

# APLIKASI ARTIFICIAL INTELLIGENCE UNTUK MENDETEKSI OBJEK BERBASIS WEB MENGGUNAKAN *LIBRARY* *TENSORFLOW JS, REACT JS* DAN *COCO DATASET*

Pipit Dewi Arnesia<sup>1</sup>, Naufal Arif Pratama<sup>2</sup>, Fitri Sjafrina<sup>3</sup>

<sup>1,3</sup> Jurusan Sistem Informasi STMIK Jakarta STI&K

Jl. BRI No. 17 Radio Dalam Kebayoran Baru Jakarta Selatan

<sup>2</sup> Jurusan Sistem Informasi Fakultas Ilmu Komputer & Teknologi Informasi Universitas Gunadarma  
Jl. Margonda Raya 100 Pondok Cina Depok

<sup>1</sup>pdarnesia@gmail.com

<sup>2</sup>naufalarif92@gmail.com

<sup>3</sup>fitris596@gmail.com

## Abstrak

Penelitian ini adalah mengenai rancang bangun aplikasi *Artificial Intelligence* untuk mendeteksi objek berbasis web yang didukung oleh *voice command* dengan tujuan memudahkan penggunaannya. Aplikasi ini memanfaatkan model dari *Coco dataset*, *library Tensorflow Js* untuk implementasi *machine learning* pada web, dan *library React Js* dalam pembuatan aplikasi web. Aplikasi ini diujicoba dengan menggunakan metode *black box* untuk menguji fitur yang dimiliki yaitu *voice command* dan *object detection* (deteksi objek). Uji coba dilakukan menggunakan *browser Google Chrome* pada perangkat *smartphone Realme 3*. Aplikasi ini diharapkan dapat membantu orang-orang yang mengalami gangguan pada matanya agar dapat beraktifitas yang lebih nyaman.

**Kata kunci:** *Artificial Intelligence, Machine Learning, Object Detection, Tensorflow Js.*

## I. PENDAHULUAN

Perkembangan teknologi terjadi sangat pesat, dan hal ini tidak dapat dicegah ataupun dihentikan. Inovasi-inovasi baru terus menerus bermunculan. Yang terjadi adalah perkembangan teknologi ini mempengaruhi kehidupan manusia sehari-hari, memudahkan banyak pekerjaan manusia itu sendiri.

Mata merupakan bagian tubuh manusia yang berfungsi untuk menerima cahaya yang kemudian akan direfleksikan menjadi bayangan atau gambar. Mata sendiri sangat sensitif terhadap cahaya, banyak penyakit mata yang dapat mengganggu aktivitas sehari-hari seperti rabun dekat, rabun jauh, katarak, buta warna, dan lain-lain. Beberapa masalah tersebut, seperti rabun dekat dan rabun jauh dapat diselesaikan dengan menggunakan kacamata. Selain itu, bisa juga diselesaikan menggunakan *machine learning* yaitu dengan

memanfaatkan *object detection*, untuk membantu mengatasi permasalahan pada gangguan mata dalam mendeteksi objek.

*Machine Learning* merupakan bagian dari *Artificial Intelligence* yang memiliki banyak cabang, diantaranya *Object Detection* atau *Object Recognize*. Pada *object detection*, kita dapat mendeteksi objek yang terdeteksi pada kamera. Tingkat keakuratan tersebut tergantung pada jumlah data yang dipakai untuk melatih model pada *machine learning*.

*Artificial Intelligence* merupakan kecerdasan buatan yang diciptakan oleh manusia agar dapat mengoperasikan komputer atau suatu sistem. Kecerdasan buatan tersebut dibuat untuk mempermudah pekerjaan manusia. Dari hasil observasi tentang aplikasi *artificial intelligence* yang ada, dapat dijadikan referensi untuk pembuatan aplikasi *artificial intelligence* berbasis web dengan memanfaatkan *Coco dataset* dan *library Tensorflow Js* untuk implementasi *machine learning* pada web, serta dan *library React Js* dalam pembuatan aplikasi web.

Kemudahan serta keunggulan yang diberikan melalui fasilitas *machine learning* memudahkan *developer* dalam pemrograman atau proses pembuatan aplikasi. Aplikasi ini dirancang agar pengguna atau *developer* tidak perlu lagi menuliskan kode, hanya cukup dengan memberikan data yang sudah dilatih.

Penelitian ini adalah mengenai rancang bangun aplikasi *Artificial Intelligence* berbasis web untuk membantu mengatasi permasalahan pada gangguan mata dalam mendeteksi objek. Aplikasi ini diharapkan dapat membantu orang-orang yang mengalami gangguan pada matanya agar dapat beraktifitas yang lebih nyaman.

## II. METODOLOGI PENELITIAN

Metode pada penelitian ini adalah seperti berikut:

### 1. Studi Pustaka

Mencari data dan informasi mengenai materi penelitian ini dari buku, referensi dan literature lainnya.

### 2. Pembuatan Aplikasi

Proses ini melalui tahapan-tahapan sebagai berikut:

- Tahap Perencanaan dilakukan terhadap sistem yang dibuat, seperti studi literatur dan studi lapangan.
- Tahap Analisis dilakukan untuk menganalisis sistem yang akan dibuat, dan kebutuhan yang dibutuhkan untuk pembuatan dan menjalankan aplikasi.
- Tahap Perancangan dilakukan untuk merancang sistem menggunakan struktur navigasi dan UML, serta merancang tampilan untuk aplikasi.
- Tahap Implementasi dilakukan dengan menggunakan bahasa pemrograman *JavaScript* dengan *library React Js* untuk pembuatan web dan *Tensorflow Js* sebagai *library* dari *machine learning* untuk menjalankan proses *machine learning* pada web, model *Coco dataset*, dan pembuatan *voice command*.
- Tahap Uji Coba dilakukan dengan menggunakan metode *black box* dengan memberikan beberapa objek untuk dideteksi melalui *webcam* atau kamera *smartphone* dan perintah melalui *voice command* untuk menguji apakah hasil yang diimplementasikan sesuai dengan rancangan yang telah dibuat.

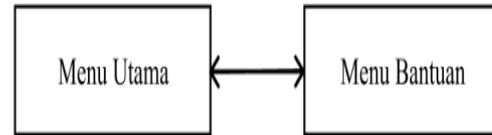
### A. Perancangan Sistem

Tahap perancangan sistem meliputi perancangan struktur navigasi, perancangan *use case diagram*, dan perancangan *activity diagram*.

#### a. Struktur Navigasi

Perancangan aplikasi ini menggunakan struktur navigasi non-linier. Struktur navigasi non-linier merupakan struktur tidak terurut yang merupakan pengembangan dari struktur navigasi linier, namun pada non-linier dianjurkan untuk membuat percabangan. Saat mengakses aplikasi, hal pertama yang akan pengguna lihat adalah halaman utama dan terdapat tombol bantuan yang terletak pada pojok kanan atas halaman.

Terdapat juga menu bantuan yang menjelaskan tentang web yang dibuat, selain itu terdapat jumlah perintah apa saja yang dapat digunakan, panduan cara penggunaan web, serta jumlah objek yang dapat dideteksi. Struktur navigasi tersebut dapat dilihat pada Gambar 1.



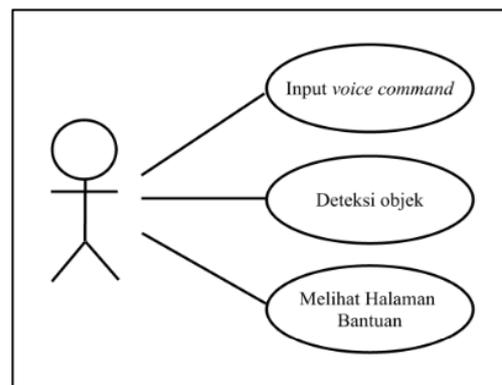
Gambar 1. Struktur Navigasi Aplikasi

#### b. Use Case Diagram

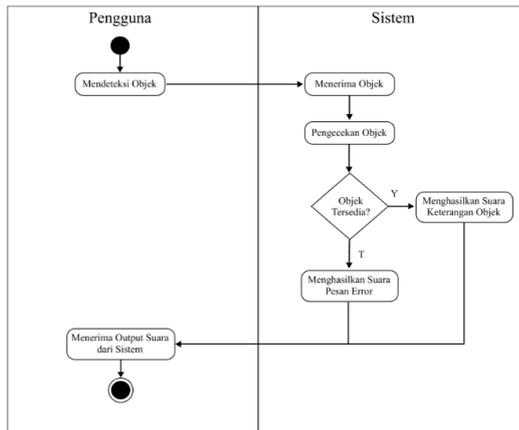
*Use Case Diagram* yang digunakan pada pembuatan aplikasi ini yang menggambarkan bagaimana cara pengguna dapat mengakses aplikasi yang dibuat. Dari *use case diagram* pada Gambar 2. dapat dilihat bahwa pengguna dalam menggunakan aplikasi dapat melakukan 3 hal yaitu melakukan *voice command* dengan perintah yang tersedia, deteksi objek dengan objek yang tersedia pada aplikasi dan melihat halaman bantuan untuk melihat penjelasan dari website, daftar *voice command* dan objek yang tersedia.

#### c. Activity Diagram

Dari gambar 3 terlihat bahwa saat pengguna melakukan deteksi objek dengan menginput objek ke dalam sistem. Kemudian, sistem akan menerima objek tersebut dan mulai melakukan pengecekan apakah objek tersebut tersedia atau tidak. Jika objek tersebut tersedia maka akan menghasilkan suara tentang keterangan objek tersebut. Jika tidak, maka akan menghasilkan suara pesan *error*. Saat sistem telah memberikan output berupa suara, maka pengguna akan menerima hasil output tersebut dari sistem.



Gambar 2. Use Case Diagram Pengguna

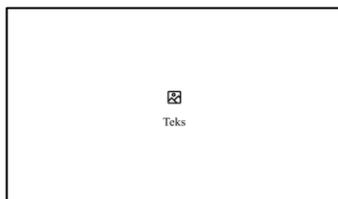


Gambar 3. Activity Diagram Aplikasi

B. Perancangan Tampilan

a. Rancangan Halaman Loading

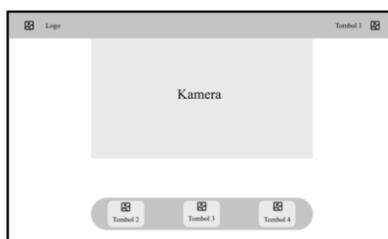
Pada rancangan halaman *loading*, dibuat dengan animasi agar pengguna mengetahui kalau website sedang dimuat dan data sedang diunduh. Gambar rancangan dari halaman *loading* dapat dilihat pada Gambar 4.



Gambar 4. Rancangan Halaman Loading

b. Rancangan Halaman Menu Utama

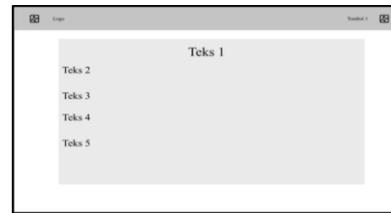
Pada rancangan halaman menu utama, dibuat langsung dengan menampilkan kamera yang dapat langsung digunakan dengan tombol untuk menjalankan *voice command*, menangkap objek, dan memutar mode kamera yang terletak pada bagian bawah *website*. Hal ini ditujukan agar pengguna dapat dengan mudah menekan tombol. Pada bagian atas halaman, terdapat logo di sebelah kiri atas dan tombol menu bantuan di sebelah kanan atas. Hal ini ditujukan agar pengguna dapat langsung melihat tombol bantuan saat pertama kali mengakses *website*. Gambar rancangan dari halaman menu utama dapat dilihat pada Gambar 5.



Gambar 5. Rancangan Halaman Menu Utama

c. Rancangan Halaman Menu Bantuan

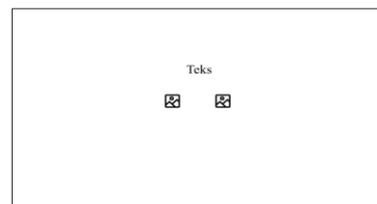
Halaman menu bantuan dirancang dengan terstruktur mulai dari penjelasan tentang aplikasi, cara penggunaan aplikasi, jumlah perintah yang dapat digunakan, dan jumlah objek yang dapat dideteksi. Gambar rancangan halaman menu bantuan dapat dilihat pada Gambar 6.



Gambar 6. Rancangan Halaman Menu Bantuan

d. Rancangan Halaman Error

Halaman *error* dirancang untuk menginformasikan kepada pengguna ketika *browser* atau perangkat yang digunakan tidak mendukung untuk mengakses web yang dibuat. Gambar rancangan halaman *error* dapat dilihat pada gambar 7.



Gambar 7. Rancangan Halaman Error

III. HASIL DAN PEMBAHASAN

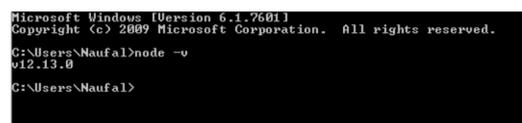
A. Implementasi

Implementasi fungsi pemrograman dilakukan menggunakan bahasa pemrograman JavaScript dengan *library* React Js sebagai alat dalam pembuatan *website* dan Tensorflow Js sebagai *library* dari *machine learning* untuk menjalankan proses *machine learning* pada *website*.

Tahap implementasi dilakukan dengan mengimplementasikan *library* React Js, pembuatan *user interface* pada *website*, implementasi *library* Tensorflow Js, dan pembuatan perintah *voice command* pada *website*.

a. Implementasi Library React Js

React Js adalah *library* untuk membuat *user interface* pada *website*. Untuk mengimplementasikannya dengan cara *install* Node Js terlebih dahulu. Kemudian pilih *command prompt* atau terminal dan jalankan perintah seperti pada Gambar 8. untuk melihat jika Node Js berhasil *install*.



Gambar 8. Pengecekan Node Js

Selanjutnya, jalankan perintah seperti pada Gambar 9 untuk menginstall aplikasi web dengan React js.

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Naufal>npm create-react-app alice
```

Gambar 9. Instalasi React Js

### b. Implementasi Rancangan Tampilan (User InterFace)

Pada tahap ini dilakukan implementasi rancangan tampilan dalam perangkat lunak atau perangkat terkomputerisasi dengan fokus pada tampilan dan sentuhan pada *website* untuk menghasilkan tampilan yang lebih dinamis.

#### 1) Halaman Menu Utama

Tampilan halaman menu utama ditunjukkan pada Gambar 10.



Gambar 10. Tampilan Halaman Menu Utama

#### 2) Halaman Loading

Tampilan halaman *loading* ditunjukkan pada Gambar 11.



Gambar 11. Tampilan Halaman Loading

#### 3) Halaman Menu Bantuan

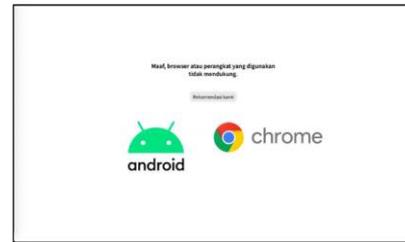
Tampilan halaman menu bantuan, ditunjukkan pada Gambar 12.



Gambar 12. Tampilan Halaman Menu Bantuan

#### 4) Halaman Error

Tampilan halaman *error*, ditunjukkan pada Gambar 13.



Gambar 13. Tampilan Halaman Error

### c. Implementasi Tensorflow Js

*Library* Tensorflow Js dibutuhkan dalam penerapan *machine learning* pada *website* khususnya pada *JavaScript*. Dengan Tensorflow Js, *JavaScript* dapat menjalankan fungsi untuk memproses *machine learning* pada *website*.

Selanjutnya adalah mempersiapkan model untuk digunakan pada *website*. Pada tahapan ini, model harus sudah siap digunakan saat pengguna pertama kali membuka *website* agar tidak terjadi proses pengambilan model pada server setiap kali pengguna melakukan pendeteksian objek.

Dengan memuat model saat pertama kali mengakses *website*, dapat mempercepat proses pendeteksian setiap objek yang ada. Hal ini dikarenakan model telah tersimpan pada *website* dan telah siap digunakan dan dapat mempercepat proses setiap pendeteksian objek. Jika *website* dapat berjalan dengan cepat maka pengalaman pengguna dalam menggunakan *website* akan menjadi lebih baik.

Persiapan data pada penelitian ini memanfaatkan model data dari Coco Ssd. Model Coco Ssd memiliki 80 objek yang dapat dideteksi. Dari model data tersebut dilakukan pengklasifikasian objek menjadi seperti berikut:

1. Model data makhluk hidup memiliki 11 objek.
2. Model data kendaraan memiliki 8 objek.
3. Model data makanan memiliki 10 objek.
4. Model data alat makan memiliki 7 objek.
5. Model data alat elektronik memiliki 6 objek.
6. Model data alat olahraga memiliki 10 objek.
7. Model data peralatan dapur memiliki 5 objek.
8. Model data perabotan memiliki 6 objek.
9. Model data aksesoris memiliki 5 objek.
10. Model data benda di jalan raya memiliki 5 objek.
11. Model data peralatan rumah memiliki 7 objek.

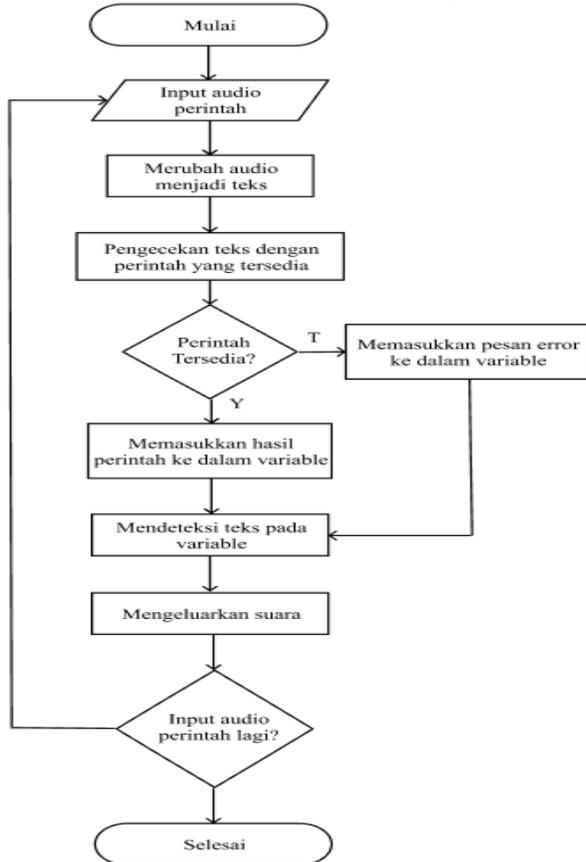
Setelah mengetahui objek yang dapat dideteksi, selanjutnya melakukan penerjemahan data dari bahasa Inggris ke dalam bahasa Indonesia dengan fungsi yang dibuat pada program.

### d. Implementasi Voice Command

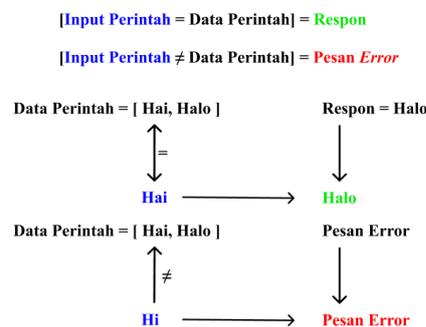
Implementasi *voice command* pada *website* membutuhkan Web API *Speech Recognition* untuk mendeteksi suara agar suara pengguna dapat dideteksi oleh *browser*. Untuk penerapannya pada React Js menggunakan *library* React *speech recognition*. Pada proses pembuatan *voice command*, *browser* akan mendeteksi suara dari pengguna, kemudian

suara tersebut akan diubah menjadi teks. Teks tersebut yang akan menjadi perintah untuk menjalankan fungsi pendeteksi objek atau perintah yang lainnya. *Flowchart* untuk membuat *voice command* dapat dilihat pada Gambar 14.

Data pada *voice command* memiliki tipe data array yang berfungsi untuk menampung semua kemungkinan perintah yang memiliki makna sama untuk menghasilkan hasil perintah yang sesuai dari maksud perintah tersebut. Rumus dari pembuatan data pada *voice command* terlihat pada Gambar 15.



Gambar 14. *Flowchart* Membuat *Voice Command*



Gambar 15. Rumus Pembuatan Data pada *Voice Command*

Dari rumus tersebut dapat dijelaskan bahwa setiap input perintah akan dicocokkan dengan data perintah. Jika pada data

perintah terdapat input perintah maka hasil perintah yang akan dikeluarkan, jika tidak tersedia maka pesan *error* yang akan dikeluarkan. Selanjutnya adalah pembuatan data *voice command*. Terdapat 50 perintah yang dibuat pada program ini.

B. Uji Coba Aplikasi

Aplikasi akan diuji coba menggunakan metode *black box*. Dengan metode *black box* aplikasi akan diuji coba secara keseluruhan sesuai dengan fitur yang dimiliki. Uji coba ini dilakukan menggunakan *browser* Google Chrome dengan perangkat *smartphone* Realme 3.

-Uji Coba Fitur

Uji coba yang pertama adalah uji coba terhadap fitur utama yang ada seperti mode kamera, *voice command*, deteksi objek, dan halaman bantuan. Hasil dari uji coba fitur dapat dilihat pada Tabel 1.

Tabel 1. Tabel Hasil Uji Coba Fitur

| Aksi  | Yang diharapkan                             | Pengamatan                                  | Kesimpulan                  |
|---|---|---|-----------------------------|
| Menekan tombol ganti mode kamera depan                      | Mode kamera berubah menjadi kamera depan    | Mode kamera berubah menjadi kamera depan    | [v] Diterima<br>[ ] Ditolak |
| Menekan tombol ganti mode kamera belakang                   | Mode kamera berubah menjadi kamera belakang | Mode kamera berubah menjadi kamera belakang | [v] Diterima<br>[ ] Ditolak |
| Menekan tombol <i>voice command</i> dengan perintah 'halo'  | Mengeluarkan suara 'halo'                   | Mengeluarkan suara 'halo'                   | [v] Diterima<br>[ ] Ditolak |
| Menekan tombol tangkap objek dengan mendeteksi objek laptop | Mengeluarkan suara 'ini adalah laptop'      | Mengeluarkan suara ini adalah laptop        | [v] Diterima<br>[ ] Ditolak |
| Menekan tombol menu bantuan                                 | Menampilkan halaman bantuan                 | Menampilkan halaman bantuan                 | [v] Diterima<br>[ ] Ditolak |

Berdasarkan dari tabel hasil uji coba fitur dapat ditarik kesimpulan bahwa setiap fitur yang ada berjalan dengan sesuai tanpa adanya error, kesalahan program dan logika program.

-Uji Coba Perintah *Voice Command*

Pada uji coba perintah *voice command* dilakukan 2 tahap yaitu uji coba dengan data normal dengan 15 sampel dari 50 data dimana perintah yang dicoba adalah perintah yang tersedia seperti pada Tabel 2 dan uji coba dengan data salah dimana perintah yang dicoba adalah perintah yang tidak tersedia seperti pada Tabel 3.

Tabel 2. Tabel Hasil Uji Coba Perintah *Voice Command* (Data Normal)

| Perintah                | Yang diharapkan   | Pengamatan   | Kesimpulan                  |
|-------------------------|---|--|-----------------------------|
| Hai                     | Mengeluarkan suara 'halo'   | Mengeluarkan suara 'halo'                                  | [v] Diterima<br>[ ] Ditolak |
| Tanggal berapa sekarang | Mengeluarkan suara 'sekarang tanggal ...' (tanggal saat dilakukan perintah) | Mengeluarkan suara 'sekarang tanggal 5'                    | [v] Diterima<br>[ ] Ditolak |
| Hari apa sekarang       | Mengeluarkan suara 'sekarang hari ...' (hari saat dilakukan perintah)       | Mengeluarkan suara 'sekarang hari rabu'                    | [v] Diterima<br>[ ] Ditolak |
| Bulan apa sekarang      | Mengeluarkan suara 'sekarang bulan ...' (bulan saat dilakukan perintah)     | Mengeluarkan suara 'sekarang bulan mei'                    | [v] Diterima<br>[ ] Ditolak |
| Tahun berapa sekarang   | Mengeluarkan suara 'sekarang tahun ...' (tahun saat dilakukan perintah)     | Mengeluarkan suara 'sekarang tahun 2021'                   | [v] Diterima<br>[ ] Ditolak |
| Terima kasih            | Mengeluarkan suara 'sama - sama, senang membantu anda'                      | Mengeluarkan suara 'sama - sama, senang membantu anda'     | [v] Diterima<br>[ ] Ditolak |
| Apa itu                 | Mengeluarkan suara 'itu adalah ...' (objek yang dideteksi)                  | Mengeluarkan suara 'itu adalah laptop'                     | [v] Diterima<br>[ ] Ditolak |
| Apa ini                 | Mengeluarkan suara 'ini adalah ...' (objek yang dideteksi)                  | Mengeluarkan suara 'ini adalah laptop'                     | [v] Diterima<br>[ ] Ditolak |
| Ada apa di depan        | Mengeluarkan suara 'di depan kamu ada ...' (objek yang dideteksi)           | Mengeluarkan suara 'di depan kamu ada laptop dan keyboard' | [v] Diterima<br>[ ] Ditolak |

**Tabel 3.** Tabel Hasil Uji Coba Perintah Voice Command (Data Salah)

| Perintah            | Yang diharapkan  | Pengamatan   | Kesimpulan                  |
|---------------------|--|--|-----------------------------|
| Dimana saya         | Mengeluarkan suara 'Maaf, perintah yang dimasukkan belum tersedia' | Mengeluarkan suara 'Maaf, perintah yang dimasukkan belum tersedia' | [v] Diterima<br>[ ] Ditolak |
| Jam berapa sekarang | Mengeluarkan suara 'Maaf, perintah yang dimasukkan belum tersedia' | Mengeluarkan suara 'Maaf, perintah yang dimasukkan belum tersedia' | [v] Diterima<br>[ ] Ditolak |

|                    |  |  |                             |
|--------------------|--|--|-----------------------------|
| Dibelakang ada apa | Mengeluarkan suara 'Maaf, perintah yang dimasukkan belum tersedia' | Mengeluarkan suara 'Maaf, perintah yang dimasukkan belum tersedia' | [v] Diterima<br>[ ] Ditolak |
|--------------------|--|--|-----------------------------|

Pada tabel dari hasil uji coba perintah *voice command* dengan menggunakan data normal dan data salah dapat ditarik kesimpulan bahwa *voice command* berjalan dengan lancar tanpa adanya kesalahan program atau *error*.

**-Uji Coba Deteksi Objek**

Pada uji coba deteksi objek terdapat 2 tahap yaitu uji coba deteksi objek dengan menggunakan data normal dimana objek yang dipakai adalah objek yang terdaftar dan uji coba objek dengan menggunakan data salah dimana objek yang dipakai adalah objek yang tidak terdaftar atau objek yang terlihat tidak jelas. Pada uji coba ini menggunakan 10 sampel objek dari 80 objek yang terdaftar.

**Tabel 4.** Tabel Hasil Uji Coba Deteksi Objek (Data Normal)

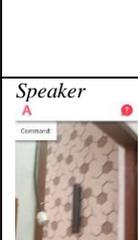
| Objek   | Yang diharapkan                                   | Pengamatan  | Kesimpulan                  |
|---|---|---|-----------------------------|
| Laptop<br>  | Mengeluarkan suara 'ini laptop'                   | Mengeluarkan suara 'ini laptop'                   | [v] Diterima<br>[ ] Ditolak |
| Remote<br> | Mengeluarkan suara 'ini adalah remote dan remote' | Mengeluarkan suara 'ini adalah remote dan remote' | [v] Diterima<br>[ ]         |

|   |  |  |                             |
|---|--|--|-----------------------------|
| Motor<br>  | Mengeluarkan suara 'ini adalah motor'  | Mengeluarkan suara 'ini adalah motor'  | [v] Diterima<br>[ ] Ditolak |
| Bangku<br> | Mengeluarkan suara 'ini adalah bangku' | Mengeluarkan suara 'ini adalah bangku' | [v] Diterima<br>[ ] Ditolak |

|   |  |  |                             |
|---|--|--|-----------------------------|
|    | Mengeluarkan suara 'ini adalah meja makan' | Mengeluarkan suara 'ini adalah meja makan' | [v] Diterima<br>[ ] Ditolak |
|    | Mengeluarkan suara 'ini adalah orang'      | Mengeluarkan suara 'ini adalah orang'      | [v] Diterima<br>[ ] Ditolak |
|    | Mengeluarkan suara 'ini adalah mobil'      | Mengeluarkan suara 'ini adalah mobil'      | [v] Diterima<br>[ ] Ditolak |
|   | Mengeluarkan suara 'ini adalah sepeda'     | Mengeluarkan suara 'ini adalah sepeda'     | [v] Diterima<br>[ ] Ditolak |
|  | Mengeluarkan suara 'ini adalah cangkir'    | Mengeluarkan suara 'ini adalah cangkir'    | [v] Diterima<br>[ ] Ditolak |

Tabel 5. Tabel Hasil Uji Coba Deteksi Objek (Data Salah)

| Objek   | Yang diharapkan  | Pengamatan   | Kesimpulan                  |
|---|--|--|-----------------------------|
|  | Mengeluarkan suara 'maaf objek tidak terdeteksi atau belum terdaftar, silahkan coba deteksi kembali. Pastikan objek terdeteksi dengan benar' | Mengeluarkan suara 'maaf objek tidak terdeteksi atau belum terdaftar, silahkan coba deteksi kembali. Pastikan objek terdeteksi dengan benar' | [v] Diterima<br>[ ] Ditolak |

|   |  |  |                             |
|---|--|--|-----------------------------|
|  | Mengeluarkan suara 'maaf objek tidak terdeteksi atau belum terdaftar, silahkan coba deteksi kembali. Pastikan objek terdeteksi dengan benar' | Mengeluarkan suara 'maaf objek tidak terdeteksi atau belum terdaftar, silahkan coba deteksi kembali. Pastikan objek terdeteksi dengan benar' | [v] Diterima<br>[ ] Ditolak |
|  | Mengeluarkan suara 'maaf objek tidak terdeteksi atau belum terdaftar, silahkan coba deteksi kembali. Pastikan objek terdeteksi dengan benar' | Mengeluarkan suara 'maaf objek tidak terdeteksi atau belum terdaftar, silahkan coba deteksi kembali. Pastikan objek terdeteksi dengan benar' | [v] Diterima<br>[ ] Ditolak |

#### IV. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, aplikasi *artificial intelligence* berbasis web untuk mendeteksi obyek berhasil dibuat menggunakan *library* Tensorflow Js dan React Js dengan memanfaatkan model dari Coco dataset sesuai dengan yang direncanakan.

Aplikasi ini berhasil mendeteksi 80 objek dari model Coco dengan baik dan dapat menjalankan semua perintah *voice command* yang terdaftar dengan lancar. Berdasarkan hasil percobaan tersebut, aplikasi ini dapat dijalankan pada perangkat Android dan laptop dengan menggunakan *browser* Google Chrome.

#### V. SARAN

Saran terhadap penelitian ini yaitu agar aplikasi *artificial intelligence* berbasis web ini dapat dikembangkan dengan harapan menggunakan model *machine learning* sendiri agar dapat menambahkan objek yang dapat dideteksi, menambahkan perintah *voice command* lebih banyak agar lebih membantu pengguna dalam menggunakan aplikasi dan mengoptimalkan performa aplikasi agar dapat berjalan dengan cepat tanpa adanya gangguan. Selain itu, dapat ditambahkan fitur seperti pemandu saat jalan kaki untuk membantu pengguna dalam berjalan dan pengembangan aplikasi diberbagai platform seperti aplikasi pada android dan iphone.

#### REFERENSI

- [1] Al Kadafi, A., & Utamingrum, F., "Deteksi Objek Penghalang Secara Real-Time Berbasis Mobile Bagi Penyandang Tunanetra Menggunakan Analisis Blob", Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 2, No. 1 (2017): 423-432.

- [2] Budiarto, Jian, & Jihadil Qudsi, "Deteksi Citra Kendaraan Berbasis Web Menggunakan JavaScript Framework Library", *Jurnal Matrik*, Vol. 18, No. 1 (2018): 125-133. <https://doi.org/10.30812/matrik.v18i1.330>.
- [3] Constantin, Yoshua, Ucu Darusalam, & Novi Dian Nathasia, "Aplikasi Personal Assistant Berbasis Voice Command Pada Sistem Operasi Android Dengan NLP", *Journal of Information Technology and Computer Science*, Vol. 5, No. 2 (2020): 121 - 128.
- [4] David Wibisono, Rolly Intan, & Endang Setyati, "Perancangan dan Pembuatan Aplikasi Android Pengkonversian Suara Menjadi Teks dalam Bahasa Indonesia dengan Machine Learning untuk Membantu Tunarungu", *Jurnal Infra*, Vol. 6, No. 1 (2018).
- [5] Dufan J. P. Manajang, Sherwin R.U.A. Sompie, & Agustinus Jacobus, "Implementasi Framework Tensorflow Object Detection Dalam Pengklasifikasian Kendaraan Bermotor", *Jurnal Teknik Informatika*, Vol. 15, No. 3 (2020): 171-178. <https://doi.org/10.35793/jti.15.3.2020.29775>.
- [6] Goodfellow, Ian, Yoshua Bengio, & Aaron Courville, "Deep Learning", United Kingdom, MIT Press: 2016.
- [7] Joshi, Ameet V., "Machine Learning Artificial Intelligence", Germany, Springer International Publishing: 2019.
- [8] Mohri, Mehryar, Afshin Rostamizadeh, & Amel Altawakar, "Foundation of Machine Learning", Second Edition. United Kingdom, MIT Press: 2019. <https://doi.org/10.1007/s00362-019-01124-9>.
- [9] Muhammad T.P. "Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web", *Jurnal Informatika: Jurnal Pengembangan IT*, Vol. 03, No. 01 (2018): hal. 126–129.
- [10] Nurul Isna Ganggalia, Apri Junaidi, & Fahrudin Mukti Wibowo, "Prototype Alat Pengendali Lampu dengan Perintah Suara menggunakan Arduino Uno Berbasis Web", *Jurnal RESTI*, Vol. 3, No. 3, (2019): 389 - 394. <https://doi.org/10.29207/resti.v3i3.1124>.
- [11] Prisky Ratna Aningtiyas, Agus Sumin, & Setia Wirawan, "Pembuatan Aplikasi Deteksi Objek Menggunakan TensorFlow Object Detection API dengan Memanfaatkan SSD MobileNet V2 Sebagai Model Pra – Terlatih", *Jurnal Ilmiah KOMPUTASI*, Vol. 19, No. 3 (2020). <http://dx.doi.org/10.32409/jikstik.19.3.68>