

DETEKSI OBJEK BERBASIS WEB SECARA REAL-TIME DENGAN UMPAN BALIK SUARA UNTUK PENYANDANG TUNANETRA

REAL-TIME WEB-BASED OBJECT DETECTION WITH VOICE FEEDBACK FOR THE BLIND

Calvin Ferry Alvino¹, Richard G. Mayopu²

Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana^{1,2}

calvin.ferry.alvino@gmail.com¹

ABSTRACT

Artificial Intelligence (AI) enables new opportunities in assistive technologies for visually impaired individuals. This study proposes a real-time web-based object detection system with integrated voice feedback to support environmental awareness. The application is developed using React.js and TensorFlow.js with the COCO-SSD model, running entirely on the client side without requiring additional hardware. Distance estimation is performed using a bounding box-based approximation, while audio output is generated through the Web Speech API. System evaluation is conducted through functional testing and performance measurement, including frame rate, response latency, and detection confidence as an indicator of model certainty. Experimental results show that the system achieves 15–30 FPS, detection confidence of 70–95%, and voice response latency of 150–300 ms. The main contribution of this study is the integration of detection, distance estimation, and voice feedback in a lightweight, installation-free web application, demonstrating its potential as an accessible assistive solution.

Keywords: Object Detection, TensorFlow.js, React.js, Accessibility, Visually Impaired

ABSTRAK

Kecerdasan buatan (*Artificial Intelligence/AI*) membuka peluang baru dalam pengembangan teknologi asistif bagi penyandang tunanetra. Penelitian ini mengusulkan sistem deteksi objek berbasis web secara real-time yang dilengkapi dengan umpan balik suara untuk meningkatkan kesadaran lingkungan pengguna. Aplikasi dikembangkan menggunakan React.js dan TensorFlow.js dengan model COCO-SSD yang berjalan sepenuhnya di sisi klien tanpa memerlukan perangkat keras tambahan. Estimasi jarak dilakukan menggunakan pendekatan berbasis bounding box, sedangkan keluaran suara dihasilkan melalui *Web Speech API*. Evaluasi sistem dilakukan melalui pengujian fungsional dan pengukuran performa, meliputi frame rate, latensi respons, serta tingkat kepercayaan deteksi sebagai indikator kepastian model. Hasil eksperimen menunjukkan bahwa sistem mampu mencapai 15–30 FPS, tingkat kepercayaan deteksi 70–95%, serta latensi suara 150–300 milidetik. Kontribusi utama penelitian ini adalah integrasi deteksi objek, estimasi jarak, dan umpan balik suara dalam aplikasi web ringan tanpa instalasi.

Kata Kunci: Deteksi Objek, TensorFlow.js, React.js, Aksesibilitas, Tunanetra

PENDAHULUAN

Perkembangan teknologi kecerdasan buatan (*Artificial Intelligence/AI*) terus menghadirkan solusi inovatif untuk meningkatkan aksesibilitas bagi berbagai kelompok masyarakat, termasuk penyandang tunanetra (Ariyani dkk., 2022). Bagi pengguna tunanetra, kemampuan mengenali objek di sekitar serta memahami kondisi spasial masih menjadi tantangan besar akibat keterbatasan persepsi visual (Hulu dkk., 2024), (Garno dkk., 2025). Aktivitas sehari-hari seperti mengidentifikasi objek di sekitar, mendeteksi keberadaan orang, atau

memperkirakan jarak objek memerlukan sistem bantu yang mampu menyampaikan informasi secara cepat dan jelas melalui media non-visual (Yusuf dkk., 2022).

Kemajuan teknologi web modern memungkinkan model pembelajaran mesin dijalankan langsung di dalam lingkungan browser tanpa memerlukan perangkat keras khusus maupun instalasi tambahan (Rosa dkk., 2020). TensorFlow.js memungkinkan deteksi objek secara *real-time* menggunakan kamera perangkat standar sehingga membuka peluang pengembangan aplikasi AI yang ringan dan mudah diakses (Arnesia dkk., 2022). Dibandingkan

dengan pendekatan berbasis *mobile* atau *embedded system*, solusi berbasis browser memiliki keunggulan dalam hal aksesibilitas, privasi, serta kemudahan implementasi karena seluruh proses dilakukan di sisi klien tanpa mengirim data ke server eksternal (Rafiq dkk., 2025).

Penelitian sebelumnya telah mengembangkan sistem deteksi objek sebagai alat bantu dengan berbagai pendekatan. Arnesia dkk. (2022) mengembangkan sistem deteksi objek berbasis web menggunakan TensorFlow.js dan React.js, namun sistem tersebut masih berfokus pada hasil deteksi visual tanpa integrasi estimasi jarak maupun umpan balik suara otomatis. Sementara itu, Yusuf dkk. (2022) mengembangkan sistem berbasis perangkat keras menggunakan *OpenCV* dan *YOLOv4-Tiny*. Meskipun memiliki performa deteksi yang baik, sistem tersebut memerlukan perangkat tambahan sehingga membatasi aksesibilitas serta meningkatkan kompleksitas implementasi (Moh Yusup dkk., 2024).

Perbandingan penelitian terdahulu dengan sistem yang diusulkan dapat dilihat pada Tabel 1.

Tabel 1. Perbandingan Sistem yang Diusulkan dengan Penelitian Sebelumnya

Aspek	Arnesia et al. (2022)	Yusuf et al. (2022)	Sistem yang Diusulkan
Platform	Web (Browser)	Embedded Hardware (Raspberry Pi)	Web (Browser)
Framework / Library	TensorFlow.js, React.js	OpenCV, YOLOv4-Tiny	TensorFlow.js, React.js
Model Deteksi	COCO-SSD	YOLOv4-Tiny	COCO-SSD
Estimasi Jarak	Tidak tersedia	Berbasis sensor	Bounding Box-Based
Umpan Balik Suara	Tidak tersedia	Tersedia	Web Speech API
Kontribusi Utama	Real-Time Web Detection	Embedded Assistive System	Integrated Web Detection with Distance Estimation and Voice Feedback

Sumber: (Arnesia dkk., 2022) dan (Yusuf dkk., 2022)

Tabel 2. Perbandingan Sistem yang Diusulkan dengan Penelitian Sebelumnya

Penelitian	Kelemahan
------------	-----------

Arnesia et al.	Tidak menyediakan umpan balik suara
Yusuf et al.	Membutuhkan perangkat keras tambahan
Penelitian ini	Estimasi jarak masih sederhana

Sumber: (Arnesia dkk., 2022) dan (Yusuf dkk., 2022)

Berbeda dengan penelitian sebelumnya, penelitian ini mengintegrasikan deteksi objek real-time, estimasi jarak berbasis *bounding box*, serta umpan balik suara dalam satu aplikasi berbasis web yang berjalan sepenuhnya di sisi klien (Arnesia dkk., 2022). Sistem dikembangkan menggunakan React.js dan TensorFlow.js dengan model COCO-SSD, sehingga seluruh proses mulai dari pengambilan video hingga keluaran suara dilakukan langsung di dalam browser tanpa memerlukan server backend maupun perangkat tambahan.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem bantu berbasis web guna meningkatkan kesadaran lingkungan dan aksesibilitas bagi penyandang tunanetra. Kontribusi utama penelitian ini adalah integrasi deteksi objek, estimasi jarak, dan umpan balik suara dalam aplikasi ringan (*lightweight*) yang menekankan aspek aksesibilitas, privasi data, serta kemudahan implementasi pada berbagai perangkat.

METODE

Metode penelitian ini dirancang untuk menjelaskan tahapan perancangan, pengembangan, dan evaluasi sistem deteksi objek berbasis web secara real-time yang diusulkan. Pendekatan yang digunakan meliputi desain penelitian, perancangan arsitektur sistem, metode estimasi jarak, implementasi sistem, serta pengujian performa untuk memastikan sistem dapat berjalan secara optimal dalam mendukung kebutuhan aksesibilitas bagi penyandang tunanetra.

2.1 Desain Penelitian

Penelitian ini menggunakan metode *Research and Development (R&D)* dengan tujuan menghasilkan produk berupa aplikasi deteksi objek berbasis web secara *real-time* yang memanfaatkan teknologi kecerdasan buatan (Moore dkk., 2023). Model pengembangan yang digunakan mengacu pada tahapan *ADDIE (Analysis, Design, Development, Implementation, Evaluation)* yang disederhanakan sesuai kebutuhan penelitian (Spatioti dkk., 2022). Tahapan penelitian meliputi analisis kebutuhan, perancangan sistem, implementasi, serta pengujian awal.

Pendekatan *R&D* dipilih karena penelitian ini berfokus pada perancangan, pengembangan, dan evaluasi sistem bantu yang bersifat fungsional, bukan hanya analisis teoritis (Waruwu, 2024). Metode ini memungkinkan pengembangan produk secara sistematis serta perbaikan berulang selama proses implementasi.

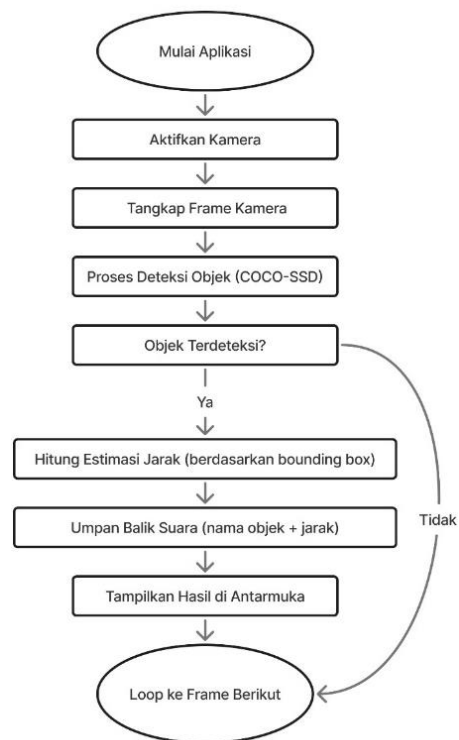
Model *ADDIE* dipilih karena memiliki kerangka kerja yang terstruktur dan fleksibel, sehingga setiap tahapan analisis, perancangan, pengembangan, implementasi, dan evaluasi dapat dilakukan secara berurutan dengan tetap memungkinkan penyesuaian berdasarkan hasil pengujian (Ebrahimi dkk., 2025). Pada penelitian ini, model tersebut disederhanakan dengan menekankan pada pengembangan teknis dan pengujian fungsional yang sesuai dengan aplikasi AI berbasis web (Muhammad Rusdi dkk., 2022).

Modifikasi model *ADDIE* dalam penelitian ini terletak pada tahap evaluasi yang difokuskan pada pengujian fungsional dan pengukuran kinerja sistem, bukan pada validasi pengguna dalam skala besar. Evaluasi dilakukan melalui pengujian performa seperti frame rate (FPS), latensi respons suara, serta konsistensi deteksi untuk memastikan sistem dapat berjalan secara stabil dalam kondisi *real-time*.

2.2 Arsitektur Sistem

Arsitektur sistem dirancang sebagai alur pemrosesan video secara *real-time*, di mana frame video yang diperoleh dari kamera perangkat diproses menggunakan *TensorFlow.js* dengan model *COCO-SSD* untuk melakukan deteksi objek (Arnesia dkk., 2022). Ukuran *bounding box* yang dihasilkan dari proses deteksi digunakan sebagai dasar untuk memperkirakan jarak objek terhadap kamera. Selanjutnya, informasi berupa nama objek dan estimasi jarak dikonversi menjadi keluaran suara menggunakan *Web Speech API*, sehingga sistem mampu memberikan umpan balik audio kepada pengguna.

Sistem melakukan pemrosesan video secara *real-time* menggunakan *TensorFlow.js* sebagai mesin inferensi yang menjalankan model deteksi objek *COCO-SSD* langsung di dalam *browser*.



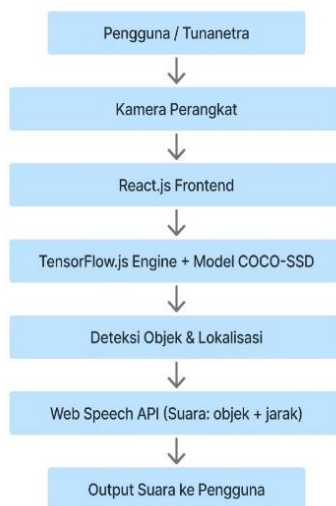
Gambar 1. Alur Sistem Aplikasi
(*Application Flowchart*)

Alur sistem dimulai dari proses inisialisasi aplikasi, kemudian sistem mengaktifkan kamera untuk memperoleh input visual secara *real-time*. Selanjutnya, sistem menangkap frame video secara terus-menerus dan memrosesnya

menggunakan model *COCO-SSD* untuk mendeteksi objek.

Apabila objek terdeteksi, sistem akan menghitung estimasi jarak berdasarkan ukuran *bounding box*. Informasi objek dan jarak kemudian dikonversi menjadi suara menggunakan *Web Speech API*. Proses ini berjalan secara berulang (*looping*) selama aplikasi aktif.

Pendekatan *client-side* dipilih untuk mengurangi latensi dan menghilangkan ketergantungan terhadap server. Selain itu, pendekatan ini meningkatkan privasi karena data tidak dikirim ke server eksternal serta memungkinkan aplikasi berjalan di berbagai perangkat tanpa instalasi tambahan (Gde dkk., 2025).



Gambar 2. Diagram Blok Sistem (System Block Diagram)

Diagram blok menggambarkan alur data dari pengguna hingga keluaran suara. Kamera perangkat menangkap data visual yang kemudian diproses oleh antarmuka *React.js*. Selanjutnya, data diproses oleh *TensorFlow.js* untuk mendeteksi objek. Hasil deteksi digunakan untuk menentukan lokasi dan ukuran *bounding box*, yang kemudian digunakan untuk menghitung estimasi jarak. Informasi tersebut dikonversi menjadi suara menggunakan *Web Speech API* dan disampaikan kepada pengguna.

2.3 Metode Estimasi Jarak Objek

Estimasi jarak objek dilakukan menggunakan pendekatan berbasis

bounding box dengan persamaan sebagai berikut:

$$D = \frac{K}{H}$$

Pada persamaan tersebut, D menyatakan jarak objek dalam satuan meter, H merupakan tinggi *bounding box* dalam satuan piksel, dan K adalah konstanta kalibrasi yang diperoleh melalui proses pengujian awal sistem. Nilai K diperoleh melalui proses kalibrasi empiris dengan menempatkan objek referensi pada jarak tertentu. Dalam penelitian ini, kalibrasi dilakukan pada jarak 5 meter menggunakan objek manusia sebagai referensi. Pengukuran dilakukan sebanyak lima kali untuk memperoleh nilai rata-rata yang stabil.

Konstanta K dihitung menggunakan persamaan:

$$K = \frac{D}{H}$$

Berdasarkan hasil kalibrasi, diperoleh nilai K sebesar ± 850 . Sebagai contoh, jika tinggi *bounding box* sebesar 170 piksel, maka jarak objek dapat dihitung sebagai berikut:

$$D = \frac{850}{170} = 5 \text{ meter}$$

Metode ini bersifat sederhana dan tidak mempertimbangkan sudut pandang maupun variasi ukuran objek, namun cukup efektif untuk memberikan estimasi jarak kasar yang berguna dalam konteks sistem bantu bagi tunanetra.

2.4 Implementasi Sistem

Sistem dikembangkan menggunakan *React.js* sebagai kerangka kerja antarmuka pengguna untuk membangun aplikasi web yang interaktif dan responsif. *TensorFlow.js* digunakan untuk memuat dan menjalankan model *COCO-SSD*. Proses deteksi dilakukan secara berulang menggunakan fungsi *requestAnimationFrame* untuk menjaga performa *real-time*.

Selain itu, sistem menampilkan informasi pendukung seperti *frames per second (FPS)*, penggunaan CPU, dan kapasitas RAM sebagai indikator performa sistem selama berjalan.

2.5 Pengujian Sistem

Pengujian dilakukan untuk mengevaluasi fungsi deteksi objek, estimasi jarak, serta respons umpan balik suara. Data performa digunakan sebagai indikator stabilitas sistem saat dijalankan di dalam *browser*.

Pengujian dilakukan terhadap beberapa kategori objek dengan variasi jarak antara 1 hingga 10 meter. Setiap objek diuji sebanyak lima kali untuk mengamati konsistensi hasil deteksi dan estimasi jarak.

Akurasi deteksi dalam penelitian ini mengacu pada tingkat kepercayaan (*confidence score*) yang dihasilkan oleh model *COCO-SSD* melalui *TensorFlow.js*. Nilai akurasi diperoleh dari pengamatan berulang dalam berbagai kondisi pengujian.

Rentang akurasi yang diperoleh (70%–95%) mencerminkan konsistensi deteksi objek dalam berbagai kondisi jarak dan pencahayaan, bukan berdasarkan metrik formal seperti *Intersection over Union (IoU)*.

Penelitian ini tidak menggunakan metrik evaluasi formal seperti *mean Average Precision (mAP)* maupun *Intersection over Union (IoU)* karena model *COCO-SSD* yang digunakan merupakan model pralatih (*pre-trained model*) yang dijalankan tanpa proses pelatihan ulang (*retraining*). Fokus penelitian ini terletak pada integrasi sistem berbasis web secara *real-time* serta evaluasi performa operasional seperti *frame rate*, latensi respons suara, dan konsistensi deteksi dalam konteks aplikasi bantu. Oleh karena itu, nilai *confidence score* digunakan sebagai indikator stabilitas deteksi selama proses inferensi berlangsung.

Tabel 3. Konfigurasi Lingkungan Pengujian

Parameter	Deskripsi
-----------	-----------

Perangkat Kamera	Laptop Webcam
Resolusi	640 × 480 piksel
Peramban	Google Chrome
Kondisi Pencahayaan	Dalam ruangan (<i>indoor</i> , pencahayaan normal)
Jenis Perangkat	Laptop
Koneksi Internet	Tidak diperlukan saat inferensi

Sumber: Hasil pengujian penelitian

Pengukuran penggunaan CPU dan RAM dilakukan untuk mengetahui tingkat konsumsi sumber daya sistem selama proses inferensi berjalan secara *real-time*. Pengambilan data dilakukan menggunakan fitur *Task Manager* pada sistem operasi untuk memantau persentase penggunaan CPU dan memori secara keseluruhan, serta *Chrome DevTools Performance Monitor* untuk mengamati konsumsi sumber daya pada level peramban (*browser*).

Pemantauan dilakukan selama aplikasi aktif dan menjalankan proses deteksi objek secara kontinu. Setiap skenario pengujian berlangsung selama 20–30 detik, di mana sistem memproses ratusan frame video secara berulang. Nilai penggunaan CPU dan RAM yang dicatat merupakan nilai rata-rata (*average usage*) selama periode pengamatan tersebut, bukan nilai puncak (*peak usage*), sehingga dapat merepresentasikan beban kerja sistem secara lebih stabil dan realistis dalam kondisi operasional normal.

HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil eksperimen yang diperoleh dari implementasi aplikasi deteksi objek *real-time* berbasis web yang diusulkan. Evaluasi difokuskan pada performa deteksi, estimasi jarak, respons umpan balik suara, serta stabilitas sistem dalam kondisi *real-time*.

3.1 Hasil

Sistem yang diusulkan berhasil melakukan deteksi objek secara *real-time* pada berbagai kategori, termasuk manusia,

botol, kendaraan, dan peralatan rumah tangga. Pada resolusi 640×480 piksel menggunakan kamera bawaan laptop, sistem mampu mencapai kecepatan pemrosesan sebesar 15–30 frames per second (FPS).

Estimasi jarak berbasis pendekatan *bounding box* menghasilkan keluaran yang terukur dalam rentang 0,5 hingga 11 meter, sebagaimana ditunjukkan pada Tabel 4.

Tabel 4. Hasil Pengujian Deteksi Objek

Objek	Observasi & Kesimpulan
<p><i>Bottle</i></p> 	<p>Jarak: 1 m Umpan balik suara: "Bottle terdeteksi, pada jarak 1 m." Tingkat kepercayaan deteksi: 90–95%</p>
<p><i>Remote</i></p> 	<p>Jarak: 2 m Umpan balik suara: "Remote terdeteksi, pada jarak 2 m." Tingkat kepercayaan deteksi: 85–90%</p>
<p><i>Person</i></p> 	<p>Jarak: 3 m Umpan balik suara: "Person terdeteksi, pada jarak 3 m." Tingkat kepercayaan deteksi: 85–90%</p>
<p><i>Motorcycle</i></p> 	<p>Jarak: 5 m Umpan balik suara: "Motorcycle terdeteksi, pada jarak 5 m." Tingkat kepercayaan deteksi: 80–90%</p>
<p><i>Person</i></p>	<p>Jarak: 7 m</p>



Umpan balik suara: "Person terdeteksi, pada jarak 7 m."
Tingkat kepercayaan deteksi: 80–90%

Person



Jarak: 10 m
Umpan balik suara: "Person terdeteksi, pada jarak 10 m."
Tingkat kepercayaan deteksi: 70–80%

Car



Jarak: 11 m
Umpan balik suara: "Car terdeteksi, pada jarak 11 m."
Tingkat kepercayaan deteksi: 80–95%

Nilai tingkat kepercayaan deteksi berada pada rentang 70% hingga 95%, tergantung pada jarak objek dan ukurannya dalam frame kamera. Nilai kepercayaan yang lebih tinggi diperoleh pada jarak yang lebih dekat, sedangkan nilai yang lebih rendah terjadi ketika objek terlihat lebih kecil.

Metode estimasi berbasis *bounding box* menghasilkan pengukuran jarak yang bersifat pendekatan (*approximate*), dengan deviasi yang meningkat pada jarak yang lebih jauh. Pada jarak di atas 7 meter, rata-rata kesalahan estimasi berkisar $\pm 0,5$ –1 meter.

Sistem umpan balik suara yang diimplementasikan menggunakan *Web Speech API* menunjukkan waktu respons antara 150–300 milidetik setelah proses deteksi. Waktu respons ini memungkinkan

penyampaian informasi audio secara hampir *real-time*.

```
const ut = new SpeechSynthesisUtterance(text);
window.speechSynthesis.speak(ut);
```

Gambar 3. Umpan Balik Suara

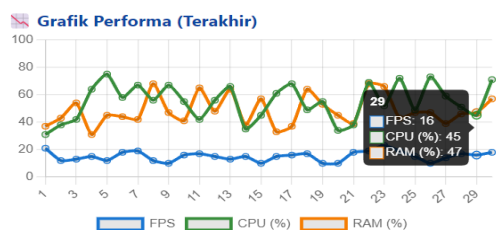
Gambar 3 menunjukkan mekanisme umpan balik suara yang terintegrasi dalam arsitektur sistem. Dengan mengonversi hasil deteksi menjadi suara sintesis secara langsung di dalam *browser*, sistem mampu mempertahankan latensi yang rendah sekaligus menjaga privasi pengguna. Pendekatan ini meningkatkan aksesibilitas dengan menyediakan informasi audio secara *real-time* tanpa memerlukan layanan pemrosesan eksternal.

Tabel 5. Pengujian Performa (FPS, CPU, RAM)

Jarak Asli (m)	Estimasi (m)	Error (m)
1	1.1	0.1
3	2.8	0.2
5	5.0	0.0
7	6.2	0.8
10	9.0	1.0

Tabel 6. Pengujian Performa (FPS, CPU, RAM)

Parameter	Hasil
Rata-rata FPS	15–30 FPS
CPU (<i>simulated</i>)	30–55%
RAM (<i>simulated</i>)	30–50%
Respons suara	150–300 ms



Gambar 3. Umpan Balik Suara

Berdasarkan Tabel 6, sistem menunjukkan performa yang stabil dalam kondisi *real-time*. Aplikasi berjalan pada 15–30 FPS, dengan penggunaan CPU sebesar 30–55% dan RAM sebesar 30–50%.

Pengujian sistem dilakukan sebanyak 10 kali untuk setiap objek dan skenario

jarak. Setiap pengujian berlangsung selama 20–30 detik, menghasilkan sekitar 300–900 frame per percobaan. Secara keseluruhan, sistem memproses sekitar 3.000–9.000 frame selama seluruh eksperimen.

Hasil pengujian menunjukkan bahwa sistem mampu mempertahankan performa deteksi yang konsisten serta umpan balik suara yang stabil pada sebagian besar skenario pengujian.

3.2 Pembahasan

Hasil eksperimen menunjukkan bahwa sistem berbasis *browser* yang diusulkan mampu mencapai performa *real-time* tanpa memerlukan akselerasi perangkat keras khusus. Kecepatan *frame rate* sebesar 15–30 FPS sudah cukup untuk menjaga kontinuitas persepsi dalam aplikasi bantu.

Penurunan tingkat kepercayaan deteksi objek yang bergerak dalam frame gambar. Keterbatasan ini sejalan dengan karakteristik *convolutional neural networks (CNN)* yang cenderung memiliki performa lebih rendah dalam mendeteksi objek yang bergerak (Walle dkk., 2022).

Meskipun metode estimasi jarak berbasis *bounding box* tidak memberikan presisi tinggi, pendekatan ini dinilai cukup memadai untuk tujuan aksesibilitas. Dalam konteks sistem bantu, kesadaran spasial secara umum (*approximate awareness*) lebih penting dibandingkan akurasi pengukuran yang sangat presisi.

Penggunaan pemrosesan di sisi klien (*client-side*) menghilangkan latensi jaringan serta meningkatkan privasi data karena tidak ada transmisi data ke server eksternal. Selain itu, pendekatan ini juga meningkatkan aksesibilitas sistem, karena aplikasi dapat dijalankan langsung melalui *browser* tanpa instalasi maupun infrastruktur *backend*.

3.3 Perbandingan dengan Penelitian Sebelumnya

Jika dibandingkan dengan penelitian Arnesia dkk. (2022), sistem yang diusulkan tidak hanya menampilkan hasil deteksi

visual tetapi juga menambahkan estimasi jarak dan umpan balik suara otomatis dalam satu arsitektur terpadu berbasis browser. Dibandingkan dengan Yusuf dkk. (2022) yang menggunakan pendekatan embedded hardware berbasis YOLOv4-Tiny, sistem ini tidak memerlukan perangkat tambahan dan sepenuhnya berjalan di sisi klien (client-side) sehingga lebih ringan dan mudah diakses. Dari sisi performa, kecepatan 15–30 FPS menunjukkan bahwa pendekatan berbasis TensorFlow.js masih mampu mempertahankan kinerja real-time tanpa akselerasi GPU khusus.

SIMPULAN

Berdasarkan hasil pengembangan, implementasi, dan pengujian aplikasi deteksi objek *real-time* berbasis web menggunakan *TensorFlow.js* dan *React.js*, dapat disimpulkan bahwa sistem yang dibangun mampu berjalan secara efektif dalam mendukung aksesibilitas bagi penyandang tunanetra. Aplikasi berhasil melakukan deteksi objek secara *real-time* menggunakan model *COCO-SSD* yang berjalan sepenuhnya di sisi klien (*client-side*) melalui *browser*, dengan kecepatan pemrosesan mencapai 15–30 *frames per second (FPS)* sehingga memenuhi kebutuhan sistem *real-time*.

Fitur estimasi jarak berbasis perhitungan tinggi *bounding box* mampu memberikan perkiraan jarak objek dalam rentang 0,5–11 meter. Meskipun metode ini bersifat pendekatan (*approximate*) dan memiliki keterbatasan pada jarak yang lebih jauh, hasil yang diperoleh dinilai cukup untuk memberikan kesadaran spasial bagi pengguna. Selain itu, fitur umpan balik suara menggunakan *Web Speech API* mampu memberikan respons cepat dalam kisaran 150–300 milidetik, sehingga informasi objek dan jarak dapat disampaikan secara langsung tanpa memerlukan layanan eksternal.

Integrasi komponen antarmuka seperti tampilan *bounding box*, label objek, serta informasi performa (*FPS*, CPU, RAM) menunjukkan bahwa penggunaan

React.js efektif dalam membangun aplikasi web yang responsif dan terstruktur. Secara keseluruhan, sistem ini mampu mengubah informasi visual menjadi informasi audio secara *real-time* dan dapat diakses langsung melalui *browser* tanpa instalasi, sehingga berpotensi menjadi solusi ringan (*lightweight*) untuk meningkatkan aksesibilitas digital.

Selain itu, evaluasi berbasis pengguna (*user-based evaluation*) menggunakan instrumen seperti *System Usability Scale (SUS)* atau pengujian langsung terhadap penyandang tunanetra diperlukan pada penelitian lanjutan untuk mengukur aspek kenyamanan, kemudahan penggunaan, dan efektivitas sistem dalam situasi nyata.

Namun demikian, penelitian ini masih memiliki keterbatasan, khususnya pada metode estimasi jarak yang masih bergantung pada ukuran objek dan posisi kamera, serta belum dilakukannya pengujian langsung dengan pengguna tunanetra. Oleh karena itu, pengembangan selanjutnya disarankan untuk meningkatkan akurasi estimasi jarak menggunakan metode seperti *depth estimation*, kamera stereo, atau sensor tambahan, mengoptimalkan performa pada perangkat dengan spesifikasi rendah, serta melakukan evaluasi berbasis pengguna.

DAFTAR PUSTAKA

- Ariyani, S., Nugroho, A. B., & Mubarak, A. S. T. (2022). Object Detection Tool for the Blind Based on Ai Mobilenet on Raspberry Pi 3B. *Journal of Electrical Engineering and Computing (ELKOM)*, 4(1), 73–90.
- Arnesia, P. D., Pratama, N. A., & Syafrina, F. (2022). Artificial Intelligence Application for Web-Based Object Detection Using Tensorflow Js, React Js and Coco Dataset Libraries. *JSiI (Journal System Information)*, 9(1), 62–69.
- Ebrahimi, F., Masoudian, T., & Khiabani, M. M. (2025). Integrating ADDIE Needs Assessment with Kirkpatrick

- Evaluation: A Systematic Review. *Asian Journal of Education and Social Studies*, 51(3), 350–376. <https://doi.org/10.9734/ajess/2025/v51i31832>
- Garno, G., Sarwosri, S., & Febriandiza, A. (2025). User Experience Accessibility of Web Content for People with Cognitive Disabilities Using the Double Diamond Method. *National Journal of Technology and Information Systems*, 3(1), 282–293.
- Gde, P., Bhanuartha, A., Pinandito, A., & Akbar, M. A. (2025). Analisis Perbandingan Server Side dan Client Side Data Fetching pada Framework Next.js (Studi Kasus Aplikasi Online Course) (Vol. 9, Nomor 3). <http://j-ptiik.ub.ac.id>
- Hulu, F. A., Setyawan, G. C., & Lase, K. J. D. (2024). Implementation of Object Recognition and Tracking System on ESP32CAM-Based 4WD Mobile Robot for Navigation in 2D Space. *Jutisi: Scientific Journal of Informatics Engineering and Information Systems*, 13(2), 1622–1634.
- Moh Yusup, R., Faris Anugrah, A., Desmonda Muslimah, D., Mentari Widya Ningrum Permana, S., Yuliani, S., & Majalengka, U. (2024). PENDETEKSIAN OBJEK MENGGUNAKAN OPENCV DAN METODE YOLOv4-TINY UNTUK MEMBANTU TUNANETRA. *Hal. 59 Journal of Computer Science and Information Technology (JCSIT)*, 1.
- Moore, K. A., Leighton, T., & Ostrowsky, J. T. (2023). A research and development (R&D) roadmap for broadly protective coronavirus vaccines: A pandemic preparedness strategy. *Vaccine*, 41, 2101–2112.
- Muhammad Rusdi, Rina Alfah, & Haji Sirajuddin. (2022). IMPLEMENTATION OF THE ADDIE MODEL (ANALYSIS, DESIGN, DEVELOPMENT, IMPLEMENTATION) IN PHP- BASED E-LEARNING IN THE ERA OF PANDEMIC.
- Rafiq, M. A., Zen, M., & Kurniawan, F. (2025). Development of Voice-Based Applications to Help Accessibility of Information for Pertuni Members Using the Design Thinking Method. *Journal of Computer Technology Information Systems (JUKTISI)*, 4(1), 167–173.
- Rosa, A. A., Simon, B. A., & Lieanto, K. S. (2020). Portable Air Pollutant Detection System Using MQ-7 and MQ-135 Sensors. *ULTIMA Computing*, 12(1).
- Spatioti, A. G., Kazanidis, I., & Pange, J. (2022). A comparative study of the ADDIE instructional design model in distance education. *Information*, 13(9).
- Walle, H., De Runz, C., Serres, B., & Venturini, G. (2022). A Survey on Recent Advances in AI and Vision-Based Methods for Helping and Guiding Visually Impaired People. *Applied Sciences (Switzerland)*, 12(5). <https://doi.org/10.3390/app12052308>
- Waruwu, M. (2024). Metode Penelitian dan Pengembangan (R&D): Konsep, Jenis, Tahapan dan Kelebihan. *Jurnal Ilmiah Profesi Pendidikan*, 9(2), 1220–1230. <https://doi.org/10.29303/jipp.v9i2.2141>
- Yusuf, R. M., Anugrah, A. F., Muslimah, D. D., Permana, S. M. W. N., Yuliani, S., & Majalengka, U. (2022). OBJECT DETECTION USING OPENCV AND YOLOv4-TINY METHOD TO HELP THE BLIND. *Journal of Computer Science and Information Technology (JCSIT)*, 1, 59.