

19103020255_Abrham Galatiano Ohny.docx.pdf

by simitiga@unpkdr.ac.id 1

Submission date: 16-Jun-2025 11:58AM (UTC+0800)

Submission ID: 2588161304

File name: 19103020255_Abrham_Galatiano_Ohny.docx.pdf (5.91M)

Word count: 6258

Character count: 36866

15

BAB I

PENDAHULUAN

A. Latar Belakang

Tunanetra adalah istilah umum yang digunakan untuk kondisi seseorang yang mengalami gangguan atau hambatan dalam indra penglihatannya [1], maka orang yang memiliki kondisi tersebut untuk mengetahui keberadaan objek, orang tunanetra akan lebih bergantung kepada indra selain penglihatan yaitu pendengaran, pembau, peraba, dan lain-lain.

Namun hanya dengan menggunakan indra selain penglihatan orang yang memiliki kondisi tersebut tidak dapat mengenali objek secara langsung. Contohnya benda mati yang harus meraba benda tersebut secara menyeluruh agar dapat dikenal. Karena penyandang tunanetra tidak dapat mengenali objek secara langsung maka beresiko terluka karena meraba sebuah benda tajam. Selain itu penyandang tunanetra lebih peka terhadap suara, sehingga dapat mengenali lawan bicaranya dengan mendengar nada bicara, cara bicara, dan lain-lain. Maka dari itu fokus penyandang tunanetra lebih mudah terpecah jika pada posisi dimana terdapat lebih dari satu objek yang mengeluarkan suara secara bersamaan. Seperti contoh kantin, dimana terdapat lebih dari satu kumpulan orang yang bercakap.

Dari permasalahan diatas ditawarkan sebuah solusi, yaitu membuat sebuah sistem pengenalan objek menggunakan *YOLOv8*. *Input* dari sistem berupa *frame* yang ditangkap melalui kamera lalu diproses sehingga dapat mengenali objek. Untuk *output* dari sistem berupa suara dengan kata-kata objek apa yang dikenal. Sistem ini masih dibuat pada platform desktop, karena keterbatasan resource, untuk kedepannya sistem ini dapat di-implementasikan pada platform lain.

B. Identifikasi Masalah

1. Penyandang tunanetra tidak dapat mengenali benda secara langsung sehingga beresiko melukai diri sendiri
2. Penyandang tunanetra sensitif terhadap kebisingan

C. Batasan Masalah

Adanya beberapa batasan masalah pada penelitian ini :

1. Platform yang tersedia oleh penulis masih berupa desktop (*Windows*)
2. Untuk model kecerdasan buatan untuk sistem ini masih menggunakan yang sudah disediakan oleh *YOLOv8* yaitu *YOLOv8n*
3. Maksimal resolusi untuk deteksi yaitu 640x640 (resolusi kamera)
4. Sistem dapat berjalan dengan menggunakan *library* yang tersedia untuk bahasa pemrograman *python* yaitu *playsound* untuk mengeluarkan suara, dan *gTTS* untuk fungsi *text-to-speech*
5. Sistem membutuhkan koneksi internet
6. Penggunaan sistem ini berfokus untuk digunakan penyandang tunanetra di sebuah ruangan atau *indoor* (dapur, kamar, dan lain-lain)
7. Sistem yang dibuat masih berbentuk *prototype* dimana masih ada banyak yang diperbaiki, fungsi yang harus ditambahkan, maupun dibuat ulang sehingga berfungsi secara optimal.

D. Rumusan Masalah

Rumusan masalah yang terkait dengan penelitian ini yaitu. Bagaimana membuat sistem pengenalan objek menggunakan *YOLOv8* untuk penyandang tunanetra mengenali objek yang disekitarnya.

E. Tujuan Penelitian

Tujuan penelitian ini yaitu, membuat sebuah sistem pengenalan objek menggunakan *YOLOv8* yang digunakan untuk penyandang tunanetra mengenali objek di sekitarnya.

F. Manfaat dan Kegunaan Penelitian

Harapan penulis dalam manfaat dan kegunaan dari penelitian ini yaitu membantu peneliti selanjutnya yang memiliki masalah relevan dengan penelitian penulis. Dan dengan penelitian ini dapat meringankan masalah yang dialami oleh orang-orang yang memiliki kondisi yang disebutkan.

G. Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah metode *Research and Development (R&D)* dengan tujuan mengembangkan sebuah sistem

deteksi objek menggunakan *YOLOv8* yang digunakan oleh penyandang tunanetra sebagai penglihatan

Dikutip dari penelitian sebelumnya yang menggunakan metode penelitian yang sama. Menurut Sugiyono metode penelitian *Research and Development (R&D)* adalah metode penelitian yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut. Untuk menghasilkan produk tertentu digunakan penelitian yang bersifat analisis kebutuhan dan untuk menguji keefektifan produk tersebut supaya dapat berfungsi di masyarakat luas [2]. Maka diperlukan untuk menguji efektivitas sebuah produk, dalam penelitian ini dengan hasil akhir produk berupa sistem deteksi objek menggunakan *YOLOv8*.

Demi mengukur tingkat kepuasan pengguna, peneliti menggunakan metode *System Usability Scale (SUS)* yang akan digunakan dalam bentuk kuesioner. Pada penelitian sebelumnya yang dilakukan oleh Dian Megah Sari ²⁹ metode *System Usability Scale (SUS)*, juga digunakan untuk mengevaluasi aplikasinya dengan mengukur tingkat kepuasan pengguna [3].

H. Jadwal Penelitian

Tabel I.1 : Jadwal penelitian

| Kegiatan | Minggu 1 | | | | | | | Minggu 2 | | | | | | | Minggu 3 | | | | | | | Minggu 4 | | | | | | |
|-----------------------------|----------|---|---|---|---|---|---|----------|---|---|---|---|---|---|----------|---|---|---|---|---|---|----------|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Pembuatan Sistem | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Analisis Efektivitas Sistem | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Evaluasi | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Penulisan Laporan | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

I. Sistematika Laporan

Adanya sistematika untuk penelitian ini yang merupakan paparan mengenai bab yang ada pada laporan skripsi dan berisi penjelasan pada setiap babnya, berikut adalah rancangan laporan skripsi:

BAB I PENDAHULUAN

Bab pertama pendahuluan berisi mengenai latar belakang, identifikasi, rumusan, batasan masalah, tujuan, manfaat dan kegunaan, metode penelitian yang digunakan

BAB II TINJAUAN PUSTAKA

Bab kedua tunjauan pustaka berisi mengenai landasan teori, kajian pustaka, dan desain yang menjelaskan mengenai dasar-dasar teori yang berhubungan dengan penelitian yang dilakukan, dan rancangan desain sistem awal

BAB III ANALISA DAN DESAIN SISTEM

Bab ketiga analisa dan desain sistem berisi mengenai analisa data yang telah didapatkan oleh penulis, dan perancangan desain sistem menurut hasil analisa data tersebut

BAB IV HASIL DAN EVALUASI

Bab keempat menjelaskan mengenai kinerja sistem yang dibuat oleh penulis di lapangan, hasil tersebut akan diolah pada evaluasi

BAB V PENUTUP

Bab terakhir berisi mengenai kesimpulan dan saran oleh penulis untuk penelitian selanjutnya

2 BAB II
TINJAUAN PUSTAKA

A. Landasan Teori

1. Kecerdasan Buatan (*Artificial Intelligence*)

Menurut yang terkutip dari artikel berjudul “*Causability and explainability of artificial intelligence in medicine*” ditulis oleh Holzinger A, dkk. *Artificial Intelligence (AI)*, merupakan bidang ilmu komputer tertua dan sangat luas menangani semua aspek yang meniru fungsi kognitif untuk pemecahan masalah di dunia nyata dan membangun sistem yang belajar dan berpikir layaknya manusia. Oleh karena itu, sering disebut kecerdasan buatan [4].

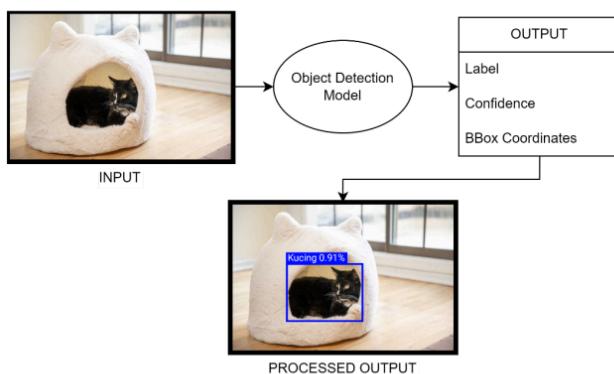
Tujuan awal dari pembuatan kecerdasan buatan adalah untuk membantu tugas-tugas manusia, namun seiring berjalannya waktu perkembangan kecerdasan buatan dapat juga menggantikan tugas-tugas manusia [5]. Di jaman sekarang *AI* atau kecerdasan buatan dipelajari secara luas pada perangkat *mobile*, dimana bertujuan untuk meningkatkan kualitas komputasi dan membuka kemungkinan untuk aplikasi baru seperti *face unlock*, *speech recognition*, *natural language processing*, *virtual reality*, dan masih banyak lagi. Namun kecerdasan buatan membutuhkan kapabilitas komputasi yang sangat besar untuk melakukan *training* dan *learning* yang kompleks [6].

Dalam konteks kecerdasan buatan, *Machine Learning (ML)* dianggap sebagai proses pembelajaran yang terinspirasi oleh pengalaman manusia. Ini memungkinkan sistem untuk memperbaiki analisisnya seiring waktu, menggunakan algoritma komputasi. Agar dapat membuat rekomendasi atau keputusan secara mandiri, algoritma ini menggunakan sekumpulan masukan dan keluaran *data* yang besar untuk mengenali pola sehingga mesin dapat belajar secara efektif. Setelah pengulangan dan modifikasi algoritma yang memadai, mesin dapat menerima masukan dan memprediksi keluaran [7].

2. Deteksi Objek (*Object Detection*)

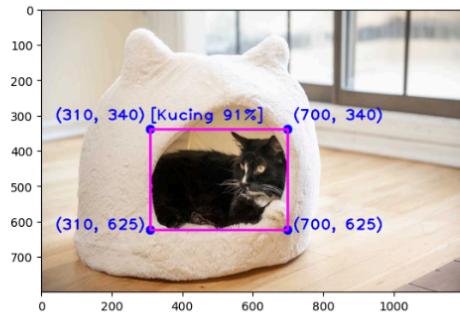
Deteksi objek, sebagai salah satu masalah paling mendasar dan menantang dalam *computer vision*, telah mendapat perhatian besar dalam beberapa tahun terakhir. Evolusi teknologi yang pesat dalam pendekslsian objek dan dampaknya yang besar terhadap bidang visi komputer telah diakui [8].

Deteksi objek merupakan kecerdasan buatan yang dibuat layaknya ketika manusia mengenali sebuah benda. Perbedaan antara kecerdasan buatan deteksi objek dan kecerdasan manusia dalam mengenali objek adalah, kecerdasan buatan memerlukan sebuah mesin agar bisa berjalan dan memerlukan komputasi yang kompleks agar dapat mengenali sebuah benda [9].



Gambar 2.1 : Contoh implementasi Object Detection Model

Gambar 2.1 merupakan implementasi model *Object Detection*, dapat dilihat data yang dibutuhkan sebagai *input* merupakan sebuah gambar, data tersebut akan diproses menggunakan model dan mengeluarkan *output* titik koordinasi *Bounding Box*, *Confidence*, dan *Label*. Setelah mendapatkan hasil data *output* dari model, hasil tersebut diolah kembali dengan gambar dari *input* sehingga menghasilkan gambar dengan baris yang melingkupi objek, nama objek, dan nilai presentase kepercayaan model.



Gambar 2.2 : Pengolahan output data dari Object Detection Model

Gambar 2.2 merupakan bagaimana *output* yang di dapat dari model diolah kembali. Dapat dilihat terdapat 4 titik yang melingkupi objek, ke-empat titik tersebut merupakan *Bounding Box* yang digunakan untuk membuat baris-baris. Isi dari *bounding box* yaitu x_{min} , y_{min} , $width$, dan $height$, menurut gambar 2.2 maka nilainya adalah 310, 340, 700, dan 625. Untuk *Label* dan *Confidence* memiliki nilai “Kucing”, dan “91%” nilai tersebut didapatkan dari hasil prediksi model *Object Detection*.

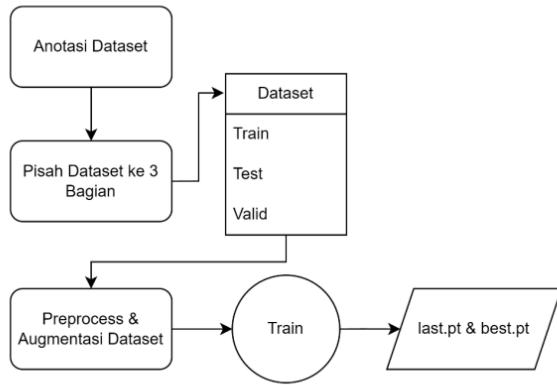
3. YOLO (*You Only Look Once*)

Salah satu kecerdasan manusia adalah kemampuan dalam mengenali suatu objek yang ada di sekitarnya. Dari berbagai perkembangan teknologi saat ini menunjang sebuah mesin untuk dapat belajar seperti manusia dalam mengenali sebuah objek. Mesin sendiri membutuhkan kecerdasan buatan untuk dapat mengenali dan mengklasifikasikan sebuah objek citra digital. Salah satu algoritma yang sering digunakan dalam bidang deteksi objek adalah *YOLO (You Only Look Once)* [10].

Menurut artikel didalam jurnal “*Machines 2023*” yang berjudul “*YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection*” dan ditulis oleh Muhammad Hussain. *YOLO* yang merupakan singkatan dari *You Only Look Once* pertama diperkenalkan pada tahun 2015 di sebuah paper yang dibuat oleh Joseph Redmon yang berjudul “*You Only Look Once: Unified*,

Real-Time Object Detection". Konsep "unified" yang digunakan *YOLO* memungkinkan prediksi simultan dari beberapa *bounding box* maupun probabilitas *class*, dimana meningkatkan kecepatan dan akurasi. *YOLO* terus berkembang dengan pesat, ke-efektifitas dan potensi yang dimiliki oleh konsep *YOLO* telah dikembangkan lebih lanjut oleh beberapa penulis, hingga sekarang tahun 2023 telah dirilisnya *YOLOv8* [11].

YOLOv8 memperkenalkan pengoptimalan tambahan dan modul baru untuk meningkatkan akurasi dan efisiensi. *YOLOv8* sendiri adalah evolusi dari versi sebelumnya yang memperkenalkan beberapa perbaikan dan kelebihan baru. Salah satu kelebihan utama *YOLOv8* adalah peningkatan performa dan akurasi dalam deteksi objek [12].



Gambar 2.3 : Proses pelatihan model Object Detection YOLOv8

Sebagai langkah pertama dari pelatihan model *YOLOv8* yaitu anotasi *dataset*. Anotasi *dataset* atau bisa juga disebut *Image Labelling*, merupakan sebuah proses penambahan informasi pada gambar-gambar (*dataset*) yang bertujuan untuk membantu mesin dalam "melihat" dari detail spesifik dalam sebuah gambar yang sudah memiliki *label* [13].



Gambar 2.4 : Contoh proses anotasi dataset menggunakan Roboflow

Terdapat banyak aplikasi yang dapat digunakan untuk memudahkan proses anotasi *dataset*, salah satunya adalah *Roboflow*. Berikut komparasi beberapa aplikasi anotasi yang telah dilakukan oleh *Prakash, D* pada jurnal miliknya yang berjudul “*A Comparative Analysis of Object Identification Labelling Platforms: Basketball Perspective*” [13].

Tabel 2.1 : Komparasi beberapa aplikasi untuk anotasi

| Aplikasi | Kelebihan | Kekurangan |
|----------|--|---|
| Roboflow | <ul style="list-style-type: none"> - Gratis dengan ketentuan - Fitur bebayar yang memiliki harga terjangkau - Menyediakan <i>Pre-build models</i> untuk <i>active learning</i> - Ideal untuk aplikasi <i>real-time</i> tanpa bergantung pada infrastruktur <i>cloud</i> - Komunitas dan <i>support team</i> yang aktif dalam pembecahan masalah dan pertanyaan-pertanyaan | <ul style="list-style-type: none"> - Maksimal <i>dataset</i> berjumlah 1000 gambar setelah uji coba - <i>Tools</i> untuk anotasi yang terbatas - Integrasi yang terbatas - Fitur yang sedikit pada versi gratis |

| | | |
|---------------|--|---|
| Makesense.ai | <ul style="list-style-type: none"> - Gratis dan bersifat <i>open-source</i> - Pembuatan proyek, dan kolaborasi yang tidak terbatas - <i>Format output</i> yang fleksibel (<i>YOLO, CSV, VOC, XML</i>, dll) - <i>Tools</i> anotasi yang serbaguna | <ul style="list-style-type: none"> - Terdapat banyak fitur lebih lanjut pada versi berbayar - Pada versi gratis <i>dataset</i> yang tersimpan memiliki aktifitas waktu terbatas hingga 30 hari - Beberapa tugas yang kompleks memerlukan koneksi internet yang cepat |
| SentiSight.ai | <ul style="list-style-type: none"> - Menggunakan metode pembayaran “<i>pay-as-you-go</i>”, dimana pengguna hanya membayar model yang digunakan saja - <i>Interface</i> yang <i>User-friendly</i> - Mendukung beragam tipe data - Model <i>deployment</i> yang <i>offline</i> | <ul style="list-style-type: none"> - Fungsional yang krusial sangat terbatas pada versi gratis - Memungkinkan terdapat kekurangan dalam fitur lebih lanjut untuk pemrosesan teks dan analisis gambar. - Dapat memiliki harga yang mahal untuk tipe proyek besar |
| Labelbox | <ul style="list-style-type: none"> - Mendukung beragam tipe data, dan berjumlah besar - <i>Tools</i> untuk anotasi data yang kokoh - Fitur lebih lanjut seperti penjelasan model, <i>ML framework integration</i>, dll - Menawarkan banyak fitur untuk keamanan data | <ul style="list-style-type: none"> - Hanya menyediakan versi berbayar - Dapat memiliki harga yang sangat mahal untuk proyek kecil atau pengguna individual - Beberapa fitur memerlukan pemahaman teknis - Tidak untuk pemula |

| | | |
|---------------|--|---|
| SuperAnnotate | <ul style="list-style-type: none"> - Menyediakan <i>tools</i> untuk kolaborasi seperti <i>shared workspace</i> - Komunitas, dan tim pendukung yang aktif dan responsif - Fitur lebih lanjut dan <i>workflows</i> yang fleksibel | <ul style="list-style-type: none"> - Hanya menyediakan versi berbayar - Fokus ke spesifik tipe data - Fitur lebih lanjut memerlukan pemahaman teknis |
|---------------|--|---|

Seperti yang dipaparkan pada tabel 2.1 bahwa setiap aplikasi memiliki keuntungan dan kekurangan masing-masing. Jika mencari aplikasi yang mudah digunakan atau *beginner friendly* dengan anggaran terbatas, maka *Roboflow* merupakan pilihan yang bagus. Namun perlu diketahui bahwa *dataset* memiliki batasan tidak lebih dari 10,000 gambar pada versi gratisnya dan 1,000 gambar setelah uji coba pemula [13].

Dengan menggunakan *Roboflow* proses pemisahan, *preprocessing*, dan augmentasi *dataset* dimudahkan. Berikut penjelasan beberapa fitur didalam *Roboflow*:

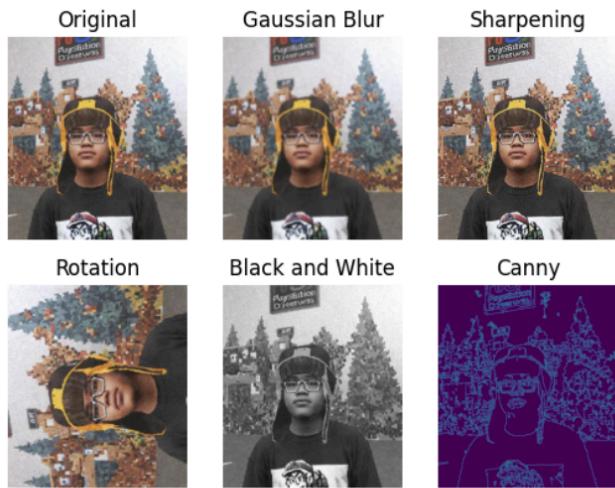
- a. *Auto-split*, fitur ini mengotomatisasi proses pemisahan *dataset*, dan konfigurasi yang dapat disesuaikan dengan mengganti jumlah presentase. Contoh: 70% train, 20% test, 20% valid.
- b. *Generate version*, fitur ini juga mengotomatisasi proses *preprocessing*, dan augmentasi *dataset*. Konfigurasi juga didukung dimana dapat menyesuaikan keinginan pengguna.

4. OpenCV

Nama “*OpenCV*” sendiri diambil dari “*open source computer vision*”, berawal dari *Intel Research Lab* dimana pada saat itu berinisiatif untuk pendekatan lanjutan dalam membangun aplikasi *CPU-intensive*. Seperti yang dipresentasikan namanya *OpenCV* merupakan *library computer vision* bersifat *open-source*, selain itu merupakan salah satu *library* populer yang digunakan untuk implementasi aplikasi *computer*

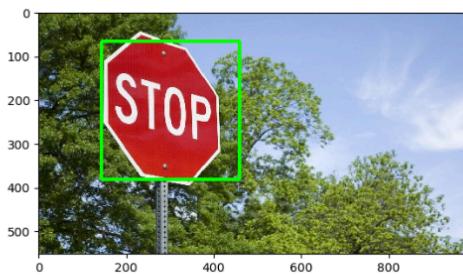
vision, dimana fungsi utamanya adalah untuk memanipulasi, memproses dan menganalisis gambar. [14].

Terdapat lebih 2,500 algoritma optimasi yang bisa digunakan untuk memanipulasi sebuah gambar. *OpenCV* mendukung berbagai bahasa pemrograman seperti *C*, *C++*, *Python*, *Java*, dan *PHP*. Beberapa aplikasi *OpenCV* yang bisa diterapkan, diantaranya adalah Identifikasi dan Pengenalan Objek, Deteksi dan Pengenalan Wajah, dan masih banyak lagi aplikasi yang menerapkan *library* ini [15].



Gambar 2.5 : Contoh hasil manipulasi gambar menggunakan OpenCV

Gambar 2.6 merupakan hasil dari penggunaan *Image Manipulation* menggunakan *OpenCV*. Gambar dan video yang digunakan sebagai *input* dalam aplikasi *computer vision* dipresentasikan sebagai *matrix* untuk menyimpan berbagai detail sebuah gambar yaitu lebar, tinggi, kedalaman, *channel* (*BGR* atau *RGB*), dan yang lainnya [14].



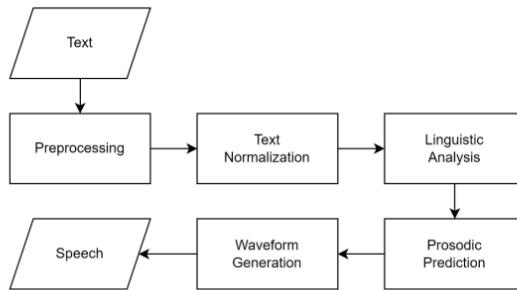
Gambar 2.6 : Image Recognition menggunakan OpenCV

5. TTS (Text-to-Speech)

22

Menurut sebuah artikel berjudul “*Conventional and contemporary*

approaches used in text to speech synthesis: a review” yang ditulis oleh Kaur, N dan Singh, P pada tahun 2022. TTS atau yang dapat disebut *Text-to-Speech*, merupakan kemampuan sistem untuk menghasilkan suara yang menyamai suara manusia dari teks tertulis [16].



Gambar 2.7 : Alur sistem TTS

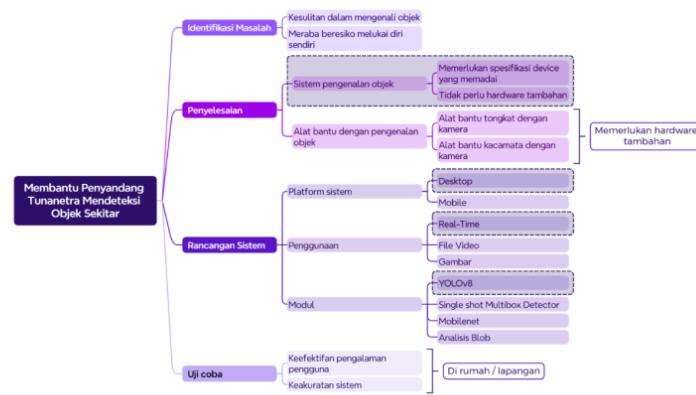
Dalam konversi *Text-to-Speech*, teks *input* teranalisa kemudian teks tersebut terkonversi ke versi suara untuk diputar. Fungsi ini memiliki efektivitas ketika seseorang mengerti bahasa asing tetapi tidak mengerti bagaimana pengucapannya. Selain itu fungsi ini juga sangat membantu kepada orang-orang yang memiliki gangguan pada pengliatannya dalam membaca [17].

GTTS (Google Text-to-Speech) merupakan *library* pada bahasa pemrograman *python* yang berfungsi untuk mengonversi sebuah teks ke *audio file*. Hasil dari proses *GTTS* adalah berupa *file mp3*, untuk proses

pembuatannya membutuhkan koneksi internet agar terhubung ke *Google Translates text-to-speech API* [18].

B. Kajian Pustaka

Berikut kajian pustaka dengan isi rangkuman-rangkuman dari peneliti sebelumnya yang memiliki latar belakang yang relevan dengan penelitian ini. Gambar 2.2 merupakan *Mind Map Diagram* melampirkan beberapa poin, dan tabel 2.1 merupakan rangkuman dari penelitian sebelumnya yang memiliki latar belakang relevan dengan penelitian ini.



Gambar 2.8 : Mind map diagram kajian pustaka

Pada gambar 2.8 merupakan *Mind Map Diagram* melampirkan beberapa poin dan keputusan dari penulis. “Identifikasi Masalah” dimana merangkum identifikasi masalah yang ada, “Penyelesaian” merupakan ide-ide yang dapat digunakan untuk mengatasi masalah yang dihadapi. Setelah menemukan ide solusi tersebut, mulai merancang sistem dimana berada di poin “Rancangan Sistem”, dan untuk poin “Uji coba” dilakukan untuk memaksimalkan sistem yang dirancang pada saat implementasi di lapangan.

Pada tabel 2.2 merupakan rangkuman dari penelitian sebelumnya yang memiliki masalah relevan. Terdapat 5 penelitian yang dirangkum dengan berbagai solusi yang ditawarkan.

Tabel 2.2 : Penelitian relevan

| No | Nama | Tahun | Judul | Objek Penelitian | Pembahasan | Kebaharuan |
|----|-----------------------------|-------|--|--|--|---|
| 1 | Fuady, Nehru, dan Anggraeni | 2020 | Deteksi Objek Menggunakan Metode Single Shot Multibox Detector Pada Alat Bantu Tongkat Tunanetra Berbasis Kamera | Alat bantu tongkat | Penyandang Tunanetra mengalami kesulitan untuk bermavigasi dikarenakan keterbatasan penginderaan yang dialami | Dirancangnya alat bantu tongkat tunanetra untuk pendekslsian objek berbasis kamera |
| 2 | Risaldi dan Utaminingrum | 2019 | Deteksi Objek Penghalang secara Real Time berbasis Aplikasi Mobile dengan Metode Gray Level Co-Occurrence Matrix dan K-neares Neighbor bagi Penyandang Tunanetra | Perangkat mobile | Tunanetra memiliki suatu kondisi indra penglihatannya. Dengan pesatnya perkembangan teknologi, penggunaan perangkat mobile mampu melakukan banyak proses | Dibangunnya sistem dengan menggunakan kamera dari perangkat mobile sebagai pengganti dari indra penglihatan |
| 3 | Pradana dan Hajianto Sula | 2023 | Pengembangan Aplikasi Pendekripsi Objek Untuk Tunanetra Menggunakan Operator Tepi Canny Dengan Library OpenCV Berbasis Android | Botol, Perangkat mobile, laptop, mouse, dan keyboard | Berdasarkan survei yang dilakukan oleh Biro Pusat Statistik Provinsi Jawa Timur. Penyandang Tunanetra tidak dapat melakukan aktivitas secara mandiri tanpa | Dibuatnya aplikasi pendekripsi objek untuk penyandang tunanetra dan anak usia dini berbasis android |

| | | | | | | |
|--|--|--|--|--|--|--|
| | | | | | ada bantuan orang lain atau menggunakan tongkat bergaris merah | |
|--|--|--|--|--|--|--|

Tabel 2.2 : Tabel penelitian relevan (Lanjutan)

| No | Nama | Tahun | Judul | Objek Penelitian | Pembahasan | Kebaharuan |
|----|-------------------------------|-------|---|--|--|--|
| 4 | Al Kadafi dan Utaminingsrum | 2018 | Deteksi Objek Penghalang Secara Real-Time Berbasis Mobile Bagi Penyandang Tunanetra Menggunakan Analisis Blob | Perangkat mobile | Dalam ilmu computer vision sangat memungkinkan para penyandang tunanetra mampu melakukan aktivitas berjalan seperti orang normal pada umumnya | Dibangun sebuah sistem berbasis computer vision yang diterapkan pada sebuah perangkat mobile untuk mendeteksi halangan secara real-time ketika penyandang tunanetra berjalan didalam ruangan |
| 5 | Ariyani, Nugroho, dan Mubarok | 2022 | Alat Bantu Pendekripsi Objek Untuk Tuna Netra Berbasis AI Mobilenet Pada Raspberry Pi 3B | Raspberry Pi, dan sensor jarak HC SR04 | Saat ini teknologi alat bantu untuk penyandang disabilitas telah berkembang pesat. Salah satunya alat bantu untuk tunanetra dengan memanfaatkan kamera yang memiliki fungsi sebagai mata | Dirancangnya alat pendekripsi objek dengan cara penggunaan layaknya kacamata |

ANALISA DAN DESAIN SISTEM**C. Analisa Sistem****1. Analisa Sistem Lama**

Sistem pengenalan objek untuk penyandang tunanetra sebelumnya, mungkin memerlukan langkah manual dan waktu yang lama. Berikut poin-poin analisa sistem lama:

- a. *Preprocessing* secara manual. Data gambar mungkin harus diolah secara manual sebelum digunakan untuk *training*. Ini bisa termasuk penyesuaian kualitas gambar, pengaturan kontras, dan pemotongan gambar agar objek menjadi lebih terlihat.
- b. Pelatihan *Custom Model*. Pembuatan dan pelatihan model untuk pengenalan objek bisa sangat rumit. Metode tradisional mungkin melibatkan penggunaan teknik pengolahan citra klasik atau deteksi objek yang membutuhkan pengetahuan khusus dan banyak waktu.
- c. Menggunakan versi algoritma yang lama. *YOLOv8* merupakan versi terbaru dan yang stabil saat ini, terdapat berbagai fitur baru hingga optimasi algoritma yang mungkin dapat membantu.

Dengan menggunakan algoritma *YOLOv8* mengenali objek dalam gambar akan lebih cepat dan akurat, mengurangi ketergantungan pada langkah-langkah manual seperti yang dijelaskan di atas.

2. Analisa Sistem Yang Diusulkan

- a. Analisa kebutuhan fungsi

Berikut merupakan poin-poin dalam pembuatan sistem pengenalan objek untuk penyandang tunanetra:

- 1) Pengenalan objek secara *real-time*. Sistem harus mampu mendekripsi dan mengidentifikasi objek dalam waktu nyata untuk memberikan respon cepat kepada pengguna
- 2) Akurasi tinggi. Merupakan bagian penting dalam sistem untuk memiliki tingkat keakuratan yang tinggi dalam mengenali objek, mengingat tujuan pengguna dari sistem ini yaitu penyandang

tunantara yang mengandalkan informasi visual untuk berinteraksi dengan lingkungan mereka.

- 3) Pelaporan objek yang terdeteksi. Sistem harus mampu melaporkan objek yang terdeteksi dengan jelas dan tepat kepada pengguna, dapat melalui suara atau getaran.
- 4) Beradaptasi dengan lingkungan. Sistem harus mampu beradaptasi ketika lingkungan pengguna minim dengan pencahayaan, maka sistem akan otomatis menyalaikan senter dari perangkat. Sehingga sistem dapat selalu membantu pengguna setiap saat.

b. Analisa kebutuhan data

Dalam pembuatan sistem pengenalan objek dibutuhkan sebuah data *input* yang akan diproses. Data ini berupa sebuah gambar dari hasil tangkapan kamera *smartphone* atau yang lainnya, selain data berupa gambar sistem ini dapat juga bekerja dengan *video* dengan cara yaitu memproses setiap *frame* pada *video*, lalu hasilnya akan dimunculkan *frame by frame*. Seperti contoh terdapat *video* yang berdurasi 1 menit dan memiliki 24 *fps* (*frames per second*), maka pada *video* tersebut memiliki 1,440 *frame* hasil tersebut didapat dari penjumlahan:

$$\text{total frame} = \text{frame per second} * \text{durasi video dalam detik}$$



Gambar 3.1 : Contoh Input Data



Gambar 3.2 : Contoh Output Data

Berikut contoh *input* dan *output* dari penggunaan *YOLOv8* pada umumnya. Terlihat pada gambar 3.1 yang merupakan sebuah gambar, ini akan diproses sehingga sistem dapat mengenali objek yang ada pada gambar tersebut. Setelah menyelesaikan proses tersebut sistem akan mengeluarkan *output* berupa teks dengan *bounding box* yang akan dijelaskan pada penjelasan gambar 3.2.

Pada gambar 3.2 merupakan contoh *output* data, terlihat bahwa terdapat garis berbentuk kotak berwarna biru yang mengelilingi objek (kucing). Baris berbentuk kotak ini dinamakan *bounding box* yang memiliki tujuan mengetahui letak objek (kucing) berada dalam sebuah gambar. Selain itu terdapat teks “Kucing 0.91%” merupakan peng gabungan dari 2 teks yaitu label (“Kucing”) dan presentase *confidence* (“0.91%”). Teks label bertujuan untuk membedakan sebuah objek dengan objek lainnya. Presentase *confidence* bertujuan untuk mengetahui seberapa yakin sebuah sistem mengenali objek tersebut dalam bentuk nilai presentase 0% hingga 1%

Pada penelitian ini penulis merancang sistem agar dapat bekerja secara *real-time*. Untuk prosesnya hampir sama seperti proses menggunakan *video*, perbedaannya adalah menggunakan beberapa *library* tambahan seperti *OpenCV* agar mendapatkan gambar dari kamera.

Library GTTS digunakan untuk konversi teks ke suara yang merupakan *output* dari sistem ini. Teks yang dirubah menjadi suara ini, berisi mengenai deksripsi dari hasil deteksi objek antara lain yaitu nama objek, dan banyak objek.

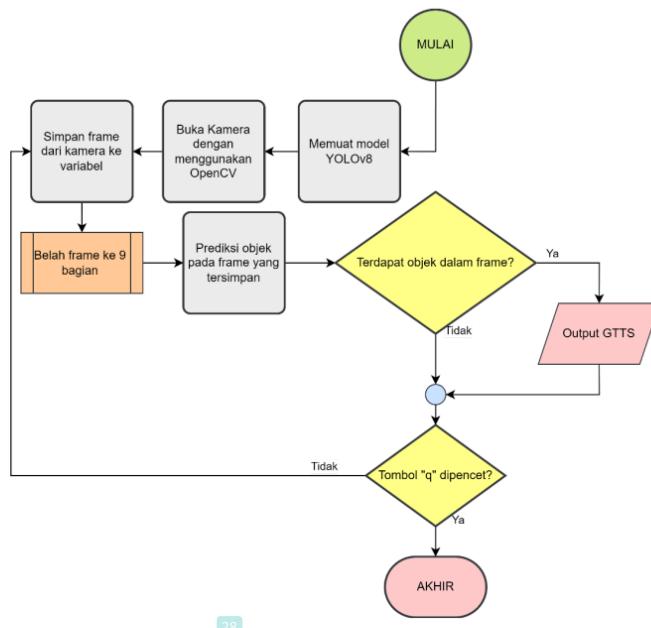
c. Analisa kebutuhan perangkat

Alat yang dapat dengan mudah dibawa kemana-mana, dan sudah memiliki kamera, senter, dan speaker bawaan. Peniliti mengusulkan *smartphone* adalah alat yang ideal untuk sistem yang akan dikembangkan.

D. Desain Sistem (Arsitektur)

Desain sistem yang digunakan pada penelitian ini berbentuk *flowchart*.

Flowchart merupakan *diagram* yang menjelaskan alur dari sebuah sistem mulai dari awal hingga akhir. Pada gambar 3.3 merupakan desain *flowchart* yang akan digunakan dalam sistem



28
Gambar 3.3 : Flowchart sistem

Berikut penjelasan dari gambar 3.3 yang merupakan *flowchart* sistem penulis. Seperti yang ada di gambar, proses pertama dari sistem yaitu memuat model yang sudah disediakan dari *YOLOv8*. Setelah itu membuka kamera dengan menggunakan *library OpenCV* dan menyimpan *frame* gambarnya ke variabel. Proses selanjutnya yaitu memanggil fungsi untuk membelah *frame* ke 9, jika melihat gambar 3.4 bagian-bagian ini merupakan “kiri atas”, “atas”, “kanan atas”, “kiri”, “depan”, “kanan”, “kiri bawah”, “bawah”, dan yang terakhir “kanan bawah”. Pembelahan *frame* ini bertujuan untuk mendapatkan posisi dari objek didalam *frame*, sehingga pengguna dapat bennavigasi.

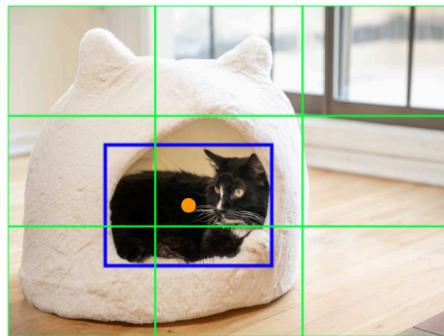
Selanjutnya adalah memprediksi keberadaan objek dari gambar yang sudah disimpan dari variabel tadi. Sebelum proses mengeluarkan *output* berupa suara kata-kata deskripsi objek, terdapat kondisi dimana jumlah objek atau nama objek berbeda dari sebelumnya jika benar maka outputkan suara, dan jika tidak maka melanjutkan proses. Pada akhir dari sistem terdapat juga

sebuah kondisi dimana jika tombol “q” dipencet maka sistem akan berakhir, dan jika tidak maka sistem berlanjut.



Gambar 3.4 : Implementasi fungsi pembelahan frame

Gambar 3.4 merupakan bagaimana sistem akan membelah frame ke 9 bagian. Karena sistem hanya mengetahui panjang, lebar, channel warna, dan koordinasi sebuah gambar maka penulis merancang fungsi ini agar sistem dapat mengetahui arah-arah. Pada bagian kiri adalah “full display” frame dimana didapatkan dari tangkapan kamera sistem. Lalu setelah dibelah ke beberapa bagian, akan menjadi seperti di sebelah kanan.



Gambar 3.5 : Gambaran fungsi arah

Gambar 3.5 merupakan gambaran bagaimana sistem akan mengetahui letak dari objek yang terdeteksi. Dapat dilihat sistem membedakan beberapa bagian yang akan diperjelas pada tabel 3.1

Tabel 3.1 : Bagian-bagian contoh gambar

| Warna | Deskripsi |
|--------|---|
| Hijau | Ruang lingkup dari ke 9 bagian yang dibelah dari hasil fungsi |
| Biru | <i>Bounding Box</i> yang merupakan hasil dari prediksi model YOLOv8 |
| Oranye | Titik tengah dari <i>Bounding Box</i> |

Seperti yang dipaparkan pada tabel 3.1 bahwa garis berwarna hijau menandakan area-area sisi yang telah di bawah, garis berwarna biru adalah bounding box yang didapatkan melalui prediksi model YOLOv8, dan titik berwarna oranye merupakan titik tengah bounding box. Dan jika melihat gambar 3.4 lagi, posisi titik tengah bounding box berada di area “Front” atau depan. Dengan gambar 3.6 akan dijelaskan bagaimana sistem akan bekerja pada saat terdapat lebih dari 1 objek



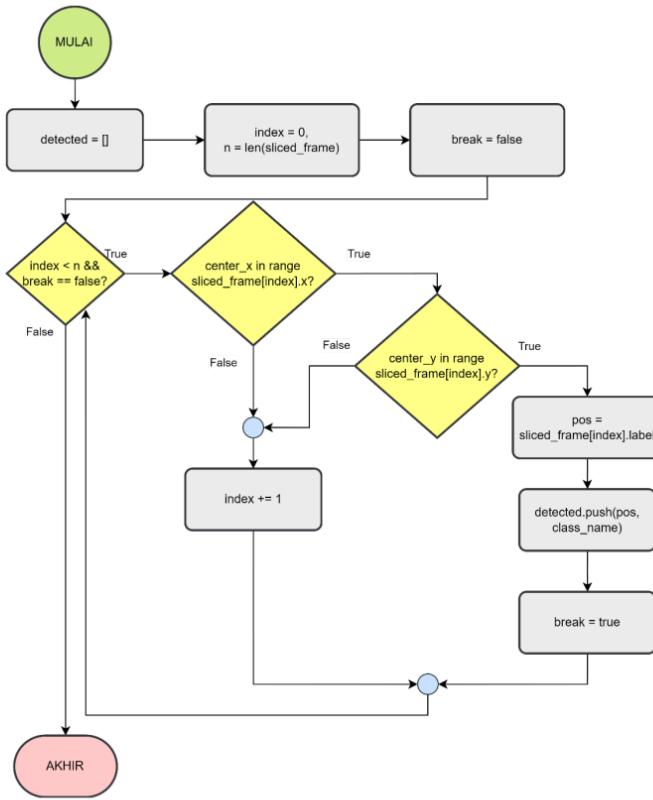
Gambar 3.6 : Gambaran fungsi arah dengan beberapa objek

Tabel 3.2 : Hasil deteksi dari gambaran

| Area Sisi | Objek Terdeteksi |
|------------|------------------|
| Kiri atas | Tidak ada |
| Atas | Monitor |
| Kanan atas | Tidak ada |
| Kiri | Laptop |
| Depan | Keyboard |

| Kanan | Bunga |
|------------|-----------|
| Kiri Bawah | Tidak ada |
| Bawah | Tidak ada |
| Kanan | Tidak ada |
| Bawah | |

Gambar 3.6 merupakan gambaran bagaimana sistem bekerja pada saat terdapat lebih dari 1 objek terdeteksi, dan tabel 3.2 adalah pemaparan dari objek-objek yang terdeteksi dipisahkan dengan setiap area sisinya. Terlihat bahwa dengan menggunakan fungsi pembelahan frame ke beberapa bagian, sistem dapat lebih mudah menentukan letak objek-objek yang terdeteksi. Untuk penentuan letak objek diperlukannya sebuah algoritma yaitu:

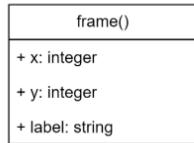


Gambar 3.7 : Flowchart algoritma penentuan letak objek

Di atas merupakan algoritma yang akan digunakan untuk sistem dapat menentukan letak objek, menurut area *frame* yang sudah dibelah. “detected” merupakan variabel untuk menyimpan kumpulan posisi dan nama label dari objek yang terdeteksi. Untuk isi dari “pos” adalah area *frame* yang melingkupi titik tengah *bounding box* (“kiri atas”, “atas”, “kanan atas”, dll. Dapat dilihat pada gambar 3.4). Dan “class_name” merupakan nama label dari objek yang terdeteksi (contoh: “kucing”).

Variabel “center_x” dan “center_y” menyimpan nilai dari titik koordinasi *bounding box* objek yang terdeteksi, “x” menyimbolkan sisi *horizontal*, dan “y” sisi *vertical*. “sliced_frame” adalah variabel yang

menyimpan kumpulan objek “frame” yang memiliki atribut titik koordinasi *horizontal* (“x”), *vertical* (“y”), dan label dari seluruh area *frame* yang sudah dibilah ke 9 bagian. Berikut paparan dari struktur objek frame dan kumpulan label area frame.



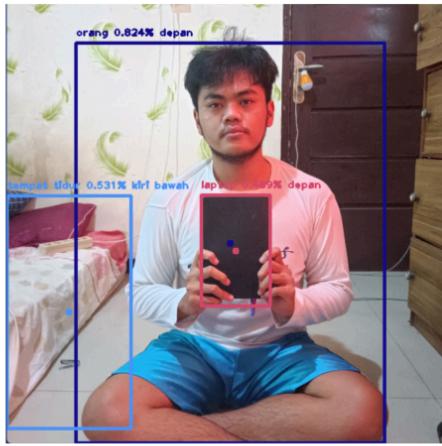
Gambar 3.8 : Struktur objek frame

Tabel 3.3 : Kumpulan label area frame

| Inde x | Label |
|-----------|------------------|
| 0 | “Kiri Atas” |
| 1 | “Atas” |
| 2 | “Kanan Atas” |
| 3 | “Kiri” |
| 4 | “Depan” |
| 5 | “Kanan” |
| 6 | “Kiri Bawah” |
| 7 | “Bawah” |
| 8 | “Kanan Bawah” |

A. Hasil**1. Hasil simulasi sistem**

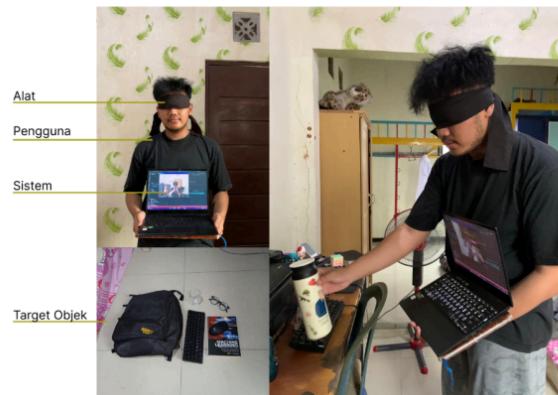
Sebelum melakukan implementasi sistem, peneliti melakukan simulasi atau uji coba demi mengetahui apakah sistem dapat bekerja sesuai rancangan. Dengan hasil, sistem mampu mengenali objek dengan input gambar atau video dari kamera, dan memiliki rata-rata 30 fps (*frames per second*). Selain mampu mengetahui arah objek yang dikenali, sistem juga dapat mengeluarkan suara sebagai pemberitahuan pengguna. Jika melihat pada gambar 4.1 suara yang dikeluarkan oleh sistem yaitu “Pada bagian depan terdapat orang, laptop. Pada bagian kiri bawah terdapat tempat tidur”.



Gambar 4.1 : Hasil output sistem

Pada gambar 4.1 sistem masih memiliki kekurangan dalam mengenali objek, terlihat bahwa objek yang terdeteksi oleh sistem yaitu (“tempat tidur”, 0,531%), (“orang”, 0,824%), dan (“laptop”, 0,489), namun sebenarnya “laptop” adalah “buku”. Dengan adanya kekurangan ini peneliti menguji sistem menggunakan *Confusion Matrix* agar mengetahui performa sistem dalam mengenali objek lebih lanjut.

Simulasi sistem pada penelitian ini dilakukan di ruangan kamar, guna mereplika skenario penggunaan dalam sehari-hari. Pengguna diharuskan mendapatkan objek yang disebutkan oleh pihak lain, dengan kondisi pengguna hanya dapat memanfaatkan fungsi dari sistem, peletakan objek secara acak oleh pihak selain pengguna. Keakuratan pengguna direkam sebagai penilaian uji coba. Karena sistem belum dapat mendeteksi halangan seperti tembok. Pihak lain diharuskan memberitahu pengguna jika terdapat halangan di dekat arah jalan pengguna. Berikut gambar pelaksanaan uji coba implementasi peneliti:



Gambar 4.2 : Pelaksanaan Uji Coba Implementasi Sistem

Pada gambar 4.2 terlihat pelaksanaan uji coba sistem memiliki 4 komponen utama yaitu: alat, pengguna, sistem, target objek, dan pihak lain. Dimana “alat” ini bertujuan untuk mereplika kondisi “penyandang tunanetra menggunakan sistem”. Lalu terdapat “pengguna”, dengan menggunakan komponen “alat”, “pengguna” dapat bergiliran dengan “pihak lain”. Setelah itu komponen inti yaitu “sistem”, tugas dari komponen ini adalah memberitahu letak dan objek apa yang berada di area “pengguna”. Komponen “target objek” menjadi tujuan pengguna untuk diambil. Dan yang terakhir yaitu “pihak lain”, bertugas untuk merekam keakuratan “pengguna”, menempatkan “target objek” secara acak, dan

memberitahu “pengguna” jika terdapat halangan. Berikut hasil dari uji coba implementasi sistem:

Tabel 4.1 : Hasil Uji Coba Implementasi Sistem

| Putaran ke- | Target | Diambil | Tempat | Terdeteksi |
|-------------|-----------------|-----------------|--------|-----------------|
| 1 | Botol Minuman A | Botol Minuman A | Meja | Botol |
| 2 | Keyboard | Keyboard | Meja | Papan Ketik |
| 3 | Tas | Tas | Lantai | Ransel |
| 4 | Cangkir | Botol Kapsul | Meja | Cangkir |
| 5 | Cangkir | Cangkir | Meja | Cangkir |
| 6 | Mouse | Mouse | Lantai | Mouse |
| 7 | Botol Minuman A | - | Kasur | Tempat Tidur |
| 8 | Tas | - | Kasur | Tempat Tidur |
| 9 | Handphone | Handphone | Meja B | Telepon Selular |
| 10 | Keyboard | Keyboard | Meja B | Papan Ketik |
| 11 | Botol Minuman B | - | Meja | Vas |
| 12 | Handphone | Handphone | Meja | Telepon Selular |
| 13 | Buku | - | Kasur | Tempat Tidur |
| 14 | Botol Minuman B | - | Kasur | Vas |
| 15 | Buku | Buku | Kasur | Tempat Tidur |
| 16 | Botol Minuman B | - | Meja B | Vas |
| 17 | Mouse | Mouse | Meja B | Mouse |
| 18 | Handphone | Handphone | Kasur | Telepon Selular |
| 19 | Buku | Buku | Lantai | Buku |
| 20 | Botol Minuman A | Botol Minuman A | Meja B | Botol |

Dari hasil uji coba implementasi sistem (tabel 4.1) yang dilakukan oleh peneliti sebanyak 20 putaran, dengan beberapa objek yang digunakan. Pada isi dari tabel tersebut terlihat bahwa terdapat beberapa putaran ada yang kosong dan salah. Hal ini dikarenakan sistem belum dapat mendeteksi objek tersebut.

Salah satu objek yang membuat sistem kesulitan mendeteksi yaitu “Botol Minuman B” dimana botol minuman ini sering terdeteksi sebagai “Vas”. Selain itu tempat objek juga berpengaruh pada beberapa objek, yaitu “Kasur / Tempat Tidur”, pada jarak sekitar 2 meter dari “target

“objek” yang berada di kasur, sistem hanya mendeteksi “Tempat Tidur” dan tidak ada objek lain di tempat tersebut.

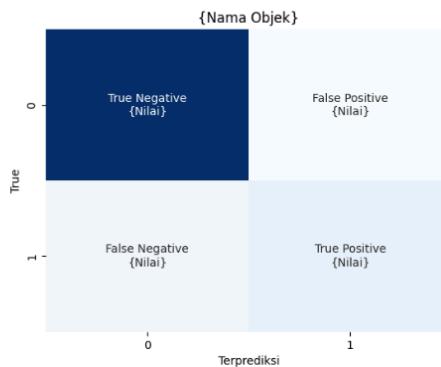
2. Hasil Implementasi Sistem

Penulis juga berkerja sama dengan guru-guru SLB Dharma Putra Daha, dalam implementasi sistem untuk penyandang tunanetra dengan skenario berlokasiikan di dalam ruangan, alat-alat.

Model SUS memiliki 5 skala Likert yaitu “Sangat Tidak Setuju (STS)”, “Tidak Setuju (TS)”, “Netral (N)”, “Setuju (S)”, dan “Sangat Setuju (ST)”. Dari hasil skor yang diperoleh kemudian ditampilkan berdasarkan interpretasi dan aturan SUS serta melalui tahap perhitungan demi mendapatkan skor rata-rata dari semua responden atau pengguna. Hasil dari skor tersebut akan disesuaikan menggunakan rumus skala SUS, untuk mengetahui kategori apa yang telah diperoleh dari skor tes rata-rata.

B. Evaluasi

Sesuai yang diuraikan sebelumnya hasil sistem yang dilakukan peneliti memiliki kekurangan dalam mengenali objek. Maka dari itu peneliti menguji coba performa sistem menggunakan *Confusion Matrix*. Uji coba akan dilakukan menggunakan 35 file gambar yang terdapat objek-objek. Peneliti menulis semua objek yang terdapat dalam gambar lalu disimpan untuk membandingkan dengan hasil prediksi sistem. Selanjutnya peneliti memvisualisasikan 5 hasil perbandingan menggunakan grafik, dengan struktur pada gambar 4.3. Terakhir peneliti memaparkan semua hasil uji coba yang dilakukan dalam bentuk tabel (tabel 4.2) dan grafik.



Gambar 4.3 : Struktur Grafik Confusion Matrix

Gambar 4.2 merupakan struktur grafik yang digunakan oleh peneliti dengan penjelasan sebagai berikut:

- “{Nama Objek}”, dan “{Nilai}” merupakan nilai yang berubah-ubah menyesuaikan hasil *Confusion Matrix* dari sebuah objek.
- “*True Negative*” (*TN*) atau (0, 0) adalah jumlah total dari *file* dimana sistem memprediksi “tidak ada objek”, dan sebenarnya “tidak ada objek”.
- “*False Negative*” (*FN*) atau (1, 0) adalah jumlah total dari *file* yang diprediksi oleh sistem “tidak ada objek”, namun sebenarnya “terdapat objek”
- “*False Positive*” (*FP*) atau (0, 1) merupakan jumlah total *file* yang diprediksi oleh sistem “terdapat objek”, namun sebenarnya “tidak ada objek”
- “*True Positive*” atau (1, 1) adalah jumlah total *file* yang diprediksi “terdapat objek” oleh sistem, dan sebenarnya “terdapat objek”

Untuk mengetahui tingkat keakuratan sistem dalam mendeteksi sebuah objek. Peneliti menggunakan penjumlahan sebagai berikut:

$$\text{akurasi} = \frac{TP+TN}{P+N}$$

Dimana:

TP: jumlah *True Positive* atau hasil prediksi yang benar dan sebenarnya benar

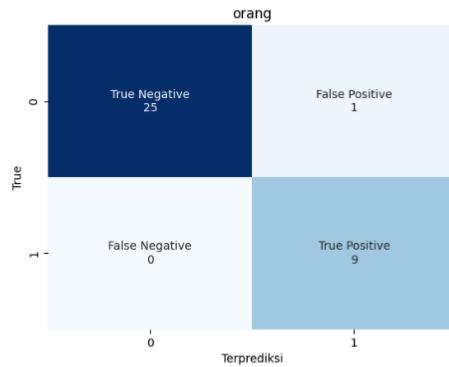
TN: jumlah *True Negative* atau hasil prediksi yang salah dan sebenarnya salah

P: atau *Positive* adalah hasil dari penjumlahan $TP + FN$

N: atau *Negative* adalah hasil dari penjumlahan $FP + TN$

Berikut 5 hasil grafik dari uji coba yang dilakukan oleh peneliti menggunakan *Confusion Matrix*:

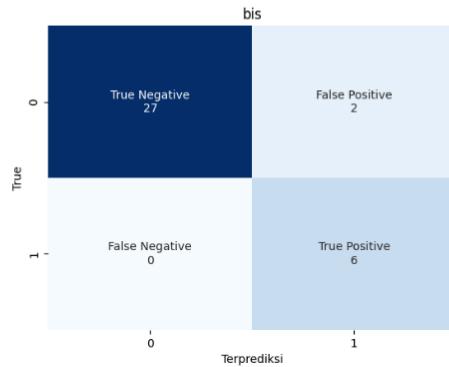
3. Confusion Matrix Objek “Orang”



Gambar 4.4 : Plot Confusion Matrix objek “orang”

Dalam gambar diatas dinyatakan bahwa sistem dapat mendekripsi dengan benar, terdapat objek “orang” didalam 9 file, dan 25 file yang tidak memiliki objek “orang”. Kesalahan sistem juga diketahui, dimana terdapat 1 file yang terprediksi “orang” oleh sistem, tetapi sebenarnya hanya 9 file.

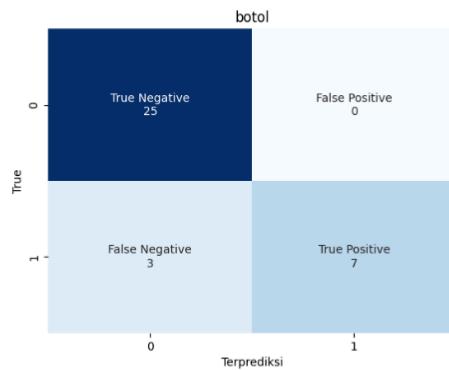
4. Confusion Matrix Objek “bis”



Gambar 4.5 : Plot Confusion Matrix objek “bis”

Pada uji coba objek “bis”, peneliti menyediakan 6 *file* yang memiliki objek “bis”, dan sistem memprediksi terdapat 8 *file*. Sehingga dapat disimpulkan bahwa sistem mampu memprediksi semua *file* yang disediakan oleh peneliti, namun terdapat kesalahan dimana sistem memprediksi adanya objek “bis” yang seharusnya tidak ada.

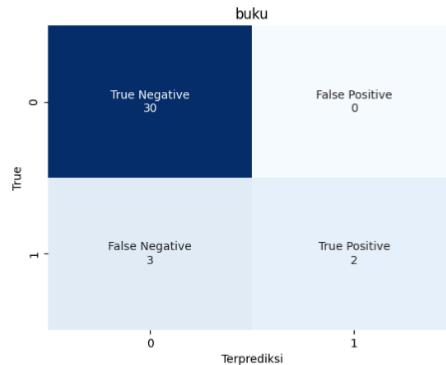
5. Confusion Matrix Objek “botol”



Gambar 4.6 : Plot Confusion Matrix objek “botol”

Gambar 4.6 memperlihatkan performa sistem dalam mendeteksi objek “botol”. Dari total 10 file gambar yang memiliki objek “botol”, hanya 7 yang terprediksi benar oleh sistem.

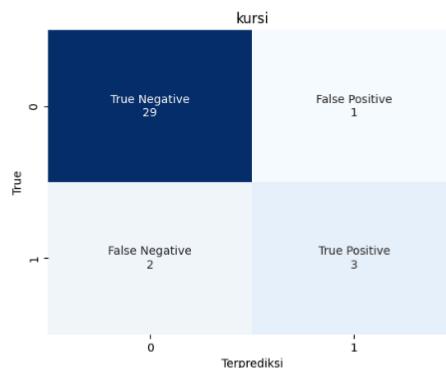
6. Confusion Matrix Objek “buku”



Gambar 4.7 : Plot Confusion Matrix objek “buku”

Dengan gambar 4.7 yang merupakan grafik dari hasil *Confusion Matrix* dari objek “buku”, dapat disimpulkan bahwa dari 5 file yang memiliki objek “buku”, hanya 2 file saja yang terdeteksi oleh sistem.

7. Confusion Matrix Objek “mobil”



Gambar 4.8 : Plot Confusion Matrix objek “kursi”

Gambar diatas merupakan grafik untuk hasil *Confusion Matrix* dari objek “kursi”. Dapat disimpulkan bahwa dari 5 *file* yang memiliki objek “kursi”. Sistem memprediksi adanya 4 *file*, dan 2 *file* tidak terdeteksi oleh sistem

8. Hasil Confusion Matrix Dengan Semua Objek Terdeteksi

Tabel 4.2 : Hasil confusion matrix dengan semua objek terdeteksi (tabel)

| Objek | Confusion Matrix | | | | | | | | Akuras i | | |
|-----------|------------------|--------|--------|--------|--------|--------|--------|--------|-------------|--|--|
| | T N | | F N | | T P | | F P | | | | |
| | T P | F P | F N | T N | P P | P N | N P | N N | | | |
| Bis | 27 | 0 | 6 | 2 | 6 | 8 | 9 | 2 | 0.9429 | | |
| Botol | 25 | 3 | 7 | 0 | 0 | 7 | 5 | 2 | 0.9143 | | |
| Buku | 30 | 3 | 2 | 0 | 5 | 2 | 0 | 3 | 0.9143 | | |
| Cangkir | 26 | 3 | 5 | 1 | 8 | 6 | 7 | 2 | 0.8857 | | |
| Dasi | 34 | 0 | 1 | 0 | 1 | 1 | 4 | 3 | 1.0000 | | |
| Gajah | 34 | 0 | 1 | 0 | 1 | 1 | 4 | 3 | 1.0000 | | |
| Gelas | 34 | 0 | 1 | 0 | 1 | 1 | 4 | 3 | 1.0000 | | |
| Kursi | 29 | 2 | 3 | 1 | 5 | 4 | 0 | 3 | 0.9143 | | |
| Laptop | 31 | 1 | 2 | 1 | 3 | 3 | 2 | 3 | 0.9429 | | |
| Lemari Es | 29 | 1 | 5 | 0 | 6 | 5 | 9 | 2 | 0.9714 | | |
| Mangkuk | 31 | 0 | 4 | 0 | 4 | 4 | 1 | 3 | 1.0000 | | |

| | | | | | | | | | |
|-----------------|----|---|---|---|---|---|--------|--------|--------|
| Meja Makan | 34 | 0 | 1 | 0 | 1 | 1 | 3 4 | 3 4 | 1.0000 |
| Microwave | 32 | 0 | 3 | 0 | 3 | 3 | 3 2 | 3 2 | 1.0000 |
| Mobil | 30 | 0 | 5 | 0 | 5 | 5 | 3 0 | 3 0 | 1.0000 |
| Mouse | 30 | 2 | 3 | 0 | 5 | 3 | 3 0 | 3 2 | 0.9429 |
| Orang | 25 | 0 | 9 | 1 | 9 | 0 | 6 2 | 5 2 | 0.9714 |
| Oven | 33 | 1 | 0 | 1 | 1 | 1 | 3 4 | 3 4 | 0.9429 |
| Papan Ketik | 32 | 0 | 2 | 1 | 2 | 3 | 3 3 | 2 | 0.9714 |
| Pemanggang Roti | 34 | 1 | 0 | 0 | 1 | 0 | 4 3 | 5 3 | 0.9714 |
| Ransel | 34 | 0 | 1 | 0 | 1 | 1 | 3 4 | 3 4 | 1.0000 |
| Remote | 33 | 0 | 1 | 1 | 1 | 2 | 3 4 | 3 3 | 0.9714 |
| Sofa | 32 | 0 | 3 | 0 | 3 | 3 | 2 3 | 2 3 | 1.0000 |
| Tanaman Pot | 34 | 0 | 1 | 0 | 1 | 1 | 3 4 | 3 4 | 1.0000 |
| Telepon Selular | 34 | 0 | 1 | 0 | 1 | 1 | 3 4 | 3 4 | 1.0000 |
| Televisi | 30 | 0 | 4 | 1 | 4 | 5 | 1 3 | 0 3 | 0.9714 |
| Tempat Tidur | 30 | 0 | 4 | 1 | 4 | 5 | 1 3 | 0 3 | 0.9714 |

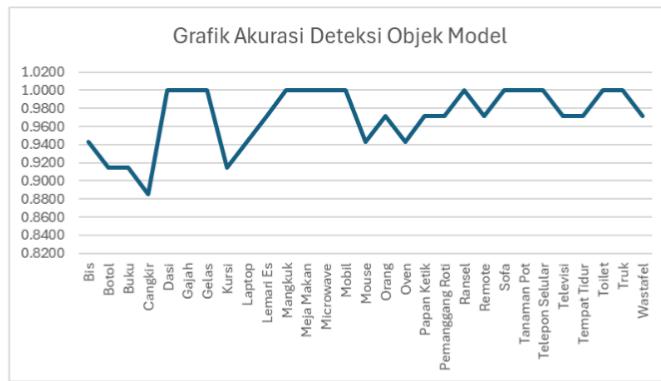
| | | | | | | | | | |
|----------|----|---|---|---|---|---|---|---|--------|
| Toilet | 33 | 0 | 2 | 0 | 2 | 2 | 3 | 3 | 1.0000 |
| Truk | 32 | 0 | 3 | 0 | 3 | 3 | 2 | 2 | 1.0000 |
| Wastafel | 31 | 1 | 3 | 0 | 4 | 3 | 1 | 2 | 0.9714 |

Tabel 4.1 merupakan hasil *confusion matrix* dan penjumlahan akurasi yang dilakukan penulis. Seperti yang tertulis sebelumnya bahwa pada uji coba *confusion matrix* terdapat 4 nilai yang dihasilkan yaitu “ TN ” (True Negative), “ FN ” (False Negative), “ $True Positive$ ” (True Positive), dan “ FP ” (False Positive). Kolom “Total” mencakup beberapa nilai yaitu (P) total file yang sebenarnya “terdapat objek”, (P') total file yang diprediksi “terdapat objek”, (N) total file yang sebenarnya “tidak ada objek”, dan (N') total file yang diprediksi “tidak ada objek”. Untuk mendapatkan nilai dari (P') dan (N') adalah sebagai berikut:

$$P' = TP + FP$$

$$N' = TN + FN$$

Pada kolom “Akurasi”, mempresentasikan nilai keakuratan sistem dalam mengenali objek yang diuji. Nilai tersebut dimulai dari 0 hingga 1, atau 0% hingga 100%.



Gambar 4.9 : Hasil *confusion matrix* dengan semua objek tedeteksi (grafik)

Gambar 4.9 mempresentasikan grafik dengan hasil dari tabel 4.1, dapat terlihat pada bagian bawah merupakan objek-objek yang diujikan, dan bagian kiri adalah nilai akurasi sistem dalam mendeteksi objek tersebut.

Dapat disimpulkan bahwa sistem mampu mendeteksi dengan nilai keakuratan maksimal 1.00, minimal 0.88, dan rata-rata 0.97. Perlu diketahui bahwa uji coba yang dilakukan oleh peneliti hanya menggunakan 35 *file* gambar. Dimana tidak menutup kemungkinan adanya nilai yang tidak sama, jika menggunakan *file* gambar yang lebih banyak ataupun lebih sedikit.

Pada kasus ini terlihat bahwa objek “Dasi”, “Gajah”, “Gelas”, “Meja Makan”, “Ransel”, “Tanaman Pot”, dan “Telepon Selular” memiliki nilai akurasi yang tinggi yaitu 1 atau 100%. Dikarenakan *file* gambar untuk uji coba yang sedikit.

5 18
BAB V
PENUTUP

A. Kesimpulan

Dari semua yang tertulis pada penelitian ini, peneliti menyimpulkan bahwa. Dengan menggunakan alat yang digunakan oleh peneliti yaitu *laptop*, pembuatan sistem deteksi objek bertujuan untuk membantu penyandang tunanetra mengenali objek sekitar, bekerja dengan lancar.

Sistem mampu mendeteksi secara *real-time* dengan resolusi 640x640 pada nilai rata-rata ³⁴ *fps* 30. Selain itu, dari hasil uji coba yang dilakukan peneliti menggunakan 35 gambar dan 29 objek sebagai validasi, *model* deteksi objek memiliki nilai rata-rata akurasi diatas ³⁶ 85%. Sehingga dapat disimpulkan bahwa *model YOLOv8n* bekerja dengan baik pada sistem.

Selain mendeteksi objek, peneliti juga merancang fungsi tambahan yaitu navigasi, dan *text-to-speech*. Dimana tujuan dari fungsi tersebut tidak lain adalah untuk membantu penyandang tunanetra dalam mengenali objek sekitar dengan cara memberitahu secara langsung. Fungsi-fungsi yang disebutkan juga bekerja baik pada sistem dengan bantuan koneksi internet.

B. Saran

Dari hasil penelitian yang dilakukan peneliti yaitu pembuatan sistem deteksi objek untuk membantu penyandang tunanetra menggunakan *YOLOv8*, sistem dapat bekerja dengan lancar. Namun sistem yang dibuat oleh peneliti juga memiliki kekurangan, dalam penggunaan sistem, dan validasi deteksi objek yang menggunakan validasi sedikit.

Karena keterbatasan alat, sistem hanya dapat bekerja di perangkat *desktop* dan memerlukan koneksi internet sehingga akan menimbulkan masalah lain bagi penyandang tunanetra, yaitu susahnya membawa perangkat yang besar dalam penggunaan sehari-hari. Dimana perangkat sistem seharusnya dapat lebih mudah dibawa, seperti contoh *smartphone*, ringan serta memiliki fitur yang dapat membantu, salah satunya adalah senter ketika berada di ruangan gelap. Dengan memanfaatkan fitur dari *smartphone*, memungkinkan sistem untuk membantu penyandang tunanetra dalam situasi apapun.

Hasil uji coba model objek deteksi, yang menggunakan 35 gambar dan 29 objek sebagai validasi, dan memiliki rata-rata akurasi diatas 85%. Peneliti menyimpulkan bahwa efisiensi pengenalan objek masih jauh dari sempurna, terdapat banyak objek yang memiliki akurasi dibawah rata-rata.

Dengan segala kekurangan pada penelitian ini, peneliti berharap bahwa di kedepannya. Agar peneliti lain yang memiliki masalah relevan, dapat memanfaatkan penelitian ini untuk pengembangan baru maupun penelitian lanjutan, demi tercapainya tujuan yang sama.

ORIGINALITY REPORT

13%
SIMILARITY INDEX

11%
INTERNET SOURCES

5%
PUBLICATIONS

5%
STUDENT PAPERS

PRIMARY SOURCES

- | | | |
|----|---|-----|
| 1 | unsri.portalgaruda.org Internet Source | 1% |
| 2 | journal.student.uny.ac.id Internet Source | 1% |
| 3 | www.jurnal.stmik-yadika.ac.id Internet Source | 1% |
| 4 | j-ptiik.ub.ac.id Internet Source | 1% |
| 5 | digilibadmin.unismuh.ac.id Internet Source | 1% |
| 6 | 123dok.com Internet Source | 1% |
| 7 | Submitted to UIN Syarif Hidayatullah Jakarta Student Paper | 1% |
| 8 | repository.unpkediri.ac.id Internet Source | <1% |
| 9 | ojs.trigunadharma.ac.id Internet Source | <1% |
| 10 | perpustakaan.jak-stik.ac.id Internet Source | <1% |
| 11 | repository.polinela.ac.id Internet Source | <1% |
| 12 | Submitted to Universitas Sumatera Utara Student Paper | |

| | | | |
|----|---|--|------|
| | | | <1 % |
| 13 | jurnal.unmuhjember.ac.id Internet Source | | <1 % |
| 14 | www.scribd.com Internet Source | | <1 % |
| 15 | es.scribd.com Internet Source | | <1 % |
| 16 | www.mdpi.com Internet Source | | <1 % |
| 17 | Submitted to Soongsil University Student Paper | | <1 % |
| 18 | repository.uin-suska.ac.id Internet Source | | <1 % |
| 19 | docplayer.info Internet Source | | <1 % |
| 20 | publikasi.dinus.ac.id Internet Source | | <1 % |
| 21 | Submitted to Transylvania University Student Paper | | <1 % |
| 22 | jyx.jyu.fi Internet Source | | <1 % |
| 23 | tunasbangsa.ac.id Internet Source | | <1 % |
| 24 | elibrary.unikom.ac.id Internet Source | | <1 % |
| 25 | adoc.pub Internet Source | | <1 % |
| | repository.polman-babel.ac.id | | |

| | | |
|----|---|------|
| 26 | Internet Source | <1 % |
| 27 | www.techrxiv.org Internet Source | <1 % |
| 28 | Farrell Ega Santoso, Ika Ratna Indra Astutik. "Decision Support System For Computer Recommendations Using Web-Based Fuzzy Tahani Logic Method", Procedia of Engineering and Life Science, 2021 Publication | <1 % |
| 29 | Flendio Marselino Tambajong Flendio, Rizal Sengkey, Sary D. E. Paturusi. "Analisa Pengalaman Pengguna pada Aplikasi Media Sosial Tiktok Shop Menggunakan Metode System Usability Scale", Jurnal Teknik Informatika, 2024 Publication | <1 % |
| 30 | chinese.scitechnol.com Internet Source | <1 % |
| 31 | link.springer.com Internet Source | <1 % |
| 32 | repository.trisakti.ac.id Internet Source | <1 % |
| 33 | Taufiq Hidayat Nazar, Raha Bahari. "AKAD JUAL BELI AKUN GAME ONLINE MOBILE LEGENDS DALAM PERSPEKTIF HUKUM EKONOMI SYARIAH", Mu'amalah : Jurnal Hukum Ekonomi Syariah, 2022 Publication | <1 % |
| 34 | carpictc.blogspot.com Internet Source | <1 % |

| | | |
|----|---|------|
| 35 | id.123dok.com Internet Source | <1 % |
| 36 | pt.scribd.com Internet Source | <1 % |
| 37 | repository.maranatha.edu Internet Source | <1 % |
| 38 | repository.radenintan.ac.id Internet Source | <1 % |
| 39 | www.benhil.net Internet Source | <1 % |
| 40 | www.slideshare.net Internet Source | <1 % |
| 41 | Robinso Rohi, Jefonses Pote, Alfrian Talakua. "PERANCANGAN DAN IMPLEMENTASI SISTEM INFORMASI PERPUSTAKAAN BERBASIS WEBSITE MENGGUNAKAN METODE WATERFALL DI SD MASEHI KAMBANIRU 2", Jurnal Informatika dan Teknik Elektro Terapan, 2022 Publication | <1 % |

Exclude quotes Off
Exclude bibliography Off

Exclude matches Off