penggunaan *library* tersebut dengan studi kasus yang sederhana.

Selain itu, daripada menjeaskan secara tertulis, penulis memutuskan untuk membuat sebuah video yang memberikan penjelasan yang mungkin

akan lebih mudah dipahami dan mungkin juga akan dapat lebih mendetail. Video tersebut dapat ditemukan di tautan berikut: <https://github.com/mochamaddarmawanh/skripsi/tree/main/video>.

Dengan demikian, penjelasan ini akan lebih fokus pada memberikan panduan praktis mengenai penggunaan *library face-api.js*, daripada membahas proses pembuatan sistem ini yang mungkin saja ada beberapa hal yang tidak penting untuk dijelaskan yang tidak ada hubungannya dengan pembangunan sistem pengenalan wajah nantinya.

4.2.2. Antarmuka Pengguna & Design Sistem

Pada point antar muka dan design sistem penulis terinspirasi dari demo *TensorFlow* pada bagian "*TensorFlow for Web*" atau pada kategori "*Tensorflow.js*". Salah satu contoh demo yang menjadi sumber inspirasi adalah pada point "*Teachable Machine*", yang dapat diakses melalui tautan berikut (<https://www.tensorflow.org/js/demos>). Disana dapat ditemukan informasi lebih lanjut pada bagian yang berjudul "*Teachable Machine*".

Di bawah ini adalah beberapa tampilan yang telah dibuat, yang terinspirasi oleh konsep "*Teachable Machine*" dari *TensorFlow for Web*:

a. Halaman Awal

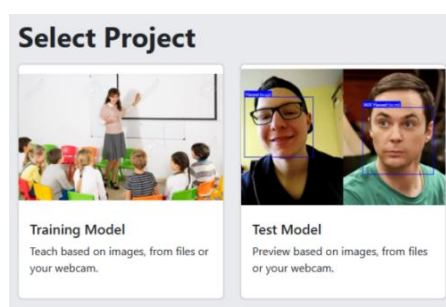


Fig. 26 Gambar Tampilan Awal Sistem

Pada halaman ini pengguna di haruskan memilih 1 dari 2 pilihan, antara men-training (upload wajah) atau pengujian (men-identifikasi atau melakukan pengenalan pada gambar wajah yang di input).

b. Halaman Training

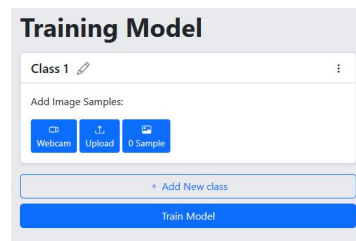


Fig. 27 Halaman Training (Tampilan Awal / Pilihan)

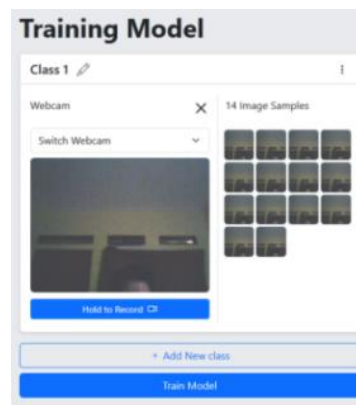


Fig. 28 Tampilan Upload Melalui Webcam

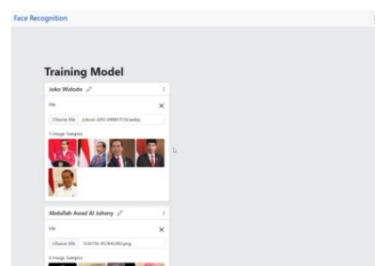


Fig. 29 Tampilan Upload Melalui Pilih File

Pada halaman training, pengguna dapat mengunggah gambar wajah melalui file atau webcam. Selain itu, pengguna dapat memodifikasi nama, mengunduh yang sudah disimpan dalam local storage, dan menghapus contoh yang telah diambil. Setelah semua langkah selesai, pengguna dapat menjalankan pelatihan sistem, dan gambar-gambar tersebut akan diolah untuk diambil hasil komputasinya. Hasilnya kemudian akan disimpan dalam database pada tabel yang diberi nama "face".

- c. Halaman testing (Mengidentifikasi atau Proses Pengenalan Wajah pada Gambar/Citra yang Diberikan)

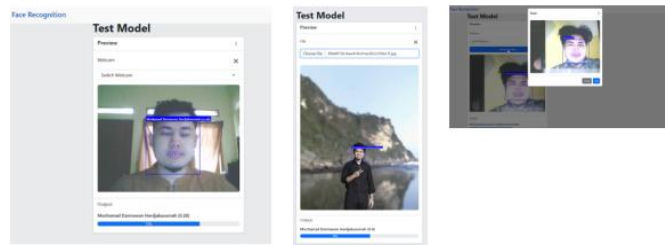


Fig. 30 Pengenalan Melalui Live Webcam atau File, dan Deteksi Spoofing

Pada halaman pengenalan, konsepnya, di mana pengguna dapat memilih metode webcam atau file. Pada halaman pengujian ini, terdapat tombol "Detect Spoofing" yang dapat digunakan untuk mendeteksi serangan spoofing. Meskipun dalam konteks penelitian ini deteksi spoofing belum sepenuhnya real-time, setidaknya langkah-langkah pendeteksian telah diimplementasikan untuk mengatasi tantangan tersebut.

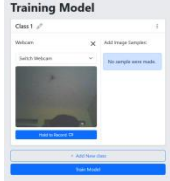
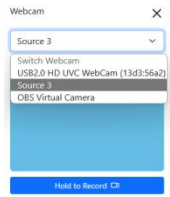
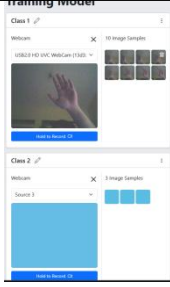
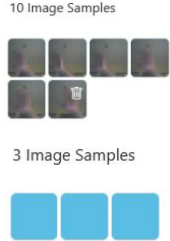

4.2. Rencana Pengujian


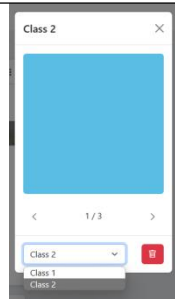
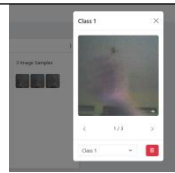
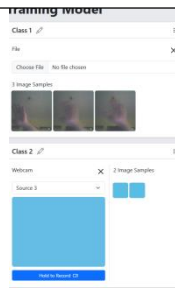

Rencana pengujian adalah konsep pengujian terhadap fungsi-fungsi yang ada di dalam aplikasi yang dibangun, apakah fungsional dari aplikasi berfungsi sesuai yang diharapkan atau tidak. Rencana pengujian yang dilakukan pada aplikasi ini yaitu sebagai berikut:

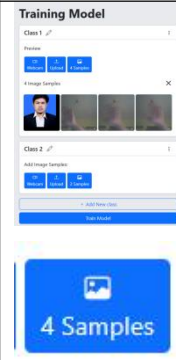

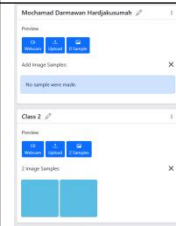
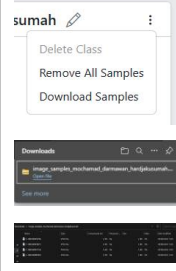

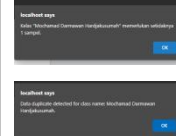
a. Pengujian Fungsionalitas

Pengujian yang dilakukan terhadap fungsionalitas perangkat lunak ini menggunakan metode Black Box. Pengujian ini merupakan pengujian yang berfokus pada persyaratan fungsional perangkat lunak. Tujuan dari pengujian dengan metode Black Box ini adalah untuk menemukan kesalahan fungsi pada perangkat lunak yang telah dibangun. Selain itu, pengujian ini dilakukan dengan mencoba semua kemungkinan yang terjadi dan dilakukan berulang-ulang. Jika dalam pengujian ditemukan kesalahan, maka akan dilakukan penelusuran dan perbaikan untuk memperbaiki kesalahan yang terjadi.

4.3. Pengujian Skenario I Black Box Halaman Training

PENGUJIAN	HASIL YANG DIHARAPKAN	IMAGE	KESIMPULAN
Tombol Webcam	Ketika tombol ditekan, video akan diaktifkan dan daftar webcam akan ditampilkan. Jika tidak ada gambar kelas yang tersimpan, maka munculkan alert bootstrap, tetapi jika terdapat gambar yang telah disimpan, maka tampilkan gambar-gambar tersebut tentunya dengan ID yang sesuai.		Diterima
Switch Webcam	Mengganti ke kamera lain yang tersedia.		Diterima
Hold to Record	Gambar akan ditangkap dan disimpan ke ID imgSample_ sesuai ID kelas yang sebelumnya di lakukan proses cropping.		Diterima
Hapus Gambar	Menghapus satu gambar pada saat klik icon trash ketika di hover.		Diterima
Modal	Diharapkan gambar muncul di modal dengan penomoran sesuai gambar dipilih. Nama kelas juga diharapkan muncul di atas. Pagination untuk pindah gambar juga muncul dengan jumlah		Diterima

	sesuai jumlah gambar di kelas dipilih. Dan terakhir terdapat tombol hapus dengan ikon trash merah.		
Uji pagination	Diharapkan pagination di modal berfungsi dengan seharusnya untuk mengganti gambar atau jika secara sistem memuat ulang modal dengan ID gambar dan ID kelas yang sesuai.		Diterima
Select Kelas	Memuat ID gambar dan ID kelas yang sesuai saat dipilih.		Diterima
Uji Tombol Trash di Modal	Menghapus gambar yang muncul dan secara otomatis me-refresh atau memuat ulang dengan ID di depannya.		Diterima
Tombol Upload Melalui File	Menampilkan form untuk memilih file. Gambar yang disimpan di penyimpanan lokal juga ditampilkan di bawahnya sesuai gambar kelas nya.		Diterima
Upload Melalui File	Ketika onchange gambar yang dipilih dimasukkan ke dalam container di bawahnya yang sebelumnya di lakukan proses cropping.		Diterima

Tombol Lihat Sampel	Diharapkan tombol ini terdapat nomor yang mempresentasikan jumlah dari gambar yang telah disimpan dari kelas. Dan ketika diklik, akan membuka sampel gambar yang telah disimpan di penyimpanan lokal.		Diterima
Ganti Nama Kelas	Diharapkan ganti nama kelas berjalan dengan seharusnya dengan cara melihat di console browser apakah nama kelas yang terganti sudah sesuai ID nya atau masih acak.		Diterima
Hapus Semua Sampel	Diharapkan action ini akan menghapus semua gambar sesuai dengan kelas yang dipilih.		Diterima
Unduh Semua Sampel	Diharapkan pengguna dapat mengunduh semua sampel dalam bentuk ZIP dengan nama ZIP sesuai nama kelas dan di dalam gambar nya dalam format JPG.		Diterima
Tambah Kelas Baru	Diharapkan sistem dapat menambahkan card kelas baru dengan ID yang baru untuk membedakan antara kelas.		Diterima
Tombol Train Model	Menyimpan gambar dari kelas yang dibuat ke dalam tabel komputasi, jumlah gambar ke dalam kolom sum, dan nama		Diterima

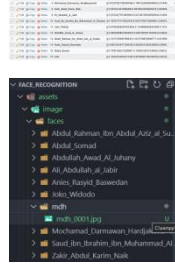


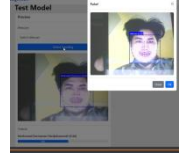

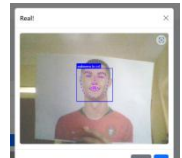
	<p>kelas ke dalam kolom name.</p> <p>Sistem juga membuat folder baru di server dengan nama kelas untuk menyimpan gambar.</p> <p>Sistem akan memeriksa jika ada kelas tanpa gambar atau jika ada nama kelas yang sama di database.</p>		
--	---	---	--

Table 2 Pengujian Halaman Training

4.4. Pengujian Skenario II Black Box Halaman Test

PENGUJIAN	HASIL YANG DIHARAPKAN	IMAGE	KESIMPILAN
Webcam	Muncul kotak, landmark dan nama di webcam.		Diterima
Test Spoofing Dengan Handphone	Ketika di kasih gambar melalui handphone diharapkan sistem akan mendeteksi ini adalah fake.	 	Diterima
Test Spoofing Dengan Kertas Print	Diharapkan semua percobaan berhasil di deteksi ini adalah fake.	 	Tidak Konsisten (penjelasan lebih lanjut dibawah setelah tabel ini)

		<div><div>Falsif</div><div>Falsif</div><div>Falsif</div><div>Falsif</div><div>Falsif</div><div>Falsif</div><div>Falsif</div><div>Falsif</div><div>Falsif</div><div>Falsif</div><div>Falsif</div></div>	
Melalui File	Diharapkan sistem dapat mendeteksi wajah pada gambar yang diberikan atau di upload.	<div><div>Test Model</div><div>Preview</div><div>File</div><div>Choose File</div><div>Download (1.5MB)</div><div>Output</div><div>John Wicaksono (0.95)</div><div>Model Terima: Ben Aduki Aduki (0.95)</div><div>Test Model</div><div>Preview</div><div>File</div><div>Choose File</div><div>Download (1.5MB)</div><div>Output</div><div>Mochamad Darmawan Hardjokusumah (0.95)</div></div>	Diterima

Table 3 Pengujian Halaman Test